

REPOSITORIO ACADÉMICO DIGITAL INSTITUCIONAL

Sistema generado de evaluaciones y encuestas

Autor: Fabiola García Rangel

**Tesis presentada para obtener el título de:
Ing. en Sistemas Computarizados [sic]**

**Nombre del asesor:
Héctor Guerrero Guadarrama**

Este documento está disponible para su consulta en el Repositorio Académico Digital Institucional de la Universidad Vasco de Quiroga, cuyo objetivo es integrar, organizar, almacenar, preservar y difundir en formato digital la producción intelectual resultante de la actividad académica, científica e investigadora de los diferentes campus de la universidad, para beneficio de la comunidad universitaria.

Esta iniciativa está a cargo del Centro de Información y Documentación "Dr. Silvio Zavala" que lleva adelante las tareas de gestión y coordinación para la concreción de los objetivos planteados.

Esta Tesis se publica bajo licencia Creative Commons de tipo "Reconocimiento-NoComercial-SinObraDerivada", se permite su consulta siempre y cuando se mantenga el reconocimiento de sus autores, no se haga uso comercial de las obras derivadas.





**UNIVERSIDAD
VASCO DE QUIROGA**

RECONOCIMIENTO DE VALIDEZ OFICIAL DE ESTUDIOS
POR ACUERDO 952006 DE LA S.E.P. CLAVE 16PSU0014Q

LICENCIATURA EN SISTEMAS COMPUTARIZADOS

**SISTEMA GENERADOR
DE EVALUACIONES Y ENCUESTAS**

TESIS

QUE PARA OBTENER EL TITULO DE
LICENCIADA EN SISTEMAS
COMPUTARIZADOS

PRESENTA
FABIOLA GARCIA RANGEL

ASESOR DE TESIS:
ING. HECTOR GUERRERO GUADARRAMA

MORELIA, MICH., MAYO DE 1997



AVALA T14



**UNIVERSIDAD
VASCO DE QUIROGA**

RECONOCIMIENTO DE VALIDEZ OFICIAL DE ESTUDIOS
POR ACUERDO 952006 DE LA S.E.P. CLAVE 16PSU0014Q

LICENCIATURA EN SISTEMAS COMPUTARIZADOS

**SISTEMA GENERADOR
DE EVALUACIONES Y ENCUESTAS**

TESIS

QUE PARA OBTENER EL TITULO DE
LICENCIADA EN SISTEMAS
COMPUTARIZADOS

PRESENTA
FABIOLA GARCIA RANGEL

ASESOR DE TESIS:
ING. HECTOR GUERRERO GUADARRAMA

MORELIA, MICH., MAYO DE 1997



BIBLIOTECA

CAMPUS SANTA MARIA

Gracias

PRESENTACION 1

A mis papás por su amor infinito

OBJETIVOS PLANTEADOS 4

A mis hermanos por dar alegría a mi vida.

HIPOTESIS 5

A Elias por su gran amor y paciencia.

AREAS DE INTERES 6

*Al Ing. Héctor Guerrero por su ayuda
desinteresada en la realización de este trabajo y
especialmente por su amistad.*

PLANEACION DEL SISTEMA 7

ANALISIS DEL PROBLEMA 10

INVESTIGACION DE INSTRUMENTOS DE EVALUACION 10

INSTRUMENTOS DE EVALUACION 12

LAS PRUEBAS OBJETIVAS Y LOS CUESTIONARIOS PRECODIFICADOS 13

CLASIFICACION DE LAS PRUEBAS 15

A mis maestros por sus enseñanzas.

ANALISIS DE LOS REACTIVOS DE EVALUACION 20

RECOMENDACIONES Y RESTRICCIONES PARA LA CONSTRUCCION DE REACTIVOS 25

*Y sobre todo gracias a Dios por la oportunidad
de ser.*

DE FUNDACION 25

DISEÑO DE LA SOLUCION 27

OBJETIVOS DEL DISEÑO 27

ESPECIFICACION DE LOS ELEMENTOS LOGICOS DEL DISEÑO 27

INDICE

ESTRUCTURA Y PROCESO DE EVALUACION DE LOS REACTIVOS	29
PRESENTACION	1
DISEÑO DE LAS BASES DE DATOS	35
OBJETIVOS PLANTEADOS	4
DICCIONARIO DE DATOS	38
HIPOTESIS	5
DISEÑO LOGICO	42
OBJETIVOS	42
AREAS DE APLICACION	6
DISEÑO DE SISTEMAS FACILES DE MANTENER	43
DISEÑO ESTRUCTURADO	44
PLANEACION DEL DESARROLLO DEL SISTEMA	7
ESPAÑOL ESTRUCTURADO	46
ANALISIS DEL PROBLEMA	10
INVESTIGACION DE REQUERIMIENTOS	10
INSTRUMENTOS DE EVALUACION	12
LAS PRUEBAS OBJETIVAS Y LOS CUESTIONARIOS PRECODIFICADOS	13
CLASIFICACION GENERAL DE LOS REACTIVOS	15
ANALISIS DE LOS REACTIVOS DE EVALUACION	20
RECOMENDACIONES Y RESTRICCIONES PARA LA CONSTRUCCION DE REACTIVOS DE EVALUACION	25
DISEÑO FISICO	65
HERRAMIENTAS PARA EL DESARROLLO DE SISTEMAS	88
DISEÑO DE LA SOLUCION	27
OBJETIVOS DEL DISEÑO	28
ESPECIFICACION DE LOS ELEMENTOS LOGICOS DEL DISEÑO	28
EL DISEÑO FISICO DEL SISTEMA	

INDICE

ESTRUCTURA Y PROCESO DE EVALUACION DE LOS REACTIVOS	29
GENERACION DE CALIFICACIONES	76
CONECTIVIDAD	81
DISEÑO DE LAS BASES DE DATOS	35
DICCIONARIO DE DATOS	36
DISEÑO LOGICO	42
OBJETIVOS.-	42
DISEÑO DE SISTEMAS CONFIABLES	42
DISEÑO DE SISTEMAS FACILES DE MANTENER	43
DISEÑO ESTRUCTURADO	44
HERRAMIENTAS DEL DISEÑO LOGICO	45
ESPAÑOL ESTRUCTURADO	46
DISEÑO DE LAS SALIDAS	48
MODULOS DEL SISTEMA	51
<i>Módulo de reactivos</i>	52
<i>Módulo de grupos</i>	55
<i>Módulo de formación del guión</i>	57
<i>Módulo del examen impreso</i>	61
<i>Módulo del examen interactivo</i>	63
DISEÑO FISICO	65
HERRAMIENTAS PARA EL DESARROLLO DE SISTEMAS	66
GENERADORES DE CODIGO	68
LA CORRECTA PRACTICA DE LA CODIFICACION	70
PROGRAMACION ORIENTADA A EVENTOS	72
EL DISEÑO FISICO DEL SISTEMA	73

GENERACION DEL EXAMEN INTERACTIVO	73
GENERACION DE CALIFICACIONES	76
CONECTIVIDAD.....	81
PRUEBA Y DEPURACION.....	86
IMPLANTACION Y EVALUACION.....	90
CONCLUSIONES Y RECOMENDACIONES.....	93
BIBLIOGRAFIA.....	95
MANUAL DEL USUARIO.....	98
REPORTES.....	113
CODIGO FUENTE.....	123

La problemática actualmente detectada es la siguiente:

PRESENTACION



Una actividad que ha acompañado permanentemente al proceso de enseñanza es la evaluación, de una u otra forma siempre se está interesado por conocer los logros obtenidos.

En el libro "Psicotecnia pedagógica" de la Dirección General de Educación Tecnológica Industrial (DGETI) el Ing. Juan Leonardo Sánchez Cuéllar comenta:

"Cada actividad exige una evaluación continua, toda tarea implica objetivos, metas a alcanzar que deben evaluarse con el objeto de fijar nuevas finalidades. Cuando se puede medir aquello que se habla o expresa en números, entonces se podrá asegurar que se posee algún conocimiento sobre él".

De ahí la importancia de la evaluación, no solamente en el ámbito educativo, sino en todas las áreas y más importante aún el uso adecuado de las herramientas para realizarla.

Es por eso que el "Sistema Generador Evaluaciones y Encuestas" tiene como finalidad ser una aplicación preparada para resolver algunas de las tareas más frecuentes en la elaboración de este tipo de mediciones, ya sea en forma de exámenes o bien de encuestas.

* consultar bibliografía

La problemática actualmente detectada es la siguiente:

1. Falta de herramientas para el diseño de cuestionarios. Las únicas utilizadas son la máquina de escribir y en el mejor de los casos un procesador de textos.
2. Inversión de tiempo en la modificación o actualización de un cuestionario ya elaborado.
3. Duplicación de la información almacenada, ya sea en hojas o en discos magnéticos.
4. Desorganización cuando diseñan cuestionarios de más de un tipo o categoría
5. Carencia de herramientas para la aplicación y evaluación de cuestionarios

Todos estas carencias percibidas me llevaron a darme cuenta de la necesidad de diseñar una herramienta para solucionar los problemas mencionados al elaborar un cuestionario ya que dicha actividad es primordial tanto en el área de evaluación como de recolección de información.

Lo anterior no significa que no existan ya algunos intentos de hacer algo para solucionar el problema, pero se ha observado que presentan algunos inconvenientes:

- Tienen una interfaz que no es muy accesible al usuario
- Se limitan a respuestas de opción múltiple
- No permiten seleccionar preguntas y combinar cuestionarios
- No clasifican ni ordenan los reactivos
- No contemplan la posibilidad de calificar el cuestionario
- Su aplicación no es interactiva.

Mi propuesta no sólo se limita a eliminar esos inconvenientes, sino que por medio de la investigación detallada del problema se tratará de ofrecer una solución que abarcará todos los aspectos del problema y que al cumplir con estos objetivos se convierta en un instrumento indispensable para maestros, investigadores y todo aquel que diseñe preguntas y repuestas con cualquier finalidad.

El trabajo está dividido en varias secciones que se agrupan de la siguiente manera:

General.-

1.- La determinación de los objetivos y alcance del sistema.

- Objetivos generales y específicos planteados
- Hipótesis
- Áreas de aplicación esperadas

2.- Planeación del trabajo

Particulares.-

3.- Desarrollo del sistema

- Análisis del problema
- Diseño del sistema
- Pruebas y depuración
- Implantación y evaluación

4.- Conclusiones y recomendaciones

5.- Anexos

- Manual del usuario
- Reportes impresos
- Código fuente

OBJETIVOS PLANTEADOS

2

General.-

Diseñar, desarrollar e implantar un sistema computacional que simplifique la tarea de formular cuestionarios para evaluaciones y encuestas, y además sea un núcleo útil para el desarrollo de otras aplicaciones que den tratamiento a los datos recabados.

Particulares.-

- Facilitar la elaboración, modificación, edición y emisión de cuestionarios.
- Construir una herramienta que sirva de base para el desarrollo de aplicaciones relacionadas con el análisis de datos obtenidos a través de cuestionarios.
- Proveer la posibilidad de ordenar, clasificar, seleccionar, almacenar, eliminar y/o modificar los reactivos de un cuestionario.
- Dar una amplia gama de alternativas para poder diseñar los reactivos a la medida de la necesidad.
- Permitir la aplicación de exámenes en forma interactiva.
- Dar la posibilidad de calificar los exámenes por medio del mismo sistema.

La elaboración del sistema computacional para la generación de evaluaciones y encuestas:

1. Facilitará la elaboración, modificación, edición y emisión de cuestionarios.
2. Ofrecerá la posibilidad de ordenar, clasificar, seleccionar, almacenar, eliminar y/o modificar los reactivos de un cuestionario.
3. Disminuirá el tiempo empleado en la elaboración de cuestionarios, permitiendo dedicar mayor tiempo a la fase creativa del trabajo.
4. Permitirá la aplicación y calificación automática de los exámenes interactivos.
5. Mejorará la calidad total de los cuestionarios.

Existen una diversidad de motivos por los cuales se quiera realizar un cuestionario, sin embargo a fin de cuentas todos estos motivos representan algún tipo de medición o conteo.

1. Análisis del problema

Medir es determinar el valor cuantitativo de una cosa en relación a otra que sirve como patrón. Así tenemos que se puede medir el nivel de inteligencia, económico, de preferencias, de estudios, etc.

• Diseño físico

Así mismo se puede medir el resultado de un esfuerzo para saber si el producto logrado reúne las condiciones satisfactorias previstas o se necesita modificar los procedimientos, métodos, materiales etc. de manera que con esta retroalimentación se logren las metas propuestas.

Es por eso que el Sistema Generador de Evaluaciones y Encuestas no estará limitado a ciertas áreas de aplicación, por el contrario tratará de ser de tal forma versátil que pueda dar respuesta a cualquier necesidad de diseño de cuestionarios precodificados.

Las áreas para las que se espera una aplicación inmediata son:

- Educación
- Medicina
- Mercadotecnia
- Psicología
- Estadística
- Sociología

PLANEACION DEL DESARROLLO DEL SISTEMA

5

Análisis del problema o investigación preliminar

Uno de los aspectos fundamentales en el desarrollo de sistemas es comprender el problema en su totalidad; esto significa estudiarlo hasta poder definirlo y describirlo de tal manera, que de este conocimiento se puedan seleccionar los elementos que son relevantes para el desarrollo del sistema.

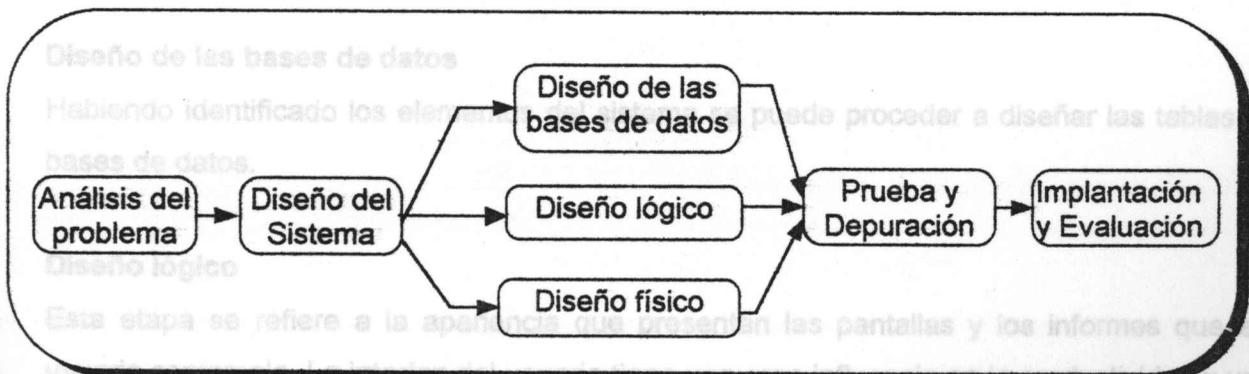
1. Análisis del problema

2. Diseño del sistema

- Diseño de las Bases de Datos
- Diseño lógico
- Diseño físico

3. Prueba y depuración

4. Implantación y evaluación



Dichas etapas corresponden al *Ciclo de vida clásico del desarrollo de sistemas*, el cual es un enfoque del desarrollo de sistemas basados en computadora. Dentro de cada fase se usarán las herramientas del método de *Análisis estructurado*, ya que éste nos va a permitir

conocer al sistema en forma lógica y manejable, al mismo tiempo que proporciona la base para asegurar que no se omite ningún detalle pertinente.

Los elementos del Análisis estructurado que se usarán son:

- símbolos gráficos
- diccionario de datos
- descripción de procesos y procedimientos
- estándares para describir y documentar el sistema.

Análisis del problema o investigación preliminar

Uno de los aspectos fundamentales en el desarrollo de sistemas es comprender el problema en su totalidad; esto significa estudiarlo hasta poder definirlo y describirlo de tal manera, que de este conocimiento se puedan seleccionar los elementos que son relevantes para el desarrollo del sistema.

La finalidad de este análisis o investigación preliminar consiste en preparar el escenario y los elementos con los que se va a trabajar durante el diseño del sistema.

Diseño del sistema

En esta etapa se producen los detalles que establecen la forma en la que el sistema cumplirá con los requerimientos identificados en la etapa de análisis.

Diseño de las bases de datos

Habiendo identificado los elementos del sistema se puede proceder a diseñar las tablas o bases de datos.

Diseño lógico

Esta etapa se refiere a la apariencia que presentan las pantallas y los informes que el usuario contempla. La interfaz del usuario tiene una gran influencia en la productividad y un fuerte impacto en el entorno de trabajo; constituye el factor primordial en una época de dependencia generalizada de las computadoras.

Diseño físico

Resulta crucial diseñar el programa de forma que se minimice el esfuerzo preciso para mantenerlo. Sin una planificación, el mantenimiento del software se convierte pronto en una parte importante del costo total.

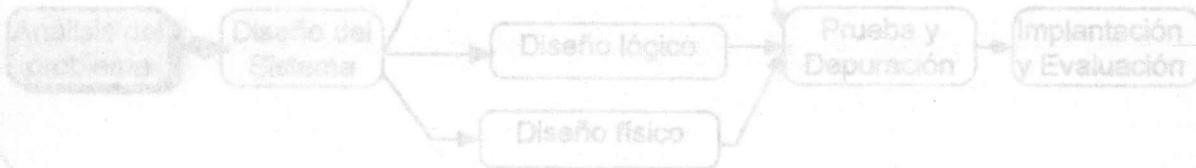
Prueba de sistemas

Antes de que los usuarios trabajen con la aplicación, ésta debe superar un programa definido de pruebas para evitar posteriores sucesos indeseables.

Durante esta fase el sistema se emplea de manera experimental para asegurarse de que el software no tenga fallas, es decir que funciona de acuerdo con las especificaciones y en la forma que se espera que lo haga.

Implantación y evaluación

La implantación es el proceso de instalar el sistema, capacitar a los usuarios y construir todos los archivos de datos necesarios para utilizarlo.



En la investigación documental se recurrió principalmente a la consulta de libros de Psicotecnia pedagógica y Didáctica (ver bibliografía), para conocer los requerimientos pedagógicos y didácticos de los cuestionarios de evaluación.

En lo que se refiere a las entrevistas, se realizaron a personas con experiencia en el ámbito educativo y de investigación, con la finalidad de comparar la información teórica obtenida de las fuentes bibliográficas con la experiencia de la práctica de aquellas personas que utilizan estas herramientas.

Los resultados obtenidos se documentaron de la siguiente forma:

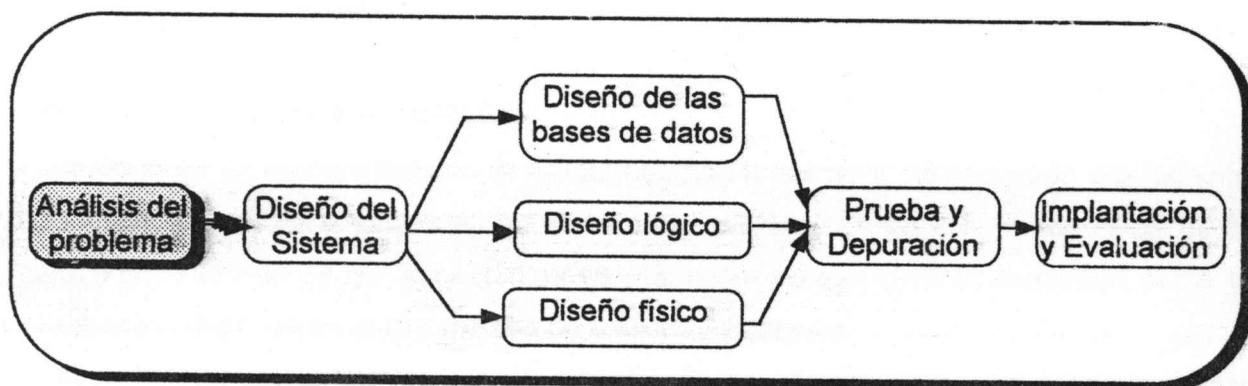
ANALISIS DEL PROBLEMA

- Se recopiló la Clasificación General de los reactivos.
- Se realizó el análisis de cada reactivo de evaluación y se determinó los criterios de evaluación y criterios de aplicación.

6

En los siguientes capítulos se presenta el resumen y análisis de la información recopilada.

El objetivo de esta etapa es comprender situaciones, no resolver problemas. Solo después de comprender en su totalidad el sistema, se está en posición de generar recomendaciones para su diseño.



En la investigación documental se recurrió principalmente a la consulta de libros de Psicotecnia pedagógica y Didáctica (ver bibliografía), para conocer los requerimientos pedagógicos y didácticos de los cuestionarios de evaluación.

En lo que se refiere a las entrevistas, se realizaron a personas con experiencia en el ámbito educativo y de investigación, con la finalidad de comparar la información teórica obtenida de las fuentes bibliográficas con la experiencia de la práctica de aquellas personas que utilizan estas herramientas.

Los resultados obtenidos se documentaron de la siguiente forma:

- Se elaboró un resumen con una revisión general de los cuestionarios precodificados.
- Se recopiló la Clasificación General de los reactivos.
- Se realizó el análisis de cada reactivo de evaluación y se determinó su estructura, criterios de evaluación y criterios de aplicación.

En los siguientes capítulos se presenta el resumen y análisis de la información recopilada.

B) PRUEBAS DE COMPOSICIÓN

Consisten en un número limitado de preguntas que el aplicador ha elaborado previamente para que el aplicando conteste con cierta amplitud de acuerdo con el contenido de la pregunta. También se les llama subjetivas o abiertas porque tanto la resolución como la evaluación depende en mayor medida de criterios personales.

Debe utilizarse principalmente para la medición de aquellos productos del aprendizaje que no se pueden medir con los elementos de la pregunta objetiva por la complejidad de sus conductas:

- a) de respuesta restringida
- b) de respuesta extensiva

C) PRUEBAS DE EJECUCIÓN

Este tipo de pruebas examinará por medio de un producto acabado o un proceso.

Haciendo un estudio de esta clasificación, descartaremos de primera instancia del análisis y descripción de reactivos las pruebas subjetivas como los son las de composición y ejecución por el hecho de que nuestra herramienta tanto para el diseño como para la evaluación de los cuestionarios es cien por ciento objetiva.

Instrumentos de Evaluación

A) PRUEBAS OBJETIVAS

Son instrumentos cuya función principal es el registro exacto de un objetivo, ya sea de definición, de conceptos, de memorización, de identificación de situaciones, de traducción de pensamientos, de aplicaciones prácticas o resolución de problemas. Lo valioso de estas pruebas es que dan oportunidad de examinar un mayor contenido; son equitativas, porque el aplicador evalúa únicamente lo que conteste el aplicando de acuerdo con las respuestas previstas, son económicas en cuanto a tiempo de resolución y evaluación. Exigen cuidado en su redacción en cuanto a contenido y forma: claridad, sencillez y precisión.

A este tipo de pruebas también se les llama cerradas porque limitan o "cierran" las repuestas a una sola posibilidad.

B) PRUEBAS DE COMPOSICIÓN

Consisten en un número limitado de preguntas que el aplicador ha elaborado previamente para que el aplicando conteste con cierta amplitud de acuerdo con el contenido de la pregunta. También se les llama subjetivas o abiertas porque tanto la resolución como la evaluación depende en mayor medida de criterios personales.

Debe utilizarse principalmente para la medición de aquellos productos del aprendizaje que no se pueden medir con los elementos de la pregunta objetiva por la complejidad de sus conductas.

a) de respuesta restringida

b) de respuesta extensiva

C) PRUEBAS DE EJECUCIÓN

Este tipo de pruebas examinan por medio de un producto acabado o un proceso. Haciendo un estudio de esta clasificación, descartaremos de primera instancia del análisis y descripción de reactivos las pruebas subjetivas como los son las de composición y ejecución por el hecho de que nuestra herramienta tanto para el diseño como para la evaluación de los cuestionarios es cien por ciento objetiva.

Las pruebas objetivas y los cuestionarios precodificados

A las pruebas objetivas también se les puede llamar **cuestionarios precodificados** ya que las preguntas están formuladas de tal manera que sólo exigen elegir respuestas preestablecidas de acuerdo con el código que se ha escogido. La elección puede efectuarse mediante una señal en el lugar reservado para tal efecto (por ejemplo una cruz o tilde).

El uso de cuestionarios precodificados puede extenderse a varias aplicaciones que podemos resumir en dos grandes grupos de acuerdo a su uso:

- Cuestionarios de evaluación
- Cuestionarios de encuesta

Cuestionarios de evaluación.-

Los reactivos que pertenecen a los cuestionarios de evaluación tienen la finalidad de registrar el logro de un objetivo, como lo es la adquisición de un determinado conocimiento, por lo cual la evaluación de la respuesta está íntimamente ligada con su estructura. Como ejemplo tenemos los siguientes cuestionarios:

- Exámenes de admisión
- Exámenes psicométricos
- Exámenes de diagnóstico
- Exámenes académicos parciales o finales de cualquier nivel educativo

Estos reactivos tienen la particularidad de que su evaluación puede ser realizada por medios electrónicos, por lo cual el Sistema Generador de Evaluaciones y Encuestas, además de proveer las herramientas necesarias para el diseño de los reactivos, tendrá la facultad de ser interactivo, es decir, podrá ser contestado directamente en la computadora y evaluado de forma automática.

Cuestionarios de encuesta.-

La finalidad principal de este tipo de cuestionarios es la recolección de información, la cual puede ser utilizada para investigaciones sociales, estudios de mercado, económicos, de población, de opinión por solo mencionar algunos, para después ser tratada, contabilizada y/o analizada para los fines que convenga. Los resultados obtenidos por este tipo de cuestionarios generalmente son cuantitativos.

Algunas características importantes de este tipo de reactivos son:

- Dirigidos a grupos medianos o grandes de personas (más de 100 individuos)
- Los sujetos de las encuestas pueden encontrarse dispersos en una organización, una colonia, una ciudad, un estado etc.
- Para evitar confusiones algunas veces son dirigidos por un encuestador.
- Los reactivos que presentan la variante tamiz no pueden ser precodificados.

Por dichas características, el modo interactivo no se considera una opción viable ni conveniente en este tipo de cuestionarios, de tal manera que el Sistema Generador de Evaluaciones y Cuestionarios constituirá una herramienta útil para el diseño de esta clase de reactivos y dejará la posibilidad de que, de acuerdo a las necesidades particulares del problema, se desarrolle una aplicación que permita la captura de los datos obtenidos y el análisis de la información.

B) Reactivos de complementación o canavá

Definición.-

Consiste en expresiones verdaderas de donde se han eliminado previamente alguna palabra, frase, número o símbolo al cual el aplicando debe completar.

Pueden ser de dos tipos:

a) De complementación simple:

Ejemplo:

- * La República Mexicana está situada al sur de.....

b) De complementación sugerida:

Ejemplo:

- * La vaca y el caballo por tener cuatro patas se llaman (cuadrúpedos/bípedos)

Clasificación general de los reactivos

Definición.-

La forma de estos reactivos consta de una base que también se le conoce como cuerpo. Por la forma de las preguntas, los reactivos se clasifican de la siguiente forma:

A) Reactivos de respuesta breve.

Definición.-

Estos reactivos como los de complementación, detallados en el inciso siguiente, consisten en que la respuesta se manifiesta mediante una palabra, frase, número o símbolo. En particular se caracteriza por el planteamiento de una interrogante y a partir de la pregunta planteada se espera una respuesta. Se le considera objetiva porque la respuesta es invariable.

Ejemplo:

Ejemplo:

- ¿Cómo se denominan los nombres que convienen a todas las personas, animales o cosas?.....
- ¿Cuántos hijos tiene?.....

B) Reactivos de complementación o caneavá

Definición.-

Consiste en expresiones verdaderas de donde se han eliminado previamente alguna palabra, frase, número o símbolo el cual el aplicando debe completar.

Pueden ser de dos tipos:

a) De complementación simple:

Ejemplo:

- La República Mexicana está situada al sur de.....

b) De complementación sugerida:

Ejemplo:

- La vaca y el caballo por tener cuatro patas se llaman (cuadrúpedos/bípedos).....

C) Reactivos de opción múltiple

respuestas:

Definición.- situación social básica

a.- Ciudad

La forma de estos reactivos consta de una base que también se le conoce como cuerpo, tronco o pie, el cual presenta una situación o problema planteados explícita o implícitamente, en una pregunta directa o en un enunciado incompleto. En el caso de evaluaciones la otra parte del reactivo lo forman las opciones que contienen la respuesta correcta y otras posibles respuestas incorrectas llamadas "distractores" su función es confundir a los evaluados que no están seguros de la respuesta. En cualquier otro caso todas las respuestas son válidas. Corrección-Incorrecta etc. En cada caso solo existen solo dos posibilidades y una respuesta.

Pueden ser de dos tipos:

a) De respuesta única.

Ejemplo: activos Falso-Verdadero evalúan aspectos de hechos específicos. ()

• Chile exporta a los demás países.....()

• a) cobre b) carbón c) salitre d) boratos culturales de la escuela? ()

a) SI b) NO

• ¿Por qué partido votaría usted las próximas elecciones?.....()

• a) PAN b) PRI c) PRD d) OTRO onto

b) De respuesta múltiple

Ejemplo en fines de evaluación este tipo de reactivo se propone que el evaluador sea capaz

• Los dos países americanos de mayor extensión territorial son: () he () s. procesos,

a) Brasil b) México c) Chile d) Canadá pueda reflejar una lista de preferencias,

aptitudes etc.

Ejemplo:

D) Reactivos de pareamiento, correspondencia o correlación.

Definición. digestivo de acuerdo con el orden que suceden.

Es una variante modificada de los reactivos de opción múltiple, se conoce con cualquiera de los nombres mencionados y consiste en una serie de hechos, acontecimientos o definiciones que se colocan en dos columnas las cuales se llaman: "premisas" y "respuestas". El objetivo es relacionar las dos columnas.

Ejemplo:

- Premisas de acuerdo a sus preferencias: respuestas
- () Una institución social básica a.- Ciudad
- () Un ejemplo de grupo social secundario b.- Familia
- () jugar en lugares cerrados c.- Nación
- () salir de compras

E) Reactivos dicotómicos

Definición.-

Consisten en aseveraciones o declaraciones que el alumno ha de marcar como Falso-Verdadero, Sí-No, Buena-Mala, Correcta-Incorrecta etc. En cada caso solo existen solo dos posibilidades y una respuesta. Muestra que es capaz de identificar un hecho o una cosa por sus características principales.

Ejemplo:

- Los reactivos Falso-Verdadero evalúan aspectos de hechos específicos. ()
a) Falso la plantas b) Verdadero la línea punteada
- ¿Le gustaría participar en las actividades culturales de la escuela? ()
a) SI en sentido inverso b) NO la raíz
- Órgano de sostén, de nutrición y de reserva

F) Reactivos de jerarquización u ordenamiento

Definición.-

Si es con fines de evaluación este tipo de reactivo se propone que el evaluador sea capaz de indicar el orden lógico o cronológico de una secuencia de hechos, procesos, acontecimientos o fenómenos; de otro modo puede reflejar una lista de preferencias, aptitudes etc.

Ejemplo:

- Anota dentro del paréntesis el número correspondiente a cada uno de los hechos del proceso digestivo de acuerdo con el orden que suceden.
() Masticación
- () Degollación
- () Aprehensión
- () Quimificación

- a) Excelente b) Bueno c) Malo d) No lo conoce

• Ordene de acuerdo a sus preferencias:

- () Un nuevo día
- () jugar al aire libre
- () jugar en lugares cerrados
- () salir de compras
- () ver televisión

G) Identificación conceptual

Definición.- de corroboración.

Tienden a que el alumno demuestre que es capaz de identificar un hecho o una cosa por sus características principales.

Ejemplo:

• Las siguientes cuestiones están arregladas por grupos y cada grupo se refiere a una parte de las plantas. Escríbela en la línea punteada:

• Indica con una paloma, cuáles de las actividades has seguido o intentado por cuenta

Crece en sentido inverso a la raíz

() Órgano de sostén, de nutrición y de reserva

() Asegura la circulación de la savia

() RESPUESTA:.....

() Grabar cintas

H) Reactivos de lista continua

Definición.-

Consisten en la presentación de 3 a 5 conceptos, hechos, sucesos etc. que se anotan señalados por letras o números. A continuación se presentan varias proposiciones (5 a 25) 5 por cada concepto, que describen hechos o características relacionadas con los conceptos.

Ejemplo:

Relaciones las dos columnas de acuerdo a la opinión que usted tenga de los siguientes programas

- a) Excelente b) Bueno c) Malo d) No lo conoce

Análisis de los reactivos de evaluación

() Un nuevo día

() Al despertar reactivos objetivos.-

() A sangre fría

() Ciudad desnuda

() Hechos los resultados serán precisos y concretos y no se alterarán por efectos de juicios, preferencias individuales o estado de ánimo de quienes califican,

2. Liberan al alumno de la tarea de la redacción y lo obligan a contestar

l) Reactivos de corroboración. sabe

Definición.-

Los estudiantes emplean la mayor parte del tiempo en leer y pensar. Consisten en una lista de proposiciones las cuales podrán ser seleccionadas de acuerdo a la instrucción.

Con una simple selección de una respuesta, el alumno habrá demostrado un conocimiento de acuerdo con el propósito de la evaluación, con el consecuente ahorro

Ejemplo:

- Indica con una paloma, cuáles de las actividades has seguido o intentado por cuenta propia. establecidas.

() Escribir un cuento

() Escribir un poema

() Hacer trabalenguas

() Grabar cintas

Desventajas.-

e Su preparación es difícil y lleva tiempo.

e El que elabora el cuestionario es el responsable de que la prueba contemple las siguientes características:

1. Unívoca: que los reactivos tengan una sola respuesta inconfundible y precisa

2. Inequivoca: que el lenguaje sea tan claro y preciso que evite cualquier interpretación falsa o errónea por parte de los examinados

Análisis de los reactivos de evaluación

Ventajas de los reactivos objetivos.-

- ☞ Evalúan objetivamente los contenidos de la materia, es decir:
 1. Los resultados serán precisos y concretos y no se alterarán por efectos de juicios, preferencias individuales o estado de ánimo de quienes califican,
 2. Liberan al alumno de la tarea de la redacción y lo obligan a contestar concretamente lo que sabe.
- ☞ Los estudiantes emplean la mayor parte del tiempo en leer y pensar.
- ☞ Con una simple selección de una respuesta, el alumno habrá demostrado un conocimiento de acuerdo con el propósito de la evaluación, con el consecuente ahorro de tiempo y la posibilidad de hacer una evaluación más extensa.
- ☞ Las preguntas están formuladas de tal manera que solo exigen respuestas preestablecidas.
- ☞ Por la naturaleza objetiva y unívoca de las respuestas (que no intervienen criterios personales para calificar y que cada reactivo tiene una sola respuesta inconfundible y precisa), la calificación de los cuestionarios puede hacerse más rápida y eficazmente por medio de un sistema computarizado.

Desventajas.-

- ☞ Su preparación es difícil y lleva tiempo.
- ☞ El que elabora el cuestionario es el responsable de que la prueba contemple las siguientes características:
 1. **Unívoca:** que los reactivos tengan una sola respuesta inconfundible y precisa
 2. **Inequívoca:** que el lenguaje sea tan claro y preciso que evite cualquier interpretación falsa o errónea por parte de los examinados

3. **Adaptada:** que se encuentre al nivel de los examinados.

A) Reactivos de respuesta breve.-

Ventajas.-

- ☞ Elimina casi por completo la posibilidad de adivinar la respuesta
- ☞ Es una forma simple de pregunta a la que el alumno se adapta naturalmente.
- ☞ Es particularmente útil en solución de problemas de física y matemáticas, en donde el resultado de un complejo razonamiento y cálculo puede ser expresado en pocos símbolos.

Desventajas.-

- ☞ Este tipo de evaluación no es completamente objetiva como las del tipo de selección de respuesta debido a que:
 1. Es casi inevitable que los alumnos den respuestas que tengan que ser revisadas particularmente y dejen sin validez la objetividad de la prueba.
 2. La cantidad de sinónimos que puede tener una palabra puede hacer muy grande el universo de respuestas.
 3. Las diferentes formas de escribir una palabra pueden provocar que una palabra escrita incorrectamente sea considerada como errónea, a pesar de que el objetivo de la pregunta esté contestado correctamente.

B) Reactivos de complementación o canevá

a) Complementación simple.-

Debido a la semejanza estructural de este tipo de reactivos con el de respuesta breve las ventajas y desventajas son las mismas para ambos tipos.

b) Complementación sugerida.-

Ventajas.-

☞ Permite al alumno visualizar la respuesta correcta y escribirla exactamente como se requiere, por lo que, a diferencia de la complementación simple, la respuesta es unívoca tanto sintáctica como morfológicamente.

☞ No da lugar a ambigüedades por el uso de sinónimos o por mala ortografía

C) Reactivos de opción múltiple..

Ventajas.-

☞ Se adaptan a la evaluación de la mayor parte de los objetivos del aprendizaje, en sus diferentes niveles taxonómicos, como:

1. conocimiento, comprensión y aplicación

2. capacidad para resolver problemas (análisis)

3. capacidad para hacer predicciones

☞ Como el número de alternativas es mayor a 2, la posibilidad de adivinar la respuesta correcta es menor que en los reactivos dicotómicos.

Desventajas.-

☞ La construcción de uno de estos reactivos con cuatro opciones puede requerir tanto trabajo como la construcción de cuatro dicotómicos o de respuesta simple, dado que los distractores deben ser plausibles y gramaticalmente congruentes con la pregunta.

D) Reactivos de correspondencia.-

Ventajas.-

☞ Es compacto porque usa las mismas alternativas para un grupo de preguntas, lo cual se ve reflejado en eficiencia en términos de tiempo y espacio.

☞ Resultan más fáciles de elaborar por no necesitar tantos distractores.

✖ No es útil para aprendizajes complejos.

Desventajas.-

ⓐ Identificación conceptual.-

☞ No son adecuados para aprendizajes complejos.

Ventajas.-

☞ Quizá la principal desventaja es que requiere gran cuidado para que una respuesta no sea considerada inválida por claves irrelevantes, alternativas inválidas y/o un arreglo difícil de manejar.

Desventajas.-

E) Reactivos dicotómicos.-

☞ Por requerir una respuesta no sugenda tienen las mismas desventajas que los

Ventajas.-

☞ Son útiles para explorar la memorización de hechos específicos.

☞ Se adaptan para evaluar la habilidad de distinguir conceptos populares de verdades científicamente validadas.

☞ Son convenientes para cualquier reactivo que solo tenga dos respuestas posibles y una correcta.

Desventajas.-

☞ Material como inferencias, interpretaciones, explicaciones y generalizaciones no puede ser puesto en la forma Falso- Verdadero, dado que no pueden ser considerados como completamente falsos o completamente verdaderos.

F) Reactivos de Ordenamiento.-

Ventajas.-

☞ Son útiles para examinar aprendizajes de aquellos asuntos que tienen entre sí una relación cronológica, genética, lógica o proporcional.

Desventaja.-

- ☞ No es útil para aprendizajes complejos.

G) Identificación conceptual.-

Ventajas.-

- ☞ Son útiles para el reconocimiento y/o identificación de un hecho a través de sus características principales.

Desventajas.-

- ☞ Por requerir una respuesta no sugerida tienen las mismas desventajas que los reactivos de respuesta breve.
- ☞ La elaboración de las proposiciones es un trabajo largo y difícil.

H) Reactivos de lista continua.-

Tanto las ventajas como las desventajas son las mismas que los reactivos de correspondencia, la diferencia estriba en que un mismo concepto puede ser respuesta de varias premisas.

CONCLUSIONES

Los reactivos de respuesta breve, de complementación simple y de identificación conceptual, a pesar de sus ventajas, no cumplen los requisitos de objetividad indispensables para poder ser precodificados y trasladarlos a modelos entendibles por la computadora, por lo cual no serán seleccionados como herramientas de evaluación dentro del sistema propuesto.

Respuesta única.

Recomendaciones y restricciones para la construcción de reactivos de evaluación.

Respuesta múltiple.

A continuación se presenta una recopilación de los criterios más importantes que han de considerarse para la construcción de los reactivos de evaluación anteriormente analizados,

Dicotómicos.

Complementación, Respuesta única y respuesta múltiple.

- ▲ Deben elaborarse con el fin de medir resultados importantes que se presenten a partir de enunciados significativos.
- ▲ No es recomendable plantear afirmaciones literales de los textos en la premisa del reactivo, ya que pueden resultar muy generales y confusas.
- ▲ La premisa puede ser una pregunta directa o un enunciado incompleto. Se recomienda la pregunta directa porque encierra en sí misma el problema que se quiere resolver y es más fácil de comprender.
- ▲ Redacte la premisa de la forma más clara, simple y correctamente posible.
- ▲ Todos los distractores deben ser gramaticalmente congruentes con la pregunta.
- ▲ Los distractores deben ser plausibles.
- ▲ Haga variar la longitud de la respuesta correcta para evitar que ésta sea una posible clave.
- ▲ Cada reactivo debe tener únicamente una contestación correcta
- ▲ Procure que cada reactivo sea independiente de los demás
- ▲ Las respuestas serán una palabra, frase corta, símbolo o número.

Complementación.

- ▲ El número recomendable de opciones es 2; el número límite 4.

Respuesta única.

- ▲ El número recomendable de opciones es 5 el máximo 6.

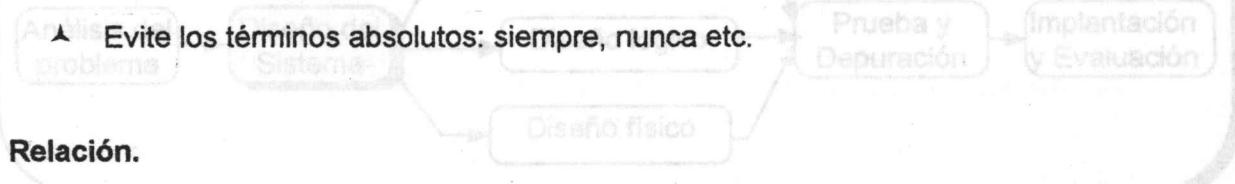
Respuesta múltiple.

- ▲ El número recomendable de opciones es 5 el máximo 6.



Dicotómicos.

- ▲ Evite el uso de planteamientos negativos, especialmente aquéllos que contienen doble negación.
- ▲ Evite la inclusión de elementos que proporcionen pistas en las respuestas. Como los determinantes específicos: usualmente, a menudo, normalmente etc.



- ▲ Evite los términos absolutos: siempre, nunca etc.

Relación.

- ▲ Use material homogéneo en las premisas y respuestas.
- ▲ Las respuestas deben consistir en palabras, frases cortas, números u otro elemento que pueda ser examinado fácilmente.
- ▲ Proporcione una respuesta de más como distractor.
- ▲ El número máximo de premisas es 10, el mínimo 3

Ordenamiento

- ▲ Los reactivos deben ser homogéneos.
- ▲ El número máximo de premisas es de 6, el mínimo 4.

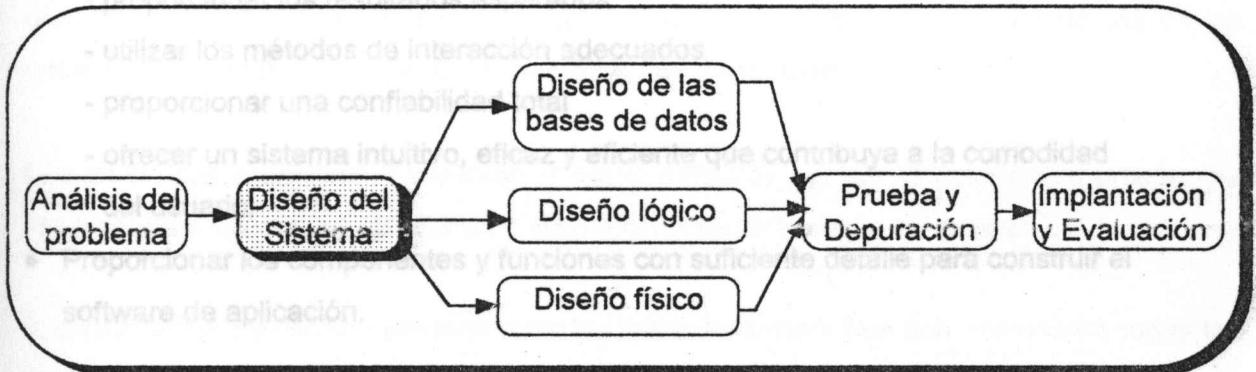
¹ Taylor, E. S. An Interim Report on Engineering Design, MIT 1959.



DISEÑO DE LA SOLUCION



- Especificar los elementos del diseño lógico, es decir entradas, salidas, estructuras de datos, bases de datos y procedimientos
- Satisfacer las necesidades de los usuarios en términos de:
 - efectuar en forma correcta los procedimientos
 - presentar en forma apropiada la información
 - proporcionar los resultados esperados
 - utilizar los métodos de interacción adecuados
 - proporcionar una confiabilidad adecuada
 - ofrecer un sistema intuitivo, eficaz y eficiente que contribuya a la comodidad
- Proporcionar interfaces, menús y funciones con suficiente detalle para construir el software de aplicación.



Definición.

El diseño es el primer paso de la fase de desarrollo de cualquier sistema. Puede definirse como: "...el proceso de aplicar distintas técnicas y principios con el propósito de definir un dispositivo, proceso o sistema con los suficientes detalles como para permitir su realización física".¹

Transición del análisis hacia el diseño.

