

Mobile Cloud Applications Development through the Model Driven Approach: A Systematic Mapping Study

Emanuel Sanchiz*, Magalí González†, Nathalie Aquino‡ & Luca Cernuzzi§

Departamento de Electrónica e Informática - DEI

Universidad Católica “Nuestra Señora de la Asunción”

Asunción, Paraguay

Email: {*emanuel.sanchiz, †mgonzalez, ‡nathalie.aquino, §lcernuzzi}@uc.edu.py

Abstract—Currently, a growing interest is being caused by mobile cloud applications. Improvements related to the portability of these applications among different platforms and different service providers are a critical need. Model Driven Development (MDD) constitutes one of the alternatives to address the portability problem. This work presents a systematic mapping study that analyzes different proposals that apply MDD to the development of mobile cloud applications and that, at the same time, consider the improvement of the portability of these applications. Even though we have identified just a few studies related to our subject of interest, the validation experiences that are presented in them, encourage the adoption of MDD to address the portability problem. However, further validation experiences that consider more complex cases in industrial environments will be required to justify the benefits of MDD in a substantial manner.

Keywords—systematic mapping study, mobile cloud applications, model driven development, portability.

1. Introducción

En estos últimos años, el número de usuarios de teléfonos inteligentes y tabletas ha ido creciendo rápidamente, tanto que ha superado al de usuarios de computadoras personales, ya sean de escritorio o portátiles¹. En consecuencia, el interés por el desarrollo de aplicaciones para dispositivos móviles ha aumentado. También cabe destacar la trascendencia de la nube como un potenciador o extensor de la limitada capacidad de los citados dispositivos [1], [2]. En este trabajo, nuestro interés se centra en las aplicaciones móviles con funcionalidades en la nube. Cuando hablamos de funcionalidades en la nube nos referimos a algún módulo o implementación, por ejemplo, algún procedimiento, servicio o base de datos, cuyo alojamiento o ejecución se da en la nube.

En este sentido, dichas aplicaciones contemplan dos ambientes distintos, el del móvil y el de la nube. Los ambientes

están constituidos por sistemas operativos, lenguajes de programación, librerías, servicios, entornos de trabajo, etc. En el ambiente móvil, un problema crítico está relacionado al mejoramiento de la portabilidad entre plataformas [3], [4], [5]. Esto constituye un problema crítico porque cada plataforma, a su vez, tiene su propio ambiente, por lo cual, desarrollar aplicaciones para cada una de ellas implica un mayor costo, tiempo y esfuerzo. En otras palabras, para una aplicación determinada, es necesario construir una versión, prácticamente una nueva aplicación, por cada plataforma distinta. Así mismo, en la nube, también encontramos que la portabilidad es un problema crítico, conocido como *vendor lock-in*, donde los proveedores de servicios, además de contemplar sus propios ambientes, tienen sus propios servicios y políticas [6], [7]. Por lo tanto, al considerar aplicaciones que contemplan, a la vez, el ambiente móvil y el de la nube, el problema de la portabilidad se acentúa aún más.

El desarrollo dirigido por modelos (MDD - *Model Driven Development*) se perfila como una solución adecuada para mejorar la portabilidad de las aplicaciones. De hecho, una de las motivaciones principales del enfoque dirigido por modelos es mejorar la portabilidad mediante la proposición de modelos independientes de plataforma (PIM - *Platform Independent Model*) y reglas de transformación que permiten reutilizar el diseño de la lógica de la solución para distintas plataformas destino [8], [9]. Dicha reutilización busca contribuir a un ahorro de esfuerzo, costo y tiempo en el desarrollo de aplicaciones que deben ser utilizadas en distintas plataformas.

Tomando como base un trabajo previo [10], hemos identificado ciertos aspectos que, creemos, podrían tener un impacto positivo sobre el problema de la portabilidad en el ámbito del desarrollo de aplicaciones móviles usando MDD. Esos aspectos son los siguientes:

- Navegación orientada a funciones, es decir, una navegación basada más en las funcionalidades que necesita el usuario que en la estructura de la información representada, teniendo en cuenta que varios métodos y metodologías MDD arrancan del modelo conceptual de datos para diseñar la navegación [11], [12], [13], [14]. El diseño del acceso a los datos a través de la interfaz de usuario debe ser independiente de la disposición o estructura de almacenamiento de los mismos. Es

Este trabajo ha sido desarrollado con el soporte de CONACYT, en el contexto del proyecto “Mejorando el proceso de desarrollo de software: propuesta basada en MDD” (14-INV-056).

1. Morgan Stanley, enlace: <https://goo.gl/Ifzmvz>

decir, ante un cambio de la estructura de datos el diseño de la navegación no se debería ver afectado. En adición, al existir dicha independencia, se tiene más libertad y flexibilidad para adaptar la navegación a las necesidades y requerimientos del usuario, lo cual es fundamental para el éxito de las aplicaciones móviles [15], [16].

