



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

FACULTAD DE INGENIERÍA

**“SISTEMA INTEGRAL DE ADMINISTRACIÓN Y
EVALUACIÓN DE REACTIVOS DE OPCION
MÚLTIPLE EN LÍNEA UTILIZANDO AJAX”
(SIAEROM)**

T E S I S

**QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN**

PRESENTAN:

**RAMÍREZ LERMA GERARDO
RIVERO DOMÍNGUEZ ABRAHAM**

**DIRECTOR DE TESIS:
ING. LUCIRALIA HERNÁNDEZ HERNÁNDEZ**

MÉXICO, D.F. 2007





Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

Índice

Introducción	I
I Marco Teórico	
1.1 Modelo de Datos	1
1.1.1 Introducción	1
1.1.2 Entidades	1
1.1.2.1 Atributos	2
1.1.2.2 Identificación de la Llave Primaria	2
1.1.2.3 Tipos de Atributos	3
1.1.2.4 Generalización de Jerarquías	3
1.1.3 Relaciones	4
1.1.3.1 Relaciones N-arias y Relaciones con Atributos	5
1.1.4 Información Derivada	6
1.1.4.1 Atributos Derivados	7
1.2 Modelo de Hipertexto	8
1.2.1 Introducción	8
1.2.2 Unidades	9
1.2.2.1 Unidad de Datos	10
1.2.2.2 Unidad Multidatos	10
1.2.2.3 Unidad Índice	11
1.2.2.4 Unidad de Entrada	11
1.2.3 Enlaces	12
1.2.3.1 Especificación de Enlaces	12
1.2.4 Organización del Hipertexto	13
1.2.4.1 Vistas del Sitio	13
1.2.4.2 Áreas y Páginas	13
1.2.5 Modelo Administrativo del Contenido	15
1.2.5.1 Operaciones	16
1.2.5.1.1 Operaciones Predefinidas	16
1.2.5.1.1.1 Creación del Objeto	16
1.2.5.1.1.2 Eliminación del Objeto	17
1.2.5.1.1.3 Modificación de Objetos	17
1.2.5.1.2 Operaciones Genéricas	17
II Análisis de Requerimientos del SIAEROM	
2.1 Especificación de Grupos	19
2.2 Especificación de los Casos de Uso	21
2.3 Especificación de Diccionario de Datos	28
2.4 Especificación de Vistas del Sitio	31

2.5 Especificación de Lineamientos de Estilo	33
2.6 Justificación de Herramientas	37
III Diseño del SIAEROM	
3.1 Diseño de Arquitectura	41
3.1.1 Introducción	41
3.1.2 Dimensiones del Diseño de Arquitectura	41
3.1.2.1 Objetivos del Diseño de Arquitectura	41
3.1.2.2 Restricciones en el Diseño de Arquitectura	42
3.1.2.3 Escenarios de Implantación de la Arquitectura	42
3.1.3 Diseñando la Arquitectura de Hardware y Red	42
3.2 Diseño de Datos	44
3.2.1 Identificación de Entidades Núcleo	44
3.2.2 Diseño de Subesquemas Núcleo	45
3.2.3 Definición del Subesquema de Acceso	45
3.2.4 Definición del los Subesquemas de Interconexión	46
3.2.5 Definición de los Subesquemas de Personalización	46
3.3 Diseño de Hipertexto	48
3.3.1 Diseño Elemental	48
3.3.2 Diseño a Detalle	50
3.3.2.1 Especificación de las Páginas en Notación WebML	58
3.3.2.1.1 Vista del Sitio Personal Evaluador	58
3.3.2.1.2 Vista del Sitio Personal a Evaluar	66
3.3.2.1.3 Vista del Sitio Personal Administrador	67
IV Implementación del SIAEROM	
4.1 Implementación de Datos	69
4.1.1 Mapeo Estándar	69
4.1.2 Mapeo de las Entidades	69
4.1.3 Mapeo de las Relaciones	70
4.1.4 Formato de Transmisión de Datos	75
4.2 Implementación de Hipertexto	75
4.2.1 Gestor de Pruebas	77
4.2.2 Gestor de Contenido de Prueba	80
V Pruebas del SIAEROM	
5.1 Pruebas Funcionales	83
5.1.1 Prueba de Unidad	84
5.1.2 Prueba de Integración	86
5.1.3 Prueba de Sistema	86
5.2 Pruebas de Uso	86
5.3 Pruebas de Desempeño	87
5.3.1 Estimando la Carga de Trabajo	87
5.3.2 Estableciendo el Medio de la Prueba	87

5.3.3 Verificando el Desempeño	88
5.4 Aspectos no Funcionales	91
5.5 Estándares de la Web	91
VI Conclusiones	93
VII Manual de Usuario	95
VIII Glosario	103
XI Regencias Bibliográficas	107

Introducción

En la actualidad las herramientas y tecnologías Web facilitan tareas que antes se tornaban tediosas y consumían tiempo considerable. Sin embargo, con el auge de los sistemas de comunicación, como el Internet, se ha permitido a los usuarios realizar procesos con mayor facilidad, comodidad, casi en cualquier hora y además, sin la necesidad de la intervención directa de personal entre los procesos de la aplicación Web y el usuario.

En estos tiempos de afluencia comunicativa, actividades como realizar compras por Internet, servicios de mensajería o publicación de material de diversa índole (como bibliográfico), son comunes entre la gente que visita sitios de Internet. Actividades que anteriormente se realizaban de manera tradicional, ahora son realizadas en forma electrónica o vía Internet. Para el caso de la educación no es una excepción, esto representa una revolución en los paradigmas de la enseñanza y de la educación, ya que en muchos sitios de Internet se pueden encontrar variado y extenso material de diversas áreas del conocimiento, en muchos casos, de libre acceso.

Ahora, muchos de esos materiales educativos que se encuentran en Internet adquieren otra función que la de educar, y es la de proporcionar al usuario que los consulta un certificado que respalde la dominación del tema, dicho en otras palabras, es el proporcionar un grado o calificación.

El sistema a desarrollar es una herramienta de construcción y evaluación de pruebas, cuya intención sea de dotar al evaluador de una interfaz amigable para la generación de reactivos con sus respectivas opciones, para que posteriormente el sistema publique tal contenido en algún espacio del Internet, donde los alumnos puedan consultarlo y ser evaluados. La evaluación del alumno se logra cuando el sistema despliega la información de la prueba que será resuelta por él en tiempo límite. Cuando el alumno finaliza la prueba el sistema evaluará las opciones elegidas y generará un resultado que corresponderá a la calificación obtenida por el alumno.

OBJETIVO:

Diseñar e implementar una aplicación Web, que permita a los profesores de asignatura, crear y administrar en línea, cuestionarios y exámenes, con la finalidad de que puedan ser contestados también en línea, por sus alumnos y procesados por la aplicación.

ORGANIZACIÓN DE LA TESIS

Este trabajo de tesis está organizado en cinco capítulos, describiendo de manera general el proceso de desarrollo de la aplicación Web. Desde una fase de requerimientos hasta una fase de pruebas, tal y como lo marcan los modelos de desarrollo de software.

Introducción

El *Capítulo 1* describe las bases sobre las que se cimienta el diseño de la aplicación Web. Comenzando por una breve explicación sobre las primitivas del lenguaje de modelado de datos Entidad-relación, cuyos elementos son las entidades y sus relaciones. Las entidades son contenedores de elementos de datos y poseen propiedades traducidas en atributos con un tipo asociado, mientras las relaciones definen asociaciones entre entidades. Continuando, con la explicación del lenguaje de modelado de hipertexto, el cual, está basado en la noción de unidades, páginas, y enlaces. Las unidades describen las piezas elementales de contenido a ser desplegado, las páginas indican como las unidades deberían ser ensambladas en su conjunto, y los enlaces describen la conexión entre unidades y/o páginas. Múltiples hipertextos, llamados vistas del sitio, pueden ser definidos bajo el mismo contenido, para ofrecer diferentes puntos de vista a diferentes usuarios.

El *Capítulo 2* enfoca en el análisis de requerimientos, una actividad dedicada a la colección y especificación de los requerimientos de la aplicación, previa a las fases de modelado y diseño. La recolección de requerimientos enfoca en identificar a los usuarios y grupos, definición funcional, de datos, y personalización de requerimientos, además de los requerimientos no funcionales acerca de la presentación, usabilidad, ejecución, disponibilidad, escalabilidad, seguridad, y mantenimiento. Los requerimientos funcionales son formalizados por medio de diagramas de casos de uso en UML, los conceptos base y las vistas del sitio son expresadas por medio de un diccionario de datos y de mapas del sitio; finalmente, los lineamientos de estilo visual son expresados en la forma de maquetas de interfaz que establecen los elementos gráficos y partes estáticas del ambiente de presentación.

El *Capítulo 3* describe la actividad del diseño de datos y muestra la forma particular que esta tarea asume en el contexto de la Web. La estructura de datos de las aplicaciones Web frecuentemente presentan una organización regular, en la cual diversos sub-esquemas interconectados pueden ser reconocidos, cada uno centrado sobre una entidad central representando un objeto del negocio. Además, este capítulo presenta las actividades del diseño de hipertexto. El diseño ocurre de una forma descendente: Inicialmente, un bosquejo del esquema de hipertexto es obtenido al dividir cada vista del sitio identificada durante el análisis de requerimientos en áreas, y asignando un conjunto de funciones a cada área, la cual soporta la navegación de objetos base, de acceso o interconexión, o de operaciones de gestión de contenido. Entonces, un esquema esbozado de cada área es refinado en un esquema detallado, especificado en WebML; en esta fase, el diseñador establece las unidades actuales, los enlaces, las operaciones, y las páginas de cada vista del sitio.

El *Capítulo 4* trata con el mapeo de esquemas de datos conceptual hacia las fuentes de datos físicos, y con la lógica de programación empleada en la generación de código para pasar de una notación WebML a una notación adecuada a la tecnología del lado del servidor. Para generar las fuentes de datos se emplearán scripts que gestionen las tablas y campos de la base de datos, del lado de la lógica de programación de la aplicación se describen genéricamente las instrucciones para la gestión de su contenido y de su estructura.

El *Capítulo 5* describe las tareas involucradas en la fase de pruebas y validación de la aplicación. Aspectos funcionales y no funcionales califican el correcto funcionamiento y lo

que se espera de la aplicación Web en cuando a la satisfacción de requerimientos. El capítulo presenta las distintas técnicas de verificar el correcto funcionamiento del sistema, desde el momento de su implementación e integración, hasta la puesta en funcionamiento.

Capítulo I

Marco Teórico

1.1 Modelo de Datos

El modelado de datos, es una de las disciplinas más consolidadas de las Tecnologías de la Información, para la cual existen lenguajes y procedimientos bien definidos. Por lo cual, el presente capítulo, no propone otro lenguaje de modelado de datos, sino que explota la notación más exitosa y popular, conocida como el modelo Entidad-Relación (E-R).

Este capítulo introduce los conceptos esenciales de modelado de datos, suficientes para especificar el esquema de datos de la aplicación Web.

1.1.1 Introducción

El propósito del modelado de datos es permitir especificar la información que será usada por la aplicación de manera intuitiva y formal. El resultado del modelado de datos es un *esquema conceptual*.

El diseño de dicho esquema es preliminar tanto al diseño de las funciones del negocio que operarán sobre los datos y a la implementación de las estructuras físicas que soporten el almacenamiento, extracción y actualización de los datos.

Los principales elementos del modelo Entidad-Relación son las entidades, definidas como contenedores de datos estructurados y las relaciones, las cuales representan asociaciones semánticas entre las entidades.

1.1.2 Entidades

Una entidad representa una descripción de las características comunes de un conjunto de objetos del mundo real. Una entidad tiene una *población*, la cual es el conjunto de objetos que son descritos por la entidad.

Las entidades se denotan por rectángulos, con su nombre en la parte superior como se muestra en la figura 1.1.

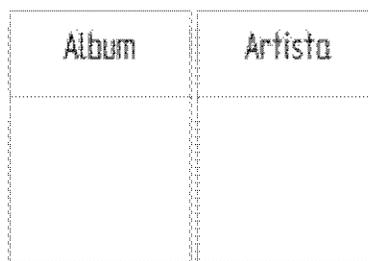


Figura 1.1 Notación gráfica para las entidades

1.1.2.1 Atributos

Los atributos representan las propiedades de objetos del mundo real, que son relevantes para propósitos de la aplicación. Estos atributos están asociados con las entidades, con el significado de que todas las instancias de una entidad estarán caracterizadas por el mismo conjunto de atributos.

Es admisible que una instancia de entidad tenga un valor nulo para uno o más de sus atributos.

- Un valor nulo puede denotar que un determinado atributo, no aplica a una instancia de entidad en particular.
- Un valor nulo puede denotar que el valor de cierto atributo es desconocido para cierta instancia de entidad.

Los atributos son representados de manera gráfica dentro de la caja de entidad, debajo del nombre como se muestra en la figura 1.2.



Figura 1.2 Notación gráfica para entidades y atributos

1.1.2.2 Identificación de la Llave Primaria

Todas las instancias de una entidad deben distinguirse a través de una identidad única que permita su identificación. Para expresar la identidad única de las instancias de entidad, uno o más atributos pueden ser definidos como la *llave primaria* de la entidad.

Los atributos que conforman la llave primaria deben satisfacer las siguientes restricciones:

- Su valor debe estar definido (esto es, no puede ser nulo), para cada instancia de entidad.
- Su valor debe ser único para cada instancia de entidad.

Es una buena práctica el definir la llave primaria de las entidades usando un atributo de propósito especial, llamado identificador de objeto (OID), cuyo único propósito es de asignar un identificador distinto a cada instancia de entidad.

Gráficamente, los atributos llave, son distinguidos por un pequeño icono de llave a la derecha del nombre del atributo.

1.1.2.3 Tipos de Atributos

Los atributos pueden tener un tipo, lo cual significa que adquieren valores de un dominio bien definido, por ejemplo, el conjunto de los números enteros. El modelo Entidad – Relación no prescribe ningún conjunto específico para el tipo de los atributos, pero es una buena práctica el expresar los tipos de los atributos en el modelo de datos, tanto para hacer la especificación más expresiva, como para dirigir la implementación de datos.

Los tipos de datos soportados normalmente por los productos de bases de datos son:

Tipo de Dato	Descripción
Cadena (String)	Una secuencia “pequeña” de caracteres.
Texto (Text)	Una secuencia “larga” de caracteres. Los tipos texto, pueden ser refinados especificando su tipo MIME, por ejemplo: text/html.
Entero (Integer)	Un tipo entero numérico.
Flotante (Float)	Un tipo flotante numérico.
Fecha (Date)	Una fecha del calendario.
Hora (Time)	Una instancia temporal del tiempo.
Booleano (Bolean)	Un valor verdadero o falso.
Enumeración (Enumeration)	Una secuencia de valores definidas por los usuarios.
BLOB	Objeto binario grande. (Binary Large Object).
URL	Localizador uniforme de recursos web.

1.1.2.4 Generalización de Jerarquías

El modelo Entidad – Relación permite al diseñador, organizar las entidades en una jerarquía, en la que se comparten características comunes. La jerarquía de generalización básica (también llamada jerarquía Es-un) tiene una superentidad y una o más subentidades. Cada subentidad hereda todos los atributos y relaciones definidas en la superentidad y agrega de manera local, atributos y relaciones.

Una jerarquía de generalización no está limitada a dos niveles, una subentidad puede estar especializada en una o más subentidades, resultando en una jerarquía con una profundidad arbitraria.

Las siguientes suposiciones permiten asegurar que el esquema Entidad – Relación es fácil de implementar usando cualquier tecnología de base de datos convencional.

1. Cada entidad está definida como la especialización de cuando mucho una superentidad. En términos técnicos, no se permite herencia múltiple.
2. Cada instancia de una superentidad está especializada exclusivamente en una subentidad.

3. Cada entidad aparece cuando mucho, en una jerarquía de generalización.

1.1.3 Relaciones

Las relaciones representan conexiones semánticas entre las entidades. La forma más simple de una relación es la binaria, la cual conecta dos entidades. Las relaciones que involucran a más de dos entidades, llamadas relaciones N-arias, son permitidas; sin embargo, el uso de relaciones N-arias no es promovido, debido a que pueden ser expresadas de manera equivalente a través de muchas relaciones binarias.

Cada relación binaria es caracterizada por dos roles de relación, cada uno expresando la función que una de las entidades participantes juega en la relación (ver figura 1.3).



Figura 1.3 Notación gráfica para roles de relación

Los roles de relación pueden ser anotados con restricciones de cardinalidad mínima y máxima, respectivamente denotando el número máximo y mínimo de objetos de la entidad destino a los cuales cualquier objeto de la entidad origen pueden ser relacionados (ver figura 1.4).



Figura 1.4 Notación gráfica para roles de relación con su respectivas cardinalidades.

- Los valores relevantes para la cardinalidad mínima son cero y uno; una relación se dice que es opcional a su entidad origen si la cardinalidad mínima es cero, y obligatoria en cualquier otro caso.
- Los valores relevantes para la cardinalidad máxima son uno o muchos, la segunda opción denotada por “N”.

Basado en las restricciones de cardinalidad máxima, las relaciones son nombradas “uno a uno”, si ambos roles de relación tienen una cardinalidad máxima de uno, “uno a

muchos” si uno de los roles de relación tiene cardinalidad máxima de uno y el otro de N, o “muchos a muchos” si ambos roles de relación tienen cardinalidad máxima de N.

1.1.3.1 Relaciones N-arias y Relaciones con Atributos

Una relación N-aria es equivalente a una entidad “central” y N relaciones binarias, conectando la entidad central con cada una de las entidades de la relación N-aria. Las restricciones sobre la cardinalidad de las relaciones binarias, tendrán así, una cardinalidad mínima y máxima igual a uno del lado de la entidad central, para expresar el hecho que una instancia de la entidad central deberá estar conectada con exactamente una instancia de cada una de las demás entidades, dado que no corresponde a un objeto del mundo real, sino un artefacto técnico que denota la conexión de N objetos del mundo real (ver figura 1.5).

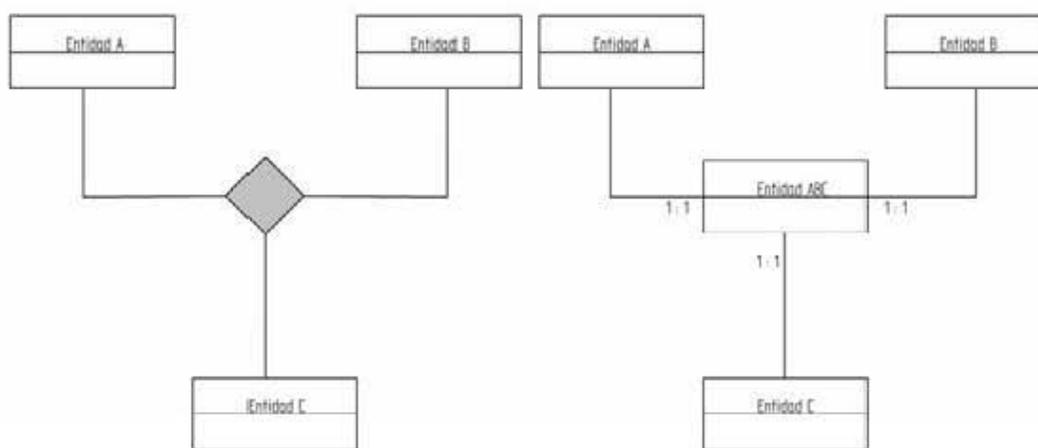


Figura 1.5 Relaciones N-arias: expresada como una construcción primitiva (izquierda) y empleando relaciones binarias y entidades (derecha)

Una relación (binaria) con atributos es equivalente a una entidad “central”, conectada por dos relaciones binarias a las entidades participantes de la relación con atributos (ver figura 1.6). También en este caso, las restricciones sobre la cardinalidad de las relaciones binarias deberán tener una cardinalidad mínima y máxima en uno de los lados de la entidad central, para expresar el hecho de que una instancia de esta entidad deberá estar conectada a exactamente una instancia de las otras entidades, dado que no corresponde a un objeto del mundo real, sino una notación para expresar los atributos relevantes a dos objetos del mundo real.

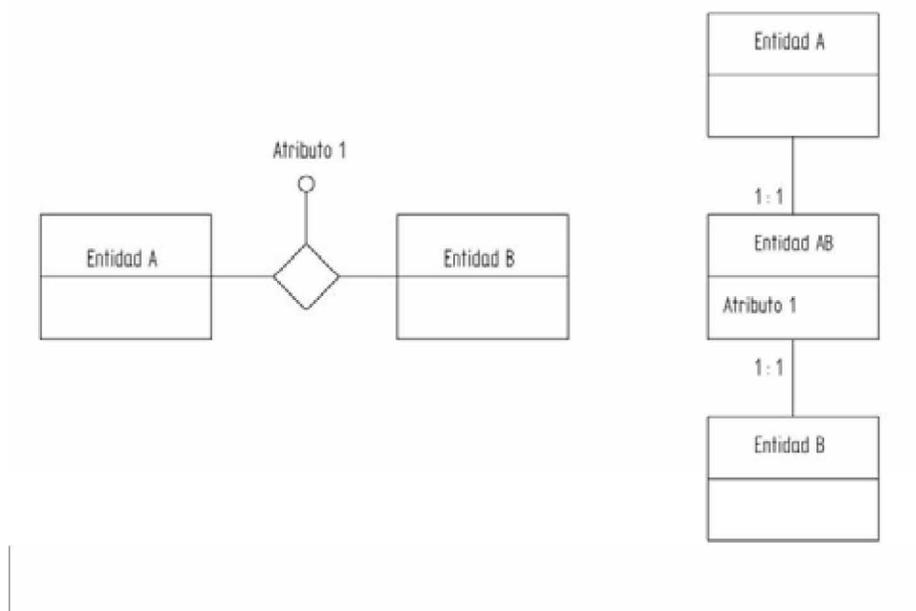


Figura 1.6 Relaciones con atributos: expresada como una construcción primitiva (izquierda) y empleando relaciones binarias y entidades (derecha)

El resultado final del esquema Entidad – Relación, consistirá al final de entidades y relaciones sin atributos.

Una relación N-aria con atributos será tratada de manera similar: se crea una entidad central con los atributos, y las N relaciones binarias son dibujadas entre la entidad central y las otras entidades involucradas.

1.1.4 Información Derivada

Los atributos y relaciones que pueden ser calculados a partir de otros, se denominan *derivados*.

El modelo Entidad – Relación no incluye ninguna notación estándar para caracterizar los atributos y las relaciones como derivadas, ni tampoco un lenguaje para expresar sus reglas de cómputo (la manera en que han de obtenerse a partir de otros elementos).

Sin embargo, la especificación de un atributo o de una relación puede ser fácilmente extendida, para soportar el modelado de la información derivada.

- Un atributo o relación es denotado(a) como derivada, mediante la adición de un carácter (/) en frente del nombre del atributo o de la relación.
- La regla de cómputo que define el atributo derivado o la relación es especificada como una expresión añadida a la declaración del atributo o de la relación.

La siguiente figura muestra dos ejemplos de atributos derivados. Entre sus atributos,

la Entidad-Artículo incluye dos atributos regulares, Precio y Descuento, y un atributo derivado /PrecioDescantado, el cual es calculado como el valor de la expresión (Precio * Descuento).

La Entidad-Artista contiene un atributo derivado /NumeroDeAlbumes, el cual es calculado como el valor de la expresión count(Artista.ArtistaAlbum). Esta expresión cuenta el número de álbumes asociados con un artista de acuerdo con el rol de relación ArtistaAlbum. La sub-expresión Artista.ArtistaAlbum es un ejemplo de *expresión de ruta*, la cual es usada para recorrer un rol de relación que emana desde una entidad (ver figura 1.7).



Figura 1.7

1.1.4.1 Atributos Derivados

La siguiente figura muestra un ejemplo de una relación derivada: la entidad Artista es asociada a la entidad Pista mediante una relación derivada /ArtistaPista, la cual es la concatenación de dos relaciones: entre un artista y sus álbumes y entre un álbum y sus pistas.

La regla de derivación es expresada en uno de los dos roles de relación a través de una expresión de ruta. En este ejemplo, la regla de derivación es aplicada al rol de relación de la entidad Artista a la Entidad-Pista, y es formalmente especificado añadiendo al rol de declaración la expresión: $\text{Artista.PistasPublicadas} = \text{Artista.ArtistaAlbum.AlbumPista}$ (ver figura 1.8).



Figura 1.8 Relación derivada.

1.2 Modelo de Hipertexto

A diferencia del modelado de datos, la cual es una disciplina bien consolidada, el modelado de hipertexto es una disciplina aún más joven, carente aún, de una bien establecida base de conceptos. WebML, el lenguaje usado en el presente trabajo, provee las primitivas para el modelado de hipertexto, tomando prestado del modelo Entidad Relación la idea de usar conceptos de especificación simples y expresivos, soportados por una notación gráfica intuitiva. Por lo tanto debe ser percibido por el diseñador, como una extensión natural del modelo Entidad-Relación, el cual permite al programador extender el esquema de datos con la especificación de hipertextos usados para publicar y manipular la información.

1.2.1 Introducción

La meta del modelado de hipertexto es la especificación de la organización de las interfaces que interactúan con el usuario en una aplicación Web. Para lograr ese fin, se deben emplear herramientas para plasmar de manera simple e intuitiva los aspectos involucrados en el modelado, como dividir la extensa aplicación en módulos principales, incorporar en cada módulo un conjunto de funciones asociadas a una cierta clase de usuarios que origine una división en submódulos, esta disposición implica crear una topología de hipertexto para cada módulo que sea traducido en páginas, elementos de contenido y enlaces que soporten la interacción y navegación del usuario.

El lenguaje WebML proporciona primitivas para el modelado del hipertexto, toma la idea del modelo Entidad-Relación para representar simple e intuitivamente los conceptos, notaciones y métodos de diseño en una notación gráfica. Este proceso permite expandir el esquema de datos de la aplicación con la especificación del hipertexto cuya finalidad sea la de manipular y publicar los datos del contexto, así como el contenido de la aplicación Web. El proceso de desarrollo del modelado tiende a enfocarse a un nivel abstracto y conceptual, y no contempla un análisis a detalle en su diseño e implementación, ni mucho menos en la funcionalidad entre otras aplicaciones Web.

Las partes principales de WebML son las *páginas*, las *unidades*, y los *enlaces* organizados en construcciones modulares llamadas *áreas* y *vistas de sitio*.

1.2.2 Unidades

Las *unidades* son piezas elementales para la publicación de contenido y especifican los datos de entrada del usuario. Estas piezas son bloques de construcción para las *páginas*. Ofrecen una distinta forma de organizar dinámicamente el contenido extraído de las entidades y relaciones del esquema de datos. WebML soporta cinco tipos de unidades:

- *Unidades de datos*: muestran información sobre un objeto simple.
- *Unidades multidatos*: presentan información sobre un conjunto de objetos.
- *Unidades índices*: Muestra una lista de las propiedades descriptivas de algunos objetos, sin presentar a gran detalle la información que los caracteriza.
- *Unidades scroller*: hacen posible la visualización y navegación de un conjunto ordenado de objetos.
- *Unidades de entrada*: permiten la adquisición de datos por parte del usuario.

Las primeras cuatro unidades modelan la publicación de la información, mientras la quinta unidad está enfocada en la adquisición de información por parte de los usuarios. Las unidades de datos y multidatos representan el actual contenido de los objetos, que son sus fuentes; mientras las unidades índices y scroller indican la selección del objeto.

Las cuatro unidades se clasifican como unidades de contenido, porque representan contenido extraído del esquema de datos, además será necesario especificar el origen de tal contenido. Para ello, WebML utiliza dos conceptos:

- *La fuente*. Nombra las entidades fuentes donde se extrae el contenido del objeto.
- *El selector*. Lista los parámetros utilizados para extraer el contenido del objeto alojado por la entidad fuente. Estos parámetros son llamados selectores paramétricos y su origen radica en los atributos y reglas de relación, para el caso del modelo Entidad-Relación las llaves de los atributos son la opción.

Las unidades se representan gráficamente por un rectángulo, encerrando el nombre de la unidad con su respectivo icono, como el mostrado por la figura 1.9. Para el caso de unidades de contenido, la fuente y el selector se colocan debajo del rectángulo.

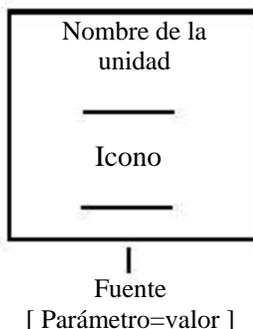


Figura 1.9 Representación gráfica de la unidad de contenido

1.2.2.1 Unidad de Datos

Las unidades de datos sólo publican contenido de una entidad dada. En la figura 1.10 se muestra la notación gráfica que representa una unidad de datos con nombre ArtistaPequeño y su fuente.



Figura 1.10 Notación gráfica en WebML para una unidad de datos, con su representación en HTML.

1.2.2.2 Unidad Multidatos

Estas unidades representan múltiples objetos de una entidad conjunta, la unidad aloja a varios o a todos los objetos y sus características disponibles en la entidad. Esta unidad es usada para procesar todos los objetos desplegados. En la figura 1.11 se muestra la notación para representar la unidad multidato. Esta unidad se encuentra definida bajo la entidad Artista, que despliega todos los objetos disponibles.



Figura 1.11 Notación gráfica en WebML para una unidad de multidatos, con su representación en HTML

1.2.2.3 Unidad Índice

La unidad índice presenta en lista a múltiples objetos de una entidad, sin incluir contenido. Esta unidad es utilizada para seleccionar algún objeto en particular. En la figura 1.12 se muestra la notación gráfica de una unidad índice, con sus propiedades. Ésta se encuentra definida bajo la entidad Álbum y despliega todos los objetos que pueden ser seleccionados.

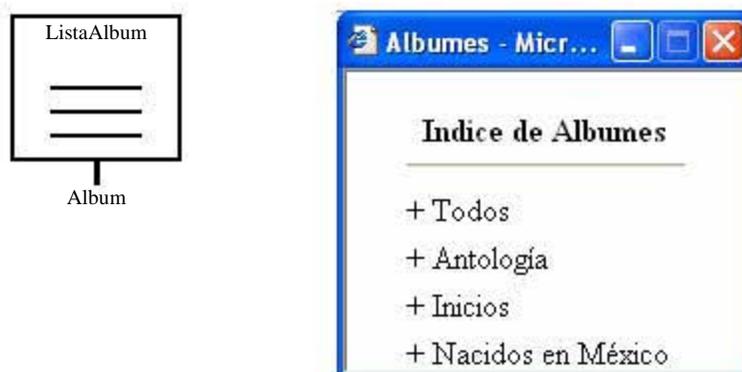


Figura 1.12 Notación gráfica en WebML para una unidad índice, con su representación en HTML

1.2.2.4 Unidad de Entrada

Las unidades de entrada soportan entradas de datos provenientes de formularios. Son usadas para reunir datos, empleadas para realizar lo siguiente:

- Realizar búsquedas sobre los objetos de una entidad, y
- Suministrar parámetros a operaciones como actualización, login y servicios externos.