En otras palabras, el diseño es la conversión de los requerimientos en métodos que los satisfagan. La esencia del diseño de sistemas es la selección y especificación de las características del sistema de información a partir de los conocimientos adquiridos en la etapa de análisis.

¹ Taylor., E. S. An Interim Report on Engineering Design, MIT 1959.

Objetivos del diseño.

Lo que se persigue al diseñar el sistema es:

- Especificar los elementos del diseño lógico, es decir entradas, salidas estructuras de datos, bases de datos y procedimientos
- Satisfacer las necesidades de los usuarios en términos de:
 - efectuar en forma correcta los procedimientos
 - presentar en forma apropiada la información
 - proporcionar los resultados esperados
 - utilizar los métodos de interacción adecuados
 - proporcionar una confiabilidad total
 - ofrecer un sistema intuitivo, eficaz y eficiente que contribuya a la comodidad del usuario.
- Proporcionar los componentes y funciones con suficiente detalle para construir el software de aplicación.

Especificaciones de los elementos lógicos del diseño.

En esta fase se determinan las especificaciones detalladas del nuevo sistema, es decir aquellas que describen sus características: salidas, entradas, estructuras de datos, bases de datos y los procedimientos, todo en una forma que satisfaga los requerimientos del proyecto.

Elemento	Descripción
Clase	Clase del reactivo
Clave	Clave del reactivo
Esfera	Esfera que cubre el reactivo
Escuela	Institución para la cual se utiliza el reactivo
Tema	Tema del reactivo
Grado	Grado o grupo al que se aplicará el reactivo
Valor Absoluto (VA)	Puntuación absoluta del reactivo
Valor Relativo (VR)	Puntuación relativa del reactivo
Tiempo	Tiempo en minutos para responder el reactivo
Libre 1	Uso del evaluador
Libre 2	Uso del evaluador
Fecha C	Fecha de creación
Fecha M	Fecha última modificación

Elementos particulares.-

Estructura y proceso de evaluación de los reactivos.

En esta sección se determinarán los elementos necesarios para la definición y aplicación de

Estructura general de los reactivos.-

La estructura de los reactivos se divide en dos tipos de elementos, aquellos que son comunes a todos los reactivos y los que son exclusivos a su estructura.

Los primeros tendrán la finalidad de facilitar la clasificación, ordenación, diseño y calificación de los exámenes las cuales son actividades que pertenecen a todos los reactivos. La determinación de estos campos se hizo con la información obtenida en las entrevistas y tomando en cuenta las herramientas disponibles.

Definición de un reactivo.-

Los elementos exclusivos o particulares están definidos por las características propias del reactivo, mismas que se estudiaron y definieron durante la etapa del análisis.

A continuación se definen primeramente aquellos elementos que son comunes a todos los reactivos y luego, según cada clase de reactivo, sus elementos particulares.

Elementos comunes.-

Elemento	Descripción
Clase	Clase del reactivo
Clave	Clave del reactivo
Elabora	Profesor que elabora el reactivo
Escuela	Institución para la cual se utiliza el reactivo
Materia	Materia del reactivo
Tema	Tema del reactivo
Grado	Grado o grupo al que se aplicará el reactivo
Valor Absoluto (VA)	Puntuación absoluta del reactivo
Valor Relativo (VR)	Puntuación relativa del reactivo
Tiempo	Tiempo en minutos para responder el reactivo
Libre 1	Uso del evaluador
Libre 2	Uso del evaluador
Fecha_C	Fecha de creación
Fecha_M	Fecha última modificación

Elementos particulares.-

Aplicación:

En esta sección se determinarán los elementos necesarios para la definición y aplicación de los reactivos (estructuras de datos), el proceso de evaluación y se expondrán algunas consideraciones especiales para cada reactivo.

Se podrá observar que será necesario almacenar las posiciones en las que se desplegaron las respuestas de cada examen; esto se debe principalmente a que en cada emisión, estos últimos tendrán una ubicación diferente de las respuestas, con la ventaja consecuente de que a mayor número de exámenes las probabilidades de que haya dos tipos de exámenes iguales son más escasas.

Definición de un reactivo.-

En lo que se refiere a la definición de los reactivos es común a todos ellos y se hará una sola vez para cada reactivo diferente. Se capturarán las claves de los elementos comunes a los reactivos (Profesor, Escuela, Materia, Tema, Grado, Libre_1, Libre_2, etc.), si ya existen y si no, se podrán dar de alta. Además se capturará la premisa o enunciado y las respuestas, siendo obligatorio que la respuesta correcta siempre se capture en el primer casillero.

1) Dicotómicos

Estructuras de datos.-

Dato	Utilidad	Tipo de dato
Premisa	Enunciado o pregunta	Memo
a_Resp(2)	Grupo de respuestas	Arreglo de cadenas
cSel	Selección actual	Alfabético
nVA	Valor absoluto obtenido	Numérico
nVR	Valor relativo obtenido	Numérico
nPos	Posición de la respuesta correcta	Numérico

Proceso de evaluación.-

Si cSel = CHR(nPos + 64)

nVA := VA

nVR := VR

SINO

nVA := 0

nVR := 0

FINSI



Aplicación:

En esta clase de reactivo como en repuesta única y respuesta múltiple la respuesta no se teclea, sino que se selecciona la letra de la opción que se cree que es correcta, ya sea por medio del teclado o por medio del mouse, de tal manera que lo que se compara es la letra que corresponde a la selección actual con la letra que corresponde a la respuesta correcta.

2) Opción múltiple respuesta única

Estructura de datos.-

Base	Utilidad	Tipo de dato
Premisa	Enunciado	Memo
Resp(6)	Grupo de respuestas	Vector de cadenas
cSel	Selección	Alfabético
nVA	Val. absoluto obtenido	Numérico
nVR	Val. relativo obtenido	Numérico
nPos	Posición correcta	Numérico

Proceso de evaluación.-

Si cSel = CHR(nPos + 64)

nVR := VR

nVA := VA

SINO

nVR := 0

nVA := 0

FINSI

Nota:

Las consideraciones hechas para los reactivos dicotómicos son las mismas que para los reactivos de Opción múltiple respuesta única.

3) Complementación sugerida.-

Estructura de datos.-

Variable	Utilidad	Tipo de dato
Premisa	Enunciado	Memo
Resp(6)	Grupo de respuestas	Vector de cadenas
cSel	Selección tecleada	Cadena de caracteres
nVA	Val. absoluto obtenido	Numérico
nVR	Val. relativo obtenido	Numérico

Proceso de Evaluación.-

```

Si cSel = Resp(1)
    nVR := VR
    nVA := VA
SINO
    nVR := 0
    nVA := 0
FINSI

```

6) Jerarquización u Ordenamiento

Aplicación de los reactivos de complementación:

1. Este tipo de reactivos tiene la particularidad de que la respuesta seleccionada debe ser tecleada por el evaluado.
2. La respuesta correcta dentro del vector de respuestas, por conveniencia, será siempre el primer elemento. Al igual que los reactivos de opción múltiple respuesta única y dicotómicos.
3. Para esta clase de reactivos se hará la comparación directa de la respuesta seleccionada con la respuesta correcta.
4. No se aceptarán cadenas que no correspondan a alguna de las opciones propuestas.

Proceso de evaluación.-

4) Respuesta múltiple.-

Estructuras de datos.-

Base	Utilidad	Tipo de dato
Premisa	Enunciado	Memo
Resp(6)	Grupo de respuestas	Vector de cadenas
nSel(2)	Selección	Vector numérico
nVA	Valor absoluto obtenido	Numérico
nVR	Valor relativo obtenido	Numérico
nPos(2)	Posiciones correctas	Vector numérico

Proceso de evaluación.-

* ordenación ascendente de las selecciones

```

Si nSel(1) > nSel(2)
    temp := nSel(1)
    nSel(1) := nSel(2)
    nSel(2) := temp
FINSI
Si nSel(1) = nPos(1) and nSel(2) = nPos(2) && las dos son correctas
    nVA := VA
    nVR := VR
SINO
    Si nSel(1) = nPos(1) or nSel(2) = nPos(2) && una es correcta
        nVA := VA/2
        nVR := VR/2
    SINO
        nVA := 0
        nVR := 0
        && ninguna es correcta
    FINSI
FINSI

```



Proceso de evaluación.-

1. Esta clase de reactivos permite dos respuestas correctas. Para conservar la uniformidad con los reactivos de complementación sugerida y de respuesta única las respuestas correctas se colocarán en los dos primeros elementos del arreglo.
2. El vector nPos() contendrá las posiciones correctas actuales en orden ascendente.
3. El vector nSel() contendrá las selecciones del evaluado en orden ascendente.

FINHAZ

6) Jerarquización u Ordenamiento

Estructuras de datos.-

Base	Utilidad	Tipo de dato
Prem(10)	Grupo de premisas	Vector de cadenas
nSel(10)	Selección	Vector numérico
nVA	Valor absoluto obtenido	Numérico
nVR	Valor relativo obtenido	Numérico
nPos(10)	Posiciones correctas	Numérico

Proceso de evaluación.-

nAciertos := 0

i:=1

HAZ MIENTRAS i <=n

SI nSel(i) = nPos(i)

nAciertos := nAciertos + 1

FINSI

i := i + 1

FINHAZ

nVA := (VA/n) * nAciertos

nVR := (VR/n) * nAciertos

Nota:

La captura de las premisas deberá hacerse en el orden correcto, ya que éste servirá de referencia para la evaluación del reactivo.

6) Pareamiento, correspondencia o relación

Estructuras de datos.-

Base	Utilidad	Tipo de dato
Prem(10,1)*	Grupo de premisas	Vector de cadenas
Resp(10,2)*	Grupo de respuestas	Vector de cadenas
nSel(10)	Selección	Vector numérico
nVA	Valor absoluto obtenido	Numérico
nVR	Valor relativo obtenido	Numérico
nPos(10)	Posiciones correctas	Vector numérico

* incluyendo el distractor, que podrá ser una premisa, una respuesta o ambas.

Proceso de evaluación.-

nAciertos := 0, i := 1

HAZ MIENTRAS $i \leq n$

SI $nSel(i) = nPos(i)$

nAciertos := nAciertos + 1

FINSI

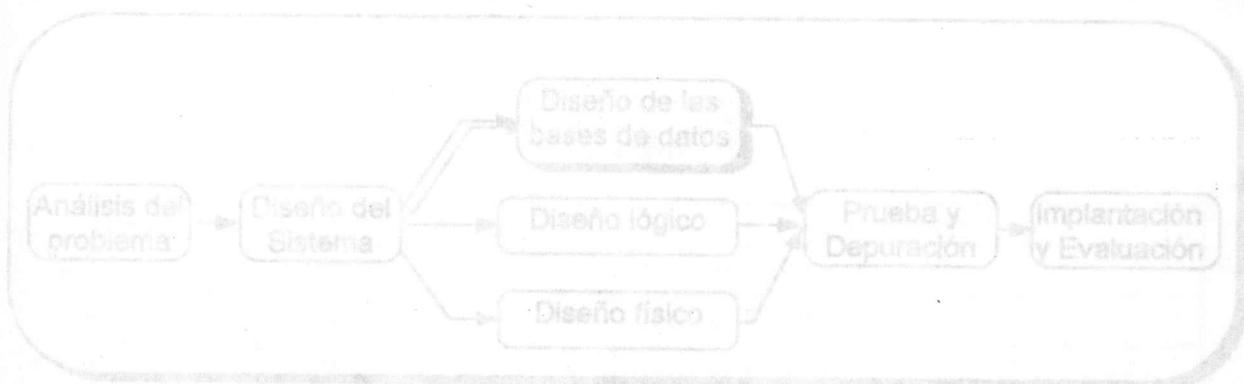
i := i + 1

FIN HAZ

nVA := $(VA/n) * nAciertos$

nVR := $(VR/n) * nAciertos$

DISEÑO DE LAS BASES DE DATOS



Podría decirse que el diseño de datos es la primera y de alguna manera la más importante de las actividades de diseño. El impacto de la estructura de datos sobre la estructura del programa, hace que el diseño de datos tenga una gran influencia en la calidad del software.

El diseño de las bases de datos consiste en determinar qué información requiere ser almacenada de forma permanente, los tipos de datos que contiene esta información y el tamaño de esos datos. Así mismo es importante formar entidades que contengan los grupos de datos pertenecientes al mismo contexto de esta manera la redundancia se evita y la introducción de datos se simplifica, y proveer de los mecanismos necesarios para relacionar los datos cuando sea necesario.

A continuación se presenta una descripción de la utilidad de cada base de datos que se diseñó, sus estructuras, sus índices, y las relaciones que mantienen con otros archivos.

DISEÑO DE LAS BASES DE DATOS

Base de Datos de Reactivos (Evaluaci.dbf)

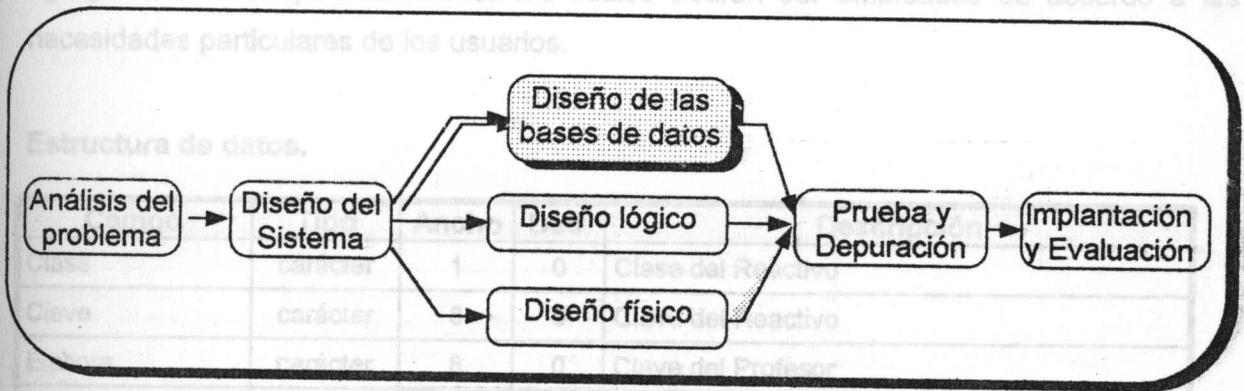
DATOS

Descripción

La información que contiene esta base de datos es la que concierne a los reactivos. La estructura fue formada a partir de los resultados de los requerimientos, en el cual se determinó que sería útil que con cada reactivo se agregaran una serie de atributos que facilitarían la clasificación y selección de los mismos. Además se agregaron dos campos comodines los cuales podrán ser empleados de acuerdo a las necesidades particulares de los usuarios.



Estructura de datos.



Podría decirse que el diseño de datos es la primera y de alguna manera la más importante de las actividades de diseño. El impacto de la estructura de datos sobre la estructura del programa, hace que el diseño de datos tenga una gran influencia en la calidad del software.

El diseño de las bases de datos consiste en determinar qué información requiere ser almacenada de forma permanente, los tipos de datos que contiene esta información y el tamaño de esos datos. Así mismo es importante formar entidades que contengan los grupos de datos pertenecientes al mismo contexto de esta manera la redundancia se evita y la introducción de datos se simplifica. y proveer de los mecanismos necesarios para relacionar los datos cuando sea necesario.

A continuación se presenta una descripción de la utilidad de cada base de datos que se diseñó, sus estructuras, sus índices, y las relaciones que mantienen con otros archivos.

Diccionario de Datos

Esta base de datos tiene un índice con la siguiente expresión (Clase + Clave), con el objetivo de tener un identificador propio e irrepetible de cada reactivo. Como puede verse, se permite tener reactivos dentro de la misma Clase, pero si en

Descripción

La información que contiene esta base de datos es la que concierne a la definición de los reactivos. La estructura fue formada a partir de los resultados del análisis de requerimientos, en el cual se determinó que sería útil que con cada reactivo se incluyera una serie de atributos que facilitarían la clasificación y selección de los mismos. Además se agregaron dos campos comodines los cuales podrán ser empleados de acuerdo a las necesidades particulares de los usuarios.

Estructura de datos.

Campo	Tipo	Ancho	Dec.	Descripción
Clase	carácter	1	0	Clase del Reactivo
Clave	carácter	8	0	Clave del Reactivo
Elabora	carácter	8	0	Clave del Profesor
Escuela	carácter	8	0	Clave de la Escuela
Materia	carácter	8	0	Clave de la Materia
Tema	carácter	8	0	Clave del Tema
Grado	carácter	8	0	Clave del Grado
VA	numérico	5	2	Valor Absoluto del reactivo
VR	numérico	5	2	Valor Relativo del reactivo
Minutos	numérico	5	2	Tiempo de respuesta estimado
Libre_1	carácter	8	0	Clave Libre para el usuario
Libre_2	carácter	8	0	Clave Libre para el usuario
Fecha_M	date	8	0	Fecha de la última modificación
Fecha_C	date	8	0	Fecha de creación del reactivo
Premisa	memo			Premisa del reactivo
RespChk	numérico	5	0	Verificador del arreglo de respuestas
Respuestas	memo			Arreglo de respuestas

Índices.

Esta base de datos tiene un índice con la siguiente expresión (Clase + Clave), con el objetivo de tener un identificador propio e irrepetible de cada reactivo. Como puede apreciarse no podrán existir claves repetidas dentro de la misma Clase, pero si en diferentes.

La razón por la cual se optó por clasificar los campos de Profesor, Escuela, Materia, Tema, Grado, Libre_1 y Libre_2 por medio de una clave es para evitar ambigüedades y redundancias y reducir el esfuerzo de captura y el espacio de almacenamiento que provoquen una disminución en la efectividad del sistema, es decir que solamente se reconocerán Claves que hayan sido dadas de alta previamente en la Base de Datos del Catálogo de Claves.

Relación con otras Bases de Datos.

Base de Datos	Llave	Tipo de relación
Catalogo.dbf	Clase + Clave	Cliente

Base de Datos del Catálogo.(Catalogo.dbf)

Descripción.

Esta base de datos contiene las Claves y su descripción, utilizadas en la tabla de Reactivos.

Estructura de datos.

Campo	Tipo	Ancho	Dec.	Descripción
Tipo	carácter	1	0	Clase de la clave
Clave	carácter	8	0	Clave
Nombre	carácter	50	0	Descripción de la clave

Índices.

Esta base de datos tiene un índice con la expresión (Tipo + Clave), con el objetivo de identificar a las descripciones de las claves. No podrán existir claves repetidas dentro de la misma Clase, pero sí en diferentes Clases.

Relación con otras Bases de Datos.

Base de Datos	Llave	Tipo de relación
Evaluaci.dbf	Clase + Clave	Servidor

Base de Datos del Guión. (<Guion>.dbg)

Descripción.

Esta tabla contiene los datos necesarios para formar el guión del examen. Se caracteriza porque tiene una extensión .DBG.

Estructura de datos.

Campo	Tipo	Ancho	Dec.	Descripción
Clase	carácter	1	0	Clase del reactivo
Clave	carácter	8	0	Clave del reactivo
VA	numérico	5	2	Valor Absoluto del reactivo
VR	numérico	5	2	Valor Relativo del reactivo
Orden	carácter	10		Orden de impresión de las respuestas
Orden2	carácter	10		Orden de impresión 2ª columna. Relación

Relación con otras Bases de Datos.

Base de Datos	Llave	Tipo de relación
Evaluaci.dbf	Clase + Clave	Cliente
Bitacora.dbf	Clave	Cliente
Examen.dbf	Clase + Clave	Servidor

Base de Datos del Examen. (<examen>.dbe)

Descripción.

Una vez formado el Guión de un examen, éste puede ser aplicado cuantas veces sea necesario y cada vez que se realice este proceso se podrá formar un examen nuevo o añadirlo a uno existente (solo si se trata de un examen interactivo; en los exámenes impresos esta base de datos no se utiliza). La extensión que caracteriza a esta tabla es .DBE.

Estructura de datos.

Campo	Tipo	Ancho	Dec.	Descripción
Examen	carácter	4	0	Numeración consecutiva de los exámenes
Pagina	carácter	3	0	No. de página al que pertenece el reactivo
Clase	carácter	1	0	Clase del reactivo
Clave	carácter	8	0	Clave del reactivo
Orden	carácter	10	0	Orden de despliegue de las respuestas
Orden2	carácter	10	0	Orden de despliegue de la 2ª columna
Respuesta	carácter	10	0	Respuesta(s) del alumno
VA_O	numérico	5	2	Valor Absoluto obtenido
VR_O	numérico	5	2	Valor Relativo obtenido

Índices.

El índice corresponde a la expresión (Examen + Página) que es de utilidad cuando se está resolviendo el examen interactivo para localizar la página en la que se está trabajando.

Relación con otras Bases de Datos.

Base de Datos	Llave	Tipo de relación
Guion.dbf	Clase + Clave	Cliente
Bitacora.dbf	Página	Cliente

Base de Datos de la Bitácora.(Bitacora.dbf)

Descripción.

Esta Base de Datos tiene la función de guardar información general referente a los archivos, al entorno, los exámenes etc. La utilidad de esta base de datos radica en que a pesar de contener información hasta cierto punto heterogénea, está provista de los elementos necesarios para distinguirla, y actuar como una base de datos que contiene otras bases de datos; dicha característica es posible gracias al campo memo que da la posibilidad de guardar todas las clases de tipos de datos de manera independiente en cada registro.

Estructura de datos.

Estructura de datos.

Campo	Tipo	Ancho	Dec.	Descripción
Tipo	carácter	1	0	Clasificación de la información que contiene el registro y el tipo de datos que guarda el campo Memo
Clave	carácter	8	0	Número aleatorio que es la llave de acceso a los registros
DatosChk	numérico	5	0	Comprobación del campo Memo
Datos	memo			Arreglo con la información pertinente

Relación con otras bases de datos

Índices.

El índice de esta base de datos se llama ALEATORIO, y utiliza el campo Clave. Sirve para relacionar el guión y el examen con la bitácora para obtener la información que de ellos se almacenó en dicha base de datos.

Relación con otras Bases de Datos.

Base de Datos	Llave	Tipo de relación
Examen.dbf	Clave	Servidor
Guion.dbf	Clave	Servidor

Base de datos de Calificaciones (<nombre>.dbc)

Descripción.

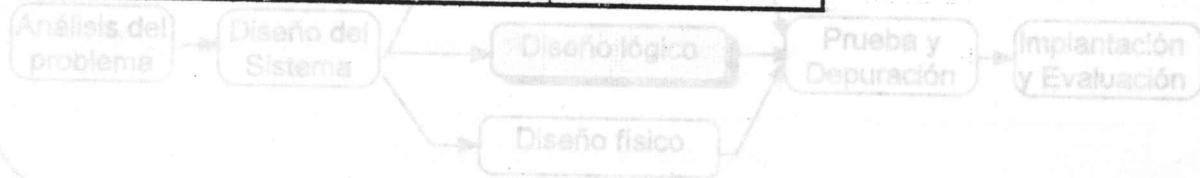
Esta base de datos contiene los datos de los alumnos y las calificaciones obtenidas en los exámenes interactivos; es generada al realizar el proceso de calificar y seleccionar la salida de los resultados hacia archivo. Podrá haber un archivo de calificaciones (*.DBC) por cada archivo de examen (*.DBE).

Estructura de datos.

Campo	Tipo	Ancho	Dec.	Descripción
Guion	carácter	12	0	Nombre del guión que dió forma al examen
Alumno	carácter	50	0	Nombre del alumno que resolvió el examen
Absoluto	numérico	5	2	Calificación absoluta obtenida
Relativo	numérico	5	2	Calificación relativa obtenida (en base a 100)
Aplico	carácter	8	0	Clave del profesor que aplicó el examen
Fecha	date	8	0	Fecha en que fué aplicado el exámen

Relación con otras bases de datos

Base de Datos	Llave	Tipo de relación
Bitacora.dbf	Clave	Cliente



El diseño lógico es la etapa en la que se describen las especificaciones detalladas del sistema, es decir, salidas, entradas, archivos, bases de datos y procedimientos.

Objetivos.-

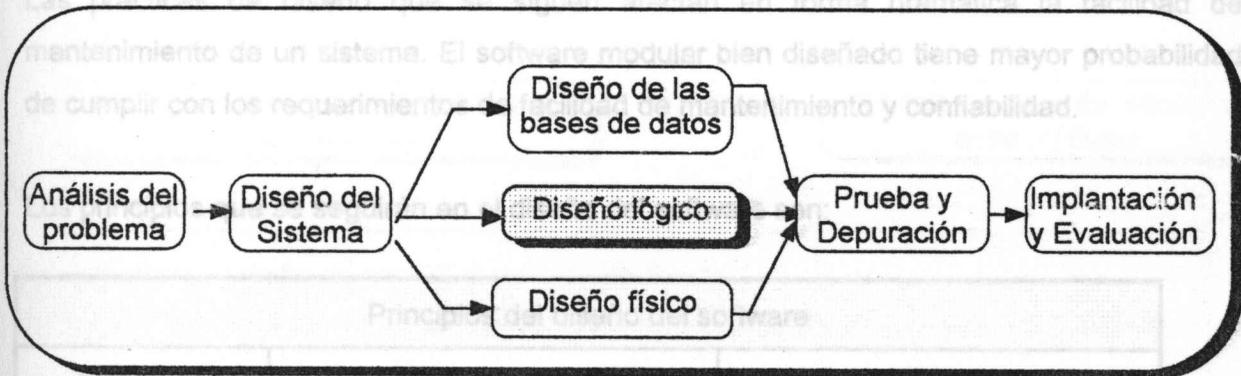
Los dos objetivos operacionales de diseño que siempre se buscan son la confiabilidad y la facilidad de mantenimiento del sistema. A continuación se detallan las características de cada uno de estos.

Diseño de sistemas confiables.-

Hay dos niveles de confiabilidad. El primero es que el sistema cumpla con los requerimientos correctos, es decir que realice las operaciones que quiere el usuario y como este lo quiere. El segundo nivel tiene que ver con los resultados reales que el sistema entrega al usuario. En este nivel, la confiabilidad del sistema se entrelaza con la ingeniería de software y su desarrollo.



1. Precisión al definir los requerimientos del usuario
2. Preparar lo mejor posible la documentación del sistema
3. Hacer uso de estándares de diseño
4. Realizar un diseño modular



El diseño lógico es la etapa en la que se describen las especificaciones detalladas del sistema, es decir, salidas, entradas, archivos, bases de datos y procedimientos.

Objetivos.-

Los dos objetivos operacionales de diseño que siempre se buscan son la confiabilidad y la facilidad de mantenimiento del sistema. A continuación se detallan las características de cada uno de estos.

Diseño de sistemas confiables.-

Hay dos niveles de confiabilidad. El primero es que el sistema cumpla con los requerimientos correctos, es decir que realice las operaciones que quiere el usuario y como este lo quiere. El segundo nivel tiene que ver con los resultados reales que el sistema entrega al usuario. En este nivel, la confiabilidad del sistema se entrelaza con la ingeniería de software y su desarrollo.



Diseño de sistemas fáciles de mantener.-

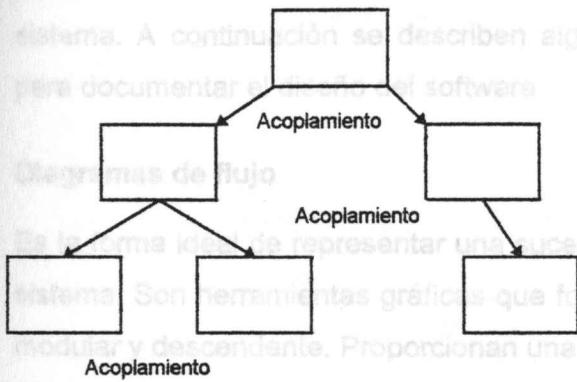
Las claves para reducir las necesidades de mantenimiento, al igual que para hacer posible que se realicen las tareas esenciales de la forma más eficiente son:

1. Precisión al definir los requerimientos del usuario
2. Preparar lo mejor posible la documentación del sistema
3. Hacer uso de estándares de diseño
4. Realizar un diseño modular

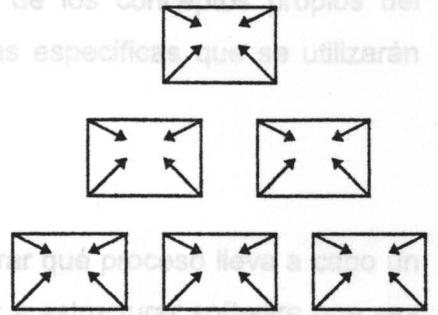
Las prácticas de diseño que se siguen afectan en forma dramática la facilidad de mantenimiento de un sistema. El software modular bien diseñado tiene mayor probabilidad de cumplir con los requerimientos de facilidad de mantenimiento y confiabilidad.

Los principios que se seguirán en el diseño del software son:

Principios del diseño del software		
Principio	Descripción	Objetivo
Modularidad y Fragmentación	Diseño de un sistema orientado a módulos	Diseñar la estructura en forma descendente con módulos que realicen funciones específicas
Acoplamiento	La fuerza de relación entre módulos	Maximizar la independencia entre los módulos minimizando el acoplamiento
Cohesión (Integración)	La fuerza de las relaciones dentro de un módulo	Maximizar la cohesión; los elementos altamente relacionados deben estar en el mismo módulo.
Tamaño	Número de instrucciones que componen a un módulo	Limitar el tamaño de forma que la función de todo el módulo se centre en un solo propósito.
Uso compartido	Uso de un módulo por otros módulos	Evitar la duplicación permitiendo que los módulos sean llamados por otros que necesitan la función de cada uno



Acoplamiento: fuerza de relaciones entre módulos



Cohesión: fuerza de relaciones entre módulos

Acoplamiento y cohesión en el diseño de software

Diseño Estructurado.

El diseño estructurado es una aproximación modular y descendente a la tarea de desarrollo del software. Los procesos se dividen en su componentes mínimos (razonablemente admisibles), actividad denominada descomposición. El sistema entero puede así diseñarse y codificarse por módulos.

Como puede notarse las ventajas de la modularidad y el diseño estructurado son claras:

- Es posible probar los módulos según se van terminando, detectando sus problemas más fácilmente que si se integrara todo el sistema antes de hacerle pruebas.
- Un sistema modular se puede comprender mejor y su mantenimiento resulta por tanto más simple; esto es cierto no solo para el desarrollador inicial que tiene que hacer cambios más tarde, sino para otras personas que puedan heredar el programa.

Herramientas de diseño lógico.

Existen algunas herramientas de diseño que a la vez de ayudar en la tarea del diseño, constituyen vías preferentes de comunicación al usuario de los conceptos propios del sistema. A continuación se describen algunas herramientas específicas que se utilizarán para documentar el diseño del software

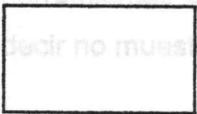
Diagramas de flujo

Es la forma ideal de representar una sucesión lógica o ilustrar qué proceso lleva a cabo un sistema. Son herramientas gráficas que forzan al diseñador a estructurar software que sea modular y descendente. Proporcionan una estructura a la que se puede ajustar el desarrollo del software de aplicación.

Elementos básicos.

Son tres los elementos principales usados en el desarrollo de los diagramas de flujo: *proceso, decisión e iteración.*

Proceso.



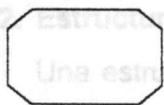
Los procesos o pasos de un programa se representan mediante un rectángulo. Este símbolo representa la iniciación de variables, actividades de entrada y salida y las llamadas para ejecutar otros procedimientos.

Decisión.



El símbolo de decisión representa condiciones alternativas que pueden ocurrir y que el programa debe de poder manejar. Son el equivalente a la estructura de programación IF-THEN-ELSE.

Iteración.



El símbolo de iteración representa los ciclos y repetición de operaciones mientras exista una condición dada o hasta que se presente una condición.

Conector.



Este símbolo es muy útil cuando se desea representar el enlace entre dos o más rutinas también se utiliza para facilitar la lectura y comprensión de los diagramas de flujo.

3. Estructuras de iteración.

En las actividades rutinarias de operación es común encontrar que algunas de aquellas se repiten mientras existen ciertas condiciones o hasta que éstas se presentan.

Español Estructurado.

La sintaxis de estas estructuras es la siguiente:

MIENTRAS < condición >

El español estructurado o pseudocódigo es otra herramienta del diseño lógico que es muy útil para evitar los problemas de ambigüedades del lenguaje al establecer condiciones y acciones, tanto en procedimientos como en decisiones. Este método no hace uso de las herramientas gráficas, en su lugar utilizan declaraciones para describir los procesos. Es decir no muestra las acciones o decisiones, las declara.

Al igual que los diagramas de flujo, el español estructurado emplea tres tipos básicos de declaraciones para describir un proceso:

1. Estructuras de decisiones

Dichas estructuras aparecen cuando se pueden emprender dos o más acciones, lo que depende de una condición específica.

La sintaxis de las estructuras de decisiones es la siguiente:

SI <condición>

<acciones>

SINO

<acciones>

FINSI

2. Estructuras de secuencia

Una estructura de secuencia es solo un paso o acción incluido en un proceso. Éste no depende de la existencia de una condición y, cuando se encuentra, siempre se lleva a cabo. Por lo general se emplean varias instrucciones en secuencia para describir un proceso.

La sintaxis se forma por medio de expresiones verbales.

Guardar en el archivo

Capturar datos, etc.

3. Estructuras de iteración.

En las actividades rutinarias de operación es común encontrar que algunas de aquéllas se repiten *mientras* existen ciertas condiciones o *hasta* que éstas se presentan.

La sintaxis de estas estructuras es la siguiente:

MIENTRAS < condición >

< acciones >

FINMIENTRAS

b) Reportes de soluciones.- estos reportes permiten imprimir la guía de un examen contestado con las respuestas correctas. Es una herramienta muy útil para la calificación de los exámenes impresos.

Diseño de las salidas.

Una de las características más importantes para los usuarios de un sistema de información son las salidas que éste produce, y es necesario que se sigan ciertos principios para su diseño, ya que si la interfaz no tiene calidad es confusa o demasiado compleja, los usuarios no tendrán un medio efectivo para la comunicación con el sistema y por lo tanto al no acceder a la funcionalidad del sistema les puede parecer tan poco necesario que evitarán usarlo.

Reportes de calificaciones.- imprime la base de datos seleccionada que contiene un informe de calificaciones previamente elaborado.

Idealmente alguien que nunca ha visto una aplicación debería ser capaz de sentarse, mirar la pantalla y comprender el sistema. Tal facilidad de uso no siempre es posible, pero con un poco de imaginación y el seguimiento de buenos principios en el diseño será más posible lograrlo.

Principios para el diseño de la interfaz.

Diseño de la salida impresa.

Un informe o reporte es un compendio o resumen de datos que se encuentran almacenados en una base de datos que se muestra según convenga en cada caso en un formato determinado.

El sistema debe ser intuitivo. Este sistema que debe presentar toda la información que el usuario necesita saber de una manera que la entienda inmediatamente, sin tener que aprender algo nuevo. Para lograr esto es conveniente tomar en cuenta todo los estándares existentes con los cuales el usuario está familiarizado.

Como ya se menciona en el capítulo del diseño físico, FoxPro posee un generador de informes que puede ahorrar tiempo de programación y ser de mucha utilidad, por tal motivo se obtuvo provecho de esta característica del lenguaje de programación seleccionado.

En el esfuerzo por realizar una interfaz muy sencilla para los usuarios principiantes no hay que olvidar a los avanzados que generalmente prefieren usar el teclado que el ratón. Por lo cual es conveniente presentar una interfaz dual de forma que se pueda usar tanto el ratón como los métodos abreviados del teclado para interactuar con el sistema.

La determinación de los reportes que estarán disponibles se hizo en base al estudio de los requerimientos de los usuarios, tanto de operación como de mantenimiento del sistema, de tal manera que los reportes provistos son:

- Consistencia.

Todas las ventanas, campos y controles debe comportarse de las misma manera, tanto a) Reportes de reactivos.- esta opción permite realizar la impresión de un guión previamente elaborado. Este tipo de reportes están diseñados especialmente para permitir tener el control de los reactivos que hayan sido dados de alta en el sistema.

b) Reportes de soluciones.- estos reportes permiten imprimir la guía de un examen impreso, es decir, este mismo pero contestado con las respuestas correctas. Es una herramienta muy útil para la calificación de los exámenes impresos.

c) Reportes de grupos.- permite imprimir las claves de los grupos y su descripción, que hasta el momento han sido dadas de alta; ayuda a mantener el control de los grupos que clasifican a los reactivos.

d) Reportes de calificaciones.- imprime la base de datos seleccionada que contiene un informe de calificaciones previamente elaborado.

- Indulgencia.

Todo el mundo comete errores, la interfaz diseñada debe considerarlo y ser tan indulgente como sea posible para que los errores ocurran. La indulgencia abarca dos niveles: prevención y cura.

Diseño de la salida en pantalla.

Principios para el diseño de la interfaz.

Las medidas preventivas toman lugar antes de que las acciones potencialmente

- El sistema debe ser intuitivo.

Esto significa que debe presentar toda la información que el usuario necesita saber de una manera que la entienda inmediatamente, sin tener que aprender algo nuevo. Para lograr esto es conveniente tomar en cuenta todo los estándares existentes con los cuales el usuario está familiarizado. Ctrl + Z), la tecla de escape (Esc) o la opción Cancelar entran en acción.

En el esfuerzo por realizar una interfaz muy sencilla para los usuarios principiantes no hay que olvidar a los avanzados que generalmente prefieren usar el teclado que el ratón. Por lo cual es conveniente presentar una interfaz dual de forma que se pueda usar tanto el ratón como los métodos abreviados del teclado para interactuar con el sistema.

- Consistencia.

Todas las ventanas, campos y controles debe comportarse de las misma manera, tanto dentro del sistema como con los estándares externos aceptados, lo cual reducirá la cantidad de información que el usuario nuevo tiene que aprender y por lo tanto la aplicación será más fácil de usar.

- Retroalimentación.

Es importante mantener al usuario informado de lo que esta pasando en todo momento, especialmente cuando es una operación que va a durar algún tiempo en proceso.

- Control del usuario.

Siempre es conveniente dar a los usuarios cierto control del ambiente en el cual esté trabajando, dejarlos decidir el tamaño y posición de las ventanas o cuáles mantener abiertas. Esto los hará sentir que el sistema realmente se adapta a sus necesidades y no que ellos son los que se adaptan a las necesidades del sistema.

- Indulgencia.

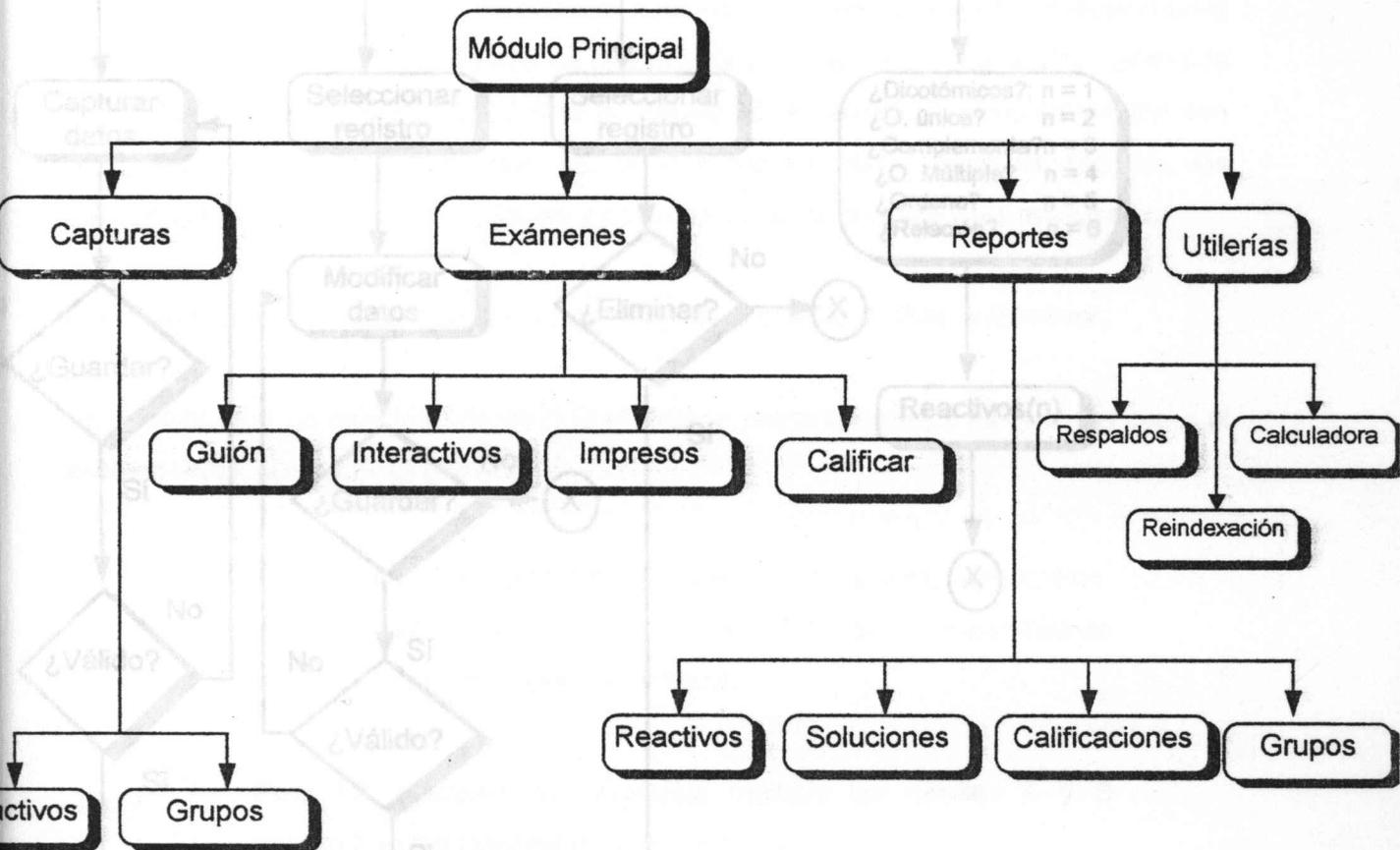
Todo el mundo comete errores, la interfaz diseñada debe considerarlo y ser tan indulgente como sea posible cuando los errores ocurran. La indulgencia abarca dos niveles: prevención y cura.

Las medidas preventivas toman lugar antes de que las acciones potencialmente destructivas ocurran. Una buena medida es no proporcionar teclas rápidas para este tipo de acciones o bien requerir la confirmación cuando estas operaciones se realizan.

La cura ocurre después del hecho, una vez que se hizo algo que no debió haber hecho. Aquí el comando UNDO (Ctrl + Z), la tecla de escape (Esc) o la opción Cancelar entran en acción.

Módulos del sistema

Dentro de la etapa del diseño lógico es necesario especificar las operaciones principales que realizará el sistema, por esta razón a continuación se presenta de cada módulo su diagrama de flujo, descripción del manejo externo de cada módulo, y la forma en que estos funcionan.



Módulo de Reactivos.

Descripción.

La captura de los reactivos es un proceso fundamental del sistema y su operación es muy sencilla. Tiene la característica de uniformidad de presentación de las pantallas para ser operado por usuarios con varias particularidades.

Las ideas básicas de este módulo se concentran en los elementos que se determinaron en la etapa de diseño como los componentes de la estructura del reactivo. Una vez comprendida la estructura de los reactivos, se diseñaron los formularios como una lista de reactivos. Los reactivos se definen para estar vacíos a excepción de los reactivos de tipo relación, los valores absoluto y relativo, los minutos y como es obvio, la estructura de los reactivos.

El primer elemento de diseño para los reactivos es el tipo de pregunta, Editar, Eliminar.

