Implementación de la metodología OOHDM en el desarrollo del sistema web SIREG.

Implementation of the OOHDM methodology in the development of the SIREG web system.

Gerardo Aguilar Sámano* (1).
Estudiante, Tecnológico Nacional de México, I. T. de Acapulco.
aguilarsamano@gmail.com.

Jorge Carranza Gómez (2). Tecnológico Nacional de México, I. T. de Acapulco, jorge.cg@acapulco.tecnm.mx.

Juan Miguel Hernández Bravo (3). Tecnológico Nacional de México, I. T. de Acapulco,
juan.hb@acapulco.tecnm.mx.

José Antonio Montero Valverde (4). Tecnológico Nacional de México, I. T. de Acapulco,
jose.mv@acapulco.tecnm.mx.

*corresponding author.

Artículo recibido en enero 12, 2021; aceptado en febrero 26, 2021.

Resumen.

El siguiente artículo presenta la implementación de la metodología de desarrollo de aplicaciones Web OOHDM (Método de diseño hipermedia orientado a objetos) en el desarrollo del SIREG (Sistema de Regularización y Escrituración de Predios), el cual permitirá al Instituto de Vivienda y Suelo Urbano de Guerrero llevar un control de los procesos administrativos necesarios para la regularización y escrituración de predios pertenecientes a su patrimonio.

Palabras clave: OOHDM, Aplicación Web, UML, Diseño de Software.

Abstract.

The following article presents the implementation of the Object-Oriented Hypermedia Design Method Web application development methodology (OOHDM) in the development of the SIREG (Land Regularization and Deed System), which will allow the Guerrero Institute of Housing and Urban Land to keep a control of the necessary administrative processes for the regularization and deed of properties belonging to its patrimony.

Keywords: OOHDM, Web Application, UML, Software Design.

1. Introducción.

En el ciclo de vida del software la implementación de una metodología de diseño Web es de suma importancia, ya que esto permite modelar cada una de las etapas con diagramas estandarizados que nos ayudan en principio a documentar el sistema y, por otro lado, sentar las bases para futuras actualizaciones.

Si bien el auge de las metodologías ágiles en la actualidad ha revolucionado el diseño y desarrollo de software, existe en ellas un vacío sustancial en cuanto al diseño de software se refiere. En el caso específico de *Scrum*, una vez terminado el *Sprint Backlog*, no se cuenta con especificaciones para realizar modelos o diagramas que le permitan al equipo de desarrollo interpretar con exactitud los requerimientos obtenidos anteriormente. Schwabe y Rossi (1998) señalan que: "Las metodologías tradicionales de ingeniería de software, o las metodologías para desarrollar sistemas de información utilizando bases de datos, no contienen abstracciones útiles capaces de facilitar la tarea de especificar aplicaciones que encarnen la metáfora del hipertexto. Por ejemplo, no proporcionan ninguna noción de vinculación y se dice muy poco sobre cómo incorporar hipertexto en la interfaz. Además, a medida que aumenta el tamaño, la complejidad y la cantidad de aplicaciones, se necesita un enfoque sistemático que ayude a lidiar con la complejidad y permita la evolución y la reutilización del conocimiento de diseño previamente recopilado" (Schwabe & Rossi, 1998, págs. 1-2).

El objetivo principal de contar con una metodología de diseño clara y con procedimientos concretos, es facilitar la comunicación entre el cliente y el equipo de desarrollo. Un buen diseño de software se traduce en aplicaciones Web fiables, robustas y con capacidad de mejora continua.

Si bien, existen distintas metodologías de diseño Web, es pertinente señalar que OOHDM de acuerdo al análisis realizado por Molina Ríos, Zea Ordoñez, Contento Segarra y García Zerda, se consolida como la metodología de mayor implementación debido a su fácil adaptabilidad en todo proyecto. (Molina Ríos, Zea Ordoñez, Contento Segarra, & García Zerda, 2017, pág. 17)

Una vez señalado lo anterior y para realizar el diseño del Sistema SIREG (Aguilar Sámano, Carranza Gómez, Hernández Bravo, & Montero Valverde, 2020), se utilizará la metodología de diseño Web OOHDM que a continuación se detalla.

2. Metodología OOHDM.

La metodología *Object Oriented Hypermedia Design Method* (OOHDM) propuesta por Daniel Schwabe y Gustavo Rossi, cuyas siglas en español significan Método de Diseño Hipermedia Orientado a Objetos, es una extensión de HDM (*Hypermedia Design Method*). Sus características principales son:

- Está basada en el paradigma de la orientación a objetos.
- Utiliza la nomenclatura propuesta por UML (Lenguaje Unificado de Modelado, por sus siglas en inglés)
- Propone un proceso predeterminado para el que indica las actividades a realizar y los productos que se deben obtener en cada fase del desarrollo.

