

Propuesta de artefactos basados en una notación con grafos y conjuntos para el modelado conceptual de aplicaciones Web

Carlos Daniel Lima Gámez, Silvestre Gustavo Peláez Camarena, Ulises Juárez Martínez,
María Antonieta Abud Figueroa, Celia Romero Torres

Instituto Tecnológico de Orizaba, División de Estudios de Posgrado e Investigación,
Orizaba, Veracruz,
México

{lgcarlosd, ujuarez71}@gmail.com, sgpelaez@yahoo.com.mx,
mabud@itodepi.edu.mx, cromerotres@hotmail.com

Resumen. En el ámbito del desarrollo de aplicaciones Web existen metodologías como UWE (Uml Web Engineering) [1], W2000 [2] entre otras, sin embargo, es práctica común que los desarrolladores omitan su empleo y lleven a cabo de manera más bien empírica que ordenada el desarrollo de las aplicaciones; cuando se utiliza un procedimiento metodológico, comúnmente, los modelos de estas metodologías se construyen basados en el Lenguaje Unificado de Modelado (UML), lo que implica mayor nivel de conocimiento del mismo para la aplicación correcta de las metodologías basadas en éste. Por lo anterior, se proponen en el presente trabajo, una serie de artefactos basados en el uso de grafos y teoría de conjuntos, para el diseño del modelado conceptual de aplicaciones Web, que permitan alcanzar los resultados deseados de manera confiable y expedita por parte del desarrollador.

Palabras clave: Artefactos, aplicaciones Web, grafos, conjuntos.

Proposal of Artifacts based on Notation of Graphs and Sets for Conceptual Modelling of Web Applications

Abstract. In the area of Web application development there are methodologies as UWE (UML Web Engineering) [1] and W2000 [2] among others, however, it is common practice for developers skip their use and perform in empirical way rather than orderly the development of the applications; when a methodology is used, usually its models are built based on the Unified Modelling Language (UML), which implies higher level of knowledge thereof for proper application of methodologies based on it. Therefore, in this paper are proposed a series of artifacts based on the use of graphs and set theory to design the conceptual modeling of Web applications to achieve the desired results reliably and expeditiously by the developer.

Keywords: Artifacts, Web applications, graphs, sets.

1. Introducción

Hoy por hoy, el uso de las aplicaciones Web juega un rol importante como parte de la TI, debido a que día a día las empresas, instituciones educativas y diferentes ámbitos de gobierno ofrecen una amplia variedad de servicios vía Internet. Se puede mencionar como característica importante de las aplicaciones Web, su constante evolución, dado que sus requisitos cambian de manera más rápida que los de una aplicación tradicional y por lo tanto necesitan desarrollarse en menor tiempo. Debido a esta situación, y si los requisitos cambian antes de la liberación de la aplicación, esta podría ya no ser útil y el tiempo y esfuerzo que se dedicó a su desarrollo se perdería.

Una parte importante del modelado de las aplicaciones Web es el modelo navegacional, ya que en él se establecen las rutas que los usuarios pueden seguir en la aplicación para llegar a su objetivo. Por este motivo es necesario un modelo navegacional comprensible.

Otro aspecto relevante, es que las aplicaciones Web tienen características que no se encuentran en una aplicación tradicional, entre las cuales destacan su evolución constante y el aspecto navegacional, razones por las que no pueden tratarse de la misma manera que las aplicaciones tradicionales [3]. Algunas metodologías como UWE [1], W2000 [2] entre otras, basan el desarrollo de sus modelos en la notación de UML, por lo tanto, para aquellos desarrolladores que no tienen el nivel de dominio adecuado de este lenguaje, resulta de difícil comprensión por lo que sus modelos pueden no reflejar lo que en ellos quieren plasmar y por lo tanto al finalizar su aplicación no logran alcanzar las metas pretendidas en el proyecto [6].

Por lo anteriormente descrito, en este trabajo, se proponen artefactos basados en el uso de conjuntos y grafos para el modelado del contenido y la navegación respectivamente, dando como resultado un modelo de clases con una notación con base en la teoría de conjuntos y un modelo navegacional con una notación con símbolos propios de la navegación de una aplicación Web.

