


Auxiliar uml

marcelo vargas

Related papers

[Download a PDF Pack](#) of the best related papers 



[Tesis Modelado Visual de Aplicaciones Lotus Domino](#)

Hiriam Perez

2.3.- UML

UML es un lenguaje de modelación estandarizado para especificar, visualizar, construir y documentar todos los elementos de un sistema de software [11]. Este lenguaje es una síntesis de varias notaciones originalmente desarrolladas por Grady Booch, Jim Rumbaugh, Ivar Jacobson y muchos otros.

Desde 1991 empezaron a publicarse libros sobre metodologías de modelado y aunque eran independientes y cada una tenía sus cualidades, el objetivo era el mismo. Esto continuó así hasta 1994, cuando Rumbaugh, quien trabajaba con su metodología conocida como OMT, se unió a Rational y empezó a trabajar junto con Booch, hasta que finalmente, en octubre de 1995 presentaron el Método Unificado, el cual se convertiría posteriormente en lo que hoy se conoce como UML. Esto fue en 1997, cuando Jacobson y su metodología “Objectory” se unió al equipo e introdujeron el Lenguaje Unificado de Modelación al OMG para que se convirtiera en un estándar. Como se ilustra en la figura 2.2, UML es una síntesis de diferentes notaciones, creadas por Grady Booch, Jim Rumbaugh e Ivar Jacobson, a quienes se les conoce como “Los Tres Amigos”

El uso y diseño de extensiones se vuelve una necesidad para este proyecto, ya que al momento de considerar los elementos estándar de la modelación con UML, se vuelve evidente de que no todos los elementos de la aplicación encajan dentro de su descripción [3].

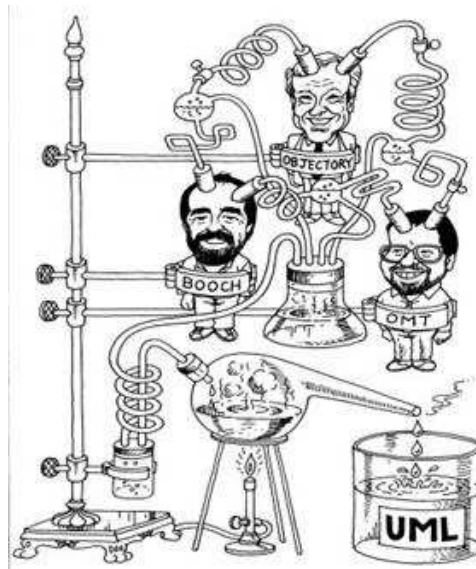


Figura 2.2.- Origen del UML

Terry Quatrani muestra un ejemplo de un modelo usando UML e introduce el uso de una extensión para html [11]. Su ejemplo se basa en un pequeño sistema de registro de cursos para alumnos y maestros y en este caso inicia con un diagrama de actividad, el cual se muestra en la figura 2.3.

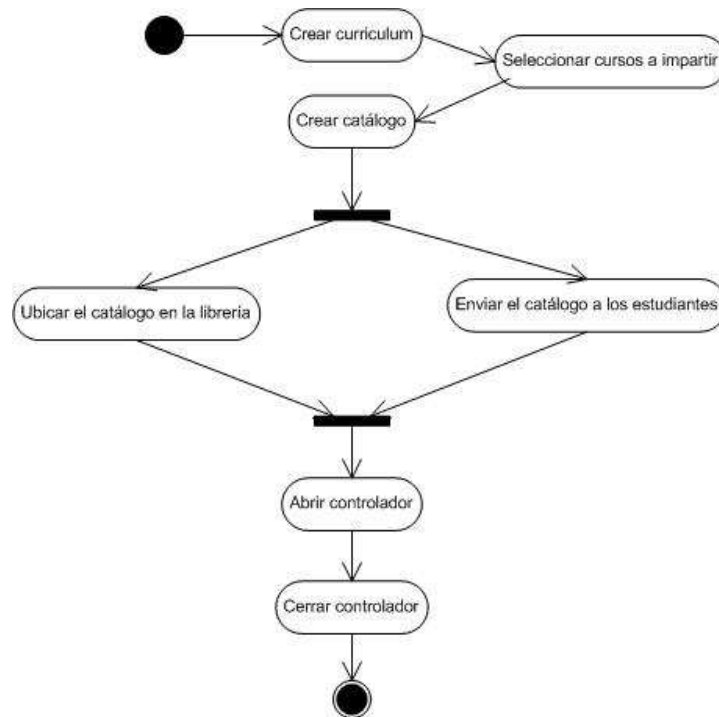


Figura 2.3.- Diagrama de actividad

El diagrama en la figura 2.3 muestra un flujo de actividades representadas en rectángulos redondeados. Las actividades son por lo general estados de acción y se puede transferir al siguiente estado cuando la acción ha sido completada. El círculo negro representa el inicio del diagrama de actividad, donde inicia el control del flujo. Las flechas indican transiciones de un estado a otro y no se puede realizar la transición hasta que la actividad haya sido completada. Las barras de sincronización (representadas en el diagrama como dos barras negras horizontales) indican cuando algunas actividades suceden en paralelo.

El siguiente diagrama que muestra el ejemplo del sistema de registros de cursos para alumnos y maestros es el diagrama de casos de uso, este se crea ubicando primeramente a los actores. Un actor es alguien o algo que es externo al sistema, pero que va a interactuar con el mismo. Para uso del ejemplo, se requieren los siguientes actores: controlador, profesor, estudiante y un sistema de cobro, los cuales están ilustrados en la figura 2.4.

Se puede decir que un caso de uso es una parte de la funcionalidad del sistema, no es un módulo de software, sino algo que le da valor al actor.

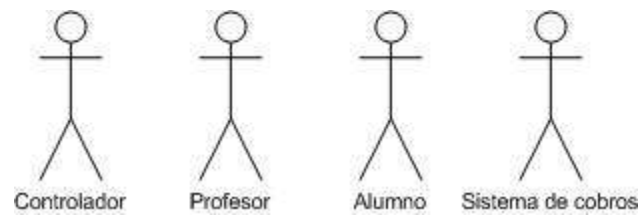


Figura 2.4.- Actores

La figura 2.5 muestra los diagramas de casos de uso, los cuales están representados por óvalos. La manera más fácil de identificar los casos de uso de un sistema es observando a los actores y preguntarse cuál va a ser su relación con el sistema. En el ejemplo de la figura 2.5, el controlador va a mantener el currículum, el profesor va a solicitar su temario, el estudiante va a formar su horario y el sistema de cobro va a recibir la información de los cargos.

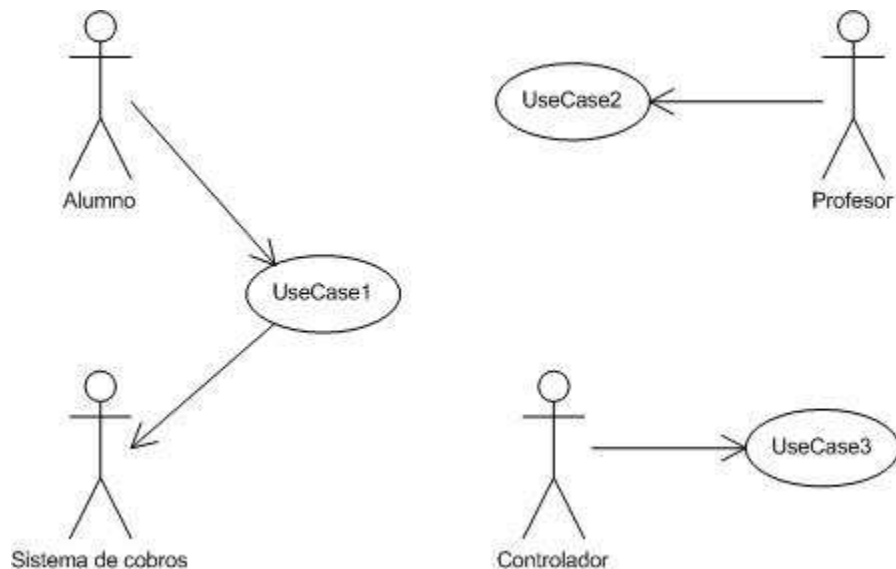


Figura 2.5.- Diagrama de casos de uso

El siguiente paso es documentar cada caso de uso. Cada uno debe describir el flujo de eventos y esto es realizado desde el punto de vista del actor. Debe detallar que es lo que el sistema va a proveer al actor cuando el caso de uso sea ejecutado. Por lo general indica cómo es que se inicia el caso de uso y como termina. Una vez realizados estos pasos, contamos con una

buena perspectiva general del sistema, muestra que es lo que esta fuera del sistema (actores) y la funcionalidad que el sistema debe de proveer (casos de uso).

Un punto a tomar en cuenta sobre los casos de uso es la realización del caso, o como es que va a ser ejecutado el caso de uso. Esto se puede realizar mediante el uso de tres tipos de diagramas: de secuencia, de colaboración y de clases. A este último también se le llama una vista de las clases participantes.

Diagramas de secuencia: Aquí se muestran las interacciones entre distintos objetos organizados en una secuencia de tiempo. Se puede utilizar el flujo de eventos para determinar qué objetos e interacciones van a ser necesarias para lograr la funcionalidad especificada por el flujo de eventos. En el diagrama de secuencia de la figura 2.6, se muestra como un estudiante se registra exitosamente en un curso. El estudiante llena una forma la envía, la forma se comunica con el manejador y le solicita agregar al estudiante al curso indicado. El manejador le informa al curso que desea agregar un estudiante, el curso le pregunta a la sección 1 si se encuentra con cupo. En este caso, la sección 1 contesta afirmativamente y entonces el curso le solicita a la seccion1 agregar al estudiante a su lista. Este diagrama muestra paso a paso al cliente que es lo que va a ocurrir en el sistema.

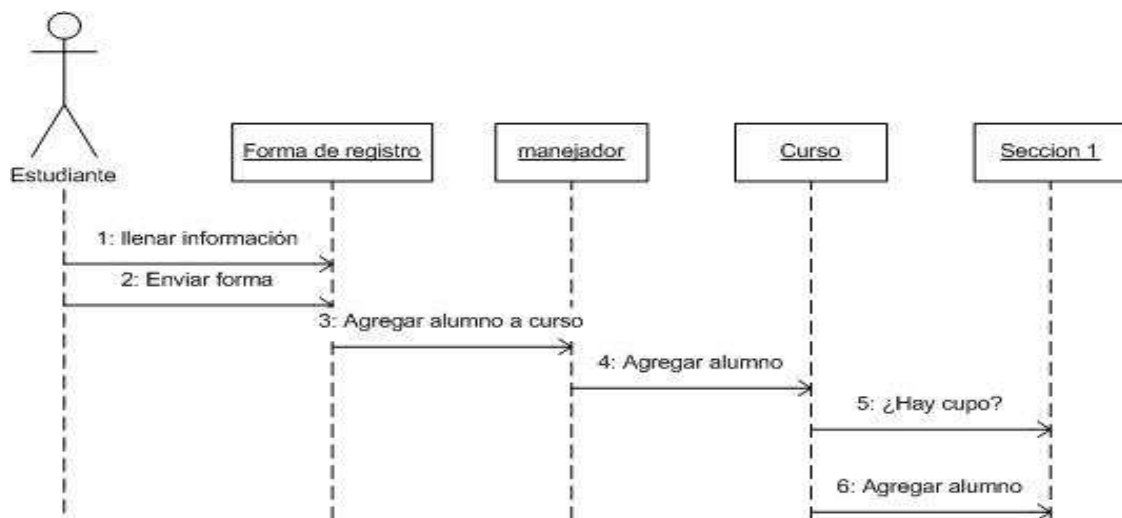


Figura 2.6.-: Diagrama de secuencia

Diagramas de colaboración: En el diagrama mostrado en la figura 2.7 se muestran los objetos y sus relaciones con los actores, solo que no de acuerdo al tiempo, sino que son mostrados basándose en las relaciones entre los objetos. Este tipo de diagramas es de gran

utilidad cuando se quieren visualizar todos los mensajes entre algunos objetos para un caso de uso o escenario en particular, especialmente en el caso de un escenario grande.

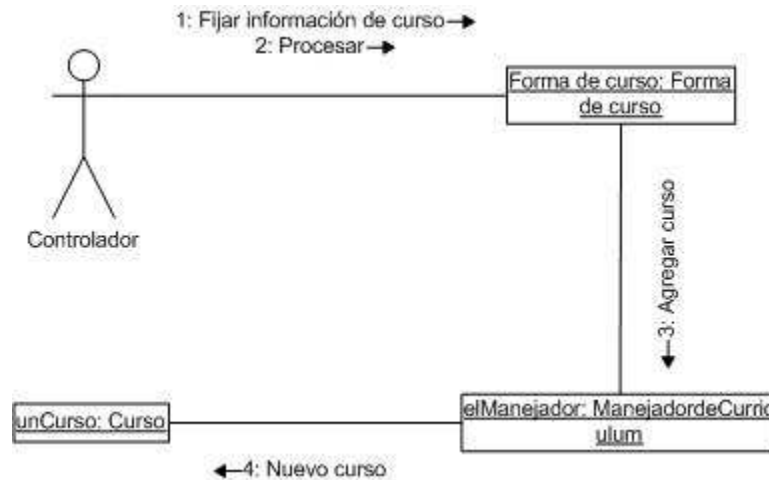


Figura 2.7.- Diagrama de colaboración

Diagramas de clases: Una clase es una colección de objetos con una estructura común, al igual que su comportamiento, sus relaciones y su semántica. Se pueden identificar examinando los objetos de los diagramas de secuencia y de colaboración y en UML son representados por un rectángulo con tres compartimientos. En la figura 2.8 se muestran las clases involucradas con el ejemplo.

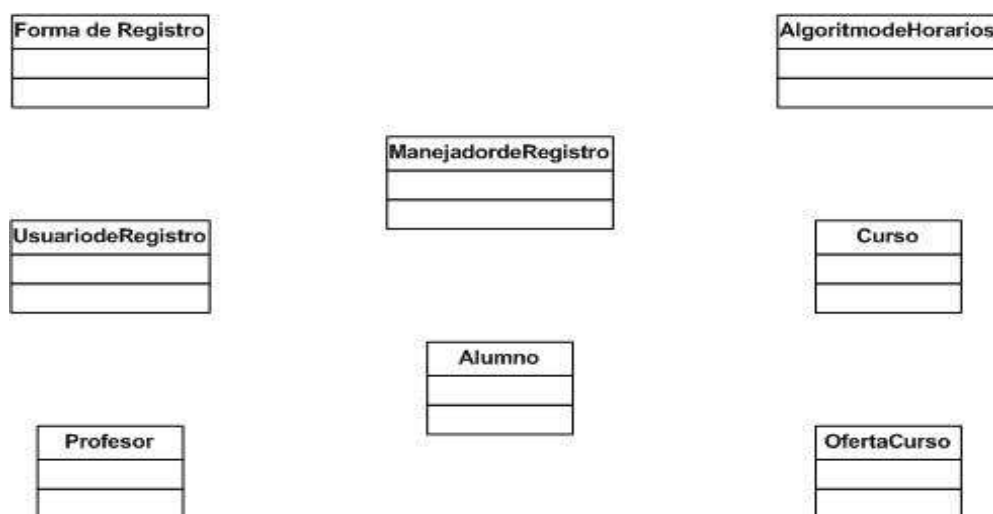


Figura 2.8.- Clases

El compartimiento superior de cada recuadro muestra el nombre de la clase, el de en medio muestra su estructura (atributos) y el inferior su comportamiento (operaciones). La herramienta Microsoft Visio, usada para la elaboración de los diagramas aquí mostrados utiliza los signos (-) en cada renglón de los atributos y el signo (+) en cada renglón de las operaciones a manera de poder identificarlas y diferenciarlas.