En la figura 1.13 se muestra un ejemplo de este tipo de unidades.



Figura 1.13 Notación gráfica en WebML para una unidad de entrada, con su representación en HTML

1.2.3 Enlaces

Las *páginas* y *unidades* no permanecen como elementos aislados, los enlaces representan la parte medular del modelado de hipertexto, estos permiten la navegación de un punto a otro, además pasan los parámetros de una unidad a otra en la que son requeridos para el procesamiento apropiado del contenido de una página. Las páginas y unidades comparten relación por medio de enlaces, que especifican la ruta o camino de navegación permitido entre páginas, son las selecciones disponibles para el usuario y son los causantes de procesar el contenido de las unidades desplegadas en la página.

1.2.3.1 Especificación de Enlaces

Un *anchor* es un elemento activo utilizado por el usuario para interactuar con el hipertexto, conceptualiza la idea abstracta de un enlace. En el caso de HTML, un anchor se refiere al valor que posee en el atributo *href* de la etiqueta anchor para visualizar otra página o para actualizar la página actual, también se refiere al envío de información a otras páginas e incluso a la actual por medio de un botón. Por lo tanto, la función del anchor es habilitar la navegación del hipertexto y transportar la información de una unidad a otra.

En terminología WebML, los enlaces se clasifican de acuerdo con la fuente-destino como:

- *Enlace inter-página*. Aquel, que cruza los límites de la página.
- *Enlace intra-página*. Aquel, cuya fuente y destino está dentro de la misma página. Este enlace se presenta entre la conexión de unidades.

Y en cuanto al transporte de información:

- *Enlace contextual*. Refiere al enlace transportando información.
- *Enlace no-contextual*. Refiere a aquel que no transporta información.

La notación gráfica del enlace es una flecha con dirección apuntando el destino de la conexión entre unidades o páginas. Se añaden los parámetros enviados al destino, cuando

los hay, colocando el predicado sobre la flecha.

1.2.4 Organización del Hipertexto

La especificación del hipertexto puede ser organizada jerárquicamente utilizando construcciones modulares como vistas de sitio y áreas.

1.2.4.1 Vistas del Sitio

Un conjunto de *páginas* se agrupan en una *vista del sitio*, éstas representan un hipertexto referente al conjunto de requerimientos asociados a un grupo específico de usuarios, esto es, cada grupo podrá acceder a las partes del contenido que les corresponda. Podrán existir múltiples *vistas del sitio* para un mismo esquema de datos. Una vista del sitio empaqueta una serie de páginas enlazadas y unidades, tal construcción modulada se caracteriza por un nombre definido por el usuario y un espacio definido por áreas, las cuales contienen un conjunto de páginas.

En la figura 1.14 se muestra un ejemplo de una notación gráfica para una vista del sitio. La vista del sitio se llama *Álbum*, agrupa varias páginas. La página *Artista* despliega la lista de artistas contenidas por la unidad *TodosArtistas*. La página *IndiceAlbum* despliega el dato de un artista por la unidad *DatoArtistaPequeño* y, además la lista de sus álbumes por parte de la unidad *IndiceAlbum*. Finalmente, la página *Álbum* provee la información referente a un álbum simple desplegado por alguna unidad de datos e índice.

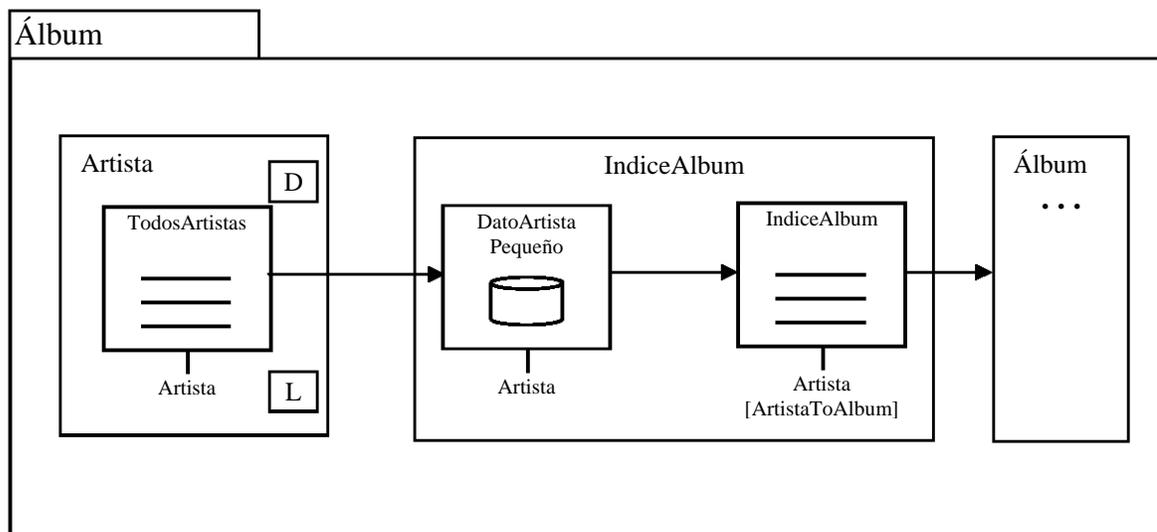


Figura 1.14 Notación gráfica WebML de una vista del sitio

1.2.4.2 Áreas y Páginas

Las *vistas del sitio* podrán descomponerse en *áreas*, estas *áreas* representan un grupo de *páginas* con un propósito específico de acuerdo con la necesidad de cada grupo de

usuarios. WebML proporciona primitivas para organizar las páginas y vistas del sitio, como las siguientes:

- *Áreas*. Alojjan páginas o, recursivamente, sub-áreas. Los enlaces pueden dibujarse por todo el espacio definido en el área, e incluso, pueden cruzar sus fronteras, en este caso los enlaces son llamados enlaces interárea.
- *Página inicio*. Es la página seleccionada por defecto en el sitio o presentada después de que el usuario haya obtenido el permiso de acceso por medio del login. La página de inicio debe ser única para cada vista del sitio. Se utiliza la letra “H” dentro del icono de la página para denotar la propiedad home (inicio) de la página. En la declaración textual de la página se agrega la palabra clave *home*.
- *Página default*. Representa la página activa cuando se accede al área. En esta página debe estar definida el área. En la notación gráfica se añade la letra “D” dentro del icono de la página para denotar esta propiedad, y en la declaración textual se agrega la palabra clave *default*.
- *Página landmark*. Representa la página accesible desde otras áreas o páginas definidas en la misma vista del sitio o en la misma super-área. Se agrega la letra “L” en el icono de la página, y en la declaración textual se adiciona la palabra clave *landmark*.

Las propiedades default y landmark pueden extenderse a áreas. Un área default es aquella subárea accedida por default al momento de acceder a su superárea. Y un área landmark es un área accesible desde otras áreas definidas en la vista el sitio o superáreas.

En la figura 1.15 se ilustra el uso de áreas, vistas del sitio y propiedades añadidas a páginas.

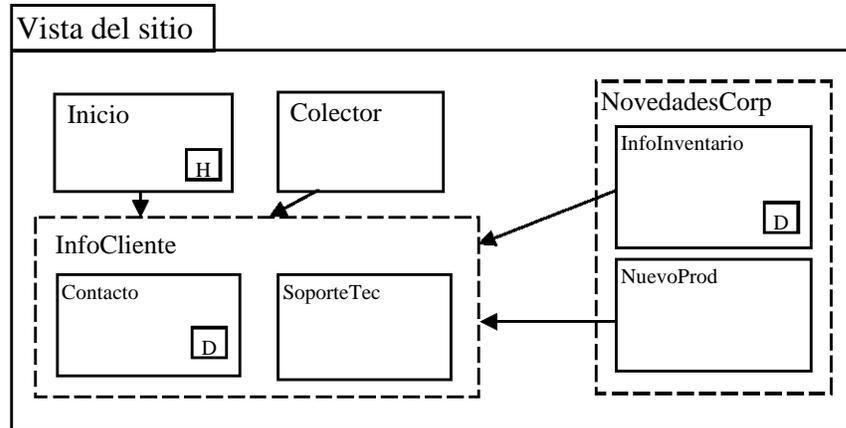


Figura 1.15 Descomposición de una vista del sitio en áreas, incluye propiedades de páginas y de áreas.

Las áreas que contienen otras áreas, llamadas superáreas, son representadas por rectángulos con líneas punteadas, mientras las áreas simples y subáreas son representadas por rectángulos con líneas normales.

1.2.5 Modelo Administrativo del Contenido

Las aplicaciones administrativas de contenido montadas en Web, ofrecen ejecución de operaciones sobre los datos de las aplicaciones Web, ya sea para cambiar o actualizar contenido. Tales operaciones están involucradas en acciones como el llenado de formularios referentes al perfil del usuario, la adición de artículos al carrito de compras, o la actualización del contenido publicado en sitios Web. Existen programas externos que administran y procesan datos, cuyas entradas dependen del usuario y de sus selecciones dentro de la interfaz que conecta a esos programas. Algunas aplicaciones muy comunes son las tareas del login, logout, y correo electrónico.

WebML ofrece una notación para representar cada operación sin afectar el modelo de datos. La especificación de las operaciones se basa en extensiones del modelo del hipertexto, las cuales son dos y son las siguientes:

- *Unidades de operación.* Utilizadas para expresar algún proceso como resultado de la navegación de algún enlace. Denotan la manipulación de un dato o la ejecución de un servicio externo.
- *Enlaces de salida.* Agrupa los enlaces de salida para las unidades de operación, tales enlaces son los enlaces-OK y los enlaces-KO, que asocian las operaciones de éxito o fallo, respectivamente. Permite tomar caminos de acción acorde a la ejecución de una operación.

Existen dos tipos de operaciones usadas en el modelado administrativo del contenido para una aplicación Web, las operaciones predefinidas y las operaciones genéricas.

1.2.5.1 Operaciones

Las unidades de operación en WebML o simplemente operaciones, son un tipo de unidades colocadas fuera de las páginas y enlazadas a otras operaciones o a otras unidades de contenido definidas dentro de páginas. La diferencia entre una unidad contenido, es que no despliegan contenido sino que ejecutan alguna acción, por esa razón son colocadas fuera de las páginas. Y la semejanza con una unidad de contenido, una operación pueden tener un objeto fuente y selectores, pueden recibir parámetros de entrada por medio de sus enlaces de entrada, y pueden proporcionar valores a sus parámetros por sus enlaces de salida.

Las operaciones obedecen a ciertos principios de diseño, como:

- La posibilidad de poseer múltiples enlaces de entrada.
- La posibilidad de enlazar varias operaciones para formar secuencias.
- Cada operación tiene un sólo enlace OK y uno KO.
- Una operación puede tener cualquier número de enlaces de transporte de salida, los cuales, son utilizados para especificar los parámetros de enlace asociados a las unidades contenido o a otras operaciones.

Las operaciones no tienen la facultad de despliegue de contenido, sólo conducen al elemento de la página que lo desplegará.

1.2.5.1.1 Operaciones Predefinidas

WebML ofrece unidades de operación predefinidas para representar a operaciones predefinidas. Estas unidades disponen de primitivas para actualizar las instancias de las entidades y sus relaciones, como la creación, modificación, eliminación de objetos, y la conexión y desconexión de éstos a través de relaciones, así como otras operaciones de utilidad como el login, logout y correo electrónico. Estas operaciones administrativas de contenido pueden agruparse en transacciones, cuya finalidad es representar una secuencia de procesos de actualización ejecutadas como una sola unidad.

1.2.5.1.1.1 Creación del Objeto

La primer operación pre-fabricada es la unidad creación, la cual ejecuta la creación de una nueva instancia. En la figura 1.16 muestra un ejemplo de operaciones de creación. La intención es crear una instancia de la entidad Artista por medio de la operación crear.

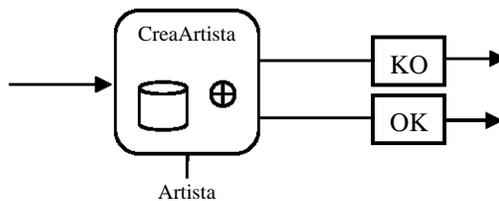


Figura 1.16 Notación gráfica en WebML para una unidad de operación para crear objetos, en este caso una instancia Artista.

1.2.5.1.1.2 Eliminación del Objeto

La unidad de eliminación es usada para eliminar uno o más objetos de una entidad dada. Los objetos aplicados esta operación son seleccionados o desplegados por unidades índices. El ejemplo de la figura 1.17 muestra una unidad eliminación

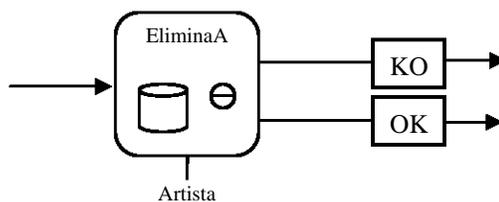


Figura 1.17 Notación gráfica en WebML para una unidad de operación para eliminar objetos, en este caso una instancia Artista.

1.2.5.1.1.3 Modificación de Objetos

La unidad modificación es usada en la actualización de uno o más objetos de una entidad dada.

En el ejemplo de la figura 1.18 muestra una unidad modificación.

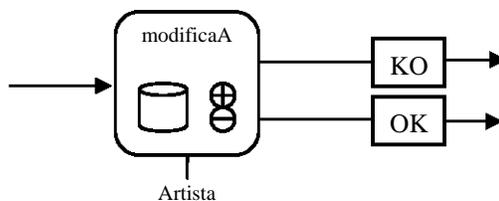


Figura 1.18 Notación gráfica en WebML para una unidad de operación para modificar objetos, en este caso una instancia Artista.

1.2.5.1.2 Operaciones Genéricas

Las operaciones genéricas interactúan en el contexto de WebML, pero sus estructuras y componentes internos no pueden ser representados en WebML, ya que las

Capítulo 1: Marco Teórico

acciones que manipulan estas operaciones no son contempladas como parte constitutiva del sitio. Más claramente, estas operaciones están asociadas a recursos disponibles fuera de los límites de una aplicaciones Web, ya que estos servicios y componentes son asociados a negocios bancarios como el pago electrónico.

Una operación genérica es una caja negra cuyos detalles internos no son especificados en WebML; sin embargo, éstas pueden ser enlazadas a unidades WebML, o bien a páginas.

Capítulo II

Análisis de Requerimientos del SIAEROM

2.1. Especificación de Grupos.

En la mayoría de los casos, una aplicación Web estará dirigida a diferentes categorías de usuarios, los quienes, pueden ser identificados a priori durante el diseño, y formalizados en un conjunto de grupos de usuarios. Los grupos pueden estar organizados de manera jerárquica, para denotar que un determinado grupo de usuarios con requerimientos homogéneos se especializan en subgrupos, los cuales agregan propiedades nuevas a los supergrupos.

Tabla 2.1 Grupo I: Personal Evaluador.

<i>Nombre</i>	Personal Evaluador.
<i>Descripción</i>	Personal encargado de definir y administrar el contenido de las pruebas, así como también, de consultar y registrar el resultado obtenido a partir de la solución de las mismas.
<i>Atributos</i>	Nombre, Apellido, Correo Electrónico, Nombre de Usuario, Contraseña. Los datos son proporcionados de manera explícita por el usuario.
<i>Supergrupo</i>	Ninguno.
<i>Subgrupo</i>	Ninguno.
<i>Casos de Uso Relevantes</i>	“login”, “editar datos personales”, “definir prueba nueva”, “abrir prueba”, “guardar prueba”, “insertar reactivo en prueba”, “editar reactivo de la prueba”, “eliminar reactivo de la prueba”, “alta de grupos”, “baja de grupos”, “asociar usuario: personal a evaluar, con algún grupo”, “desasociar usuario: personal a evaluar, de algún grupo”, “consultar resultados”, “salir del sistema”.
<i>Objetos accesibles en modo lectura</i>	Resultados de pruebas.
<i>Objetos accesibles en modo administración de contenidos</i>	Pruebas, alumnos, grupos, datos personales.

Capítulo 2: Análisis de Requerimientos

Tabla 2.2 Grupo II: Personal a Evaluar.

<i>Nombre</i>	Personal a Evaluar.
<i>Descripción</i>	Personal quien contesta las pruebas, puede consultar el resultado de dicha operación.
<i>Atributos</i>	Nombre, Apellido, Correo Electrónico, Nombre de Usuario, Contraseña. Los datos son proporcionados de manera explícita por el usuario.
<i>Supe-grupo</i>	Ninguno.
<i>Subgrupo</i>	Ninguno.
<i>Casos de Uso Relevantes</i>	“login”, “editar datos personales”, “contestar prueba”, “enviar prueba resuelta”, “consultar resultados”, “salir del sistema”.
<i>Objetos accesibles en modo lectura</i>	Pruebas, resultados de pruebas.
<i>Objetos accesibles en modo administración de contenidos</i>	Datos Personales.

Tabla 2.3 Grupo III: Personal Administrador del Sistema.

<i>Nombre</i>	Personal Administrador del Sistema.
<i>Descripción</i>	Personal encargado en dar de alta/baja a miembros del grupo personal evaluador.
<i>Atributos</i>	Nombre, Apellido, Correo Electrónico, Nombre de Usuario, Contraseña. Los datos son proporcionados de manera explícita por el usuario.
<i>Supergrupo</i>	Ninguno.
<i>Subgrupo</i>	Ninguno.
<i>Casos de Uso Relevantes</i>	“login”, “crear cuenta de usuario: personal evaluador”, “eliminar cuenta de usuario: personal evaluador”, “editar datos personales”, “salir del sistema”.
<i>Objetos accesibles en modo lectura</i>	Ninguno.
<i>Objetos accesibles en modo administración de contenidos</i>	datos personales, datos personales administrador del sistema.

2.2. Especificación de los Casos de Uso

Un caso de uso expresa una unidad de interacción entre los usuarios de un grupo particular y la aplicación. Cada caso de uso, es descrito mediante una hoja de especificaciones como se realiza a continuación:

Grupo I: Personal Evaluador

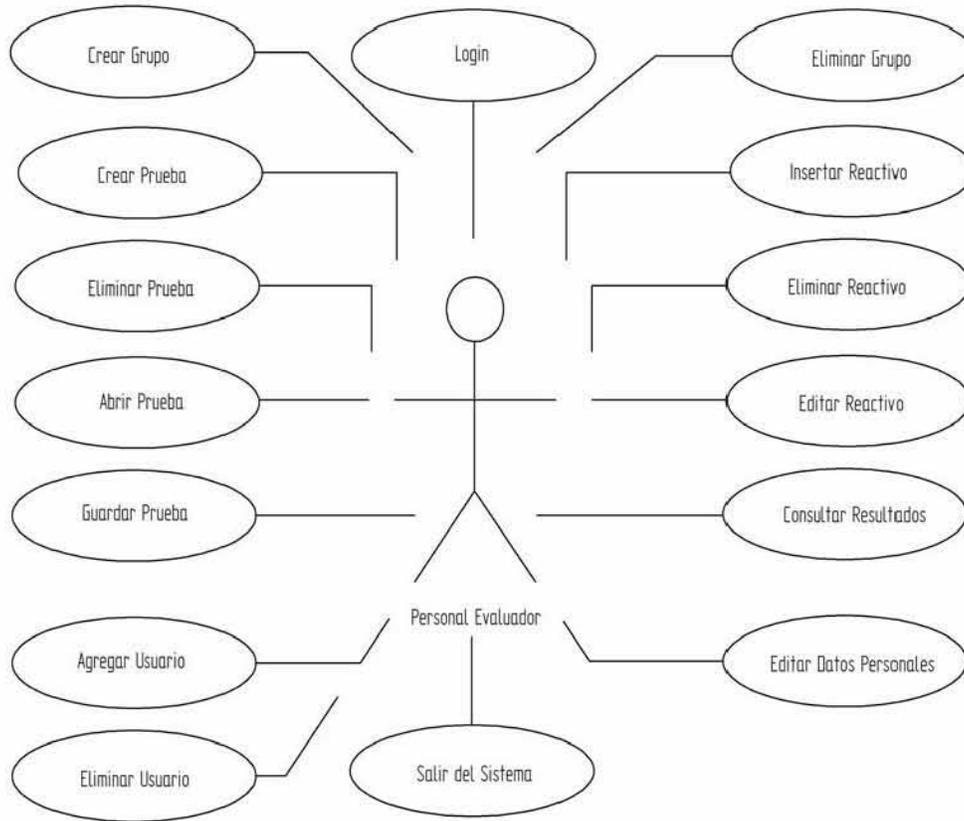


Figura 2.1 Diagrama de Casos de Uso Personal Evaluador.

Tabla 2.4 Caso de Uso: Login Personal Evaluador.

Título	Login
Propósito	Determinar si el usuario en cuestión, pertenece al grupo del personal evaluador.
Precondición	El usuario posee una cuenta del grupo personal evaluador.
Poscondición	El usuario tendrá acceso al menú del personal evaluador.
Procedimiento	<ol style="list-style-type: none"> 1. El usuario teclea un nombre de usuario y una contraseña. 2. Se verifica en la base de datos del sistema que el usuario pertenezca al grupo del personal evaluador.

Capítulo 2: Análisis de Requerimientos

	<ol style="list-style-type: none">3. Si la condición previa es verdadera, se le presenta al usuario un menú, en el que podrá ejecutar tareas propias del personal evaluador.4. Si los datos ingresados no pertenecen a algún registro de la base de datos, se le niega el acceso al usuario.
--	---

Tabla 2.5 Caso de Uso: Editar Datos Personales. Personal Evaluador.

<i>Título</i>	Editar datos personales
<i>Propósito</i>	Que el personal evaluador, tenga la posibilidad de cambiar sus datos personales, su nombre de usuario o bien su password.
<i>Precondición</i>	El usuario se ha autenticado como miembro del grupo personal evaluador.
<i>Poscondición</i>	El registro de los datos personales de un profesor es alterado.
<i>Procedimiento</i>	El profesor, a través de un formulario, modifica sus datos personales.

Tabla 2.6 Caso de Uso: Definir/ Eliminar/ Editar Prueba. Personal Evaluador.

<i>Título</i>	Definir / Eliminar / Editar Prueba
<i>Propósito</i>	Definir el contenido de una prueba, o bien, editar o eliminar el contenido de una ya existente.
<i>Precondición</i>	El usuario se ha autenticado como miembro del grupo personal evaluador.
<i>Poscondición</i>	Se contará con una nueva definición de prueba, o se eliminará una ya existente
<i>Procedimiento</i>	<ol style="list-style-type: none">1. Se selecciona de un menú, el número de grupo y la clave de la asignatura para crear, editar o eliminar una prueba perteneciente a dicho grupo.2. A través de una interfase especializada se realiza cualquiera de las tareas anteriores.3. Si se crea una prueba o se edita una existente, se crea o se actualiza el registro correspondiente en la base de datos.4. Si se elimina una prueba existente, se elimina el registro correspondiente de la base de datos.

Tabla 2.7 Caso de Uso: Crear/ Eliminar Grupos. Personal Evaluador.

<i>Título</i>	Crear / Eliminar Grupos
<i>Propósito</i>	Que el personal evaluador defina grupos a través de la especificación de un número de grupo y la clave de la asignatura, para que posteriormente se puedan definir pruebas y asociar alumnos a los mismos.
<i>Precondición</i>	El usuario se ha autenticado como miembro del grupo personal evaluador.
<i>Poscondición</i>	Se contará con un nuevo grupo, para el cual se podrán definir pruebas o bien se podrán asociar alumnos al mismo.
<i>Procedimiento</i>	A través de una interface especializada el profesor podrá dar de alta / baja grupos.

Tabla 2.8 Caso de Uso: Asociar Alumnos a Grupo. Personal Evaluador.

<i>Título</i>	Asociar Alumnos a Grupo
<i>Propósito</i>	Que el personal evaluador, pueda dar de alta a uno o varios alumnos para que tengan acceso a las pruebas definidas para dicho grupo.
<i>Precondición</i>	<ol style="list-style-type: none"> 1. El usuario se ha autenticado como miembro del grupo personal evaluador. 2. El profesor ha definido grupos con anterioridad.
<i>Poscondición</i>	Se asociará(n) uno o varios alumnos a un grupo en particular.
<i>Procedimiento</i>	<ol style="list-style-type: none"> 1. El profesor seleccionará de un menú el grupo para el cual desea dar de alta / baja alumnos. 2. A través de una interface especializada el profesor dará de alta/baja a uno o varios alumnos.

Tabla 2.9 Caso de Uso: Consultar Calificaciones. Personal Evaluador.

<i>Título</i>	Consultar Calificaciones
<i>Propósito</i>	Que el personal evaluador, tenga acceso a los resultados obtenidos al resolver una prueba.
<i>Precondición</i>	<ol style="list-style-type: none"> 1. El usuario se ha autenticado como miembro del grupo personal evaluador. 2. El profesor ha definido grupos con anterioridad. 3. El profesor ha definido pruebas para un grupo en particular.
<i>Poscondición</i>	Se podrán consultar la calificación de alguna prueba en particular.
<i>Procedimiento</i>	<ol style="list-style-type: none"> 1. El profesor selecciona de un menú el grupo para el cual desea ver los resultados. 2. De un segundo menú, se selecciona una prueba en particular.

Capítulo 2: Análisis de Requerimientos

3. A través de una interfase especializada se podrán consultar los resultados.

Grupo II: Personal a Evaluar

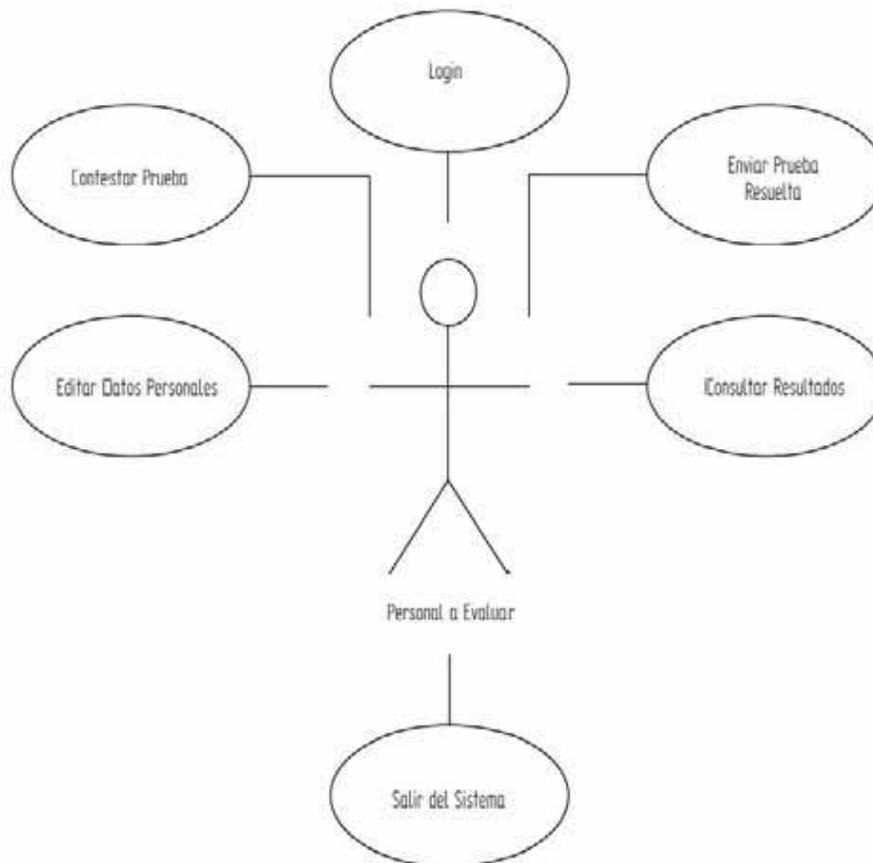


Figura 2.2 Diagrama de Casos de Uso: Personal a Evaluar.

Tabla 2.10 Caso de Uso Login Personal a Evaluar.

Título	Login
Propósito	Determinar si el usuario en cuestión, pertenece al grupo de usuarios que pueden contestar una o varias pruebas.
Precondición	El usuario ha sido registrado por su profesor en algún grupo.
Poscondición	El usuario tendrá acceso al menú: personal a evaluar.
Procedimiento	<ol style="list-style-type: none">1. El usuario teclea un nombre de usuario y una contraseña.2. Se verifica en la base de datos del sistema que el usuario pertenezca al grupo del personal a evaluar.

	<ol style="list-style-type: none"> 3. Si la condición previa es verdadera, se le presenta al usuario un menú, en el que podrá ejecutar tareas propias del personal a evaluar. 4. Si los datos ingresados no pertenecen a algún registro de la base de datos, se le niega el acceso al usuario.
--	--

Tabla 2.11 Caso de Uso: Editar Datos Personales. Personal a Evaluar.

<i>Título</i>	Editar datos personales
<i>Propósito</i>	Que el personal a evaluar, tenga la posibilidad de cambiar sus datos personales, su nombre de usuario o bien su password.
<i>Precondición</i>	El usuario se ha autenticado como miembro del grupo personal a evaluar.
<i>Poscondición</i>	El registro de los datos personales de un alumno es alterado.
<i>Procedimiento</i>	El alumno, a través de un formulario, modifica sus datos personales.

Tabla 2.12 Caso de Uso: Contestar/ Enviar Prueba. Personal a Evaluar.

<i>Título</i>	Contestar / Enviar Prueba
<i>Propósito</i>	Que el personal a evaluar pueda resolver una prueba y enviarla al servidor, para que pueda ser evaluado.
<i>Precondición</i>	<ol style="list-style-type: none"> 1. El usuario ha sido registrado por su profesor en algún grupo. 2. Que la prueba a ser resuelta, no haya sido contestada con anterioridad por el usuario, a menos que el profesor así lo permita.
<i>Poscondición</i>	Se contará con el registro de la calificación, la hora y la fecha en que fue resuelta la prueba.
<i>Procedimiento</i>	<ol style="list-style-type: none"> 1. El alumno selecciona de un menú, alguna prueba perteneciente a algún grupo en el que se encuentre inscrito y que todavía no haya contestado. 2. A través de un formulario que representa la prueba, el alumno selecciona las respuestas que él considera correctas dentro del tiempo definido para contestar la misma. 3. Si ha terminado de contestar la prueba, y aún queda tiempo en el reloj, podrá enviar sus respuestas al servidor mediante algún control contenido dentro del formulario. 4. Si el tiempo en el reloj se agotó, se cierra el formulario y se envían las repuestas al servidor.

Capítulo 2: Análisis de Requerimientos

Tabla 2.13 Caso de Uso: Consultar Calificaciones. Personal a Evaluar.

