



INSTITUTO POLITÉCNICO NACIONAL
CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN



**“PROGRAMAS DE CÓMPUTO PARA LA
GENERACIÓN AUTOMÁTICA DE EXÁMENES MATEMÁTICOS
Y LA EVALUACIÓN EN EL ENTORNO DE INTERNET”**

**TESIS QUE PARA OBTENER EL GRADO DE
MAESTRO EN CIENCIAS EN COMPUTACIÓN ELECTRÓNICA
PRESENTA:**

Lic. Jorge Arturo Reyes Bonilla

Directores de tesis:
M. en C. Leopoldo A. Galindo Soria
Dr. Oscar Camacho Nieto

México D. F., junio 2008

Agradecimientos:

A mi maestro el M. en C. Leopoldo Galindo Soria por el gran apoyo que me ha dado en los momentos más difíciles de mi vida, así por los conocimientos y la confianza que durante años he recibido de él.

Al Dr. Oscar Camacho Nieto por su apoyo para concluir la maestría.

A mis sinodales por sus correcciones, sus preguntas y observaciones.

Dedicatoria:

A quienes en el futuro, con el deseo de aprender, leerán esta obra.

ÍNDICE

Índice.....	i
Índice de Figuras.....	iv
Índice de Cuadros y Tablas.....	v
Cuadros.....	vi
Tablas.....	vii
Abreviaturas Empleadas.....	ix
Símbolos Empleados.....	x
Glosario.....	xiv
 Capítulo I.-	
Justificación Objetivos y Metodología.....	1
Contenido del capítulo.....	1
1.1 Antecedentes.....	2
1.2 Planteamiento del problema.....	3
1.3 Objetivos del proyecto de tesis.....	3
1.3.1 Objetivo general.....	3
1.3.2 Objetivos específicos.....	3
1.4 Justificación de la tesis.....	4
1.5 Beneficios esperados.....	4
1.6 Alcances de la tesis.....	5
1.7 Metodología del proyecto de tesis.....	7
1.8 Organización de la tesis.....	11
 Capítulo II.-	
Marco Contextual y Conceptual.....	13
Contenido del capítulo.....	13
2.1 Marco contextual.....	14
2.1.1 Desarrollo histórico de la educación a distancia.....	14
2.1.2 E-learning.....	15
2.1.3 Educación a distancia y “e-learning” en México.....	21
2.2 Marco conceptual.....	24
2.2.1 Pirámide conceptual.....	25
2.2.2 Breve descripción de los términos.....	27
Resumen del capítulo.....	30

Capítulo III.-

Análisis de la situación actual.....	31
Contenido del capítulo.....	31
3.1 La evaluación en la educación.....	32
3.2 Propuestas actuales de evaluación en la formación en línea (OnLine).....	38
3.2.1 Plataformas internacionales.....	38
I) Blackboard.....	38
II) Moodle.....	39
III) Hot Potatoes.....	41
IV) Claroline.....	42
V) Maple.....	42
VI) Knowledge Presenter®2006.....	42
VII) Otros.....	44
3.2.2 La experiencia nacional.....	44
I) EVA y SOFÍA en el IPN.....	44
II) COSNET.....	47
III) La experiencia de la Maestría en Ingeniería de Sistemas.....	49
Resumen del capítulo.....	51

Capítulo IV.-

Arquitectura de los programas de autoevaluación

“clásico” y “modificado”.....	53
Contenido del capítulo.....	53
4.1 Descripción general.....	54
4.1 Descripción Técnica.....	54
4.2.1 Sobre los lenguajes utilizados.....	54
4.2.2 Peculiaridades de Javascript. Dificultades en la modelación.....	57
4.2.3 Sobre la plataforma utilizada.....	60
4.3 La vista de los casos de uso.....	60
4.4 La vista física.....	60
4.5 Dependencias del componente aecap _x .htm.....	68
4.6 Elaboración de los diagramas de actividad.....	74
4.6.1 Diagramas de actividad del proceso de autoevaluación.....	74
4.6.2 Diagrama de actividad de la función Estadist().....	76
4.6.3 Diagrama de actividad de la función Estadist2().....	80
Resumen del capítulo.....	86

Capítulo V.-

Arquitectura del programa “Generador

Aleatorio de Exámenes Matemáticos”..... 87

Contenido del capítulo..... 87

El programa “Generador Aleatorio de Exámenes Matemáticos”
(GAEM)..... 88

5.2 Descripción técnica..... 89

5.2.1 Los lenguajes de programación utilizados..... 89

5.2.2 Sobre la plataforma utilizada..... 90

5.3 La vista de los casos de uso..... 90

5.4 La vista física..... 91

5.5 Dependencias del componente exam.htm..... 93

5.6 Dependencias del componente Tar_x.htm..... 96

5.7 Diagramas de actividad de la función Azar()..... 99

5.8 La función Preguntas()..... 102

5.9 Diagrama de actividad de la función Examen()..... 103

Resumen del capítulo..... 106

Capítulo VI.-

Resultados obtenidos..... 107

Contenido del capítulo..... 107

6.1 Resultados de los programas de autoevaluación
“clásico” y “modificado”..... 108

6.1.1 Resultados del programa de autoevaluación "clásico"..... 108

6.1.2 Resultados del programa de autoevaluación "modificado"... 109

6.2 Resultados del “Generador Aleatorio de Exámenes
Matemáticos”..... 110

Resumen..... 114

Capítulo VII.-

Conclusiones y trabajos futuros..... 115

Contenido del capítulo..... 115

7.1 Conclusiones..... 116

7.2 Trabajos futuros..... 121

Referencias Bibliográficas..... 123

Anexos.

Anexo I. **Componente aevcap.htm**..... I.1

Anexo II. **Componente M2.js**..... II.1

Anexo III. **Componente exams.htm**..... II.1

Anexo IV. **Componente Tar_x.htm**..... IV.1

Anexo V. **Función Preguntas() del componente bas1.js**..... V.1

Anexo VI. **Función Azar() del componente pazar.js** Azar..... VI.1

Anexo VII. **Función Examen() del componente exams.js**..... VII.1

Manual de Operación..... a

ÍNDICE DE FIGURAS

Figura 1.1 Algunos aspectos que deben ser evaluados.....	5
Figura 1.2 Algunos actores y casos de uso fundamentales en el proceso de evaluación.....	6
Figura 1.3 UP coordina las múltiples cadenas de trabajo de un proyecto de software.....	8
Figura 1.4 La relación de la arquitectura de un sistema de software y los casos de uso según RUP.....	9
Figura 2.1 Modalidades de la educación a distancia.....	17
Figura 2.2 Educación “online” parte de la educación por medios electrónicos.....	18
Figura 2.3 La evaluación “e-learning” y su modalidad “Online”.....	18
Figura 2.4 Modalidades de plataformas para la gestión de la enseñanza en línea.....	21
Figura 2.5 Estructura de la pirámide conceptual.....	26
Figura 3.1 Categorías de la taxonomía de Bloom.....	32
Figura 3.2 Los módulos de Blackboard.....	39
Figura 3.3 Módulos constituyentes de Moodle.....	40
Figura 3.4 Ejemplos de software para la enseñanza asistida por computadora.....	42
Figura 3.5 Reactivos de selección múltiple creados con la aplicación Flash llama KwnoledegePresenter®2006.....	43
Figura 3.6 Los espacios de Eva	45
Figura 3.7 Los espacios de SOFÍA.....	46
Figura 3.8 Módulos del examen en línea de COSNET.....	48
Figura 4.1 Casos de uso del programa de autoevaluación “clásico”.....	61
Figura 4.2 Casos de uso del programa de autoevaluación “modificado”.....	62
Figura 4.3 Paquetes del sistema de autoevaluación.....	63
Figura 4.4 Paquetes y componentes del programa de autoevaluación “clásico”.....	64
Figura 4.5 Paquetes y componentes del programa de autoevaluación “modificado”.....	66
Figura 4.6 Diagrama de componentes (completo) del programa de autoevaluación modificado.....	67
Figura 4.7 Objetos del componente aevcap_x.htm.....	68
Figura 4.8 Objetos del componente M2.js.....	73

Figura 4.9 Diagrama de actividades para el despliegue el examen de falso/verdadero y opción múltiple.....	75
Figura 4.10 Primera parte del diagrama de actividades de la función Estadist().....	77
Figura 4.11 Segunda parte del diagrama de actividad de la función Estadist().....	78
Figura 4.12 Diagrama UML describiendo la actividad de la función Estadist2() (primera parte).....	81
Figura 4.13 Diagrama UML describiendo la actividad de la función Estadist2() (segunda parte).....	82
Figura 5.1 Diagrama de casos de uso del GAEM.....	91
Figura 5.2 La vista física del programa “Generador Aleatorio de Exámenes Matemáticos”.....	92
Figura 5.3 Objetos y componentes de la página Web “exams.htm”.....	93
Figura 5.4 Objetos asociados al componente tar1.htm.....	97
Figura 5.5 Objetos JavaScript que participan en la generación aleatoria de exámenes matemáticos.....	98
Figura 5.6 Diagrama de actividad de la función azar(num, max).....	100
Figura 5.7 Código JavaScript de la función Exam(P, Z).....	105
Figura 6.1 El examen de autoevaluación “clásico”.....	109
Figura 6.2 La ventana con el resultado de la autoevaluación.....	110
Figura 6.3 Examen de opción múltiple para la evaluación automatizada.....	111
Figura 6.4 Resultados de la autoevaluación y la retroalimentación.....	112
Figura 6.5 Página Web con los accesos a los exámenes.....	113
Figura 6.6 Página Web con el botón que activa el GAEM.....	114
Figura 6.7 Muestra 1 del examen obtenido con el GAEM.....	115
Figura 6.8 Muestra 2 del examen obtenido con el GAEM.....	116
Figura 6.9 Reactivo donde se realiza interactividad con el usuario.....	117
Figura 6.10 Reactivo donde se observa el desplazamiento de una línea.....	117

Figura a. El examen de evaluación "clásica"	Op2
Figura b. La ventana con el resultado de la autoevaluación.....	Op3
Figura c. Examen de opción múltiple para la evaluación automatizada.....	Op4
Figura d. Resultado de la evaluación y la retroalimentación.....	Op5
Figura e. Página Web con los accesos a los exámenes.....	Op6
Figura f. Página Web con el botón que activa el GAEM.....	Op7
Figura g. Muestra 1 del examen obtenido con el GAEM.....	Op8
Figura h. Muestra 2 del examen obtenido con el GAEM.....	Op9
Figura i. Reactivo se realiza interactividad con el usuario. Con el apuntador del ratón se desplaza una línea (reactivo multimedia).....	Op10
Figura j. Reactivo donde se observa el desplazamiento de una línea.....	Op10

ÍNDICE DE CUADROS Y TABLAS

Cuadros:

Cuadro 3.1 Reactivo de respuesta corta.....	34
Cuadro 3.2 Reactivo de opción múltiple de respuesta única.....	35
Cuadro 3.3 Reactivo de opción múltiple de respuesta múltiple.....	35
Cuadro 3.4 Reactivo de correspondencia.....	36
Cuadro 3.5 Reactivo de identificación.....	37
Cuadro 4.1 En XML los elementos del script deben ser incluidos entre la sección CDATA.....	56
Cuadro 4.2 Fragmento del documento HTML indicando las rutas de acceso al código de las hojas de estilo, del programa JavaScript y el “background” del documento HTML.....	65
Cuadro 4.3 Código de la hoja de estilo.....	65
Cuadro 4.4 Código de los objetos	65
Cuadro 4.5 Fragmento del código del examen de falso/verdadero (inicio).....	69
Cuadro 4.5 Fragmento del código del examen de falso/verdadero (final).....	70
Cuadro 4.6 Código HTML de las entradas de la forma con las opciones de preguntas.(inicio).....	70
Cuadro 4.6 Código HTML de las entradas de la forma con las opciones de preguntas (final).....	71
Cuadro 4.7 Objetos Globales (públicos) en el script.....	74
Cuadro 4.8 Programa en lenguaje JavaScript para la autoevaluación automatizada “clásico”.....	79
Cuadro 4.9 Código JavaScript de la función Estadist2() (inicio).....	83
Cuadro 4.9 Código JavaScript de la función Estadist2() (final).....	84
Cuadro 4.10 Función que se declara al pulsar el botón “REINTENTAR”.....	85
Cuadro 5.1 Código con las referencias a los componentes hojas de estilo, al objeto td{ } y al componentes “tapiz7.jpg”.....	94
Cuadro 5.2 Celda de la tabla con la referencia a la página del EXAMEN1, el documento tar1.htm.....	95
Cuadro 5.3 Código del archivo tar1.htm con las referencias al script (inicio).....	96
Cuadro 5.3 Código del archivo tar1.htm con las referencias al script (final).....	91
Cuadro 5.4 La forma <form> que define la entrada del botón que activará el GAEM.....	97
Cuadro 5.5 Código JavaScript de la función azar(num, max).....	101
Cuadro 5.6 Código JavaScript de la función Preguntas().....	102
Cuadro 5.7 Código JavaScript de la función Examen().....	104
Cuadro 6.1 Fragmento de código de la función Estadist2().....	109

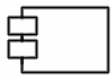
Tablas:

Tabla 1.1 Los pasos de la investigación de la presente tesis.....	10
Tabla 3.1 Algunas herramientas de software para la evaluación y su dirección electrónica.....	44
Tabla 3.2 Capacidades de evaluación en diferentes plataformas y herramientas de software.....	50
Tabla 4.1 Lenguajes y plataformas utilizadas en la ejecución de los programas.....	60
Tabla 5.1 Lenguajes y plataformas utilizadas para ejecutar el programa GAEM.....	90
Tabla 6.1 Sistemas operativos y navegadores con los cuales se probó la ejecución del programa de evaluación "clásico".....	108
Tabla 6.2 Sistemas operativos y navegadores con los cuales se probó la ejecución del programa Generador Aleatorio de Exámenes Matemáticos.....	110
Tabla 6.3 Sistemas operativos y navegadores con los cuales se probó la ejecución del programa de evaluación "modificado".....	111
Tabla 6.4 Comparación de GAEM y desarrollos no nacionales, difundidos en Internet.....	113

Abreviaturas Empleadas

AICC.	Aviation Industry Computer Based Training Committee (Comité de Entrenamiento Basado en Computadora de la Industria de la Aviación)
CAA.	Computer Assisted Assessment (Evaluación asistida por computadora)
CBA.	Computer- Based Assessment (Evaluación Basada en Computadora)
CBT.	Computer Based Training (Entrenamiento basado en Computadora)
CMS.	Content Management System (Sistema de Gestión de Contenidos)
COSNET.	Consejo del Sistema Nacional de Educación Tecnológica
EVA.	Espacios Virtuales de Aprendizaje
SOFÍA.	Sistema Orientado Fundamentalmente a la Individualidad del Aprendizaje.
LMS.	Learning Management System (Sistema de Gestión de Aprendizaje)
LCMS.	Learning Content Management System (Sistema de Gestión de Contenidos y Aprendizaje)
SCORM.	Sharable Content Object Reference Model (Modelo de Referencia de Objetos de Contenido Distribuible)
UML	Unified Modeling Language (Lenguaje de Modelado Unificado)
WAA.	Web- Assisted Assessment (Evaluación Asistida por Web)
WBA.	Web-Based Assessment (Evaluación Basada en el Web)
WBT.	Web Based Training (Entrenamiento Basado en Web)
XHTML	eXtended Hypertext Markup Language (Lenguaje de Marcado de Hipertexto Extendido)
XML.	eXtended Markup Language (Lenguaje de Marcado Extendido)

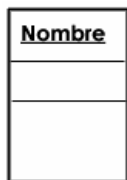
Símbolos Empleados



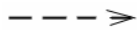
Componente



Paquete



Objeto



**Relación de dependencia
o de uso**



Agregación



Relación de herencia



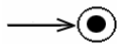
Decisión



Diagrama de estado



Estado inicial



Estado final



Interfaz

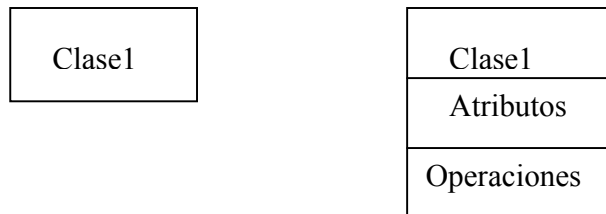
Glosario de Términos

- AICC** Aviation Industry CBT (Computer Based Training) Committee (Comité de Entrenamiento Basado en Computadora de la Industria de la Aviación). Es una asociación internacional de entrenamiento de profesionales basado en tecnología. Establece lineamientos para la industria de aviación a partir de las tecnologías CBT (Computer Based Training, Entrenamiento Basado en Computadora) y WBT (Web Based Training, Entrenamiento Basado en el Web) .
- Arquitectura** Estructura organizativa de un sistema que incluye su descomposición en partes, mecanismos de interacción y principios de guía que proporcionen información sobre el diseño del mismo. La arquitectura incluye la selección de los elementos estructurales y las interfaces mediante las que se conectan, la organización a gran escala de los elementos estructurales y la topología de su conexión, su comportamiento en las colaboraciones entre dichos elementos. [Rumbaugh, 2000]
- Artefacto** Pieza de información que es utilizada o producida por un desarrollo de software [Booch, 1999].

Clase

Definición de atributos o propiedades (variables) y métodos. Los métodos realizan operaciones. La definición de a clase no reserva memoria, sus instancias sí. Una clase describe un conjunto de objetos con un rol y roles equivalentes en un sistema. [Stevens, 2002]

En UML (Unified Modeling Language, Lenguaje Unificado de Modelado), una clase se representa con un rectángulo, dentro del cual se anota el nombre de la clase. También se puede ampliar la descripción de clase usando compartimientos: El segundo compartimiento lista los atributos, el tercero las operaciones



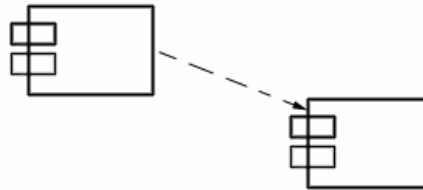
Representación de una clase en UML.

Componente De acuerdo a la documentación de UML (Unified Modeling Language, Lenguaje Unificado de Modelado), es “parte de implementación de un sistema que se puede distribuir, que incluye código software (fuente binario o ejecutable), pero que incluye también documentos de negocio, etc., en un sistema humano” [Stevens, 2002].

Hay varios tipos comunes de componentes; teniendo en cuenta por ejemplo el proceso de compilación, un componente puede ser

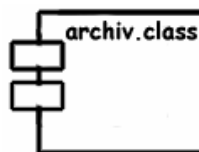
- Código fuente. Archivo con el código de una clase, que tiene que estar disponible cuando se compila
- Código objeto binario. Como las bibliotecas de clase, que depende de cualquier código objeto con el cual tiene que estar enlazado para formar un programa ejecutable.
- Una aplicación ejecutable. Una hoja de cálculo electrónica, un gestor de base de datos.

Un componente se representa conforme al Lenguaje Unificado de Modelado (UML) con un rectángulo y dos más pequeños adheridos a él. Las relaciones entre componentes se indican con flechas de líneas punteadas.



Componentes y dependencia.

El mismo símbolo con líneas gruesas indica un componente ejecutable (Booch,1999); por ejemplo un archivo de extensión .class de java.



Un componente ejecutable, su nombre se anota dentro del icono.

UML permite la personalización de iconos; por ejemplo un archivo de extensión .java (código fuente), una página Web, se puede representar del modo siguiente:



**Representación personalizada de un
archivo fuente, una página Web y una imagen.**

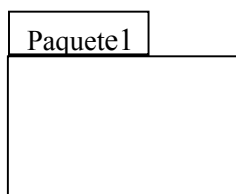
En el caso de notación personalizada los nombres van fuera del icono

Interfaz

Es una colección de operaciones que especifican un servicio de una clase o componente. Por lo tanto la interfaz describe el comportamiento visible externamente de ese elemento.

Paquete

Un paquete es una colección de elementos del modelo. Los elementos contenidos en un paquete pueden dibujarse dentro del símbolo del mismo. En UML (Unified Modeling Language, Lenguaje Unificado de Modelado), un paquete se representa con un rectángulo con otro más pequeño (Figura 0.3) en el cual se suele anotar su nombre [Stevens, 2002].



Representación UML de un paquete

SCORM

Sharable Content Object Reference Model (Modelo de Referencia de Objetos de Contenido Distribuible). Es una colección de estándares y especificaciones para el "e-learning" basado en el Web. Usa XML (eXtended Markup Language) y se basa en los resultados de AICC (CBT), IEEE y Ariadne.

UML	Unified Modeling Language (Lenguaje Unificado de Modelado), es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura decisiones y conocimiento sobre los sistemas que se deben construir. Se usa para entender, diseñar, hojear, configurar, mantener y controlar la información sobre tales sistemas. Esta pensado para usarse con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios. [Rumbaugh, 2000]
Vista	Es un subconjunto de UML (Unified Modeling Language, Lenguaje Unificado de Modelado), que modela construcciones que representan un aspecto del sistema. En el nivel superior las vistas se dividen en tres áreas: Clasificación estructural, comportamiento dinámico y gestión del modelo.
XML	Extensible Markup Language (Lenguaje de Marcado Extensible). Pretende preocuparse más por el significado de los datos que por su aspecto (como HTML). XML es una subserie simplificada del SGML (Standar Generalized Markup Language, Lenguaje Estandarizado y Generalizado de Marcado). Pretende distinguir entre abstracción e información de reproducción. XML es una notación universal (formato de datos) que permite a las computadoras almacenar y transferir datos inteligentes por cualquier sistema de computadora, permitiendo una integración en bases de datos de los documentos [Goldfarb, 1999].

“PROGRAMAS DE CÓMPUTO PARA LA GENERACIÓN AUTOMÁTICA DE EXÁMENES MATEMÁTICOS Y LA EVALUACIÓN EN EL ENTORNO DE INTERNET”

Resumen

El producto de este trabajo son tres programas de cómputo: dos de autoevaluación y un tercero es un programa generador de exámenes matemáticos de forma aleatoria, todos ellos aplicables al entorno de Internet.

En la tesis se ha realizado una investigación histórico- laboral, para ubicar el campo de aplicación computacional de los programas propuestos y la modalidad educativa en la que se insertan.

Se trata el problema de la evaluación en la educación y las características de los reactivos de las pruebas pedagógicas, las cuales deben ser consideradas al desarrollar buenos reactivos en los sistemas de evaluación automatizada “Online”.

Se examina varios casos de programas de cómputo de gestión del conocimiento de origen nacional como extranjero, tales como Moodle, BlackBoard, EVA, Sofia; enfatizando sus capacidades de evaluación automatizada y en sus modalidades de pruebas pedagógicas. También se revisan herramientas para la elaboración de reactivos de pruebas pedagógicas entre ellas KnowledgePresenter de Flash.

Dicha revisión, no sólo contribuyó a caracterizar la situación actual de estas herramientas computacionales, sino también para establecer las limitaciones de los sistemas nacionales de evaluación, constreñidos muchas veces a una prueba de opción múltiple y a la evaluación de los aprendizajes.

Los programas de cómputo desarrollados, son una buena propuesta para el caso de evaluación del aprendizaje; aportando la posibilidad de retroalimentar al alumno y no sólo entregarle una contabilidad de errores. Otra aportación es la de permitir la generación de exámenes multimedia. Sin embargo, deben incorporarse a sistemas de evaluación “robustos”, en los cuales ésta se dirija no sólo a los alumnos, sino también a los profesores, autoridades, instituciones y la sociedad, vinculando los resultados a un manejador de bases de datos.

En el presente documento se expone la estructura del proyecto de tesis, los procesos investigación realizados para su conclusión, así como el marco conceptual que la sustenta. Se ha dado importancia a la modelación de los programas de cómputo propuestos, tanto de la arquitectura, como de las funciones desarrolladas; todo ello en términos de UML (Unified Modeling Language, Lenguaje de Modelado Unificado). Tales actividades no sólo permiten entender el software, sino también facilitar su posterior evolución y adecuación o inserción en un sistema de evaluación de mayor alcance y versatilidad.

“COMPUTER PROGRAMS FOR THE AUTOMATIC GENERATION OF MATHEMATICS EXAMS AND EVALUATION ON INTERNET ENVIRONMENT”

Abstract

The product of this work are three computer programs: Two are autoevaluation and third one it is a generating program of mathematical examinations of random form, all of them applicable to the Internet environment.

In the thesis an historical and labor research has been realized, for to be situated the field of computational applications of proposed programs and education modality in the one that is inserted.

The problem of the evaluation treats itself in the education and the characteristic of the reactivities of pedagogical proofs; which must be considered on having for developed good reactivities in the systems of automated evaluation “Online”.

One exams several case of computer programs of management knowledge of national origin as foreign as *Moodle*, *BlackBoard*, *EVA*, *SOFIA*; emphasizing his capacities of automated evaluations and his modalities of pedagogical proofs. Also checks tools for elaboration of reactivities of pedagogical proofs, *KnowledgePresenter* of *Flash* between them.

The review, not only contributed to characterize the current situation of these tools and programs of management learning, but also to establish the limitations of the national systems of evaluation, restricted often to test of multiple option and evaluation of the learnings.

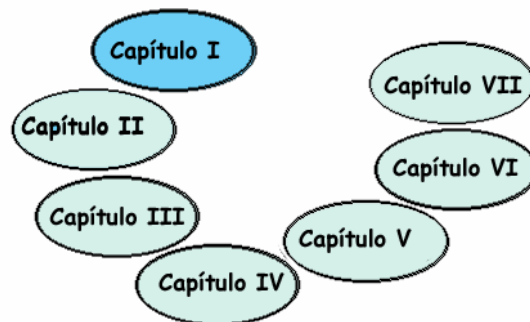
The computer programs developed, are good offer for the case of evaluation of the learning; them contributing to the possibility of feed-back for the pupil and not only to deliver him, an accounting mistake. Another contribution is it of the generation of exams allows multimedia. However, must the programs are incorporate to “robust” systems of evaluation, in which this ones the evaluation not only is direct on the student, also too the teacher, authority, institute and society, linking the results to management Data Base.

In this document explain the structure of the thesis project, research process to realized, also the conceptual frame that sustains her. One has

given importance to the modeling of computer programs proposed, so much the architecture, so development function, all in UML's (Unified Modeling Language) terms. Such activities not only allow understand the software, but too facilitate his later evolution and adequacy or insertion in a system of evaluation and adequacy in a system of evaluation of major scope und versatility.

CAPÍTULO I.-

Justificación, Objetivos y Metodología.



Contenido del capítulo:

En este capítulo se tratarán los antecedentes, planteamiento del problema, la justificación de la tesis, los objetivos, beneficios, alcances y límites del trabajo realizado. Se expone también la metodología empleada.

Señala los pasos seguidos en la investigación de la tesis, descripción de la estructura de la tesis o capitulado.

1.1 Antecedentes

En esta tesis se examinan tres programas de cómputo desarrollados para la modalidad a distancia de la Maestría en Ingeniería de Sistemas, que se impartió en la ESIME unidad Zacatenco. Los dos primeros son programas de autoevaluación automatizada creados por el autor de la tesis e incorporados en algunos de los materiales multimedia de dichos programas; de otra forma se hubiera tenido que hacer uso de alguna herramienta externa como “Hot Potatoes” o las propias de Blackboard, quedando siempre limitados en la capacidad de creación y de dominio tecnológico.

Teóricamente el proceso de evaluación debe contribuir a rectificar el rumbo del evaluado, en caso de que los resultados de ésta no sean favorables; pero la realidad es de que si la evaluación va acompañada de un registro de los resultados y ello compromete la continuidad de los evaluados, no se rectifica el rumbo, sino que más bien se reinicia, al no promoverse el alumno al siguiente curso. La "autoevaluación" **es un subconjunto de la evaluación**, pero busca más que asignar una calificación y registrarla en una base de datos (lo cual no hacen los programas propuestos), coadyuvar al autoaprendizaje del alumno.

La autoevaluación mantiene los requerimientos y rigurosidad de las pruebas pedagógicas, pero los resultados de las pruebas no se registran en una base de datos, con la finalidad de dar una calificación definitiva; en cambio, abre la posibilidad de retroalimentación y de facilitar el aprendizaje.

El desarrollo del primer programa llamado de "**autoevaluación clásica**", porque los resultados del programa, así como la naturaleza de sus reactivos, son comúnmente empleados, casi cualquier sistema de evaluación computarizado o “en línea”, cuenta con un programa que realiza tareas análogas: No va más allá de reportar qué preguntas se contestaron erróneamente y la calificación del examen; por cierto éste comúnmente consiste en reactivos de "falso-verdadero" los cuales por exclusión el alumno obtiene la respuesta verdadera.

El algoritmo aquí reportado para el “programa clásico”, es desarrollo propio e independiente del sustentante de la tesis. El valor de este primer programa desde el punto de vista de la programación, residió en que permitió hallar el algoritmo clave, para los exámenes de opción múltiple.

Partiendo de la evaluación automatizada de exámenes de opción múltiple, se incorporó la capacidad de **retroalimentar** a los alumnos: El programa es más "inteligente", le dice al alumno en caso de error, porqué razones la

opción seleccionada es incorrecta. Tal programa se conoce como de **“autoevaluación modificado”**.

Un tercer programa correspondió a las necesidades de los profesores de matemáticas, que requerían exámenes en línea pero que pudieran ser diferentes para cada alumno; de allí surgió la idea de un **“Generador Aleatorio de Exámenes Matemáticos” (GAEM)**, capaz de presentar expresiones matemáticas.

Los programas son “scripts” escritos en JavaScript, no pretenden ser más que componentes reutilizables para un sistema de evaluación más poderoso. Aunque creados por el autor de la tesis, no se presentan como un descubrimiento que, en otra parte del mundo, no se halla realizado o esté en vías de realizarse; sin embargo, son originales y resuelven necesidades propias. Es aporte, sincero a fortalecer los sistemas “Online” que se desarrollen el Instituto Politécnico Nacional.

1.2 Planteamiento del problema

Ante la demanda de educación y el avance de las tecnologías de información y comunicación (TIC), en México se ha ido avanzando en el desarrollo cursos en línea ("on line"); es decir, materiales a los cuales se puede acceder vía Internet; sin embargo, el desarrollo de sistemas de evaluación y ejercicios de autoevaluación, así como generadores de exámenes aleatorios de temas matemáticos, se halla rezagado.

La presente tesis propone dos programas de autoevaluación y un generador de exámenes matemáticos para el entorno de Internet.

1.3 Objetivos del proyecto de tesis

1.3.1 Objetivo General

Contribuir al desarrollo de un sistema de evaluación “en línea” (Online), basado en Web, para software de gestión, administración de contenidos educativos, así como del aprendizaje, en escuelas de educación superior y posgrado, en México.

1.3.2 Objetivos específicos

1. Proponer programas de corrección automática de pruebas pedagógicas y un generador de exámenes matemáticos, para el entorno de Internet.

2. Investigar las herramientas computacionales actuales desarrolladas u orientadas a los procesos de evaluación en línea para realizar un análisis y diagnóstico que permitan evidenciar las limitantes, alcances y ventajas de las propuestas presentadas.
3. Describir y modelar los desarrollos de **software para corrección automática de pruebas pedagógicas de opción múltiple, propuestos en la tesis.**
4. Describir y modelar el programa de **generación aleatoria y automatizada de exámenes matemáticos**, propuesto en la tesis.

1.4 Justificación de la tesis

Del estudio de programas de cómputo para la gestión y administración del conocimiento, desarrollados por universidades y empresas extranjeras, se desprende que es parte esencial del software para la enseñanza asistida y basada en la computadora (e-learning), los programas de evaluación (incluyendo la autoevaluación).

En México, hay intento de desarrollar sistemas de gestión y administración del conocimiento, pero aún no se logra superar en muchos casos el nivel de sistemas del tipo CMS (Content Management System, Sistema de Gestión de Contenidos); es decir, se trata de portales que dan acceso a contenidos, con muchas limitaciones; una de ellas es la de carencia de un sistema de evaluación en línea el cual comprenda diversas modalidades de ejercicios de ensayo de examen con retroalimentación; además, de la capacidad de realizar de forma automática la corrección de la pruebas pedagógicas presentadas al alumno y establecer una relación con una base de datos.

1.5 Beneficios esperados

En algunos casos como en el IPN, en lugar de enfrentar las carencias de software en el ámbito “Online”, se intenta paliar las deficiencias adoptando una plataforma como Blackboard, la cual aunque ya probada y con cierto prestigio, no es gratuita, ni su adopción nos hace avanzar en el terreno tecnológico. En todo caso, tratando de evadir los costos se adopta Moodle, pero no se resuelve el problema fundamental de nuestra carencia de participación en el terreno de la creación y desarrollo de software, de proponernos implementar una plataforma de gestión y administración del conocimiento (con un sistema de evaluación automatizada) a nuestra medida, que responda a nuestras necesidades.

La tesis por lo anterior tiene cierto valor en cuanto es una propuesta para atacar el problema de la evaluación automatizada, vía los ejercicios de

autoevaluación y con el “*Generador Aleatorio de Exámenes Matemáticos*” (GAEM). Como resultado del trabajo de tesis al modelar los desarrollos expuestos en esta tesis, los prepara para su evolución posterior, su inserción en un sistema de evaluación robusto parte de un sistema de gestión y administración del conocimiento al que el IPN y el país tienen la obligación de generar.

1.6 Alcances de la tesis

Evaluar incluye los programas y currículos, la institución educativa, a los profesionales de la educación y de la formación, el software y hardware utilizado en las diferentes fases de la actividad educativa (administración, gestión del conocimiento, evaluación) y la propia evaluación. También se involucra a las políticas educativas, así como a las reformas e innovaciones del sistema educativo [Cappelletti, 2000]; en éste caso el ámbito e-learning (Figura 1.1):

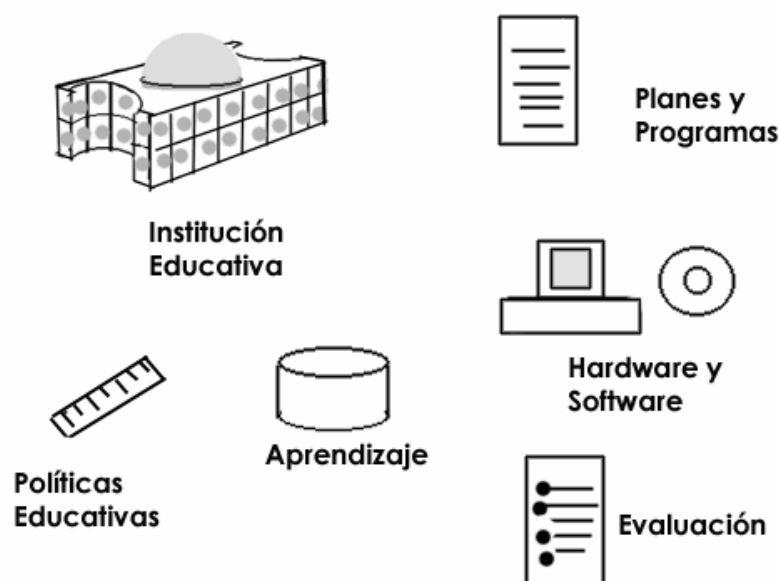


Figura 1.1 Algunos aspectos que deben ser evaluados

La evaluación no se limita al alumno, pues abarca a otros involucrados (actores) en el proceso educativo: profesores, tutores, expertos, autoridades, elementos curriculares, así como a centros de aprendizaje en la Figura 1.2, se muestra de forma simplificada algunos de los actores y casos de uso:

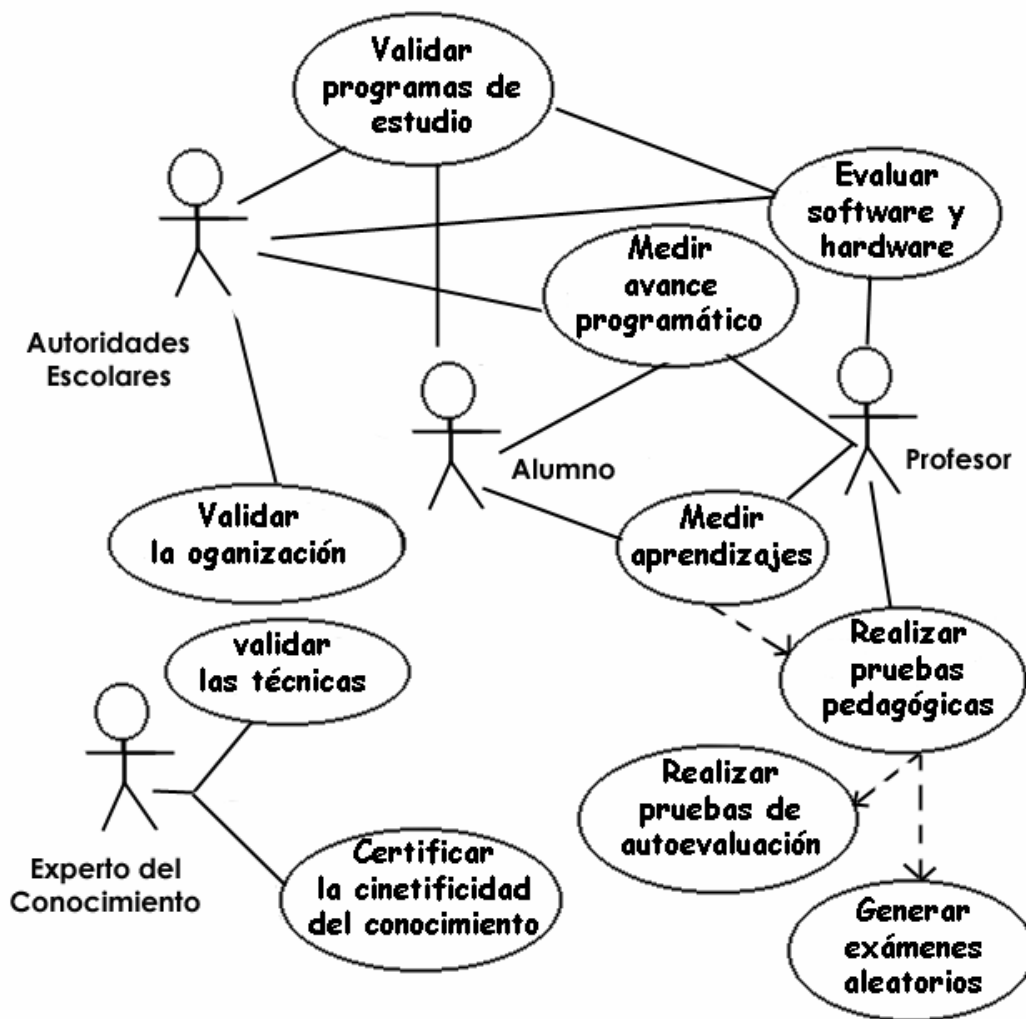


Figura 1.2 Algunos actores y casos de uso fundamentales en el proceso de evaluación

Sin embargo, la presente tesis **se ha enfocado al ámbito de los aprendizajes**, en el cual los actores fundamentales son el alumno y el profesor a su cargo. En este caso la principal herramienta la constituyen las pruebas pedagógicas, que en su versión para el Web (Online) son principalmente de opción múltiple y respuesta breve. Las páginas Web que despliegan los exámenes activan mediante eventos (generalmente con un "click" sobre un botón) los programas de corrección automática.

El trabajo de tesis sólo muestra el haber logrado desarrollar dos **propuestas** para exámenes de autoevaluación de reactivos de opción y respuesta múltiple, no desarrolla pruebas pedagógicas de correspondencia

o identificación. Esto se debe a que los programas aquí presentados respondieron a necesidades de un proceso educativo real, donde los reactivos fueron aportados por especialistas, que verificaron la correcta actividad de los programas y propusieron sólo reactivos de opción múltiple.

Las pruebas de correspondencia e identificación, así como las de respuesta breve se hallan fuera de los límites de esta tesis, como un desarrollo futuro.

En la tesis se describe y modela la actividad de un **tercer programa** el Generador Aleatorio de Exámenes Matemáticos (GAEM), que a partir de una base de datos de reactivos y de forma aleatoria despliega expresiones matemáticas, a partir de archivos gráficos. **El GAEM no es un editor de reactivos**, sino que a partir de una base de reactivos genera aleatoriamente los exámenes. GAEM realizó sólo esta actividad, pues no logra evaluar las respuestas a los exámenes propuestos, lo cual implicaría de lograrse automatizar, el desarrollo de la inteligencia artificial capaz de analizar el razonamiento seguido; por ejemplo, en el desarrollo de teoremas matemáticos, así como de algoritmos.

1.7 La metodología para el desarrollo del proyecto de tesis

Para sustentar la tesis, se realizó un estudio de plataformas internacionales, así como de experiencias nacionales de desarrollos de software para la gestión del aprendizaje y la evaluación, enfocándose en las características de las pruebas pedagógicas (si los reactivos son de respuesta, breve, opción múltiple o de otra naturaleza), poniendo énfasis en los casos donde se realiza evaluación automatizada y si existe alguna propuesta de pruebas de reactivos obtenidos de forma aleatoria.

Se ha hecho una investigación bibliográfica y documental, para lo cual ha sido de gran utilidad las consultas en Internet. Sin embargo, las entrevistas han ayudado mucho sobre todo en lo relativo a los sistemas nacionales.

En lo relativo al desarrollo de software, es un hecho probado que seguir una metodología orientada a objetos (OO), permite una mejor solución en términos de los requerimientos del software; que tendrá mejores posibilidades de evolucionar y reutilizarse. Sin embargo, aunque desde el punto de vista conceptual puede realizarse un análisis y diseño OO, no todos los lenguajes pueden realizar implementaciones al modo en que lo hace Java o C++. En estos lenguajes es posible definir clases con sus métodos y propiedades; en otros no es posible.

En JavaScript (el lenguaje que se utilizó para el desarrollo de los programas), aún cuando existen objetos, no pueden definirse clases; además, el uso de funciones no miembros que se activan por eventos como el “click” del mouse, obliga también a considerar algunos conceptos de la programación estructurada clásica que, sin embargo, pueden ser admitidos o expresados en términos de los diagramas del **Lenguaje de Modelado Unificado** herramienta del **Proceso Unificado**, originalmente impulsado por **Rational Software Corporation**, hasta que esta compañía fue comprada por IBM en 2003. Las siglas **RUP (Rational Unified Process)** identifican al conjunto de metodologías y software **que ahora como propietaria IBM promueve**; para evitar problemas legales algunos metodólogos hablan simplemente del Proceso Unificado (**UP**).

Aunque UML, permite modelar las diferentes fases del desarrollo de un producto de software, aquí, de lo que se trata esencialmente, es de: modelar y describir diferentes vistas de programas de cómputo expuestos en la tesis, con ello tener una mejor idea de su arquitectura, poder reutilizarlo y hacerlo evolucionar. No se trata de planear el desarrollo de un software, sino explicar los componentes y la actividad de los programas desarrollados por el sustentante de la tesis.

El UP pretende responder a las necesidades de la construcción de software de dimensiones industriales, complejo, potente y generado en el menor tiempo posible.

El UP surge ante la necesidad de coordinar (Figura 1.3), las múltiples cadenas de trabajo de un gran proyecto de software, que ante las múltiples fasetas del desarrollo se tenga un método común, un proceso genérico (que sirva a múltiples proyectos), que proporcione una guía para ordenar las actividades de un equipo, dirija las tareas de cada desarrollador por separado y del equipo como un todo, especifique los artefactos que deben desarrollarse y ofrezca criterios para el control y la medición de los productos y actividades del proyecto [Jacobson, 2000]:



Figura 1.3 UP coordina las múltiples cadenas de trabajo de un proyecto de software

El Proceso Unificado propone que los “casos de uso” (equivalentes con los requisitos funcionales) sean los que guíen el proceso de construcción de software, su arquitectura, la cual se describe por diferentes vistas del

software, las cuales contemplan los aspectos dinámicos y estáticos más significativos del sistema. Para **UP**, la arquitectura y los casos de uso evolucionan en forma paralela: La arquitectura debe permitir el desarrollo de los casos de uso requeridos, actuales y futuros. Esto solo es posible si la arquitectura se diseña para que el sistema evolucione (Figura 1.4).



Figura 1.4 La relación de la arquitectura de un sistema de software y los casos de uso según RUP

UP, permite realizar la modelación de diferentes vistas de la arquitectura del sistema de software, para ello cuenta con una herramienta muy importante que es el Lenguaje de Modelado Unificado (**UML**: Unified Modeling Language, Lenguaje de Modelado Unificado).

UML, sirve en esta tesis para modelar ciertas vistas y tener así una mejor idea de la arquitectura del software.

UP, ha sabido retomar aspectos utilizados en la época en que aún eran incipientes las metodologías de desarrollo no orientadas a objetos; es decir en la época clásica de la programación estructurada. Así el Lenguaje de Modelado Unificado (**UML** del Proceso Unificado), como herramienta es capaz de describir el proceso de codificación en programas donde hay funciones aisladas; es decir, que no son métodos de una clase.

