

Universidad Nacional Autónoma de México

Facultad de Estudios Superiores Cuautitlán

***“MODELO DE SISTEMA PARA REALIZAR MATERIAL DIDÁCTICO
DE APOYO A DOCENTES”***

TESIS

**QUE PARA OBTENER EL TÍTULO DE:
LICENCIADO EN INFORMÁTICA
PRESENTAN:**

***VALENCIA JUÁREZ SANDRA MAGALY
VILCHIS GUERRA OSCAR***

ASESOR: ING. OSCAR HERNANDEZ SÁNCHEZ



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



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.

INDICE

INTRODUCCIÓN.....	1
1. TECNOLOGIA Y EDUCACIÓN.....	3
1.1 POR QUÉ USAR LOS MEDIOS TECNOLÓGICOS EN LA EDUCACIÓN.....	4
1.2 IMPORTANCIA Y CARACTERÍSTICAS DE LA TECNOLOGIA APLICADA A LA ENSEÑANZA.....	4
1.3 SELECCIÓN DE LA TECNOLOGÍA INFORMÁTICA ADECUADA.....	6
1.4 SOFTWARE EDUCATIVO Y OTROS MEDIOS TECNOLÓGICOS APLICADOS A LA EDUCACIÓN.....	6
1.4.1 E – LEARNING.....	10
1.4.2 M – LEARNING.....	11
1.5 EVALUACION DE SOFTWARE EDUCATIVO.....	11
1.5.1 EVALUCION DEL PROGRAMA POR LA FUNCIONALIDAD EN SU USO.....	12
1.5.2 EVALUACION DEL EL PROGRAMA COMO OBJETO PEDAGÓGICO.....	14
1.6 EL IMPACTO DE LAS NUEVAS TECNOLOGIAS INFORMATICAS EN ESTUDIANTES Y DOCENTES.....	17
1.6.1 APLICACIONES DIDACTICAS DE LA COMPUTADORA EN EL AULA.....	18
1.6.2 ACTITUDES HACIA LAS NUEVAS TECNOLOGÍAS EN LA EDUCACIÓN.....	19
1.7 TECNOLOGIAS APLICADAS A LA EDUCIÓN EN MÉXICO.....	21
2. ANALISIS Y DISEÑO DE SISTEMAS DE INFORMACIÓN.....	24
2.1 ESTRATEGIAS PARA EL DESARROLLO DE SISTEMAS.....	26
2.2 METODO DE DESARROLLO POR ANÁLISIS ESTRUCTURADO.....	27
2.2.1 ANALISIS DE FLUJO DE DATOS.....	27
2.2.1.1 CARACTERÍSTICAS DE LA ESTRATEGIA DE FLUJO DE DATOS.....	27
2.2.1.2 HERRAMIENTAS DE FLUJO DE DATOS.....	28
2.3 METODO DEL CICLO DE VIDA DEL DESARROLLO DE SISTEMAS.....	38
2.3.1 IDENTIFICACIÓN DE PROBLEMAS, OPORTUNIDADES Y OBJETIVOS.....	39
2.3.2 DETERMINACIÓN DE LOS REQUERIMIENTOS DE INFORMACIÓN.....	42
2.3.3 ANÁLISIS DE LAS NECESIDADES DEL SISTEMA.....	46
2.3.4 DISEÑO DEL SISTEMA RECOMENDADO.....	47
2.3.5 DESARROLLO Y DOCUMENTACIÓN DEL SOFTWARE.....	50
2.3.6 IMPLEMENTACIÓN Y EVALUACIÓN DEL SISTEMA.....	53
3. TIPOS DE SOFTWARE.....	54
3.1 SISTEMAS OPERATIVOS.....	55
3.1.1 TIPOS DE SISTEMAS OPERATIVOS.....	56
3.1.2 CLASIFICACION DE LOS SISTEMAS OPERATIVOS.....	56
3.2 SOFTWARE DE APLICACIÓN.....	57
3.2.1 SOFTWARE DE APLICACIÓN ESPECÍFICA.....	58
3.2.2 SOFTWARE DE APLICACIÓN GENERAL.....	59
3.3 SOFTWARE DE COMUNICACIÓN.....	60
3.4 LENGUAJES.....	60
3.4.1 CLASIFICACIÓN DE LOS LENGUAJES DE PROGRAMACIÓN.....	61
3.4.1.1 CLASIFICACIÓN DE LOS LENGUAJES DE ACUERDO A SU NIVEL.....	62
3.4.1.2 CLASIFICACIÓN DE LOS LENGUAJES DE ACUERDO A SUS APLICACIONES PRINCIPALES.....	67
4. ANALISIS Y DISEÑO DE BASES DE DATOS.....	70
4.1 SISTEMAS MANEJADORES DE BASES DE DATOS (SMBD).....	72
4.1.1 MODELOS DE DATOS.....	72
4.1.2 INDEPENDENCIA DE DATOS.....	72
4.1.3 REESTRUCTURACIÓN DE LA BASE DE DATOS.....	73
4.1.3.1 ARQUITECTURA DE SMBD.....	73
4.1.3.2 AMBIENTE OPERACIONAL DE UNA BASE DE DATOS.....	74
4.1.3.3 INTEGRIDAD DE LA BASE DE DATOS.....	74
4.1.3.4 DISTRIBUCIÓN DE LA BASE DE DATOS.....	75
4.1.3.5 CONSIDERACIONES DE DISEÑO EN BASES DE DATOS DISTRIBUIDAS.....	75
4.1.3.6 HERRAMIENTAS CASE Y DICCIONARIOS DE DATOS.....	75
4.2 SELECCIÓN DE UN SMBD.....	76
4.2.1 MONITOREO DE LA BASE DE DATOS.....	77
4.2.2 MEDICIÓN DEL DESEMPEÑO.....	77
4.2.3 REQUERIMIENTOS DE ALMACENAMIENTO.....	78
4.2.4 METODOLOGIAS DE DISEÑO.....	79
4.2.5 TÉCNICAS DE DISEÑO.....	79
4.2.6 ETAPAS DE DISEÑO.....	81
4.3 REQUERIMIENTOS DE ESPECIFICACION Y DISEÑO DE BASES DE DATOS.....	83
4.3.1 HERRAMIENTAS CASE.....	85

4.3.2 DISEÑO TÉCNICO DE LA BASE DE DATOS.....	85
4.4 MODELO RELACIONAL.....	86
4.4.1 DISEÑO DE LA BASE DE DATOS RELACIONAL.....	87
4.4.2 ESTRUCTURA BASICA.....	89
4.4.3 PROPIEDADES DE RELACIONES.....	91
4.5 EL MODELO ENTIDAD-RELACIÓN	91
4.6 NORMALIZACIÓN DE UNA BASE DE DATOS.....	95
4.6.1 GRADOS DE NORMALIZACIÓN	96
4.6.1.1 PRIMERA FORMA NORMAL	96
4.6.1.2 SEGUNDA FORMA NORMAL	98
4.6.1.3 TERCERA FORMA NORMAL.....	100
5. SISTEMAS DE AUTOAPRENDIZAJE ELECTRÓNICO.....	102
5.1 PROGRAMACIÓN DE LA ENSEÑANZA INDIVIDUALIZADA.....	103
5.1.1 EL PROGRESO DE LA ENSEÑANZA.....	105
5.1.2 MEDICIÓN, EVALUACIÓN, ESTADÍSTICA Y PLANEACIÓN INVESTIGATIVA.....	106
5.2 PRINCIPIOS Y CONDICIONES DEL APRENDIZAJE.....	106
5.2.1 EL CONDICIONAMIENTO CLÁSICO Y OPERANTE.....	107
5.2.1.1 EL CONDICIONAMIENTO CLÁSICO.....	107
5.2.1.2 CONDICIONAMIENTO OPERANTE.....	108
5.2.2 MODELO DE APRENDIZAJE ACUMULATIVO.....	112
5.2.2.1 TIPOS DE APRENDIZAJE.....	112
5.3 TEORÍAS PROCESO ENSEÑANZA-APRENDIZAJE.....	115
5.3.1 LA PERSPECTIVA CONDUCTISTA: SKINNER.....	115
5.3.2 AL APRENDIZAJE SIGNIFICATIVO DE AUSUBEL.....	117
5.3.3 APRENDIZAJE POR DESCUBRIMIENTO: BRUNER.....	118
5.3.4. LA TEORÍA DE PIAGET.....	119
5.3.5. PROCESAMIENTO DE LA INFORMACIÓN: GAGNÉ.....	120
5.3.6. EL CONSTRUCTIVISMO DE PAPERT.....	122
5.3.7. CONSTRUCTIVISMO Y MEDIACIÓN.....	123
5.3.8 RETENCIÓN Y TRANSFERENCIA.....	125
6. CASO PRACTICO (MODELO DE SISTEMA PARA REALIZAR MATERIAL DIDÁCTICO DE APOYO A DOCENTES).....	126
6.1 DESCRIPCIÓN DEL MODELO.....	127
6.2 DIAGRAMA DE MÓDULOS.....	129
6.3 DESCRIPCIÓN DE TABLAS DE LA BASE DE DATOS.....	130
6.4 DIAGRAMA DE BASE DE DATOS	131
6.5 MANUAL DE USUARIO	132
CONCLUSIONES.....	150
BIBLIOGRAFÍA	151
ANEXO I	153

DEDICATORIA

A mis padres

Quiero dedicar esta tesis a mis padres quienes son muy importantes para mí, porque han dado verdadero significado a la palabra familia y porque con ellos he aprendido muchas cosas, entre ellas que cada momento de la vida se debe afrontar con fe, valentía, voluntad y optimismo.

Porque son ellos quienes me han impulsado para llegar a este día tan importante de mi vida y por muchas cosas más, este es un pequeño tributo a todos sus esfuerzos.

AGRADECIMIENTOS

La vida es como una obra en la que en cada capítulo se va desarrollando parte de una historia; pues bien, así ha sido mi vida, una historia en la cual han aparecido personajes que han dejado una huella imborrable en la misma, así pues deseo agradecerles a aquellos que han estado presentes regalándome parte de su conocimiento, su filosofía ante la vida y de su tiempo, pues esto me ha ayudado a crecer.

A Dios.

Gracias por estar siempre a mi lado cuando más te he necesitado y darme fortaleza y alivio. Por permitirme llegar hasta este día para entregarlo a mis padres como un presente. Gracias a por mi familia y porque has puesto en mi camino a personas excepcionales de las que he aprendido mucho para crecer como persona y como profesionalista.

A mi madre.

Gracias por haber estado siempre a mi lado para protegerme y cuidarme, por haber puesto tanto empeño en mi formación , por cada minuto que has estado junto a mí y por haberme impulsado y mostrado que la voluntad es lo más importante para lograr nuestros objetivos. Gracias por tus palabras de aliento cuando estoy triste, por todo tu cariño y apoyo en todo momento, por preocuparte por mí siempre, porque eres la persona con quien siempre podré contar. Gracias porque me enseñaste que cada día se puede aprender algo nuevo. Por ser mi madre, mi amiga y un ejemplo a seguir.

A mi padre.

Gracias por ser valiente, porque de esa fuerza he aprendido como salir adelante, porque tu ejemplo siempre ha sido más fuerte que cualquier palabra, porque cada vez que te observaba leer un libro con tanta atención inculcabas en mí el deseo de aprender, siendo mi ejemplo para tomar cariño por el estudio y enfrentar la vida con una sonrisa. Gracias por haberte desvelado conmigo después de una larga jornada de trabajo enseñándome las matemáticas que no entendía y por tus palabras de aliento en los malos momentos.

Gracias por ser mi padre, mi maestro de la vida y mi apoyo.
Y porque sin todo lo anterior, no hubiera podido llegar hasta este punto de mi vida.

A mi hermana.

Gracias hermanita porque siempre me has escuchado y me ayudas a pensar con claridad en momentos difíciles, porque cuando me siento sola, tu estás ahí para demostrarme que el mundo no es tan malo como parece y por tener siempre palabras de aliento para seguir adelante.

A mi hermano.

Gracias por ser una personita tan noble, por protegerme y ayudarme a pensar con claridad, por alegrarme cuando estoy afligida y por estar conmigo para compartir momentos gratos y tristes.

Prof. Oscar Hernández S.

Gracias por confiar en mi y apoyarme para lograr que este trabajo de tesis se realizara de manera correcta, por haberme obsequiado ese conocimiento a través de sus pláticas dentro del aula y fuera de ella y por aclarar siempre mis dudas.

U.N.A.M.

Le doy gracias a la U.N.A.M. pues me dió la oportunidad de pertenecer a esta prestigiosa institución y hoy tener la satisfacción de presentar esta tesis para concluir un ciclo más en mis estudios y poder servir a mi patria aplicando el conocimiento adquirido.

A mi compañero de tesis Oscar Vilchis G.

Gracias Oscar por el apoyo brindado para la elaboración de esta tesis, por haber decidido conjuntar las habilidades y creatividad de ambos para el correcto desarrollo de este trabajo que con mucho esfuerzo, dedicación y cariño hemos logrado terminar, pero sobre todo gracias por ser un gran amigo.

A mis Amigos.

Gracias a todos mis amigos por todas las aventuras que hemos pasamos juntos y porque hicieron de esta etapa una de las más emocionantes de mi vida al compartir todo tipo de momentos: dolorosos, gratos, divertidos y meditabundos. Gracias amigos.

DEDICATORIA

A mis padres

Les dedico este trabajo de tesis, ya que siempre han dado la vida y todo su ser por mí, por enseñarme que el trabajo y la perseverancia nos ayudan a forjarnos como personas para ser mejores seres humanos, además de ser mi guía, educarme y darme todo su amor.

Gracias padres por apoyarme en todo momento y ser el motor para luchar y lograr todas mis metas, así como para llegar a este día que es muy importante en mi vida, nunca tendré con que agradecerles todo lo que me han dado, los quiero muchísimo.

AGRADECIMIENTOS

Gracias dios por guiarme y darme ese aliento día a día para seguir adelante y tomar decisiones correctas, brindándome tu amor comprensión y cariño.

A mi padre.

Te agradezco papá, por enseñarme a tomar lo mejor de todas las situaciones que se ponen en el camino, por inculcarme el valor al trabajo y sobretodo por ser mi mejor amigo.

A mi madre.

Mi fresita hermosa, te agradezco por darme tu cobijo y ayudarme a levantarme cuan me sentía caído y sin fuerzas gracias por ser ese confidente en todo momento, brindándome tu comprensión y cariño.

A mi esposa.

Por formar parte de mi vida apoyarme siempre y estar a mi lado en cualquier adversidad así como en los mejores momentos de mi vida como este, gracias amor por ser mi compañera te amo.

Profr. Oscar Hernández Sánchez.

Por apoyarnos en este trabajo de tesis, así como todas la horas que nos brindo transmitiéndonos sus conocimientos y enseñanzas en el aula, así como su dedicación y tiempo para llevar a termino este proyecto.

A la UNAM.

Muchísimas gracias a la máxima casa de estudios por brindarme su cobijo durante mis estudios profesionales, obteniendo en sus aulas los conocimientos para poder enfrentar los retos de la vida profesional, y sobretodo por tener el honor de ser universitario y formar parte de esta enorme institución.

A mi compañera de tesis Magaly Valencia (Magy).

Magy gracias por todo el apoyo para concretar este trabajo de tesis, uniendo esfuerzos para conseguir el buen termino de este proyecto, y sobretodo por ser una compañera y amiga durante toda la carrera y fuera de ella, gracias maguita.

A mis Informáticos

Gracias amigos por compartir todos esos momentos, aprendiendo de ustedes y llevándome un cachito de todos en lo bueno lo malo y lo peor, gracias informaticos, ami tocayo aguantarme desde el 1er semestre y ser parte del equipo oscars al cuadrado, a Carlitos por ser un buen amigo y escuchar siempre, a luix gracias por acermre reir tanto e irnos de farra, Alice por ser mi amiga incondicional, en fin a todos por estar siempre ahí gracias.

INTRODUCCIÓN

En la actualidad la tecnología ha avanzado en grandes proporciones y esta ha sido aplicada a diversos ámbitos de la sociedad, uno de ellos es la educación; sin embargo, a pesar de que se han obtenido grandes avances en la aplicación de la tecnología informática en este rubro, no podemos negar que este desarrollo también ha sido grandemente relegado debido a razones económicas en nuestro país principalmente, por consiguiente, no existen en el mercado nacional suficientes herramientas para la creación de material educativo además de que los contenidos en esta área son altamente regionales.

Sólo como antecedente podemos mencionar que actualmente la Secretaría de Educación Pública (SEP) está convocando a concursos para desarrollar software educativo, esto con la finalidad de que este material proporcione soporte a la enseñanza dentro de las aulas educativas, sin embargo, los materiales inscritos en dicho concurso no han cumplido con las características de software debido a que sólo son apuntes de los profesores integradas en presentaciones power point.

Es por ello, la presente tesis tiene como objetivo primordial crear un modelo de sistema que se utilice como herramienta auxiliar en la construcción de actividades didácticas interactivas orientadas al autoaprendizaje, principalmente.

Para ello, es de vital importancia conocer tanto las diferentes corrientes teóricas del aprendizaje, las formas de evaluación de los materiales educativos y la forma de evaluación de los estudiantes; así como las técnicas de programación, el ciclo de vida de un sistema, las herramientas auxiliares para el desarrollo de un sistema, el análisis y diseño de sistemas y el impacto empresarial y ético de los sistemas dentro de la sociedad. Cabe resaltar que el presente trabajo está basado en la teoría constructivista por ser la más utilizada y respaldada por las instituciones educativas a nivel nacional y mundial.

Objetivos Generales

- Diseñar un modelo de software educativo que apoye el proceso de enseñanza-aprendizaje y que los profesores puedan realizar actividades didácticas con un mínimo de conocimientos de manejo de software de diseño.
- Generar conocimiento en áreas de interés para alumnos de diferentes grados de enseñanza, que permita un continuo desarrollo y actualización tanto de sus estudiantes como de los profesores.
- Complementar el proceso de enseñanza-aprendizaje con ayuda de tecnologías informáticas.

Objetivos Específicos

- Facilitar la construcción de herramientas educativas y que estas se puedan reutilizar por otros profesores o alumnos.
- Utilización de medios electrónicos para la enseñanza y aprendizaje continuo del alumno.
- Proporcionar un software de alta calidad pedagógica y técnica.

Hipótesis

El presente trabajo de tesis muestra un modelo de software constructor de actividades didácticas dirigido a docentes y alumnos, los primeros podrán crear actividades y podrán utilizarlas para el desarrollo educativo; por otra parte los alumnos podrán utilizar estas actividades para reafirmar el conocimiento adquirido en clase.

Cabe señalar que este modelo de software no es un sistema el cual compita o resuelva los problemas enseñanza aprendizaje y que solamente ayuda a facilitar la comprensión y la asimilación de la información como apoyo al docente.

CAPITULO 1

Tecnología y

Educación

1. TECNOLOGÍA Y EDUCACIÓN

La Educación debe ligarse necesariamente con la dinámica de cambio y adaptación constante, ya que nuestro entorno evoluciona a pasos agigantados en materia científico-cultural y desarrollo tecnológico, ya que cada vez crecen las necesidades, intereses sociales e individuales y el mundo del trabajo.

Por lo tanto, la educación debe responder a demandas y necesidades complejas y diversas; por lo que requiere de formas y planteamientos no tradicionales y susceptibles de actuar con prontitud en esa amplia realidad.

Considerando lo anterior el sector educativo ha tomado en cuenta las aportaciones de las Nuevas Tecnologías que son un conjunto de herramientas, soportes y canales para el tratamiento y acceso a la información; toda vez que ofrecen disponibilidad, potencialidad, la posibilidad de conseguir nuevos y más variados objetivos y versatilidad en los procesos de formación.

Desde esta perspectiva al intervenir las nuevas tecnologías en la educación, éstas contribuyen a apoyar instrumentalmente los programas y procesos de formación.

1.1 POR QUÉ USAR LOS MEDIOS TECNOLÓGICOS EN LA EDUCACIÓN

"Todos los recursos didácticos que han aparecido en los dos últimos siglos, desde libros de texto y pizarras a proyectores de diapositivas, vídeos y ordenadores, se reúnen ahora en una sola estación de trabajo interactiva. Las clases de mañana verán estaciones de presentación interactivas unidas a redes de más amplia área que harán llegar a los estudiantes, audio, vídeo y datos tanto en el lugar de estudio como fuera de él. La utilización de distintos canales permite al profesorado tener en cuenta los diferentes estilos cognitivos".¹

Por lo tanto, el software educativo alienta la exploración, la auto expresión y un sentido de dominio al permitir a los estudiantes manipular sus componentes. Los entornos multimedia activos favorecen la comunicación, la cooperación y la colaboración entre el profesor y el alumnado. El multimedia hace el aprendizaje estimulante, atractivo y divertido.

1.2 IMPORTANCIA Y CARACTERÍSTICAS DE LA TECNOLOGÍA APLICADA A LA ENSEÑANZA

Los medios tecnológicos físicos e intangibles, tales como las redes y el software educativo, han sido entendidos como instrumentos de ayuda a la comunicación y a la formación. Por tanto, son de naturaleza constructiva del aprendizaje.

¹ Multimedia y el proceso de enseñanza-aprendizaje en la Educación Superior (Multimedia and the Teaching-Learning Process in Higher Education), Lamb J.A. 1992.

Lógicamente su importancia va a depender tanto del alumno, como del tipo de actividad, contenido u objetivo pretendidos. Por lo que, al elegir o diseñar una tecnología educativa hay que considerar las actividades mentales que estimulan la facilidad para transmitir información, la capacidad para satisfacer la exigencia de la actividad didáctica y para conectar con las características cognitivas del alumno.

Una recomendación al crear un medio o recurso tecnológico aplicado a la educación es establecer su sentido en el contexto formativo, esto es, en establecer su papel en relación con las necesidades, objetivos, contenidos, actividades, tipo de alumno, estructura de relaciones profesor-alumno, etc, y es que los medios tecnológicos educativos, por sí solos, no mejoran la enseñanza o el aprendizaje. Lo harán en la medida en que se hayan seleccionado adecuadamente y con funcionalidad respecto a las necesidades del proceso de enseñanza-aprendizaje en el que se hayan de instalar y al que han de servir.

La situación actual se caracteriza por la fuerza de relación y de complementación que se da entre Tecnología y Enseñanza, y es que los nuevos medios tecnológicos están logrando no sólo cambiar los sistemas de relación ser humano-medio, sino también instalarse como componente cultural

En general, las funciones que deben desempeñar los medios tecnológicos en una situación didáctica son:

1.-INNOVACIÓN: En este sentido un medio genera cambios en el modelo de enseñanza y en el tipo de aprendizaje, pero también puede actuar como reforzador de un modelo previo o generar cambios superficiales.

2.-MOTIVACIÓN: La medida en que refuerza una situación educativa y puede hacerlo de una forma más atractiva.

3.-ESTRUCTURADORA DE LA REALIDAD (Generador del conocimiento): Son una guía metodológica que apoya la actividad docente y discente estableciendo un tipo de relación con la enseñanza y el aprendizaje.

4.-FORMATIVA: Pueden condicionar el tipo de aprendizaje, esto es, provocan un tipo de actividad mental en los alumnos, ayudan a transmitir determinados tipos de valores en la medida en que apoyan la presentación de contenidos, guían, facilitan y organizan la acción didáctica entre docente y alumno.

5.-SOLICITADORA U OPERATIVA: organizan las experiencias de aprendizaje a la vez que establecen un tipo de contacto con la realidad que representan y exigen un tipo de contacto con ellos mismos.

1.3 SELECCIÓN DE LA TECNOLOGÍA INFORMÁTICA ADECUADA.

La tecnología es inmensa, podemos contemplar el mundo de las comunicaciones como son las redes o satélites² por mencionar algunos o el aspecto intangible pero no menos importante, es decir, el software, ambos aspectos son complementarios y aplicables a la educación.

Su valor va a depender, fundamentalmente del contexto e intencionalidad metodológica. Sus cualidades y posibilidades van a permitir solamente establecer las virtudes funcionales del medio en la estrategia didáctica elegida. Lo que debe quedar claro es que la tecnología utilizada nunca será la clave en la decisión de la estrategia metodológica a usar en el proceso de enseñanza-aprendizaje.

De un modo específico, podríamos establecer las siguientes dimensiones de análisis para la valoración de las tecnologías:

1. Virtudes intrínsecas: Valorar en primer lugar las características y prestaciones que por sí mismo puede ofrecer un determinado medio o recurso.

2. Funcionalidad curricular: En este caso se trataría de valorar el grado o nivel de adaptación de un medio a la situación didáctica, que sea compatible e idóneo a las condiciones de la situación de enseñanza y aprendizaje a la que debe de servir. Y el grado de adecuación a los sujetos de aprendizaje y a la secuencia didáctica.

3. Costo y rentabilidad: El costo del mantenimiento, las exigencias de equipamientos complementarios además de la eficacia y rentabilidad didáctica han de ser aspectos a considerar para determinar su uso.

Cabe destacar que por cuestiones del objetivo del presente documento, nos enfocaremos a la explicación de las tecnologías informáticas intangibles o software aplicado a la educación.

1.4 SOFTWARE EDUCATIVO Y OTROS MEDIOS TECNOLÓGICOS APLICADOS A LA EDUCACIÓN.

El software educativo es una herramienta que tiene como función principal ayudar al estudiante a obtener o reafirmar conocimientos y habilidades.

Las características comunes son:

- Inmaterialidad
- Interactividad

² En México se cuenta con RED EDUSAT.

- Instantaneidad
- Innovación

Fundamentalmente su potencialidad para la enseñanza radica en su capacidad de:

- Secuenciar las tareas de aprendizaje.
- Respetar el ritmo individual de cada alumno.
- Favorecer la participación y actividad del alumno.
- Favorecer las respuestas.

No podemos dejar de considerar, por otra parte, que las Nuevas Tecnologías, dada la situación en que nos movemos, se plantean como un hecho trascendente y apremiante. En primer lugar derivan y provocan una aceleración en los cambios, avances científico-técnicos y en segundo lugar, porque paradójicamente, provocan cambios de todo tipo en las estructuras sociales, económicas, laborales e individuales.

Software de formación local o específico.

Entenderemos por "software de formación local o específico" al medio cuya acción se circunscribe a aquello para lo que están diseñados. Como ejemplos de ellos podríamos citar:

a) El caso de la "Simulación" en la enseñanza.

Una simulación es la producción de la apariencia de algo real y/o de sus efectos, cuando ese producto puede ser manipulado y se asemeja a la situación real que representa.

La simulación mejora:

- La motivación.
- La comprensión de los mecanismos que rigen el proceso o realidad que se simula.
- La capacidad de transferencia y aplicación de los conocimientos.
- Las actitudes positivas hacia el proceso de aprendizaje.

La simulación permite:

- Aproximar la realidad a aprender con la posibilidad de "intervenir" en ella.
- La investigación con esa realidad en condiciones que no podrían asumirse de modo real.
- Operar con las variables que componen la realidad "simulada", de tal modo que se creen "realidades virtuales".

b) Enseñanza Asistida por Sistemas Tutoriales "Inteligentes" (I.T.S.)

Su acción se basa en el diseño de lo que se denomina "**entornos reactivos**" y que actúan en base a los siguientes componentes:

Modelo de alumno: "El sistema necesita un modelo de los conocimientos del alumno sobre el contenido, incluidas sus falsas concepciones, para orientarle convenientemente en el aprendizaje. Se trata de conocer que errores se comenten y por qué, para promover estrategias de aprendizaje que se anticipen a ellos"³.

Modelo de Profesor: El sistema "simula" la conducta de un profesor experto. En base a dichos componentes se suelen usar estrategias de este tipo:

- Proporcionar orientación al alumno sobre la forma de resolver un problema, modificando el entorno para que se puedan ensayar distintas alternativas de solución.
- Formular preguntas al alumno que le ayuden a razonar y a formular o modificar sus propios conceptos. Para ello se suelen incluir simulaciones y juegos a través de los que el alumno puede descubrir hechos y leyes.
- Proporcionar tareas para evaluar respuestas y detectar concepciones erróneas.

c) Los Sistemas Expertos

Los sistemas expertos permiten asistir en el análisis y resolución de problemas complejos. Intentan simular la conducta de un experto humano en un dominio específico del conocimiento.

Un sistema experto debe contener, la información que manejaría un experto en un área específica y además, la forma de razonamiento que utiliza, esto es, la forma en que el experto hace funcional el conocimiento.

d) El Vídeo Interactivo

El Vídeo Interactivo es un sistema que se ha convertido en un medio clave para el tratamiento de la información y para su presentación a los usuarios, pues combina las posibilidades del vídeo convencional con las virtualidades de la informática, lo que permite el tratamiento conjunto de imágenes, sonido, texto y gráficos y la **interactuación** de dicho usuario a través de un ordenador, que, en la práctica, se erige en el elemento de control del sistema.

Su característica distintiva es la **interactuación**, esto es, la relación que el receptor de la información puede establecer con dicha información. Debido a ello,

³ Enseñanza y aprendizaje en la educación superior. Julián López Yáñez, Araceli Estebaranz García, Carlos Marcelo García 1990

su virtualidad formativa es muy alta, dada su capacidad para hacer participar al alumno y dado su nivel de motivación, eficacia y "realismo".

e) Hipertexto o Hipermedia

Las aplicaciones Hipertexto o Hipermedia permiten combinar información (texto, gráficos, tablas, imágenes, sonido), de tal modo que el usuario puede pasar de un concepto a otro o de un documento a otro a través de una estructura que, a modo de red, permite ir construyendo la información requerida por el usuario en un determinado dominio.

Medios de formación abierta

Se establecen como una modalidad de enseñanza a distancia, pero desde una perspectiva distinta, usando a la vez estrategias de enseñanza a distancia y presencial, por lo que se vienen entendiendo como estrategias para el aprendizaje abierto, tal y como establecieron Lewis y Spencer en 1986.

Se caracterizan por:

- Adaptarse y ayudar al usuario.
- Proporcionar posibilidades de comunicación con otras personas.
- Proporcionar acceso a los materiales de aprendizaje.
- Proporcionar ayuda en el currículo.
- Proporcionar las funciones de seguimiento y tutorización.

Son considerados medios de formación abierta: La video-conferencia, el correo electrónico, (con sus listas de discusión), las Conferencias electrónicas, Mensajería, Hipertexto en red, El trabajo cooperativo, el Videotex, etc.

a) Sistemas de información general, profesional y ocupacional

Los sistemas que colocan la información al alcance de todo tipo de usuario en donde tiene acceso a las posibilidades de formación como a las vías de acceso a ella.

b) Sistemas de Información y Formación (La Red INTERNET)

INTERNET es la red de redes, la conexión "total" entre computadoras a través de un protocolo o idioma común (TCP/IP: *Transmission Control Protocol / Internet Protocol*), de un modo "transparente", de tal manera que el usuario no aprecia ni necesita los recursos necesarios para la comunicación.

Los servicios más frecuentes de INTERNET son:

- **E-Mail:** Correo electrónico.
- **E-Conferences:** Derivadas del E-Mail y que constituyen toda una red de servicios de información.
- **FTP (File Transfer Protocol):** Intercambio de ficheros a través de la red.
- **Transferencia de archivos p2p.** Intercambio de archivos punto a punto.
- **Transmisiones de voz por I.P.** Se pueden realizar llamadas telefónicas por solo conectándose aun nodo con internet.
- **Blogs.** Es una especie de bitácora en al cual un autor o autores recopilan cronológicamente textos o artículos.
- **Por mención algunos.**

INTERNET contiene herramientas que incorporan la posibilidad de trabajar hipertextos o hipermedia, como el WWW, bases de datos, libros y revistas, etc. de dominio público o restringido.

Software para desarrolladores.

El "Software para desarrolladores" proporciona opciones con las cuales los profesores y público en general, interesado en el desarrollo de material educativo (de apoyo, cursos en línea o multimedios) pueden llevar a cabo la producción de forma rápida facilitando sus tareas en el desarrollo.

1.4.1 E – LEARNING

E-learning significa literalmente aprendizaje electrónico. Constituye una propuesta de formación que contempla su implementación predominantemente mediante internet, haciendo uso de los servicios y herramientas que esta tecnología provee.⁴

Dentro de la modalidad a distancia, el *e-learning* es una de las opciones que actualmente se utiliza con mayor frecuencia para atender la necesidad de educación continua o permanente. La generación de programas de perfeccionamiento profesional no reglados está en crecimiento debido a que existe un reconocimiento de que los trabajadores se capaciten y se adapten a los nuevos requerimientos productivos. El e-learning, dadas sus características y el soporte tecnológico que lo respalda, se constituye en una alternativa para aquellos que combinan trabajo y actualización, ya que no es necesario acudir a una aula permanente.

Si la educación a distancia es, desde sus orígenes, una opción para atender la formación de personas adultas, el e-learning tiene la ventaja de que los usuarios

⁴ <http://es.wikipedia.org/wiki/E-learning>

eligen sus propios horarios, y puede entrar a la plataforma desde cualquier lugar donde puedan acceder a una computadora y tengan conexión a internet.

1.4.2 M – LEARNING

Se denomina **aprendizaje electrónico móvil**, en inglés, **m-learning**, a una metodología de enseñanza y aprendizaje valiéndose del uso de pequeños y maniobrables dispositivos móviles, tales como teléfonos móviles, celulares, agendas electrónicas, tablets PCs, pocket pc, i-pods y todo dispositivo de mano que tenga alguna forma de conectividad inalámbrica.

La educación va incorporando intensivamente las nuevas tecnologías de la comunicación, pasando por varias etapas. Diversos conceptos describen ese fenómeno, según avanza la tecnología: EAO (Enseñanza apoyada por el ordenador), multimedia educativo, tele-educación, enseñanza basada en web (web-based teaching), aprendizaje electrónico (e-learning), etc.

Tanto desde el simple uso de la computadora y los soportes multimedias, como el advenimiento de Internet y las redes en general, todo ha servido para apoyar el proceso de enseñanza-aprendizaje en sus diferentes modalidades y aspectos.

De un tiempo a esta parte, se vienen incorporando a nuestras vidas, cada vez con más fuerza, las tecnologías móviles, y por lo tanto, está surgiendo lo que denominamos **mobile learning** o m-learning y que consiste en usar estos aparatos electrónicos para aprender.

Esto está generando gran expectativa en el sistema educativo, sobre el que se están realizando interesantes iniciativas empresariales y proyectos de investigación

1.5 EVALUACIÓN DE SOFTWARE EDUCATIVO.

La evaluación de software educativo se ha centrado tradicionalmente en dos momentos:

- 1. El desarrollo:** Con el fin de corregir y perfeccionar el programa.
- 2. El uso de este tipo de materiales:** Para juzgar su eficiencia y los resultados que con él se obtienen.

El primer tipo de evaluación es el más frecuente: los diseñadores de software mantienen como mecanismo necesario la evaluación del proceso de diseño y producción. Evalúan los aspectos computacionales y algunos de los referidos a la relación hombre máquina y a la facilidad o funcionalidad de uso del programa. No es aun frecuente la participación de un evaluador educativo o pedagógico en estos procesos; aunque es cierto que los temas de la relación hombre-máquina y del

usuario como criterio de calidad, están cobrando gran fuerza en el ámbito de la ingeniería de software.

Se trata de pruebas en situación real o muy similar, con el fin de incorporar cuando todavía hay tiempo, las mejoras que una experiencia de uso real haga aconsejables. Estas pruebas suelen tener lugar en centros universitarios de producción de software educativo.

Objetivo de la evaluación.

Se establece como finalidad de la evaluación de software educativo: orientar un uso pedagógicamente adecuado.

Esto supone un cambio de orientación a las tres prácticas descritas. No se evalúa para hacer ajustes durante el proceso de diseño o desarrollo, puesto que en su mayoría se trata de software editado en versión final por casas comerciales. No se trata de una evaluación de prueba, con el fin de convalidar el programa terminado en situación real, antes de la edición definitiva. Tampoco se pretende realizar evaluaciones de campo, una vez esté el programa en uso, se evalúa el uso real, pero no con intenciones de calificar el software, sino para revisar y completar las guías de uso y como un elemento más de la evaluación global de las acciones del proyecto Conexiones.

La finalidad, entonces, es ayudar al usuario, sobre todo al docente, en el uso del programa, haciendo énfasis en los aspectos pedagógicos, metodológicos, ideológicos y culturales que, de todas formas, contiene.

1.5.1 EVALUACIÓN DEL PROGRAMA POR LA FUNCIONALIDAD EN SU USO.

a) El programa como objeto material.

Se examinan dos aspectos:

- **Equipo requerido:** Descripción de los requerimientos de equipo mínimos que exige el programa para funcionar; esta información suele aparecer en los folletos que acompañan al CD, medio de almacenamiento ya usual. En la guía de uso se completan los aspectos que hacen referencia a las condiciones de instalación de las instituciones participantes en el proyecto.
- **Utilidad:** Medida en que el sistema es fácil de aprender y fácil de utilizar.

A continuación se examinan algunos aspectos de *utilidad*:

b) Facilidad de aprendizaje.

Medida en que el usuario novel comprende cómo utilizar inicialmente el sistema y cómo a partir de esta utilización llegar a un máximo nivel de conocimiento y uso del sistema.

Por lo que debe ser:

- **Predictivo:** los conocimientos adquiridos por el usuario son suficientes para poder determinar los resultados de sus futuras interacciones
- **Sintetizable:** habilidad del usuario para evaluar los efectos de las operaciones anteriores al estado actual (capacidad de captar los cambios de estado que produce cada operación)
- **Familiar:** correlación entre el conocimiento que tiene el usuario y el conocimiento que necesita para una interacción efectiva
- **Consistente:** medida en que todos los mecanismos son usados siempre de la misma manera.

c) Flexibilidad.

Multiplicidad de formas en las que el usuario y el sistema intercambian información

- **Iniciativa de diálogo:** quien tiene la iniciativa en la conducción del diálogo; hay o no libertad para iniciar cualquier acción en el sistema
- **Diálogo multi-hilo:** un hilo de un diálogo es un subconjunto coherente del mismo. Si el sistema soporta diálogos multihilo al mismo tiempo.
- **Migración de tareas:** Transferencia del control, del sistema al usuario y viceversa, para la ejecución de tareas: medida en que se puede pasar de una tarea a otra, pasar una a segundo plano o repartirse entre ambas
- **Adaptabilidad:** Si el sistema puede adaptarse a distintos usuarios

f) Solidez.

Características de la interacción que permiten lograr los objetivos, y su asesoramiento

- **Recuperabilidad:** Posibilidad del usuario para corregir una acción una vez ha reconocido un error
- **Tiempos de respuesta:** tiempo que necesita el sistema para expresar los cambios al usuario
- **Adecuación a las tareas:** en qué grado los servicios del sistema soportan todas las tareas que el usuario quiere hacer y la manera en que el usuario las comprende.

g) Mecanismos de soporte.

Recursos de ayuda y forma en que el usuario puede utilizarlos.

- **Disponibilidad:** posibilidad de consultar la ayuda en cualquier momento, sin tener que salir de la aplicación
- **Precisión y detalle:** medida en que la ayuda cubre todo el sistema, con concisión
- **Consistencia:** en términos de contenidos, terminología y estilo
- **Robustez:** que soporte más que el sistema, en términos de funcionamiento
- **Flexibilidad:** en que medida permite interactuar de manera adecuada a las necesidades del usuario
- **No obstructiva:** que no impida el uso normal de la aplicación
- **Organización del texto de ayuda:** lenguaje, longitud de frase y párrafo; cantidad de texto; espacios en blanco; gráficos e iconos.

1.5.2 EVALUACIÓN DEL EL PROGRAMA COMO OBJETO PEDAGÓGICO.

Para realizar una evaluación correcta del programa como objeto pedagógico se tienen que revisar:

a) Contenido Científico.

Se trata de evaluar la calidad y cantidad de la información ofrecida:

- **Exactitud, actualidad:** Fechas de edición; referencias o fuentes citadas; términos técnicos; datos estadísticos. Visión de Ciencia; visión de tecnología.
- **Adecuación:** valor absoluto: significatividad de los contenidos en sí mismos; valor relativo: adecuación en nivel de tratamiento a la situación pedagógica dada.

b) Contenido socio-cultural e ideológico.

- **Visión sociocultural:** a qué grupos sociales (o culturales) se refieren los ejemplos, los personajes, los problemas planteados. Qué muestran las ilustraciones: representación racial, género, referencias geográficas, etc.
- **Personajes:** reales, imaginarios; sexo; edad; raza; nacionalidad; condición o estado, patronos, obreros, campesinos, militares...
- **Marcos espacio-temporales:** contexto geográfico (urbano, rural, mar, montaña); medio de referencia (flora, fauna, estaciones); épocas de referencia; medio tecnológico y objetos de la vida cotidiana

- **Contexto social:** representación del trabajo; categorías socioprofesionales representadas; familia (composición); habitación (casa, cabaña, finca, conjunto urbano...)
- **Situaciones y temas:** vida cotidiana (en la casa en la escuela, en el trabajo); situaciones excepcionales (crisis; héroes)
- **Ideología implícita:** justicia y autenticidad (presentación de los hechos sin distorsión y en perspectiva).
- **Valores:** contribución a la paz, a la tolerancia, a la formación de actitudes culturales y ecológicas

c) Contenido pedagógico.

Se trata de determinar la adecuación pedagógica de los objetivos y contenidos, frente a los usuarios, su nivel y el programa que están desarrollando.

- **Intenciones formativas:** lo que pretende el programa, los objetivos de aprendizaje que persigue, explícita o implícitamente
- **Conocimientos previos:** si los usuarios dominan los conocimientos previos, en caso que el programa los requiera
- **Niveles de aprendizaje:** qué niveles de aprendizaje (hechos, conceptos, principios, habilidades valores) pretende desarrollar el programa
- **Organización:** la progresión del aprendizaje responde a qué tipo de secuencia pedagógica: rígida, espiral o controla por el usuario. En este caso, ¿son necesarias instrucciones o de progreso o es preferible que el usuario encuentre sus propias secuencias?
- **Adecuación curricular:** los objetivos y contenidos del programa se pueden integrar con facilidad al curriculum vigente
- **Organizadores y autoevaluación:** contiene síntesis (resúmenes), ejercicios (con o sin respuesta), complementos informativos. Contiene evaluaciones, auto evaluaciones respuestas razonadas, refuerzo, sistema de seguimiento de logros, evaluación sumativa

d) Comunicación.

Se trata de evaluar la forma del mensaje (significante), es decir el conjunto de recursos que permiten transmitir un mensaje de un emisor a un receptor.

- **Sentido de la comunicación:** dirección y control de la interacción programa-usuario Unidireccional, bidireccional, control del usuario sobre la secuencia, multitareas, multivías...
- **Formas del mensaje.:** los aspectos formales de los códigos elegidos (text, audio, fotos, animación, gráficos, colores) se justifican en sí y frente a la función que se espera de ellos.

e) Método.

Qué metodología, implícita o explícita, contiene para la exposición de las ideas, la organización del trabajo, las formas de uso que determina.

- **Secuencias:** se componen de una serie de partes que están presentes regularmente
- **Estructura:** el programa es un elemento de enseñanza, de aprendizaje o de enseñanza-aprendizaje.
- **Guías o manuales:** el programa viene acompañado de un manual para el maestro, el alumno, el usuario en general.
- **Elementos de organización interna:** el programa incluye instrucciones de empleo, índices, objetivos, léxico, preguntas/ejercicios/, respuestas razonadas, recapitulaciones, evaluaciones
- **Facilitadores:** modo de empleo, índice de materias, lista de objetivos, léxico, referencias, fuentes, plan de capítulos, resúmenes, preguntas, ejercicios, tareas, correcciones control de logro, llamadas.
- **Papel del maestro:** se limita a dar instrucciones de uso; es necesario para complementar, aclarar o integrar la información; es hacer un seguimiento del uso y de los logros del estudiante
- **Exigencias de aprendizaje:** el programa exige principalmente (con mayor frecuencia, como acciones centrales) al estudiante acciones y habilidades para: memorizar información, construir conceptos, seguir instrucciones, construir secuencias aprendizaje propias, hacer preguntas, construir respuestas originales, relacionar lo aprendido con otros conocimientos, colaborar con compañeros.
- **Distribución de tiempos:** un estudiante típico, en una sesión de trabajo normal con el programa, distribuye su tiempo en (% aprox.) : aprender a navegar y buscar información desplazándose por el programa, leer texto, escuchar narración; plantear preguntas al programa; responder preguntas, realizar tareas o ejercicios.

f) Adaptabilidad.

- **Limitaciones metodológicas:** El programa impone un método al docente, o éste tiene opción de escoger objetivos, ritmos de trabajo, secuencias.
- **Limitaciones para el alumno:** El programa ofrece diferentes maneras de entrada; ofrece ejercicios diferentes y graduados según el nivel de los alumnos; posibilidades diferentes de utilización, de acuerdo con las necesidades e intenciones del usuario.

1.6 EL IMPACTO DE LAS NUEVAS TECNOLOGÍAS INFORMÁTICAS EN ESTUDIANTES Y DOCENTES.

El final de este siglo se ha caracterizado por una serie de necesidades y relaciones que el hombre ha establecido con productos tecnológicos derivados de la propia dinámica del desarrollo técnico y científico.

Este desarrollo, a través de las Nuevas Tecnologías ha dado respuesta a necesidades muy concretas de información, comunicación, productividad y creación de nuevos códigos que permiten acceder a un lenguaje y un conocimiento cada vez más universal e interactivo.

La evolución que éstos han tenido a lo largo del tiempo y que han dado origen a sistemas tecnológicos cada vez más sofisticados, involucrando la participación interactiva del hombre en las diferentes áreas que circunscriben su vida, es lo que realmente nos permite hablar de nuevas tecnologías

Dos de ellas, la comunicación e informática ha facilitado como nunca antes el acceso a la información y a la cultura; de esta forma, la concepción de la "aldea global" permite, hoy en día, que los conocimientos y la información sean accesibles a todas las personas. Este hecho posibilita la retroalimentación y la interacción entre diferentes culturas y proyectos en desarrollo, permitiendo igualmente, la homogeneización de la información y la participación abierta a diversas tareas relacionadas con la acción del hombre por el hombre.

Integración

Esta característica se refiere a la posibilidad que tienen las tecnologías, que inicialmente se desarrollaron aisladas, de integrarse para aumentar, flexibilizar y diversificar sus posibilidades de utilización: televisión y satélite, computadora y teléfono, teléfono y satélite etc.

Interactividad

Este es un término que se ha utilizado en el terreno comunicacional y educativo, en el sentido de "diálogo" que puede tener una persona con un interlocutor sea éste una persona o un medio y generalmente se conoce como interacción.

Esta interactividad también puede darse al interior de los propios programas de software, cuando estos tienen la posibilidad de que el usuario se retroalimente o navegue con la flexibilidad que le ofrece el propio programa.

Contextualización

Se refiere a la posibilidad que tiene el propio desarrollo tecnológico para formar parte del proceso de globalización de la economía y la cultura, y correr paralelamente en la complejidad de las relaciones sociales.

1.6.1 APLICACIONES DIDÁCTICAS DE LA COMPUTADORA EN EL AULA

La transmisión de un contenido de una manera clara, coherente e ilustrativa ha sido desde siempre la tarea principal de la didáctica.

Las potencialidades de la tecnología han auxiliado de una manera importante a la didáctica en esta tarea,

Justo éstas son las potencialidades de la tecnología que permiten facilitar la comprensión de contenidos a través de imágenes, audio, simulaciones y efectos en tercera dimensión.

Diversos estudios psicológicos señalan que el aprendizaje de un niño es mayor cuando se estimula la mayoría de sus sentidos, con un proceso rápido y variado, es por eso que gustan de la televisión y de los videojuegos, ya que éstos presentan imágenes, colores y sonidos que llaman su atención y estimulan más de uno de sus sentidos.

Las potencialidades técnicas que presenta la computadora (audio, imagen, video, efectos especiales, hipertexto y simulaciones mismas) tienen que ver con ciertas capacidades cognitivas del alumno.

Audio

Exige del estudiante, cognitivamente hablando: procesos de identificación, comparación (con una experiencia previa) análisis y síntesis de lo que se está escuchando.

Imagen

La imagen capta la atención del estudiante a través de la vista. La imagen permite que el alumno se forme un referente conceptual más claro acerca de lo que está viendo, aunque también supone una interpretación acorde a una experiencia previa y un impacto positivo o negativo acorde a la personalidad del sujeto.

La imagen en la computadora generalmente es empleada de múltiples maneras: para conceptualizar o aclarar un concepto.

Video

En términos educativos esta potencialidad es importante porque muestra al alumno la imagen de personajes o acontecimientos que tal vez no se habían visto antes, lo cual amplía su percepción o conocimiento sobre algún suceso histórico.

Hipertexto

El hipertexto, es la escritura no secuencial a un texto que bifurca, que permite que el lector elija y que se lea mejor en una pantalla interactiva.

Simulación

Un programa de simulación pretende presentar de forma artificial una situación para que el alumno actúe como si se tratara de una realidad. Se emplea generalmente para evitar riesgos y costos, por lo que educativamente es muy valioso, ya que el alumno aplica conocimientos sin tener ningún riesgo personal.

Existen múltiples programas de software estructurados bajo una lógica determinada, lineales o con posibilidades de navegación que apoyan una materia o área específica. Quizá los softwares más abundantes son los de matemáticas, física y química, ya que se considera que son las materias más difíciles para el alumno, y por ello se tiene que trabajar el contenido de una manera más comprensible para su asimilación.

Otros apoyos que se han utilizado con estas características de auxiliar didáctico ha sido el CD ROM, que es un disco compacto con gran capacidad de almacenamiento de información, por lo que resulta ideal como soporte del software multimedia, esto es integración de audio, video y animación.

1.6.2 ACTITUDES HACIA LAS NUEVAS TECNOLOGÍAS EN LA EDUCACIÓN

Como hemos visto, las nuevas tecnologías de comunicación e información han abierto al mundo un sin número de opciones que permiten lograr una mayor eficiencia en todos los ámbitos en los que se aplican. Sin embargo, no podemos negar que, si bien las nuevas tecnologías han favorecido en gran medida distintas actividades del ser humano, igualmente su surgimiento ha traído consigo ciertas posturas de rechazo, al ser considerado su uso como enajenante, individualista, masificante y destructivo.

Al paso del tiempo estas posturas negativas que se presentaron de manera más generalizada en un primer momento, se han ido sustituyendo por expectativas positivas que los resultados de sus diversas aplicaciones han generado. El medio educativo ha sido un claro reflejo de esto.

Para la teoría de la Acción Razonada una actitud es "Una predisposición aprendida a responder de manera consistentemente favorable o desfavorable con respecto a un objeto dado"⁵. Con respecto al Modelo planteado por Ajzen y Fishbein sobre la teoría de la Acción Razonada, los humanos hacen un uso sistemático de la información de la que disponen, de modo que antes de conducirse o no de cierto modo consideran las implicaciones de sus acciones usando la información a su disposición, hacen juicios, forman evaluaciones hasta llegar a una decisión; esto lo hacen con base en sus creencias.

Así, la totalidad de las creencias sirve como base de información que determina sus actitudes, intenciones y conductas.

De esta forma, de acuerdo con el modelo de la teoría de la Acción Razonada las actitudes que una persona pueda tener poseen tres componente básicos: uno afectivo, otro cognoscitivo y un último conductual (Gagné, 1987; Reich y Adcock, 1980, Sarabia, 1992; Javiedes, 1996).

- El componente afectivo de la actitud incluye la dirección e intensidad como características afectivas que implican la reacción del individuo hacia el objeto actitudinal como un todo y las reacciones a cada uno de los atributos que pueda poseer.
- El componente cognoscitivo está formado por el contexto informativo y la perspectiva temporal de la actitud. El primero se refiere al conjunto de creencias estereotipos y conocimiento factual que la persona posee con respecto al objeto actitudinal; la perspectiva temporal se refiere al grado en el cual el desarrollo futuro del objeto está integrado a la perspectiva actitudinal presente.
- El componente conductual describe el curso de la acción que podrá tomar el individuo respecto a la actitud.

La transformación que producen las innovaciones tecnológicas en la educación debe ser introducida por un cambio voluntario de los docentes y no sólo por la propaganda de expertos o por exigencias de decretos gubernamentales.

El uso de la computadora como un elemento más en la metodología de la enseñanza, implica un cambio en las funciones de los maestros; cambio que puede crear una ruptura en la concepción armónica que tienen de su papel en el salón de clases generando así una actitud negativa hacia el objeto que transgrede, y por lo tanto una resistencia hacia su introducción.

⁵ Understanding attitudes and predicting social behavior. Actitudes de comprensión y predicción de comportamiento social), Ajzen, I., & Fishbein, M. Editorial Prentice-Hall. 1980.

En nuestro país, el maestro promedio se ve forzado a adoptar la tecnología para "no quedarse atrás" y competir laboralmente, soportando diversas presiones de carácter grupal e institucional que se traducen en incentivos educativos y laborales de diversa índole. El lado amable de estas presiones lo constituye una incipiente cultura informática que impulsan los corrillos de profesores interesados, en una labor de persuasión y apoyo mutuo en el quehacer cotidiano.

La introducción de la computadora al salón de clases, exige a los maestros un esfuerzo mayor del acostumbrado y una evaluación de sus capacidades frente a un aprendizaje que implica la incorporación de un lenguaje nuevo, la adquisición y asimilación de conocimientos y habilidades que no guardan una relación estrecha con su desempeño cotidiano e inclusive un cierto manejo de ansiedad y frustración frente a la tecnología.

1.7 TECNOLOGÍAS APLICADAS A LA EDUCACIÓN EN MÉXICO

En la actualidad, la integración de las nuevas tecnologías de información, su impacto, la comunicación y la competencia con otras culturas son elementos propulsores hacia el cambio de paradigmas, por ello, en México se han implantado diversas estrategias en materia de educación como es el programa Enciclomedia, implantado a nivel primaria.

Programa Enciclomedia.

Enciclomedia es un sistema educativo concebido y aplicado en México por la S.E.P. para niveles de educación básica, que integra herramientas y medios a fin de enriquecer la enseñanza y aprendizaje en el salón de clases. Operará en 165 mil 615 aulas, de 5° y 6° grados, en el territorio nacional.

Enciclomedia contiene trece libros digitalizados de quinto y sexto grados de educación primaria, enlaza a la biblioteca de aula, a fotografías, mapas, visitas virtuales, videos, películas, audios, interactivos y otros recursos y como apoyo docente se incorporará la edición digital de cuatro ficheros de matemáticas y español, y ocho libros para el maestro, Es también un espacio para la colaboración organizada donde maestros, instituciones públicas, privadas e individuos podrán proponer materiales y recursos. Incluye 1 computadora personal, 1 proyector, 1 impresora, 1 fuente de poder, 1 mueble para computadora y 1 pizarrón interactivo o antirreflejante.

Convenio SEP/Microsoft

Otra de las estrategias del gobierno federal ha sido el convenio SEP/Microsoft.

La Secretaría de Educación Pública y Microsoft-México, con el propósito de dotar a escuelas de educación básica con los programas de cómputo necesarios para mejorar las habilidades del proceso de enseñanza-aprendizaje, firmaron un amplio acuerdo de cooperación tecnológica denominado "Alianza por la Educación",

Con una estrategia a cinco años, se estima que este acuerdo impactará a un universo de más de 125 mil planteles de educación básica, mediante la entrega de los programas más avanzados de Microsoft a cerca de 200 mil computadoras.

Este es un importante paso hacia la modernización y uso de la tecnología, así como en lo referente a capacitación y actualización de profesorado y estudiantes, donde se calcula que durante este periodo se llegará a más de 800 mil profesores y más de 20 millones de alumnos para desarrollar todo su potencial a través del acceso y la capacitación en tecnologías de cómputo.

La iniciativa Alianza por la Educación busca invertir en el futuro resolviendo prioridades locales en educación y haciendo que el software sea accesible para estudiantes y maestros que lo requieran y que no puedan costearlo de otra manera

En esta iniciativa, Microsoft participará con tres proyectos distintos: Learning Grants (fondos para el desarrollo de estrategias de este programa), Fresh Start (software para PCs donadas al sector educativo) y School Agreement (acuerdos locales de licenciamiento con instituciones académicas), lo que representa una importante inversión económica encaminada a impulsar la calidad de la educación en México

El programa de Fresh Start, como parte de la iniciativa de Alianza por la Educación, juega un papel importante porque ayudará a las escuelas a sacar el mayor provecho de las computadoras donadas a este sector, promoviendo el acceso de los estudiantes a las nuevas tecnologías mediante la implementación más difundida de PCs en los salones de clase.

Las computadoras donadas recibidas por las escuelas cuentan originalmente con una licencia de Windows, además del programa de Fresh Star que proporciona una licencia de Windows 98 o Windows 2000 para estas PCs (Pentium II o anteriores) en escuelas de educación básica sin cargo adicional.

Alianza por la Educación abre la posibilidad de contar con descuentos educativos muy superiores a los ya de por sí precios rebajados del software de Microsoft para áreas educativas, a través del programa de licenciamiento por volumen para el sector académico

En el convenio de concertación que realizaron la S.E.P. y MICROSOFT en su cláusula PRIMERA, punto numero 4 la SEP se compromete a :

“Modernizar la infraestructura tecnológica de las instituciones educativas de la SEP, considerando las soluciones que ofrece Microsoft ®, de conformidad con las disposiciones legales aplicables.”

Y en el mismo documento Microsoft se compromete a otorgar gratuitamente las licencias de uso para Windows 98 y Windows 2000 a las instituciones educativas que reciben la donación de computadores Pentium II o anteriores, asumiendo de

buena fe que dichas computadores se encontraban originalmente licenciadas con el sistema operativo Windows, pero que las licencias no fueron donadas con las computadoras.

CAPITULO 2

Análisis y Diseño de Sistemas de Información.

2. ANÁLISIS Y DISEÑO DE SISTEMAS DE INFORMACIÓN.

CONCEPTO

ANÁLISIS DE SISTEMAS.

El análisis es el recuento de aquellos elementos de una situación dada que parece mantener entre ellos relaciones tales que permiten representar una situación bajo la forma de un conjunto coherente y articulado de variables, en el que cada variable no se estudia así misma, sino en referencia al sistema y a las funciones que cumple.

DISEÑO DE SISTEMAS

El Diseño de Sistemas define el proceso de aplicar ciertas técnicas y principios con el propósito de definir un dispositivo, un proceso o un sistema, con suficientes detalles como para permitir su interpretación y realización física⁶.

Por lo tanto, el desarrollo de sistemas busca analizar sistemáticamente la entrada, procesamiento, almacenamiento y salida de datos en un contexto en particular para implementar mejoras automatizadas en diversas áreas.

En una organización ya sea lucrativa o no, las personas trabajan en tiempo real, muchos procedimientos ocurren de manera paralela y el tratamiento de información es continuo, por lo que, el analista de sistemas debe considerar todos los aspectos de la institución o compañía para realizar un sistema de calidad.

Determinar los requerimientos del sistema (es decir, las características necesarias de un sistema) es un proceso de continua evolución. El analista primero comprende el sistema actual y luego lo evalúa.

Para comprender el sistema, el analista comienza con plantear preguntas relacionadas con la finalidad del sistema, sus entradas, salidas y procesos involucrados.

En proyectos de sistemas grandes, varios analistas conducen la investigación y ellos mismos dividen el trabajo en función tal que cada una pueda realizar de manera independiente mientras que, al mismo tiempo, comunican a los demás información y detalles importantes.

Cabe destacar que cada estrategia depende de la habilidad de los analistas para hacer uso de las técnicas para detectar hechos y recopilar detalles relacionados con el sistema.

⁶ Análisis y Diseño de Sistemas. Kendall & Kendall. Prentice Hall 3ª Edición. México 1997

2.1 ESTRATEGIAS PARA EL DESARROLLO DE SISTEMAS .

El desarrollo de sistemas es una tarea que se puede manejar como proyecto, es decir, debe tener un inicio y un fin, una secuencia de pasos o etapas y debe ajustarse a un presupuesto.

Los proyectos de sistemas se sugieren por dos razones:

1. Para experimentar en problemas que les lleven por si mismos a soluciones de sistemas.
2. Para reconocer oportunidades y hacer mejoras mediante la actualización, alteración o instalación de nuevos sistemas.

Para el desarrollo de sistemas es de vital importancia conocer las diversas metodologías, como son:

- Método del desarrollo de análisis estructurado.
- Método del ciclo de desarrollo de sistemas.
- Método de prototipo de sistemas.

La estrategia de desarrollo consistirá en seleccionar el método adecuado a la situación específica de cada proyecto para cumplir el objetivo efectivo. Por otro lado el seleccionar una metodología errónea para un proyecto podría ocasionar pérdida de tiempo y dinero durante el desarrollo. De ahí la importancia de definir una buena estrategia desde el principio.