Estos compartimentos pueden ser usados a conveniencia del modelador, de forma que se vea solo el nombre, el nombre y sus atributos o los tres. Se debe tener en consideración al nombrar las clases, usar un vocabulario del dominio y usarlo como un estándar. El diagrama de clases debe incluir además información sobre las relaciones y dependencia de las clases, así como la multiplicidad y su navegación, entre otras cosas. La figura 2.9 muestra el diagrama de clases completo para el ejemplo en cuestión.

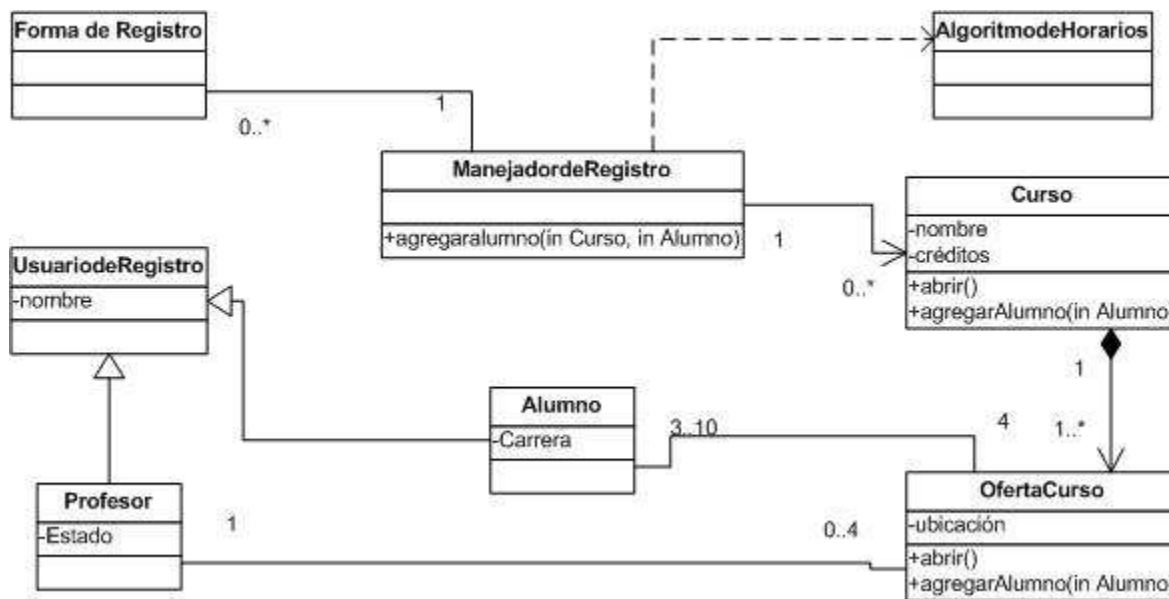


Figura 2.9.- Diagrama de clases

Las relaciones representan caminos de comunicación entre objetos. Existen tres tipos de relaciones: de asociación, agregación y dependencia.

- Las relaciones de asociación representan una conexión bi-direccional entre clases y son representadas por UML como una línea que conecta a las clases relacionadas. En la figura 2.9 se puede encontrar esta asociación entre las clases **FormadeRegistro** y **ManejadordeRegistro**.

- Las relaciones de agregación son una forma más fuerte de relación, donde ésta se da entre un todo y sus partes. Esto le indica al desarrollador que existe un alto acoplamiento entre esas clases de objetos. En UML se representan con una línea que conecta a las clases relacionadas y tiene una figura de diamante junto a la clase que representa al todo. Este tipo de relación se puede apreciar entre las clases **Curso** y **OfertaCurso** ilustradas en la figura 2.9.
- Las relaciones de dependencia, que son una relación más débil entre un cliente y un proveedor, donde el cliente no tiene el conocimiento semántico del proveedor y quiere decir que necesita el servicio, pero no sabe que existe. En UML son representadas por una línea punteada que apunta del cliente hacia el proveedor. En el diagrama de la figura 2.9 se puede encontrar esta asociación entre las clases **ManejadordeRegistro** y **AlgoritmodeHorarios**.

La multiplicidad define cuantos objetos participan en una relación. Es el número de instancias de una clase en relación a una instancia de la otra clase y esta se debe realizar en cada lado de la relación.

La navegación es representada con una flecha y aunque las asociaciones y agregaciones son bi-direccionales por defecto, es por lo general deseable restringir la navegación a una sola dirección. Este último punto no es necesario en todas las relaciones, sino solo en las que el modelador quiera asegurarse que sea unidireccional.

Si existen relaciones entre superclases y subclases, se puede indicar herencia de comportamientos de una a otra usando un triángulo indicando que una clase es usuaria de otra. Una subclase hereda la definición de la superclase de la que procede, sin embargo puede agregarle o hacerle cambios a la definición. Una superclase puede tener diversas subclases y éstas pueden tener sus propias subclases.

El siguiente proceso es visualizar el sistema dentro del mundo real y para esto se pueden usar los diagramas de componentes y de despliegue. El primero se puede observar en la figura 2.10. Este se usa para ilustrar las organizaciones y dependencias entre los componentes de software, incluyendo componentes de código fuente, de tiempo de ejecución o ejecutables. UML los muestra como un rectángulo grande con dos pequeños a un lado. Los círculos representan

interfaces. En el caso del diagrama 2.10, se indica que el componente **Controlador.exe** es dependiente de interfaces con **Cursos.dll**, con **Personas.dll** y con **Cobros.exe**.

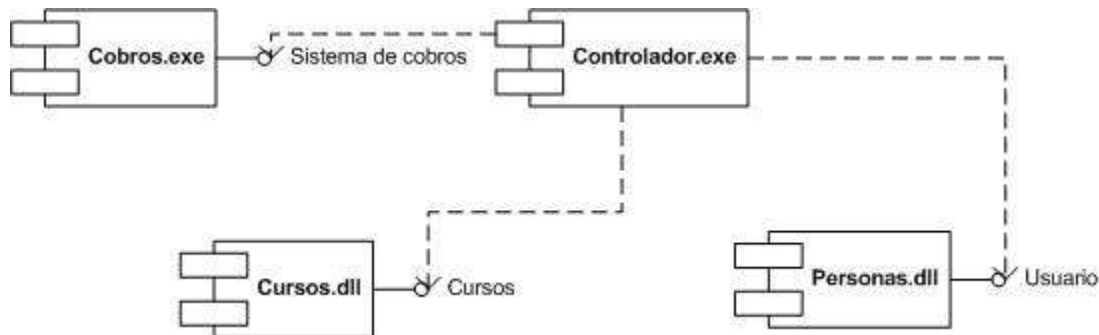


Figura 2.10.- Diagrama de componentes

El diagrama de despliegue es crucial, ya que muestra a los procesadores del sistema y las conexiones entre ellos, también muestran la distribución de los componentes en la empresa y se sabe qué se ejecuta en cuál procesador (figura 2.11).

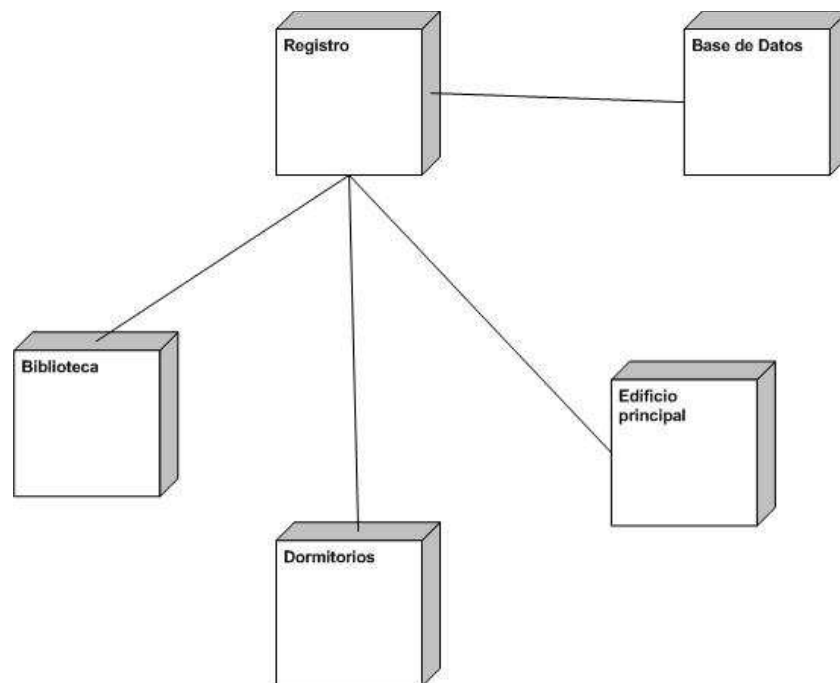


Figura 2.11.- Diagrama de despliegue

Por último, la notación de UML se puede extender para complacer las necesidades de todo modelador y ofrece el concepto de estereotipos. Esto es, se puede tomar un elemento de

modelación básico y darle más significado. Los estereotipos pueden ser usados para clasificar y extender asociaciones, relaciones de herencia, clases y componentes.

La figura 2.12 muestra un diagrama con los estereotipos web del ejemplo representado con los diagramas anteriores. Los pequeños engranes representan acciones que son ejecutadas en el servidor, así que se puede visualizar lo que es ejecutado por el cliente, lo que es ejecutado en el servidor y con que objetos tiene que tratar cada uno. Por ejemplo, desde la página **Inicio** existe un enlace o liga hacia la **Lista de categorías**. En la página **Inicio**, el usuario sigue la liga y se lleva a cabo una función por parte del servidor para revisar el catalogo, obteniendo las categorías existentes en la clase **Catalogo**. La respuesta del servidor es mostrar la página con la lista de catálogos.

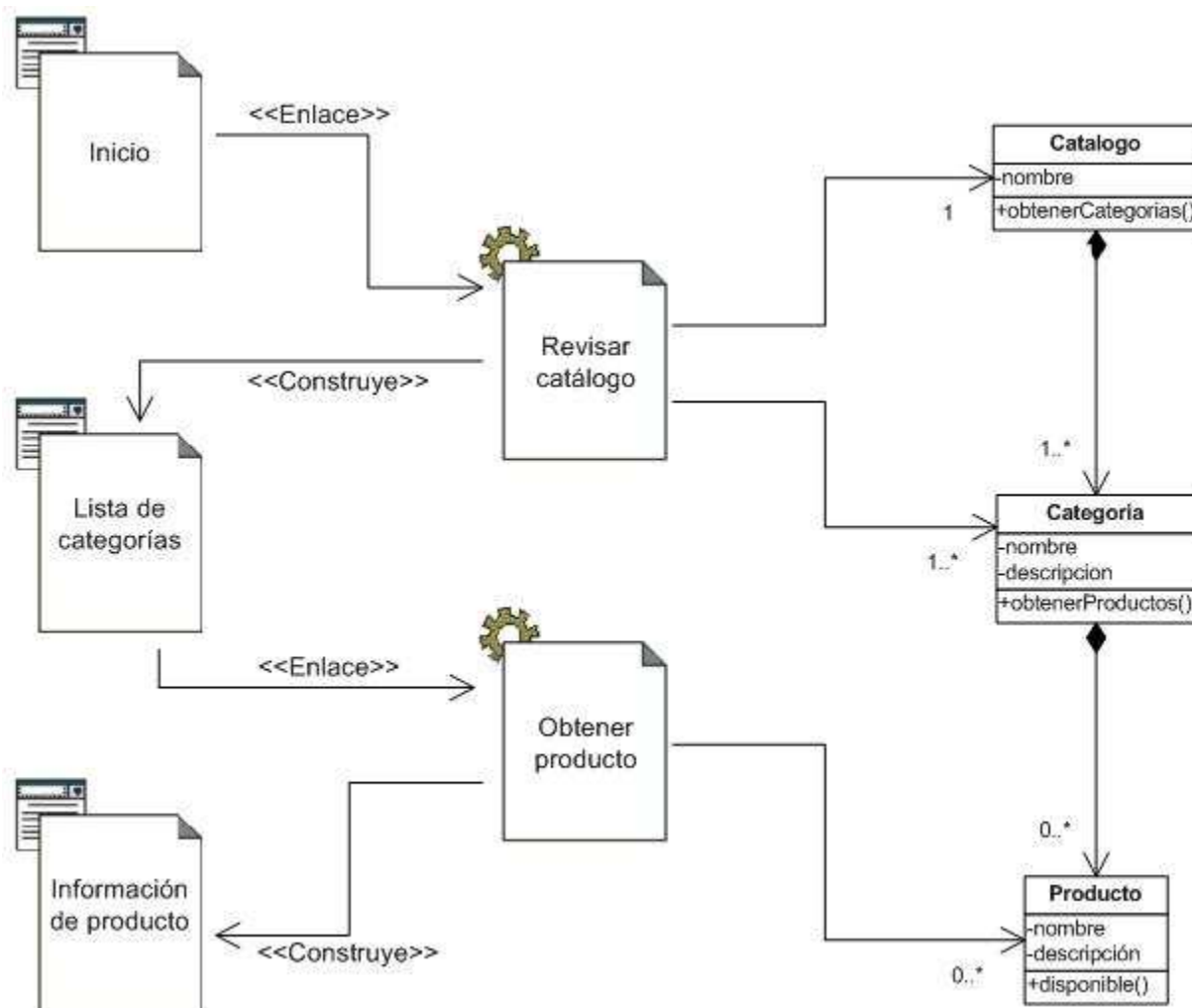


Figura 2.12.- Ejemplo de estereotipos web

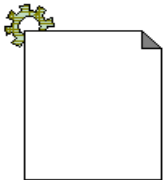
Aunque exista una gran diversidad de características y funciones específicas en cada aplicación web, se presentan elementos en común que pueden agruparse de tal manera que mediante abstracción se llegue a un grupo de modelos básicos que sirven para modelar las arquitecturas de cualquier tipo de aplicación.

Actualmente se está empezando a considerar una definición de estereotipos de algunos de los componentes que integran una aplicación web. Dicha definición aun no se encuentra integrada a las aplicaciones dedicadas a la modelación de aplicaciones, ni se cuenta con una definición clara de su integración a las metodologías existentes de modelación.

2.4.- EXTENSIÓN DE UML PARA APLICACIONES WEB

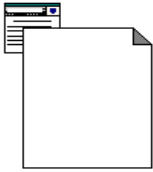
La extensión actual de UML para aplicaciones web define una serie de estereotipos, valores etiquetados y limitantes que permiten modelar aplicaciones web, los cuales son aplicados a ciertos componentes de las mismas y nos permite representarlos dentro del mismo modelo, en conjunto con los diagramas que describen el resto del sistema.