<i>Título</i>	Consultar Calificaciones
<i>Propósito</i>	Que el personal a evaluar, tenga acceso a los resultados obtenidos al resolver una prueba.
<i>Precondición</i>	<ol style="list-style-type: none">1. El usuario se ha autenticado como miembro del grupo personal a evaluar.2. El alumno ha sido registrado por su profesor en algún grupo.3. El alumno haya contestado alguna(s) prueba(s) para dicho(s) grupo(s).
<i>Poscondición</i>	Se podrán consultar la calificación de alguna prueba en particular.
<i>Procedimiento</i>	<ol style="list-style-type: none">1. El alumno selecciona de un menú el grupo para el cual desea ver los resultados.2. De un segundo menú, se selecciona una prueba en particular.3. A través de una interfase especializada se podrán consultar los resultados.

Grupo III: Personal Administrador

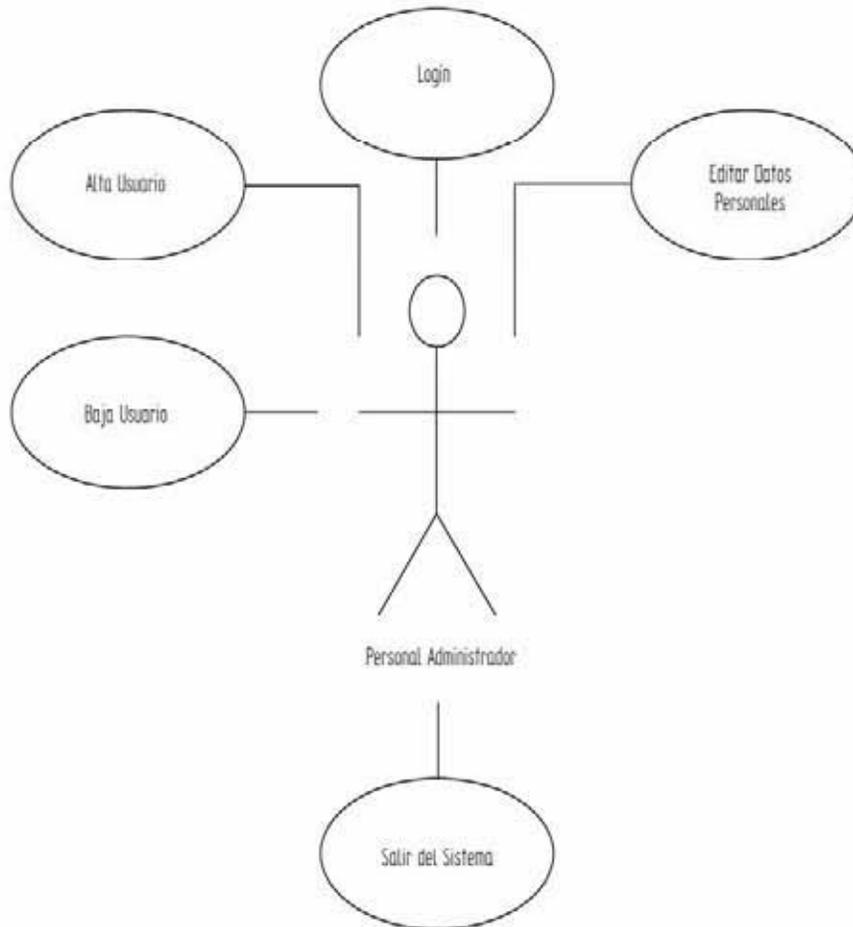


Figura 2.3 Diagrama de Casos de Uso: Personal Administrador.

Tabla 2.14 Caso de Uso: Login Administrador del sistema.

Título	Login
Propósito	Determinar si el usuario en cuestión, es el administrador del sistema.
Precondición	El usuario posee la cuenta de administrador.
Poscondición	El usuario tendrá acceso al menú: personal administrativo.
Procedimiento	<ol style="list-style-type: none"> 1. El usuario teclea un nombre de usuario y una contraseña. 2. Se verifica en la base de datos del sistema que el usuario sea el administrador. 3. Si la condición previa es verdadera, se le presenta al usuario un menú, en el que podrá ejecutar tareas

Capítulo 2: Análisis de Requerimientos

	propias del personal administrativo. 4. Si los datos ingresados no pertenecen a algún registro de la base de datos, se le niega el acceso al usuario.
--	--

Tabla 2.15 Caso de Uso: Editar Datos Personales. Administrador del Sistema.

<i>Título</i>	Editar datos personales
<i>Propósito</i>	Que el personal administrativo, tenga la posibilidad de cambiar sus datos personales, su nombre de usuario o bien su password.
<i>Precondición</i>	El usuario se ha identificado como miembro del grupo personal administrador.
<i>Poscondición</i>	El registro de los datos personales del administrador es alterado.
<i>Procedimiento</i>	El administrador, a través de un formulario, modifica sus datos personales.

Tabla 2.16 Caso de Uso: Alta/ Baja del Grupo Personal Evaluador. Administrador del Sistema.

<i>Título</i>	Alta / Baja miembro del grupo personal evaluador
<i>Propósito</i>	Dar de alta / baja a un miembro del grupo personal evaluador.
<i>Precondición</i>	El usuario se ha autenticado como administrador del sistema.
<i>Poscondición</i>	Se creará o se eliminará una cuenta de algún miembro del grupo: personal evaluador
<i>Procedimiento</i>	<ol style="list-style-type: none">1. Si es una alta, a través de un formulario, se ingresan los datos personales, el nombre de usuario y el password de algún miembro del grupo: personal evaluador.2. Si es una baja, a través de un área de selección se elimina la cuenta deseada.

2.3. Especificación de Diccionario de Datos

La especificación de la información de diccionario produce la lista de los objetos principales de información, identificados durante la recolección de requerimientos.

Tabla 2.17 Descripción Objeto Prueba.

<i>Nombre</i>	Prueba
<i>Sinónimos</i>	Cuestionario, examen.
<i>Descripción</i>	Una serie de preguntas con sus respectivos incisos, acerca de algún tema.
<i>Propiedades</i>	-Nombre del Instrumento. - Contenido

	<ul style="list-style-type: none"> * Encabezado <ul style="list-style-type: none"> • Referencia Institucional. • Nombre del Instrumento. • Instrucciones • Tiempo de Resolución. * Preguntas. <ul style="list-style-type: none"> • Incisos. - Fecha de Creación. - Fecha de Modificación.
<i>Relaciones</i>	<ul style="list-style-type: none"> • Prueba-Grupo: Relaciona a un grupo con sus respectivas pruebas. • Prueba-PruebaResuelta: La subentidad pruebaResuelta especializa la entidad Prueba, y contiene la información a cerca de quién, cuándo y cuál fue el resultado obtenido al resolver una instancia de Prueba.
<i>Componentes</i>	Ninguno

Tabla 2.18 Descripción Objeto PruebaResuelta.

<i>Nombre</i>	PruebaResuelta
<i>Sinónimos</i>	Ninguno.
<i>Descripción</i>	Una subentidad de Prueba, representa una prueba que ha sido remitida al sistema por algún alumno.
<i>Propiedades</i>	<ul style="list-style-type: none"> * Calificación. * Fecha de Solución * Hora de Solución.
<i>Relaciones</i>	<ul style="list-style-type: none"> • Prueba-PruebaResuelta: Especializa la entidad Prueba, contiene información a cerca de quién, cuándo y cuál fue el resultado, al resolver una instancia de Prueba.
<i>Componentes</i>	Ninguno

Tabla 2.19 Descripción Objeto Grupo.

<i>Nombre</i>	Grupo
<i>Sinónimos</i>	Ninguno.
<i>Descripción</i>	Una entidad que representa un grupo de una asignatura en particular.
<i>Propiedades</i>	<ul style="list-style-type: none"> * No de Grupo. * Clave de la Asignatura
<i>Relaciones</i>	<ul style="list-style-type: none"> • Grupo-Prueba: Relaciona un grupo con sus respectivas pruebas. • Grupo-Profesor: Relaciona un grupo con su

Capítulo 2: Análisis de Requerimientos

	respectivo profesor. <ul style="list-style-type: none">• Grupo-Alumno: Asocia un conjunto de alumnos con su respectivo grupo
<i>Componentes</i>	Ninguno

Tabla 2.20 Descripción Objeto Profesor.

<i>Nombre</i>	Profesor
<i>Sinónimos</i>	Instructor.
<i>Descripción</i>	Una entidad que representa a un miembro del grupo personal evaluador.
<i>Propiedades</i>	* Nombre. * Apellido. * email. * Nombre de Usuario. * Password.
<i>Relaciones</i>	<ul style="list-style-type: none">• Profesor-Grupo: Relaciona un profesor con sus respectivos grupos.
<i>Componentes</i>	Ninguno

Tabla 2.21 Descripción Objeto Alumno.

<i>Nombre</i>	Alumno
<i>Sinónimos</i>	Ninguno.
<i>Descripción</i>	Una entidad que representa a un miembro del grupo personal a evaluar.
<i>Propiedades</i>	* Nombre. * Apellido. * email. * Nombre de Usuario. * Password.
<i>Relaciones</i>	<ul style="list-style-type: none">• Alumno-Grupo: Relaciona un alumno con los grupos en los que se encuentra inscrito.• Alumno-PruebaResuelta: Permite al alumno consultar los resultados de las pruebas que ha resuelto.
<i>Componentes</i>	Ninguno

Tabla 2.22 Descripción Objeto Administrador.

<i>Nombre</i>	Administrador
<i>Sinónimos</i>	Ninguno.
<i>Descripción</i>	Una entidad que representa a un miembro del grupo personal administrador.
<i>Propiedades</i>	* Nombre.

	* Apellido. * email. * Nombre de Usuario. * Password.
Relaciones	Ninguna
Componentes	Ninguno

2.4. Especificación de Vistas del Sitio

La especificación de vistas del sitio, identifica las listas de vistas de un sitio necesarias para satisfacer los requerimientos de los grupos identificados. Las entradas de una especificación de vista del sitio, es la lista de grupos de usuarios, la lista de casos de uso y el diccionario de datos. Una vista del sitio sirve a los casos de uso asociados a uno o más grupos de usuarios, y ofrece acceso a funciones de gestión de contenido sobre los elementos seleccionados.

Las siguientes especificaciones de vistas del sitio corresponden a los grupos principales del sistema Web, para el personal evaluador, el personal a evaluar y para el personal administrador. Además de incluyen los mapas de vistas del sitio que agrupan los casos de uso para cada grupo de usuarios.

Nombre: *Interfaz Personal evaluador*

Descripción: Permite crear y administrar pruebas con sus respectivas acciones en reactivos, como insertarlos, editarlos y eliminarlos de la prueba. Además permite crear y eliminar grupos, así como asignar y eliminar alumnos de dichos grupos.

Grupos de usuarios: Personal Evaluador.

Casos de uso: Crear, abrir, eliminar y guardar prueba. Insertar, editar y eliminar reactivos. Asociar/desasociar usuarios Alumno a grupos específicos. Dar de alta y baja grupos. Consultar resultados y login.

Nombre del área	Descripción	Objetos accedidos /administrados	Nivel de prioridad
1. Modificar datos. (MPerfilEva)	Modifica datos personales del Personal Evaluador.	Profesor	Media
2. Alta-baja de grupos. (AdmGrupos)	Agrega o elimina grupos y sus respectivas	Grupos	Alta

Capítulo 2: Análisis de Requerimientos

3. Asociar alumnos a grupos. (AsigAlGpo)	asignaturas. Asigna o agrega alumnos a grupos asociados a asignaturas, o bien, los elimina.	Alumnos Grupos	Alta
4. Crear-editar cuestionario. (GestCuestion)	Crea, edita, guarda y abre espacio de trabajo para cuestionarios. Además inserta, edita y elimina reactivos.	Cuestionarios	Media
5. Consulta calificaciones. (Gest-Calificación)	Consulta los resultados de las calificaciones obtenidas por los usuarios evaluados.	Prueba Resuelta	Baja

Nombre: *Interfaz Personal administrador*

Descripción: Permite agregar, eliminar y editar cuentas de usuario pertenecientes al grupo del personal evaluador. También permite editar la cuenta del administrador del sistema.

Grupos de usuarios: Administrador del sistema.

Casos de uso: Crear usuario Profesor, eliminar usuario Profesor, editar datos del administrador y login.

Nombre del área	Descripción	Objetos accedidos /administrados	Nivel de prioridad
1. Editar datos del administrador. (MPerfilAdm)	Modifica datos personales del Personal Administrador.	Administrador	Media
2. Gestión de usuarios, Profesor. (MPerfilUsuarios)	Agrega o elimina cuentas de usuario Profesor.	Profesor	Alta

Nombre: *Interfaz Personal a evaluar*

Descripción: Permite visualizar los resultados en las pruebas asignadas a cada grupo de usuarios Alumno. Además de modificar sus datos personales y acceder a la resolución de dichas pruebas.

Grupos de usuarios: Personal a evaluar.

Casos de uso: Editar datos personales, contestar prueba, enviar prueba resuelta, consultar resultados y login.

Nombre del área	Descripción	Objetos accedidos /administrados	Nivel de prioridad
1. Modificar datos del Alumno. (MPerfile)	Modifica datos personales del personal a evaluar.	Alumno	Media
2. Resuelve prueba. (ResuelvePrueba)	Resuelve los reactivos de la prueba en cuestión en tiempo límite.	Prueba PruebaResuelta	Alta
3. Consulta resultados. (Consulta-Calificación)	Consulta los resultados obtenidos en diversas pruebas previamente realizadas por este grupo de usuarios.	PruebaResuelta	Alta

2.5. Especificación de Lineamientos de Estilo

La especificación de lineamientos de estilo establece reglas para la presentación de las páginas, a ser utilizadas en la producción de las interfaces. La especificación de guías de estilo cubren los siguientes puntos:

- **Especificación de mallas estándar en las páginas.** Establece en una tabla las capas en las cuales el contenido estático y dinámico será organizado, tal tabla está arreglada de renglones, columnas y celdas. Además la especificación de la malla dicta el tamaño de varios elementos de la malla.

- **Especificación de la posición del contenido.** Establece reglas para asignar contenido estándar en posiciones seleccionadas dentro de la malla de la página. Los lineamientos obligan a los elementos con similar distribución en pantalla ser colocados en la misma posición, con el propósito de reducir la desorientación del usuario.

En nuestro caso, los pasos involucran a páginas principales del sistema Web, como lo son: la página para la gestión de pruebas, la página que asocia alumnos a grupos, y la página utilizada para la resolución de pruebas. Las siguientes figuras (2.4 a 2.6) ilustran las especificaciones de mallas para las páginas anteriormente mencionadas.

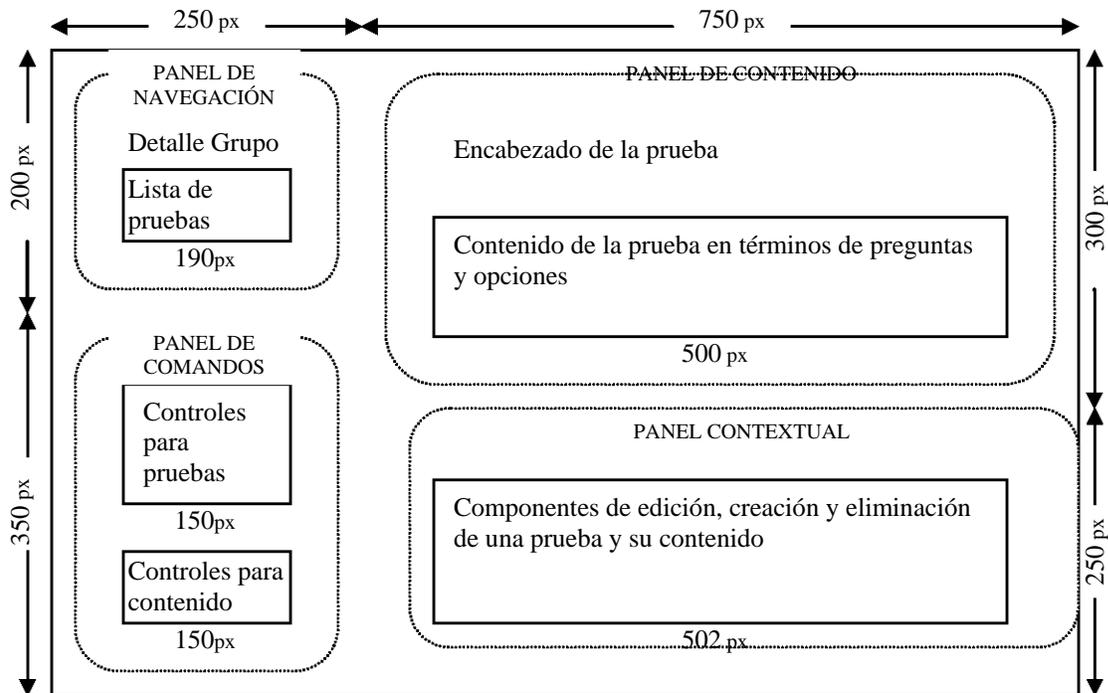


Figura 2.4 Especificación de la malla y posicionamiento del contenido de la página para la gestión de pruebas.

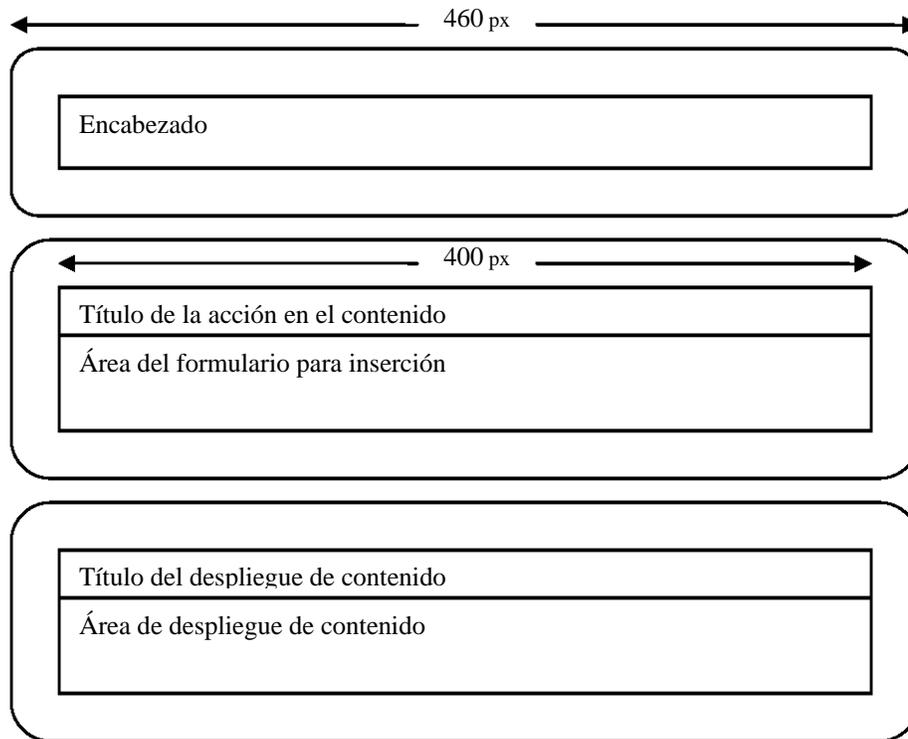


Figura 2.5 Especificación de la malla y posicionamiento del contenido de páginas asociadas a altas, bajas y despliegue de instancias.

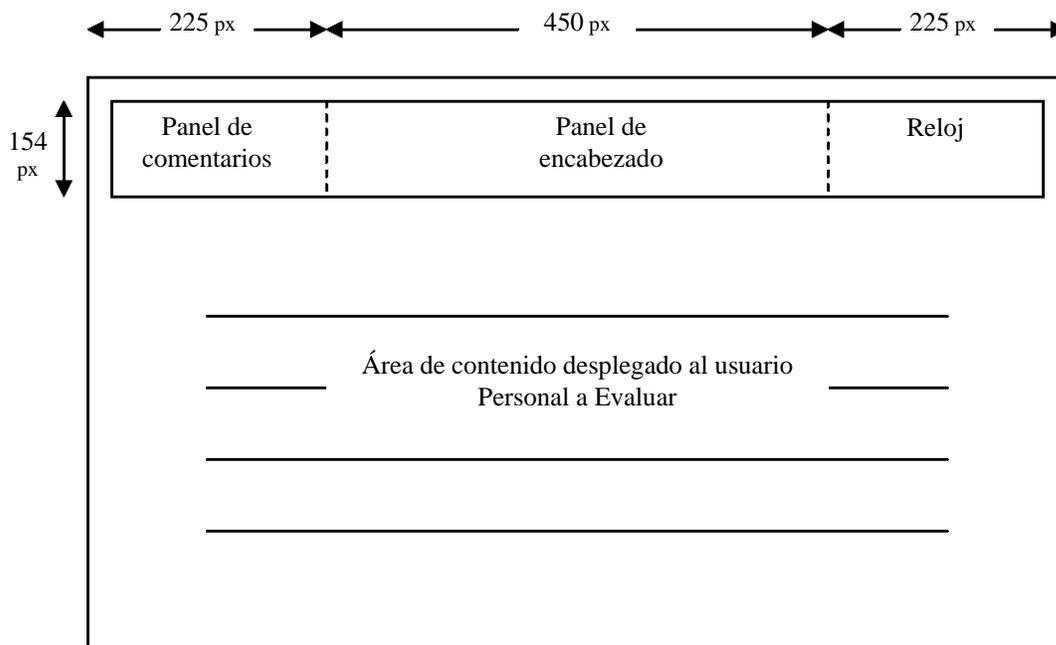


Figura 2.6 Especificación de la malla y posicionamiento del contenido de página asociada al despliegue de la prueba.

Capítulo 2: Análisis de Requerimientos

- **Lineamientos gráficos.** Refieren a las reglas de formato para elementos gráficos como fuentes, colores, bordes y márgenes. Las reglas son recurrentemente aplicados a elementos de página como textos, tablas, enlaces, menús, listas, entre otros más.

Los siguientes lineamientos gráficos corresponden a las vistas del sitio de los grupos principales de usuarios del sistema Web, tales lineamientos están descritos por medio de hojas de estilo en cascada o CSS¹. Los estilos descritos en las tablas ilustran de manera general las reglas de formato.

Reglas CSS para la vista del sitio Personal a Evaluar.
--

<pre>.bloque { background-color:#FFFFFF; border:ridge; border -color:#CCCCCC; border -width:medium; font-family:Verdana, Arial, Helvetica, sans-serif; font-size:12px; height:auto; width:auto; margin:5px; padding:10px; }</pre>

Reglas CSS para la vista del sitio Personal Evaluador.
--

<pre>#LayerCont { width:650px; height:150px; border:2px solid #006699; background color:#FFFFFF;} .botonosC { width: 80px; height: 25px; color:#000000; font-family: Verdana, Arial, Helvetica, sans-serif; font-size: 13px;} .textofield { color: #FFFFFF; font-family: Verdana, Arial, Helvetica, sans-serif; font-size: 12px; background-color: #990033; text-align: center; font-weight: bold;} #InsLyCu { font-family:sans serif,tahoma,arial,helvetica; font-size:14px; font-weight:bold; text-align:center} .txt { position:absolute; left:10px; top:0px; width:650px; font-family:tahoma,arial,helvetica,sans-serif; font-size:12px; text-align:justify;}</pre>

¹ Cascading Style Sheets, conjunto de reglas que sirven para definir el formato de los elementos del documento.

Reglas CSS para la vista del sitio Personal administrador.
.txt { Position:absolute; left:10px; top:0px; width:650px; font-family:tahoma,arial,Helvetica,sans-serif; font-size:12px; text-align:justify;}

2.6. Justificación de herramientas

La justificación de las herramientas de software usadas en el desarrollo de la aplicación, tiene por objeto, clarificar la razón por la cual, se eligieron las herramientas usadas en el momento de implementar la solución, a continuación se describen los criterios usados para la toma de dicha decisión:

- **Software Libre**

Nos encontramos claramente en la mitad de un océano de cambios en el negocio del software. La mayoría del software de consumo masivo es distribuido de manera gratuita hoy en día; clientes de e-mail, navegadores web, juegos, e incluso suites de oficina. El software de consumo, se está comercializando con pérdidas, bajo la premisa de que su uso, impulsará el consumo de otros productos o servicios, tales como: hardware de servidores, sistemas operativos, conectividad, publicidad, etc. El precio al menudeo de una pieza de software, de esta manera, no es un indicador confiable de su calidad o del nivel de excentricidad del usuario.

- **Costo**

PHP no cuesta. La combinación Apache / PHP / MySQL se ejecuta de manera eficiente en hardware relativamente barato, que ni siquiera se podría considerar usar para IIS / ASP / SQL Server.

A pesar de que MySQL ha sido licenciado como software de “código abierto” para algunas aplicaciones, nunca ha sido software desarrollado por la comunidad. MySQL AB es una entidad comercial con intereses comerciales. Aún así, existen muchas circunstancias en las que MySQL se puede usar de manera gratuita (básicamente nada redistribuible, lo cual cubre la mayoría de los proyectos PHP).

Para propósitos de comparación. La siguiente tabla muestra algunas figuras al menudeo para productos similares en los Estados Unidos. Todos los precios se encuentran en dólares y corresponden a servidores Web públicos de un procesador, con la herramienta de bases de datos usada más frecuentemente.

	ASP / SQL Server	Coldfusion Mx / SQL Server	JSP / Oracle	PHP / MySQL
Herramienta de Desarrollo	\$ 0 – 2499	\$599	0 – 2000	\$0 – 249

Capítulo 2: Análisis de Requerimientos

Servidor	\$ 999	\$ 2298	0 – 35,000	\$0
RDBMS	\$ 4999	\$ 4999	\$ 15,000	\$0 – 220

Fuente: PHP 5 & MySQL Bible, Willey 2004

- **Facilidad de Uso**

PHP es fácil de aprender, comparado con otras maneras de lograr una funcionalidad similar. A diferencia de las Páginas de Servidor Java (JSP) o de CGI basado en C, PHP no necesita un

conocimiento profundo de un lenguaje mayor, antes de que puedan hacerse peticiones triviales a una base de datos o a un servidor remoto.

Muchas de las funciones más útiles (tales como la de abrir una conexión a una base de datos Oracle o extraer un correo de un servidor IMAP) se encuentran predefinidas.

- **PHP es HTML embebido**

La capacidad de PHP para ser embebido en código HTML trae consigo las siguientes consecuencias:

- Php puede ser embebido en código producido por editores WYSIWYG (What you see is what you get).
- PHP contribuye a la división del trabajo entre diseñadores y programadores.
- Cada línea de HTML no necesita ser rescrita en algún lenguaje de programación.
- PHP puede reducir los costos de trabajo e incrementar la eficiencia, dado a que su curva de aprendizaje es poco profunda y a su facilidad de uso.
- PHP no necesita ser compilado a código binario antes de que pueda ser probado o usado.

- **Compatibilidad a través de Plataformas**

PHP y MySQL se ejecutan de manera nativa en cada distribución popular de UNIX (incluyendo Mac OS X) y Windows. Una cantidad bastante significativa de servidores http corren bajo estas dos clases de sistemas operativos.

PHP es compatible con los tres servidores Web líderes en el mercado: Servidor Apache http para Unix y Windows, Servidor de Información en Internet de Microsoft (IIS), y Servidor Empresarial Netscape. No es necesaria la compatibilidad entre el servidor Web y MySQL, ya que PHP se comunica directamente con el manejador de bases de datos.

- **Php no está basado en etiquetas**

PHP es un lenguaje de programación. Coldfusion², en contraste, es un conjunto de etiquetas predefinidas, como HTML. En PHP, se puede definir cualquier función con sólo teclear un nombre y una definición. En Coldfusion, se deben emplear etiquetas desarrolladas por otras personas o ir a través del proceso de Extender Etiquetas Personalizadas.

- **Estabilidad**

La palabra estable significa dos cosas diferentes en este contexto.

- El servidor no necesita ser re-inicializado con frecuencia.
- El software no cambia de manera radical, trayendo problemas de incompatibilidad de versión a versión.

Ambas connotaciones aplican tanto a MySQL como a PHP.

- **Velocidad**

PHP es bastante rápido en su ejecución, en particular cuando es compilado como un módulo de Apache en Unix. El servidor MySQL, una vez iniciado, ejecuta peticiones bastante complejas en un tiempo record.

A pesar de que no se cuentan con pruebas extensivas formales, existe evidencia anecdótica y muchas pruebas pequeñas sugieren que PHP es cuando menos tan rápido como ASP y que supera en desempeño a ColdFusion y a JSP en la mayoría de las aplicaciones.

- **Muchas Extensiones**

Php hace fácil la comunicación con otros programas y protocolos. La conexión a bases de datos es particularmente fuerte además de ODBC. PHP soporta un número grande de los protocolos mayormente usados como POP3, IMAP y LDAP. PHP 4 agregó soporte a Java y arquitecturas de objetos distribuidos (COM y CORBA), haciendo el desarrollo n-tier una posibilidad por primer vez. PHP5 extiende aún más este soporte, ofreciendo una librería incorporada de gráficos y soporte XML con DOM y simpleXML.

- **Ajax**

Ajax es la abreviatura de JavaScript Asíncrono y XML, es un término acuñado a principios del 2005 por Jesse James Garret, un autor y desarrollador Web. Hablando de manera estricta, Ajax es solamente una pequeña (aunque particularmente popular) parte de

² Coldfusion es una tecnología que permite desarrollar aplicaciones web.

Capítulo 2: Análisis de Requerimientos

JavaScript. El término, como es comúnmente empleado, no se refiere a una tecnología por sí misma, como en el caso de JavaScript.

En el esquema general de las cosas, lo que comúnmente es referido por Ajax es una combinación de las siguientes tecnologías.

- XHTML
- CSS (Hojas de Estilo en Cascada)
- El DOM (Modelo Objeto del Documento) accesible a través de JavaScript.
- XML, el formato de la información transferida entre el servidor y el cliente.
- XMLHttpRequest, objeto requerido para extraer información del servidor.

El beneficio principal de Ajax, es que la mayor parte del procesamiento requerido por la aplicación ocurre dentro del explorador del usuario, y que las peticiones de información al servidor usualmente son cortas. Así, con Ajax, se pueden ofrecer al usuario aplicaciones web basadas en datos, sin sacrificar el desempeño como en enfoques anteriores, los cuales requerían que el servidor regresara páginas completas en respuesta a las acciones del usuario.

Capítulo III

Diseño del SIAEROM

3.1 Diseño de Arquitectura

El diseño de arquitectura se concentra en la elección de hardware, red y componentes que integrarán al sistema, para encontrar la mezcla de estos componentes que satisfaga los requerimientos de la aplicación, y al mismo tiempo respete las restricciones económicas y técnicas del proyecto.

