

85



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

CAMPUS ARAGON

993557

SISTEMA VIRTUAL DE ENSEÑANZA-APRENDIZAJE
EN INGENIERIA BASICA
(SIVEA)

Acumulado de un Disco Compacto.

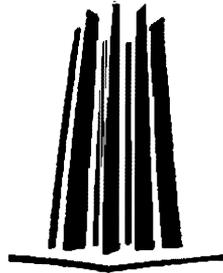
T E S I S

QUE PARA OBTENER EL TITULO DE:

INGENIERO MECANICO ELECTRICISTA

P R E S E N T A

IGNACIO DE LA ROSA ROSAS





Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

CAMPUS ARAGON

**SISTEMA VIRTUAL DE ENSEÑANZA – APRENDIZAJE EN
INGENIERIA BASICA.
(SIVEA)**

TESIS

**QUE PARA OBTENER EL TITULO DE
INGENIERO MECANICO ELECTRICISTA**

PRESENTA

IGNACIO DE LA ROSA ROSAS.

Con respeto:

A Dios:

Gracias señor por permitirme llegar hasta aquí siempre bajo tus leyes y mandamientos y con todas las bendiciones que me has otorgado.

A mis hermanas:

Marcela

Guadalupe

Nancy

Román

A cada uno de ellas les agradezco el ser como son y el vivir cada etapa de nuestras vidas juntos. Dado que Dios me dio conciencia y hasta que él nos lo permita.

Al

Dr. Salvador Rivera Domínguez

La Sra. Celia Domínguez Vázquez.

Que con sus consejos y apoyo han sido como unos padres para mí. Gracias

A

Horacio León Camacho

Por ser un verdadero amigo aquel en el que se puede confiar y siempre tiene tiempo para escucharme. Gracias

Y especialmente con todo mi amor

A Mónica.

Quien ha sido mi compañera amiga y esposa quien paz y estuvo siempre a mi lado en el momento de realizar este trabajo para los sacrificios y devotos, dándome amor y alegrías en todo momento. Su apoyo constante en mis momentos de flaqueza borraron las dudas y supo darme los ánimos para seguir adelante. Mami amor lo logramos

A mis padres:

Ing. Ignacio De la rosa Valle

Sra. Margarita Rosas Islas

A quienes les debo la vida y que siempre procuraron darme lo mejor que con su sacrificio y esfuerzo lograron hacer de su hijo un profesionalista. gracias papas este triunfo también es suyo.

A mi hija:

Angela Margot.

Quien llegó como una bendición mandada por Dios y que por ti pequeña seguiré siempre adelante.

A:

Ing. Enrique A. Islas Martínez

Sra. María L. Gaytán.

Y a toda la gente de Teconost:

Por abrir una puerta cuando otra se me había cerrado por que con sus consejos y apoyo han despertado en mí el deber de hacer mi propia carrera y tratarlo con dedicación y esfuerzo, tal como ustedes lo han hecho.

Al Ing. Juan Gastalón P. por su paciencia y comprensión por entender y apoyarme en la realización de este trabajo.

INDICE

<u>INTRODUCCION.</u>	1
<u>CAPITULO I</u>	
CONCEPTOS BÁSICOS	3
Educación, La tendencia al futuro.	4
Educación virtual.	23
Desarrollo y utilización de sistemas enseñanza asistida por computadora (EAC).	33
Algunos Ejemplos.	46
<u>CAPITULO II</u>	
ESTUDIO DE CAMPO.	52
Consideraciones generales.	53
Investigación y análisis del problema.	57
Planteamiento del problema.	
Resultados.	60
<u>CAPITULO III</u>	
DISEÑO DEL SISTEMA.	77
Diseño de sistemas informáticos.	78
Metodologías de desarrollo de Software.	82
Desarrollo de un sistema orientado a objetos.	101

CAPITULO IV

OPERACIÓN DE SIVEA Y SUS MODULOS	117
Que es SIVEA.	119
Requisitos del sistema.	119
Como esta formado SIVEA.	121
Que y cuales son los módulos de SIVEA.	124
Operación básica.	126
Manejo de los módulos.	131

CAPITULO V

APLICACIONES Y RESULTADOS	148
CONCLUSIONES	150

ANEXOS

ANEXO A	II
En busca de la clase virtual.	
ANEXO B	V
Guía de evaluación y examen de aplicación.	
ANEXO C	XII
Introducción al lenguaje de programación VB.	
ANEXO D	XXXVIII
Código de SIVEA.	
ANEXO E	LXIX
Ayuda del sistema GENEVAL.	

<u>BIBLIOGRAFIA</u>	A
----------------------------	----------

<u>GLOSARIO DE TERMINOS.</u>	C
-------------------------------------	----------

UNAM

INGENIERIA MECANICA ELECTRICA.

TEMA DE TESIS

Que como prueba escrita de su examen Profesional para obtener el titulo de Ingeniero Mecánico eléctrico, deberá desarrollar el C. Ignacio de la Rosa Rosas.

**SISTEMA VIRTUAL DE ENSEÑANZA APRENDIZAJE EN INGENIERIA BASICA.
SIVEA.**

DELIMITACION DEL TEMA

El presente sistema se enfoca en el uso de un software que permita el estudio, la enseñanza y la evaluación de las áreas de matemáticas y física esenciales en el estudio de la ingeniería.

OBJETIVO

Diseñar un sistema que sea una herramienta de apoyo en el proceso de enseñanza – aprendizaje para aplicarse en el área de la ingeniería.

ALCANCE.

Este sistema podrá ser aplicado en el área básica de la ingeniería, en el estudio de las matemáticas y la física.

DESCRIPCION

El presente trabajo cubre a través de múltiples y amplias presentaciones en forma de sencillos programas, el material didáctico de la formación profesional en el campo laboral e Industrial. Además esto constituye, una fuente de información valiosa para los docentes, especialistas y técnicos ya en su trabajo práctico sobre la forma apropiada y según las reglas de proceder. El sistema puede utilizarse tanto para la enseñanza práctica como para el periodo de formación.

Los múltiples módulos del sistema que se presentan con una gama de imágenes, conceptos y la división de tareas, facilitan el manejo y comprensión de los temas desarrollados.

En la configuración de los módulos del sistema se ha dado una especial importancia a la claridad de los conceptos tratados, que se logra con la correspondencia de imágenes y texto esto es posible gracias a la multimedia.

Para la ilustración de los conceptos aquí tratados, los módulos cuentan con una gama de ejemplos que facilitan su comprensión. Todos los cálculos se han realizado por computadora lo que permite una exactitud y una aproximación a la realidad que le da un matiz práctico a este.



INTRODUCCION

Hoy en día los grandes pasos que ha dado el crecimiento de la tecnología han generado una revolución de información, lo cual ha hecho indispensable el uso de las computadoras. La aparición de las computadoras se ha impuesto en todos los ambientes sociales, su fácil utilización, su bajo precio y su tamaño, están modificando el ritmo de las relaciones sociales. Las aplicaciones de estas son variadas y cada vez más sofisticadas, El uso de las computadoras ha ido creciendo y se ha ido difundiendo su uso en diversos campos. Se les puede encontrar desde ambientes industriales para el desarrollo de complejos procesos de manufactura, hasta el uso de estas en los hogares para el entretenimiento; su uso se ha propagado en casi cualquier actividad humana y en diversas instituciones y empresas; las instituciones educativas no sido la excepción, actualmente con el desarrollo de los programas de educación a distancia y específicamente el desarrollo de la educación virtual se ha generado una transformación en la forma de enseñar y aprender; en diversos países desarrollados se está diseñando programas, sistemas y aplicaciones para facilitar este proceso.

Los docentes en nuestro país, están en lo general, alejados de este tema, a la que ven, como algo misterioso, y con un alto grado de aprendizaje, o como una técnica de fuerte capacidad educativa, que puede llevar a confusiones, así como también se cae en el error de pensar que un software de este tipo es demasiado costoso.

Es por esto, que se requiere el desarrollo y creación de tecnología, propiamente mexicana, y que en este trabajo de tesis, pretendo mostrar las grandes aplicaciones y el potencial educativo que nos ofrece la programación y el uso de las computadoras como una herramienta aplicada a la enseñanza – aprendizaje (E-A). Este trabajo ha sido un esfuerzo de cuatro años, en los que se han realizado mejoras y críticas.

En el capítulo uno se explica como se ha ido transformando la educación y cual sería su tendencia al futuro, Mostrando como las instituciones educativas están buscando formas alternativas de enseñanza con un menor costo, y sobre todo tratando de difundirla a un número mayor de individuos, entre las tecnologías educativas modernas destaca la enseñanza a través de la educación virtual, por lo cual se explica que es la Universidad y la educación virtual, así como una propuesta de cómo deberá ser el docente (facilitador) y el estudiante (aprendiz) en un ambiente virtual de aprendizaje. Se realiza un comparativo entre la educación tradicional y la educación a distancia, Enseñando las virtudes y defectos que estas tienen. Es en este tema donde se desarrolla el concepto de Enseñanza Asistida por computadora o EAC por sus siglas, se explica que es y cual es su aplicación en la educación; Se explica, además el rol que juega la computadora en el proceso E – A. Finalmente se muestra algunos ejemplos de este tipo de tecnología y donde se están usando.

En el capítulo dos se describe el estudio de campo que se realizó en la Universidad Nacional Autónoma de México, Campus Aragón, en el área de ingeniería, específicamente el área Mecánica – Eléctrica, para determinar las necesidades y requerimientos que debería tener el sistema por desarrollar, cabe mencionar que mucho del trabajo desarrollado en este sistema parte de la experiencia adquirida durante mi etapa de estudiante dentro de la preparatoria y en la universidad y de mi experiencia como docente, primero en la Escuela Nacional Preparatoria durante seis años y después en el sector privado en el Centro de Desarrollo Industrial donde actualmente laboro. Se muestra la evaluación aplicada, así como los resultados obtenidos.

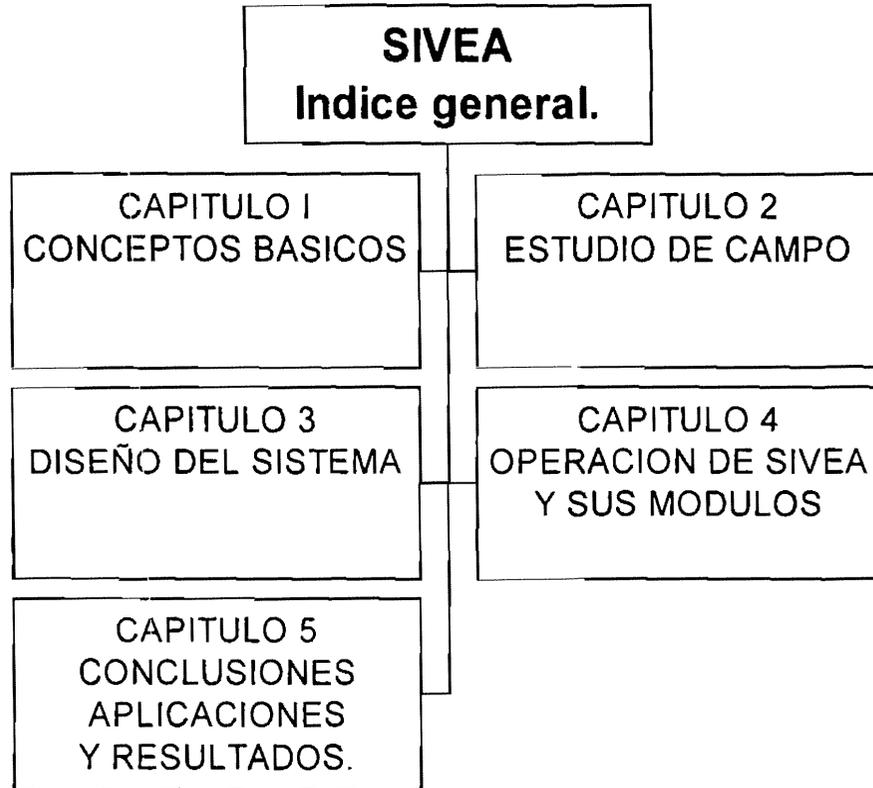
Partiendo de los resultados obtenidos en la investigación desarrollada, en el capítulo tres, se trata el diseño del sistema, se explica por que el uso de programación orientada objetos (OO), que es la programación OO, como se diseña y desarrolla un sistema OO.

Una vez desarrollado el sistema; en el capítulo cuatro se define que es el Sistema Virtual de Enseñanza – Aprendizaje en ingeniería Básica o SIVEA, cuales son sus componentes, cuales son los requisitos para su uso y sus características. Este capítulo podría decirse que es el manual de operación para el sistema, ya que explica detalladamente el uso y manejo de sus módulos, así como todas las herramientas que contiene para sacar un mayor provecho de él.

Finalmente en el capítulo cinco, se analizan las ventajas y desventajas, así como las aplicaciones que tiene SIVEA, se muestran algunos de los resultados que se han obtenido al usarlo así como posibles áreas de mejora. No se puede caer en el caso extremo de querer realizar toda enseñanza a través del computador. Debemos tener presente que es un medio, y no el único y por lo tanto debe convivir con los libros, pizarras, murales, videos, etc., esto podría ser caso de otro estudio. Este trabajo pretende ser una aplicación no solo en universidades, también en la industria ya que en este último punto es donde se ha encontrado una fuerte aplicación, a través de cursos de capacitación al personal obrero y administrativo, lo cual ha reflejado un aprendizaje rápido, eficiente y con calidad, esto ha permitido a los aprendices (Empleados) tener mejores armas para desarrollarse en el campo laboral, así como en el campo social, ya que a través de la educación se provoca un mejor nivel de vida

ORGANIGRAMA PRINCIPAL

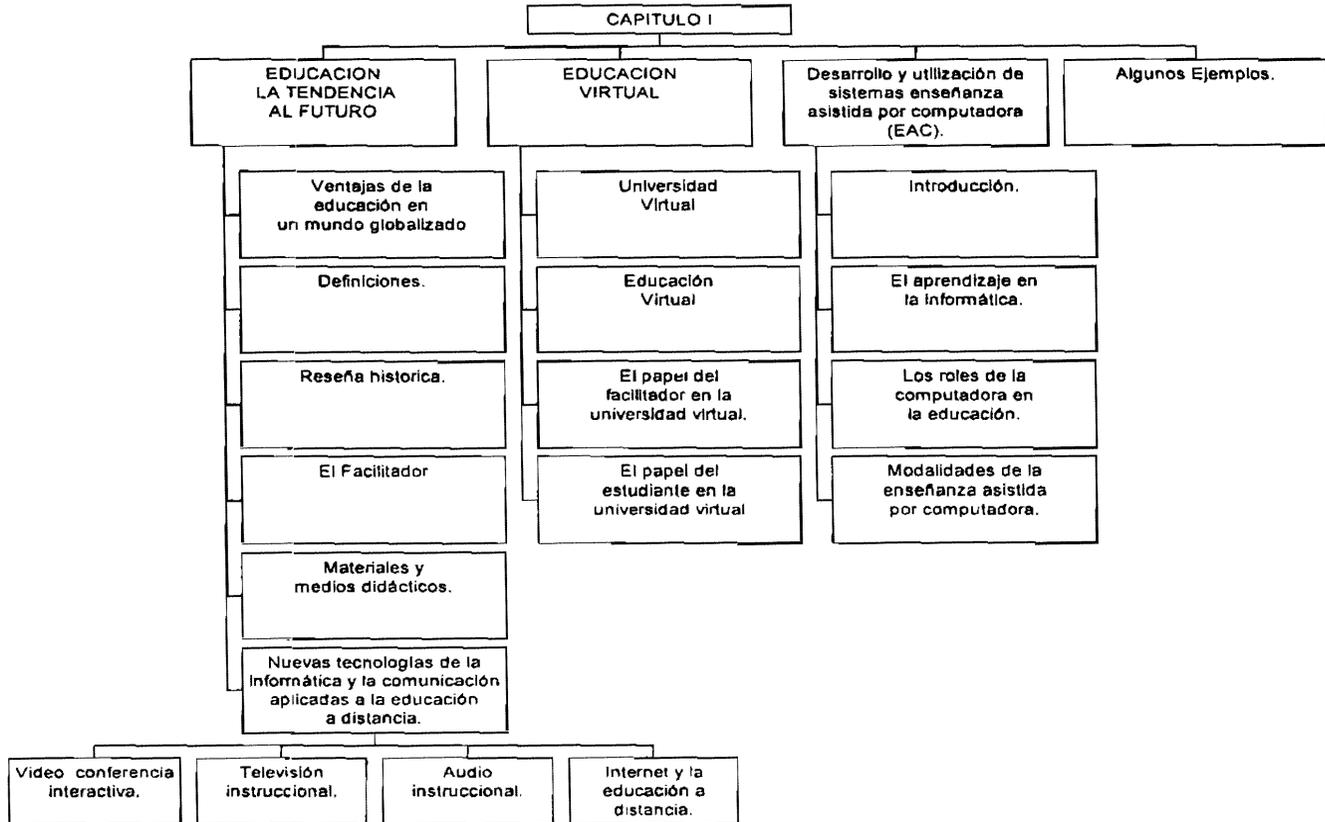
SISTEMA VIRTUAL DE ENSEÑANZA APRENDIZAJE.



CAPITULO I

CONCEPTOS BASICOS

CONCEPTOS BASICOS





EDUCACION, LA TENDENCIA AL FUTURO.

"Como la fuerte demanda de educación existente no puede ni podrá encontrar respuesta adecuada en los sistemas tradicionales, se debe recurrir a nuevas formas educativas capaces de satisfacer tales demandas".

Es así como en América Latina, al percibir que la educación constituye un factor importante para alcanzar el desarrollo, surgen los sistemas de Educación a Distancia como una alternativa más que viable para superar el subdesarrollo.

(Casas, Miguel. 1989)

Ventajas de la Educación en un mundo globalizado.

La revolución tecnológica está transformando nuestra sociedad de modo tan significativo como las dos revoluciones industriales lo hicieron anteriormente y se establecerá sólida y justificadamente como la tercera revolución de la historia moderna, excepto que, ahora, lo que se está transfiriendo a las máquinas es el trabajo cerebral y no el muscular. Algunos de los aspectos de esta transformación forman un parte del fenómeno denominado **GLOBALIZACION**.

La globalización permite que el mundo funcione como un mercado de aldea. Actualmente, a través de computadoras, se puede comprar, vender e intercambiar libremente bienes, información y servicios por todo el mundo.



El sistema nervioso de la economía mundial empieza a transmitir sus señales a través de una red de módems, fibra óptica, e-mails, satélites y teléfonos inalámbricos.

Por otro lado, la globalización está despegando a los individuos de su sentido de pertenencia nacional, para incorporarlos a un universo económico, técnico y espiritual común a todos los hombres. Nuestro planeta se está convirtiendo rápidamente en un círculo cuyo centro no está en ninguna parte, y su periferia por todos lados. En este sentido, la red Internet representa la aceptación más común de la globalización inmediata.

Actualmente la globalización está generando una masa de individuos cuya capacidad de adaptación a la velocidad de la supercarretera de la información los lleva a una estación: los desocupados; los comerciantes minoristas; los empleados de empresas públicas privatizadas y los profesionales de clase media que no pueden actualizar los conocimientos que la demanda laboral exige.

Debido al atraso tecnológico y a la dificultad para acceder al mercado de capitales. Así cada vez hay menos empresarios que antes, y los que quedan, lo hacen aumentando la productividad de sus trabajadores, no el número de personal.



Por cada trabajador que sale del mercado hay una familia que tiene que cambiar sus expectativas de vida y de consumo.

Estamos en una nueva fase de la historia mundial en la que será necesario cada vez un número menor de trabajadores para producir los bienes y servicios de la población mundial. Las tecnologías de la información y las fuerzas del mercado, están polarizando a la población mundial en dos frentes irreconciliables: una elite cosmopolita de *analistas simbólicos*, que controla las tecnologías y las fuerzas de la producción, y un grupo cada vez mayor de desempleados, con poca o ninguna esperanza de reubicarse en la nueva economía global.

Para contrarrestar estos efectos, la mayoría de los especialistas coinciden en que el factor que decidirá el destino de las naciones en la economía global será el nivel educativo de sus poblaciones.

Este trabajo facilitará a todos aquellos que quieren elevar su nivel de competencia hasta alcanzar el perfil de este nuevo modelo de desarrollo, accederá a los conocimientos necesarios para el caso.

Bill Clinton, en su última visita a la Argentina, enumeró las prioridades de su gobierno: **Educación y capacitación permanente** para que todos puedan participar de los beneficios del libre mercado en los lugares a los que aún no ha llegado. Por su parte, el primer ministro inglés, anunció que las prioridades de su gobierno son tres: **Educación, Educación y Educación.**

En el futuro las personas, aunque no tengan que ir a la escuela, deberán acceder a una educación que les permita adquirir destrezas laborales específicas; ya que los nuevos empleos exigen conocimiento, imaginación e inteligencia intensiva.

En este sentido, los gobernantes comienzan a replantearse el rol de la educación y las modalidades que adoptará en sus países, con el objetivo de volverla más eficiente frente a la necesidad de democratizar el acceso al conocimiento y mejorar el aprendizaje.



En el mundo del mañana, donde la información será más abundante, facilitar el acceso a la Educación, no sólo es una obligación moral de la sociedad, sino también un medio para incrementar la productividad y mejorar la calidad de vida de las personas.

Actualmente en los Estados Unidos se cree que las inversiones en educación se están tornando ineficientes ya que no producen mejoras visibles en contenido ni en calidad, por lo que piensan que su sistema educativo está sufriendo una crisis profunda, y para contrarrestar posibles efectos nocivos, ya se ha comenzado a impartir clases y conferencias vía satélite, por una fracción del costo de la educación convencional.

Por esta razón, se estima que la universidad norteamericana no sobrevivirá como institución residencial ya que sus edificios de hoy, son totalmente antieconómicos, inadecuados e innecesarios a la hora de expandir la oferta educativa, bajo el sistema tradicional.

Por todo lo expuesto, en los países desarrollados se emplea en forma creciente la Educación a Distancia como un instrumento que permite cambiar en forma rápida y eficiente la enseñanza impartida desde un modelo de escuela común y centralizada, a uno flexible y descentralizado, que mejora la dinámica social y facilita el acceso al conocimiento.



Es necesario que, en países en desarrollo, la Educación a Distancia cobre mayor importancia ya que es más fácil lograr que la escuela vaya a los alumnos y no los alumnos a la escuela.

Con la Educación a Distancia dificultades de tiempo, distancia y dinero, se resuelven fácilmente y no sólo esto, la Educación a Distancia brinda a las personas la oportunidad de tomar cursos, de escuchar a conferencistas extranjeros, y la posibilidad de contactar a otros estudiantes de distintas escalas sociales, culturales y económicas.

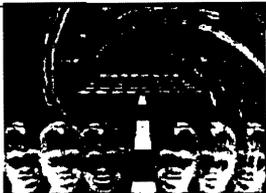
Como resultado, los estudiantes no sólo acceden fácilmente al conocimiento, sino también adquieren una nueva habilidad social:

"La posibilidad de comunicarse y colaborar con una amplia gama de compañeros que nunca antes habrían conocido".



Aplicando nuevas tecnologías en los programas de enseñanza a distancia se logra una mayor integración del alumno en el proceso de aprendizaje, se crea un nuevo sistema de enseñanza que potencia al alumno y le permite estar más involucrado en el proceso de enseñanza.

Las posibilidades de la Educación a Distancia son muy prometedoras y en determinadas circunstancias y situaciones, la enseñanza a distancia tiene mucho sentido.



"Si el problema es la falta de escuela, la alternativa será una escuela virtual".

La conclusión que se desprende de este análisis es que incorporando la Educación a Distancia a los programas, se tienen amplias perspectivas de mejorar la educación impartida, no por sustitución de las maneras de enseñar y de aprender que implican la proximidad física, sino por su potenciación y realce.

Para que esto se lleve a cabo con éxito, una nueva cantidad de temas deben ser tenidos en cuenta, ya que la Educación implica mucho más que la transferencia de conocimientos de maestro a alumno.



Una educación efectiva se logra encendiendo la llama del aprendizaje en el corazón de los estudiantes, para ello es necesario construir vínculos entre estudiantes y maestros. La dedicación y la capacidad de los maestros fue, es y seguirá siendo la herramienta educativa más importante.



Definiciones

La Educación a Distancia es un término genérico, difícil de definir, ya que en él se incluyen las estrategias de enseñanza aprendizaje, que en el mundo se denominan de diferentes formas.

Sin embargo, de todas las definiciones existentes se pueden extraer las siguientes extraídas de:

BARRANTES ECHAVARRÍA, Rodrigo. (1992). Educación a Distancia. EUNED. San José de Costa Rica. Es una estrategia educativa basada en la aplicación de la tecnología del aprendizaje sin la limitación del lugar, tiempo, ocupación o edad de los estudiantes.

(José Luis García Llamas, 1986). Son aquellas formas de estudio que no son guiadas o controladas directamente por la presencia de un profesor en el aula, pero se beneficia de la planificación y guía de los tutores a través de un medio de comunicación que permita la interrelación profesor - alumno.

(Pío Navarro Alcalá, 1980). La Educación a Distancia es una estrategia para operacionalizar los principios y fines de la educación permanente y abierta, de manera que cualquier persona, independiente del tiempo y del espacio, pueda convertirse en sujeto protagonista de su aprendizaje.

(Miguel A. Ramos Martínez, 1985).

Reseña Histórica

Según estudios, la Educación a Distancia organizada se remonta al siglo XVIII, con un anuncio publicado en 1728 por la Gaceta de Boston, en donde se refería a un material auto- instructivo para ser enviado a los estudiantes con posibilidad de tutorías por correspondencia.

En 1840, Isaac Pitman organizó en Inglaterra un intento rudimentario de educación por correspondencia. En 1843 se formó la "Phonographic Correspondence Society" para encargarse de correcciones de ejercicios taquigráficos.

Menos académico fue el intento de enseñar minería y prevención de accidentes mineros por el Mining Herald, un periódico de Pennsylvania. Fue Thomas Foster el que tuvo esta iniciativa y esto constituyó el comienzo de las Escuelas Internacionales por correspondencia (ICS) de Scranton, Pennsylvania.

En la Europa Occidental y América del Norte, la Educación a Distancia empezó en las urbes industriales del Siglo XIX, con el fin de atender a las minorías, que por diferentes motivos, no asistieron a escuelas ordinarias.

Al finalizar la Segunda Guerra Mundial, se produjo una expansión de esta modalidad para facilitar el acceso a los centros educativos en todos los niveles, especialmente en los países industrializados occidentales, en los centros europeos y en las naciones en desarrollo "tercermundistas". Esto obedeció al incremento de la demanda de mano de obra calificada registrada.

La educación universitaria empieza a utilizar esta modalidad para facilitar el acceso a ciertas profesiones y ocupaciones a los estudiantes a menor costo.

Como la educación tradicional se caracteriza por su elevado costo, se pensó en la Educación a Distancia como una forma de ofrecer cupos adicionales en una forma más rápida y económica, especialmente en países muy grandes y de escasa población, que intentaban vencer la distancia, como Suecia y Canadá.

Así, en la década de los '60 y '70 se ha dado una marcada expansión de la Educación a Distancia, tanto en el terreno práctico como en el teórico. Entre 1960 y 1975 se fundaron en África más de veinte instituciones de Educación a Distancia. Entre 1972 y 1980, en Australia, el número de instituciones a distancia pasó de 15 a 48. Sin embargo es en los países industrializados o desarrollados como Canadá, Inglaterra, Alemania, los Estados Unidos y Japón, donde se le dio más valor a esta modalidad.

Ya desde sus inicios este tipo de enseñanza tuvo que enfrentarse a la desconfianza de quienes veían en ésta una "oportunidad menor", o quienes temían el desarrollo de un sistema más flexible, más dinámico y por supuesto más atractivo.

Hay que señalar que la educación por correspondencia ha servido de base a las diversas opciones que se han materializado en este campo y que en general pretenden ampliar el acceso a la enseñanza, fruto de una nueva actitud pedagógica que ubica al alumno en primera fila y a la institución en la segunda.

No cabe duda que existe un creciente interés por la Educación a Distancia que plantea todo un desafío en el mundo por conocer más sobre esta metodología

Enseñanza a Distancia y Estudio Independiente: Características principales

Se puede observar que a partir de la separación de maestro y estudiante se deriva el concepto de "Distancia", dentro del cual, la noción de Estudio Independiente es primordial. Por lo tanto, el Estudio Independiente es un proceso motivado por los objetivos de cada estudiante y recompensado por sus valores intrínsecos.

Las siguientes características, describen en pocas palabras al Estudio Independiente:

Debido a la separación física, la interacción entre los estudiantes y maestros esta mediatizada.

La utilización de distintos medios para compensar la separación física del estudiante es esencial en el estudio independiente. Esta forma permite al alumno planificar su propio aprendizaje, proponiéndose metas, buscando y utilizando los recursos que estén a su alcance. Si el estudiante tiene un grupo de objetivos, puede alcanzarlos haciendo uso de sus destrezas, según las metas que él se propone y no solamente las que le indique su maestro.

Educación a Distancia y Educación Presencial: Principales Diferencias

La Educación a Distancia ha surgido como un intento de dar respuesta a las nuevas demandas sociales que la Educación Presencial no ha podido atender, pero resulta incorrecto suponer que aquella pueda sustituir totalmente a esta última.

Ambas formas educativas pueden beneficiarse mutuamente de su coexistencia y acción.

En el siguiente cuadro se detallan las principales diferencias entre:

Educación Presencial	Educación a Distancia
<p>El profesor y los estudiantes están físicamente presentes en un mismo espacio y tiempo (durante las clases).</p> <p>La voz del profesor y su expresión corporal son los medios de comunicación por excelencia. Se les llama presenciales a estos medios porque restringen la comunicación a un aquí y a un ahora.</p> <p>Otros medios visuales y sonoros son muy poco utilizados en la clase convencional y sólo sirven como apoyos didácticos o para complementar la acción del profesor.</p> <p>La comunicación oral característica en la enseñanza convencional está acompañada normalmente por gestos y movimientos de la comunicación no verbal.</p> <p>La relación directa, presencial, de los que se comunican hace que el diálogo pueda producirse también aquí y ahora de manera inmediata.</p>	<p>El profesor y los estudiantes pueden no estar presentes físicamente en el mismo espacio ni en el mismo tiempo.</p> <p>Para que la comunicación se produzca, es necesario crear elementos mediadores entre el docente y el alumno.</p> <p>La voz y el esquema temporal, o son sustituidos por otros medios no-presenciales, o serán registrados en grabaciones sonoras y visuales para ser transmitidos luego a otro espacio y en otro tiempo. Los medios no son simples ayudas didácticas sino portadores de conocimiento que sustituyen al profesor.</p> <p>Adquieren gran importancia los medios como ser la palabra escrita (dominante por antonomasia), además de la radio, la televisión y otros medios audiovisuales.</p> <p>La relación no-presencial de los que se comunican, es una forma de diálogo que por no acontecer aquí ni ahora, puede llamarse "diálogo diferido". O sea, el comunicador debe continuar un mensaje completo y esperar un tiempo para recibir la comunicación, de retorno en forma similar, al igual que ocurre con una carta.</p>



Materiales y Medios Didácticos

Los recursos o medios didácticos son el nexo entre las palabras y la realidad. La ordenación de los recursos es tarea compleja, ya que son el soporte que da coherencia al proceso de enseñanza - aprendizaje que servirá para motivar al estudiante en éste.

Los materiales serán los vehículos a través de los cuales se enviará al destinatario una serie de contenidos curriculares necesarios para desarrollar un curso en forma mediatizada. Nadie puede negar que en un sistema a distancia el rol de los materiales de ser portadores de contenidos es imprescindible. Se necesita un vehículo que desarrolle o presente los contenidos mostrando la particular visión que el programa tiene acerca de ellos. Es ésta una función necesaria pero no suficiente.

El material suple la ausencia de un profesor. Por eso debe tener condiciones que faciliten una "conversación didáctica" guiada. Deben orientar el aprendizaje, proporcionándole al alumno ayudas didácticas para acceder eficazmente a los contenidos y proponer actividades y espacios de participación para la necesaria contextualización y transferencia de los conocimientos.

Específicamente es función de los materiales didácticos:

Favorecer la autonomía, que es requisito indispensable en un sistema a distancia.

Despertar curiosidad científica en el destinatario, motivar para seguir estudiando y mantener la atención.

- Relacionar la experiencia, los conocimientos previos, con los nuevos que se proponen.
- Facilitar el logro de los objetivos propuestos en el curso. Presentar la información adecuada, esclareciendo los conceptos complejos o ayudando a esclarecer los puntos más controvertidos.
- Poner en marcha el proceso de pensamiento en el destinatario, proponiendo actividades inteligentes y evitando, en lo posible, aquellas que estimulen sólo la retención y la repetición.
- Propiciar la creatividad.

El proceso de Elaboración de los Materiales

La complejidad del proceso de elaboración de los materiales depende de la calidad pedagógica y académica deseada para el producto final y de la variedad, alcance y complejidad en la integración de los medios empleados.

La producción de materiales (intelectual y física) y de los recursos humanos requeridos pueden adoptar múltiples formas que dependen de los objetivos de calidad del producto final, de la disponibilidad financiera y de la posibilidad de contar con distintos especialistas.

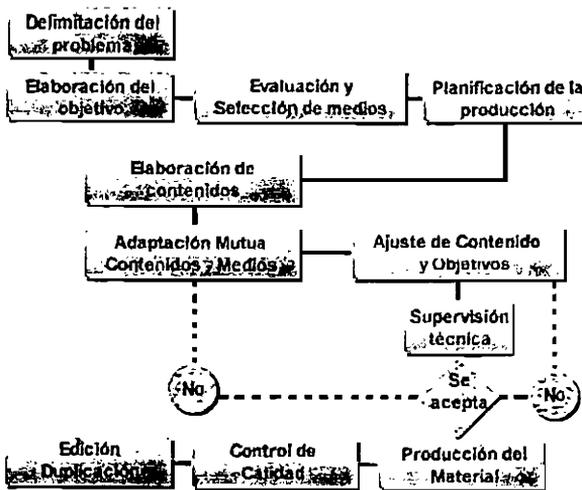
La conformación de equipos interdisciplinarios constituye, el ámbito ideal para la elaboración y producción de materiales, ya que a través del trabajo en equipo se enriquece el producto final.

En Educación a Distancia lo más importante es hacer un buen diseño general del sistema que apunte al logro de los objetivos planteados respetando el enfoque establecido. Para ello se integran distintos medios, teniendo en cuenta las posibilidades de cada uno en relación con los componentes del diseño didáctico.

Pasos aconsejados para la elaboración de materiales

Independientemente del material que se elija y, aún cuando sabemos que el proceso de elaboración de materiales para Educación a Distancia asume características diferenciadas según el marco teórico que se maneje, podemos sugerir una serie de pasos y recomendaciones que pueden ser; útiles por su generalidad y flexibilidad

En el siguiente esquema, se presentan las etapas y pasos que conforman la producción de materiales didácticos.



Delimitación del problema: La primera tarea del equipo debe ser definir claramente el problema que se abordará en el material. Esto implica una visión global de la situación, tanto desde el punto de vista de las necesidades institucionales, como de la información a desarrollar y desde las características de los destinatarios y su contexto.

Elaboración de Objetivos: En función del problema definido se procede a establecer con precisión qué se propone lograr con los materiales a producir. Debería quedar clara, aquí, la relación entre los objetivos y la solución del problema.

Evaluación y Selección de Medios: Íntimamente ligado con el proceso de fijación de objetivos está la selección de los medios más adecuados para lograrlos. Aquí el equipo optará por el o los medios más adecuados entre los seleccionados por el proyecto. Los criterios que primarán en esta selección tendrán que ver básicamente con la naturaleza de objetivos y contenidos. Los demás criterios (características de los destinatarios, cobertura y recursos disponibles) ya han sido tenidos en cuenta en la selección previa de medios).

Planificación de la Producción: Luego de seleccionar los distintos medios, el equipo procede a la planificación de su producción teniendo en cuenta los requerimientos de insumos temporales y materiales de cada medio, las etapas del modelo de producción, los tiempos y los responsables.

Elaboración de Contenidos: En función de lo planificado, el especialista en contenidos procede a su desarrollo. Para ello seguirá las orientaciones generales del proyecto y las pautas específicas que el equipo productor haya elaborado.

Básicamente deberá :

- ◆ Tener siempre presente el objetivo perseguido.
- ◆ Desarrollar la información con claridad, proporcionando ejemplos y explicaciones extra.
- ◆ Sugerir bibliografía.

Adaptación mutua de Contenidos y Medios: En esta etapa del proceso de producción se procede a adaptar los contenidos a la forma propia del lenguaje del medio o los medios seleccionados, teniendo en cuenta la función que cumplirá dentro del programa.

Ajuste de Contenidos - Objetivos: Aquí la coordinación del programa procederá a supervisar lo realizado hasta el momento a fin de determinar su coherencia interna. Se cotejarán los contenidos con los objetivos propuestos. Si hay coherencia, el proceso continúa en el paso siguiente, de lo contrario se volverá a la instancia anterior y se deberán proponer los ajustes necesarios.

Pre-producción del Material: En esta instancia se procede a darle al material su estructura definitiva, donde, de acuerdo a las características propias del medio utilizado, se presentarán los contenidos, actividades, problemas, etc. Este paso estará a cargo del diseñador didáctico, guionista de audio, video, etc.).

Supervisión Técnico - Académica: En esta etapa se produce una nueva revisión de lo realizado. El objetivo fundamental de este paso es comprobar que durante el diseño de los materiales no se hayan producido distorsiones en los contenidos, especialmente en la traducción de los mismos al lenguaje propio de cada medio. También se controla que la "conversación didáctica" sea adecuada para el destinatario. Si esto se cumple, el proceso continúa en el paso siguiente, de lo contrario se vuelve al anterior.

Producción del Material: En esta etapa se procede a la producción técnica del material de acuerdo a las características específicas y propias del medio seleccionado. Esto estará a cargo de los productores que, según el medio, serán editores, diseñadores gráficos, productores de radio y TV, etc.

Control de Calidad: Este es el último control del material antes de que lo reciba el destinatario. La coordinación técnica del programa hace una revisión crítica del material teniendo en cuenta parámetros de calidad ya definidos, estilos, lenguajes, diseños, etc. Pasa luego a juicio de expertos, quienes desde una mirada externa y especializada juzgan tanto la calidad académica como técnica del material.

Edición, duplicación: Una vez concluidos los controles de calidad y aceptado el producto final como material apto, se procede a su edición y duplicación.

Criterios para seleccionar los medios

Los distintos medios didácticos utilizados en la Educación a Distancia deben ser analizados a partir de sus condiciones concretas de función y, principalmente, por las funciones pedagógicas que puedan cumplir en relación con las necesidades de Educación de una determinada población.

Los medios visuales, auditivos, audiovisuales e informáticos, imponen ciertas características a los mensajes. Hay grandes diferencias entre transmitir una información en un material impreso y difundirla a través de la televisión a canal abierto. En cada caso, la relación de los destinatarios con los medios asume características distintas, lo que condiciona la propia estructura de los mensajes.

En los medios audiovisuales se utilizan ciertos recursos como la inmediatez, la redundancia y la repetición, que no son tan apropiados para los medios impresos. Esto es así porque mientras los medios audiovisuales se distribuyen en el tiempo haciendo más lógica la reiteración, los impresos lo hacen en el espacio: al permanecer fijos y completos en el tiempo, los mensajes escritos hacen que la redundancia y la repetición sean consideradas tediosas cuando no incorrectas.

También es diferente la forma en que las personas se relacionan con los distintos medios: la lectura de un texto exige el dominio de ciertas habilidades lectoras y el conocimiento de algunas normas de lectura predeterminadas; a su vez, permite interrupciones, retornos y relecturas de forma inmediata. *Por otro lado, interactuar con una computadora implica otro tipo de dominio donde se exige no sólo alfabetización en lecto escritura sino también conocimientos de informática; a su vez, una computadora permite un grado mayor de interacción que el texto impreso.*

En síntesis podemos diferenciar los distintos medios por sus características intrínsecas, por la relación que con ellos deben establecer los destinatarios y por las características de los mensajes.

Todo ello nos lleva a reflexionar acerca de la importancia de realizar una correcta elección de medios en el diseño global de la estrategia.

Para garantizar una racional elección de los mismos se sugiere tener en cuenta los siguientes criterios:

- ◆ Objetivos perseguidos
- ◆ Contenido a transmitir
- ◆ Características de los destinatarios
- ◆ Recursos disponibles (técnicos y económicos)

Analizando en detalle los aspectos enunciados podrá determinarse si el objetivo que se persigue, asociado al correspondiente contenido, requiere explicaciones, demostraciones, redundancia variada, interacciones, análisis de datos, presentación y producción de síntesis, etc. Es claro que algunos medios son más convenientes que otros para cumplir estas funciones. Si por ejemplo, nuestro objetivo es lograr que los destinatarios reflexionen acerca de ciertas normativas éticas propias de la función pública, el medio impreso se muestra como aconsejable. En cambio si nuestro propósito es que adquieran ciertas habilidades comunicativas para la atención al público, entonces los medios auditivos o audiovisuales resultarán más aptos para lograrlo.

Respecto de las características de los destinatarios, hay que tener en cuenta los hábitos, destrezas, conocimientos y recursos de la población destinataria en relación con las posibilidades y exigencias de los distintos medios.

Con relación a los recursos, es importante que los planificadores realicen un completo relevamiento de los medios que están disponibles en el contexto del proyecto y que podrían ser utilizados por el mismo.

A veces sucede que se eligen los medios tradicionales cuando las organizaciones poseen interesantes recursos no utilizados o estarían en condiciones de incorporar nuevas tecnologías en condiciones ventajosas. Ejemplos de ello lo constituyen la existencia del correo electrónico en muchas instituciones que lo subutilizan y la simplicidad y economía del uso de la audio conferencia, recursos que aún son poco incorporados a los proyectos de Educación a Distancia.

Un análisis prolijo de la disponibilidad de medios o de las posibilidades económicas para incorporarlos asegurará al proyecto un uso racional de los recursos existentes y le permitirá una elección de los mismos con mayores fundamentos y posibilidad de éxito.

Con respecto a la cobertura geográfica y poblacional que alcanzará el proyecto, hay que tener en cuenta que proyectos multitudinarios con una gran dispersión geográfica tendrán más necesidad de incorporar medios masivos y de gran alcance que aquellos proyectos acotados en el espacio y en la población.

La duración de un programa también es un criterio a analizar en el momento de seleccionar los medios para desarrollarlo. Cuanto mayor sea el tiempo asignado más se justificará la elección de medios de complejo desarrollo y de alto costo, ya que un uso reiterado de los mismos amortiza la inversión inicial tanto técnica como económica. Por lo contrario un programa de corta duración, tal vez no lo justifique.

Es importante dejar en claro que una buena selección de los medios a utilizar en un proyecto de Educación a Distancia deberá tener en cuenta todos los criterios enunciados. Por más recursos que se dispongan, no será pertinente la elección de ningún medio que no permita el eficaz logro de los objetivos planteados o que no se adapte a la idiosincrasia de los destinatarios.

Una vez analizados los criterios, en el siguiente cuadro se muestra una clasificación de los medios disponibles para su utilización en la Educación a Distancia.

Medios utilizados en la Educación a Distancia

Medio	Ejemplo:
Impresos	Texto guía, Manuales, Unidades didácticas, Fotografías, Láminas.
Auditivos	Programa de radio, Audio casete, Audio conferencia.
Audiovisuales.	Emisión de T.V., Video, Videoconferencia.
	Computadora más software herramienta.
Informáticos:	Computadora más software tutorial.
	Sistema multimedia.
Nuevas tecnologías de la informática:	Internet (Correo electrónico, Foros de discusión, WEB, etc.) Videoconferencia digital, TV interactiva, etc.



El facilitador

El concepto de facilitador responde a una concepción de educación individualizada, esto es: se atiende las características del alumno pero se actúa dentro de un sistema de educación colectiva.

Es diferente a la educación individual pues en ésta el alumno se apoya demasiado en el profesor, inhibiendo sus estímulos autodidácticos. De la educación socializada se diferencia en que la enseñanza colectiva implica la utilización de los mismos estímulos para todos los alumnos, sin descender al proceso de aprendizaje individual.

La característica fundamental del facilitador es la de cumplir la función de ser el nexo interactuante entre la organización general del sistema y los alumnos, capaz de captar las expectativas, necesidades, intereses y reacciones y de intervenir en el proceso de retroalimentación académica y pedagógica.

Si bien es cierto que los materiales establecen un nexo entre las partes, es el facilitador el que cumple la tarea de asegurar la efectividad de dicho nexo, poniéndose en contacto con los destinatarios durante el proceso cuando sea necesario. Es por eso que sus funciones generales son las de orientar y motivar este proceso, aunque las formas concretas que asuman estas funciones se redefinan en la interacción con los participantes.

Sus características principales son:

Atiende a la persona, ya que debe ser un sistema flexible que se acomode a cada uno de los alumnos de acuerdo con su personalidad, intereses, capacidades y conocimientos y se debe centrar en la personalidad del sujeto y despertarle conciencia de sus posibilidades y limitaciones.

Existe una cooperación entre el facilitador y el alumno, esto se da a través del trabajo de grupo, en el cual, con el aporte de todos, se logra que el pensamiento aflore, se verbalice y se discuta. El entusiasmo del profesor es esencial para salir adelante con esta metodología de enseñanza.

El Profesor – facilitador

El facilitador no es un profesor en el sentido tradicional, su trabajo esencial no es transmitir información. Debe ser un crítico constructivo, que ayuda al alumno a salir de ciertas dificultades y explorar nuevos campos.

El principal objetivo del facilitador es capacitar al alumno para que trabaje por sí mismo, piense por sí mismo y construya su propio cuerpo de conocimientos sobre la materia que estudia.

Funciones del facilitador.

Las funciones propias del facilitador son:

- ◆ Motivar y promover el interés de los participantes en el estudio de las temáticas propuestas.
- ◆ Guiar y/o reorientar al alumno en el proceso de aprendizaje atendiendo a sus dudas o dificultades, aportando ejemplos clarificatorios.
- ◆ Ampliar la información, sobre todo en aquellos temas más complejos.
- ◆ Evaluar el proceso de aprendizaje seguido por los participantes.
- ◆ Participar en el diseño de las evaluaciones de aprendizaje.
- ◆ Intervenir, junto a los otros facilitadores del curso, en las reuniones de coordinación general, aportando datos útiles sobre los alumnos, los materiales y el sistema en general.
- ◆ *El facilitador no es portador de contenidos, papel que en estos sistemas cumplen los materiales, sino un facilitador del aprendizaje.*

Lo principal es tener en cuenta cómo el trabajo del facilitador se vincula con los materiales informáticos. La fuente básica de la información la dan los textos y el trabajo del facilitador cumpliría la función de guiar esa información, ampliarla y resolver problemas encontrados en ella, orientar sobre la bibliografía y sobre los trabajos por realizar.

Este trabajo debe ser planificado; sólo así se facilitará la atención individual al alumno.

Nuevas Tecnologías de la Informática y la Comunicación Aplicadas a la Educación a Distancia

Las nuevas tecnologías permiten una mayor integración del alumno "distante" al proceso de aprendizaje. Integrando sonido, movimiento, imagen y texto, se crea un nuevo sistema de enseñanza que potencia al alumno y le permite estar más involucrado en el proceso de enseñanza. Por estas razones, las tecnologías aplicadas a la Educación a Distancia se están expandiendo rápidamente, y los docentes de programas de enseñanza a distancia, deben estar mejor preparados y organizados para incorporarlas.

Estar al tanto del equipamiento disponible y no deben dejarse llevar por la tecnología, es una condición necesaria para que la incorporación de nuevas tecnologías produzca mejoras en los programas de enseñanza.

Como el factor más importante para alcanzar el éxito en la Educación a Distancia es el grado con el cual los facilitadores son capaces de producir actividades estructuradas con una buena aplicación de la tecnología, a continuación se presentara una breve descripción de las tecnologías que hoy en día se encuentran disponibles en el país.

Videoconferencia Interactiva

La Videoconferencia Interactiva es una herramienta eficaz que puede usarse en el ámbito de la Educación a Distancia. Este sistema puede integrarse en los programas de Educación a Distancia con una adaptación mínima al plan de estudios de los cursos y puede diseñarse para favorecer la comunicación por medio del video y audio bidireccional entre múltiples localidades.

Características

La mayoría de los sistemas de Videoconferencia Interactiva utilizan el video digital comprimido para la transmisión de imágenes en movimiento por medio de las redes de transmisión de datos tales como alta capacidad Servicios Integrados a las Redes Digitales - (ISDN) Integrated Services Data Networks. El proceso de condensación vídeo imágenes reduce la cantidad de datos transmitidos a través de las líneas, transmitiendo sólo los cambios producidos en los cuadros de imágenes. Por haber minimizado el ancho de banda exigido para la transmisión de imágenes, la condensación de vídeo imágenes redujo también los costos de transmisión.

Las Videoconferencias Interactivas, a menudo se transmiten por medio de líneas del teléfono especializadas denominadas T-1. Estas líneas trabajan a altas velocidades y son muy eficaces para esta tecnología, pero se alquilan por medio de circuitos especiales y tienen un costo de mantenimiento mensual relativamente alto. Por otro lado, los costos de comunicación se calculan en función de la distancia y en el tiempo de comunicación. Los sistemas de videoconferencia interactiva pueden operar a distintas velocidades de transmisión de datos, es decir a varios fragmentos de capacidad de líneas T-1. Un sistema de Videoconferencia Interactiva también puede compartir una línea T-1 con la transmisión de otro tipo de datos digitales como ser transmisiones de Internet o transferencias de archivos.

La Videoconferencia Interactiva normalmente es usada para conectar dos sitios remotos empleando sofisticada tecnología de computadoras. El centro de la Videoconferencia Interactiva es el codec (codificador/decodificador). Éste es el dispositivo electrónico que transmite y recibe las señales de video que los miembros de la clase verán en sus monitores de televisión. Puede ser más fácil de pensar en el codec como un módem sumamente sofisticado. Un módem toma datos digitales y lo transmite a través de las

líneas de teléfono regulares. El codec toma las señales analógicas, las comprime y digitaliza transmitiendo las señales a través de las líneas del teléfono digitales.

Además de monitores de televisión, otro tipo de equipamiento es necesario para hacer una Videoconferencia Interactiva exitosa. Pueden incorporarse varias de las tecnologías instruccionales de uso más corriente como ser: los videos, micrófonos, cámaras, y computadoras.

Relativamente nueva la configuración "dial out", permite el uso de líneas telefónicas múltiples para conectar dos o más sitios en la misma conferencia. Acceder a líneas múltiples en forma simultánea puede ser muy difícil en localidades pequeñas. Además, el costo del uso del cableado telefónico puede ser prohibitivo. El costo de la llamada se multiplica por el número de líneas utilizadas en la conferencia.

Ventajas y Limitaciones de la Videoconferencia Interactiva

Ventajas	Limitaciones
<p>Permite "en tiempo real" establecer contacto visual entre los estudiantes y el instructor o entre estudiantes localizados en sitios remotos.</p> <p>Soporta el uso de diversos medios de comunicación: Las pizarras, documentos electrónicos, escritos a mano y videos pueden incorporarse a la transmisión.</p> <p>Permite la conexión con expertos de otras situaciones geográficas.</p> <p>Puede proporcionar acceso para los estudiantes de necesidades especiales.</p> <p>Provee un acceso adicional a los estudiantes de sitios remotos.</p>	<p>Los costos iniciales del equipo y el arrendamiento de las líneas necesarias para transmitir realizar las videoconferencias pueden ser prohibitivos.</p> <p>Las compañías que producen los equipos desarrollan sus propios métodos de condensación de imágenes lo que genera (a veces) un problema de incompatibilidad de equipos. Aunque se han establecido protocolos para permitir las comunicaciones entre las distintas marcas de equipos, este compromiso sólo se cumple en cierto grado.</p> <p>A menos que el instructor no realice un esfuerzo adicional, los estudiantes remotos pueden permanecer desinvolucrados con el curso.</p> <p>Si los audiovisuales, como videos, escritos y documentos electrónicos no se preparan apropiadamente, los estudiantes pueden perder un tiempo valioso al leerlos o tratar de entenderlos.</p> <p>Si la transmisión entre los sitios no es realizada por los medios más convenientes, los estudiantes pueden observar "las imágenes con fantasmas" cuando rápidos movimientos ocurren en tiempo real.</p>

Las videoconferencias interactivas pueden ser una herramienta instruccional muy eficaz para el educador a distancia. Como con otras tecnologías, su utilidad está directamente relacionada a que el instructor entienda sus beneficios, limitaciones y las distintas estrategias de utilización.

Televisión Instruccional

La Televisión instruccional (TVI), es un sistema efectivo para la distribución de contenidos a distancia que puede integrarse en el plan de estudios en tres niveles básicos:

Sola lección: Los programas emitidos tratan un solo tema específico o desarrollan un solo concepto, proporcionando una introducción a la lección, una apreciación global del tema, o un resumen.

Unidad selecta: Son una serie de programas que proveen el contenido fundamental de una o varias unidades de aprendizaje o lecciones del plan de estudios del curso.

Curso completo: Pueden integrarse una o más series de programas de una sola lección en un curso completo de duración variable. La TV se integra con otro tipo de materiales instruccionales como por ejemplo: materiales impresos.

La TV instruccional puede ser:

Pasiva, es la que involucra programas pre-producidos que son distribuidos, generalmente, por videocassetes o por tecnologías basadas en la transmisión de imágenes como el cable o satélite.

Interactiva, es la que mantiene las oportunidades de la interacción del espectador con un instructor en vivo o en un sitio lejano. Por ejemplo, la televisión bidireccional con audio bidireccional les permite a todos los estudiantes ver y actuar recíprocamente con el docente o instructor. Al mismo tiempo, las cámaras instaladas en los sitios remotos le permiten al maestro ver a todos los estudiantes participando. También es posible configurar el sistema para que todos los estudiante distribuidos en diferentes sitios se puedan ver entre sí.

Ventajas y Limitaciones de la Televisión Instruccional

Ventajas	Limitaciones
<p>Imágenes y movimiento pueden combinarse en un solo formato para que puedan ilustrarse conceptos complejos o abstractos a través de la simulación visual. La frase: "una imagen vale más que mil palabras" suena muy verdadera.</p>	<p>Es caro obtener una calidad de transmisión aceptable. La producción de video consume mucho tiempo y puede demandar muchos recursos técnicos, requiriendo a menudo medios y equipamiento relativamente sofisticados.</p>
<p>La Televisión Instruccional es una manera eficaz de llevar a los estudiantes a nuevos ambientes (la luna, un país extranjero, o a través de la lente de un microscopio).</p>	<p>Sin una producción profesional, los programas de TV instruccional completos parecen a menudo amateurs. Una vez terminados, los programas de TV instruccional pueden ser difíciles de revisar, corregir y actualizar.</p>
<p>Tiempo y espacio pueden derrumbarse, para que puedan captarse eventos de manera de estar presentes cuando ellos ocurran.</p>	<p>Fijar auditorios interactivos para un programa de TV instruccional puede requerir de equipos, medios, y de personal especializados.</p>
<p>Es muy eficaz para introducir, resumir y repasar conceptos.</p>	<p>La mayoría de los cursos pre-producidos de TVI que usan medios de comunicación masivos, apuntan a un tipo de estudiante "promedio" por consiguiente, ellos pueden ser ineficaces para alcanzar a los estudiantes con necesidades especiales.</p>
<p>Puede usarse eficazmente como una herramienta motivadora.</p>	<p>Cuando es usado en forma pasiva, sin la interacción, su efectividad instruccional puede limitarse.</p>

En el Diseño Instruccional para la TV Instruccional el desafío es pensar en términos visuales. Tomando ventaja de la imagería visual de la TVI se puede caer fácilmente en excesos de disertación. Las maneras cuidadosamente planeadas de mostrar en lugar de decir pueden mejorar la efectividad instruccional de la TV instruccional.

Porque los maestros y los estudiantes están físicamente separados por la distancia, el desafío del maestro es reducir psicológicamente la brecha producida, no sólo a través de un uso apropiado de la tecnología sino también a través del uso de prácticas de la enseñanza eficaces. Una buena enseñanza debe asegurar que se desarrolle una relación efectiva entre los estudiantes y el maestro.

Audio Instruccional

Las herramientas del audio instruccionales interactivas para el educador a distancia incluyen el teléfono, audio conferencia, y radio de onda corta. Las Audio conferencias pueden ser solamente de audio o pueden estar reforzadas por imágenes o transmisión de datos, lo que se denomina conferencia audio - gráfica. Las Audio conferencias típicamente utilizan el sistema de telefonía pública para enlazar dos o más sitios remotos.

Cuando en las audio conferencias participan grandes grupos de personas se usan dispositivos adicionales para reducir el ruido y la interferencia. Entre los componentes técnicos de una audio conferencia tipo pueden estar: Teléfonos sin manos ó de manos libres, parlantes, micrófonos; puentes de audio que interconectan múltiples líneas de teléfono y controladores de mandos de sonido, además de un conductor o locutor que facilite las interacciones.

La conferencia audio gráfica combina la tecnología para comunicación de voz con imágenes y/o transmisiones de datos. Mientras la voz sigue siendo el medio de comunicación principal, los periféricos audio gráficos proporcionan el componente visual. Los dispositivos periféricos audio gráficos incluyen la pizarra electrónica, los videos, y la computadora personal.

Ventajas y Limitaciones de la Audio conferencia

Ventajas	Limitaciones
La Audio conferencia es comparativamente más barata de instalar, operar y mantener.	Se puede encontrar una resistencia inicial hasta que los usuarios se familiaricen con el equipo y sepan cómo usarlo eficazmente.
Usa la tecnología telefónica disponible y, por consiguiente, puede llegar a muchos estudiantes.	Puede ser un tanto impersonal porque no transmite las señales no verbales como el lenguaje corporal: sonrisas, gestos, abrazos, movimientos de la mano, etc.
Generalmente es una tecnología muy familiar tanto para los maestros como para los estudiantes y es relativamente fácil usar.	Existen grandes restricciones en cuanto al tipo de contenido que puede entregarse en el formato oral.
Es un medio interactivo y permite la participación directa de los estudiantes y del instructor. Los estudiantes tienen muchas oportunidades para interactuar con sus pares, con el instructor y con expertos remotos invitados.	
Puede ser muy efectiva cuando es usada en combinación con otros medios incluyendo los materiales impresos, el video, y las computadoras.	

El desafío en el diseño de un curso basado en el Audio, es aumentar al máximo las capacidades del medio minimizando sus limitaciones. La ventaja mayor de la Audio conferencia es su capacidad interactiva bidireccional. Su limitación mayor es la ausencia de comunicación visual.

Generalmente, las técnicas útiles para impartir una buena instrucción son las mismas independientemente, si el maestro y el estudiante están en la misma aula o separados por muchas millas. Sin embargo, algunas técnicas de enseñanza son más eficaces que otras para las audio conferencias. Hay que tener en cuenta que este es un medio instruccional que carece de comunicación visual.

Internet y la Educación de Distancia

Internet es la red de computadoras más grande y poderosa del mundo. Abarca 1.3 millones de computadoras con direcciones de Internet que son usadas por más de 30 millones de personas en más de cincuenta países.

Con el acceso a Internet, los educadores a distancia y sus estudiantes pueden usar:

El correo electrónico (e-mail) - Como el correo postal, el e-mail se usa para intercambiar mensajes u otro tipo de información con personas. En lugar de ser entregado por el servicio postal a una dirección postal, el e-mail es entregado por software de Internet a través de una red de computación a una dirección electrónica. **Cartelera electrónica:** pueden acceder a través de Internet a muchas carteleras electrónicas de uso público. Dos de las más comunes en Internet son USENET y LISTSERV. USENET es una colección de miles de grupos de discusión organizados por temas, cubriendo el mundo enter. LISTSERV también proporciona foros de discusión de una variedad de temas ordenados áreas de interés especial.

World-Wide Web, (WWW) la Web es una excitante e innovadora herramienta de Internet. Oficialmente WWW se describe como una amplia área hipermedial de recuperación de información que apunta a dar acceso universal a un amplio universo de documentos". La Web proporciona a los usuarios de Internet la posibilidad de acceder a una amplia variedad de recursos (imágenes, textos, datos, sonido y video) disponibles en Internet. Software popular, como Netscape e Internet Explorer, facilitan la navegación y uso de la Web. La unidad básica que conforma la WEB es la "home page" ó "página de la casa." Donde cada organización e incluso cada usuario individual puede crear un sitio que contenga toda la información que les interese presentar. Las capacidades del hipertexto de la Web facilitan el enlace de la información dentro de su propio sitio e inclusive con otros sitios de la Web

Las Posibilidades instruccionales de Internet

Los educadores a distancia acostumbran a usar Internet y la Web para que los estudiantes obtengan un básico entendimiento de cómo navegar y obtener las ventajas de estar conectados a una red mundial de computadoras.

Las posibilidades instruccionales que brinda Internet pueden sintetizarse en:

Usar el E-mail como medio de correspondencia informal: La retroalimentación por parte del puede recibirse más rápidamente que los mensajes enviados por correo. Los estudiantes pueden leer los mensajes a su conveniencia y fácilmente pueden guardarlos para ser utilizados más tarde.

Establecer una cartelera electrónica del curso: Los estudiantes a distancia trabajan a menudo en aislamiento sin la ayuda y apoyo de sus compañeros. Preparando una cartelera electrónica se puede alentar la interacción entre estudiantes. Con una conferencia de clase por computadora, los estudiantes individuales pueden mandar por correo sus comentarios o preguntas a toda la clase, y cada uno de los otros alumnos puede responder libremente. La conferencia también puede usarse para mandar por correo todas las modificaciones del esquema de la clase o del plan de estudios, así como nuevas actividades, test con sus respuestas.

A medida que cada vez son más las universidades, escuelas, compañías, y personas particulares que se conectan a Internet, más posibilidades se abren para los educadores a distancia para superar tiempo y distancias para captar a más estudiantes. Lograr que los estudiantes se familiaricen con los recursos disponibles en Internet, es parte del desafío instruccional.



EDUCACION VIRTUAL.

Universidad Virtual

A continuación se presenta algunas de las definiciones de Universidad Virtual que se han recopilado durante el transcurso de esta investigación:

1.1 La Universidad Virtual provee educación a distancia, utilizando para ello los sistemas de telecomunicaciones y redes electrónicas con el apoyo complementario de herramientas didácticas desarrolladas en multihipermedios tales como discos compactos, Internet, producción de material audiovisual e impreso como medios tecnológicos que rompen las barreras y las limitaciones de la educación tradicional.

1.2 La Universidad Virtual lleva a cabo un sistema de enseñanza y aprendizaje que opera a través de las más avanzadas tecnologías de telecomunicaciones y redes electrónicas, por medio de las cuales se ofrecen cursos en los niveles de preparatoria y profesional así como programas de maestría y educación continua. Los componentes en una universidad que son indispensables para tener una experiencia universitaria (deben estar presentes todos ellos para que se pueda hablar de una universidad virtual) son:

- ◆ Aulas (donde se imparten cursos en vivo).
- ◆ Bibliotecas (donde se lleva a cabo consultas y autoaprendizaje)
- ◆ Laboratorios (donde se hacen ejercicios prácticos e investigación)
- ◆ Cubículos de profesores (donde se consulta con los profesores) Auditorios y Salones de Conferencias (para coloquios, congresos y otros eventos especiales).
- ◆ Actividades Culturales y Artísticas.
- ◆ Espectáculos.
- ◆ Áreas Sociales (para estudiantes y para profesores).
- ◆ Publicaciones especializadas y de difusión.
- ◆ Grupos de Interés y Asociaciones.
- ◆ Áreas Deportivas.
- ◆ Áreas Comerciales.
- ◆ Áreas Administrativas.

Vistos como procesos:

- ◆ Aprendizaje.
- ◆ Enseñanza.
- ◆ Investigación.
- ◆ Socialización.

1.3 La universidad virtual, no tiene sitio, ni está en un solo lugar, ni tiene planes de estudio fijos, ni profesores de tiempo completo y que es capaz, a pesar de esto, de formar profesionales capaces para el mercado de trabajo. Pero no hay que confundirse con la universidad distribuida, porque ésta es el mismo concepto rígido en el cual los estudiantes tomarán cursos aquí y allá, o quizá apoyados en videoconferencias, para cumplir con un programa académico tradicional. Lo único nuevo que introducen es que los alumnos podrán tomar cursos donde haya buenos profesores en determinadas materias y para las cuales su universidad sede no cuente con ellos. Esto es otra cosa pero en sí es una buena idea.

1.4 La universidad virtual deber tener todo los componentes de una universidad real exceptuando uno. No es necesario estar físicamente en el mismo lugar.

1.5 La Universidad Virtual es una comunidad compuesta por académicos y estudiantes, no por máquinas y satélites. La Universidad Virtual busca extender el servicio educativo a públicos nacionales e internacionales, enriquecer y ampliar el aprendizaje de los alumnos así como proporcionarles flexibilidad en tiempo y espacio, crear y difundir una nueva concepción de la enseñanza que utilice racionalmente la tecnología, fomentando el desarrollo de grupos multidisciplinarios y cooperativos en el diseño y gestión de programas educativos, promoviendo la investigación educativa y amplia la cobertura geográfica de la labor docente de los catedráticos mejor calificados con el fin de beneficiar a un mayor número de alumnos, actualizando, desarrollando y apoyando académicamente a un número cada vez mayor de personas mediante la recepción y transmisión de cursos y tele conferencias en vivo que permitan establecer un intercambio de opiniones entre los diversos públicos participantes.

1.6 La Universidad Virtual propone una institución sin recinto, un sistema de educación basado en aprendizaje vitalicio y en la interacción ciberespacial, en donde el estudiante asume la necesidad de aprender y establecer su propio ritmo de autoaprendizaje. Es un enfoque novedoso donde los títulos no generan jerarquías y que promueve la globalización del proceso de enseñanza. Su principal propósito es educar a personas más capaces, responsables, críticas y cooperativas, que propongan soluciones factibles a los problemas que aquejan a la humanidad.

1.7 La Universidad Virtual es un puente que permite acercar a los más reconocidos profesores del Sistema Universitario, a estudiantes de todo el mundo a través de las más adecuadas tecnologías de telecomunicaciones y redes electrónicas. Como un importante valor agregado de esta opción educativa se encuentra la posibilidad de recibir cursos en diferentes lugares y tiempos por medio de diversas e innovadoras tecnologías. El potencial de intercambios académicos y culturales a través de la Universidad Virtual abre relevantes oportunidades para influir decisivamente en el mejoramiento de los sistemas educativos del mundo entero.

1.8 La Universidad Virtual es una enorme red telemática conformada por bibliotecas, redes electrónicas y laboratorios virtuales que interconectan áreas geográficas distantes generando mayor calidad y motivando el aprendizaje a distancia.

1.9 La Universidad virtual es en realidad una institución global que sirve a los estudiantes desde todas partes y que podrá fomentar sobre el sistema externo del grado de cualquier universidad tradicional del mundo y a la vez se ofrecerán grados ampliamente de dispersión sin tener que asistir a las clases físicamente.

1.10 La universidad virtual se dedica al estudio interdisciplinario y a un análisis de barreras entre especialistas. Es así como podemos afirmar que esta constituida por equipos de investigación, cada equipo debe tener expertos en disciplinas diferentes, debe estar conformada por grupos de estudiantes, así la ordenación de miembros depende de su especialización externa. Ellos se evalúan con base en su contribución. Al estudiante se le da entonces un grado virtual, el grado virtual indica la participación en el proceso interactivo e interdisciplinario.

1.11 La universidad virtual es meramente una red de la investigación interdisciplinaria que sirve como medio para compartir sus resultados. Cada investigación tiene un representante sobre la asamblea de la Universidad Virtual, el equipo debe alcanzar un consenso sobre como lanzar su voto central, la función de la Universidad es informar, conectando a la gente quien esta interesada en la investigación del mismo tema.

La Universidad virtual administra los registros de los estudiantes que participan, es realmente como un conmutador conectado a los estudiantes en todas partes del mundo, para alentarlos a trabajar juntos en maneras interdisciplinarias.

1.12 La Universidad Virtual es un catalizador para el cambio dentro del mundo de los estudiantes conectados en todo el mundo, muchos de los que no son afiliados con ninguna universidad o el colegio. Es un medidor entre los estudiantes independientes y el académico de la comunidad.

1.13 La universidad Virtual es la que tiene unos especialistas en un área que no importa porque estos especialistas pueden tener un foro propio, un equipo de investigación tiene el mundo entero para buscar en el mismo tema desde diferente disciplina. La Universidad virtual puede verse como el corazón perdido del mundo académico.

1.14 La Universidad Virtual ofrece una sofisticada línea de ambiente, donde la experiencia sensitiva se asocia a la universidad verdadera, pocas de las entidades actualmente se describen como Universidades Virtuales esto hace que la oferta sea un conjunto de servicios que son más o menos adecuados como los sustitutos para la interacción de los procesos tradicionales de vida universitaria. Se corregiría más a la etapa actual de desarrollo, para rechazar la jornada inspirada y referida a las líneas de las universidades tradicionales.

1.15 Hacia una concepción de la Universidad Virtual, podemos determinar los siguientes tipos de universidad virtual:

- La universidad virtual formalizada
- La universidad virtual electrónica
- Características de la universidad virtual, (componentes de la universidad virtual):
- El aula electrónica,
- El aula inteligente, (aula virtual),
- La tele clase (Audio conferencia, la tele conferencia audio gráfica, la tele conferencia electrónica), El laboratorio virtual: de la ficción a la realidad.

1.16 La Universidad Virtual es la promotora de una globalización educativa a través del acceso electrónico e información y expertos en todo el mundo; propicia en los alumnos el desarrollo de un tipo de pensamiento constructivo y creativo y ofrece al profesor la oportunidad de trabajar en cátedras colectivas.

1.17 La Universidad Virtual es lograr un proceso de enseñanza - aprendizaje en el cual la tecnología toma un papel significativo como apoyo al profesor y al alumno, sin importar el tiempo y el espacio en el que se encuentre cada uno, caracterizándose principalmente por la aplicación combinada de las telecomunicaciones y de la multimedia en el proceso de enseñanza - aprendizaje. Con las telecomunicaciones se logra vencer las barreras del tiempo y del espacio. Con la multimedia se hace uso de la computadora como herramienta de integración de medios como audio, vídeo, sonido, imágenes, texto, animación, entre otros.

Algunas de las características de este modelo son descritas a continuación:

1. Utiliza los medios y recursos de las redes de comunicación electrónica, en el ambiente educativo.
2. Introduce el término de aprendizaje a distancia, concebido no solo como el medio de compartir escritos y lecturas, sino también, como el amplio rango de tecnologías de comunicación interactiva que nos libera del modelo tradicional de difusión y comunicación.
El aprendizaje a distancia utiliza términos como la tele-enseñanza, la cual puede ser definida, como el proceso de enseñanza- aprendizaje desarrollado mediante actividades basadas en redes de comunicación. Permite el uso del correo electrónico para la discusión de ideas, simulaciones en ambientes multiusuarios, video conferencias.
Mucha de la enseñanza continúa siendo administrada en forma sincrónica, en un momento dado, en un lugar determinado, aunque algunos componentes asíncronos son incorporados.
Implica la inmersión del proceso de enseñanza aprendizaje en las Redes de Comunicación, de tal manera que, haciendo uso de sus múltiples posibilidades, se obtiene una puerta abierta al mundo actualizado de la investigación.
3. Implementa el Aprendizaje Distribuido, modelo que permite que el docente, los estudiantes y los contenidos estén localizados en diferentes sitios, no centralizados, de tal forma que la enseñanza y el aprendizaje ocurren independientemente del tiempo y el lugar. Este modelo puede ser utilizado conjuntamente con los cursos basados en las clases tradicionales, con cursos tradicionales a distancia o **puede ser aplicado para crear salones de clases totalmente virtuales.**
4. Permite la creación de Libros de texto electrónicos en multimedia interactivos integrando imágenes, sonido, y otros archivos binarios. Estos textos pueden incluir hiperencadenamientos para permitir el aprendizaje exploratorio o pueden permanecer lineales para proveer más control sobre el contenido
5. Genera Materiales de lectura e instruccionales en línea caracterizados por hipertexto, imágenes, sonido, video, animación, y combinaciones de otros documentos.
6. Suministra Acceso a sistemas remotos. Los encadenamientos, a través de INTERNET pueden suministrar acceso a recursos de librerías y a otras bases de datos académicas, independientemente de la variabilidad de su ubicación geográfica.
7. Permite Comunicaciones asincrónicas entre los estudiantes y los docentes: Esta tecnología permite que los estudiantes elaboren y envíen sus asignaciones, proyectos y evaluaciones a través de medios electrónicos pero también permite las discusiones sincrónicas del grupo, en ambientes virtuales, donde usando el Web los integrantes del curso se pueden reunir y discutir formalmente e informalmente la asignación o lograr las metas del proyecto del grupo. Ofrece una gran cantidad de formas para lograr sesiones de tormentas de ideas, donde todos los participantes están actualmente, virtualmente presentes.
8. Logra el Aprendizaje experimental creando realidades virtuales donde estudiantes y facultad pueden participar en ejercicios simulados.
9. Se adapta al ritmo de aprendizaje del Usuario. Este modelo puede ser eficazmente utilizado en el aprendizaje individual, de tal forma que el alumno puede avanzar de acuerdo a sus propias necesidades
10. La información se encuentra almacenada en medios magnéticos, permitiendo que el individuo tenga libertad de movimiento dentro de la información, avance o retroceda a cierto tema y profundice en un tópico determinado.
11. El individuo toma el tiempo necesario para aprender, organizando el tiempo en la manera que más le convenga.
12. La información puede ser presentada en forma más clara y concisa, permitiendo además que el individuo involucre su propia creatividad, pudiendo dar su propio significado a la Información.
La ciencia de la información y la tecnología del cómputo han constituido en nuestra época un nuevo paradigma, cuyos efectos han llegado a todos los campos del conocimiento, cambiando las formas del pensamiento científico, ocupando un papel central en la actividad económica y en las formas de organización, causando un impacto político, social y cultural tan amplio e intenso como la mecánica clásica lo hizo en su tiempo. Con el proyecto de Universidad Virtual se espera poner en manos de la academia nuevas tecnologías de para aplicarlas en el proceso de enseñanza aprendizaje de tal forma que permitan el surgimiento de nuevas generaciones de hombres cualitativamente distintos a nosotros, en sus capacidades cognoscitivas, creativas y afectivas.

Educación Virtual

De tal forma la educación virtual esta íntimamente ligada a los conceptos de Universidad virtual; por lo que a continuación se dan las siguientes definiciones:

2.1 La educación virtual es una entrega de alta tecnología de la enseñanza y el profesor aprende una nueva experiencia que le permite estar en una ubicación diferente de los participantes. Es algo totalmente diferente al concepto tradicional de aula, y lleva un nuevo conjunto de expectativas para las dos partes involucradas tanto para el profesor "presentador" como para el participante "remoto".

2.2 La educación virtual marca una primera premisa para cualquier planificador de una Universidad Virtual, será la de tratar de llevar en forma paralela, educación donde antes no la había, sin superponerse a la acción de la Universidad Tradicional sino está en condiciones de brindar una educación mejor. Cuanto a más gente posibilitemos el servicio de educación, tendremos menores costos relativos (inversión por alumno) y mayor impacto en el mejoramiento de nuestra sociedad.

Por lo tanto este tipo de alumno necesitará una gran carga interactiva con el material con el que va a trabajar. A esta persona no les sirven las conferencias, los manuales, etc. , si no hay un espacio donde pueda poner en práctica un intercambio real o dialéctico que le permita incorporar el conocimiento.

Por lo tanto, bien podemos afirmar, que la educación virtual, no es para todos o por lo menos no lo es, si lo planteamos en el sentido tradicional de un emisor (profesor) que opta por un modelo (video, audio, texto) y un receptor que no está preparado para coordinar su aprendizaje, con una tipología distinta a la de su personalidad. En la educación tradicional, el material educativo que se enseña, se agota en sí mismo o en un trabajo práctico posterior.

En la educación virtual, esto no debe existir, ya que los estímulos educativos y sobre todo los encontrados en Internet, no provienen de una fuente sino que existe definitivamente una fuente inagotable que esta actualizándose día a día.

2.3 Enseñar (y aprender) en un ambiente " virtual" es muy diferente a hacerlo en el ambiente de una clase normal: en primer lugar, el proceso de enseñar se prolonga y distribuye en el tiempo en lugar de estar restringido a una fecha y duración precisas y rígidas. Esto nos libera de tener que cubrir apresuradamente tópicos para "cumplir" con plazos preestablecidos.

Lo que es más, la educación virtual no requiere una presentación o una "puesta en escena" de infraestructura como una clase presencial. En lugar de esto, destina ese tiempo a organizar el curso, definir asignaciones para los estudiantes, responder a sus preguntas y analizar su trabajo calificándolo así como resolver "problemas de tecnología".

El beneficio más evidente de la educación virtual reside en que brinda a estudiantes y profesores mucho más tiempo y flexibilidad en términos de plazos y desplazamientos.

Si bien hay plazos estrictos para entregar los trabajos, los estudiantes disponen de mucho más tiempo mientras tengan un teléfono cerca para poder llevar adelante su tarea (esto agrega días o semanas que antes se perdían en traslados) y permite que la educación no sea interrumpida tan fácilmente por viajes o traslados.

El solo hecho del incremento de la interacción personalizada entre docente y alumnos constituye por sí sólo el más grande de sus logros desde el punto de vista del diseño Instruccional; es difícil imaginarse cómo podría lograrse semejante interacción entre todos los estudiantes en un aula tradicional.

Además, de hecho, permite disminuir el número de los grupos formándolos por afinidades y además es posible incluir fácilmente expertos externos.

2.4 La informática es un complemento de los métodos tradicionales de enseñanza; ella permite crear un nuevo tipo de relación entre el utilizador, la información y el educador. Por esto podemos definir la educación virtual como la estimuladora en la utilización de la informática en la educación, aplicándola como un medio para el aprendizaje de individuos. Donde podemos mencionar:

Tele-enseñanza. Tele-Tutoría. Tele-Seminario Auto estudio Producción de materiales Sistemas de enseñanza.

2.6 La educación virtual es la integración de sistemas informáticos de apoyo a las tareas de enseñanza y en investigación de nuevas arquitecturas de sistemas informáticos cooperativos, donde intercambian opiniones y experiencias un grupo de personas dirigidas por un emisor.

2.7 La Educación virtual se consideraba que era una herramienta eficaz como apoyo a la enseñanza convencional y no como un sustituto de ésta. Pero hoy pienso después de haber leído un par de artículos de Internet que la Educación virtual nos permite desarrollar una nueva forma de enseñanza en la cual se debe utilizar de un modo racional y coherente las posibilidades que nos ofrecen las tecnologías actuales. Todo ello además, sin sacrificar los recursos didácticos inherentes a la formación presencial y aportando una mayor flexibilidad tanto en horarios como en el acceso a información complementaria, y una gran variedad de nuevas posibilidades lo cuál, a la larga, disminuye los costos formativos y hace los cursos más versátiles.