A continuación se muestran los estereotipos definidos en extensión de UML creada por Conallen [4]:

<p><u>Nombre:</u> Server Page</p> <p><u>Clase de Metamodelo:</u> Class</p> <p><u>Descripción:</u> Una página de servidor representa una página web que tiene scripts que son ejecutados por el servidor. Estos scripts interactúan con recursos en el servidor (bases de datos, lógica de negocios, sistemas externos, etc.). Las operaciones del objeto representan las funciones en el script y sus atributos representan las variables que son visibles en el alcance de la página (accesibles por toda función de la página).</p> <p><u>Icono:</u> </p> <p><u>Limitantes:</u> Las páginas de servidor solo pueden tener relaciones con objetos en el servidor.</p> <p><u>Valores etiquetados:</u> Scripting Engine: Ya sea el lenguaje o el motor que deba ser usado para ejecutar o interpretar esta página (JavaScript, VBScript, Perl, etc.)</p>
--

<p><u>Nombre:</u> Client Page</p> <p><u>Clase de Metamodelo:</u> Class</p> <p><u>Descripción:</u> Una instancia de una página cliente es una página web con formato HTML. Como cualquier página HTML, es una mezcla de datos, presentación e incluso lógica. Las páginas cliente son mostradas por</p>
--

navegadores de cliente y pueden contener scripts que son interpretados por el navegador. Las funciones de las páginas cliente se dirigen a funciones en etiquetas de scripts en la página. Los atributos de la página cliente se dirigen a variables declaradas en las etiquetas de scripts de las páginas que son accesibles por cualquier función de la página (en su alcance).



Icono:

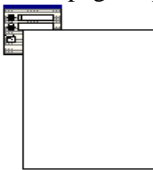
Limitantes: Ninguna.

Valores etiquetados: TitleTag – El título de la página como se muestra en el navegador. BaseTag – El URL base para quitar referencias a URL's relativos. BodyTag – El conjunto de atributos para la etiqueta <body> que fija los atributos de fondo y texto por defecto.

Nombre: Form

Clase de Metamodelo: Class

Descripción: Una clase estereotipada como <<form>> es una colección de campos de entrada que forman parte de una página cliente. Una clase Form direcciona directamente a la etiqueta form de HTML. Sus atributos representan los campos de entrada de la forma de HTML (cajas de entrada, áreas de texto, botones de radio y campos ocultos. Un <<form>> no tiene operaciones dado que no pueden ser encapsuladas en una forma. Cualquier operación que interactúe con la forma sería propiedad de la página que contiene dicha forma.



Icono:

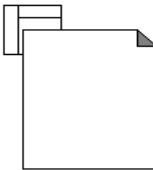
Limitantes: Ninguna.

Valores etiquetados: Method – El método usado para enviar datos al URL de acción, ya sea GET o POST.

Nombre: Frame Set

Clase de Metamodelo: Class

Descripción: Un conjunto de marcos (frame set) es un contenedor de múltiples páginas web. El área de vista rectangular es dividida en marcos rectangulares pequeños. Cada marco puede ser asociado con un <<target>> de nombre único, aunque no es siempre necesario. El contenido de un marco puede ser una página web u otro conjunto de marcos. Una clase estereotipada como Frame Set direcciona directamente a una página web frameset y a la etiqueta frame de HTML. Dado que un frameset es una página cliente, puede tener operaciones y atributos, pero son solo activados por navegadores que no muestran marcos.



Icono:

Limitantes: Ninguna.

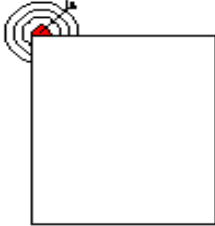
Valores etiquetados: Rows – El valor del atributo de renglones de la etiqueta <frameset> de HTML. Esta es una cadena de texto con alturas de renglón delimitadas por comas. Cols – El valor del atributo de columnas de la etiqueta <frameset> de HTML. Esta es una cadena de texto con anchos de

columna delimitados por comas.

Nombre: Target

Clase de Metamodelo: Class

Descripción: Un blanco (target) es un compartimiento nombrado en una ventana de navegador en donde se pueden mostrar las páginas web. El nombre de la clase estereotipada es el nombre del blanco. Por lo general, un blanco es un marco de una ventana definida como frameset, sin embargo puede ser una instancia o ventana del navegador. Asociaciones del tipo <<Targeted link>> especifican blancos como si el lugar fuera una página web nueva a ser mostrada.



Icono:

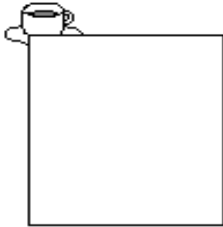
Limitantes: El nombre de un blanco debe ser único para cada cliente del sistema. Esto quiere decir que solo una instancia de un blanco puede existir en el mismo cliente.

Valores etiquetados: Ninguno.

Nombre: JavaScript

Clase de Metamodelo: Class

Descripción: En un navegador habilitado para JavaScript es posible simular objetos definidos por el usuario con funciones JavaScript. Las instancias <<JavaScript>> existen solo en el contexto de páginas cliente.



Icono:

Limitantes: Ninguna.

Valores etiquetados: Ninguno.

Nombre: Link

Clase de Metamodelo: Association

Descripción: Una liga (link) es un apuntador de una página cliente a otra <<Page>>. En un diagrama de clases, una liga es una asociación entre una <<client page>> y ya sea otra <<client page>> o una <<server page>>. Una asociación de liga direcciona directo a la etiqueta ancla de HTML.

Icono: Ninguno.

Limitantes: Ninguna.

Valores etiquetados: Parameters – Una lista de nombres de parámetros que debe ser pasada junto con la requisición de la página ligada.

Nombre: Targeted Link

Clase de Metamodelo: Association

Descripción: Similar a una asociación <<link>>, un <<targeted link>> es una liga en donde la página asociada es mostrada en otro blanco. Esta asociación lleva directamente a la etiqueta ancla de HTML, con el blanco especificado por el atributo target de la etiqueta.

Icono: Ninguno

Limitantes: Ninguna

Valores etiquetados: Parameters – una lista de nombres de parámetros que debe ser pasada junto con la solicitud de la página ligada.

Nombre: Frame Content
Clase de Metamodelo: Association
Descripción: Una asociación de contenido de marco es una asociación de agregación que expresa el contenido de un marco de otra página o blanco. También puede señalar hacia otro frameset, indicando marcos anidados.
Icono: Ninguno
Limitantes: Ninguna
Valores etiquetados: Row – Un entero indicando el renglón específico del marco en el frameset de la página o blanco asociado a la que corresponde.

Nombre: Submit
Clase de Metamodelo: Association
Descripción: Una asociación <<submit> siempre es entre un <<form>> y un <<server page>>. Las formas envían los valores de sus campos al servidor a través de <<server pages>> para procesamiento. El servidor web procesa el <<server page>>, el cual acepta y usa la información en la forma enviada. Esta relación indica que página (o páginas) pueden procesar la forma, y sobre que de formas un <<server page>> tiene conocimiento.
Continua:
Continua:
Icono: Ninguno
Limitantes: Ninguna
Valores etiquetados: Parameters – una lista de nombres de parámetros que deben ser pasados junto con la solicitud de la página ligada.

Nombre: Builds
Clase de Metamodelo: Association
Descripción: La relación <<builds>> es una relación especial que une el espacio entre las páginas cliente y servidor. Las páginas servidor solo existen en el servidor y son usadas para construir páginas cliente. Esta asociación identifica que página servidor es responsable de la creación de una página cliente. Esta es una relación direccional, dado que la página cliente no contiene conocimiento alguno de cómo es que vino a existir. Una página servidor puede construir múltiples páginas cliente, pero una página cliente solo puede ser construida por una página servidor.
Icono: Ninguno
Limitantes: Ninguna
Valores etiquetados: Ninguno

Nombre: Redirect
Clase de Metamodelo: Association
Descripción: Una relación <<redirect>> es una asociación unidireccional con otra página web. Puede ser direccionada tanto para o hacia páginas cliente o servidor. Si la relación se origina de una <<server page>> entonces indica que el procesamiento de la solicitud de la página puede continuar con la otra página. Esto no indica que la página destino siempre participe en la construcción de la página cliente, solo que puede hacerlo. Esta relación en particular no es por completo estructural, dado que la invocación en si de la operación de redireccionamiento debe ser hecha programáticamente en el código de la página que la origina. Si la relación se origina de una <<client page>> entonces, esto indica que la página destino será automáticamente solicitada por el navegador, sin petición del usuario. Un valor de retraso de tiempo puede ser fijado para especificar un retraso (en segundos) antes de que la segunda página solicitada. Este uso de redireccionamiento corresponde a la etiqueta META y el valor equivalente en HTTP a “Refresh”.
Icono: Ninguno
Limitantes: Ninguna
Valores etiquetados: Delay – La cantidad de tiempo que debe esperar una página cliente antes de redireccionar a la página siguiente. Este valor corresponde a al atributo Content de la etiqueta META.

Nombre: Object
Clase de Metamodelo: Association
Descripción: Una asociación entre una página cliente y un objeto que esta integrado en ella. El objeto es por lo general un Java Applet o un control ActiveX. Esta asociación lleva en parte a la etiqueta <object> de HTML.
Icono: Ninguno
Limitantes: Ninguna
Valores etiquetados: Object Parameters – Una lista delimitada por punto y coma de nombres de parámetros y valores opcionales que deben ser ubicados en las etiquetas <param> de la definición de la etiqueta <object>.

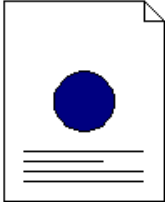
Nombre: IIOP
Clase de Metamodelo: Association
Descripción: IIOP (Internet Inter-Orb Protocol) es un tipo especial de relación entre objetos del lado del cliente y objetos del lado del servidor. Representa un mecanismo de comunicación diferente al HTTP para comunicaciones cliente-servidor. Típicamente, esta relación es entre Java Beans ene. Cliente y Enterprise Java Beans en el servidor.
Continua:
Continua:
Icono: Ninguno
Limitantes: Ninguna
Valores etiquetados: Ninguno

Nombre: RMI
Clase de Metamodelo: Association
Descripción: RMI (Remote Method Invocation) es una mecanismo para Java Applets y Beans para enviar mensajes a otros Java Beans en maquinas diferentes. Por lo general esta relación es entre Java Beans o Applets en el cliente y Enterprise Java Beans en el servidor.
Icono: Ninguno
Limitantes: Ninguna
Valores etiquetados: Ninguno

Nombre: Input Element
Clase de Metamodelo: Attribute
Descripción: Un elemento de entrada es un atributo de un objeto <<form>>. Conduce directamente a la etiqueta <input> en una forma HTML. Es usada para introducir una palabra o línea de texto. Los valores etiquetados asociados con este atributo estereotipado corresponden a los atributos de la etiqueta <tag>. Para completar los valores requeridos para la etiqueta de HTML, el nombre del atributo es usado como el nombre de la etiqueta <input> y el valor inicial del atributo es el valor de la etiqueta.
Icono: Ninguno
Limitantes: Ninguna
Valores etiquetados: Type – El tipo de control de entrada a ser usado que puede ser {Text, Number, Password, Checkbox, Radio, Submit, Reset}. Size – Especifica el tamaño del área a alojar de la pantalla, en caracteres. Maxlength – El numero máximo de caracteres que puede introducir el usuario.

Nombre: Select Element
Clase de Metamodelo: Attribute
Descripción: Un control de entrada usado en formas. El control seleccionado permite al usuario seleccionar uno o más artículos de una lista. La mayoría de los navegadores muestran este control como una caja de lista o combo.
Icono: Ninguno
Limitantes: Ninguna
Valores etiquetados: Size – Especifica cuantos artículos deben ser mostrados al mismo tiempo. Multiple – Un valor boleando que indica si múltiples artículos pueden ser seleccionados a la vez o no.

<u>Nombre:</u> Text Area Element
<u>Clase de Metamodelo:</u> Attribute
<u>Descripción:</u> Un control de entrada usado en formas, que permite entrada de múltiples líneas.
<u>Icono:</u> Ninguno
<u>Limitantes:</u> Ninguna
<u>Valores etiquetados:</u> Rows – 1 numero de líneas de texto visibles. Cols – El ancho visible del control en promedio al largo de caracteres.

<u>Nombre:</u> Page
<u>Clase de Metamodelo:</u> Component
<u>Descripción:</u> Un componente página es una página web. Puede ser solicitada por nombre desde un navegador. Una página puede contener o no scripts de cliente o servidor. Por lo general, los componentes página son archivos de texto accesibles por el servidor web, pero también pueden ser módulos compilados que son cargados e invocados por el servidor web.
Continúa:
Continúa:
Al final cuando son accesadas por un servidor web (ya sea como archivo o ejecutable) una página produce un documento con formato HTML que envía respuesta a la solicitud del navegador.

<u>Icono:</u>
<u>Limitantes:</u> Ninguna
<u>Valores etiquetados:</u> Path – La ruta requerida para especificar a la página web en el servidor web. Este valor debe ser relativo al directorio raíz de la aplicación web (del sitio).

Las reglas a seguir para obtener un sitio bien formado que plantea la extensión son las siguientes:

- Realización de componentes: Los componentes pueden realizar estereotipos de clases <<server page>>, <<client page>> o <<frameset>>.
- Generalización: Todos los elementos de modelación en una generalización deben ser del mismo estereotipo.
- Asociación: Las páginas cliente pueden tener a lo mas una relación <<builds>> con una página servidor, aunque una página servidor puede tener relaciones <<builds>> múltiples con páginas cliente diferentes.

Aparte de las combinaciones estándar de UML, las siguientes combinaciones son permitidas para cada estereotipo:

From: \ To:	Client Page	Server Page	Frameset	Target	Form

Client Page	Link Targeted Link Redirect	Link Targeted Link Redirect	Link Targeted Link Redirect	Dependency	Aggregation
Server Page	Builds Redirect	Redirect	Redirect Builds		
Frameset	Frame Content		Frame Content	Frame Content	
Target					
Form	Aggregated By	Submit			

Tabla 2.1.- Combinaciones de estereotipos.

3.2.- DEFINICIÓN DE ACTORES

Los actores que tendrán interacción con el sitio, son deducidos de las actividades que se realizarán en el sitio. En la figura 3.1 se muestran los usuarios que de alguna manera están involucrados con el sitio: el maestro del curso y los alumnos. Los alumnos pueden estar inscritos o no en la materia, pero tomando en cuenta que solo los alumnos inscritos deben tener acceso a los exámenes en línea y solo los maestros registrados deben poder modificar el banco de reactivos para los exámenes. Los usuarios no registrados son considerados como visitantes. Para efectos de la modelación a efectuar, vamos a considerar a los maestros y alumnos registrados, ya que con ellos se puede representar la funcionalidad total del sitio.

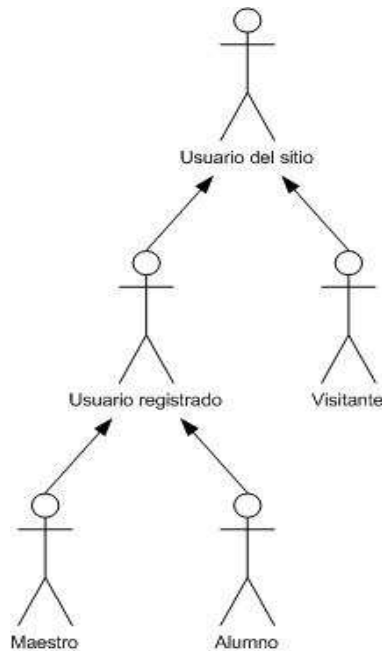


Figura 3.1.- Jerarquía de actores

3.3.- CASOS DE USO

Los casos de uso que se presentan a continuación tienen la intención de dar una perspectiva más clara de las actividades que un usuario podrá realizar dentro del sitio web de la materia de Computación II, para la cual tomaremos en cuenta tres casos de uso principales, correspondientes a las funciones de mostrar información, actualizar información y contestar exámenes en línea.

