

Diagramas de navegación en aplicaciones Web

Diagrams navigation in Web applications

Juan Carlos Medina Martínez*

Lorena Alonso Guzmán**

Víctor Manuel Hernández Alarcón***

Claudia Mondragón Gómez****

Fecha de recepción: 3 de abril de 2013

Fecha de aprobación: 5 de mayo de 2013

Resumen

En general, las aplicaciones web son complejos sistemas de software, que en su desarrollo se requieren utilizar sólidas metodologías enfocadas en la ingeniería. El desarrollo de aplicaciones web se debe llevar a cabo con la elaboración de modelos acordes con su construcción.

Palabras clave: Aplicaciones web, casos de usos, diagramas, requerimientos, análisis, diseños, implementación

* Universidad Autónoma de Guerrero Unidad Académica de Ingeniería Av. Lázaro Cárdenas S/N, CU. Correo electrónico: jcmedina74@yahoo.com.mx

** Universidad Autónoma de Guerrero Unidad Académica de Ingeniería Av. Lázaro Cárdenas S/N, CU. Correo electrónico: vmanuel.hdez@gmail.com

*** Universidad Autónoma de Guerrero Unidad Académica de Ingeniería Av. Lázaro Cárdenas S/N, CU. Correo electrónico: alonso.lore@facebook.com

**** Universidad Autónoma de Guerrero Unidad Académica de Ingeniería Av. Lázaro Cárdenas S/N, CU. Correo electrónico: clau_mon6@hotmail.com

Abstract

In general, Web applications are complex software systems, which in its development requires focused methodologies used solid engineering. The development of Web applications should be carried out with the con chords modeling construction.

Key words: Web applications, use cases, diagrams, requirements, analysis, design, implementation.

1. Introducción

Un software se considera un producto triunfante cuando satisface las necesidades de las personas que lo utilizan, en cuanto se desempeña sin fallas durante un largo periodo, cuando es fácil de modificar e incluso más fácil de usar. Para cumplir con esto es indispensable que al desarrollar un software se aplique una adecuada ingeniería.

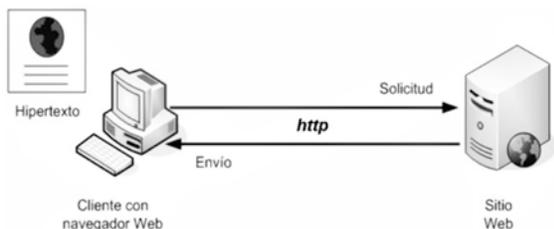
Las aplicaciones web son demasiado complejas que su desarrollo, construcción y mantenimiento han llevado a definir una nueva área dentro de la ingeniería de las ciencias computacionales: la llamada ingeniería Web (Kappel *et al.*, 2006). La ingeniería Web es una rama alterna a la ingeniería del software [2], pero que también muchos de sus preceptos se basan en conceptos definidos y demostrados por la ingeniería del software [3]. La ingeniería Web es un enfoque relacionado al desarrollo, construcción y mantenimiento de los sistemas basados en Web [2].

2. Aplicaciones web y su representación

2.1. La web

La web es un sistema gráfico de información de hipertexto e hipermedia, distribuido, global, interactivo, dinámico e independiente de la plataforma, que funciona en Internet. La Web hace uso del lenguaje de hipertexto conocido como HTML (Hypertext Markup Language, lenguaje marcador de hipertexto) que sirve para crear una red de documentos, los cuales están enlazados unos a otros y que se encuentran localizados alrededor del mundo. La Web es un nivel de infraestructura, es un espacio creado bajo ingeniería con la meta de especificar lenguajes y protocolos (World Wide Web, 2011, 18 de octubre). La figura 1 esquematiza de manera sencilla la idea fundamental de lo que es la Web.

Figura 1. Representa fundamental de la Web



Fuente: elaboración propia.

2.2. Las aplicaciones web

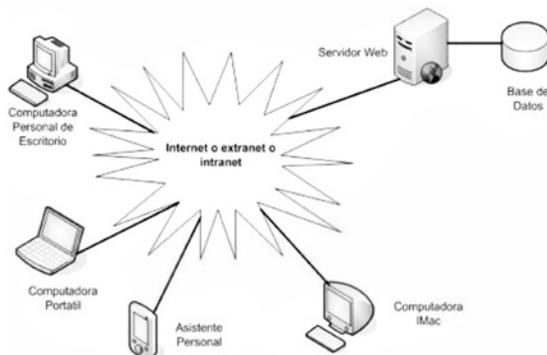
Una aplicación web es un sistema hipermedia en donde los recursos se encuentran vinculados unos a otros, por lo que debe verse como un sistema de nodos interconectados a través de vínculos. Estos vínculos proporcionan la forma para navegar entre los recursos de la aplicación. Muchos de los vínculos conectan a documentos textuales, pero el sistema puede ser utilizado para distribuir hipermedia y datos personalizados de igual forma [4].