2. Trabajos relacionados

En [4] se menciona la poca importancia que se le da al modelado de aplicaciones Web ya que algunos desarrolladores piensan que es pérdida de tiempo y los que llegan a modelar solo lo hacen en las primeras etapas del proceso de desarrollo y después estos modelos no se actualizan con los cambios que se hacen en la etapa de desarrollo. A partir de estos problemas nace la Arquitectura Dirigida por Modelos (MDA) en el cual los modelos evolucionan y se transforman hasta generar de forma automática el código fuente de la aplicación o una parte del mismo. También se plantea el problema de que en la actualidad aunque existen muchas herramientas basadas en la arquitectura MDA que generan código fuente, ninguna de ellas permite generar una aplicación 100% funcional, por tal razón, en [4] se plantea la necesidad de generar una metodología para el modelado de aplicaciones Web que se oriente en la generación del código fuente y que permita a los arquitectos y analistas definir de forma sencilla el comportamiento del sistema. En [5] se menciona el tema de la educación con base en la Web y las variaciones que introdujo en los modelos de la educación formal, ciertos aspectos tales como las comunidades de aprendizaje,

aprendizaje autónomo y colaborativo adquirieron mayor relevancia. Por tal motivo, en [5] se propone una forma de modelar la organización de una comunidad educativa colaborativa en la Web, con el uso de grafos. Para lograr el modelado se describe el esquema propuesto que se basa en una estructura de dígrafo jerárquico el cual contiene un nodo raíz que representa al maestro encargado del curso, se describen con detalle las relaciones que se producen entre los diferentes participantes de la comunidad en el grafo mediante el uso de arcos continuos bidireccionales y arcos unidireccionales. Se concluye que modelar la comunidad no es una tarea sencilla ya que requiere de planificaciones sucesivas conforme evoluciona la comunidad pero gracias a la propuesta del modelado mediante estructuras de grafos resulta alentador ya que para el manejo de estas estructuras existen diversos algoritmos para realizar las planificaciones que se mencionan. En [6] se menciona que el desarrollo de aplicaciones Web es una tarea compleja y desafiante que cuenta con muchas fases y con diferentes tipos de usuarios finales, también se menciona que la mayoría de las aplicaciones Web se desechan o no alcanzan a ganar popularidad y la razón es porque la aplicación no logra las metas de usabilidad o no es lo que los clientes esperaban.

Se afirma que el fracaso de que una aplicación Web no alcance las metas de usabilidad radica principalmente en la pobre comunicación entre los diferentes equipos de desarrollo los cuales generalmente son de diferentes disciplinas (especialistas en interfaces humano-computadora, diseñadores de experiencias de usuario, equipo de aseguramiento de la calidad y equipo de marketing). También se menciona que para facilitar el desarrollo de las aplicaciones Web se necesita un protocolo de comunicación que sea sólido y auto descriptivo para todos los miembros de los diferentes equipos en etapas tempranas del desarrollo. Se propone para tratar de dar solución al problema descrito, un enfoque navegacional con base en bosquejos el cual sea traducible a otros modelos arquitectónicos y a su vez a la posibilidad de generación de código, el diagrama contiene solamente los componentes esenciales que se relacionan con la navegación e interacción del usuario con la aplicación.

3. Artefactos propuestos

Se proponen artefactos para el modelado de contenido y modelo navegacional, como a continuación se detalla:

3.1 Modelado de contenido

El Modelado de contenido se obtiene a partir del modelo de clases definido mediante la notación propuesta con base en la teoría de conjuntos [7, 8] estableciendo como referencia las siguientes reglas:

Regla 1: La letra inicial del nombre de la clase se debe escribir con mayúscula, los atributos y métodos se deben escribir con sus respectivas llaves separados por comas como se establece en la notación de la teoría de conjuntos, por ejemplo:

Persona{ {nombre, apellido, edad},{caminar(), correr()}}

Regla 2: La ausencia de atributos o métodos en la definición de una clase se representa con el símbolo de conjunto vacío " \emptyset ", ejemplo:

Persona{ \emptyset , {caminar(), correr()}}

Regla 3: Para la representación de una clase abstracta se antepone "Abs_" al inicio del nombre de la clase, por ejemplo:

Abs_Persona{{nombre, apellido, edad}, {caminar(), correr()}}

Regla 4: Para representar una interfaz se antepone "In_" al inicio del nombre de la clase, ejemplo:

In_Persona{ \emptyset , {caminar(), correr()}}

Regla 5: Para una clase que implementa una interfaz, se representa utilizando el símbolo de implicación " \Rightarrow ", ejemplo:

Alumno \Rightarrow In_Persona

Regla 6: La herencia se representa utilizando el símbolo " \cup " para unión de conjuntos, ejemplo:

Clase 1: Persona{{nombre, apellido}, {caminar()}}

Clase 2: Alumno{{numctrl, carrera}, {estudiar()}}

"Alumno hereda de persona", se representa como:

Alumno \cup Persona

Regla 7: La composición y agregación de clases, se representan utilizando los símbolos " \in " y " \subset " respectivamente, por ejemplo, para las siguientes clases:

Videoclub{{direccion, telefono}, {agregarPel()}}

Socio{{id, nombre}, {actualizarDatos()}}

SocioVip{ \emptyset , {aplicaDesc()}}

Pelicula{{idp, nombre, genero}, {actualizarInfo()}}

La representación de la composición se muestra en el siguiente ejemplo utilizando el símbolo " \in ":

Pelicula \in Videoclub

El ejemplo a continuación muestra cómo representar la agregación mediante el símbolo " \subset ":

Socios \subset Videoclub

Regla 8: Para establecer la asociación entre clases, utilizar el símbolo de intersección " \cap ", por ejemplo para las siguientes clases:

Busca

Cliente \cap Hotel

La notación anterior significa que existe la asociación entre la clase "Cliente" y la clase "Hotel", la palabra "busca" sobre el símbolo de asociación especifica el tipo de relación que existe entre las dos clases; para el ejemplo se lee "Cliente busca hotel".

Regla 9: Para indicar la multiplicidad (en agregación, composición y asociación), realizarlo como en el siguiente ejemplo:

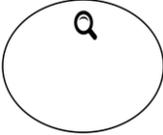
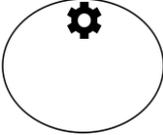
Busca
 Cliente 1..* \cap 1..* Hotel

El ejemplo anterior se lee: "Uno o más clientes buscan uno o más hoteles".

3.2 Modelo navegacional

Los modelos navegacionales muestran las rutas posibles que un usuario puede seguir dentro de la aplicación en función del rol que se le asigne, para realizar este modelo se propone una notación basada en grafos, ésta consta de los elementos que se observan en la Tabla 1.

Tabla 1. Elementos para realizar el modelo navegacional.

Símbolo	Nombre	Descripción
	Nodo Navegacional	Representa un punto de la navegación en el que se le muestra información al usuario.
	Nodo Menú	Representa un menú, es decir, sirve para manejar rutas alternativas de navegación.
	Nodo Consulta	Representa puntos de la navegación donde la aplicación solicita información al usuario.
	Nodo Índice	Representa puntos de navegación donde al usuario se le presenta una lista de posibles resultados para visualizar
	Nodo Proceso	Representa procesos/tareas a realizar en la aplicación
	Acceso Usuario	Sirve para establecer restricciones en la navegación, se le añade un identificador con un rol de usuario y se coloca al inicio de un enlace que sale de cualquiera de los nodos descritos anteriormente para indicar que solamente ese usuario tiene acceso a ese destino de navegación

Símbolo	Nombre	Descripción
	Enlace de navegación	Representa un enlace de navegación de un nodo de navegación a otro, no puede usarse con nodos de proceso/consulta, cuando se omite la punta de flecha se indica que la navegación es bidireccional.
	Enlace de Proceso	Representa un enlace de proceso que va de un nodo de proceso/consulta a otro, no se puede usar para nodos de navegación (para nodos de navegación se usan enlaces de navegación), como en el caso del enlace de navegación si se omite la punta de flecha se indica que la navegación es bidireccional.

4. Primeras pruebas

Para probar los artefactos propuestos, se realizó el modelado de dos casos; el primero consta de una aplicación que muestra los socios de un videoclub, el segundo trata de un sitio de venta de libros.

En la Fig. 1 se muestra el caso de uso del primer caso (Videoclub).