3.3.1.- Caso de uso: mostrar información

En estas funciones del sitio, el usuario selecciona algunas de las opciones del menú principal y el sitio muestra la información en la ventana correspondiente. La figura 3.2 ilustra este caso de uso de una manera simplificada.

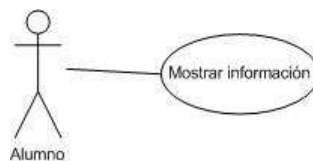


Figura 3.2.- Caso de uso “Mostrar información”

Las actividades que se realizan dentro del caso de uso de la figura 3.2, contiene varias actividades, las cuales pueden verse con detalle en el diagrama de la figura 3.3 mostrada a continuación:

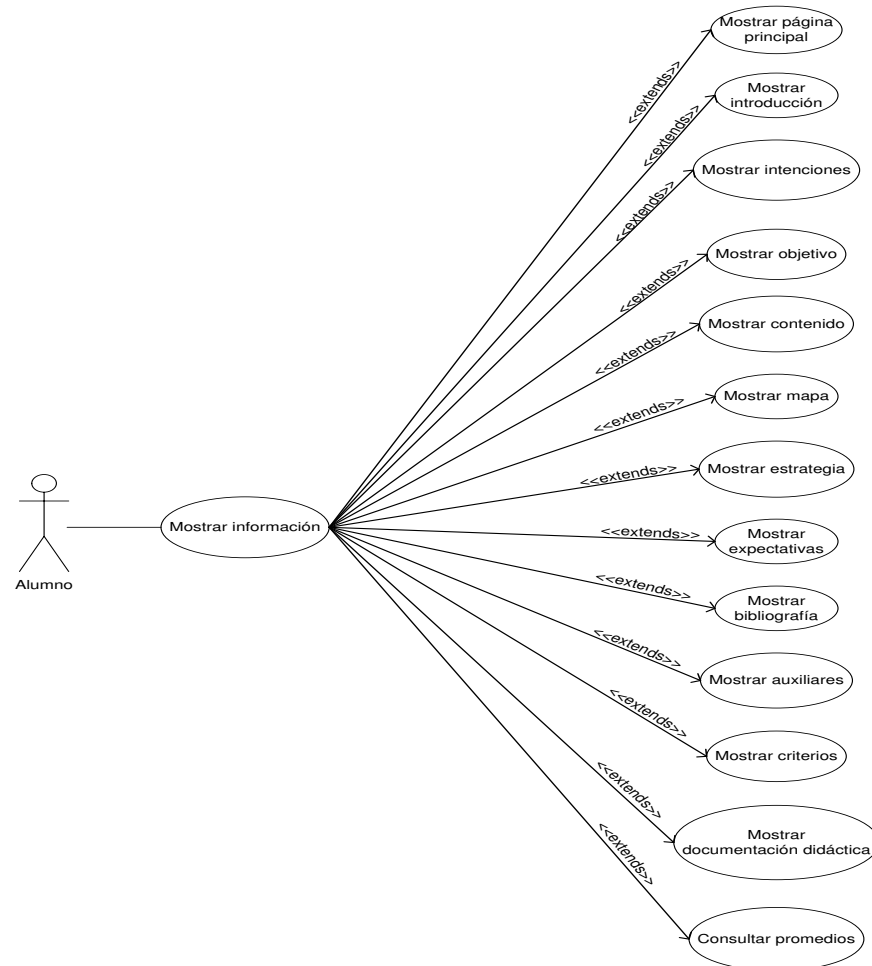


Figura 3.3.- Caso de uso “Mostrar información” detallado

3.3.1.1.- Caso de uso: Mostrar página principal

Actores: Alumno (iniciador), sitio web.

Propósito: Mostrar al alumno la página principal del sitio web del curso con el que desea interactuar.

Curso normal de eventos:

Acción del actor

1.- El alumno selecciona la opción “Principal” del menú principal.

Respuesta del sitio web

2.- El sitio despliega en la ventana correspondiente la página principal del sitio.

- 3.- El alumno ve la página principal y tiene la opción de hacer una nueva selección del menú.

3.3.1.2.- Caso de uso: Mostrar introducción

Actores: Alumno (iniciador), sitio web.

Propósito: Mostrar al alumno una breve introducción al curso al cual se desea tener acceso así como una cordial bienvenida a dicho curso.

Curso normal de eventos:

Acción del actor	Respuesta del sitio web
1.- El alumno selecciona la opción “Introducción” del menú principal.	2.- El sitio despliega en la ventana correspondiente la información necesaria para detallar un texto que permitirá introducir al usuario en el curso a interactuar.
3.- El alumno ve la información referente a la bienvenida e introducción y tiene la opción de hacer una nueva selección del menú.	

Dado que todos los casos de uso referentes a mostrar información tienen el mismo curso de eventos, basta con sólo explicar el propósito de cada uno de ellos.

3.3.1.3.- Caso de uso: Mostrar intenciones

Propósito: Mostrar al alumno mediante texto una breve explicación de las intenciones que el curso desea plasmar en el alumno.

3.3.1.4.- Caso de uso: Mostrar objetivo

Propósito: Mostrar al alumno los objetivos generales que el curso o materia desea obtener.

3.3.1.5.- Caso de uso: Mostrar contenido

Propósito: Mostrar al alumno el contenido temático del curso o materia a impartir.

3.3.1.6.- Caso de uso: Mostrar mapa

Propósito: Mostrar al alumno un mapa conceptual con la organización temática del curso así como también los prerrequisitos que el curso demandara.

3.3.1.7.- Caso de uso: Mostrar estrategia

Propósito: Mostrar al la estrategia de trabajo del curso virtual, esto es, la forma en que se va ha trabajar en el transcurso del estudio de el curso seleccionado.

3.3.1.8.- Caso de uso: Mostrar expectativas

Propósito: Mostrar al alumno una pagina web que explique las responsabilidades del curso tanto por parte del aluno, como por parte del profesor.

3.3.1.9.- Caso de uso: Mostrar bibliografía

Propósito: Mostrar al alumno una pagina web que liste las referencias bibliográficas a usar en el curso. Estas pueden ser bibliografía básica y/o adicional. También puede incluir una lista con los recursos disponibles en la red, por ejemplo, la biblioteca digital del ITESM.

3.3.1.10.- Caso de uso: Mostrar auxiliares

Propósito: Mostrar al usuario una pagina web que liste las herramientas auxiliares al curso disponibles para descarga por el alumno. Estas pueden ser tutoriales, manuales, compiladores, etc.

3.3.1.11.- Caso de uso: Mostrar criterios

Propósito: Mostrar al alumno una pagina web que explique el criterio de evaluación del curso, con los porcentajes correspondientes a exámenes, tareas, etc. También explica las políticas de evaluación correspondientes al curso, así como las del instituto.

3.3.1.12.- Caso de uso: Mostrar documentación didáctica

Propósito: Mostrar al alumno una pagina web que contenga información sobre el curso que pueda servir como punto de referencia para los maestros que impartan el curso, así como sugerencias de actividades para realizar en clase. También se puede incluir una reseña histórica del curso en la institución.

3.3.1.13.- Caso de uso: Consultar promedios

Propósito: Permitir al alumno consultar una tabla general de promedios, para que pueda observar su desempeño en el curso.

3.3.2.- Casos de uso: actualizar información

El sitio cuenta con unas funciones en las que el maestro puede actualizar la información con la que funciona el sitio, como lo es la actualización del banco de reactivos con que se generan los exámenes, el control de los alumnos que tienen acceso a los exámenes, el contenido teórico de los temas y la actualización de la lista de promedios.

3.3.2.1.- Caso de uso: Registrar maestro

Actores: Maestro (iniciador), Sitio web.

Propósito: Permitir al maestro del curso registrarse en el sitio web para descargar o actualizar la información que se muestra en el sitio.

Curso normal de eventos:

Acción del actor	Respuesta del sitio web
1.- El maestro selecciona la opción “Docentes” del menú principal.	
	2.- El sitio solicita el nombre de usuario y clave de acceso del maestro.
3.- El maestro ingresa la información solicitada.	
	4.- El sitio verifica que el maestro esté registrado.
	5.- Si el maestro está registrado, el sitio muestra en la ventana un listado con los directorios en donde se almacenan los archivos manipulables: banco de reactivos, registro de alumnos, tareas, auxiliares y tabla de promedios.

6.- El maestro se encuentra registrado y puede seleccionar una de las opciones ofrecidas.

Cursos alternos:

L5.- Si el maestro no se encuentra registrado, el sitio mostrará una página de error de registro. En caso de no contar con un registro, el maestro deberá solicitar al administrador web que lo registre (en caso de no ser él mismo).

3.3.2.2.- Caso de uso: Descargar archivos

Actores: Maestro (iniciador), Sitio web.

Propósito: Permitir al maestro del curso, una vez registrado en el sitio, descargar archivos con información del curso, ya sea bancos de reactivos, tabla de registro de alumnos, tareas, auxiliares y tabla de promedios.

Curso normal de eventos:

Acción del actor

Respuesta del sitio web

1.- El maestro, ya registrado, selecciona una de las opciones ofrecidas.

2.- El sitio muestra el contenido de los archivos en el directorio seleccionado, cada uno con una liga para descargar o actualizar.

3.- El maestro selecciona descargar un archivo.

4.- El sitio descarga el archivo a la computadora del maestro e informa que la descarga ha sido exitosa.

5.- El maestro puede hacer una nueva selección de las opciones.

3.3.2.3.- Caso de uso: Actualizar archivos

Actores: Maestro (iniciador), Sitio web.

Propósito: Permitir al maestro del curso, una vez registrado en el sitio, actualizar archivos con información del curso, ya sea bancos de reactivos, tabla de registro de alumnos, tareas, auxiliares y tabla de promedios, o en su defecto, subir nuevos archivos.

Curso normal de eventos:

Acción del actor

Respuesta del sitio web

1.- El maestro, ya registrado, selecciona una de las opciones ofrecidas.

2.- El sitio muestra el contenido de los archivos en el directorio seleccionado, cada uno con una liga para descargar o actualizar.

3.- El maestro selecciona actualizar un archivo.

- 4.- El sitio solicita al maestro la ruta de acceso en donde se encuentra el archivo que se desea incluir en el directorio.
- 5.- El maestro indica la ruta de acceso.
- 6.- El sitio graba el archivo en el servidor, dentro del directorio actual.
- 7.- El maestro puede hacer una nueva selección de las opciones.

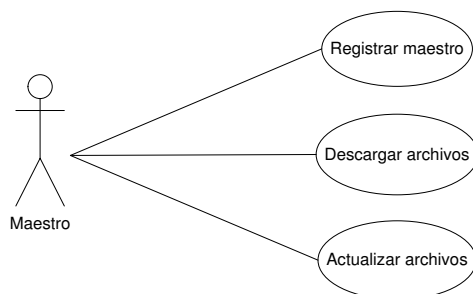


Figura 3.4.- Casos de uso de la sección “actualizar información”

3.3.3.- Casos de uso: Contestar exámenes en línea

El sitio debe contar con la capacidad de que los alumnos autorizados por el maestro del curso, puedan contestar exámenes en línea y ver los resultados de los mismos dentro del mismo sitio.

3.3.3.1.-Caso de uso: Registrar alumno

Actores: Alumno (iniciador), Sitio web.

Propósito: Permitir al alumno registrarse en el sitio para poder realizar un examen en línea, o en el caso de que el alumno ya ha contestado el examen, le informe el resultado del mismo.

Curso normal de eventos:

- | Acción del actor | Respuesta del sitio web |
|---|---|
| 1.- El alumno selecciona la opción “Exámenes” del menú principal. | |
| | 2.- El sitio solicita el nombre de usuario y clave de acceso del alumno. |
| 3.- El alumno ingresa la información solicitada. | |
| | 4.- El sitio verifica que el alumno esté registrado. |
| | 5.- Si el registro del alumno es exitoso, el sitio muestra en la ventana correspondiente un |

listado de ligas que llevan a los exámenes planeados para el curso.

- 6.- El alumno puede seleccionar de la lista el examen que desea contestar.

Cursos alternos:

- L5.- Si el alumno no se encuentra registrado, el sitio mostrará una página de error de registro. En caso de no contar con un registro, el alumno deberá solicitar al maestro del curso que lo registre.

3.3.3.2.-Caso de uso: Realizar examen

Actores: Alumno (iniciador), Sitio web.

Propósito: Permitir al alumno realizar un examen en línea y poder ver la calificación del examen contestado.

Curso normal de eventos:

Acción del actor	Respuesta del sitio web
1.- Una vez registrado, el alumno puede seleccionar la liga correspondiente al examen que desea contestar.	2.- El sitio verifica si el examen ya fue contestado o no. 3.- Si no ha sido contestado, el sitio genera un examen aleatorio, tomando de los reactivos correspondientes al examen solicitado.
4.- El alumno selecciona las respuestas de las opciones presentadas en el examen, contando con un límite de tiempo predefinido.	
5.- El alumno envía el examen contestado de regreso al sistema.	6.- El sitio califica el examen comparando las respuestas dadas por el alumno con las que están en la base de datos y registra una calificación en la tabla de promedios.

- 7.- El sitio muestra en una página nueva el resultado del examen.
- 9.- El alumno tiene la opción de hacer la solicitud de otro examen.

Cursos alternos:

L2.- Si el examen ya fue contestado, se pasa hasta L7.

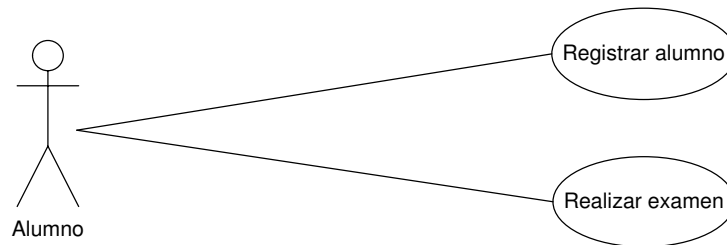


Figura 3.5.- Casos de uso de la sección “contestar exámenes en línea”

3.4.- NAVEGACIÓN DEL SITIO

Dentro de la planeación del proyecto, Franklin propone esquematizar la navegación básica del sitio tomando en cuenta las actividades planteadas por los casos de uso mediante un diagrama de componentes [5]. La organización básica del sitio se puede apreciar mediante el diagrama de la figura 3.6.

En el diagrama de navegación de la figura 3.6 se indica la página principal con el nombre de “Index”, el cual es el nombre que por lo general se les da a las páginas de inicio de un sitio. En la parte inferior izquierda de cada paquete se indica la región del sitio a la que corresponde, las áreas principales son “/”, que corresponde a la raíz del sitio; “/información/”, que corresponde a las páginas encargadas de mostrar información; “/docentes/”, que corresponde al área de actualización de información por el maestro y “/exámenes/” que corresponde al área en donde los alumnos solicitan la elaboración o los resultados de un examen.