2.1 Fases de la Metodología OOHDM.

En su planteamiento inicial, Schwabe y Rossi indican que OOHDM se compone de cuatro actividades diferentes: diseño conceptual, diseño de navegación, diseño de interfaz abstracta e implementación.

"OOHDM comprende cuatro actividades diferentes, a saber, diseño conceptual, diseño de navegación, diseño e implementación de interfaz abstracta. Se realizan en una combinación de estilos de desarrollo incrementales, iterativos y basados en prototipos. Durante cada actividad, se crea o enriquece un conjunto de modelos orientados a objetos que describen preocupaciones de diseño particulares a partir de iteraciones anteriores" (Schwabe & Rossi, 1998, pág. 1).

Sin embargo, a través del tiempo, diversos autores señalan la necesidad de incluir una actividad adicional inicial: la obtención de requerimientos (ver Figura 1).

Figura 1. Fases de OOHDM. Elaboración propia.

2.1.1 Obtención de requerimientos. En esta fase se identifican los actores y las tareas o funciones que cada uno podrá realizar. Posteriormente se identifica la información que los actores deberán proveer al sistema, así como el resultado que esperan obtener del mismo. El artefacto con el que se diseña esta fase son los diagramas de casos de uso.
De acuerdo con Rumbaugh, Jacobson y Booch (2007), el objetivo de los diagramas de casos de uso es "Modelar la funcionalidad de un sistema tal como lo perciben los agentes externos, denominados actores, que interactúan con el sistema desde un punto de vista particular. Un caso de uso es una unidad de funcionalidad expresada como una transacción entre los actores y el sistema" (Rumbaugh, Jacobson, & Booch, 2007, pág. 31).

2.1.2 Diseño conceptual. En esta fase se construye el modelo orientado a objetos que represente el dominio de la aplicación y el resultado de esta fase nos arroja el modelo de clases con sus relaciones. El artefacto con el que se diseña esta fase es el diagrama de clases.

"En OOHDM, el esquema conceptual está construido por clases, relaciones y subsistemas. Las clases son descritas como en los modelos orientados a objetos tradicionales. Sin embargo, los atributos pueden ser de múltiples tipos para representar perspectivas diferentes de las mismas entidades del mundo real" (Silva & Mercerat, 2001, pág. 4).

2.1.3 Diseño Navegacional. En esta fase se debe diseñar la aplicación tomando en cuenta las tareas que el usuario va a realizar sobre el sistema, lo anterior basado en el modelo de clases generado en la fase anterior. Como resultado se obtendrán las clases navegacionales que se componen de enlaces (que se derivan de las relaciones), nodos (que representan las vistas), y las estructuras de acceso. En esta fase se pueden utilizar técnicas de modelado orientadas a objetos, así como patrones de diseño.

Ya que no existe un diagrama UML específico para la implementación de esta fase, para la construcción del diagrama de clases navegacionales es factible utilizar un diagrama similar al implementado en la fase anterior, pero haciendo uso de las extensiones Web para UML (WAE), establecidas por Jim Conallen (1999): "Una extensión de UML se expresa en términos de estereotipos, valores etiquetados y restricciones. Combinados, estos mecanismos nos permiten extender la notación de UML, lo que nos permite crear nuevos tipos de bloques de construcción que podemos usar en el modelo. El estereotipo, una extensión del vocabulario del lenguaje, nos permite adjuntar un nuevo significado semántico a un elemento modelo. Los estereotipos se pueden aplicar a casi todos los elementos del modelo y generalmente se representan como una cadena entre un par de comillas angulares: «». Sin embargo, también se pueden representar con un nuevo icono" (Conallen, 1999, pág. 4).

Por otra parte, el diagrama de contexto navegacional, supone el camino o ruta que determinado actor puede realizar para cumplir una acción en particular. Esto se puede diseñar en forma particular para cada actor de acuerdo a los permisos que se le otorguen, o de forma general indicando las distintas rutas existentes en todo el sistema Web.

2.1.4 Diseño de Interfaz Abstracta. Esta fase comprende la definición de objetos que serán perceptibles para el usuario. La separación entre el diseño navegacional y el diseño de interfaz abstracta permitirá construir diferentes interfaces para el mismo modelo navegacional. En esta fase se utilizan los Modelos de Vistas Abstractas de Datos, cuya función es representar las entradas y salidas de la interfaz que interactúan con el usuario, también llamados ADV

por sus siglas en inglés (*Abstract Data View*), cuya función es representar las entradas y salidas de la interfaz que interactúan con el usuario.

"Los ADV son objetos en el sentido de que tienen un estado y una interfaz, donde la interfaz se puede ejercitar a través de mensajes (en particular, eventos externos generados por el usuario). Los ADV son abstractos en el sentido de que solo representan la interfaz y el estado, y no la implementación" (Schwabe & Rossi, Developing Hypermedia Applications using OOHDM, 1998, pág. 14).

