

# Un marco de referencia para la caracterización de desarrollos de aplicaciones Web

A. Bianchini, A. Suárez, C. A. Pérez Díaz

Depto. de Computación y Tecnología de la Información, Universidad Simón Bolívar. Valle de Sartenejas, Baruta, Edo. Miranda, Venezuela.

## RESUMEN

*Durante la última década se han propuesto varias metodologías y enfoques con la finalidad de mejorar la calidad del desarrollo de aplicaciones Web. Existen propuestas que proveen técnicas y mecanismos para especificar modelos de producto y modelos de proceso, pero dejan de lado aspectos relacionados con desarrollos basados en distintas situaciones. Además algunas propuestas, tanto en el área empresarial como en la académica, no son lo suficientemente flexibles para configurarse según las situaciones y condiciones del proyecto que se desea llevar a cabo. Esas condiciones incluyen tipo y complejidad de la aplicación Web, tipos de modelos a crear, características del equipo de desarrollo, recursos tecnológicos, entre otros. Este artículo presenta un marco de referencia para caracterizar el desarrollo de aplicaciones Web. En base a las características del proyecto, es posible adecuar una enfoque metodológico basado en la situación de desarrollo, y de esta manera el enfoque puede ser más flexible dejando de lado algunas consideraciones. El marco de referencia se ajusta a los fundamentos de la Ingeniería Web y a los principios de la Ingeniería Situacional de Métodos.*

## ABSTRACT

*Over the past decade several methodologies have been proposed to improve the quality of Web application development. There are proposals that provide techniques and mechanisms to specify the product model, but leave apart some aspects about the process development model for shaping the development and the generation of products based on different situations. Besides, some industry and academic methods are not flexible enough to react according to the different situations and conditions of the projects to be. These conditions may include application type and complexity, models to be created, development team characteristics, technological resources, among others. This paper presents KANON, a framework to characterize web applications development. With the characteristics of the project to carry out, it is possible to adapt a methodological approach based on development situation, and in some way the approach can be more flexible leaving apart the prescriptive guideline. The framework uses the foundations of Web Engineering and the principles of method creation from Situational Method Engineering.*

**Keywords:** Ingeniería Web, Ingeniería Situacional de Métodos, desarrollo de aplicaciones Web.

## 1. Introducción

La complejidad de las aplicaciones Web se ha incrementado significativamente: desde sistemas estáticos orientados a la diseminación de contenidos de información, hasta sistemas dinámicos como transacciones en línea del tipo *e-banking*, *e-commerce*, entre otras.

Durante la última década se han propuesto varias metodologías para mejorar la calidad del desarrollo de aplicaciones Web. Entre ellas destacan OOHDM [R95], OOWS [PFP03], WebML [CFB00] y UWE [KK02]. En efecto, se ha dado una atención considerable a la Ingeniería Web, una disciplina que provee un enfoque sistemático y disciplinado para desarrollar, documentar y mantener aplicaciones Web.

La Ingeniería Web es más compleja que la tradicional Ingeniería de Software, puesto que la primera conlleva nuevos asuntos a resolver como el diseño de interfaces muy

ricas (RIA, para la presentación e interacción), el soporte al modelado de la navegación (enfoque hipertexto) y el modelado de los contenidos Web.

Los métodos y modelos de la Ingeniería de Software no son lo suficientemente expresivos si se consideran las características específicas de las aplicaciones Web [SK06]. Además el desarrollo de estas aplicaciones es caracterizada por [Zie09]: i) equipo de desarrollo de pocas personas y multidisciplinario; ii) pequeñas entregas de funcionalidades (procesos iterativos e incrementales); iii) número importante de involucrados provenientes de campos y áreas distintos; iv) en muchos casos las especificaciones de requisitos es informal; y v) tiempos de entrega del producto final relativamente cortos.

En el desarrollo de aplicaciones, existen métodos que pueden ser considerados genéricos, por lo tanto se utilizan en cualquier tipo de aplicación Web a desarrollar, diferentes dominios y diversas situaciones. En la práctica, un método rígido o genérico no siempre encaja bien para

cada dominio de desarrollo [VVP08]. Generalmente hay acuerdos sobre los pasos y fases que siguen los métodos en la Ingeniería Web, ya que ellos establecen procesos típicos: desde la captura de requisitos hasta la implementación, y para cada fase proponen el uso de distintas técnicas y procedimientos. Existe también un acuerdo acerca de los productos que se deben generar y los procesos a seguir, sin embargo proveen tareas complejas en lugar de brindar guías a los desarrolladores [SKB08]. Por ejemplo, una fase exhaustiva de captura de requisitos puede ser innecesaria en una aplicación Web de pequeño tamaño, o de muy baja complejidad.

Algunas propuestas metodológicas dejan de lado aspectos relacionados con el modelo de desarrollo a seguir y la generación de productos en base a diferentes situaciones. Una situación es la combinación de circunstancias en un momento dado, y posiblemente en una organización dada. Una situación afecta la forma de trabajar y el tipo de producto a generar, y puede ser la combinación de diferentes factores como el dominio de la aplicación, la categoría específica de la aplicación Web, nivel de complejidad, capacidades del equipo de desarrollo, tiempos y costos relacionados con el desarrollo, entre otros. En cada nuevo desarrollo, los diseñadores deben adaptar o extender, de alguna manera, el método para así llevar a cabo el trabajo [VVP08].

