

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

COLEGIO DE CIENCIAS Y HUMANIDADES
UNIDAD ACADÉMICA DE LOS CICLOS
PROFESIONALES Y POSGRADO

INSTITUTO DE INVESTIGACIONES EN MATEMÁTICAS APLICADAS
Y EN SISTEMAS

DESARROLLO DE UN SISTEMA
DE ENSEÑANZA AUXILIADO POR COMPUTADORA

QUE PARA OBTENER EL GRADO DE
MAESTRO EN CIENCIAS DE LA COMPUTACIÓN

P R E S E N T A

JORGE IVAN EUAN AVILA

ASESOR: DR. ALEJANDRO BUCHMANN SAUTER

TESIS CON
FALLA DE ORIGEN



Universidad Nacional
Autónoma de México

UNAM



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 está 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.

I N D I C E

I) Introducción.

II) Conceptos acerca de los SEAC.

II.1) Evolución de los Sistemas de Enseñanza Auxiliados por Computadora (SEAC).

II.2) Definición y clasificación de los SEAC.

II.3) Evaluación de los SEAC.

II.4) Calidad de los SEAC.

II.5) Viabilidad de los SEAC.

II.6) Hardware utilizado en los SEAC.

II.7) Futuro de los SEAC.

III) Lineamientos que se consideraron en el desarrollo.

III.1) Proceso de enseñanza aprendizaje. Modelo de Anderson y Faust.

III.2) Especificación de objetivos.

III.3) Evaluación en la sistematización.

III.4) Planeación de la instrucción.

IV) Análisis del sistema.

IV.1) Descripción general del sistema.

IV.2) Configuración de Hardware.

I N D I C E

I) Introducción.

II) Conceptos acerca de los SEAC.

II.1) Evolución de los Sistemas de Enseñanza Auxiliados por Computadora (SEAC).

II.2) Definición y clasificación de los SEAC.

II.3) Evaluación de los SEAC.

II.4) Calidad de los SEAC.

II.5) Viabilidad de los SEAC.

II.6) Hardware utilizado en los SEAC.

II.7) Futuro de los SEAC.

III) Lineamientos que se consideraron en el desarrollo.

III.1) Proceso de enseñanza aprendizaje. Modelo de Anderson y Faust.

III.2) Especificación de objetivos.

III.3) Evaluación en la sistematización.

III.4) Planeación de la instrucción.

IV) Análisis del sistema.

IV.1) Descripción general del sistema.

IV.2) Configuración de Hardware.

V) Diseño del sistema.

V.1) Notación y convenciones utilizadas.

V.2) Especificación funcional y diseño detallado del sistema.

V.2.1) Elaboración de la guía para el instructor.

V.2.2) Consulta de la guía para el instructor.

V.2.3) Elaboración de los contenidos.

V.2.4) Elaboración de preguntas o reactivos.

V.2.5) Elaboración de la guía para el alumno.

V.2.6) Consulta de los datos del alumno por el instructor.

V.2.7) Consulta de la guía para el alumno.

V.2.8) Aplicación de la instrucción.

V.2.9) Consulta de los datos del alumnos por el alumno.

VI) Discusión y Conclusiones.

APENDICES

A. Manual del instructor.

B. Manual del alumno.

C. Programas del sistema.

D. Material de ejemplo.

E. Bibliografía

I) Introducción.

En mi labor profesional he podido desarrollar entre otros campos el de la docencia universitaria. En este campo es donde he sentido la carencia de una preparación docente que permitiera desde mis inicios realizarla con mayor efectividad. Durante siete años he laborado como profesor en la Facultad de Ingeniería de la UNAM en la División de Ingeniería Mecánica y Eléctrica . En este tiempo, he buscado las alternativas que me permitan desarrollar mejor la labor docente y he podido constatar el interés que existe en la Facultad de Ingeniería por motivar a los profesores en su búsqueda.

No obstante todo este interés por mejorar la actividad docente, persisten en la Facultad problemas de índole académico.

De acuerdo a las estadísticas de los últimos años, se sabe que dentro de los dos primeros años abandonan la Facultad cerca del 50% de los alumnos que se inscriben y que de los restantes sólo el 50% se titula. De esta manera aumenta año con año, el número de alumnos que deben una o varias materias y que ya han sobrepasado el tiempo límite de siete años y medio que fija la Legislación Universitaria para ser considerados como alumnos regulares con derecho a inscripción.

Esta misma Legislación otorga a los estudiantes en esta situación tres alternativas para terminar sus estudios. La primera,

SEAC cap.I. Introducción.

aplicable a cualquier alumno, es la de presentar seis exámenes extraordinarios por semestre; la segunda válida solamente para los estudiantes que ya hayan acreditado el sesenta por ciento de los créditos totales de la carrera, es la de inscribirse en dos materias como oyentes y presentar su examen final en el tercer periodo de exámenes extraordinarios del semestre; la tercera, aplicable exclusivamente a los alumnos que ya han cubierto el servicio social y acreditado el seminario y no deben más de dos materias, es la de presentarlas en 'examen especial', en cualquier fecha.

Resultado de esta situación analizada es la existencia de un gran número de alumnos que tienen que presentar exámenes extraordinarios para la terminación de sus estudios.

Con el propósito de auxiliar al alumno la Facultad ha instrumentado guías de estudio, asesorías, programas de orientación al alumno, libros de texto, etc.

Considerando esto, se ha desarrollado un trabajo con el título Desarrollo de un Sistema de Enseñanza Auxiliado por Computadora y cuyo objetivo general es la elaboración de un material escrito acerca de la asignatura de Estructuras de Datos que sigue normas didácticas y pedagógicas más la utilización de las facilidades de los equipos de cómputo para integrar un Sistema de Enseñanza Auxiliado por Computadora (SEAC).

Con este trabajo se intenta poner una herramienta más que ayude a

SEAC cap.I. Introducción.

los alumnos a preparar sus exámenes extraordinarios y que sirva de complemento al curso normal de Estructuras de Datos.

La idea de incorporar un SEAC surge principalmente por el vínculo con el área computacional y no con el área Pedagógica, ya que si bien, he desarrollado actividades docentes no he cursado una carrera de Pedagogía. A esto obedece el que no se haya hecho un análisis más profundo para buscar una alternativa diferente al uso de una computadora y que el aspecto pedagógico sea restringido.

El trabajo se desarrolló considerando que la Facultad cuenta con equipos de cómputo que pueden ser utilizados para instalar este sistema y que ya se contaba con el material de Estructuras de Datos que serviría de prueba. Este material es utilizado actualmente en la Facultad como los apuntes de la materia. Fueron editados en 1982 por la Facultad de Ingeniería y la Coordinación del Sistema de Universidad Abierta de la UNAM, posteriormente en 1984 fueron editados por la UNAM dentro del programa del libro de texto universitario [ref 4].

El sistema terminado incorpora herramientas para el instructor que le permiten: elaborar los contenidos de las unidades, mantener un banco de preguntas o reactivos, elaborar una guía para el alumno, consultar una guía sobre el uso del sistema y consultar el desempeño de los alumnos. Para el alumno el sistema permite: inscripción al curso; controlar el avance, retroceso, etc., de cuadros; terminar una sesión, consultar la guía del

SEAC cap.I. Introducción.

alumno y consultar su desempeño.

Actualmente el sistema se encuentra a nivel de pruebas funcionales y se han elaborado cuatro unidades con sus bancos de reactivos. Los programas se desarrollaron en un computador HP-150 en las instalaciones del Centro de Investigación y Estudios Avanzados del IPN (CINVESTAV) Unidad Mérida y no ha sido puesto en uso para el ambiente que hemos descrito.

SEAC cap.II. Conceptos acerca de los SEAC

II) Conceptos acerca de los SEAC.

II.1) Evolución de los SEAC.

La idea de desarrollar una máquina para la enseñanza, se remonta a 1926; en esos tiempos el profesor Sydney L. Pressey del departamento de Psicología de la Universidad de Ohio construyó una máquina con el propósito de evaluar el aprovechamiento de sus alumnos. Esta máquina consistía de un dispositivo que presentaba exámenes de tipo Opción Múltiple y para cada una de las posibles respuestas la máquina contaba con un teclado que las asociaba. El alumno debía de oprimir la tecla que él consideraba correspondía a la respuesta; si acertaba, la máquina lo pasaba al siguiente reactivo; si erraba, se registraba su respuesta y se le daba oportunidad de apretar otra hasta que diera la respuesta correcta.

Hacia 1950, otro profesor de la Universidad de Harvard, el psicólogo Burrhus F. Skinner construyó la máquina que dió origen a la enseñanza programada. La máquina consistía de una caja con dos ventanas, una para presentarle al alumno los contenidos dentro de los cuales existía o no una pregunta y la otra para que el alumno diera su respuesta y una manivela que lo pasaba al siguiente cuadro dándole también la respuesta a la pregunta anterior.

Con la aparición de la computadora, la construcción de máquinas mecánicas para la enseñanza fue abandonada. El uso de las

SEAC cap.II. Conceptos acerca de los SEAC

computadoras como un medio de instrucción data de 1950 cuando a una IBM 650 se le conectaron varios teletipos y máquinas de escribir con el propósito de ejecutar unos programas para el entrenamiento de personal. Los programas de entrenamiento presentaban el material para que el usuario los leyera y después hacían preguntas que debían de contestarse en un formato muy elemental. La secuencia de presentación era lineal en un modo cíclico de instrucción-evaluación de pedazos cortos de material. Estos, fueron programados por los instructores en un lenguaje que les resultaba difícil de utilizar. Esto, fué lo que dió origen a que se escribieran lenguajes de programación que facilitarían la escritura de material para la enseñanza. COURSEWRITER de IBM es un ejemplo de programa con el cual los instructores pueden escribir directamente sus ideas.

En 1963 el Instituto para el Estudio de las Matemáticas en las Ciencias Sociales de la Universidad de Stanford, comenzó un programa de investigación y desarrollo de Sistemas de Enseñanza Auxiliados por Computadora (SEAC). En 1964 se presentó la versión preliminar de un curso de Matemáticas elementales el cual fue utilizado por un grupo de 41 alumnos en la modalidad de Instrucción y Práctica, para 1966 ya existían trabajando con la máquina 270 estudiantes de nivel elemental y 60 de nivel medio. Entre 1966 y 1967, también en Stanford, se desarrolló un sistema para la enseñanza en modo Tutorial; para este proyecto se utilizó una IBM 1500 y terminales de tubos de rayos catódicos (CRT) y un proyector de 16mm en cada estación de enseñanza. De

SEAC cap.II. Conceptos acerca de los SEAC

1970 a la fecha en la misma Universidad se ha puesto especial énfasis en el desarrollo de nuevos cursos y de revisar los ya existentes. Algunos de los nuevos cursos son : Lógica, Teoría de Conjuntos, Ruso, Introducción al Edlgaro, Introducción a LISP y a BASIC, y otros cursos.

En la Universidad de Illinois en 1960 se inició un proyecto con la colaboración de Control Data Corporation y la National Science Foundation llamado PLATO (Programmed Logic for Automatic Teaching Operations). Actualmente PLATO es uno de los sistemas * más importantes de EAC en el mundo y se encuentra en la quinta versión de desarrollo. Existe en los EU, una red de más de 50 ciudades en las que se ofrecen cursos para entrenamiento de personal y de instrucción a consumidores. Estos han sido utilizados en líneas aéreas, bancos, compañías de seguros entre las más importantes. En las Universidades, PLATO ha sido utilizado en Illinois, Quebec, Arizona, Colorado, Delaware, Florida State y Minnesota.

En 1972, la cooperación MITRE, C. Victor Bunderson y la Universidad Brigham Young iniciaron un proyecto llamado TICCIT (Time Shared, Interactive, Computer-Controlled Information Television), cuyo propósito es usar las minicomputadoras y la tecnología de la televisión para instrumentar programas de

*EAC (Enseñanza Asistida por Computadora)

SEAC cap.II. Conceptos acerca de los SEAC

instrucción. Actualmente TICCIT es también uno de los sistemas más importantes de EAC en el mundo.

Otra corporación importante en los EU es CCC (Computer Curriculum Corporation) la cual ofrece un equipo de cómputo de hasta 96 estaciones las cuales pueden operar simultáneamente y para el cual ya tienen una gran cantidad de cursos implementados en diferentes niveles.

Actualmente ofrecen herramientas para elaborar sistemas de EAC los siguientes fabricantes de equipos de cómputo: CDC ofrece PLATO, Sperry Univac ofrece ASET (Author System for Education and Training), HP a IDF (Instructional Dialogue Facility), IBM a Coursewriter, DEC a DECAL (Digital Equipment CAI * Author Language).

En los últimos años se han desarrollado con éste mismo propósito, lenguajes como LOGO, PILOT, CAST, etc., cuya utilización se ha extendido en las microcomputadoras y ofrecen gran facilidad para el manejo de sonidos, colores y animación sobre los cursos que se programan. Estos efectos resultan en una presentación novedosa y fácil de programar. El uso de microcomputadoras por un número creciente de usuarios sin preparación formal en computación ha promovido la creación de sistemas de EAC cada vez más sofisticados para asistir al usuario en el aprendizaje del manejo

CAI (Computer Aided Instruction) son las siglas para denominar a los sistemas de EAC.

SEAC cap.II. Conceptos acerca de los SEAC

de su máquina. Ejemplos son los de Hewlett Packard, IBM-PC, Columbia, etc. Estos sistemas de apoyo trascienden la repetición del manual en línea debido a la interacción que tienen con el usuario.

II.2) Definición y clasificación de los SEAC.

Los Sistemas de Enseñanza Auxiliados por Computadora, se definen como un proceso que involucra directamente a la computadora en el logro de los objetivos del proceso de enseñanza aprendizaje.

Se dividen en dos categorías:

a) los llamados 'Computer Aided Instruction (CAI)' o también llamados 'Computer Assisted Learning (CAL)' y

b) los 'Computer Managed Instruction (CMI)'.

Los sistemas CMI tienen funciones que los hacen más amplios que CAI y que permiten administrar la instrucción. Tienen entre otras las siguientes características principales:

- Contempla los planes de estudio.
- Contempla los programas de estudio.
- Registra información acerca del alumno.
- Monitorea el avance del alumno.
- Diagnostica y sugiere actividades para el alumno.
- Evalúa el aprendizaje.
- Proporciona información al profesor para la

planeación de su curso.

-Proporciona información al alumno sobre su desempeño.

La forma de interactuar de estos sistemas con el alumno usualmente se clasifica en:

- .Instrucción por práctica
- .Tutorial
- .Simulación
- .Juego
- .Resolución de problemas

A continuación se explica cual es la forma de operar de cada uno de estos modos:

Instrucción por práctica.- Es una forma complementaria al trabajo que desarrolla el profesor en el salón de clase y consiste en la resolución de ejercicios de práctica sobre conceptos vistos en clase. Además, incluye la realimentación que se proporciona al alumno para que conozca la habilidad que ha logrado con el material. Los programas son capaces de aumentar el grado de dificultad de los ejercicios dependiendo del desempeño del alumno o a petición de él.

Tutorial.- La instrucción no es precisamente complementaria a la del salón de clase, ya que los programas intentan enseñar nuevos

SEAC cap.II. Conceptos acerca de los SEAC

conceptos y junto con ellos se presentan preguntas entremezcladas. El formato de las preguntas es generalmente de relación de columnas, completar frases o de respuestas cortas abiertas. Evaluar al alumno, es también una tarea de los programas.

Simulación.- En la simulación el instructor usa a la computadora como una herramienta para poner al alumno en situaciones que no serían fácilmente conseguidas. Aquí, el alumno se entrena observando el comportamiento de una situación que cambia por los datos que él proporciona.

Juego.- El juego es también una forma de simulación que estimula o hace atractiva la participación del alumno.

Resolución de Problemas.- En este modo el alumno ya posee ciertas habilidades para el manejo del material y con la resolución de problemas se pretende que alcance un mejor dominio de ellas.

II.3) Evaluación de los SEAC.

Evaluar los SEAC con el objeto de escoger el más adecuado no es una tarea fácil. El proceso de enseñanza aprendizaje conlleva el establecimiento sistemático de objetivos y el análisis de las tareas de aprendizaje. Estas tareas, deben de estar presentes en la elaboración de los SEAC. Actualmente no existen estándares que permitan desarrollar SEAC. El desarrollo

SEAC cap.II. Conceptos acerca de los SEAC

de estos estándares ha sido difícil debido a la diversidad de opiniones sobre el tema y a que el estado actual de la computación vislumbra nuevos e importantes avances. Un intento de estandarización es el producto de un año de trabajo en la Universidad de Columbia analizando programas de EAC para microcomputadoras. De acuerdo a esto, los requerimientos mínimos que deben considerarse son los siguientes:

1) El rol curricular usado.- La función de la computadora en el proceso de enseñanza aprendizaje debe ser perfectamente delimitado. Actualmente se han identificado tres modos de emplearla :

a) Como suplemento al curriculum regular. Para ello las herramientas utilizadas son la solución de problemas, simulación e instrucción por práctica. A este modo también se le conoce como Adjunto.

b) Para la enseñanza completa de una unidad curricular. Se utiliza generalmente en forma tutorial enseñando conceptos y procedimientos con práctica intercalada, este modo es llamado Mainline.

c) Como un sistema administrador de la enseñanza. Con el cual es posible evaluar el grado en que cada estudiante ha alcanzado los objetivos y el nivel en el que se encuentra.

2) El modo de interacción usado.- Tutorial, Instrucción por práctica, simulación, etc.

3) La secuencia en la presentación del material.- La secuencia del material usualmente se organiza de manera lineal y no es posible alterarla. Son mejores los sistemas que permiten que el estudiante escoja su propia secuencia y que siga una ruta diferente que se ajuste a sus necesidades. El sistema o el instructor pueden sugerir una secuencia y el alumno estar en la posibilidad de aceptarla o de modificarla e inclusive pueden existir diferentes secuencias dependiendo del nivel del alumno.

4) El formato para la presentación del material.- La presentación de los textos en un formato adecuado facilita su lectura. Las posibilidades de manejar los textos dependen de las características de los equipos terminales. La mayoría no tiene la calidad de los materiales impresos, por esto, hay que tener cuidado al diseñar los formatos. En muchas ocasiones una gráfica puede transmitir mejor una idea que un texto. Combinar el uso de gráficas y textos de manera adecuada, crea un ambiente ameno para la instrucción.

5) El control del alumno sobre el sistema.- Dejar que el alumno tenga control sobre ciertos aspectos del material, es una característica que puede dar buenos resultados. Entre otras cosas el sistema debe dejar que el alumno: termine cuando quiera, revise las instrucciones desde cualquier punto del material, seleccione el número de ejercicios que quiere resolver y la dificultad de éstos, seleccione el número de ejemplos que desea revisar, que recurra a orientaciones en el material, etc.

SEAC cap.II. Conceptos acerca de los SEAC

6) El empleo de las ventajas de un CMI.- Los programas pueden aprovechar las capacidades del equipo para implementar las funciones que hemos definido para un CMI.

7) El uso de la realimentación.- La realimentación ha demostrado ser útil en el proceso de enseñanza aprendizaje, su uso, varía de acuerdo a las características o estilo de la instrucción, buscando con ella la mayor efectividad para la adquisición del conocimiento.

8) Uso de la generación aleatoria de ejercicios.- El orden de los ejercicios para el mismo material debe cambiarse de una sesión a otra, ya que con esto se tiene la posibilidad de aplicar el conocimiento o lo aprendido a situaciones distintas.

9) El manual del instructor y del alumno.- Los manuales son las herramientas para que los usuarios conozcan la operación del sistema. El instructor debe conocer los elementos que le ofrece el sistema para la presentación del material y para la administración de la instrucción, así como de su uso. Para el alumno, conocer la forma de registrarse, las opciones o menús, las convenciones, etc. Los manuales pueden estar dentro de la computadora como parte integral de los SEAC.

El desarrollo de SEAC siguiendo ciertos lineamientos pueden resultar en sistemas aceptables.

SEAC cap.II. Conceptos acerca de los SEAC

II.4) Calidad de los SEAC.

La calidad de estos sistemas depende del hardware, del software y del courseware (material preparado por el instructor). Algunos datos que se tienen revelan lo siguiente: en 1977 Human Resources Research Organization, publicó un lista con cerca de 350 escuelas que han utilizado exitosamente los sistemas de EAC y una lista de razones por las que se considera exitoso su uso; la Universidad Estatal de Florida informa que el curso de Física en EAC requirió del 17% de tiempo menos que el curso tradicional y el promedio de las calificaciones en los exámenes finales fué alto; en la Escuela Médica del Sur de California el programa de Anestesiología reveló que con poco entrenamiento en periodos cortos de tiempo, los alumnos alcanzaron un nivel aceptable en su comportamiento profesional; en Stanford el curso de Ruso ayudó significativamente a los alumnos en el examen final. Hay que reconocer que el uso de éstos sistemas en muchos casos es complementario y por el momento son pobres comparados con un aula bien trabajada.

II.5) Viabilidad de los SEAC.

La relación costo-beneficio de los SEAC tiende por un lado a disminuir (hardware), mientras que por el otro tiende a aumentar (software). Una estación para PLATO de 1976 a 1978, había disminuido su costo en cerca del 50%. El método aceptado para

SEAC cap.II. Conceptos acerca de los SEAC

calcular el costo, es sumar el costo del hardware, el software, el courseware, las comunicaciones, la implementación y el mantenimiento; todo esto se divide entre el número de horas-estudiante usado, lo que nos da una medida del costo unitario por alumno la cual puede compararse con la del sistema tradicional. En los EU por ejemplo, la instrucción tradicional se incrementó en un 13% en 1978-79-80, mientras que los SEAC disminuyeron en 5%. Esta tendencia ha seguido hasta 1985 y en proporciones seguramente mayores ya que el costo del hardware de 1980 a 1985 ha disminuido en cerca de un 40%. Considerar el costo como el único parametro de viabilidad no es la mejor práctica ya que deben considerarse factores sociales.

II.6) Hardware utilizado en los SEAC.

Usualmente se han utilizado terminales de video para la salida y el teclado como el principal dispositivo de entrada. Las terminales que se utilizan tienen posibilidades de graficar, de emitir algunos sonidos, son en color, se puede hacer animación y la calidad de los caracteres que se imprimen es muy buena. Los teclados a veces son combinados con el uso de pantallas sensibles a los dedos o a la luz, teclados gráficos, cursores gráficos, etc.

El procesador, la memoria primaria y secundaria que se utilizan, varían en una gran cantidad de situaciones; actualmente se están utilizando los microprocesadores y unidades de diskette.

SEAC cap.II. Conceptos acerca de los SEAC

Como características deseables adicionales encontramos combinar estos equipos con el uso de proyectores, de videograbadoras, de videodisco, etc. El uso de voz también es muy deseable, pero se estima que en aproximadamente 5 años estará disponible. Por otro lado, los equipos deben ser utilizados de manera general y no solo con el propósito de auxiliares en la instrucción.

El tamaño de los equipos utilizados puede dividirse en 5 categorías:

- Grandes sistemas trabajando en tiempo compartido (maneja cientos de estudiantes)
- Sistemas medianos trabajando en tiempo compartido (manejan de 50 a 100 estudiantes)
- Pequeños sistemas trabajando en tiempo compartido (manejan de 2 a 50 estudiantes)
- Sistemas de un sólo programa con ocasional acceso a un sistema remoto (manejan un sólo estudiante).
- Sistemas de un sólo programa (manejan un sólo estudiante).

II.7) Futuro de los SEAC.