2.1.5 Implementación. Una vez concluidas las fases mencionadas anteriormente, únicamente es necesario llevar los objetos a un lenguaje concreto de programación, para obtener así la implementación ejecutable de la aplicación. Si bien los ADV nos facilitan la interpretación de la idea que el *Product Owner* desea como producto final, no tienen que ser necesariamente una representación exacta de los mismos. La interfaz real se tiene que adaptar a los elementos de navegación que se desarrollan con la finalidad de que la aplicación sea lo más intuitiva posible para el usuario final.

3. Desarrollo.

3.1 Diagrama de Casos de Uso.

De acuerdo a lo establecido en la primera fase de la metodología descrita en el punto 2.1.1, una vez identificados los actores y las funciones que realizarán en el sistema, se procede a la elaboración del diagrama de casos de uso (ver Figura 2).

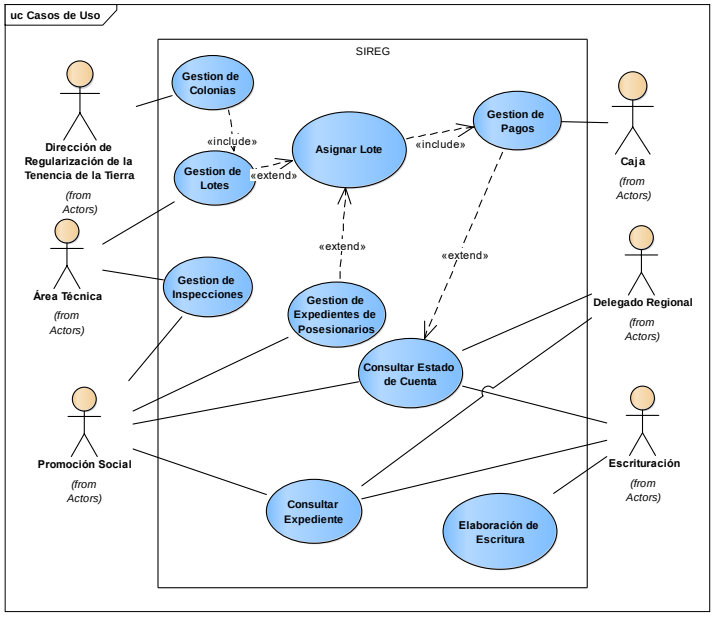


Figura 2. Diagrama de Casos de Uso. Elaboración propia.

3.2 Diagrama de clases.

Una vez elaborado el diagrama de casos de uso, se construye el modelo orientado a objetos que represente el dominio de la aplicación y el resultado de esta fase nos arroja el modelo de clases con sus atributos y relaciones (ver Figura 3).

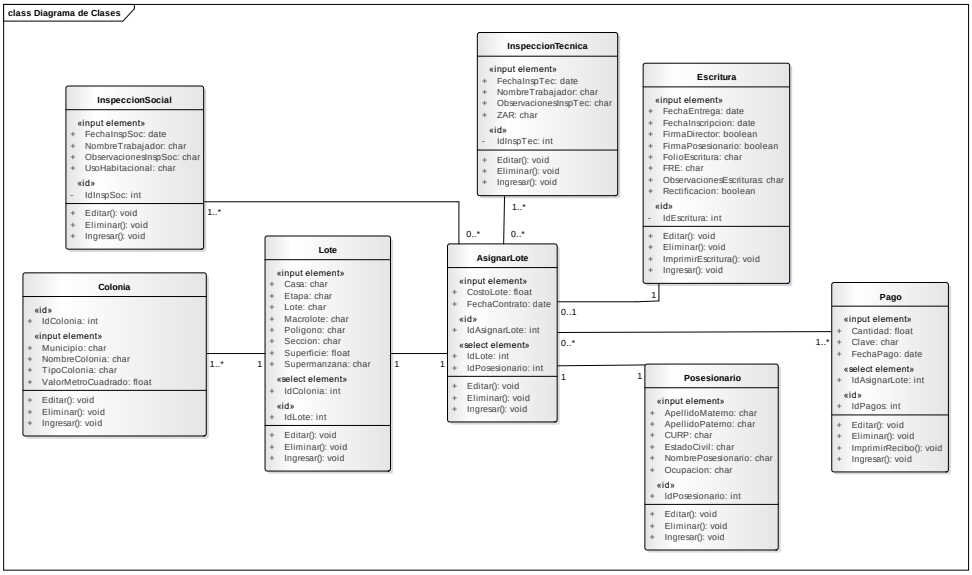


Figura 3. Diagrama de clases. Elaboración propia.