La disciplina para construir un método para un proyecto específico basado en una situación es la Ingeniería Situacional de Métodos [Bri96].

El objetivo general de esta propuesta reside en que el desarrollo de aplicaciones Web requiere de enfoques flexibles que permitan a los desarrolladores configurar el proceso de desarrollo y la generación de modelos de productos, de acuerdo a un conjunto de atributos sobre el proyecto a llevar a cabo. Además el enfoque debe guiar los desarrolladores en la creación de modelos y artefactos, necesarios para especificar la aplicación Web a desarrollar.

Como producto de un proyecto de investigación llevado a cabo durante los últimos cinco años, en el Laboratorio de Lenguajes del Departamento de Computación y Tecnología de la Información de la Universidad Simón Bolívar (Caracas), se está definiendo una metodología de desarrollo de aplicaciones Web bajo el patrón MVC [BOS05]. La metodología en lo sucesivo se llamará WEBFDM – *Web application Flexible Development Method* [Bia10].

Con WEBFDM, en proceso de nuevas definiciones, se pretende flexibilizar el diseño de aplicaciones Web que siguen la arquitectura del patrón MVC, a partir de la descripción del problema o sistema deseado, con el propósito de conducir de una manera expedita a una propuesta de diseño en términos de los componentes que deben conformar una aplicación Web.

WebFDM apoya la separación de tareas en el proceso de desarrollo de una aplicación Web, ya que se complementa y puede ser combinada con cualquier otra herramienta para la implementación de la capa del servidor de la aplicación; mientras que lo concerniente al modelo de datos puede ser diseñado e implementado utilizando los métodos tradicionales preferidos por el desarrollador.

Para obtener un enfoque flexible se está rediseñando la metodología WEBFDM y su herramienta CASE, llamada COHESIÓN [DSB10] la cual brinda apoyo y soporte total en el uso de la metodología.

La flexibilidad esperada se alinea a la creación de una nueva instancia de la metodología según una situación de desarrollo. Por lo tanto WEBFDM sigue los principios de la Ingeniería Situacional de Métodos y los fundamentos de la Ingeniería Web. Entre otros elementos, para la nueva versión de la metodología se requiere de estrategias y el estudio de indicadores y atributos que puedan caracterizar un proyecto, su ambiente y sus circunstancias del desarrollo.

La principal contribución de este artículo es KANON, un marco de referencia que define la colección de atributos para caracterizar aplicaciones Web y sus circunstancias de desarrollo. Con este marco de referencia, la herramienta COHESIÓN determina y crea un nuevo método basado en la situación de desarrollo. De esta manera la metodología WEBFDM se configura en base a los principios de la Ingeniería Situacional de Métodos.

Este artículo está organizado de la siguiente forma: en la sección 2 se exponen algunas definiciones para introducir la propuesta; en la sección 3 se presentan los trabajos relacionados. La propuesta se describe en la sección 4. Finalmente, las conclusiones y trabajos a futuro se presentan en la sección 5.

## 2. Definiciones preliminares

En esta sección se presentan algunas definiciones necesarias para la descripción del marco de referencia, y se presentan términos y conceptos del área de la Ingeniería Situacional de Métodos.

### 2.1 Ingeniería Situacional de Métodos

Desde la aparición de la Ingeniería de Software, se han desarrollado diversos métodos para encarar los proyectos de desarrollo. Algunos métodos se orientan a la especificación y desarrollo del producto final, y otros quizás se enfocan más al proceso de desarrollo. Sin embargo, es sabido que no todo método sirve idealmente para cualquier desarrollo, por lo que generalmente los desarrolladores llevan a cabo adaptaciones, en algunos casos no formales de tal o cual método. Esto genera un problema de reutilización del método, cuando tales adaptaciones no están ajustadas a criterios formales de utilización.

La Ingeniería de Métodos es la disciplina para construir nuevos métodos a partir de métodos ya existentes. Está enfocado al diseño, construcción y evaluación de métodos, técnicas y herramientas de soporte para el desarrollo de sistemas de software [Bri96].

Un tipo de Ingeniería de Métodos es la Ingeniería Situacional de Métodos (*Situational Method Engineering*) cuya finalidad es la construcción de métodos específicos para un proyecto dado, y basado en situaciones de desarrollo. La Ingeniería Situacional de Métodos (ISM) propone estrategias para la selección de métodos, llamados



bloques de métodos (tomados de un repositorio), para crear uno nuevo basado en las características y las circunstancias de desarrollo.

La ventaja principal de aplicar ISM son las siguientes: i) construir o crear un método base que permita la definición de métodos para un dominio específico, reutilizando bloques validados; ii) adaptar, de forma sencilla, métodos basados en las necesidades de un proyecto específico, con sus restricciones y características [VVP08]; iii) proveer de técnicas basadas en la reutilización de bloques ya existentes, en la construcción de nuevos métodos, más flexibles y mejor adaptados a la situación de desarrollo.

La ISM no está orientada a la adquisición de métodos listos para usar y adquiridos de un proveedor, sino que su interés es la construcción local para una organización específica o para un enfoque metodológico específico para llevar a cabo el proyecto de desarrollo.

