

MODELADO CONCEPTUAL DE UNA APLICACIÓN WEB USANDO LA METODOLOGÍA OOWS: UN CASO PRÁCTICO

*M. Castillo-Montes, J. Vargas-Enríquez & L. García-Mundo**

Tecnológico Nacional de México-Instituto Tecnológico de Ciudad Victoria, Boulevard Emilio Portes Gil, #1301, Pte. A.P. 175, C.P. 87010, Cd. Victoria, Tamaulipas, México.

**lgarcm64@gmail.com*

RESUMEN. Este trabajo presenta el uso de la metodología OOWS (Object Oriented Web Solution) para el desarrollo de un modelo conceptual de una aplicación Web que gestiona el expediente clínico de los pacientes de un consultorio médico. El modelo conceptual diseñado para la aplicación Web "Gestión de expedientes clínicos", permite comprender mejor la estructura funcional de esta aplicación y facilita su mantenimiento, al describir de manera visual su estructura, así como la relación de las clases que la componen.

PALABRAS CLAVE: Expediente clínico, pacientes, estructura funcional.

ABSTRACT. This work presents the use of the OOWS methodology (Object Oriented Web Solution) for the development of a conceptual model of an application Web that manages the clinical files of patients in a medical office. The conceptual model designed for the "Management of clinical files" Web application, allows a better understanding of the functional structure of this application and facilitates its maintenance, by visually describing the structure as well as the relationship with the classes that compose it.

KEY WORDS: Clinical files, patients, functional structure.

1. INTRODUCCIÓN

Las aplicaciones Web, son aplicaciones software que se ejecutan a través de un navegador, en diferentes plataformas y arquitecturas, sin tener que ser instaladas en alguna plataforma en particular. Al ser aplicaciones software pueden modelarse utilizando la Ingeniería de Software (IS). La IS es una disciplina en la que, intervienen diferentes procesos que permiten diseñar soluciones software para arquitecturas específicas empleadas generalmente de manera local en un equipo de cómputo.

En un inicio, las aplicaciones Web se diseñaban utilizando las prácticas de la IS, sin embargo, por su naturaleza deben tener en cuenta aspectos particulares que las aproximaciones tradicionales de la IS no soportan. Aunado a esto, el acelerado crecimiento y la expansión de Internet han dado lugar a aplicaciones Web cada vez más complejas y dinámicas. Surge como consecuencia, la necesidad de desarrollar aplicaciones funcionales que utilicen la práctica de la IS en el contexto de la Web, en la disciplina conocida como Ingeniería Web (IW). La IW se centra en propuestas metodológicas de desarrollo Web, con el fin de mejorar tanto

la calidad del proceso de desarrollo como la del producto final.

La IW se define como "La aplicación de una aproximación sistemática, disciplinada y cuantificable al desarrollo, operación y mantenimiento de aplicaciones basadas en la Web o la aplicación de la ingeniería al software basado en la Web" (Murugesan et al., 2001). En los últimos años han surgido una serie de metodologías que ayudan a sistematizar la construcción de soluciones en ambientes Web, las cuales proporcionan mecanismos de abstracción que facilitan la conceptualización y desarrollo de estos sistemas. Esta conceptualización es representada mediante un conjunto de modelos que ayudan a mejorar la calidad y la eficiencia de su desarrollo. Los modelos conceptuales capturan de forma abstracta la representación detallada de la aplicación Web y a partir de ellos es factible la generación sistemática del código que la implementa.

La diferencia entre el modelado de las aplicaciones de software tradicionales y las aplicaciones Web, reside principalmente en la interacción del usuario con las aplicaciones, ya que los requisitos de funcionalidad, navegación

y presentación son diferentes o no existen en las aplicaciones software. Existen diversas propuestas metodológicas para crear modelos conceptuales de aplicaciones Web, que han sido probadas y aceptadas en el desarrollo de diferentes aplicaciones. Las metodologías permiten la definición de modelos que capturan los aspectos estructurales, de navegación y de presentación de este tipo de software. En el ámbito científico de la IW, existen propuestas para modelar diversas situaciones, tales como *WSDM* (De Troyer y Casteleyn, 2003), *WebML* (Ceri et al., 2003), *WA-UML* (Raoudha et al., 2006), *UWE* (Koch, 2000), *OOWS* (Fons et al., 2003), *OOH* (Gómez et al., 2000) y *OOHDM* (Schwabe et al., 1996), así como propuestas de arquitecturas para aplicaciones Web, como *WebSA* (Beigbeder, 2007), *Interactive Web Modelling Language* (Wright, 2007) e *IFML* (OMG, 2015).

