



UNIVERSIDAD CATÓLICA "NUESTRA SEÑORA DE LA ASUNCIÓN"
FACULTAD DE CIENCIAS Y TECNOLOGÍA

PROYECTO FINAL DE CARRERA

INGENIERÍA INFORMÁTICA

PRIMERA DEFENSA

Un Enfoque MDD para el desarrollo de RIA

Alumno:

Lic. Guido NUÑEZ

Tutores:

Ing. Magalí GONZÁLEZ

D. Sc. Luca CERNUZZI

Enero, 2016

Asunción, Paraguay

1. Introducción

La web ha ido evolucionando continuamente con el pasar de los años. Anteriormente las aplicaciones web estaban caracterizadas por un procesamiento realizado del lado del servidor donde el cliente solo se encargaba de solicitar y desplegar contenido estático. Esto generaba interacciones muy limitadas con el usuario, necesidad de recargar toda la página para realizar la navegación, tiempos de respuesta muy largos y dificultad de desplegar animaciones y contenidos multimedia.

Para enfrentarse a estas limitaciones han aparecido las Aplicaciones Enriquecidas de Internet (RIA). Este nuevo tipo de aplicaciones ha permitido mejorar notablemente la experiencia del usuario principalmente en cuanto a aspectos de presentación e interacción [1]. Las RIA poseen cuatro características principales que son la distribución de datos, la distribución de la lógica de negocios, la comunicación asíncrona entre cliente y servidor y la mejora de la interfaz de usuario. Se han propuesto muchas tecnologías para facilitar el desarrollo de RIA y las mismas han logrado incrementar la productividad del desarrollador, sin embargo, presentan desventajas como la carencia de instrumentos de alto nivel que abstraigan detalles de implementación, la propensión a errores, la falta de consideración del ciclo de vida completo del software, entre otras más [2] [3].

Los enfoques de Desarrollo Dirigido por Modelos (MDD) ofrecen soluciones a estas limitaciones, permitiendo crear aplicaciones a través de la especificación de modelos y la posterior generación de código a partir de ellos, logrando así un mayor nivel de abstracción [4]. Es conveniente que estos enfoques adopten un estándar como la Arquitectura Dirigida por Modelos (MDA) para dirigir el proceso de desarrollo y garantizar una mayor interoperabilidad y portabilidad entre sistemas [5].

En los últimos años, han surgido numerosos enfoques MDD para la implementación de RIA, muchos de ellos extendiendo enfoques para el desarrollo de aplicaciones web tradicionales. MoWebA es un enfoque MDD para el desarrollo de aplicaciones web tradicionales que adopta el estándar MDA y que permite la definición de un Modelo Específico de Arquitectura (ASM) para la implementación de extensiones [6]. En [7] se ha elaborado una extensión para la arquitectura RIA contemplando aspectos de la capa de presentación. Otra extensión conveniente sería abarcar aspectos de las capas de datos, lógica de negocios y comunicación de las RIA.

El problema que se busca resolver es la necesidad de mejorar la experiencia del usuario en la web, potenciando el desarrollo de RIA y dotando a las aplicaciones web tradicionales con características propias de las RIA. Esta necesidad surge debido a que los usuarios de la web se van acostumbrando a mejores apariencias gráficas, patrones de interacción avanzados, contenidos multimedia y menores tiempos de respuesta. En consecuencia, hoy en día se solicita a desarrolladores proporcionar interacciones ricas en casi cualquier aplicación web.

Para atacar el problema, se tiene como objetivo general definir un enfoque MDD para el desarrollo de RIA que extienda el enfoque MoWebA y se centre en la implementación de las características de distribución de datos, distribución de lógica de negocios y comunicación asíncrona entre cliente y servidor de las RIA. Este objetivo comprende los siguientes objetivos específicos:

- Analizar los principales enfoques MDD para el desarrollo de RIA que extienden enfoques para el diseño y desarrollo de aplicaciones web tradicionales.
- Proponer metamodelos para la generación de un ASM y reglas de transformación para la generación de código que contemplen las características de distribución de datos, distribución de lógica de negocios y comunicación asíncrona entre cliente y servidor de las RIA.
- Realizar un análisis de la propuesta a partir de un caso de prueba.

El presente proyecto final de carrera está siendo realizado en marco del proyecto “Mejorando el proceso de desarrollo de software: Una propuesta basada en MDD” y está financiada por el Consejo Nacional de Ciencias y Tecnología (CONACYT).

Lo que resta del documento está organizado de la siguiente manera: la sección 2 introduce la definición de RIA, con sus características principales y tecnologías para su desarrollo. La sección 3 presenta los conceptos de MDD, MDA y el enfoque MoWebA. En la sección 4 presentamos un análisis de proyectos RIA que abarcan las características de distribución de datos, distribución de lógica de negocios y comunicación asíncrona entre cliente y servidor, a partir de un mapeo sistemático de la literatura realizado para la recolección de información y presentando un cuadro comparativo con los enfoques de desarrollo obtenidos. Finalmente, la sección 5 describe la propuesta de este proyecto final de carrera y su estado actual de ejecución.

2. Las Aplicaciones Enriquecidas de Internet: RIA

En los inicios de la World Wide Web (WWW) las aplicaciones de internet poseían una arquitectura cliente-servidor, caracterizándose por una navegación a través de links, recarga entera de la página ante cualquier interacción del usuario, elementos de interfaz poco interactivos, almacenamiento de datos y lógica de negocios implementados totalmente del lado del servidor y una comunicación síncrona entre las partes. Estos factores afectaban notablemente a la experiencia del usuario en cuanto a aspectos de presentación e interacción y generaban tiempos de respuesta prolongados.

Para contrarrestar estas limitantes de las aplicaciones web tradicionales aparecen las Aplicaciones Enriquecidas de Internet. Brambilla et al. definen a las RIA (del inglés Rich Internet Application) como “aplicaciones web que explotan el poder de clientes web para incrementar la adaptabilidad y usabilidad de las interfaces de usuario, ofreciendo funcionalidades similares a las de aplicaciones de escritorio. Las RIA siguen el paradigma cliente-servidor, pero a diferencia de las aplicaciones web tradicionales, ellas son capaces de transferir el procesamiento de la interfaz de usuario, lógica de negocio y manejo de datos al cliente, utilizando posiblemente comunicaciones asíncronas” [8].