## 2.2 Bloques de método y enfoques para la construcción de nuevos métodos

En la literatura se utilizan distintos términos para hacer referencia a los bloques de métodos (*building-blocks*) que son el núcleo de los enfoques para la creación de métodos en la ISM. Se utiliza el término Fragmento de Método (*method fragment*) el cual incluye la perspectiva producto (modelo, artefacto) y la perspectiva proceso (actividad, técnica). Se utilizan también los términos Pedazo de Método (*method chunk*) y Componente de Método (*method component*). En este artículo se utilizará el término Fragmento de Método el cual se caracteriza por tener exactamente un objetivo final que puede ser obtenido por una o más actividades que son realizadas utilizando una o más técnicas incluidas en ese fragmento, y que pueden generar uno o más artefactos o modelos. Respecto a la construcción de métodos, los enfoques más utilizados se resumen en la tabla 1.

Tabla 1. *Enfoques para la creación de Métodos en la ISM*. Adaptado de [H-SR10]

ENFOQUE	DESCRIPCIÓN
Basado en ensamblaje	Un método nuevo se construye a partir de fragmentos de métodos ya existentes. Los fragmentos pueden ser categorizados en fragmentos de producto y fragmentos de proceso. Este enfoque sigue la estrategia de reutilización.
Basado en la extensión	El método requerido es creado a partir de un método existente el cual se extiende haciendo uso de patrones y rellenando las faltas del método original. La ISM define estrategias y guías para llevar a cabo el proceso de extensión
Basado en paradigma	Este enfoque se fundamenta en el concepto que un nuevo método se obtiene a partir de una abstracción de un método existente, o por la instanciación de un metamodelo. Se le llama también enfoque basado en la evolución

Una revisión muy completa sobre la terminología y enfoques utilizados en la Ingeniería Situacional de Métodos se puede encontrar en [H-SR10].

## 3. Trabajos relacionados

En base a la revisión bibliográfica llevada a cabo, se identificaron distintas propuestas que consideran la utilización de métodos basado en las situaciones de desarrollo. Entre los trabajos más resaltantes se encuentra los de [VBSD06] y [SKHB09] en los cuales se sostiene que los métodos de desarrollo de aplicaciones Web no cubren las necesidades específicas para la implementación de sistemas manejadores de contenidos (*Content Management Systems*, CMS por sus siglas en inglés). Los autores presentan WEM (*Web Engineering Method*) como un método creado en base a los principios de la Ingeniería Situacional de Métodos utilizando en enfoque basado en el ensamblaje para la creación de un nuevo método para el diseño en el dominio de los CMS.

Siempre en el dominio de los CMS, destaca el trabajo de [VVP08]. Los autores consideran que el método OOWS adolece de expresividad para definir aplicaciones Web en algunos dominios como los sistemas manejadores de contenidos. Por lo tanto definen el metamodelo del método OOWS con el propósito de aplicar la Ingeniería Situacional de Métodos, de forma de mejorar el desarrollo de CMS.

En el enfoque llamado *Method Association*, propuesto por [LJS\*08] se selecciona y se construyen métodos a partir de cinco enfoques de modelado Web orientado a modelo (*model-driven*), para que se ajusten el dominio del proyecto de desarrollo.

Basado en [Lah07] el enfoque más apropiado para el desarrollo de aplicaciones Web es un método construido para una organización específica, con fragmentos de métodos reutilizables, y se adecua el método base de forma que pueda apoyar las características propias del proyecto. El autor propone un marco de referencia, basado en la Ingeniería de Métodos, que incluye el proceso para la construcción de métodos y un repositorio para los métodos, los fragmentos de métodos, las configuraciones, las reglas y el almacenamiento de las características de la situación de desarrollo.

En [ELW06] no se utilizan los principios de la ISM, sin embargo proponen algunas características relacionadas con el desarrollo de aplicaciones Web, como el número de usuarios, cantidad de miembros del equipo de desarrollo, costos de documentación, entre otros. Estos indicadores se utilizan para dimensionar e identificar la complejidad del desarrollo.

En [BVK07] basado en una experiencia en Eslovenia, se propone un enfoque orientado a la práctica, llamado Configuración de Procesos (*Process Configuration*). En su trabajo se expone como se crea un enfoque, efectivo y ajustado a las circunstancias del proyecto y la situación de desarrollo, para lograr un método específico para cada proyecto.

En [KSB10] se propone una colección de factores para caracterizar aplicaciones Web y la construcción de métodos nuevos a partir de fragmentos de métodos ya existentes

utilizando la Ingeniería Situacional de Métodos. Los autores además definen tres tipos de guías para los desarrolladores: i) la selección del modelo de proceso de diseño más apropiado; ii) la selección de los fragmentos de métodos más apropiados; y iii) la aplicación de los fragmentos de métodos seleccionados.

En el trabajo de [KDS07] se afirma que la selección de fragmentos de métodos es el proceso fundamental para la creación de nuevos métodos. La contribución significativa del trabajo es una tipología de características de proyectos de Ingeniería de Software, donde se consideran cuatro dimensiones: i) organización (importancia del proyecto, impacto, tamaño, nivel de innovación); ii) aspecto humano (resistencia a conflictos, experiencia, claridad, número de involucrados, entre otros); iii) dominio de la aplicación (rigurosidad y formalismo, interrelaciones, integración con otros sistemas, complejidad, tipo de aplicación); y iv) estrategia de desarrollo (organización del proyecto, estrategias de desarrollo, número de objetivos, entre otros).

