

EST 2016, 19º Concurso de Trabajos Estudiantiles

## Un caso de estudio sobre la aplicación de UWE para la generación de Sistemas Web

Damián P. Rotta <sup>1a</sup>, Gonzalo S. Pallotta <sup>1a</sup>, Héctor E. Klikailo <sup>1a</sup>, Edgardo A. Belloni <sup>2b</sup>

<sup>1</sup> Estudiantes y <sup>2</sup> Profesor de la Cátedra de Ingeniería de Software  
Carrera de Ingeniería en Informática en Universidad Gastón Dachary

<sup>a</sup>{dprotta, gspallotta, heklikailo}@citic.dachary.edu.ar

<sup>b</sup>ebelloni@ugd.edu.ar

**Resumen.** El surgimiento y adopción de metodologías que promueven la aplicación de un enfoque de desarrollo dirigido por modelos, ciertamente, representa un paso significativo en la evolución de la Ingeniería Web. Con tal enfoque se apunta a reducir el esfuerzo inherente al desarrollo de sistemas Web complejos, así como también, a mejorar la calidad, eficiencia y reusabilidad en tales desarrollos. En este artículo se resume una experiencia, de índole académica y a nivel de estudiantes de grado, que involucra la aplicación de la metodología UML-Based Web Engineering (UWE) en el desarrollo de un Sistema de Gestión de Información de Pacientes (MHC-PMS).

**Palabras Claves:** Ingeniería de Software, Ingeniería Web, MDD, MDWE, UWE.

### 1 Introducción

#### 1.1 Contexto

En este artículo se resume un **trabajo de cátedra** realizado -como parte componente de las obligaciones académicas definidas- para la aprobación de la asignatura Ingeniería de Software; en el contexto de la carrera de grado que cursan actualmente los autores de este documento. Dicho trabajo ha promovido la **realización de actividades de proyecto, diseño e implementación** para el desarrollo de un sistema de software determinado; a partir de la integración y aplicación de contenidos cubiertos en la asignatura. Los **objetivos genéricos definidos y alcanzados** para el trabajo fueron los de:

- Seleccionar uno de los casos de estudio establecidos por la cátedra, identificando y comprendiendo sus características y restricciones de desarrollo esenciales.
- Analizar distintos procesos de desarrollo de software con el fin de identificar ventajas y desventajas de su aplicación al caso de estudio seleccionado.
- Desarrollar el caso seleccionado aplicando el proceso (o combinación de procesos) que resultase(n) más adecuado(s), teniendo en cuenta el análisis realizado previamente; y acordando el alcance del desarrollo e implementación con la cátedra.

## 1.2 Elección del caso de estudio

Particularmente, y en relación al primero de tales objetivos, se pudo optar entre los casos de estudio consignados en [1] y [2]. La selección recayó en el sistema de gestión de información de pacientes para centros de asistencia a la salud mental (MHC-PMS). Dicha selección se sostiene en que representa un sistema con un tamaño relativamente acotado; implicando por lo tanto un desarrollo que podría cumplirse dentro de las restricciones temporales que se nos establecieron. Además, dicha opción implicaba el uso de tecnología Web, y el abordaje de requerimientos no funcionales –como los de seguridad o disponibilidad- no antes enfrentados por el grupo de desarrollo del proyecto.

Otras consideraciones tomadas en cuenta, y que impactarían luego sobre la decisión sobre el tipo de proceso de desarrollo seleccionado, fueron: el grupo de desarrollo del sistema iba a constar de 3 personas; y que eventualmente el sistema debería ser certificable, por ende, se debería contar con una documentación completa del sistema y cumplir con legislaciones vigentes [3].

## 1.3 Elección del proceso de desarrollo

Por otra parte, respecto de los procesos de desarrollo susceptibles de analizar su aplicación al caso de estudio, basados en [4] [5] [6] y los conocimientos previos de los autores de este documento, se consideraron los procesos de desarrollo Iterativos e incrementales como Unified Process (UP), métodos ágiles como eXtreme Programming (XP), Clásicos como Waterfall Model, y Model-driven Web Engineering, como UML-Based Web Engineering (UWE). De esta manera, se identificaron ventajas y desventajas de aplicar tales procesos al desarrollo del caso de estudio seleccionado; teniendo fundamentalmente como criterios de análisis las características y restricciones tanto inherentes al caso como las de nuestro propio equipo de desarrollo.

La comparación resultante del análisis mencionado en la sección anterior permitió sostener que el proceso más adecuado de aplicar al caso era el de MDD, debido a que permitiría la implementación en distintas plataformas con menor esfuerzo, tendría en cuenta la reutilización a altos niveles de abstracción, incluiría mecanismos de trazabilidad que permitirían la adaptación a cambios en los requerimientos con relativa facilidad y aseguraría que la documentación se mantenga actualizada y consistente facilitando la comunicación con stakeholders y posibles entes reguladores. Dentro del marco de MDD y por razones que serán expuestas con más detalle en la sección siguiente, el proceso de desarrollo elegido fue UWE (UML-Based Web Engineering) que también puede enmarcarse dentro de Model Driven-Web Engineering (MDWE).

## 1.4 Organización de este documento

Considerando el contexto expuesto hasta aquí, y en relación a los objetivos señalados para el trabajo, restan describir aún: las especificidades tanto del caso de estudio como del proceso de desarrollo seleccionados; y la aplicación de dicho proceso al caso. Justamente, ese es el propósito específico del resto de este documento, que se organiza como se señala a continuación.

En la siguiente sección se reseñan definiciones y antecedentes relevantes para el trabajo realizado, constituyendo un breve **marco conceptual y metodológico** del mismo.

La sección 3, introduce el **caso de estudio abordado (MHC-PMS)**, describiendo sumariamente su contexto, usuarios, requerimientos funcionales y no funcionales esenciales, así como también, la delimitación de un alcance básico.