Así pues combinamos las tecnologías disponibles (multimedia, Internet, etc.) en una perfecta simbiosis creando un sistema metodológico que permite al alumno acceder a un material formativo dinámico, flexible y atractivo, haciendo especial hincapié en la versatilidad y facilidad de uso del sistema, así como en su robustez y viabilidad en la utilización cotidiana en el mundo real, teniendo en cuenta las limitaciones y trabas impuestas por agentes externos con que nos encontramos diariamente, como puede ser la manifiesta congestión y saturación de las redes telefónicas, etc., que hacen inviable a corto plazo y con costos razonables, opciones interesantes como puede ser el caso de la video-conferencia o la pizarra compartida por varios alumnos, etc.

2.10 La educación Virtual se puede decir que pretende desarrollar cursos y otras actividades educativas sin que todos los participantes tengan que estar simultáneamente en el mismo lugar. Esta apoyada intensivamente de las tecnologías de las telecomunicaciones, permitiendo la interactividad de los integrantes en tiempo real. Algo de lo que podemos concluir indiscutiblemente es que se mejoran las actividades complementarias a las clases presenciales, de forma que la comunicación entre profesores y alumnos o alumnos entre sí, fuera del acto de las clases es mucho mejor.

2.11 En la Educación Virtual no se requiere de una presentación con infraestructura como una clase presencial, en lugar de esto se destina el tiempo a organizar el curso, definir asignaciones, responder a sus preguntas y analizar su trabajo. De esta manera los estudiantes dedican mucho más tiempo para pensar más profundamente.

2.12 La Educación virtual impulsa el pensamiento reflexivo alentando a los estudiantes que tienen dificultades para expresarse o a los que no dominan el idioma, llegando a la conclusión que la Educación Virtual incrementa la capacidad de pensamiento crítico y las habilidades para resolver programas prácticos.

El papel del facilitador en la Universidad Virtual.

El papel del facilitador es, finalmente, ser el protagonista de cualquier actividad educativa, y que sin su esfuerzo e interés cualquier iniciativa en este campo está orientada al fracaso. Pese a ello en las instituciones carecen habitualmente de mecanismos de incentivación (económica o de otro tipo) para motivar y recompensar el esfuerzo que los facilitadores tienen que hacer para adaptar su forma tradicional de enseñar a las nuevas modalidades.

El facilitador debe enfocar la atención de los participantes en un monitor, más que tenerlos dentro de un aula. El monitor es más pequeño que el aula.

De allí que el papel del educador se transforme en un híbrido entre un profesor tradicional, un conductor televisivo o radial, un compañero de trabajo, un navegante (en el sentido de quien trabaja con las cartas náuticas que permiten a un barco dirigirse de un puerto a otro) y un preparador intelectual (a la manera de un preparador físico deportivo, pero tomando como eje el intelecto).

Otro grupo de consideraciones se refieren a los aspectos metodológicos que van apareciendo y hacen prever cambios que tendrán que ser estudiados y valorados cuidadosamente. Entre estos aspectos enumeramos:

1.- Con base en lo expuesto anteriormente el facilitador debe cambiar el concepto de lo que es la Universidad para los nuevos estudiantes que se vinculan a este medio y llevarlos de esta manera a tener claras las metas y los objetivos que se deben cumplir con esta nueva educación. Sin duda alguna el facilitador debe interactuar mucho más con los estudiantes fuera del horario de clase para resolver dudas o simplemente para responder su correo electrónico.

2.- Ahora bien, se debe tener un estudio muy bien diseñado con los objetivos que debe cumplir un facilitador en esta nueva enseñanza, esto debido a que se puede presentar dificultad de aceptación por parte de los profesores. Estos síntomas pueden ser iguales a los que los usuarios de las empresas sintieron cuando sus puestos pretendían ser automatizados por computadoras o que todo su ambiente de trabajo cambiaría de lo manejado manualmente a una pantalla de una computadora.

3.- La aplicación de la tecnología del satélite en la educación, involucra cambios en el proceso de enseñanza - aprendizaje y en el modelo educativo tradicional. En primer lugar, la señal se produce en aulas transmisoras donde se encuentra el profesor emisor. Su clase siempre debe ser planeada en coordinación con un productor de televisión con el fin de que ambos logren explotar al máximo las características del medio.

4.- Con la fusión de la informática con las comunicaciones (la teleinformática), se ha convertido la computadora en un fabuloso apoyo para el docente; gracias a sus diversas prestaciones como la producción de acetatos a color, la edición, la multimedia, o la comunicación simultánea de voz, datos y video es ya una verdadera revolución. Por ello algunos docentes están pasando directamente de la tiza y el tablero a modernos sistemas de tele conferencia, sin haber trasegado por ayudas educativas intermedias como son el retroproyector, las diapositivas o la micro enseñanza

5.- Aun así, los multimedios (ayudas educativas con el soporte de la computadora) están siendo subutilizados por la mayoría de actores que intervienen en la gestión, la docencia y la capacitación, al permitir que pasen de largo los beneficios de la teleinformática educativa para un amplio sector de la población potencialmente activa. Todos estos cambios indudablemente comprometen al profesor a aprender sobre ambientes virtuales de tele clase, tele trabajo y tele asesoría. Divulgar paquetes de software para transmisión de voz e imagen en Internet. Conocer metodologías para el diseño de paquetes auto educativos para implementarlos en ambientes Intranet y Extranet.

6.- Las consecuencias pedagógicas implícitas en todo esto nos dan idea del largo camino que aun queda por recorrer, sobre todo porque la gran mayoría de aportaciones al campo de la informática educativa se quedan todavía en el ámbito de computación, del software denominado " educativo " o utilización indiscriminada de dispositivos y herramientas sin una base educativa suficiente. Todo ese largo camino a recorrer implica un tiempo de adaptación, pero realmente ¿ cuánto es el tiempo del que se dispone para el profesor? ¿Durante qué lapso el sistema educativo puede permanecer en ese estancamiento de utilizar solo software educativo y utilitarios, como un medio que da paso a paso, sin afanes?. Dado que no es posible que todos los alumnos trabajen al mismo tiempo en la misma actividad, se preparan entonces actividades simultáneas pero diferentes en donde el maestro pueda trabajar en cada sesión con un grupo diferente de alumnos.

El facilitador, hombre libre, ciudadano libre, es consciente de que enseña a otros hombres y a otros ciudadanos igualmente libres. Los principios de su ciencia pueden ser discutidos. El profesor transmite conocimiento e información. El conocimiento es esa información sobre la que se ha reflexionado e implica un aporte a la ciencia que cultiva o explica. La diversidad de opiniones, para ser diversidad de pensamientos, debe fundarse en el conocimiento. La comunicación en la diversidad de pensamientos beneficia. La comunicación es conversación y la conversación es comprensión, una ventaja educativa que las máquinas difícilmente pueden aportar.

La universidad debe formar a los hombres según un ideal. Para ello requiere un sistema coherente que rija sus finalidades y sus posibilidades. Sólo así ese ideal se cumple y da sus frutos: el hombre que una sociedad desea. La enseñanza virtual derriba las barreras del aula tradicional.

El papel del estudiante en la universidad virtual

El estudiante local puede recibir clases con profesores de otros países. Al finalizar el curso, el alumno además de aprobar su materia puede recibir créditos de la universidad que impartió el curso. Los cursos que ofrecen otras universidades extranjeras amplían las opciones posibles de alta calidad curricular para los alumnos a distancia. El alumno ya no tiene que viajar para estudiar, asistir a un curso, seminario o escuchar una conferencia especializada. La Universidad Virtual puede conectar la universidad que distribuye el programa y recibirlo. El alumno tiene garantizado que recibe clases de un maestro emisor que se le considera, por sus académicos y evaluaciones, como el mejor calificado.

El alumno ya no tiene que tomar notas, porque recibe copias impresas de todos los apoyos de texto y gráficas que vaya el profesor a presentar en esa sesión. El alumno recibe desde el primer día de la clase la programación total, por sesiones, de lecturas, tareas, trabajos y fechas de exámenes. También un compendio con el todas las sesiones, lecturas de artículos, notas, casos, etc., que el maestro vaya a referirse durante el semestre. Esto varía dependiendo de la clase.

El alumno virtual forma parte de la denominada vanguardia tecnológica en educación a distancia. El ambiente educativo virtual permitirá frenar un poco la migración del campo a la ciudad en busca de centros educativos, pues serán los centros educativos quienes a través de los medios de comunicación buscarán al campesino, quien a su vez será un agente multiplicador de la cultura ecológica al acceder a un mejor nivel de instrucción y formación integral.

Así mismo, la implementación del Aula Virtual permitirá frenar un poco la tala indiscriminada de bosques, debido al reemplazo del papel por el medio magnético. Favorecerá la descontaminación ambiental al aminorar los desplazamientos al interior del casco urbano en busca de los distintos centros educativos y se aliviará el caos del tráfico capitalino.

En términos culturales, el efecto ambiental del Aula Virtual implica cambios tales como: la globalización del saber en todas las áreas del conocimiento, la modificación del comportamiento en las relaciones alumno - maestro, la familiarización del estudiante con las diversas expresiones culturales navegando a través de las redes, la emergencia de un nuevo espacio cultural para relacionarse con los demás, el renacimiento de nuevas actitudes en la visión cosmopolita de un mundo amplio y cambiante.

La clase virtual es mucho más rica que la tradicional por varias razones:

1. Los estudiantes dedican mucho más tiempo para pensar a profundidad los temas antes de discutirlos y los profesores en forma constante a dar estos requerimientos con bastante anticipación, así como responder a las preguntas de otros compañeros en forma diaria.

2. En una clase presencial, si el profesor hace una pregunta, puede lograr que tres o cuatro estudiantes respondan con sus ideas espontáneamente; en el aula virtual, los estudiantes tienen más tiempo para meditar sus respuestas y organizarlas en periodos más largos (a veces horas o días) y cada uno da una respuesta. Esto no solamente impulsa el pensamiento reflexivo sino que alienta a los estudiantes que tienen dificultades para expresarse fluidamente en la clase (por ejemplo, los que no dominan el idioma).

3. Sin embargo, ninguna evaluación o investigación de cursos que se conozca muestra esto como un resultado de los ambientes "virtuales" en los que nos estamos moviendo de hecho, probablemente porque éstas clases de capacidades (Ej.: pensamiento crítico, solución de problemas) no son medidas en forma directa por los medios típicos a través de los que se toman exámenes y se otorgan títulos. Creo que éste es un aspecto del aprendizaje virtual que merece mucha más investigación.

No hay duda acerca de que ser un profesor o estudiante virtual aumenta ciertas habilidades y capacidades. La interacción virtual requiere buenas capacidades de comunicación (especialmente escrita).

También hacen falta habilidades en el uso de computadoras si viene no en un nivel muy "técnico", la habilidad para aprender a usar nuevo software y resolver sus problemas es indudablemente la más importante habilidad tecnológica requerida.

Un aspecto muy importante para la enseñanza / aprendizaje virtuales son las consecuencias que traerán para la naturaleza misma de las organizaciones educacionales: en la medida en que los profesores virtuales (y sus alumnos) no necesitan aulas u otras instalaciones edilicias (auditorios, cafeterías, gimnasios, dormitorios, estacionamientos, etc.) ¿cual será exactamente el rol de la institución educativa? Obviamente la provisión de PC's se vuelve crítica aunque pueden obtenerse privadamente.

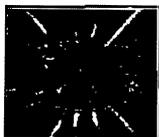
La administración incluye el procesamiento de admisiones, registrar los alumnos, cobrar las cuotas, manejar las becas y otorgar los títulos. Si bien todas estas funciones pueden en última instancia ser realizadas en forma electrónica, requieren personal y gerenciamiento.

Debería resultar claro que la educación virtual traerá cambios significativos al sistema educacional. Algunos serán positivos, otros pueden ser negativos, otros sólo serán diferentes a lo conocido.

El proceso de enseñanza aprendizaje tradicional utilizado desde hace tiempo en las aulas de clase, en el cual el profesor imparte su clase y los alumnos reciben los conocimientos, requiere de una modernización que facilite al profesor la transmisión de sus conocimientos y la asimilación de los mismos por parte del alumno. Esta necesidad ha dado origen a una constante búsqueda de herramientas educativas que permitan hacer uso de tecnologías de apoyo a la enseñanza con el fin de: vencer barreras que impiden la realización del proceso, sustentar los conocimientos que el profesor trata de comunicar y permitir que el proceso de transmisión recepción se realice de manera efectiva.

La prospectiva basada en la planeación mediante la metodología de escenarios futuros nos anuncia, que "para los próximos diez años el 40% de las universidades existentes hoy habrán desaparecido, desplazadas por las instituciones educativas que utilicen las nuevas tecnologías de educación virtual"

Los nuevos escenarios educativos y empresariales le están permitiendo a la universidad y a las organizaciones inteligentes un nuevo servicio al alumno trabajador, o al alumno de la provincia, y a sus clientes externos mediante prácticos elementos de tele virtualidad. Para ofrecer este servicio es necesario aprender a configurarlos con criterios de Tecnología Apropiada, usarlos y replicarlos para beneficio de la academia, y la capacitación de las empresas que quieran ser competitivas en el mercado. El cambio que ha tenido la tecnología educativa en los últimos cinco años no se ha compaginado con el desarrollo de los métodos didácticos en el aula de clase. Para plantearle a sus alumnos un nuevo paradigma en la educación, el profesor aún lo explica usando tiza y tablero cuando de hecho existen nuevas técnicas y metodología educativa para innovarlas hacia el aula.



Desarrollo y Utilización de Sistemas de Enseñanza Asistida por Computadora (EAC)

INTRODUCCION

Hablar de computación, es hablar de un tema apasionante en todos los sentidos, nos hace soñar sobre el futuro, nos hace discutir sobre las tecnologías apropiadas y sus costos, las políticas para desarrollar una industria, institución y un país. Pero fundamentalmente hablar de computación o Informática es hablar de la necesidad de recursos humanos capacitados, de los cambios en la forma de trabajar y los nuevos empleos, de las nuevas posibilidades de desarrollo individual y hasta de aprendizaje por computadora; hablar de computación es hablar de educación. No en balde se habla ya de la nueva sociedad de la información, de la Informatización de la sociedad. ¿Se está preparado para este cambio?, ¿Se está preparando a los estudiantes para este cambio?, ¿Se está preparando a las personas para decidir y conducir este cambio?.

En los comienzos de la era informática, los maestros vieron a la computadora como un instrumento temible. No sabían qué hacer con ella, no la comprendían y no podían, por lo tanto, transmitir nada a sus alumnos. Hubo un momento de optimismo. Fue cuando apareció el lenguaje Logo, un sistema intuitivo y sencillo, creado por Seymour Papert en el Instituto Tecnológico de Massachusetts (MIT). Su difusión contribuyó para que la escuela pudiese entender a la computadora como una herramienta con dos caras. Por un lado era un objeto de estudio. Pero, por el otro, el principal, era un instrumento para enseñar. Sin embargo, una vez más, el sistema educativo tomó por el rumbo equivocado: "Se abrieron los laboratorios de informática, que sacaron las computadoras de las aulas...". En esos laboratorios de informática se redujo a las computadoras a ser sólo objeto de estudio. Es claro que para aprender algo de las computadoras en educación, sobre todo en el caso de adultos, no se tiene que volver a explicar desde el principio cómo son las computadoras, para eso hay muchos textos de iniciación a las computadoras. Y así se perdió, casi por completo, su función más poderosa en la escuela: la de ser un maravilloso instrumento para aprender.

EL APRENDIZAJE EN LA INFORMÁTICA.

De acuerdo con los psicólogos el aprendizaje es la clave del comportamiento "inteligente". Una definición del aprendizaje sería la facultad de adaptarse respecto al mundo exterior, modificando las estructuras internas que rigen el comportamiento para aceptar o hacer suyas estas nuevas relaciones con el mundo externo. En términos de conocimiento, es el modificar las estructuras mentales, para relacionar e incrementar la información nueva y sobretodo aumentar con ellas la capacidad de inferir, deducir y generar nuevo conocimiento. En consecuencia, el aprendizaje es el internalizar no solo información, sino mecanismos para seguir aprendiendo. Uno de los problemas actuales de la Computación (en su rama de Inteligencia Artificial) es la formulación algorítmica del comportamiento humano y en particular del aprendizaje.

La dificultad de los problemas que conlleva el aprendizaje son en general bastante más difíciles que aún los problemas teóricos clásicos de la Informática (por ejemplo determinar lo "computable", o saber cuantos pasos o transformaciones mínimamente se necesitan para realizar una operación). Una parte de esta dificultad proviene del hecho de no poder acotar hasta dónde el aprendizaje llega o cuáles son sus métodos.

Existen tres tipos de aprendizaje:

- El Aprendizaje de memoria
- El Aprendizaje de un conjunto de información organizándola, simplificándola y asociándola (una materia como Geografía o Historia o la Aritmética)
- El Aprendizaje de un comportamiento en una situación dada, por ejemplo jugar al tenis, viajar, aprender un oficio, el aprender a resolver problemas diferentes con Matemáticas etc.

Estos tres tipos de aprendizaje tienen niveles de complejidad diferentes, lo cual es diferente a la dificultad. Así para ciertas personas les es más difícil o más fácil el aprender a hacer algo que el memorizar algo. La complejidad estriba en el significado de lo que implica o representa ese aprendizaje. Así, el memorizar significa simplemente guardar y poderse acordar de algo directamente. El siguiente nivel implica el aprender cosas en conjunto, con su significado así como el poder trabajar con ellas para relacionarlas. El último nivel significa el aprender a resolver situaciones nuevas, teniendo un patrón, o estructura de respuesta fundamental. El punto común de estas operaciones es el de aumentar los conocimientos con la finalidad de poderlos utilizar posteriormente.

La idea de mejorar es el punto esencial en la definición de aprendizaje, ya que no sólo significa aumentar o incorporar nuevas cosas sino también el ser capaz de comportarse mejor ante una situación análoga.

Si se trasladan las ideas anteriores al campo de la Computación se podría tratar de definir un sistema informático que mejore su desempeño y ámbito de operaciones como consecuencia de ensayos especiales llamados de aprendizaje, que permiten modificar el objeto en cuestión. Actualmente esto es posible en ámbitos o contextos muy reducidos. Es claro que el objeto es un programa que de alguna manera se perfecciona como consecuencia de ensayos y que esta perfección aunque inducida exteriormente no es realizada explícitamente. De alguna manera en la Computación el enfoque de aprendizaje trata de hacer intervenir al hombre lo menos posible, comprendiendo y haciendo explícito las condiciones en las cuales el objeto aumenta el conocimiento.

El conocimiento es la tercera etapa de la cadena datos- información - conocimiento que actualmente se maneja en computación. Así las primeras máquinas trabajaban únicamente con datos, es decir representaciones de hechos aislados. La mayoría de las máquinas trabajan actualmente con información, es decir datos estructurados y relacionados que tienen ya un sentido. Actualmente se plantea y se hace los primeros prototipos de las "máquinas inteligentes" cuyo objeto de trabajo es el conocimiento, es decir la incorporación como un objeto de manipulación no sólo a la información encadenada sino a los mecanismos intrínsecos de estos para inducir o deducir. Dentro de este contexto el aprender sería el comprender la semántica de las transformaciones que permitan deducir y modificar la representación y el establecimiento de una estructura de datos utilizable para la solución de problemas del mismo tipo, es decir que utilice la misma información. Es claro que no es posible entonces hablar de aprendizaje sin hablar igualmente de percepción, comprensión y de pensamiento, pero esto ya es tema de la Inteligencia Artificial, de lo cual no se hablara en este trabajo.

Es en este último sentido que no sólo las computadoras "inteligentes" de la quinta generación y el futuro que ayudan a enseñar y a aprender se interesan en el aprendizaje, sino que es una preocupación fundamental de la Informática y Computación teórica el tema del aprendizaje. Este tema del aprendizaje es curiosamente también el tema central de la pedagogía y de la educación. Aquí se unen Educación y Computación ya que ambos tienen en sus objetos de estudio al aprendizaje. Es en la medida que se entienda mejor este proceso del aprendizaje y se pueda representar y desarrollar en computadora como mejor se podrán hacer aplicaciones útiles a los seres humanos para su aprendizaje. Lo importante aquí ya no es, discutir si el conocimiento y el aprendizaje es exclusivo del ser humano, (o de algunos animales como actualmente se cree) sino como representarlo y capturarlo, describir los "mecanismos", el significado y la forma de adquisición del conocimiento, lo que dará la clave del aprendizaje.

PLANEACION, ESTRATEGIAS Y FORMAS EDUCATIVAS

Obviamente el camino para preparar recursos humanos en el uso de nuevas tecnologías es llevándolas a las instituciones educativas, esto implica que los planeadores de la educación diseñen medios para la familiarización de los docentes y alumnos con el uso de tales tecnologías.

La conveniencia de incluir los medios informáticos en las instituciones educativas esta determinada por el gran significado que en la planeación educativa se pretende tener. Ya que afecta directamente a los tres ejes más importantes de la planeación educativa (otros que los costos y el problema de la eficiencia): la extensión de la educación para todos, la posibilidad que la inclusión de tales medios garantice el acceso equitativo de la población escolar a la educación y finalmente que eventualmente mejore la calidad de la educación.

Se debe considerar la potencialidad de uso de la informática en el proceso enseñanza aprendizaje, independientemente de la forma de resolver o visualizar la situación anterior. Pero eso sería ignorar que los medios llevan el mensaje o lo que es lo mismo que las modalidades pueden tener implicaciones en las estrategias.

Aunque las estrategias pedagógicas para implantar un programa educativo caen dentro de la planeación educativa, ya que las modalidades escogidas pueden tener repercusiones en las estrategias o viceversa es conveniente al menos presentarlas aquí. La selección de una estrategia pedagógica proviene de la definición de una política educativa, de los objetivos mismos que se pretende con la educación y el manejo de sus medios.

Toda forma educativa pretende aproximar la conducta de los individuos de un grupo a los valores y normas conformados en un modelo del grupo social de referencia. El análisis de lo educativo ha llevado a identificar empíricamente dos grandes grupos: las espontáneas y las formales. Las primeras forman parte inherente de las instituciones humanas y no han sido intencionalmente desarrolladas; las segundas se han producido deliberadamente con ciertos propósitos, se reproducen mediante procesos racionalizados y requieren generalmente de un soporte administrativo para funcionar. Generalmente las autoridades

educativas se han orientado a planear y desarrollar estas segundas, que caen bajo su control o vigilancia; sin embargo se podría fomentar con mucho éxito las formas espontáneas.

Como estrategias pedagógicas de las modalidades formales podemos señalar a la enseñanza escolar, la educación abierta, la enseñanza a distancia, la capacitación, el entrenamiento en servicio, etc.

La aplicación de la tecnología educativa es el proceso de sistematizar los tres tipos de componentes, considerándola como el desarrollo, operación y evaluación de sistemas educativos directos o indirectos, que pueden ser replicables y que cuando se replican producen confiablemente los resultados educativos propuestos a la luz de las teorías del aprendizaje y la comunicación valiéndose de recursos humanos y técnicos. Aunque el término de tecnología educativa ha sido frecuentemente relacionado con todo tipo de máquinas que coadyuvan al proceso enseñanza - aprendizaje (audiovisuales, tipográficas, microcomputadoras, etc.) no todo tipo de equipo beneficia por sí sólo al aprendizaje, ni obviamente desplazará al docente.

No basta con poseer una gran cantidad / calidad de recursos materiales si no existe una estrategia de organización pedagógica.

Las estrategias pueden agruparse en tres sistemas educativos.

- Tradicionales : caracterizados por una correspondencia en los componentes tecnológicos con los empleados hace una generación.
- Tradicionales reformados: en los cuales se han introducido mejoras o adiciones a los recursos tecnológicos; y
- Innovadores : desarrollados por la tecnología educativa y caracterizados por adoptar una organización diferente en el aula con respecto al estudiante o con la organización escolar

Sistemas tradicionales.

Estos sistemas funcionan sobre una estructura estrictamente formal incluida en una organización altamente burocratizada, exige la presencia de los participantes en el proceso Enseñanza- Aprendizaje, tiene las características de ser masiva, pasiva y reforzante del papel dependiente del alumno

Es masiva en tanto supone una homogeneización en los estilos de aprender, ignora la individualización y frecuentemente castiga la desviación con respecto a la norma. Es pasiva por considerar que el estudiante es el ignorante que hay que redimir. Es reforzante de un papel dependiente en el sentido que su mensaje educativo no tolera la independencia y autonomía.

Sistemas reformados.

La mayor parte de estos sistemas ha surgido como forma de cubrir nichos no cubiertos por el sistema tradicional. Estas modificaciones no han sido necesariamente efectivas o deseables, su proliferación parece ser el acicate de una política improvisada, de una moda tecnológica y la disposición a una mayor experimentación apoyados en el avance y difusión de concepciones más industriales o sistémicas que de alternativas pedagógicas.

Analizando en conjunto el cambio tecnológico de un sistema educativo, el introducir modificaciones a diferentes niveles puede lograr estrategias de cambio total (sistemas innovadores) aunque en términos generales, las reformas a sistemas tradicionales han estado vinculadas a sus necesidades y peculiaridades, condicionando el tipo de experiencia que se ha diseñado e implantado en cada uno de ellos.

Sistemas innovadores.

Se consideran innovaciones y no modificaciones, cuando hay cambios estructurales en el proceso educativo. Un principio fundamental de estos sistemas es la necesidad de atender a los objetivos precisos de la enseñanza, y no solo los aspectos secundarios. La otra característica es la de considerar los procesos con enfoques diferentes, con visiones diferentes y no solo con medios diferentes. El uso de un equipo no es de por sí innovador en la educación aunque el aparato sea una novedad, si se está haciendo lo mismo. Así por ejemplo, al usar algún sistema innovador se ha descubierto que la velocidad de aprendizaje depende de la capacidad de cada quien; Incluso los estudiantes más lentos pueden dominar hechos complejos, si se da tiempo suficiente para avanzar en cada paso necesario al aprendizaje. Lo anterior puede ser una innovación respecto al proceso diferenciador de la educación tradicional.

Al examinar gran parte de los nuevos sistemas es obvio que la innovación tiene un importante papel en los intentos de mejorar la educación en particular, los aspectos relacionados con la enseñanza individualizada, revisión de planes de estudio, métodos de enseñanza, equipo escolar, diseño de instalaciones, uso de recursos educativos y la dirección del maestro, de aquí que la innovación no siempre sea por equipo novedoso sino por organización. De aquí que las formas, métodos, procesos educativos sean un campo abierto para la innovación. En particular deben ser un enorme potencial para los países en desarrollo para mejorar sus programas educativos sin tener por eso que gastar desproporcionadamente en educación al pretender ofrecer lo mismo que los países "avanzados". La computación se le ha descrito como un sistema innovador en educación principalmente por sus características de interactividad y de individualización

Toda estrategia pedagógica es la sistematización, estructuración e interacción en diferentes proporciones dentro de una estructura de tres tipos de componentes. Así toda estrategia pedagógica combina tres tipos de componentes:

- Técnico - sociales,
- psicometodológicos y
- Medios.

Todos ellos se organizan bajo una política educativa para producir los resultados educativos esperados. Cada componente puede adquirir determinadas características que le darán especificidad ante un problema educacional.

Componente de técnicas sociales

Los atributos más importantes definibles en las técnicas sociales de una estrategia pedagógica son: temporalidad, cobertura, equidad, formalidad y apertura

La temporalidad se refiere a la cobertura en el tiempo tanto en lo referente a la simultaneidad o a la posibilidad de hacer disponible la educación a cualquier momento para el estudiante. La cobertura puede ser localizada, cubriendo a un grupo de población en áreas circunscritas aunque tiene el potencial de poseer el atributo de penetración pudiendo llegar a las poblaciones independientemente de su dispersión o de su inaccesibilidad geográfica. La masificación y la equidad, implican el permitir el mismo número de oportunidades educativas con independencia de la ubicación geográfica, social, económica o política de la población. La formalidad implica todo el aspecto de logística necesario en el reconocimiento y obligatoriedad de ciertos estudios, esto tiene sus implicaciones ya que en tales casos se tiene que cubrir el programa aprobado por las autoridades educativas de determinada manera. Finalmente la apertura puede implicar a semejanza de estrategias tradicionales una interacción directa y presencia simultánea (minimamente educando- máquina) y a diferencia de la educación tradicional no excluye la posibilidad de trabajo productivo.

Componente de los medios

El medio por excelencia es la computadora. Este tiene a su vez como atributos a los siguientes: el costo, el alcance (limitado- amplio), sentido (único- de emisor a receptor, o bidireccional emisor- receptor-emisor) y suficiencia o capacidad de un medio para cumplir su cometido.

El costo no es sólo en materiales, equipo y programación, sino también en la capacitación del personal y en la creación de nuevas plazas para personal especializado, infraestructura, mobiliario etc.

Componente psicometodológico

Los atributos más relevantes respecto a este componente son: papel del alumno, interacción agente - alumno y el tipo de proceso.

El primer atributo es sobre el papel del alumno entre pasivo o activo. Se considera generalmente que una participación eminentemente activa del alumno es la que da mejor resultados en el aprendizaje, pero el tipo de sujeto o educando podría ser poco disciplinado, de aquí que los sistemas educativos no siempre quieran fomentar el activismo. El segundo establece una interacción maestro- alumno, siendo esta directa o indirecta. En la enseñanza por computadora es predominantemente indirecta por encontrarse "mediada" por la computadora. Finalmente el tipo de proceso o secuenciación didáctica, lo cual tiene sus repercusiones en la forma de aprender, la medición de dicho aprendizaje y las prioridades de lo enseñable (lo cual llega a ser asunto de política).

Las modalidades de enseñanza asistida por computadora en términos generales difieren en el aspecto psicometodológico lo que significa que los otros dos componentes pueden ser aplicables a cada modalidad

LOS ROLES DE LA COMPUTADORA EN LA EDUCACION

El enfoque general de las Computadoras en la Educación se le ha llamado en Inglés como "Computer Based Education" (CBE), cuya traducción sería Educación Basada en Computadoras. Esta abarca tanto a la enseñanza de las computadoras como al uso de las computadoras en la Educación. En el primer caso el objeto de estudio son las computadoras y en el segundo es la educación. Dado que en este trabajo se va a referir principalmente a lo segundo es decir a los empleos de la Computadora en la Educación, fue que se presentó como se conceptualiza actualmente el problema central de la Educación el aprendizaje y su conexión profunda con la Computación o Informática y como esta conexión va más allá del uso instrumental de una por otra, aunque en buena medida lo que se va a tratar aquí son esos usos instrumentales, ya que son los más desarrollados. No se tratará aquí, en consecuencia, de la Educación sobre las Computadoras, que sería la enseñanza de la Programación, de la Arquitectura de Computadoras, sus componentes etc.

Es claro que existen muchas clasificaciones de como puede ser entendido este campo de las computadoras en la Educación. Esto es el resultado de una teorización de una práctica y no de una teoría practicada como generalmente ha sido la Pedagogía. Es decir, esto es, algo raro en la Educación, ya que en su mayoría los estudios y clasificaciones surgen como el tratar de poner en la práctica una teoría. En el caso de las computadoras todo indica que los hechos han rebasado por lo rápido a cualquier teorización, es decir hay una urgencia en poner a funcionar algo de lo cual todavía no se tiene una teoría. Esto ha empujado e improvisado las experiencias que con las computadoras se han hecho en la Educación. Aquí ha prevalecido un gran pragmatismo, típico de la tecnología. No es sino a posteriori que se trata de racionalizar los esfuerzos dados y sus perspectivas. La clasificación que se presentó en el capítulo anterior es un intento de esto.

El enfoque más común, en la literatura de computadoras en la Educación es el popularizado por Robert Taylor. El significado de este intento de clasificación es la ubicación de la computadora en su relación con la persona en el proceso de enseñanza.

Algunos autores han querido reducir este enfoque de Taylor a los modos de diálogo entre la computadora y el usuario, pero en realidad el enfoque de Taylor no es quién lleva la dirección del diálogo sino como el aprendizaje se da. No obstante esta diferencia, conviene anotar aquí, los modos de diálogo que puede haber. Existen dos tipos de diálogo: dirigido y libre.

El diálogo dirigido, como su nombre lo indica es la computadora o el estudiante quienes pueden conducir el diálogo o interacción sobre un conjunto pre- programado de opciones. Cuando es la computadora quien dirige el diálogo, se ha experimentado como una forma de tutorial el presentar el material a aprender a través exclusivamente de preguntas como en un diálogo. Esto es una variante o estrategia del modo tutorial dirigida a los alumnos que desconocen totalmente el tema que se desea enseñar y que rechazan la lectura y la conceptualización prefiriendo la problematización, sin embargo es ampliamente utilizada en la enseñanza asistida por computadora. Simula un diálogo entre maestro y alumno, la computadora tiene información que el alumno debe aprender a través de algún tipo de interacción. El estudiante aprendería conceptos de un área específica tal como lo haría en una situación directa con el profesor.

El diálogo libre es cuando no existe un plan pre- programado de enseñanza y se cuentan con recursos tanto de la máquina como del estudiante para poder variar el diálogo. Así por ejemplo sería cuando un estudiante interroga una base de datos y eventualmente que la base de datos le pudiera proponer alguna relación a explorar.

El enfoque de diálogo es más instrumental que de fondo por lo que no es suficiente para entender ni clasificar el proceso de aprendizaje. Así el pensamiento de Taylor tiene mayor validez. El libro de Robert Taylor, es citado como un clásico en la literatura de la EAC, tiene un título cuya traducción sería "La computadora en la educación: Tutor, Herramienta y Aprendiz". El tema central de esta obra es el pretender básicamente dividir los roles o papeles educativos de la computadora en tres categorías: como tutor o maestro, como herramienta auxiliar del aprendizaje y la computadora como aprendiz que necesita ser enseñada a realizar las cosas. Esto ha sido uno de los grandes marcos conceptuales que se han utilizado por la mayoría de los libros en educación y computadoras, el clasificar las diferentes modalidades a partir de estos tres tipos. La propuesta de Taylor es que el rol más importante y de mayor futuro o impacto deberá ser la última, la de aprendiz.

La computadora como maestro.

En este caso la computadora asume el rol del maestro, de alguna manera la computadora es el tutor. Para esto la computadora tiene que estar programada por expertos, el sujeto de la enseñanza es el estudiante, el cual recibe el material, contesta a preguntas y es evaluado por la computadora, en el mejor de los casos trata de acomodarse a un amplio rango de estudiantes y sus diferencias. Cuando se habla de computadoras como maestro o tutor se está entonces aquí agrupando a todo lo que se conoce actualmente alrededor de la Enseñanza Asistida por Computadora. Es decir se agrupa a todas las formas de enseñanza así: la Instrucción Asistida por Computadora (en Inglés Computer Managed Instruction con siglas CMI), el Entrenamiento Asistido por Computadora (en Inglés Computer Assisted Training, con siglas CAT). Se debe de notar que aquí el Aprendizaje Asistido por Computadora, se le separa de la enseñanza y se le deja para el rol de aprendiz. Sin embargo muchos autores tienden a confundir la Enseñanza Asistida por Computadora con el Aprendizaje Asistido por Computadora, presentándolo a este último como una moda o un enfoque más activo de lo mismo.

En el rol tutorial se agrupan aquellas prácticas con la computadora que tienen las siguientes características:

- ❖ Existe un material pre - definido y establecido de conocimientos a enseñar
- ❖ Este conjunto de información, habilidades o conocimiento, se encuentra embebido dentro del paquete o programa de cómputo, no es una entidad aparte
- ❖ La computadora asume el rol directivo frente al alumno, en la que la computadora dirige, propone o directamente enseña o muestra algo que el alumno tiene que resolver, estudiar, repetir, practicar o aprender.

Así la computadora bajo este rol de maestro es tutor en un sentido amplio y no restringido del término ya que no solamente las lecciones tipo un libro o salón de clases como la modalidad tutorial (como se verá más adelante), caen en este punto sino también las otras modalidades como la ejercitación y práctica con la computadora, así como ciertas formas de simulación y juegos.

Bajo este enfoque de tutor, a menudo también se le agrega o combina, la Instrucción Administrada por Computadora, ya que al igual que la Enseñanza Asistida por Computadora, tiene el objetivo de controlar al alumno y ayudar a este rol de "imposición" sobre el alumno.

El caso de la Instrucción Administrada por Computadora (cuyas siglas en Español serían IAC, en inglés es muy conocido bajo sus siglas de CMI o Computer Managed Instruction), involucra el uso de computadoras para administrar el proceso en sí mismo del aprendizaje, por ejemplo haciendo seguimiento de los avances de los alumnos, inspeccionando sus archivos de respuestas, frecuencia de uso etc. que hacen con lecciones computarizadas de EAC, administrando pruebas, asignando tareas, o asignaciones, indicando las lecciones estudiadas a los estudiantes. etc.

La Instrucción Administrada por Computadora no utiliza únicamente material computarizado. El IAC puede administrar o emplear bajo la supervisión de la computadora también al material tradicional como proponer a los estudiantes lecturas de libros, consultas a la biblioteca, realizar prácticas de laboratorios, realizar discusiones con otros alumnos, etc. La IAC es también un gran respaldo a la responsabilidad administrativa del y en el salón de clases, es decir, permite el programar las lecciones y exámenes, preparar los ejercicios y el resolverlos, corregir asignaciones y exámenes y guardar registro de estudiantes, así como sus notas asistencias, trabajos entregados, etc.

El proceso de diagnóstico o corrección de exámenes es también, en buena medida la aplicación más generalizada del IAC, ya que proviene de una historia en la automatización de exámenes. Antes de las computadoras se empleaban hojas de respuestas con perforaciones, marcas que se podían comparar y más modernamente los lectores de marcas de lápiz (en Inglés "scanners"). Estos procesos permiten no solo la corrección, como se hacía antes con lectoras de marcas de lápices en pruebas de opción múltiple, sino también actualmente directamente de lectura de caracteres de maquinilla con lectores ópticos especializados o más simplemente con el teclado de la computadora o con tarjetas perforadas; se piensa que en un futuro no muy lejano se podrá también contestar mediante escritura manual siempre y cuando esta sea legible.

El proceso de exámenes con computadora también involucra el diseñar exámenes (con varios modelos incluso) y suministrarlos. Aunque sigue siendo el tipo de examen de selección múltiple el que predomina, mediante la computadora es posible analizar otro tipo de respuestas. En especial si el estudiante responde desde una terminal es factible incorporar otro tipo de pruebas, ya que los lectores ópticos de caracteres todavía son un instrumento caro, delicado y a veces impreciso. Lo importante de tener la corrección de los exámenes en computadora es el poder diagnosticarlos, hacer cruzamiento de información lo que permitiría saber por ejemplo cuestiones tan sencillas como saber cuales fueron las preguntas que más fallaron los estudiantes y qué relación tienen entre sí, por ejemplo si pertenecen a un mismo tema; lo que implicaría que tal vez se tenga que repasar esa unidad, sacar el perfil del estudiante, etc.

Resulta también muy importante el ir guardando rastro de los progresos que está haciendo cada estudiante. Con estos rastros y mediante programas muy sencillos es factible visualizar el progreso, compararlo con su historial y con los demás estudiantes e intentar un primer diagnóstico sobre las dificultades y ventajas de cada alumno. Hacer una especie de estudio de "fortalezas y debilidades" de cada estudiante para así poder suplir sus debilidades con ayuda de sus fortalezas.

La computadora como herramienta

Como sistema innovador considera que la informática puede ser utilizada como un medio o herramienta en y para la enseñanza. Una gran expectativa de la informática se ubica en aquellas modalidades educativas que ya sea por el bajo número de docentes y la poca infraestructura existente propicien el uso de medios y herramientas masivos para la educación.

Se debe tener en mente que el uso de estos dispositivos sea para recolectar, procesar, memorizar y/o transmitir información, puede ayudar al estudiante a incrementar sus conocimientos sobre aspectos específicos y mejorar sus habilidades en áreas tales como estrategias de búsqueda (recuérdese que la mayoría de los problemas en Inteligencia Artificial hacen uso de estrategias de búsqueda, por lo que es una función inteligente por excelencia) y clasificación, en la escritura, en la conceptualización, en las matemáticas.

Además de los cursos de computación (especialmente los de "literaria") el otro gran uso actualmente en las escuelas es el empleo de las herramientas computarizadas. El impacto de las herramientas computarizadas se ha menospreciado, tal vez por que no se ha entendido bien su papel importante. Las herramientas son extensiones de las capacidades mentales en un campo o dominio especializado, proveyendo lo que se necesita para realizar las tareas con menor esfuerzo. Tal vez en esto último radica su gran popularidad, las herramientas no son hechas para los programadores, son diseñadas para que se evite el programar, hasta un cierto nivel. Ya se indicó que no todo mundo va a ser programador, pero si todo

mundo debiese ser un usuario inteligente de las computadoras, y con las herramientas es la manera de hacerlo fácilmente. Las herramientas han empezado a sustituir el papel que tenía la enseñanza de la programación en los cursos elementales de "literaria" o alfabetización, ya que normalmente se sabía de estos cursos son una "embarrada" de conocimiento en programación pero sin poder utilizar efectivamente la computadora en lo cotidiano.

La característica principal de las buenas herramientas es su flexibilidad son independientes de la aplicación que se vaya a hacer, es decir que sean neutras ante lo que se puede enseñar o aprender. No contiene nada del contenido del sujeto del tema a aprender. Pero es a través del uso con la herramienta de un tema, que se aprenden indirectamente algunas propiedades o se ejercitan algunas habilidades. Sin embargo cada vez más surge paquetería que se puede considerar como herramientas que aunque circunscritas y con algo de contenido de la aplicación son instrumentos de uso y aprendizaje, por ejemplo la paquetería contable permite en cierta medida el aprender contabilidad.

Bajo este rol de herramienta, la computadora, se emplea para una multitud de pequeños problemas y necesidades que surgen y necesitan algún tratamiento de la información o su almacenamiento. La mayoría de las escuelas hoy en día, usan a las computadoras de esta manera. Ya dejó de ser una vergüenza el emplear a la computadora como una maquinilla de escribir, ya que el escribir también es una función inteligente y la presentación de un texto sin errores y con calidad, se vuelve cada vez más una exigencia. Así se les permiten ya en muchas escuelas, incluso a los estudiantes el realizar sus asignaciones mediante un "procesador de textos" o realizar algunas tareas, dibujos y cálculos de tipo general con una computadora. No será raro ver el día en que detrás de cada escritorio en la biblioteca haya una computadora, para consultar, escribir notas, hacer cálculos etc. Además del uso de paquetes tipificados como herramientas, en algunos textos se presenta como una modalidad aparte a la Consulta y Solución de Problemas ("Problem Solving"). Bajo este aspecto la computadora sirve para consultar bancos de información externos como los bancos bibliográficos o los de datos estadísticos. En el caso de solución de problemas frecuentemente se piensa en la computadora como una calculadora programable de gran memoria. Lo que evita el consultar tablas trigonométricas o de logaritmos, resolver ecuaciones y cualquier otro cálculo matemático

En algunos salones de clases los profesores también utilizan a la computadora como una herramienta auxiliar que permite obviamente realizar cálculos en las clases de matemáticas, ciencias o administración, para probar fórmulas, para generar ejercicios numéricos o hacer simplemente demostraciones de laboratorio. Otros ejemplos podrían ser el desarrollo de una base de datos sobre un tema para que él o sus compañeros puedan consultar toda la información sobre se tema. Existen numerosos paquetes de este exitoso rol.

La computadora como aprendiz

En este caso, la computadora asume el papel del alumno que necesita ser enseñado para realizar algo. Entonces el estudiante es quien enseña a la computadora. Para realizar lo anterior el estudiante se comunica con la computadora mediante un lenguaje. Aquí la enseñanza que recibe el usuario o estudiante es indirecta, ya que no puede enseñar lo que no conoce y puede ser parcial es decir, el humano le enseña algo que no entiende en su totalidad.

Este rol se presenta como la alternativa computarizada entre la máquina de enseñar versus la máquina de aprendizaje. Algunos de los pioneros en computación educativa, particularmente Arthur Luehrmann y Seymour Papert, se dieron cuenta que la mayoría de las aplicaciones educativas con la computadora, habían sido pensadas como máquinas que puedan enseñar, más que como máquinas de aprendizaje. Estos autores han argüido que una buena parte de las aplicaciones educativas podrían y debería ser empleando las capacidades más inteligentes de una computadora. Pero sobre todo devolverle el papel conductor al estudiante o sujeto del aprendizaje.

Así a manera de metáfora, dado que es el estudiante el guía en lo que quiere aprender, es creativo y diseña como puede aprender empleando la computadora, es que se dice que la computadora es enseñada o es el aprendiz. La idea de base es muy antigua, ya que la mejor manera de aprender es enseñando, esto obliga al maestro a reflexionar e interiorizarse de un tema no solo lo suficiente para manejarlo, sino para responder cualquier duda o situación que se presente con él. En estricto sentido esto no está ausente de paquetes que le sirven de herramientas, desde lenguajes hasta simulaciones y proyectos computarizados. Tampoco está totalmente ausente de contenido de lo que puede aprender con una de estas herramientas sui generis. El enfoque de "enseñar" a una computadora no es extraño, de hecho este último sentido es el más natural para una computadora, ésta no sabe hacer nada por sí misma hasta que se le dan los programas o instrucciones para que haga algo.

Sin embargo el que tiene necesidad de aprender es el estudiante. En este caso se invierte el rol y de alguna manera el estudiante que aprende trata de enseñar a la computadora no sólo las cosas que tiene él que aprender sino frecuentemente tiene que enseñarle además el cómo están hechas esas cosas, que relaciones tienen etc. Cosas por demás interesantes y altamente formativas que además sirven de refuerzo a la memorización pura. Frecuentemente los maestros señalan que ellos verdaderamente han entendido un tema cuando lo han tenido que explicar a otros, esto los ha obligado a interiorizarse del tema a cuestionarlo y no sólo a repetirlo. Algunos ejemplos de esto serían los nuevos paquetes que simulan una situación (por ejemplo el espacio interplanetario, o la bolsa de valores) en la que el estudiante recorre o trabaja y de manera indirecta se da cuenta de los mecanismos que controlan la situación sin que estos se le digan explícitamente, el estudiante "los descubre". Otros ejemplos podrían ser el diseñar un programa que enseñe a la computadora a hacer cierto tipo de figuras, o imágenes, e incluso geometría (es así que se presenta generalmente al Lenguaje LOGO como un instrumento de este rol). De manera más sofisticada y difícil para el estudiante, es el pedirle que empleando algún lenguaje de autor realice una lección para enseñarle algo a sus compañeros y todavía más difícil el pedirle que realice un sistema experto que genere respuestas en una combinatoria de búsqueda de soluciones.

De esta manera se rebasa el nivel únicamente de usuario de la computadora y se toma un rol activo en su propio proceso de enseñanza- aprendizaje, además de aprender a usar la computadora en su vida diaria. Estos autores y sus seguidores indican que de esta manera al tratar de enseñar no sólo se mejoran sus procesos cognitivos, sino que colateralmente tiene el estudiante que desarrollar otras habilidades, como las de expresión, análisis de un problema etc.

La computadora como auxiliar administrativo

Este rol queda excluido dentro del esquema de Taylor, pero dada su importancia y para completar la clasificación que se dio en líneas anteriores es conveniente presentarla brevemente.

A veces es omitido el gran papel que juega la computadora en la administración de una escuela. En la escuela como en cualquier otra oficina se necesita de una telefonista que se encarga de dirigir las llamadas a las extensiones, hacer citas y recordarlas, informar sobre horarios, dar mensajes, informar si una persona se encuentra y localizarla etc., con la ayuda de una computadora esto será posible. Dentro de esta modalidad la computadora es un gran apoyo a la administración clásica de las escuelas como de cualquier otra empresa.

La administración clásica en una escuela tiene una gran cantidad de tareas rutinarias específicas al servicio que presta a la comunidad, como el envío de circulares a los alumnos y sus padres de familia, el recibir la matrícula y mensualidades de los estudiantes, el pago de la nómina, la contabilidad de la escuela, el inventario, el asignar horarios, salones y cursos a los maestros, contar con listas de los alumnos y otras aplicaciones.

Es frecuente en las escuelas que el primer interés de emplear la computadora administrativamente gire alrededor del rol de memoria ágil y gigantesca que permita en cualquier momento encontrar los registros de los estudiantes: notas, pagos, salones, seriación de cursos, emitir boletas, transcripciones de créditos, certificados, etc., pero lo más importante es que permite sintetizar buena parte de esta información.

MODALIDADES DE LA ENSEÑANZA ASISTIDA POR COMPUTADORA

La Enseñanza Asistida por Computadora hace que recaiga parte de la responsabilidad de la enseñanza en la computadora misma. En este caso se trataría de transmitir al alumno algún tipo de conocimiento por ejemplo una lección o un curso en su totalidad, a través de la computadora. Como ya se indicó con anterioridad esto se puede realizar, ya sea proveyendo instrucción o ejercicios, pero también proponiéndole juegos o simulaciones en las que para manejarlos o dominarlos, el estudiante tiene que aprender algo directamente que se le enseña. Las principales modalidades de la Enseñanza Asistida por Computadora son: la forma tutorial, la de ejercitación y práctica, y los juegos y simulaciones didácticas

1.- Forma tutorial.

Esta forma se desenvuelve basándose en la lógica de la enseñanza programada. En vez del texto escrito en una página de un libro, el texto aparece en la pantalla de la computadora, tal vez la única diferencia es que los ejercicios o preguntas en vez de ser dejados hasta el final de un capítulo, se van insertando a medida que se avanza en el desarrollo de la lección. El aprendiz contesta las preguntas que se le someten a consideración y la computadora le responde si estuvo correcto o no. Eventualmente como en la enseñanza programada de tipo ramificada las respuestas del estudiante dirigen la secuencia de la próxima unidad a aprender. El papel del estudiante se limita a responder, no le está "permitido" plantear nuevos interrogantes a la máquina, otros que sobre el funcionamiento. Puede dar la impresión de enseñanza individualizada y su principal ventaja es la de ir al ritmo del estudiante.

2.- Forma de Ejercitación y Práctica.

A esta forma se le conoce a menudo bajo su denominación en Inglés "Drill and Practice" Esta modalidad consiste en ayudar al aprendizaje mediante la realización de numerosos ejercicios. La computadora propone una serie de ejercicios programados que el estudiante debe resolver y ayuda a reforzar los conceptos adquiridos previamente. La máquina indica el camino a seguir en caso de error, lo cual permite generar la ilusión del auto-control en el proceso de conocimiento. Dado que el presentar los problemas en abstracto es poco divertido, frecuentemente esta modalidad es disfrazada o sumergida en una historia fantástica, por ejemplo un mago que enseña y propone adivinanzas, una caja registradora que enseña a sumar y restar proponiendo la administración de dicha caja, etc.

3.- Formas de Simulación y Juegos Didácticos.

Esta modalidad puede tener diversas alternativas. Todas estas formas giran alrededor de sumergir al estudiante en un proceso dinámico de cambio, sea real o imaginario. Lo importante es que ya no son preguntas directas las que tiene que contestar el estudiante sino el maniobrar o manipular la simulación o ser un contendiente en el juego. El tema del juego o la simulación es el tema del sujeto que se quiere enseñar.

Una de las formas es la que se puede denominar "descubrimiento guiado" (o inducido). El estudiante puede realizar experiencias pre-programadas simuladas orientadas a redescubrir las relaciones entre los parámetros que definen por ejemplo una ley de la Física, de la Biología, y otras disciplinas. Lo anterior adquiere una mayor importancia en juegos (llamados así por que hay varios contendientes) que simulan la complejidad en la toma de decisiones en la Administración y Ciencias Socio - económicas, pero rara vez dan resultados mas allá de una sensibilización.

Se dice también que el estudiante puede "adquirir experiencia" introduciendo variaciones en algún parámetro del modelo estudiado. La computadora da una respuesta inmediata acerca de los efectos de la variación introducida, en el conjunto del modelo. Por último, el estudiante también puede "construir" su modelo y controlar su validez. A veces se vende como juego computarizado algunas versiones de la modalidad de Ejercitación y Práctica como es el caso del juego del "ahorcado".

A continuación se muestra una distribución típica de algunos de los usos didácticos de la computadora que se han hablado aquí:

USO RELATIVO EN LA DECADA PASADA DE ALGUNAS MODALIDADES DE LA ENSEÑANZA ASISTIDA POR COMPUTADORA Y USOS EDUCATIVOS.	
ALFABETIZACION COMPUTACIONAL	36 %
EJERCITACION Y PRACTICA	24 %
CONSULTA Y SOLUCION DE PROBLEMAS	20 %
SIMULACION Y JUEGOS	14 %
TUTORIALES	6 %

Información obtenida de Internet basándose en estudios realizados en el año de 1979

“ Desarrollo y Utilización de Sistemas de Enseñanza Asistida por Computadora ”

A continuación se presentan tres ejemplos de sistemas de EAC, mostrando que es lo que ofrecen, donde se utilizan y sus propuestas educativas:

- ❖ Sistema: Biblioteca Virtual de la Fundación Universitaria Manuela Beltrán
- ❖ Diseñado por: la Fundación Universitaria Manuela Beltrán
- ❖ Entorno: Internet (Windows)
- ❖ País de origen: Bogotá Colombia.



La Biblioteca Virtual de la Fundación Universitaria Manuela Beltrán es una institución que brinda apoyo fundamental a la educación e investigación en el nororiente colombiano. Queda situada en la ciudad de Bucaramanga (Santander), una ciudad de alto desarrollo tecnológico, social y cultural. Contamos con equipos de alta tecnología para acceso a Internet, Realidad Virtual y consultas concurrentes de bases de datos científicas. Además poseemos personal altamente calificado en las distintas áreas tecnológicas y de desarrollo científico



Su Misión.

Ser una institución líder en el ofrecimiento de servicios de información y comunicación para el desarrollo de procesos investigativos mediante el uso de nuevas tecnologías y así contribuir a mejorar la calidad de vida de los colombianos integrando los sectores educativo y productivo.

Posee los siguientes tipos de acceso :

❖ **Presencial :**

Cuenta con 100 computadores IBM Aptiva Multimedia para la consulta de Internet en su sede ubicada en la Carrera 27 No. 33-106 en Bucaramanga Colombia.

❖ **Remoto Conmutado :**

Tiene un canal de alta velocidad al backbone de Internet en Estados Unidos, además de un buen número de líneas telefónicas para el fácil acceso a nuestros usuarios.

❖ **Remoto Dedicado :**

Posee un grupo de diseño en Internetworking, el cual brinda la mejor solución en su conexión institucional a Internet.

Realidad Virtual Estamos en el proceso de adquisición e instalación de equipos para la ejecución y desarrollo de ambientes de realidad virtual completos (para todos los sentidos posibles hasta hoy) de muy alta tecnología únicos en la región.

Se esta desarrollando el proyecto de poner toda la Biblioteca Virtual en VRML y Java para que nos consulten en realidad virtual vía Internet.

[\[Misión\]](#) [\[Servicios\]](#) [\[Búsqueda\]](#) [\[Capacitación\]](#) [\[Sugerencias\]](#)

Este sitio y todo su contenido son propiedad de la Biblioteca Virtual de la Fundación Universitaria Manuela Beltrán. Cualquier reproducción parcial o total será castigada por las leyes de derechos de autor vigentes. Biblioteca Virtual UMB © 1998

- ❖ Sistema: Escuela virtual
- ❖ Diseñado por: La Escuela Virtual
- ❖ Entorno: Internet (Windows)
- ❖ Pais de origen: México.



Introducción

La Escuela Virtual es una organización no lucrativa, integrada por personas con distintas formaciones profesionales e intereses, nacida con el afán de contribuir al aprendizaje por medio de Internet.

La Sociedad Contemporánea pone al alcance de los jóvenes una cantidad enorme de información, tanta, que es preciso crear herramientas que faciliten su clasificación y comprensión para que se pueda obtener el mayor provecho de ella. El uso de estas herramientas tecnológicas permite a los estudiantes expandir sus oportunidades de aprendizaje duradero y significativo, y sirve como un apoyo en las labores de investigación y desarrollo de proyectos. Así, los prepara para el Siglo XXI.



Misión

En México, como en el resto del mundo, el número de usuarios de Internet crece aceleradamente, un buen número de estos, son jóvenes entre 13 y 17 años. Sin embargo, no existen muchos recursos diseñados especialmente para los estudiantes y sus maestros, ni suficientes centros de información en español que sean útiles para ellos.

La mayoría de los jóvenes utiliza Internet con fines de entretenimiento, buscando juegos, personajes, artistas, y comunicándose con otros jóvenes de distintas partes del mundo. Introduciendo material educativo organizado de manera amena y fácil de consultar, utilizamos Internet de una manera más acorde con los valores que se busca fomentar en la educación.

La Escuela Virtual, nace entonces con la misión de ser un auxiliar de los jóvenes, sus padres y sus maestros en la educación secundaria, de manera que se utilice Internet como una herramienta tanto para enseñar como para aprender. La Escuela Virtual pretende convertirse en un lugar de encuentro virtual, donde personas de distintas edades y formaciones puedan colaborar, interactuar, publicar y descubrir recursos de aprendizaje.

Queremos que la Escuela Virtual se convierta en un lugar de referencia cuando se busque información para la tarea, o recursos para el salón de clases. Queremos, mediante esta página, que la experiencia de aprender sea significativa y divertida.

Ventajas

Cada día, son más las personas que obtienen información, venden o compran productos, consiguen trabajos utilizando Internet; dada la cantidad de información que se maneja, es preciso construir un espacio dedicado a los jóvenes y maestros, de manera que toda la información contenida en éste sea en beneficio de su desarrollo. En un espacio de esta naturaleza, se maximizan los beneficios de Internet.

Objetivos

Los objetivos de la Escuela Virtual, son los siguientes:

1. Convertirse en un *centro de acopio* de información, principalmente en español, enfocado específicamente a los estudiantes de secundaria y sus maestros.
2. Proporcionar a los estudiantes información sobre temas específicos de sus tareas escolares, tanto ofreciendo documentos como mediante un asesor *"en línea"*.
3. Involucrar a los maestros en la creación de nuevos recursos educativos.
4. Ser una herramienta para los maestros, un lugar donde puedan descubrir, discutir y proponer nuevas ideas.
5. Motivar a los estudiantes a descubrir nuevas maneras de aprender, más allá de la clase y del texto escolar.

En la Escuela Virtual, pretendemos facilitar las tareas de los estudiantes, haciendo accesible para ellos información sobre sus tareas escolares, de manera que éstas se conviertan en un quehacer agradable, divertido y provechoso. Utilizamos la tecnología más avanzada de manera que se obtenga el máximo beneficio educativo. No pretendemos sustituir a la escuela como opción educativa, no hacemos sus tareas ni resolvemos sus problemas. Es importante señalar que en la Escuela Virtual, el maestro se convierte en un *estudiante activo*, ya que se le facilita la búsqueda de nuevos y mejores recursos en la enseñanza, al mismo tiempo que conoce sobre lo que se hace en distintos lugares del mundo.

En la Escuela Virtual, trabajamos diariamente buscando información relacionada con el curriculum escolar mexicano, sobre todo de nivel secundaria. Pretendemos que la información que se encuentre en la página sea preferentemente en español, sin embargo, se incluyen algunas excelentes referencias en inglés. Dada la naturaleza cambiante de Internet, sabemos que esta no será nunca una tarea terminada, por lo que nuestra página se actualiza permanentemente.

- ❖ Sistema: programas para la enseñanza de Bioquímica asistida por computadora.
- ❖ Diseñado por: El Departamento de Bioquímica y Biología Molecular de la Facultad de Medicina de la UNAM
- ❖ Entorno: Windows
- ❖ País de origen: México.

ENSEÑANZA DE BIOQUIMICA ASISTIDA POR COMPUTADORA.

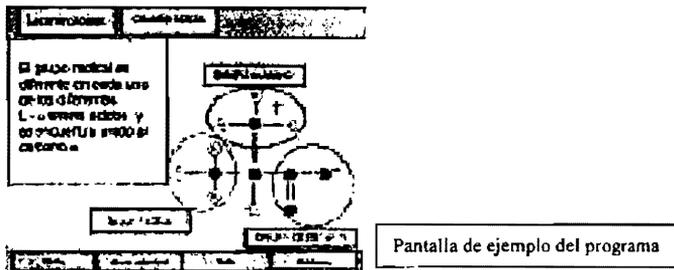
LABORATORIO DE DESARROLLO DE PROGRAMAS DE CÓMPUTO PARA LA ENSEÑANZA DE BIOQUÍMICA ASISTIDA POR COMPUTADORA

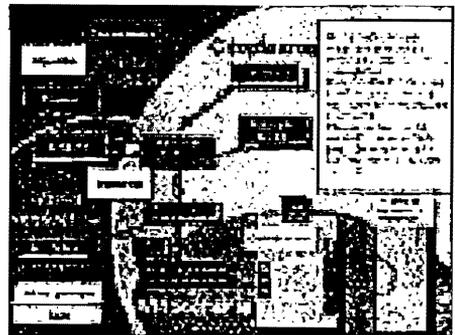
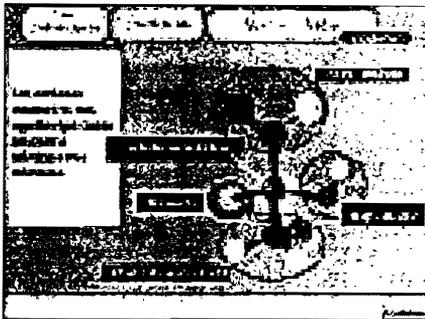
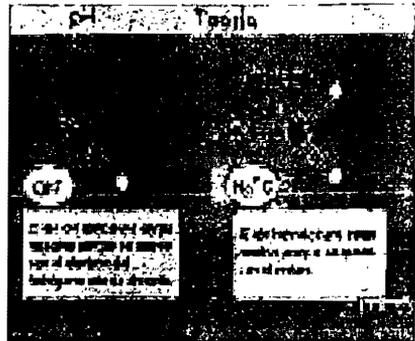
En el Departamento de Bioquímica y Biología Molecular de la Facultad de Medicina de la UNAM, se han desarrollado varios programas para la enseñanza de Bioquímica asistida por computadora. Los programas, tienen como objetivo el proveer a los alumnos y docentes con una herramienta adicional para repasar algunos temas básicos de Bioquímica en forma fácil y entretenida. Los programas han sido probados con alumnos del curso de Bioquímica y Biología Molecular del primer año de la Carrera de Médico Cirujano, con muy buenos resultados, mejorando en todos el aprendizaje del tema. Las demostraciones de los siguientes programas están disponibles:

- pH
- Los aminoácidos
- Los carbohidratos
- Digestión y transporte de carbohidratos
- Glucólisis y glucogenolisis

Algunas pantallas de los paquetes en acción

Estos programas requieren una computadora PC con 1.5Mb de espacio en disco duro para cada programa, 2 Mb de memoria RAM, Windows 3.1 o mejor y monitor Súper VGA con 256 colores. Los programas tienen un costo de \$25.00 USD Dlls cada uno por licencia individual.





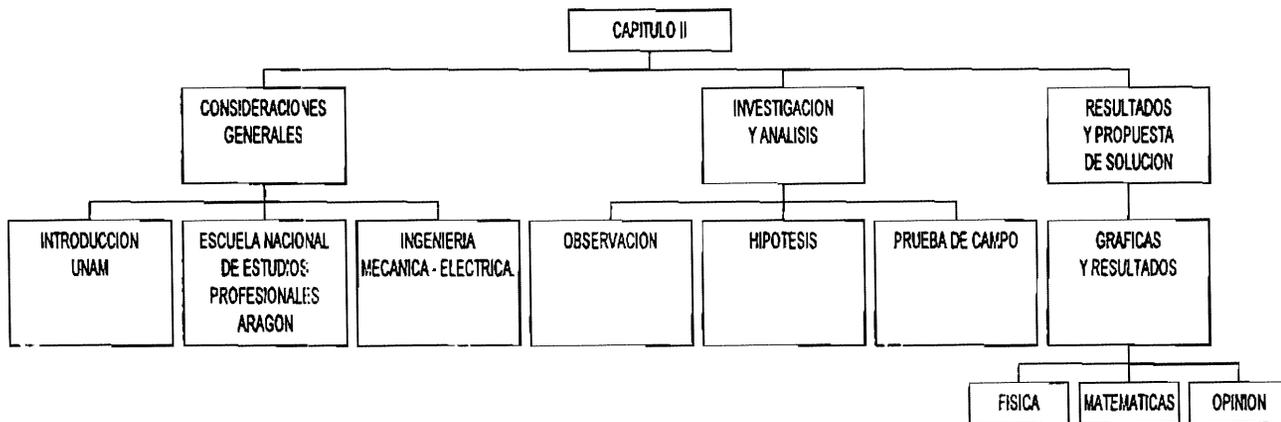
Pantallas de ejemplo del programa, donde por medio de imágenes y textos se trata de aplicar la enseñanza virtual.

Como se puede observar existen diversas instituciones dedicadas al desarrollo y uso del EAC; que están diseñando, desarrollando y explotando esta herramienta. Existen muchas mas que podrían ser excelentes ejemplos de esto, pero considero que mostrando algunos ejemplos de lo que se ha hecho se puede tener una clara visión de lo que será la Educación virtual y su tendencia al futuro; y esto solo es el comienzo.

CAPITULO II

ESTUDIO DE CAMPO

ESTUDIO DE CAMPO





CONSIDERACIONES GENERALES

INTRODUCCION



UNAM

La Universidad Nacional Autónoma de México fue fundada hace más de 400 años, es una institución líder en el continente americano, heredera de una extensa tradición que se remonta a la Real Universidad de México, inaugurada por el virrey don Luis de Velasco en el año de 1553.

Esta institución, en función de su historia y sus logros, ha marcado la pauta de la actividad intelectual, científica, humanista y tecnológica, constituyéndose en el centro de educación superior de mayor relevancia en el país. Como ha señalado el doctor José Sarukhán, rector de la UNAM, "La universidad constituye el proyecto cultural más importante que ha logrado nuestro país en este siglo".

A la UNAM le corresponde asumir el compromiso que tiene con la sociedad de conservar, generar y transmitir el conocimiento científico, humanístico, artístico y tecnológico mediante la docencia, la investigación y la difusión de la cultura.

En su larga historia, la UNAM ha experimentado cambios profundos. De un conjunto de escuelas diseminadas se ha transformado en un complejo sistema universitario, producto primordialmente de su desarrollo académico, de la diversificación de sus servicios, y de la composición de una estructura académica y administrativa congruente con ese desarrollo.





Este complejo sistema académico cuenta con cuatro niveles educativos: el bachillerato, el técnico, la licenciatura y el posgrado; administra el Sistema Universidad Abierta; posee un amplio y diversificado campo de disciplinas, temas y problemas especializados que son el objeto de estudio de los institutos y centros de investigación científica y humanística; desarrolla un variado conjunto de actividades académicas y artísticas de extensión, y tiene una amplia red de servicios de apoyo.

En sus institutos y centros se realizan la mayor parte de la investigación que se genera en el país. Cuenta para esta función con instalaciones a lo largo de todo el territorio nacional, que van desde laboratorios hasta dos buques oceanográficos, dotados del equipo del más alto nivel.

En el campo de la docencia, la UNAM representa la opción educativa más importante en el ámbito nacional, desde la educación media superior hasta el posgrado.

En el nivel de estudios profesionales, la Universidad Nacional en sus 6 campus- Ciudad Universitaria, Acatlán, Aragón, Cuautitlán, Iztacala y Zaragoza.- por medio de sus facultades y escuelas, y de la unidad Académica de los Ciclos Profesional y de Posgrado el Colegio de Ciencias y Humanidades ofrecen 69 Carreras de nivel licenciatura y siete de nivel técnico. De ellas, el Sistema Universidad Abierta imparte 17 licenciaturas y una carrera técnica. En el posgrado imparte más de 300 especializaciones, maestrías y doctorados. Así mismo, desarrolla un amplio programa de educación Continua en sus diferentes modalidades, como son cursos, talleres y diplomados, en prácticamente todas las áreas.

La Universidad posee cerca del 60 % del acervo bibliográfico nacional, distribuido en 164 bibliotecas, con la más amplia gama de servicios, lo que representa un excelente soporte para la docencia y la investigación. Cuenta con bancos de información y con los sistemas más avanzados de consulta de acervos automatizados, tanto nacionales como internacionales, en prácticamente todos los campos de conocimiento.

Su planta académica, la gama de carreras profesionales, los servicios de apoyo y sus instalaciones, son recursos de una Universidad de excelencia. Pertenecer a la UNAM es un privilegio.

Así la UNAM, comprometida con el avance y el desarrollo del país prepara profesionistas en las áreas de la ciencia, la tecnología, las artes y las humanidades. Sus egresados han hecho importantes aportaciones en el ámbito nacional e internacional mediante proyectos, investigaciones e innovaciones en distintos ámbitos.

El quehacer de la Universidad se sustenta en un marco normativo integrado por la ley Orgánica, el Estatuto General y los estatutos y reglamentos derivados de éstos.

El artículo 1º de la Ley Orgánica señala que: "La universidad Nacional Autónoma de México es una corporación pública -organismo descentralizado del Estado-, dotada de plena capacidad jurídica y que tiene por fines impartir educación superior para formar profesionistas, investigadores, profesores, universitarios y técnicos a la sociedad; Organizar y realizar investigaciones, principalmente acerca de las condiciones y problemas nacionales, y extender con la mayor amplitud posible, los beneficios de la cultura".



**Escuela Nacional de
Estudios Profesionales
Aragón**

La escuela Nacional de Estudios profesionales Aragón se encuentra localizada en el área metropolitana de la ciudad de México específicamente esta se encuentra en Av. Rancho Seco s/n en Bosques de Aragón, ciudad Nezahualcóyotl, en el estado de México. En esta institución se realizan los estudios profesionales para diversas carreras como son del área de Ingeniería, Derecho, Arquitectura, entre otras.

Dentro de sus instalaciones cuenta con biblioteca, áreas de laboratorios, centro de computó, gimnasio y canchas de diversas disciplinas deportivas y una decena de edificios en los cuales cuenta con las aulas en las que se imparten las clases de las carreras antes mencionadas.

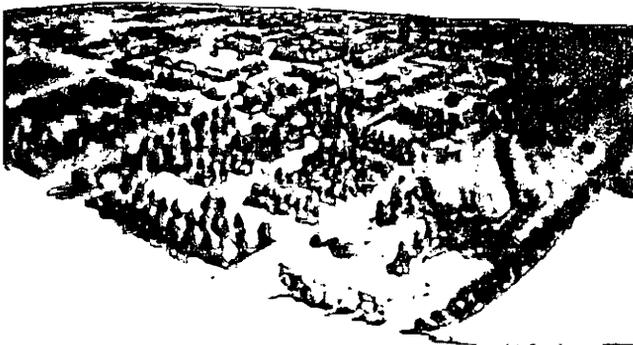
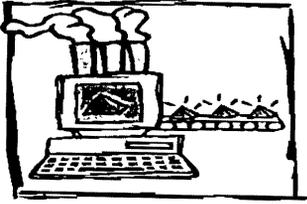


FOTO PANORAMICA DE LAS INSTALACIONES DE LA ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES ARAGON.



INGENIERIA MECANICA ELECTRICA

La carrera de Ingeniería Mecánica eléctrica forma al profesionista que, mediante el uso de la Física, las Matemáticas, las técnicas de ingeniería, economía y Administración, transforma la naturaleza por medio de dispositivos mecánicos, eléctricos y electrónicos.

Dependiendo del área elegida para desempeñarse – Mecánica, eléctrica y electrónica o industrial- enfrentará diferentes condiciones laborales, que van desde el taller o la industria, hasta el trabajo de campo y el de investigación.

La tarea que realiza este profesionista brinda diversos beneficios, ya que gracias a él, la población puede tener acceso tanto a bienes necesarios en su vida cotidiana – máquinas y herramientas- como a servicios que le son indispensables – suministro de energía eléctrica y sistemas de comunicación y de transportes.

Al concluir su preparación académica, el egresado será un profesionista responsable, organizado, reflexivo y analítico, creativo e inventivo, y con conocimientos suficientes para actuar en el diseño, la construcción, la fabricación, la adaptación, el montaje, la operación y el mantenimiento y la innovación tecnológica de sistemas en las áreas Mecánica, Eléctrica, Electrónica e Industrial y, por lo tanto capaz de contribuir en el desarrollo del país.

Para iniciar el estudio de la licenciatura, el aspirante requiere haber cursado el área de las ciencias físico – Matemáticas en el bachillerato, además de poseer:

- Conocimientos básicos de matemáticas, Física y Química.
- Facilidad en el manejo de lenguajes numéricos y formulas
- Interés por los avances científicos y tecnológicos.
- Habilidad para la solución práctica d problemas de la ingeniería Mecánica Eléctrica e industrial.
- Capacidad para el trabajo de equipo.
- Aptitud para la toma de decisiones.
- Constancia y tenacidad en la actividad emprendida.
- Facilidad para reaccionar serenamente ante emergencias.
- Inventiva y creatividad
- Buena memoria.

La situación tecnológica y económica del país requiere de mejores profesionistas egresados de esta carrera, quienes deberán ser pieza clave en la consolidación de una tecnología propia que permita la sustitución de importaciones y dé impulso al desarrollo industrial, contribuyendo así al incremento en la producción de bienes y en el mejoramiento de la funcionalidad de los servicios públicos en las áreas industrial de las comunicaciones, energéticos y transportes.



INVESTIGACION Y ANALISIS DEL PROBLEMA

Actualmente ingresan a la UNAM 1,719 alumnos de primer ingreso a la carrera de Ingeniería Mecánica eléctrica, en el campus Aragón y la Facultad de Estudios Superiores de Cuautitlán, por lo cual esta carrera se considera con una alta demanda, en estos tiempos se requieren nuevas soluciones para atender esta fuerte demanda, por lo tanto se hace necesario el crear nuevas tecnología que faciliten el proceso de enseñanza aprendizaje

En este trabajo de tesis se planteo el siguiente objetivo * Diseñar un sistema que sea una herramienta de apoyo para el proceso de enseñanza – aprendizaje para aplicarse en el área básica de ingeniería

Para el desarrollo de esta investigación se siguió el siguiente método

- ⇒ Observación
 - Delimitar el objeto de estudio
 - Fijar límites de la investigación
- ⇒ Hipótesis
 - una solución alternativa al problema
- ⇒ Diseño experimental (del alumno objeto al alumno pensante)
 - Variables
 - Independiente - aquella que no podemos controlar pero podemos analizar
 - Dependiente - Variable que depende de los resultados obtenidos
 - Prueba de campo
 - Comprobar la hipótesis con análisis del proceso de autoaprendizaje
 - Gráficas
- ⇒ Propuesta de solución

Observación

El estudio se realizó a estudiantes de primer ingreso del área mecánica – eléctrica del campus Aragón, se tomo dos grupos como muestra de la población de la carrera de IME y se les aplicó una evaluación diagnóstico que permitiría conocer las áreas de oportunidad que presentan los alumnos de reciente ingreso, ya que el software a diseñar estaría enfocado a estos alumnos

Hipótesis

Por lo tanto se planteó la siguiente hipótesis

Si los alumnos de la carrera de ingeniería presentan un bajo rendimiento académico entonces con el uso del sistema virtual de enseñanza - aprendizaje en ingeniería básica los alumnos tendrá una herramienta de apoyo para elevar su desempeño.

Diseño experimental

• Variables

✓ Independiente

Los alumnos de la carrera de ingeniería presentan un bajo rendimiento académico

Se delimitó la aplicación de este sistema a esta población estudiantil, ya que en primer instancia, son estos alumnos los que presentan mayor dificultad al realizar sus estudios de ingeniería, esto se puede atribuir a varias razones.

- Pocos o carencia de hábitos de estudio
- Falta de concientización de la importancia de los conocimientos de matemáticas y física
- Pobre motivación y por lo tanto desempeño mínimo
- Deficiente o pobre formación en el área físico – matemática
- Poca información y difusión de la carrera de ingeniería
- Conceptualización errónea de la ingeniería

✓ Dependiente

El diseño de un Sistema Virtual de Enseñanza – aprendizaje será una herramienta de apoyo para mejorar su aprovechamiento

Basándose en la investigación hecha en el capítulo uno sobre tecnologías actuales para la enseñanza aprendizaje, se contemplo que una solución posible al bajo rendimiento académico de los estudiantes, estaría en el desarrollo de una herramienta de cómputo para apoyo de sus actividades estudiantiles

• Prueba de campo.

Se aplicó un examen de diagnóstico que fue diseñado con el fin de evaluar los conocimientos de Matemáticas y física de los alumnos de primer ingreso y así demostrar el nivel de conocimientos que presentaban, este examen se incluye en el anexo B de este trabajo Cabe aclarar que la selección de reactivos se hizo basándose en consejos del asesor de tesis y fueron tomados del libro " Fundamentos de física y matemáticas " editado en la escuela Nacional de Estudios Profesionales Aragón por el Ing Fernando Alcántara Rodríguez. El examen constaba de 55 reactivos los cuales de acuerdo al tema estaban dispuestos de la siguiente forma:

SUBTEMA	REACTIVO(S)	Número de reactivos
PARTE I	MATEMATICAS	
ARITMETICA	1-10	10
ALGEBRA	11-20	10
TRIGONOMETRIA	21	1
GEOMETRIA ANALITICA	22-27	6
CALCULO DIFERENCIAL	28-29	2
CALCULO INTEGRAL	30-31	2
TOTAL		31
PARTE II	FISICA	
ESTATICA	32	1
DINAMICA	33-36	4
CINEMATICA	37-39	3
HIDRAULICA	40-41	2
TERMODINAMICA	42-43	2
ACUSTICA	44	1
OPTICA	45	1
ELECTRICIDAD	46-50	5
TOTAL		19
PARTE III	OPINION	
5 Reactivos	En esta parte se evalúa el impacto y aceptación que tendrá el sistema de software que se desea realizar.	

Relación de los reactivos para su aplicación en el examen de diagnostico de acuerdo al tema que se evaluó.