En [SKB08] se proponen los siguientes factores para caracterizar una aplicación Web: tipo de aplicación, complejidad, similitud con otras aplicaciones, clarificación del problema a resolver, experiencia del equipo diseñador.

Otro enfoque utilizado para la caracterización de aplicaciones Web se describe en [KPRR06]. Su propuesta está basada en el estándar ISO/IEC 9126-1 que se utiliza para la evaluación de características de calidad de software. Este estándar considera tres dimensiones: producto, uso y desarrollo, y sus respectivos atributos. Los autores incluyen además, la Evolución, como una característica que afecta las tres dimensiones.

#### 4. Descripción general de la propuesta.

Una razón asociada con la flexibilidad de un método es la posibilidad que este pueda adecuarse a diferentes situaciones de desarrollo. “En el desarrollo de aplicaciones Web es imposible adherirse a un plan rígido y predefinido. Es vital reaccionar en forma flexible a los cambios de condiciones” [KPRR06, p. 16].

Durante el proceso de definición de una metodología para el desarrollo de aplicaciones Web – WEBFDM –, surgió la necesidad de incorporarle mecanismos para convertirla en una metodología flexible [Bia10]. La metodología tiene una herramienta CASE, llamada COHESIÓN, que provee el soporte necesario para el desarrollo de aplicaciones. Considerando las potencialidades y beneficios de la Ingeniería Situacional de Métodos, COHESIÓN fue reestructurada como una colección de fragmentos de método (Figura 1).

En la actualidad WEBFDM, con el soporte de COHESIÓN, provee varios fragmentos de método tales como: Captura y elicitación de requisitos, Motor Caracterización de Desarrollos Web, Definición de Casos de Uso, Identificación de actores, Modelado de aplicaciones CRUD, Generación de casos de uso a partir del Modelo de Datos, Soporte para trabajo Colaborativo, entre otros.

Por ejemplo, el fragmento de método llamado “Modelado de aplicaciones CRUD” tiene asociado las actividades para generar los modelos necesarios para crear

aplicaciones Web con las funcionalidades para llevar a cabo las operaciones Crear, Leer, Actualizar y Eliminar.

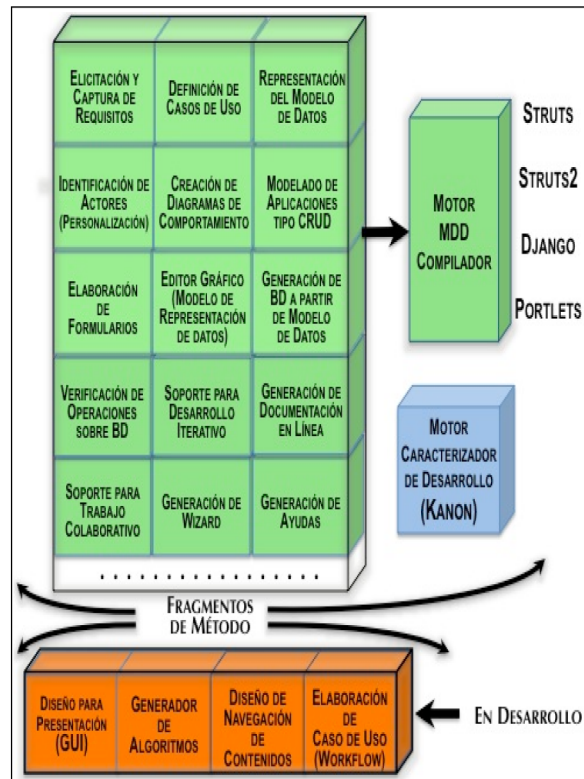


Figura 1: Colección de fragmentos de métodos en COHESIÓN

La herramienta CASE COHESIÓN incluye un motor orientado a modelo, el cual es el responsable de generar la aplicación Web a ser desarrollada para distintas plataformas de despliegue [DSB10]. La versión actual genera aplicaciones para Struts, Struts2, Django y Portlet.

El enfoque a proponer considera que la herramienta COHESIÓN será configurada de acuerdo a las características identificadas en la situación de desarrollo. Una situación es determinada por varios atributos asociados al producto final, el ambiente organización, el uso, la tecnología involucrada, entre otros.

A continuación se expondrá el marco de referencia para la caracterización del desarrollo de aplicación llamado KANON, que está implementado como un fragmento de método de COHESIÓN, y la definición de dimensiones y atributos que se consideran.

#### 4.1 El marco de referencia KANON

El “Motor Caracterizador del Desarrollo de Proyectos Web – KANON”, es el fragmento de método encargado de identificar y detectar las características de la aplicación Web a desarrollar y su situación de desarrollo. El enfoque de construcción utilizado en KANON es basado en paradigma. El marco de referencia que se propone en este artículo está basado en los fundamentos de la Ingeniería Web, el cual considera los siguientes conceptos acerca el



desarrollo de aplicaciones Web: niveles (los cuales incluyen el contenido, el hipertexto y la presentación), los aspectos (hacen referencia a la estructura y el comportamiento de cada elemento en los niveles), las fases de desarrollo y la posibilidad de personalizar cada uno de los conceptos anteriores.