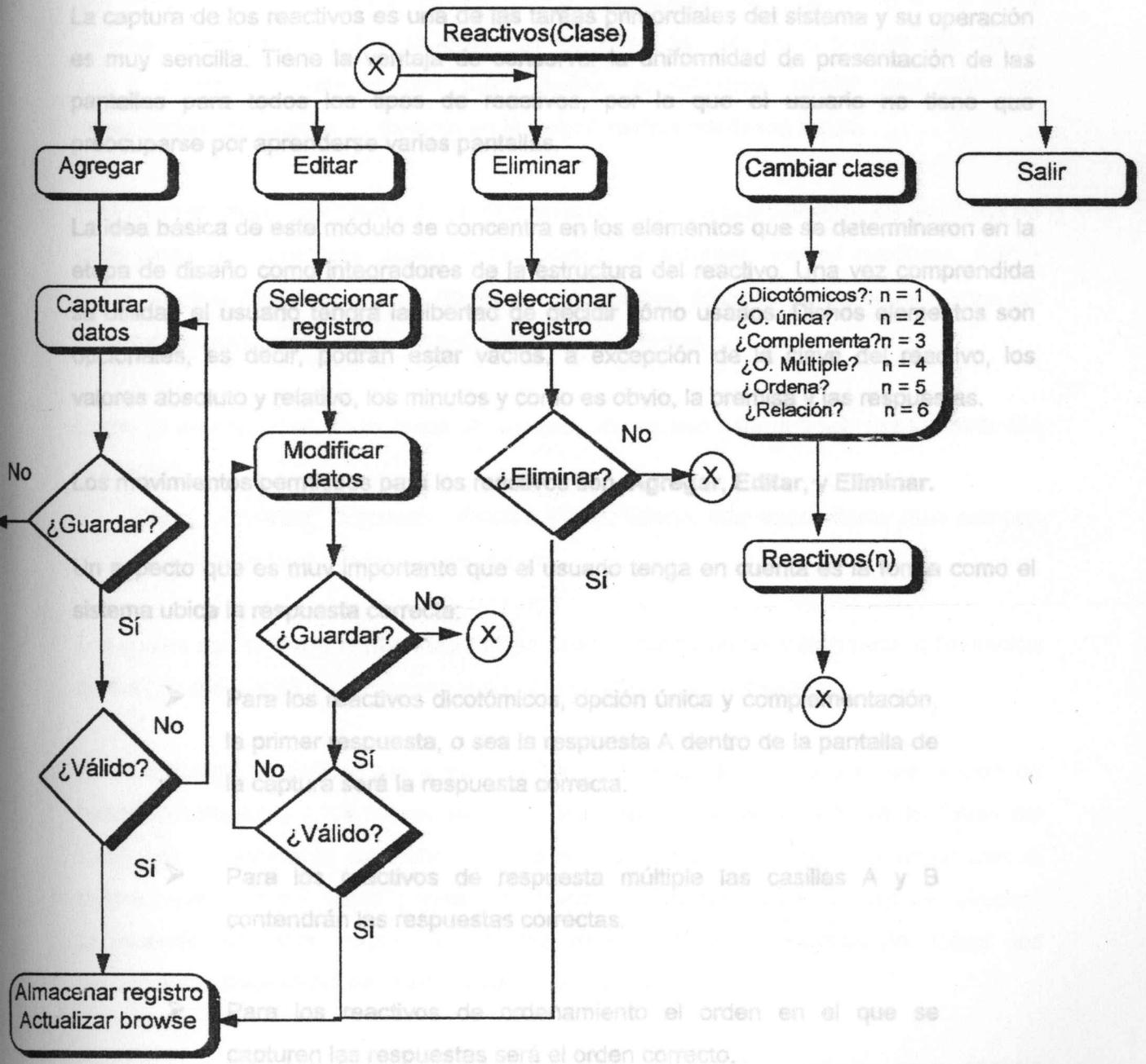
Para los reactivos que es importante que el usuario tenga en cuenta los reactivos como el tipo de pregunta, se ubica la respuesta correcta.

Para los reactivos dicotómicos, opción única y complementación, la primera respuesta, o sea la respuesta A dentro de la pantalla de captura, será la respuesta correcta.

Para los reactivos de respuesta múltiple las casillas A y B contendrán las respuestas correctas.

Para los reactivos de ordenamiento el orden en el que se capturen las respuestas será el orden correcto.

Para los reactivos de relación las parejas correctas serán aquellas que estén como parejas en la captura.



Reactivos.

Descripción.

La captura de los reactivos es una de las tareas primordiales del sistema y su operación es muy sencilla. Tiene la ventaja de conservar la uniformidad de presentación de las pantallas para todos los tipos de reactivos, por lo que el usuario no tiene que preocuparse por aprenderse varias pantallas.

La idea básica de este módulo se concentra en los elementos que se determinaron en la etapa de diseño como integradores de la estructura del reactivo. Una vez comprendida su utilidad el usuario tendrá la libertad de decidir cómo usarlos. Dichos elementos son opcionales, es decir, podrán estar vacíos, a excepción de la clave del reactivo, los valores absoluto y relativo, los minutos y como es obvio, la premisa y las respuestas.

Los movimientos permitidos para los reactivos son **Agregar, Editar, y Eliminar.**

Un aspecto que es muy importante que el usuario tenga en cuenta es la forma como el sistema ubica la respuesta correcta:

- Para los reactivos dicotómicos, opción única y complementación, la primer respuesta, o sea la respuesta A dentro de la pantalla de la captura será la respuesta correcta.
- Para los reactivos de respuesta múltiple las casillas A y B contendrán las respuestas correctas.
- Para los reactivos de ordenamiento el orden en el que se capturen las respuestas será el orden correcto.
- Para los reactivos de relación las parejas correctas serán aquellas que estén como parejas en la captura.

Funcionamiento.

La primera actividad necesaria para diseñar un cuestionario es la definición de los reactivos, la cual se realiza dentro del módulo de capturas del SIGEE. Dado que en la mayoría de los casos ésta es la tarea más laboriosa, fue necesario diseñar una interfaz que a la vez que fuera sencilla, fuera también lo suficiente versátil para cubrir las necesidades del usuario, evitándole en la mayor medida posible la fatiga.

Este módulo trabaja por medio de un examinador o browse que presenta los reactivos de cada clase por separado, dando la posibilidad de **Agregar** nuevos reactivos, **Editar** algún reactivo existente o bien **Eliminar** un reactivo además de conmutar entre las clases de reactivos por medio de combinaciones de teclas.

Como ya se mencionó en la etapa de análisis, un reactivo está definido por los atributos siguientes:

- ☛ Clave, Profesor, Escuela, Materia, Tema, Grado, dos clases libres (dos campos sin etiqueta para el uso que se desee)

Los cuales son elementos que serán útiles como criterios de selección para la formación de los guiones para los cuestionarios tanto interactivos como impresos.

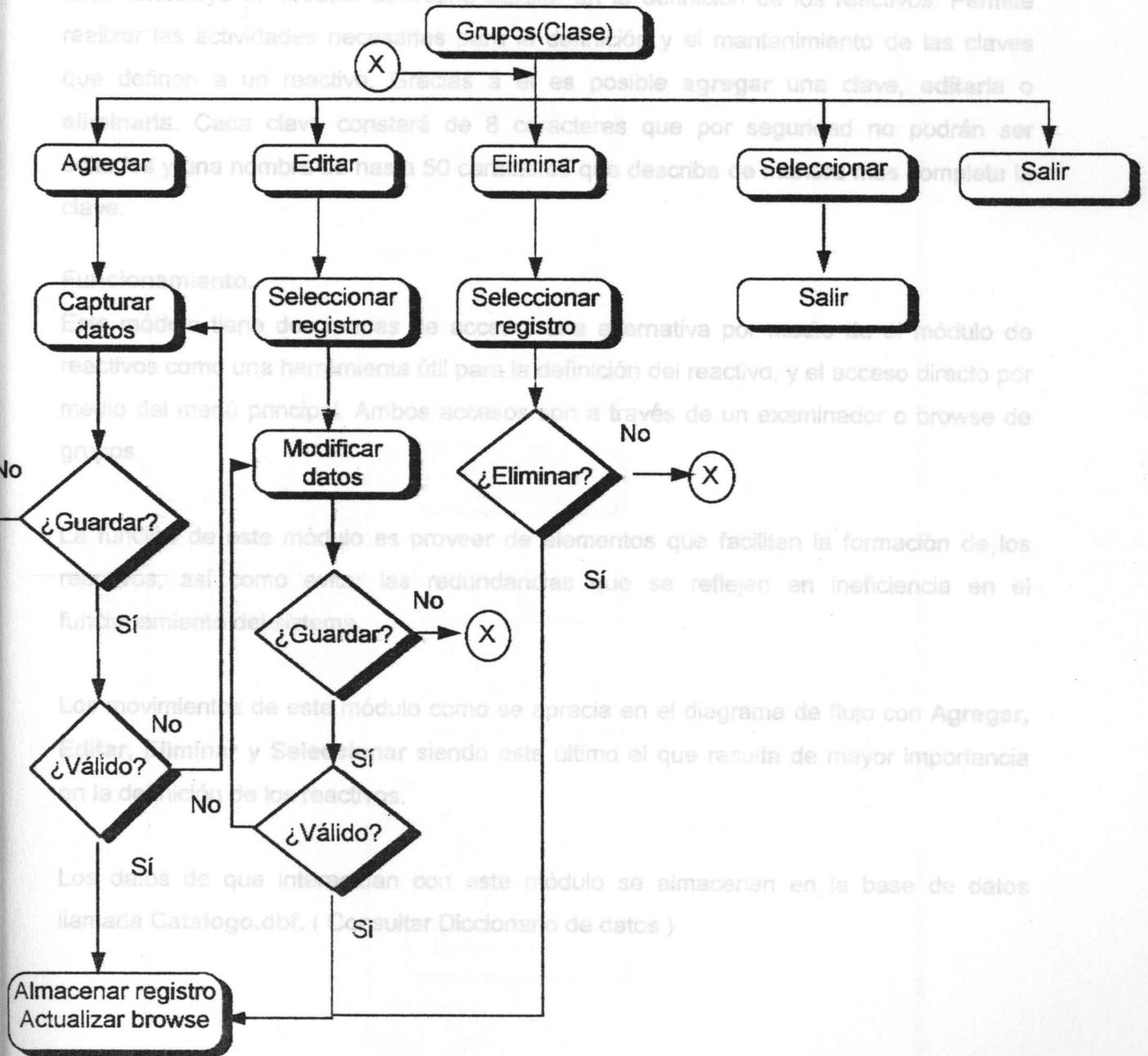
Estos atributos se definen por medio de claves de 8 caracteres, y una descripción de hasta 50 caracteres (los cuales se encuentran almacenados en la Base de Datos del Catálogo). Al momento de definir un reactivo es posible también interactuar con el módulo que controla estos grupos de claves (ver diagrama de Módulo de Grupos) permitiendo de esta forma la máxima flexibilidad y proporcionando todas las herramientas necesarias para la definición del reactivo.

Cabe señalar que una ficha de definición de reactivo podrá tener campos vacíos pero no claves que no pertenezcan al Catálogo.

Módulo de Grupos.

Descripción.

Este constituye un módulo accesorio auxiliar en la definición de los reactivos. Permite realizar las actividades necesarias para el mantenimiento de las claves que definen a un reactivo. Es posible agregar una clave, editarla o eliminarla. Cada clave constará de 8 caracteres que por seguridad no podrán ser una combinación de letras y números de 50 caracteres que describa el grupo.



Grupos. Módulo de Formación del guión.

Descripción.

Este constituye un módulo accesorio auxiliar en la definición de los reactivos. Permite realizar las actividades necesarias para la definición y el mantenimiento de las claves que definen a un reactivo. Gracias a él es posible **agregar** una clave, **editarla** o **eliminarla**. Cada clave constará de 8 caracteres que por seguridad no podrán ser editados y una nombre de hasta 50 caracteres que describa de manera más completa la clave.

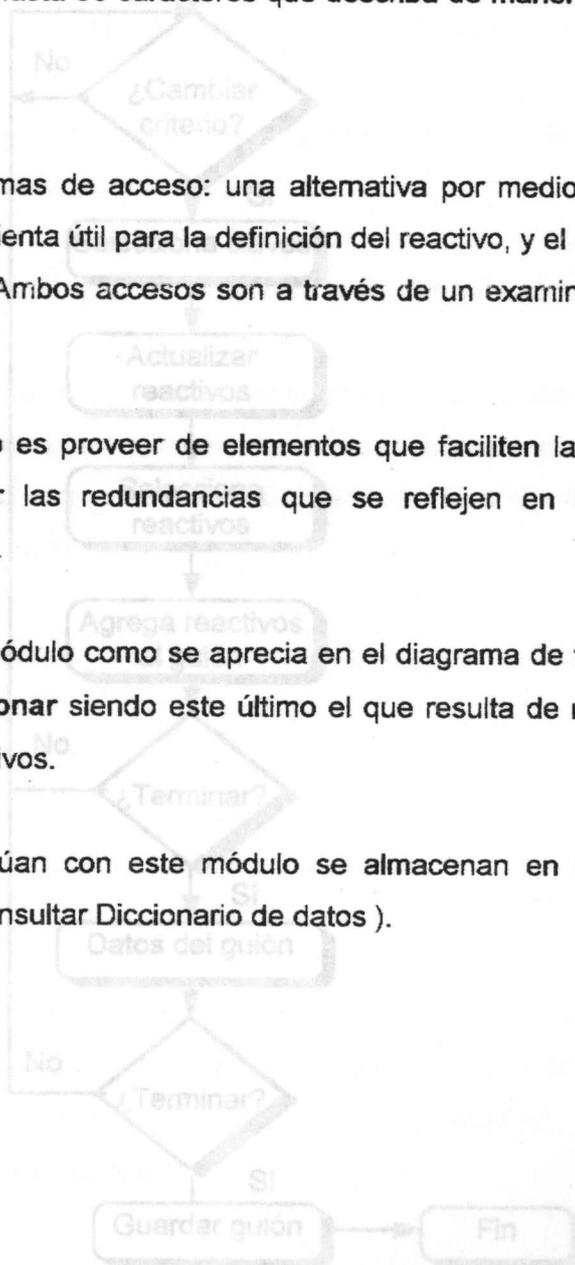
Funcionamiento.

Este módulo tiene dos formas de acceso: una alternativa por medio de el módulo de reactivos como una herramienta útil para la definición del reactivo, y el acceso directo por medio del menú principal. Ambos accesos son a través de un examinador o browse de grupos.

La función de este módulo es proveer de elementos que faciliten la formación de los reactivos, así como evitar las redundancias que se reflejen en ineficiencia en el funcionamiento del sistema.

Los movimientos de este módulo como se aprecia en el diagrama de flujo con **Agregar**, **Editar**, **Eliminar** y **Seleccionar** siendo este último el que resulta de mayor importancia en la definición de los reactivos.

Los datos de que interactúan con este módulo se almacenan en la base de datos llamada **Catalogo.dbf**. (Consultar Diccionario de datos).



Módulo de Formación del guión.

Descripción.

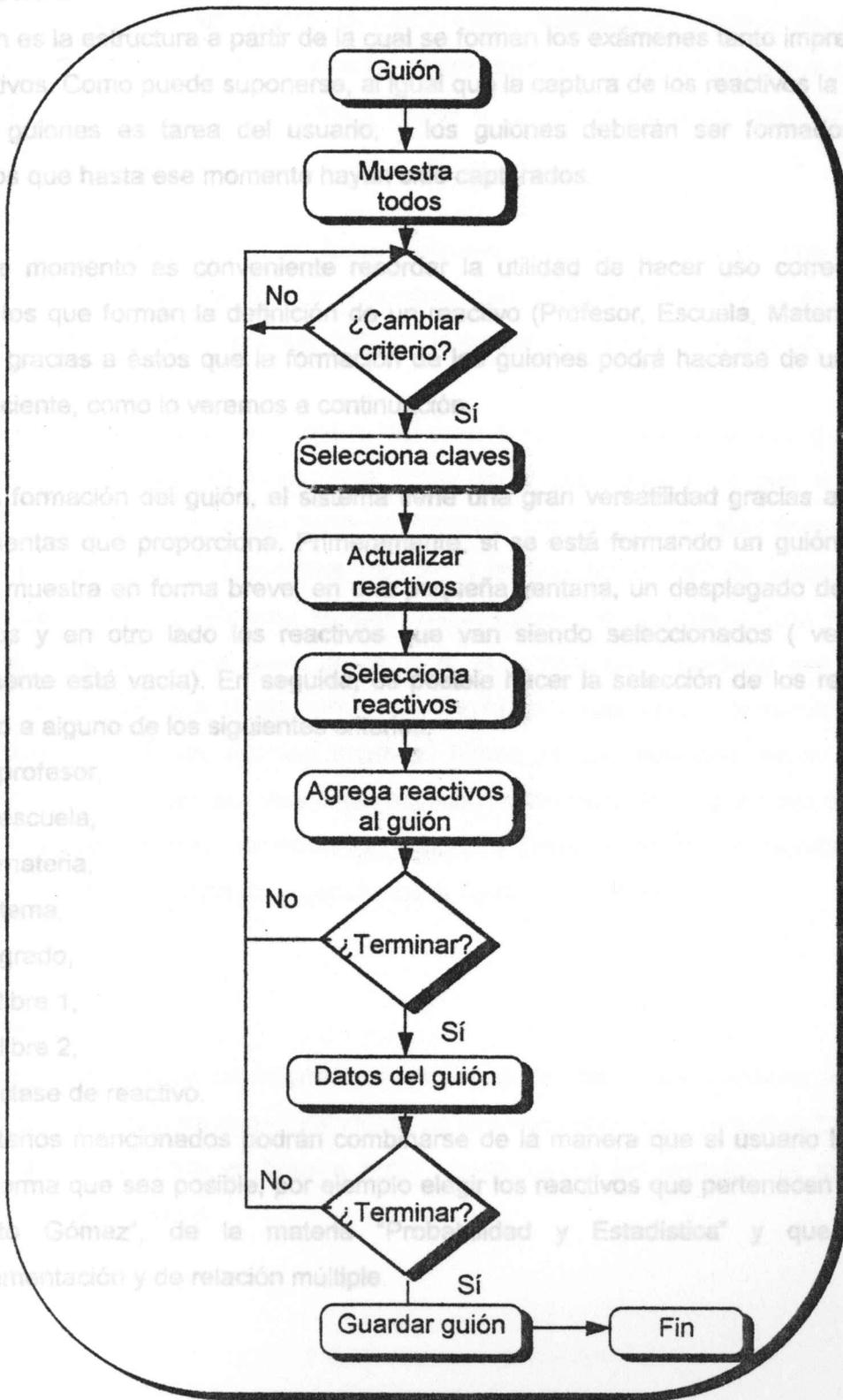
El guión es la estructura a partir de la cual se forman los exámenes tanto impresos como interactivos. Como puede suponerse, la captura de los reactivos y la formación de los guiones es tarea del usuario, los guiones deberán ser formados con los reactivos que hasta ese momento han sido capturados.

En este momento es conveniente recordar la utilidad de hacer uso correcto de los elementos que forman la definición de un guión (Profesor, Escuela, Materia, etc.), ya que es gracias a éstos que la formación de los guiones podrá hacerse de una manera más eficiente, como lo veremos a continuación.

Para la formación del guión, el sistema ofrece una gran versatilidad gracias a las varias herramientas que proporciona. Cuando el usuario está formando un guión nuevo, el SIGEE muestra en forma breve en una pequeña ventana, un desplegado de todos los reactivos y en otro lado los reactivos que van siendo seleccionados (ventana que inicialmente está vacía). En seguida se explicará la selección de los reactivos de acuerdo a alguno de los siguientes criterios:

- a) Por profesor.
- b) Por escuela.
- c) Por materia.
- d) Por tema.
- e) Por grado.
- f) Por libro 1.
- g) Por libro 2.
- h) Por clase de reactivo.

Los criterios mencionados podrán combinarse de la manera que el usuario lo requiera, de tal forma que sea posible elegir los reactivos que pertenecen al maestro "Roberto Gómez", de la materia "Probabilidad y Estadística" y que sean de complementación y de relación múltiple.



Formación del guión.

Descripción.

El guión es la estructura a partir de la cual se forman los exámenes tanto impresos como interactivos. Como puede suponerse, al igual que la captura de los reactivos la formación de los guiones es tarea del usuario; y los guiones deberán ser formados con los reactivos que hasta ese momento hayan sido capturados.

En este momento es conveniente recordar la utilidad de hacer uso correcto de los elementos que forman la definición de un reactivo (Profesor, Escuela, Materia etc.), ya que es gracias a éstos que la formación de los guiones podrá hacerse de una manera más eficiente, como lo veremos a continuación.

Para la formación del guión, el sistema tiene una gran versatilidad gracias a las varias herramientas que proporciona. Primeramente, si se está formando un guión nuevo, el SIGEE muestra en forma breve, en una pequeña ventana, un desplegado de todos los reactivos y en otro lado los reactivos que van siendo seleccionados (ventana que inicialmente está vacía). En seguida, es posible hacer la selección de los reactivos de acuerdo a alguno de los siguientes criterios:

- a) Por profesor,
- b) Por escuela,
- c) Por materia,
- d) Por tema,
- e) Por grado,
- f) Por libre 1,
- g) Por libre 2,
- h) Por clase de reactivo.

Los criterios mencionados podrán combinarse de la manera que el usuario lo requiera, de tal forma que sea posible, por ejemplo elegir los reactivos que pertenecen al maestro "Roberto Gómez", de la materia "Probabilidad y Estadística" y que sean de complementación y de relación múltiple.

Los criterios que sean ignorados indicará que se desean todos campos, y será posible así mismo incluir o excluir los campos que estén vacíos.

asigna automáticamente el sistema. Ejemplo:

Una vez seleccionados los criterios la ventana que despliega los reactivos se actualizará de acuerdo a la selección, con lo cual, se podrá identificar a los reactivos que clasificaron dentro de los criterios seleccionados y decidir si realmente se desea hacer la selección de estos reactivos o no.

Esta selección podrá hacerse en forma colectiva o individual, es decir, que se pueden seleccionar todos los reactivos que concordaron con los criterios, o bien hacer una depuración manual de los mismos decidiendo cuáles seleccionar y cuáles no.

Cuando finalmente se tiene el guión deseado entonces será necesario guardarlo en forma de archivo para posteriormente poder hacer uso de él. Para que el guión sea guardado es necesario proporcionar un nombre con el cual lo identificaremos, este nombre podrá ser de 5 caracteres máximo y el sistema le dará la extensión .DBG, además del nombre el sistema también da la opción de guardar alguna información adicional, como lo es una breve descripción, el nombre de la persona que elaboró el sistema, la fecha en que se elaboró, y tres líneas de encabezado que servirán como tal en la primera hoja del examen impreso. Como ya se mencionó estos datos son opcionales, pero pueden ser muy útiles especialmente después de que haya transcurrido bastante tiempo desde la formación de un guión y desee recordar el contenido del guión por medio de la información que usted mismo haya proporcionado.

formaran el guión definitivo.

Su estructura es la siguiente:

Clase C(1). Clave C(8).

Funcionamiento.

Un guión, como ya se mencionó, es una base de datos que contiene los campos mínimos necesarios para la generación de un examen impreso o interactivo. La selección de los reactivos se hace en base a algún criterio (según los atributos del reactivo), en forma individual o ambos.

El motivo por el cual el nombre del guión es de 5 caracteres máximo es para dar la opción de usar el mismo nombre para los exámenes más un número consecutivo que asigna automáticamente el sistema. Ejemplo:

Guión: MATE.DBG
Examen: MATE001.DBE
MATE002.DBE, etc.

Dicho nombre es proporcionado por el diseñador del guión, así como alguna otra información que ya se mencionó en la descripción de este módulo

Esta rutina tiene además la posibilidad de editar un guión previamente elaborado que será mandado como parámetro si es que éste es el caso.

Estructuras de datos.

En la formación del guión intervienen las siguientes bases de datos.

<Guion>.DBG: Como ya se mencionó en secciones anteriores es la base de datos a partir de la cual se generan tanto el examen impreso como interactivo. (Consultar sección de diccionario de datos)

Reactivos Base de datos efímera o *cursor* que contiene los reactivos que responden a los criterios seleccionados o bien los reactivos del guión que se están editando.

Su estructura es la siguiente:

Clase C(1), Clave C(8)

CurGuion Base de datos efímera o *cursor* que contiene los reactivos que formarán el guión definitivo.

Su estructura es la siguiente:

Clase C(1), Clave C(8).

Una vez terminado el proceso de formación del guión el cursor **CurGuion** se convierte en **<Guion>.DBG** que será la base de datos que conservará permanentemente la definición de dicho guión.

Examen: Módulo del examen impreso.

Descripción.

El primer paso consiste en seleccionar el guión (diseñado previamente) que contiene los reactivos deseados, para lo cual se presenta una lista con todos los guiones generados hasta el momento, los cuales se conocen por la extensión .DBG Para identificar el contenido de los guiones se hace uso de los siguientes procedimientos:

1. Por medio del nombre, ya que este es dado por los usuarios
2. Consultando la información del archivo, en la cual se guarda una breve descripción del guión, quién lo diseñó y un encabezado de tres líneas.

Si no existiere un guión con las características deseadas, el SIGEE proporciona la facilidad de poder generar un nuevo guión y así poder ser seleccionado.

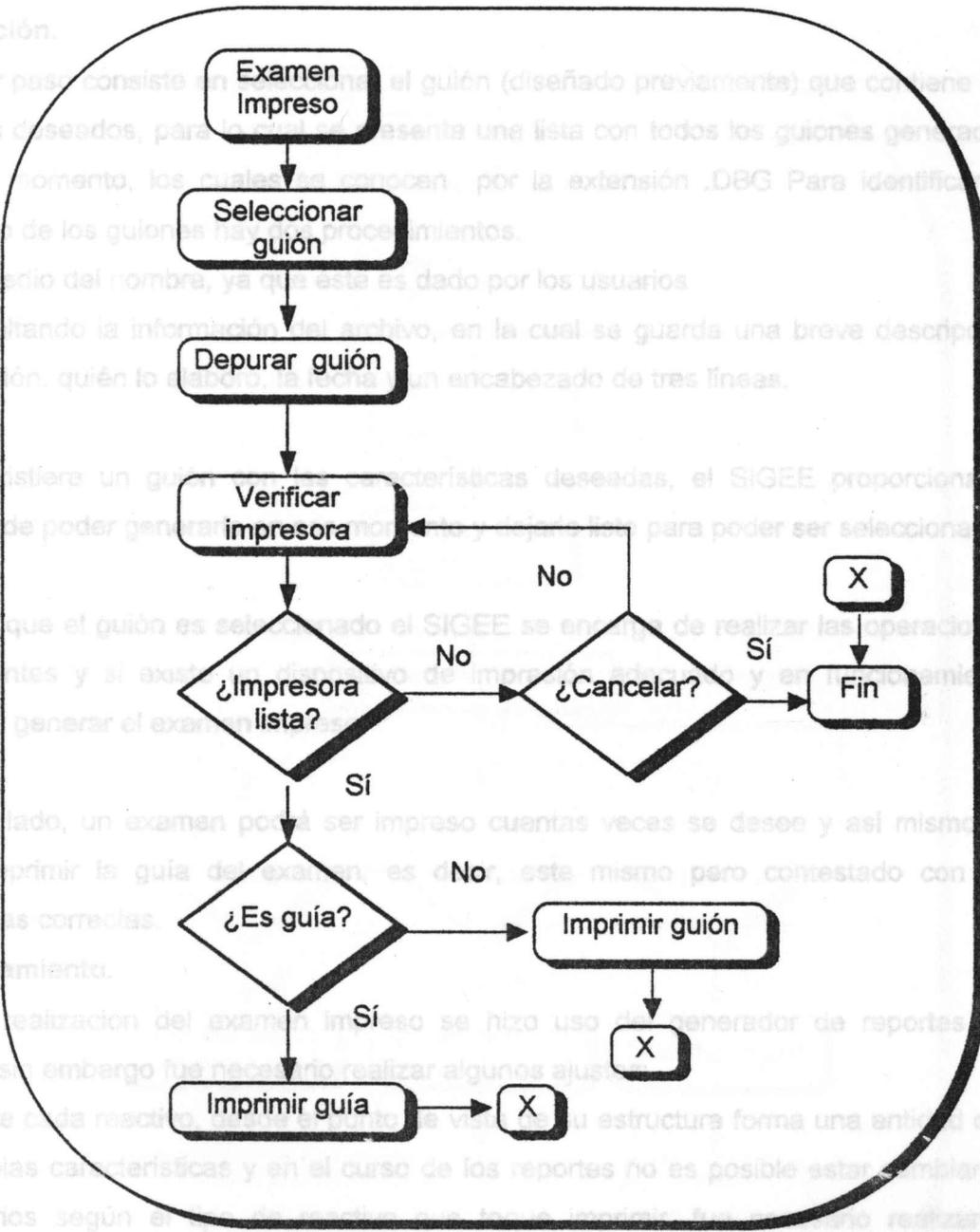
Una vez que el guión es seleccionado el SIGEE se encarga de realizar las operaciones convenientes y al existir el archivo de impresión se genera el examen y en caso contrario entonces generar el examen en un archivo de texto.

Por otro lado, un examen puede ser impreso cuantas veces se desee y así mismo se podrá imprimir la guía del examen, es decir, este mismo pero contestado con las respuestas correctas.

Funcionamiento.

Para la realización del examen impreso se hizo uso del generador de reportes del FoxPro, sin embargo fue necesario realizar algunos ajustes.

Dado que cada reactivo tiene una estructura forma una entidad con sus propias características y en el curso de los reportes no es posible estar cambiando de criterios según el tipo de reactivos que se va a imprimir, fue necesario realizar la conversión de todas las clases de reactivos y construir una unidad que fuera tratada homogéneamente por los reportes. Esto se realizó por medio de funciones especializadas en cada tipo de reactivo que construyeron para el reporte, una variable que contiene a todo el reactivo, y de esta manera pudieran ser tratadas de manera distinta.



Examen impreso. exámenes interactivos.

Descripción.

El primer paso consiste en seleccionar el guión (diseñado previamente) que contiene los reactivos deseados, para lo cual se presenta una lista con todos los guiones generados hasta el momento, los cuales se conocen por la extensión .DBG Para identificar el contenido de los guiones hay dos procedimientos.

1. Por medio del nombre, ya que éste es dado por los usuarios
2. Consultando la información del archivo, en la cual se guarda una breve descripción del guión. quién lo elaboró, la fecha y un encabezado de tres líneas.

Si no existiera un guión con las características deseadas, el SIGEE proporciona la facilidad de poder generarlo en ese momento y dejarlo listo para poder ser seleccionado.

Una vez que el guión es seleccionado el SIGEE se encarga de realizar las operaciones convenientes y si existe un dispositivo de impresión adecuado y en funcionamiento entonces generar el examen impreso.

Por otro lado, un examen podrá ser impreso cuantas veces se desee y así mismo se podrá imprimir la guía del examen, es decir, este mismo pero contestado con las respuestas correctas.

Funcionamiento.

Para la realización del examen impreso se hizo uso del generador de reportes del FoxPro, sin embargo fue necesario realizar algunos ajustes:

Dado que cada reactivo, desde el punto de vista de su estructura forma una entidad con sus propias características y en el curso de los reportes no es posible estar cambiando de criterios según el tipo de reactivo que toque imprimir, fue necesario realizar la conversión de todas las clases de reactivos y construir una unidad que fuera tratada homogéneamente por los reportes. Esto se realizó por medio de funciones especializadas en cada tipo de reactivo que construyeron para el reporte, una variable que contiene a todo el reactivo, y de esta manera pudieran ser tratadas de manera indistinta.

Exámenes Interactivos

Módulo de exámenes interactivos.

Son dos los elementos necesarios para la aplicación de un examen interactivo:

1. El guión, del cual se obtienen los reactivos y las opciones
2. El examen, en el cual se guardan los datos del alumno, el orden en el que se presentaron las respuestas de los reactivos, las calificaciones de las contestaciones del alumno. (Ver diccionario de datos EXA00000)

Para aplicar el examen es necesario crear los elementos y proporcionar los datos del alumno para que el examen pueda ser aplicado.

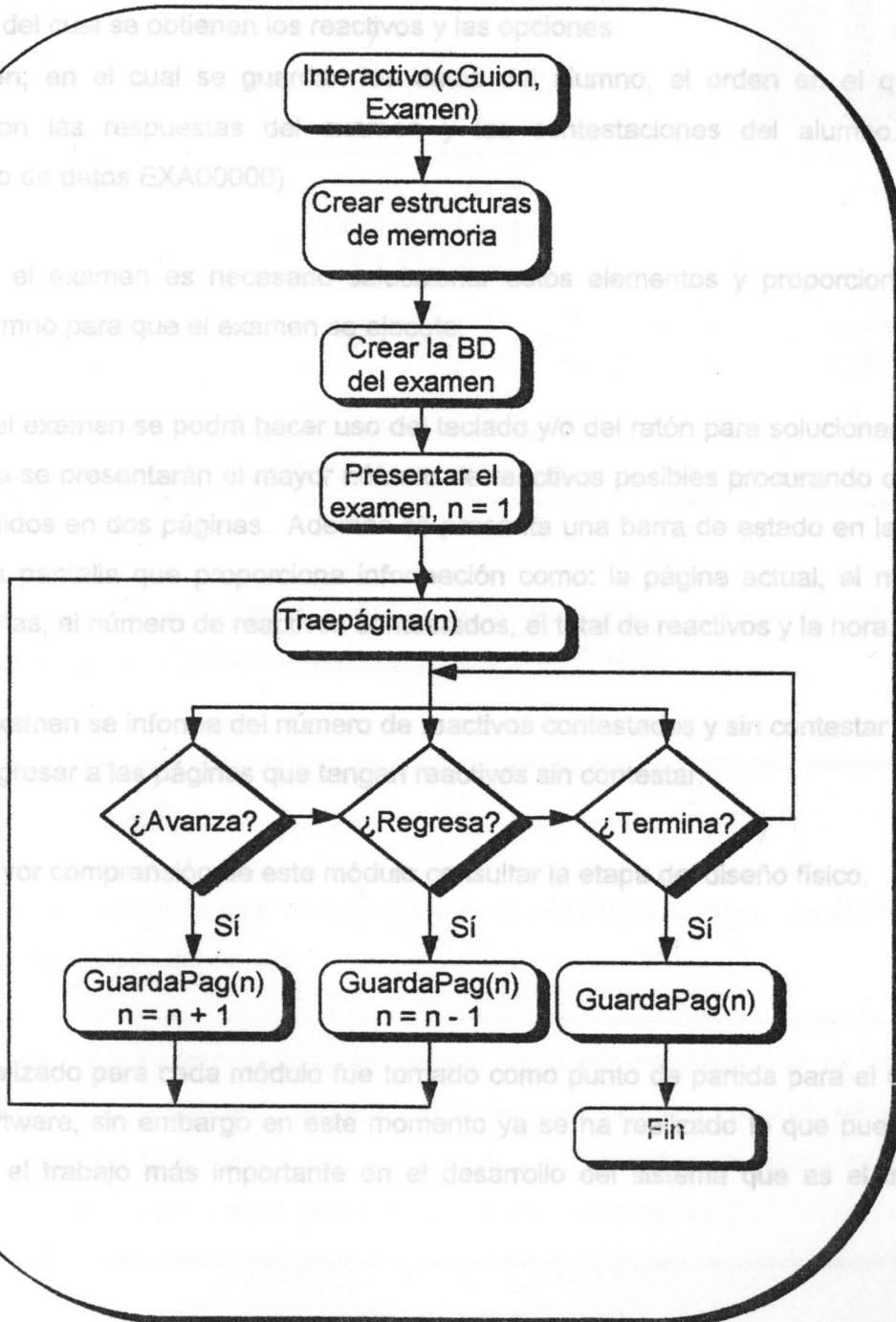
Una vez en el examen se podrá hacer uso de teclado y/o del ratón para solucionar. En cada pantalla se presentarán el mayor número de reactivos posibles procurando que no queden divididos en dos páginas. Además se tendrá una barra de estado en la parte inferior de la pantalla que presentará información como: la página actual, el número total de páginas, el número de reactivos contestados, el total de reactivos y la hora.

Al salir del examen se informará del número de reactivos contestados y sin contestar con la opción de regresar a las páginas que tengan reactivos sin contestar.

Para una mejor comprensión de este módulo consultar la etapa de diseño físico.

NOTA:

El diseño realizado para cada módulo fue tomado como punto de partida para el diseño físico del software, sin embargo en este momento ya se ha realizado el trabajo que puede ser considerado el trabajo más importante en el desarrollo del sistema que es el trabajo creativo.



Exámenes Interactivos. DISEÑO FISICO

Son dos los elementos necesarios para la aplicación de un examen interactivo:

1. El **guión**, del cual se obtienen los reactivos y las opciones
2. El **examen**, en el cual se guardan los datos del alumno, el orden en el que se presentaron las respuestas del examen y las contestaciones del alumno. (Ver diccionario de datos EXA00000)

Para aplicar el examen es necesario seleccionar estos elementos y proporcionar los datos del alumno para que el examen se ejecute.

Una vez en el examen se podrá hacer uso del teclado y/o del ratón para solucionarlo. En cada pantalla se presentarán el mayor número de reactivos posibles procurando que no queden divididos en dos páginas.. Además se presenta una barra de estado en la parte inferior de la pantalla que proporciona información como: la página actual, el número total de páginas, el número de reactivos contestados, el total de reactivos y la hora.

Al salir del examen se informa del número de reactivos contestados y sin contestar con la opción de regresar a las páginas que tengan reactivos sin contestar.

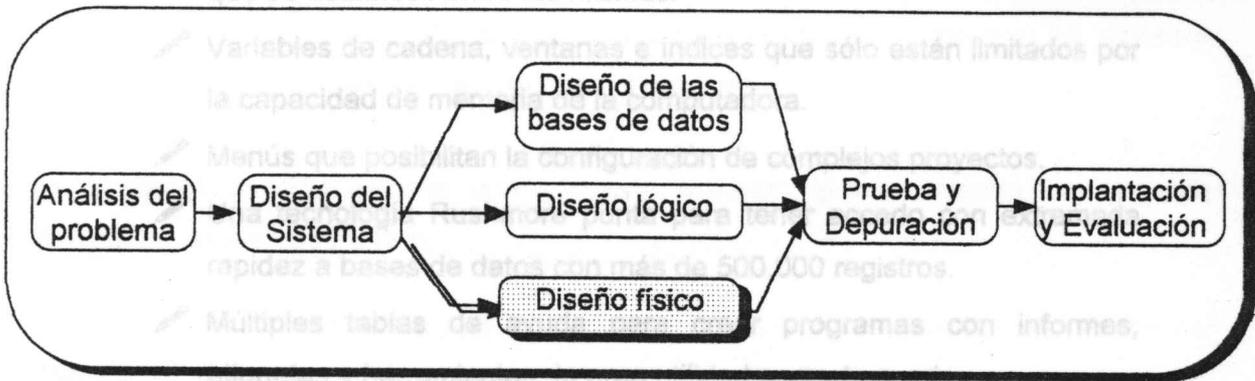
Para una mayor comprensión de este módulo consultar la etapa del diseño físico.

El lenguaje de programación que se eligió para el desarrollo del SIGEE, fue el FoxPro y a continuación se detalla el por qué.

NOTA

El diseño realizado para cada módulo fue tomado como punto de partida para el diseño físico del software, sin embargo en este momento ya se ha realizado lo que puede ser considerado el trabajo más importante en el desarrollo del sistema que es el trabajo creativo.

Son muchos los motivos que contribuyen a la universalidad de FoxPro entre los que destacamos los siguientes:



El diseño físico también conocido como desarrollo del software es la etapa de conversión de las características diseñadas en un programa de computadora.

Uno de los aspectos más importantes en este momento es la adecuada selección del lenguaje de programación.

El lenguaje de programación que se eligió para el desarrollo del SIGEE, fue el FoxPro y a continuación se detalla el por qué.

FoxPro es un poderoso sistema de gestión de bases de datos que se ha hecho famoso por su moderna interfaz y sus eficientes métodos basados en la capacidad de diálogos entre ventanas que permiten construir proyectos muy complejos y desarrollar menús, ventanas, informes, etiquetas etc.

Son muchos los motivos que contribuyen a la universalidad de FoxPro entre los que destacamos los siguientes:

1. **Propiedades:** Solo precisa de 280 K de memoria. FoxPro en su versión extendida de 32 bits, utiliza con éxito sus denominados índices compactos que ahorran memoria y tiempo.
2. **Mejoras:**
 - Su software es también válido para un entorno Windows.
 - Se destaca su gran eficiencia en el manejo de ventanas y en diálogo que se establece entre las mismas.
 - Variables de cadena, ventanas e índices que sólo están limitados por la capacidad de memoria de la computadora.
 - Menús que posibilitan la configuración de complejos proyectos.
 - Una tecnología Rushmore punta para tener acceso con extremada rapidez a bases de datos con más de 500,000 registros.
 - Múltiples tablas de ayuda para crear programas con informes, etiquetas y herramientas de gran utilidad para el usuario.
 - Índices compactos con filtraje opcional y rapidez en la búsqueda de los datos.
 - Originales campos memo para almacenar texto en las bases de datos.
 - Utilísimos y poderosos generadores de informes y etiquetas.
 - Posibilidad de hacer compatible el lenguaje dBase III y dBase IV con el compilador FoxPro.
3. **Mejoras:**
 - Sofisticados procedimientos para la introducción de datos desde la consola.

Herramientas para el desarrollo de sistemas.

Una herramienta es cualquier dispositivo que, cuando se emplea en forma apropiada, mejora el desempeño de una tarea. Las herramientas en el desarrollo de sistemas son esenciales ya que mejoran la forma en que ocurre el desarrollo y tienen influencia sobre la calidad del resultado final.

Los resultados extienden en tres formas la capacidad del desarrollador:

1. Proporcionan el potencial para mejorar la productividad del desarrollador al disminuir el tiempo en que se desarrollan las aplicaciones gracias a que facilitan sobretodo el desarrollo de procesos largos y tediosos.
2. Mejoran la eficiencia. Las herramientas sugieren procedimientos que conducen al empleo de procesos más eficientes. En otras palabras, ayudan a realizar los procesos de la manera correcta.
3. Mejoran la calidad del producto final. Cuando las herramientas mejoran los procesos, por lo general esto también ocurre con los resultados.

La siguiente sección expone una pequeña revisión de herramientas con especial dedicación al entorno FoxPro.

Herramientas informáticas de apoyo a la ingeniería de software. (CASE).

Las herramientas CASE (Computer Aided Software Engineering) contribuyen a automatizar el diseño y desarrollo de programas. Parece lógico que los programadores aprovechen la potencia de las computadoras para simplificar su trabajo, tal como lo hacen en favor de muchas otras personas. Estas herramientas suelen incluir las prestaciones siguientes:

Gráficos.-

Una biblioteca de símbolos para confeccionar diagramas de flujo de datos, diagramas de descomposición, diagramas de acción u otras técnicas semejantes.

Metodologías.-

Soporte para, al menos una metodología particular de diseño, como el diseño estructurado.

Diccionario de datos.-

Un medio para definir elementos de datos y las estructuras de archivos, así como su utilización en las pantallas e informes que compondrá el sistema.

Diseño de pantallas e informes. propias de generación de código, a través de CREATE
Posibilidad de definir rápidamente la apariencia de pantallas e informes, tanto para el desarrollo de prototipos como para generación automática de código. propio código fuente (modificable).

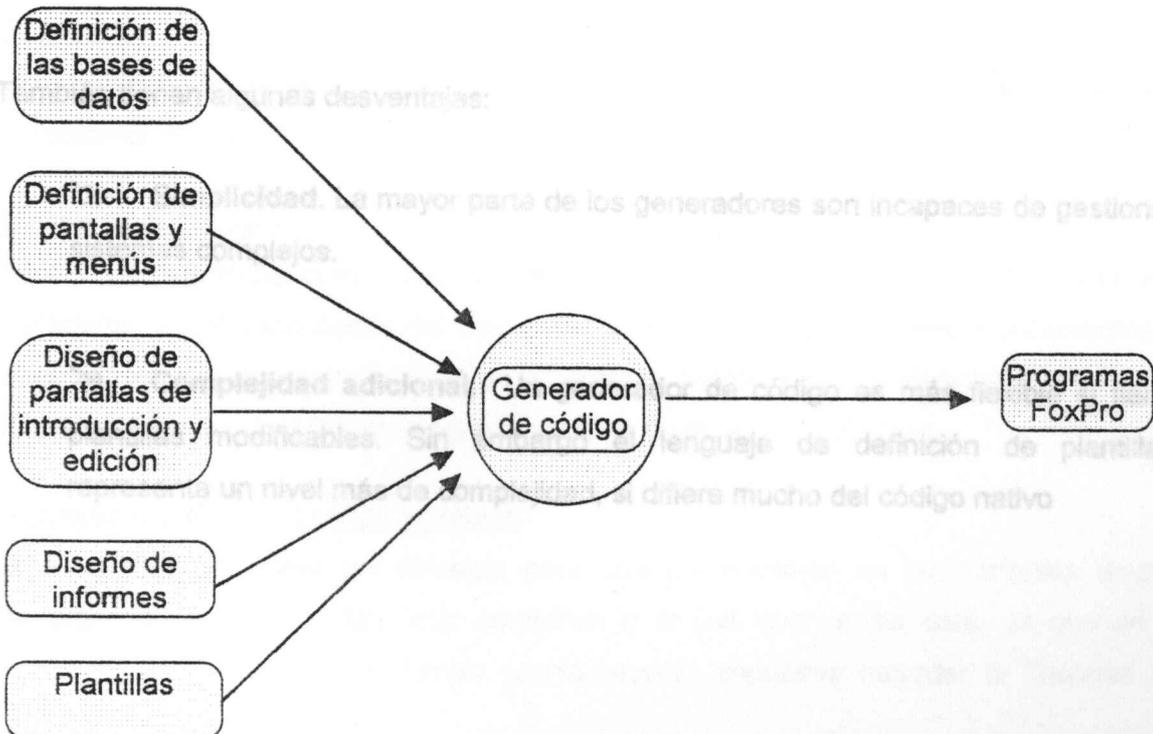
Documentación del sistema.

Producción de listados del código fuente convenientemente documentados, de referencias cruzadas de variables y de diagramas de estructuras del programa.

Generadores de código.

Los generadores de código o de aplicaciones existen desde hace algún tiempo; son anteriores a la aparición del término CASE, y en la actualidad se han integrado dentro de este concepto.

La idea en la que se basan los generadores es: el programador sólo define los elementos de datos, la disposición de las pantallas de introducción y presentación de datos, los menús y los formatos de los informes; y todo ello es interpretado por el generador para producir el código fuente que implementa el sistema definido.



FoxPro dispone de posibilidades propias de generación de código, a través de CREATE SCREEN, CREATE MENU, y FOXAPP, que permiten al programador definir determinada disposición de pantalla o menú, y dejar que FoxPro genere su propio código fuente (modificable).