Las aplicaciones web surgen para satisfacer dos necesidades de información dentro de las organizaciones [5]:

1. Debe de ser accesible desde cualquier lugar dentro de una organización e incluso desde el exterior.
2. Debe de compartirse entre todas las partes interesadas, con el fin de tener acceso a ella de manera completa o parcial en todo momento.

La figura 2 esquematiza la conexión de varios tipos de dispositivos a un servidor para la utilización de una aplicación Web.

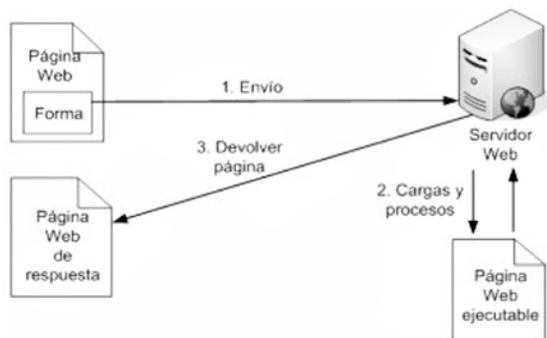
Figura 2. Representación de una conexión a una aplicación web



Fuente: elaboración propia.

La figura 3 muestra de manera clara la forma en cómo se lleva el procesamiento de datos en una aplicación Web. La aplicación Web es el software donde el usuario navega, introduce y obtiene datos, y que afecta el estado de la lógica del negocio. En términos muy sencillos, una aplicación Web es un sistema que permite a los usuarios ejecutar la lógica de negocios a través de un navegador Web [6].

Figura 3. Una aplicación web en ejecución



Fuente: elaboración propia.

2.3. La ingeniería web

La ingeniería web, según Pressman (2009), toma para sí muchos conceptos, principios y prácticas de la ingeniería del software. Además, acentúa actividades técnicas y administrativas similares. Los principios fundamentales de la ingeniería Web son los mismos que los de la ingeniería de software (Kappel *et al.*, 2006):

- a) La formulación clara de los objetivos y requisitos.
- b) El desarrollo sistemático de una aplicación web mediante modelos de proceso y fases.
- c) La planeación cuidadosa de las fases.
- d) La auditoria continúa de todo el proceso.

2.4. Características en el desarrollo de aplicaciones web

Es importante darse cuenta de que el desarrollo de las aplicaciones Web tiene ciertas características que lo hacen diferente del desarrollo del software tradicional. Las aplicaciones Web presentan las siguientes características de desarrollo descritas por [2]:

- La constante evolución en los requisitos y funcionalidades.
- Son intrínsecamente diferentes del software tradicional ya que el contenido incluye texto, gráficas, imágenes, audio y/o video integrados.
- Están hechos para ser utilizados por una muy variable comunidad de usuarios.
- Están conducidos por contenido: incluyen la creación y desarrollo del contenido.
- Demandan una buena apariencia y comportamiento, favoreciendo la creatividad visual y la incorporación de multimedia en la presentación y la interfaz.
- Se desarrollan de acuerdo a un calendario apretado y bajo presión de tiempos.

- Las repercusiones de una falla o la insatisfacción de los usuarios pueden ser bastante peor que los sistemas de aplicaciones convencionales.
- Son desarrollados por equipos pequeños de personas con diversos trasfondos, habilidades y conocimientos que son comparados a un equipo de desarrolladores de software multidisciplinario.
- Una rápida respuesta a los cambios constantes que existen en los avances de las tecnologías web y en el surgimiento de nuevos estándares que puedan ser utilizados.
- Su entrega es completamente diferente a la del software tradicional ya que se enfrentan a una variedad de dispositivos de despliegue, soporte de hardware, software y redes con una muy variada velocidad de acceso.
- La seguridad y privacidad son necesarios y demandan más que el software tradicional.
- La web ejemplifica un vínculo muy grande entre el arte y la ciencia que generalmente se encuentran en el desarrollo del software.

2.5. Instrumentos para el modelado

Desde la perspectiva de la ingeniería, un instrumento es un producto de trabajo o artefacto que se produce durante el tiempo que se lleva a cabo una acción del ciclo de vida o fase del software con el fin de facilitar la construcción de una aplicación Web, éstos pueden ser documentos o código de software o modelos [3][7].