En KANON una Situación de Desarrollo determina una nueva instancia del método a ser creado utilizando la colección de fragmentos de método de COHESIÓN.

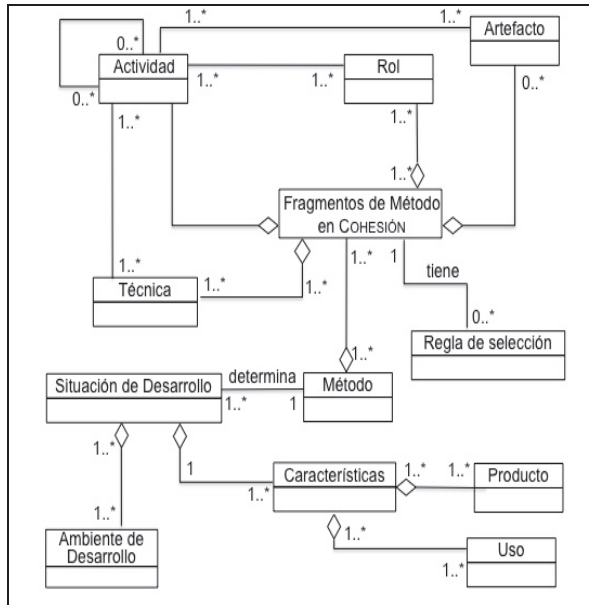


Figura 2: Metamodelo de KANON.

En el metamodelo que se muestra en la Figura 2, un nuevo método es determinado por la Situación de Desarrollo, como una combinación de las Características de la aplicación Web (Producto y Uso) y del Ambiente de Desarrollo. El concepto de Evolución en la aplicación viene considerado como otra situación de desarrollo. En la Tabla 2 se resumen las dimensiones y algunos atributos que son considerados en KANON.

El marco de referencia considera la Categoría de aplicación Web como uno de los atributos en la dimensión “Producto”, ya que dependiendo de la categoría de aplicación, se requerirán recursos y técnicas distintas para desarrollarla, y se deberán crear distintos modelos según el caso. Además la categorización de aplicaciones tiene implícito el nivel de complejidad, puesto que cada nueva categoría se relaciona con las funcionalidades, uso de nuevas tecnologías, experiencias de desarrollo, entre otras. KANON utiliza una relación Categoría-Modelos de forma que pueda guiar al diseñador en la creación de modelos y el modelo de proceso que debe llevarse a cabo. Todos los fragmentos de método de COHESIÓN están al mismo nivel de granularidad.

Cada fragmento de método en COHESIÓN tiene un conjunto de reglas y precondiciones. Además se incluye un descriptor asociado a cada atributo en las diferentes dimensiones. El descriptor especifica la existencia/ausencia (y en algunos casos valores) de cada atributo considerado en KANON. Si las condiciones de selección especificadas se

satisfacen, el fragmento de método es candidato para su incorporación en el nuevo método.

Tabla 2. Dimensiones y atributos utilizados en KANON

DIMENSIONES	DESCRIPCIÓN DE ATRIBUTOS
Producto	<ul style="list-style-type: none"> <li>• <b>Categoría</b> (orientada a contenido, interactiva, transaccional, colaborativa, portal etc.)</li> <li>• <b>Modelos</b> (Contenido, Hipertexto, Presentación, Adecuación, Requisitos, Casos de Uso)</li> <li>• <b>Complejidad</b> (Requisitos, Documentación, Criticidad, Tamaño/Escala)</li> <li>• <b>Dominio</b></li> </ul>
Uso	<ul style="list-style-type: none"> <li>• <b>Contexto Social</b> (usuarios, involucrados, personalización)</li> <li>• <b>Contexto Técnico</b> (redes, dispositivos, plataforma de despliegue)</li> <li>• <b>Contexto Natural</b> (disponibilidad, contexto físico: intranet, extranet, Internet)</li> </ul>
Ambiente de Desarrollo	<ul style="list-style-type: none"> <li>• <b>Equipo de desarrollo</b> (experiencia, multidisciplinario, conocimiento del dominio, ambiente, herramientas de apoyo)</li> <li>• <b>Infraestructura Técnica</b> (innovadora, novedosa, tradicional)</li> <li>• <b>Proceso</b> (modelo: ágil, disciplinado; planificación, recursos, compromisos, estándares de calidad)</li> <li>• <b>Integración con otros sistemas</b> (internos, externos)</li> </ul>
Evolución	Considerada como una nueva situación.

#### 4.2 Un ejemplo usando KANON

A continuación se describe un ejemplo de utilización del marco de referencia KANON. El mismo se basa en una aplicación Web a ser utilizada para una empresa de Consultoría. A continuación, en la Tabla 3 se listan los requisitos más significativos de la aplicación.

