

# Perfil UML para el desarrollo de aplicaciones WAP

Ricardo Soto D., Mauricio Camara J.

Escuela de Ingeniería Informática, Pontificia Universidad Católica de Valparaíso, Chile

E-mail: ricardo.soto@ucv.cl, mauricio.camara.j@mail.ucv.cl.

Av. Brasil 2241 Fax: 56-32-273859 Casilla 4059, Valparaíso, Chile.

## Abstract

UML (Unified Modeling Language) es el lenguaje de modelado más utilizado para especificar y documentar sistemas informáticos. Sin embargo, UML es un lenguaje de propósito general, por lo cual muchas veces prescinde de elementos para modelar y representar conceptos concretos de dominios más específicos. Como solución, OMG (Object Management Group) creó los perfiles, un mecanismo proporcionado para extender la sintaxis y semántica de UML para poder expresar conceptos más específicos de determinados dominios de aplicación.

En este trabajo se presenta un perfil UML para el desarrollo de aplicaciones WAP (Wireless Applications Protocol). El objetivo principal del perfil propuesto, es extender UML para entregar elementos específicos (clases, estereotipos, valores etiquetados y restricciones) que permitan a los desarrolladores modelar aplicaciones WAP. Si bien la expresividad de los diagramas tradicionales de UML permite modelar episodios importantes del proceso, todavía sigue siendo un dominio demasiado específico que difícilmente puede abordarse completamente sin extender el lenguaje. En el proceso existen aspectos navegacionales, de diseño y construcción que no se pueden desarrollar utilizando los elementos tradicionales de UML. No obstante utilizando los elementos específicos creados por el perfil propuesto, se pueden solucionar completamente e incluso conseguir una expresividad mucho mayor para los desarrolladores.

*Palabras Claves:* Ingeniería en Software, UML, perfiles UML, Metamodelado, WAP.

## 1. Introducción

Los sistemas informáticos son cada vez más complejos y sofisticados, lo que representa un importante reto para los ingenieros y desarrolladores de software. En un principio la preocupación más importante era poner énfasis en la definición de la estructura y calidad del código final, ahora esta preocupación se ha trasladado en dedicar más tiempo y esfuerzo al modelado del sistema.

UML, el lenguaje de modelado más utilizado para especificar y documentar sistemas informáticos, es un lenguaje de modelado de propósito general, por lo que se puede utilizar para especificar diferentes sistemas y diversos dominios de aplicación (empresarial, aeronáutica, universitaria, etc.)

Esta característica de generalidad que entrega UML proporciona una gran flexibilidad al momento de modelar sistemas. Sin embargo, existen ocasiones en las que es mejor contar con lenguajes de modelado más específicos, para así poder representar características más concretas de ciertos dominios particulares. Esto sucede, por ejemplo, cuando la sintaxis de UML no permite expresar los conceptos específicos del dominio, o cuando se desea restringir y especializar los constructores propios de UML, que suelen ser demasiado genéricos y numerosos.

Esta falta de expresividad también emerge al modelar aplicaciones WAP, donde los elementos tradicionales de UML no permiten abordar los aspectos claves de navegación, diseño y construcción de la aplicación. Como solución a esta problemática se presenta un nuevo perfil UML, el cual pretende entregar elementos específicos para el dominio en cuestión, basándose en forma completa en los mecanismos de extensión propuestos por OMG.

El presente trabajo está organizado de la siguiente manera: La sección 2 entrega los conceptos básicos del metamodelado y el MOF (Meta-Object Facility)[1]. La sección 3 presenta el desarrollo del perfil para aplicaciones WAP propuesto. La sección 4 la aplicación del perfil a un caso de estudio, para finalizar en la sección 5 con las conclusiones y trabajos futuros.

## 2. Metamodelado y MOF

El metamodelado es un mecanismo que permite construir formalmente lenguajes de modelado, como son UML o CWM (Common Warehouse Metamodel)[2]. El metamodelo de un lenguaje de modelado es una definición exacta de todos sus elementos (clases, asociaciones, estereotipos, etc.) mediante conceptos y reglas de cierto metalenguaje.

Para entender mejor los conceptos relacionados a metalenguajes y metamodelos. OMG propone una arquitectura de 4 capas [3][4], orientada a estandarizar todos los conceptos de modelación, desde los modelos más abstractos a los metamodelos.

### 2.1 Arquitectura de 4 Capas de Modelado definida por OMG

Los niveles definidos por OMG son denominados comúnmente con las iniciales M0, M1, M2 y M3. A continuación se describen estos niveles.