- Separación del modelado de la capa de presentación con respecto al modelado de la navegación y el comportamiento. La capa que presenta mayores dificultades para la portabilidad es la de presentación [4]. Por lo tanto, una clara separación de sus detalles con respecto a los de las demás capas, puede contribuir al mejoramiento de la portabilidad.
- Incorporación de un nuevo nivel de abstracción en el modelado, un modelo específico de arquitectura (ASM - *Architecture Specific Model*). Ante el advenimiento de nuevas tecnologías, conceptos, patrones, funciones, los cuales establecen nuevas arquitecturas de aplicaciones, como en este caso, la arquitectura de las aplicaciones móviles con funcionalidades en la nube, podría ser conveniente proponer un método de adaptación y evolución que responda a dichos cambios. Consideramos al ASM con el propósito de reutilizar el modelado del PIM (diseño de procesos, estructuras de datos, navegación, etc.) e incorporar a dicho modelado los aspectos y elementos que contemplan las nuevas arquitecturas, a través de reglas de transformación de modelo a modelo. Esto supondría un mejoramiento en la portabilidad del PIM y también podría significar, consecuentemente, un ahorro de esfuerzo, tiempo y costos en el diseño.

En este trabajo presentamos un mapeo sistemático de la literatura sobre el desarrollo, basado en MDD, de aplicaciones móviles con funcionalidades en la nube. Además, el estudio se ha enfocado a identificar, reconocer y analizar las propuestas que hacen frente al desafío de la portabilidad. Por lo tanto, cada propuesta identificada en el estudio ha sido analizada en relación a los aspectos que presentamos previamente y que creemos que pueden tener un impacto positivo sobre el problema de la portabilidad en el contexto del desarrollo de aplicaciones móviles con MDD. Además, se han analizado también los principales aportes, limitaciones y experiencias de validación de las propuestas.

En este trabajo, hemos optado por realizar un mapeo sistemático de la literatura, que se caracteriza por ser más general que una revisión sistemática de literatura, ya que su principal objetivo es brindar una visión global sobre un tema de interés, con evidencias empíricas o no [17], y esto está en correspondencia con el objetivo de este trabajo.

Es de esperar que este estudio abra una perspectiva que, a futuro, otros investigadores puedan enriquecer, a la vez que también puedan enriquecerse con los análisis y resultados de este trabajo.

Lo que queda del documento se organiza de la siguiente manera: en la sección 2 se describen los trabajos relacionados. Luego en las secciones 3, 4 y 5 se presentan respectivamente la planificación, la ejecución y los resultados del mapeo sistemático de la literatura. En la sección 6 se

analizan los resultados, en la sección 7 se especifican las limitaciones del estudio para en la sección 8 presentar la conclusión del mismo.

2. Trabajos Relacionados

Como punto de partida de este trabajo, primeramente hemos llevado a cabo una búsqueda de mapeos y revisiones sistemáticas de literatura que contemplen el desarrollo de aplicaciones móviles con funcionalidades en la nube bajo el enfoque dirigido por modelos. A pesar de no haber identificado la existencia de tales trabajos, hemos encontrado estudios que de alguna manera se relacionan con el nuestro y que se consideran a continuación.

Heitkötter et al. [18] presentan un análisis centrado en el desarrollo de aplicaciones móviles portables donde se comparan varias soluciones multi-plataforma. Dichas soluciones no son propuestas dirigidas por modelos, mas brindan perspectivas alternativas de desarrollo que se enfocan en aliviar el problema de la portabilidad entre plataformas del ambiente móvil.

Ribeiro y da Silva [19] presentan una serie de tecnologías enfocadas al desarrollo de aplicaciones móviles multi-plataforma, entre las cuales, dos soluciones contemplan el enfoque dirigido por modelos. Dicho estudio brinda un análisis comparativo, en el cual se pueden apreciar características, ventajas y desventajas de estas dos soluciones contra las demás consideradas. Esto refuerza nuestra hipótesis de la utilidad del enfoque dirigido por modelos.

Di Martino et al. [20] describen un conjunto de proyectos que aplican el enfoque dirigido por modelos para dar soporte a la portabilidad e interoperabilidad en la nube. Este trabajo brinda un panorama de distintas propuestas dirigidas por modelos, aunque se centran sólo en el ambiente de la nube. Además, se abarca un espectro más amplio que el desarrollo de software, incluyendo técnicas de ingeniería reversa sobre sistemas legados.

En los trabajos citados, no hemos identificado un seguimiento de los lineamientos de un estudio sistemático de literatura. Finalmente, al no haber identificado ni revisiones ni mapeos sistemáticos de literatura que aborden el desarrollo de aplicaciones móviles con funcionalidades en la nube bajo el enfoque dirigido por modelos, nos vimos motivados a realizar el mapeo sistemático que se describe en las siguientes secciones.

3. Planificación del Mapeo Sistemático de la Literatura

La planificación del mapeo sistemático de literatura, la llevamos a cabo teniendo en cuenta las directivas propuestas en Genero et al. [17] y en Kitchenham y Charters [21].