3.1.1 Introducción

El trabajo del diseñador es guiado por objetivos técnicos tales como el desempeño, escalabilidad, disponibilidad, preservación de estado y seguridad, y es restringido por factores económicos y organizacionales, incluyendo la disponibilidad y la habilidad de los recursos humanos, los costos permitidos, las políticas de outsourcing, la disponibilidad de hardware y licencias de software, etc.

3.1.2 Dimensiones del Diseño de Arquitectura

Esta sección contempla metas principales, restricciones y escenarios del diseño de arquitectura.

3.1.2.1 Objetivos del Diseño de Arquitectura

La definición de la arquitectura de la aplicación debe asegurar la prestación del nivel de servicio adecuado con respecto a las siguientes dimensiones:

Desempeño: La aplicación debe sostener la carga de trabajo esperada, expresada por parámetros tales como el número máximo de usuarios concurrentes, el número de peticiones de páginas por unidad de tiempo, o bien, el tiempo máximo para entregar una página al cliente.

Escalabilidad: la arquitectura debe ser extensible, de tal manera que cuando la carga de trabajo aumente, sea posible añadir poder de cómputo y mantener el desempeño estable.

Disponibilidad: La aplicación debe operar de manera continua, y las fallas no deben afectar de manera significativa el servicio entregado a los usuarios.

Preservación del Estado: el estado de interacción del usuario (representado, por ejemplo, por la información de sesión mantenida en el servidor) debe ser preservada, aún en el caso en el que la aplicación sea distribuida en muchas máquinas o se presenten fallas.

Seguridad: la información hospedada en el repositorio de datos y transmitida entre la aplicación y sus usuarios debe estar protegida, y los usuarios deben ser identificados y permitirles el acceso únicamente al contenido y a las funciones que necesitan para realizar su trabajo.

3.1.2.2 Restricciones en el Diseño de Arquitectura

Las siguientes variables afectan el proceso de toma de decisiones:

1. *Costo.* Cada recurso cuesta, por lo que toda configuración requiere una inversión diferente, en términos de procesadores, infraestructura de red, interfaces, y licencias de software. El presupuesto de la aplicación puede limitar la elección de los recursos de hardware y los productos de software.
2. *Complejidad:* algunas configuraciones son más sencillas de instalar y mantener que otras. La no disponibilidad o el costo de habilidades técnicas especializadas pueden restringir el diseño de arquitectura.
3. *Estándares corporativos e infraestructura:* la aplicación debe ser implantada sobre la infraestructura IT de la corporación, lo cual puede restringir la selección de los procesadores o de los productos de software.

3.1.2.3 Escenarios de Implantación de la Arquitectura

Una de las decisiones más importantes es dónde mantener la arquitectura diseñada. En tiempos recientes, se observa una tendencia hacia el outsourcing de la infraestructura IT, incluso en las compañías grandes. Tres diferentes escenarios se presentan:

Interno: la arquitectura es preservada dentro de la empresa y mantenida por el departamento interno de IT.

Housed: la arquitectura de la aplicación es mantenida por el departamento IT interno de la compañía, pero está físicamente instalada con un proveedor externo.

Hospedada: la arquitectura de la aplicación se encuentra localizada en las premisas de un proveedor externo, el cual, también la mantiene.

3.1.3 Diseñando la Arquitectura de Hardware y Red

La arquitectura más completa incluye cuatro componentes: el Servidor Web, la máquina de ejecución de los scripts, el servidor de aplicaciones, y el servidor de bases de datos.

Dada su simplicidad, y la restricción económica tanto de software como de hardware, a continuación se describe la arquitectura elegida, sobre la cual se implementó la aplicación:

Configuración de Servidor Único

En tal configuración, los componentes básicos de la arquitectura, el servidor Web, la máquina de ejecución de programas y la base de datos, se encuentran instalados en la misma máquina. El host reside en la Intranet y puede estar conectado a Internet vía un router, lo que permite a los usuarios de Internet hacer peticiones HTTP al servidor Web.

El router localizado entre el Internet y la Intranet actúa como un firewall, el cual es el elemento que separa el medio externo hostil de la red interna. El firewall está típicamente equipado con reglas de acceso / control, las cuales filtran los requerimientos de entrada y de salida y decide cuáles admitir y cuáles impedir. En la figura 3.1 se ilustra una arquitectura hardware y de red en configuración de un servidor único.

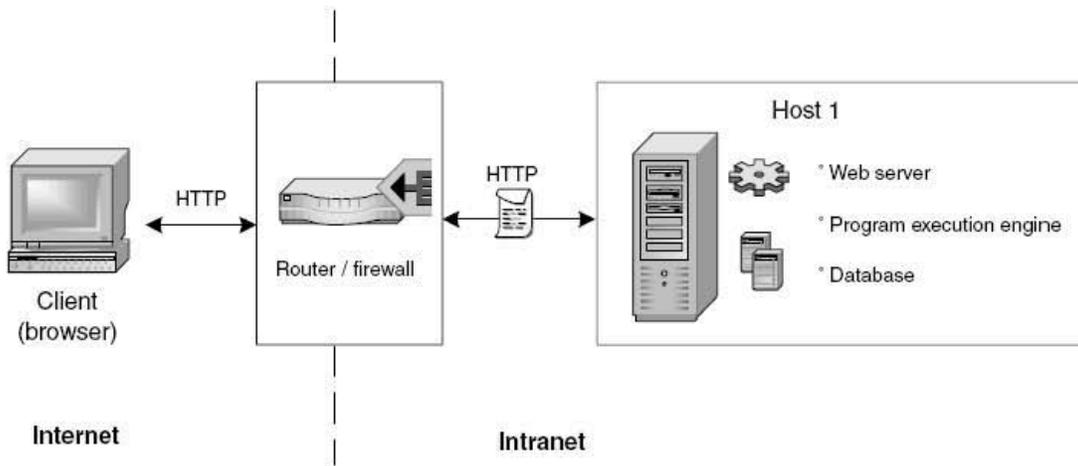


Figura 3.1 Configuración de Servidor Único

La configuración de servidor único puede ser evaluada a la luz de los objetivos del diseño de arquitectura y sus restricciones.

Desempeño: El desempeño de la arquitectura de servidor único, está limitada al hecho que la máquina de ejecución de scripts y el sistema manejador de la base de datos son aplicaciones memoria y CPU-intensivas, y por lo tanto existe un conflicto en el uso de los recursos de la máquina, lo cual puede producir un cuello de botella.

Escalabilidad: Escalar el sistema requiere agregar poder al servidor, o elegir un hardware más sofisticado, y aumentar el número de procesos concurrentes.

En general, las PCs y estaciones de trabajo, no están concebidas para multiparalelismo intensivo, por lo que ofrecen escalabilidad limitada.

Disponibilidad: En esta arquitectura cada elemento de software y de hardware constituye un punto de falla: si deja de funcionar, todo el sistema se detiene. La tolerancia a

las fallas puede mejorarse al incrementar el costo del sistema o su complejidad, agregando fuentes de hardware redundantes, por ejemplo, CPUs múltiples y discos espejo, e instalando múltiples procesos ejecutando diferentes instancias del servidor Web, la máquina de script, y la base de datos.

Mantenimiento del Estado: Los datos de sesión de usuario pueden ser almacenados en la memoria principal, por la máquina de ejecución del programa. Dado que existe únicamente un servidor, el cual procesa todas las peticiones de entrada, los datos de la sesión del usuario, están localmente disponibles en el servidor que se encuentra atendiendo la petición.

Seguridad: éste es el aspecto más débil de la arquitectura de servidor único, dado que los atacantes que hayan roto la seguridad del firewall y el servidor Web pueden tomar control del servidor y ganar acceso directo a la base de datos, violando la protección de datos.

Costo: el costo es bajo, siempre y cuando no se requiera paralelismo masivo. El agregar más procesadores, RAM y discos más rápidos al servidor, o cambiar a una clase de hardware más sofisticada, incrementa el costo, y muchas veces puede sobrepasar el gasto de una configuración alternativa, hecha de varios servidores de bajo costo.

Complejidad: La configuración de un servidor único es la opción más simple. En particular evita todas las tareas técnicas de montar una intranet compleja y de conectar la máquinas que contienen los diferentes puertos de la aplicación.

La solución de un servidor único, implementada en una computadora de bajo costo, es viable para aplicaciones Web de pequeña escala, no críticas. Aplicaciones Web, en las que la simplicidad y una inversión limitada son los objetivos más prominentes, en la cual la seguridad de los datos no es esencial.

3.2 Diseño de Datos

El diseño de datos es la actividad en la cual el conocimiento acerca de la aplicación es transformada en una descripción de componentes de software, como esquemas Entidad-Relación.

3.2.1 Identificación de las Entidades Núcleo

A partir del diccionario de datos definido durante la especificación de requerimientos, inicialmente se elige un objeto, el cual juega el papel de objeto nuclear, definido como *Prueba*. Por lo que la actividad de diseño de datos comienza con la inclusión de la entidad (Prueba) en el esquema de datos.

3.2.2 Diseño de Subesquemas Núcleo

En la segunda fase, el subesquema nuclear centrado en la entidad núcleo: *Prueba* es detallado.

La entidad *Prueba* tiene exclusivamente propiedades simples (Nombre, Fecha de Creación, Fecha de Modificación y Contenido) las cuales son modeladas a través de la inclusión de atributos del tipo apropiado en la entidad *Prueba*. Debido a que esta entidad no posee atributos multi-valorados (que comprendan varias propiedades) el subesquema nuclear de la aplicación consistirá exclusivamente del objeto nuclear. La figura 3.2 ilustra el *objeto Prueba* y sus propiedades representadas por un gráfico que incluye el nombre del objeto y la lista de propiedades que lo compone.

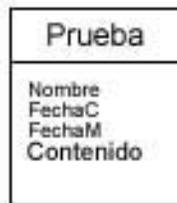


Figura 3.2 Objeto Nuclear de la Aplicación.

Cabe señalar, que el atributo denominado “Contenido” contendrá el texto que representa el contenido de la prueba, el cual ha sido estructurado mediante el uso de xml, para que pueda ser procesado al momento de leerlo por la aplicación.

3.2.3 Definición del Subesquema de Acceso

Al final de la primera fase del diseño de datos, el subesquema nuclear se encuentra definido. Posteriormente, el acceso al objeto nuclear debe ser especificado.

A partir de los casos de uso asociados a los usuarios *profesores* y a los usuarios *alumnos*, se deduce que el acceso a las *pruebas* debe estar implementado a través de un concepto de categorización. Por lo tanto, la entidad “*Grupo*” es introducida y es conectada a la entidad *Prueba*. La entidad *Grupo* está definida a través de los atributos “No de Grupo” y “Clave de la Asignatura”. Debido a que el propósito de incluir la entidad *Grupo* es el de proporcionar un acceso jerárquico a las Pruebas, no es necesario incluir demasiados atributos para representarla. La figura 3.3 muestra el subesquema de acceso a la entidad núcleo *Prueba* correspondiente al sistema en construcción (SIAEROM).



Figura 3.3 Sub-esquema de Acceso a Entidad Prueba

Un segundo elemento es introducido en el esquema de acceso del concepto *Prueba*: una subentidad *PruebaResuelta* que especializa a la entidad *Prueba*, la cual contendrá instancias que representan los resultados de pruebas ya resueltas, puesto que es necesario llevar un registro de quién ha resuelto una Prueba en particular, cuándo fue resuelta, y cuál fue el resultado obtenido a partir de la solución de dicha prueba. Todo esto, para soportar algunos de los casos de uso definidos en la especificación de requerimientos para los usuarios *profesores* y *alumnos*.

3.2.4 Definición de los Subesquemas de Interconexión

Debido a que únicamente se cuenta con un subesquema nuclear, no es posible definir conexiones asociadas a este subesquema.

3.2.5 Definición de los Subesquemas de Personalización

Hasta el momento, el modelo de la base de datos, representa la organización de las pruebas; sin embargo el principal objetivo de la aplicación es que los alumnos y profesores tengan acceso a las mismas. Por lo que las entidades *Alumno* y *Profesor* (definidas a través de los atributos simples: Nombre, Apellido, E-Mail, Nombre de Usuario y Password), se

encontrarán conectadas a través de relaciones a la entidad *Grupo* respectivamente, para que cada grupo de usuarios tenga acceso a las instancias de la entidad *Prueba* correspondientes.

También se define dentro de este sub-esquema la entidad *Administrador*, la cual no se encuentra relacionada con alguna entidad enunciada con anterioridad, sino que únicamente su función en la aplicación, es la de contener los datos personales y los datos de autenticación del usuario administrador de las cuentas otorgadas a los usuarios profesores. En la figura 3.4 se ilustra el subesquema de personalización de sistema SIAEROM.

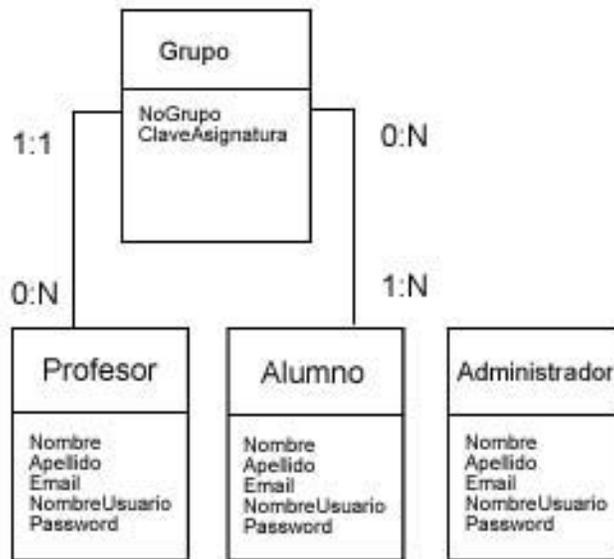


Figura 3.4 Subesquema de Personalización

Finalmente, el esquema Entidad-Relación del sistema SIAEROM es el mostrado en la figura 3.5:

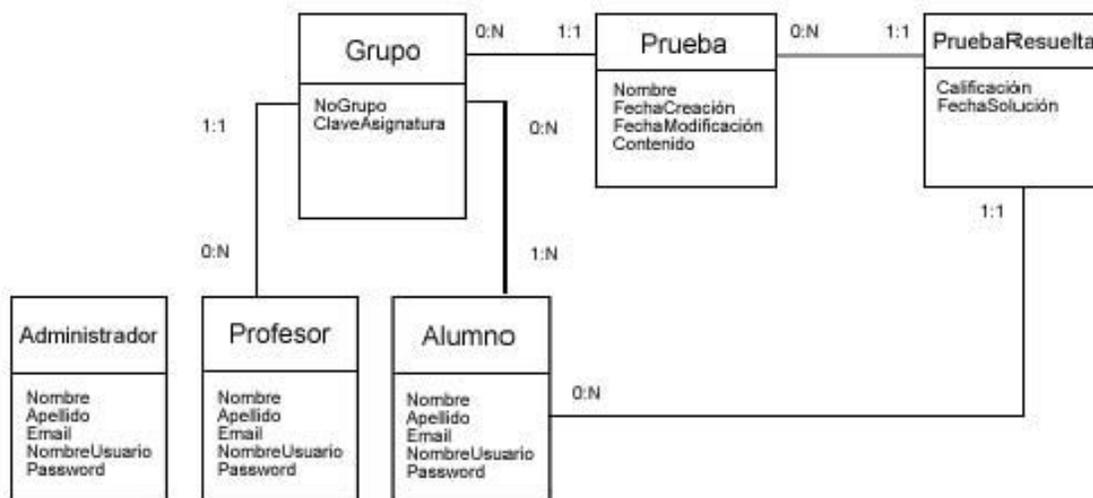


Figura 3.5 Esquema Entidad-Relación de la Aplicación

3.3 Diseño de Hipertexto

El diseño del hipertexto especifica las vistas del sitio a ser construidas en el esquema de datos definido en el diseño de datos, que nos ayuda a realizar la publicación de contenido y la manipulación de servicios identificados durante el análisis de requerimientos.

El diseño parte de dos fases: la fase elemental y la fase a detalle. A continuación, se procede a desarrollar el diseño del sistema Web, de acuerdo con los pasos descritos en cada fase.

3.3.1 Diseño Elemental

Establece el primer esquema de cada vista del sitio, mapea los elementos del esquema de datos a áreas de vistas del sitio.

El primer paso consiste en identificar las áreas. La identificación de las áreas empieza reexaminando los requerimientos funcionales y los mapas de las vistas, las cuales incorporan una idea premilitar de la división de la aplicación en submódulos. Una vez establecidas las áreas, el siguiente paso es expresar la visibilidad de cada área, las opciones son: landmark³, default⁴ e interna. Los esquemas siguientes, representados desde la figura

³ Landmark: término utilizado para indicar que un área es accesible desde otras áreas. Se utiliza la letra 'L' para representar a un área landmark.

⁴ Default: califica al área de acceso inicial sobre otras cuando se ingresa a la vista del sitio que la encierra. La letra 'D' representa a este tipo de área.

3.6 hasta la figura 3.8, representan las vistas del sitio considerando estos dos primeros pasos en cada grupo de usuarios del sistema.

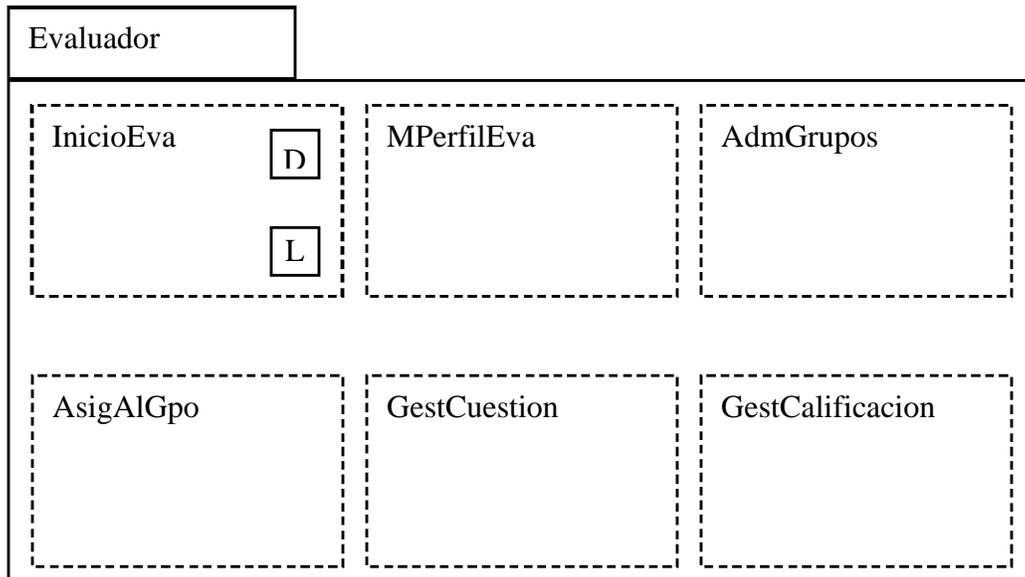


Figura 3.6 Vista del sitio para el grupo de usuarios *Personal Evaluador* con sus respectivas áreas y visibilidad

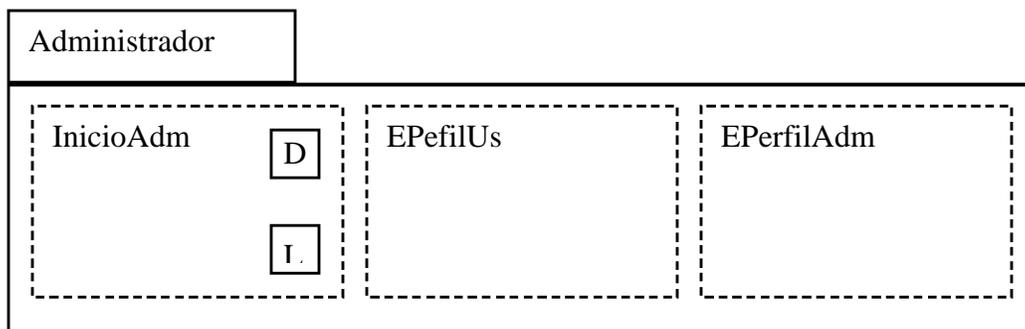


Figura 3.7 Vista del sitio para el grupo de usuarios *Administrador* con sus respectivas áreas y visibilidad

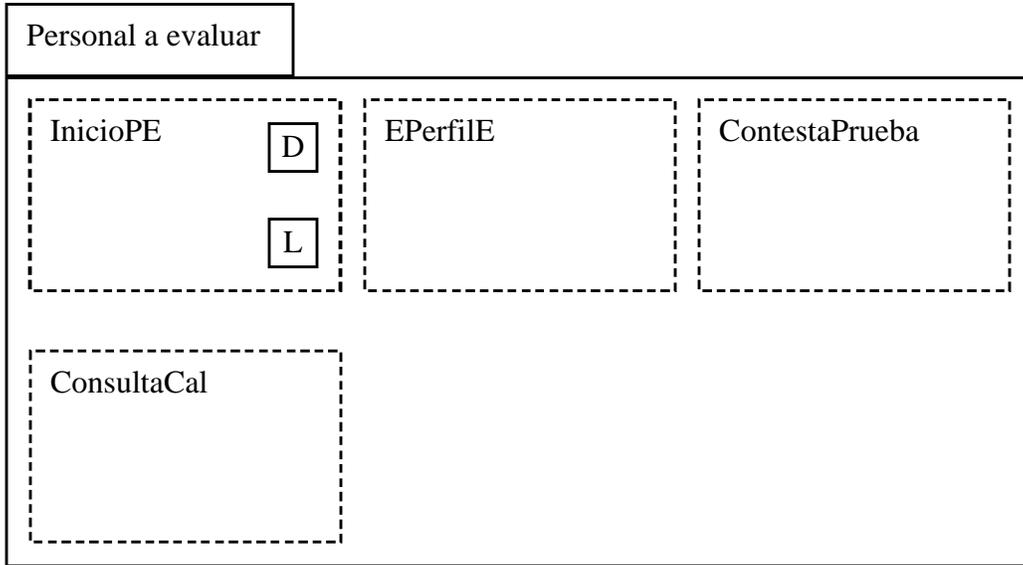


Figura 3.8 Vista del sitio para el grupo de usuarios *Personal a Evaluar* con sus respectivas áreas y visibilidad

El último paso del diseño elemental es la especificación del contenido asociado con cada área. Los mapas de las vistas del sitio realizados durante el análisis de requerimientos contienen una descripción textual informal de los servicios ofrecidos en cada área, esta descripción detalla el contenido y funciones correspondientes a cada área, ilustrando los roles que los objetos del esquema de datos juegan en la construcción del área. El contenido de un área está expresado en términos de *fragmentos del hipertexto*⁵, que representan acciones aplicadas a los objetos del esquema de datos, tanto a sus entidades como a sus relaciones.

3.3.2 Diseño a Detalle

Los esquemas bosquejados de las vistas del sitio son progresivamente revisados hasta que llegan a ser colecciones de páginas y unidades en notación WebML, las cuales obedecen a los requerimientos del usuario. El diseño a detalle implica realizar iteradamente los siguientes pasos: identificar páginas, especificar sus visibilidades, y especificar cada página su correspondiente notación en WebML.

Los primeros dos pasos del diseño a detalle requieren identificar las páginas y clasificarlas como home, landmark, o internas, contemplando así, la visibilidad de cada página. La identificación procede en dividir un área en páginas, cada página es asignada una porción de contenido y funciones asociadas al área encerrada, esta distribución separa características en común para cada página. Mientras que, la visibilidad de cada página determina la accesibilidad que poseen cuando se navega por el hipertexto.

⁵ Expresan procesos hacia las entidades para operar sus instancias y sus relaciones. Eso es para crear, editar, eliminar instancias de cierta entidad, así mismo para sus relaciones. Los fragmentos como *core*, *modify*, etc. son descritos en el glosario.

Los esquemas siguientes contemplan los cinco pasos descritos anteriormente, abarcan tanto el diseño elemental como el diseño a detalle. Estos esquemas incluyen *fragmentos del hipertexto* asociados con las funciones que realizarán en el sistema, la visibilidad de cada área, la división de cada área en páginas, y la visibilidad de cada página.

Los esquemas a continuación, correspondientes desde la figura 3.9 hasta la figura 3.14, pertenecen a las áreas de las vistas del sitio del *Personal Evaluador* describiendo los fragmentos de hipertexto correspondientes a cada página.

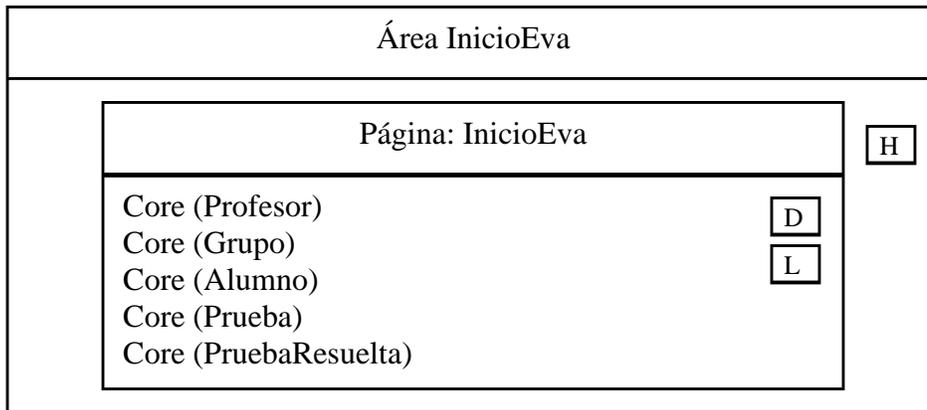


Figura 3.9 Área *InicioEva* correspondiente a la vista del sitio del *Personal Evaluador*

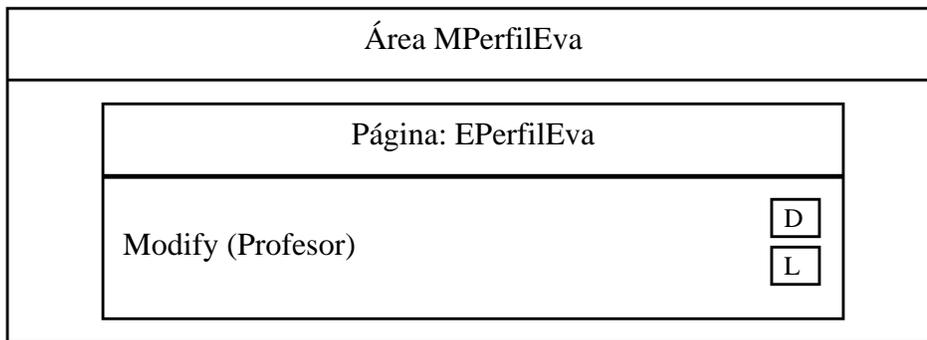


Figura 3.10 Área *MPerfilEva* correspondiente a la vista del sitio del *Personal Evaluador*

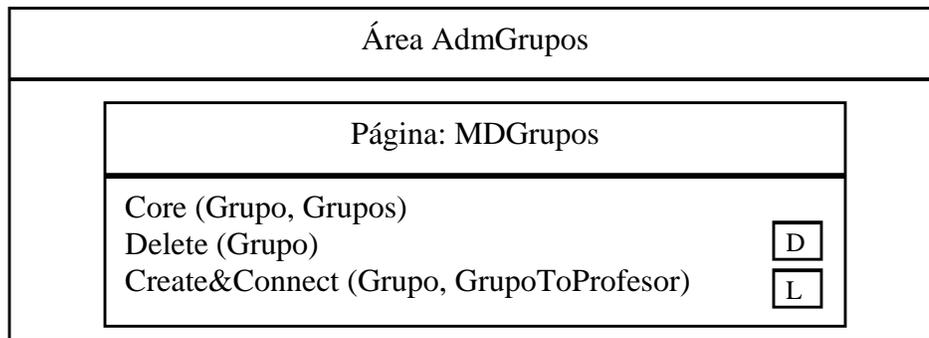


Figura 3.11 Área *MGrupos* correspondiente a la vista del sitio del *Personal Evaluador*

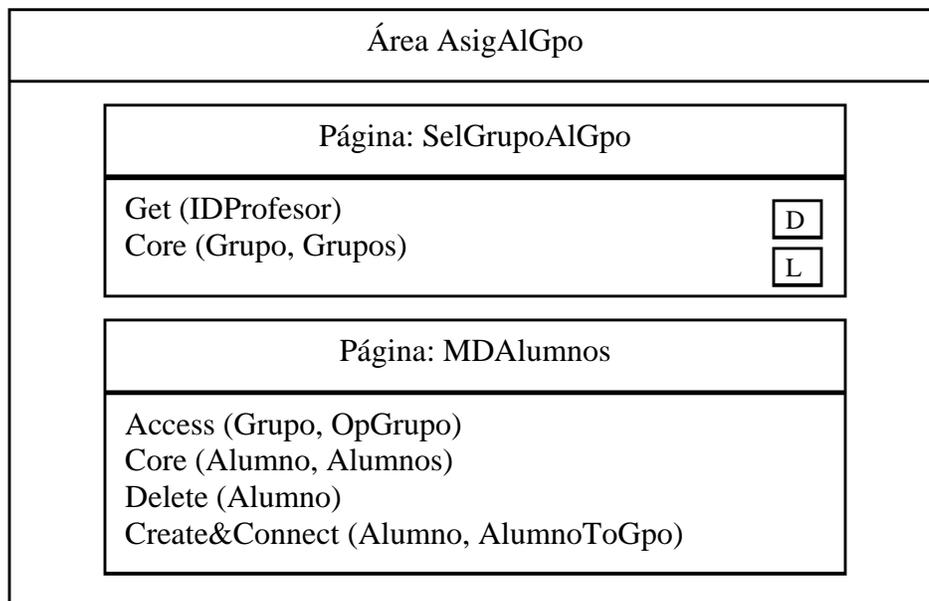


Figura 3.12 Área *AsigAlGpo* correspondiente a la vista del sitio del *Personal Evaluador*

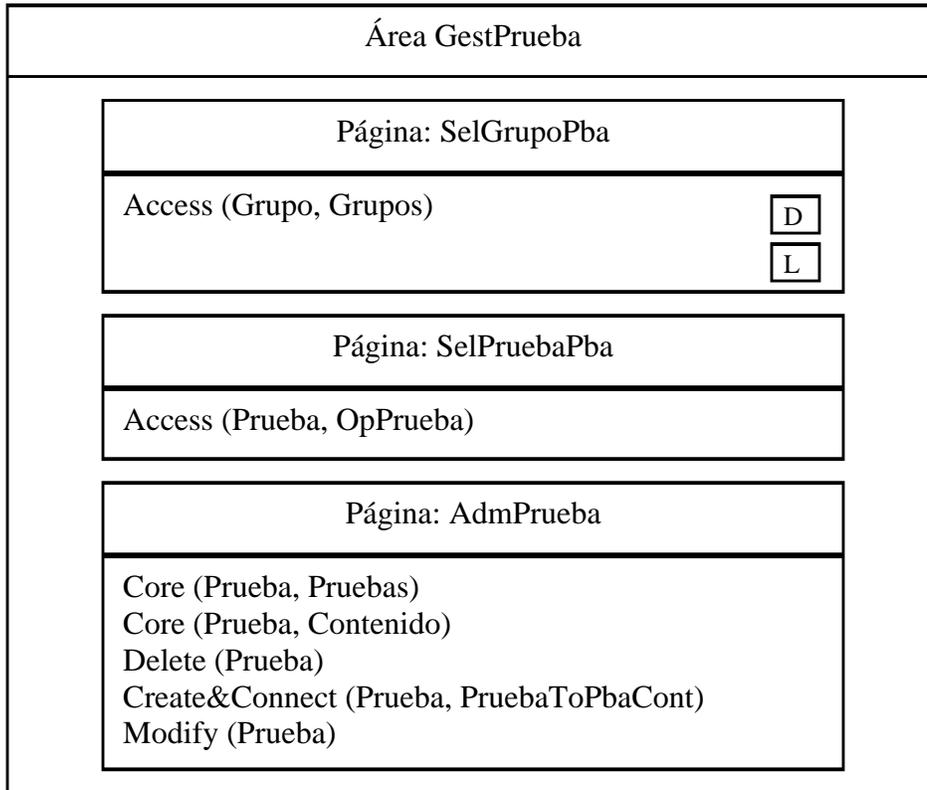


Figura 3.13 Área *GestPrueba* correspondiente a la vista del sitio del *Personal Evaluador*

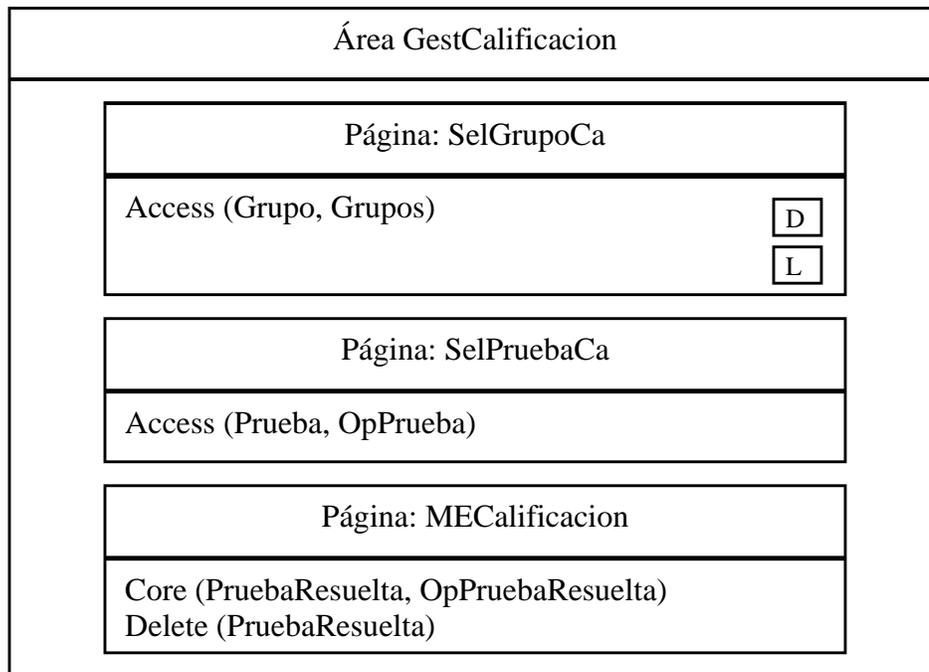


Figura 3.14 Área *GestCalificacion* correspondiente a la vista del sitio del *Personal Evaluador*

Los esquemas de las figuras 3.15, 3.16 y 3.17 pertenecen a las áreas de las vistas del sitio del *Personal Administrador*.