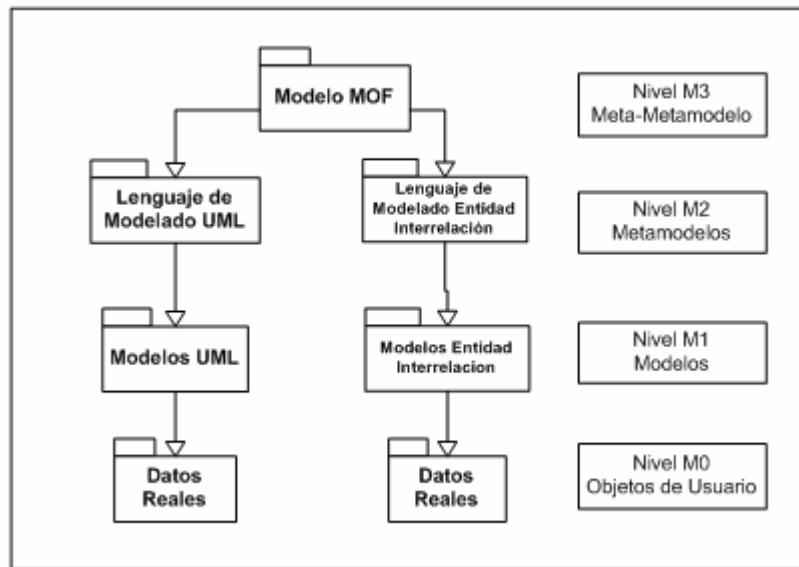
**El nivel M0 (instancias):** Modela el sistema real, y sus elementos son los datos de un sistema de información. Por ejemplo, “Juan Pérez”, que vive en “Av. Libertad 123”.

**El nivel M1 (modelo del sistema):** Los elementos del nivel M1 son los modelos de los datos. Por ejemplo, entidades como “Persona”, “Auto”, atributos como “dirección”, “patente”, y relaciones como “vender”, “comprar”, etc.

**El nivel M2 (el metamodelo):** Los elementos del nivel M2 son los lenguajes de modelado, por ejemplo UML. En este caso los conceptos del nivel M2 podrían ser, “clase”, “atributo” o “asociación”.

**El nivel M3 (el meta-metamodelo):** En el nivel superior de la arquitectura conceptual definida por OMG se encuentra el MOF. Este es un lenguaje abstracto creado para definir lenguajes de modelado como son UML o CWM.

En la figura nº1 se visualizan los 4 niveles de la arquitectura conceptual definida por OMG.



*Figura 1 Esquema General de Metamodelos*

### 3. Construcción del perfil UML para el desarrollo de aplicaciones WAP

Para crear una aplicación WAP, es importante contar con un lenguaje de modelado que especifique y documente un sistema de forma precisa. Los diagramas tradicionales de UML permiten modelar etapas importantes del proceso, sin embargo existen aspectos de diseño, navegación y construcción que difícilmente pueden abordarse completamente sin extender UML [5].

Para solucionar esto se propone extender UML creando un perfil específico para el desarrollo de aplicaciones WAP. Para la creación de este perfil se utilizará un procedimiento que consta de 5 etapas [6][7][8][9], para ello hay que tener especial consideración en los elementos que lo componen, como son los estereotipos, valor etiquetados y restricciones a utilizar.

**Estereotipos:** Están definidos por un nombre y por una serie de elementos del metamodelo sobre los que puede asociarse. Los estereotipos representan las nuevas características que son agregadas al metamodelo UML para extender este lenguaje.