Debe insistirse que la tesis no es la propuesta de un software futuro, sino que se dan a conocer tres programas básicos para que puedan ser reutilizados en un sistema “Online” robusto. En consecuencia, del Proceso o metodología UP, se utilizarán aquellos artefactos que permitan describir, modelar el funcionamiento de los programas y que permitan depurarlos y hacerlos evolucionar.

Considerando lo anterior los pasos seguidos para el desarrollo de la tesis [Galindo, 2007] y la obtención del producto, se describen en la Tabla 1.1:

Tabla 1.1 Los pasos de la investigación de la presente tesis

Objetivo ¿Qué obtener? (general)	Metodología ¿Qué hacer?	Técnicas ¿Cómo hacer?	Herramientas ¿Con qué hacer?	Metas ¿Qué obtener? (particular)
*Obtener marco de referencia sobre experiencias, propuestas y beneficios de Sistemas de Administración de Aprendizaje.	*Investigar el desarrollo de sistemas de evaluación y autoevaluación en línea	*Entrevistas. *Búsqueda de información bibliográfica y en Internet	*Conexión a Internet. Procesador de textos	*Información sobre ambientes de administración de aprendizaje "Online"
*Investigar propuestas actuales.	*Investigar características de BlackBoard, Moodle, Hot Potatoes, MathLab, productos derivados de Flash de Macromedia entre otros; esencialmente en lo que respecta a la evaluación. Investigar propuestas nacionales.	*Entrevistas. *Obtener información de los programas seleccionados y probar tutoriales o "demos" de los mismos.	*Conexión a Internet. *Procesador de textos.	*Identificar las características de sus propuestas de evaluación.
*Describir la arquitectura de los programas de autoevaluación "clásico" y "modificado" desarrollados por el sustentante de la tesis	*Abordar la clasificación estructural, el comportamiento dinámico y gestión del modelo de los programas de autoevaluación "clásico" y "modificado"	*Hacer una revisión del código de las páginas Web donde se utilizan los programas desarrollados así como de las funciones escritas en JavaScript que los componen	*Utilizar el Lenguaje de Modelado Unificado (UML). *Creación de imágenes en Photoshop. *Utilizar para crear diagramas editores de texto como Word	*La vista de los caos de uso de los programas propuestos. *La vista de diseño con diagramas de objetos, clases y de actividad de las funciones involucradas. *vista de gestión con diagramas de "paquetes".
*Describir la arquitectura del programa "Generador Aleatorio de Exámenes Matemáticos" desarrollado por el sustente de la tesis	*Abordar la clasificación estructural, el comportamiento dinámico y gestión del modelo del programa Generador de Exámenes Matemáticos.	*Hacer una revisión del código de las páginas Web, así como de las funciones escritas en JavaScript	*Utilizar el Lenguaje de Modelado Unificado (UML). *Creación de imágenes en Photoshop. *Utilizar para crear diagramas editores de texto como Word	*Diagramas de casos de uso de los programas propuestos. *La vista de diseño, física y de gestión.
*Redacción de la tesis	*Para fundamentar la tesis, la metodología para desarrollar un tema de tesis de maestría propuesta por Galindo [Galindo, 2006]	*Desarrollar un capitulo, describiendo su contenido temático.	*Word, *Photoshop.	*Documento de tesis de maestría

1.8 Organización de la tesis

El contenido del presente trabajo de tesis se ha desarrollado en siete capítulos, cuya temática fundamental es la siguiente:

Capítulo I. Presenta la justificación de la tesis, describe objetivos, beneficios esperados, alcances y límites, así como la metodología utilizada y las herramientas empleadas para el desarrollo de la tesis.

Capítulo II. Trata el contexto en que se desarrollan los programas de cómputo aquí expuestos, la discusión histórica permite aproximar al ámbito de conocimiento y de la actividad productiva, en la cual se insertan los programas que se han desarrollado y descritos en la presente tesis. Se exponen los conceptos fundamentales de la tesis, conforme a la metodología de Galindo [Galindo 2006].

Capítulo III. Describe la situación actual. Analiza varios casos de software internacional y de producción nacional, con herramientas de evaluación. Trata el problema de la evaluación. Expone los casos de programas internacionales (plataformas) involucrados en la educación por medios electrónicos ("e-learning") y más particularmente en la educación en línea (Online), así como las propuestas y experiencias nacionales en programas de evaluación y cómo se contempla ésta en la perspectiva de los intentos institucionales para el desarrollo de sistemas de gestión de objetos de aprendizaje. Se justifica la tesis y se plantean sus objetivos.

Capítulo IV. Se realiza una descripción de los lenguajes utilizados en los desarrollos propuestos para la presente tesis. Explica algunas particularidades del lenguaje de programación JavaScript utilizado en los programas de cómputo que dan lugar a ciertas dificultades en su modelación en términos del Lenguaje de Modelado Unificado (UML) y cómo se han resuelto. Se realiza la modelación de algunas vistas correspondientes a la clasificación estructural y el comportamiento dinámico de dos programas de autoevaluación propuestos: El examen de "**autoevaluación clásico**" y un examen de "**autoevaluación modificado**" para facilitar la retroalimentación de los alumnos.

Capítulo V. Realiza la descripción técnica, modelación estructural y del comportamiento dinámico, del programa denominado "**Generador Aleatorio de Exámenes Matemáticos**" mediante el Lenguaje de Modelado Unificado (UML). Dicho programa es otra propuesta del sustentante de la tesis a las necesidades de profesores de matemáticas y de ingeniería por contar con pruebas pedagógicas en línea con reactivos diferentes a los del tipo opción múltiple, de correspondencia o complementación.

Capítulo VI. Expone el comportamiento de los programas en diferentes plataformas sistemas operativos y navegadores, contrasta la propuesta con otras análogas de difundidas en Internet.

Capítulo VII. Presenta conclusiones y propuestas futuras de desarrollo del trabajo de tesis.

Al final del documento se presenta las **referencias bibliográficas** y de Internet, así como **anexos de los programas**.

Resumen del Capítulo

Los programas expuestos en la tesis son resultado de necesidades de la modalidad “virtual” de la Maestría en Ingeniería de Sistemas de la ESIME Zacatenco.

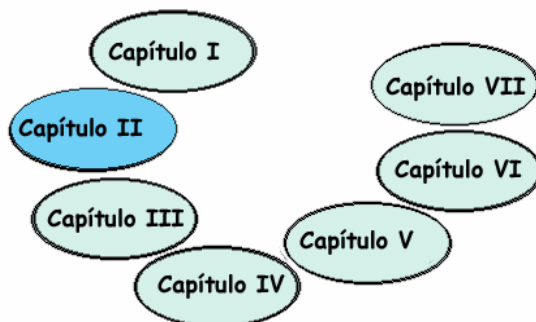
Entre los objetivos de la tesis se halla el modelar los tres programas propuestos llamados aquí de evaluación “clásico”, “modificado” y un Generador de Exámenes Matemáticos; para ello se ha utilizado el Lenguaje de Modelado Unificado (UML, Unified Modeling Language), describiendo las vistas del software, conforme al Proceso Unificado (UP).

Se han expuesto los pasos de la investigación: Objetivos, metodologías técnicas, herramientas, metas.

La tesis se ha dividido en siete capítulos que abarcan justificación, objetivos, metodologías, investigación contexto histórico de los desarrollos presentados, plataformas o programas actuales, descripción de la arquitectura de los tres programas elaborados, pruebas realizadas y conclusiones.

CAPÍTULO II.-

Marco contextual y conceptual



Contenido del capítulo:

Se discute el marco contextual histórico-laboral, en el que se presenta la necesidad desarrollar el software expuesto en la presente tesis, intentando ubicar en qué campo de las aplicaciones computacionales tiene utilidad los desarrollos aquí propuestos.

Trata los avances en educación a distancia, e-learning, las plataformas “Online” distinguiendo los desarrollos **CMC** (Content Management System, *Sistemas Administrador de Contenido*), **LMS** (Learnig Management System, *Sistema Administrador del Aprendizaje*), **LCMS** (Learning Content Management System, *Sistema Administrador de Contenidos y Aprendizaje*) y las experiencias nacionales de la educación a distancia y espacios virtuales.

Se expone el marco conceptual para cuya identificación se ha hecho uso de la metodología par el desarrollo de una tesis de maestría propuesta por Galindo [Galindo, 2006] y más específicamente de su pirámide conceptual. Conforme a dicha metodología, en éste capítulo se ha realizado una breve descripción de los términos empleados en el trabajo.

2.1 Marco contextual

2.1.1 El desarrollo histórico de la educación a distancia

Hasta hace algunos años en México, el recinto académico con sus centenares de alumnos se nos aparece como una “casa abierta”, a la cual todo aspirante puede incorporarse; sin embargo, siempre han existido limitaciones a su ingreso.

Remontándonos a épocas anteriores en México, durante el período colonial, la primera de una serie de restricciones fue la de no pertenecer a cierta casta u estamento: Los hijos de españoles y criollos asistían a colegios como el de San Pedro y San Pablo o el de San Ildefonso, en tanto que, los hijos de los “indios” nobles al colegio de San Gregorio. En la misma época la capacitación para el trabajo, prácticamente inexistente, se hallaba en el ámbito de los talleres artesanales los cuales mantenían el secreto de sus técnicas, bajo las normas de las cofradías.

La educación de “masas”, a la cual pueden asistir potencialmente todos los ciudadanos emana como conquista del movimiento liberal que arranca del gobierno de Valentín Gómez Farías y culmina con la Revolución Mexicana. Sin embargo, ello, no ha resuelto para gran parte de la población sus necesidades de formación profesional ni de capacitación para el trabajo.

Desde luego, la problemática arriba señalada no es privativa de México sino que es compartida mundialmente, lo cual ha dado lugar a diferentes propuestas no escolarizadas que incluyen los cursos por correspondencia (utilizando: impresos, casetes de cintas magnetofónicas, películas, etc.), programas radiofónicos, televisión educativa, etc.; todas ellas reconocidas hoy en día como modalidades de la **educación a distancia**.

La educación a distancia no fue una ocurrencia del siglo XX y mucho menos del XXI; basta recordar que en 1728, en Boston se publicó en el diario “La Gaceta” un material “autoinstruccional”, ofreciéndose tutorías por correo.[Web0, 2006].

En 1840, Isaac Pitman organiza en Inglaterra un intento rudimentario de educación por correspondencia al impartir cursos de estenografía [Web1, 2006]. En 1856, Gustav Langescheist en Alemania, utilizó el correo postal para enseñar idiomas. En el año de 1883 el abaratamiento del correo llevó a que el instituto Cahutaqua de Nueva York emitiera títulos a distancia [Web1a, 2006].

En el siglo XX, tras la segunda guerra mundial y posiblemente ante las dificultades materiales derivadas de la reconstrucción, no es extraño que

surgiera en 1947 *Radio Sorbonne* transmitiendo cursos magistrales en casi todas las materias de la Facultad de Letras y Ciencias Humanas; para 1962, en España, hay una experiencia de Bachillerato Radiofónico y en 1969 nace la *Open University* de Inglaterra, el “arquetipo” de la universidad virtual.

La Open University, fue creada en función de la creencia de que la tecnología de la información y comunicación podía promover el aprendizaje de alto nivel de calidad a personas que no hubieran tenido la oportunidad de asistir a universidades convencionales y dar de este modo una solución al problema de la exclusión; muchas instituciones a partir de los años 70, siguieron el modelo de la Open University con éxito.

Así se ha ido desarrollando la llamada educación a distancia que suele requerir en algunos casos sincronía del transmisor y receptor, con la presencia del alumno al modo del sistema tradicional formal escolarizado, pero también con la posibilidad de que el alumno asista sólo un tiempo mínimo por semana. Se trata de las modalidades "presenciales" y "semipresenciales" de la educación a distancia; en el primer caso un ejemplo típico es la telesecundaria, el segundo caso ha sido utilizado en muchos cursos del llamado Campus Virtual Politécnico.

2.1.2 E-learning

La educación a distancia tomaría nuevas vertientes con la aparición de la computadora; la introducción de ésta en la educación fue al principio como auxiliar en el cálculo numérico y en la graficación. Al intentar aplicarse a la enseñanza de niños bajo una visión más pedagógica, lleva al desarrollo del software de autoría (authoring o de creación), que facilitarán a su vez la elaboración de programas de cómputo multimedia por los profesores; pero también se avanza en la aplicación de la computadora *al proceso de evaluación*.

Al principio mientras no se habían desarrollado las redes informáticas, se utilizaron lectores ópticos para capturar los resultados de exámenes impresos repartidos a los alumnos, quienes tenían que llenar los “alveolos” de respuestas (exámenes de opción múltiple).

Paralelo al uso de la computadora se extendió la digitalización de medios como audio, video, imágenes; la extensión del concepto de hipertexto a estos recursos permitió el paso de la multimedia a la hipermedia, enriqueció las páginas electrónicas y gestó el Web.

La entrega de contenido vía los medios electrónicos, incluyendo Internet, intranets, extranets, retransmisiones vía satélite, casetes de audio/video,

televisión interactiva y CD-ROM, queda comprendida con el término “*e-learning*”.

Con base a lo anterior, la propuesta de avanzar por la vía del “e-learning” ha sido tomada en serio por muchas entidades educativas y productivas. Según Eva I. Fernández [Fernández, 2004] “e-learning”, es un sistema de teleinformación que aprovecha las actuales estructuras de Internet e Intranet, convirtiendo parte de éstas en un medio que permita la impartición de acciones formativas no presenciales, evidentemente sin la necesidad de que las partes implicadas coincidan en espacio tiempo, proporcionando un abanico de soluciones que aúnan la adquisición de conocimiento, habilidades y capacidades.

El término “e-learning” cubre un extenso conjunto de aplicaciones y procesos incluyendo la “Educación Asistida por Computadora” (CAT Computer Assisted Training), “Aprendizaje Basado en la Computadora” (CBT “Computer Based Training), “Aprendizaje Asistido por Web” (WAT Web Assisted Training), “Aprendizaje Basado en el Web” (WBT Web-Based Training), que representan la ayuda parcial o total de la computadora, así como la ayuda total o parcial del Web. El WAT y el WBT son abarcados por el concepto de educación en línea (Online).[Web1ab, 2006]

Las aplicaciones y procesos enmarcados en el WBT, corresponden a la educación fundamentada, expresamente desarrollada para el entorno del Web. Particularmente, en este entorno, se ubican los programas desarrollados y descritos en la presente tesis.

La Figura 2.1, esquematiza la amplitud de la educación a distancia:

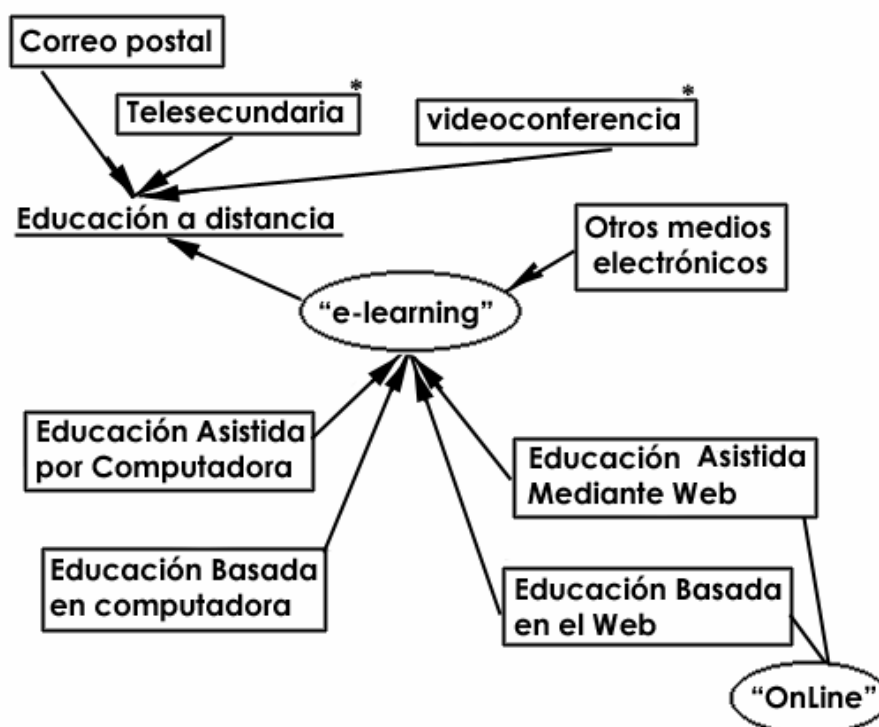


Figura 2.1 Modalidades de la educación a distancia. Se muestra a "e-learning" como un concepto atribuido a la información digital, incluyendo la "Online". Con asterisco (*) se indican las variantes "presencial" y/o "semipresencial".

En esta tesis, sostenemos una concepción diferente a la del M. en C. Rubén Peredo y el Dr. Leonid Sherematov, investigadores del Centro de Investigación en Computación (CIC) del IPN e impulsores de la plataforma EVA (Espacios Virtuales de Aprendizaje), quienes al menos en el año 2002, identifican "e-learning" con "online", excluyendo a la enseñanza mediante CD-ROM [Peredo, 2002].

El "Online-learning", constituye sólo una parte de la formación tecnológica y describe el aprendizaje vía Internet, intranet y extranet. El aprendizaje "Online" en su forma básica incluye el texto, gráficos del curso, ejercicios de evaluación, el control periódico de los resultados del aprendizaje. En su forma más avanzada incluye audio, video, grupos de discusión con expertos y compañeros, acceso a los materiales del curso (objetos de aprendizaje), intercambio de información, uso del correo electrónico.

La diferencia entre "e-learning" y "online" ha sido ampliamente discutido por Marcos Navarro [Weblab, 2006], y por los autores de Weblearning Resources [Weblearning, 2006], del primer artículo retomamos la Figura 2.2.

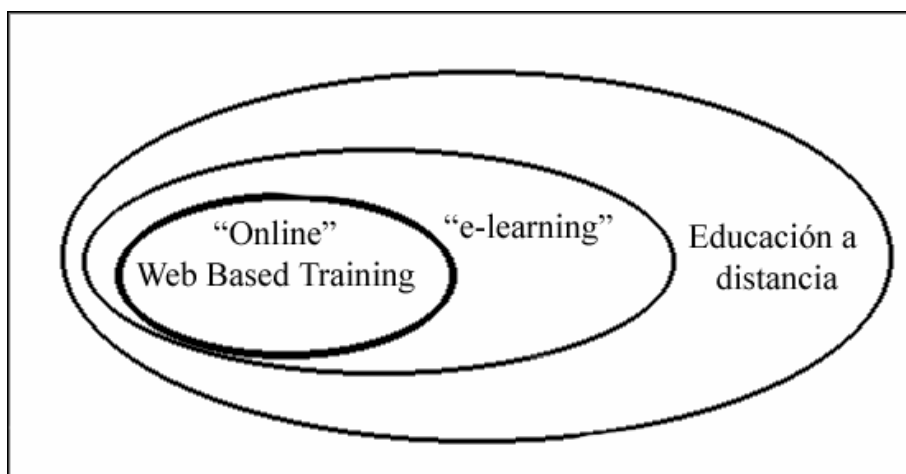


Figura 2.2 La educación "Online" parte de la educación por medios electrónicos "e-learning"

"E-learning" tiene un ingrediente relevante en la **evaluación**, que de acuerdo a si su alcance u automatización es parcial o completa puede describirse como "Evaluación Asistida por Computadora" (Computer Assisted Assessment, CAA), "Evaluación Basada en la Computadora" (Computer-Based Assessment, CBA). Su vertiente "Online" es la "Evaluación Asistida mediante Web" (Web-Assisted Assessment, WAA) y la "Evaluación Basada en Web" (Web-Based Assessment, WBA), según la mayor o menor automatización en Web [Web1ac]. La Figura 2.3, muestra las modalidades "e-learning" de la evaluación:

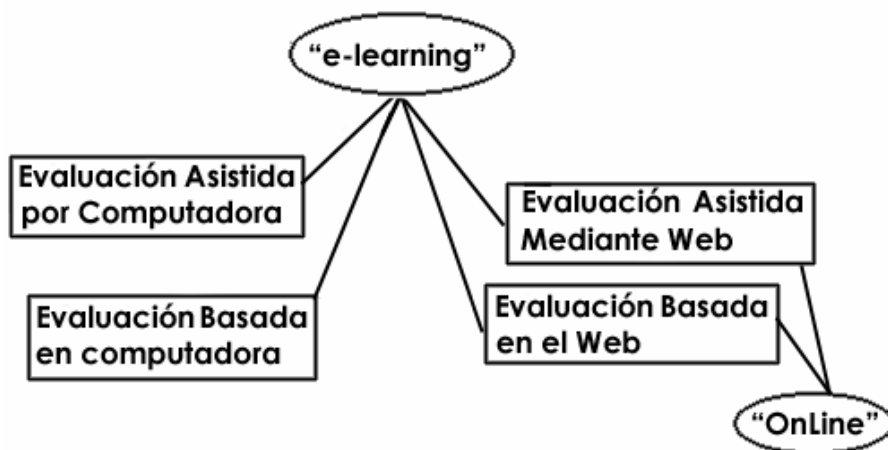


Figura 2. 3 La evaluación "e-learning" y su modalidad "Online"

La proliferación del uso de la computadora y de las redes de computadoras, así como la aparición del World Wide Web proporcionó nuevas herramientas a los docentes y nuevas alternativas a los demandantes de formación profesional, así como capacitación para el trabajo, al crearse la posibilidad de aprender bajo un modelo asíncrono.

Las potencialidades de la tecnología computacional en la educación han despertado interés para las empresas (educativas o no) demandantes de servicios educativos, ante la posibilidad de aminorar el alto costo que implica la capacitación por el desplazamiento de sus empleados, ausencia durante el proceso de formación o capacitación, limitaciones de espacio, etc.; en tanto que para las instituciones o empresas que pretenden satisfacer la demanda de educación, la falta de recursos para mantener sus instalaciones y renovarlas.

Para la década de los 90 del siglo XX, con la difusión del Web surgen plataformas que permiten la creación y gestión de cursos completos para la Web sin que sean necesarios profundos conocimientos de programación y diseño gráfico. El número creciente de estas plataformas muestra que el mercado educativo "a distancia", comienza a interesar a muchas instituciones.

En los EUA, a mediados de la década de los 80 había 300 mil estudiantes matriculados en tanto que a mediados de los 90 llegaban a 700 mil, previendo ya un nivel de más de 2 millones en la primera década del siglo XXI. [Web1, 2006]. Las compañías que han emergido, muchas de ellas con base en una universidad, han seguido incluso un proceso análogo a la de los grupos industriales y financieros, comenzando con alianzas y terminando con fusiones, un ejemplo de ello es la fusión (merge) de Blackboard y WebCT [Web1ad, 2006].

Las plataformas consisten en software para servidor de Internet/Intranet cuyas características fundamentales son:

- i) Gestionar el acceso de los usuarios, controlar su aprendizaje, historial y emisión de informes.
- ii) Gestionar los cursos, los objetos de aprendizaje, realizar la evaluación del aprendizaje.
- iii) Gestionar los foros de discusión, chats, videoconferencia.

Dependiendo de sus capacidades podemos hallar tres tipos de plataformas o software e-learning:

- a) **CMS** (Content Management System, Sistema Administrador de Contenido), es el tipo el más sencillo de las plataformas; **no posee herramientas elaboradas de colaboración** (foros, chat). No es posible

gestionar pre-requisitos, planes de carrera; contribuye a publicar contenido, el cual se halla en un repositorio y apoyar el acceso a él.[Web1c, 2006], [Web1ab, 2006]. Las siglas CMS indican también un Sistema Administrador de Cursos (Course Management Systems) para los cuales no es muy relevante someterse a estándares para intercambiar, sino que se mantienen como propietarios de sus módulos [Weblearning, 2006].

b) **LMS** (Learning Management System, Sistema Administrador del Aprendizaje). Representa un nivel superior de complejidad; permite comunicar y transferir información, evaluar al alumno y al profesor. Normalmente, se basan en el Web para que se tenga acceso a contenidos de aprendizaje y su administración [Web1d, 2006], [Web1ab, 2006].

c) **LCMS** (Learning Content Management System, Sistema Administrador de Contenidos y Aprendizaje). Es el sistema más robusto gestiona los contenidos de aprendizaje; reutiliza contenidos, los cuales se almacenan en una base de datos centralizada; cuenta con herramientas de colaboración, evaluación y análisis, búsqueda y recuperación de contenidos, así como de su creación [Web1e, 2006], [Web1ab, 2006].

Las tareas de los LCMS, se realizan utilizando etiquetas XML y siguen ciertos estándares como **AICC** (Aviation Industry Computer Based Training Committee, *Comité de Entrenamiento Basado en Computadora de la Industria de Aviación*) y **SCORM** (Sharable Content Object Reference Model, *Modelo de Referencia de Objetos Distribuibles*), con lo cual pueden publicar materiales en diversos formatos y plataformas e incluso dispositivos inalámbricos; cuentan con herramientas para la creación de contenidos y la evaluación automatizada.

Las diferencias más relevantes entre **LMS** y **LCMS**, es que en estos últimos destaca la organización de contenido reutilizable, herramientas para la creación (autoría) de contenido; por ejemplo, para desarrollo de reactivos y su corrección (evaluación).

De acuerdo a Navarro [Web1ab, 2006], la principal funcionalidad de un **LMS** es administrar estudiantes y dar seguimiento a su aprendizaje, participación y desempeño asociados con todo tipo de actividades de capacitación. Por otro lado, un **LCMS** administra contenidos u objetos de conocimiento el cual busca ofrecerse a la persona indicada en el momento indicado.

La Figura 2.4 describe de forma sintética, las principales características de las plataformas CMS, LMS y LCMS:

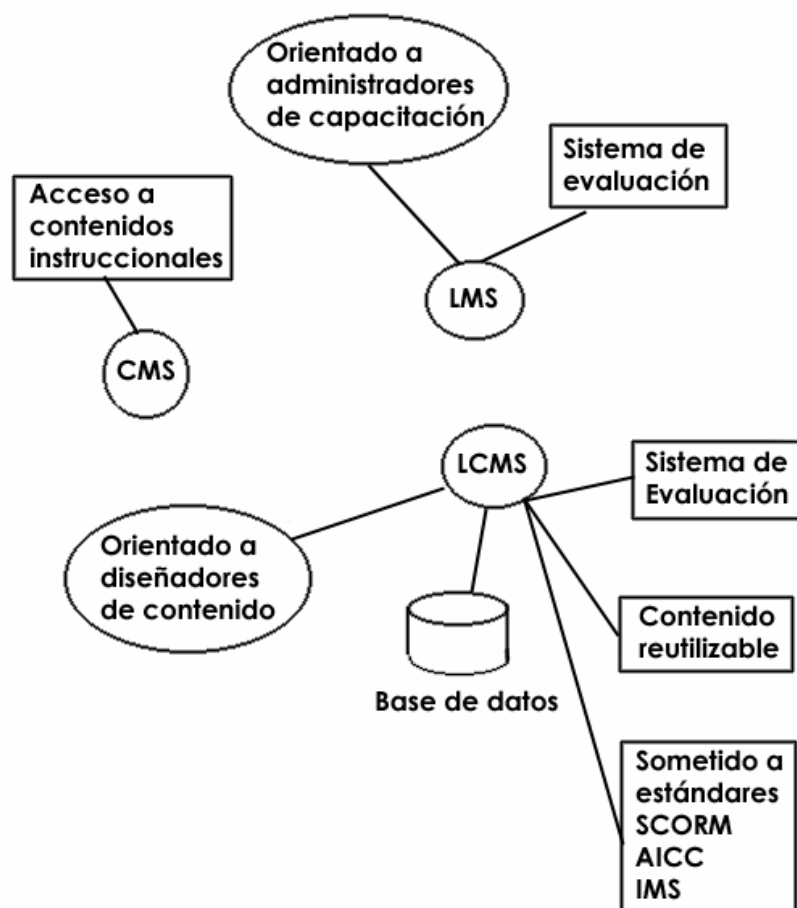


Figura 2. 4 Modalidades de plataformas para la gestión de la enseñanza en línea.

2.1.3 Educación a distancia y "e-learning" en México

México, tiene una relevante experiencia en educación a distancia en términos de la televisión educativa: La primera transmisión se da a propósito de la VII Asamblea de Cirujanos, en el quirófano principal del antiguo Hospital Juárez en 1948.

En 1968, surge la telesecundaria en México bajo la responsabilidad de la Unidad de Educación Audiovisual, hoy en día Unidad de Telesecundaria.

[Web1b, 2006]. Para el tiempo en que los satélites “Solidaridad I” y “Solidaridad II” son puestos en órbita, los **once mil planteles** de telesecundaria son dotados de antenas parabólicas, con modernos aparatos para recibir la señal codificada, digitalizada y comprimidas a través de la red EDUSAT [Web2, 2006], misma que ha permitido ofrecer la telesecundaria a toda Centroamérica. [Web3, 2006].

Cabe mencionar que se ha explorado en México con la educación a distancia en cursos para nivel de primaria, educación para adultos (INEA), secundaria y hasta posgrado. Fácilmente se puede hallar en la experiencia de la televisión, los antecedentes de lo que hoy conocemos como videoconferencia base del trabajo en diversos “campus virtuales” como el existente en el IPN.

El arribo de la Tecnología de Información y Comunicación en escala ampliada al Instituto Politécnico Nacional (IPN) de México, obliga a difundir ésta entre los maestros para su uso en el aula, a la vez que a preparar a los alumnos en profesiones dedicadas al desarrollo del hardware y software. A finales de 1995, arranca una serie de cursos por videoconferencia para la producción de software multimedia en plataformas Macintosh que concluye en 1996, con el propósito de impulsar “laboratorios de multimedia”.

La creación del Campus Virtual Politécnico llegó sin embargo con cierto retraso respecto a otras entidades educativas del país, pues fueron instituciones privadas las que primero se abocaron adentrarse en los asuntos de la “educación virtual” o “e-learning”: La Universidad Anáhuac en 1995, fue la primera en crear un espacio “e-learning” bajo el nombre “Universidad Virtual Anáhuac” (UVA) [Garrido, 2004].

En 1996, el Instituto Tecnológico de Estudios Superiores de Monterrey crea la “Universidad Virtual ITESM”, cuyos antecedentes datan de 1989 con un “Sistema de Universidad Virtual” para la transmisión de videos. Este adelanto responde a la necesidad de mantenerse en sintonía con las universidades del sur de los EUA, en particular en términos de su recertificación [Garrido, 2004].

De lado de las instituciones públicas la Universidad Tecnológica de la Mixteca con la finalidad de hacer más accesible la educación de calidad, sin importar su localización geográfica y aprovechando el potencial de ventajas de las Tecnologías de Información y Comunicación (TIC) surge en 1997, la universidad virtual “Virtual UT”. En 1998 inicia en la modalidad “virtual”, la Maestría en Computación, Especialidad en Sistemas Distribuidos [Web3b, 2006].

A partir del año 2000 se da la explosión de las universidades virtuales a lo largo de la República: En la Universidad Regiomontana (2000), Universidad Autónoma de Tamaulipas (2001), Universidad Juárez de Tabasco (2001). En la UNAM (2001), con el centro de Alta Tecnología de Educación a Distancia [Garrido, 2004].

Una mención especial debe hacerse de la Universidad de Colima, pues aunque en 2003, crea el Programa de Educación a Distancia, se sabe que ha tenido una participación relevante desde 1983, en términos de "e-learning" con la automatización de bibliotecas, creación de Centros Editor de Discos Compactos (CENEDIC), dando gran impulso a la multimedia por computadora [Garrido, 2004].

El IPN, en el año 2000, crea el "Campus Virtual Politécnico", aunque hay un antecedente en 1995 del programa de Educación Continua y a Distancia, tras propuestas no muy consolidadas, el Instituto establece en 2003 los llamados "Ambientes Virtuales de Aprendizaje" (AVA).

Tal vez retomando la idea de AVA del IPN, la Universidad de Guadalajara crea su portal de innovación educativa: "Ambiente Virtual de Aprendizaje", la ventaja de ésta universidad es de que desde 1998 viene utilizando las plataformas WebCT y Blackboard. Esta última introducida también en el IPN.

Eva Fernández [Fernández, 2004], sostiene que los conceptos de "campus virtual" o "universidad virtual" deben utilizarse cuando se hace referencia a espacios de comunicación interpersonal o a través de grupos entre la institución universitaria y el alumnado que ofrezcan servicio a toda la comunidad educativa; es decir, el campus virtual sirve a los alumnos porque le aporta medios para el aprendizaje en tanto que al profesor y administradores, medios para la gestión.

La "Universidad Virtual", se sustenta en la independencia tiempo-espacio del alumno respecto a un horario rígido y un espacio de trabajo definido. El alumno no requiere estar presente en un cierto recinto, verse obligado a tener en frente a un maestro o grupo de maestros, en un horario o instalaciones predefinidas. El alumno puede ser antípoda de su "maestro" y sin embargo poder participar en un proceso de enseñanza- aprendizaje.

Tanto en la educación semi-presencial a distancia, como en la educación estrictamente virtual (no presencial), los responsables de facilitar el conocimiento, hacen uso de tecnología de la educación, sistemas computarizados para la manipulación de audio, video, animaciones, etc. Sin embargo, la llamada "Universidad Virtual" ha tenido que poner mayor cuidado en la preparación de los medios para la instrucción, desde los mecanismos de comunicación, medios didácticos y sistemas de evaluación.

La universidad virtual, no puede cerrarse a ninguna alternativa que facilite el conocimiento y permita que el alumno lo haga suyo, pero es claro ha sido ésta vertiente de la educación la que ha sabido aprovechar el World Wide Web y las redes de computadoras entre las que destaca Internet. En muchos casos por “Educación Virtual” se entiende una educación “Online”

Nos encontramos ya en el mundo compañías que incursionan desde hace algunos años por el terreno de la educación semipresencial a distancia y en lo que es un incipiente ámbito de una educación virtual futura, en donde paradójicamente nos entregan como lo más avanzado sistemas de evaluación automatizada con reactivos de falso/verdadero y selección múltiple.

Tales sistemas causan, por lo que hemos visto gran admiración entre algunos de sus promotores dentro de las instituciones donde los pretenden insertar; asumiendo la incompetencia de sus propios profesionales para desarrollarlos, disponen de cuantiosas sumas en sistemas de gestión educativa desarrollados por compañías extranjeras.

Así bajo tal supuesto no es extraño que instituciones como el IPN con gran potencial de desarrollo de software no haya dado una salida concreta para no digamos superar los sistemas de evaluación, sino al menos tenerlos.

La razón de esta tesis va en la dirección de aportar al país y al IPN en particular, propuestas para cubrir huecos en nuestra participación mundial en el mercado mundial de la educación por medios electrónicos (e-learning) y en particular en la educación por Internet (Online).

Con los antecedentes expuestos hasta el momento, ahora podemos presentar el:

2.2 Marco conceptual

En el desarrollo del trabajo de tesis se sigue la metodología propuesta por Galindo [Galindo, 2006], quien sugiere después de detectar un problema en un medio ambiente de trabajo, crear una pirámide de conceptos relativos al problema, en cuya base se hallan los más generales y de ahí hacia la cúspide los de más particularidad.

Galindo, le da gran importancia a la “pirámide conceptual” pues considera que permite delimitar el ámbito de estudio, definir claramente los conceptos que empleará en la tesis. Describir o definir los conceptos involucrados, no sólo ayuda al defensor de la tesis, sino también al lector de ella.

Debe realizarse un análisis de la situación actual, semejante a un “Estado del Arte” y justificación de la propuesta de solución del problema y definir el objetivos general y particulares.

Se busca de acuerdo a las ideas de Galindo [Galindo, 2002], tener claros los conceptos en torno a la tesis, comenzando por el título de la misma, el sustentante debe saber de qué está hablando, para acotar el mundo a definir los conceptos que se emplearán, a definir su producto principal y y los objetivos.

2.2.1 Pirámide conceptual

Siguiendo a Galindo [Galindo, 2006], se define la pirámide conceptual.

La base de la pirámide tendrá el concepto más general y de ahí hacia arriba, se va particularizando. La cúspide de la pirámide será entonces: *el principal producto a obtener* (sistema, modelo, herramienta, etc.), y arriba de ella; el *objetivo del proyecto de tesis* (Figura 2. 5).

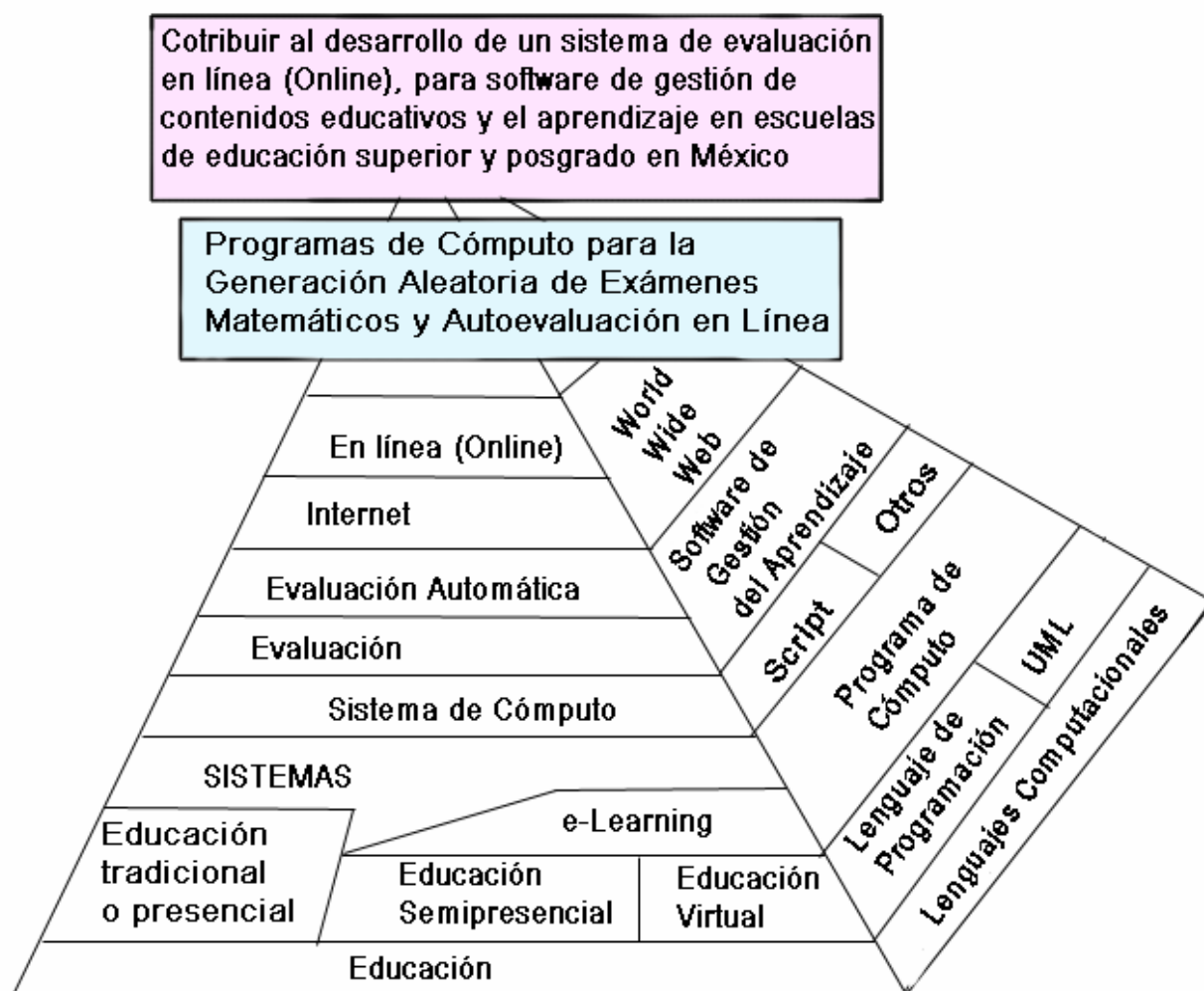


Figura 2.5 Estructura de la Pirámide Conceptual [Fuente propia]

Considerando esta estructura, a continuación, se desglosan los principales términos que se han de utilizar en el texto de la presente tesis:

2.2.2 Breve descripción de términos

A continuación, se realizará una breve descripción de los términos principales mencionados **en la pirámide conceptual**, independientemente de las definiciones del glosario:

Educación	La preparación para la vida la cual se da en el hogar y fuera de él. Se considera también un proceso para transmitir el bagaje cultural de una comunidad o grupo social a fin de perpetuar su propia existencia y su continuo desarrollo. Dicha actividad se realiza principalmente en los centros educativos y actualmente por medios electrónicos. Nos interesa en particular la educación impartida por el Estado. [La Enciclopedia, 2004]
Educación presencial	Ha sido llamada así la educación formal escolarizada; es decir aquella en la cual los cursos teórico-prácticos, así como los aspectos administrativos se realizan en recintos predefinidos para su uso por el alumnado, existe una sincronía en tiempo espacio en la relación alumno-maestro [Fuente propia].
Educación semipresencial y a distancia	Se ha denominado así a los cursos en los cuales se ha utilizado video conferencia y los alumnos se presentan en sedes de recepción de la transmisión. Existe aún sincronía tiempo-espacio. Puede contar con “facilitadores” del conocimiento cuyas funciones aunque aún análogas a las del maestro convencional, realiza tareas más relacionadas con la preparación de los recursos y medios didácticos; por ejemplo el desarrollo de software hipermedial [Fuente propia].
Educación virtual	Sistema de enseñanza aprendizaje en el cual existe una asincronía tiempo espacio entre el alumno y el maestro, muchas veces llamado “facilitador”, cuya intervención principal ahora es involucrarse en el desarrollo de los medios de facilitar el conocimiento. Aquí el trabajo del antiguo maestro cambia al participar en la generación, propuesta y desarrollo de los recursos multimedia; por ejemplo, de software hipermedial de características superiores, que permitan al alumno el autoaprendizaje asistido con los recursos más avanzados de las Tecnologías de Información y Comunicación (TIC). [Fuente propia].

E-learning	Educación que utiliza todos los recursos posibles de digitalización, incluye a la <i>educación en línea</i> (OnLine), la <i>“educación mediada por computadora”</i> , y los <i>“ambientes virtuales de aprendizaje”</i> basados en el Web. En ocasiones se le relaciona con sistemas informáticos de administración de aprendizaje. [Web1ab, 2006]
Evaluación	Acción y efecto de evaluar. Durante mucho tiempo se concibió exclusivamente como valoración de los conocimientos aptitudes, capacidades y rendimientos de los alumnos. Pero en la actualidad no sólo se reduce a aprovechamiento del alumno, sino también al programa de estudios, los textos, el desempeño del profesor. La evaluación puede tener muchos objetivos además del rendimiento escolar. [Cappelletti, 2004]
Internet	Acrónimo de International Network of Computers, red informática que conecta a escala mundial miles de redes regionales y varios millones de computadoras. Entre los servicios que presenta se puede enumerar el correo electrónico, acceso a boletines electrónicos, bases de datos, acceso a archivos y páginas Web. Las comunicaciones entre las computadoras se realizan por línea telefónica (fibra óptica, satélites artificiales y antenas parabólicas. Cada computadora se identifica con una dirección numérica; para facilitar el trabajo se utilizan caracteres alfanuméricos, pero las direcciones de esta forma construidas son posteriormente decodificadas por el sistema y transmitida de forma numérica. [La enciclopedia, 2004]
Lenguaje	Un lenguaje es un conjunto de términos formados a partir de un conjunto finito de símbolos. Las secuencias de símbolos tienen longitud finita, pero no existen limitaciones en cuanto a su longitud. Sea Σ_s el conjunto de símbolos y $\varnothing = \{ \}$ el conjunto vacío, $\Sigma = \Sigma_s \cup \varnothing$, donde Σ es el alfabeto, cuando se usa el lenguaje natural las secuencias de símbolos se conocen como palabras, pero a menudo (para el caso general), se les denomina cadenas. La cadena vacía ε es una palabra en cualquier alfabeto y es una secuencia vacía de símbolos. En particular el lenguaje compuesto por ninguna cadena es el lenguaje vacío ϕ que se distingue del formado por la cadena vacía $\{\varepsilon\}$ [Kelley, 1995].