Mientras que, en la sección 4, a medida que se describe el **enfoque (UWE) de MDWE aplicado al caso**, se presentan modelos construidos para el desarrollo del mismo. Adicionalmente, se resumen consideraciones de implementación, tanto en relación a la plataforma tecnológica de destino (JSF) como así también, respecto de la herramienta utilizada (UWE4JSF) para la generación automática de código. Además, se señalan consideraciones respecto de la prueba del software desarrollado.

Finalmente, se delinearán **conclusiones** y se introducen **líneas de trabajo futuro**.

## 2 Marco Conceptual y Metodológico

La **Ingeniería Web** ha sido definida como una disciplina emergente que promueve el empleo de enfoques sistemáticos, disciplinados y cuantificables, para lograr el desarrollo eficiente de sistemas y aplicaciones Web con atributos de alta calidad –cf.: [7], [8]. En particular, dicha disciplina científico-tecnológica se enfoca en la proposición, estudio sistemático, experimentación y mejora continua de metodologías, técnicas y herramientas que constituyan el soporte esencial del desarrollo de aplicaciones Web a lo largo de todo su ciclo de vida. De esta manera, abarca procesos de ingeniería de requisitos, diseño arquitectónico y detallado, construcción, evaluación y evolución, que tienen en cuenta características y aspectos que diferencian a las aplicaciones Web de otros tipos de sistemas de información, software o aplicaciones tradicionales –cf.: [9], [10].

Un paso importante en la evolución de la Ingeniería Web lo constituye el surgimiento de metodologías que se fundan en la aplicación de un enfoque ingenieril de desarrollo dirigido o guiado por modelos –**Model-Driven Development (MDD)**. Este enfoque ha sido definido como: “Un paradigma de desarrollo que usa modelos como artefactos principales del proceso de desarrollo. Usualmente, en MDD la implementación es generada (semi)automáticamente desde los modelos” [11]. MDD propone reducir el esfuerzo inherente al desarrollo de aplicaciones y sistemas complejos, así como también, mejorar la calidad y eficiencia de los desarrollos. Para lograr esto, se definen y proveen mecanismos útiles para la especificación de abstracciones relevantes en términos de modelos conceptuales o específicos de dominio –a distintos niveles de detalle, y perspectiva, que permiten tratar de manera separada sus diferentes aspectos, e inicialmente, con independencia de la plataforma de destino del desarrollo. Adicionalmente, mediante transformaciones sistemáticas de estos modelos y de manera (semi)automática, se generan implementaciones específicas para determinadas plataformas tecnológicas.

Concretamente, entre las ventajas de aplicar MDD se pueden citar las señaladas en [12]: incremento de la productividad; mejor adaptación a los cambios tecnológicos y de requisitos; generación consistente de artefactos del desarrollo; reutilización de modelos y transformaciones; mejoras en la comunicación con los usuarios y entre desarrolladores; y la posibilidad de postergar la toma de decisiones de carácter tecnológico.

Teniendo en cuenta que las ventajas que promueve MDD podrían ser aprovechadas en el ámbito del desarrollo Web, surge entonces el concepto de **Model-Driven Web Engineering (MDWE)**. Término el cual refiere a la adopción del paradigma dirigido por modelos en metodologías de desarrollo Web. Dentro de la última década, se han realizado numerosos esfuerzos de investigación y desarrollo en el contexto de MDWE, surgiendo así diferentes propuestas de nuevas metodologías, extensiones de lenguajes de modelado, herramientas y frameworks para facilitar su aplicación, así como también, criterios de evaluación de su calidad –cf.: [6], [13], [14], [15], [16], [17], [18], [19].

En cuanto a las metodologías de MDWE existentes, se consideraron para el trabajo a las siguientes:

- **NDT (Navigational Development Techniques)** [20] es un enfoque metodológico de MDWE que cubre el ciclo de vida completo en el desarrollo de aplicaciones web, incluido la fase de pruebas y control de calidad. Originalmente, este método fue creado solo para ingeniería de requerimientos para el desarrollo de aplicaciones web, gradualmente fue evolucionando hasta convertirse en una metodología.
- **UWE (UML-Based Web Engineering)** [21] es una metodología completamente basada en UML (Unified Modeling Language) que cubre por completo el ciclo de vida de desarrollo de aplicaciones web y RIA (Rich Internet Applications). UWE describe una metodología sistemática usando exclusivamente técnicas, notaciones y extensiones UML.
- **WebML (Web Modeling Language)** [22] es una notación visual y una metodología para el diseño de aplicaciones web, este método provee especificaciones gráficas y formales para el proceso de diseño de aplicaciones web. A partir del 2013 este método (incluyendo sus notaciones) fue extendido enfocándose en el modelado de interfaces de usuario (front-end) y fue propuesto a OMG (Object Management Group) para ser un estándar, llamado IFML (Interaction Flow Modelling Language) [23].

A partir del análisis de las metodologías mencionadas anteriormente, se optó por el uso de UWE debido a que usa una notación gráfica basada por completo en UML definida en un profile que permite trabajar con sus metamodelos en herramientas CASE que incluyan esta posibilidad, reduciendo el tiempo de aprendizaje para los desarrolladores quienes ya se encontraban familiarizados con este lenguaje de modelado.

### 3 Caso de Estudio: MHC-PMS

Para el desarrollo del trabajo, que se resume en este documento, se ha seleccionado como caso de estudio el introducido en [24], denominado “Sistema de gestión de información de pacientes para centros de asistencia a la salud mental” (MHC-PMS) -de aquí en adelante, MHC-PMS.

#### 3.1 Requerimientos del Sistema MHC-PMS

Dicho sistema administra la información de los pacientes y los tratamientos que reciben.