Cada una de las metodologías presentan ventajas y desventajas ante cada proyecto. Una estrategia podría ser la combinación de métodos ya que estos no son excluyentes.

Para que el analista logre su objetivo de mejora, ya sea en el sistema existente o en la creación de un nuevo sistema debe:

- Aprender los detalles y procedimientos del sistema en uso.
- Obtener una idea de las futuras demandas que la organización requerirá.
- Documentar los detalles del sistema existente para su revisión y discusión.
- Evaluar la eficiencia y efectividad del sistema actual tomando en cuenta las nuevas necesidades.
- Documentar las características que debe tener el nuevo sistema con un nivel detallado que permita comprender sus componentes y su funcionamiento.
- Fomentar la participación de los usuarios actuales y futuros durante todo el proceso.

2.2 MÉTODO DE DESARROLLO POR ANÁLISIS ESTRUCTURADO.

CONCEPTO.

El análisis estructurado es un método para el análisis de sistemas manuales y automatizados, que conduce al desarrollo de especificaciones para sistemas nuevos o para efectuar modificaciones a los ya existentes.

OBJETIVO.

El objetivo que persigue el análisis estructurado es organizar las tareas asociadas con la determinación de requerimientos para obtener la comprensión total y exacta de una situación. A partir de aquí se obtendrán los nuevos requerimientos.

COMPONENTES.

El análisis estructurado hace uso de los siguientes componentes:

1. Símbolos gráficos: Para identificar y describir los componentes de un sistema y sus relaciones.
2. Diccionario de datos: Descripciones de los datos utilizados en el sistema.
3. Descripciones de procesos y procedimientos: Declaración de las actividades importantes que forman parte del sistema
4. Reglas: Estándares para describir y documentar el sistema en forma correcta y completa.

Para documentar el sistema existente y determinar los requerimientos de información por medio del método estructurado es esencial el análisis de flujo de datos.

2.2.1 ANÁLISIS DE FLUJO DE DATOS.

El análisis de flujo de datos estudia el empleo de los datos en cada actividad.

Documenta los hallazgos con diagramas de flujo de datos que muestran en forma gráfica la relación entre procesos y datos del sistema y los sitios donde son utilizados.

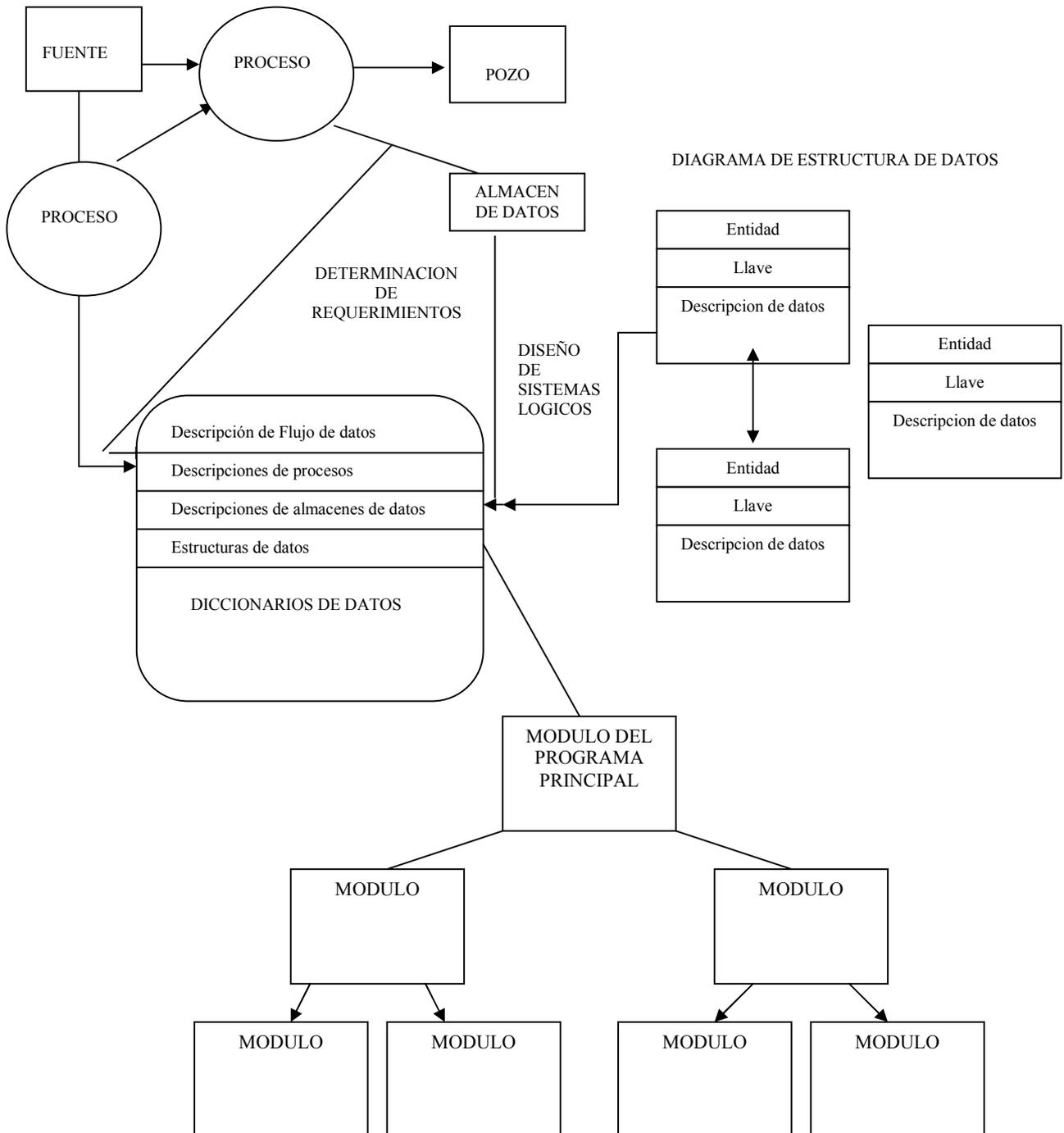
2.2.1.1 CARACTERÍSTICAS DE LA ESTRATEGIA DE FLUJO DE DATOS.

Los componentes de la estrategia de flujo de datos abarcan tanto la determinación de los requerimientos de diseño del sistema, desde donde se originan los datos, como se utilizan, como cambian y hacia donde van.

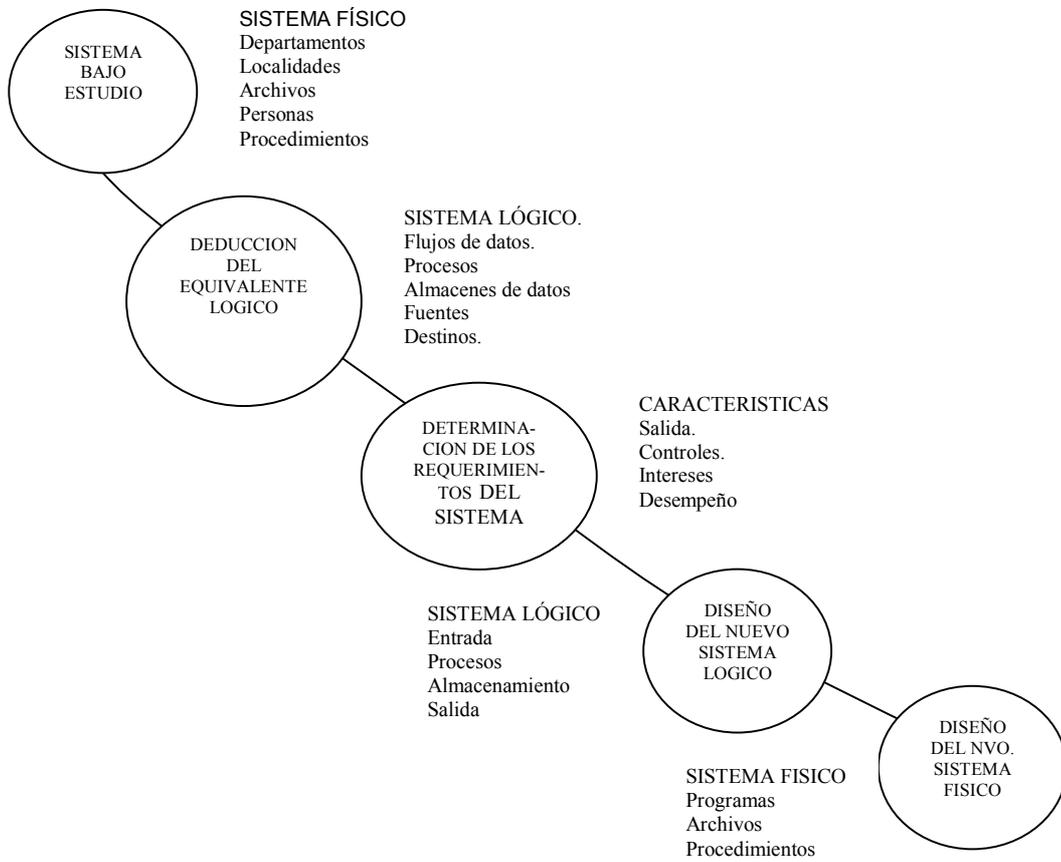
2.2.1.2 HERRAMIENTAS DE FLUJO DE DATOS.

1. *Diagrama de flujo de datos:* Es una herramienta gráfica que se emplea para describir y analizar el movimiento de datos a través del sistema, tanto entrada, salida y almacenamiento de datos.
2. *Diccionario de datos:* El diccionario contiene los datos del sistema. También identifica los procesos que se emplean los datos y los sitios donde se necesita el acceso inmediato de la información.
3. *Diagrama de estructura de datos:* Este diagrama es una descripción de la relación entre entidades de un sistema y el conjunto de información relacionada con la entidad.
4. *Grafica de estructura:* Herramienta de diseño que muestra con símbolos la relación entre módulos de procesamiento y el conjunto de información relacionada con la entidad.

RELACIÓN DE LOS COMPONENTES DE ANÁLISIS ESTRUCTURADO.



SECUENCIAS DE PANORAMAS LÓGICOS Y FÍSICOS DEL SISTEMA



1. Diagrama de flujo de datos

El análisis de flujo de datos permite al analista listar las áreas de interés en la organización y estudiarlas examinando los datos que entran en el proceso, para ello, se debe estudiar primero el sistema en uso, esto es, las actividades y procesos que ocurren en el presente.

Así, el sistema físico se traslada en una descripción lógica que se centra en datos y procesos.

Durante el análisis de flujo de datos se evalúan todos los detalles en términos de los componentes lógicos de flujo de datos, procesos, almacenes de datos, orígenes y destinos.

En todas las etapas de diseño que siguen, los requerimientos del sistema se trasladan en detalles de diseño lógico. En las fases de construcción, como la programación del software para computadora, las aplicaciones lógicas son trasladadas en características físicas y en sistemas de información que trabaja.

El análisis y diseño de sistemas de información es el escenario del estudio que sigue sobre el análisis de flujo de datos.

- Diagramas Físicos: Proporcionan un panorama del sistema en uso, que es dependiente de la implantación, que muestra que tareas se llevan a cabo. Las características físicas incluyen:
 - Nombres de personas.
 - Nombre enumerados de formatos y documentos.
 - Nombres de departamentos.
 - Archivos y transacciones.
 - Dispositivos
 - Ubicaciones.
 - Procedimientos.

- Diagramas lógicos de flujo de datos: Se centra en el flujo de datos entre los procesos sin considerar los dispositivos específicos.

El desarrollo de una descripción completa y exacta del sistema en uso comienza con el desarrollo de diagramas físicos, con lo cual el analista plasma en ellos, los siguientes:

1. Identificación de las personas y lo que hacen, las formas, documentos, actividades y el equipo empleado para el procesamiento.
2. La relación que existe entre las personas, localidades y los documentos con cada entidad.
3. Proporcionan un camino para validar o verificar el punto de vista de usuario sobre la forma en que opera el nuevo sistema en uso.

El diagrama lógico es una visión respectiva de la implantación actual y proporciona la base para examinar la combinación de procesos, flujo de datos, almacenes de datos, entradas y salidas sin tomar en cuenta dispositivos físicos.

El diagrama lógico de flujo de datos se obtiene del diagrama físico de flujo al llevar a cabo lo siguiente:

- Señalar los datos necesarios en este momento para un proceso, no los documentos que los contienen.
- Indicar el flujo de los procedimientos y no entre personas.
- Consolidar los almacenes de datos redundantes.
- Remover los procesos innecesarios.

Reglas generales para el dibujo de diagramas lógicos de flujo de datos.

1. Todos los datos tienen una entrada, proceso y salida dentro de un procedimiento.
2. El nombre del diagrama de flujo de datos debe ser representativo del proceso al que representa.
3. El diagrama de flujo debe contener los datos esenciales del proceso.
4. El proceso debe ser independiente excepto en sus insumos y sus salidas.
5. Los procesos siempre están en continua ejecución.

2. Diccionario de datos.

Un diccionario de datos es un catálogo o depósito de elementos de un sistema.

En un diccionario de datos se encuentra la lista de todos los elementos que forman parte del flujo de datos en todo el sistema.

En el diccionario de datos podremos verificar cuantos caracteres abarca un determinado dato o que otros nombres recibe en distintas partes del sistema.

Este se desarrolla durante el análisis de flujo de datos y auxilia a los analistas que participan en la determinación de los requerimientos del sistema.

Este se desarrolla durante el análisis de flujo de datos y auxilia a los analistas que participan en la determinación de los requerimientos del sistema.

Los diccionarios de datos son importantes ya que:

- Permiten manejar los detalles en sistemas grandes.
- Comunican un significado para cada elemento del sistema.
- Permiten documentar las características del sistema.
- Facilitan el análisis de los detalles.
- Ayudan a localizar errores y omisiones del sistema.

3. Diagramas de estructura de datos.

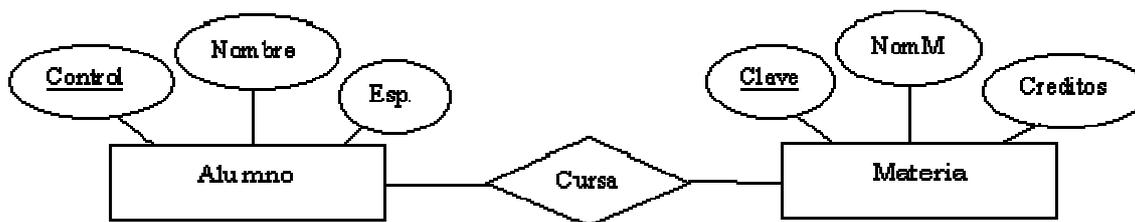
Un diagrama de estructura de datos es un esquema que representa el diseño de una base de datos. Este modelo se basa en representaciones entre registros por medio de ligas, existen relaciones en las que participan solo dos entidades (binarias) y relaciones en las que participan más de dos entidades (generales) ya sea con o sin atributo descriptivo en la relación.

La forma de diagramado consta de dos componentes básicos:

- Celdas: representan a los campos del registro.
- Líneas: representan a los enlaces entre los registros.

Un diagrama de estructura de datos de red, especifica la estructura lógica global de la base de datos; su representación gráfica se basa en el acomodo de los campos de un registro en un conjunto de celdas que se ligan con otro(s) registro(s), ejemplificaremos esto de la siguiente manera:

Consideremos la relación alumno-curso-materia donde la relación curso no tiene atributos descriptivos:



Las estructuras de datos según la cardinalidad se representan en los siguientes casos:

Cuando el enlace no tiene atributos descriptivos

Caso 1. Cardinalidad Uno a Uno.



Diagrama de estructura de datos

Caso 2. Cardinalidad Muchos a uno.

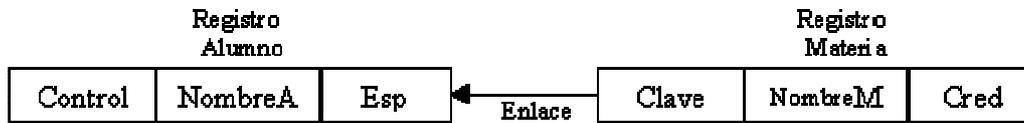


Diagrama de estructura de datos

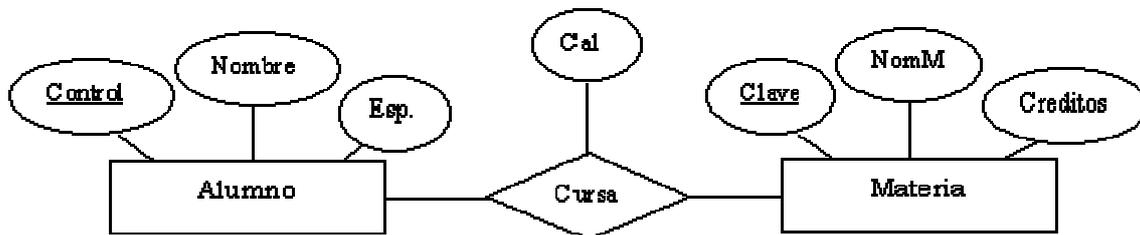
Caso 3. Cardinalidad Muchos a muchos.



Diagrama de estructura de datos

Cuando el enlace tiene atributos descriptivos.

Consideremos que a la relación cursa le agregamos el atributo Cal (calificación), nuestro modelo E-R quedaría de la siguiente manera:



La forma de convertir a diagramas de estructura de datos consiste en realizar lo siguiente:

1. Realizar la representación de los campos del registro agrupándolo en sus celdas correspondientes.
2. Crear nuevo registro, denominado Calif, para este caso, con un solo campo, el de cal (calif).
3. Crear los enlaces indicando la cardinalidad de :
 - AluCal, del registro Calif al registro *Alumno*.
 - MatCal, del registro *Calif* al registro *Materia*.

- *AluCal y MatCal son solo los nombres que emplearemos para identificar el enlace, pueden ser otros y no son empleados para otra cosa.*

Los diagramas de estructuras de datos según la cardinalidad se transforman en:

Caso 1. Cardinalidad uno a uno.

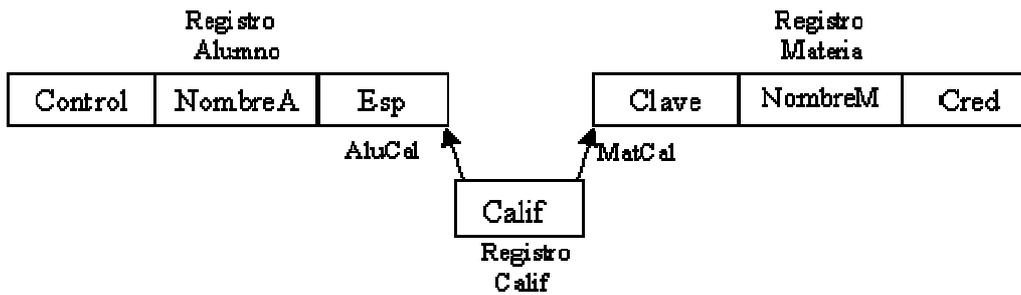


Diagrama de estructura de datos

Caso 2. Cardinalidad Uno a muchos.

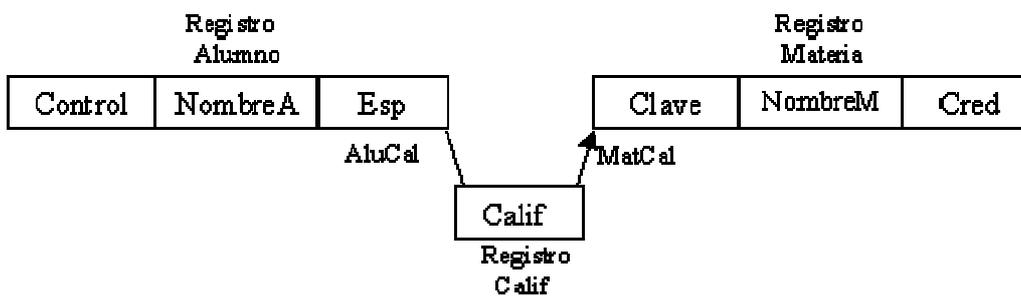


Diagrama de estructura de datos

Caso 3. Cardinalidad Muchos a muchos.

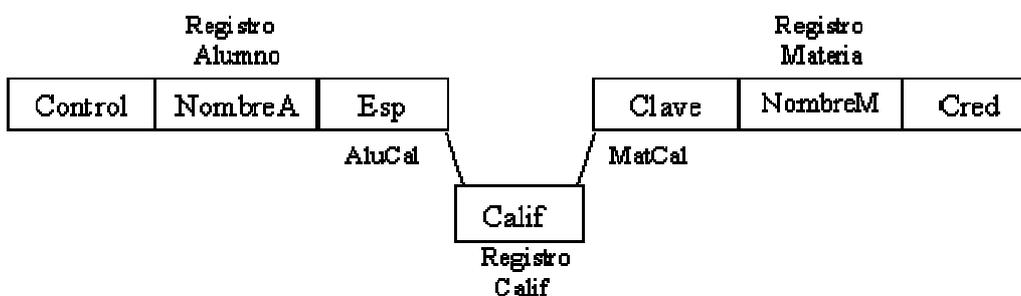
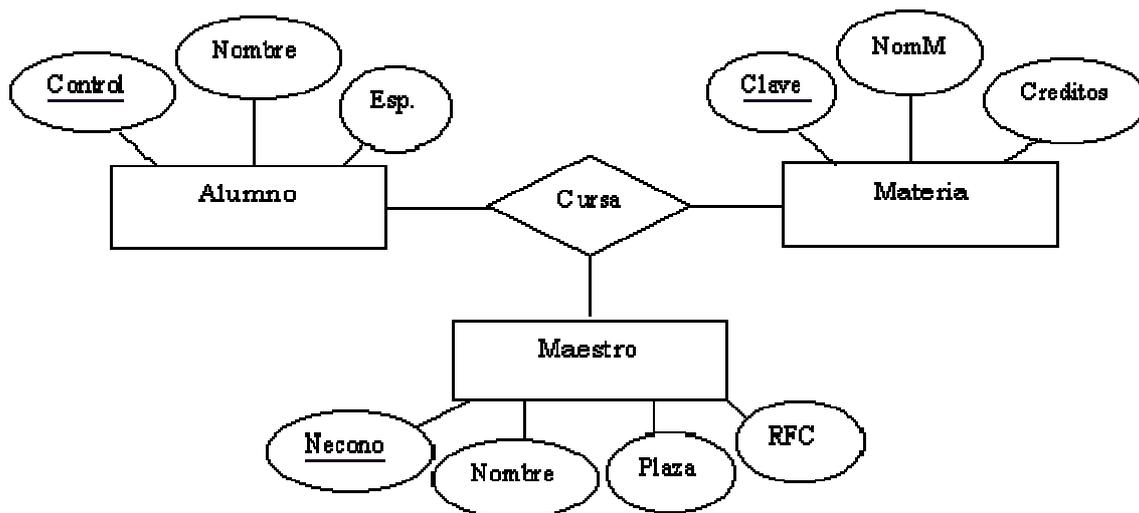


Diagrama de estructura de datos

Diagramas de estructura de datos cuando intervienen más de dos entidades y el enlace no tiene atributos descriptivos.

Consideremos que a la relación alumno-curso-materia le agregamos la entidad maestro, quien es el que imparte dicha materia.

Nuestro diagrama E-R quedaría de la siguiente manera:



La transformación a diagramas de estructura de datos se realiza mediante los siguientes pasos:

1. Crear los respectivos registros para cada una de las entidades que intervienen en el modelo.
2. Crear un nuevo tipo de registro que llamaremos Renlace, que puede no tener campos o tener solo uno que contenga un identificador único, el identificador lo proporcionará el sistema y no lo utiliza directamente el programa de aplicación, a este registro se le denomina también como registro ficticio o de enlace o unión.

Siguiendo los pasos anteriores nuestra estructura finalmente es: (Considerando una relación con cardinalidad Uno a Uno)

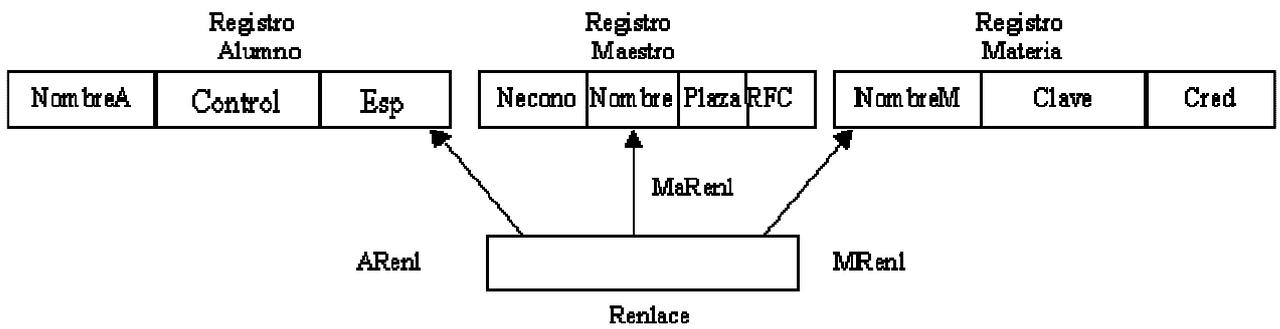


Diagrama de estructura de datos

Ahora si nuestro enlace tuviera atributos descriptivos, se crea el registro con los campos respectivos y se liga indicando el tipo de cardinalidad de que se trate.

En este caso tomamos el ejemplo anterior con cardinalidad uno a uno y le agregamos a la relación el atributo calif. (calificación).

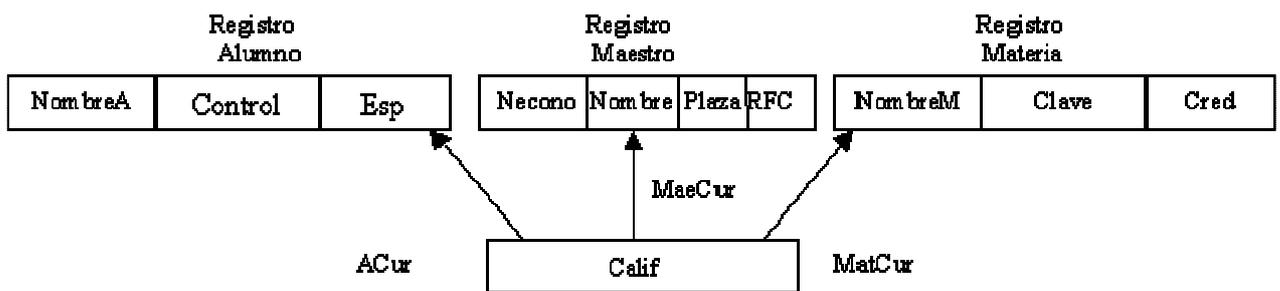
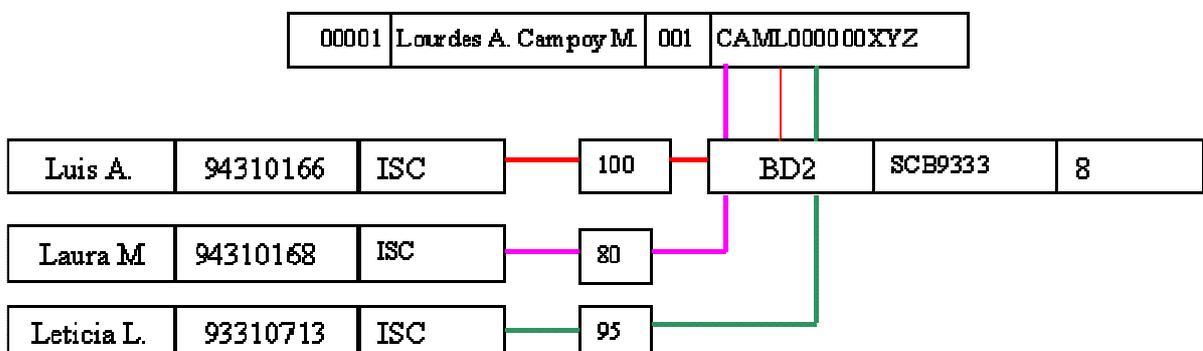


Diagrama de estructura de datos

Considerando el anterior diagrama de estructura de datos, una instancia de este sería:

La estructura quedaría:



Este diagrama nos indica que los alumnos Luis A. Laura M. y Leticia L. cursaron la materia Base de datos 2 con La maestra Ing. Lourdes A. Campoy M obteniendo una calificación de 100,80,95 respectivamente.

Este modelo fue desarrollado en 1971 por un grupo conocido como CODASYL: Conference on Data System Languages, Data Base Task Group, el modelo CODASYL ha evolucionado durante los últimos años y existen diversos productos DBMS orientados a transacciones.

2.3 MÉTODO DEL CICLO DE VIDA DEL DESARROLLO DE SISTEMAS.

CONCEPTO.

El ciclo de vida del desarrollo de sistemas es el conjunto de actividades de los analistas, diseñadores y usuarios, que necesitan llevarse a cabo para desarrollar y poner en marcha un sistema de formación. Se debe tener presente que en la mayoría de las situaciones del negocio, las actividades están íntimamente relacionadas y son inseparables⁷.

Además de ser un enfoque por fases del análisis y diseño que sostiene que los sistemas son desarrollados de mejor manera mediante el uso de un ciclo específico de actividades de el analista y del usuario.

Los analistas no están de acuerdo respecto al número exacto de fases que conforman este ciclo de vida del desarrollo de sistemas, pero, por lo general, alaban su enfoque organizado.

Es recomendable el ciclo de vida para proyectos de gran escala cuando involucra a varios departamentos, cuando se tienen los procedimientos bien establecidos o cuando se tiene que trabajar con un equipo de personas.

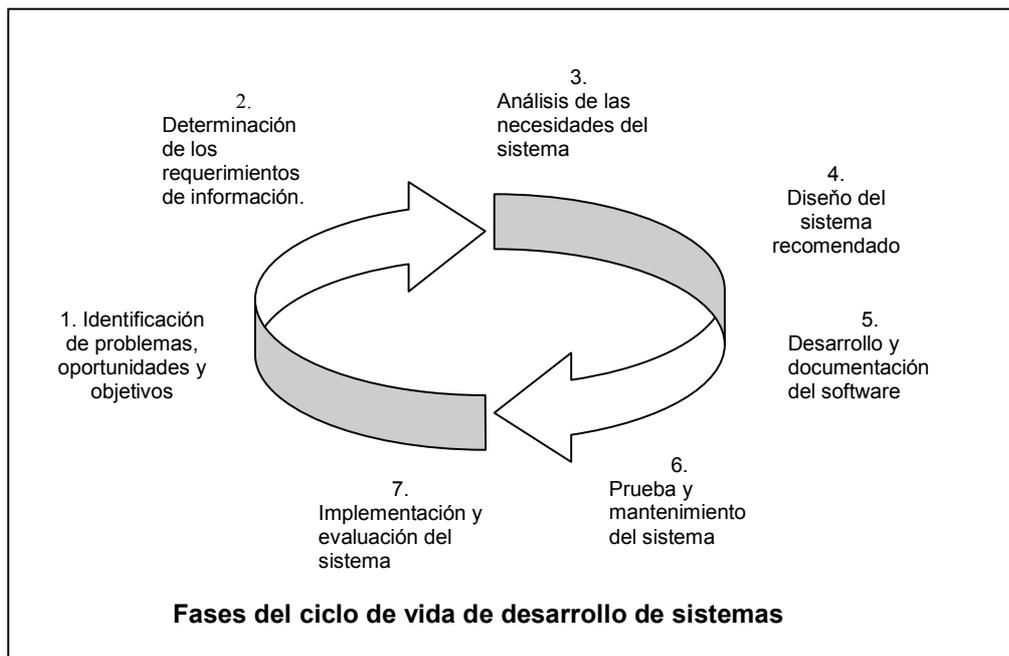
De acuerdo con Kendall en su libro "Análisis y desarrollo de sistemas" el ciclo está dividido en siete fases.

1. Identificación de problemas, oportunidades y objetivos.
2. Determinación de los requerimientos de información.
3. Análisis de las necesidades del sistema.
4. Diseño del sistema recomendado.
5. Desarrollo y documentación del software.
6. Pruebas y mantenimiento del sistema.
7. Implementación y evaluación del sistema.

⁷ Análisis y Diseño de Sistemas. Kendall & Kendall. Prentice Hall 3ª Edición. México 1997

Aunque cada fase es presentada individualmente, nunca se lleva a cabo como un paso aparte; en vez de ello, varias actividades pueden suceder simultáneamente, las actividades pueden ser repetidas. Esta es la razón por la cual es más útil pensar que el ciclo se logra en fases (con actividades traslapándose) y no en pasos separados.

A continuación se detalla cada uno de estas fases.



Análisis y Diseño de Sistemas. Kendall & Kendall 1997

2.3.1 IDENTIFICACIÓN DE PROBLEMAS, OPORTUNIDADES Y OBJETIVOS.

En la primera fase del ciclo de vida del desarrollo de sistemas el analista tiene que ver con la identificación de problemas, oportunidades y objetivos.

Esta etapa es crítica para el éxito del resto de proyecto, debido a que nadie quiere desperdiciar el tiempo subsiguiente resolviendo el problema equivocado.

Identificación del problema.

La primera fase requiere que el analista observe honestamente lo que está sucediendo en un negocio. Luego, junto con los demás miembros de la organización, el analista hace resaltar los problemas. Frecuentemente estos ya han sido vistos por los demás, y son la razón por la cual el analista fue llamado

inicialmente.

A los administradores no les gusta pensar que su organización tenga problemas, y ni siquiera hablar de ellos o compartirlos con alguien externo. Sin embargo, los buenos administradores se dan cuenta que el reconocimiento de síntomas de problemas o en un momento posterior, el diagnosticar los problemas por ellos mismos y enfrentarlos, es imperativo si el negocio va al seguir funcionando a su más alto potencial.

Los problemas salen a la superficie en muchas formas diferentes. Una forma de conceptualizar cuáles son los problemas, y cómo se presentan, o pensar sobre ellos como situaciones donde los objetivos nunca se han cumplido o ya no se están cumpliendo. La retroalimentación útil da información acerca de la diferencia entre el desempeño actual y el pretendido. De esta forma, la retroalimentación hace resaltar los problemas.

En algunos casos son descubiertos problemas que requieren los servicios de analistas de sistemas, debido a que no se están cumpliendo las medidas de desempeño. Los problemas (o síntomas de problemas) con procesos que son visibles en la salida, y que pueden requerir la ayuda de un analista, incluyen errores excesivos y trabajo desarrollado demasiado lento, incompleto, en forma incorrecta o incluso que no se realiza.

Otros síntomas de problemas se hacen evidentes cuando las personas no logran los objetivos de desempeño básicos. Cambios en el comportamiento de los empleados, tales como ausentismo extrañamente alto, gran insatisfacción en el trabajo o alta rotación de trabajadores, pueden servir para alertar a los administradores de que hay problemas potenciales. Cualquiera de estos síntomas, aislados o en combinación, pueden ser razón suficiente para solicitar la ayuda de un analista de sistemas.

Aunque las dificultades como las mencionadas anteriormente suceden dentro de la organización. La retroalimentación sobre qué tan bien está satisfaciendo la organización sus objetivos pretendidos puede llegar de fuera en forma de queja o sugerencias de los clientes, vendedores o proveedores, o bien ventas perdidas o inesperadamente bajas. Esta retroalimentación a partir de ambiente externo es extremadamente importante y no debe ser ignorada.

Oportunidades.

Las oportunidades son situaciones que el analista considera pueden ser mejoradas por medio de sistemas de información computarizados. El aprovechar las oportunidades puede permitir que el negocio gane un avance competitivo o ponga un estándar de la industria.

El analista de sistemas sirve como el catalizador y experto de soporte principalmente al ser capaz de ver donde pueden ser mejorados los procesos.

En forma optimista, las oportunidades pueden ser concebidas como el anverso de los problemas, aunque en algunas culturas las crisis también significan oportunidades. Lo que surge como un problema perturbante para un administrador, puede ser cambiado en una oportunidad para un analista de sistemas.

Las mejoras a los sistemas pueden ser definidas como cambios que darán como resultado beneficios aumentados y que valen la pena. Existen muchas posibilidades de mejoras, incluyendo:

1. Aceleración de un proceso.
2. Agilización de un proceso mediante la eliminación de pasos innecesarios o duplicados.
3. Combinación de procesos.
4. Reducción de errores en la entrada por medio de cambios en formas y pantallas.
5. Reducción de salida redundante.
6. Mejora en la integración de sistemas y subsistemas.
7. Mejora de la satisfacción del trabajador con el sistema.
8. Mejora de la facilidad de interacción de los clientes, proveedores y vendedores, con el sistema.

Se encuentra dentro de las capacidades del analista de sistemas el darse cuenta de las oportunidades para mejoras. Sin embargo, las personas que están en contacto diario con el sistema pueden ser mejores fuentes de información acerca de las mejoras que deben realizarse. Si las mejoras ya han sido sugeridas, se necesita la experiencia propia para que ayude a determinar si valen la pena y la forma en que pueden ser implementadas.

Aunque la lista de mejoras dada anteriormente es larga, no es exhaustiva. La parte difícil de sugerir o implementar mejoras, es que deben valer el desconcierto y costo del cambio.

Objetivos.

La identificación de objetivos es también un componente importante de la primera fase. En primer lugar, el analista debe descubrir lo que está tratando de hacer el negocio. Luego será capaz de ver si algún aspecto de la aplicación de sistemas de información puede ayudar para que al negocio alcance sus objetivos atacando problemas específicos u oportunidades.

El propósito de determinar los objetivos es obtener las razones que se encuentran tras los requerimientos.

Un objetivo es una frase que cuando se lleva a cabo elimina el problema o aprovecha una oportunidad. Los objetivos son como pequeñas metas. También deben ser concisos, claros, y medibles. Cada objetivo soporta a una parte de la meta, si se llegan a alcanzar todos los objetivos se ha alcanzado la meta total.

Las personas involucradas en la primera fase son los usuarios, analistas y administradores de sistemas que coordinan el proyecto. Las actividades de esta fase consisten en entrevistas a los administradores de los usuarios. Sumarización del conocimiento obtenido, estimación del alcance del proyecto y documentación de los resultados. La salida de esta fase es el estudio de factibilidad que contiene una definición del problema y la Sumarización de los objetivos.

Dicho estudio de factibilidad abarca tres aspectos:

Factibilidad técnica. ¿Puede realizarse el trabajo para el proyecto con el equipo actual, tecnología de software y el personal disponible? Si se requiere nueva tecnología, ¿qué probabilidades hay de que pueda desarrollarse?

Factibilidad económica. ¿Existen suficientes beneficios en la creación del sistema para hacer que los costos sean aceptables? O, en forma inversa, ¿son tan altos los costos como para que el proyecto no deba llevarse a cabo?

Factibilidad operativa. ¿Se utilizará el sistema si se desarrolla y pone en marcha? Habrá resistencia de los usuarios, que los posibles beneficios reducirán del sistema.

El estudio de factibilidad se lleva a cabo con un pequeño grupo de gente, familiarizada con las técnicas de los sistemas de información, que entienden la parte de la empresa que será afectada por el proyecto y tienen los conocimientos suficientes del proceso de análisis y diseño de sistemas.

Si el grupo de usuarios no tiene los suficientes fondos en su presupuesto y desea atacar problemas que no están relacionados, o los problemas no requieren un sistema de cómputo, puede ser recomendada una solución manual y el proyecto de sistemas ya no continúa o se realiza posteriormente.

2.3.2 DETERMINACIÓN DE LOS REQUERIMIENTOS DE INFORMACIÓN.

La siguiente fase a la que entra el analista es la de la determinación de los requerimientos de información para los usuarios particulares involucrados. Entre las herramientas utilizadas para definir los requerimientos de información en el negocio se encuentran: muestra e investigación de los datos relevantes, entrevistas, cuestionarios, el comportamiento de los tomadores de decisiones y su ambiente de oficina y hasta la elaboración de prototipos.

En esta fase el analista está esforzándose por comprender qué información necesitan los usuarios para realizar su trabajo.

Así como comprender los procesos realizados en la empresa o ente donde se implementará el sistema; para lo cual, el analista deberá responderse las siguientes preguntas:

- ¿Qué es lo que se hace?
- ¿Cómo se hace?
- ¿Con qué frecuencia se presenta?
- ¿Qué tan grande es el volumen de transacción o decisiones?
- ¿Cuál es el grado de eficiencia con el que se efectúan las tareas?
- ¿Existe algún problema?
- ¿Si existe algún problema que tan serio es? ¿Y Cuál es la causa que lo origina?

Se puede ver que varios de los métodos para determinar los requerimientos de información involucran la interacción directa con los usuarios.

Algunos de ellos son:

- Muestreos
- Entrevistas
- Cuestionarios
- Observación
- Prototipos.

Muestreos

El proceso de seleccionar sistemáticamente elementos representativos de una población es llamado muestreo. El objeto del muestreo es seleccionar y estudiar documentos, tales como facturas, reportes de ventas y memorándums, o tal vez seleccionar y entrevistar, dar cuestionarios u observar a miembros de la organización.

El muestreo puede reducir costos, velocidad de recolección de datos, hacer potencialmente que el estudio sea más efectivo y posiblemente reducir la ascendencia en el estudio.

Un analista de sistemas debe seguir cuatro pasos en el diseño de una buena muestra.

1. Determinar la población misma.
2. Decidir el tipo de muestra.
3. Calcular el tamaño de muestra.
4. Planear los datos que necesitan ser recolectados o descritos.

Los tipos de muestras útiles para un analista de sistemas son: de conveniencia, intencionada, aleatoria simple y aleatoria compleja. El último tipo incluye las subcategorías de muestreo sistemático y muestreo estratificado. Hay varios lineamientos a seguir para la determinación del tamaño de muestra. El analista de sistemas puede hacer una decisión subjetiva en relación con el estimado de intervalo aceptable. Luego se selecciona un nivel de confianza y puede ser calculado el tamaño de muestra necesario.

Los datos relevantes muestran dónde ha estado la organización y hacia dónde creen sus miembros que está yendo.

Es necesario que sean analizados documentos cuantitativos y cualitativos. Debido a que los documentos son mensajes persuasivos, debe ser reconocido que el cambiarlos también puede cambiar a la organización.

Entrevista:

La entrevista es un método que usa el analista de sistemas para la recolección de datos sobre los requerimientos de información. El analista de sistemas escucha buscando objetivos, sentimientos, opiniones y procedimientos informales en entrevistas con los tomadores de decisiones de la organización. También vende el sistema durante las entrevistas. Las entrevistas son diálogos de preguntas y respuestas planeados por anticipado entre dos personas.

Hay cinco pasos que deben tomarse para la planeación previa de las entrevistas:

1. Lectura de material de fondo
2. Establecimiento de objetivos de la entrevista
3. Decisión de a quién entrevistar
4. Preparación del entrevistado
5. Decisión sobre el tipo y estructura de las preguntas

Las preguntas tienen dos tipos básicos: abiertas y cerradas.

Las preguntas abiertas dejan abiertas todas las opciones de respuesta para la entrevista.

Las preguntas cerradas limitan las opciones posibles de la respuesta.

Las entrevistas pueden estar estructuradas en tres formas básicas estructura de pirámide, de embudo o de rombo. Las estructuras pirámide comienzan con preguntas cerradas y detalladas y se amplían a preguntas más generales.

Las estructuras de embudo comienzan con preguntas abiertas generales y luego se estrechan a preguntas cerradas más específicas.

Las estructuras de rombo combinan las fuerzas de las otras dos estructuras pero se llevan más tiempo para realizarse. Hay compromisos involucrados sobre la decisión de cómo estructurar para realizar las preguntas y secuencia de preguntas de la entrevista.

Después de la entrevista, el entrevistador debe escribir un reporte que liste los puntos principales que se proporcionaron, así como opiniones de lo que fue dicho.

Es extremadamente importante documentar la entrevista la más pronto posible después de que haya sido realizada.

Cuestionarios.

Mediante el uso de cuestionarios los analistas de sistemas pueden recolectar datos sobre actitudes, creencias, comportamientos y características de gentes importantes en la organización. Los cuestionarios son útiles si: las personas de la organización están ampliamente dispersas, muchas gentes están involucradas con el proyecto de sistema, se necesita un trabajo exploratorio antes de recomendar alternativas o hay una necesidad para la sensibilización del problema antes de que se realicen entrevistas.

Una vez que han sido articulados los objetivos del cuestionario, el analista puede comenzar a escribir preguntas abiertas o cerradas. La selección de la redacción es extremadamente importante y debe reflejar el lenguaje de los miembros de la organización. Idealmente, las preguntas deben ser simples, específicas, sin ascendencia, sin menosprecio, técnicamente precisas y dirigidas a aquellos que tienen el conocimiento.

La asignación de escalas es el proceso de asignar números u otros símbolos a un atributo o característica. Tal vez quiera el analista de sistemas usar escalas para medir las actitudes o las características de los interlocutores o para hacer que los interlocutores actúen como jueces sobre el tema del cuestionario.

Las cuatro formas de medición son escalas nominales, ordinales, de intervalo y de relación. La forma de medición es frecuentemente indicada por los datos, y el análisis de los datos es a su vez indicado en alguna medida por la forma de medición.

Los analistas de sistemas necesitan tomar en consideración la validez y la confiabilidad. La validez significa que el cuestionario mide lo que el analista de sistemas pretendió medir. La confiabilidad significa que los resultados son consistentes.

Los analistas deben ser cuidadosos para evitar problemas como lenidad, tendencia central y el efecto de halo cuando construyen escalas.

Observaciones.

Los analistas por medio de la observación obtienen apreciaciones sobre lo que se hace realmente, permite ver de primera mano las relaciones entre los tomadores de decisiones en una organización, comprenden la influencia del ambiente físico de éste, interpretan los mensajes enviados por el tomador por medio de su vestimenta y el acomodo de su oficina.

Esta fase sirve para formar la imagen que el analista tiene de la organización y sus objetivos. Algunas veces solamente se completan las dos primeras fases del ciclo de vida del desarrollo de sistemas. Este tipo de estudio puede tener

diferentes propósitos, y es realizado típicamente por un especialista llamado analista de información.

Las personas involucradas en esta fase son los analistas y los usuarios, típicamente los administradores de las operaciones y los trabajadores de las operaciones. El analista de sistemas necesita conocer los detalles de las funciones actuales del sistema: quién (las personas que están involucradas), qué (la actividad del negocio), dónde (el ambiente donde se lleva a cabo el trabajo), cuándo (en qué momento) y cómo (de qué manera se desarrollan los procedimientos actuales) del negocio bajo estudio. El analista debe preguntar porqué el negocio usa el sistema actual. Puede haber muy buenas razones para desarrollar el negocio usando los métodos actuales, y deben ser considerados cuando se diseña cualquier sistema nuevo.

Sin embargo, si la razón de las operaciones actuales es que "siempre se han hecho así", el analista puede desear la mejora de los procedimientos. La reingeniería de procesos del negocio puede ayudar a enmarcar un enfoque para volver a pensar en el negocio en forma creativa. Al término de esta fase, el analista debe comprender el por qué de las funciones del negocio y tener información completa sobre las personas, objetivos, datos y procedimientos involucrados.

2.3.3 ANÁLISIS DE LAS NECESIDADES DEL SISTEMA.

La siguiente fase que realiza el analista de sistemas involucra el análisis de las necesidades del sistema.

Nuevamente, herramientas y técnicas especiales ayudan para que el analista haga las determinaciones de los requerimientos. Una herramienta de éstas es el uso de diagramas de flujo de datos para diagramar la entrada, proceso y salida de las funciones del negocio en forma gráfica estructurada.

A partir de los diagramas de flujo de datos se desarrolla un diccionario de datos, que lista todos los conceptos de datos usados en el sistema, así como sus especificaciones, si son alfanuméricos y que tanto espacio ocupan cuando se imprimen.

Durante esta fase el analista de sistemas también analiza las decisiones estructuradas que se hacen. Las decisiones estructuradas son aquellas para las que pueden ser determinadas las condiciones como alternativas de condición, acciones y reglas de acción. Hay tres métodos principales para el análisis de decisiones estructurales: lenguaje estructurado, tablas de decisión y árboles de decisión.

No todas las decisiones de la organización son estructuradas, pero todavía es importante que el analista de sistemas las comprenda. Las decisiones

semiestructuradas (decisiones tomadas bajo riesgo) son sustentadas frecuentemente por los sistemas de apoyo a decisiones. Cuando se analizan decisiones semiestructuradas, el analista examina las decisiones con base en el grado de habilidad para la toma de decisiones requerida, el grado de complejidad del problema y la cantidad de criterios considerados cuando se toma la decisión.

El análisis de las decisiones de criterios múltiples (decisiones en las que deben ser balanceados muchos factores) también parte de esta fase. Se dispone de muchas técnicas para el análisis de decisiones de criterios múltiples, incluyendo el proceso de compromiso y el uso de métodos ponderados.

En este punto del ciclo de vida del desarrollo de sistemas, el analista prepara una propuesta de sistema que resume lo que ha sido encontrado, proporciona análisis de costo / beneficio de las alternativas y hace recomendaciones sobre lo que debe ser hecho (en caso de haberlo). Si alguna de las recomendaciones es aceptable para la administración, el analista continúa sobre ese curso. Cada problema de sistema es único y nunca hay una sola solución correcta. La manera en que se formula una solución o recomendación depende de la capacidad y entrenamiento profesional individual de cada analista.

2.3.4 DISEÑO DEL SISTEMA RECOMENDADO.

En esta fase del ciclo de vida del desarrollo de sistemas, el analista usa la información recolectada anteriormente para realizar el diseño lógico del sistema de información. El analista diseña procedimientos precisos para la captura de datos, a fin de que los datos que van a entrar al sistema de información sean correctos. Además, el analista también proporciona entrada efectiva para el sistema de información mediante el uso de técnicas para el buen diseño de formas y pantallas.

Parte del diseño lógico del sistema de información es diseñar la interfaz de usuario. La interfaz conecta al usuario con el sistema y es, por lo tanto, extremadamente importante. Ejemplos de interfaces de usuario incluyen un teclado para introducir preguntas y respuestas, menús en pantalla para elegir comandos del usuario y un ratón para seleccionar opciones.

La fase de diseño también incluye el diseño de archivos o bases de datos que guardarán la mayor parte de los datos necesarios para los tomadores de decisiones de la organización. Una base de datos bien organizada es la base para todos los sistemas de información. En esta fase, el analista también trabaja con los usuarios para diseñar la salida (ya sea en pantalla o impresa) que satisfaga sus necesidades de información.

Por último el analista debe diseñar procedimientos de control y respaldo para proteger al sistema y a los datos y producir paquetes de especificaciones de programa para los programadores. Cada paquete debe contener diseños de

entrada y salida, especificaciones de archivos y detalles de procesamiento y también puede incluir árboles o tablas de decisión, diagramas de flujo de datos, un diagrama de flujo de sistemas y los nombres y funciones cualesquiera de las rutinas de código que ya hayan sido escritas.

Los diseñadores son los responsables de dar las especificaciones de software completas y claramente delineadas, para ello pueden ayudarse de diagramas, tablas y símbolos especiales.

Diseño de la interfaz de usuario

El objetivo debe ser diseñar interfaces que ayuden a los usuarios a proporcionar la información que necesitan proporcionar y obtener del sistema los siguientes objetivos:

1. Efectividad lograda por medio del diseño de interfaces que permitan a los usuarios acceder al sistema en una forma que sea congruente con sus necesidades individuales.
2. Efectividad mostrada por medio de interfaces que aumenten la efectividad de la captura de datos y reduzcan errores.
3. Demostrar consideración al usuario diseñando interfaces adecuadas y que el sistema les proporciona la retroalimentación adecuada.
4. Productividad mostrada por su adecuación a los principios ergonómicos establecidos en el diseño de interfaces y espacios de trabajo para los usuarios.

La interfaz de usuario tiene dos componentes principales: el lenguaje de programación, que es la parte de la computadora al usuario de la transacción, y el lenguaje de acción, que caracteriza la parte usuario a computadora. Ambos conceptos juntos cubren la forma y contenido del término interfaz de usuario.

Las interfaces de lenguaje natural son, tal vez el sueño ideal de los usuarios no experimentados, debido a que permiten que, los usuarios interactúen con la computadora en su lenguaje de todos los días o natural. No se sugiere habilidades especiales del usuario.

Interfaces de preguntas y respuestas.

En este tipo de interfaz la computadora muestra una pregunta al usuario en la pantalla. Para interactuar, el usuario teclea una respuesta y la computadora

interactúa sobre esta información dada en una forma preprogramada, típicamente moviéndose a la siguiente pregunta.

Tal como sucede en la mayoría de las interfaces de pregunta y respuesta, el sistema de cómputo dirige la secuencia de cuestionamiento. El usuario responde a lo que se le pregunta. Por ejemplo, para responder a la frase “Ahora teclee la cantidad de restricciones”, el usuario teclea el número adecuado.

Los programadores tratan de elaborar preguntas para desplegarlas en una interfaz de pregunta y respuesta en una forma concisa y comprensible, pero también necesitan anticipar los tipos de respuesta que dará el usuario y que aceptará el sistema. Una mayor amplitud para la respuesta de los usuarios se traduce directamente en un incremento de la complejidad requerida para el programa.

Cuando se diseñan interfaces se toma una decisión sobre que tanta flexibilidad hay que permitir a los usuarios para la respuesta de preguntas. Los usuarios requieren instrucciones sobre que tanta flexibilidad se les permite.

Es posible incluir ayuda adicional, o avisos, para recordarle al usuario las respuestas aceptables, y muchos programadores lo hacen. Esto es importante, debido a que si los usuarios gastan demasiado esfuerzo adicional para buscar las respuestas, o la manera de responder, pueden quedar insatisfechos y no querer usar el sistema.

La experiencia sugiere que conforme los usuarios llegan a ser más productivos con el sistema, pueden impacientarse con las preguntas detalladas repetitivas. Tal vez prefieran una opción que les permitan responder a versiones abreviadas de las preguntas.

Los nuevos usuarios, que no están familiarizados con la computadora, pueden encontrar más confortables las interfaces de pregunta y respuesta, obteniendo rápidamente confianza por medio de sus éxitos. Sin embargo, los diseñadores de sistema necesitan prevenirse de que un nuevo usuario sobreestime las capacidades del sistema más allá de las realmente presentes.

Menús

En forma similar, una interfaz de menú proporciona al usuario una lista en pantalla de las selecciones disponibles.

Los menús pueden ser anidados dentro de otros para llevar al usuario través de opciones de un programa. Los menús anidados tienen algunas ventajas. Permiten que la pantalla aparezca menos amontonada, lo que es consistente con el buen diseño de la pantalla.

Interfaces de llenado de forma (formas entrada / salida)

Las formas en pantalla para mostrar o capturar información deben ser visualmente atractivas sin dejar de lado la sencillez para no lastimar la vista de los usuarios, por lo tanto las recomendaciones para crear una interfaz de calidad visual son las siguientes:

- Los campos en blanco que pueden ser resaltados con colores inversos.
- El tamaño de caracteres debe ser el apropiado para hacer los datos legibles, evitando forzar la vista del usuario.
- Considerar el espacio dado entre caracteres, figuras o imágenes.
- Procurar que su lugar de trabajo tenga luz suave.
- Evitar los reflejos en el monitor y los focos directos de luz frente a los ojos. Si el monitor es a colores, emplear tonos suaves, evitando los colores escandalosos o agresivos a la mirada.
- Tomar descansos. Unos instantes cada pocos minutos y descansos más largos cada hora.

2.3.5 DESARROLLO Y DOCUMENTACIÓN DEL SOFTWARE.

En la quinta fase del ciclo de vida del desarrollo de sistemas el analista trabaja con los programadores para desarrollar cualquier software original que se necesite. Algunas de las técnicas estructuradas para el diseño y documentación de software incluyen diagramas estructurados, el método HIPO, diagramas de flujo, diagramas Nassi-Schneiderman y Warnier-Orr y pseudocódigo. El analista de sistemas usa uno o más de estos dispositivos para comunicar al programador lo que necesita ser programado.

Durante esta fase, el analista también trabaja con los usuarios para desarrollar documentación efectiva para el software, incluyendo manuales de procedimientos. La documentación le dice al usuario la manera de usar el software y también qué hacer si se suceden problemas con el software.

Los programadores tienen un papel principal en esta fase conforme al diseño, codifican y eliminan errores de sintaxis de los programas de computadora. Para asegurar a calidad, un programador puede realizar ya sea un diseño o un ensayo del código, explicando las partes complejas del programa a un equipo de otros programadores.

PRUEBAS Y MANTENIMIENTO DEL SISTEMA.

Antes de que pueda ser usado, el sistema de información debe ser probado. Es mucho menos costoso encontrar problemas antes de que el sistema sea entregado a los usuarios. Algunas de las pruebas son realizadas por los programadores solos, y otras por los analistas de sistemas junto con los progra-

madores. Primero se ejecuta una serie de pruebas para que destaquen los problemas con datos de ejemplo y eventualmente con datos reales del sistema actual.

En muchas organizaciones, las pruebas son conducidas por personas ajenas al grupo que escribió los programas originales; con esto se persiguen asegurar por una parte que las pruebas sean completas e imparciales y por otra parte, que el software sea más confiable.

El mantenimiento del sistema y de su documentación comienza en esta fase y es efectuado rutinariamente a lo largo de la vida del sistema de información. Mucho del trabajo rutinario del programador consiste en el mantenimiento, ya que los negocios gastan gran cantidad de dinero en dicho mantenimiento. Muchos de los procedimientos sistemáticos que emplea el analista a lo largo del ciclo de vida del desarrollo del sistema pueden ayudar a asegurar que el mantenimiento se mantenga al mínimo.

La importancia del mantenimiento:

Después de que el sistema está instalado se le debe dar mantenimiento, esto significa que los programas de computadora deben ser modificados y mantenidos actualizados. La figura 1.3 muestra la cantidad promedio de tiempo gastada en mantenimiento en una instalación MIS típica. Las estimaciones del tiempo gastado por los departamentos en mantenimiento ha ido de 48 al 60 por ciento del tiempo total empleado en el desarrollo de sistemas. Queda muy poco tiempo para nuevo desarrollo de sistemas. Conforme aumenta la cantidad de programas escritos, también aumenta la cantidad de mantenimiento que requieren.

El mantenimiento se realiza por dos razones. La primera de estas es para corregir errores de software. Sin importar que tan completamente se pruebe el sistema, se deslizan errores en los programas de computadora. Los errores del software comercial para microcomputadoras son a veces documentados como "anomalías conocidas", y son corregidos cuando son lanzadas nuevas versiones del software o versiones intermedias.

En el software personalizado los errores deben ser corregidos conforme son detectados.

La otra razón para realizar el mantenimiento del sistema es para mejorar las capacidades del software en respuesta a las necesidades organizacionales cambiantes y, por lo general, involucran algunas de las siguientes tres situaciones:

1. Los usuarios frecuentemente solicitan características adicionales después de que se familiarizan con el sistema de cómputo y sus capacidades. Éstas características solicitadas pueden ser tan simples como el desplegado de totales adicionales en un reporte o tan complicadas como el desarrollo de nuevo software.

2. El negocio cambia a través del tiempo. Se debe modificar el software para abarcar cambios tales como nuevos requerimientos de reportes gubernamentales o comparativos, la necesidad de producir nueva información para clientes.

El hardware y software están cambiando a ritmo acelerado. Un sistema que usa tecnología antigua puede ser modificado para usar las capacidades de una tecnología más nueva.

La siguiente figura ilustra la cantidad de recursos, tiempo y dinero, gastados en el desarrollo y mantenimiento del sistema. El área bajo la curva representa la cantidad monetaria total gastada. Se puede ver que a la largo del tiempo es probable que el costo de mantenimiento exceda al del desarrollo del sistema. En cierto punto es más conveniente realizar un nuevo estudio del sistema, debido a que el costo de mantenimiento continuado es claramente mayor que la creación de un sistema de información completamente nuevo.