Una de las interrogantes es: Qué sucederá con los SEAC en el

SEAC cap.II. Conceptos acerca de los SEAC

futuro?. Existen muchas opiniones al respecto y el consenso general manifiesta que la evolución tecnológica, el desarrollo de la psicología, la investigación sobre inteligencia artificial, la construcción de expertos, los avances en telecomunicaciones, etc, dejarán hacia 1990 un desarrollo importante en el área de los SEAC.

SEAC cap.III. Lineamientos que se consideraron en el desarrollo

III) Lineamientos que se consideraron en el desarrollo.

III.1) Proceso de enseñanza aprendizaje. Modelo de Anderson y Faust.

El proceso de enseñanza aprendizaje es generalmente definido como un conjunto de fases sucesivas en que se cumple el fenómeno intencional de la educación y de la instrucción. El concepto moderno conforme a las actuales tendencias pedagógicas, determina que enseñanza y aprendizaje no son factores independientes, sino que existe una relación, ya que el tipo de enseñanza dependerá de cuál sea el aprendizaje del que recibe dicha enseñanza y al mismo tiempo el aprendizaje es producto de la enseñanza.

Hasta el presente existen muchos intentos de diversos autores interesados en desarrollar modelos para el proceso de enseñanza aprendizaje desde una perspectiva sistemática y basados en las aportaciones que se han hecho a la clarificación de las variables psicológicas involucradas en el proceso.

Todos estos modelos se agrupan bajo el nombre genérico de Modelos de Sistematización de la Enseñanza. Los modelos desarrollados manejan ciertos elementos comunes y esenciales. las características generales que comparten los modelos son:

a) Manifiestan entre sus primeros pasos la clarificación de sus propósitos y objetivos de enseñanza. Es aquí donde se menciona qué es lo que se pretende lograr como resultado de la

SEAC cap.III. Lineamientos que se consideraron en el desarrollo
instrucción.

b) El establecimiento de los mecanismos necesarios para llegar a
conocer cuál es el nivel de los estudiantes, qué es lo que ya
conocen, si manejan el material, etc.

c) La implantación misma de la instrucción, contemplando toda la
actividad relacionada: elección del método de enseñanza, etc.

d) La determinación de si los alumnos alcanzaron los objetivos,
si la enseñanza fué efectiva, etc.

Existen varios modelos de sistematización de la enseñanza como el
de Popham y Baker, el de Bela Banathy, el de Anderson y Faust,
etc. En el presente trabajo, nos basaremos en el de Anderson y
Faust (1970) por ser un modelo que contempla las fases
principales del proceso de enseñanza aprendizaje y las aborda de
forma sencilla.

El modelo de Anderson y Faust (1970), inicia con la
especificación de objetivos, procede a la elaboración de
instrumentos de medición con base en los objetivos para una
evaluación diagnóstica y sumaria. Una vez elaborados se procede a
la aplicación de la premedición o evaluación diagnóstica. En
función de los datos obtenidos, se verifica si los conocimientos
previos son suficientes o si ya domina los objetivos. En el

SEAC cap.III. Lineamientos que se consideraron en el desarrollo
instrucción.

b) El establecimiento de los mecanismos necesarios para llegar a
conocer cuál es el nivel de los estudiantes, qué es lo que ya
conocen, si manejan el material, etc.

c) La implantación misma de la instrucción, contemplando toda la
actividad relacionada: elección del método de enseñanza, etc.

d) La determinación de si los alumnos alcanzaron los objetivos,
si la enseñanza fué efectiva, etc.

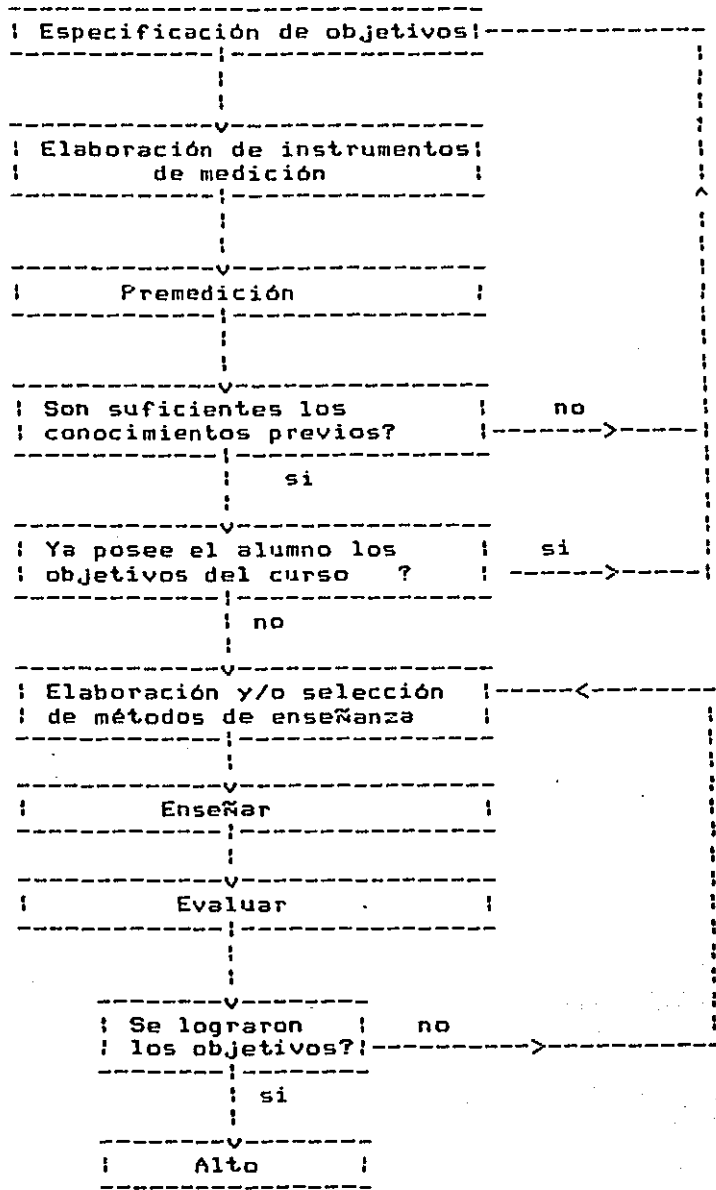
Existen varios modelos de sistematización de la enseñanza como el
de Popham y Baker, el de Bela Banathy, el de Anderson y Faust,
etc. En el presente trabajo, nos basaremos en el de Anderson y
Faust (1970) por ser un modelo que contempla las fases
principales del proceso de enseñanza aprendizaje y las aborda de
forma sencilla.

El modelo de Anderson y Faust (1970), inicia con la
especificación de objetivos, procede a la elaboración de
instrumentos de medición con base en los objetivos para una
evaluación diagnóstica y sumaria. Una vez elaborados se procede a
la aplicación de la premedición o evaluación diagnóstica. En
función de los datos obtenidos, se verifica si los conocimientos
previos son suficientes o si ya domina los objetivos. En el

SEAC cap.III. Lineamientos que se consideraron en el desarrollo primer caso, si el alumno no tiene los conocimientos previos se regresa a los objetivos y se hace un replanteamiento de ellos. En el segundo caso, si el alumno ya posee los objetivos, nuevamente se regresa a ellos para su modificación. Si los alumnos no poseen los objetivos, entonces se procede a la selección o desarrollo de métodos de enseñanza, con base en los objetivos. Paso seguido, se procede al hecho mismo de la enseñanza basándose en todo lo mencionado antes (objetivos, premedición, métodos, etc.).

Una vez que se dió la enseñanza procedemos a la evaluación del logro de los objetivos, si acaso éstos no se cumplieron hay que investigar las causas y si se cumplieron damos por terminada esta pequeña fase del proceso educativo.

SEAC cap.III. Lineamientos que se consideraron en el desarrollo



Modelo de Anderson y Faust.

SEAC cap.III. Lineamientos que se consideraron en el desarrollo

Revisaremos ahora cada uno de los componentes esenciales del modelo seleccionado.

III.2) Especificación de objetivos.

Al hablar de los objetivos de la educación se encuentra que la connotación generalmente aceptada, es la que los define como: "las intenciones o propósitos que describen en términos de cambios conductuales que se espera del estudiante al finalizar un ciclo de instrucción".

III.3) Evaluación en la sistematización.

Evaluar constituye uno de los elementos de toda actividad docente, es el proceso a través del cual, el instructor conoce sus éxitos o sus fracasos. Por lo general, ésto lo determina conociendo el grado en que sus estudiantes dominan aquello que les enseñó durante la instrucción. La evaluación se lleva a cabo en diferentes momentos del proceso de enseñanza aprendizaje con una multiplicidad de funciones.

De acuerdo con Tyler (1970), Paulson (1973) y Brink (1977) entre otros autores, el evaluar es un proceso a través del cual se obtiene información y ésta se aprecia en referencia a valores específicos o a la luz de normas, que permiten la emisión de juicios de valor tendientes a tomar decisiones referentes a ratificar o rectificar la situación evaluada.

SEAC cap.III. Lineamientos que se consideraron en el desarrollo

Dentro del marco de la sistematización de la enseñanza la evaluación se aplica desde antes de iniciar el proceso de instrucción, llamándose a este tipo evaluación diagnóstica, a lo largo del proceso de instrucción denominándose formativa y al finalizar la instrucción, evaluación sumaria.

a) Evaluación diagnóstica.- Antes de iniciar la instrucción el instructor debe saber si el alumno tiene los requisitos del curso o si ya domina los objetivos del curso. Para ello se aplica una evaluación llamada diagnóstica.

b) Evaluación formativa.- Durante el proceso de instrucción, es común que se lleven a cabo evaluaciones periódicas ya sea al finalizar un tema, unidad o grupo de unidades. A este tipo de evaluaciones se les llama formativas y es a través de ellas que el profesor y los alumnos pueden saber si están alcanzando los objetivos y no tendrán que esperar a finalizar el curso para saberlo y tomar las medidas correctivas si así se demandase.

c) Evaluación sumaria.- Al finalizar el periodo de instrucción, la evaluación sumaria indicará cómo funcionó todo el proceso, indicando si los estudiantes alcanzaron los objetivos generales propuestos.

Existen muchos recursos de los que puede disponer el profesor para realizar la evaluación, uno de los más empleados es la administración de pruebas, entre otras razones, porque se aplican en el momento justo y adecuado, se puede obtener un

SEAC cap.III. Lineamientos que se consideraron en el desarrollo

producto permanente de la ejecución de los alumnos, se puede planear sus alcances y estructura, se pueden aplicar al mismo tiempo a muchos estudiantes, son económicas, etc.

Un reactivo de evaluación es una pregunta o problema que solicita del estudiante una respuesta o solución. Las pruebas se elaboran en base a reactivos. En base a la respuesta que solicitan los reactivos pueden dividirse en: reactivos de respuesta semiestructurada, aquéllos en los que el alumno tiene que escribir una respuesta breve como los de completamiento y en reactivos de respuesta estructurada cuando el alumno tiene que elegir entre dos o más respuestas alternativas como los de opción múltiple, de apareamiento, de jerarquización y de respuesta alterna.

III.4) Planeación de la instrucción.

Cuando planeamos la instrucción, intentamos mejorar el aprendizaje y una de las condiciones para mejorar el aprendizaje de los educandos es que el profesor conozca cuál es el fenómeno y qué prácticas o estrategias se pueden seguir para producirlo. Derivadas de las teorías psicológicas del aprendizaje, las variables del aprendizaje permiten identificar las condiciones tanto del ambiente como del interior del sujeto que producen y/o determinan la calidad y cantidad del aprendizaje. Son de especial interés aquéllas que el docente puede manejar, ya que ello le

SEAC cap.III. Lineamientos que se consideraron en el desarrollo permite tener ingerencia en la calidad del aprendizaje. Cuando las variables se encuentran en el interior del individuo se denominan intrapersonales. Estas se conciben como estados mentales inherentes al desarrollo de los individuos, factores biológicos, de pensamiento, de personalidad, fatiga y actitudes. Cuando las variables se encuentran fuera del individuo tiene la característica que pueden ser controladas y hacen referencia a aspectos como: práctica, materiales de enseñanza, progresión de la dificultad, conocimiento de los objetivos, participación activa, retroalimentación y discriminación, factores de grupo y sociales, motivación extrínseca, características del profesor y estrategias instruccionales.

Estas ideas han sido consideradas para la elaboración del sistema y del material de prueba. En el análisis y en la especificación del sistema se indican cuáles y con qué amplitud han sido incorporadas.

SEAC cap.IV. Análisis del sistema

IV) Análisis del sistema.

IV.1) Descripción general del sistema.

Como punto de partida para el análisis del sistema se tomaron del modelo de Anderson y Faust las etapas siguientes:

- Especificación de objetivos
- Elaboración de instrumentos de medición
- Planeación de la instrucción
- Aplicación de la medición
- Aplicación de la instrucción
- Evaluación

También, se consideraron los procesos administrativos siguientes:

- Registro de Alumnos y seguimiento
- Consulta de instructores y alumnos

A continuación se describe de manera general lo que para cada etapa el sistema debe hacer.

Especificación de objetivos.

Permitir la captura, modificación y despliegue del conjunto de objetivos que se persiguen con el material.

Elaboración de instrumentos de medición.

Permitir la captura y modificación de diferentes tipos de preguntas, como pueden ser las que a continuación se mencionan:

SEAC cap.IV. Análisis del sistema

- .Opción múltiple
- .Falso o verdadero
- .Completamiento
- .Relación de columnas
- .Jerarquización

Planeación de la instrucción.

Permitir la captura y modificación del texto que constituye el contenido del curso. Se ha sugerido el siguiente formato, el cual a juicio del instructor puede ser o no respetado:

- .Elaboración del índice del material
- .Especificación de objetivos
- .Aplicación de una prueba diagnóstica
- .Elaboración del texto, figuras, ejercicios y pruebas formativas
- .Elaboración de resumen
- .Especificación de bibliografía
- .Aplicación de una prueba sumaria

De acuerdo a este formato, se requiere de una mecánica que permita la especificación y aplicación de ejemplos, la integración de las preguntas en una prueba, así como la calificación de ellas.

Las pruebas, son la integración de un conjunto de reactivos y pueden contener cualquier combinación de los tipos de reactivos ya mencionados. Cada prueba debe contener los criterios de

SEAC cap.IV. Análisis del sistema

evaluación.

Aplicación de la medición.

Permitir la aplicación de una prueba en cualquier etapa del proceso y repetirla el número de veces que el alumno lo solicite.

Aplicación de la instrucción.

Permitirle al alumno avanzar y retroceder sobre el texto, elegir un ejemplo, elegir una prueba, abandonar la sesión, etc.

Evaluación.

Determinar si el alumno debe o no seguir con el estudio del material, comunicarle en que preguntas falló y recomendarle que debe hacer para que la instrucción avance.

Registro de alumnos y seguimiento.

Capturar los datos personales del alumno y registrar el desempeño de él en cada sesión de estudio.

Consulta de instructores y alumnos.

Consultar el archivo de datos estadísticos.

Elaboración de guías.

Permitir la elaboración de las guías del instructor y del alumno.

IV.2) Configuración del Equipo.

Por sus características las microcomputadoras se presentan como

SEAC cap.IV. Análisis del sistema

una alternativa para los SEAC, su bajo costo y sus capacidades para el manejo de gráficas, sonido, respuesta inmediata, etc permiten incorporar un hardware, el cual reduce el costo total del sistema.

La configuración bajo la cual fue planeado el desarrollo del trabajo consiste de:

- a) CPU , un microprocesador
- b) Memoria primaria, RAM con un mínimo de 256kb
- c) Memoria secundaria, Diskettes 2 unidades
- d) Entrada, teclado alfanumérico
- e) Salida, CRT

La instrumentación del sistema se realizó en una HP-150 con las siguientes características generales:

Unidad central.- El sistema HP-150 esta basado en un microprocesador de intel 8088, con una arquitectura interna de 16 bits.

Memoria primaria.- Consiste de una memoria RAM de 256 Kbytes, una memoria estática de 6 Kbytes para el manejo de la pantalla y una memoria ROM de 160 Kbytes.

Teclado.- El teclado esta configurado en varios subconjuntos de teclas. Uno de ellos corresponde al juego de caracteres de máquina de escribir, tipo Qwerty, con adaptación al sistema

SEAC cap.IV. Análisis del sistema

castellano.

En total, consta de 107 teclas, entre las que se incluyen las correspondientes al control del cursor, a las funciones de edición, al keypad numérico (18 teclas), y un grupo de 8 teclas de función programables, ubicadas en la parte superior del tablero.

Pantalla.- El monitor es monocromático de fósforo verde (P31), y su dimensión es de 7X5 pulgadas, con un formato de 27 líneas de 80 columnas. La línea inferior esta reservada para mensajes relativos al estado del sistema. Las dos líneas inmediatas superiores contienen los comandos correspondientes a las teclas de función programables. De este modo quedan 24 líneas, que es lo habitual para la aplicación de textos y gráficas. La resolución, trabajando en modo alfanumérico, es de 720 por 378 puntos, mientras que en modo gráfico es de 512 por 390.

Memorias secundaria.- Consta de dos unidades Sony para discos de 3.5 pulgadas de simple cara y con 270 Kbytes de capacidad en cada uno de ellos.

Programas básicos.- Cuenta con el sistema operativo MS-DOS, lenguajes de alto nivel como Basic, Cobol, Pascal, etc. y la utilería del sistema.

Los programas se escribieron en el lenguaje Pascal debido a que en las instalaciones solo tenemos los lenguajes Basic, Fortran 77 y Pascal. Por otro lado al elegir Pascal se consideraron las siguientes características: presenta un conjunto de estructuras de control semejante al propuesto por la Programación

Estructurada y por esto, la programación se facilita; es modular, el programa se compone de partes con funciones muy bien definidas; se ejerce control sobre el tipo de los datos; es transportable sobre todo si se usan las estructuras estandares; es compilable; el manejo de cadenas de caracteres es sencillo; etc.

SEAC cap.V. Diseo del sistema.

V) Diseo del sistema.

V.1) Notación y convenciones utilizadas.

Antes de iniciar la especificación se da una explicación de la notación y convenciones utilizadas. El análisis, diseño y programación del sistema han sido desarrollados siguiendo las ideas de Ed Yourdon, Tom de Marco, Gane, Sarson conocidas como Análisis y Diseño Estructurado. La especificación se basa en la construcción de diagramas de flujos de datos, cuyos componentes fundamentales son: el flujo de datos (F) representado por vectores o flechas, los procesos (P) por rectangulos , los almacenes de datos (A) por rectángulos con paréntesis en los costados y las entidades externas (E) representados por rectangulos con una doble raya en la parte superior. La especificación se complementa con la documentación de cada uno de los diagramas la cual consiste de :

Para los flujos de datos(F)

- .nombre del flujo
- .una descripción del flujo
- .de que componente a que componente va
- .las estructuras de datos que fluyen

Para las estructuras de datos (Ed)

- .nombre de la estructura
- .una narrativa del contenido de la estructura
- .una especificación del contenido de acuerdo a la

siguiente notación.

nombre-de-la-estructura=atributo1+
atributo2+
.....

atributo : atributo subrayado es llave

{atributo} : es un campo de repetición

[atributo] : es opcional

.descripción de los flujos o almacenes donde aparece

Para los almacenes de datos (A)

- .una narrativa del contenido
- .flujos que llegan
- .flujos que salen
- .estructuras de datos que contiene

Para los datos elementales

- .nombre del dato
- .tipo (numérico, alfanumérico,etc)
- .valor (valores típicos o rangos)
- .longitud en bytes
- .estructuras donde aparece

Para los procesos (P)

- .nombre del proceso

.resumen lógico del proceso

El diseño de los programas se ha hecho construyendo la carta de estructura, que consiste en una especificación arborecente desde la idea más general hasta los procesos más elementales. Cada proceso elemental se describe utilizando pseudocódigo. Para especificar los formatos de las pantallas tanto de captura de datos como de instrucción, se utiliza una gráfica que simula una pantalla de CRT.

V.2) Especificación funcional y diseño detallado del sistema.

El diagrama general del sistema (fig V.1) muestra que existen únicamente tres entidades externas al sistema quienes en este caso son los usuarios del sistema.

(E01) SERVICIOS EDUCATIVOS

(E02) INSTRUCTOR/MAESTRO

(E03) ALUMNO

De acuerdo al análisis y con el propósito de hacer un sistema flexible las necesidades se agruparon de la forma siguiente:

I. ELABORACION DE LA GUIA DEL INSTRUCTOR (SEACO)

Para lo cual es necesario:

(P01) Elaborar la guía del instructor.

II. ELABORACION DEL MATERIAL (SEAC1)

Para lo cual es necesario:

(P02) Consultar la guía para el instructor.

(P03) Elaborar los contenidos.

(P04) Elaborar las preguntas o reactivos.

(P05) Elaborar la guía para el alumno.

(P06) Consultar el seguimiento.

III. APLICACION DE LA INSTRUCCION (SEAC2)

Para lo cual es necesario:

(P07) Consultar la guía del alumno.

(P08) Aplicar el material.

(P09) Consultar el seguimiento.

Los almacenes de datos que fueron identificados son :

(A01) La guía del instructor

(A02) Los contenidos de las unidades

(A03) Los reactivos de las unidades

(A04) La guía del alumno

(A05) Los datos personales del alumno

Las relaciones entre las entidades externas, los procesos y los almacenes de datos pueden ser visualizadas con los flujos que se especifican en el diagrama(fig V.1).

SEAC cap.V. Diseo del sistema.

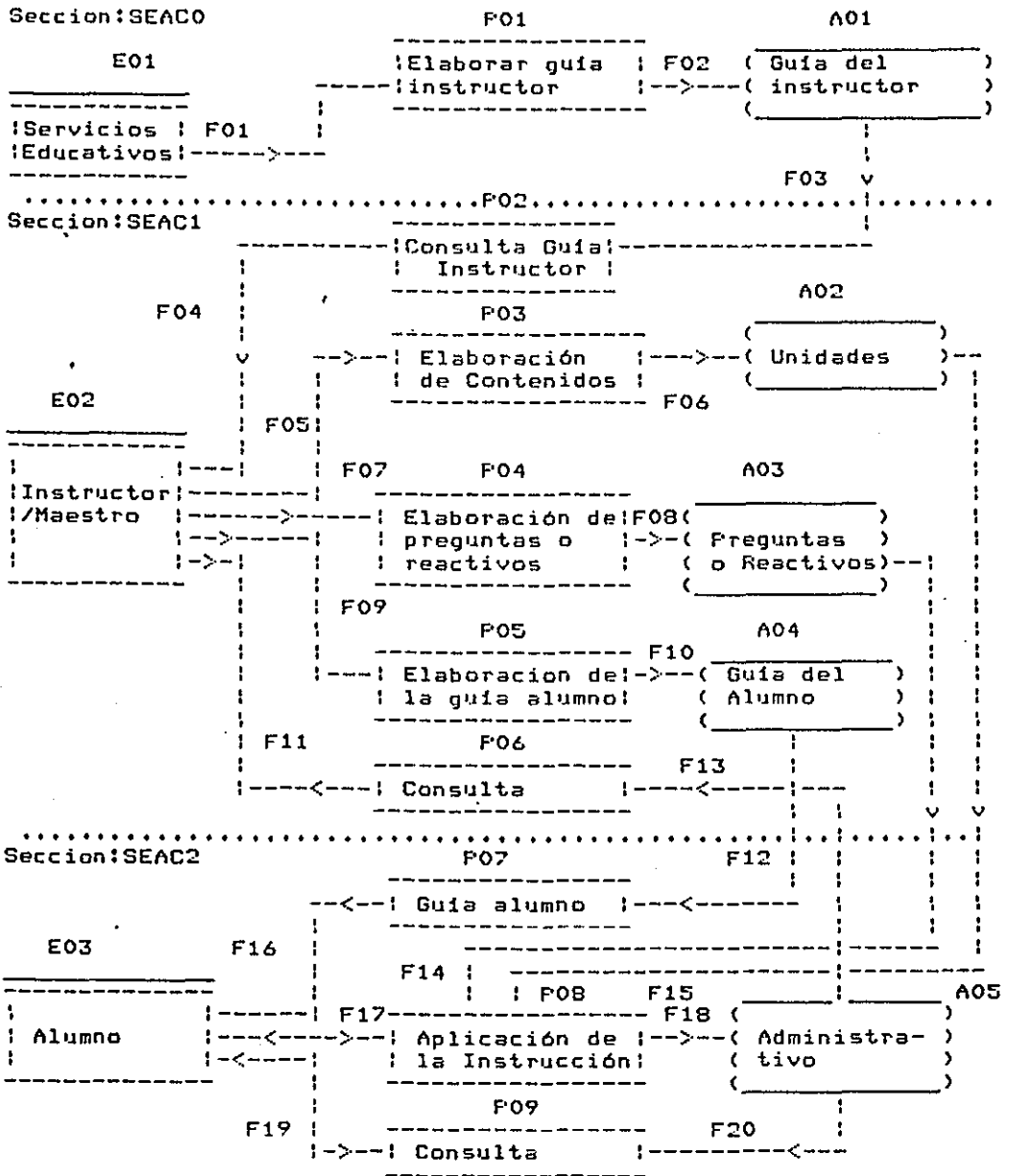


Fig U.1. Diagrama General del Sistema.