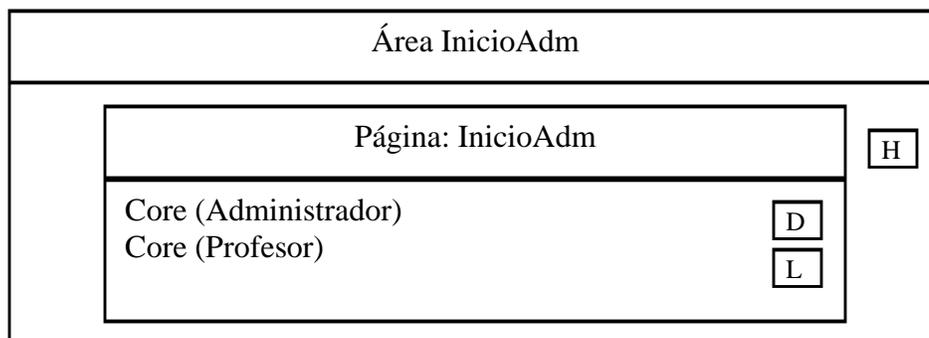


Figura 3.15 Área *InicioAdm* correspondiente a la vista del sitio del *Personal Administrador*

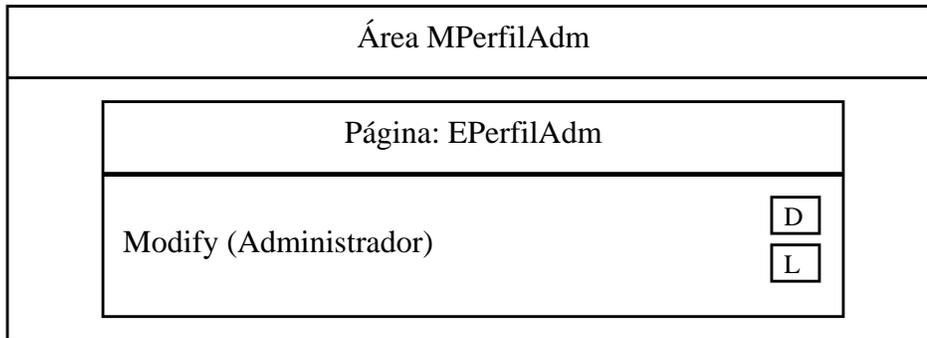


Figura 3.16 Área *MPerfilAdm* correspondiente a la vista del sitio del *Personal Administrador*

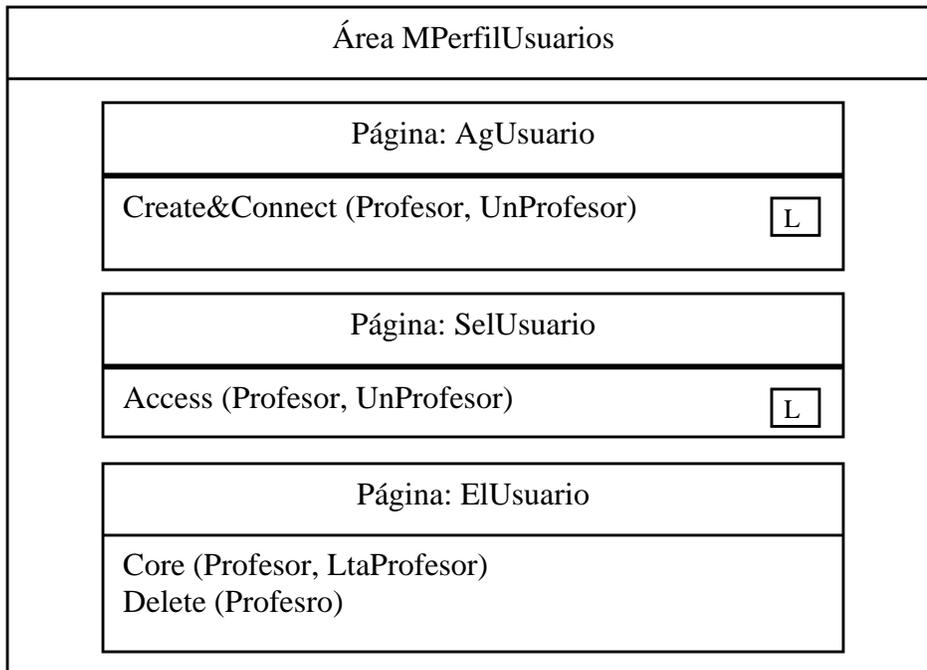


Figura 3.17 Área *MPerfilUsuarios* correspondiente a la vista del sitio del *Personal Administrador*

Los esquemas siguientes, comenzado de la figura 3.18 hasta la figura 3.21, pertenecen a las áreas de las vistas del sitio del *Personal a Evaluar*.

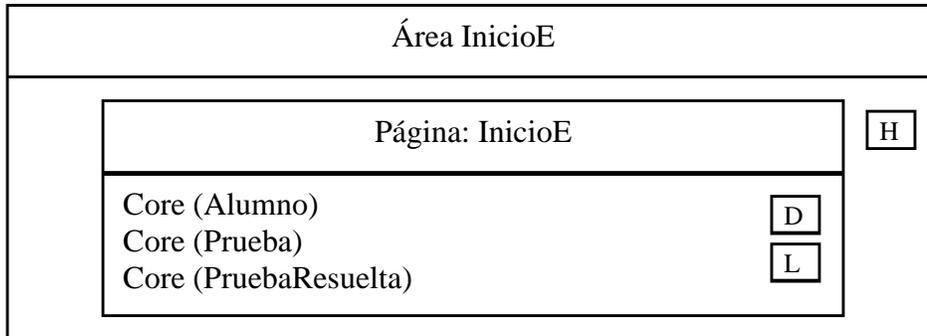


Figura 3.18 Área InicioAdm correspondiente a la vista del sitio del *Personal a Evaluar*

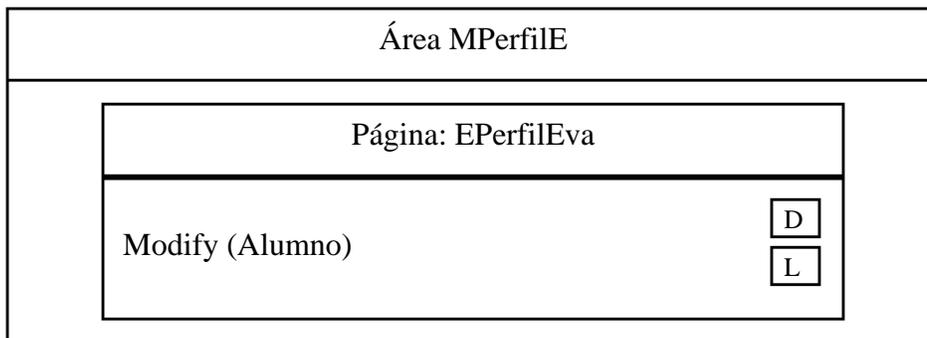


Figura 3.19 Área MPerfile correspondiente a la vista del sitio del *Personal a Evaluar*

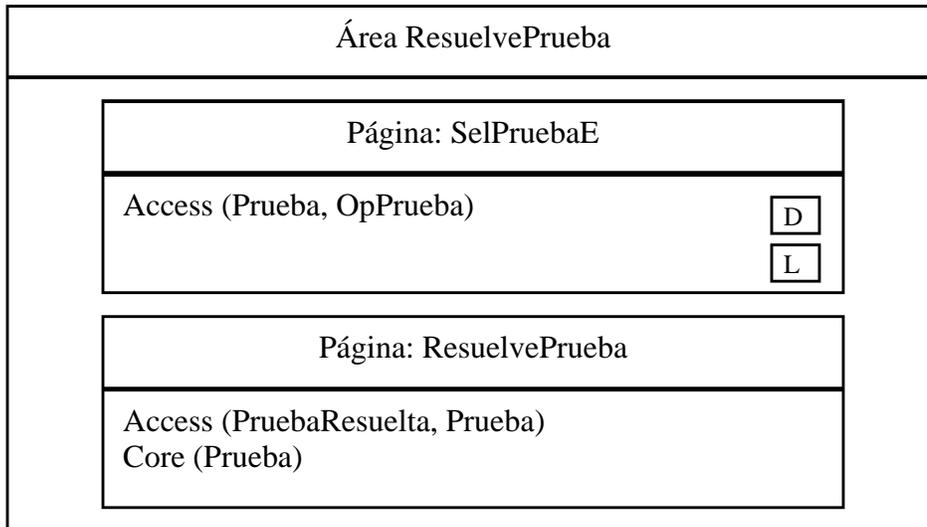


Figura 3.20 Área *ResuelvePrueba* correspondiente a la vista del sitio del *Personal a Evaluar*

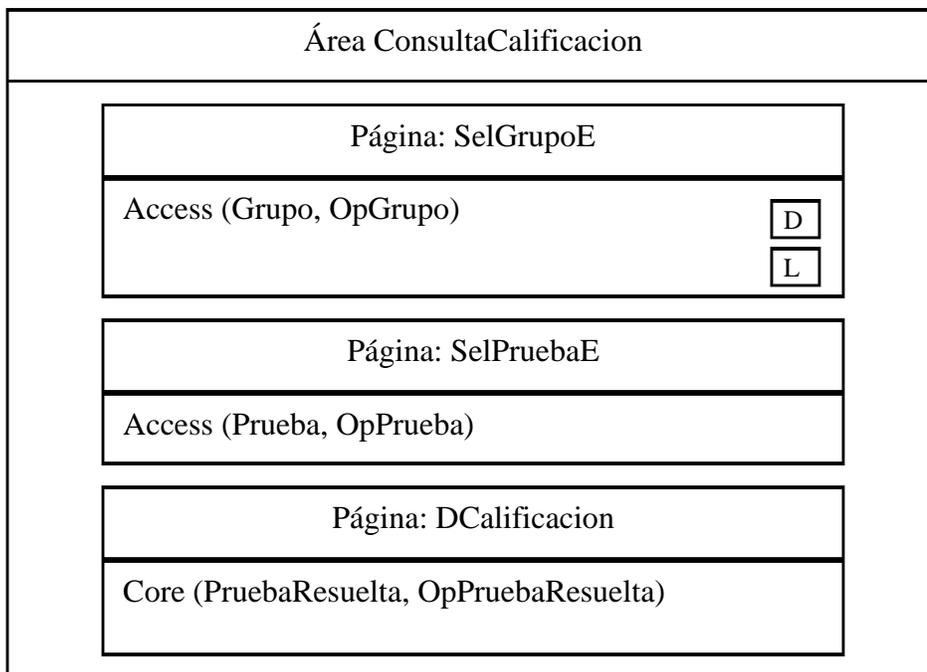


Figura 3.21 Área *ConsultaCalificacion* correspondiente a la vista del sitio del *Personal a Evaluar*

El tercer y último paso del diseño a detalle y de la fase de diseño es especificar las páginas identificadas en pasos anteriores en notación WebML, para ello, se requerirá los elementos descritos en apartados anteriores que denotan la estructura del hipertexto, como lo son las unidades y sus relaciones. La culminación de esta fase permite construir la versión en papel del sistema Web, para posteriormente con ayuda de lenguajes de programación se proceda a implementar dicha versión en algún lenguaje para Web, en nuestro caso, con la tecnología DHTML y Ajax que involucran los lenguajes HTML, hojas de estilo, Javascript, PHP y XML.

Las especificaciones en WebML del sistema no serán descritas en su totalidad, ya que la intención de los esquemas contemplados en este apartado es permitir modelar las partes importantes del sistema Web, además, de que algunas páginas son similares a otras en cuanto a la notación WebML, lo único, que cambiaría son las entidades a las que hacen referencia.

Recordando que, los esquemas del hipertexto ilustran la publicación y flujo de contenido a través del hipertexto, mas no la posición y volumen de tal contenido en el sistema Web, ya que, el diseño del hipertexto es un proceso abstracto y conceptual. En esta última fase del diseño, cada página contendrá en términos de unidades y relaciones la funcionalidad esperada con base en a los fragmentos de hipertexto bosquejados anteriormente.

3.3.2.1 Especificación de las Páginas en Notación WebML

A continuación se especifican las vistas del sitio para cada grupo de usuarios en notación WebML.

3.3.2.1.1 Vista del Sitio Personal Evaluador

Comenzando por especificar las páginas del Personal Evaluador. La única página del área *inicioEva* contiene el menú de opciones para este grupo de usuarios, las unidades índice representan las opciones correspondientes a cada entidad accesibles por estos usuarios, como lo son: Grupo, Prueba, Alumno, Profesor, PruebaResuelta y Grupo disponibles en otras áreas. El siguiente esquema en WebML de la figura 3.22 muestra la página *InicioEva*.

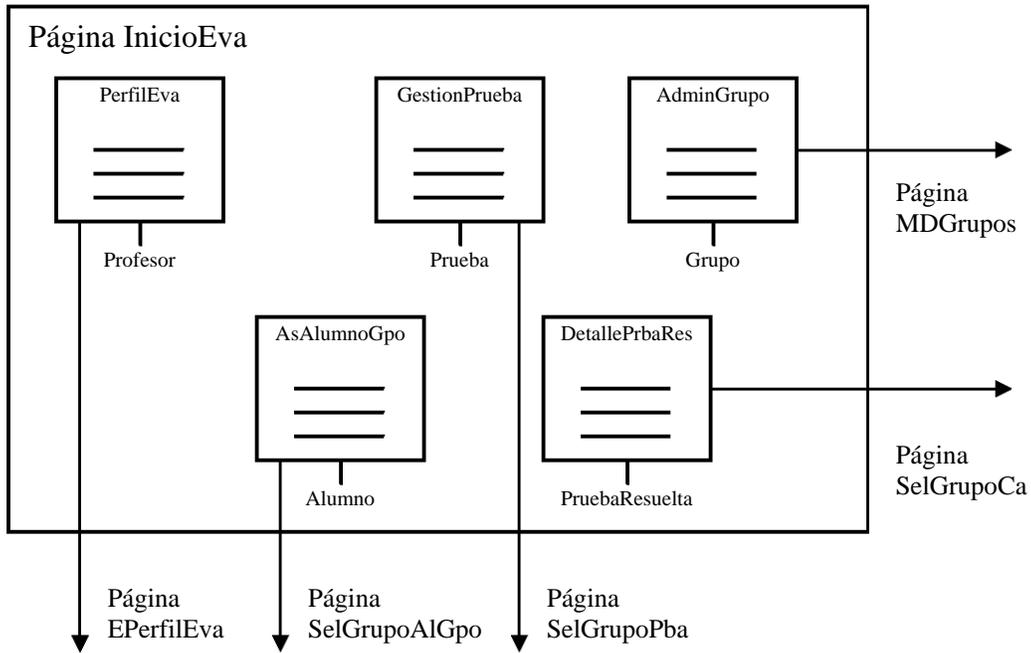


Figura 3.22 Página *InicioEva* del área *InicioEva* correspondiente a la vista del sitio del *Personal Evaluador*

La página *EPerfilEva* encerrada por el área con su nombre integra las unidades de operación y relaciones asociadas con la modificación del objeto perteneciente a la entidad Profesor, además contempla unidades de despliegue de contenido. La notación en WebML de esta página queda de la siguiente manera, mostrada en la figura 3.23:

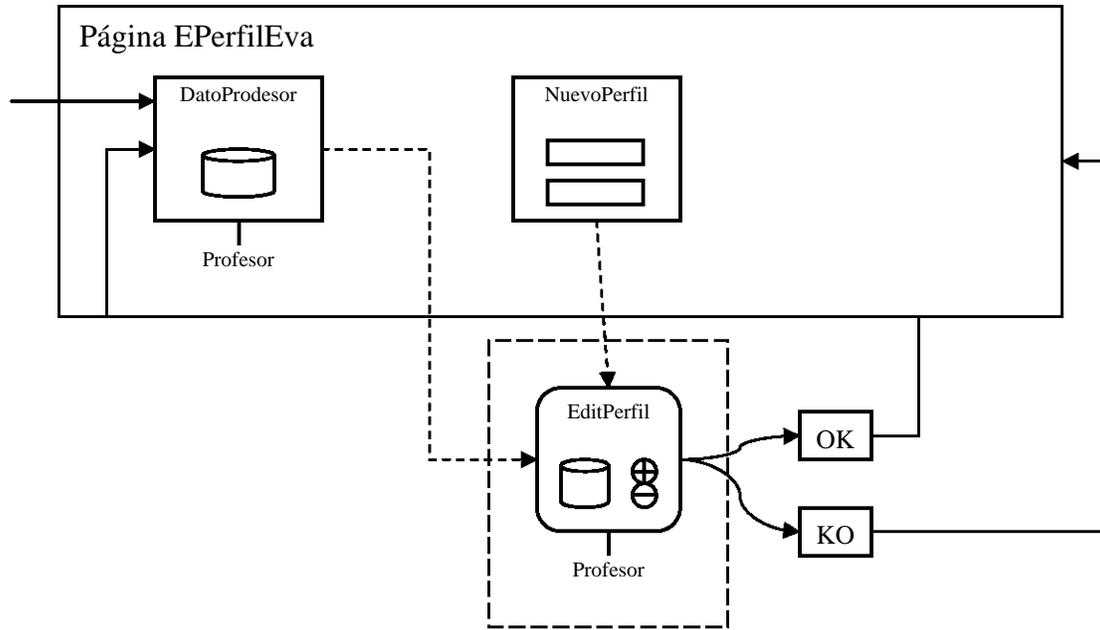


Figura 3.23 Página *EPerfilEva* del área *EPerfilEva* correspondiente a la vista del sitio del *Personal Evaluador*

En el caso del área *GestPrueba*, existen diversas páginas que posibilitan la visualización, creación, edición y eliminación de archivos que contienen las pruebas y sus respectivos reactivos. La página continuación del menú de opciones del grupo de usuarios *Personal Evaluador* es la página *SelGrupoPba*, la cual permite seleccionar el grupo disponible para posteriormente ingresar a la página *AdmPrueba*. En esta página el usuario *Personal Evaluador* dispone de varios paneles que dotan de las funcionalidades principales para gestionar las pruebas y sus respectivos reactivos.

La página *SelGrupoPba* está constituida por unidades de selección de objetos asociados a la entidad Grupo, mientras que la página *AdmPrueba* se compone de subpáginas asociadas a los diversos paneles para la gestión de pruebas. Recordando que el contenido del panel contextual está con base en a las elecciones del usuario en el panel de control, y en el caso del panel de contenido está de acuerdo con las acciones realizadas en los paneles de control, contextual y de navegación. Los siguientes dos esquemas en notación WebML de las figuras 3.245 y figura 3.25, muestran la página *AdmPrueba*, dividida en dos partes, una dedicada a la gestión de pruebas y la otra a la gestión del contenido de la prueba.

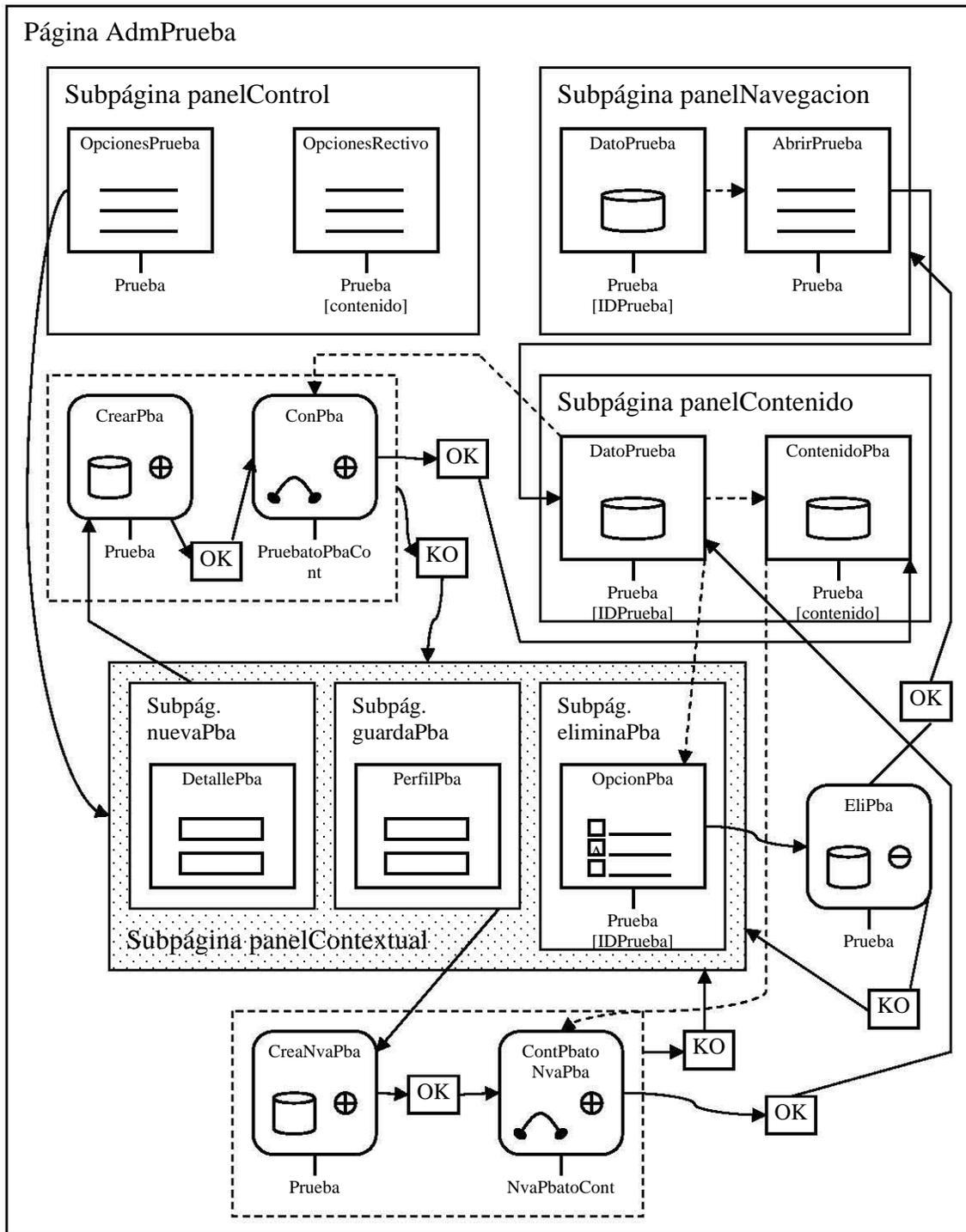


Figura 3.24 Primera parte de la página AdmPrueba del área GestCuestion correspondiente a la vista del sitio del Personal Evaluador

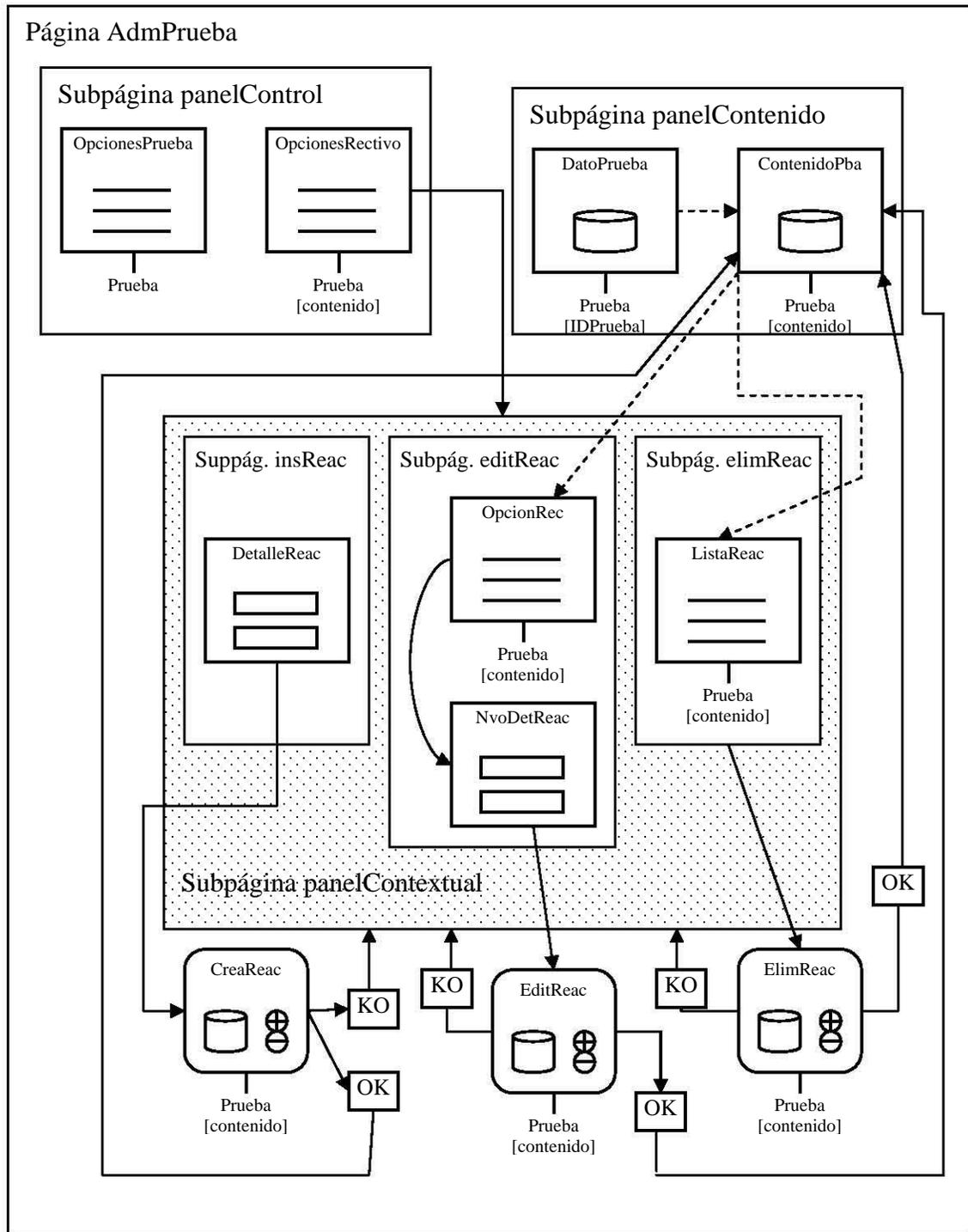


Figura 3.25 Segunda parte de la página *AdmPrueba* del área *GestCuestion* correspondiente a la vista del sitio del *Personal Evaluador*

La página *MDGrupos* perteneciente al área *AdmGrupos* añade y elimina instancias de la entidad *Grupos*, las unidades asociadas a este comportamiento son las unidades de operación y unidades de despliegue de contenido. El esquema en WebML de esta página está en la siguiente figura 3.26:

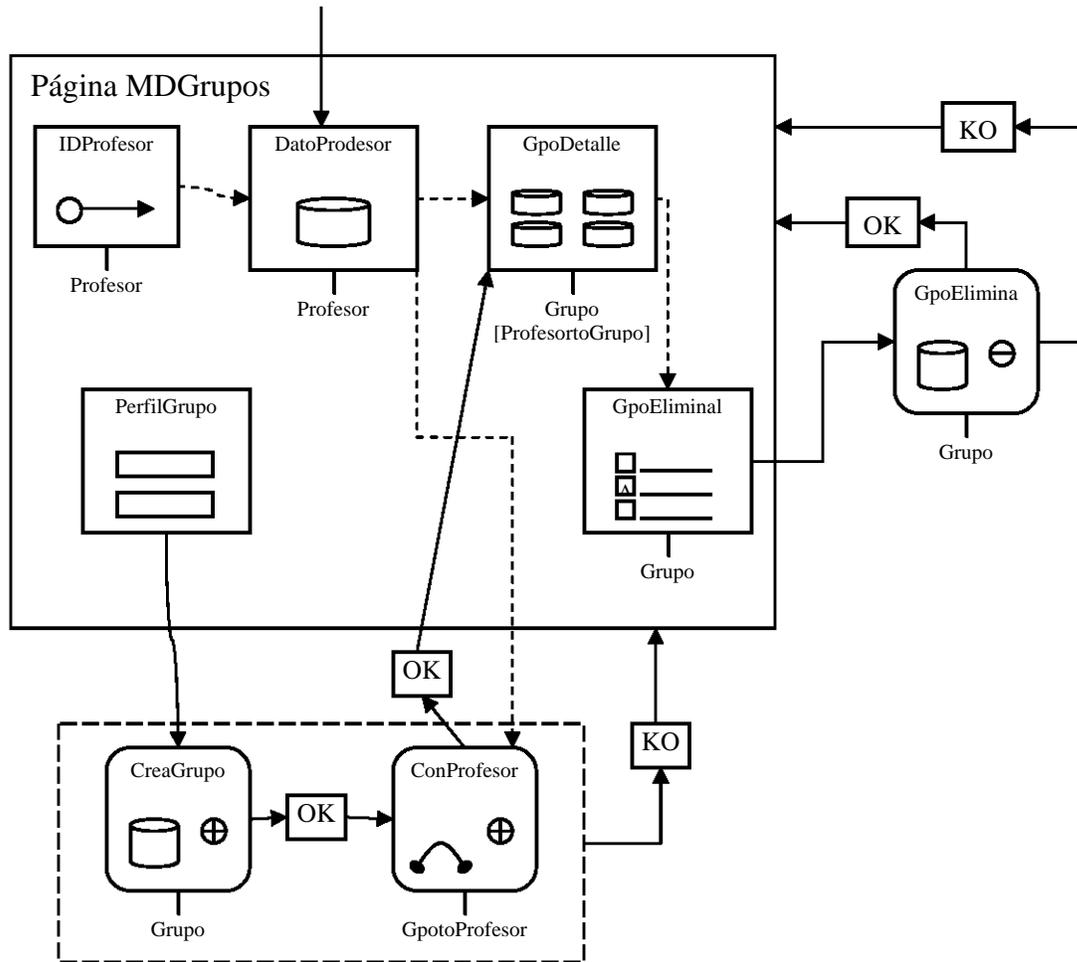


Figura 3.26 Página *MDGrupos* del área *AdmGrupos* correspondiente a la vista del sitio del *Personal Evaluador*

La página *SelGpoAlGpo* se accede primero cuando el usuario ingresa al área *AsigAlGpo*, las unidades índice permiten seleccionar el parámetro de la entidad *Grupo* requerido para mostrar la información en la página *MDAlumnos*. Los esquemas en WebML se muestran a continuación en las figuras 3.27 y 3.28.