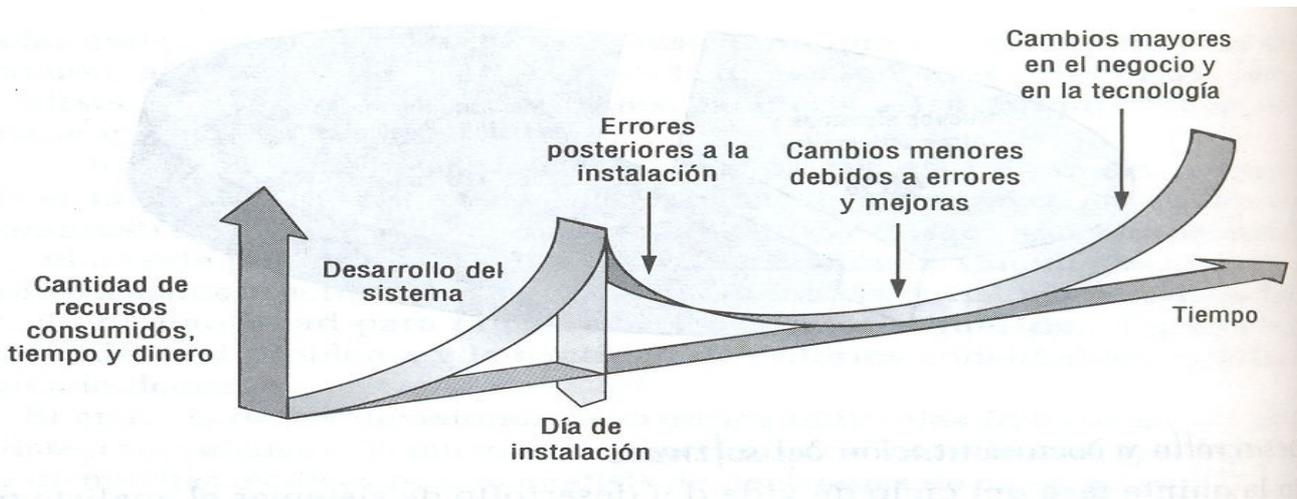


Figura 1.3

Análisis y Diseño de Sistemas. Kendall & Kendall 1997

Resumiendo, el mantenimiento es un proceso continuo a lo largo del ciclo de vida de un sistema de información. Después de que es instalado el sistema de información, el mantenimiento por lo general toma la forma de corrección de errores de programa no detectados previamente. Una vez que son corregidos, el sistema alcanza un estado estable proporcionando servicios confiables a sus usuarios. El mantenimiento durante este periodo puede consistir en la eliminación de unos cuantos errores no detectados anteriormente y la actualización del sistema con una cuantas mejoras.

2.3.6 IMPLEMENTACIÓN Y EVALUACIÓN DEL SISTEMA.

En esta fase del desarrollo del sistema el analista ayuda a implementar el sistema de información. Esto incluye el entrenamiento de los usuarios para que manejen el sistema. Algún entrenamiento es hecho por los proveedores, pero la supervisión del entrenamiento es responsabilidad del analista de sistemas. Adicionalmente, el analista necesita un plan para una conversión suave del sistema antiguo al nuevo. Este proceso incluye la conversión de archivos de formatos antiguos a nuevos de la construcción de una base de datos, la instalación de equipo y la puesta del nuevo sistema en producción.

La evaluación se muestra como parte de esta fase final del ciclo de vida del desarrollo del sistema, principalmente para efectos de discusión. De hecho, la evaluación se realiza durante cada fase. Un criterio principal que debe ser satisfecho es si los usuarios pretendidos ya están usando el sistema.

Debe hacerse notar que a veces los sistemas trabajan en forma cíclica. Cuando un analista termina una fase del desarrollo del sistema y pasa a la siguiente el descubrimiento de un problema puede obligar a que el analista regrese a la fase anterior y modifique el trabajo que allá hizo.

Por ejemplo, durante la fase de prueba el programador puede descubrir que el programa no trabaja correctamente, ya sea debido a que no se escribió código para apoyar determinadas partes del diseño del sistema o aquel diseño fue incompleto. En cualquier caso deben ser modificados los programas, y el analista puede tener que cambiar algunos de los materiales del diseño del sistema. A su vez, esto puede necesitar que el analista se reúna con el usuario y vuelva a investigar cómo funciona una actividad específica del negocio.

CAPITULO 3

Tipos de Software.

3. TIPOS DE SOFTWARE

El software es el conjunto de instrucciones que las computadoras emplean para manipular datos. Sin el software, la computadora sería un conjunto de medios sin utilizar. Al cargar los programas en una computadora, la máquina actuará como si recibiera una educación instantánea; de pronto "sabe" cómo pensar y cómo operar.

El Software es un conjunto de programas, documentos, procedimientos, y rutinas asociados con la operación de un sistema de computo. Distinguiéndose de los componentes físicos llamados hardware. Comúnmente a los programas de computación se les llama software; el software asegura que el programa o sistema cumpla por completo con sus objetivos, opera con eficiencia, esta adecuadamente documentado, y suficientemente sencillo de operar. Es simplemente el conjunto de instrucciones individuales que se le proporciona al microprocesador para que pueda procesar los datos y generar los resultados esperados. El hardware por si solo no puede hacer nada, pues es necesario que exista el software, que es el conjunto de instrucciones que hacen funcionar al hardware.

Se pueden distinguir los siguientes tipos de software:

- Sistemas Operativos.
- Lenguajes de Programación.
- Software de Aplicación
- Software de Comunicación.

3.1 SISTEMAS OPERATIVOS.

El Sistema Operativo, también llamado O.S. (por sus siglas en ingles) es un conjunto de programas de base que controla a la computadora y que actúa de intermediario entre el usuario, la computadora y los programas de aplicación, traduciendo las órdenes del usuario o de las aplicaciones en instrucciones que puede entender el ordenador.

El O.S. es capaz de coordinar el equipo de la computadora y supervisar las funciones de entrada, procesamiento, almacenamiento y salida.

Otra definición posible y bastante aceptada define un sistema operativo como una capa compleja entre el hardware y el usuario, concebible también como una máquina virtual, que facilita al usuario o al programador las herramientas e interfaces adecuadas para realizar sus tareas informáticas, abstrayéndole de los complicados procesos necesarios para llevarlas a cabo.

Otra definición, más coloquial, denomina sistema operativo a todo el software que viene con la computadora antes de que se instale ninguna aplicación. Según esta

definición, orientada al usuario final, las herramientas de desarrollo no son necesarias y, sin embargo, se consideran elementos imprescindibles un amplio número de programas orientados a diferentes tareas, como editores de texto, administradores de archivos, navegadores, etc.

Funciones:

- a. Facilitar el uso de la computadora, ya que el software proporciona una interfaz que es mas favorable al usuario donde el recibe su información principal.
- b. El sistema operativo administra los recursos del computador en un intento por cumplir con los objetivos globales del sistema siendo el principal la eficiencia en su trabajo.
- c. Proporciona programas con un conjunto de servicios que pueden ayudar en el funcionamiento de muchas tares comunes.
- d. Comparte los recursos de la maquina entre varios procesos al mismo tiempo.

3.1.1 TIPOS DE SISTEMAS OPERATIVOS.

1. **SISTEMAS OPERATIVOS DE LINEAS DE COMANDO:** Son software estrictamente basado en texto y operado por comandos, a lo largo de los años ochenta, el sistema operativo mas difundido en las computadoras fue el MS-DOS.
2. **SISTEMA OPERATIVO DE AMBIENTE GRAFICO:** Estos son software que permiten la integración de texto con imágenes de alta resolución.

3.1.2 CLASIFICACIÓN DE LOS SISTEMAS OPERATIVOS.

La clasificación de los sistemas operativos se basan en la clase de interfaz del usuario que ellos proporcionan y gran parte de su terminología surge de la forma en que este se le presenta al usuario.

Una forma de clasificacion de los sistemas operativos puede ser de la siguiente forma:

De acuerdo al número de usuarios que el sistema pueda atender a la vez, se clasifican en:

- **Sistema de un solo trabajo:** Ejecuta solo un trabajo de un usuario a la vez, lo encontramos a menudo en microcomputadores y computadores personales ya que es el tipo de sistemas operativos mas antiguos; estos

son empleados normalmente en los computadores estándar, ya que a causa del tamaño limitado de su memoria y a la falta de canales y de otros recursos, sería difícil soportar más de un usuario en dicha máquina.

- **Sistema de multiprogramación:** Permite ejecutar varios trabajos de usuarios al mismo tiempo, su sistema operativo se encarga de intercambiar el C.P.U. entre los distintos trabajos de usuario proporcionando un ambiente adecuado de modo que los trabajos no se interfieran. Su propósito básico es mejorar el rendimiento permitiendo así que los recursos de sistema se compartan entre varios trabajos
- **Sistemas de multiprocesamiento:** Es semejante al anterior pero en este hay más de un C.P.U. disponible.

Por el tipo de acceso que proporciona al usuario:

- **Sistemas de procesamiento por lotes:** Se define como una secuencia de proposiciones de control almacenadas en forma legible para la máquina. El sistema operativo puede leer y ejecutar una serie de dichos trabajos sin otra intervención humana que las de ciertas funciones como el montaje de cintas y discos.
- **Sistemas de tiempo compartido:** Es aquel que proporciona acceso interactivo o conversacional a varios usuarios. Su sistema operativo ejecuta mandatos conforme los recibe, intentando dar a cada usuario un tiempo de respuesta razonablemente corto para cada mandato.
- **Sistema de tiempo real:** Está planeado para responder con rapidez a señales externas como las generadas por sensores de datos, y se emplean por ejemplo, en computadores vigilantes así como también en aquellos que controlan procesos críticos, en cuanto al tiempo, como la operación de un reactor nuclear o el vuelo de una nave espacial.

3.2 SOFTWARE DE APLICACIÓN.

Los programas de aplicación son un conjunto de instrucciones precodificadas que se diseñan para resolver problemas de diversos tópicos. El operador de la máquina sólo necesita seleccionar el programa de aplicación adecuado.

Es por ello que el software de aplicación se clasifica en:

- Software de aplicación Específica.
- Software de Aplicación General

3.2.1 SOFTWARE DE APLICACIÓN ESPECÍFICA.

Los diseñadores de software han creado muchos programas para uso específico, diseñado cada uno para resolver un tipo de problema, como ejemplo están los siguientes:

- Contabilidad de personal
- Programas educativos
- Contabilidad General.
- Cuentas por cobrar y por pagar
- Nómina
- Listas para correspondencia
- Programas de calificaciones para maestros
- Juegos de computadora.

Este tipo de software puede servir para satisfacer una necesidad relativamente estrecha. Por ejemplo, un paquete de cuentas por cobrar no puede utilizarse para llevar el control de entradas de personal.

Los programas financieros también pueden balancear chequeras, tarjetas de crédito, impuestos generados por las entradas, saldos y salidas personales.

El software educativo tiene un uso muy extenso, y cubre casi cualquier asunto, desde aprender el alfabeto hasta prerrepararse para un examen. Estos programas utilizan técnicas de simulación, exámenes prácticos y tutores computarizados.

Los niños pueden aprender computación y mejorar sus aptitudes de resolución de problemas.

Los estudiantes se benefician de la instrucción asistida por computadora a cualquier hora. Con programas de exámenes computarizados, los estudiantes pueden verificar y corregir sus respuestas instantáneamente. Más aún, la instrucción computarizada puede programarse específicamente como las tareas de un estudiante lo requieran. En un ejercicio práctico, por ejemplo, puede incluirse información adicional de refuerzo cada vez que el estudiante comete un error.

El software de negocios también es muy abundante. Existen paquetes disponibles para casi cualquier programa de administración, desde contabilidad general hasta nóminas. Muchos de esos programas han probado su confiabilidad saliendo airoso de los exámenes críticos por contadores públicos importantes.

3.2.2 SOFTWARE DE APLICACIÓN GENERAL.

El software para usos generales tiene cinco aplicaciones: hojas de cálculo electrónicas, procesadores de palabras, comunicaciones de datos, manejo de bases de datos y gráficas.

1. Hojas de cálculo electrónicas: Permiten manipular varios datos que pueden acomodarse en renglones y columnas. Estas hojas liberan de la tediosa tarea de recalcular a mano, ahorrando una gran cantidad de tiempo.

Algunas hojas de cálculo también permiten importar o intercambiar información con otros paquetes de aplicación. Esto permite realizar cálculos adicionales con datos ya procesados, e imprimir reportes. Los siguientes son solo algunos de los usos de las hojas de cálculo en labores administrativas:

- Requerimientos de materiales y mano de obra.
- Planeamientos de productos.
- Predicción de la producción.
- Comercialización de mercadeo.
- Adquisición y fusiones comerciales de bienes.
- Análisis de flujo efectivo.
- Registros de chequera.
- Balance de gastos personales.

2. Procesadores de palabras: Simplifica las tareas de escribir, editar e imprimir un documento. El procesamiento de textos o palabras no reduce el tiempo que se necesita para producir un informe completo desde el principio pero si proporciona los medios para revisar y generar documentos tan perfectos sin tener que estar tecleando borradores. Además de proporcionar revisión ortográfica, gramatical y sugerencia de sinónimos.

3. Comunicación de Datos: Permite a las personas transmitir información vital en poco tiempo por medio de modem o redes que permiten compartir recursos.

4. Manejo de Bases de Datos: Organiza, Actualiza y almacena registros y archivos en forma prácticamente ilimitada.

5. Gráficas: La gente puede comprender una gráfica bien hecha mucho más rápidamente que las estadísticas impresas de las que están basadas. Como formas de impartir información las gráficas pueden ofrecer ventajas similares sobre los métodos de la lectura o audibles.

Un paquete de software para manejo de base de datos ahorra una gran cantidad de tiempo al acceder información desde un archivo.

3.3 SOFTWARE DE COMUNICACIÓN.

Para lograr que dos o más computadoras se comuniquen entre sí, es necesario que exista tanto un medio *físico* como uno lógico que permita a ambas entablar comunicación. El *software de comunicación* se refiere al conjunto de programas que cumple la función lógica descrita anteriormente.

Actualmente existe en el mercado de las comunicaciones de datos una amplia variedad de software de comunicación, entre los que se encuentran:

- Smartcomm (Ambiente Windows).
- Quick Link II.
- Procomm Plus (Ambiente DOS).
- SLIP (Serial on Line Internet Protocol).
- TCP/ P
- SMTP(Simple mail transfer protocol)
- ftp
- PPT (Pert to Pert Protocol).
- Camaleon.
- Winsockets.

Los tres primeros productos no permiten copiar información mediante el programa FTP (File Transfer Protocol o Protocolo de Transferencia de Archivos), mientras que el resto *sí* lo permite.

El software de SLIP, desarrollado por la Universidad de Minnesota de los EUA permite tener acceso a todos los servicios de la red Internet desde cualquier computadora personal utilizando un módem y una línea telefónica.

3.4 LENGUAJES.

Es el software que constituye la herramienta para la confección de las diferentes aplicaciones y programas.

Los lenguajes de programación cierran el abismo de las computadoras, que sólo trabajan con números binarios, y los humanos, que prefieren utilizar palabras y otros sistemas de numeración. El expresar instrucciones completamente en un código binario es un proceso demasiado lento, difícil y muy sujeto a errores, aún para los más hábiles y experimentados programadores. Los lenguajes de programación permiten a las personas escribir instrucciones con base, en esencia en el idioma en inglés, números y símbolos matemáticos que las computadoras pueden convertir a código binario mediante programas denominados procesadores de lenguaje. Los programadores utilizan lenguajes de programación para elaborar paquetes de software como procesadores de palabras u hojas de cálculo, que pueden ser usados por una persona que no tenga experiencia en programación.

Los programadores suelen categorizar los lenguajes de programación de acuerdo con su legibilidad. Los lenguajes de bajo nivel son muy difíciles de entender, y requieren que los programadores codifiquen las instrucciones con gran detalle.

Los lenguajes de alto nivel utilizan enunciados parecidos al inglés que aún los programadores novatos pueden dominar con un mínimo de entendimiento.

Los procesadores de lenguaje traducen las instrucciones de un lenguaje de programación a código binario. El conjunto de instrucciones escritas en un lenguaje de programación es el código fuente. El conjunto de instrucciones binarias ya traducidas que la computadora puede ejecutar, es el código objeto.

Los procesadores de lenguaje pueden ser compiladores o intérpretes. Un compilador traduce completamente el código fuente de un programa a código objeto en lenguaje de máquina, mientras revisa en busca de errores que el programador hubiera cometido. El programa objeto, que es el que realiza todo procesamiento real, entra en operación si el código fuente no tiene errores. En contraste, un intérprete traduce el código fuente con una instrucción a la vez, del mismo modo en que una persona interpreta traduciría un discurso frase por frase.

Los lenguajes de programación son “lenguas” o “idiomas” ya establecidos, comprensibles tanto por el programador, como por la máquina, que permiten desarrollar los programas. Existen muchos y muy variados, y evolucionan constantemente, tanto en potencia como en facilidad de uso. La mayoría está escrito en Inglés, aunque cada vez son más visuales e intuitivos.

Ejemplos: Pascal, Turbo C++, CA-Clipper, Visual Basic, Visual C++, Java, Cobol, C, Fortran, HTML, Prolog, CSMP, SIMAN, Lisp, Lingo, SQL, Delphi, dBase, etc.

3.4.1 CLASIFICACIÓN DE LOS LENGUAJES DE PROGRAMACIÓN.

Los Lenguajes de Programación pueden clasificarse de varias manera, las principales son las siguientes:

- Por su Nivel.
- Por sus Aplicaciones principales.

3.4.1.1 CLASIFICACIÓN DE LOS LENGUAJES DE ACUERDO A SU NIVEL.

Esta clasificación se refiere a qué tan cercanos están los lenguajes a las características físicas de la computadora o qué tan próximos están a las necesidades del programador.

LENGUAJE MAQUINA.

Primer Lenguaje de Programación, el único que entiende directamente la computadora: utiliza el alfabeto binario formado por los símbolos 0 y 1 (apagado / encendido). Son realmente difíciles de aprender y de manejar, ya que no resultan nada cercanos al ser humano, pero son muy rápidos, ya que se ahorra mucho tiempo en la respuesta, puesto que se evitan las traducciones intermedias. A este lenguaje también se le considera un Lenguaje de Bajo nivel.

Una serie de lenguaje maquina son:

```
0010 0000 0000 1001
1001 0001 1001 1110
```

Como se puede observar, estas instrucciones serán fáciles de leer por la computadora y difíciles por un programador, y viceversa. Esta razón hace difícil de escribir programas en código o lenguaje maquina y requiere buscar otro lenguaje para comunicarse con la computadora, pero que sea más fácil de escribir y leer por el programador.

Para evitar la tediosa tarea de escribir programas en lenguaje máquina se han diseñado otros lenguajes de programación que facilitan la escritura y posterior ejecución de los programas.

LENGUAJE DE BAJO NIVEL (ENSAMBLADOR).

Surge con la evolución del Lenguaje Maquina, sustituye las cadenas de símbolos binarios por nemotécnicos: ADD, SUB, DIV.

Dependen de las características de cada maquina y sólo son comprensibles por especialistas. Aparecieron durante la primera generación de computadoras.

```
0110    1001    1010    1011
ADD     M,      N,      P
ADD suma                                DIV dividir
SUB resta                                LDA cargar acumulador
MPY multiplicar                          STO..almacenar
```

Una instrucción típica en ensamblador puede ser:

```
ADD x, y, z
```

Esta instrucción significa que se debe sumar los números almacenados en las direcciones x, y, y almacenar el resultado en la dirección z. El programa ensamblador traducirá.

```
01010010000000100010010100100110
00100100001000100010011100100101
11000101000000100010011100101000
101011010000000011001101
01010010000000010010100100000000
00100100001000010010010100101001
.
.
.
ETC.
```

Programa en código máquina

```
MOVE A, B
ADD C, A
COMP C, D
BE X
MOVE E, O
X, ADD A, E
.
.
.
ETC.
```

Programa en ensamblador

Después que un programa ha sido escrito en lenguaje ensamblador se necesita un programa llamado ensamblador que lo traduzca a código máquina.

LENGUAJE DE ALTO NIVEL.

Son desarrollados para el usuario de la computadora, permiten que los procedimientos se expresen con un estilo comprensible; sin embargo, la computadora no los entiende directamente. Algunos de estos lenguajes fueron desarrollados durante la Segunda Generación de computadoras y la mayoría durante la Tercera Generación, por ejemplo: Pascal, BASIC, Ada, FORTRAN, etc.

Usan traductores (compiladores o interpretes), son independientes de la máquina, las instrucciones no dependen del diseño de la máquina, los programas son portables o transportables, pueden ejecutarse en diferentes máquinas.

Por ejemplo, las líneas siguientes son: una línea de un programa en Turbo Pascal y su línea equivalente en pseudo código.

IF (x=y) AND (z=w) THEN WRITE ('esto es una prueba');

si (x=y) y (z=w) entonces ESCRIBIR ('esto es una prueba');

Estas líneas se pueden comprender fácilmente conociendo la traducción de las palabras inglesas IF(si), THEN (entonces), WRITE(escribir / imprimir), AND(y) y sin necesidad de mucha explicación.

Es posible escribir también operaciones como:

$Z := x + y$ *la suma $x + y$ se asigna a la variable z .*

COMPILADORES E INTERPRETES.

El código generado a través de los Lenguajes de Programación (Código Fuente) debe ser convertido en una aplicación que pueda ser ejecutada y utilizada. Los compiladores e interpretes son una especie de traductores o conversores del lenguaje de programación (comprensible para el ser humano) a un lenguaje comprensible por la computadora (cercano a la máquina).

TIPOS DE PROGRAMA:

Una vez escrito un programa éste puede estar en código fuente, código objeto o código ejecutable.

- **Código Fuente:** recibe este nombre el conjunto de instrucciones tal y como fueron escritos por el programador en el lenguaje de alto nivel (también se le conoce como programa fuente).
- **Código Objeto:** son las instrucciones de un programa cuando ya han sido traducidas al lenguaje máquina (también se le conoce como programa objeto).
- **Código Ejecutable:** es un código objeto que puede ser ejecutado por la computadora de inmediato (no todos los códigos objeto son ejecutables).

LOS INTERPRETES.

Traduce a lenguaje máquina, cada línea del programa fuente y la ejecuta de inmediato.

Traducen en Lenguaje de Alto Nivel a Lenguaje Máquina, se encargan de traducir **cada instrucción, una por una** (o cada línea de instrucciones) contenida en un programa escrito en cualquier lenguaje de alto nivel a instrucciones en código binario, comprensible por las computadoras.

Los interpretes no producen código objeto, por ello la ejecución de un programa requiere forzosamente del código fuente. Además, los programas en lenguaje interpretado se ejecutan con más lentitud que aquellos en lenguaje compilado.

LOS COMPILADORES.

A diferencia de los intérpretes que traducen instrucción por instrucción o línea por línea, los compiladores traducen **todo el programa entero** y luego lo integran generando un programa ejecutable por si solo. Con esto se consigue mas eficacia e independencia , ya que una vez compilado el programa, el compilador no tiene por qué estar presente, pudiendo ser transportado el programa ejecutable a cualquier computadora, sin ver el listado de ordenes, ni manejar el compilador.

El compilador es un lenguaje especial que traduce a un programa fuente (escrito en lenguaje de alto nivel) a un código objeto.

El código o programa fuente es el conjunto de instrucciones en lenguaje de alto nivel.

Un compilador analiza las líneas del programa e identifica ciertas estructuras para traducirlas adecuadamente a lenguaje maquina. Una vez que se ha efectuado la compilación puede prescindirse del programa fuente.

El proceso de compilación del Programa Fuente se verifica en cuatro etapas:

- **Análisis de léxico:** durante esta etapa se verifica que las palabras utilizadas en el programa fuente pertenezcan al lenguaje.
- **Análisis de sintaxis:** esta parte de la compilación examina cada instrucción del programa fuente para que las instrucciones estén organizadas correctamente.
- **Generación de código intermedio:** este produce un código simbólico que permitirá efectuar la equivalencia con el lenguaje maquina.
- **Generación de código maquina:** en esta ultima etapa se elabora el programa objeto.

El código o programa objeto son las instrucciones de un programa que ya han sido traducidas a lenguaje maquina.

El código ejecutable es un código objeto que puede ser ejecutado por la computadora de inmediato.

A modo de resumen podemos decir que los compiladores producen programas que, una vez depurados y puestos a punto, son ejecutables directamente y los

intérpretes, aunque tienen unas necesidades menores de memoria a la hora de traducir los programas fuente, nos hipotecan para siempre en cuanto al tiempo de ejecución de los programas.

Existen en el mercado compiladores para todos los lenguajes de alto nivel; los intérpretes en cambio sólo se han desarrollado para algunos lenguajes de programación siendo los más conocidos los de BASIC.

LENGUAJES DE PROGRAMACIÓN

Un lenguaje usado para escribir instrucciones para la computadora. Permite que el programador exprese el procesamiento de datos de una forma simbólica sin tener en cuenta los detalles específicos de máquina. Las sentencias que escribe el programador se denominan lenguaje fuente, y son traducidas al lenguaje de máquina de la computadora por medio de programas llamados ensambladores, compiladores e intérpretes. Por ejemplo, cuando un programador escribe multiplicar horas por tarifa, el verbo MULTIPLICAR debe traducirse a un código que signifique multiplicación y HORAS y TARIFA deben convertirse en direcciones de memoria donde están verdaderamente almacenados esos elementos.

Como los lenguajes humanos, cada lenguaje de programación tiene su propia gramática y sintaxis. Existen muchos dialectos del mismo lenguaje, y cada dialecto requiere su propio sistema de traducción. El ANSI (American National Standards Institute - Instituto Americano de Estándares Nacionales) ha establecido estándares para muchos lenguajes de programación, y cada lenguaje del que se dice que se adhiere a las normas ANSI, es un lenguaje libre de dialectos. Sin embargo, la inclusión de nuevas características en los estándares ANSI puede llevar años, y, como resultado de ello, emergerán siempre nuevos dialectos.

Los lenguajes de programación pueden englobarse en dos grandes categorías: lenguajes ensambladores de bajo nivel y lenguajes de alto nivel. Los lenguajes ensambladores están disponibles para cada familia de CPU, y traducen una línea de código en una instrucción de máquina. Los lenguajes de alto nivel traducen sentencias de programación en varias instrucciones de máquina.

En la actualidad, los lenguajes de alto nivel más comunes para negocios son el BASIC, COBOL, dBASE y Pascal. El FORTRAN Y El APL se utilizan en áreas científicas, y el C se emplea extensamente en software comercial. El LISP y el Prolog son usados en aplicaciones de inteligencia artificial, el FORTH es utilizado en el control de procesos y el Ada es empleado por el Department of Defense. La nueva ola es la programación orientada a objetos, y el C++ está ganando popularidad.

3.4.1.2 CLASIFICACIÓN DE LOS LENGUAJES DE ACUERDO A SUS APLICACIONES PRINCIPALES.

Esta clasificación se refiere a las ventajas que tienen los lenguajes para resolver ciertos problemas específicos.

Los principales lenguajes pueden clasificarse de la siguiente manera:

- **Lenguajes Clásicos o Tradicionales.**
- **Lenguajes Visuales (orientados a objetos).**
- **Lenguajes de Internet.**

LENGUAJES CLÁSICOS:

Estos lenguajes están basados en un lenguaje cercano al ser humano en el que se escribe el código necesario para realizar las operaciones que se requieren (miles y miles de líneas generalmente), que posteriormente es traducido o compilado generando un programa ejecutable. Los más conocidos son: el PASCAL, BASIC, C. Dependiendo del programa que se requiera realizar pueden ser mejores unos u otros.

LENGUAJES VISUALES:

Son lenguajes más avanzadas que los lenguajes clásicos y están basados en objetos. Cada entidad del programa (eventos, acciones, datos, etc.), es un objeto sobre el que se definen operaciones. Esto permite almacenar los objetos (con todo su código de programa), en una serie de librerías y utilizarlos cuando sea necesario, ahorrando al programador el trabajo de tener que volver a programarlos. Son lenguajes muy intuitivos y fáciles de utilizar, y sustituyen las tediosas líneas de código, por sencillas y directas representaciones gráficas.

LENGUAJES DE INTERNET:

Son los lenguajes específicos para la creación de páginas Web de Internet, y realizar su programación (motores de búsqueda, seguridad, establecimiento de comunicaciones, etc.).

Son la última generación de lenguajes. Existen distintos tipos, dependiendo del grado de especialización, aunque algunos, como el HTML, son lenguajes basados en sencillas marcas o TAG.

La siguiente tabla muestra los principales lenguajes de programación y su utilidad.

Lenguajes.	Utilidad.
BASIC, PASCAL, Ada, C y C++ Logo	Aprendizaje de programación. Programación de sistemas. Lenguaje didáctico.
COBOL, RPG, Natural, SQL	Gestión de empresas.
Algol, FORTRAN, APL	Calculo numérico.
CLIPPER, ACCESS, dBASE, Delphi, SQL	Bases de datos.
Smalltalk, Actor, C++, Eiffel	Programación orientada a objetos (OOP).
Visual Basic, Visual C	Programación Windows.
HTML, JAVA, PERL	Internet (páginas WEB).
Lingo	Programas multimedia.
Prolog, Lisp	Inteligencia artificial.
VRLM	Realidad virtual.
Mapper, Cogen, Linc	Lenguajes de cuarta generación 4GL.

Por ultimo para la resolución de un problema, deben seguirse por lo menos los siguiente pasos:

1. Definición del Problema.
2. Análisis del Problema.
3. Diseño del algoritmo.
4. Programación (transformación del algoritmo en un programa).
5. Ejecución y validación del Programa (Pruebas).
6. Mantenimiento del Programa.

Estas fases constituyen lo que se denomina: Ciclo del Vida del Software o Ciclo de Vida del Desarrollo de un Sistema, son realizadas por seres humanos, solo la ejecución de los programas es realizada por la computadora.

1. **Definición del Problema.** El programador debe reconocer cuál es el problema, estableciendo con claridad cuales son los datos disponibles y a qué resultados se desea llegar.
2. **Análisis del Problema.** Es un estudio profundo del problema y de la situación que lo rodea. El programador deberá entenderlo por completo y proponer una solución para poder transformar los datos disponibles en resultados.
3. **Diseño del algoritmo.** Se planea la estrategia que permitirá alcanzar la solución del problema.

4. **Programación** (transformación del algoritmo en un programa). Consiste en traducir el algoritmo diseñado a un lenguaje de programación y aplicarlo en una computadora.
5. **Ejecución y validación del Programa** (Pruebas). Verifican su funcionamiento para corregir posibles errores.
6. **Mantenimiento del Programa**. El programador se encarga de modificar el programa según vayan apareciendo necesidades adicionales.

CAPITULO 4

Análisis y Diseño de Bases de Datos.

4. ANÁLISIS Y DISEÑO DE BASES DE DATOS

Las bases de datos son esenciales para el depósito de datos en el sistema de información.

El fácil acceso a una variedad de datos desde cierto número de lugares permite que el sistema de información responda con rapidez a las necesidades de quienes toman decisiones en la organización, en tanto que un acceso deficiente obstaculiza las respuestas o el proceso de la información dentro de la organización.

Al crear una base de datos con una correcta estructura y por lo tanto mayor rapidez las decisiones no se retardarán y se tomarán datos completos, lo que nos conducirá a obtener una información concisa, precisa y sobre todo veraz, lo cual será reflejado en una excelente toma de decisiones, no solo a nivel gerencial, sino dentro de todos los ámbitos de la organización.

La estructura de la base de datos debe ser flexible ante las necesidades de cambio de una organización; por ello al crecer una base de datos se debe tomar en cuenta las necesidades actuales del sistema y además visualizar los posibles crecimientos de la misma, tomando en cuenta éstas situaciones cuando la base de datos requiera almacenar nuevos datos y diseñar nuevas relaciones a partir de las nuevas decisiones o incluir opciones para permitir que estos cambios se realicen.

El desarrollo de los sistemas de Bases de datos creó la necesidad de usar procedimientos de diseño que pudieran enfrentar los problemas que según la complejidad de las bases de datos y por la integración de estas con las funciones del usuario.

Es por ello que durante el diseño de la base de datos se debe tomar en cuenta los siguientes puntos:

- La integración más estrecha de las funciones organizacionales.
- Una variedad de usuarios, de los cuales cada uno emplea diferentes subconjuntos y relaciones de datos.
- Las diferentes estructuras físicas posibles a fin de implementar estos sistemas, la diversidad, prevenir la disponibilidad de sistemas manejada en la base de datos.
- Los requerimientos del sistema del usuario, que son menos estables y más fáciles de cambiar que los primeros programas.

4.1 SISTEMAS MANEJADORES DE BASES DE DATOS (SMBD).

El SMBD (Sistema Manejador de Base de Datos) proporciona los términos y las estructuras apropiadas para un sistema físico. El modelo de datos es el término aceptado para las estructuras de datos que facilita un SMBD a la interfaz del usuario. Los modelos de datos se seleccionaron a fin de proporcionar construcciones que puedan ser moldeables de acuerdo a los problemas que pueda tener el usuario. Para esto se debe de realizar un mapa o representación de las relaciones de los datos que el usuario maneja. Además, el modelo de datos representa, de modo directo, el sistema del usuario.

4.1.1 MODELOS DE DATOS.

En gran parte los sistemas manejadores de bases de datos, funcionan bajo los modelos relacional, jerárquico y de red, ya que el primero ofrece una interfaz tabular para el usuario. El argumento a su favor es que, como mucha gente está familiarizada con las tablas y las usa como parte de su trabajo diario, no les es difícil usar las estructuras tabulares para modelar sus empresas. Los otros modelos de datos más conocidos, los modelos jerárquicos y de red, se basan en registros. En ambos modelos, el modelo de empresa del usuario se convierte en un tipo de registro enlazado. El modelo de enlace es una relación padre-hijo o una relación propietario-miembro entre tipos de registros. Los tipos de registro en una estructura jerárquica sólo pueden tener un padre, mientras que los tipos de registro en una estructura en red pueden tener más de un padre.

Los modelos de datos jerárquico y de red pueden modelar una gran variedad de problemas, donde es factible clasificar en forma jerárquica sus entidades. Las estructuras organizacionales son un ejemplo.

La disposición de una interfaz natural tiene una connotación importante en ambientes donde más de un usuario puede acceder la base de datos. Los usuarios pueden ver diferentes conjuntos de objetos como los más trascendentes y dar más importancia a diferentes relaciones. Así, cada usuario puede desear la interfaz requerida para resaltar los diferentes aspectos naturales de la base de datos. El término *vista de usuario* se emplea con regularidad para referirse a la interfaz natural proporcionada para un usuario en particular. Un sistema manejador de bases de datos debe ofrecer diferentes vistas de la misma base de datos para diferentes usuarios.

4.1.2 INDEPENDENCIA DE DATOS.

Para proporcionar un modelo natural, el modelo de datos debe ser independiente de las estructuras dependientes de computación como índices, ordenamientos y trayectorias de acceso físico. Estas características de computación no son naturales para los problemas del usuario. Ellas pueden distraer a los diseñadores, quienes sin advertirlo hacen mapeos que satisfacen algunos aspectos de la

computación pero que no representan con naturalidad la empresa. El resultado puede ser sistemas difíciles de mantener y usar. El objetivo de los sistemas manejadores de bases de datos es presentar una interfaz para el usuario, que resalte la estructura lógica de un problema del usuario y sea independiente de la estructura física de la computadora.

La independencia entre las estructuras lógica y física a menudo se llama *independencia de datos física*. Tiene una propiedad importante: la capacidad de cambiar la estructura física sin afectar la lógica. Las modificaciones físicas se deben de realizar si cambian los patrones de uso. Por ejemplo, es factible añadir nuevos índices si un ítem se utiliza con frecuencia como una clave de acceso. Como los programas se refieren a la estructura lógica, no resultarán afectados con los cambios físicos. Los programas, entonces, serán independientes de la estructura física y se mantendrán mucho más fácilmente.

Análisis y diseño de bases de datos.

La independencia de los datos físicos tiene un efecto significativo en el proceso de diseño de la base de datos; es decir, separa el diseño en dos pasos, el diseño lógico y el físico, en el primero, el modelo de la empresa se convierte en una estructura de base de datos lógica. Esta conversión no considera la naturaleza detallada de la estructura de datos físicos eventuales. En el segundo, esto es, el diseño físico, las estructuras de datos físicos se cambian y se adicionan a la estructura de la base de datos lógica seleccionada durante el diseño lógico.

4.1.3 REESTRUCTURACIÓN DE LA BASE DE DATOS.

En muchas ocasiones los requerimientos de usuario cambian durante la vida de la base de datos, entonces es posible cambiar la estructura de la base de datos añadiendo nuevas entidades y relaciones o que cambien las relaciones existentes. Las bases de datos se pueden cambiar para satisfacer los requerimientos alternas, al ofrecer varias utilerías para reestructurar la base de datos existente. Lo ideal es que estos cambios representen un efecto mínimo para el sistema existente. En particular, los programas que no requieran los cambios no deberán afectarse.

4.1.3.1 ARQUITECTURA DE SMBD.

La arquitectura de un sistema manejador de bases de datos comprende a su objetivo principal que es proporcional al facilidad para:

- Definir las estructuras lógicas de las bases de datos
- Definir la estructura física seleccionada
- Definir las vistas del usuario
- Accesar la base de datos definida
- Definir las estructuras para almacenar los datos

En conjunto los anteriores componentes conforman la arquitectura de una base de datos.

4.1.3.2 AMBIENTE OPERACIONAL DE UNA BASE DE DATOS.

En un ambiente operacional de bases de datos es casi siempre accedido desde una aplicación manejada por el usuario la cual accede de acuerdo a las funciones del usuario hacia una parte de las bases de datos por ejemplo la actualización de una cuenta bancaria, en ese momento su aplicación invoca a la base de datos de clientes y despliega los datos del cliente y el desde su aplicación puede modificar los estados de cuenta por ejemplo de su cliente.

Los sistemas operativos ofrecen diferentes tipos de control para ejecutar un programa. La diferencia está en el número de programas que pueden activar al mismo tiempo, en la cantidad de usuarios que pueden correr el mismo programa activado y en las interacciones del programa de aplicación con los otros componentes del sistema computacional. Las clases de control del programa que proporciona el sistema operativo forman el *modo de procesamiento*. El modo de procesamiento necesario para aplicaciones particulares depende del ambiente operacional del usuario.

Se pueden distinguir dos ambientes operacionales:

- Una base de datos que habrá de usar sólo una o pocas personas.
- Una base de datos que es una parte integral de una operación de negocios y que, por tanto, debe ser accesible para muchos usuarios, quienes pueden estar en varios lugares.

4.1.3.3 INTEGRIDAD DE LA BASE DE DATOS.

La integridad de los datos tiene un significado especial en un ambiente multiusuario, porque se pueden perder si más de un programa opera concurrentemente en el mismo ítem de datos. Esto se puede prever si se utiliza un control de concurrencia el cual es necesario para asegurar la exactitud de la base de datos. En un ambiente concurrente, la exactitud se define como el estado final de la base de datos debe ser el mismo, como si los programas se ejecutaran en serie, para evitar problemas de pérdida de datos e información.

El método más común de control de concurrencia utiliza bloqueos. Se mantiene un bloqueo para cada componente de la base de datos. Un programa obtiene la totalidad de bloqueos antes de realizar cualquier actualización. El control de concurrencia debe determinar que, por lo menos, un programa obtenga el bloqueo para alguna parte de la base de datos. Si un programa desea mover un ítem bloqueado al almacenamiento funcional del programa, debe esperar a que el programa anterior libere el bloqueo. La implementación depende de cómo se logre

esto. En muchas implementaciones, los programas de usuario tienen comandos para bloquear los registros requeridos antes de actualizarlos. Es responsabilidad del programa obtener el bloqueo.

4.1.3.4 DISTRIBUCIÓN DE LA BASE DE DATOS.

En la mayor parte de los lugares que se manejan grandes cantidades de información y datos se recolectan y usan datos en muchos lugares. Algunos datos recolectados en un lugar particular pueden ser de interés sólo en esa ubicación. Otro dato quizá sea de importancia en más de un lugar. Además, hay ciertas ventajas en mantener algunos datos del proyecto cerca de su ubicación de uso, mientras que otros pueden estar en forma central. Una base de datos que mantiene más de una ubicación se llama *base de datos distribuida*. La distribución de la base de datos introduce nuevos requerimientos para el proceso de diseño y las necesidades técnicas del programa SMBD.

4.1.3.5 CONSIDERACIONES DE DISEÑO EN BASES DE DATOS DISTRIBUIDAS.

Las dos decisiones principales que se deben realizar en el diseño de bases de datos distribuidas son: dividir las bases en partes y saber dónde almacenar cada una. Los términos para estas actividades de diseño son *fragmentación* y *asignación*. El primero describe la subdivisión lógica de una base de datos en partes, cada una de las cuales es un fragmento. El segundo se refiere a la decisión de dónde almacenar cada parte.

Una base de datos se puede fragmentar de la siguiente forma:

1. Haciendo cada relación un fragmento separado.
2. Haciendo series de renglones de una relación un fragmento separado.
3. Haciendo conjuntos con los atributos de una relación un fragmento separado.
4. Combinando cualesquiera de los puntos anteriores.

4.1.3.6 HERRAMIENTAS CASE Y DICCIONARIOS DE DATOS

Las herramientas CASE han introducido una dimensión nueva por completo en la administración de bases de datos y así como en el rol de los diccionarios de datos en esta área. Podemos extender la idea de los diccionarios de datos para incluir las herramientas CASE, que posibilitan la integración de los requerimientos del usuario en el diccionario de datos, que en sí mismo incluye una herramienta CASE como interfase o es la herramienta CASE en sí misma. Esto ofrece dos nuevas oportunidades:

1. La repercusión de un cambio en la base de datos para los requerimientos del usuario se puede evaluar con rapidez.
2. En algún momento es posible generar en forma directa un sistema desde el modelo del usuario.

En relación con la segunda oportunidad, los sistemas del usuario se pueden generar directos desde el modelo y, lo que es más importante, también es factible realizar con rapidez cualquier cambio en los requerimientos. Esta capacidad se puede efectuar con mayor facilidad mediante las herramientas CASE, con las cuales es relativamente sencillo generar el modelo de datos desde el modelo E-R(Entidad Relación). Lo que resta es proponer un método para especificar el comportamiento o las actividades del sistema y convertir tales especificaciones directamente en programas de computación.

Como resultado, los diccionarios de datos se están convirtiendo en el depósito de toda la información relacionada con el sistema, empezando con los modelos del usuario e incluyendo los niveles de diseño hasta las definiciones físicas reales. Quienes desarrollan programas están comenzando a llamarlo enciclopedia del sistema.

4.2 SELECCIÓN DE UN SMBD.

Para seleccionar un sistema manejador de base de datos es necesario que cumpla con un conjunto bien definido de las necesidades operacionales y funcionales que requiere el usuario. Por tanto, antes de que se pueda seleccionar el SMBD, se debe determinar los requerimientos del SMBD, como:

- El tipo de usuario final de la base de datos y, en consecuencia, los requerimientos del lenguaje de acceso.
- Los requerimientos de control de la privacidad.
- El tipo de reestructuración que soportará el SMBD.
- La cantidad de datos que soportará el SMBD.
- Los requerimientos para controlar la recuperación y la concurrencia.
- Los requerimientos para monitorear el desempeño.
- Los métodos de acceso generales que soporta el SMBD.
- La localización donde se necesitan los datos.

Cualquier SMBD que no satisfaga estas necesidades, en general, ya no se considera más.

Con frecuencia, se usan puntos de referencia para valorar y comparar el desempeño técnico en diferentes SMBD. La idea de los puntos de referencia es

escribir los mismos programas de prueba para todos los candidatos del SMBD con el fin de comprobar las características críticas.

4.2.1 MONITOREO DE LA BASE DE DATOS.

El monitoreo se realiza después de haber implementado la base de datos ya que sirve para medir la cantidad de tráfico que hay en las operaciones concurrentes a la base de datos. La mayoría de los sistemas manejadores de bases de datos tienen sistemas o programas para su monitoreo y miden:

El número de registros de cada tipo accesado.
El número de veces que se recorre un enlace.
El uso de varios índices.

Los datos arrojados por el monitoreo arrojan por lo general los cambios en los patrones de uso o comparar el uso de los recursos reales con el empleo esperado. Y mediante a estos resultados el administrador de la base de datos puede realizar variaciones a las estructuras para una mejor utilización de la base.

4.2.2 MEDICIÓN DEL DESEMPEÑO.

Para estimar el desempeño de las bases de datos hay que tomar en cuenta dos estimaciones del desempeño principales son el uso del almacenamiento y el tiempo de respuesta de las operaciones críticas.

La estimación del almacenamiento indica la cantidad de almacenamiento secundario necesario para guardar la base de datos y el tiempo de respuesta sirve como una medida del desempeño en las operaciones en línea. Se valora al calcular el número de registros físicos que se transmiten mediante la interfaz del registro físico. En los sistemas centralizados, la cantidad de registros transmitidos debe ser adecuada para el tiempo de respuesta de la operación y puede servir para comparar sistemas alternativos, al usar el desempeño de las transacciones como un criterio. El diseño distribuido también necesita estimar el número de registros transferidos. El objetivo es reducir este número.

Otro valor importante, en especial en sistemas más grandes, es la utilización del canal de disco. Los programas soportan varios canales y cada uno tiene una capacidad limitada para transferir registros. Por lo tanto, los diseñadores deben verificar que no sea sobrepasada esta capacidad.

Cada paso de una trayectoria de acceso se usa para estimar el número de registros lógicos accesados en cada paso. Luego, con esta estimación se determina el número de transferencias de registros físicos. Estos dos pasos se llaman análisis lógico y análisis físico, en este texto.

4.2.3 REQUERIMIENTOS DE ALMACENAMIENTO.

Para realizar una estimación de los requerimientos de almacenamiento

Las estimaciones de los requerimientos de almacenamiento usan la tabla de requerimientos de almacenamiento y cada columna en esta tabla sirve para calcular el tamaño de un archivo. A continuación, se describen los renglones de la tabla:

- *Tamaño del registro de datos.* La cantidad de almacenamiento para cada registro de un archivo; la suma de todos los tamaños de los ítems de datos y los apuntadores en el registro.
- *Ocurrencias del registro.* El número de ocurrencias del registro en el archivo.
- *Almacenamiento de datos.* El número de ocurrencias multiplicado por el tamaño del registro de datos.
- *Tamaño del bloque en el hardware.* El tamaño, en términos de varias unidades de almacenamiento de un bloque de hardware; un sector del disco, una pista y los segmentos son tres ejemplos. Si el tamaño es variable, se indica un rango. Se considera que el contenido de un bloque se transfiere como un registro físico.
- *Factor de bloqueo.* El número de registros lógicos almacenados en un bloque de hardware o transferidos como registros físicos. Se indica un rango variable si está soportado en el programa de organización del archivo.
- *Bloques del hardware de datos.* El número de bloques del hardware necesarios para almacenar los datos en el archivo. Es el número de ocurrencias en el registro dividido entre el factor de bloqueo.
- *Bloques de sobreflujo.* El número de bloques del equipo asignados para el área de sobre flujo de un archivo.
- *Bloques del hardware para el índice.* El número de bloques de hardware necesarios para almacenar el índice de un archivo.
- *Los bloques totales en el hardware.* Los bloques de hardware para los datos, más los bloques de hardware para los índices, más los espacios de sobre flujo para un archivo.
- *Almacenamiento físico total.* Los bloques totales en el hardware multiplicados por el tamaño del bloque de hardware para un archivo.

4.2.4 METODOLOGÍAS DE DISEÑO.

Selección de las metodologías del diseño

- Las técnicas utilizadas en el diseño de las bases de datos se pueden combinar en las metodologías de diseño. El objetivo de la integración es crear un proceso de diseño completo, que reduzca los requerimientos de los datos del usuario a una estructura de base de datos en computadora. Ese proceso, usualmente, se llama metodología del diseño e incluye:

Una secuencia de pasos de diseño; cada uno resuelve un problema de diseño bien definido.

- Las técnicas para solucionar los problemas de diseño en cada paso, usando las semánticas apropiadas para los pasos.
- Las técnicas para combinar todos los pasos en una metodología de diseño integrada, permitiendo que los resultados de un paso se usen como los datos de entrada del paso siguiente.
- Un método de documentación para mantener un seguimiento de todas las decisiones y los modelos elaborados durante el diseño.

Las metodologías de diseño constituyen un conjunto de técnicas que se aplican de acuerdo con una secuencia de pasos. Cada paso tiene una meta específica y a menudo usa diferentes técnicas. La determinación de los requerimientos del usuario, por ejemplo puede emplear un conjunto de técnicas, la definición de una base de datos un conjunto de técnicas muy diferentes y la optimización de la estructura física de la base de datos quizá necesite otra.

Estas técnicas pueden dar como resultado muchos datos, y la coordinación de actividades. Por esta razón las metodologías de diseño se apoyan en las llamadas herramientas CASE.

4.2.5 TÉCNICAS DE DISEÑO

Selección de las técnicas de diseño

Para evitar problemas se deben seleccionar las técnicas de diseño, utilizando los términos apropiados en cada etapa. Estas técnicas también deben permitir al diseñador satisfacer todos los criterios de diseño en una etapa particular y se necesitan seleccionar para que el resultado de una etapa se use directamente como los datos de entrada para la etapa siguiente.

La elección de las técnicas de diseño se debe basar en la teoría de diseño del sistema computacional, que proporciona los principios para usar estas opciones.

Dicha teoría todavía se está desarrollando y por tanto sólo cubre parte del proceso de diseño. La teoría relacional proporciona las bases para estructurar los datos.

La mayoría de las metodologías combinan el análisis relacional con un modelo semántico, que captura la información acerca del sistema. El modelo semántico, entonces, se convierte en un modelo relacional y se revisa que esté en la forma normal más alta. El siguiente paso convierte este modelo en un modelo de implementación. Aumentar la formalización en esta área, permite que las herramientas CASE lleven a cabo la conversión, a menudo con la asistencia del diseñador. Además del modelado de datos, la mayoría de las metodologías también incluye técnicas para el proceso de modelado y el flujo de datos. Estas técnicas se deben integrar con el análisis y el diseño de la base de datos, en una metodología de diseño.

Las técnicas de diseño son de dos clases:

- Técnicas de análisis de datos para la semántica de datos del usuario.
- Técnicas de diseño técnico para convertir los resultados del análisis en una implementación técnica de la base de datos.

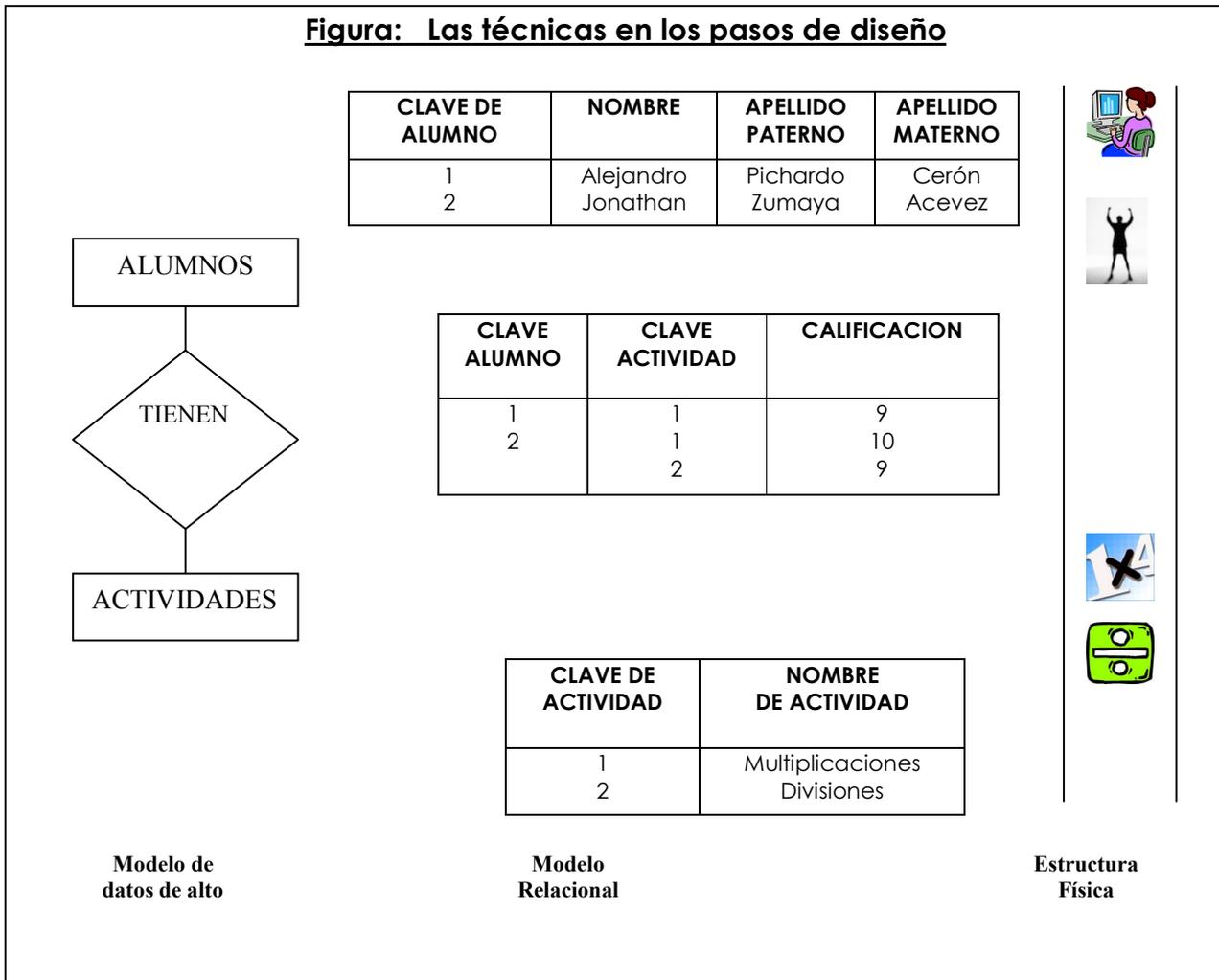
Para garantizar un almacenamiento satisfactorio y un acceso eficiente a nuestra base de datos deberemos utilizar técnicas que remuevan los datos redundantes y que implementen el uso eficiente de los recursos disponibles, tomando en cuenta la identificación de necesidades de los usuarios y la estructura adecuada de los datos.

La siguiente figura muestra el resultado de aplicar estas técnicas: un modelo de alto nivel de los datos mismos, que son las entidades de mayor interés. En este caso las entidades son ALUMNOS y ACTIVIDADES y el hecho es que los alumnos realicen actividades para reafirmar conocimientos. La técnica que se representa en este modelo se conoce como análisis de datos o a veces técnica de análisis semántica; cuyo objetivo es capturar todos los requerimientos de datos del usuario.

El siguiente paso convierte el modelo de alto nivel en una serie de tablas que describen mejor estas entidades procurando evitar redundancias en los datos por medio de la teoría relacional. En esta etapa el modelo que representa los datos se conoce como modelo relacional.

Por último, las tablas se convierten en estructuras físicas de la base de datos. Entonces se usa una nueva serie de criterios relacionados con aspectos que minimizan el almacenamiento de datos o aseguran un buen desempeño en respuesta a las consultas.

Figura: Las técnicas en los pasos de diseño



4.2.6 ETAPAS DE DISEÑO.

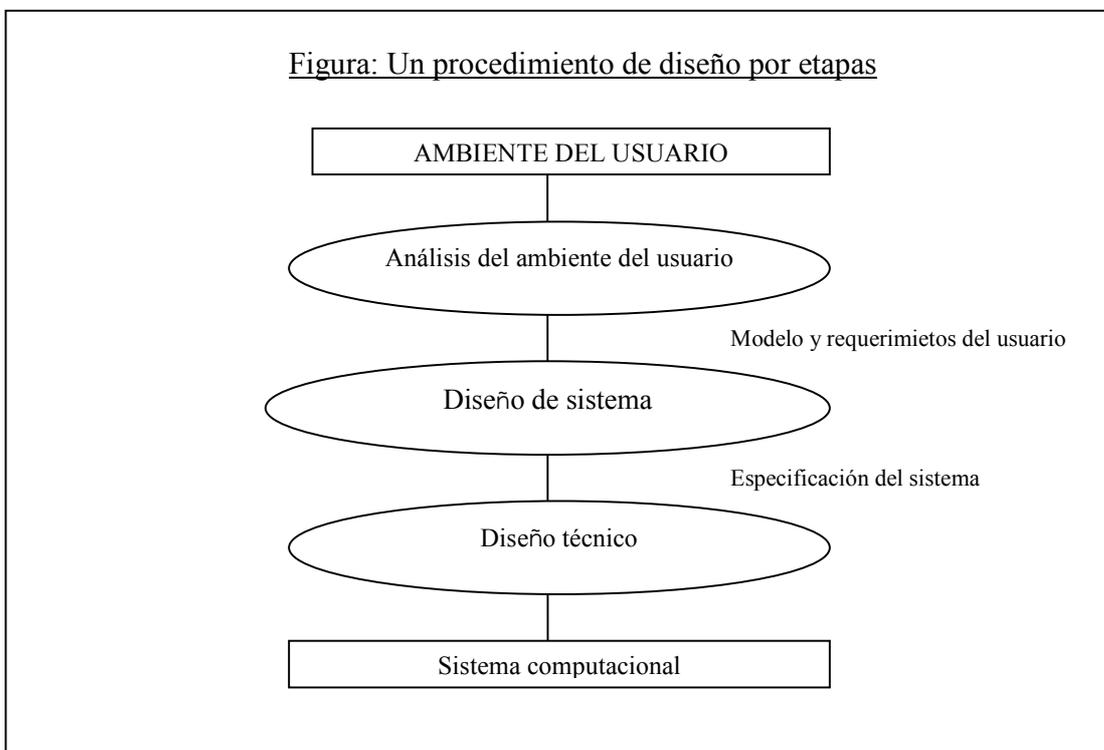
El desarrollo de sistemas con bases de datos crea la necesidad de usar procedimientos de diseño que pudieran enfrentar los problemas que surgen por la complejidad de éstas y por la integración de las mismas con las funciones del usuario.

Anteriormente el diseño de base de datos se basaba en la selección de la mejor estructura física para producir reportes, por lo que, determinaban el número de archivos para cada función del sistema y después seleccionaban las técnicas apropiadas de indización para un desempeño satisfactorio.

Pero este criterio actualmente no o es válido. Ahora el diseñador debe tomar en cuenta:

- La integración más estrecha de las funciones organizacionales.
- Una variedad de usuarios, de los cuales cada uno emplea diferentes subconjuntos y relaciones de datos.
- Las diferentes estructuras físicas disponibles a fin de implementar estos sistemas; la diversidad proviene sobre todo de la disponibilidad de sistemas manejadores de bases de datos.
- Los requerimientos del sistema del usuario, que son menos estables y más fáciles de cambiar que todos los programas.
- Eliminar la redundancia de datos.
- Proporcionar estructuras de datos estables que se cambien con facilidad de acuerdo a requerimientos del usuario.
- Permitir a los usuarios hacer consultas sobre los datos.
- Mantener relaciones complejas entre los elementos de los datos.
- Soportar una gran variedad de necesidades de toma de decisiones.

Por lo tanto, al realizar el diseño de base de datos debemos tener en cuenta el proceso mostrado en siguiente figura; el cual soporta la tendencia contemporánea de diseño separando las decisiones de requerimientos del usuario de las decisiones de diseño técnico.



La primera etapa es analizar el ambiente actual del usuario, incluyendo los datos y los flujos de datos. El resultado de esta etapa es un modelo del ambiente del usuario y sus requisitos (Análisis del ambiente del usuario).

Una vez que se produce el modelo y los requerimientos del usuario el diseñador debe planear un nuevo sistema para satisfacerlos (Diseño del sistema).

Para lograrlos, el diseñador puede examinar las estructuras y los procedimientos actuales e identificar cualquier problema existente. Se pueden delinear entonces las estructuras y procedimientos.

Si los usuarios están de acuerdo con las propuestas, estas pueden incluirse en la especificación del sistema, las cuales deben tener las siguientes características (Diseño Técnico):

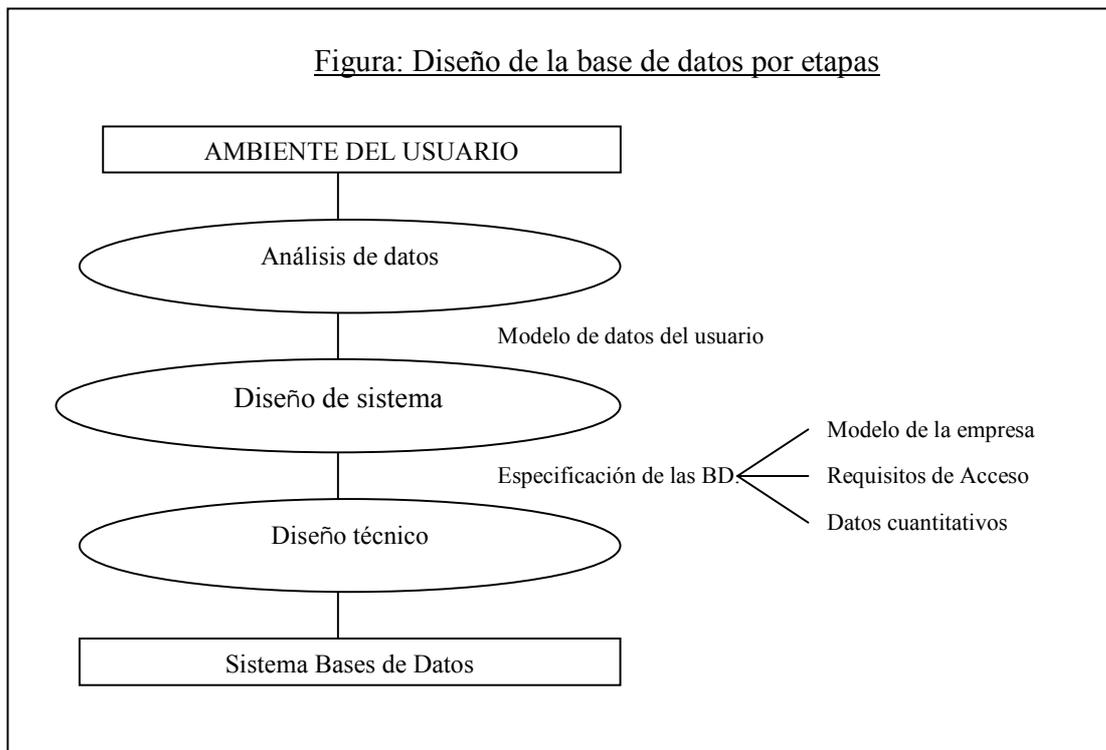
- Ser claro y preciso.
- Tener una representación correcta de los requerimientos.
- Usar técnica de documentación que entiendan con claridad los usuarios y el equipo técnico de computación.