El propósito de este artículo es describir un caso práctico del uso de la metodología *OOWS* para construir el modelo conceptual de navegación de una aplicación Web que gestiona el archivo clínico de los pacientes de un consultorio médico. La principal contribución de este trabajo es la descripción práctica del uso de la metodología *OOWS*, para el modelado y diseño de la navegación de una aplicación Web real. El resto de este trabajo está organizado de la siguiente forma. La sección 2 presenta algunos trabajos relacionados con el uso de la metodología *OOWS*, la sección 3 describe la metodología *OOWS* para el diseño de modelos conceptuales de aplicaciones Web. En la sección 4 se muestra el desarrollo de un modelo conceptual para la aplicación Web "Gestión de expedientes clínicos" utilizando el *OOWS*. Por último, las conclusiones de este trabajo se presentan en la sección 5.

2. TRABAJO RELACIONADO

En esta sección se presentan algunos trabajos en los cuales se han creado modelos conceptuales de aplicaciones Web, utilizando la metodología *OOWS*. Ejemplos de estos trabajos son: Valverde, 2007. Expone un caso práctico sobre la creación de un modelo conceptual de una aplicación Web que gestiona una base de datos de películas en Internet, aplicando la metodología *OOWS* en un entorno de desarrollo dirigido por modelos llamado

OOWS Suite, el cual proporciona herramientas de soporte para la aplicación de la metodología *OOWS*. Pastor, 2001. Propone por primera la metodología *OOWS*, la cual provee de mecanismos para tratar con el desarrollo de sistemas de información de hipermedios y aplicaciones de comercio electrónico en entornos Web. La principal contribución de este trabajo es la introducción del modelo de navegación el cual está completamente embebido en el proceso de modelado conceptual. Valderas, 2014. Presenta una aproximación del desarrollo de un caso de estudio de caso en el que se aplica la metodología *OOWS* para construir el modelo conceptual de un portal Web universitario. Fons, 2000. Propone la metodología *OOWS* cómo una extensión del método de producción orientado a objetos enfocado al software, en el cual introduce las expresiones necesarias para capturar los requisitos de presentación y de navegación de las aplicaciones Web. Quintero, 2008. Describe una extensión para la metodología *OOWS*, que permite integrar aplicaciones externas incluyendo adecuaciones y extensiones, a las primitivas conceptuales del *OOWS*, de consulta y actualización de datos.

3. METODOLOGÍA OOWS

El *OOWS* es una metodología de IW que proporciona soporte para el desarrollo de aplicaciones Web. Esta metodología es una extensión del modelo de IS Orientado a Objetos (*OO-Method*) (Pastor et al., 2001), el cual se centra en la captura de la estructura estática y el comportamiento de un sistema mediante tres modelos:

1. Un modelo Estructural: define la estructura estática del sistema a través de la definición de sus clases y de las relaciones entre ellas utilizando diagrama de clases UML (Unified Modeling Language).
2. Un modelo Dinámico: describe las secuencias válidas de estados de los objetos en la vida real mediante Diagramas de Transición de Estados UML, así como las interacciones entre los objetos de diferentes clases por medio de Diagramas de Secuencia UML.
3. Un modelo Funcional: contiene las semánticas de los cambios de estado de un

objeto para definir el efecto de sus servicios usando una especificación formal textual.

Con el propósito de soportar las características que se enfatizan en el desarrollo Web, el OOWS parte del *OO-Method* e introduce tres nuevos modelos que se construyen de forma consecutiva:

1. Modelo de Usuarios: especifica la categorización sobre los tipos de usuarios que interactúan con el sistema, además de la interrelación entre ellos.
2. Modelo de Navegación: especifica la visibilidad del sistema de los datos y su funcionalidad, además de los caminos válidos de la estructura del sistema (semántica de navegación) para cada tipo de usuario. Su especificación parte de una vista global a una vista específica detallada de cada tipo de usuario.
3. Modelo de Presentación: especifica los requisitos de presentación para los elementos definidos por el Modelo de Navegación, por lo que se encuentra fuertemente ligado a él. Este modelo utiliza los contextos de navegación en términos de interacción del usuario con el sistema.