3.3 Diagrama de clases navegacionales y diagrama de contexto navegacional.

La construcción de estos diagramas supone un primer acercamiento de la aplicación con la interacción que tendrá con los usuarios finales. Por lo que sintetizan lo plasmado en los diagramas anteriores hacia un nivel de abstracción intermedio. El diagrama de clases navegacionales presenta un mapa general de la aplicación que, con ayuda de las extensiones Web para UML, muestra todos los caminos posibles de la misma (ver Figura 4). Mientras que el diagrama de contexto navegacional muestra las restricciones que tendrán los usuarios de acuerdo al rol que se les establezca al inicio de sesión (ver Figura 5).

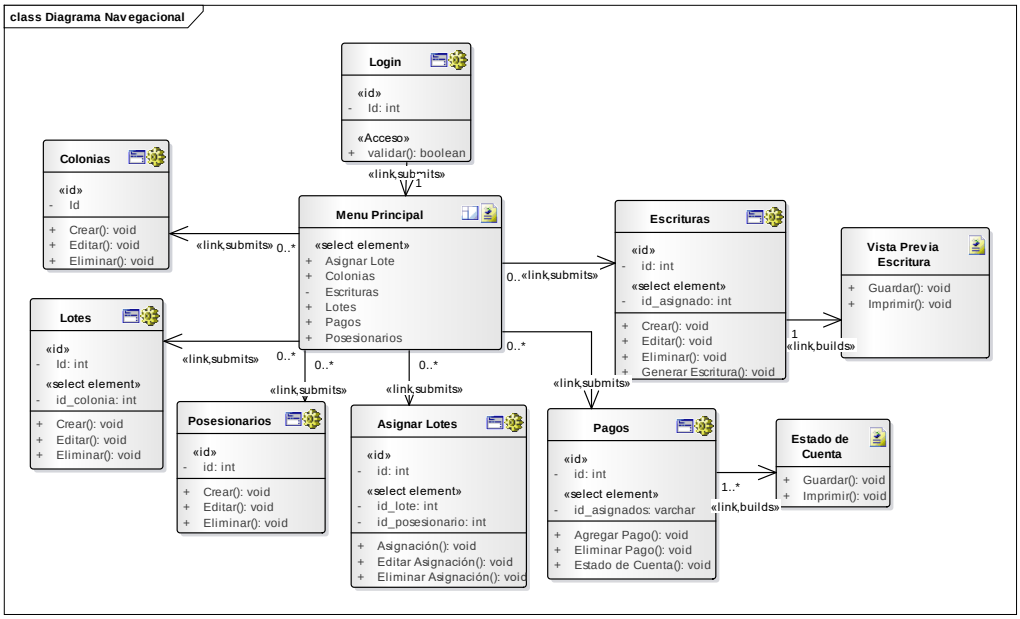


Figura 4 Diagrama de clases navegacionales. Elaboración Propia

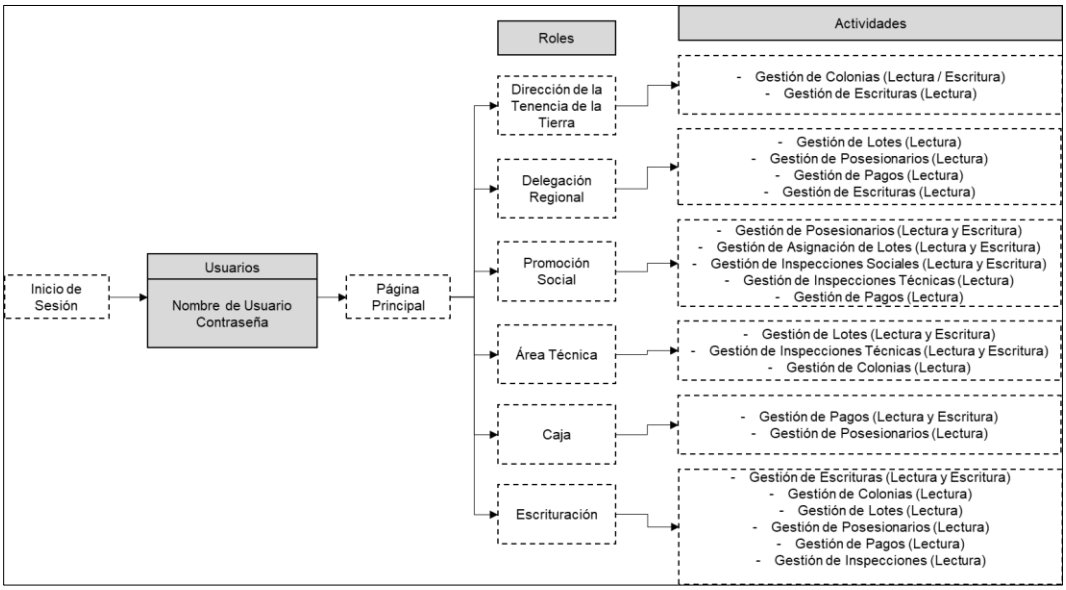


Figura 5. Diagrama de contexto navegacional. Elaboración propia.