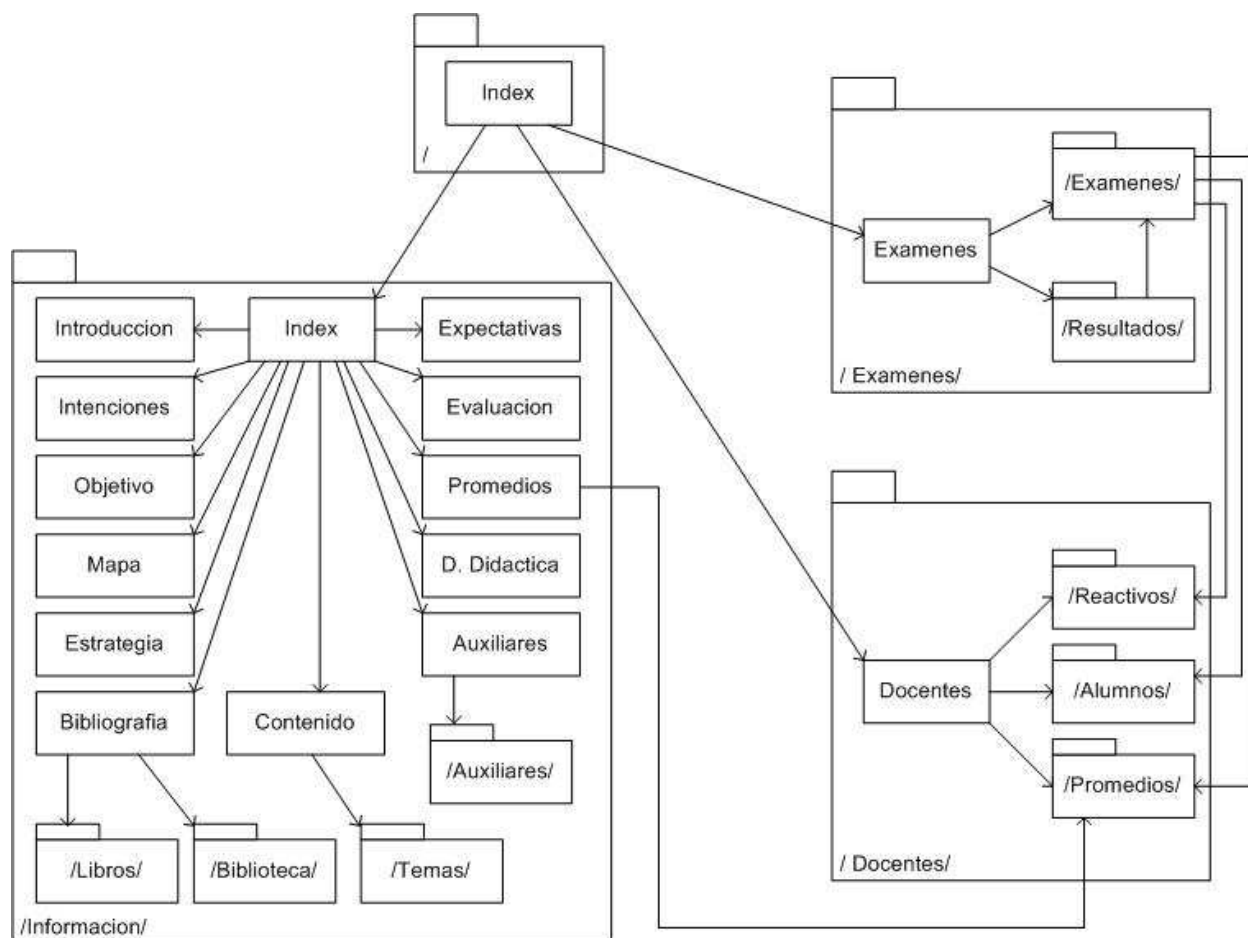


Figura 3.6: Organización básica del sitio

Dentro del paquete “/información/” se pueden observar otros tres paquetes, los cuales corresponden a las áreas de información detallada de libros, acceso a la biblioteca digital y la información contenida en los temas del curso. Dentro del paquete “/docentes/” se encuentran otros paquetes correspondientes a las áreas de “/reactivos/”, correspondiente a la actualización del banco de datos para los exámenes; “/alumnos/”, en donde se mantiene el control de los alumnos autorizados para presentar exámenes y “/promedios/”, que es donde se actualiza la información de las calificaciones obtenidas a lo largo del curso. En el paquete “/exámenes/” se encuentran otros dos paquetes mas: “/exámenes/” y “/resultados/”, el primero contiene lo referente a lo exámenes que los alumnos podrán contestar y el segundo lleva un registro de los exámenes contestados y de mostrar los resultados.

3.5.- DIAGRAMAS DE SECUENCIA

Una vez habiendo construido los casos de uso que muestran las actividades del sitio web de Computación II, podemos ver el flujo de acciones entre el sitio y los actores involucrados de una forma gráfica, esto se puede apreciar mediante los diagramas de secuencia.

Para representar las acciones de mostrar páginas con material informativo, basta con tomar el caso de uso de la figura 3.2 en la sección 3.3.1 y diagramar la interacción que el alumno tiene con el sistema, la cual se puede apreciar en la figura 3.7.

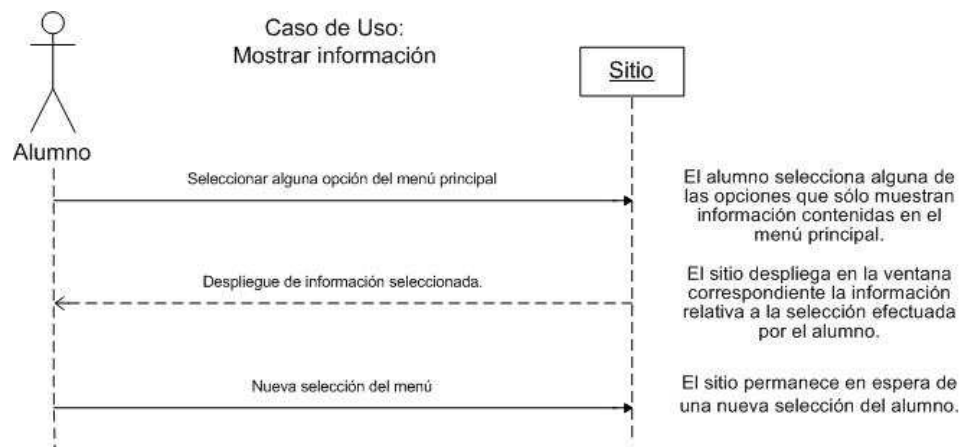


Figura 3.7: Diagrama de secuencia del sitio para el caso de uso “Mostrar información”

El diagrama de la figura 3.7 puede aplicarse a cada una de las actividades correspondientes a páginas que se limitan a mostrar información, indicadas en el diagrama 3.3 de la sección 3.3.1. Por ejemplo, la figura 3.8 muestra el diagrama de secuencia correspondiente al caso de uso “Mostrar página principal”.

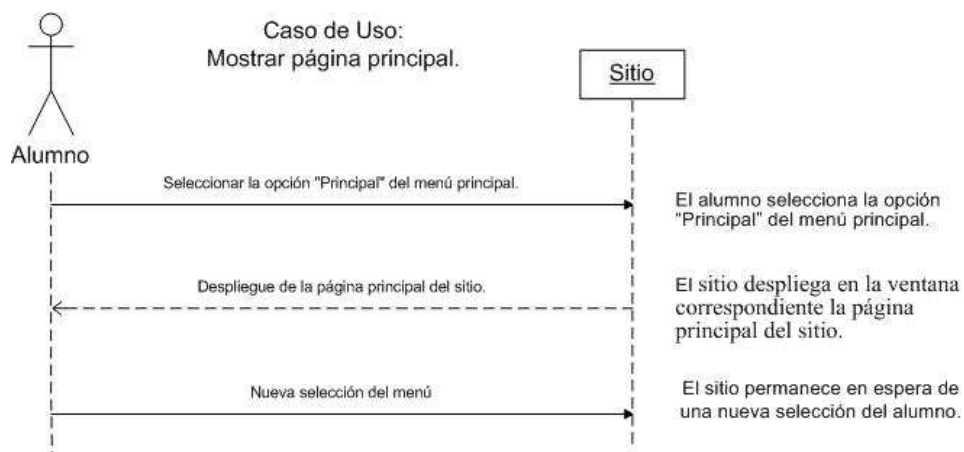


Figura 3.8: Diagrama de secuencia del sitio para el caso de uso “Mostrar página principal”

En lo referente a los casos de uso de la sección 3.3.2 se pueden observar tres acciones distintas: registrar al maestro, descargar archivos del servidor y grabar archivos en el servidor, ilustrados por las figuras 3.9, 3.10 y 3.11 respectivamente



Figura 3.9: Diagrama de secuencia del sitio para el caso de uso "Registrar maestro"



Figura 3.10: Diagrama de secuencia del sitio para el caso de uso "Descargar archivos"



Figura 3.11: Diagrama de secuencia del sitio para el caso de uso “Actualizar archivos”

Para los casos de uso de la sección 3.3.3, contestar exámenes en línea, se pueden considerar los siguientes diagramas de secuencia: registrar alumno (figura 3.12), y realizar examen (figura 3.13). En el caso de que el examen solicitado, ya ha sido contestado, las interacciones se ilustran en la figura 3.14.

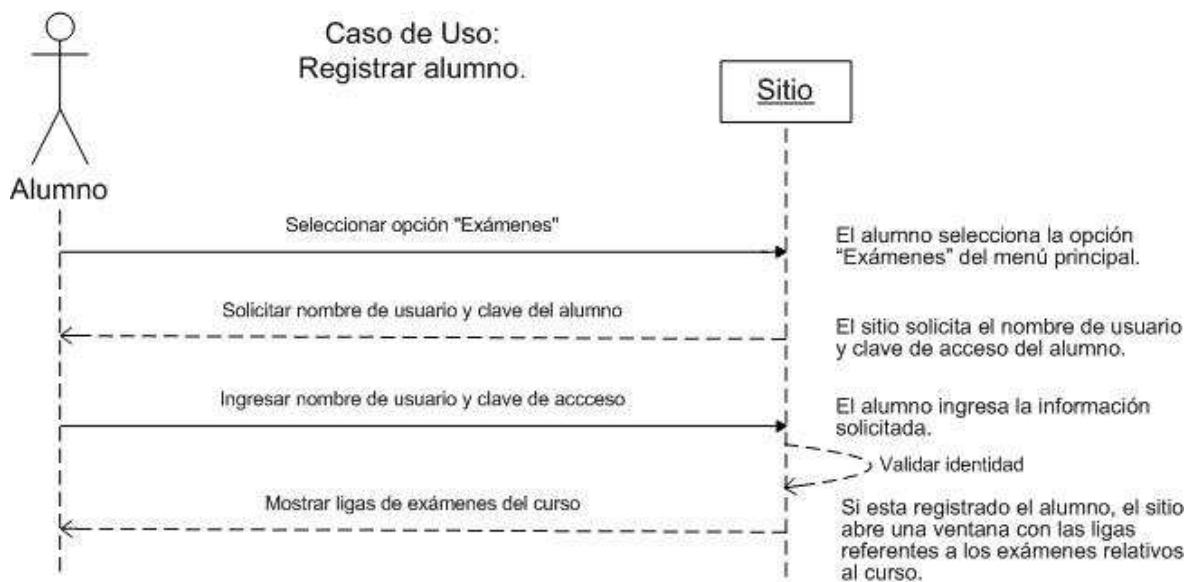


Figura 3.12: Diagrama de secuencia del sitio para el caso de uso “Registrar alumno”

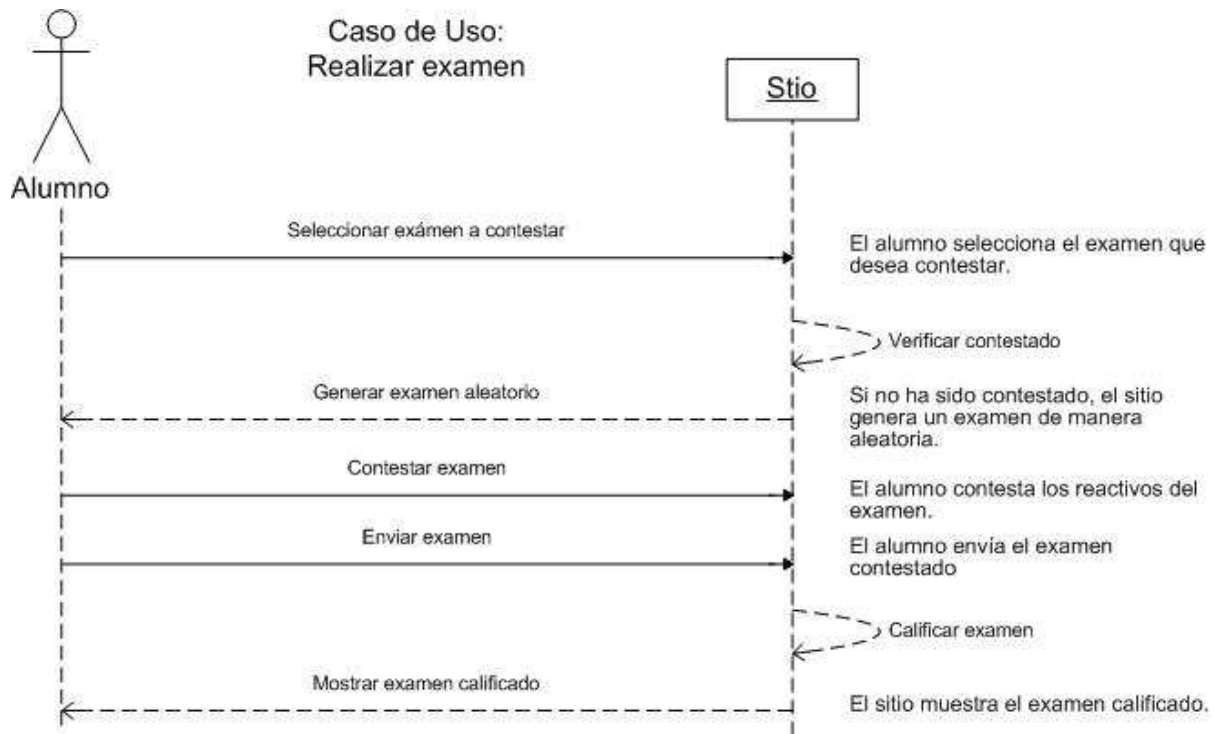


Figura 3.13: Diagrama de secuencia del sitio para el caso de uso “Realizar examen”

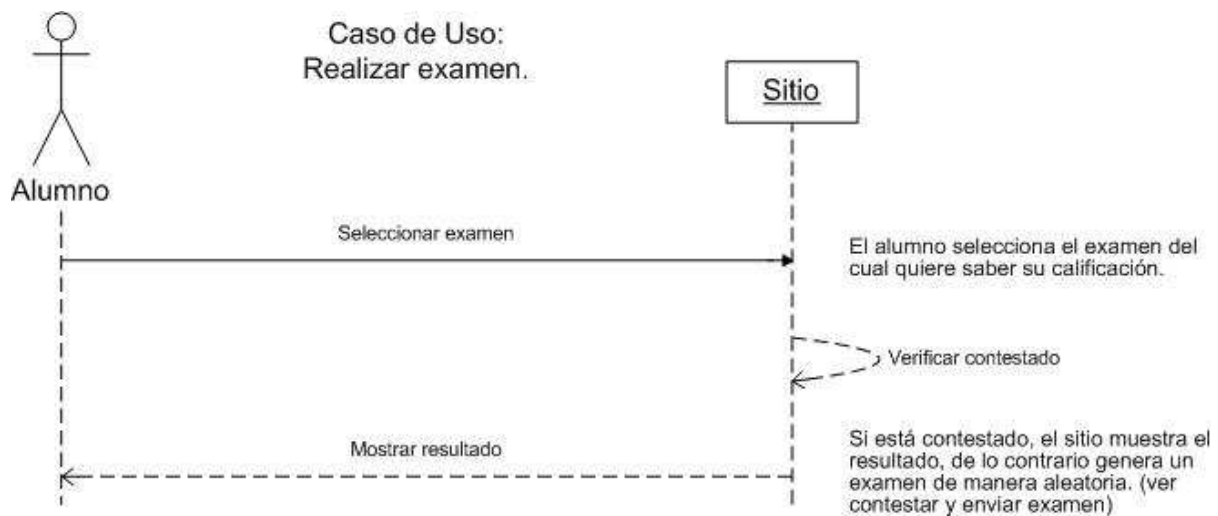


Figura 3.14: Diagrama de secuencia del sitio para el caso de uso “Realizar examen” curso alterno

Con los diagramas construidos en las secciones incluidas en este capítulo, se ilustra el análisis de las funciones del sitio y se pueden identificar su interacción con los usuarios, el siguiente capítulo trata sobre el diseño del sitio, incluyendo notación de la extensión para aplicaciones web de UML.