Lenguaje de Modelado Unificado	El lenguaje de Modelado (UML , Unified Modeling Language) es un lenguaje gráfico para visualizar, especificar, construir y documentar los artefactos de un sistema con gran cantidad de software. UML proporciona una forma estándar de escribir los planos de un sistema, cubriendo tanto las cosas conceptuales, tales como procesos del negocio y funciones del sistema, como las cosas concretas, tales como las clases escritas en un lenguaje de programación específico, esquemas de bases de datos y componentes de software reutilizables. [Booch, et al. 1999]
Lenguaje de programación	Los lenguajes de programación se estructuran en reglas léxicas, sintácticas y semánticas. En otras palabras, los términos y símbolos del lenguaje se hallan definidas, no son arbitrarios, permiten generar instrucciones siguiendo ciertas reglas de escritura y tienen significados definidos. [Informática, 1986]
Programa de cómputo	Conjunto de instrucciones perfectamente legible por la computadora y destinado a realizar un determinado trabajo o solucionar un problema. [Informática, 1986]
Script	Son programas pequeños en archivos de texto que se interpretan a medida que se necesitan [Castro, 2001] Un lenguaje utilizado para crear scripts es JavaScript.
Sistema	Para van Gigch [Van Gigch, 2000] Sistema es un conjunto de elementos relacionados, en el que sus elementos pueden ser “conceptos”, “objetos” y “sujetos”. Para los estudiosos de la termodinámica será una porción de materia aislada del resto del universo por límites reales o imaginarios, a dicho sistema lo caracterizan ciertas variables, el estado de un sistema; su comportamiento lo determinan los valores de esas variables.
Sistema de cómputo	Realiza cuatro actividades básicas: 1) Entrada 2) Almacenamiento 3) Procesamiento y 4) Salida de datos y/o información.
Universidad virtual	Es un espacio de comunicación interpersonal, en el cual no sólo se pone a disposición de usuarios objetos de aprendizaje correspondientes a un programa, sino también servicios administrativos, de biblioteca y de comercio, principalmente. El antecedente se halla en la educación a distancia “clásica” apoyada en la televisión y en los satélites. Actualmente su soporte fundamental es Internet. Una característica es que los maestros evalúan a los estudiantes de manera remota, fuera del salón de clases y se comunican mediante correos electrónicos. [Contreras, 1997]

World Wide Web (red de alcance mundial) Constituye un conjunto de archivos de datos, llamadas páginas Web, que contienen imágenes fijas, video, texto. Los programas de cómputo que se utilizan para desplegar las páginas Web se llaman navegadores [La enciclopedia, 2004]

Resumen del capítulo

Para el desarrollo del texto de la tesis se ha utilizado la metodología de Galindo [Galindo, 2006] la cual sugiere la construcción de una pirámide de conceptos en cuya base se hallan los conceptos más generales y al ascender los más particulares. En la cúspide se halla el producto y objetivos de la tesis.

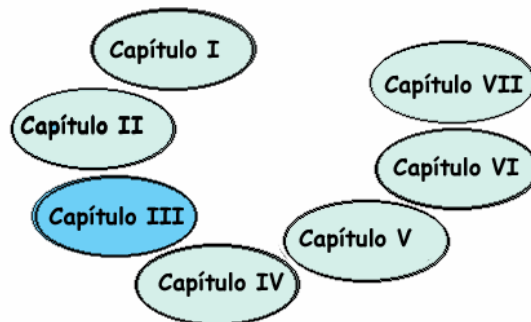
Se ha expuesto en el marco contextual la existencia de sistemas de software o plataformas para el “e-learning” y en particular en la llamada educación “Online”, tanto internacionales como nacionales, particularizando en el IPN, cuya experiencia aún no se consolida y en consecuencia la contribución de esta tesis va en el sentido de participar en la consolidación del “Campus Virtual Politécnico”.

Para la mejor comprensión de la tesis se ha expuesto un glosario en donde se describen los términos fundamentales utilizados en la tesis de acuerdo a la metodología de Galindo.

Expuestos los conceptos básicos para el desarrollo de la tesis, se procederá al análisis de la situación actual.

CAPÍTULO III.-

Análisis de la Situación Actual



Contenido del Capítulo

Trata el concepto de evaluación, de las pruebas pedagógicas y los reactivos que las componen y cuáles de éstos se adecuan mejor al contexto de la educación o formación en línea (OnLine).

Se describen los casos de software internacional desarrollado para la gestión del conocimiento (entornos para la administración del aprendizaje) tal como BlackBoard, Moodle, Claroline, EVA, SOFÍA y sus propuestas de evaluación. También se consideran e productos más orientados a la evaluación como Hot Potatoes o que tienden a ampliar su ámbito inicial para incorporar herramientas de evaluación tal como: Maple y autoría de reactivos como el programa KnowledgePresenter®2006 de Flash.

En cuanto a la experiencia nacional, se abordan los casos de EVA, SOFÍA, la experiencia de COSNET en la evaluación automatizada, así como de la maestría en Ingeniería de Sistemas de la ESIME Zacatenco.

3.1 La evaluación en la educación

En la década del 70 del siglo XX, en el medio educativo se abrió una etapa de interés por dar al proceso enseñanza aprendizaje un sentido más científico, profundizándose en el estudio de la didáctica, con ello de los modelos educativos, los medios y recursos didácticos, así como de la evaluación de los aprendizajes.

En México hay que ser claros, la orientación científica del trabajo educativo fue impulsado por las autoridades, “de arriba a abajo”; pero de una u otra forma pegó en las bases docentes. Las décadas del 70 y 80 del siglo XX representan la era de organizar el trabajo en torno a objetivos. La “objetivitis”, se trasladó del mundo académico al resto de la sociedad interesada en la planificación, en racionalizar procesos.

De tal etapa surge la idea de medir el aprendizaje, considerando si se había logrado alcanzar cierto objetivo y además logrado una modificación en la conducta. Se discutió en torno a la estructura de las pruebas pedagógicas de si los reactivos debían o no corresponder a una cierta categoría de Bloom (conocimiento, comprensión, aplicación, análisis, síntesis y evaluación. Figura 3.1) y por tanto, si tenían en un momento dado más peso que otro [Suárez, 2002].

Los profesores en algunos casos descubrieron y en otros lograron aplicar de manera más consciente, diversas modalidades de exámenes *orales, de desarrollo, preguntas cortas y de selección*, comprendiendo sus limitaciones.

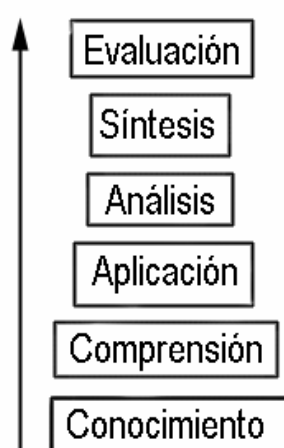


Figura 3.1 Categorías de la taxonomía de Bloom y su nivel de complejidad ascendente

En los años posteriores, llegaron los críticos al conductismo (a veces en forma más bien no crítica) y plantearon otros conceptos u enfoques en el proceso enseñanza aprendizaje y entonces surgió la idea de la educación basada en competencias, ya no se trataba de que si se había modificado la conducta, sino si el alumno había alcanzado o no cierta competencia.

Como fuere, de cada etapa ha quedado algo en términos prácticos y una de ellas es que la evaluación era algo más que un examen. El profesor (instructor o facilitador del conocimiento), al final de un curso tiene que asignar una calificación, pero que ésta tiene que ser resultado de una serie de consideraciones. La evaluación es un proceso continuo, se basa en el conjunto de actividades casos prácticos, ejercicios, trabajos y debates que se llevan a efecto en los diferentes módulos o etapas de un curso. Permite hacer seguimiento progresivo del aprendizaje alcanzado.

La evaluación educativa consiste en utilizar una serie de procedimientos destinados a comprobar que se han conseguido o no las metas y objetivos propuestos, identificar los factores o razones que han influido en los resultados (éxito o fracaso) y formular las recomendaciones particulares que permiten tomar decisiones con el fin de introducir correcciones o los reajustes que sean necesarios. Evaluación es medir, calificar, estimar y acreditar [López, 2004].

Desde el punto de vista educativo la medición se plantea como un proceso mediante el cual se establece la cantidad de características que un escolar posee, éstas pueden ser de eficiencia del dominio cognoscitivo, habilidades, destrezas, intereses, capacidades y aptitudes.

La **prueba pedagógica u objetiva**, es un instrumento de medición compuesto por una serie de reactivos, que permite apreciar si los objetivos de aprendizaje han sido logrados, son de gran valor en el área cognoscitiva. Una prueba pedagógica puede incluir reactivos del tipo: i) Complementación ii) Respuesta Breve c) Tema breve d) Identificación e) Correspondencia f) Opción múltiple: de respuesta única y de respuesta múltiple [López 2004]

La llamada “educación virtual”, en sentido estricto y los sistemas intermedios entre la enseñanza formal escolarizada, como son los sistemas “semi presenciales y a distancia”, han explorado diversos procedimientos de evaluación y autoevaluación (ejercicios de autoevaluación), basados en exámenes que de acuerdo a las condiciones, se orientan más a una u otra de las cuatro modalidades señaladas.

Veamos a continuación, ventajas y desventajas en el contexto del aprendizaje o formación en “línea” (e-learning), así como sus posibilidades de aplicación de algunos tipos de pruebas pedagógicas:

a) El examen oral. Permite interacción directa con el alumno y el profesor; ahí el alumno no puede copiar, su relevancia es tal que se siguen utilizando en la obtención de grados académicos. La limitación es la necesidad de sincronía con el tutor, facilitador o profesor responsable de curso, así como los requerimientos tecnológicos para la comunicación personalizada con todos los examinados.

b) El examen de desarrollo de un tema. Permite evaluar capacidades de razonamiento, de crítica, si la respuesta tiene congruencia lógica; requiere capacidad de redacción y puede influir mucho el criterio del profesor para la calificación. Aunque, se puede desarrollar “en línea”, asincrónicamente no se puede por el momento apoyar por medio de un sistema computacional, por el desarrollo aún limitado de la tecnología de la inteligencia artificial.

c) El examen con reactivos de respuestas corta. Utiliza la memorización, los alumnos responden de la misma manera por lo que pueden copiarse; esta modalidad puede apoyarse en un sistema computacional, puede automatizarse (ejemplo 3.1):

Ejemplo 3.1

En C una variable float corresponde a un almacenamiento de ____ bytes

Cuadro 3.1 Reactivo de respuesta corta.

d) El examen con reactivos opción múltiple. Es el que mejor se puede apoyar en la etapa actual en un sistema computacional, los alumnos responden fácilmente (si tienen los conocimientos básicos). Los reactivos pueden ser con a) respuesta única b) con respuesta múltiple.

Reactivos de opción múltiple de respuesta única. Lo idóneo es la utilización de reactivos con cinco opciones ya que puede haber acierto por azar, lo cual hace poco recomendable a las opciones de falso y verdadero, que también, desde luego se pueden utilizar. En ciertas situaciones, también se aplican a cursos de matemáticas o física (ejemplo 3.2); sin embargo, hay ciertas dudas de los profesores de matemáticas en su utilización.

Ejemplo 3.2

El volumen de la esfera se calcula mediante

(a). $V = \frac{4}{3}\pi r^3$, (b). $V = a^3$, (c). $V = \frac{5}{2}\pi r^3$,
 (d). $V = \sqrt{a^3}\pi r$, (e). $V = \pi r^2 h$,

Cuadro 3.2 Reactivo de opción múltiple de respuesta única

Reactivos de opción múltiple de respuesta múltiple. Se debe incluir en el reactivo la información indispensable. Si se seleccionan tres respuestas, se deben incluir cuatro distractores (ejemplo 3.3).

Ejemplo 3.3

El lenguaje JavaScript

- | | |
|---|-----|
| 1. Se basa en clases | () |
| 2. Un objeto de la jerarquía superior de JavaScript es document | () |
| 3. Es interpretado | () |
| 4. Requiere como en C++ definir el tipo de las variables | () |
| 5. Math es un objeto del navegador | () |

Cuadro 3.3 Reactivo de opción múltiple de respuesta múltiple

e) Reactivos de correspondencia. La columna de respuestas debe tener al menos dos distractores, los términos correspondientes no deben quedar en la misma línea, así como debe haber afinidad en ambas columnas [López, 2004], ejemplo 3.4.

Ejemplo 3.4**Instrucciones: Escriba en la derecha el número que corresponda**

- | | |
|-----------------------|---------------|
| 1. Bejarne Stroustrup | () Fortran |
| 2. Alan Kay | () C++ |
| 3. Dahl y Nygoard | () Smalltalk |
| 4. Grady Booch | () C |
| 5. John Backus | () Simula67 |
| 6. James Gosling | |
| 7. Steve Jobs | |

Cuadro 3.4 Reactivo de correspondencia

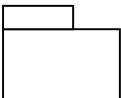
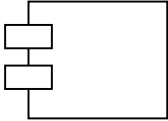
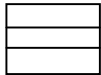
Los reactivos de correspondencia adoptan variantes en desarrollos multimedia; en ellos se mueven objetos con el apuntador de ratón, suele usarse mucho en software educativo para niños.

En el IPN el Dr. Leonid Sheremetov y el M. en C. Rubén Peredo [Sheremetov, et al. 2006] han desarrollado sistemas de evaluación para el Web que incluyen dichos reactivos, utilizando ActionScript el lenguaje de programación de Flash, programa integrador de medios. De hecho, son variantes de las propuestas que se pueden desarrollar con KnowledgePresenter®2006 de Flash, el cual se discute más adelante.

f) Reactivos de Identificación. Se utilizan esquemas, dibujos, mapas; se trata de asociar respuestas con alguna imagen o porción de ella. Es recomendable no se incluya menos de cinco cuestiones, ni más de 12; las preguntas deben tener afinidad y no mezclar temas diferentes [López, 2004]. Fácilmente se pueden realizar propuestas de reactivos multimedia como los propuestos en el caso de correspondencia. El ejemplo 3.5 muestra un reactivo de identificación:

Ejemplo 3.5

Escriba en el paréntesis el número que corresponda

- | | | |
|-----------------------------|----|---|
| () Objeto | 1. |  |
| () paquete | 2. |  |
| () Agregación | 3. |  |
| () Componente | 4. |  |
| () Relación de dependencia | 5. |  |

Cuadro 3.5 Reactivo de Identificación

En los primeros cinco años del siglo XXI, los sistemas evaluadores WBT todavía eran bastante rudimentarios, era común que la mayor parte del software relacionado con la evaluación fuese, como bien hacen notar Peredo y Sheremetov [Peredo, et al. 2002], se limitada a reactivos de opción múltiple y falso verdadero (una variante de la primera), restringiendo con ello las posibilidades de retroalimentación del alumno, la qué, de hecho como tal objetivo no parece existir.

En el momento de concluir esta tesis, existe ya una gran variedad de propuestas de reactivos, para el Web aprovechando las capacidades multimedia que este entorno permite. Las plataformas tienden a incorporar herramientas de autoría (creación) para que los profesores participen en el diseño de diversas variantes de los reactivos, anteriormente descritos.

3.2 Las propuestas actuales de evaluación en la formación en línea (Online)

3.2.1 Plataformas internacionales

En el mundo existen diversas compañías que se han interesado en la administración de ambientes virtuales de aprendizaje; sin embargo a la cabeza se hallan Blackboard y Moodle (Modular Object Oriented Dynamic Learning Environment, *Ambiente de Aprendizaje Dinámico de Módulos Orientados a Objetos*); se describe también Claroline y Maple, así como una herramienta de evaluación que es Hot Potatoes. Algunos otros menos relevantes se indican en un resumen presentado en la Tabla 3.1

1) Blackboard

BlackBboard ha sido instalado en el IPN, es un administrador de recursos educativos y de usuarios, hasta el momento ha tenido poco éxito en el IPN tanto por la poca participación de la comunidad politécnica en la educación no presencial, la falta de comprensión de lo que se debe hacer en este contexto y como siempre, por el divorcio entre los grupos de productores de multimedia en el IPN y las autoridades impulsoras de estos proyectos.

La versión actual Blackboard Academic Suite [Web4, 2006] es una familia de aplicaciones integradas para el aprendizaje, construcción de comunidades de aprendizaje y compartición del conocimiento, consta de: i) Blackboard Learning System ii) Blackboard Content System iii) Blackboard Community System.

-Blackboard Learning System. La compañía Blackboard anuncia esta parte de la suite como la que aporta herramientas que incrementan la eficiencia del instructor en la evaluación de las habilidades del estudiante con bancos de reactivos personales o institucionales; con preguntas del tipo falso o verdadero (opción binaria) o de opción múltiple (de respuesta única o con respuesta múltiple), reactivos de espacios en blanco, respuesta breve, ensayo, con o sin límite de tiempo. Las evaluaciones pueden hacerse una sola vez o pueden repetirse. Se acompaña de reporte del progreso del estudiante [Web5, 2006].

-Blackboard Content System, permite a los usuarios almacenar y administrar contenidos y mantener diferentes versiones de los documentos administrando el espacio en disco y el ancho de banda.; en la versión del IPN, el profesor instala su material en un espacio gráficamente identificado a modo de carpetas.

-Blackboard Community System, busca crear un ambiente de aprendizaje en línea, abierto, interoperable con las normas de la industria. Cuenta con un portal para cada campus que permite el acceso a la información escolar, como son los resultados de exámenes. La información se asegura para los niveles individual, de grupo e institucional. Puede administrar diversas instituciones y administra e-Commerce de libros y membrecías organizacionales.

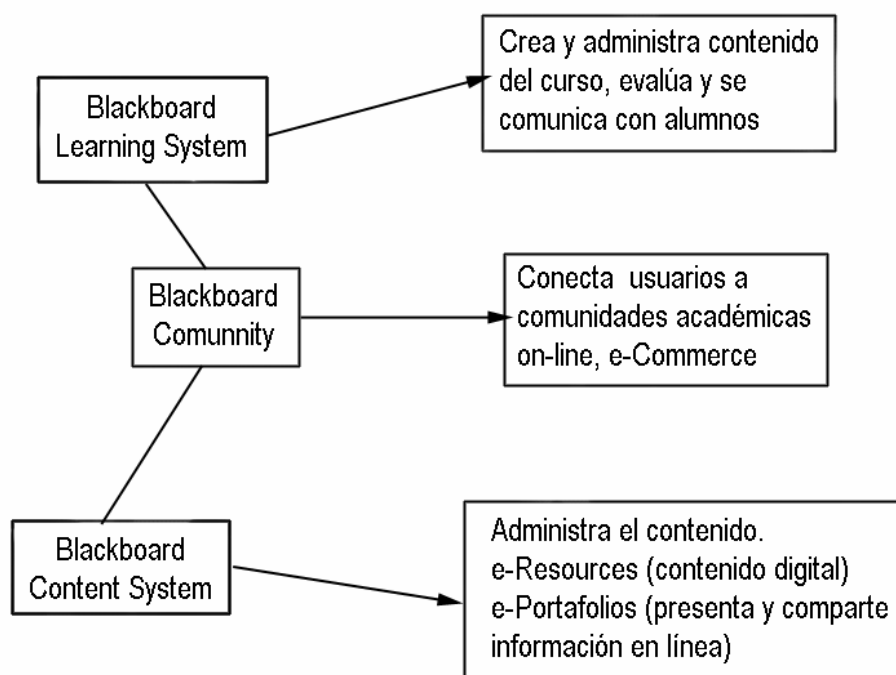


Fig. 3.2 Los módulos de Blackboard

Los profesores pueden distribuir sondeos y evaluaciones con puntajes automáticos. Pueden crear estas evaluaciones desde cero o basándose en “banco de evaluadores” con preguntas elaboradas personalmente, por la institución o disponibles en el mercado.

II) Moodle

Moodle, es un sistema de administración de cursos (Course Management System -CMS-), también autocalificado como Ambiente Virtual de Aprendizaje (Virtual Learning Enviroment -VLE-). [Web6, 2006] es software de código fuente abierto, diseñado usando “sólidos principios pedagógicos” para ayudar a crear comunidades de aprendizaje en línea.

Los trabajos que se asignan a los alumnos son recuperados por éstos, mediante la herramienta de evaluación o asignación (*Assignment Tool*), contiene tareas que pueden involucrar texto en línea o presentaciones en clase. En particular la herramienta *Choice*, genera preguntas con cierto número de opciones que permite retroalimentar el curso que se está impartiendo. [Web7, 2006].

Moodle, cuenta con herramientas para generar las preguntas “en línea”, crea el examen automáticamente, también genera reactivos de relación. Se pueden obtener los resultados de esta evaluación y presentar retroalimentación, señalando el error del alumno.

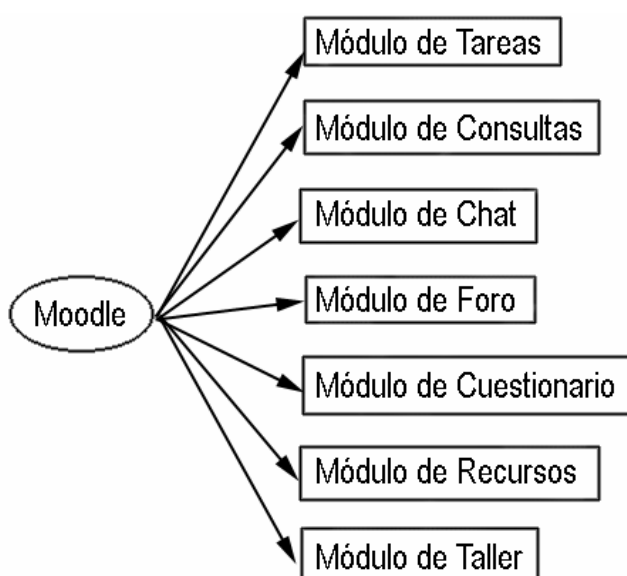


Figura 3.3 Módulos constituyentes de Moodle

Siendo realmente de nuestro interés para la tesis se describen a continuación el Módulo Cuestionario de Moodle (Web8, 2006)

- Los profesores pueden definir una base de datos de preguntas que podrán ser reutilizadas en diferentes cuestionarios.
- Las preguntas pueden ser almacenadas en categorías de fácil acceso, y estas categorías pueden ser "publicadas" para hacerlas accesibles desde cualquier curso del sitio.
- Los cuestionarios se califican automáticamente, y pueden ser recalificados si se modifican las preguntas.
- Los cuestionarios pueden tener un límite de tiempo a partir del cual no estarán disponibles.
- El profesor puede determinar si los cuestionarios pueden ser resueltos varias veces y si se mostrarán o no las respuestas correctas y los comentarios.

-
- Las preguntas y las respuestas de los cuestionarios pueden ser mezcladas (aleatoriamente) para disminuir las copias entre los alumnos.
 - Las preguntas pueden crearse en HTML y con imágenes.
 - Las preguntas pueden importarse desde archivos de texto externos.
 - Los intentos pueden ser acumulativos, y acabados tras varias sesiones.
 - Las preguntas de opción múltiple pueden definirse con una única o múltiples respuestas correctas.
 - Pueden crearse preguntas de respuesta corta (palabras o frases).
 - Pueden crearse preguntas tipo verdadero/falso.
 - Pueden crearse preguntas de emparejamiento.
 - Pueden crearse preguntas aleatorias.
 - Pueden crearse preguntas numéricas (con rangos permitidos).
 - Pueden crearse preguntas de respuesta incrustada (estilo "cloze") con respuestas dentro de pasajes de texto.
 - Pueden crearse textos descriptivos y gráficos.

Dada su naturaleza de ser un software libre, de código abierto Moodle a finales del año 2006, comenzó a ser utilizado por algunas instancias del IPN, en particular la coordinación del Campus Virtual Politécnico.

III) Hot Potatoes.

Hot Potatoes, es un conjunto de herramientas de autoría (authoring) desarrollado por el equipo del University of Victoria CALL Laboratory Research and Development que, permite elaborar ejercicios interactivos basados en páginas Web de seis tipos básicos. La interactividad se logra con JavaScript. Las herramientas son **JQuiz**: Crea ejercicios de elección múltiple. Esta herramienta en la versión 6.0 fusionó a otra denominada **JBC** [Web8c, 2006]. **JCloze**: Crea ejercicios de rellenar huecos. **JCross**: Crea crucigramas. **JMatch**: Crea ejercicios de emparejamiento u ordenamiento. **JMix**: Crea ejercicios de reconstrucción de frases o frases a partir de palabras desordenadas. Se trata pues de un software que permite crear los reactivos, sin que el profesor tenga que meterse a programar en JavaScript al modo en que se ha hecho en este tesis, en Hot Potatoes sólo se elige el modelo de ejercicio y se introducen los datos [Web8b, 2006].

Hot Potatoes, es considerado como una herramienta externa de Moodle [Web9, 2006], programa que incluye interfaces para generar evaluaciones con distintos tipos de preguntas, se halla constituido de varios módulos que permiten 1) Crear texto con espacios en blanco que deben ser llenados 2) Crear cuestionarios de respuestas múltiples, cortas, híbridas y múltiple selección 3) Preguntas de relación de los elementos de una lista con otra. 4) Armar frases dadas sus partes 5) Crucigramas con campos de texto para añadir una palabra [Web9, 2006].

IV) Claroline

Claroline es un software basado en PHP y MySQL para administrar cursos en línea. Se propone diseñar y sincronizar el aprendizaje colaborativo, compartiendo calendarios, área de documentos, herramientas de administración de grupos. Utiliza reactivos (llamados “quiz”) de **opción múltiple propuestos mediante un generador de reactivos**. Utiliza los estándares de SCORM (Sharable Content Object Reference Model, *Modelo de Referencia de Objetos Distribuibles*) y de IMS (Global Learning Consortium Inc, *Consortio de Aprendizaje Global*), organismo éste último que promueve la adopción de especificaciones abiertas para la tecnología de aprendizaje interoperable. [Web9a, 2006]

V) Maple

Maple 10 es un software muy utilizado y conocido en el área de matemáticas, contiene una herramienta de evaluación, plantillas para crear preguntas de relleno de espacios en blanco, y de opción múltiple. Las preguntas pueden contener gráficos, consejos, retroalimentación y variables generadas aleatoriamente. [Web9b, 2006]



Figura 3.4 Ejemplos del software para la enseñanza asistida por computadora.

VI) KnowledgePresenter®2006

Es un software de Macromedia Flash, especializado para crear reactivos, lecciones y presentaciones. Una herramienta de autoría, sin “plugins”, la versión: KnowledgePresenter®2006 Profesional; incluye KPLearning Manager System que, genera reportes, enlaza a cursos, administra usuarios; KP-Simulator permitiendo la captura de pantalla, creando

animaciones. KP-Capture captura imágenes; KP-Documentary Marker, crear videos a partir de imágenes.

Con KnowledgePresenter, se pueden crear reactivos multimedia de **selección múltiple**, **selección múltiple con aleatorización** (con tiempo límite y retroalimentación), **emparejamiento** (con capacidad de arrastra texto), **respuesta breve**, **listas desplegables** (respuesta breve que se selecciona de una lista desplegable), **entrada de texto forzada** (en un cuadro de diálogo sólo se muestran las letras que darán la respuesta correcta), **blancos** (se pulsa sobre la parte de la imagen que responda a la pregunta) [Web9c, 2006]. La Figura 3.5, es la propuesta de herramientas de Flash para el mundo e-learning (Online):

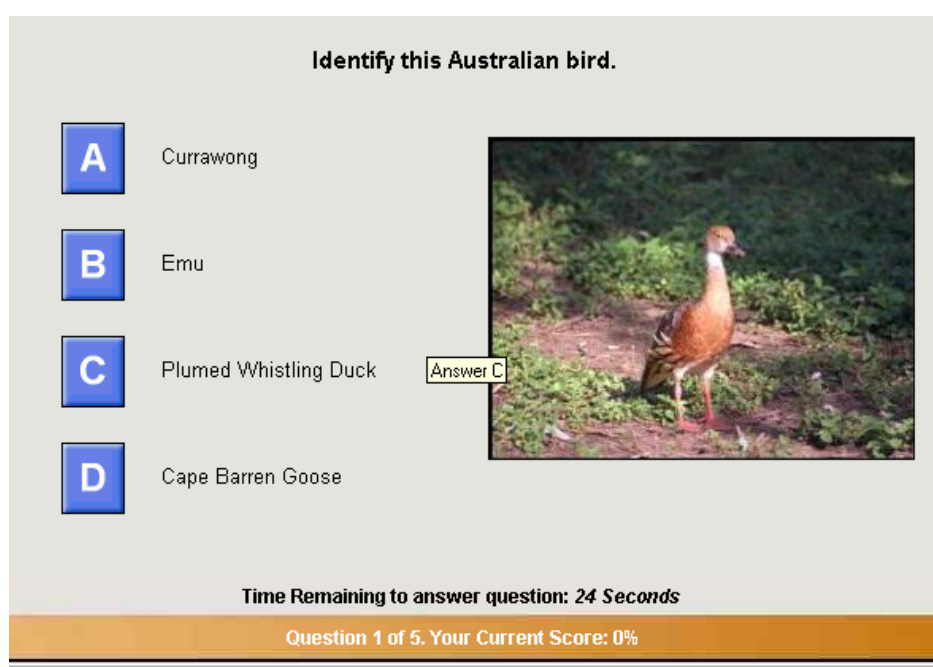


Figura 3.5 Reactivo de selección múltiple creado con la aplicación de Flash llamada: KwnoledgePresenter®2006

VII) Otros productos semejantes.

En la Tabla 3.1, se incluyen algunas herramientas elaboradas en diversas universidades para llevar a cabo la evaluación y gestión a través del Web

Tabla 3.1 Algunos herramientas de software para la evaluación y su dirección electrónica.

Herramienta	Procedencia	URL. Internet
CASTLE	Leicester University	http://www.le.ac.uk/cc/ltg/castle
CVU (Clyde Virtual University)	Strathclyde University	http://www.googlesyndicatedsearch.com/u/strathclydeuni?q=CVU
GEODATA	SouthHampton University	http://www.geodata.soton.ac.uk/elecpub/quiz.html
waMCQ	Dundee University	http://www.dundee.ac.uk./meded/mcq
NetQuest	Bristol University	http://www.ilrt.bris.ac.uk/netquest/

La tabla evidencia que las universidades, sobre todo en Inglaterra están incorporando sistemas automatizados para la evaluación en línea.

3.2.2 La experiencia nacional

I) Los sistemas EVA y SOFÍA en el IPN

El IPN también ha intentado avanzar en el desarrollo de software para la administración de enseñanza Online. Un proyecto al cual se le dio gran relevancia fue EVA (*Espacios Virtuales de Aprendizaje*); sin embargo al no comprenderse que era un proyecto de investigación el cual debía de aprender de sus errores tras un proceso de prueba, permitiéndole continuar su desarrollo, se le sobrestimó y cayó en la desgracia del desinterés político-institucional y fue reemplazado por SOFÍA (*Sistema Orientado Fundamentalmente a la Individualidad de Aprendizaje*), que aunque también llegó a existir como software, su presencia institucional fue detenida.

Eva, tendería a constituirse en un **LCMS**, uno de sus pilares debería ser el contenido reutilizable (objetos de aprendizaje reutilizable), consideraba los siguientes espacios:

- i) *Espacio de Conocimiento* con toda la información académica
- ii) *Espacio de Consulta*, donde los alumnos podrían preguntar a los maestros o condiscípulos
- iii) *Espacio de Colaboración*, estableciendo enlaces entre los mismos miembros de la comunidad virtual
- iv) *Espacio de Tecnificación*, que básicamente de refiere a laboratorios donde los alumnos pueden experimentar [Web10, 2006].

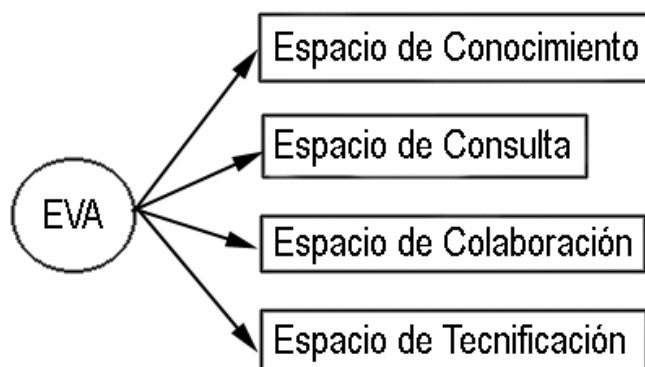


Figura 3.6 Los espacios de EVA

Eva contaba con una base de datos, cuyo sistema de Bases de Datos llamado manejador Access no pudo responder a la demanda provocando la “caída del sistema”, ello podía corregirse, más se le negó apoyo. Aunado a esto hay que tomar en cuenta que la comunidad politécnica no estaba ni administrativa, ni tecnológicamente preparada para incorporarse a esta plataforma.

Muchos grupos creativos del IPN, que aportaban desarrollos que podían ser útiles a EVA no fueron incluidos o informados, ni siquiera invitados formalmente a enriquecerla, y quienes así lo hicieron no lograron desarrollar el concepto de "polilibro" más allá de una versión digitalizada del clásico apunte en el IPN; por ende, no se podía esperar en modo alguno una conexión con un sistema de evaluación, en las etapas tempranas del proyecto.

Sin embargo los impulsores de EVA como el Dr. Leonel Sheremetov y el M. en C. Rubén Peredo trabajaron arduamente para crear un sistema de evaluación para EVA utilizando Componentes Reutilizables de Aprendizaje, con la herramienta Flash, así como Objetos Inteligentes Reutilizables [Sheremetov, 2006] [Peredo, et al. 2004].

La “segunda etapa de EVA”, se denominó SOFÍA, pretendió ser independiente de pedagogías, de manera que cualquier institución o corriente pudiera utilizarla sin ambages. SOFÍA consiste en un esquema de administración del conocimiento que lo administra *per sé* y además, administra a la gente que trabaja con el conocimiento. Otra gran diferencia es la concepción de materiales de estudio, el concepto de “polilibro” es sustituido por el de “átomos de conocimiento”.

SOFIA, es para el público en general (no sólo para el medio académico) como EVA. Los espacios de SOFIA son análogos a los de EVA pero la manera de trabajar con ellos es diferente.

Un coordinador de SOFIA, el Dr. Juan Luis Díaz de León llegó a manifestar *“Yo soy de la idea de que se puede aprender a distancia, pero no podemos certificar a distancia. Se podrán inventar 20 mil artilugios para comprobar la legalidad de un examen a distancia; sin embargo, yo considero que no hay nada como estar presente para evaluar”* [Web11, 2006].

Las últimas líneas, dan la impresión de una subestimación en SOFÍA, del proceso evaluatorio; sin embargo, la Dra. Victoria Bajar Símsolo [Web12, 2006] y colaboradores lo plantean de forma diferente; pues los exámenes se constituyen en cada caso en forma particular, tomando enunciados en forma aleatoria de un gran repositorio de preguntas que se llama: “problemario”.

Cada vez que se solicita un examen, éste se genera, por tema o por unidad por lo cual no hay exámenes repetidos, a menos que el repositorio sea reducido; es decir, muy pocos enunciados. De acuerdo, a la misma autora SOFÍA es la reunión de cuatro espacios que se describen en la Figura 3.7:

Espacio de Conocimiento	Espacio de Administración
Espacio de Colaboración	Espacio de Navegante

Figura 3.7 Los espacios de SOFÍA

En el espacio de conocimiento se ubica la información que da pie al conocimiento, disponiéndose en forma de árbol. Los elementos básicos fundamentales para llevar a cabo los procesos de aprendizaje en SOFÍA, son las hojas del árbol, llamadas unidades de conocimiento. Los materiales de estudio reciben el nombre de objetos de aprendizaje. Los usuarios buscan la información navegando, el espacio de navegación registra su avance. Hay exámenes por tema si el usuario es “regular” (inscrito en un programa) o por unidad si el navegante no está registrado en un programa formal [Web12, 2006].

SOFÍA preveía la existencia de preguntas con respuestas de **opción múltiple** o por **opciones múltiples**, preguntas de **respuesta fija** con cadena de caracteres y preguntas de **respuesta abierta**. Las primeras dos se **evalúan automáticamente**, las últimas requieren intervención de un evaluador [Web12, 2006].

III) COSNET

Otra experiencia mexicana relevante ha sido la del Consejo del Sistema Nacional de Educación Tecnológica (COSNET), con la creación de un sistema de evaluación en línea del examen de ingreso a la Educación Media Superior Tecnológica. En la actualidad la Subsecretaría de Educación e Investigación Tecnológica (SEIT) a través del Consejo del Sistema Nacional de Educación Tecnológica coordina la evaluación de la Educación Media Superior Tecnológica.

La evaluación de ingreso proporciona un diagnóstico de los aspirantes a ingresar a la educación media superior tecnológica con el fin de implantar acciones que mejoren y fortalezcan su perfil de ingreso para contribuir a cursar satisfactoriamente el tipo correspondiente.

La finalidad del examen de ingreso a la educación media superior del Sistema de Educación Tecnológica es identificar habilidades con que cuentan los aspirantes para la solución de problemas a partir de análisis de situaciones cotidianas, la elaboración de hipótesis y juicios; la solución de problemas matemáticos, el dominio y comprensión del lenguaje escrito. En consecuencia, el examen cubre tres áreas: Razonamiento formal, capacidad de aprendizaje de las matemáticas y habilidad verbal.

El sistema de evaluación en línea del examen de ingreso a la educación media superior tecnológica "utiliza la tecnología y bondades de Internet, tiene como objeto ser una herramienta de apoyo académico para los aspirantes, profesores y autoridades involucradas en el proceso educativo" [Web13, 2006]

La realización de la evaluación en línea ha permitido eliminar procesos y costos que implican reproducción y empaquetado de materiales, distribución a las diferentes direcciones generales, estatales, planteles, proceso de captura, de información a los planteles, envío de resultados a diferentes dependencias, definición de los resultados. A estos procesos que "no agregan valor al proceso de evaluación" implicaban la participación de muchas personas de diferentes instancias administrativas de la SEIT.

Con el proceso antiguo había problemas de seguridad por posible filtración de información debido a los puntos intermedios que intervenían. Los costos de 12 millones de pesos, se redujo a 1 millón de pesos.

Las ventajas más destacadas son:

- i) Se reduce la cadena de distribución.
- ii) Reducción de los costos de operación.
- iii) Reducción de los tiempos de distribución y trabajo.

El sistema propuesto por COSNET, pretende sustituir la evaluación tradicional, así como la distribución de los instrumentos de la evaluación, se podrá conocer los resultados de la evaluación desde cualquier parte del mundo y disponer de la información en cualquier momento. Es de resaltar que antes de ponerse en práctica en el proceso real, se realizó una prueba piloto el 1 de julio de 2004.

El sistema de evaluación en línea usa "tecnologías Web" como es PHP, JavaScript entre otras. Trabaja en servidores Web UNIX, LINUX y Windows. Realiza sus operaciones combinando datos generales de páginas de contenidos dinámicos con el manejo de MySQL.

El examen de ingreso en línea a la educación superior consta de tres módulos por el momento, el *Sistema de administración remota*, el *DEMO* y la *Aplicación* del examen de ingreso en línea (Figura 3.8):

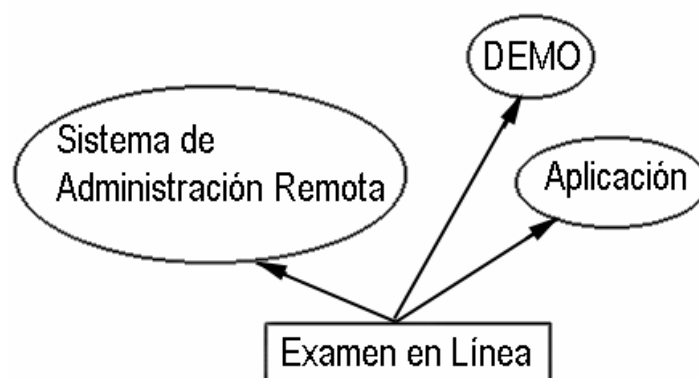


Figura 3.8 Módulos del Examen en Línea de COSNET

Sistema de Administración del Sistema COSNET

Realiza múltiples tareas administrativas de una manera cómoda y personalizada, permite a las institutos administrar los aspirantes de nuevo ingreso, descargar los instrumentos y herramientas no sólo de la evaluación del ingreso, sino que además de los estudios de la evaluación institucional del segmento de egresados y profesores que coordina COSNET.

Los módulos DEMO y Aplicación del sistema COSNET

Consta de: 1) Bienvenida al sistema y autenticación por parte del aplicador del plantel; 2) Selector del plantel a donde el aspirante debe ingresar; 3) Registro o verificación de datos personales y escolares del aspirante; 4) Preparación del aspirante a realizar la evaluación; 5) Nivel del razonamiento formal; 6) Nivel de capacidades de aprendizaje de matemáticas; 7) Nivel de habilidad verbal; 8) Presentación de los resultados de evaluación. *El examen consta de 110 preguntas con cinco opciones de respuesta cada una.* [Web13, 2006]

La herramienta DEMO tiene como objetivo capacitar a los aspirantes mostrándoles cómo será la Aplicación en Línea.

III) La experiencia de la Maestría en Ingeniería de Sistemas de la ESIME Zacatenco

Una última experiencia analizada en este proyecto, es la del sistema semipresencial y a distancia del Programa de la Maestría en Ingeniería de Sistemas de la Escuela Superior de Ingeniería Mecánica y Eléctrica Unidad Zacatenco del IPN, utilizando el sistema de video conferencia se impartió desde la ciudad de México hacia los nodos: Campeche, CanCún, Culiacán, Morelia, Reynosa, Tijuana, Mexicali, Ensenada y ESIME Allende en el D.F.

En esta experiencia se desarrollaron, para la mayor parte de los cursos, materiales de apoyo multimedia, materiales despleables como páginas Web que lo mismo se entregaban en CD-ROM que se tenía acceso a ellos desde la red del IPN. Las evaluaciones se realizaron en la mayor parte de los casos de forma tradicional. Una gran ayuda para establecer la comunicación con los alumnos fue el correo electrónico y la introducción de “Chats”; se sabe de un trabajo incipiente basado en Delphi de reactivos de selección múltiple para una materia que no se consolidó.

En dos materias, se comenzó a experimentar una modalidad automatizada de autoevaluación, que no sólo informara de los errores cometidos, sino también retroalimentara a los alumnos, cuyo programa de cómputo fue

desarrollado por el sustentante de la presente tesis, con base en el desarrollo previo de un programa de autoevaluación "clásico". Aunado a ello, se ampliaron las posibilidades de evaluación al crearse el programa "Generador Aleatorio de Exámenes Matemáticos" (GAEM).

Con el programa GAEM, se dio un paso hacia un sistema de utilidad en la evaluación de exámenes matemáticos (o de contenido matemático), su desarrollo fue resultado de la propuesta del profesor Efraín Martínez Ortiz coordinador de la maestría en Ingeniería de Sistemas en su modalidad "virtual", ante las dudas o reticencia de algunos profesores para aplicar pruebas pedagógicas de opción múltiple (o algunas de sus variantes) en las áreas de ingeniería.

De este modo, se logra proponer modificaciones a los tan usados exámenes de selección múltiple y de otro con el GAEM se logra la selección al azar de los reactivos constituyentes del examen, pero con la capacidad de despliegue de imágenes y de toda clase de medios.

Por las características de algunas plataformas de gestión de educación "Online", en términos de si comparten herramientas desarrolladas en la presente tesis y mostrados en la Tabla 3.2, nuestros programas comparten actividades con otros sistemas; sin embargo, es software con cierto grado de originalidad que puede contribuir al desarrollo de la enseñanza basada en el Web dentro del IPN:

Tabla 3.2 Capacidades de evaluación en diferentes plataformas y herramientas de software. *Sin datos

<i>Tipo de Reactivo/ Software</i>	Blackboard	Moodle	Hot Potatoes	Claroline	Maple	EVA	SOFÍA	ESIME Zac.	COSNET
Falso/ verdadero	Si/Aut	Si/Aut.	Si/Aut.	No	No	Si	Si	Si/Aut	No
Opción Múltiple	Si/Aut.	Si/Aut.	Si/Aut.	Si/Aut	Si	Si	Si	Si/Aut.	Si/Aut.
Correspondencia	Si/Aut.	Si/Aut	Si/Aut	*		Si	Si	No	No
Respuesta breve	Si/Aut.	Si/Aut	Si/Aut	*	Si	Si	Si	No	No
Ensayo	Si/No Aut.	Si/No Aut.	*	*	No	No	Si	No	No
Examen aleatorio	No	Si	*	Si		Si	Si		
Examen matemático aleatorio	No	No	*		Si	No	No	Si/No Aut.	No

Resumen del capítulo

La evaluación es un proceso continuo cuyo objetivo es medir. Uno de los enfoques que aún se sostienen es el de evaluar el logro de objetivos de acuerdo a las categorías de Bloom, una forma de evaluar es utilizando las pruebas pedagógicas, de las cuales hay varios tipos de acuerdo a los reactivos: de ensayo, respuesta corta, opción múltiple, correspondencia. Los tres últimos tipos de reactivos son los que mejor se prestan para la evaluación automatizada.

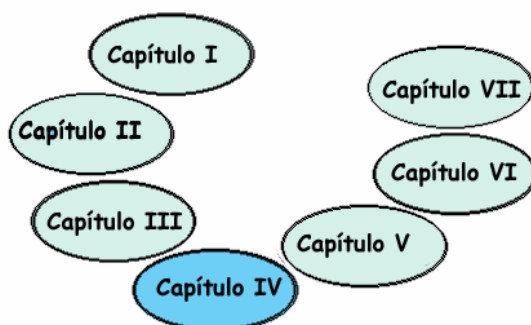
En la tesis se pretende describir la organización de programas de cómputo desarrollados por el sustentante tanto para la corrección automática de pruebas pedagógicas de opción múltiple (incluyendo las de falso/verdadero), como la generación aleatoria de exámenes matemáticos.