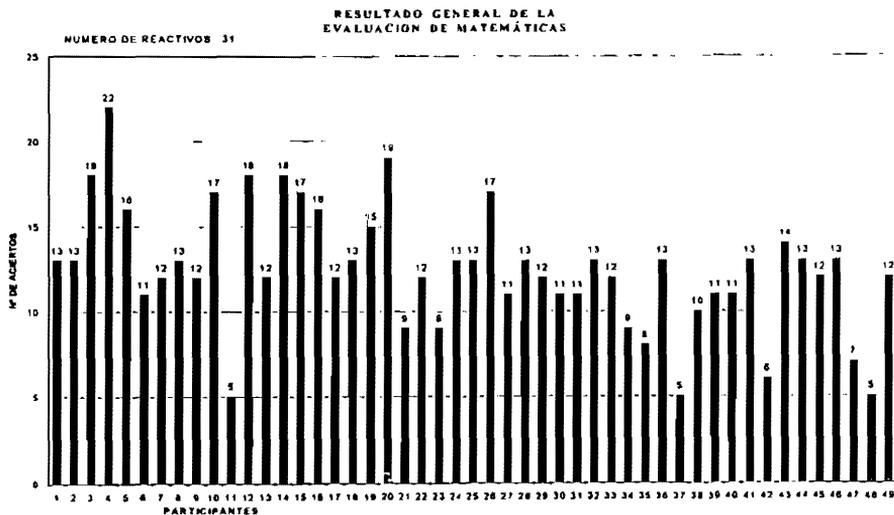
La evaluación de los exámenes arrojó los siguientes resultados:

Grupos aplicados: 2

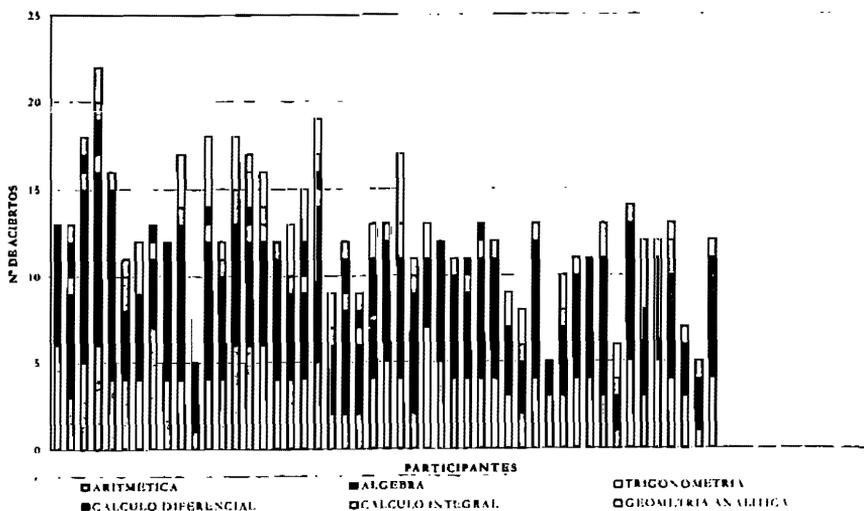
RESULTADOS PARCIALES.

RESULTADOS DEL GRUPO 1:

Número de alumnos evaluados: 49



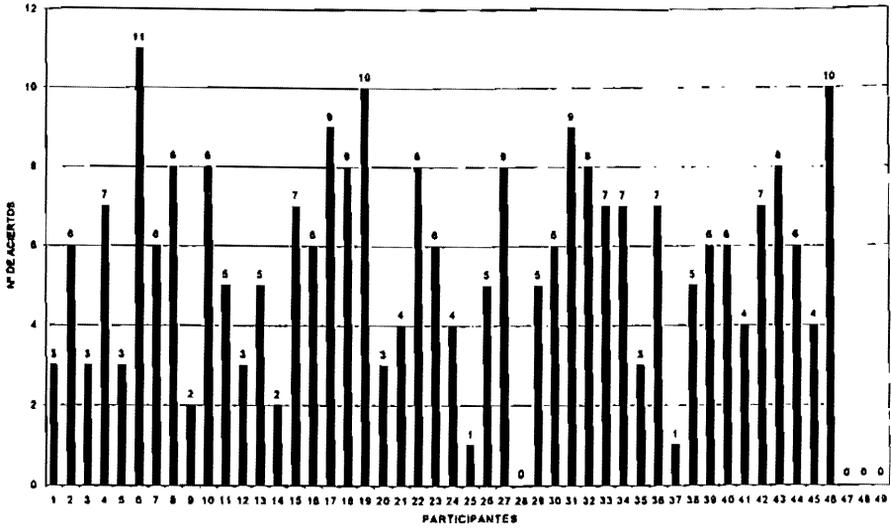
RESULTADOS DE MATEMÁTICAS POR MATERIA



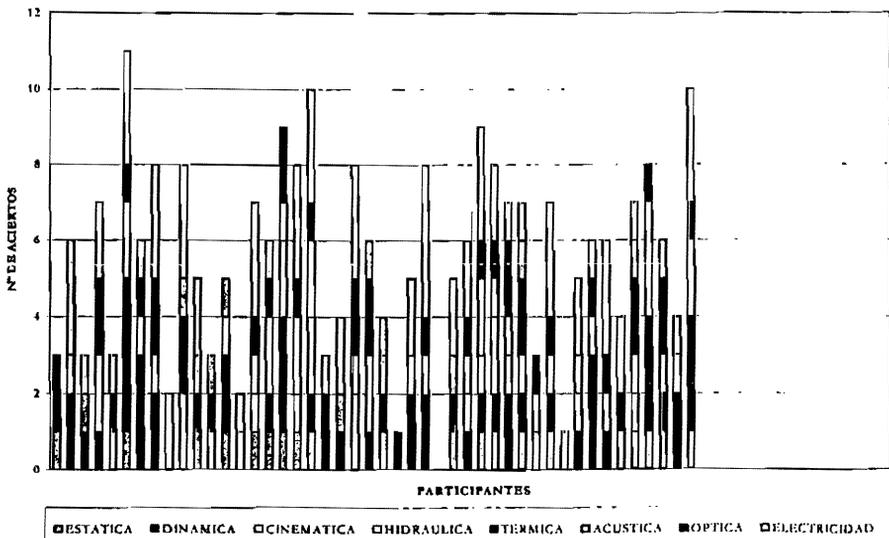
S
L

**RESULTADO GENERAL DE LA
EVALUACION DE FISICA GRUPO : 1**

NUMERO DE REACTIVOS : 10

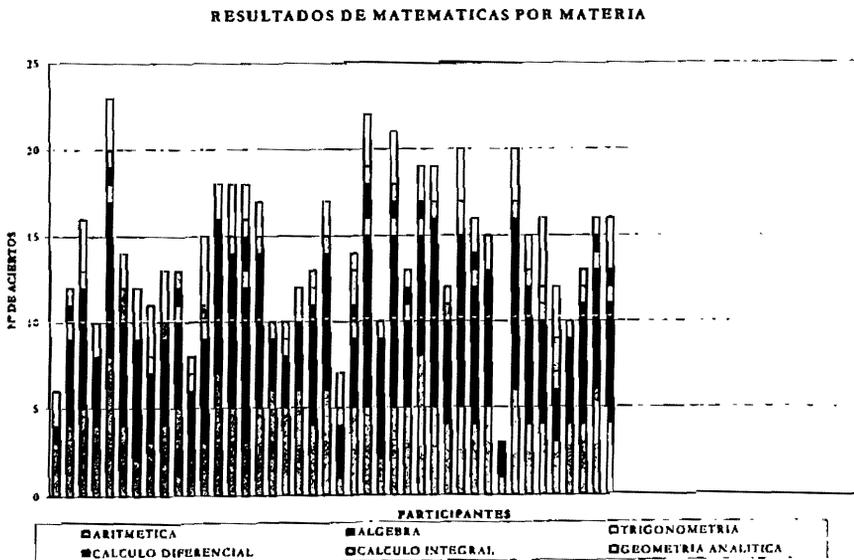
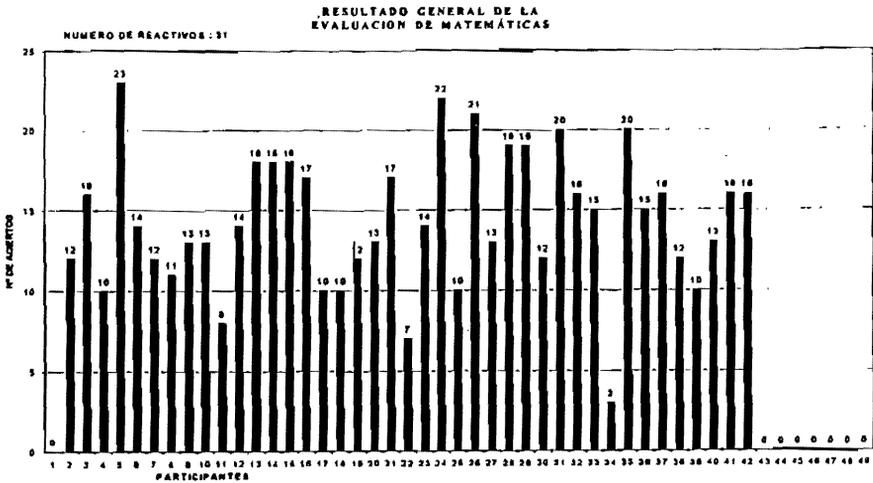


RESULTADOS DE FISICA POR MATERIA

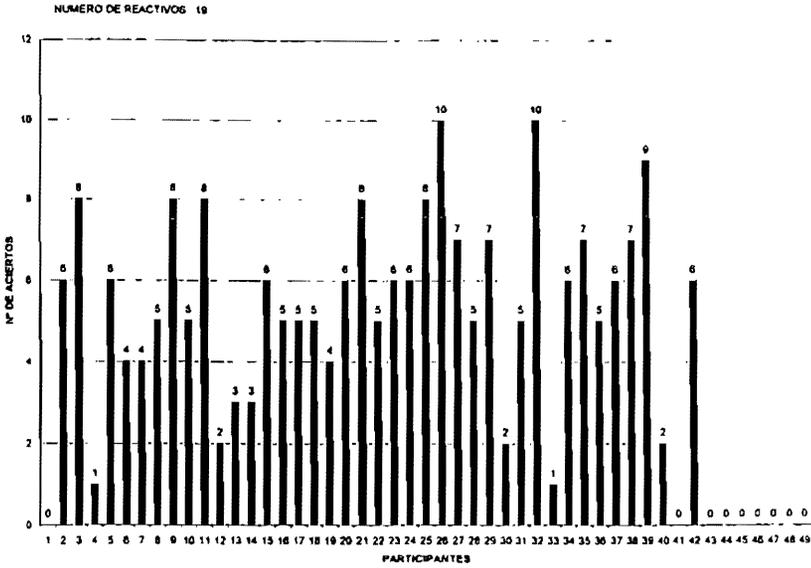


RESULTADOS DEL GRUPO 2:

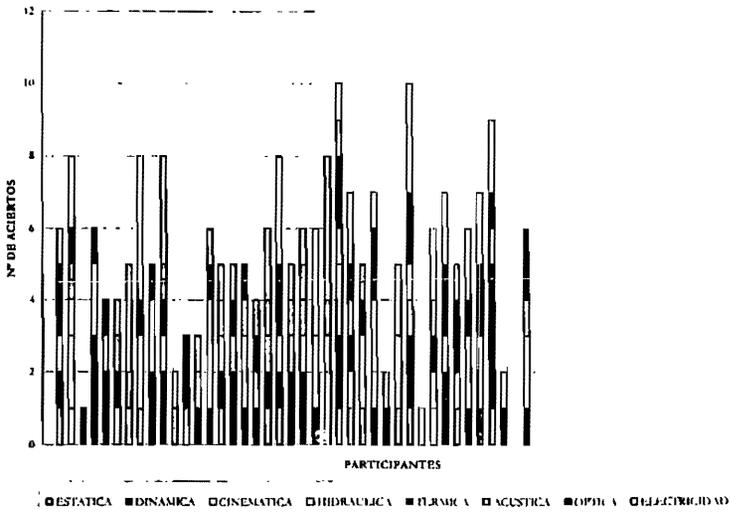
Número de alumnos evaluados: 42



RESULTADO GENERAL DE LA EVALUACION DE FISICA GRUPO : 2



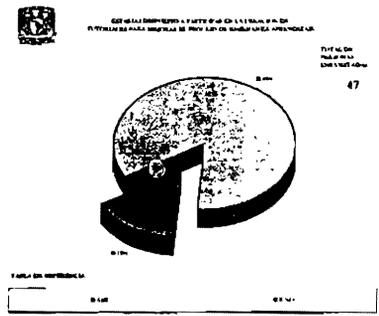
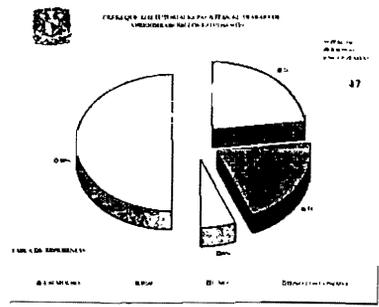
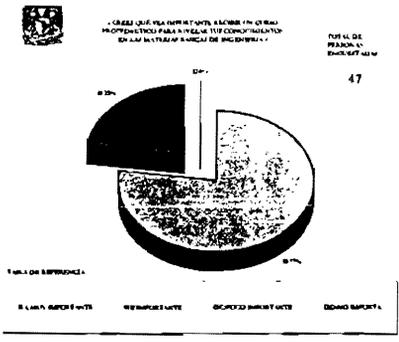
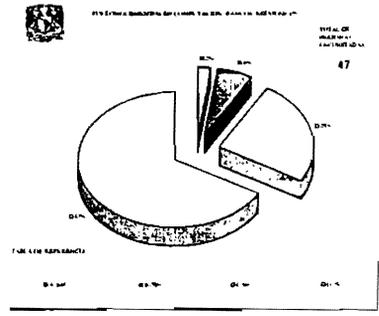
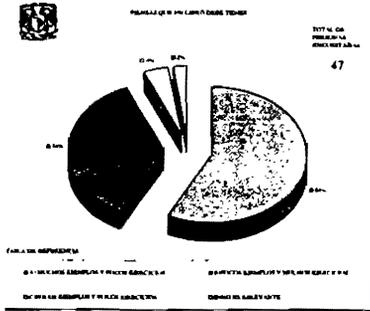
RESULTADOS DE FISICA POR MATERIA



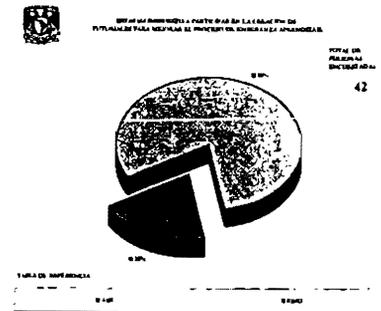
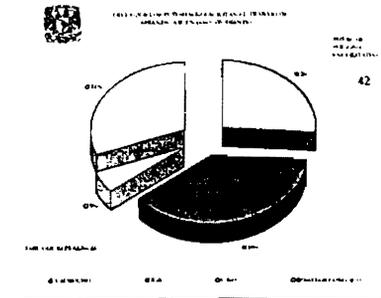
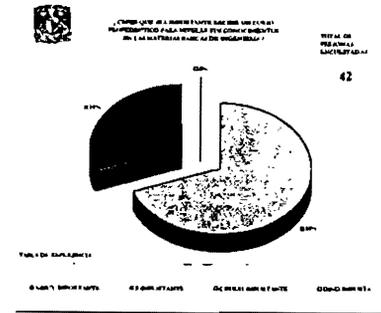
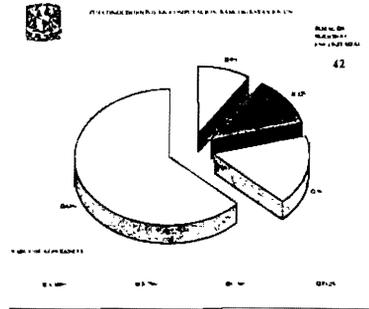
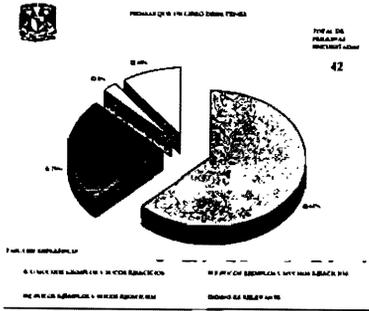
OPINION

Dentro de la evaluación de diagnóstico se anexaron algunos reactivos para conocer cual sería la respuesta e impacto que podría tener el sistema y la respuesta fue la siguiente:

Resultados del grupo 1.



Resultados del grupo 2.



RESULTADOS FINALES.

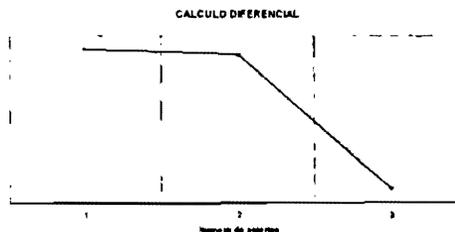
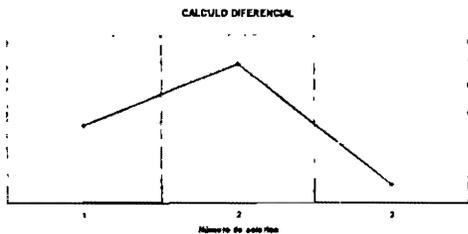
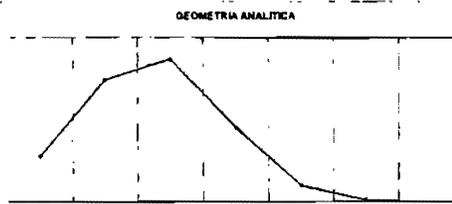
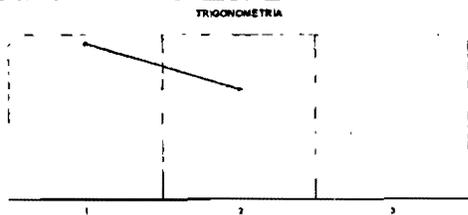
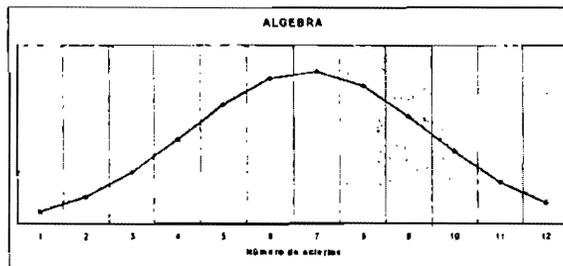
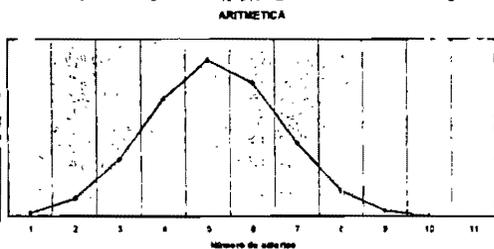
Al terminó de este estudio de campo se obtuvo los siguientes resultados:

ESTADÍSTICAS DE MATEMÁTICAS**RESULTADOS FINALES DE AMBOS GRUPOS DE LA EVALUACION DE MATEMÁTICAS.**

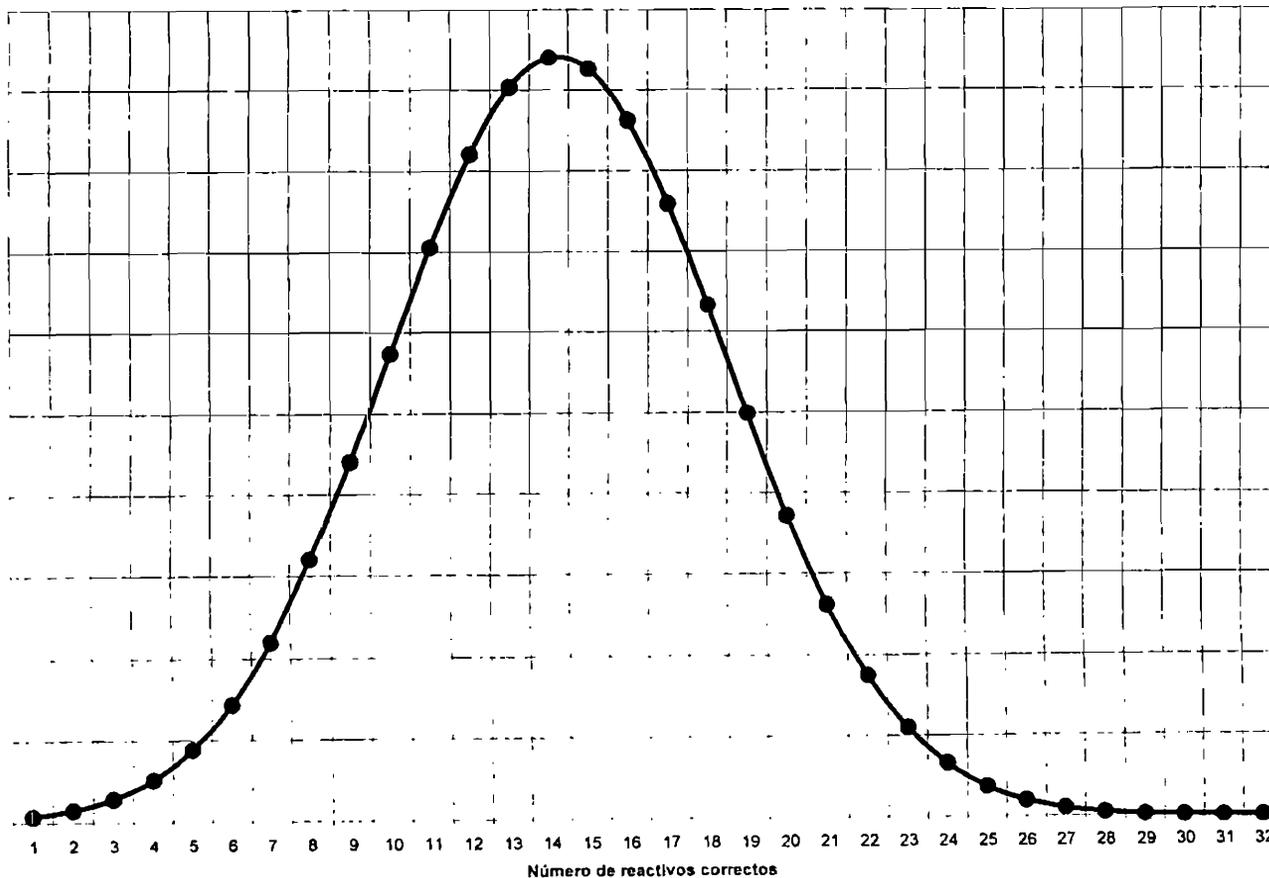
REACTIVO	GRUPO 1	GRUPO 2	SUMA	MEDIA	Mediana	MODA	MIN	MAX	MIN	MAX	MIN	MAX	Desv.	
ALUMNOS	47	42	89				GRUPO1	GRUPO1	GRUPO2	GRUPO2	TOTAL	TOTAL	Estándar.	
ARITMÉTICA														
1	38	37	75	37.5	38						37	38		
2	21	17	38	19	19						17	21		
3	16	9	25	12.5	13						9	16		
4	17	20	37	18.5	19						17	20		
5	4	2	6	3	3						2	4		
6	17	17	34	17	17						17	17		
7	28	30	58	29	29						28	30		
8	21	16	37	18.5	19						16	21		
9	24	32	56	28	28						24	32		
10	6	4	10	5	5						4	6		
PARCIALES	3.9183673	4.3809524	8.2993	4.1496599		4	4	1	7	1	8	1	8	1.507085
ÁLGEBRA														
11	27	22	49	24.5	25						22	27		
12	30	30	60	30	30						30	30		
13	40	35	75	37.5	38						35	40		
14	28	33	61	30.5	31						28	33		
15	17	21	38	19	19						17	21		
16	21	23	44	22	22						21	23		

17	25	19	44	22	22						19	25	
18	32	23	55	27.5	28						23	32	
19	27	30	57	28.5	29						27	30	
20	25	20	45	22.5	23						20	25	
PARCIALES	5.5510204	6.0952381	11.646	5.8231293	6	6	2	10	1	10	1	10	2.591857
TRIGONOMETRIA													
21	14	23	37	18.5	19						14	23	
PARCIALES	0.2857143	0.547619	0.8333	0.4166667	0	0	0	1	0	1	0	1	0.493919
GEOMETRIA ANALITICA													
22	23	27	50	25	25						23	27	
23	9	8	17	8.5	9						8	9	
24	10	9	19	9.5	10						9	10	
25	14	18	32	16	16						14	18	
26	7	9	16	8	8						7	9	
27	5	12	17	8.5	9						5	12	
PARCIALES	1.3877551	2	3.3878	1.6938776	2	1	0	4	0	4	0	4	1.106094
CALCULO DIFERENCIAL											0		
28	30	21	51	25.5	26						21	30	
29	11	5	16	8	8						5	11	
PARCIALES	0.8367347	0.6190476	1.4558	0.7278912	1	1	0	2	0	2	0	2	0.629553
CALCULO INTEGRAL											0		
30	10	9	19	9.5	10						9	10	
31	12	13	25	12.5	13						12	13	
PARCIALES	0.4489796	0.5238095	0.9728	0.4863946	0	0	0	2	0	2	0	2	0.672683
TOTAL DE ACIERTOS	12.44898	14	26.449	13.22449	13	13	1	22	0	23	0	23	4.235469

RESULTADOS POR MATERIA DE MATEMATICAS



RESULTADOS FINALES DE MATEMATICAS



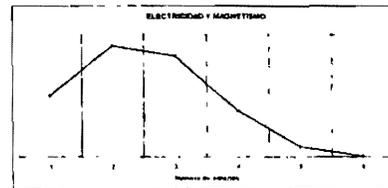
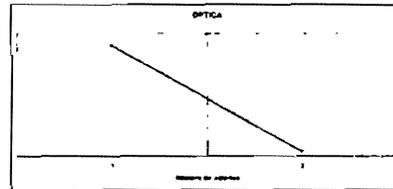
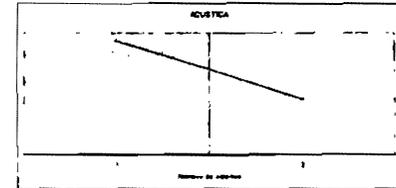
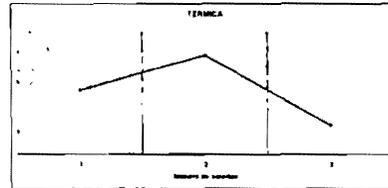
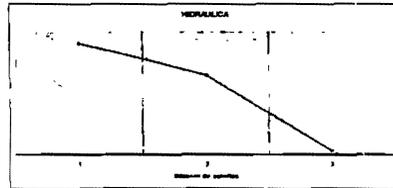
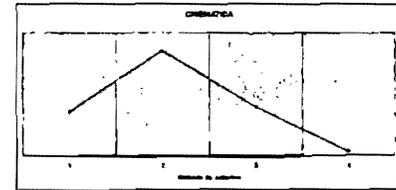
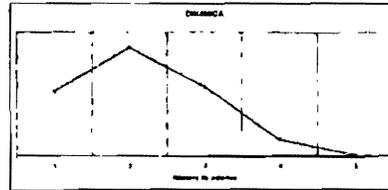
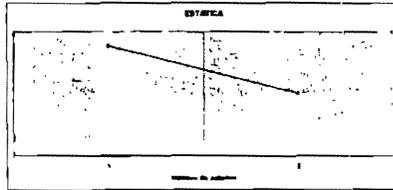
ESTADISTICAS DE FISICA

RESULTADOS FINALES DE AMBOS GRUPOS DE LA EVALUACION DE FISICA.

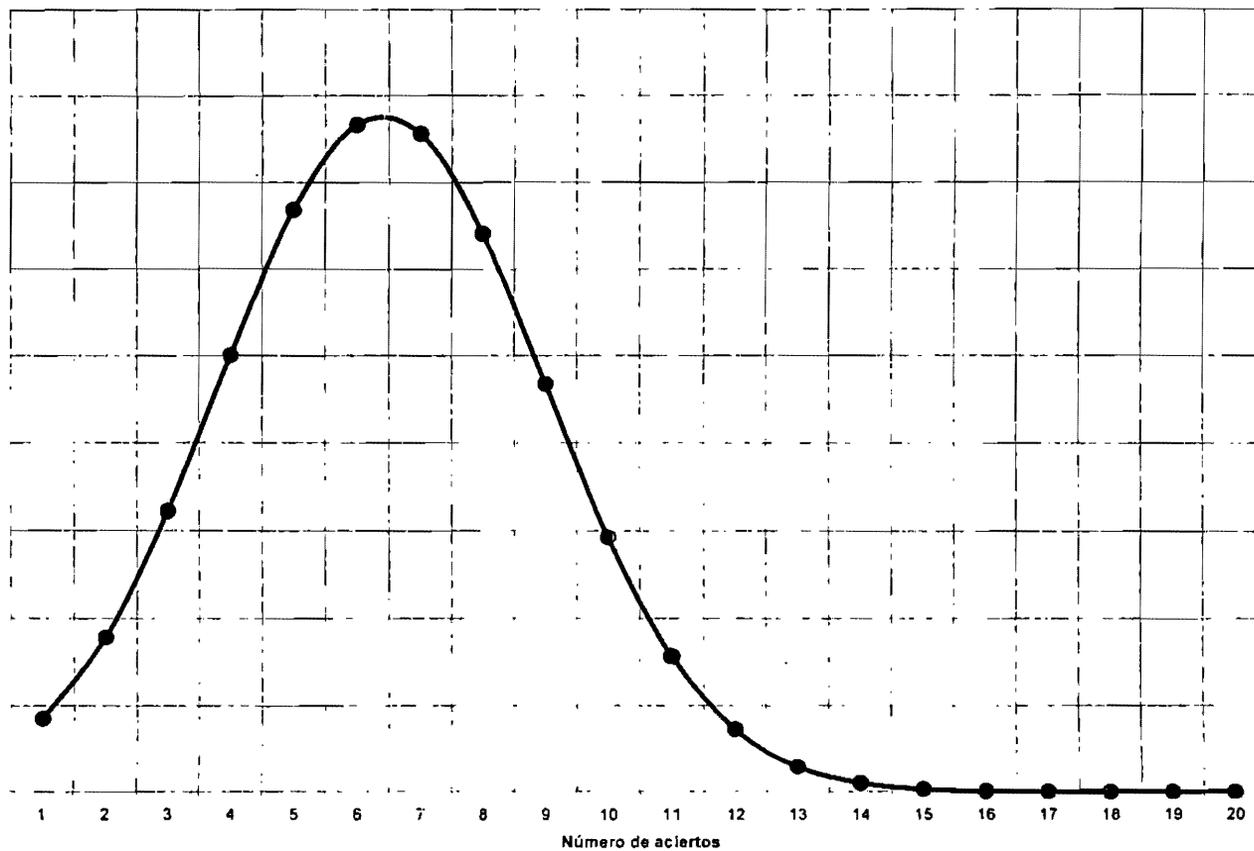
GRUPO	1	2	SUMA	MEDIA	Mediana	MODA	MIN	MAX	MIN	MAX	MIN	MAX	Desv
ALUMNOS	47	42	89				GRUPO 1	GRUPO 2	GRUPO 2	GRUPO2	TOTAL	TOTAL	estándar
ESTATICA													
32	19	14	33	16.5	17						14	19	
PARCIALES	0.4042553	0.3333333	0.7376	0.3687943	0	0	0	1	0	1	0	1	0.485752
DINAMICA													
33	15	12	27	13.5	14						12	15	
34	12	15	27	13.5	14						12	15	
35	16	8	24	12	12						8	16	
36	8	6	14	7	7						6	8	
PARCIALES	1.0851064	0.976905	2.0613	1.0306484	1	0	0	4	0	4	0	4	1.016337
CINEMATICA													
37	16	26	42	21	21						16	26	
38	11	13	24	12	12						11	13	
39	12	13	25	12.5	13						12	13	
PARCIALES	0.8297872	1.2380952	2.0679	1.0339412	1	1	0	3	0	3	0	3	0.783023
HIDRAULICA													
40	12	9	21	10.5	11						9	12	
41	10	12	22	11	11						10	12	

PARCIALES	0.4680851	0.2857143	0.7538	0.376899 7	0	0	0	2	0	2	0	2	0.61247 7
TERMICA											0		
42	21	21	42	21	21						21	21	
43	14	12	26	13	13						12	14	
PARCIALES	0.7446809	0.7857143	1.5304	0.765197 6	1	0	0	2	0	2	0	2	0.78367 5
ACUSTICA											0		
44	15	15	30	15	15						15	15	
PARCIALES	0.3191489	0.3571429	0.6763	0.338145 9	0	0	0	1	0	1	0	1	0.47539 0
OPTICA											0		
45	8	4	12	6	6						4	8	
PARCIALES	0.1702128	0.0952381	0.2655	0.132725 1	0	0	0	1	0	1	0	1	0.34347 9
ELECTRICIDAD											0		
46	24	14	38	19	19						14	24	
47	4	6	10	5	5						4	6	
48	11	9	20	10	10						9	11	
49	11	8	19	9.5	10						8	11	
50	21	14	35	17.5	18						14	21	
PARCIALES	1.5106383	1.2142857	2.7249	1.362462	1	0	0	4	0	4	0	4	1.20964 0
TOTAL DE ACIERTOS	5.5319149	5.2857143	10.818	5.408814 6	6	6	0	11	0	10	0	11	2.57085 5

RESULTADOS POR MATERIA DE FISICA



RESULTADOS FINALES DE FISICA



CONCENTRADO DE REACTIVOS DONDE SE EVALUA EL IMPACTO Y ACPETACION DEL SISTEMA DE SOFTWARE QUE SE DESEA REALIZAR.

NUMERO DE PARTICIPANTES:

89

GRUPO1

47

GRUPO2

42

ELABORO: IGNACIO DE LA ROSA R.

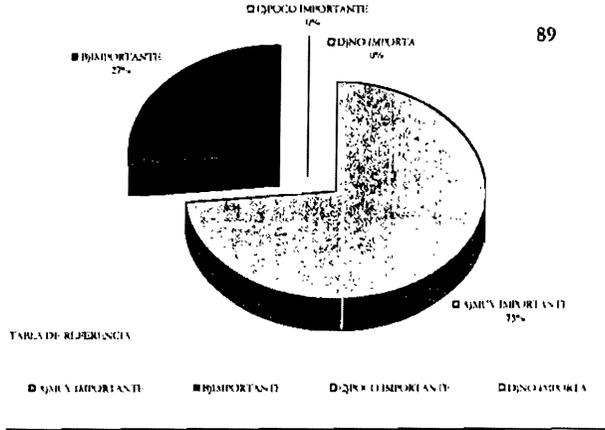
REACTIVO	GRUPO	GRUPO	TOTALES	
51	1			PIENSAS QUE UN LIBRO DEBE TENER
A	27	25	52	A) MUCHOS EJEMPLOS Y POCOS EJERCICIOS
B	17	10	27	B) POCOS EJEMPLOS Y MUCHOS EJERCICIOS
C	2	1	3	C) POCOS EJEMPLOS Y POCOS EJERCICIOS
D	1	4	5	D) NO ES RELEVANTE
52				¿ CREES QUE SEA IMPORTANTE RECIBIR UN CURSO PROPEDEUTICO PARA NIVELAR TUS CONOCIMIENTOS EN LAS MATERIAS BASICAS DE INGENIERIA?
A	37	29	66	A) MUY IMPORTANTE
B	11	13	24	B) IMPORTANTE
C	0	0	0	C) POCO IMPORTANTE
D	0	0	0	D) NO IMPORTA
53				TUS CONOCIMIENTOS EN COMPUTACION (BASICOS) ESTAN EN UN:
A	1	4	5	A) 100%
B	3	5	8	B) 75%
C	12	7	19	C) 50%
D	32	27	59	D) 25%
54				¿ CREES QUE LOS TUTORIALES FACILITAN EL TRABAJO DE APRENDIZAJE EN LOS ESTUDIANTES ?
A	11	11	22	A) SI MUCHO
B	10	16	26	B) SI
C	3	2	5	C) NO
D	24	13	37	D) NO LOS CONOZCO
55				ESTARIAS DISPUESTO A PARTICIPAR EN LA CREACION DE TUTORIALES PARA MEJORAR EL PROCESO DE ENSEÑANZA APRENDIZAJE
A	39	33	72	A) SI
B	7	9	16	B) NO
C	0	0	0	



¿CREES QUE SEA IMPORTANTE RECIBIR UN CURSO PROPEDÉUTICO PARA NIVELAR TUS CONOCIMIENTOS EN LAS MATERIAS BÁSICAS DE INGENIERÍA?

TOTAL DE PERSONAS ENCUESTADAS

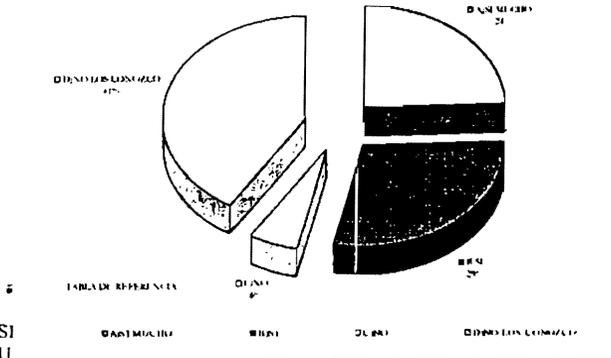
89



¿CÓMO DEBEN SER LAS HORAS DE TRABAJO DE UN INGENIERO EN LOS DÍAS DE LA SEMANA?

TOTAL DE PERSONAS ENCUESTADAS

89



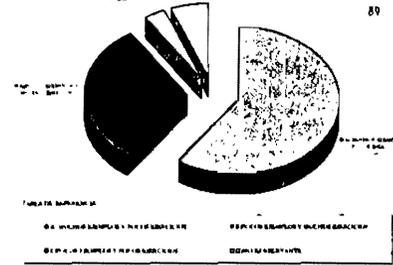
SI
U



PIENSAS QUE UN LIBRO DEBE TENER

TOTAL DE PERSONAS ENCUESTADAS

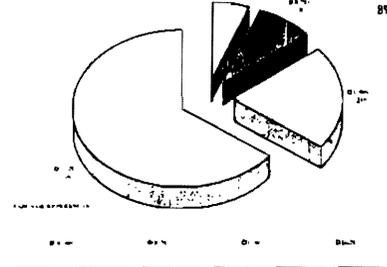
89



¿CÓMO DEBEN SER LAS OPORTUNIDADES DE TRABAJO EN UN INGENIERO?

TOTAL DE PERSONAS ENCUESTADAS

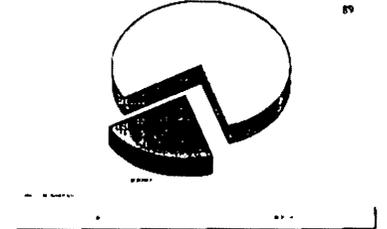
89



¿CÓMO DEBEN SER LAS OPORTUNIDADES DE TRABAJO EN UN INGENIERO?

TOTAL DE PERSONAS ENCUESTADAS

89



Ignacio c

Propuesta de solución

Tomando en cuenta estos resultados, se decidió diseñar un sistema que fuera un apoyo para los estudiantes que se integran a la carrera de ingeniería y que les permitiera emprender el aprendizaje por su cuenta y no solo dejar la responsabilidad de esto al profesor.

Al desarrollar este sistema se plantearon las siguientes premisas:

1. Deberá cubrir los temas de matemáticas y física, necesarios para el estudio de la ingeniería.
2. Permitirá que el alumno practique y ensaye los conocimientos adquiridos.
3. Tendrá el diseño de una "universidad Virtual"
4. Podrá ser utilizado por cualquier estudiante de ingeniería del campus Aragón.
5. Tendrá una forma de evaluar el desempeño y aprendizaje del alumno.
6. Debe ser una herramienta amigable y sencilla de utilizar.

Al desarrollar el software se observó que en un sistema el cual tiene sólo algunos temas de matemáticas y de física y estos con un número determinado de ejercicios, sería poco atractivo, ya que una vez que el alumno logra acceder a dichos temas y realizar los ejercicios, ya no tendría más aplicación, por lo cual se determinó que el sistema debería ser flexible y expansible, con esto me refiero a que debería tener la flexibilidad de poder "crecer" y es en este punto donde entra la participación del facilitador, el cual su tarea principal sería:

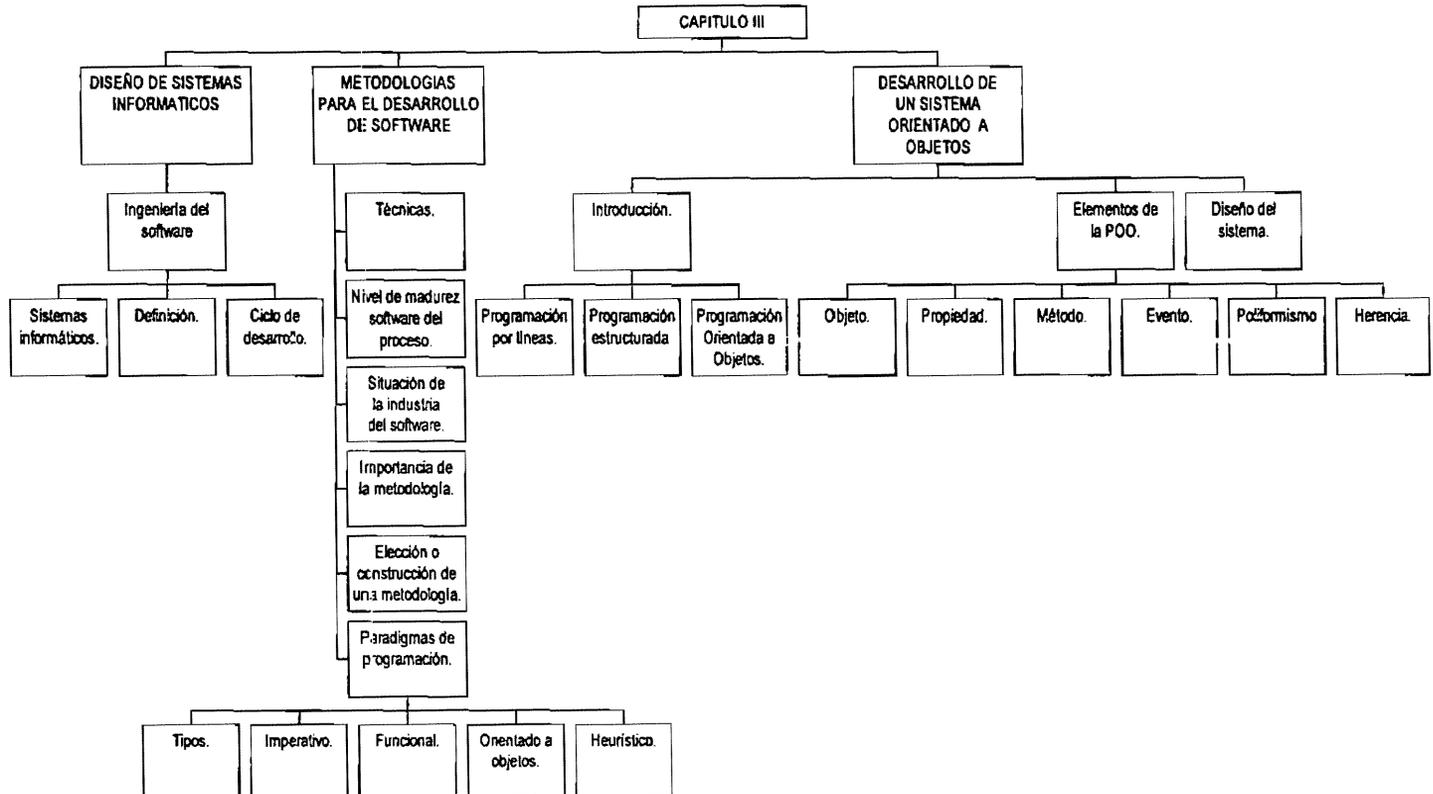
- ♦ Crear y agregar más temas y ejercicios de las materias del sistema
- ♦ Asignar tareas y trabajos a los alumnos.
- ♦ Revisar y valorar el aprendizaje de los alumnos
- ♦ Enriquecer con sus conocimientos el sistema.

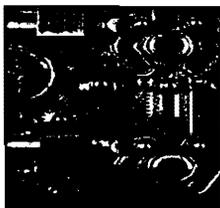
Si bien es cierto que una persona puede aportar sus conocimientos para el desarrollo de una teoría, la participación de un cuerpo docente, dará un potencial al sistema que será una poderosa herramienta para su aplicación y uso. Solo con la conjugación del binomio alumno – facilitador se podrá conseguir resultados excelentes, ya que el sistema por sí solo no garantiza el proceso de enseñanza – aprendizaje, y debe ser una responsabilidad total y compartida por ambas partes en busca de una Universidad acorde al nuevo milenio.

CAPITULO III

DISEÑO DEL SISTEMA

DISEÑO DEL SISTEMA





DISEÑO DE SISTEMAS INFORMATICOS.

Ingeniería del Software

• **Sistemas informáticos.**

Un sistema informático utiliza computadoras para almacenar los datos de una organización y ponerlos a disposición de quien se considere oportuno. Un sistema puede ser tan simple como el indicado seguidamente. Una persona tiene un microordenador y le introduce datos, tan elementales, como por ejemplo las ventas diarias de una pequeña empresa. Se produce una entrada por cada venta. En la entrada

se declara el elemento vendido, por ejemplo un yogur con frutas del bosque, la cantidad de elementos vendidos, por ejemplo cuatro y el precio de venta unitario, por ejemplo \$ 5.00 . Cada entrada se almacena como un registro de un archivo en el disco del microordenador. Al finalizar el día se puede generar un informe de las ventas totales y las tendencias de las mismas. El usuario puede utilizar esta información para decidir qué elementos debe tener en el almacén o para planificar campañas publicitarias.

Sin embargo la mayor parte de los sistemas son más complejos. Normalmente una empresa tiene más de un ordenador, por ejemplo uno para la gestión de ventas y otro para la contabilidad y procesos asociados. Los sistemas de información tienen muchas cosas en común. La mayoría de ellos están formados por personas, equipos y procedimientos.

Las personas son un componente esencial en cualquier sistema de información, producen y utilizan la información de sus actividades diarias para decidir lo que se debe hacer. Las decisiones pueden ser rutinarias o complejas.

Los sistemas de información deben soportar diversas clases de actividades del usuario, por eso deben de establecerse procedimientos que aseguren que los datos correctos llegan a las personas adecuadas en su momento.

Por último está el equipo, es decir las computadoras y todos los dispositivos necesarios.

• Ingeniería del Software

El entorno actual de desarrollo de sistemas software viene adoleciendo de:

- retrasos considerables en la planificación
- poca productividad
- elevadas cargas de mantenimiento
- demandas cada vez más desfasadas con las ofertas
- baja calidad y fiabilidad del producto
- Dependencia de los realizadores esto es lo que se ha denominado comúnmente como "crisis del software".

Ello ha venido originado por una falta de:

- formalismo y metodología
- herramientas de soporte
- administración eficaz

Actualmente está surgiendo una gran expectativa ante la evolución de la Ingeniería del Software, al ir apareciendo nuevos métodos y herramientas formales que van a permitir en el futuro un planteamiento de ingeniería en el proceso de elaboración de software. Dicho planteamiento vendrá a moderar la demanda creciente por parte de los usuarios, permitiendo dar respuesta a los problemas de:

- administración
- calidad
- productividad
- fácil mantenimiento

Este último es uno de los grandes problemas, pues puede llegar a suponer un coste superior al 60% del total del coste del software.

Las nuevas metodologías suponen un enfoque integral del problema, abarcando todas las fases, que en su mayoría no se consideraban en los desarrollos tradicionales. En particular son fundamentales la reducción de costes y plazos, así como la calidad del producto final.

Estas tecnologías constituyen la denominada "Ingeniería del Software", que se puede definir como el **tratamiento sistemático de todas las fases del ciclo de vida del software**. Hay otras definiciones, pero todas inciden en la importancia de una disciplina de ingeniería para el desarrollo de software.

• Ciclo de desarrollo

La formalización del proceso de desarrollo se define como un marco de referencia denominado ciclo de desarrollo del software o ciclo de vida del desarrollo del software o **ciclo de vida del desarrollo** "El período de tiempo que comienza con la decisión de desarrollar un producto software y finaliza cuando se ha entregado éste". Este ciclo, por lo general incluye, una fase de requisitos, fase de diseño, fase de implantación, fase de prueba, y a veces, fase de instalación y aceptación.

El ciclo de desarrollo software se utiliza para estructurar las actividades que se llevan a cabo en el desarrollo de un producto software. A pesar de que no hay acuerdo acerca del uso y la forma del modelo, este sigue siendo útil para la comprensión y el control del proceso de desarrollo. Seguidamente se exponen las distintas aproximaciones de desarrollo.

Tipos de ciclos de vida

Aproximación convencional

Se introdujo como una técnica rígida para mejorar la calidad y reducir los costos del software. Tradicionalmente es conocido como "modelo en cascada", porque su filosofía es completar un paso antes de empezar el siguiente, cada paso con un alto grado de exactitud. Los principales problemas que se han detectado en esta aproximación son debidos a que se comienza estableciendo todos los requisitos del sistema:

En muchas ocasiones no es posible disponer de unas especificaciones correctas desde el primer momento, porque puede ser difícil para el usuario establecer al inicio todos los requisitos. En otras hay cambio de parecer de los usuarios sobre las necesidades reales cuando ya se ha comenzado el proyecto, siendo probables los verdaderos requisitos no se reflejen en el producto final.

Otro de los problemas de esta aproximación es que los resultados no se ven hasta muy avanzado el proyecto, por lo tanto la realización de modificaciones, si ha habido un error, es muy costosa.

Esta aproximación es la más empleada por los ingenieros informáticos.

Aproximación prototipo

Esta aproximación consiste en realizar la fase de definición de requisitos del sistema caracterizada por estos tres factores:

- Un alto grado de iteración
- Un muy alto grado de participación del usuario
- Un uso extensivo de prototipos

Las premisas clave de esta aproximación son:

- 1 - Que los prototipos constituyen un medio mejor de comunicación que los modelos en papel
- 2- Que la iteración es necesaria para canalizar, en la dirección correcta, el proceso de aprendizaje. Esta aproximación se enfoca a mejorar la efectividad del proceso de desarrollo y no a mejorar la eficacia de ese proceso.

Aproximación evolutiva

En esta aproximación el énfasis está en lograr un sistema flexible y que se pueda expandir de forma que se pueda realizar muy rápidamente una versión modificada del sistema cuando los requisitos cambien. Se diferencia de la aproximación anterior, en que en esta los requisitos cambian continuamente, lo cual implicaría en el caso previo que las iteraciones no tendrían fin.

Aproximación incremental

Es un concepto muy parecido al de desarrollo evolutivo, y frecuentemente comprendido en la aproximación del desarrollo evolutivo. Se comienza el desarrollo del sistema para satisfacer un subconjunto de requisitos especificados. Las últimas versiones prevén los requisitos que faltan. De esta forma se logra una rápida disponibilidad del sistema, que aunque incompleto, es utilizable y satisface algunas de las necesidades básicas de información. La diferencia con la aproximación anterior es que en este caso cada versión parte de una previa sin cambios pero con nuevas funciones, mientras que la aproximación evolutiva cada vez se desarrolla una nueva versión de todo el sistema

Aproximación espiral

Esta nace con el objetivo de captar lo mejor de la aproximación convencional y de la de prototipo, añadiendo un nuevo componente, el análisis de riesgo.

Esquemáticamente se puede ilustrar mediante una espiral, con cuatro cuadrantes que definen actividades. En la primera vuelta de la espiral se definen los objetivos, las alternativas y las restricciones y se analizan y se identifican los riesgos. Si como consecuencia del análisis de riesgo se observa que hay incertidumbre sobre el problema entonces en la actividad correspondiente a la ingeniería se aplicará la aproximación prototipo cuyo beneficio principal es el de reducir la incertidumbre de la naturaleza del problema de información y los requerimientos que los usuarios establecen para la solución a ese problema. Al final de esta primera vuelta alrededor de la espiral el usuario evalúa los productos obtenidos y puede sugerir modificaciones. Se comenzaría avanzando alrededor del camino de la espiral realizando las cuatro actividades indicadas a continuación. En cada vuelta de la espiral, la actividad de ingeniería ser desarrollada mediante la aproximación convencional o ciclo de desarrollo en cascada o mediante la aproximación de prototipos.

Actividades	Acciones
Planificación	Determinación de alternativas, identificación y resolución de riesgos
Ingeniería	Desarrollo y verificación del producto de siguiente nivel
Evaluación del cliente	Valoración de los resultados del proceso de desarrollo

Aproximación basada en transformaciones

Con la aparición de las herramientas CASE junto con los generadores de código, el ciclo de desarrollo software en cascada ha cambiado a un ciclo de vida basado en transformaciones

La utilización de herramientas CASE afecta a todas las fases del ciclo de vida del software. Este ciclo de vida se puede considerar como una serie de transformaciones. Primero se definen los requisitos del sistema, seguidamente existe un proceso de transformación que hace que la especificación se convierta en un diseño lógico del sistema. Posteriormente, este sufre otro proceso de transformación para lograr un diseño físico, es decir que responda a la tecnología destino.

La tecnología CASE propone que estos procesos de transformación sean lo más automatizables posible. Sus ventajas son:

- Posibilidad de comprobación de errores en etapas iniciales de desarrollo
- Posibilidad de realizar el mantenimiento en el ámbito de especificación
- Soporte de rastreabilidad de los requisitos
- Soporte de reusabilidad
- Potencia la especificación orientada al problema



METODOLOGIAS DE DESARROLLO DE SOFTWARE

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software. Por lo tanto es como un libro de recetas de cocina, en el que se van indicando paso a paso todas las actividades a realizar para lograr el producto software deseado, indicando además qué personas deben participar en el desarrollo de las actividades y qué papel deben de hacer en las mismas. Además detallan la información que se debe producir como resultado de una actividad y la información necesaria para comenzar la actividad.

• Técnicas

Las técnicas indican cómo debe ser realizada una actividad técnica determinada identificada en la metodología. Combina el empleo de unos modelos o representaciones gráficas junto con el empleo de unos procedimientos detallados. Se debe tener en consideración que una técnica determinada puede ser utilizada en una o más actividades de la metodología de desarrollo de software. Además se debe tener mucho cuidado cuando se quiere cambiar una técnica por otra debido a varios motivos.

• Nivel de madurez software del proceso

El Instituto de Ingeniería del Software (SEI) de la Universidad Carnegie Mellon de Pittsburgh ha diseñado un método para evaluar el nivel de madurez del proceso de desarrollo del software de una empresa u organismo. El proceso se evalúa mediante un cuestionario y las respuestas se emplean para determinar una magnitud denominada "Nivel de Madurez del Proceso"

El valor obtenido es un indicador de toda la empresa, aunque pueda darse el caso de que en algún departamento de la empresa tenga un nivel de madurez mayor o inferior al resultante. Los niveles de madurez del proceso son cinco

1.- Inicial: La empresa no dispone de procesos y controles definidos

Se trabaja con procedimientos que no están normalizados, es decir, procedimientos tanto del propio desarrollo de software como de su planificación y control, que no están establecidos explícitamente antes de su uso.

Por otro lado las técnicas y/o herramientas que se emplean para el desarrollo del software carecen de una integración entre las mismas y únicamente son empleadas en algunas fases del ciclo de vida del software.

La característica de las empresas que están en este nivel es que no hay un control de la gestión de proyectos software efectivo, porque puede suceder que la empresa disponga de procedimientos y técnicas formales, tanto de gestión como del proceso, y de herramientas, pero no se utilizan de manera estándar en todos los proyectos

2.- Repetible. La empresa tiene métodos estandarizados facilitando procesos repetibles

Las empresas que se encuentran en este nivel son las que disponen de un control básico de la gestión de proyectos, gestión de calidad y gestión de la configuración.

El problema en este tipo de organización es que introducir cualquier cambio tiene un alto grado de riesgo de fracaso.

3.- Procesos definidos. La empresa monitoriza y mejora sus procesos.

Las empresas que se encuentran en este nivel se caracterizan por disponer de:

- Un grupo de proceso, cuyo objetivo es el de mejorar el proceso software
- Una metodología de desarrollo software que describa las actividades técnicas y de gestión requeridas para la adecuada ejecución del proceso de desarrollo.

4.- Proceso gestionado. La empresa posee controles avanzados, métricas y retroalimentación

Las empresas que han alcanzado este nivel disponen de un control de los costes y calidad de las principales etapas del proceso. Es pre- requisito que exista una metodología de desarrollo software para realizar una medición efectiva.

5. - Proceso de optimización. La optimización emplea métricas con propósitos de optimización.

En este nivel, la empresa tiene los medios para identificar los elementos más débiles del proceso y mejorarlos.

El Modelo de Madurez ha resultado ser especialmente útil para mostrar el estado y las áreas de problemas claves en numerosas empresas de software. Una evaluación de este tipo puede ayudar a:

- las empresas a identificar su nivel de madurez específico
- la dirección de la empresa estableciendo una estructura para implementar las acciones de mejoras prioritarias.

Una vez identificado el nivel de madurez de la empresa, ésta puede concentrarse en aquellos elementos que pueden ayudarte a avanzar hacia el siguiente nivel.

• Situación de la industria del software

A través de los informes anuales del Ministerio de Industria sobre el estado del sector informático español, es fácil apreciar un claro crecimiento en software, servicios y consumibles, frente a un decremento registrado en el hardware, según se muestra en la siguiente tabla.

Concepto	1990	1991	%
Hardware	492985	481400	-2.3
Software	76343	85542	12
Servicios	199752	230134	15.2
Consumibles	17132	19867	15.9
TOT AL	786108	816943	3.9

De la que se deduce una clara orientación de la informática española hacia los servicios y el software. Por otra parte los crecimientos, a partir del año 1991, han sido bajos con relación a los años previos que eran porcentajes de dos dígitos. Aunque parte de estos resultados pueden considerarse como efecto de la crisis económica.

En México al igual que en España se carece de datos fiables sobre la situación real de la industria software, ya que no se han realizado evaluaciones objetivas.

En EE.UU. las evaluaciones efectuadas en importantes empresas, dan como resultado que el 81 % de las empresas se situó en el nivel 1, el 12% en el 2, el 7% en el 3 y ninguna en los niveles 4 ó 5. En Europa y Japón los porcentajes son parecidos.

Por todo ello, se puede suponer que las empresas de software en México se encuentran en los niveles 1 y 2 del Modelo de Madurez. Es decir empresas que tienen definidos procedimientos formales para la planificación, seguimiento, desarrollo, etc., pero que no disponen de un mecanismo de dirección que se asegure que esos procedimientos formales están siendo utilizados. Además cuando no hay tiempo suficiente, estos procedimientos son abandonados y tan solo se realizan tareas de codificación y prueba.

Las herramientas no se han integrado suficientemente bien en el proceso, ni aplicadas uniformemente. A menudo, el mantenimiento presenta serios problemas, ocasionando así un colapso en el desarrollo de nuevas aplicaciones. El porcentaje de recursos dedicados a mantenimiento es aproximadamente el 65%, por lo que es muy difícil responder a las demandas de nuevas aplicaciones software de la empresa.

• Importancia de la metodología

Hay un gran número de factores que repercuten en la persona que trabaja dentro de un entorno de desarrollo software. Los cambios en el sistema operativo, el lenguaje de programación, la organización del proyecto, o los estándares establecidos para los diferentes aspectos del ciclo de vida de un proyecto pueden repercutir tanto en el trabajador como en la cantidad de trabajo que puede realizar

La productividad, cómo una medida cuantitativa de la cantidad de trabajo que puede ser realizada por una persona, se puede alterar de distintas maneras, alguna de ellas tan simple como, por ejemplo, enseñar a todos los implicados en el trabajo a escribir a máquina. Este hecho, sin ir más lejos, podría tener un mayor impacto en la productividad que el de introducir unas nuevas herramientas software o técnicas de diseño.

Sin embargo la productividad no tiene en consideración la calidad del producto. Por ejemplo los trabajadores en una planta de ensamblaje de ordenadores pueden producir 100 ordenadores por hora, pero esta medida no es útil, en cuanto que los ordenadores pueden requerir trabajo adicional para *corregir* problemas surgidos en la etapa del ensamblaje. Lo mismo ocurre en el desarrollo de software, el objetivo es establecer un entorno que no sólo mejore la productividad del que lo desarrolla, sino que también genere la creación de mejores productos.

Un ejemplo de diferentes metodología se puede observar en la fabricación de coches. Volvo adoptó una organización de equipo para la producción de coches, así como para mejorar la satisfacción del trabajador: averiguaron que sus trabajadores podían trabajar conjuntamente e una manera efectiva, compartir responsabilidades y variar tareas. BMW (Bayerische Motor Werke) averiguó que tal organización no se podía aplicar a su entorno, dada la dificultad de comunicación entre sus trabajadores. Tanto Volvo como BMW producen coches de las mismas características. Con la misma calidad y precios, pero diferencias en su personal les hacen emplear diferentes metodologías.

Es obvio que el elemento más importante en cualquier empresa de desarrollo de software es disponer de personas con una elevada calificación, y sin embargo ello no asegura el éxito en la consecución de los objetivos propuestos, ya que existe el peligro de una falta de conjunción, producida por la manera personal de desarrollar el software de cada individuo, por muy bueno que este sea, y la imposibilidad de un auténtico trabajo en equipo.

El Modelo de Madurez, viene a indicarnos que los mejores informáticos necesitan un entorno disciplinado y estructurado en el cual puedan realizar un trabajo en equipo, para lograr unos productos con alta calidad.

El ingeniero de software es una persona que trabaja en equipo, que conoce que lo que él realiza es un componente que combinar con otros para formar un sistema. Es consciente de que el componente software que diseña debe poseer los principios de la Ingeniería del Software para que el sistema final sea satisfactorio.

Los programadores tradicionales argumentan que la aplicación de una metodología supone una gran carga. Es cierto, pero si no se emplea una metodología pueden surgir los siguientes problemas.

- Resultados impredecibles
- Detección tardía de errores
- La introducción de nuevas herramientas afectará perjudicialmente al proceso
- Cambios de organización también afectarán al proceso
- Resultados distintos con nuevas clases de productos

La situación actual se debe ver como una situación en la que la empresa que comience a poner los elementos necesarios para mejorar el proceso software tendrá mucha más ventaja competitiva frente a las demás.

Elección o construcción de una metodología

Crterios para evaluar una metodología

En el momento de adoptar un estándar o construir una metodología, se han de considerar unos requisitos deseables, por lo que seguidamente se proponen una serie de criterios de evaluación de dichos requisitos.

La metodología debe ajustarse a los objetivos

Cada aproximación al desarrollo de software está basada en unos objetivos a lograr. Por ello la metodología que se elija debe recoger el aspecto filosófico de la aproximación deseada, es decir que los objetivos generales del desarrollo deben estar implementados en la metodología de desarrollo.

La metodología debe cubrir el ciclo entero de desarrollo de software

Para ello la metodología ha de realizar unas etapas:

- Investigación
- Análisis de requisitos
- Diseño

La metodología debe integrar las distintas fases del ciclo de desarrollo

- **Rastreabilidad.** Es importante poder referirse a otras fases de un proyecto y fusionarlo con las fases previas. Es importante poder moverse no sólo hacia adelante en el ciclo de vida, sino hacia atrás de forma que se pueda comprobar el trabajo realizado y se puedan efectuar correcciones.
- **Fácil interacción entre etapas del ciclo de desarrollo.** Es necesaria una validación formal de cada fase antes de pasar a la siguiente. La información que se pierde en una fase determinada queda perdida para siempre, con un impacto en el sistema resultante.

La metodología debe incluir la realización de validaciones

La metodología debe detectar y corregir los errores cuanto antes. Uno de los problemas más frecuentes y costosos es el aplazamiento de la detección y corrección de problemas en las etapas finales del proyecto. Cuanto más tarde sea detectado el error más caro será corregirlo.

Por lo tanto cada fase del proceso de desarrollo de software deberá incluir una actividad de validación explícita.

La metodología debe soportar la determinación de la exactitud del sistema a través del ciclo de desarrollo.

La exactitud del sistema implica muchos asuntos, incluyendo la correspondencia entre el sistema y sus especificaciones, así como que el sistema cumple con las necesidades del usuario. Por ejemplo, los métodos usados para análisis y especificación del sistema deberían colaborar a terminar con el problema del entendimiento entre los informáticos, los usuarios, y otras partes implicadas.

Esto implica una comunicación entre usuario y técnico amigable y sencilla, exenta de consideraciones técnicas.

La metodología debe ser la base de una comunicación efectiva.

Debe ser posible gestionar a los informáticos, y éstos deben ser capaces de trabajar conjuntamente. Ha de haber una comunicación efectiva entre analistas, programadores, usuarios y gestores, con pasos bien definidos para realizar progresos visibles durante la actividad del desarrollo.

La metodología debe funcionar en un entorno dinámico orientado al usuario

A lo largo de todo el ciclo de vida del desarrollo se debe producir una transferencia de conocimientos hacia el usuario. La clave del éxito es que todas las partes implicadas han de intercambiar información libremente. La participación del usuario es de importancia vital debido a que sus necesidades evolucionan constantemente. Por otra parte la adquisición de conocimientos del usuario la permitirá la toma de decisiones correctas.

Para involucrar al usuario en el análisis, diseño y administración de datos, es aconsejable el empleo de técnicas estructuradas lo más sencillas posible. Para esto, es esencial contar una buena técnica de diagramación.

La metodología debe especificar claramente los responsables de resultados

Debe especificar claramente quienes son los participantes de cada tarea a desarrollar, debe detallar de una manera clara los resultados de los que serán responsables.

La metodología debe poder emplearse en un entorno amplio de proyectos software

- **Variedad.** Una empresa deberá adoptar una metodología que sea útil para un número grande de sistemas que vaya a construir. Por esta razón no es práctico adoptar varias metodologías en una misma empresa.
- **Tamaño, vida.** Las metodologías deberán ser capaces de abordar sistemas de distintos tamaños y rangos de vida.
- **Complejidad.** La metodología debe servir para sistemas de distinta complejidad, es decir puede abarcar un departamento, varios de departamentos o varias empresas.
- **Entorno.** La metodología debe servir con independencia de la tecnología disponible en la empresa.

La metodología se debe de poder enseñar

Incluso en una organización sencilla, serán muchas las personas que la van a utilizar, incluso los que se incorporen posteriormente a la empresa. Cada persona debe entender las técnicas específicas de la metodología, los procedimientos organizativos y de gestión que la hacen efectiva, las herramientas automatizadas que soportan la metodología y las motivaciones que subyacen en ella.

La metodología debe estar soportada por herramientas CASE

La metodología debe estar soportada por herramientas automatizadas que mejoren la productividad, tanto del ingeniero de software en particular, como la del desarrollo en general.

El uso de estas herramientas reduce el número de personas requeridas y la sobrecarga de comunicación, además de ayudar a producir especificaciones y diseños con menos errores, más fáciles de probar, modificar y usar.

La metodología debe soportar la eventual evolución del sistema

Normalmente durante su tiempo de vida los sistemas tienen muchas versiones, pudiendo durar incluso más de 10 años. Existen herramientas CASE para la gestión de la configuración y otras denominadas "Reverse Engineering" para ayudar en el mantenimiento de los sistemas no estructurados, permitiendo estructura los componentes de éstos facilitando así su mantenimiento.

La metodología debe contener actividades conducentes a mejorar el proceso de desarrollo de software.

Para mejorar el proceso es básico disponer de datos numéricos que evidencian la efectividad de la aplicación del proceso con respecto a cualquier producto software resultante del proceso. Para disponer de estos datos, la metodología debe contener un conjunto de mediciones de proceso para identificar la calidad y coste asociado a cada etapa del proceso. Sería ideal el uso de herramientas CASE.

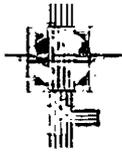
• Criterios más importantes

Se han realizado pocos estudios para determinar qué factores son los más importantes en el momento de seleccionar una metodología de desarrollo de software. El más reciente es el realizado por Sachidanandam Sakthivec, donde la conclusión obtenida, es que no todos los requisitos son iguales de importantes para los profesionales de desarrollo de software.

Esta investigación se basó en requisitos de la metodología. Posteriormente mediante un programa que utiliza algoritmos computacionales AHP (Analytic Hierarchy Process), identificó la importancia relativa de los requisitos de la metodología para cada participante en la encuesta

Las conclusiones de la investigación son:

- La capacidad de una metodología para desarrollar sistemas con la calidad requerida es el requisito más importante para los profesionales del desarrollo.
- Coincidencia con los investigadores en la importancia que tiene el criterio de satisfacción del usuario en la calidad del nuevo sistema
- La capacidad para desarrollar gran variedad de sistemas tiene mayor importancia que los aspectos relacionados con la productividad.
- Los profesionales prefieren como segundo requisito metodología soporte todas las etapas del desarrollo.



PARADIGMAS DE PROGRAMACION

Un paradigma de programación es un modelo básico de diseño y desarrollo de programas. Un modelo que permite producir programas con unas directrices específicas, tales como: estructura modular, fuerte cohesión, alta rentabilidad, etc.

Para algunos puede resultar sorprendente que existan varios paradigmas de programación. La mayor parte de los programadores están familiarizados con un único paradigma, el de la programación procedimental. Sin embargo existen multitud de ellos atendiendo a alguna particularidad metodológica o funcional, como por ejemplo el basado en reglas de gran aplicación en la ingeniería del conocimiento para el desarrollo de sistemas expertos, en que el núcleo del mismo son las reglas de producción del tipo "if then"; el de programación lógica, basado en asertos y reglas lógicas que define un entorno de programación de tipo conversacional, deductivo, simbólico y no determinista; el de programación funcional, basado en funciones, forma funcionales para crear funciones y mecanismos para aplicar los argumentos, y que define un entorno de programación interpretativo, funcional y aplicativo, el de programación heurística que aplica para la resolución de los problemas "reglas de buena lógica" que presentan reflejos de ser correctas aunque no se garantiza su éxito, modelando el problema de una forma adecuada para aplicar estas heurísticas atendiendo a su representación, estrategias de búsqueda y métodos de resolución; el de programación paralela; el basado en restricciones; el basado en el flujo de datos; el orientado al objeto, etc

Un paradigma de programación es una colección de modelos conceptuales que juntos modelan el proceso de diseño y determinan, al final, la estructura de un programa.

Esa estructura conceptual de modelos está pensada de forma que esos modelos determinan la forma correcta de los programas y controlan el modo en que pensamos y formulamos soluciones, y al llegar a la solución, se debe de expresar mediante un lenguaje de programación. Para que este proceso sea efectivo las características del lenguaje deben reflejar adecuadamente los modelos conceptuales de ese paradigma.

Cuando un lenguaje refleja bien un paradigma particular, se dice que soporta el paradigma, y en la práctica un lenguaje que soporta correctamente un paradigma, es difícil distinguirlo del propio paradigma, por lo que se identifica con él.

• Tipos de paradigmas de programación

Estudios han descrito tres categorías de paradigmas de programación:

- a) Los que soportan técnicas de programación de bajo nivel (Ej.: copia de ficheros frente estructuras de datos compartidos)
- b) Los que soportan métodos de diseño de algoritmos (Ej.: divide y vencerás, programación dinámica, etc.)
- c) Los que soportan soluciones de programación de alto nivel, como los descritos en el punto anterior

Estos estudios también señala lo diferentes que resultan los lenguajes de programación que soportan cada una de estas categorías de paradigmas. Sólo se comentarán los paradigmas que soportan la programación de alto nivel.

Se agrupan en tres categorías de acuerdo con la solución que aportan para resolver el problema

- a) Solución procedimental u operacional. Describe etapa a etapa el modo de construir la solución. Es decir señala cómo obtener la solución.
- b) Solución demostrativa. Es una variante de la procedimental. Especifica la solución describiendo ejemplos y permitiendo que el sistema generalice la solución de estos ejemplos para otros casos. Aunque es fundamentalmente procedimental, el hecho de producir resultados muy diferentes a ésta, hace que sea tratada como una categoría separada.
- c) Solución declarativa. Señala las características que debe tener la solución, sin describir cómo procesarla. Es decir señala qué se desea obtener pero no cómo obtenerlo.

Paradigmas procedimentales u operacionales

La característica fundamental de estos paradigmas es la secuencia computacional realizada etapa a etapa para resolver el problema. Su mayor dificultad reside en determinar si el valor computado es una solución correcta del problema, por lo que se han desarrollado multitud de técnicas de depuración y verificación para probar la corrección de los problemas desarrollados basándose en este tipo de paradigmas.

Pueden ser de dos tipos básicos: Los que actúan modificando repetidamente la representación de sus datos (efecto de lado); y los que actúan creando nuevos datos continuamente (sin efecto de lado).

Los paradigmas con efecto de lado utilizan un modelo en el que las variables están estrechamente relacionadas con direcciones de la memoria del ordenador. Cuando se ejecuta el programa, el contenido de estas direcciones se actualiza repetidamente, pues las variables reciben múltiples asignaciones, y al finalizar el trabajo, los valores finales de las variables representan el resultado.

Existen dos tipos de paradigmas con efectos de lado:

- el imperativo
- el orientado a objetos

Los paradigmas sin efecto de lado no incluyen a los que tradicionalmente son denominados paradigmas funcionales. Sin embargo es importante distinguir la solución funcional procedimental de la solución funcional declarativa.

Los paradigmas procedimentales definen la secuencia explícitamente, pero esta secuencia se puede procesar en serie o en paralelo. En este segundo caso el procesamiento paralelo puede ser asíncrono (cooperación de procesos paralelos) o síncrono (procesos simples aplicados simultáneamente a muchos objetos).

Paradigmas declarativos

En este tipo, un programa se construye señalando hechos, reglas, restricciones, ecuaciones, transformaciones y otras propiedades derivadas del conjunto de valores que configuran la solución.

A partir de esta información el sistema debe proporcionar un esquema que incluya el orden de evaluación que compute una solución. Aquí no existe la descripción de las diferentes etapas a seguir para alcanzar una solución, como en el caso anterior.

Estos paradigmas permiten el uso de variables para almacenar valores intermedios, pero no para actualizar estados de información.

Dado que estos paradigmas especifican la solución sin indicar cómo construirla, en principio eliminan la necesidad de probar que el valor calculado es el valor solución. En la práctica, mientras que muchos de los paradigmas secuencia de control y efecto de lado que requiera la noción de estado, las soluciones son todavía producidas como construcciones más bien que como especificaciones. Por lo que los paradigmas resultantes y los lenguajes que los soportan no son verdaderamente declarativos, sino pseudodeclarativos.

En este grupo se encuentran: el funcional, el lógico y el de transformación.

En principio, los paradigmas declarativos no son soluciones inherentes de tipos serie o paralelo, ya que no dirigen la secuencia de control y no pueden alterar el natural no-paralelismo del algoritmo. No obstante, los paradigmas pseudodeclarativos requieren al menos un limitado grado de secuencia, y por lo tanto admiten versiones en serie y paralelo.

Paradigmas demostrativos

Cuando se programa bajo un paradigma demostrativo (también llamada programación por ejemplos), el programador no especifica procedimentalmente cómo construir una solución. En su lugar, presentan soluciones de problemas similares y permite al sistema que genere una solución procedimental a partir de estas demostraciones. Los esquemas individuales para generalizar tales soluciones van desde simular una secuencia procedimental o inferir intenciones.

Los sistemas que inferen, intentan generalizar usando razonamiento basado en el conocimiento. Una solución basada en la inferencia intenta determinar en qué son similares un grupo de datos u objetos, y, a partir de ello, generalizar estas similitudes.

Otra solución es la programación asistida: el sistema observa acciones que el programador ejecuta, y si son similares o acciones pasadas, intentará inferir cuál es la próxima acción que hará el programador.

Las dos principales objeciones al sistema de inferencia son:

- Si no se comprueban exhaustivamente pueden producir programas erróneos que trabajan correctamente con los ejemplos de prueba, pero que fallen posteriormente en otros casos
- La capacidad de inferencia es tan limitada, que el usuario debe de guiar el proceso en la mayoría de los casos.

Los resultados más satisfactorios de los sistemas de inferencia son en áreas limitadas, donde el sistema tenía un conocimiento semántico importante de la aplicación.

El mayor problema que se presenta con estos sistemas, es conocer cuándo un programa es correcto. En el caso de los sistemas procedimentales, se consigue estudiando el algoritmo y el resultado de juegos de ensayo apropiados.

En el caso de los sistemas demostrativos el algoritmo se mantiene en una representación interna, y su estudio se sale del ámbito de estos sistemas. Por lo que la veracidad de la decisión se debe hacer exclusivamente sobre la base de la eficiencia del algoritmo sobre los casos específicos de prueba.

La programación demostrativa es del tipo "bottom-up" y se adapta bien a nuestra capacidad de pensar. Sin embargo en la mayor parte de los paradigmas la resolución del problema se efectúa aplicando métodos abstractos "top-down".

• Paradigma imperativo

Este paradigma se caracteriza por un modelo abstracto de computadora que consiste en un gran almacenamiento de memoria.

La computadora almacena una representación codificada de un cálculo y ejecuta una secuencia de comandos que modifican el contenido de ese almacenamiento. Este paradigma viene bien representado por la arquitectura **Von Neuman**, ya que utiliza este modelo de máquina para conceptualizar las soluciones: "Existe un programa en memoria que se va ejecutando secuencialmente, y que toma unos datos de la memoria, efectúa unos cálculos y actualiza la memoria".

La programación en el paradigma imperativo consiste en determinar qué datos son requeridos para el cálculo, asociar a esos datos unas direcciones de memoria, y efectuar paso a paso una secuencia de transformaciones en los datos almacenados, de forma que el estado final represente el resultado correcto.

En su forma pura este paradigma sólo soporta sentencias simples que modifican la memoria y efectúan bifurcaciones condicionales e incondicionales. Incluso cuando se añade una forma simple de abstracción procedimental, el modelo permanece básicamente sin cambiar. Los parámetros de los procedimientos son "alias" de las zonas de memoria, por lo que pueden alterar su valor, y no retorna ningún tipo de cálculo. La memoria también se puede actualizar directamente mediante referencias globales.

El paradigma imperativo debe su nombre al papel dominante que desempeñan las sentencias imperativas. Su esencia es el cálculo iterativo, paso a paso, de valores de nivel inferior y su asignación a posiciones de memoria.

Si se analizan las características fundamentales de este tipo de paradigma se detectan las siguientes:

- Concepto de celda de memoria ("variable") para almacenar valores. El componente principal de la arquitectura es la memoria, compuesto por un gran número de celdas donde se almacenan los datos. Las celdas tienen nombre (concepto de variable) que las referencian, y sobre los que se producen efectos de lado y definiciones de alias.
- Operaciones de asignación. Estrechamente ligado a la arquitectura de la memoria, se encuentra la idea de que cada valor calculado debe ser "almacenado", es decir asignado a una celda. Esta es la razón de la importancia de la sentencia de asignación en el paradigma imperativo. Las nociones de celda de memoria y asignación en bajo nivel, se tienden a todos los lenguajes de programación y fuerzan en los programadores un estilo de pensamiento basado en la arquitectura Von Neumann.
- Repetición. Un programa imperativo, normalmente realiza su tarea ejecutando repetidamente una secuencia de pasos elementales, ya que en este modelo computacional la única forma de ejecutar algo complejo es repitiendo una secuencia de instrucciones.