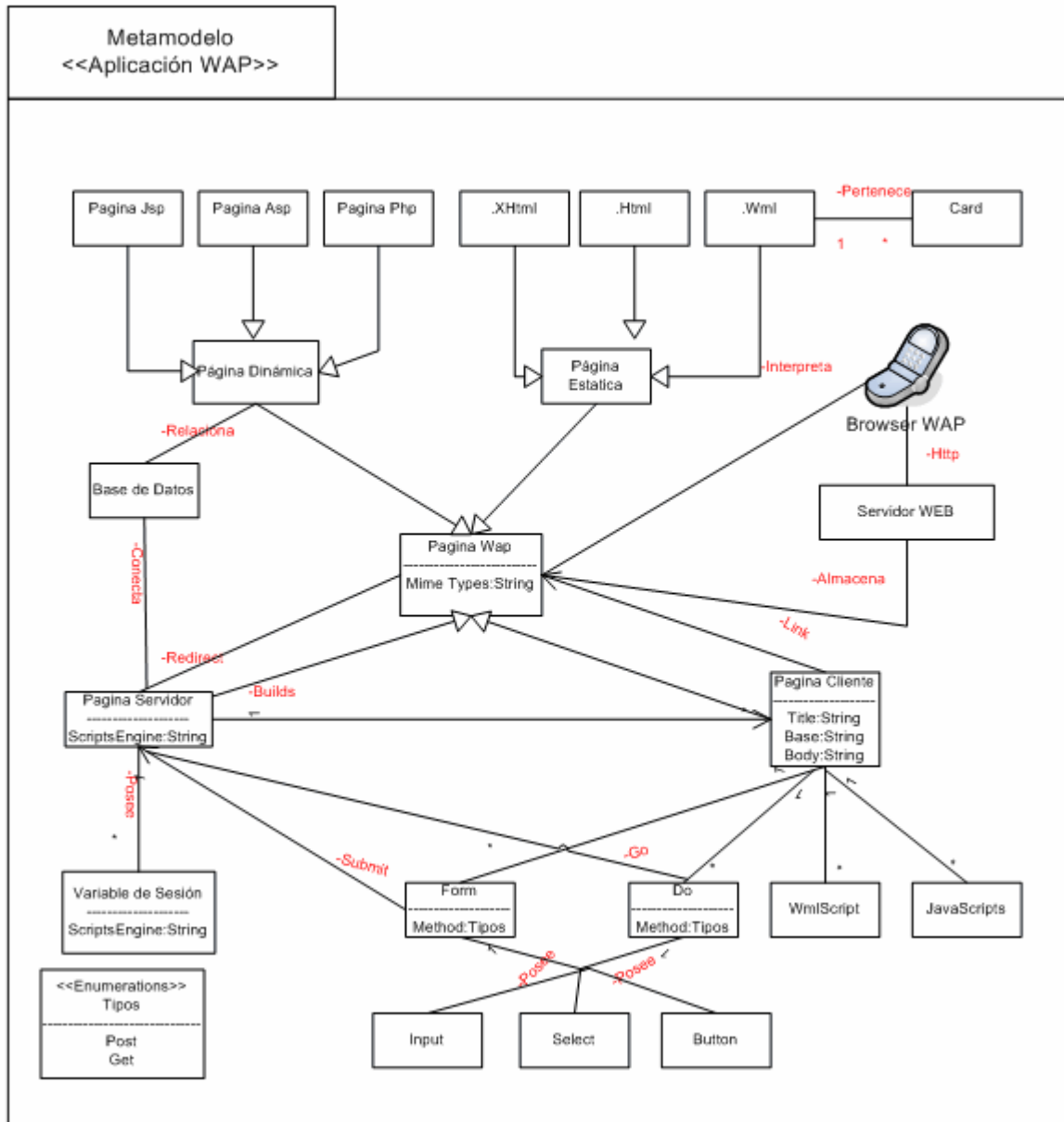
**Valores Etiquetados:** Son meta-atributos que se asocian a una metaclass del metamodelo extendido por un perfil. Todo valor etiquetado ha de contar con un nombre y un tipo, y se asocia a un determinado estereotipo.

**Restricciones:** Las restricciones se asocian a los estereotipos e imponen condiciones sobre los elementos del metamodelo que han sido estereotipados. Las restricciones se transcriben en lenguaje natural o en OCL (Object Constraint Language)[10], lenguaje definido por OMG como el estándar para transcribir restricciones y consultas.

A continuación se explica en detalle el desarrollo de cada una de las 5 etapas para obtener el perfil el UML:

## Paso 1: Creación del Metamodelo

El primer paso que se debe realizar para crear un perfil UML, es definir el metamodelo de la plataforma o dominio de aplicación que se pretenda estudiar. En este caso el dominio de aplicación a representar es el modelado de aplicaciones WAP y los elementos que formarán el metamodelo, aquellos que se utilizan para programar e implementar aplicaciones WAP (página WAP, form, link, do, card, go, etc.)[11][12]. El metamodelo del modelado WAP se visualiza en la figura n°2.



**Figura n°2 Metamodelo de WAP**

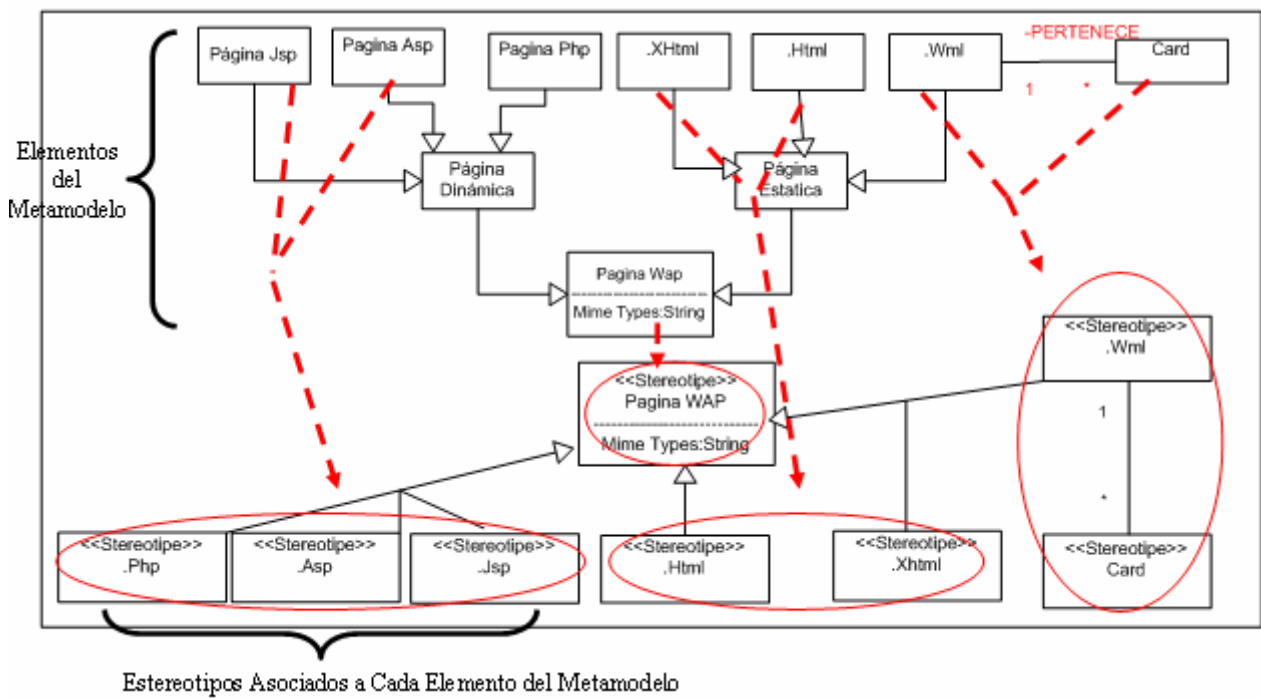
El elemento principal de una aplicación WAP, es la página WAP (entidad central del metamodelo). Las páginas WAP pueden ser estáticas si sólo muestran información o dinámicas si existe interacción con el usuario (ingreso de datos, lectura de datos personalizados, etc.). Centrándose en la programación de un sitio WAP, las páginas dinámicas pueden estar escritas, por ejemplo, en JSP, ASP o PHP y las páginas estáticas podrían ser del tipo XHTML, HTML, WML, JSP, ASP o PHP.