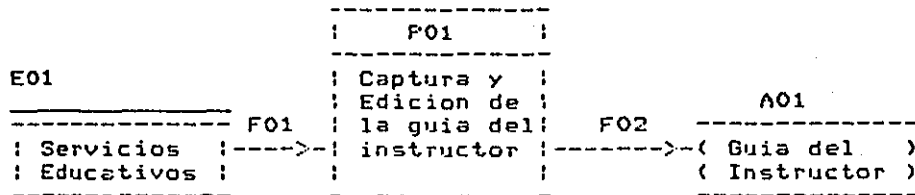
SEAC cap.V. Diseo del sistema.

DESCRIPCION DE LA SECCION SEACO.

V.2.1) Elaboración de la guía para el instructor.

Para hacer adecuadamente el material y hacer buen uso del sistema, el instructor cuenta con una guía que se elabora en Servicios Educativos. La guía hace énfasis en los pasos del modelo y en la forma como deben utilizarse las herramientas del sistema propuesto.

Diagrama del proceso



Especificación de flujos de datos

Nombre:F01.
Descrip:Datos acerca de la guía del instructor.
Est. de Dat.:Ed1.

Nombre:F02.
Descrip:Datos depurados de la guía del instructor.
Est. de Dat.:Ed1.

Especificación de la estructura de datos

Ed1. Es un texto en donde cada línea se identifica con un 13 (CR) seguido de un 10 (LF).

!..línea1...1310.....línea2....1310.....líneaX...1310eof!

Almacen de datos

A01. Es un archivo secuencial con la estructura Ed1.

SEAC cap.V. Diseo del sistema.

Nombre. El nombre es Guia del instructor(GUIAINS)

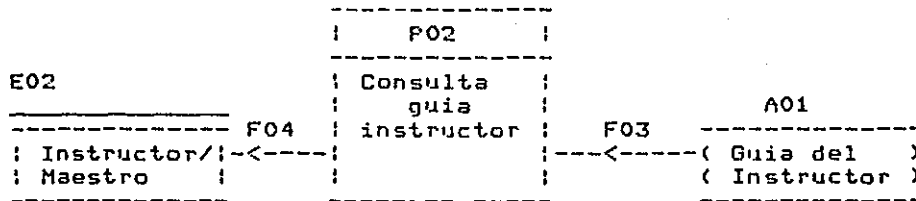
Proceso

P01. Captura y edici3n de la guia del instructor.
El programa que se utiliza es cualquier editor de texto que separe cada lnea con CRLF.

DESCRIPCION DE LA SECCION SEAC1.

V.2.2) Consulta de la guia para el instructor.

Diagrama del proceso



Especificaci3n de flujos de datos

Nombre:F03.
Descip:Guia del instructor.
Est. de Dat.:Ed1.

Nombre:F04.
Descrip:Guia del instructor en paginas.
Est. de Dat.:Ed1.

Proceso

P02. El despliegue de la guia se hace con el programa "instructor" que se describe m3s adelante.

V.2.3) Elaboraci3n de los contenidos.

Para la elaboraci3n de los contenidos se utiliza el editor

de texto del equipo. El formato es totalmente libre para el instructor y se aprovechan todas las facilidades de este editor. Para cumplir con los objetivos de este trabajo se dan las siguientes facilidades:

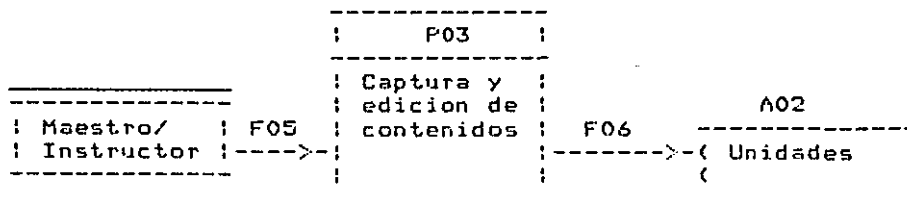
Ñ Este caracter colocado al principio de una línea indica cambio de hoja. Se utiliza para indicar que cierta parte del contenido se despliegue limpiando la pantalla.

NE# Esta secuencia indica que se ejecute el ejemplo marcado con #. # puede ser cualquier caracter del teclado. Estos ejemplos se programan y se incluyen en la biblioteca BIR1SEAC.PAS como PROCEDURE XX y se modifica el procedure EJEMPLO incorporando un 'if c=# then XX'.

NPaa#,aa#,...;# Esta secuencia indica que se ejecuten los reactivos cuyos nombres son aa#,aa#,... los cuales constituyen una prueba. Los nombres son dos letras cualquiera seguida de un número de un solo dígito. Con ';#' se especifica el nivel de ejecución que se desea alcanzar. Si el resultado de la prueba es mayor o igual a #. La acción que se toma es indicarle al alumno el nivel de ejecución logrado. Si es menor que ### se le invita a que revise nuevamente el material y si es mayor simplemente continúa.

Ejemplo:NPOM1,RC2;80

Diagrama del proceso



Especificación de flujos de datos .

Nombre:F05.
Descrip: Contenido de cada una de las unidades.
Est.de Dat.: Ed2.

Nombre:F06.
Descrip: Contenido depurado de las unidades.
Est.de Dat.: Ed2.

Especificación de la estructura de datos

Ed2=Ed1

Especificación del almacen de datos.

A02. Es un archivo secuencial con Ed2.
Nombre. Para cada unidad el nombre es UNIDAD#.

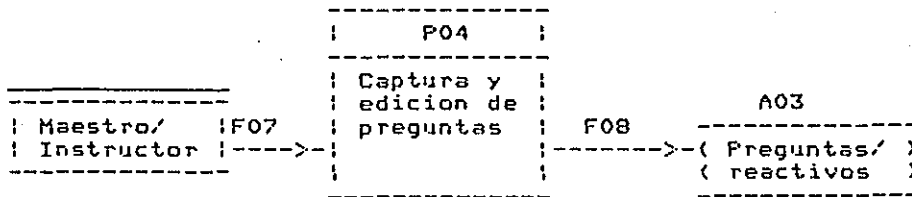
V.2.4) Elaboración de preguntas o reactivos.

Para elaborar las preguntas se utiliza también el editor de texto y para el logro de los objetivos del presente trabajo se dan las siguientes facilidades:

Ñaa# Esta secuencia indica el inicio de una pregunta cuyas siglas que lo identifican son 'aa' que pueden ser dos símbolos cualquiera del teclado y uno o más números a continuación.

????### Una secuencia de símbolos de interrogación indican que en esa posición dentro del reactivo se espera una respuesta del alumno. A esta secuencia debe seguirle la respuesta ###. El número de símbolos de interrogación es variable y no necesariamente de la longitud de la respuesta, basta con ser mayor.

Diagrama del proceso



Especificación de flujos de datos

Nombre:F07.
Descrip: Lleva las preguntas y sus respuestas.
Est.de Dat.: Ed3.

Nombre:F08.
Descrip: Preguntas y respuestas depuradas.
Est.de Dat.: Ed3.

Especificación de la estructura de datos

Ed3=Ed1

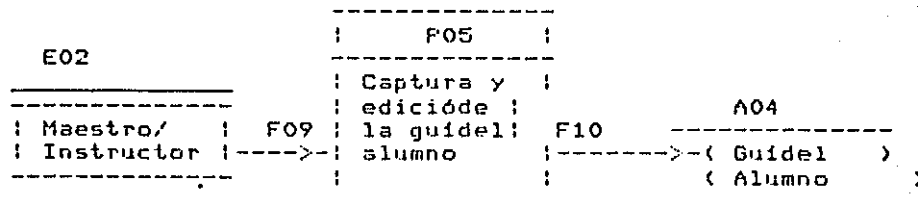
Especificación del almacen de datos.

A03. Es un archivo secuencial con Ed3.
Nombre. Para cada unidad el nombre es REACT#.

SEAC cap.V. Diseo del sistema.

V.2.5) Elaboración de la guía para el alumno.

Diagrama del proceso



Especificación de flujos de datos

Nombre:F09.
Descrip: Datos de la guía para el alumno.
Est.de Dat.: Ed4.

Nombre:F10.
Descrip: Datos depurados de la guía del alumno.
Est.de Dat.: Ed4.

Especificación de la estructura de datos

Ed4=Ed1

Especificación del almacén de datos.

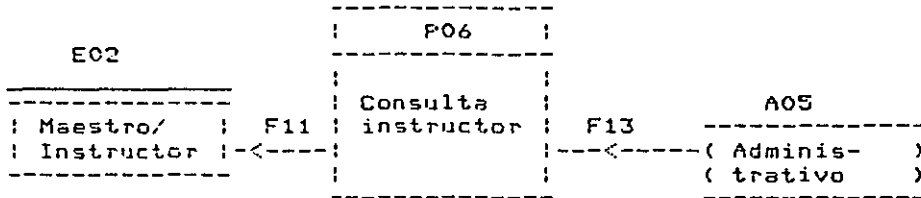
A04. Es un archivo secuencial con Ed4.
Nombre. Guía del alumno (GUIAALU).

V.2.6) Consulta de los datos del alumno por el instructor.

Con el identificador del alumno, el instructor puede recuperar los datos del alumno.

SEAC cap.V. Diseo del sistema.

Diagrama del proceso



Especificación de flujos de datos

Nombre:F11.
Descrip: Datos del desempeño del alumno en el formato de salida.
Est.de Dat.: Ed5.

Nombre:F13.
Descrip: Datos del desempeño del alumno.
Est.de Dat.: Ed5.

Especificación de la estructura de datos

DATPERAL Ed5

.contiene los datos del alumno
.estructura:

ALUMNO= NO-CUENTA+
FECHA+
NOMBRE-ALUMNO+
NO-DE-SEMESTRE+
NO-MATERIAS+
TRADICIONAL+
EXTRAORDINARIO+
NO-MATERIAS+
ANTECEDENTES
[SESION]

SESION= INIC-SESION+
NO-REACT+
CALIFICACION+
FIN-SESION

.flujos donde se encuentra: F13,F18,F20

SEAC cap.V. Diseo del sistema.

Especificación de datos elementales

nombre	tipo	valor	long
NO-CUENTA	N		8
FECHA	N		6
NOMBRE-ALUMNO	A		40
NO-DE-SEMESTRE	N	1a10	2
NO-MATERIAS	N	1a9	1
TRADICIONAL	B	1o0	1
EXTRAORDINARIO	B	1o0	1
ANTECEDENTES	B	1o0	1
INIC-SESION	A	-	1
NO-REACT	A		VAR
CALIFICACION	N	1a100	3
FIN-SESION	A	-	1

Especificación almacén de datos

ADMINISTRATIVO [A05]

- .contiene los datos de los alumnos
- .flujos que llegan: F18
- .flujos que salen: F13,F20
- .estructuras que tienes: [ATPERAL [Ed5]

Operación del proceso SEAC1

Para manejar las opciones del proceso SEAC1, se cuenta con un menú que permite elegir cualquiera de las siguientes operaciones:

- Desplegar la guía del instructor
- Elaborar contenidos, Elaborar instrumentos de medición, Elaborar la guía del alumno
- Consultar el archivo de seguimiento
- Abandonar el proceso

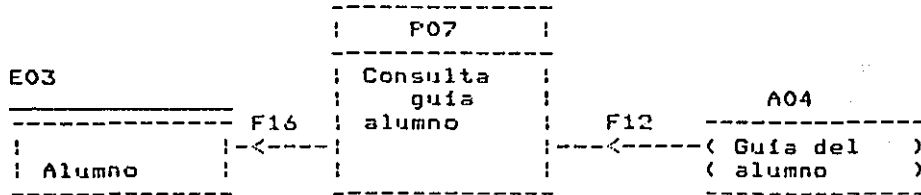
Cada opción esta asociada con las teclas programadas y la

SEAC cap.V. Diseo del sistema.

DESCRIPCION DEL PROCESO SEAC2

V.2.7) Consulta de la guía para del alumno

Diagrama del proceso



Especificación de flujos de datos

Nombre:F16.
Descrip:Guía del alumno en el formato de salida.
Est. de Dat.:Ed4.

Nombre:F12.
Descrip:Guía del alumno.
Est. de Dat.:Ed4.

Proceso

P07. El despliegue de la guía se hace con el programa "instructor" que se describe más adelante.

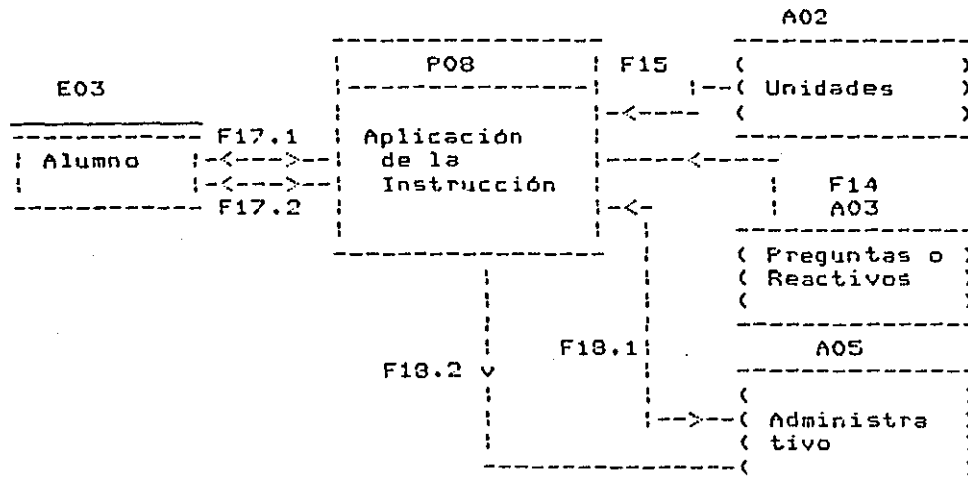
V.2.8) Aplicación de la instrucción.

La aplicación de la instrucción contempla la inscripción y la aplicación del material. La inscripción se realiza por excepción, esto es, si el número de cuenta no existe se da de alta al alumno. La aplicación se realiza permitiendo que el alumno avance hoja por hoja hacia adelante, hacia atrás, a la primera hoja, a

SEAC cap.V. Diseo del sistema.

la ultima hoja y a una hoja en especifico. El programa es capaz de realizar en su caso despliegue de ejemplos , aplicaci3n de reactivos y llevar un seguimiento del alumno.

Diagrama del proceso



Especificacion de flujos de datos.

- Nombre:F17.1
 .de APLICACION DE LA INSTRUCCION [P08] a ALUMNO [E03].
 .lleva la pantalla de inscripci3n
 .estructura: texto desplegado por programa.
- .de ALUMNO [E03] a APLICACION DE LA INSTRUCCION [P08].
 .lleva los datos del alumno.
 .estructura: DATPERAL [Ed5]
- .de APLICACION DE LA INSTRUCCION [P08] a ALUMNO [E03].
 .lleva la pantalla de elecci3n de la unidad
 .estructura: texto desplegado por programa.
- .de ALUMNO [E03] a APLICACION DE LA INSTRUCCION [P08].
 .lleva el n3mero de la unidad elegida
 .estructura: NUMUNI [Ed6]

SEAC cap.V. Diseo del sistema.

Nombre:F17.2

- .de APLICACION DE LA INSTRUCCION [P08] a ALUMNO [E03].
- .lleva la pantalla de las opciones del proceso.
- .estructura: texto desplegado por programa.

- .de ALUMNO [E03] a APLICACION DE LA INSTRUCCION [P08].
- .lleva la eleccion del alumno.
- .estructura: ELECCION [Ed7]

- .de APLICACION DE LA INSTRUCCION [P08] a ALUMNO [E03].
- .lleva el material al alumno.
- .estructuras que fluyen: UNIDAD# [Ed2]

- .de ALUMNO [E03] a APLICACION DE LA INSTRUCCION [P08].
- .lleva las respuestas de participacion del alumno.
- .estructuras que fluyen

Nombre:F15.

- .de UNIDADES [A02] a APLICACION DE LA INSTRUCCION [P08]
- .lleva el contenido
- .estructura de datos: UNIDAD# [Ed2]

Nombre:F14.

- .de PREGUNTAS/REACTIVOS [A03] a APLICACION DE LA INSTRUCCION [P08]
- .lleva los reactivos
- .estructura de datos: PREGUN# [Ed3]

Nombre:F18.1

- .de APLICACION DE LA INSTRUCCION [P08] a ADMINISTRATIVO [A05].
- .lleva los datos del alumno ya depurados y en el formato para almacenarlos.
- .estructura de datos: DATPERAL [Ed5]

Nombre:F18.2

- .de APLICACION DE LA INSTRUCCION [P08] a ADMINISTRATIVO [A05].
- .lleva los datos del alumno por sesión.
- .estructura de datos: SESION

Especificación de estructuras de datos

NUMUNI [Ed6]

SEAC cap.V. Diseño del sistema.

- .contiene el número de la unidad
- .estructura: Línea de caracteres
- .flujos donde se encuentra: [F17.1]

ELECCION [Ed7]

- .contiene la elección del alumno
- .estructura: Línea de caracteres enviada por tecla programada
- .flujos donde se encuentra: [F17.2]

Resumen lógico del proceso

Identifica el alumno
Selecciona la unidad
Carga el contenido
Despliega las opciones del proceso
Si es CAMBIO DE CUADRO ADELANTE entonces
 Avanza el material 25 líneas
Si es CAMBIO DE CUADRO ATRAS entonces
 Retrocede el material 25 líneas
Si es PRIMERA HOJA entonces
 imprime la hoja uno
Si es ULTIMA HOJA entonces
 imprime la última hoja
Si es CUADRO NUMERO entonces
 lectura del número del cuadro
 imprime la hoja
Si es SALIDA entonces
 termina el proceso
fin

Método de Acceso

Con el objeto de facilitar el almacenamiento y la recuperación de los datos en el almacén ADMINISTRATIVO los registros se accesan de la siguiente forma:

Una función de Hash. Basada en la función módulo y propuesta por Floyd [ref 8]. Con la cual mapearemos a un archivo de índices.

SEAC cap.V. Diseo del sistema.

Un archivo de Indices. Este archivo contiene la llave y la direcci3n de los datos en el archivo de datos. Cada registro del archivo contiene m3ltiples pares de llave-direcci3n y son aquellas llaves sin3nimas. Si el registro se llena de llaves, contiene un campo liga para extenderse.

Un archivo de datos. Es un archivo que contiene los datos propiamente dicho y con el objeto de manejar estructuras de longitud variable al final tiene un campo para el manejo de una liga.

La estructura de los archivos se muestra a continuaci3n:

Funci3n de HASH -----> archivo de cubetas

archivo de cubetas:

longitud del registro 127 bytes
reg 1= contador de registros
reg 2 a reg 11 = cubetas 1 a 10

estructura del registro:

llave(1),apt(1),llave(2),apt(2),.....,cont,liga

llave(i)- es la llave y su longitud es : variable
apt(i)- apunta a los datos y su longitud es de: 4bytes
cont- n3mero de llaves en la cubeta su longitud: 4bytes
liga- apunta a la continuaci3n de la cubeta,longitud: 4bytes

archivo de datos:

longitud del registro 255 bytes
registro 1= contador de registros
registro 2 a registro n = registros de datos

SEAC cap.V. Diseo del sistema.

estructura del registro:

datos,liga

datos- son los datos y su longitud es de 25bytes

liga - apunta a la continuaci3n de los datos su longitud es de 4 bytes

Diseño de pantallas

Pantalla para captura de datos inscripci3n alumno

```

-----
|           Sistema de enseñaanza auxiliada por computadora           |
|           [inscripci3n]                                             |
|-----|
| no de cuenta .....|
| nombre .....|
| fecha .....|
| semestre que cursas .....|
| ya tomaste el curso tradicional (SI,NO) ..|
| estas preparando el extraordinario (SI,NO) ..|
| has tomado otro curso en la computadora(SI,NO) ..|
| cuantas materias ,llevas este semestre .....|
| ya tienes los antecedentes para este curso(SI,NO) ..|
|-----|

```

Pantalla principal durante la instrucc3n

```

-----
| Sistema de enseñaanza auxiliada por computadora [nombre ] |
| [No.Ho.]      [instructor]                [calificacion] |
|-----|
|
|
|
|-----|
| Adelante| atras |Primero|Ultimo|Cuad No. |           |salida |
|         |      |      |      |      |           |      |
|-----|
| f1      | f2      | f3      | f4      | f5      |           | f8      |

```

Diseño del programa

El programa tiene la siguiente estructura modular:

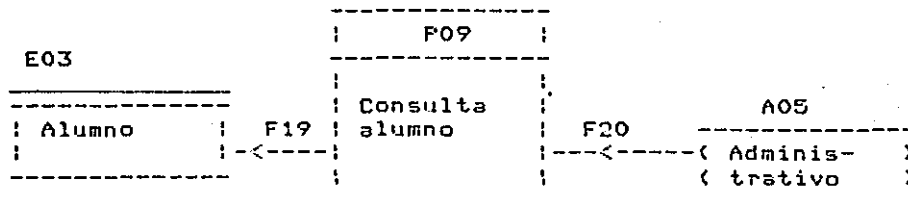
SEAC cap.V. Diseo del sistema.

- nivel 0 Instruc (programa principal)
- nivel 1 Inscripcion (Realiza la inscripci3n del alumno)
 - Entrada (Lectura del n3mero de la unidad)
 - adm1 (Registra inicio de sesi3n)
 - Eschoja (Escribe y despacha una hoja del texto)
 - adm3 (Registra fin de sesi3n)
- nivel 2 Maneja (Asigna las tareas dependiendo del comando)
- nivel 3 Ejemplo (Despacha los ejemplos)
 - Prueba (Despacha prueba)
 - adm2 (Registra desempe1o en prueba)
- nivel4 Pilaej (Ejemplo de pila)
 - colaej (Ejemplo de cola)
 - ccolaej (Ejemplo de cola ligada)
 - liligada (Ejemplo de lista ligada)
 - Buscap (Busca una pregunta)
 - splica (Aplica un reactivo)
 - Condici3n (Eval3a el resultado de la prueba)

IV.2.9) Consulta de los datos del alumno por el alumno.