Las pruebas pedagógicas basadas en reactivos de opción múltiple, se hallan presentes en las plataformas internacionales como Blackboard, Moodle, Hot Potatoes, algunos de estos sistemas de software son incorporados a los ambiente virtuales de aprendizaje en el IPN, por lo cual ante la necesidad de desarrollos propios, se justifica el desarrollo de software como aquí propuesto.

Habiendo establecido los objetivos y haciendo una revisión de software con posibilidades de evaluación en línea, lo cual permite valorar la aportación de los programas de cómputo propuestos en la presente tesis, procederemos a su descripción, comenzando con los programas de autoevaluación

CAPÍTULO IV.-

Arquitectura de los Programas de Autoevaluación "Clásico" y "Modificado"



Contenido del Capítulo

Se hace una descripción técnica de los programas de autoevaluación “clásico y modificado”, que abarca los lenguajes utilizados y las plataformas en que han sido probados.

En este capítulo, se explica la razón del uso de JavaScript en el desarrollo de los programas, así como algunas de sus peculiaridades que dan lugar a la utilización crítica de los elementos de representación propios de UML (Unified Modeling Language, *Lenguaje Unificado de Modelado*). Asimismo se justifica el paso del código HTML (Hypertext Markup Language, *Lenguaje de Marcado de Hipertexto*) a XHTML (eXtended Hypertext Markup Language, *Lenguaje de Marcado de Hipertexto Extendido*), en el desarrollo de las páginas Web.

Mediante diagramas de casos de uso se describen la funcionalidad del sistema desde el punto de vista del usuario. Se detallan componentes y objetos participantes así como sus dependencias; se modela la vista física con diagramas de actividad de las funciones JavaScript de los objetos HTML que intervienen en la presentación de un examen, su corrección automatizada de y la retroalimentación para el alumno.

4.1 Descripción general

Los fundamentos para la evaluación automatizada de pruebas pedagógicas del tipo opción múltiple de respuesta única o múltiple se halla en los dos programas que el autor de la tesis desarrolló y los cuales se han denominado "clásico" y "modificado".

El programa de "clásico", se denomina así, porque simplemente realiza su tarea de dar un resultado automatizado, contabilidad de errores si los hay, sin intervención del profesor. El usuario no puede esperar más allá de esta respuesta, sobre todo si se trata de un ejercicio de "autoevaluación". Corresponde al tipo de programa de evaluación automatizada más difundido durante los primeros años del siglo XXI, cuando se hizo cada vez más relevante la educación en línea.

El programa "modificado", al igual que el anterior contabiliza errores e indica los reactivos errados, pero añade retroalimentación, la cual puede ser tan compleja como se desee; hecho muy relevante cuando el propósito real del sistema educativo es el aprendizaje por encima de la exclusión de alumnos.

A continuación, se realizará la descripción técnica y arquitectónica de los dos programas de autoevaluación:

4.2 Descripción Técnica

4.2.1 Sobre los lenguajes de programación utilizados

Los programas de autoevaluación "clásico" y "modificado", se muestran al alumno mediante una página Web; tras resolver el examen y para iniciar la corrección automatizada, se pulsan botones que activan las funciones que realizan el proceso, mostrando el resultado en una misma ventana, incluyendo en el caso del programa de autoevaluación "modificado", la retroalimentación.

Para lograr lo anterior, se ha utilizado HTML (más exactamente XHTML), JavaScript y el lenguaje de hojas de estilo CSS (Hojas de Estilo en Cascada).

JavaScript, es un lenguaje para scripts, archivos de texto que se interpretan a medida que se encuentran: El código de este lenguaje embebido en el documento HTML, es interpretado una y otra vez, conforme

se requiera en la máquina del cliente. El código JavaScript se puede incorporar entre las etiquetas <head> y </head> del documento HTML, la porción de código va entre las etiquetas <script> y </script>.

JavaScript, es un lenguaje con influencia de Java, pero no es Java; algunos lo consideran extensión de HTML, pues lo enriquece con objetos y manejadores de eventos, permitiendo una mayor interactividad de los documentos para su transmisión en Internet y más concretamente dentro de páginas HTML. La facilidad de interactividad, la respuesta rápida a los eventos y facilidad de programación son factores que se consideraron para su elección en el desarrollo de los programas expuestos en la presente tesis.

Ya se ha mencionado en capítulos anteriores que se han realizado desarrollos con **Flash de Macromedia** para actividades análogas a las desempeñadas por nuestros programas. Flash es una herramienta de edición profesional para la creación de publicaciones Web, Flash cuenta con un lenguaje de creación de script que se llama **ActionScript** y es muy parecido a JavaScript.

En la documentación de Flash MX se dice que: **“El documento ECMA-262 de la Asociación europea de fabricantes de PC (ECMA, European Computers Manufacturers Association) se deriva de JavaScript y sirve de estándar internacional para el lenguaje JavaScript, ActionScript se basa en la especificación ECMA-262”** [Flash, 2002]

El M. en C. Ruben Peredo, el M. en C. Leandro Balladares y el Dr. Leonid Sheremetov, critican algunos desarrollos que embebidos como el código JavaScript, reducen la flexibilidad de los componentes [Peredo, et al. 2004]; sin embargo ello se puede superar cargando las llamadas “aplicaciones remotas de JavaScript”, las cuales se almacenan como archivos independientes en el servidor. El atributo src especifica un URL que identifica al servidor y la ubicación del guión que se quiere cargar [Manger, 1997].

Los scripts de JavaScript, durante mucho tiempo se han embebido en páginas Web desarrolladas tradicionalmente en el lenguaje de marcado de hipertexto (HTML, Hypertext Markup Language)); sin embargo, se notará que en nuestro caso hay algunos cambios, pues se trata de una transición a XHTML, que es HTML extendido, basado en XML (eXtended Markup Language), una simplificación de SGML (Standard Generalized Markup Language).

El paso hacia XHTML, se debe a que con el tiempo las páginas Web se fueron convirtiendo en algunos casos en verdaderos ejemplos de mala programación y dejadez, aunque visiblemente aptos. Existen navegadores

"perdona vidas", que han contribuido a realizar mezclas vergonzantes de código estándar HTML y especificaciones particulares de editores navegadores, así como de la "inventiva" del novel programador el cual logra adecuar de algún modo el código, para que sea visiblemente aceptable.

Las páginas Web aquí desarrolladas incluyen requerimientos que facilitarán su migración hacia documentos XHTML entre los que destacan

1. Ser documentos bien formados, con "tags" (etiquetas) de cierre encajando correctamente como en `<p> <b> texto </b> </p>`. Otra forma de sobreponer etiquetas es ilegal.
2. Inclusión de los "tag" de terminación como `<p> </p>`; elementos los cuales HTML 4.0 permitía omitir.
3. Entrecomillado de los atributos (valores)
4. Terminación de los elementos vacíos con `/>` como `<br />`, `<img />`.
5. Anotar los elementos y atributos en minúsculas.
6. Encerrar los elementos de `<script> </script>` y `<style> </style>` dentro de una sección marcada con CDATA como se muestra en el Cuadro 4.1

```
<script>
<![CDATA [
..... Contenido....
]]>
</script>
```

Cuadro 4.1 En XML los elementos del script deben ser incluidos entre la sección CDATA

La última consideración se omite con scripts y hojas de estilo externas, páginas independientes con extensión .js, o .css respectivamente. Los scripts de JavaScript referencian en la etiqueta `<script>` los documentos externos; se trata de las llamadas "aplicaciones remotas" ya mencionada anteriormente, mismas que se han considerado en las páginas Web utilizadas tanto para el código JavaScript, como para las hojas de estilo (CSS).

7. Los documentos XHTML deben hacer referencia a una de las siguientes Definiciones de Tipo de Documento (DTD): **Strict**, **Transitional**, **Frameset**, incorporándose al comienzo del documento.

a) Strict

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/strict.dtd">
```

Se utiliza cuando se da formato a los documentos a través de CSS (Cascading Style Sheets), cuando no se recurre a las etiquetas <table> y

 para controlar la forma en que los navegadores muestran el contenido del documento.

b) Transitional

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/transitional.dtd">
```

Se utiliza cuando no se describe la presentación de los documentos por medio de hojas de estilo en cascada.

c) Frameset

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/frameset.dtd">
```

Se utiliza cuando los documentos incorporan marcos (frames).

8. El elemento raíz debe ser <html> y salvo la declaración del tipo de documento, no debe añadirse nada antes de él, debiendo incluir una etiqueta que especifique el espacio de nombres, que es xmlns con valor xhtml1:

```
<html xmlns="http://www.w3.org/TR/xhtml1">
```

4.2.2 Peculiaridades de JavaScript. Dificultades en la modelación

La razón de esta sección es la de aclarar conceptos para permitir la identificación de los objetos participantes y su representación en términos del Lenguaje de Modelado Unificado (UML), lo cual debe dar lugar a un mejor entendimiento de la arquitectura del sistema de autoevaluación automatizada.

Aunque podemos crear y utilizar objetos e instancias en JavaScript, *no se puede hacer uso de la herencia* y de otras características de la Programación Orientada a Objetos (POO), por ello se dice que dicho lenguaje está *basado en objetos* [Booch, 1994]. En JavaScript al igual que en SmallTalk (propiamente el primer lenguaje realmente orientado a objetos), se habla sólo de objetos y de instancias de un objeto.

JavaScript aunque con una sintaxis análoga a C, es sin embargo, más cercano al lenguaje de programación *Self*, que omite la noción de clases, derivando un objeto de otro (su prototipo), por medio de copiado y

refinado. El concepto de prototipo es una variante de la POO, en la cual el reuso de comportamientos (reconocida como herencia en lenguajes basados en clases), es hecho por clonación de objetos existentes los cuales sirven de prototipos. JavaScript, se considera sin clases, orientado a prototipos; su programación basada en instancias.

En C++ y Java, existe una diferencia entre *clase* e *instancia* de una clase (objeto); las clases definen propiedades y métodos que caracterizan a los objetos, como un tipo de datos abstracto, no describen casos concretos como las instancias u objetos. En JavaScript no existen estas diferencias, sólo existen objetos, se dice que es un lenguaje basado en prototipos: Un objeto es usado como un molde a partir del cual se obtendrán las propiedades iniciales para un nuevo objeto. Cualquier objeto puede ser el prototipo de otro, permitiendo al segundo objeto compartir propiedades del primero.

En lenguajes como C++ o Java, existe herencia a través de las clases, en JavaScript la herencia se da a través del mecanismo de prototipos.

JavaScript cuenta con tres tipos de objetos: i) **Internos** (incorporados o predefinidos) ii) Del **navegador** (objetos de la jerarquía superior) y iii) **Objetos de HTML reflejados**. Los objetos internos, son objetos que el desarrollador puede manipular tales como cadenas, arreglos y fechas.

Entre los “objetos del navegador” destacan *document*, *history*, *location* y *window*. Los “Objetos de HTML reflejados” son las anclas imágenes y objetos de formularios <form> como campos de texto y botones, de hecho las formularios se pueden entender como objetos contenedores de objetos (áreas de texto, campos de texto, botones de radio, casillas de opción).

Los objetos no internos de JavaScript se organizan jerárquicamente; así resulta que los objetos de la de la jerarquía superior tienen descendientes, los cuales **se conocen como propiedades** [Manger, 1997]. Así resulta que un **objeto de HTML** es también una **propiedad** del objeto *document*.

El objeto *document* se constituye de las “propiedades” **links[]**, **images[]**, **anchors[]**, **applets[]**, **forms[]** y **embed[]** que son arreglos que guardan información sobre el tag o etiqueta HTML correspondiente, por ejemplo de las etiquetas , <form>, etc. Las entradas del arreglo guardan la información de los “tag” en el orden de su aparición en el documento HTML. Por su parte **forms[]**, incluye al arreglo **elements[]**, el cual tiene la información de las entradas de la forma.

Así por ejemplo la expresión:

document.forms[1].elements[0].value

Indica el valor del primer elemento, del segundo formulario (<form>), del documento actual.

Las peculiaridades del lenguaje, obliga a realizar la modelación arquitectónica sin considerar el concepto de clase, pero también tratar como objetos a entidades que en lenguajes como C++, serían simples propiedades, como es el caso de los arreglos.

El Lenguaje de Modelado Unificado (**UML**), ha considerado los casos en que los desarrolladores requieren modelar una entidad muy particular, recomendándose utilizar los iconos estándar pues permitirá su representación mediante herramientas ya existentes, en estos casos se usan los "estereotipos", cadenas de texto entrecomilladas, que permiten extender el vocabulario de UML [Booch, et al. 1999].

Pero también UML permite el uso de iconos definidos por el usuario, cuyos nombres se colocan fuera de los iconos [Booch, et al. 1999].

En algún punto de este capítulo en la descripción de las vistas del modelo se expone cómo hemos resuelto algunos problemas de representación mediante iconos definidos por el usuario, mostrándose en las Figuras 4.4 y 4.5.

En la representación de un ambiente Web, ha sido de gran utilidad lo expuesto en la obra conjunta sobre UML de Grady Booch, James Rumbaugh e Ivar Jacobson [Booch, et al. 1999], al tratar los iconos de componentes.

4.2.3 Sobre la plataforma utilizada

Los programas se desarrollaron para ser soportados por las más diversas plataformas existentes en Internet, por ello los programas fueron probados tanto en equipos Macintosh, como PC-IBM; desplegándose en navegadores como Netscape, Explorer, Mozilla FireFox , Flock, Opera y Safari en diferentes versiones.

Se sabe de las incompatibilidades todavía existentes entre los modelos de Objetos de Netscape y Explorer; sin embargo se ha intentado superar estas limitantes al desarrollar código compatible con ambos navegadores.

Los programas se incorporaron en servidores cuyo sistema operativo es Linux, pero también se distribuyeron en CD-ROM.

Con base en lo expuesto los lenguajes y plataformas utilizadas, en los dos programas se describen a continuación:

Tabla 4.1 Lenguajes y plataformas utilizadas en la ejecución de los programas

	Descripción de arquitectura	Lenguaje de Script	Otros lenguajes para el desarrollo de la página Web	Navegador de prueba	Sistema Operativo
Programa de evaluación "clásico" y "modificado"	UML	JavaScript	XHTML CSS	Firefox, Flock, IExplorer, Opera Netscape, Safari	Mac OSX Mac Classic Windows Linux

4.3 La vista de los casos de uso

La primera propuesta de un sistema de autoevaluación automatizada, la versión llamada aquí "clásica", cuyo tipo de reactivos y tareas desarrolladas por el programa, se halla presente en casi todos los sistemas de evaluación, es un simple caso de falso o verdadero; el usuario (actor) elige la opción que considera correcta en cada pregunta y después de haber recorrido todo el examen, el alumno decide solicitar evaluación o volver a intentar contestar el examen. Si solicita evaluación el sistema presentará una mensaje (ventana de alerta), indicando cuáles fueron las preguntas correctas y cuáles las incorrectas.

Tal comportamiento del software debe ser modelado. Los casos de uso representan el comportamiento externo de todo el sistema tal y como se muestra a los usuarios exteriores. Un caso de uso describe qué hace un sistema) un subsistema, clase o interfaz, pero no especifica cómo lo hace.

En la práctica los casos de uso son expresiones verbales que describen algún comportamiento del vocabulario del sistema que se está modelando. Entendiéndose por vocabulario en UML, las cosas que son importantes desde un cierto punto de vista particular. [Booch, et al.1999]

Los diagramas de casos de uso muestran fundamentalmente, a los actores (roles de usuarios) y relaciones de dependencias. El caso de uso produce un valor para un actor. En consecuencia, utilizando los diagramas de casos de uso, se describen los requisitos (propiedades o comportamientos) del programa de autoevaluación "clásico", los "valores" para el usuario (Figura 4.1):

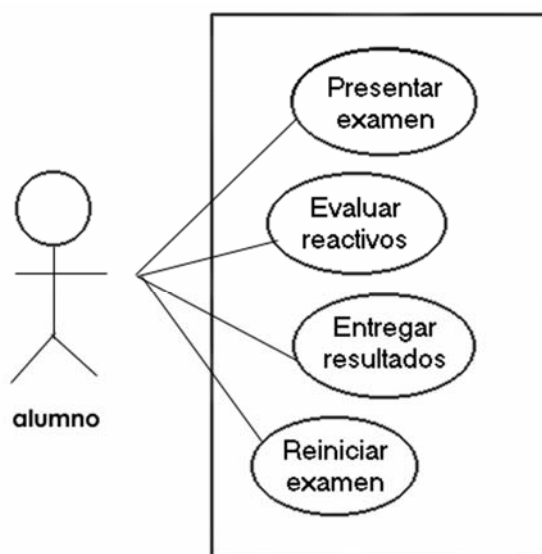


Figura 4.1 Caso de uso del programa de autoevaluación "clásico"

La segunda propuesta de programa, considera que se debe orientar al usuario para mejorar el aprendizaje y no sólo darle la contabilidad de errores y aciertos. Se presenta una ventana explicando en qué consiste su error; se trata de del programa de autoevaluación "modificado".

El diagrama de los casos de uso del programa de autoevaluación "modificado" se muestra en la Figura 4.2:

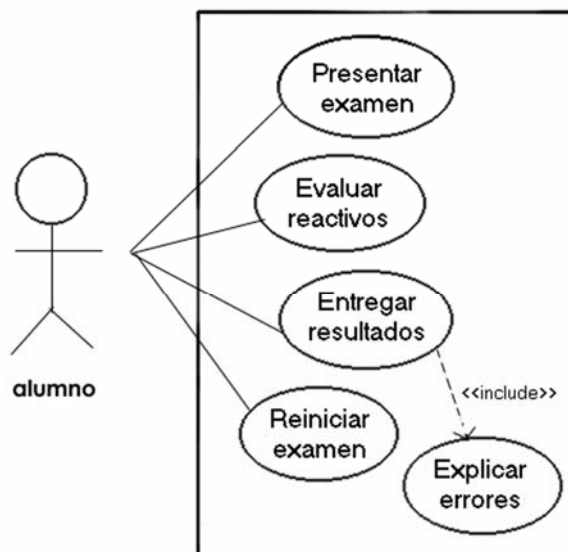


Figura. 4.2 Caso de uso del programa de autoevaluación “modificado”

En el diagrama de casos de uso se muestra un caso de uso subordinado [Booch, et al. 1999], la relación de inclusión identificada mediante <<include>> especifica "la posición en un flujo de eventos en la cual el caso de uso base incluye el comportamiento de otro caso de uso" [Booch, et al. 1999].

4.4 La vista física

La estructura de un programa de cómputo puede ser analizada desde el punto de vista lógico o físico. La vista física modela la estructura de la implementación de la aplicación en si misma, su organización en componentes, las dependencias entre componentes y la asignación de clases a dichos componentes; también modela su despliegue en nodos de ejecución.

Por otro lado en UML, la organización de los elementos de modelado como clases, interfaces, componentes, nodos, colaboraciones, casos de uso, diagramas de casos de uso, interacciones y colaboraciones se realiza en paquetes; los paquetes también pueden contener otros paquetes. En nuestro trabajo los paquetes nos han permitido la organización de los componentes; gráficamente son dos rectángulos y físicamente los hemos hecho coincidir con las carpetas. En este sentido lo que describiría en UML el área de “gestión” (la organización para el trabajo), aquí permite extender la vista física.

En la Figura 4.3, se representan los paquetes del sistema de autoevaluación, destacándose la descripción de los componentes; es decir las partes físicas y reutilizables del sistema, `aevcapx.htm` designa al documento HTML que corresponde a la autoevaluación “x” ($x = 1, 2, \dots$); Por ejemplo si se trata de la evaluación del capítulo 1, sería `aevcap1.htm`. `M2.js` es el nombre del archivo con el código fuente de JavaScript.

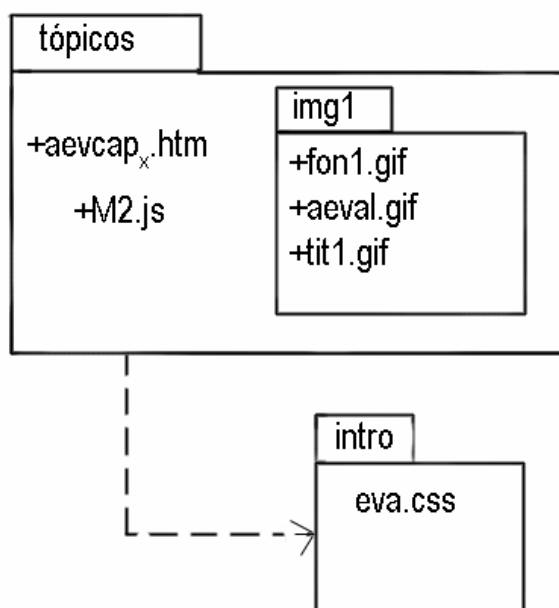


Figura 4.3 Paquetes del sistema de autoevaluación

El paquete “intro”, contiene al archivo con el código de la hoja de estilo denominado `eva.css` y el paquete “img1” contiene la imagen (`fon1.gif`) del background de la página Web `aevcapx.htm` que contiene al formulario desde donde se da la orden de ejecución del programa; elementos no relevantes en la ejecución del programa pero si para configurar la página Web, se trata de las imágenes `aeval1.gif` y `tit1.gif`, que representan etiquetas (letreros).

En UML, se indica la visibilidad de los elementos con los signos + (público), - (privado), # (protegido). Esto permite determinar fácilmente la accesibilidad desde fuera del paquete. La Figura 4.3, también muestra las dependencias entre paquetes, concretamente mediante dependencias de importación.

La Figura 4.4, describe que el documento **`aevcapx.htm`** utiliza (indicado en UML **con una flecha segmentada**) a `M2.js`, `eva.css` y a `fon1.gif`. Lo cual corresponde con las líneas de código del Cuadro 4.2:

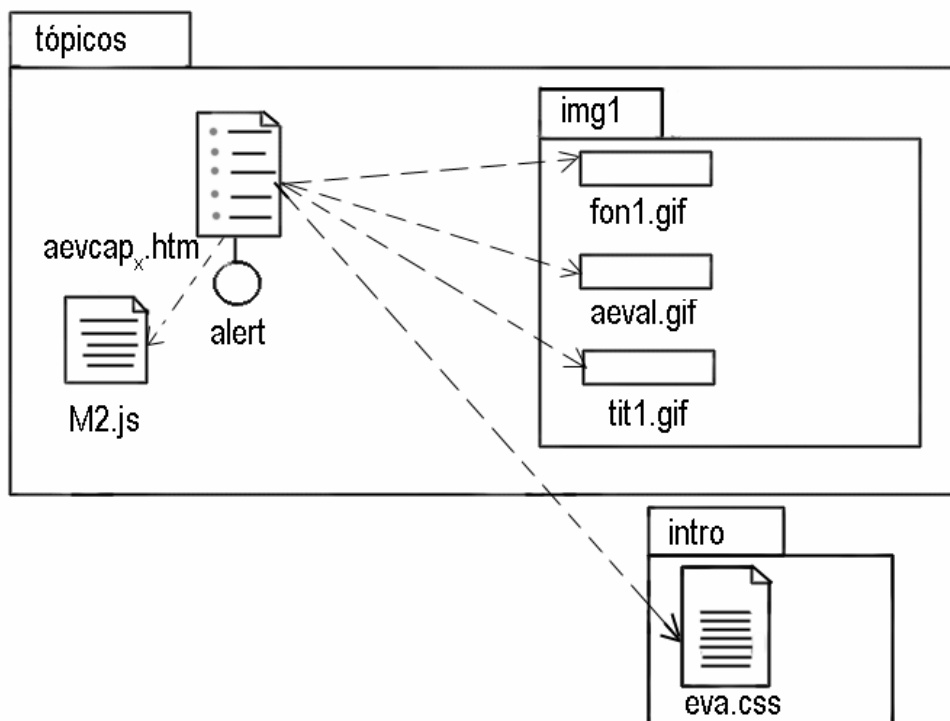


Figura 4.4 Paquetes y componentes del sistema de evaluación "clásico"

UML. permite sustituir los iconos "canónicos" utilizados para representar un componente, por iconos definidos por el usuario, los símbolos definidos por el usuario tienen nombres colocados fuera de los iconos. En esta tesis se utilizan iconos definidos por el usuario para el código fuente (componentes M2.js, eva.css, aevcap_x.htm) y objetos gráficos (componentes de extensión.gif).

Un elemento estructural (desde el punto de vista de UML) que salta a la vista en la Figura 4.4, es la interfaz (representada con un círculo), con ella se intenta destacar el efecto, no la implementación, especificar un servicio que ofrece cierta clase u objeto, en este caso el objeto window de la jerarquía superior de JavaScript, a través de su método **alert()**:

```
<head>
<link rel = stylesheet href = "../intro/eva.css" type = "text/css">
<script src= "M2.js">
</script>
</head>
<body background = "img1/fon1.gif">
```

Cuadro 4.2. Fragmento del documento HTML, indicando las rutas de acceso al código de la hojas de estilo, del programa JavaScript y el “background” del documento HTML.

Los documentos con la extensión “.css” contienen las definiciones de estilo que se van a usar, como tipo de letra, color, márgenes, alineación, propiedades de tablas; permitiendo con ello modificar rápidamente las características del documento HTML. Las definiciones de estilo se importan desde los documentos HTML con la etiqueta <link>.

Las definiciones de hojas de estilo pueden ser conceptualizadas como clases que heredan propiedades a los objetos HTML [Bobadilla 2000].

En el Cuadro 4.3, se muestra el código de la hoja de estilo aplicable a la página aevcap_x.htm:

```
body{ font-family: Arial, Garamond, Verdana; font-size: 10pt;
margin-left: 1cm; margin-right: 1cm;
margin-top: 0.5cm; margin-bottom: 1cm;
text-align: justify}
```

Cuadro 4.3. Código de la hoja de estilo

En el Cuadro 4.4, el código permite incorporar imágenes que son “rótulos” o “etiquetas” que describen la página, una vez desplegada:

```
<div align = “center”>
<img src = "img1/tit1.gif" />

<img src = "img1/aeval.gif" />
</div>
```

Cuadro 4.4 Código de objetos

La Figura 4.4, también pone de manifiesto una interfaz, concretamente una ventana generada con el método **alert()** del objeto **window** de JavaScript, la cual manda el mensaje de aciertos y errores.

La **segunda versión de la autoevaluación** automatizada el programa “modificado”, se describe en términos de los componentes mostrados en la Figura 4.5:

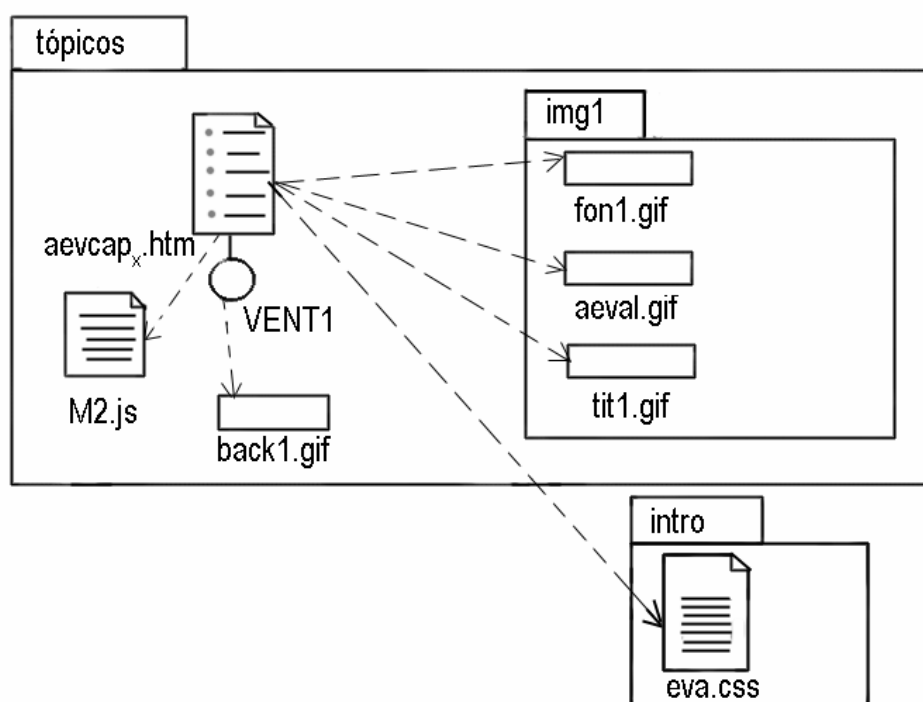


Figura 4. 5 Paquetes y componentes del sistema de autoevaluación modificado.

Note que la diferencia de la representación de la Figura 4.4 y la Figura 4.5 reside en la imagen (back1.gif) que es el “background” de la página Web desplegada por la ventana y generada por la activación de la función Estadist2(), un programa JavaScript. En este caso la Interfaz (círculo en la Figura 4.5), consiste en una ventana (llamada **VENT1**), diferente a la ventana activa; cuenta con barra de corrimiento (scroll) y es resultado de la aplicación del método **open()** del objeto **window** de JavaScript. En la Figura 4.6, se describen los objetos que integran el componente aevcap_x.

La vista de la arquitectura del programa dada por la Figura 4.6, no muestra la diferencia esencial de los programas “clásico” y “modificado”,

pues el código de las funciones que permiten resultados diferentes en cada caso se halla en el mismo archivo M2.js. Las diferencias realmente se aprecian al describir las dependencias.

Una observación importante que llevó a modificar el diagrama de paquetes de la Figura 4.5, fue al realizarse pruebas sobre diferentes plataformas, pues se halló que en la PC-IBM pentium IV con Windows 98, con navegador IExplorer, lo mismo que un equipo Macintosh con sistema operativo Mac Os9 e Internet Explorer, el método open() del objeto window de JavaScript, creaba una página Web virtual; es decir, físicamente inexistente.

Sin embargo, al ejecutar el programa "modificado" en un equipo con Xp, no se generaba la ventana de respuestas, pues se requiere de un archivo de extensión .htm, físicamente existente, de ahí que al diagrama de la Figura 4.6, se le añadiera el componente **reval.htm**:

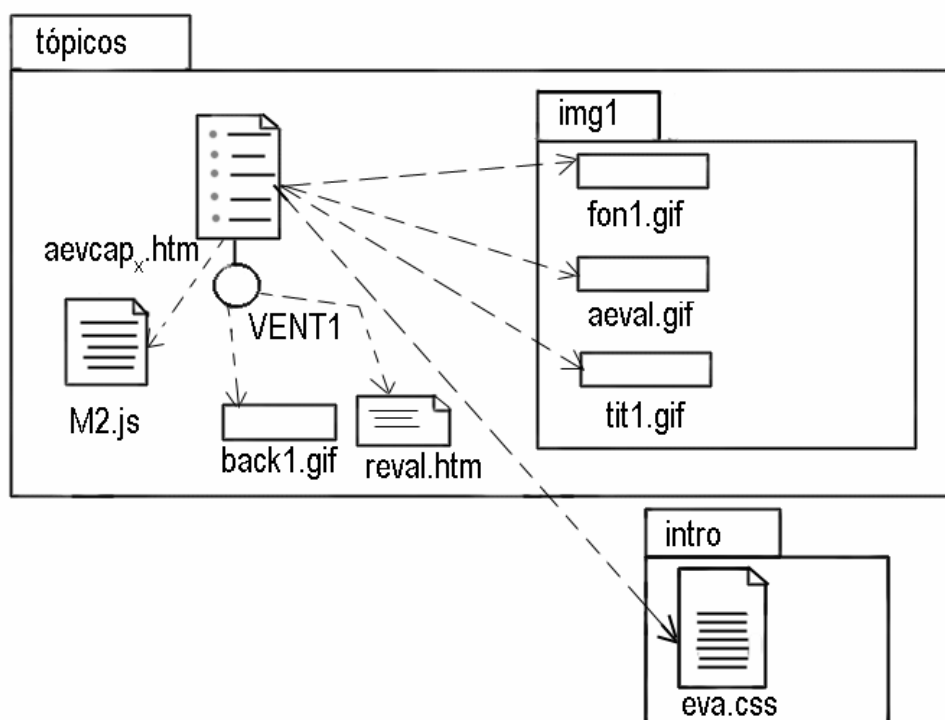


Figura 4.6 Diagrama de componentes (completo) del programa de autoevaluación modificado.

4.5 Dependencias del componente aevcap_x.htm

Ya se ha señalado que aevcap_x.htm representa a las páginas Web con los componentes responsables del despliegue, corrección y muestra de resultados de las pruebas pedagógicas. En el código fuente de las páginas Web destacan las etiquetas <form>, código en transición a XHTML y código JavaScript de las funciones Estadist(), Estadist2(), Reset() y Reset2().

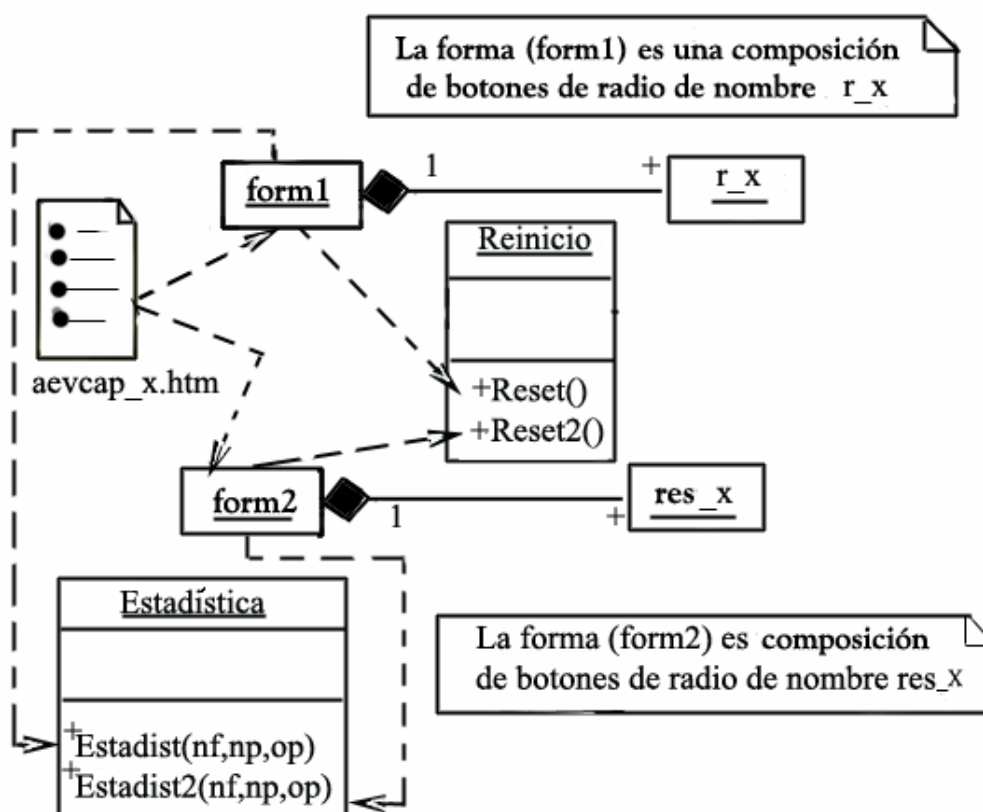


Figura 4.7 Objetos del componente aevcap_x.htm

La representación anterior (Figura 4.7), corresponde a una descripción de los objetos del componente de aevcap_x.htm: Los objetos <form> que en el código de la página corresponde a <form name = "form1"> y <form name = "form2">, son mostrados como una composición de las entradas <input type = "radio" name = `r_x`> para "form1", e <input type= "radio" name = `res_x`> para "form2". En los nombres de las entradas, `x` se sustituye por 1, 2, 3....., de acuerdo al número de opciones por pregunta.

La relación entre las formas (etiquetas <form>) y entradas (etiquetas <input>), según UML es una composición (indicado con el rombo negro), de uno a muchos. La composición es una forma de asociación de agregación con fuerte sentido de posesión y tiempo de vida coincidente con las partes del conjunto [Rumbaugh, et al. 2000].

Los objetos formas <form1> y <form2> hacen uso de dos objetos o funciones de JavaScript, cuyo código fuente se halla en el componente M2.js. Las funciones de los objetos “Estadística” y “Reinicio”, son activadas por evento, pulsando botones de la forma. En <form1> se activan Estadist() y Reset(); en cambio, desde <form2> se activan Estadist2() y Reset2().

La primera forma <form name = “form1”> corresponde al caso de preguntas de falso y verdadero, la segunda forma <form name = “form2”> a la de opción múltiple, corresponde a una versión “más inteligente”, pues es capaz de aportar al usuario una información mayor que el número de aciertos y la información de las preguntas de respuesta incorrecta.

El Cuadro 4.5, contiene el código HTML (XHTML) para el examen de falso/verdadero, mostrando el mecanismo de activación de la función JavaScript Estadist(), se observará que toda letra acentuada se ha sustituido por un nemónico de un conjunto de códigos subconjunto del estándar "ISO/IEC 10646-1:1993" llamado UCS-2 que usa 2 bytes, a diferencia de unicode que usa 4 bytes. El uso de códigos de este tipo, se hace obligatorio en XHTML y en documentos XML; por ejemplo á, debe reemplazarse por "´" (Código á), ó por "ó", etc.; también debe utilizarse el código o sus nemónicos, con las letras ñ, letras con diéresis, inicio de interrogación y admiración.

Cuadro 4.5 Fragmento del código del examen de falso/verdadero (Inicio)

```
<p><b>Conteste con Verdadero o Falso</b></p>

<form name="form1">

<ol>
<li><p>Una t&acute;tica de recursos humanos es una
pol&iacute;tica o un programa concreto que ayuda a avanzar hacia el
objetivo estrat&eacute;gico de la empresa.</p>
<input type="radio" name="r1" value="1"> VERDADERO<br />
<input type="radio" name="r1" value="0"> FALSO <br /><br />
```

```

<li><p>Una de las paradojas del potencial de desarrollo encarnado en
el conocimiento científico y tecnológico aplicado a los
sistemas productivos, ha sido el de sus impactos negativos en el
empleo.</p>
<input type="radio" name="r2" value="1"> VERDADERO<br />
<input type="radio" name="r2" value="0"> FALSO<br /><br />

//*****Aquí van más entradas del objeto <form>

//*****Sólo se muestran dos preguntas

</ol>

<!-- *****Funciones JavaScript ***** -->

<input type="button" value="EVALUAR", onClick="Estadist(0, 20, 2)">
<input type="reset" value="REINTENTAR", onClick="RESET(0, 20)">
<!-- ***** -->

</form>

```

Cuadro 4.5 Fragmento del código del examen de falso/verdadero (Final)

El Cuadro 4.6, contiene parte del código HTML para el examen de opción múltiple, mostrando el mecanismo de activación de la función JavaScript Estadist2():

Cuadro 4.6. Código HTML de las entradas de la forma con las opciones de preguntas (Inicio)

```

<b>SELECCION MÚLTIPLE.</b>
<p> Marque la opción que considere correcta.</p>

<form name="form2">

<ol>
<li><p>La utilización deliberada por parte de la empresa de los
recursos humanos para lograr o mantener una ventaja sobre sus
competidores en el mercado es:</p>

<input type="radio" name="res1" value="1">una estrategia de
recursos humanos<br/>

```

```

<input type="radio" name ="res1" value="0">un programa de recursos
humanos<br/>

<input type="radio" name ="res1" value="0"><p>una t&acute;tica de
recursos humanos<br /><br />

<li><p>¿Cu&acute;l de los siguientes desaf&iacute;os se considera un
desaf&iacute;o del entorno?</p>

<input type="radio" name ="res2" value="0">Reestructuraci&oacute;n
organizativa<br />

<input type="radio" name ="res2" value="1">Evoluci&oacute;n del
trabajo y del papel de la familia<br />

<input type="radio" name ="res2" value="0">Empowerment<br /><br
/>

//*****Aquí deben ir más entradas de la formas

//***** Sólo se muestran dos preguntas

</ol>

<!-- ***** Funciones JavaScript ***** -->

<input type="button" value ="EVALUAR", onClick="Estadist2(1,30,3)">
<input type ="reset" value ="REINTENTAR", onClick="RESET2(1,30)">
<!-- ***** -->
</form>

```

Cuadro 4.6. Código HTML de las entradas de la forma con las opciones de preguntas (Final)

En JavaScript la implementación de funciones aisladas es una parte esencial de la programación. Estos bloques de código (subprogramas) se activan por evento. En lenguajes orientados a objetos como Smalltalk, sólo pueden declararse las operaciones como métodos, ya que el lenguaje no permite declarar procedimientos o funciones separadas de ninguna clase.

Más en cambio, en lenguajes como Object Pascal, C++, CLOS y Ada permiten al desarrollador escribir operaciones como subprogramas libres,

(equivalentes a las funciones no miembro de C++); se trata de procedimientos o funciones que sirven como operaciones no primitivas sobre un objeto u objetos de la misma o distintas clases.

Como los subprogramas libres se agrupan típicamente según las clases sobre los que se ha construido; por tanto, Grady Booch [Booch 1994], llama a tales colecciones de subprogramas libres **utilidades de clases**.

Estas funciones libres significaron un problema en la modelación de los objetos y las relaciones del componente **M2.js**.

De acuerdo a Grady Booch, en C++ y Smalltalk, es una práctica habitual recoger todos los subprogramas libres relacionados lógicamente y declararlos como parte de una clase que no tiene estado. El nombre de la “utilidad de clase” no tiene más significado que el de dar nombre adecuado a un grupo de funciones no miembro.

JavaScript no se basa en clases, sino en objetos, por ello las antiguas “utilidades de clase” de Grady Booch, serán aquí objetos. Las dos funciones en JavaScript del componente M2.js, **Estadist()** y **Estadist2()**, así como **RESET()** y **RESET2()** fueron consideradas como “métodos” de los objetos “Estadística” y “Reinicio” (físicamente inexistente) en la Figura 4.7, considerando que sus nombres, son simplemente una ayuda para agrupar a las funciones lógicamente relacionadas.

Esta forma de reunir a funciones o subprogramas, es útil para simplificar algunos diagramas, pero en otros, no ayuda a establecer una correcta descripción de las relaciones de clase, por ello en el diagrama que describe los objetos constituyentes, se consideró otra representación (Figura 4.8).

Las funciones u operaciones libres deben ser tratadas desde el punto de vista de una concepción orientada a objetos, como tales entidades de programación. Se aprovecha la libertad de UML de proponer representación de un elemento del modelo, definiéndolo en el propio modelo, utilizando los diagramas estándar aunque con significado y uso diferente [Rumbaugh, et al. 2000]. Tales funciones pueden ser descritas como clases parametrizadas [Fower, Kendall 1999], concepto que proviene de C++ (moldes o plantillas), utilizado principalmente para los casos en que se puede cambiar **el tipo** del parámetro como en `template<class T> class Vector`, pero también se pueden tener las opciones siguientes [Web14, 2006]:

```
template<class T>
template <class T, class U>
template<class T, int N>
template <class T = char>
```

```
template <int T, func(int)>
```

Las funciones “libres” de JavaScript las consideraremos como clases parametrizadas, lo cual permite destacar los parámetros que reciben, así se muestra en la Figura 4.8, al modelar el componente M2.js.

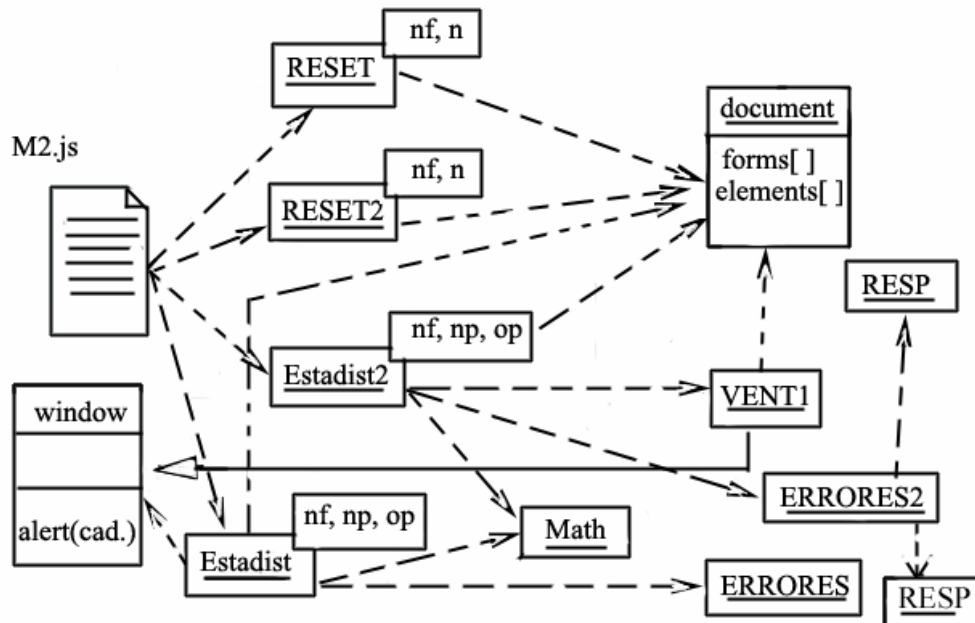


Figura 4.8 Objetos del componente M2.js

El objeto Estadist() (función JavaScript), se activa por evento, hace uso de los siguientes objetos internos:

- i) **ERRORES[]**, que guarda el número de pregunta cuya respuesta fue errónea.
- ii) El objeto del navegador **document**, cuyas propiedades **forms[]** y **elements[]**, permitirán seleccionar el objeto HTML **<form>** correspondiente a la opción de falso - verdadero, y analizar cada opción seleccionada.
- iii) También se hace uso de un objeto ventana, resultado de activar el método alert() del objeto window.
- iv) **Estadist2()** utiliza también algunos objetos empleados por la función **Estadist()**, pero se añade el objeto **ERRORES2[]** que a su vez usa el objeto **RESP[]** el cual contiene los textos a desplegarse de acuerdo a la opción incorrecta seleccionada. Note que **Estadist2()** hace uso del objeto **VENT1**, una ventana definida

por el programador, la cual desplegará los resultados, los textos de información al alumno; la relación de herencia de **VENT1** con el objeto window se indica con línea gruesa y el triángulo blanco (Figura 4.8).

- v) Tanto Estadist() como Estadist2() usan un objeto Math.

Algunos objetos comunes a las funciones **Estadist()** y **Estadist2()**, así como aquellos utilizados únicamente por la última función (**ERRORES2[], RESP[], texto2**), pero que pudieran ser empleados en posteriores modificaciones del programa, se anotaron en el script a modo de variables globales.