Las herramientas CMS y RADD actuales que se utilizan para la rápida producción de aplicaciones Web generan códigos ejecutables o códigos fuentes que son en su mayoría complejas y confusas pero funcionales, y listas

para ponerse en marcha. Por lo que, la utilización de modelos con autoría de bajo nivel no serían indispensables al utilizar estos tipos de herramientas, lo más recomendable es hacer uso de modelos de autoría de alto nivel. El desarrollo de software al ser un área que incursiona dentro del campo de la ingeniería está forzosamente vinculado al desarrollo de modelos. En el desarrollo de aplicaciones web se requiere de una práctica de modelado que pocas veces es llevada a cabo bajo un perfil acorde a ellos. Los modelos no son para nada pérdida de tiempo, sino que estos representan los planos de la arquitectura que tendrá la aplicación web y son los cimientos importantes de cómo se va a construir.

En las fases de desarrollo de aplicaciones Web es necesario elaborar la documentación pertinente basada en estándares de calidad y tomando en cuenta las distintas fases que

conforman el desarrollo de la aplicación. Se propone utilizar una estructura para documentar los productos de trabajo que se elaboren durante el desarrollo de la aplicación Web. Dicha estructura es una adaptación basado en el estándar IEEE Std 830-1998 [7] (Grünbacher, 2006; Sommerville, 2011), en el estándar IEEE Std 1016-1998 y de las estructuras de documentación propuestas por [7] para los productos de trabajo que se elaboren en las fases de requisitos, análisis, diseño e implementación.

Los instrumentos para el modelado en esta investigación representan un sólido punto de partida para la implementación, ya que toman en cuenta las dimensiones de la arquitectura del modelado de aplicaciones web [10]. La tabla 1 agrupa a los instrumentos en cada una de las fases de la arquitectura y los aspectos del modelado a los que corresponden.

Tabla 1. Se Instrumentos propuestos para el modelado de aplicaciones web

Fase	Instrumento	Aspecto
Requisitos	Lista de especificaciones	_____
Análisis	Modelo de contenido	Estructural
	Modelo de casos de uso	Estructural y de comportamiento
	Modelo de presentación	Estructural
	Tabla de trazabilidad entre el modelo de casos de uso y requisitos funcionales	_____
	Tabla de trazabilidad entre los modelos de presentación y casos de uso	_____

Continúa

Fase	Instrumento	Aspecto
Diseño	Modelo lógico de la base de datos	Estructural
	Modelo de la estructura web	Estructural
	Modelo de navegación	Comportamiento
	Modelo de implantación	Físico
	Tabla de trazabilidad entre los modelos de navegación y casos de uso	_____
	Tabla de trazabilidad entre los modelos de navegación y de presentación	_____
Implementación	Modelo físico de la base de datos	Estructural
	Interfaces gráfica de la aplicación	Estructural

Fuente: elaboración propia.

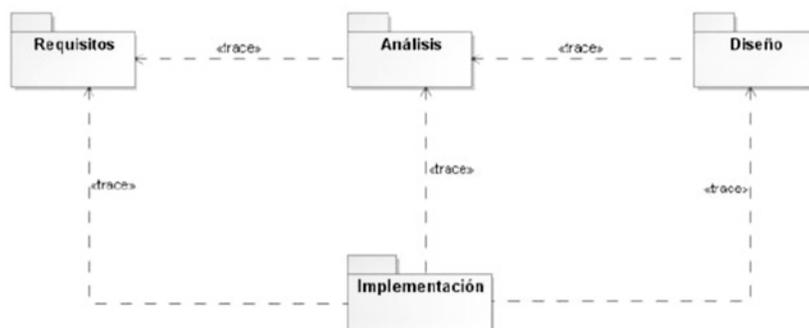
El conjunto de instrumentos que se proponen presentan las siguientes características:

1. Se basan en conceptos de orientación a objetos.
2. Presentan un perfil web.
3. Se orientan en las dimensiones de la arquitectura para las aplicaciones web.
4. Para su elaboración se puede utilizar herramientas CASE.
5. Son independientes de cualquier plataforma web de implantación.
6. Son independientes de la utilización de cualquier tipo de herramienta de tipo CMS y RADD.
7. Son independientes a cualquier proceso, por lo que permite la libertad de aplicar el enfoque que se considere el más adecuado.

2.5.1. Trazabilidad entre los instrumentos propuestos

Los productos de trabajo que se elaboran en el desarrollo de software deben seguir un principio de trazabilidad o rastreo, donde cada uno de los productos que se vayan construyendo en una de la fase se conecten de forma directa con otros productos de trabajo construidos en fases anteriores (Jacobson *et al.*, 2006). Por lo que cada fase debe tener una relación de traza como se muestra en la figura 3 representado con el estereotipo <<trace>>. El crear trazas es importante en el desarrollo de aplicaciones web por razones de comprensión y de propagación de cambios. Los instrumentos basados en tablas que se proponen permiten que se respete dicha trazabilidad entre las fases.