4. DESARROLLO DEL MODELO CONCEPTUAL DE NAVEGACIÓN

El modelo conceptual que se presenta en este trabajo corresponde a la aplicación Web “Gestión de expedientes clínicos”, cuya función es obtener y representar la semántica de la navegación de esta aplicación. Para construir el modelo de navegación se parte de un modelo de clases previamente definido y a partir de él se definen tres modelos (ver sección 3): modelo de usuarios, modelo de navegación y modelo de presentación. La aplicación Web estará integrada por una base de datos y formularios que permitirán organizar la información de los pacientes. Esta información es administrada por el médico, quien cuenta con los permisos necesarios para crear, modificar y actualizar los expedientes clínicos de sus pacientes. El médico asigna a cada uno de sus pacientes un usuario y contraseña para que puedan consultar su expediente clínico cuando ellos lo requieran. El diagrama de casos de uso de la Figura 1 muestra los actores principales de esta aplicación Web así como las acciones requeridas por cada uno de ellos.



Figura 1. Diagramas de casos de uso de la aplicación Web “Gestión de expedientes clínicos”.

Los casos de uso se pueden definir mediante la aplicación de un cuestionario cuyos resultados

permiten establecer los diferentes usuarios que tendrá la aplicación, así como sus requisitos.

Una vez establecidos los requisitos de la aplicación Web, se determinó que para su operación se requería la participación de dos usuarios autenticados (ver Figura 1). El modelado conceptual de aplicaciones Web, parte de un modelo que define la estructura estática de un sistema mediante la definición de sus clases y sus relaciones (diagrama de clases UML).

4.1. Diagrama de Clases

En esta subsección se muestra una vista parcial del diagrama de clases construido para

4.2. Modelo de Usuarios

La primera fase en el diseño de un modelo de navegación, es la construcción de un diagrama de usuarios, o diagrama de agentes. En este diagrama, los usuarios identificados se organizan jerárquicamente mediante relaciones de herencia, permitiendo establecer especializaciones de navegación, así como la reutilización a los usuarios hijos de los modelos de navegación de su usuario padre.

El diagrama muestra a los usuarios potenciales que interactúan con la aplicación, las interrelaciones entre ellos y su modo de acceso al sistema. Los usuarios o agentes pueden ser de dos tipos:

1. Agentes instanciables. Representan a los usuarios que la aplicación Web reconocerá, y pueden ser de dos tipos:

- *Anónimos*. Usuarios que no requieren ninguna identificación para utilizar la aplicación (representados por símbolo "?").
- *Registrados*. Usuarios que requieren identificarse para utilizar la aplicación (representados por un símbolo de candado).

2. Agentes abstractos. Son los usuarios de la aplicación que definen mecanismos de especialización (representados por un símbolo "X"). Estos usuarios tienen asociado un modelo de navegación que puede ser reutilizado por los usuarios que lo heredan.

la aplicación Web "Gestión de expedientes clínicos" (ver Figura 1).

En este diagrama se muestran las clases que permiten modelar la la aplicación Web "Gestión de expedientes clínicos" (ver Figura 2). En este diagrama se muestran las clases que permiten modelar. El diagrama muestra, a través de los atributos de las clases, la información que contendrá el expediente del paciente, así como cada una de las acciones o métodos que se pueden realizar.

Después de analizar los diferentes accesos y la visibilidad de la información de la aplicación Web, se detectaron tres tipos de usuarios: internauta, paciente y médico (Ver Figura 2). El usuario internauta es un usuario anónimo que tiene un rol de visitante (no requiere identificación), sin embargo, si se autentica mediante un usuario y contraseña se produce un cambio de rol, lo que ocasiona que su interacción y sus vistas cambien. Por tanto, los usuarios paciente y médico son agentes abstractos que tienen los mismos privilegios que los internautas, pero tienen, además, cada uno de ellos, nuevas propiedades de navegación diferentes. Esto significa que tanto el médico como el paciente tendrán su propio mapa de navegación: el usuario internauta tiene acceso solamente a la información general del sitio, el agente paciente es un usuario que debe identificarse y tiene acceso solamente a su expediente clínico, el usuario médico debe identificarse y tiene control total de la información.