```
var Buenas = 0;
var Malas = 0;
var ERRORES = new Array();
var ERRORES2 = new Array();
var RESP = new Array(30);
var texto="";
var texto2 = "";
```

Cuadro 4.7. Objetos globales (públicos) en el script.

4.6 Diagramas de actividad

4.6.1 Diagramas de actividad del proceso de autoevaluación

En la Figura 4.9, se muestra el diagrama de actividades propuestas aplicando las técnicas de UML, del proceso de despliegue de los exámenes y la activación de las funciones de corrección automatizada. Un diagrama de actividades representa un grafo de actividades, que es un caso especial de una máquina de estados en el cual todo o la mayoría de los estados son estados de actividad o estados de acción y en el que la mayoría de las transiciones son disparadas por la finalización de una actividad en los estados de origen. Un grafo de actividad muestra un procedimiento o flujo de trabajo [Rumbaugh, et al. 2006].

El uso típico de un estado de actividad es modelar un paso en la ejecución de un procedimiento.

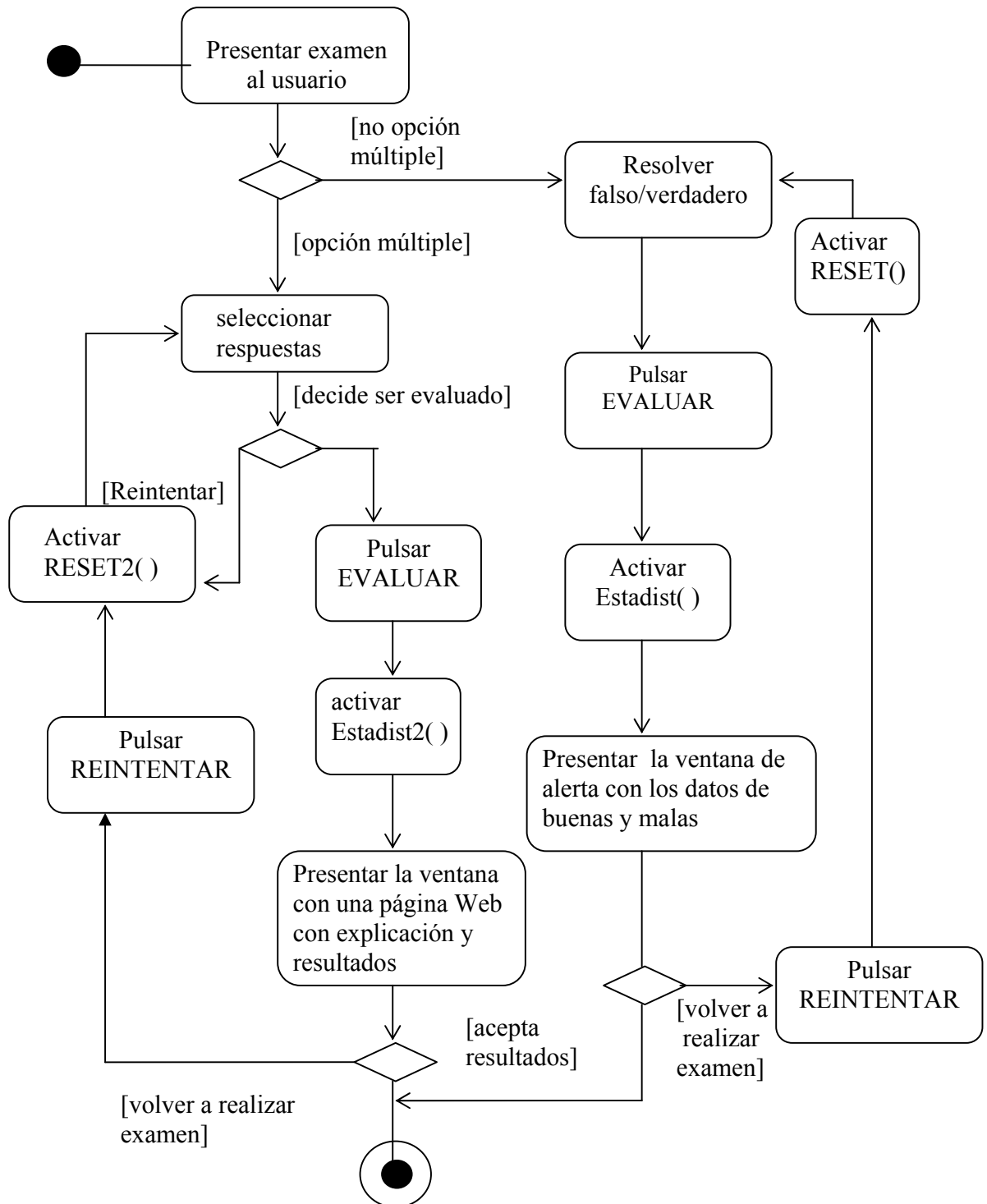


Figura 4.9 Diagrama de actividad para el despliegue el examen

4.6.2 Diagrama de actividad de la función Estadist()

La actividad de la función Estadist() consiste en hacer un recorrido por la forma <form> del documento HTML que le corresponde analizar de acuerdo al parámetro nf), en cada pregunta detecta sus entradas (el número de ellas las da el parámetro op); si la entrada (<input type = "radio">) fue seleccionada, se analiza su propiedad value, si corresponde a una respuesta correcta, incrementa la variable BUENAS, de lo contrario incrementa malas y guarda en el arreglo ERRORES[] el número de pregunta dado por la expresión $\text{Math.round}(i/op) + 1$, donde i lleva la cuenta de las entradas de la forma recorridas.

Puesto que una entrada fue seleccionada se “brinca” a la siguiente pregunta y vuelve a analizar sus entradas, en caso de que no se detecte selección de la opción, se analiza la siguiente opción dentro de una misma pregunta, hasta hallar una opción seleccionada y proceder como se ha mencionado. Una vez recorridas todas las preguntas si el número de “buenas” es menor al número máximo de reactivos (en nuestro caso 10), lo cual indica la existencia de respuestas malas, se envía un mensaje que incluye el número de reactivo errado, de lo contrario se felicita al alumno.

La función termina reestableciendo los valores de Buenas = 0 y Malas = 0, para que puedan volver a ser reutilizados por esta función o por la que actuará en el examen de opción múltiple.

En la Figura 4.10, se muestra el diagrama de actividades de esta función, para desarrollarlo se consideró la forma en que Deitel y Deitel [Deitel 2004] tratan el asunto de los algoritmos en términos de diagramas de actividad.

Según dichos autores:

“un diagrama de actividad modela el flujo de trabajo (también conocido como actividad) de una parte de un sistema de software. Dichos flujos de trabajo pueden incluir una porción de un algoritmo [...]. Al igual que el pseudocódigo, los diagramas de actividad ayudan a los programadores a desarrollar y representar algoritmos[...]. Los diagramas de actividad muestran claramente cómo operan las estructuras de control” [Deitel 2004].

Deitel y Deitel dedican varias páginas a mostrar la “equivalencia” en términos de UML de estructuras de control como if-else, while, for, do-while. Y en ese sentido hemos retomado algunas de sus ideas.

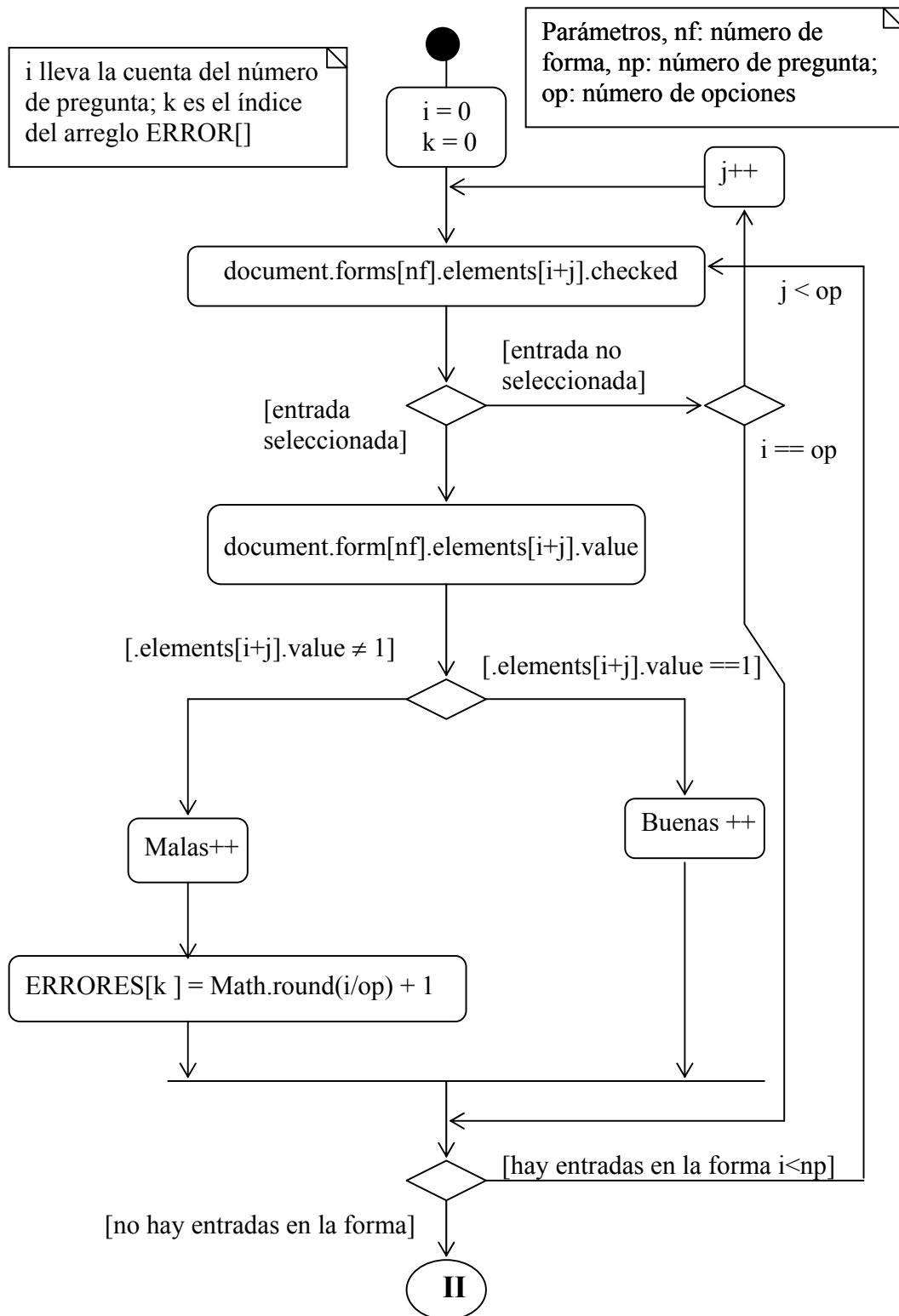
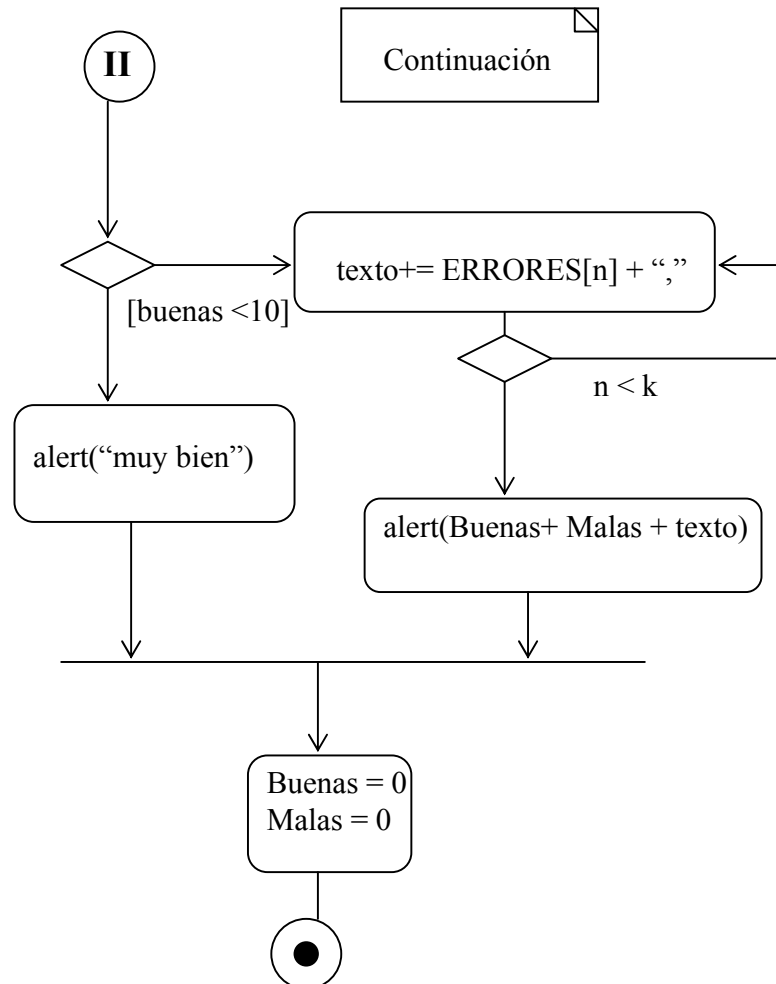


Figura 4.10 Primera parte del diagrama de actividad de la función Estadist()**Figura 4.11 Segunda parte del diagrama de actividad de la función Estadist()**

El objeto **Math**; mediante su método **round()**, ayuda a determinar el número de pregunta errada, dato que es guardado en el objeto **ERROR[]**. Los números de pregunta errada conforman un texto que se pasa junto con el número de buenas y malas al método **alert()** del objeto **window** que desplegará una ventana con mensaje.

El código de la función Estadist(), se muestra en el Cuadro 4.8:

```

function Estadist(nf,np,op)
{
    //nf número de formulario
    //np número de pregunta
    //número de opciones del reactivo o pregunta
    i= 0;
    k=0;
    do{
        //recorre las opciones de la i-ésima pregunta
        //y prueba la seleccionada
        for(j=0;j<op;j++)
            if(document.forms[nf].elements[i+j].checked)
            {
                if(document.forms[nf].elements[i+j].value ==1)
                {
                    Buenas++;
                    //Si es buena interrumpe el recorrido
                    break;
                } else
                {
                    Malas++;
                    ERRORES[k]= Math.round(i/op)+1;
                    k++;
                    //se guarda el número de pregunta errada
                    // y se interrumpe el recorrido
                    break;
                }
            }
        i=i+op;
    }while(i<np);
    if (Buenas <10)
    {
        texto ="Error en la pregunta \n";
        // k es el número de errores
        // el número de reactivo errado conforman un texto
        for(n=0; n<k; n++)
            texto+= ERRORES[n]+",";
        alert("BUENAS:"+Buenas+"\n"+"Malas:"+Malas+"\n"+texto +" \n");}
    else
        alert("MUY BIEN");
        Buenas =0;
        Malas =0;
    }
}

```

Cuadro 4.8. Programa en lenguaje JavaScript para la autoevaluación automatizada "clásico".

4.6.3 Diagrama de actividad de la función Estadist2()

Respecto al diagrama de actividad de la función Estadist2() mostrado en las Figuras 4.10 y 4.11, hay tres diferencias esenciales en Estadist2():

1) Dentro del ciclo donde el objeto interno Math obtiene con su método round(), el número de reactivo erróneo y lo guarda en ERRORES[i+j], entra en acción el arreglo RESP[i+j] de respuestas, que de acuerdo al elemento o entrada de la forma (con nombre "form2" y cuyo número de programa o identificación JavaScript lo da el parámetro nf), seleccionado y calculado mediante la suma de índices i + j, pasa su contenido al objeto ERRORES2[k].

2) En la segunda parte del diagrama de la función Estadist2() (Figura 4.12 y 4.13), se muestra que el contenido de ERRORES2 se concatena con las cadenas text2 la cual se ha inicializado con un mensaje, al igual que la cadena text1. Tal como se mostró en el Cuadro 4.7 son cadenas (objetos) de carácter global.

3) Se ha definido un objeto ventana denominada VENT1, uno de cuyos parámetros es el nombre del archivo "reval.htm". VENT1 hace uso del objeto documento de la jerarquía superior de objetos del navegador, el cual a su vez activa su método writeln, cuyos parámetros son cadenas concatenadas, incluyendo text1 y text2. La cadena resultante define el contenido de una página Web, su archivo es "reval.htm", cuyo despliegue muestra como una ventana independiente al activarse Estadist2(), al pulsar "EVALUAR" desde la página del examen.

El código de la función JavaScript Estadist2(), se muestra en el Cuadro 4.8:

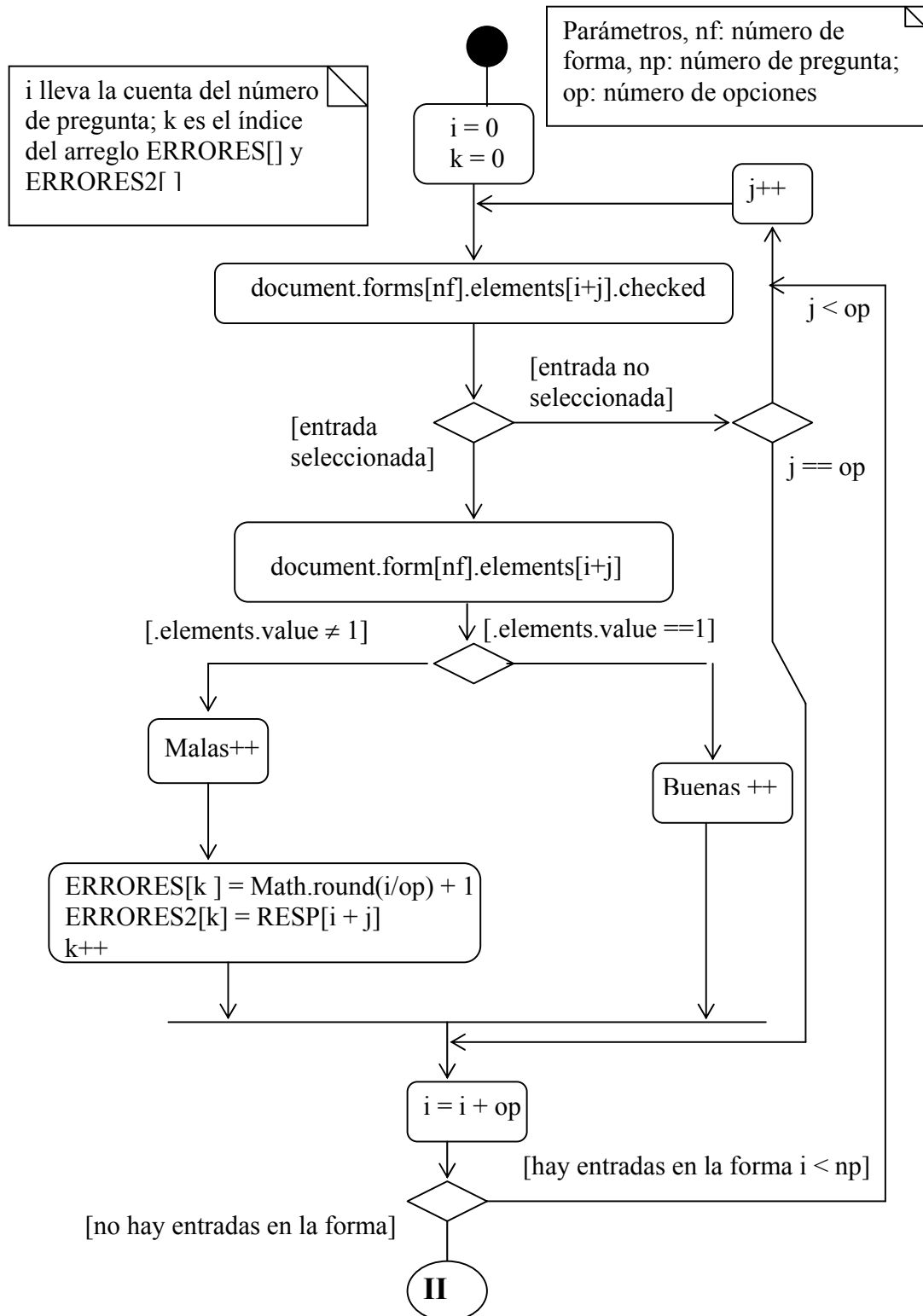


Figura 4.12 Diagrama de actividad (UML), describiendo el comportamiento de la función Estadist2() (Primera parte)

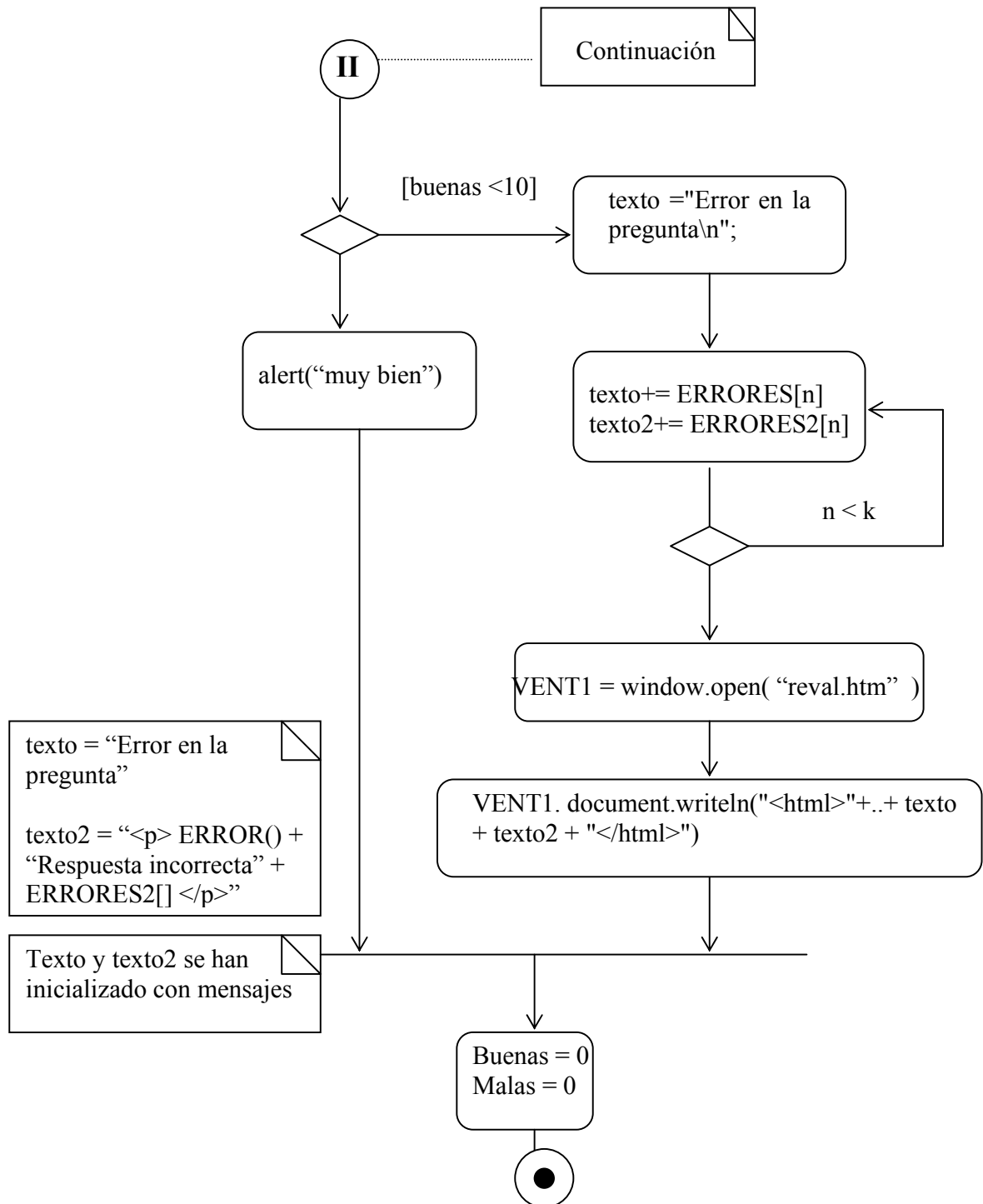


Figura 4.13 Diagrama de actividad (UML), describiendo el comportamiento de la función Estadist2() (Segunda parte)

Cuadro. 4.9 Código JavaScript de la función Estadist2() (Inicio)

```
function Estadist2(nf,np,op)
{
    //nf número de forma
    //np número de pregunta
    //op número de opción

    RESP[0] = " \n"

    RESP[1] = "Usted marc\u00f3 como correcta la respuesta b). Incorrecta. Un
    programa de recursos humanos se refiere a un medio de ayuda a lograr las
    metas estrat\u00e9gicas de la organizaci\u00f3n. La respuesta correcta es la
    a).\n"

    RESP[2] = "Usted marc\u00f3 como correcta la respuesta c). Incorrecta. Una
    t\u00e9cnica de recursos humanos es una pol\u00edtica particular de
    recursos humanos o un programa que ayude a la obtenci\u00f3n de las
    metas estrat\u00e9gicas de la organizaci\u00f3n. La respuesta correcta es la
    a).\n"

    //***** Sólo se han anotado tres opciones (véase el anexo )

    //*****

    i= 0;
    k=0;
    do{
        //recorre las opciones de la i-ésima pregunta y prueba la
        //seleccionada
        for(j=0;j<op;j++)
            if(document.forms[nf].elements[i+j].checked)
            {
                if(document.forms[nf].elements[i+j].value ==1)
                {
                    Buenas++;
                    //Si es buena interrumpe el recorrido
                    break;
                }
            }
        else
    }
```

```

{
    Malas++;
        ERRORES[k]= Math.round(i/op)+1;
        ERRORES2[k] = RESP[i+j];
        k++;
        //se guarda el número de pregunta errada, de acurdo
        // a la pregunta se guarda la respuesta al usuario en ERRORES2[ ]
        // y se interrumpe el recorrido
        break;
    }
}

i=i+op;
}while(i<np);

if (Buenas <10)
{
    texto ="Error en la pregunta\n";
    //k es el número de errores
    for(n=0; n<k; n++)
    {
        texto += ERRORES[n]+",";
        texto2 += "<p>" + "P" + ERRORES[n] + "." + " " + "<font color =
'red'><b>RESPUESTA INCORRECTA.</b></font>" + " " + ERRORES2[n] + "
"</p> <hr /> width='400'>";
    }
    //se utiliza fundamentalmente el objeto document

    VENT1=window.open("reval.htm","VENT1","width=500,height=200,scrollbars=
1");
    VENT1.document.writeln( "<html> <head> </head> <body background =
'back1.gif'>");
    VENT1.document.writeln("<p><b>Buenas: </b>"+Buenas+"<br />"+<b>Malas:
</b>"+ "&#160;" +Malas+"<br />"+ "<font color = 'blue'><b>"+texto +
"</b></font>" + "</p>"+ texto2);
    VENT1.document.writeln("</body></html>");
    VENT1.document.close();
}
else
    //no hay errores
    alert("MUY BIEN.\n Respuestas Correctas");
    Buenas =0;
    Malas =0;
}

```

Cuadro. 4.9 Código JavaScript de la función Estadist2() (Final)

En el Cuadro 4.9, se aprecia que el objeto interno JavaScript **RESP[]**, un arreglo, se inicializa con cadenas (strings), que contienen caracteres unicode, un conjunto de caracteres requerido por los modernos estándares como XML, Java, ECMA Script (JavaScript), la forma oficial de implementar la norma ISO/IEC 10646; se halló que de no incluirse el despliegue no dará resultados adecuados si se usa código HTML o nemónicos como en el cuadro 4. 5 [Web18]. Las cadenas son las respuestas que dará el programa de acuerdo a la selección realizada por el usuario del programa, cada pregunta tiene un juego de posibles respuestas.

Así del Cuadro 4.9, podemos ver que la pregunta 1 tiene tres opciones. Si la selección es correcta la cadena contiene simplemente el caracter de salto de línea, cuando la selección es errónea, el programa guarda la respuesta que deberá dar al usuario, en el objeto interno **ERRORES2[]**; en tanto que el objeto interno **ERRORES[]**, guarda el número de preguntas erradas.

El contenido de **ERRORES[]**, mediante una iteración (aplicación de un “for”) se concatena y forma la cadena “texto” (**texto += ERRORES[]**), dentro del mismo ciclo se conforma la cadena texto2 con código html e incorporando el contenido de **ERRORES2[]**, indicando un párrafo diferente para cada componente de este arreglo.

Finalmente la cadena text2, se pasará como parámetro a la ventana definida con **window.open()**, que es la ventana llamada **VENT1**, representada en la Figura 4.6, como una interfaz.

El Cuadro 4.10, muestra el código JavaScript de las funciones que restablecen las condiciones para la autoevaluación:

```
function RESET(nf, n)
{
    for(j=0;j<n;j++)
        document.forms[nf].elements[j].checked=false;
}

function RESET2(nf, n)
{
    VENT1.close();
    for(j=0;j<n;j++)
        document.forms[nf].elements[j].checked=false;
}
```

Cuadro 4.10. Funciones que se activan al pulsar el botón "REINTENTAR"

Las funciones que restablecen las condiciones iniciales de los objetos <form>, asignan el valor "false", en la propiedad "checked" de cada entrada de la forma, dado por el arreglo **elements[]**, en la función **RESET2[]**, se asegura además el cierre de la ventana abierta con `window.open()`, llamada aquí **VENT1**.

Resumen del capítulo

Para el desarrollo de las páginas Web con las pruebas pedagógicas, las funciones activadas por evento se ha utilizado código HTML que se adecuó para migrar a XHTML, JavaScript y código de hojas de estilo.

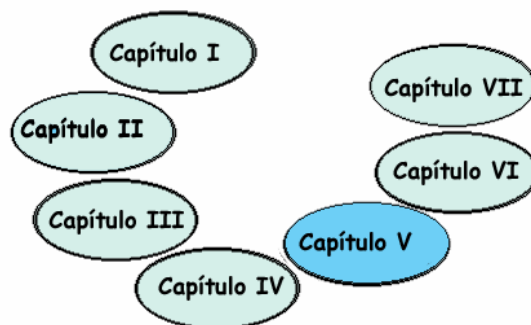
JavaScript es un lenguaje basado en prototipos no en clases, presenta dificultades en su aplicación por los diferentes DOM (sostenidos por Explorer y Netscape). Por lo cual el software desarrollado ha sido probado en cuanto su portabilidad.

Se ha descrito mediante UML los casos de uso, así como la vista física mostrando los paquetes y los componentes, así como las relaciones entre los objetos y sus relaciones. Los diagramas de actividad muestran tanto el funcionamiento general de los programas de autoevaluación, como el procesamiento de las funciones **Estadist()** y **Estadist2()**.

A continuación se procederá a describir la arquitectura del programa "Generador Aleatorio de Exámenes Matemáticos".

CAPÍTULO V.-

Arquitectura del Programa “Generador Aleatorio de Exámenes Matemáticos” (GAEM)



Contenido del Capítulo

Se describe la arquitectura del programa de cómputo, llamado en esta tesis “Generador Aleatorio de Exámenes matemáticos” (**GAEM**), características técnicas como lenguajes utilizados y plataforma en que se ha probado. Se discute el alcance y ventajas del tipo de programa desarrollado.

Se muestran los diagramas de casos de uso y la vista física que incluye la descripción de componentes, objetos y sus dependencias. Se describe el diagrama de actividad de las funciones escritas en JavaScript: `Azar( )` y `Examen()`, utilizadas respectivamente para la aleatorización de la selección de reactivos y el despliegue del examen.

5.1 El programa “Generador Aleatorio de Exámenes Matemáticos” (GAEM)

Una de las principales tareas en la enseñanza “en línea”, es hallar formas de evaluación acordes con el contexto de Internet. Como dice el M. en C. Efraín Martínez Ortiz impulsor de la modalidad virtual de la Maestría en Ingeniería de Sistemas, donde se aplicaron los programas aquí propuestos, el “talón de Aquiles” del sistema de aprendizaje virtual, es la evaluación.

En un gran número de casos se han utilizado sistemas de evaluación automatizada consistente en preguntas de falso o verdadero u opción múltiple, que a los profesores de matemáticas por lo regular no convencen, a pesar de que se han utilizado ya, en evaluaciones a nivel institucional, como en las del Consejo del Sistema Nacional de Evaluación Tecnológica (COSNET).

Algunos matemáticos docentes afirman que la evaluación del tipo opción múltiple mide el conocimiento de manera superficial, además de que se presta al acierto de forma aleatoria. La matemática es razonamiento, el matemático docente busca medir la capacidad de razonar y no solamente el resultado.

El matemático docente pregunta al alumno sometido a la evaluación ¿De dónde sale el resultado? Califica o da más valor al procedimiento que al resultado. La matemática expresa, argumenta lógicamente; no es saber calcular, es saber razonar. Es realizar inferencias a partir de los axiomas o principios fundamentales.

A las objeciones respecto a la naturaleza del reactivo que conforma la prueba pedagógica, se suman las dudas que se tienen respecto a la objetividad de una evaluación a distancia; por ello, se consideró más apropiado personalizar el examen; es decir a cada alumno que solicite evaluación, se le asignará un examen diferente. La meta anterior se puede lograr mediante la selección aleatoria de reactivos, para lo cual se propuso un **Generador Aleatorio de Exámenes Matemáticos (GAEM)**.

Los exámenes se conformarán a partir de un banco de reactivos lo suficientemente amplia. Si por alguna circunstancia, en algún momento esto no es posible de realizar, se deben proponer exámenes con el menor número posible de preguntas. La última consideración no está muy alejada de la realidad, pues existen cursos (como en Programación Matemática) donde el tiempo invertido en la resolución de un problema es tal que no es posible plantear a los alumnos 10 preguntas.

Asumamos que un examen consta de dos preguntas y suponiendo una base de reactivos de 16 preguntas, el número posible de exámenes que se pueden obtener es

$$\binom{n}{r} = \frac{n!}{(n-r)!r!} = \binom{16}{2} = \frac{16!}{14!2!} = 120$$

Suponiendo que se tienen que resolver 6 exámenes, de acuerdo al teorema fundamental de conteo que dice que si una acción se realiza en p formas, una segunda acción en q formas y así sucesivamente, el número total de formas en que se puede realizar la acción es $p \times q \times r \times s \times t \dots$.

Así, el número total de exámenes posibles de realizar es $(120)^6$

La probabilidad de exámenes repetidos parece pequeña; sin embargo sólo será mayor la certidumbre de ello, si el número de reactivos es grande respecto al número de preguntas del examen.

5.2 Descripción Técnica

5.2.1 Los lenguajes de programación utilizados

El programa GAEM, se ha desarrollado para realizar su tarea en un entorno de Web, el programa mismo abre una ventana que a su vez despliega una página Web la cual muestra el examen.

Por lo anterior el lenguaje básico es HTML; sin embargo al igual que en los programas de autoevaluación “clásico” y “modificado” se ha procedido a realizar ajustes que permitan “migrar” a XHTML.

El lenguaje utilizado para las funciones activadas por evento es JavaScript, seleccionado por su facilidad de programación, rapidez y su interacción natural con los objetos HTML particularmente cuando se utilizan las etiquetas `<form>`

Otro lenguaje que también se ha usado es el de hojas de estilo Cascading Style Sheets nivel 2 (CSS2), el cual permite a los autores y usuarios, asociar un estilo (por ejemplo fonts, espaciado, señales sonoras) a documentos estructurados como HTML y aplicaciones XML. La gran ventaja de este lenguaje es de que permite separar el estilo de presentación del contenido de los documentos, simplificando la creación y mantenimiento de los sitios Web [Web 15, 2007].

5.2.2 Sobre la plataforma utilizada

Debido a las diferencias en cuanto al Modelo de Objetos de Documento (Document Object Model) de JavaScript en Netscape y Microsoft, sólo se ha admitido código portable, probado en estos navegadores y en equipos PC-IBM, así como en equipos Macintosh, bajo Windows, Mac Classic y OSX.

A continuación se resumen los lenguajes y plataformas utilizadas en el desarrollo del programa propuesto

Tabla 5.1 Lenguajes y plataformas utilizadas para ejecutar el programa GAEM

	Descripción de arquitectura	Lenguaje de Script	Otros lenguajes para el desarrollo de la página Web	Navegador de prueba	Sistema Operativo
Programa Generador de Exámenes Matemáticos	UML	JavaScript	XHTML CSS	Firefox, Flock, Explorer, Netscape, Safari	Mac OSX Mac Classic Windows Linux

5.3 La vista de los casos de uso

Los intereses o necesidades del alumno y el profesor (tutor o facilitador) son diferentes, en el programa **GAEM**, el primero desea solicitar un examen y ser atendida esa solicitud; no le interesa si su examen es diferente al recibido por algún otro alumno; en cambio, al profesor además de que el programa presente un examen a solicitud del usuario, le interesa que los exámenes sean distintos para cada alumno y que se muestren las expresiones matemáticas.

Estos comportamientos se reflejan en el diagrama de casos de uso, el cual debe incorporar fundamentalmente, a los actores, relaciones y casos de uso.

La mayoría de las veces los casos de uso se utilizan para el modelado del comportamiento de un elemento, ya sea un sistema completo, un subsistema o una clase. Cuando se modela el comportamiento de estos elementos, es importante centrarse **en lo que hace** el elemento, no en cómo lo hace [Booch, et al.1999].

Los diagramas de casos de uso incluyen fundamentalmente, a los actores, los casos de uso y las relaciones. La Figura 5.1, muestra el diagrama de casos de uso del programa **Generador Aleatorio de Exámenes Matemáticos (GAEM)**.

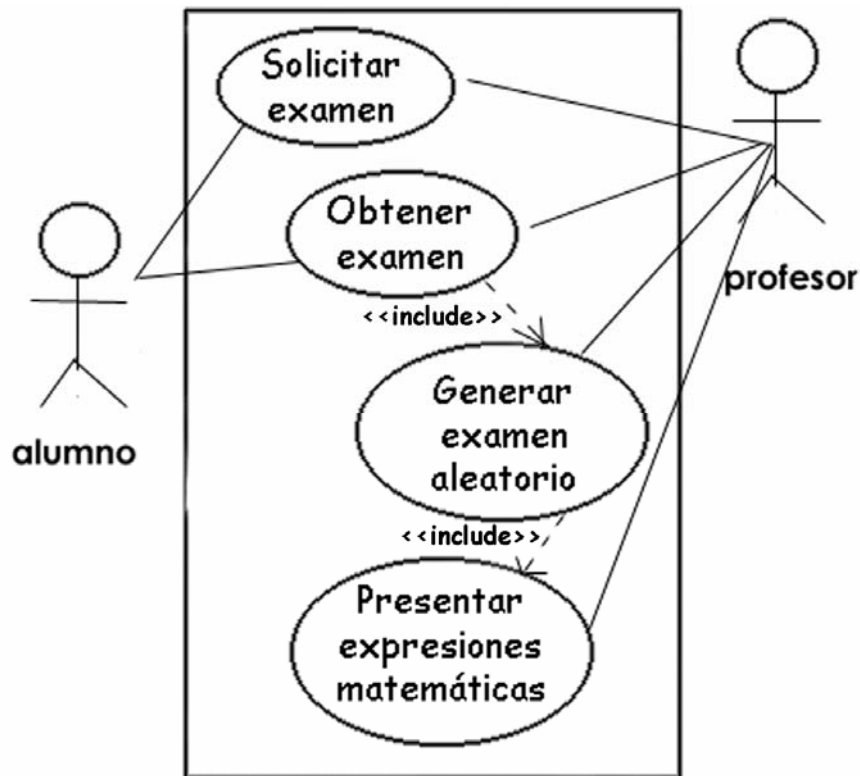


Figura 5.1 Diagrama de caso de uso del GAEM

5.4 La vista física

Para mayor comprensión de la arquitectura y funcionamiento del Generador de Aleatorio de Exámenes Matemáticos (GAEM), se hará una descripción de la vista física como se muestra en la Figura 5.2. En dicha Figura el concepto de paquete de UML se ha hecho coincidir con el físico de carpeta.

El paquete examen contiene la página Web “exams.htm”, la cual establece relaciones o dependencias con los paquetes “tareass”, “code” e “imag”. El paquete “tareass” contiene los paquetes “tarea1”, “tarea2”, etc. Cada paquete “tarea” se ha indicado como tarea_x, el subíndice $x = 1, 2, \dots, n$. Cada uno de esos paquetes contiene a su vez los componentes tar_x.htm que corresponden a un examen específico (tar1.htm, tar2. htm, etc., de ahí el subíndice); en realidad desplegarán el botón que activará el programa **GAEM** y como respuesta presentará el examen en una ventana independiente.

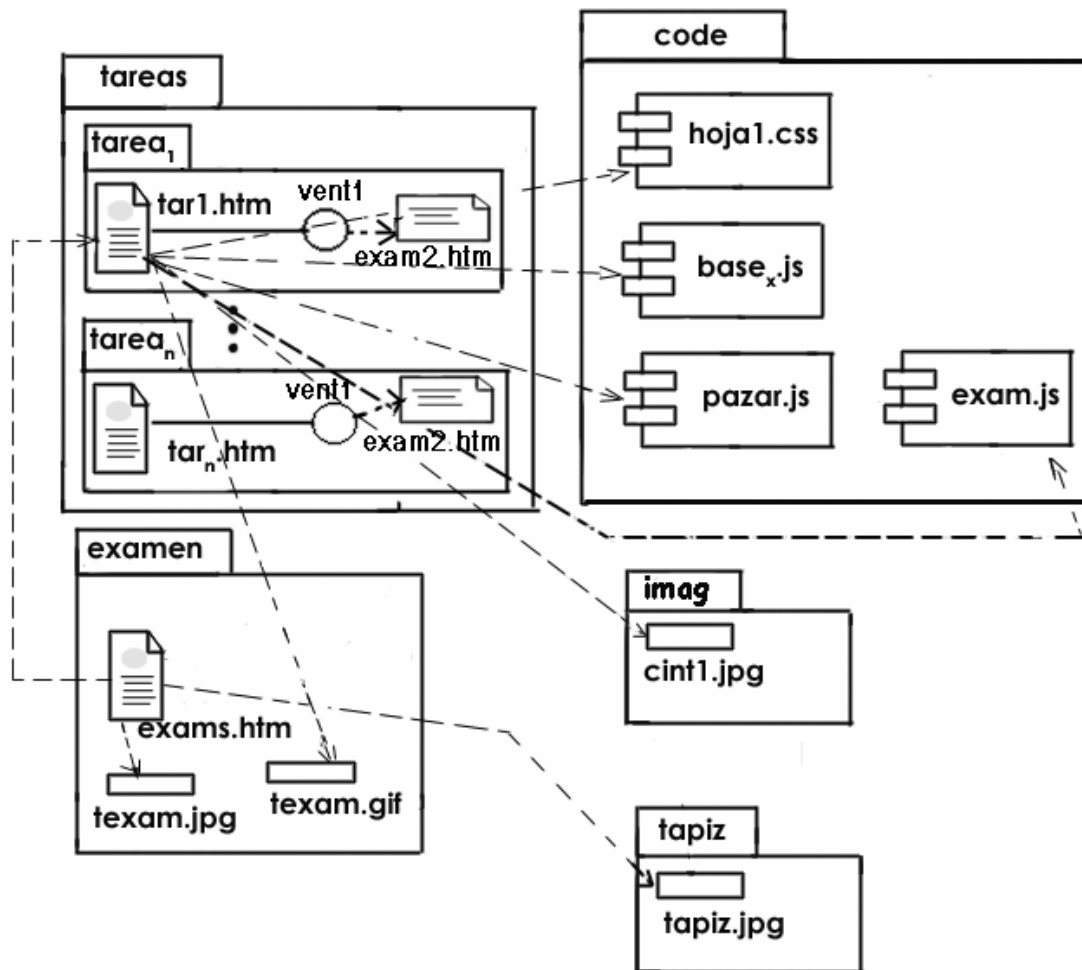


Fig. 5. 2 Vista física del programa
Generador Aleatorio de Exámenes Matemáticos

UML permite sustituir los iconos "estándar", como es el icono "canónico" para los componentes consistente en el rectángulo de dos pestañas, por iconos definidos por el usuario [Booch, et al. 1999]. Tal como se muestra en la Figura 5.2 los componentes que consisten en código fuente como exams.htm, tar_n.htm, pazar.js, base_x.js y los elementos gráficos de extensión .jpg, llevan su nombre fuera del diagrama.

Otro icono del usuario es exam2.htm, corresponde a un archivo de extensión .htm, que debe ser incluido en los sistemas con XP, pues de otra manera, la interfaz que se genera con el método open() del objeto window de la jerarquía superior de objetos de JavaScript, aquí denominada **vent1**, no desplegará el examen obtenido aleatoriamente. Es una situación análoga a la ya reportada en el capítulo IV, para el programa de autoevaluación "modificado", dicho archivo es generado virtualmente en sistemas con windows 95 y Mac OS9.

5.5 Las dependencias en el componente exams.htm

La Figura 5.3, describe los objetos del componente “exams.htm”, objetos tabla “<table>” de HTML (XHTML), las cuales se distinguen en esta tesis con subíndices. Para JavaScript la presencia de varias tablas no es problema pues las distingue por su orden de aparición. Una de las tablas contiene las referencias a los componentes tar_x.htm, páginas Web que activarán la función JavaScript **Examen()**, la encargada propiamente de la actividad del Generador.

Existen algunos objetos "decorativos" como cint1.jpg o tapiz7.jpg; objetos hojas de estilo para conformar el texto en tablas por ello su nombre se indicó mediante el nombre “<style>td”. El objeto tapiz7.jpg es el “background” de la página, le da color y textura.

Un objeto HTML del tipo <tabla> de la página Web “exams.htm”, contiene las referencias a las páginas Web que generan el examen para cada temática particular. Las páginas que activan el programa **GAEM** son representadas como componentes e indicado como tar_x.htm, donde el subíndice x =1, 2, ..., n; es decir, se trata de los archivos fuente tar1.htm, tar2.htm... Se ha indicado la relación de la tabla con los componentes como una relación uno a muchos. La tabla mantiene referencias (anclas <a href>), hacia cada uno de los componentes tar_x.htm.

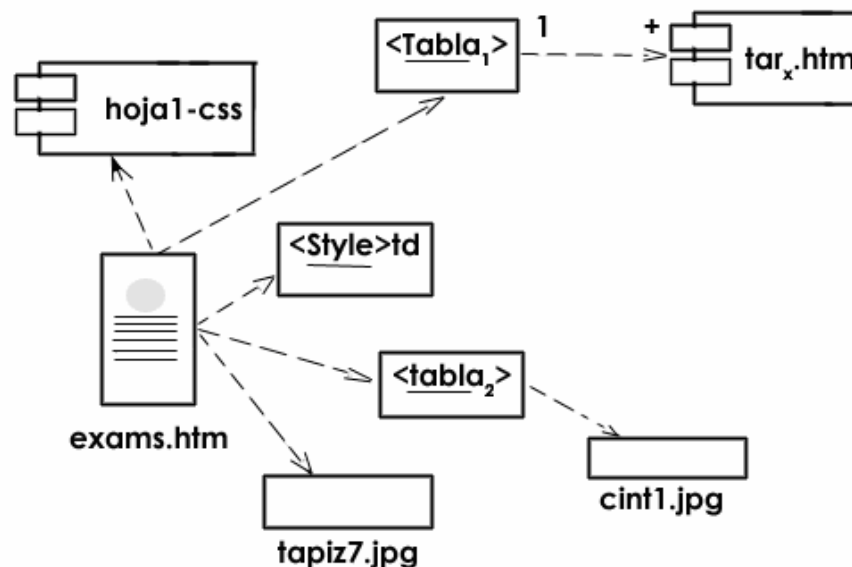


Figura 5.3 Objetos y componentes de la página Web “exams.htm”

El componente “exams.htm” (Figura 5. 3), es la página Web que representa la primera vista del **GAEM** en la cual se enuncian las condiciones para poder realizar el exámenes, las cuales consisten en la realización de ciertas tareas; asimismo se muestran letreros de acceso a cada uno de los exámenes. En un programa más avanzado debería de haber una verificación de que el alumno entregó tareas para poder acceder a un examen.

El Cuadro 5.1, muestra parte del código de la página “exams.htm”, se muestra el vínculo al componente (hoja de estilo) “hoja1.css”, el objeto de estilo entre las etiquetas <style> </style> con las instrucciones para el formato de las celdas; así como el direccionamiento al componente tapiz7.jpg que da el background de la página “exams.htm”.

Cuadro 5.1 Código con las referencias a los componentes hojas de estilo, y al objeto de estilo td{ } y al componente “tapiz7.jpg”.