Las principales ventajas de los generadores de código son las siguientes:

✍ **Velocidad.** Permiten automatizar la tediosa tarea de codificar las pantallas de introducción de datos y de generación de informes, favoreciendo que el programador se concentre en tareas de dificultad más intelectual que mecanográfica.

✍ **Método de aprendizaje.** Las plantillas de los generadores de código producirán un código consistente, conforme a las reglas de la buena programación, lo que servirá a los programadores a adoptar estas costumbres.

✍ **Uniformidad.** La salida del generador es predecible y por lo tanto el mantenimiento es relativamente sencillo.

También tienen algunas desventajas:

✍ **Simplicidad.** La mayor parte de los generadores son incapaces de gestionar sistemas complejos.

✍ **Complejidad adicional.** Un generador de código es más flexible si tiene plantillas modificables. Sin embargo el lenguaje de definición de plantillas representa un nivel más de complejidad, si difiere mucho del código nativo

La correcta práctica de la codificación.

Las recomendaciones comúnmente aceptadas para la escritura de programas FoxPro pretenden simplificar la lectura y comprensión de los programas, ya sea durante el desarrollo inicial o el mantenimiento futuro. Por sí solo FoxPro no obliga a seguir ningún estándar; son los propios programadores quienes imponen su propia disciplina para la aplicación de estas reglas a fin de hacer una buena codificación.

Sugerencias de codificación en FoxPro

Empleo de mayúsculas y minúsculas

Es conveniente que durante la codificación se siga un mismo criterio para el empleo de mayúsculas y minúsculas, ya que esto ayudará a identificar más rápidamente los comandos, variables, archivos, comentarios etc.

Sangrado de las estructuras de control

Algunas de las sentencias de FoxPro, tales como Do While... EndDo e If...Else..EndIf, implementan estructuras lógicas de control. Podemos ver sin dificultad cuáles con los enunciado que abarcan si sangramos las líneas contenidas dentro de ellas. El beneficio se incrementa si estas estructuras se encuentran anidadas.

Espaciado

No debemos rehuir al uso de líneas en blanco. El uso de estas ayuda a separar actividades diferentes.

Líneas de comentarios

FoxPro permite incluir comentarios poniendo un asterisco al comienzo de una línea o dos caracteres ampersand detrás del comando. Cuanto más explícitos sean los comentarios más fácil será el mantenimiento de los programas. Siempre es recomendable incluir unas líneas al principio del programa para recordar la finalidad de ese código.

Nombres significativos de las variables

Resulta muy útil hacer un esfuerzo para que los nombres de las variables tengan relación con el tipo de dato que contienen o el uso que se les dará; ya que de lo contrario, con el paso del tiempo podría resultar imposible recordar la finalidad de algunas variables.

Empleo de palabras clave completas

FoxPro permite abreviar sus palabras clave con los cuatro primeros caracteres, sin embargo es preferible evitar esta práctica que reduce la legibilidad de los programas.

Espaciado de las expresiones

El uso de operadores aritméticos o relacionales, sobre todo en ecuaciones de tamaño considerable debe hacerse usando un espaciado adecuado de los operandos, de tal manera que no resulte tan complicado reestructurar o revisar las ecuaciones.

Importancia de la programación orientada a eventos.

La característica sobresaliente de esta estrategia de programación es que permite trabajar en la forma en que pensamos: es una aproximación más exacta a la manera en que trabaja la inteligencia humana, que la tradicional programación secuencial.

Programación orientada a eventos contra Programación Secuencial.

En un sentido básico un programa orientado a eventos es aquel que es controlado por eventos externos; en contraste con un programa secuencial en el cual se ejecutan una secuencia de pasos en orden predefinido y exclusivamente en ese orden.

En la programación orientada a eventos no hay un flujo predefinido, simplemente se definen los eventos que se pueden manejar y se espera hasta que algo pasa. Por ejemplo cuando se hace una selección en un menú, el manejador de eventos es llamado para procesar la selección.

Programación orientada a eventos contra Multitareas.

Las aplicaciones controladas por eventos tienen que tratar con muchos de los mismos problemas que los sistemas operativos multitarea. Cada ventana de aplicaciones pueda ser considerada un proceso aparte y cuando está activa su programa conserva su lugar y termina el proceso.

Actualmente Windows y FoxPro gracias a FoxPro para Windows, Visual FoxPro 3 para Windows 3.x y Visual FoxPro para Windows 95, proporcionan la última combinación de control de eventos y ambiente multitarea permitiendo correr múltiples aplicaciones simultáneas.

Programación orientada a eventos.

La programación orientada a eventos es el futuro del diseño de software. La implementación de una interfaz de usuario en FoxPro 100% orientada a eventos es todavía una tecnología en evolución. Muchas características necesarias para un soporte completo de una aplicación orientada a eventos no existen o bien están incompletas en el lenguaje FoxPro, por lo que tienen que ser implementadas manualmente.

para la mejor comprensión del diseño físico del sistema, así como también se dedica un apartado

Importancia de la programación orientada a eventos. de conectividad del sistema. Los

La característica sobresaliente de esta estrategia de programación es que permite trabajar en la forma en que pensamos; es una aproximación más exacta a la manera en que trabaja la inteligencia humana, que la tradicional programación secuencial.

Programación orientada a eventos contra Programación Secuencial.

En un sentido básico un programa orientado a eventos es aquel que es controlado por eventos externos; en contraste con un programa secuencial en el cual se ejecutan una secuencia de pasos en orden predefinido y exclusivamente en ese orden.

para que el alumno lo resuelva y que el maestro lo califique, sino que es mucho más cómodo que el

En la programación orientada a eventos no hay un flujo predefinido, simplemente se definen los eventos que se pueden manejar y se espera hasta que algo pasa. Por ejemplo cuando se hace una selección en un menú, el manejador de eventos es llamado para procesar la selección.

Está claro que para poder tener acceso a esta opción es necesario contar con una red

Programación orientada a eventos contra Multitareas. indispensable ya que el sistema

Las aplicaciones controladas por eventos tienen que tratar con muchos de los mismos problemas que los sistemas operativos multitarea. Cada ventana de aplicaciones puede ser considerada un proceso aparte y cuando está activa su programa conserva su lugar y termina el proceso.

de el hecho de que ya existe un guión con los reactivos que se desean ser aplicados.

Actualmente Windows y FoxPro gracias a FoxPro para Windows, Visual FoxPro 3 para Windows 3.x y Visual FoxPro para Windows 95, proporcionan la última combinación de control de eventos y ambiente multitarea permitiendo correr múltiples aplicaciones simultáneas.

El diseño físico del sistema.

En la fase del diseño lógico del sistema se realizó un estudio individual de los módulos principales, se presentó su diagrama de flujo así como una descripción de la forma en que opera y, en una forma no muy técnica de su funcionamiento interno. Dentro del diseño físico se consideró que es conveniente retomar dos de los módulos más importantes del sistema y documentarlo con mayores detalles técnicos para la mejor comprensión del diseño físico del sistema, así como también se dedica un apartado especial para explicar como se manejaron los aspectos de conectividad del sistema. Los módulos son:

- La generación de los exámenes interactivos.
- La calificación de los exámenes interactivos.

Generación del examen interactivo.

Descripción.

Una de las principales cualidades del SIGEE es la aplicación de exámenes interactivos, ya que en este tipo de exámenes no es necesario imprimir el examen, para que el alumno lo resuelva y que el maestro lo califique, sino que es mucho más cómodo que el alumno resuelva su examen en la computadora, y la calificación sea automática, con la ventaja adicional de que si dos o más personas están resolviendo el mismo examen en un centro de cómputo cada uno observará una distribución diferente de las respuestas.

Está claro que para poder tener acceso a esta opción es necesario contar con una red de computadoras, aunque de cualquier forma no es indispensable ya que el sistema puede funcionar también en forma monousuario. (Consultar apartado 'Conectividad').

Funcionamiento.

Para describir el funcionamiento de la generación de un examen interactivo partiremos de el hecho de que ya existe un guión con los reactivos que se desea sean aplicados, los cuales fueron tomados de la base de datos de reactivos Evaluaci.dbf. (Para conocer la estructura de estos archivo consulte la sección de Diccionario de datos.)

El primer paso consiste en crear un arreglo de datos en memoria en el cual se almacenen todos los datos que sean importantes para la presentación del reactivo en pantalla como lo son: el número de reactivo que le toca en el examen, el número de respuestas que contiene, el tamaño o alto del reactivo, la página a la cual pertenece, el orden en el que se desplegaron las respuestas, las contestaciones que ha dado el alumno, entre otros. La dimensión de este arreglo será relativa al número de reactivos del examen.

Una vez en esta etapa es posible presentar la primer página del examen, para realizar este paso se utilizan una serie de funciones especializadas en el tratamiento de cada reactivo, es decir, que como cada tipo de reactivo presenta sus propias modalidades de respuesta y operación es necesario que sean tratados en forma particular.

Por ejemplo, los reactivos dicotómicos solamente permiten una de dos respuestas A o B o bien vacío en caso de que no se conozca la respuesta, por el contrario los reactivos de relación puede contener hasta diez respuestas y una respuesta vacía puede ser correcta si es que esa opción no tiene correspondiente.

Modelo operativo.

En este momento el alumno ya tiene en su pantalla el examen y tiene la opción de avanzar o retroceder a través de sus páginas, lo cual tendrá como efecto que cada que cambie de página las respuestas del alumno sean transferidas del arreglo de memoria al examen y que una nueva página sea transferida del arreglo a la pantalla. Gracias a este mecanismo en caso de una falla de energía eléctrica lo más que se perderá de información será una página de respuestas del examen.

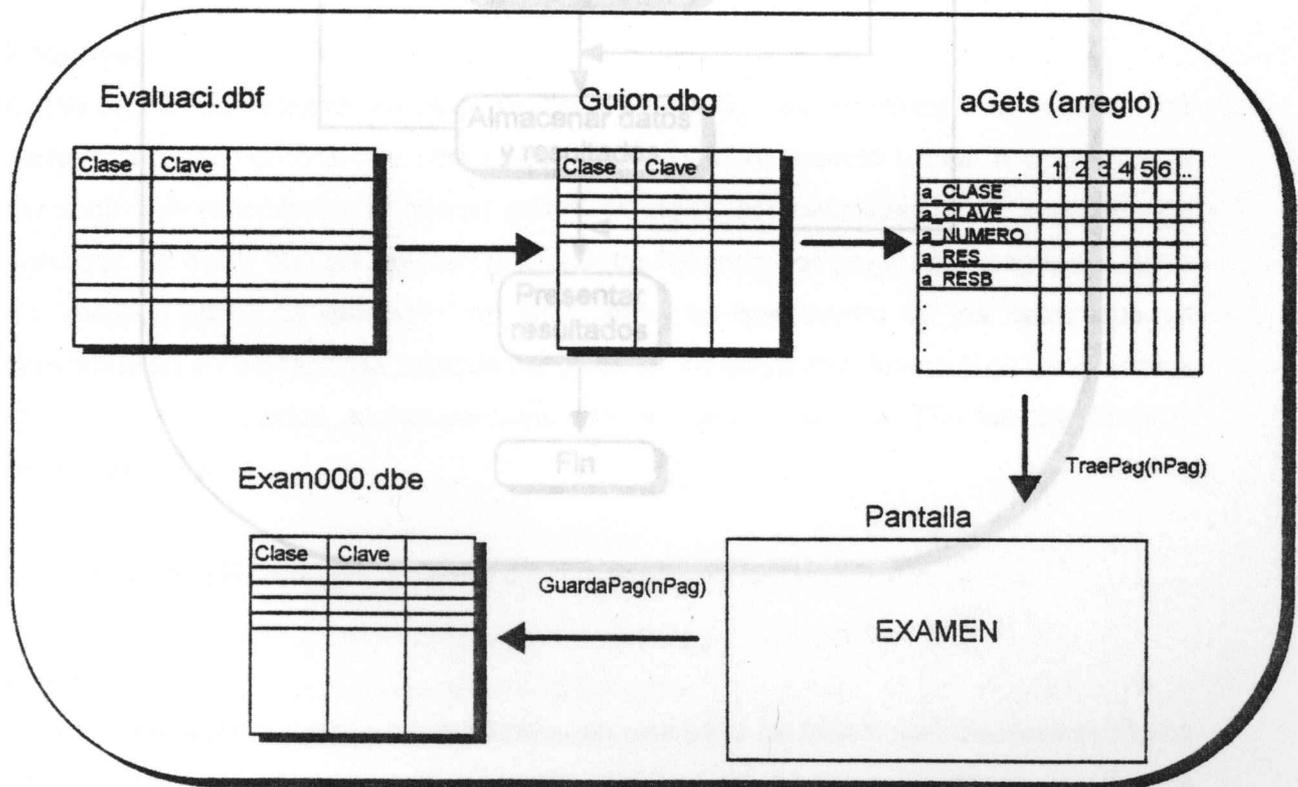
Estructuras de datos.

Arreglo aGets.

La siguiente tabla representa un renglón del arreglo aGets. Por comodidad y claridad en la programación se convino el uso de la macrosustitución, así en lugar de usar la notación aGets(nReactivo, 1), se sustituirá con aGets(nReactivo, a_CLASE), lo cual da una idea más clara del dato al que estamos accediendo.

Columna	Tipo	Utilidad
a_CLASE	carácter	relación con Evaluaci.dbf
a_CLAVE	carácter	relación con Evaluaci.dbf
a_NUMERO	numérico	numeración en el examen
a_RES	numérico	número de opciones
a_RESB	numérico	botón seleccionado
a_RES0	carácter	respuesta del alumno
a_RES1	carácter	respuesta del alumno
a_RES2	carácter	respuesta del alumno
a_RES3	carácter	respuesta del alumno
a_RES4	carácter	respuesta del alumno
a_RES5	carácter	respuesta del alumno
a_RES6	carácter	respuesta del alumno
a_RES7	carácter	respuesta del alumno
a_RES8	carácter	respuesta del alumno
a_RES9	carácter	respuesta del alumno
a_ALTO	numérico	alto del reactivo
a_PAG	numérico	página del reactivo
a_ORDEN	carácter	orden de las respuestas
a_ORDEN2	carácter	orden de la 2ª columna Rela
a_RELE	numérico	número de elementos Rela

Modelo operativo.



Descripción:

Califica

Este módulo tiene como objetivo calificar las respuestas de los exámenes interactivos y generar un reporte ya sea en archivo, pantalla o impresora; dichas respuestas se encuentran en una base de datos (Consultar la estructura de la base de datos del examen en la sección Diccionario de datos). Este archivo contiene los datos necesarios para poder realizar el proceso de calificación de los exámenes y es creado por el usuario en el momento que se crea los exámenes. A manera de recordatorio diremos que el nombre de estos archivos consta de cinco caracteres provistos por el usuario, que pueden ser los mismos o diferentes, pertenecen, y tres cifras que sirven para poder crear una secuencia de los mismos caracteres iniciales y una cifra consecutiva que es asignada automáticamente por el sistema.

Una vez seleccionado el examen el sistema hace la comparación y evaluación de las respuestas y obtiene las calificaciones de cada uno, el tiempo que muestra el estado del proceso.

Al finalizar la operación, el usuario puede generar el destino de reporte.

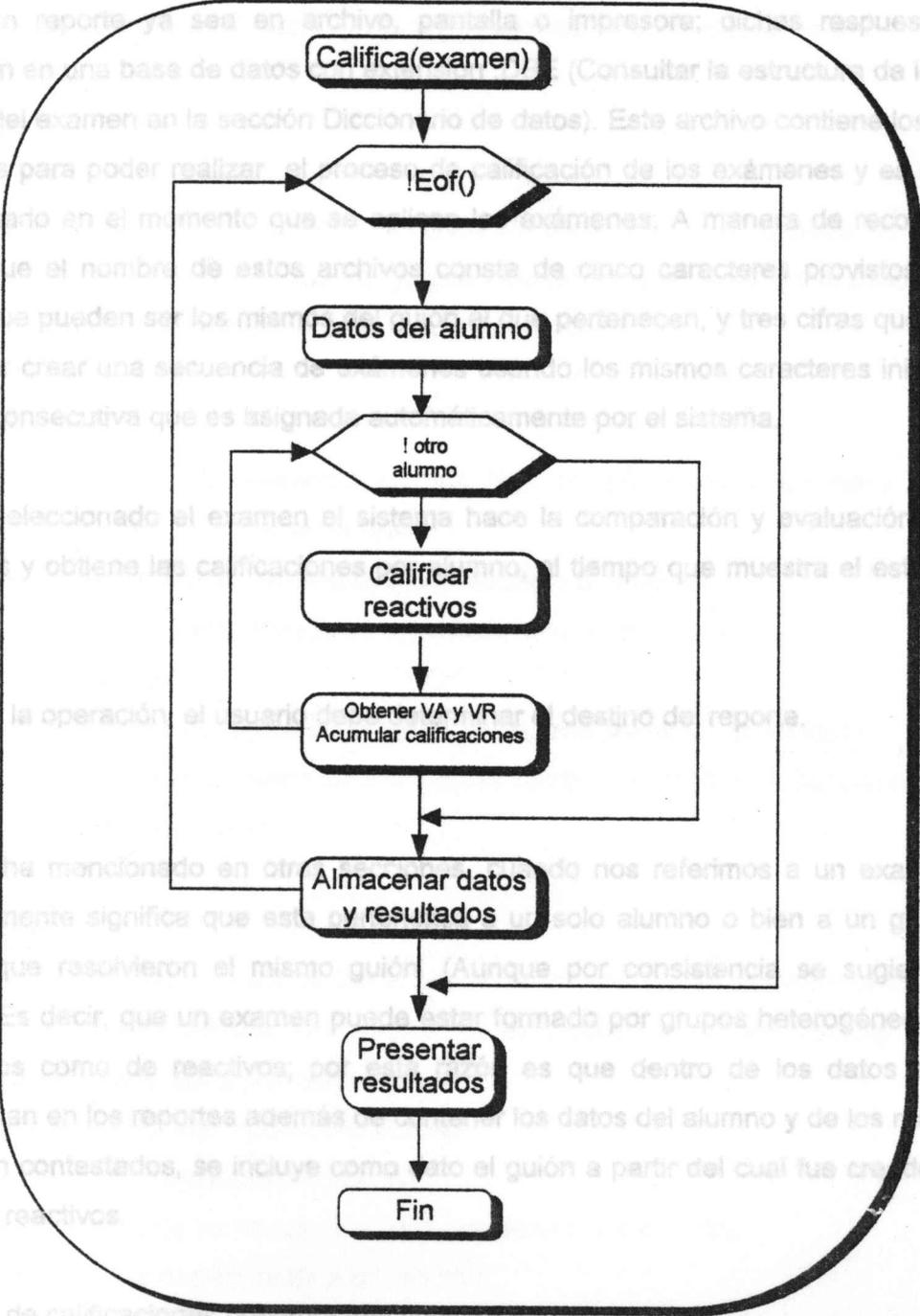
Reportes:

Como se ha mencionado en otros módulos cuando nos referimos a un examen no necesariamente significa que este sea para un solo alumno o bien a un grupo de alumnos que resolvieron el mismo guión (Aunque por consistencia se sugiera esta práctica), es decir, que un examen puede estar formado por grupos heterogéneos tanto de alumnos como de reactivos; por lo tanto es que dentro de los datos que se proporcionan en los reportes además de los datos del alumno y de los reactivos que fueron contestados, se incluye como parte del guión a partir del cual fue creado cada bloque de reactivos.

El módulo de calificaciones genera tres tipos de reportes:

A. En archivo.

Este tipo de reporte guarda los resultados en una base de datos que conserva el mismo nombre del examen pero con extensión .DBC. Esta base de datos se conservará



Descripción.

Este módulo tiene como objetivo calificar las respuestas de los exámenes interactivos y generar un reporte ya sea en archivo, pantalla o impresora; dichas respuestas se encuentran en una base de datos con extensión .DBE (Consultar la estructura de la base de datos del examen en la sección Diccionario de datos). Este archivo contiene los datos necesarios para poder realizar el proceso de calificación de los exámenes y es creado por el usuario en el momento que se aplican los exámenes. A manera de recordatorio diremos que el nombre de estos archivos consta de cinco caracteres provistos por el usuario, que pueden ser los mismos del guión al que pertenecen, y tres cifras que sirven para poder crear una secuencia de exámenes usando los mismos caracteres iniciales y una cifra consecutiva que es asignada automáticamente por el sistema.

Una vez seleccionado el examen el sistema hace la comparación y evaluación de las respuestas y obtiene las calificaciones por alumno, al tiempo que muestra el estado del proceso.

Al finalizar la operación, el usuario debe determinar el destino del reporte.

Reportes

Como se ha mencionado en otras secciones, cuando nos referimos a un examen no necesariamente significa que este pertenezca a un solo alumno o bien a un grupo de alumnos que resolvieron el mismo guión. (Aunque por consistencia se sugiera esta práctica). Es decir, que un examen puede estar formado por grupos heterogéneos tanto de alumnos como de reactivos; por esta razón es que dentro de los datos que se proporcionan en los reportes además de contener los datos del alumno y de los reactivos que fueron contestados, se incluye como dato el guión a partir del cual fue creado cada bloque de reactivos.

El módulo de calificaciones presenta tres tipos de reportes:

A. En archivo.

Este tipo de reporte guarda los resultados en una base de datos que conserva el mismo nombre del examen pero con extensión .DBC. Esta base de datos se conservará

permanentemente y podrá ser impresa en el momento que usuario lo requiera. Por otro lado solamente podrá existir un archivo de calificaciones por examen, por lo que si se deseara volver a generar el archivo de calificaciones que ya existe este tendría que sobrescribirse.

Módulo Calificar Exámenes Interactivos

B. En pantalla.

Los reportes en pantalla se presentan a través de un browse que contiene los mismos campos que la base de datos que se genera cuando el reporte es en archivo, con la diferencia de que la presentación de los datos es efímera.

Parámetros:

C. En impresora

Los reportes impresos se presentan en una hoja tamaño carta y los datos son los mismos que en los dos reportes anteriores.

En seguida se presentan los campos que contendrán los reportes

1. Nombre del guión a partir del que se creó cada bloque del examen
2. Nombre del alumno
3. Calificación absoluta (Sumatoria de todos los valores absolutos obtenidos)
4. Calificación relativa (Sumatoria de los valores relativos obtenidos / Sumatoria de los valores relativos esperados)

Si cTipo = Dicotómicos

Funcionamiento

Para realizar el proceso de calificación el primer paso consiste en seleccionar la BD del examen que se desea calificar. Una vez hecho esto cada registro es procesado según los datos que contiene de la siguiente manera:

A) Si es un registro de información del alumno :

1. Se obtiene la información del alumno
2. Se reinicia los valores necesarios para contabilizar los aciertos

B) Si un registro que corresponde a un reactivo:

1. Según la clase de reactivo, el orden en el que se desplegaron las opciones y la respuesta del alumno se asigna una puntuación entre 0 y 1.
2. Se calcula el valor absoluto y relativo obtenido multiplicando la puntuación por el VA y el VR esperados.
3. Se acumulan los valores obtenidos para obtener la calificación final del alumno.

Por la importancia que tiene la función de calificación de cada reactivo de acuerdo a su clase se proporciona a continuación el algoritmo el español estructurado:

Sino

Si cTipo = Solución múltiple

cResp1 = CHR(EN("1", cTexE) + 64)

Módulo Calificar Exámenes Interactivos

Descripción: Obtiene la respuesta correcta de un reactivo, compara con la respuesta del alumno y regresa un número entre 0 y 1 (0 = incorrecto; 1 = correcto)

Parámetros:

cTipo: Tipo del reactivo (dicotómico, complementación etc.)

cTexE: Cadena que contiene el orden en el que se desplegaron las respuestas en el examen

cTexR: Cadena que contiene el orden en el que se desplegó la segunda columna en los reactivos de Relación.

cExaResp: Cadena que contiene la o las respuestas del alumno.

Si cTipo = Dicotómicos

cCorrecta = CHR(AT("1", cTexE) + 64) && se obtiene la letra que contiene la resp. correcta

Si cCorrecta == cExaResp

Regresa 1

FinSi

Sino

Si cTipo = Solución única

cCorrecta = CHR(AT("1", cTexE) + 64) && se obtiene la letra de la resp. correcta

Si cCorrecta == cExaResp

Regresa 1

FinSi

FinSi

Sino

Si cTipo = Complementación

Si cExaResp = "1"



Conectividad. Regresa 1

FinSi

Sino Cuando una base de datos tiene que ser compartida por varios usuarios, se precisa de un sistema que pueda ser instalado bajo la forma de una red o bien por medio de un servidor.

Si cTipo = Solución múltiple

cResp1 = CHR(EN("1", cTexE) + 64)

cResp2 = CHR(EN("1", cTexE) + 64)

Si cExaResp(1) == cResp1 Y cExaResp(2) == cResp2

Regresa 1

Sino

Si cExaResp(1) == cResp1 Y cExaResp(2) == cResp2

Regresa 0.5

FinSi

FinSi

FinSi

Sino

Si cTipo == Ordenamiento La estructura de datos que se mantiene en el sistema es la siguiente:

nAciertos = 0

Mientras i <= LEN(cTexE)

Si cExaResp(i) == cTexE(i)

nAciertos = nAciertos + 1

FinSi

i = i + 1

FinMientras

Regresa (nAciertos/LEN(cTexE))

Sino

Si cTipo == Ordenamiento

nAciertos = 0

Mientras i <= LEN(cTexE)

Si cExaResp(i) == CHR(cTexE(i) + 64)

nAciertos = nAciertos + 1

FinSi

i = i + 1

FinMientras

Regresa (nAciertos/LEN(cTexE))

FinSi

Regresa 0

Conectividad.

Cuando una base de datos tiene que ser compartida por varios usuarios, se precisa de un sistema multiusuario que pueda ser instalado bajo la forma de una red o bien por medio de un sistema operativo multiusuario.

Los sistemas para operar en redes locales deben poseer medios sofisticados para mantener la seguridad de la base de datos y evitar al máximo los errores.

Si son varios los usuarios que tienen acceso a la base de datos conviene establecer fórmulas de control y restringir ciertos usos con el fin de que los usuarios no se interfieran en su trabajo o realicen modificaciones erróneas partiendo de datos no actualizados.

La estrategia que siguió para mantener estos controles en el sistema es la siguiente:

En los movimientos de altas existe la posibilidad de que dos o más usuarios den de alta la misma clave en el mismo momento, dado que la verificación se hace en la captura, ambos registros se darían de alta, dando lugar a inconsistencias en el sistema. Por este motivo se tomó la decisión de bloquear el archivo en lugar de solo bloquear el registro para realizar la alta, lo cual es justificable considerando que el tiempo que el archivo se encontrará bloqueado es mínimo y que hacerlo sí evitará posibles inconsistencias dentro de la base de datos. (Ver diagramas LockAlta y FBloqueo)

Como los movimientos de bajas no involucran el caso de las claves repetidas como en las altas, solo se bloquea el registro que está siendo dado de baja. Ver diagramas LockBaja y RBloqueo)

En el movimiento de edición puede suceder que dos o más usuarios deseen editar simultáneamente el mismo registro, no siendo necesario bloquear el archivo porque al no poder ser modificada la clave se evita que los registros puedan ser duplicados.

Funcionamiento en red de los exámenes interactivos.

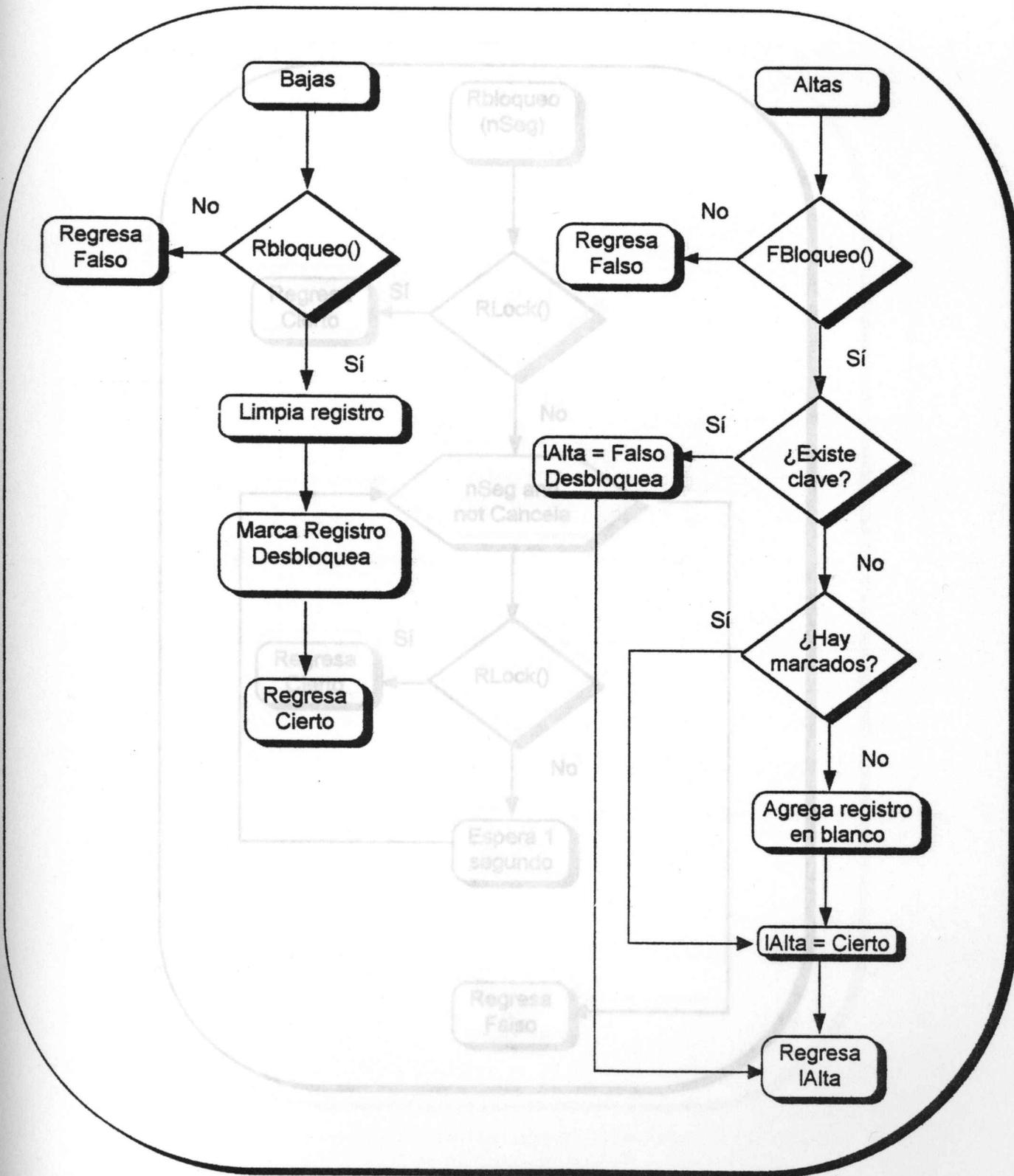
Cuando un grupo de n personas están resolviendo n exámenes dentro de el mismo sistema de red, la estrategia de red se realiza de la siguiente manera:

- Se localiza la base de datos del examen en la cual se guardarán las respuestas, dicha base de datos podrá ser la misma para todos, diferente para cada persona o bien una por cada determinado número de personas.
- Se asigna una identificación única por cada persona que resuelve el examen, la cual es un número consecutivo según el archivo en el que se vayan a guardar las respuestas.
- Se guarda la estructura del examen por persona.
- Se desbloquea el archivo; en este momento las demás estaciones de trabajo que estén en espera de guardar la estructura podrán bloquear el archivo y realizar la misma operación.

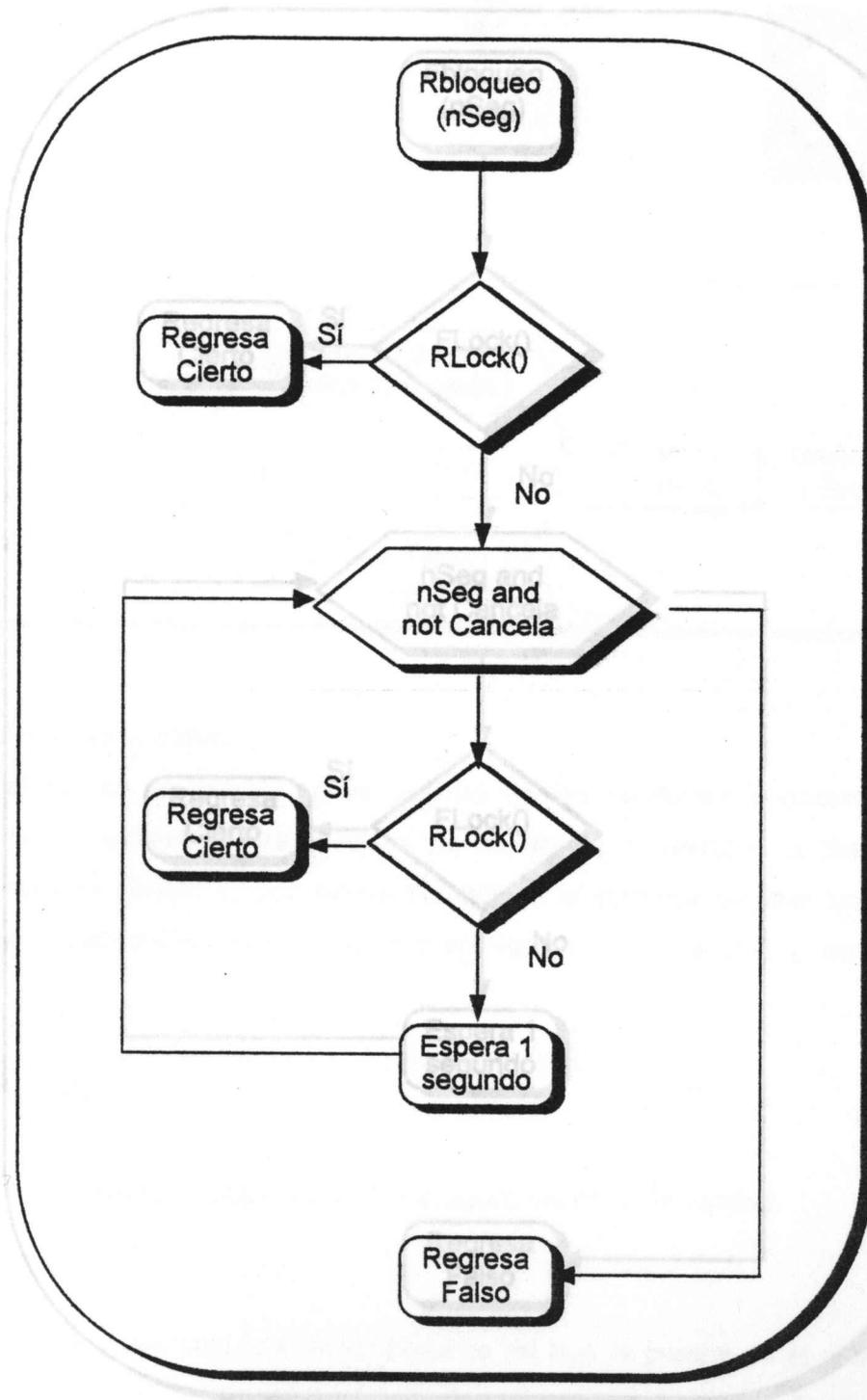
Gracias a esta estrategia, no es necesario volver a bloquear el archivo en ningún otro momento, ya que cada examen tiene apartado e identificado su espacio en el archivo. De esta manera, lo que resta del proceso es que el evaluado conteste las páginas del examen, las cuales se van guardando en el archivo cada vez que se cambia de página, evitando que una falla en el suministro de la energía ocasione la pérdida de los exámenes.

Para conocer el funcionamiento integral de los exámenes interactivos consultar la sección donde se describe dicho módulo.(Exámenes Interactivos)

Funciones de Bajas y Altas (Registros)

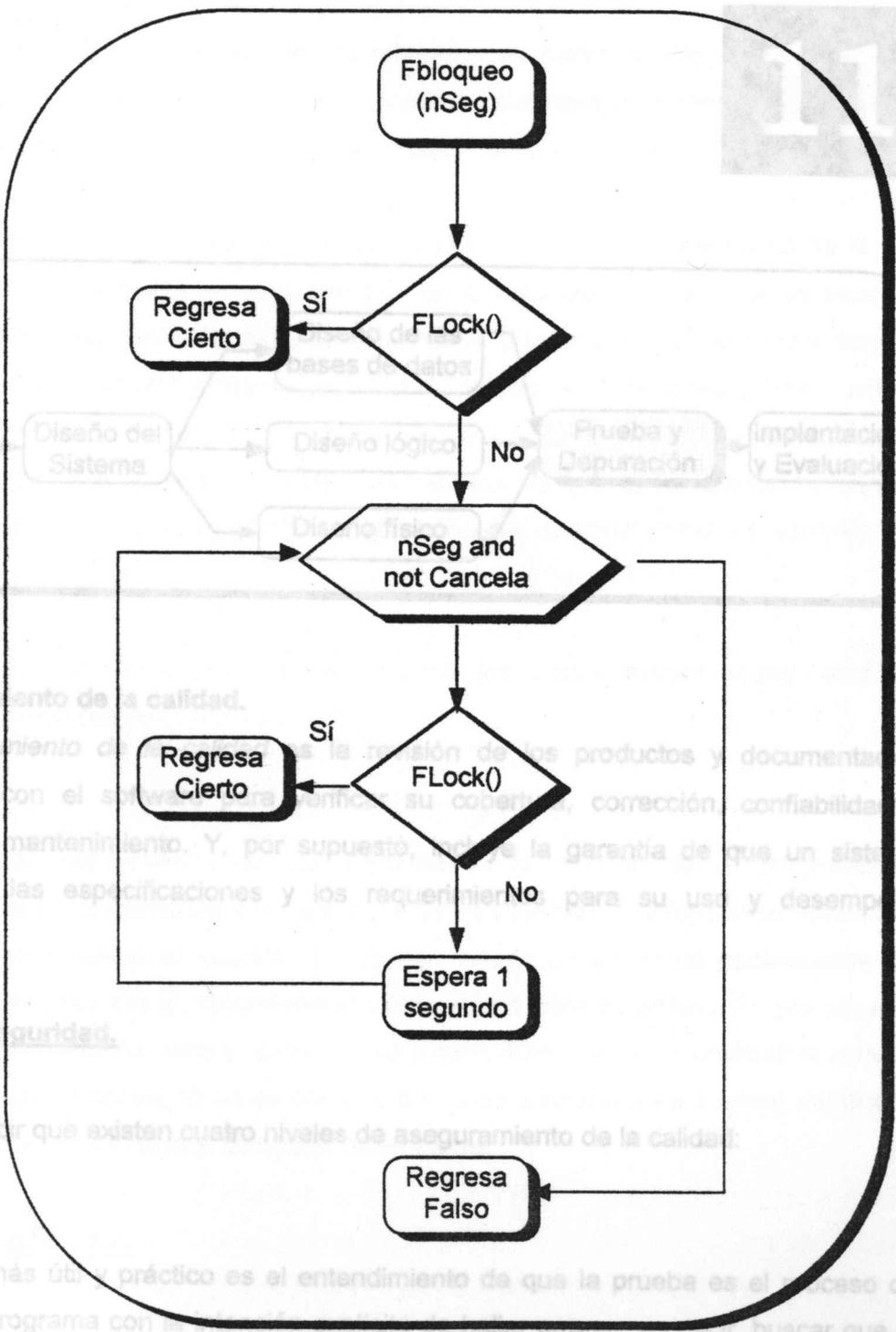


Función Rbloqueo (Bloqueo de Registros)



Función Fbloqueo (Bloqueo de Archivos)

11



PRUEBA Y DEPURACION

11



El aseguramiento de la calidad.

El *aseguramiento de la calidad* es la revisión de los productos y documentación relacionada con el software para verificar su cobertura, corrección, confiabilidad y facilidad de mantenimiento. Y, por supuesto, incluye la garantía de que un sistema cumple con las especificaciones y los requerimientos para su uso y desempeño deseados.

Niveles de seguridad.

Se puede decir que existen cuatro niveles de aseguramiento de la calidad:

Prueba.-

El enfoque más útil y práctico es el entendimiento de que la prueba es el proceso de ejecutar un programa con la intención explícita de hallar errores, es decir, buscar que el

programa falle. También es bueno recordar que un programa de prueba efectivo no garantiza la confiabilidad del sistema. La confiabilidad es asunto del diseño.

Verificación y validación.

Como la prueba, la verificación tiene la intención de hallar errores. Se lleva a cabo ejecutando un programa en un ambiente simulado. La *validación* se refiere al proceso del uso del software en un ambiente no simulado para hallar sus errores.

La primera verificación a veces es llamada *prueba alfa*. La retroalimentación de la fase de validación generalmente produce cambios en el software para resolver los errores y las fallas que se descubren. Se elige un conjunto de instalaciones usuarias que ponen a trabajar el sistema en un ambiente real. Estas instalaciones de *prueba beta* usan el software de manera cotidiana. El sistema está a prueba en toda la extensión de la palabra; excepto que los usuarios están inadvertidos de que están usando un sistema que puede fallar. Sin embargo tanto la operación del sistema como los usuarios son reales.

La validación puede durar varios meses, durante los cuales pueden ocurrir fallas y el software será modificado.

Certificación.

La certificación del software es una garantía de lo correcto de un programa, su importancia va en aumento para las aplicaciones de sistemas de información. Una forma de certificar el software es cuando la empresa asigna un grupo de especialistas que cuidadosamente revisan la documentación del sistema para determinar lo que afirma el vendedor que el sistema hace y cómo lo lleva a cabo. Entonces ellos prueban el software contra estas afirmaciones. Si no se encuentran serias discrepancias o fallas, certificarán que el software hace lo que la documentación afirma.

Estrategias de prueba.

Ya se ha indicado que la filosofía detrás de la prueba es la de encontrar errores. Un caso de *prueba* es un conjunto de datos que el sistema procesará como entrada normal. Sin

embargo los datos se crean con la intención expresa de determinar si el sistema los procesará correctamente.

Hay dos estrategias generales para la prueba del software.

Prueba de código.-

Por medio de esta estrategia se examina la lógica del programa. Lo que se realiza es desarrollar casos que produzcan la ejecución de cada instrucción en el programa o módulo.

Pruebas de especificación.-

Para llevar a cabo esta prueba se examinan las especificaciones que señalan lo que el sistema debe hacer y cómo lo debe llevar a cabo bajo diferentes condiciones.

En esta estrategia se trata al sistema como si fuera una caja negra: no se estudia ni se prueban las instrucciones del programa, simplemente se comprueba que el sistema arroje los resultados que se esperaban.

Manejo de las prácticas de prueba.

Independientemente de cual sea la estrategia de prueba que se esté siguiendo, hay ciertas prácticas preferidas para garantizar que la prueba sea útil. Los niveles de prueba y los tipos de datos de prueba son importantes en el proceso real.

Niveles de prueba.-

Existen dos niveles en los cuales se debe realizar las pruebas:

Pruebas parciales.

Una manera de realizar pruebas a fondo a un sistema es por medio de las pruebas parciales, en las cuales se deben probar todos y cada uno de los módulos del sistema de manera independiente, tomando como un módulo las unidades de software que se ensamblan e integran para llevar a cabo una función específica.

Los casos de prueba deben ser exhaustivos, es decir, que deben probar cada opción y condición.

Las pruebas parciales se pueden llevar a cabo de manera ascendente, comenzando con los niveles más pequeños y de nivel inferior. Para cada módulo un programa conductor ejecuta el módulo y proporciona los datos necesarios, de esta forma se pide que el módulo se desempeñe en la forma en que lo haría al encajarse dentro de el sistema. Después de probar los niveles inferiores, la atención se centra en los del siguiente nivel.

Prueba de sistemas.

Esta no prueba el software en sí, sino la integración de cada módulo en el sistema. También busca las discrepancias entre el sistema y su objetivo original, especificaciones y documentación del sistema. La preocupación principal es la compatibilidad de los módulos individuales.

Las pruebas de sistemas también deben verificar que los tamaños de los archivos son adecuados y de que los índices se construyeron correctamente.

Diseño de datos de prueba.

Uso de datos de prueba reales.-

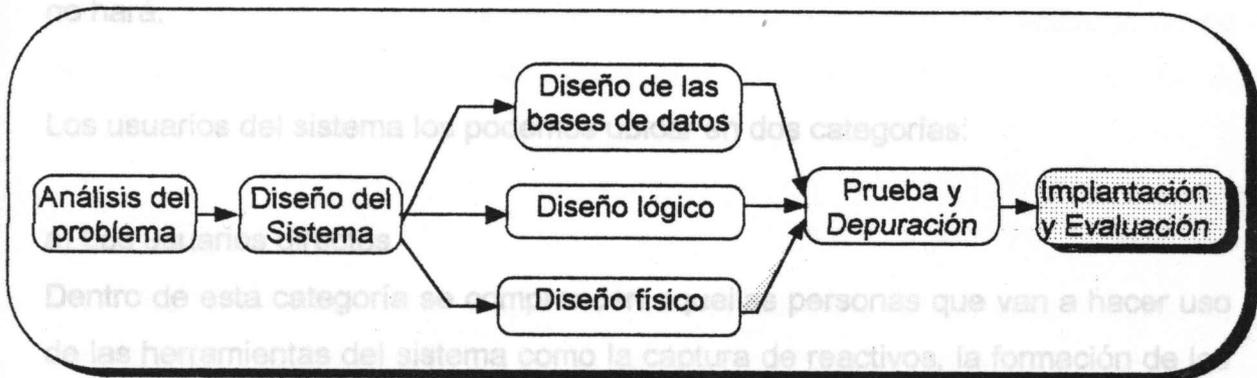
Los datos deben ser obtenidos de la que será la fuente real del sistema. El problema de este tipo de datos es que el sesgo hacia los valores típicos no proporciona una verdadera prueba del sistema y de hecho ignora los casos más probables que causan la falla del mismo.