La planificación del mapeo contempla la definición de las preguntas de investigación, la especificación del protocolo de búsqueda, la derivación de la cadena de búsqueda y de los criterios de inclusión y exclusión, y la especificación de las fuentes de consulta.

3.1. Preguntas de Investigación

Como ya se ha mencionado previamente, el objetivo de este estudio es generar una compilación y un análisis de artículos relacionados a trabajos enfocados en el mejoramiento de la portabilidad de las aplicaciones móviles con funcionalidades en la nube, cuyos procesos de desarrollo estén basados en MDD. Es de interés analizar cada uno de estos trabajos con respecto a si proveen, o no, un modelo de la navegación orientado a funciones, una separación de la capa de presentación con respecto a las demás capas de modelado, y una incorporación de un ASM, ya que consideramos que estos aspectos pueden resultar beneficiosos para encontrar soluciones al problema de la portabilidad. Además, también es de interés identificar los aportes y limitaciones de estos trabajos, así como los resultados de las evaluaciones que presentan en relación a la problemática de la portabilidad.

Teniendo en cuenta estos objetivos e intereses, se han definido las siguientes preguntas de investigación:

- **PI1:** ¿Cuáles son las propuestas dirigidas por modelos para el desarrollo de aplicaciones móviles con funcionalidades en la nube, enfocadas en el problema de la portabilidad?
En las propuestas identificadas:
- **PI2:** ¿Se identifican evidencias que muestren la adopción del modelado navegacional orientado a funciones?
- **PI3:** ¿Se considera una clara separación del modelado de la capa de presentación con respecto al modelado de la navegación y del comportamiento?
- **PI4:** ¿Se identifica el establecimiento de una distinción entre los niveles de abstracción de modelado PIM y ASM?
- **PI5:** ¿Qué otros aportes y limitaciones se presentan en relación al problema de la portabilidad entre plataformas?
- **PI6:** ¿Se presentan evaluaciones? En caso afirmativo, ¿son positivos los resultados con respecto a la problemática de la portabilidad?

3.2. Protocolo de Búsqueda

El protocolo de búsqueda que se ha seguido para realizar este mapeo consiste de los siguientes pasos:

- 1) Derivar términos de las preguntas de investigación para elaborar una cadena de búsqueda inicial.
 - a) Utilizar términos similares, relacionados y siglas para obtener la cadena final de búsqueda.
- 2) Consultar las fuentes de información con la cadena de búsqueda definida. Centrar la búsqueda sobre títulos, resúmenes y palabras claves. Adaptar la cadena de búsqueda según el formato y restricciones de cada herramienta de búsqueda. Las fuentes a consultar incluyen bibliotecas digitales y servicios de indexación, comprendiendo revistas y conferencias.
- 3) Aplicar un procedimiento de selección de trabajos a partir de los resultados de la consulta de las fuentes

de información. Dicho procedimiento consiste en la aplicación de unos criterios de selección en tres etapas:

- a) Primera etapa: aplicar los criterios de selección teniendo en cuenta el título de los trabajos identificados en la consulta de las fuentes de información (paso 2).
- b) Segunda etapa: aplicar los criterios de selección teniendo en cuenta los resúmenes y palabras clave de los trabajos seleccionados en el paso 3(a).
- c) Tercera etapa: aplicar los criterios de selección teniendo en cuenta el texto completo de los trabajos seleccionados en el paso 3(b). Descartar los artículos repetidos y los artículos que correspondan a distintas publicaciones de un mismo trabajo o propuesta.

Dentro del procedimiento de selección de estudios, en cuanto al rango de años de publicación, no consideramos ninguna restricción.

3.3. Cadena de Búsqueda

La primera pregunta de investigación (PI1) ha sido utilizada para derivar, a partir de ella, términos, y formar la cadena inicial de búsqueda. Esto fue así porque a partir de esa pregunta se pueden identificar las propuestas relevantes para este estudio. Luego, dicha cadena fue enriquecida con siglas, términos similares y relacionados. La cadena de búsqueda resultante es la siguiente:

((“model driven” OR “model-driven” OR MDD OR MDE OR MDA OR “model based” OR “model-based”) AND (mobile OR smartphone OR tablet OR Android OR “windows phone” OR iOS) AND (cloud OR “remote server” OR backend))

La cadena de búsqueda fue adaptada según los requerimientos y restricciones de cada herramienta de búsqueda, dichas adaptaciones se pueden apreciar en el Cuadro 1.

3.4. Criterios de Selección

Las demás preguntas de investigación (PI2, PI3, PI4, PI5, PI6) han sido utilizadas para guiar la definición de los criterios de inclusión y exclusión de trabajos.

3.4.1. Criterios de inclusión. Incluimos estudios y propuestas que: i) contemplan alguna de las siguientes propiedades: navegación orientada a funciones, separación del diseño de la capa de presentación de la de navegación y/o comportamiento, consideración de un nivel de abstracción adicional de modelado, como el ASM; ii) presentan alguno de los siguientes elementos: meta-modelos, perfiles, reglas de transformación, marcos de trabajo, herramientas.