El último paso de este procedimiento es convertir la especificación en un sistema de computación (Sistema Computacional).

La separación del análisis y el diseño técnico se hace de manera deliberada. Su objetivo es retardar tanto como se posible la consideración d restricciones técnicas, tales como la selección de la clave de acceso al archivo o los comandos de acceso del programa. Es decir, si estas limitaciones se consideran demasiado pronto, se pueden imponer decisiones innecesarias a los usuarios.

4.3 REQUERIMIENTOS DE ESPECIFICACIÓN Y DISEÑO DE BASES DE DATOS.

Una vez definidas las etapas de diseño, podremos observar como funciona el análisis los datos dentro del mismo, lo que nos conducirá a lograr una correcta estructuración de la base de datos.



La primera etapa consiste en analizar los datos en el sistema del usuario. Este paso, que por lo regular se llama análisis de datos y es parte medular del análisis del sistema total ya que produce un modelo de datos del usuario que se emplea en la etapa de diseño del sistema para producir una especificación de una base de datos.

1. El modelo de la empresa:

- a. Los ítems de datos en la empresa.
- b. Las relaciones de datos en la empresa.

2. El acceso a los requerimientos:

- a. La manera en la cual se van a almacenar o acceder los datos.
- b. Que el acceso sea en línea o por lotes.

3. Los datos cuantitativos:

- a. La frecuencia con la que se accesan los datos.
- b. Tiempo y lugar en que los datos deben estar disponibles.
- c. Volumen de los datos.
- d. Lugar en el que se capturan los datos.

Estos tres componentes se necesitan para implementar una base de datos y, por lo tanto, es necesario estar familiarizado con los ítems de datos y sus relaciones con el propósito de almacenar los datos requeridos en la base de datos.

El acceso a los requerimientos se usa para estructurar los datos en una forma que los haga más accesibles para todos los usuarios interesados y autorizados.

Los datos cuantitativos son necesarios para asegurar que los sistemas seleccionados tengan suficiente capacidad para almacenar todos los datos.

4.3.1 HERRAMIENTAS CASE.

El último componente de la metodología de diseño es el programa que soporta el proceso de diseño. La tendencia prevaleciente es soportar el análisis y el proceso de diseño por medio de herramientas automatizadas, las cuales se conocen por lo general como herramientas de ingeniería de software asistida por computadora.

Las herramientas CASE sirven para crear el soporte para la documentación y el diseño.

A su vez es más fácil introducir y usar modelos de diseño estructurado porque no es necesario dibujar y redibujar los modelos del sistema, es decir, los cambios se introducen en las herramientas CASE, las cuales a su vez producen modelos actualizados.

Por otro lado, la ayuda al diseño es más compleja y la mayoría de las herramientas CASE no pueden hacer sino más que convertir modelos de un nivel próximo y proporcionar un soporte sencillo de verificación para proporcionar la exactitud del modelo, lo cual ayuda a los diseñadores a eliminar redundancias o sugerir diseños físicos alternativos.

Las herramientas CASE pueden soportar varios modelos y definir varios modelos de realización. La mayoría de las metodologías y herramientas CASE son adecuadas para otros fines, aparte del modelado de datos, como para modelos que describen los procesos del sistema y el flujo de datos.

4.3.2 DISEÑO TÉCNICO DE LA BASE DE DATOS.

La siguiente gran etapa del diseño de una base de datos es el diseño técnico.

Al diseñador se le da la especificación de la base de datos y un modelo para llevarla a cabo. Este modelo define las estructuras de los datos disponibles. El diseñador debe entonces usar el modelo para construir la base de datos, al seleccionar:

- Los métodos para almacenar todos los datos requeridos.
- El programa que se usa para mantener y acceder los datos.

- La estructura de la base de datos que permitirá a los usuarios acceder
- Los datos en la forma requerida.
- Los métodos para distribuir los datos en varios lugares.

En el diseño técnico, la implementación física de la base de datos debe satisfacer los siguientes criterios, los cuales se identificaron en el primer paso (análisis de datos):

1. *Disponibilidad de los datos.* Los datos requeridos y sus relaciones se almacenan en la base de datos.
2. *Confiabilidad de los datos.* Los datos no se perderán ni se dañarán.
3. *Actualización de los datos.* El valor de los datos es el valor más reciente del ítem de datos.
4. *Consistencia en los datos.* Los mismos valores de datos se obtendrán para el mismo ítem, en las diferentes consultas y en igual tiempo.
5. *Flexibilidad en los datos.* Se pueden hacer extensiones con facilidad para satisfacer nuevos requerimientos.
6. *Eficiencia en los datos.* Los datos se almacenan y se recuperan a un costo mínimo.

El diseño técnico también especifica los programas que se usarán para acceder la base de datos. Estos programas pueden construirse con programas de bases de datos, lenguajes procedurales y herramientas de desarrollo para el usuario final.

4.4 MODELO RELACIONAL.

Como ya hemos visto el diseño técnico selecciona las mejores técnicas de implementación a fin de almacenar los datos definidos por el modelo del sistema.

El análisis de los datos se debe basar en un modelo formal, que satisfaga los siguientes objetivos:

1. Que pueda usarse para identificar los requisitos del usuario y presentarlos en una forma tal que los entiendan fácilmente el usuario y los profesionales de computación.
2. Que se pueda convertir con facilidad en una implementación física.
3. Que proporcione reglas y criterios para estructurar la base de datos.

El modelo relacional fue uno de los primeros en enfocar esos tres objetivos, es decir, las características esperadas de un buen diseño.

4.4.1 DISEÑO DE LA BASE DE DATOS RELACIONAL.

El modelo relacional proporciona una interfaz natural que simplifica el diseño y el uso de la base de datos. Los lenguajes pueden reducir el tiempo de desarrollo del programa y por tanto son atractivos en ambientes de procesamiento de datos comerciales. Sin embargo existen problemas en las implementaciones del modelo relacional, ya que se necesita un lenguaje poderoso, como el álgebra relacional, para alcanzar todo el potencial de un sistema de base de datos relacional. Las consultas en un lenguaje relacional deben poder usar cualquier atributo como una clave para tener acceso a la poderosa capacidad e indexar⁸.

Si se piensa convertir un modelo relacional o una estructura de registros lógicos a un diseño relacional es prácticamente trivial. Cada relación o registro, en la estructura de registro lógico, se transforma en una relación en el esquema de la base de datos. El paso de diseño siguiente es agregar los índices, para satisfacer los requerimientos de acceso que necesitan los usuarios para el manejo de información de la base de datos. Un diagrama de las trayectorias de acceso de registro lógico se puede usar para este propósito. Cada paso de acceso tiene los nombres de los ítems con los valores conocidos en el momento de la ejecución. Estos ítems se transforman en los ítems de clave de los índices, para que los registros se recuperen en forma directa.

Conversión de los subconjuntos

Las estructuras de subconjunto le presentan al diseñador otro problema: representar cada subconjunto con una relación separada o combinar algunos conjuntos y reducir el número de relaciones. Lo ideal, por supuesto, sería tener una relación aparte para cada subconjunto, así como una relación para cada conjunto de entidad. En este caso, no habría homogeneidad y todas las relaciones estarían en la forma normal más alta. Pero tal diseño tendría un gran número de relaciones y los gastos asociados con su creación y mantenimiento serían muy altos. La alternativa sería combinar los subconjuntos y sus orígenes en una sola relación. Las cuatro opciones para combinar los conjuntos que se pueden usar para este propósito son:

- Tener las relaciones separadas para cada conjunto y subconjunto de entidad.
- Adicionar los atributos propietarios para cada subconjunto y tener una relación por subconjunto; pero no las relaciones propietarias.
- Adicionar los atributos del subconjunto a la relación propietaria y no tener

⁸ Diseño de Bases de datos relacionales con acces y SQL server. Rebecca M. Riordan. McGrawHill. España 2000.

relaciones en el subconjunto, sino una relación propietaria.

- Combinar todos los atributos del subconjunto en la única relación y tener una relación para el propietario y otra para todos los subconjuntos.

El primer paso de diseño técnico, para convertir el modelo de la empresa en una estructura inicial del modelo de datos. El diseño inicial satisface dos criterios: uno es asegurar que todas las entidades, relaciones y atributos capturados en el modelo de la empresa, del análisis de requerimientos, también aparezcan en el modelo de datos, a fin de que todos los datos necesarios para los usuarios existan en la base de datos. El otro criterio es asegurar que no exista la repetición de datos.

Las personas encargadas de diseñar la bases de datos deben, satisfacer muchos otros criterios durante la implementación de la base de datos y estos criterios de diseño están dentro de las siguientes clases :

- Criterios de estructura, que se relacionan con la estructura de los datos en aspectos como la consistencia, redundancia, flexibilidad y completos.
- Criterios de desempeño, que se refieren al uso eficiente de los datos, en particular la localización, la confiabilidad de los datos, la actualidad y los aspectos de desempeño que se relacionan con el uso del almacenamiento, la localización de los datos y el tiempo de respuesta.
- Criterios del desarrollo de programas, que se relacionan con el acceso a los datos y la facilidad de desarrollo de programas de aplicación.

El diseño técnico debe convertir el modelo de datos en una implementación que satisfaga estos criterios. Los criterios de la estructura son cumplidos preservando la correspondencia con las formas normales de mayor grado cuando se define la base de datos. Con los métodos analizados en los capítulos anteriores, esto se logra al convertir cada relación en un tipo de registro separado. Por eso, el diseño inicial satisface los criterios de estructura.

Para satisfacer los criterios de desempeño, los diseñadores pueden ajustar el diseño inicial y así reducir el uso del almacenamiento y mejorar el desempeño. Y para satisfacer los criterios de desarrollo, se necesitan lenguajes que mejoren la expresividad y las técnicas de desarrollo de programas para reducir su tiempo de creación y, a su vez, hacer los programas más adaptables.

¿Qué criterios usar?

Aunque hay un acuerdo general sobre los criterios de diseño, los procedimientos de diseño, a menudo, enfatizan diferentes clases de criterios. Este énfasis depende de dos cosas. En primer lugar, la misma aplicación puede enfatizar

ciertos criterios. Por ejemplo, para los sistemas de transacción de alto volumen, el funcionamiento y el tiempo de respuesta quizá sean más importantes que el tiempo de desarrollo de programas o los criterios de estructura. Los sistemas de computación personal casi siempre se concentran en la reducción de los tiempos de desarrollo de programas.

La otra razón para cambiar el énfasis es la creciente especialización de los programas de base de datos. Varios sistemas de programación se han mejorado con componentes que satisfacen algunos criterios en forma automática. Por ejemplo, los optimizadores pueden seleccionar de manera automática las trayectorias para mejorar el funcionamiento, por lo que eliminan el desempeño como un criterio de diseño importante. También, pueden revisar el uso de los datos y calcular la mejor localización para éstos en los sistemas distribuidos. En forma similar, la disponibilidad de una gran cantidad de almacenamiento puede significar que el uso de este almacenamiento ya no es un criterio importante.

Los sistemas basados en registros son mejores para el cambio futuro del desempeño y, conforme los sistemas orientados a objetos y las ideas avancen, el énfasis en los criterios de desarrollo de programas crecerá. Por eso, se puede esperar un cambio gradual de los criterios de desempeño a los criterios de desarrollo de programas. Estos criterios incluyen la capacidad para usar de nuevo el programa de computación o construir un lenguaje semántico que aumente la comunicación con los usuarios.

Compromisos entre los criterios

Otro factor que complica el diseño es el de las relaciones entre los criterios. Estas relaciones son tales que un conjunto de criterios a menudo se cumple a expensas de otro. Por ejemplo, es factible introducir la redundancia para mejorar el funcionamiento, pero a costa de los criterios de estructura, porque hemos introducido redundancia en los datos.

4.4.2 ESTRUCTURA BASICA.

El modelo debe servir como medio de comunicación a los usuarios y el personal de computación, al darles una interfaz que entiendan con facilidad. La dependencia de esta interfaz de la computadora es importante ya que elimina las consideraciones previas restricciones impuestas por los dispositivos físicos produce especificaciones del usuario que enfatizan sus necesidades lógicas.

El modelo relacional utiliza tablas que proporcionan esta interfaz, donde el modelo de datos de la empresa se especifica en una serie de tablas o relaciones.

La presentación tabular de las relaciones satisface el primer objetivo del análisis de datos: lo entienden con facilidad los usuarios y los programadores y es un complemento que se puede usar para especificar los datos en una organización.

El segundo objetivo, la conversión de una implementación física, también se satisface con el modelo relacional. Una propuesta obvia es transferir en forma directa el modelo relacional a una máquina. Para hacerlo, el sistema manejador de la base de datos que soporta el modelo relacional debe estar disponible en un sistema computacional. Puede declararse en modo directo una serie particular de relaciones mediante el lenguaje de definición que proporciona el sistema.

El tercer objetivo es poder juzgar la calidad del diseño. El modelo relacional proporciona los siguientes criterios para las buenas estructuras de diseño de datos.

1. Cada evento debe almacenarse sólo una vez en la base de datos para lograr la consistencia de datos.
2. La base de datos debe ser consistente en estas operaciones.
3. La base de datos debe ser fácil de cambiar, ya que se debe rediseñar a medida de las necesidades que presenta la compañía.

TERMINOLOGÍA

La estrecha correspondencia del modelo relacional con las estructuras del usuario ha tenido como resultado dos terminologías.

Una emplea palabras cotidianas para referirse a las relaciones y a las etiquetas como tablas, que se componen de columnas y renglones.

La otra se basa en la estructura matemática de las relaciones, y de acuerdo con esta terminología, los renglones son n tuplos y las columnas son los atributos de la relación o tabla; además, en la base de datos las columnas y los atributos deben tener solamente un nombre.

Los diseñadores deben seleccionar los nombres del atributo de mayor relevancia por conveniencia del diagrama para fortalecer el modelo relacional porque aclara la representación relacional, lo cual hace una herramienta de comunicación más efectiva.

- **Atributos y Dominios.** Los valores de los atributos vienen de un dominio y el dominio para un atributo define los valores válidos o tipos de datos que pueden tomar los atributos de cada tuplo. Cada dominio se define como una serie de valores y cada atributo se asocia con uno de estos dominios.
- **Claves Relacionales.** Una clave es una serie de atributos, cuyo valor identifica un renglón único en una relación. Así, dos renglones o n tuplos no pueden tener el mismo valor de clave, lo que a veces se llama propiedad de unicidad.
- **Superclaves.** Una superclave puede incluir cualquier número de atributos que posean la propiedad de unicidad.

- Claves de Relación. Una clave de relación encadena dos de éstas para que las siguientes tres propiedades se mantengan en todo momento y para cualquier caso de la relación:
 1. Unicidad: El conjunto de atributos tiene un valor único para cada tuplo de la relación.
 2. No redundancia: Si un atributo se elimina de la serie, los demás no tendrán la propiedad de unicidad.
 3. Validez: No puede ser nulo ningún valor de atributo en la clave.

Cuando hay una clave importante, se conoce como clave primaria.

- Clave extranjeras. El término clave extranjera se usa en relación con la integridad de la base de datos. Una clave extranjera en la relación R1 es un conjunto de 1 o más atributos que están en una clave de relación en otra relación , R2, pero no una clave de relación R1.

4.4.3 PROPIEDADES DE RELACIONES.

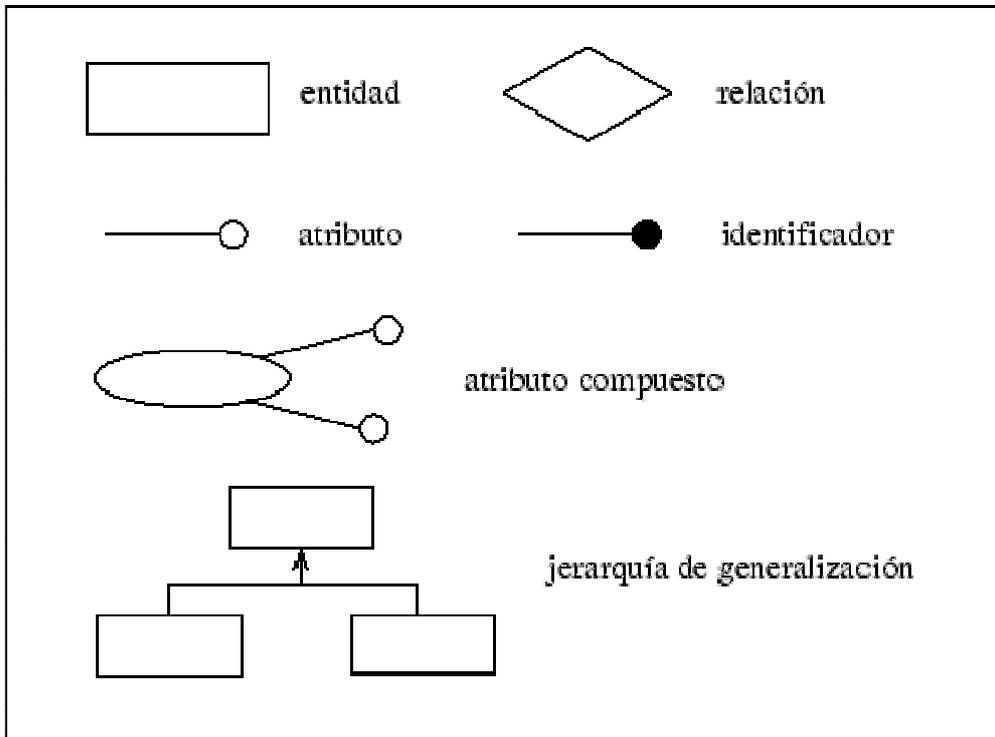
El análisis anterior sugiere que las propiedades de las relaciones hagan énfasis en la naturaleza lógica más que la naturaleza física del modelo relacional.

1. Hay una columna en la relación para cada atributo y cada columna tiene un número único para la relación.
2. Las entradas en la columna parten del mismo dominio.
3. El orden de las columnas o atributos en la relación no es significativo.
4. El orden de los renglones no es significativo.
5. No hay renglones duplicados.
6. Cada relación puede tener cualquier número de claves de relación, que a veces se usan para seleccionar renglones únicos en la relación.

4.5 EL MODELO ENTIDAD-RELACIÓN.

El modelo entidad-relación es el modelo conceptual más utilizado para el diseño conceptual de bases de datos. Fue introducido por Peter Chen en 1976. El modelo entidad-relación está formado por un conjunto de conceptos que permiten describir la realidad mediante un conjunto de representaciones gráficas y lingüísticas.

Originalmente, el modelo entidad-relación sólo incluía los conceptos de entidad, relación y atributo. Más tarde, se añadieron otros conceptos, como los atributos compuestos y las jerarquías de generalización, en lo que se ha denominado *modelo entidad-relación extendido*.



Entidad

Cualquier tipo de objeto o concepto sobre el que se recoge información: cosa, persona, concepto abstracto o suceso. Por ejemplo: coches, casas, empleados, clientes, empresas, oficios, diseños de productos, conciertos, excursiones, etc. Las entidades se representan gráficamente mediante rectángulos y su nombre aparece en el interior. Un nombre de entidad sólo puede aparecer una vez en el esquema conceptual.

Hay dos tipos de entidades: fuertes y débiles. Una *entidad débil* es una entidad cuya existencia depende de la existencia de otra entidad. Una *entidad fuerte* es una entidad que no es débil.

Relación (interrelación)

Es una correspondencia o asociación entre dos o más entidades. Cada relación tiene un nombre que describe su función. Las relaciones se representan gráficamente mediante rombos y su nombre aparece en el interior.

Las entidades que están involucradas en una determinada relación se denominan *entidades participantes*. El número de participantes en una relación es lo que se denomina *grado* de la relación. Por lo tanto, una relación en la que participan dos

entidades es una relación *binaria*; si son tres las entidades participantes, la relación es *ternaria*; etc.

Una *relación recursiva* es una relación donde la misma entidad participa más de una vez en la relación con distintos papeles. El nombre de estos papeles es importante para determinar la función de cada participación.

La *cardinalidad* con la que una entidad participa en una relación especifica el número mínimo y el número máximo de correspondencias en las que puede tomar parte cada ocurrencia de dicha entidad. La participación de una entidad en una relación es *obligatoria (total)* si la existencia de cada una de sus ocurrencias requiere la existencia de, al menos, una ocurrencia de la otra entidad participante. Si no, la participación es *opcional (parcial)*. Las reglas que definen la cardinalidad de las relaciones son las *reglas de negocio*.

A veces, surgen problemas cuando se está diseñado un esquema conceptual. Estos problemas, denominados *trampas*, suelen producirse a causa de una mala interpretación en el significado de alguna relación, por lo que es importante comprobar que el esquema conceptual carece de dichas trampas. En general, para encontrar las trampas, hay que asegurarse de que se entiende completamente el significado de cada relación. Si no se entienden las relaciones, se puede crear un esquema que no represente fielmente la realidad.

Una de las trampas que pueden encontrarse ocurre cuando el esquema representa una relación entre entidades, pero el camino entre algunas de sus ocurrencias es ambiguo. El modo de resolverla es reestructurando el esquema para representar la asociación entre las entidades correctamente.

Otra de las trampas sucede cuando un esquema sugiere la existencia de una relación entre entidades, pero el camino entre una y otra no existe para algunas de sus ocurrencias. En este caso, se produce una pérdida de información que se puede subsanar introduciendo la relación que sugería el esquema y que no estaba representada.

Atributo

Es una característica de interés o un hecho sobre una entidad o sobre una relación. Los atributos representan las propiedades básicas de las entidades y de las relaciones. Toda la información extensiva es portada por los atributos. Gráficamente, se representan mediante bolitas que cuelgan de las entidades o relaciones a las que pertenecen.

Cada atributo tiene un conjunto de valores asociados denominado *dominio*. El dominio define todos los valores posibles que puede tomar un atributo. Puede haber varios atributos definidos sobre un mismo dominio.

Los atributos pueden ser simples o compuestos. Un *atributo simple* es un atributo que tiene un solo componente, que no se puede dividir en partes más pequeñas que tengan un significado propio. Un *atributo compuesto* es un atributo con varios componentes, cada uno con un significado por sí mismo. Un grupo de atributos se representa mediante un atributo compuesto cuando tienen afinidad en cuanto a su significado, o en cuanto a su uso. Un atributo compuesto se representa gráficamente mediante un óvalo.

Los atributos también pueden clasificarse en monovalentes o polivalentes. Un *atributo monovalente* es aquel que tiene un solo valor para cada ocurrencia de la entidad o relación a la que pertenece. Un *atributo polivalente* es aquel que tiene varios valores para cada ocurrencia de la entidad o relación a la que pertenece. A estos atributos también se les denomina *multivaluados*, y pueden tener un número máximo y un número mínimo de valores. La *cardinalidad* de un atributo indica el número mínimo y el número máximo de valores que puede tomar para cada ocurrencia de la entidad o relación a la que pertenece. El valor por omisión es.

Por último, los atributos pueden ser derivados. Un *atributo derivado* es aquel que representa un valor que se puede obtener a partir del valor de uno o varios atributos, que no necesariamente deben pertenecer a la misma entidad o relación.

Identificador

Un identificador de una entidad es un atributo o conjunto de atributos que determina de modo único cada ocurrencia de esa entidad. Un identificador de una entidad debe cumplir dos condiciones:

1. No pueden existir dos ocurrencias de la entidad con el mismo valor del identificador.
2. Si se omite cualquier atributo del identificador, la condición anterior deja de cumplirse.

Toda entidad tiene al menos un identificador y puede tener varios identificadores alternativos. Las relaciones no tienen identificadores.

Jerarquía de generalización

Una entidad E es una generalización de un grupo de entidades E^1, E^2, \dots, E^n , si cada ocurrencia de cada una de esas entidades es también una ocurrencia de E . Todas las propiedades de la entidad genérica E son heredadas por las subentidades.

Cada jerarquía es total o parcial, y exclusiva o superpuesta. Una jerarquía es *total* si cada ocurrencia de la entidad genérica corresponde al menos con una ocurrencia de alguna subentidad. Es *parcial* si existe alguna ocurrencia de la entidad genérica que no corresponde con ninguna ocurrencia de ninguna subentidad. Una jerarquía es *exclusiva* si cada ocurrencia de la entidad genérica

corresponde, como mucho, con una ocurrencia de una sola de las subentidades. Es *superpuesta* si existe alguna ocurrencia de la entidad genérica que corresponde a ocurrencias de dos o más subentidades diferentes.

Un *subconjunto* es un caso particular de generalización con una sola entidad como subentidad. Un subconjunto siempre es una jerarquía parcial y exclusiva.

4.6 NORMALIZACIÓN DE UNA BASE DE DATOS.

La normalización es un conjunto de reglas que sirven para ayudar a los diseñadores a desarrollar un esquema que minimice los problemas de lógica en la base de datos, cada regla está basada en la que le antecede. La normalización se adoptó porque el viejo estilo de poner todos los datos en un solo lugar, como un archivo o una tabla de la base de datos, era ineficiente y conducía a errores de lógica cuando se trataba de manipular los datos⁹. Por ejemplo, una tabla de una tienda. Si almacena todos los datos en la tabla Clientes, ésta podría verse como se muestra a continuación:

Clientes
ID_Cliente
Nombre
Apellidos
Nombre_Producto1
Costo_Producto1
Imagen_Producto1
Nombre_Producto2
Costo_Producto2
Imagen_Producto2
Fecha_Pedido
Cantidad_Pedido
Nombre_Cia_Envios

La tabla se ha descrito de manera abreviada pero aun así representa la idea general.

¿Cómo podría añadir un nuevo cliente en su tabla Clientes? Debería añadir un producto y un pedido también. ¿Qué tal si quisiera emitir un informe de todos los productos que vende? No podría separar fácilmente los productos de los clientes con una simple instrucción SQL. Lo bello de las bases de datos relacionales, si están bien diseñadas, es que puede hacer esto fácilmente.

La normalización también hace las cosas fáciles de entender. Los seres humanos tenemos la tendencia de simplificar las cosas al máximo. Lo hacemos con casi todo desde los animales hasta con los automóviles. Vemos una imagen de gran

⁹ Diseño y Administración de bases de datos. Gary W, James V. Hansen. Prentice Hall. 2ª. Edición. España 1997.

tamaño y la hacemos menos compleja agrupando cosas similares juntas. Las guías que la normalización provee crean el marco de referencia para simplificar la estructura. En su base de datos de muestra es fácil detectar que usted tiene tres diferentes grupos: clientes, productos y pedidos. Si sigue las guías de la normalización, podría crear las tablas basándose en estos grupos.

El proceso de normalización tiene un nombre y una serie de reglas para cada fase. Esto puede parecer un poco confuso al principio, pero poco a poco irá entendiendo el proceso, así como las razones para hacerlo de esta manera. A la mayoría de la gente le encantan las hojas de cálculo por la forma en la que manejan sus datos. El tiempo que le lleve reconfigurar su esquema para ajustarlo al proceso de normalización, siempre será bien invertido. Al fin y al cabo, esto le tomará menos tiempo que el que tendría que invertir, para cortar y pegar sus columnas de datos para generar el informe que quiere su jefe.

Otra ventaja de la normalización de su base de datos es el consumo de espacio. Una base de datos normalizada puede ocupar menos espacio en disco que una no normalizada. Hay menos repetición de datos, lo que tiene como consecuencia un mucho menor uso de espacio en disco.

4.6.1 GRADOS DE NORMALIZACIÓN.

Existen básicamente tres niveles de normalización: Primera Forma Normal (1NF), Segunda Forma Normal (2NF) y Tercera Forma Normal (3NF). Cada una de estas formas tiene sus propias reglas. Cuando una base de datos se conforma a un nivel, se considera normalizada a esa forma de normalización. Por ejemplo, supongamos que su base de datos cumple con todas las reglas del segundo nivel de normalización. Se considera que está en la Segunda Forma Normal. No siempre es una buena idea tener una base de datos conformada en el nivel más alto de normalización. Puede llevar aun nivel de complejidad que pudiera ser evitado si estuviera en un nivel más bajo de normalización.

4.6.1.1 PRIMERA FORMA NORMAL.

La regla de la Primera Forma Normal establece que las columnas repetidas deben eliminarse y colocarse en tablas separadas. Ésta es una regla muy fácil de seguir. Observe el esquema de la tabla Clientes de la base de datos.

Clientes

ID_Cliente
Nombre
Apellidos
Nombre_Producto1
Costo_Producto1
Imagen_Producto1
Nombre_Producto2

Costo_Producto2
Imagen_Producto2
Fecha_Pedido
Cantidad_Pedido
Nombre Cia Envios

La tabla tiene varias columnas repetidas. Éstas se refieren principalmente a los productos. De acuerdo con la regla, debe eliminar las columnas repetidas y crearles su propia tabla.

Eliminación de datos repetidos en una base de datos

Clientes

ID_Clientes
Nombre
Apellidos
Direccion
Numero_Pedido
Fecha_Pedido
Cantidad_Pedido
Clave_Cia_Envios
Nombre_Ci_

Pedidos

Nombre_Productos
Costo_Producto
Imagen_Producto
Envíos

Ahora tiene dos tablas. Pero todavía hay un problema. No hay forma de relacionar los datos de la tabla original con los de la nueva tabla. Para hacerlo, debe añadir un campo clave a la segunda tabla de forma que se establezca la relación. Añada a la tabla Productos una clave primaria que se llame ID_Producto y añada una clave a la tabla Clientes que la relacione con la tabla Productos. El campo ID_Producto es el candidato ideal.

Primera Forma Normal

Clientes

ID_Productos
ID_Clientes
Nombre
Apellidos
Direccion
Numero_Pedido

Fecha_Pedido
Cantidad_Pedido
Clave_Cia_Envios

Pedidos

ID_Productos
Nombre_Productos
Costo_Producto
Imagen_Producto

Así, se ha establecido una relación uno a varios. Ésta representa lo que la base de datos estará haciendo en la vida real. El cliente tendrá muchos productos que podrá comprar, sin importar cuántos otros clientes quieran comprarlos también. Además, el cliente necesitará haber pedido un producto para ser un cliente. Usted ya no está obligado a añadir un cliente cada vez que añade un nuevo producto a su inventario.

Poner la base de datos en la Primera Forma Normal resuelve el problema de los encabezados de columna múltiples. Muy a menudo, los diseñadores de bases de datos inexpertos harán algo similar a la tabla no normalizada. Una y otra vez, crearán columnas que representen los mismos datos. En una empresa de servicios de electricidad, había una base de datos para el control de refacciones de una planta nuclear. La tabla de su base de datos, la cual contenía los números de parte de las refacciones, tenía una columna repetida más de treinta veces. Cada vez que una nueva parte se tenía que dar de alta, se creaba una nueva columna para almacenar la información. Obviamente, el diseño de la base de datos era bastante pobre y, por lo mismo, resultaba una pesadilla para sus programadores/administradores.

La normalización ayuda a clarificar la base de datos ya organizarla en partes más pequeñas y más fáciles de entender. En lugar de tener que entender una tabla gigantesca y monolítica que tiene muchos diferentes aspectos, usted sólo tiene que entender objetos pequeños y más tangibles, así como las relaciones que guardan con otros objetos también pequeños. No es necesario mencionar que un mejor entendimiento del funcionamiento de su base de datos conducirá aun mejor aprovechamiento de sus activos.

4.6.1.2 SEGUNDA FORMA NORMAL.

La regla de la Segunda Forma Normal establece que todas las dependencias parciales se deben eliminar y separar dentro de sus propias tablas.

Una dependencia parcial es un término que describe a aquellos datos que no dependen de la clave de la tabla para identificarlos. En la base de datos de muestra, la información de pedidos está en cada uno de los registros. Sería mucho

más simple utilizar únicamente el número del pedido. El resto de la información podría residir en su propia tabla. Una vez que haya organizado la información de pedidos.

Eliminación de las dependencias parciales -Segunda Forma Normal

Clientes

ID_Productos
ID_Clientes
Nombre
Apellidos
Direccion
Numero_Pedido
Nombre_Cia_Envios

Pedidos

ID_Productos
Nombre_Productos
Cantidad_Pedido Costos_Productos

Productos

ID_Producto
Fecha_Compra
Imagen_Producto

De nuevo, al organizar el esquema de esta forma puede reflejar el mundo real en su base de datos. Tendría que hacer algunos cambios en sus reglas del negocio para que esto fuera aplicable, pero para ilustrar la normalización, así está bien.

Una de las mayores desventajas de la normalización es el tiempo que lleva hacerlo. La mayoría de la gente está demasiado ocupada, y emplear tiempo para asegurarse de que sus datos están normalizados cuando todo funciona más o menos bien, parece ser un desperdicio de tiempo. Pero no es así. Usted tendrá que emplear más tiempo arreglando una base de datos no normalizada que el que emplearía en una normalizada.

Al haber alcanzado la Segunda Forma Normal, usted puede disfrutar de algunas de las ventajas de las bases de datos relacionales. Por ejemplo, puede añadir nuevas columnas a la tabla Clientes sin afectar a las tablas Productos y Pedidos. Lo mismo aplica para las otras tablas. Alcanzar este nivel de normalización permite que los datos se acomoden de una manera natural dentro de los límites esperados.

Una vez que ha alcanzado el nivel de la Segunda Forma Normal, se han controlado la mayoría de los problemas de lógica. Puede insertar un registro sin un

exceso de datos en la mayoría de las tablas. Observando un poco más de cerca la tabla Clientes, vemos la columna Nombre_Cia_Envios. Ésta no es dependiente del cliente. El siguiente nivel de normalización explicará cómo solucionar esto.

4.6.1.3 TERCERA FORMA NORMAL.

La regla de la Tercera Forma Normal señala que hay que eliminar y separar cualquier dato que no sea clave. El valor de esta columna debe depender de la clave. Todos los valores deben identificarse únicamente por la clave. En la base de datos de muestra, la tabla Clientes contiene la columna Nombre_Cia_Envios, la cual no se identifica únicamente por la clave. Podría separar estos datos de la tabla y ponerlos en una tabla aparte.

Eliminación de los datos que no son claves para la Tercera Forma Normal

Clientes

ID_cliente
ID_Producto
Numero_Pedido
ID_Cia_Envios
Nombre
Apellidos
Direccion

Producto

ID_Producto
Nombre_Producto
Costos_Producto

PedidoMaestro

ID_Pedido
Fecha_Pedido
Cantidad_Pedidos
Foto_Producto

PedidoDetallado

ID_PedidoDetallado
ID_Pedido

Cias_Envios

ID_Cia_Envios
Nombre_Cia_Envios
Fecha_Pedido
Cantidad_Pedido

Ahora todas las tablas están en la Tercera Forma Normal. Esto le da más flexibilidad y previene errores de lógica cuando inserta o borra registros. Cada columna en la tabla está identificada de manera única por la clave, y no hay datos repetidos. Esto provee un esquema limpio y elegante, que es fácil de trabajar y expandir.

Qué tan lejos debe llevar la normalización

La normalización es una ciencia subjetiva. Determinar las necesidades de simplificación depende de usted. Si su base de datos va a proveer información aun solo usuario para un propósito simple y existen pocas posibilidades de expansión, normalizar sus datos hasta la 3FN sea quizá algo extremo. Las reglas de normalización existen como guías para crear tablas que sean fáciles de manejar, así como flexibles y eficientes.

Existen seis niveles más de normalización que no se han analizado aquí. Ellos son Forma Normal Boyce-Codd, Cuarta Forma Normal (4NF), Quinta Forma Normal (5NF) o Forma Normal de Proyección-Unión, Forma Normal de Proyección-Unión Fuerte, Forma Normal de Proyección-Unión Extra Fuerte y Forma Normal de Clave de Dominio. Estas formas de normalización pueden llevar las cosas más allá de lo que necesita. Éstas existen para hacer una base de datos realmente relacional. Tienen que ver principalmente con dependencias múltiples y claves relacionales.

En resumen

La normalización es una técnica que se utiliza para crear relaciones lógicas apropiadas entre tablas de una base de datos. Ayuda a prevenir errores lógicos en la manipulación de datos. La normalización facilita también agregar nuevas columnas sin romper el esquema actual ni las relaciones.

Existen varios niveles de normalización: Primera Forma Normal, Segunda Forma Normal, Tercera Forma Normal, Forma Normal Boyce-Codd, Cuarta Forma Normal, Quinta Forma Normal o Forma Normal de Proyección-Unión, Forma Normal de Proyección-Unión Fuerte, Forma Normal de Proyección-Unión Extra Fuerte y Forma Normal de Clave de Dominio. Cada nuevo nivel o forma lo acerca más a hacer su base de datos verdaderamente relacional. Las tres primeras formas de normalización proveen suficiente nivel de normalización para cumplir con las necesidades de la mayoría de las bases de datos.

Normalizar demasiado puede conducir a tener una base de datos ineficiente y hacer a su esquema demasiado complejo para trabajar.

CAPITULO 5

Sistemas de Autoaprendizaje Electrónico

5. SISTEMAS DE AUTOAPRENDIZAJE ELECTRÓNICO.

5.1 PROGRAMACIÓN DE LA ENSEÑANZA INDIVIDUALIZADA.

Se ha formulado una programación enfocada al estudiante tomado individualmente y como respuesta a las prácticas de la escuela tradicional, este esquema nos permite planear y llevar a cabo la instrucción individual, ya que pretende capacitar a los maestros para poner en práctica sus conocimientos sobre el aprendizaje, el desarrollo humano y el contenido de las asignaturas. El modelo o esquema mencionado es la base de una nueva forma de enseñanza llamada *Educación Individualmente Orientada* (EIO), que se inició en los años 60 y tuvo por promotor a Klausmeier y sus ayudantes pertenecientes al "Centro de Investigación y Desarrollo del Aprendizaje Conceptual", en Wisconsin. Esta nueva forma de enseñanza está reemplazando rápidamente los sistemas tradicionales. Un representante de la Asociación de Relaciones Públicas de las Escuelas Nacionales, después de haber entrevistado a un buen número de maestros y rectores de la EIO la describió del modo siguiente:

Parte del planteamiento de la EIO se propone alcanzar claros y concretos objetivos por medio de la enseñanza individualizada: no se trata de la enseñanza individual que solo consiste en el aprendizaje mediante elementos materiales con muy poca o ninguna participación del maestro, sino de una instrucción individualizada que se realiza con elementos propios y en la que los procedimientos no son más que una parte del programa de enseñanza de cada estudiante, por consecuencia el alumno al interactuar con una aplicación didáctica realizada por un profesor o persona relacionada con algún tema adquirir los conocimientos mediante el material que maneja en la aplicación.

La enseñanza individualizada ha significado el impulso más notorio para mejorar la educación norteamericana y en todo el mundo por medio de las tecnologías que se encuentran al alcance de la gran mayoría de los alumnos. Todos los profesores reconocen que una clase de 25 alumnos, aunque todos tengan la misma edad, puede presentar diferencias muy importantes en cuanto a capacidades de los alumnos, mayor interés en una materia que en otra o particular preferencia por una determinada metodología¹⁰.

La mayoría de los maestros reconoce que cada alumno es una persona independiente con su propia individualidad. Pero a pesar de considerarlo así, se ha obligado a los alumnos, sin excepción, a sujetarse a un sistema establecido sin tener en cuenta sus condiciones personales. La EIO trata de resolver el problema estimulando programas educativos individuales que den a cada estudiante la posibilidad de lograr sus objetivos, lo cual exige que:

¹⁰ Enciclopedia Técnica de la Educación. Emiliano Martínez Rodríguez, Sergio Sánchez Cerezo. Santillana. Madrid España 1998.

La planeación de los programas educativos permita a cada estudiante avanzar a un ritmo personal.

El suministro de textos, ayudas audiovisuales, experiencias y otros materiales y actividades se adapten al modo individual de aprendizaje.

Muy a menudo la enseñanza se fracciona cuando se concibe en forma individual.

El modelo programático de enseñanza de la EIO ayuda a los maestros a disponer del mejor modo posible sus actividades docentes y las condiciones que requiere cada estudiante en particular. Dicho modelo provee el nivel de capacidad inicial del alumno, su tasa de aprovechamiento, su modo particular de aprendizaje, su escala de motivación y otras características relacionadas con el programa educativo de cada escuela.

El modelo de enseñanza programada para un estudiante en particular ofrece semejanzas con otros modelos ya propuestos por Glaser, Bloom, y Carroll, quienes destacan también la importancia de determinar los objetivos didácticos, de evaluar la preparación del estudiante para alcanzar el objetivo, de medir su progreso y sus resultados finales y por último de acomodar las circunstancias a cada estudiante según su capacidad para alcanzar los objetivos. El presente modelo de instrucción programada para un solo estudiante fue aplicado por primera vez a nivel nacional en escuelas elementales y secundarias, en los primeros años de la década del 70. Una evaluación cuidadosa de este primer intento demostró gran éxito del personal escolar en la adecuación del modelo a una o dos áreas del plan de estudios, durante el primer año.

El modelo de enseñanza programada y otros similares, como se dijo anteriormente, se están utilizando actualmente en colegios secundarios y universidades en su empeño de mejorar la calidad de la educación. De igual importancia resulta el hecho de que muchos editores están publicando materiales didácticos que incluyen la postulación de objetivos, provistos de medios de evaluación de los resultados logrados por los alumnos, y que sugieren actividades que permiten tanto a los profesores como a los estudiantes alcanzar tales objetivos.

A continuación se darán los pasos del modelo de instrucción programada para un solo estudiante. Indique si cada paso es en principio una responsabilidad del profesor o de otras personas.

- a. *Formular los objetivos educativos para los estudiantes.*
- b. *Identificar los objetivos apropiados para un subgrupo de estudiantes.*
- c. *Evaluar la preparación de cada estudiante para alcanzar cada objetivo.*
- d. *Establecer los objetivos de la enseñanza que debe alcanzar cada estudiante.*
- e. *Elaborar un programa de enseñanza adecuado a cada alumno.*
- f. *Evaluar el logro de los objetivos por parte cada estudiante.*

5.1.1 EL PROGRESO DE LA ENSEÑANZA.

En la actualidad los resultados de la investigación en el campo del aprendizaje y las habilidades humanas, en la enseñanza y en otros aspectos igualmente importantes que se han ido presentando en tal forma que tanto los profesores como los estudiantes pueden servirse de ellos más fácilmente, de esto que los materiales didácticos en cualquier forma pero ahora principalmente en medios electrónicos constituyen una ayuda para los profesores en la aplicación del modelo de enseñanza programada para un estudiante individual. Dado que este tipo de investigación y desarrollo es nuevo en materia de educación, podemos considerar que el mejoramiento en la calidad de la educación responde a algunos de los propósitos y contribuciones de la investigación y del desarrollo científico.

La Investigación. La investigación en psicología educativa implica el estudio sistemático del aprendizaje, la enseñanza y el desarrollo humanos. Parte de la investigación se realiza directamente con miras al descubrimiento de los hechos y principios del aprendizaje y el desarrollo humanos, mientras que otra se realiza indirectamente con la intención específica de comprender y mejorar la instrucción.

Desarrollo de productos. Schutz explicó el desarrollo de productos de la siguiente manera. Primero, se establecen las especificaciones del producto. El siguiente paso consiste en elaborar y ensayar en el laboratorio métodos y materiales acordes con estas especificaciones. Una vez que este primer intento se pueda considerar exitoso, el siguiente paso será preparar los productos, materiales y procedimientos para empleados y someterlos a prueba de manera concreta en el aula de clase. Se trata aquí de lograr un producto que resulte atractivo para los alumnos, fácil de manejar para los profesores, que su empleo tome muy corto tiempo y finalmente que sea lo más económico posible. Cada uno de los pasos precedentes implica un ciclo repetido: desarrollo, prueba y luego perfeccionamiento para eliminar las deficiencias del producto y aumentar su efectividad. En la etapa formativa o de desarrollo es esencial dicha evaluación continua del producto, con base en los objetivos fijado.

Klausmeier destacó la necesidad de continuar a gran escala el perfeccionamiento de los materiales, técnicas, procesos, equipo y dispositivos de organización en materia de enseñanza. Ya que gracias a la colaboración entre la investigación y el desarrollo ha sido posible construir puentes sobre grandes ríos, levantar rascacielos, transmitir sonido y fotografías a color vía satélite y sostener al hombre en el espacio.

5.1.2 MEDICIÓN, EVALUACIÓN, ESTADÍSTICA Y PLANEACIÓN INVESTIGATIVA.

La medición es un proceso mediante el cual se define la extensión, las dimensiones o la cantidad de alguna cosa¹¹. La extensión de tiempo para completar una tarea, la distancia que alcanza un objeto al ser lanzado a una cierta altura por mencionar algunas, pero la mayoría de las medidas en educación encierran demostración del rendimiento de los alumnos por ejemplo :

Se desea determinar qué tanto ha aprendido a deletrear un niño, se desarrolla una prueba de tipo oral o escrita u observamos muestras de la escritura del niño y luego contamos el número de deletreos correctos e incorrectos. De esta forma se determina el porcentaje de aprendizaje dando un valor a cada una de los aciertos o errores.

La evaluación es el proceso mediante el cual se juzga el valor de algo siguiendo un criterio o una medida preestablecida. Se pueden hacer juicios de valoración sobre las características y el rendimiento de los estudiantes, sobre materiales de enseñanza, métodos, etc. Por ejemplo, podemos evaluar como satisfactorio el aprovechamiento de un estudiante en historia si obtiene un puntaje máximo previsto, en una prueba. Podemos incluso evaluar la conducta de un estudiante como satisfactoria aunque presente interrupciones ocasionales en el trabajo, a condición de haber permitido expresamente algunas fallas menores de su parte. Como ya dijimos, es esencial una medición adecuada antes de proceder a la evaluación. Además, se requieren ciertos criterios con los cuales puedan relacionarse las mediciones.

5.2 PRINCIPIOS Y CONDICIONES DEL APRENDIZAJE.

¿Qué condiciones internas de los alumnos, tales como el grado de motivación y las experiencias pasadas, resultan esenciales en la obtención de información de hechos, conceptos y otros resultados del aprendizaje? ¿Qué condiciones externas, tales como la calidad del material de instrucción y la cantidad de práctica, facilitan el aprendizaje? Con el fin de encontrar una respuesta a estas preguntas y a otras similares, se han llevado a cabo miles de experimentos en laboratorios y escuelas y de ellos se han derivado innumerables teorías y modelos. A continuación expondremos información sobre cinco teorías que se consideran las más representativas.

¹¹ Psicología Cognitiva. Estrategias en la Práctica Docente. Cynthia Klingler, Guadalupe Badillo. México. 2000

5.2.1 EL CONDICIONAMIENTO CLÁSICO Y OPERANTE.

Los principios del condicionamiento fueron establecidos inicialmente mediante estudios de laboratorio. Staats hace una presentación clara sobre dichos principios y sus aplicaciones al aprendizaje escolar. Muchas de sus ideas se basan en los trabajos originales de Parlov y Skinner

5.2.1.1 EL CONDICIONAMIENTO CLÁSICO.

En el condicionamiento de primer orden un estímulo neutral, en el estímulo neutral se trata de asociar una respuesta no condicionada como por ejemplo ver comida que nos lleva a una respuesta no condicionada de salivación (o sea que no es necesario condicionar el estímulo alimento para provocar la respuesta incondicionada de la salivación), pero si a esto le agregamos un sonido o imagen adicional, y se presentan repetidamente los dos estímulos a la vez, retirando el alimento gradualmente y entonces el sólo sonido o imagen provoca finalmente la salivación. Así el sonido o imagen se convierte en el estímulo condicionado y la salivación en la respuesta condicionada.

En el condicionamiento de orden superior, que constituye el paso siguiente, el sonido o imagen que produce la respuesta de salivación, desempeña ahora la misma función del alimento, es decir, de estímulo no condicionado. Cuando se presenta un nuevo estímulo neutral, esta vez un círculo rojo, inmediatamente antes del sonido o imagen, la vista de dicho círculo sólo, después de aparecer repetidas veces simultáneamente con el sonido o imagen, provoca finalmente la salivación.

Algunos psicólogos explican todo aprendizaje primordialmente como asociación entre dos estímulos o estímulos y respuestas a través del condicionamiento de primer orden y de orden superior. Así por ejemplo, los niños adquieren sus primeras palabras mediante la asociación de nombres y cosas, luego construyen frases y oraciones por encadenamiento o asociando consecutivamente las palabras que componen las oraciones.

Los principios del condicionamiento clásico ya mencionados pueden expresarse de manera más formal. Cada vez que un estímulo que produce una respuesta condicionada aparece a intervalos cortos de tiempo con un nuevo estímulo neutral, que no produce dicha respuesta, existe la tendencia del nuevo estímulo a producir la reacción. Y es más, cuando el nuevo estímulo, ahora condicionado, produce la respuesta condicionada, puede manejarse con otro estímulo neutral, de lo cual resulta que la respuesta se da bajo el control del segundo estímulo, previamente neutral.

5.2.1.2 CONDICIONAMIENTO OPERANTE.

En el condicionamiento clásico, un estímulo originalmente neutral provoca una respuesta mediante la aparición simultánea y repetida con un estímulo incondicionado. No ocurre lo mismo en el condicionamiento operante o instrumental, en el cual la respuesta debe aparecer antes de que se dé un reforzador positivo, un premio por ejemplo, o antes de retirar un reforzador negativo, como es el caso de un estímulo aversivo. Por lo tanto, en el condicionamiento operante, cuando a la respuesta sigue un premio ésta tiende a repetirse. Por ejemplo, un niño que tiene hambre y dice "por favor", se le da alimento. Cada vez que se repita esta secuencia, habrá mayor posibilidad de que el niño cuando sienta hambre, diga "por favor" para obtener el alimento. El alimento sirve entonces como un reforzador positivo de la respuesta u operante "por favor".

A menudo los niños experimentan estímulos aversivos asociados con sentimientos de dolor. Por ejemplo, un niño toca un aparato de calefacción. Al quitar el aparato (estímulo operante) desaparece el estímulo aversivo (el calor) y esto re fuerza al operante. Generalmente no aplicamos de modo deliberado estímulos de dolor, como el calor, a los niños, pero sí a los animales de laboratorio. Otro ejemplo: un profesor aísla a un alumno de los demás (estímulo aversivo) hasta que el chico copie una página de un libro; cuando la haya copiado, el profesor suspende el aislamiento. La supresión de la condición de aislamiento fortalece la conducta de copiar. No cabe duda de que la supresión de un estímulo aversivo, fortalece la respuesta que lleva a dicha supresión. Esto difiere, del hecho de castigar una conducta indeseada cuando al niño no se le da oportunidad de hacer algo que pudiera suspender el castigo. Resulta imposible predecir los efectos de un castigo que siga un comportamiento no relacionado con la suspensión del mismo. Sin embargo pueden ser tres los efectos, a saber: cuando el niño conoce las razones del castigo que le aplica un adulto por determinada acción, tiende a no repetirla. Por lo tanto, el castigo aplicado a un acto cada vez que este se comete, puede llevar a su extinción. Por otra parte, el castigo puede también suprimir el acto, solamente en presencia de la persona que castiga, de tal manera que en su ausencia, el niño repite el acto, especialmente si no ha desarrollando aún otra actividad que le reemplace la que ocasiona el castigo. El tercer efecto posible es que la persona que castiga se convierta en un estímulo aversivo para el niño, que ha llegado a asociarla con el castigo.

Se puede resumir las principales generalizaciones sobre el condicionamiento operante. Cuando un reforzador positivo sigue de cerca determinada respuesta, se aumenta la posibilidad de que dicha respuesta aparezca de nuevo. Cuando se suprime un reforzador negativo-inmediatamente después de determinada respuesta que lleva a dicha supresión, ocurre lo mismo que en el caso anterior. O sea que el condicionamiento de orden superior se da también en el condicionamiento operante. Cuando aparece un estímulo neutral simultáneamente con un estímulo positivo reforzador, el primero adquiere poder reforzador por sí mismo, después de repetidas apariciones simultáneas. Los reforzadores

secundarios generalmente son de tipo social. Entre ellos se cuentan los cumplidos dirigidos a los demás, las respuestas o gestos cariñosos; las destinaciones tales como el dinero, los honores, las buenas calificaciones; y las manifestaciones aprobatorias de grupo, como el aplauso, la risa o la atención.

Olvido y condicionamiento

Los psicólogos que explican el aprendizaje en términos de condicionamiento generalmente no utilizan el termino "olvido" y prefieren referirse a él con la palabra *extinción* que encierra la idea de debilitamiento de las respuestas, ya que como se ha observado, las respuestas o reacciones que se vigorizan con el refuerzo pueden debilitarse con el esfuerzo. Por ejemplo, los niños que después de un refuerzo tiran una palanca como respuesta a un estímulo de color rojo, dejarán finalmente de hacerlo si el refuerzo no se hace sentir después de mover la palanca. Este hecho produce una de las dificultades primarias del condicionamiento operante, pues más importante que el refuerzo de una respuesta es la escala variable según la cual se refuerza cada nueva respuesta en particular lo que resulta más eficaz que una razón fija después de cada respuesta. Diversos profesionales encaran a diario el problema de saber cuándo y cómo puede suspenderse el refuerzo sin producir extinción. Se supone que la frecuencia del refuerzo puede mermarse a medida que se vigoriza la respuesta, advirtiendo que la habilidad para lograrla bien a menudo sirve como auto refuerzo, ya que en este caso no se necesita el refuerzo externo. De todos modos, sigue pendiente el problema de decidir cuándo debe disminuirse o suspenderse el refuerzo.

Aplicaciones en el aprendizaje escolar

Antes de ocuparnos directamente de cualquier aplicación queremos hacer notar que muchos psicólogos y educadores creen que la mayor parte de la práctica educativa debe fundamentarse en los principios de la modificación de la conducta. Las prácticas corrientes de codificación de la conducta en educación provienen en gran medida de los principios de condicionamiento tal como los describe Thoreson. Otros psicólogos y educadores creen que la conformación de la conducta del estudiante por el sistema de premios y castigos sin conocimiento de parte del alumno no debería aplicarse.

El uso deliberado del refuerzo positivo, particularmente en forma de recompensas tangibles, origina a menudo objeciones éticas y se relaciona con efectos perjudiciales que pueden derivarse de tal práctica. La postura más comúnmente adoptada consiste en afirmar que la conducta deseable debería ser intrínsecamente satisfactoria. Se teme que si las personas son recompensadas con frecuencia, dejarán de comportarse convenientemente, a menos que se les premie cada vez y dejen de responder cuando el premio se suspenda definitivamente. Además se pretende que la práctica de la recompensa no sólo establece una conducta débil y fácil, sino que el refuerzo ocasional probablemente

interfiere el desarrollo de la espontaneidad, de la creatividad, el movimiento de los sistemas intrínsecos de motivación y otras características de la personalidad. Algunos de los más fogosos críticos consideran que el uso deliberado del refuerzo es falso, truculento y significa un insulto a la integridad personal de los seres humanos.

Los otros fenómenos que han sido estudiados por los teóricos del condicionamiento y por otros psicólogos incluyen la discriminación y la generalización. Estos fenómenos no son de ninguna manera exclusivos de las teorías del condicionamiento. Discriminar y generalizar, como operaciones mentales, son también creaciones de importancia en las teorías cognoscitivas que buscan la explicación del aprendizaje de la información de hechos, conceptos y principios.

Aprendizaje Por Observación

Las diversas etapas de la conducta se aprenden inicialmente mediante la observación e imitación de un modelo o comportamiento, se considera que dichos aspectos no surgen gradualmente mediante el refuerzo diferencial y el encadenamiento de unidades individuales de estímulo-respuesta (E+R) como ocurría en las teorías del condicionamiento, sin embargo, la contigüidad interviene en el aprendizaje de dichas etapas, que ya adquiridas se fortalecen mediante el refuerzo.

El aprendizaje inicial de algunas actitudes y conductas de tipo social tales como la agresión, la dependencia, el autocontrol y las relaciones sexuales, así como la eliminación de los miedos, las fobias y otras formas del comportamiento que interfieren el ajuste personal o social del individuo así como en el comportamiento del mismo ya que también se aprende por medio de la observación de dichos acontecimientos.

Modelos de imitación

Los modelos que los niños observan e imitan pueden clasificarse en simbólicos, de la vida real y representativos. En el hogar, los padres y los parientes constituyen los modelos de la vida real para los niños más pequeños. Los profesores y otras personas de la comunidad son también modelos de la vida real para muchos escolares. Aquellos modelos que se presentan a los niños en forma de instrucción oral o escrita, de dibujos o ambos simultáneamente, son simbólicos. Los modelos que se sirven de los medios audiovisuales, particularmente la televisión, son de carácter representativo.

A menudo se les concede mucha atención en los hogares y en las escuelas a los modelos ejemplares. Tales modelos demuestran conductas de adaptación social, que las personas que tienen a su cargo la educación de los niños, consideran como más ideales. Este aspecto socializador de la educación es

probablemente más difícil de lo que comúnmente se cree. Buena parte de la conducta socializada, así como de la conducta antisocial y desviada, se adquiere mediante la imitación de modelos.

Los efectos de la imitación

Bandura y Walters afirman que la adquisición inicial de las etapas de la conducta, o las reacciones son primordialmente el producto de la observación de sucesos próximos en el tiempo y luego de la imitación mental o abierta de las respuestas.

Por otra parte, el subsecuente fortalecimiento de conductas ya adquiridas, mediante la observación y la respuesta, depende del refuerzo, como ya lo hemos dicho. Por lo tanto, la contribución más importante de la teoría del aprendizaje por observación, consiste en su explicación de cómo se aprende *inicialmente* el comportamiento. Los procesos resultan más claros si consideramos tres efectos diferentes de la observación y la imitación.

1. Efecto modelador. Un observador examina e imita un modelo, el observador reacciona la primera vez de manera abierta o introvertida, adaptando a su propia persona las conductas del modelo, para que esto ocurra el modelo debe ofrecer un comportamiento nuevo para el observador, que este sea capaz de imitar. Por ejemplo, a unos niños de preescolar se les mostró una película de un modelo que realizaba diversos actos agresivos que los niños no habían visto anteriormente, los niños imitaron el comportamiento del modelo.

2. Efecto desinhibidor. La observación y la imitación de un modelo pueden igualmente debilitar o fortalecer reacciones de inhibición, ya adquiridas por el observador. El comportamiento agresivo en los niños generalmente es desaprobado por los adultos y hasta cierto punto, inhibido por parte del niño. Sin embargo, la observación de un modelo como los padres, el profesor o incluso un personaje de la televisión, que presenta un comportamiento agresivo, debilita la inhibición; y como resultado, el niño comienza a manifestar la misma forma de comportamiento agresivo del modelo, que previamente estaba inhibido en él. Además, otros comportamientos agresivos que estaban igualmente inhibidos pueden salir a flote. Por lo tanto esto es lo que denominamos efecto desinhibidor la liberación de toda una serie de comportamientos que generalmente están inhibidos. Se han demostrado, por otra parte, que algunas otras formas de conducta desviada se desinhiben mediante la observación de un modelo que ejecuta dicha conducta libremente.

De igual manera, la observación y la imitación provocan el fortalecimiento de respuestas inhibitorias. Es más probable que esto ocurra bajo las siguientes circunstancias: el modelo puede someterse a sufrir consecuencias dolorosas, como resultado de determinada conducta. Entonces el observador inhibe dicha conducta que percibe asociada con el dolor, ya que probablemente teme el castigo si manifiesta dicho comportamiento. De esta manera se explica la tendencia de los niños a temer las mismas cosas que temen sus padres.