```
//archivo exams.htm
<!DOCTYPE html PUBLIC “--//W3C//DTD XHTML 1.0      Strict//EN”
“http://www.w3.org/TR/xhtml1/DTD/strict.dtd”>

<html xmlns=”http://www.w3.org/TR/xhtml1”>
<head>
<link rel = stylesheet href = “../code/hoja1.css” type = “text/css”>
<style>

td{ font-family:arial; font-size:11pt;padding-left:10;padding-
right:10;padding-top:10;padding-bottom:10;}
</style>
</head>
<body background = “../tapiz/tapiz7.jpg” link=”black” >
//***** Aquí va el resto del código correspondiente al <body>-

//*****
</body>
</html>
```

En el Cuadro 5.2, se muestra parte de la tabla (archivo “exams.htm”) que contiene las anclas <a href> a los componentes tar_x.htm. Esto es a

los exámenes correspondientes a cada etapa del curso. Sólo se ha mostrado aquí parte del código, el resto se muestra en el Anexo III.

El despliegue de este código se puede ver en la Figura e del "manual de operación"

Cuadro 5.2 Celda de la tabla con la referencia a la página del EXAMEN1, el documento tar1.htm

```
<table align = "center">
<tr>
<td bgcolor="#FFFFCC" align = "center">
<a href="../tareas/tarea1/tar1.htm">
<b> EXAMEN I</b>
</a>
</td>
</tr>
//*****Aquí van otras celdas

//*****

</table>
```

5. 6 Dependencias del componente Tar_x.htm

La página Web, correspondiente al examen seleccionado (por ejemplo la página tar1.htm), al pulsar el botón “GENERAR” se presenta una ventana con un examen de reactivo seleccionados aleatoriamente.

Parte del código de la página se muestra en el Cuadro 5.3; incorpora las referencias a los script de JavaScript. El código completo se muestra en el anexo IV.

En las referencias, el archivo base1.js contiene los reactivos que serán seleccionados como parte de la función **Preguntas()** (anexo V). La referencia al archivo **pazar.js** permitirá el acceso al código de la rutina de **Azar()**, que obtiene los números de reactivos seleccionados (anexo VI). El código de la rutina **Examen()** que utiliza a los datos obtenidos de las rutinas **Preguntas()** y **Azar()**, se halla en el archivo **exam.js** (anexo VII):

Cuadro 5.3 Código del archivo tar1.htm con las referencias a los Scripts

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/strict.dtd">

<html xmlns="http://www.w3.org/TR/xhtml1">
<head>
<title> </title>
<link rel =stylesheet href="../../code/hoja1.css" type ="text/css">
<style>
td{padding-left:10;padding-right:10; padding-top=10;}
</style>
<script language="JavaScript" src="../../code/base1.js">
</script>
<script language ="JavaScript" src="../../code/pazar.js">
</script>
<script language="JavaScript" src="../../code/exam.js">
</script>
</head>
<body background = "../../tapiz/tapiz6.jpg">
```

Cuadro 5.3 Código del archivo tar1.htm con las referencias a los Scripts

En el Cuadro 5.4, se muestra el fragmento de código correspondiente al formulario con la entrada que define el botón. Nótese que la función Examen se activa por el evento “onClick”.

Cuadro 5.4 la forma <form> que define la entrada del botón que activará el GAEM.

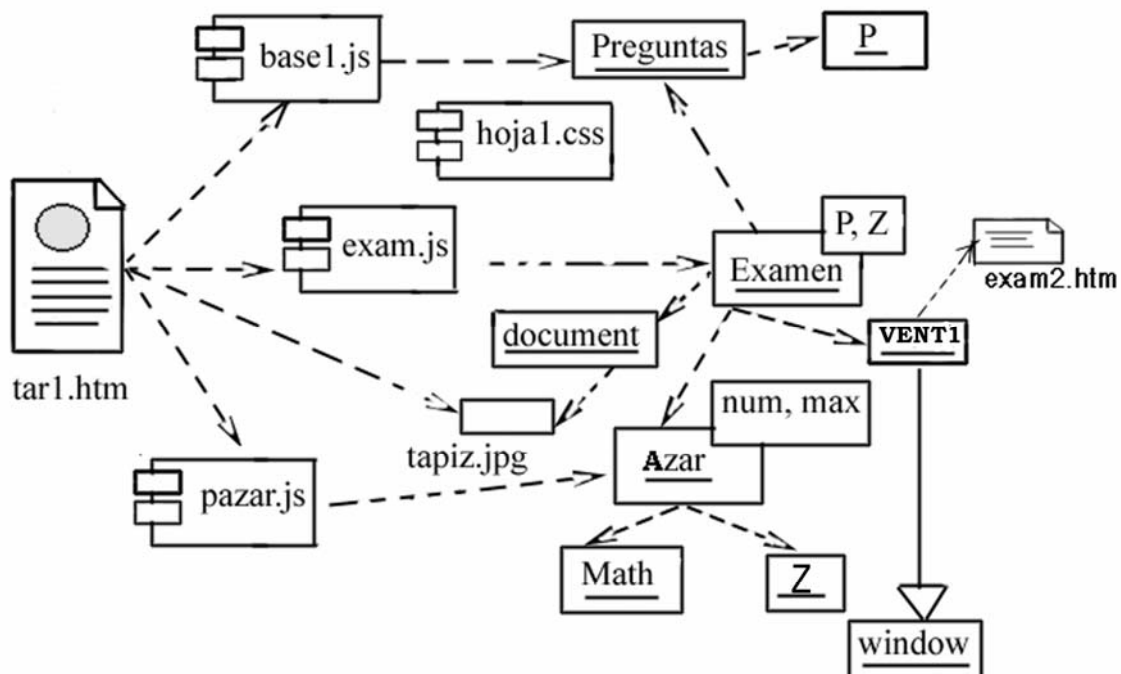
```

<form>
  <input type="button" value="GENERAR"
onClick="Examen(Preguntas(), Azar(2,16))"; >
</form>

```

La Figura 5.4, muestra las dependencias del componente tar1.htm (objetos y componentes); destacan los componentes de extensión “.js”, que contienen los scripts con el código JavaScript. Los objetos constituyentes o relacionados con estos archivos de script son también evidenciados en dicha figura.

Es en la porción de código mostrada en el Cuadro 5.4, **donde podemos realizar las modificaciones necesarias para hacer variar el número de preguntas del examen obtenido por el GAEM**. El parámetro que regresa la función **Azar(2,16)** y aportado a la función **Examen()**, se halla determinado por la selección de 2 preguntas de 16.

**Figura 5.4** Objetos asociados al componente tar1.htm

Esta elección de dos preguntas, fue una decisión del profesor del curso, y puede ser modificada, pero no en tiempo de "ejecución", pues no es decisión del alumno, sino predeterminada, como una condición del examen.

El generador aleatorio de exámenes matemáticos se basa en la actividad de la función `Examen()`, un objeto de JavaScript, que a su vez requiere de los parámetros aportados por los objetos `Preguntas()` y `Azar()`. En la Figura 5.5 se resaltan las dependencias del objeto `Examen()`, ya expuestas de algún modo en la Figura 5.4:

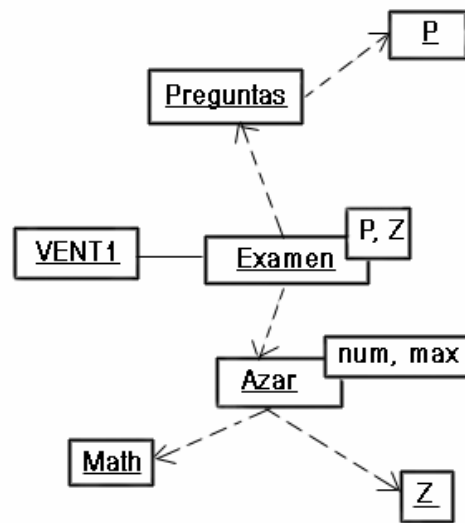


Figura 5.5 Objetos JavaScript que participan en la generación aleatoria de exámenes matemáticos

El objeto `Examen()`, recibe los parámetros **P** (preguntas) y **Z** (número obtenido aleatoriamente), respectivamente de la actividad de los objetos JavaScript (funciones **Preguntas()** y **Azar()**). La imagen también muestra la interfaz `VENT1`, la cual es el examen matemático producido aleatoriamente. Aquí la Interfaz se describe con un diagrama de objeto.

5.7 Diagrama de actividad de la función Azar()

La función **Azar()** realiza el cálculo de números aleatorios mediante el objeto interno **Math** que aporta los métodos **random()** y **round()**, la primera función devuelve un número entre 0 y 1; la segunda redondea al entero más cercano.

El número x , obtenido mediante **Math.random()** y **Math.round()**, es transformado en un número tal que $0 < x < 100$; condición que cambiaría si el número de los reactivos de un banco de reactivos del examen tuviera 100 o más reactivos.

El primer número obtenido ($i = 0$) siempre se incorpora sin condiciones al arreglo **Z[]** de números obtenidos aleatoriamente, para cualquier otro número debe verificarse que no se halle repetido en dicho arreglo, de otro modo deberá volverse a realizar el proceso de obtención de un nuevo número. Cada incorporación de números al arreglo **Z[]** incrementa el índice i , hasta que $i \geq \text{num}$, donde num es el número de preguntas del examen.

Una descripción más completa de la actividad de la función **Azar()** se muestra en la Figura 5.6; parte del código se muestra en el cuadro 5.5 y en el anexo VI.

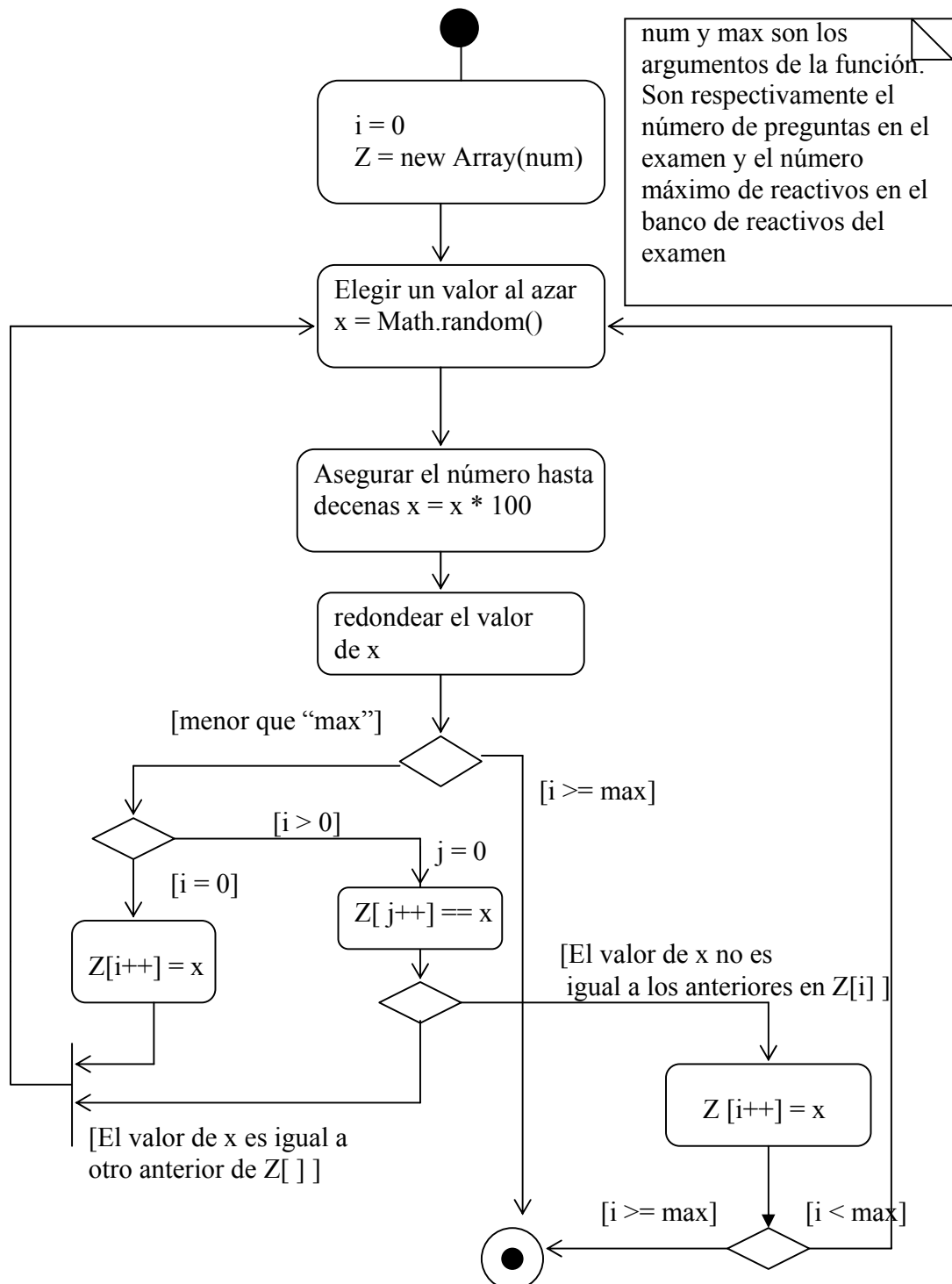


Figura 5.6 Diagrama de actividad de la función Azar(num, max)

Cuadro 5.5 Código JavaScript de la función Azar(num, max).

```
//componente (archivo fuente) pazar.js

<!--
function Azar(num,max)
{ var x;
  var i=0;
  var j;
  var Z = new Array(num);
  do{
    x=Math.random(); // obtiene números entre 0 y 1
    x=x*100;         //se aseguran número hasta decenas
    x=Math.round(x); //redondea el valor
    if(x<max) //se seleccionan menores a "max"
    {
      //sólo el primer dato se carga libremente
      if(i==0)
        Z[i++] =x;
      else {
        for(j=0;j<i;j++){
          //sólo se guardan datos que no sean iguales a los ya obtenidos
          if(Z[j]== x){
            //no guarda si son iguales, vuelve a obtener otro
            break;
          }
        }
        if(Z[j]!=x)
          Z[i++] = x;
      }
    }
  }

  }while(i<num); //el total de números solicitados

  return Z; //obtenemos un arreglo
}

//-->
```

5.8 La función Preguntas()

La función **Preguntas()** contiene un objeto , el arreglo **P[]** el cual debe redefinirse al variar el número de reactivos. El Cuadro 5.6 que contiene parte del código evidencia que la función **Preguntas()** es un banco de reactivos que para el caso particular conforma el componente “base1.js”, nombre del archivo fuente con los reactivos del examen 1. La actividad de **Preguntas()** consiste simplemente en aportar el objeto **P[]** a la función **Examen()**. Otras funciones **Preguntas()** de contenido diferente (la asignación a **P[]** es diferente) se hallan en los archivos **base_x.js**.

Cuadro 5.6 Código JavaScript de la función Preguntas()

```
//Archivo fuente “base1.js”
<!--
function Preguntas()
{
    var P= Array(16);

    //*****Preguntas
    P[0] = "<p><b>1.</b> Expresa el vector &#160; ( 2, -4, 1 ) &#160;
    como <b>combinaci&#225;n lineal</b> de los vectores: </p>
    <p><div align='center'>( 3, 2, 1 ); ( -2, 0, 3 ) &#160; y &#160; ( -4, 3,
    0)</div></p>";

    P[1]= "<p><b>2.</b> &#191;El vector &#160; ( <img src='1_2.gif'
    align='center', 2 ) &#160; es <b>combinaci&#225;n convexa</b>
    de los vectores &#160; ( 3, 0 ) &#160; y &#160; ( 0, <img src='3_2.gif'
    align='center'> ) ? <br>Justifique la respuesta.</p>";

    P[2]= "<p><b>3.</b> Dado el <b>hiperplano</b>: &#160;
    2x<sub>1</sub> - 2x<sub>2</sub> + 4x<sub>3</sub> +
    6x<sub>4</sub> = 7 &#160; &#191; en qu&#233;
    <b>semiespacio</b> est&#225; el vector &#160; ( 6, 1, 7, 2 )
    ?</p>";

    //*****
    //*****Aquí van más preguntas sólo se muestran 3
    //*****
    return P;
}

//-->
```

El arreglo `P[ ]`, se inicializa con una cadena de caracteres que consiste de código XHTML, por ello las letras acentuadas como á, ó, deben ser reemplazadas por su código o nemónicos que representan caracteres latinos referentes a un subconjunto del estándar "ISO/IEC 1046-1:1993", denominado UCS-2, que usa dos bytes.

Así en vez de á debe anotarse (`á`) o el nemónico "`á`", el nemónico para ó es "`ó`". Análogamente podemos usar nemónicos o código en las letras como ñ (`ñ`); ü (`ü`); el caracter que representa un espacio en blanco se anota como "` `"; el signo "¿", debe anotarse "`¿`" (nemónico `¿`).

En el anexo V (**Función Preguntas() del componente base1.js**), se muestra la incorporación completa de 16 reactivos al arreglo.

5.9 Diagrama de actividades de la función Examen()

La función **Examen()** cuyo código se halla en el anexo VII y en el Cuadro 5.7 tiene como parámetros **P** y **Z** que son respectivamente el arreglo con reactivos y el de los números de reactivos seleccionados al azar. En la función Examen se ha definido un objeto ventana que al desplegarse presenta la página "exam2.htm"; el contenido de la página se define mediante el objeto document y sus métodos `writeln( )` y `write( )`.

Es de resaltar que gracias a las facilidades del propio lenguaje de desarrollo de estas funciones como es JavaScript, la función **Examen(P, Z)** que como se indica recibe explícitamente los parámetros **P** y **Z**, implícitamente recibe el número de preguntas que contendrá el examen, pues siendo **Z** un arreglo, **Z.length** da el número de componentes del arreglo, que coincide con el número de datos almacenados en dicho arreglo que son los números de preguntas seleccionadas al azar.

Cuadro 5.7 Código JavaScript de la función Examen()

```
//archivo fuente “exam.js”

<!--
function Examen(P, Z)
{ var i;
  var win=window.open("exam2.htm","VENT1","width=600,height=400,
    resizable=1,scrollbars=1,left=100,top=250");

  win.document.writeln("<HTML><head><link rel =stylesheet
href='../..'/code/hoja1.css' type = 'text/css'></head><body background
= '../..'/tapiz/tapiz6.jpg'>"+ "<br />" + "<div align = 'justify'>");

    for(i=0; i<Z.length; i++){
      win.document.writeln(P[Z[i]]+"<br />");
      win.document.writeln("<div align='center'><hr size='4' /></div>");
    }
    win.document.writeln("<br />");

    win.document.write("</div></body></HTML>");
  }

  //-->
```

En el Cuadro 5.7, se muestra que la función hace uso de los componentes “**hoja1.css**” y “**tapiz6.jpg**”. En la Figura 5.7, se muestra el diagrama de actividades de la función **Examen()**.

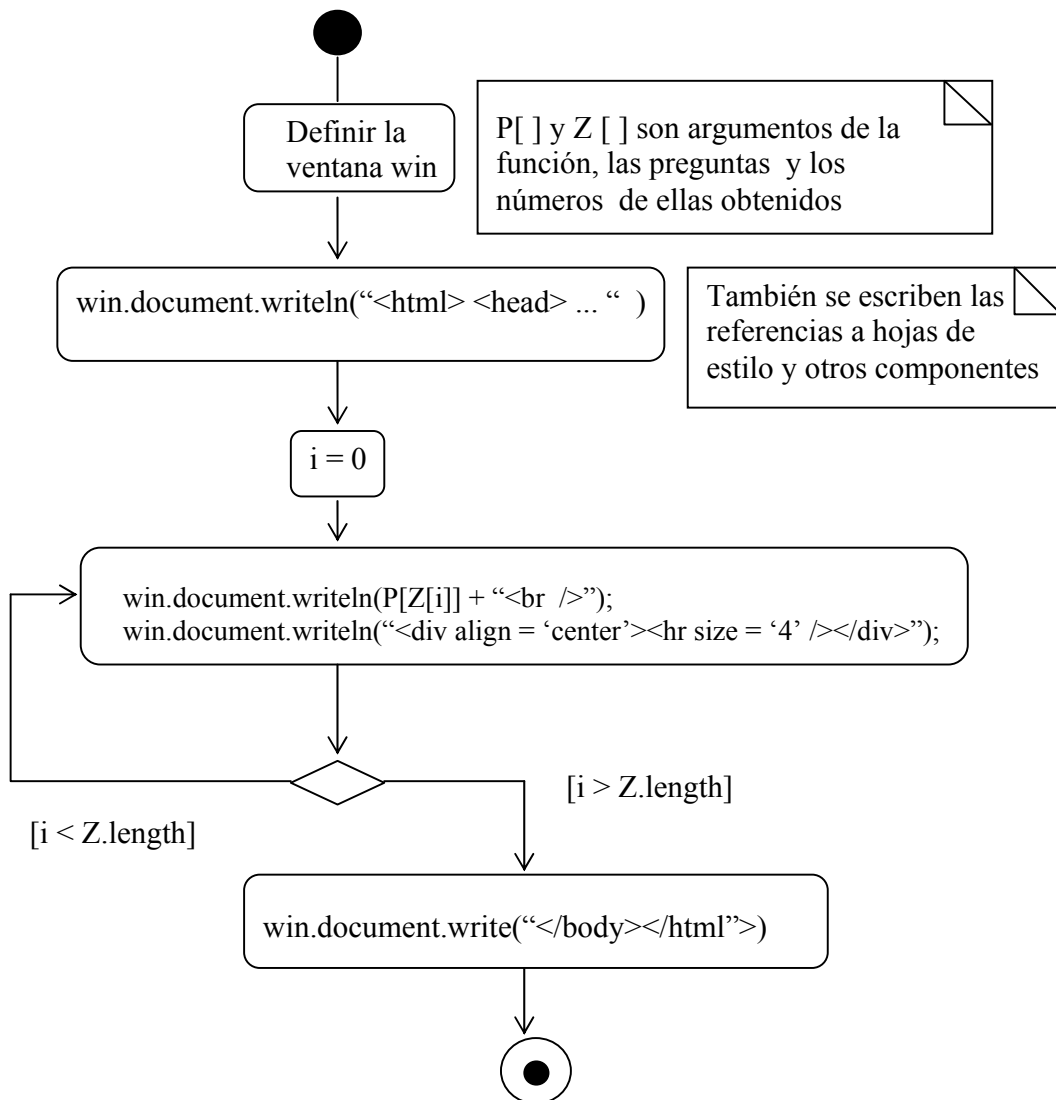


Figura 5.7 Código JavaScript de la función Examen(P, Z)

En el diagrama de la Figura 5.7, sólo se indica que se define una ventana, diferente a la ventana activa del navegador, con el método **open()** del objeto window. La ventana generada se asigna, para un mejor control a la variable **win**; en el Cuadro 5.7, se aclara que el nombre JavaScript de esta ventana independiente es “**VENT1**”.

El objeto **document** de la jerarquía superior de objetos JavaScript, es el responsable de generar la página Web con las especificaciones de las preguntas y formato propuesto, sobre una ventana llamada **VENT1**; del

Cuadro 5.7, se tiene que la página Web desplegada, en la nueva ventana, se denomina “**exam2.htm**”.

Resumen del capítulo

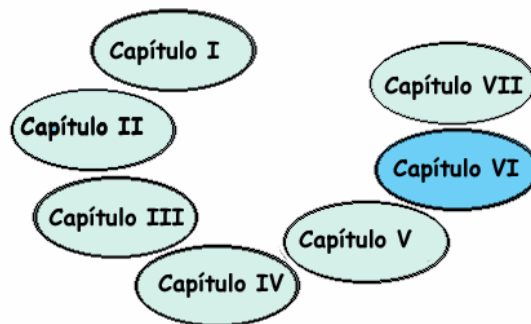
El Generador Aleatorio de Exámenes Matemáticos, fue escrito utilizando JavaScript e insertando las funciones que se activan por evento en páginas Web desarrolladas con código HTML en transición a XHTML, así como lenguaje de hojas de estilo CSS.

La función Examen(), usa los parámetros P y Z que son arreglos, el primero es aportado por la función Preguntas() y el segundo Z por la función Azar(). El objeto P[] es un arreglo con preguntas y Z[] es un arreglo con los números de preguntas que integrarán el examen. La actividad principal en la función Examen(), la realiza el objeto document, responsable de dar formato y escribir (con el método writeln), las preguntas en la pagina Web que se desplegará sobre la ventana independiente del navegador llamada VENT1.

A continuación se muestran los resultados obtenidos del funcionamiento de los programas de cómputo de autoevaluación y del Generador Aleatorio de Exámenes Matemáticos.

CAPÍTULO VI.-

Resultados obtenidos



Contenido del Capítulo

Se discuten los resultados del funcionamiento de los programas en diversos navegadores y sistemas operativos, entre los cuales se hallan Linux, Windows 98, Windows Milenium, Xp, Mac Os9, Mac OSX y los navegadores de prueba Mozilla FireFox, IExplorer, Flock, Opera, Netscape Communicator, Safari.

Se describen algunos desarrollos que comparten características de los programas presentados en la tesis.

6.1 Resultados del programa de autoevaluación “clásico” y “modificado”.

Se realizaron pruebas del comportamiento de los programas en diferentes plataformas (sistemas operativos y máquinas). Los sistemas operativos utilizados son las versiones de Windows 98, Windows 2000, XP, Mac Os 9 y OS X, en equipos PC y Macintosh. Los navegadores utilizados fueron Netscape 8.01, Opera 9.27, Mozilla-Firefox 2.0.0.14, IExplorer 6.0, Safari 3.1.1, Flock 1.1.2

El servidor con la información consistente en documentos html, archivos con código java Script de extensión .js, hojas de estilo .css, así como diversos elementos gráficos de extensión .gif, .jpg, se sustenta en el sistema operativo Linux, el servidor de servicios HTTP es Apache.

6.1.1 Resultados del programa de autoevaluación "clásico".

El programa de evaluación “clásico”, corresponde a un examen de falso/verdadero, el algoritmo de cómputo en esencia es un caso particular del programa de evaluación de exámenes de opción múltiple. Por ello, habiendo desarrollando el programa para la automatización de las correcciones de examen, fácilmente se logró adoptarlo al caso de mayor número de opciones, que es presentado en el programa de autoevaluación "modificado".

Tabla 6.1 Sistemas operativos y navegadores con los cuales se probó la ejecución del programa de evaluación "clásico".

El símbolo ✓ indica ejecución correcta

Naveg. \ Sist. Op.	Windows 98	Windows 2000	Windows XP	Mac Os 9	Mac Os X
IExplorer	✓	✓	✓	✓	✓
Flock	✓	✓	✓	✓	✓
Opera	✓	✓	✓	✓	✓
Mozilla FireFox	✓	✓	✓	✓	✓
Safari	✓	✓	✓	✓	✓

6.1.2 Resultados del programa de evaluación "modificado"

Se halló una diferencia de comportamiento, en la ejecución del programa de evaluación modificado al desplegar la ventana de resultados, ya que con IExplorer 6.0, Mozilla-Firefox 2.0.0.14, Netscape 8.01 bajo Windows 98, y en Mac Os 9 al utilizar el método open() del objeto Window de JavaScript, éste crea una página Web virtual, en la cual se muestran los resultados y explicaciones, lo cual no sucede con Windows XP y Mac OS X, pues dicha página debe existir físicamente, por ello fue incorporada la página de despliegue o componente **reval.htm**.

Resuelto este problema, se encontró que sólo el navegador Safari, aunque es rápido en la interpretación de código JavaScript, **no** es capaz de realizar un despliegue correcto de la ventana de resultados, porque no logra incorporar objetos gráficos al aplicar el método document(). En el Cuadro 6.1 se muestra parte del código presentado en el Cuadro 4.9, correspondiente a la función Estadist2()

Cuadro 6.1 Fragmento de código de la función Estadist2()

```
VENT1=
window.open("reval.htm","VENT1","width= 500,height= 200,scrollbars=1");
VENT1.document.writeln("<html><head> </head> <body background =
'back1.gif'>");
VENT1.document.writeln("<b>Buenas:                </b>"+Buenas+"<br
/>"+ "<b>Malas:    </b>"+ "&#160;" +Malas+"<br />"+ "<font color =
'blue'><b>"+texto + "</b></font>" + "<br /> <br />"+ texto2);
VENT1.document.writeln("</body></html>");
VENT1.document.close();
```

Como puede apreciarse en el cuadro 6.1, document(), contribuye a conformar una página cuyo imagen de fondo lo determina el código

<body background = 'back1.gif'>

Esta porción de código **no es interpretada correctamente por el navegador Safari**, como se expondrá más adelante, la explicación se halla en que al parecer Safari no logra referenciar imágenes, mediante el objeto **document**.

En las versiones de Safari para OS X 10.2 se observó que al activarse la ventana presenta la página Web correctamente y después la deja en blanco. La Tabla 6. 2 resume las pruebas realizadas sobre diferentes navegadores.

Tabla 6.2 Sistemas operativos y navegadores con los cuales se probó la ejecución del programa Generador Aleatorio de Exámenes Matemáticos.

Naveg. \ Sist. Op.	Ejecución correcta: ✓			Ejecución incorrecta: X	
	Windows 98	Windows 2000	Windows XP	Mac OS 9	Mac OS X
IEExplorer	✓	✓	✓	✓	✓
Flock	✓	✓	✓	✓	✓
Opera	✓	✓	✓	✓	✓
Mozilla FireFox	✓	✓	✓	✓	✓
Safari	X	X	X	X	X

Al investigar sobre programas parecidos a los que aquí se llaman de "autoevaluación clásico" y "modificado", encontramos casos en que han incorporado la idea de aleatoriedad y cierto manejo de multimedia, en algunos casos la obtención de exámenes al azar, los anuncian como generadores de exámenes, por ello se ejemplifican en la siguiente sección.

6.2 Resultados del “Generador Aleatorio de Exámenes Matemáticos” (GAEM)

Al igual que en el programa de autoevaluación "modificado", al realizar las pruebas en diferentes plataformas (sistemas operativos y máquinas), como son las versiones de Windows 98, XP y Macintosh se presenta una diferencia de comportamiento, en el despliegue de la ventana de resultados, al ejecutarse el programa **GAEM**, ya que con IEExplorer 6.0, Mozilla-Firefox 2.0.0.3, Netscape 8.01 bajo Windows 98, y en Mac Os 9 al utilizar el método open() del objeto window de JavaScript, éste crea una página Web virtual, en la cual se despliegan los resultados y explicaciones, lo cual no sucede con Windows XP, pues dicha página debe existir físicamente, por ello fue incorporada la página de despliegue o componente exam2.htm.

Resolviendo el problema anterior, las pruebas de ejecución sobre los sistemas operativos como Windows 98, Windows 2000, Windows Xp, Mac OS 9, Mac OS x, Linux y los navegadores más utilizados como Opera, IEExplorer, Flock, Opera, Mozilla Firefox, se encontró que el único navegador que no interpreta correctamente el programa GAEM, es Safari, ya que no es capaz de desplegar imágenes al aplicar los métodos write() o

writeln() del objeto **document**, de forma análoga al problema de interpretación hallado en el programa de autoevaluación "modificado".

En el caso de Safari para la versión de OSX 10.2, hallamos al igual que en el caso del programa de autoevaluación "modificado", que despliega correctamente la ventana con el examen, pero después deja en blanco la página.

La tabla 6.3, reúne los resultados hallados al ejecutar el GAEM sobre diferentes sistemas operativos y navegadores.

Tabla 6.3 Sistemas operativos y navegadores con los cuales se probó la ejecución del programa de evaluación "modificado".

Naveg. \ Sist. Op.	Ejecución correcta: ✓			Ejecución incorrecta: X	
	Windows 98	Windows 2000	Windows XP	Mac Os 9	Mac Os X
IEexplorer	✓	✓	✓	✓	✓
Flock	✓	✓	✓	✓	✓
Opera	✓	✓	✓	✓	✓
Mozilla FireFox	✓	✓	✓	✓	✓
Safari	X	X	X	X	X

Los exámenes desplegados mediante el “Generador Aleatorio de Exámenes Matemáticos” (**GAEM**) pueden mostrar tablas y expresiones matemáticas diversas.

Generación de exámenes con tablas coloridas imágenes de diversos grados de complejidad, audio y video, propondrán en el futuro nuevos tipos de exámenes matemáticos.

El programa GAEM es capaz de desplegar exámenes matemáticos de forma aleatoria, no importa la complejidad de las relaciones matemáticas, asimismo despliega tablas coloridas, incluso applets de Java.

Al investigar en torno a software generador de exámenes se encontraron propuestas de software que se presentan como generadores de exámenes incluso matemáticos, algunas de ellas son:

MATEMÁTICA IES. Una página Web cuyo objetivo **no** es tener una base de datos de ejercicios de esta área, sino el poder seleccionar un grupo de ejercicios e imprimir en una forma totalmente configurable: Colocando título personalizado, elegir el número de ejercicios que se crea pertinente. Cuenta con objetos de aprendizaje como animaciones, presentaciones. y

recursos como OpenOffice. Este desarrollo data del año de 2006 [Web19, 2008].

Generador de Exámenes (Ibertalleres). Programa que genera exámenes con preguntas de Prevención de Riesgos Laborales elegidas aleatoriamente entre una amplia lista para contestar on-line. Así mismo las respuestas son colocadas de forma aleatoria de modo que es prácticamente imposible que se creen exámenes iguales o incluso parecidos.

Para generar un cuestionario se tiene que seleccionar el número de preguntas que se desean o pulsar sobre el botón 'Generar' en el caso de querer un examen con un número de preguntas igual al que marca el desplegable [Web20, 2008].

Este generador de exámenes, *no es para matemáticas*, combina nuestra idea de aleatoriedad y presentación de un examen que puede ser evaluado de forma automática y que da el resultado al modo en que se tiene en el programa de autoevaluación "clásico".

Las versiones de este software son para equipos PC con sistema operativo Windows.

Avaluator 3.0. Este software está diseñado para evaluar alumnos de los diferentes centros educativos de España. Es un programa que consta de dos aplicaciones: Un editor pensado para formular las preguntas de un control y un programa que pregunta y evalúa los contenidos formulados por el editor.

El programa edita y pregunta 5 tipos de cuestiones:

- Tipo test : hasta 8 opciones
- Tipo pregunta con respuesta corta (no dejando escribir mientras cada letra se escriba correctamente)
- Tipo pregunta referida a una imagen con respuesta corta (no dejando escribir mientras cada letra se escriba correctamente)
- Tipo pregunta referida a colorear una zona de un gráfico.
- Tipo pregunta referida a la comprensión de un pequeño texto.

Desgraciadamente la versión gratuita que se ha probado sólo contempla la materia de ciencias sociales [Web21, 2008].

Avaluator 3.0., muestra sin embargo un gran adelanto en la educación para niños, pues con multimedia y preguntas de respuesta corta se logra una evaluación automatizada, por lo demás amena. Al igual que en nuestro caso del programa "clásico" indica dónde se cometieron errores, con la modalidad de que vuelve a presentar el texto de la pregunta.

Las versiones de este software son para equipos PC con sistema operativo Windows.

Random Test Generator Pro. Un producto de Hirtle Software. Es un software orientado al profesor permitiéndole crear bancos de pruebas, a partir de los cuales se obtienen exámenes. Tiene la opción de poder escuchar las preguntas. Es un software que permite utilizar recursos multimedia, para incrustarlos en el examen. Incluso puede editar pruebas para su aplicación en Internet [Web22, 2008].

La versión de prueba tiene varias limitaciones; sin embargo se observa que es una modalidad de opción múltiple, en las que en vez de seleccionar botones de radio, se pulsan botones. Se puede incorporar explicación a la respuesta.

Las versiones de este software son para equipos PC con sistema operativo Windows.

Tabla 6.4 Comparación de GAEM y desarrollos no nacionales, difundidos en Internet.

	Examen Aleatorio	Base de datos de ejercicios	Capaz de incorporar multimedia	Examen de matemáticas	Evaluación automatizada
GAEM	Si	Si	Si	Si	No
Matemática IES	No	Si	No	Si	No
Generador de Exámenes	Si	Si	No	No	No
Avaluator 3.0	No	No	Si	Si	Si
Random Test Generator Pro	No	Si	Si	Si	Si

Resumen del capítulo

Se realizaron pruebas en diferentes sistemas operativos y navegadores, hallándose que el programa de autoevaluación "clásico", funciona correctamente con las plataformas probadas; no así el programa de autoevaluación modificado el cual no ejecuta correctamente con el navegador Safari, debido a que no interpreta las referencias a archivos de imágenes.

Las pruebas con el Generador Aleatorio de Exámenes Matemáticos (GAEM) nos indican que realiza sus tareas adecuadamente en diversas plataformas; sin embargo no despliega correctamente en el navegador Safari, pues no mantiene estable la página..

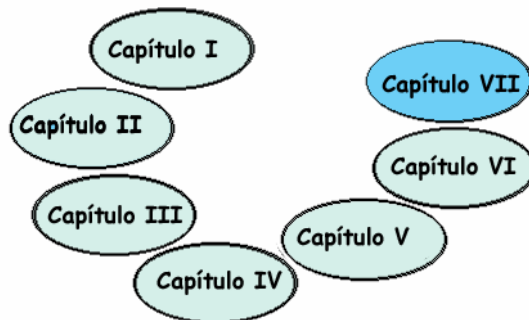
Comparando con otros programas de difusión en Internet, hallamos que existen combinaciones de la idea de aleatoriedad con variantes de los programas de autoevaluación "clásico" y "modificado", siendo más original nuestra propuesta del Generador Aleatorio de Exámenes Matemáticos.

Se realizó una revisión de programas generadores de exámenes como matemática IES, Generador de Exámenes, Avaluator 3.0, Random Test Generator Pro y se comparó con nuestro desarrollo GAEM.

.

CAPÍTULO VII.-

Conclusiones y trabajos futuros



Contenido del Capítulo

Se exponen las conclusiones derivadas de la investigación, tanto en lo relativo a la utilidad de los programas propuestos, su coincidencia o similitud con otros desarrollos, de las pruebas realizadas, en diferentes sistemas operativos y browsers.

Se proponen los desarrollos futuros y condiciones para mejorar las propuestas actuales.

7. 1 Conclusiones

La presente tesis es una contribución al desarrollo de sistemas de evaluación en línea, presentando los resultados de dos desarrollos para evaluación automatizada y un generador aleatorio de exámenes matemáticos. En el documento presentado se realizó la modelación de los tres programas propuestos utilizando UML (Unified Modeling Language, *Lenguaje de Modelado Unificado*), aplicando las facilidades del mismo al utilizar iconos propuestos por el usuario, lo cual permite mayor expresividad en la descripción de la arquitectura del sistema.

Uno de los lenguajes utilizados en el desarrollo de las funciones modulares de los programas descritos en la tesis fue JavaScript, al no basarse en clases, sino en prototipos, y contar con "funciones libres", que no son métodos de alguna clase u objeto, planteó problemas de representación que fueron resueltos utilizando iconos análogos a los utilizados por UML para clases parametrizadas.

Se ha cubierto el objetivo de desarrollar dos programas de evaluación, que aquí se han denominado "clásico" y "modificado", sin que los resultados pasen a una base de datos, por ende el alumno, puede reintentar una y otra vez, hasta contestar correctamente, de ahí que se trate más estrictamente de ejercicios de autoevaluación; sin embargo, los algoritmos propuestos se pueden orientar a exámenes de asignación de calificación para el alumno.

De la revisión hecha en el capítulo II, se concluye que la automatización del proceso de evaluación es un ingrediente importante en el software de administración y gestión de conocimientos. Sin embargo, evaluar va más allá del aspecto tradicional de medir los aprendizajes.

Limitándonos a los aprendizajes, del capítulo III se concluye que las pruebas pedagógicas que presente cualquier programa de evaluación que se desarrolle, deben llenar requisitos que los estudiosos de la didáctica han fundamentado ampliamente. Los profesores deben ser cuidadosos al proponer los exámenes, para que el sistema de evaluación, cumpla también sus cometidos en términos didácticos y pedagógicos.