3.4.2. Criterios de exclusión. Excluimos estudios y propuestas que: i) no están relacionados a la ingeniería de software; ii) no siguen un enfoque dirigido por modelos; iii) no tienen relación con el desarrollo de aplicaciones móviles; iv) no contemplan el modelado y la generación de funcionalidades en la nube; v) no consideran el problema de la portabilidad a nivel de plataforma.

Cuadro 1. CADENA DE BÚSQUEDA POR FUENTE CONSULTADA

Biblioteca digital	Cadena de búsqueda
IEEEExplore	((("model driven" OR "model-driven" OR MDD OR MDE OR MDA OR "model based" OR "model-based") AND (mobile OR smartphone OR tablet OR Android OR "windows phone" OR iOS) AND (cloud OR "remote server" OR backend))
ScienceDirect	((("model driven" OR "model-driven" OR MDD OR MDE OR MDA OR "model based" OR "model-based") AND (mobile OR smartphone OR tablet OR Android OR "windows phone" OR iOS) AND (cloud OR "remote server" OR backend))) AND LIMIT-TO(topics, "model,cloud")
ACM	((("model driven" OR "model-driven" OR MDD OR MDE OR MDA OR "model based" OR "model-based") AND (mobile OR smartphone OR tablet OR Android OR "windows phone" OR iOS) AND (cloud OR "remote server" OR backend))
DBLP	(model.driven MDD MDE MDA)(mobile smartphone ios android)
Google Scholar	allintitle: mobile cloud "model driven" OR MDD OR MDE OR MDA OR "model based" OR smartphone OR tablet OR remote OR server OR backend OR Android OR "windows phone" OR iOS

3.5. Fuentes de Consulta

En principio, elegimos las fuentes de consultas según el listado propuesto por Kitchenham y Charters [21]. Las mismas consisten en bibliotecas digitales y servicios de indexación. Algunas bibliotecas que se citan en dicho listado, como Citeseer library e Inspec, no arrojaban resultados coincidentes con nuestra área de estudio, por lo cual nos centramos en las bibliotecas citadas más adelante. Sin embargo, los servicios de indexación han identificado estudios pertenecientes a distintas bibliotecas digitales, incluyendo a varias que no consideramos directamente. A continuación se citan las fuentes de consultas consideradas:

- Bibliotecas digitales: IEEEEXplore, ACM y Science Direct.
- Servicios de indexación: Google Scholar y DBLP.

4. Ejecución del Mapeo Sistemático de la Literatura

El Cuadro 2 presenta las cantidades de artículos que han resultado de la ejecución del proceso de búsqueda. En base al protocolo que se ha presentado previamente, la columna *Búsqueda* del Cuadro 2 presenta la cantidad de artículos que se ha encontrado en cada fuente digital en base a la cadena de búsqueda definida para este estudio (paso 2 del protocolo). De manera similar, las columnas *Criterios en título*, *Criterios en resumen y palabras clave*, y *Criterios en texto completo* muestran las cantidades de artículos que han

resultado después de la ejecución de los pasos en los que se aplican los criterios de inclusión y exclusión, pasos 3(a), 3(b), y 3(c) del protocolo, respectivamente. Como se puede ver, finalmente, cinco artículos han quedado seleccionados para ser analizados en este mapeo. En el Cuadro 3 se listan los cinco artículos seleccionados.

Cabe comentar que en la etapa de aplicar los criterios de selección sobre los títulos, incluimos todos aquellos estudios en cuyos títulos se hace referencia, al menos, al enfoque dirigido por modelos y al ambiente móvil, ya que obtuvimos resultados de otras áreas, los cuales no se referían al desarrollo de aplicaciones móviles. En esta etapa, consideramos todos los criterios de selección, excepto los criterios de exclusión iv) y v), que se refieren a la contemplación de funcionalidades en la nube y al problema de portabilidad, respectivamente, por ser cuestiones más particulares, ya que del título no obtenemos suficiente información acerca del contenido del artículo, aunque sí asumimos que al menos deberían haber menciones acerca del uso del enfoque MDD y del ambiente móvil.

En la etapa de aplicar los criterios de selección sobre resúmenes y palabras claves, nos enfocamos en seleccionar los artículos que contemplan la portabilidad como problemática principal, y en excluir los que no. Aquí consideramos todos los criterios de selección, excepto el criterio de exclusión iv), el cual se refiere a la contemplación de funcionalidades en la nube. Esto fue así porque buscamos trabajos que se enfocan en el problema de la portabilidad, entonces asumimos que en los resúmenes y en las palabras claves debería haber mención de dicha problemática.

Finalmente, en la etapa de aplicar el análisis sobre el texto completo de los artículos, todos los criterios de selección fueron considerados.

5. Resultados

El Cuadro 3 presenta un resumen de los resultados de este mapeo, para cada pregunta de investigación.

A continuación, los resultados asociados a cada pregunta de investigación se describen brevemente.