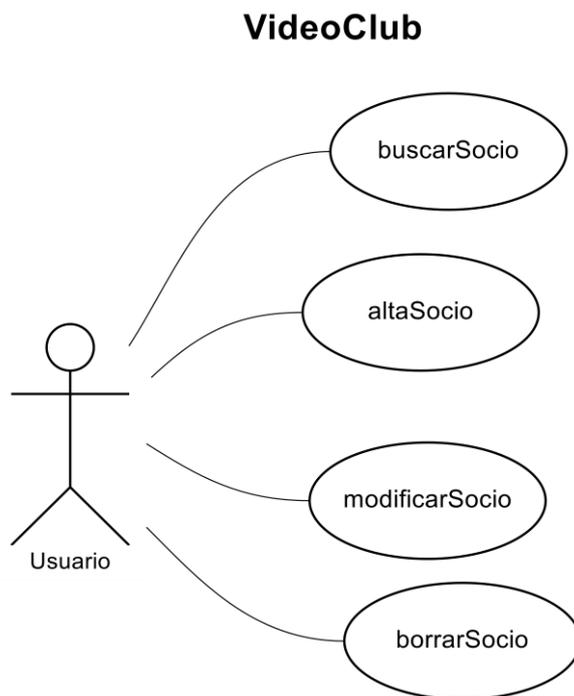


Fig. 1. Caso de uso del videoclub.

A continuación, con base en la notación propuesta de teoría de conjuntos, se desarrolla el modelo de clases de la aplicación como se muestra en la Fig. 2.

Definición de clases	Relaciones entre clases
Videoclub{{descripcion},∅}	Socio 1..* ∈ 1 Videoclub
Socio{{nombre, apellidos, email},∅}	Socio 1 ∩ 1 Vigencia
Vigencia{{inicio, fin },∅}	Socio 1 ∩ 1 Telefono
Telefono{{codigo_area, lada, numero},∅}	Socio 1 ∩ 1 Direccion
Direccion{{dir, cpostal, ciudad, pais},∅}	

Fig. 2. Modelo de clases con base en la notación de la teoría de conjuntos (videoclub).

Como último artefacto para este caso se presenta el modelo navegacional que se obtiene para la aplicación con base en la notación propuesta para este fin (ver Fig. 3).

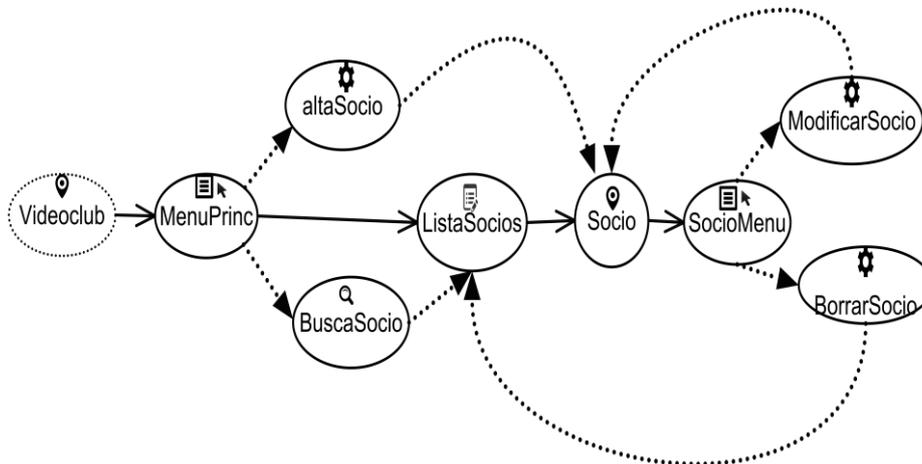


Fig. 3. Modelo navegacional con la notación propuesta (videoclub).

La interpretación del Modelo navegacional anterior es la siguiente: el nodo navegacional con línea discontinua indica el punto de inicio de la navegación, es decir la página principal de la aplicación, el *nodo menú* ("MenuPrinc") muestra tres posibles destinos a seguir ("altaSocio", "ListaSocios", "BuscaSocio"), si se da de alta un nuevo socio ("altaSocio") al finalizar se mostrará la información del nuevo socio, otra opción es ver directamente la lista de todos los socios existentes ("ListaSocios"), pero si se busca por un contacto en específico ("BuscaSocio") se mostrará la lista ("ListaSocios") con los contactos que coincidan con los criterios de búsqueda, cada uno con su respectivo enlace donde se muestra su información a detalle ("Socio"), cada contacto cuenta con dos opciones ("SocioMenu"), la primera para representar la funcionalidad de actualizar un contacto("ModificarSocio") donde al término muestra la información actualizada del contacto y la segunda opción ("BorrarSocio") donde al terminar se regresará nuevamente a la lista de contactos.

A continuación se presentan los artefactos realizados para el segundo caso: en la Fig. 4 se observa el caso de uso para la aplicación "Venta de libros".