Los usuarios del sistema incluyen personal clínico como médicos, enfermeros y visitadores de salud (enfermeros que visitan a las personas a domicilio para verificar su tratamiento). Los usuarios no médicos incluyen recepcionistas que hacen citas, personal de archivo médico que organiza el sistema de registros y personal administrativo que redacta informes.

El sistema debe registrar información de pacientes como son el nombre, dirección, edad, pariente más cercano, etc., datos sobre las consultas como fecha, médico, impresiones personales del paciente, etc. e información sobre condiciones y tratamientos de los pacientes. Los informes se elaboran a intervalos regulares para el personal médico y los administradores de la autoridad sanitaria.

El diseño global del sistema debe considerar requerimientos de privacidad y seguridad lógica. El sistema tiene que estar disponible cuando se necesite, de otro modo la seguridad de los pacientes estaría comprometida y sería imposible prescribir a los pacientes el medicamento correcto. Aquí existe un conflicto potencial: la privacidad es más fácil de mantener cuando existe sólo una copia de los datos del sistema.

En la figura 1 se muestra la arquitectura que pretende lograr el sistema MHC-PMS, a partir de una base de datos y un servidor centralizado que permite a distintos usuarios acceder a la información almacenada.

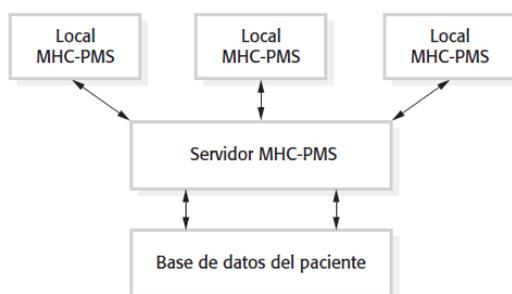


Fig. 1. Organización del MHC-PMS [24]

### 3.2 Alcances pautados para el desarrollo

Se destaca que para la elaboración del trabajo se han relevado una serie de requerimientos, tanto funcionales como no funcionales, de los cuales se pretende introducir una versión parcial en este artículo, por cuestiones de espacio. Concretamente, se desarrollan aquí dos Casos de Uso; cuya descripción se presenta en [25], basados en el esquema presentado por [26]: Registrar Paciente; y Validar usuario.

## 4 Desarrollo: Aplicación de UWE al MHC-PMS

El enfoque metodológico **UML-Based Web Engineering (UWE)** [21] [27] [28] está destinado a proveer soporte en el desarrollo de aplicaciones y sistemas Web con especial foco en la sistematización, personalización y la generación (semi)automática de código. Es un enfoque Orientado a Objetos, iterativo e incremental basado tanto en el Unified Modeling Language (UML) como en el Unified Software Development Process (UP). La

notación utilizada para su diseño se corresponde con un profile “liviano” de UML presentado en [29].

UWE utiliza notación y diagramas UML para el análisis y diseño de aplicaciones Web. Para características Web específicas como nodos y links externos, el profile de UWE incluye estereotipos, tags y restricciones definidas por los elementos del modelado. UWE cubre aspectos de navegación, presentación, procesos de negocios, así como también, aspectos de adaptación.

En la figura 2 se muestran las dependencias que existe entre los distintos tipos de modelos en UWE relación tratada en la sección 4.1 donde se consigna las transformaciones realizadas entre dichos modelos.

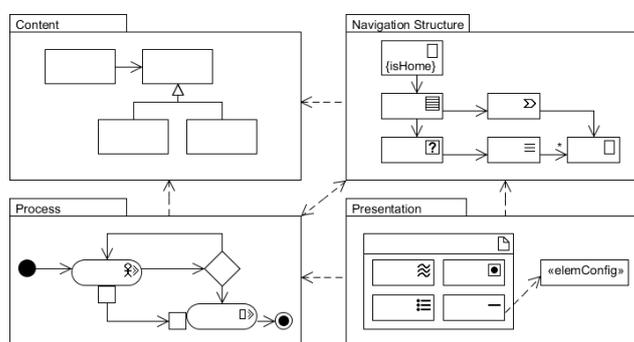


Fig. 2. Vista General de las Transformaciones entre modelos de UWE. [21]

#### 4.1 Modelos y Transformaciones

Para la realización del modelado, se utilizó un plugin llamado MagicUWE [30], que fue desarrollado para ser usado dentro de la herramienta CASE MagicDraw [31]. Dicho plugin incluye el profile para poder realizar las transformaciones entre modelos, que como se propone en [32] consistieron en:

- Modelo de requerimientos a Modelo de Contenido: El procedimiento básico es la conversión de los objetos de los diagramas de actividad en clases.
- Modelo de Contenido a Modelo de Navegación: En principio existirá una nueva clase de navegación por cada clase en el diagrama de contenido, con el mismo nombre y con el estereotipo <<navigationClass>>.
- Modelo de Navegación a Modelo de Presentación: Hay una relación uno a uno entre las clases de ambos modelos, la transformación tendrá en cuenta esta relación generando una clase con el mismo nombre y con el estereotipo <<presentationGroup>>
- Modelo de Navegación a Modelo de Estructura de Procesos: Dibujar cada clase con el estereotipo <<processClass>> en un Diagrama de estructura de procesos. La aparición de una clase nueva sería un error.
- Modelo de Navegación a Modelo de Flujo de Procesos: Crear un nuevo flujo de procesos (diagrama de actividad) por cada processClass con el mismo nombre en el diagrama de navegación.

UWE sigue la idea de la separación de modelos introducida por OOHD (Object Oriented Hypermedia Design Model), aunque propone la inclusión de algunas nuevas características. Su enfoque MDD es quizás uno de los más completos ya que ofrece un metamodelo para cada modelo de UWE (Requerimientos, Contenido, Navegación, Presentación y Procesos), con un conjunto de transformaciones para derivar algunos modelos a partir de otros, definidas en base a QVT como lenguaje estándar [33].