El término RIA apareció por primera vez en un documento de Macromedia en marzo del año 2002 donde se introdujo la tecnología Macromedia Flash MX [9]. A partir de allí empezaron a realizarse investigaciones y desarrollarse tecnologías que permitieron su implantación.

Las RIA son una combinación de la presentación e interactividad de aplicaciones de escritorio con la arquitectura y forma de distribución de las aplicaciones web [10]. La capa de presentación con su interacción asociada es trasladada al cliente, la computación de la página y datos de la misma se distribuyen entre servidor y cliente y la comunicación se realiza mayormente en forma asíncrona. Este esquema permite obtener elementos de interfaz más interactivos, evitar la recarga innecesaria de la página, soportar contenidos multimedia, evitar transferencia de datos redundantes y minimizar tiempos de respuesta mejorando notablemente la experiencia del usuario.

Como ejemplos de RIA en la actualidad tenemos aplicaciones como Facebook¹, Flickr², Google Docs³, Google Maps⁴, SlideRocket⁵ y Youtube⁶.

¹ Facebook. <https://www.facebook.com/>

² Flickr. <https://www.flickr.com/>

³ Google Docs. <https://docs.google.com/>

⁴ Google Maps. <https://www.google.com/maps/>

⁵ SlideRocket. <http://www.sliderocket.com/>

2.1 Características de las RIA

Las RIA poseen cuatro características principales [10] [11]:

Distribución de datos: además del almacenamiento de datos del lado del servidor propio de las aplicaciones web tradicionales, las RIA soportan almacenamiento del lado del cliente. Los datos en el cliente pueden ser temporales o persistentes permitiendo su manipulación y preparación antes de su envío al servidor. Esta característica de distribución de datos permite realizar una validación de datos local y el uso offline de la aplicación.

Distribución de lógica de negocios: las RIA permiten realizar operaciones complejas del lado del cliente. Estas operaciones incluyen ordenado y filtrado de datos antes de enviarlos al servidor, validación en vivo (acompañado de almacenamiento de datos local), uso offline, navegación y reordenamiento de la página.

Comunicación asíncrona: es posible utilizar comunicaciones síncronas y asíncronas. Tanto el cliente como el servidor son capaces de iniciar la comunicación. La comunicación asíncrona permite la recarga de partes de la página y operaciones del tipo push y pull.

Mejora de interfaz de usuario: la presentación y comportamiento de la interfaz de usuario de las RIA se ven altamente enriquecidos. Existen widgets y contenedores modernos y personalizables que mejoran la interfaz de usuario. Una página está compuesta de subpáginas donde cada una maneja la interacción del usuario de forma independiente actualizando y desplegando elementos individuales de la interfaz de usuario. Estas mejoras permiten la implementación de animaciones, multimedia y operaciones del tipo drag&drop.

2.2 Tecnologías para el desarrollo de RIA

Podemos desarrollar las RIA a partir de diversas tecnologías. Estas tecnologías se diferencian unas de otras en la forma en que la RIA es implementada y desplegada al usuario. La clasificación de estas tecnologías puede también ser utilizada para clasificar a las RIA en sí.

A continuación se presentan las diferentes tecnologías y su clasificación [10] [3] [12]:

Basadas en Scripting: las páginas web incluyen scripts (entendibles por navegadores modernos) que hacen referencia a elementos de presentación. HTML y CSS (Cascading Style Sheets) son utilizados para la definición y personalización de elementos de la interfaz y Javascript se utiliza para la creación de los scripts que definen la lógica del lado del cliente posibilitando solicitudes asíncronas al servidor. La técnica mayormente utilizada es AJAX (Asynchronous Javascript And XML). Una gran variedad de frameworks han sido desarrollados para superponer la complejidad de este tipo de aplicaciones y superar incompatibilidades con navegadores, entre ellos tenemos Backbone⁷, Dojo⁸, GWT⁹, jQuery¹⁰, Prototype¹¹ y la propuesta del W3C (World Wide Web Consortium) HTML5.

Basadas en Plugin: se ejecutan en plugins de navegadores que pueden venir instalados previamente o requieran la instalación por parte del usuario. Este enfoque provee un mejor desempeño al de la

⁶ Youtube. <https://www.youtube.com/>

⁷ Backbone. <https://www.backbone.com/>

⁸ Dojo. <https://dojotoolkit.org/>

⁹ GWT. <http://www.gwtproject.org/>

¹⁰ jQuery. <https://jquery.com/>

¹¹ Prototype. <http://prototypejs.org/>

utilización de Javascript y dota a la aplicación con procesamiento de eventos y renderizado avanzado. Las tecnologías para la creación de estas aplicaciones son Adobe Flash¹², Adobe Flex¹³, OpenLaszlo¹⁴, entre otras.

Ambientes de Ejecución Específicos: ejecutados en una aplicación independiente lanzada desde el navegador. Proponen lo mejor en cuanto a uso offline de la aplicación, acceso al sistema operativo y almacenamiento en el cliente. Las tecnologías más utilizadas son Adobe AIR¹⁵, JavaFX¹⁶ y Java Web Start¹⁷.

Basadas en Navegador: especificadas en un lenguaje declarativo construido a partir de XML (eXtensible Markup Language) reconocido por un navegador. A partir de este lenguaje los desarrolladores definen los distintos elementos e interacciones correspondientes. No requiere la extensión del navegador, sin embargo, es posible que las aplicaciones no puedan ser desplegadas en otros navegadores. Una tecnología utilizada para estas aplicaciones es Mozilla XUL¹⁸.

Mediante estas tecnologías podemos implementar las características RIA presentadas en la sección anterior. Como podemos observar en la Tabla 1 las tecnologías basadas en plugin y en ambientes de ejecución específicos cubren todas las características RIA y las basadas en scripting tienen limitaciones en cuanto a contenidos multimedia y persistencia de datos en el cliente.

Características RIA Tecnologías RIA	Presentación Enriquecida	Almacenamiento de Datos en el Cliente	Lógica de Negocios en el Cliente (y Distribuida)	Comunicación Cliente-Servidor
Basadas en Scripting	Limitada: Sin multimedia	Limitado: Sin persistencia	Si	Si
Basadas en Plugin	Si	Si (mediante plugins adicionales)	Si	Si
Ambientes de Ejecución Específicos	Si	Si	Si	Si

Tabla 1: Características cubiertas por los tipos de tecnologías.