A este tipo de paradigma de programación se le suele llamar algorítmico, dado que el significado de algoritmo es análogo al de receta, método, técnica, procedimiento o rutina, y se define como "un conjunto finito de reglas diseñadas para crear una secuencia de operaciones para resolver un tipo específico de problemas". De esta forma para N. Wirth, un programa viene definido por la ecuación:

Algoritmos + Estructura de Datos = Programas

No obstante, entendemos que aunque el concepto de algoritmo encaja en otros tipos de paradigmas, es privativo del tipo de programación procedimental en el que su característica fundamental es la secuencia computacional.

Atendiendo a los lenguajes imperativos, cabe clasificarlos en "orientados a expresiones" y "orientados a sentencias", según jueguen las expresiones o sentencias un papel más predominante en el lenguaje, respectivamente. Ambos son términos relativos y no se pueden aplicar de forma absoluta. Se puede decir que C, FORTRAN; Algol, Pascal, son lenguajes orientados a expresiones, mientras que COBOL y PL/1 están orientados a sentencias.

Las expresiones se han encontrado útiles principalmente porque son simples y jerárquicas y pueden combinarse uniformemente para construir expresiones más complejas. Si pues, las expresiones no sufren influencias de la arquitectura de Von Neumann. Como ejemplos de programación imperativa, se muestran en lenguaje Pascal la generación de números primos mediante la criba de Eratóstenes, y en lenguaje C la ordenación de datos mediante el método de la burbuja.

Números Primos

(* Genera números primos en el rango 2..n, utilizando la criba de Eratóstenes *)
Program primos(input, output)

Const n=50

Var i: 2..n;

j: 2..25;

iprimo: **boolean**

Begin

for i:=2 to n do

Begin (* ¿ Es primo y ? *)

j:=2; iprimo:=true;

While iprimo and (j<=i div 2) do

if ((y mod j) <>0)then
 j:=j+1

else
 iprimo:=false;

(* Si es primo imprime su valor *)

if primo then write (y:3)

End

End.

En este programa se destacan las tres características principales del paradigma imperativo:

- Celdas de memoria: Declaraciones de var (i,j, primo)
- Operaciones de asignación de tipo destructivo: j:=2, iprimo:=true, ...
- Sentencias repetitivas: for y while

Ordenar

/* Ordena un conjunto de datos leídos desde el dispositivo de entrada, utilizando el método de la burbuja */

#include <stdio.h>

void ordenar (int *, int);

void escribir (int *, int);

main()

```
{
    int n=0, dato;
    int a[50];
    scanf("%d", &dato);
    while(dato != EOF)
    {
        a[n++] = dato;
        scanf("%d",&dato);
    }
    ordenar(a,n-1);
    escribir(a,n-1);
}
```

/* función ordenar */

void ordenar (int a [], int n)

```
{
    int i,j, tem;
    for (i=0; i<n; y++)
        for (j=i+1; j<=n; j++)
            if (a[i]>a[j])
            {
                tem = a[i];
                a[i]=a[j];
                a[j]=tem;
            }
}
```

void escribir (int a [], int n)

```
{
    int i;
    for (y=0; i<n; y++)
    {
        if (!(i % 10)) printf ("\n");
        printf ("% 5d", a[i]);
    }
}
```

La función ordenar compara parejas de datos contiguos e intercambia su contenido si el primero es mayor que el segundo. Y en cada iteración de la variable *j*, coloca en la cabeza de la lista que está ordenando el dato menor (la burbuja más ligera). En este ejemplo, al igual que en anterior predominan los tres componentes descritos del paradigma imperativo.

• Paradigma funcional

El paradigma funcional está basado en el modelo matemático de composición funcional. En este modelo, el resultado de un cálculo es la entrada del siguiente, y así sucesivamente hasta que una composición produce el valor deseado.

No existe el concepto de celda de memoria que es asignada o modificada. Más bien, existen valores intermedios que son el resultado de cálculos anteriores y las entradas a cálculos subsiguientes. Tampoco existen sentencias imperativas y todas las funciones tienen transparencia referencial.

La programación funcional incorpora el concepto de función como objeto de primera clase, lo que significa que las funciones se pueden tratar como datos (pueden pasar como parámetros, calculadas y devueltas como valores normales, y mezcladas en el cálculo con otras formas de datos)

En este paradigma el informático concibe la solución como una composición de funciones. Por ejemplo, para ordenar una lista, se puede diseñar la solución como una concatenación de listas más pequeñas, cada una de las cuales ya está clasificada. Esto reduce el problema a seleccionar las listas más pequeñas. La forma en que se especifican las funciones puede variar. Se pueden especificar procedimentalmente o matemáticamente mediante su definición, sin secuencia de control.

Un ejemplo de lenguaje que soporta la forma procedimental del paradigma funcional, es el COMMONLISP, considerado como el estándar del lenguaje de programación LISP. Se ha desarrollado como un lenguaje funcional procedimental, y que admite ambos tipos de construcciones: la composición funcional con transferencia referencial y la asignación de variables con secuencias de control en la ejecución del programa.

El lenguaje LISP ("List Processing") fue diseñado por John McCarthy, en 1959, en el Instituto Tecnológico de Massachusetts, incorporando los siguientes elementos en su expresión más simple

a) Funciones primitivas predefinidas:

Tratamiento de listas

CAR Devuelve la cabeza de la lista
 CDR Devuelve la cola de la lista
 CONS Construye una lista CONS a partir de otras dos
 LIST Construye una lista a partir de varias sublistas

Asignación, relaciones booleanas y predicados

SET y SETQ Evalúan sus argumentos
 EQ Comprueba si dos argumentos son iguales
 GREATERP y LESSP Compara si el arg 1 es mayor/ menor que el 2
 NULL Comprueba si la lista es nula
 ATOM; LISTP NUMBERP Averiguan el tipo de un objeto
 OR, AND y NOT Predicados de suma, multiplicación y negación

Aritméticas

PLUS, DIFFERENCE, TIMES, QUOTIENT, REMAINDER Suma, resta, multiplicación, cociente, resto

Condicionales e iterativas

COND Condicional con varios predicados
 LOOP Función iterativa no declarativa

b) Formas funcionales para crear y combinar funciones

DEFUN Función que permite crear otras funciones

c) Objetos de datos

LISP dispone de esta estructura como fundamental

Un programa LISP utiliza:

Átomos Cadenas de números o caracteres
 identificadores Son átomos no numéricos

Seguidamente se presentan en LISP dos ejemplos. El primero sirve para comprobar si un número es primo o no; el segundo ordena de forma ascendente una lista de datos.

Primo

::: Determina si un número es primo o no

```
(DEFUN PRIMO(N)
  COND((EQ N 2) T) ((EQ N 3) T)
  (T (PRIMO1 N (QUOTIENT N 2))))))
(DEFUN PRIMO1 (N Y)
  COND((EQ (REMAINDER N Y) 0) NIL)
  ((EQ Y 2) T)
  (T (PRIMO1 N (DIFFERENCE Y 1))))
```

Y al ejecutarlo, por ejemplo se tendría,

```
(PRIMO 3) T
(PRIMO 8) NUL
```

Ordena

::: Ordena de forma ascendente los elemento de una lista

```
(DEFUN ORDENAR (LISTA)
  COND((NULL LISTA) NIL)
  (T (CONS (CAR (INVERSO (BURBUJA LISTA)))
  (ORDENAR (CDR (INVERSO (BURBUJA LISTA)))))))
(DEFUN BURBUJA (LISTA)
  (COND((NULL (CDR LISTA)) LISTA)
  ((GRATERP (CAR LISTA) (CADR LISTA))
  (CONS (CAR LISTA) (BURBUJA (CDR LISTA))))
  (T (CONS (CADR LISTA)
  (BURBUJA (CONS (CAR LISTA) (CDDR LISTA)))))))
```

y al ejecutarse se tendría, por ejemplo,

```
(ORDENAR 20 8 1 3) (1,3,8,20)
```

la función INVERSO invierte la lista

• Paradigma orientado a objetos

La Orientación a Objetos está constituyendo una metodología de diseño y desarrollo de software barato. Esta metodología de análisis, diseño y programación que configura las fases fundamentales del ciclo de vida de un sistema informático, se está asentando como la estructura metodológica de los años 90, y se presenta educacionalmente como paradigmática en el desarrollo de aplicaciones.

La Orientación a Objetos se puede definir como "una disciplina de ingeniería de desarrollo y modelado de software que permite construir más fácilmente sistemas complejos a partir de componentes individuales".

La Orientación a Objetos permite una representación más directa del mundo real, reduciendo fuertemente la transformación radical normal desde los requerimientos del sistema, definidos en términos del usuario, a las especificaciones del sistema, definidas en términos del ordenador.

Actualmente, la tendencia de la Ingeniería Informática es la de producir componentes reutilizables para ensamblarlos a otros y obtener así, el producto. Estos elementos reutilizables son denominados "Componentes Integrados de Software" (CIS) por su teórica similitud con los "componentes integrados de hardware (chips), innovación que revolucionó la industria informática hace 20 años. El Paradigma Orientado a Objetos (PaOO) es pues una filosofía de desarrollo y empaquetamiento de software que permite crear unidades funcionales extensibles y genéricas, de forma que el usuario las pueda aplicar según sus necesidades y de acuerdo con las especificaciones del sistema a desarrollar.

La Orientación a Objetos proporciona mejores herramientas para:

- Modelar el mundo real de un modo más cercano a la perspectiva del usuario.
- Interactuar fácilmente con un entorno computacional, usando metáforas familiares.
- Construir componentes reutilizables de software y librerías específicas, de estos componentes fácilmente extensibles.
- Modificar y ampliar con facilidad la implementación de estos componentes sin afectar al resto de su estructura.

En cuanto a los elementos fundamentales que configuran el Paradigma Orientado a Objetos, algunos autores los centran en tres: Los Tipos Abstractos de Datos, la Herencia y la Identidad de los Objetos. Mientras que otros señalan siete aspectos básicos a considerar para una verdadera orientación al objeto:

- Estructura modular basada en objetos, dado que los sistemas en esta metodología son modularizados sobre la base de sus estructuras de datos.
- Abstracción de Datos, porque los objetos son descritos como implementaciones de tipos de datos abstractos.
- Gestión automática de memoria, de forma que los objetos no utilizados sean asignados por el propio sistema sin intervención del programador
- Clases, en las que cada tipo no simple sea un módulo, y cada módulo de alto nivel sea un tipo.
- Herencia, que permita que una clase sea definida como una extensión o restricción de otra
- Polimorfismo y enlace dinámico, de forma que las entidades del programa puedan referenciar en tiempo de ejecución a objetos de diferentes clases.

- Herencia múltiple y repetida para que se pueda declarar una clase como heredera de varias, e incluso de ella misma.

Por otra parte, el Paradigma Orientado a Objetos presenta diferente terminología según la etapa del ciclo de vida que se contempla. Así, en las fases de análisis (análisis orientado a objetos-AOO) y de diseño (diseño orientado a objetos-DOO), se utilizan los conceptos: Identificación del Objeto, Identificación de las Estructuras y Materias, Definición de Elementos, Definición de Atributos, Conexión de Mensajes, Instancias, Definición de Servicios. Mientras que en la etapa de programación (programación orientada a objetos-POO), los términos a utilizar son: Tipo de Dato Abstracto, Objeto, Mensaje, Clase, Sobrecarga, Entace Dinámico, Polimorfismo, Herencia, Variables de Clase o de Instancia, Métodos, Constructores, Destruyores, Genericidad, Aserción, Invariante, ...

Todos estos términos son metáforas del paradigma, y configuran el lenguaje o "jerga" que utiliza el ingeniero informático en el diseño y desarrollo de software aplicando esta metodología.

Las Ciencias de la Computación están llenas de metáforas que han llegado a trascender el círculo especializado del informático y son utilizadas de forma rutinaria por la inmensa mayoría del público, debido a la incorporación del ordenador en la actividad diaria. Dos excelentes ejemplos son los términos "mensajes" y "ventanas". Estas palabras presentan poca semejanza real con los conceptos que representan en el lenguaje habitual. No obstante, los términos informáticos han quedado acuñados y los utilizamos indiscriminadamente, aparentemente sin gran dificultad.

El Paradigma Orientado a Objetos se describe a menudo usando el concepto Objeto/Mensaje, en el que cada objeto (elemento autónomo de información creado en tiempo de ejecución) es solicitado para realizar un determinado servicio el usuario. Mediante el envío a ese objeto del mensaje apropiado. El solicitante no precisa conocer cómo el objeto proporciona el servicio pedido la implementación es interna al objeto y la gestiona el suministrador del objeto. El énfasis se produce en qué se puede obtener más bien que en cómo se obtiene.

Un programa orientado a objetos viene definido por la ecuación:

$$\text{Objetos} + \text{Mensajes} = \text{Programa}$$

En el que el objeto es una instancia de una clase, la cual implementa un tipo abstracto de dato (TAD). Y el mensaje es la información específica que se envía al objeto para que ejecute una determinada tarea. Un TAD define conjuntos encapsulados de objetos similares, con una colección asociada de operaciones; y especifica la estructura y el comportamiento de los objetos. Las especificaciones estructuradas del TAD describen las características de los objetos pertenecientes a ese TAD, y las especificaciones de comportamiento describen qué mensajes son aplicables a cada objeto

Algunos lenguajes orientados a objetos, como el C++, Object LISP, etc., son extensiones de lenguajes convencionales que fueron diseñados básicamente como imperativos, por lo que estas extensiones no consideran a los objetos como entidades activas que reciben mensajes, más bien invocan a procedimientos o funciones que pasan objetos; no obstante, incorporan las características más relevantes POO, en particular: la encapsulación de datos, el polimorfismo con entace dinámico, la generalidad y la herencia.

El lenguaje paradigmático de este tipo de programación es el SMALLTALK que fue diseñado a principios de los años 70 en el Centro de Investigación Xerox (Palo Alto, USA) con el fin de disponer de un potente sistema informático donde el usuario pudiera almacenar y manipular la información ágilmente. Durante los diez años siguientes se produjeron tres sistemas base: SMALLTALK-1 SMALLTALK-76 y SMALLTALK-80. Actualmente, existen dos versiones del SMALLTALK, V1 y V2, producidas por Xerox y distribuidas por Parc Place Systems, siendo aceptada la V2 como estándar del sistema SMALLTALK-80. También la firma Digitalk Inc. ha desarrollado un SMALLTALK simplificado, destinado a fines pedagógicos, con el nombre de SMALLTALK V.

No obstante, dada la relevancia que está adquiriendo este tipo de programación es usual que todos los lenguajes de propósito general de una cierta utilización incorporen, como lo han hecho el C, el PASCAL y el LISP, los conceptos que definen el PaOO, quedando el SMALLTALK como lenguaje puramente académico.

En concreto, podemos señalar que el entorno de programación orientado a objetos es un entorno:

- Procedimental, que incorpora el concepto de encapsulación de datos.
- Extensible, ya que permite ampliar los componentes software sin necesidad de la burbuja de reprogramar todo el componente.
- Reutilizable, por permitir crear un programa sobre la base de una serie de componentes definidos en librerías de componentes.
- Interactivo a través de una interfaz gráfica de usuario que proporciona la funcionalidad necesaria en materia de edición, puesta a punto de programas y archivo de información.

Herramientas CASE.

CASE (Ingeniería de Software Asistida por Computadora). El crecimiento de la industria CASE reviste una particular importancia para el desarrollo de software. Las herramientas CASE son el equivalente para la industria del software de las herramientas CAD (Diseño asistido por computadora) para actividades como la organización de un circuito y el diseño de chips.

Las herramientas CASE utilizan representaciones gráficas en la pantalla para ayudar a automatizar la planeación, el análisis, el diseño y la generación del software.

• Paradigma heurístico

Muchas de las más interesantes y difíciles tareas de programación implican utilizar el ordenador para resolver problemas del tipo: "¿Cuál es el camino más corto?" "Listar todos los casos posibles", "¿Existe una disposición de elementos que satisfaga?". Las características de tales problemas implica potencialmente una búsqueda exhaustiva de todas las posibles combinaciones de algún conjunto finito, que si no está controlado puede producir una "explosión combinatoria" (incremento exponencial del espacio de búsqueda con la dimensión del problema) imposible de tratar.

La Programación Heurística ha venido a significar el uso del conocimiento específico del dominio para cubrir esta explosión de posibilidades guiando la búsqueda por las direcciones más prometedoras. Y se puede definir como "aqueel tipo de programación computacional que aplica para la resolución de problemas reglas de buena lógica (reglas del pulgar) denominadas heurísticas, las cuales proporcionan entre varios cursos de acción uno que presenta visos de ser el más prometedor, pero no garantiza necesariamente el curso de acción más efectivo."La Programación Heurística implica una forma de modelizar el problema en lo que respecta a la representación de su estructura, estrategias de búsqueda y métodos de resolución, que configuran el Paradigma Heurístico.

Este tipo de programación se aplica con mayor intensidad en el campo de la Inteligencia Artificial (I.A.), y en especial, en el de la Ingeniería del Conocimiento, dado que el ser humano opera la mayor parte de las veces utilizando heurísticas, un hecho cierto que una heurística es la conclusión del razonamiento humano en un dominio específico, por lo que es normal que este tipo de programación que encuadrado en el área de la I.A., ya que implementa el conocimiento humano, dado por la experiencia, utilizando reglas de buena lógica.

Como se ha señalado inicialmente, un paradigma de programación es un modelo básico de diseño e implementación de programas. Un modelo que permite producir programas de acuerdo con una metodología específica. Así, el paradigma de programación estructurada se basa en estructuras modulares, con fuerte cohesión en el módulo y bajo acoplamiento entre ellos, desarrollo "top-down", utilización de diagramas privilegiados, etc.

El Paradigma Heurístico define, pues, un modelo de resolución de problemas en el que se incorpora alguna componente heurística sobre la base de:

- Una representación más apropiada de la estructura del problema para su resolución con técnicas heurísticas
- La utilización de métodos de resolución de problemas aplicando funciones de evaluación con procedimientos específicos de búsqueda heurística para la consecución de las metas.

Por otra parte, la Programación Heurística se presenta y utiliza desde diferentes puntos de vista:

- Como técnica de búsqueda para la obtención de metas en problemas no algorítmicos, o con algoritmos que generan explosión combinatoria (Ej. damas, ajedrez, etc.).
- Como un método aproximado de resolución de problemas utilizando funciones de evaluación de tipo heurístico (Ej.. algoritmos A*, AO*)
- Como método de poda para estrategias de programas que juegan, aunque estos métodos no son realmente heurísticos (Ej.. poda alfa-beta).

Se aconseja utilizar un modelo heurístico cuando:

- Los datos, limitados e inexactos, utilizados para estimar los parámetros modelo pueden contener errores inherentes muy superiores a los proporcionados con la solución de una buena heurística.
- Se utiliza un modelo simplificado, que por si es una representación imprecisa de un problema real, por lo que la solución "óptima" es puramente académica.
- No se dispone de un método exacto que sea fiable para ser aplicado en un modelo del problema; o si existe, es intratable computacionalmente.
- Se desea mejorar la eficacia de un algoritmo optimizador aplicado al modelo por ejemplo, proporcionando buenas soluciones de inicio, guiando la búsqueda y reduciendo el número de soluciones candidatas
- Se tiene la necesidad de resolver el mismo problema frecuentemente, o una base de tiempo real, y el tratamiento heurístico significa un ahorro computacional

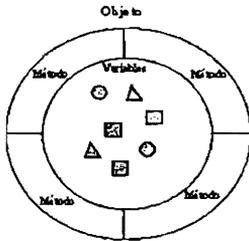
En general, un modelo heurístico es aconsejable si puede proporcionar resultados superiores a los del modelo actual. Las especificaciones más relevantes del tratamiento heurístico deben tener en cuenta las características de la heurística, de la información y de las especificaciones del problema de acuerdo con las siguientes condiciones,

- Una buena heurística debe ser simple, con requerimientos razonables de memoria, con velocidad de búsqueda que no produzca incrementos polinomiales, ni exponenciales, precisa, robusta, que proporcione soluciones múltiples, y que disponga de un buen criterio de parada que incorpore el conocimiento obtenido durante la búsqueda.
- La información a tratar es fundamentalmente simbólica, inexacta o limitada, "incremental" y basada en el conocimiento.
- Las especificaciones del problema pueden ser: de optimización o de satisfacción; que produzcan una o múltiples soluciones; con tratamiento en tiempo real o no; con decisión interactiva o no, etc.

La Programación Heurística no ha producido un lenguaje específico de programación, debido a que las heurísticas, al "ser reglas de sentido común", se pueden implementar con cualquiera de los lenguajes descritos en los diferentes paradigmas de programación. Por otra parte, al aplicarse este tipo de programación, fundamentalmente, en el campo de la I.A., las heurísticas están siendo implementadas mayoritariamente en herramientas de este área.

No obstante, si se tuviera que definir un lenguaje heurístico las características más destacables que debería incorporar son:

- Ser un lenguaje conversacional, que permitiera una interacción directa con el programador para la definición e implementación del problema.
- Tratamiento de estructuras "incrementales, que implementen programas que vayan ampliando el cuerpo de conocimiento que, sobre la base de la experiencia, configure y refine el modelo heurístico.
- Tratamiento fundamentalmente simbólico, dado que la mayor parte de los problemas que precisan tratamiento heurístico tienen estructura simbólica.
- Unidades funcionales autónomas que posibiliten modelar una heurística y su mecanismo de ejecución, definiendo módulos independientes
- Estructuras de datos que permitan describir estados de problemas y relaciones entre estados.
- Estructuras procedimentales de control y de proceso (o de definición) que permitan la ejecución coherente del modelo heurístico, y posibiliten la adquisición y utilización del conocimiento adquirido en el proceso de resolución del problema.



DESARROLLO DE UN SISTEMA ORIENTADO A OBJETOS.

La "orientación a objetos" será la más importante de las tecnologías que surjan en los años noventas. Bill Gates, Presidente de Microsoft Corp.

INTRODUCCION

Algunas de las modernas técnicas de programación están creando un pequeño caos en la industria del desarrollo de software. Normalmente, este caos se produce en quienes han programado con metodologías anteriores y que, obviamente, causan algunos dolores de cabeza entender las nuevas. A continuación se presenta una historia breve de la evolución de la programación, desde sus orígenes hasta nuestros días.

• Programación por líneas.

Una de las primeras estrategias en la programación fue la ejecución secuencial de instrucciones proporcionadas a la máquina. Todo el programa era un papel con una serie de instrucciones, una detrás de la otra, y sobrevinieron problemas cuando las aplicaciones aumentaron de tamaño y complejidad, pues fue necesario hacer bifurcaciones, repetir procesos (con la consecuente duplicidad de código), etcétera, lo que llegaba a hacer de los programas una maraña comparable con un plato de espagueti. Por esto, a la postre, a este sistema se le llamó "Programación espagueti". Cobol, Fortran y BASIC son los clásicos representantes de esta línea.

```

10 DIM NOMBRE=" "
20 DIM A=0: DIM B=0: DIM C=0
30 READ NOMBRE
40 PRINT "HOLA ": PRINT "NOMBRE"
50 PRINT " SE REALIZARA UNA SUMA"
60 INPUT A: INPUT B
70 C= A+ B
80 PRINT " EL RESULTADO ES: "
90 PRINT C
100 STOP

```

- Ejemplo de un programa de "SUMA" de dos valores hecho en programación por líneas.

- **Programación estructurada.**

Los problemas inherentes al código lineal se resolvieron en gran parte cuando se introdujo la llamada "Programación estructurada". Esta línea de programación trajo consigo nuevos conceptos basados en "divide y vencerás" que incluyen: 1) la creación de núcleos de código pequeños y reutilizables llamados Procedimientos, que permitan ejecutar tareas repetitivas sin necesidad de duplicar código del todo y 2) la instauración del concepto de Variable pública y Variable local, la primera es accesible a todos los procedimientos del programa, y la segunda, sólo en el procedimiento que la contiene. Estos conceptos trajeron grandes avances a la programación y, de igual modo, permitieron el establecimiento de lenguajes 100% estructurados como C, Pascal y algunas versiones de BASIC estructurado como Quick BASIC

PROGRAM EJEMPLO;

Var

A,B,C : Integer
NOMBRE : string

Begin

A:=0;
B:=0;
C:=0;

Write (" Dame tu Nombre: ");
Readln(NOMBRE);

Write (" Hola ");
Write (NOMBRE);

Writeln (" A continuación se realizará una suma ");

Write (" Dame el primer valor : ");
Readln(A);

Write (" Dame el segundo valor : ");
Readln(B);

Write (" El resultado de la suma es : ");
Write (C);

END.

- *Ejemplo del mismo programa de "SUMA", en programación estructurada.*

Aunque la programación estructurada, hecha entre mediados y finales de los años sesenta, sigue siendo uno de los métodos de programación más utilizados, poco a poco se ha visto mermada por sus carencias frente a las nuevas exigencias de cómputo del mundo moderno. Posiblemente, la programación estructurada como tal, tarde mucho en morir, tal vez más de lo que los entendidos pudiesen esperar.

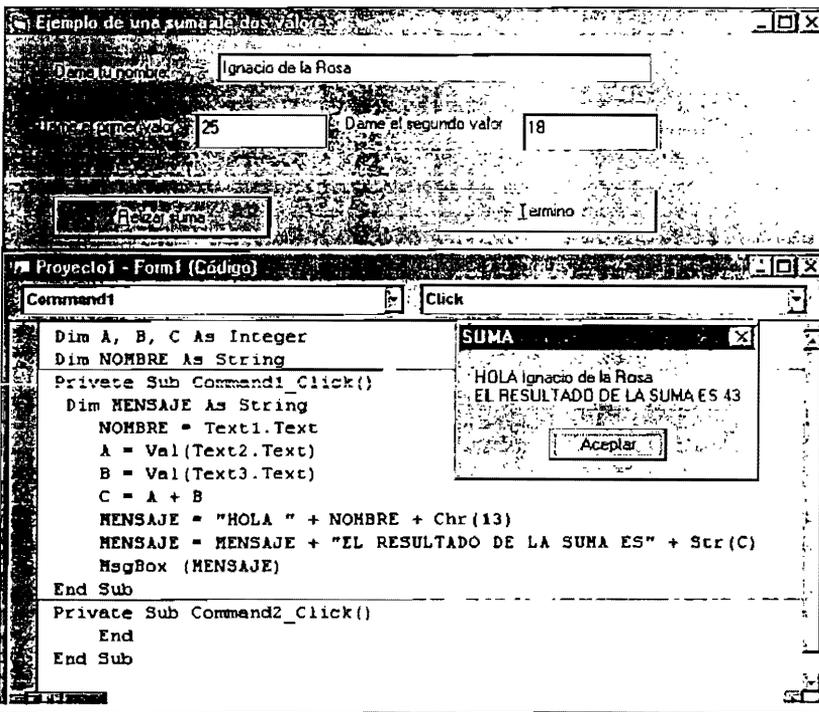
• Programación orientada a objetos.

La Programación orientada a objetos (POO) es una nueva capa de esta evolución que se basa en la ideología de "muchas hormigas juntas mueven una montaña". En realidad, la propuesta es sencilla: crear individuos con características y facultades propias, y unirlos con un propósito común.

El concepto puede explicarse con el siguiente ejemplo: imagine que existe un individuo X que tiene la habilidad de manejar la computadora. Si un empresario lo contrata, será exactamente para aprovechar sus capacidades en pro de los intereses de la empresa. De hecho, dado que X tiene estas habilidades, será más fácil adaptarlo a las necesidades expresas que en tal rubro se tienen. Esto, por supuesto, es más fácil que entrenar a uno nuevo; esto es, de manera muy general, la ideología que define la POO. El uso de entidades propias que pueden ejecutar satisfactoriamente una tarea, o aprovechar sus capacidades para adaptartas a las necesidades de la tarea, es más fácil que crear una entidad desde el inicio.

Cada entidad se llama objeto, el cual envuelve una serie de características y acciones para llevar a cabo una tarea. Un objeto Individuo puede tener entre sus características distintivas el color de sus ojos, de su piel, de su cabello, su complejión, estatura, etcétera. Las acciones del individuo definen las cosas que puede hacer, como caminar, comer, hablar. Estas acciones pueden afectar exclusivamente al individuo que las lleva a cabo o a terceros, por ejemplo, cuando hable con otros. El tercero puede reaccionar a la acción proveniente del primer individuo.

• Ejemplo del mismo programa de "SUMA", en programación Orientada a Objetos.



Elementos de la POO

Nota: Para la descripción de los conceptos y temas aquí presentados se utilizó el lenguaje de programación Visual Basic* debido a que el sistema realizado fue hecho en esta herramienta de programación.

• Un objeto.

Aunque ya se había definido lo que es un objeto, se hará un análisis más detallado: un objeto es una unidad de código y datos. Este objeto representa un grupo de variables y procedimientos relacionados que pueden ser manipulados como una sola unidad.

Las personas nos formamos conceptos desde temprana edad. Cada concepto es una idea particular o una comprensión de nuestro mundo. Los conceptos adquiridos nos permiten sentir y razonar acerca de las cosas en el mundo. A estas cosas a las que se aplican nuestros conceptos se llaman objetos. Un objeto podría ser real o abstracto, como los ejemplos siguientes:

- Una factura.
- Una organización.
- Un automóvil.
- Un mecanismo en un dispositivo de robótica.
- Un proceso para llenar un pedido.
- Este trabajo de tesis.

En el análisis y diseño orientado a objetos (OO), lo que interesa es el comportamiento del objeto. Al construir un software OO se basa en los tipos de objetos. El software que implanta el objeto contiene estructuras de datos y operaciones que expresan dicho comportamiento. Las operaciones se codifican como métodos. Los procedimientos en un objeto definen sus acciones y son llamados Métodos. Las variables son usadas normalmente para establecer y regresar atributos o comportamiento y son llamadas Propiedades.

Un objeto puede estar compuesto por otros objetos. Estos últimos, a su vez, puede estar compuesto de objetos, del mismo modo que una máquina está formada por partes y éstas, también están formadas por otras partes. Esta estructura intrincada de los objetos permite definir objetos muy complejos.

Los conceptos que se poseen se aplican a tipos determinados de objetos. Por ejemplo empleado se aplica a los objetos que son personas empleadas en una organización. Algunas instancias de empleado podrían ser Juan Carranza, Jaime Rosas, etc. En el análisis orientado a objetos, estos conceptos se llaman tipos de objetos; las instancias se llaman objetos.

Un tipo de objeto es una categoría de objeto
Un objeto es una instancia de un tipo de objeto.

El mundo de las bases de datos define tipos de entes, como cliente, empleado o parte. Existen muchas instancias de cada tipo de ente. Por ejemplo, las instancias de empleado son Juan Carranza, Jaime Rosas, etc. Del mismo modo, el mundo orientado a objetos define tipos de objetos e instancias de tipos de objetos. Por ejemplo, un tipo de objeto podría ser factura y un objeto podría ser factura #51783.

Sin embargo, el término objeto tiene diferencias fundamentales con el término ente. Ente sólo se refiere a los datos. En general, se almacena un registro por cada ente. Objeto se refiere a los datos y a los métodos mediante los cuales se controla a los propios datos.

Un objeto es una combinación de código y datos que se puede tratar como una unidad. Un objeto puede ser una parte de una aplicación, como un control o un formulario. También puede ser un objeto una aplicación entera.

La tabla siguiente describe ejemplos de tipos de objetos que puede utilizar en Visual Basic*.

Ejemplo	Descripción
Botón de comando	Son objetos los controles de un formulario, como botones de comandos y marcos.
Formulario	Cada formulario de un proyecto de Visual Basic* es un objeto distinto.
Base de datos	Las bases de datos son objetos y contienen otros objetos, como campos e índices.
Gráfico	Un gráfico de Microsoft Excel* es un objeto.

De dónde vienen los objetos

Cada objeto se define mediante una *clase*. Para comprender la relación entre un objeto y su clase, se puede pensar en el molde de las galletas y las galletas. El molde es la clase. Define las características de cada galleta, como por ejemplo el tamaño y la forma. Se utiliza la clase para crear objetos. Los objetos son las galletas.

Mediante dos ejemplos se puede aclarar la relación que existe entre las clases y los objetos.

- Los controles del cuadro de herramientas de Visual Basic representan clases. El objeto conocido como control no existe hasta que se dibuja en un formulario. Cuando se crea un control, está creando una copia o *instancia* de la clase del control. Esta instancia de la clase es el objeto al que se hará referencia en la aplicación.
- El formulario en el que se trabaja en tiempo de diseño es una clase. En tiempo de ejecución, Visual Basic* crea una instancia de esa clase de formulario.

La ventana Propiedades muestra la clase y la propiedad Name de los objetos de una aplicación de Visual Basic*, como se muestra en la figura:

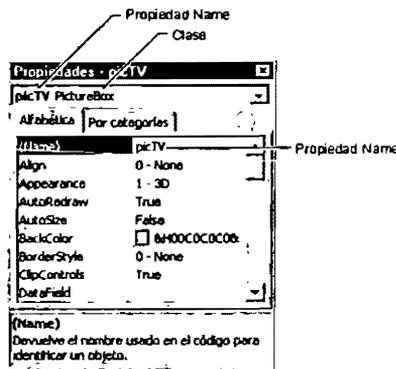


Figura Los nombres de objetos y clases se muestran en la ventana Propiedades

* Microsoft, Project, Visual Basic, Word y Excel son marcas registradas por MICROSOFT, co

Se crean todos los objetos como copias idénticas de sus clases. Una vez que existen como objetos individuales, es posible modificar sus propiedades. Por ejemplo, si se dibuja tres botones de comando en un formulario, cada objeto botón de comando es una instancia de la clase CommandButton. Cada objeto comparte un conjunto de características y capacidades comunes (propiedades, métodos y eventos), definidos por la clase. Sin embargo, cada uno tiene su propio nombre, se puede ser activar y desactivar por separado y se puede colocar en una ubicación distinta del formulario, etc.

Un objeto proporciona código que no se tiene que escribir. Por ejemplo, se puede crear cuadros de diálogo **Abrir archivo** y **Guardar archivo**, pero no se tiene por qué. En vez de eso, se puede utilizar el control de diálogo común (un objeto) que Visual Basic proporciona. Se podría escribir una agenda y código de administración de recursos, pero no se tiene por qué. En su lugar, se puede utilizar los objetos Calendar, Resources y Tasks que proporciona Microsoft Project *.

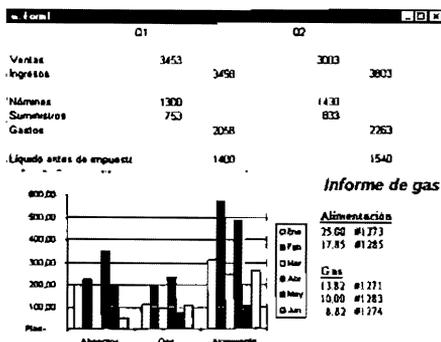
Visual Basic* puede combinar objetos de otros orígenes

Visual Basic proporciona las herramientas que permiten combinar objetos de distintos orígenes. Ahora se puede construir soluciones personalizadas mediante la combinación de las características más poderosas de Visual Basic y de las aplicaciones que aceptan Automatización (antes conocido como Automatización OLE). *Automatización* es una característica del *Modelo de objetos componentes (COM)*, un estándar de la industria utilizado por las aplicaciones para exponer objetos a las herramientas de desarrollo y otras aplicaciones.

Se puede construir aplicaciones si agrupa entre sí controles intrínsecos de Visual Basic* y se puede utilizar también objetos que proporcionen otras aplicaciones. Es posible colocar estos objetos en un formulario de Visual Basic*:

- Un objeto Gráfico de Microsoft Excel *
- Un objeto Hoja de cálculo de Microsoft Excel *
- Un objeto Documento de Microsoft Word *

Podría utilizarse estos objetos para crear una aplicación con un cuadro de comprobación como el que se muestra en la figura. Esto ahorrará tiempo ya que no se tendrá que escribir el código que reproduzca la funcionalidad proporcionada por los objetos de Microsoft Excel* y Word *



* Microsoft, Project, Visual Basic, Word y Excel son marcas registradas por MICROSOFT co

• Propiedad

Los objetos aceptan propiedades, métodos y eventos. Los datos de un objeto (configuración o atributos) se llaman *propiedades*. Se puede cambiar las características de un objeto si modifica sus propiedades. Por ejemplo un radio. Una propiedad del radio es su volumen. Se diría que un radio tiene la propiedad "Volumen" que se puede ajustar modificando su valor. Supóngase que se establece el volumen de la radio de 0 a 10. Si se pudiera controlar el radio, se podría escribir el código en un procedimiento que modificara el valor de la propiedad "Volumen" de 3 a 5 para hacer que el radio suene más alto:

```
Radio.Volumen = 5
```

Control de objetos mediante sus propiedades

Las propiedades individuales varían ya que se puede establecer u obtener sus valores. Se pueden establecer algunas propiedades en tiempo de diseño. Se puede utilizar la ventana Propiedades para establecer el valor de dichas propiedades sin tener que escribir el código. Algunas propiedades no están disponibles en tiempo de diseño, por lo que se necesitará escribir código para establecer esas propiedades en tiempo de ejecución.

Las propiedades que se establecen y obtienen en tiempo de ejecución se llaman *propiedades de lectura y escritura*. Las propiedades que sólo puede leer en tiempo de ejecución se llaman *propiedades de sólo lectura*.

Establecimiento de los valores de las propiedades

Se puede establecer el valor de una propiedad cuando se desee modificar la apariencia o el comportamiento de un objeto. Por ejemplo, se puede modificar la propiedad Text de un cuadro de texto para modificar el contenido del cuadro de texto.

Para establecer el valor de una propiedad, se utiliza la sintaxis siguiente:

```
objeto.propiedad = expresión
```

Las instrucciones siguientes demuestran cómo se establecen las propiedades:

```
Text1.Top = 200           ' Establece la propiedad Top a 200 twips.
Text1.Visible = True     ' Muestra el cuadro de texto.
Text1.Text = "hola"      ' Muestra 'hola' en el cuadro de texto.
```

Obtención de los valores de las propiedades

Se puede obtener el valor de una propiedad cuando se desee encontrar el estado de un objeto antes de que el código realice acciones adicionales (como asignar el valor a otro objeto). Por ejemplo, puede devolver la propiedad Text a un control de cuadro de texto para determinar el contenido del cuadro de texto antes de ejecutar código que pueda modificar el valor.

En la mayoría de los casos, para obtener el valor de una propiedad se utiliza la sintaxis siguiente:

```
variable = objeto.propiedad
```

También se puede obtener el valor de una propiedad como parte de una expresión más compleja, sin tener que asignar el valor de la propiedad a una variable. En el siguiente código de ejemplo se calcula la propiedad **Top** del nuevo miembro de una matriz de controles como la propiedad **Top** del miembro anterior más 400:

```
Private Sub cmdAdd_Click()
    ' [instrucciones]
    optButton(n).Top = optButton(n-1).Top + 400
    ' [instrucciones]
End Sub
```

Sugerencia Si se va a utilizar el valor de una propiedad más de una vez, el código se ejecutará más rápidamente si se almacena el valor en una variable.

• Método

Además de propiedades, los objetos tienen métodos. Los métodos son parte de los objetos del mismo modo que las propiedades. Generalmente, los métodos son acciones que se desea realizar, mientras que las propiedades son los atributos que puede establecer o recuperar. Por ejemplo, marca un número de teléfono para hacer una llamada. Se podría decir que el teléfono tiene un método "Marcar" y podría utilizar esta sintaxis para marcar el número de ocho cifras 55552222:

```
Teléfono.Marcar 55552222.
```

Los métodos especifican la forma en que se controlan los datos de un objeto. Los métodos en un tipo de objeto sólo hacen referencia a las estructuras de datos de ese tipo de objeto. No deben tener acceso directo a las estructuras de datos de otros objetos. Para utilizar la estructura de datos de otro objeto, deben mandar un mensaje a este. El tipo de objeto empaca juntos los tipos de datos y los métodos.

Un objeto es entonces una cosa cuyas propiedades están representadas por tipos de datos y su comportamiento por métodos. Mientras que los diversos procedimientos que pueden operar sobre el objeto se conocen como sus métodos.

Un método asociado con el tipo de objeto factura podría ser aquél que calcule el total de una factura. Otro podría transmitir la factura a un cliente. Otro podría verificar de manera periódica si la factura ha sido pagada y en caso, contrario, añadir cierta tasa de interés.

Realizar acciones con métodos

Los métodos pueden afectar a los valores de las propiedades. Por ejemplo, en el ejemplo de la radio, el método **EstablecerVolumen** cambia la propiedad **Volumen**. De forma similar, en Visual Basic[®], los cuadros de lista tienen una propiedad **List** que se puede modificar con los métodos **Clear** y **AddItem**.

Uso de métodos en el código

Cuando se utiliza un método en el código, la forma en que escribe la instrucción depende de los argumentos que necesite el método y si el método devuelve o no un valor. Cuando un método no necesita argumentos, se escribe el código mediante la sintaxis siguiente:

```
objeto.método
```

En este ejemplo, el método **Refresh** vuelve a dibujar el cuadro de imagen:

```
Picture1.Refresh ' Obliga a redibujar el control.
```

Algunos métodos, como el método **Refresh**, no necesitan argumentos y no devuelven valores.

Si el método necesita más de un argumento, se debe separar los argumentos mediante comas. Por ejemplo, el método **Circle** utiliza argumentos que especifican la ubicación, el radio y el color de un círculo en un formulario:

```
' Dibuja un círculo azul con un radio de 1200 twips.  
Form1.Circle (1600, 1800), 1200, vbBlue
```

Si conserva el valor de devolución de un método, debe poner los argumentos entre paréntesis. Por ejemplo, el método **GetData** devuelve una imagen del Portapapeles:

```
Picture = Clipboard.GetData (vbCFBitmap)
```

Si no hay valor de devolución, los argumentos se ponen sin paréntesis. Por ejemplo, el método **AddItem** no devuelve ningún valor:

```
List1.AddItem "sunombre" 'Agrega el texto 'sunombre' a un cuadro de lista.
```

• Evento

Un evento es una acción reconocida por un formulario o un control. Las aplicaciones controladas por eventos ejecutan código como respuesta a un evento. Cada formulario y control de Visual Basic* tiene un conjunto de eventos predefinidos. Si se produce uno de dichos eventos y el procedimiento de evento asociado tiene código, Visual Basic llama a ese código.

Los objetos también tienen eventos. Los eventos se desencadenan cuando cambia algún aspecto del objeto. Por ejemplo, una radio podría tener el evento "CambiarVolumen" y el teléfono podría tener el evento "Sonar".

Aunque los objetos de Visual Basic reconocen automáticamente un conjunto predefinido de eventos, se puede decidir cuándo y cómo se responderá a un evento determinado. A cada evento le corresponde una sección de código (un procedimiento de evento). Cuando se desea que un control responda a un evento, se escribe el código en el procedimiento de ese evento.

Los tipos de eventos reconocidos por un objeto varían, pero muchos tipos son comunes a la mayoría de los controles. Por ejemplo, la mayoría de los objetos reconocen el evento Click: si un usuario hace clic en un formulario, se ejecuta el código del procedimiento de evento Click del formulario; si un usuario hace clic en un botón de comando, se ejecuta el código del procedimiento de evento Click del botón. El código en cada caso será diferente.

He aquí una secuencia típica de eventos en una aplicación controlada por eventos:

1. Se inicia la aplicación y se carga y muestra la aplicación.
2. La aplicación (o un control) recibe un evento. El evento puede estar causado por el usuario (por ejemplo, por la pulsación de una tecla), por el sistema (por ejemplo, un evento de cronómetro) o, de forma indirecta, por el código (por ejemplo, un evento Load cuando el código carga otra aplicación).
3. Si hay código en el procedimiento de evento correspondiente, se ejecuta.
4. La aplicación espera al evento siguiente.

Nota Muchos eventos se producen junto con otros eventos. Por ejemplo, cuando se produce el evento DbClick, se producen también los eventos MouseDown, MouseUp y Click.

• Polimorfismo

Polimorfismo significa que muchas clases pueden proporcionar la misma propiedad o el mismo método y que el que llama no tiene por qué saber la clase a la que pertenece el objeto antes de llamar a la propiedad o al método.

Por ejemplo, una clase Pulga y una clase Tiranosaurio podrían tener las dos un método Morder. El polimorfismo significa que puede invocar Morder sin saber si el objeto es una Pulga o un Tiranosaurio, aunque seguramente lo descubrirá después.

Los siguientes temas describen la implementación del polimorfismo en Visual Basic y cómo puede utilizarlo en sus programas.

- **Cómo proporciona Visual Basic* el polimorfismo** La mayoría de los lenguajes orientados a objetos proporcionan polimorfismo mediante la herencia; Visual Basic utiliza la técnica de interfaz múltiple del Modelo de objetos componentes (COM).
- **Crear e implementar una interfaz** Un amplio ejemplo de código muestra cómo puede crear una interfaz abstracta Animal e implementarla para las clases Tiranosaurio y Pulga.
- **Implementar propiedades** Las interfaces que implementa pueden tener tanto propiedades como métodos, aunque hay algunas diferencias en la forma de implementar las propiedades

• Herencia

Principio por el cual una clase se puede derivar de otra clase ya existente, y hereda las características del padre. Un ejemplo:

Se tiene el objeto fruta, con algunas propiedades inherentes a la fruta. Luego el objeto manzana, que es hijo de fruta y por lo tanto hereda las propiedades de la fruta pero puede tener otras propiedades propias de la manzana. El hijo hereda las características del papá, pero tiene otras propias y mejora (o empeora) algunas características del papá.

Diseño del sistema (Creación de la interfaz de usuario).

La mayoría de los principios del diseño de una interfaz de usuario son los mismos que los principios de diseño básicos que se imparten en los cursos elementales de arte. Los principios de diseño elementales de composición, color, etc. se aplica por igual a la pantalla de un equipo que a una hoja de papel o a un lienzo.

Composición: la apariencia de una aplicación.

La composición o distribución de un formulario no sólo afecta a su atractivo estético, sino que también tiene un tremendo impacto en la facilidad de uso de la aplicación. La composición incluye factores tales como la colocación de los controles, la coherencia de los elementos, su facilidad de uso, el uso del espacio en blanco y la sencillez del diseño.

Colocación de los controles.

En la mayoría de los diseños de interfases, no todos los elementos son de igual importancia. Es necesario un diseño cuidadoso para asegurar que los elementos más importantes sean inmediatamente accesibles para el usuario. Los elementos importantes o utilizados con más frecuencia deben tener una posición prominente; los elementos menos importantes deben estar relegados a posiciones menos prominentes.

En la mayoría de los idiomas, se nos enseña a leer de izquierda a derecha y de arriba a abajo en una página. Lo mismo se aplica a la pantalla de un equipo: los ojos de la mayoría de los usuarios irán primero a la parte superior izquierda de la pantalla, por lo que el elemento más importante debe estar allí. Por ejemplo, si la información de un formulario se refiere a un cliente, el campo con el nombre del cliente debe mostrarse allí donde se vea primero. Los botones, como **Aceptar** o **Siguiente**, deben estar situados en la parte inferior derecha de la pantalla; normalmente el usuario no los utilizará hasta que haya terminado de trabajar con el formulario.

La agrupación de elementos y controles también es importante. Se debe agrupar la información de forma lógica, de acuerdo con su función o su relación. Como las funciones están relacionadas, los botones para desplazarse por una base de datos deben estar agrupados visualmente en lugar de estar esparcidos por el formulario. Esto mismo puede aplicarse a la información; los campos de nombre y dirección se encuentran generalmente agrupados, ya que están estrechamente relacionados. En muchos casos, se puede utilizar marcos para reforzar la relación entre controles.

Coherencia de los elementos de la interfaz.

La coherencia es una virtud en el diseño de una interfaz de usuario. Una apariencia coherente aporta armonía a una aplicación: todo parece encajar perfectamente. La falta de coherencia de una interfaz puede provocar confusión y puede hacer parecer que la aplicación es caótica, desorganizada y barata, pudiendo llegar incluso a provocar al usuario dudas sobre la fiabilidad de la aplicación.

Para obtener coherencia visual, se debe establecer una estrategia de diseño y convenciones de estilo antes de empezar el desarrollo. Los elementos de diseño tales como los tipos de controles, el estándar de tamaño y agrupación de controles, y las opciones de fuente deben establecerse previamente. Puede crear prototipos de los posibles diseños para ayudar a tomar las decisiones de diseño.

La gran variedad de controles disponibles en el lenguaje tienta a utilizarlos todos. Se debe evitar la utilización de todos: elija el subconjunto de controles que mejor se adapte a la aplicación en particular. Mientras que los controles de cuadro de lista, cuadro combinado, cuadrícula y árbol pueden utilizarse para presentar listas de información, es mejor adoptar un único estilo en la medida de lo posible.

Además, se debe utilizar los controles de forma apropiada; si bien un control de cuadro de texto puede definirse como de sólo lectura y utilizarse para presentar texto, un control de etiqueta suele ser el más apropiado para este propósito. Se debe mantener la coherencia en el valor de las propiedades de los controles: si en un sitio se utiliza un color de fondo blanco para el texto modificable, no utilizar un fondo gris en otro sitio a menos que se tenga una buena razón para ello.

La coherencia entre los diferentes formularios de una aplicación es importante para su facilidad de uso. Si en un formulario utiliza un fondo gris y efectos en tres dimensiones y un fondo blanco en otro, parecerá que los formularios no están relacionados entre sí. Se debe adoptar un estilo y mantenerlo en toda la aplicación, incluso si ello implica volver a diseñar algunas características.

Facilidades: la forma sigue a la función.

Las *facilidades* son pistas visuales de la función de un objeto. Aunque el término no sea muy familiar, a su alrededor hay muchos ejemplos de ellas. El manillar de una bicicleta tiene hendiduras para colocar los dedos, una facilidad que hace obvio que ése es el sitio donde poner las manos. Los botones, los mandos y los conmutadores luminosos son todos facilidades; sólo con mirarlos puede deducirse su propósito.

La interfaz de usuario también hace uso de estas facilidades. Por ejemplo, los efectos tridimensionales utilizados en los botones de comando hacen pensar que están para ser presionados. Si diseñara un botón de comando con bordes planos, perdería esta facilidad y el usuario no tendría claro que es un botón de comando. Hay casos en que los botones planos pueden ser apropiados, como en juegos o en aplicaciones multimedia; está bien mientras mantenga la coherencia en toda la aplicación.

Los cuadros de texto también aportan un grado de facilidad: los usuarios esperan que un cuadro con bordes y un fondo blanco contenga texto modificable. Aunque es posible presentar un cuadro de texto sin bordes (`BorderStyle = 0`), lo haría parecer una etiqueta y para el usuario no sería obvio que es modificable.

Uso del espacio en blanco.

El uso del *espacio en blanco* dentro de la interfaz de usuario puede contribuir a resaltar elementos y aumentar su facilidad de uso. Aunque el espacio en blanco no tiene que ser necesariamente blanco, hace referencia al uso del espacio vacío entre y alrededor de los controles de un formulario. Si hay demasiados controles en un formulario puede parecer una interfaz congestionada, dificultando la localización de un campo o control concreto. Hay que incorporar espacio en blanco al diseño para hacer énfasis en los elementos del diseño.

El espaciado coherente entre los controles y la alineación vertical y horizontal de los elementos hacen que el diseño sea también más sencillo. De igual forma que el texto de un periódico está dispuesto en columnas con el mismo espacio entre las líneas, una interfaz ordenada la hace más legible.

Visual Basic* proporciona varias herramientas que facilitan el ajuste del espaciado, la alineación y el tamaño de los controles. Los comandos **Alinear**, **Igualar tamaño**, **Espacio horizontal**, **Espacio vertical** y **Centrar en el formulario** se encuentran en el menú **Formato**.

Simplificación.

Quizás el principio más importante en el diseño de la interfaz sea la sencillez. Cuando se trata de aplicaciones, si la interfaz parece difícil, probablemente lo sea. Pensando un poco se puede crear una interfaz que parezca (y sea) fácil de usar. Además, desde un punto de vista estético, siempre es preferible un diseño limpio y sencillo.

Un error frecuente en el diseño de la interfaz es el de modelarla a partir de objetos del mundo real. Imagínese, por ejemplo, que se pide que se cree una aplicación para completar impresos de seguros. La reacción natural sería diseñar una interfaz que duplicara de forma exacta el impreso en papel. Esto crea varios problemas: la forma y las dimensiones de un impreso en papel son diferentes a las de una pantalla, la duplicación de un impreso le limita al uso de cuadros de texto y casillas de verificación, y el usuario no obtiene un beneficio real.

Es mucho mejor diseñar una nueva interfaz, proporcionando quizás un duplicado impreso (con presentación preliminar) del impreso original en papel. Agrupando los campos de forma lógica y utilizando una interfaz con fichas o varios formularios enlazados puede presentar toda la información sin tener que hacer desplazamientos de pantalla. También se puede utilizar controles adicionales, como cuadros de lista con opciones predefinidas, lo que reduce la cantidad de datos que tiene que escribir el usuario.

La mejor prueba de la sencillez de uso de una aplicación es observar la aplicación en uso. Si un usuario típico no puede realizar inmediatamente una tarea deseada sin ayuda, quizás debería volverse a diseñar.

Uso del color y de imágenes.

El uso del color en una interfaz puede aportar atractivo visual, pero es fácil abusar de ello. Con tantos monitores capaces de presentar millones de colores, se tiene la tentación de utilizarlos todos. El color, como todos los demás principios básicos de diseño, puede ser problemático si no se tiene en cuenta en el diseño inicial.

Las preferencias de color varían mucho; puede que los gustos de los usuarios no coincidan con los suyos. El color puede evocar emociones y si el destino del diseño es internacional, ciertos colores pueden tener un significado cultural. Normalmente, lo mejor es ser conservador, y emplear colores más suaves y neutros.

Por supuesto, la elección de los colores también puede estar influida por los destinatarios de la aplicación, así como por el tono o el estado de ánimo que intente transmitir. Los rojos, verdes y amarillos brillantes pueden ser apropiados para aplicaciones infantiles, pero en raras ocasiones provocarían una impresión de responsabilidad fiscal en una aplicación bancaria.

Se pueden utilizar pequeñas cantidades de colores brillantes para hacer énfasis y llamar la atención sobre un área importante. Como norma, se debe limitar el número de colores en una aplicación y el esquema de colores debe ser coherente. Lo mejor es adoptar la paleta estándar de 16 colores si es posible; aunque los medios tonos puede hacer que algunos colores desaparezcan cuando se presentan en un monitor de 16 colores.

Otra consideración en cuanto al uso del color es la ceguera a ciertos colores. Mucha gente no es capaz de diferenciar entre distintas combinaciones de colores primarios como rojo y verde. Para alguien con esta limitación, el texto rojo sobre un fondo verde no sería visible.

Imágenes e iconos.

El uso de imágenes e iconos también puede agregar interés visual a la aplicación pero, de nuevo, es esencial un diseño cuidadoso. Las imágenes pueden transmitir información sin necesidad de incluir texto, pero las imágenes son percibidas de manera diferente por personas diferentes.

Las barras de herramientas con iconos que representen distintas funciones son un elemento útil de una interfaz, pero si el usuario no puede identificar la función representada por el icono, pueden ser contraproducentes. Al diseñar los iconos de una barra de herramientas, se debe tener en cuenta en otras aplicaciones y las normas estándar ya establecidas. Por ejemplo, muchas aplicaciones utilizan una hoja de papel con una esquina doblada para representar el icono Nuevo archivo. Puede que haya una mejor metáfora para esta función, pero si se representa de forma diferente, podría provocar confusión en el usuario.

También es importante considerar el significado cultural de las imágenes. Muchos programas utilizan una imagen de un buzón de estilo rural con una bandera para representar las funciones de correo. Se trata de un icono principalmente norteamericano; los usuarios de otros países o culturas probablemente no lo reconozcan como buzón.



Un icono que representa un buzón

Quando se diseñe iconos e imágenes, procúrese que sean sencillos. Las imágenes complejas con muchos colores no se ven bien cuando se presentan como iconos de 16 por 16 píxeles de las barras de herramientas o cuando se presentan con resoluciones de pantalla superiores.

Elección de fuentes.

Las fuentes también son una parte importante de la interfaz de usuario, ya que con frecuencia comunican información importante al usuario. Se tiene que elegir las fuentes que sean fácilmente legibles a resoluciones diferentes y en diferentes tipos de monitores. Lo mejor es adoptar fuentes sencillas con o sin remate siempre que sea posible. Script y otras fuentes decorativas generalmente quedan mejor impresas que en una pantalla y pueden ser difíciles de leer con tamaños en puntos pequeños.

A menos que piense distribuir fuentes junto con su aplicación, debe ceñirse a las fuentes estándar de Windows como Arial, New Times Roman o System. Si el sistema del usuario no incluye una fuente concreta, el sistema la sustituirá, dando como resultado una apariencia completamente distinta de la pretendida. Si el destino de su aplicación es internacional, tendrá que investigar las fuentes disponibles en los idiomas de destino. Además, deberá tener en cuenta la expansión del texto cuando diseñe para otros idiomas; las cadenas de texto pueden ocupar hasta un 50% de espacio más en otros idiomas.

De nuevo, es importante la coherencia del diseño a la hora de elegir las fuentes. En la mayor parte de los casos, no deberá utilizarse más de dos fuentes con dos o tres tamaños en puntos distintos dentro de una misma aplicación. Demasiadas fuentes pueden hacer que su aplicación parezca una nota de petición de rescate.

Resumen.

La POO llama Propiedades a las características, Métodos a las acciones y Eventos a las reacciones a los estímulos externos provenientes de otros objetos o del ambiente que les rodea. Tanto las Propiedades como los Métodos están envueltos en la entidad llamada Objeto. A esto se le llama Encapsulamiento u Ocultación, dado que el código inherente a cada uno está fuera de la vista del ambiente de desarrollo, bajo la ideología de "cada quien sus cosas".

La comunicación entre estos objetos se logra mediante interfases, las cuales proporcionan un medio que permite la interacción entre estos objetos. Todos aquellos que se encuentren bajo una misma interfase son polimórficos. El polimorfismo describe el efecto de poder invocar al método de un objeto, sin conocer exactamente el tipo de objeto, con lo que se logra que se ejecute la acción apropiada de ese objeto en particular.

Finalmente, la Herencia es la capacidad de poder crear otros objetos a partir de otro. La herencia es un punto clave en la POO y de suma importancia para el desarrollo de sistemas basados en POO. Microsoft Visual C++, Borland Delphi y Java son claros representantes de esta ideología. VB queda, aunque no del todo, fuera de esta clasificación.

- Puntos en contra de la POO.

El soporte de la POO para el desarrollo de módulos de software tipo plug-and-play se aplica sólo a nivel del código fuente; no es un estándar binario. Esto significa que no existe ningún mecanismo estándar a través del cual los objetos puedan trabajar juntos. Los módulos de desarrollo individuales podrían requerir sus propias interfases en código fuente y así, ejercer una carga muy fuerte en tiempo y costos de desarrollo. No sólo es necesario considerar las diferentes interfases; además, la selección del lenguaje de programación se convierte en un obstáculo potencial. Un desarrollador podría desear utilizar un objeto escrito en un lenguaje no familiar, lo que requerirá, por tanto, aún más inversión en la curva de aprendizaje.

El hecho de que la compatibilidad está limitada a nivel del código fuente trae consigo que la POO no enfrenta el importante problema de las revisiones. Cuando un objeto es actualizado, revisado y reemplazado, cualquier programa que lo utilice necesita volver a compilarse y probablemente a distribuirse. Mientras que los programadores están ya acostumbrados a compilar y enlazar de nuevo cada vez que actualizan una aplicación, los problemas de la redistribución elevan el nivel de sus preocupaciones. Por ejemplo, esto trae la desagradable posibilidad de que las DLLs se sobrescriban unas a otras. Ciertamente, el problema de la compatibilidad de software se resuelve parcialmente gracias a la POO, pero ésta no es una respuesta completa, pero si ante todo ya se usan lenguajes como Visual C++ o Delphi, pues parte del problema está resuelto.

- Visual Basic

Desde que Visual Basic vio la luz a principios de esta década, ha sido un lenguaje basado en objetos. Esto es, se crean aplicaciones usando objetos previamente creados como formularios, controles y otros del sistema (como App, Clipboard, Printer, etcétera). Bajo esta ideología, para usar un cuadro de texto en un programa, lo único que hay que hacer es seleccionarlo desde un cuadro de herramientas (que concentra a todos los objetos o controles de este tipo) y colocarlo en un formulario, para después manipular sus propiedades y métodos, de modo que se pueda personalizar su comportamiento y apariencia.

Esto hace que el desarrollo de aplicaciones sea mucho más sencillo que antes, dado que el diseño de interfaces y el desarrollo de tareas se reduce significativamente mediante el uso de objetos prediseñados, que funcionan en una ideología de Conectar y usar (una verdadera implantación de la tecnología plug and play) que otros lenguajes empezaron a aprovechar posteriormente, como Delphi y Visual C++. Sin embargo, la industria de las computadoras no considera necesariamente a los lenguajes de programación basados en objetos como lenguajes de Programación orientada a objetos. Esto es porque la POO se refiere a mucho más que tan sólo usar objetos existentes durante el desarrollo.

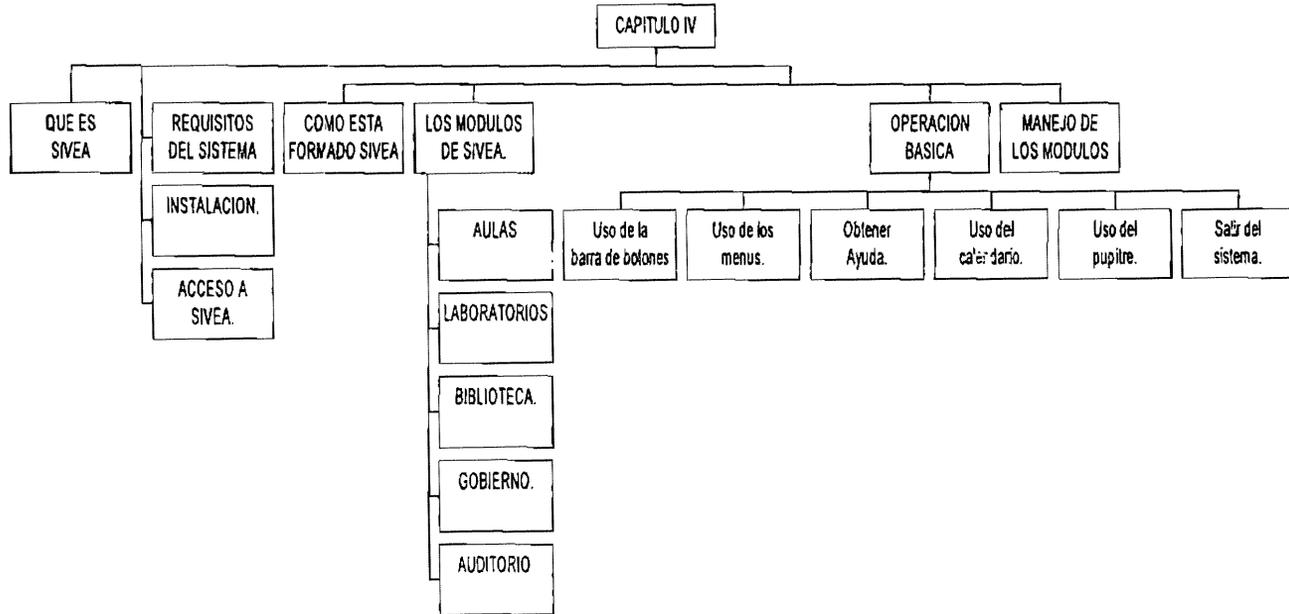
La POO se refiere a una cierta cantidad de principios de diseño, así como a la utilización de una serie de técnicas de desarrollo. En el anexo C de este trabajo de investigación se da una introducción a la programación de esta poderosa herramienta POO.

Para concluir, la programación moderna no es más que el reflejo de las nuevas necesidades creadas a partir de los nuevos entornos gráficos multitareas que no eran tan populares hace menos de ocho años. Aunque la POO existe desde hace varios años, su verdadera utilidad se está haciendo patente a últimas fechas y aún más con la liberación de las primeras versiones de OLE en Windows 3.1 allá por 1991. Sin menoscabo de esto, el entender con propiedad las directivas de la programación moderna permitirá hacer más de lo cotidiano algo simple y productivo.

CAPITULO IV

OPERACIÓN DE SIVEA Y SUS MODULOS.

OPERACIÓN DE SIVEA Y SUS MODULOS.





SIVEA.

Contenido.

Que es SIVEA.

Requisitos del sistema.

Instalación de SIVEA.

Acceso a SIVEA

Cómo está formado SIVEA.

Los módulos de SIVEA.

Operación básica.

Manejo de los módulos.

• Sistema Virtual de Enseñanza – Aprendizaje en Ingeniería básica.

El Sistema Virtual de Enseñanza- Aprendizaje o SIVEA por sus siglas es un sistema informático cuya función es la de ser una herramienta de enseñanza aprendizaje en ingeniería básicamente aunque su campo de aplicación puede ser expandido a otras áreas de la educación. Fue creado a partir de una necesidad de una herramienta para ayudar a los alumnos de primer ingreso a facilitar su aprendizaje y por complemento facilitar la labor docente.

Se concibió a partir de la experiencia como alumno y más adelante como facilitador y docente en la cual me he desempeñado.

• Requerimientos del sistema.

El sistema requiere los siguientes componentes:

SOFTWARE:

Windows 95* o posterior.

Tener instalado Office 95* o cualquier versión posterior con los siguientes paquetes instalados:

Procesador de textos WORD*.

Paquete para presentaciones POWERPOINT* y su visor de presentaciones.

Excel* (Recomendado).

Access*. (Recomendado).

NOTA WINDOWS, WORD, OFICCE, EXCEL, POWERPOINT y ACCESS son marcas registradas por MICROSOFT. CO. Y para su uso se deberá tener la respectiva licencia

HARDWARE:

Computadora de escritorio o portátil con:

Procesador PENTIUM o mejor.

32 megas en RAM (64 o 128 es mucho mejor)

Por lo menos 100 megas disponibles en disco duro.

Multimedia.

1 mega en video.

Resolución de 800 x 600 (en resoluciones mas bajas las pantallas apareceran incompletas).

Mouse , teclado.

• Instalación de SIVEA.

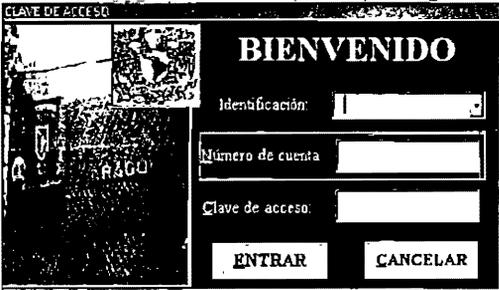
Para la instalación de SIVEA se deberá insertar el CD con el sistema y se debe ejecutar el programa de instalación que se encuentra en el directorio raíz del CD. A continuación se deberá seguir los pasos que indica el programa.

En el momento de seleccionar el directorio a instalar tendrá que ser en el directorio raíz y con la ruta indicada por el programa, es decir deberá instalarse en el directorio "C:\SIVEA" de lo contrario el programa puede no funcionar correctamente y fallará en algunas de sus aplicaciones.

Una vez que se ha terminado la instalación se recomienda reiniciar el equipo con el fin de que el sistema pueda concretar su proceso de instalación.

• Acceso a SIVEA

Para el acceso al sistema deberá buscarse el icono de nombre SIVEA y seguido de esto dar doble clic en este icono, después de algunos segundos aparecerá una ventana solicitando su clave de acceso, así como también su número de cuenta o RFC y tipo de usuario.



The image shows a login window for SIVEA. The title bar reads "CLAVE DE ACCESO". The main heading is "BIENVENIDO". Below the heading, there are three input fields: "Identificación:", "Número de cuenta", and "Clave de acceso:". At the bottom, there are two buttons: "ENTRAR" and "CANCELAR". The background of the window is dark and textured.

En SIVEA se tienen tres tipos de Usuarios:

Alumno: Es aquel que tiene acceso parcial al sistema, podrá entrar a las aulas, consultar clases, realizar ejercicios, tareas y exámenes, tiene derecho de acceso a la biblioteca y a los laboratorios; pero no tiene acceso al módulo de gobierno, es decir no puede generar clases ni modificar a los usuarios o los contenidos del sistema.

Facilitador: Tiene todos los permisos del alumno, es decir tiene acceso a cualquier módulo, puede generar clases, tareas, contenidos, prácticas, evaluaciones, etc.; pero no puede crear, modificar o eliminar usuarios.

Director: Tiene acceso a todo el sistema, puede crear, modificar y eliminar usuarios del sistema.

NOTA: Si usted no cuenta con una clave de usuario deberá ser registrado ya que de lo contrario no se podrá tener acceso al sistema de ninguna otra forma. (Para conseguir una clave solicítela al administrador del sistema).

En el caso de ser alumno se le pedirá el número de cuenta y de ser facilitador o director deberá dar su RFC con el que fue registrado el sistema.

CLAVE DE ACCESO

BIENVENIDO

Identificación: ESTUDIANTE

Número de cuenta: 8914006-4

Clave de acceso:

ENTRAR CANCELAR

BIENVENIDO

Identificación: FACILITADOR

RFC: RORITJ1104

Clave de acceso:

ENTRAR CANCELAR

Ya que se han tecleado los datos de forma correcta seleccionar el botón de ENTRAR, si se ha hecho de forma correcta el sistema mostrara una pantalla de inicio, de lo contrario se tendrá otra oportunidad para teclear los datos.

NOTA: El sistema permite teclear tres veces los datos, después de esto se bloquea y no podrá entrar a este.

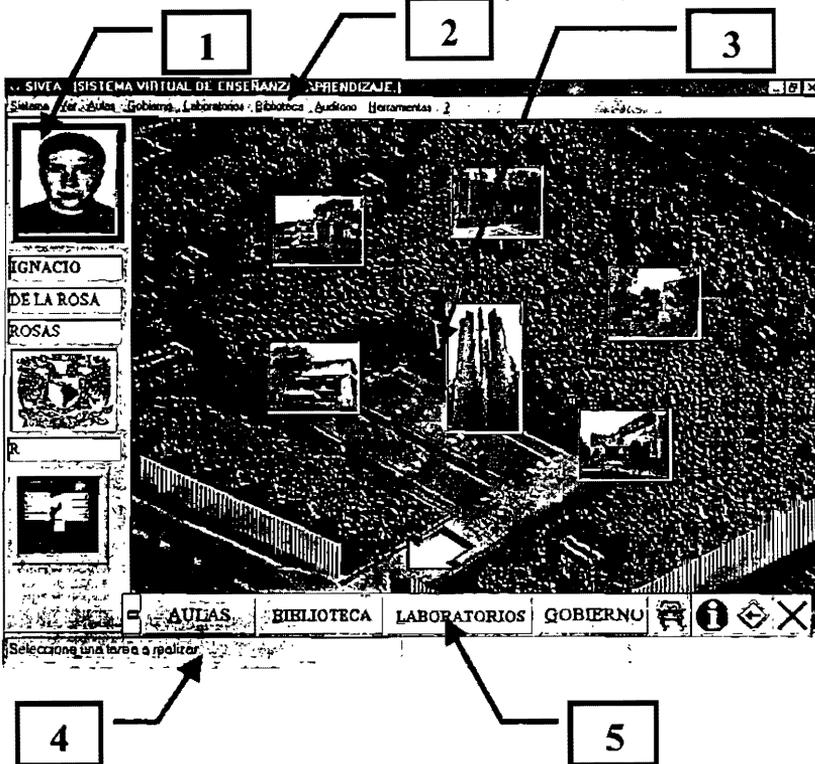
• Cómo está formado SIVEA.



SIVEA es un sistema multimedia de fácil manejo, se puede operar mediante el uso del mouse o mediante el teclado utilizando las opciones en la barra de menús. El sistema es amigable y las pantallas son de fácil comprensión, se ha buscado la sencillez en las explicaciones y operación del sistema de tal modo que se pueda utilizar sin necesidad de un tutor presente.

La operación del sistema se hace de igual forma que cualquier otro paquete que corra bajo el entorno de WINDOWS* por lo que sí se está familiarizado con el uso de ventanas, botones y cajas de diálogo se podrá utilizar SIVEA sin ningún problema.

La pantalla principal de SIVEA se compone de los siguientes componentes:



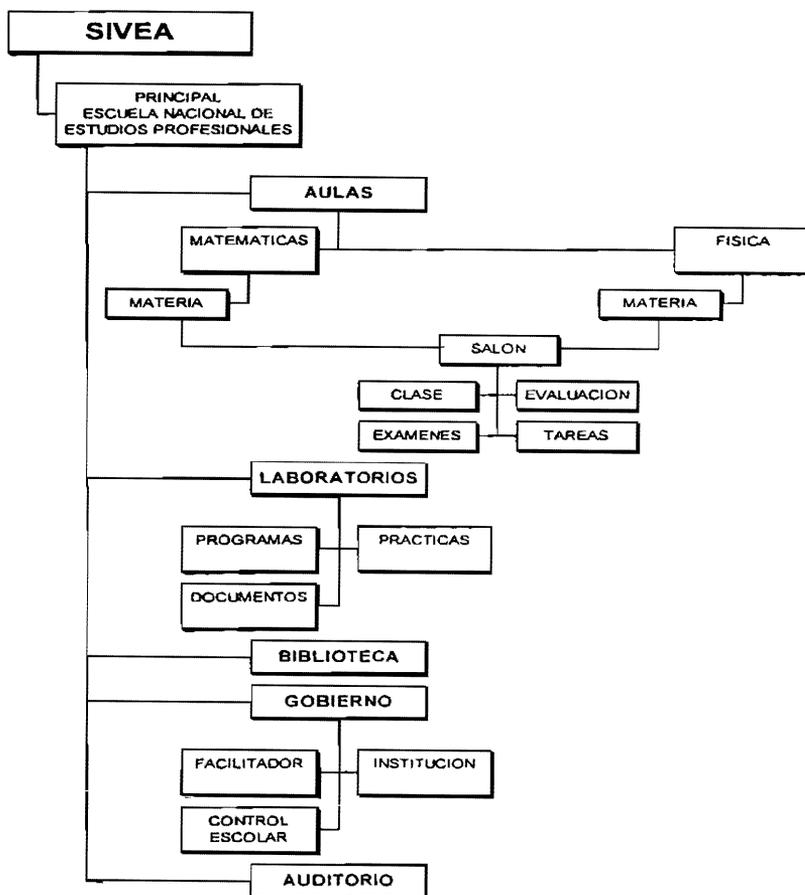
1. BARRA DE DATOS.- En esta barra se muestran los datos del usuario, como son : nombre, RFC o número de cuenta y su fotografía; Así como también el botón de ayuda.
2. BARRA DE MENUS.- Aquí se muestran los menús que se pueden acceder con el teclado o con el mouse.
3. AREA DE TRABAJO. En esta zona de la pantalla se muestran todos los datos, formas y gráficos del sistema. Para ejecutar algún comando basta con hacer clic sobre la imagen que presente la pantalla.
4. ZONA DE MENSAJES. En esta zona se muestran los mensajes de apoyo al usuario.

* NOTA WINDOWS, WORD, OFFICE, EXCEL, POWERPOINT y ACCESS son marcas registradas por MICROSOFT. CO. Y para su uso se deberá tener la respectiva licencia.

5. BARRA DE BOTONES. Esta barra muestra los botones para ejecutar comandos que se encuentran en la barra de menús, los botones que hay en la barra de botones cambian de acuerdo a la pantalla en la que se encuentre trabajando.

A continuación se presenta un mapa con los módulos del sistema.

Diagrama de bloques de SIVEA.



Cuando se entra al sistema se visualiza el módulo principal denominado "ESCUELA PROFESIONAL DE ESTUDIOS PROFESIONALES", ahí se observan los módulos principales del sistema, los cuales se explican a continuación.

• Módulos.

El sistema se compone de cinco módulos que son:

- Aulas.**
- Laboratorios.**
- Biblioteca.**
- Gobierno.**
- Auditorio.**

Para acceder a cualquiera de estos módulos basta con dar un clic sobre la imagen que se despliega sobre el área de trabajo o dar clic sobre la barra de botones en el botón que lleva el nombre del módulo o seleccionar de la barra de menús el módulo deseado.

A continuación se explica que contiene cada uno de los módulos.



AULAS

En el módulo de aulas se tiene acceso a las dos materias disponibles que son matemáticas y física. A este módulo pueden acceder cualquier usuario y en él se pueden trabajar: las clases, los ejercicios, evaluaciones, etc.



LABORATORIOS.

En los laboratorios se tienen tres opciones: programas, documentos y prácticas. Todos los usuarios tienen acceso a este módulo.



BIBLIOTECA.

En este módulo se pueden consultar documentos de texto, presentaciones o cualquier documento que pueda ser leído mediante los controles **ACTIVEX**™. Todos los usuarios tienen acceso a este módulo.



GOBIERNO.

En este módulo se controla el sistema, se puede dar el mantenimiento a usuarios, clases, laboratorios, biblioteca y asignar claves para usuarios. Solo los facilitadores y directores podrán entrar a este módulo.



AUDITORIO

En este módulo se ve una presentación acerca del trabajo de tesis realizado de este sistema. Todos los usuarios tienen acceso a este.

** NOTA ACTIVEX es marcas registradas por MICROSOFT. CO. Y para su uso se deberá tener la respectiva licencia.*

• Operación básica.

- **Uso de la barra de botones.**
- **Uso de los menús.**
- **Obtener Ayuda.**
- **Uso del Calendario.**
- **Uso del pupitre.**
- **Salir del sistema.**

- **Uso de la barra de botones.**

Descripción de la barra.

La barra de botones es una herramienta que permite moverse a lo largo del sistema y seleccionar los comandos y funciones que tiene el sistema. Esta barra siempre esta visible y se mostrara sobre todas las otras ventanas que se tengan activas(Incluso ventanas que no sean del sistema).



Dependiendo del módulo que se encuentre activo se tendrá visible los botones de comando de ese respectivo módulo, aunque se cuentan con cinco botones que son generales, es decir siempre estarán presentes para la barra de botones.

- 
Botón de movimiento de barra de botones.
 Este al ser presionado despliega la barra de titulo de esta ventana lo que permite moverla de lugar a donde se desee. Una vez que se ha colocado en el lugar deseado , hacer clic de nueva cuenta en este botón para fijar la barra de botones.

- 
Botón de salir de SIVEA.
 Este al ser presionado despliega la ventana de termino de sesión, permite salir del sistema. (Para mas detalle ver el tema *de salir del sistema.*).Puede ser utilizado en cualquier parte del sistema, excepto cuando se encuentra dentro del salón de clases localizado en el módulo de AULAS. Si se desea salir de SIVEA ,se deberá primero salir del salón para después salir del sistema.

- 
Botón ACERCA DE.
 Este al ser presionado muestra la ventana de "ACERCA DE SIVEA" donde muestra los créditos del sistema, así como también se puede ver los parámetros de la computadora en la cual se esta trabajando.