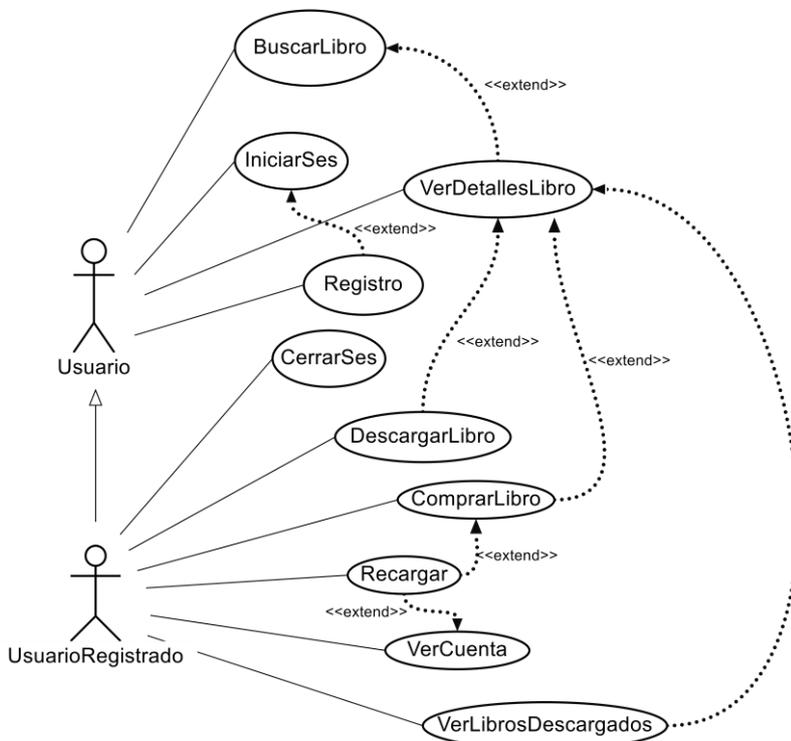


Fig. 4. Caso de uso de sitio de venta de libros.

Posteriormente en la Fig. 5 se observa el modelo de clases para este caso:

Definición de Clases:

- Libro {{nombre, precio, linkdescarga, descripcion}, ∅}
- Autor {{nombre, pais}, ∅}
- Capitulo {{nombre, pagin, pagfin}, ∅}
- Usuario {{nombre, contrasenia, creditos}, {comprarAlbum(), recargar(), guardar()}}
- Sesion {∅, {establecerSesion(), cerrarSesion()}}

Relaciones:

- Sesion 1 ∩ 1 Usuario
- Usuario 1 ∩ * Libro
- Capitulo 1..* ∩ 1 Libro
- Libro * ∩ 1 Autor

Fig. 5. Modelo de clases con base en la notación de la teoría de conjuntos.

Para finalizar se presenta en la Fig. 6 el modelo navegacional del sitio de venta de libros.