Se despliega la informaci3n registrada durante el proceso de instrucci3n, referente al desempe1o del alumno.

Diagrama del proceso



Especificaci3n de flujos de datos

Nombre: F19.
Descrip: Datos del desempe1o del alumno en el formato de salida.
Est.de Dat.: Ed5.

SEAC esp.V. Datos del sistema.

Nombre:F20.
Descrip: Prueba del desempeño del alumno.
Est.no Est.: 015.

Resumen lógico del proceso

Lectura del nombre de cuenta
Busca el número de cuenta
Si el número existe
 Lectura de otros personales
 Escritura a la pantalla
 Mientras en esta sesión
 Lectura de los datos de la sesión
 Escritura de la sesión

Diseño de las pantallas.

El formato de la pantalla es el mismo de la captura de datos con la información de las sesiones de la siguiente forma:

Ses	Uni	Pruebas
1	2	01,F02 85%
2	2	02 100%, 001,F03 90%

Ses - Es el número de la sesión.

Uni - Es la unidad estudiada.

Pruebas - Son los números de los reactivos agrupados por prueba y el resultado.

Operación del proceso SEAC2

Para el manejo de las opciones de SEAC2 el menú cuenta con:

- Despliegue de la guía del alumno
- Aplicación de la instrucción
- Consulta del alumno
- Salida del proceso

Las operaciones de esta opción se especifican en archivos de tipo

SEAC cap.V. Flujo del proceso.

Nombre: F00.
Descrip: Flujo del desarrollo del alumno.
Est.de Desc: 005.

Resumen lógico del proceso

Lectura del número de cuenta
Búsqueda al número de cuenta
Si el número existe
 0 Lectura de otros personales
 Escritura a la pantalla
 Mientras exista sesión
 0 Lectura de los datos de la sesión
 Escritura de la sesión

Diseño de las pantallas

El formato de la pantalla es el mismo de la captura de datos con la información de las sesiones de la siguiente forma:

Ses	Uni	Pruebas
1	0	CR1,FV2 85%
2	0	CR2 100%, CR1,FV3 90%

Ses - Es el número de la sesión.

Uni - Es la unidad estudiada.

Pruebas - Son los números de los reactivos agrupados por prueba y el resultado.

Operación del proceso SEACC

Para el manejo de las sesiones de SEACC al manejar cuenta con:

- Despliegue de la guía del alumno
- Aplicación de la instrucción
- Consulta del alumno
- Salida del proceso

Las operaciones de esta opción se especifican en archivos de tipo

SEAC cap.V. Diseo del sistema.

.BAT cuyos nombres son a0001.bat a a0003.bat y sp002.bat.

La pantalla de SEAC2 es la siguiente:

Sistema de enseanza auxiliada por computadora			
M E N U P R I N C I P A L (para el alumno)			
Guia	Apli	Consul	Salida
Alumno	Instruc		
f1	f2	f7	f8

Guia Alumno f1

Despliega la gua del alumno utilizando las facilidades de los comandos de INSTRUCTOR.

Apli Instruc f2

Llama al programa INSTRUCTOR para el inicio del estudio de una Unidad.

Consulta f3

Llama al proceso de consulta para despliegue de resultados.

Salida f8

Regresa el control a MS-DOS

VI) Discusión y conclusiones:

En cuanto al análisis del sistema podemos anotar:

- a) El modelo de Anderson y Faust el cual se intentó seguir paso a paso, sirvió únicamente como guía y permitió determinar las facilidades que se requieren para la elaboración y aplicación de un curso.
- b) Se procuró que estas facilidades fueran flexibles, de tal forma que el instructor hiciera un libre uso de ellas.
- c) Se procuró trascender la simple repetición de un texto por la computadora. Para ello, se elaboraron programas que: manejan ejemplos con salidas gráficas y que son susceptibles de ser conducidos por el alumno, otros que permiten administrar la instrucción y por último aquellos que capacitan al instructor.

En cuanto al diseño del sistema podemos anotar:

- a) La utilización de un editor de texto es suficiente para la captura y modificación de los contenidos, sin embargo no lo es para la aplicación de la instrucción.
- b) Se desarrolló un conjunto de comandos que se incluyen en los contenidos - para: el despliegue de la información, la especificación de preguntas, la integración de preguntas en pruebas, la aplicación y evaluación de pruebas y la aplicación de ejemplos.
- c) Las preguntas se especifican en forma sencilla, basta con indicar la posición y longitud de la respuesta para que el sistema reserve esta área para trabajo del alumno. Con esta estrategia es posible especificar prácticamente cualquier tipo de pregunta.
- d) No se encontró una forma sencilla de manejar los aspectos gráficos y las

SEAC cap.VI. Discusión y Conclusiones.

operaciones sobre ellos. En el caso de su aplicación al material de Estructura de Datos, se desarrollaron programas específicos para lograrlo. En el ejemplo de Pila Contigua se muestran los límites de la pila, el tope y los elementos que en cada momento tiene la pila. Sobre esta estructura es posible agregar y retirar elementos y conforme se aplican las operaciones se muestra el crecimiento o reducción de la pila entre los límites de implementación. El tope se mueve de acuerdo a las operaciones que se realizan.

e) Se requirieron una gran cantidad de operaciones para el manejo de la pantalla y la lectura de datos.

En cuanto a la programación podemos anotar:

El sistema se programó en el lenguaje Pascal y se logró una estructura a base de funciones primitivas que permiten una fácil y rápida modificación o adaptación del sistema ya sea para agregar nuevas facilidades o para transportar el sistema a otros equipos.

En cuanto a la instalación podemos anotar:

Se utiliza poco espacio de memoria secundaria para los programas ya que únicamente se requieren dos discos. Uno de ellos contiene las facilidades para el instructor y el otro las del alumno.

*En cuanto a la metodología utilizada en el desarrollo podemos anotar:
(Análisis Estructurado de Yourdon)*

No fue fácil su utilización debido al tipo de sistema. Los datos, su estructura y las operaciones sobre ellos no fueron fácilmente especificados con

SEAC cap.VI. Discusión y conclusiones.

Las herramientas del análisis estructurado.

De manera general también podemos decir:

- a) El sistema maneja el material de Estructura de Datos, sin embargo puede ser utilizado en cursos en los que la presentación de los contenidos sea similar.*
- b) Como complemento a un curso tradicional, puede ser fuertemente orientado a la práctica de conceptos y procedimientos.*
- c) Con respecto a los objetivos inicialmente planteados en este trabajo, es necesario utilizar este sistema en la práctica para contar con mayores datos que permitan una evaluación mas completa.*

GUIA PARA LA ELABORACION DEL MATERIAL Y USO DE LOS RECURSOS DE S.E.A.C.

En el presente manual se explican detalladamente los pasos que hay que seguir en la preparación de los contenidos de cada una de las unidades que integran el material de Estructuras de Datos.

~

Paso 1.

El material debe dividirse en 'unidades'.

Cada 'unidad' debe integrarse de las siguientes 'secciones':

- Objetivos
Los objetivos pueden dividirse en generales y específicos.
- Índice
El índice es la especificación del contenido.
- Contenido
El desarrollo del contenido lo constituyen definiciones, explicaciones, ejemplos, ejercicios, evaluaciones, etc con lo cual esperamos que el alumno alcance los objetivos.
- Bibliografía
- Resumen

~

Para la captura de los contenidos se selecciona la opción 'Editor' tecla 'F2' la cual habilita un programa editor de texto. El tipo de texto debe ser 'documento' y el nombre del archivo debe ser 'UNIDAD#' donde # es 0,1,2,...,n. Después de instruir con estos dos datos al editor estamos en posibilidad de introducir el contenido de la unidad. El estilo, está totalmente abierto al instructor y puede incorporar los siguientes comandos dentro del texto para facilitarse el trabajo:

~

Back Este caracter colocado al principio de una línea indica
Slash cambio de hoja. Se utiliza para indicar que cierta
parte del contenido se despliegue limpiando la
pantalla.

Back Esta secuencia indica que se ejecute el ejemplo marcado
SlashE# con #. # puede ser cualquier caracter del teclado.
Estos ejemplos se programan y se incluyen a la
biblioteca RIBISEAC.PAS como PROCEDURE XX y se modifica
el procedure EJEMPLO incorporando un 'if c=# then XX'.

~

Back Esta secuencia indica que se ejecuten los reactivos
SlashPa##, cuyos nombres son ss#,ss#,.. los cuales constituyen
ss#,...;# una prueba. Los nombres son dos letras cualquiera
seguida de un número de un solo dígito. Al finalizar
los nombres de los reactivos se puede colocar ';' seguido de un número, esto indica la calificación

mínima que pueda obtener el alumno para continuar satisfactoriamente con el material.
Ejemplo:Back SlashOM1,RC2;80

¶

La evaluación se hace considerando que cada reactivo tiene un peso del 100%. Si en una prueba hay tres reactivos la calificación en cada reactivo se promedia.

A continuación encontrará un ejemplo de un trozo de material que incluye: objetivos, índice, definiciones, explicaciones, ejemplos, evaluaciones y bibliografía.

¶

UNIDAD IV Estructuras de datos compuestas: Listas no lineales.

Back Slash

Objetivo General

El alumno al finalizar la unidad, elaborara algoritmos computacionales que utilicen estructuras de datos no lineales.

Back Slash

Objetivos Especificos

1. Reconocera una estructura no lineal.
2. Identificará los elementos de una gráfica.
3. Representará gráficas en la computadora.
4. Escribirá algoritmos de recorrido de gráficas.

Back Slash

Índice

Introducción
Conceptos y definiciones de gráficas
Representación de gráficas
Arboles
Arboles binarios
Algoritmos

Back Slash

Introducción

En la unidad III se precisó que una lista lineal es una estructura de datos en la que las relaciones entre los elementos se define por un solo criterio; el propósito de esta unidad es estudiar estructuras mas complejas tales como las gráficas en las que las relaciones entre los elementos son en más de un criterio. Se ha separado el estudio de los arboles debido a la importancia de este material.....

(de esta forma continuamos introduciendo el material)

Back SlashEs

(estamos indicando que en este punto se ejemplifique 'a')

Back SlashPOM3

(estamos indicando que se aplique el reactivo OM3)

Back SlashFFV2,OM6,RC1;95

(estamos indicando que se apliquen los reactivos FV2,OM6 y RC1 y

que se verifique si el grado de ejecución es mayor o igual a 95%)

~

A continuación se verá como se despliega el material al alumno

~

UNIDAD IV Estructuras de datos compuestas: Listas no lineales.

~

Objetivo General

El alumno al finalizar la unidad, elaborará algoritmos computacionales que utilicen estructuras de datos no lineales.

~

Objetivos Específicos

1. Reconocerá una estructura no lineal.
2. Identificará los elementos de una gráfica.
3. Representará gráficas en la computadora.
4. Escribirá algoritmos de recorrido de gráficas.

~

Índice

Introducción
 Conceptos y definiciones de gráficas
 Representación de gráficas
 Árboles
 Árboles binarios
 Algoritmos

~

Introducción

En la unidad III se precisó que una lista lineal es una estructura de datos en la que las relaciones entre los elementos se define por un solo criterio; el propósito de esta unidad es estudiar estructuras mas complejas tales como las gráficas en las que las relaciones entre los elementos son en más de un criterio. Se ha separado el estudio de los árboles debido a la importancia de este material.....

(de esta forma continuaría el alumno viendo el material)

~

Paso 2

Para la captura de los reactivos se selecciona la opción 'Editor' tecla 'F2' la cual sibilita un programa editor de texto. El tipo de texto debe ser 'documento' y el nombre del archivo debe ser 'REACT#' donde # es 0,1,2,...,n. Después de instruir con estos dos datos al editor estamos en posibilidad de introducir los reactivos de acuerdo al siguiente formato.

~ Back 'as' se sugiere que sean las dos primeras letras del Slashaa# tipo de reactivo y # un número. De esta forma para un reactivo de opción múltiple su identificación podría ser OM1.

2

- ? En la especificación del reactivo es posible colocar en cualquier posición uno o más símbolos de interrogación y estos señalarán el lugar en donde se espera una respuesta del alumno. Siguiendo a la interrogación se coloca la respuesta. Inmediatamente después que el alumno da su respuesta esta se cotaja y en caso de ser errónea se le hace saber haciendo sonar la campana.

2

A continuación se dan ejemplos para los tipos de reactivos sugeridos.

-Opción múltiple

Back SlashOM1

Es importante en la gráfica que $\langle u,v \rangle$ sea diferente que $\langle v,u \rangle$

- 1 Gráfica pasada
- 2 Gráfica dirigida
- 3 Gráfica no dirigida
- 4 Gráfica cíclica

Respuesta (??2)

2

-Falso verdadero

Back SlashFV1

Una gráfica pasada es aquella que tiene un valor asociado en los arcos.

Falso o Verdadero (??V)

2

-Jerarquización

Back SlashJQ1

Para el siguiente árbol de el orden de visita de los siguientes nodos si se recorre en preorder. $A(B(D,E),C(F(H,I),G))$

C(?5)	D(?3)	A(?1)	B(?2)	G(?9)
F(?6)	I(?8)	E(?4)	H(?7)	

2

-Completamiento

Back SlashCO1

Los arcos de la forma $\langle a1,a2 \rangle, \langle a2,a3 \rangle, \langle a3,a4 \rangle, \dots$ constituyen una ??????????????????trayectoria

2

-Relación de columnas

Back SlashRC1

1 Lista no lineal	(23) A X B
2 Hoja	(21) Arbol
3 Producto cartesiano	(22) Nodo con grado externo igual a cero
4 Raiz	(24) Nodo con grado interno igual a cero

~

A continuación se mostrará como el alumno ve los reactivos aquí explicados. Todos están agrupados en una sola prueba POM1,FV1,JG1,CO1,RC1;80 y con ejecución mínima del 80%.

NPOM1,FV1,JG1,CO1,RC1;80

Paso 3

Elaborar la guía para el alumno.

La guía para el alumno la constituyen las instrucciones del instructor acerca de como debe hacer uso del material y del sistema. La opción 'Editor' tecla 'F2' sirve para elaborar un archivo de tipo documento con el nombre 'GUIAALU' para ser desplegado al alumno antes de iniciar su instrucción.

También se recomienda que se incorpore parte de este documento en la UNIDAD.

~

***** Fin de la Guía del Instructor *****

GUIA DEL ALUMNO

Para un mejor uso de este material deberán seguirse las siguientes instrucciones. Cada unidad se ha elaborado siguiendo criterios didácticos que se consideran útiles para mejorar la enseñanza y el aprendizaje. Al inicio de la unidad se establecen los objetivos de aprendizaje con el objeto de dar a conocer el comportamiento buscado. También se incorpora un índice que junto con los objetivos permiten obtener una visión general de la estructura y organización del contenido.

En el contenido existen ejemplos que se han elaborado haciendo uso de las capacidades gráficas y que se ofrecen en modo de simulación. Estos ejemplos requieren de la participación del usuario.

También a lo largo del contenido se encuentran ejercicios y pruebas, las cuales deben de contestarse. Los resultados del desempeño se exhibe y se da a conocer en el momento de las respuesta. Si la respuesta no es correcta suena la campana. En algunos casos al finalizar la aplicación de un ejercicio o prueba se dan instrucciones parao continuar con el estudio.

Al final de la unidad se encuentra una prueba de autoevaluación que debe de contestarse. Al terminarla se conocen los resultados del desempeño.

Se agrega también al final una bibliografía para ampliar la información sobre lo expuesto.

Para ingresar al sistema se pide el numero de cuenta y en caso de no estar registrado se piden algunos datos complementarios; inmediatamente despues se pide el numero de la unidad que se quiere estudiar y si existe entras al sistema.

Nota: Las respuestas en caso de ocupar todo el espacio marcado no requieren que al final se oprimas la tecla 'RETURN'

***** Fin de la guía del alumno *****

Este apendice contiene la biblioteca y los programas del sistema de Enseñanza auxiliada por computadora.

{Biblseac.pas: es la biblioteca de programas de SEAC}
{JUNIO 22 de 1985- Fac. Ing. de la UNAM, CINVESTAV, IIMAS}

```

()
{funcion para leer caracter por caracter del teclado}
function dos:qq(command,parameter:word):byte;extern;

()
procedure modo(JGO:char);
{selecciona el modo de operar de la pantalla}
{hay 3 modos normal @ grafico R matematico A}
begin
    write(chr(27),'')',JGO);
end;

()
procedure funtec(const numtec,encab,mensa:string);
{asigna funciones a las teclas}
{ejemplo: funtec('1','nombtecl','envia')}
begin
    write(chr(27),'%f0a',numtec,'K2vi:8d&L',encab,mensa,chr(13));
end;

()
procedure abilita;
{habilita la asignacion de funciones}
begin
    write(chr(27),'%jB');
end;

()
procedure desabil;
{desabilita las funciones asignadas a las teclas}
var i:integer;
begin
    write(chr(27),'%j@');
    for i:=1 to 8 do funtec(chr(48+i),' ',' ');
    abilita;
end;

()
procedure retardo(n:integer);
{retarda la ejecucion n segundos}
var i:integer;
begin
    for i:=1 to n do write(chr(27),'@');
end;

```



```

{ }
procedure enha(OPCION:char);
{opciones borra @,inverso B,subraya D,medio brillo H hay otras ver man}
begin
    write(chr(27),'&d',OPCION);
end;

{ }
procedure lonlin(var a:string;var b:integer;c:integer);
{calcula la longitud de un string b. c longitud maxima}
begin
    b:=c;
    while (a[b]=' ') and (b>0) do b:=b-1;
end;

{ }
procedure cursor(RENGLON,COLUMNA:integer);
{coloca el cursor en cualquier punto de la pantalla entre 0,0 y 23,79}
const DIEZ=10;
var RDIGITO1,RDIGITO2,CDIGITO1,CDIGITO2: integer;
begin
    RDIGITO1:=RENGLON div DIEZ;
    RDIGITO2:=RENGLON mod DIEZ;
    CDIGITO1:=COLUMNA div DIEZ;
    CDIGITO2:=COLUMNA mod DIEZ;
    write(chr(27),'&a',chr(48+CDIGITO1),chr(48+CDIGITO2),'&c');
    write(chr(48+RDIGITO1),chr(48+RDIGITO2),'&r');
end;

{ }
procedure cursorderecha;
{mueve el cursor una posicion a la derecha}
begin
    write(chr(27),'&D');
end;

{ }
procedure cursorabajo;
{mueve el cursor 1 posicion hacia abajo}
begin
    write(chr(27),'&B');
end;

{ }
procedure memlock(ren:integer);
{pone un candado a la memoria desde ren}
begin
    cursor(ren,0);
    write(chr(27),'&l');
end;

procedure unmemlock;
{abre candado de la proteccion a memoria}
begin

```

```
    write(chr(27),'m');
end;

()
procedure campana(n:integer);
{suena la campana n veces}
var i:integer;
begin
    for i:=1 to n do write(chr(7));
end;

()
procedure limcar;
{limpia la caratula del sistema}
var i:integer;
begin
    cursor(1,1);
    for i:=1 to 78 do write(' ');
    cursor(3,1);
    for i:=1 to 78 do write(' ');
    cursor(5,0);
    write(chr(27),'J');
end;

()
procedure formatmode(r1,c1,n:integer;x:char);
{solo permite lectura y escritura en la pantalla desde r1,c1 n caracteres
{x es el tipo de embellecimiento del campo}
var i:integer;
begin
    cursor(r1,c1);
    enha(x);
    write(chr(27),'I');
    for i:=1 to n do write(' ');
    write(chr(27),'J');
    enha('@');
end;

()
procedure eformatmode;
{desabilita las protecciones en la pantalla}
begin
    write(chr(27),'W');
end;

()
procedure dformatmode;
{habilita las protecciones en la pantalla}
begin
    write(chr(27),'X');
end;
```

⌋

```

procedure limpant(ren:integer);
{limpia totalmente la pantalla desde el renglon ren}
begin
    cursor(ren,0);
    write(chr(27),'J');
end;

```

⌋

```

procedure limplin(ren,col:integer);
{limpia la linea ren desde col hasta el final de linea}
begin
    cursor(ren,col);
    write(chr(27),'K');
end;

```

⌋

```

procedure lector(var linea: cadena; max:integer;var cuent,band:integer);
{lee del teclado una cadena de caracteres en la variable linea}
{cuent numero de caracteres leidos}
{max maximo numero de caracteres que pueden leerse}
{si band=1 solo se dio carry return}
{si se da back space se borra un caracter}

```

```

const    trece=13;
label    1;
var      cod:byte;
begin
    band:=0;cuent:=0;
    repeat
    1:      cod:=dosxqq(1,0);
           if (cod=8) and (cuent=0) then goto 1;
           if (cod=8) and (cuent>=0) then
           begin
               cuent:=cuent-1;
               goto 1;
           end;
           if (cod=trece) and (cuent=0) then begin band:=1;return;
           if (cod=trece) then return;
           cuent:=cuent+1; linea[cuent]:=chr(cod);
    until (cuent=max);
end;

```

⌋

```

procedure decharn(var caracteres:string;desde,cuantos:integer;
                 var numero:integer);
{convierte de caracteres a numero}
var i,j:integer;
    potencia: array [1..5] of integer;
begin
    potencia[1]:=1;potencia[2]:=10;potencia[3]:=100;potencia[4]:=1000;
    potencia[5]:=10000;
    numero:=0;
    j:=0;
    for i:=1 to cuantos do
    begin
        if (caracteres[desde+cuantos-i]) in ['0'..'9'] then
        begin
            j:=j+1;
            numero:=numero+(ord(caracteres[desde+cuantos-i])
                          -48)*potencia[j];
        end;
    end;
end;

```

```

C)
procedure pantalla;
{dibuja el formato de la pantalla}
var I:integer;
begin
    limpart(0);
    cursor(0,0);
    modo('K');
    enha('H');
    write(chr(14),'Q');for I:=0 to 77 do write(';');write('W',chr(15));
    cursor(1,0);enha('H');write(chr(14),':',chr(15));enha('@');
    cursor(1,77);enha('H');write(chr(14),':',chr(15));enha('@');
    cursor(2,0);
    enha('H');
    write(chr(14),':');for I:=0 to 77 do write(',');write(':',chr(15));
    cursor(3,0);enha('H');write(chr(14),':',chr(15));enha('@');
    cursor(3,77);enha('H');write(chr(14),':',chr(15));enha('@');
    cursor(4,0);
    enha('H');
    write(chr(14),'A');for I:=0 to 77 do write(';');write('S',chr(15));
    cursor(5,0);
end;

```

```

procedure finbufin(var buf:lcadena;var lol:integer);
{pone una marca al final del bloque}
begin
    lol:=lol+1;
    buf[lol]:=chr(1);
end;

```

```

procedure listabuf(var buf:lcadena);
{lista el contenido de buf hasta encontrar chr(1)}
var
  nl,i:integer;
begin
  i:=1;
  nl:=0;
  while buf[i]<>chr(1) do
  begin
    if buf[i]=chr(0) then
      begin
        writeln;
        i:=i+1;
        nl:=nl+1;
        if (nl mod 4)=0 then
          begin
            readln;
            cursor(18,0);
          end;
        end;
      end
    else
      begin
        write(buf[i]);
        i:=i+1;
      end;
    end;
  end;
  readln;
end;

```

```

{}
procedure leecub(numreg:integer;var cubeta:reg1);
{lee en la posicion numreg del archivo de cubetas}
begin
  seek(b,numreg);
  read(b,cubeta);
end;

```

```

{}
procedure escub(numreg:integer;var cubeta:reg1);
{escribe en la posicion numreg del archivo de cubetas}
begin
  seek(b,numreg);
  write(b,cubeta);
end;