**Modelo de Requerimientos**

Utiliza diagramas de casos de uso de UML. Los estereotipos principales propuestos por el profile de UWE son: <<navigation>> para tareas de navegación y <<webProcess>> para otras tareas. Además, se puede extender este modelo con la realización de Diagramas de Actividades para aportar mayor comprensión acerca del sistema a desarrollar.

En la figura 3 se presenta un diagrama de casos de uso sobre el caso de estudio que pretende especificar el comportamiento del sistema en su interacción con los usuarios mostrando la relación entre los actores y los casos de uso del sistema, ambos caracterizados con estereotipos definidos por UWE.

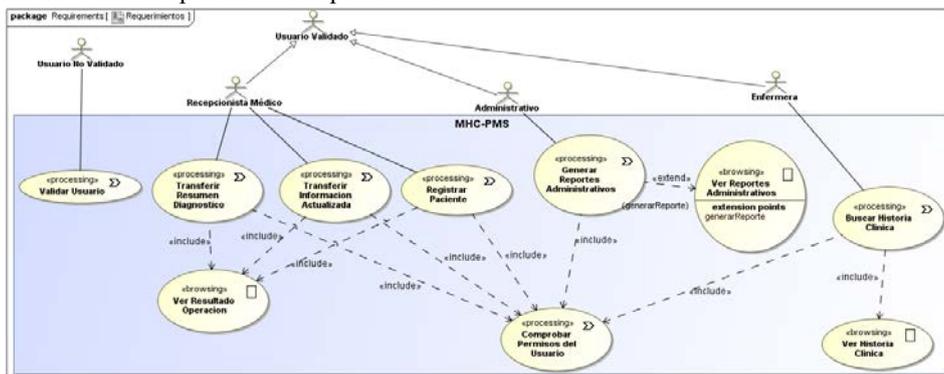


Fig. 3. Modelo de Casos de Uso (UWE): MHC-PMS.

**Modelo de Contenido**

En él se representa la información del dominio, sus datos persistentes, mediante un diagrama de clases UML. En la figura 4 se presenta el modelo de contenido del MHC-PMS mediante un diagrama de clases de UML, donde se puede observar las distintas clases que forman parte del sistema con sus respectivos atributos y estereotipos definido por UWE.

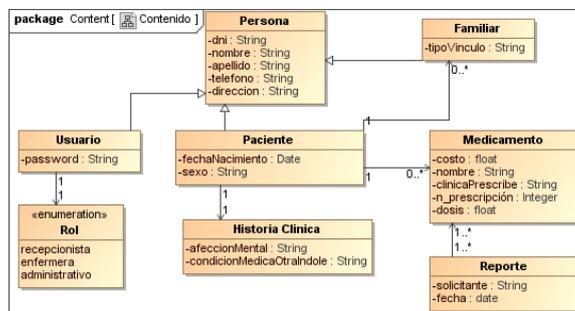


Fig. 4. Modelo de Contenido (UWE): MHC-PMS.

**Modelo de Navegación**

Tiene como objetivo la representación de nodos y enlaces de la estructura de hipertexto, y el diseño de las rutas de navegación mediante diagramas de clases UML.

En la figura 5 se presenta el modelo de navegación del MHC-PMS, donde los distintos nodos, enlaces de la estructura de hipertexto, el diseño de las rutas de navegación y la relación que existe entre los distintos nodos del sistema, se encuentran caracterizados por estereotipos definidos por UWE. Dichos nodos se encuentran agrupados en paquetes para poder clasificarlos según al tipo de gestión del sistema al que pertenece, el cual puede ser de paciente o de usuarios.

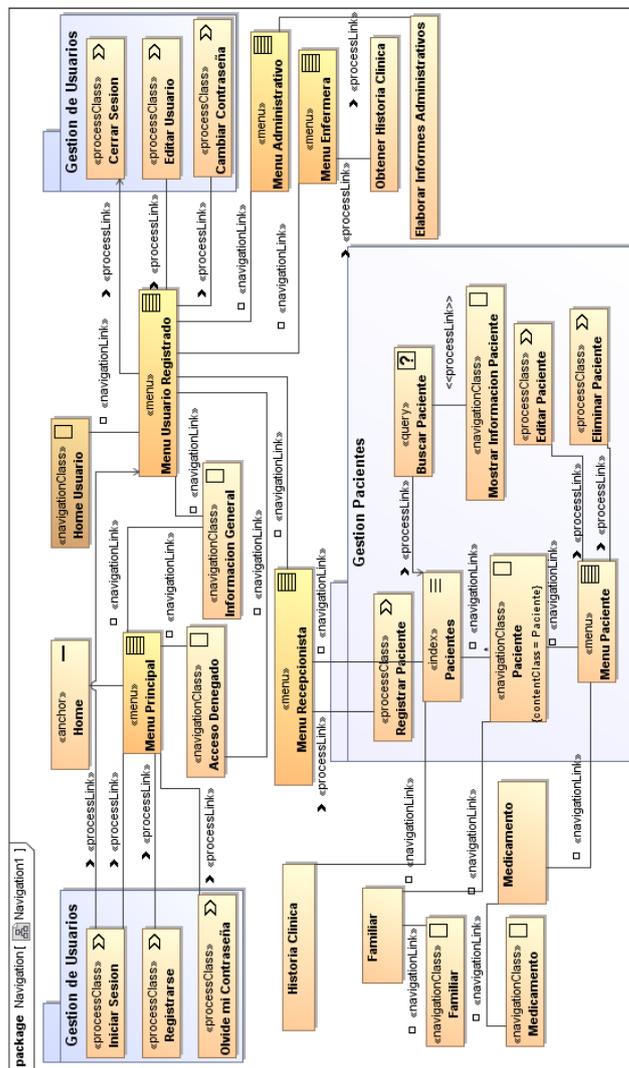


Fig. 5. Modelo de Navegación (UWE): MHC-PMS.