Una página WAP puede poseer scripts que son ejecutados por el cliente (página cliente) o scripts que son ejecutados por el lado del servidor (página servidor)[13].

En una página cliente aparecen entidades como form, do, los cuales tiene como función capturar información dada por el usuario. Elementos javascripts o wmlscripts, son scripts que se ejecutan en el browser WAP. Finalmente por el lado del servidor existen las variables de sesión, las cuales, se almacenan en el servidor.

### Paso 2: Definición del perfil UML

El segundo paso es la definición del perfil. Para esto se debe crear un estereotipo por cada elemento del metamodelo (página WAP, form, do, página JSP, página HTML, submit, etc.) que deseamos incluir en el perfil. Los estereotipos deben tener el mismo nombre que los elementos del metamodelo, para así mantener una unión entre ambas etapas. A modo de ejemplo, en la figura n°3, se seleccionaron algunas entidades del metamodelo, creando para cada una el estereotipo correspondiente.



**Figura n°3** Relación entre Metamodelo y Estereotipos

### Paso 3: Identificar los elementos a extender

El paso 3 consiste en identificar las elementos de UML que se van a extender con cada estereotipo. Ejemplo de tales elementos son las clases, asociaciones, atributos, paquetes, etc. En la figura n°4 se observan los estereotipos asociados a la estructura UML “Clase”.