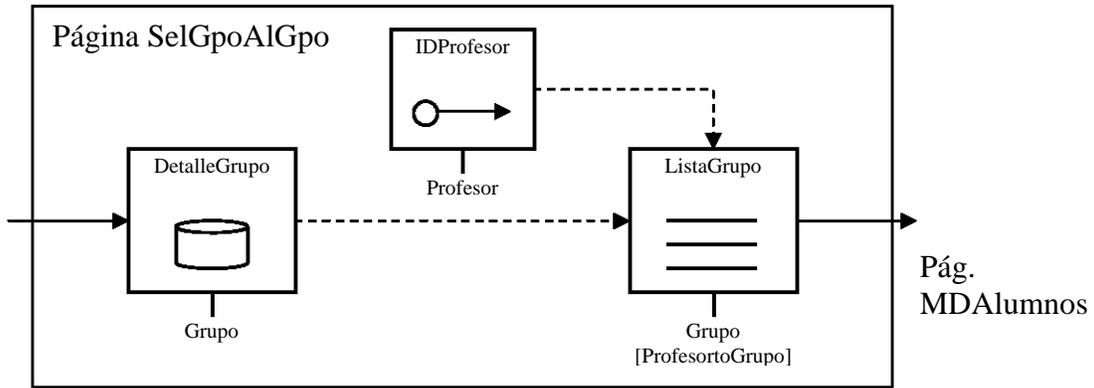


Figura 3.27 Página *SelGpoAlGpo* del área *AsigAlGpo* correspondiente a la vista del sitio del *Personal Evaluador*

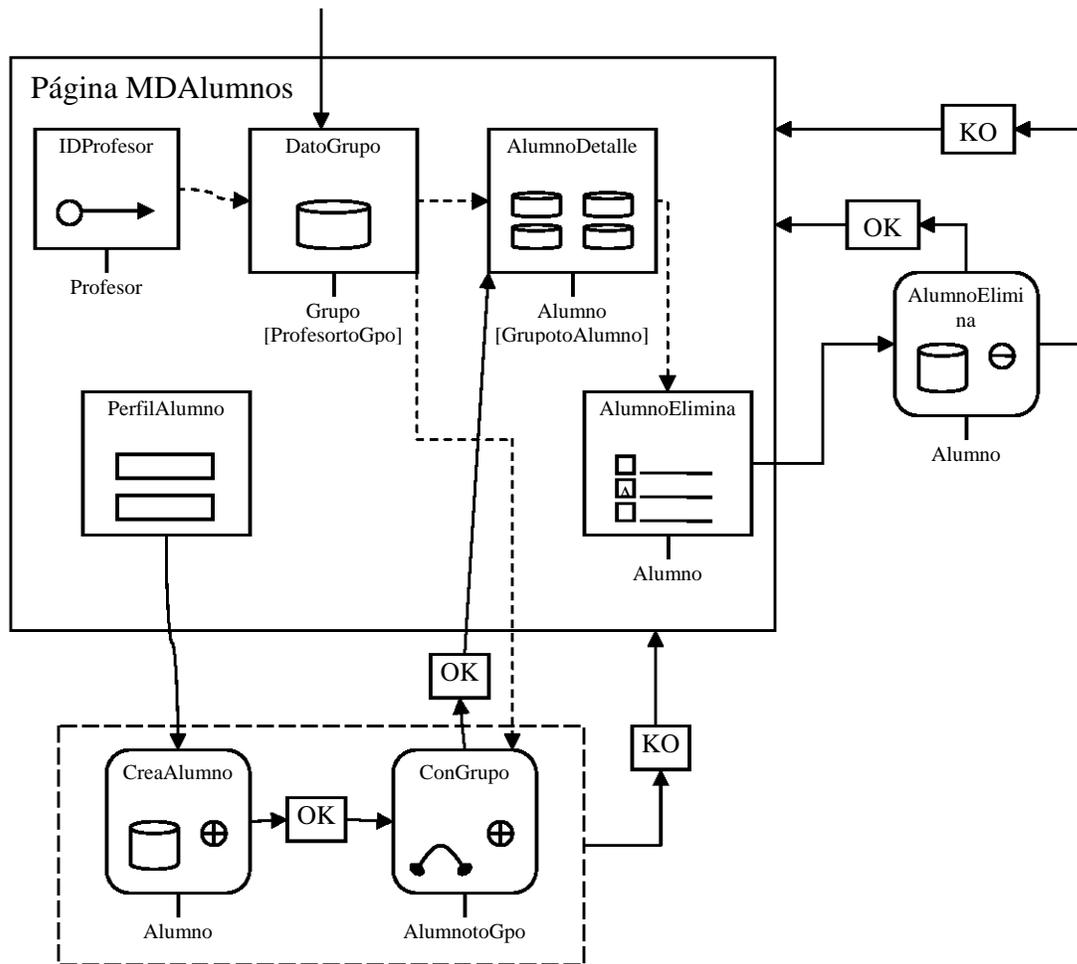


Figura 3.28 Página *MDAlumnos* del área *AsigAlGpo* correspondiente a la vista del sitio del *Personal Evaluador*

La última área de esta vista del sitio, el área *GestCalificación*, se compone de tres páginas, la página *SelGrupoCa* que permite al usuario de elegir los objetos de la entidad Grupos, cuyos parámetros son solicitados por la página *SelPruebaCa*. En esta página el usuario selecciona las instancias de la entidad *PruebaResuelta* para mostrar algunos de sus parámetros en la página *MECalificacion*. La página *MECalificacion* realiza operaciones de eliminación y despliegue de contenido pertenecientes a las calificaciones de las pruebas resueltas. La notación en WebML de estas páginas se ilustran enseguida (figura 3.29 y figura 3.30).

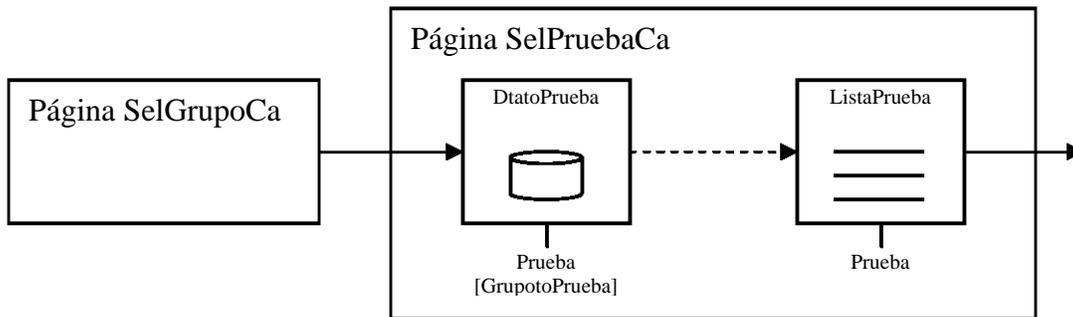


Figura 3.29 Páginas antecesoras a la página *MECalificacion* del área *GestCalificacion* correspondiente a la vista del sitio del *Personal Evaluador*

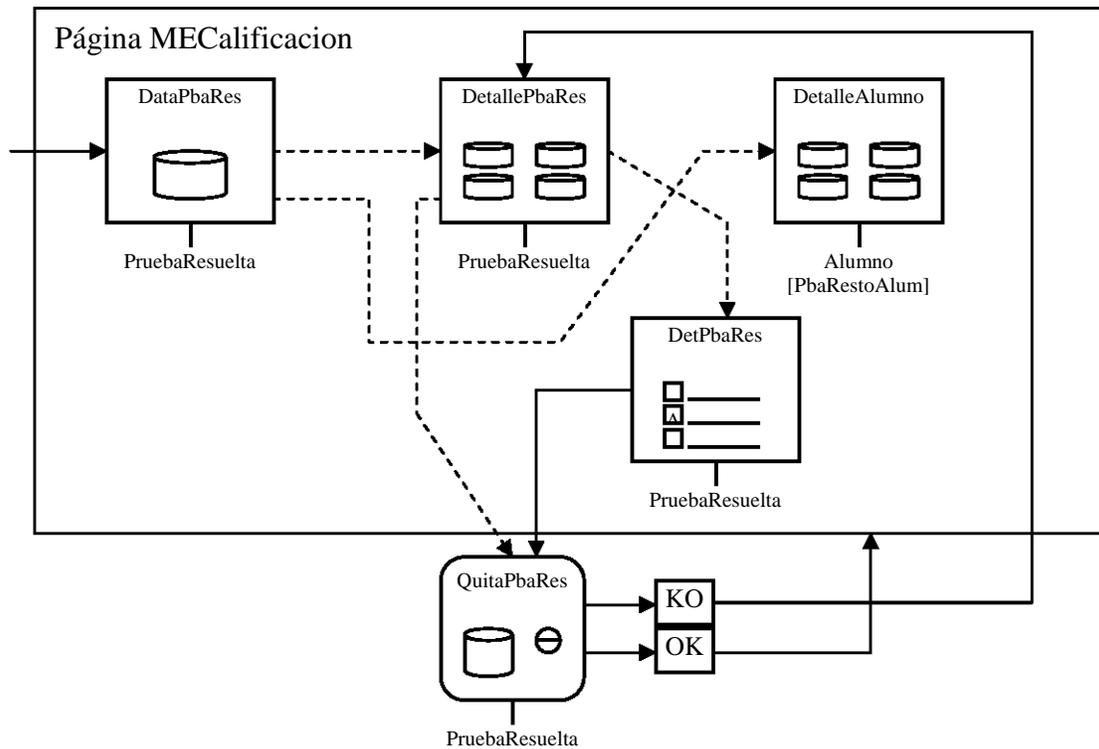


Figura 3.30 Página MECalificacion del área GestCalificacion correspondiente a la vista del sitio del Personal Evaluador

3.3.2.1.2 Vista del Sitio Personal a Evaluar

El área *InicioPE* contiene unidades de despliegue de contenido asociado a las entidades accesibles por este grupo de usuarios. La notación en WebML es similar al contenido en la página del área *inicioEva*; sin embargo, las instancias contempladas corresponden a Alumno, PruebaResuelta y Grupo.

La página *DCalificacion* correspondiente al área *ConsutaCalificacion* es similar a la propuesta en el área *GestCalificacion* de la vista del sitio *Personal Evaluador*; sin embargo, la página *DCalificacion* no dispone la posibilidad de eliminar alguna instancia de la entidad *PruebaResuelta*.

La última área, *ResuelvePrueba*, contempla dos páginas, la página *SelGrupoE* que permite seleccionar la instancia grupo requerida en la página *SelPrueba*. En esta última página las unidades de despliegue muestran el contenido de la instancia Prueba ligada al Grupo seleccionado anteriormente. La notación WebML de la página *ResuelvePrueba* se muestra enseguida, tomando en cuenta que sus páginas antecesoras tienen similitud a las vistas en otras vistas del sitio. Esta página permite a los usuarios de esta área realizar la resolución de las pruebas disponibles en su cuenta y sólo una a la vez. La figura 3.31 ilustra el conjunto de páginas encerradas en el área *ResuelvePrueba*.

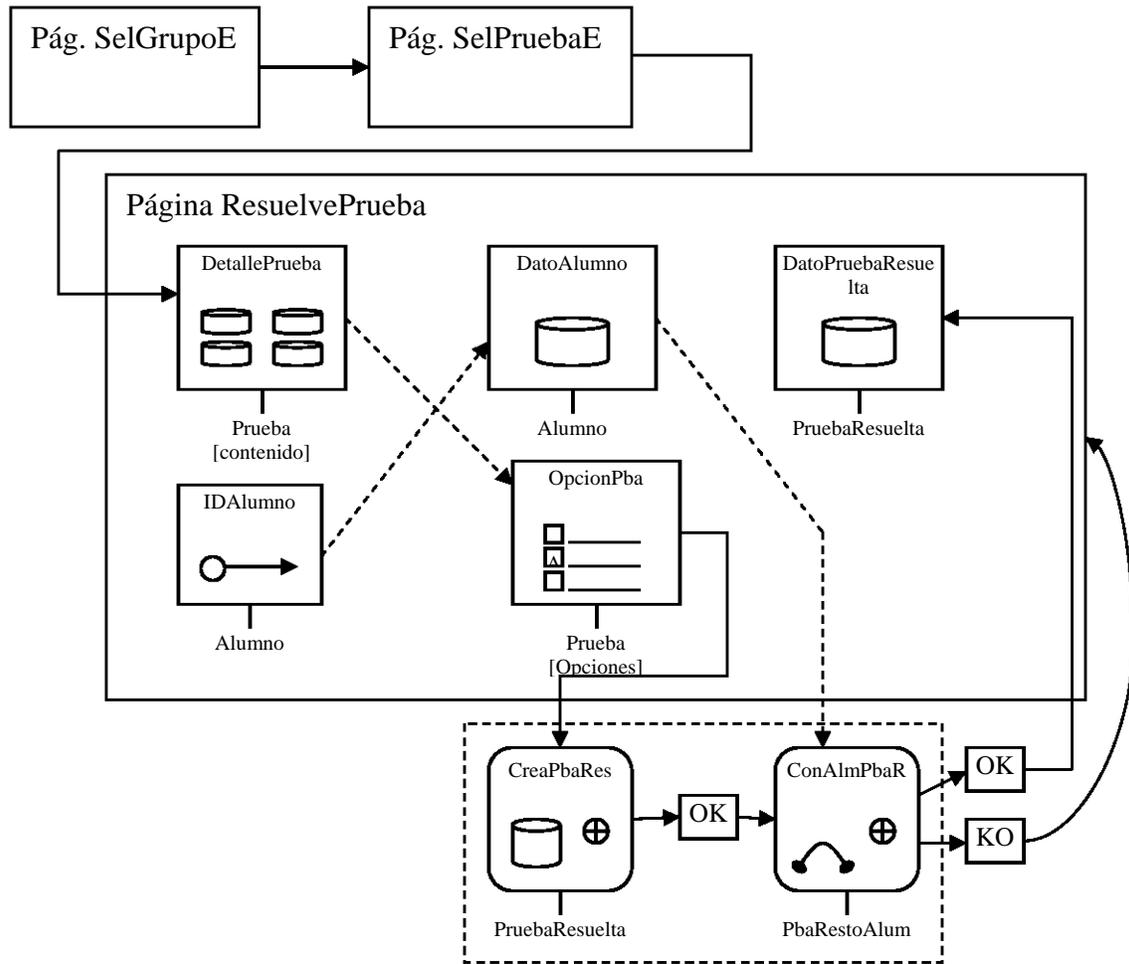


Figura 3.31 Área *ResuelvePrueba* y sus respectivas páginas. La página *ResuelvePrueba* permite al *Personal a Evaluar* realizar una prueba

3.3.2.1.3 Vista el Sitio Personal Administrador

Las dos áreas contemplan operaciones y manipulación de instancias similares a los propuestos en anteriores vistas del sitio, sólo que las entidades con que trabajan cambian. Tales entidades son el Administrador y Profesor, cuyas unidades corresponden a crear, modificar y eliminar instancias. Las respectivas notaciones en WebML para estas áreas no son necesarias describirlas, ya que son semejantes a áreas de otro grupo de usuarios.

Capítulo IV

Implementación del SIAEROM

El proceso de traducir los diagramas en notación WebML a algún lenguaje de programación es definido en esta fase de implementación. El contenido estático de las páginas del sistema Web no son contemplados en la notación WebML, es decir, la publicación de contenido no relacionado con las entidades definidas en el diagrama de datos no son descritos en notación WebML, ya que tal contenido estático no son propiamente instancias de alguna entidad. Para cuestiones de publicación de contenido estático sólo se describe su contenido en los *mapas de vistas del sitio*. En nuestro caso, el sistema Web interactúa constantemente con las entidades del esquema de datos y por tal motivo los esquemas en notación WebML representan en su totalidad al sistema Web.

4.1 Implementación de Datos

La implementación de datos trata sobre el mapeo del esquema conceptual de datos a fuentes concretas de datos, las que permiten la publicación (y en ocasiones la administración) del contenido en la Web.

4.1.1 Mapeo Estándar

Inicialmente, se deriva el esquema relacional a partir de un diagrama Entidad-Relación. Dentro de los problemas clásicos que son tratados en la mayoría de los textos de diseño de bases de datos, figuran los siguientes:

- El mapeo de entidades y sus atributos (BLOB).
- El mapeo de relaciones.
- El mapeo de jerarquías de generalización.

Las entidades son traducidas a tablas relacionales, mientras que las relaciones pueden ser traducidas en tablas “puente”, o en columnas extras, agregadas a una tabla para referenciar a otra. Las jerarquías de generalización conllevan traducciones alternativas, cuyos “pros” y “contras” pueden depender de la distribución de los datos y de las consultas esperadas.

4.1.2 Mapeo de las Entidades

Cada entidad en los diagramas entidad relación se transforma en una tabla relacional, con tantas columnas como el número de atributos en la entidad, más una columna para el OID (object identifier), el cual sirve al propósito de identificar de manera única a las instancias de la entidad, definida como la *llave primaria* de la tabla.

Si alguno de los atributos posee con anterioridad el rol de identificador de objeto en la aplicación, y tiene un dominio simple (por ejemplo, el código de un producto), éste puede ser usado como llave primaria, omitiendo la columna OID; sin embargo, las llaves primarias son usadas de manera intensiva en el mapeo de las relaciones, por lo que se sugiere evitar atributos relacionados con la aplicación y usar tipos de atributos manejados por el sistema, tales como los tipos especiales de columna llamados *serial* o *autonumber*, provistos por algunos DBMS, los cuales garantizan que el sistema crea un valor distinto de la columna para cada renglón de la tabla, liberando al código de la aplicación de la carga de producir valores distintos. Además, los valores OID nunca deben ser actualizados, debido a que ellos representan la identidad de un objeto, no una propiedad del mismo.

4.1.3 Mapeo de las Relaciones

El mapeo de las relaciones depende de su cardinalidad. El mapeo de relaciones muchos-a-muchos, requiere la introducción de una tabla dedicada, consistente de las columnas de llaves primarias de las entidades participantes en la relación. La llave primaria de la tabla puente comprende las columnas de llaves primarias de las relaciones participantes; esto refleja el significado de una relación binaria, la cual requiere que dos mismos objetos no puedan participar de la misma relación más de una vez.

Los valores que son almacenados en las columnas de la tabla puente representan referencias a las llaves primarias de los objetos participantes de la relación, Por lo que, éstos valores no se pueden elegir de manera libre, sino que son restringidos a un subconjunto de los valores de llaves primarias que figuran en las tablas de las entidades participantes. Esta restricción es un ejemplo de la restricción de *integridad en la base de datos*, esta es, una propiedad que debe cumplirse en todo estado de la base de datos, para que su contenido sea válido. Las restricciones de integridad son una característica importante de las bases de datos, y son soportadas por el lenguaje SQL. El cual incluye sentencias para expresarlas en la definición del esquema de la base de datos.

La restricción necesaria para el mapeo de relaciones puede ser expresada en SQL usando una restricción de integridad de llave foránea en la definición de la tabla puente. Una restricción de llave foránea conecta una tabla hijo con una tabla padre, imponiendo que los valores de algunas columnas de la tabla hijo, llamadas columnas referentes, concuerde con alguno de los valores de las columnas seleccionadas de la tabla padre, llamadas columnas referidas. En el caso más simple las columnas referentes constan de una columna, y la columna referida es la llave primaria de la tabla padre, lo que significa que los renglones de la tabla hijo, incluyen un apuntador a los objetos de la tabla padre.

Sintácticamente, una restricción de llave foránea es expresada en la definición de la tabla hijo. La sentencia:

```
foreign key Entity1_OID referentes Entity1_Table
```

En la declaración de la tabla hijo `Rel1_Table` especifica una restricción de integridad referencial entre la columna referente `Entity1_OID` y la columna de la llave primaria de la tabla padre `Entity1_Table`.

La base de datos relacional y el lenguaje SQL proveen mecanismos para restaurar de manera automática la base de datos a un estado válido, cuando una actualización ocurre, la cual produce una violación de integridad referencial. La integridad referencial puede ser violada por cuatro tipos de actualización, los cuales pueden producir renglones en la tabla hijo con valores erróneos de las columnas en las columnas de llaves foráneas.

- La eliminación o actualización de llaves primarias en renglones de la tabla padre.
- La inserción o actualización de llaves primarias en la tabla hijo.

En SQL, es posible expresar cuatro acciones que serán tomadas por el sistema de administración de bases de datos en respuesta a una violación de integridad referencial.

- *Cascada*: la eliminación o actualización de una llave primaria del renglón referenciado es propagado al renglón referente.
- *Nulo*: la columna de la llave foránea en el renglón referente es definido como nulo.
- *Default*: la columna de la llave foránea en el renglón referente es puesta a una valor por defecto, establecido en la definición de la columna de la llave foránea.
- *Sin Acción*: La transacción causante de la violación a la integridad es abortada. Esta opción es la opción por defecto.

Sintácticamente, la reparación a la integridad es agregada a la definición de la llave foránea en la tabla hijo; por ejemplo, la siguiente declaración SQL, especifica que la eliminación de un renglón de la tabla padre (`Entity1_Table`) es seguida por la eliminación de todas las tablas asociadas en la tabla hijo.

```
Foreign key Entity1_OID references Entity1_Table on delete cascade
```

Las relaciones uno-a-muchos son traducidas mediante la adición de una columna extra a la tabla de la entidad participante de la relación con cardinalidad máxima igual a uno.

De manera similar a las relaciones muchos-a-muchos, una restricción de llave formánea debe ser agregada a la definición de la tabla hijo, para expresar que los valores de la tabla referente son un subconjunto de los valores de la llave primaria de la tabla referenciada. Para prevenir violaciones a la integridad, una acción de reparación SQL puede ser asociada con la restricción de integridad referencial: dicha acción puede ser *HACER NULO*, si la relación es opcional (esto es, la cardinalidad mínima es igual a cero), o *CASCADA*, si la relación es obligatoria (esto es, la cardinalidad mínima es igual a uno).

Capítulo 4: Implementación

Nótese que la opción ON DELETE CASCADE causa la eliminación del registro referente en su totalidad, el cual representa la totalidad del objeto participante en la relación, y no solamente la instancia de la relación.

El caso de relaciones uno-a-uno es manejado como es caso de uno-a-muchos. La columna de la llave foránea que mapea la relación es preferentemente agregada a una tabla participante con una cardinalidad 1:1, si existe, para evitar valores nulos.

Al aplicar las reglas estándar de mapeo, cada entidad corresponderá a una tabla relacional, con tantas columnas como el número de atributos en la entidad; dado que todas las entidades con excepción de Alumno (noCuenta) y Grupo (noGrupo, claveAsignatura) carecen de una llave apropiada, agregamos a cada tabla una columna OID (object identifier) numérica y la definimos como la llave primaria. Cada relación corresponde ya sea a una tabla puente (si la cardinalidad es muchos-a-muchos) o a una llave foránea (si la cardinalidad es uno-a-muchos).

A continuación se muestran las definiciones SQL de manera completa.

```
Create table Profesor (  
    idProfesor Int NOT NULL AUTO_INCREMENT,  
    nombre Varchar(25) NOT NULL,  
    apellido Varchar(25) NOT NULL,  
    email Varchar(30),  
    nUsuario Varchar(8) NOT NULL,  
    password Varchar(20) NOT NULL,  
    UNIQUE (idProfesor),  
    Primary Key (idProfesor)) ENGINE = MyISAM;
```

```
Create table Alumno (  
    noCuenta Int UNSIGNED NOT NULL,  
    nombre Varchar(25) NOT NULL,  
    apellido Varchar(25) NOT NULL,  
    email Varchar(30),  
    nUsuario Varchar(8) NOT NULL,  
    password Char(20) NOT NULL,  
    UNIQUE (noCuenta),  
    Primary Key (noCuenta)) ENGINE = MyISAM;
```

```
Create table Administrador (  
    nombre Varchar(25) NOT NULL,  
    apellido Varchar(25) NOT NULL,  
    email Varchar(30),  
    nUsuario Varchar(8) NOT NULL,  
    password Varchar(20) NOT NULL) ENGINE = MyISAM;
```

```
Create table Grupo (  
    noGrupo Tinyint UNSIGNED NOT NULL,  
    cAsignatura Smallint UNSIGNED NOT NULL,  
    idProfesor Int UNSIGNED NOT NULL,  
    Primary Key (noGrupo,cAsignatura)) ENGINE = MyISAM;
```

```
Create table Prueba (  
    idPrueba Int NOT NULL AUTO_INCREMENT,  
    noGrupo Tinyint UNSIGNED NOT NULL,  
    cAsignatura Smallint UNSIGNED NOT NULL,  
    idProfesor Int UNSIGNED NOT NULL,  
    nPrueba Char(25) NOT NULL,  
    fCreacion Date NOT NULL,  
    fModificacion Date NOT NULL,  
    contenido Text NOT NULL,  
    UNIQUE (idPrueba),  
    Primary Key (idPrueba)) ENGINE = MyISAM;
```

```
Create table PruebaResuelta (  
    noCuenta Int UNSIGNED NOT NULL,  
    idPrueba Int UNSIGNED NOT NULL,  
    calificacion Tinyint UNSIGNED NOT NULL,  
    fResolucion Date NOT NULL,  
    Primary Key (idPrueba, noCuenta)) ENGINE = MyISAM;
```

```
Create table AlumnoGrupo (  
    noCuenta Int UNSIGNED NOT NULL,  
    noGrupo Tinyint UNSIGNED NOT NULL,  
    cAsignatura Smallint UNSIGNED NOT NULL,  
    idProfesor Int UNSIGNED NOT NULL,  
    Primary Key (noCuenta, noGrupo, cAsignatura)) ENGINE = MyISAM;
```

Capítulo 4: Implementación

A continuación se describen las secuencias SQL usadas para definir las restricciones relativas a la integridad referencial de la base de datos:

```
Alter table Grupo add Foreign Key (idProfesor) references Profesor (idProfesor)
on delete restrict on update restrict;
```

```
Alter table PruebaResuelta add Foreign Key (noCuenta) references Alumno
(noCuenta) on delete restrict on update restrict;
```

```
Alter table AlumnoGrupo add Foreign Key (noCuenta) references Alumno
(noCuenta) on delete restrict on update restrict;
```

```
Alter table Prueba add Foreign Key (noGrupo,cAsignatura) references Grupo
(noGrupo,cAsignatura) on delete restrict on update restrict;
```

```
Alter table Prueba add Foreign Key (idProfesor) references Profesor (idProfesor)
on delete restrict on update restrict;
```

```
Alter table AlumnoGrupo add Foreign Key (noGrupo,cAsignatura) references
Grupo (noGrupo,cAsignatura) on delete restrict on update restrict;
```

```
Alter table PruebaResuelta add Foreign Key (idPrueba) references Prueba
(idPrueba) on delete restrict on update restrict;
```

4.1.4 Formato de Transmisión de datos

El formato empleado en la aplicación Web para la transmisión de datos entre el cliente y el servidor es XML con el protocolo de comunicación HTTP⁶. El contenido de las Pruebas son enviadas al servidor en una estructura jerárquica en notación XML, el siguiente esquema define tal estructura.

```

<?xml vresion="1.0" encoding="UTD" stanalone="yes"?>
<contenido>
  <institucion> ... </institucion>
  <evaluador> ... </evaluador>
  <nombre> ... </nombre>
  <tiempo><hora> ... </hora><minuto> ... </minuto></tiempo>
  <instrucciones> ... </instrucciones>
  <pregunta id=1> ... </pregunta>
    <opcion id= 1 value=V> ... </opcion>
    <opcion id= 1 value=F> ... </opcion>
    <opcion id= 1 value=F> ... </opcion>
    <opcion id= 1 value=F> ... </opcion>
    ...
  <pregunta id=N> ... </pregunta>
    <opcion id= N value=F> ... </opcion>
    <opcion id= N value=F> ... </opcion>
    <opcion id= N value=V> ... </opcion>
    <opcion id= N value=F> ... </opcion>
</contenido>

```

Los nodos referentes a preguntas y opciones corresponden a los reactivos de la Prueba, mientras que los restantes nodos describen el perfil de la Prueba.