**Modelo de Presentación**

Representa el layout subyacente a los modelos de navegación y procesos mediante una presentación abstracta, ya que una presentación concreta requeriría la especificación de propiedades físicas adicionales. Utiliza un diagrama de clases de UML para modelar la estructura.

En las figuras 6 y 7 se presentan los modelos de presentación para el caso de estudio tratado, donde se pretende mostrar las clases de navegación y de procesos a las que pertenece a cada página web. Se puede observar, también, cómo se relacionan los elementos de distintas páginas, como llegar hasta ellas y los mensajes de error que pueden darse durante la interacción.

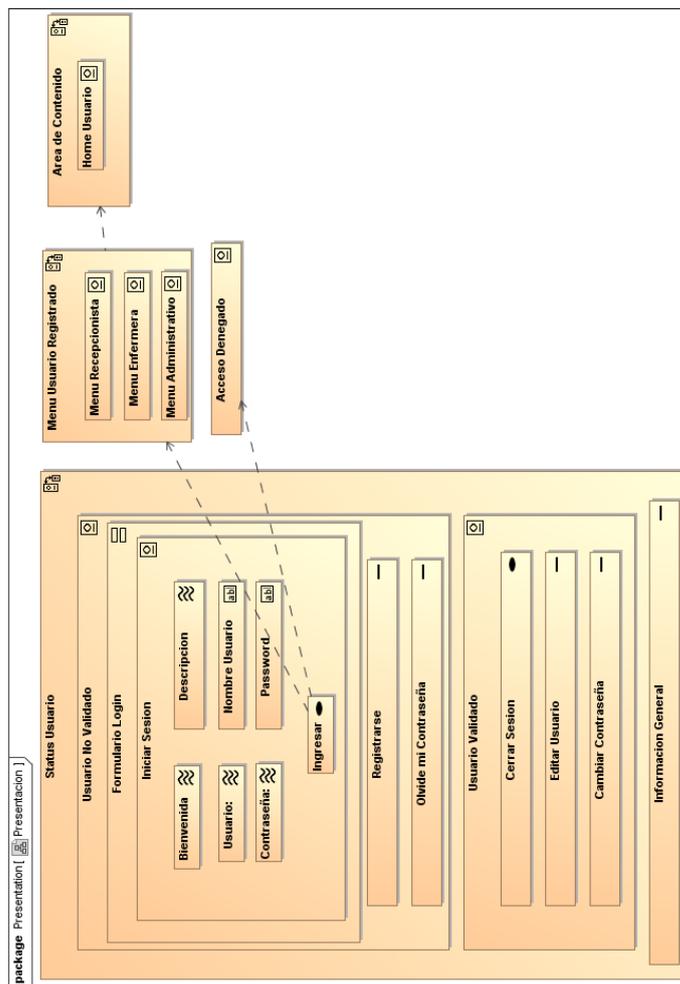


Fig. 6. Modelo de Presentación (UWE): MHC-PMS.

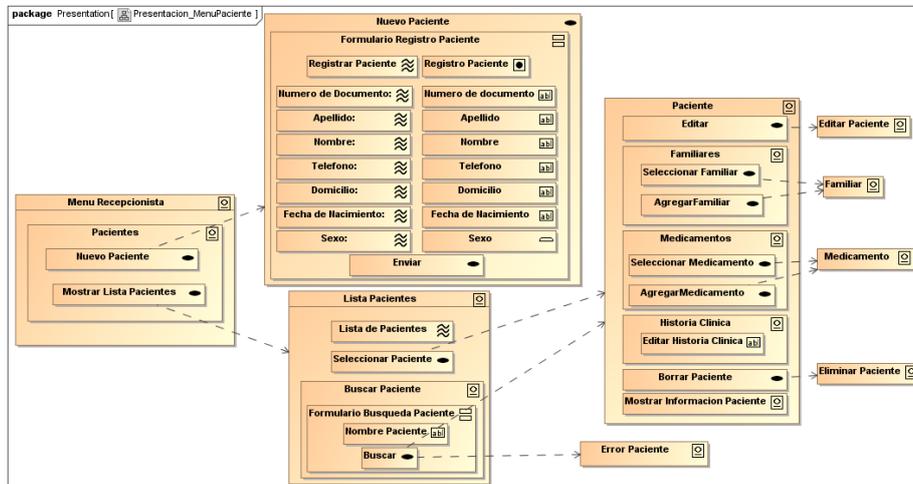


Fig. 7. Modelo de Presentación (UWE): Menú Paciente – MHC-PMS.

### Modelo de Procesos

Representa los aspectos dinámicos de la aplicación Web y especifica funcionalidad cómo transacciones y workflows de actividades. Se modela mediante un diagrama de actividades de UML, y es resultado de refinar el diagrama de actividades modelado durante la especificación de requerimientos. Muestra el flujo de la ejecución representado por nodos de actividad conectados, los nodos de control que proveen constructores de flujo de control como decisiones y sincronización y nodos de objetos que representan el flujo de datos.

En la figura 8 se presenta el modelo de estructura de procesos en donde mediante un diagrama de clases obtenido luego de haber usado la transformación de modelos de navegación a modelos de estructura de procesos, se pretende describir las relaciones entre las distintas «processClass».

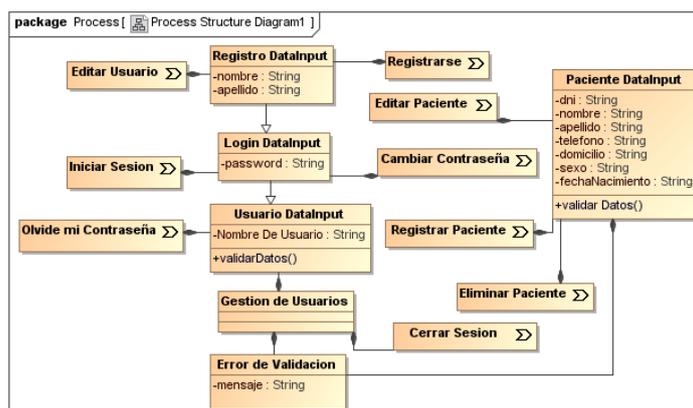


Fig. 8. Modelo de Estructura del Proceso (UWE): MHC-PMS.