- **PI1:** ¿Cuáles son las propuestas dirigidas por modelos para el desarrollo de aplicaciones móviles con funcio-

Cuadro 2. BÚSQUEDA EN FUENTES DIGITALES

Fuentes digitales		Búsqueda	Criterios en título	Criterios en resumen y palabras clave	Criterios en texto completo
Bibliotecas digitales	IEEE	65	1	1	5
	ACM	18	0	0	
	Science Direct	137	2	2	
Servicios de index.	Google Scholar	146	3	2	
	DBLP	51	37	17	
Total de resultados		417	43	22	5

Cuadro 3. RESUMEN DE LOS RESULTADOS POR PREGUNTAS DE INVESTIGACIÓN

Detalles	Trabajos				
	WebRatio	MD ²	MobiCloud	Steiner et al.	Ruokonen et al.
Artículos	[4]	[22]	[23]	[24]	[25]
Breve Descripción	Framework de desarrollo con lenguaje de modelado gráfico basado en IFML-mobile, IFML, UML and BPMN	Lenguaje de modelado textual basado en el esquema MVC	Lenguaje de modelado textual basado en el esquema MVC	Lenguaje de modelado textual basado en el esquema MVC	Modelado gráfico para la generación de interfaces de usuario basados en servicios web
PI1	Sí	Sí	Sí	Sí	Sí
PI2	Sí	Sí	No	Sí, pero con cierta limitación	No
PI3	No	Sí	No, no queda clara la separación	No, no queda clara la separación	No, no queda clara la separación
PI4	No	No	No	No	No
PI5	Sí, contempla MDD + enfoque de desarrollo de código abierto, REST como interfaz de comunicación	Sí, contempla portabilidad del servidor basada en el JAVA Server, REST como interfaz de comunicación	Sí, contempla REST como interfaz de comunicación	No	No
PI6	Sí	Sí	Sí	Sí, pero faltan detalles de la evaluación	No

nalidades en la nube, enfocadas en el problema de la portabilidad?

Como se ha comentado previamente, en este mapeo identificamos finalmente cinco estudios verificando el cumplimiento de los criterios de selección (ver Cuadro 2 y Cuadro 3). Estos estudios constituyen la respuesta a la primera pregunta de investigación. Es decir, las propuestas dirigidas por modelos para el desarrollo de aplicaciones móviles con funcionalidades en la nube y enfocadas en el problema de la portabilidad son WebRatio [4], MD² [22], MobiCloud [23], la propuesta de Steiner et al. [24], y la propuesta de Ruokonen et al. [25]. Sobre ellas, llevamos a cabo un análisis para responder a las preguntas restantes.

Entre los artículos recopilados, el que está asociado al trabajo más completo y con más peso en la industria es el de Brambilla et al. Dicho trabajo se corresponde con la plataforma de desarrollo WebRatio² y utiliza modelado gráfico. MD², basado en el esquema Modelo-Vista-Controlador (MVC), se puede considerar como el segundo trabajo en cuanto a alcance de modelado y generación, el mismo ha llevado a cabo experimentos con socios de la industria. De los restantes trabajos, dos también basados en MVC, los mismos consideran casos más básicos y simplificados para el modelado y la generación de las aplicaciones.

Si bien hemos identificado más estudios que se refieren a la aplicación de MDD al ambiente móvil, sólo hallamos estos cinco estudios que contemplan el diseño y la generación de funcionalidades en la nube.

- **PI2:** ¿Se identifican evidencias que muestren la adopción del modelado navegacional orientado a funciones? En WebRatio, en MD² y en la propuesta de Steiner et al., identificamos la adopción de un modelado de la

navegación independiente al modelado de datos (modelado estático), por lo cual es posible obtener un diseño de la navegación orientado a funciones, en donde se puede asegurar la independencia de la estructura de datos. En WebRatio, a través del lenguaje IFML-Mobile se permite definir las transiciones entre los distintos elementos de la interfaz de usuario y, por supuesto, las transiciones entre las distintas pantallas que conforman la estructura de la aplicación. Así también para MD², el modelado de la navegación se da en el módulo del controlador (esquema MVC) y se permite modelar distintos escenarios de transiciones a través del concepto de *workflows*. En la propuesta de Steiner et al. se contempla un modelado de la navegación pero limitado al subconjunto de patrones disponibles tanto en la plataforma iOS como en la plataforma Android. Se podría obtener una navegación orientada a funciones pero con limitaciones. En MobiCloud y en la propuesta de Ruokonen et al. no se contempla un modelado de la navegación, por lo cual no se garantiza una navegación orientada a funciones.

- **PI3:** ¿Se considera una clara separación del modelado de la capa de presentación con respecto al modelado de la navegación y del comportamiento?