3. Efecto activador. Mediante la observación e imitación de un modelo, el observador puede acomodar su comportamiento con las respuestas que ya forman parte de su repertorio. Es decir que dichas respuestas son activadas. Este efecto puede demostrarse experimentalmente, aunque uno pueda pensar que sólo ocurre si se conoce la historia del observador. Los servicios voluntarios o las contribuciones de dinero, dedicarse de lleno a una actividad determinada y comer alimentos indiscriminadamente, son ejemplos de comportamientos que pueden activarse a través de la observación de modelos apropiados. La nota que sigue resulta interesante en lo que concierne a las relaciones entre estos tres efectos:

Los efectos modeladores sólo son posibles si el modelo exhibe respuestas que el observador no ha aprendido anteriormente, mientras que los efectos desinhibidores sólo se producen si existen ya las inhibiciones. No obstante, en muchos casos de conducta desviada, el modelo actúa de una manera que resulta a la vez nueva para el observador y socialmente rechazada; en tales casos, pueden producirse los tres efectos simultáneamente, y entonces se hace prácticamente imposible identificar su relativa contribución al origen de la desviación.

5.2.2 MODELO DE APRENDIZAJE ACUMULATIVO.

Gagné ha sintetizado el conocimiento que se refiere a varios tipos de aprendizaje y ha formulado un *modelo de aprendizaje acumulativo*. Según este autor, los efectos del aprendizaje son acumulativos, es decir, que cada individuo desarrolla destrezas de mayor nivel y adquiere más conocimiento en la medida en que asimila capacidades que se forman sucesivamente una sobre otra. Estas capacidades se refieren a los ocho tipos de aprendizaje que ya hemos estudiado suficientemente. La naturaleza acumulativa que caracteriza a estas ocho clases, desde el simple aprendizaje de una señal, hasta la compleja solución de un problema. Otra característica del modelo consiste en que explica el aprendizaje en función de las condiciones internas del estudiante que son esenciales en cada uno de los tipos mencionados (condiciones internas), como también las condiciones de la situación en que ocurre el aprendizaje (condiciones externas). Vamos a considerar ahora los tipos de aprendizaje y las condiciones en que se llevan a cabo.

5.2.2.1 TIPOS DE APRENDIZAJE

Como ya se dijo, existen ocho tipos de aprendizaje, sucesivos, cada uno de los cuales, empezando en el tercero, se construye sobre los anteriores. (El aprendizaje E-+R no depende del aprendizaje por señales). Ahora, trataremos de dar una definición breve de cada uno de los ocho tipos, de señalar sus condiciones propias, y de dar algunas de sus aplicaciones a la enseñanza escolar. Gagné teorías del aprendizaje (1970).

Tipo 1: aprendizaje por señales. Este tipo de aprendizaje corresponde directamente al condicionamiento clásico. Aprender a responder ante las señales es una forma común de la conducta refleja que se da en los animales, y en menor escala en los seres humanos. Como se observó en la discusión sobre el condicionamiento clásico, el aprendizaje por señales se presenta cuando ocurren dos estímulos casi simultáneamente y la respuesta emitida previamente por el solo estímulo incondicionado es provocada por el nuevo estímulo condicionado. Las respuestas relativas a las señales condicionadas son difusas y emotivas y el organismo las produce involuntariamente. Conocemos muy bien las respuestas reflejas tales como el parpadeo ante un destello de luz o un soplo de viento o como la tensión muscular causada por un sonido fuerte y súbito.

Los sentimientos de agrado y desagrado que tienen que ver con las personas, las cosas y los acontecimientos pueden adquirirse mediante el aprendizaje por señales. De igual manera, las palabras condicionadas a respuestas no condicionadas, pueden convertirse en señales de control de la conducta. Por ejemplo, el maestro puede convertir una palabra en un grito para llamar la atención de los estudiantes que hacen ruido. En este caso, un estímulo verbal puede producir una respuesta refleja de atención.

Tipo 2: aprendizaje por la respuesta al estímulo (E +R). Los movimientos precisos de los músculos como respuesta a estímulos específicos que se han discriminado entre otros estímulos son el producto principal del aprendizaje E+R. La respuesta que el niño aprende a dar ante una orden como "siéntate", "repite la palabra libro", son el producto muy largo de esta clase de aprendizaje asociativo. Este producto puede llamarse también conexión, operante discriminado o respuesta instrumental.

Un requisito para el aprendizaje E+R es la contigüidad temporal entre la respuesta y el estímulo. Además, se refuerza la respuesta deseada para garantizar que sólo dicha respuesta corresponde su estímulo particular. Otras respuestas que se den estímulo particular no serán reforzadas y por lo consiguiente se extinguirán.

Antes de ir a la escuela los niños han adquirido muchas respuestas del tipo E+R y pueden adquirirse con mucha facilidad durante los años de escuela incluyen aquí las respuestas vocales y los movimientos especializados de las manos, los dedos y los miembros frente a los estímulos verbales que pueden presentarse oralmente o por escrito.

Tipos 3 y 4: cadenas motrices y cadenas verbales. pueden combinar o encadenar dos o más respuestas motrices separadas para desarrollar una habilidad compleja. Se pueden combinar dos o más respuestas verbales, por ejemplo palabras, para formar la asociación de dos voces, como en "sal y pimienta", o se puede formar una frase. Pueden aprenderse también las cadenas que combinan las conexiones motrices con las verbales.

Además, para resistir a la extinción, la cadena debe conducir a un estado de cosas satisfactorias, es decir, debe ser reforzada y este refuerzo debe seguir inmediatamente la respuesta encadenada. Gagné ilustra el encadenamiento de las respuestas motrices y verbales utilizando el acto de una niña al reclamar su muñeca. Quizás inicialmente la niña esté de pie en su cuna, cuando su madre le dice "acuéstate"; la niña ha aprendido a responder a este estímulo con la fórmula E+R. Una vez que se acuesta, se pone en juego la próxima conexión de la secuencia: hace los movimientos incipientes de abrazar a su muñeca ausente, lo cual origina el próximo eslabón en la cadena, es decir, el reclamo de su muñeca.

Tipo 5: aprendizaje discriminatorio. La discriminación de un estímulo específico entre otros estímulos es esencial tanto en el aprendizaje por señas como en la forma simple E+R. Por ejemplo, el sonido de una campana debe provocar una respuesta, dicho sonido particular habrá de discriminarse entre otros.

Tipo 6: aprendizaje conceptual. Gagné define el aprendizaje conceptual en términos de E+R cuando se aprende a dar una respuesta común a estímulos que son diferentes por varios aspectos. Por ejemplo, un niño reacciona ante cualquier perro con la expresión "guau guau", aunque sea de diferente raza color o tamaño. La característica importante del aprendizaje conceptual consiste en que habiendo asimilado un concepto, uno es capaz de identificar otros ejemplos de dicho concepto sin necesidad de más aprendizaje, 10 que no es cierto en los cinco tipos que ya hemos presentado, pues en éstos, toda asociación nueva, cadena o discriminación debe aprenderse una vez que aparece.

Para adquirir un concepto se debe contar con el prerrequisito de las discriminaciones y con la capacidad para dar una respuesta común. Así, para aprender el concepto de perro, el niño debe (a) haber discriminado entre un perro y otros objetos y también entre otro perro y otros objetos y (b) ser capaz de percibir 10 que los dos perros tienen en común y de responder como si fueran de la misma clase. Estas discriminaciones entre perros y otros seres deben estar a la orden del alumno de una contigüidad temporal estrecha de modo que se puedan restablecer con el fin de formar el concepto de perro. Alguna repetición de la secuencia de aprendizaje y la confirmación de la seguridad de las respuestas resultan también esenciales para adquirir los conceptos.

El aprendizaje conceptual tiene una destacada posición en la vida escolar porque los conceptos constituyen parte sustancial del conocimiento organizado en materias tales como las matemáticas, las ciencias, los estudios sociales y el lenguaje. Los niños comienzan a aprender conceptos en el jardín infantil y continúan este aprendizaje a través de los años de escolaridad.

Tipo 7: Aprendizaje por reglas. Gagné define la regla conforme a la terminología E+R como una cadena de dos o más conceptos. El aprendizaje por reglas se ejemplifica al dar las respuestas correctas que corresponden a la información

contenida en afirmaciones del tipo "los triángulos equiláteros son semejantes en su forma"; "un sustantivo puede ser singular o plural"; "una respuesta E+R se fortalece mediante refuerzo". Estas afirmaciones pueden memorizarse y repetirse posteriormente. Sin embargo, para aprender por reglas se debe tener la capacidad de dar las respuestas correctas de acuerdo con las afirmaciones.

Además, se entiende que el individuo que ha adquirido una regla puede aplicarla a la solución de problemas.

Es indispensable en el aprendizaje de reglas poseer los conceptos que ellas encierran y la capacidad para dar las respuestas especificadas por la regla, lo cual es semejante a la capacidad para formar cadenas verbales. De acuerdo con el modelo acumulativo, el encadenamiento de los conceptos para formar reglas es un simple proceso asociativo que se cumple generalmente por medio de instrucciones verbales que terminan proponiendo al alumno que de un ejemplo de la regla y también que la formule.

El aprendizaje de reglas probablemente ocupa un puesto más importante que el aprendizaje conceptual en virtud de su empleo en la vida escolar. La estructura o el material orgánico de muchas materias, tales como la historia y las matemáticas, se concreta en reglas.

Tipo 8; Solución de problemas. La solución de problemas es la forma más elevada del aprendizaje, pues permite al individuo definir nuevas ideas independientemente de los demás.

5.3 TEORÍAS PROCESO ENSEÑANZA-APRENDIZAJE.

5.3.1 LA PERSPECTIVA CONDUCTISTA: SKINNER.

Aunque un gran número de autores podrían consignarse bajo la etiqueta de *conductismo*, sin lugar a dudas, la mayor influencia ejercida en el campo educativo vendrá de la mano de **Skinner**, formulador del condicionamiento operante y la enseñanza programada.

El conductismo parte de una concepción empirista del conocimiento. La asociación es uno de los mecanismos centrales del aprendizaje. La secuencia básica es: Estimulo - Respuesta.

La principal influencia conductista en el diseño de software la encontramos en la teoría del **condicionamiento operante** de Skinner. Cuando ocurre un hecho que actúa de forma que incrementa la posibilidad de que se dé una conducta, este hecho es un reforzador. Según Martí "las acciones del sujeto seguidas de un reforzamiento adecuado tienen tendencia a ser repetidas (si el reforzamiento es positivo) o evitadas (si es negativo). En ambos casos, el control de la conducta

viene del exterior". En palabras de Skinner "toda consecuencia de la conducta que sea recompensante o, para decirlo más técnicamente, reforzante, aumenta la probabilidad de nuevas respuestas".

Sus desarrollos en cuanto al diseño de materiales educativos se materializarán en la enseñanza programada y su célebre máquina de enseñar.

Las primeras utilizaciones educativas de los ordenadores se basan en la enseñanza programada de Skinner, consistiendo en la presentación secuencial de preguntas y en la sanción correspondiente de las respuestas de los alumnos.

A este uso de la computadora se le denominará EAO (o CAI en inglés, Computer Assisted Instruction): se centra en programas de ejercitación y práctica muy precisos basados en la repetición. Bajo las premisas de la individualización de la instrucción, la EAO cobrará un gran auge a partir de mediados de los años 60 de la mano de Patrick Suppes (Delval, 1986; Solomon, 1987).

Tal y como apuntan Araújo y Chadwick (1988), cada paso capacita al sujeto para abordar el siguiente, lo que implica que el material debe elaborarse en pequeñas etapas permitiendo así numerosas respuestas que deben ser convenientemente reforzadas. La secuencia del material será lineal y consustancial a la propia materia en el mayoría de los casos.

Para Skinner, el sujeto no ha de tener ninguna dificultad si el material ha sido bien diseñando. Hay que destacar, la importancia de los buenos programadores de material.

Sintetizando las aportaciones de diversos autores (Colom, Sureda, Salinas, 1988; Martí, 1992) en el siguiente cuadro pasamos a exponer las ventajas e inconvenientes más relevantes de la EAO:

VENTAJAS	INCONVENIENTES
Facilidad de uso; no se requieren conocimientos previos	Alumno pasivo
Existe cierto grado de interacción	No es posible la participación del educador para el planteamiento de dudas, etc.
La secuencia de aprendizaje puede ser programada de acuerdo a las necesidades del alumno	Excesiva rigidez en la secuencia de los contenidos, que impide el tratamiento de respuestas no previstas
Feedback inmediato sobre cada respuesta	No se sabe por qué un ítem es correcto o incorrecto
Favorecen automatización de	Fragmentación de contenidos

habilidades básicas para aprendizajes más complejos	excesivamente uniforme y reductora, sea cual sea la materia
Proporciona enseñanza individualizada	Individualización muy elemental; no tiene en cuenta el ritmo, no guía

Sin embargo la EAO ha continuado desarrollándose solventando algunos de los inconvenientes descritos.

5.3.2 AL APRENDIZAJE SIGNIFICATIVO DE AUSUBEL.

La *teoría del aprendizaje significativo* de Ausubel se centra en el aprendizaje de materias escolares fundamentalmente. La expresión "significativo" es utilizada por oposición a "memorístico" o "mecánico".

Para que un contenido sea significativo ha de ser incorporado al conjunto de conocimientos del sujeto, relacionándolo con sus conocimientos previos.

Ausubel destaca la importancia del aprendizaje **por recepción**. Es decir, el contenido y estructura de la materia los organiza el profesor, el alumno "recibe". Dicha concepción del aprendizaje se opondría al aprendizaje **por descubrimiento** de Bruner.

En cuanto a su influencia en el diseño de software educativo, Ausubel, refiriéndose a la instrucción programada y a la EAO, comenta que se trata de medios eficaces sobre todo para proponer situaciones de descubrimiento y simulaciones, pero no pueden sustituir la realidad del laboratorio.

Destaca también las posibilidades de los ordenadores en la enseñanza en tanto posibilitan el control de muchas variables de forma simultánea, si bien considera necesario que su utilización en este ámbito venga respaldada por "una teoría validada empíricamente de la recepción significativa y el aprendizaje por descubrimiento" (Ausubel, Novak y Hanesian, 1989, 339).

Sin embargo, uno de los principales problemas de la EAO estriba en que "no proporciona interacción de los alumnos entre sí ni de éstos con el profesor" (Ausubel, Novak y Hanesian, 1989, 263). Señala también el papel fundamental del profesor, por lo que respecta a su capacidad como guía en el proceso instructivo ya que "ninguna computadora podrá jamás ser programada con respuestas a *todas* las preguntas que los estudiantes formularán" (Ausubel, Novak y Hanesian, 1989, 339).

Por otra parte, prefiere la instrucción programada mediante libros y critica la técnica de fragmentación en pequeños pasos propia de la EAO inicial, y se muestra partidario de aquellos materiales bien estructurados que favorecen la individualización.

5.3.3 APRENDIZAJE POR DESCUBRIMIENTO: BRUNER.

Aprendizaje por descubrimiento es una expresión básica en la teoría de Bruner que denota la importancia que atribuye a la acción en los aprendizajes. La resolución de problemas dependerá de como se presentan estos en una situación concreta, ya que han de suponer un reto, un desafío que incite a su resolución y propicie la transferencia del aprendizaje. Los postulados de Bruner están fuertemente influenciados por Piaget.

Lo más importante en la enseñanza de conceptos básicos, es que se ayude a los niños a pasar progresivamente de un pensamiento concreto a un estadio de representación conceptual y simbólica más adecuada al pensamiento. De lo contrario el resultado es la memorización sin sentido y sin establecer relaciones. "Es posible enseñar cualquier cosa a un niño siempre que se haga en su propio lenguaje" (Araujo y Chadwick, 1988, 41). Según esto, y centrándonos en un contexto escolar, si es posible impartir cualquier materia a cualquier niño de una forma honesta, habrá que concluir que todo currículum debe girar en torno a los grandes problemas, principios y valores que la sociedad considera merecedores de interés por parte de sus miembros.

Por otra parte, refiriéndonos a los materiales para el aprendizaje, Bruner propondrá la estimulación cognitiva mediante materiales que entrenen en las operaciones lógicas básicas.

El descubrimiento favorece el desarrollo mental, ya que consiste en transformar o reorganizar la evidencia de manera de poder ver más allá de ella.

Sobre una secuencia instructiva:

- * Disponer la secuencia de forma que el estudiante perciba la estructura.
- * Promover la transferencia.
- * Utilización de contraste.
- * Ir de lo concreto a lo abstracto en función del grado de maduración del sujeto.
- * Posibilitar la experiencia de los alumnos.
- * Revisiones periódicas a conceptos ya aprendidos (*currículum en espiral*).

Proceso de enseñanza:

- * Captar la atención.
- * Analizar y presentar la estructura del material de forma adecuada.
- * Importante que el alumno describa por si mismo o que es relevante para la resolución de un problema.
- * Elaboración de una secuencia efectiva.
- * Provisión de refuerzo y retroalimentación que surge del éxito de problema resuelto.

5.3.4. LA TEORÍA DE PIAGET.

El enfoque básico de Piaget es la epistemología genética, es decir, el estudio de cómo se llega a conocer el mundo externo a través de los sentidos atendiendo a una perspectiva evolutiva.

Para Piaget el desarrollo de la inteligencia es una adaptación del individuo al medio. Los procesos básicos para su desarrollo son: adaptación (entrada de información) y organización (estructuración de la información). Los autores Araujo y Chadwick mencionan que la adaptación es un equilibrio que se desarrolla a través de la asimilación de elementos del ambiente y de la acomodación de esos elementos por la modificación de los esquemas y estructuras mentales existentes, como resultado de nuevas experiencias.

Establece tres estadios del desarrollo, que tienen un carácter universal:

Sensoriomotor, operaciones concretas y operaciones formales.

Desde esta óptica, el planteamiento de una secuencia de instrucción:

- * Ha de estar ligada al nivel de desarrollo del individuo (aunque un individuo se encuentre en un estadio puede haber regresiones, y también puede darse que en determinados aspectos el individuo esté más avanzado que en otros).
- * La secuencia ha de ser flexible.
- * El aprendizaje se entiende como proceso.
- * Importancia de la actividad en el desarrollo de la inteligencia.

* Los medios deben estimular experiencias que lleven al niño a preguntar, descubrir o inventar.

* Importancia del ambiente.

Si bien Piaget no se mostrara partidario de la "instrucción por ordenador", la influencia de sus ideas se dejará notar fuertemente en Papert.

5.3.5. PROCESAMIENTO DE LA INFORMACIÓN: GAGNÉ.

Su teoría pretende ofrecer unos fundamentos teóricos que puedan guiar al profesorado en la planificación de la instrucción.

En su teoría, aprendizaje e instrucción se convierten en las dos dimensiones de una misma teoría, puesto que ambos deben estudiarse conjuntamente.

El fundamento básico es que para lograr ciertos resultados de aprendizaje es preciso conocer:

- a) Las condiciones internas que intervienen en el proceso.
- b) Las condiciones externas que pueden favorecer un aprendizaje óptimo.

Siguiendo a Gros, en sus inicios sus estudios tienen un enfoque cercano al conductismo y progresivamente irá incorporando elementos de otras teorías. Así podría decirse que Gagné, aunque se sitúa dentro del cognitivismo, utiliza elementos de otras teorías para elaborar la suya:

- Conductismo: especialmente de Skinner, da importancia a los refuerzos y el análisis de tareas.
- Ausubel: la importancia del aprendizaje significativo y de la motivación intrínseca.
- Teorías del procesamiento de la información: el esquema explicativo básico sobre las condiciones internas.

¿Cómo explica Gagné las diferentes **condiciones internas** que intervienen en el aprendizaje? Elabora un esquema que muestra las distintas fases en el proceso de aprendizaje, teniendo en cuenta que estas actividades internas tienen una estrecha conexión con las actividades externas, lo que dará lugar a determinados resultados de aprendizaje. Estas fases son: motivación, comprensión, adquisición, retención, recuerdo, generalización, ejecución y realimentación. Veamos pues como las condiciones externas afectan a los diferentes procesos internos que tienen lugar durante el aprendizaje.

Gagné define las condiciones externas como aquellos eventos de la instrucción que permiten que se produzca un proceso de aprendizaje. Viene a ser la acción que ejerce el medio sobre el sujeto. Así, la finalidad del diseño instructivo es intentar que estas condiciones externas sean lo más favorables posibles a la situación de aprendizaje.

Se trata, pues, de organizar las condiciones externas para alcanzar un determinado resultado de aprendizaje, adecuando la instrucción a cada proceso de aprendizaje: ordenar los factores externos para mejorar la motivación del alumno, su atención, su adquisición, su retención, etc.

Según los resultados de aprendizaje que se pretendan alcanzar deberán organizarse las condiciones externas. Para Gagné dependiendo del tipo de aprendizaje a realizar se requerirán diferentes tipos de capacidades: habilidades intelectuales, información verbal, estrategias cognitivas, actitudes o destrezas motoras.

Si hasta aquí hemos sintetizado los fundamentos de su teoría del aprendizaje, veamos ahora las bases de su *teoría de la instrucción*.

Siguiendo las aportaciones de Gros para realizar el diseño instructivo los pasos a seguir son los siguientes:

- Identificar el tipo de resultado que se espera de la tarea que va a llevar a cabo el sujeto (lo que viene a llamarse "análisis de la tarea"). Ello posibilitaría descubrir qué condiciones internas son precisas y qué condiciones externas son convenientes.
- Una vez determinado el resultado que se desea alcanzar hay que identificar los componentes porcentuales de la tarea, es decir, los requisitos previos, de manera que sirvan de apoyo al nuevo aprendizaje.

Teniendo en cuenta que la teoría de Gagné pretende ofrecer un esquema general como guía para que los educadores creen sus propios diseños instructivos, adecuados a los intereses y necesidades de los alumnos, veamos la repercusión de su teoría en el diseño de *software*.

Las aportaciones de Gagné supusieron una alternativa al modelo conductista para el diseño de programas, centrándose más en los *procesos* de aprendizaje. Sus dos contribuciones más importantes son según Gros:

- a) Sobre el tipo de motivación (los refuerzos). Considerar en un programa el refuerzo como motivación intrínseca (recordemos que en un programa conductista el refuerzo es externo). Por ello, el *feedback* es informativo, que no sancionador, con el objeto de orientar sobre futuras respuestas.

b) El modelo cognitivo de Gagné es muy importante en el diseño de *software* educativo para la formación. Su teoría ha servido como base para diseñar un modelo de formación en los cursos de desarrollo de programas educativos. En este sentido, la ventaja de su teoría es que proporciona pautas muy concretas y específicas de fácil aplicación.

En síntesis, la teoría de Gagné proporciona unas pautas de trabajo para la selección y ordenación de los contenidos y las estrategias de enseñanza, siendo así de gran utilidad para los diseñadores. Es de destacar la labor de **Merrill**, que desarrollará una teoría de la instrucción (no de aprendizaje) a partir de la Gagné.

En la actualidad, un objetivo prioritario de Merrill es el desarrollo de modelos prescriptivos para la elaboración de materiales educativos informáticos. Merrill considera necesario proporcionar una metodología y herramientas que sirvan de guía en el diseño y desarrollo de materiales informáticos educativos. Considera la fase de desarrollo como fundamental para un uso efectivo del ordenador en educación, añadiendo que la finalidad del ordenador es ser de utilidad al profesor, no sustituirlo.

5.3.6. EL CONSTRUCTIVISMO DE PAPERT.

Papert, creador del lenguaje LOGO, propone un cambio sustancial en la escuela: un cambio en los objetivos escolares acorde con el elemento innovador que supone el ordenador.

El lenguaje LOGO será el primer lenguaje de programación diseñado para niños. Utilizará instrucciones muy sencillas para poder desplazar por la pantalla el dibujo de una tortuga, pudiendo construir cualquier figura geométrica a partir de sus movimientos. Su pretensión básica es que los sujetos lleguen a dominar los conceptos básicos de geometría.

Para Papert, el ordenador reconfigura las condiciones de aprendizaje y supone nuevas formas de aprender.

Ya hemos comentado que una fuente importante de su obra serán las teorías de Piaget, con quien estuvo estudiando durante cinco años en el Centro de Epistemología Genética de Ginebra. Sin embargo, Piaget no veía mayores ventajas en el uso del ordenador para "modernizar la clase de estructuras mentales que postulaba", Papert se sintió rápidamente atraído por esa idea. Tanto es así que pronto entró en contacto con los investigadores pioneros en Inteligencia Artificial, campo del que recibiría también notorias influencias.

Es de aquí que recogerá su interés por simular con el ordenador los procesos cognitivos con el fin de estudiar con más detalle su naturaleza. Por otro lado, parte

de los postulados piagetianos, entendiendo al sujeto como agente activo y "constructivo" del aprendizaje.

Para ello, Papert plantea a Piaget desde una vertiente "más intervencionista". Así, dos serán los aspectos de este autor sobre los que Papert incidirá más, máxime entendiendo que Piaget no los desarrolló suficientemente: las estructuras mentales potenciales y los ambientes de aprendizaje.

Intentará que mediante el ordenador el niño pueda llegar a hacerse planteamientos acerca de su propio pensamiento, tarea esta difícilmente realizable sin su concurrencia.

El lenguaje LOGO será una pieza clave, pues mediante la programación el niño podrá pensar sobre sus procesos cognitivos, sobre sus errores y aprovecharlos para reformular sus programas (Martí, 1992). En otras palabras, la programación favorecerá las actividades metacognitivas.

El lenguaje LOGO supone un material lo suficientemente abierto y sugerente para elaborar sus propios proyectos, modificarlos y mejorarlos mediante un proceso interactivo.

Para Papert la utilización adecuada del ordenador puede implicar un importante cambio en los procesos de aprendizaje del niño. Se trata, pues, de un medio revolucionario, ya que puede llegar a modificar las formas de aprender.

Pero el uso del ordenador no debe limitarse al uso escolar tradicional, relegando al alumno a un segundo plano. El ordenador debería ser una herramienta con la que llevar a cabo sus proyectos y tan funcional como un lápiz.

La visión de Papert sobre las posibilidades del ordenador en la escuela como una herramienta capaz de generar cambios de envergadura es ciertamente optimista: "La medicina ha cambiado al hacerse cada vez más técnica; en educación el cambio vendrá por la utilización de medios técnicos capaces de eliminar la naturaleza técnica del aprendizaje escolar" (Papert).

5.3.7. CONSTRUCTIVISMO Y MEDIACIÓN.

Martí propone la superación de las limitaciones a los métodos de Papert mediante una propuesta basada en un doble eje: Aplicación a situaciones específicas instructivas del constructivismo y Mediación del aprendizaje (a través del medio informático y a través de otras personas).

Es posible que a través de la exploración individual el sujeto pueda adquirir determinados esquemas generales de conocimiento, pero mucho más difícil será que consiga alcanzar aprendizajes específicos.

Será necesario definir la situación instructiva partiendo de las ideas previas de los sujetos, de sus intuiciones y también será preciso definir el tipo de intervención de otras personas: profesor y alumnos.

La utilización de un determinado vehículo o medio para la aprehensión de los significados supone tener en cuenta las características específicas de ese medio. Así, el ordenador propiciará un contexto de aprendizaje diferente al de otro medio.

Asimismo, partiendo de los postulados vygotskianos cabe destacar el papel del adulto y los iguales en el proceso de aprendizaje, ofreciendo una labor de *andamiaje* que apoyará al sujeto en su aprendizaje. Para entender el concepto de *andamiaje* es preciso hacer referencia a otro punto clave en la teoría de Vygotsky; nos referimos al concepto de *Zona de Desarrollo Próximo* (ZDP). Como Vygotsky señala "no es otra cosa que la distancia entre el nivel real de desarrollo, determinado por la capacidad de resolver independientemente un problema, y el nivel de desarrollo potencial, determinado a través de la resolución de un problema bajo la guía de un adulto o en colaboración con otro compañero más capaz" (Vygotski).

En este sentido, algunos de los autores de tendencia neovygotskiana destacan el importante papel que juega el profesor en la utilización de software instructivo. Es el caso de Mercer y Fisher (1992), para los que el papel más relevante en todo proceso de enseñanza-aprendizaje reside en la comunicación, en el contexto cultural y en el lugar donde dicho proceso se lleva a cabo. Así, los autores aluden al ya mencionado concepto de *andamiaje*, o a la ayuda que el profesor ofrece al alumno para que pueda solventar por si mismo una situación problemática, para determinar su aplicabilidad a situaciones de EAO.

Mercer y Fisher consideran que pese a la importancia de la fase de diseño de *software*, en cuanto a los resultados instructivos, su aplicación en cada situación distinta supondrá también unos procesos y problemática diferentes. De esta manera, los procedimientos y resultados de cualquier actividad basada en el ordenador surgirán a través de la charla y actividad conjunta entre maestro y alumnos. Es decir, el mismo *software* usado con combinaciones diferentes de maestros y alumnos en ocasiones diferentes, generará actividades distintas. Estas actividades distintivas se llevarán a cabo en escalas de tiempo diferente, generarán problemas diferentes para los alumnos y maestros y casi tendrán ciertamente resultados de aprendizaje diferentes. Aparte del propio software, la influencia fundamental en la estructura y resultados de una actividad basada en el ordenador vendrá ligada a la figura del maestro.

5.3.8 RETENCIÓN Y TRANSFERENCIA

se dice que se ha logrado una retención cuando ha alcanzado un resultado cuando se incorpora por primera vez en el cuadro de conducta del alumno. Por ejemplo, cuando un niño deletrea correctamente una palabra por primera vez o resuelve un problema, decimos que tal niño ha logrado este resultado particular (retención). Cuando más adelante el niño repite correctamente tales actividades, concluimos que lo que fue aprendido inicialmente se retiene o recuerda.

En general se dice que una persona cualquiera que sea tiene una retención cuando logra resolver de manera adecuada algún problema que se le presente teniendo así un resultado particular de ella y que ese conocimiento lo almacena o lo recuerda en ocasiones que se le suele presentar una situación parecida y ya cuenta con los precedentes para poderla resolver.

La transferencia de conocimientos tiene lugar cuando cualquier hecho que se aprenda en una determinada situación se pueda utilizar en una situación nueva o diferente.

Para estimular la retención y la transferencia se pueden utilizar las siguientes preguntas: ¿Cuál es la causa del olvido? ¿Qué condiciones facilitan la retención? ¿Qué resultados del aprendizaje tienen la mayor posibilidad de transferencia? ¿Qué condiciones facilitan la transferencia a partir de una situación original de aprendizaje?.

CAPITULO 6

Caso

Práctico

(Modelo de sistema para realizar material didáctico de apoyo a docentes)

6. CASO PRACTICO.

6.1 DESCRIPCIÓN DEL MODELO.

En este proyecto propone facilitar la tarea de crear un software el cual pudiera realizar actividades didácticas y tutoriales, ya que el uso de estos son muy útiles en la enseñanza sobretodo a niveles básicos de educación, en aquellas áreas del conocimiento donde se presentan problemas. Frecuentemente en un libro, por ejemplo, el alumno se encuentra con la resolución impresa del problema en un solo paso, con todos los procesos intermedios obviados; un tutorial interactivo o una actividad interactiva por medio de la computadora, posibilita al alumno indagar todas las etapas intermedias hasta lograr el conocimiento o adquirir la destreza buscada.

Justificación

El trabajo consiste en un software, que auxilie a los docentes para la enseñanza y el aprendizaje autónomo de diferentes tipos de información ya que también permite crear tutoriales o actividades didácticas, para un tema en específico que se quiera desarrollar.

El sistema consta, por una parte de un entorno virtual de aprendizaje desarrollado en Visual Basic 2005 Express Edition, con una interfaz de fácil manejo por medio de botones y pantallas, que permiten al usuario un desarrollo dentro de la aplicación tanto para el desarrollo de actividades didácticas como para la utilización y aprendizaje de los trabajos presentados como una actividad didáctica.

La herramienta Visual Basic 2005 Express Edition, se utilizo para el desarrollo de la aplicación, debido a que es una versión gratuita para desarrollar y por consiguiente no produjo ningún tipo de gasto para adquirir un software de desarrollo, de ser una herramienta bastante completa para el desarrollo de aplicaciones.

La tecnología informática en la educación abre nuevos objetivos de investigación y desarrollo educativo, al considerar la posibilidad que brinda la computadora para:

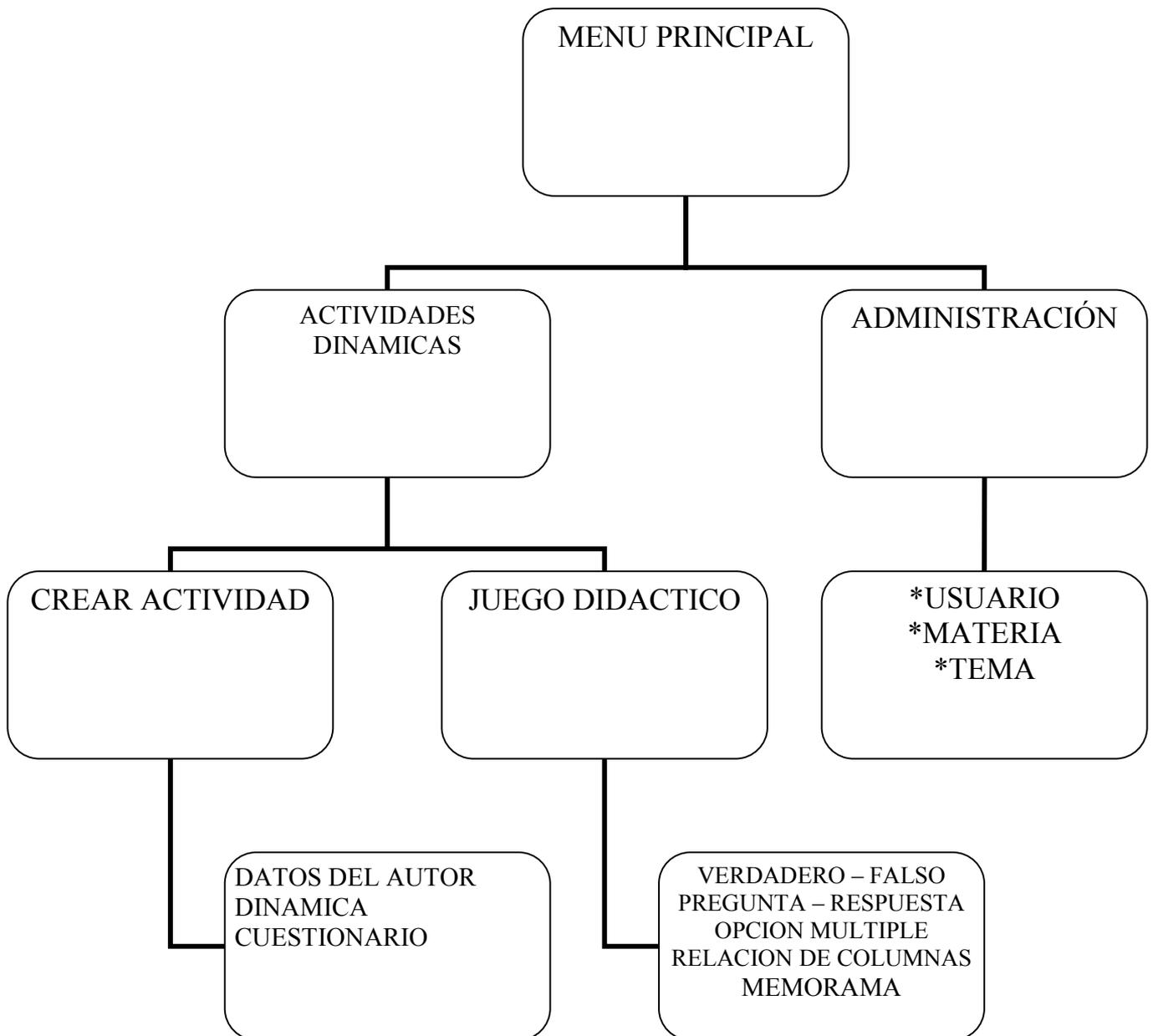
- convertirse en excelente herramienta para la simulación y de materiales para,
- crear plataformas que permiten la comunicación y brindan un gran beneficio a los protagonistas de la educación, y
- promover el desarrollo de las capacidades individuales, sin importar qué diferencias cognitivas o físicas posean las personas.

Esto plantea un desafío al sistema educativo tradicional proponiendo, por medio del uso de las nuevas tecnologías, cambiar la configuración del proceso didáctico y metodológico usado, donde el centro es el docente, quien concentra el conocimiento y lo expone, a modelos más abiertos y flexibles, situando en el centro al estudiante, preparado para la toma de decisiones y la elección de su ruta de aprendizaje, en un proceso de mayor autonomía y autoaprendizaje para toda la vida. El papel del docente debe replantearse, convirtiéndose en un facilitador de la información, con capacidad para la construcción del conocimiento y para crear un verdadero autoaprendizaje tutorizado en los estudiantes.

Es necesario ayudar a los docentes a posibilitar este cambio en sus prácticas, apoyando el trabajo cooperativo y favoreciendo el intercambio de experiencias, la reflexión sobre la práctica y la búsqueda de nuevas propuestas organizativas y metodológicas que potencien su profesionalización docente.

El presente trabajo apunta a mejorar el proceso de enseñanza aprendizaje promoviendo el protagonismo del sujeto (estudiante), como ser que elabora activamente el conocimiento y facilitando el trabajo que para alumno y profesor supone la tarea de formación.

6.2 Diagrama de módulos



6.3 Descripción de tablas de la base de datos.

USUARIO		
Nombre de Columna	Tipo	Descripción
USU_claveint	int	ID del usuario (autoincrementable)
USU_nombre	varchar(50)	Nombre del usuario
USU_aPaterno	varchar(50)	Apellido paterno del usuario
USU_aMaterno	varchar(50)	Apellido materno del usuario
USU_tipoUsuario	int	Tipo de usuario(Alumno, profesor,administrador)
USU_crea	int	Bit de permiso para crear actividades dinámicas
USU_consulta	int	Bit de permiso para consultar actividades dinámicas
USU_participa	int	Bit de permiso para participar en actividades dinámicas
USU_Fecharegistro	datetime	Fecha en que se ingresa el usuario
USU_cuenta	int	Numero de cuenta del usuario
USU_password	varchar(20)	Password del usuario
USU_activo	int	Bit que indica si el usuario esta ctivo en base o no

TIPOUSUARIO		
Nombre de Columna	Tipo	Descripción
TIPUS_tipo	int	Clave del tipo de usuario
TIPUS_Descripcion	varcha(50)	Decripcion del tipo de usuario (administrador, alumno, profesor)

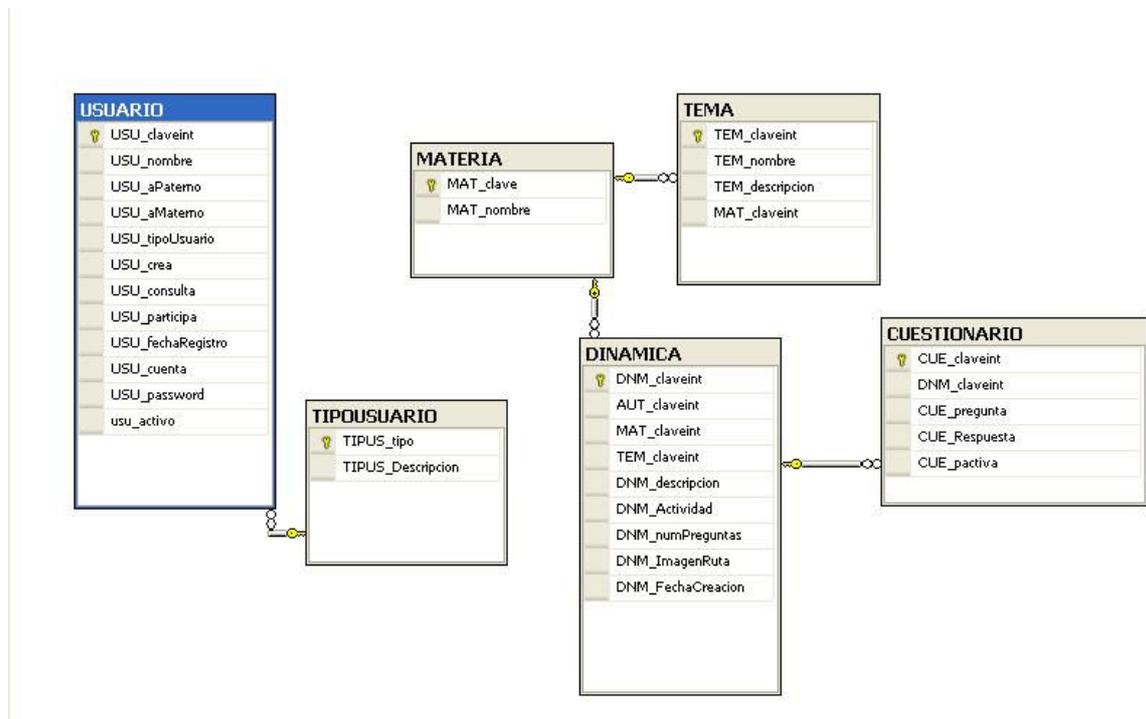
MATERIA		
Nombre de Columna	Tipo	Descripción
MAT_clave	int	ID de la materia (autoincrementable)
MAT_nombre	varchar(100)	Descripción de la materia)

TEMA		
Nombre de Columna	Tipo	Descripción
TEM_Claveint	int	ID del tema (autoincrementable)
TEM_nombre	varchar(200)	Nombre del tema
TEM_descripcion	varchar(200)	Descripción del tema
MAT_claveint	int	Clave de la materia a la que pertenece el tema

DINAMICA		
Nombre de Columna	Tipo	Descripción
DNM_claveint	int	ID de la dinamica (autoincrementable)
AUT_claveint	int	ID Autor de la dinamica
MAT_claveint	int	ID de la materia
TEM_claveint	int	ID del tema
DNM_descripcion	varchar(100)	Descripción de la dinamica
DNM_actividad	varchar(50)	Nombre de la actividad dinamica
DNM_numPreguntas	int	Numero de preguntas del cuestionario
DNM_ImagenRuta	varchar(50)	Ruta de la imagen representativa de la actividad dinamica (solo se utiliza cuando se juega memorama)
DNM_FechaCreacion	datetime	Fecha de creación de la dinamica

CUESTIONARIO		
Nombre de Columna	Tipo	Descripción
CUE_claveint	int	ID del cuestionario (autoincrementable)
DNM_claveint	int	ID del la dinamica
CUE_pregunta	varchar(500)	En este campo se guardan las preguntas del cuestionario
CUE_respuesta	varchar(500)	En este campo se guardan las repuestas del cuestionario
CUE_pactiva	int	Bit que indica si la pregunta esta activa o fue eliminada

6.4 Diagrama de base de datos



6.5 MANUAL DE USUARIO

1. PANTALLA DE BIENVENIDA.

Al ingresar al programa ProAprender, aparecerá la pantalla pidiendo Usuario y contraseña, la cual es proporcionada por el administrador, una vez que éste haya dado de alta en la base al nuevo usuario.



Una vez ingresados el nombre de usuario y contraseña, el usuario debe dar clic en el botón Entrar; así el programa buscará en la base los datos proporcionados por el usuario para dar autorización de proseguir o denegar la entrada a dicho usuario. En caso de que el usuario esté autoriza de entrar, aparecerá el siguiente mensaje.



2. MENU PRINCIPAL

Una vez que ha ingresado al programa, aparecerá el menú inicial, el cual tiene dos opciones:

1. Actividades didácticas
2. Administración



1. Actividades didácticas.

Esta opción permite

- a) Crear actividad.
- b) Entrar a un juego didáctico.



3. CREAR ACTIVIDAD DIDACTICA

En caso de elegir CREAR ACTIVIDAD, se podrá observar la siguiente pantalla que tiene como objetivo ingresar los datos del autor, la dinámica y el cuestionario del que se conformará dicha se dinámica.

Consta de tres apartados:

1. Datos de autor: En este apartado guardar los datos del autor, o modificarlos al dar clic en el botón guardar o modificar que aparecerá de acuerdo a lo que esté realizando.
2. Datos de Dinámica: Aquí deberá guardar los datos de la dinámica que se está creando, es decir, la materia, el tema, la imagen representativa del tema y el número de preguntas. Al dar clic en el botón guardar, se guardarán los datos.
3. Cuestionario: Aquí podrá guardar las preguntas y respuestas que conformarán la dinámica. Cada vez que se agregue un par de pregunta-respuesta dar clic en el botón agregar; así ud. podrá observar las preguntas-respuestas en forma de lista en la parte inferior.

En caso de que quiera eliminar una pregunta-respuesta deberá seleccionarla de la lista de preguntas y dar clic en eliminar.

The screenshot shows a software window titled "Crear A. Didáctica". It is divided into three main sections:

- Datos de Autor:** Includes a text field for "Autor" (containing "Admin") and a "GUARDAR" button.
- Datos de Dinámica:** Includes dropdown menus for "Materia" (selected "CIENCIAS SOCIALES") and "Tema" (selected "INTRODUCCION A LAS CIENCIAS SOCIALES"). It also has a "Nombre de la Dinámica" text field, an "Imagen" text field with an "Examinar" button, and a "No. de Preguntas" dropdown menu.
- Cuestionario:** Features a "Pregunta" text field with an "Agregar" button, a "Respuesta" text field, and two large text areas labeled "Preguntas" and "Respuestas" with "Eliminar" buttons.

Al terminar de agregar los datos de la dinámica, solo deberá dar clic en el botón cancelar que se encuentra en la parte superior derecha del cuadro y podrá observar nuevamente el menú principal.

4. JUEGO DIDACTICO

Para iniciar un juego didáctico, deberá dar clic en el botón JUEGO DIDACTICO.

The screenshot shows a software window titled "MenuPrincipal". At the top right, it says "Usuario Admin". Below this is a blue banner with the text "*** MENU PRINCIPAL ***". In the center, there are two blue buttons: "Actividades Didácticas" and "Administración". At the bottom, there is a dark blue bar with the text "Actividades Didácticas" on the left and a button labeled "Juego Didáctico" on the right. Below the "Juego Didáctico" button is another button labeled "Crear Actividad".

Posteriormente aparecerá la siguiente pantalla para que el usuario elija la materia, el tema y el tipo de juego al que desea entrar; como son:

- Juego de Verdadero-Falso
- Unión de Pregunta respuesta
- Memorama
- Relación de columnas
- Opción múltiple

Elegir Juego Didáctico

Materia: ANALISIS
Tema: ANALISIS
Tipo de Juego: Verdadero Falso

Buscar

Elija un tema

DESCRIPCION	PREGUNTAS	Autor	Clave
ANALISIS Y DISEÑO DE SISTEMAS	6	Admin Admin	67

Salir

Una vez elegidos la materia, tema y juego dar clic en el botón buscar y obtendrá una lista de las actividades encontradas.

Para poder acceder al juego solo tiene que dar doble clic al registro.

5. TIPOS DE JUEGOS DIDACTICOS.

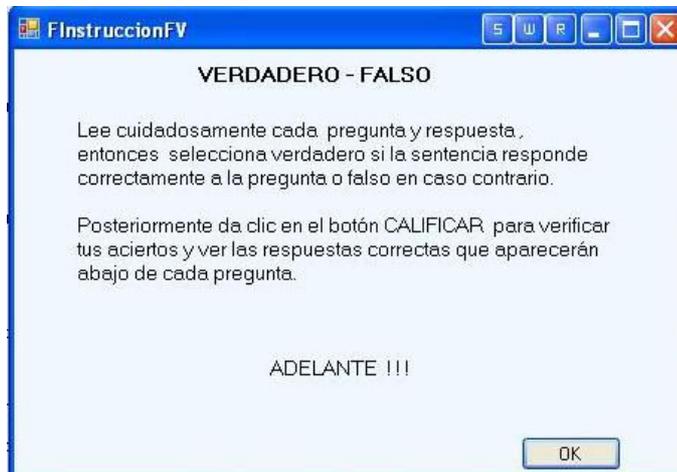
Al principio de cada juego se presentarán las instrucciones.

Además en la pantalla de cada juego aparece un menú en la parte superior izquierda, el cual tiene dos opciones

- Ayuda : que muestra las instrucciones y
- Otro juego: Sale del juego actual.

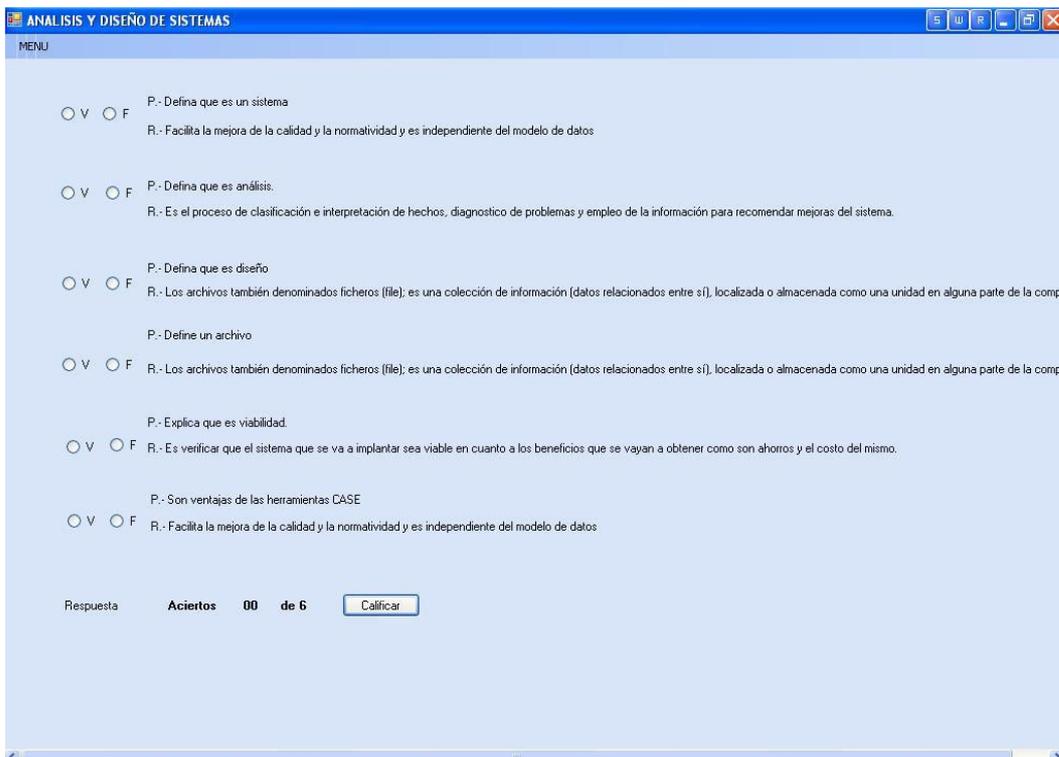
A. JUEGO DE VERDADERO – FALSO:

La primera pantalla que aparecerá son las instrucciones

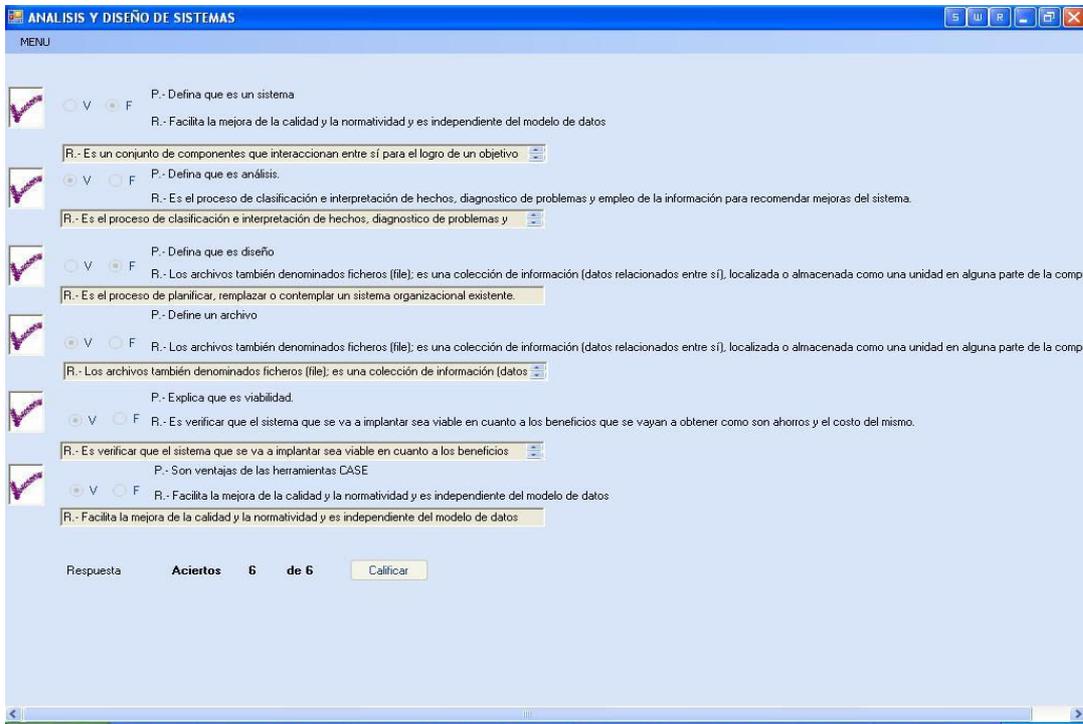


Para proseguir deberá dar clic en el botón de OK, entonces la pantalla del juego aparece para que las preguntas sean respondidas.

V – para verdadero F - para falso

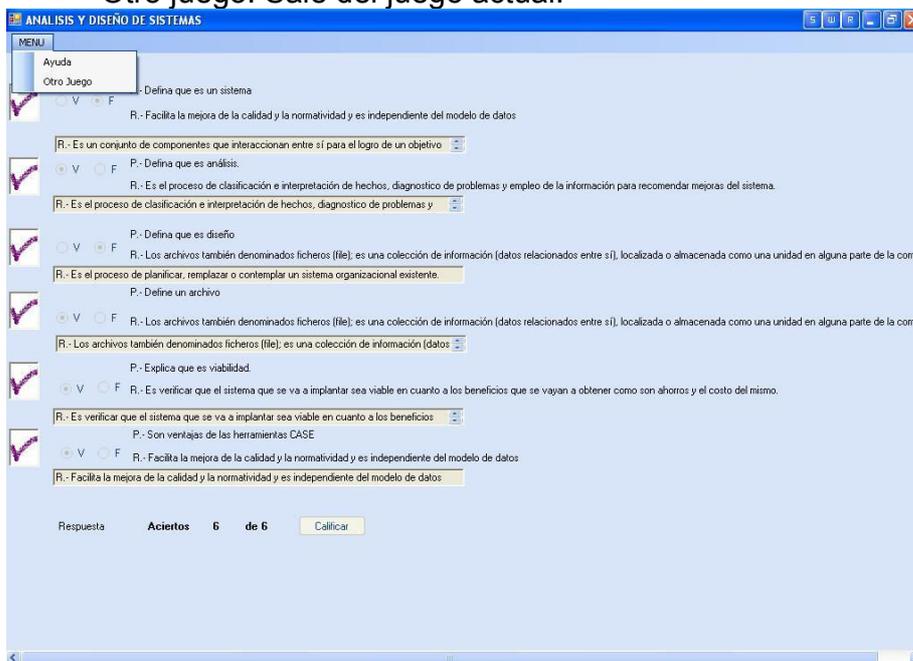


Al terminar deberá dar clic en calificar para que se muestren aciertos, de la siguiente manera.



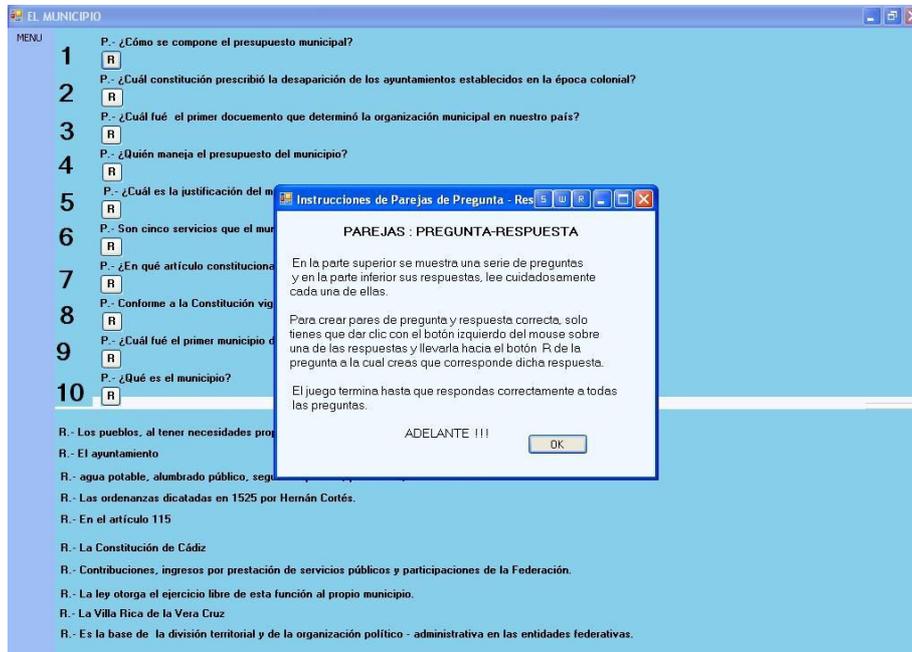
En la pantalla de Este juego aparece un menú en la parte superior izquierda, el cual tiene dos opciones

- Ayuda : que muestra las instrucciones y
- Otro juego: Sale del juego actual.



B. UNION DE PREGUNTA – RESPUESTA.

La primera pantalla que se presenta son las instrucciones del juego

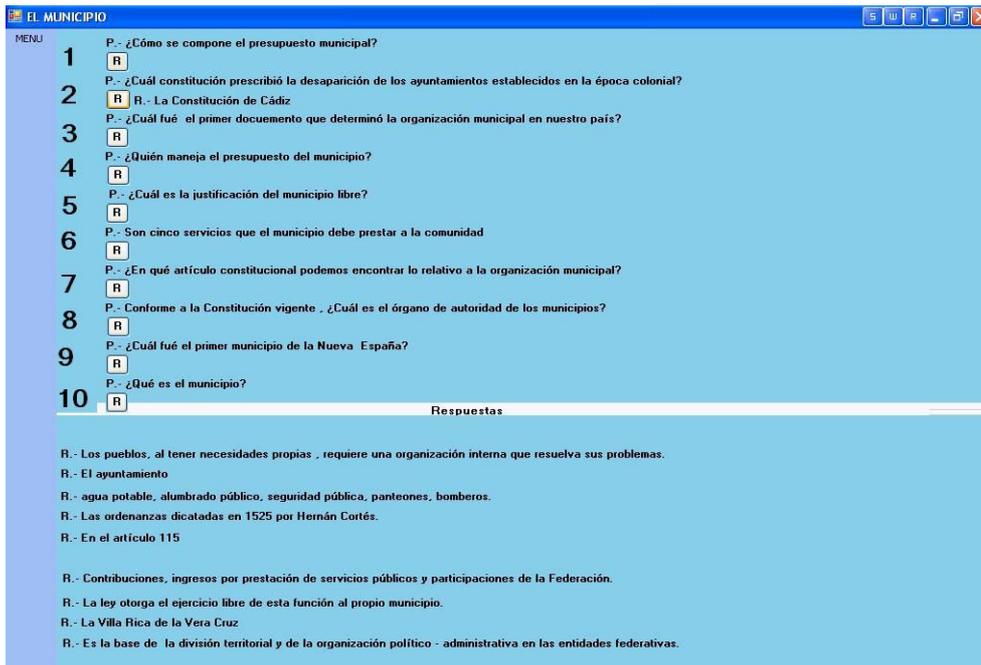


Dar clic en el botón OK para salir de la pantalla de instrucciones y empezar a jugar.