Las tecnologías basadas en navegador no son consideradas debido a que son raramente utilizadas en la práctica.

Si bien las tecnologías basadas en scripting no cubren todas las características, éstas son las más utilizadas para desarrollar RIA. Esto se da debido a que la técnica utilizada (AJAX) es bien aceptada en la comunidad de desarrolladores ya que posee un standard bien establecido y es de código abierto. Además no requiere instalación de software por parte de los usuarios dejando el procesamiento de la aplicación en manos del navegador. Para sobreponerse a las limitaciones, éstas tecnologías pueden combinarse con las tecnologías basadas en plugin y así obtener mejores representaciones de video y almacenamiento local.

3. El Desarrollo Dirigido por Modelos: MDD

Brambilla et al. definen al desarrollo dirigido por modelo o MDD (del inglés Model Driven Development) como “un paradigma de desarrollo que utiliza modelos como artefactos primarios en el

¹² Adobe Flash. <http://www.adobe.com/la/products/flash.html>

¹³ Adobe Flex. <http://www.adobe.com/la/products/flex.html>

¹⁴ OpenLaszlo. <http://openlaszlo.org/>

¹⁵ Adobe Air. <http://www.adobe.com/la/products/air.html>

¹⁶ JavaFX. <http://docs.oracle.com/javafx/>

¹⁷ Java Web Start. <http://www.oracle.com/technetwork/java/javase/javawebstart/index.html>

¹⁸ Mozilla XUL. <https://developer.mozilla.org/es/docs/XUL>

proceso de desarrollo”. Además destacan: “por lo general, en MDD la implementación es generada de forma (semi)automática a partir de los modelos” [4].

MDD posee tres pilares fundamentales: los modelos, las transformaciones y las herramientas. Los modelos constituyen el núcleo dentro del proceso de desarrollo, éstos dejan de ser solo documentación para ahora formar parte de la implementación del sistema, están escritos en lenguajes bien definidos y a diferentes niveles de abstracción. Las transformaciones son aplicadas sobre los modelos de modo a obtener modelos más concretos o código. Las herramientas facilitan la creación de modelos y la especificación y posterior ejecución de transformaciones.

La idea es partir de modelos bien abstractos, realizar una serie de transformaciones modelo a modelo para obtener modelos más concretos hasta finalmente realizar una transformación modelo a texto para la obtención de la implementación del sistema. La iniciativa MDD busca obtener un mayor nivel de abstracción en la solución de problemas, aumentar la confianza en la automatización de tareas y el uso de estándares para facilitar las actividades de desarrollo [13].

Si bien las RIA pueden desarrollarse siguiendo un enfoque basado en código o basado en frameworks (más utilizado comúnmente), realizar un trabajo en donde se adopte el enfoque MDD resulta interesante, considerando de que según sus propulsores el mismo aporta las siguientes ventajas [13] [14]:

- Incremento de la productividad de desarrolladores.
- Mejora de la calidad de software generado.
- Mayor reusabilidad de modelos y transformaciones.
- Contempla más fases del ciclo de vida del software.
- Separación de la especificación e implementación de un sistema.
- Manejo de complejidad a través de abstracciones de alto nivel.
- Adaptación a cambios de requerimientos y cambios tecnológicos.
- Mejora de la comunicación con los usuarios y entre desarrolladores
- Mejor interoperabilidad entre sistemas.
- Mayor portabilidad entre plataformas.

3.1 Arquitectura Dirigida por Modelos

La Arquitectura Dirigida por Modelos o MDA (del inglés Model Driven Architecture) es la propuesta de desarrollo dirigido por modelos de la OMG (Object Management Group). Consiste en una arquitectura general que provee lenguajes de modelado y transformaciones estándares para un proceso MDD.

MDA especifica tres niveles de abstracción [4]:

CIM (Computation-Independent Model): modelo de la aplicación sin tener en cuenta aspectos computacionales.

PIM (Platform-Independent Model): definición de la estructura y comportamiento de la aplicación sin tener en cuenta la plataforma destino.

PSM (Platform-Specific Model): definición de la estructura y comportamiento de la aplicación para una plataforma determinada.

Para un CIM dado es posible tener más de un PIM asociado o “mapeado”. A su vez, cada PIM puede mapearse a más de un PSM.

Entre los estándares propuestos por MDA se encuentran MOF¹⁹ (MetaObject Facilities), UML²⁰ (Unified Modeling Language), CWM²¹ (Common Warehouse Metamodel), XMI²² (XML Metadata Interchange) y CORBA²³ (Common Object Request Broker Architecture).

MDA establece que el proceso de desarrollo de software debe empezar con la especificación del PIM, mediante un lenguaje de modelado basado en MOF. A través de herramientas de desarrollo MDA se transforma el PIM en PSM y luego se transforma el PSM en la implementación deseada. Esta implementación puede ser C#.NET, CORBA, EJB, XML/SOAP, entre otras [5].

MDA busca derivar valor a partir de los modelos, establecer una mejor interoperabilidad y portabilidad entre sistemas y separar el diseño de una aplicación en diferentes fases y modelos que permitan estandarizar y mejorar el proceso de desarrollo de software.

3.2 MoWebA

MoWebA (Model Oriented Web Approach) es un enfoque MDD para el desarrollo de aplicaciones web. El enfoque adopta el estándar MDA, define diferentes aspectos metodológicos, se basa en una arquitectura por capas, propone metamodelos, y se complementa con herramientas de modelado, transformación y generación de código [6].

A diferencia de la mayoría de los enfoques que proveen un modelado basado en una navegación orientada a datos partiendo desde un modelo conceptual o de datos, MoWebA concibe una navegación orientada a funciones permitiendo que el proceso de modelado empiece desde un modelo navegacional.

MoWebA se basa fuertemente en una separación de conceptos buscando proponer un modelado orientado a la navegación, automatización de tareas, adopción de estándares y el manejo de la evolución de ambientes web considerando las nuevas tendencias tecnológicas. Mediante esto se busca lograr una mayor interoperabilidad y extensibilidad de sistemas web.