Mientras que en las figuras 9 y 10 se presentan los modelos de Flujo del Proceso que especifica las actividades conectadas con cada «processClass» definida en el modelo de estructura de procesos. En este modelo se puede diferenciar entre las actividades realizadas por el usuario y las actividades realizadas por el sistema mediante los estereotipos definidos por UWE.

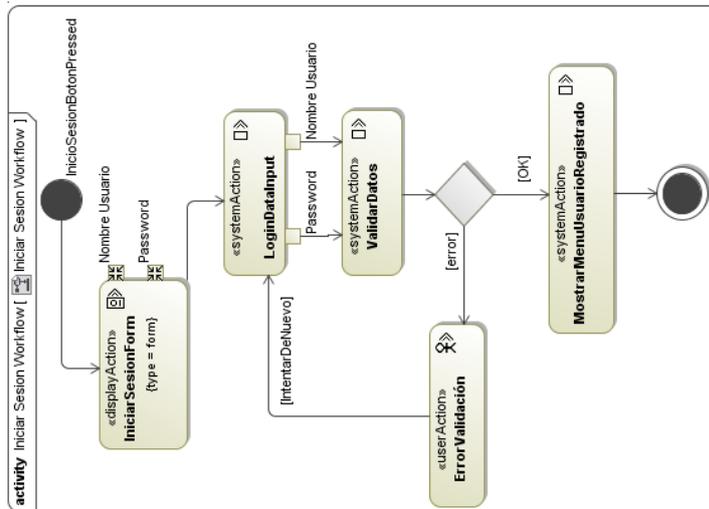


Fig.9. Modelo de Proceso (UWE): Iniciar Sesión MHC-PMS.

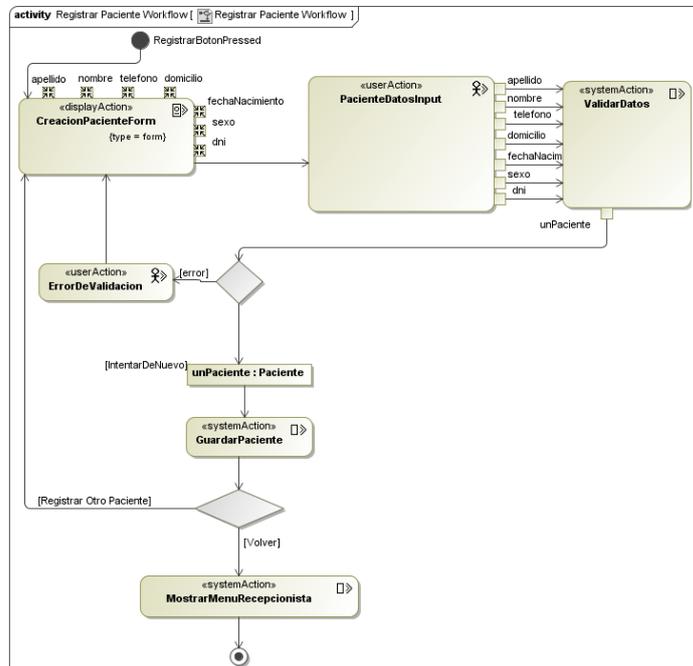


Fig. 10. Modelo de Proceso (UWE): Registrar Paciente MHC-PMS.

### Modelo de Usuario

Representa información específica del usuario y de sesión, permitiendo la personalización, mediante diagramas de clases UML. Se caracteriza por el uso de <<visitClass>> para representar sesiones. En la figura 11 se presenta el modelo de usuario el cual implica la definición de características adaptativas de navegación en función de las preferencias, conocimiento o tareas de usuario. En el modelo se observan atributos y distintas operaciones que puede realizar el usuario del sistema según el rol designado al mismo.

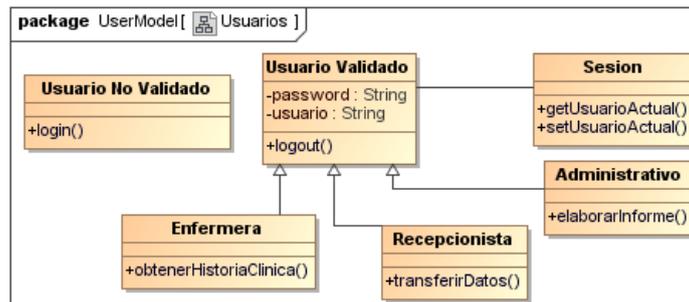


Fig. 11. Modelo de Usuario (UWE): MHC-PMS.

### 4.2 Implementación

UWE propone una herramienta para la obtención automática de código JSF (JavaServer Faces) usando un plug-in para Eclipse denominado UWE4JSF [34], aunque solamente funcional para la versión Galileo [35] del mencionado IDE. Para el modelado, usando esta herramienta se utiliza una versión revisada y extendida del profile de UWE. Siguiendo las transformaciones propuestas, el Modelo de Presentación es mapeado a elementos específicos de la plataforma.

Adicionalmente, se experimentó la derivación manual de modelos a PHP puro [36], y al framework CodeIgniter [37] con el fin de ganar experiencia y comprender cómo podrían definirse transformaciones en JET [38], Acceleo [39] o Xpand [40] según la propuesta de Eclipse Model-to-Text (M2T) [41] para así poder construir modelos dependientes de estas plataformas, y mecanismos de generación de código para ellas.