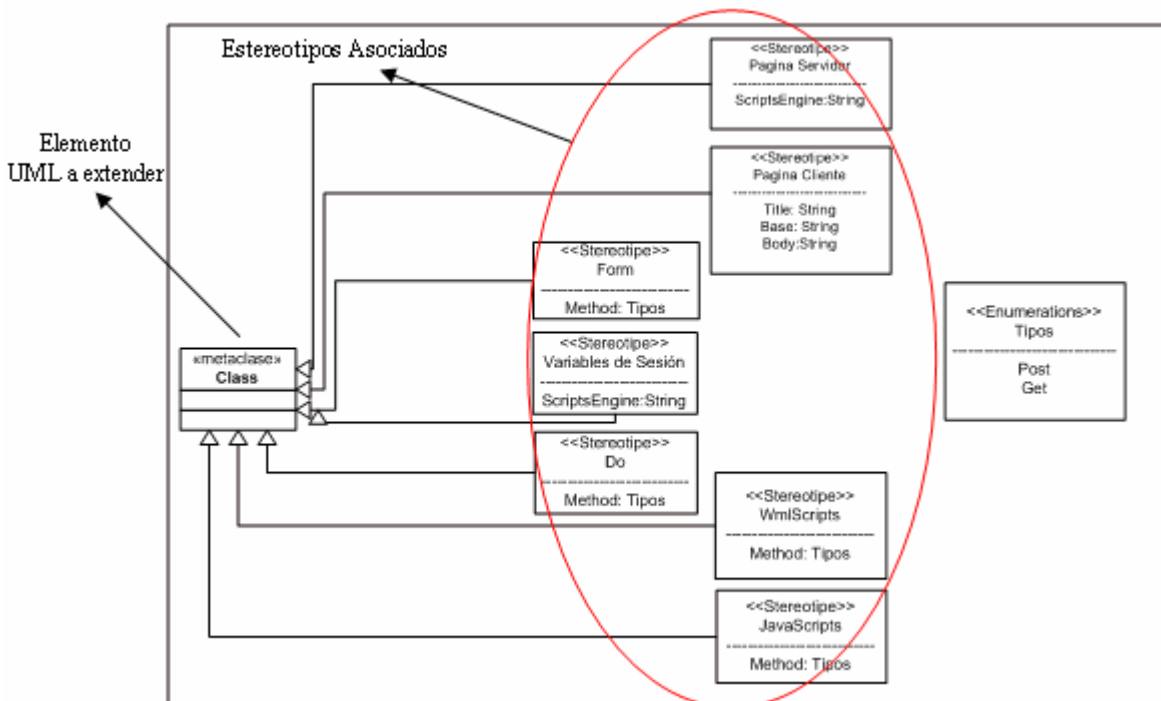


Figura n°4 Elemento UML a extender

#### Paso 4: Definición de Valores Etiquetados

Otro punto importante a considerar, es que por cada atributo presente en el metamodelo se debe agregar el valor etiquetado correspondiente en el perfil UML. Los valores etiquetados son atributos adicionales que se asocian al perfil. En la figura n°5 se visualiza una entidad del metamodelo que posee un atributo, el cual, será agregado al perfil UML como valor etiquetado.

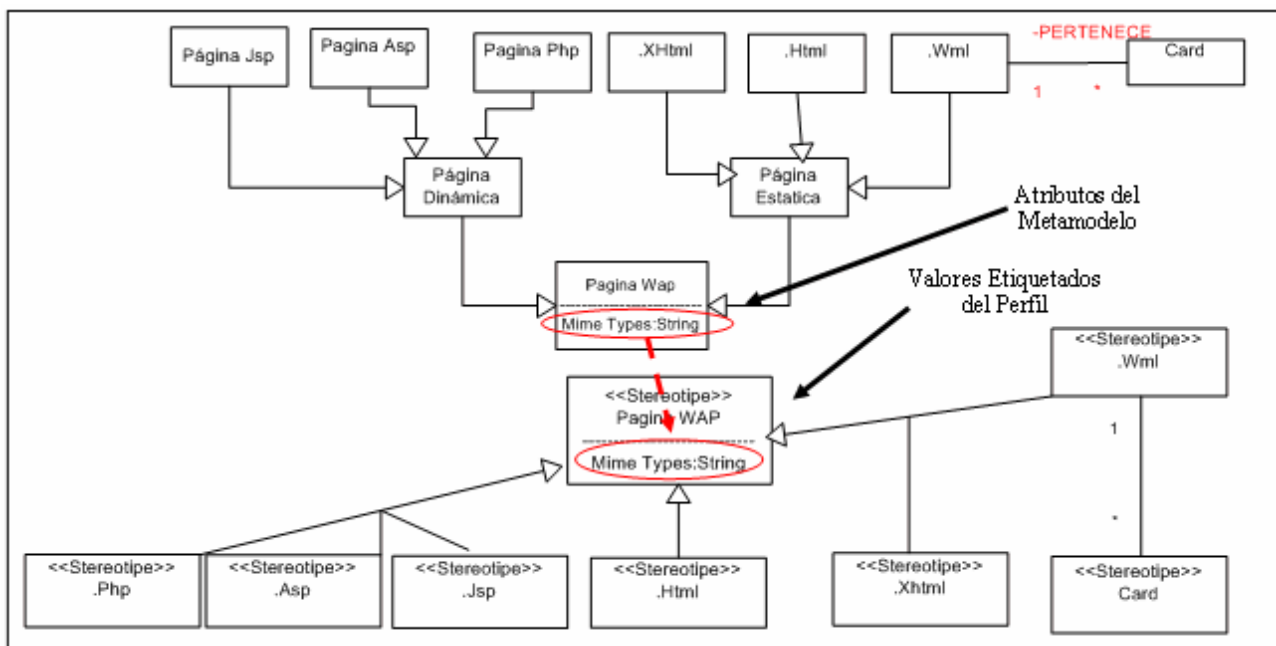


Figura n°5 Valores Etiquetados del perfil

## Paso 5: Definición de Restricciones

El último paso es definir las restricciones que forman parte del perfil. Las restricciones se definirán en lenguaje OCL, lenguaje definido por OMG como el estándar para transcribir restricciones y consultas. A continuación se visualizan tres restricciones del perfil.

**Restricción 1:** La clase “Form” no puede tener funciones asociadas a ella. Cualquier operación que interactué con este elemento debe estar definida en la clase “Página WAP” que contiene a la clase “Form”.

```
Context UML::InfrastructureLibrary::Core::Constructs::Class Inv:  
  self.isStereotyped("Form") implies  
  self.Operations -> isEmpty()
```