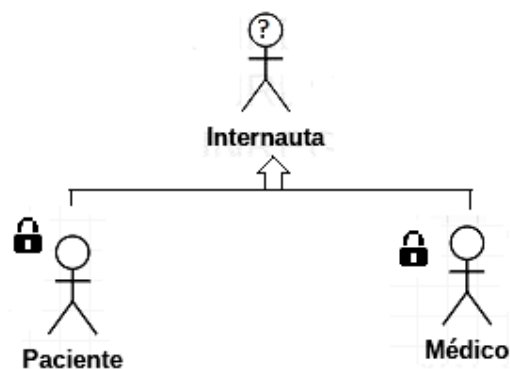


Figura 2. Tipos de usuarios de la aplicación Web "Gestión de expedientes clínicos".

4.3. Modelo de Navegación

El modelo de navegación permite establecer los requisitos de navegación de una aplicación Web. Este modelo crea una vista o *mapa de navegación* que muestra los *caminos de navegación* para cada tipo de usuario que interactúa con el sistema de acuerdo a lo que se define en el diagrama de clases UML. La semántica de navegación de cualquier aplicación Web se captura por cada uno de los agentes identificados, mostrando así el acceso que tendrán cada uno de ellos de acuerdo a los requisitos establecidos. La construcción de este modelo de navegación se realiza mediante la identificación de los usuarios que interactúan con la aplicación Web, así como con la construcción de los mapas de navegación (vista de navegación) por cada uno de estos usuarios.

4.3.1. Mapas de navegación

La segunda fase en el diseño del modelo de navegación es la construcción de los mapas de navegación para cada uno de los diferentes tipos de usuarios de la aplicación. Los mapas de navegación se definen a través de un grafo dirigido cuyos elementos son:

1. Nodos. Son unidades de interacción con el usuario del sistema, gráficamente representados mediante paquetes UML (Selic, et al., 2015).
2. Arcos. Representan los caminos de navegación válidos entre los nodos, gráficamente representados por flechas.

La Figura 3 muestra el mapa de navegación para el usuario internauta, el cual define la vista del sistema para este usuario. El mapa le

proporciona al usuario internauta, la accesibilidad a tres nodos de navegación (Inicio, Servicios y Acceso). Cada uno de estos nodos representa un *contexto de navegación* (estereotipado con la palabra <<context>>) que proporciona una vista de la información y la funcionalidad definidas en el diagrama de clases. Los contextos de navegación etiquetados con la letra “E” son de tipo Exploración. Significa que son alcanzables desde cualquier página del sitio. Uno de los nodos se etiqueta con una “H” indicando que este contexto corresponde a la página de inicio de la aplicación. Es importante destacar que este mapa es visible para todos los usuarios, quienes al autenticarse en el sistema provocan un cambio de contexto y su mapa se extiende a las páginas alcanzables dependiendo del tipo de usuario.

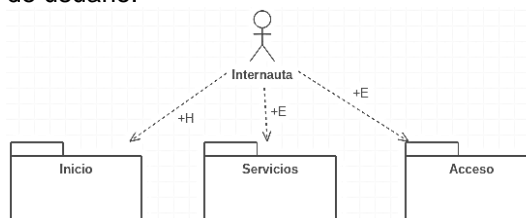


Figura 3. Mapa de navegación para el usuario internauta.

Cuando un usuario inicia una sesión en la vista de acceso, se produce un cambio de contexto y a partir de esta vista se hacen alcanzables otras páginas. Las vistas etiquetadas con una “S”, representan las páginas que son alcanzables solamente mediante una secuencia o camino de navegación. La Figura 4 muestra el mapa de navegación con contextos de navegación de secuencia alcanzables para los usuarios autenticados.