### 4.3 Prueba del Software

UWE no propone herramientas específicas para la realización de pruebas, pero respecto a la verificación de los modelos recomienda el uso de Hugo/RT [42]. Se optó por la realización manual de casos de prueba que permitan verificar: el login de cada uno de los roles, y el registro de un nuevo paciente y aquellas situaciones alternativas que pueden presentarse en alguno de esos escenarios (usuario y/o contraseña inválida, campos vacíos, usuarios ya registrados, etc.), estos se presentan en [43] basándose en un estándar de IEEE [44].

## 5 Conclusiones

En este artículo se ha presentado un trabajo que involucra el desarrollo (parcial) de un Sistema de gestión de información de pacientes para centros de asistencia a la salud mental (MHC-PMS). Para esto se ha aplicado un enfoque ingenieril de desarrollo Web dirigido por modelos: el proceso que prescribe UML-Based Web Engineering (UWE).

A partir de la experimentación con dicho enfoque metodológico se pudieron observar las ventajas de aplicar un enfoque sustentado en MDD, como ser: la transformación (semi)automática entre modelos; un aumento de compatibilidad y portabilidad al utilizar modelos estándares; un incremento de la productividad; y muchas otras relacionadas con tener el sistema modelado desde diferentes vistas, algunas de las cuales gracias a su alto nivel de abstracción facilitan la comunicación con los stakeholders. Entre otras ventajas observables se pueden citar también, ya como propias de UWE: la reducción de la curva de aprendizaje gracias a la utilización de modelos estándares y la disponibilidad de vasta documentación disponible. No obstante, fue considerada la necesidad de herramientas para transformaciones modelo a texto para tecnologías distintas a JSF, así como también, la de enfrentar algunos problemas que surgen con la interoperabilidad entre herramientas provistas.

El principal requerimiento no funcional con el que se debió cumplir dadas las características del caso de estudio era el de la seguridad lógica. Si bien UWE no pone especial consideración en ello, se puede mencionar que es un foco de interés relativamente reciente, presentándose trabajos –cf.: [45], que proponen una técnica orientada a aspectos para modelar el control de acceso en aplicaciones Web basado en el modelo de navegación de UWE. Además, en [46] se trata el mismo aspecto de autenticación pero sumando un login tracer para lidiar con múltiples intentos de login fallidos.

Por otra parte, más allá de haberse cumplido con los objetivos del trabajo, la experiencia realizada con UWE ha resultado ser tanto formativa como motivadora, planteando nuevos desafíos a enfrentar en el mediano plazo. Entre los desafíos emergentes, se señala el de extender el enfoque ingenieril aplicado para permitir la generación de código ejecutable en otras plataformas tecnológicas Web, distintas a las provistas actualmente por UWE. Este planteo se funda en la convicción de que puede lograrse una mayor adopción de metodologías, en el contexto de MDWE, en la medida que se brinde soporte a plataformas de destino avanzadas y de amplia aplicación por parte de desarrolladores Web. Asimismo, tal posibilidad de extensión del enfoque aplicado resalta una de las ventajas de su adopción; el hecho de poder reutilizar modelos del dominio portando (semi)automáticamente sistemas Web complejos a otras plataformas, cuando aquella elegida originalmente para su implementación se considere ya obsoleta.

En particular, para las extensiones a explorar se considerará a priori a frameworks de desarrollo basados en PHP, como los de CodeIgniter [37], Laravel [47] y/o Yii [48], teniendo en cuenta su amplia utilización. Dichas extensiones implican la definición de modelos de diseño, y sus transformaciones a código, específicos para tales plataformas. Para esto resulta necesario compilar las abstracciones relevantes en las mismas. Adicionalmente, a partir de la construcción de dichos modelos dependientes de plataformas PHP específicas, se avizora la posibilidad de definir también un modelo de diseño intermedio e independiente que abstraiga características comunes entre las plataformas cubiertas; lo que constituiría un modelo MVC Web para frameworks PHP.

## Referencias

- [1] «Ian Sommerville's book website,» [En línea]. Available: <https://ifs.host.cs.st-andrews.ac.uk/Books/SE9/CaseStudies/index.html>.
- [2] «Ian Sommerville's book website,» [En línea]. Available: <http://iansommerville.com/software-engineering-book/case-studies/>.
- [3] *Ley Nacional N° 26.529: Derechos del paciente, historia clínica y consentimiento informado*, Argentina.
- [4] I. Sommerville, *Software Engineering*, 10th ed., Pearson, 2015.
- [5] M. L. Despa, «Comparative study on software development methodologies,» vol. 5, p. 3, 2014.
- [6] G. Aragón, M. J. Escalona, M. Lang y J. R. Hilera, «An analysis of model-driven web engineering methodologies,» *International Journal of Innovative Computing, Information and Control*, vol. 9, n° 1, pp. 413-436, 2013.
- [7] S. Murugesan, Y. Deshpande, S. Hansen y A. Ginige, «Web Engineering: A New Discipline for Development of Web-based Systems,» de *Web Engineering*, Springer Berlin Heidelberg, 2001, pp. 3-13.
- [8] I. S. f. W. E. (ISWE), «eb Engineering Community Portal,» IT-Management and Web Engineering research group at the University of Karlsruhe, [En línea]. Available: <http://www.webengineering.org/>. [Último acceso: Mayo 2016].
- [9] A. Ginige y S. Murugesan, «Web engineering: an introduction,» *MultiMedia, IEEE*, vol. 8, n° 1, pp. 14-18, Enero-Marzo 2001.
- [10] R. S. Pressman y D. Lowe, *Web Engineering: A Practitioner's Approach*, 1st ed., McGraw-Hill, 2009.
- [11] M. Brambilla, J. Cabot y M. Wimmer, *Model-Driven Software Engineering in Practice*, 1st ed., Morgan & Claypool, 2012.
- [12] C. Pons, R. Giandini y G. Pérez, *Desarrollo de software dirigido por modelos: Conceptos teóricos y su aplicación práctica*, 1ra ed., La Plata, Buenos Aires: Universidad Nacional de La Plata, 2010.
- [13] F. Domínguez-Mayo, M. J. Escalona, M. Mejías, M. Ross y G. Staples, «Quality evaluation for model-driven web engineering methodologies,» *Information and Software Technology*, vol. 54, n° 11, pp. 1265-1282, 2012.
- [14] M. A. O. Mukhtar, M. Fadzil Bin Hassan, J. Bin Jaafar y L. Ab Rahim, «WSDMDA: An Enhanced Model Driven Web Engineering Methodology,» de *2014 IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, IEEE, 2014, pp. 484-489.
- [15] A. Fatolahi, S. S. Some y T. C. Lethbridge, «Model-driven web development for multiple platforms,» *Journal of Web Engineering*, vol. 10, n° 2, pp. 109-152, 2011.