El enfoque define dos tipos de procesos complementarios: el proceso de modelado y el proceso de transformación que serán explicados en las secciones siguientes.

3.2.1 Proceso de Modelado

El proceso de modelado consiste en la especificación de los diferentes diagramas que representan a la aplicación web. Se consideran las abstracciones CIM, PIM y PSM descritas por MDA. Además se presenta una abstracción más llamada ASM (Architecture Specific Model). El CIM es utilizado para la especificación de los requerimientos funcionales. El PIM define cinco tipos de modelos generales: Modelo de Entidad, Modelo Navegacional, Modelo de Comportamiento, Modelo de Presentación y Modelo de Usuario. A su vez, cada uno de estos modelos define uno o más diagramas para la especificación de sus funcionalidades. El ASM introducido por MoWebA sirve para enriquecer a los modelos con información acerca de una arquitectura específica, como por ejemplo REST, RIA, SOA

¹⁹ MOF. <http://www.omg.org/mof/>

²⁰ UML. <http://www.uml.org/>

²¹ CWM. <http://www.omg.org/cwm/>

²² XMI. <http://www.omg.org/spec/XMI/>

²³ CORBA. <http://www.corba.org/>

(Service Oriented Applications). El PSM se utiliza para añadir a los modelos consideraciones específicas para cierta plataforma. La especificación de estas abstracciones se realiza de manera progresiva y sistemática en siete etapas. Las etapas del 1 a 6 contemplan el CIM y el PIM, mientras que la etapa 7 se refiere al ASM y PSM. Cada etapa define una serie de diagramas definidos según las dependencias entre modelos, nivel de granularidad de la tarea a realizar y el tipo de modelado.

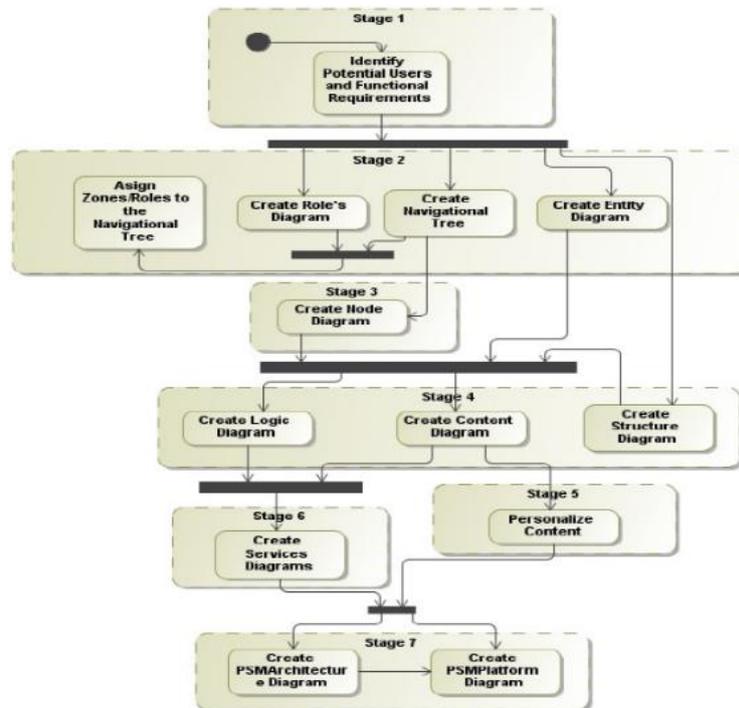


Figura 1: Proceso de modelado en MoWebA.

3.2.2 Proceso de Transformación

El proceso de transformación implica las actividades y el uso de herramientas necesarias para realizar la transformación de la especificación inicial en la implementación final a lo largo del proceso de desarrollo. Existen dos fases esenciales: la transformación del PIM al ASM/PSM y la transformación del ASM/PSM al ISM (Implementation Specific Model). El ISM consiste en la implementación generada automáticamente sin intervención manual del desarrollador. La fase de PIM-ASM/PSM consiste en una transformación de modelo a modelo (M2M) realizada de forma semiautomática, mientras que la fase de ASM/PSM-ISM es una transformación de modelo a texto (M2T) realizada de forma automática.

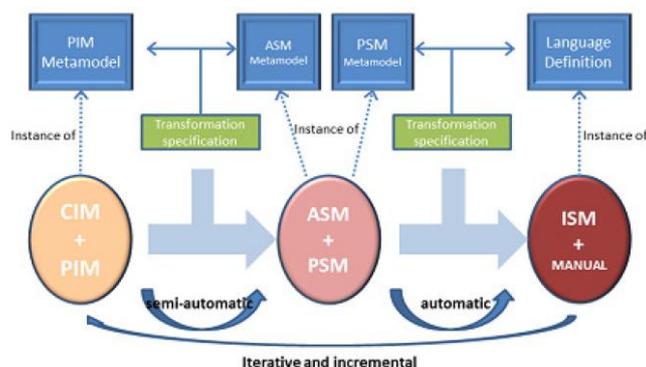


Figura 2: Proceso de transformación en MoWebA.

4. Análisis de Proyectos RIA para la Distribución de Datos, Distribución de Lógica de Negocios y Comunicación Asíncrona entre Cliente y Servidor: un Mapeo Sistemático de Literatura

Esta sección presenta en primer lugar el procedimiento realizado para la obtención de información y posteriormente un análisis de los resultados obtenidos acompañado de una tabla comparativa. A partir de esto se determinan debilidades potenciales de los enfoques seleccionados, dando lugar a nuestra propuesta de solución.

La recolección de información y el análisis de proyectos fueron realizados siguiendo los pasos de un mapeo sistemático de literatura.

Un mapeo sistemático de literatura consiste en una revisión amplia de estudios primarios en un área de interés para la identificación de evidencias en el área [15]. A diferencia de una revisión sistemática de literatura que es construida a partir de estudios meramente empíricos e identifica mejores prácticas con respecto a procedimientos, tecnologías, métodos o herramientas, el mapeo consiste en una visión más global, proponiendo una clasificación y un análisis temático de la literatura sobre el área de interés [16].

Es por ello, que hemos optado por realizar una aproximación a un mapeo sistemático de literatura para obtener información y analizar los distintos proyectos para el desarrollo de RIA que abarcan las características de distribución de datos, distribución de lógica de negocios y comunicación asíncrona entre cliente y servidor.