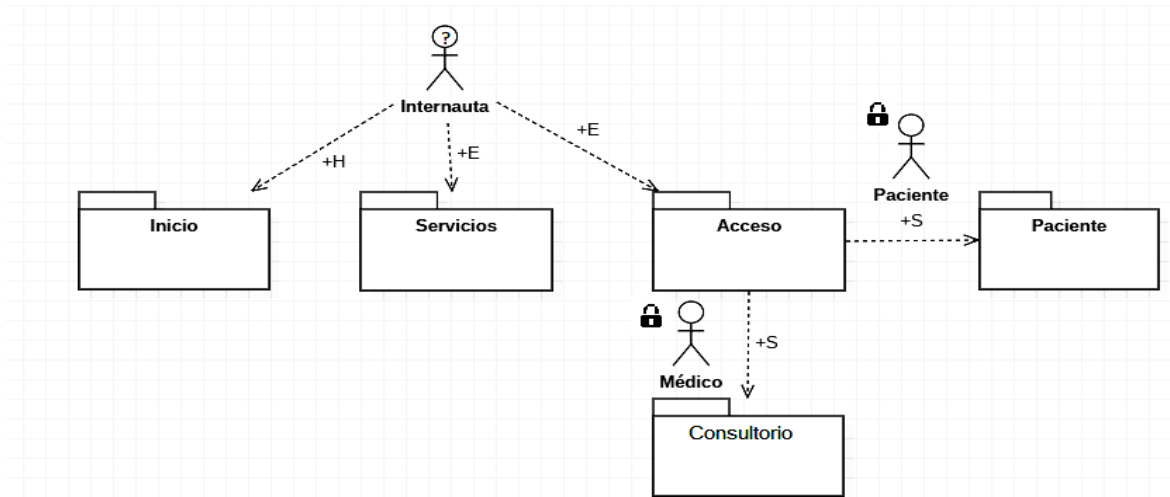


Figura 4. Mapa de navegación con contextos de secuencia alcanzables para usuarios autenticados.

Cuando se ha definido la navegación (mapas de navegación) para cada agente o tipo de usuario detectado, se definen los contextos de navegación. Un contexto de navegación está compuesto por un conjunto de clases de navegación que se estereotipan con la palabra «view». Las clases de navegación hacen referencia a clases identificadas en el diagrama de clases y contienen solo los atributos y operaciones que estarán disponibles para la vista del usuario de un determinado contexto. La Figura 6, muestra las clases de navegación “Médico” y “ExpedientePaciente” del contexto de navegación “Consultorio”. Un contexto de navegación debe incluir una clase **principal** o **directora** (clase “Medico” del contexto “Consultorio”) y opcionalmente **clases complementarias** (clase “ExpedientePaciente” en el contexto Consultorio) que complementan la información de la clase principal. Las relaciones entre las clases de navegación representadas por una línea continua establecen un enlace de secuencia para navegar a un contexto destino activando la navegación hacia ese vínculo. En el contexto

“Consultorio”, existen 2 relaciones de secuencia que, aunque apuntan a la misma clase, sus contextos y sus métodos de recuperación de información son distintos. Uno de ellos permite el ingreso de pacientes nuevos y el otro proporciona la consulta del expediente de los pacientes registrados previamente. Así mismo, en el contexto “Pacientes Registrados” se agrega otra vista de secuencia llamada “Historia Clínica” cuya finalidad es consultar el historial médico del paciente. Éste historial contiene toda la información definida en la clase “ExpedientePaciente” que está asociada al paciente.

Cuando la relación de navegación entre las clases es representada por una línea discontinua, se indica una recuperación de información de la clase que se conecta (En la clase “ExpedientePaciente” del contexto “Pacientes Registrados” recupera la información de las clases “Recetas” y “Estudios” así como los métodos definidos en la clase “Medico” para realizar las operaciones necesarias en la base de datos).