- [16] J. A. Aguilar, A. Zaldívar-Colado, C. Tripp-Barba, A. Ocegueda, O. V. García, J. A. A. Brito y E. Zurita, «Techniques and Tools for Web Requirements in NDT, UWE and,» *Journal of Computer Science Technology Updates*, vol. 2, nº 1, pp. 25-31, 2015.
- [17] N. Koch, A. Knapp, G. Zhang y H. Baumeister, «Uml-Based Web Engineering: An Approach Based on Standards,» de *Web Engineering: Modelling and Implementing Web Applications*, G. Rossi, O. Pastor, D. Schwabe y L. Olsina, Edits., London, Springer London, 2008, pp. 157-191.
- [18] M. Brambilla, S. Comai, P. Fraternali y M. Matera, «Designing web applications with webml and webratio,» de *Web Engineering: Modelling and Implementing Web Applications*, Springer, 2008, pp. 221-261.
- [19] M. Brambilla y P. Fraternali, *Interaction Flow Modeling Language: Model-Driven UI Engineering of Web and Mobile Apps with IFML*, 1st ed., Elsevier, 2015.
- [20] I. W. y. T. T. (IWT2), «IWT2,» [En línea]. Available: <http://iwt2.org/actividad-grupo/investigacion/resultados/ndt/>.
- [21] L. -. L.-M.-U. München, «UWE – UML-based Web Engineering,» Research Unit of Programming and Software Engineering, [En línea]. Available: <http://uwe.pst.ifi.lmu.de/>. [Último acceso: Mayo 2016].
- [22] «The WebML Modeling Language,» [En línea]. Available: <http://www.webml.org/webml/page1.do>.
- [23] «IFML: The Interaction Flow Modeling Language,» [En línea]. Available: <http://www.ifml.org/>.
- [24] I. Sommerville, *Ingeniería de Software*, 9na ed., L. M. Cruz Castillo, Ed., Naucalpan de Juárez, Estado de México: Pearson Educación de México, 2011.
- [25] *Anexo A*.
- [26] C. Larman, *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development*, Prentice Hall, 2001.
- [27] N. Koch y A. Kraus, «The expressive power of UML-Based Web Engineering,» de *Second International Workshop on Web-oriented Software Technology (IWWOST02)*, vol. 16, 2002, pp. 105-119.
- [28] N. Koch, A. Kraus y R. Hennicker, «The authoring process of the UML-Based web engineering approach,» de *First International Workshop on Web-Oriented Software Technology*, Valencia, 2001.
- [29] C. Kroiß y N. Koch, «UWE Metamodel and Profile - User Guide and Reference,» LMU Technical Report 0802, München, 2008.
- [30] «MagicUWE,» [En línea]. Available: <http://uwe.pst.ifi.lmu.de/toolMagicUWE.html>.
- [31] «MagicDraw,» [En línea]. Available: <http://www.nomagic.com/products/magicdraw.html>.

- [32] «MagicUWE Transformations,» [En línea]. Available: <http://uwe.pst.ifi.lmu.de/toolMagicUWEReferenceV1.3.html#transformations>.
- [33] N. Koch, «Transformation techniques in the model-driven development process of UWE,» de *Workshop proceedings of the sixth international conference on Web engineering*, 2006.
- [34] L. -. L.-M.-U. München, «UWE – UML-based Web Engineering,» [En línea]. Available: <http://uwe.pst.ifi.lmu.de/toolUWE4JSF.html>.
- [35] Eclipse. [En línea]. Available: <https://eclipse.org/galileo/>.
- [36] T. P. Group, «PHP,» [En línea]. Available: <http://php.net/>.
- [37] B. C. I. o. Technology, «CodeIgniter,» [En línea]. Available: <http://www.codeigniter.com/>.
- [38] «JET,» [En línea]. Available: <https://eclipse.org/modeling/m2t/?project=jet#jet>.
- [39] «Acceleo,» [En línea]. Available: <http://www.eclipse.org/acceleo/>.
- [40] «Xpand,» [En línea]. Available: <https://eclipse.org/modeling/m2t/?project=xpand#xpand>.
- [41] «Eclipse M2T,» [En línea]. Available: <https://eclipse.org/modeling/m2t/>.
- [42] U. Augsburg. [En línea]. Available: <https://www.informatik.uni-augsburg.de/en/chairs/swt/sse/hugort/>.
- [43] *Anexo B*.
- [44] IEEE, «IEEE Standard for Software and System Test Documentation (829-1998),» de *IEEE Std 829-2008*, IEEE, 2008, pp. 1-150.
- [45] B. Marianne, «Integration of Security Aspects in Web Engineering,» München, Germany, 2011.
- [46] D. Pramod , «Modeling security aspects in model driven web application development,» *International Journal Of Computer Science And Applications*, vol. 1, n° 1, pp. 67-70, Junio 2008.
- [47] T. Otwell, «Laravel,» [En línea]. Available: <https://laravel.com/>.
- [48] Y. S. LLC, «Yii Framework,» [En línea]. Available: <http://www.yiiframework.com/>.