Tabla 3. Requisitos principales de una aplicación Web para una empresa de Consultoría

Propósito general	La empresa de Consultoría requiere de una aplicación Web de forma que todos sus consultores puedan registrar las horas trabajadas en cada proyecto en los cuales están involucrados, alrededor del mundo.
Datos	<p>Requisitos</p> <ol style="list-style-type: none"> <li>Cada consultor tiene un “usuario” y “contraseña” para ingresar a la aplicación Web.</li> <li>Los datos a ser registrados son: número de proyecto, fecha, horas trabajadas, detalles del trabajo desarrollado (actividades llevadas a cabo).</li> <li>Toda entrada de datos debe ser validada.</li> </ol>

Transaccionales	<p>a) Cada consultor puede crear y editar un nuevo registro de actividades, y tiene acceso únicamente a los proyectos que le corresponden.</p> <p>b) Cada transacción registrada por cada consultor debe actualizar el Sistema de Nómina y el Sistema de Facturación de la Empresa de Consultoría.</p>
Interfaz	<p>a) El estilo de interacción es basado en texto, incluyendo la captura de datos con formularios.</p> <p>b) Los consultores pueden utilizar la aplicación Web a través de sus computadores y desde sus teléfonos inteligentes, desde cualquier lugar, y a cualquier hora.</p>
No funcionales	<p>a) La aplicación debe estar disponible en línea en una semana.</p> <p>b) Los consultores deben participar directamente en el diseño de la aplicación.</p> <p>c) Se debe considerar la seguridad de los datos (información de los consultores y proyectos). La aplicación debe estar disponible 24/7.</p> <p>d) El equipo de trabajo es de 4 personas y están localizadas y trabajan en el mismo lugar.</p>

Los valores de los atributos resultantes de la aplicación del marco de referencia KANON, se resumen en la Tabla 4.

Tabla 4. Utilización del marco de referencia KANON

DIMENSIONES	RESULTADOS DE KANON
Producto	<p><b>Categoría:</b> Transaccional;</p> <p><b>Modelos a crear:</b> Modelo de Datos, Modelo de Presentación, Modelo de Casos de Uso mediante Diagramas de Comportamiento, Definición de Actores.</p> <p><b>Complejidad:</b> Baja, No crítica.</p>
Uso	<p><b>Contexto Social:</b> usuarios e involucrados conocidos;</p> <p><b>Contexto Técnico:</b> plataforma de despliegue dual (Teléfonos inteligentes y PC);</p> <p><b>Contexto Natural:</b> disponibilidad (24/7), intranet</p>
Ambiente de desarrollo	<p><b>Equipo de desarrollo:</b> con experiencia, equipo pequeño (4 miembros);</p> <p><b>Localización:</b> mismo lugar de trabajo;</p> <p><b>Herramienta de soporte:</b> disponible para Struts2;</p> <p><b>Infraestructura Tecnológica:</b> NA (no aplica);</p> <p><b>Proceso:</b> ágil; <b>Integración:</b> con sistemas legados (Nómina y Facturación).</p>

Los fragmentos de método seleccionados por el marco de referencia, de forma que se utilice en la creación de un nuevo método para el ejemplo, se muestra a continuación, en la Figura 3.

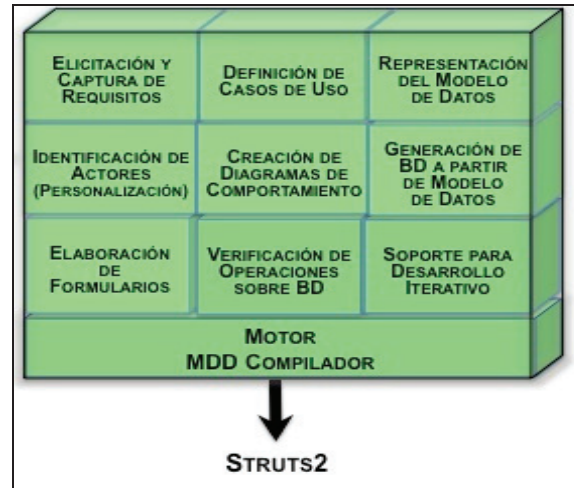


Figura 3: Colección de Fragmentos de Método requeridos para la aplicación Web del ejemplo.

La generación de modelos y artefactos (óvalos), las actividades a llevar a cabo (rectángulos) y el proceso que deberá llevarse a cabo para el ejemplo, se muestra en la figura 4.

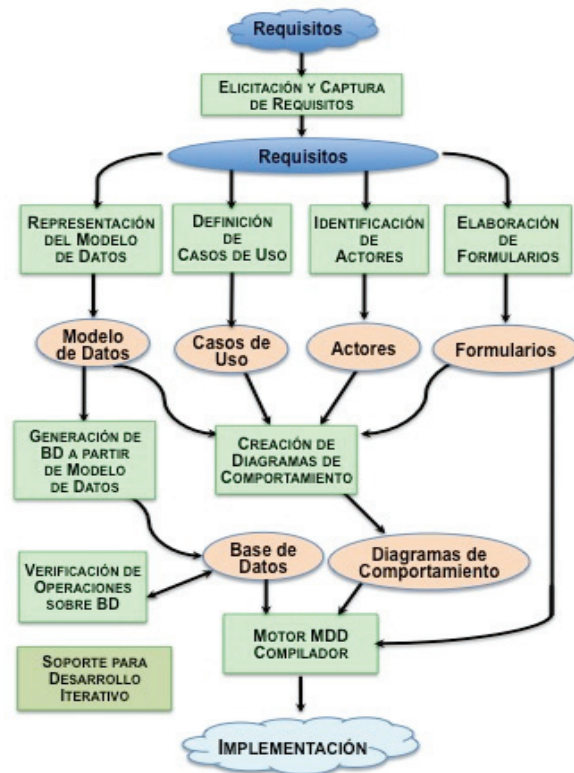


Figura 4: Configuración del nuevo método, con actividades y artefactos, para el ejemplo.