Figura 3. Trazabilidad entre las fases del desarrollo de aplicaciones Web



Fuente: elaboración propia.

2.6. Instrumentos de la fase de requisitos

Los instrumentos propuestos para la fase de requisitos son tres listas de especificaciones: de los actores del sistema, de requisitos funcionales y de requisitos no funcionales.

2.6.1. Especificación de los actores del sistema

Los actores son los usuarios que harán uso de la aplicación Web, por usuario se entiende cualquier cosa externa al sistema que interactúa con él. Un actor podría ser una persona, otro sistema de software, un dispositivo de hardware, etc. [12].

El instrumento de especificación de los actores del sistema consta de actor y función.

2.6.2. Especificación de los requisitos funcionales

Los requisitos funcionales son aquellos que representan las funcionalidades o capacidades que la aplicación Web deberá cumplir a los usuarios [9]. El instrumento de especificación

de los requisitos funcionales consta de clave, descripción del requisito funcional y actor implicado.

2.6.3. Especificación de los requisitos no funcionales

Los requisitos no funcionales son aquellos que especifican propiedades especiales que la aplicación Web deberá presentar, como restricciones en el entorno o en la implementación, rendimiento, dependencia de la plataforma, facilidad de mantenimiento, extensibilidad y fiabilidad. Los requisitos no funcionales son restricciones sobre la operación del sistema [7]. El instrumento de especificación de los requisitos funcionales consta de tipo, clave y descripción del requisito no funcional.

2.7. Instrumentos de la fase de análisis

2.7.1. Modelo de contenido

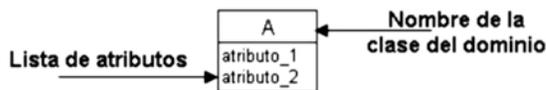
El modelo de contenido contiene elementos estructurales que proporcionan la visión de los requisitos de contenido para una apli-

cación web. Dichos elementos estructurales incluyen objetos de contenido (v. g., datos, texto, imágenes, fotografías, video y audio) que se presentarán en la aplicación [3].

2.7.1.1. Modelo de dominio

El diagrama del dominio se centra en conocer el dominio del problema para el que se va a desarrollar una aplicación Web. Un diagrama de dominio está conformado por un conjunto de clases de muy alto nivel de abstracción donde se representan los objetos del dominio del problema y sus relaciones (Fowler y Scott, 2003; Schwinger y Koch, 2006; Koch, 2007; Koch, 2010). EL diagrama de dominio puede fácilmente transformarse a un modelo de persistencia de datos.

Figura 4. Representación de una clase de dominio



Fuente: elaboración propia.

2.7.1.2. Diccionario de las clases de dominio

El diccionario de clases de dominio tiene como función describir el contenido de las clases definidos en el diagrama de dominio. En él se listan todos los elementos de datos que conforman a la clase de dominio para permitir su mejor comprensión [3].

2.7.2. Modelo de casos de uso

Un modelo de casos de uso es un modelo visual que se utiliza para modelar las interacciones entre la aplicación Web y los actores externos a él (Sommerville, 2011).

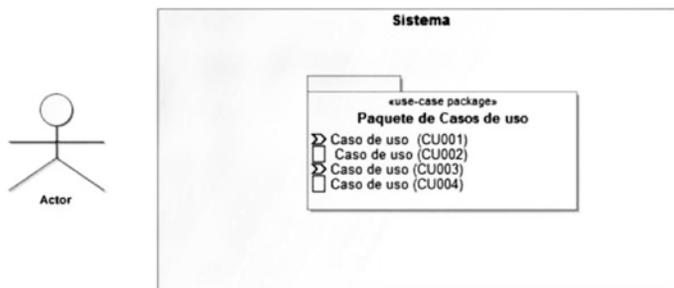
El modelo de casos es la primera vista arquitectónica de la aplicación Web donde se reflejan la estructura y los escenarios de alto nivel del sistema propuesto al cliente y los usuarios (Jacobson *et al.*, 2006).

2.7.2.1. Diagrama de empaquetamiento de casos de uso

En una aplicación web, los casos de uso deben de ser agrupados mediante el mecanismo de paquete proporcionado por UML, que es la forma de agrupar elementos relacionados. Los paquetes deberán tener el estereotipo <<useCasePackage>> para representar dicho mecanismo de agrupación. Los paquetes son estructuras jerárquicas que puede contener más paquetes además de casos de uso [13] La figura 5. 9 muestra la notación y el estereotipo de un paquete como instrumento para el empaquetamiento de casos de uso acompañado por un actor.

El diagrama debe de ir acompañado por el instrumento de especificación de empaquetamiento de casos de uso. Para especificar los paquetes se puede aplicar la plantilla que se muestra en la figura 5.