Botón REGRESO.

Este botón permite el regreso de una pantalla a la anterior, es decir si se entra a algún módulo, por ejemplo AULAS y se quiere regresar al módulo PRINCIPAL, basta con hacer clic en este botón para realizar esta acción. Este botón puede ser utilizado para regresar de cualquier ventana o módulo en el que se encuentre trabajando. Si se encuentra en el módulo PRINCIPAL y se presiona este botón deberá aparecer la ventana de finalización de sesión en SIVEA.



Botón de cerrar barra de botones.

Este al ser presionado este botón oculta la barra de botones. Si se desea que vuelva a ser visible, se tendrá que seleccionar la opción de "barra de botones visible", que se encuentra en el menú de ver. Para más detalle ver el tema de uso de menús.

Utilizando la barra de botones.

Se puede manejar el sistema utilizando únicamente la barra de botones y el mouse esto facilita su uso, para esto se deberá presionar el botón de la opción deseada hasta llegar a la opción buscada a continuación se muestra un mapa de acceso de la barra de botones.

o Uso de los menús.

Los menús de SIVEA permiten el acceso a los módulos y funciones usando el teclado, basta con presionar la tecla de ALT y la letra que se encuentra subrayada en la palabra de cada uno de los menús. Por ejemplo para acceder al menú de *sistema* se puede teclear ALT y la letra S y se desplegará el menú

Los menús en SIVEA se muestran a continuación.



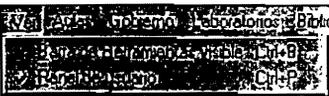
Los menús se muestran enseguida y la explicación de cada opción se revisa posteriormente en cada una de las explicaciones de su respectiva pantalla o módulo

MENU SISTEMA

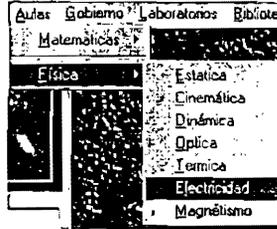
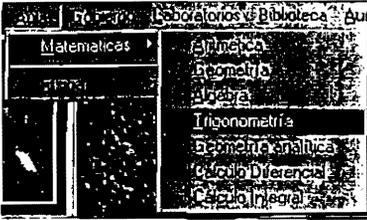


Por ejemplo la explicación de la opción salir de SIVEA se da en el tema "SALIR DEL SISTEMA" esta tema incluido en este documento mas adelante.

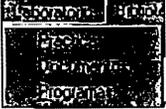
MENU DE VER



MENU DE AULAS



MENU LABORATORIOS



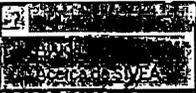
MENU DE BIBLIOTECA Y AUDOTORIO



MENU HERRAMIENTAS



MENU AYUDA



□ **Obtener Ayuda.**

Para obtener ayuda sobre el sistema se puede hacer de varias formas:

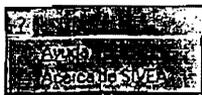
En el sistema:

Dentro del sistema se puede acceder a este archivo de ayuda por cualquiera de estas formas:



En la barra de datos, hacer clic en la imagen con el signo de interrogación.

En la barra de menús hacer clic en el signo de interrogación y después seleccionar la opción de Ayuda.



Fuera del sistema:

Consultando el manual: El manual de usuario del sistema se incluye en este trabajo de tesis como el capítulo cuatro y explica todo lo relacionado al uso y manejo de este.

Consultando soporte técnico: Se puede utilizar el apoyo de un programador o experto en sistemas el cual este familiarizado con el manejo de objetos, bases de datos y programación orientada a objetos.

□ **Uso del Calendario.**

El calendario es una herramienta que permite visualizar los días, meses y años mediante una ventana , esto puede ayudar a programar fechas, establecer tiempos de estudio y fechas de realización de evaluaciones y tareas

□ **Uso del Pupitre.**

El pupitre es una herramienta que combina las principales herramientas básicas en WINDOWS que son.

Procesador de texto (BLOC DE NOTAS)

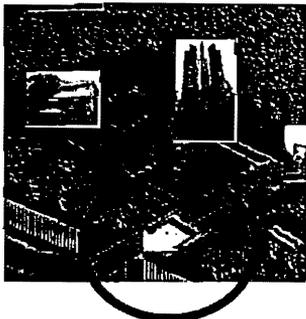
Calculadora.

Paquete de realización de imágenes y dibujos.(PAINT).

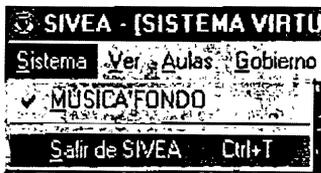
Para ejecutar cualquiera de estas herramientas, basta con hacer clic en el icono que representa la herramienta:

□ **Salir del sistema.**

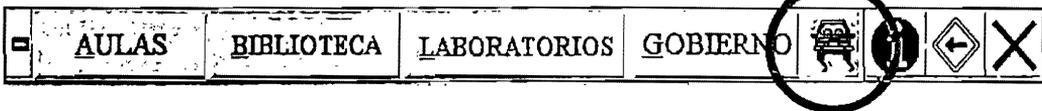
Para salir del sistema en cualquier momento se puede hacer de varias formas:
Hacer clic en la flecha de salida en la pantalla principal .



En la barra de menús hacer clic en sistema y dar clic en "Salir de SIVEA" o con las teclas CTRL+ T



En la barra de botones hacer clic en el botón del automóvil.



Inmediatamente después se vera una ventana de confirmación de salida del sistema:



Si se teclaea "S" o se da clic en "SI" se dará por terminada la sesión. Si no se desea finalizar sesión dar clic en "NO" o teclrear "N".

• Manejo de los módulos.

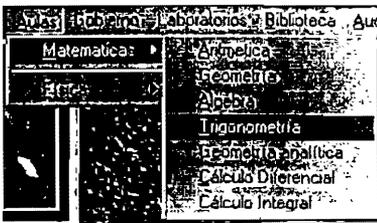
- Aulas.
- Gobierno.
- Laboratorios.
- Biblioteca.
- Auditorio.

- **AULAS.**

El módulo de aulas es el módulo en el cual el alumno tendrá una mayor interacción por lo cual se da una explicación completa a continuación. Para acceder a este módulo se hace de cualquiera de estas formas:

En la barra de menús seleccionar la opción de aulas y el salón al cual se desea acceder, ejemplo.

Para acceder al salón de trigonometria hacer clic como se muestra en a imagen:



O con el teclado la secuencia:
Alt + A (Aulas) , M (Matemáticas) , y T (Trigonometría)

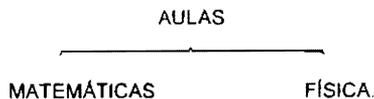
También se puede realizar en la barra de botones haciendo clic en el botón de aulas:



O haciendo clic en el área de trabajo en el icono con el aula



Una vez que se accede al módulo de aulas las dos opciones que se tienen son:



Cada materia a su vez se divide en los siguientes temas:

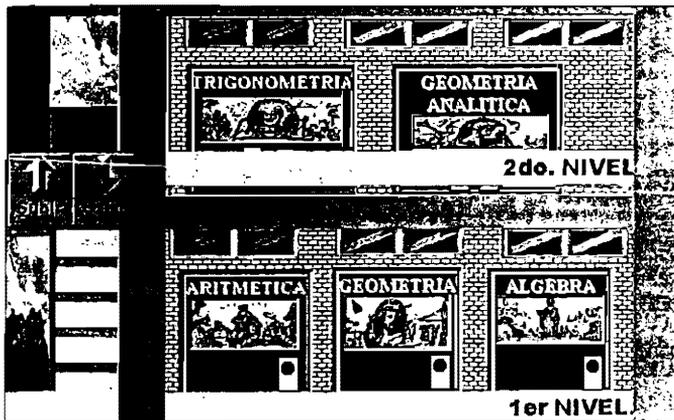
MATEMÁTICAS	FÍSICA
ARITMÉTICA	ESTÁTICA
GEOMETRÍA	DINÁMICA
ÁLGEBRA	CINEMÁTICA.
TRIGONOMETRÍA	ÓPTICA
GEOMETRÍA ANALÍTICA	TÉRMICA
CÁLCULO DIFERENCIAL	ELECTRICIDAD.
CÁLCULO INTEGRAL.	MAGNETISMO.

Para acceder a la materia mover el mouse hasta seleccionar cualquiera de los dos edificios.



También se puede hacer clic en la barra de botones y seleccionar el tema que se desea consultar.

Una vez seleccionada la materia (Matemáticas o física) aparecerá la siguiente pantalla:



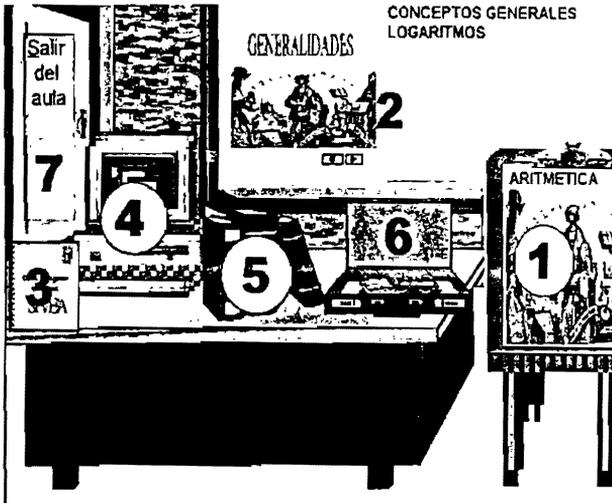
El edificio del tema se compone de los siguientes elementos:

1. Botones de subir y bajar de nivel. Cada edificio (Materia) consta de 3 niveles considerando la dificultad o el grado del tema cuando se presiona estos botones se podrá subir o bajar de los niveles y observar los temas que cubre cada nivel.

Esto también puede hacerse usando la barra de desplazamiento para movimiento mas lento en cuanto los niveles en pantalla.

2. Puertas del tema. Cuando se hace doble clic en cualquiera de las puertas se tendrá acceso al tema de interés.

Ya seleccionado el tema se vera una pantalla la cual es llamada EL SALON de clases y esta formado por los siguientes elementos:



1. ROTAFOLIO. Muestra el tema que se esta observando en el momento.
2. PIZARRON. Muestra una lista con los subtemas disponibles y una muestra del subtema que esta activado en este momento, para su acceso inmediato; si se quiere acceder a este subtema hacer doble clic en la imagen de este subtema en el pizarrón y esperar unos segundos (NOTA: si el tiempo de espera se extiende mas de unos cuantos segundos, se recomienda reinicializar el sistema).

Si se desea cambiar el subtema, se deberá seleccionar de la lista de temas disponibles en el pizarrón

3. LA GUIA . Este es un archivo el cual muestra información de ayuda sobre las clases (Que son, como se acceden a ellas, o como salir de estas).Para acceder a este archivo hacer un doble clic en este icono.
4. COMPUTADORA. Este es un icono que permite acceder a GENEVAL en su módulo de ejercicios y solo se podrá realizar ejercicios de práctica cuando se entre a este sistema por medio de este elemento.
5. LIBROS: En este icono se puede ver la lista de tareas asignadas por el facilitador para este tema se podrá mandar respuestas de las tareas y revisión de las asignadas.
6. PORTAFOLIO DEL FACILITADOR. En este se encuentran las evaluaciones del tema y al acceder a este con un doble clic pasara a GENEVAL en su modo de aplicación de evaluaciones (GENEVAL es un sistema adicional a SIVEA para la realización de evaluaciones y cuya ayuda se incluye en el sistema).
7. PUERTA DE SALIDA. Si se da un clic en este icono se saldrá del SALON y se regresara al módulo de AULAS.

□ GOBIERNO

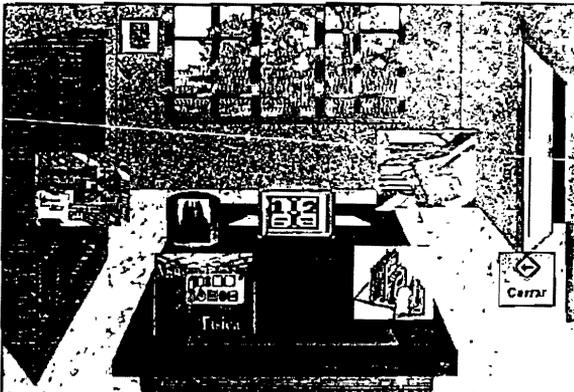
Este módulo es muy importante en el sistema ya que es el que permite el control y mantenimiento de este . A este solo tendrá acceso el personal que este autorizado por el director y los alumnos no tienen acceso a este, para otorgar claves de acceso a los facilitadores y administradores del sistema se deberá hacer mediante el uso del módulo de control escolar. A continuación se da una explicación de sus funciones



- Controlar y dar mantenimiento a los usuarios del sistema (dar de alta, de baja , modificar datos y otorgar claves).
- Controlar y dar mantenimiento al módulo de aulas mediante el uso del módulo *oficina del facilitador*.
- Controlar y dar mantenimiento a los módulos de laboratorios y biblioteca.

Enseguida se desarrolla de manera mas amplia la explicación de cada elemento.

Oficina del facilitador.



Este módulo da acceso a la oficina del facilitador que permite editar clases, generar y revisar evaluaciones, generación y revisión de tareas, Observar listados de alumnos y ver el calendario. La forma de acceder a cada una de estas funciones es dando un clic en el componente que representa la función. Para salir de este módulo hacer clic en el botón de *Cerrar*.

A continuación se muestra la función de cada uno de los elementos:



Generar Clase.

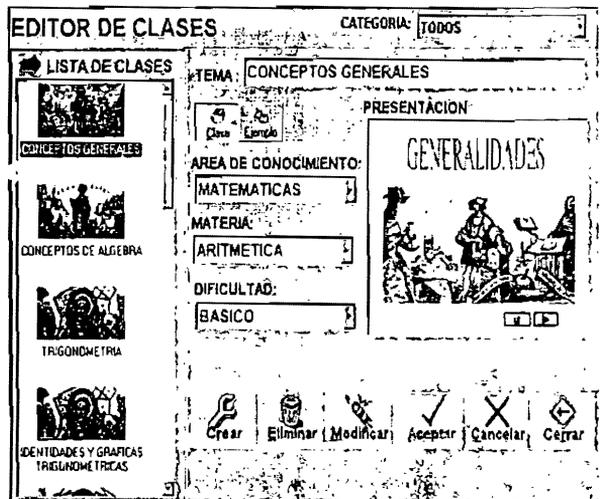
El generador de clases permite crear, eliminar y editar las clases que serán mostradas en el módulo de aulas en el salón .

Este esta formado por las siguientes partes:

Lista de clases: Muestra todas las clases de todos los temas que se tengan cargados, cuando se da clic en el área de categoría se puede cambiar la vista de estas clases en :
Iconos grandes,
Iconos pequeños
Y detalles de cada clase.

Para expandir o contraer el tamaño de la lista presionar el botón con el símbolo de la flecha y la lista cambiara de dimensiones.

Área de despliegue de tema. En esta se muestra el título del tema seleccionado de la lista de clases una imagen de la presentación, el área de conocimiento a la que pertenece (física o matemáticas) , la materia (por ejemplo: trigonometría, Aritmética, Electricidad, etc.) y la dificultad del tema.



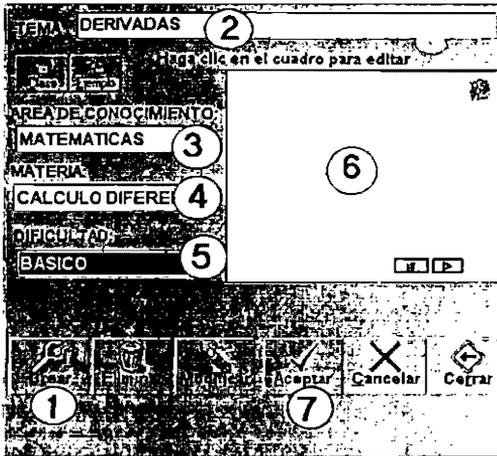
Cabe mencionar que las presentaciones están realizadas en el paquete PowerPoint* y se deberá tener licencia para poder utilizar este sistema.



Si se desea consultar acerca de que es una clase y como se conforma una clase en SIVEA, se recomienda revisar la presentación Guía de clases, que se puede observar en el módulo de aulas en el salón cuando se tiene acceso a cualquier tema.

Como se crea una clase.

Para crear una clase se deberá estar familiarizado con la creación de presentaciones en *PowerPoint** y después seguir los siguientes pasos:



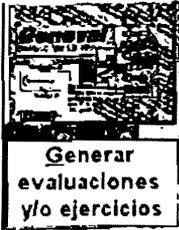
1. Presionar el botón de Crear .
2. Escribir el nombre del tema de la clase (por ejemplo: Derivadas), en la caja de texto de *TEMA*.
3. Determinar el área de conocimiento(Para este ejemplo será Matemáticas), en el listado de *AREA DE CONOCIMIENTO*
4. Seleccionar la materia del tema en el listado de *MATERIA*. (La materia sería Calculo diferencial para el ejemplo).
5. Determinar la dificultad del tema (Básico , Medio o Avanzado).
6. Crear la presentación en *PowerPoint** (Se requiere tener el software instalado) o cargando una presentación ya realizada :

- a. Si la presentación no existe haciendo doble clic en el cuadro de formato en blanco. El formato en blanco permite agregar el Título ,crear hipervínculos a otras pantallas de la presentación y utilizar los botones al terminar guardar el archivo en disco duro e indicar el nombre
- b. Si ya se realizó previamente la presentación y solo quiere mandarla a llamar desde el disco duro presionar el botón derecho del mouse sobre la presentación en blanco , y después se deberá indicar la *ruta de acceso del archivo*.

7. Finalmente presionar el botón de **Aceptar**..

Otras operaciones con clases.

- Para eliminar una clase dar clic en el botón **ELIMINAR** y confirmar la operación si se está seguro de esto.
- Si se desea modificar algún dato sobre la clase presionar el botón de **MODIFICAR**. Y cuando se termine la edición de la clase presionar el botón de **Aceptar**.
- Para cancelar cualquier operación presionar el botón **Cancelar**.



Generar evaluaciones (GENEVAL).

Permite crear, eliminar ,editar , imprimir y estudiar las evaluaciones mediante el programa GENEVAL que se incluye en el sistema

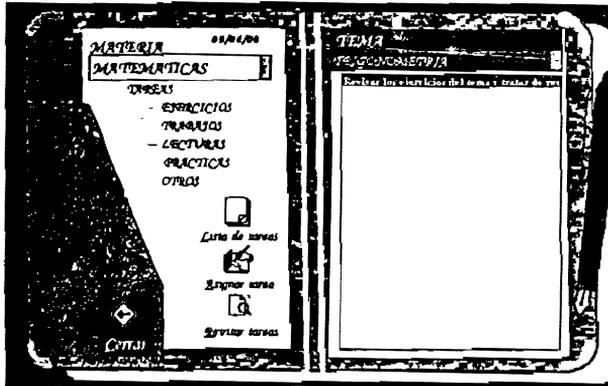
El programa GENEVAL es todo un sistema de generación de evaluaciones y el archivo de ayuda se anexa en este trabajo en el ANEXO E de este sistema y en programa se puede consultar al acceder al menú de ayuda del programa.





Generar tareas

Este módulo genera tareas para que sea realizadas por los alumnos, para su posterior revisión por parte del facilitador.



Las tareas se clasifican de acuerdo a las siguientes categorías:

- EJERCICIOS.
- TRABAJOS.
- LECTURAS.
- PRACTICAS
- Y OTROS: Aquellas tareas las cuales no entran en ninguna de las categorías anteriores.

Primero deberá determinarse la materia (FÍSICA o MATEMÁTICAS), seguido de esto hacer clic en el tema (por ejemplo trigonometría).

Después de esto se podrá escoger lo

que se desea hacer:

Para observar las tareas asignadas presionar el icono de *Lista de tareas*.

Título de la tarea
 Escriba aquí el título de la tarea.

Descripción de la tarea:
 Escriba aquí lo que el alumno deberá de realizar

Fecha de asignación:
 13/04/00

Fecha de entrega:
 13/04/00

Prioridad:
 ALTA MEDIA BAJA Sin Prioridad

Para agregar tareas al sistema hacer clic en el icono de *Asignar tareas*

Se debe escribir los datos de acuerdo a como se van requiriendo, colocando el título de la tarea, si tiene alguna explicación o datos relevantes que el alumno debe conocer en el cuadro de descripción. Después indicar la fecha de asignación y fecha de entrega. Si se va anexar un archivo que se requiera agregar a la tarea dar un clic en el botón de Insertar archivo, seguido de esto indicar la prioridad de esta tarea (baja, media, alta o sin prioridad), es decir al indicar la prioridad se deberá dar mayor importancia a una tarea con prioridad alta que a una sin prioridad o prioridad baja o media. Por último presionar el botón de *Aceptar* y la tarea será agregada a la lista de tareas por resolver.

Finalmente para revisar y retroalimentar las tareas que los alumnos han contestado hacer clic en el icono de *Revisar tareas*.

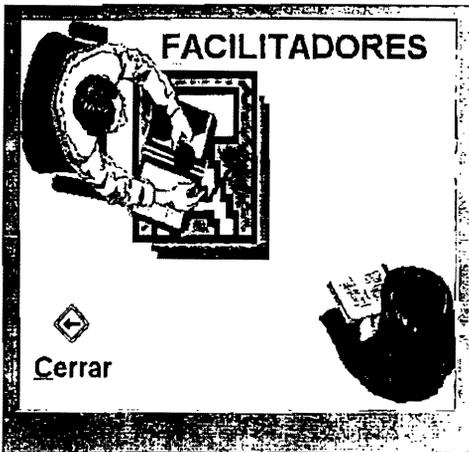


Ver listas de alumnos.

Al presionar este botón se tendrá acceso a los listados de los alumnos que se encuentran registrados en SIVEA, se podrá ahí consultar los datos de los alumnos, se recomienda ser discretos con la información y solo utilizarla para fines académicos.

DOCUMENTO		NOMBRE	APELLIDOP	APELLIDOM	Dirección

Control escolar



Este componente del módulo de GOBIERNO permite controlar los registros de los usuarios de SIVEA: Facilitadores y Alumnos.

Para acceder al control de facilitadores dar un clic en la imagen del facilitador y aparecerá el siguiente cuadro:



Si se desea observar a los facilitadores que están registrados dar clic en la opción **Lista de facilitadores**.

Para Ingresar, dar de baja o modificar datos de los facilitadores hacer clic en **Control de facilitadores**.

Cuando se accede al control de facilitadores se vera una pantalla como la que sigue:

FACILITADORES			
No. de RFC:	NOMBRE:	APELLIDOP:	APELLIDOM:
R	IGNACIO	DE LA ROSA	ROSAS
Fotografía:	Dirección:	Cotonia:	
	VALLE DEL MIXTECO # 140	VALLE DE	
	Ciudad:	Estado o Provincia:	Cód Postal:
	NEZAHUALCOYOT	MEXICO	57100
	NúmTeléfono:	NomCorreoElectrónico:	
	57803696	idelarosa@cedelmx.net.mx	
	Grado Académico:	Especialidad:	
	Ingeniería Mecánica - Eléctrica	Electrónica	
	Escuela de procedencia: ENEP ARAGON UNAM		

Si se desea ingresar un facilitador a SIVEA hacer clic en el botón de **Agregar**, seguido de esto escribir los datos del facilitador de acuerdo a como se van solicitando; para agregar la foto hacer clic en la imagen y buscar el archivo que contiene la foto (Se recomienda que la imagen exista previamente y que sea guardada en un formato de bit map bmp). Finalmente hacer clic en aceptar para confirmar la operación.

Si se desea eliminar al facilitador hacer clic en el botón **Eliminar** y después confirmar la operación.

Para modificar alguno de los datos del facilitador hacer clic en **Modificar**, realizar las modificaciones y dar clic en el botón de **Aceptar**.

Para cancelar cualquiera operación hacer clic en el botón de **Cancelar**.

Para cerrar esta pantalla hacer clic en **Cerrar**.

Para el control de alumnos hacer doble clic en la imagen del alumno.



Si se desea observar a los facilitadores que están registrados dar clic en la opción **Lista de alumnos**.

Para Ingresar, dar de baja o modificar datos de los facilitadores hacer clic en **Control de alumnos**.

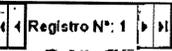
ALUMNOS

Cuando se accede al control de alumnos se vera una pantalla como la que sigue:

ALUMNOS

NO CUENTA: 000000000	NOMBRE: Juan	APELLIDOP: Perez	APELLIDOM: Gonzalez
Fotografía: 	Dirección: Calle x	Colonia: El buen rostro	
	Ciudad: Mexico	Estado o Provincia: D.F.	CódPostal: 30
	NúmTeléfono: 555555555	NumCorreoElectrónico:	
	BACHILLERATO: ENP 5	Especialidad: FISICO	SEMESTRE:






Si se desea ingresar un alumno a SIVEA hacer clic en el botón de **Agregar**, seguido de esto escribir los datos del alumno de acuerdo a como se van solicitando; para agregar la foto hacer clic en la imagen y buscar el archivo que contiene la foto (Se recomienda que la imagen exista previamente y que sea guardada en un formato de bit map bmp) Finalmente hacer clic en aceptar para confirmar la operación

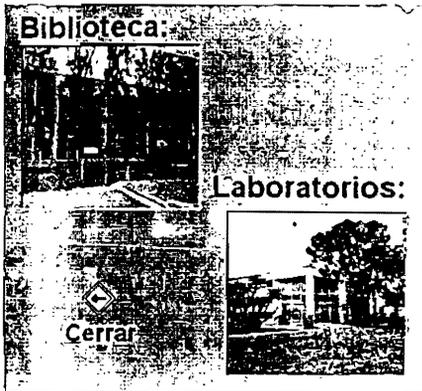
Si se desea eliminar al alumno hacer clic en el botón **Eliminar** y después confirmar la operación

Para modificar alguno de los datos del alumno hacer clic en **Modificar**, realizar las modificaciones y dar clic en el botón de **Aceptar**

Para cancelar cualquiera operación hacer clic en el botón de **Cancelar**.

Para cerrar esta pantalla hacer clic en **Cerrar**.

Institución.



Este elemento permite controlar la biblioteca y los laboratorios, es decir dar de alta, eliminar o modificar los elementos de estos dos módulos.

A continuación se explica cada uno de estos elementos:

Si se desea acceder al elemento de biblioteca hacer clic en la imagen de esta o en caso de los laboratorios hacer lo mismo en la imagen respectiva.

Una vez seleccionado el elemento se vera la siguiente pantalla:

Biblioteca:

Si se selecciono la imagen de biblioteca aparecerá la ventana de control de biblioteca

Para agregar un documento a la biblioteca se debe seguir los siguientes pasos:

1. Presionar el botón de **Agregar**.
2. Ingresar los datos del documento:

Escribir la clave el estante en el cual será ubicado (esto se refiere al tipo de documento que es:

- 0-FÍSICA.
- 1-MATEMÁTICAS.
- 2-TÉCNICOS.
- 3-ADMINISTRATIVOS.
- 4-HUMANISTAS Y OTRAS CIENCIAS)

Después dar el nombre del documento, indicar el autor, el tipo de documento y para finalizar indicar la localización en donde se encuentra el documento (El documento deberá existir previamente en cualquier medio magnético o de almacenamiento y que se pueda tener acceso por medio de una búsqueda de archivo , ya que de no existir o no tenerlo disponible para el sistema podría causar un error en el sistema.

3. Finalmente presionar el botón de **Aceptar**

Si se desea eliminar algún documento hacer clic en el botón **Eliminar** y después confirmar la operación Cabe hacer mención que solo se elimina el registro del documento en la base de datos de SIVEA y que si se desea eliminarlo totalmente del medio magnético deberá de eliminarse con algún programa como el explorador de WINDOWS*.

Para modificar alguno de los datos del documento hacerlo directamente escribiendo los nuevos datos y dar clic en el botón de **Aceptar**.

Para cancelar cualquiera operación hacer clic en el botón de **Cancelar**

Para cerrar esta pantalla hacer clic en **Cerrar**.

Laboratorios.

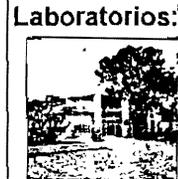
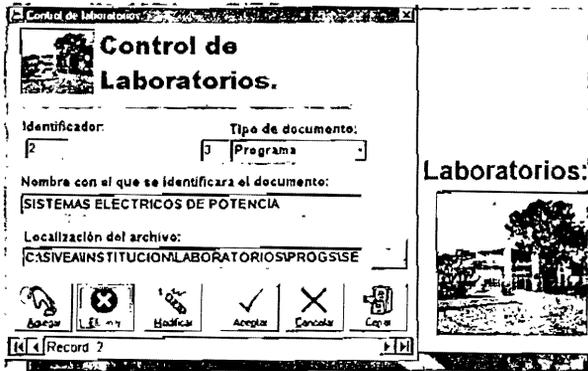
Si se selecciono la imagen de laboratorios aparecerá la ventana de control de laboratorios

Para agregar un elemento a los laboratorios se debe seguir los siguientes pasos

1. Presionar el botón de **Agregar**

- 2. Ingresar los datos requeridos
- Indicar de que tipo de elemento se trata.
- DOCUMENTO
- PROGRAMA.
- PRACTICA.

Después dar el nombre del elemento , y para finalizar indicar la localización en donde se encuentra el documento (El documento deberá existir previamente en cualquier medio magnético o de almacenamiento y que se pueda tener acceso por medio de una búsqueda de archivo , ya que de no existir o no tenerlo disponible para el sistema podría causar un error en el sistema



3 Finalmente presionar el botón de **Aceptar**

Si se desea eliminar algún documento hacer clic en el botón **Eliminar** y después confirmar la operación. Cabe hacer mención que solo se elimina el registro del documento en la base de datos de SIVEA y que si se desea eliminarlo totalmente del medio magnético deberá de eliminarse con algún programa como el explorador de WINDOWS*.

Para modificar alguno de los datos del documento hacer clic en **Modificar** , realizar las modificaciones y dar clic en el botón de **Aceptar**.

Para cancelar cualquiera operación hacer clic en el botón de **Cancelar**.

Para cerrar esta pantalla hacer clic en **Cerrar**.

• LABORATORIOS

El módulo de laboratorios se compone de tres bloques los cuales permite acceder a diferentes funciones cada uno ,para entrar dar un clic en el botón del bloque al que se desea acceder ; a continuación se explica cada uno de estos bloques:



DOCUMENTOS: Permite acceder a documentos relacionados con las técnicas que se manejan en cuanto lo básico de la ingeniería

PRACTICAS. Son archivos ejecutables los cuales presentan ejercicios sencillos o propuestas de prácticas a realizar.

PROGRAMAS: Son programas de otros proveedores y tipos que se relacionen con la ingeniería y que se deseen agregar a la lista de programas que pueda ejecutar SIVEA. **NOTA.** Para agregar un programa a este sistema se deberá tener una licencia donde el fabricante del producto autorice la libre distribución del programa o su uso en instituciones académicas

Para salir de este módulo se realiza haciendo clic en el botón *Regresar* si no esta activa la pantalla de bloque, en caso de estar activada presionar dos veces sobre el botón

o BIBLIOTECA

El módulo de biblioteca es el mas sencillo del sistema y en el se podrá acceder a documentos de diversas materias que hayan sido agregados en la biblioteca (para saber como agregar documentos a la biblioteca ver la ayuda sobre el módulo de gobierno)

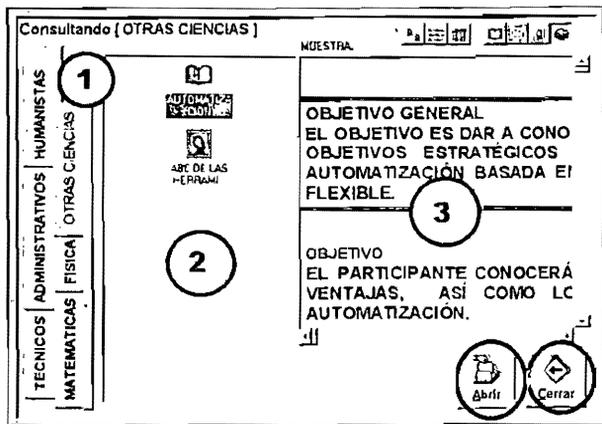


La manera de acceder a los documentos de la biblioteca será haciendo clic en el botón de abrir (Encerrado en el círculo rojo en la figura). E inmediatamente se vera el módulo con sus bloques los cuales se describen a continuación.

1 SEPARADORES DE AREA.
 Cuando se hace un clic en este se accede a una determinada área ; las áreas disponibles son:
 FÍSICA.
 MATEMÁTICAS.
 TÉCNICOS.
 ADMINISTRATIVOS.
 HUMANISTAS Y OTRAS CIENCIAS.

2.CUADRO DE ICONOS. En el se muestran los temas disponibles sobre el área seleccionada previamente de los separadores de área.

3. CUADRO DE MUESTRA. En este marco o cuadro se muestra el contenido del tema seleccionado en el cuadro de iconos



Para acceder al tema en cuestión hacer clic en el botón de abrir marcado con un círculo negro(mas oscuro en la impresión).

Para cerrar el módulo hacer clic en **Cerrar**. Y para salir de este hacer clic en **Cerrar** de nueva cuenta.

▣ AUDITORIO.

El módulo de auditorio es una presentación realizada para ser mostrada durante el examen profesional, la cual pretende mostrar la importancia de este sistema, así como los componentes y en resumen todo la labor realizada para lograr este trabajo de tesis.



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO.
ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES
ARAGÓN.

SIVEA

SISTEMA
VIR TUAL DE
ENSEÑANZA
APRENDIZAJE
CON
TECNOLOGÍA BÁSICA

Por Ignacio de la Rosa R.

Para concluir, la forma en que se utilice este sistema y el máximo aprovechamiento de este depende totalmente de los usuarios y en la medida que sea utilizado se podrá encontrar mejores formas de explicar su operación y uso, y serán los mismos usuarios aquellos que retroalimenten al sistema y determinen la mejor manera de usarlo y aplicarlo

CAPITULO V

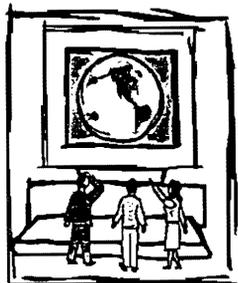


**APLICACIONES
Y RESULTADOS
CONCLUSIONES**

APLICACIONES Y RESULTADOS CONCLUSIONES.



APLICACIONES Y RESULTADOS



Sin duda uno de los retos más grandes de las telecomunicaciones en la educación será afrontar con mucha seriedad los mundos digitales a los que nos veremos enfrentados, pero las telecomunicaciones no podrán solas asumir estos retos, con la ayuda de la Realidad Virtual será mucho más sencillo lograr mundos muy reales

Uno de los impactos más grandes que tendremos será el cambio social que se dará en nuestra sociedad, tendremos que afrontarlo con mucha seriedad pues llegará el momento en que no sepamos diferenciar entre la realidad y la virtualidad

El objetivo principal de desarrollar un trabajo de esta naturaleza fue la de contar con una herramienta de apoyo para los alumnos y para los facilitadores (profesores de la carrera) en los temas de física y matemáticas.

Un objetivo secundario que se gestó durante el desarrollo de este trabajo fue el que el ingeniero actual está alejado del uso de sistemas de cómputo. Este sistema permitiría actualizarse, las empresas se verían fuertemente beneficiadas por el hecho de los ingenieros con los que cuentan accedan a este tipo de información sin necesidad de ir a una institución en particular o a una hora establecida, ya que podrían realizar su autoaprendizaje en la misma empresa o en sus hogares al ritmo que ellos lo consideraran conveniente

¿Por qué realizar un software sobre "Aprendizaje y enseñanza en Ingeniería básica", cuando existen tantos programas educativos en el mercado actualmente? Además de faltar herramientas de aprendizaje enfocadas para los cursos de la asignatura, por otro lado existen programas que se dedican en general a presentar de manera introductoria a las computadoras, o un lenguaje "educativo" como Logo, o en el mejor de los casos se pasa revista a algunas modalidades y paquetes de la enseñanza asistida por computadora. Sin embargo, esto no es suficiente para tratar con las computadoras en una escuela o ambiente educativo, cuando parece que las computadoras son un tema aislado o separado de la tarea docente. Aunque no hay teoría definitiva al respecto, ¿Cómo se realizaría un programa de computación con fines educativos?, ¿Cuándo conviene realizar un programa, cuánto costaría?, ¿Qué hacer si se quiere introducir y comprar computadoras en la escuela, qué hay que prever? ¿Cómo se administraría un aula de computadoras en la escuela? ¿Cómo comprar, seleccionar y evaluar equipo, programas"? etc. Quedan muchas preguntas de tipo práctico, sin contestar en los libros tradicionales de computadoras y educación.

Además de estudiantes, se contempló como parte de los usuarios potenciales a los profesores; para educadores que en esta época ya no sólo no pueden permitirse el ignorar la herramienta computacional, sino que están deseosos de utilizarla de manera efectiva en su quehacer cotidiano. Estos facilitadores de Educación en términos generales ya sufrieron los cursos de iniciación computacional y le han perdido el miedo a la computadora, pero de ahí a utilizarla efectivamente dentro del medio académico existe aun una brecha, les falta mucho por aprender no sólo de computación sino también de cómo usarla para fines educativos

Los usuarios de SIVEA y en general de este tipo de herramientas educativas computacionales son los alumnos de licenciaturas o bachilleratos que pueden pensar que el mundo educativo les puede proveer el campo de aplicación de sus conocimientos básicos en física y matemáticas. A los facilitadores hay que mostrarles que lo difícil de realizar una lección educativa computarizada no es el programarla (aunque esta pueda tener su dificultad sobretodo en el manejo de animación, colores, sonidos y archivos), sino en su concepción y diseño de tal manera que no se olvide nada, que sea pedagógicamente adecuada y bien realizada desde el punto de vista de la facilitación del proceso de enseñanza – aprendizaje.

Debido a lo anterior este trabajo de tesis pretende además de servir de referencia y consulta en el tema (no de los productos educativos computacionales, que al cabo de unos pocos años pasan de moda), es una tesis que pretende revisar y mostrar algunas técnicas de cómo realizar un herramienta de enseñanza – aprendizaje , Evaluación, sesiones de aprendizaje , etc.) y no sólo mostrar que es lo que se puede hacer. El hombre y menos aún el maestro son esclavos de la técnica, ni de sistemas ciegos, es en la educación donde se siembra tal vez lo más precioso del hombre: la libertad de pensamiento, de creatividad, de crítica, por eso tiende que entender y resolver el maestro la crítica sobre el uso e introducción a las computadoras en la educación así como entender sus limitaciones e implicaciones.

Este trabajo de tesis cubre los siguientes temas tratando de explicar el por que y para que el uso de SIVEA :

- -el facilitador como utilizador inteligente de material computacional y herramientas de cómputo, en su quehacer educativo.
- -al facilitador como autor de material instruccional con computadora en sus diferentes modalidades.
- -el facilitador como evaluador y seleccionador de material de cómputo para la escuela.

Como se trato en este trabajo de tesis, utilizado para despejar y limpiar algunos malentendidos, falsas especulaciones y creencias que sobre las computadoras en educación , tratando de justificar la importancia del uso de la computadora en las instituciones académicas en estos días, y cuales son las tendencias a futuro en cuanto en estos temas, aun existe un largo camino por andar y tantos estudios por realizar que permitan tener un control de estos procesos y sus consecuencias. Se consideran las diferentes formas de introducción de las computadoras en la educación, desde la enseñanza de la computación misma, hasta la computarización de la enseñanza. El impacto de la entrada de las computadoras en la escuela y sobre algunos de los principales parámetros educativos como el lenguaje, la cultura , el libro y el rol del maestro. Se trato de describir también algunos cambios curriculares y en particular el papel de los cursos usando medios electrónicos, la visión de futuro que se tiene que tener y se presentan algunas de las tecnologías que prometen más en este campo. Se habla de manera general de las múltiples aplicaciones de la computación en la educación, el empleo de las telecomunicaciones en la educación, o los usos del videoconferencias y otras formas de audiovisual en conjunción con las computadoras etc.

CONCLUSIONES



Es un hecho que podemos lamentar, pero que no se puede entrar a discusión (no en este trabajo de tesis), que el nivel de computarización del sistema educativo y, en particular, el nivel de integración de la actividad docente en clase con actividades de enseñanza asistida por computadora, es mucho más bajo en América Latina que, por ejemplo, en los Estados Unidos, Canadá, Francia, y otros países altamente industrializados.

A primera vista, esto puede considerarse una consecuencia natural de la diferencia de niveles salariales entre " Norte " y " Sur ". Es mucho más fácil y justificable incorporar una computadora a la escuela cuando su costo equivale a un mes de sueldo de maestro o profesor, que cuando su costo equivale a un año de sueldo del docente.

Pero esta diferencia en tecnificación tiene efectos secundarios que tienden a aumentar las diferencias en cuanto a posibilidades de formación de los estudiantes. En efecto, la introducción de computadoras en el sistema educativo tiende a aumentar la adaptación de los procesos de enseñanza a las condiciones de cada estudiante, así como a facilitar el desarrollo de procesos cognitivos de alto nivel, brindando oportunidades difíciles de replicar en la clase convencional.

En este choque entre las aspiraciones de mejoramiento del proceso educativo, y las limitaciones presupuestales tan conocidas en América Latina, es necesario que un cuadro de profesionales esté bien preparado, en cada país, para asesorar a los organismos de enseñanza en cuanto a aquellas áreas del conocimiento y procesos de aprendizaje que pueden beneficiarse especialmente de la computarización (aunque sea parcial) del proceso de enseñanza.

Asimismo, cada país debe preparar un cuadro de profesionales capaz e evaluar los programas de computación educativa existentes en el mercado, y capaz de adaptarlos o, por lo menos, definir la forma en que deben ser adaptados a las condiciones y cultura locales.

Finalmente, después de todo este tiempo en el cual he tenido la oportunidad de desarrollar este trabajo, me he dado cuenta de que este tipo trabajos de ingeniería requiere de mas tiempo para su desarrollo y de que siempre podrán ser mejorados y superados, por lo que espero que este trabajo abra la puerta al desarrollo de software mexicano que trascienda mas allá de la aplicación casera y que pueda ser utilizado en todo el mundo; este es un sueño que yo tengo y que estoy seguro que algún día lo podré alcanzar.

ANEXOS

ANEXO A

En busca de la Clase Virtual.

A continuación se anota un extracto acerca de algunas reflexiones hechas por John Tiffin, Lalita Rajasingham en su texto "En Busca de la Clase Virtual", realizado en Barcelona, España, Editorial Paidós, 1997, 274 pgs.

¿Que tipo de sistema se necesita para preparar a las personas a vivir en una sociedad de la información?

Necesitamos un sistema de educación que no forme personas para que salgan a ocupar puestos de trabajo en las empresas, sino que realmente nos preparen para afrontar los nuevos cambios y recibir las nuevas tecnologías como el futuro de una manera muy profesional, esto debido a que los sistemas actuales de educación no han sido diseñados para esto. Los gobiernos del mundo junto con los Ministerios de Educación y en cooperación con las instituciones educativas, desde colegios hasta las universidades deben empezar a reunirse pensando en lo que nos espera para los años próximos.

¿Significa el tele aprendizaje el final del aula tal y como la conocemos?

El aula tal y como la conocemos no se conservara, aunque lo que haremos será trasladarla a un medio diferente y para lograr esto iremos de la mano con la realidad virtual, es así que por medio de ella lograremos llegar a un tele aprendizaje dentro de un aula, nos podremos seguir reuniendo pero esta vez en un espacio virtual.

¿Cómo es posible tener el efecto de una clase sin la realidad del aula?

Sería posible seguir teniendo el efecto de una clase en el aula con la ayuda de otras tecnologías existentes como lo es por ejemplo la videoconferencia, además la realidad virtual deberá ayudarnos a despegar en este sentido pero en un futuro existirán tecnologías más poderosas y potentes que nos ofrecerán el efecto de una clase virtual.

¿Puede la tecnología informática proporcionar un sistema de comunicación alternativo para el aprendizaje que sea al menos igual de efectivo?

Si, la tecnología lo podrá hacer, de hecho la creación del hipertexto proporcionará un sistema de comunicación increíblemente bueno para así lograr un nuevo sistema de aprendizaje o alternativo, el empleo de esta tecnología logrará realmente cambios muy buenos en el aprendizaje.

¿ Que será posible hacer con la realidad virtual dentro de cincuenta años?

Será posible fuera de lograr clases demasiado reales llegaremos a poder circular por ciudades virtuales, centros comerciales, donde podremos comprar, estacionar nuestros autos casi crearemos un mundo paralelo al real, esto gracias a todos los sensores que serán creados por el hombre para poder llegar dentro de cincuenta años a casi no poder diferenciar entre lo real y lo irreal. Así que podemos concluir que las clases serán muy reales debido a que lograremos tener delante de los ojos dos pantallas líquidas que surtirán efectos demasiado reales.

¿ Qué forma tendrá una clase virtual?

Una clase será similar a las reuniones que hoy en día tenemos en un aula normal, con la diferencia que las personas que asistan a dichas reuniones serán totalmente virtuales y podrán estar en cualquier parte del globo terráqueo, esto conllevará a una globalización de la educación y el grado cultural de los alumnos será bastante elevado. Claro que estas reuniones estarán acompañadas de las más altas tecnologías y aparatos que harán posible llegar a emplear buena parte de la tecnología de información para lograr llegar a tener una clase por nivel excelente, estos aparatos son computadoras, modem, líneas telefónicas, cámaras de televisión entre otros.

¿ Qué capacidades potenciales tendrá?

Las capacidades potenciales serán bastante altas debido que llegaremos a una globalización de la educación y se desarrollara una nueva manera de aprendizaje en un aula virtual con todos los mecanismos suficientes para que los alumnos desarrollen todo su potencial en aprender a aprender.

¿ Es posible que una clase virtual sea una mejora con respecto al aula convencional como sistema de comunicación para el aprendizaje en la venidera sociedad de información?

Si, de hecho vamos a utilizar mucha tecnología aplicada al aprendizaje virtual y esto indudablemente mejorará el sistema de comunicación que existía en el aula convencional, en el aula convencional el alumno tímido será tímido siempre dentro del salón, mientras que esta clase de niños y personas perderán el miedo a participar en un aula virtual, pues tendrán muchos mas mecanismos para dirigirse a los compañeros o hacia el mismo profesor, por ejemplo el uso del correo electrónico.

¿ Será ese el lugar en el que nuestros hijos y sus hijos aprenderán a aprender?

Si, nuestros hijos aprenderán a aprender porque cambian todos los hábitos con respecto a los que obtuvimos nosotros los estudiantes tradicionales, ellos más que estudiantes serán investigadores en los diferentes temas que el profesor lleve propuesto para discutir en clase, siempre nosotros como estudiantes vamos exigiendo y ellos exigirán demasiado, así que el papel del profesor cambiará de modo sustancial, cada vez deberá ser más preparado, pues el nivel del estudiante será cada vez más alto.

¿Utilizaremos esta extraordinaria tecnología para avanzar en la forma en que aprendemos, o la utilizaremos para crear aulas virtuales que sean versiones virtuales de las aulas convencionales?

Indudablemente la realidad virtual la utilizaremos al principio para crear aulas virtuales que sean versiones virtuales de las aulas convencionales, pero con el transcurrir del tiempo y el avance de la tecnología nos servirán para llegar a puntos en que avancemos a velocidades altas para poder evolucionar en la forma de aprender, toda tecnología que se aplique de una manera adecuada en cualquier campo (en este caso la educación) significara un avance en la búsqueda de nuevos y satisfactorios resultados.

¿ Cómo encontraremos el nuevo paradigma educativo?

Este nuevo paradigma educativo lo encontraremos en la medida que las tecnologías se acoplen totalmente a la educación, pues sé ira dando poco a poco, esto lo afirmo porque lo que podrá llegar a existir o lo que podremos desarrollar todavía no lo vemos tan próximo, pero con la suma de todos estos factores se mejorará el nivel educativo en el mundo y llegaremos a la creación de este nuevo paradigma que todos estamos esperando.

¿ Que seriedad tiene un marco como este?

Se tiene mucha seriedad al afirmar todo este tipo de cosas, a esto podremos llegar en menos tiempo del que nos esperamos, todas las tecnologías que estamos desarrollando a finales de este siglo están avanzando a pasos gigantescos, así que esto que se describió parece un cuento como el de Alicia en el país de las maravillas será lo suficientemente real en unos años y nosotros quedaremos con los unos simples visionarios que no alcanzamos a percibir todo lo que en realidad se vivirá.

¿Cuales serán las consecuencias sociales, económicas, políticas y psicológicas si la clase virtual llega a ser la forma dominante de instrucción?

Entre las consecuencias sociales podremos citar que nos iremos encerrando poco a poco en nuestras casas pues gracias a un computadora tendremos acceso a todo, desde una biblioteca digitalizada, como documentos relacionados con cualquier tema, grupos de discusión, correo electrónico, videoconferencias que nos evitaran los desplazamientos normales que tenemos hoy en día, en las económicas no sé hasta donde se pueda llegar a afectar pero las compras o la adquisición de lo necesario se seguirá realizando sin que llegue a afectar a nivel macro económico a un país, las psicológicas serán de pronto las más graves pues estos cambios si afectaran nuestro modo de vida radicalmente al punto en que nos volveremos ermitaños en el mundo real navegando en un mundo virtual, los ojos nuestro serán cuadrados pues no quitaremos los ojos de una pantalla y esto si no lo sabemos manejar se nos puede convertir en un problema, podremos llegar a convertir a nuestros hijos es simples maquinas programables.

ANEXO B

Guía de reactivos aplicados en la evaluación de diagnóstico

A continuación se lista los reactivos aplicados en la evaluación de diagnóstico aplicada a los alumnos de primer ingreso de la carrera de IME.

SUBTEMA	REACTIVO(S)	Número de reactivos
PARTE I MATEMATICAS		
ARITMETICA	1-10	10
ALGEBRA	11-20	10
TRIGONOMETRIA	21	1
GEOMETRIA ANALITICA	22-27	6
CALCULO DIFERENCIAL	28-29	2
CALCULO INTEGRAL	30-31	2
TOTAL		31
PARTE II FISICA		
ESTATICA	32	1
DINAMICA	33-36	4
CINEMATICA	37-39	3
HIDRAULICA	40-41	2
TERMODINAMICA	42-43	2
ACUSTICA	44	1
OPTICA	45	1
ELECTRICIDAD	46-50	5
TOTAL		19
PARTE III OPINION		
5 Reactivos	En esta parte se evalúa el impacto y aceptación que tendrá el sistema de software que se desea realizar.	

Relación de los reactivos para su aplicación en el examen de diagnóstico de acuerdo al tema que se evalúa.

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

ESCUELA NACIONAL DE ESTUDIOS
PROFESIONALES

ARAGON

EXAMEN DIAGNOSTICO

1998.

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES ARAGON

Examen de diagnostico

El presente examen tiene como único fin el evaluar tus conocimientos, no tiene valor curricular. Para contestar usa la hoja de respuestas, no hagas ninguna anotación aquí.

PARTE I MATEMATICAS.

1.- Parte de las matemáticas que tiene por objeto el estudio de los números.
a) Aritmética b) Algebra c) Geometría d) Trigonometría.

2.- Es una verdad intuitiva que tiene suficiente evidencia para ser aceptado.
a) Postulado b) Teorema c) Corolario d) Teoría

3.- Es una verdad no evidente pero demostrable.
a) Postulado b) Teorema c) Corolario d) Teoría

4.- Valor que tiene un número por su forma o figura.
a) Valor relativo b) Valor Absoluto c) Valor compuesto
d) Ninguno de los anteriores

5.- De dos o más números es el menor número que contiene un número exacto de veces a cada uno de ellos.
a) M.C.D. b) Número compuesto c) Número primo d) m.c.d.

6.- Es un número primo:
a) 1749 b) 2234 c) 987 d) 1157

7.- La simplificación de $(\frac{9}{10}) / (\frac{2}{3} - \frac{1}{4})$ es

a) 1 55/329 b) 54/65 c) 721/10 d) 2 1/2

8.- 10 Hombres trabajando en la construcción de un puente hacen 3/5 de la obra en 8 días si se retiran 8 hombres, ¿ Cuanto tiempo el resto tardara en terminar la obra?

a) 80 días b) 3 días c) 26 2/3 días d) No se puede saber.

9.- Hallar el 1/8 % de 96
a) 10 b) 9.8 c) .8 d) 12

10.- El logaritmo de 0.9463
a) $\bar{4}.4224$ b) $\bar{3}.3424$ c) 0.4224 d) No se puede saber.

11.- Resolver aplicando la fórmula general cuadrática $x^2 - 3x + 2 = 0$
a) $x = -1, -2$ b) $x = 1, 2$ c) $x = 1+i, 2+i$ d) $x = 0$

12.- La solución por determinantes de $\begin{vmatrix} A & -B \\ C & D \end{vmatrix}$ es:

a) CD / AB b) AC / BD c) AD / CB d) BD / AC

13.- reduciendo $-6m+8n-5-m-n-6m-11$ se obtiene
a) $-12m-9n$ b) $m+8n-16$ c) $m-n-11$ d) $-13m+7n-6$

14.- La suma de $-m+6mn-4s$ mas $6s-am-5mn$ mas $-2s-5mn+3am$ es :
a) $12s+3am-4mn$ b) $12s+am$ c) $8s+am-4m$
d) $am-4mn$

15.- La división de $-8a^2 + 12ab - 4b^2$ entre $b - a$ es:
a) $8^a - 4b$ b) $8^a - ab - 4b$ c) $8^a - ab$ d) $4ab$

16.- La solución de $((15)^2)^4)^2$ es:

a) 15^8

b) 15^{16}

c) 15^4

d) 15^{32}

17.- La Factorización de $m(a-b) + (a-b)n$ es:

a) $(a-b)mn$

b) $(a-b)(m+n)$

c) $ab(m+n)$

d) $am-bn$

18.- Al multiplicar $\sqrt{x} \times \sqrt{2x^2}$ se obtiene:

a) $3x$

b) $2\sqrt{x^3}$

c) $\sqrt{2x^3}$

d) $x^3\sqrt{2}$

19.- La solución para el sistema de ecuaciones: $7x - 4y = 5$ es:

$9x + 8y = 13$

a) $y=1$

b) $y=-1$

c) $x=1$

d) $x=-\frac{1}{2}$

x = $\frac{1}{2}$

x = $\frac{1}{2}$

y = $\frac{1}{2}$

y = -1

20.- La solución de $6x - 9 > 3x + 15$ es:

a) $x < 8$

b) $x > 8$

x = 8

x > -8

21.- Si tiene el siguiente triángulo:

A = 23.3°

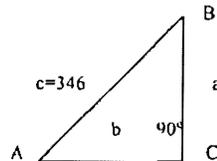
a = ?

B = ?

b = ?

C = 90°

c = 346



a) B = 66.7°

b) B = 66.7°

c) B = 60.7°

d) B = 66.7°

a = 200

a = 100

a = 137

a = 137

b = 137

b = 318

b = 300

b = 318

22.- La distancia entre los puntos $(0, 0)$ y $(5, 12)$ es:

a) 5

b) -13

c) 12

d) 13

23.- La ecuación de la recta que pasa por el origen y tiene por pendiente $\frac{4}{3}$ es:

a) $3x-4y=0$

b) $4x-3y=0$

c) $-4x+3y=0$

d) $-3x-4y=0$

24.- La ecuación de la circunferencia de centro $C(2,0)$ y radio 2 es:

a) $x^2 - 4x + y^2 = 0$

b) $4x^2 - 3x + 2y^2 = 0$

c) $x^2 + 4x + y^2 = 0$

a) $x^2 + 4xy + y^2 = 0$

25.- La ecuación de la parábola cuyo vértice es $V(-2, -4)$, $P=1$ con eje focal paralelo al eje "y" y se abre hacia la parte positiva del eje "y" es:

a) $x^2 - 4x - 4y - 12 = 0$

b) $x^2 + 4x - 4y - 12 = 0$

c) $x^2 - 4x - 4y + 12 = 0$

d) $x^2 - 4x - 4y = 0$

26.- La ecuación de la elipse con las siguientes condiciones $2a=8$, $F'(-2, 0)$, $F(2, 0)$ y $C(0, 0)$ es:

a) $-3x^2 - 4y^2 + 48 = 0$

b) $3x^2 - 4y^2 + 48 = 0$

c) $3x^2 - 4y^2 - 48 = 0$

d) $x^2 - 4y^2 + 48 = 0$

27.- La ecuación de la hipérbola con $2a = 3$, $F'(-2, 0)$ y $F(2, 0)$ es:

a) $28x^2 + 36y^2 + 63 = 0$

b) $28x^2 + 86y^2 - 60 = 0$

c) $x^2 - 36y^2 - 63 = 0$

d) $28x^2 - 36y^2 - 63 = 0$

28.- La derivada de $f(x) = x^3 - x^2 + x - 2$ es:

a) $f'(x) = 3x^2 - 2x + 1$

b) $f'(x) = 3x^3 - 2x^2 + x$

c) $f'(x) = 3x - 2x + 1$

d) $f'(x) = 3x - 2x + x$

29.- Los puntos máximos y mínimos de la ecuación $y = x^4 - 4x^3 + 16x$ son:

a) p. max = $(-1, 5)$

b) p. max = $(1, -3)$

c) p. max = $(1, -5)$

d) p. max = $(-1, -5)$

p. min. = $(1, -3)$

p. min. = $(-1, -5)$

p. min. = $(-1, 3)$

p. min. = $(1, -3)$

30.- La solución de la integral indefinida $\int (6x^3 - 4x^4 + 2x - 3x^2 + 10) dx$ es:

a) $\frac{3}{2}x^4 - \frac{4}{5}x^5 - x^3 + x^2 + 10x + C$

b) $\frac{3}{2}x^4 - \frac{4}{5}x^5 - x^3 + x^2 + 10x$

c) $x^4 x^5 - x^3 + x^2 + 10x + C$

d) $18x^4 - 16x^5 - 6x^3 + 2x^2 + 10$

31.- El cálculo de la integral definida $\int_{-2}^1 (x-1)^{2/3} dx$

a) 0

b) -2.5

c) 2.5

d) $2x + x$

32.- Los extremos de una cuerda de 10 m. de longitud están unidos a dos ganchos colocados en un techo horizontal a una distancia entre si de 6 mts. En el punto medio de la cuerda se cuelga un peso de 64 Kp. La tensión en cada uno de los segmentos de la cuerda sería:
a) 1 Kp b) 640 Kp c) 64 kp d) No se puede saber

33.- Un camión vacío cuya masa es de 2000 Kg, tiene una aceleración máxima de 1 m/seg² ¿Cuál será su máxima aceleración cuando lleva una carga de 1000 kg ?

a) 1 m/seg² b) 0.66 m/seg² c) 1000 m/seg² d) 1.5 m /seg²

34.- Cuanta fuerza centrípeta se necesita para mantener en movimiento una piedra de 0.5 kg. en un círculo horizontal de 1 m de radio a una velocidad de 4 m/seg.?

a) 0.5 N b) 4.5 N c) 8 N d) 2.5 N

35.-Cuanto trabajo se realiza al levantar un libro de 2 kg desde el suelo hasta una altura de 1.5 m ?

a) 30 J b) 28 J c) 4 J d) 29.4 J

36.- ¿ Cuanta energía potencial tiene el libro en su nueva posición ?

a) 30 J b) 0 J c) 29. J d) 4 J

37.-¿Cuánto tiempo tarda la luz para viajar del sol a la tierra, si la distancia es de 1.5×10^{11} metros ?

a) 15×10^8 m/s b) 3×10^8 m/s c) 8×10^3 m/s d) 15×10^8 m/s

38.-Un avión debe tener una velocidad de 50 m/s para levantar el vuelo ¿ Cuál debe ser la aceleración del avión si debe elevarse en una pista de $\frac{1}{2}$ de Km ?

a) 2.5 m/s² b) 5 m/s² c) 1.5 m/s² d) 5 m/s²

39.- Una rueda de 80 cm. de diámetro gira a 120 revoluciones por minuto (R.P.M) ¿ Cuál será la velocidad angular de la rueda, en rad/s ?

a) 12.6 rad/s b) 12 rad/s c) 0.6 rad/seg d) -12.6 rad/s

40.-Una viga de roble de 10 cm x 20 cm. x 4m. Tiene una masa de 58 kg. La densidad del roble sería:

a) 0.725 kg/ M³ b) 725 Kg / M³ c) 7.25 Kg/ M³ d) 527 Kg/ M³

profundidad es:

a) 3.2×10^5 N b) 20×10^5 N c) 3.2 N d) 3.1×10^3 N

42.- La conversión de 80°F a °C es:

a) 80°C b) 26.7°C c) 15.5°C d) -26.7°C

43.-¿qué cantidad de calor debe de suministrarse a 3 kg. De agua para elevar su temperatura de 20°C a 80°C

a) 100 Kcal b) 180 cal c) 180 Kcal d) 1.80 Kcal.

44.- ¿Cuál es la velocidad del sonido en el aire?

a) 3.8 m/seg b) 3.24 m/seg c) 343.2 Km/seg d) 343.2 m/seg

45.-Si un lente tiene 0.4 dioptrías de poder , ¿Cuál es su distancia focal ¿

a) 2.5 m b) 10m c) 0.4m d) 5.2 m

46.-La intensidad de corriente que circula por un conductor de un brasero eléctrico que tiene una resistencia en caliente de 22 Ohms y se enchufa a una línea de 110 V es:

a) 220 A b) 5 A c) 10 A d) 3 A

47.- Se tiene una batería con una F.E.M. de 15 V y una resistencia interna de 0.05 Ohms, ¿Cuál será su tensión en los bornes cuando se carga con 10 A.?

a) 15.5 V b) 15V c) 0 V d) 9.5 V

48.- Un condensador de un circuito tiene una capacidad de 1.2 MicroF y la diferencia de potencial entre sus bornes vale 3000 V. La energía almacenada en él sería:

a) 5.4 J b) 10J c) 0J d) No se puede calcular

49.-La resistividad de una vaina de cobre es de $1,756 \times 10^{-8}$ Ohmsm, tiene 2 m. de longitud y 8mm de diámetro. Su resistencia sería:

- a) 4 Ohms b) 6.99×10^{-4} Ohms c) 6.99 Ohms d) 6×10^{-9} Ohms

50.-¿ Cuál será el número de espiras del secundario de un transformador utilizado para elevar la tensión de 120 V a 1800 V, sabiendo que el primario consta de 100 espiras?

- a) 200 espiras b) 50 espiras c) 1500 espiras d) 500 espiras

PARTE III

OPINION

51.- Piensas que un libro debe tener

- a) muchos ejemplos y pocos ejercicios
b) pocos ejemplos y muchos ejercicios
c) pocos ejemplos y pocos ejercicios.
d) No es relevante.

52.- Crees que sea importante recibir un curso propedeutico para nivelar tus conocimientos en las materias básicas de ingeniería.

- a) Muy importante b) importante c) Poco importante d) No

53.- Tus conocimientos en computación(básicos) están en un:

- a) 100% b) 75% c) 50% d) 25%

54.- Crees que los tutoriales facilitan el trabajo de aprendizaje en los estudiantes.

- a) Si mucho b) Si c) No d) No los conozco.

55.- Estarías dispuesto a participar en la creación de tutoriales para mejorar el proceso enseñanza – aprendizaje.

- a) Si b) No

ANEXO C

INTRODUCCION A LA PROGRAMACION DEL LENGUAJE VISUAL BASIC.

- **Introducción**

Este tema está dedicado a una introducción al lenguaje de programación Visual Basic. Está basado en las instrucciones o palabras reservadas de Visual Basic. No pretende ser un texto sobre métodos generales de programación, ni tampoco permite cubrir las inmensas posibilidades que ofrece este lenguaje. Está dirigido a la explicación de los fundamentos de este lenguaje de programación.

El enfoque se ha restringido para aclarar y resaltar mejor lo indispensable para entender este lenguaje. Programar una computadora es el "arte" de decir a la máquina lo que tiene que hacer de modo que ésta pueda entenderlo. Contar hasta 100 parece algo muy natural para todos nosotros, sin embargo las computadoras deben ser instruidos paso a paso de como hacerlo.

- **Visual Basic.**

¿Qué es Visual Basic? La palabra "Visual" hace referencia al método que se utiliza para crear la interfaz gráfica de usuario (GUI). En lugar de escribir numerosas líneas de código para describir la apariencia y la ubicación de los elementos de la interfaz, simplemente se puede arrastrar y colocar objetos prefabricados en su lugar dentro de la pantalla. Si se ha utilizado alguna vez un programa de dibujo como Paint, se tendrá una idea de las habilidades necesarias para crear una interfaz de usuario efectiva.

La palabra "Basic" hace referencia al lenguaje BASIC (Beginners All-Purpose Symbolic Instruction Code), un lenguaje utilizado por más programadores que ningún otro lenguaje en la historia de la informática o computación. Visual Basic ha evolucionado a partir del lenguaje BASIC original y ahora contiene centenares de instrucciones, funciones y palabras clave, muchas de las cuales están directamente relacionadas con la interfaz gráfica de Windows. Los principiantes pueden crear aplicaciones útiles con sólo aprender unas pocas palabras clave, pero, al mismo tiempo, la eficacia del lenguaje permite a los profesionales acometer cualquier objetivo que pueda alcanzarse mediante cualquier otro lenguaje de programación de Windows.

El entorno de programación de Visual Basic

• Requisitos para ejecutar Visual Basic.

Para ejecutar Visual Basic, se tiene que disponer de cierto hardware y software instalado en su equipo. Entre los requisitos del sistema cabe citar los siguientes:

- Microsoft Windows NT 3.51 o posterior, o Microsoft Windows 95.
- Microprocesador 80486 o superior.
- Un disco duro con un mínimo de 50 megabytes de espacio disponible para una instalación completa.
- Una unidad de CD-ROM.
- Pantalla VGA o de mayor resolución, compatible con Microsoft Windows.
- 16 MB de RAM.
- Un *mouse* (ratón) u otro dispositivo de puntero.

Para entender el proceso de desarrollo de una aplicación, es útil comprender algunos de los conceptos clave alrededor de los cuales está construido Visual Basic. Puesto que Visual Basic es un lenguaje de desarrollo para Windows, es necesario familiarizarse con el entorno Windows.

• Funcionamiento de Windows: ventanas, eventos y mensajes

Un estudio profundo del funcionamiento interno de Windows necesitaría un libro completo. No es necesario tener un profundo conocimiento de todos los detalles técnicos. Una versión reducida del funcionamiento de Windows incluye tres conceptos clave: ventanas, eventos y mensajes.

Una ventana es simplemente una región rectangular con sus propios límites. Existen varios tipos de ventanas: una ventana Explorador en Windows 95, una ventana de documento dentro de un programa de proceso de textos o un cuadro de diálogo que emerge para recordar una cita.

El sistema operativo Microsoft Windows administra todas estas ventanas asignando a cada una un único número identificador (controlador de ventana o *hWnd*). El sistema controla continuamente cada una de estas ventanas para ver si existen signos de actividad o eventos. Los eventos pueden producirse mediante acciones del usuario, como hacer clic con el *mouse* (ratón) o presionar una tecla, mediante programación o incluso como resultado de acciones de otras ventanas.

Cada vez que se produce un evento se envía un mensaje al sistema operativo. El sistema procesa el mensaje y lo transmite a las demás ventanas. Entonces, cada ventana puede realizar la acción apropiada, basándose en sus propias instrucciones para tratar ese mensaje en particular (por ejemplo, volverse a dibujar cuando otra ventana la ha dejado al descubierto).

• Descripción del modelo controlado por eventos

En las aplicaciones tradicionales o "por procedimientos", la aplicación es la que controla qué partes de código y en qué secuencia se ejecutan. La ejecución comienza con la primera línea de código y continúa con una ruta predefinida a través de la aplicación, llamando a los procedimientos según se necesiten.

En una aplicación controlada por eventos, el código no sigue una ruta predeterminada; ejecuta distintas secciones de código como respuesta a los eventos. Los eventos pueden desencadenarse por acciones del usuario, por mensajes del sistema o de otras aplicaciones, o incluso por la propia aplicación. La secuencia de estos eventos determina la secuencia en la que se ejecuta el código, por lo que la ruta a través del código de la aplicación es diferente cada vez que se ejecuta el programa.

Puesto que no puede predecir la secuencia de los eventos, el código debe establecer ciertos supuestos acerca del "estado del mundo" cuando se ejecute. Cuando se haga suposiciones (por ejemplo, que un campo de entrada debe contener un valor antes de ejecutar un procedimiento para procesar ese valor), se debe estructurar la aplicación de forma que asegure que esa suposición siempre será válida (por ejemplo, deshabilitando el botón de comando que inicia el procedimiento hasta que el campo de entrada contenga un valor).

El código también puede desencadenar eventos durante la ejecución. Por ejemplo, cambiar mediante programación el texto de un cuadro de texto hace que se produzca el evento Change del cuadro de texto. Esto causaría la ejecución del código (si lo hay) contenido en el evento Change. Si supone que este evento sólo se desencadenará mediante la interacción del usuario, se podría ver resultados inesperados. Por esta razón es importante comprender el modelo controlado por eventos y tenerlo en cuenta cuando se diseña una aplicación.

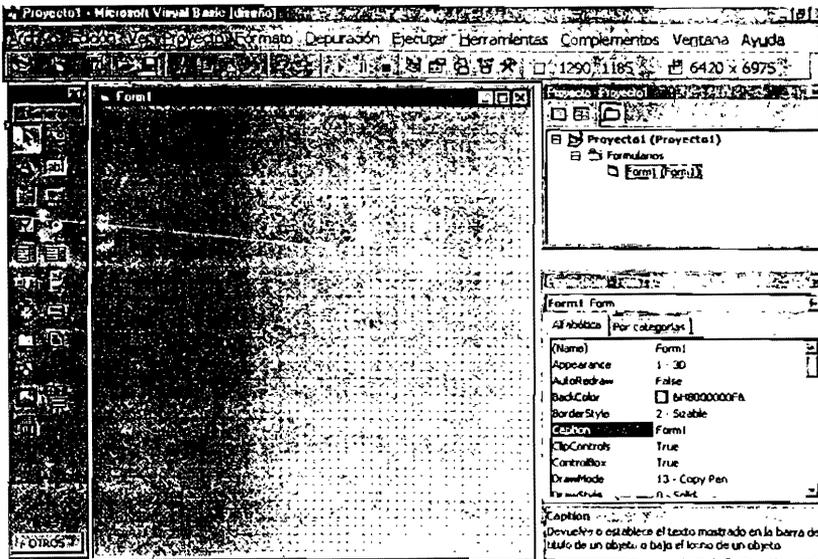
• Desarrollo interactivo

El proceso de desarrollo de las aplicaciones tradicionales se puede dividir en tres etapas diferentes: escritura, compilación y comprobación del código. A diferencia de los lenguajes tradicionales, Visual Basic utiliza una aproximación interactiva para el desarrollo, difuminando la distinción entre los tres pasos.

En la mayoría de los lenguajes, si se comete un error al escribir el código, el compilador intercepta este error cuando comienza a compilar la aplicación. Se deberá encontrar y corregir el error y comenzar de nuevo con el ciclo de compilación, repitiendo el proceso para cada error encontrado. Visual Basic interpreta el código a medida que lo escribe, interceptando y resaltando la mayoría de los errores de sintaxis en el momento. Es casi como tener un experto vigilando cómo escribe el código.

Además, para interceptar errores sobre la marcha, Visual Basic también compila parcialmente el código según se escribe. Cuando se esté preparado para ejecutar y probar la aplicación, tardará poco tiempo en terminar la compilación. Si el compilador encuentra un error, quedará resaltado en el código. Puede corregir el error y seguir compilando sin tener que comenzar de nuevo.

El entorno integrado de desarrollo de Visual Basic (IDE) consta de los siguientes elementos.



Barra de menús

Presenta los comandos que se usan para trabajar con Visual Basic. Además de los menús estándar **Archivo**, **Edición**, **Ver**, **Ventana** y **Ayuda**, se proporcionan otros menús para tener acceso a funciones específicas de programación como **Proyecto**, **Formato** o **Depuración**.

Menús contextuales

Contienen accesos directos a acciones que se realizan con frecuencia. Para abrir un menú contextual, se deberá hacer clic con el botón secundario del *mouse* en el objeto que está usando. La lista específica de opciones disponibles en el menú contextual depende de la parte del entorno en la que se hace clic con el botón secundario del *mouse*. Por ejemplo, el menú contextual que aparece cuando hace clic con el botón secundario del *mouse* en el cuadro de herramientas permite mostrar el cuadro de diálogo **Componentes**, ocultar el cuadro de herramientas, acoplar o desacoplar el cuadro de herramientas, o agregar una ficha personalizada al cuadro de herramientas.

Barras de herramientas

Proporcionan un rápido acceso a los comandos usados normalmente en el entorno de programación. Se deberá hacer clic en un botón de la barra de herramientas para llevar a cabo la acción que representa ese botón. De forma predeterminada, al iniciar Visual Basic se presenta la barra de herramientas Estándar. Es posible activar o desactivar otras barras de herramientas adicionales para modificar, diseñar formularios desde el comando **Barras de herramientas** del menú **Ver**. Las barras de herramientas se pueden acoplar debajo de la barra de menús o pueden "flotar" si selecciona la barra vertical del borde izquierdo y la arrastra fuera de la barra de menús.

Cuadro de herramientas

Proporciona un conjunto de herramientas que puede usar durante el diseño para colocar controles en un formulario.

Ventana Explorador de proyectos

Enumera los formularios y módulos del proyecto actual. Un *proyecto* es la colección de archivos que usa para generar una aplicación.

Ventana Propiedades

Enumera los valores de las propiedades del control o formulario seleccionado. Una *propiedad* es una característica de un objeto, como su tamaño, título o color.

Examinador de objetos

Enumera los objetos disponibles que puede usar en su proyecto y le proporciona una manera rápida de desplazarse a través del código. Puede usar el Examinador de objetos para explorar objetos en Visual Basic y otras aplicaciones, ver qué métodos y propiedades están disponibles para esos objetos, y pegar código de procedimientos en su aplicación.

Diseñador de formularios

Funciona como una ventana en la que se personaliza el diseño de la interfaz de su aplicación. Agregue controles, gráficos e imágenes a un formulario para crear la apariencia que desee. Cada formulario de la aplicación tiene su propia ventana diseñador de formulario.

Ventana Editor de código

Funciona como un editor para escribir el código de la aplicación. Se crea una ventana editor de código diferente para cada formulario o módulo del código de la aplicación.

Ventanas Inmediato, Locales e Inspección

Estas ventanas adicionales se proporcionan para la depuración de la aplicación. Sólo están disponibles cuando ejecuta la aplicación dentro del IDE.

El lenguaje de programación de Visual Basic

Dentro de la programación del lenguaje de Visual Basic se deberá tener en cuenta los siguientes conceptos.

Asignación de valores, tipos

Valor

Supongamos que la memoria de la computadora está compuesta por muchas casillas de igual tamaño y alineadas unas detrás de otra. En cada una puede colocarse un valor, y de cada una se puede extraer un valor. Estos valores se usan para representar información útil. Seguidamente se muestra un programa completo,

Variables

En Visual Basic puede utilizar variables para almacenar valores temporalmente durante la ejecución de una aplicación. Las variables tienen un nombre (la palabra que utiliza para referirse al valor que contiene la variable) y un tipo de dato (que determina la clase de datos que la variable puede almacenar)

Se puede considerar una variable como un marcador de posición en memoria de un valor desconocido. Por ejemplo, supóngase que se está creando un programa para una frutería que hara un seguimiento del precio de las manzanas. No sabe el precio de una manzana o la cantidad que se ha vendido hasta que no se produce realmente la venta. Puede utilizarse dos variables para almacenar los valores desconocidos (vamos a llamarlos PrecioManzanas y ManzanasVendidas). Cada vez que se ejecuta el programa, el usuario especifica los valores de las dos variables. Para calcular las ventas totales y mostrarlas en un cuadro de texto llamado txtVentas, el código debería parecerse al siguiente

```
txtVentas.txt = PrecioManzanas * ManzanasVendidas
```

La expresión devuelve un total distinto cada vez, dependiendo de los valores que indique el usuario. Las variables le permiten realizar un cálculo sin tener que saber antes cuáles son los valores especificados.

En este ejemplo, el tipo de dato de PrecioManzanas es **Currency**; el tipo de dato de ManzanasVendidas es **Integer**. Las variables pueden representar otros muchos valores como valores de texto, fechas, diversos tipos numéricos e incluso objetos

Almacenamiento y recuperación de datos en variables

Se utiliza instrucciones de asignación para realizar cálculos y asignar el resultado a una variable:

```
ManzanasVendidas = 10 ' Se pasa el valor 10 a la variable.  
ManzanasVendidas = ManzanasVendidas + 1 'Se incrementa la variable.
```

Observe que el signo igual del ejemplo es un operador de asignación, no un operador de igualdad; el valor (10) se asigna a la variable (ManzanasVendidas).

Declaración de variables

Declarar una variable es decirle al programa algo de antemano. Se declara una variable mediante la instrucción **Dim**, proporcionando un nombre a la variable:

Dim nombre_variable [As tipo]

Las variables que se declaran en un procedimiento mediante la instrucción **Dim** sólo existen mientras se ejecuta el procedimiento. Cuando termina el procedimiento, desaparece el valor de la variable. Además, el valor de una variable de un procedimiento es *local* de dicho procedimiento; es decir, no se puede tener acceso a una variable de un procedimiento desde otro procedimiento. Estas características le permiten utilizar los mismos nombres de variables en distintos procedimientos sin preocuparse por posibles conflictos o modificaciones accidentales.

El nombre de una variable:

- Debe comenzar con una letra.
- No puede incluir un punto o un carácter de declaración de tipo.
- No debe exceder de 255 caracteres.
- Debe ser única en el mismo *alcance*, que es el intervalo desde el que se puede hacer referencia a la variable: un procedimiento, formulario, etc.

La cláusula opcional **As tipo** de la instrucción **Dim** permite definir el tipo de dato o de objeto de la variable que se va a declarar. Los tipos de datos definen el tipo de información que almacena la variable. Algunos ejemplos de tipos de datos son **String**, **Integer** y **Currency**. Las variables también pueden contener objetos de Visual Basic u otras aplicaciones. Algunos ejemplos de tipos de objeto de Visual Basic, o clases, son **Object**, **Form1** y **TextBox**.

Constantes

A menudo se verá que el código contiene valores constantes que reaparecen una y otra vez. O puede que el código dependa de ciertos números que resulten difíciles de recordar (números que, por sí mismos, no tienen un significado obvio).