Como se observa en la figura anterior, el fragmento de método “Soporte para Desarrollo Iterativo” es un fragmento que brinda funcionalidades en la utilización de COHESIÓN y la metodología y no se muestran los artefactos que se generan por cuestión de espacio.

Utilizando el mismo caso, supóngase que apareciese un nuevo requisito, por ejemplo: “los miembros del equipo de desarrollo están localizados en lugares distintos del país”, entonces los atributos de la dimensión Ambiente de Desarrollo y toda la Situación de Desarrollo sufriría un cambio. En esta nueva situación, KANON seleccionará también el fragmento de método “Soporte al Ambiente Colaborativo” (ver Figura 2) de forma que el equipo de trabajo pueda hacer uso de la funcionalidad que brinda COHESIÓN para el trabajo colaborativo en el uso de la metodología.

Si el ejemplo a realizar hubiese sido una aplicación Web solo con funcionalidades CRUD (*Create, Read, Update, Delete*), los únicos fragmentos de método seleccionados por KANON serían: Elicitación y Captura de Requisitos, Representación del Modelo de Datos, Generación de Base de Datos a partir de Modelo de Datos, Elaboración de Formularios, Modelado de Aplicaciones tipo CRUD y el Motor MDD Compilador.

## 5. Conclusiones y trabajos a futuro

La contribución principal de este artículo es el marco de referencia, KANON, para caracterizar la situación de desarrollo de aplicaciones Web. Este marco de referencia define una colección de atributos relacionados con el Producto a desarrollar, su Uso y el ambiente de desarrollo, de forma que se pueda determinar la Situación de Desarrollo para un proyecto.

Las dimensiones identificadas y sus respectivos atributos determinan la combinación de Fragmentos de método, para crear un nuevo método ajustado a la situación de desarrollo real. KANON está siendo implementado en COHESIÓN como un fragmento de método.

El marco de referencia también se utiliza para ayudar a los diseñadores en la utilización de la metodología WEBFDM con el soporte de COHESIÓN. Si ocurriese un cambio en la colección de fragmentos disponibles en COHESIÓN, entonces el marco de referencia puede gestionar el cambio en un método ya creado.

WEBFDM con el soporte de COHESIÓN incorpora las características y mecanismos de configuración necesarios para hacer que la metodología sea más flexible, con desarrollo orientado a modelo, y capaz de ajustar el proceso de desarrollo, la creación de modelos y la implementación del producto final basado en una situación dada. En la actualidad el marco de referencia se está validando a través de la caracterización de diferentes aplicaciones Web que se han desarrollado en la universidad.

El marco de referencia propuesto en este artículo incluye nuevos atributos y una organización distinta de las dimensiones que las utilizadas por [KPRR06]. Los nuevos atributos están relacionados con el Dominio y la relación Categoría-Modelo.

Comparada con la propuesta de [KDS07], KANON extiende las características de un proyecto porque considera factores asociados a los requisitos de la Ingeniería Web como son el modelo de hipertexto y el de presentación.

La propuesta extiende además la colección de factores identificados por [SKB08] y por [KSN10]. Estos trabajos consideran los siguientes tipos o categorías de aplicaciones: kiosco (orientado a contenido), sistemas de información Web, aplicaciones adaptativas y aplicaciones transaccionales (*e-commerce*). En KANON se considera una mayor variedad de categorías: desde las aplicaciones orientadas a contenidos hasta aplicaciones Web colaborativas, ubicuas y portales.

El próximo paso en esta investigación es la extensión de KANON de forma que se puedan incluir nuevos atributos y mecanismos para caracterizar el desarrollo de aplicaciones Web, visto que la tecnología, el uso y las plataformas de despliegue cambian y evolucionan constantemente.

## Referencias

- [BBOS07] BIANCHINI A., BLANCH R., ORTEGA M., SUÁREZ A.: Diseño de una herramienta para el desarrollo de aplicaciones Web basadas en Struts. Proceedings CIAWI IADIS Iberoamericana. Vila Real, Portugal. Diciembre 2007.
- [BOS05] BIANCHINI A., ORTEGA M., SUÁREZ A.: Una Metodología de Diseño de Aplicaciones Web bajo el patrón MVC. Actas del XIII Encuentro Chileno de Computación. Jornadas Chilenas de Computación 2005. Universidad Austral de Chile, Valdivia. Chile. Noviembre 2005.
- [Bia10] BIANCHINI A.: WEBFDM: a Web application Flexible Development Methodology. Proceedings ICWI 2010. Timisoara, Rumania, pp. 427-431. October 2010.
- [BVK07] BAJEC M., VAVPOTIC D., KRISPER M.: Practice-driven approach for creating project-specific software development methods. Journal of Information and Software Technology. Vol. 49, Issue 4, pp. 345-365. 2007.
- [Bri96] BRINKKEMPER S.: Method engineering: engineering of information systems development methods and tools. Information & Software Technology, 38(4), pp. 275-280. 1996.
- [CFB00] CERI S., FRATERNALI P., BONGIO A.: Web modeling language (WebML): a modeling language for designing web sites. WWW9 Computer Networks, 33, (1-6), pp. 137-157. 2000.
- [DSB10] DÍAZ P. C. A., SUÁREZ A., BIANCHINI A.: COHESIÓN: A Model-Driven Development Tool for Web Applications. Internal Report. Computer Science and Information Technology Department. Universidad Simón Bolívar. Caracas. September 2010. Disponible en <http://www ldc.usb.ve/~abianc/cohesion/report e-tecnico-1.pdf>
- [ELW06] ENGELS G., LOHMANN M., WAGNER A.: The Web Application Development Process. In