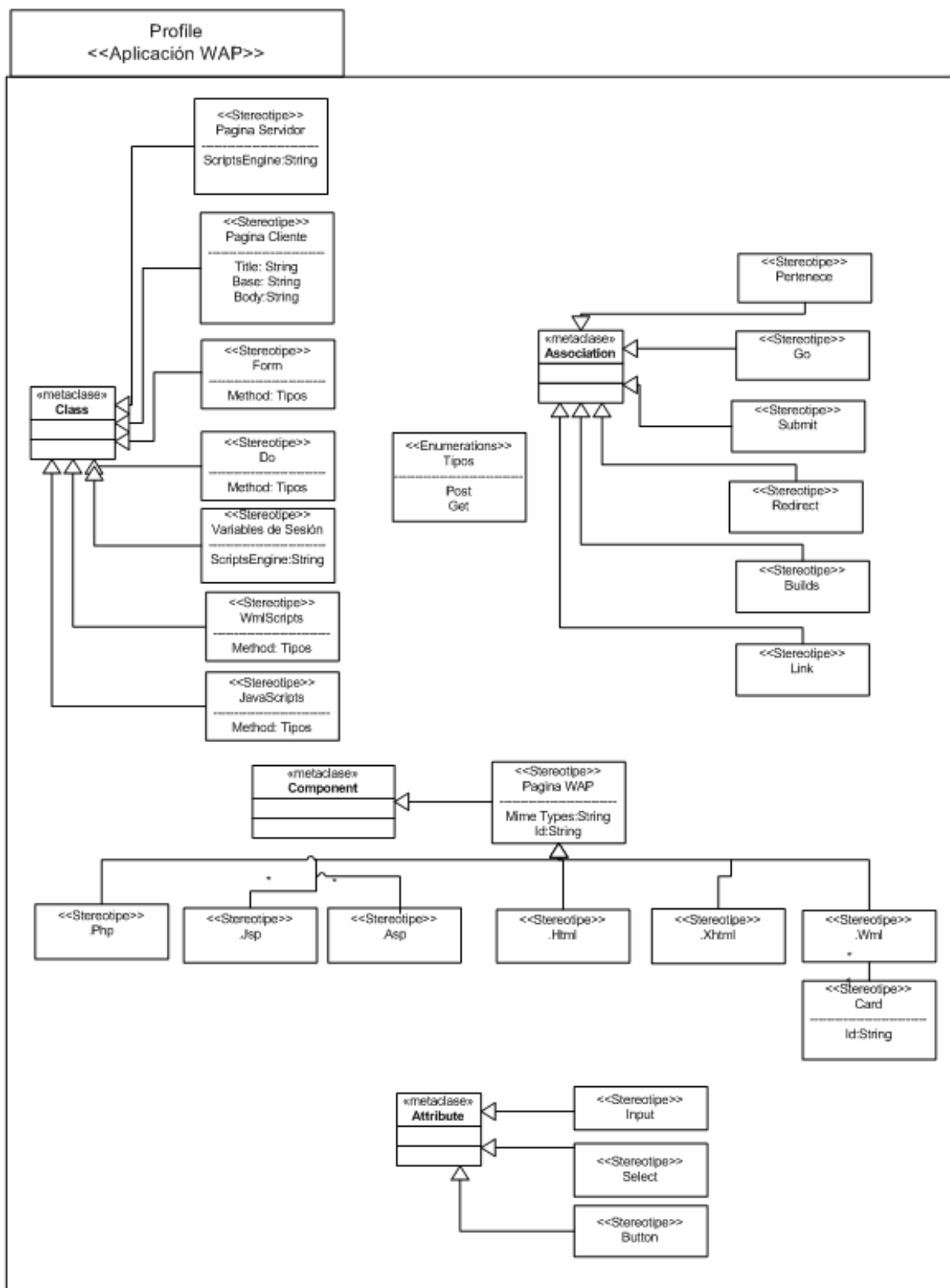
**Restricción 2:** La clase “Do” no puede tener funciones asociadas a ella. Cualquier operación que interactué con este elemento debe estar definida en la clase “Página WAP” que contiene a la clase “Do”.

```
Context UML::InfrastructureLibrary::Core::Constructs::Class Inv:  
  self.isStereotyped("Do") implies  
  self.Operations -> isEmpty()
```

**Restricción 3:** El Atributo “ScriptsEngine” de la clase “variables de sesión” debe ser igual al “ScriptsEngine” de la clase “página servidor” relacionada.

```
Context UML::InfrastructureLibrary::Core::Constructs::Class Inv:  
  self.isStereotyped("Variables de Sesión") implies  
  (Class.isStereotyped("Pagina Servidor").ScriptsEngine =  
  Class.isStereotyped("variables de Sesión").ScriptsEngine)
```

Finalmente el perfil completo se aprecia en la figura nº6, en él aparecen todos los elementos del metamodelo que se agregan a UML, con sus respectivos valores etiquetados, además se especifican las estructuras de UML que serán extendidas.



*Figura n°6 perfil Modelado WAP*



#### 4. Utilización del perfil en un caso de estudio

Es importante recordar que uno de los principales objetivos que los perfiles deben cumplir, es la inclusión de nuevas características a los modelos de UML. Para entender de forma concreta como se aplica un perfil, es necesario mostrar las características del perfil aplicándolo a un sistema real.

Para ello se utilizará un caso de estudio, en el cual se presentarán los modelos desarrollados para una librería virtual. Esta aplicación posee dos grandes funcionalidades. La primera es realizar la venta de libros a través de un dispositivo celular y la segunda es realizar el mantenimiento del sitio, es decir, la aplicación debe permitir agregar, eliminar y modificar libros. Los siguientes diagramas permiten visualizar la extensión realizada por el perfil propuesto y su aplicación al caso de estudio.