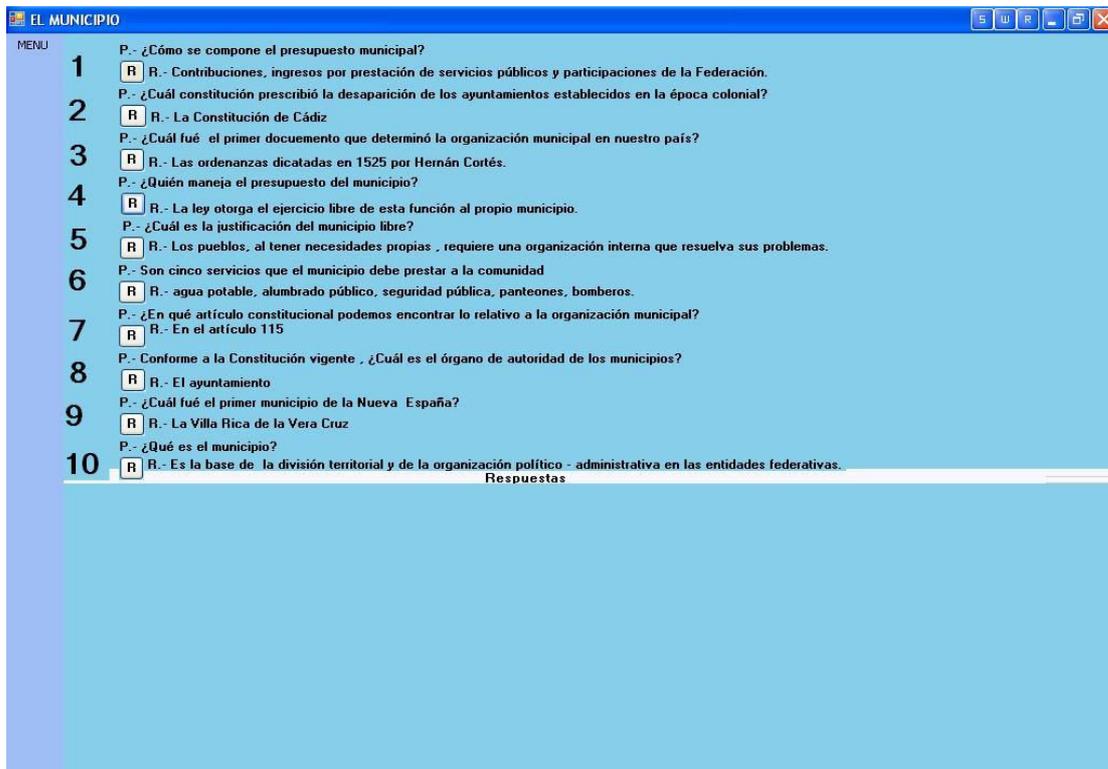
El objetivo de este juego es unir las respuestas que se encuentran en la parte inferior de la pantalla con su pregunta. Dando clic en la respuesta y arrastrándola hasta el botón R (Respuesta) que se encuentra debajo de cada pregunta.

Si la respuesta es incorrecta, ésta regresará a su lugar de origen, entonces el jugador tendrá que encontrar la respuesta correcta e intentarlo nuevamente.

En caso de que la respuesta conteste correctamente a la pregunta, entonces se quedará en la posición de la respuesta a dicha pregunta.

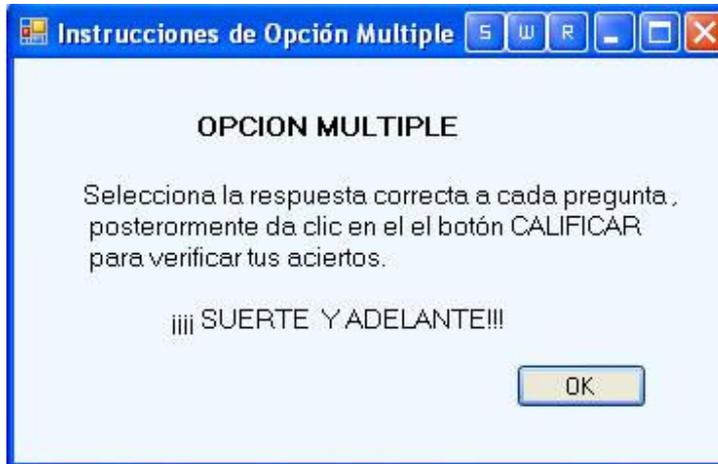


El juego termina hasta que el jugador haya contestado correctamente a las preguntas como se muestra en la siguiente pantalla.

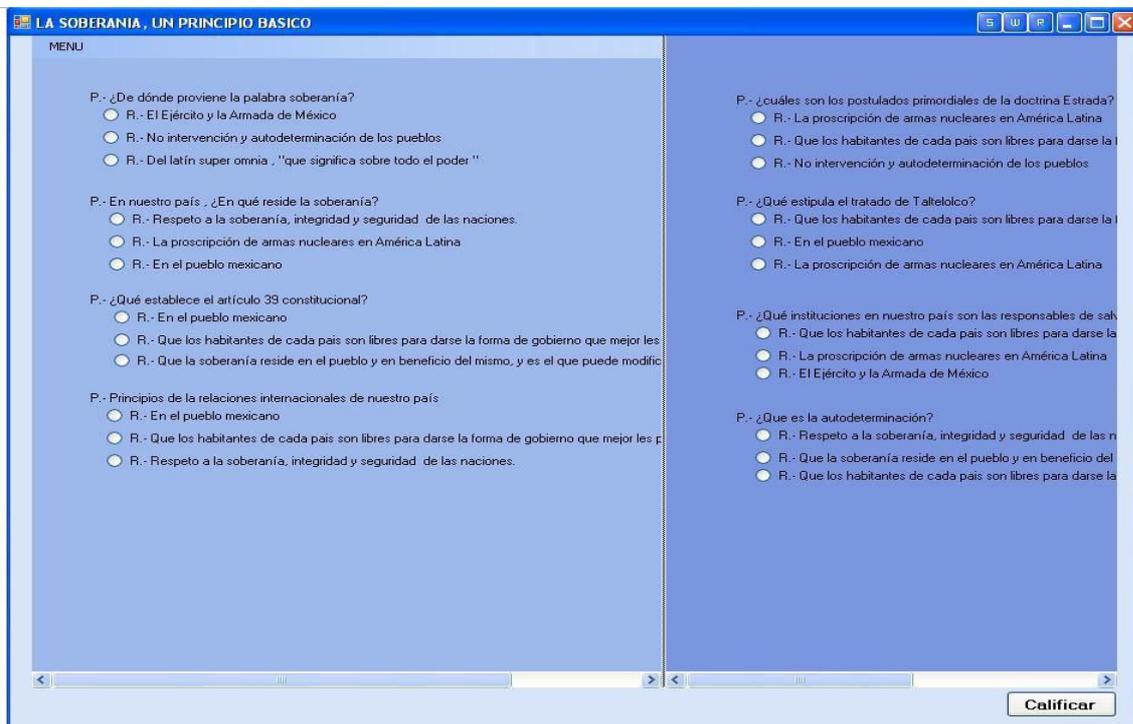


OPCION MULTIPLE

La primera pantalla en el juego son las instrucciones.



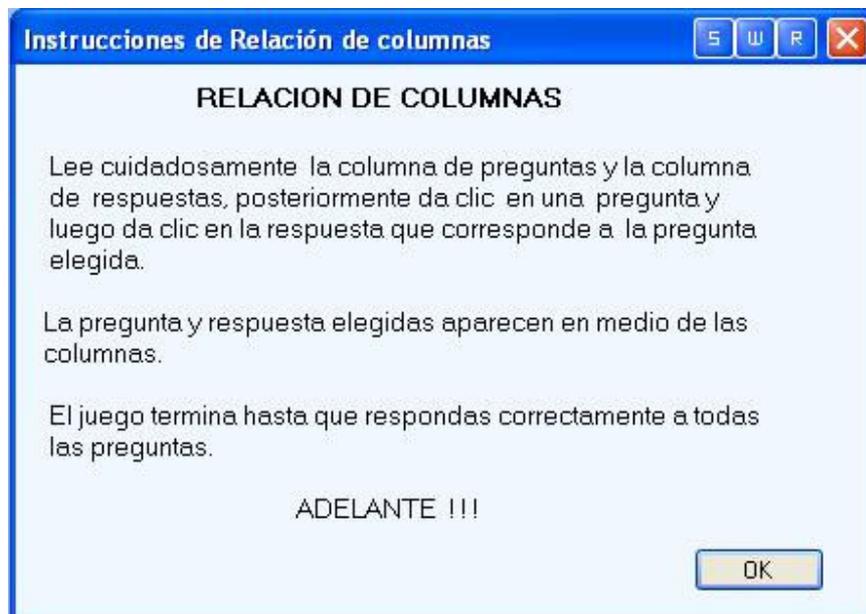
Al dar clic en el botón OK aparecerá el juego de OPCION MULTIPLE, donde el objetivo es seleccionar la opción que responda a la pregunta correctamente



Una vez respondidas las preguntas dar clic en el botón Calificar para verificar las respuestas.

C. RELACION DE COLUMNAS

En este juego la primera pantalla que aparece son las instrucciones



Después de dar clic en el botón OK aparecerá la pantalla del juego; el que se muestra la columna de preguntas a la izquierda y las respuestas a la derecha, así el usuario podrá dar clic en la pregunta y respuesta de tal forma que haga pares de pregunta-respuesta



En caso de que el par de pregunta respuesta sea incorrecto, aparecerá la siguiente pantalla con la leyenda de “Respuesta Incorrecta”

SOBERANIA NACIONAL

MENU

P.- ¿Qué significado tiene nuestro Escudo Nacional? 1

P.- Son tres grupos indígenas asentados actualmente en territorio nacional 2

P.- Autores del Himno Nacional 3

P.- ¿Quiénes eran los encargados de pintar las historias en las culturas mesoamericanas? 4

P.- ¿Qué es la banda presidencial? 5

P.- Tres veces en las que la bandera debe izarse a media asta 6

P.- Tres veces en las que la bandera debe izarse a toda asta 7

P.- ¿En dónde están resguardados los modelos autenticados del escudo nacional? 8

P.- ¿Cuál es la leyenda impresa en la bandera usada por López Rayón? 9

P.- ¿Cuáles son las tres partes de las que está compuesta la bandera? 10

P.- Pintor mexicano que representa en sus cuadros las luchas sociales del pueblo 11

P.- ¿Qué institución interviene directamente en la adquisición de la nacionalidad? 12

Respuesta Incorrecta



Pregunta 1

Respuesta E

A R.- Moharra, asta y regatón

B R.- La fundación de Tenochtitlan

C R.- 28 de Febrero, 21 de Mayp, 13 de Septiembre

D R.- Diego Rivera

E R.- Los tlacuilos

F R.- En el Archivo General de la Nación, Museo Nacional de Historia y Casa de Moneda

G R.- Es una representación de la bandera nacional

H R.- mayas, zapotecos, mixtecos

I R.- Francisco González Bocanegra y Jaime Nunó

J R.- No hizo casa igual por otra nación

K R.- 21 Enero, 5 de Mayo, 27 de Septiembre

L R.- La familia

En caso de que el par pregunta-respuesta sea correcto, aparecerá la leyenda “Respuesta correcta” como aparece en la siguiente pantalla

GLOSARIO DE CONTABILIDAD

MENU

P.- BALANCE GENERAL 1

P.- CARGAR 2

P.- BALANZA 3

P.- ABONAR 4

P.- ALMACEN 5

P.- La CONTABILIDAD se define como: 6

Respuesta Correcta !!!!



Pregunta 1

Respuesta B

A R.- Débitar o Cargar

B R.- Es el estado contable que muestra la situación financiera a una fecha determinada.]

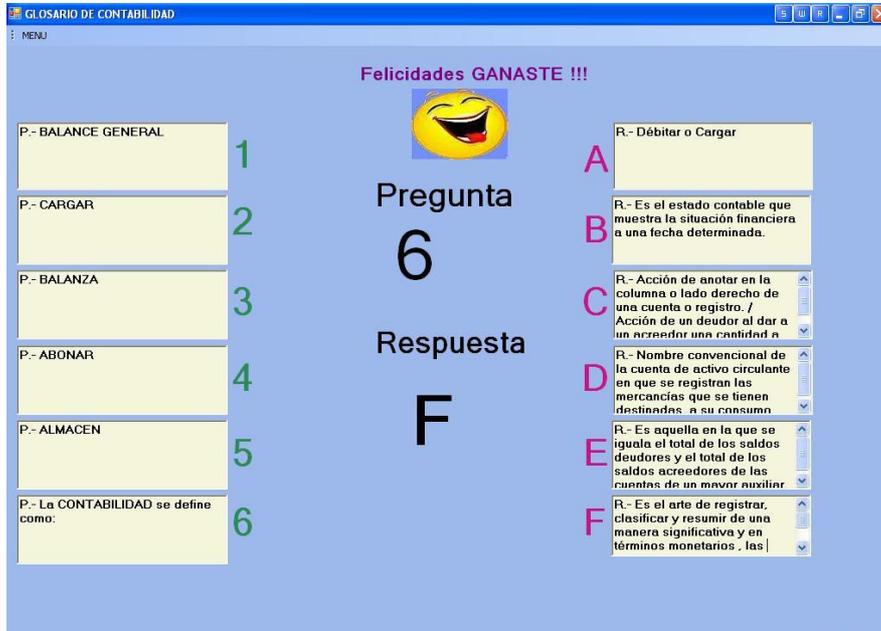
C R.- Acción de anotar en la columna o lado derecho de una cuenta o registro. / Acción de un deudor al dar a un acreedor una cantidad a

D R.- Nombre convencional de la cuenta de activo circulante en que se registran las mercancías que se tienen destinadas a su consumo

E R.- Es aquella en la que se iguala el total de los saldos deudores y el total de los saldos acreedores de las cuentas de un mayor auxiliar

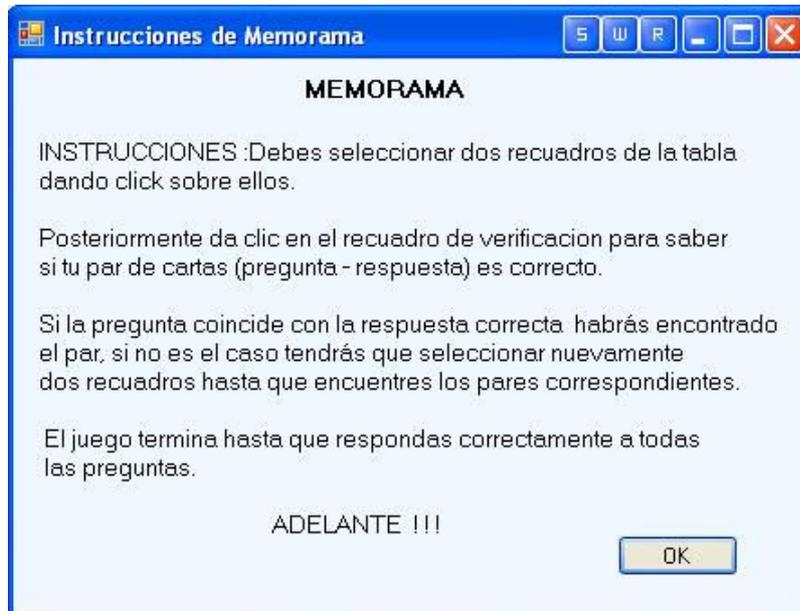
F R.- Es el arte de registrar, clasificar y resumir de una manera significativa y en términos monetarios, las

Al terminar de completar los pares de pregunta-respuesta, usted verá una pantalla como la siguiente.



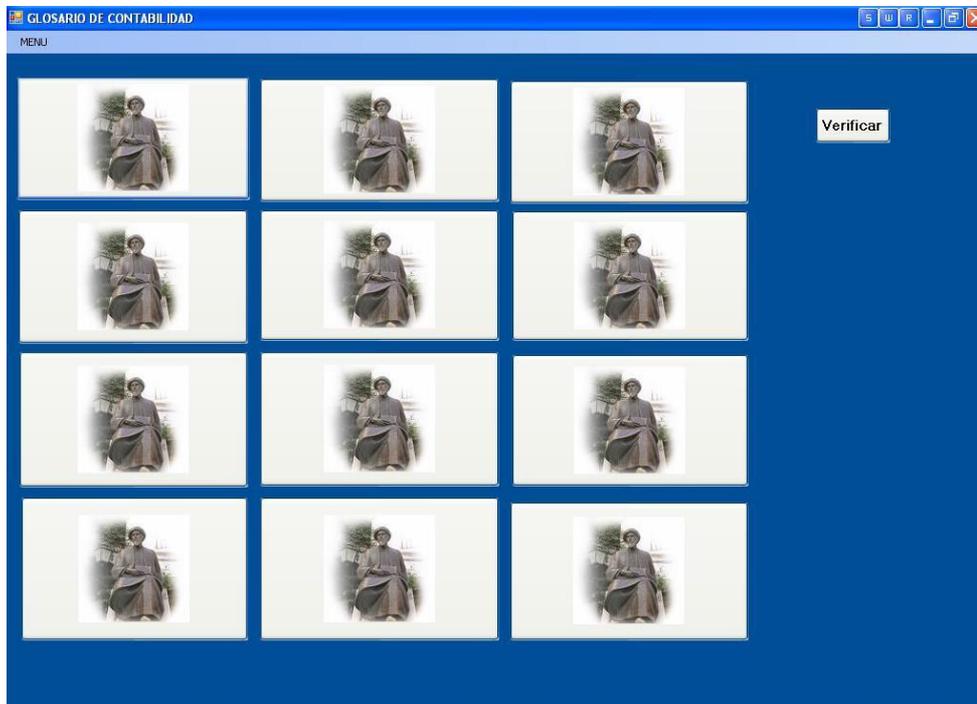
D. MEMORAMA

En el juego de memorama la primera pantalla que aparecerá son las instrucciones, como se muestra a continuación.

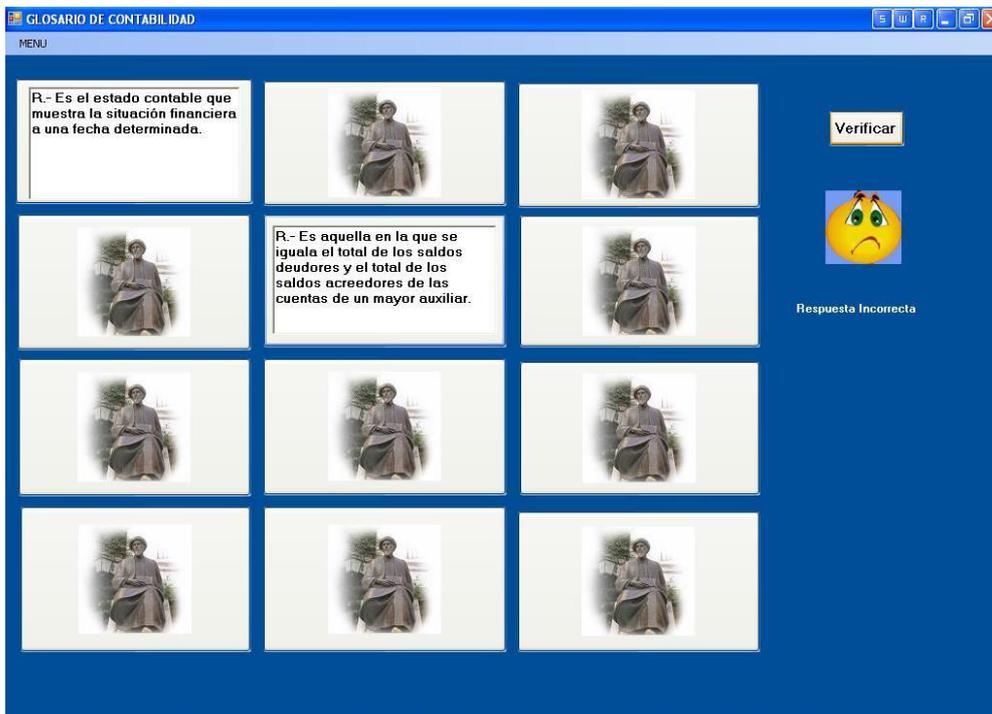


Una vez leídas las instrucciones podrá dar clic para continuar con el juego, y aparecerá el tablero que a continuación se presenta.

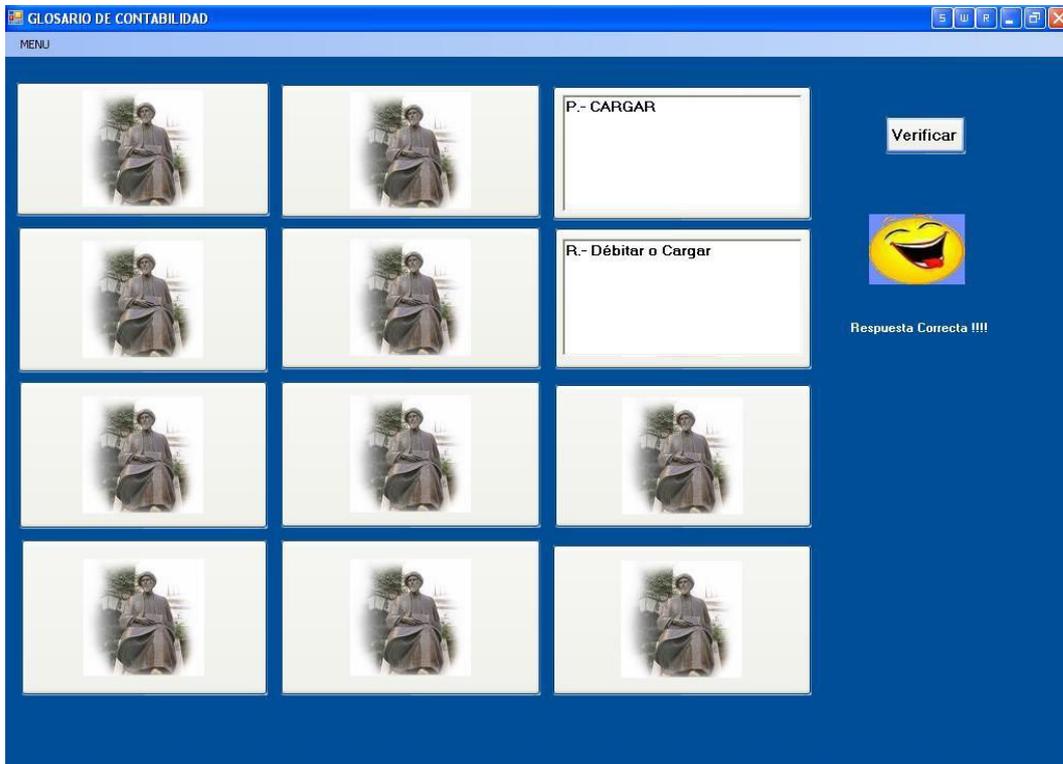
El objetivo de este juego es encontrar los pares correctos de pregunta respuesta, una vez elegidas dos cartas, el usuario deberá dar clic en el botón VERIFICAR para cotejar si coincide la pregunta con la respuesta correcta.



En caso de que el par de pregunta-respuesta sea incorrecto, aparecerá una pantalla como la siguiente con la leyenda de "Respuesta incorrecta"



En caso de que haya acertado al par de pregunta-respuesta, entonces aparecerá una pantalla como la siguiente , con la leyenda “Respuesta correcta”



Una vez que el usuario encuentra todos los pares de pregunta-respuesta el juego termina y aparecerá la leyenda “Felicidades Ganador” como aparece a continuación.



6. ADMINISTRACION.

Este apartado tiene la función de dar de alta o modificar los datos del usuario, materia o tema solo a usuarios dados de alta como administradores.

Consta de tres cuadros cada uno específico a su función:

1. Usuario : Puede dar de alta o modificar los datos de un usuario, es decir, su nombre, apellidos, password y el tipo de usuario que tendrá ya que puede ser Profesor o alumno. Además de los permisos que se le asignarán.
2. Materia : Le ayuda a buscar y dar de alta las materias en caso de que no existe en la base de datos o modificar el nombre.
3. Tema: Le ayuda a buscar y dar de alta los temas en caso de que no existe en la base de datos o modificar el nombre.

The screenshot shows a web application window titled 'Administración' with a user logged in as 'Admin'. The interface is divided into three main sections:

- USUARIO:** A search section with a 'Cuenta' input field and a 'Buscar' button. Below it are 'Datos Personales' with fields for 'Nombre' (containing 'A. Paterno'), 'A. Materno' (containing 'A. Materno'), and 'Password'. There are 'Modificar' and 'Limpiar' buttons. A 'Tipo Usuario' dropdown is set to 'Profesor'. A 'Permisos' section has checkboxes for 'Administrador', 'Crear Actividad', and 'Participante de A. D.'.
- MATERIA:** A search section with a 'Materia' input field and a 'Buscar' button. Below it are 'Modificar Materia' and 'Agregar Materia' sections, each with a 'Materia' input field and a 'Guardar' button.
- TEMA:** A search section with a 'Tema' input field and a 'Buscar' button. Below it are 'Descripción' and 'Materia' (set to 'CIENCIAS SOCIALES') input fields, each with a 'Guardar' button.

1. Administrar un usuario.

Primeramente debe escribir el número de cuenta de usuario para verificar si existe en la base de datos.

En caso de que no exista el usuario, usted podrá darlo de alta

- a) Para dar de alta un usuario solamente tiene que agregar los datos en los cuadros correspondiente a nombre, apellido materno y apellido paterno, el tipo de usuario que desea ingresar y los permisos que se le otorgarán, es decir, como administrador, para poder crear una actividad didáctica o solamente como participante de una actividad.

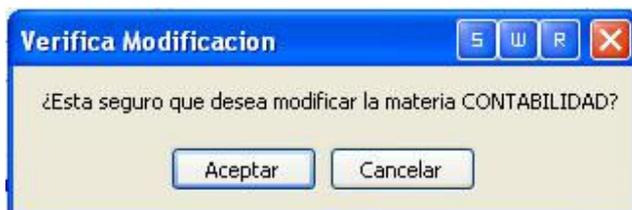
En caso de que el usuario exista, podrá modificar sus datos, como lo muestra la siguiente pantalla:

1. Administrar una materia

En este apartado podrá buscar una materia, escribiendo únicamente el nombre en el cuadro de MATERIA y posteriormente dando clic en el botón BUSCAR

En caso de que la materia exista en la base de datos, ésta aparecerá en el cuadro recuadro de abajo.

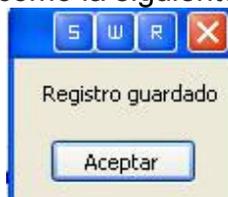
Modificar materia: Para modificar la materia deberá dar doble clic en la materia que desea modificar en el recuadro donde apareció en la búsqueda; posteriormente la materia aparecerá en el segundo cuadro de modificar materia, así ud. Podrá hacer el cambio de la materia y al dar clic en el botón MODIFICAR, la aplicación muestra un cuadro de confirmación



Dar clic en CANCELAR para cancelar la modificación o clic en ACEPTAR para modificar el registro, en este caso aparecerá la siguiente leyenda.



Agregar Materia: Para agregar una materia deberá dar clic en la materia y posteriormente dar clic en el botón GUARDAR. Entonces aparecerá una leyenda como la siguiente:



1. Administrar tema.

Para administrar un tema , lo primero que debe hacer es buscarlo en el primer apartado de tema; para ello deberá escribir el tema y dar clic en BUSCAR.

En caso de que el tema exista, aparecerá enlistado en el cuadro de abajo. En caso de que la materia no exista entonces podrá ingresarla a la base de datos; para ello deberá proporcionar el nombre del tema en el campo tema , la descripción del tema en el cuadro descripción y la materia con la que se relaciona y dar clic en el botón GUARDAR.

Administración Usuario Admin

En este espacio puedes administrar los datos de los usuarios, materias y temas de las aplicaciones didácticas. Salir

USUARIO
Busca el usuario por su número de cuenta. Puedes modificar sus datos o agregar nuevos usuarios.

Cuenta

Datos Personales

Nombre
A. Paterno
A. Materno
Password

Tipo Usuario
Tipo de Usuario:

Permisos

Administrador
 Crear Actividad
 Participante de A. D.

MATERIA
Buscar Materia
Materia:

CLAVE	MATERIA
16	ANALISIS
17	ANALISIS 2

Modificar Materia
Materia:

Agregar Materia
Materia:

TEMA
Tema:
Descripción:
Materia:

CLAVE	TEMA
11	CONCEPTOS BASICOS DE CONTABI

Para salir de este apartado, deberá dar clic en salir y regresará al MENÚ PRINCIPAL.

Conclusión

En el desarrollo de este modelo de sistema, se pudieron aplicar conocimientos adquiridos durante la carrera de las diferentes materias cursadas, además de conjuntar estos conocimientos con la experiencia laboral. Al darnos cuenta de las carencias de materiales con los que cuenta profesores esto en el sentido informático, ya que en escuelas del ámbito público no se encuentran recursos mayores para adquirir paquetes de apoyo al docente, siguiendo así utilizando métodos tradicionales como el pizarrón y el gis.

Este modelo de sistema para realizar material didáctico de apoyo a docentes trata de dar una herramienta para aquellos profesores que aunque no manejen del todo una computadora puedan de forma fácil y sencilla crear material didáctico, para poder impartir su clase o incluso repasar o manejar temas y que sean de un mas fácil comprensión para los alumnos.

El modelo como se menciona anteriormente no trata de ser una base forzosa para la enseñanza solo se plantea para tener una herramienta mas para el apoyo a los docentes manejando los contenidos y se obteniendo así una mayor comprensión de los contenidos de clase, ayudando también a acortar los tiempos de preparación de los materiales de apoyo además de encaminar a los profesores que se empapen en el uso de tecnologías como la computadora.

BIBLIOGRAFÍA

1. Redes Iniciación y Referencia. Jesús Sánchez Allende, Joaquín López Lérica. McGrawHill. Madrid España.2000.
2. Cibersociedad. Los Retos Sociales Ante un Nuevo Mundo Digital. Luis Joyanes. McGrawHill. España. 1997.
3. Ingeniería de Software. Richard Farley. McGrawHill. México. 1994.
4. Análisis y Diseño de Sistemas. Kendall & Kendall. Prentice Hall 3ª Edición. México 1997
5. Análisis y Diseño de Sistemas de Información .Jeffrey L. Whitten. 3ª. Edición España. 1996.
6. El Tutor OnLine. La enseñanza a Través de Internet. Julia Duggleby. Ediciones Deuslo S. A. España 2001.
7. La Educación en la Era de la Informática. Claudia de Moura Castro. Banca Interamericana de Desarrollo. Colombia. 1997.
8. Psicología Cognitiva. Estrategias en la Práctica Docente. Cynthia Klingler, Guadalupe Badillo. México. 2000.
9. Comportamiento Organizacional. Conceptos, Problemas y Prácticas. Angela Kickni, Rober Kretner. McGrawHill.. México 2003.
10. Enciclopedia Técnica de la Educación. Emiliano Martinez Rodríguez, Sergio Sánchez Cerezo. Santillana. Madrid España 1998.
11. Aprendizaje, Habilidades Humanas y Conducta. Herbert J. Klausmeier, William Goodwin. Oxford University Press. 1999.
12. CCNA Companion Guide Networking Academy Program. CISCO Systems Inc., 3a. Edición. CISCO Press. 2004.
13. Redes de Computadoras. Andrew S. Tanenbaum. Prentice Hall. 4a. Ed. 2002.
14. Sistemas de Información Gerencial. Kenneth C. Laudon, Jane P. Laudon. Pearson. 8a. Edición. 2004.
15. Análisis y Diseño de bases de datos. I. T. Hawryszkiewicz, editorial Megabyte, Noriega editores, México 1994.
16. Fundamentos de bases de datos. Hery F. Korth, Abraham Silberschatz, Editorial Mc. Graw Hill, 2a. Edición, España 1993.
17. Análisis y Diseño de sistemas. Centro de Computación Profesional de México. McGrawHill. México. 2001.
18. Análisis y Diseño de sistemas de Información. Senn James A. McGrawHill. México. 1989.
19. Análisis y Diseño Práctico de sistemas Cliente – Servidor. Ruble A. David. Prentice may. México.1997.
20. Fundamentos de sistemas de bases de datos. Ramez A. Elmasri, Shamkant B. Navathe, Editorial Addison Wesley, 3a. Edición ,España 2002.
21. Diseño y Administración de bases de datos. Gary W, James V. Hansen. Prentice Hall. 2ª. Edición. España 1997.
22. Diseño de Bases de datos relacionales con acces y SQL server. Rebecca M. Riordan. McGrawHill. España 2000.
23. Multimedia y el proceso de enseñanza-aprendizaje en la Educación Superior(Multimedia and the Teaching-Learning Process in Higher Education),

- Lamb J.A. 1992.
24. Enseñanza y aprendizaje en la educación superior, Julián López Yáñez, Araceli Estebaranz García,, Carlos Marcelo García, Editorial Octaedro, España 1999.
 25. Understanding attitudes and predicting social behavior. Actitudes de comprensión y predicción de comportamiento social), Ajzen, I., & Fishbein, M. Editorial Prentice-Hall. 1980.
 26. <http://es.wikipedia.org/wiki/E-learning>

Anexo I

Ejemplos del código fuente del modelo de software educativo

A continuación, se mostrara una parte del código fuente desarrollado para ilustrar el desarrollo aunque no se encuentre en su totalidad debido a el gran grueso que ocuparía.

CODIGO DE APLICACIÓN. FORMULARIO PASSWORD

```
Imports System.Data.SqlClient
Public Class FPassword
    Dim oconexion As New Conexion
    Public claveusuario As Integer
    Public nombreusuario As String
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Dim Resultados As DataSet
    Dim a As DataSet
    Dim tabla As DataTable
    Dim cadena As String
'HACE UNA CONSULTA DE LOS DATOS DEL USUARIO QUE QUIERE INGRESAR A
LA APLICACION
cadena = "Select * from usuario where usu_cuenta = " & "" & "" &
TxBoxUsuario.Text.ToString & "" & "" & " And usu_password = " & "" & "" &
TxBoxPd.Text.ToString & "" & ""
'VERIFICA SI EXISTE EL USUARIO
Resultados = oconexion.SelectRows(a, oconexion.sqlCnn.ToString,
cadena)
    If Resultados.Tables.Count > 0 Then
        tabla = Resultados.Tables(0)
        ' SI EL USUARIO EXISTE EN DB , LE DA LA BIENVENIDA
        If tabla.Rows.Count > 0 Then
            MessageBox.Show("Bienvenid@ " &
tabla.Rows(0)("USU_nombre"))
            claveusuario = tabla.Rows(0)("USU_claveint")
            nombreusuario = tabla.Rows(0)("USU_nombre")
        If tabla.Rows(0)("USU_crea") = 0 Then MenuPrincipal.Button1.Visible
= False
        If tabla.Rows(0)("USU_consulta") = 0 Then
            MenuPrincipal.Button4.Visible = False
            MenuPrincipal.Show()
            Me.Hide()
        'SI NO EXISTE EL USUARIO EN DB , MANDA MENSAJE DE USUARIO NO VALIDO
        Else
            MessageBox.Show("Usuario no valido")
        End If
        End If
        'VACIA LA VARIABLE DE RESULTADOS
        Resultados.Clear()
        Resultados.EndInit()
    End Sub
Private Sub Button3_Click_1(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button3.Click
'CIERRA APLICACION
Close()
End Sub
```

FORMULARIO MENU PRINCIPAL

```
Public Class MenuPrincipal
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
'MUESTRA FORMULARIO DE ALTAS
gbADidacticas.Visible = False
Altas.Show()
End Sub
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
'MUESTRA FORMULARIO DE DIDACTICAS
gbADidacticas.Visible = True
End Sub
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)
'MUESTRA FORMULARIO DE DIDACTICAS
gbADidacticas.Visible = False
End Sub
Private Sub MenuPrincipal_Load(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles MyBase.Load
'ESCRIBE EN LA ETIQUETA DE USUARIO EL NOMBRE DEL USUARIO
LbNombreUsuario.Text = FPassword.nombreusuario.ToString
End Sub
Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click
'MUESTRA EL FORMULARIO DE DIDACTICA
FCreaAdidactica.Show()
Me.Hide()
End Sub
Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button5.Click
'MUESTRA EL FORMULARIO DE JUEGO
FPrincipalJuega.Show()
Me.Hide()
End Sub
```

FORMULARIO ALTAS

```
Imports System.Data.SqlClient
Imports System.Windows.Forms
Public Class Altas
```

```
'DECLARACION DE VARIABLES
Dim oconexion As New Conexion
Dim DataSetTipoUsuario As DataSet
Dim DataSetMateria As DataSet
Dim DataSetListaMateria As DataSet
Dim DataSetTema As DataSet
Dim TemaBuscado As Integer
Dim CM As Integer
Dim indice As Integer
Private Sub Altas_Activated(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Activated
'DESACTIVAR LOS CHECKBOX DE PERMISOS
CheckBox1.Checked = False
CheckBox3.Checked = False
End Sub
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
'DECLARACION DE VARIABLES
Dim comando As New SqlCommand
Dim param As SqlParameter
Dim APaterno As SqlParameter
Dim AMaterno As SqlParameter
Dim cuenta As SqlParameter
Dim password As SqlParameter
Dim parCrea As SqlParameter
Dim parCon As SqlParameter
Dim parPar As SqlParameter
Dim tipoUsu As SqlParameter
Dim bcre As Integer
Dim bcon As Integer
Dim bpar As Integer
Dim porCapturar As Integer
Dim resp As MsgBoxResult
INICIALIZACION DE VARIABLE QUE VERIFICARÁ QUE TODAS LAS VARIABLES
ESTEN CAPTURADAS
porCapturar = 0
'VERIFICA QUE TODAS LAS CASILLAS ESTEN CAPTURADAS
If Len(TextBox1.Text) <= 0 Then porCapturar = 1
If Len(TextBox2.Text) <= 0 Then porCapturar = 1
If Len(TextBox3.Text) <= 0 Then porCapturar = 1
If Len(TextBox4.Text) <= 0 Then porCapturar = 1
If Len(TextBox5.Text) <= 0 Then porCapturar = 1
'CONDICION PARA VERIFICAR QUE TODAS LAS CASILLAS ESTEN
CAPTURADAS
If porCapturar = 1 Then
resp = MsgBox("Debe capturar todos los datos, para guardar el
registro", MsgBoxStyle.OkOnly, "Verificar")
End If
'VERIFICA QUE LOS VALORES ENTREGADOS EN LOS CHECKBOX DE LOS PERMISOS
If resp = MsgBoxResult.Ok Then
Else
bcre = 0
bcon = 0
bpar = 0
If CheckBox1.Checked = True Then
bcre = 1
End If
If CheckBox2.Checked = True Then
bcon = 1
End If
If CheckBox3.Checked = True Then
bpar = 1
End If
'LLAMA A STORED PROCEDURE PARA AGREGAR NVO. USUARIO A BASE DE DATOS
Try
'llama sp
comando.CommandType = CommandType.StoredProcedure
comando.CommandText = "NvoUsuario"
'manda parametro @nombre
param =
comando.Parameters.Add("@nombre"
, SqlDbType.VarChar)
param.Direction = ParameterDirection.Input
param.Value = Trim(TextBox1.Text)
'manda parametro @aPaterno
APaterno = comando.Parameters.Add("@aPaterno", SqlDbType.VarChar)
APaterno.Direction = ParameterDirection.Input
APaterno.Value = Trim(TextBox2.Text)
'manda paramtro @AMaterno
AMaterno = comando.Parameters.Add("@AMaterno", SqlDbType.VarChar)
AMaterno.Direction = ParameterDirection.Input
AMaterno.Value = Trim(TextBox3.Text)
'manda parametro @cuenta
cuenta = comando.Parameters.Add("cuenta", SqlDbType.VarChar)
cuenta.Direction = ParameterDirection.Input
cuenta.Value = Trim(TextBox4.Text)
'manda parametro @password
password = comando.Parameters.Add("password", SqlDbType.VarChar)
password.Direction = ParameterDirection.Input
password.Value = Trim(TextBox5.Text)
'manda parametro @crea
parCrea = comando.Parameters.Add("crea",
SqlDbType.Int)
parCrea.Direction = ParameterDirection.Input
parCrea.Value = bcre
'manda parametro @consulta
```

Modelo de sistema para realizar material didáctico de apoyo a docentes

```

parCon = comando.Parameters.Add("consulta", SqlDbType.Int)
parCon.Direction = ParameterDirection.Input
parCon.Value = bcon
'manda parametro @participa
parPar = comando.Parameters.Add("participa", SqlDbType.Int)
parPar.Direction = ParameterDirection.Input
parPar.Value = bpar
'manda tipoUsuario
tipoUsu = comando.Parameters.Add("tipoUsuario", SqlDbType.Int)
tipoUsu.Direction = ParameterDirection.Input
tipoUsu.Value = ComboBox2.Selected.Value
conexion.ExecuteSP(comando)
MessageBox.Show("Registro guardado")
buscarRegistroUsuario()
Catch A As Exception
    MessageBox.Show(A.Message)
End Try
End If
End Sub
Private Sub Altas_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    'LA VARIABLE DE TIPO DE USUARIO ES LLAMADA DESDE PARA SER MOSTRADA EN ESTE FORMULARIO
    LbNombreUsuario.Text = FPassword.nombrequesuario.ToString
    Dim cadena As String
    'LLENA COMBOBOX DE TIPO DE USUARIO DE ACUERDO AL CATALOGO DE USUARIOS
    Try
        DataSetTipoUsuario = conexion.GetRows("select * from tipoUsuario")
        ComboBox2.DataSource = DataSetTipoUsuario.Tables(0)
        ComboBox2.DisplayMember = "Tipus_Descripcion"
        ComboBox2.ValueMember = "Tipus_tipo"
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try
    'SELECCIONA TODAS LAS MATERIAS REGISTRADAS EN BASE PARA SER MOSTRADAS EN 'COMBO DE MATERIAS
    Try
        cadena = "select * from materia "
        DataSetListaMateria = conexion.GetRows(cadena)
        ComboBox1.DataSource = DataSetListaMateria.Tables(0)
        ComboBox1.DisplayMember = "mat_nombre"
        ComboBox1.ValueMember = "mat_clave"
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try
End Sub
Private Sub TIPOUSUARIOBindingNavigatorSaveItem_Click_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TIPOUSUARIOBindingNavigatorSaveItem.Click
    'BLINDING NAVIGATOR DE TIPO DE USUARIO
    Me.Validate()
    Me.TIPOUSUARIOBindingSource.EndEdit()
End Sub
Me.TIPOUSUARIOTableAdapter.Update(Me.AprenderDataSet.TIPOUSUARIO)
End Sub
Private Sub ComboBox2_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ComboBox2.SelectedIndexChanged
    'OBTIENE EL INDICE CADA VEZ QUE SE ELIGE UNA OPCION EN COMBO DE TIPO DE USUARIO
    ComboBox2.DataSource.GetType()
End Sub
Private Sub BBuscar_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BBuscar.Click
    'LLAMA LA FUNCION DE BUSCAR USUARIO
    buscarRegistroUsuario()
End Sub
Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button4.Click
    'DECLARACION DE VARIABLES
    Dim comando As New SqlCommand
    Dim param As SqlParameter
    Dim APaterno As SqlParameter
    Dim AMaterno As SqlParameter
    Dim cuenta As SqlParameter
    Dim password As SqlParameter
    Dim parCrea As SqlParameter
    Dim parCon As SqlParameter
    Dim parPar As SqlParameter
    Dim tipoUsu As SqlParameter
    Dim bcre As Integer
    Dim bcon As Integer
    Dim bpar As Integer
    Dim porCapturar As Integer
    Dim respuesta As MsgBoxResult
    'INICIALIZACION DE VARIABLES PARA SIGUIENTES PROCESOS
    porCapturar = 0
    bcre = 0
    bcon = 0
    bpar = 0
    'VERIFICA QUE TODAS LAS CASILLAS ESTEN CAPTURADAS
    If Len(TextBox1.Text) <= 0 Then porCapturar = 1
    If Len(TextBox2.Text) <= 0 Then porCapturar = 1
    If Len(TextBox3.Text) <= 0 Then porCapturar = 1
    If Len(TextBox4.Text) <= 0 Then porCapturar = 1
    If Len(TextBox5.Text) <= 0 Then porCapturar = 1
    'SI FALTAN DATOS POR CAPTURAR, MANDA MENSAJE
    If porCapturar = 1 Then
        MsgBox("Falta por capturar datos personales", MsgBoxStyle.OkOnly, "Verificar")
    Else
        If CheckBox1.Checked = True Then
            bcre = 1
        End If
        If CheckBox2.Checked = True Then
            bcon = 1
        End If
        If CheckBox3.Checked = True Then
            bpar = 1
        End If
    End If
End If
End Sub
End If
' SI NO FALTAN DATOS POR CAPTURAR, ENTONCES PREGUNTA SI DESEA MODIFICAR
respuesta = MsgBox("¿Esta seguro que desea modificar el registro?", MsgBoxStyle.OkCancel, "Verifica Modificacion")
If respuesta = MsgBoxResult.Ok Then
    'MODIFICAR EL REGISTRO
    Try
        'llama sp
        comando.CommandType = CommandType.StoredProcedure
        comando.CommandText = "ModificaRegistro"
        'manda parametro @nombre
        param = comando.Parameters.Add("@nombre", SqlDbType.VarChar)
        param.Direction = ParameterDirection.Input
        param.Value = Trim(TextBox1.Text)
        'manda parametro @aPaterno
        APaterno = comando.Parameters.Add("@aPaterno", SqlDbType.VarChar)
        APaterno.Direction = ParameterDirection.Input
        APaterno.Value = Trim(TextBox2.Text)
        'manda parametro @AMaterno
        AMaterno = comando.Parameters.Add("@AMaterno", SqlDbType.VarChar)
        AMaterno.Direction = ParameterDirection.Input
        AMaterno.Value = Trim(TextBox3.Text)
        'manda parametro @cuenta
        cuenta = comando.Parameters.Add("cuenta", SqlDbType.VarChar)
        cuenta.Direction = ParameterDirection.Input
        cuenta.Value = Trim(TextBox4.Text)
        'manda parametro @password
        password = comando.Parameters.Add("password", SqlDbType.VarChar)
        password.Direction = ParameterDirection.Input
        password.Value = Trim(TextBox5.Text)
        'manda parametro @crea
        parCrea = comando.Parameters.Add("crea", SqlDbType.Int)
        parCrea.Direction = ParameterDirection.Input
        parCrea.Value = bcre
        'manda parametro @consulta
        parCon = comando.Parameters.Add("@consulta", SqlDbType.Int)
        parCon.Direction = ParameterDirection.Input
        parCon.Value = bcon
        'manda parametro @participa
        parPar = comando.Parameters.Add("@participa", SqlDbType.Int)
        parPar.Direction = ParameterDirection.Input
        parPar.Value = bpar
        'manda tipoUsuario
        tipoUsu = comando.Parameters.Add("tipoUsuario", SqlDbType.Int)
        tipoUsu.Direction = ParameterDirection.Input
        tipoUsu.Value = ComboBox2.SelectedValue
        conexion.ExecuteSP(comando)
    Catch A As Exception
        MessageBox.Show(A.Message)
    End Try
    MsgBox("Registro Modificado ", MsgBoxStyle.OkOnly, "Cuadro resumen")
Else
    MsgBox("Registro NO Modificado ", MsgBoxStyle.OkOnly, "Cuadro resumen")
End If
End If
End Sub
Public Function buscarRegistroUsuario()
    'DECLARACION DE VARIABLES
    Dim Resultados As DataSet
    Dim a As DataSet
    Dim tabla As DataTable
    Dim cadena As String
    'LLAMA A LA FUNCION DE LIMPIAR PANTALLA
    LimpiarPantalla()
    'VERIFICA QUE LOS DATOS DEL USUARIO ESTEN CAPTURADOS PARA POSTERIORMENTE BUSCARLO EN DB
    If (Len(TextBox1.Text) > 0 And Len(TextBox2.Text) > 0 And Len(TextBox3.Text) > 0) Or (Len(TextBox4.Text) > 0) Then
        cadena = "select * from usuario where usu_cuenta = " & """" & Trim(TextBox4.Text.ToString) & """" & " and usu_activo= 1"
        Resultados = conexion.SelectRows(a, conexion.sqlCnn.ToString, cadena)
        'SI EL USUARIO EXISTE, MUESTRA LOS DATOS EN PANTALLA
        If Resultados.Tables.Count > 0 Then
            tabla = Resultados.Tables(0)
            If tabla.Rows.Count > 0 Then
                TextBox1.Text = tabla.Rows(0)("USU_nombre")
                TextBox2.Text = tabla.Rows(0)("USU_aPaterno")
                TextBox3.Text = tabla.Rows(0)("USU_aMaterno")
                TextBox4.Text = tabla.Rows(0)("USU_cuenta")
                TextBox5.Text = tabla.Rows(0)("USU_password")
                If (tabla.Rows(0)("usu_crea") = 1) Then
                    CheckBox1.Checked = True Else CheckBox1.Checked = False
                    If (tabla.Rows(0)("usu_participa") = 1) Then
                        CheckBox3.Checked = True Else CheckBox3.Checked = False
                        ComboBox2.SelectedValue = tabla.Rows(0)("USU_tipoUsuario")
                        Button4.Visible = True
                        Button1.Visible = False
                        Button4.Visible = False
                    End If
                End If
            End If
        End If
        Mensaje
    Else
        MessageBox.Show("No se encontró el registro")
        Button1.Visible = True
        Button4.Visible = False
    End If
End If
End Function
Public Function limpiarPantalla()
    'FUNCION PARA LIMPIAR LA PANTALLA
    TextBox1.Text = ""
    TextBox2.Text = ""
    TextBox3.Text = ""
    TextBox5.Text = ""
    CheckBox1.Checked = False
    CheckBox3.Checked = False
End Function

```

Modelo de sistema para realizar material didáctico de apoyo a docentes

```

End Function
Private Sub Button2_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Button2.Click
    'LIMPIA PANTALLA
    TextBox4.Text = ""
    limpiarPantalla()
    TextBox4.Focus()
End Sub
Private Sub Button9_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)
    'DECLARACION DE VARIABLES
    Dim cadena As String
    Dim Resultados As DataSet
    Dim a As DataSet
    Dim tabla As DataTable
    'LLENA EL DATAGRIDVIEW DE LAS MATERIAS
    cadena = "select * from materia order by mat_clave desc"
    Resultados = oconexion.SelectRows(a, oconexion.sqlConn.ToString,
cadena)
    If Resultados.Tables.Count > 0 Then
        tabla = Resultados.Tables(0)
    End If
    TextBox7.Text = ""
    TextBox7.Focus()
End Sub
Private Sub DataGridView1_CellContentClick(ByVal sender As
System.Object, ByVal e As
System.Windows.Forms.DataGridViewCellEventArgs)
    'EL NOMBRE DE LA MATERIA ELEGIDA APARECERÁ EN EL TEXTBOX DE MATERIA
    TextBox7.Text =
    DataSetMateria.Tables(0).Rows(e.RowIndex)("materia")
End Sub
Private Sub Button6_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)
    Dim comando As New SqlCommand
    Dim param As SqlParameter
    Dim materia As SqlParameter
    Dim AMaterno As SqlParameter
    Dim resp As MsgBoxResult
    Dim porcapturar As Integer
    porcapturar = 0
    'VERIFICA QUE TODAS LAS CASILLAS ESTEN CAPTURADAS
    If Len(TextBox7.Text) <= 0 Then porcapturar = 1
    If porcapturar = 1 Then
        resp = MsgBox("Debe capturar todos los datos, para guardar el
registro", MsgBoxStyle.OkOnly, "Verificar")
    End If
    If resp = MsgBoxResult.Ok Then
        'AGREGA NUEVA MATERIA
    Else
        Try
            'llama sp
            comando.CommandType = CommandType.StoredProcedure
            comando.CommandText = "NvoMateria"
            'manda parametro @nombre
            param = comando.Parameters.Add("@nombre", SqlDbType.VarChar)
            param.Direction = ParameterDirection.Input
            param.Value = Trim(TextBox7.Text)
            oconexion.ExecuteSP(comando)
            MsgBox.Show("Registro guardado")
        Catch A As Exception
            MsgBox.Show(A.Message)
        End Try
    End If
End Sub
Private Sub Button3_Click_1(ByVal sender As System.Object, ByVal e
As System.EventArgs)
    Dim Resultados As DataSet
    Dim a As DataSet
    Dim tabla As DataTable
    Dim cadena As String
    'LLENA DATAGRIDVIEW CON DATOS DE MATERIA
    Try
        cadena = "select mat_clave as CLAVE, mat_nombre as
MATERIA from materia where " & "mat_nombre like '%" &
Trim(TextBox8.Text.ToString) & "%'"
        DataSetMateria = oconexion.GetRows(cadena)
        DataGridView1.DataSource = DataSetMateria.Tables(0)
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try
End Sub
Private Sub Button7_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)
    Dim comando As New SqlCommand
    Dim clave As SqlParameter
    Dim nombre As SqlParameter
    Dim porCapturar As Integer
    Dim respuesta As MsgBoxResult
    porCapturar = 0
    'VERIFICA QUE TODAS LAS CASILLAS ESTEN CAPTURADAS
    If Len(TextBox7.Text) <= 0 Then porCapturar = 1
    'If Len(TextBox6.Text) <= 0 Then porCapturar = 1
    If porCapturar = 1 Then
        MsgBox("Falta por capturar datos", MsgBoxStyle.OkOnly, "Verificar")
    Else
        Respuesta = MsgBox("¿Esta seguro que desea modificar el registro?",
MsgBoxStyle.OkCancel, "Verifica Modificacion")
        If respuesta = MsgBoxResult.Ok Then
            Try
                'llama sp
                comando.CommandType =
CommandType.StoredProcedure
                comando.CommandText = "SPModificaMateria"
                'manda parametro @clave
                clave = comando.Parameters.Add("@clave", SqlDbType.Int)
                clave.Direction = ParameterDirection.Input
                clave.Value = CM
                'manda parametro @nombre
                nombre = comando.Parameters.Add("@nombre", SqlDbType.VarChar)
                nombre.Direction = ParameterDirection.Input
                nombre.Value = Trim(TextBox7.Text)
                oconexion.ExecuteSP(comando)
            Catch A As Exception
                MsgBox.Show(A.Message)
            End Try
        End If
    End If
End Sub
Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button5.Click
    Dim Resultados As DataSet
    Dim a As DataSet
    Dim tabla As DataTable
    Dim cadena As String
    'LLENA DATAGRIDVIWE CON LOS TEMAS EXISTENTES
    Try
        cadena = "select TEM_claveint as CLAVE, TEM_nombre as TEMA
, TEM_descripcion as DESCRIPCION, mat_nombre as MATERIA from Tema
inner join materia on tema.mat_claveint = materia.mat_clave where
TEM_nombre like '%" & Trim(TextBox10.Text.ToString) & "%'"
        DataSetTema = oconexion.GetRows(cadena)
        DataGridView2.DataSource = DataSetTema.Tables(0)
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try
End Sub
Private Sub Button10_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button10.Click
    Dim comando As New SqlCommand
    Dim nombre As SqlParameter
    Dim descripcion As SqlParameter
    Dim claveMateria As SqlParameter
    Dim resp As MsgBoxResult
    Dim porcapturar As Integer
    porcapturar = 0
    'VERIFICA QUE TODAS LAS CASILLAS ESTEN CAPTURADAS
    If Len(TextBox10.Text) <= 0 Then porcapturar = 1
    If Len(RichTextBox1.Text) <= 0 Then porcapturar = 1
    If Len(ComboBox1.Text) <= 0 Then porcapturar = 1
    'SI FALTAN DATOS POR CAPTURAR MANDA MENSAJE
    If porcapturar = 1 Then
        resp = MsgBox("Debe capturar todos los datos, para guardar el
registro", MsgBoxStyle.OkOnly, "Verificar")
    End If
    If resp = MsgBoxResult.Ok Then
        Try
            'llama sp
            comando.CommandType = CommandType.StoredProcedure
            comando.CommandText = "NvoTema"
            'manda parametro @nombre
            nombre = comando.Parameters.Add("@nombre", SqlDbType.VarChar)
            nombre.Direction = ParameterDirection.Input
            nombre.Value = Trim(TextBox10.Text)
            'manda parametro @descripcion
            descripcion = comando.Parameters.Add("@descripcion",
SqlDbType.VarChar)
            descripcion.Direction = ParameterDirection.Input
            descripcion.Value = Trim(RichTextBox1.Text)
            'manda parametro @materia
            claveMateria = comando.Parameters.Add("@materia", SqlDbType.Int)
            claveMateria.Direction = ParameterDirection.Input
            claveMateria.Value = Trim(ComboBox1.SelectedValue)
            oconexion.ExecuteSP(comando)
            MsgBox.Show("Registro guardado")
        Catch A As Exception
            MsgBox.Show(A.Message)
        End Try
    End If
End Sub
Private Sub DataGridView1_CellMouseClick(ByVal sender As Object,
ByVal e As System.Windows.Forms.DataGridViewCellMouseEventArgs)
    'OBTIENE LA MATERIA Y LA DESPLIEGA EN EL TEXT DE MATERIA
    TextBox7.Text =
    DataSetMateria.Tables(0).Rows(e.RowIndex)("materia")
End Sub
Private Sub DataGridView2_CellMouseClick(ByVal sender As Object,
ByVal e As System.Windows.Forms.DataGridViewCellMouseEventArgs)
    Handles DataGridView2.CellMouseClick
    'AL ELEGIR UN TEMA EN, SE DESPLIEGARAN LOS DATOS DEL MISMO EN LAS
CASILLAS CORRESPONDIENTES DEL APARTADO DE TEMA PARA QUE LOS VEA EL
USUARIO
    TextBox10.Text = DataSetTema.Tables(0).Rows(e.RowIndex)("TEMA")
    RichTextBox1.Text =
    DataSetTema.Tables(0).Rows(e.RowIndex)("DESCRIPCION")
    ComboBox1.Text = DataSetTema.Tables(0).Rows(e.RowIndex)("MATERIA")
    Temabuscado = DataSetTema.Tables(0).Rows(e.RowIndex)("CLAVE")
End Sub
Private Sub Button11_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button11.Click
    Dim comando As New SqlCommand
    Dim descripcion As SqlParameter
    Dim nombre As SqlParameter
    Dim materia As SqlParameter
    Dim claveTema As SqlParameter
    Dim respuesta As MsgBoxResult
    Dim porCapturar As Integer
    porCapturar = 0
    'VERIFICA QUE TODAS LAS CASILLAS ESTEN CAPTURADAS
    If Len(TextBox10.Text) <= 0 Then porCapturar = 1
    If Len(RichTextBox1.Text) <= 0 Then porCapturar = 1
    If Len(ComboBox1.Text) <= 0 Then porCapturar = 1
    If porCapturar = 1 Then
        MsgBox("Falta por capturar datos", MsgBoxStyle.OkOnly, "Verificar")
    Else
        respuesta = MsgBox("¿Esta seguro que desea modificar el registro?",
MsgBoxStyle.OkCancel, "Verifica Modificacion")
        If respuesta = MsgBoxResult.Ok Then
            Try
                'llama sp