Uso de datos de prueba artificiales,.

Estos se crean solamente con fines de prueba, y se utilizan para probar todas las combinaciones de formatos y valores. Frecuentemente un equipo diferente al que elabora el software diseña los datos de prueba artificiales.

IMPLANTACION Y EVALUACION

12



La implantación incluye todas aquellas actividades que tienen lugar para realizar la transición del sistema anterior al nuevo. Como ya sabemos en este momento la manera en que trabaja el sistema anterior es cien por ciento manual, y la elaboración del nuevo sistema nació no de una solicitud por parte de los usuarios, sino como una propuesta de una nueva manera más rápida y eficiente del desempeño de las actividades. Por esa razón la implantación del sistema debe dejar convencidos a los usuarios de que el uso del sistema les redituara en términos tangibles a corto plazo como lo es tiempo, la seguridad y la eficiencia.

Son aquellos que utilizan las salidas o reportes del sistema. En nuestro caso son tres los aspectos que se cubren en la implantación:

- Capacitación del personal
- Procedimientos de conversión
- Revisión después de la implantación.

Son breves instrucciones las cuales pueden ser escritas u orales y que generalmente son dadas por los usuarios directos del sistema.

Capacitación.

Aún los sistemas elaborados por las grandes empresas pueden tener éxito o fracasar debido a la forma en que se operan y usan. Por lo tanto, la calidad de la capacitación recibida por el personal relacionado con el sistema ayuda u obstruye, y puede llegar a impedir, la implantación exitosa de un sistema de información. Aquellos que estén relacionados con el sistema deben conocer con detalle cuáles serán sus papeles, cómo pueden usar el sistema y qué hará o qué no hará.

Los usuarios del sistema los podemos ubicar en dos categorías:

a) Los usuarios directos

Dentro de esta categoría se comprenden aquellas personas que van a hacer uso de las herramientas del sistema como la captura de reactivos, la formación de los guiones y la generación y aplicación de los exámenes. Se puede decir que son quienes aprovechan todas las ventajas que el sistema ofrece y por lo mismo deben conocer a profundidad lo que el sistema hace, familiarizarse con todas las pantallas y procesos y saber como actuar ante cualquier contingencia.

Lo recomendable para este tipo de usuarios es la elaboración de un manual de operación además de una capacitación personal.

b) Los usuarios indirectos.

Son aquellos que utilizan las salidas o reportes del sistema. En nuestro caso son las personas que reciben los cuestionarios, ya sean impresos o interactivos para resolverlos. Estos usuarios no requieren saber mucho a cerca del funcionamiento del sistema, solamente saber interpretar lo que indica el reporte o salida y la manera en que el sistema espera que actúe. Su capacitación se refiere a unas breves instrucciones las cuales pueden ser escritas u orales y que generalmente son dadas por los usuarios directos del sistema.

Método de conversión.

El método que se recomienda para la instalación del sistema es el de *sistemas paralelos*. Este es el método más seguro de conversión. Con este enfoque, los operadores siguen trabajando con el sistema anterior de la forma acostumbrada con unos grupos y otros con el nuevo sistema, de esta forma se garantiza que el sistema esté funcionando adecuadamente y la conversión se va haciendo en cierta medida por etapas, capacitando y acostumbrando poco a poco a los usuarios indirectos y a los directos al uso del nuevo sistema. Sin embargo no se descarta que sea posible usar algún otro método de conversión en forma satisfactoria.

La evaluación.

Una vez terminadas estas etapas el sistema quedará funcionando durante el tiempo que sea necesario en forma de versión beta, y se estará evaluando periódicamente el funcionamiento para comprobar que el sistema cumple con las expectativas y determinar si es necesario que se realicen algunas mejoras.

- A) Facilitara el diseño, elaboración, modificación, edición, emisión y aplicación de cuestionarios precodificados.
- B) Diera la posibilidad de ordenar, clasificar, seleccionar, almacenar y modificar tanto los reactivos individualmente como los cuestionarios.
- C) Ofreciera la manera de elaborar reactivos a la medida de la necesidad.
- D) Permitiera la aplicación de cuestionario en forma interactiva.
- E) Realizara la calificación de los cuestionarios interactivos en forma automática.



CONCLUSIONES Y RECOMENDACIONES

13

Vivimos en una época de constante evolución, en la que como profesionistas capaces e instruidos estamos comprometidos a participar en la continua búsqueda del perfeccionamiento de nuestro entorno. Necesidades y problemas, hay muchos, sin embargo no debemos quedarnos en la posición de críticos y espectadores, sino aceptar el reto que estas deficiencias nos plantean y proponer nuevas alternativas para solucionarlos. Debemos ser promotores del cambio.

Es por eso que el Sistema Generador de Exámenes y Encuestas nace como una propuesta para resolver un problema concreto: simplificar la tarea de diseñar, elaborar y aplicar cuestionarios. La consigna era desarrollar una herramienta computacional que:

A) Facilitara el diseño, elaboración, modificación, edición, emisión y aplicación de cuestionarios precodificados.

B) Diera la posibilidad de ordenar, clasificar, seleccionar, almacenar y modificar tanto los reactivos individualmente como los cuestionarios.

C) Ofreciera la manera de elaborar reactivos a la medida de la necesidad.

D) Permitiera la aplicación de cuestionario en forma interactiva.

E) Realizara la calificación de los cuestionarios interactivos en forma automática.



Una vez terminado el sistema se llegó a las siguientes conclusiones:

El sistema es propositivo e innovador.

Satisface una necesidad real de usuarios reales. Propone una nueva metodología para la elaboración de cuestionarios y ofrece herramientas para hacerlo.

El sistema es de aplicación general.

La aplicación del sistema no está restringida a una área o problema específico, sin embargo, también hay que reconocer que el ámbito en el que se puede obtener mayor provecho es el académico.

Proporciona una interfaz intuitiva y de fácil manejo.

El diseño de las pantallas fue hecho pensando en la comodidad de los usuarios.

Ofrece un manejo eficiente de la información almacenada.

Los procesos de definición, formación y aplicación de cuestionarios están cuidadosamente diseñados para el adecuado funcionamiento del sistema.

Cumple con los objetivos planteados al principio del proyecto.

Con la evaluación final del sistema se puede decir que el sistema realiza las actividades que se esperan de él, de tal forma que se puede afirmar que la meta del proyecto fue alcanzada satisfactoriamente.

Actualmente se ha cubierto la primer meta señalada: el desarrollo e implantación del sistema, sin embargo ahora el reto es evolucionar, seguir mejorando el funcionamiento del sistema, recibir las nuevas propuestas y trazar una trayectoria ascendente en el uso del sistema, de tal manera que cada vez más usuarios aprovechen sus ventajas, creando un sistema en continuo perfeccionamiento.

Finalmente se recomienda a los usuarios den a conocer sus comentarios acerca del uso y operación del sistema, para de esta manera tomar en cuenta sus opiniones en versiones posteriores.



BIBLIOGRAFIA

DOCTRINA, METODOLOGIA Y EVALUACION PEDAGÓGICAS.

Víctor Hugo Bolaños Martínez

Colnamaep

México, 1981

A PRACTICAL INTRODUCTION TO MEASUREMENT AND EVALUATION

H.H. Remmers, N.L.Gage, J. Francis Rummel

Second Edition

1966

VADEMECUM DEL MAESTRO DE ESCUELA PRIMARIA

José de Jesús Velázquez Sánchez

Ed. Porrúa

México, 1974

Colnamaep

México, 1981

LA CIENCIA DE LA EDUCACIÓN

Francisco Larroyo

Ed. Porrúa

México 1971

PSICOTÉCNICA PEDAGÓGICA

DGETI

Junio 1984



EL PROCESO DE LA INVESTIGACIÓN CIENTÍFICA

Mario Tamayo Y Tamayo

Fundamentos de investigación edición

Editorial LIMUSA.

México, D.F. 1986

ADMINISTRACIÓN DE DATOS Y ARCHIVOS POR COMPUTADORA

CÓMO HACER UNA INVESTIGACIÓN

Noe Pérez Ávila Primera edición

Editorial EDIPSA. Primera edición

México, D.F. 1989

SOFTWARE DE SISTEMAS. INTRODUCCIÓN A LA PROGRAMACIÓN DE

EL MÉTODO CIENTÍFICO

Arturo Roseablueth

Editorial Mc. Graw Hill. Primera edición

México, D.F. 1985

CÓMO HACER UNA TESIS

Umberto Eco

Editorial GEDISA

Barcelona, España, 1994

FOXPRO 2.6 PARA MS-DOS Y PARA WINDOWS A SU ALCANCE

ANÁLISIS Y DISEÑO DE SISTEMAS DE INFORMACIÓN

James A. Senn Graw Hill. Primera edición

Editorial Mc. Graw Hill. Segunda edición

México, D.F. 1994

APLICACIONES DE GESTIONES CON FOXPRO.

Bill Chambers

Editorial Anaya Multimedia América. Tercera edición

México, D.F. 1994



INGENIERÍA DEL SOFTWARE, UN ENFOQUE PRÁCTICO

Roger S. Presman

Editorial Mc. Graw Hill. Tercera edición

México, D.F. 1993

ADMINISTRACIÓN DE DATOS Y ARCHIVOS POR COMPUTADORA

Antonio Arranz Ramonet

Editorial LIMUSA. Primera edición

México, D.F. 1984

SOFTWARE DE SISTEMAS. INTRODUCCIÓN A LA PROGRAMACIÓN DE SISTEMAS

Leland L. Beck

Editorial Addison-Wesley Iberoamericana

México, D.F. 1988

INSIDE FOXPRO 2.5 FOR DOS

Bob Grommes

New Riders Publishing

FOXPRO 2.5 PARA MS-DOS Y PARA WINDOWS A SU ALCANCE

Jose Javier Garcia Badel

Editorial Mc. Graw Hill. Primera edición

México, D.F. 1994

APLICACIONES DE GESTIONES CON FOXPRO.

Bill Chambers

Editorial Anaya Multimedia América. Tercera edición

México, D.F. 1994



EL SISTEMA GENERADOR DE EVALUACIONES Y ENCUESTAS.....1

REQUERIMIENTOS DE HARDWARE.....1

CARACTERÍSTICAS.....1

MANUAL

EL MENU PRINCIPAL.....2

DEL

LAS OPCIONES DEL MENU PRINCIPAL.....3

OPCIONES DEL MENU SISTEMA.....5

USUARIO

Resumen (F2).....5

Grupos (F3).....8

Construcción de Guías (F4).....9

Exámenes impresos (F6) y Guías impresas (F7).....11

Exámenes Interactivos (F8).....11

Calificar interactivos (F9).....12

Reportes.....12

Respaldos.....13

Selección de archivos.....14

Extensiones del SIGEE.....15

Teclas rápidas.....15

EL SISTEMA GENERADOR DE EVALUACIONES Y ENCUESTAS.....	1
REQUERIMIENTOS DE HARDWARE.....	1
CARACTERISTICAS.....	1
EL MENU PRINCIPAL.....	2
LAS OPCIONES DEL MENU PRINCIPAL.....	3
OPCIONES DEL MENU SISTEMA.....	5
<i>Reactivos. (F2).....</i>	<i>5</i>
<i>Memoria RAM.....</i>	<i>2 MB</i>
<i>Grupos. (F3).....</i>	<i>8</i>
<i>Construir guión (F4).....</i>	<i>9</i>
<i>Exámenes impresos (F6) y Guías impresas (F7).....</i>	<i>11</i>
<i>Exámenes Interactivos (F8).....</i>	<i>11</i>
<i>Calificar interactivos (F9).....</i>	<i>12</i>
<i>Reportes.....</i>	<i>12</i>
<i>Respaldos.....</i>	<i>13</i>
<i>Selección de archivos.....</i>	<i>14</i>
<i>Extensiones del SIGEE.....</i>	<i>15</i>
<i>Teclas rápidas.....</i>	<i>15</i>

El Sistema Generador de Evaluaciones y Encuestas.

El control del sistema se hace por medio de un menú que contiene las operaciones principales del sistema. El acceso a las opciones puede ser por medio del teclado o del mouse. El Sistema Generador de Exámenes y Encuestas es una herramienta computacional diseñada para resolver las tareas básicas de la elaboración de cuestionarios como: diseñar reactivos, formar, imprimir, aplicar y evaluar cuestionarios, y algunas otras tareas relacionadas.

La aplicación del SIGEE está principalmente enfocada a los maestros e investigadores, sin embargo, no está restringido exclusivamente a estas áreas, sino por el contrario, puede dar respuesta a cualquier necesidad de elaboración de cuestionarios precodificados.

Requerimientos mínimos de hardware.

Procesador	Intel 386 SX o mejor
Memoria RAM	2 MB
HD	2 MB

Características.

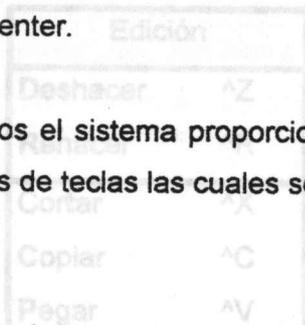
- Sistema diseñado para trabajar en ambiente de MS-DOS.
- Permite el trabajo en un sistema de red.
- Interfaz controlable tanto con el teclado como con el ratón.
- Interfaz intuitiva y sencilla, a la que el usuario se acostumbrará fácilmente.
- Procesos consistentes entre ellos y con los estándares de computación, lo cual hace que sea más sencillo de aprender para el usuario.
- La orientación a los eventos en el diseño de la interfaz permite que el usuario se sienta cómodo con el sistema, ya que es éste el que se adapta a sus necesidades y no viceversa.

El menú principal. menú principal.

El control del sistema se hace por medio de un menú que contiene las operaciones principales del sistema. El acceso a las opciones puede ser por medio del teclado o del ratón.

El menú Edición como su nombre lo indica contiene las operaciones de edición de texto

Si se hace uso del teclado, el menú se activa con la tecla ALT y la letra que está resaltada. Las opciones de cada menú se activan con la letra resaltada o por medio de las flechas del teclado y la tecla enter.

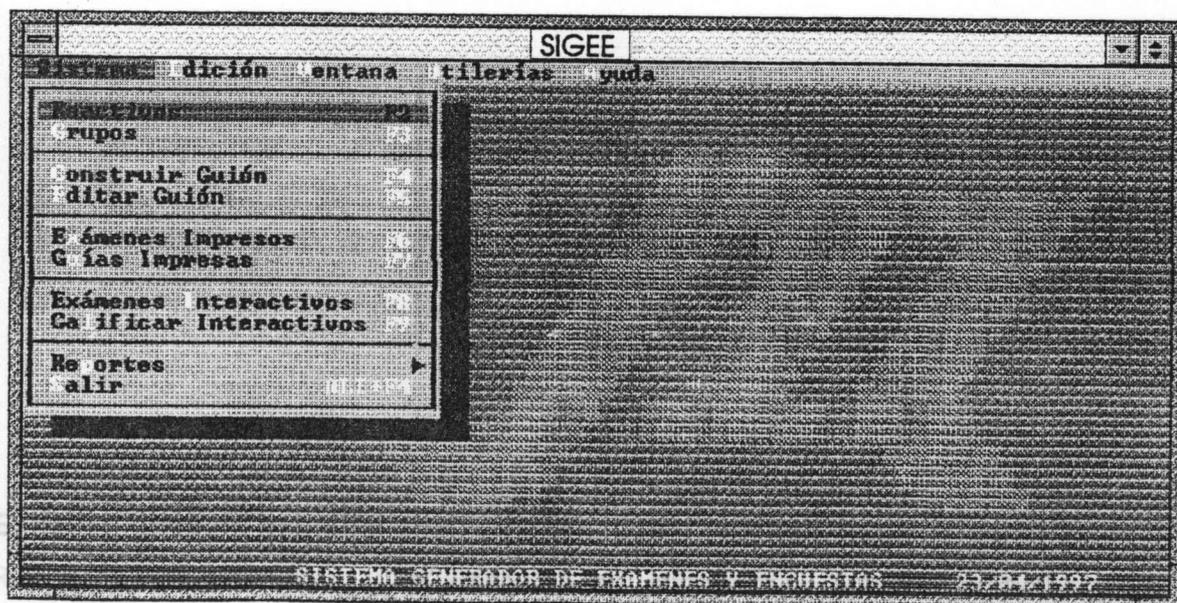


Para los usuarios más avanzados el sistema proporciona accesos rápidos o atajos por medio de algunas combinaciones de teclas las cuales se muestran junto con cada opción del menú.

Si se hace uso del ratón bastará con señalar la opción deseada y pulsar el botón izquierdo, para activarla.

El menú Ventanas da el control al usuario del tamaño y posición de las ventanas permitiéndole en cierta medida personalizar el ambiente en el que trabaja.

A continuación se muestra el aspecto del menú principal del sistema.



funciones de una calculadora.

Las opciones del menú principal.

El menú **Sistema** contiene las opciones que controlan las tareas principales del sistema y las cuales se describirán detalladamente en las siguientes secciones.

El menú **Edición** como su nombre lo indica contiene las operaciones de edición de texto las cuales son:

Edición	
Deshacer	^Z
Rehacer	^R
Cortar	^X
Copiar	^C
Pegar	^V

El menú **Ventana** da el control al usuario del tamaño y posición de las ventanas permitiéndole en cierta medida personalizar el ambiente en el que trabaja.

Ventana	
Mover	^F7
Tamaño	^F8
Maximizar	^F10
Minimizar	^F9

El menú **Utilerías** contiene dos funciones útiles para los usuarios como lo es la Regeneración de índices y el Respaldo de información. Además proporciona las funciones de una calculadora.

Opciones del menú Sistema.

Utilerías
Restaurar índices
Respaldos
Calculadora

Reactivos (F2)

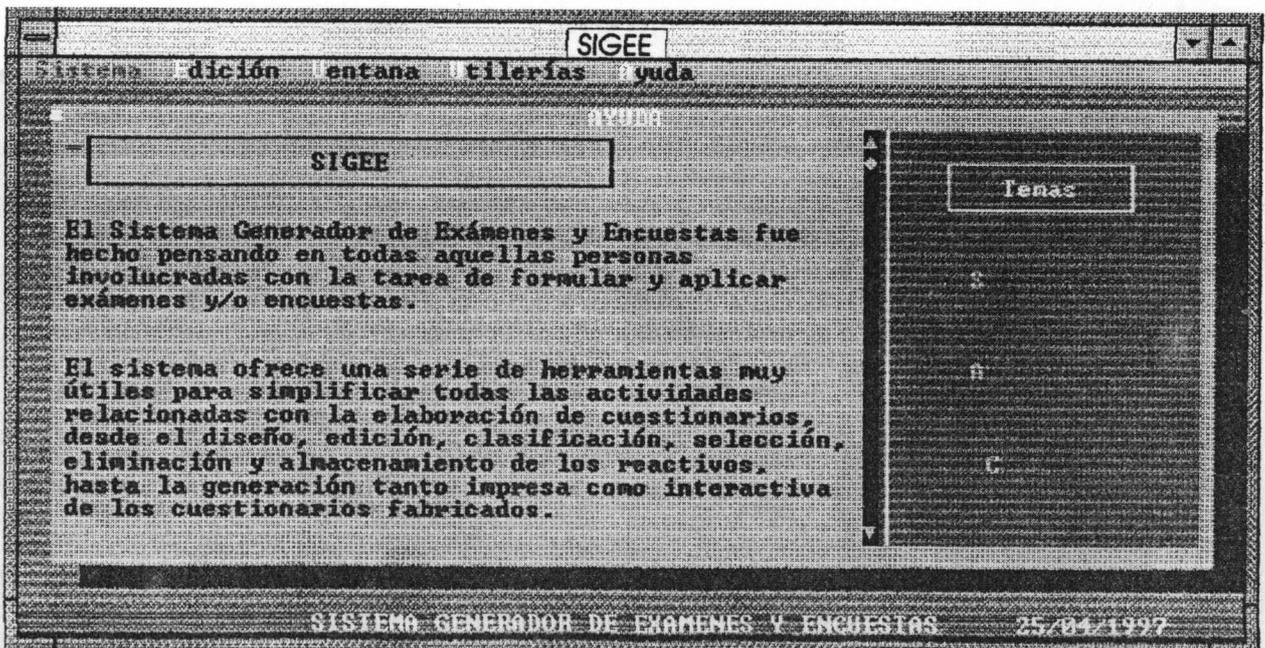
Este módulo tiene como función principal la captura de los reactivos.

Las clases de reactivos que soporta el sistema son las siguientes:

La **Ayuda** es accesible por medio del menú y por medio de la tecla F1. Esta opción es contextual, es decir que se proporcionará la ayuda dependiendo de el módulo que en ese momento se esté ejecutando. Por otra parte se muestra un índice de temas con el cual se podrá navegar en los tópicos de la ayuda según la información que el usuario requiera.

La pantalla que presenta la ayuda es la siguiente:

Cada una de estas clases tiene su propia ficha de captura, que será diferente de las demás dependiendo de la estructura particular de cada reactivo, sin embargo conservando la misma filosofía de trabajo entre ellas.



Opciones del menú Sistema.

Reactivos. (F2)

Este módulo tiene como función principal la captura de los reactivos.

Las clases de reactivos que soporta el sistema son las siguientes:

Dicotómicos
Opción múltiple respuesta única
Complementación sugerida
Opción múltiple respuesta doble
Ordenamiento o jerarquización
Relación o correspondencia.

Cada una de estas clases tiene su propia ficha de captura, que será diferente de las demás dependiendo de la estructura particular de cada reactivo, sin embargo conservando la misma filosofía de trabajo entre ellas.



The screenshot shows a window titled 'SIGEE' with a menu bar containing 'Sistema', 'Reactivos', 'Edición', 'Ventana', 'Herramientas', 'Ayuda', and 'Reactivos'. The main area displays a table with the following data:

Clave	Elabora	Escuela	Materia
ABRIL-97	BEISI	MOTO	MATEMATI
D2-01-03	PABIOLAG	UUAQ	INFORMAT
EGR-0001	PABIOLAG	UUAQ	INFORMAT
PIRSTCER	TITA	UUAQ	HISTORIA
MS-DOS	PABIOLAG	UUAQ	INFORMAT
SEEC-001	GARP-030	UUAQ	LABORAL
SOCIALES	ANANI	IESVAQ	
TELESEC.	ESTEBAN	UUAQ	FISICA I

At the bottom of the window, it reads 'SISTEMA GENERADOR DE EXAMENES Y ENCUESTAS' and '23/04/1997'.

La figura anterior corresponde al browse o examinador de los reactivos. Éste presenta la lista de reactivos clasificados según la clase de los mismos.

Las opciones que corresponden a los reactivos son:

Reactivos	
Agregar	Alt+F5
Editar	Alt+F6
Eliminar	Alt+F7
Clases	➤
Cerrar	Ctrl+F4

Una vez activado el browse o examinador del catálogo se podrá seleccionar la clave deseada, ya sea por medio del menú Grupos que en ese momento se ha activado o por medio de la barra espaciadora. (Para mayores detalles ver módulo de grupos)

Los campos que se capturan en todas las pantallas son:

Clave	Es un identificador único para los reactivos dentro de cada clase.
Profesor	Clave del profesor que elabora el reactivo
Escuela	Clave de la escuela o institución para la cual se elabora el reactivo.
Materia	Materia a la que pertenece.
Tema	Tema con el que está relacionado
Grado	Grado al que se aplicará el reactivo
Libre_1	Libre para los usuarios
Libre_2	Libre para los usuarios
Valor Absoluto	Valor Absoluto del reactivo
Valor Relativo	Valor Relativo del reactivo
Minutos	Tiempo en que se resuelve el reactivo
Premisa	Enunciado o pregunta
Respuestas	Repuestas y distractores

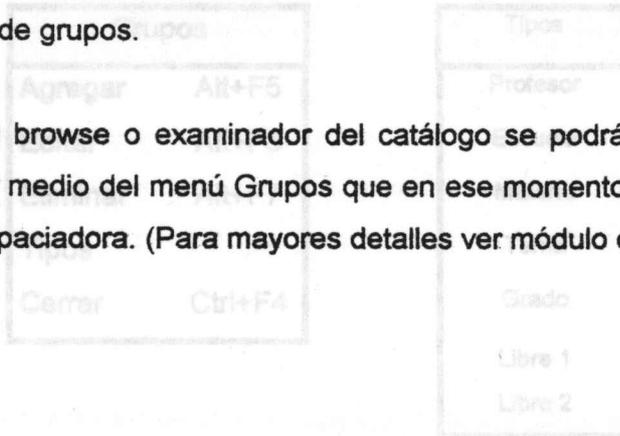
Los campos que no pueden quedar vacíos son:

Clave, Valor Absoluto, Valor Relativo, Minutos, Premisa, Respuestas.

Todos los demás podrán estar vacíos o bien contener una clave que debe existir en el catálogo de claves.

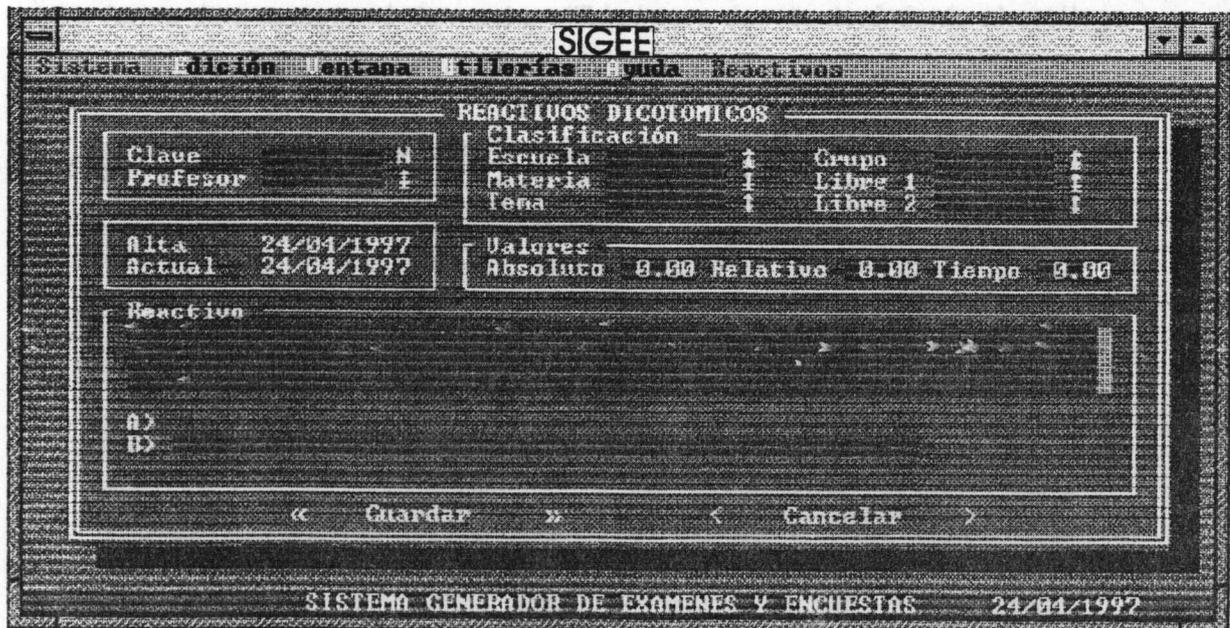
El objetivo de este módulo es facilitar las capturas de los reactivos. Como ya se mencionó anteriormente los reactivos están formados por una serie de campos que La captura de los campos del catálogo de claves puede realizarse de dos formas, una recurriendo al catálogo de las claves y otra tecleando directamente la clave en el lugar destinado para ello. Si se escoge la primera opción podrá hacerse lo siguiente:

1. Presionar enter dentro del campo para activar el catálogo de grupos.
2. Presionar el botón que está al lado de cada campo para activar el catálogo de grupos.



Una vez activado el browse o examinador del catálogo se podrá seleccionar la clave deseada, ya sea por medio del menú Grupos que en ese momento se ha activado o por medio de la barra espaciadora. (Para mayores detalles ver módulo de grupos)

Este módulo tiene dos caminos de acceso: por medio del menú principal o en aquellas pantallas en las cuales se requiere información de este catálogo a través de un botón. La



Grupos. (F3) Tipos (F4).

La finalidad primordial de este módulo es facilitar las capturas de los reactivos. Como ya se mencionó anteriormente los reactivos están formados por una serie de campos que definen su estructura y es precisamente el módulo de grupos el que se encarga de controlar estos campos. Todas las claves se encuentran almacenadas en una base de datos común y son distinguidos por el grupo al que pertenecen de esta manera cuando se accede a esta base de datos se hace por tipos.

Las opciones de este módulo son: también es posible editario. La pantalla que se presenta para ambos casos es la siguiente:

Grupos		Tipos	
Agregar	Alt+F5	Profesor	
Editar	Alt+F6	Escuela	
Eliminar	Alt+F7	Materia	
Tipos	➤	Tema	
Cerrar	Ctrl+F4	Grado	
		Libre 1	
		Libre 2	

Este módulo tiene dos caminos de acceso: por medio del menú principal o en aquellas pantallas en las cuales se requiere información de este catálogo a través de un botón. La primer forma de acceso será de mucha utilidad para dar mantenimiento al catálogo y en la segunda para proporcionar información requerida por el sistema.



The screenshot shows a terminal window titled 'SIGEE' with a menu at the top: 'Sistema', 'dicción', 'entana', 'clorias', 'yuda'. The main content is a table titled 'CATALOGO DE PROFESORES' with two columns: 'Clave' and 'Nombre'. The table lists the following data:

Clave	Nombre
ANAHÍ	ANAHÍ JIMENEZ MARTIN MORELOS
BEISI	BETSADA MORALES
CARLOS	CARLOS ESPEJEL
CONDE	CONDE DE ALMAY
ELIAS	ELIAS DOMINGUEZ MALDONADO
ESTEBAN	ESTEBAN CABALLERO
FABIOLAC	FABIOLA GARCIA RANCEL
FAUSTO	FAUSTO GAMINO RAMIREZ
JUANITA	JUANITA MONROY
LULA	LOURDES RANGEL LOPEZ
MARIA	MARIA LA DEL BARRIO
MARIANO	PROFRE MARIANO MORALES
MARTIN	STUEVE MARTIN
MINI	MINUERA
NEREIDA	NEREIDA

At the bottom of the terminal window, it says 'SEDESA - GENERADOR DE EXAMENES Y ENCUESTAS' and '24/06/2007'.

Construir guión (F4).

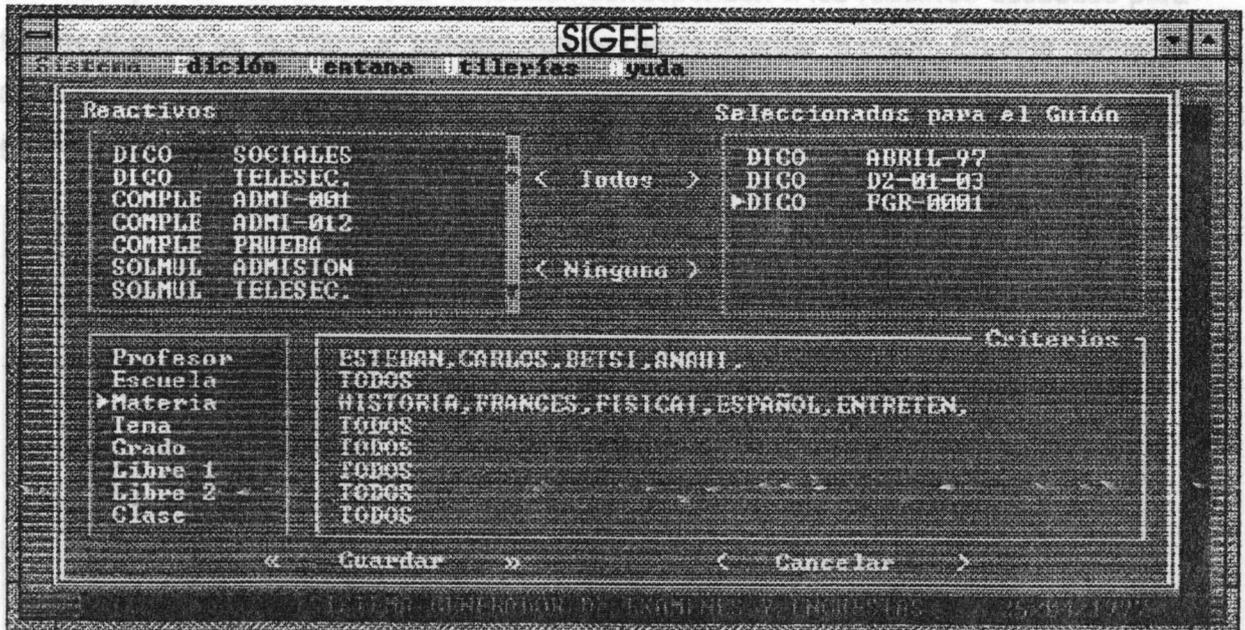
Llamamos guión al conjunto de reactivos que serán utilizados para la formación de un examen o encuesta ya sea impresa o interactiva.

La formación del guión es una de las herramientas más importantes del sistema, la cual podrá utilizarse una vez que han sido dados de alta algún número de reactivos.

Además de poder crear un guión también es posible editarlo. La pantalla que se presenta para ambos casos es la siguiente:

Por otra parte en la parte inferior de la pantalla se presentan los campos del catálogo de claves con los cuales se determinarán los criterios de selección de reactivos. Cada campo es un botón que conduce a una nueva pantalla de criterios en la cual se podrá marcar aquellos campos que se desea sean incluidos o excluidos de la selección. Dicha selección actuará como un filtro para los reactivos que serán presentados en la pantalla principal de la formación del guión.

Una vez que los reactivos de la selección corresponden a los reactivos deseados para



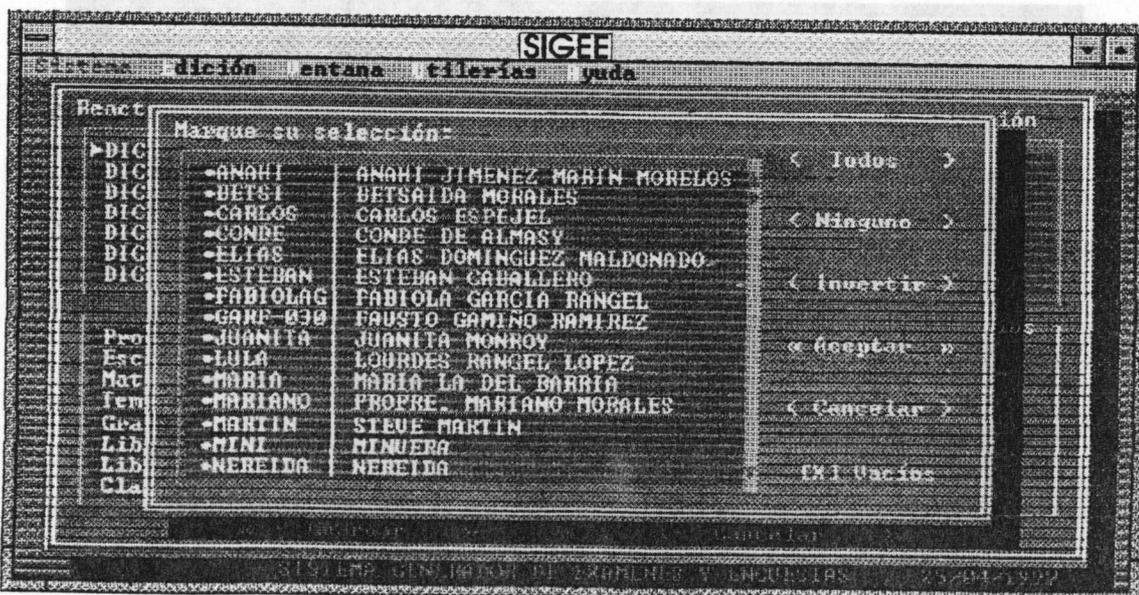
En caso de que se trate de la creación de un guión nuevo la lista que tiene el nombre de reactivos presenta información de todos los reactivos que existen actualmente. La información presentada es el tipo de reactivo al que pertenece y la clave que lo identifica. Si el caso es la modificación de un guión existente esta lista presenta los reactivos que pertenecen al guión que se está modificando.

Para realizar dicha selección se hace uso de una rutina que da servicio a todos. La lista llamada Selección de reactivos inicialmente está vacía, en ella se irán acumulando los reactivos que se vayan seleccionando de la lista de reactivos. (partado de "Selección de archivos".)

Por otra parte en la parte inferior de la pantalla se presentan los campos del catálogo de claves con los cuales se determinarán los criterios de selección de reactivos. Cada campo es un botón que conduce a una nueva pantalla de criterios en la cual se podrá marcar aquellos campos que se desea sean incluidos o excluidos de la selección. Dicha selección actuará como un filtro para los reactivos que serán presentados en la pantalla principal de la formación del guión.

Exámenes Interactivos (F3)

Una vez que los reactivos de la selección corresponden a los reactivos deseados para formar el guión se le indica al sistema que se desea guardar el guión. En este momento aparece una nueva pantalla en la cual se introducirán los datos que son necesarios para guardar el guión.



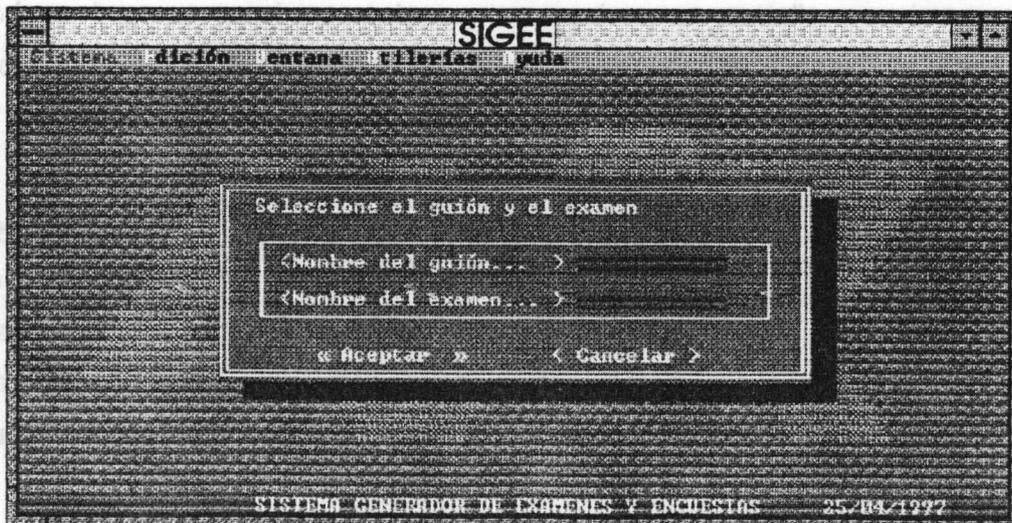
Exámenes impresos (F6) y Guías impresas (F7).

Como ya se mencionó anteriormente, los exámenes ya sean impresos o interactivos son contruidos a partir de un guión, en el cual se encuentra la referencia a los reactivos que se quiere incluir en el examen. Por esta razón estos módulos solicitan la selección de un guión. Para realizar dicha selección se hace uso de una rutina que da servicio a todos los módulos y que presenta diferentes opciones según el tipo de archivo que se esté seleccionando. (Para tener más detalles a cerca de esta rutina consultar el apartado de "Selección de archivos".)

Una vez que es seleccionado el guión ya sea para la guía o para el examen el sistema detectará si existe un dispositivo de impresión listo para trabajar de no ser así desplegará un mensaje de advertencia para reintentar o cancelar el proceso. (Consultar anexo de reportes)

Exámenes Interactivos (F8)

El módulo de exámenes interactivos presenta una pantalla para solicitar el nombre del guión que se va a emplear para elaborar el examen y el nombre del archivo en el que se van a guardar las respuestas del alumno y la disposición del examen.



Una vez hecho lo anterior se introducen los datos del alumno y el examen puede ser contestado.

Los reactivos pueden ser contestados con el teclado o con el ratón y el alumno tendrá la libertad de contestar en el orden que mejor le convenga.

El sistema despliega el mayor número de preguntas posibles por cada pantalla y se podrá avanzar entre las pantallas por medio de las teclas de Avance y Retroceso de página, o bien con los botones de avance y retroceso de la ventana.

El examen se termina con la tecla ESC o con el botón de cierre de la ventana, en este momento el sistema informa de el número de reactivos contestados y se ofrece la posibilidad de regresar a contestar aquellas respuestas que quedaron vacías o bien terminar el examen.

Calificar interactivos (F9).

Una vez que uno o más exámenes interactivos han sido aplicados existe la posibilidad de realizar una calificación automática de los mismos. Bastará con seleccionar el examen que se desea calificar y especificar el destino del reporte que puede ser en archivo, en cuyo caso tendrá una extensión .DBC (Consultar tabla de extensiones) y podrá ser impreso en el momento que se requiera, en pantalla o en impresora. (Para conocer un ejemplo de un reporte de calificaciones consultar el anexo de reportes).

Reportes .

Son cuatro los reportes que presenta el sistema:

Catálogo.

Este reporte contiene la información de las claves y sus descripciones existentes actualmente en el catálogo de claves.

Guión.

Es una vía alternativa de imprimir guiones o exámenes.

Guía

Es una vía alternativa para imprimir guías de exámenes.

Calificaciones

Imprime archivos de calificaciones generados en el módulo de Calificar exámenes interactivos (*.DBE).

Respaldos

Este módulo es una opción del módulo de utilerías. Permite realizar el respaldo o copia de cualquier tipo de archivo existente dentro del sistema, dicha copia puede ser en otra ruta del disco duro o bien en otra unidad de disco flexible.

Para los guiones las opciones habilitadas son:

Seleccionar	Quando el módulo requiere la selección de un guión
Editar	Editar el guión seleccionado
Nuevo	Crear un nuevo guión
Información	Obtener información del guión seleccionado

Para los archivos de exámenes:

Seleccionar	Quando el módulo requiere la selección de un examen
Nuevo	Crear un nuevo examen



Selección de archivos.

La rutina de selección de archivos es muy útil para el sistema, ya que es llamada desde varios módulos. Consiste de una lista de varios archivos comunes según sea el módulo que lo llama, por ejemplo si es llamado por el módulo de editar guiones los archivos presentados en la lista tendrán extensión .DBG. si es llamado por el módulo de imprimir calificaciones tendrán extensión .DBC etc.

Así mismo dependiendo de el tipo de archivos presentados estarán habilitadas o deshabilitadas las opciones de esta rutina.