Asimismo se concluye que las plataformas de gestión de contenidos y aprendizajes, incluyen sistemas de evaluación, que incorporan propuestas en algunos casos parecidas a las aquí presentadas, aunque en lo tocante al aspecto de generación de exámenes matemáticos, tenemos una propuesta más original.

En el capítulo VI, se expusieron algunos ejemplos de generadores de exámenes, casi siempre conforme más complejo es el programa, implicará

un costo, que el politécnico de aplicar o impulsar desarrollos como los aquí propuestos, no sólo tendrá ahorros monetarios, sino el control tecnológico de los mismos.

Por otro lado, cabe mencionar que buena parte de los programas que generan exámenes es software para que uso o satisfacción de necesidades del maestro directamente, bien para confeccionar el examen u obtenerlo de una base de información. Se han hallado ejemplos donde los exámenes son de opción múltiple con aleatoriedad, análogos al programa de "autoevaluación modificado" aquí propuesto, destacándose la tendencia a la multimedia.

Parte del trabajo de tesis se ha enfocado a un ámbito específico de la evaluación en la cual privilegia la retroalimentación para el alumno, pues tal como realizan su actividad estos programas de evaluación automatizada llamados "clásico" y "modificado", los resultados sirven al alumno fundamentalmente, puede hablar de "ejercicios de autoevaluación", pues el alumno obtiene un resultado pero la calificación **no se registra en alguna base de datos** con la finalidad de ponderarse para una decisión final.

El primer programa de autoevaluación se ha denominado "clásico" en el sentido de que gran número de plataformas de gestión de contenidos de aprendizaje, presentan éste tipo exámenes y el proceso de corrección automatizada **sólo aporta una contabilidad de aciertos**. Se indica el número de aciertos y errores, así también informa en qué preguntas se eligieron opciones incorrectas.

Con el programa de autoevaluación "modificado", en cambio, se logró desarrollar una propuesta original, que permite la intervención de tutores en los procesos de enseñanza aprendizaje. En este sentido, la relevancia del programa de autoevaluación "modificado" y considerando su ulterior desarrollo, reside tanto en la capacidad de retroalimentar al alumno, como también permite que los expertos pueden aportar experiencias en la resolución de problemas complejos, contribuyendo significativamente al aprendizaje, particularmente allí donde se requiera entrenamiento de personal, pues la versión "modificada" (con retroalimentación) del programa de resultados automatizados, puede incluir aplicaciones multimedia; para ello debe modificarse la cadena que inicializa el arreglo RESP[] (anexo II).

Los exámenes de opción múltiple con un mayor número de preguntas son una buena oportunidad para proponer mecanismos de retroalimentación de los alumnos, ya que no resulta obvio como en los reactivos del tipo falso/verdadero donde se halla el error; en la versión "modificada" del programa de evaluación los resultados se despliegan en una página Web.

Es necesario utilizar caracteres unicode para el despliegue correcto de caracteres latinos, en las cadenas que inicializan RESP[], no así en el código que despliega páginas Web, sin embargo debe utilizarse los códigos para caracteres latinos en páginas Web del estándar ISO-IEC 10646.

Al realizar las pruebas en diferentes plataformas (sistemas operativos y máquinas), como son las versiones de Windows 95, XP y Macintosh. Logramos hallar una diferencia de comportamiento, en el despliegue de la ventana de resultados, al ejecutarse el programa de autoevaluación "modificado", ya que en Windows 98, y en Mac Osx al utilizar el método open() del objeto window de JavaScript, éste crea una página Web virtual, en la cual se despliegan los resultados y explicaciones, lo cual no sucede con Windows XP, pues dicha página debe existir físicamente, por ello fue incorporada la página de despliegue o componente **reval.htm**.

El **Generador Aleatorio de Exámenes Matemáticos** (GAEM), desarrollado construye exámenes a partir de un banco de reactivos, se ha logrado con él 1) El control del algoritmo de selección aleatoria de reactivos de pruebas pedagógicas y 2) desplegar expresiones matemáticas, que por el momento muchas de ellas son imágenes en formato "gif".

El primer punto facilitará preparar diversas modalidades de exámenes para su corrección automática (opción múltiple, correspondencia, etc.), pero con la posibilidad de que los reactivos sean obtenidos aleatoriamente. El segundo punto abre la posibilidad de reactivos multimedia, incluyendo la incorporación de programas implementados con lenguajes de Modelado de Realidad Virtual.

Sin embargo, el que las páginas Web de expresiones matemáticas aún se basen en el uso de imágenes para generarlas, tiene inconvenientes; la principal de ellas es la falta de proporcionalidad entre el tamaño del texto y la imagen de la expresión matemática, lo cual se hace muy evidente al cambiar las condiciones de despliegue.

De algún modo el producto Web se halla atado a las condiciones iniciales de resolución y tamaño de la fuente. Algunos aspectos como son la alineación con la fuente o la transparencia pueden ser paliados; sin embargo el modo en que se trata el alineamiento por diferentes browsers puede dar lugar a respuestas visuales inesperadas, sobre todo en expresiones matemáticas muy complejas. Finalmente el tratamiento de imágenes requiere de más ancho de banda y en páginas muy saturadas puede haber despliegues tardíos [web15b, 2007].

MathML, parece ser una posibilidad para el manejo futuro de expresiones matemáticas, su aplicación irá aparejada tanto a la difusión de visores y

browsers que decodifiquen la información para desplegar las ecuaciones, así como a la capacitación de personal, para la reescritura de numerosos documentos para el Web logrado cierto nivel de calidad.

Por otro lado, el hecho de que los exámenes de matemáticas (por ejemplo), propuestos por el **GAEM**, no puedan ser corregidos todavía de forma automática, emitiendo una calificación, no los excluye al igual que en los casos propuestos de los programas de autoevaluación, de tener que ser complementados con software que permita manejar bases de datos con los resultados parciales de exámenes y de actividades complementarias que lleven a la toma de una decisión.

El **GAEM** no permite que el estudiante establezca el número de preguntas del examen. Esto lo define el profesor o tutor en función de la naturaleza del contenido del examen, por su complejidad y laboriosidad. Sin embargo, el número de preguntas del examen, fácilmente puede ser modificado por el profesor o tutor, al modificar los parámetros de la función Examen() (Capítulo V).

Con respecto al GAEM, al realizar las pruebas en diferentes plataformas (sistemas operativos y máquinas), como son las versiones de Windows 95, XP y Macintosh. Logramos hallar una diferencia de comportamiento, en el despliegue de la ventana de resultados, al ejecutarse el programa de autoevaluación "modificado", ya que en Windows 98, y en MacOSx al utilizar el método open() del objeto window de JavaScript, éste crea una página Web virtual, en la cual se despliegan los resultados y explicaciones, lo cual no sucede con Windows XP, pues dicha página debe existir físicamente, por ello fue incorporada la página de despliegue o componente **exam2.htm**.

De las pruebas realizadas con diferentes navegadores y sistemas operativos podemos concluir que en equipos PC, el browser Safari no es recomendable para ejecutar los programas propuestos en la tesis, pues los métodos write() y writeln() de document, no logra referenciar correctamente imágenes (en PC) o no mantiene el despliegue de la página, dejándola en blanco (Mac OSX 10.2).

De las experiencias expuestas en los capítulos I y II, se puede concluir que un sistema *completo de evaluación de los aprendizajes*, debe ser capaz de autenticar al usuario, presentar el examen al solicitante del mismo, dar un resultado aprobatorio o no al examinado y enviar los resultados a una base de datos, para su posterior utilización estadística y de consulta.

Las evaluaciones automatizadas basadas en exámenes o que al menos éstos sean una parte importante de aquéllas, deben contar con mecanismos de seguridad para que la información no sea alterada, tanto

la que se entrega como la que se recibe; que el sistema no pueda ser intervenido para alterar calificaciones obtenidas. Los exámenes en particular los de opción múltiple, no deben dar pistas para hallar las respuestas, a quien conozca algo de programación y vaya a realizar el examen.

En un examen de autoevaluación, cuyo objetivo es retroalimentar al alumno, coadyuvar a detectar puntos de debilidad en el aprendizaje no debe preocuparnos la seguridad, por ejemplo si se descubre alguna pista en el código del programa, que de indicios de respuesta correcta, lo que se trata es que el alumno descubra la respuesta y como no afecta de forma inmediata la calificación, el alumno sólo perdería el tiempo, en indagar más allá de lo que es pertinente. Sin embargo, ello no puede ser la justificación para la apropiación indebida de los productos intelectuales. No se puede permitir el acceso al código fuente de los programas, debe restringirse el acceso a los componentes de software de evaluación.

Un sistema de evaluación automatizada “robusto” tiene la capacidad de identificar al usuario, emitir el examen al solicitante, realizar el proceso evaluatorio, realizar estadísticas que incluyan gráficos de algún alumno en particular o de conjuntos de alumnos, almacenar resultados globales e identificar las preguntas donde se presentó más tendencia al error por parte de los evaluados; finalmente, poder tener acceso a información tanto los alumnos, profesores, investigadores, así como las autoridades vía Internet o intranet.

Las evaluaciones automatizadas basadas en examen o que al menos éstos sean una parte importante de ellas, deben contar con mecanismos de seguridad para que la información no sea alterada, tanto la que se entrega como la que se recibe, que el sistema no pueda ser intervenido para alterar calificaciones obtenidas y en particular los exámenes de opción no deben dar pistas para hallar las respuestas, a quien conozca algo de programación y vaya a realizar el examen.

No ha sido el propósito de esta tesis desarrollar un sistema de evaluación “robusto”; sin embargo, el producto de ella forma parte esencial de un sistema de tales características. El sistema de autoevaluación “clásico” y el “modificado”, así como el “Generador Aleatorio de exámenes Matemáticos”, son propuestas que no pueden quedar fuera de un sistema de gestión educativa en un entorno de Web, los principios computacionales para su desarrollo son los mismos.

Un complemento o parte importante de un sistema de evaluación que las instituciones educativas comienzan a generar y utilizar cada vez más es el manejador de la base de datos para un entorno Web. Como evidencia de ello tenemos el caso de la Secretaría de Educación Pública (SEP) de México

que mediante una página Web permite el acceso a los padres de familia el acceso de la información del alumno mediante su CURP o datos del alumno [Web16, 2007].

El IPN, con el propósito de apoyar a las actividades de difusión y transparencia de la información manejada en esta institución para el conocimiento de sus integrantes y del público en general, se ha propuesto crear un sitio Web para la consulta e historial académico de alumnos de los niveles medio superior y superior [Web17, 2007]. Esto parece que va siendo realidad con el sistema SIGUE el cual pretende tener los datos de alumnos, profesores y materias, controla la documentación entregada por los alumnos y será un medio para que los padres de familia puedan conocer el avance académico de sus hijos, pues podrá generar reportes y estadísticas diversas.

Todo ello sin embargo debe ser parte de un sistema integral para el trabajo en línea, para el desarrollo de los ambientes virtuales de aprendizaje y en particular de un sistema de evaluación. Se tienen entonces avances en el nivel de presentación, ahora debe hallarse la pieza que los conecte con un sistema de evaluación.

7.2 Trabajos futuros

A partir de los programas expuestos en esta tesis, como parte de un sistema de evaluación es necesario:

1. Desarrollar programas que cubran otro tipo de reactivos; por ejemplo de respuesta corta y de correspondencia.
2. Incorporar la aleatoriedad y límites de tiempo en pruebas de corrección automática.
3. Experimentar con el desarrollo de pruebas pedagógicas multimedia, incluyendo programas desarrollados con lenguaje de modelado de realidad virtual.
4. Ir sustituyendo las páginas Web en las que predominaban las imágenes para representar expresiones matemáticas, por código MathML; a la vez que de manera franca y consciente acelerar la migración a XHTML y XML.
5. Tanto los programas de evaluación automatizada como en aquellos en los que aún el profesor realice la corrección del examen, debe desarrollarse el programa manejador de bases de datos con la facilidades que dichos programas prestan.

Referencias:

Bibliográficas

[Bobadilla, 2000]

Bobadilla, Sancho Jesús. *HTML dinámico, ASP y JavaScript a través de ejemplos*. Alfaomega. Bogotá 2000. Pp.119-132

[Booch, 1994]

Booch, Grady. *Análisis y diseño orientado a objetos con aplicaciones*. Addison Wesley 2a. edición . USA 1994. Pp. 104 y 217

[Booch, et al. 1999]

Booch, Grady; Rumbaugh, James; Jacobson Ivar. *El Lenguaje Unificado de Modelado*. Addison Wesley Madrid 1999. Pp. XIX, 25, 36, 37, 41, 195, 198, 199, 334, 408

[Cappelletti, 2004]

Cappelletti, Isabel. *Evaluación educativa. Fundamentos y prácticas*. Siglo XXI.México 2004. Pp. 9, 63

[Castro, 2001]

Castro, Elizabeth. *Perl y CGI*. Prentice Hall. España 2a. edición 2001. Pág. 2

[Contreras, 1997]

Contreras; Mayen Rita. “Reflexiones en torno al uso de la tecnología de la información en el terreno educativo” en *Soluciones avanzadas*. Año 5 No. 46. 15 de junio, México 1997. Pp. 10-15

[Darnell, 1997]

Darnell, Rick, et. al. *HTML 4.0. Unleashed*. Sams net. USA 1997. Pp. 43-50

[Deitel, 2004]

Deitel, Harvey M. ; Deitel, Paul J. *Cómo programar en Java*. 5a. edición Pearson México 2004. Pp. 104-110 y 172-180

[Fernández, 2004]

Fernández, Eva I. *E-learning. Implantación de proyectos de formación on-line*. Alfaomega Ra-Ma. México 2004. Pp. 4,5

[Flower, 1999]

Flower, Martin; Scott, Kendall. *UML gota a gota*. Addison Wesley Longman de México, México 1999. Pp. 108-110

[Flash, 2002]

Utilización de Flash. Macromedia EUA 2002. Pág. 165

[Galindo, 2006]

Galindo, Soria Leopoldo “Una metodología para el desarrollo y redacción de un proyecto de tesis de maestría” 3er. Congreso Internacional de Ingeniería Electromecánica y de Sistemas. SEPI ESIME Zacatenco Noviembre 25-29, 2006.

[Garrido, 2004]

Garrido, Najera Celso. “La educación virtual en México: universidad y aprendizaje tecnológico” en *Memoria de la 1a. Conferencia Internacional ELAC* 2004.

http://www.elacvirtual.net/documents/conferencias_elac/I_conferencia/11_celsojoridy.pdf, revisado diciembre 2006

[Goldfarb, 1999]

Goldfarb, Charles; Prescod, Paul. *Manual de XML*. Prentice Hall. Madrid 1999. Pp. 11, 79

[Informática, 1986]

Gran Enciclopedia Informática. Informática básica 2. Ediciones Nueva Lente. Madrid 1986. Pág. 8

[Jacobson, 2000]

Jacobson, Ivar; Booch, Grady; Rumbaugh, James. *El Proceso Unificado de Desarrollo de Software*. Addison Wesley, España. 2000. Pp. 4-7

[Kelley, 1995]

Kelley; dean. *Teoría de autómatas y lenguajes firmes*. Prentice Hall. Madrid 1995. Pág. 30

[López, 2004]

López, Torres Marcos. *Evaluación Educativa*. Trillas, México 2004, Pp. 5, 41-47

[La enciclopedia, 2004]

La enciclopedia. (20 tomos), tomos 7, 11. Salvat. Colombia 2004. Pp. 4929, 8190, 8191

[Manger, 1997]

Manger, Jason J. *Fundamentos de JavaScript*. McGraw Hill, México 1997. Pp. 4, 83-84

[Peredo, et al. 2002]

Peredo, Valderrama Ruben; Sheremetov Leonid; et al. *Generador de exámenes utilizando componentes de software programables reutilizables inteligentes para materiales multimedia en espacios virtuales de aprendizaje (EVA)*.

INFOREDU 2002.

<http://espejos.unesco.org.uy/simplac2002/ie.html>, revisado enero 2007

[Peredo, et al. 2004]

Peredo, Valderrama Ruben; Balladares, Ocaña Leandro; Sheremetov, Leonid B. *Development of intelligent reusable learning objects for Web.based education systems*. Expert systems with applications Volume 28, Issue 2, February 2005, Pages 273-283

[Rumbaugh, et al. 2000]

Rumbaugh, James; Jacobson, Ivar; Booch, Grady. *El lenguaje unificado de modelado. Manual de referencia*. Addison Wesley. Madrid 2006. Pp. 93, 130-132, 190, 307

[Sheremetov, et al. 2006]

L. Sheremetov, R. Peredo, L.. Balladares. *Student evaluation system for Web basesd on learning components and agent technology*. Proceedings of the fifth IASTED International Conference WEB.BASED EDUCATION. January 23-25, 2006. Puerto Vallarta México.

[Stevens, 2002]

Stevens, Perdita; Pooley, Rob. *Utilización de UML en ingeniería del software con objetos y componentes*. Addison Wesley. México 2002. Pp. 71, 174

[Suárez, 2002]

Suárez, Díaz Reinaldo. *La educación. Teorías educativas. Estrategias de aprendizaje*. Trillas México 2002. Pág. 83

[van Gigch, 2000]

van Gigch, John P. *Teoría general de sistemas*. Trillas México 2000. Pág. 17

En Internet

[Weblearning, 2006]

Weblearning Resources.

<http://www.knowledgeability.biz/weblearning/>, revisado diciembre 2006.

[Web0, 2006]

“Perspectivas de la educación a distancia y la educación de personas jóvenes y adultas de América Latina y el Caribe” en *Decisio. Educación a distancia para adultos*. <http://www.formatex.org/micte2006/virtual/pdf/181.pdf>, revisado septiembre 2006

[Web1, 2006]

Mendoza, Espinoza Ramón. *El desarrollo de cursos a distancia en la World Wide Web mediante plataformas virtuales “WebCT” en el mundo universitario moderno*.

http://www.cvc.cervantes.es/obref/formacion_virtual/metodologia/roman.htm
revisado septiembre 2006

[Web1a, 2006]

Jalil, Angulo R. I.; et. al. *Learning Object. Evaluación histórica*.

<http://www.formatex.org/micte2006/virtual/pdf/181.pdf>, revisado noviembre 2006

[Web1ab, 2006]

Navarro, Buendía Marcos. *Evaluación de plataformas de e-learning de licencia pública*. Proyecto de Carrera. Universitat de Valencia. Febrero 2004.

<http://www.uv.es/ticape/docs/nabuen/pfcarrera.pdf>, revisado noviembre 2006.

[Web1ac, 2006]

Fontán, Montesinos Ma. Teresa. *Evaluar a través de Internet. Bit Pixel. Revista de Medios y Educación. España*. Pp. 79-88

<http://redalyc.uaemex.mx/redalyc/pdf/368/36802406.pdf>, revisado octubre 2006.

[Web1ad, 2006]

Blackboard an WebCt complete Merge.

<http://www.blackboard.com/webct>, revisado octubre 2006.

[Web1b, 2006]

Chávez, Adriana. *Televisión educativa o televisión para aprender*. Razón y Palabra.

<http://www.razonypalabra.org.mx/anteriores/n36/achavez.html>, revisado noviembre 2006.

[Web1c, 2006]

Content Management System. Wikipedia, the free encyclopedia.

http://en.wikipedia.org/wiki/Content_management_system, revisado noviembre 2006.

[Web1d, 2006]

Learning management System. Wikipedia, the free encyclopedia.

http://en.wikipedia.org/wiki/Learning_Management_System, revisado noviembre 2006.

[Web1e, 2006]

Learning Content Management System. Wikipedia, the free encyclopedia.

http://en.wikipedia.org/wiki/Learning_content_management_system, revisado noviembre 2006.

[Web2, 2006]

Carrasco Dávila Alan F. *La telesecundaria en México*

<http://www.redespecialweb.org/ponencias6/originales6/alantelesec.doc>, revisado septiembre 2006

[Web3, 2006]

La palabra digital. Entrevista con la lic. Alejandra Ortiz Boza

http://comunidad.uls.edu.mx/public_html/publicaciones/onteanqui/b13/pal-digital.htm, revisado septiembre 2006.

[Web3b, 2006]

Universidad Tecnológica de la Mixteca.

http://www.utm.mx/Nuestra_universidad/historia.html, revisado diciembre 2006.

[Web4, 2006]

Blackboard.

<http://www.blackboard.com/>, revisado diciembre 2006.

[Web5, 2006]

Blackboard Academic Suite. Descripción general de las capacidades del producto.

http://library.blackboard.com/docs/AS/Blackboard_Academic_Suite_Capabilities_SP.pdf, revisado diciembre 2006.

[Web6. 2006]

Student Computing. Moodle (My Courses)

<http://www.augsburg.edu/stucomp/moodle/>, revisado 2006.

[Web7, 2006]

Student Manual 1.54

<http://moodle.token.fi/mod/book/print.php?id=5119> revisado diciembre 2006

[Web8, 2006]

Moodle Doc. Características

http://docs.moodle.org/es/Caracter%C3%ADsticas#M.C3.B3dulo_Cuestionario
revisado noviembre 2006

[Web8a, 2006]

Hot Potatoes. Introducción.

<http://www.aula21.net/segunda/hotpotatoes.htm>, revisado diciembre 2006

[Web8b, 2006]

Hot Potatoes. Home Page

<http://hotpot.uvic.ca/>, revisado diciembre 2006

[Web8c, 2006]

Hot Potatoes. half-Backed Software.Tutoriales.

http://platea.pntic.mec.es/~iali/CN/Hot_Potatoes/intro.htm, revisado
diciembre 2006

[Web9, 2006]

Moodle: Tema 1- Sistema de Calificación, Revisión, evaluación y calificaciones.

<http://blog.pucp.edu.pe/item/3754>, revisado noviembre 2006

[Web9a, 2006]

Claroline: Open source e-learning

<http://www.claroline.net>, revisado noviembre 2006

[Web9b, 2006]

Addlink. Software científico. Maple 10.

<http://www.addlink.es/productos.asp?pid=512>, revisado noviembre 2006

[Web9c, 2006]

KnowledgePresenter®2006.

<http://www.knowledgepresenter.com/>, revisado diciembre 2006

[Flash 2007]

Flash (macromedia)

<http://www.shambles.net/pages/school/Flash/>, revisado enero 2007

[Web10, 2006]

Sherentov IPN, Leonid B. *Proyecto Eva, espacios virtuales de aprendizaje*
http://internet2.dgsca.unam.mx/seminario_nov99/ponencias/edu-distancia/eva_ipn/SHEREMETOVLeonid.html, revisado diciembre 2006
[Web11, 2006]

SOFIA. *Entrevista con el doctor José Luis Díaz de León coordinador del proyecto.*
<http://www.ciberhabitat.gob.mx/academia/proyectos/sofia/>, revisado noviembre 2006

[Web12, 2006]

Bajar, Símsolo Raquel; et. al. *SOFÍA (Sistema Orientado Fundamentalmente a la Individualidad del Aprendizaje) Un nuevo paradigma de la educación virtual en el IPN.* <http://www.auip.org/archivos/sofia.pdf>, revisado octubre 2006

[Web13, 2006]

SEP Subsecretaría de Educación E Investigación Tecnológica. *Sistema de evaluación en línea del examen de ingreso a la educación media superior tecnológica.*
<http://www.sep.gob.mx/work/resources/LocalContent/64337/1/4csepi46.doc>
revisado noviembre 2006

[Web14, 2006]

C++ Tutorial.
<http://www.cplusplus.com/doc/tutorial/tut5-1.html>, revisado septiembre 2006

[Web15, 2007]

W3C. *Cascading Style Sheets level 2. CSS2 Specification*
<http://www.w3.org/TR/1998/REC-CSS2-19980512/>, revisado enero 2007

[Web15b, 2007]

Mathematical Markup Language (MathML) 1.01 Specification. Introduction.
<http://www.w3.org/TR/REC-MathML/chapter1.html>, revisado enero 2007

[Web16, 2007]

Secretaría de Educación Pública. Administración Federal de Servicios Educativos en el Distrito federal. Consulta de Calificaciones
<http://www.afsedf.sep.gob.mx/SIEBDF01/Calif/calif000.jsp>, revisado enero 2007

[Web17, 2007]

Resumen Ejecutivo. DSE-IPN
<http://www.sep.gob.mx/work/resources/LocalContent/64331/1/1csepi01.doc>
Revisado enero 2007

[Web18, 2008]

What is unicode. <http://www.unicode.org/standard/WhatIsUnicode.html>,
Revisado abril 2008

[Web19, 2008]

¿Webquest una nueva estrategia? Generador de Exámenes Matemáticos.
MATEMÁTICAS IES.

<http://lourdes.bitacoras.com/archivos/2006/06/22/generador-de-examenes-de-matematicas>, revisado en mayo 2008

Revisado abril 2008

[Web20, 2008]

Juegos. Generador de Exámenes.

http://www.ibertalleres.com/guias/guia_palencia/Juegos/juegos.htm

Revisado abril 2008

[Web21, 2008]

Avaluator 3.6

<http://www.portalprogramas.com/descargar-AVALUATOR.html>

Revisado abril 2008

[Web22, 2008]

topshareware.com. Random test Generator-PRO 8.2

<http://www.topshareware.com/Random-Test-Generator-PRO-transfer-12255.htm>

Revisado abril 2008

Anexo I

Componente aevcap1.htm

```
//*****
// archivo aevcap1.htm contiene el examen de autoevaluación
// "clásico" (preguntas de falso/verdadero) y "modificado"
//*****

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/strict.dtd">

<html xmlns="http://www.w3.org/TR/xhtml1">
<head>

<link rel="stylesheet" href="../intro/eva.css" type="text/css">
<script src="M2.js">
</script>

</head>

<body background = "img1/fon1.gif">

<div align = "center">
<p> <img src = "img1/tit1.gif" /></p>


</div>

<p><b>Conteste con Verdadero o Falso</b></p>

<form name="form1">

<!-- Examen de respuesta binaria activa el programa de -->
<!-- autoevaluación llamado aquí "clásico" pues sólo realiza -->
<!-- la contabilidad de aciertos y errores -->

<ol>
<li><p>Una técnica de recursos humanos es una política o
un programa concreto que ayuda a avanzar hacia el objetivo
estratégico de la empresa.</p>
<input type="radio" name = "r1" value="1"> VERDADERO<br />
<input type="radio" name = "r1" value="0"> FALSO <br /><br />
```

<p>Una de las paradojas del potencial de desarrollo encarnado en el conocimiento científico y tecnológico aplicado a los sistemas productivos, ha sido el de sus impactos negativos en el empleo.</p>

<input type="radio" name="r2" value="1"> VERDADERO

<input type="radio" name="r2" value="0"> FALSO

<p>Las exigencias del entorno son fuerzas externas a la empresa que afectan a sus resultados pero que están fuera del control de los directivos.</p>

<input type="radio" name="r3" value="1"> VERDADERO

<input type="radio" name="r3" value="0"> FALSO

<p>La planificación estratégica de recursos humanos se refiere a la utilización deliberada por parte de la empresa de los recursos humanos para lograr o mantener una ventaja sobre sus competidores.</p>

<input type="radio" name="r4" value="1"> VERDADERO

<input type="radio" name="r4" value="0"> FALSO

ANEXO I

<p>Fomentar la participación de los directivos de línea es beneficioso para la planificación estratégica de recursos humanos.</p>

<input type="radio" name="r5" value="1"> VERDADERO

<input type="radio" name="r5" value="0"> FALSO

<p>El derecho de los empleados es un ejemplo de elección estratégica de RRHH.</p>

<input type="radio" name="r6" value="1"> VERDADERO

<input type="radio" name="r6" value="0"> FALSO

<p>La relativa contribución de una estrategia de RRHH a los buenos resultados de una empresa aumenta cuanto menor sea el grado en que la estrategia de RRHH permite a la empresa sacar provecho de sus competencias distintivas.</p>

<input type="radio" name="r7" value="0"> VERDADERO

<input type="radio" name="r7" value="1"> FALSO

<p>La formulaciÓn e implantaciÓn de estrategias que hacen una empresa relativamente autÓnoma, incluso si forma parte de una gran corporaciÓn se denomina estrategia corporativa.</p>

<input type="radio" name="r8" value="0"> VERDADERO

<input type="radio" name="r8" value="1"> FALSO

<p>Un clima emprendedor es una de las dos dimensiones principales de la cultura de la empresa cuando se implanten las estrategias de RRHH.</p>

<input type="radio" name="r9" value="1"> VERDADERO

<input type="radio" name="r9" value="0"> FALSO

<p>La auditoria de RRHH implica a menudo la evaluaciÓn del propio departamento de RRHH.</p>

<input type="radio" name="r10" value="1"> VERDADERO

<input type="radio" name="r10" value="0"> FALSO

<!-- *****Funciones JavaScript ***** -->

<input type="button" value="EVALUAR", onClick="Estadist(0,20,2)">

<input type="reset" value="REINTENTAR", onClick="RESET(0,20)">

<!-- ***** -->

</form>

<hr size="3" color="#5b97af" />

SELECCIÓn MÚLTIPLE.

<p> Marque la opciÓn que considere correcta.


```

<!-- Ahora viene el examen de selecci3n m3ltiple -->
<!-- Que activar3 la funci3n Estadist2(). el programa -->
<!-- llamado aqu3 "modificado" pues adem3s de contabilizar-->
<!-- aciertos y errores, retroalimenta -->

```

```

<form name = "form2">
  <ol>

```

```

    <li><p>La utilizaci3n deliberada por parte de la empresa de los
    recursos humanos para lograr o mantener una ventaja sobre sus
    competidores en el mercado es:</p>

```

```

    <input type="radio" name ="res1" value="1">una estrategia de recursos
    humanos<br />

```

```

    <input type="radio" name ="res1" value="0">un programa de recursos
    humanos<br />

```

```

    <input type="radio" name ="res1" value="0">una t3cnica de
    recursos humanos<br /><br />

```

```

    <li><p>¿Cu3l de los siguientes desaf3os se considera un
    desaf3o del entorno?</p>

```

```

    <input type="radio" name ="res2" value="0">Reestructuraci3n
    organizativa<br />

```

```

    <input type="radio" name ="res2" value="1">Evoluci3n del trabajo y
    del papel de la familia<br />

```

```

    <input type="radio" name ="res2" value="0">Empowerment<br /><br />

```

```

    <li><p>¿Cu3l de los siguientes desaf3os obliga a una
    empresa a destinar parte de su presupuesto a la capacitaci3n de
    sus trabajadores?</p>

```

```

    <input type="radio" name ="res3" value="1">Carencia de
    calificaci3n entre los empleados <br />

```

```

    <input type="radio" name ="res3" value="0">Rapidez de los cambios<br />

```

```

    <input type="radio" name ="res3" value="0">Legislaci3n<br /><br />

```

```

    <li><p>La contribuci3n de una estrategia de RRHH al rendimiento
    de una empresa aumenta cuando:</p>

```

☐cuanto m s se ajuste la estrategia de RRHH a la estrategia general organizativa de la empresa

☐cuanto menor sea el ajuste entre la estrategia de RRHH con el entorno en el que la empresa est ; operando

☐cuanto menos se amolde la estrategia de RRHH a las exclusivas caracter sticas organizativas

 Cu l de las siguientes opciones se considera un objetivo de la planificaci n estrat gica de recursos humanos?<p>

☐Fomento del comportamiento reactivo frente al proactivo

☐Refuerzo de la estrategia de la empresa en su conjunto

☐Identificaci n de las oportunidades y debilidades de los RRHH

<p> Cu l de las siguientes opciones no forma parte de los tipos de estrategia de negocio de Porter?</p>

☐Liderazgo en costos

☐Diferenciaci n

☐Estrategia de defensa

<p> Cu l de los siguientes forma parte del pensamiento estrat gico de RRHH?</p>

☐Entender la naturaleza y los estilos del liderazgo

☐Entender el proceso de planificaci n de negocios estrat gicos

☐Comprender procesos clave como por ejemplo resolver un problema

<p>¿Cuál es las siguientes opciones estratégicas de RRHH es básica para el éxito de un empleado con una organización?

<input type="radio" name="res8" value="1">Contratación

<input type="radio" name="res8" value="0">Flujos de trabajo

<input type="radio" name="res8" value="0">Ruptura laboral

<p>A la estrategia del negocio, definida por Miles y Snow como estrategia defensora, se ajusta una estrategia de RRHH basada en:</p>

<input type="radio" name="res9" value="1">El estricto control del trabajo

<input type="radio" name="res9" value="0">La capacitación en creatividad

<input type="radio" name="res9" value="0">Las descripciones generales de los puesto de trabajo

<p>Usted definiría a una empresa que se dedica a la publicidad como</p>

<input type="radio" name="res10" value="1">Buscadora

<input type="radio" name="res10" value="0">Defensora

<input type="radio" name="res10" value="0">Basada en el liderazgo en costos

<!-- ***** Funciones JavaScript ***** -->

<input type="button" value="EVALUAR", onClick="Estadist2(1,30,3)">

<input type="reset" value="REINTENTAR", onClick="RESET2(1,30)">

<!-- ***** -->

</form>

</body>

</html>

Anexo II

Componente M2.js

//***Código del script. Programa escrito en JavaScript
//*****de las funciones Estadist() y Estadist2()**

```

<!--
Buenas = 0;
Malas = 0;
var ERRORES = new Array();

var ERRORES2 = new Array();

var VENT1;

var RESP = new Array(30);


var texto="";
var texto2 = "";


function Estadist(nf,np,op)
{

i= 0;
k=0;
do{
  for(j=0;j<op;j++)
    if(document.forms[nf].elements[i+j].checked)
    {

      if(document.forms[nf].elements[i+j].value ==1)
      { Buenas++;
        break;}
      {Malas++;
        ERRORES[k]= Math.round(i/op)+1;
        k++;
        break;

      }
    }
}

```

II2

```

    }
    i=i+op;

    }while(i<np);
if (Buenas <10)
{
    texto ="Error en la pregunta \n";

    for(n=0; n<k; n++)
    texto+= ERRORES[n]+",";
    alert("BUENAS:"+Buenas+"\n"+"Malas:"+Malas+"\n"+texto +" \n");}
    else
    //
    alert("MUY BIEN");

    Buenas =0;
    Malas =0;

}

function Estadist2(nf,np,op)
{
    RESP[0] = " \n"

    RESP[1] = "Usted marc\u00f3 como correcta la respuesta b). Incorrecta.
    Un programa de recursos humanos se refiere a un medio de ayuda a lograr
    las metas estrat\u00e9gicas de la organizaci\u00f3n. La respuesta
    correcta es la a).\n"

    RESP[2] = "Usted marc\u00f3 como correcta la respuesta c). Incorrecta.
    Una t\u00e9cnica de recursos humanos es una pol\u00edtica particular
    de recursos humanos o un programa que ayude a la obtenci\u00f3n de las
    metas estrat\u00e9gicas de la organizaci\u00f3n. La respuesta correcta es
    la a).\n"

    RESP[3] = "Usted marc\u00f3 como correcta la respuesta a). Incorrecta. La
    reestructuraci\u00f3n organizativa pertenece a los desaf\u00edos
    organizacionales. La respuesta correcta es la b).\n"

    RESP[4] = " \n"

```

RESP[5] = "Usted marc\u00f3 como correcta la respuesta c). Incorrecta. El Empowerment es un desaf\u00edo individual. La respuesta correcta es la b).\n"

RESP[6] = " \n"

RESP[7] = "Usted marc\u00f3 como correcta la respuesta b). Incorrecta. Aunque la rapidez de los cambios requiere la capacitaci\u00f3n del empleado, no afecta demasiado al presupuesto para capacitaci\u00f3n. La respuesta correcta es la a).\n"

RESP[8] = "Usted marc\u00f3 como correcta la respuesta c) Incorrecta. La legislaci\u00f3n no afecta en la actualidad a la capacitaci\u00f3n de manera especial. La respuesta correcta es la a).\n"

RESP[9] = " \n"

RESP[10] = "Usted marc\u00f3 como correcta la respuesta b). Incorrecta. Si la estrategia de RRHH no se ajusta al entorno, entonces no hay consistencia con la estrategia del negocio y la estrategia de RRHH. La respuesta correcta es la a).\n"

RESP[11] = "Usted marc\u00f3 como correcta la respuesta c). Incorrecta. Si la estrategia de RRHH no se amolda a las caracter\u00edsticas exclusivas de la organizaci\u00f3n, no existe consistencia para mantener esas caracter\u00edsticas exclusivas. La respuesta correcta es la a).\n"

RESP[12] = "Usted marc\u00f3 como correcta la respuesta a). Incorrecta. Incluso se pudiera considerar, pero al rev\u00e9s: fomento del comportamiento proactivo frente al reactivo. La respuesta correcta es la c).\n"

RESP[13] = "Usted marc\u00f3 como correcta la respuesta b). Incorrecta. El refuerzo de la estrategia de la empresa en su conjunto es un planteamiento demasiado vago. La respuesta correcta es la c).\n"

RESP[14] = " \n"

RESP[15] = "Usted marc\u00f3 como correcta la respuesta a). Incorrecta. Liderazgo en costos s\u00ed forma parte de la estrategia de negocios de Porter. La respuesta correcta es la c).\n"

RESP[16] = "Usted marc\u00f3 como correcta la respuesta b). Incorrecta. La diferenciaci\u00f3n s\u00ed forma parte de la estrategia de negocios de Porter. La respuesta correcta es la c).\n"

RESP[17] = " \n"

RESP[18] = "Usted marc\u00f3 como correcta la respuesta a). Incorrecta. Ello forma parte de las actividades clave de la ARH, pero no de su pensamiento estrat\u00e9gico. La respuesta correcta es la b)\n"

RESP[19] = " \n"

RESP[20] = "Usted marc\u00f3 como correcta la respuesta c). Incorrecta. La ARH debe comprender los procesos clave como el enunciado, pero no forma parte de su pensamiento estrat\u00e9gico. La respuesta correcta es la b)\n"

RESP[21] = " \n"

RESP[22] = "Usted marc\u00f3 como correcta la respuesta b). Incorrecta. El flujo de trabajo no es b\u00e9lico para el \u00e9xito de un empleado dentro de la organizaci\u00f3n. La respuesta correcta es la a)\n"

RESP[23] = "Usted marc\u00f3 como correcta la respuesta c). Incorrecta. La ruptura laboral es un proceso no deseado por los costos que acarrea. La respuesta correcta es la a)\n"

RESP[24] = " \n"

RESP[25] = "Usted marc\u00f3 como correcta la respuesta b). Incorrecta. La capacitaci\u00f3n en creatividad es indispensable en las organizaciones con estrategias basadas en la innovaci\u00f3n y la flexibilidad. La respuesta correcta es la a)\n"

RESP[26] = "Usted marc\u00f3 como correcta la respuesta c). Incorrecta. Las descripciones generales de los puestos de trabajo son indispensables en las organizaciones con estrategias basadas en la innovaci\u00f3n y la flexibilidad. La respuesta correcta es la a)\n"

RESP[27] = " \n"

RESP[28] = "Usted marc\u00f3 como correcta la respuesta b). Incorrecta. Una empresa dedicada a la biotecnolog\u00eda es una empresa buscadora debido a la necesidad de desarrollar nuevos productos. La respuesta correcta es la a)\n"

RESP[29] = "Usted marc\u00f3 como correcta la respuesta c). Incorrecta. Dado la importancia del desarrollo de nuevos productos, este tipo de empresa no puede basar su estrategia en el estricto control de costos. La respuesta correcta es la a)\n"

```

i= 0;
k=0;
do{
  for(j=0;j<op;j++)
    if(document.forms[nf].elements[i+j].checked)
    {

      if(document.forms[nf].elements[i+j].value ==1)
      { Buenas++;
        break;
      }
      else
      {Malas++;
        ERRORES[k]= Math.round(i/op)+1;
        ERRORES2[k] = RESP[i+j];
        k++;
        break;
      }
    }
  i=i+op;

}while(i<np);

```

```

if (Buenas <10)
{

texto ="Error en la pregunta\n";

for(n=0; n<k; n++)
{texto += ERRORES[n]+",";

texto2 += "<p>" + "P" + ERRORES[n] + "." + " " + "<font color =
'red'><b>RESPUESTA INCORRECTA.</b></font>" + " " + ERRORES2[n] +
"</p> <hr width='400' />";

}

var Marcas = "<html>" + "<script> var vent_ab = true; </script>";
VENT1=
window.open("reval.htm","VENT1","width=500,height=200,scrollbars=1");
VENT1.document.writeln(Marcas + "<body background = 'back1.gif'>");

```


II6

```

VENT1.document.writeln("<b>Buenas: </b>" + Buenas + "<br />" + "<b>Malas:
</b>" + "&#160;" + Malas + "<br />" + "<font color = 'blue'><b>" + texto +
"</b></font>" + "<br /><br />" + texto2);
VENT1.document.writeln("</body></html>");
VENT1.document.close();

```

```

}
else
//
alert("MUY BIEN.\n Respuestas Correctas");

```

```

Buenas =0;
Malas =0;

```

```

}

```

```

function RESET(nf, n)
{
for(j=0;j<n;j++)
document.forms[nf].elements[j].checked=false;
}

```

```

function RESET2(nf, n)
{
for(j=0;j<n;j++)
document.forms[nf].elements[j].checked=false;
self.location = "aevcap1.htm";
}

```

```

function VENT_AB()
{
if(VENT1.vent_ab == true)
VENT1.close();

```

```

}
//-->

```

Anexo III

Componente exams.htm

```
//*****

//Código HTML de la primera página Web correspondiente al
// Generador aleatorio de exámenes matemáticos, contiene los
// vínculos a los exámenes.

//*****

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/strict.dtd">

<html xmlns="http://www.w3.org/TR/xhtml1">
<head>
<link rel =stylesheet href="../code/hoja1.css" type ="text/css">
<style>

td{font-family:arial;                font-size:11pt;padding-left:10;padding-
right:10;padding-top:10;padding-bottom:10;}
</style>
</head>
<body background = "../tapiz/tapiz7.jpg" link="black" >
<map name = "NAVEGA">

<area shape = "rect" coords = "78,10,118,22" a href = "../index.htm">
<area shape = "rect" coords = "510,10,547,23" a href =
"../acerca/mapav.htm">

</map>

<table align= "center">
<tr>

<td>


</td>

</tr>
</table>

<table align = "center">
```

III2

```
<tr>
<td>

</td>
</tr>
</table>
```

```
<div align = "center">
<img src = "../../examen/texam.jpg">

</div>
```

```
<table align = "center" bgcolor= "#99FFFF" border = "1" width="700">
<tr>
<td>
<div align = "justify">
<p>Antes de intentar acceder al <b>GENERADOR DE
EXAMENES</b>, se debe haber realizado y entregado previamente
las tareas que se encuentran al final de cada sección
temática, local;celas f;ilmente con ayuda del
<b>mapa</b>.</p>

<p>Al acceder a la página del número de examen
seleccionado, y pulsar el botón correspondiente el <b>GENERADOR DE
EXAMENES</b> le presentará; dos preguntas.</p>
<p> <b>Deberán contestarse y entregarse dos preguntas de cada
examen.</b></p>
</div>
</td>
</tr>
</table>
```

```
<table align = "center">
<tr>
<td>

</td>
</tr>
</table>
```

```
</br>
<table align = "center">
<tr>
```

```
<td bgcolor="#FFFFCC" align = "center">
<a href="../tareas/tarea1/tar1.htm">
<b> EXAMEN I</b>
</a>
</td>
</tr>
```

```
<tr>
<td height = "20">
&#160;
</td>
</tr>
```

```
<tr>
<td bgcolor="#99FFCC" align = "center">
<a href="../tareas/tarea2/tar2.htm">
<b>EXAMEN II</b>
</a>
</td>
</tr>
```

```
<tr>
<td height = "20">
&#160;
</td>
</tr>
```

```
<tr>
<td bgcolor="#FFFFCC" align = "center">
<a href="../tareas/tarea3/tar3.htm">
<b>EXAMEN III</b>
</a>
</td>
</tr>
```

```
<tr>
<td height = "20">
&#160;
</td>
</tr>
```

```
<tr>
<td bgcolor="#99FFCC" align = "center">
<a href="../tareas/tarea4/tar4.htm">
<b>EXAMEN IV</b>
</a>
</td>
```

III4

</tr>

<tr>

<td height = "20">

&#160;

</td>

</tr>

<tr>

<td bgcolor="#FFFFCC" align = "center">

EXAMEN V

</td>

</tr>

<tr>

<td height = "20">

&#160;

</td>

</tr>

<tr>

<td bgcolor="#99FFCC" align = "center">

EXAMEN VI

</td>

</tr>

<tr>

<td height = "20">

&#160;

</td>

</tr>

<tr>

<td bgcolor="#FFFFCC" align = "center">

EXAMEN VII

</td>

</tr>

<tr>

```
<td height = "20">
&#160;
</td>
</tr>
```

```
<tr>
<td bgcolor="#99FFCC" align = "center">
<a href="../tareas/tarea8/tar8.htm">
<b>EXAMEN VIII</b>
</a>
</td>
</tr>
</table>
```

```
<p>&#160;</p>
```

```
<table align = "center">
<tr>
<td>

</td>
</tr>
</table>
```

```
<table align= "center">
<tr>
```

```
<td>
```

```
<IMG src="../imag/botones/barra14.jpg" border = "NO" usemap =
"#NAVEGA">
</td>
```

```
</tr>
</table>
```

```
</body>
</html>
```