**Diagrama de Componentes:** El diagrama de componentes se utiliza para modelar la vista estática de un sistema. Muestra la organización y las dependencias entre un conjunto de componentes. El diagrama de componentes de la figura n°7, muestra las páginas WAP y las asociaciones entre las distintas páginas pertenecientes al módulo de administración.

En el sistema cada página es tratada como un componente ya que dentro posee las clases de diseño necesarias que definen su comportamiento. Cada una de las clases de diseño han sido obtenidas del perfil. En el diagrama se aprecian los estereotipos definidos anteriormente, como <<card >>, <<Pagina Php>>, <<Pagina Wml>>.

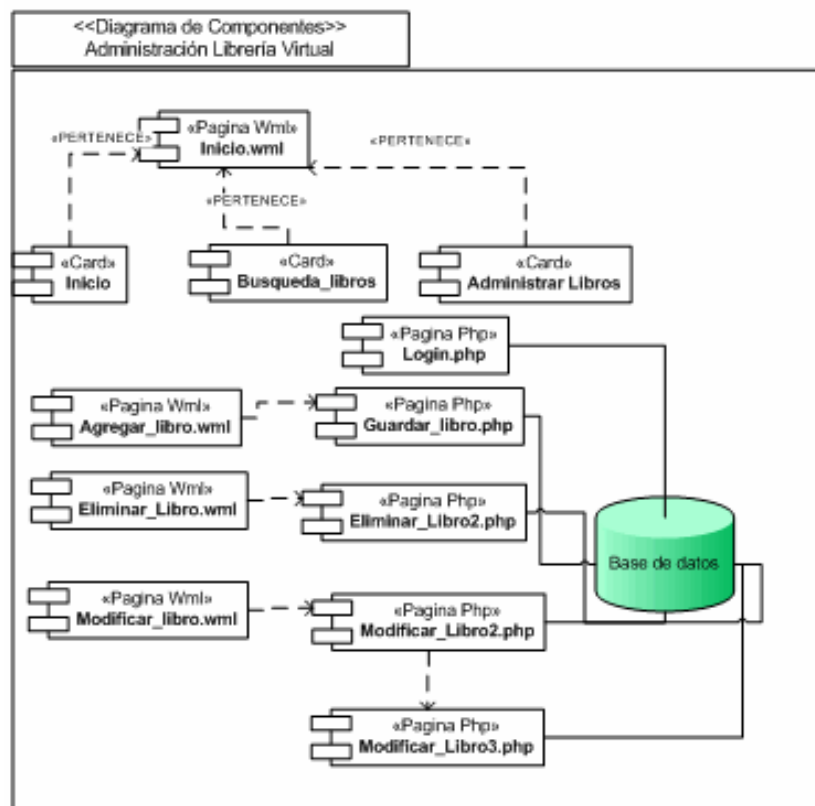
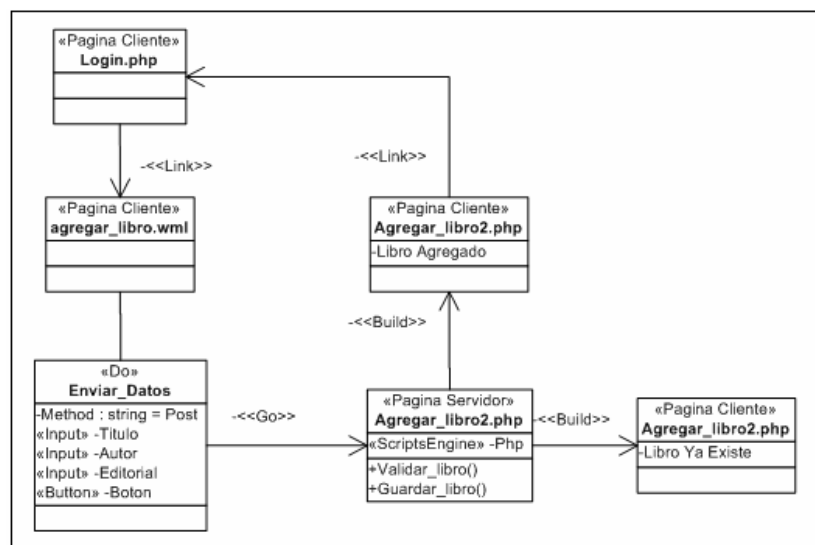


Figura n°7 Diagrama de Componentes Librería Virtual

**Diagrama de Clases de Diseño:** El siguiente modelo muestra algunas de las clases, relaciones y atributos, definidos en el perfil y que son aplicados a un sistema WAP. En el diagrama de clases de diseño las páginas WAP pueden ser páginas cliente o servidor, dependiendo si las operaciones que realiza son locales (clientes) o necesitan acceso al servidor (página servidor). Otro elemento que aparece en el modelo es el estereotipo <<Do>>, el cual indica, la existencia de un formulario dentro una página cliente. Para la relación existente entre cada página WAP, se pueden utilizar los estereotipos <<Go>>, <<Link>>, <<Submit>>, <<Redirect>>, por su nombre se subentiende lo que cada uno representa.