```

```

{}
procedure leedat(numreg:integer;var datos:reg2);
{lee en la posicion numreg del archivo de datos}
begin
  seek(a,numreg);
  read(a,datos);
end;

```

```

{}
procedure escdst(numreg:integer;var datos:reg2);
{escribe en la posicion numreg del archivo de datos}
begin
    seek(a,numreg);
    write(a,datos);
end;

{}
procedure denumc(var caracteres:string;desde,cuantos,numero:integer);
{cambia un numero entero a cadena de caracteres}
var
    i:integer;
begin
    i:=desde+cuantos-1;
    while (numero<>0) and (i>=desde) do
        begin
            caracteres[i]:=chr((numero mod 10)+48);
            numero:=numero div 10;
            i:=i-1;
        end;
    while i>=desde do begin caracteres[i]:=' ';i:=i-1; end;
end;

{}
procedure buscencu(cubeta:reg1;llave:cadena;longllave:integer;
    var apuntador,posicion:integer);
{busca una llave en la cubeta}
var
    liga:array [1..4] of char;
    i,j,dif: integer;
begin
    i:=0;j:=0;dif:=1;posicion:=0;
    while (dif=1)and(i<119) do
        begin
            i:=i+1;
            if (i mod (longllave+4))=0 then
                begin
                    posicion:=posicion+1;
                    j:=0;
                    dif:=0;
                    while (j<longllave) and (dif=0) do
                        begin
                            j:=j+1;
                            if cubeta[i-(longllave+4)+j]<>llave[j] +
                                dif:=1;
                        end;
                    end
                end;
            if dif=0 then
                decharn(cubeta,i-3,4,apuntador)
            else
                apuntador:=0;
        end;
end;

```

{}

```

procedure hash(llave:cadena;ncubetas,ncar:integer;var direccion:integer);
{calcula el numero de la cubeta a la que va la llave}
{la funcion es la propuesta por Floyd's con c=.618034}
var
  nument,num:real4;
  i:integer;
  ch:char;
begin
  num:=0.0E0;
  for i:=1 to ncar do num:=numford(llave[i])*i;
  num:=0.618034*num;
  nument:=trunc(num);
  direccion:=trunc(ncubetas*(num-nument) + 2);
end;

```

{}

```

procedure busca(llave:cadena;longllave:integer;
               var ultimo,posicion,apuntador:integer);
{busca en las cubetas la direccion de un registro}
{si apuntador=0 la llave no existe}
{si apuntador<>0 esa es la posicion del registro en datos}
{en ultimo regresa el ultimo registro buscado}
{en posicion regresa la posicion dentro del registro}
var
  liga,direccion:integer;
  cubeta:reg1;
begin
  apuntador:=0;
  hash(llave,10,longllave,direccion);
  while (direccion<>0) and (apuntador=0) do
  begin
    leecub(direccion,cubeta);
    if b.errs = 3 then
    begin { no existe esta cubeta}
      ultimo:=direccion;
      direccion:=0;
      b.errs:=0;
    end
    else
    begin
      decharrn(cubeta,124,4,liga);
      buscencu(cubeta,llave,longllave,apuntador,posic);
      if (liga=0) or (apuntador<>0) then ultimo:=direccion:=liga;
    end;
  end;
end;
end;

```

{}

```

procedure mensaje(clave:integer);
{envia mensajes a los usuarios}
var   linea:cadena;
      band,ncar:integer;
begin
  dformatmode;
  cursor(22,10);anha('B');
  case clave of
    0:   write('error 0: no existe la llave. ');
    1:   write('error 1: ya existe la llave. ');
    2:   write('error 2 ');

    end;
  write('Pres RETURN para cont.. ');
  formatmode(22,70,1,'N');
  eformatmode;
  cursor(22,70);lector(linea,1,ncar,band);
  dformatmode;
end;

{}
procedure empaca(var datos:string;linea:cadena;desde,cuantos:integer);
{empaca los datos en el registro}
var   i:integer;
begin
  for i:=1 to cuantos do datos[desde+i-1]:=linea[i];
end;

{}
procedure limpbuf(var x:string;desde,hasta:integer);
var   i:integer;
begin
  if (desde<=hasta) then for i:=desde to hasta do x[i]:=' ';
end;

{}
procedure agregacub(ultimo:integer;llave:cadena;longllave:integer;
                   var ndatos:integer);
var   ncubetas,nllaves,i:integer;
      numdat,datos:reg2;
      numcub,cubeta:reg1;
      linea:cadena;
begin
  leecub(ultimo,cubeta);
  decharn(cubeta,120,4,nllaves);
  if b.errs =3 then
  begin nllaves:=0;b.errs:=0;limpbuf(cubeta,1,127);end;
  if nllaves < ((119 div (longllave+4))
  then
    begin      (la llave entra a este registro de la cubeta)

```



```

nllaves:=nllaves+1;
leedat(1,numdat);
decharn(numdat,1,4,ndatos);
if a.errs=3 then
  begin ndatos:=1;a.errs:=0;limpbuf(numdat,1,255);end;

ndatos:=ndatos+1;
empaca(cubeta,llave,(nllaves-1)*(longllave+4)+1,longllave+4);
denumc(cubeta,(nllaves-1)*(longllave+4)+longllave+1,4,ncubetas);
denumc(cubeta,120,4,nllaves);
escub(ultimo,cubeta);
denumc(numdat,1,4,ndatos);
escdat(1,numdat);

end
else (la llave se incorpora a un nuevo registro de la cubeta)
begin
leecub(1,numcub);
decharn(numcub,1,4,ncubetas);
ncubetas:=ncubetas+1;
if ncubetas=0 then ncubetas:=11;
limpbuf(numcub,1,127); (describe reg 0 de cubetas)
denumc(numcub,1,4,ncubetas);
escub(1,numcub);
denumc(cubeta,124,4,ncubetas); (liga las cubetas)
escub(ultimo,cubeta);
limpbuf(cubeta,1,127); (describe la nueva cubeta)
empaca(cubeta,llave,1,longllave);
leedat(1,numdat);
decharn(numdat,1,4,ndatos);
ndatos:=ndatos+1;
denumc(cubeta,longllave+1,4,ndatos);
escub(ncubetas,cubeta);
limpbuf(numdat,1,255); (describe reg 0 de datos)
denumc(numdat,1,4,ndatos);
escdat(1,numdat);

end;

end;

()
procedure escritor(var datos:string;desde,cuantos:integer);
var i:integer;
begin
  if cuantos>0 then for i:=desde to desde+cuantos-1 do write(datos);
end;

()
procedure quitacub(dt:integer;ultimo,posicion,longllave:integer);
var cubeta:regi;
    llave:cadena;
    i:integer;
begin
  leecub(ultimo,cubeta);
  for i:=1 to longllave+4 do llave[i]:=' ';
  empaca(cubeta,llave,(posicion-1)*(longllave+4)+1,longllave+4);

```

```

        escub(ultimo,cubeta);
end;

()
procedure escdat1(var apuntador:integer;var bufin:lcadena;
                 longbufin:integer);
{escribe registros ligados a los archivos de datos}
var   j,i,liga:integer;
      datos:reg2;
begin
    j:=0;
    while longbufin>0 do
    begin
        limpbuf(datos,1,255);
        i:=0;
        while (i<251) and (longbufin>0) do
        begin
            longbufin:=longbufin-1;
            j:=j+1;
            i:=i+1;
            datos[i]:=bufin[j];
        end;

        if longbufin>0 then
        begin
            liga:=apuntador+1;
            denumc(datos,252,4,liga);
            escdat(apuntador,datos);
            apuntador:=apuntador+1;
        end
        else
            escdat(apuntador,datos);
        end;
        limpbuf(datos,1,255);
        denumc(datos,1,4,apuntador);
        escdat(1,datos);
    end;
end;

```

```

()
procedure leedat1(var apuntador:integer;var bufin:lcadena);
{lee del archivo de datos registros ligados}
var   j,i,liga:integer;
      datos:reg2;
begin
    j:=0;liga:=apuntador;
    while liga<>0 do
    begin
        leedat(liga,datos);
        for i:=1 to 251 do begin
            j:=j+1;
            bufin[j]:=datos[i];
        end;
        decharr(datos,252,4,liga);
    end;
end;

```

```

end;

program seac1(input,output);
{programa para desplegar las opciones que tiene el instructor en SFAC}

{se incluye la biblioteca}
{$include:'bib0seac.pas'};

{caratula de opciones del proceso seac1}
begin
    dformatmode;
    pantalla;
    cursor(1,12);
    write('Sistema de enseñanza auxiliado por computadora');
    cursor(3,6);
    write(' M E N U   P R I N C I P A L   SEAC1   (Para el Instructo
    cursor(5,0);
{asigna funciones a las teclas}
    desabil;
    funtec('1','GUIA Ins','m0001');
    funtec('2','Editor ','m0002');
    funtec('3','Consulta','m0003');
    funtec('B',' Salida ','sp001');
    abilita;
    dformatmode;formatmode(5,0,1,'S');eformatmode;

end.

Comando: m0001.bat
ren guains cualquiera
leesec
ren cualquiera guains
seac1

Comando: m0002.bat
ws
seac1

Comando: m0003.bat
consulta
seac1

program sp001(input,output);
{$include:'bib0seac.pas'}
begin
    cursor(0,0);
    limpant(0);
    dformatmode;

end.

```

EST. TERCER UN. REBE
SALIR DE LA BIBLIOTECA

```

program leesec(input,output);
{programa de aplicacion de textos}

const   longpag=16;

type    cadena= string (80);
        buff1=string(32000);    {contenido}

var     x:text;
        l:buff1;
        s,t,llave,lines,comando:cadena;
        pilal,pila:array[1..64] of integer;
        totpreg,numpreg,totpregml,pregml,topei,tope,
        band,ultimo,posicion,ncar,apuntador:integer;
        bandl,pagina,nl,n,nn,np,np1,o,ncl,ncm,contcar,inic:integer;
        carl,car:char;

{se incluye la biblioteca}
{#include:'bib0seac.pas'}

procedure pantinst;
{rellena pantalla del proceso instructor}
begin
    cursor(1,12);
    write('Sistema de ensenanza auxiliado por computadora');
    cursor(3,12);
    write('INSTRUCTOR');
    cursor(5,0);
end;

procedure entrada;
{lectura del numero de la unidad y nombres de archivos}
begin
    assign(x,'datos');
    x.mode:=sequential;
    reset(x);
end;

procedure asignaft;
{asigna funciones a las teclas}
begin
    desabil;
    funtec('1','Siguiet','n   ');
    funtec('2','Anterior','a   ');
    funtec('3','Primero ','p   ');
    funtec('4','Ultimo  ','u   ');
    funtec('5','Cuad. No','c   ');
end;

```

```

    funtec('8', ' Salida ', 's ');
    abilita;
end;

procedure lecbuff1(var c:char);
(lectura de buff1 carac x carac)
begin
    if ord(l[n])>128 then l[n]:=chr(ord(l[n])-128);
    c:=l[n];
    n:=n+1;
end;

procedure leebuff1;
(lee del archivo unidad a buff1)
begin
    nl:=0;
    tope:=1;
    ncl:=0;
    band1:=0;
    pila[tope]:=1;
    while not eof(x) do
        begin
            while not eoln (x) do
                begin
                    ncl:=ncl+1;
                    l[ncl]:=x^;
                    if (l[ncl]='N') and (band1=1) then
                        begin
                            nl:=0;
                            tope:=tope+1;
                            pila[tope]:=ncl;
                        end;
                    band1:=1;
                    if l[ncl]=chr(10) then
                        begin
                            nl:=nl+1;
                            if (nl mod longpag)=0 then
                                begin
                                    nl:=0;
                                    tope:=tope+1;
                                    pila[tope]:=ncl+1;
                                    band1:=0;
                                end;
                            end;
                    end;
                    get(x);
                end;
            get(x);
            ncl:=ncl+1;
            l[ncl]:=chr(13);
            ncl:=ncl+1;
            l[ncl]:=chr(10);
            nl:=nl+1;
            if (nl mod longpag)=0 then
                begin

```

```

        band1:=0;
        n1:=0;
        tope:=tope+1;
        pila[tope]:=ncl+1;
    end;
end;

end;

procedure eschoja;
{impresion de una hoja o pagina }
var c:char;
begin
    n:=pila[tope];
    if (tope+1)>np then nn:=ncl+1
        else nn:=pila[tope+1];
    repeat
        leebuff1(c);
        if c='%' then limpant(5)
            else write(c);
    until n=nn;
end;

begin
    {inicia parte principal}
    {asigna archivo de alumnos}

    dformatmode;
    pantalla;      {lineas pantalla}
    pantinst;     {pantalla instructor}
    entrada;      {abre archivos de la unidad}
    leebuff1;     {lectura de unidad}
    np:=tope;     {no. de hojas}
    nnp:=np;

    limpant(5);
    asignaft;     {asigna funciones}
    tope:=0;

    repeat
        readln(comando);      {lee función}
        if comando[1]='n' then {cuadro adelante}
        begin
            tope:=tope+1;
            if tope>nnp then tope:=nnp;
            cursor(3,3);write(tope:2);limpant(5);
            eschoja;
        end;

        if comando[1]='a' then {cuadro atras}
        begin
            tope:=tope-1;
            if tope<1 then tope:=1;
        end;
    until comando[1]='q';
end;

```

```

        cursor(3,3);write(tope:2);limpant(5);
        eschoja;
    end;

    if comando[1]='p' then {primero}
    begin
        tope:=1;
        cursor(3,3);write(tope:2);limpant(5);
        eschoja;
    end;

    if comando[1]='u' then {ultimo}
    begin
        tope:=nnp;
        cursor(3,3);write(tope:2);limpant(5);
        eschoja;
    end;

    if comando[1]='c' then {cualquiera}
    begin
        cursor(3,3);write('HoJ');
        dformatmode;formatmode(3,7,2,'J');aformatmode;
        cursor(3,7);lector(lines,2,ncar,band);
        dformatmode;
        cursor(3,3);anha('@');write(' ');
        if band=1 then
        begin
            cursor(3,3);write(tope:2);limpant(5);
            eschoja;
        end
        else
        begin
            if ncar=1 then decharn(lines,1,1,ncar)
            else decharn(lines,1,2,ncar);
            tope:=ncar;
            if ncar>nnp then tope:=nnp;
            if ncar<1 then tope:=1;
            cursor(3,3);write(tope:2);limpant(5);
            eschoja;
        end;
    end;
until comando[1]='s'; {termina}
close(x);
dformatmode;formatmode(5,0,1,'S');aformatmode;
end.

```

```

program consulta(input,output);
{reporte del desempeño del alumno}

const   longllave=8;
        longapunt=4;

type    cadena= string (80);
        reg1=string (127);
        reg2=string (255);
        lcadena=string(1);

var     a:text(256);{buffer de archivo de datos}
        b:text(128);{buffer del archivo de indices}
        llave,linea: cadena;
        cubeta:reg1;
        datos:reg2;
        calif,cont,sesion,band,ultimo,posicion,ncar,apuntador:integer;
{ultimo-es la ultima cubeta visitada}
{posicion-posicion dentro de la cubeta donde se encuentra la llave}
{apuntador-direccion de la informacion en el archivo de datos}

{se incluye la biblioteca de SEAC}
{finclude: 'bibiseac.pas'}

begin                                     {programa principal}
    dformatmode;
{caratula de opciones del proceso p001}
    pantalla;
    limcar;
    cursor(1,12);
    write('Sistema de ensenanza auxiliado por computadora');
    cursor(3,12);
    write('CONSULTA');
    cursor(5,0);
{asigna funciones a las teclas}
    desabil;
    abilita;

    assign(a,'aldatos.asc');              {se asigna archivo de datos}
    a.mode:=direct;
    rewrite(a);
    a.trap:=true;

    assign(b,'alindice.asc');             {se asigna archivo de indices}
    b.mode:=direct;
    rewrite(b);
    b.trap:=true;

```



```

limpant(5);cursor(5,0);
enha('H');writeln('NUMERO de cuenta ');
dformatmode;formatmode(5,20,8,'N');eformatmode;

cursor(5,20);lector(linea,longllave,ncar,band);
dformatmode;
if band=1 then return;
limpbuf(linea,ncar+1,longllave);
busca(linea,longllave,ultimo,posicion,apuntador);
if spuntador <>0 then
begin
    leedat(apuntador,datos);
    cursor(7,0);
    enha('H');writeln('fecha de hoy (ddmmaa)');
    enha('H');writeln('tu nombre');
    enha('H');writeln('numero de semestres estudiados');
    enha('H');writeln('No. de cursos que llevas esta semest');
    enha('H');writeln('ya tomaste el curso tradicional?');
    enha('H');writeln('es para extraordinario?');
    enha('H');write('ya tienes los antecedentes?');

    (protege la pantalla)
    dformatmode;
    formatmode(7,40,6,'N');
    formatmode(8,40,30,'N');formatmode(9,40,2,'N');
    formatmode(10,40,2,'N');formatmode(11,40,2,'N');
    formatmode(12,40,2,'N');formatmode(13,40,2,'N');
    eformatmode;

    (imprime los datos)
    cursor(7,40);escritor(datos,9,6);
    cursor(8,40);escritor(datos,15,30);
    cursor(9,40);escritor(datos,45,2);
    cursor(10,40);escritor(datos,47,2);
    cursor(11,40);escritor(datos,49,2);
    cursor(12,40);escritor(datos,51,2);
    cursor(13,40);escritor(datos,53,2);

    dformatmode;
    decharn(datos,55,3,ncar);
    cont:=59;
    sesion:=0;
    cursor(15,0);write('Ses Uni Calificaciones');
    while cont<ncar do
    begin
        writeln;
        sesion:=sesion+1;
        write(sesion:2,' ');
        write(datos[cont], ' ');
        cont:=cont+1;
        while datos[cont]<>'*' do
        begin
            if datos[cont]=' ' then
            begin

```

```

                                decharn(datos,cont,4,calif);
                                write(calif:3,'% ');
                                cont:=cont+4;
                                end
                                else
                                begin
                                if datos[cont]=',' then cont:=c;
                                else begin
                                write(datos[cont]);
                                cont:=cont+1 end;
                                end;
                                end;
                                cont:=cont+1;
                                end;
                                else
                                mensaje(0);
                                close(a);close(b);
                                dformatmode;formatmode(5,0,1,'S');eformatmode;
                                end.

```

```

program seac2(input,output);
{programa para desplegar las opciones que tiene el alumno en SEAC}

{se incluye la biblioteca}
{$include:'bib0seac.pas'};

{caratula de opciones del proceso seac1}
begin
    pantalla;
    cursor(1,12);
    write('Sistema de ensenanza auxiliado por computadora');
    cursor(3,6);
    write(' M E N U      P R I N C I P A L      SEAC2 (Para el Alumno)');
    cursor(5,0);
    {asigna funciones a las teclas}
    desabil;
    funtec('1','GUIA Alu','a0001');
    funtec('2','Instrucc','a0002');
    funtec('3','Consulta','a0003');
    funtec('8','Salida ','sp001');
    abilita;
end.

Comando: s0001.bat
ren guiaslu dstos
leesec
ren datos guiaslu
seac2

```

```
Comando: s0002.bat
```

```

instruc
seac2
Comando: a0003.bat
consulta
seac2

```

```

program instruc(input,output);
(programa de aplicacion del material)
{Instructor}

```

```

const   longllave=8;
        longspunt=4;
        longpag=16;

```

```

type    cadena= string (80);
        reg1=string (127);
        reg2=string (255);
        lcadena= string(1023);
        buff1=string(32000);   (contenido)
        buff2=string(10000);  (preguntas)

```

```

var     a:text(256);
        b:text(128);
        y,x:text;
        l:buff1;
        m:buff2;
        s,t,llave,linea,comando:cadena;
        cubeta:reg1;
        datos:reg2;
        bufin:lcadena;
        pila1,pila:array[1..64] of integer;
        totpreg,numpreg,totpregmal,pregmal,topel,tope,
        band,ultimo,posicion,ncar,apuntador:integer;
        band1,pagina,nl,n,nn,nnp,np,np1,o,ncl,ncm,contcar,inic:integer;
        car1,car:Char;
        resultado:real;
        nosigo:boolean;

```

{np- numero de hojas}

{nn - ultimo de la hoja actual}

{l y m - buffers msterisl y preguntas}

{pila - apuntadores al inicio de cada Hoja}

{tope - hoja actual}

{nl - contador de lineas}

{n y o - caracter actual de l y m}

{ncl y ncm - Longitud de l y m}

{car y car1 - caracter leído de l y m}

{ultimo-es la ultima cubeta visitada}

{posicion-posicion dentro de la cubeta donde se encuentra la llave}

{apuntador-direccion de la informacion en el archivo de datos}

```
{se incluye la biblioteca}
{$include:'bibiseac.pas'};
```

```
procedure adm1;
{marca el inicio del registro de sesion}
begin
    decham(datos,55,3,inic);
    contcar:=inic+1;
    datos[contcar]:=linea[1];
end;
```

```
procedure adm2;
{marca el nombre del reactivo}
var    i:integer;
begin
    contcar:=contcar+1;
    datos[contcar]:=',';
    for i:=1 to ncar do datos[i+contcar]:=linea[i];
    contcar:=contcar+ncar;
    contcar:=contcar+1;
    datos[contcar]:='-' ;
end;
```

```
procedure adm3;
{marca calificación del reactivo}
begin
    denumc(datos,contcar+1,4,trunc(resultado));
    contcar:=contcar+4;
end;
```

```
procedure adm4;
{marca el fin del registro de sesion}
begin
    contcar:=contcar+1;
    datos[contcar]:='*';
    denumc(datos,55,3,contcar);
    escdat(apuntador,datos);
end;
```

```
procedure pantinst;
{rellena pantalla del proceso instructor}
begin
    cursor(1,12);
    write('Sistema de enseñanza auxiliado por computadora');
    cursor(3,12);
    write('INSTRUCTOR');
    cursor(5,0);
end;
```

```
procedure entrada;
{lectura del número de la unidad y nombres de archivos}
label 1;
```

```

var      fname1,fname2:cadena;
begin
1:      dformatmode;
        limplin(7,0);
        cursor(7,0);enha('H');
        write('NUMERO de la unidad que quieres estudiar ');
        formatmode(7,51,1,'J');
        eformatmode;
        cursor(7,51);
        lector(lines,1,ncar,band);
        copystr('unidad',fname1);
        fname1[7]:=linea[1];
        copystr('pregun',fname2);
        fname2[7]:=linea[1];
        dformatmode;

{asigna archivo}
        if band=0 then
            begin
                assign(:,fname1);
                x.mode:=sequential;
                x.trap:=true;
                reset(x);
                if x.errs<>0 then
                    begin
                        x.errs:=0;
                        goto 1;
                    end;

                assign(y,fname2);
                y.mode:=sequential;
                reset(y);
                y.trap:=true;
            end;
        end;

end;

procedure asignaft;
{asigna funciones a las teclas}
begin
    desabil;
    funtec('1','Siguiente','n');
    funtec('2','Anterior','a');
    funtec('3','Primero','p');
    funtec('4','Ultimo','u');
    funtec('5','Cuad. No','c');
    funtec('8','Salida','s');
    abilita;
end;

procedure noacerto;
{indica que la respuesta fue incorrecta}
begin
    campana(3);
    pregmal:=pregmal+1;
end;