Se ha elaborado un protocolo de revisión que se describe a continuación, estableciendo la pregunta de investigación, el proceso de búsqueda, los criterios de inclusión y exclusión y el modo de aplicarlos. Este protocolo fue analizado y aprobado por los miembros del grupo de investigación del proyecto.

Se han definido dos preguntas de investigación que guían el proceso, son las siguientes:

¿Qué enfoques de desarrollo dirigido por modelos existen para el desarrollo de características de distribución de datos, distribución de procesos y comunicación asíncrona entre cliente y servidor de aplicaciones enriquecidas de internet?

¿Qué lenguajes de modelado, herramientas, tecnologías utilizan los enfoques identificados?

A partir de esta pregunta se identificaron los términos principales y sus términos alternativos para la elaboración de la cadena de búsqueda siguiente:

(Model Driven OR MDD) AND (Rich Internet Applications OR RIA)

Esta cadena fue aplicada en cinco librerías digitales elegidas debido a sus aportes relevantes a la ingeniería de software: IEEEExplore²⁴, ACM Digital Library²⁵, ScienceDirect²⁶, SpringerLink²⁷ y Google Scholar²⁸.

²⁴ IEEEExplore. <http://ieeexplore.ieee.org/>

²⁵ ACM Digital Library. <http://dl.acm.org/>

²⁶ ScienceDirect. <http://www.sciencedirect.com/>

²⁷ SpringerLink. <http://link.springer.com/>

²⁸ Google Scholar. <https://scholar.google.com/>

Se buscaron artículos de conferencias, revistas y capítulos de libros en los idiomas inglés y español publicados hasta agosto de 2015.

Sobre los resultados obtenidos se aplicaron los siguientes criterios de inclusión:

1. Enfoques o extensiones de enfoques de desarrollo basados o dirigidos por modelos de RIA.
2. Aplicaciones o usos de enfoques de desarrollo de RIA.
3. Estados del arte, mapeos sistemáticos de literatura, estudios comparativos y de evaluación de distintos enfoques de desarrollo de RIA.
4. Herramientas o extensiones de herramientas utilizadas por enfoques de desarrollo de RIA.

Y además fueron aplicados los siguientes criterios de exclusión:

1. El enfoque no es dirigido por modelos o basado en modelos.
2. Desarrollo de aplicaciones que no son RIA (Web 2.0 puede incluirse).
3. Duplicados.
4. No contempla las características de distribución de datos, distribución de procesos o comunicación asíncrona entre cliente y servidor, o bien, no incluye información sobre los lenguajes de modelado, herramientas o tecnologías utilizadas.

Primeramente estos criterios fueron aplicados tras leer el título y resumen de los artículos resultantes de la búsqueda en las librerías digitales. Luego de filtrar resultados, se procedió a aplicar los criterios analizando la introducción y conclusión de los trabajos; además se buscaron palabras clave por el artículo completo y se analizó del contexto de dichas palabras. Posteriormente se procedió a la lectura completa de los artículos resultantes, obteniendo así los resultados finales.

La figura 3 muestra un resumen de la cantidad de artículos obtenidos en cada fase del proceso, como puede observarse fueron finalmente seleccionados 13 artículos.

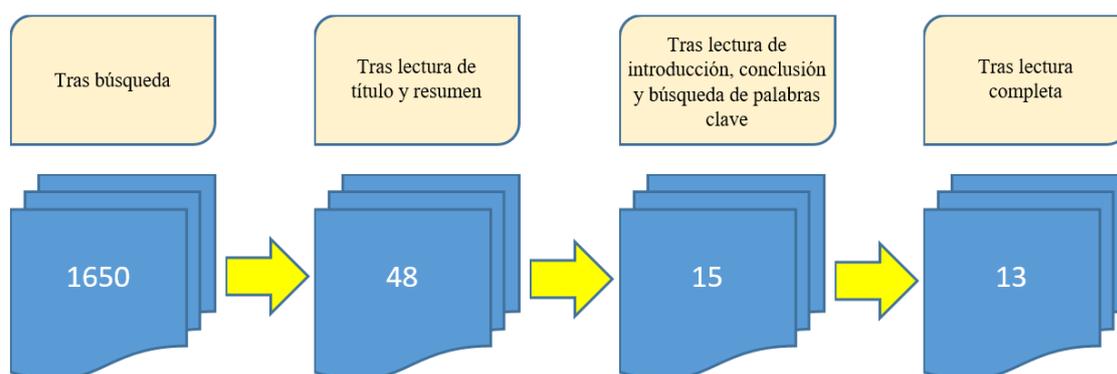


Figura 3: Artículos obtenidos en cada fase del proceso de recolección de información.

Tras un análisis de los artículos seleccionados se han respondido las preguntas de investigación de la siguiente manera:

¿Qué enfoques de desarrollo dirigido por modelos existen para el desarrollo de características de distribución de datos, distribución de procesos y comunicación asíncrona entre cliente y servidor de aplicaciones enriquecidas de internet?

Se han identificado 8 enfoques para desarrollar RIA que contemplan las características de distribución de datos, distribución de lógica de negocios y comunicación asíncrona entre cliente y servidor. A continuación describimos cada enfoque brevemente:

OOWS Extension extiende el enfoque OOWS method para contemplar el desarrollo de RIA. El enfoque propone un metamodelo para especificar la interfaz de usuario en un nivel abstracto y en otro nivel concreto para definir widgets y reacciones ante eventos. Dado esto, el enfoque está centrado en la característica de mejora de la interfaz de usuario de las RIA. Sin embargo permite el modelado de comunicaciones asíncronas mediante reglas de eventos especificadas mediante XText²⁹, permitiendo que un servicio sea ejecutado asíncronamente ante la ocurrencia de un evento [19].

El enfoque MDD propuesto por José Luis Herrero y Pablo Carmona presenta un perfil para extender UML con conceptos de aplicaciones web definiendo nuevos estereotipos y valores etiquetados, presentando además un framework de soporte y reglas de transformación para generación de código. Mediante nuevos estereotipos, se modelan solicitudes de ejecución de servicios web, distinguiéndolas en dos tipos, síncronas y asíncronas. Las asíncronas se pueden extender en los tipos callback (que incluye una operación a ejecutar) y pull (que permite almacenar respuestas del servidor) [20].