En estos casos, se puede mejorar mucho la legibilidad del código y facilitar su mantenimiento si utiliza constantes. Una *constante* es un nombre significativo que sustituye a un número o una cadena que no varía. Aunque una constante recuerda ligeramente a una variable, no puede modificar una constante o asignarle un valor nuevo como ocurre con una variable. Hay dos orígenes para las constantes:

- Constantes *intrínsecas* o *definidas por el sistema* proporcionadas por aplicaciones y controles. Las constantes de Visual Basic se muestran en Visual Basic (VB), Visual Basic para aplicaciones (VBA) y las bibliotecas de Objeto de acceso de datos (DAO) en el Examinador de objetos. Otras aplicaciones que proporcionan bibliotecas de objetos, como Microsoft Excel y Microsoft Project, también proporcionan una lista de constantes que puede utilizar con sus objetos, métodos y propiedades.
- Las constantes *simbólicas* o *definidas por el usuario* se declaran mediante la instrucción **Const**.

Creación de constantes

La sintaxis para declarar una constante es la siguiente:

```
[Public|Private] Const nombre_constante[As tipo] = expresión
```

El argumento *nombre_constante* es un nombre simbólico válido (las reglas son las mismas que para crear nombres de variable) y *expresión* está compuesta por constantes y operadores de cadena o numéricos; sin embargo, no se puede utilizar llamadas a funciones en *expresión*.

Una instrucción **Const** puede representar una cantidad matemática o de fecha y hora:

```
Const conPi = 3.14159265358979
Public Const conMaxPlanetas As Integer = 9
Const conFechaSalida = #1/1/95#
```

Se puede utilizar también la instrucción **Const** para definir constantes de cadena:

```
Public Const conVersion = "07.10.A"
Const conNombreClave = "Enigma"
```

Alcance de las constantes definidas por el usuario

Una instrucción **Const** tiene igual alcance que una declaración de variable y se le aplican las mismas reglas:

- Para crear una constante que sólo exista en un procedimiento, declárese dentro del procedimiento.
- Para crear una constante disponible para todos los procedimientos de un módulo, pero no para el código que está fuera del módulo, declárese en la sección Declaraciones del módulo.
- Para crear una constante disponible en toda la aplicación, declárese la constante en la sección Declaraciones de un módulo estándar y colóquese delante de **Const** la palabra clave **Public**. No se pueden declarar las constantes públicas en un módulo de clase o de formulario.

Tipos de datos

Las variables son marcadores de posición que se utilizan para almacenar valores, tienen nombre y un tipo de dato. El tipo de dato de la variable determina cómo se almacenan los bits que representan esos valores en la memoria del equipo. Cuando se declare una variable, también puede proporcionarle un tipo de dato. Todas las variables tienen un tipo de dato que determina la clase de datos que pueden almacenar.

De forma predeterminada, si no proporciona un tipo de dato, la variable toma el tipo de dato **Variant**. El tipo de dato **Variant** es como un camaleón; puede representar diferentes tipos de datos en distintas situaciones. No tiene que convertir estos tipos de datos cuando los asigne a una variable **Variant**: Visual Basic realiza automáticamente cualquier conversión necesaria.

Sin embargo, si se sabe que una variable almacenará siempre un tipo de dato determinado, Visual Basic tratará de forma más eficiente los datos si se declara la variable con ese tipo. Por ejemplo, se presenta mejor una variable que va a almacenar nombres de personas como el tipo de dato **String**, ya que un nombre está siempre compuesto de caracteres.

Los tipos de datos se aplican a otras cosas además de a las variables. Cuando se asigna un valor a una propiedad, dicho valor tiene un tipo de dato; los argumentos de las funciones tienen también tipos de datos. De hecho, todo lo relacionado con datos en Visual Basic tiene un tipo de dato.

Declarar variables con tipos de datos

Antes de utilizar una variable que no sea **Variant** se debe utilizar las instrucciones **Private**, **Public**, **Dim** o **Static** para declararla *As tipo*. Por ejemplo, la siguiente instrucción declara un tipo **Integer**, **Double**, **String** y **Currency**, respectivamente:

```
Private I As Integer
Dim Cantidad As Double
Static SuNombre As String
Public PagadoPorJuan As Currency
```

La instrucción de declaración puede combinar varias declaraciones, como en las instrucciones siguientes:

```
Private I As Integer, Amt As Double
Private SuNombre As String, PagadoPorJuan As Currency
Private Prueba, Cantidad, J As Integer
```

Nota Si no se proporciona un tipo de dato, se asigna a la variable el tipo predeterminado. En el ejemplo anterior, las variables **Prueba** y **Cantidad** tienen un tipo de dato **Variant**. Esto puede sorprender si por experiencia con otros lenguajes de programación lleva a esperar que todas las variables contenidas en la misma instrucción de declaración tengan el mismo tipo que ha especificado (en este caso, **Integer**).

Tipos de datos numéricos

Visual Basic proporciona varios tipos de datos numéricos. **Integer**, **Long** (entero largo), **Single** (signo flotante de simple precisión), **Double** (signo flotante de doble precisión) y **Currency**. Utilizar un tipo de dato numérico emplea normalmente menos espacio de almacenamiento que un tipo **Variant**.

Si se sabe que una variable siempre va a almacenar números enteros (como 12) en vez de números fraccionarios (como 3.57), declárese como un tipo **Integer** o **Long**. Las operaciones con enteros son más rápidas y estos tipos consumen menos memoria que otros tipos de datos. Resultan especialmente útiles como variables de contador en bucles **For...Next**.

Si la variable contiene una fracción, declárelase como variable **Single**, **Double** o **Currency**. El tipo de dato **Currency** acepta hasta cuatro dígitos a la derecha del separador decimal y hasta quince dígitos a la izquierda; es un tipo de dato de signo fijo adecuado para cálculos monetarios. Los números de signo flotante (**Single** y **Double**) tienen más intervalo que **Currency**, pero pueden estar sujetos a pequeños errores de redondeo.

Nota Los valores de signo flotante se pueden expresar como *mmmEeee* o *mmmDeee*, donde *mmm* es la mantisa y *eee* el exponente (potencia de 10). El valor positivo más alto de un tipo de dato **Single** es 3,402823E+38 ó 3,4 veces 10 a la 38ª potencia; el valor positivo más alto de un tipo de dato **Double** es 1,79769313486232D+308 o alrededor de 1,8 veces 10 a la 308ª potencia. Si utiliza **D** para separar la mantisa y el exponente en un literal numérico, el valor se tratará como un tipo de dato **Double**. Igualmente, utilizar **E** de la misma manera hace que el valor se trate como un tipo de dato **Single**.

El tipo de dato Byte

Si la variable contiene datos binarios, declárelase como matriz de tipo **Byte** (Las matrices se describen en "Matrices", más adelante en este mismo tema). Utilizar variables **Byte** para almacenar datos binarios los preserva durante las conversiones de formato. Cuando se convierten las variables **String** entre los formatos ANSI y Unicode, los datos binarios de la variable resultan dañados. Visual Basic puede convertir automáticamente entre ANSI y Unicode al:

- Leer archivos
- Escribir archivos
- Llamar a archivos DLL
- Llamar a métodos y propiedades en objetos

Todos los operadores que funcionan con enteros funcionan con el tipo de dato **Byte** excepto el de resta unaria. Puesto que **Byte** es un tipo sin signo con el intervalo 0 - 255, no puede representar un valor negativo. Así, para la resta unaria, Visual Basic se convierte antes el tipo **Byte** en un entero con signo.

Es posible asignar todas las variables numéricas entre sí y a variables del tipo **Variant**. Visual Basic redondea en vez de truncar la parte fraccionaria de un número de signo flotante antes de asignarlo a un entero.

El tipo de dato String

Si se tiene una variable que siempre contendrá una cadena y nunca un valor numérico, puede declararla del tipo **String**:

```
Private S As String
```

Así se podrá asignar cadenas a esta variable y manipularla mediante funciones de cadena.

```
S = "Base de datos"
S = Left(S, 4)
```

De forma predeterminada, una variable o argumento de cadena es una *cadena de longitud variable*; la cadena crece o disminuye según le asigne nuevos datos. También se puede declarar cadenas de longitud fija. Especifíquese una *cadena de longitud fija* con esta sintaxis:

String * tamaño

Por ejemplo, para declarar una cadena que tiene siempre 50 caracteres de longitud, se debe utilizar un código como este:

```
Dim NombreEmp As String * 50
```

Si se asigna una cadena con menos de 50 caracteres, `NombreEmp` se rellenará con espacios en blanco hasta el total de 50 caracteres. Si asigna una cadena demasiado larga a una cadena de longitud fija, Visual Basic simplemente truncará los caracteres.

Puesto que las cadenas de longitud fija se rellenan con espacios al final, se verá que las funciones **Trim** y **RTrim**, que quitan los espacios en blanco, resultan muy útiles cuando se trabaja con ellas.

Las cadenas de longitud fija se pueden declarar en módulos estándar como **Public** o **Private**. En módulos de clase y formulario, las cadenas de longitud fija deben declararse como **Private**.

El tipo de dato Boolean

Si se tiene una variable que siempre contendrá solamente información del tipo **verdadero** y **falso**, si y no o **activado** o **desactivado**, puede ser declarada del tipo **Boolean**. El valor **predeterminado** de **Boolean** es **False**. En el siguiente ejemplo, **blnEjecutando** es una variable **Boolean** que almacena un simple sí o no.

```
Dim blnEjecutando As Boolean
    ' Comprueba si la cinta está en marcha.
    If Recorder.Direction = 1 Then
        blnEjecutando = True
    End If
```

El tipo de dato Date

Los valores de fecha y hora pueden almacenarse en el tipo de dato específico **Date** en variables **Variant**. En ambos tipos se aplican las mismas características generales a las fechas.

Cuando se convierten otros tipos de datos numéricos en **Date**, los valores que hay a la izquierda del signo decimal representan la información de fecha, mientras que los valores que hay a la derecha del signo decimal representan la hora. Medianoche es 0 y mediodía es 0.5. Los números enteros negativos representan fechas anteriores al 30 de diciembre de 1899.

El tipo de dato Object

Las variables **Object** se almacenan como direcciones de 32 bits (4 bytes) que hacen referencia a objetos dentro de una aplicación o de cualquier otra aplicación. Una variable declarada como **Object** es una variable que puede asignarse subsiguientemente (mediante la instrucción **Set**) para referirse a cualquier objeto real reconocido por la aplicación.

```
Dim objDb As Object
Set objDb = OpenDatabase("c:\vb5\biblio.mdb")
```

Convertir tipos de datos

Visual Basic proporciona varias funciones de conversión que puede utilizar para convertir valores en tipos de datos específicos. Por ejemplo, para convertir un valor a **Currency**, utilice la función **CCur**:

```
PagoPorSemana = CCur(horas * PagoPorHora)
```

Funciones de conversión	Convierten una expresión en
Cbool	Boolean
Cbyte	Byte
Ccur	Currency
Cdate	Date
Cdbl	Double
Cint	Integer
CLng	Long
CSng	Single
CStr	String
Cvar	Variant
CVErr	Error

Nota Los valores que se pasan a una función de conversión deben ser válidos para el tipo de dato de destino o se producirá un error. Por ejemplo, si se intenta convertir un tipo Long en un Integer, el tipo Long debe estar en el intervalo válido del tipo de dato Integer.

El valor Empty

A veces se necesitará saber si se ha asignado un valor a una variable existente. Una variable **Variant** tiene el valor **Empty** antes de asignarle un valor. El valor **Empty** es un valor especial distinto de 0, una cadena de longitud cero ("") o el valor Null. Puede probar el valor **Empty** con la función **IsEmpty**:

```
If IsEmpty(Z) Then Z = 0
```

Cuando un **Variant** contiene el valor **Empty**, puede ser utilizado en expresiones; se trata como un 0 o una cadena de longitud cero, dependiendo de la expresión.

El valor **Empty** desaparece tan pronto como se asigna cualquier valor (incluyendo 0, una cadena de longitud cero o **Null**) a una variable **Variant**. Puede establecer una variable **Variant** de nuevo como **Empty** si asigna la palabra clave **Empty** al **Variant**.

El valor Null

El tipo de dato **Variant** puede contener otro valor especial: **Null**. **Null** se utiliza comúnmente en aplicaciones de bases de datos para indicar datos desconocidos o que faltan. Debido a la forma en que se utiliza en las bases de datos, **Null** tiene algunas características únicas:

- Las expresiones que utilizan **Null** dan como resultado siempre un **Null**. Así, se dice que **Null** se "propaga" a través de expresiones; si cualquier parte de la expresión da como resultado un **Null**, la expresión entera tiene el valor **Null**.
- Al pasar un **Null**, un **Variant** que contenga un **Null** o una expresión que dé como resultado un **Null** como argumento de la mayoría de las funciones hace que la función devuelva un **Null**.
- Los valores **Null** se propagan a través de funciones intrínsecas que devuelven tipos de datos **Variant**.

También se puede asignar un **Null** mediante la palabra clave **Null**:

```
Z = Null
```

Se puede utilizar la función **IsNull** para probar si una variable **Variant** contiene un **Null**:

```
If IsNull(X) And IsNull(Y) Then
    Z = Null
Else
    Z = 0
End If
```

El valor Error

En un **Variant**, **Error** es un valor especial que se utiliza para indicar que se ha producido una condición de error en un procedimiento. Sin embargo, a diferencia de otros tipos de error, no se produce el tratamiento de errores a nivel normal de aplicación. Esto permite, incluso a la propia aplicación elegir alternativas basadas en el valor del error. Los valores de error se crean convirtiendo números reales en valores de error mediante la función **CVErr**.

Procedimientos

Se puede simplificar las tareas de programación si se divide los programas en componentes lógicos más pequeños. Estos componentes, llamados *procedimientos*, pueden convertirse en bloques básicos que permiten mejorar y ampliar Visual Basic.

Los procedimientos resultan muy útiles para condensar las tareas repetitivas o compartidas, como cálculos utilizados frecuentemente, manipulación de texto y controles, y operaciones con bases de datos.

Hay dos ventajas principales cuando se programa con procedimientos:

- Los procedimientos permiten dividir los programas en unidades lógicas discretas, cada una de las cuales se puede depurar más fácilmente que un programa entero sin procedimientos.
- Los procedimientos que se utilizan en un programa pueden actuar como bloques de construcción de otros programas, normalmente con pocas o ninguna modificación.

En Visual Basic se utilizan varios tipos de procedimientos:

- Procedimientos **Sub** que no devuelven un valor
- Procedimientos **Function** que devuelven un valor
- Procedimientos **Property** que pueden devolver y asignar valores, así como establecer referencias a objetos.

Procedimientos

Un procedimiento **Sub** es un bloque de código que se ejecuta como respuesta a un evento. Al dividir el código de un módulo en procedimientos **Sub**, es más sencillo encontrar o modificar el código de la aplicación.

La sintaxis de un procedimiento **Sub** es la siguiente

```
[Private]Public[Static]Sub nombre_procedimiento (argumentos)  
    instrucciones
```

End Sub

Cada vez que se llama al procedimiento se ejecutan las *instrucciones* que hay entre **Sub** y **End Sub**. Se pueden colocar los procedimientos **Sub** en módulos estándar, módulos de clase y módulos de formulario. De forma predeterminada, los procedimientos **Sub** son **Public** en todos los módulos, lo que significa que se les puede llamar desde cualquier parte de la aplicación.

Los *argumentos* de un procedimiento son como las declaraciones de variables; se declaran valores que se pasan desde el procedimiento que hace la llamada.

Resulta muy útil en Visual Basic distinguir entre dos tipos de procedimientos **Sub**, *procedimientos generales* y *procedimientos de evento*.

Procedimientos generales

Un procedimiento general indica a la aplicación cómo realizar una tarea específica. Una vez que se define un procedimiento general, se le debe llamar específicamente desde la aplicación. Por el contrario, un procedimiento de evento permanece inactivo hasta que se le llama para responder a eventos provocados por el usuario o desencadenados por el sistema.

¿Por qué crear procedimientos generales? Una razón es que muchos procedimientos de evento distintos pueden necesitar que se lleven a cabo las mismas acciones. Es una buena estrategia de programación colocar las instrucciones comunes en un procedimiento distinto (un procedimiento general) y hacer que los procedimientos de evento lo llamen. Esto elimina la necesidad de duplicar código y también hace que la aplicación sea más fácil de mantener.

Procedimientos de evento

Cuando un objeto en Visual Basic reconoce que se ha producido un evento, llama automáticamente al procedimiento de evento utilizando el nombre correspondiente al evento. Como el nombre establece una asociación entre el objeto y el código, se dice que los procedimientos de evento están adjuntos a formularios y controles.

- Un procedimiento de evento de un control combina el nombre real del control (especificado en la propiedad **Name**), un carácter de subrayado (_) y el nombre del evento. Por ejemplo, si se desea que un botón de comando llamado `cmdPlay` llame a un procedimiento de evento cuando se haga clic en él, utilice el procedimiento `cmdPlay_Click`.
- Un procedimiento de evento de un formulario combina la palabra "Form", un carácter de subrayado y el nombre del evento. Si desea que un formulario llame a un procedimiento de evento cuando se hace clic en él, utilícese el procedimiento `Form_Click` (Como los controles, los formularios tienen nombres únicos, pero no se utilizan en los nombres de los procedimientos de evento.) Si está utilizando el formulario MDI, el procedimiento de evento combina la palabra "MDIForm", un carácter de subrayado y el nombre del evento, como `MDIForm_Load`.

Todos los procedimientos de evento utilizan la misma sintaxis general.

Sintaxis de un evento de control

Sintaxis de un evento de formulario

```
Private Sub nombrecontrol_nombreevento (argumentos)
```

Instrucciones

```
End Sub
```

```
Private Sub Form_nombreevento (argumentos)
```

Instrucciones

```
End Sub
```

Aunque se puede escribir procedimientos de evento nuevos, es más sencillo utilizar los procedimientos de código que facilita Visual Basic, que incluyen automáticamente los nombres correctos de procedimiento.

También es conveniente establecer la propiedad **Name** de los controles antes de empezar a escribir los procedimientos de evento para los mismos. Si se cambia el nombre de un control tras vincularle un procedimiento, se deberá cambiar también el nombre del procedimiento para que coincida con el nuevo nombre del control. De lo contrario, Visual Basic no será capaz de hacer coincidir el control con el procedimiento. Cuando el nombre de un procedimiento no coincide con el nombre de un control, se convierte en un procedimiento general.

funciones

Visual Basic incluye funciones incorporadas o intrínsecas, como `Sqr`, `Cos` o `Chr`. Además, se puede utilizar la instrucción `Function` para escribir procedimientos `Function`.

La sintaxis de un procedimiento `Function` es la siguiente:

```
[Private|Public][Static]Function nombre_procedimiento (argumentos) [As tipo]
```

instrucciones

End Function

Al igual que un procedimiento `Sub`, un procedimiento `Function` es un procedimiento diferente que puede tomar argumentos, realizar una serie de instrucciones y cambiar el valor de los argumentos. A diferencia de los procedimientos `Sub`, los procedimientos `Function` pueden devolver un valor al procedimiento que realiza la llamada. Hay tres diferencias entre los procedimientos `Sub` y `Function`:

- Generalmente, se llama a una función incluyendo el nombre y los argumentos del procedimiento en la parte derecha de una instrucción o expresión mayor (*valor_retorno = función()*).
- Los procedimientos `Function` tienen tipos de datos, al igual que las variables. Esto determina el tipo del valor de retorno. (En ausencia de la cláusula `As`, el tipo es el tipo predeterminado `Variant`.)
- Se devuelve un valor asignándole al propio *nombre_procedimiento*. Cuando el procedimiento `Function` devuelve un valor, se puede convertir en parte de una expresión mayor.

Por ejemplo, se podría escribir una función que calculara el tercer lado, o hipotenusa, de un triángulo rectángulo, dados los valores de los otros dos lados:

```
Function Hipotenusa (A As Integer, B As Integer)As String
    Hipotenusa = Sqr(A ^ 2 + B ^ 2)
End Function
```

Se llama a un procedimiento `Function` de la misma forma que a las funciones incorporadas en Visual Basic:

```
Label1.Caption = Hipotenusa(CInt(Text1.Text),
CInt(Text2.Text))
strX = Hipotenusa(Width, Height)
```

Introducción a los operadores.

En Visual Basic los operadores son símbolos especiales, que al combinarse convenientemente con valores, producen un nuevo valor como resultado. Los tipos de operadores a tratar seguidamente son:

Multiplicativos	Multiplicación, división y módulo
Aditivos	Suma, resta
Relación	Es mayor que, es menor que
Igualdad	Es igual a, es distinto a
Lógico	y, o
Unario	Menos unario, negación lógica

Estructuras de control.

Las estructuras de control permiten controlar el flujo de ejecución del programa. Si no se controla mediante instrucciones de control de flujo, la lógica del programa fluirá por las instrucciones de izquierda a derecha y de arriba a abajo. Aunque se pueden escribir algunos programas sencillos con un flujo unidireccional y aunque se puede controlar parte del flujo mediante operadores para regular la precedencia de las operaciones, la mayor parte del poder y utilidad de un lenguaje de programación deriva de su capacidad de cambiar el orden de las instrucciones mediante estructuras y bucles

Básicamente se cuenta con las siguientes estructuras de control:

- **Estructuras de decisión** Introducción a las estructuras de decisión utilizadas para bifurcar.
- **Estructuras de bucle** Introducción a las estructuras de bucle utilizadas para repetir procesos.

Estructuras de decisión.

Los procedimientos de pueden probar condiciones y, dependiendo de los resultados de la prueba, realizar diferentes operaciones. Entre las estructuras de decisión que acepta se incluyen las siguientes:

- **If...Then**
- **If...Then...Else**
- **Select Case**

If...Then

La estructura **If...Then** es usada para ejecutar una o más instrucciones basadas en una condición. Se puede utilizar la sintaxis de una línea o un *bloque* de varias líneas:

If condición Then instrucción

*If condición Then
instrucciones*

End If

Condición normalmente es una comparación, pero puede ser cualquier expresión que dé como resultado un valor numérico. El lenguaje interpreta este valor como **True** o **False**; un valor numérico cero es **False** y se considera **True** cualquier valor numérico distinto de cero. Si *condición* es **True**, se ejecuta todas las *instrucciones* que siguen a la palabra clave **Then**. Se puede utilizar la sintaxis de una línea o de varias líneas para ejecutar una instrucción basada en una condición (estos dos ejemplos son equivalentes):

```
If cualquierFecha < Now Then cualquierFecha = Now
```

```
If cualquierFecha < Now Then
    cualquierFecha = Now
End If
```

Obsérvese que el formato de una única línea de **If...Then** no utiliza la instrucción **End If**. Si desea ejecutar más de una línea de código cuando *condición* sea **True**, se debe utilizar la sintaxis de bloque de varias líneas **If...Then...End If**.

```
If cualquierFecha < Now Then
    cualquierFecha = Now
    Timer1.Enabled = False ' Desactiva el control Timer.
End If
```

If...Then...Else

Se debe utilizar un bloque **If...Then...Else** para definir varios bloques de instrucciones, uno de los cuales se ejecutará:

```
If condición1 Then
    [bloque de instrucciones 1]

    [Elseif condición2 Then
        [bloque de instrucciones 2]]

    [Else
        [bloque de instrucciones n]]
End If
```

Se evalúa primero *condición1*. Si es **False**, se procede a evaluar *condición2* y así sucesivamente, hasta que encuentre una condición **True**. Cuando se encuentra una condición **True**, se ejecuta el bloque de instrucciones correspondientes y después ejecuta el código que sigue a **End If**. Opcionalmente, puede incluir un bloque de instrucciones **Else**, que se ejecutará si ninguna de las condiciones es **True**.

If...Then...Elseif es un caso especial de **If...Then...Else** ya que puede tener cualquier número de cláusulas **Elseif** o ninguna. Puede incluir una cláusula **Else** sin tener en cuenta si tiene o no cláusulas **Elseif**.

Por ejemplo, la aplicación podría realizar distintas acciones dependiendo del control en que se haya hecho clic de una matriz de controles de menú:

```
Private Sub mnuCut_Click (Index As Integer)
    If Index = 0 Then ' Comando Cortar.
        CopyActiveControl ' Llama a procedimientos generales.
        ClearActiveControl
    ElseIf Index = 1 Then ' Comando Copiar.
        CopyActiveControl
    ElseIf Index = 2 Then ' Comando Borrar.
        ClearActiveControl
    Else ' Comando Pegar.
        PasteActiveControl
    End If
End Sub
```

Se puede agregar más cláusulas **Elseif** a la estructura **If...Then**. Sin embargo, esta sintaxis puede resultar tediosa de escribir cuando cada **Elseif** compara la misma expresión con un valor distinto. Para estas situaciones, Se puede utilizar la estructura de decisión **Select Case**.

Select Case

La estructura **Select Case** es una alternativa a **If...Then...Else** para ejecutar selectivamente un bloque de instrucciones entre varios bloques de instrucciones. La instrucción **Select Case** ofrece posibilidades similares a la instrucción **If...Then...Else**, pero hace que el código sea más legible cuando hay varias opciones.

La estructura **Select Case** funciona con una única expresión de prueba que se evalúa una vez solamente, al principio de la estructura. Se compara el resultado de esta expresión con los valores de cada **Case** de la estructura. Si hay una coincidencia, ejecuta el bloque de instrucciones asociado a ese **Case**:

Select Case *expresión_prueba*

[**Case** *lista_expresiones1*
{*bloque de instrucciones 1*}]

[**Case** *lista_expresiones2*
{*bloque de instrucciones 2*}]

[**Case Else**
{*bloque de instrucciones n*}]

End Select

Cada *lista_expresiones* es una lista de uno o más valores. Si hay más de un valor en una lista, se separan los valores con comas. Cada *bloque de instrucciones* contiene cero o más instrucciones. Si más de un **Case** coincide con la expresión de prueba, sólo se ejecutará el bloque de instrucciones asociado con la primera coincidencia. Se ejecuta las instrucciones de la cláusula (opcional) **Case Else** si ningún valor de la lista de expresiones coincide con la expresión de prueba.

Por ejemplo, si se agrega otro comando al menú **Edición** en el ejemplo **If...Then...Else** Podría agregar otra cláusula **Elseif** o podría escribir la función con **Select Case**:

```
Private Sub mnuCut_Click (Index As Integer)
  Select Case Index
    Case 0 ' Comando Cortar.
      CopyActiveControl ' Llama a procedimientos generales.
      ClearActiveControl
    Case 1 ' Comando Copiar.
      CopyActiveControl
    Case 2 ' Comando Borrar.
      ClearActiveControl
    Case 3 ' Comando Pegar.
      PasteActiveControl
    Case Else
      frmFind.Show ' Muestra el cuadro de diálogo Buscar.
  End Select
End Sub
```

~~End Sub~~

La estructura **Select Case** evalúa una expresión cada vez al principio de la estructura. Por el contrario, la estructura **If...Then...Else** puede evaluar una expresión diferente en cada instrucción **Elseif**. Sólo se puede sustituir una estructura **If...Then...Else** con una estructura **Select Case** si la instrucción **If** y cada instrucción **Elseif** evalúa la misma expresión.

Estructuras de bucles

Es difícil aburrir a una computadora con una rutina. Uno de los puntos fuertes de esta es su capacidad para repetir tareas con rapidez, precisión y sin quejarse. Puede decirse a una computadora que haga lo mismo millones de veces. El lenguaje Visual Basic dispone de varias sentencias que permiten programar acciones repetidas, que se verán a continuación.

Las estructuras de bucle le permiten ejecutar una o más líneas de código repetidamente. Las estructuras de bucle que acepta Visual Basic son:

- **Do...Loop**
- **For...Next**
- **For Each...Next**

Do...Loop

Se deberá usar el bucle **Do** para ejecutar un bloque de instrucciones un número indefinido de veces. Hay algunas variantes en la instrucción **Do...Loop**, pero cada una evalúa una condición numérica para determinar si continúa la ejecución. Como ocurre con **If...Then**, la *condición* debe ser un valor o una expresión que dé como resultado **False** (cero) o **True** (distinto de cero).

En el ejemplo de **Do...Loop** siguiente, las *instrucciones* se ejecutan siempre y cuando *condición* sea **True**:

Do While *condición*

instrucciones

Loop

Quando se ejecuta este bucle **Do**, primero se evalúa *condición*. Si *condición* es **False** (cero), se salta todas las instrucciones. Si es **True** (distinto de cero), se ejecutan las instrucciones y se vuelve a la instrucción **Do While** y se prueba la condición de nuevo.

Por tanto, el bucle se puede ejecutar cualquier número de veces, siempre y cuando *condición* sea distinta de cero o **True**. Nunca se ejecutan las instrucciones si *condición* es **False** inicialmente. Por ejemplo, este procedimiento cuenta las veces que se repite una cadena de destino dentro de otra cadena repitiendo el bucle tantas veces como se encuentre la cadena de destino:

```
Function ContarCadenas (cadenalarga, destino)
    Dim posición, contador
    posición = 1
    Do While InStr(posición, cadenalarga, destino)
        posición = InStr(posición, cadenalarga, destino) + 1
        contador = contador + 1
    Loop
    ContarCadenas = contador
End Function
```

Si la cadena de destino no está en la otra cadena, `InStr` devuelve 0 y no se ejecuta el bucle.

Otra variante de la instrucción `Do...Loop` ejecuta las instrucciones primero y prueba *condición* después de cada ejecución. Esta variación garantiza al menos una ejecución de *instrucciones*:

Do
 instrucciones

Loop While *condición*

Hay otras dos variantes análogas a las dos anteriores, excepto en que repiten el bucle siempre y cuando *condición* sea `False` en vez de `True`.

Hace el bucle cero o más veces **Hace el bucle al menos una vez**

Do Until *condición*

Instrucciones

Loop Do

Instrucciones

Loop Until *condición*

For...Next

Los bucles `Do` funcionan bien cuando no se sabe cuántas veces se necesitará ejecutar las instrucciones del bucle. Sin embargo, cuando se sabe que se van a ejecutar las instrucciones un número determinado de veces, es mejor elegir el bucle `For...Next`. A diferencia del bucle `Do`, el bucle `For` utiliza una variable llamada contador que incrementa o reduce su valor en cada repetición del bucle. La sintaxis es la siguiente:

For *contador* = *iniciar* **To** *finalizar* [**Step** *incremento*]

Instrucciones

Next [*contador*]

Los argumentos *contador*, *iniciar*, *finalizar* e *incremento* son todos numéricos.

Nota El argumento *incremento* puede ser positivo o negativo. Si *incremento* es positivo, *iniciar* debe ser menor o igual que *finalizar* o no se ejecutarán las instrucciones del bucle. Si *incremento* es negativo, *iniciar* debe ser mayor o igual que *finalizar* para que se ejecute el cuerpo del bucle. Si no se establece *Step*, el valor predeterminado de *incremento* es 1.

Al ejecutar el bucle `For`, Visual Basic:

1. Establece *contador* al mismo valor que *iniciar*.
2. Comprueba si *contador* es mayor que *finalizar*. Si lo es, se sale del bucle.
(Si *incremento* es negativo, Visual Basic comprueba si *contador* es menor que *finalizar*.)
3. Ejecuta *instrucciones*.
4. Incrementa *contador* en 1 o en *instrucciones*, si se especificó.
5. Repite los pasos 2 a 4.

Este código imprime los nombres de todas las fuentes de pantalla disponibles:

```
Private Sub Form_Click ()
    Dim I As Integer
    For i = 0 To Screen.FontCount
        Print Screen.Fonts(i)
    Next
End Sub
```

For Each...Next

El bucle **For Each...Next** es similar al bucle **For...Next**, pero repite un grupo de instrucciones por cada elemento de una colección de objetos o de una matriz en vez de repetir las instrucciones un número especificado de veces. Esto resulta especialmente útil si no sabe cuántos elementos hay en la colección.

He aquí la sintaxis del bucle **For Each...Next**:

For Each *elemento* **In** *grupo*

Instrucciones

Next *elemento*

Por ejemplo, el siguiente procedimiento **Sub** abre Biblio.mdb y agrega el nombre de cada tabla a un cuadro de lista.

```
Sub ListTableDefs()
    Dim objDb As Database
    Set objDb = OpenDatabase("c:\vb\biblio.mdb", True, False)
    For Each TableDef In objDb.TableDefs()
        List1.AddItem TableDef.Name
    Next TableDef
End Sub
```

Las restricciones siguientes son aplicables al bucle **For Each...Next**:

- Para las colecciones, *elemento* sólo puede ser una variable **Variant**, una variable **Object** genérica o un objeto mostrado en el Examinador de objetos.
- Para las matrices, *elemento* sólo puede ser una variable **Variant**. No puede utilizar **For Each...Next** con una matriz de tipos definidos por el usuario porque un **Variant** no puede contener un tipo definido por el usuario.

Matrices.

Si se ha programado en otros lenguajes, probablemente se conocerá el concepto de matrices. Las matrices permiten hacer referencia por el mismo nombre a una serie de variables y utilizar un número (índice) para distinguirlas. Esto ayuda a crear código más pequeño y simple en muchas situaciones, ya que se puede establecer bucles que traten de forma eficiente cualquier número de casos mediante el número del índice. Las matrices tienen un límite superior e inferior y los elementos de la matriz son contiguos dentro de esos límites. Puesto que Visual Basic asigna espacio para cada número de índice, evítense declarar las matrices más grandes de lo necesario.

Todos los elementos de una matriz tienen el mismo tipo de datos. Por supuesto, cuando el tipo de dato es **Variant**, los elementos individuales pueden contener distintas clases de datos (objetos, cadenas, números, etc). Puede ser declarada una matriz de cualquiera de los tipos de datos fundamentales.

En Visual Basic hay dos tipos de matrices: las *matrices de tamaño fijo* que tienen siempre el mismo tamaño y las *matrices dinámicas* cuyo tamaño cambia en tiempo de ejecución.

Declaración de matrices de tamaño fijo

Hay tres formas de declarar una matriz de tamaño fijo, dependiendo del alcance que desee que tenga la matriz:

- Para declarar una *matriz pública*, se debe utilizar la instrucción **Public** en la sección Declaraciones de un módulo para declarar la matriz.
- Para crear una *matriz a nivel de módulo*, se debe utilizar la instrucción **Private** en la sección Declaraciones de un módulo para declarar la matriz.
- Para crear una *matriz local*, se debe utilizar la instrucción **Private** en un procedimiento para declarar la matriz.

Establecimiento de los límites superior e inferior

Cuando se declare una matriz, se debe poner el nombre de la matriz y el límite superior entre paréntesis. El límite superior no puede exceder el intervalo de un tipo de dato **Long** (-2.147.483.648 a 2.147.483.647). Por ejemplo, estas declaraciones de matrices pueden aparecer en la sección Declaraciones de un módulo

```
Dim Contadores(14) As Integer ' 15 elementos.
Dim Sumas(20) As Double      ' 21 elementos.
```

Para crear una matriz pública, simplemente utilícese **Public** en lugar de **Dim**

```
Public Contadores(14) As Integer
Public Sumas(20) As Double
```

Las mismas declaraciones dentro de un procedimiento utilizan **Dim**:

```
Dim Contadores(14) As Integer
Dim Sumas(20) As Double
```

La primera declaración crea una matriz de 15 elementos, con números de índice que van de 0 a 14. La segunda crea una matriz de 21 elementos, con números de índice que van de 0 a 20. El límite inferior predeterminado es 0.

Para especificar el límite inferior, proporciónese explícitamente (como tipo de dato **Long**) mediante la palabra clave **To**:

```
Dim Contadores(1 To 15) As Integer
Dim Sumas(100 To 120) As String
```

En las anteriores declaraciones, los números de índice de **Contadores** van de 1 a 15 y los números de índice de **Sumas** van de 100 a 120.

Matrices que contienen otras matrices

Es posible crear una matriz **Variant** y llenarla con otras matrices de distintos tipos de datos. El código siguiente crea dos matrices, una que contiene enteros y otra que contiene cadenas. Se declara entonces una tercera matriz **Variant** y se llena con las matrices de enteros y cadenas.

```
Private Sub Command1_Click()

    Dim intX As Integer      ' Declara la variable de contador.

    ' Declara y llena una matriz de enteros.
    Dim contadoresA(5) As Integer
    For intX = 0 To 4
        contadoresA(intX) = 5
    Next intX
    ' Declara y llena una matriz de cadenas.
    Dim contadoresB(5) As String
    For intX = 0 To 4
        contadoresB(intX) = "adiós"
    Next intX

    Dim arrX(2) As Variant 'Declara una matriz nueva de dos
    miembros.
    arrX(1) = contadoresA() ' Llena la matriz con las otras
    matrices.
    arrX(2) = contadoresB()
    MsgBox arrX(1)(2) ' Muestra un miembro de cada matriz.
    MsgBox arrX(2)(3)

End Sub
```

Matrices multidimensionales.

A veces se necesitará hacer un seguimiento de la información relacionada de una matriz. Por ejemplo, para hacer un seguimiento de cada pixel de la pantalla del equipo, se necesitará referirse a sus coordenadas X e Y. Esto se puede hacer mediante una matriz multidimensional para almacenar los valores.

En Visual Basic se puede declarar matrices de varias dimensiones. Por ejemplo, la instrucción siguiente declara una matriz bidimensional de 10 por -10 en un procedimiento:

```
Static MatrizA(9, 9) As Double
```

Se puede declarar una o ambas dimensiones con límites inferiores explícitos

```
Static MatrizA(1 To 10, 1 To 10) As Double
```

Se puede hacer que tenga más de dos dimensiones. Por ejemplo:

```
Dim MultiD(3, 1 To 10, 1 To 15)
```

Esta declaración crea una matriz que tiene tres dimensiones con tamaños de 4 por 10 por 15. El número total de elementos es el producto de las tres dimensiones, es decir, 600.

Nota Cuando se comience a agregar dimensiones a una matriz, el almacenamiento total que necesita la matriz se incrementa considerablemente, por lo que se debe utilizar las matrices multidimensionales con sumo cuidado. Se debe tener especial cuidado con las matrices **Variant**, ya que son más grandes que otros tipos de datos.

Sistemas de Bases de Datos

• Definición y Características de un SBD.

Un *Sistema de Bases de Datos (SBD)* es una serie de recursos para manejar grandes volúmenes de información, sin embargo no todos los sistemas que manejan información son bases de datos.

Un sistema de bases de datos debe responder a las siguientes *características*:

- **Independencia de los Datos.** Es decir, que los datos no dependen del programa y por tanto cualquier aplicación puede hacer uso de los datos.
- **Reducción de la Redundancia.** Llamamos redundancia a la existencia de duplicación de los datos, al reducir ésta al máximo conseguimos un mayor aprovechamiento del espacio y además evitamos que existan *inconsistencias* entre los datos. Las *inconsistencias* se dan cuando nos encontramos con datos contradictorios.
- **Seguridad.** Un SBD debe permitir que tengamos un control sobre la seguridad de los datos.

• Recursos que componen un SBD

Un SBD estará formado por:

- Personas
- Máquinas
- Programas
- Datos

1 Los Datos.

Es lo que se conoce como base de datos propiamente dicha. Para manejar estos datos utilizamos una serie de programas.

2.- Los Programas.

Son los encargados de manejar los datos, son conocidos como DBMS (*Data Base Management System*) o también SGBD (*Sistema Gestor de Base de Datos*). Los DBMS tienen dos funciones principales que son:

- La definición de las estructuras para almacenar los datos.
- La manipulación de los datos.

• Distintos Niveles de un SBD

Los SBD pueden ser estudiados desde 3 niveles distintos:

1.- Nivel Físico.

Es el nivel real de los datos almacenados. Es decir como se almacenan los datos, ya sea en registros, o como sea. Este nivel es usado por muy pocas personas que deben estar cualificadas para ello. Este nivel lleva asociada una representación de los datos, que es lo que denominamos Esquema Físico.

2.- Nivel Conceptual.

Es el correspondiente a una visión de la base de datos desde el punto de visto del mundo real. Es decir tratamos con la entidad u objeto representado, sin importarnos como está representado o almacenado. Este nivel lleva asociado el Esquema Conceptual.

3.- Nivel Visión.

Son partes del esquema conceptual. El nivel conceptual presenta toda la base de datos, mientras que los usuarios por lo general sólo tienen acceso a pequeñas parcelas de ésta. El nivel visión es el encargado de dividir estas parcelas. Un ejemplo sería el caso del empleado que no tiene porqué tener acceso al sueldo de sus compañeros o de sus superiores. El esquema asociado a éste nivel es el Esquema de Visión.

Los 3 niveles vistos, componen lo que conocemos como arquitectura de base de datos a 3 niveles.

A menudo el nivel físico no es facilitado por muchos DBMS, esto es, no permiten al usuario elegir como se almacenan sus datos y vienen con una forma estándar de almacenamiento y manipulación de los datos.

La arquitectura a 3 niveles se puede representar como sigue
 Subesquema de Visión, Subesquema de Visión, Subesquema de Visión, ...
 Esquema Conceptual
 Esquema Físico

• Modelos de Datos

Para representar el mundo real a través de esquemas conceptuales se han creado una serie de modelos:

Modelización

Existen multitud de estos modelos que se conocen como Modelos de Datos, algunos de estos modelos son:

- Modelo Relacional de Datos
- Modelo de Red
- Modelo Jerárquico

1.- Modelo Relacional de Datos.

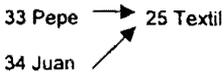
Representa al mundo real mediante tablas relacionadas entre sí por columnas comunes. Ejemplo.:

Num_empleado	Nombre	Sección
33	Pepe	25
34	Juan	25

Num_sección	Nombre
25	Textil
26	Pintura

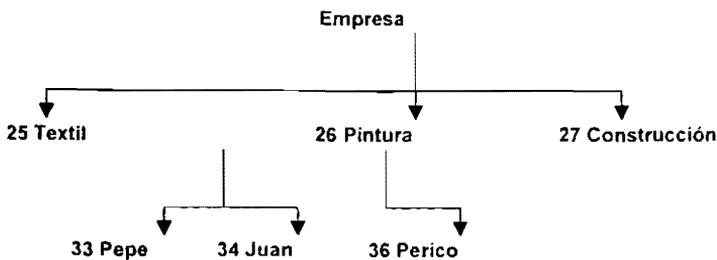
2.- Modelo de Red.

Representamos al mundo real como registros lógicos que representan a una entidad y que se relacionan entre sí por medio de flechas. Ejemplo.:



3.- Modelo Jerárquico.

Tiene forma de árbol invertido. Un padre puede tener varios hijos pero cada hijo sólo puede tener un padre. Ejemplo:



• Personas Relacionadas con un SBD.

1.- Administrador de Base de Datos (DBA).

Define los esquemas de la base de datos; estructuras y esquemas de los 3 niveles. Más que una persona suele ser un grupo de personas.

2.- Programador de Aplicaciones.

Utilizando un lenguaje de alto nivel y llamadas en DML crean programas para usar la base de datos.

3.- Usuarios Casuales.

Son usuarios que tienen conocimientos de los DL, hacen uso de los DML de modo interactivo (es decir a través del procesador de consultas)

4.- Usuarios Ingenuos.

Emplean el SBD sin conocimientos de informática, es decir emplean los programas de aplicación.

ANEXO D

CODIGO DE SIVEA.

FORMA: CODIGO.

Dim actual As Integer
Dim anterior As Double

Dim pag As Integer
Dim ant As Integer
Dim tc(12) As String
Dim i As Integer

```
Private Sub CALCOL()
    Co0 = codigo(0).BackColor
    Co1 = codigo(1).BackColor
    Co2 = codigo(2).BackColor
    Co3 = codigo(3).BackColor
    vez = 0
    valorf = ""
    Do While vez < 2
        If vez = 0 Then
            colt = Co0
        Else
            colt = Co1
        End If
        Select Case colt
            Case Negro
                p = "0"
            Case cafe
                p = "1"
            Case rojo
                p = "2"
            Case Naranja
                p = "3"
            Case Amarillo
                p = "4"
            Case Verde
                p = "5"
            Case Azul
                p = "6"
            Case Violeta
                p = "7"
            Case Gris
                p = "8"
            Case Blanco
                p = "9"
        End Select
        vez = vez + 1
        valorf = valorf + p
    Loop
    valof = Val(valorf)
    Select Case Co2
        Case Negro
            m = 1
            mt$ = "-"
        Case cafe
            m = 10
            mt$ = "-."
        Case rojo
```

```

        m = 100
        mt$ = "-."
    Case Naranja
        m = 1
        mt$ = "Kilo"
    Case Amarillo
        m = 10
        mt$ = "Kilo"
    Case Verde
        m = 100
        mt$ = "Kilo"
    Case Azul
        m = 1
        mt$ = "Mega"
    Case oro
        m = 0.1
    Case plata
        m = 0.01
    End Select
    valf = valof * m
    text1(0).Text = valf
    Select Case Co3
        Case oro
            tov$ = "5"
        Case plata
            tov$ = "10"
        Case sincolor
            tov$ = "20"
    End Select
    text1(2).Visible = True
    text1(1).Visible = True
    text1(1).Text = tov$
    text1(2).Text = mt$
End Sub

Private Sub AniPushButton1_Click()
End Sub

Private Sub cancelcol_Click(Index As Integer)
    If Index = 0 Then
        PANTAC.Visible = False
    Else
        pantc1.Visible = False
    End If
    codigo(actual).BackColor = anterior
End Sub

Private Sub codcol_Click(Index As Integer)
    Select Case Index
        Case 0
            Col = Negro
        Case 1
            Col = cafe
        Case 2
            Col = rojo
        Case 3
            Col = Naranja
```

```

    Case 4
        Col = Amarillo
    Case 5
        Col = Verde
    Case 6
        Col = Azul
    Case 7
        Col = Violeta
    Case 8
        Col = Gris
    Case 9
        Col = Blanco
    Case 10
        Col = oro
    Case 11
        Col = plata
    Case 12
        Col = sincolor
    Case 13
        Col = oro
    Case 14
        Col = plata
    End Select
    codigo(actual).BackColor = Col
    PANTAC.Visible = False
    pantc1.Visible = False
End Sub

Private Sub codigo_Click(Index As Integer)
    actual = Index
    anterior = codigo(actual).BackColor
    If Index = 3 Then
        pantc1.Visible = True
        PANTAC.Visible = False
    Else
        pantc1.Visible = False
    End Select
    Select Case Index
        Case 0
            If = 1275
        Case 1
            If = 1550
        Case 2
            If = 2100
    End Select
    PANTAC.Visible = True
    If Index = 2 Then

        p3.Visible = True
        p4.Visible = False
    Else
        p3.Visible = False
        p4.Visible = True
    End If

    PANTAC.Left = If
End If
End Sub
```

```

Private Sub codigo_MouseMove(Index
As Integer, Button As Integer, Shift As
Integer, X As Single, Y As Single)
    pos = codigo(Index).Left
    Select Case Index
    Case 0
        X = pos
        tex = "Primera cifra significativa"
    Case 1
        X = pos + 450
        tex = "Segunda cifra significativa"
    Case 2
        X = pos + 900
        tex = "Factor multiplicativo"
    Case 3
        X = pos + 1600
        tex = "Tolerancia admisible "
    End Select
    FLECHA.Left = X
    FLECHA.Caption = tex
End Sub

Private Sub Color_MouseMove(Index
As Integer, Button As Integer, Shift As
Integer, X As Single, Y As Single)
    If X > 75 And X < 1000 Then
        PColor.Visible = True
        Select Case Index
        Case 0
            mt = "Plata"
            mt1 = "-"
            mt2 = "0.01"
            mt3 = "± 5 %"
        Case 1
            mt = "Oro"
            mt1 = "-"
            mt2 = "0.1"
            mt3 = "± 10 %"
        Case 2
            mt = "Negro"
            mt1 = "0"
            mt2 = "1"
            mt3 = "-"
        Case 3
            mt = "Cafe"
            mt1 = "1"
            mt2 = "10"
            mt3 = "-"
        Case 4
            mt = "Rojo"
            mt1 = "2"
            mt2 = "100"
            mt3 = "-"
        Case 5
            mt = "Naranja"
            mt1 = "3"
            mt2 = "1.000"
            mt3 = "-"
        Case 6
            mt = "Amarillo"
            mt1 = "4"
            mt2 = "10.000"
            mt3 = "-"
        Case 7
            mt = "Verde"
            mt1 = "5"
            mt2 = "100.000"
            mt3 = "-"
        Case 8
            mt = "Azul"
            mt1 = "6"
            mt2 = "1.000.000"
            mt3 = "-"
        Case 9
            mt = "Violeta"
            mt1 = "7"
            mt2 = "-"
            mt3 = "-"
        Case 10
            mt = "Gris"
            mt1 = "8"
            mt2 = "-"
            mt3 = "-"
        Case 11
            mt = "Blanco"
            mt1 = "9"
            mt2 = "-"
            mt3 = "-"
        Case 12
            mt = "Sin color"
            mt1 = "-"
            mt2 = "-"
            mt3 = "± 20 %"
        End Select
        mtit.Caption = mt
        Mtex(0).Caption = mt1
        Mtex(1).Caption = mt2
        Mtex(2).Caption = mt3
        muestra.BackColor =
        Color(Index).BackColor
        Shape9.FillColor =
        Color(Index).BackColor
        mcom.SetFocus
    End If
End Sub

Private Sub combot_Click()
    FDemo.Show
    FDemo.Panel1.Caption = "George
    Simon Ohm"
    FDemo.foto.Picture =
    COMBOT.Picture
    pelicula = 1
    tc(0) = " ( 1 7 8 7 - 1 8
    5 4 ) "
    tc(1) = " Físico alemán nacido en
    Erlangen."
    tc(2) = " Descubrió las propiedades
    cuantitativas "
    tc(3) = " de las corrientes eléctricas y
    formuló "
    tc(4) = "la ley que lleva su nombre
    Autor de "
    tc(5) = "numerosas memorias e
    importantes obras ."
    tc(6) = "entre ellas . "Teoria
    matemática de las "
    tc(7) = "corrientes eléctricas "
    FDemo.lab2.Text = ""
    For i = 0 To 12
        FDemo.lab2.Text = FDemo.lab2.Text +
        tc(i)
    Next i
End Sub

Private Sub comfin_Click()
    Unload Me
End Sub

Private Sub f1_Click(PreviousTab As
Integer)
    pag = F1 Tab
End Sub

Private Sub Form_Load()
    lista1(0).AddItem "-"
    lista1(0).AddItem "Kilo"
    lista1(0).AddItem "Mega"
    lista1(1).AddItem "5"
    lista1(1).AddItem "10"
    lista1(1).AddItem "20"
End Sub

Private Sub lab1_Click(Index As Integer,
Value As Integer)
    If Index = 0 Then
        RESISTOR.Enabled = True
        valor.Enabled = False
        FLECHA.Visible = True
    Else
        RESISTOR.Enabled = False
        valor.Enabled = True
        text1(0).SetFocus
        FLECHA.Visible = False
    End If
End Sub

Private Sub Label7_Click()
End Sub

Private Sub lista1_Click(Index As
Integer)
    text1(1).Visible = False
    text1(2).Visible = False
End Sub

Private Sub lista1_KeyPress(Index As
Integer, KeyAscii As Integer)
    If Index = 0 Then
        If KeyAscii = 13 Then
            lista1(1).SetFocus
        End If
    Else
        If KeyAscii = 13 Then
            CALCULA
        End If
    End If
End Sub

Private Sub mcom_Click()
    PColor.Visible = False
End Sub

Private Sub com1_Click(Index As
Integer)
    Select Case Index
    Case 0
        Panel1(0).Visible = True
    Case 1
        Panel1(0).Visible = False
    Case 2
        End
    End Select
End Sub

```



```

Msg = "No se ha asignado un
valor "
MsgBox Msg, 54, " Error de
datos"
Else
If v1 = "" Then
Msg = "El factor multiplicativo no
ha sido asignado"
MsgBox Msg, 54, " Error de
datos"
Else
If v2 = "" Then
Msg = "La tolerancia no ha sido
especificada"
MsgBox Msg, 54, " Error de
datos"
Else
CALCULA
End If
End If
End If
RESISTOR.Enabled = False
End If
End Sub

Private Sub SSPanel6_Click()
pizarron.Visible = False
End Sub

Private Sub Text1_GotFocus(Index As
Integer)
If Index = 0 Then
text1(0).Text = ""
RESISTOR.Enabled = False
End If
End Sub

Private Sub text1_KeyPress(Index As
Integer, KeyAscii As Integer)
If Index = 0 Then
If KeyAscii = 13 Then
lista1(0).SetFocus
End If
End If
End Sub

Private Sub Tono(digito As Integer)
Select Case digito
Case 0
color1 = Negro
Case 1
color1 = cafe
Case 2
color1 = rojo
Case 3
color1 = Naranja
Case 4
color1 = Amarillo
Case 5
color1 = Verde
Case 6
color1 = Azul
Case 7
color1 = Violeta
Case 8
color1 = Gris
Case 9
color1 = Blanco
End Select
End Sub

```

```

FORMA:FRMACCESO.

Dim VECECLAVE As Integer
Sub ChecaCla()
'Checa que el nombre se encuentre en
la base de datos

If
Trim(DATACLAVE.Recordset(2).Value)
= Trim(CBID.Text) Then
If
Trim(DATACLAVE.Recordset(3).Value)
= Left(Trim(TXTCUENTA.Text), 9) Then
NOCUENTA =
TXTCUENTA.Text
X =
DATACLAVE.Recordset(1).Value
ACceso = Mid(X, 1, 2)
UnidadAyu =
datayu.Recordset(0).Value
SIVEA.Show

Else
MENSA = sspcuenta.Caption
MsgBox "El " & MENSA & "
no es valido", 64, "Atención"
TXTCUENTA.SetFocus
End If

Else
MsgBox "LA CLAVE NO
CORRESPONDE AL USUARIO". 64,
"Atención"
TXTPASS.SetFocus

End If

End Sub
Sub APRUEBACLave()
'Accepta o rechaza la entrada de un
usuario

If TXTCUENTA.Text = "" Then
MsgBox "NO SE HA
INTRODUCIDO EL NUMERO DE
CUENTA", 64, "Atención"
TXTCUENTA.SetFocus
Else
DATACLAVE.Recordset.MoveFirst
CLAVE = UCase(TXTPASS.Text)
If Trim(CLAVE) <> "" Then

DATACLAVE.Recordset.FindFirst
"LLAVE=" & CLAVE & ""
If
DATACLAVE.Recordset.NoMatch =
False Then

Call ChecaCla
Else
MsgBox "USUARIO NO
REGISTRADO", 64, "Atención"
CBID.SetFocus
End If

Else
MsgBox "NO SE HA
INTRODUCIDO LA CLAVE", 64,
"Atención"
TXTPASS.SetFocus

End If
End If
End Sub

```

```

Sub CHECACLAVE(Index As Integer)
'Permite acceder la clave, en caso de
error o no registro
'sale del sistema

Dim CLAVE As String

VECECLAVE = VECECLAVE + 1
If VECECLAVE <= 3 Then
Select Case Index
Case 0
Call APRUEBACLave
Case 1
End
End Select
Else
End
End If

End Sub

Private Sub APB1_Click(Index As
Integer)
'Si el boton de aceptar es presionado
checa la clave
Me.MousePointer = 11
Call CHECACLAVE(Index)
End Sub

Private Sub APB1_KeyPress(Index As
Integer, KeyAscii As Integer)
'Si es presionado la tecla de enter en el
boton de aceptar chequea la clave
If KeyAscii = 13 Then
Call CHECACLAVE(Index)
End If
End Sub

Private Sub CBID_Click()
'Diferencia entre el tipo de usuario
alumno y los demás

If CBID.Text = "ESTUDIANTE" Then
sspcuenta.Caption = "&Número de
cuenta"
TXTCUENTA.Text = ""
Else
sspcuenta.Caption = "&RFC"
TXTCUENTA.Text = ""
End If

End Sub

Private Sub CBID_KeyPress(KeyAscii
As Integer)
'Cambia de cuadro a cuadro de texto
If KeyAscii = 13 Then
If sspcuenta.Visible = True Then
TXTCUENTA.SetFocus
Else
TXTPASS.SetFocus
End If
End If

End Sub

Private Sub Form_Activate()
'Activa número de cuenta como texto
activo.

TXTCUENTA.SetFocus

```

<pre> End Sub Private Sub Form_Load() 'Agrega los tipos de usuarios disponible en SIVEA e inicializa forma On Error Resume Next OLEM.CreateEmbed UNIDAD + "sivea\SISTEMA\UTILES\SONIDOS\acc eso.wav" OLEM.Refresh OLEM.DoVerb (0) On Error Resume Next CBID.AddItem "FACILITADOR" CBID.AddItem "DIRECTOR" CBID.AddItem "ESTUDIANTE" VECESCLAVE = 0 'If MusicaON = 1 Then OLEm.CreateEmbed UNIDAD + "sivea\SISTEMA\UTILES\SONIDOS\AC CESO.wav" FRMACCESO.OLEm ENTRACAN = 0 End Sub Private Sub TXTCUENTA_KeyPress(KeyAscii As Integer) 'Cambia de cuadro a cuadro de texto If KeyAscii = 13 Then TXTPASS.SetFocus End If End Sub Private Sub TXTPASS_KeyPress(KeyAscii As Integer) 'Cambia de cuadro a cuadro de texto If KeyAscii = 13 Then Me.MousePointer = 11 Call CHECACLAVE(0) End If End Sub </pre>	<pre> KEY_NOTIFY + KEY_CREATE_LINK + READ_CONTROL ' Tipos principales de claves del Registro Const HKEY_LOCAL_MACHINE = &H00000002 Const ERROR_SUCCESS = 0 Const REG_SZ = 1 Cadena Unicode terminada en Null Const REG_DWORD = 4 Número de 32 bits Const gREGKEYSYSINFOLOC = "SOFTWARE\Microsoft\Shared Tools Location" Const gREGVALSYSINFOLOC = "MSINFO" Const gP_TGKEYSYSINFO = "SOFTWARE\Microsoft\Shared Tools\MSINFO" Const gREGVALSYSINFO = "PATH" Private Declare Function RegOpenKeyEx Lib "advapi32" Alias "RegOpenKeyExA" (ByVal hKey As Long, ByVal lpSubKey As String, ByVal ulOptions As Long, ByVal samDesired As Long, ByRef phkResult As Long) As Long Private Declare Function RegQueryValueEx Lib "advapi32" Alias "RegQueryValueExA" (ByVal hKey As Long, ByVal lpValueName As String, ByVal lpReserved As Long, ByRef lpType As Long, ByVal lpData As String, ByRef lpcbData As Long) As Long Private Declare Function RegCloseKey Lib "advapi32" (ByVal hKey As Long) As Long Private Sub cmdSysInfo_Click() Call StartSysInfo End Sub Private Sub cmdOK_Click() Unload Me End Sub </pre>	<pre> ' Prueba a obtener del Registro la información del sistema sobre el nombre y la ruta del programa. If GetKeyValue(HKEY_LOCAL_MACHINE , gREGKEYSYSINFO, SysInfoPath) Then ' Prueba a obtener del Registro la información del sistema sobre la ruta del programa.. Elseif GetKeyValue(HKEY_LOCAL_MACHINE , gREGKEYSYSINFOLOC, SysInfoPath) Then ' Comprueba la existencia de una versión conocida de un archivo de 32 bits If (Dir(SysInfoPath & "MSINFO32.EXE") <> "") Then SysInfoPath = SysInfoPath & "MSINFO32.EXE" ' Error - Imposible encontrar el archivo Else GoTo SysInfoErr End If ' Error - Imposible encontrar la entrada de Registro Else GoTo SysInfoErr End If Call Shell(SysInfoPath, vbNormalFocus) Exit Sub SysInfoErr. MsgBox "System Information is Unavailable At This Time", vbOKOnly End Sub Public Function GetKeyValue(KeyRoot As Long, KeyName As String, SubKeyRef As String, ByRef KeyVal As String) As Boolean Dim i As Long ' Contador de bucle Dim rc As Long ' Código de retorno Dim hKey As Long ' Controlador a una clave de Registro abierta Dim hDepth As Long Dim KeyValType As Long ' Tipo de dato de una clave de Registro Dim tmpVal As String ' Almacén temporal de una valor de clave de Registro Dim KeyValSize As Long ' Tamaño de la variable de la clave de Registro </pre>
<p>FORMA: FRMACERCA</p> <pre> Option Explicit ' Opciones de seguridad de claves del Registro.. Const READ_CONTROL = &H20000 Const KEY_QUERY_VALUE = &H1 Const KEY_SET_VALUE = &H2 Const KEY_CREATE_SUB_KEY = &H4 Const KEY_ENUMERATE_SUB_KEYS = &H8 Const KEY_NOTIFY = &H10 Const KEY_CREATE_LINK = &H20 Const KEY_ALL_ACCESS = KEY_QUERY_VALUE + KEY_SET_VALUE + _ KEY_CREATE_SUB_KEY + KEY_ENUMERATE_SUB_KEYS + _ </pre>	<pre> Private Sub Form_Load() Carga ventana de acerca de Me.Caption = "Acerca de " & App.Title lblVersion.Caption = "Versión " & App.Major & "." & App.Minor & " " & App.Revision If MusicaON = 1 Then Call TocaMus(1) End If End Sub Public Sub StartSysInfo() On Error GoTo SysInfoErr Dim rc As Long Dim SysInfoPath As String </pre>	<pre> ' Abre la clave de Registro en la raíz {HKEY_LOCAL_MACHINE } </pre>

```
rc = RegOpenKeyEx(KeyRoot,
KeyName, 0, KEY_ALL_ACCESS,
hKey) ' Abre la clave de Registro

If (rc <> ERROR_SUCCESS) Then
GoTo GetKeyError ' Trata el
error.

tmpVal = String$(1024, 0)
' Asigna espacio para la variable
KeyValSize = 1024
' Marca el tamaño de la variable
```

* Recupera valores de claves de Registro

```
rc = RegQueryValueEx(hKey,
SubKeyRef, 0, _
KeyValType, tmpVal,
KeyValSize) ' Obtiene o crea un valor de clave
```

```
If (rc <> ERROR_SUCCESS) Then
GoTo GetKeyError ' Trata el error
```

```
If (Asc(Mid(tmpVal, KeyValSize, 1)) =
0) Then ' Win95 agrega una
cadena terminada en Null...
```

```
tmpVal = Left(tmpVal, KeyValSize -
1) ' Se encontró Null, se extrae
de la cadena
Else
WinNT no tiene una cadena terminada
en Null.
```

```
tmpVal = Left(tmpVal, KeyValSize)
' No se encontró Null, sólo se extrae la
cadena
End If
```

' Determina el tipo de valor de la
clave para conversión...

```
Select Case KeyValType
' Busca tipos de datos...
Case REG_SZ
' Tipo de dato de la cadena de la clave
de Registro
KeyVal = ImpVal
' Copia el valor de la cadena
Case REG_DWORD
' El tipo de dato de la cadena de la clave
es Double Word
For i = Len(tmpVal) To 1 Step -1
' Convierte cada byte
KeyVal = KeyVal +
Hex(Asc(Mid(tmpVal, i, 1))) ' Genera el
valor carácter a carácter
Next
KeyVal = Format$("&h" + KeyVal)
' Convierte Double Word a String
End Select
```

```
GetKeyValue = True
' Vuelve con éxito
rc = RegCloseKey(hKey)
' Cierra la clave de Registro
Exit Function
' Salir
```

```
GetKeyError. ' Restaurar después de
que ocurra un error .
KeyVal = ""
' Establece el valor de retorno para una
cadena vacía
GetKeyValue = False
' Devuelve un error
rc = RegCloseKey(hKey)
' Cierra la clave de Registro
End Function
```

FORMA: FRMAGENDA

```
Private Sub Calendar1_DblClick()
fecha1 = Calendar1.Value
If frm1.Value = 0 Then
On Error Resume Next
frmTEMAS.txtFields(3) = fecha1
Else
On Error Resume Next
frmTEMAS.txtFields(3) = fecha1
End If
```

```
On Error Resume Next
If frmTEMAS.txtFields(3) >
frmTEMAS.txtFields(4) Then
frmTEMAS.txtFields(4) =
frmTEMAS.txtFields(3)
End If
```

End Sub

```
Private Sub Form_Load()
Calendar1.Value = Date
End Sub
```

FORMA: FRMALUMNOS

```
Private Sub cmdAgrega_Click()
datPrimaryRS.Recordset.MoveLast
grdDataGrid.SetFocus
SendKeys "{down}"
End Sub
```

```
Private Sub cmdEliminar_Click()
With datPrimaryRS.Recordset
On Error Resume Next
.Delete
.MoveNext
If .EOF Then MoveLast
End With
End Sub
```

```
Private Sub cmdRenovar_Click()
'Esto sólo es necesario en aplicaciones
con múltiples usuarios
datPrimaryRS.Refresh
End Sub
```

```
Private Sub cmdActualizar_Click()
datPrimaryRS.UpdateRecord
datPrimaryRS.Recordset.Bookmark =
datPrimaryRS.Recordset.LastModified
End Sub
```

```
Private Sub cmdCerrar_Click()
Screen.MousePointer = vbDefault
Unload Me
End Sub
```

```
Private Sub
datPrimaryRS_Error(DataErr As Integer,
Response As Integer)
' Aquí es dónde puede situar el código
de tratamiento de error
' Si desea ignorar los errores, quite el
comentario de la siguiente línea
' Si desea capturarlos, agregue código
aquí para controlarlos
MsgBox "Data error event hit err" &
Error$(DataErr)
Response = 0 ' Desprecia el error
End Sub
```

```
Private Sub datPrimaryRS_Reposition()
Screen.MousePointer = vbDefault
On Error Resume Next
' Mostrar la posición actual de registro
para dynasets y snapshots
datPrimaryRS.Caption = "Record " &
(datPrimaryRS.Recordset.AbsolutePosi
tion + 1)
End Sub
```

```
Private Sub
datPrimaryRS_Validate(Action As
Integer, Save As Integer)
' Aquí se sitúa el código de validación
' Este evento se invoca cuando ocurre
la siguiente acción
```

```
Select Case Action
Case vbDataActionMoveFirst
Case vbDataActionMovePrevious
Case vbDataActionMoveNext
Case vbDataActionMoveLast
Case vbDataActionAddNew
Case vbDataActionUpdate
Case vbDataActionDelete
Case vbDataActionFind
Case vbDataActionBookmark
Case vbDataActionClose
Screen.MousePointer = vbDefault
End Select
Screen.MousePointer = vbHourglass
End Sub
```

```
Private Sub Form_Unload(Cancel As
Integer)
Screen.MousePointer = vbDefault
End Sub
```

```
Private Sub Form_Resize()
On Error Resume Next
' Esto redimensionará la cuadrícula al
cambiar el tamaño del formulario
grdDataGrid.Height = Me.ScaleHeight -
datPrimaryRS.Height -
picButtons.Height - 30
End Sub
```

FORMA FRMAULAS

```
Dim PASAME As Integer
Sub PONAULAS(dato As Integer)
Dim lMX As Object
```

```
SIVEA.ActiveForm.MousePointer = 11
ventana = 100
MateSele = dato + 1
SIVEA.DATACAPI.Refresh
```

```
SIVEA.DATACAPI.Recordset.MoveFirst
If dato = 0 Then
```

```

frmmenu.ssbots.Tab = 4
z = 0
Else
frmmenu.ssbots.Tab = 6
cual = "ESTATICA"

SIVEA.DATACAPI.Recordset.FindFirst
"CAPITULO=" & cual & ""
z = 7
End If

For i = 0 To 6
texto =
SIVEA.DATACAPI.Recordset.CAPITULO.Value
Image1(i).Picture =
FRMFOTOSA.imatemp(z).Picture
SALON(i).Caption = texto
If
SIVEA.DATACAPI.Recordset.EOF Then
Exit For
Else

SIVEA.DATACAPI.Recordset.MoveNext
End If
z = z + 1
Next i

SalonZoom(0).Visible = True
SIVEA.ActiveForm.MousePointer = 0

End Sub
Sub DondeEstas(i As Integer)
ssescalas(0).Enabled = True
ssescalas(1).Enabled = True
Select Case i
Case 0
ssescalas(0).Enabled = False
Case 1
ssescalas(1).Enabled = False
End Select

End Sub
Sub llenar(cual As String)
SIVEA.DATACAPI.Refresh

SIVEA.DATACAPI.Recordset.MoveFirst

SIVEA.DATACAPI.Recordset.FindFirst
"CAPITULO=" & cual & ""

If
SIVEA.DATACAPI.Recordset.NoMatch = False Then
MESA =
SIVEA.DATACAPI.Recordset.DESCRIP.Value
On Error Resume Next
Label1.Caption = MESA
End If
End Sub

Private Sub Command1_Click(Index As Integer)
SIVEA.ActiveForm.MousePointer = 11
Select Case Index
Case 0
sspanelex.Visible = False
Case 1
Call PasaAulas(1, PASAME)
End Select
SIVEA.ActiveForm.MousePointer = 0
End Sub

Private Sub Form_Activate()
If MusicaON = 1 Then
Call TocaMus(2)
Else
SIVEA.OLEP.DoVerb (0)
End If

frmmenu.ssbots.Tab = 3
End Sub

Private Sub Form_Load()
If SIVEA.mnubarraH.Checked = True Then
frmmenu.Show
End If
End Sub

Private Sub imaedificio_MouseMove(Index As Integer, Button As Integer, Shift As Integer, X As Single, Y As Single)
PICEDIFICIO(0).Visible = False
PICEDIFICIO(1).Visible = False
PICEDIFICIO(2).Visible = False

If Y > 2625 And Y < 5600 Then
If X > 1365 And X < 3950 Then
If PICEDIFICIO(1).Visible = False Then
PICEDIFICIO(1).Visible = True
End If
End If
End If

If Y > 1725 And Y < 5139 Then
If X > 6675 And X < 10900 Then
If PICEDIFICIO(0).Visible = False Then
PICEDIFICIO(0).Visible = True
End If
End If
End If

If Y > 5138 And Y < 5370 Then
If X > 6279 And X < 7950 Then
If PICEDIFICIO(2).Visible = False Then
PICEDIFICIO(2).Visible = True
End If
End If
End If
End Sub

Private Sub Image1_MouseDown(Index As Integer, Button As Integer, Shift As Integer, X As Single, Y As Single)
Call PasaAulas(Button, Index)
End Sub

Private Sub Image4_Click()
Call frmmenu.Regreso
End Sub

Private Sub Label2_Click(Index As Integer)
SalonZoom(0).Visible = True
ventana = 100
End Sub

End Sub

Private Sub PICEDIFICIO_Click(Index As Integer)
Call PONAULAS(Index)
End Sub

Private Sub salon_Click(Index As Integer)
Call PasaAulas(1, Index)
End Sub
Sub PasaAulas(cual As Integer, inde As Integer)
If cual = 2 Then
On Error Resume Next
sspanelex.Caption =
SALON(inde).Caption
imamuestra.Picture =
Image1(inde).Picture
sspanelex.Visible = True
Call llenar(sspanelex.Caption)
PASAME = inde
Else
On Error Resume Next
OLEM.Action = 9
FRMAULAS.MousePointer = 11
TemaSele = inde + 1
frmsalon.Show
frmsalon.rotafolio.Picture =
Image1(inde).Picture
frmsalon.Caption =
SALON(inde).Caption
FRMAULAS.MousePointer = 0
Unload FRMAULAS
End If
End Sub

Private Sub sscalas_Click(Index As Integer)
Select Case Index
Case 0
If edificioM.Top = -105 Then
edificioM.Top = 5
Call DondeEstas(Index)
Else
edificioM.Top = -105
End If
Case 1
If edificioM.Top = -105 Then
edificioM.Top = -210
Call DondeEstas(Index)
Else
edificioM.Top = -105
End If
End Select
vscalesa.Value = edificioM.Top

End Sub

Private Sub vscalesa_Change()
edificioM.Top = vscalesa.Value
If vscalesa.Value <= -180 Then
 piso.Visible = True
Else
 piso.Visible = False
End If
If vscalesa.Value > -85 Then
Call DondeEstas(0)
Else
If vscalesa.Value < -125 Then
Call DondeEstas(1)
Else

```

<pre> Call DondeEstas(2) End If End If End Sub </pre>	<pre> Sub LLENALISTA(AREA As Integer, tipD As String) Call cualLlena sspan1.Caption = "Consultando [" & SSTab1.Caption & "]" ' Crea una variable para agregar objetos ListItem Dim itmX As ListItem ' Mientras el registro no sea el ultimo, agrega un objeto ListItem. ' Establece la Database como la base de datos BIBLIO.MDB Set myDb = DBEngine.Workspaces(0).OpenDatabas e("C:\sive\vnstitucion\gobierno\registros \Biblioteca.mdb") ' Establece el conjunto de registros como la tabla Set myRs = myDb.OpenRecordset("general" dbOpenDynaset) myRs.MoveFirst temp = "" While Not myRs.EOF If myRs.ESTANTE = AREA Then If tipD = "TD" Then temp = "TD" tipD = myRs.tipodoc End If If myRs.tipodoc = tipD Then Set itmX = ListView1.ListItems.Add(, , CStr(myRs.nombre) & " " & NOMBRE DEL CURSO Select Case myRs.tipodoc Case "WORD" itmX.Icon = 1 itmX.SmallIcon = 1 ' Establece un icono desde ImageList2 mx = "DOCUMENTO DE TEXTO" Case "EXL" itmX.Icon = 2 itmX.SmallIcon = 2 mx = "HOJA DE CALCULO" Case "PPP" itmX.Icon = 3 itmX.SmallIcon = 3 mx = "PRESENTACION" End Select ' Si el campo ID CURSO no es nulo, establece SubItem , como el If Not IsNull(myRs.AUTOR) Then On Error Resume Next </pre>	<pre> itmX.SubItems(1) = CStr(myRs.AUTOR) ' AUTOR de LIBRO End If If Not IsNull(myRs.CLAVE) Then itmX.SubItems(2) = CStr(myRs.CLAVE) ' CLAVE de LIBRO End If If Not IsNull(myRs.tipodoc) Then itmX.SubItems(3) = CStr(myRs.tipodoc) ' TIPO de LIBRO End If ' If mx <> "" Then ' itmX.SubItems(1) = mx ' ID de CURSO End If End If End If If temp = "TD" Then tipD = "TD" End If myRs.MoveNext ' Pasa al siguiente registro Wend On Error Resume Next ListView1.SortKey = 2 ListView1.View = 0 End Sub Private Sub MNUABRE_Click() sspan1.Visible = True End Sub Private Sub mnuFin_Click() Call frmMenu.Regres Unload Me End Sub Private Sub mnumante_Click() frmmanio.Show End Sub Private Sub MNUMATE_Click() frmconsulta.Show End Sub Private Sub mnutable_Click() frmconsulta.Show End Sub Private Sub Form_Activate() If MusicaON = 1 Then Call TocaMus(3) Else SIVEA.OLEP.DoVerb(0) End If End Sub Private Sub Form_Load() sspan1.Width = 10 TipD = "TD" </pre>
<pre> FORMA FRMBIBLIO. Dim TipD As String Sub ponpiza(TEMASEL As String) 'DIRE = "C:\SIVE\VNSTITUCION\AULAS\ nombre = TEMASEL Data1.Recordset.FindFirst "NOMBRE =" & nombre & "" If Data1.Recordset.NoMatch = False Then archi = Data1.Recordset.LOCALIZA Err.Clear OLEDIR = archi On Error Resume Next OLE1.Refresh OLE1.CreateEmbed OLEDIR If (Err) Then If Err = 30015 Then MsgBox "No es un objeto válido." Else MsgBox Error\$ End If End If VSB1.Min = OLEM.Top HSB1.Min = OLEM.Left VSB1.Max = OLEM.Top + OLEM.Height HSB1.Max = OLEM.Left + OLEM.Width End If End Sub Sub cualLlena() Dim itmX As ColumnHeader ' Agrega ColumnHeaders El ancho de las columnas es el ancho ' del control dividido por el número de objetos ColumnHeader. ListView1.ListItems.Clear ' ListView1.ColumnHeaders.Clear Set itmX = ListView1.ColumnHeaders.Add(, , "NOMBRE", ListView1.Width / 4) Set itmX = ListView1.ColumnHeaders.Add(, , "AUTOR", ListView1.Width / 4) Set itmX = ListView1.ColumnHeaders.Add(, , "CLAVE", ListView1.Width / 4) Set itmX = ListView1.ColumnHeaders.Add(, , "TIPO DE ARCHIVO", ListView1.Width / 3) ListView1.ListItems.Clear ' ListView1.ColumnHeaders.Clear Set itmX = ListView1.ColumnHeaders.Add(, , "NOMBRE", ListView1.Width / 4) Set itmX = ListView1.ColumnHeaders.Add(, , "AUTOR", ListView1.Width / 4) Set itmX = ListView1.ColumnHeaders.Add(, , "CLAVE", ListView1.Width / 4) Set itmX = ListView1.ColumnHeaders.Add(, , "TIPO DE ARCHIVO", ListView1.Width / 3) ListView1.ListItems.Clear ' ListView1.ColumnHeaders.Clear Set itmX = ListView1.ColumnHeaders.Add(, , "NOMBRE", ListView1.Width / 4) Set itmX = ListView1.ColumnHeaders.Add(, , "AUTOR", ListView1.Width / 4) Set itmX = ListView1.ColumnHeaders.Add(, , "CLAVE", ListView1.Width / 4) Set itmX = ListView1.ColumnHeaders.Add(, , "TIPO DE ARCHIVO", ListView1.Width / 3) ListView1.ListItems.Clear ' ListView1.ColumnHeaders.Clear Set itmX = ListView1.ColumnHeaders.Add(, , "NOMBRE", ListView1.Width / 4) Set itmX = ListView1.ColumnHeaders.Add(, , "AUTOR", ListView1.Width / 4) Set itmX = ListView1.ColumnHeaders.Add(, , "CLAVE", ListView1.Width / 4) Set itmX = ListView1.ColumnHeaders.Add(, , "TIPO DE ARCHIVO", ListView1.Width / 3) End Sub </pre>		