```

```

procedure lecbuff1(var c:char);
{lectura de buff1 carac x carac}
begin
    if ord(l[n])>128 then l[n]:=chr(ord(l[n])-128);
    c:=l[n];
    n:=n+1;
end;

procedure lecbuff2(var c:char);
{lectura de buff2 carac x carac}
begin
    if ord(m[o])>128 then m[o]:=chr(ord(m[o])-128);
    c:=m[o];
    o:=o+1;
end;

procedure pagrega(var tope,max:integer);
{agrega a una pila}
begin
    if tope=max then
    begin
        cursor(12,40);write('OVERFLOW');
        return;
    end
    else
    begin
        cursor(tope,10);write(' ');
        tope:=tope-1;
        cursor(tope,10);write('tope -->');
        cursor(tope,21);write('[xxxxxxxx]');
    end;
end;

procedure pretira(var tope,min:integer);
{retira de una pila}
begin
    if tope=min then
    begin
        cursor(12,40);write('UNDERFLOW');
        return;
    end
    else
    begin
        cursor(tope,10);write(' ');
        cursor(tope,21);write(' | ');
        tope:=tope+1;
        cursor(tope,10);write('tope -->');
    end;
end;

procedure pilaej;
{ejemplo de las operaciones sobre una pila}
var tope,max,min:integer;

```

```

begin
  tope:=15;          min:=15;          max:=10;
  limpant(5);
  cursor(6,10);write('Ejemplo sobre las operaciones sobre PILA');
  cursor(15,20);write('---MIN-----');
  cursor(14,25);write(':');
  cursor(13,25);write(':');
  cursor(12,25);write(':');
  cursor(11,25);write(':');
  cursor(10,25);write(':');
  cursor(9, 20);write('---MAX-----');

  desabil;
  funtec('1',' agregar','a   ');
  funtec('2',' retirar','r   ');
  funtec('3',' salida ','o   ');
  abilita;

  comando[1]:='a';
  while comando[1] <> 'o' do
  begin
    cursor(5,0);
    readln(comando);
    cursor(12,40);write(' ');
    if comando[1]='a' then pagrega(tope,max);
    if comando[1]='r' then pretira(tope,min);
  end;
end;

procedure cagrega(var u,p,max,min:integer);
{agrega a una cola}
begin
  if u=max then
  begin
    cursor(12,40);write('OVERFLOW');
    return;
  end
  else
  begin
    if u=0 then
    begin
      u:=min;p:=min;
      cursor(u,15);write('P -->');
      cursor(u,21);write('[XXXXXXXX]');
      cursor(u,32);write('<-- U');
    end
    else
    begin
      cursor(u,32);write(' ');
      u:=u-1;
      cursor(u,21);write('[XXXXXXXX]');
      cursor(u,32);write('<-- U');
    end
  end;
end;
end;

```

```

procedure cretira(var u,p,max,min:integer);
('retira de una cola')
begin
  if p=0 then
  begin
    cursor(12,40);write('UNDERFLOW');
    return;
  end
  else
  begin
    if u=p then
    begin
      cursor(p,15);write(' ');
      cursor(p,21);write(' ');
      cursor(p,32);write(' ');
      p:=0;u:=0;
    end
    else
    begin
      cursor(p,15);write(' ');
      cursor(p,21);write(' ');
      p:=p-1;
      cursor(p,15);write('P -->');
    end;
  end;
end;

```

```

procedure colaej;
('ejemplo de las operaciones sobre una cola')
var u,p,max,min:integer;

begin
  u:=0; p:=0; min:=20; max:=10;
  limpant(5);
  cursor(6,10);write('Ejemplo de las operaciones sobre COLA');
  cursor(21,20);write('---MIN-----');
  cursor(9,20);write('---MAX-----');

  desabil;
  funtec('1',' agregar','a ');
  funtec('2',' retirar','r ');
  funtec('3',' salids ','o ');
  abilita;

  comando[1]:='a';
  while comando[1]<>'o' do
  begin
    cursor(5,0);
    readln(comando);
    cursor(12,40);write(' ');
    if comando[1]='a' then cagrega(u,p,max,min);
    if comando[1]='r' then cretira(u,p,max,min);
  end;
end;

```



```

procedure ccagrega(var u,p,max,min:integer);
{agrega a una cola circular}
begin
  if ((u=max) and (p=min)) or (p=u-1) then
  begin
    cursor(12,40);write('OVERFLOW');
    return;
  end
  else if u=0 then
  begin
    u:=min;p:=min;
    cursor(u,15);write('P -->');
    cursor(u,21);write('[XXXXXXXX]');
    cursor(u,32);write('<-- U');

  end
  else if (u=max) and (p<>min) then
  begin
    cursor(u,32);write(' ');
    u:=min;
    cursor(u,21);write('[XXXXXXXX]');
    cursor(u,32);write('<-- U');

  end
  else
  begin
    cursor(u,32);write(' ');
    u:=u-1;
    cursor(u,21);write('[XXXXXXXX]');
    cursor(u,32);write('<-- U');

  end
end;

procedure ccretira(var u,p,max,min:integer);
{retira de una cola circular}
begin
  if p=0 then
  begin
    cursor(12,40);write('UNDERFLOW');
    return;
  end
  else if u=p then
  begin
    cursor(p,15);write(' ');
    cursor(p,21);write(' ');
    cursor(p,32);write(' ');
    p:=0;u:=0;

  end
  else if p=max then
  begin
    cursor(p,15);write(' ');
    cursor(p,21);write(' ');
    p:=min;
    cursor(p,15);write('P -->');

  end
end;

```



```

cursor(12,50);write(' C:LI(X)=n');
cursor(13,50);write(' B:LI(LI(n))=n');

desabil;
funtac('1',' A ','a ');
funtac('2',' B ','b ');
funtac('3',' C ','c ');
funtac('4',' D ','d ');
funtac('3',' salida ','o ');
abilita;

comando[1]:='a';
while comando[1]<>'o' do
begin
  cursor(5,0);
  readln(comando);
  if comando[1]='a' then
  begin
    cursor(16,30);write('---!-----');
    cursor(15,37);write('!');
    cursor(14,37);write('!');
    cursor(13,37);write('^');
    cursor(12,37);write('!');
  end;
  if comando[1]='b' then
  begin
    cursor(17,16);write('-----!--');
    cursor(16,16);write('!');
    cursor(15,16);write('!');
    cursor(14,16);write('!');
    cursor(13,16);write('^');
    cursor(12,16);write('!');
  end;
  if comando[1]='c' then
  begin
    cursor(10,22);write('-----!---');
    cursor(11,29);write('!');
    cursor(12,29);write('!');
    cursor(13,29);write('v');
    cursor(14,29);write('!');
  end;
  if comando[1]='d' then
  begin
    cursor(9,18);write('!---!-----');
    cursor(10,25);write('!');
    cursor(11,25);write('!');
    cursor(12,25);write('v');
    cursor(13,25);write('!');
    cursor(14,25);write('!');
  end;
end;
end;
end;

```

```

procedure leebuff1;
(lee del archivo unidad a buff1)
begin
  nl:=0;
  tope:=1;
  ncl:=0;
  band1:=0;
  pila[tope]:=1;
  while not eof(x) do
    begin
      while not eoln (x) do
        begin
          ncl:=ncl+1;
          l[ncl]:=x^;
          if (l[ncl]='$') and (band1=1) then
            begin
              nl:=0;
              tope:=tope+1;
              pila[tope]:=ncl;
            end;
          band1:=1;
          if l[ncl]=chr(10) then
            begin
              nl:=nl+1;
              if (nl mod longpag)=0 then
                begin
                  nl:=0;
                  tope:=tope+1;
                  pila[tope]:=ncl+1;
                  band1:=0;
                end;
            end;
          end;
          get(x);
        end;
      get(x);
      ncl:=ncl+1;
      l[ncl]:=chr(13);
      ncl:=ncl+1;
      l[ncl]:=chr(10);
      nl:=nl+1;
      if (nl mod longpag)=0 then
        begin
          band1:=0;
          nl:=0;
          tope:=tope+1;
          pila[tope]:=ncl+1;
        end;
    end;
end;

procedure leebuff2;
(lee del archivo preguntas a buff2)
begin
  tope1:=0;

```

```

ncm:=0;
while not eof(y) do
  begin
    while not eoln (y) do
      begin
        ncm:=ncm+1;
        m[ncm]:=y^;
        if (m[ncm]='N') then
          begin
            tope1:=tope1+1;
            pilai[tope1]:=ncm;
          end;
        get(y);
      end;
    get(y);
    ncm:=ncm+1;
    m[ncm]:=chr(13);
    ncm:=ncm+1;
    m[ncm]:=chr(10);
  end;
end;

procedure aplica;
{aplica una pregunta}
var
  ren,col,lol,lcn,numcar,j,i,K,mas:integer;
  delalumno:cadena;
  respuestas:array[1..32]of cadena;
  carxpreg:array[1..32]of integer;
  carxresp:array[1..32]of integer;
  rencol:array[1..32,1..2] of integer;

begin
  {tope>npi no existe esta pregunta}
  if tope1>npi then return;
  {si existe}
  {inicializa variables}
  band1:=0;
  dformatmode;
  limpart(5);
  pregms1:=0;
  numpreg:=0;
  ren:=5; col:=0;
  {termina la linea}
  lectbuff2(car1);lectbuff2(car1);
  {empieza el despliegue de la pregunta}
  while car1<>'N' do
    begin
      numcar:=0;
      lectbuff2(car1);
      if car1='?' then
        begin
          numpreg:=numpreg+1;{actualiza inicio campo}
          rencol[numpreg,1]:=ren;
          rencol[numpreg,2]:=col;
          while car1='?' do

```

```

begin
    numcar:=numcar+1;
    band1:=1;
    enha('J');
    write(chr(27),'[');
    write('?'); col:=col+1;
    write(chr(27),']');
    enha('@');
    lecbuff2(car1);
end;
carxpreg[numpreg]:=numcar;
end;
(sí acaba de colocar un campo lee respuesta)
if band1=1 then
begin
    band1:=0;
    mas:=0;
    while (car1<>' ') and (car1<>chr(13)) do
        begin
            mas:=mas+1;
            respuestas[numpreg,mas]:=car1;
            lecbuff2(car1);
        end;
    carxresp[numpreg]:=mas;
end;
(describe cualquier otro carácter)
if car1<>'N' then
begin
    write(car1);
    col:=col+1;
    if car1=chr(10) then
        begin
            col:=0;
            ren:=ren+1;
        end;
end;
end;

end;
(lectura de las respuestas del alumno)
eformatmode;
lol:=0;
for i:=1 to numpreg do
begin
    j:=1;
    cursor(rencol[i,1],rencol[i,2]);
    lector(linea,carxpreg[i],lon,band);
    if carxresp[i]=lon then
        begin
            while (linea[j]=respuestas[i,j]) and (j<=lon)
                do j:=1+j;
            if j<=lon then noacerto;
        end
    else
        noacerto;
end;
dformatmode;

```

```
end;
```

```
procedure buscap;
```

```
(busca la pregunta);
```

```
var i,j:integer;
```

```
begin
```

```
  tope1:=0;
```

```
  band1:=1;
```

```
  while band1=1 do
```

```
  begin
```

```
    tope1:=tope1+1;
```

```
    if tope1>np1 then return;
```

```
    band1:=0;
```

```
    o:=pils1[tope1];
```

```
    lecbuff2(car1);
```

```
    for j:=1 to ncar do
```

```
    begin
```

```
      lecbuff2(car1);
```

```
      if linea[j]<>car1 then band1:=1;
```

```
    end;
```

```
  end;
```

```
end;
```

```
procedure condicion;
```

```
(verifica el nivel de ejecucion y fija el tope);
```

```
var nivel,i:integer;
```

```
c:char;
```

```
begin
```

```
  lecbuff1(c);
```

```
  i:=0;
```

```
  while c<>chr(13) do
```

```
  begin
```

```
    i:=i+1;
```

```
    linea[i]:=c;
```

```
    lecbuff1(c);
```

```
  end;
```

```
  decharn(linea,1,i,nivel);
```

```
  if resultado<nivel then
```

```
  begin
```

```
    limpant(5);
```

```
    cursor(7,5);
```

```
    enha('B');
```

```
    write('El nivel de ejecucion logrado no fue suficiente');
```

```
    cursor(8,5);
```

```
    enha('B');
```

```
    writa('Se te recomienda repetir el estudio del material');
```

```
    retardo(3);
```

```
    cursor(7,5);enha('@');cursor(8,5);enha('@');
```

```
  end
```

```
  else
```

```
  begin
```

```
    limpant(5);
```

```
    cursor(7,5);
```

```
    enha('B');
```

```
    write('El nivel de ejecucion logrado te permite continuar');
```

```

        nnp:=np;
        retardo(3);
        cursor(7,5);enha('@');
    end;
end;

procedure prueba(var c:char);
{maneja las pruebas}
begin
    cursor(3,30);enha('B');write('[Estas en prueba]');
    totpreg:=0;
    totpregmal:=0;
    while (c<>chr(13)) and (c<>'') do
    begin
        ncar:=0;
        lecbuff1(c);
        while (c<>',') and (c<>chr(13)) and (c<>'') do
        begin
            ncar:=ncar+1;
            linea[ncar]:=c;
            lecbuff1(c);
        end;
        adm2; {registra nombre del reactivo}
        buscap;
        aplica;
        totpreg:=totpreg+numpreg;
        totpregmal:=totpregmal+pregmal;
    end;
    resultado:=100-((totpregmal*100)/totpreg);
    cursor(3,70);
    enha('C');
    write('Cal=',trunc(resultado):3,'%');
    retardo(3);
    cursor(3,70);
    enha('@');
    write(' ');
    adm3; {registra desempeno en prueba}
    cursor(3,30);enha('@');write(' ');
    if c<>chr(13) then condicion;
end;

procedure ejemplo;
{maneja los ejemplos}
var
    c:char;
begin
    lecbuff1(c);
    limpant(5);
    cursor(3,30);
    write('Quieres un ejemplo (s/n)');
    dformatmode;formatmode(3,55,1,'N');eformatmode;
    cursor(3,55);lector(linea,1,ncar,band);
    dformatmode;
    if linea[1]<>'n' then
    begin

```



```

        if c='a' then pilaej;
        if c='c' then colaej;
        if c='d' then ccolaej;
        if c='e' then lligada;
        asignaft;
    end;
    cursor(3,30);enha('@');write(' ');
end;

```

```

procedure maneja;
{asigna las tareas a los modulos}
var c:char;
begin
    lecbuff1(c);
    if (c='e') or (c='E') then ejemplo;
    if (c='p') or (c='P') then prueba(c);
    while c<>chr(10) do lecbuff1(c);
    limpant(5);
end;

```

```

procedure eschoja;
{impresion de una hoja o pagina }
var c:char;
begin
    n:=pila[topel];
    if (topel+1)>np then nn:=nc1+1
        else nn:=pila[topel+1];
    repeat
        lecbuff1(c);
        if c='N' then maneja
            else write(c);
    until n=nn;
end;

```

```

procedure inscripcion(var nosigo:boolean);
{inscripcion al sistema}
begin
    nosigo:=true;
    limpant(5);cursor(5,0);
    enha('M');writeln('NUMERO de cuenta ');
    dformatmode;formatmode(5,20,8,'N');aformatmode;

    cursor(5,20);lector(lines,longllave,ncar,band);
    dformatmode;
    if band=1 then
        begin
            nosigo:=false;
            return;
        end;
    limpbuf(lines,ncart+1,longllave);
    busca(lines,longllave,ultimo,posicion,apuntador);
    if apuntador = 0 then
        begin

```

```

{asigna funciones a las teclas}
  desabil;
  abilita;
{datos a capturar}
  cursor(10,0);
  enha('H');writeln('fecha de hoy (ddmmaa)');
  enha('H');writeln('tu nombre');
  enha('H');writeln('numero de semestres estudiados');
  enha('H');writeln('No. de cursos que llevas este semestre');
  enha('H');writeln('ya tomaste el curso tradicional?');
  enha('H');writeln('es para extraordinario?');
  enha('H');write('ya tienes los antecedentes?');

                                {protege la pantalla}
  dformatmode;
  formatmode(10,40,6,'N');
  formatmode(11,40,30,'N');formatmode(12,40,2,'N');
  formatmode(13,40,2,'N');formatmode(14,40,2,'N');
  formatmode(15,40,2,'N');formatmode(16,40,2,'N');
  eformatmode;

  copystr(lines,llave);                                {lectura de los datos de la materia}
  empaca(datos,linea,1,8);
  cursor(10,40);lector(lines,6,ncar,band);
  limpbuf(lines,ncar+1,6);empaca(datos,linea,9,6);
  cursor(11,40);lector(lines,30,ncar,band);
  limpbuf(linea,ncar+1,30);empaca(datos,linea,15,30);
  cursor(12,40);lector(lines,2,ncar,band);
  limpbuf(linea,ncar+1,2);empaca(datos,linea,45,2);
  cursor(13,40);lector(linea,2,ncar,band);
  limpbuf(lines,ncar+1,2);empaca(datos,linea,47,2);
  cursor(14,40);lector(linea,2,ncar,band);
  limpbuf(linea,ncar+1,2);empaca(datos,linea,49,2);
  cursor(15,40);lector(linea,2,ncar,band);
  limpbuf(linea,ncar+1,2);empaca(datos,linea,51,2);
  cursor(16,40);lector(linea,2,ncar,band);
  limpbuf(lines,ncar+1,2);empaca(datos,linea,53,2);
  dformatmode;                                {desabilita formato de protecci-
                                {la agrega al archivo de cubetas}
  agregacub(ultimo,llave,longllave,apuntador);

                                {agrega los datos al archivo de datos}
  {este archivo no necesita de registros ligados por eso posicion=
  posicion:=0;
  denumc(datos,55,3,58);
  denumc(datos,252,4,posicion);
  escdat(apuntador,datos);
  end
  else leedat(apuntador,datos);

  cursor(1,60);enha('H');
  ncar:=0;

```

```

while (ncar<15) and (datos[15+ncar]<>' ') do
begin
    write(datos[ncar+15]);
    ncar:=ncar+1;
end;

end;

begin
    {inicia parte principal}
    {asigna archivo de alumnos}

desabil;sbilita;

assign(a,'aldatos.eac');
a.mode:=direct;
rewrite(a);
a.trap:=true;

assign(b,'alindice.esc');
b.mode:=direct;
rewrite(b);
b.trap:=true;

pantalla;          {lineas pantalla}
pantinst;         {pantalla instructor}

inscripcion(nosigo); {inscripcion del alumno}
if nosigo then    {si no da no. de cta termina}
begin
    entrada;      {abre archivos de la unidad}
    if band=0 then {si no da No de unidad termina}
    begin
        adm1;          {registra unidad en aldatos}
        limpart(5);
        cursor(10,10);
        enha('B');
        write('Un momento se esta cargando la unidad');

        leebuff1;      {lectura de unidad}
        np:=tope;      {no. de hojas}
        leebuff2;      {lectura de pregunta}
        np1:=tope1;    {no. de preguntas}
        nnp:=np;

        limpart(5);
        assignaft;     {asigna funciones}
        tope:=0;

        repeat
            readln(comando);          {lee funcion}

            if comando[1]='n' then    {cusdro adelant
begin

```

```

tope:=tope+1;
if tope>nnp then tope:=nnp;
cursor(3,3);write(tope:2);limpant(5);
eschoja;
end;

if comando[1]='a' then {cuadro atras}
begin
tope:=tope-1;
if tope<1 then tope:=1;
cursor(3,3);write(tope:2);limpant(5);
eschoja;
end;

if comando[1]='p' then {primero}
begin
tope:=1;
cursor(3,3);write(tope:2);limpant(5);
eschoja;
end;

if comando[1]='u' then {ultimo}
begin
tope:=nnp;
cursor(3,3);write(tope:2);limpant(5);
eschoja;
end;

if comando[1]='c' then {cualquiera}
begin
cursor(3,3);write('HoJ');
dformatmode;formatmode(3,7,2,'J');eforma;
cursor(3,7);lector(linea,2,ncar,band);
dformatmode;
cursor(3,3);enha('@');write(' ');
if band=1 then
begin
cursor(3,3);write(tope:2);limpar;
eschoja;
end
else
begin
if ncar=1 then decharn(linea,1,1);
else decharn(linea,1,2,ncar);
tope:=ncar;
if ncar>nnp then tope:=nnp;
if ncar<1 then tope:=1;
cursor(3,3);write(tope:2);limpar;
eschoja;
end;
end;
end;

until comando[1]='s'; {termina}

adm4; {registra fin de sesion}

```

```
                end;  
            end;  
            close(a);close(b);close(y);close(x);  
end.
```

PRESENTACION

La facultad de Ingeniería de la UNAM se ha distinguido siempre por el gran interés de sus autoridades y profesores para elaborar material didáctico de apoyo a la docencia. En efecto, desde la fundación del Real Seminario de Minería, en 1772, primer colegio antecesor de la Facultad algunos catedráticos escribieron apuntes que complementaron o sustituyeron a los libros europeos que en esos tiempos estaban en uso, o que describieran los procedimientos, técnicas, inventos y descubrimientos que en su seno se iban desarrollando. Es muy satisfactorio que esta y otros libros de la Facultad puedan contribuir al desarrollo social, científico y tecnológico de nuestro país.

Dr. Octavio Agustín Rascon Chavez.

~

PROLOGO DE LOS AUTORES

Recientemente las ciencias de la computación han cobrado gran importancia debido a los avances científicos y tecnológicos cuyos resultados han permitido su utilización en áreas tales como: la industria, la educación, la salud, la economía, y otras actividades cotidianas del ser humano. En esta obra se pretenden cubrir los temas de lo que se conoce como Estructuras de Datos, ya que los contenidos académicos de esta materia son de fundamental importancia para el estudio y aplicación de las computadoras. ~

Expresamos nuestro reconocimiento a la Lic. Irma Inojosa Felix por su valiosa colaboración en la adaptación pedagógica de esta obra y a los alumnos Vicente Cordova R. y Jose Luis Rodriguez que contribuyeron en la recopilación de la información.

Jorge I Euan Avila
Luis G Cordero Barboza,

MEXICO 1984

~

GUIA DEL ALUMNO

Para un mejor uso de este material se deben seguir las siguientes instrucciones. Cada unidad se ha elaborado siguiendo criterios didácticos que se consideran útiles para mejorar la enseñanza y el aprendizaje. Al inicio de la unidad se establecen los objetivos de aprendizaje con el objeto de que al finalizar el estudio se conozca el comportamiento logrado. También se incorpora un índice que junto con los objetivos permitirán obtener una visión general de la estructura y organización del contenido.

R

En el contenido se encuentran ejemplos que se han elaborado haciendo uso de las capacidades gráficas y que se ofrecen en modo de simulación. Estos ejemplos requieren de participación.

También a lo largo del contenido se encuentran ejercicios y pruebas, las cuales deben de contestarse. Los resultados del desempeño se exhibirán y se darán en el momento de dar la respuesta. Si la respuesta no es correcta sona la campana. En algunos casos al finalizar la aplicación de un ejercicio o prueba se darán instrucciones de como continuar con el estudio.

R

Al final de la unidad se encuentra una prueba de autoevaluación que hay que contestar. Al terminarla se conocen los resultados del desempeño. Se agrega también al final una bibliografía en donde ampliar los conocimientos sobre lo expuesto.

 Nota: NO olvide que si se llena el espacio de respuesta NO se requiere oprimir la tecla RETURN.