Para los guiones las opciones habilitadas son:

Seleccionar	Cuando el módulo requiere la selección de un guión
Editar	Editar el guión seleccionado
Nuevo	Crear un nuevo guión
Información	Obtener información del guión seleccionado

Para los archivos de exámenes:

Seleccionar	Cuando el módulo requiere la selección de un examen
Nuevo	Crear un nuevo examen

Para los archivos de calificaciones:

Seleccionar	Seleccionar un archivo de calificaciones
-------------	--



Extensiones del SIGEE

Extensión	Tipo de Archivo
.DBF	Bases de datos del sistema
.DBG	Guiones
.DBE	Exámenes
.DBC	Calificaciones

Teclas rápidas

F1	Ayuda	Ctrl+W	Aceptar
F2	Reactivos	ESC	Cancelar
F3	Grupos	Ctrl+Z	Deshacer
F4	Construir guión	Ctrl+R	Rehacer
F5	Editar guión	Ctrl+X	Cortar
F6	Exámenes impresos	Ctrl+C	Copiar
F7	Guías impresas	Ctrl+V	Pegar
F8	Exámenes interactivos	Ctrl+F7	Mover ventana
F9	Calificar interactivos	Alt+1	R. dicotómicos
Ctrl+F8	Tamaño ventana	Alt+2	R. Opción única
Ctrl+F9	Maximizar ventana	Alt+3	R. Complementación
Ctrl+F10	Minimizar ventana	Alt+4	R. Opción Múltiple
Alt+F7	Eliminar (claves/reactivos)	Alt+5	R. Ordenamiento
Ctrl+F4	Cerrar browse	Alt+6	R. Relación
Alt+F4	Salir del sistema		

CLAVE	NOMBRE
DIFICIL1	PRIMER PARCIAL
DIFICIL2	SEGUNDO PARCIAL
FACILITO	PRIMER NIVEL
PROPEDEU	PREPARATORIA PROPEDEUTICA
TECNOLOG	PREPARATORIA TECNOLOGICA
AVANZADO	NIVEL AVANZADO DE INGLES
BASICOS	NIVEL BASICO DE INGLES
INGLES	CURSO DE INGLES
INTERMED	NIVEL INTERMEDIO DE INGLES
NUEVOS	REACTIVOS DE LA NUEVA GENERACION
ANAH001B	COLEGIO ANAHUAC A.C.
C.B.T.S	COLEGIO DE BACHILLERES
CEM	CENRO DE ESTUDIOS ESCOLARES DE MICHOACAN
MIRAFLO	COLEGIO MIRAFLORES
MOTO	INSTITUTO MOTOLINIA
NOBELMOR	INSTITUTO NOBEL DE MORELIA
PRUEBA	CLASE DE PRUEBAS
RECTOR	ESCUELA PRIMARIA RECTOR NIGUEL HIDALGO
SALESIAN	COLEGIO SALESIANO DE MORELIA
TEC. 3	ESCUELA SECUNDARIA TECNICA NOMBRE 3
TEC.65	ESCUELA SECUNDARIA TECNICA 65
TEC.82	ESCUELA SECUNDARIA TECNICA 82
TELESECU	PROGRAMA DE TELESECUNDARIAE
UVAQ	UNIVERSIDAD VASCO DE QUIROGA A.C.
VALLA	INSTITUTO VALLADOLID
3F. PREPA	TERCER AÑO F DE PREPARATORIA
ADMINIST	LICENCIATURA EN SISTEMAS COMPUTARIZADOS
BENJAMIN	BENJAMIN MONTES JARAMILLO
CYALSEMA	PRIMER SEMESTRE GRUPO A
CYALSEMB	PRIMER SEMESTRE GRUPO B
NUEVO	CLASES NUEVAS
SEGUNDO	SEGUNDO AÑO DE PREPARATORIA
SISTEMAS	SEPTIMO SEMESTRE DE SISTEMAS COMPUTARIZADOS
TERCERO	TERCER AÑO PREPARATORIA TECNOLIGICA
ADMINIST	TEORIA ADMINISTRATIVA
AUDITOR	AUDITORIA ADMINISTRATIVA
C. SOCIA	CIENCIAS SOCIALES I
C.NATURA	CIENCIAS NATURALES
CIVIL	DERECHO CIVIL
ENTRETEN	ENTRETENIMIENTO
ESPAÑOL	ESPAÑOL
FISICA I	FISICA PRIMER CURSO
FRANCES	CURSO INTENSIVO DE FRANCES
GEOGRAFI	GEOGRAFIA MUNDIAL
HISTORIA	HISTORIA UNIVERSAL
INFORMAT	INFORMATICA I

REPORTES

SISTEMA GENERADOR DE EXAMENES Y ENCUESTAS
 REPORTE DEL CATALOGO DE CLAVES

FECHA 28/04/1997

TIPO	CLAVE	NOMBRE	
LIBRE_1	DIFICIL1	PRIMER PARCIAL	
	DIFICIL2	SEGUNDO PARCIAL	
	FACILITO	PRIMER NIVEL	
LIBRE_2	PROPEDEU	PREPARATORIA PROPEDEUTICA	
	TECNOLOG	PREPARATORIA TECNOLOGICA	
	AVANZADO	NIVEL AVANZADO DE INGLES	
ESCUELA	BASICOS	NIVEL BASICO DE INGLES	
	INGLES	CURSO DE INGLES	
	INTERMED	NIVEL INTERMEDIO DE INGLES	
	NUEVOS	REACTIVOS DE LA NUEVA GENERACION	
	ANAH001B	COLEGIO ANAHUAC A.C.	
	C.B.T.S	COLEGIO DE BACHILLERES	
	CEEM	CENTRO DE ESTUDIOS ESCOLARES DE MICHOACAN	
	CEM	CENTRO ESCOLAR MICHOACANO	
	IESVAQ	INSTITUTO DE ESTUDIOS SUPERIORES VASCO DE QUIROGA	
	LASROSAS	CONSERVATORIO DE LAS ROSAS	
	MEDICINA	FACULTAD DE MEDICINA DE LA UNIVERSIDAD MICHOACANA	
	MIRAFLOR	COLEGIO MIRAFLORES	
	MOTO	INSTITUTO MOTOLINIA	
	NOBELMOR	INSTITUTO NOBEL DE MORELIA	
	PRUEBA	CLASE DE PRUEBAS	
	RECTOR	ESCUELA PRIMARIA RECTOR MIGUEL HIDALGO	
	SALESIAN	COLEGIO SALESIANO DE MORELIA	
	TEC. 3	ESCUELA SECUNDARIA TECNICA N0MERO 3	
	TEC.65	ESCUELA SECUNDARIA TECNICA 65	
	TEC.82	ESCUELA SECUNDARIA TECNICA 82	
	TELESECU	PROGRAMA DE TELESECUNDARIAE	
	UVAQ	UNIVERSIDAD VASCO DE QUIROGA A.C.	
	VALLA	INSTITUTO VALLADOLID	
	GRADO	3F.PREPA	TERCER AÑO F DE PREPARATORIA
		ADMINIST	LICENCIATURA EN SISTEMAS COMPUTARIZADOS
		BENJAMIN	BENJAMIN MONTES JARAMILLO
		CYA1SEMA	PRIMER SEMESTRE GRUPO A
CYA1SEMB		PRIMER SEMESTRE GRUPO B	
NUEVO		CLASES NUEVAS	
SEGUNDO		SEGUNDO AÑO DE PREPARATORIA	
SISTEMAS		SEPTIMO SEMESTRE DE SISTEMAS COMPUTARIZADOS	
TERCERO		TERCER AÑO PREPARATORIA TECNOLOGICA	
MATERIA		ADMINIST	TEORIA ADMINISTRATIVA
	AUDITOR	AUDITORIA ADMINISTRATIVA	
	C. SOCIA	CIENCIAS SOCIALES I	
	C.NATURA	CIENCIAS NATURALES	
	CIVIL	DERECHO CIVIL	
	ENTRETEN	ENTRETENIMIENTO	
	ESPAÑOL	ESPAÑOL	
	FISICA I	FISICA PRIMER CURSO	
	FRANCES	CURSO INTENSIVO DE FRANCES	
	GEOGRAFI	GEOGRAFIA MUNDIAL	
	HISTORIA	HISTORIA UNIVERSAL	
	INFORMAT	INFORMATICA I	

TIPO CLAVE NOMBRE

TIPO	CLAVE	NOMBRE
MATERIA	INGLES	CURSO INTENSIVO DE INGLES
	LABORAL	DERECHO LABORAL
	MATEMATI	MATEMATICAS I
	MUSICA	HISTORIA DE LA MUSICA
ALUMNO	PEDAGOGI	PSICOTECNIA PEDAGOGICA
	QUIMICA	QUIMICA
	RELIGION	EDUCACION RELIGIOSA
	SO-PREPA	SISTEMAS OPERATIVOS
ELABORA	ANAHI	ANAHI JIMENEZ MARIN MORELOS
	BETSI	BETSAIDA MORALES
	CARLOS	CARLOS ESPEJEL
	CONDE	ABRAHAM NEGRETE MURILLO
	ELIAS	ELIAS DOMINGUEZ MALDONADO
	ESTEBAN	ESTEBAN CABALLERO
	FABIOLAG	FABIOLA GARCIA RANGEL
	GARF-030	FAUSTO GAMIÑO RAMIREZ
	JUANITA	JUANITA MONROY
	LULA	LOURDES RANGEL LOPEZ
	MARIA	MARIA LA DEL BARRIA
	MARIANO	PROFRE. MARIANO MORALES
	MARTIN	STEVE MARTIN
	MINI	MINERVA CHAVEZ
	NEREIDA	NEREIDA GONZALEZ T.
	PATY_M	PATRICIA MANRIQUEZ
	PEDRITO	PEDRO MORELOS LEON
	SEBASTIA	SEBASTIAN BALLESTEROS
	TITA	DR. MAURICIO TORRES R.
TEMA	ADMISION	EXAMEN DE ADMISION A LA SECUNDARIA
	BARROCO	AUTORES DE MUSICA BARROCA
	DISEÑO	DISEÑO DE SISTEMAS DE INFORMACION
	ECOLOGIA	ECOLOGIA
	EL AGUA	EL AGUA
	HISTORIA	HISTORIA DE LA COMPUTACION
	IMSS	INSTITUTO MEXICANO DEL SEGURO SOCIAL
	INGLES01	INTRODUCE YOURSELF
	INTRO.	INTRODUCCION A LA MATERIA
	INVENTOS	INVENTOS E INVENTORES
	LABORATO	LABORATORIO
	MAGNETIS	MAGNETISMO
	ORTOGRAF	GRAMATICA Y ORTOGRAFIA
	PROCADMI	PROCESO ADMINISTRATIVO
	RENACIMI	EL RENACIMIENTO EN LA LITERATURA
	SISOP001	SISTEMAS OPERATIVOS
	UNIDAD1	QUEBRADOS
	UNIDAD2	SEGUNDA UNIDAD
	UNIDAD3	TERCERA UNIDAD
	WIN-95	WINDOWS 95

SISTEMA GENERADOR DE EXAMENES Y ENCUESTAS
 REPORTE DE CALIFICACIONES
 ADMIS001

FECHA 10/04/1997

ALUMNO	ABS.	REL.	GRUPO	APLICO	FECHA
Karen Estevez Arredondo Guión -> ADMIS.DBG	12.77	94.08	3F.PREPA	FABIOLAG	10/04/1997
Ana Cecilia Botello Gómez Guión -> ADMIS.DBG	9.63	80.86	3F.PREPA	FABIOLAG	10/04/1997
Marissa Camarena Paz Guión -> ADMIS.DBG	7.48	53.52	3F.PREPA	FABIOLAG	10/04/1997
Ada Melissa Carrillo Rodríguez Guión -> ADMIS.DBG	10.13	86.78	3F.PREPA	FABIOLAG	10/04/1997
Karina Coria Vargas Guión -> ADMIS.DBG	8.18	77.18	3F.PREPA	FABIOLAG	10/04/1997
Domínguez Domínguez Susy Guión -> ADMIS.DBG	11.30	89.77	3F.PREPA	FABIOLAG	10/04/1997
Helvetiella Leal Munguía Guión -> ADMIS.DBG	9.06	76.76	3F.PREPA	FABIOLAG	10/04/1997
Erick Benjamín Negrete Martínez Guión -> ADMIS.DBG	6.33	68.33	3F.PREPA	FABIOLAG	10/04/1997
Ana Paula Pantoja Valero Guión -> ADMIS.DBG	10.51	87.42	3F.PREPA	FABIOLAG	10/04/1997
Abraham Carrillo Aguilar Guión -> ADMIS.DBG	7.70	51.25	3F.PREPA	FABIOLAG	10/04/1997
Celia Esmeralda Ayala Sánchez Guión -> ADMIS.DBG	7.10	50.06	3F.PREPA	FABIOLAG	10/04/1997

- 6.- El artículo 123 de la Constitución Mexicana garantiza una jornada máxima de 8 horas. Por ello el obrero que trabaja más tiempo tendrá derecho a:
- A) Su incorporación al sindicato
 - B) El pago de tiempo extra
 - C) La repartición de utilidades
 - D) Su ingreso al seguro social

SECRETARIA DE EDUCACION EN EL ESTADO
Examen de admisión para secundarias
PERIODO 1997 - 1998

ELABORO B) FELIZ FABIOLA GARCIA RANGEL
INSTITUCION UNIVERSIDAD VASCO DE QUIROGA A.C.
FECHA E) CUI 10/04/1997

- 3.- Estaba baja la neblina en la pista, lo cual obligó al piloto
- 1.- Los quebrados impropios son los que es más grande el numerador que el denominador. ()
A) Verdadero
B) Falso
- 2.- La esposa de Benito Juárez fue: ()
A) Margarita Maza
B) Josefa Ortiz de Domínguez
- 3.- SERRUCHO es a MADERA como: () ()
A) CABLE a LUZ
B) MARTILLO a CLAVO
C) ESTUFA a FUEGO
D) CIMIENTO a CASA
E) ZAPATO a HOMBRE
F) BARNIZ a MUEBLE
- 10.- Son dos obras de Gabriel García Márquez. () ()
- 4.- Un sinónimo de RIÑA es: ()
A) ACUERDO
B) CONTIENDA
C) DISCULPA
D) GROSERIA
- 5.- Una de las causas por las que un alto número de habitantes de las zonas rurales van a establecerse a las ciudades es: () ()
A) Visitar a sus parientes
B) La necesidad de aumentar sus ingresos
C) La insalubridad que existe en el campo
D) La existencia de centros comerciales
E) Aprender las costumbres de la ciudad
- 6.- El artículo 123 de la Constitución Mexicana garantiza una jornada máxima de 8 horas. Por ello el obrero que trabaje más tiempo tendrá derecho a: () ()
A) Su incorporación al sindicato
B) El pago de tiempo extra
C) La repartición de utilidades
D) Su ingreso al seguro social
- 13.- Ordena cronológicamente los siguientes autores: () ()
() CHOPIN
() BEETHOVEN
() LIST
() BACH

7.- En Canadá se habla inglés y francés, por lo tanto su pueblo es porque habla dos idiomas.

- A) LINGÜISTA
- B) FELIZ
- C) BILINGÜE
- D) EXPRESIVO
- E) CULTO
- F) COMUNICATIVO

8.- Estaba baja la neblina en la pista, lo cual obligó al piloto a el avión y así evitar entrar en ella.

- A) EQUILIBRAR
- B) PROTEGER
- C) ACELERAR
- D) CONTROLAR
- E) DESVIAR

9.- Obras de la mitología griega escritas por Homero:

() ()

- A) El cantar de los cantares
- B) El Ramayana
- C) La Odisea
- D) La Eneida
- E) La Iliada

10.- Son dos obras de Gabriel García Márquez.

() ()

- A) Los de abajo
- B) Cien años de soledad
- C) Navidad en las montañas
- D) EL coronel no tiene quien le escriba
- E) Marianela

11.- Las palabras que llevan el acento en la última sílaba se llaman:

() ()

- A) Esdrújulas
- B) Agudas
- C) Sobreesdrújulas
- D) Graves
- E) Agudas

12.- En la evolución se buscan algunas comparaciones entre las estructuras actuales y sus...

() ()

- A) Fósiles
- B) Mutaciones
- C) Nuevas generaciones
- D) Evoluciones

13.- Ordena cronológicamente los siguientes autores:

- () CHOPIN
- () BEETHOVEN
- () LIZT
- () BACH

14.- Ordena en forma ascendente (de menor a mayor), según el número atómico, los siguientes elementos químicos.

- Litio
- Radón
- Helio
- Carbono
- Aluminio
- Berilio
- Sodio
- Hidrógeno

15.- Ordene de menor a mayor las instancias a las cuales deben recurrir los trabajadores en caso de problemas legales

- Nivel Regional
- Nivel Estatal
- Nivel Municipal
- Nivel Nacional

(A)

16.- Ordena cronológicamente las siguientes eras:

- devónico
- pérmico
- arqueozoica
- silúrico
- proterozoica
- cámbrico

(A)

(B)

17.- Relaciona según cada expresión con la palabra que convenga

- | | |
|--|------------------------------|
| <input type="checkbox"/> Entrar con furia | A) Espada |
| <input type="checkbox"/> Espacio estrecho que comunica | B) Falta de vigor o esfuerzo |
| <input type="checkbox"/> Arma de combate | C) Pasillo |
| <input type="checkbox"/> Entra sin tener derecho | D) Irrumpir |
| <input type="checkbox"/> Desaliento | E) Intruso |
| <input type="checkbox"/> Falta de cuidado u orden | F) Conjurar |
| <input type="checkbox"/> Alejar el daño o peligro | |

(B)

5.- Una de las causas por las que un alto número de habitantes de las zonas rurales van a establecerse a las ciudades es:

- A) Visitar a sus parientes
- B) La necesidad de aumentar sus ingresos
- C) La insalubridad que existe en el campo
- D) La existencia de centros comerciales
- E) Aprender las costumbres de la ciudad

(B)

6.- El artículo 123 de la Constitución Mexicana garantiza una jornada máxima de 8 horas. Por ello el obrero que trabaje más tiempo tendrá derecho a:

- A) Su incorporación al sindicato
- B) El pago de tiempo extra
- C) La repartición de utilidades
- D) Su ingreso al seguro social

(B)

SECRETARIA DE EDUCACION EN EL ESTADO
Examen de admisión para secundarias
PERIODO 1997 - 1998

GUIA

ELABORO FABIOLA GARCIA RANGEL
INSTITUCION UNIVERSIDAD VASCO DE QUIROGA A.C.
FECHA 10/04/1997

- 1.- Los quebrados impropios son los que es más grande el numerador que el denominador. (A)
A) Verdadero
B) Falso
- 2.- La esposa de Benito Juárez fue: (A)
A) Margarita Maza
B) Josefa Ortiz de Domínguez
- 3.- SERRUCHO es a MADERA como: (B)
A) CABLE a LUZ
B) MARTILLO a CLAVO
C) ESTUFA a FUEGO
D) CIMIENTO a CASA
E) ZAPATO a HOMBRE
F) BARNIZ a MUEBLE
- 4.- Un sinónimo de RIÑA es: (B)
A) ACUERDO
B) CONTIENDA
C) DISCULPA
D) GROSERIA
- 5.- Una de las causas por las que un alto número de habitantes de las zonas rurales van a establecerse a las ciudades es: (B)
A) Visitar a sus parientes
B) La necesidad de aumentar sus ingresos
C) La insalubridad que existe en el campo
D) La existencia de centros comerciales
E) Aprender las costumbres de la ciudad
- 6.- El artículo 123 de la Constitución Mexicana garantiza una jornada máxima de 8 horas. Por ello el obrero que trabaje más tiempo tendrá derecho a: (B)
A) Su incorporación al sindicato
B) El pago de tiempo extra
C) La repartición de utilidades
D) Su ingreso al seguro social

11.- Ordena cronológicamente los siguientes autores:

- (3) CHOPIN
- (2) BEETHOVEN
- (4) LIET
- (1) BACH

7.- En Canadá se habla inglés y francés, por lo tanto su pueblo es porque habla dos idiomas.

BILINGÜE

-
- A) LINGÜISTA
 - B) FELIZ
 - C) BILINGÜE
 - D) EXPRESIVO
 - E) CULTO
 - F) COMUNICATIVO

8.- Estaba baja la neblina en la pista, lo cual obligó al piloto a el avión y así evitar entrar en ella.

DESVIAR

-
- A) EQUILIBRAR
 - B) PROTEGER
 - C) ACELERAR
 - D) CONTROLAR
 - E) DESVIAR

9.- Obras de la mitología griega escritas por Homero:

(E) (C)

- A) El cantar de los cantares
- B) El Ramayana
- C) La Odisea
- D) La Eneida
- E) La Iliada

10.- Son dos obras de Gabriel García Márquez.

(D) (B)

- A) Los de abajo
- B) Cien años de soledad
- C) Navidad en las montañas
- D) EL coronel no tiene quien le escriba
- E) Marianela

11.- Las palabras que llevan el acento en la última sílaba se llaman:

(B) (E)

- A) Esdrújulas
- B) Agudas
- C) Sobreesdrújulas
- D) Graves
- E) Agudas

12.- En la evolución se buscan algunas comparaciones entre las estructuras actuales y sus...

(A) (B)

- A) Fósiles
- B) Mutaciones
- C) Nuevas generaciones
- D) Evoluciones

13.- Ordena cronológicamente los siguientes autores:

- (3) CHOPIN
- (2) BEETHOVEN
- (4) LIZT
- (1) BACH

14.- Ordena en forma ascendente (de menor a mayor), según el número atómico, los siguientes elementos químicos.

- (3) Litio
- (8) Radón
- (2) Helio
- (5) Carbono
- (7) Aluminio
- (4) Berilio
- (6) Sodio
- (1) Hidrógeno

15.- Ordene de menor a mayor las instancias a las cuales deben recurrir los trabajadores en caso de problemas legales

- (2) Nivel Regional
- (3) Nivel Estatal
- (1) Nivel Municipal
- (4) Nivel Nacional

16.- Ordena cronológicamente las siguientes eras:

- (5) devónico
- (6) pérmico
- (1) arqueozoica
- (4) silúrico
- (2) proterozoica
- (3) cámbrico

17.- Relaciona según cada expresión con la palabra que convenga

- | | |
|-----------------------------------|-------------------------------|
| (D) Entrar con furia | A) Espada |
| (C) Espacio estrecho que comunica | B) Falta de vigor o esfuerzo |
| (A) Arma de combate | C) Pasillo |
| (E) Entra sin tener derecho | D) Irrumpir |
| (B) Desaliento | E) Intruso |
| (*) Falta de cuidado u orden | F) Conjurar |
| (F) Alejar el daño o peligro | |


```

* -----
*          SIGEE.PRG
* Módulo Inicial del Sistema Generador de
* Ex menes y Cuestionarios
* -----
#define Cierto .t
#define Falso .f

PUSH KEY CLEAR
PUSH MENU _Msysmenu

PRIVATE ISigue, cTema, cAyuda
Do Inicia
cTema = "SIGEE"
cAyuda = InstAyuda(cTema)
ISigue = Cierto
DO Main.Mpr
READ
POP MENU _Msysmenu
POP KEY ALL
= InstAyuda(cAyuda)
DO Salida

RETURN

* -----
* Inicia.prg
* Preparación del ambiente, apertura de las bases de datos,
* activación del archivo de procedimientos SIGEEPRO.PRG
* -----
CLEAR
POP KEY ALL
CLOSE DATABASES
ON KEY LABEL ALT+F10 *
CLEAR MACROS
SET SAFETY OFF
SET EXCLUSIVE ON
SET DATE TO YMD
SET CENTURY ON
SET EXACT OFF
SET CONFIRM OFF
SET DELETED ON
SET TALK OFF
SET PROCEDURE TO Sigeepro.prg
SET MEMOWIDTH TO 60
SET NEAR ON

Do Tapiz
Do Prepara With .t

RETURN

* -----
* Sigee.pro
* Archivo de procedimientos
* -----
#define CLASES "LIBRE_1LIBRE_2ESCUELAGRADO
MATERIAELABORATEMA "
#define TIPOS "12EGMPT"
#define REAC_TIPOS "DICOTOMICOS SOLUCION UNICA
COMPLEMENTACION SOLUCION MULTIPLEORDENAMIENTO
RELACION "
#define REAC_ABVTIPOS "DICO
SOLUNICOMPLESOLMULORDENARELACI"
#define Falso .f
#define Cierto .t
#define TipoBAJA "p"

FUNCTION ClaseTipo
PARAMETERS cClase
cClase = At(cClase, TIPOS)
RETURN IIF(cClase > 0, RTrim(SubStr(CLASES, (cClase - 1) * 7 + 1,
7)), "")

FUNCTION GuardaArreglo
PARAMETERS IGuarda
If IGuarda
SAVE TO MEMO Respuestas ALL LIKE a_Resp

```

```

REPLACE RespChk WITH VAL(SYS(2007,Respuestas))
Else
IF Val(Sys(2007,Respuestas)) = RespChk
RESTORE FROM MEMO Respuestas ADDITIVE
ENDIF
EndIf
RETURN

FUNCTION SiNo
PARAMETERS cTexto, nObjeto
PUSH KEY CLEAR
PRIVATE nOpcion, ISI
ISI = Falso
nObjeto = IIF(Type("nObjeto") = "N" And InList(nObjeto, 1, 2),
nObjeto, 2)
DEFINE WINDOW elimina_g FROM INT((SROW()-
6)/2),INT((SCOL()-40)/2) ;
TO INT((SROW()-
6)/2)+5,INT((SCOL()-40)/2)+39 ;
FLOAT NOCLOSE SHADOW NOMINIMIZE
DOUBLE COLOR SCHEME 7
ACTIVATE WINDOW elimina_g NOSHOW
?? Chr(7)
CLEAR TYPEHEAD
@ 3,3 GET nOpcion PICTURE "@*HN" + IIF(nObjeto = 1,
"?\|<Si;|<No", "\<Si;|?!\<No") SIZE 1,10,12 DEFAULT 1 ;
VALID CONFIRMA()
@ 1,0 SAY PadC(AllTrim(cTexto), 38)
ACTIVATE WINDOW elimina_g
READ CYCLE MODAL DEACTIVATE .t OBJECT nObjeto
RELEASE WINDOW elimina_g
POP KEY
RETURN ISI
FUNCTION Confirma
ISI = (nOpcion = 1)
CLEAR READ
RETURN

FUNCTION Inform
PARAMETER cTex
PRIVATE IBell
IBell = (SET("BELL") == "ON")
SET BELL ON
?? Chr(7)
If !Empty(cTex)
WAIT WINDOW NOWAIT cTex
EndIf
If IBell
SET BELL OFF
EndIf
RETURN

* -----
* EsClave : valida que las claves capturadas existan en el catalogo
* -----
FUNCTION EsClave
PARAMETER cTipo, cClave
RETURN EMPTY(cClave) OR SEEK(cTipo + cClave,1)

FUNCTION AltoRes
PRIVATE i
i = 1
For i = 1 to ALEN(a_Resp, 1)
If EMPTY(a_Resp[i])
EXIT
EndIf
Next
i = i - 1
RETURN i

FUNCTION AltoRela
PRIVATE i
i = 1
For i = 1 to ALEN(a_Resp, 1)
If EMPTY(a_Resp[i, 1]) AND EMPTY(a_Resp[i, 2])
EXIT
EndIf
Next
i = i - 1

```

```

RETURN i
FUNCTION AltoMemo
PRIVATE cTex, nAlto
cTex = AllTrim(Premisa)
Do While Left(cTex, 2) = Chr(13) + Chr(10)
cTex = AllTrim(SubStr(cTex, 3))
EndDo
cTex = AllTrim(cTex)
Do While Right(cTex, 2) = Chr(13) + Chr(10)
cTex = AllTrim(Left(cTex, Len(cTex) - 2))
EndDo
STORE MEMBLINES(cTex) TO nAlto
RETURN nAlto

```

```

FUNCTION AltoReac
PRIVATE nMemo, nResp
nMemo = AltoMemo()
nResp = AltoRes()
RETURN nMemo + nResp

```

```

FUNCTION TrimPremisa
PRIVATE cTemp, i
cTemp = ""
For i = 1 To 31
If !InList(i, 10, 13)
cTemp = cTemp + Chr(i)
EndIf
EndFor
Premisa = AllTrim(ChrTran(m.premisa, cTemp, ""))
Do While Left(m.premisa, 2) = Chr(13) + Chr(10)
Premisa = AllTrim(SubStr(m.premisa, 3))
EndDo
Premisa = AllTrim(m.premisa)
Do While Right(m.Premisa, 2) = Chr(13) + Chr(10)
Premisa = AllTrim(Left(m.Premisa, Len(m.Premisa) - 2))
EndDo

```

```
RETURN
```

```

FUNCTION NombreArchivo
PARAMETERS cFileName, nSize
PRIVATE i, cCar, cTemp
cTemp = ""
cExt = ""
If Empty(nSize)
nSize = 5
EndIf
For i = 1 To Len(cFileName)
cCar = SubStr(cFileName, i, 1)
If !IsDigit(cCar)
EXIT
EndIf
EndFor
cFileName = SUBSTR(cFileName, i)
For i = 1 To Len(cFileName)
cCar = SubStr(cFileName, i, 1)
If IsDigit(cCar) Or IsAlpha(cCar) Or cCar $ "_$"
cTemp = cTemp + cCar
EndIf
EndFor
RETURN RTrim(PadR(Upper(cTemp), nSize))

```

```

FUNCTION ReacTipo
PARAMETERS cTipo, lAbreviado
If lAbreviado
RETURN IIF(cTipo $ "123456", SubStr(REAC_ABVTIPOS,
(Val(cTipo) - 1) * 6 + 1, 6), "")
EndIf
RETURN IIF(cTipo $ "123456", RTrim(SubStr(REAC_TIPOS,
(Val(cTipo) - 1) * 17 + 1, 17)), "TODOS")

```

```

FUNCTION SelReacTipo
PARAMETERS cTipo
cTipo = IIF(Empty(cTipo) Or !(cTipo $ "123456"), "1", cTipo)
DEFINE POPUP POP_SelReac FROM 10, 10 MARGIN SCROLL
SHADOW ;
TITLE " Seleccione " COLOR SCHEME 4
DEFINE BAR 1 OF POP_SelReac PROMPT "<Dicotmicos"
DEFINE BAR 2 OF POP_SelReac PROMPT "Soluci3n <Unica"
DEFINE BAR 3 OF POP_SelReac PROMPT "<Complementaci3n"
DEFINE BAR 4 OF POP_SelReac PROMPT "Soluci3n <Multiple"
DEFINE BAR 5 OF POP_SelReac PROMPT "<Ordenamiento"
DEFINE BAR 6 OF POP_SelReac PROMPT "<Relaci3n"
DEFINE BAR 7 OF POP_SelReac PROMPT "<Todos"
ON SELECTION POPUP POP_SelReac DEACTIVATE POPUP
POP_SelReac
ACTIVATE POPUP POP_SelReac BAR (IIF(cTipo == "", 7,
Val(cTipo)))
DEACTIVATE POPUP POP_SelReac
RELEASE POPUP POP_SelReac
RETURN IIF(Between(Bar(), 1, 6), Str(Bar(), 1), IIF((Bar() == 7),
"", cTipo))

```

```

FUNCTION LockBaja
PARAMETERS cField
PRIVATE lBaja, cError
lBaja = Cierto
cError = On("Error")
ON ERROR lBaja = Falso
If RBloqueo()
BLANK
If Empty(cField)
REPLACE Tipo WITH TipoBAJA
Else
REPLACE &cField WITH TipoBAJA
EndIf
UNLOCK
Else
lBaja = Falso
EndIf
ON ERROR &cError
RETURN lBaja

```

```

FUNCTION LockAlta
PARAMETERS cKey
PRIVATE lAlta, cError
lAlta = Cierto
cError = On("Error")
ON ERROR lAlta = Falso
If RBloqueo()
If Seek( cKey )
lAlta = Falso
Else
If !Seek(TipoBAJA)
APPEND BLANK
EndIf
EndIf
If !lAlta
UNLOCK
EndIf
Else
lAlta = Falso
EndIf
ON ERROR &cError
RETURN lAlta

```

```

FUNCTION RBloqueo
PARAMETERS nSeg
If RLock()
RETURN Cierto
EndIf
CLEAR TYPEAHEAD
nSeg = Seconds() + IIF(Empty(nSeg), 60, nSeg)
Do While nSeg > Seconds()
WAIT WINDOW "(R) Intentando Acceso a la Red " +
LTrim(Str(nSeg - Seconds())) NOWAIT
If RLock()
WAIT CLEAR
RETURN Cierto
EndIf

```

```

    If InKey(.5) = 27
        RETURN !Inform("Cancelado")
    EndIf
    EndDo
    RETURN Falso
FUNCTION FBloqueo
    PARAMETERS nSeg
    If FLock()
        RETURN Cierto
    EndIf
    CLEAR TYPEAHEAD
    nSeg = Seconds() + IIF(Empty(nSeg), 60, nSeg)
    Do While nSeg > Seconds()
        WAIT WINDOW "(F) Intentando Acceso a la Red " +
LTrim(Str(nSeg - Seconds())) NOWAIT
        If FLock()
            WAIT CLEAR
            RETURN Cierto
        EndIf
        If InKey(.5) = 27
            RETURN !Inform("Cancelado")
        EndIf
    EndDo
    RETURN Falso

```

```

FUNCTION InstAyuda
    PARAMETERS cTema
    PRIVATE cAntes, nSelect
    nSelect = Select()
    If !Used("Ayuda")
        USE Ayuda In 0 ALIAS Ayuda ORDER Tema
    EndIf
    Select Ayuda
    cAntes = Ayuda.Tema
    If Type("cTema") <> "C"
        cTema = ""
    EndIf
    SEEK cTema
    If !Found()
        GOTO TOP
    EndIf
    ON KEY LABEL F1 Do Ayuda.spr With cTema
    Select (nSelect)
    RETURN cAntes

```

```

FUNCTION OffAyuda
    POP KEY ALL
    If Used("Ayuda")
        USE IN Ayuda
    EndIf
    RETURN

```

```

* _____ *
* Reactivo.prg
* Programa principal que controla la captura de los
* reactivos
* _____ *
PARAMETERS cClase, cClave
PRIVATE nSelect, IHabil, ISel, cTitulo, cAyuda, cTema
* ..... RESGUARDA
PUSH KEY CLEAR
nSelect = SELECT()
cTema = ("CLASES DE REACTIVOS")
cAyuda = InstAyuda(cTema)
* ..... INICIO
SELECT Evaluaci
SET ORDER TO clave
GOTO TOP
IHabil = .F.
If Empty(cClase)
    cClase = "1"
EndIf
If Type("cClave") = "C"
    cClave = PADR(cClave, 3, "")
    ISel = .T.

```

```

Else
    cClave = SPACE(8)
    ISel = .F.
Endif
* ..... TITULO
DO CASE
    CASE cClase = "1"
        cTitulo = "DICOTOMICOS"
    CASE cClase = "2"
        cTitulo = "SOLUCION UNICA"
    CASE cClase = "3"
        cTitulo = "COMPLEMENTACION"
    CASE cClase = "4"
        cTitulo = "SOLUCION MULTIPLE"
    CASE cClase = "5"
        cTitulo = "ORDENAMIENTO"
    CASE cClase = "6"
        cTitulo = "RELACION"
    OTHERWISE
        cTitulo = "OTROS"
ENDCASE

```

```

cTitulo = "REACTIVOS " + cTitulo
* ..... BROWSE
DEFINE WINDOW tabla FROM 0,0 TO 18,38 GROW FLOAT
CLOSE MINIMIZE ;
    NONE COLOR SCHEME 10 ZOOM SHADOW
    MOVE WINDOW tabla CENTER
    BROWSE WINDOW tabla NOMENU NOAPPEND NODELETE
NOWAIT SAVE;
    FIELDS Clave :R, Elabora :R, Escuela :R,
Materia :R, Tema :R,;
    Grado :R, Libre_1 :R, Libre_2 :R TITLE cTitulo

```

```

KEY cClase
RELEASE WINDOW tabla
SEEK cClase + cClave
* ..... MENUS
PUSH MENU _Maysmenu
DO Reactivo.mpr
* ..... CONTROL
IHabil = .T.
ACTIVATE WINDOW Reactivos
BROWSE LAST KEY cClase REST
* ..... LIMPIEZA
ON KEY
RELEASE WINDOW Reactivos
POP MENU _Maysmenu
POP KEY
= InstAyuda(cAyuda)
SELECT (nSelect)
* ON KEY
RETURN cClave

```

```

FUNCTION Elimina
    PRIVATE nRecNo, cBandera
    PUSH KEY CLEAR
    cBandera = Clase - Clave
    If SiNo(" ELIMINO EL REGISTRO ?", 2)
        If !LockBaja("Clase")
            = Inform("Registro bloqueado por otro usuario")
        EndIf
        SEEK cBandera
    EndIf
    POP KEY
    RETURN

```

```

FUNCTION OpcionR
    PARAMETER cClaseM
    If cClaseM = cClase
        RETURN cClase
    EndIf
    DEFINE WINDOW tabla FROM 0,0 TO 18,38 GROW
    FLOAT CLOSE MINIMIZE ;
        NONE COLOR SCHEME 10 ZOOM SHADOW
        MOVE WINDOW tabla CENTER
    DO CASE
        CASE cClaseM = "1"

```

```

cTitle = "REACTIVOS DICOTOMICOS"
CASE cClaseM = "2"
  cTitle = "REACTIVOS SOLUCION UNICA"
CASE cClaseM = "3"
  cTitle = "REACTIVOS"
COMPLEMENTACION
CASE cClaseM = "4"
  cTitle = "REACTIVOS SOLUCION
MULTIPLE"
CASE cClaseM = "5"
  cTitle = "REACTIVOS ORDENAMIENTO"
CASE cClaseM = "6"
  cTitle = "REACTIVOS RELACION"
ENDCASE
RELEASE WINDOW Reactivos
BROWSE WINDOW tabla NOMENU NOAPPEND
NODELETE NOWAIT SAVE;
FIELDS Clave :R, Elabora :R,
Escuela :R, Materia :R, Tema :R;
Grado :R, Libre_1 :R, Libre_2 :R
TITLE cTitle KEY cClaseM
RELEASE WINDOW tabla
RETURN cClaseM
* ----- *
* Grupos.prg
* Programa principal que controla el módulo de grupos
* ----- *
PARAMETERS cTipo, cClave
PRIVATE nSelect, lHabil, lSel, cTitle, cTema, cAyuda
* ----- RESGUARDA
PUSH KEY CLEAR
nSelect = SELECT()
cTema = ("CATALOGO DE GRUPOS")
cAyuda = InstAyuda(cTema)
* ----- INICIO
SELECT Catalogo
SET ORDER TO clave
GOTO TOP
lHabil = .F.
cTipo = UPPER(cTipo)
IF Type("cClave") = "C"
  cClave = PADR(cClave, 8, " ")
  lSel = .T.
ELSE
  cClave = SPACE(8)
  lSel = .F.
ENDIF
* ----- TITULO
DO CASE
CASE cTipo = "A"
  cTitle = "ALUMNOS"
CASE cTipo = "P"
  cTitle = "PROFESORES"
CASE cTipo = "E"
  cTitle = "ESCUELAS"
CASE cTipo = "M"
  cTitle = "MATERIAS"
CASE cTipo = "T"
  cTitle = "TEMAS"
CASE cTipo = "G"
  cTitle = "GRUPOS"
CASE cTipo = "1"
  cTitle = "CLASES LIBRES_1"
CASE cTipo = "2"
  cTitle = "CLASES LIBRES_2"
OTHERWISE
  cTitle = "OTROS"
ENDCASE
cTitle = "CATALOGO DE " + cTitle
* ----- BROWSE
DEFINE WINDOW tabla FROM 0,0 TO 18,40 GROW FLOAT
CLOSE MINIMIZE ;
NONE COLOR SCHEME 10 ZOOM SHADOW
MOVE WINDOW tabla CENTER
BROWSE WINDOW tabla NOMENU NOAPPEND NODELETE
NOWAIT SAVE;

```

```

FIELDS Clave :R, Nombre :28 :R TITLE cTitle
KEY cTipo
RELEASE WINDOW tabla
SEEK cTipo + cClave
* ----- MENUS
PUSH MENU _Msysmenu
DO Grupos.mpr
* ----- CONTROL
lHabil = .T.
ACTIVATE WINDOW Catalogo
ON KEY LABEL SpaceBar KEYBOARD "{AJH-F8}" CLEAR
BROWSE LAST KEY cTipo REST
* ----- LIMPIEZA
ON KEY
RELEASE WINDOW Catalogo
POP MENU _Msysmenu
POP KEY
= InstAyuda(cAyuda)
GO TOP
SELECT(nSelect)
* ON KEY
RETURN cClave
FUNCTION Elimina
PRIVATE nRecNo, cBandera
PUSH KEY CLEAR
cBandera = Tipo + Clave
If SiNo("ELIMINO EL REGISTRO ?", 2)
  If !LockBaja("Tipo")
    = Inform("No se pudo realizar la operación")
  Endif
  SEEK cBandera
Endif
POP KEY
RETURN
FUNCTION Selecciona
PUSH KEY CLEAR
If (lHabil And !Empty(Clave)) Or !Inform()
  cClave = Clave
  RELEASE WINDOW Catalogo
Endif
POP KEY
RETURN
* -----
* -----
* * 1997/04/08 GRUPOS.SPR 16:48:17 *
* -----
PARAMETERS lnew
#REGION 1
PUSH KEY CLEAR
#REGION 0
REGIONAL m.currarea, m.talkstat, m.compstat
IF SET("TALK") = "ON"
  SET TALK OFF
  m.talkstat = "ON"
ELSE
  m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS
m.currarea = SELECT()
IF NOT WEXIST("edita_g");
  OR UPPER(WTITLE("EDITA_G")) == "EDITA_G.PJX";
  OR UPPER(WTITLE("EDITA_G")) == "EDITA_G.SCX";
  OR UPPER(WTITLE("EDITA_G")) == "EDITA_G.MNX";
  OR UPPER(WTITLE("EDITA_G")) == "EDITA_G.PRG";
  OR UPPER(WTITLE("EDITA_G")) == "EDITA_G.FRX";
  OR UPPER(WTITLE("EDITA_G")) == "EDITA_G.QPR"
  DEFINE WINDOW edita_g;
  FROM INT((SROW()-8)/2),INT((SCOL()-54)/2);

```

```

54)/2)+53;
TO INT((SROW()-8)/2)+7,INT((SCOL()-
TITLE ('+cTitle+');
FLOAT;
NOCLOSE;
SHADOW;
NOMINIMIZE;
DOUBLE;
COLOR SCHEME 5
ENDIF
#REGION 1
PRIVATE nOpcion, i, cClave, Clave, cTema, cAyuda
cClave = ""
Clave = ""
cTema = "CATALOGO DE GRUPOS"
cAyuda = InstAyuda(cTema)
If INew
SCATTER MEMVAR BLANK
Else
SCATTER MEMVAR
ENDIF
#REGION 1
IF WVISIBLE("edita_g")
ACTIVATE WINDOW edita_g SAME
ELSE
ACTIVATE WINDOW edita_g NOSHOW
ENDIF
@ 1,3 SAY "Clave .:";
SIZE 1,8, 0
@ 1,12 GET m.Clave;
SIZE 1,8;
DEFAULT "";
PICTURE "@!";
WHEN INew;
VALID _rj100q0x()
@ 3,21 SAY "Nombre .:";
SIZE 1,8, 0
@ 3,12 GET m.Nombre;
SIZE 1,37;
DEFAULT "";
PICTURE "@!";
@ 3,3 SAY "Nombre .:";
SIZE 1,8, 0
@ 5,6 GET nOpcion;
PICTURE "@*HN \!<Guardar;!?!<Cancelar";
SIZE 1,12,15;
DEFAULT 1;
VALID _rj100qk3()
@ 0,1 TO 4,50
IF NOT WVISIBLE("edita_g")
ACTIVATE WINDOW edita_g
ENDIF
READ CYCLE;
WHEN _rj100r3h();
DEACTIVATE .t.
RELEASE WINDOW edita_g
SELECT (m.currarea)
#REGION 0
IF m.talkstat = "ON"
SET TALK ON
ENDIF
IF m.compstat = "ON"
SET COMPATIBLE ON
ENDIF
#REGION 1
POP KEY
= InstAyuda(cAyuda)
RETURN
FUNCTION _rj100q0x && m.Clave VALID
#REGION 1

```

```

cClave = ""
For i = 1 to LEN(m.Clave)
If SubStr(m.Clave,i,1) != "-"
cClave = cClave +
SubStr(m.Clave,i,1)
ENDIF
EndFor
m.Clave = PadR(cClave,8)
Show Get m.Clave
RETURN
FUNCTION _rj100qk3 && nOpcion VALID
#REGION 1
If nOpcion = 1
If INew
m.Tipo = cTipo
GO TOP
If Seek(m.Tipo + m.Clave) or
Empty(m.Clave) or Empty(m.Nombre)
WAIT "Clave repetida o
campos vacuos" WINDOW NOWAIT
_CUROBJ = ObjNum(m.Nombre)
RETURN
Else
If LockAlta(m.Tipo + m.Clave)
GATHER MEMVAR
UNLOCK
Else
= Inform("No se pudo
completar la operaci3n")
ENDIF
Else
IF RBloqueo()
GATHER MEMVAR
Else
= Inform("No se pudo
completar la operaci3n")
ENDIF
ENDIF
CLEAR READ
RETURN
FUNCTION _rj100r3h && Read Level When
*
* When Code from screen: GRUPOS
*
#REGION 1
If INew
@ 1, 22 SAY "Nuevo" COLOR ( Scheme(5, 7)
ENDIF
*
* -----
*
* 04/05/97 DICO.SPR 14:34:30 *
*
* -----
*
PARAMETERS INew
#REGION 1
#define Cierto .t.
#define Falso .f.
#define MAX 2
#define ColorHat COLOR( " " + SCHEME(5, 7) )
#define ColorGet COLOR( SCHEME(5, 2) )
#define ColorTex COLOR SCHEME 5
PUSH KEY CLEAR
#REGION 0
REGIONAL m.currarea, m.talkstat, m.compstat
IF SET("TALK") = "ON"
SET TALK OFF
m.talkstat = "ON"

```

```

ELSE
    m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

m.currarea = SELECT()

IF NOT WEXIST("capdico");
    OR UPPER(WTITLE("CAPDICO")) = "CAPDICO.PJX";
    OR UPPER(WTITLE("CAPDICO")) = "CAPDICO.SCX";
    OR UPPER(WTITLE("CAPDICO")) = "CAPDICO.MNX";
    OR UPPER(WTITLE("CAPDICO")) = "CAPDICO.PRG";
    OR UPPER(WTITLE("CAPDICO")) = "CAPDICO.FRX";
    OR UPPER(WTITLE("CAPDICO")) = "CAPDICO.QPR"
    DEFINE WINDOW capdico;
        FROM INT((SROW()-20)/2),INT((SCOL()-74)/2)
        ;
        TO INT((SROW()-20)/2)+19,INT((SCOL()-
74)/2)+73;
        TITLE " REACTIVOS DICOTOMICOS ";
        FLOAT;
        NOCLOSE;
        SHADOW;
        NOMINIMIZE;
        DOUBLE;
        COLOR SCHEME 5
    ENDIF

#REGION 1
PRIVATE nOpcion, a_Resp, nSelect, i, lLock, cAyuda, cTema
DIMENSION a_Resp[100]

KEYBOARD ""
= INKEY()
lLock = Falso
nSelect = SELECT()
cTema = "REACTIVOS DICOTOMICOS"
cAyuda = InstAyuda(cTema)

SELECT Evaluaci
SET ORDER TO Clave

If Inew
    a_Resp = Space(50)
    SCATTER MEMVAR BLANK MEMO FIELDS EXCEPT Respuestas
    Fecha_c = Date()
    Fecha_m = Date()
Else
    SCATTER MEMVAR MEMO FIELDS EXCEPT Respuestas
    = GuardaArreglo(Falso)
    For i = 1 To MAX
        a_Resp[i] = PadR(a_Resp[i], 50)
    EndFor
EndIf

#REGION 1
IF WVISIBLE("capdico")
    ACTIVATE WINDOW capdico SAME
ELSE
    ACTIVATE WINDOW capdico NOSHOW
ENDIF
@ 3,66 SAY "□";
    SIZE 1,1, 0
@ 1,44 SAY "□";
    SIZE 1,1, 0
@ 3,44 SAY "□";
    SIZE 1,1, 0
@ 4,1 TO 7,23
@ 5,3 SAY "Alta";
    SIZE 1,4, 0
@ 6,3 SAY "Actual";
    SIZE 1,6, 0
@ 0,1 TO 3,23
@ 8,1 TO 16,70
@ 6,57 SAY "Tiempo";
    SIZE 1,6, 0
@ 1,3 SAY "Clave";
    SIZE 1,5, 0