<pre> End Sub Private Sub HSB1_Change() OLEM.Left = -(HSB1.Value) End Sub Private Sub ListView1_Click() On Error Resume Next Call ponpiza(ListView1.SelectedItem) End Sub Private Sub ListView1_ColumnClick(ByVal ColumnHeader As ComctlLib.ColumnHeader) 'se orden por los subelementos de esa columna. 'Establece el SortKey como el Index del ColumnHeader - 1 ListView1.SortKey = ColumnHeader.Index - 1 'Establece Sorted a True para ordenar la lista ListView1.Sorted = True End Sub Private Sub OLE1_DbClick() On Error Resume Next OLEM.DoVerb (-1) End Sub Private Sub SCommand1_Click(Index As Integer) If Index = 0 Then If sspan1.Width <= 30 Then SCommand1(0).Picture = picb(0).Picture sspan1.Width = 9700 Else OLEM.DoVerb (-1) End If Else If sspan1.Width <= 30 Then Call frmMenu.Regreso Unload Me Else SCommand1(0).Picture = picb(1).Picture sspan1.Width = 30 End If End If End Sub Private Sub STab1_Click(PreviousTab As Integer) Call LLENALISTA(SSTab1.Tab, TiD) End Sub Private Sub Toolbar1_ButtonClick(ByVal Button As ComctlLib.Button) ' Cambia ListView con el toolbar Select Case Button.Key Case "icono" X = 0 On Error Resume Next ListView1.SortKey = 0 Case "icono1" X = 1 ListView1.SortKey = 1 </pre>	<pre> Case "lista" X = 3 ListView1.SortKey = 0 Case "wd" TiD = "WORD" Case "ex" TiD = "EXCEL" Case "ppp" TiD = "PPP" Case "ll" TiD = "TD" Case "reporte" X = 3 ListView1.SortKey = 0 End Select ListView1.View = X. End Sub Private Sub VSB1_Change() OLEM.Top = -(VSB1.Value) End Sub </pre>	<pre> Else ' coneava Value = 0 'End If Call LibrenaOle(0) End If End Sub Sub listavista() ' Crea una variable de objeto para el objeto ColumnHeader Dim i As Integer Dim X As Integer Dim cimX As ColumnHeader ' Agrega ColumnHeaders. El ancho de las columnas es el ancho del control dividido por el número de objetos ColumnHeader. Set cimX = ListView1.ColumnHeaders.Add(, "AREA", 3000) Set cimX = ListView1.ColumnHeaders.Add(, "MATERIA", 2000) Set cimX = ListView1.ColumnHeaders.Add(, "TEMA", 1500) Set cimX = ListView1.ColumnHeaders.Add(, "DIFICULTAD", 500) ListView1.Icons = ImageList1 ListView1.SmallIcons = ImageList2 ' Crea variables de objeto para objetos de Data Access Dim myDb As Database, myRs As Recordset ' Establece la Database como la base de datos BIBLIOMDE Set myDb = DBEngine.Workspaces(0).OpenDatabase(dIRECTORIOBASES) ' Establece el conjunto de registros como la tabla "Autores". Set myRs = myDb.OpenRecordset("CLASES", dbOpenDynaset) ' Crea una variable para agregar objetos ListItem. Dim itmX As ListItem myRs.MoveFirst ' Mientras el registro no sea el último, agrega un objeto ListItem While Not myRs.EOF Set itmX = ListView1.ListItems.Add(, CStr(myRs!SubTema), 1) ' NOMBRE DEL TEMA Select Case myRs!materia Case 1 Select Case myRs!CAPITULO Case 1 itmX.Icon = 1 </pre>
FORMA:FRMCLASES		
	<pre> Dim ACCIO As String Sub BuscaLista() On Error Resume Next TXTSUBTEMA.Text = ListView1.SelectedItem.nombre = TXTSUBTEMA.Text DataClases.Recordset.MoveFirst DataClases.Recordset.FindFirst "SUBTEMA=" & nombre & "" If DataClases.Recordset.NoMatch = False Then AREA = DataClases.Recordset.materia Combo1.ListIndex = AREA - 1 Tema = DataClases.Recordset.CAPITULO If AREA = 2 Then Tema = Tema + 7 End If Combo2.ListIndex = Tema - 1 DIFI = DataClases.Recordset.DIFICULTAD Combo3.ListIndex = DIFI - 1 If DataClases.Recordset.clase = 1 Then tipoa.Buttons("key01").Value = tbrPressed Else If DataClases.Recordset.clase = 4 Then tipoa.Buttons("KEY02").Value = tbrPressed End If End If ' cone = DataClases.Recordset.coneava If cone = True Then On Error Resume Next labele.Caption = DataClases.Recordset.LOCAEava coneava.Value = 1 </pre>	

<pre> Case 2 itmX.Icon = 2 Case 3 itmX.Icon = 3 Case 4 itmX.Icon = 4 Case 5 itmX.Icon = 5 Case 6 itmX.Icon = 6 Case 7 itmX.Icon = 7 End Select itmX.SmallIcon = 1 'Establece un icono desde Imagelist2 Case 2 Select Case myRs!CAPITULO Case 1 itmX.Icon = 8 Case 2 itmX.Icon = 9 Case 3 itmX.Icon = 10 Case 4 itmX.Icon = 11 Case 5 itmX.Icon = 12 Case 6 itmX.Icon = 13 Case 7 itmX.Icon = 14 End Select itmX.SmallIcon = 2 End Select ' Si el campo MATERIA no es nulo, establece SubItem 1 como él If Not IsNull(myRs!materia) Then Select Case myRs!materia Case 1 itmX.SubItems(1) = "MATEMATICAS" + " MATERIA Case 2 itmX.SubItems(1) = "FISICA" + MATERIA End Select End If ' Si el CAPITULO no es Null, establece SubItem 2 como él. If Not IsNull(myRs!CAPITULO) Then itmX.SubItems(2) = myRs!CAPITULO End If If Not IsNull(myRs!DIFICULTAD) Then Select Case myRs!DIFICULTAD Case 1 itmX.SubItems(3) = "BASICO" Case 2 itmX.SubItems(3) = "MEDIO" Case 3 itmX.SubItems(3) = "AVANZADO" End Select </pre>	<pre> End If myRs.MoveNext ' Pasa al siguiente registro Wend ListView1.SortKey = 2 ListView1.View = 0 End Sub Private Sub Combo1_Click() If Combo1.ListIndex = 1 Then Cor = 02.ListIndex = 7 Else Combo2.ListIndex = 0 End If End Sub Private Sub Combo4_Click() Select Case Combo4.Text Case "AREA DE CONOCIMIENTO" ListView1.SortKey = 1 ListView1.View = 1 Case "MATERIA" ListView1.SortKey = 3 ListView1.View = 3 Case "TODOS" ListView1.SortKey = 2 ListView1.View = 0 End Select End Sub Private Sub Form_Load() ventana = 400 Me.MousePointer = 11 Combo1.AddItem "MATEMATICAS" Combo1.AddItem "FISICA" If MusicaON = 1 Then Call TocaMus(10) Else SIVEA.OLEP.DoVerb (0) End If DATACAPI.Refresh DATACAPI.Recordset.MoveLast X = DATACAPI.Recordset.RecordCount DATACAPI.Recordset.MoveFirst For i = 1 To X Combo2.AddItem DATACAPI.Recordset.CAPITULO.Value If Not DATACAPI.Recordset.EOF Then DATACAPI.Recordset.MoveNext Enc Next i Combo3.AddItem "BASICO" Combo3.AddItem "MEDIO" Combo3.AddItem "AVANZADO" Combo4.AddItem "AREA DE CONOCIMIENTO" Combo4.AddItem "MATERIA" Combo4.AddItem "TODOS" </pre>	<pre> Call listavista ACCIO = "" Me.MousePointer = 0 End Sub Private Sub Form_Unload(Cancel As Integer) If MusicaON = 1 Then On Error Resume Next SIVEA.ActiveForm.OLEMAction = 9 End If Call frmfacil.muestra(True) End Sub Private Sub ListView1_Click() Call BuscaLista End Sub Sub LibreriaOle(Acc As Integer) Dim Filenum As Variant Dim DIRE. OLEDIR As String Screen.MousePointer = 11 If Acc = 0 Then DIRE = "C:\SIVEA\INSTITUCIONVAULAS\ archi = DataClasses.Recordset.LOCARCHIVO Else If Acc = -1 Then DIRE = "C:\SIVEA\INSTITUCIONVAULAS\ Mensaje = "Escriba el nombre de la clase " ' Establece el mensaje Mensaje = Mensaje + Chr(13) + "indicando el directorio de fisica o matematicas" Mensaje = Mensaje + Chr(13) + "esta debe estar dentro del directorio de " Mensaje = Mensaje + Chr(13) + "C:\SIVEA\VAULAS\ Mensaje = Mensaje + Chr(13) + "EJEMPLO" Mensaje = Mensaje + Chr(13) + "ESCRIBIR : MATEMATICAS\NOMBRE DEL ARCHIVO]" Mensaje = Mensaje + Chr(13) + "O FISICA\NOMBRE DEL ARCHIVO]" Mensaje = Mensaje + Chr(13) + "NO AGREGUE LA EXTENSION DEL ARCHIVO" Titulo = "INGRESAR DIRECCION DE LA PRESENTACION DE LA CLASE" ' Establece el titulo Predeterminado = "MATEMATICAS" ' Establece el valor predeterminado. ' Muestra el mensaje, el titulo y el valor predeterminado MIVALOR = InputBox(Mensaje, Titulo, Predeterminado) If MIVALOR <> "" Then </pre>
---	---	---

<pre> archi = Trim(MIVALOR) + .PPT" End If Else DIRE = "C:\SIVE\DOCUMENTOS\PROFESOR V" archi = "formato.ppt" End If End If OLEDIR = DIRE + archi OLE1(0).OLETypeAllowed = 1 OLE1(0).SourceItem = OLEDIR OLE1(0).SourceDoc = OLEDIR OLE1(0).Class = "PowerPoint.Show" On Error Resume Next OLE1(0).CreateLink OLEDIR If (Err) Then If Err = 30015 Then MsgBox "No es un objeto válido " Else MsgBox Error\$ End If End If ' Restablece el puntero del ratón Screen.MousePointer = 0 End Sub Private Sub OLE1_MouseDown(Index As Integer, Button As Integer, Shift As Integer, X As Single, Y As Single) Select Case Button Case 1 On Error Resume Next OLE1(0).DoVerb -2 Case 2 Call LibreaOle(-1) End Select End Sub Private Sub SSbot1_Click(Index As Integer) TXTSUBTEMA.Enabled = True For i = 0 To 2 SSbot1(i).Enabled = False Next i Select Case Index Case 0 Combo1.Enabled = True Combo2.Enabled = True Combo3.Enabled = True SSbot1(3).Enabled = True SSbot1(4).Enabled = True OLE1(0).Enabled = True Call LibreriaOle(1) TXTSUBTEMA.Enabled = True TXTSUBTEMA.Text = "" TXTSUBTEMA.SetFocus </pre>	<pre> Label1(4).Caption = "Haga clic en el cuadro para editar" ACCIO = "C" tipoa.Enabled = True Case 1 noforma = "C:\SIVE\DOCUMENTOS\PROFESOR Vformato.ppt" If noforma = OLE1(0).SourceDoc Then MsgBox ("No se debe borrar el formato") Else T = "Esta por eliminar un clase" + Chr(13) T = T + "¿Confirmar operación?" + Chr(13) res = MsgBox(T, 33, "Eliminar una clase") If res = 1 Then archi = "C:\SIVE\INSTITUCIONAULAS" archi = archi + (DataClasses.Recordset.LOCARCHIVO. Value) T = "¿Eliminar también el archivo en disco?" res = MsgBox(T, 33, "Eliminar una clase") If res = 1 Then Kill archi End If DataClasses.Recordset.Delete ListView1.ColumnHeaders.Clear ListView1.ListItems.Clear Call listavista Call BuscaLista End If End If SSbot1(4).Enabled = False Index = 4 Case 2 Combo1.Enabled = True Combo2.Enabled = True Combo3.Enabled = True SSbot1(3).Enabled = True SSbot1(4).Enabled = True Label1(4).Caption = "Haga clic en el cuadro para editar" OLE1(0).Enabled = True tipoa.Enabled = True ACCIO = "E" Case 3 If TXTSUBTEMA.Text = "" Then MsgBox ("la clase debe tener un nombre ") Else AREA = Combo1.ListIndex + 1 Tema = Combo2.ListIndex + 1 If AREA = 2 Then Tema = Tema - 7 End If DIFI = Combo3.ListIndex + 1 If ACCIO = "C" Then NUME = 1 </pre>	<pre> DataClasses.Recordset.MoveFirst FIN = False Do If NUME < 10 Then NUMES = "0" + Trim(Str(NUME)) Else NUMES = Str(NUME) End If CLAVE = Trim("0" + Trim(Str(AREA)) + "0" + Trim(Str(Tema)) + Trim(NUMES)) DataClasses.Recordset.FindFirst "CLAVE = " & CLAVE & "" If DataClasses.Recordset.NoMatch = True Then FIN = True Else NUME = NUME + 1 End If Loop Until FIN = True DataClasses.Recordset.MoveLast DataClasses.Recordset.AddNew Else DataClasses.Recordset.Edit CLAVE = DataClasses.Recordset.CLAVE End If 'On Error Resume Next OLEDIR = Mid(OLE1(0).SourceDoc, 27, (Len(OLE1(0).SourceDoc) - 26)) If OLEDIR = "ORVformato.ppt" Then OLEDIR = "" End If If tipoa.Buttons("key01").Value = tbrPressed Then DataClasses.Recordset.clase = 1 Else If tipoa.Buttons("KEY02").Value = tbrPressed Then DataClasses.Recordset.clase = 4 End If End If DataClasses.Recordset.SubTema = Trim(TXTSUBTEMA.Text) DataClasses.Recordset.materia = AREA DataClasses.Recordset.CAPITULO = Tema DataClasses.Recordset.DIFICULTAD = DIFI </pre>
---	---	---

```
DataClases.Recordset.LOCARCHIVO =
OLEDIR

DataClases.Recordset.LOCAeva = "cc"

DataClases.Recordset.CLAVE = CLAVE

DataClases.Recordset.Update
If Err.Number <> 0 Then
    Msg = "Error #" &
Str(Err.Number) & " se generó por " _
& Err.Source & Chr(13) &
Err.Description
MsgBox Msg, "Error",
Err.HelpFile, Err.HelpContext
End If

ListView1.ColumnHeaders.Clear
ListView1.ListItems.Clear
Call listavista

SSbot1(3).Enabled = False
SSbot1(4).Enabled = False

End If

Case 4
    SSbot1(3).Enabled = False
Case 5
    Unload Me

End Select

If (Index > 2) And (Index < 5) Then
    Combo1.Enabled = False
    Combo2.Enabled = False
    Combo3.Enabled = False
    For i = 0 To 2
        SSbot1(i).Enabled = True
    Next i
    Label1(4).Caption =
"PRESENTACION"
TXTSUBTEMA.Enabled = False
OLE1(0).Enabled = False
tipoa.Enabled = False
End If

End Sub

Private Sub SSCommand1_Click()
    If Picture1.Width = 3000 Then
        Picture1.Width = 9000
        SSCommand1.Picture =
ImageList3.ListImages(2).Picture
        ListView1.Arrange = 1
    Else
        Picture1.Width = 3000
        SSCommand1.Picture =
ImageList3.ListImages(1).Picture
        ListView1.Arrange = 2
    End If
    ListView1.Width = Picture1.Width -
300
    ListView1.Refresh
End Sub
```

```
FORMA: FRMCONSULTA
Private Sub cmdCerrar_Click()
    Screen.MousePointer = vbDefault
```

```
Unload Me
End Sub

Private Sub
datPrimaryRS_Error(DataErr As Integer,
Response As Integer)
    'Aquí es dónde puede situar el código
de tratamiento de error
    'Si desea ignorar los errores, quite el
comentario de la siguiente línea
    'Si desea capturarlos, agregue código
aquí para controlarlos
    MsgBox "Data error event hit err " &
Error$(DataErr)
    Respónse = 0 'Despreciar el error
End Sub

Private Sub datPrimaryRS_Repositron()
    Screen.MousePointer = vbDefault
    On Error Resume Next
    'Mostrar la posición actual de registro
para dynasets y snapshots
    datPrimaryRS.Caption = "Record " &
(datPrimaryRS.Recordset.AbsolutePosit
ion + 1)
End Sub

Private Sub
datPrimaryRS_Validate(Action As
Integer, Save As Integer)
    'Aquí se sitúa el código de validación
    'Este evento se invoca cuando ocurre
la siguiente acción
    Select Case Action
        Case vbDataActionMoveFirst
        Case vbDataActionMovePrevious
        Case vbDataActionMoveNext
        Case vbDataActionMoveLast
        Case vbDataActionAddNew
        Case vbDataActionUpdate
        Case vbDataActionDelete
        Case vbDataActionFind
        Case vbDataActionBookmark
        Case vbDataActionClose
            Screen.MousePointer = vbDefault
    End Select
    Screen.MousePointer = vbHourglass
End Sub

Private Sub Form_Unload(Cancel As
Integer)
    Screen.MousePointer = vbDefault
End Sub

Private Sub Form_Resize()
    On Error Resume Next
    'Esto redimensionará la cuadrícula al
cambiar el tamaño del formulario
    grdDataGrid.Height = Me.ScaleHeight -
datPrimaryRS.Height -
picButtons.Height - 30
End Sub

FORMA:FRMDATOSF

Private Sub cmdAgregar_Click()
    datPrimaryRS.Recordset.AddNew
    Call Habbitot(True)
    accusu = "ALTA"
End Sub

Private Sub cmdEliminar_Click()
```

```
T = "Esta por eliminar a un facilitador"
+ Chr(13)
T = T + "" + Chr(13)
T = T + "NOTA: El facilitador será
eliminado" + Chr(13)
T = T + "Incluso como usuario " +
Chr(13)
T = T + "¿Confirmar operación?" +
Chr(13)
res = MsgBox(T, 33, "Eliminar una
FACILITADOR")
If res = 1 Then

    X = txtFields(0) Text
    Data1.Refresh

    With Data1 Recordset
        MoveFirst
        .FindFirst "NOCTA="" & X & ""
        z = .NOCTA.Value
        If NoMatch = False Then
            .Delete
            MoveNext
            If EOF Then
                MoveLast
            End If
            .MoveFirst
            For i = 1 To RecordCount
                .num Value = i
                .d.Value =
Trim(Left(.id Value, 5)) & Trim(Str(i))
                Update
                MoveNext
                If EOF Then
                    Exit For
                End If
                Next i
                MoveLast
            End If
        End With
        Data1.Refresh

        With datPrimaryRS.Recordset
            Delete
            .MoveNext
            If EOF Then
                .MoveLast
            End If
        End With
        datPrimaryRS.Refresh
    End If

End Sub

Private Sub cmdRenovar_Click()
    'Esto sólo es necesario en aplicaciones
con múltiples usuarios
    On Error Resume Next
    datPrimaryRS.Refresh
    Call Habbitot(False)
End Sub

Private Sub cmdActualizar_Click()
    If txtFields(0) Text = "" Then
        MsgBox ("Error en la captura de
datos")
    Else
        datPnmaryRS.UpdateRecord
        datPrimaryRS.Recordset.Bookmark =
datPrimaryRS.Recordset.LastModified
```

```

Call HabiBot(False)
IdeN = txtFields(0).Text
Call
ENTRAUSUARIO("FACILITADOR")
End If
End Sub

Private Sub cmdCerrar_Click()
Screen.MousePointer = vbDefault
Unload Me
End Sub

Private Sub Command1_Click()
Call HabiBot(True)
acciusu = "MODI"
End Sub
Sub HabiBot(que As Boolean)

Dim Opu As Boolean

cmdActualizar.Enabled = que
cmdRenovar.Enabled = que
IMAFOTO.Enabled = que

For i = 0 To 14
txtFields(i).Enabled = que
Next i

If que = True Then
Opu = False
Else
Opu = True
End If

cmdAgrega.Enabled = Opu
cmdEliminar.Enabled = Opu
cmdCerrar.Enabled = Opu
Command1.Enabled = Opu
datPrimaryRS.Visible = Opu

End Sub

Private Sub datPrimaryRS_Error(DataErr As Integer,
Response As Integer)
"Aquí es dónde puede situar el código
de tratamiento de error
'Si desea ignorar los errores, quite el
comentario de la siguiente línea
'Si desea capturarlos, agregue código
aquí para controlarlos
MsgBox "Data error event hit err:" &
Error$(DataErr)
Response = 0 'Despreciar el error
End Sub

Private Sub datPrimaryRS_Reposition()
Screen.MousePointer = vbDefault
On Error Resume Next
'Mostrar la posición actual de registro
para dynasets y snapshots
datPrimaryRS.Caption = "Registro N°." &
(datPrimaryRS.Recordset.AbsolutePosi
tion + 1)
End Sub

Private Sub datPrimaryRS_Validate(Action As
Integer, Save As Integer)
'Aquí se sitúa el código de validación
'Este evento se invoca cuando ocurre
la siguiente acción

```

```

Select Case Action
Case vbDataActionMoveFirst
Case vbDataActionMovePrevious
Case vbDataActionMoveNext
Case vbDataActionMoveLast
Case vbDataActionAddNew
Case vbDataActionUpdate
Case vbDataActionDelete
Case vbDataActionFind
Case vbDataActionBookmark
Case vbDataActionClose
Screen.MousePointer = vbDefault
End Select
Screen.MousePointer = vbHourglass
End Sub

Private Sub Form_Load()
HabiBot (False)
End Sub

Private Sub Form_Unload(Cancel As
Integer)
Screen.MousePointer = vbDefault
End Sub

Private Sub IMAFOTO_Click()
X = Buscalma(frmDATOSf, 0)

On Error Resume Next
IMAFOTO.Picture = LoadPicture(X)
txtFields(4).Text = X

End Sub

Private Sub txtFields_Change(Index As
Integer)

If (Index = 0) And
(txtFields(0).Enabled = False) Then

busca = txtFields(4).Text

If busca = "" Then
busca = UNIDAD +
"SIVEA\SISTEMA\UTILES\BMP\SUNA
Mico.BMP"
End If

On Error Resume Next
IMAFOTO = LoadPicture(busca)

End If

End Sub

FORMA: FRMDATOSG
Private Sub cmdAgrega_Click()
datPrimaryRS.Recordset.AddNew
Call HabiBot(True)
acciusu = "ALTA"
txtFields(0).SetFocus

End Sub

Private Sub cmdEliminar_Click()
T = T + "Esta por eliminar a un
alumno" + Chr(13)
T = T + "" + Chr(13)

```

```

T = T + "NOTA: El alumno será
eliminado" + Chr(13)
T = T + "Incluso como usuario " +
Chr(13)
T = T + "¿Confirmar operación?" +
Chr(13)
res = MsgBox(T, 33, "Eliminar una
ALUMNO")
If res = 1 Then

X = txtFields(0).Text
Data1.Refresh

With Data1.Recordset
MoveFirst
.FindFirst "NOCTA="" & X & ""
z = NOCTA.Value
If NoMatch = False Then
Delete
MoveNext
If EOF Then
.MoveLast
End If
.MoveFirst
For i = 1 To .RecordCount
.Edit
num Value = i
id Value =
Trim(Left(id Value, 5)) & Trim(Str(i))
.Update
.MoveNext
If EOF Then
Exit For
End If
Next i
MoveLast

End If
End With
Data1.Refresh

With datPrimaryRS.Recordset
Delete
.MoveNext
If EOF Then
.MoveNext
End If
End With
datPrimaryRS.Refresh

End If

End Sub

Private Sub cmdRenovar_Click()
'Esto sólo es necesario en aplicaciones
con múltiples usuarios
On Error Resume Next
datPrimaryRS.Refresh
Call HabiBot(False)
End Sub

Private Sub cmdActualizar_Click()
If txtFields(0).Text = "" Then
MsgBox ("Error en la captura de
datos")
Else
datPrimaryRS.UpdateRecord
datPrimaryRS.Recordset.Bookmark =
datPrimaryRS.Recordset.LastModified
Call HabiBot(False)

```

```

IdeN = txtFields(0).Text
Call ENTRAUSUARIO("ALUMNO")
End If
End Sub

```

```

Private Sub cmdCerrar_Click()
Screen.MousePointer = vbDefault
Unload Me
End Sub

```

```

Private Sub Command1_Click()
Call Habitbot(True)
acciusu = "MODI"
End Sub
Sub Habitbot(que As Boolean)

```

```
Dim Opu As Boolean
```

```

cmdActualizar.Enabled = que
cmdRenovar.Enabled = que
IMAFOTO.Enabled = que

```

```

For i = 0 To 14
txtFields(i).Enabled = que
Next i

```

```

If que = True Then
Opu = False
Else
Opu = True
End If

```

```

cmdAgrupar.Enabled = Opu
cmdEliminar.Enabled = Opu
cmdCerrar.Enabled = Opu
Command1.Enabled = Opu
datPrimaryRS.Visible = Opu

```

```
End Sub
```

```

Private Sub
datPrimaryRS_Error(DataErr As Integer,
Response As Integer)
MsgBox "Data error event hit err:" &
Error$(DataErr)
Response = 0 "Despreciar el error"
End Sub

```

```

Private Sub datPrimaryRS_Reposition()
Screen.MousePointer = vbDefault
On Error Resume Next
Mostrar la posición actual de registro
para dynasets y snapshots
datPrimaryRS.Caption = "Registro Nº: "
&
(datPrimaryRS.Recordset.AbsolutePosit
ion + 1)
End Sub

```

```

Private Sub
datPrimaryRS_Validate(Action As
Integer, Save As Integer)
'Aquí se sitúa el código de validación
Este evento se invoca cuando ocurre
la siguiente acción
Select Case Action
Case vbDataActionMoveFirst
Case vbDataActionMovePrevious
Case vbDataActionMoveNext
Case vbDataActionMoveLast
Case vbDataActionAddNew
Case vbDataActionUpdate
Case vbDataActionDelete

```

```

Case vbDataActionFind
Case vbDataActionBookmark
Case vbDataActionClose
Screen.MousePointer = vbDefault
End Select
Screen.MousePointer = vbHourglass
End Sub

```

```

Private Sub Form_Load()
Habitbot (False)
End Sub

```

```

Private Sub Form_Unload(Cancel As
Integer)
Screen.MousePointer = vbDefault
End Sub

```

```
Private Sub IMAFOTO_Click()
```

```
X = Buscalma(frmDATOSG, 0)
```

```

On Error Resume Next
IMAFOTO.Picture = LoadPicture(X)
txtFields(4).Text = X

```

```
End Sub
```

```
Private Sub txtFields_Change(Index As
Integer)
```

```

If (Index = 0) And (txtFields(0).Enabled
= False) Then
busca = txtFields(4).Text
If busca = "" Then
busca = UNIDAD +
"SIVEA\SISTEMA\UTILES\BMP\SUNA
Mico BMP"
End If
On Error Resume Next
IMAFOTO = LoadPicture(busca)

```

```
End If
```

```
End Sub
```

```
Private Sub txtFields_KeyPress(Index
As Integer, KeyAscii As Integer)
```

```

If (KeyAscii = 13) And (Index < 14)
Then
If Index = 3 Then
Index = 4
End If
txtFields(Index + 1).SetFocus
End If
End Sub

```

```
FORMA: FRMENEPE
```

```

Dim inde, norep As Integer
Dim x1(5) As Integer
Dim y1(5) As Integer

```

```

Private Sub Form_Activate()
Me.MousePointer = 0
End Sub

```

```
Private Sub Form_Load()
```

```

Me.MousePointer = 11
On Error Resume Next
Unload FRMENTRA

```

```
If MusicaON = 1 Then
```

```

Call TocaMus(4)
Else
SIVEA OLEP DoVerb (0)
End If

```

```

If ENTRACAN = 1 Then
Else
ENTRACAN = 1
End If

```

```

For i = 0 To 4
x1(i) = PIC1ENEP(i).Left
y1(i) = PIC1ENEP(i).Top
PIC1ENEP(i).Height = 70
PIC1ENEP(i).Width = 90
Next i

```

```

inde = -1
If SIVEA.mnubarraH.Checked = True
Then

```

```
Call SIVEA.barraOnTop
```

```

End If
Me.MousePointer = 0
End Sub

```

```
Private Sub image2_Click()
```

```
Call FinSesion
```

```
End Sub
```

```

Private Sub
image4_MouseMove(Button As Integer,
Shift As Integer, X As Single, Y As
Single)

```

```

For i = 0 To 4
PIC1ENEP(i).Left = x1(i)
PIC1ENEP(i).Top = y1(i)
PIC1ENEP(i).Height = 70
PIC1ENEP(i).Width = 90
Next i
End Sub

```

```
Private Sub Label1_Click(Index As
Integer)
```

```

If (Acceso = "ST") And (Index = 3)
Then
MsgBox ("No esta autorizado el
acceso a este modulo")
Else
On Error Resume Next
OLEM.Action = 9
Call ADONDEVOY(Index)
End If
End Sub

```

```
Private Sub PIC1ENEP_Click(Index As
Integer)
```

```

If (Acceso = "ST") And (Index = 3)
Then
MsgBox ("No esta autorizado el
acceso a este modulo")
Else
Call ADONDEVOY(Index)
End If
End Sub

```

```

Private Sub
PIC1ENEP_MouseMove(Index As
Integer, Button As Integer, Shift As
Integer, X As Single, Y As Single)

```

```
Select Case Index
Case 0
Mensa1 = "Visitar las aulas y tomar
clase"
Case 1
Mensa1 = "Dirigirse a los
laboratorios y realizar las prácticas"
Case 2
Mensa1 = "Ir a la biblioteca a
consultar algún libro"
Case 3
Mensa1 = "Establecer el orden y
buen funcionamiento de la escuela"
Case 4
Mensa1 = "Conocer mas acerca
del sistema y de la ENEP aragón."
Case Else
Mensa1 = "Seleccione algun lugar
para trabajar."
End Select
```

```
For i = 0 To 4
If i <> Index Then
Label1(i).Visible = False
Label11(i).Visible = False
PIC1ENEP(i).Height = 75
PIC1ENEP(i).Width = 75
PIC1ENEP(i).Top = y1(i)
PIC1ENEP(i).Left = x1(i)
End If
Next i
```

```
If inda <> Index Then
PIC1ENEP(Index).Top = y1(Index)
- 50 PIC1ENEP(Index).Left = x1(Index)
- 50 PIC1ENEP(Index).Height = 150
PIC1ENEP(Index).Width = 200
End If
```

```
SIVEA.statbar1.SimpleText = Mensa1
Label1(Index).Visible = True
Label11(Index).Visible = True
inde = Index
```

End Sub

```
Private Sub picenep_Click(Index As
Integer)
If Index = 2 Then
frmacerca.Show
End If
End Sub
```

```
Private Sub picenep_Db1Click(Index As
Integer)
If Index = 2 Then
frmacerca.Show
End If
End Sub
```

FORMA: FRMENTRA

```
Dim X As Integer
Option Explicit

Private Sub Form_KeyPress(KeyAscii
As Integer)
Unload Me
End Sub
```

```
Private Sub Form_Load()
FRMENTRA.MousePointer = 11

On Error Resume Next
SetWindowPos FRMENTRA.hWnd,
HWND_TOPMOST, FRMENTRA.Left /
15, _
FRMENTRA.Top / 15,
FRMENTRA.Width / 15, _
FRMENTRA.Height / 15,
SWP_NOACTIVATE Or
SWP_SHOWWINDOW

lblVersion.Caption = "Versión " &
App.Major & "." & App.Minor & "." &
App.Revision

lblProductName.Caption = App.Title

X = 0
Data1.Refresh
UNIDAD = Data1.Recordset(1) Value

'On Error Resume Next
OLEM.CreateEmbed UNIDAD +
"sivea\SISTEMA\UTILES\SONIDOS\sacc
eso.wav"
frmENTRA.OLEM
End Sub
```

```
Private Sub Form_Unload(Cancel As
Integer)
End Sub

Private Sub Frame1_Click()
Unload Me
End Sub

Private Sub Timer1_Timer()
X = X + 1
If (X >= 1) And (X < 4) Then
Unload Me
End If
End Sub
```

FORMA:FMRFACIL

```
Dim actual As Integer
Private Sub Form_Load()
ventana = 300
actual = 1
If MusicaON = 1 Then
Call TocaMus(5)
Else
SIVEA.OLEP.DoVerb (0)
End If
End Sub

Private Sub Form_Unload(Cancel As
Integer)
Call frmmenu.Regreso
```

```
End Sub

Private Sub
Image1_MouseMove(Button As Integer,
Shift As Integer, X As Single, Y As
Single)
Call muestra(True)
```

```
End Sub

Private Sub sscgob2_Click(Index As
Integer)
On Error Resume Next
OLEM.Action = 9
Select Case Index
```

```
Case 0
Call Egeveal(4)
Case 1
FRMCLASES.Show
Case 2
FrmTarea Show
Case 3
FRMLISTALUMNO.Show
Case 4
Call Cierrame
Case 5
frmagenda Show
End Select
```

```
If Index = 4 Then
Unload frmfacil
Else
Call muestra(False)
End If
```

```
End Sub

Sub muestra(Acc1 As Boolean)

Image2.Enabled = Acc1
Image3.Enabled = Acc1
Image4.Enabled = Acc1
If Acc1 = False Then
frmmenu.Hide
Else
frmmenu.Show
End If
```

```
End Sub
Sub Cierrame()
ventana = 300
Call frmmenu.Regreso
```

```
End Sub

Private Sub sscgob2_MouseMove(Index
As Integer, Button As Integer, Shift As
Integer, X As Single, Y As Single)
If actual <> Index Then

For i = 0 To 3
Label1(i).Visible = False
sscgb2(i).BevelWidth = 1
Next i
If Index < 4 Then
sscgb2(Index).BevelWidth = 3
Label1(Index).Visible = True
actual = Index
End If
```

```
End If

End Sub
```

FORMA:FRMGOBIERNO
Dim VALORb, disMI As Integer

<pre> Private Sub AniPushButton1_Click(Index As Integer) Call Regreso End Sub Private Sub apbf_Click(Index As Integer) Select Case Index Case 0 FRMLISTAFACIL.Show Case 1 frmDATOSf.Show Case 2 FRMLISTALUMNO.Show Case 3 frmDATOSG.Show End Select End Sub Private Sub Form_Load() ventana = 2 If MusicaON = 1 Then Call TocaMus(6) Else SIVEA OLEP.DoVerb (0) End If VALORb = gobitem(0).Height End Sub Sub Acomoda(dadoi As Integer) FRMGOBIERNO.MousePointer = 11 disMI = VALORb Do disMI = disMI - 5 gobitem(0).Height = disMI gobitem(1).Height = disMI gobitem(2).Height = disMI gobitem(3).Height = disMI On Error Resume Next gobitem(dadoi).Height = VALORb Loop Until disMI <= 5 SSTab1 Tab = dadoi SSTab1 Visible = True SSTab1 SelfFocus Select Case dadoi Case 1 frmmenu.ssbots.Tab = 8 OLEM.Action = 9 frmfaul Show End Select FRMGOBIERNO.MousePointer = 0 End Sub Private Sub Form_Unload(Cancel As Integer) Call frmmenu.Regreso End Sub Private Sub gobitem_Click(Index As Integer) Call Acomoda(Index) End Sub </pre>	<pre> Private Sub IMA_Click(Index As Integer) panelce(0).Visible = False panelce(1).Visible = False If Index < 2 Then panelce(Index).Visible = True Label1(Index) Visible = True End If Select Case Index Case 0 Case 2 End Select End Sub Private Sub IMA_MouseMove(Index As Integer, Button As Integer, Shift As Integer, X As Single, Y As Single) Label1(0) Visible = False Label1(1) Visible = False If Index < 2 Then Labr*'1(Index).Visible = True End If End Sub Private Sub Image3_Click() frmLABS1 Show End Sub Private Sub Image5_Click() Unload Me End Sub Private Sub Image6_Click() frmmano Show End Sub Private Sub Image7_Click() frmusuarios.Show End Sub Private Sub panelce_MouseMove(Index As Integer, Button As Integer, Shift As Integer, X As Single, Y As Single) If Index < 2 Then panelce(Index).Visible = True Label1(Index) Visible = True End If End Sub Private Sub sscgob_Click(Index As Integer) FRMHISTORIAL.Show End Sub Private Sub sscgob2_Click(Index As Integer) End Sub Private Sub sscgob3_Click(Index As Integer) frmALUMNOS Show End Sub </pre>	<pre> Private Sub SSTab1_Click(PreviousTab As Integer) SSTab1 Visible = False End Sub Private Sub Timer1_Timer() If Image2.BorderStyle = 1 Then Image2 BorderStyle = 0 Else Image2 BorderStyle = 1 End If End Sub Sub Regreso() For i = 0 To 3 On Error Resume Next gobitem(i) Height = VALORb Next i SSTab1 Visible = False panelce(0) Visible = False panelce(1) Visible = False frmmenu Show End Sub FORMA.FRMLABS Dim VALORb As Integer Dim nombre As String ' Crea variables de objeto para objetos de Data Access Dim myDb As Database, myRs As Recordset Private Sub Form_Load() VALORb = SSCLAB(0) Height If MusicaON = 1 Then Call TocaMus(7) Else SIVEA OLEP DoVerb (0) End If ' Establece la Database como la base de datos BIBLIO MDB Set myDb = DBEngine.Workspaces(0) OpenDatabas e("C:\sivea\institucion\gobierno\registros \bdsivea mdb") ' Establece el conjunto de registros como la tabla "Autores". Set myRs = myDb.OpenRecordset("labs", dbOpenDynaset) End Sub Private Sub Form_Unload(Cancel As Integer) Call frmmenu.Regreso End Sub Private Sub Image4_Click() Call momenu("S") If sspanlab.Visible = True Then sspanlab.Visible = False For i = 0 To 2 SSCLAB(i) Height = VALORb </pre>
--	---	---

```

Next I
Else
  Unload Me
End If
End Sub

Sub LLENALISTA(AREA As String)

  Call cualLlena

  ' Crea una variable para agregar
  objetos ListItem.
  Dim itmX As ListItem

  ' Mientras el registro no sea el último,
  agrega un objeto ListItem.

  myRs.MoveFirst

  While Not myRs.EOF

    If myRs!tipod = AREA Then

      Set itmX =
      ListView1.ListItems.Add(, ,
      CStr(myRs!nombre), 1) ' NOMBRE
      DEL CURSO

      Select Case myRs!tipod
      Case 1
        itmX.Icon = 1
        itmX.SmallIcon = 1
        'Establece un icono desde ImageList2
        mx = "Documento"
      Case 2
        itmX.Icon = 2
        itmX.SmallIcon = 2
        mx = "Práctica de
laboratorio"
      Case 3
        itmX.Icon = 3
        itmX.SmallIcon = 3
        mx = "Archivo ejecutable"

      End Select
      ' Si el campo ID CURSO no es
      nulo, establece SubItem 1 como él.

      If Not IsNull(myRs!id) Then
        itmX.SubItems(1) =
        CStr(myRs!id) ' ID de CURSO.
      End If

      If mx <> "" Then
        itmX.SubItems(1) = mx ' ID de
CURSO.
      End If

      End If

      myRs.MoveNext ' Pasa al
siguiente registro.

    Wend

    ListView1.SortKey = 2
    ListView1.View = 0
  End Sub

  Sub cualLlena()

    Dim ctmX As ColumnHeader
    ' Agrega ColumnHeaders. El ancho
    de las columnas es el ancho
    ' del control dividido por el número de
    objetos ColumnHeader

    ListView1.ListItems.Clear
    ' ListView1.ColumnHeaders.Clear
    Set ctmX =
    ListView1.ColumnHeaders.Add(, ,
    "NOMBRE", ListView1.Width / 3)
    Set ctmX =
    ListView1.ColumnHeaders.Add(, ,
    "IDCURSO", ListView1.Width / 3)
    Set ctmX =
    ListView1.ColumnHeaders.Add(, , "TIPO
DE ARCHIVO", ListView1.Width / 3)

    ListView1.ListIcons = ImageList1
    ListView1.SmallIcons = ImageList2

  End Sub

  Private Sub
  ListView1_ColumnClick(ByVal
  ColumnHeader As ColumnHeader)
    ' se orden por los subelementos de
    esa columna
    Establece el SortKey como el Index
    del ColumnHeader - 1
    ListView1.SortKey =
    ColumnHeader.Index - 1
    ' Establece Sorted a True para
    ordenar la lista
    ListView1.Sorted = True
  End Sub

  Private Sub ListView1_DblClick()

    Call Ejecuta(nombre)
  End Sub

  Private Sub ListView1_ItemClick(ByVal
  Item As ComctlLib.ListItem)
    nombre = ListView1.SelectedItem
  End Sub

  Private Sub
  ListView1_KeyPress(KeyAscii As
  Integer)
    If KeyAscii = 13 Then
      Call Ejecuta(nombre)

    End If
  End Sub

  Sub Ejecuta(busno As String)
    'procedimiento para ejecutar archivos
    Dim RetVal
    frmLabs.MousePointer = 11
    Call momenu("H")
    myRs.MoveFirst
    myRs.FindFirst "nombre =" & busno
    & ""
    If myRs.NoMatch = False Then
      Select Case myRs!tipod
      Case 1
        Call ponpiza(myRs!DIRECC)
      Case 2
        On Error Resume Next
        codigo.Show
      Case 3

        Call ponpiza(myRs!DIRECC)

        If Err <> 0 Then
          MsgBox "Este documento
debera ser editado de otra forma"
          End If

        End Select

        End If

        frmLabs.MousePointer = 0
      End Sub

      Sub ponpiza(OLEDIR As String)
        On Error Resume Next
        OLE1.CreateLink OLEDIR

        If (Err) Then
          If Err = 30015 Then
            MsgBox "No es un objeto
válido."
          Else
            MsgBox Error$
          End If
        End If
      End Sub

      Private Sub Toolbar1_ButtonClick(ByVal
      Button As Button)
        ' Cambia ListView con el toolbar
        Select Case Button.Key
        Case "icono"
          X = 0
          ListView1.SortKey = 0
        Case "iconito"
          X = 1
          ListView1.SortKey = 1
        Case "lista"
          X = 2
          ListView1.SortKey = 2
        Case "reporte"
          X = 3
          ListView1.SortKey = 0
        End Select
        ListView1.View = X
      End Sub

      Private Sub SSCLAB_Click(Index As
      Integer)
        Dim disMI As Integer

        disMI = VALORb

        Do
          disMI = disMI - 5
          SSCLAB(0).Height = disMI
          SSCLAB(1).Height = disMI
          SSCLAB(2).Height = disMI
          SSCLAB(Index).Height = VALORb
        Loop Until disMI <= 5

        Call AccionaPan
        Call LLENALISTA(Index + 1)
      End Sub

      Sub AccionaPan()
        sspanlab.Visible = True
      End Sub

      Private Sub
      SSCLAB_MouseMove(Index As Integer,

```

```

Button As Integer, Shift As Integer, X As
Single, Y As Single)
Dim men As String

Select Case Index
Case 0
    men = "Documentos
relacionados con las prácticas,
presentaciones, textos, etc."
Case 1
    men = "Prácticas realizadas en
el sistema"
Case 2
    men = "Programas y
aplicaciones ejecutables,relacionados
con la ingeniería"
End Select
pan1.Caption = men
End Sub
    
```

FORMA FRMCONTROLABS

```

Private Sub cmbus_Click()
txtFields(3).Text = Buscalma(frmLABS1,
1)
End Sub

Private Sub cmdAgregar_Click()
With datPrimaryRS.Recordset
.MoveLast
X = RecordCount + 1

End With
datPrimaryRS.Recordset.AddNew
txtFields(0).Text = X
cmbus.Enabled = True

End Sub

Private Sub cmdEliminar_Click()
mens = " ¿Esta segura de
eliminar este "
mens = mens & Chr(13) & "
documento de los laboratorios?"
mens = mens & Chr(13) & Chr(13) & "
NOTA. Si lo hace solo se eliminara"
mens = mens & Chr(13) & " el registro
en la base de datos"
X = MsgBox(mens, vbYesNo)

If X = vbYes Then
With datPrimaryRS.Recordset
.Delete
.MoveNext
If .EOF Then .MoveLast
MoveFirst
For i = 1 To .RecordCount
.Edit
id.Value = i
.Update
.MoveNext
If .EOF Then Exit For
Next i
.MoveLast
End With

End If

End Sub

Private Sub cmdmodi_Click()
datPrimaryRS.Recordset.Edit
cmbus.Enabled = True
    
```

```

End Sub

Private Sub cmdRenovar_Click()
'Esto sólo es necesario en aplicaciones
con múltiples usuarios
datPrimaryRS.Refresh
cmbus.Enabled = False

End Sub

Private Sub cmdActualizar_Click()
datPrimaryRS.UpdateRecord
datPrimaryRS.Recordset.Bookmark =
datPrimaryRS.Recordset.LastModified
cmbus.Enabled = False

End Sub

Private Sub cmdCerrar_Click()
Screen.MousePointer = vbDefault
Unload Me
End Sub

Private Sub Combo1_Click()

txtFields(1).Text = Combo1.ListIndex
+ 1

End Sub

Private Sub
Combo1_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
txtFields(2).SetFocus
End If
End Sub

Private Sub Command1_Click()

End Sub

Private Sub
datPrimaryRS_Reposition()
Screen.MousePointer = vbDefault
On Error Resume Next
'Mostrar la posición actual de registro
para dynasets y snapshots
datPrimaryRS.Caption = "Record. " &
(datPrimaryRS.Recordset.AbsolutePosi
tion + 1)
End Sub

Private Sub
datPrimaryRS_Validate(Action As
Integer, Save As Integer)
'Aquí se sitúa el código de validación
'Este evento se invoca cuando ocurre
la siguiente acción
Select Case Action
Case vbDataActionMoveFirst
Case vbDataActionMovePrevious
Case vbDataActionMoveNext
Case vbDataActionMoveLast
Case vbDataActionAddNew
Case vbDataActionUpdate
Case vbDataActionDelete
    
```

```

Case vbDataActionFind
Case vbDataActionBookmark
Case vbDataActionClose
Screen.MousePointer = vbDefault
End Select
Screen.MousePointer = vbHourglass
End Sub

Private Sub Form_Load()

Combo1.Clear
Combo1.AddItem "Documento"
Combo1.AddItem "Práctica"
Combo1.AddItem "Programa"

End Sub

Private Sub Form_Unload(Cancel As
Integer)
Screen.MousePointer = vbDefault
End Sub

Private Sub txtFields_Change(Index As
Integer)
If Index = 1 Then
Select Case txtFields(Index).Text
Case 1
te = "Documento"
Case 2
te = "Práctica"
Case 3
te = "Programa"
End Select
Combo1.Text = te

End If
End Sub

Private Sub txtFields_KeyPress(Index
As Integer, KeyAscii As Integer)
If Index = 2 Then
If KeyAscii = 13 Then
txtFields(3).SetFocus
End If
End If
End Sub
    
```

FORMA:FRMLISTAFACIL

```

Private Sub cmdCerrar_Click()
Screen.MousePointer = vbDefault
Unload Me
End Sub

Private Sub
datPrimaryRS_Error(DataErr As Integer,
Response As Integer)
MsgBox "Data error event hit err." &
Error$(DataErr)
Response = 0 'Despreciar el error
End Sub

Private Sub
datPrimaryRS_Reposition()
Screen.MousePointer = vbDefault
On Error Resume Next
End Sub

Private Sub
datPrimaryRS_Validate(Action As
Integer, Save As Integer)
'Aquí se sitúa el código de validación
'Este evento se invoca cuando ocurre
la siguiente acción
Select Case Action
    
```

```

Case vbDataActionMoveFirst
Case vbDataActionMovePrevious
Case vbDataActionMoveNext
Case vbDataActionMoveLast
Case vbDataActionAddNew
Case vbDataActionUpdate
Case vbDataActionDelete
Case vbDataActionFind
Case vbDataActionBookmark
Case vbDataActionClose
    Screen.MousePointer = vbDefault
End Select
Screen.MousePointer = vbHourglass
End Sub
    
```

```

Private Sub Form_Unload(Cancel As Integer)
    Screen.MousePointer = vbDefault
End Sub
    
```

```

Private Sub Form_Resize()
    On Error Resume Next
    'Esto redimensionará la cuadrícula al
    cambiar el tamaño del formulario
    grdDataGrid.Height = Me.ScaleHeight -
    picButtons.Height - 30
End Sub
    
```

FORMA: FRMLISTAALUMNO

```

Private Sub cmdCerrar_Click()
    Screen.MousePointer = vbDefault
    Unload Me
End Sub
    
```

```

Private Sub datPrimaryRS_Error(DataErr As Integer, Response As Integer)
    'Aquí es dónde puede situar el código de
    tratamiento de error
    'Si desea ignorar los errores, quite el
    comentario de la siguiente línea
    'Si desea capturarlos, agregue código
    aquí para controlarlos
    MsgBox "Data error event hit err:" &
    Errors(DataErr)
    Response = 0 'Despreciar el error
End Sub
    
```

```

Private Sub datPrimaryRS_Reposition()
    Screen.MousePointer = vbDefault
    On Error Resume Next
End Sub
    
```

```

Private Sub datPrimaryRS_Validate(Action As Integer, Save As Integer)
    'Aquí se sitúa el código de validación
    'Este evento se invoca cuando ocurre
    la siguiente acción
    Select Case Action
        Case vbDataActionMoveFirst
        Case vbDataActionMovePrevious
        Case vbDataActionMoveNext
        Case vbDataActionMoveLast
        Case vbDataActionAddNew
        Case vbDataActionUpdate
        Case vbDataActionDelete
        Case vbDataActionFind
        Case vbDataActionBookmark
        Case vbDataActionClose
            Screen.MousePointer = vbDefault
    End Select
    
```

```

Screen.MousePointer = vbHourglass
End Sub
    
```

```

Private Sub Form_Unload(Cancel As Integer)
    Screen.MousePointer = vbDefault
End Sub
    
```

```

Private Sub Form_Resize()
    On Error Resume Next
    'Esto redimensionará la cuadrícula al
    cambiar el tamaño del formulario
    grdDataGrid.Height = Me.ScaleHeight -
    picButtons.Height - 30
End Sub
    
```

FORMA: FRMMANTO

```

Private Sub cmdAgregar_Click()
    datPrimaryRS.Recordset.AddNew
End Sub
    
```

```

Private Sub cmdEliminar_Click()
    With datPrimaryRS.Recordset
        .Delete
        .MoveNext
        If .EOF Then MoveLast
    End With
End Sub
    
```

```

Private Sub cmdRenovar_Click()
    'Esto sólo es necesario en aplicaciones
    con múltiples usuarios
    datPrimaryRS.Refresh
End Sub
    
```

```

Private Sub cmdActualizar_Click()
    datPrimaryRS.UpdateRecord
    datPrimaryRS.Recordset.Bookmark =
    datPrimaryRS.Recordset.LastModified
End Sub
    
```

```

Private Sub cmdCerrar_Click()
    Screen.MousePointer = vbDefault
    Unload Me
End Sub
    
```

```

Private Sub datPrimaryRS_Error(DataErr As Integer, Response As Integer)
    MsgBox "Data error event hit err:" &
    Errors(DataErr)
    Response = 0 'Despreciar el error
End Sub
    
```

```

Private Sub datPrimaryRS_Reposition()
    Screen.MousePointer = vbDefault
    On Error Resume Next
    'Mostrar la posición actual de registro
    para dynasets y snapshots
    datPrimaryRS.Caption = "Record: " &
    (datPrimaryRS.Recordset.AbsolutePosition + 1)
End Sub
    
```

```

Private Sub datPrimaryRS_Validate(Action As Integer, Save As Integer)
    'Aquí se sitúa el código de validación
    'Este evento se invoca cuando ocurre
    la siguiente acción
    Select Case Action
        Case vbDataActionMoveFirst
    
```

```

Case vbDataActionMovePrevious
Case vbDataActionMoveNext
Case vbDataActionMoveLast
Case vbDataActionAddNew
Case vbDataActionUpdate
Case vbDataActionDelete
Case vbDataActionFind
Case vbDataActionBookmark
Case vbDataActionClose
    Screen.MousePointer = vbDefault
End Select
Screen.MousePointer = vbHourglass
End Sub
    
```

```

Private Sub Form_Unload(Cancel As Integer)
    Screen.MousePointer = vbDefault
End Sub
    
```

FORMA: FRMMENU

```

Sub Regreso()
    Call TocaMus(0)
    If (frmMenu.Visible = False) And
    (SIVEA.mnuherra.Checked = True)
    Then
        frmMenu.Visible = True
    End If
    
```

```

Select Case ventana
Case 0
    FRMSALE.Show
Case 1
    FRMENEP.WindowState = 2
    FRMENEP.Show
    FRMAULAS.Hide
    frmMenu.ssbots.Tab = 0
    ventana = 0
    
```

```

Case 2
    FRMENEP.WindowState = 2
    FRMENEP.Show
    FRMBIBLIO.Hide
    frmMenu.ssbots.Tab = 0
    ventana = 0
Case 3
    FRMENEP.WindowState = 2
    FRMENEP.Show
    frmMenu.ssbots.Tab = 0
    ventana = 0
    
```

```

Case 4
    FRMENEP.WindowState = 2
    FRMENEP.Show
    FRMGOBIERNO.Hide
    ventana = 0
    frmMenu.ssbots.Tab = 0
    
```

```

Case 5
    FRMAULAS.WindowState = 2
    FRMAULAS.Show
    frmSalon.Hide
    ventana = 1
    frmMenu.ssbots.Tab = 0
Case 100
    
```

```

FRMAULAS.SalonZoom(0).Visible =
False
    
```

```

frmMenu.ssbots.Tab = 3
ventana = 1
    
```

```

Case 200
    frmSalon.Show
    Unload pizarra
    ventana = 5
    
```

```

Case 300
    
```

<pre> ventana = 4 FRMGGOBIERNO.Show frmmenu.ssbots Tab = 5 End Select End Sub Private Sub SSCAULAS_Click(Index As Integer) Call FRMAULAS.PONAUULAS(Index) End Sub </pre>	<pre> Call Egeveal(0) Case "bar13" LIBRO Show Case "bar14" Call Egeveal(1) Case "bar15" Call frmsalon.PUPITREontop End Select frmmenu.Show End Sub </pre>	<pre> pupitre.Height / 15, SWP_NOACTIVATE Or SWP_SHOWWINDOW Else SetWindowPos pupitre,hWnd, HWND_NOTOPMOST, pupitre Left / 15, pupitre Top / 15, pupitre Width / 15, pupitre.Height / 15, SWP_NOACTIVATE Or SWP_SHOWWINDOW </pre>
<pre> Private Sub SSCfisica_Click(Index As Integer) Call FRMAULAS.PasaAulas(1, Index) End Sub Private Sub SSCMate_Click(Index As Integer) Call FRMAULAS.PasaAulas(1, Index) End Sub Private Sub SSCCONTROL_Click(Index As Integer) Select Case Index Case 0 Call Regreso Case 1 Call FinSesion Case 2 frmacerca.Show Case 3 frmmenu.Hide SIVEA.mnubarraH.Checked = False SIVEA.Ton/bar2.Buttons.Item(1).Value = fbrUnpressed End Select End Sub Private Sub SSCPRINCIPAL_Click(Index As Integer) Call ADONDEVOY(Index) End Sub Private Sub SSRibbon1_Click(Value As Integer) If Value = 0 Then frmmenu.Caption = "" Else frmmenu.Caption = "Mover barra principal" End If End Sub Private Sub Toolbar1_ButtonClick(ByVal Button As ComctlLib.Button) Call SelecMenu(Button.Key) End Sub Sub SelecMenu(CLAVE As String) frmmenu.Hide Select Case CLAVE Case "bar11" frmsalon.OLE1(0).DoVerb(0) Case "bar12" </pre>	<pre> FORMA: FRMSALE Private Sub Form_Deactivate() MsgBox "SALGÓ" End Sub Private Sub Form_KeyPress(KeyAscii As Integer) If UCase(Chr(KeyAscii)) = "S" Then End Else Unload Me End If End Sub Private Sub Form_Load() if frmmenu.ssbots.Tab = 7 Then MsgBox ("Salga del aula antes de abandonar SIVEA") Unload Me Else ventana = -1 Call ApagaCualquierSonido If MusicaON = 1 Then Call TocaMus(8) Else SIVEA.OLEP.DoVerb(0) End If End If End Sub Private Sub Form_LostFocus() Unload Me End Sub Private Sub LBSINO_Click(Index As Integer) Select Case Index Case 0 End Case 1 Unload Me End Select End Sub </pre>	<pre> End Sub Private Sub Command1_Click() frmsalon.MousePointer = 11 frmmenu.ssbots.Tab = 23 frmmenu.pb1.Value = 0 list1.Text = Space(10) + Label1.Caption frmsalon.MousePointer = 0 frmsalon.Hide pzarron Show End Sub Private Sub Form_Activate() frmmenu.ssbots Tab = 7 frmmenu.Visible = True ventana = 5 X = frmsalon.Caption frmsalon.Label1.Caption = X frmsalon.list1.Text = X frmsalon.Label6.Caption = X End Sub Sub INISALON() frmsalon.MousePointer = 11 For i = 1 To 99 Arretema(i) = "" Next i DATALISTA.Refresh DATALISTA.Recordset.MoveLast X = DATALISTA.Recordset.RecordCount DATALISTA.Recordset.MoveFirst list1.Clear CtaClase = 1 For i = 1 To X mate = DATALISTA.Recordset.materia.Value Tema = DATALISTA.Recordset.CAPITULO.Value If MateSele = mate Then If TemaSele = Tema Then list1.AddItem DATALISTA.Recordset.SubTema.Value Arretema(CtaClase) = DATALISTA.Recordset.SubTema.Value CtaClase = CtaClase + 1 </pre>
<pre> Case "bar12" SetWindowPos pupitre,hWnd, HWND_NOTOPMOST, pupitre Left / 15, pupitre Top / 15, pupitre Width / 15, pupitre.Height / 15, SWP_NOACTIVATE Or SWP_SHOWWINDOW End Sub </pre>	<pre> FORMA: FRMSALON Dim xoriginal, yoriginal As Integer Sub PUPITREontop() 'mnuSysInfo(Index).Checked = Not mnuSysInfo(Index).Checked If mnuSysInfo(Index).Checked Then SetWindowPos pupitre,hWnd, HWND_NOTOPMOST, pupitre Left / 15, pupitre Top / 15, pupitre Width / 15, </pre>	<pre> End If End If If Not DATALISTA.Recordset.EOF Then DATALISTA.Recordset.MoveNext End If Next i list1.Text = Label1.Caption Call ponpiza(list1.List(0)) xoriginal = OLE1(1).Width yoriginal = OLE1(1).Height frmsalon.MousePointer = 0 </pre>

```

frmmenu.ssbots.Tab = 7
End Sub

Private Sub Form_GotFocus()
frmmenu.Visible = True
End Sub

Private Sub Form_Load()

Me.MousePointer = 11

If MusicaON = 1 Then
Call TocaMus(9)
Else
SIVEA OLEP.DoVerb (0)
End If
Call INISALON

Me.MousePointer = 0

End Sub

Private Sub Form_Unload(Cancel As Integer)
FRMAULAS.SalonZoom(0).Visible = False
End Sub

Private Sub Image2_MouseMove(Index As Integer, Button As Integer, Shift As Integer, X As Single, Y As Single)

For i = 0 To 3
Label2(i).Visible = False
Shape1(i).Visible = False
Next i

Select Case Index
Case 0
mens = "Ver tareas y trabajos"
Case 1
mens = "Realizar evaluación"
Case 2
mens = "Prácticas de conocimientos"
Case 3
mens = "Dejar el salón, pasar al patio principal"
Case Else
If frmmenu.Visible = False Then
frmmenu.Visible = True
End If

End Select

If Index < 4 Then
Label2(Index).Visible = True
Shape1(Index).Visible = True
End If

SIVEA.statbar1.SimpleText = mens
End Sub

Private Sub Label2_Click(Index As Integer)
frmmenu.Visible = False

Select Case Index
Case 0
LIBRO.Show
Case 1
Call Egeneval(6)
Case 2

```

```

Call Egeneval(5)
Case 3
Call frmmenu.Regreso
End Select

End Sub

Private Sub list1_Click()
Call ponpiza(list1.List(1).ListIndex)
End Sub

Sub ponpiza(TEMASEL As String)

DIRE =
"C:\SIVEA\INSTITUCIONAULAS"
nombre = TEMASEL

frmsalon.DATALISTA.Recordset.FindFirst "Subtema = " & nombre & ""
If
frmsalon.DATALISTA.Recordset.NoMatch = False Then
archi =
frmsalon.DATALISTA.Recordset.LOCALRCHIVO
OLEDIR = DIRE + archi

On Error Resume Next
OLE1(0).CreateLink OLEDIR

If (Err) Then
If Err = 30015 Then
MsgBox "No es un objeto válido "
Else
MsgBox Error$
End If

End If

End Sub

Private Sub OLE1_LostFocus(Index As Integer)
OLE1(1).Width = xoriginal
OLE1(1).Height = yoriginal

End Sub

Private Sub OLE1_MouseDown(Index As Integer, Button As Integer, Shift As Integer, X As Single, Y As Single)
frmmenu.Visible = False
If Button = 2 Then
OLE1(1).Width = xoriginal
OLE1(1).Height = yoriginal
End If
End Sub

Private Sub OLE1_MouseMove(Index As Integer, Button As Integer, Shift As Integer, X As Single, Y As Single)
Dim mens As String
If Index = 0 Then
mens = "Ver la clase"
Else
mens = "Ver la guía de ayuda de clases"
End If

```

```

SIVEA.statbar1.SimpleText = mens
End Sub

Private Sub OLE1_MouseUp(Index As Integer, Button As Integer, Shift As Integer, X As Single, Y As Single)
If Button = 2 Then
If Index = 1 Then
OLE1(1).Width = 3000
OLE1(1).Height = 3000
End If
End If
End Sub

FORMA FRMTAREA

Dim nova As Boolean
Dim arrem(14) As String
Dim TipoTi As String

Private Sub sscgob2_Click(Index As Integer)
Unload Me
End Sub
Sub MuevePantallas()
For i = 1 To 2
pictarea(i).Left = -20000
picbots(i).Left = -2000
Next i

End Sub

Private Sub apb_Click(Index As Integer)

If Combo1.Text = "" Then
MsgBox ("Seleccione la materia con la cual se desea trabajar")
Else
If Combo2.Text = "" Then
MsgBox ("Seleccione el tema con la cual se desea trabajar")
Else
If (Index > 0) And (Index < 3) Then
TreeView1.Enabled = False
Call MuevePantallas
pictarea(Index).Left = 5250
picbots(Index).Left = 2800

End If

Select Case Index
Case 0
Call MuevePantallas
Case 1
txtarea.Text = "Escriba aquí el título de la tarea."
Ltdescr.Text = "Escriba aquí lo que el alumno deberá de realizar"
OLEM.Visible = False
For i = 1 To 2
Ibpriori(i).BackColor = WHITE
Next i
Ibfecha(0).Caption = Date
Ibfecha(1).Caption = Date
Data1.Recordset.AddNew
Case 2
BuscarTa
Case 3
GuardaTa
Case 4, 6
TreeView1.Enabled = True

```

```

Call MuevePantaa
End Select
End If
End If
End Sub
Sub GuardaTa()
    nova = False
    With Data1.Recordset
        fechaa.Value =
    lbfecaa(0) Caption
        fechae.Value =
    lbfecaa(1) Caption
        titulo.Value = txtarea.Text
        descrip.Value = txtdescri.Text
        If OLEM.Visible = True Then
            ARCHIVO.Value =
OLEM.SourceDoc
        Else
            ARCHIVO.Value = ""
        End If
        For i = 0 To 2
            If lbprior(i).BackColor =
YELLOW Then
                PRIO = i
            End If
        Next i
        Select Case PRIO
            Case 2
                imi = "BAJA"
            Case 1
                imi = "MEDI"
            Case 0
                imi = "ALTA"
            Case Else
                imi = ""
        End Select
        priori.Value = imi
        If Combo1.Text =
"MATEMATICAS" Then
            materia.Value = "01"
        Else
            materia.Value = "02"
        End If
        posi = (Combo2.ListIndex + 1)
        .Tema.Value = arremtem(posi)
        tipot.Value = TipoTt
        On Error Resume Next
        .Update
    End With
    TreeView1.Enabled = True
    Call MuevePantaa
End Sub
Sub BuscarTa()
Set nodx = TreeView2.Nodes.Add(, ,
"A", "UNAS TAREAS")
Set nodx =
TreeView2.Nodes.Add("A", twwChild,
"A0", "JUAN PEREZ")

```

```

Set nodx =
TreeView2.Nodes.Add("A", twwChild,
"B0", "PANCHO X")
Set nodx =
TreeView2.Nodes.Add("A", twwChild,
"C0", "CHOFIS A")
nodx.EnsureVisible
End Sub
Private Sub apbc_Click()
    Unload Me
End Sub
Private Sub cmfecha_Click(Index As
Integer)
    valor = Index
    nova = True
    Call ponfecha
End Sub
Private Sub Combo1_Click()
    Dim dato, DATI As String
    Dim i As Integer
    Combo2.Clear
    If Combo1.Text = "MATEMATICAS"
Then
        dato = "01"
    Else
        dato = "02"
    End If
    With Data2.Recordset
        .MoveFirst
        i = 1
        Do While Not EOF
            DATI = Left(.CLAVE.Value, 2)
            If dato = DATI Then
                Combo2.AddItem
(.CAPITULO.Value)
                arremtem(i) = CLAVE.Value
                i = i + 1
            End If
            If EOF Then
                Exit Do
            Else
                .MoveNext
            End If
        Loop
    End With
End Sub
Private Sub Command1_Click()
    OLEDIR = buscaArchivo(FrmTarea)
Err.Clear
On Error Resume Next
OLEM.CreateLink OLEDIR
If (Err) Then
    If Err = 30015 Then
        MsgBox "No es un objeto
válido."
    Else
        MsgBox Error$
    End If

```

```

End If
OLEM.Visible = True
End Sub
Private Sub Form_Load()
    Dim nodx As Node
    Set nodx = TreeView1.Nodes.Add(, ,
"R", "TAREAS")
    Set nodx =
TreeView1.Nodes.Add("R", twwChild,
"A0", "EJERCICIOS")
    Set nodx =
TreeView1.Nodes.Add("R", twwChild,
"B0", "TRABAJOS")
    Set nodx =
TreeView1.Nodes.Add("R", twwChild,
"C0", "LECTURAS")
    Set nodx =
TreeView1.Nodes.Add("R", twwChild,
"D0", "PRACTICAS")
    Set nodx =
TreeView1.Nodes.Add("R", twwChild,
"E0", "OTROS")
    nodx.EnsureVisible
    LBFECHEA.Caption = Date
    Combo1.Clear
    Combo1.AddItem "MATEMATICAS"
    Combo1.AddItem "FISICA"
    Data2.Refresh
    lbfecaa(0).Caption = Date
    lbfecaa(1).Caption = Date
    TipoTt = ""
    Call MuevePantaa
End Sub
Private Sub Form_Unload(Cancel As
Integer)
    Call frmfact.muestra(True)
End Sub
Private Sub lpriori_Click(Index As
Integer)
    For i = 0 To 3
        lbprior(i).BackColor = WHITE
    Next i
    lbprior(Index).BackColor = YELLOW
End Sub
Private Sub lbfecaa_Change(Index As
Integer)
    Dim x1, x2, x3, X, d1, d2, d3, diahoy
As Integer
    If nova = True Then
        If lbfecaa(0).Caption >
lbfecaa(1).Caption Then

```

```

MsgBox ("La fecha de entrega
debe ser posterior a la fecha de
asignación")
lbfecaa(1) = lbfecaa(0)
End If

x1 = Day(lbfecaa(Index))
x2 = Month(lbfecaa(Index)) * 100
x3 = Year(lbfecaa(Index))
X = x1 + x2 + x3

d1 = Day(LBFECCHA)
d2 = Month(LBFECCHA) * 100
d3 = Year(LBFECCHA)
diaohoy = d1 + d2 + d3

If X < diaohoy Then
MsgBox ("La fecha debe ser
posterior a el día de hoy")

End If

End If
End Sub

Sub editaTarea()

TreeView1.Enabled = False

pictarea(1).Left = 5270
picbots(1).Left = 2700

busca =
Trim(List1.List(List1.ListIndex))

With Data1.Recordset
FindFirst "Descrip=" & busca &
---

If NoMatch = False Then
nova = False
lbfecaa(0).Caption =
.fecaa Value
lbfecaa(1).Caption =
fechae Value

txttarea.Text =
tituloT Value
txtdescri.Text =
.descri Value

If .ARCHIVO.Value = ""
Then
OLEM.Visible = False
Else
OLEM.Visible = True
End If
PRIO =
UCase(.priori.Value)
Select Case PRIO
Case "BAJA"
imi = 2
Case "MEDI"
imi = 1
Case "ALTA"
imi = 0
Case Else
imi = 3
End Select
'lbpriori.Caption =
.priori.Value

For i = 0 To 2

```

```

lbpriori(i).BackColor =
WHITE

Next i
lbpriori(imi).BackColor =
YELLOW

End With

Data1.Recordset.Edit

End Sub

Private Sub List1_DblClick()
Call editaTarea
End Sub

Private Sub List1_KeyPress(KeyAscii
As Integer)
If KeyAscii = 13 Then
Call editaTarea
Else
If KeyAscii = 8 Then
mens = "¿Esta seguro que desea
eliminar esta tarea?"
X = MsgBox(mens, 35, "Eliminar
tarea")
End If
End If

End Sub

Sub TreeView1_Click()
Dim dato As String
Dim posi As String

List1.Clear

busca =
UCase(TreeView1.Nodes(TreeView1.Se
lectedItem.Index).Text)

TipoTl = busca

If Combo1.Text = "MATEMATICAS"
Then
dato = "01"
Else
dato = "02"
End If
posi = arrem(Combo2.ListIndex + 1)

With Data1.Recordset

.MoveLast
cuantos = RecordCount + 1
.MoveFirst
cualvaz = 0

Do While Not .EOF

If cualvaz = 0 Then
.FindFirst "tipot=" & busca & ""
Else
.FindNext "tipot=" & busca & ""
End If

If .NoMatch = False Then
If .materia.Value = dato Then
If .Tema.Value = Trim(posi)

Then
List1.AddItem
.descri.Value
End If
End If

```

```

End If

cualvaz = cualvaz + 1
If cualvaz > cuantos Then
Exit Do
Else
End If
.MoveNext
Loop

End With
End Sub

Private Sub txtdescri_DblClick()
txtdescri.Text = ""

End Sub

Private Sub txttarea_DblClick()
txttarea.Text = ""

End Sub

FORMA: FRMUSUARIO

Private Sub cmdEliminar_Click()
X = MsgBox("Esta seguro de eliminar
esta clave", vbYesNo)

If X = vbYes Then
End If
End Sub

Private Sub cmdRenovar_Click()
"Esto sólo es necesario en aplicaciones
con multiples usuarios
.datPrimaryRS.Refresh
.FRA1.Enabled = False
Unload Me
End Sub

Private Sub cmdActualizar_Click()
Dim i As Integer
Dim correcto As Boolean

correcto = True

For i = 0 To 4
If txtFields(i).Text = "" Then
correcto = False
i = 4
End If
Next i
If correcto = False Then
MsgBox ("No se han completado los
datos")
Else
.datPrimaryRS.UpdateRecord
.datPrimaryRS.Recordset.Bookmark
= datPrimaryRS.Recordset.LastModified
.FRA1.Enabled = False
Unload Me
End If

End Sub

Private Sub cmdCerrar_Click()
Screen.MousePointer = vbDefault
Unload Me
End Sub

Private Sub Combo1_Click()
txtFields(2).Text = Combo1.Text

```

```

If Usuario = "ALUMNO" Then
  TER = "STMYN"
  lblLabels(3).Caption = "NUMERO
DE CUENTA:"
Else
  lblLabels(3).Caption = "RFC:"
  Select Case Combo1.ListIndex
    Case 0
      TER = "FCMYN"
    Case 1
      TER = "DTMYN"
  End Select
End If

txtFields(1).Text = TER &
Trim(txtFields(0) Text)

End Sub

Private Sub
Combo1_KeyPress(KeyAscii As Integer)
  If KeyAscii = 13 Then
    txtFields(4).SetFocus
  End If
End Sub

Private Sub Data1_Validate(Action As
Integer, Save As Integer)

End Sub

Private Sub
datPrimaryRS_Error(DataErr As Integer,
Response As Integer)
  MsgBox "Data error event hit err:" &
Error$(DataErr)
  Response = 0 'Despreciar el error
End Sub

Private Sub datPrimaryRS_Reposition()
  Screen.MousePointer = vbDefault
  On Error Resume Next
  'Mostrar la posición actual de registro
  para dynasets y snapshots
  datPrimaryRS.Caption = "Record: " &
(datPrimaryRS.Recordset.AbsolutePosit
ion + 1)
End Sub

Private Sub
datPrimaryRS_Validate(Action As
Integer, Save As Integer)
  'Aquí se sitúa el código de validación
  'Este evento se invoca cuando ocurre
  la siguiente acción
  Select Case Action
    Case vbDataActionMoveFirst
    Case vbDataActionMovePrevious
    Case vbDataActionMoveNext
    Case vbDataActionMoveLast
    Case vbDataActionAddNew
    Case vbDataActionUpdate
    Case vbDataActionDelete
    Case vbDataActionFind
    Case vbDataActionBookmark
    Case vbDataActionClose
      Screen.MousePointer = vbDefault
  End Select
  Screen.MousePointer = vbHourglass
End Sub

Private Sub Form_Activate()
  Call ModAlta(acciusu, IdeN)

```

```

End Sub

Private Sub Form_Load()
  Combo1.Clear
  If Usuario = "ALUMNO" Then
    Combo1.AddItem "ESTUDIANTE"
  Else
    Combo1.AddItem "FACILITADOR"
    Combo1.AddItem "DIRECTOR"
  End If
End Sub

End Sub
Sub ModAlta(accius As String, ide As
String)

  ide = Left(ide, 9)

  If acciu = "ALTA" Then
    FRA1.Enabled = True
    Combo1.SetFocus
    With datPrimaryRS.Recordset
      MoveLast
      X = RecordCount + 1
      AddNew
      End With
    txtFields(0) Text = X
    txtFields(3).Text = ide
  Else
    FRA1.Enabled = True
    With datPrimaryRS.Recordset
      MoveFirst
      FindFirst "nocta =" & ide & ""
      If .NoMatch = False Then
        Edit
      End If
    End With
  End If
End Sub

Private Sub Form_Unload(Cancel As
Integer)
  Screen.MousePointer = vbDefault
End Sub

Private Sub txtFields_Change(Index As
Integer)
  If Index = 2 Then
    Combo1.Text = txtFields(2) Text

    If Usuario = "ALUMNO" Then
      lblLabels(3).Caption =
"NUMERO DE CUENTA."
    Else
      lblLabels(3).Caption = "RFC:"
    End If
  End If
End Sub

Private Sub txtFields_KeyPress(Index
As Integer, KeyAscii As Integer)
  If Index = 3 Then
    If KeyAscii = 13 Then
      txtFields(4).SetFocus
    End If
  End If

  If Index = 4 Then
    If KeyAscii = 13 Then
      mens = "Escriba de nuevo la
clave que acaba de dar"

```

```

X = InputBox(mens, "Confirmar
clave")
  If X <> txtFields(4) Text Then
    MsgBox ("La clave no es la
correcta intente de nuevo")
    txtFields(4) Text = ""
  End If
End If
End If
End Sub

FORMA. LIBRO

Public Sub Escogelema()
  Dim tex(99) As String
  Dim i, lim As Integer
  If TreeView1.SelectedItem.Index <> 1
  Then
    'salon.Label1.Caption =
TreeView1.SelectedItem.Text
    Select Case
TreeView1.SelectedItem.Index
      Case 2
        tex(1) = "Conjuntos y el sistema
de números reales "
        tex(2) = "Aritmetica y
propiedades de los números reales "
        tex(3) = "Exponentes "
        tex(4) = "Notación científica "
        tex(5) = "Resolución de
ecuaciones "
        tex(6) = "Uso de ecuaciones para
resolver problemas "
        tex(7) = "Aplicaciones de las
ecuaciones "
        lim = 7
      Case 3
        tex(1) = "Tablas y graficas "
        tex(2) = "El sistema de
coordenadas rectangulares."
        tex(3) = "Pendiente de una recta
no vertical "
        tex(4) = "Formulación de
ecuaciones de rectas "
        tex(5) = "Introducción a las
funciones "
        tex(6) = "Gráficas de otras
funciones."
        lim = 6
    End Select

    Unload Me

  End If
End Sub

Private Sub APBANEXA_Click()
  OLEDIR = buscaArchivo(LIBRO)
  On Error Resume Next
  ole2.CreateEmbed OLEDIR

  If (Err) Then
    If Err = 30015 Then
      MsgBox "No es un objeto
válido "
    Else
      MsgBox Error$
    End If
  End If
End Sub

ole2.Visible = True

```

```

End Sub

Private Sub apcancel_Click()
    Text1.Text = ""
    ole2.Visible = False
    ole2.SourceDoc = ""
    PonPanta (1)
End Sub

Private Sub cmin_Click()
    Unload Me
End Sub

Private Sub cmtarea_Click(Index As Integer)
    Call PonPanta(Index)
End Sub
Sub PonPanta(cual As Integer)
    Dim imi As Integer
    For i = 0 To 2
        pictarea(i).Left = -20000
        Next i

        pictarea(cual).Left = 3750
        pictarea(cual).Top = 680

        busca =
Trim(List1.List(List1.ListIndex))

        If cual = 1 Then
            Label3.Caption = busca
        End If

        With Data1.Recordset
            MoveFirst
            Do While Not .EOF
                If busca = "" Then
                    Exit Do
                Else
                    .FindFirst "Descrip="" & busca & ""
                    if NoMatch = False Then
                        lbfechaa(0).Caption =
                        .fechaa.Value
                        lbfechaa(1).Caption =
                        .fechaa.Value
                        lbfechae(0).Caption =
                        .fechae.Value
                        lbfechae(1).Caption =
                        .fechae.Value
                        LBTAREA.Caption =
                        .tituloT.Value

                        If (Len(.ARCHIVO.Value) >
0) Then
                            OLE1(1).Visible = True
                        Else
                            OLE1(1).Visible = False
                        End If
                        PRIO =
UCase(.priori.Value)
                        Select Case PRIO
                            Case "BAJA"
                                imi = 0
                            Case "MEDI"
                                imi = 1
                            Case "ALTA"
                                imi = 2
                        End Select
                        lbpriori.Caption =
                        20000
                        For i = 0 To 2
                            impriori(i).Left = -
                            Next i
                            impriori(im).Top = 3270
                            impriori(im).Left = 3120
                        End Do
                        End If
                        MoveNext
                        End If
                        Loop
                        End With
                    End Sub
                    Private Sub Command1_Click()
                        Unload Me
                    End Sub

                    Private Sub Form_Load()
                        Dim ncdx As Node
                        Set ncdx = TreeView1.Nodes.Add(,
"R", "TAREAS")
                        Set ncdx =
TreeView1.Nodes.Add("R", twwChild,
"A0", "EJERCICIOS")
                        Set ncdx =
TreeView1.Nodes.Add("R", twwChild,
"B0", "TRABAJO")
                        Set ncdx =
TreeView1.Nodes.Add("R", twwChild,
"C0", "LECTURAS")
                        Set ncdx =
TreeView1.Nodes.Add("R", twwChild,
"D0", "PRACTICAS")
                        Set ncdx =
TreeView1.Nodes.Add("R", twwChild,
"E0", "OTROS")

                        ncdx.EnsureVisible

                        busca = frmSalon.Caption

                        Label1.Caption = busca
                        LBFECHA.Caption = Date
                    End Sub

                    Private Sub Form_LostFocus()
                        Unload Me
                    End Sub

                    Private Sub Image2_Click(Index As Integer)
                        Dim i As Integer
                        Dim strNodos As String
                        strNodos = strNodos &
TreeView1.Nodes(i).Index & " " &
"Clave: " & TreeView1.Nodes(i).Key &
" " &
"Texto: " & TreeView1.Nodes(i).Text
& Chr(10)
                        Next i
                        MsgBox strNodos
                    End Sub

                    Private Sub ListView1_DbClick()
                        Call PonPanta(1)
                    End Sub

                    Private Sub Text1_DbClick()
                        Text1.Text = ""
                    End Sub

                    Private Sub TreeView1_Click()
                        Dim busca As String
                        Dim dato As String
                        Dim posi As String
                        Dim arretem(14) As String
                        Call PonPanta(0)
                        List1.Clear

                        busca =
UCase(TreeView1.Nodes(TreeView1.Se
lectedItem.Index).Text)

                        If MateSele = 2 Then
                            dato = "02"
                        Else
                            dato = "01"
                        End If

                        With Data2.Recordset
                            .MoveFirst
                            For i = 1 To 14
                                arretem(i) = CLAVE.Value
                                MoveNext
                                If .EOF Then
                                    Exit For
                                End If
                            Next i
                            End With
                            If MateSele = 2 Then
                                X = 7
                            Else
                                X = 0
                            End If

                            posi = arretem(TemaSele + X)

                            With Data1.Recordset

                                MoveLast
                                cuantos = .RecordCount + 1
                                .MoveFirst
                                cualvaz = 0

                                Do While Not .EOF

                                    If cualvaz = 0 Then
                                        .FindFirst "tipot="" & busca & ""
                                    Else
                                        .FindNext "tipot="" & busca & ""
                                    End If

                                    If NoMatch = False Then
                                        If .materia.Value = dato Then
                                            If Tema.Value = Trim(posi)
                                                Then
                                                    List1.AddItem
                                                    .descrp.Value
                                                    End If
                                        End If
                                    End If
                                End While
                            End With
                    End Sub
                End Sub
            End Sub
        End Sub
    End Sub