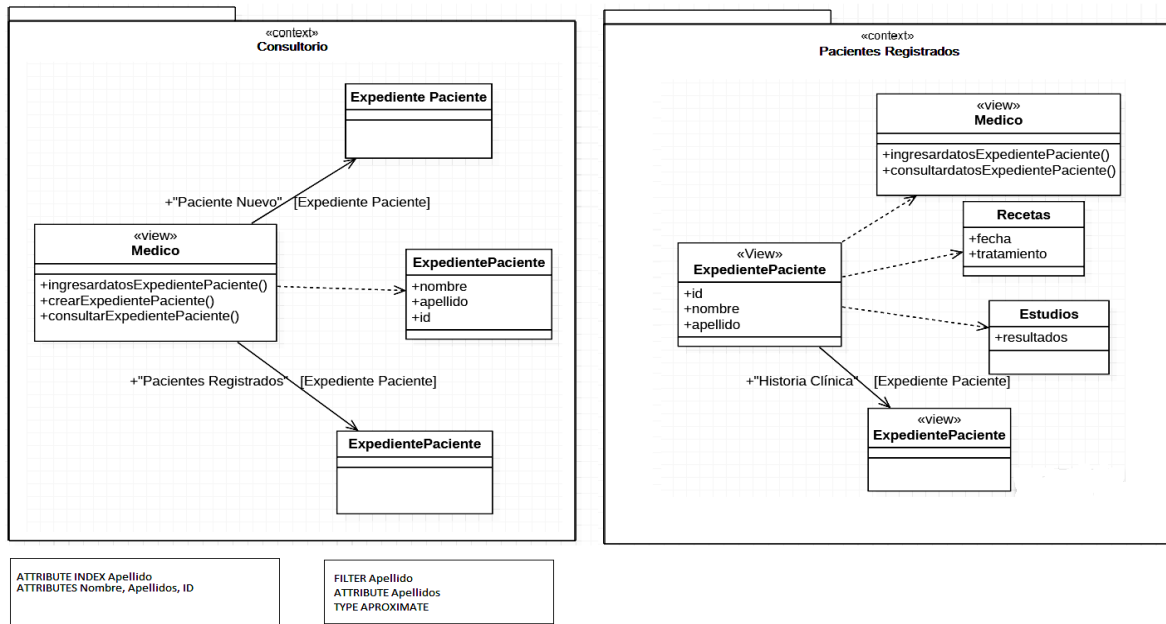


Figura 5. Mapas de navegación de los contextos “Consultorio” y “Pacientes Registrados”.

4.3.2. Diseño de Navegación Avanzado

Los contextos de navegación recuperan la información de las clases que se definen en el esquema conceptual. Cuando la información que se recupera es demasiada, resulta difícil poder manipularla. Para ayudar a los usuarios a explorar y navegar a través de toda esa cantidad de información, el OOWS establece dos mecanismos para navegar y filtrar información en un contexto de navegación: índices y filtros. Esas características se describen en un cuadro de línea discontinua en la parte inferior de los contextos de navegación. Los índices proporcionan acceso indexado a la población de la clase principal a través de uno o varios atributos. La propiedad indexada puede pertenecer a la clase de principal (atributo índice) o a cualquier clase complementaria (índice de relación). Cuando el índice es de relación, éste debe especificarse. Cuando se activa un índice, se crea una lista de todos los valores posibles para los atributos indexados. Al elegir uno de estos valores, todos los objetos que tienen el mismo valor para esta propiedad se mostrarán en una vista de búsqueda. Esta vista de búsqueda describe la información que estará disponible para ayudar al usuario a seleccionar una instancia. Esta instancia seleccionada se activará en el contexto de navegación. Un filtro define una condición para restringir la información que se va a recuperar. Existen 3 tipos de filtros:

1. Exacto. Recupera el conjunto de información que coincide exactamente con el valor indicado.
2. Aproximado. Recupera el conjunto de información que es semejante al valor indicado.
3. Rango. Recupera el conjunto de información que se encuentra entre un valor máximo y un valor mínimo.

En el modelo conceptual de la aplicación Web “Gestión de expedientes clínicos” el índice utilizado es el apellido del paciente. El filtro establecido es el del valor aproximado debido a la posible duplicidad de los apellidos de los pacientes (ver Figura 6). Para diferenciarlos se usarán los atributos “nombre” e “id”.

4.4. Modelo de Presentación

Una vez que el modelo de navegación es construido, los contextos de navegación que se definieron en él, son la base para especificar los requisitos de presentación de la aplicación Web. En el Modelo de Presentación se define la forma en que se presentará la aplicación al usuario, es decir el *layout* (información) de las interfaces. Los requisitos de presentación son especificados mediante patrones asociados a las primitivas de los contextos de navegación (clases de navegación, enlaces de navegación, mecanismos de búsqueda, etc.). Los patrones

de presentación definen el nivel de acceso, búsqueda (filtros) y las relaciones entre diferentes clases. Existen 3 patrones básicos de presentación:

Paginación de información. Captura la semántica de desplazamiento lógico de la información. Las instancias recuperadas se dividen en bloques lógicos, los cuales se muestran uno a la vez. Se definen además los mecanismos de avance y retroceso entre las páginas lógicas obtenidas. La información que se requiere para la paginación es:

1. Cardinalidad. Define la cantidad de objetos de un bloque lógico. Puede ser estática o dinámica, es decir se muestra siempre una misma cantidad de objetos o se puede definir la cantidad de objetos que se van a mostrar cada vez.
2. Modos de acceso. Puede ser secuencial (para acceder al siguiente, al anterior, al primero y al último bloque) o aleatorio para acceder directamente a cualquier bloque.
3. Circularidad. Permite que el conjunto de bloques lógicos se comporte de manera circular.

Estos patrones pueden ser aplicados a la clase directora, a la relación de navegación, a un índice o a un filtro.