```

```

@ 6,27 SAY "Absoluto";
    SIZE 1,8, 0
@ 6,42 SAY "Relativo";
    SIZE 1,8, 0
@ 1,27 SAY "Escuela";
    SIZE 1,7, 0
@ 2,27 SAY "Materia";
    SIZE 1,7, 0
@ 3,27 SAY "Tema";
    SIZE 1,4, 0
@ 2,49 SAY "Libre 1";
    SIZE 1,7, 0
@ 3,49 SAY "Libre 2";
    SIZE 1,7, 0
@ 0,25 TO 4,70
@ 0,26 SAY "Clasificaci#n";
    SIZE 1,15, 0
@ 5,25 TO 7,70
@ 8,2 SAY "Reactivo";
    SIZE 1,10, 0
@ 1,66 SAY "□";
    SIZE 1,1, 0
@ 2,66 SAY "□";
    SIZE 1,1, 0
@ 2,21 SAY "□";
    SIZE 1,1, 0
@ 2,3 SAY "Profesor";
    SIZE 1,8, 0
@ 2,44 SAY "□";
    SIZE 1,1, 0
@ 1,49 SAY "Grupo";
    SIZE 1,5, 0
@ 1,12 GET m.Clave;
    SIZE 1,8;
    DEFAULT "";
    PICTURE "@I";
    WHEN Inew
@ 2,12 GET nElabora;
    PICTURE "@IVN";
    SIZE 1,10,1;
    DEFAULT 0;
    WHEN MDown();
    VALID ValidBoton("P")
@ 2,12 GET m.elabora;
    SIZE 1,8;
    DEFAULT "";
    PICTURE "@KI";
    VALID ValidCampo("P")
@ 1,35 GET nEscuela;
    PICTURE "@IVN";
    SIZE 1,10,1;
    DEFAULT 0;
    WHEN MDown();
    VALID ValidBoton("E")
@ 1,35 GET m.escuela;
    SIZE 1,8;
    DEFAULT "";
    PICTURE "@KI";
    VALID ValidCampo("E")
@ 2,35 GET nMateria;
    PICTURE "@IVN";
    SIZE 1,10,1;
    DEFAULT 0;
    WHEN MDown();
    VALID ValidBoton("M")
@ 2,35 GET m.materia;
    SIZE 1,8;
    DEFAULT "";
    PICTURE "@KI";
    VALID ValidCampo("M")
@ 3,35 GET nTema;
    PICTURE "@IVN";
    SIZE 1,10,1;
    DEFAULT 0;
    WHEN MDown();
    VALID ValidBoton("T")
@ 3,35 GET m.tema;
    SIZE 1,8;
    DEFAULT "";

```

```

    PICTURE "@K";
    VALID ValidCampo("T")
@ 1,57 GET nGrado;
    PICTURE "@*IVN";
    SIZE 1,10,1;
    DEFAULT 0;
    WHEN MDown();
    VALID ValidBoton("G")
@ 1,57 GET m.grado;
    SIZE 1,8;
    DEFAULT "";
    PICTURE "@K";
    VALID ValidCampo("G")
@ 2,57 GET nLibre_1;
    PICTURE "@*IVN";
    SIZE 1,10,1;
    DEFAULT 0;
    WHEN MDown();
    VALID ValidBoton("1")
@ 2,57 GET m.libre_1;
    SIZE 1,8;
    DEFAULT "";
    PICTURE "@K";
    VALID ValidCampo("1")
@ 3,57 GET nLibre_2;
    PICTURE "@*IVN";
    SIZE 1,10,1;
    DEFAULT 0;
    WHEN MDown();
    VALID ValidBoton("2")
@ 3,57 GET m.libre_2;
    SIZE 1,8;
    DEFAULT "";
    PICTURE "@K";
    VALID ValidCampo("2")
@ 5,12 SAY m.fecha_c;
    SIZE 1,10;
    PICTURE "@E"
@ 6,12 SAY m.fecha_m;
    SIZE 1,10;
    PICTURE "@E"
@ 6,36 GET m.va;
    SIZE 1,5;
    DEFAULT 0;
    PICTURE "@K"
@ 6,51 GET m.wr;
    SIZE 1,5;
    DEFAULT 0;
    PICTURE "@K"
@ 6,64 GET m.minutos;
    SIZE 1,5;
    DEFAULT 0;
    PICTURE "@K"
@ 9,3 EDIT m.premisa;
    SIZE 3,66,500;
    PICTURE "@K";
    DEFAULT "";
    SCROLL;
    WHEN Cierto ColorTex;
    VALID _rji0v8qfq()
@ 13,6 GET a_Resp[1];
    SIZE 1,50;
    DEFAULT "";
    PICTURE "@K";
    WHEN Cierto ColorHot
@ 14,6 GET a_Resp[2];
    SIZE 1,50;
    DEFAULT "";
    PICTURE "@K"
@ 17,14 GET nOpcion;
    PICTURE "@*HN \<Guardar;?<Cancelar";
    SIZE 1,18,10;
    DEFAULT 1;
    VALID _rji0v8qxw()
@ 13,3 SAY "A";
    SIZE 1,2, 0
@ 14,3 SAY "B";
    SIZE 1,2, 0
@ 5,26 SAY "Valores";

```

```

    SIZE 1,9, 0
RETURN
IF NOT WVISIBLE("capdico")
    ACTIVATE WINDOW capdico
ENDIF
READ CYCLE;
    WHEN _rji0v8s6m();
    DEACTIVATE Cierta
RELEASE WINDOW capdico
SELECT (m.currarea)

#REGION 0
IF m.talkstat = "ON"
    SET TALK ON
ENDIF
IF m.compstat = "ON"
    SET COMPATIBLE ON
ENDIF

#REGION 1
POP KEY
= InstAyuda(cAyuda)
SELECT ( nSelect )
RETURN

#REGION 1
FUNCTION ValidCampo
PARAMETER cClase
PRIVATE cCampo

cCampo = "m." + ClaseTipo(cClase)
If LastKey() = 13 and !MDown()
    &cCampo = Grupos(cClase, &cCampo)
SHOW GET &cCampo
EndIf
RETURN

FUNCTION ValidBoton
PARAMETER cClase
PRIVATE cCampo

cCampo = "m." + ClaseTipo(cClase)
&cCampo = Grupos(cClase, &cCampo)
SHOW GET &cCampo
RETURN

FUNCTION _rji0v8qfq && m.premisa VALID
#REGION 1
Do TrimPremisa
SHOW GET m.Premisa

FUNCTION _rji0v8qxw && nOpcion VALID
#REGION 1
SELECT Evaluaci
SET ORDER TO Clave
If nOpcion = 1
    Fecha_m = DATE()
    Clase = "1"
*
DO CASE
CASE !EsClave("P", m.Elabora)
    WAIT "Clave Profesor no valida"
WINDOW NOWAIT
    _CUROBJ = ObjNum(m.Elabora)
RETURN
CASE !EsClave("E", m.Escuela)
    WAIT "Clave Escuela no valida"
WINDOW NOWAIT
    _CUROBJ = ObjNum(m.Escuela)
RETURN
CASE !EsClave("M", m.Materia)

```

```

SET MEMOWIDTH TO 80
WAIT "Clave Materia no vlida"
WINDOW NOWAIT
ObjNum(m.Materia)
_CUROBJ =
RETURN
CASE !EsClave("T", m.Tema)
WAIT "Clave Tema no vlida"
WINDOW NOWAIT
ObjNum(m.Tema)
_CUROBJ =
RETURN
CASE !EsClave("G", m.Grado)
WAIT "Clave Grupo no vlida"
WINDOW NOWAIT
ObjNum(m.Grado)
_CUROBJ =
RETURN
CASE !EsClave("1", m.Libre_1)
WAIT "Clave Libre 1 no vlida"
WINDOW NOWAIT
ObjNum(m.Libre_1)
_CUROBJ =
RETURN
CASE !EsClave("2", m.Libre_2)
WAIT "Clave Libre 2 no vlida"
WINDOW NOWAIT
ObjNum(m.Libre_2)
_CUROBJ =
RETURN
CASE EMPTY(m.VA) or EMPTY(m.VR) or
EMPTY(m.Minutos)
WAIT "Valor vacio" WINDOW
NOWAIT
_CUROBJ = ObjNum(m.VA)
RETURN
CASE EMPTY(m.premisa)
WAIT "Premisa vacia" WINDOW
NOWAIT
_CUROBJ = ObjNum(m.premisa)
RETURN
CASE EMPTY(a_Resp[1]) OR EMPTY(a_Resp[2])
WAIT "Respuesta vacia" WINDOW
NOWAIT
_CUROBJ = ObjNum(a_Resp[1])
RETURN
ENDCASE
nuevo
If !New
If Seek(m.Clase + m.Clave) or Empty(m.Clave)
WAIT "Clave no vlida" WINDOW NOWAIT
_CUROBJ = ObjNum(m.Clave)
RETURN Cierto
Endif
If !LockAlta(m.Clase+m.Clave)
= Inform("No se pudo completar la operaci#n")
_CUROBJ = ObjNum(Clave)
RETURN
Else
!Lock = Cierto
Endif
Else
If !RBloqueo()
= Inform("No se pudo completar la
operaci#n")
_CUROBJ = ObjNum(Clave)
RETURN
Else
!Lock = Cierto
Endif
Endif
a_Resp[1] = ALLTRIM( a_Resp[1] )
a_Resp[2] = ALLTRIM( a_Resp[2] )
Do TrimPremisa
GATHER MEMVAR MEMO FIELDS EXCEPT Respuestas
DO GuardaArreglo with Cierto
Endif
If !Lock
UNLOCK
EXIT
Endif

```

```

CLEAR READ
RETURN
FUNCTION _rji0v8s6m && Read Level When
*
* When Code from screen: DICO
*
#REGION 1
If !New
@ 1, 21 SAY "N" COLOR ( Scheme(5, 7) )
Endif
*
* PrintExa.prg
* Llamado por: RepoExa.frx
* Funci#n: Controlar la impresi#n seg#n la clase
* de reactivo
*
#define CRLF Chr(13) + Chr(10)
#define ANCHO 60
#define LINEA
PARAMETERS nInC
PRIVATE i
DO CASE
CASE Evaluaci.Clase = "1" OR Evaluaci.Clase = "2"
nReactivo = nReactivo + nInC
RETURN HazUnica()
CASE Evaluaci.Clase = "3"
nReactivo = nReactivo + nInC
RETURN HazComple()
CASE Evaluaci.Clase = "4"
nReactivo = nReactivo + nInC
RETURN HazMultiple()
CASE Evaluaci.Clase = "5"
nReactivo = nReactivo + nInC
RETURN HazOrdena()
CASE Evaluaci.Clase = "6"
nReactivo = nReactivo + nInC
RETURN HazRelacion()
ENDCASE
RETURN ""
FUNCTION HazUnica
PRIVATE cTex, i, cAntes, cTemp, cOrden, nResp, nCar
SET MEMOWIDTH TO ANCHO
RESTORE FROM MEMO Evaluaci.Respuestas ADDITIVE
nResp = AltoRes()
If Empty(cOrden)
cOrden = Ordena(nResp)
REPLACE Orden WITH cOrden
Else
cOrden = Orden
Endif
If nInC == 1
a_Correcto[nReactivo] = Guia(Evaluaci.Clase,
cOrden)
Endif
cAntes = Transform(nReactivo, "9999999") + ". "
cTex = ""
cTemp = QuitaCR(Evaluaci.Premisa)
For i = 1 To Max(1, MemLines(cTemp))
cTex = cTex + cAntes + PadR(MLine(cTemp, i), ANCHO) + ;
IF(i = MemLines(cTemp), " " + ;
IF(nInC > 0, a_Correcto[nReactivo], " ") + " ", "") + CRLF
cAntes = Space(10)
EndFor
For i = 1 To nResp
nCar = INT(VAL(SUBSTR(cOrden, i, 1)))
If Empty(a_Resp[nCar])
EXIT
Endif
cTex = cTex + cAntes + Chr(i + 64) + " " + a_Resp[nCar] + CRLF
EndFor

```

```
SET MEMOWIDTH TO 80
RETURN IIF(Right(cTex, 2) = CRLF, Left(cTex, Len(cTex) - 2), cTex)
```

```
FUNCTION HazComple
PRIVATE cTex, i, cAntes, cTemp, nResp, cCar, cOrden
```

```
SET MEMOWIDTH TO ANCHO
cAntes = Transform(nReactivo, "9999999") + ".-."
cTex = ""
cTemp = QuitaCR(Evaluaci.Premisa)
RESTORE FROM MEMO Evaluaci.Respuestas ADDITIVE
For i = 1 To Max(1, MemLines(cTemp))
  cTex = cTex + cAntes + PadR(MLine(cTemp, i), ANCHO) + ;
  IIF(i = MemLines(cTemp), CRLF + Space(10) + a_Resp[1] +
  CRLF + LINEA, "") + CRLF
  cAntes = Space(10)
EndFor
```

```
  nResp = AltoRes()
  If Empty(Orden)
    cOrden = Ordena(nResp)
    REPLACE Orden WITH cOrden
  Else
    cOrden = Orden
  Endif
```

```
For i = 1 To nResp
  nCar = INT(VAL(SUBSTR(cOrden, i, 1)))
  If Empty(a_Resp[nCar])
    EXIT
  Endif
  cTex = cTex + cAntes + Chr(i + 64) + " " + a_Resp[nCar] + CRLF
EndFor
  If ninc == 1
    a_Correcto[nReactivo] = Guia(Evaluaci.Clase,
  cOrden)
```

```
  Endif
SET MEMOWIDTH TO 80
RETURN IIF(Right(cTex, 2) = CRLF, Left(cTex, Len(cTex) - 2), cTex)
```

```
FUNCTION HazMultiple
PRIVATE cTex, i, cAntes, cTemp, nResp, nCar, cOrden
```

```
SET MEMOWIDTH TO ANCHO
RESTORE FROM MEMO Evaluaci.Respuestas ADDITIVE
  nResp = AltoRes()
  If Empty(Orden)
    cOrden = Ordena(nResp)
    REPLACE Orden WITH cOrden
  Else
    cOrden = Orden
  Endif
  If ninc == 1
    a_Correcto[nReactivo] = Guia(Evaluaci.Clase,
  cOrden)
  Endif
```

```
cAntes = Transform(nReactivo, "9999999") + ".-."
cTex = ""
cTemp = QuitaCR(Evaluaci.Premisa)
For i = 1 To Max(1, MemLines(cTemp))
  If ninc == 1
    cTex = cTex + cAntes + PadR(MLine(cTemp, i),
  ANCHO) + ;
    IIF(i = MemLines(cTemp), " (" +
  SUBSTR(a_Correcto[nReactivo], 1, 1) + ")" (" ;
    + SUBSTR(a_Correcto[nReactivo], 2, 1) + ")" + "" + CRLF
  Else
    cTex = cTex + cAntes + PadR(MLine(cTemp, i), ANCHO) + ;
    IIF(i = MemLines(cTemp), " ( ) ( )", "") +
  CRLF
  Endif
  cAntes = Space(10)
EndFor
```

```
For i = 1 To nResp
  nCar = INT(VAL(SUBSTR(cOrden, i, 1)))
  If Empty(a_Resp[nCar])
    EXIT
```

```
  Endif
  cTex = cTex + cAntes + Chr(i + 64) + " " + a_Resp[nCar] + CRLF
EndFor
SET MEMOWIDTH TO 80
RETURN IIF(Right(cTex, 2) = CRLF, Left(cTex, Len(cTex) - 2), cTex)
```

```
FUNCTION HazOrdena
PRIVATE cTex, i, cAntes, cTemp, cOrdena, nResp, nCar
```

```
SET MEMOWIDTH TO ANCHO
cAntes = Transform(nReactivo, "9999999") + ".-."
cTex = ""
cTemp = QuitaCR(Evaluaci.Premisa)
For i = 1 To Max(1, MemLines(cTemp))
  cTex = cTex + cAntes + PadR(MLine(cTemp, i), ANCHO) + CRLF
  cAntes = Space(10)
EndFor
```

```
RESTORE FROM MEMO Evaluaci.Respuestas ADDITIVE
  nResp = AltoRes()
  If Empty(Orden)
    cOrden = Ordena(nResp)
    REPLACE Orden WITH cOrden
  Else
    cOrden = Orden
  Endif
```

```
For i = 1 To nResp
  nCar = INT(VAL(SUBSTR(cOrden, i, 1)))
  If Empty(a_Resp[nCar])
    EXIT
  Endif
  cTex = cTex + cAntes + "(+ ALLTRIM(STR(nCar)) + )" +
  a_Resp[nCar] + CRLF
EndFor
  If ninc == 1
    a_Correcto[nReactivo] = Guia(Evaluaci.Clase,
  cOrden)
```

```
  Endif
SET MEMOWIDTH TO 80
RETURN IIF(Right(cTex, 2) = CRLF, Left(cTex, Len(cTex) - 2), cTex)
```

```
FUNCTION HazRelacion
PRIVATE cTex, i, cAntes, cTemp, cOrden1, cOrden2, nCarE, nCarR,
  nSize
```

```
SET MEMOWIDTH TO ANCHO
cAntes = Transform(nReactivo, "9999999") + ".-."
cTex = ""
cTemp = QuitaCR(Evaluaci.Premisa)
For i = 1 To Max(1, MemLines(cTemp))
  cTex = cTex + cAntes + PadR(MLine(cTemp, i), ANCHO) + CRLF
  cAntes = Space(10)
EndFor
  cOrden1 = ""
  cOrden2 = ""
  nCarE = 0
  nCarR = 0
```

```
RESTORE FROM MEMO Evaluaci.Respuestas ADDITIVE
  If Empty(Orden) OR Empty(Orden2)
    cOrden1 = Ordena( ALen(a_Resp, 1) )
    cOrden2 = Ordena( ALen(a_Resp, 1) )
    REPLACE Orden WITH cOrden1
    REPLACE Orden2 WITH cOrden2
  Else
    cOrden1 = Orden
    cOrden2 = Orden2
  Endif
  For i = 1 To ALen(a_Resp, 1)
    nCarE = INT(VAL(SUBSTR(cOrden1, i, 1)))
    nCarR = INT(VAL(SUBSTR(cOrden2, i, 1)))
    If nCarE = 0 AND i <= LEN(cOrden1)
      nCarE = 10
    Endif
    If nCarR = 0 AND i <= LEN(cOrden2)
      nCarR = 10
    Endif
```

```
  If EMPTY( a_Resp[nCarE, 1] )
    cOrden1 = STUFF(cOrden1, i, 1, ' ')
  Endif
  If EMPTY( a_Resp[nCarR, 2] )
```

```

cOrden2 = STUFF(cOrden2, i, 1, '')
EndIf
EndFor
cOrden1 = CHRTRAN(cOrden1, ',', '')
cOrden2 = CHRTRAN(cOrden2, ',', '')
If ninc == 1
    a_Correcto[nReactivo] = Guia(Evaluaci.Clase,
cOrden1, cOrden2)
EndIf
nSize = MAX(LEN(cOrden1), LEN(cOrden2))
For i = 1 to nSize
    nCarE = INT(VAL(SUBSTR(cOrden1, i, 1)))
    nCarR = INT(VAL(SUBSTR(cOrden2, i, 1)))
    If nCarE = 0
        nCarE = 10
    EndIf
    If nCarR = 0
        nCarR = 10
    EndIf
    If !EMPTY( a_Resp[nCarE, 1])
        If ninc == 1
            cTex = cTex + cAntes + "(" +
IIF(EMPTY(SUBSTR(a_Correcto[nReactivo], i, 1)), "", ";
SUBSTR(a_Correcto[nReactivo], i, 1)) + ") " + ;
PADR(a_Resp[nCarE, 1], 30)
Else
            cTex = cTex + cAntes + "( ) " +
PADR(a_Resp[nCarE, 1], 30)
EndIf
Else
            cTex = cTex + SPACE(35)
EndIf
    If !EMPTY( a_Resp[nCarR, 2])
        cTex = cTex + CHR(64 + i) + " " +
a_Resp[nCarR, 2] + CRLF
Else
            cTex = cTex + CRLF
EndIf
EndFor
SET MEMOWIDTH TO 80
RETURN IIF(Right(cTex, 2) = CRLF, Left(cTex, Len(cTex) - 2), cTex)

FUNCTION QuitaCR
PARAMETERS cTex
cTex = AllTrim(cTex)
Do While Left(cTex, 2) = Chr(13) + Chr(10)
    cTex = AllTrim(SubStr(cTex, 3))
EndDo
cTex = AllTrim(cTex)
Do While Right(cTex, 2) = Chr(13) + Chr(10)
    cTex = AllTrim(Left(cTex, Len(cTex) - 2))
EndDo
RETURN cTex

* ----- *
* RepoCali.prg
* Programa que controla el reporte de las calificaciones
* ----- *
#define Cierto .t.
#define Falso .f.
PARAMETER IGuia
PRIVATE IRepError, ILimpia, cExamen, IEstado, IContinua

STORE Falso TO ILimpia, IRepError
cExamen = ""
STORE Cierto TO IEstado, IContinua

Do While !IRepError and IContinua
    ILimpia = Falso
    Do SelfFile.spr with cExamen, ".DBC",
"Seleccione el examen "
    If !Empty(cExamen) and IContinua
        WAIT WINDOW "Imprimiendo.."
NOWAIT
        If !Used(cExamen)
            USE (cExamen +
.DBC") IN 0 ALIAS Examen

```

```

EndIf
Select Examen
Do while SYS(13) = 'OFFLINE'
    Do Intental.spr
    If IEstado = .F.
        IRepError =
Exit
EndIf
EndDo
If !IRepError
    REPORT FORM
Else
    WAIT WINDOW
EndIf
WAIT CLEAR
If Used ("Examen")
    USE IN Examen
EndIf
EndDo

RETURN

If ILimpia
    _PLINENO = 0
    ILimpia = .F.
EndIf

If _PLINENO + MemLines(PrintExa(0)) >= _PLENGTH - 6
    ILimpia = .T.
    RETURN .T.
EndIf
RETURN .F.

If ILimpia
    _PLINENO = 0
    ILimpia = .F.
EndIf

If _PLINENO + MemLines(PrintRes(0)) >= _PLENGTH - 6
    ILimpia = .T.
    RETURN .T.
EndIf
RETURN .F.

* ----- *
* 1997/04/05 15:30:31 *
* ----- *

PARAMETERS cguion

#REGION 1
#define Cierto .t.
#define Falso .f.
#define VAR_REAC ReacTipo(Reactivo.Clase, Cierto) +
"+Reactivo.Clave
#define VAR_GUION ReacTipo(CurGuion.Clase, Cierto) +
"+CurGuion.Clave
PUSH KEY CLEAR

#REGION 0
REGIONAL m.currea, m.talkstat, m.compstat

IF SET("TALK") = "ON"
    SET TALK OFF
    m.talkstat = "ON"
ELSE
    m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

```

```

m.currarea = SELECT()

#REGION 1
PRIVATE nOpcion, nSelect, cSelProf, cSelEscu, cSelMate;
cSelTema, cSelGrad, cSelL_1,
cSelL_2, cSelClase;
cReactivo, cClase, cClave, cCurGui,
IContinua, ;
IVaciosP, IVaciosE, IVaciosM, IVaciosT,
IVaciosG,;
IVacios1, IVacios2, aFiltros, nFiltros, INuevo, cTema, cAyuda

DIMENSION aFiltros[8]

aFiltros[1] = "Profesor"
aFiltros[2] = "Escuela"
aFiltros[3] = "Materia"
aFiltros[4] = "Tema"
aFiltros[5] = "Grado"
aFiltros[6] = "Libre 1"
aFiltros[7] = "Libre 2"
aFiltros[8] = "Clase"
nFiltros = 1

IContinua = Cierta
nSelect = SELECT()
cTema = "FORMACION DE GUIONES"
cAyuda = InstAyuda(cTema)
cSelProf = ""
cSelEscu = ""
cSelMate = ""
cSelTema = ""
cSelGrad = ""
cSelL_1 = ""
cSelL_2 = ""
cSelClase = ""

STORE Cierta TO IVaciosP, IVaciosE, IVaciosM, IVaciosT,
IVaciosG,;
IVacios1, IVacios2

If Empty(cGuión)
  Do Filtra WITH
  INuevo = Cierta
Else
  If File(cGuión) And UPPER(RIGHT(cGuión, 4))
  == "DBG"
  If !Used( LEFT(cGuión,
LEN(cGuión) - 4) )
  USE (cGuión) IN 0
ORDER 0
  Endif
  CREATE CURSOR Reactivo (Clase
C(1), Clave C(8))
  INDEX ON Clase + Clave TAG
  Clave ADDITIVE
  Select Reactivo
  APPEND FROM (cGuión) FIELDS
  Clase, Clave FOR !Empty(Clase)
  Else
  = Inform("El guiñ " + cGuión + " no
existe")
  Select(nSelect)
  POP KEY
  RETURN
  Endif
  Endif

CREATE CURSOR CurGuión (Clase C(1), Clave C(8))
Select CurGuión
INDEX ON Clase + Clave TAG Clave ADDITIVE
Select Reactivo

cReactivo = VAR_REAC
cCurGui = VAR_GUIÓN

```

```
#REGION 1
```

```

DEFINE POPUP _rji0x8ugg ;
PROMPT FIELD VAR_REAC ;
SCROLL ;
MARGIN ;
MARK "□"

DEFINE POPUP _rji0x8u7 ;
PROMPT FIELD VAR_GUIÓN ;
SCROLL ;
MARGIN ;
MARK "□"

#REGION 1
IF WVISIBLE("guión")
  ACTIVATE WINDOW guión SAME
ELSE
  ACTIVATE WINDOW guión NOSHOW
ENDIF
@ 10,16 TO 19,71
@ 0,0 SAY " Reactivos " ;
SIZE 1,11, 0
@ 10,60 SAY " Criterios " ;
SIZE 1,11, 0
@ 0,43 SAY "Seleccionados para el Guiñ" ;
SIZE 1,27, 0
@ 1,1 GET cReactivo ;
PICTURE "@&N" ;
POPUP _rji0x8ugg ;
SIZE 9,29 ;
DEFAULT "" ;
VALID _rji0x8vst() ;
COLOR SCHEME 6
@ 3,31 GET nGuión ;
PICTURE "@VN \<Todos;\<Ninguno" ;
SIZE 1,11,3 ;
DEFAULT 1 ;
VALID _rji0x8xg7()
@ 1,43 GET cCurGui ;
PICTURE "@&N" ;
POPUP _rji0x8u7 ;
SIZE 9,29 ;
DEFAULT "" ;
WHEN _rji0x8zi3() ;
VALID _rji0x902w() ;
COLOR SCHEME 6
@ 10,1 GET nFiltros ;
PICTURE "@&N" ;
FROM aFiltros ;
SIZE 10,14 ;
DEFAULT 1 ;
VALID Filtros() ;
COLOR SCHEME 6
@ 11,18 SAY IIF(cSelProf == "", "TODOS", ChrTran(cSelProf, "~ ",
".")) ;
SIZE 1,52
@ 12,18 SAY IIF(cSelEscu == "", "TODOS", ChrTran(cSelEscu, "~ ",
".")) ;
SIZE 1,52
@ 13,18 SAY IIF(cSelMate == "", "TODOS", ChrTran(cSelMate, "~ ",
".")) ;
SIZE 1,52
@ 14,18 SAY IIF(cSelTema == "", "TODOS", ChrTran(cSelTema, "~ ",
".")) ;
SIZE 1,52
@ 15,18 SAY IIF(cSelGrad == "", "TODOS", ChrTran(cSelGrad, "~ ",
".")) ;
SIZE 1,52
@ 16,18 SAY IIF(cSelL_1 == "", "TODOS", ChrTran(cSelL_1, "~ ",
".")) ;
SIZE 1,52
@ 17,18 SAY IIF(cSelL_2 == "", "TODOS", ChrTran(cSelL_2, "~ ",
".")) ;
SIZE 1,52
@ 18,18 SAY Reactipo(cSelClase) ;
SIZE 1,52
@ 20,13 GET nOpcion ;
PICTURE "@HN \!<Guardar;\!<Cancelar" ;
SIZE 1,17,13 ;
DEFAULT 1 ;

```

```

VALID _rji0x92wkd)
RETURN
IF NOT WVISIBLE("guion")
ACTIVATE WINDOW guion
ENDIF
READ CYCLE ;
SHOW _rji0x93yo()
RELEASE WINDOW guion
SELECT (m.currarea)
RELEASE POPUPS _rji0x8ugg, _rji0x8ul7
#REGION 0
IF m.talkstat = "ON"
SET TALK ON
ENDIF
IF m.compstat = "ON"
SET COMPATIBLE ON
ENDIF
#REGION 1
POP KEY
= InstAyuda(cAyuda)
SELECT (nSelect)
RETURN
#REGION 1
FUNCTION Filtras
GOTO TOP IN Reactivo
cReactivo = VAR_REAC
Do Case
Case nFiltros = 1
Do Criterios.spr WITH "P", cSelProf, IVaciosP
Case nFiltros = 2
Do Criterios.spr WITH "E", cSelEscu, IVaciosE
Case nFiltros = 3
Do Criterios.spr WITH "M", cSelMate, IVaciosM
Case nFiltros = 4
Do Criterios.spr WITH "T", cSelTema, IVaciosT
Case nFiltros = 5
Do Criterios.spr WITH "G", cSelGrad, IVaciosG
Case nFiltros = 6
Do Criterios.spr WITH "1", cSelL_1, IVacios1
Case nFiltros = 7
Do Criterios.spr WITH "2", cSelL_2,
IVacios2
Case nFiltros = 8
cSelClase = SelReactipo(cSelClase)
EndCase
Do Filtra WITH cSelProf, cSelEscu, cSelMate, cSelTema, cSelGrad,;
cSelL_1, cSelL_2, cSelClase
GOTO TOP IN Reactivo
cReactivo = VAR_REAC
_CUROBJ = ObjNum(cReactivo)
SHOW GETS
RETURN
FUNCTION _rji0x8vst && cReactivo VALID
#REGION 1
PRIVATE nRecNo, nRecActual
If !Empty(cReactivo) And !Eof('Reactivo')
SELECT Reactivo
cClase = Clase
cClave = Clave
nRecNo = RecNo()
SKIP
If Eof()
GOTO nRecNo
SKIP -1
ENDIF
nRecActual = IIF(Eof() Or Bof(), 0, RecNo())
GOTO nRecNo
DELETE
If nRecActual == 0
GOTO TOP
Else
GOTO nRecActual

```

```

ENDIF
cReactivo = VAR_REAC
SHOW GET cReactivo
If !Empty(cReactivo)
_CUROBJ = ObjNum(cReactivo)
ENDIF
SELECT CurGuion
SEEK cClase + cClave
If !Found()
INSERT INTO CurGuion VALUES (cClase, cClave)
ENDIF
cCurGui = VAR_GUION
SHOW GET cCurGui
If Empty(cReactivo) And !Empty(cCurGui)
_CUROBJ = ObjNum(cCurGui)
ENDIF
ENDIF
RETURN
FUNCTION _rji0x8xg7 && nGuion VALID
#REGION 1
Do Case
Case nGuion = 1
SELECT Reactivo
GOTO TOP
Do While !EOF()
cClase = Clase
cClave = Clave
DELETE
SKIP
cReactivo = VAR_REAC
If !Empty(cReactivo)
_CUROBJ = ObjNum(cReactivo)
ENDIF
SELECT CurGuion
SET ORDER TO TAG Clave
Seek cClase + cClave
If !Found()
INSERT INTO CurGuion (Clase, Clave);
VALUES (cClase, cClave)
ENDIF
cCurGui = VAR_GUION
If Empty(cReactivo) And !Empty(cCurGui)
_CUROBJ = ObjNum(cCurGui)
ENDIF
SELECT Reactivo
EndDo
CREATE CURSOR Reactivo (Clase C(1), Clave C(8))
SELECT Reactivo
INDEX ON Clase + Clave TAG Clave ADDITIVE
Case nGuion = 2
SELECT CurGuion
GOTO TOP
Do While !EOF()
cClase = Clase
cClave = Clave
DELETE
SKIP
cCurGui = VAR_GUION
If !Empty(cCurGui)
_CUROBJ = ObjNum(cCurGui)
ENDIF
SELECT Reactivo
SET ORDER TO TAG Clave
Seek cClase + cClave
If !Found()
INSERT INTO Reactivo (Clase, Clave);
VALUES (cClase, cClave)
ENDIF
cReactivo = VAR_REAC
If Empty(cCurGui) And !Empty(cReactivo)
_CUROBJ = ObjNum(cReactivo)
ENDIF
SELECT CurGuion
EndDo
CREATE CURSOR CurGuion (Clase C(1), Clave C(8))
Select CurGuion
INDEX ON Clase + Clave TAG Clave ADDITIVE
EndCase

```

```
SHOW GETS
RETURN
```

```
FUNCTION _rj0x&zi3 && cCurGui WHEN
#REGION 1
If Empty(cCurGui)
GOTO TOP IN CurGuiOn
cCurGui = VAR_GUION
EndIf
RETURN !Empty(cCurGui)
```

```
FUNCTION _rj0x902w && cCurGui VALID
#REGION 1
PRIVATE nRecNo, nRecActual
If !Empty(cCurGui) And !Eof('CurGuiOn')
SELECT CurGuiOn
cClase = Clase
cClave = Clave
nRecNo = RecNo()
SKIP
If Eof()
GOTO nRecNo
SKIP -1
EndIf
nRecActual = IIF(Eof() Or Bof(), 0, RecNo())
GOTO nRecNo
DELETE
If nRecActual == 0
GOTO TOP
Else
GOTO nRecActual
EndIf
cCurGui = VAR_GUION
SHOW GET cCurGui
If !Empty(cCurGui)
_CUROBJ = ObjNum(cCurGui)
EndIf
SELECT Reactivo
SET ORDER TO TAG Clave
SEEK cClase + cClave
If !Found()
INSERT INTO Reactivo VALUES (cClase, cClave)
EndIf
cReactivo = VAR_REAC
SHOW GET cReactivo
If Empty(cCurGui) And !Empty(cReactivo)
_CUROBJ = ObjNum(cReactivo)
EndIf
EndIf
RETURN
```

```
FUNCTION _rj0x92wk && nOpcion VALID
#REGION 1
Do Case
Case nOpcion = 1
If Empty(RecCount("CurGuiOn"))
= Inform("No hay
reactivos en la selecci3n")
_CUROBJ =
ObjNum(cReactivo)
RETURN
EndIf
Do DatosGui.spr With !Nuevo
If !Continua = Cierto
Select Reactivo
USE
Select CurGuiOn
USE
CLEAR READ
EndIf
Case nOpcion = 2
Select Reactivo
USE
Select CurGuiOn
USE
CLEAR READ
EndCase
RETURN
```

```
FUNCTION _rj0x93yo && Read Level Show
PRIVATE currwind
STORE WOUTPUT() TO currwind
* Show Code from screen: GUION
*
#REGION 1
IF SYS(2016) = "GUION" OR SYS(2016) = ""
ACTIVATE WINDOW guion SAME
@ 11,18 SAY IIF(cSelProf == "", "TODOS",
ChrTran(cSelProf, "~", ";"));
SIZE 1,52, 0
@ 12,18 SAY IIF(cSelEscu == "", "TODOS",
ChrTran(cSelEscu, "~", ";"));
SIZE 1,52, 0
@ 13,18 SAY IIF(cSelMate == "", "TODOS",
ChrTran(cSelMate, "~", ";"));
SIZE 1,52, 0
@ 14,18 SAY IIF(cSelTema == "", "TODOS",
ChrTran(cSelTema, "~", ";"));
SIZE 1,52, 0
@ 15,18 SAY IIF(cSelGrad == "", "TODOS",
ChrTran(cSelGrad, "~", ";"));
SIZE 1,52, 0
@ 16,18 SAY IIF(cSelL_1 == "", "TODOS",
ChrTran(cSelL_1, "~", ";"));
SIZE 1,52, 0
@ 17,18 SAY IIF(cSelL_2 == "", "TODOS",
ChrTran(cSelL_2, "~", ";"));
SIZE 1,52, 0
@ 18,18 SAY ReactTipo(cSelClase);
SIZE 1,52, 0
ENDIF
IF NOT EMPTY(currwind)
ACTIVATE WINDOW (currwind) SAME
ENDIF
```

```
-----
*
*
* 1997/04/08 CRITERIO.SPR 16:42:34 *
*
-----
```

```
PARAMETERS ctipo, ccriterio, lvacios
#REGION 1
#define MARCA "□"
#define Cierto .t
#define Falso .f
PUSH KEY CLEAR
If Empty(ctipo)
POP KEY
RETURN
EndIf
#REGION 0
REGIONAL m.currarea, m.talkstat, m.compstat
IF SET("TALK") = "ON"
SET TALK OFF
m.talkstat = "ON"
ELSE
m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS
m.currarea = SELECT()
#REGION 1
PRIVATE i, cClave, nTodos, nNinguno, nInvertir, nOpcion, cListG;
cCampo, nSelect, cTema, cAyuda
nSelect = Select()
cTema = ("CRITERIOS DE SELECCION")
cAyuda = InstAyuda(cTema)
```

```

CREATE CURSOR Grupos (Marcado C(1), Clave C(8), Nombre
C(50))
SELECT Catalogo
SCAN FOR Tipo = cTipo
INSERT INTO Grupos VALUES (" ", Catalogo.Clave,
Catalogo.Nombre)
ENDSCAN

```

```

SELECT Grupos
DO CASE
CASE Empty(cCriterio)
CASE cCriterio == ""
REPLACE ALL MARCADO WITH MARCA
OTHERWISE
REPLACE ALL MARCADO WITH MARCA FOR Clave $ cCriterio
ENDCASE
GOTO TOP
cListG = Marcado+Clave+" "+Nombre

```

```

#REGION 1
DEFINE POPUP _rj10ztdj8 ;
PROMPT FIELD Grupos.Marcado+Grupos.Clave+"
"+Grupos.Nombre ;
SCROLL ;
MARGIN ;
MARK "□"

```

```

#REGION 1
IF WVISIBLE("criterios")
ACTIVATE WINDOW criterios SAME
ELSE
ACTIVATE WINDOW criterios NOSHOW
ENDIF

```

```

@ 0,1 SAY "Marque su selecci#n:" ;
SIZE 1,20,0

```

```

@ 1,1 GET cListG ;
PICTURE "@&N" ;
POPUP _rj10ztdj8 ;
SIZE 17,42 ;
DEFAULT "" ;
VALID _rj10ztdxb() ;
COLOR SCHEME 6

```

```

@ 1,45 GET nTodos ;
PICTURE "@*HN \<Todos" ;
SIZE 1,12,1 ;
DEFAULT 1 ;
VALID _rj10ztedk()

```

```

@ 4,45 GET nNinguno ;
PICTURE "@*HN \<Ninguno" ;
SIZE 1,12,1 ;
DEFAULT 1 ;
VALID _rj10zter0()

```

```

@ 7,45 GET nInvertir ;
PICTURE "@*HN \<Invertir" ;
SIZE 1,12,1 ;
DEFAULT 1 ;
VALID _rj10ztf6u()

```

```

@ 10,45 GET nOpcion ;
PICTURE "@*VT \<Aceptar;\<Cancelar" ;
SIZE 1,12,2 ;
DEFAULT 1 ;
VALID _rj10ztf1c()

```

```

@ 16,46 GET IVacios ;
PICTURE "@*C \<Vacios" ;
SIZE 1,10 ;
DEFAULT 0

```

```

IF NOT WVISIBLE("criterios")
ACTIVATE WINDOW criterios
ENDIF

```

```

READ CYCLE

```

```

RELEASE WINDOW criterios
SELECT (m.currarea)

```

```

RELEASE POPUPS _rj10ztdj8

```

```

#REGION 0
IF m.talkstat = "ON"
SET TALK ON
ENDIF
IF m.compstat = "ON"
SET COMPATIBLE ON
ENDIF

```

```

#REGION 1
If Used("Grupos")
USE IN Grupos
EndIf
POP KEY
= InstAyuda(cAyuda)
SELECT (nSelect)
RETURN

```

```

FUNCTION _rj10ztdxb && cListG VALID
#REGION 1
If !Empty(cListG)
REPLACE Marcado WITH IIF(Empty(Marcado), MARCA, "")
SKIP
If EOF()
GOTO BOTTOM
EndIf
cListG = Marcado+Clave+" "+Nombre
SHOW GET cListG
_CUROBJ = ObjNum(cListG)
EndIf
RETURN

```

```

FUNCTION _rj10ztedk && nTodos VALID
#REGION 1
PRIVATE nRecNo
nRecNo = IIF(EOF() Or Bof(), 0, RecNo())
REPLACE ALL Marcado WITH MARCA
If nRecNo = 0
GOTO TOP
Else
GOTO nRecNo
EndIf
IVacios = Cierto
cListG = Marcado+Clave+" "+Nombre
SHOW GETS
_CUROBJ = ObjNum(cListG)

```

```

FUNCTION _rj10zter0 && nNinguno VALID
#REGION 1
PRIVATE nRecNo
nRecNo = IIF(EOF() Or Bof(), 0, RecNo())
REPLACE ALL Marcado WITH ""
If nRecNo = 0
GOTO TOP
Else
GOTO nRecNo
EndIf
IVacios = Falso
cListG = Marcado+Clave+" "+Nombre
SHOW GETS
_CUROBJ = ObjNum(cListG)

```

```

FUNCTION _rj10ztf6u && nInvertir VALID
#REGION 1
PRIVATE nRecNo
nRecNo = IIF(EOF() Or Bof(), 0, RecNo())
REPLACE ALL Marcado WITH IIF(Empty(Marcado), MARCA, "")
If nRecNo = 0
GOTO TOP
Else
GOTO nRecNo
EndIf
IVacios = IIF(IVacios,Falso,Cierto)
cListG = Marcado+Clave+" "+Nombre
SHOW GETS
_CUROBJ = ObjNum(cListG)

```

```

FUNCTION _rj10ztf1c && nOpcion VALID
#REGION 1
If nOpcion = 1

```

```

COUNT ALL FOR !Empty(Marcado) TO nOpcion
DO CASE
CASE nOpcion = RecCount() AND !Vacios = Cierta
cCriterio = ""
CASE nOpcion = 0
cCriterio = ""
OTHERWISE
cCriterio = IIF(!Vacios,
SPACE(8) + " ", " ")
cCriterio = ""
SCAN FOR !Empty(Marcado)
cCriterio = Clave + "-" + cCriterio
ENDSCAN
ENDCASE
Endif
CLEAR READ
RETURN
* Filtra.prg
* Funci3n: Realizar el filtro de la base de datos
* Evaluaci, seg3n los campos dados como par -
* metros
PARAMETERS cMae, cEsc, cMate, cTema, cGrado, cLi_1, cLi_2,
cClase
PRIVATE cMacro
If Empty(cClase)
cClase = SPACE(12)
Endif
If Empty(cMae)
cMae = SPACE(8)
Endif
If Empty(cEsc)
cEsc = SPACE(8)
Endif
If Empty(cMate)
cMate = SPACE(8)
Endif
If Empty(cTema)
cTema = SPACE(8)
Endif
If Empty(cGrado)
cGrado = SPACE(8)
Endif
If Empty(cLi_1)
cLi_1 = SPACE(8)
Endif
If Empty(cLi_2)
cLi_2 = SPACE(8)
Endif
cMacro = ""
If cMae != ""
cMacro = cMacro + 'Elabora $ cMae'
Endif
If cEsc != ""
cMacro = cMacro + IIF(Empty(cMacro), "", " And ") +
'Escuela $ cEsc'
Endif
If cMate != ""
cMacro = cMacro + IIF(Empty(cMacro), "", " And ") +
'Materia $ cMate'
Endif
If cTema != ""
cMacro = cMacro + IIF(Empty(cMacro), "", " And ") +
'Tema $ cTema'
Endif
If cGrado != ""
cMacro = cMacro + IIF(Empty(cMacro), "", " And ") +
'Grado $ cGrado'
Endif
If cLi_1 != ""
cMacro = cMacro + IIF(Empty(cMacro), "", " And ") +
'Libre_1 $ cLi_1'
Endif
If cLi_2 != ""
cMacro = cMacro + IIF(Empty(cMacro), "", " And ") +
'Libre_2 $ cLi_2'
Endif