RUX-Model permite la especificación de la interfaz de usuario siguiendo un proceso de cuatro etapas: conexión con un modelo hipertextual, definición de la interfaz abstracta, definición de la interfaz concreta y especificación de la interfaz final que termina en generación de código. Al igual que OOWS Extension, RUX-Model está orientado a la especificación de la interfaz de usuario, pero permite modelar comunicación asíncrona a través de eventos y acciones asociadas mediante extensiones de UiML [21] contemplando métodos de solicitud del tipo GET y POST [18].

WebML for RIA extiende los modelos de datos e hipertextual de WebML para abarcar características RIA. Las entidades del modelo de datos son marcadas con niveles de persistencia que pueden ser “database” (permanente en el servidor) o “client” (temporal en el cliente) logrando así distribución de datos entre cliente y servidor. El modelo hipertextual expresa la estructura del frontend distinguiendo en páginas del tipo cliente o servidor. Además en este modelo se define el contenido de las páginas, los enlaces y las operaciones lanzadas por el usuario. Estas operaciones son marcadas con la localización en la que se encuentran pudiendo ser en el servidor, cliente o una combinación de ambos obteniendo así distribución de lógica de negocios entre cliente y servidor [22].

WebML for RIA + RUX-Model combina las ventajas de las dos últimas propuestas mencionadas utilizando WebML for RIA para definir los modelos de datos e hipertextual para abarcar las características de distribución de datos y lógica de negocios entre cliente y servidor y RUX-Model para definir la interfaz de usuario y soportar comunicación asíncrona entre las partes [23].

²⁹ Xtext. <https://eclipse.org/Xtext/>

La propuesta de Toffeti Carughi extiende WebML for RIA con soporte para comunicación asíncrona definido sobre aplicaciones colaborativas que utilizan tecnologías de empuje. El modelo de datos es extendido con entidades que representan eventos y tipos eventos, mientras que el modelo hipertextual agrega dos nuevos tipos de operaciones, “send event” y “receive event” para la notificación de eventos y lanzamiento de reacciones. Las reacciones son modeladas a través de un “event view” especificado a través de cadena de operaciones [24].

Brambilla et al. proponen especificar una RIA a través de procesos de negocios descritos mediante BPMN, transformar esta especificación en los modelos de datos e hipertextual de WebML y aplicar RUX-Model para obtener la interfaz de usuario. Se define un modelo BPMN como instancia de un metamodelo de procesos propuesto. Mediante este modelo se logra la distribución de datos agrupando objetos de datos en lanes (objeto BPMN para categorizar actividades de un proceso) del tipo cliente o servidor y anotaciones sobre los objetos indicando su nivel de persistencia que puede ser persistente o volátil. La lógica de negocios es representada mediante actividades y flechas de flujo de control agrupadas en lanes del tipo cliente, servidor o cliente-servidor indicando la posible distribución. Flechas de flujo de control indican la comunicación entre actividades y procesos y permiten anotaciones que especifican el tipo de transmisión, síncrona o asíncrona [8].

OOH4RIA Extension dota a OOH4RIA [25] con dos nuevos modelos, el modelo de características y el modelo de componentes. El modelo de características está formalizado por un metamodelo MOF y especifica las diferentes características que se pueden considerar en el diseño de una RIA por medio de artefactos (entidades, componentes, atributos). Estos artefactos permiten representar almacenamiento de datos del lado del cliente del tipo persistente y volátil, lógica de negocios con localización en el cliente, servidor o cliente-servidor y estilos de comunicación que pueden ser RPC o asíncrono basado en mensajes. El modelo de componentes es una instancia de un perfil UML y representa la estructura visual de la aplicación y una topología de componentes [26].

Las columnas 2, 3 y 4 de la tabla 2 presentan las características soportadas por cada enfoque mencionado. Podemos observar que cuatro de los ocho enfoques consideran todas las características de distribución de datos, distribución de lógica de negocios y comunicación asíncrona entre cliente y servidor. Además se puede apreciar que la característica de comunicación asíncrona se encuentra presente en siete enfoques, siendo la más contemplada.

¿Qué lenguajes de modelado, herramientas, tecnologías utilizan los enfoques identificados?

Los lenguajes de modelado, herramientas de desarrollo y tecnologías de implementación utilizados por los enfoques identificados pueden visualizarse en las columnas 5, 6 y 7 respectivamente en la tabla 2. La mitad de los enfoques se basan en WebML, un lenguaje de modelado concebido originalmente para el desarrollo de aplicaciones web tradicionales, junto con su herramienta de soporte, WebRatio³⁰ [17]. Además de WebML la notación visual propuesta por RUX-Model es también requerida por otros enfoques y desarrollada haciendo uso de la herramienta RUX-Tool³¹ [18]. Dada la utilización de estos lenguajes de modelado, el empleo de UML se ve limitado solo a dos enfoques de desarrollo. En cuanto a tecnologías de implementación, la mayormente utilizada es OpenLaszlo, seguida por Adobe Flex.

³⁰ WebRatio. <http://www.webratio.com/>

³¹ RUX-Tool. <http://www.homeria.com/>

Enfoques	Distribución de Datos	Distribución de Lógica de Negocios	Comunicación Asíncrona	Lenguaje de Modelado	Herramienta de Desarrollo	Tecnología de Implementación
OOWS Extension [19]	✗	✗	✓	XText	OOWS model compiler	Adobe Flex
Enfoque MDD para aplicaciones web de alta calidad [20]	✗	✗	✓	Extensión de UML	Eclipse	Javascript
RUX-Model [18]	✗	✗	✓	DSL visual	RUX-Tool	Adobe Flex, AJAX, OpenLaszlo
WebML for RIA [22]	✓	✓	✗	Extensión de WebML	WebRatio	OpenLaszlo
WebML for RIA + RUX-Model [23]	✓	✓	✓	Extensión de WebML, DSL visual de RUX-Model	WebRatio, RUX-Tool	OpenLaszlo, Adobe Flex, AJAX, XAML
WebML for RIA para aplicaciones web colaborativas [24]	✓	✓	✓	Extensión de WebML	WebRatio	OpenLaszlo
Enfoque para Diseño Conceptual de RIA basado en procesos de negocios [8]	✓	✓	✓	BPMN, WEBML, DSL visual de RUX-Model	WebRatio, RUX-Tool	No especificado
OOH4RIA Extension [26]	✓	✓	✓	Czarnecki notation, UML	OOH4RIA Tool	RichFaces

Tabla 2: Comparación de enfoques para desarrollo de RIA.