4.2 Implementación de Hipertexto

La implementación del sistema Web SIAEROM se centraliza en la página gestora de Pruebas, ya que muchas acciones en ella son comunes en otras páginas. Como se describió en el diseño de hipertexto, las acciones que suceden en esta página están aplicadas a las pruebas y su contenido, por ello la descripción del código estará organizada en dos secciones, una correspondiente a las pruebas en sí, y otra refiriéndose al contenido de cada prueba. Antes de describir con código cada acción en la página Pruebas, primero se detallará la manera de estructurar las pruebas y su contenido acorde en la arquitectura establecida al principio de este capítulo.

⁶ Protocolo de Transmisión de Hipertexto. Define un conjunto de normas de acuerdo a cómo los mensajes y otros datos deberían ser formateados e intercambiados entre servidores y browsers.

Como la aplicación Web a implementar será desplegada por los browsers, la estructura que éstos gestionan con el contenido que albergan en memoria está dada por el DOM, el *Modelo de Objetos de Documento*⁷. Para acceder, modificar o manipular los objetos del DOM y sus propiedades, Javascript dispone de métodos para realizarlo, y así manipular la estructura del documento generada en memoria por el browser. Tal documento representa la página Web, pero para nuestro caso específico, sólo una parte de éste será útil en la construcción y manipulación de las pruebas. La estructura del DOM correspondiente al documento de una prueba se puede representar por un árbol como el mostrado por la siguiente figura 4.1:

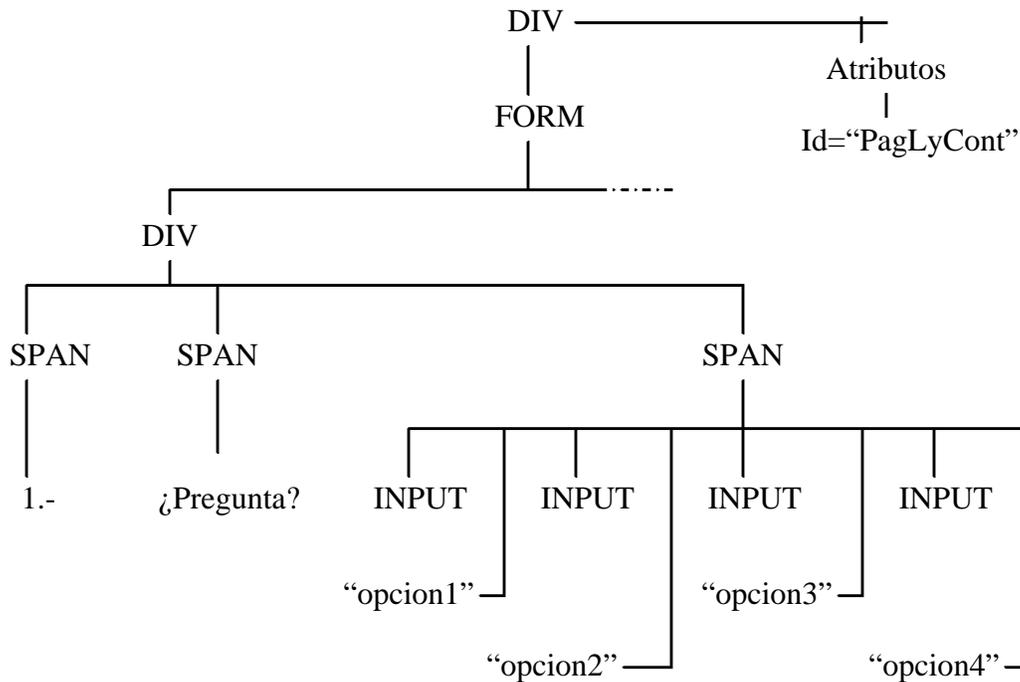


Figura 4.1 DOM de la prueba, los objetos SPAN, FORM, DIV e INPUT representan etiquetas en el lenguaje HTML. Algunos nodos se omitieron por cuestiones de legibilidad

El DOM de la Prueba es desplegado en el panel de contenido. La jerarquía de este árbol tiene como raíz a un elemento DIV identificado durante todo el sistema como *PagLyCont*, que es fijo durante la gestión de pruebas, en cambio el nodo FORM alberga a los nodos correspondientes a más de una pregunta con sus cuatro opciones, y este nodo sufrirá alteraciones cuando se creen o eliminen pruebas. Ahora, cada opción de un reactivo además de texto es acompañada de un objeto que selecciona la opción deseada por el

⁷ Es una jerarquía de objetos predefinidos que constituyen a una página web. Tales objetos poseen propiedades y métodos, estos últimos permiten acceder y modificar los valores predefinidos.

usuario (en este caso por un *radiobutton*⁸), el cual almacena el valor asociado a la opción correcta de la respectiva pregunta. Y para el caso de agregar o eliminar reactivos, los nodos de interés son los hijos del nodo FORM, los nodos DIV.

Ahora, es momento de pasar a la descripción general del código empleado en la implementación de la página gestora de Pruebas.

4.2.1 Gestor de Pruebas

En cuanto a la gestión de las pruebas, se disponen de acciones como crear, eliminar, abrir y guardar pruebas. Recordando que, cada prueba es una instancia de la Entidad Prueba correspondiente a la tabla de la base de datos Prueba. Así que, el nombre de la prueba como su contenido son almacenados en los registros respectivos de *nombre* y *contenido*, además que el identificador de cada instancia está reservado en el campo *idPrueba*.

Al crear o abrir alguna Prueba en la página gestora de Pruebas implica crear un nuevo nodo FORM, no antes verificar el estado anterior del espacio de trabajo, ya que pudiese haber una Prueba previamente creada o abierta. Para lo cual se procede a buscar algún nodo hijo existente en el nodo raíz DIV del documento de la Prueba, que posiblemente este nodo, puede ser creado o reemplazado por otro nodo según sea el caso y la elección del usuario. El siguiente código muestra la validación del nodo raíz DIV y la incursión del nodo FORM en caso que el usuario lo desee.

⁸ Objeto de interfaz de usuario usado en la construcción de páginas Web que permite al usuario seleccionar alguna opción. Se construye con la etiqueta INPUT aplicando el valor del atributo `type='radio'`.

```
//Verifica la existencia de algún nodo hijo en el nodo DIV
//identificado como PagLyCont
if (document.getElementById('PagLyCont').hasChildNodes())
{
    if (confirm('Actualmente tienes un trabajo \nDeseas
                borrarlo y crear un nuevo cuestionario?'))
    {
        var numnodos=nodoForm.childNodes.length;
        //Elimina nodos hijos existentes en el nodo DIV
        //identificado como PagLyCont
        while(document.getElementById('PagLyCont').hasChildNodes())
        {
            document.getElementById('PagLyCont').removeChild(
                document.getElementById('PagLyCont').firstChild);
        }
        //Crea el objeto FORM de formulario que contendrá los reactivos
        nodoForm=document.createElement('form');
        var PagLyCont = document.getElementById('PagLyCont')
        PagLyCont.appendChild(nodoForm);
    }
}
else //En caso que no exista algún nodo hijo, crea el nodo FORM
{
    //nodoForm es el nodo del elemento FORM
    nodoForm=document.createElement('form');
    var PagLyCont = document.getElementById('PagLyCont');
    PagLyCont.appendChild(nodoForm);
}
```

El caso específico de crear una Prueba, sólo es necesario tener el nodo FORM libre, ya posteriormente el usuario insertará los reactivos en otro módulo. La creación de tal Prueba ocurre cuando es creado su perfil en la subpágina *nuevaPba* del panel contextual.

El caso particular de abrir una Prueba es distinto al de crearla, ya que tal instancia es seleccionada de una lista colocada en el panel de navegación (o sub-página navegación referida en notación WebML). Esta lista despliega las posibles Pruebas disponibles en la sesión del usuario. Al momento de presentarse el evento doble-click sobre algún elemento de la lista de Pruebas, el manejador de eventos de Javascript lanza la función que captura el identificador de la Prueba *idPrueba* a ser enviado como parámetro al servidor que retornará el contenido de dicha instancia (Prueba). Dicho identificador se captura en la lista de Pruebas *lstPrueba* situada en el panel de navegación (o subpágina navegación en notación WebML) para posteriormente reenviarse en caso de actualizar la Prueba.

Eliminar una Prueba implica eliminar la instancia de la entidad Prueba, para ello se recurre al identificador de la prueba *idPrueba* en cuestión, que es tomado al momento en que el usuario selecciona alguna Prueba enlistada en la subpágina *eliminaPba* situada en el panel contextual. Tal identificador es utilizado para ejecutar alguna sentencia en SQL que elimina el registro en la base de datos. El siguiente código ilustra la captura del identificador de la Prueba contenido en el atribuido *value* de la lista *lstPrueba* al momento que una opción es seleccionada por parte del usuario, para posteriormente asignarla al parámetro *idPrueba* y enviarla al servidor.

```
//Confirma la eliminación del nodo seleccionado en la lista
//lstPruebaEli
if(confirm('Deseas eliminar la prueba: '+
    document.forms['formSelectPba2'].lstPruebaEli.options[
    document.forms['formSelectPba2'].lstPruebaEli.selectedIndex
    ].text))
{
var selectnodo =
    document.forms['formSelectPba2'].lstPruebaEli.options[
    posicion].value;
var paramsDeletePrueba = 'idPrueba=' + selectnodo;
//Envía parámetros al servidor de manera asíncrona
xmlHttp.open("GET", serverAddress + "?" + paramsDeletePrueba +
    "&action="+option, true);
xmlHttp.onreadystatechange = handleDeletePrueba;
xmlHttp.send(null);
}
```

El proceso de guardar una instancia Prueba ocurre cuando el contenido de cada prueba es procesado de tal forma que se pasa de un DOM que reconoce el browser a un DOM con formato XML, listo a ser almacenado en el registro *contenido* de la base de datos. Los parámetros enviados al servidor proceden de dos maneras: cuando la Prueba se actualiza, el identificador *idPrueba* y el contenido son requeridos; o sólo el contenido cuando la instancia se crea, ya que el servidor genera el *idPrueba*. El siguiente código ilustra a grandes rasgos el proceso de guardado.

```
//Función que transforma la incompatibilidad del DOM
function fguardarC()
{
  var salidaP='';
  //Nodo que sujeta el contenido de la Prueba.
  var nodoRaiz =
    document.getElementById('PagLyCont').getElementsByTagName('div
  ');

  ...

  for (var i=0; i<nodoRaiz.length; i++)
  {
    //Procesa la salida del contenido para el servidor.
    salidaP += '<pregunta id="' + encodeURIComponent(i + 1) + '>'
      + encodeURIComponent(nodoRaiz[i].getElementsByTagName(
        'span')[1].childNodes[0].data) + '</pregunta>';
    for (var j=0; j<=3 ;j++)
    {
      salidaP += '<opcion id="' + encodeURIComponent(i + 1) + '"
        value="' +
        encodeURIComponent(nodoRaiz[i].getElementsByTagName(
          'span')[2].childNodes[j*3].attributes['value'].nodeValue
        ) + '>' + encodeURIComponent(nodoRaiz[i].
          getElementsByTagName('span')[2].childNodes[j*3+1].data)
        + '</opcion>';
    }
  }
  return salidaP;
}
```

4.2.2 Gestor del Contenido de Prueba

Las acciones correspondientes en la gestión del contenido de una Prueba consisten en insertar, editar y eliminar reactivos, sin antes, disponer de una Prueba en el espacio de trabajo, ya sea haber creado o abierto alguna Prueba. El proceso de gestión de contenido implica modificar la entidad Prueba.

La inserción de algún reactivo en la jerarquía del DOM deriva crear un nuevo nodo DIV ligado al reactivo en cuestión, cada reactivo posee sus propios elementos que lo caracterizan como el número del reactivo, su nombre, y sus cuatro opciones con sus respectivos botones *radio*. Cada reactivo representa una parte del árbol del DOM que será añadido uno tras otro hasta completar el conjunto deseado por el usuario. El esquema en notación WebML describe este proceso como la modificación de la entidad Prueba y no una nueva inserción ya que la entidad existe y es utilizada actualmente, así que un nuevo reactivo se crea llenando su perfil en el campo de entrada de datos. El siguiente código describe las acciones de inserción de un reactivo al árbol DOM de la Prueba.

```

//Inserta nuevo nodo reactivo al nodo padre (Prueba)
function finsertarR(f)
{
  ...
  var divnodo=document.createElement('div');
  //modoForm representa el elemento form de la jerarquía del DOM
  nodoForm.appendChild(divnodo);
  //Crea nodo para número de preguntas
  var spannum=document.createElement('span');
  var numerorea=numnodos + '.- ';
  spannum.appendChild(document.createTextNode(numerorea));
  //Crea nodo pregunta
  var spanpregunta=document.createElement('span');
  spanpregunta.appendChild(document.createTextNode(
    document.forms['formInsertarR'].ReacLyIR.value ));
  spanpregunta.appendChild(document.createElement('br'));
  //Crea las opciones, tanto texto como botón radio
  var spanopciones=document.createElement('span');
  var opcion1 = paintRadios(tradio + numnodos);
  opRadioInRe = opcion1.value =
    document.forms['formInsertarR'].CorrLyIR[0].checked;
  opcion1.defaultChecked = opRadioInRe;
  opcion1.disabled="disabled";
  spanopciones.appendChild(opcion1);
  spanopciones.appendChild(document.createTextNode(
    document.forms['formInsertarR'].Opc1LyIR.value ));
  spanopciones.appendChild(document.createElement('br'));
  ...
  //Agrega no elementos anteriormente creados al nodo DIV
  //disponible en cada reactivo
  divnodo.appendChild(spannum);
  divnodo.appendChild(spanpregunta);
  divnodo.appendChild(spanopciones);
  ...
}

```

El caso eliminación de un reactivo, sólo requiere la posición del elemento nodo en el árbol DOM correspondiente al elemento padre, el nodo FORM. Según el esquema en notación WebML descrito en la fase de diseño, el usuario debe seleccionar un elemento asociado al contenido de la entidad Prueba que procederá a modificar dicha entidad, concretamente en el campo *contenido*. El código siguiente describe de manera general el proceso de eliminado del reactivo, donde la variable *posición* almacena el valor elegido por el usuario de la lista *NumLyER*, este valor representa el nodo hijo a eliminar del nodo padre FROM.

```
//Elimina nodo reactivo del nodo padre (Prueba)
function feliminarR()
{
    var posición =
document.forms['formEliminarR'].NumLyER.selectedIndex;
    ...

    document.getElementById('PagLyCont').getElementsByName(
'form')[0].removeChild(document.getElementById('PagLyCont').
getElementsByName('div')[posición-1]);
    ...
}
```

Editar un reactivo sólo procede en cambiar los valores asignados anteriormente a los elementos nodo del nodo reactivo en cuestión, para lograr esta tarea, el usuario inicialmente elige un reactivo del DOM de la Prueba a partir de una lista, que posteriormente mostrará los valores de sus elementos en el Panel contextual, finalizando con las modificaciones deseadas a cada elemento. La notación en WebML detalla el procedimiento descrito, y como en casos anteriores involucra la modificación de la entidad Prueba. Las siguientes líneas de código muestran el proceso de edición de un reactivo.

```
//Edita nodo reactivo elegido de la lista ElecLyER
function feditarR()
{
    ...
    var eleccionr =
document.forms['formEditarR'].ElecLyER.selectedIndex;
    //Muestra la pregunta del cuestionario en el campo reactivo
document.getElementById('PagLyCont').getElementsByName('div')
[eleccionr1].getElementsByName('span')[1].childNodes[0].
data = document.forms['formEditarR'].ReacLyER.value;

    //Muestra las opciones de pregunta del cuestionario en los campos
//opciones correspondientes
document.getElementById('PagLyCont').getElementsByName('div')
[eleccionr1].getElementsByName('span')[2].childNodes[1].
data = document.forms['formEditarR'].OpclLyER.value;
    ...
}
```

Capítulo V

Pruebas del Sistema SIAEROM

Prueba del sistema es la actividad de verificar la concordancia de la aplicación implementada, a los requerimientos funcionales y no funcionales. Los más relevantes aspectos en la prueba y evaluación de aplicaciones Web son los siguientes:

- Pruebas funcionales: El comportamiento de la aplicación es verificado con respecto a los requerimientos funcionales. La prueba funcional debe descomponerse en actividades clásicas como pruebas de unidad (o de módulo), pruebas de integración, y pruebas de sistema.
- Pruebas de usabilidad: Los requerimientos no funcionales de caso de uso, la eficacia de comunicación, y el apego a estándares formales deben verificarse contra las vistas del sitio.
- Pruebas de desempeño: La exhibición y tiempo de respuesta de la aplicación Web debe evaluarse en condiciones de carga de trabajo promedio y máximas. En caso de un inadecuado nivel de servicio, la arquitectura presentada debe monitorearse y analizarse para identificar y remover cuellos de botella.

Las pruebas no son continuación una de la otra, cada prueba posee su dominio trabajo, en el caso de pruebas funcionales éstas son realizadas mientras se diseña y construye el sistema, realimentándose con los requerimientos establecidos para el sistema. Mientras que las pruebas de desempeño se aplican cuando el sistema está en preoperación. A continuación se describen con más detalle las pruebas anteriormente enlistadas, tomando como centro de prueba la aplicación Web actual.

5.1 Pruebas Funcionales

La prueba comienza a nivel módulo y se trabaja hacia la integración del sistema. La prueba se centra en cada módulo individual, que asegure a cada módulo el correcto funcionamiento como unidad. Posteriormente, se integran los módulos para formar el sistema completo, que verifique el paso de parámetros entre unidades de manera correcta, y que este proceso no altere la funcionalidad que anteriormente se había logrado como unidad individual. Finalmente, se comprueban los criterios de validación, que proporcionen los elementos que satisfagan todos los requisitos funcionales y de comportamiento esperadas por los usuarios, asegurando el funcionamiento total del sistema. Los tres pasos anteriormente se ilustran en el siguiente diagrama de la Figura 5.1, tomando a la aplicación real como objeto de estudio.

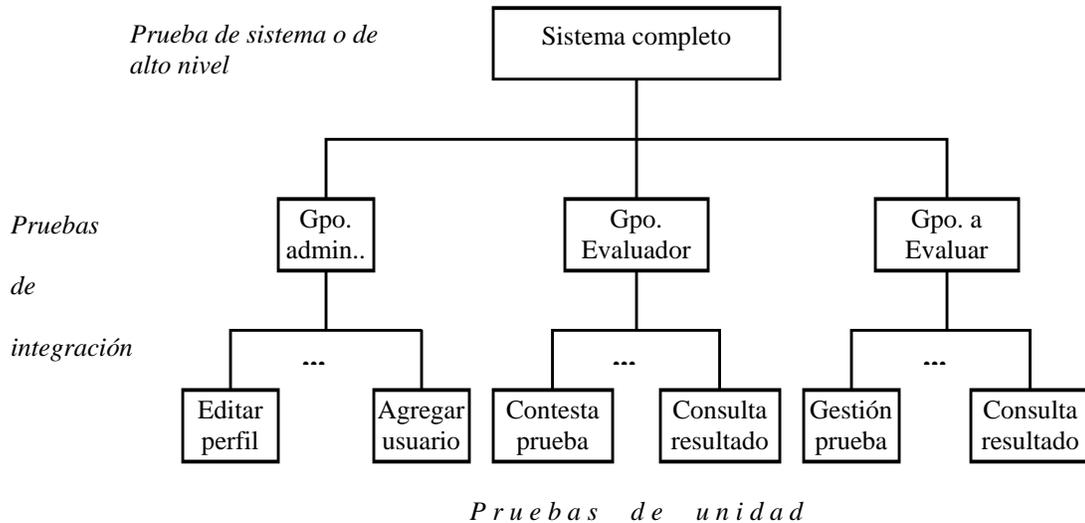


Figura 5.1 Organización de la aplicación Web real propuesta

En este diagrama las unidades corresponden a los casos de uso de cada grupo de usuario, estos casos de uso asocian actividades dedicadas en la aplicación por parte de los usuarios. Para la documentación de las pruebas funcionales, se tomarán casos representativos de los grupos más recurrentes en el sistema, el grupo Personal Evaluador y del grupo Personal a Evaluar.

5.1.1 Prueba de Unidad

La prueba de unidad se aplica a la menor unidad de diseño del sistema, el módulo, se prueban los procesos definidos en los casos de uso, con la finalidad de descubrir errores dentro del mismo. Esta tarea limita el rango de errores presentes en toda la aplicación, ya que es más fácil detectar y corregir fallas en ciertas partes de la aplicación que todo el sistema en conjunto.

Tomando como módulo de prueba al gestor de pruebas perteneciente al grupo de usuarios Personal a Evaluar, cuyo propósito es crear, guardar, eliminar y editar pruebas. Las entradas para este módulo son: noGrupo, cAsignatura, nombre, apellido, id, que se han fijado localmente para la verificación funcional del módulo. Y las salidas de este módulo dependen de los casos de uso efectuados por el usuario.

Cada caso de uso, procesa una tarea distinta que conlleva a tomar parámetros de entrada que le convenga, efectúa sus algoritmos y emite parámetros que modifican las instancias de las entidades del esquema de datos correspondiente, así que, el proceso interno de cada caso de uso produce sus propios parámetros de entrada/salida y efectos. Esta forma de trabajo conduce a dividir cada módulo en submódulos. Así, el submódulo *Abrir Prueba* corresponde al caso de uso *Abrir Prueba*.

5.1.2 Prueba de Integración

La prueba de integración toma los módulos probados en unidad para construir la estructura del sistema completo libre de fallas. La prueba pretende agrupar cada módulo con su subordinado hasta integrar el sistema. La secuencia de integración no puede dejarse a medias, ya que existen módulos dependientes de otros más alejados que podrían ocasionar detecciones difíciles de errores o defectuosas funcionalidades no contempladas en los casos de uso.

Ahora, el funcionamiento individual de los módulos se da con libre de errores, lo siguiente es fusionar dichos módulos para estructurar el sistema. Anteriormente, en la prueba de unidades se fijaron las entradas a cada módulo como locales, en la integración dichas variables son obtenidas por los módulos a través de transmisión de valores entre módulos. La prueba asegura que el paso de parámetros entre módulos se dé sin que deteriore el funcionamiento logrado en la prueba de unidades. Aunque, en ocasiones la integración sólo procede en juntar los módulos sin que exista paso de parámetros.

Un caso propuesto de prueba, es la integración del módulo *Alta-Baja de Grupos* al módulo *Menú del Evaluador*. Esta situación requiere que el módulo *Menú del Evaluador* pase el identificador *id* del Evaluador al módulo subordinado. La siguiente tabla muestra los efectos.

Grupo: *Personal Evaluador*
Módulos involucrados: *Menú del Evaluador* vs. *Alta-Baja de Grupos*.

Efectos	Observaciones
Del menú del Evaluador existe un enlace que al accionarlo direcciona al módulo de Grupos.	El comportamiento individual mostrado en la prueba de unidades no cambia ahora que se han integrado los módulos involucrados.

Las observaciones se realizan en cada integración hasta completar el sistema. Si la integración está libre de errores, se procede con la siguiente prueba.

5.1.3 Prueba de Sistema

La prueba de sistema monitorea los posibles errores presentes en la aplicación montada en un ambiente real; donde todo el conjunto de casos de uso, asociado a cada grupo de usuarios, presente los resultados esperados. Para ello, los usuarios ingresan a la aplicación, manipulan y exponen los casos de uso disponibles en su sesión. Al final se realiza una valoración de conformidad que asegure la satisfacción de los requisitos funcionales.

5.2 Pruebas de Uso

Las pruebas de uso evalúan todos los otros requerimientos que son sean considerados en los requerimientos funcionales, relevantes para culminar las metas del

negocio, para este caso, la aplicación Web. Así mismo, el apego en estándares que respalden la correcta codificación del sistema.

5.3 Pruebas de Desempeño

A continuación se describen los aspectos que involucra el realizar pruebas de desempeño, tales pruebas, exponen el sistema a ambientes reales en puesta en marcha.

5.3.1 Estimando la Carga de Trabajo

El punto de inicio para dimensionar la arquitectura y verificar su desempeño es la especificación de requerimientos de desempeño. Tales requerimientos cualifican el número y tipo de página peticionada por el cliente, y el tiempo necesario para responder esas peticiones de manera satisfactoria. En muchos casos, la carga de trabajo debe ser caracterizado por los siguientes parámetros:

- **Número de páginas peticionadas:** Indica el promedio y máximo número de peticiones emitidas por los clientes, expresada en páginas por segundo.
- **Número de usuarios actuales:** Expresa una predicción del promedio y máximo número de usuarios que accederán simultáneamente a la aplicación. Este número afecta a la aplicación back-end debido a la larga cola de datos en espera a ser procesados.
- **Tiempo de respuesta:** Indica el máximo número de segundos que el cliente debería esperar por la respuesta. Lo cual, no debería exceder de 3 segundos según el tiempo de respuesta mental del usuario. Este valor ignora los problemas de transmisión en la red.
- **Mezcla de peticiones:** Es el peso expresado en probabilidad, que se le asigna a cada página debido al comportamiento del usuario, ya que una aplicación está hecha de distintas páginas de distinta complejidad e interés del usuario.

5.3.2 Estableciendo el Medio de la Prueba

La prueba de desempeño debe realizarse en la infraestructura de hardware, software y de red reales utilizando la aplicación real. Dado que el tiempo de desarrollo de aplicaciones en ocasiones es duradero y costoso no es posible retrasar la verificación de la arquitectura hasta el final del proceso de desarrollo. Para anticiparse al análisis de funcionalidad, debe ser posible la construcción de un prototipo de aplicación y tenerla disponible durante la fase de diseño de arquitectura. Tal prototipo debe implementar únicamente un subconjunto de páginas correspondientes a los casos de uso más críticos y debe reproducir, lo más posible, los accesos a datos y consultas de actualización de la aplicación real.

Considerando los puntos anteriores y las posibles páginas de mayor recurrencia, se establecen los siguientes criterios para tomar en cuenta las páginas más representativas por parte de los usuarios de la aplicación Web, así como el ambiente de prueba:

- Del lado del grupo del Personal Evaluador, las peticiones de mayor probabilidad de presencia recaen en las páginas que gestionen las pruebas y calificaciones de los alumnos. Tomando la nomenclatura realizado en el diseño de Hipertexto, las páginas con mayor peso son: AdmPrueba proveniente del área GestPrueba, y la página MECalificacion perteneciente al área GestCalificacion.
- Del lado del grupo del Personal a Evaluar, las recurrentes peticiones recaen en las páginas que son ligadas a la resolución de las pruebas y al despliegue de resultados (o calificaciones del alumno). Tales páginas son: ResuelvePrueba encerrada por el área ResuelvePrueba, y la página DCalificacion perteneciente al área ConsultaCalificacion.
- Del lado del grupo Personal Administrador, no existe peso en alguna de sus páginas debido a que es un solo usuario en este grupo.
- La arquitectura de prueba real prosee un solo servidor único que incluye los cuatro componentes: Servidor Web, la máquina de ejecución de scripts, el servidor de aplicaciones y el servidor de bases de datos.
- El escenario de implantación de la arquitectura corresponde a un hospedaje o hosting, por lo cual, los clientes acceden a la aplicación Web vía Internet.

Las características y páginas tomadas en cuenta para las pruebas no forman parte de un prototipo, el ambiente de trabajo está montado sobre la aplicación real bajo cargas de trabajo simuladas.

5.3.3 Verificando el Desempeño

Una vez que la configuración de la arquitectura está montada en el lugar ideado, se procede a probar dicho ambiente con estrés de trabajo (peticiones) para determinar la aceptable y factible funcionalidad de los componentes del sistema. Probando una aplicación Web requiere simular el comportamiento de los clientes, generando un número razonable de peticiones hacia las páginas de la aplicación basándose en la recurrencia de los clientes. Esta prueba es soportada por herramientas de prueba especializadas, que ofrecen las siguientes funciones:

- Definen un conjunto de peticiones de páginas que simulan una sesión en uso. Para cada sesión en uso es posible especificar un conjunto de páginas a llamar, y el porcentaje de peticiones que deberían solicitarse a cada página en el conjunto.
- Definen un número máximo de clientes que envían peticiones a la aplicación y expone el comportamiento de cada cliente.
- Definen un ambiente de programación y ejecución para la sesión de prueba.
- Recolectan reportes acerca de varios parámetros de funcionalidad, como el número de páginas por segundo, la duración total de la sesión de prueba, el número de errores tanto de comunicación como del servidor.

Configuración del Servidor de Pruebas.

Procesador: 2.6GHz/800Mhz/4mb Cache, Dual-Core Intel Xeon
 RAM: 2GB DDR2 400MHZ
 Sistema Operativo: Windows Server 2003 R2, Standard Edition

a) Número de páginas solicitadas al servidor.