Dicha separación la identificamos en MD², donde la navegación se modela en el módulo del controlador, aislando el modelado de los detalles de la capa de presentación (elementos gráficos, estilos, y demás) en el módulo de vista (esquema MVC). En MobiCloud y en la propuesta de Ruokonen et al. no se menciona el modelado de la navegación, por lo cual no queda clara la separación de dichos aspectos y podríamos pensar, en principio, que no hay consideración de dicha separación. Así también, en la propuesta de Steiner et al. no se especifica claramente el modo de diseño

2. WebRatio, enlace: <http://goo.gl/WsJ2Df>

de la capa de presentación, ni la de comportamiento y navegación, por lo cual, tampoco queda clara la separación de tales capas. Sin embargo, en WebRatio se distingue una mezcla de detalles de la capa de presentación y el modelado de la navegación, ya que se definen elementos gráficos y al mismo tiempo se modela la navegación entre los elementos y las distintas pantallas. Dicha mezcla podría estar disminuyendo la portabilidad del diseño de ambas capas.

- **PI4:** ¿Se identifica el establecimiento de una distinción entre los niveles de abstracción de modelado PIM y ASM?

Sólo WebRatio define una manera de extender su PIM (a partir del lenguaje IFML³) para obtener modelos orientados al ambiente móvil, y para el lado de la nube incorpora otros esquemas de modelado, como UML y BPMN. Sin embargo, no hemos identificado que en WebRatio se considere o reconozca la diferenciación conceptual entre el PIM y el ASM. En este sentido, la aplicación de dichas extensiones al PIM lo modifican directamente y, posiblemente, disminuyen su reusabilidad y portabilidad, ya que pueden incorporar información de ambientes, arquitecturas o plataformas específicas. En los demás trabajos sólo se habla de la generación directa de las implementaciones específicas de plataforma a partir de los modelos del PIM.

- **PI5:** ¿Qué otros aportes y limitaciones se presentan en relación al problema de la portabilidad entre plataformas?

Como sólo contemplamos trabajos que se basan en MDD para el desarrollo de aplicaciones móviles con funcionalidades en la nube, los cinco trabajos seleccionados abstraen al desarrollador de los detalles específicos de implementación de los ambientes móvil y de la nube, mediante el modelado independiente de plataforma, y permiten la generación de implementaciones para distintas plataformas a partir del mismo modelado. Esto último significa un ahorro de esfuerzo, costo y tiempo, considerando el problema de la dificultad de portabilidad entre plataformas.

Además de los otros aspectos que hemos verificado a través de las tres preguntas anteriores, llevamos a cabo un análisis para identificar otras estrategias que complementen la aplicación de MDD para el mejoramiento de la portabilidad de las aplicaciones.

WebRatio contempla la generación de código móvil basado en el Apache Cordova⁴, de enfoque híbrido o *native wrapper*, además de la generación de código Java estándar para el lado de la nube, basando su portabilidad en el Java Server. En general, podríamos afirmar que WebRatio apuesta por un complemento entre MDD y la generación de implementaciones de código abierto, el cual se considera como un enfoque que brinda mayor flexibilidad para abordar la cues-

tion del *vendor lock-in*⁵. Dicho complemento podría apreciarse como una combinación adecuada para hacer frente a las dificultades propias de la problemática de portabilidad [4], [22]. Así mismo, en WebRatio se provee la generación automática de un método de sincronización inteligente, que se podría considerar clave en el ahorro de los recursos de los dispositivos móviles en los procesos de comunicación entre ambos ambientes. Dicha implementación puede ser generada igualmente para las distintas plataformas. Sin embargo, para el diseño de la aplicación del lado móvil y del lado de la nube, en WebRatio se utilizan distintos lenguajes de modelado, con entornos de trabajo distintos. Esto podría significar, en algún sentido, una limitación en cuanto a la abstracción de las diferencias de estos ambientes, y a la necesidad de aprender a trabajar con dos entornos distintos [4].

En MD², MobiCloud y Steiner et al. se aprecian lenguajes de dominio específico, textuales, basados en el patrón de diseño MVC, donde a partir del mismo lenguaje se generan implementaciones de plataforma específica, tanto para el ambiente móvil como de la nube. Se trabaja con un sólo lenguaje y entorno de modelado, lo cual podría constituirse en una ventaja en cuanto a la curva de aprendizaje si comparamos con WebRatio. Estos tres trabajos consideran la generación nativa de código para el lado móvil. Generalmente, este tipo de código es el preferido debido a que presentan mejores posibilidades para un aprovechamiento más adecuado de los recursos de los dispositivos móviles, lo cual a su vez significa la obtención de aplicaciones con mejor desempeño y presentación. Así como en WebRatio, en MD² se contempla la generación de aplicaciones Java, basando la portabilidad de la implementación en el Java Server.

En MobiCloud, se generan implementaciones específicas para dos proveedores propietarios de servicios en la nube, Amazon (EC2) y Google App Engine (GAE). En cuanto a desventajas, el planteamiento del modelado y de la generación en MobiCloud, se limita a operaciones CRUD y a la generación de la comunicación entre el ambiente móvil y de la nube. Así también, en la propuesta de Steiner et al., se contempla la generación para GAE, pero no se considera la generación para otro proveedor de la nube.

De entre todas las propuestas, identificamos a REST como la arquitectura de comunicación más adoptada. La uniformidad de la interfaz de comunicación se constituye en una propiedad muy positiva para el mejoramiento de la portabilidad en la comunicación entre el ambiente móvil y de la nube, e igualmente a través de las distintas plataformas que contemplan estos ambientes [26], [27].