```

```

End If
End If

cualvaz = cualvaz + 1
If cualvaz > cuantos Then
Exit Do
Else
End If
' MoveNext
Loop

End With
-----

End Sub

Private Sub
TreeView1_KeyPress(KeyAscii As
Integer)
If KeyAscii = 13 Then
Call Escopgetema
End If
End Sub

```

FORMA: SIVEA

```

Sub barraOnTop()
SetWindowPos frmmenu.hWnd,
HWND_TOPMOST, frmmenu.Left / 15,
- frmmenu.Top / 15,
frmmenu.Width / 15, -
frmmenu.Height / 15,
SWP_NOACTIVATE Or
SWP_SHOWWINDOW

End Sub

Sub LLENADATOS()
Dim BUSCACTA As String
BUSCACTA = NOCUENTA
Prim = Asc(UCase(Mid(NOCUENTA,
1, 1)))

If ACceso = "ST" Then
SIVEA.DATANOM.Refresh

SIVEA.DATANOM.Recordset.FindFirst
"NOCTA=" & BUSCACTA & ""
If
SIVEA.DATANOM.Recordset.NoMatch
= False Then
Ibdatos(0).Caption =
SIVEA.DATANOM.Recordset.nombre.V
alue
Ibdatos(1).Caption =
SIVEA.DATANOM.Recordset.APELLID
OP.Value
Ibdatos(2).Caption =
SIVEA.DATANOM.Recordset.APELLID
OM.Value
Ibdatos(3).Caption =
SIVEA.DATANOM.Recordset.NOCTA.V
alue
On Error Resume Next
LAFOTO.Picture =
LoadPicture(SIVEA.DATANOM.Records
et.foto.Value)
Else

```

```

MsgBox "ERROR EN LA
BUSQUEDA DE EL USUARIO", 64,
"Atención"
End
End If
Else
SIVEA.DATANF.Refresh

SIVEA.DATANF.Recordset.FindFirst
"RFC=" & BUSCACTA & ""
If
SIVEA.DATANF.Recordset.NoMatch =
False Then
Ibdatos(0).Caption =
SIVEA.DATANF.Recordset.nombre.Val
ue
Ibdatos(1).Caption =
SIVEA.DATANF.Recordset.APELLIDOP
.Value
Ibdatos(2).Caption =
SIVEA.DATANF.Recordset.APELLIDO
M.Value
Ibdatos(3).Caption =
SIVEA.DATANF.Recordset.RFC.Value
On Error Resume Next
LAFOTO.Picture =
LoadPicture(SIVEA.DATANF.Recordset
.foto.Value)
Else
MsgBox "ERROR EN LA
BUSQUEDA DE EL USUARIO", 64,
"Atención"
End
End If

End If
End Sub

Private Sub Image1_Click(Index As
Integer)
Call EntraAyuda
End Sub

Private Sub Label2_Click()
Call EntraAyuda
End Sub

Private Sub MDIForm_Activate()

SIVEA.MousePointer = 0

If SIVEA.MNUMUSICA.Checked =
True Then
On Error Resume Next

SIVEA.ActiveForm.OLEM.DoVerb(0)
Else
On Error Resume Next
SIVEA.ActiveForm.OLEM.Action
= 9
End If

End Sub
Sub Avisame()
Dim MIFECHA As Date
Dim DHOY As Date
Dim MENSA As String
MIFECHA = #12/29/00#
DHOY = Date

If DHOY <= MIFECHA Then

```

```

MENSA = "ESTA LICENCIA VENCE
EL DIA 29 DE DICIEMBRE DEL 2000"
MsgBox (MENSA)

Else
MENSA = "ESTA LICENCIA ESTA
VENCIDA PARA ADQUIRIR " + Chr(13)
MENSA = MENSA + "UNA NUEVA
CONTACTE AL PROVEEDOR DEL
SOFTWARE"
MsgBox (MENSA)
End
End If

End Sub
Sub ENTRANDO()

FRMACCESO.PICENT.Visible =
True

NumMus = 4
With FRMACCESO.PGB1
.Value = 0
'sonido DE AVISO en caso de
estar activado
OLEP.CreateEmbed UNIDAD +
"SIVEA\SISTEMA\UTILES\SONIDOS\SI
NG WAV"
OLEP.Refresh

.Value = 5
OLEM(0).CreateEmbed UNIDAD +
"sivea\SISTEMA\UTILES\SONIDOS\sacc
i wav"
OLEM(0).Refresh

.Value = 10
Load OLEM(1)
OLEM(1).Enabled = True
OLEM(1).CreateEmbed UNIDAD +
"sivea\SISTEMA\UTILES\SONIDOS\Intr
o wav"
OLEM(1).Refresh
.Value = 15

If FRMACCESO.APB1(1).Value =
True Then
End
End If

Load OLEM(2)
OLEM(2).Enabled = True
On Error Resume Next
OLEM(2).CreateEmbed UNIDAD +
"sivea\SISTEMA\UTILES\SONIDOS\SIV
S2 wav"
OLEM(2).Refresh

If FRMACCESO.APB1(1).Value =
True Then
End
End If
.Value = 20
Load OLEM(3)
OLEM(3).Enabled = True
On Error Resume Next
OLEM(3).CreateEmbed UNIDAD +
"sivea\SISTEMA\UTILES\SONIDOS\SIV
S3 wav"
OLEM(3).Refresh

If FRMACCESO.APB1(1).Value =
True Then

```

<pre> End End If Value = 30 Load OLEM(4) OLEM(4).Enabled = True On Error Resume Next OLEM(4).CreateEmbed UNIDAD + "sivea\SISTEMA\UTILES\SONIDOS\SV EAPRIN.wav" OLEM(4).Refresh Value = 40 Load OLEM(5) OLEM(5).Enabled = True On Error Resume Next OLEM(5).CreateEmbed UNIDAD + "sivea\SISTEMA\UTILES\SONIDOS\SV S7.wav" OLEM(5).Refresh Value = 50 Load OLEM(6) OLEM(6).Enabled = True On Error Resume Next OLEM(6).CreateEmbed UNIDAD + "sivea\SISTEMA\UTILES\SONIDOS\SV S1.WAV" OLEM(6).Refresh Value = 60 Load OLEM(7) OLEM(7).Enabled = True On Error Resume Next OLEM(7).CreateEmbed UNIDAD + "sivea\SISTEMA\UTILES\SONIDOS\SV S4.wav" OLEM(7).Refresh Value = 70 Load OLEM(8) OLEM(8).Enabled = True On Error Resume Next OLEM(8).CreateEmbed UNIDAD + "sivea\SISTEMA\UTILES\SONIDOS\SV e.wav" OLEM(8).Refresh Value = 80 Load OLEM(9) OLEM(9).Enabled = True On Error Resume Next OLEM(9).CreateEmbed UNIDAD + "sivea\SISTEMA\UTILES\SONIDOS\SV S5.wav" OLEM(9).Refresh Value = 90 Load OLEM(10) OLEM(10).Enabled = True On Error Resume Next OLEM(10).CreateEmbed UNIDAD + "sivea\SISTEMA\UTILES\SONIDOS\SV 1.wav" OLEM(10).Refresh Value = 100 End With End Sub Private Sub MDIForm_Load() If vez = 1 Then Call ENTRANDO </pre>	<pre> vez = 0 SIVEA.WindowState = 2 End If frmmenu.ssbots.Tab = 0 MusicaON = 1 If Acceso = "ST" Then mngob.Enabled = False Else mngob.Enabled = True End If Call LLENADATOS ventana = 0 'Set nodopri = SIVEA.VentanaRaiz.Nodes.Add(, , "R", "ENEP ARAGON") 'nodopri.EnsureVisible DIRECPRI = UNIDAD & "SIVEA\SISTEMA" 'FRMENTRA.OLEM Action = 9 FRMENEP.Show Unload FRMACCESO End Sub Private Sub SSSCONTROL_Click(Index As Integer) End End Sub Private Sub SSCPRINCIPAL_Click(Index As Integer) End Sub Private Sub MDIForm_Resize() If SIVEA.WindowState = 1 Then If mnuarraH.Checked = True Then frmmenu.Hide End If Else If mnuarraH.Checked = True Then frmmenu.Show End If End If Call Avisame End Sub Private Sub MDIForm_Unload(Cancel As Integer) End End Sub Private Sub mnuacercade_Click() frmacerca.Show End Sub Private Sub mnuatge_Click() SIVEA.ActiveForm.MousePointer = 11 Call FRMAULAS.PONAULAS(0) Call FRMAULAS.PasaAulas(1, 2) </pre>	<pre> Call frm salon.INISALON SIVEA.ActiveForm.MousePointer = 0 End Sub Private Sub mnualum_Click() FRMGGOBIERNO.Show Call FRMGGOBIERNO.Acomoda(0) End Sub Private Sub mnuarit_Click() Call FRMAULAS.PONAULAS(0) Call FRMAULAS.PasaAulas(1, 0) Call frm salon.INISALON End Sub Private Sub mnuayu_Click() Call EntraAyuda End Sub Private Sub mnuarraH_Click() If mnuarraH.Checked = False Then frmmenu.Show mnuarraH.Checked = True Else frmmenu.Hide mnuarraH.Checked = False End If End Sub Private Sub mnubiblio_Click() FRMBIBLIO.Show End Sub Private Sub mnucauldif_Click() Call FRMAULAS.PONAULAS(0) Call FRMAULAS.PasaAulas(1, 5) Call frm salon.INISALON End Sub Private Sub mnucaLEN_Click() 'Carga el CALENDARIO en SIVEA Call LLAMAFECHA End Sub Private Sub mnucaLint_Click() Call FRMAULAS.PONAULAS(0) Call FRMAULAS.PasaAulas(1, 6) Call frm salon.INISALON End Sub Private Sub mnucaLumtis_Click() FRMLISTAI UMNO.Show End Sub Private Sub mnucaLummanto_Click() frmDATOSG.Show End Sub Private Sub mnucaLumfanto_Click() frmDATOSI.Show End Sub Private Sub mnucaLumfacilis_Click() FRMLISTAFACIL.Show End Sub </pre>
---	---	--

Private Sub mnucine_Click() Call FRMAULAS.PONAUULAS(1) Call FRMAULAS.PasaAulas(1, 2) Call frm salon.INISALON End Sub	If MNUMUSICA Checked = True Then MNUMUSICA.Checked = False MusicaON = 0 Call TocaMus(0)	End Sub Private Sub mnprac_Click() frm labs.Show Call frm labs.AccionaPan Call frm labs.LLENALISTA(2) End Sub
Private Sub mnudlabs_Click() frm LABS1 Show End Sub	Else MNUMUSICA Checked = True MusicaON = 1 On Error Resume Next	Private Sub mnprog_Click() frm labs Show Call frm labs.AccionaPan Call frm labs.LLENALISTA(3) End Sub
Private Sub mnudina_Click() Call FRMAULAS.PONAUULAS(1) Call FRMAULAS.PasaAulas(1, 2) Call frm salon.INISALON End Sub	If OLEP AppIsRunning = True Then OLEP.Action = 9 End If	Private Sub mnupupitre_Click() Carga el pupitre en SIVEA Err.Clear On Error Resume Next ARCHIVO = DIRECPRI + "PROGRAMAS\PUPITRE.EXE" ValRet = Shell(ARCHIVO, 1) If Err.Number <> 0 Then MsgBox ("A ocurrido un error al tratar de abrir el pupitre") Err Clear End If
Private Sub mnudocu_Click() frm labs Show Call frm labs.AccionaPan Call frm labs.LLENALISTA(1) End Sub	SIVEA.ActiveForm.OLEM.Refresh SIVEA.ActiveForm.OLEM.DoVerb (0) 'Call QueVentana	End Sub
Private Sub mnuelec_Click() Call FRMAULAS.PONAUULAS(1) Call FRMAULAS.PasaAulas(1, 5) Call frm salon.INISALON End Sub	End If End Sub Sub QueVentana() Select Case ventana Case 0 On Error Resume Next OLET = UNIDAD + "sivea\SISTEMA\UTILES\SONIDOS\SIV EAPRIIN.wav" Case 1 'FRMAULAS OLET = UNIDAD + "sivea\SISTEMA\UTILES\SONIDOS\SIV S5.wav" Case 2 'FRMBIBLIO OLET = UNIDAD + "sivea\SISTEMA\UTILES\SONIDOS\SIV S3.wav" Case 3 OLET = UNIDAD + "sivea\SISTEMA\UTILES\SONIDOS\SIV S4.wav" Case 4 'FRMGOBIERNO OLET = UNIDAD + "sivea\SISTEMA\UTILES\SONIDOS\SIV S1.wav" Case 5 'frmsalon	End Sub
Private Sub mnufaciligencia_Click() FRMCLASES.Show End Sub	Case 1u0 Case 200 Case 300 End Select	Private Sub mnuter_Click() Call FRMAULAS.PONAUULAS(1) Call FRMAULAS.PasaAulas(1, 4) Call frm salon.INISALON End Sub
Private Sub mnuest_Click() Call FRMAULAS.PONAUULAS(1) Call FRMAULAS.PasaAulas(1, 0) Call frm salon.INISALON End Sub	End Sub Private Sub mnuso_Click() frm usuarios Show End Sub	Private Sub mnuter_Click() Call FRMAULAS.PONAUULAS(1) Call FRMAULAS.PasaAulas(1, 3) Call frm salon.INISALON End Sub
Private Sub mnufaciligeneval_Click() Call Egeneval(4) End Sub	Private Sub mnuso_Click() frm usuarios Show End Sub	Private Sub mnuserpan_Click() If mnuserpan.Checked = True Then mnuserpan.Checked = False picpan.Width = 10 Else mnuserpan.Checked = True picpan.Width = 2000 End If End Sub
Private Sub mnufacillyt_Click() frm Tarea Show End Sub	Private Sub mnuso_Click() frm usuarios Show End Sub	Private Sub Termina_Click() Call FinSesion End Sub
Private Sub mnugeo_Click() Call FRMAULAS.PONAUULAS(0) Call FRMAULAS.PasaAulas(1, 4) Call frm salon.INISALON End Sub	Private Sub mnuso_Click() frm usuarios Show End Sub	Private Sub Timer1_Timer() If vez = 0 Then Xprin = Xprin + 1 If Xprin = 10 Then If MusicaON = 1 Then Call TocaMus(NumMus) End If Xprin = 0 End If End If
Private Sub mnugeo1_Click() Call FRMAULAS.PONAUULAS(0) Call FRMAULAS.PasaAulas(1, 1) Call frm salon.INISALON End Sub	Private Sub mnuso_Click() frm usuarios Show End Sub	
Private Sub mnnumagn_Click() Call FRMAULAS.PONAUULAS(1) Call FRMAULAS.PasaAulas(1, 6) Call frm salon.INISALON End Sub	Private Sub mnuso_Click() frm usuarios Show End Sub	
Private Sub MNUMUSICA_Click() Dim X As String	Private Sub mnuso_Click() Call FRMAULAS.PONAUULAS(1) Call FRMAULAS.PasaAulas(1, 3) Call frm salon.INISALON	

<pre> End Sub Private Sub Toolbar2_ButtonClick(ByVal Button As Button) Select Case Button.Key Case "mnu1" Call barraOnTop frmmenu.ssbots.Tab = 0 Case "mnu2" pupitre.Show End Select End Sub </pre>	<pre> Global dibIn As Integer Global mano As Integer Global pelicula As Integer Global Hazlo As Integer Global Asistente As Integer Global TTema As String Option Explicit Global DispCalc As Object Global nivelD As Integer </pre>	<pre> 'Ejecuta a la calculadora On Error GoTo errhandle X = Shell("c:\SIVEA\SISTEMA\PROGRAMA SIEEXEC\calculadora.exe ", 1) ' Empezar calculadora X = DoEvents() ' esto da a la calculadora el paso Exit Sub errhandle: MsgBox "Object Not Created " Exit Sub </pre>
<p>MODULO1 TESIS.GLO</p>		
<pre> DefInt A-Z #If Win32 Then 'Declaraciones de 32 bits Declare Sub SetWindowPos Lib "User32" (ByVal hWnd As Long, ByVal hWndInsertAfter As Long, ByVal X As Long, ByVal Y As Long, ByVal cx As Long, ByVal cy As Long, ByVal wFlags As Long) #Else Declare Sub SetWindowPos Lib "USER" (ByVal h1%, ByVal h2%, ByVal X%, ByVal Y%, ByVal cx%, ByVal cy%, ByVal f%) #End If Declare Function WinHelp Lib "USER" (ByVal hWnd As Integer, ByVal Helpfiles As String, ByVal Wcommand As Integer, ByVal dwData As Any) As Integer Global Const HWND_TOPMOST = -1 Global Const HWND_NOTOPMOST = - 2 Global Const SWP_NOACTIVATE = &H10 Global Const SWP_SHOWWINDOW = &H40 Global Const plata = &HC0C0C0 Global Const oro = &H25BFFA Global Const Negro = &H0& Global Const cafe = &H4080& Global Const rojo = &HFF& Global Const naranja = &HA0FF& Global Const Amarillo = &HFFFF& Global Const Verde = &HC000& Global Const Azul = &HFF000& Global Const Violeta = &H800080 Global Const Gris = &H808080 Global Const Blanco = &HFFFFFF Global Const sincolor = &HC0E0FF Global Const ColorHoja = &HBDF8FB Global Const DIRECTORIOBASES = "C:\SIVEA\INSTITUCION\GOBIERNO\R EGISTROS\BDS\SIVEA.MDB" Global ventana As Integer Global nodopri As Node Global NOCUENTA As String Global Tema As String Global SubTema As Integer Global color1 As Double Global compo As Integer </pre>	<pre> Global TemaSele As Integer Global MateSele As Integer Global CtaClase As Integer Global UNIDAD As String Global Direc1 As String Global Arretema(99) As String Global fechaf As Date Global valorf As Integer Global Usuario As String Global acciusu As String Global IdeN As String Global Acceso As String Global ENTRACAN As Integer Global MusicaON As Integer Global UnidadAyu As String Global vez As Integer Global NumMus As Integer Sub TocaMus(i As Integer) Dim J As Integer If SIVEA.OLEM(i).ApplsRunning = False Then '10 son los sonidos temporalmente For J = 0 To 10 On Error Resume Next SIVEA.OLEM(J).Action = 9 Next J NumMus = 1 On Error Resume Next SIVEA.OLEM(i).DoVerb (0) End If End Sub Sub EntraAyuda() SIVEA.CommonDialog1.HelpFile = UnidadAyu + "sivea\sistema\ayuda\ayuda.hlp" SIVEA.CommonDialog1.HelpCommand = cdlHelpContents SIVEA.CommonDialog1.ShowHelp Presentar el contenido de la Ayuda de Visual Basic. End Sub Sub ENTRAUSUARIO(quien As String) Usuario = quien frmusuarios.Show End Sub Sub calculadora() Dim X As Integer </pre>	<pre> MsgBox "Ejecuta graficadora On Error GoTo errhandl X = Shell("c:\SIVEA\SISTEMA\PROGRAMA SIEEXEC\grafica.exe ", 1) ' Empezar graficadora X = DoEvents() Exit Sub errhandl: MsgBox "error No se encontro el archivo" Exit Sub End Sub Sub LLAMAFECHA() frmagenda.Show End Sub Sub ponfecha() Call LLAMAFECHA End Sub Sub Main() vez = 1 Xprin = 0 FRMENTRA.Show FRMACCESO.Show End Sub Sub momenu(que As String) If SIVEA.mnubarrari.Checked = True Then If que = "H" Then frmmenu.Hide SIVEA.mnubarraH.Checked = False Else frmmenu.Show SIVEA.mnubarraH.Checked = True End If Else frmmenu.Show SIVEA.mnubarraH.Checked = True End If End Sub </pre>

<pre> Sub Egeneral(Paso As Integer) Dim ValRet As Integer Dim Conf, rutaG As String Dim X As Variant Dim confi As String UNIDAD = "C:\\" Direc1 = "SIVEA\SISTEMA\PROGRAMAS\IGENE RAE\\" confi = UNIDAD + Direc1 + "Configura.idr" On Error Resume Next Kill confi Open confi For Output As #1 ' Abre el archivo para operaciones de salida. Print #1, Paso Print #1, UNIDAD Close #1 'On Error GoTo errhandl 'GENERAL ValRet = rutaG = UNIDAD + Direc1 + "general.EXE" X = Shell(rutaG, vbMaximizedFocus) X = Shell(rutaG, vbNormalFocus) ' Exit Sub 'errhandl ' MsgBox "error No se encontro el archivo" ' Exit Sub End Sub Function buscaArchivo(obj As Object) As String Dim X As String Dim direcole As String Dim MENSA As String obj.CommonDialog1.Filter = (Todos los archivos (*.*)" obj.CommonDialog1.DefaultExt = "(documentos de texto(*.doc;*.txt))" MENSA = "" doc;*.txt" obj.CommonDialog1.filename = MENSA direcole = "c:\\" ChDir direcole obj.CommonDialog1.InitDir = direcole obj.CommonDialog1.ShowOpen If obj.CommonDialog1.CancelError = False Then Do 'x = Dir(CommonDialog1.filename) X = obj.CommonDialog1.filename If (X = "") Or (X = MENSA) Then </pre>	<pre> MsgBox ("error ,vuleva a intentarlo") obj.CommonDialog1.ShowOpen End If Loop Until X <> "" buscaArchivo = X End If End Function Function Buscalma(obj As Object, ia As Integer) As String Dim X As String Dim direcole As String Dim MENSA As String If ia = : Then obj.CommonDialog1.Filter = "(Todos los archivos de programas)" obj.CommonDialog1.DefaultExt = "(Archivos de programas(*.exe;*.com)" MENSA = "" exe;*.com" Else obj.CommonDialog1.Filter = "(Todos los archivos de imagenes)" obj.CommonDialog1.DefaultExt = "(Archivos de imagenes(*.bmp;*.pcx)" MENSA = "" bmp;*.pcx" End If obj.CommonDialog1.filename = MENSA direcole = UNIDAD On Error Resume Next ChDir direcole obj.CommonDialog1.InitDir = direcole obj.CommonDialog1.ShowOpen If obj.CommonDialog1.CancelError = False Then Do X = obj.CommonDialog1.filename If (X = MENSA) Then MsgBox ("error ,vuleva a intentarlo") obj.CommonDialog1.ShowOpen End If Loop Until X <> "" If (X = "") Then X = UNIDAD + "sivea\sistema\utiles\bmps\unamico.bm p" End If Buscalma = X End If End Function Sub FinSesion() On Error Resume Next SetWindowPos FRMSALE hWnd, HWNK_TOPMOST, FRMSALE.Left / 15, _ FRMSALE.Top / 15, FRMSALE.Width / 15, _ FRMSALE.Height / 15, SWP_NOACTIVATE Or SWP_SHOWWINDOW </pre>	<pre> End Sub Sub ADONDEVOY(Index As Integer) SIVEA.ActiveForm.MousePointer = 11 Select Case Index Case 0 ventana = 1 FRMAULAS.Show frmmenu.ssbots.Tab = 3 Case 1 ventana = 3 frmtabs.Show frmmenu.ssbots.Tab = 2 Case 2 ventana = 2 FRMBIBLIO.Show frmmenu.ssbots.Tab = 1 Case 3 ventana = 4 FRMGOBIERNO.Show frmmenu.ssbots.Tab = 5 Case 4 If SIVEA.mnubarraH.Checked = True Then frmmenu.Hide SIVEA.mnubarraH.Checked = False End If FRMENEP.oie2.DoVerb (0) End Select If Index < 4 Then Unload FRMENEP End If SIVEA.ActiveForm.MousePointer = 0 End Sub </pre> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>MODULO2. CONSTANTES Y VARIABLES</p> <pre> ' General Global ARREMUS(0 To 50) Global MUSACT As Object ' Arrange Method ' for MDI Forms Global Const CASCADE = 0 Global Const TILE_HORIZONTAL = 1 Global Const TILE_VERTICAL = 2 Global Const ARRANGE_ICONS = 3 ' Properties ' Colors Global Const BLACK = &H0& Global Const RED = &HFF& Global Const GREEN = &HFF00& Global Const YELLOW = &HFFFF& Global Const BLUE = &HFF0000 Global Const MAGENTA = &HFF00FF Global Const CYAN = &HFFFFFF00 Global Const WHITE = &HFFFFFF& </pre> </div>
--	---	--

```
' System Colors
Global Const SCROLL_BARS =
&H80000000 ' Scroll-bars gray
area
Global Const DESKTOP = &H80000001
' Desktop.
Global Const ACTIVE_TITLE_BAR =
&H80000002 ' Active window
caption.
Global Const INACTIVE_TITLE_BAR =
&H80000003 ' Inactive window
caption.
Global Const MENU_BAR =
&H80000004 ' Menu
background
Global Const
WINDOW_BACKGROUND =
&H80000005 ' Window background.
Global Const WINDOW_FRAME =
&H80000006 ' Window frame.
Global Const MENU_TEXT =
&H80000007 ' Text in menus.
Global Const WINDOW_TEXT =
&H80000008 ' Text in windows.
Global Const TITLE_BAR_TEXT =
&H80000009 ' Text in caption, size
box, scroll-bar arrow box..
Global Const ACTIVE_BORDER =
&H8000000A ' Active window
border
Global Const INACTIVE_BORDER =
&H8000000B ' Inactive window
border
Global Const
APPLICATION_WORKSPACE =
&H8000000C ' Background color of
multiple document interface (MDI)
applications.
Global Const HIGHLIGHT =
&H8000000D ' Items selected
item in a control.
Global Const HIGHLIGHT_TEXT =
&H8000000E ' Text of item selected
in a control

Global Const BUTTON_FACE =
&H8000000F ' Face shading on
command buttons
Global Const BUTTON_SHADOW =
&H80000010 ' Edge shading on
command buttons
Global Const GRAY_TEXT =
&H80000011 ' Grayed (disabled)
text This color is set to 0 if the current
display driver does not support a solid
gray color.
Global Const BUTTON_TEXT =
&H80000012 ' Text on push
buttons.

' Enumerated Types

' MsgBox parameters
Global Const MB_OK = 0 ' OK
button only
Global Const MB_OKCANCEL = 1
' OK and Cancel buttons
Global Const
MB_ABORTRETRYIGNORE = 2 '
Abort, Retry, and Ignore buttons
Global Const MB_YESNOCANCEL = 3
' Yes, No, and Cancel buttons
Global Const MB_YESNO = 4
' Yes and No buttons
Global Const MB_RETRYCANCEL = 5
' Retry and Cancel buttons

Global Const MB_ICONSTOP = 16
' Critical message
Global Const MB_ICONQUESTION =
32 ' Warning query
Global Const MB_ICONEXCLAMATION
= 48 ' Warning message
Global Const MB_ICONINFORMATION
= 64 ' Information message

Global Const MB_APPLMODAL = 0
' Application Modal Message Box
Global Const MB_DEFBUTTON1 = 0
' First button is default

Global Const MB_DEFBUTTON2 = 256
' Second button is default
Global Const MB_DEFBUTTON3 = 512
' Third button is default
Global Const MB_SYSTEMMODAL =
4096 ' System Modal

' MsgBox return values
Global Const IDOK = 1 ' OK
button pressed
Global Const IDCANCEL = 2
' Cancel button pressed
Global Const IDABORT = 3
' Abort button pressed
Global Const IDRETRY = 4
' Retry button pressed
Global Const IDIGNORE = 5
' Ignore button pressed
Global Const IDYES = 6 ' Yes
button pressed
Global Const IDNO = 7 ' No
button pressed

-----
----- variables -----
-----

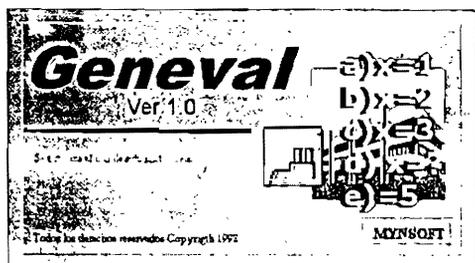
Dim ECUACION As String
Global qhacer As String
Global TiManual(499) As Integer
Global PagxVer(99) As Integer
Global DIRECPRI As String
Global ARCHIVO As String

Global Xprn As Integer
```

ANEXO E

AYUDA DE GENEVAL.

BIENVENIDO AL SISTEMA DE GENERADOR DE EVALUACIONES



Este documento le permitirá conocer los por menores del diseño y generación de evaluaciones profesionales.

CONTENIDO

El presente archivo describe de forma general la manera de utilizar el generador de evaluaciones GENEVAL para realizar evaluaciones profesionales en cuestión de minutos.

QUE ES GENEVAL

El GENERador de EVALuaciones GENEVAL surge como una necesidad de disponer de un sistema confiable y sistematizado para generar evaluaciones en diversos temas de manera que se pueda tener un banco de reactivos centralizados

GENEVAL permite la creación de evaluaciones del tipo de preguntas de incisos o de opción múltiple como mejor se le conoce, permitiendo diseñar dos tipos diferentes de reactivos: Preguntas cerradas con una sola respuesta y ejercicios de aplicación (para mas detalle de estos tipos consulte el glosario de términos al final de este archivo). GENEVAL permite generar hasta cinco diferentes versiones de evaluación con un limite de reactivos que se tengan disponibles en el banco.

EL ENTORNO

El sistema corre bajo el entorno de WINDOWS 95, si cuenta con una versión menor a esta el sistema no funcionara.

Este entorno es un ambiente amigable y presenta la información por medio de gráficos. Su funcionamiento se basa principalmente en el uso de tres conceptos fundamentales: Ventanas, iconos y botones.

En las ventanas se muestra toda la información referente a un tópico o tema y permite a su vez contener controles que van a realizar las tareas que el usuario desee y que el sistema

permita. Estos controles pueden ser cuadros de dialogo, imágenes, botones, iconos, cajas simples y combinadas, etiquetas, listas, etc.

Los iconos son imágenes o gráficos que ejecutan una tarea una vez que son seleccionados. Su ejecución puede ser por medio de un clic o un doble clic.

Los botones emiten una orden de instrucción al sistema y le indican cual será la acción a realizar es muy importante presionar el botón que mejor describa la acción que desea realizar ya que de lo contrario puede borrar o dañar información que sea importante para el sistema.

REQUERIMIENTOS

Los requerimientos para el sistema son:

- ◆ Una computadora personal con procesador 486, Pentium o superior.
- ◆ Disco duro con al menos 10 MB de espacio disponible.
- ◆ Monitor SVGA color o monocromático (se recomienda una resolución de 800 x 600 pixeles ya que si usted corre a una resolución menor por ej. 640 x 480 no se visualizaran todos los componentes de las ventanas)
- ◆ Coprocesador matemático (Mejora la realización de las operaciones que realiza el sistema)
- ◆ Mouse y teclado

INSTALACION

La forma de instalar GENEVAL es muy simple solo se deberá seguir los siguientes pasos

1. Acceder al administrador de archivos (Explorador de WINDOWS).
2. Buscar el directorio de nombre "SIVEA" y acceder a él.
3. Ahí encontrara un icono con el nombre de "INSTALACION DE GENEVAL" hacer doble clic en él
4. Seguir las instrucciones del instalador (SETUP)

NOTA : Si se siguen correctamente los pasos para la instalación, GENEVAL se instalará en el directorio seleccionado por ejemplo: C:\ARCHIVOS DE PROGRAMAS\GENEVAL.

LO BASICO

Para acceder al programa solo hay que buscar el icono de programa que contiene el archivo ejecutable, es decir buscar el icono de nombre GENEVAL; este se encuentra localizado en el directorio que se selecciono para su instalación (por ejemplo: C:\ARCHIVOS DE PROGRAMAS\GENEVAL). Hacer doble clic en el icono o seleccionarlo y dar ENTER (El tipo de acceso puede variar pero no debe tardar mas de unos cuantos segundos).

Fig. 1.2 .Pantalla típica de GENEVAL.

Controles

En el sistema vera usted los siguientes controles:

1. Barra de menús.
2. Barra de herramientas.
3. Barra de curso.
4. Botones de "Ayuda " y "Acerca de..."
5. Área de trabajo.
6. Barra de mensajes

1.-Barra de Menús

En esta barra usted encontrara seis menús: Principal, Evaluaciones, Edición Ver, Herramientas y Ayuda.

En el menú principal usted contará con las siguientes opciones:



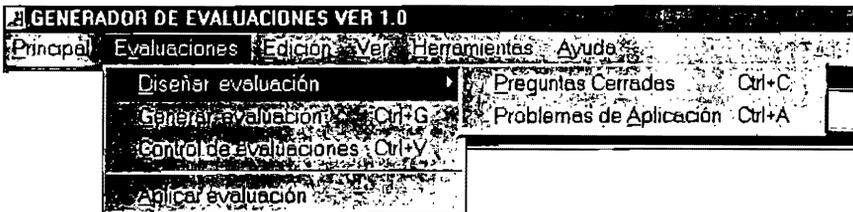
Cursos : Cuenta con dos opciones que son:

Catálogo (CTRL+T): Permite abrir la ventana de catálogo de cursos, ahí puede seleccionar el curso con el cual desea trabajar.

Control de cursos (CTRL+U): Abre la ventana de Control de cursos en ella puede dar de alta, editar, modificar y seleccionar el curso que desee.

Terminar (CTRL+S): Salir del sistema.

Dentro del Menú de Evaluaciones se tienen las siguientes opciones:



Diseñar evaluación: Esta opción a su vez cuenta con dos alternativas diferentes:

Preguntas Cerradas (CTRL + C): Crear reactivos de incisos donde se tiene una pregunta y cinco posibles respuestas.

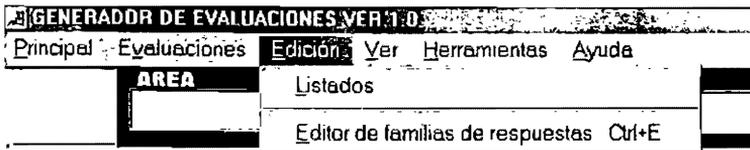
Problemas de aplicación (CTRL + A): Diseñar reactivos que presentan un problema cuya solución esta dado por una ecuación o formula.

Generar evaluación (CTRL+G): Le da acceso a la ventana para diseñar evaluaciones de forma manual.

Control de evaluación (CTRL+V): Permite el acceso a la ventana de control de evaluaciones donde se puede aplicar, eliminar o crear una evaluación.

Aplicar evaluación: Muestra la ventana de evaluaciones disponibles donde se puede seleccionar una para su aplicación.

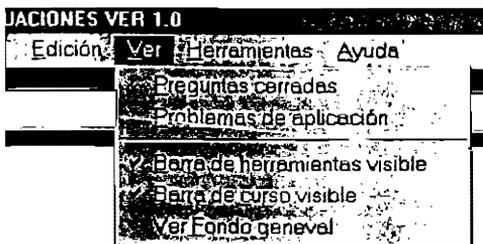
En el menú de edición podrá observar lo siguiente:



Listados: Esta opción abre la ventana de listados donde se pueden imprimir en pantalla, archivo u hoja de papel los datos contenidos en las tablas de datos

Editor de familia de respuestas (CTRL+E) le permite abrir la ventana del editor de familia de respuestas.

El menú de Ver cuenta con las siguientes opciones:



Preguntas cerradas: Permite entrar al módulo de preguntas cerradas solo para consultas, es decir no se permite la creación, modificación o eliminación de reactivos.

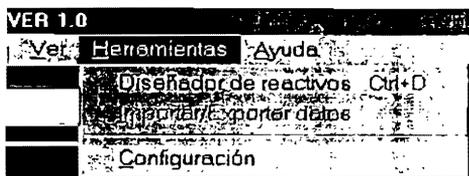
Problemas de aplicación: Tiene la misma función que la opción anterior, solo que lo realiza en el módulo de problemas de aplicación.

Barra de herramientas visible: Si se encuentra marcada con una paloma la barra de herramientas (marcada con 2 en la ilustración 1.2) será visible de lo contrario esta estará oculta.

Barra de cursos visible: Si se encuentra marcada con una paloma la barra de curso visible (marcada con 3 en la ilustración 1.2) será visible de lo contrario esta estará oculta.

Ver fondo general: Dibuja en el fondo de la ventana el logo del sistema (indicado con un 5 en la figura 1.2)

En el menú de herramientas se cuenta con las siguientes alternativas:



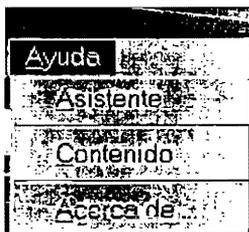
Diseñador de reactivos (CTRL+D): Permite acceder a la ventana de selección de diseño de reactivos

Importar / Exportar datos: Permite pasar datos de una unidad determinada a otra.

Configuración: Permite establecer los parámetros de acceso al sistema ,así como configurar datos del sistema

En el menú de ayuda.

Se cuenta con las opciones de:



Asistente: Abre el asistente de generador de exámenes que permite crear evaluaciones profesionales en solo ocho pasos.

Contenido: Muestra este archivo.

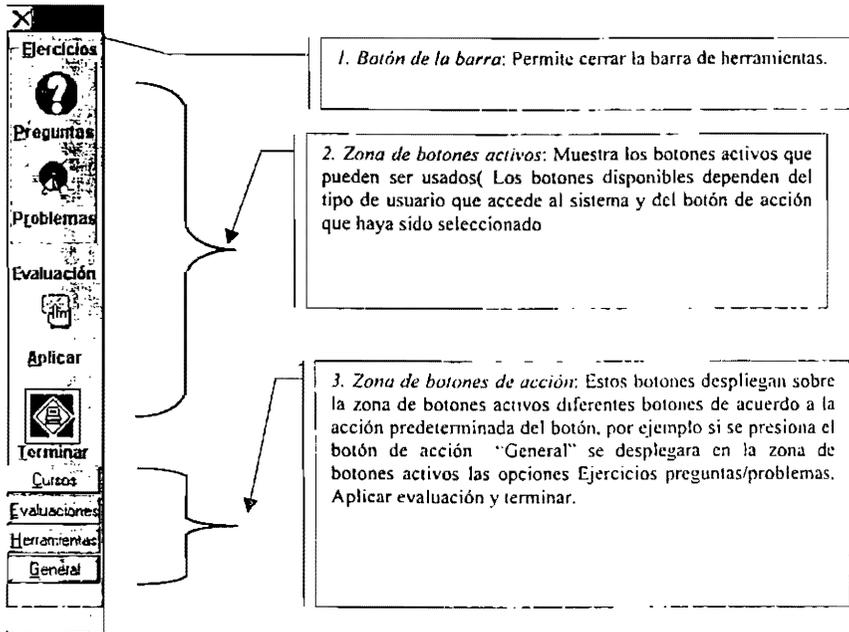
Acerca de... : Muestra una pantalla de créditos del sistema.

SALIR DEL SISTEMA

Para salir del sistema haga clic en el botón de TERMINAR en la barra de herramientas o Seleccione TERMINAR en el Menú principal. También puede presionar CTRL + S en cualquier pantalla para finalizar la sesión de trabajo.

2. Barra de herramientas.

En esta barra se muestran los botones que activan las vistas que su nombre indican, estos botones funcionan igual que si seleccionara la opción desde la barra de menús. Esta barra se compone de tres partes:



Botón de acción cursos: Muestra las opciones de Catálogo de cursos y control de cursos.

Botón de acción Evaluaciones: Muestra las opciones de

Botón de Herramientas : Presenta las opciones

Botón de acción General: Están disponibles las opciones de Ejercicios preguntas/problemas y Aplicar evaluación.

Nota: En todos los botones de acción aparece la opción terminar cuya función es la de salir del sistema.

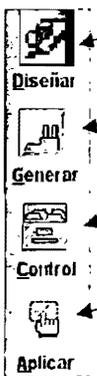
Cursos



Abre la ventana de catálogo de cursos.

Abre la ventana de control de cursos, para crear, editar, eliminar y seleccionar cursos..

Evaluaciones



Abre la ventana de diseño de reactivos, donde se puede seleccionar el tipo de reactivos a crear, editar o eliminar.

Abre la ventana de generar evaluaciones en donde se pueden crear las evaluaciones de manera manual.

Abre la ventana de control de cursos, para crear, editar, eliminar y seleccionar cursos..

Abre la ventana de realización de evaluaciones via computadora, donde se pueden aplicar evaluaciones y ser calificadas en el momento que se realiza y termina.

Herramientas



Despliega una ventana en la que puede seleccionar el tipo de listado que desea ver , ya sea en pantalla o impresora.

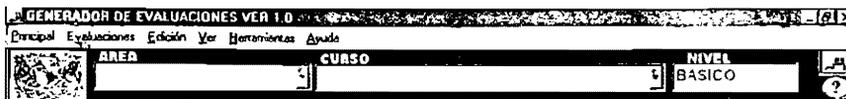
Acceso al asistente para generar evaluaciones, que permite crear evaluaciones profesionales en ocho sencillos pasos.



Terminar una sesión de trabajo; sale del programa.

3. Barra de curso.

En esta barra se muestra el logo de la institución, así como los siguientes datos



1 Área del curso. Se tienen cinco áreas de alcance en los cursos técnicos administrativos y humanistas, si usted por ejemplo crea un curso de mecánica, lo más adecuado sería asignarlo al área técnica mecánica, pero si usted lo crea de liderazgo este curso debería pertenecer al área humanista

2 Curso El nombre con el cual se identifica el curso y es el identificador para cada grupo de reactivos de un tema en específico

3 Nivel del curso Para darse una idea de lo complicado o sencillo del tema cada curso esta clasificado por: básico, medio o avanzado, cuando se crea un curso deberá especificarse la dificultad de este

4. Botones de ayuda y acerca de...

Estos botones se encuentran localizados en la esquina superior derecha de su pantalla y su función es la de mostrar este archivo y mostrar una pantalla de créditos respectivamente

5. Área de trabajo.

En esta área se despliegan las ventanas que son activadas por el usuario, estas ventanas pueden ser: el catálogo, el editor de preguntas cerradas y el de problemas de aplicación, el editor de cursos, etc

6. Barra de mensajes.

En esta barra vera información acerca de lo que usted ha seleccionado y que acción esta por realizarse.

Puntos básicos para generar una evaluación.

Lo básico para generar una evaluación es lo siguiente.

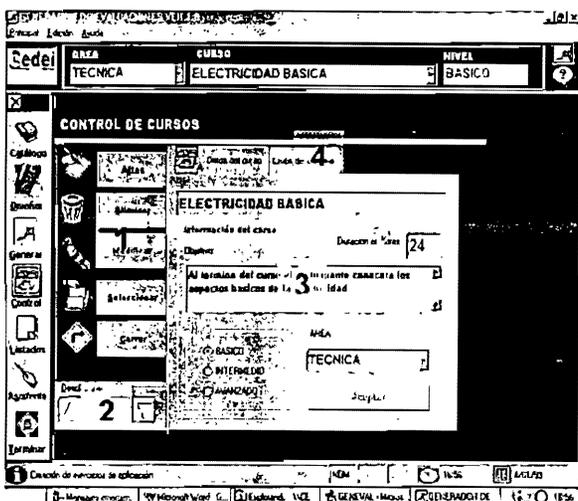
- 1 Debe existir el curso del tema que se desea crear la evaluación (Esto se detalla en como [crear un curso](#))
- 2 Se debe contar con un número de reactivos (ya sean preguntas cerradas o problemas de aplicación, este tema se detalla mas adelante en el tema de [cómo generar reactivos](#))
3. Debe generar la evaluación de manera manual o utilizando el asistente

COMO CREAR UN CURSO

Un curso es un tema en particular del cual se desean generar evaluaciones

Por ejemplo en el sistema está dado de alta el curso de electricidad. Este cuenta con un objetivo particular y un número de horas destinados a la aplicación de este curso (este número de horas puede variar de acuerdo a quien va dirigido el curso)

Para crear, modificar, eliminar o seleccionar un curso deberá acceder al modulo de CONTROL



En el modulo de control se puede observar que esta formado por cuatro áreas

- 1.El área de botones
- 2.El área de botones de desplazamiento de cursos
- 3.El área de despliegue de información del curso
- 4.La lista de cursos disponibles

El área de botones

En esta área se encuentran cinco botones con los cuales usted puede hacer lo siguiente

ALTAS: Crear un nuevo curso - Después de presionar este botón presione el botón de aceptar para dar de alta el curso

NOTA: si el curso ya existe o se teclaea mal algún dato el sistema mandara un mensaje de error pidiendo que vuelva a intentar la operación tecleando correctamente los datos

ELIMINAR: Si usted desea eliminar un curso, por cualquier motivo, debera eliminarlo de la siguiente forma:

1. Seleccione el curso. (Tenga cuidado de no eliminar un curso que no sea el indicado, ya que no hay forma de recuperarlo).
2. Presione el botón de eliminar (Aparecerá un mensaje solicitando que confirme la acción a realizar).
3. Si usted esta seguro de que desea eliminar el curso presione el botón del cuadro de mensaje de "ACEPTAR".

Nota: Cuando se elimina un curso, automáticamente se eliminan todos los reactivos que tenga este curso. Sea cuidadoso al eliminar cursos.

MODIFICAR: Cuando se desea cambiar el nombre de un curso, el número de horas, el nivel, el área o el objetivo, se deberá presionar el botón de "MODIFICAR" En ese momento se activara el área de despliegue de curso (3) y podrá usted cambiar cualquiera de los parámetros antes mencionados. Para aceptar los cambios presione el botón de "aceptar"

Nota: Si usted desea cancelar los cambios presione le botón de cancelar.



Bibliografía consultada:

Ing. ALCANTARA RODRÍGUEZ FERNANDO Fundamentos de Física y matemáticas Editado en la escuela Nacional de estudios profesionales Aragón

APARICI, R. y GARCIA, A (1988) El material didáctico de la UNED Medios Audiovisuales ICE Madrid.

BAATH. (1988). Lista de ideas para la construcción de los cursos de educación a distancia Distance Education. International Perspectives Londres

BARRANTES ECHAVARRÍA, Rodrigo. (1992) Educación a Distancia EUNED San José de Costa Rica.

BOUCHE PERIS, Henri. (1988) El valor de los recursos de apoyo - ilustración y medios audiovisuales en el aprendizaje a distancia de materias filosóficas EUNED Tercer Congreso Iberoamericano de Educación a Distancia. San José de Costa Rica.

MENA, Marta. (1996) La Educación a Distancia en el Sector Público. INAP Buenos Aires

RODINO, Ana M. (1987) Los medios audiovisuales en la enseñanza a distancia y su uso en la enseñanza a distancia. EUNED San José de Costa Rica

DERTOUZOS, Michael L (1997) *Qué Será* Editorial Planeta Buenos Aires

REMESEIRA, Claudio (1997) Turbo capitalismo global Informe especial Revista Apertura Edición N°66. Marzo 1997 Buenos Aires

HOLMBERG, Borje. (1980) the didactics of Distance Education. Course Unit 5 ZIFF Londres

HOLMBERG, Borje. (1983) Students Support Services Introduction to Distance Education International perspectives Londres

Galbreath, J. (1995) Compressed Digital Videoconferencing. Educational Technology. 35(1), 31-38.

Glossary of Terms. (1996)

Reed, J. and Woodruff, M. (1995). Using compressed video for distance learning.

Woodruff, M & Mosby, J. (1996) A brief description of videoconferencing Videoconferencing in the classroom and library

Lochte, R.H. (1993) Interactive television and instruction Englewood Cliffs, NJ Educational Technology Publications.

Oliver, E.L.(1994). Video tools for distance education. In B. Willis (Ed.), Distance education: Strategies and tools (pp. 165-195). Englewood Cliffs, NJ: Educational Technology Publications.

Sponder, B.M. (1990). Distance education in rural Alaska: An overview of teaching and learning practices in audioconference courses. (Monograph Series in Distance Education No. 1). Fairbanks, AK: University of Alaska Center for Cross-Cultural Studies. (ED 325 276)

Wolcott, L.L. (1990). Audio tools for distance education. In B. Willis (Ed.), Distance education: Strategies and tools (pp. 135-164). Englewood Cliffs, NJ: Educational Technology Publications

Ackermann, E. (1996). Tools for teaching: The World Wide Web and a Web Browser

Kochmer, J. (1995). Internet passport: Northwestnet's guide to our world online. Bellevue, WA: NorthWestNet and Northwest Academic Computing Consortium, Inc.

Hughes, K. (1994). Entering the World-Wide Web: A guide to cyberspace. Enterprise Integration Technologies.

Información sobre educación virtual extraída de la investigación realizada por

- **Duván O. Hernández P.**
Ingeniero de Sistemas Escuela de Administración de Negocios - Bogotá, Colombia

- **Dr. Luis Osin .**
Desarrollo y Utilización de Sistemas de Enseñanza Asistida por Computadora

Guía de carreras UNAM 1996.
Universidad Nacional Autónoma de México

La programación moderna Por David Garza
© 1998 Microsoft Corporation
Todos los derechos reservados

Libros en pantalla y ayuda de Microsoft® Visual Basic®

Sistema de programación para Windows® 95 y Windows NT®, versión 5.0 Microsoft, ActiveX, FoxPro, Microsoft QuickBasic, MS, MS-DOS, PivotTable, Rushmore, Visual Basic, Visual C++, Visual SourceSafe, Win32, Windows y Windows NT son marcas comerciales o marcas registradas de Microsoft Corporation en los Estados Unidos y en otros países.

TrueType es una marca registrada de Apple Computer, Inc.

Otros nombres de productos y compañías aquí mencionados pueden ser marcas comerciales de sus respectivos propietarios.

GLOSARIO DE TERMINOS

Alfabetización computacional ("computer literacy"). Conjunto de habilidades, percepciones y conocimientos sobre computación, sus aplicaciones e implicaciones que permiten a una persona funcionar dentro de una sociedad cuya información se encuentra computarizada

Acumulador: Circuitos o registros de almacenamiento temporal del resultado de una operación aritmética o lógica en la computadora.

Administración de la instrucción por computadora : Es el uso de la computadora para administrar los procesos de enseñanza o instruccionales, tales como generar exámenes, corregirlos, crear y mantener la hoja de notas, controlar el uso de "courseware" instruccional, realizar reportes, etc. Se le conoce frecuentemente por sus siglas en Inglés: CMI.

Administración Escolar por Computadora Es el uso administrativo clásico de una escuela empleando a la computadora: nómina, contabilidad, inventario, cuentas por cobrar etc. Generalmente se le agregan a este concepto cuestiones específicas de la administración de una escuela que no tienen que ver con la enseñanza en sí tales como: admisión y registro de alumnos, circulares a padres de familia, control del cobro de matrícula asignación de salones y horarios a grupos etc.

Algoritmo: Conjunto de instrucciones dispuestas en forma de pasos secuenciales (actualmente) de ejecución, que transforman datos e información de entrada, para obtener un resultado en un número finito de pasos

Animación: movimiento de un dibujo o gráfica a través de la pantalla. Se emplea para efectos demostrativos de la evolución de un fenómeno en el tiempo, o simular el desplazamiento de un objeto así como el lograr efectos visuales. Generalmente se logra mediante la programación del dibujo rápido de un objeto gráfico, seguido del borrado de las partes que cambian y el volverlo a dibujar.

Aprendizaje Asistido por Computadora. enfoque de la educación basada en computadora en la cual se hace énfasis en el sujeto que incrementa su conocimiento, sin que a este se le enseñe o muestre explícitamente, el sujeto lo tiene que inferir o apropiárselo por sí mismo, empleando para ello algunos programas didácticos de computadora. El proceso ya no es de inculcación de información sino de capturar conocimiento. Las técnicas de descubrimiento y la programación en LOGO se señalan como buenos exponentes de este enfoque. Sin embargo este énfasis puede darse a cualquier modalidad de la Enseñanza por Computadora, por lo que se le toma como sinónimo aunque en el fondo no lo sea

Archivo ("file"): Información (o conjunto de datos estructurados) homogénea almacenada en un disco o diskette e identificados por un nombre. Está formado por un conjunto de registros ("records") que dan la información acerca de un individuo y están etiquetadas. Una base de datos consiste de un grupo estructurado y relacionado de archivos.

Banco de Datos: es el contenido como información de una base de datos

Barrido ("scrolling"): Es el movimiento de las líneas en una pantalla de tal manera que la línea superior desaparece cuando entra una línea en la parte inferior.

Base de datos: Es la organización y forma de manejo de una colección de datos (banco de datos) interrelacionados, estructurados y almacenados en disco (en archivos manejados de manera transparente para el usuario), dentro de un conjunto sin redundancias perjudiciales e innecesarias para su consulta y actualización. Frecuentemente bajo este nombre se quiere dar a entender al sistema manejador de una base de datos (DBMS) consistente en el paquete que permite entrar, actualizar, recuperar, organizar, consultar, visualizar y generar reportes sobre la base de datos. Bit: Abreviación en inglés de "binary digit". Es la unidad más pequeña que conforma un dato para ser almacenado o procesado por una computadora digital. Generalmente toma únicamente los valores de uno o cero.

Bucle ("loop"): Conjunto de operaciones que son repetidas de manera iterativa.

"Bug": Errores (voluntarios o no) en los programas, que hacen que los programas no funcionen como se tenía previsto.

Byte: Espacio requerido para almacenar un carácter (por ejemplo un signo, una letra, o un dígito). Generalmente consta de ocho dígitos binarios o "bits", razón por la cual se le llama también octete. Cadena: Uno o varios caracteres alfanuméricos o especiales, incluyendo el espacio o blanco (en este caso tiene que estar encerrados entre comillas, para saber el comienzo y fin de la cadena). Es simplemente la secuencia de los códigos de esos caracteres y no representa un valor o cantidad, por lo que puede ser utilizada como nombre de una variable o instrucción en un lenguaje de cómputo (siempre que se respete la sintaxis del lenguaje) o como un dato sobre el cual se va en principio a realizar una operación no numérica como ordenar, consultar, comparar etc

Cargar ("load"): Significa el meter a la memoria de la computadora un programa o archivo que se encuentra en diskette o cualquier otro dispositivo periférico.

Categorías del aprendizaje: Son los cinco dominios de capacidades que un estudiante puede aprender, tal y como lo propuso Gagné: 1) habilidades intelectuales; 2) estrategias cognitivas; 3) información verbal. 4) habilidades motoras; 5) actitudes.

Chip: Es un conjunto de microcircuitos electrónicos generalmente de silicio, encapsulados en una pastilla de cerámica y con unas patas que permiten insertarlos en módulos que están conectados en una tarjeta de circuitos. Los chips pueden ser únicamente para memoria o para operaciones del CPU. Código: es la representación de un carácter o dígito en la computadora para su uso. Código absoluto o código objeto es cuando este conjunto de caracteres representan las instrucciones ejecutables directamente por la computadora. El código ASCII (American Standard Code for Information Interchange) es una representación homologada de los caracteres alfanuméricos y especiales que se usan en los lenguajes de computadora mediante una secuencia de 8 bits (Byte).

Código Binario: representación de cualquier número real mediante bits en el sistema base dos o sistema (numérico) binario.

Código Fuente: Es el conjunto de instrucciones de un programa escrito en un lenguaje de alto nivel y que ya se encuentran en la computadora o alguno de sus periféricos en una representación homologada como la ASCII, pero que no es directamente ejecutable por la computadora y que necesita de un atraducción (interpretación o compilación).

Comando: Es un elemento de un lenguaje sencillo de programación que consiste en órdenes o comandos eventualmente modificados o precisados por una secuencia de parámetros

Compilador: Es un programa especializado en traducir de un lenguaje de computación de alto nivel (código fuente) a un lenguaje llamado máquina (código objeto) listo para la ejecución en una computadora

Computadora: Conjunto de dispositivos, generalmente electrónicos, que permiten procesar y almacenar información siguiendo las instrucciones de un programa.

Computadora Analógica: Computadora cuyas entradas o salidas no son datos o información codificada digitalmente sino variaciones continuas de voltaje (en contraposición a los dos niveles binarios de voltaje que manejan las digitales), para representar, procesar, efectuar mediciones o controlar fenómenos externos. Computadora Digital: Es la computadora más común, la información y datos que trabaja están codificados digitalmente (de manera binaria, discreta y finita). Comunicación de datos: Transferencia de datos de una computadora a otra utilizando línea telefónica o conexión directa por medio de cables

Conferencia por computadora: Define al conjunto de aplicaciones de las computadoras y las telecomunicaciones para conducir discusiones en las que interactúan diversos usuarios desde puntos geográficos distantes.

Corrección de exámenes por computadora: Conjunto de programas que interactúan generalmente con algún dispositivo de entrada como los rastreadores ópticos o los lectores de marcas de lápiz, que permiten analizar las respuestas, calificar, señalar la respuesta correcta, contar las respuestas correctas y realizar alguna estadística sobre el examen o conjunto de exámenes.

Correo electrónico: Uso de la computadora para transmitir y recibir mensajes de otras computadoras, empleando para esto la simulación de buzones por medio de archivo.

Correr ("run"): Sinónimo de ejecutar un programa, es decir la realización del conjunto de instrucciones de un programa por parte de la computadora. "Courseware": Conjunto de material tanto de "software" como material impreso que lo acompaña, específicamente diseñado para uso instruccional o de enseñanza, generalmente en el salón de clase.

CPU: Siglas en inglés para la unidad de procesamiento central de una computadora. Tales siglas se emplean también en español y definen al conjunto de circuitos de una computadora que se encarga de realizar todas las operaciones de control, lógico aritméticas y direccionamiento a la memoria. \

Criptografía: Técnica consistente en codificar los mensajes que se transmiten o almacenan especialmente en o entre computadoras, para que no se entiendan directamente y necesiten de un desciframiento.

Cursor: Es un marcador que indica el lugar en la pantalla en el cual espera una entrada por parte del usuario. El cursor generalmente es un guión que parpadea pero puede ser una línea, un cuadrado, una flecha, dependiendo de la computadora. El cursor se puede mover tanto por las teclas de flecha como por algún dispositivo externo como el ratón.

Dato: Cualquier información utilizada por un programa tanto numérica como alfabética que no tiene algún significado en sí, si no dentro de un contexto con otros datos con los que se relaciona (información).

Diagrama de Flujos: Es un diagrama que muestra la secuencia de las instrucciones de un programa con todo y sus ramificaciones y la lógica necesaria para las ramificaciones, repeticiones, etc.

Directorio: Lista de los archivos almacenados en un disco u otro dispositivo periférico.

Disco duro: Disco rígido de plástico con sustrato magnético que almacena una gran cantidad de información. Tiene que estar al vacío y generalmente es fijo o no reemplazable dentro de una computadora.

Diskette: Es el medio más común de almacenamiento de programas y archivos de datos que son leídos y escritos en una unidad de diskettes en las microcomputadoras. Consiste de una base plástica común con un sustrato magnético en el que se almacenan los datos en pistas magnéticas concéntricas cubierta por una capa de mylar y otros lubricantes encerradas dentro de un sobre plástico. Se les conoce también como "floppy disk" o discos flexibles. Existen de varios tamaños y densidades. Documentación: Conjunto de manuales y comentarios en los programas de cómputo que explican los objetivos, el diseño, el uso correcto, las aplicaciones, ejemplos y posibles errores de un "software".

Editor: Es un programa especial que permite la creación y modificación de un archivo, por ejemplo texto o datos.

Educación basada en computadora: Define al campo de aplicaciones que utiliza la computadora como soporte de cualquier proceso educativo o de apoyo a la educación, en consecuencia comprende a la enseñanza asistida por computadora, a la administración escolar y a la administración de la educación asistida por computadora.

Ejercitación y práctica ("Drill and Practice"): Modalidad de la enseñanza asistida por computadora que pretende el aprendizaje de conocimientos o habilidades a través de la realización de ejercicios y la visualización de las respuestas apropiadas.

En línea ("on line"): Técnica de trabajo con una computadora de manera instantánea o no diferida ("batch" o procesamiento en lotes).

Encendido caliente ("boot"): Se refiere cuando una microcomputadora es encendida al hecho de que automáticamente corre un programa fijo que se encuentra en el ROM, y ejecuta las instrucciones necesarias para cargar y comenzar al sistema operativo y a un programa más grande que generalmente se encuentra en un diskette.

Enseñanza asistida por computadora (EAC): Referida también como instrucción asistida por computadora, define el campo de metodologías, técnicas, y programas que empleando la computadora pretenden enseñar algo a un estudiante. De alguna manera auxilia o sustituyen al maestro proveyendo información, ejercicios, demostrando conceptos y convirgiendo las ideas dentro de un modo interactivo que requiere respuesta del estudiante y que a su vez modula la dirección de la instrucción.

Ergonomía: Estudio de las interacciones fisiológicas entre el hombre y la máquina cuyos factores permiten una mejor adaptación de los dispositivos para aumentar la eficacia y disminuir los problemas como la fatiga, los errores y falta de comodidad. Estrategia del control del aprendizaje: Procedimientos instruccionales que permiten a los estudiantes de un "courseware" tomar decisiones sobre las lecciones, las actividades y exámenes a seguir. Expresión: instrucción de un lenguaje de computadora que de acuerdo a la sintaxis del lenguaje relaciona datos (variables o constantes) entre sí y con operadores.

Firmware: Programa formado por "micro-instrucciones" que se encuentran en memoria fija tipo ROM dentro de una computadora.

Generación de exámenes por computadora: Programa de computadora que a partir de una base de datos de reactivos (preguntas y respuestas que conforman un examen), genera al azar el conjunto de preguntas que conforman un examen sobre uno o varios tópicos.

Grupo de trabajo ("working pool"): Conjunto de elementos de la modalidad de ejercitación y práctica que van a ser enseñados y preguntados al estudiante en un orden no predefinido.

"Hardware": Es el equipo computarizado, incluyendo a todos los componentes físicos, internos y externos, de una computadora y sus periféricos.

Herramienta: Paquete de cómputo que ofrece todas las facilidades para desempeñar una actividad sin tener que programar la computadora. Por ejemplo el procesador de textos ofrece todas las facilidades de maquinilla y en general de producir textos. Se puede aprender algo indirectamente empleando herramientas, por ejemplo contabilidad empleando una hoja electrónica de cálculo. Así indirectamente mejoran sus facultades de descubrimiento, decisión y deducción. Se contraponen a la enseñanza o instrucción asistida por computadora cuyo énfasis y métodos se dedican a la inculcación del conocimiento y habilidades, pero sobretodo porque el contenido explícitamente trata sobre el tema a aprender, mientras que la herramienta es libre de contenido. Hoja de notas ("grade book"): Conjunto de programas de computadora que permiten al maestro llevar memoria de las calificaciones de exámenes y asignaciones dejadas a los estudiantes, permite calcular la calificación final y generar diferentes reportes de avance.

Hoja electrónica de cálculo: Paquete de aplicación que utiliza columnas y renglones de una hoja simulada en la memoria de la computadora para la realización y repetición fácil de cálculos en las celdas de dicha hoja.

Iconos: Símbolos visuales empleados para representar opciones dentro de un menú o datos.

Inicializar ("formatting"): Es el preparar un diskette nuevo o virgen para que reciba información bajo las convenciones de un sistema operativo.

Instrucción ("statement"): Estructura sintáctica básica de un lenguaje de computadoras, que empleando una estructura y generalmente palabras reservadas y expresiones le indican a la computadora la realización de una acción específica. Instrucción Asistida por Computadora: Sinónimo de Enseñanza Asistida por Computadora, muy conocida por sus siglas en Inglés: CAI.

Inteligencia Artificial: Rama de la computación que se encarga de estudiar, desarrollar e imitar los procesos humanos inteligentes. Principalmente se enfoca hacia los procesos de razonamiento e inferencia (sistemas expertos), el aprendizaje, la adaptación y autocorrección, el reconocimiento y generación de formas y patrones y el control y combinación del movimiento (robótica). Interacción: Forma de operar con un lenguaje de computadora en el cual durante la ejecución el usuario puede entrar datos, información y eventualmente instrucciones. Interactivo o conversacional: forma de comunicación dialógica o bidireccional entre el hombre y la computadora. Interfaz (a veces mal llamado Interfase): Dispositivo físico electrónico que permite la conexión de una computadora con un periférico, generalmente a través de cables. La interfase en serie transmite y recibe información de un bit a la vez, mientras que la interfase paralela transmite o recibe un "byte" o palabra a la vez.

Intérprete: Lenguaje de alto nivel parecido a un compilador pero en el momento en que se traducen las instrucciones al lenguaje de máquina se mandan éstas a ejecutar.

Justificación: Alineamiento vertical para la impresión de un texto tanto a la izquierda como a la derecha, en una hoja a ser impresa.

Laboratorio de computadoras: Un local dentro de una escuela en el cual, o bien existe una computadora con terminales o bien existe un conjunto de microcomputadoras, así como todos los dispositivos tales como impresora, que permiten a los estudiantes realizar los trabajos asignados con computadora.

Lenguajes de autor: Lenguajes de computadora de alto nivel que permiten a un maestro sin mucho conocimiento en computación la realización de un "courseware" sencillo, en algunos casos simplemente respondiendo a una serie de preguntas y dando los formatos adecuados.

Lenguaje de Computadora: conjunto de primitivas, estructuras sintácticas y semánticas que describen un conjunto de instrucciones y la forma y tipo de datos con los que va a trabajar la computadora, sea por comandos directos o por programas.

Lógica Binaria o Algebra de Boole (Booleana): Sistema lógico y algebraico en el que únicamente se tienen dos valores posibles: Verdadero y Falso, a veces también se maneja como 0 y 1. Shannon aplicó esta lógica a los circuitos de conmutación y creo lo que se conoce como Circuitos Combinatorios. En esta lógica solo hay dos constantes 0 o 1, y todas las variables y expresiones que se puedan armar pueden valer sólo 0 o 1. Los circuitos de una computadora se basan en esta lógica y en la teoría de autómatas. No confundir con sistema binario.

Marco ("Frame"): Unidad pedagógica básica de presentación de un subobjetivo de una lección que forma parte de un "courseware".

Memoria: Parte de una computadora donde se almacena datos o instrucciones. Cuando ésta es de tipo permanente se le llama ROM y cuando es temporal o volátil se le llama RAM. Cuando está dentro de la computadora y trabaja directamente con el CPU se le llama memoria principal, cuando es externa a la computadora y trabaja directamente con la memoria principal se le llama memoria secundaria.

Microcomputadora: Sinónimo de computadora personal consistente en la categoría más pequeña de computadoras y cuyo CPU (actualmente) se encuentra en un solo chip llamado microprocesador.

Modem: Dispositivo que permite transformar la información digital para ser enviada y recibida a través de líneas telefónicas.

"Overlay" (superposición): Es la sobreposición que se puede realizar en una pantalla de dos o más imágenes, generalmente a través de recuadros o ventanas.

Monitor o Pantalla: Se emplea este término en dos sentidos, el primero como sinónimo de monitor o "display", pero dentro del contexto de enseñanza asistida por computadora como la información que en un momento dado se presenta en el monitor ante los ojos del estudiante. Aunque es deseable que un marco quepa en una pantalla no es indispensable.

Monitor "bit-map": Es un tipo de monitor o pantalla de display que permite direccionar cada uno de los puntos de la pantalla, llamados "pixel". Esto se contrasta con los monitores de rastreo en el cual solamente pueden direccionar una línea completa y la gráfica se forma bariendo todas las líneas del monitor de arriba hacia abajo. Es claro que los monitores "bit-map" tienen un mayor poder de resolución.

Paginación: Técnica del "courseware" que contrasta con el barrido en la cual cada imagen llamada "pantalla" es reemplazada enteramente por otra.

Paquete o Software de Aplicación: Conjunto de Programas que realizan o satisfacen a toda una aplicación, por ejemplo: el procesador de textos, la hoja electrónica de cálculo, los sistemas manejadores de bases de datos etc.

Periféricos: Es cualquier dispositivo físico (de "hardware") de entrada, salida y almacenaje que puede ser conectado a una computadora.

Pixel: Es el punto más pequeño que puede ser direccionado en una pantalla o monitor de computadora (es la abreviación en inglés de las palabras "picture element").

Pizarra de anuncios ("bulletin board"): Conjunto de archivos dentro de un sistema de correo electrónico en el cual los participantes pueden escribir mensajes para uno, varios o todos los usuarios del sistema y a su vez puede ser consultado por el usuario mismo.

Pizarrón electrónico: Dispositivo que se conecta a la computadora consistente en un pizarrón que no es otra cosa que una pantalla grande, en la que la tiza o gis permite indicar a la computadora lo que se está señalando.

Procesamiento: Nombre que se le da a cualquier modificación o transformación que sufren datos o información al interior de una computadora. Las transformaciones son principalmente para calcular o arreglar información, pero también para meter en memoria, consultar, transmitir y visualizar o presentar información.

Procesador de texto ("word processor"), a veces mal llamado procesador de la palabra: Paquete que permite la creación, modificación e impresión de texto y en general, el de cualquier material escrito. Las versiones avanzadas de esto cuenta con facilidades de corrección de ortografía, diccionario etc. Cuando se le combina con gráficas para producir páginas de alta calidad para impresión de una publicación se le llama "desk-top publishing".

Programa: Un conjunto de instrucciones escritas en un lenguaje de computadora que tienen un principio y un fin y realizan una tarea específica.

Prospectiva: Es el estudio de las tendencias y realización de los escenarios deseables para un futuro, sus condiciones y sus consecuencias.

Protocolo de Comunicación: es el conjunto de parámetros técnicos, reglas y lenguaje bajo las cuales se establece una comunicación entre dos computadoras para intercambiar datos.

Ramificación: Es el punto de decisión que permite controlar la secuencia en un programa.

Ratón: Dispositivo de entrada empleado para mover el cursor dentro de la pantalla y eventualmente para seleccionar alguna opción. El ratón se desplaza sobre una superficie plana cuyo movimiento es repetido por el cursor en la pantalla.

Rastreo ("debugging"): Es el proceso de encontrar y fijar los errores en un programa para poderlos corregir. Se entiende por fijar un error el hecho de circunscribirlo de tal manera que no afecte otras partes del programa.

Red ("network"): Sistema de cómputo en el que varias computadoras se encuentran conectadas, ya sea para transmitir-recibir datos (telecomunicación), o para compartir instrucciones a ejecutar (proceso distribuido).

Retroalimentación ("feedback"): conjunto de acciones y mensajes que genera un "courseware" como resultado inmediato de una petición o respuesta del estudiante.

Salvar: Anglismo ("save") para indicar que se guarda algo (un programa, archivo, datos) de la memoria de la computadora (RAM) a un medio de memoria secundaria, generalmente diskette. Cuando un conjunto de información se guarda en memoria secundaria bajo un solo nombre se le llama archivo.

Simulación: Conjunto de programas instruccionales que permiten experimentar en un modelo de computadora una situación dinámica real.

Sistema Binario o sistema numérico binario: Se refiere al sistema de representación de números y a su aritmética, cuya base es dos. En este sistema todos los dígitos de un número cualquiera son formados exclusivamente con 0 ó 1. No confundir con lógica binaria o álgebra de Boole. También se emplea para indicar una decisión que involucra dos alternativas posibles, como por ejemplo verdadero y falso.

Sistema de computadora: Conjunto de dispositivos "hardware" que permite funcionar a una computadora y sus periféricos: CPU, memoria interna, dispositivos de entrada y de salida.

Sistema operativo: Conjunto de programas básicos de cualquier computadora que le permiten efectuar sus funciones primarias y trabajar con sus periféricos.

"Software": Cualquier programa, paquete, o sistema que contiene instrucciones para que una computadora haga algo.

Subrutina: Es un sinónimo de subprograma, es decir una parte autocontenida de un programa de computadora que realiza una función específica pero que tiene que ser llamada a ejecución por un programa principal.

Teclas de función: Conjunto de teclas especiales numeradas que no tienen una definición fija y que por lo tanto pueden ser utilizadas por un programa para facilitar la realización rápida de algunas instrucciones.

Teletexto: Es un sistema de información en el que un video y a veces una computadora capata, selecciona, presenta en la pantalla la información que se distribuye desde un punto central.

U.A.L. Unidad Aritmética y Lógica (en inglés ALU): Es la parte del CPU encargada de realizar todas las operaciones aritméticas como sumar o restar y las lógicas como comparar, desplazar etc.

Unidad de control (UC): es la parte del CPU que se encarga de organizar y dirigir el funcionamiento de las demás unidades de una computadora. Forma junto con el UAL el CPU.

Unidad de diskette: Dispositivo generalmente inserto en el chasis de una microcomputadora que permite leer y escribir información en los diskettes ("floppy disks").

Variables: Nombres o etiquetas dados a lugares en la memoria de la computadora que contienen datos (valores) en un programa, dado que son llamados por un programa como etiquetas el valor puede cambiar.

Vector: Una lista ordenada de variables del mismo tipo (nombres, números, etc.).

Ventanas: Es una porción encuadrada de la pantalla que es manejada independientemente de la información desplegada en el fondo de la pantalla. Las pantallas se superponen ("overlay") y pueden desplazarse dentro de la pantalla o cambiar el tamaño y forma.

Videotexto: es un sistema de información de comunicación bidireccional entre una computadora que tiene acceso a grandes bancos de datos centrales y los pone a disposición de un usuario que mediante una computadora o un video de videotexto interroga y selecciona información que despliega visualmente. Existen sistemas que permiten el acceso a bancos de noticias, de entretenimiento, información y los pone a disposición de las personas como si fuera una computadora.

Objeto: Es una persona, lugar o cosa. Ejemplo: Una manzana. Todos los objetos tienen un estado y un comportamiento. Esto es muy importante. Para el ejemplo de la manzana, la manzana (hablando de una manzana normal de color rojo) recién bajada de un árbol, está completa, es dura y de color rojo. Este es su estado. Uno sabe que si se muerde la manzana va a sonar un "crunch", y esto se puede interpretar como un comportamiento. Después de haberla mordido la manzana cambió de estado. Los objetos pueden tener más de un comportamiento y más de un estado.

Clases: Un grupo de objetos con las mismas características. Una clase contiene:

1. Una descripción de las variables internas de los objetos de la clase.
2. Las operaciones que se pueden aplicar al objeto de la clase.

Encapsulación: Principio en el cual un objeto esconde la información de su implementación a otro objeto y a los usuarios. En otras palabras los objetos no te dicen cómo están hechos por dentro.

La encapsulación tiene dos beneficios primordiales:

1. Modularidad. El código de un objeto puede ser escrito y se puede mantener independiente del código de otros objetos. Un objeto se puede mover de sistema en sistema, se puede quitar, modificar y volver a colocar sin alterar el sistema general.
2. Esconder la información (*information hiding*). Un objeto tiene una interfase con la que otros objetos se pueden comunicar, pero puede mantener información privada para sí misma que puede cambiar en cualquier momento sin afectar a los objetos que dependen de esta.

Herencia: Principio por el cual una clase se puede derivar de otra clase ya existente, y hereda las características del padre. Un ejemplo:

Tengo el objeto fruta, con algunas propiedades inherentes a la fruta. Luego tengo el objeto manzana, que es hijo de fruta y por lo tanto hereda las propiedades de la fruta pero puede tener otras propiedades propias de la manzana. Este es un ejemplo sencillo pero más o menos ilustra lo que estoy tratando de explicar. El hijo hereda las características del papá, pero tiene otras propias y mejora (o empeora) algunas características del papá.

Mensaje: Los objetos se comunican unos con otros mediante mensajes.

Polimorfismo: Principio en el cual objetos de diferentes clases pueden entender el mismo mensaje pero responder de manera distinta.

ANSI: ANSI es el juego de caracteres estándar más utilizado por los equipos personales. Como el estándar ANSI sólo utiliza un byte para representar un carácter, está limitado a un máximo de 256 caracteres y signos de puntuación. Aunque es adecuado para el inglés, no acepta totalmente muchos otros idiomas.

DBCS: DBCS se utiliza en los sistemas Microsoft Windows que se distribuyen en la mayor parte de Asia. Proporciona soporte para muchos alfabetos de Asia Oriental, como el chino, el japonés y el coreano. DBCS utiliza los números de 0 a 128 para representar el juego de caracteres ASCII. Algunos números mayores de 128 funcionan como *caracteres de cabecera* que no son realmente caracteres sino simples indicadores de que el valor siguiente es un carácter de un juego de caracteres no occidental. En DBCS, los caracteres ASCII tienen una longitud de 1 byte, mientras que los caracteres japoneses, coreanos y de otros países de Asia Oriental tienen una longitud de 2 bytes.

Unicode: Unicode es un esquema de codificación de caracteres que utiliza 2 bytes por *cada* carácter.

ISO (International Standards Organization) define un número dentro del intervalo 0 a 65.535 ($2^{16} - 1$) por cada carácter y símbolo de cada idioma (más algunos espacios vacíos para futuras ampliaciones). En todas las versiones de 32 bits de Windows, el Modelo de objetos componentes (COM), que es la base de las tecnologías OLE y ActiveX, utiliza Unicode. Unicode es totalmente compatible con Windows NT. Aunque Unicode y DBCS tienen caracteres de doble byte, los esquemas de codificación son completamente diferentes.

Visual Basic utiliza Unicode para almacenar y manipular cadenas. Unicode es un juego de caracteres en el que se emplean 2 bytes para representar cada carácter. Algunos otros programas, como la API de Windows 95, utilizan ANSI (American National Standards Institute) o DBCS para almacenar y manipular cadenas. Cuando mueva cadenas fuera de Visual Basic, puede que encuentre diferencias entre Unicode y ANSI/DBCS. En la tabla siguiente se muestran los juegos de caracteres ANSI, DBCS y Unicode en diferentes entornos.