```

Modelo de sistema para realizar material didáctico de apoyo a docentes

```

        comando.CommandType =
CommandType.StoredProcedure
        comando.CommandText = "ModificaTema"
        'manda parametro @nombre
nombre = comando.Parameters.Add("@nombre", SqlDbType.VarChar)
        nombre.Direction = ParameterDirection.Input
        nombre.Value = Trim(TextBox10.Text)
        'manda parametro @descripcion
descripcion = comando.Parameters.Add("@descripcion",
SqlDbType.VarChar)
        descripcion.Direction =
ParameterDirection.Input
        descripcion.Value = Trim(RichTextBox1.Text)
        'manda materia
materia = comando.Parameters.Add("@materia", SqlDbType.Int)
        materia.Direction = ParameterDirection.Input
        materia.Value = ComboBox1.SelectedValue
        'manda tema modificado
claveTema = comando.Parameters.Add("@claveTema", SqlDbType.Int)
        claveTema.Direction = ParameterDirection.Input
        claveTema.Value = TemaBuscado
        oconexion.Executesp(comando)
        Catch A As Exception
            MessageBox.Show(A.Message)
        End Try
MsgBox("Registro Modificado ", MsgBoxStyle.OkOnly, "Cuadro
resumen")
Else
MsgBox("Registro NO Modificado ", MsgBoxStyle.OkOnly, "Cuadro
resumen")
End If
End If
End Sub
Private Sub ComboBox1_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles ComboBox1.Click
    Dim cadena As String
    'DESPLIEGA LOS TEMAS EN EL COMBO DE MATERIA PARA QUE EL USUARIO
VISUALICE LAS MATERIAS EXISTENTES EN DB
    Try
        cadena = "select * from materia "
        DataSetListaMateria = oconexion.GetRows(cadena)
        ComboBox1.DataSource = DataSetListaMateria.Tables(0)
        ComboBox1.DisplayMember = "mat_nombre"
        ComboBox1.ValueMember = "mat_clave"
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try
End Sub
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
    Dim Resultados As DataSet
    Dim a As DataSet
    Dim tabla As DataTable
    Dim cadena As String
    'BUSCA LA MATERIA QUE EL USUARIO HAYA INGRESADO
    Try
        cadena = "select mat_clave as CLAVE, mat_nombre as MATERIA from
materia where " & "mat_nombre like '%" &
Trim(TextBox8.Text.ToString) & "%"
        DataSetMateria = oconexion.GetRows(cadena)
        DataGridView1.DataSource = DataSetMateria.Tables(0)
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try
End Sub
PrivateSub Button6_Click_1(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button6.Click
    Dim comando As New SqlCommand
    Dim materia As SqlParameter
    Dim mmaterno As SqlParameter
    Dim resp As MsgBoxResult
    Dim porcapturar As Integer
    Dim cadena As String
    'VERIFICA SI EXISTE SI EL REGISTRO EXISTE Y SI ES ASI LO MUESTRA
AL USUARIO
    Try
        cadena = "select mat_clave as CLAVE, mat_nombre as MATERIA from
materia where " & "mat_nombre = " & Trim(TextBox12.Text.ToString)
        DataSetMateria = oconexion.GetRows(cadena)
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try
    porcapturar = 0
    'VERIFICA QUE TODAS LAS CASILLAS ESTEN CAPTURADAS
    If Len(TextBox12.Text) <= 0 Then porcapturar = 1
    If porcapturar = 1 Then
        resp = MsgBox("Debe capturar el datos, para guardar el registro",
MsgBoxStyle.OkOnly, "Verificar")
    End If
    If resp = MsgBoxResult.Ok Then
        Else
            Try
                'llama sp de nueva materia
                comando.CommandType = CommandType.StoredProcedure
                comando.CommandText = "NvoMateria"
                'manda parametro @nombre
                param = comando.Parameters.Add("@nombre",
SqlDbType.VarChar)
                param.Direction = ParameterDirection.Input
                param.Value = Trim(TextBox12.Text)
                oconexion.Executesp(comando)
                MessageBox.Show("Registro guardado")
                ' buscarRegistroUsuario()
            Catch A As Exception
                MessageBox.Show(A.Message)
            End Try
        End If
    End Sub
    Private Sub Button7_Click_1(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button7.Click
        Dim comando As New SqlCommand
        Dim clave As SqlParameter
        Dim nombre As SqlParameter
        Dim porCapturar As Integer
        Dim respuesta As MsgBoxResult
        Dim VALORANTERIOR As String

        VALORANTERIOR =
DataSetMateria.Tables(0).Rows(indice) ("materia")

        porCapturar = 0
        'VERIFICA QUE TODAS LAS CASILLAS ESTEN CAPTURADAS
        If Len(TextBox7.Text) <= 0 Then porCapturar = 1
        If porCapturar = 1 Then
            MsgBox("Falta por capturar datos", MsgBoxStyle.OkOnly,
"Verificar")
        Else
            respuesta = MsgBox("¿Esta seguro que desea modificar la
materia " + VALORANTERIOR + "?", MsgBoxStyle.OkCancel, "Verifica
Modificacion")
            If respuesta = MsgBoxResult.Ok Then
                Try
                    'llama sp para modificar materia
                    comando.CommandType =
CommandType.StoredProcedure
                    comando.CommandText = "SPModificaMateria"
                    'manda parametro @clave
                    clave = comando.Parameters.Add("@clave",
SqlDbType.Int)
                    clave.Direction = ParameterDirection.Input
                    clave.Value = CM
                    'manda parametro @nombre
                    nombre = comando.Parameters.Add("@nombre",
SqlDbType.VarChar)
                    nombre.Direction = ParameterDirection.Input
                    nombre.Value = Trim(TextBox7.Text)

                    oconexion.Executesp(comando)
                Catch A As Exception
                    MessageBox.Show(A.Message)
                End Try
            End Try
            MsgBox("Registro Modificado ", MsgBoxStyle.OkOnly,
"Cuadro resumen")
        Else
            MsgBox("Registro NO Modificado ",
MsgBoxStyle.OkOnly, "Cuadro resumen")
        End If
    End If
    TextBox7.Text = ""

    Dim Resultados As DataSet
    Dim tabla As DataTable
    Dim cadena As String
    'ACTUALIZA EL DATAGRIDVIEW DE MATERIA
    Try
        cadena = "select mat_clave as CLAVE, mat_nombre as
MATERIA from materia where " & "mat_nombre like '%" &
Trim(TextBox8.Text.ToString) & "%"
        DataSetMateria = oconexion.GetRows(cadena)
        DataGridView1.DataSource = DataSetMateria.Tables(0)
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try
End Sub
Private Sub DataGridView1_CellContentClick_1(ByVal sender As
System.Object, ByVal e As
System.Windows.Forms.DataGridViewCellEventArgs) Handles
DataGridView1.CellContentClick
    'AL ELEGIR UNA MATERIA DEL DATAGRIDVIEW DE MATERIA, ESTA SE MUESTRA
EN EL TEXTBOX MATERIA
    CM = DataSetMateria.Tables(0).Rows(e.RowIndex) ("CLAVE")
    TextBox7.Text =
DataSetMateria.Tables(0).Rows(e.RowIndex) ("materia")
    GroupBox8.Visible = True
    indice = e.RowIndex
End Sub
Private Sub Button8_Click_2(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button8.Click
    'MUESTRA EL MENU PRINCIPAL
    MenuPrincipal.Show()
    Me.Hide()
End Sub
End Class
FORMULARIO CREA DIDACTICA
Imports System.Data.SqlClient
Imports System.Windows.Forms
Imports System.IO
Public Class FcreaAdidactica
    'DECLARACION DE VARIABLES GLOBALES
    Dim oconexion As New Conexion
    Dim claveDinamica As Integer
    Dim DataSetTema As DataSet
    Dim cuentaPregunta As Integer
    Dim numActividad As Integer
    Private Sub FcreaAdidactica_Load(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles MyBase.Load
        'ASIGNACION DE VARIABLE NOMBREUSUARIO A LA ETIQUETA DE
USUARIO DE ESTE FORMULARIO
        lblUsuario.Text = FPassword.nombreusuario.ToString
    End Sub

```

```

'DECLARACION DE VARIABLES
Dim Resultados As DataSet
Dim a As DataSet
Dim tabla As DataTable
Dim cadena As String
Dim MateriaClave As String
Dim DSMateria As DataSet
cuentaPregunta = 0
Dim DataSetListaMateria As DataSet

'LEENA COMBOBOX DE MATERIA
Try
    cadena = "select * from materia "
    DataSetListaMateria = oconexion.GetRows(cadena)
    CbMateria.DisplayMember = "mat_nombre"
    CbMateria.ValueMember = "mat_clave"
    CbMateria.DataSource = DataSetListaMateria.Tables(0)
Catch ex As Exception
    MsgBox(ex.Message)
End Try

'OBTIENE EL VALOR QUE CONTIENE EL COMBO DE MATERIA
MateriaClave = CbMateria.SelectedValue.ToString()
'LEENA EL COMBOBOX DE TEMA DE ACUERDO CON LA MATERIA QUE SE
ESTE MOSTRANDO
Try
    cadena = "select * from tema where mat_claveint = " &
MateriaClave.ToString
    DataSetTema = oconexion.GetRows(cadena)
    CbTema.DisplayMember = "tem_nombre"
    CbTema.ValueMember = "tem_claveint"
    CbTema.DataSource = DataSetTema.Tables(0)
Catch ex As Exception
    MsgBox(ex.Message)
End Try

'OBTIENE LOS DATOS DEL AUTOR DESDE LA DB
cadena = "Select * from datosAutor where
datosAutor.usu_claveint = " & FPassword.claveusuario
oconexion.SqlConn.ToString, cadena)
Resultados = oconexion.SelectRows(a,
oconexion.SqlConn.ToString, cadena)
If Resultados.Tables.Count > 0 Then
    tabla = Resultados.Tables(0)
    If tabla.Rows.Count > 0 Then
        RTResenia.Text = tabla.Rows(0)("AUT_resenia")
        btnModificar.Visible = True
    Else
        btnGuardar.Visible = True
    End If
End If
End Sub

Private Sub btnGuardar_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnGuardar.Click
'DECLARACION DE VARIABLES
Dim comando As New SqlCommand
Dim claveUsu As SqlParameter
Dim reseniaAutor As SqlParameter
Dim porCapturar As Integer

porCapturar = 0

If (Len(RTResenia.Text) > 5) Then
    porCapturar = 1
Else
    porCapturar = 0
End If

If porCapturar = 1 Then
    'llama stored procedure para agregar nuevo autor
    Try
        'llama sp nuevo autor
        comando.CommandType = CommandType.StoredProcedure
        comando.CommandText = "SPNvoAutor"

        'manda parametro @clave del usuario
        claveUsu = comando.Parameters.Add("@claveUsuario",
SqlDbType.Int)
        claveUsu.Direction = ParameterDirection.Input
        claveUsu.Value = FPassword.claveusuario

        'manda parametro @reseniaAutor
        reseniaAutor =
comando.Parameters.Add("@reseniaAutor", SqlDbType.VarChar)
        reseniaAutor.Direction = ParameterDirection.Input
        reseniaAutor.Value = Trim(RTResenia.Text)

        oconexion.ExecuteNonQuery(comando)
        MsgBox.Show("Registro guardado")
    Catch A As Exception
        MsgBox.Show(A.Message)
    End Try
End If
End Sub

Private Sub btnModificar_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnModificar.Click
'DECLARACION DE VARIABLES
Dim comando As New SqlCommand
Dim claveUsu As SqlParameter
Dim reseniaAutor As SqlParameter
Dim porCapturar As Integer
'INICIALIZACION DE VARIABLE porCapturar
porCapturar = 0
'VERIFICACION DE LOS CAMPOS QUE SE DEBEN CAPTURAR
If (Len(RTResenia.Text) > 5) Then
    porCapturar = 1
Else
    porCapturar = 0
End If
'SI TODOS LOS CAMPOS ESTAN CAPTURADOS SE LLAMA AL
PROCEDIMIENTO DE MODIFICACION DE AUTOR

If porCapturar = 1 Then
    Try
        'llama sp
        comando.CommandType = CommandType.StoredProcedure
        comando.CommandText = "SPModificaReseniaAutor"

        'manda parametro @clave del usuario
        claveUsu = comando.Parameters.Add("@claveUsuario",
SqlDbType.Int)
        claveUsu.Direction = ParameterDirection.Input
        claveUsu.Value = FPassword.claveusuario

        'manda parametro @reseniaAutor
        reseniaAutor =
comando.Parameters.Add("@reseniaAutor", SqlDbType.VarChar)
        reseniaAutor.Direction = ParameterDirection.Input
        reseniaAutor.Value = Trim(RTResenia.Text)

        oconexion.ExecuteNonQuery(comando)
        MsgBox.Show("Registro modificado")
    Catch A As Exception
        MsgBox.Show(A.Message)
    End Try
End If
End Sub

Private Sub BtnExaminar_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles BtnExaminar.Click
'DECLARACION DE VARIABLES
Dim ruta As String
Dim openFileDialog1 As New OpenFileDialog()
Dim valorMateria As Integer
Dim valorTema As Integer
'INICIALIZACION DE LA VARIABLE valormateria CON EL VALOR
SELECCIONADO DEL COMBO DE MATERIA
valorMateria = CbMateria.SelectedValue
'INICIALIZACION DE VARIABLE valortema CON EL VALOR DEL
COMBO DE TEMA
valorTema = CbTema.SelectedValue
'ABRIR EL DIALOGO CON LAS EXTENSIONES PREDETERMINADAS DE
BMP, JPG, GIF, JPEG, WMF, PNG
openFileDialog1.Filter = "Images
(*.BMP;*.JPG;*.GIF;*.JPEG;*.WMF;*.PNG)|*.BMP;*.JPG;*.GIF;*.JPEG;*.W
MF;*.PNG|All files (*.*)|*.*"

' Allow the user to select multiple images.
openFileDialog1.Multiselect = True
openFileDialog1.Title = "Selector de Imagen"

'ABRA OPENDIALOG PARA OBTENER LA RUTA DE IMAGEN
' Y LA ESCRIBE EN EL COMBO MATERIA
If openFileDialog1.ShowDialog() =
Windows.Forms.DialogResult.OK Then 'DialogResult.OK Then
    If openFileDialog1.FileName <> "" Then
        ruta = openFileDialog1.FileName.ToString
        TBRuta.Text = ruta
        CbMateria.SelectedValue = valorMateria
        CbTema.SelectedValue = valorTema
    End If
End If

End Sub

Private Sub BtnAgregar_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles BtnAgregar.Click
Dim comando As New SqlCommand
Dim dnmClave As SqlParameter
Dim pregunta As SqlParameter
Dim respuesta As SqlParameter
Dim cadena As String
Dim DataSetDinamica As DataSet
'SELECCIONA UN REGISTRO DE DINAMICA
Try
    cadena = "select top 1 dnm_claveint, dnm_actividad from
dinamica order by dnm_claveint DESC"
    DataSetDinamica = oconexion.GetRows(cadena)
    claveDinamica =
DataSetDinamica.Tables(0).Rows(0)("dnm_claveint")
Catch ex As Exception
    MsgBox(ex.Message)
End Try
'CUENTA EL NUMERO DE PREGUNTAS PARA VERIFICAR QUE SOLO
INGRESE EL NUMERO CAPTURADO INICIALMENTE
If cuentaPregunta < CbNoPreguntas.Text Then
    Try
        'llama sp "SPCuestionarioInserta"
        comando.CommandType = CommandType.StoredProcedure
        comando.CommandText = "SPCuestionarioInserta"
        'manda parametro clave de dinamica
        dnmClave = comando.Parameters.Add("@dnm_claveint",
SqlDbType.Int)
        dnmClave.Direction = ParameterDirection.Input
        dnmClave.Value = claveDinamica
        'manda parametro pregunta
        pregunta = comando.Parameters.Add("@pregunta",
SqlDbType.VarChar)
        pregunta.Direction = ParameterDirection.Input
        pregunta.Value = "P.- " & TbPregunta.Text
        'manda parametro respuesta
        respuesta = comando.Parameters.Add("@respuesta",
SqlDbType.VarChar)
        respuesta.Direction = ParameterDirection.Input
        respuesta.Value = "R.- " & TbRespuesta.Text
        oconexion.ExecuteNonQuery(comando)
        MsgBox.Show("Pregunta registrada")
        ListBox1.Items.Add(TbPregunta.Text)
        ListBox2.Items.Add(TbRespuesta.Text)
        cuentaPregunta = cuentaPregunta + 1
    Catch Ex As Exception
        MsgBox.Show(Ex.Message)
    End Try
End Try

```

```

Else
    MessageBox.Show("Limite de preguntas")
End If
'LIMPIA TEXTBOX DE PREGNTA Y RESPUESTA
TbPregunta.Clear()
TbRespuesta.Clear()
TbPregunta.Focus()
'AL LLEGAR AL LIMITE DE PREGUNTAS SE MUESTRA UN MENSAJE
If cuentaPregunta = CbNoPreguntas.Text Then
    MessageBox.Show("Limite de preguntas")
End If

End Sub

Private Sub ListBox1_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ListBox1.SelectedIndexChanged
    'AL SELECCIONAR UN ITEM DE LISTBOX DE PREGUNTAS, TAMBIEN SE SELECCIONA EL ITEM DE RESPUESTAS
    ListBox2.SelectedIndex = ListBox1.SelectedIndex
End Sub

Private Sub ListBox2_MouseDown(ByVal sender As Object, ByVal e As System.Windows.Forms.MouseEventArgs) Handles ListBox2.MouseDown
    'AL SELECCIONAR UN ITEM DE RESPUESTA , TAMBIEN SE SELECCIONARA UN ITEM DE PREGUNTAS
    ListBox1.SelectedIndex = ListBox2.SelectedIndex
End Sub

Private Sub ListBox2_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ListBox2.SelectedIndexChanged
    'AL SELECCIONAR UN ITEM DE LISTBOX DE PREGUNTAS, TAMBIEN SE SELECCIONA EL ITEM DE RESPUESTAS
    ListBox1.SelectedIndex = ListBox2.SelectedIndex
End Sub

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click
    Dim comando As New SqlCommand
    Dim dnmClave As SqlParameter
    Dim pregunta As SqlParameter
    Dim respuesta As SqlParameter
    Dim indiceSeleccionado, i, totalPreg As Integer
    Dim arregloPreguntas(20) As String
    Dim ArregloRespuestas(20) As String

    Try
        'LLAMA SP PARA ELIMINAR PREGUNTAS
        comando.CommandType = CommandType.StoredProcedure
        comando.CommandText = "SPEliminaPregunta"

        'manda parametro de clave de dinamica
        dnmClave = comando.Parameters.Add("@dinamicaClave", SqlDbType.Int)
        dnmClave.Direction = ParameterDirection.Input
        dnmClave.Value = claveDinamica

        'manda parametro pregunta
        pregunta = comando.Parameters.Add("@pregunta", SqlDbType.VarChar)
        pregunta.Direction = ParameterDirection.Input
        pregunta.Value = ListBox1.Text

        'manda parametro respuesta
        respuesta = comando.Parameters.Add("@respuesta", SqlDbType.VarChar)
        respuesta.Direction = ParameterDirection.Input
        respuesta.Value = ListBox2.Text

        oconexion.ExecuteNonQuery(comando)
        MessageBox.Show("Pregunta eliminada")
        cuentaPregunta = cuentaPregunta - 1

    Catch ex As Exception
        MessageBox.Show(ex.Message)
    End Try

    'obtiene el indice la pregunta a eliminar
    indiceSeleccionado = ListBox1.SelectedIndex
    'obtiene el numero total de preguntas
    totalPreg = ListBox1.Items.Count
    'ingresa en un arreglo las preguntas y respuestas
    For i = 0 To totalPreg - 1
        arregloPreguntas(i) = ListBox1.Items(i).ToString
        ArregloRespuestas(i) = ListBox2.Items(i).ToString
    Next
    'elimina en la lista de preguntas y respuestas los elegidos para eliminar por el usuario
    ListBox1.Items.Clear()
    ListBox2.Items.Clear()

    'llena los listbox de preguntas y respuestas con las preguntas y respuestas que no se eliminaron
    For i = 0 To totalPreg - 1
        If indiceSeleccionado <> i Then
            ListBox1.Items.Add(arregloPreguntas(i))
            ListBox2.Items.Add(ArregloRespuestas(i))
        End If
    Next

End Sub

Private Sub Button1_Click_2(ByVal sender As System.Object, ByVal e As System.EventArgs)
    'cierra el formulario
    Me.Close()
End Sub

Private Sub btnCancela_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnCancela.Click
    'habilita el groupBox de preguntas
    GBPreguntas.Enabled = True
    Me.Close()
    'muestra formulario principal
    MenuPrincipal.Show()
End Sub

Private Sub CbMateria_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CbMateria.SelectedIndexChanged
    'DECLARACION DE VARIABLES
    Dim cadena As String
    Dim MateriaClave As String
    'OBTIENE EL VALOR DE MATERIA Y LA GUARDA EN LA VARIABLE MATERIACLAVE
    MateriaClave = CbMateria.SelectedValue.ToString()
    'ACTUALIZA EL COMBOBOX DE TEMA CUANDO LA MATERIA ELEGIDA POR EL USUARIO CAMBIA
    Try
        cadena = "select * from tema where mat_claveint = " & MateriaClave.ToString
        DataSetTema = oconexion.GetRows(cadena)
        CBtema.DataSource = DataSetTema.Tables(0)
        CBtema.DisplayMember = "tem_nombre"
        CBtema.ValueMember = "tem_claveint"
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try
End Sub

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Dim comando As New SqlCommand
    Dim param As SqlParameter
    Dim claveMateria As SqlParameter
    Dim descrip As SqlParameter
    Dim tem As SqlParameter
    Dim act As SqlParameter
    Dim preg As SqlParameter
    Dim ruta As SqlParameter
    Dim nomima As SqlParameter
    Dim nomimagen As SqlParameter
    Dim extimg As String
    Dim extimg As String
    Dim rutaImagenGuardar As String
    Dim vacia As Integer
    Dim i As Integer
    Dim j As Integer
    Dim cadena As String
    Dim Resultados As DataSet
    Dim tabla As DataTable
    Dim ImagenArchivo As String
    Dim nuIma As Integer
    Dim a As DataSet
    Dim valorMateria As Integer
    Dim valorTema As Integer
    Dim SourceFile As String
    Dim DestinationFile As String

    nomImagen = ""
    extimg = ""
    vacia = 0
    rutaImagenGuardar = ""
    'VERIFICA QUE TODAS LOS CAMPOS REQUERIDOS ESTEN CAPTURADOS
    If CbMateria.Text = "" Then
        vacia = 1
    End If
    If CBtema.Text = "" Then
        vacia = 1
    End If
    If tbNombreDinamica.Text = "" Then
        vacia = 1
    End If
    If CbNoPreguntas.Text = "" Then
        vacia = 1
    End If

    '.....Verifica los datos de la imagen
    If Len(TBRuta.Text) > 0 Then
        For i = 1 To Len(TBRuta.Text.ToString)
            If Mid(TBRuta.Text, i, 1) = "\" Then
                j = i
            End If
            Next i
            nomImagen = Mid(TBRuta.Text, j + 1, (Len(TBRuta.Text.ToString) - j - 4)) 'deberia ser el nombre solo de la img sin extension '(((Len(TBRuta.Text.ToString) + 1) - j + 1) ))
            extimg = Mid(TBRuta.Text, Len(TBRuta.Text.ToString) - 2, 3)
        End If
        'VERIFICA SI LA IMAGEN EXISTE EN DB
        cadena = "select * from imagen"
        Resultados = oconexion.SelectRows(a, oconexion.sqlCnn.ToString, cadena)
        'CHECA EL MAXIMO NUMERO DE IMAGEN
        If Resultados.Tables.Count > 0 Then
            tabla = Resultados.Tables(0)
            If tabla.Rows.Count > 0 Then
                nuIma = (tabla.Rows.Count + 1).ToString
            End If
        End If
        'CONCATENA EL NOMBRE DE LA IMAGEN CON LA EXTENSION
        ImagenArchivo = nomImagen + "." + extimg
        'VERIFICA SI EL USUARIO AGREGÓ UNA RUTA DE ARCHIVO DE IMAGEN
        If TBRuta.Text = "" Then
            vacia = 1
        End If
    End Sub

```

```

End If
'ASIGNA LA VARIABLE VALORMATERIA DEL VALOR QUE TIENE VALOR
MATERIA
valorMateria = CbMateria.SelectedValue
'ASIGNA LA VARIABLE VALORTEMA DEL VALOR QUE TIENE VALOR
TEMA
valorTema = CBTema.SelectedValue

'SI FALTAN POR CAPTURAR DATOS, MANDA MENSAJE DE CAPTURAR
TODOS LOS DATOS
If vacia = 1 Then
    MsgBox("Debe capturar todos los datos de la dinámica")
    CbMateria.SelectedValue = valorMateria
    CBTema.SelectedValue = valorTema
End If
'SI EL USUARIO HA CAPTURADO TODOS LOS DATOS, LLAMA EL SP
PARA CREAR DINAMICA
If vacia = 0 Then
    SourceFile = TBRuta.Text ' Define source file name.
    DestinationFile = "C:\Proyecto ver1\Imágenes\" +
ImagenArchivo ' Define target file name.
    FileCopy(SourceFile, DestinationFile) ' Copy source
to target.
    Try
        'llama sp
        comando.CommandType = CommandType.StoredProcedure
        comando.CommandText = "SPNvoDinamica"

        'manda parametro
        param = comando.Parameters.Add("@autorClave",
SqlDbType.Int)
        param.Direction = ParameterDirection.Input
        param.Value = FPassword.claveusuario
        'manda parametro
        claveMateria = comando.Parameters.Add("@matClave",
SqlDbType.Int)
        claveMateria.Direction = ParameterDirection.Input
        claveMateria.Value = CbMateria.SelectedValue
        'manda parametro
        descrip = comando.Parameters.Add("@desDin",
SqlDbType.VarChar)
        descrip.Direction = ParameterDirection.Input
        descrip.Value = tbNombreDinamica.Text
        'manda parametro
        tem = comando.Parameters.Add("@temaClave",
SqlDbType.Int)
        tem.Direction = ParameterDirection.Input
        tem.Value = CBTema.SelectedValue
        'manda parametro
        act = comando.Parameters.Add("@actividad",
SqlDbType.VarChar)
        act.Direction = ParameterDirection.Input
        act.Value = ""
        'manda parametro
        preg = comando.Parameters.Add("@numPreguntas",
SqlDbType.Int)
        preg.Direction = ParameterDirection.Input
        preg.Value = CbNoPreguntas.Text
        'manda parametro
        ruta = comando.Parameters.Add("@imagenRuta",
SqlDbType.VarChar)
        ruta.Direction = ParameterDirection.Input
        ruta.Value = DestinationFile.ToString

        'manda parametro
        nomima = comando.Parameters.Add("@nombreImagen",
SqlDbType.VarChar)
        nomima.Direction = ParameterDirection.Input
        nomima.Value = nomImagen.ToString

        oconexion.Executesp(comando)
        MessageBox.Show("Registro guardado")
    Catch m As Exception
        MessageBox.Show(m.Message)
    End Try
    End If
    End Sub
End Class

FORMULARIO MENU PRINCIPAL JUEGA
Public Class FPrincipalJuega
    Dim oconexion As New Conexion
    Dim DataSetListaMateria As DataSet
    Dim DataSetMateria As DataSet
    Dim DataSetTema As DataSet
    Public claveDinamica As Integer

    Private Sub FPrincipalJuega_Load(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles MyBase.Load
        Dim cadena As String
        Dim MateriaClave As String
        'LLENA EL COMBO DE MATERIA CON LAS MATERIAS EXISTENTES EN
DB
        Try
            cadena = "select * from materia "
            DataSetListaMateria = oconexion.GetRows(cadena)
            CbMateria.DisplayMember = "mat_nombre"
            CbMateria.ValueMember = "mat_clave"
            CbMateria.DataSource = DataSetListaMateria.Tables(0)
        Catch ex As Exception
            MsgBox(ex.Message)
        End Try
        'OBTIENE LA MATERIA SELECCIONADA EN CLAVE DE MATERIA
MateriaClave = CbMateria.SelectedValue.ToString()
        'LLENA COMBO BOX CON LOS TEMAS DE ACUERDO A LA MATERIA
        Try
            cadena = "select * from tema where mat_claveint = " &
MateriaClave.ToString
            DataSetTema = oconexion.GetRows(cadena)
            CBTema.DisplayMember = "tem_nombre"
            CBTema.ValueMember = "tem_claveint"
            CBTema.DataSource = DataSetTema.Tables(0)
        Catch ex As Exception
            MsgBox(ex.Message)
        End Try
        End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Button1.Click
        Dim cadena As String
        Dim arreglo() As String
        ' DESPLIEGA LAS ACTIVIDADES DIMNAMICAS DE ACUERDO A LA
MATERIA Y TEMA ELEGIDOS
        Try
            cadena = "select dnm_descripcion as DESCRIPCION,
DNM_numPreguntas as PREGUNTAS, usu_nombre + ' ' + USU_apaterno as
Autor, dnm_claveint as Clave from dinamica inner join usuario on
dinamica.aut_claveint = usuario.usu_claveint where mat_claveint= " &
CbMateria.SelectedValue & " and tem_claveint= " &
CBTema.SelectedValue
            DataSetMateria = oconexion.GetRows(cadena)
            DataGridView1.DataSource = DataSetMateria.Tables(0)
        Catch ex As Exception
            MsgBox("No existe dinamica con estos datos")
        End Try
        End Sub

    Private Sub DataGridView1_CellDoubleClick(ByVal sender As
Object, ByVal e As System.Windows.Forms.DataGridViewCellEventArgs)
Handles DataGridView1.CellDoubleClick
        Dim juego As String
        Dim preguntas As Integer
        Dim titulo As String
        'SI EL USUARIO HA SELECCIONADO UN JUEGO, ENTONCES VERIFICA
CUAL TABLERO DEBE SER MOSTRADO CON LA DINAMICA QUE ELGIO
        If cbJuego.Text <> "" Then
            claveDinamica =
DataSetMateria.Tables(0).Rows(e.RowIndex)("clave")
            juego = cbJuego.Text
            preguntas =
DataSetMateria.Tables(0).Rows(e.RowIndex)("Preguntas")
            titulo =
DataSetMateria.Tables(0).Rows(e.RowIndex)("DESCRIPCION")

            'SE ELIGE EL TABLERO DE ACUERDO CON EL NUMERO DE
PREGUNTAS Y TIPO DE JUEGO
            Select Case juego
                Case "Memorama"
                    Select Case preguntas
                        Case 6
                            FMemorama6P.Text = Titulo
                            FMemorama6P.Show()
                        Case 8
                            FMemorama7P.Text = Titulo
                            FMemorama7P.Show()
                        Case 9
                            FMemorama9P.Text = Titulo
                            FMemorama9P.Show()
                        Case 10
                            FMemorama10P.Text = Titulo
                            FMemorama10P.Show()
                        Case 12
                            FMemorama12P.Text = Titulo
                            FMemorama12P.Show()
                    End Select
                    FInstruccionMemorama.Show()
                Case "Relación de Columnas"
                    Select Case preguntas
                        Case 6
                            FDosColumn6P.Text = Titulo
                            FDosColumn6P.Show()
                        Case 8
                            FDosColumn8P.Text = Titulo
                            FDosColumn8P.Show()
                        Case 9
                            FDosColumn9P.Text = Titulo
                            FDosColumn9P.Show()
                        Case 10
                            FDosColumn10P.Text = Titulo
                            FDosColumn10P.Show()
                        Case 12
                            FDosColumn12P.Text = Titulo
                            FDosColumn12P.Show()
                    End Select
                    FInstruccionRC.Show()
                Case "Opción Múltiple"
                    Select Case preguntas
                        Case 6
                            FOpcionMultiple6P.Text = Titulo
                            FOpcionMultiple6P.Show()
                        Case 8
                            FOpcionMultiple8P.Text = Titulo
                            FOpcionMultiple8P.Show()
                        Case 9
                            FOpcionMultiple9P.Text = Titulo
                            FOpcionMultiple9P.Show()
                        Case 10
                            FOpcionMultiple10P.Text = Titulo
                            FOpcionMultiple10P.Show()
                    End Select
            End If
        End If
    End Sub
End Class

```

```

Case 12
    FOpcionMultiple12P.Text = Titulo
    FOpcionMultiple12P.Show()
End Select
FInstruccionOM.Show()

Case "Pregunta - Respuesta"
Select Case preguntas
Case 6
    FJalaRespuesta6P.Text = Titulo
    FJalaRespuesta6P.Show()
Case 8
    FJalaRespuesta8P.Text = Titulo
    FJalaRespuesta8P.Show()
Case 9
    FJalaRespuesta9P.Text = Titulo
    FJalaRespuesta9P.Show()
Case 10
    FJalaRespuesta10P.Text = Titulo
    FJalaRespuesta10P.Show()
Case 12
    FJalaRespuesta12P.Text = Titulo
    FJalaRespuesta12P.Show()
End Select
FInstruccionJalaRespuesta.Show()

Case "Verdadero Falso"
Select Case preguntas
Case 6
    FVerdaderoFalso6P.Text = Titulo
    FVerdaderoFalso6P.Show()
Case 8
    FVerdaderoFalso8P.Text = Titulo
    FVerdaderoFalso8P.Show()
Case 9
    FVerdaderoFalso9P.Text = Titulo
    FVerdaderoFalso9P.Show()
Case 10
    FVerdaderoFalso10P.Text = Titulo
    FVerdaderoFalso10P.Show()
Case 12
    FVerdaderoFalso12P.Text = Titulo
    FVerdaderoFalso12P.Show()
End Select
FInstruccionVF.Show()

End Select

Else
    MsgBox("Debe Elegir un Tipo de Juego")
End If
End Sub

Private Sub Button2_Click_1(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button2.Click
    'MUESTRA EL MENU PRINCIPAL
    MenuPrincipal.Show()
    Me.Hide()
End Sub

Private Sub Button3_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs)

    MsgBox(CBMateria.SelectedVale)

End Sub

Private Sub CBMateria_SelectedIndexChanged(ByVal sender As
Object, ByVal e As System.EventArgs) Handles
CBMateria.SelectedIndexChanged
    Dim cadena As String
    Dim MateriaClave As String
    MateriaClave = CBMateria.SelectedVale.ToString
    'LLENA EL COMBO DE TEMA DE ACUERDO CON CON LA MATERIA
ELEGIDA POR EL USUARIO

    CBTema.Text = ""
    Try
        cadena = "select * from tema where mat_claveint = " &
MateriaClave.ToString
        DataSetTema = oconexion.GetRows(cadena)
        CBTema.DataSource = DataSetTema.Tables(0)
        CBTema.DisplayMember = "tem_nombre"
        CBTema.ValueMember = "tem_claveint"
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try
End Sub

End Class

```

FORMULARIO DINAMICA DE DOS COLUMNAS

```

Public Class FDosColum6P
    Dim oconexion As New Conexion
    Dim Pregunta As String
    Dim Respuesta As String
    Dim bandera As Integer
    Dim enunciado1 As String
    Dim enunciado2 As String
    Dim DataSetDinamica As DataSet
    Dim ParesCorrectos As Integer = 0
    Dim termina As Integer = 0
    Dim numRTB1 As Integer
    Dim numRTB2 As Integer

    Private Sub FDosColum6P_Load(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles MyBase.Load
        Dim dinamica As Integer
        Dim cadena As String

        Dim i As Integer
        Dim Preguntas As New ArrayList()
        Dim Respuestas As New ArrayList()
        Dim value As Integer
        bandera = 1

        dinamica = FPrincipalJuega.claveDinamica
        'Llena Preguntas y respuestas'''
        'VERIFICA A CUALDE LOS DOS CASOS PERTENECE AL MOMENTO DAR
CLIC EN LA PRIMERA PREGUNTA
        'SI ES 1 ENTONCES SE GUARDA LA PREGUNTA 1 EN VARIABLE
ENUNCIADO1
        'SI ES 2 ENTONCES SE GUARDA EN LA VARIABLE ENUNCIADO2 Y SE
VERIFICA SI ES PAR CON LA RESPUESTA SELECCIONNADA

        Try
            cadena = "select cue_pregunta as pregunta,
cue_respuesta as respuesta from cuestionario where dnm_claveint =
" & dinamica & " and cue_pactiva = 1 " & " order by cue_claveint"
            DataSetDinamica = oconexion.GetRows(cadena)
        Catch ex As Exception
            MsgBox(ex.Message)
        End Try

        For i = 0 To 5

            Preguntas.Add(DataSetDinamica.Tables(0).Rows(i)("Pregunta"))
            Respuestas.Add(DataSetDinamica.Tables(0).Rows(i)("Respuesta"))
        Next

        'Llena Fila de Preguntas
        'boton1
        Randomize()
        value = CInt(Int(((Preguntas.Count - 1) * Rnd() + 1))
RichTextBox1.Text = Preguntas(value)
Preguntas.RemoveAt(value)

        'boton2
        Randomize()
        value = CInt(Int(((Preguntas.Count - 1) * Rnd() + 1))
RichTextBox2.Text = Preguntas(value)
Preguntas.RemoveAt(value)

        'boton3
        Randomize()
        value = CInt(Int(((Preguntas.Count - 1) * Rnd() + 1))
RichTextBox3.Text = Preguntas(value)
Preguntas.RemoveAt(value)

        'boton4
        Randomize()
        value = CInt(Int(((Preguntas.Count - 1) * Rnd() + 1))
RichTextBox4.Text = Preguntas(value)
Preguntas.RemoveAt(value)

        'boton5
        Randomize()
        value = CInt(Int(((Preguntas.Count - 1) * Rnd() + 1))
RichTextBox5.Text = Preguntas(value)
Preguntas.RemoveAt(value)

        'boton6
        RichTextBox6.Text = Preguntas(0)

        'Llena Fila de Respuestas
        'boton7
        Randomize()
        value = CInt(Int(((Respuestas.Count - 1) * Rnd() + 1))
RichTextBox7.Text = Respuestas(value)
Respuestas.RemoveAt(value)

        'boton8
        Randomize()
        value = CInt(Int(((Respuestas.Count - 1) * Rnd() + 1))
RichTextBox8.Text = Respuestas(value)
Respuestas.RemoveAt(value)

        'boton9
        Randomize()
        value = CInt(Int(((Respuestas.Count - 1) * Rnd() + 1))
RichTextBox9.Text = Respuestas(value)

```

```

Respuestas.RemoveAt(value)
'boton10
Randomize()
value = Cint(Int(((Respuestas.Count - 1) * Rnd() + 1))
RichTextBox10.Text = Respuestas(value)
Respuestas.RemoveAt(value)
'boton11
Randomize()
value = Cint(Int(((Respuestas.Count - 1) * Rnd() + 1))
RichTextBox11.Text = Respuestas(value)
Respuestas.RemoveAt(value)
'boton12
RichTextBox12.Text = Respuestas(0)

End Sub

Private Sub RichTextBox1_Click(ByVal sender As Object, ByVal e
As System.EventArgs) Handles RichTextBox1.Click
Select Case bandera
'VERIFICA A CUALDE LOS DOS CASOS PERTENECE AL MOMENTO
DAR CLIC EN LA PRIMERA PREGUNTA
'SI ES 1 ENTONCES SE GUARDA LA PREGUNTA 1 EN VARIABLE
ENUNCIAD01
'SI ES 2 ENTONCES SE GUARDA EN LA VARIABLE ENUNCIAD02 Y
SE VERIFICA SI ES PAR CON LA RESPUESTA SELECCIONNADA

Case 1
Label15.Text = "1"
bandera = bandera + 1
enunciad01 = RichTextBox1.Text
numRTB1 = 1

Case 2
Label15.Text = "1"
bandera = bandera + 1
enunciad02 = RichTextBox1.Text
numRTB2 = 1
'aqui va la funcion para comprobar si es el par
VerificaFAres()

Case 3

End Select
End Sub

Private Sub RichTextBox3_Click(ByVal sender As Object, ByVal e
As System.EventArgs) Handles RichTextBox3.Click
Select Case bandera
'VERIFICA A CUALDE LOS DOS CASOS PERTENECE AL MOMENTO
DAR CLIC EN LA PRIMERA PREGUNTA
'SI ES 1 ENTONCES SE GUARDA LA PREGUNTA 1 EN VARIABLE
ENUNCIAD01
'SI ES 2 ENTONCES SE GUARDA EN LA VARIABLE ENUNCIAD02 Y
SE VERIFICA SI ES PAR CON LA RESPUESTA SELECCIONNADA

Case 1
Label15.Text = "3"
bandera = bandera + 1
enunciad01 = RichTextBox3.Text
numRTB1 = 3

Case 2
Label15.Text = "3"
bandera = bandera + 1
enunciad02 = RichTextBox3.Text
numRTB2 = 3
'aqui va la funcion para comprobar si es el par
VerificaFAres()

Case 3

End Select
End Sub

Private Sub RichTextBox4_Click(ByVal sender As Object, ByVal e
As System.EventArgs) Handles RichTextBox4.Click
'VERIFICA A CUALDE LOS DOS CASOS PERTENECE AL MOMENTO DAR
CLIC EN LA PRIMERA PREGUNTA
'SI ES 1 ENTONCES SE GUARDA LA PREGUNTA 1 EN VARIABLE
ENUNCIAD01
'SI ES 2 ENTONCES SE GUARDA EN LA VARIABLE ENUNCIAD02 Y SE
VERIFICA SI ES PAR CON LA RESPUESTA SELECCIONNADA

Select Case bandera
Case 1
Label15.Text = "4"
bandera = bandera + 1
enunciad01 = RichTextBox4.Text
numRTB1 = 4

Case 2
Label15.Text = "4"
bandera = bandera + 1
enunciad02 = RichTextBox4.Text
numRTB2 = 4
'aqui va la funcion para comprobar si es el par
VerificaFAres()

Case 3

End Select
End Sub

Private Sub RichTextBox5_Click(ByVal sender As Object, ByVal e
As System.EventArgs) Handles RichTextBox5.Click
'VERIFICA A CUALDE LOS DOS CASOS PERTENECE AL MOMENTO DAR
CLIC EN LA PRIMERA PREGUNTA
'SI ES 1 ENTONCES SE GUARDA LA PREGUNTA 1 EN VARIABLE
ENUNCIAD01
'SI ES 2 ENTONCES SE GUARDA EN LA VARIABLE ENUNCIAD02 Y SE
VERIFICA SI ES PAR CON LA RESPUESTA SELECCIONNADA

Select Case bandera
Case 1
Label15.Text = "5"
bandera = bandera + 1
enunciad01 = RichTextBox5.Text
numRTB1 = 5

Case 2
Label15.Text = "5"
bandera = bandera + 1
enunciad02 = RichTextBox5.Text
numRTB2 = 5
'aqui va la funcion para comprobar si es el par
VerificaFAres()

Case 3

End Select
End Sub

Private Sub RichTextBox6_Click(ByVal sender As Object, ByVal e
As System.EventArgs) Handles RichTextBox6.Click
'VERIFICA A CUALDE LOS DOS CASOS PERTENECE AL MOMENTO DAR
CLIC EN LA PRIMERA PREGUNTA
'SI ES 1 ENTONCES SE GUARDA LA PREGUNTA 1 EN VARIABLE
ENUNCIAD01
'SI ES 2 ENTONCES SE GUARDA EN LA VARIABLE ENUNCIAD02 Y SE
VERIFICA SI ES PAR CON LA RESPUESTA SELECCIONNADA

Select Case bandera
Case 1
Label15.Text = "6"
bandera = bandera + 1
enunciad01 = RichTextBox6.Text
numRTB1 = 6

Case 2
Label15.Text = "6"
bandera = bandera + 1
enunciad02 = RichTextBox6.Text
numRTB2 = 6
'aqui va la funcion para comprobar si es el par
VerificaFAres()

Case 3

End Select
End Sub

Private Sub RichTextBox7_Click(ByVal sender As Object, ByVal e
As System.EventArgs) Handles RichTextBox7.Click
'VERIFICA A CUALDE LOS DOS CASOS PERTENECE AL MOMENTO DAR
CLIC EN LA PRIMERA PREGUNTA
'SI ES 1 ENTONCES SE GUARDA LA PREGUNTA 1 EN VARIABLE
ENUNCIAD01
'SI ES 2 ENTONCES SE GUARDA EN LA VARIABLE ENUNCIAD02 Y SE
VERIFICA SI ES PAR CON LA RESPUESTA SELECCIONNADA

Select Case bandera
Case 1
Label16.Text = "A"
bandera = bandera + 1
enunciad01 = RichTextBox7.Text
numRTB1 = 7

Case 2
Label16.Text = "A"
bandera = bandera + 1
enunciad02 = RichTextBox7.Text
numRTB2 = 7
'aqui va la funcion para comprobar si es el par
VerificaFAres()

Case 3

End Select
End Sub

Private Sub RichTextBox8_Click(ByVal sender As Object, ByVal e
As System.EventArgs) Handles RichTextBox8.Click
'VERIFICA A CUALDE LOS DOS CASOS PERTENECE AL MOMENTO DAR
CLIC EN LA PRIMERA PREGUNTA
'SI ES 1 ENTONCES SE GUARDA LA PREGUNTA 1 EN VARIABLE
ENUNCIAD01
'SI ES 2 ENTONCES SE GUARDA EN LA VARIABLE ENUNCIAD02 Y SE
VERIFICA SI ES PAR CON LA RESPUESTA SELECCIONNADA

Select Case bandera
Case 1
Label16.Text = "B"
bandera = bandera + 1
enunciad01 = RichTextBox8.Text
numRTB1 = 8

Case 2
Label16.Text = "B"
bandera = bandera + 1
enunciad02 = RichTextBox8.Text
numRTB2 = 8
'aqui va la funcion para comprobar si es el par
VerificaFAres()

Case 3

End Select
End Sub

```

```

'aquí va la funcion para comprobar si es el par
VerificaPAres()

Case 3

End Select
End Sub

Public Function VerificaPAres()
    Dim PregRespCorrecta As Integer
    Dim EnunciadoTipo As String
    Dim i As Integer

    PregRespCorrecta = 0
    EnunciadoTipo = ""
    'VERIFICA SI ENUNCIADO1 ES PREGUNTA ; SI ES ASI, LA
    VARIABLE PregRespCorrecta = 1;
    'Y VARIABLE ParesCorrectos AUMENTA 1, SI NO ENTONCES ES UNA
    RESPUESTA

    For i = 0 To 5
        If enunciado1 =
        DataSetDinamica.Tables(0).Rows(i)("Pregunta") Then
            If enunciado2 =
            DataSetDinamica.Tables(0).Rows(i)("Respuesta") Then
                PregRespCorrecta = 1
                ParesCorrectos = ParesCorrectos + 1
            End If
        Else
            EnunciadoTipo = "R"
        End If
    Next

    'SI ES UNA RESPUESTA ENTONCES VERIFICA SI ES PAR DE LA
    PREGUNTA ELEGIDA
    If EnunciadoTipo = "R" Then
        For i = 0 To 5
            If enunciado1 =
            DataSetDinamica.Tables(0).Rows(i)("Respuesta") Then
                If enunciado2 =
                DataSetDinamica.Tables(0).Rows(i)("Pregunta") Then
                    PregRespCorrecta = 1
                    ParesCorrectos = ParesCorrectos + 1
                End If
            End If
        Next
    End If

    'SI SON PARES DE PREGUNTA-RESPUESTA ENTONCES DESPLIEGA
    LEYENDA RESPUESTA CORRECTA
    If PregRespCorrecta = 1 Then
        lblMensaje.Text = "Respuesta Correcta !!!! "
        HF.Visible = True
        uf.Visible = False

        .....
        'AL SELECCIONAR UNA PREGUNTA O RESPUESTA CAMBIA DE
        COLOR EL RICHTEXTBOX
        Select Case numRTB1
            Case 1
                RichTextBox1.BackColor = Color.Beige
            Case 2
                RichTextBox2.BackColor = Color.Beige
            Case 3
                RichTextBox3.BackColor = Color.Beige
            Case 4
                RichTextBox4.BackColor = Color.Beige
            Case 5
                RichTextBox5.BackColor = Color.Beige
            Case 6
                RichTextBox6.BackColor = Color.Beige
            Case 7
                RichTextBox7.BackColor = Color.Beige
            Case 8
                RichTextBox8.BackColor = Color.Beige
            Case 9
                RichTextBox9.BackColor = Color.Beige
            Case 10
                RichTextBox10.BackColor = Color.Beige
            Case 11
                RichTextBox11.BackColor = Color.Beige
            Case 12
                RichTextBox12.BackColor = Color.Beige
        End Select
        Select Case numRTB2
            Case 1
                RichTextBox1.BackColor = Color.Beige
            Case 2
                RichTextBox2.BackColor = Color.Beige
            Case 3
                RichTextBox3.BackColor = Color.Beige
            Case 4
                RichTextBox4.BackColor = Color.Beige
            Case 5
                RichTextBox5.BackColor = Color.Beige
            Case 6
                RichTextBox6.BackColor = Color.Beige
            Case 7
                RichTextBox7.BackColor = Color.Beige
            Case 8
                RichTextBox8.BackColor = Color.Beige
        End Select
    End Sub

Private Sub RichTextBox9_Click(ByVal sender As Object, ByVal e
As System.EventArgs) Handles RichTextBox9.Click
    'VERIFICA A CUALDE LOS DOS CASOS PERTENECE AL MOMENTO DAR
    CLIC EN LA PRIMERA PREGUNTA
    'SI ES 1 ENTONCES SE GUARDA LA PREGUNTA 1 EN VARIABLE
    ENUNCIADO1
    'SI ES 2 ENTONCES SE GUARDA EN LA VARIABLE ENUNCIADO2 Y SE
    VERIFICA SI ES PAR CON LA RESPUESTA SELECCIONNADA

    Select Case bandera
        Case 1
            Label16.Text = "C"
            bandera = bandera + 1
            enunciado1 = RichTextBox9.Text
            numRTB1 = 9
        Case 2
            Label16.Text = "C"
            bandera = bandera + 1
            enunciado2 = RichTextBox9.Text
            numRTB2 = 9
            'aquí va la funcion para comprobar si es el par
            VerificaPAres()
        Case 3

    End Select
End Sub

Private Sub RichTextBox10_Click(ByVal sender As Object, ByVal e
As System.EventArgs) Handles RichTextBox10.Click
    'VERIFICA A CUALDE LOS DOS CASOS PERTENECE AL MOMENTO DAR
    CLIC EN LA PRIMERA PREGUNTA
    'SI ES 1 ENTONCES SE GUARDA LA PREGUNTA 1 EN VARIABLE
    ENUNCIADO1
    'SI ES 2 ENTONCES SE GUARDA EN LA VARIABLE ENUNCIADO2 Y SE
    VERIFICA SI ES PAR CON LA RESPUESTA SELECCIONNADA

    Select Case bandera
        Case 1
            Label16.Text = "D"
            bandera = bandera + 1
            enunciado1 = RichTextBox10.Text
            numRTB1 = 10
        Case 2
            Label16.Text = "D"
            bandera = bandera + 1
            enunciado2 = RichTextBox10.Text
            numRTB2 = 10
            'aquí va la funcion para comprobar si es el par
            VerificaPAres()
        Case 3

    End Select
End Sub

Private Sub RichTextBox11_Click(ByVal sender As Object, ByVal e
As System.EventArgs) Handles RichTextBox11.Click
    'VERIFICA A CUALDE LOS DOS CASOS PERTENECE AL MOMENTO DAR
    CLIC EN LA PRIMERA PREGUNTA
    'SI ES 1 ENTONCES SE GUARDA LA PREGUNTA 1 EN VARIABLE
    ENUNCIADO1
    'SI ES 2 ENTONCES SE GUARDA EN LA VARIABLE ENUNCIADO2 Y SE
    VERIFICA SI ES PAR CON LA RESPUESTA SELECCIONNADA

    Select Case bandera
        Case 1
            Label16.Text = "E"
            bandera = bandera + 1
            enunciado1 = RichTextBox11.Text
            numRTB1 = 11
        Case 2
            Label16.Text = "E"
            bandera = bandera + 1
            enunciado2 = RichTextBox11.Text
            numRTB2 = 11
            'aquí va la funcion para comprobar si es el par
            VerificaPAres()
        Case 3

    End Select
End Sub

Private Sub RichTextBox12_Click(ByVal sender As Object, ByVal e
As System.EventArgs) Handles RichTextBox12.Click
    'VERIFICA A CUALDE LOS DOS CASOS PERTENECE AL MOMENTO DAR
    CLIC EN LA PRIMERA PREGUNTA
    'SI ES 1 ENTONCES SE GUARDA LA PREGUNTA 1 EN VARIABLE
    ENUNCIADO1
    'SI ES 2 ENTONCES SE GUARDA EN LA VARIABLE ENUNCIADO2 Y SE
    VERIFICA SI ES PAR CON LA RESPUESTA SELECCIONNADA

    Select Case bandera
        Case 1
            Label16.Text = "F"
            bandera = bandera + 1
            enunciado1 = RichTextBox12.Text
            numRTB1 = 12
        Case 2
            Label16.Text = "F"
            bandera = bandera + 1
            enunciado2 = RichTextBox12.Text
            numRTB2 = 12
        Case 3
    End Select
End Sub

```

```

        RichTextBox8.BackColor = Color.Beige
    Case 9
        RichTextBox9.BackColor = Color.Beige
    Case 10
        RichTextBox10.BackColor = Color.Beige
    Case 11
        RichTextBox11.BackColor = Color.Beige
    Case 12
        RichTextBox12.BackColor = Color.Beige
    End Select
    '.....
    ' SI TODAS LAS PREGUNTAS Y RESPUESTAS SON PARES ,
    FELICITA AL USUARIO
    If ParesCorrectos = 6 Then
        lblMensaje.Text = "Felicidades GANASTE !!!"
        termina = 1
        HF.Visible = True
        uf.Visible = False
    End If
    'SI NO SON IGUALES LAS PREGUNTAS Y RESPUESTAS ENTONCES
    DESPLIEGA "RESPUESTA INCORRECTA"
    ElseIf ((PregRespCorrecta = 0) And (termina = 0)) Then
        lblMensaje.Text = "Respuesta incorrecta"
        HF.Visible = False
        uf.Visible = True
    End If
    bandera = 1
    enunciado1 = ""
    enunciado2 = ""
    PregRespCorrecta = 0
    EnunciadoTipo = ""

End Function

Private Sub RichTextBox2_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles RichTextBox2.Click
    'VERIFICA A CUALDE LOS DOS CASOS PERTENECE AL MOMENTO DAR
    CLIC EN LA PRIMERA PREGUNTA
    'SI ES 1 ENTONCES SE GUARDA LA PREGUNTA 1 EN VARIABLE
    ENUNCIADO1
    'SI ES 2 ENTONCES SE GUARDA EN LA VARIABLE ENUNCIADO2 Y SE
    VERIFICA SI ES PAR CON LA RESPUESTA SELECCIONNADA

    Select Case bandera
    Case 1
        Label15.Text = "2"
        bandera = bandera + 1
        enunciado1 = RichTextBox2.Text
        numRTB1 = 2
    Case 2
        Label15.Text = "2"
        bandera = bandera + 1
        enunciado2 = RichTextBox2.Text
        numRTB2 = 2
        'aquí va la función para comprobar si es el par
        VerificaPares()
    Case 3

    End Select
End Sub

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) _
    'SALIR DEL JUEGO
    If MessageBox.Show("¿Esta seguro de salir del juego?",
    "Salida", _
        MessageBoxButtons.OKCancel,
        MessageBoxIcon.Exclamation) = DialogResult.OK Then
        FPrincipalJuego.Show()
        Me.Hide()
    End If
End Sub

Private Sub AyudaToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
    AyudaToolStripMenuItem.Click
    'MUESTRA ALAS INSTRUCCIONES
    FInstruccionRC.Show()
End Sub

Private Sub SalirToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
    SalirToolStripMenuItem.Click
    'PREGUNTA SI DESEA SALIR DEL JUEGO
    If MessageBox.Show("¿Esta seguro de salir del juego?",
    "Salida", _
        MessageBoxButtons.OKCancel,
        MessageBoxIcon.Exclamation) = DialogResult.OK Then
        FPrincipalJuego.Show()
        Me.Close()
    End If
End Sub
End Class

Dim enunciado2 As String
Dim DataSetDinamica As DataSet
Dim ParesCorrectos As Integer = 0
Dim termina As Integer = 0
Dim numRTB1 As Integer
Dim numRTB2 As Integer
Dim TOMARESPUESTA As Integer = 0
Dim ETIQUETA As Integer = 0

Dim Preguntas As New ArrayList()
Dim Respuestas As New ArrayList()
Dim buenas As Integer

Private Sub FJalaRespuesta6P_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load

    Dim dinamica As Integer
    Dim cadena As String
    Dim i As Integer
    Dim value As Integer
    bandera = 1

    dinamica = FPrincipalJuego.claveDinamica
    'Llena Preguntas y respuestas'
    Try
        cadena = "select cue_pregunta as pregunta,
        cue_respuesta as respuesta from cuestionario where dnm_claveint =
        " & dinamica & " and cue_pactiva = 1 " & " order by cue_claveint"
        DataSetDinamica = oconexion.GetRows(cadena)
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try

    For i = 0 To 5
        Preguntas.Add(DataSetDinamica.Tables(0).Rows(i)("Pregunta"))
        Respuestas.Add(DataSetDinamica.Tables(0).Rows(i)("Respuesta"))
    Next

    'Llena Fila de Preguntas
    'boton1
    Randomize()
    value = CInt(Int(((Preguntas.Count - 1) * Rnd()) + 1))
    Label1.Text = Preguntas(value)
    Preguntas.RemoveAt(value)

    'boton2
    Randomize()
    value = CInt(Int(((Preguntas.Count - 1) * Rnd()) + 1))
    Label2.Text = Preguntas(value)
    Preguntas.RemoveAt(value)

    'boton3
    Randomize()
    value = CInt(Int(((Preguntas.Count - 1) * Rnd()) + 1))
    Label3.Text = Preguntas(value)
    Preguntas.RemoveAt(value)

    'boton4
    Randomize()
    value = CInt(Int(((Preguntas.Count - 1) * Rnd()) + 1))
    Label4.Text = Preguntas(value)
    Preguntas.RemoveAt(value)

    'boton5
    Randomize()
    value = CInt(Int(((Preguntas.Count - 1) * Rnd()) + 1))
    Label5.Text = Preguntas(value)
    Preguntas.RemoveAt(value)

    'boton6
    Label6.Text = Preguntas(0)

    'Llena Fila de Respuestas
    'boton7
    Randomize()
    value = CInt(Int(((Respuestas.Count - 1) * Rnd()) + 1))
    Label7.Text = Respuestas(value)
    Respuestas.RemoveAt(value)

    'boton8
    Randomize()
    value = CInt(Int(((Respuestas.Count - 1) * Rnd()) + 1))
    Label8.Text = Respuestas(value)
    Respuestas.RemoveAt(value)

    'boton9
    Randomize()
    value = CInt(Int(((Respuestas.Count - 1) * Rnd()) + 1))
    Label9.Text = Respuestas(value)
    Respuestas.RemoveAt(value)

    'boton10
    Randomize()
    value = CInt(Int(((Respuestas.Count - 1) * Rnd()) + 1))
    Label10.Text = Respuestas(value)
    Respuestas.RemoveAt(value)

    'boton11
    Randomize()
    value = CInt(Int(((Respuestas.Count - 1) * Rnd()) + 1))
    Label11.Text = Respuestas(value)
    Respuestas.RemoveAt(value)

    'boton12
    Label12.Text = Respuestas(0)

End Sub

Private Sub FJalaRespuesta6P_MouseMove(ByVal sender As Object,
ByVal e As System.Windows.Forms.MouseEventArgs) Handles
    Me.MouseMove

    'VERIFICA QUE LA RESPUESTA SEA LA CORRECTA A LA PREGUNTA,
    SI ES ASÍ, LA RESPUESTA QUEDA COMO PAR DE LA PREGUNTA;
    'SI NO ASÍ LA RESPUESTA REGRESA A SU LUGAR.