3.4 Diagramas de interfaz abstracta.

El diseño de los ADV permite elaborar bocetos de la interfaz gráfica perceptible para el usuario. Se pueden elaborar usando lápiz y papel, software de diseño gráfico o hasta sofisticadas herramientas CASE (*Computer Aided Software Engineering*, Ingeniería de Software Asistida por Computadora). Sin embargo, su finalidad primordial es la de mostrar al cliente vistas preliminares de la aplicación que le permitan realizar los cambios necesarios antes de que el equipo de desarrollo las codifique en un lenguaje de programación (ver Figuras 6, 7 y 8).

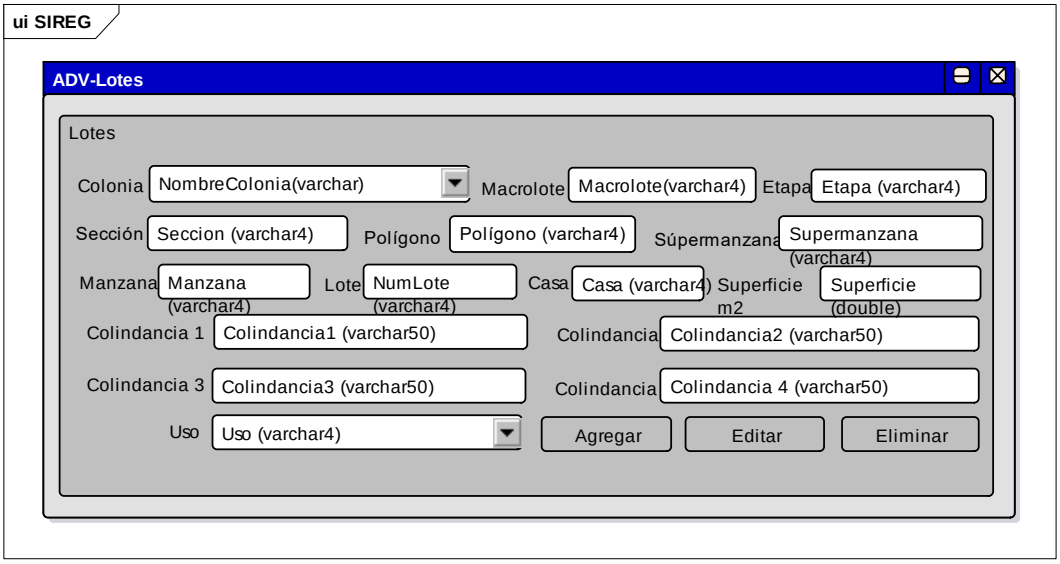


Figura 6. Diagrama de interfaz abstracta del módulo Gestión de Lotes. Elaboración propia.

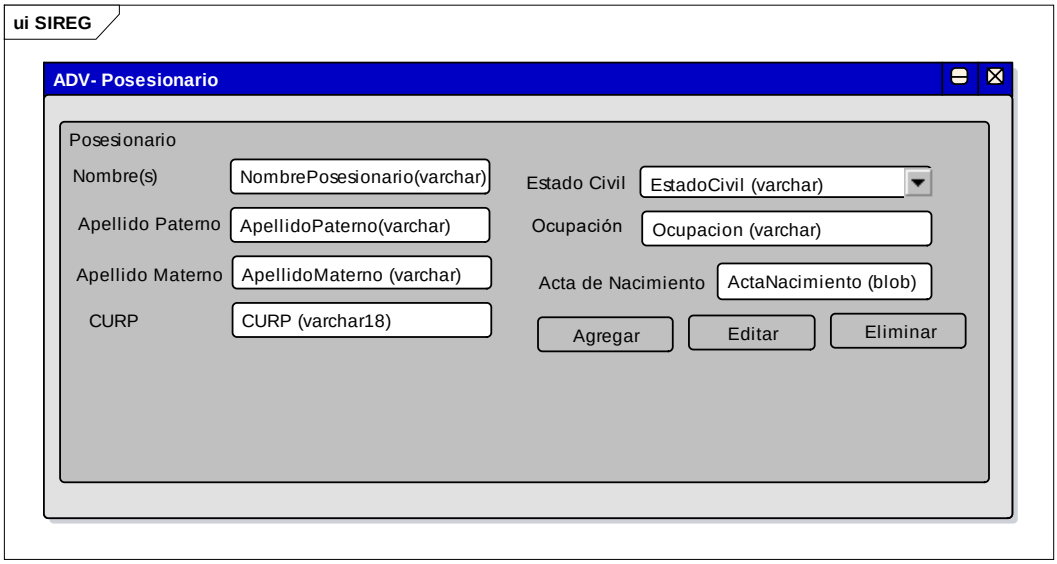


Figura 7. Diagrama de interfaz abstracta del módulo Gestión de Posesionarios. Elaboración propia.