**Diagrama 1: Agregar Libro:** El diagrama de la figura n°8 representa la interacción entre las distintas clases del perfil, necesarias para el proceso de agregar un nuevo libro a una Base de Datos.



*Figura n°8 Diagrama Clases de Diseño: Agregar Libro*

El modelado completo del sitio implica tener un diagrama de clases de diseño por cada una de las páginas del sistema, cada una de la cuales se incorpora en el diagrama de componentes. El perfil propuesto permite visualizar claramente los elementos que cada página posee y asimismo entender la interacción y comportamiento tanto a nivel de diseño como de implementación de cada uno de los elementos que participan en el sistema.

## 5. Conclusión

La creación de perfiles y extensiones UML, se debe realizar cuando los modelos tradicionales no entregan la expresividad necesaria para representar las características específicas de dominios particulares, como ocurre con el proceso de modelado de una aplicación WAP.

En este trabajo se presentó un perfil UML para el desarrollo de aplicaciones WAP. El cual se obtuvo a partir de un proceso compuesto por 5 etapas, el que entregó como resultado un perfil UML constituido esencialmente por los elementos utilizados para diseñar e implementar sitios WAP.

Luego el perfil fue aplicado en el desarrollo de un caso de estudio permitiendo modelar en forma clara, simple y expresiva los aspectos de navegación, diseño y construcción que difícilmente pueden ser modelados con los elementos tradicionales de UML. Además se pone a disposición de

los desarrolladores una herramienta nueva de modelado que permite, por su expresividad diseñar aplicaciones WAP sin omitir aspectos fundamentales de las etapas de diseño e implementación de este tipo de aplicaciones.

Es tremendamente útil para el desarrollador contar con procedimientos, restricciones, elementos y notaciones específicas al dominio que se pretende modelar. De esta forma se crean diagramas de mayor definición y expresividad al dominio en cuestión, obteniendo como resultado una disminución en el tiempo utilizado en avanzar desde las etapas de modelado a las etapas de implementación.

El perfil UML propuesto en este documento es el inicio de un completo estudio del proceso de desarrollo de sistemas WAP, el cual será aplicado en el desarrollo de distintos tipos de sistemas para equipos móviles, con el objetivo de obtener experiencia para desarrollar patrones y una herramienta CASE que permita simplificar y clarificar aún más, todo el proceso de desarrollo de este tipo de aplicaciones.

## 6. Referencias

- [1] Object Management Group: MOF 2.0 Core Specification (2004), <http://www.omg.org/cgi-bin/doc?ptc/2004-10-15>  
<http://www.omg.org/cgi-bin/doc?ptc/2004-10-15>
- [2] Object Management Group: CWM 1.1 (2002)  
<http://www.omg.org/technology/documents/formal/cwm.htm>
- [3] Object Management Group: UML 2.0 Infrastructure Specification (2003),  
<http://www.omg.org/cgi-bin/doc?ptc/2003-09-15>.
- [4] Object Management Group: UML 2.0 Superstructure Specification, (2004)  
<http://www.omg.org/cgi-bin/doc?ptc/2004-10-02>.
- [5] Soto Ricardo & Rodríguez Nibaldo, “New UML 2.0 based models to design WAP Applications”. Seventh International Conference on UML Modeling Languages and Applications, UML 2004. Lisboa, Portugal.
- [6] Blankenhorn Kai & Jeckle Mario, “A UML Profile for GUI Layout”, NODe 2004, LNCS 3263, pp.110-121, 2004 Springer-Verlag Berlin Heidelberg 2004.
- [7] L. Fuentes, J. M. Troya, A. Vallecillo. “Using UML Profiles for Documenting Web-based Application Frameworks”. Annals of Software Engineering, Vol. 13, pp. 249-264, June 2002.
- [8] Vincenzo Grassi, Raffaella Mirandola and Antonino Sabetta, “A UML Profile to Model Mobile Systems”. Seventh International Conference on UML Modeling Languages and Applications, UML 2004. Lisboa, Portugal.
- [9] Tewfik Ziadi, Loïc Héloüet, and Jean-Marc Jézéquel, “Towards a UML Profile for Software Product Lines”, PFE 2003, LNCS 3014, pp. 129–139, 2004 Springer-Verlag Berlin Heidelberg 2004.
- [10] Object Management Group: UML 2.0 OCL (Object Constraint Language Specification) (2003), <http://www.omg.org/cgi-bin/doc?ptc/2003-10-14>.
- [11] Open Mobile Alliance: Wireless Markup Language v2.0  
<http://www.openmobilealliance.org/tech/affiliates/wap/wap-238-wml-20010911-a.pdf>
- [12] Open Mobile Alliance: Technical Section  
<http://www.openmobilealliance.org/tech/affiliates/index.html>
- [13] J. Conallen. “*Building Web Applications with UML*”, Addison Wesley, 1999