CAPITULO IV: DISEÑO

El diseño descrito en este capítulo muestra las clases necesarias para el funcionamiento del sitio y la forma en que interactúan entre ellas, se incluyen los patrones de contenido de las paginas principales y con base en ellos, se incluyen también diagramas de navegación del sitio y diagramas de funcionalidad, con ayuda de la extensión de UML para aplicaciones web descrita en el capítulo 2.

El diseño con patrones, cuando se va a trabajar con múltiples sitios, permite usar la funcionalidad de un sitio en otro igual o similar en cuanto a su estructura, organización y código reutilizable.

4.1.- DIAGRAMA DE CLASES

El diagrama de clases muestra la naturaleza estática del sitio, muestra la existencia de clases y sus relaciones en una visión lógica, su navegabilidad y multiplicidad [11]. En la figura 4.1 se muestra el diagrama de clases del sitio web “Computación II”

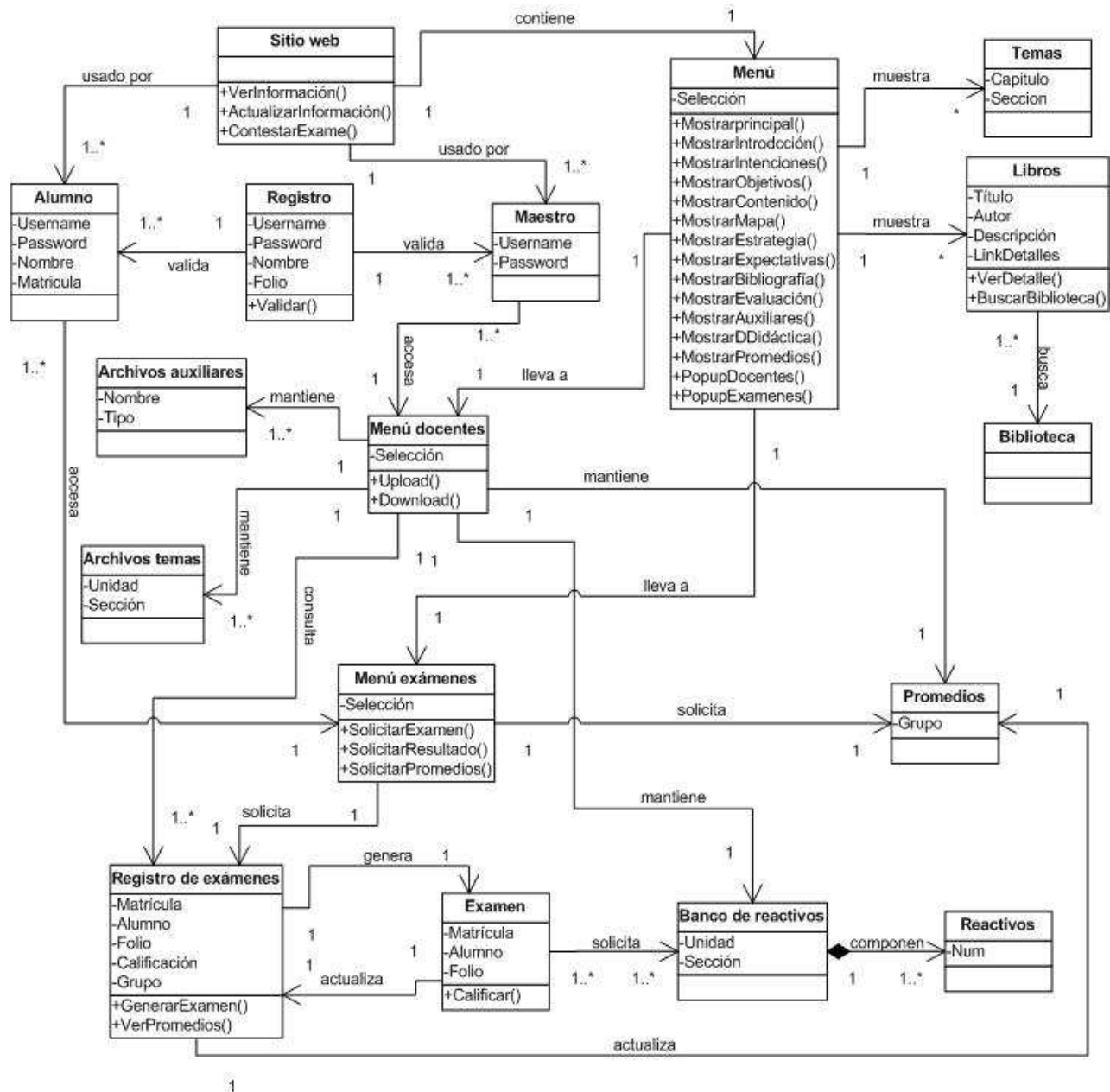


Figura 4.1.- Diagrama de clases

4.2.- PATRONES DE DISEÑO

Para poder visualizar de una manera más clara las funciones del sitio web y tomando en cuenta que la modelación basada en el sitio web de Computación II será utilizada para desarrollar también otros sitios educativos web similares, es importante contar con ciertos patrones de diseño que nos sirvan para modelar este tipo de sitios. Las paginas web presentan

cierta complejidad que no se puede capturar de una manera intuitiva con UML por si solo, por lo que se han ideado extensiones que permitan realizar la abstracción necesaria para visualizarlas.

Jim Conallen publicó en 1999 una extensión para UML, la cual permite de manera dinámica diseñar las operaciones que se llevan a cabo en un sitio web [4], dicha extensión se encuentra descrita dentro del marco teórico (capítulo 2).

Los patrones de las figuras 4.2, 4.3 y 4.4 representan la estructura general que se requiere en los sitios web educativos que se planean construir. La figura 4.2 indica el contenido general del sitio web educativo.

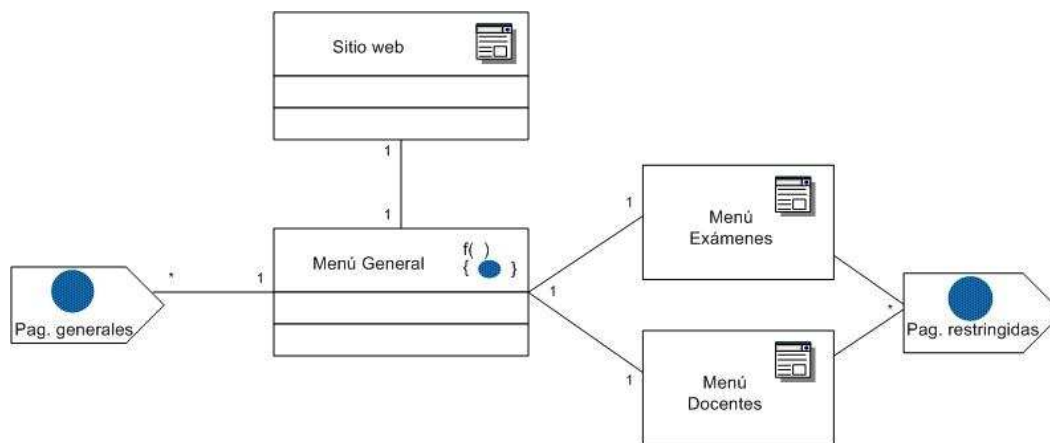


Figura 4.2.- Patrón del sitio web educativo

En la figura 4.3 se observa el patrón correspondiente del contenido que debe llevar la sección en donde solo pueden acceder los maestros del curso.

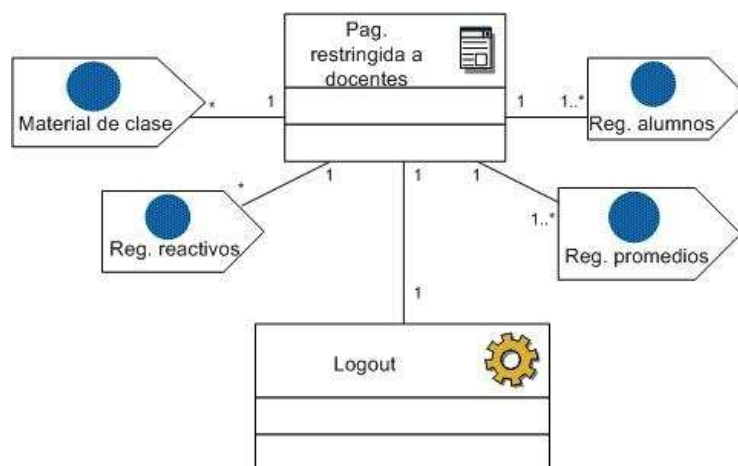


Figura 4.3.- Patrón de la sección de acceso de docentes

El patrón de la figura 4.4 permite ver lo que un alumno registrado puede acceder dentro del sitio web.

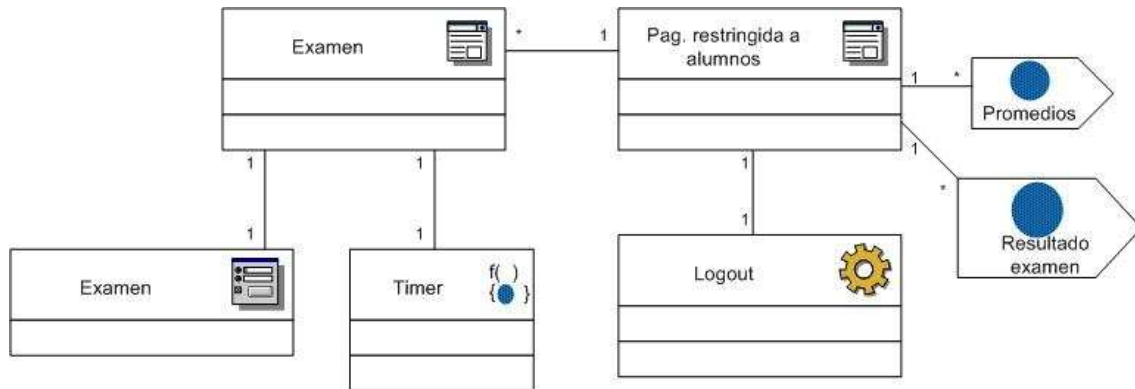


Figura 4.4.- Patrón de la sección de acceso de alumnos

Al aplicar los patrones de las figuras 4.2, 4.3 y 4.4 al sitio web de Computación II, se pueden generar los siguientes diagramas: La figura 4.5 indica cuáles son las ligas necesarias desde la página principal (Index), cabe mencionar que estas ligas son accesibles para todo usuario del sitio (registrado y no registrado).

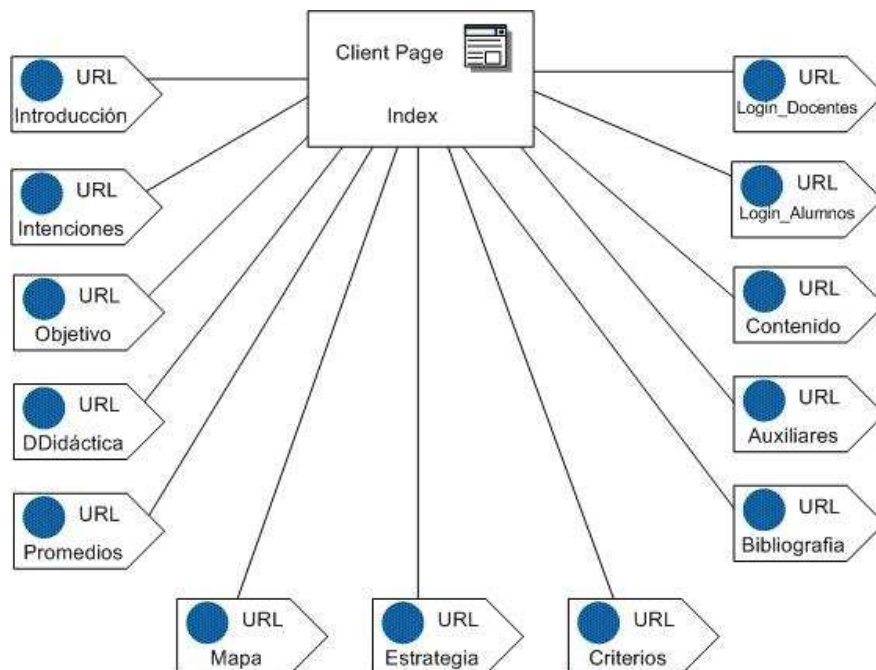


Figura 4.5.- Página inicial

Las páginas en donde se despliega la información referente a los temas del curso deben contener una o más ligas a las bibliografías usadas en el tema así como a archivos auxiliares, este tipo de paginas se pueden representar mediante el diagrama de la figura 4.6.

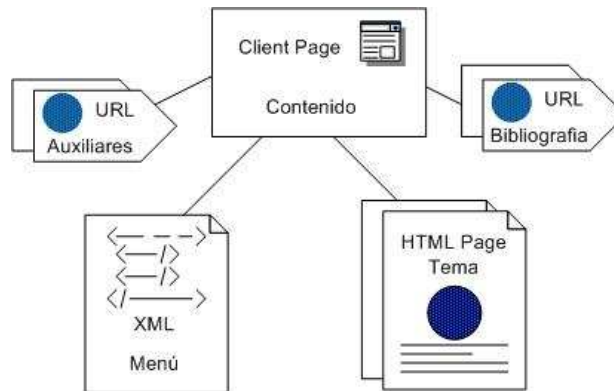


Figura 4.6.- Páginas de contenido

En cuanto al acceso de usuarios registrados, tenemos el caso del acceso de maestro(s) del curso. El diagrama 4.7 indica que el menú disponible para un maestro registrado se presenta en una ventana nueva del navegador, así como las ligas a los recursos que un maestro puede administrar.