Muchos de los enfoques mencionados proponen una metodología desde cero, sin embargo, otros parten de una propuesta existente, combinándola con otra o agregando nuevas funcionalidades. Es notable que no todos los enfoques cubren todas las características. Además el hecho de que los lenguajes de modelado sean concebidos como una combinación de otros lenguajes hace que el aprendizaje de un lenguaje específico sea más complejo. También se ve que el uso de UML y MOF es muy escaso, y esto llama la atención dado que ellos forman parte del estándar de MDA y de la OMG. Además se observa que la mayoría de las herramientas de desarrollo utilizadas se encuentran muy atadas a un lenguaje de modelado, a excepción de Eclipse, y no son herramientas que pueden utilizarse de forma universal con otros lenguajes. En cuanto a tecnologías de implementación podemos ver que la mayoría utilizan OpenLaszlo y Adobe Flex dejando de lado a un conjunto de tecnologías como el caso de Dojo, jQuery, Adobe Flash y las demás mencionadas en la sección de marco teórico.

Este proyecto final de carrera busca superar las debilidades propuestas mediante un enfoque MDD para desarrollar RIA con soluciones que permitan mejorar la experiencia del desarrollador y consecuentemente la del usuario final.

5. Propuesta de Trabajo

5.1 Problemática

El problema principal que este proyecto busca atacar es la necesidad de otorgar al usuario una mejor experiencia en la web. Una mejor experiencia se traduce en mejor adaptabilidad, capacidades interactivas mejoradas, menor tiempo de respuesta e interfaz de usuario enriquecida. Esto es logrado mediante tecnologías del lado del cliente y comunicaciones asíncronas.

Las RIA ofrecen este tipo de soluciones por lo que son elegidas como objeto de desarrollo para este proyecto. Facilitar el desarrollo de este tipo de tecnologías es crucial para su fácil y rápida adopción y compenetración en la web actual. Mediante enfoques MDD podemos abstraer y agilizar el desarrollo de estas aplicaciones.

Hemos visto en la sección anterior que existen enfoques MDD para desarrollar RIA pero éstos presentan desventajas como: no cubren todas las características RIA mencionadas, no siguen el estándar MDA, definen nuevos lenguajes de modelado ocasionando el aumento de la curva de aprendizaje, están limitados a ciertas herramientas y utilizan las mismas tecnologías de implementación.

Es por ello, que partimos de un enfoque ya existente, MoWebA, que es basado en MDA y que permite la definición de un ASM para realizar extensiones orientadas a arquitecturas específicas. La extensión a realizar sería el soporte a las características RIA tratadas buscando utilizar nuevas tecnologías de implementación. A partir de esto facilitamos la implementación de RIA y logramos que estas aplicaciones alcancen a una mayor población en la web, ofreciendo sus ventajas.

5.2 Descripción General del Proyecto

El proyecto consiste en la definición de un enfoque MDD que facilite el desarrollo de RIA. Particularmente busca desarrollar una RIA que abarque principalmente las características de distribución de datos, distribución de lógica de negocios y comunicación asíncrona entre cliente y servidor. Para ello, se partirá desde el enfoque para desarrollo de aplicaciones web MoWebA, el cual será extendido con nuevas funcionalidades. Mediante metamodelos y perfiles que permitan la definición de un ASM, las nuevas funcionalidades serán modeladas y se realizará un mapeo a código funcional mediante reglas de transformación. Los metamodelos serán expresados haciendo uso del lenguaje de metamodelado MOF y los perfiles se expresarán mediante UML. Para la capa de datos se extenderá el diagrama de entidades de MoWebA. Los diagramas candidatos a extender para las capas restantes son el diagrama de contenido y el diagrama de nodos. Las reglas de transformación serán elaboradas utilizando la herramienta Acceleo³².

A partir de estas definiciones un desarrollador de software podrá crear el modelo de un sistema utilizando instancias de los perfiles, posteriormente realizará la transformación del modelo elaborado a código funcional de forma automática y luego procederá a realizar ajustes manuales de ser necesario para obtener la RIA final.

La implementación generada para la capa de datos consistirá en código SQL funcional del sistema de gestión de base de datos MySQL³³ para el lado del servidor y código javascript creado a través del API IndexedDatabase³⁴ que forma parte de la especificación de HTML5 para el lado del cliente.

³² Acceleo. <http://www.eclipse.org/acceleo/>

³³ MySQL. <https://www.mysql.com/>

³⁴ IndexedDatabase. <https://www.w3.org/TR/IndexedDB/>

Las implementaciones correspondientes a las capas de lógica de negocios y comunicación asíncrona siguen siendo evaluadas para su definición.

5.3 Estado Actual del Proyecto

Las actividades previstas para el desarrollo del proyecto se describen en la tabla 3.

N°	Actividad	Estado
1	Revisión bibliográfica y recopilación de información.	Finalizada
2	Evaluación y análisis de las metodologías MDD para el desarrollo de RIA.	Finalizada
3	Estudio de diagramas de MoWebA e identificación de aquellos a extender.	En Progreso
4	Estudio de lenguajes de metamodelado y transformación.	En Progreso
5	Desarrollo de metamodelos y reglas de transformación.	En Progreso
6	Aplicación de la propuesta desarrollada a un caso de prueba.	Pendiente
7	Elaboración del libro de tesis.	Pendiente

Tabla 3: Actividades del proyecto.