R

** Índice General **

El material se ha organizado en 8 (ocho) UNIDADES

UNIDAD 0 Prefacio

-

- Presentación
- Prólogo
- Instrucciones para el alumno
- Índice General

R

UNIDAD 1 Elementos para el estudio de las Estructuras de Datos

-

- Generalidades
- Memoria Primaria
- Memoria Secundaria

R

UNIDAD 2 Estructuras de Datos Elementales

-

- Enteros
- Fisles
- Caracteres
- Arreglos

R

UNIDAD 3 Estructuras Compuestas: Listas Lineales

-

- Pila
- Cola
- Cola Doble
- Lista Circular
- Lista Doblemente Ligada

UNIDAD 4 Estructuras Compuestas: Listas no Lineales

- Gráficas
- Arboles
- Arboles Binarios

UNIDAD 5 Archivos

- Estructura
- Acceso
- El sistema de archivos

UNIDAD 6 Ordenamientos

- Notación Algorítmica
- Análisis de Algoritmos
- Ordenamientos Internos
 - Método de Selección
 - Método de Intercambio
 - Método de Inserción
 - Método de Distribución
 - Método de Intercalación
- Ordenamientos Externos

UNIDAD 7 Búsquedas

- Por comparación de llaves
- Por transformación de llaves

UNIDAD 8 Examen de Autoevaluación

***** Fin de la Unidad0 *****

UNIDAD1 Elementos para el estudio de las Estructuras de Datos.

Objetivo General

El alumno comprenderá los aspectos básicos de la estructura de una computadora digital, que le permitirán obtener un marco de referencia para iniciar el estudio de las Estructuras de Datos.

N

Objetivos Específicos

Al finalizar el estudio de esta unidad, el alumno:

1. Reconocerá que es el hardware y las unidades funcionales en que se divide.
2. Reconocerá que es el software.
3. Identificará los elementos asociados a la organización física y lógica de la memoria primaria.
4. Identificará los elementos asociados a la organización física y lógica de la memoria secundaria.
5. Diferenciará la memoria primaria de la memoria secundaria.
6. Resolverá problemas de capacidad de almacenamiento de acuerdo con los conceptos estudiados.

N

Introducción

Para el estudio de las estructuras de datos se requiere del conocimiento de algunos aspectos relevantes de la organización y operación de la computadora que nos dan un marco de referencia.

Se estudiarán los componentes generales de una computadora, de ahí se parte hacia aquellos que son relevantes para el diseño y operación de las estructuras de datos, como son por ejemplo los medios de almacenamiento(memorias).

Estos se han dividido para su estudio, en medios de almacenamiento primario y en medios de almacenamiento secundario.

N

Generalidades.- Para el estudio de la programación es necesario conocer los elementos físicos que integran una computadora y sus características de operación, asociadas al desarrollo de programas. Los dos componentes fundamentales son el HARDWARE y el SOFTWARE. El hardware esta constituido por las partes electrónicas y mecánicas y el software por instrucciones que permiten indicar y secuenciar las operaciones sobre los datos.

N

Componentes físicos.

Para su estudio podemos dividirlos en tres unidades funcionales: los dispositivos de Entrada y Salida, la Memoria y la Unidad Central de Proceso.

Los dispositivos de E/S permiten a la computadora comunicación con el exterior. Es a través de ellos que los usuarios

proporcionan información a la computadora o reciben información de ellos.

La memoria es la unidad de almacenamiento de instrucciones y datos. Podemos distinguir dos tipos de dispositivos los primarios y los secundarios. Los primarios operan a velocidades electrónicas y en ellos se encuentran instrucciones y datos próximos a ser ejecutados. Los secundarios operan a velocidades electromecánicas y en ellos se depositan datos e instrucciones por períodos largos de tiempo.

La unidad central es el dispositivo que ejecuta las instrucciones sobre los datos. La mayoría de las operaciones se realizan en la unidad aritmética y lógica, y el secuenciamiento para la ejecución de las operaciones, es coordinada por la unidad de control.

RFFVS;100

Programas de computadora.

Se llama programa a las instrucciones y datos que proporciona el usuario a una computadora para realizar una tarea específica. Algunos programas proporcionados por el fabricante o por alguna firma de software de uso común son:

- Sistema Operativo
- Compiladores
- Intérpretes
- Ensambladores y Macroensambladores
- Cargadores y Ligadores
- Editores
- etc

Algunos programas que escriben los usuarios para sus aplicaciones son:

- Contabilidad
- Solución de un sistema de ecuaciones
- Control de velocidad
- Modelado de sistemas
- etc

¶

Memoria primaria

La memoria es el almacén de instrucciones y datos, esta es la razón por la que nos interesa conocerla mejor.

Organización física.- La UCP toma instrucciones y datos de la memoria primaria en grupos de 'n' bits llamados palabras de computadores. Un bit es el nombre que recibe un dígito binario, el cual solo puede tomar dos posibles valores el uno o el cero.

La longitud de una palabra de computadora es el número de bits que puede almacenar y en la mayoría de las máquinas esta longitud oscila entre 8 y 64. Una palabra de computadora puede representar a 2^{**n} objetos distintos.

Otra medida es el byte el cual es una agrupación de 8 bits.

¶

La memoria la constituye un conjunto con 'm' palabras de longitud 'n', en donde a cada palabra de computadora se le asocia una dirección única entre 0 y m-1.

Direcc

	n-1	..4	3	2	1	0	Bit
0	:	*	*	*	*	*	*
1	:	*	*	*	*	*	*
2	:	*	*	*	*	*	*
3	:	*	*	*	*	*	*
4	:	*	*	*	*	*	*
.	:	*	*	*	*	*	*
.	:	*	*	*	*	*	*
.	:	*	*	*	*	*	*
m-1	:	*	*	*	*	*	*

¶

Para escribir o leer una palabra de computadora desde la UCP, existen asociados a la memoria: el registro de dirección, el registro de datos y líneas de control, mismos que permiten realizar estas dos operaciones. El registro de dirección indica que localidad de memoria será afectada, el registro de datos contiene el dato que será depositado en la localidad direccionada antes de una operación de escritura y, después de una operación de lectura contiene el dato de la localidad leída. Las líneas de control indica el inicio, la operación y el término de ella.

¶

Organización Lógica.- Para representar datos o instrucciones en la memoria se requiere de un cierto número 'b' de bits dependiendo de la información de que se trate. El número 'b' de bits podrá ser menor, igual o mayor que una palabra de 'n' bits. Consideremos dos casos para ilustrar lo anterior.

¶

Caso (a)

b < n	!----- n -----!
	* * * * *
	!----- b -----!

Si $b < n$ se puede pensar en empaquetar la información.

¶

Caso (b)

b > n	!----- n -----!
	* * * * *
	!----- b -----!

En este caso hay que usar más de una palabra de computadora. ¿Cuáles usar? Consideremos los siguientes casos:

- b.1 Palabras contiguas. Las que sus direcciones son a, a+1, a+2, ..
- b.2 Cualesquiera localidades. La utilización de este esquema requiere de algún mecanismo que establezca cuales son estas localidades y el orden de utilización. Una posible solución es destinar algunos bits de la palabra para un apuntador a otra palabra de tal manera que las encadene. Este campo en la palabra es llamado 'liga'.

b.3 Una combinación de las dos anteriores.

Memoria secundaria

En la memoria primaria solo están presentes los programas en ejecución con sus correspondientes datos. Los que no están siendo utilizados deben estar en algún otro medio de almacenamiento. Si comparamos la cantidad de información normalmente en uso con la que no lo está, nos daremos cuenta de que esta última es mucho mayor. Extender la memoria primaria para almacenar esta información resulta inconveniente por lo siguiente:

El aumentar la memoria más allá de la capacidad de direccionamiento de la máquina no tendría ninguna utilidad ya que se requiere de un esquema especial de direccionamiento, se desperdicia el recurso, ya que no siempre se utiliza toda la memoria, no se tendría la facilidad de remover y sustituir por otro.

Estos factores han influido para que se utilicen medios de almacenamiento que guarden la información a un costo razonable. Estos medios de almacenamiento necesitan de unidades especiales para ser usados. El costo de estos equipos pueda variar mucho dependiendo del que se trate. Estos dispositivos normalmente operan a velocidades mucho menores que la de la memoria primaria

Organización Física.- A continuación se estudian dos de los medios de almacenamiento secundario más utilizados: la cinta y el disco magnético.

La 'cinta magnético' es una tira de material plástico recubierta de una película de material sensible a los campos magnéticos. Para indicar la longitud utilizable de una cinta se colocan dos marcas reflectantes. Comercialmente se encuentran cintas de 300,600 y 2400 pies de longitud. Los cassettes de audio y cartuchos son también utilizables en equipos de cómputo pequeño.

La unidad de lectura y escritura es un dispositivo que guarda cuatro características importantes:

- Densidad de grabación. Se refiere a la cantidad de bits que pueden ser grabados por unidad de longitud. Densidades típicas son las de 300 a 6250 BPI.
- Número de pistas. Se refiere al número de bits que pueden grabarse a lo ancho de la cinta. Normalmente son de 7 y 9 pistas.
- Velocidad de arrastre. Es la velocidad con la que la cinta pasa por la cabeza de lectura y escritura. Se mide en pulgadas sobre segundo.
- Velocidad de transferencia de datos.- Es la velocidad con la que pueden leerse o escribirse datos en la cinta. Se da en bytes por segundo.

Debemos considerar que, para leer o escribir en la cinta, esta debe alcanzar cierta velocidad nominal, lo que requiere de un

cierto tiempo. De la misma forma, cuando dejamos de leer o escribir en la cinta, se requiere de un cierto tiempo para que esta se detenga. Debido a ello, entre grupos de caracteres existe un espacio que no puede ser utilizado llamado 'inter record gap' (IRG). Al grupo de caracteres entre dos IRGs se le llama registro físico. Los registros pueden ser de longitud fija o variable y en una operación de lectura o escritura todo el registro es leído o escrito.

~

Cuando se alcanza la velocidad nominal mantenerla constante es difícil, esta es la razón por la cual, una vez que se ha grabado un registro o registros, estos no pueden ser modificados y reescritos sobre sí mismos, y es necesario volver a escribir toda la cinta en caso de modificar alguno de ellos o escribirlo al final de la cinta.

En caso de lectura de la cinta, no existe problema para leer las veces que se desee ya que solo se está sensando la información.

~

 Deberá considerarse que debido a las características ya mencionadas para leer o escribir el registro 'n' deberán haberse leído o escrito los 'n-1' registros anteriores.

~

El 'disco magnético' es un plato o disco en cuyas caras se deposita una capa de material sensible a los campos magnéticos. Actualmente se tienen discos flexibles o duros. Los duros también se encuentran apilados llamados paquete de discos (disk pack). La unidad de lectura y escritura tiene las siguientes características principales:

- Densidad de grabación.- Tiene el mismo significado que en la cinta y se mide en Bits/pulgada. En los discos flexibles llamados 'Diskettes' se habla de densidad simple o de doble densidad.
- Pistas.- La pista es el área de grabación y en este caso son círculos concéntricos.
- Sectores.- Cada pista está subdividida en segmentos de arco llamados sectores o bloques y en una operación de lectura o escritura el sector es totalmente escrito o leído.

~

 En el disco cada bloque puede ser leído o escrito sin tener que leer o escribir los anteriores, esto hace que se le considere como un dispositivo de acceso directo.

El manejo de la información en un disco resulta considerablemente más lento que en la memoria primaria ya que para leer o escribir se requiere que la cabeza de lectura o escritura tome su posición sobre la pista, después debe esperar que pase al sector deseado y

finalmente, se debe considerar el tiempo requerido para mover la información.

2

Finalmente, existen discos que pueden removerse de la unidad y otros que estan fijos, así como unidades en las que las cabezas no se mueven.

2

Organización lógicas.

La Cinta magnética.- El usuario de un sistema de cómputo espera que el sistema le ofrezca flexibilidad para manipular la información. En particular puede suceder que se agrupen conjuntos de palabras con significado especial para el usuario a las que llamaremos registros lógicos. Entonces pueden presentarse dos casos:

Caso (a) Sea 'p' el número de bytes que requieren los entes lógicos y 'l' la longitud en bytes de un bloque.

Si $p \leq l$ pueden almacenarse uno o más registros lógicos en uno físico. Al número de lógicos en uno físico se le llama factor de bloqueo y determina el número de registros lógicos que pueden ser transferidos en una operación de lectura o escritura.

2

Caso (b) Si $p > l$ sera necesario utilizar más de un registro físico y en la cinta solo podremos utilizarlos de forma contigua.

2

El disco magnético.- Cuando utilizamos un disco para guardar información pueden ocurrir los siguientes dos casos:

Caso (a) Sea 'p' el número de bytes que requieren los entes lógicos y 'l' la longitud en bytes de un bloque.

Si $p \leq l$ pueden almacenarse uno o más registros lógicos en uno físico. Al número de lógicos en uno físico se le llama factor de bloqueo y determina el número de registros lógicos que pueden ser transferidos en una operación de lectura o escritura.

2

Caso (b) Si $p > l$ sera necesario utilizar más de un registro físico. Las alternativas que consideraremos, son: el uso de bloques contiguos o el uso de sectores ligados. En el caso ligado se pueden utilizar algunos bits del bloque como el apuntador o se puede construir una tabla con los apuntadores.

2 2 PFC1, FV1, FV3, JG1, CO2; 80

2

***** Fin de la UNIDAD1 *****

Preguntas de la unidad 1

NRC1

1 Hardware	?2	Lo forman las instrucciones y datos
2 Software	?3	En ella se realizan las operaciones aritméticas y lógicas.
3 CPU	?1	CPU, memoria y entrada/salida
4 Entrada/salida	?2	Paquetes y rutinas de utilería
5 Palabra de Computadora	?7	Almacena gran cantidad de instrucciones y datos por largo tiempo.
6 Memoria primaria	?4	Permite la comunicación de la computadora con el exterior.
7 Memoria secundaria	?5	Conjunto de bits entre 8 y 64.
	?6	De ella toma la CPU datos e instrucciones próximos a ejecutarse.
	?7	Discos, cintas, diskettes.

NFV1

Si 'n' es el número de bits de una palabra se pueden representar 2^{*n} objetos distintos.

Falso o Verdadero (FoV) ?V

NFV2

Con el registro de dirección se selecciona el bit 'n' de la palabra 'm'.

Falso o Verdadero (FoV) ?F

NFV3

La dirección para un acceso a disco esta definida si se especifica la superficie, la pista y el sector.

Falso o Verdadero (FoV) ?V

NFV4

Para una operación de escritura a la memoria primaria el registro de datos almacena la dirección de la palabra.

Falso o Verdadero (FoV) ?F

NFV5

Para su estudio a la computadora la hemos dividido en tres unidades funcionales que se llaman: Entrada/salida, UCP y la memoria.

Esto es falso o verdadero (FoV) ?V

NCO1

Los programas y datos que no estan en uso se encuentran en la memoria ??????????????SECUNDARIA

NJQ1

Por su capacidad de almacenamiento jerarquice los siguientes conceptos:

- ?3 Palabra de computadora
- ?1 Bit
- ?2 Byte
- ?4 Bloque
- ?6 Disco

25 Pista

8002

Cuántos caracteres pueden almacenarse en 25.4mm de cinta a 1600 BPI.

Respuesta ??????1600

8Si la velocidad de arrastre es de 37.5 m/s y la velocidad de transferencia de datos es de 243 750 bits/s

Cuál es la densidad de grabación en bits/m ?????6500

UNIDAD2 Estructuras de datos elementales.

Objetivo General

El alumno conocerá las formas de representar, almacenar y manipular estructuras de datos elementales (Enteros, reales, caracteres, arreglos).

¶

Objetivos Específicos

Al finalizar el estudio de esta unidad el alumno:

1. Reconocerá una estructura de datos elemental.
2. Reconocerá las formas de representar en la computadora enteros, reales y caracteres.
3. Reconocerá que es un arreglo y los modos de almacenarlo en la memoria.
4. Derivará funciones de mapeo y vectores de acceso para calcular la dirección de los elementos de un arreglo almacenado en forma contigua.
5. Escribirá algoritmos para calcular la dirección de los elementos de un arreglo almacenado en forma ligada.

¶

Generalidades

Una estructura de datos en su forma más general consiste de una colección de nodos o registros que mantienen importantes relaciones entre sí.

El nodo es el elemento básico para mantener la información en una estructura de datos. Para representar la información contenida en un nodo se pueden usar una o más palabras de computadora dependiendo de la información contenida en el nodo. Un nodo puede subdividirse en campos de tal manera que sea fácil la manipulación de la información.

```
-----
| campo1 | campo2 | campo3 | campo4 |
-----
```

¶

Es común encontrar que el contenido de un campo es un entero, o un real, o una cadena de caracteres.

¶

Representación de números enteros

Un número entero es cualquier número del conjunto definido como:

$$\{ \dots, -(n+1), -n, \dots, -2, -1, 0, 1, 2, \dots, n, (n+1), \dots \}$$

¶

Cualquier número entero sin signo puede representarse por una secuencia de dígitos de la forma:

$$D_n D_{n-1} \dots D_2 D_1 D_0$$

cuyo valor o magnitud es determinado por la suma de:

$$D_n B^n + D_{n-1} B^{n-1} + \dots + D_2 B^2 + D_1 B^1 + D_0 B^0$$

donde B es la base o raíz del sistema.

̄

Ejemplo. La magnitud del número 124 en base 10 es la suma de los términos:

$$1 \cdot 10^2 + 2 \cdot 10^1 + 4 \cdot 10^0$$

̄

Para representar un número entero en la computadora se utiliza el sistema de numeración binaria o de base 2 debido a la tecnología que se utiliza.

Ejemplo. La magnitud del número 10111 en base 2 es la suma de los términos:

$$1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$$

̄

Los números enteros de acuerdo a la definición pueden ser positivos o negativos, esto hace que para representarlos en la computadora se establecen convenciones como pueden ser el método de signo y magnitud o el del complemento.

Signo y magnitud.- Se usa un bit como el signo y los restantes para la magnitud entendida de acuerdo a la expresión anterior.

̄

Complemento.- El complemento a la base esta definido como:

$C = B^n - |\text{numero}|$ donde C es la representación del número en 'n' dígitos de la base B y $|\text{numero}|$ es la magnitud del número negativo.

si usamos 3 dígitos decimales, se pueden representar 1000 numeros sin signo entre 0 y 999. La idea consiste en asignar la mitad de estos numeros para los positivos de 0 a 499 y la otra para los negativos de -1 a -500 considerando el cero como positivo.

̄

Complemento a dos.- Por la utilización de una base binaria, el complemento a dos es la formas para especificar los numeros negativos.

Ejemplo. Representar el -5 en complemento a dos con 4 dígitos binarios.

$$\begin{aligned} 5 \text{ es } & 0101 \\ -5 \text{ es } & 10000 - 0101 = 1011 \end{aligned}$$

Una forma práctica de calcular el complemento es cambiar los unos por ceros y los ceros por unos y a este número sumarle uno.

̄
La ventaja del complemento es la facilidad para realizar operaciones aritméticas.

Ejemplo. Calcular $3+4$ y $3-4$

$$3+4 = 0011 + 0100 = 0111 = 7$$

$$3-4 = 0011 + 1100 = 1111 = -1$$

(1100 es el -4 en complemento a 2)

En este caso todos los bits forman el número, pero adicionalmente el bit más significativo es el signo. Cero es igual a positivo y uno es igual negativo.

̄
Representación de números reales

Un número real, en cualquier base (B), puede ser representado como:

$$A = + (A_n A_{n-1} A_{n-2} \dots A_1 A_0 . A_{-1} A_{-2} \dots A_{-(m+1)} A_{-m})$$

donde $A_n A_{n-1} A_{n-2} \dots A_1 A_0$ es la parte entera y
 $A_{-1} A_{-2} \dots A_{-(m+1)} A_{-m}$ es la parte fraccionaria

ambas partes se separan con un punto.

̄
Con el objeto de evitar problemas en la asignación del número de dígitos en ambas partes se normaliza el número y la representación es llamada científica o de punto flotante y queda:

$$+ . A_{-1} A_{-2} A_{-3} \dots A_{-m} \times B^E$$

donde $A_{-1} A_{-2} A_{-3} \dots A_{-m}$ es la parte fraccionaria
B es la base
E es un exponente positivo o negativo

̄
El formato es el siguiente:

```
-----
|s|  exponente  |  mantiza  |
-----
|  continuación mantiza  |
-----
```

<-----Palabra de computadora----->

El número de bits para el signo es uno y para el exponente y la mantiza depende de la magnitud y de la precisión deseada. Los números cuya magnitud exceden la capacidad del exponente no pueden ser representados y los números cuya precisión no puede

ser lograda tienen que truncarse o redondearse.

¶

Para aumentar magnitud o precisión o ambas hay que asignar más bits al formato y esto se logra incorporando más palabras de computadora. En los números de 'doble precisión' lo que se hace es aumentar en un poco más del doble la cantidad de bits de la mantiza.

Otros tipos de datos son los complejos y los lógicos. Los complejos se representan como dos reales uno para la parte real y otro para la imaginaria y los lógicos en un byte o en una palabra con todos los bits en uno o todos los bits en cero.

¶

Representación de caracteres.

Para facilitar la escritura y lectura de los datos a los usuarios se desarrollan códigos simbólicos para representar cada uno de los caracteres que forman el conjunto de símbolos.

Cada símbolo o carácter es representado por un patrón de bits diferente que lo identifica; el número de bits para representar un carácter está determinado por el número de caracteres.

Algunos de los códigos más importantes que actualmente se emplean son: el de la American Standard Code for Information Interchange (ASCII, el cual requiere de 8 bits, el Binary Code Decimal (BCD) que requiere de 6 bits y el Extended BCD Interchange Code (EBCDIC) de 8 bits.

¶

Representación de arreglos.

Una de las estructuras de datos y posiblemente la más conocida y utilizada es el arreglo.

Un arreglo es una estructura con un número fijo de nodos. Al conjunto de nodos se le identifica con un nombre y a los nodos con un índice.

Para su utilización los lenguajes de programación tienen instrucciones como DIMENSION, ARRAY, etc para definirlos.

En un arreglo se pueden practicar solo operaciones de lectura y escritura y no se puede agregar o eliminar un nodo del arreglo.

¶

Almacenamiento y recuperación.

Un arreglo puede representarse en la computadora utilizando localidades contiguas o ligadas o una combinación de ambas

Organización CONTIGUA.- Los nodos ocupan localidades contiguas de la memoria. El arreglo A(2,3) estaría almacenado de la siguiente forma:

$a(1,1)$	$a(1,2)$	$a(1,3)$	$a(2,1)$	$a(2,2)$	$a(2,3)$
1	2	3	4	5	6

̄

Cuando los nodos se almacenan en forma contigua la recuperación puede hacerse utilizando:

- (a) Funciones de mapeo
- (b) Tablas o vectores de acceso

̄

(a) Funciones de mapeo.

Una función de mapeo de relación uno a uno, es una expresión que permite calcular la dirección del nodo en términos de sus índices.

 Ejemplo: $A(10)$ almacenada a partir de la localidad 10.
 una expresión que nos permite calcular la dirección es

Dir $A(i) = 9 + i$
 en forma general Dir $A(i) = \text{Direccion_Base} + i$
 La dirección de $a(4) = 9 + 4 = 13$

̄

Para arreglos de más de una dimensión pueden derivarse expresiones polinomiales que permiten la recuperación. Para un arreglo $A(n,m)$. La dirección de cualquier $A(i,j)$ se calcularía:

Dirección_Base + $r(i-1) + j$; $r = \text{No. de columnas}$

̄

Para arreglos que por la disposición de sus nodos guardan ciertas características como pueden ser: los arreglos llamados 'simétricos' o los 'triangulares' es posible con la utilización de las funciones lograr un ahorro considerable de la memoria.

̄

(b) Tablas o vectores de acceso

Algunas funciones de mapeo, por las operaciones que involucran, ocasionan una sobrecarga durante el cálculo de la dirección, un método alterno es: en lugar de hacer el cálculo tener el dato almacenado en una tabla o vector.