En cuanto al trabajo de Ruokonen et al., se considera, a partir de los modelos, la generación de interfaces de usuario para dispositivos de distintas características

3. IFML, enlace: <http://goo.gl/mWdPpo>

4. Apache Cordova, enlace: <https://goo.gl/QocB2x>

5. Más información en: <https://goo.gl/mOCr8P>

físicas (modelo, tamaño de la pantalla). Se da la identificación de patrones comunes y repetitivos dentro del modelo de procesos de negocio. En cuanto a las desventajas, se da una generación de código no nativo para las interfaces. Además, según el contexto del trabajo (Nokia Research Center, año 2008), se podría decir que está enfocado a un tipo de ambiente específico, y que no se consideran los distintos sistemas operativos que hoy tenemos.

- **PI6:** ¿Se presentan evaluaciones? En caso afirmativo, ¿son positivos los resultados con respecto a la problemática de la portabilidad?

Cuatro de los cinco trabajos analizados incluyen validaciones realizadas. Todas estas evaluaciones presentaron, en principio, resultados alentadores. El método de evaluación más utilizado, fue la comparación entre la cantidad de líneas de código de los modelos y la cantidad de líneas de código generadas a partir de dichos modelos. Este método es el utilizado por los estudios basados en lenguajes textuales de modelado. A partir de ello, dichos estudios relacionan la cantidad de líneas generada con el esfuerzo, el tiempo y el costo que tomaría la implementación manual de dichas líneas, con el seguimiento del enfoque tradicional de desarrollo. MobiCloud, MD² y Steiner et al. utilizaron dicho método y presentaron diferencias entre dichas cantidades en al menos un orden de magnitud, siendo más favorables (menores) las cantidades de líneas de código generadas a partir de los modelos. A partir de esto, se puede deducir que el esfuerzo, el tiempo y, consecuentemente, el costo de obtener una aplicación móvil con funcionalidades en la nube, para más de una plataforma, utilizando un enfoque basado en modelos, son menores en comparación a cuando se utilizan métodos tradicionales de desarrollo.

En el caso de WebRatio, se llevó a cabo una comparación entre el desarrollo dirigido por modelos y el desarrollo tradicional, considerando especialmente el ambiente del lado móvil. Tres equipos, con dos integrantes cada uno, desarrollaron una aplicación según se describe a continuación:

- Equipo 1: se encargó del desarrollo del lado del servidor, implementando servicios REST.
- Equipo 2: se encargó del desarrollo del front-end, adoptando un enfoque de desarrollo manual, tradicional.
- Equipo 3: se encargó también del desarrollo del front-end, pero adoptando el enfoque de desarrollo dirigido por modelos propuesto por WebRatio.

Los equipos 2 y 3 desarrollaron el front-end siguiendo el enfoque multi-plataforma Apache Cordova.

En cuanto a los resultados, se verificó lo siguiente:

- Del lado del servidor: 9 días-hombre fueron empleados para el desarrollo del lado del servidor, utilizando el enfoque dirigido por modelos a través de WebRatio Platform.
- Del lado del cliente:

- * Desarrollo manual en 21 días-hombre: i) implementación del estilo gráfico: 2 días-hombre; ii) desarrollo de la estructura de la aplicación y la interacción de usuario: 12 días-hombre; iii) implementación de la interacción cliente-servidor y de las notificaciones push: 7 días-hombre.
- * Desarrollo dirigido por modelos en 11 días-hombre: i) implementación del estilo gráfico: 1 día-hombre; ii) desarrollo de la estructura de la aplicación y la interacción de usuario: 7 días-hombre; iii) implementación de la interacción cliente-servidor y de las notificaciones push: 3 días-hombre.

El periodo de testeo, para todos los equipos, abarcó 2 semanas.

A partir de los datos descritos se determinaron los siguientes beneficios: ahorro de esfuerzo del 48 %; reducción de costo del 21 %; reducción de tiempo de 21 a 11 días-hombre. En todos los casos, los números favorecen al enfoque MDD propuesto en Brambilla et al. [4] (WebRatio).

Finalmente, en el trabajo de Ruokonen et al., no se presenta una evaluación de implementación, sino más bien, se plantean ciertas cuestiones relacionadas a patrones de diseño que tienen que ver con el ambiente móvil, y que pueden ser automatizadas mediante MDD. A partir de esto, y considerando a la problemática de la portabilidad, podríamos decir que en principio, los resultados de las evaluaciones son positivos, y por lo tanto alentadores.

6. Análisis de Resultados

Teniendo en cuenta el problema de la portabilidad en las aplicaciones móviles con funcionalidades en la nube, a partir de los trabajos analizados, hemos hallado evidencias iniciales positivas que soportan la aplicación de MDD para el desarrollo de dichas aplicaciones. A pesar de este aspecto positivo, la escasa cantidad de trabajos encontrados no permite emitir juicios contundentes que puedan, de algún modo, generalizar o afianzar los beneficios de la aplicación de MDD sobre el problema de la portabilidad, al menos para el tipo de aplicaciones que nos interesan.