Los puntos 1 y 2 han sido completados siguiendo el proceso descrito por un mapeo sistemático de literatura. En cuanto al punto 3, se ha realizado un estudio en profundidad del enfoque MoWebA y se han identificado diagramas para extender el enfoque teniendo en cuenta la característica de distribución de datos entre cliente y servidor de las RIA. Además, se han identificado posibles diagramas a extender para contemplar demás características. Con respecto al punto 4, se ha estudiado el lenguaje de metamodelado MOF y los perfiles UML, éstos fueron utilizados para la elaboración de MoWebA y también serán utilizados por esta propuesta. También se ha estudiado el modo de uso de la herramienta Acceleo, el generador de código que será utilizado para realizar la implementación. Considerando el punto 5, ya se ha desarrollado un metamodelo y un perfil UML para la característica de distribución de datos entre cliente y servidor de las RIA. Se está trabajando en la elaboración de metamodelos y perfiles para abarcar las características restantes. Las reglas de transformación utilizadas para la generación de código aún no fueron definidas. Una vez definidos todos los metamodelos, perfiles y reglas de transformación se procederá a la aplicación de la propuesta a un caso de prueba según lo establecido en el punto 6. Para eso, se realizará una ilustración, mediante la cual se observará una instancia de la propuesta que será analizada teniendo en cuenta ciertas métricas a definir. La redacción del libro de tesis, mencionada en el punto 7, será realizada en forma simultánea a medida que se vaya avanzando en los demás puntos.

6. Referencias

- [1] J. C. Preciado, M. Linaje, F. Sanchez, and S. Comai, "Necessity of methodologies to model rich internet applications," in *Web Site Evolution, 2005.(WSE 2005). Seventh IEEE International Symposium on*. IEEE, 2005, pp. 7–13.
- [2] G. T. Carughi, S. Comai, A. Bozzon, and P. Fraternali, "Modeling distributed events in data-intensive rich internet applications," in *Web Information Systems Engineering–WISE 2007*. Springer, 2007, pp. 593–602.
- [3] G. Toffetti, S. Comai, J. C. Preciado, and M. Linaje, "State-of-the-art and trends in the systematic development of rich internet applications," *Journal of Web Engineering*, vol. 10, no. 1, pp. 070–086, 2011.
- [4] M. Brambilla, J. Cabot, and M. Wimmer, "Model-driven software engineering in practice," *Synthesis Lectures on Software Engineering*, vol. 1, no. 1, pp. 1–182, 2012.
- [5] O. M. Group, "Mda specifications." [Online]. Available: <http://www.omg.org/mda/specs.htm>
- [6] M. González, L. Cernuzzi, and O. Pastor, "A navigational role-centric model oriented web approach – moweba," in *International Journal of Web Engineering and Technology*, 2016.

- [7] I. López, M. González, and N. Aquino, “Una propuesta para el soporte de aplicaciones rias,” Proyecto Final del Carrera, Universidad Católica Nuestra Señora de la Asunción, 2015.
- [8] M. Brambilla, J. C. Preciado, M. Linaje, and F. Sanchez-Figueroa, “Business process-based conceptual design of rich internet applications,” in *Web Engineering, 2008. ICWE'08. Eighth International Conference on*. IEEE, 2008, pp. 155–161.
- [9] J. Allaire, “Macromedia flash mx—a next-generation rich client,” *Macromedia white paper*, pp. 1–2, 2002.
- [10] P. Fraternali, G. Rossi, and F. Sánchez-Figueroa, “Rich internet applications,” *Internet Computing, IEEE*, vol. 14, no. 3, pp. 9–12, 2010.
- [11] M. Busch and N. Koch, “Rich internet applications. state-of-the-art,” *Ludwig-Maximilians-Universität München, München, Germany, Rep*, vol. 902, 2009.
- [12] J. Farrell and G. S. Nezelek, “Rich internet applications the next stage of application development,” in *Information Technology Interfaces, 2007. ITI 2007. 29th International Conference on*. IEEE, 2007, pp. 413–418.
- [13] C. Pons, R. S. Giandini, and G. Pérez, “Desarrollo de software dirigido por modelos,” 2010.
- [14] T. Stahl, M. Voelter, and K. Czarnecki, *Model-driven software development: technology, engineering, management*. John Wiley & Sons, 2006.
- [15] S. Keele, “Guidelines for performing systematic literature reviews in software engineering,” in *Technical report, Ver. 2.3 EBSE Technical Report. EBSE*, 2007.
- [16] M. Genero Bocco, J. A. Cruz-Lemus, and M. G. Piattini Velthuis, *Métodos de Investigación en Ingeniería de Software*, RA-MA, Ed., 2014.
- [17] S. Ceri, P. Fraternali, and A. Bongio, “Web modeling language (webml): a modeling language for designing web sites,” *Computer Networks*, vol. 33, no. 1, pp. 137–157, 2000.
- [18] M. Linaje, J. C. Preciado, and F. Sánchez-Figueroa, “A method for model based design of rich internet application interactive user interfaces,” in *Web Engineering*. Springer, 2007, pp. 226–241.
- [19] F. Valverde and O. Pastor, *Facing the technological challenges of web 2.0: A RIA model-driven engineering approach*. Springer, 2009.
- [20] J. L. H. Agustin and P. C. Del Barco, “A model-driven approach to develop high performance web applications,” *Journal of Systems and Software*, vol. 86, no. 12, pp. 3013–3023, 2013.
- [21] M. Abrams, C. Phanouriou, A. L. Batongbacal, S. M. Williams, and J. E. Shuster, “Uiml: an appliance-independent xml user interface language,” *Computer Networks*, vol. 31, no. 11, pp. 1695–1708, 1999.
- [22] A. Bozzon, S. Comai, P. Fraternali, and G. T. Carughi, “Conceptual modeling and code generation for rich internet applications,” in *Proceedings of the 6th international conference on Web engineering*. ACM, 2006, pp. 353–360.
- [23] J. C. Preciado, M. Linaje, S. Comai, and F. Sanchez-Figueroa, “Designing rich internet applications with web engineering methodologies,” in *Web Site Evolution, 2007. WSE 2007. 9th IEEE International Workshop on*. IEEE, 2007, pp. 23–30.
- [24] G. T. Carughi, “Modeling data-intensive rich internet applications with server push support.” in *MDWE*, 2007.
- [25] S. Meliá, J. Gomez, S. Perez, and O. Diaz, “A model-driven development for gwt-based rich internet applications with ooh4ria,” in *Web Engineering, 2008. ICWE'08. Eighth International Conference on*. IEEE, 2008, pp. 13–23.
- [26] S. Melia, J. Gomez, S. Perez, and O. Diaz, “Facing architectural and technological variability of rich internet applications,” to be published, early Access. [Online]. Available: <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5432136>