```

```

If TOMARESPUESTA = 1 Then
    If ETIQUETA = 7 Then
        Label7.Location = New
Point((FJalaRespuesta6P.MousePosition.X),
FJalaRespuesta6P.MousePosition.Y - 26)
    End If
End If
If TOMARESPUESTA = 1 Then
    If ETIQUETA = 8 Then
        Label8.Location = New
Point((FJalaRespuesta6P.MousePosition.X),
FJalaRespuesta6P.MousePosition.Y - 25)
    End If
End If
If TOMARESPUESTA = 1 Then
    If ETIQUETA = 9 Then
        Label9.Location = New
Point((FJalaRespuesta6P.MousePosition.X),
FJalaRespuesta6P.MousePosition.Y - 25)
    End If
End If
If TOMARESPUESTA = 1 Then
    If ETIQUETA = 10 Then
        Label10.Location = New
Point((FJalaRespuesta6P.MousePosition.X),
FJalaRespuesta6P.MousePosition.Y - 25)
    End If
End If
If TOMARESPUESTA = 1 Then
    If ETIQUETA = 11 Then
        Label11.Location = New
Point((FJalaRespuesta6P.MousePosition.X),
FJalaRespuesta6P.MousePosition.Y - 25)
    End If
End If
If TOMARESPUESTA = 1 Then
    If ETIQUETA = 12 Then
        Label12.Location = New
Point((FJalaRespuesta6P.MousePosition.X),
FJalaRespuesta6P.MousePosition.Y - 25)
    End If
End If
End Sub

Private Sub Label7_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Label7.Click

    'NUMERO QUE LE CORESPONDE A LA RESPUESTA ELEGIDA
TOMARESPUESTA = 1
ETIQUETA = 7

End Sub

Private Sub Button1_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Button1.Click
    Dim ETIQUETATENIA As Integer
    Dim i As Integer
    Dim respCorrecta As String = ""
    'VERIFICA CUAL ES LA RESPUESTA CORRECTA DE LA PREGUNTA EN
DB
    For i = 0 To 5
        If (DataSetDinamica.Tables(0).Rows(i)("pregunta") =
Label11.Text) Then
            respCorrecta =
DataSetDinamica.Tables(0).Rows(i)("respuesta")
        End If
    Next

    ETIQUETATENIA = ETIQUETA
TOMARESPUESTA = 0
ETIQUETA = 0

    'VERIFICA CUAL DE TODAS LAS RESPUESTAS RESPONDE A LA
PREGUNTA 1, SI LA RESPUESTA SELECCIONADA ES ZLA CORRECTA,
'LA DEJA EN LA POSICION QUE RESPONDE A LA PREGUNTA; SI NO
ES ASI, LA REGRESA A SU LUGAR DE ORIGEN
If ETIQUETATENIA = 7 Then
    If (respCorrecta.ToString <> (Label7.Text)) Then
        Label7.Location = New Point(35, 495)
    Else
        buenas = buenas + 1
    End If
End If

If ETIQUETATENIA = 8 Then
    If (respCorrecta.ToString <> (Label8.Text)) Then
        Label8.Location = New Point(35, 530)
    Else
        buenas = buenas + 1
    End If
End If

If ETIQUETATENIA = 9 Then
    If (respCorrecta.ToString <> (Label9.Text)) Then
        Label9.Location = New Point(35, 571)
    Else
        buenas = buenas + 1
    End If
End If

If ETIQUETATENIA = 10 Then
    If (respCorrecta.ToString <> (Label10.Text)) Then
        Label10.Location = New Point(35, 610)
    Else
        buenas = buenas + 1
    End If
End If

If ETIQUETATENIA = 11 Then
    If (respCorrecta.ToString <> (Label11.Text)) Then
        Label11.Location = New Point(35, 656)
    Else
        buenas = buenas + 1
    End If
    buenas = buenas + 1
End If

If ETIQUETATENIA = 12 Then
    If (respCorrecta.ToString <> (Label12.Text)) Then
        Label12.Location = New Point(35, 697)
    Else
        buenas = buenas + 1
    End If
End If
End Sub

Private Sub Label8_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Label8.Click

    'NUMERO QUE LE CORESPONDE A LA RESPUESTA ELEGIDA
TOMARESPUESTA = 1
ETIQUETA = 8

End Sub

Private Sub Label9_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Label9.Click

    'NUMERO QUE LE CORESPONDE A LA RESPUESTA ELEGIDA
TOMARESPUESTA = 1
ETIQUETA = 9

End Sub

Private Sub Label10_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Label10.Click

    'NUMERO QUE LE CORESPONDE A LA RESPUESTA ELEGIDA
TOMARESPUESTA = 1
ETIQUETA = 10

End Sub

Private Sub Label11_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Label11.Click

    'NUMERO QUE LE CORESPONDE A LA RESPUESTA ELEGIDA
TOMARESPUESTA = 1
ETIQUETA = 11

End Sub

Private Sub Label12_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Label12.Click

    'NUMERO QUE LE CORESPONDE A LA RESPUESTA ELEGIDA
TOMARESPUESTA = 1
ETIQUETA = 12

End Sub

Private Sub Button3_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Button3.Click

    Dim ETIQUETATENIA As Integer
    Dim i As Integer
    Dim respCorrecta As String = ""
    'VERIFICA CUAL ES LA RESPUESTA CORRECTA DE LA PREGUNTA EN
DB
    For i = 0 To 5
        If (DataSetDinamica.Tables(0).Rows(i)("pregunta") =
Label3.Text) Then
            respCorrecta =
DataSetDinamica.Tables(0).Rows(i)("respuesta")
        End If
    Next

    ETIQUETATENIA = ETIQUETA
TOMARESPUESTA = 0
ETIQUETA = 0

    'VERIFICA CUAL DE TODAS LAS RESPUESTAS RESPONDE A LA
PREGUNTA 1, SI LA RESPUESTA SELECCIONADA ES ZLA CORRECTA,
'LA DEJA EN LA POSICION QUE RESPONDE A LA PREGUNTA; SI NO
ES ASI, LA REGRESA A SU LUGAR DE ORIGEN
If ETIQUETATENIA = 7 Then
    If (respCorrecta.ToString <> (Label7.Text)) Then
        Label7.Location = New Point(35, 495)
    Else
        buenas = buenas + 1
    End If
End If

If ETIQUETATENIA = 8 Then
    If (respCorrecta.ToString <> (Label8.Text)) Then
        Label8.Location = New Point(35, 530)
    Else
        buenas = buenas + 1
    End If
End If

```

```

        buenas = buenas + 1
    End If
End If

If ETIQUETATENIA = 9 Then
    If (respCorrecta.ToString <> (Label9.Text)) Then
        Label9.Location = New Point(35, 571)
    Else
        buenas = buenas + 1
    End If
End If

If ETIQUETATENIA = 10 Then
    If (respCorrecta.ToString <> (Label10.Text)) Then
        Label10.Location = New Point(35, 610)
    Else
        buenas = buenas + 1
    End If
End If

If ETIQUETATENIA = 11 Then
    If (respCorrecta.ToString <> (Label11.Text)) Then
        Label11.Location = New Point(35, 656)
    Else
        buenas = buenas + 1
    End If
End If

If ETIQUETATENIA = 12 Then
    If (respCorrecta.ToString <> (Label12.Text)) Then
        Label12.Location = New Point(35, 697)
    Else
        buenas = buenas + 1
    End If
End If
End Sub

Private Sub Button2_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Button2.Click
    Dim ETIQUETATENIA As Integer
    Dim i As Integer
    Dim respCorrecta As String = ""
    'VERIFICA CUAL ES LA RESPUESTA CORRECTA DE LA PREGUNTA EN
DB

    For i = 0 To 5
        If (DataSetDinamica.Tables(0).Rows(i)("pregunta") =
Label2.Text) Then
            respCorrecta =
DataSetDinamica.Tables(0).Rows(i)("respuesta")
        End If
    Next

    ETIQUETATENIA = ETIQUETA
    TOMARESPUESTA = 0
    ETIQUETA = 0

    'VERIFICA CUAL DE TODAS LAS RESPUESTAS RESPONDE A LA
PREGUNTA 1, SI LA RESPUESTA SELECCIONADA ES ZLA CORRECTA,
'LA DEJA EN LA POSICION QUE RESPONDE A LA PREGUNTA; SI NO
ES ASI, LA REGRESA A SU LUGAR DE ORIGEN

    If ETIQUETATENIA = 7 Then
        If (respCorrecta.ToString <> (Label7.Text)) Then
            Label7.Location = New Point(35, 495)
        Else
            buenas = buenas + 1
        End If
    End If

    If ETIQUETATENIA = 8 Then
        If (respCorrecta.ToString <> (Label8.Text)) Then
            Label8.Location = New Point(35, 530)
        Else
            buenas = buenas + 1
        End If
    End If

    If ETIQUETATENIA = 9 Then
        If (respCorrecta.ToString <> (Label9.Text)) Then
            Label9.Location = New Point(35, 571)
        Else
            buenas = buenas + 1
        End If
    End If

    If ETIQUETATENIA = 10 Then
        If (respCorrecta.ToString <> (Label10.Text)) Then
            Label10.Location = New Point(35, 610)
        Else
            buenas = buenas + 1
        End If
    End If

    If ETIQUETATENIA = 11 Then
        If (respCorrecta.ToString <> (Label11.Text)) Then
            Label11.Location = New Point(35, 656)
        Else
            buenas = buenas + 1
        End If
    End If

    If ETIQUETATENIA = 12 Then
        If (respCorrecta.ToString <> (Label12.Text)) Then
            Label12.Location = New Point(35, 697)
        Else
            buenas = buenas + 1
        End If
    End If
End Sub

Private Sub Button4_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Button4.Click
    Dim ETIQUETATENIA As Integer
    Dim i As Integer
    Dim respCorrecta As String = ""
    'VERIFICA CUAL ES LA RESPUESTA CORRECTA DE LA PREGUNTA EN
DB

    For i = 0 To 5
        If (DataSetDinamica.Tables(0).Rows(i)("pregunta") =
Label4.Text) Then
            respCorrecta =
DataSetDinamica.Tables(0).Rows(i)("respuesta")
        End If
    Next

    ETIQUETATENIA = ETIQUETA
    TOMARESPUESTA = 0
    ETIQUETA = 0

    'VERIFICA CUAL DE TODAS LAS RESPUESTAS RESPONDE A LA
PREGUNTA 1, SI LA RESPUESTA SELECCIONADA ES ZLA CORRECTA,
'LA DEJA EN LA POSICION QUE RESPONDE A LA PREGUNTA; SI NO
ES ASI, LA REGRESA A SU LUGAR DE ORIGEN

    If ETIQUETATENIA = 7 Then
        If (respCorrecta.ToString <> (Label7.Text)) Then
            Label7.Location = New Point(35, 495)
        Else
            buenas = buenas + 1
        End If
    End If

    If ETIQUETATENIA = 8 Then
        If (respCorrecta.ToString <> (Label8.Text)) Then
            Label8.Location = New Point(35, 530)
        Else
            buenas = buenas + 1
        End If
    End If

    If ETIQUETATENIA = 9 Then
        If (respCorrecta.ToString <> (Label9.Text)) Then
            Label9.Location = New Point(35, 571)
        Else
            buenas = buenas + 1
        End If
    End If

    If ETIQUETATENIA = 10 Then
        If (respCorrecta.ToString <> (Label10.Text)) Then
            Label10.Location = New Point(35, 610)
        Else
            buenas = buenas + 1
        End If
    End If

    If ETIQUETATENIA = 11 Then
        If (respCorrecta.ToString <> (Label11.Text)) Then
            Label11.Location = New Point(35, 656)
        Else
            buenas = buenas + 1
        End If
    End If

    If ETIQUETATENIA = 12 Then
        If (respCorrecta.ToString <> (Label12.Text)) Then
            Label12.Location = New Point(35, 697)
        Else
            buenas = buenas + 1
        End If
    End If
End Sub

Private Sub Button5_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Button5.Click
    Dim ETIQUETATENIA As Integer
    Dim i As Integer
    Dim respCorrecta As String = ""
    'VERIFICA CUAL ES LA RESPUESTA CORRECTA DE LA PREGUNTA EN
DB

    For i = 0 To 5
        If (DataSetDinamica.Tables(0).Rows(i)("pregunta") =
Label5.Text) Then
            respCorrecta =
DataSetDinamica.Tables(0).Rows(i)("respuesta")
        End If
    Next

    ETIQUETATENIA = ETIQUETA
    TOMARESPUESTA = 0
    ETIQUETA = 0

    'VERIFICA CUAL DE TODAS LAS RESPUESTAS RESPONDE A LA
PREGUNTA 1, SI LA RESPUESTA SELECCIONADA ES ZLA CORRECTA,
'LA DEJA EN LA POSICION QUE RESPONDE A LA PREGUNTA; SI NO
ES ASI, LA REGRESA A SU LUGAR DE ORIGEN

    If ETIQUETATENIA = 7 Then
        If (respCorrecta.ToString <> (Label7.Text)) Then
            Label7.Location = New Point(35, 495)
        Else
            buenas = buenas + 1
        End If
    End If

    If ETIQUETATENIA = 8 Then
        If (respCorrecta.ToString <> (Label8.Text)) Then
            Label8.Location = New Point(35, 530)
        Else
            buenas = buenas + 1
        End If
    End If

```

```

Else
    buenas = buenas + 1
End If
End If
End If
If ETIQUETATENIA = 9 Then
    If (respCorrecta.ToString <> (Label9.Text)) Then
        Label9.Location = New Point(35, 571)
    Else
        buenas = buenas + 1
    End If
End If
If ETIQUETATENIA = 10 Then
    If (respCorrecta.ToString <> (Label10.Text)) Then
        Label10.Location = New Point(35, 610)
    Else
        buenas = buenas + 1
    End If
End If
If ETIQUETATENIA = 11 Then
    If (respCorrecta.ToString <> (Label11.Text)) Then
        Label11.Location = New Point(35, 656)
    Else
        buenas = buenas + 1
    End If
End If
If ETIQUETATENIA = 12 Then
    If (respCorrecta.ToString <> (Label12.Text)) Then
        Label12.Location = New Point(35, 697)
    Else
        buenas = buenas + 1
    End If
End If
End Sub

Private Sub Button6_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button6.Click
    Dim ETIQUETATENIA As Integer
    Dim i As Integer
    Dim respCorrecta As String = ""
    'VERIFICA CUAL ES LA RESPUESTA CORRECTA DE LA PREGUNTA EN
    DB

    For i = 0 To 5
        If (DataSetDinamica.Tables(0).Rows(i)("pregunta") =
        Label6.Text) Then
            respCorrecta =
            DataSetDinamica.Tables(0).Rows(i)("respuesta")
        End If
    Next

    ETIQUETATENIA = ETIQUETA
    TOMARESPUESTA = 0
    ETIQUETA = 0

    'VERIFICA CUAL DE TODAS LAS RESPUESTAS RESPONDE A LA
    PREGUNTA 1, SI LA RESPUESTA SELECCIONADA ES ZLA CORRECTA,
    'LA DEJA EN LA POSICION QUE RESPONDE A LA PREGUNTA; SI NO
    ES ASI, LA REGRESA A SU LUGAR DE ORIGEN

    If ETIQUETATENIA = 7 Then
        If (respCorrecta.ToString <> (Label7.Text)) Then
            Label7.Location = New Point(35, 495)
        Else
            buenas = buenas + 1
        End If
    End If

    If ETIQUETATENIA = 8 Then
        If (respCorrecta.ToString <> (Label8.Text)) Then
            Label8.Location = New Point(35, 530)
        Else
            buenas = buenas + 1
        End If
    End If

    If ETIQUETATENIA = 9 Then
        If (respCorrecta.ToString <> (Label9.Text)) Then
            Label9.Location = New Point(35, 571)
        Else
            buenas = buenas + 1
        End If
    End If

    If ETIQUETATENIA = 10 Then
        If (respCorrecta.ToString <> (Label10.Text)) Then
            Label10.Location = New Point(35, 610)
        Else
            buenas = buenas + 1
        End If
    End If

    If ETIQUETATENIA = 11 Then
        If (respCorrecta.ToString <> (Label11.Text)) Then
            Label11.Location = New Point(35, 656)
        Else
            buenas = buenas + 1
        End If
    End If

    If ETIQUETATENIA = 12 Then
        If (respCorrecta.ToString <> (Label12.Text)) Then
            Label12.Location = New Point(35, 697)
        Else
            buenas = buenas + 1
        End If
    End If
End Sub

```

```

Private Sub Button13_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

    'VERIFICA SI DESEA SALIR DEL JUEGO, SI ES ASI, SALE, EN
    CASO CONTRARIO, MUESTRA EL JUEGO NUEVAMENTE

    If MessageBox.Show("¿Esta seguro de salir del juego?",
    "Salida", _
        MessageBoxButtons.OKCancel,
        MessageBoxIcon.Exclamation) = DialogResult.OK Then
        FPrincipalJuega.Show()
        Me.Hide()
    End If
End Sub

Private Sub AyudaToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles AyudaToolStripMenuItem.Click

    'MUESTRA INSTRUCCIONES
    FInstruccionJalaRespuesta.Show()

End Sub

Private Sub SalirToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles SalirToolStripMenuItem.Click

    'VERIFICA SI DESEA SALIR DEL JUEGO, SI ES ASI, SALE, EN
    CASO CONTRARIO, MUESTRA EL JUEGO NUEVAMENTE

    If MessageBox.Show("¿Esta seguro de salir del juego?",
    "Salida", _
        MessageBoxButtons.OKCancel, MessageBoxIcon.Exclamation) =
    DialogResult.OK Then
        FPrincipalJuega.Show()
        Me.Close()
    End If
End Sub
End Class

```

FORMULARIO DINAMICA MEMORAMA

```

Public Class FMemorama6P
    Dim oconexion As New Conexion
    Dim NumBoton As Integer
    Dim bandera As Integer = 1
    Dim enunciado1 As String
    Dim enunciado2 As String
    Dim numBoton1 As Integer
    Dim numBoton2 As Integer
    Dim DataSetDinamica As DataSet
    Dim ParesCorrectos As Integer = 0
    Dim termina As Integer = 0

    Private Sub FMemorama6P_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
        Dim dinamica As Integer
        Dim cadena As String
        Dim DataSetImagen As DataSet
        Dim i As Integer
        Dim imagenBoton As String
        Dim value As Integer
        Dim Preguntas As New ArrayList()
        Dim Respuestas As New ArrayList()
        Dim cuestionario As New ArrayList()

        dinamica = FPrincipalJuega.claveDinamica
        '*****Pon imagen a tarjeta*****
        Try
            cadena = "select dnm ImagenRuta as imagen from dinamica
            where dnm_claveint = " & dinamica
            DataSetImagen = oconexion.GetRows(cadena)
        Catch ex As Exception
            MsgBox(ex.Message)
        End Try

        imagenBoton = DataSetImagen.Tables(0).Rows(0)("Imagen")

        Button1.Image = Image.FromFile(imagenBoton)
        Button2.Image = Image.FromFile(imagenBoton)
        Button3.Image = Image.FromFile(imagenBoton)
        Button4.Image = Image.FromFile(imagenBoton)
        Button5.Image = Image.FromFile(imagenBoton)
        Button6.Image = Image.FromFile(imagenBoton)
        Button7.Image = Image.FromFile(imagenBoton)
        Button8.Image = Image.FromFile(imagenBoton)
        Button9.Image = Image.FromFile(imagenBoton)
        Button10.Image = Image.FromFile(imagenBoton)
        Button11.Image = Image.FromFile(imagenBoton)
        Button12.Image = Image.FromFile(imagenBoton)

        '****Llena cara 2 de tarjetas ****
        Try
            cadena = "select cue pregunta as pregunta,
            cue_respuesta as respuesta from Cuestionario where dnm_claveint =
            " & dinamica & " and cue_pactiva = 1 " & " order by cue_claveint"
            DataSetDinamica = oconexion.GetRows(cadena)
        Catch ex As Exception
            MsgBox(ex.Message)
        End Try
    End Sub
End Class

```

```

For i = 0 To 5
    cuestionario.Add(DataSetDinamica.Tables(0).Rows(i) ("Pregunta"))
    cuestionario.Add(DataSetDinamica.Tables(0).Rows(i) ("Respuesta"))
    Next

    'boton 1
    Randomize()
    value = CInt(Int(((cuestionario.Count - 1) * Rnd() + 1))
'12 por el num. de preguntas + el num de respuestas
RichTextBox1.Text = cuestionario(value)
cuestionario.RemoveAt(value)
'boton 2
Randomize()
value = CInt(Int(((cuestionario.Count - 1) * Rnd() + 1))
'12 por el num. de preguntas + el num de respuestas
RichTextBox2.Text = cuestionario(value)
cuestionario.RemoveAt(value)

'boton 3
Randomize()
value = CInt(Int(((cuestionario.Count - 1) * Rnd() + 1))
'12 por el num. de preguntas + el num de respuestas
RichTextBox3.Text = cuestionario(value)
cuestionario.RemoveAt(value)

'boton 4
Randomize()
value = CInt(Int(((cuestionario.Count - 1) * Rnd() + 1))
'12 por el num. de preguntas + el num de respuestas
RichTextBox4.Text = cuestionario(value)
cuestionario.RemoveAt(value)

'boton 5
Randomize()
value = CInt(Int(((cuestionario.Count - 1) * Rnd() + 1))
'12 por el num. de preguntas + el num de respuestas
RichTextBox5.Text = cuestionario(value)
cuestionario.RemoveAt(value)

'boton 6
Randomize()
value = CInt(Int(((cuestionario.Count - 1) * Rnd() + 1))
'12 por el num. de preguntas + el num de respuestas
RichTextBox6.Text = cuestionario(value)
cuestionario.RemoveAt(value)

'boton 7
Randomize()
value = CInt(Int(((cuestionario.Count - 1) * Rnd() + 1))
'12 por el num. de preguntas + el num de respuestas
RichTextBox7.Text = cuestionario(value)
cuestionario.RemoveAt(value)

'boton 8
Randomize()
value = CInt(Int(((cuestionario.Count - 1) * Rnd() + 1))
'12 por el num. de preguntas + el num de respuestas
RichTextBox8.Text = cuestionario(value)
cuestionario.RemoveAt(value)

'boton 9
Randomize()
value = CInt(Int(((cuestionario.Count - 1) * Rnd() + 1))
'12 por el num. de preguntas + el num de respuestas
RichTextBox9.Text = cuestionario(value)
cuestionario.RemoveAt(value)

'boton 10
Randomize()
value = CInt(Int(((cuestionario.Count - 1) * Rnd() + 1))
'12 por el num. de preguntas + el num de respuestas
RichTextBox10.Text = cuestionario(value)
cuestionario.RemoveAt(value)

'boton 11
Randomize()
value = CInt(Int(((cuestionario.Count - 1) * Rnd() + 1))
'12 por el num. de preguntas + el num de respuestas
RichTextBox11.Text = cuestionario(value)
cuestionario.RemoveAt(value)

'boton 12
RichTextBox12.Text = cuestionario(0)

End Sub

Private Sub Button1_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) _
    ' Initialize the random-number generator.
    Randomize()
    ' Generate random value between 1 and 6.
    Dim value As Integer = CInt(Int((12 * Rnd() + 1)) '12 por
el num. de preguntas + el num de respuestas
    'Frueba.Show()
End Sub

Private Sub Button1_Click_1(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click
    'LA VARIABLES BANDERA INDICA SI ES LA PRIMERA CARTA O LA
SEGUNDA ELEGIDA POREL USUARIO,
'DE ACUERDO A ESTO, SE GUARDA LA PREGUNTA O RESPUESTA,
SEGUN CORRESPONDA
'PARA POSTERIORMENTE SER EVALUADA

Select Case bandera
    Case 1
        bandera = bandera + 1
        enunciado1 = RichTextBox1.Text
        RichTextBox1.Visible = True
        numBoton1 = 1

    Case 2
        bandera = bandera + 1
        enunciado2 = RichTextBox1.Text

RichTextBox1.Visible = True
numBoton2 = 1

    Case 3
        RichTextBox1.Visible = True
        numBoton2 = 1

End Select

End Sub

Private Sub Button2_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Button2.Click
    'LA VARIABLES BANDERA INDICA SI ES LA PRIMERA CARTA O LA
SEGUNDA ELEGIDA POREL USUARIO,
'DE ACUERDO A ESTO, SE GUARDA LA PREGUNTA O RESPUESTA,
SEGUN CORRESPONDA
'PARA POSTERIORMENTE SER EVALUADA

Select Case bandera
    Case 1
        bandera = bandera + 1
        enunciado1 = RichTextBox2.Text
        RichTextBox2.Visible = True
        numBoton1 = 2

    Case 2
        bandera = bandera + 1
        enunciado2 = RichTextBox2.Text
        RichTextBox2.Visible = True
        numBoton2 = 2

    Case 3
        RichTextBox2.Visible = True
        numBoton2 = 2

End Select

End Sub

Private Sub Button3_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Button3.Click
    'LA VARIABLES BANDERA INDICA SI ES LA PRIMERA CARTA O LA
SEGUNDA ELEGIDA POREL USUARIO,
'DE ACUERDO A ESTO, SE GUARDA LA PREGUNTA O RESPUESTA,
SEGUN CORRESPONDA
'PARA POSTERIORMENTE SER EVALUADA

Select Case bandera
    Case 1
        bandera = bandera + 1
        enunciado1 = RichTextBox3.Text
        RichTextBox3.Visible = True
        numBoton1 = 3

    Case 2
        bandera = bandera + 1
        enunciado2 = RichTextBox3.Text
        RichTextBox3.Visible = True
        numBoton2 = 3

    Case 3
        RichTextBox3.Visible = True
        numBoton2 = 3

End Select

End Sub

Private Sub Button4_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Button4.Click
    'LA VARIABLES BANDERA INDICA SI ES LA PRIMERA CARTA O LA
SEGUNDA ELEGIDA POREL USUARIO,
'DE ACUERDO A ESTO, SE GUARDA LA PREGUNTA O RESPUESTA,
SEGUN CORRESPONDA
'PARA POSTERIORMENTE SER EVALUADA

Select Case bandera
    Case 1
        bandera = bandera + 1
        enunciado1 = RichTextBox4.Text
        RichTextBox4.Visible = True
        numBoton1 = 4

    Case 2
        bandera = bandera + 1
        enunciado2 = RichTextBox4.Text
        RichTextBox4.Visible = True
        numBoton2 = 4

    Case 3
        RichTextBox4.Visible = True
        numBoton2 = 4

End Select

End Sub

Private Sub Button5_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Button5.Click
    'LA VARIABLES BANDERA INDICA SI ES LA PRIMERA CARTA O LA
SEGUNDA ELEGIDA POREL USUARIO,
'DE ACUERDO A ESTO, SE GUARDA LA PREGUNTA O RESPUESTA,
SEGUN CORRESPONDA
'PARA POSTERIORMENTE SER EVALUADA

Select Case bandera
    Case 1
        bandera = bandera + 1
        enunciado1 = RichTextBox5.Text
        RichTextBox5.Visible = True
        numBoton1 = 5

    Case 2
        bandera = bandera + 1
        enunciado2 = RichTextBox5.Text
        RichTextBox5.Visible = True
        numBoton2 = 5

    Case 3
        RichTextBox5.Visible = True
        numBoton2 = 5

End Select

End Sub

```

```

Private Sub Button6_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Button6.Click
    'LA VARIABLES BANDERA INDICA SI ES LA PRIMERA CARTA O LA
SEGUNDA ELEGIDA POREL USUARIO,
'DE ACUERDO A ESTO, SE GUARDA LA PREGUNTA O RESPUESTA,
SEGUN CORRESPONDA
'PARA POSTERIORMENTE SER EVALUADA

    Select Case bandera
        Case 1
            bandera = bandera + 1
            enunciado1 = RichTextBox6.Text
            RichTextBox6.Visible = True
            numBoton1 = 6

        Case 2
            bandera = bandera + 1
            enunciado2 = RichTextBox6.Text
            RichTextBox6.Visible = True
            numBoton2 = 6

        Case 3

    End Select
End Sub

Private Sub Button7_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Button7.Click
    'LA VARIABLES BANDERA INDICA SI ES LA PRIMERA CARTA O LA
SEGUNDA ELEGIDA POREL USUARIO,
'DE ACUERDO A ESTO, SE GUARDA LA PREGUNTA O RESPUESTA,
SEGUN CORRESPONDA
'PARA POSTERIORMENTE SER EVALUADA

    Select Case bandera
        Case 1
            bandera = bandera + 1
            enunciado1 = RichTextBox7.Text
            RichTextBox7.Visible = True
            numBoton1 = 7

        Case 2
            bandera = bandera + 1
            enunciado2 = RichTextBox7.Text
            RichTextBox7.Visible = True
            numBoton2 = 7

        Case 3

    End Select
End Sub

Private Sub Button8_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Button8.Click
    'LA VARIABLES BANDERA INDICA SI ES LA PRIMERA CARTA O LA
SEGUNDA ELEGIDA POREL USUARIO,
'DE ACUERDO A ESTO, SE GUARDA LA PREGUNTA O RESPUESTA,
SEGUN CORRESPONDA
'PARA POSTERIORMENTE SER EVALUADA

    Select Case bandera
        Case 1
            bandera = bandera + 1
            enunciado1 = RichTextBox8.Text
            RichTextBox8.Visible = True
            numBoton1 = 8

        Case 2
            bandera = bandera + 1
            enunciado2 = RichTextBox8.Text
            RichTextBox8.Visible = True
            numBoton2 = 8

        Case 3

    End Select
End Sub

Private Sub Button12_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Button12.Click
    'LA VARIABLES BANDERA INDICA SI ES LA PRIMERA CARTA O LA
SEGUNDA ELEGIDA POREL USUARIO,
'DE ACUERDO A ESTO, SE GUARDA LA PREGUNTA O RESPUESTA,
SEGUN CORRESPONDA
'PARA POSTERIORMENTE SER EVALUADA

    Select Case bandera
        Case 1
            bandera = bandera + 1
            enunciado1 = RichTextBox12.Text
            RichTextBox12.Visible = True
            numBoton1 = 12

        Case 2
            bandera = bandera + 1
            enunciado2 = RichTextBox12.Text
            RichTextBox12.Visible = True
            numBoton2 = 12

        Case 3

    End Select
End Sub

Private Sub Button11_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Button11.Click
    'LA VARIABLES BANDERA INDICA SI ES LA PRIMERA CARTA O LA
SEGUNDA ELEGIDA POREL USUARIO,
'DE ACUERDO A ESTO, SE GUARDA LA PREGUNTA O RESPUESTA,
SEGUN CORRESPONDA
'PARA POSTERIORMENTE SER EVALUADA

    Select Case bandera
        Case 1
            bandera = bandera + 1
            enunciado1 = RichTextBox11.Text
            RichTextBox11.Visible = True
            numBoton1 = 11

        Case 2
            bandera = bandera + 1
            enunciado2 = RichTextBox11.Text
            RichTextBox11.Visible = True
            numBoton2 = 11

        Case 3

    End Select
End Sub

Private Sub Button10_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Button10.Click
    'LA VARIABLES BANDERA INDICA SI ES LA PRIMERA CARTA O LA
SEGUNDA ELEGIDA POREL USUARIO,
'DE ACUERDO A ESTO, SE GUARDA LA PREGUNTA O RESPUESTA,
SEGUN CORRESPONDA
'PARA POSTERIORMENTE SER EVALUADA

    Select Case bandera
        Case 1
            bandera = bandera + 1
            enunciado1 = RichTextBox10.Text
            RichTextBox10.Visible = True
            numBoton1 = 10

        Case 2
            bandera = bandera + 1
            enunciado2 = RichTextBox10.Text
            RichTextBox10.Visible = True
            numBoton2 = 10

        Case 3

    End Select
End Sub

Private Sub Button9_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Button9.Click
    'LA VARIABLES BANDERA INDICA SI ES LA PRIMERA CARTA O LA
SEGUNDA ELEGIDA POREL USUARIO,
'DE ACUERDO A ESTO, SE GUARDA LA PREGUNTA O RESPUESTA,
SEGUN CORRESPONDA
'PARA POSTERIORMENTE SER EVALUADA

    Select Case bandera
        Case 1
            bandera = bandera + 1
            enunciado1 = RichTextBox9.Text
            RichTextBox9.Visible = True
            numBoton1 = 9

        Case 2
            bandera = bandera + 1
            enunciado2 = RichTextBox9.Text
            RichTextBox9.Visible = True
            numBoton2 = 9

        Case 3

    End Select
End Sub

Private Sub Button13_Click_1(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button13.Click

    Dim PregRespCorrecta As Integer
    Dim EnunciadoTipo As String
    Dim i As Integer

    PregRespCorrecta = 0
    EnunciadoTipo = ""
    'VERIFICA SI LA CARTA ELEGIDA ES UNA PREGUNTA; ES CASO
VERDADERO ENTONCES VERIFICA CUAL DEBE SER SU RESPUESTA

    For i = 0 To 5
        If enunciado1 =
DataSetDinamica.Tables(0).Rows(i)("Pregunta") Then
            If enunciado2 =
DataSetDinamica.Tables(0).Rows(i)("Respuesta") Then
                PregRespCorrecta = 1
                ParesCorrectos = ParesCorrectos + 1
            End If
        Else
            EnunciadoTipo = "R"
        End If
    Next
    'VERIFICA SI LA CARTA ELEGIDA ES UNA RESPUESTA; ES CASO
VERDADERO ENTONCES VERIFICA CUAL DEBE SER SU PREGUNTA

    If EnunciadoTipo = "R" Then
        For i = 0 To 5
            If enunciado1 =
DataSetDinamica.Tables(0).Rows(i)("Respuesta") Then
                If enunciado2 =
DataSetDinamica.Tables(0).Rows(i)("Pregunta") Then
                    PregRespCorrecta = 1
                    ParesCorrectos = ParesCorrectos + 1
                End If
            End If
        Next
    End If

    'SI TODOS LOS PARES HAN SIDO CONTESTADOS CORRECTAMENTE
ENTONCES MANDA MENSAJE DE FELICITACION

```

```

If PregRespCorrecta = 1 Then
    lblMensaje.Text = "Respuesta Correcta !!!! "
    hf.Visible = True
    uf.Visible = False
    'SI TODOS LOS PARES HAN SIDO CONTESTADOS CORRECTAMENTE
ENTONCES MANDA MENSAJE DE FELICITACION

If ParesCorrectos = 6 Then
    lblMensaje.Text = "Felicidades GANADOR !!!"
    termina = 1
    hf.Visible = True
    uf.Visible = False
End If

'EN CASO DE SER INCORRECTO EL PAR DE CARTAS, MANDA
MENSAJE DE RESPUESTA INCORRECTA

ElseIf ((PregRespCorrecta = 0) And (termina = 0)) Then
    lblMensaje.Text = "Respuesta Incorrecta"
    hf.Visible = False
    uf.Visible = True
    'SI EL PAR DE CARTAS ES INCORRECTO, DEJA BOCA ABAJO LA
CARTA NUEVAMENTE

Select Case numBoton1
    Case 1
        RichTextBox1.Visible = False
    Case 2
        RichTextBox2.Visible = False
    Case 3
        RichTextBox3.Visible = False
    Case 4
        RichTextBox4.Visible = False
    Case 5
        RichTextBox5.Visible = False
    Case 6
        RichTextBox6.Visible = False
    Case 7
        RichTextBox7.Visible = False
    Case 8
        RichTextBox8.Visible = False
    Case 9
        RichTextBox9.Visible = False
    Case 10
        RichTextBox10.Visible = False
    Case 11
        RichTextBox11.Visible = False
    Case 12
        RichTextBox12.Visible = False
End Select

Select Case numBoton2
    Case 1
        RichTextBox1.Visible = False
    Case 2
        RichTextBox2.Visible = False
    Case 3
        RichTextBox3.Visible = False
    Case 4
        RichTextBox4.Visible = False
    Case 5
        RichTextBox5.Visible = False
    Case 6
        RichTextBox6.Visible = False
    Case 7
        RichTextBox7.Visible = False
    Case 8
        RichTextBox8.Visible = False
    Case 9
        RichTextBox9.Visible = False
    Case 10
        RichTextBox10.Visible = False
    Case 11
        RichTextBox11.Visible = False
    Case 12
        RichTextBox12.Visible = False
End Select

'LIMPIA VAIABLES
bandera = 1
numBoton1 = 0
numBoton2 = 0
enunciado1 = ""
enunciado2 = ""
PregRespCorrecta = 0
EnunciadoTipo = ""

End Sub

Private Sub Button26_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs)

If MessageBox.Show("¿Esta seguro de salir del juego?",
"Salida", _
    MessageBoxButtons.OKCancel,
    MessageBoxIcon.Exclamation) = DialogResult.OK Then
    FPrincipalJuega.Show()
    Me.Hide()
End If
End Sub

Private Sub AyudaToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
AyudaToolStripMenuItem.Click
    FInstruccionMemorama.Show()
End Sub

Private Sub SalirToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
SalirToolStripMenuItem.Click

'VERIFICA SI DESEA SALIR DEL JUEGO, SI ES ASI, SALE, EN
CASO CONTRARIO, MUESTRA EL JUEGO NUEVAMENTE

If MessageBox.Show("¿Esta seguro de salir del juego?",
"Salida", _
    MessageBoxButtons.OKCancel,
    MessageBoxIcon.Exclamation) = DialogResult.OK Then
    FPrincipalJuega.Show()
    Me.Close()
End If
End Sub

FORMULARIO DINAMICA OPCION MULTIPLE

Public Class FOpcionMultiple6P

    Dim oconexion As New Conexion
    Dim NumBoton As Integer
    Dim bandera As Integer = 1
    Dim enunciado1 As String
    Dim enunciado2 As String
    Dim numBoton1 As Integer
    Dim numBoton2 As Integer
    Dim DataSetDinamica As DataSet
    Dim ParesCorrectos As Integer = 0
    Dim termina As Integer = 0
    Dim Respuestas As New ArrayList()

    Private Sub FOpcionMultiple6P_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        Dim dinamica As Integer
        Dim cadena As String
        Dim i As Integer
        Dim value As Integer
        Dim Preguntas As New ArrayList()

        Dim puente As New ArrayList()
        Dim tresResp As New ArrayList()

        dinamica = FPrincipalJuega.claveDinamica
        '""LENA TABLERO DE CUESTIONARIO ""

        Try
            cadena = "select cue_pregunta as pregunta,
cue_respuesta as respuesta from cuestionario where dnm_claveint =
" & dinamica & " and cue_pactiva = 1 " & " order by cue_claveint"
            DataSetDinamica = oconexion.GetRows(cadena)
        Catch ex As Exception
            MsgBox(ex.Message)
        End Try

        Label1.Text = DataSetDinamica.Tables(0).Rows(0) ("Pregunta")
        Label2.Text = DataSetDinamica.Tables(0).Rows(1) ("Pregunta")
        Label3.Text = DataSetDinamica.Tables(0).Rows(2) ("Pregunta")
        Label4.Text = DataSetDinamica.Tables(0).Rows(3) ("Pregunta")
        Label5.Text = DataSetDinamica.Tables(0).Rows(4) ("Pregunta")
        Label6.Text = DataSetDinamica.Tables(0).Rows(5) ("Pregunta")
        'AGREGA EN UN ARREGLO LAS RESPUESTAS
        For i = 0 To 5

            Respuestas.Add(DataSetDinamica.Tables(0).Rows(i) ("Respuesta"))
            Next

            '-----
            'AGREGA EN UN ARREGLO LAS RESPUESTAS
            For i = 0 To 5
                puente.Add(Respuestas.Item(i))
            Next
            '-----

            tresResp.Add(DataSetDinamica.Tables(0).Rows(0) ("Respuesta"))
            puente.RemoveAt(0)

            Randomize()
            value = CInt(Int(((puente.Count - 1) * Rnd()) + 1))

            tresResp.add(puente(value))
            puente.RemoveAt(value)

            Randomize()
            value = CInt(Int(((puente.Count - 1) * Rnd()) + 1))
            tresResp.Add(puente(value))

            '-----
            Randomize()
            value = CInt(Int(((tresResp.Count - 1) * Rnd()) + 1))
            RadioButton1.Text = tresResp(value)
            tresResp.RemoveAt(value)

            Randomize()
            value = CInt(Int(((tresResp.Count - 1) * Rnd()) + 1))
            RadioButton2.Text = tresResp(value)
            tresResp.RemoveAt(value)

            RadioButton3.Text = tresResp(0)

            '-----
            'respuesta de preg. dos
            '-----
            puente.Clear()
            tresResp.Clear()

```

```

For i = 0 To 5
    puente.Add(Respuestas.Item(i))
Next
'-----
tresResp.Add(DataSetDinamica.Tables(0).Rows(1)("Respuesta"))
'cambiar el num de pregunta
puente.RemoveAt(1) 'eliminar la respuesta correcta de puente

Randomize()
value = CInt(Int(((puente.Count - 1) * Rnd()) + 1))
tresResp.Add(puente(value))
puente.RemoveAt(value)

Randomize()
value = CInt(Int(((puente.Count - 1) * Rnd()) + 1))
tresResp.Add(puente(value))

'-----
Randomize()
value = CInt(Int(((tresResp.Count - 1) * Rnd()) + 1))
RadioButton4.Text = tresResp(value) 'cambiar num de radio
button tresResp.RemoveAt(value)

Randomize()
value = CInt(Int(((tresResp.Count - 1) * Rnd()) + 1))
RadioButton5.Text = tresResp(value) 'cambiar num de radio
button tresResp.RemoveAt(value)

button RadioButton6.Text = tresResp(0) 'cambiar num de radio

'-----
'respuesta de pregunta tres
'-----
puente.Clear()
tresResp.Clear()

For i = 0 To 5
    puente.Add(Respuestas.Item(i))
Next
'-----

tresResp.Add(DataSetDinamica.Tables(0).Rows(2)("Respuesta"))
'cambiar el num de pregunta
puente.RemoveAt(2) 'eliminar la respuesta correcta de puente

Randomize()
value = CInt(Int(((puente.Count - 1) * Rnd()) + 1))

tresResp.Add(puente(value))
puente.RemoveAt(value)

Randomize()
value = CInt(Int(((puente.Count - 1) * Rnd()) + 1))
tresResp.Add(puente(value))

'-----
Randomize()
value = CInt(Int(((tresResp.Count - 1) * Rnd()) + 1))
RadioButton7.Text = tresResp(value) 'cambiar num de radio
button tresResp.RemoveAt(value)

Randomize()
value = CInt(Int(((tresResp.Count - 1) * Rnd()) + 1))
RadioButton8.Text = tresResp(value) 'cambiar num de radio
button tresResp.RemoveAt(value)

button RadioButton9.Text = tresResp(0) 'cambiar num de radio

'-----
'resp a pregunta cuatro
'-----
puente.Clear()
tresResp.Clear()

For i = 0 To 5
    puente.Add(Respuestas.Item(i))
Next
'-----

tresResp.Add(DataSetDinamica.Tables(0).Rows(3)("Respuesta"))
'cambiar el num de pregunta
puente.RemoveAt(3) 'eliminar la respuesta correcta de puente

Randomize()
value = CInt(Int(((puente.Count - 1) * Rnd()) + 1))

tresResp.Add(puente(value))
puente.RemoveAt(value)

Randomize()
value = CInt(Int(((puente.Count - 1) * Rnd()) + 1))
tresResp.Add(puente(value))

'-----
Randomize()
value = CInt(Int(((tresResp.Count - 1) * Rnd()) + 1))
RadioButton10.Text = tresResp(value) 'cambiar num de radio
button tresResp.RemoveAt(value)

Randomize()
value = CInt(Int(((tresResp.Count - 1) * Rnd()) + 1))

button RadioButton11.Text = tresResp(value) 'cambiar num de radio
tresResp.RemoveAt(value)

RadioButton12.Text = tresResp(0) 'cambiar num de radio
button

'-----
'resp. a pregunta cinco
'-----
puente.Clear()
tresResp.Clear()

For i = 0 To 5
    puente.Add(Respuestas.Item(i))
Next
'-----

tresResp.Add(DataSetDinamica.Tables(0).Rows(4)("Respuesta"))
'cambiar el num de pregunta
puente.RemoveAt(4) 'eliminar la respuesta correcta de puente

Randomize()
value = CInt(Int(((puente.Count - 1) * Rnd()) + 1))

tresResp.Add(puente(value))
puente.RemoveAt(value)

Randomize()
value = CInt(Int(((puente.Count - 1) * Rnd()) + 1))
tresResp.Add(puente(value))

'-----
Randomize()
value = CInt(Int(((tresResp.Count - 1) * Rnd()) + 1))
RadioButton13.Text = tresResp(value) 'cambiar num de radio
button tresResp.RemoveAt(value)

Randomize()
value = CInt(Int(((tresResp.Count - 1) * Rnd()) + 1))
RadioButton14.Text = tresResp(value) 'cambiar num de radio
button tresResp.RemoveAt(value)

button RadioButton15.Text = tresResp(0) 'cambiar num de radio

'-----
'resp. a pregunta seis
'-----
puente.Clear()
tresResp.Clear()

For i = 0 To 5
    puente.Add(Respuestas.Item(i))
Next
'-----

tresResp.Add(DataSetDinamica.Tables(0).Rows(5)("Respuesta"))
'cambiar el num de pregunta
puente.RemoveAt(5) 'eliminar la respuesta correcta de puente

Randomize()
value = CInt(Int(((puente.Count - 1) * Rnd()) + 1))

tresResp.Add(puente(value))
puente.RemoveAt(value)

Randomize()
value = CInt(Int(((puente.Count - 1) * Rnd()) + 1))
tresResp.Add(puente(value))

'-----
Randomize()
value = CInt(Int(((tresResp.Count - 1) * Rnd()) + 1))
RadioButton16.Text = tresResp(value) 'cambiar num de radio
button tresResp.RemoveAt(value)

Randomize()
value = CInt(Int(((tresResp.Count - 1) * Rnd()) + 1))
RadioButton17.Text = tresResp(value) 'cambiar num de radio
button tresResp.RemoveAt(value)

button RadioButton18.Text = tresResp(0) 'cambiar num de radio

'-----
End Sub

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Dim respAlum As String
    Dim puntosBuenos As Integer
    Dim puntosMalos As Integer
    puntosBuenos = 0
    puntosMalos = 0
    respAlum = ""

    'ESCONDE LAS IMAGENES DE CORRECTO (PALOMITA) E
    INCORRECTO (TACHE)
    PictureBox1.Visible = False
    PictureBox2.Visible = False
    PictureBox3.Visible = False
    PictureBox4.Visible = False
    PictureBox5.Visible = False
    PictureBox6.Visible = False
    PictureBox7.Visible = False
    PictureBox8.Visible = False

```

```

PictureBox9.Visible = False
PictureBox10.Visible = False
PictureBox11.Visible = False
PictureBox12.Visible = False

'VERIFICA CUAL FUE LA RESPUESTA ELEGIDA POR EL USUARIO EN
CADA PREGUNTA
'Y LA DEJA EN UNA VARIABLE LLAMADA respAlum

'resp 1
If RadioButton1.Checked = True Then
    respAlum = RadioButton1.Text
End If
If RadioButton2.Checked = True Then
    respAlum = RadioButton2.Text
End If
If RadioButton3.Checked = True Then
    respAlum = RadioButton3.Text
End If

Then
If ((Respuestas.Item(0).ToString) = (respAlum.ToString))

    puntosBuenos = puntosBuenos + 1
    PictureBox1.Visible = True
Else
    PictureBox2.Visible = True
    puntosMalos = puntosMalos + 1
End If

'resp 2
respAlum = ""
If RadioButton4.Checked = True Then
    respAlum = RadioButton4.Text
End If
If RadioButton5.Checked = True Then
    respAlum = RadioButton5.Text
End If
If RadioButton6.Checked = True Then
    respAlum = RadioButton6.Text
End If

Then
If ((Respuestas.Item(1).ToString) = (respAlum.ToString))

    puntosBuenos = puntosBuenos + 1
    PictureBox3.Visible = True
Else
    PictureBox4.Visible = True
    puntosMalos = puntosMalos + 1
End If

'resp 3
respAlum = ""
If RadioButton7.Checked = True Then
    respAlum = RadioButton7.Text
End If
If RadioButton8.Checked = True Then
    respAlum = RadioButton8.Text
End If
If RadioButton9.Checked = True Then
    respAlum = RadioButton9.Text
End If

Then
If ((Respuestas.Item(2).ToString) = (respAlum.ToString))

    puntosBuenos = puntosBuenos + 1
    PictureBox5.Visible = True
Else
    PictureBox6.Visible = True
    puntosMalos = puntosMalos + 1
End If

'resp 4
respAlum = ""
If RadioButton10.Checked = True Then
    respAlum = RadioButton10.Text
End If
If RadioButton11.Checked = True Then
    respAlum = RadioButton11.Text
End If
If RadioButton12.Checked = True Then
    respAlum = RadioButton12.Text
End If

Then
If ((Respuestas.Item(3).ToString) = (respAlum.ToString))

    puntosBuenos = puntosBuenos + 1
    PictureBox7.Visible = True
Else
    PictureBox8.Visible = True
    puntosMalos = puntosMalos + 1
End If

'resp 5
respAlum = ""
If RadioButton13.Checked = True Then
    respAlum = RadioButton13.Text
End If
If RadioButton14.Checked = True Then
    respAlum = RadioButton14.Text
End If
If RadioButton15.Checked = True Then
    respAlum = RadioButton15.Text
End If

Then
If ((Respuestas.Item(4).ToString) = (respAlum.ToString))

    puntosBuenos = puntosBuenos + 1
    PictureBox9.Visible = True
Else
    PictureBox10.Visible = True
    puntosMalos = puntosMalos + 1
End If

'resp 6
respAlum = ""
If RadioButton16.Checked = True Then
    respAlum = RadioButton16.Text
End If
If RadioButton17.Checked = True Then
    respAlum = RadioButton17.Text
End If
If RadioButton18.Checked = True Then
    respAlum = RadioButton18.Text
End If

'CUENTA EL NUMERO DE RESPUESTAS CORRECTAS E INCORRECTAS
Then
If ((Respuestas.Item(5).ToString) = (respAlum.ToString))

    puntosBuenos = puntosBuenos + 1
    PictureBox11.Visible = True
Else
    PictureBox12.Visible = True
    puntosMalos = puntosMalos + 1
End If

Label9.Text = puntosBuenos.ToString
Label10.Text = puntosMalos.ToString

GANADOR
If puntosBuenos = Respuestas.Count Then
    Label11.Text = "Muchas Felicidades Ganaste!!!!!"
End If
'DE LO CONTRARIO MANDA EL NUMERO DE RESPUESTAS CORRECTAS
If puntosBuenos < Respuestas.Count Then
    Label11.Text = "Has respondido " +
puntosBuenos.ToString + " preguntas correctamente"

End If

End Sub

Private Sub AyudaToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
AyudaToolStripMenuItem.Click

'MUESTRA INSTRUCCIONES
FinstruccionOM.Show()
End Sub

Private Sub SalirToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
SalirToolStripMenuItem.Click

'VERIFICA SI DESEA SALIR DEL JUEGO, SI ES ASI, SALE, EN
CASO CONTRARIO, MUESTRA EL JUEGO NUEVAMENTE

If MessageBox.Show("¿Esta seguro de salir del juego?",
"Salida", _
    MessageBoxButtons.OKCancel,
MessageBoxIcon.Exclamation) = DialogResult.OK Then
    FPrincipalJuega.Show()
    Me.Close()
End If
End Sub

End Class

FORMULARIO DINAMICA VERDADERO Y FALSO

Public Class FVerdaderoFalso6P
Dim oconexion As New Conexion
Dim Pregunta As String
Dim Respuesta As String
Dim bandera As Integer
Dim enunciado1 As String
Dim enunciado2 As String
Dim DataSetDinamica As DataSet
Dim ParesCorrectos As Integer = 0
Dim termina As Integer = 0
Dim numRTB1 As Integer
Dim numRTB2 As Integer
Dim TOMARESPUESTA As Integer = 0
Dim ETIQUETA As Integer = 0

Dim Preguntas As New ArrayList()
Dim Respuestas As New ArrayList()
Dim buenas As Integer

Private Sub FVerdaderoFalso6P_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
Dim dinamica As Integer
Dim cadena As String

Dim i As Integer
Dim value As Integer

Dim posibleResp1 As String
Dim posibleResp2 As String

value = 0
bandera = 1
dinamica = FPrincipalJuega.claveDinamica
""""Llena Preguntas y respuestas""""
Try
    cadena = "select cue_pregunta as pregunta,
cue_respuesta as respuesta from cuestionario where dnm_claveint =
" & dinamica & " and cue_pactiva = 1 " & " order by cue_claveint"
    DataSetDinamica = oconexion.GetRows(cadena)
Catch ex As Exception

```

```

        MsgBox(ex.Message)
    End Try

    For i = 0 To 5
        Preguntas.Add(DataSetDinamica.Tables(0).Rows(i) ("Pregunta"))
        Respuestas.Add(DataSetDinamica.Tables(0).Rows(i) ("Respuesta"))
    Next

    Label1.Text = DataSetDinamica.Tables(0).Rows(0) ("Pregunta")
    Label3.Text = DataSetDinamica.Tables(0).Rows(1) ("Pregunta")
    Label5.Text = DataSetDinamica.Tables(0).Rows(2) ("Pregunta")
    Label7.Text = DataSetDinamica.Tables(0).Rows(3) ("Pregunta")
    Label9.Text = DataSetDinamica.Tables(0).Rows(4) ("Pregunta")
    Label11.Text =
        DataSetDinamica.Tables(0).Rows(5) ("Pregunta")

    RichTextBox1.Text =
        DataSetDinamica.Tables(0).Rows(0) ("Respuesta")
    RichTextBox2.Text =
        DataSetDinamica.Tables(0).Rows(1) ("Respuesta")
    RichTextBox3.Text =
        DataSetDinamica.Tables(0).Rows(2) ("Respuesta")
    RichTextBox4.Text =
        DataSetDinamica.Tables(0).Rows(3) ("Respuesta")
    RichTextBox5.Text =
        DataSetDinamica.Tables(0).Rows(4) ("Respuesta")
    RichTextBox6.Text =
        DataSetDinamica.Tables(0).Rows(5) ("Respuesta")

    'Resp1
    Randomize()
    value = CInt(Int((Respuestas.Count - 1) * Rnd() + 1))
    posibleResp1 = Respuestas(value)
    posibleResp2 = Respuestas(0)

    Randomize()
    value = CInt(Int((2 * Rnd() + 1)))

    If value = 1 Then
        Label2.Text = posibleResp1.ToString
    Else
        Label2.Text = posibleResp2.ToString
    End If

    'Resp2
    Randomize()
    value = CInt(Int((Respuestas.Count - 1) * Rnd() + 1))
    posibleResp1 = Respuestas(value)
    posibleResp2 = Respuestas(1)

    Randomize()
    value = CInt(Int((2 * Rnd() + 1)))

    If value = 1 Then
        Label4.Text = posibleResp1.ToString
    Else
        Label4.Text = posibleResp2.ToString
    End If

    'Resp3
    Randomize()
    value = CInt(Int((Respuestas.Count - 1) * Rnd() + 1))
    posibleResp1 = Respuestas(value)
    posibleResp2 = Respuestas(2)

    Randomize()
    value = CInt(Int((2 * Rnd() + 1)))
    If value = 2 Then
        Label6.Text = posibleResp1.ToString
    Else
        Label6.Text = posibleResp2.ToString
    End If

    'Resp4
    Randomize()
    value = CInt(Int((Respuestas.Count - 1) * Rnd() + 1))
    posibleResp1 = Respuestas(value)
    posibleResp2 = Respuestas(3)

    Randomize()
    value = CInt(Int((2 * Rnd() + 1)))
    If value = 2 Then
        Label8.Text = posibleResp1.ToString
    Else
        Label8.Text = posibleResp2.ToString
    End If

    'Resp5
    Randomize()
    value = CInt(Int((Respuestas.Count - 1) * Rnd() + 1))
    posibleResp1 = Respuestas(value)
    posibleResp2 = Respuestas(4)

    Randomize()
    value = CInt(Int((2 * Rnd() + 1)))
    If value = 2 Then
        Label10.Text = posibleResp1.ToString
    Else
        Label10.Text = posibleResp2.ToString
    End If

    'Resp6
    Randomize()
    value = CInt(Int((Respuestas.Count - 1) * Rnd() + 1))
    posibleResp1 = Respuestas(value)
    posibleResp2 = Respuestas(5)

    Randomize()
    value = CInt(Int((2 * Rnd() + 1)))
    If value = 2 Then
        Label12.Text = posibleResp1.ToString
    Else
        Label12.Text = posibleResp2.ToString
    End If

    RadioButton1.Enabled = True
    RadioButton2.Enabled = True
    RadioButton3.Enabled = True
    RadioButton4.Enabled = True
    RadioButton5.Enabled = True
    RadioButton6.Enabled = True
    RadioButton7.Enabled = True
    RadioButton8.Enabled = True
    RadioButton9.Enabled = True
    RadioButton10.Enabled = True
    RadioButton11.Enabled = True
    RadioButton12.Enabled = True
    Button1.Enabled = True
End Sub

Private Sub Button26_Click(By
    'PREGUNTA SI DESEA SALIR DEL JUEGOval sender As
    System.Object, ByVal e As System.EventArgs) Handles
    Button26.Click
    If MessageBox.Show("¿Esta seguro de salir del juego?",
        "Salida", -
            MessageBoxButtons.OKCancel,
            MessageBoxIcon.Exclamation) = DialogResult.OK Then
        FPrincipalJuega.Show()
        Me.Hide()
    End If
End Sub

Private Sub AyudaToolStripMenuItem_Click(ByVal sender As
    System.Object, ByVal e As System.EventArgs) Handles
    AyudaToolStripMenuItem.Click
    'MUESTRA INSTRUCCIONES
    FInstruccionVF.Show()
End Sub

Private Sub SalirToolStripMenuItem_Click(ByVal sender As
    System.Object, ByVal e As System.EventArgs) Handles
    SalirToolStripMenuItem.Click
    'PREGUNTA SI DESEA SALIR DEL JUEGO

    If MessageBox.Show("¿Esta seguro de salir del juego?",
        "Salida", -
            MessageBoxButtons.OKCancel,
            MessageBoxIcon.Exclamation) = DialogResult.OK Then
        FPrincipalJuega.Show()
        Me.Close()
    End If
End Sub

Private Sub Button1_Click_1(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles Button1.Click
    Dim respAlum As Integer
    Dim puntosBuenos As Integer
    Dim puntosMalos As Integer
    Dim Contestadas As Integer
    Dim RespCorrecta As Integer

    Dim msg As String
    Dim title As String
    Dim style As MsgBoxStyle
    Dim response As MsgBoxResult
    puntosBuenos = 0
    puntosMalos = 0
    Contestadas = 0
    PictureBox1.Visible = False
    PictureBox2.Visible = False
    PictureBox3.Visible = False
    PictureBox4.Visible = False
    PictureBox5.Visible = False
    PictureBox6.Visible = False
    PictureBox7.Visible = False
    PictureBox8.Visible = False

    PictureBox9.Visible = False
    PictureBox10.Visible = False
    PictureBox11.Visible = False
    PictureBox12.Visible = False

    'chechar que todas las preguntas hayan sido respondidas
    If (RadioButton1.Checked = True) Or (RadioButton2.Checked =
    True) Then
        Contestadas = Contestadas + 1
    End If

    If (RadioButton3.Checked = True) Or (RadioButton4.Checked =
    True) Then
        Contestadas = Contestadas + 1
    End If

    If (RadioButton5.Checked = True) Or (RadioButton6.Checked =
    True) Then
        Contestadas = Contestadas + 1
    End If

```

```

        If (RadioButton7.Checked = True) Or (RadioButton8.Checked =
True) Then
            Contestadas = Contestadas + 1
        End If

        If (RadioButton9.Checked = True) Or (RadioButton10.Checked
= True) Then
            Contestadas = Contestadas + 1
        End If

        If (RadioButton11.Checked = True) Or (RadioButton12.Checked
= True) Then
            Contestadas = Contestadas + 1
        End If

        If Contestadas <> 6 Then
            '****
            msg = "Debes reponder Verdadero o Falso a las
Preguntas"
            ' Define message.
            style = MsgBoxStyle.DefaultButton2 Or _
MsgBoxStyle.Information Or MsgBoxStyle.OkOnly
            title = "Importante" ' Define title.
            ' Display message.
            response = MsgBox(msg, style, title)
            '****
        Else
            'resp 1
            If (DataSetDinamica.Tables(0).Rows(0) ("Respuesta")) =
Label2.Text Then
                RespCorrecta = 1
            Else
                RespCorrecta = 0
            End If

            If RadioButton1.Checked = True Then
                respAlum = 1
            End If

            If RadioButton2.Checked Then
                respAlum = 0
            End If

            If RespCorrecta = respAlum Then
                puntosBuenos = puntosBuenos + 1
                PictureBox1.Visible = True
            Else
                puntosMalos = puntosMalos + 1
                PictureBox2.Visible = True
            End If

            'resp 2

            If (DataSetDinamica.Tables(0).Rows(1) ("Respuesta")) =
Label4.Text Then
                RespCorrecta = 1
            Else
                RespCorrecta = 0
            End If

            If RadioButton3.Checked = True Then
                respAlum = 1
            End If

            If RadioButton4.Checked Then
                respAlum = 0
            End If

            If RespCorrecta = respAlum Then
                puntosBuenos = puntosBuenos + 1
                PictureBox3.Visible = True
            Else
                puntosMalos = puntosMalos + 1
                PictureBox4.Visible = True
            End If

            'resp 3

            If (DataSetDinamica.Tables(0).Rows(2) ("respuesta")) =
Label6.Text Then
                RespCorrecta = 1
            Else
                RespCorrecta = 0
            End If

            If RadioButton5.Checked = True Then
                respAlum = 1
            End If

            If RadioButton6.Checked Then
                respAlum = 0
            End If

            If RespCorrecta = respAlum Then
                puntosBuenos = puntosBuenos + 1
                PictureBox5.Visible = True
            Else
                puntosMalos = puntosMalos + 1
                PictureBox6.Visible = True
            End If

            'resp 4
            If (DataSetDinamica.Tables(0).Rows(3) ("Respuesta")) =
Label8.Text Then
                RespCorrecta = 1
            Else
                RespCorrecta = 0
            End If

            If RadioButton7.Checked = True Then
                respAlum = 1
            End If

            If RadioButton8.Checked Then
                respAlum = 0
            End If

            If RespCorrecta = respAlum Then
                puntosBuenos = puntosBuenos + 1
                PictureBox7.Visible = True
            Else
                puntosMalos = puntosMalos + 1
                PictureBox8.Visible = True
            End If

            'resp 5

            If (DataSetDinamica.Tables(0).Rows(4) ("Respuesta")) =
Label10.Text Then
                RespCorrecta = 1
            Else
                RespCorrecta = 0
            End If

            If RadioButton9.Checked = True Then
                respAlum = 1
            End If

            If RadioButton10.Checked Then
                respAlum = 0
            End If

            If RespCorrecta = respAlum Then
                puntosBuenos = puntosBuenos + 1
                PictureBox9.Visible = True
            Else
                puntosMalos = puntosMalos + 1
                PictureBox10.Visible = True
            End If

            'resp 6

            If (DataSetDinamica.Tables(0).Rows(5) ("Respuesta")) =
Label12.Text Then
                RespCorrecta = 1
            Else
                RespCorrecta = 0
            End If

            If RadioButton11.Checked = True Then
                respAlum = 1
            End If

            If RadioButton12.Checked Then
                respAlum = 0
            End If

            If RespCorrecta = respAlum Then
                puntosBuenos = puntosBuenos + 1
                PictureBox11.Visible = True
            Else
                puntosMalos = puntosMalos + 1
                PictureBox12.Visible = True
            End If

            RichTextBox1.Visible = True
            RichTextBox2.Visible = True
            RichTextBox3.Visible = True
            RichTextBox4.Visible = True
            RichTextBox5.Visible = True
            RichTextBox6.Visible = True

            Label13.Text = puntosBuenos.ToString
            'Modifica controles
            RadioButton1.Enabled = False
            RadioButton2.Enabled = False
            RadioButton3.Enabled = False
            RadioButton4.Enabled = False
            RadioButton5.Enabled = False
            RadioButton6.Enabled = False
            RadioButton7.Enabled = False
            RadioButton8.Enabled = False
            RadioButton9.Enabled = False
            RadioButton10.Enabled = False
            RadioButton11.Enabled = False
            RadioButton12.Enabled = False

            Button1.Enabled = False
        End If

    End Sub
End Class

```