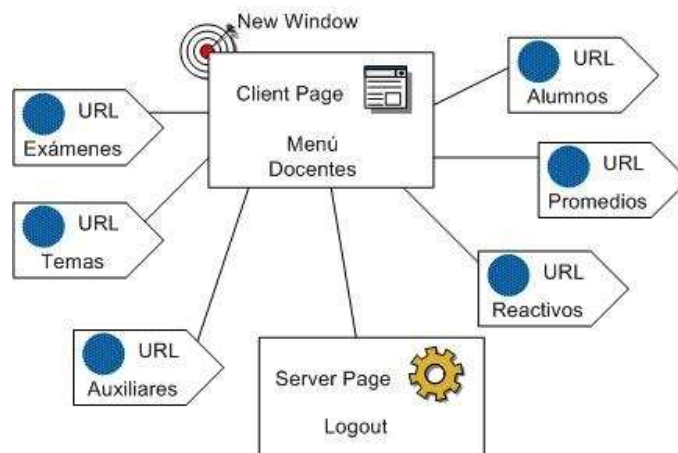


Figura 4.7.- Página de docentes registrados

Otro usuario registrado del sitio es el alumno. Un alumno tiene acceso a un menú que le permite solicitar un examen o revisar la tabla de promedios. La figura 4.8 muestra el menú disponible a los alumnos registrados.

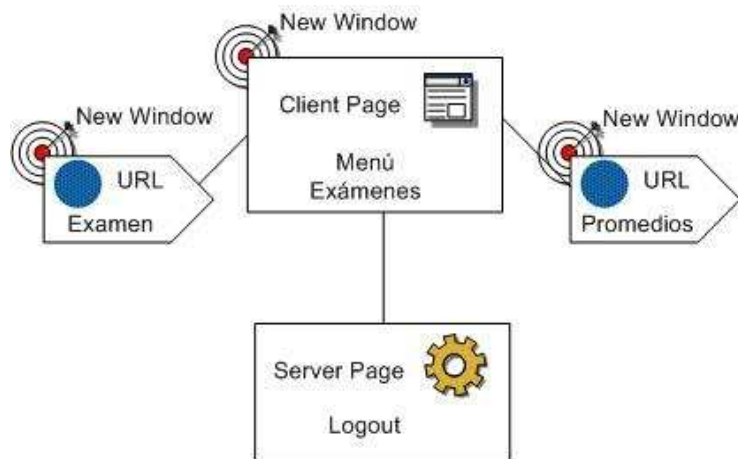


Figura 4.8.- Página de alumnos registrados

El diagrama de la figura 4.9 indica el contenido de dicha página. El examen es mostrado también en una ventana nueva e incluye una forma que contiene los reactivos a contestar y un reloj que limita el examen a un tiempo determinado.

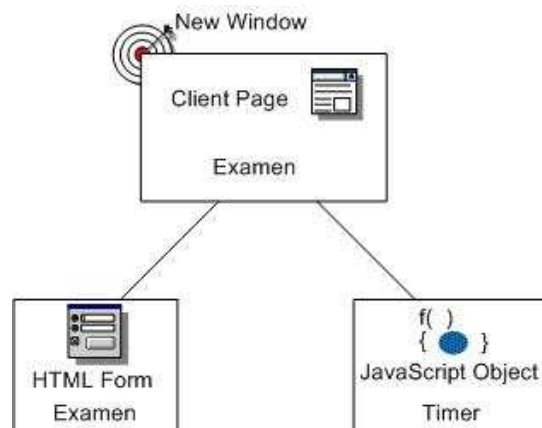


Figura 4.9.- Página de examen

4.3.- DIAGRAMAS DE NAVEGACIÓN

Para poder apreciar la navegación del sitio podemos nuevamente hacer uso de la notación propuesta por Conallen [3], que nos sirve para visualizar las operaciones que el usuario puede realizar y la respuesta que el sitio debe dar. Tomando como base, el diagrama de clases de la figura 4.1 y los diagramas de patrones de diseño, podemos construir el diagrama de navegación de la figura 4.10.

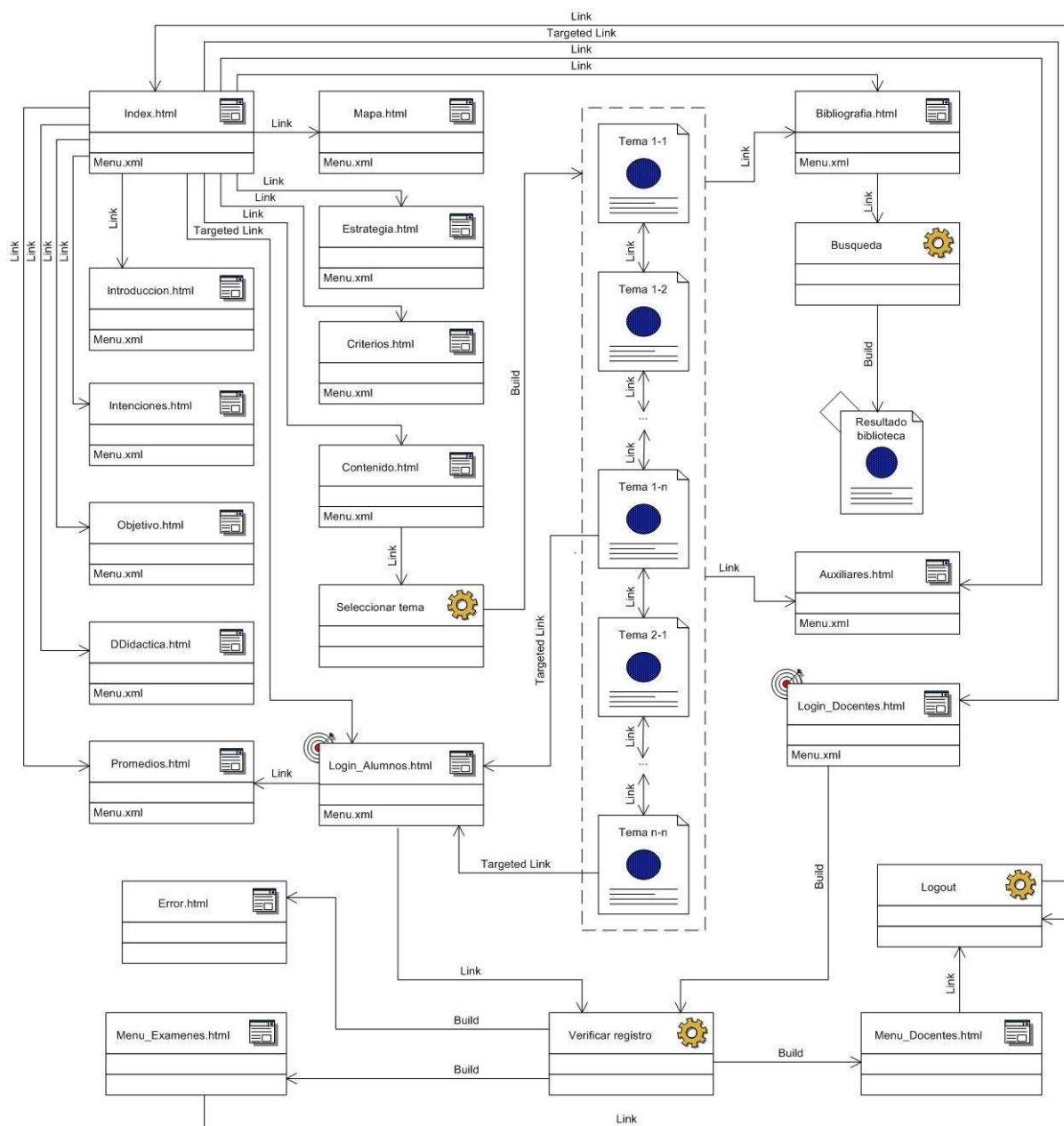


Figura 4.10.- Diagrama de navegación del sitio (principal)

En el diagrama mostrado en la figura 4.10 se pueden ver la navegación desde el menú principal, la cual incluye dos ligas a otras páginas para los usuarios registrados. La página `Menu_Examenes.html` es en donde los alumnos tienen acceso para presentar exámenes en línea y el diagrama de navegación de esa sección del sitio se puede ver en la figura 4.11. Por otro lado, la pagina `Menu_Docentes.html`, es donde los maestros pueden administrar el contenido del sitio y la navegación de esta sección se muestra en la figura 4.12.

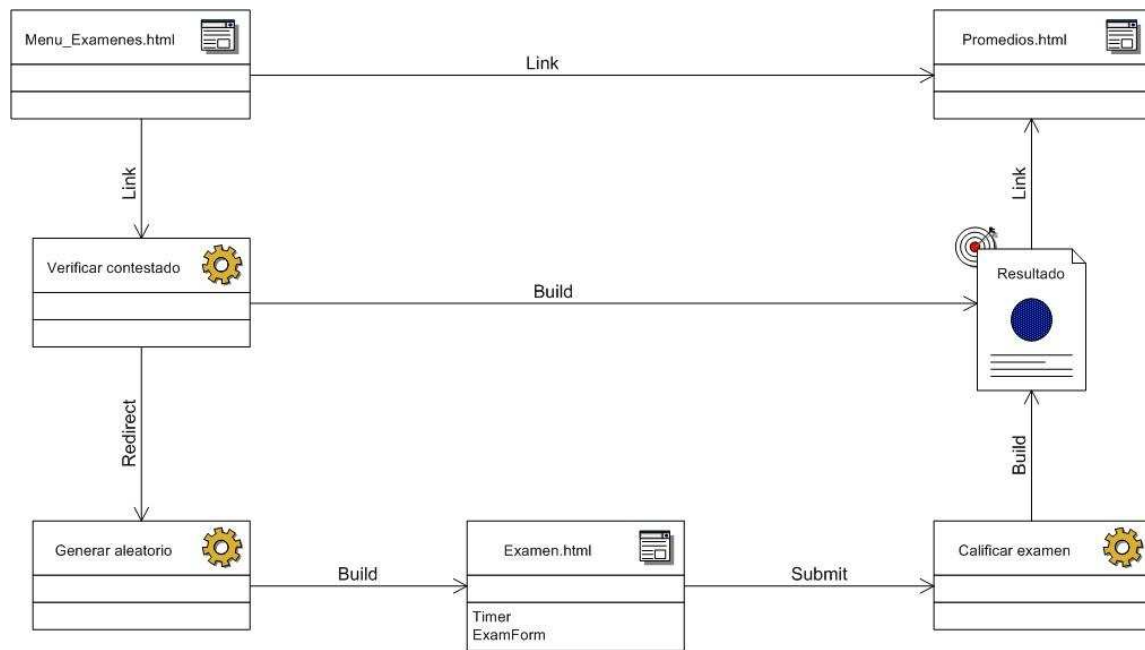


Figura 4.11.- Diagrama de navegación del sitio (alumnos)

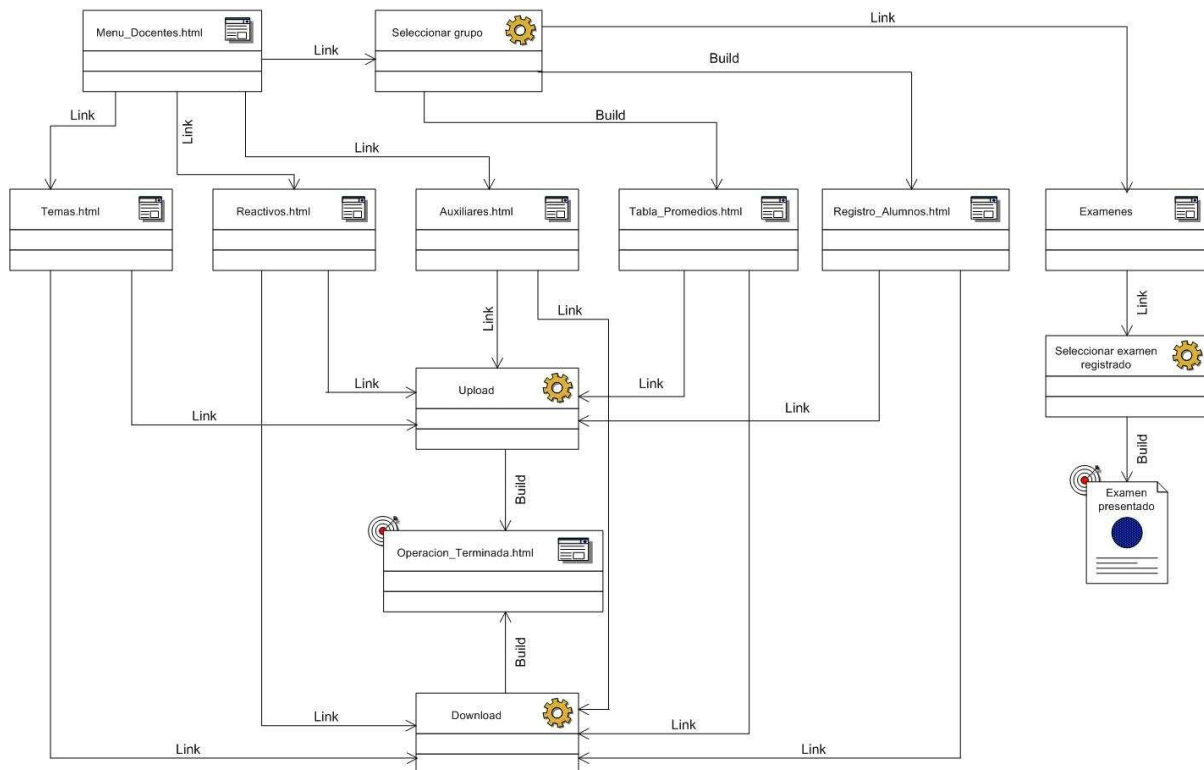


Figura 4.12.- Diagrama de navegación del sitio (maestros)

4.3.1.- DIAGRAMAS DE NAVEGACIÓN: FUNCIONALIDAD

Contando con los diagramas de clases y de navegación, podemos ver algunos detalles del funcionamiento del sitio mediante los siguientes diagramas de estereotipos web. Los diagramas de estereotipos web no permiten ver con detalle qué es ejecutado en el servidor y qué es ejecutado en el cliente así como qué interacción tiene con algún objeto del diagrama de clases [11].

Las tres funciones principales del sitio, como lo son señaladas por los casos de uso, son el poder ver información, realizar exámenes en línea (alumnos) y actualizar la información del sitio (maestros). Estas tres funciones pueden ser mejor comprendidas mediante sus diagramas de estereotipos web correspondientes, mostrados en las figuras 4.13, 4.14 y 4.15 respectivamente.