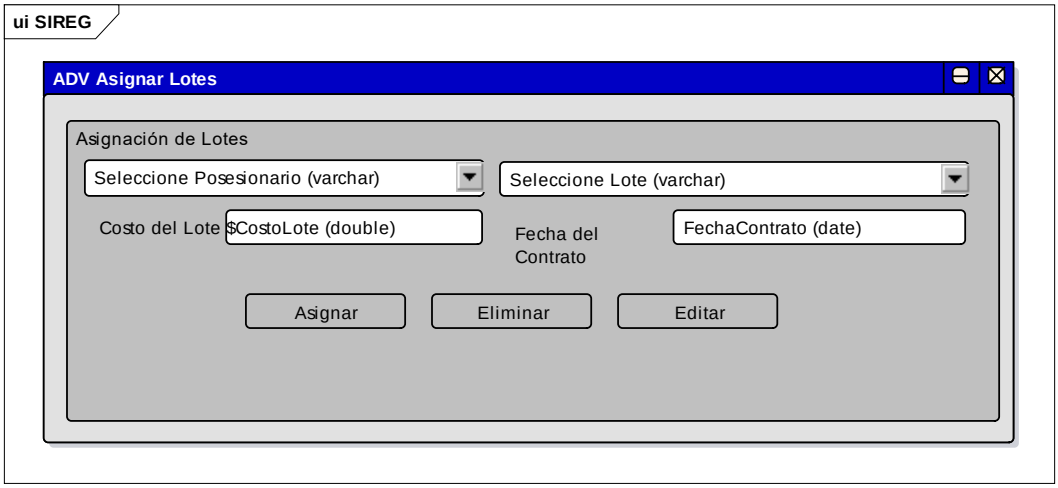


Figura 8. Diagrama de interfaz abstracta del módulo Asignar Lotes. Elaboración Propia.

3.5 Implementación.

Una vez concluidas las fases mencionadas anteriormente, únicamente es necesario llevar los objetos a un lenguaje concreto de programación, para obtener así la implementación ejecutable de la aplicación. Para la implementación se utilizó el lenguaje PHP a través del *framework* Laravel, así como el *framework* Bootstrap para la interfaz gráfica (ver Figuras 9, 10 y 11).

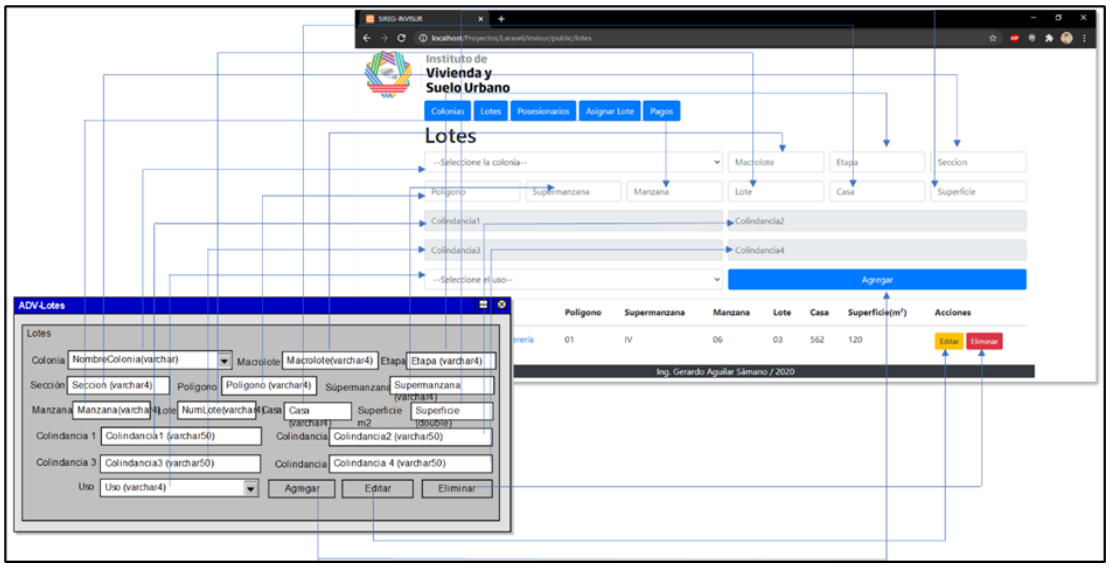


Figura 9. Implementación del módulo Lotes. Elaboración propia.