- Kappel G., Pröll B., Reich S., Retschitzegger W. (Editors): *Web Engineering - The Discipline of Systematic Development of Web Applications*, pp. 197-218. John Wiley and Sons Ltd. 2006.
- [H-SR10] HENDERSON-SELLERS B., RALYTYÉ J.: Situational Method Engineering: State-of-the-Art Review. *Journal of Universal Computer Science*, Vol. 16, Nº. 3, pp. 424-478. 2010.
- [KPRR06] KAPPEL G., PRÖLL B., REICH S., RETSCHITZEGGER W.: An Introduction to Web Engineering. In Kappel, G., Pröll, B., Reich, S. and Retschitzegger, W. (Eds), *Web Engineering: The Discipline of Systematic development of Web Application*. John Wiley and Sons, NY, pp. 1-17. 2006.
- [KK02] KOCH N., KRAUS A.: The Expressive Power of UML-based Web Engineering. *Proceedings of 2nd International Workshop on Web-oriented Software Technology (IWWOST 02)*. 2002.
- [KDS07] KORNYSHOVA E., DENECKERE R., AND SALINESI C.: Method Chunks Selection by Multicriteria Techniques: an Extension of the Assembly-based Approach. In *IFIP Vol. 244, Situational Method Engineering: Fundamentals and Experiences*, Eds. Ralyte, J., Brinkkemper, S., Henderson- Sellers B., pp. 64-78. 2007.
- [KSB10] KRAIEM N., SELMI S., BEN GHEZALA H.: A Situational Approach for Web Applications Design. *IJCSI International Journal of Computer Science Issues*. Vol. 7, Issue 3, No. 1, pp. 37-51. May 2010.
- [Lah07] LAHAJNAR S.: A Framework for Situational Web Methods Engineering. *Proceeding of the 7<sup>th</sup> International Conference on Web Engineering ICWE 2007*. Springer-Verlag Berlin, pp. 569-574. 2007.
- [LJS\*08] LUINENBURG L., JANSEN S., SOURER J., VAN DER WEERD, I., BRINKKEMPER, S.: Designing Web Content Management Systems Using the Method Association Approach. *Proceedings 4<sup>th</sup> International Workshop on Model-driven Web Engineering MDWE 2008*.
- [PFP03] PASTOR O., FONS J., PELECHANO, V.: OOWS: A method to develop web applications from web-oriented conceptual models. *Proceedings of 3rd International Workshop on Web Oriented Software Technology*. Oviedo: Luis Olsina, Oscar Pastor, Gustavo Rossi, Daniel Schwabe Editors. 2003.
- [SKB08] SELMI S., KRAIEM N., BEN GHEZALA, H. : Guidance in Web Applications Design. *Proceedings of Model Driven Information Systems Engineering: Enterprise, User and System Models MoDISE-EUS 2008*, pp. 114-125. Montpellier, France.
- [SR95] SCHWABE D., ROSSI G.: The Object Oriented Hypermedia Design Model. *Communications of the ACM*, Vol. 38, #8, pp. 45-46, August 1995.
- [SK06] SCHWINGER W., KOCH N.: Modeling web applications, in Kappel, G., Pröll, B., Reich, S. and Retschitzegger, W. (Eds), *Web Engineering – Systematic Development of web Applications*, John Wiley and Sons, NY, pp. 39-64.
- [SKHB09] SOURER J., KUPERS T., HELMS, R., BRINKKEMPER, S. 2009. *Model-Driven Web Engineering for the Automated Configuration of Web Content Management Systems*. *Proceeding of International Conference of Web Engineering ICWE 2009*. Spain. Springer Verlag, pp. 121-135. June 2009.
- [VVP08] VLAANDEREN K., VALVERDE F., PASTOR, O.: Improvement of a Web Engineering Method Applying Situational Method Engineering. In J. Cordeiro & J. Filipe (Eds.), *Proceedings of the 10th International Conference on Enterprise Information Systems (ICEIS 2008)* (Vol. ISAS-1, p. 147-154). Spain, 2008.
- [VBSV06] VAN DER WEERD, I., BRINKKEMPER S., SOURER, J., VERSENDAAL, J.: A situational implementation method for web-based content management system-applications: Method Engineering and Validation in Practice. *Software Process: Improvement and Practice* 11(5), pp. 521-538. 2006
- [Zie09] ZIEMER S.: Trade-offs for Web Application Development: Understanding and Improving Current Industrial Practices. *Doctoral theses*. Norwegian University of Science and Technology. 2009.