Figura 5. Instrumento de empaquetamiento de casos de uso



Fuente: elaboración propia

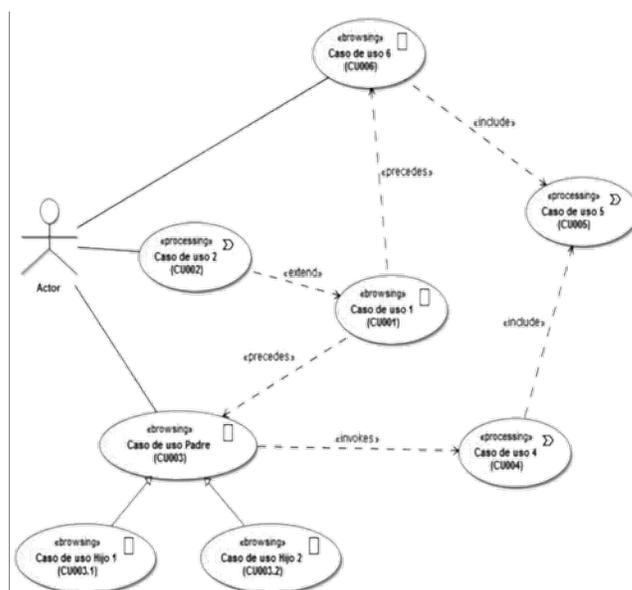
2.7.2.2. Diagrama de casos de uso

Un diagrama de casos de uso es una vista gráfica de uno o de todos los actores, casos de uso y sus interacciones identificados en la aplicación Web (Quatrani y Palistrant, 2006).

Los diagrama de casos de usos presentan un perfil Web, ese perfil se orienta por medio de estereotipos en los casos de uso y en las relaciones.

La figura 2.6 esquematiza un ejemplo abstracto utilizando un diagrama de casos de uso con perfil Web donde se utilizan cada una de los estereotipos indicados.

Figura 6. Diagrama de casos de uso con perfil web



Fuente: elaboración propia.

2.7.2.3. Especificaciones de los casos de uso

La especificación de un caso de uso permite conocer el escenario de la misma, es decir, el comportamiento del caso de uso [12].

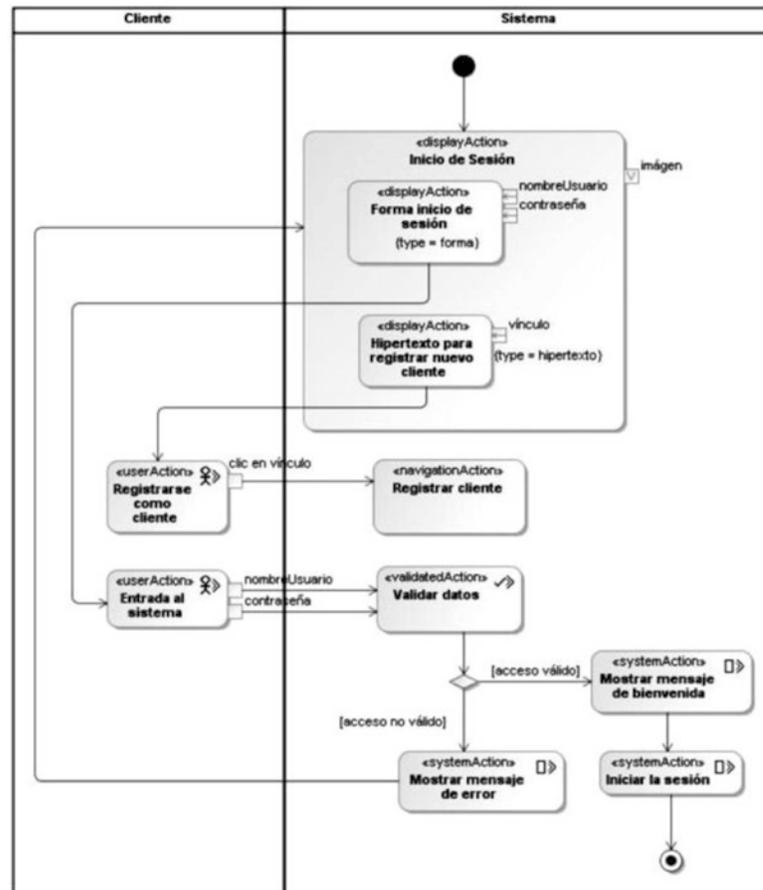
2.7.2.4. Diagrama de actividades

Un diagrama de actividades modela el flujo de proceso de un caso de uso. En conjunto con la especificación del caso de uso representan a nivel visual la dinámica la aplicación Web (Jacobson *et al.*, 2006 [12]).

Los diagramas de actividades con perfil Web representan las acciones que son parte de un caso de uso, así como los datos presentados al usuario y aquellos requeridos como entrada de datos pueden ser modelados con precisión como actividades [14][15].