```

```

If cClase != ""
cMacro = cMacro + IIF(Empty(cMacro), "", " And
") + 'Clase $ cClase'
Endif
CREATE CURSOR Reactivo (Clase C(1), Clave C(8))
SELECT EVALUACI
If !Empty(cMacro)
cMacro = "FOR " + cMacro
Endif
SCAN &cMacro
INSERT INTO Reactivo VALUES (Evaluaci.Clase,
Evaluaci.Clave)
ENDSCAN
SELECT Reactivo
INDEX ON Clase + Clave TAG Clave ADDITIVE
SET RELATION TO Clase + Clave INTO Evaluaci
If Empty(RecCount())
WAIT WINDOW "No hay reactivos que
cumplan los criterios" NOWAIT
Endif
RETURN
* Archi.prg
* Llamado por: NomExa.spr
* Par metro: Nombre del gui3n
* Funci3n: Crear un nombre de examen nuevo a partir
* de la raiz dada por el nombre del gui3n
* al que pertenece
#define cFile cRaiz + transform(n, '@L 999') + '.DBE'
#define Cierta .t.
#define Falso .f.
PARAMETER cRaiz
PRIVATE n, !Exito
n = 1
!Exito = Falso
cAuxFile = ""
For n = 1 To 999
If !FILE (cFile)
!Exito = Cierta
EXIT
Endif
EndFor
If !!Exito
= Inform("No fue posible crear el examen")
RETURN ""
Endif
If !File("EXA00000.DBF") OR !File("EXA00000.CDX")
= Inform("El archivo EXA00000 o su indice no
existe")
RETURN ""
Endif
If !Used("EXA00000")
Use EXA00000 IN 0
Endif
Select EXA00000
COPY STRUCTURE TO cFile WITH CDX
Select EXA00000
USE
RETURN cFile
* Depura.prg
* Par metro: Nombre de gui3n
* Funci3n: Elimina de un gui3n los registros que hayan sido elimi-
* nados en Evaluaci.DBF
#define Cierta .t.
#define Falso .f.
PARAMETER cFile
PRIVATE i, nDel, Flag
Flag = Falso
nDel = 0
If !Used( LEFT(cFile, LEN(cFile) - 4 )
Use (cFile) IN 0

```

```

Flag = Cierta
Endif
Select (cFile)
If FBloqueo()
WAIT WINDOW "Actualizando archivo " + cFile
NOWAIT
PACK
GOTO TOP
Do While !Eof()
If !SEEK(Clase + Clave, "Evaluaci")
DELETE
nDel = nDel + 1
Endif
SKIP
EndDo
PACK
WAIT CLEAR
UNLOCK
If Flag
USE
Else
nDel = -1
Endif
RETURN nDel

```

```

*
*
* 1997/04/08 SELFIE.SPR 16:09:58 *
*

```

```

PARAMETERS cfile, cext, ctitulo, lall
#REGION 1
PRIVATE cList, cPopUp, cMacro, cGuion, cFecha, cDescripcion, ;
cElabora, cEscuela, cTitulo1, cTitulo1, cTitulo2, cTitulo3, ;
cRaiz, a_Dir, nDir, nPos, i, nSelect, cTema, cAyuda
PUSH KEY CLEAR
nSelect = Select()
cTema = "SELECCION DE ARCHIVOS"
cAyuda = InstAyuda(cTema)
DIMENSION a_Dir[1]
a_Dir = Space(19)

If Empty(cExt)
cExt = ""
Else
cExt = AllTrim(Upper(cExt))
Endif
nDir = ADir(a_Dir, "" + cExt)
If nDir > 0
DIMENSION a_Dir[ALen(a_Dir)]
nPos = 1
For i = 1 To ALen(a_Dir) Step 5
a_Dir[nPos] = PadR(a_Dir[i], 19)
nPos = nPos + 1
EndFor
DIMENSION a_Dir[nDir]
= ASort(a_Dir)
Endif

cFile = Upper(cFile)
nPos = AScan(a_Dir, cFile, 1)
cList = IIF(nPos > 0, a_Dir[nPos], a_Dir[1])
Continua = t.

#REGION 0
REGIONAL m.currarea, m.talkstat, m.compstat
IF SET("TALK") = "ON"
SET TALK OFF
m.talkstat = "ON"
ELSE

```

```

m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

m.currarea = SELECT()

#REGION 1
#REGION 1
IF WVISIBLE("selguion")
ACTIVATE WINDOW selguion SAME
ELSE
ACTIVATE WINDOW selguion NOSHOW
ENDIF
@ 1,1 GET cList ;
PICTURE "@&N" ;
FROM a_Dir ;
SIZE 16,23 ;
DEFAULT 1 ;
WHEN Botones() ;
VALID _rj0yngnc() ;
COLOR SCHEME 6
@ 2,25 GET nBoton ;
PICTURE "@*VN
\|<Selecciona;|<Informaci;n;|<Nuevo;|<Edita;|?<Cancela" ;
SIZE 1,15,2 ;
DEFAULT 1 ;
VALID _rj0yngz3()
@ 0,2 SAY cTitulo ;
SIZE 1,38 ;
PICTURE "@T"

IF NOT WVISIBLE("selguion")
ACTVATE WINDOW selguion
ENDIF

READ CYCLE ;
WHEN Botones()

RELEASE WINDOW selguion
SELECT (m.currarea)

#REGION 0
IF m.talkstat = "ON"
SET TALK ON
ENDIF
IF m.compstat = "ON"
SET COMPATIBLE ON
ENDIF

#REGION 1
POP KEY
= InstAyuda(cAyuda)
Select(nSelect)

RETURN

#REGION 1
FUNCTION Botones
If InList(cExt, ".DBG", ".DBE")
SHOW GET nBoton, 3 ENABLE
Else
SHOW GET nBoton, 3 DISABLE
Endif
If nDir > 0
SHOW GET nBoton, 1 ENABLE
If cExt = ".DBG"
SHOW GET nBoton, 2 ENABLE
SHOW GET nBoton, 4 ENABLE
Else
SHOW GET nBoton, 2 DISABLE
SHOW GET nBoton, 4 DISABLE
Endif
Else
SHOW GET nBoton, 1 DISABLE
SHOW GET nBoton, 2 DISABLE

```

```

SHOW GET nBoton, 4 DISABLE
SHOW GET cList DISABLE
_CUROBJ = ObjNum(nBoton)

EndIf
RETURN LEFT(cList, LEN(cList) - 4)

FUNCTION _rj0yngnc && cList VALID
#REGION 1
If !Empty(cList)
If !All
cFile = cList
Else
cFile = SubStr(cList, 1, At(".", cList) - 1)
EndIf
CLEAR READ
EndIf
RETURN

FUNCTION _rj0yngz3 && nBoton VALID
#REGION 1
PRIVATE cTemp
Do Case
Case nBoton = 1 And !Empty(cList)
If !All
cFile = cList
Else
cFile = SubStr(cList, 1, At(".", cList) - 1)
EndIf
CLEAR READ
Case nBoton = 2
If cExt == ".DBG"
cTemp = SubStr(cList, 1, At(".", cList) - 1)
If !Used(cTemp)
Use (cTemp + cExt) IN 0
ORDER 0
EndIf
SELECT (cTemp)
SET RELATION TO Clave INTO
Bitacora
LOCATE FOR Empty(CLASE)
If !Found() Or Eof()
USE IN (cTemp)
RETURN Inform("No se
encontró la bit cora")
EndIf
cGuion =
SUBSTR(Bitacora.Datos, 1, 50)
cFecha =
SUBSTR(Bitacora.Datos, 51, 50)
cDescripcion =
SUBSTR(Bitacora.Datos, 101, 50)
cElabora =
SUBSTR(Bitacora.Datos, 151, 50)
cEscuela =
SUBSTR(Bitacora.Datos, 201, 50)
cTitulo1 =
SUBSTR(Bitacora.Datos, 251, 50)
cTitulo2 =
SUBSTR(Bitacora.Datos, 301, 50)
cTitulo3 =
SUBSTR(Bitacora.Datos, 351, 50)
DO InfoGui.spr
USE IN (cTemp)
Else
CLEAR READ
EndIf
Case nBoton = 3
If cExt == ".DBG"
Do Guion.spr
Else
If cExt = ".DBE"
cRaiz = ""
Do NomExa.spr
If !Empty(cRaiz)
cTemp =
Archi(ALLTRIM(cRaiz))
!Empty(cTemp)

```

```

cList = cTemp
EndIf
EndIf
EndIf
EndIf
SHOW GET cList
Case nBoton = 4
Do Guion.spr With AllTrim(cList)
SHOW GET cList
Otherwise
CLEAR READ
!Continua = .f.
EndCase
RETURN
* _____ *
* Programa que contiene las funciones para la aplicaci3n *
* de los reactivos *
* _____ *
PARAMETERS cGuion, cGuardaExa

#define MEMOANCHO 60
#define Cierto .t.
#define Falso .f.
#define MAR_INF 4
#define MAR_SUP 3

#define a_CLASE 1
#define a_CLAVE 2
#define a_NUMERO 3
#define a_RES 4
#define a_RESB 5
#define a_RES0 6
#define a_RES1 7
#define a_RES2 8
#define a_RES3 9
#define a_RES4 10
#define a_RES5 11
#define a_RES6 12
#define a_RES7 13
#define a_RES8 14
#define a_RES9 15
#define a_ALTO 16
#define a_PAG 17
#define a_ORDEN 18
#define a_ORDEN2 19
#define a_RELE 20
#define a_VA 21
#define a_VR 22
#define a_CUROBJ 23
#define a_OBJETO 24
#define a_COLMAX 24

#define COLORSCHEME 1
#define ColorBoton COLOR ("....." + Scheme(COLORSCHEME, 3))
#define ColorBorder COLOR (Scheme(COLORSCHEME, 3))
#define ColorOpcion COLOR SCHEME COLORSCHEME

PRIVATE nRen, nPagina, nOpcion, nAvance, nSelect, nReactivos, ;
nPagMax, nArea, nLast, nNext, nExit,
cExamen, ;
!Bloqueo, nAleatorio, nMargen, nContesta, ;
!Exit, nBusca, nSelect2
!Continua,

PUSH KEY CLEAR
nSelect = SELECT()
SET COLOR SET TO aplica

!Exit = Falso
nContesta = "0"
nAleatorio = ""
STORE 1 TO nLast, nNext, nExit, nBusca
!Continua = Cierto

If !Used( LEFT(cGuardaExa, LEN(cGuardaExa) - 4) )
PRIVATE
USE (cGuardaExa) IN 0 ORDER 0

```

```

Else
    SELECT (cGuardaExa)
    SET ORDER TO
EndIf
If !Used( LEFT(cGuion, LEN(cGuion) - 4) )
    USE (cGuion) IN 0 ORDER 0
EndIf
    SELECT (cGuion)

GOTO TOP
nReactivos = RecCount() - 1 && se supone que no esta vacio y que
existe

DIMENSION aGets[nReactivos, a_COLMAX]
= Crea_aGets()

nPagMax = aGets[nReactivos, a_PAG]

SELECT Evaluaci
SET BELL OFF
SET MEMOWIDTH TO MEMOANCHO

nAvance = 0
nPagina = 1
cExamen = SPACE(4)

IBloqueo = GuardaTodo()

DEFINE WINDOW Hoja FROM 0, 0 TO SRows() - 1, 79 ;
    COLOR SCHEME COLORSCHEME NONE
ACTIVATE WINDOW Hoja NOSHOW
@ 0, 0 TO WRows() - 1, WCols() - 1 "" ColorBorder
@ 1, WCols() - 1 SAY "" ColorBorder
@ WRows() - 2, WCols() - 1 SAY "" ColorBorder
@ 0, 0 SAY "p" ColorBorder
nMargen = INT((SCOLS) - LEN(cGuardaExa))/2
@ 0, nMargen SAY cGuardaExa ColorBorder

Do While IBloqueo
    @ 1, WCols() - 1 GET nLast PICTURE "@*IVN "
SIZE 1,1,1;
    WHEN MDown() VALID ValidLast()
ColorBoton

    @ WRows() - 2, WCols() - 1 GET nNext PICTURE
"@*IVN " SIZE 1,1,1;
    WHEN MDown() VALID ValidNext()
ColorBoton

    @ 0, 0 GET nExit PICTURE "@*IVT " SIZE 1,1,1;
    WHEN MDown() VALID ValidExit() ColorBoton

    = TraePagina(nPagina)
    READ VALID ValidRead() && CYCLE
    nSelect2 = Select()
    = GuardaPag()
    SELECT (nSelect2)

Do Case
    Case nAvance = 0
        Do SalirExa.spr
        If IExit
            Exit
        Else
            Do Contesta With 1
            EndIf
        Case nAvance = 1
            If nPagina < nPagMax
                nPagina = nPagina + 1
            Else
                ? CHR(7)
                WAIT WINDOW "INICIO"
                nPagina = 1
            EndIf
            Case nAvance = -1

```

```

If nPagina > 1
    nPagina = nPagina - 1
Else
    ? CHR(7)
    WAIT WINDOW "FINAL"
    nPagina = nPagMax
EndIf
EndCase
EndDo

USE IN (cGuion)
USE IN (cGuardaExa)
SELECT ( nSelect )
POP KEY
ON KEY
RELEASE WINDOW Hoja
RETURN

*
* Presenta los reactivos Dicotomicos
*

FUNCTION Dico
PARAMETERS nReactivo
PRIVATE cTex, nTemp
STORE 0 TO _MLINE

SEEK aGets[nReactivo, a_CLASE ] + aGets[nReactivo, a_CLAVE]
RESTORE FROM MEMO Respuestas ADDITIVE

cTex = Premisa
@ nRen, 1 SAY aGets[nReactivo, a_NUMERO] PICTURE "999.-"
If LEN(Premisa) < 254
    @ nRen, 7 SAY cTex SIZE MemLines(cTex),
MEMOANCHO
Else
    @ nRen, 7 SAY Premisa SIZE
MemLines(Premisa), MEMOANCHO
EndIf

nRen = Row()
@ nRen, 8 + MEMOANCHO SAY "("
@ nRen, 9 + MEMOANCHO GET aGets[nReactivo, a_RES0]
PICTURE "@K!";

VALID PostEval("1")
nRen = nRen + 1

a_Resp[1] = AllTrim(a_Resp[1])
a_Resp[2] = AllTrim(a_Resp[2])
nTemp = 3 + Max( Len(a_Resp[1]), Len(a_Resp[2]) )

If EMPTY(aGets[nReactivo, a_ORDEN ])
    aGets[nReactivo, a_ORDEN ] = Ordena(2)
EndIf

@ nRen + 0, 7 SAY "A" + IIF(SUBSTR(aGets[nReactivo, a_ORDEN
], 1, 1) == '1', a_Resp[1], a_Resp[2])
@ nRen + 1, 7 SAY "B" + IIF(SUBSTR(aGets[nReactivo, a_ORDEN
], 2, 1) == '1', a_Resp[1], a_Resp[2])
@ nRen + 0, 7 GET aGets[nReactivo, a_RESB] PICTURE "@*INV.;"
;
nTemp SIZE 1, nTemp + 1, 0 WHEN MDown() ColorOpcion ;
VALID PostEval("1", Cierlo)
@ nRen + 1, 8 + MEMOANCHO SAY "]" + ALLTRIM( STR(
aGets[nReactivo, a_VR]) ) + "]"
nRen = nRen + 3

RETURN

*
* Presenta los reactivos de Opción única
*

FUNCTION Unic
PARAMETERS nReactivo
PRIVATE cTex, nTemp, n, cBoton

```

```
SEEK aGets[nReactivo, a_CLASE] + aGets[nReactivo, a_CLAVE]
RESTORE FROM MEMO Respuestas ADDITIVE
```

```
cTex = Premisa
```

```
@ nRen, 1 SAY aGets[nReactivo, a_NUMERO] PICTURE "999.-"
```

```
If LEN(Premisa) < 254
```

```
@ nRen, 7 SAY cTex SIZE MemLines(cTex),
```

```
MEMOANCHO
```

```
Else
```

```
@ nRen, 7 SAY Premisa SIZE
```

```
MemLines(Premisa), MEMOANCHO
```

```
EndIf
```

```
nRen = Row()
```

```
@ nRen, 8 + MEMOANCHO SAY "("
```

```
@ nRen, 9 + MEMOANCHO GET aGets[nReactivo, a_RES0]
PICTURE "@K!";
```

```
VALID PostEval("2")
```

```
nRen = nRen + 1
```

```
If EMPTY( aGets[nReactivo, a_RES] )
```

```
aGets[nReactivo, a_RES] = AltoRes()
```

```
EndIf
```

```
nTemp = 0
```

```
For n = 1 to aGets[nReactivo, a_RES]
```

```
nTemp = Max( Len(a_Resp[n]), nTemp )
```

```
EndFor
```

```
nTemp = nTemp + 3
```

```
If EMPTY(aGets[nReactivo, a_ORDEN ])
```

```
aGets[nReactivo, a_ORDEN ] = Ordena(
```

```
aGets[nReactivo, a_RES] )
```

```
EndIf
```

```
For n = 1 to aGets[nReactivo, a_RES]
```

```
x = INT(VAL(SUBSTR(aGets[nReactivo, a_ORDEN ], n, 1)))
```

```
@ nRen + n - 1, 7 SAY CHR(64+n) + " " + a_Resp[x]
```

```
EndFor
```

```
cBoton = "@*INV "+ REPLICATE(" ", aGets[nReactivo, a_RES] - 1)
```

```
@ nRen + 0, 7 GET aGets[nReactivo, a_RESB] PICTURE cBoton
```

```
SIZE 1, nTemp + 1, 0 WHEN MDown() ColorOpcion ;
```

```
VALID PostEval("2", Cierito)
```

```
@ nRen + aGets[nReactivo, a_RES] - 1, 8 + MEMOANCHO SAY "I";
```

```
ALLTRIM( STR( aGets[nReactivo, a_VR])) + "I"
```

```
nRen = nRen + aGets[nReactivo, a_RES] + 1
```

```
RETURN
```

```
* Presenta los reactivos de Complementación *
```

```
FUNCTION Comp
```

```
PARAMETERS nReactivo
```

```
PRIVATE cTex, n, cBoton
```

```
SEEK aGets[nReactivo, a_CLASE] + aGets[nReactivo, a_CLAVE]
```

```
RESTORE FROM MEMO Respuestas ADDITIVE
```

```
cTex = Premisa
```

```
@ nRen, 1 SAY aGets[nReactivo, a_NUMERO] PICTURE "999.-"
```

```
If LEN(Premisa) < 254
```

```
@ nRen, 7 SAY cTex SIZE MemLines(cTex),
```

```
MEMOANCHO
```

```
Else
```

```
@ nRen, 7 SAY Premisa SIZE
```

```
MemLines(Premisa), MEMOANCHO
```

```
EndIf
```

```
nRen = Row()
```

```
nRen = nRen + 1
```

```
If EMPTY( aGets[nReactivo, a_RES] )
```

```
aGets[nReactivo, a_RES] = AltoRes()
```

```
EndIf
```

```
If EMPTY(aGets[nReactivo, a_ORDEN ])
```

```
aGets[nReactivo, a_ORDEN ] = Ordena(
```

```
aGets[nReactivo, a_RES] )
```

```
EndIf
```

```
For n = 1 to aGets[nReactivo, a_RES]
```

```
x = INT( VAL( SUBSTR( aGets[nReactivo, a_ORDEN ], n, 1) ) )
```

```
@ nRen + n - 1, 7 SAY CHR(64+n) + " " + a_Resp[x]
```

```
EndFor
```

```
cBoton = "@*INV "+ REPLICATE(" ", aGets[nReactivo, a_RES] - 1)
```

```
@ nRen + aGets[nReactivo, a_RES], 7 GET aGets[nReactivo,
```

```
a_ORDEN2];
```

```
PICTURE "@K!" SIZE 1, 50 DEFAULT " "
```

```
;VALID Posteval("3")
```

```
@ nRen + aGets[nReactivo, a_RES], 8 + MEMOANCHO SAY "I" +
```

```
ALLTRIM( STR( aGets[nReactivo, a_VR])) + "I"
```

```
nRen = nRen + aGets[nReactivo, a_RES] + 2
```

```
RETURN
```

```
* Presenta los reactivos de Opción Múltiple *
```

```
FUNCTION Mult
```

```
PARAMETERS nReactivo
```

```
PRIVATE cTex, nTemp, n, cBoton
```

```
SEEK aGets[nReactivo, a_CLASE] + aGets[nReactivo, a_CLAVE]
```

```
RESTORE FROM MEMO Respuestas ADDITIVE
```

```
cTex = Premisa
```

```
@ nRen, 1 SAY aGets[nReactivo, a_NUMERO] PICTURE "999.-"
```

```
If LEN(Premisa) < 254
```

```
@ nRen, 7 SAY cTex SIZE MemLines(cTex),
```

```
MEMOANCHO
```

```
Else
```

```
@ nRen, 7 SAY Premisa SIZE
```

```
MemLines(Premisa), MEMOANCHO
```

```
EndIf
```

```
nRen = Row()
```

```
@ nRen + 1, 6 + MEMOANCHO SAY "("
```

```
@ nRen + 1, 10 + MEMOANCHO SAY "("
```

```
@ nRen + 1, 7 + MEMOANCHO GET aGets[nReactivo, a_RES0]
```

```
PICTURE "@K!";
```

```
VALID PostEval("4")
```

```
@ nRen + 1, 11 + MEMOANCHO GET aGets[nReactivo, a_RES1]
```

```
PICTURE "@K!";
```

```
VALID PostEval("4")
```

```
nRen = nRen + 1
```

```
If EMPTY( aGets[nReactivo, a_RES] )
```

```
aGets[nReactivo, a_RES] = AltoRes()
```

```
EndIf
```

```
If EMPTY(aGets[nReactivo, a_ORDEN ])
```

```
aGets[nReactivo, a_ORDEN ] = Ordena(
```

```
aGets[nReactivo, a_RES] )
```

```
EndIf
```

```
nTemp = 0
```

```
For n = 1 to aGets[nReactivo, a_RES]
```

```
nTemp = Max( Len(a_Resp[n]), nTemp )
```

```
EndFor
```

```
nTemp = nTemp + 3
```

```
For n = 1 to aGets[nReactivo, a_RES]
```

```
x = INT(VAL(SUBSTR(aGets[nReactivo, a_ORDEN ], n, 1)))
```

```
@ nRen + n - 1, 7 SAY CHR(64+n) + " " + a_Resp[x]
```

```
EndFor
```

```
cBoton = "@*INV "+ REPLICATE(" ", aGets[nReactivo, a_RES] - 1)
```

```
@ nRen, 7 GET aGets[nReactivo, a_RESB] PICTURE cBoton ;
```

```

SIZE 1, nTemp + 1, 0 WHEN MDown() ColorOpcion ;
VALID PostEval("4", Cierto)
@ nRen + aGets[nReactivo, a_RES] - 1, 8 + MEMOANCHO SAY "I" +
ALLTRIM( STR( aGets[nReactivo, a_VR])) + "I"

```

```

nRen = nRen + aGets[nReactivo, a_RES] + 1
RETURN

```

```

* Presenta los reactivos de Ordenamiento

```

```

FUNCTION Orde
PARAMETERS nReactivo
PRIVATE cTex, nTemp, n, cBoton

```

```

SEEK aGets[nReactivo, a_CLASE ] + aGets[nReactivo, a_CLAVE]
RESTORE FROM MEMO Respuestas ADDITIVE

```

```

cTex = Premisa
@ nRen, 1 SAY aGets[nReactivo, a_NUMERO] PICTURE "999.-"
If LEN(Premisa) < 254
@ nRen, 7 SAY cTex SIZE MemLines(cTex),

```

```

MEMOANCHO
Else
@ nRen, 7 SAY Premisa SIZE
MemLines(Premisa), MEMOANCHO
Endif

```

```

nRen = Row()
nRen = nRen + 1
If EMPTY( aGets[nReactivo, a_RES] )
aGets[nReactivo, a_RES] = AltoRes()
Endif

```

```

If EMPTY(aGets[nReactivo, a_ORDEN ])
aGets[nReactivo, a_ORDEN ] = Ordena(
aGets[nReactivo, a_RES] )
Endif

```

```

For n = 1 to aGets[nReactivo, a_RES]
x = INT(VAL(SUBSTR( aGets[nReactivo, a_ORDEN ] , n, 1)))
@ nRen + n - 1, 7 SAY "(" + a_Resp[x]
EndFor

```

```

For n = 1 to aGets[nReactivo, a_RES]
@ nRen + n - 1, 8 GET aGets[nReactivo, n + a_RESB]
PICTURE "@K 9";

```

```

VALID PostEval("5")
EndFor
@ nRen + aGets[nReactivo, a_RES] - 1, 8 + MEMOANCHO SAY "I" +

```

```

ALLTRIM( STR( aGets[nReactivo, a_VR])) + "I"
RETURN

```

```

* Presenta los reactivos de Relación

```

```

FUNCTION Rela
PARAMETERS nReactivo
PRIVATE cTex, nTemp, i, n, cBoton, nCarE

```

```

SEEK aGets[nReactivo, a_CLASE ] + aGets[nReactivo, a_CLAVE]
RESTORE FROM MEMO Respuestas ADDITIVE

```

```

cTex = Premisa
@ nRen, 1 SAY aGets[nReactivo, a_NUMERO] PICTURE "999.-"
If LEN(Premisa) < 254
@ nRen, 7 SAY cTex SIZE MemLines(cTex),

```

```

MEMOANCHO
Else
@ nRen, 7 SAY Premisa SIZE
MemLines(Premisa), MEMOANCHO
Endif

```

```

nRen = Row()
nRen = nRen + 1

```

```

If EMPTY(aGets[nReactivo, a_RELE ])
i = 1
Do While i <= 10 AND (!EMPTY( a_Resp[i, 1] ) OR
!EMPTY( a_Resp[i, 2] ))
If !EMPTY( a_Resp[i, 1] )
a_Resp[i, 1] = AITrim( a_Resp[i, 1] )
)

```

```

Endif
If !EMPTY( a_Resp[i, 2] )
a_Resp[i, 2] = AITrim( a_Resp[i, 2] )
)

```

```

Endif
i = i + 1
EndDo
i = i - 1
aGets[nReactivo, a_RELE ] = i
Endif

```

```

If EMPTY( aGets[nReactivo, a_ORDEN ] ) OR EMPTY(
aGets[nReactivo, a_ORDEN2] )
aGets[nReactivo, a_ORDEN ] = Ordena(
aGets[nReactivo, a_RELE ] )
aGets[nReactivo, a_ORDEN2] = Ordena( aGets[nReactivo,
a_RELE ] )

```

```

nEnun = 0
nResp = 0
cCarE = 0
cCarR = 0

```

```

For n = 1 to aGets[nReactivo, a_RELE ]
nCarE = INT(VAL(SUBSTR(aGets[nReactivo, a_ORDEN ] , n, 1)))
nCarR = INT(VAL(SUBSTR(aGets[nReactivo, a_ORDEN2], n, 1)))
If EMPTY( nCarE )
nCarE = 10

```

```

Endif
If EMPTY( nCarR )
nCarR = 10
Endif

```

```

If !EMPTY( a_Resp[nCarE, 1] )
@ nRen + nEnun, 7

```

```

SAY "(" + a_Resp[nCarE, 1]
nEnun = nEnun + 1
Else
aGets[nReactivo, a_ORDEN ] = STUFF(aGets[nReactivo,
a_ORDEN ] , n, 1, ' ')
Endif

```

```

If !EMPTY( a_Resp[nCarR, 2] )
@ nRen + nResp, 43 SAY CHR(65
+ nResp) + " " + a_Resp[nCarR, 2]
nResp = nResp + 1

```

```

Else
aGets[nReactivo, a_ORDEN2] =
STUFF(aGets[nReactivo, a_ORDEN2], n, 1, ' ')
Endif

```

```

EndFor
aGets[nReactivo, a_ORDEN ] =
CHRTRAN(aGets[nReactivo, a_ORDEN ], ' ');
aGets[nReactivo, a_ORDEN2] =
CHRTRAN(aGets[nReactivo, a_ORDEN2], ' ');
aGets[nReactivo, a_RES] = nResp

```

```

Else
cCarE = 0
cCarR = 0
For n = 1 to LEN( aGets[nReactivo, a_ORDEN ]
)

```

```

nCarE = INT(VAL(SUBSTR(aGets[nReactivo, a_ORDEN
], n, 1)))
If EMPTY(nCarE)
nCarE = 10
* EXIT

```

```

Endif
@ nRen + n - 1, 7 SAY "(" + a_Resp[nCarE,
1]

```

```

EndFor
For n = 1 to LEN( aGets[nReactivo,
a_ORDEN2] )
nCarR = INT(VAL(SUBSTR(aGets[nReactivo,
a_ORDEN2], n, 1)))
If EMPTY(nCarR)

```

```

nCarR = 10
* EXIT
EndIf
@aResp[nCarR, 2]
EndFor
nEnun = LEN( aGets[nReactivo, a_ORDEN] )
nResp = LEN( aGets[nReactivo, a_ORDEN2] )
EndIf

nTemp = Max( nEnun, nResp )
aGets[nReactivo, a_ALTO] = nTemp + AltoMemo()
For n = 1 to nEnun
@a nRen + n - 1, 8 GET aGets[nReactivo, n + a_RESB]
PICTURE "@K!";

VALID PostEval("6")
EndFor
aGets[nReactivo, a_CUROBJ] = Objnum(aGets[nReactivo,
a_RES0])
cBoton = "@*INV "+ REPLICATE(" ", aGets[nReactivo, a_RES] - 1)
@a nRen, 43 GET aGets[nReactivo, a_RESB] PICTURE
cBoton;
SIZE 1, 31, 0 WHEN MDown() ColorOpcion ;
VALID PostEval("6", Cierto)

@a nRen + aGets[nReactivo, a_RES] - 1, 8 + MEMOANCHO SAY "I";
+ ALLTRIM( STR( aGets[nReactivo, a_VR]) ) + "I"

nRen = nRen + nTemp + 1
RETURN

FUNCTION Presenta
PARAMETERS cClase, nNumero
If Empty(cClase ) or Empty(nNumero)
RETURN
EndIf
DO CASE
CASE cClase == "1"
DO Dico With nNumero
CASE cClase == "2"
DO Unic With nNumero
CASE cClase == "3"
DO Comp With nNumero
CASE cClase == "4"
DO Mult With nNumero
CASE cClase == "5"
DO Orde With nNumero
CASE cClase == "6"
DO Rela With nNumero
EndCase
RETURN

FUNCTION TrasPagina
PARAMETERS nPag
PRIVATE i
If nPag < 1 Or nPag > nReactivos
RETURN
EndIf
@a 1, 1 CLEAR TO WRows() - 2, WCols() - 2
nRen = MAR_SUP &&* para mover
margen superior
For i = 1 to nReactivos
If (aGets[i, a_PAG] = nPag)
= Presenta( aGets[i, a_CLASE], aGets[i,
a_NUMERO])
EndIf
EndFor
= Barra()
RETURN

FUNCTION ValidNext
nAvance = 1
CLEAR READ
RETURN

FUNCTION ValidLast

```

```

nAvance = -1
CLEAR READ
RETURN

FUNCTION ValidExit
nAvance = 0
CLEAR READ
RETURN

FUNCTION GuardaTodo
PRIVATE i, nSelect

nSelect = SELECT()
SELECT (cGuardaExa)

RETURN If FBloqueo(1000)
nAleatorio = SYS(3)
Do DatosAlu.spr
PRIVATE
If !Continua
UNLOCK
RETURN Falso
EndIf

If EMPTY( RecCount() )
cExamen = "1"
cExamen = PADL(cExamen, 4, ' ')
Else
SET ORDER TO
GO BOTTOM
SCATTER FIELDS Examen
MEMVAR cExamen = cExamen + 1
VAL(ALLTRIM(m.Examen))
cExamen = cExamen + 1
cExamen = STR(cExamen,4)
EndIf

INSERT INTO (cGuardaExa) (Examen, Clave,
Pagina);
VALUES
(cExamen, nAleatorio, " 0")

For i = 1 To nReactivos
INSERT INTO (cGuardaExa) (Examen,
Pagina, Clase, Clave, Respuesta, ;
Orden, Orden2);
VALUES (cExamen, ;
(STR(aGets[i, a_PAG], 3)), ;
aGets[i, a_CLASE], ;
aGets[i, a_CLAVE], ;
aGets[i, a_RES0] + aGets[i, a_RES1] + ;
aGets[i, a_RES2] + aGets[i, a_RES3] + ;
aGets[i, a_RES4] + aGets[i, a_RES5] + ;
aGets[i, a_RES6] + aGets[i, a_RES7] + ;
aGets[i, a_RES8] + aGets[i, a_RES9], ;
aGets[i, a_ORDEN], ;
aGets[i, a_ORDEN2] )
EndFor
FLUSH
Else
WAIT WINDOW "El examen esta siendo usado
por otro usuario " NOWAIT
RETURN Falso
EndIf
UNLOCK
SELECT( nSelect )
RETURN

FUNCTION GuardaPag
PRIVATE i, nSelect
nSelect = SELECT()

```

```

If nPagina < 1
    nPagina = 1
EndIf
If nPagina > nPagMax
    nPagina = nPagMax
EndIf

For i = 1 to nReactivos
    If (aGets[i, a_PAG] = nPagina)
        Exit
    EndIf
EndFor

SELECT (cGuardaExa)
SET ORDER TO alumno
SEEK cExamen + STR(nPagina, 3)
*** ** ? aqui el bloqueo ? *** ** ** **
RETURN

If RLOCK()
    SCAN WHILE (Examen = cExamen AND
Pagina = STR(nPagina, 3) )
        REPLACE Respuesta WITH      aGets[i,
a_RES0] + aGets[i, a_RES1] + ;
a_RES3] + ;
a_RES5] + ;
a_RES7] + ;
a_RES9]
        REPLACE ORDEN WITH
aGets[i, a_ORDEN ]
        REPLACE ORDEN2 WITH
aGets[i, a_ORDEN2]
        i = i + 1
        ENDSCAN
        UNLOCK
    Else
        = Inform("La p gina " + STR(nPagina,2) + " no
pudo ser guardada")
    EndIf
    FLUSH
    SELECT( nSelect )
RETURN

```

FUNCTION Crea_aGets

```

PRIVATE i
GOTO TOP
SKIP
For i = 1 To nReactivos
    aGets[i, a_CLASE ] = CLASE
    aGets[i, a_CLAVE ] = CLAVE
    aGets[i, a_NUMERO] = i
    aGets[i, a_RES ] = 0
    aGets[i, a_RESB ] = 1
    aGets[i, a_RES0 ] = ""
    aGets[i, a_RES1 ] = ""
    aGets[i, a_RES2 ] = ""
    aGets[i, a_RES3 ] = ""
    aGets[i, a_RES4 ] = ""
    aGets[i, a_RES5 ] = ""
    aGets[i, a_RES6 ] = ""
    aGets[i, a_RES7 ] = ""
    aGets[i, a_RES8 ] = ""
    aGets[i, a_RES9 ] = ""
    aGets[i, a_ALTO ] = 0
    aGets[i, a_PAG ] = 0
    aGets[i, a_ORDEN ] = ""
    aGets[i, a_ORDEN2 ] = ""
    aGets[i, a_RELE ] = 0
    aGets[i, a_VA ] = VA
    aGets[i, a_VR ] = VR
    SKIP
EndFor

= AsignaAltos()
= AsignaPags()
RETURN

```

FUNCTION AsignaPags

```

PRIVATE nPag, i, nRenglon, MaxRen
    nPag = 1
    MaxRen = SRows() - MAR_INF - 2    && para mover
margin inferior
    nRenglon = 0
    For i = 1 To nReactivos
        nRenglon = nRenglon + aGets[i, a_ALTO] + 1
        If nRenglon > MaxRen
            nPag = nPag + 1
            nRenglon = aGets[i, a_ALTO] + 1
        EndIf
        aGets[i, a_PAG] = nPag
    EndFor
RETURN

```

FUNCTION AsignaAltos

```

PRIVATE i
Select Evaluaci
For i = 1 To nReactivos
    SEEK aGets[i, a_CLASE ] + aGets[i, a_CLAVE]
    RESTORE FROM MEMO Respuestas ADDITIVE
    If aGets[i, a_CLASE ] <> "6"
        aGets[i, a_ALTO] = AltoReac() &&
Inicia.prg
        aGets[i, a_RES ] = AltoRes()
        && Inicia.prg
    Else
        aGets[i, a_ALTO] = AltoRela() +
AltoMemo()
        aGets[i, a_RES ] = AltoRela()
    EndIf
EndFor
RETURN

```

FUNCTION Barra

```

PRIVATE nRen, cCont
    = Contesta()

@ SROWS() - MAR_INF + 3, 1 SAY "P gina " +
LTrim(STR(nPagina)) + " / " + ;
LTrim(STR(nPagMax)) + " " + ;
"Reactivos " + nContesta + " / " +
LTrim(STR(nReactivos));
+ " ";
ColorBorder
@ SROWS() - MAR_INF + 3, SCOLS() - LEN( TIME() ) - 2
SAY TIME();

```

RETURN

FUNCTION ValidRead

```

PRIVATE cTemp, nPos
Do Case
    Case InList(READKEY(), 5, 261) && flecha
abajo
        nAvance = 1
    Case InList(READKEY(), 4, 260) && flecha
arriba
        nAvance = -1
    Case InList(READKEY(), 6, 262) && p gina
arriba
        nAvance = -1
    Case InList(READKEY(), 7, 263) && p gina
abajo
        nAvance = 1
    Case InList(READKEY(), 12, 268) && Escape

```

```

nAvance = 0
Case InList(READKEY(),15, 271) && Enter
  cTemp = VarRead()
  nPos = At(" ", cTemp) + 1
  cTemp = SubStr(cTemp, nPos,
At(" ", cTemp) - nPos)
)
  cTemp = VAL( ALLTRIM( cTemp ) )
  cTemp = INT( cTemp )
  If aGets[cTemp, a_PAG] <=
nPagMax
    nAvance = 1
  EndIf
  OTHERWISE
  RETURN Falso
EndCase
RETURN
FUNCTION Contesta
PARAMETER nDonde
PRIVATE i, nSelect
nSelect = Select()
Select (cGuardaExa)
SET ORDER TO alumno
SEEK cExamen + " 1"
nTemp = 0
If Empty(nDonde)
  For i = 1 to nReactivos
    If !Empty( Respuesta )
      nTemp = nTemp + 1
    EndIf
    SKIP
  EndFor
  nContesta = LTrim(Str(nTemp))
Else
  If Type("nDonde") = "N"
    nDonde = nDonde + 1
    GOTO TOP
  SEEK cExamen + STR( nPagina, 3)
  Do While !EOF()
    If Empty(Respuesta)
      nPagina =
VAL(Pagina)
    EndIf
    EndDo
  EndIf
  Select(nSelect)
RETURN
FUNCTION Busca
Do Contesta With (nPagina)
CLEAR READ
RETURN
* ----- *
* Califica.prg
* Programa principal que realiza el proceso de califi-
* cación de los ex menes interactivos
* ----- *
#define Cierto .t.
#define Falso .f.
PRIVATE cExamen, nPuntos, nExamen, nPag, cGuion, cAlumno,
cFecha;
cGrupo, cAplica, nTotal, lContinua, cTema, cAyuda
PUSH KEY CLEAR
cTema = "CALIFICA"
cAyuda = InstAyuda(cTema)
lContinua = Cierto
cExamen = ""
nPuntos = 0
nExamen = 1
nPag = " 0"
nTotal = 0

```

```

Do While lContinua
Do SelfFile.spr With cExamen, ".DBE", "Seleccione el examen"
If Empty(cExamen) Or !lContinua
  RETURN
EndIf
If !Used(cExamen)
  USE (cExamen - ".DBE") IN 0 ORDER 0
EndIf
If RecCount(cExamen) <= 1 And Inform("El examen est
vacío")
  USE IN ( cExamen )
  LOOP
EndIf
EXIT
EndDo
CREATE CURSOR Examen (Guion C(12), Alumno C(50),
Absoluto N(5,2), Relativo N(5,2), ;
Grupo C(8), plico C(8), Fecha D)
SELECT (cExamen)
If !("CLASE+CLAVE INTO EVALUACI" $ set ("relation"))
  SET RELATION TO Clase + Clave INTO Evaluaci
ADDITIVE
EndIf
If !("CLAVE INTO BITACORA" $ set ("relation"))
  SET RELATION TO Clave INTO Bitacora ADDITIVE
EndIf
  SET ORDER TO Alumno
GOTO TOP
  WAIT WINDOW "Procesando calificaciones. Alumno " +
STR(nExamen,4) NOWAIT
  Do While !EOF()
  WAIT WINDOW "Procesando calificaciones. Alumno " +
STR(nExamen,4) NOWAIT
  If Examen == STR(nExamen, 4) And Pagina = " 0"
    cAlumno = SUBSTR(Bitacora.Datos, 1, 50)
    cGrupo = PADR(SUBSTR(Bitacora.Datos, 51,
50),8)
    cElabora = PADR(SUBSTR(Bitacora.Datos,
101, 50),8)
    cFecha
    =
CTOD(PADR(SUBSTR(Bitacora.Datos, 151, 50),10))
    cGuion = PADR(SUBSTR(Bitacora.Datos,
201, 50),12)
  EndIf
  SKIP
  nVA = 0
  nVR = 0
  nTotal = 0
  Do While Pagina <> " 0" And !EOF()
    nPuntos = Evaluax(Clase, Orden, Orden2,
Respuesta)
    nVA = nPuntos * IIF(Empty(Evaluaci.VA), 1,
Evaluaci.VA) + nVA
    nVR = nPuntos * IIF(Empty(Evaluaci.VR), 1,
Evaluaci.VR) + nVR
    nTotal = Evaluaci.VR + nTotal
    REPLACE VA_O WITH nPuntos *
IIF(Empty(Evaluaci.VA), 1, Evaluaci.VA)
    REPLACE VR_O WITH nPuntos *
IIF(Empty(Evaluaci.VR), 1, Evaluaci.VR)
    SKIP
  EndDo
  INSERT INTO Examen ;
VALUES (cGuion, cAlumno, nVA, nVR/nTotal*100,
cGrupo, cElabora, cFecha)
  nExamen = nExamen + 1
EndDo
WAIT CLEAR
Select(cExamen)
USE
Do Califica.spr
POP KEY
= InstAyuda(cAyuda)
RETURN
* ----- *
Función que obtiene las respuestas correctas y compara con las

```

respuestas obtenidas, regresando un valor entre 0 y 1.
cTipo = Tipo de reactivo

cTexE = Cadena que contiene el orden en el que desplegaron
la repuestas en el examen

cTexR = Cadena que contiene el orden en el que se desplegó
la segunda columna en los exámenes de

Relacion

cExaResp = Cadena que contiene las respuestas del alumno

FUNCTION ReacResp

PARAMETER cTipo, cTexE, cTexR, cExaResp

PRIVATE i, cCarE, cCarR, cResp, cResp2, cVal, nAciertos, cCarC

IF EMPTY(cTipo) OR EMPTY(cTexE) OR EMPTY(cExaResp)

RETURN 0

ENDIF

IF !BETWEEN(cTipo, "1", "6")

RETURN 0

ENDIF

IF cTipo = "6" AND EMPTY(cTexR)

RETURN 0

ENDIF

DO CASE

CASE INLIST(cTipo, "1", "2")

cCorrecto = CHR(AT("1", cTexE) + 64)

IF cCorrecto == ALLTRIM(cExaResp)

RETURN 1

ENDIF

CASE cTipo = "3"

cCorrecto = "1"

IF cCorrecto == ALLTRIM(cExaResp)

RETURN 1

ENDIF

CASE cTipo = "4"

cVal = 0

cCorrecto = CHR(AT("1", cTexE) + 64)

cCorrecto2 = CHR(AT("2", cTexE) + 64)

cCorrecto = cCorrecto + cCorrecto2

IF INLIST(SUBSTR(cCorrecto, 1, 1),

SUBSTR(cExaResp, 1, 1), SUBSTR(cExaResp, 2, 1))

cVal = 0.5

ENDIF

IF INLIST(SUBSTR(cCorrecto, 1, 2),

SUBSTR(cExaResp, 1, 1), SUBSTR(cExaResp, 2, 1))

cVal = cVal + 0.5

ENDIF

RETURN cVal

CASE cTipo = "5"

cCorrecto = cTexE

nAciertos = 0

FOR i = 1 TO LEN(ALLTRIM(cCorrecto))

cCarC = SUBSTR(cCorrecto, i, 1)

cCarR = SUBSTR(cExaResp, i, 1)

IF cCarC == cCarR

nAciertos = nAciertos +

1

ENDIF

NEXT

RETURN

(nAciertos / LEN(ALLTRIM(cCorrecto)))

CASE cTipo = "6"

cCorrecto = ""

nAciertos = 0

FOR i = 1 TO LEN(ALLTRIM(cTexE))

cCarE = SUBSTR(cTexE, i, 1)

cCarR = CHR(AT(cCarE, cTexR) +

64)

IF !BETWEEN(cCarR, "A", "J")

cCarR = ""

ENDIF

cCorrecto = cCorrecto + cCarR

ENDFOR

FOR i = 1 TO LEN(cCorrecto)

cCarC = SUBSTR(cCorrecto, i, 1)

cCarR = SUBSTR(cExaResp, i, 1)

IF cCarC == cCarR

nAciertos = nAciertos +

1

ENDIF

NEXT

RETURN (nAciertos / LEN(cCorrecto))

ENDCASE

RETURN 0