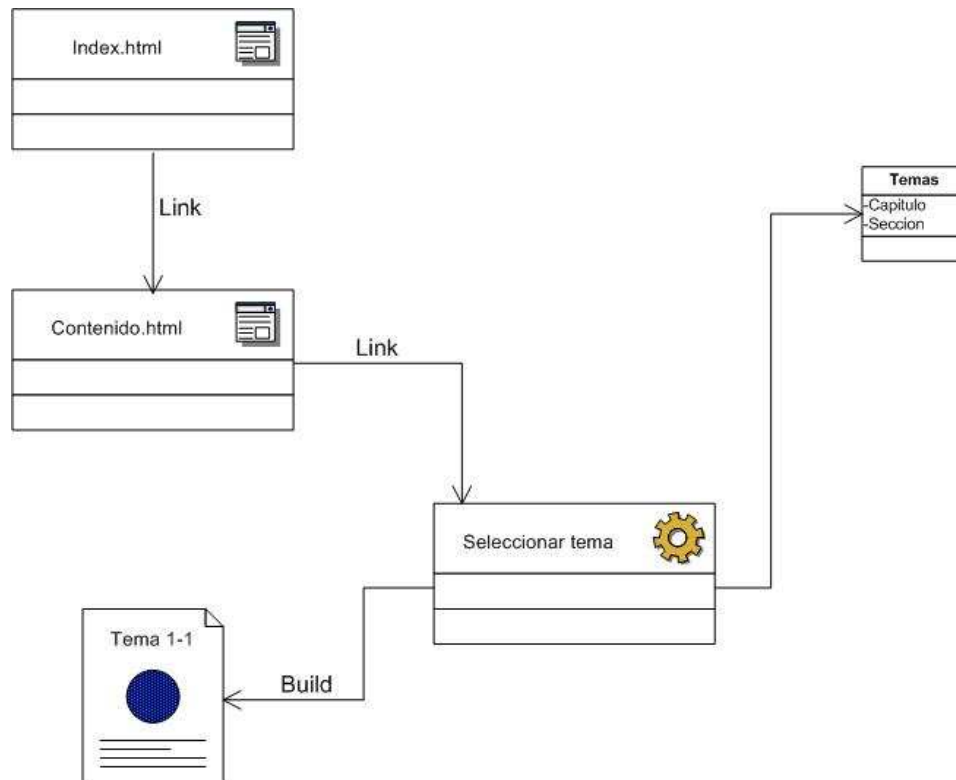


Figura 4.13.- Diagrama de funcionalidad: mostrar información

En el diagrama de la figura 4.13 se muestra la secuencia a seguir para ver la información de un tema contenido en el sitio. De la pagina Index.html se sigue la liga a la pagina Contenido.html, dicha pagina contiene ligas hacia cada uno de los temas del curso, al seleccionar

la liga correspondiente a un tema, el servidor se encarga de buscar el tema correspondiente de entre los temas disponibles para poder construir la pagina correspondiente.

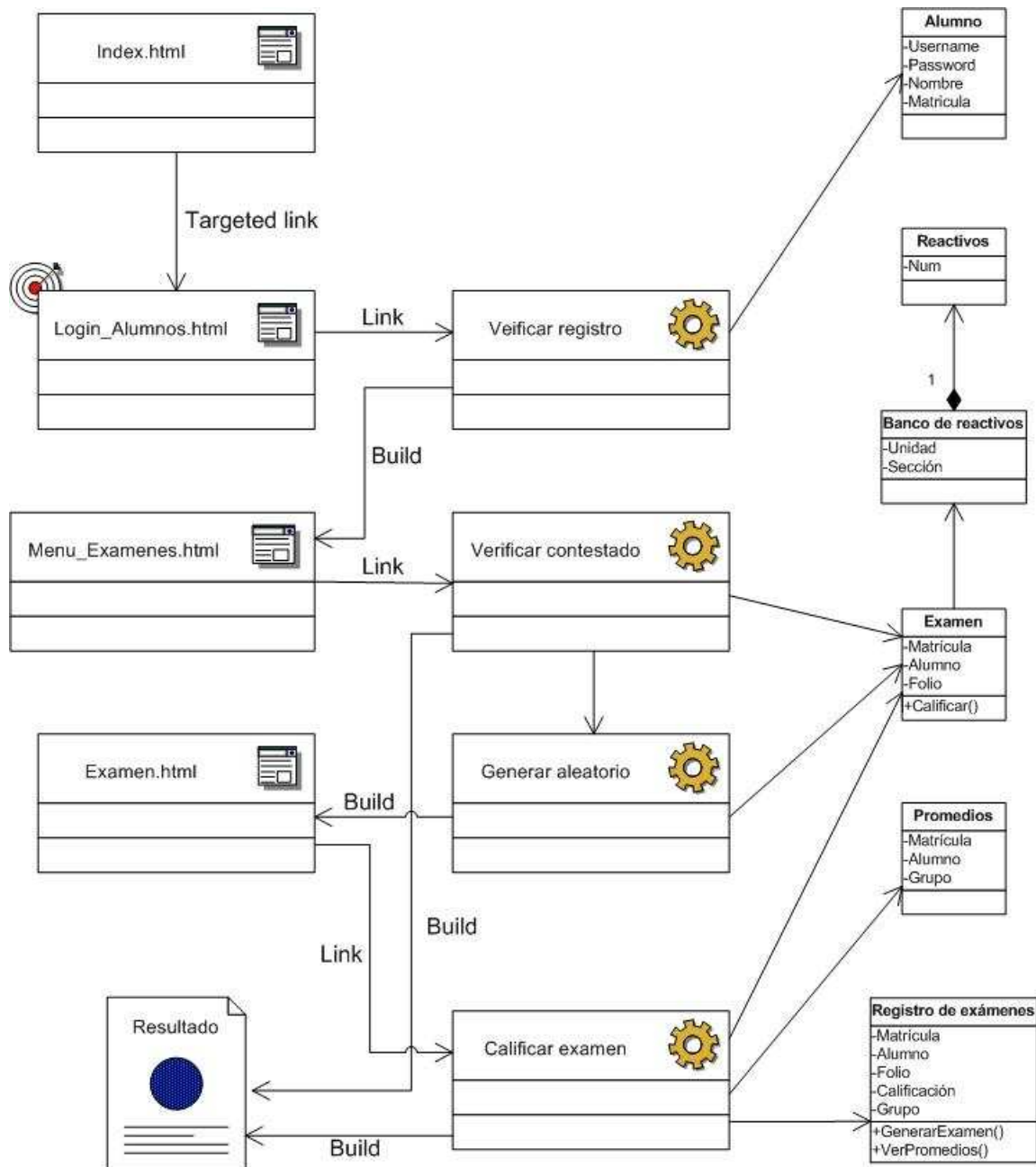


Figura 4.14.- Diagrama de funcionalidad: generar examen

En la figura 4.14 se indica la selección de la pagina Login_Alumnos.html, el cual es mostrado en una ventana nueva del navegador. Desde ahí el alumno se puede registrar ingresando su nombre de usuario y contraseña, datos que son verificados por el servidor. De ser

correctos, el servidor muestra la pagina Menu_Examenes.html, desde donde el alumno puede solicitar un examen en línea. Al seleccionar la liga de uno de los exámenes disponibles, el servidor verifica si el examen esta o no contestado, para mostrar el resultado o generar un examen aleatorio (según sea el caso). Después de presentar un examen, este es calificado, el resultado es registrado en la base de datos correspondiente y es mostrado al cliente del sitio.

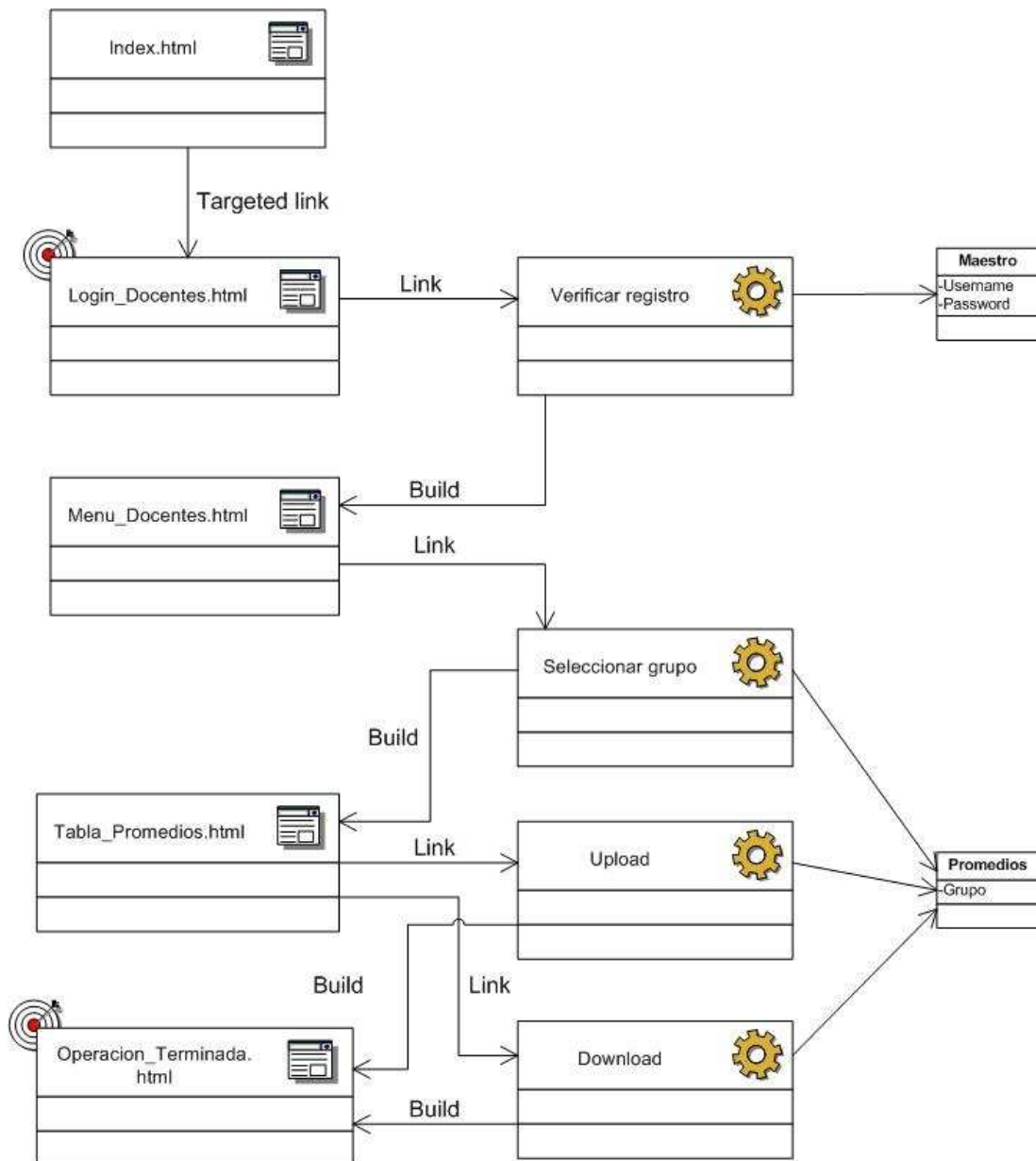


Figura 4.15.- Diagrama de funcionalidad: actualizar información

En la figura 4.15 se puede ver que cuando el maestro se registra en el sitio, el servidor lo conduce a administrar los archivos correspondientes a su grupo.

El diagrama muestra como ejemplo el manejo del archivo de la tabla de promedios, las operaciones a realizar con cualquiera de los archivos son descargar el archivo seleccionado o subir un archivo al servidor.

4.4.- DIAGRAMA DE COMPONENTES

Aunque este caso no sea muy complejo, el diagrama de componentes del sitio ilustrado en la figura 4.16 indica que se cuenta con un servidor web y servidores adicionales en donde se almacenan los archivos del sitio y los archivos de registro de usuarios.

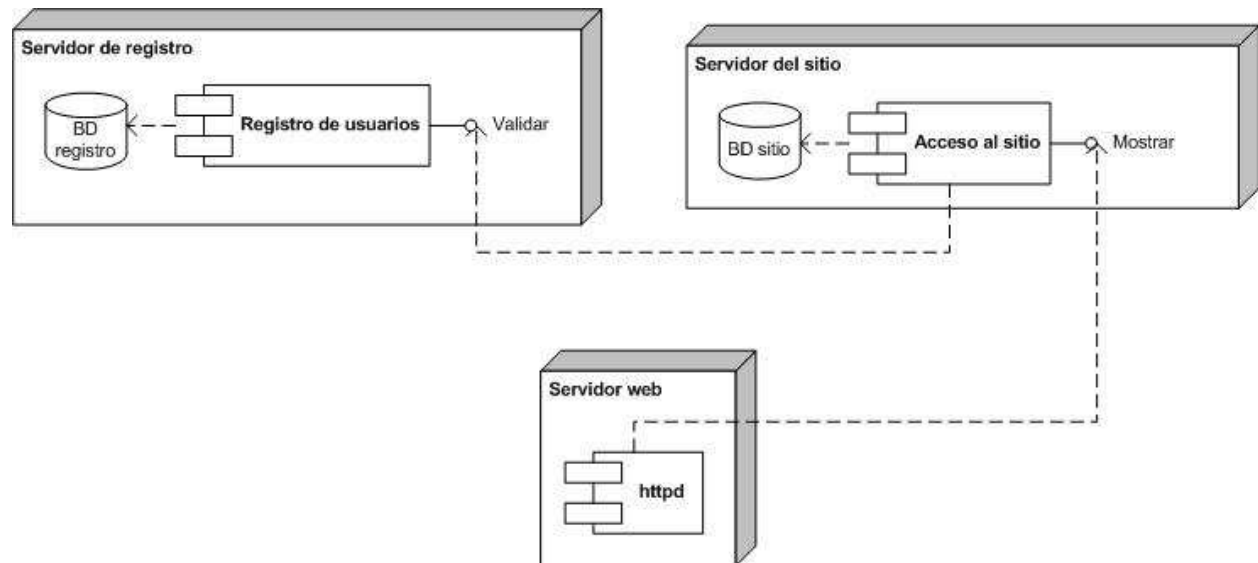


Figura 4.16.- Diagrama de componentes

Los diagramas contenidos en este capítulo componen un diseño general para los sitios web educativos que se desean implementar en algunos cursos del ITESM-CEM y en conjunto con el capítulo III componen un modelo rediseñado para la clase de Computación II, el cual, con ayuda de los patrones de diseño, servirá para el rediseño de los cursos adicionales que sean requeridos.

Referencias

- [1] Alhir, S., Unified Modeling Language extension mechanisms, <http://www.DistributedComputing.com>, 1998.
- [2] Booch, G., Rumbaugh J., Jacobson I., El lenguaje unificado de modelado, Addison Wesley Iberoamericana, 1999.
- [3] Conallen, J., Modeling web application design with UML, <http://www.conallen.org/whitepapers/webapps/ModelingWebApplications.htm>, 1999.
- [4] Conallen, J., UML extension for Web Applications 0.91, <http://www.conallen.org/technologyCorner/webextension/WebExtension091.htm>, 1999.
- [5] Franklin S., Planning your web site with UML, WebReview: cross-training for web teams, http://webreview.com/2001/05_18/developers/index01.shtml, 2001.
- [6] Fungairiño Martín Luis, Usabilidad de los sitios web. <http://www.funga.net/index.htm>, 2004.
- [7]
- [8] Larman, C., Applying UML and patterns: An introduction to object-oriented analysis and design and the unified process, 2nd edition, Pearson Education, 2001.
- [9] Muller Fabian, Introducción al desarrollo web para principiantes, <http://www.webexperto.com>, 2002.
- [10] Object Management Group, OMG Unified Modeling Language Specification, Version 1.5, <http://www.omg.org/>, 2003.
- [11] Quatrani, T., Introduction to the unified modeling language, Rational Developer Network, http://www.rational.com/media/uml/intro_rdn.pdf, 2001.
- [12] Rossi, G., Schwabe, D., Lyardet, F., Abstraction and reuse mechanisms in Web Application Models, 2000.
- [13] Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F. y Lorensen, W., Modelado y Diseño Orientado a Objetos, Prentice Hall, 1997.
- [14] Sholler, D., Web services need structure, <http://www.zdnet.com/filters/printerfriendly/0,6061,2881845-92,00.html>, 2002.