Para su mejor comprensión en la utilización del estereotipado propuesto, la figura 7 esquematiza como un ejemplo de diagrama de actividades con perfil web para un caso de uso de Inicio de sesión.

Figura 7. Diagrama de actividades con perfil Web



Fuente: elaboración propia.

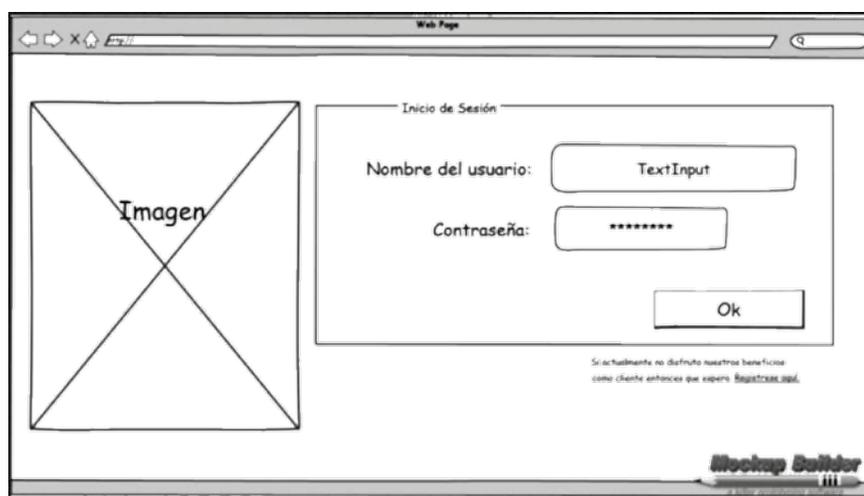
2.7.3. 2Modelo de presentación

La forma en que se presentará la información al usuario es un aspecto importante en una aplicación web, los esquemas de las interfaces del usuario permiten comprender y especificar las interacciones entre los actores y la aplicación durante la fase de análisis correlacionándolo con los requisitos capturados con anterioridad. Los esquemas de las interfaces del usuario, no sólo ayudan posteriormente a construir una adecuada interfaz gráfica en

la aplicación, sino que ayudan a comprender mejor los casos de uso y los requisitos [4] (Jacobson *et al.*, 2006).

Los esquemas de las interfaces del usuario son representaciones con características conceptuales que por medio de algunos trazos permiten definir plantillas que obliguen a mantener a las interfaces similares a lo largo de la aplicación [4]. La figura 8 representa al esquema de interfaz del usuario Inicio de sesión.

Figura 8. Esquema de interfaz del usuario



Fuente: elaboración propia.

2.8. Instrumentos de la fase de diseño

2.8.1. Modelo lógico de la base de datos

Una base de datos es una colección persistente de informaciones relacionadas que re-

presentan el dominio del problema. Una base de datos presenta tres niveles de modelo: el modelo conceptual, el modelo lógico y el modelo físico (Luque *et al.*, 2002). La tabla 2.2 representa el nivel de modelado, su descripción y los instrumentos que se utilizan.

Tabla 2. Niveles de modelado en las bases de datos

Nivel de modelado	Descripción	Instrumentos
Modelo conceptual	Descripción de los objetos del dominio del problema, sus atributos y de las relaciones entre los objetos.	<ul style="list-style-type: none"> • Modelo de contenido
Modelo lógico	Descripción de los objetos, así como de las relaciones existentes entre los objetos lógicos, identificando los atributos por los cuales pueden ser asemejados	<ul style="list-style-type: none"> • Modelo de contenido
Modelo físico	Descripción de los objetos físicos. La estructura y relaciones de los objetos son representados de forma adecuada para su posterior almacenamiento, recuperación y tratamiento.	<ul style="list-style-type: none"> • Esquema de la base de datos. • Diagrama físico de la base de datos. • Script de creación de la base de datos.

Fuente: elaboración propia.

2.8.2. Modelo de la estructura web

El modelo de la estructura web tiene como objetivo representar de forma visual la estructura lógica jerarquizada de la aplicación web mediante un diagrama estructural con perfil web que debe generarse en todo desarrollo para plasmar la organización de la información definida por medio de nodos [4] [17].

La tabla 3 representa la notación del diagrama estructural con perfil web, la cual se especifica por medio de nodos-tipo, que pueden ser simples o compuestos, y de dos mecanismos abstracta de relación: la generalización y la agregación. Además de incluir referencias a sistemas externos que no sean elementos de desarrollo [17].