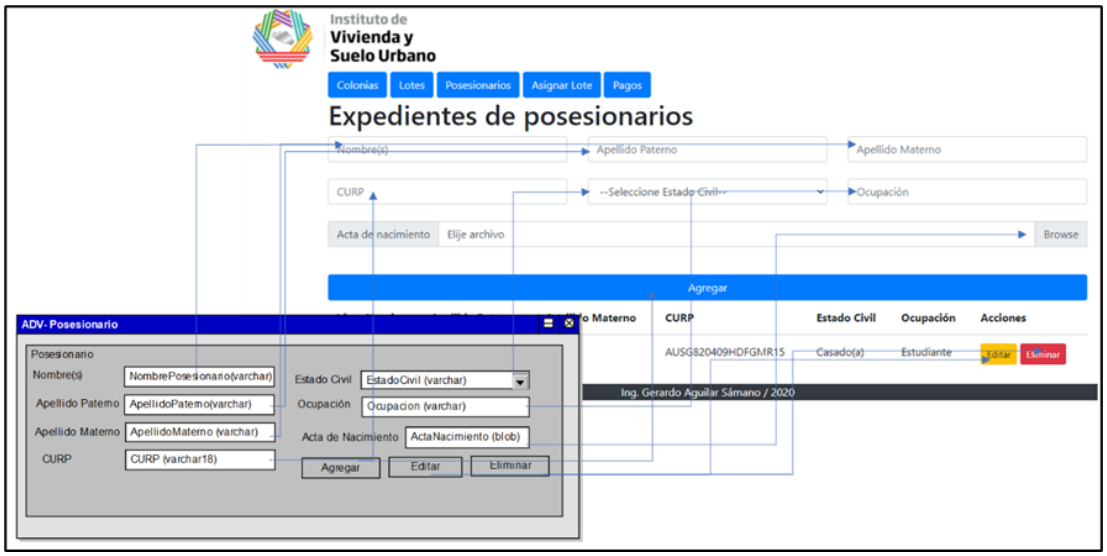


Figura 10. Implementación del módulo Posesionarios. Elaboración propia.

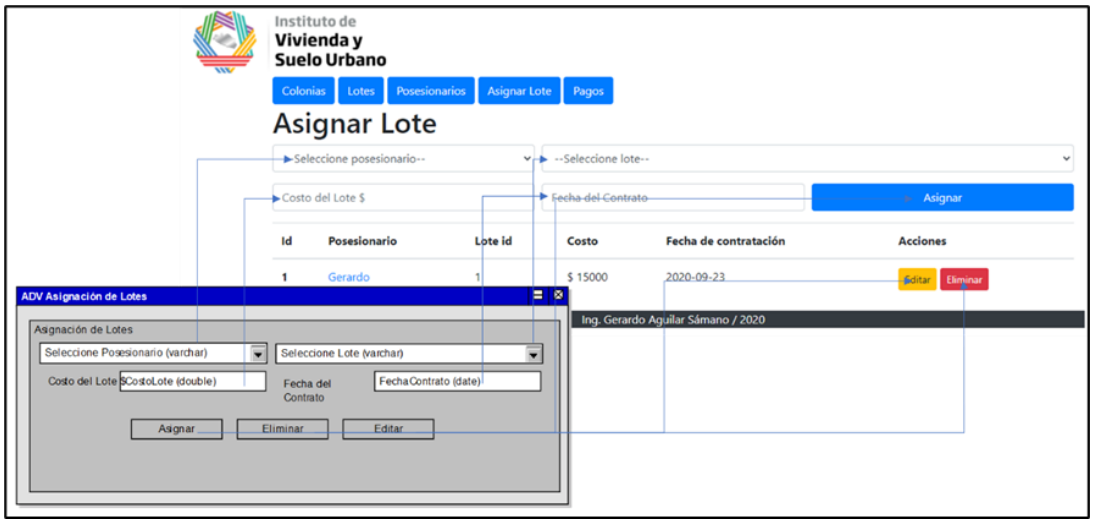


Figura 11. Implementación del módulo Asignar Lote. Elaboración propia.

Conclusiones.

Si bien OOHDM es una metodología que tiene más dos décadas de existencia, su implementación en el diseño de sistemas Web continúa siendo ampliamente requerido. El hecho de que haga uso de diagramas UML le proporciona una ventaja importante en cuanto a otras que no utilizan ningún estándar, ya que permiten un buen entendimiento y retroalimentación entre el cliente y el equipo de desarrollo. Como parte de las metodologías para desarrollar la aplicación Web SIREG, ha sido de gran ayuda ya que a la par de ser el punto de partida para el diseño de sus funcionalidades, los diagramas o artefactos resultantes, formarán parte de la documentación necesaria para futuras actualizaciones e incluso, que sean codificadas en otro lenguaje de programación orientado a objetos distinto a PHP. Cabe señalar que un área de oportunidad importante en la aplicación de la metodología OOHDM, es la falta de una herramienta CASE que contenga las plantillas necesarias para desarrollar cada una de sus fases, lo que permitiría acercar la metodología a quienes aún no implementan ninguna en el diseño de aplicaciones Web.

Referencias Bibliográficas.