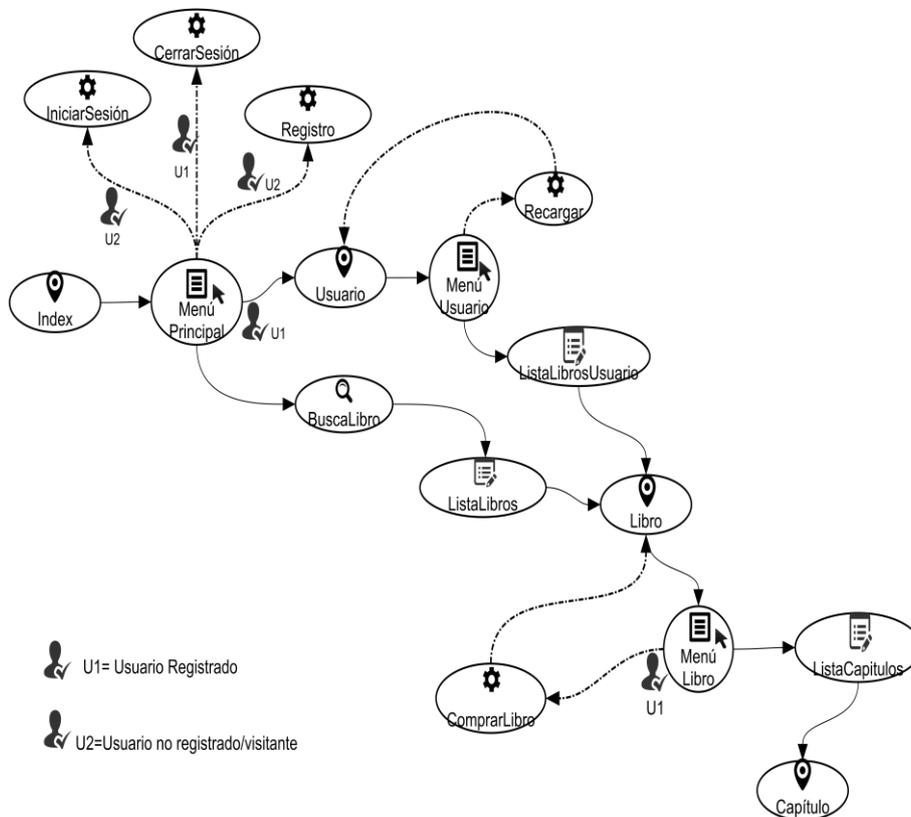


Fig. 6. Modelo de clases con base en la notación de la teoría de conjuntos.

Como se observa en el modelo de la Fig. 6, el elemento que se presenta en la Tabla 1 "Acceso Usuario" ayuda a identificar que usuarios tienen acceso a determinados recursos, como se muestra en el modelo solo el usuario registrado "U1" puede acceder a "CerrarSesión", "Usuario" y "ComprarLibro", con esto se evita el crear un diagrama de navegación para cada usuario que se identifica en la aplicación.

5. Conclusiones y trabajo a futuro

Se puede observar que la teoría de conjuntos cuenta con los elementos necesarios para la representación de un modelo de clases, sin embargo; como todo lo nuevo, es requerido familiarizarse con estos conceptos pero una vez dominada la notación es fácil de aplicar, dando como resultado un esquema más compacto y las relaciones entre las clases se comprenden mejor, además, para la actualización del modelo en caso de que haya cambios a lo largo del desarrollo de una aplicación Web, la notación propuesta resulta flexible para estos casos.

La notación propuesta para el modelo navegacional utiliza símbolos fáciles de relacionar con los aspectos navegacionales de las aplicaciones Web evitando confusiones, por lo que se considera igualmente de fácil aplicación.

Los artefactos propuestos corresponden a la fase inicial para un cambio en el procedimiento metodológico para el modelado de aplicaciones Web. Por lo tanto, como trabajo a futuro se requiere complementar artefactos a este trabajo para la adición del modelo de presentación, y estar en condiciones de posteriormente definir una metodología con base en estos artefactos para así poder realizar una validación formal de los mismos con el desarrollo de un caso de estudio.

Una vez definida la metodología, se considera la implementación de una herramienta CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Computadora) que dé el soporte y así automatizar el uso de los artefactos propuestos.

Referencias

1. Baresi, L., Colazzo, S., Mainetti, L., Morasca, S.: W2000: A modelling notation for complex Web applications. In: Mendes, E., Mosley, N. (Eds): *Web Engineering*, Springer, pp. 335–364 (2006)
2. Koch, N., Knapp, A., Zhang, G., Baumeister, H.: UML-based Web Engineering: An approach based on Standards. In: Rossi, G., Pastor, O., Schwabe, D., Olsina, L. (Eds.): *Web Engineering: Modelling and Implementing Web Applications*, Springer, pp. 157–191 (2008)
3. Sánchez Santamaria, M., García García, L.A.: *La Ingeniería Web: Desarrollo de aplicaciones Web de alta calidad*. HYPATIA, <http://hypatia.morelos.gob.mx> (2011)
4. Vera, P., Pons, C., González, C., Giulianelli, D., Rodríguez, R.: Metodología de Modelado de Aplicaciones Web Móviles Basada en Componentes. In: *XV Workshop de Investigadores en Ciencias de la Computación (WICC)*, pp. 451–455 (2013)
5. Moroni, N., Señas, P.: Uso de grafos para el modelado de experiencias educativas colaborativas basadas en la Web. In: *XIII Congreso Argentino de Ciencias de la Computación (CACIC)*, pp. 1134–1145 (2007)
6. Bouchrika, I., Ait-Oubelli, L., Rabir, A.: Mockup-based navigational diagram for the development of interactive Web applications. In: *Proceedings of the International Conference on Information Systems and Design of Communication (ISDOC)*, pp. 27–32 (2013)
7. Lipschutz, S.: *Sets and basic operations on sets: Set theory and related topics*. McGraw-Hill, pp. 5–6 (1998)
8. Lipschutz, S., Lars Lipson, M.: *Set Theory: Discrete Mathematics*. McGraw-Hill, pp. 1–22 (2007)