Tabla 3. Notación para el diagrama estructural con perfil web

Notación	Nombre: función
	<p>Nodo compuesto: es un mecanismo de abstracción con el que se representa la estructura de hiperdocumento y, como tal, no siempre tiene una existencia física. Todo nodo compuesto debe ser origen de al menos una relación de generalización o de agregación destino. Serán uno o más nodos simples o compuestos.</p>
	<p>Nodo simple: es un contenedor de información que se trata como una unidad de recuperación que siempre tiene existencia física.</p>
	<p>Elemento externo: se utiliza para realizar referencias y sistemas externos que no son objeto de desarrollo.</p>
	<p>Relación de generalización.</p>
	<p>Relación de agregación.</p>

Fuente: elaboración propia.

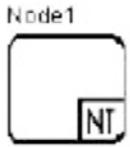
2.8.3. Modelo de navegación

El modelado de navegación incluye enlaces entre nodos o elementos externos, así como herramientas de navegación. Dicho modelo permite representar la navegación a páginas relacionadas a través de asociaciones o enlaces hipertextuales. Dichas asociaciones se etiquetan, pueden tener asociados atributos

y pueden ser unidireccionales o bidimensionales. Además, pueden incluirse relaciones n-arios con varios orígenes o destinos [17].

El modelo de navegación deberá importar los elementos definidos en el modelo estructural Web, pero agrega un elemento más en la notación presente en la tabla 4 [17].

Tabla 4. Notación para el diagrama estructural con perfil web

Notación	Nombre: función
	<p>Herramienta de navegación: estas herramientas pueden corresponder con nodos que aparecen en el modelo de la estructura web y que en el diagrama de navegación indican sus características de navegación. Por lo que, muchas agregaciones (y de herencia) suelen convertirse en una herramienta de navegación que permita acceder a los elementos que componen un nodo compuesto. También pueden ser nuevos nodos que hay que incluir porque solo tienen como objeto facilitar la navegación.</p>
	<p>Enlace hipertextual o asociación</p>

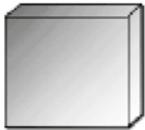
Fuente: elaboración propia.

2.8.4. Modelo de implantación

El modelo de implantación es representado mediante un diagrama de despliegue que es la encargada de definir la arquitectura física de la aplicación web por medio de no-

dos interconectados y que en el interior de cada nodo presentan la distribución de componentes que servirán para el funcionamiento del sistema [18]. La tabla 5 representa la notación de un diagrama de despliegue en UML.

Tabla 5. Notación para el modelo de despliegue

Notación	Nombre: función
	<p>Nodo: es un elemento físico que existe en tiempo de ejecución y representa un recurso computacional que generalmente tiene alguna memoria y, a menudo, capacidad de procesamiento.</p> <p>Los nodos sirven para modelar la topología del hardware sobre el que se ejecuta el sistema. Un nodo representa normalmente un procesador o un dispositivo sobre el que se pueden desplegar los componentes.</p>

	<p>Componente: un componente de software es un elemento de la aplicación web que ofrece un conjunto de servicios o funcionalidades, a través de interfaces definidas, y que si es necesario puede remplazarse.</p>
	<p>Asociación: representa una ruta de comunicación entre los nodos</p>

Fuente: elaboración propia.

2.9. Instrumentos de la fase de implantación

Los instrumentos propuestos para la fase de implementación son: el modelo físico de la base de datos y las interfaces gráficas de la aplicación.

2.9.1. Interfaces gráficas de la aplicación

La interfaz gráfica del usuario es la parte de la aplicación web ya terminada que permite al usuario interactuar con él. Las interfaces gráficas de usuario ofrecen al usuario ventanas, cuadros de diálogo, barras de herramientas, botones, listas desplegables y muchos otros elementos. Estas interfaces gráficas están relacionadas con los esquemas de las interfaces del usuario, es decir, las interfaces gráficas del usuario son la implementación del modelo de presentación obtenido en la fase de análisis [4] (Jacobson *et al.*, 2006).

3. Conclusión

Los instrumentos que se presentan se centran en las tres dimensiones de la arquitectura de modelado propuesta por la ingeniería

web: fases, aspectos y niveles. Los instrumentos son: listas de especificaciones, distintos tipos de modelos como lo son: modelo de contenido, modelo de casos de uso, modelo de presentación, modelos lógico y físicos de la base de datos, modelo de la estructura web, modelo de navegación, modelo de implementación y las interfaces gráficas de la aplicación; además de un conjunto de tablas para el rastreo de los instrumentos elaborados: tabla de trazabilidad entre el modelo de casos de uso y requisitos funcionales, tabla de trazabilidad entre los modelos de presentación y casos de uso, tabla de trazabilidad entre los modelos de navegación y casos de uso, y tabla de trazabilidad entre los modelos de navegación y de presentación.