Para esta prueba se integraron dos grupos, uno correspondiente al grupo del personal evaluador y otro al grupo de personal a evaluar, cada uno consistente de 20 miembros. Para el grupo del personal evaluador los resultados fueron los siguientes:

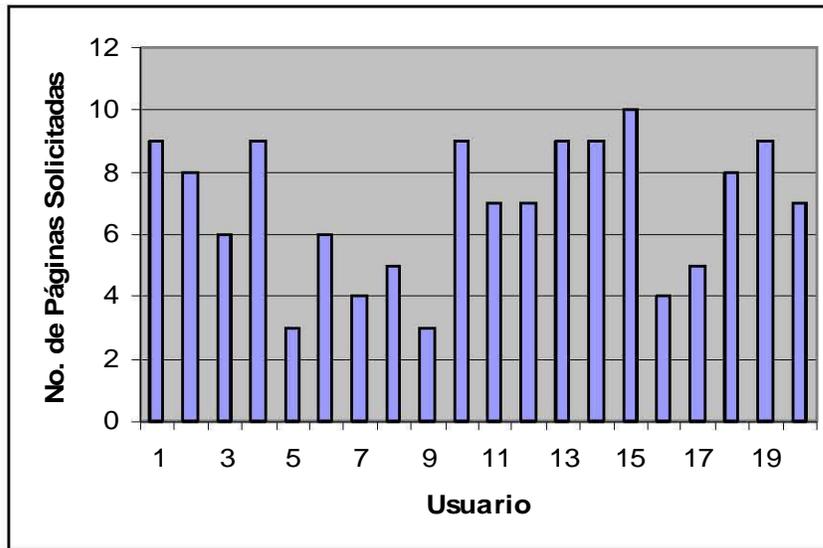


Figura 5.2 Usuario VS No. de Páginas Solicitadas

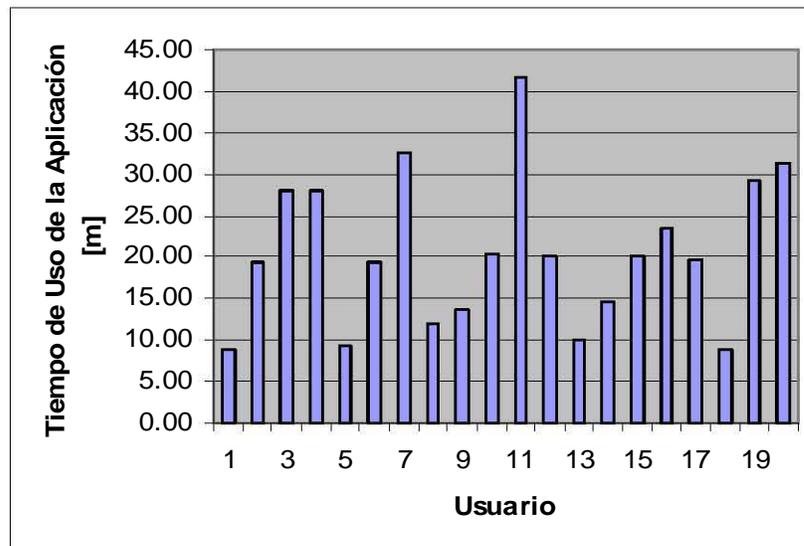


Figura 5.3 Usuario VS Tiempo Uso de la Aplicación [m]

- b) Número de Usuarios Actuales: 20 Usuarios.
- c) Tiempo Promedio de Respuesta.

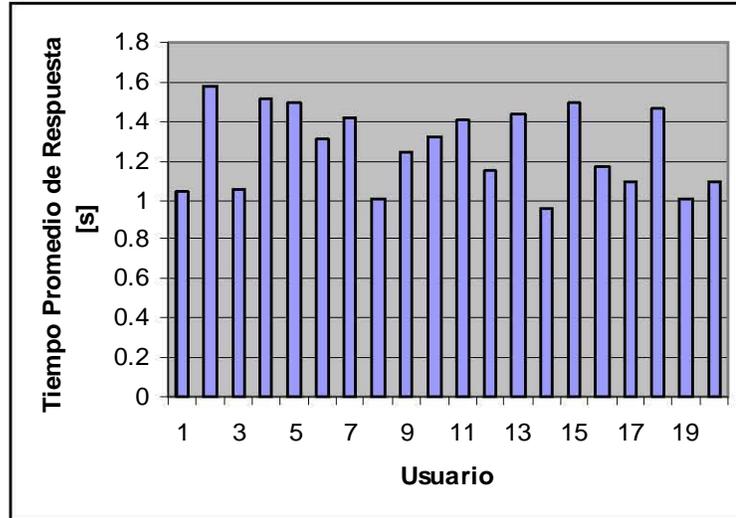


Figura 5.4 Usuario VS Tiempo Promedio de Respuesta [s]
Personal a Evaluar.

- a) Número de Páginas Solicitadas al Servidor.

Ya que únicamente se consideró en las pruebas, el caso en el que el alumno ingresa al sistema y contesta una prueba, el número de páginas solicitadas fue de cinco.

- b) Número de Usuarios Actuales: 20 Usuarios.
- c) Tiempo Promedio de Respuesta.

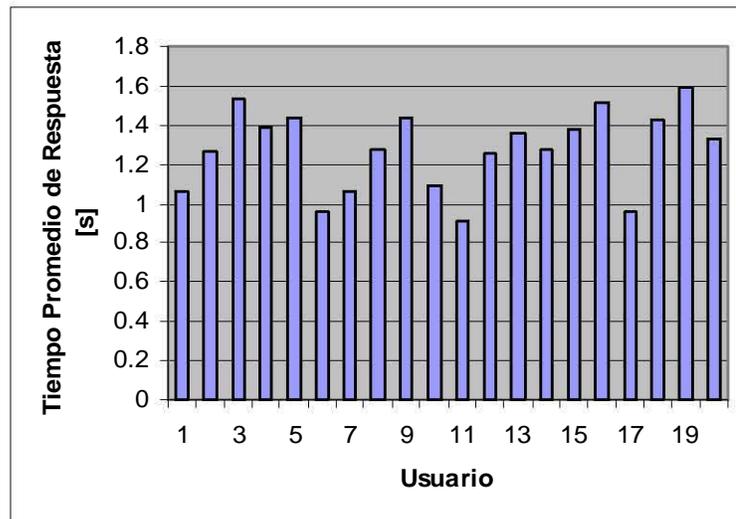


Figura 5.5 Usuario VS Tiempo Promedio de Respuesta [s]

5.4 Aspectos no Funcionales

Los aspectos no funcionales contemplan los siguientes puntos:

- *Usabilidad*: involucra la facilidad de uso de la aplicación para con el usuario en cuestión. La facilidad de uso engloba casos como el aprendizaje de las interfaces de usuario, la adherencia de la interacción de objetos como menús, enlaces o botones, y la disponibilidad de mecanismos para la orientación y asistencia al usuario en casos de desorientación.
- *Desempeño*: refiere a la eficacia con que la aplicación explota los recursos disponibles. Este punto es abordado en la sección de pruebas de desempeño.
- *Disponibilidad*: refiere a la tolerancia frecuente de errores y fallas, los cuales afectan el porcentaje en la que la aplicación está disponible a los usuarios
- *Escalabilidad*: es la habilidad para incrementar la ejecución de la aplicación en respuesta al incremento del volumen de peticiones.
- *Seguridad*: incluye aspectos como la protección de la integridad, confidencialidad y privacidad de la información.

Resumiendo los aspectos anteriormente citados y trasladados al caso real de la aplicación Web. Tal aplicación describe una interfaz gráfica amigable e intuitiva, y un manual de usuario que asiste a los diversos grupos de usuarios en las tareas previstas de funcionamiento. En cuanto a la seguridad, las diferentes zonas permitidas del sistema a los grupos de usuarios están restringidas por la cuenta que poseen, es intransferible y no visible por otros usuarios, esto es, el superusuario *Administrador* no puede visualizar el password (contraseña) de los usuarios *Alumno* ni mucho menos de los usuarios *Profesor*. Y en las cuestiones de disponibilidad y escalabilidad, dependen del servidor destino y del número de clientes accesibles a la aplicación.

5.5 Estándares de la Web

El desarrollo de la aplicación está sujeto a los estándares de la W3C⁹, los cuales exponen reglas para la manipulación de código para la Web que asegure el correcto funcionamiento del sistema en los browsers de los clientes. Las versiones mínimas aceptables que permitan una adecuada puesta en marcha de la aplicación son:

- Para las Hojas de Estilo o CSS, la especificación CSS 2.1
- Para XML, versión 1.0
- HTML 4.01
- DOM (Document Object Model) nivel 2

Tales especificaciones pueden evaluarse y probarse en línea en el sitio de la W3C. En el

⁹ World Wide Web Consortium, es un consorcio internacional que produce estándares para la World Wide Web. Está dirigida por el creador original de URL (Uniform Resource Locator), HTTP, y HTML que son las principales tecnologías sobre las que se basa la Web. La liga para contactar a W3C es: <http://www.w3.org/>.

Capítulo 5: Pruebas

sistema en cuestión, la valoración de estas especificaciones que establece la W3C fue satisfactoria, sólo presentó algunas recomendaciones no críticas que no implica corrección, como el adecuado cierre de elementos vacío (como el salto de línea).

Conclusiones

Al realizar el presente trabajo, pudimos corroborar, la popularidad extraordinaria que recientemente han adquirido las aplicaciones web, debido en parte, a que ofrecen un número mayor de avances tecnológicos sobre las aplicaciones de escritorio, dentro de los cuales, se pueden mencionar:

- *Las aplicaciones web son poco costosas y fáciles de distribuir:* Ya que la aplicación se encuentra instalada en el servidor, no es necesario incurrir en los gastos asociados con la instalación de la misma en cada uno de los equipos de los usuarios. Con una aplicación web, lo que necesitan los usuarios es una computadora (relativamente económica) la cual incluso, puede ser un cliente ligero (thin client), con un explorador web y una conexión a Internet o a la Intranet de la compañía.
- *Las aplicaciones web son fáciles de actualizar y no incurren en costos altos de actualización:* Los costos de mantenimiento de software siempre han sido significativos. Debido a que actualizar una pieza existente de software es similar a instalar una nueva, las ventajas mencionadas en el punto anterior son válidas también aquí. Tan pronto como la aplicación contenida en el servidor es actualizada, todos los usuarios de la misma adquieren la nueva funcionalidad.
- *Las aplicaciones web no imponen fuertes requerimientos sobre los usuarios finales:* Con sólo tener la aplicación web instalada en un servidor, cualquier usuario final conectado a Internet o a la Intranet de la empresa la podrá utilizar, sin importar si lo hace desde una PC con Windows, una Mac o un sistema basado en UNIX. Si la aplicación se encuentra propiamente construida, se podrá ejecutar sin dificultad en cualquier explorador moderno.
- *Las aplicaciones web facilitan el tener un contenedor centralizado de la información.* Cuando se tienen distintas localidades, las cuales necesitan tener acceso a la misma información, tener dicha información almacenada en un solo lugar es más fácil que tener distintas bases de datos en cada localidad. De esta manera se evitan errores potenciales debidos a la falta de sincronía con la que se realizan distintas operaciones sobre la misma información y se incrementa la seguridad sobre la misma.

En la actualidad, se cuenta con herramientas que permiten crear aplicaciones web, con características de usabilidad similares a las que anteriormente sólo estaban presentes en aplicaciones de escritorio. Usabilidad es el arte de satisfacer las expectativas del usuario en cuanto a que se cuente con una interfaz que le permita desarrollar su trabajo de manera eficiente. Actualmente se cuenta con herramientas como AJAX que le permiten al desarrollador de la aplicación web, crear interfaces de usuario con características de

Conclusiones

usabilidad, anteriormente exclusivas de las aplicaciones de escritorio.

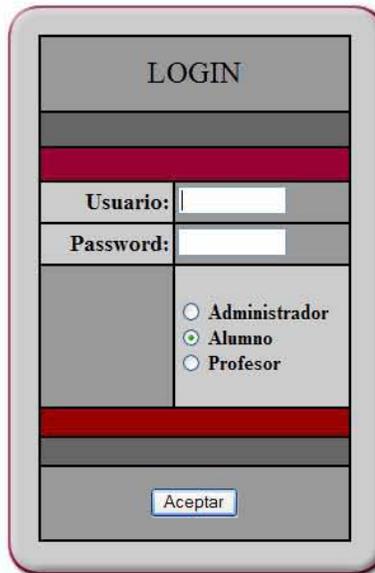
Perspectiva a Futuro del Sistema.

El sistema actualmente, implementa todos los casos de uso contemplados en la fase de diseño del mismo. Sin embargo, existen áreas donde quizás se pueda proveer al usuario de una funcionalidad mayor, en el sentido que se puedan implementar algunas de las tareas todavía llevadas a cabo de la manera tradicional, principalmente por el grupo del personal evaluador y el del personal a evaluar. Dentro de los módulos que se consideran, pudieran agregar mayor valor al sistema, se describen los siguientes:

- **Módulo de Análisis Estadístico:** La finalidad de este módulo sería la de proveer al profesor de las herramientas necesarias para llevar a cabo análisis estadísticos de los resultados obtenidos en sus grupos, por los alumnos. Principalmente se consideraría el cálculo de las medidas de tendencia central así como de las medidas de dispersión de los datos.
- **Módulo de Control Temporal en la Aplicación de Pruebas:** En el estado actual del sistema, el alumno, en el momento de haber sido dado de alta por alguno de sus profesores en algún grupo, tiene la posibilidad de ingresar al sistema y por ende, contestar el cuestionario respectivo. Lo que implica que una misma prueba puede ser contestada a diferentes tiempos por alumnos de un mismo grupo. Este módulo tiene por finalidad proveer una interfaz en la que se pueda definir la hora de inicio en la cual estará disponible una prueba para poder ser contestada por algún grupo (si el profesor así lo requiere), así como también, proveer de la funcionalidad necesaria para definir la fecha y hora en la que ningún alumno podrá ingresar al sistema para contestar la prueba.

Manual de Usuario

1. Inicialmente, se requiere que el usuario teclee su nombre de usuario, su contraseña (password) y elija a qué grupo de usuarios pertenece: *alumno*, *profesor* o *administrador*.



The screenshot shows a mobile application interface for a login screen. At the top, the word "LOGIN" is centered. Below it, there are two input fields: "Usuario:" and "Password:". To the right of these fields are three radio buttons for user roles: "Administrador", "Alumno" (which is selected), and "Profesor". At the bottom of the screen, there is a button labeled "Aceptar".

2. Si el usuario es el administrador del sistema, se le dirige al menú del administrador. Las funciones que se llevan a cabo a través de éste menú son las que se muestran a continuación:



The screenshot shows a mobile application interface for an administrator menu. At the top, the text "MENÚ ADMINISTRADOR" is centered. Below this, there are four buttons arranged vertically: "Editar Datos Personales", "Agregar Usuario", "Eliminar Usuario", and "Salir".

2.1 Si el administrador elije el botón “Editar Datos Personales”, se le presenta un formulario en el que se alteran los siguientes campos:

- Nombre.
- Apellido.
- Correo.
- Nombre de Usuario.
- Contraseña.

Cabe señalar, que los campos en los que se teclean los datos, son validados tan pronto como el campo pierde foco, es decir cuando el cursor se desplaza hacia otro, y cuando el formulario es enviado en su totalidad presionando el botón “Aceptar”. Por lo que no se guardan los datos actualizados, a menos que se tecleen los datos en el formato que son requeridos.

2.2 Si el administrador elije el botón “Agregar Usuario”, se muestra un formulario muy similar al previamente descrito. Este formulario, sirve para dar de alta a los miembros del grupo personal evaluador (profesores), de tal manera, que cuando un usuario es dado de alta a través de este formulario, tendrá la posibilidad de identificarse en el sistema como profesor e ingresar al menú correspondiente.

Este menú, también cuenta con el sistema de validación de datos, por lo que éstos deben ingresarse en el formato esperado.

2.3 Si el administrador elige el botón “Eliminar Usuario”, podrá seleccionar de una lista ordenada alfabéticamente el nombre del usuario perteneciente al grupo del personal evaluador al que desea eliminar. Para eliminar dicha cuenta, sólo basta con seleccionar el nombre del usuario y dar clic en aceptar.



2.4 Finalmente, al hacer clic en el botón “Salir” se regresa a la pantalla de inicio.

3. Si el usuario pertenece al grupo del personal evaluador, posee una cuenta del grupo y ha tecleado correctamente su nombre de usuario y contraseña (password) se le dirige al menú del personal Evaluador donde podrá realizar un conjunto de funciones asociadas a este grupo.



3.1 Si el usuario evaluador elije el botón “Editar Datos Personales”, se presenta un formulario donde alterará los siguientes campos:

- Correo del Evaluador.
- Nombre de Usuario del Evaluador.
- Contraseña del Evaluador.

Formulario Personal Evaluador	
Información Personal	
Nombre:	Luciralia
Apellido:	Hernández
E-Mail:	luciraliahdez@correo.com
Login	
Usuario:	lucy
Password:	
<input type="button" value="Aceptar"/> <input type="button" value="Cancelar"/>	

Cabe señalar, que los campos en los que se teclean los datos, son validados tan pronto como el campo pierde foco, es decir, cuando el cursor se desplaza hacia otro, y cuando el formulario es enviado en su totalidad, presionando el botón “Aceptar”. Por lo que no se podrán guardar los datos actualizados, a menos que se hallan tecleado los datos en el formato en el que son requeridos. También, los campos correspondientes al nombre y al apellido del evaluador aparecerán en el formulario, sin embargo, éstos no son susceptibles a ser editados, por razones administrativas.

- 3.2 En primer término, un usuario evaluador deberá crear uno o más grupos, de tal manera que pueda asociar pruebas y alumnos con el/los mismo(s), para ello deberá presionar el botón “Crear / Eliminar” grupos de esta manera, podrá ingresar a la interfaz correspondiente. Ésta cuenta con un campo identificado como clave de la asignatura, y otro como número de grupo, los cuales, en su conjunto, definen un grupo del cual el profesor es tutor. Además, abajo del formulario aparecerá un listado de los grupos dados de alta con anterioridad.

ALTA-BAJA DE GRUPOS

Profesor: Luciralia Hernández

Nuevo grupo

Clave de asignatura:

Número de grupo:

Agregar

Tabla de grupos

Clave de asignatura	Número de Grupo	Eliminar?
1111	1	<input type="radio"/>

Eliminar

Regresar

Cabe señalar que la aplicación, no permitirá que sean ingresados dos grupos con la misma clave de asignatura y el mismo número de grupo, y que en dicho caso, presentará un mensaje notificando al usuario que el grupo ya existe.

- 3.3 Posteriormente, el profesor podrá asociar alumnos o pruebas con el/los grupo(s) dados de alta en el punto anterior. Para asociar un listado de alumnos con un grupo en particular, el profesor deberá presionar el botón “Asociar Alumnos a Grupo” localizado en el menú principal del personal evaluador. Una vez hecho esto, se requiere que el profesor seleccione de la lista, algún grupo que haya sido dado de alta con anterioridad por el mismo, para posteriormente acceder a una interfaz en la cual podrá ingresar los datos de los alumnos del grupo. También, en esta misma interfaz podrá consultar los datos de los alumnos que haya dado de alta en un grupo con anterioridad.

ALTA-BAJA DE ALUMNOS

Profesor: Luciralia Hernández
Grupo: 1
Clave de Asignatura: 1111

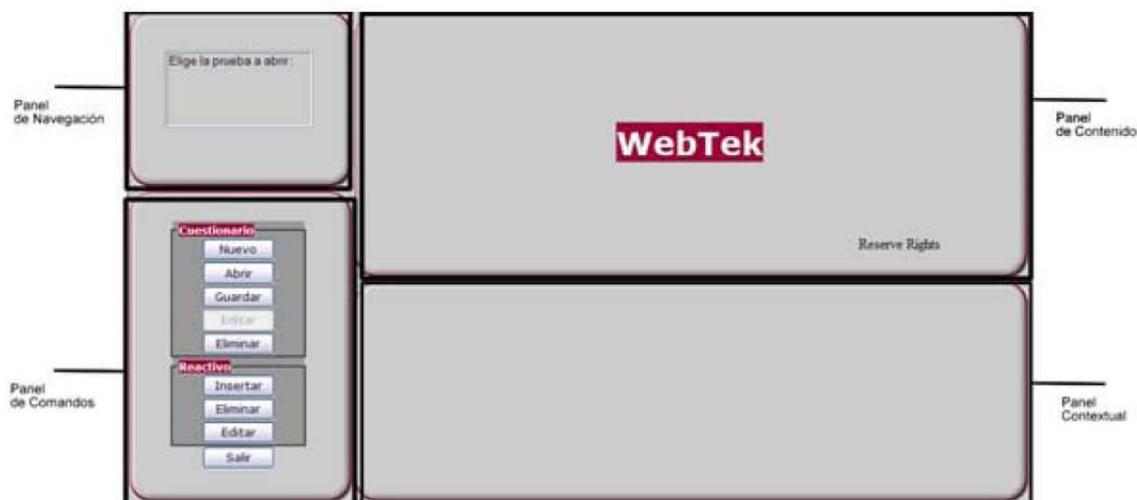
Nuevo alumno

Número de cuenta :	099607245
Nombre :	Gerardo
Apellidos :	Ramírez Lerma
Correo-e:	ammrez78@correo.com

Tabla de alumnos

Número de cuenta	Nombre	Correo-E	Eliminar?
099607245	Gerardo Ramirez Lerma	rammrez78@correo.com	<input type="radio"/>

3.4 Una vez que el profesor ha dado de alta cuando menos un grupo, podrá definir cuestionarios para que sean contestados por los alumnos. Éstos, son definidos a través de una interfaz especializada, la cual se muestra a continuación.



La interfaz consta de cuatro paneles, éstos son:

- Panel de Navegación.
- Panel de Comandos.
- Panel de Contenido.
- Panel Contextual.

Panel de Navegación: Contiene un listado de las pruebas definidas con anterioridad, para desplegar cualquiera de ellas en el panel de contenido, basta hacer clic sobre el nombre de la prueba. Para eliminar una prueba listada en este panel, es necesario pulsar el botón “Eliminar Prueba” del panel de comandos y posteriormente seleccionar de este panel la prueba que se desea eliminar.

Panel de Comandos: Contiene un conjunto de botones los cuales sirven para ejecutar comandos relacionados con la creación, eliminación y alteración de pruebas. Los botones se encuentran agrupados en dos grupos principales, el primero se relaciona con los comandos que se pueden ejecutar sobre las pruebas y estos incluyen: abrir, guardar y eliminar prueba. El segundo grupo de botones, está relacionado con comandos relacionados con la alteración de una prueba previamente definida, estos incluyen: insertar reactivo, eliminar reactivo y editar reactivo.

Panel de Contenido: Aquí se muestra la prueba que está siendo editada. Consiste en un área de texto donde se aprecia el encabezado de la prueba y los reactivos que han sido elegidos para la prueba. Los reactivos cuentan con una marca, en alguna de sus opciones, para indicar que opción, es la correcta.

Panel Contextual: Este panel es polimórfico, es decir, que cambia de contenido dependiendo del comando que se esté ejecutando a través del panel de comandos.

3.5 Si se hace clic en el botón “Consultar Calificaciones” se selecciona un grupo, y a continuación se le presentará un listado de todas las pruebas definidas para dicho grupo. Al seleccionar una prueba el profesor tendrá acceso a una interfaz en la que podrá consultar los datos de los alumnos de dicho grupo y la calificación obtenida en dicha prueba. Cabe señalar que cada registro posee también un botón para tener acceso a la prueba contestada por el alumno así como un control que al seleccionarlo y presionar el botón “Borrar” eliminará el registro correspondiente en la base de datos, permitiéndole al alumno volver a realizar la prueba.

3.6 Finalmente, si se pulsa el botón “Salir” en el Menú Profesor el usuario es dirigido hacia la interfaz inicial.

4. Si el usuario pertenece al grupo del personal a evaluar, posee una cuenta de dicho grupo y ha tecleado correctamente su nombre de usuario y contraseña (password), se le dirige al menú del Alumno donde realizará un conjunto de funciones asociadas a este grupo.

4.1 Si el usuario alumno elige el botón “Editar Datos Personales”, se le presenta un formulario en el que podrá alterar los siguientes campos:

- Correo del Alumno.
- Nombre de Usuario del Alumno.
- Contraseña del Alumno.

Cabe señalar, que los campos en los que se teclean los datos, son validados tan pronto como el campo pierde foco, es decir cuando el cursor se desplaza hacia otro botón, y cuando el formulario es enviado en su totalidad, presionando el botón “Aceptar”. Por lo que no se guardarán la información actualizada, a menos que se hallan tecleado los datos en el formato en el que son requeridos. También, los campos correspondientes al nombre y al apellido del evaluador aparecerán en el formulario, sin embargo éstos no son susceptibles a ser editados, por razones administrativas.

4.2 Si el usuario alumno hace clic en el botón “Contestar Prueba”, se le presenta un listado de los grupos en los que ha sido de alta por algún profesor, al hacer una selección se le presenta un listado con el nombre de las pruebas dadas de alta para ese grupo y que no hayan sido contestadas por dicho usuario con anterioridad. Al seleccionar una de las pruebas se le presenta una interfaz, en la que al momento de estar listo para contestar la prueba, deberá presionar el botón iniciar, para que se le presente el contenido de la prueba, y comience a correr el tiempo asignado a la misma, una vez que termine de contestar la prueba, el alumno deberá presionar el botón enviar. Cabe señalar que una vez que se termine el tiempo en el reloj, se enviara la prueba al servidor en el estado en que se encuentre impidiéndole al alumno seguir con la misma.

4.3 Si el usuario hace clic en el botón “Consultar Calificaciones” entonces se le presenta un listado donde elegirá al grupo del que desea consultar una calificación de un examen en particular y finalmente, se le presenta un listado de los exámenes contestados por los alumnos. Donde se muestra su calificación y la fecha en que fue resuelta la prueba. Así mismo, el alumno al presionar un botón contenido en dicha interfaz será capaz de revisar su examen y comparar sus respuestas con las correctas.

4.4 Finalmente, si se pulsa el botón “Salir” en el Menú Alumno, el usuario es dirigido hacia la interfaz inicial.

A

Access (*EntidadBase, EntidadAcceso1, ..., EntidadAccesoN*). Fragmento de hipertexto que denota acceso a una o más instancias de la entidad EntidadBase, por medio de la selección paso a paso a través de las entidades clasificadas EntidadAcceso1, ..., EntidadAccesoN.

Aplicación back-end. Es la parte responsable de procesar los datos del usuario con los recursos informáticos dispuestos en el servidor, para finalmente obtener una respuesta hacia el usuario por medio de la aplicación front-end.

Aplicación front-end. Es responsable de la recolección de entradas del usuario, las cuales pueden estar en una variedad de formas.

Aplicación Web. Son aplicaciones que son procesadas en un servidor Web.

Área. Agrupa a un conjunto de páginas con un fin específico, organizar las funciones de un grupo de usuarios.

B

Base de datos. Conjunto de información organizada en un archivo lógico, conteniendo uno o más archivos físicos.

C

Cliente Web. Una computadora conectada a la red del Internet solicitando al servidor Web información disponible por medio de un Web Browser.

Core (*EntidadBase, Componente1, ..., ComponenteN*). Fragmento de hipertexto que denota la publicación de contenido acerca de la entidad EntidadBase y sus sub-componentes entidades Componente1, .. , ComponenteN.

Create&Connect (*Entidad1, Regla1, ..., ReglaN*). Fragmento de hipertexto que denota la creación de una nueva instancia de la entidad Entidad1, y posiblemente la conexión de la nueva instancia a N existentes entidades por medio de la regla de relación Regla1, ..., ReglaN.

D

Datos. Valores registrados físicamente en la base de datos.

Delete (*Entidad1*). Fragmento de hipertexto que la eliminación de instancias de la entidad Entidad1.

E

Estrés de trabajo. Exponer en casos limite las solicitudes al servidor Web. La intención es valorar la respuesta del servidor a casos extremos provenientes de solicitudes del cliente.

Enlace. Especifican la ruta o camino de navegación permitido entre páginas y áreas, son las selecciones disponibles para el usuario.

F

Fragmento de hipertexto. Especifican el contenido asociado en cada área, tomando a las instancias y entidades como elementos de operación para formar las declaraciones. Estas declaraciones gestionarán y conectarán las diversas instancias contenidas en el diagrama de datos. Los fragmentos de hipertexto tienen una representación (no única) en notación WebML.

H

Hiperenlaces. Son texto o gráficos que al realizar click con el puntero del ratón, se visualiza otra información.

Hipertexto. Es el contenido publicado en una aplicación Web organizado en bloques de información expuestos uno a uno al usuario, desplazándose por cada bloque a través de enlaces. Estos bloques de información son representados por páginas.

I

Internet. Es una red global de computadoras. Una red global de computadoras enlazadas juntas para intercambiar información.

Intranet. Es una red interna, similar al Internet, pero con acceso limitado a individuos autorizados.

M

Modelo de desarrollo de software. Procedimientos organizados y estructurados en determinadas reglas que establecen la manera de desarrollar software, tales procedimientos se apoyan en los siguientes pasos: análisis de requisitos, especificación de requerimientos, diseño y arquitectura, implementación en la programación, pruebas, mantenimiento y documentación.

Modif. (Entidad1). Fragmento de hipertexto que denota la modificación de instancias de la entidad Entidad1.

P

Página. Son construcciones que albergan a diversas unidades, agrupadas en una vista del sitio. Las páginas son elementos de interfaz presentadas al usuario, quien navega por el hipertexto para acceder a estas páginas en determinada secuencia .

S

Servidor Web. Es una computadora conectada al Internet o intranet, ejecutando software capaz de manipular solicitudes HTTP. También conocida como una computadora montada para servir páginas Web e información relacionada.

Software. Es un conjunto de instrucciones que permite al hardware de la computadora permite desempeñar trabajo útil.

U

Unidad. Son piezas elementales para la publicación de contenido y datos provenientes de la entrada del usuario. Son los bloques de construcción para las páginas. En notación WebML, cada unidad posee una función y una correspondiente notación gráfica.

Unidad de operación. Son un tipo de unidad colocada fuera de las páginas y enlazadas a otras operaciones o a otras unidades de contenido situadas dentro de las páginas. Esta unidad no despliega contenido, proporciona operaciones orientadas a tareas administrativas de datos y a servicios de registro y de mensajería, y a otros servicios externos a la aplicación Web.

V

Vista del sitio. Encierran a un grupo de páginas. La vista del sitio alberga publicación de contenido y funciones asociado a cada grupo de usuarios.

W

Web. Conocida como la World Wide, es una parte del Internet que proporciona información, usando documentos que incluyen uno o más de los siguientes componentes: texto con formato, imágenes, hiperenlaces y formularios que permiten al usuario ingresar información para comunicarse directamente con el servidor.

Web browser. Es un programa de software utilizado para visualización y navegación de documentos Web.

WebML. Lenguaje de Modelado Web, proporciona primitivas (elementos gráficos) para modelar el hipertexto. Utiliza la notación gráfica heredada del diagrama Entidad-Relación para representar de una manera simple e intuitivamente los conceptos,

notaciones y métodos de diseño, cuya finalidad sea la de publicar y manipular los datos y al contenido de la aplicación Web. Las partes de WebML son: las páginas, las unidades, y los enlaces organizados en construcciones modulares como áreas y vistas del sitio.

Referencias Bibliográficas

Zakas C., Nicolas. PROFESSIONAL JAVASCRIPT FOR DEVELOPERS. Wiley Publishing, U.S.A., 2005.

Powers, Shelly. LEARNING JAVASCRIPT. O'Reilly Publisher, U.S.A., 2006.

Ceri, Stefano. DESIGN DATA-INTENSIVE WEB APPLICATIONS. Morgan Kaufmann Publishers, U.S.A. 2003.

Zaldivar Z., Orlando. APUNTES DE INGENIERÍA DE PROGRAMACIÓN. Facultad de Ingeniería, Universidad Nacional Autónoma de México, 2000.

Darie, Cristian. BUILDING RESPONSIVE WEB APPLICATIONS. Packt Publishing, U.S.A., 2006.