 Ejemplo: Considere $A(2,3)$ almacenada por renglones como en el caso anterior un vector de acceso asociado sería:

$VA=9,12$ y la función de recuperación es :

$VA(i) + j$

̄

Algunos algoritmos para resolver sistemas de ecuaciones lineales

simultaneas, requieren intercambiar las columnas para prevenir pivotes pequeños que ocasionen errores de redondeo. El uso de vectores de acceso para intercambiarlos es eficiente ya que solo bastará intercambiar los elementos del vector.

En arreglos dentados el ahorro de memoria es considerable si usamos este concepto.

̄

Organización LIGADA.- En este caso los elementos del arreglo ocupan cualquier posición y su relación esta definida por un campo liga. En este caso para el almacenamiento y la recuperación se requiere de un algoritmo más elaborado.

̄

Una organización parcialmente ligada, llamada así porque contiene partes de los datos en forma contigua es la siguiente:

:XX: 6:	: 9:3 :	:19:1 :	:29: :
t 4	6	3	1
	10 a(1,1)	20 a(1,2)	30 a(1,3)
	11 a(2,1)	21 a(2,1)	31 a(2,3)

̄

Un algoritmo sería:

```
temp:= t ; t es el apuntador a lista ligada
desde m:=1 hasta j hacer
    temp:=liga_derecha (temp);
direccion:=liga_izquierda (temp) + i;
```

Se recorre la lista hasta alcanzar la columna y se calcula la posición del renglon.

̄

Una organización totalmente ligada sería en la que las columnas también formaran una lista ligada. Esta representación, son convenientes para arreglos esparcidos. Estos arreglos contienen pocos datos diferentes de cero y las posiciones que ocupan son cualquiera del arreglo. Con una organización ligada estamos ahorrando memoria a costa de escribir algoritmos un poco más elaborados.

̄

A continuación se encuentra el examen de autoevaluación de la UNIDAD.

RP

R

*** Fin de la UNIDAD2 ***

Preguntas de la Unidad 22

RDM1

Relacione la columna de la derecha con la de la izquierda. Escribiendo dentro del parentesis el número que corresponda.

- | | | |
|----------------------------------|----|---|
| 1 Complemento a 2 | ?4 | Sirven para recuperar datos de un arreglo almacenado contiguamente. |
| 2 Arreglo | ?1 | Se calcula como $C=(2*n)-\text{numero}$. |
| 3 Contiguo | ?2 | En esta estructura no se permite aumentar el número de nodos. |
| 4 Funciones y vectores de acceso | ?3 | Las localidades son direcciones de la memoria numeradas como 1,2,3,.. |

RFV1

El RETURN (retorno) no tiene codificación en el código ASCII.

Esto es falso o verdadero (FoV) ?F

RFV2

Algunos números reales no tienen representación exacta con el formato exponente y mantiza.

Esto es falso o verdadero (FoV) ?V

RCO1

Calcular la magnitud de los siguientes números en base 10.

1023	????1023	327	????215	111001	????57
10		8		2	

RCO2

Calcular la magnitud en base diez del número 1111 en complemento a dos.

respuesta ???1

R

UNIDAD III Estructuras de datos compuestas: listas lineales

R

INDICE UNIDAD III

Objetivos específicos
Introducción
Generalidades
Pila
Cola
Cola Doble
Lista Circular
Listas Doblemente Ligada
Consideraciones sobre el almacenamiento contiguo y ligado
Cuestionario de Evaluación

R

OBJETIVOS UNIDAD III

- 1.- Identificar los conceptos de lista lineal, cola, cola doble, lista circular y lista doblemente ligada.
- 2.- Realizar las operaciones que se practican sobre una pila, cola y cola doble.
- 3.- Escribir algoritmos para las operaciones sobre una pila, cola y cola de doble entrada almacenadas en forma contigua y en forma ligada.
- 4.- Escribir los algoritmos de las operaciones que se practican en una cola circular almacenada en forma contigua y también cuando se almacena en forma ligada.
- 5.- Escribir los algoritmos de las operaciones de agregar y retirar en una lista doblemente ligada y también cuando es circular doblemente ligada.
- 6.- Resolver problemas en los que se involucre la utilización de las listas lineales.

R

INTRODUCCION

Un arreglo, tal como se ha visto anteriormente, esta constituido por un conjunto fijo de nodos y, excepcionalmente se permite aumentar o disminuir el número de nodos.

Las listas, nombre que recibe un conjunto de nodos que puede ser aumentado o disminuido, viene a darle mayor flexibilidad de programación a los usuarios, con un considerable ahorro de memoria por las operaciones que pueden practicarse sobre ellas.

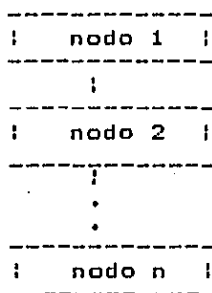
La unidad comprende el estudio de un tipo de lista: las listas lineales cuya representación y utilización serán ejemplificadas con la pila, cola y cola de doble entrada. Las listas no lineales se dejan para su estudio en la siguiente unidad.

¶

III.1 GENERALIDADES

Una lista es una estructura de datos que tiene un número variable de nodos. Una lista lineal es una lista cuyos nodos están ordenados por un solo criterio, en donde el último y el primer nodo no tiene sucesor y antecesor respectivamente. Una lista lineal es un conjunto de $X(1)$, $X(2)$, ..., $X(n)$, cuyas propiedades estructurales involucran relaciones en una sola dimensión entre sus nodos. Por ejemplo, $X(1)$ es el primer nodo, $X(n)$ es el último y el nodo $X(k)$ es precedido por $X(k-1)$ y le sigue el $X(k+1)$. Vea la siguiente figura.

¶



Representación gráfica de una lista lineal.

¶POM1

III.2 PILA

Definiciones y operaciones.

Una pila o stack es una lista lineal, en la cual las operaciones de agregar y retirar un nodo se realizan por uno de los extremos.

La implementación ferroviaria llamada i griega es un ejemplo de como se maneja una pila.

Por la forma como se agregan y retiran los elementos en la pila, el método ha sido llamado UEPS últimas entradas primeras salidas. Esto significa que solamente puede ser retirado de la pila el último elemento agregado. En esta estructura las operaciones de agregar y retirar reciben el nombre de PUSH y POP respectivamente.

Representación y algoritmos de las operaciones.

Una pila puede ser representada en la computadora eligiendo localidades contiguas de memoria esto es:

¶

Max

TOPE

Min

```

  | | | | | | | |
  K  K+1 K+2 .....K+n

```

Max y Min
TOPE

delimitan el area de implementacion
apuntador al último elemento agregado

▮

Como delimitamos el area de implementación debe verificarse que no se excedan estos limites. Antes de agregar un nuevo elemento hay que verificar si hay lugar para el, si no existe, habrá un error de pila llena (stack overflow) así como verificar en la operación de retiro si existe un elemento por retirar, si no, habrá un error de pila vacia (stack underflow).

▮

ALGORITMOS DE LAS OPERACIONES

```
procedimiento AGREGAR(tope);
```

```

si tope=Max
  entonces { escribe('la pila llena') }
  obien { si tope=nulo
          entonces {tope=Min; [tope]=datos}
          obien {tope=tope-1; [tope]=dato} }

```

▮

```
procedimiento RETIRAR(tope);
```

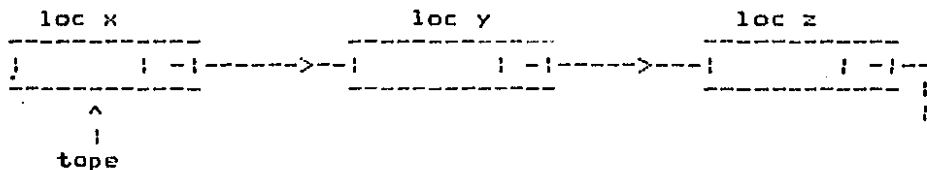
```

si tope=nulo
  entonces { escribe('la pila vacia')}
  obien {si tope=min
          entonces {datos=[tope];tope=nulo}
          obien {dato=[tope];tope=tope+1} }

```

▮

Una pila también puede ser almacenada en localidades ligadas de memoria. Una pila simplemente ligada puede representarse de la siguiente manera:



Las localidades para cada uno de los nodos de la pila son cualesquiera, pero el orden de la pila esta definido por TOPE y las ligas de la estructura. El apuntador TOPE se usa para indicar la posición del último elemento agregado.

▮

ALGORITMOS DE LAS OPERACIONES DE AGREGAR Y RETIRAR

Para agregar un nuevo elemento a una pila se requiere en forma general de:

- 1) Obtener la dirección de un nodo disponible
- 2) Agregarlo a la pila

OBTENER n

```
si tope=vacio
    entonces {liga(n)=vacio,tope=n}
    obien {liga(n)=tope,tope=n}
```

En caso de no obtener una localidad disponible, se dice que la pila esta llena y se ha alcanzado una condición o señal de OVERFLOW.

̄
Para retirar un elemento de la pila se requiere en forma general de:

- 1) Retirar el nodo apuntado por tope y
- 2) Retornarlo como disponible

Las localidades disponibles se controlan utilizando un fondo común del cual se sacan las localidades que se necesitan y al cual se regresan las localidades que ya no se usan.

```
si tope=vacio
    entonces {UNDERFLOW}
    obien {n=tope,tope=liga(tope),RETORNAR n}
```

̄
Ejemplos del uso de una pila.

a) Recursion.- La recursividad, es una formulación que permite definir un concepto en términos propios. Para conservar las direcciones de retorno se usa una pila.

b) Manejo de interrupciones.- Para regresar el control despues de una interrupción se requieren los valores del estado del proceso anterior.

c) Evaluación de expresiones.- En la notación convencional se define una jerarquia para los operandos por ejemplo $A+B*C$ se evalúa primero $B*C$ y luego se le suma A . En la notación polaca (prefija o posfija), los operadores poseen la misma jerarquia y el orden de evaluación se da por la posición que operadores y operandos guardan en la expresión. Por ejemplo en notación prefija $A/C+B/D$ es $+/AC/BD$. Su evaluación requiere del auxilio de una pila.

̄
III.3 COLA

Definición y operaciones

Una cola es una estructura de datos lineal en la cual las operaciones se realizan en ambos extremos. Se agrega por uno de

los extremos y se retira por el otro. Son ejemplos de cola, una línea de espera para pagar o cobrar en un banco, para comprar gasolina, para pasar un semáforo, etc. Un modelo para una cola es el de la figura siguiente.

```

<- DESPACHADOR <-  -----  -----  -----  -----
                   |proc 1|  |proc 2|  |proc 3|  |proc 4|
                   -----  -----  -----  -----
                   PRIMERO                                ULTIMO
                   ..... COLA .....

```

Para ilustrar la forma de operación de una cola consideremos que un nuevo elemento desea agregarse a la cola, este solamente podría hacerlo después del último y el elemento que puede retirarse de la cola es aquel que se encuentra al frente o sea el primero de la cola.

Por la forma como agregan y retiran los elementos en una cola el método ha sido llamado FEFS (primeras entradas primeras salidas) en inglés FIFO (first input first output). En algunas aplicaciones también el método es conocido como FAPS (primeros en arribar primeros en ser servidos) en inglés FCFS (first come first served).

Con esta política se entiende que solamente puede ser retirado de la cola el primer elemento agregado, pero sabemos que en algunas aplicaciones algunos elementos abandonan la cola sin ser servidos y que otros, que no están al frente de la cola pueden ser servidos, creandose en este caso una cola con prioridades. La prioridad se logra por la finalidad, por algún parentesco, por una propina, etc.

Representación y algoritmos de las operaciones

Una cola puede ser representada eligiendo localidades contiguas de la memoria de acuerdo a como se muestra en la siguiente figura.

```

MAX  U                                F  MIN
    | | | | | | | | | | | | | | | | |
    1 2 3 4 .....n

```

Para indicar el frente o el primero de la cola se requiere de un apuntador a ese elemento el cual llamaremos F y para indicar el último de la cola U.

De forma similar a la pila, la cola requiere que se delimite el área de implementación ya que no es posible dejarla crecer de manera infinita.

¶

Algoritmos de las operaciones

Agregar un elemento a la cola

```

Si U=MAX
  entonces {OVERFLOW}
  obien { Si U=vacio
            entonces {U=P=MIN, cola(U)=dato}
            obien {U=U-1, cola(U)=dato} }

```

Retirar un elemento de la cola

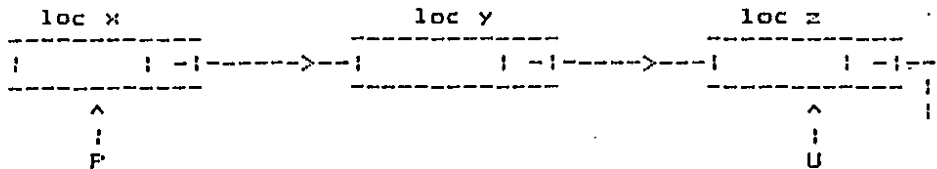
```

Si U=P=vacio
  entonces {UNDERFLOW}
  obien { Si P=U
            entonces {dato=cola(P), P=U=vacio}
            obien {dato=cola(P), P=P-1} }

```

REC

Una cola también puede ser representada usando localidades de la memoria. Vea la siguiente figura.



Para agregar un elemento se procede de la siguiente manera:

- 1.- Obtener un elemento disponible del fondo común
- 2.- Agregarlo a la cola

Algoritmo para agregar en una cola simple ligada. Como la de la figura anterior.

```

Si n se obtiene
  entonces { liga(n)=vacio,
            si U=P=vacio
              entonces {U=P=n}
              obien {liga(U)=n, U=n} }
  obien {overflow}

```

Algoritmo para retirar

```

Si U=P=vacio
  entonces {underflow}
  obien {n=P, P=liga(P), RETORNAR n
        si P=vacio entonces {U=vacio} }

```

REC

III.4 COLA DOBLE

Definición y operaciones

Una cola doble es una estructura en la cual las operaciones de agregar y retirar se practican en ambos extremos. Un elemento que desea agregarse podría hacerlo al final o al principio de la cola, los elementos al principio y al final pueden ser retirados. Por la forma como se agregan y retiran elementos, no existe un método para la cola doble, aunque es posible practicar los métodos PEPS y UEFS.

Algoritmos de las operaciones

En una cola doble la operación de agregar o de retirar se puede realizar por P o por U.

Los algoritmos para una cola doble almacenada en forma contigua o ligada se dejan al alumno como un ejercicio.

¶

III.5 LISTA CIRCULAR

Una lista circular es una estructura de datos que tiene como característica fundamental un orden en el que, al último nodo le sigue el primero.

Este tipo de lista viene a solucionar el problema de desplazamiento de los nodos sobre la memoria como ocurre en el caso de la cola, o el problema de recorrer los nodos una posición después de un borrado.

Los algoritmos ya vistos para una cola, ocasionan que la cola se desplace entre los límites de implementación. Una alternativa para evitar esto, es mantener P fijo y que, cuando se retire un elemento, se muevan los restantes una posición.

Mover los elementos una posición puede ser muy costoso si la cola es muy larga. En estos casos, la mejor alternativa es el uso de una lista circular.

Representación y algoritmos de las operaciones

Una lista circular puede ser representada en la computadora usando localidades contiguas de memoria. Con la característica de que al alcanzar MAX se continua con MIN si es que esta disponible. EL estado de OVERFLOW se alcanza cuando todas las localidades estan llenas.

Algoritmo para agregar en una cola circular contigua.

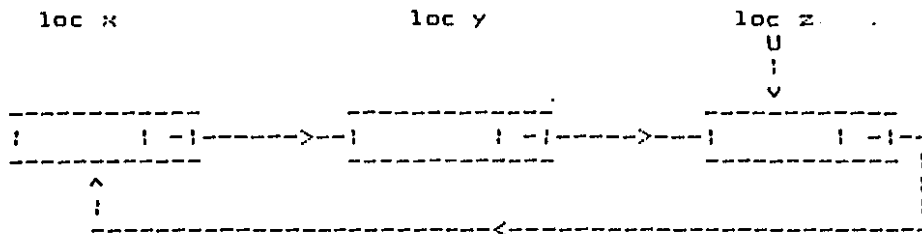
```

Si (P=U+1) o (U=MAX y P=MIN)
  entonces {OVERFLOW}
  obien {Si P=U=vacio
            entonces {U=P=MIN}
            obien {Si U=MAX
                    entonces {U=MIN}
                    obien {U=U+1} }

```

¶Ed

Una lista circular también puede almacenarse en localidades ligadas de memoria. Vea la figura siguiente:



En el manejo de una cola almacenada en forma circular los apuntadores P y U pueden reducirse a un solo apuntador U y considerar que P es la liga(U). De esta forma, el apuntador U será el único elemento que participa en las operaciones de agregar y retirar.

Algoritmos de las operaciones

Agregar en una cola ligada circular.

```

Si n se obtiene
  entonces {Si U=vacio
            entonces {liga(n)=n;U=n}
            obien {liga(n)=liga(U);liga(U)=n;U=n} }
  obien {OVERFLOW}

```

Retirar en una cola ligada circular.

```

Si U=vacio
  entonces {UNDERFLOW}
  obien {Si U=liga(U)
        entonces {n=U,U=vacio,se retorna n}
        obien {n=liga(U),liga(U)=liga(liga(U)),se retorna n}}

```

¶

III.6 LISTAS DOBLEMENTE LIGADAS

Definición y operaciones

En una lista doblemente ligada se han incluido dos campos liga; uno que señala al nodo sucesor llamado LIGA DERECHA (LD) y el otro que señala al nodo antecesor llamado LIGA IZQUIERDA (LI). Al

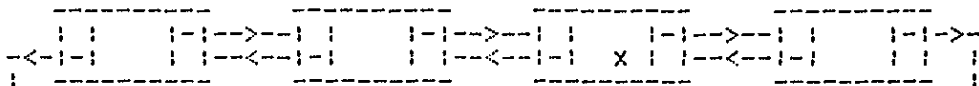
incluir estos dos campos, es posible conocer desde cualquier nodo cual es su nodo sucesor y cual su antecesor, cosa que no es fácil de lograr en una lista con una sola liga. Vea la figura siguiente.

Estructura del nodo.

```
-----
|LI|  INFO  |LD|
-----
```

▮

Lista doblemente ligada

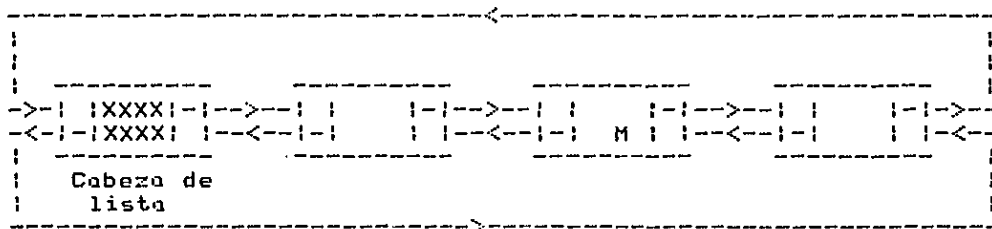


Vea a continuación como se agrega un nodo a la izquierda de X.

▮Ee

Otras operaciones serían las de agregar a la derecha de X, agregar cuando la lista esta vacía, etc. Una forma de simplificar los algoritmos para evitar el caso de lista vacía, es no permitir que la lista quede vacía, esto puede lograrse incorporando a la lista un nodo especial, llamado cabeza de lista el cual siempre permanece en ella. Para evitar los casos particulares en los extremos de la lista se sugiere hacerla circular. Vea la figura siguiente.

▮



De acuerdo a la figura podemos decir que para cualquier nodo M se cumple que:

$$LD(LI(M)) = LI(LD(M)) = M$$

▮

De esta forma el algoritmo de agregar a la derecha de cualquier nodo es el siguiente:

```
LI(n)=LI(?)
LI(n)=?
LI(LD(?))=n
LD(?)=n
```

RPOM1

Preguntas de la unidad 33

ROM1

Es una estructura en la que el número de elementos puede aumentar o disminuir.

- 1 Arreglo entero
- 2 Lista
- 3 Byte
- 4 Real de doble precisión
- 5 Dimensión 1(32)

Respuesta ??2

RFV1

En una pila o Stack por la forma como se agregan y retiran elementos el método es llamado PEPS.

Falso o Verdadero(FOV) ??F

RCO1

Una cola es una estructura de datos ??????????lineal

RRC1

??5 Es una lista en la que la estructura del nodo solo tiene un campo liga

??1 Nombre que recibe el método de agregar y retirar en una cola

??3 En esta estructura el último nodo tiene un apuntador al primero.

??4 Desde cualquier nodo es posible conocer al nodo anterior y al sucesor

??2 Los que estan formados en las tortillas

- | | |
|----------------------------|---------------------------|
| 1 PEPS o FCFS | 2 COLA |
| 3 LISTA CIRCULAR | 4 LISTA DOBLEMENTE LIGADA |
| 5 LISTA SIMPLEMENTE LIGADA | 6 PILA |

RJQ1

Considere la siguiente expresión $A*(B-C)+D/E$ transformela a notación polaca prefija y establezca el orden de evaluación

??2 b-c

??1 d/e

??3 a*(b-c)

??4 [a*(b-c)][+][d/e]

RTB1

Escriba los algoritmos de agregar y retirar en una pila doblemente ligada circular.

De return para continuar ??

1. Atkinson R
Futures: Where Will Computer-Assisted Instruction (CAI) Be in 1990.
Educational Technology, April 1978.
2. Chambers J and Sprecher J
Computer Assisted Instruction: Current Trends and Critical Issues.
California State University
Communications of the ACM, June 1980 Volume 23, Number 6
3. Cohen V
Criteria for the Evaluation of Microcomputer Courseware.
Educational Technology, January 1983.
4. Euan, J. y Cordero, L.
Apuntes de Estructuras de Datos
Libro de texto universitario
UNAM 1984.
5. Gane and Sarson
Structured System Analysis: tools and techniques
Prentice Hall, Englewood Cliffs, New Jersey. 1979.
6. Hazen M
Computer -Assisted Instruction Whit PILOT on the Apple Computer.
Educational Technology, November 1982.
7. Herriot J
CAI: A Philosophy of Educational and a System to Match
Creative Computing, April 1982.
8. Knott G
Hashing Functions
The Computer Journal, Volume 18, Number 3 .
9. Lechner H
The Computer Chronicles
Wadsworth Publishing Company/Continuing Education
Belmont California, 1985.
10. Magidson E
Issue Overview: Trends in Computer Assisted Instruction
Educational Technology, April 1978.
11. Memorias de la V conferencia de Computadoras en la Educación
Monterrey, N.L. 1985.

12. Mendez Martinez J
Paquetes Autodidácticos: Un modelo y un procedimiento de elaboración.
Tesis Facultad de Psicología de la UNAM. MEXICO
13. Mosmann C
Computer-Based Learning in Higher Education - The Missing Revolution.
Journal of Research and Development in Education
Volume 14 Number 1, 1980.
14. Morris J
Computer-Aided Instruction: Toward a New Direction.
Educational Technology, May 1983
15. Saad E y Pacheco D
Texto de Capacitación Docente
Instituto de Investigaciones Sociales y de la Educación
Mexico 1982.
16. Sistematización de la Enseñanza
CISE-UNAM
Presentación de la etapa de Objetivos
Friendman
17. Steffin S
A Suggested Model for Establishing the Validity of Computer Assisted Instructional Materials.
Educational Technology, January 1983.
18. Splittgerber F
Computer-Based Instruction: A revolution in the Making?
Educational Technology, January 1979.
19. Suppes and Macken
The Historical Path From Research and Development to Operational Use of CAI.
Educational Technology, April 1978.