Anexo IV

Componente tar_x.htm

```

<!-- ***** -->
<!-- Archivo tar1.htm -->
<!-- Código de la segunda página Web del Generador Aleatorio -->
<!-- de Exámenes Matemáticos correspondiente a un examen -->
<!-- particular muestra las referencias a los sripts -->

<!-- ***** -->
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/strict.dtd">

<html xmlns="http://www.w3.org/TR/xhtml1">
<head>
<title> </title>
<link rel =stylesheet href="../../code/hoja1.css" type = "text/css">
<style>

td{padding-left:10;padding-right:10; padding-top=10;}
</style>
<script language = "JavaScript" src="../../code/base1.js">
</script>
<script language = "JavaScript" src="../../code/pazar.js">
</script>
<script language = "JavaScript" type="text/javascript">

</script>

</head>
<body background = "../../tapiz/tapiz6.jpg">

<table align = "center">
<tr>
<td>
<div align = "center">

</div>
</td>
</tr>
</table>

<div align = "center">


```



```

<br>
</div>

<font face="tahoma" size="3" color="blue">
<p><b>Primer Examen</b></p>
</font>
<table align = "center" border = "2" bordercolor = "black" width="700">
<tr>
<td bgcolor = "#99FFFF">
<div align = "justify">
<font face="arial" size ="2">
<p><b>AL PULSAR EL BOT&Oacute;N <i>"GENERAR"</i>
OBTENDR&Aacute; UN EXAMEN. SI DESEA REALIZAR OTRO EXAMEN,
CIERRE LA VENTANA Y VUELVA A PULSAR <i>"GENERAR"</i>. EL
EXAMEN ES ALEATORIO.</p>&#160;
</font>

</div>
</td>
</tr>
</table>

```

```

<p>&#160;</p>

```

Pulse para "GENERAR" para seleccionar examen:

```

<p>&#160;</p>
<p>&#160;</p>

```

```

<!-- *****Activación por evento de las funciones JavaScript -->
<!-- *****que presentarán la ventana con la página *****-->

```

```

<div align = "center">
<form>
<input type="button" value="GENERAR"
onClick="Examen(Preguntas(),Azar(2,16))">
</form>
</div>
<!-- ***** -->

```

```
<div align = "center">  
  
</div>
```

```
</body>  
</html>
```

ANEXO V

Función Preguntas del componente bas1.js

```
//*****Archivo bas1.js referenciado mediante
//*****<script language="JavaScript" src="../../code/base1.js">
```

```
<!--
```

```
function Preguntas()
{
```

```
    var P= Array(16);
```

```
P[0] = "<p><b>1.</b> Exprese el vector &#160; ( 2, -4, 1 ) &#160; como
<b>combinaci&oacute;n lineal</b> de los vectores: </p> <p><div
align='center'>( 3, 2, 1 ); ( -2, 0, 3 ) &#160; y &#160; ( -4, 3, 0)</div></p>";
```

```
P[1]= "<b>2.</b> &#191;El vector &#160; ( <img src='1_2.gif' align='center'
/>, 2 ) &#160; es <b>combinaci&oacute;n convexa</b> de los vectores
&#160; ( 3, 0 ) &#160; y &#160; ( 0, <img src='3_2.gif' align='center' /> ) ?
<br>Justifique la respuesta.<p>";
```

```
P[2]= "<b>3.</b> Dado el <b>hiperplano</b>: &#160;  $2x_1 - 2x_2 + 4x_3 + 6x_4 = 7$  &#160;
&#191;en qu&eacute; <b>semiespacio</b> est&aacute; el vector &#160; (
6, 1, 7, 2 ) ?<p>";
```

```
P[3]="<p><b>4.</b> &#191;Cu&aacute;les de los siguientes conjuntos son
<b>convexos</b>? ¡justif&iacute;quelos!</p> A<sub>1</sub> = { ( x, y, z )
/ x + y <img src='me_igu.gif' align='center' /> 3; &#160; x - z <img
src='me_igu.gif' align='center' /> 2 }<br> A<sub>2</sub> = { ( x, y ) /
 $3x^2 + 2y^2$  <img src='me_igu.gif' align='center' /> 6 }<br> A<sub>3</sub> = { ( x, y ) / x = 1; &#160; | y | <img
src='me_igu.gif' align='center' /> 4 } ;<br> A<sub>4</sub> = { ( x, y ) / y -
x<sup>2</sup> <img src='ma_igu.gif' align='center' /> 0; &#160; x <img
src='perten.gif' align='center' /> [ -2, 2 ] } </p>";
```

```
P[4]= "<p><b>5.</b> Determine geom&eacute;tricamente el <b>casco
convexo</b> de cada uno de los conjuntos del ejercicio anterior.</p>";
```

P[5]="<p>6. Demuestre que $\mathbb{R}^n$ y cualquier subespacio vectorial de $\mathbb{R}^n$ son conjuntos convexos.</p>";

P[6]="<p>7. Considere el hiperplano $H_0 = \{ x \in \mathbb{R}^n \mid a \cdot x = 0 \}$. Demuestre que H_0 es un subespacio vectorial de $\mathbb{R}^n$; $n - 1$.</p>";

P[7]= "8. Demuestre que si A_i para $i = 1, 2, \dots, n$ son conjuntos convexos, entonces :<p> a La intersección de ellos es un conjunto convexo.
 b La unión de ellos sería; convexo?, si no es así, díe un contraejemplo.<p>";

P[8]="<p>9. Dibuje (haga un trazo correspondiente) el poliedro convexo generado por los siguientes conjuntos de vectores:<p> a $\{ (0, 0); (1, 0); (0, 1); (1, 1) \}$
 b $\{ (3, 4); (5, 0); (8, 12); (12, 5); (7, 3) \}$ ";

P[9]="10. Demuestre que una transformación lineal de $\mathbb{R}^n$ en $\mathbb{R}^m$ transforma un conjunto convexo en un nuevo conjunto convexo.";

P[10]="<p>11. <i>CINCO HERMANOS</i> es una empresa leonesa que fabrica zapatos de modelos exclusivos para caballero: botas, botines y pantuflas. El gerente de producciín tiene que decidir el mejor programa de producciín para el mes que sigue, teniendo en consideraciín ciertos requerimientos financieros y de producciín, pero que se tenga las mejores utilidades para la empresa. No existe límites en la disponibilidad de piel; sin embargo, se dispone de un mínimo de 1680 horas de producciín para el mes. El tiempo de producciín cuesta 100 pesos por hora, cada unidad de piel tiene un costo de 40 pesos. La empresa hace todas sus ventas a mayoristas que le pagan

en efectivo toda la producción. Los precios de venta al mayorista son 1000; 750 y 500 pesos respectivamente para los tres modelos. Los costos fijos de operación de la empresa para el mes siguiente son de diez mil pesos y su saldo actual de efectivo de 128, 250 pesos. El gerente informa que ya tiene pedidos para los diferentes modelos: 35 pares de botas, 60 pares de botines y 40 pares de pantuflas y está seguro que puede venderse todos los pares que se fabriquen durante el mes que excedan a los pedidos comprometidos, en consecuencia no se tendrán inventarios. El cuadro siguiente describe los datos de producción:

Modelos	Horas (Operación)	Unidades (piel)
Botas	3.50	4.50
Botines	2.50	3.25
Pantuflas	2.00	2.00

P[11]="<p>12. <i>DESCANSO FELIZ</i> se especializa en la fabricación de colchones, fabrica tres tipos de colchones: matrimonial, king-size e individual. Los tres modelos se fabrican tanto en la Planta de Tlalnepantla como en la Planta de San Luis que son propiedades de la empresa. En un día normal de ocho horas de labores continuas la producción de las plantas se da en el cuadro siguiente:

Planta	Matrimonial
--------	-------------

</div></td><td width=80 bgcolor='#FF9966'><div align='center'><i>King Size</i></div></td><td width=80 bgcolor='#FF6666'><div align='center'><i>Individual</i></div></td></tr><tr><td bgcolor='#FFFFCC'><div align='center'><i>Tlalnepantla</i></div></td><td bgcolor='#99FFCC'><div align='center'>50</div></td><td bgcolor='#FFCC99'><div align='center'>100</div></td><td bgcolor='#FFCCCC'><div align='center'>120</div></td></tr><tr><td align='center'><i>San Luis</i></div></td><td bgcolor='#99FFCC'><div align='center'>100</div></td><td bgcolor='#FFCC99'><div align='center'>80</div></td><td bgcolor='#FFCCCC'><div align='center'>420</div></td></tr><tr><td align='center'>Demanda
(Mensual)</div></td><td bgcolor='#99CCFF'><div align='center'>2960</div></td><td bgcolor='#99CCFF'><div align='center'>3200</div></td><td bgcolor='#99CCFF'><div align='center'>8400</div></td></tr></table></p>. <p>La demanda mensual de los modelos se estimó; en el departamento de Mercadotecnia, se estima que será; de 2960; 3200 y 8400 unidades de cada modelo respectivamente. El departamento de Contabilidad informa que los costos de operación diaria de las plantas son 25000 de Tlalnepantla y 35000 de San Luis. A los administradores les gustaría determinar el número óptimo de días de operación por mes de cada planta con el objeto de minimizar el costo total de producción, pero satisfaciendo la demanda estimada. </p>";

P[12]="<p>13. La compañía <i>Elegancia S.A.</i> fabrica tres productos para caballeros: A, B y C. Los administradores de la compañía deben recuperar doscientos mil pesos de costos fijos y desearían determinar la cantidad de cada uno de los productos que se deben fabricar para que cuando la empresa llegue al punto de equilibrio, la suma de los costos de producción sea mínima. Se sabe que los costos de fabricar los productos son 1200, 1000 y 500

pesos respectivamente; mientras que la venta de éstos se harían en 2500, 1800 y 1000 pesos. También informaron que hay pedidos atrasados que satisfacer antes de surtir pedidos nuevos. Los productos atrasados son de 200 del A, 400 del B y 1200 del C. </p>";

P[13]="<p>14. <i>KIDMAD</i> es una empresa pequeña que fabrica juguetes de madera. Se ha decidido fabricar yo-yos y trompo. Un estudio de mercado sugiere vender cada Yo-Yo en 30 pesos y el trompo en 35 pesos; cada yo-yo costaría fabricarlo 5 pesos, mientras que el trompo costaría 6 pesos. Los productos en el proceso de fabricación pasan por la carpintería y el departamento de acabados, en relación a la materia prima no hay problemas se pueden conseguir todo lo necesario para la producción semanal; sin embargo, sólo dispone de 90 horas en la carpintería y 125 horas en el acabado. En la tabla adjunta se da los tiempos utilizados en cada etapa del proceso, así como las demandas respectivas. </p>

 Juguete 	 <i>Carpintería</i> 	 <i>Acabado</i> 	 <i>Demanda</i>
 <i>Yo-yo</i> 	 0.25 	 0.5 	 200
 <i>Trompo</i> 	 0.5 	 0.5 	 150
 Tiempo 	 90 Horas 		

```

bgcolor='#FFCC99'><div align='center'><font face='arial,garamond'
size='2'> 125 Horas </font></div></td><td><div align='center'><font
face='arial,garamond' size='2'></font> </div> </td></tr></table></div>
</p>";

```

```

P[14]= "<p><b>15.</b> La empresa peque&ntilde;a <i>'Antena
Perfecta'</i> fabrica y vende dos tipos de antena a&eacute;rea: <b>A</b>
y <b>B</b>. La producci&oacute;n de la antena <b>A</b> requiere de
<b>cuatro horas</b> de fabricaci&oacute;n, en tanto que la
fabricaci&oacute;n del tipo <b>B</b> requiere de <b>seis horas</b>. n
la f&aacute;brica existe un total disponible de <b>ciento ochenta horas-
hombre</b> semanales para la producci&oacute;n. Un estudio de
mercado determina que lo m&aacute;ximo que pueden vender en cada
semana es <b>30</b> antenas del tipo <b>A</b> y <b>20</b> antenas
del tipo <b>B</b>. La contribuci&oacute;n neta (despu&eacute;s de
gastos) a las utilidades por cada antena del tipo <b>A</b> ser&aacute; de
<b>ochocientos pesos</b>, mientras que la antena del tipo <b>B</b>
aportar&aacute; a las utilidades en <b>mil quinientos pesos</b>.
&#191;Qu&eacute; cantidad de cada tipo de antena debe fabricar la
mpresa cada semana para maximizar sus utilidades?. </p>";

```

```

P[15]="<p><b>16.</b> La <i>'Cl&iacute;nica EMAR de Sat&eacute;lite'</i>
tiene <b>diez</b> salas quir&uacute;rgicas con todos los equipos
necesarios y tiene clasificados <b>dos</b> tipos de operaciones
seg&uacute;n el tiempo de ocupaci&oacute;n de las salas. Las del <b>tipo
I</b> ocupa en promedio media hora, mientras que las del <b>tipo II</b>
ocupa en promedio <b>dos</b> horas. Las salas pueden estar ocupadas
en promedio <b>diez</b> horas diarias y por cuestiones de seguridad a lo
m&aacute;s se pueden intervenir en <b>ciento veinte</b> operaciones del
<b>tipo I</b> y a lo m&aacute;s en <b>cuarenta y cinco</b> operaciones
del <b>tipo II</b> y por cuestiones internas no menos de <b>treinta</b>
intervenciones de ambas. La cl&iacute;nica recibe <b>500</b> pesos por
cada operaci&oacute;n del <b>tipo I</b> y <b>1500</b> pesos por cada
operaci&oacute;n del <b>tipo II</b>. La cl&iacute;nica
&#191;Cu&aacute;ntas operaciones de cada tipo debe programar
diariamente para maximizar sus ingresos?</font> </p>";

```

```

return P;

```

```

}

```

```

//-->

```


ANEXO VI

Función Azar() del componente pazar.js

//*****Archivo pazar.js

```
<!--
function Azar(num,max)
{ var x;
  var i=0;
  var j;
  var Z = new Array(num);
  do{
    x=Math.random(); // obtiene números entre 0 y 1
    x=x*100;         //se aseguran número hast decenas
    x=Math.round(x); //redondea el valor
    if(x<max) //se seleccionan menores a "max"
    {
      //sólo el primer dato se carga libremente
      if(i==0)
        Z[i++] =x;
      else {
        for(j=0;j<i;j++){
          //sólo se guardan datos que no sean iguales a los ya obtenidos
          if(Z[j]== x){
            //no guarda si son iguales, vuelve a obtener otr
            break;
          }

          if(Z[j]!=x)
            Z[i++] = x;
        }
      }
    }
  }

  }while(i<num); //el total de numeros solicitados

return Z; //obtenemos un arreglo
}

//-->
```

ANEXO VII

Función Examen() del componente exams.js

```
<!--  
  
function Examen(P,Z)  
{ var i;  
  var  
win=window.open("exam2.htm","vent1","width=600,height=400,resizable=1  
,scrollbars=1,left=100,top=250");  
  
win.document.writeln("<html><head><link          rel          =stylesheet  
href='../..'/code/hoja1.css' type  ='text/css'></head><body  background  =  
'../..'/tapiz/tapiz6.jpg'>"+ "<br>" + "<div align  = 'justify'>");  
  
    for(i=0;i<Z.length;i++){  
        win.document.writeln(P[Z[i]]+"<br>");  
win.document.writeln("<div align='center'><hr size='4'></div>");  
        }  
        win.document.writeln("<br>");  
  
win.document.write("</div></body></html>");  
  
}  
  
//-->
```

Manual de Operación

I. Programas de autoevaluación "clásico" y "modificado"

La Figura a, muestra parte de un examen que se evalúa mediante el programa "clásico", una vez llenados las opciones, se activa dicho programa pulsando el botón "Evaluar"; en cambio, el botón "Reintentar" reestablece las condiciones iniciales del examen, se puede realizar la prueba tantas veces se desee, incluso pulsando el botón evaluar, pues como se ha indicado en los capítulos anteriores, **no hay conexión** con alguna base de datos.

Atrás

Búsqueda Favoritos

C:\tesis-CIC\program\autoev\temas\aevcap1.htm

Google

Ir

Marcadores

9 bloqueados

Corrector ortográfico

Configuración

☐ FALSO

8. La formulación e implantación de estrategias que hacen una empresa relativamente autónoma, incluso si forma parte de una gran corporación se denomina estrategia corporativa.

☐ VERDADERO
☐ FALSO

9. Un clima emprendedor es una de las dos dimensiones principales de la cultura de la empresa cuando se implanten las estrategias de RRHH.

☐ VERDADERO
☐ FALSO

10. La auditoria de RRHH implica a menudo la evaluación del propio departamento de RRHH.

☐ VERDADERO
☐ FALSO

EVALUAR REINTENTAR

Figura a. El examen de autoevaluación "clásico"

El programa de evaluación se ha dicho es "clásico" en el sentido de que gran número de plataformas de gestión de contenidos de aprendizaje, contienen éste tipo de programas, en donde el proceso de corrección automatizada sólo aporta a lo más una contabilidad de aciertos. La Figura b muestra el resultado de presentado por el programa "clásico" en una ventana de mensajes, al pulsar "EVALUAR"

Se indica el número de aciertos y errores, así también informa en qué preguntas se eligieron opciones incorrectas.

Al pulsar “EVALUAR” se activa la función JavaScript “Estadist()”, desplegándose la ventana de resultados y explicaciones, como se muestra en la Figura b.

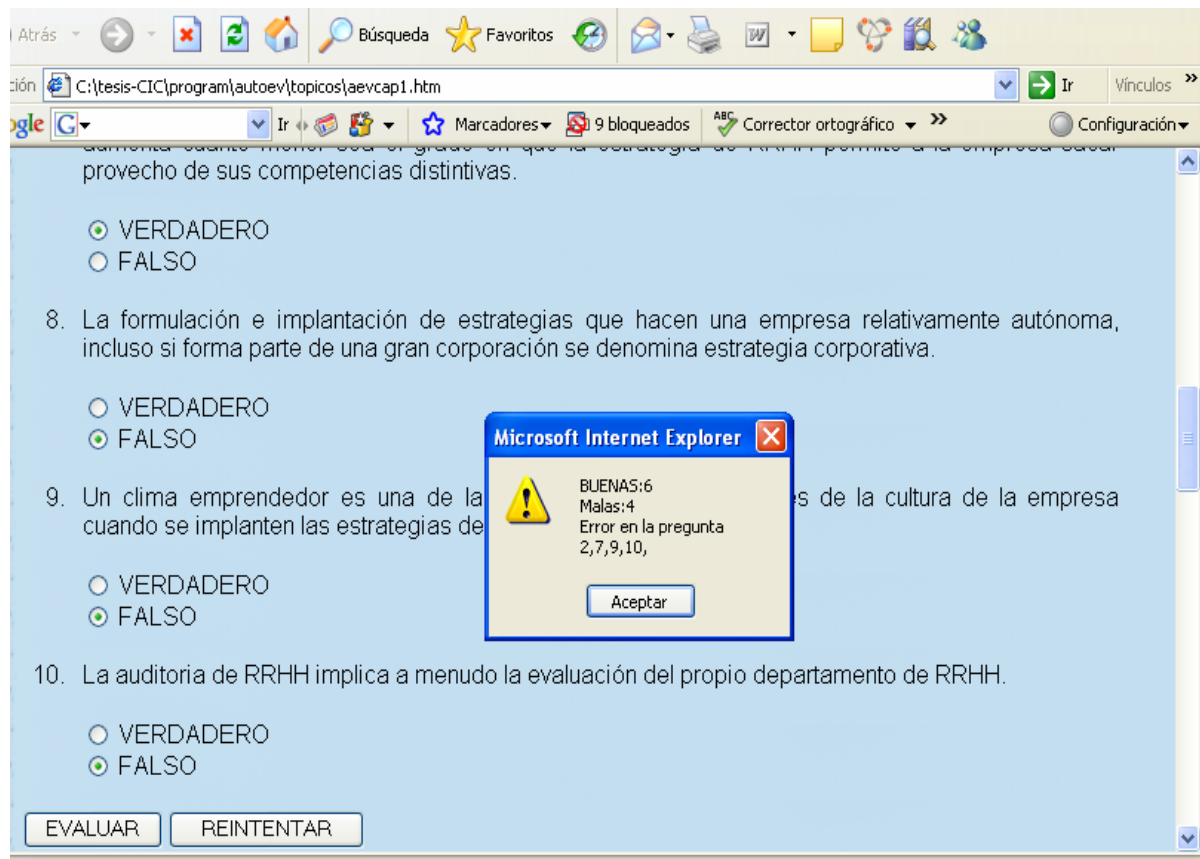


Figura b. La ventana con el resultado de la autoevaluación.

Gran número de sistemas de evaluación automatizada realizan tareas análogas a las que aquí se presentan, aun en el caso de utilizar reactivos de opción múltiple (pruebas pedagógicas con reactivos de más de dos opciones), sólo hay una contabilidad de errores. Se trata de un programa “clásico” de evaluación, con limitantes para el autoaprendizaje.

Los exámenes de opción múltiple con un mayor número de preguntas son una buena oportunidad para proponer mecanismos de retroalimentación de los alumnos, ya que no resulta obvio como en los reactivos del tipo falso/verdadero donde se halla el error; en la versión “modificada” del programa de evaluación los resultados se despliegan en una página Web.

La Figura c, muestra parte de la página Web con un examen con mayor número de opciones, los botones “EVALUAR” y “REINTENTAR”, tienen una actividad análoga al caso “clásico” de corrección automatizada; sin embargo, al pulsar “EVALUAR” se activa la función “Estadist2()”, que

genera la ventana no sólo con la contabilidad de aciertos y errores, así como el número de las preguntas con selección errada, **sino también explicaciones que fundamentan porqué es errónea la opción seleccionada.**

Atrás | Búsqueda | Favoritos | Ir | Vínculos

C:\tesis-CIC\program\autoev\temas\aevcap1.htm

Google | Ir | Marcadores | 9 bloqueados | Corrector ortográfico | Configuración

8. ¿Cuál de las siguientes opciones estratégicas de RRHH es básica para el éxito de un empleado con una organización?

- ☐ Contratación
- ☐ Flujos de trabajo
- ☐ Ruptura laboral

9. A la estrategia del negocio, definida por Miles y Snow como estrategia defensora, se ajusta una estrategia de RRHH basada en:

- ☐ El estricto control del trabajo
- ☐ La capacitación en creatividad
- ☐ Las descripciones generales de los puesto de trabajo

10. Usted definiría a una empresa que se dedica a la publicidad como

- ☐ Buscadora
- ☐ Defensora
- ☐ Basada en el liderazgo en costos

EVALUAR REINTENTAR

Figura c. Examen de opción múltiple para la evaluación automatizada

Al pulsar evaluar se despliega una ventana con la contabilidad de aciertos y explicaciones en torno a la selección errónea. Las potencialidades multimedia no se explotaron; pero es totalmente posible realizar retroalimentaciones de esta naturaleza.

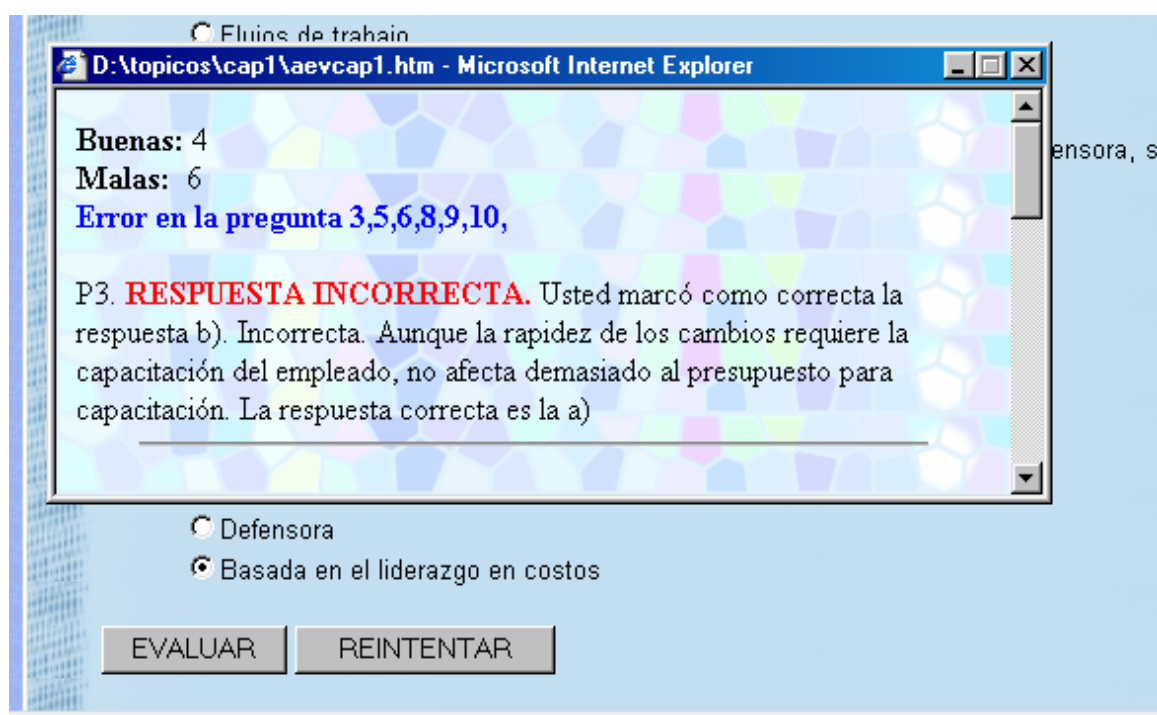


Figura d. Resultados de la autoevaluación y la retroalimentación

II. Generador Aleatorio de Exámenes Matemáticos

La Figura e muestra el despliegue de la página Web que permite seleccionar un tipo de examen, cada letrero "EXAMEN" permite acceder a una página que activará la función generadora del examen correspondiente.

Una de las ideas es presentar obtener el examen bajo la condición de haber completado cierto número de trabajos o tareas; sin embargo no se desarrolló ningún proceso automatizado de verificación de la realización de esta condición por parte del solicitante del examen.

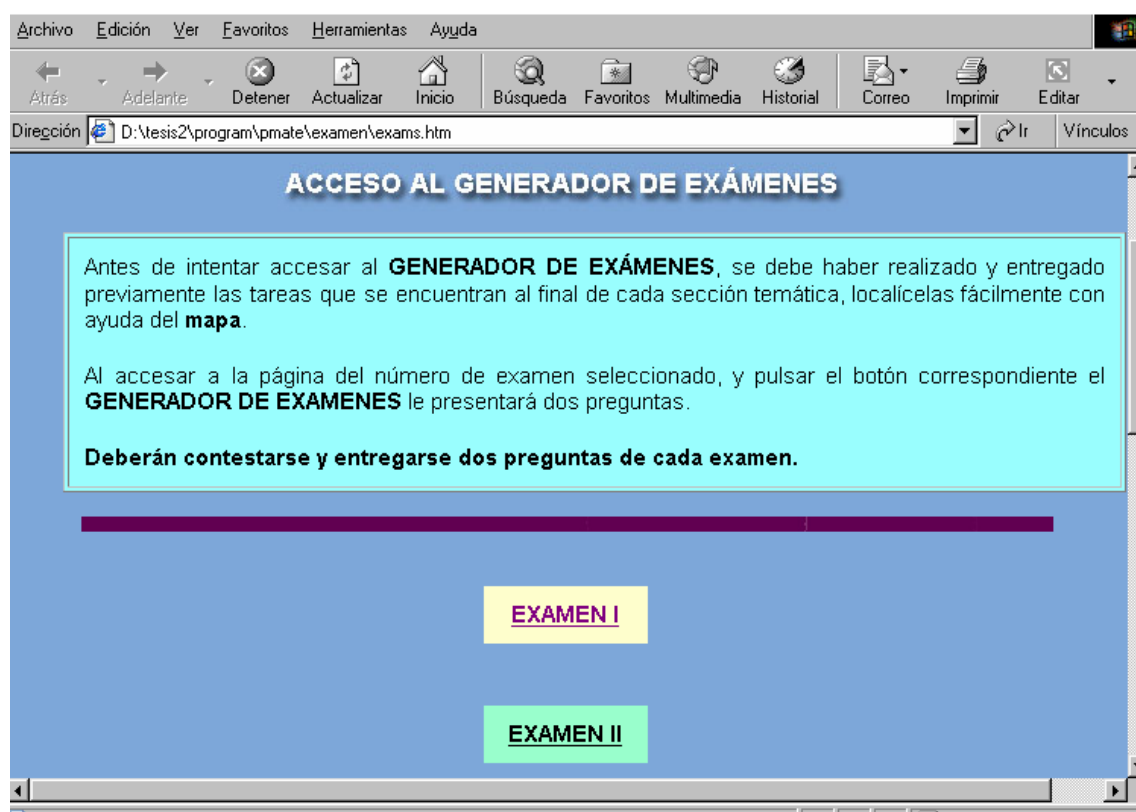


Figura e. Página Web con los accesos a los exámenes

El alumno deberá seleccionar un examen, pulsando los letreros correspondientes (Examen I, Examen II, etc.); una vez realizada esta acción se presenta la página Web de Examen (Figura f), la cual presenta el botón “GENERAR”, que al pulsarse activa la función “Examen(P, Z)” con los parámetros P un arreglo de preguntas y Z los números de preguntas obtenidos al azar.

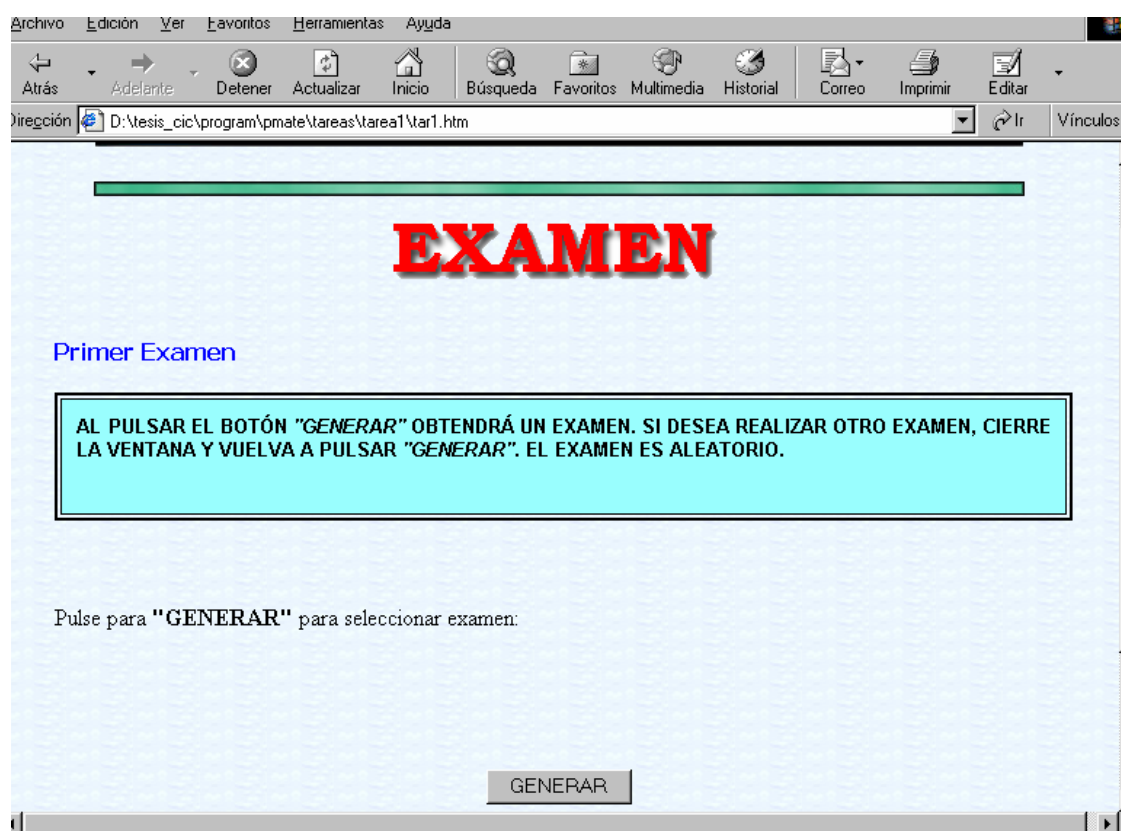


Figura f. Página Web con el botón que activa el GAEM

Los exámenes desplegados mediante el “Generador Aleatorio de Exámenes Matemáticos” (**GAEM**) pueden mostrar tablas y expresiones matemáticas diversas.

Las expresiones matemáticas más complejas que usan por ejemplo operador integral, coeficientes binomiales, matrices, etc; se basan en el despliegue de imágenes en formato GIF, las cuales tiene la propiedad de transparencia. En el capítulo siguiente se hará la crítica respectiva a este proceso; sin embargo al haberlo hecho así el GAEM, demostró su capacidad para abrirse paso a la multimedia, a la posibilidad de incorporar programas de java (applets) y de modelado de realidad virtual.

Las Figuras g y h, muestran dos tipos de exámenes aleatorios obtenidos al activar el “GAEM”.

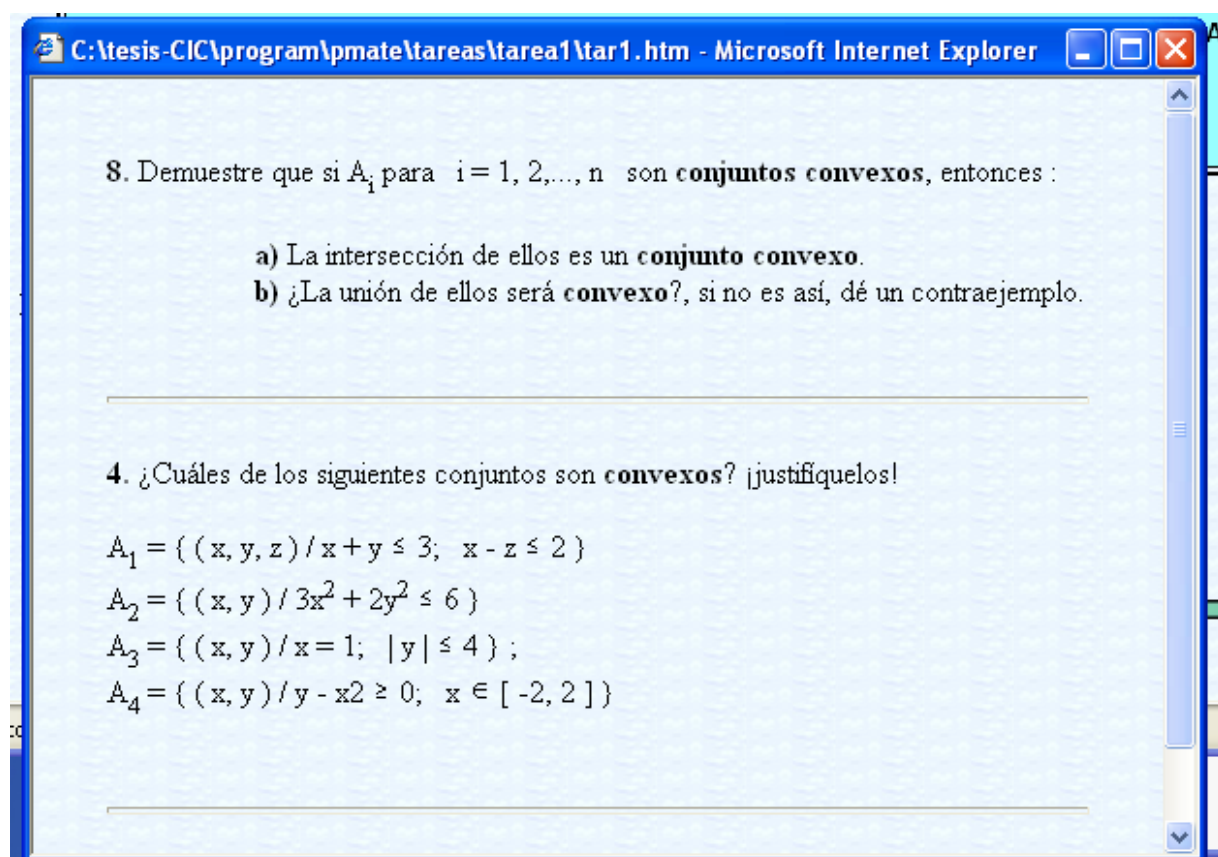
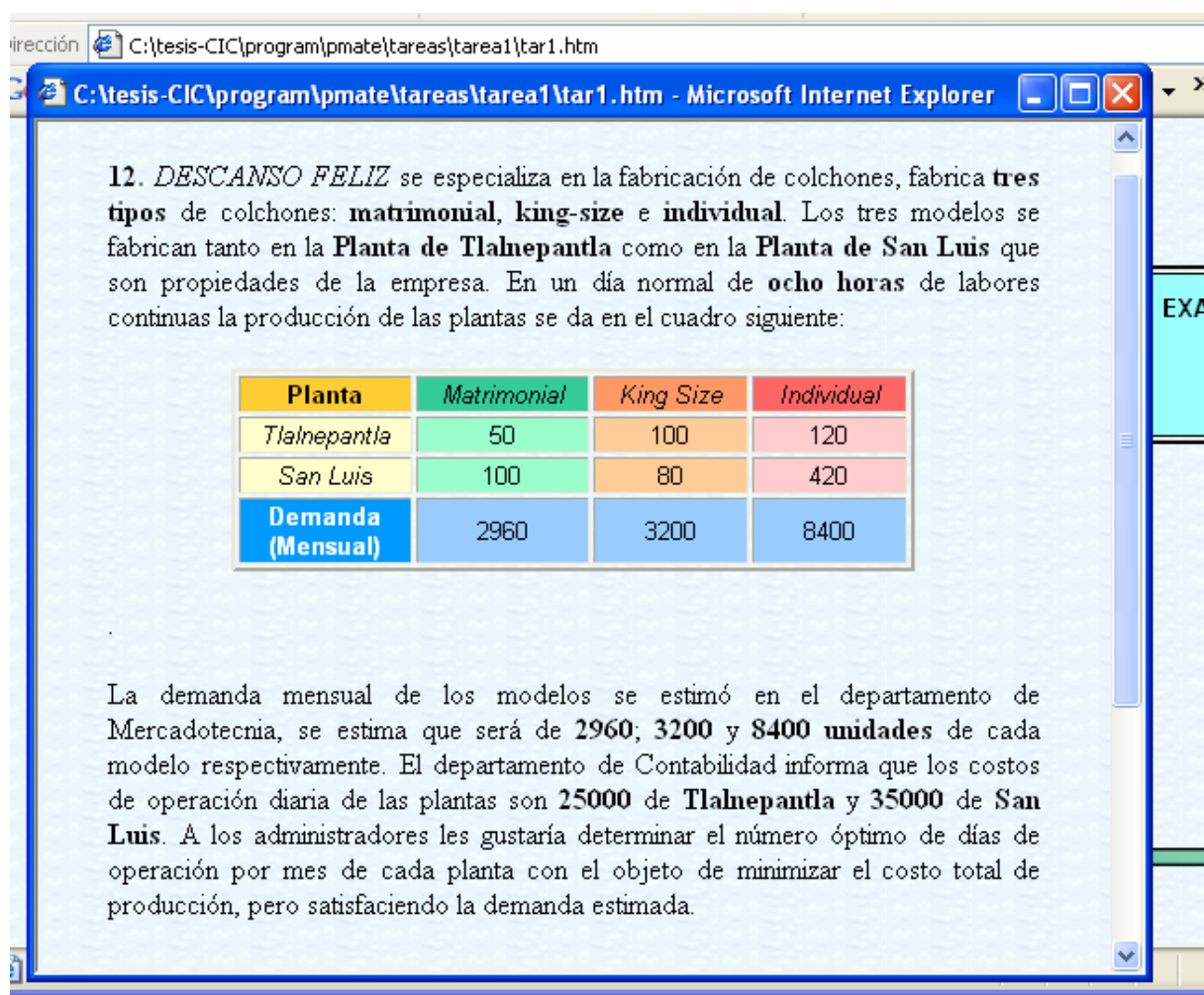


Figura g. Muestra 1 del examen obtenido con el GAEM

Generación de exámenes con tablas coloridas (Figura 6.8), imágenes de diversos grados de complejidad, audio y video, propondrán en el futuro nuevos tipos de exámenes matemáticos.



12. *DESCANSO FELIZ* se especializa en la fabricación de colchones, fabrica **tres tipos** de colchones: **matrimonial**, **king-size** e **individual**. Los tres modelos se fabrican tanto en la **Planta de Tlalnepantla** como en la **Planta de San Luis** que son propiedades de la empresa. En un día normal de **ocho horas** de labores continuas la producción de las plantas se da en el cuadro siguiente:

Planta	Matrimonial	King Size	Individual
Tlalnepantla	50	100	120
San Luis	100	80	420
Demanda (Mensual)	2960	3200	8400

La demanda mensual de los modelos se estimó en el departamento de Mercadotecnia, se estima que será de **2960**, **3200** y **8400 unidades** de cada modelo respectivamente. El departamento de Contabilidad informa que los costos de operación diaria de las plantas son **25000** de **Tlalnepantla** y **35000** de **San Luis**. A los administradores les gustaría determinar el número óptimo de días de operación por mes de cada planta con el objeto de minimizar el costo total de producción, pero satisfaciendo la demanda estimada.

Figura h. Muestra 2 del examen obtenido con el GAEM

El programa GAEM es capaz de desplegar exámenes matemáticos de forma aleatoria, no importa la complejidad de las relaciones matemáticas, asimismo despliega tablas coloridas, incluso applets de Java. En la presente etapa en la cual aún MathM1 no logra superar los problemas de compatibilidad de su visor con las diferentes máquinas; el programa presente, tiene utilidad, en un sistema de evaluación “en línea”.

Como prueba de las posibilidades de generación de exámenes multimedia del GAEM, se muestra parte de un examen interactivo desplegado, en las Figura i y j

5. ¿Cuál de los siguientes conjuntos es convexo?

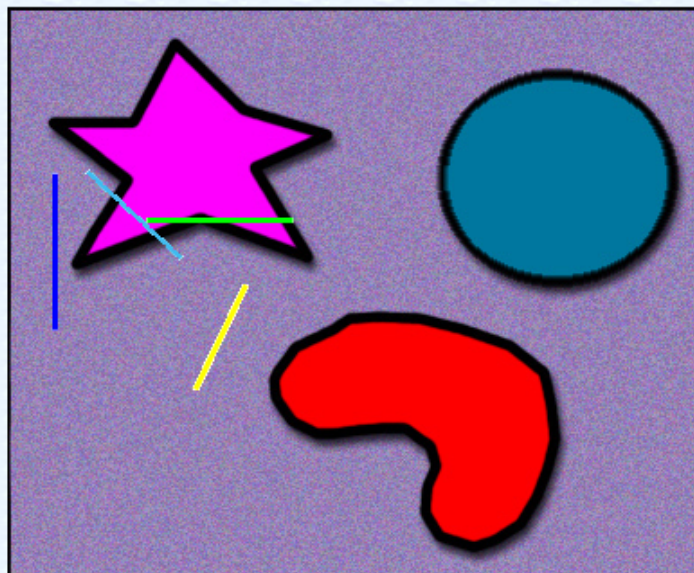


Figura i. Reactivo donde se realiza interactividad con el usuario. Con el apuntador de ratón se desplaza una línea (reactivo multimedia)

5. ¿Cuál de los siguientes conjuntos es convexo?

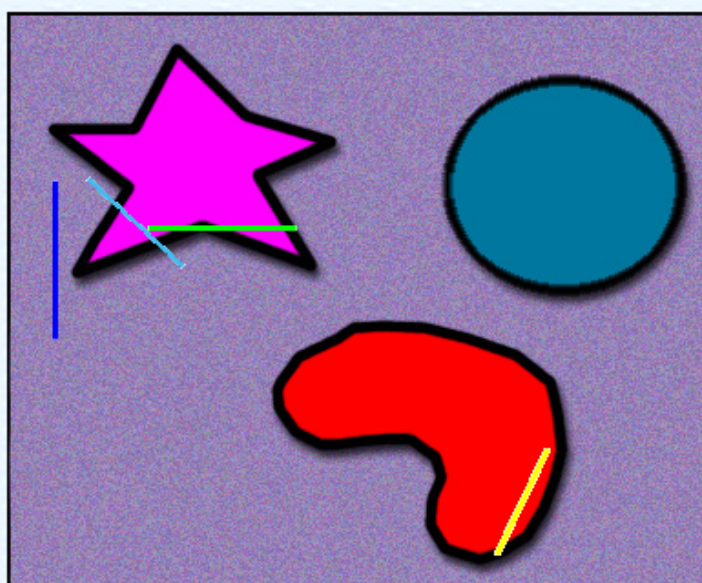


Figura j. Reactivo donde se observa el desplazamiento de una línea