El caso de estudio permite demostrar que los instrumentos de modelado pueden ser aplicados en problemas reales de desarrollo de aplicaciones web. Elaborándolos bajo un enfoque de proceso de desarrollo, aplicando estándares y ubicando su elaboración en las fases de requisitos, análisis y diseño se obtienen modelos que representan los aspectos estructurales y de comportamiento de la aplicación web que se desarrollará.

4. Referencias

- [1] Kappel et al., 2009
- [2] S. Murugesan y A. Ginige, "Web engineering: Introduction and perspectives". En W. Suh (Ed.). *Web engineering: principles and techniques*. pp. 1 - 32. United Kingdom: Idea Group Publishing. 2005.
- [3] R. S. Pressman, *Ingeniería del software: Un enfoque práctico*. Sexta edición. México: McGraw-Hill/interamericana Editores. 2009.
- [4] I. Aedo, P. Díaz, S. Montero y M. Castro, *El desarrollo hipermedia y web como proceso de ingeniería*. En M. P. Díaz, S. Montero y I. Aedo (Coords.). *Ingeniería de la Web y patrones de diseño*. España: Pearson Educación. 2005.
- [5] J. S. Castejon Garrido, "Arquitectura y diseño de sistemas Web modernos", *InforMAS, Revista de Ingeniería Informática del CIIRM*, num. 1, diciembre, 2004. [En línea] disponible en http://www.cii-murcia.es/informas/ene05/articulos/Arquitectura_y_diseño_de_sistemas_web_modernos.pdf. [30 de octubre de 2011]
- [6] L. Shklar y R. Rosen, *Web application architecture: principles, protocols and practice*. Great Britain: John Wiley & Sons, Ltd. 2003
- [7] B. Bruegge y A. H. Dutoit, *Ingeniería de software orientado a objetos*. México: Prentice-Hall. 2002.
- [8] Grünbacher, 2006;
- [9] Sommerville, 2011)
- [10] W. Schwinger y N. Koch, *Modeling Web Applications*. En Kappel, Gerti Pröll, Birgit, Reich Siegfried y Retschitzegger, Werner (Ed.). *Web engineering: The discipline of Systematic development of Web applications*. pp. 39-64. Great Britain: John Wiley & Sons, Ltd. 2006.
- [11] (Jacobson *et al.*, 2006)
- [12] Quatrani, T. y Palistrant, J. (2006). *Visual Modeling with IBM Rational Software Architect and UML. Developer Works Series*. United State of America: Pearson.
- [13] Rosenberg D. y Stephen M. (2007). *Use Case Driven Object Modeling with UML: Theory and Practice*. United States of America: Apress.
- [14] Koch, N. (2007). *Web Engineering: Modelling of Web Applications*. Diapositivas en inglés de las presentaciones realizadas los días 4 al 6 de junio de 2007 en Sevilla, España.
- [15] Koch, N. (2010). *UML-based Web Engineering*. Diapositivas en inglés de las presentaciones realizadas el día 27 de abril de 2010 en Sevilla, España.
- [16] Luque *et al*, 2002)
- [17] P. Díaz, I. Aedo y S. Montero,. *Desarrollo de hipermedia y web como ADM*. En Díaz, Ma. Paloma, Montero, Susana y Aedo Ignacio (coordinadores). *Ingeniería de la Web y patrones de diseño*. pp. 39-70. España: Pearson Educación. 2005
- [18] M. Fowler y K. Scott, *UML gota a gota*, México: Addison-Wesley Longman. 1999.
- CIW -Centro de Investigación de la Web- (2008). *Cómo funciona la Web*. Departamento de Ciencia de la Computación, Universidad de Chile. Consultado el 19 de septiembre del 2011 a las 20:30 de <www.ciw.cl>.
- Conallen, J. (1999). *Modeling Web application Design with UML*. Consultado el 26 de octubre del 2011 a las 22:00 de <<http://www.itmweb.com/essay546.htm>>.
- Eichinger, C. (2006). *Web Application Architectures*. En Kappel, Gerti Pröll, Birgit, Reich Siegfried y Retschitzegger, Werner (ed.). *Web engineering: The discipline of Systematic development of Web applications*. pp. 65 - 84. Great Britain: John Wiley & Sons, Ltd.

- Horstmann, C. S. y Cornell, G. (2003). *Java 2: Fundamentos*. España: Prentice Hall.
- Larousse (2002). *El pequeño Larousse ilustrado*. Colombia: Larousse.
- Martin R. C. (2004). *UML para Programadores Java*. España: Pearson educación.
- OMG (2011). *OMG Unified Modeling Language(OMG UML), Superstructure*. Actualmente. UML se encuentra en la versión 2. 4.1 que fue liberado en agosto de 2011. Consultado el 20 de enero del 2012 a las 21:00 de <<http://www.omg.org/spec/UML>>.