Criterio de Ordenación **Criterio de Ordenación.** Es un patrón que define el orden de la información recuperada en base al valor de uno o más atributos. Puede ser ascendente (asc) o descendente (desc). Puede ser aplicado a cualquier clase de navegación, estructuras de índices (filtros de búsqueda).

Presentación de instancias. Se definen cuatro patrones básicos: *registro*, *tabular*, *maestro-detalle* y *árbol*, son aplicables a la clase directora y las relaciones de navegación. En la Figura 6 se muestran los requisitos de presentación definidos para los contextos de navegación “Consultorio” y “Pacientes Registrados” mostrados en la. Puede observarse que para cada clase relacionada con cada contexto se definen los modelos de presentación, los cuales especifican cómo se mostrará la información al usuario. En el contexto “Consultorio”, tanto a la clase directora (“Medico”) como a la clase complementaria (“ExpedientePaciente”) se les ha especificado un patrón de tipo *registro* (ver Figura 6). Por otro lado, se ha definido que por cada “pantalla” se visualizará un “bloque lógico” con cardinalidad dinámica. En el contexto “Pacientes Registrados”, la información de la clase directora (“ExpedientePaciente”) se mostrará mediante un patrón *tabular* y la información de las clases complementarias (“Recetas”, “Estudios”) se les aplicará un patrón de tipo *registro* (ver Figura 6).

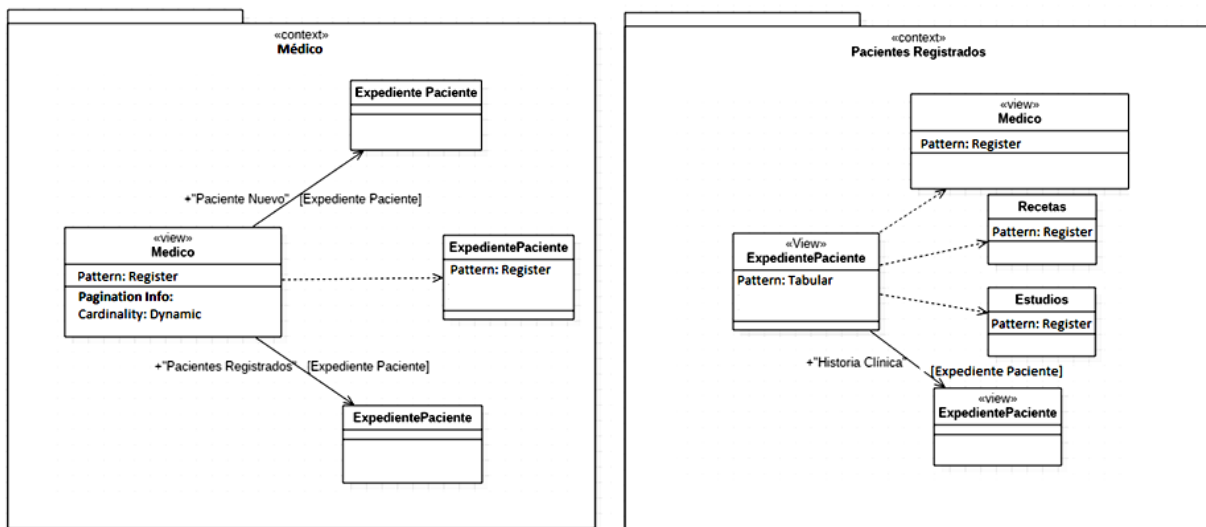


Figura 6. Modelo de Presentación de los contextos “Consultorio” y “Pacientes Registrados”.

El modelo de presentación en combinación con la información contenida en el modelo de navegación, permiten capturar los requisitos básicos para la construcción de las interfaces para cada tipo de usuario en la aplicación Web.

5. CONCLUSIONES

En este trabajo se demostró como la metodología OOWS puede ser usada para construir modelos conceptuales de navegación, específicamente aplicada al caso práctico de una aplicación Web que gestiona el archivo clínico de los pacientes de un consultorio médico.

El beneficio principal del modelo obtenido es que puede aplicarse en ambientes de compilación automática de modelos para generar el código de la aplicación Web. La aplicación práctica de la metodología OOWS a la aplicación Web "Gestión de expedientes clínicos", genera los modelos conceptuales que permiten comprender la estructura funcional de la aplicación y facilitan su mantenimiento, al describir de manera visual su estructura y la relación con las clases que la componen.