Aguilar Sámano, G., Carranza Gómez, J., Hernández Bravo, J. M., & Montero Valverde, J. (2020). Propuesta de Desarrollo de un Sistema Web de Regularización y Escrituración de Predios en el Instituto de Vivienda y Suelo Urbano de Guerrero. *Programación Matemática y Software*, *13*(3). Obtenido de http://www.progmat.uaem.mx/

Conallen, J. (1999). Modeling Web Application Architectures with UML. *Communications of the ACM*, *42*(10).

Molina Ríos, J. R., Zea Ordoñez, M. P., Contento Segarra, M. J., & García Zerda, F. G. (2017). Comparación de Metodologías en Aplicaciones Web. *3C Tecnología: glosas de innovación aplicadas a la pyme*, 1-19.

Rumbaugh, J., Jacobson, I., & Booch, G. (2007). *El Lenguaje Unificado de Modelado, Manual de Referencia* (Segunda Edición ed.). Madrid, España: Pearson Educación, S.A.

Schwabe, D., & Rossi, G. (1998). An Object Oriented Approach to Web-Based Application Design. *Facultad de Ciencias Exactas. UNLP, Argentina*.

Schwabe, D., & Rossi, G. (1998). Developing Hypermedia Applications using OOHDM. *Facultad de Ciencias Exactas. UNLP, Argentina*.

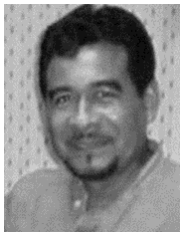
Silva, D. A., & Mercerat, B. (2001). Construyendo aplicaciones web con una metodología de diseño orientada a objetos. *Revista Colombiana de Computación, 2*(2), 1-21.

Información de los autores.

Gerardo Aguilar Sámano, es Ingeniero en Sistemas Computacionales por el Tecnológico Nacional de México / Instituto Tecnológico de Acapulco (2011). Se ha desempeñado profesionalmente como jefe del Departamento de Sistemas y Titular de la Unidad de Transparencia en el Instituto de Vivienda y Suelo Urbano de Guerrero, organismo público descentralizado del Gobierno del Estado de Guerrero. Actualmente cursa el cuarto semestre de la Maestría en Sistemas Computacionales con especialidad en Tecnologías Web en el Tecnológico Nacional de México / Instituto Tecnológico de Acapulco, acreditada en el Programa Nacional de Posgrado de Calidad del Conacyt.

Jorge Carranza Gómez, es Maestro en Tecnologías de la Información por el Tecnológico Nacional de México / Instituto Tecnológico de Zacatepec (2008) e Ingeniero en Sistemas Computacionales por el Tecnológico Nacional de México / Instituto Tecnológico de Acapulco (1994). Actualmente es Profesor Titular adscrito al Departamento de Estudios de Posgrado e Investigación del Tecnológico Nacional de México / Instituto Tecnológico de Acapulco. Ha participado como colaborador en diversos proyectos de investigación financiados. Pertenece al Reconocimiento a Profesores de Tiempo Completo (Perfil Deseable PRODEP). Forma parte como colaborador del Cuerpo Académico Sistemas Computacionales. Director de tesis y asesor de titulaciones a nivel licenciatura, así como director de 3 (tres) tesis de Maestría.

Juan Miguel Hernández Bravo, es Maestro en Tecnologías de la Información por el Tecnológico Nacional de México / IT de Zacatepec; Maestro en Ingeniería y Desarrollo de Software por el Colegio de Postgrado en Desarrollo de Software e Ingeniero en Sistemas Computacionales por el Tecnológico Nacional de México / IT de Acapulco. Actualmente es docente en el Tecnológico Nacional de México / IT de Acapulco de la Ingeniería en Sistemas Computacionales y de la Maestría en Sistemas Computacionales y cuenta con Reconocimiento al Perfil Deseable. Es miembro del Padrón Estatal de Investigadores y colaborador del Cuerpo Académico en Consolidación de Sistemas Computacionales.



José Antonio Montero Valverde, es Doctor en Ciencias Computacionales, por el Instituto Tecnológico de Estudios Superiores de Monterrey (2007); Maestro en Ciencias por el Instituto Politécnico Nacional (1987) e Ingeniero Electromecánico por el Instituto Tecnológico de Acapulco (1983). Realizó una Estancia Posdoctoral en el Instituto Nacional de Astrofísica Óptica y Electrónica, Puebla, (2010). Actualmente es Profesor Titular adscrito al Departamento de Posgrado e Investigación del Tecnológico Nacional de México / IT de Acapulco. Ha dirigido 7 (siete) proyectos financiados y participado en 8 (ocho). Autor de 15 (quince) publicaciones técnico-científicas. Ha dirigido y titulado más de 40 (cuarenta) tesis a nivel licenciatura, 1 (una) a nivel doctorado y 3 (tres) a nivel maestría.