Como trabajo futuro se planea depurar y afinar la versión preliminar del modelo de navegación obtenido a fin de realizar su implantación desarrollando la interfaz de usuario de la aplicación Web que gestiona el archivo clínico. También como trabajo futuro se planea aplicar la metodología OOWS para desarrollar modelos conceptuales de navegación de otras aplicaciones web.

6. REFERENCIAS

- Beigbeder, S. M. (2007). WebSA: un método de desarrollo dirigido por modelos de arquitectura para aplicaciones web. (Phd Thesis), Universitat d'Alacant-Universidad de Alicante, Alicante, España.
- Ceri, S. Fraternali, P., Bongio, A., Brambilla M., Comai S., Matera M. (2003). Designing Data-Intensive Web Applications. Morgan Kaufman.
- De Troyer O. (2003) and S. Casteleyn. Modelling Complex Processes from Web applications using WSDM. In IWWOST 2003. Oviedo, Spain. 2003 pp 1-12.
- Fons J. (2003) Development of Web applications from Web enhanced conceptual schemas, I.-Y. Song et al. (Eds.): ER 2003, LNCS 2813, pp. 232–245, Springer-Verlag Berlin Heidelberg 2003.
- Fons J., Pelechano V., Albert M., and Pastor O. (2003) Development of Web Applications from Web Enhanced Conceptual Schemas. In ER 2003, vol. 2813 of LNCS. Springer
- Gómez J., Cachero C., Pastor O. (2000) Extending an Object-Oriented Conceptual Modelling Approach to Web Application Design. June 2000. CAiSE'2000, LNCS 1789, Pags 79-93
- Murugesan, S., Deshpande, Y., Hansen, S. y Ginige, A. (2001) Web Engineering: A New Discipline for Development of Web-Based Systems. En S. Murugesan y Y. Deshpande (Eds.): Web Engineering. Managing Diversity and Complexity of Web Application Development. Lecture Notes in Computer Science. LNCS 2016. Páginas 3-13. Springer Verlag.
- N. Koch. (2000) Software Engineering for Adaptive Hypermedia Applications. PhD thesis, Ludwig-Maximilians-University, Munich, Germany, 2000.
- OMG. (2015). About the Interaction Flow Modeling Language specification version 1.0. Retrieved November 28, 2018
- Pastor O. (2001), An object - oriented approach to automate Web applications development, K. Bauknecht, S.K. Madria, and G. Pernul (Eds.): EC-Web 2001, LNCS 2115, pp. 16–28, Springer-Verlag Berlin Heidelberg 2001.
- Pastor, O., Gomez, J., Insfran, E., and Pelechano, V. (2001), The OO-Method approach for information systems modeling: From object-oriented conceptual modeling to automated programming. Information Systems, 26: 507–534, 2001.
- Quintero R. (2008), Desarrollo dirigido por modelos de aplicaciones Web que integran datos y funcionalidad a partir de servicios Web, Universidad Politécnica de Valencia, España, Tesis Doctoral, Marzo 2008.
- Raoudha Ben Djemaa, Ikram Amous, and Abdelmajid Ben Hamadou. (2006). WA-UML: Towards a UML extension for modelling Adaptive Web Applications. In Proceedings of the Eighth IEEE International Symposium on Web Site Evolution (WSE

- '06). IEEE Computer Society, Washington, DC, USA, 111-117.
- Schwabe D., Rossi G., and Barbosa. S. (1996) Systematic Hypermedia Design with OOHD. In ACM Conference on Hypertext, Washington, USA, 1996.
- Selic, Bran y Bock, Conrad y Cook, Steve y Rivett, Pete y Rutt, Tom y Seidewitz, Ed y Tolbert, Doug. (2015). OMG Unified Modeling Language (Version 2.5).
- Valderas P. (2014), Aplicación de un Método de Modelado de Aplicaciones WEB para el desarrollo de un Portal Web Universitario, Departamento de sistemas informáticos y computación, Universidad Politécnica de Valencia, España, Researchgate, marzo 2014.
- Valverde F. (2007) OOWS Suite: Un entorno de desarrollo para aplicaciones Web basado en MDA, Departamento de sistemas informáticos y computación, Universidad Politécnica de Valencia, España, Researchgate, enero 2007.
- Wright, J. (2007). Towards an Interactive Web Modelling Language. Paper presented at the Proceedings of the 2007 SIENZ Workshop, Auckland, New Zealand.