Podríamos considerar a WebRatio, a través de Brambilla et al. [4] como el estudio más maduro en nuestra área de interés. WebRatio contempla un modelado acorde a las implicancias del diseño de una aplicación de caso de uso real. Además, presenta las evidencias más significativas en relación a la validación del trabajo, las cuales alientan a un mayor estudio sobre la aplicación de MDD. Aunque, por otra parte, una posible limitación de WebRatio podría ser la no consideración de un entorno unificado para el desarrollo de las aplicaciones móviles con funcionalidades en la nube, teniendo en cuenta que considera el desarrollo de este tipo de aplicaciones en distintos entornos de trabajo. Luego de WebRatio, siguiendo una línea descendente en cuanto a madurez, se encuentra MD², este estudio presenta un esquema de modelado unificado, con un entorno de

trabajo unificado, donde se tienen en cuenta aspectos de modelado que consideramos importantes para el diseño de las aplicaciones móviles, como la posibilidad del modelado de la navegación orientada a funciones, y una clara separación entre el modelado de la capa de presentación y el modelado de las demás capas. Sería interesante la continuación de dicho estudio aplicado a casos más complejos en ambientes industriales. En este sentido, MD² y, especialmente, los demás artículos analizados, evidencian una etapa temprana de investigación en la aplicación de MDD para el desarrollo de aplicaciones móviles con funcionalidades en la nube. Es decir, los casos considerados por estos estudios están relacionados a operaciones CRUD básicas sobre un modelo de datos simplificado, o bien, el modelado de la navegación no es considerado o se limita a sólo ciertos casos, y la interfaz de usuario se restringe a elementos básicos (etiquetas, botones, cajas de texto) [23], [24], [25]. Cabe señalar que a pesar de las limitaciones expuestas, dichos artículos presentan evidencias positivas a favor de la adopción de MDD para el modelado y generación del tipo de aplicaciones de interés. Así que estos estudios podrían considerarse como puntos de referencia para la continuación de las investigaciones en esta área, y además, esta situación se podría considerar como una oportunidad para enriquecer dichos planteamientos.

Se podrían proponer estudios e investigaciones para enriquecer algunas de las propuestas seleccionadas (por ejemplo, [23], [24], [25]) considerando casos de usos más complejos y/o en ambientes industriales, y buscando otros patrones, procesos repetitivos, funcionalidades que puedan ser incorporadas dentro del enfoque dirigido por modelos. Una vez obtenidas mayores evidencias, sería posible elaborar conclusiones más certeras en cuanto a la factibilidad y el rendimiento de MDD en relación al mejoramiento de la portabilidad.

Si consideramos la perspectiva navegacional, WebRatio y MD² presentan un modelado de la navegación independiente del modelo de datos, con lo cual es posible obtener una navegación orientada a funciones. En MobiCloud y en la propuesta de Steiner et al. se habla de un modelado de navegación, pero limitado. Los casos considerados para el modelado de la navegación se acotan a subconjuntos de funcionalidades y características comunes entre distintas plataformas. En lugar de esto, se podrían establecer equivalencias entre las distintas implementaciones (de cada plataforma), de manera a brindar una mayor diversidad para la generación de las implementaciones. En el resto de los artículos, no identificamos el modelado de la navegación, por lo cual, podríamos decir que no hallamos una posibilidad de lograr una navegación orientada a funciones. Por eso, dichos trabajos podrían ser enriquecidos, teniendo en cuenta que el modelado de la navegación orientada a funciones puede ser una propiedad que mejoraría la portabilidad del diseño de las aplicaciones aislando las dificultades de portabilidad que se podrían presentar en la capa de datos [7], además de brindar mayor flexibilidad de uso, lo cual es clave para el éxito de las aplicaciones móviles.

Al no encontrar en la literatura analizada planteamientos

relacionados a un nivel de modelado específico para las arquitecturas emergentes, podríamos sugerir el estudio y la experimentación sobre el ASM, con transformaciones de modelo a modelo, del PIM al ASM, y el estudio de los posibles beneficios que pueda suponer para la portabilidad del diseño para distintas arquitecturas. Por ejemplo, en el caso de las aplicaciones con funcionalidades en la nube, se podrían necesitar procesos o estructuras de datos similares en ambos ambientes, entonces, a partir del PIM se podría obtener un diseño común para ambos entornos, y posteriormente, mediante la ejecución de reglas de transformación modelo a modelo, se podría obtener el modelado adaptado tanto para el lado móvil como el de la nube, con la incorporación de las restricciones, patrones, y conceptos propios de cada ambiente.

Por otra parte, sería interesante estudiar los beneficios que pueda suponer la separación del modelado de la navegación de los detalles de la capa de presentación, y compararlo con el enfoque de WebRatio, el cual en un mismo modelo considera el modelado de la navegación y detalles de la capa de presentación. En este sentido, WebRatio se podría comparar con MD², que considera una separación más clara de dichos detalles.

Considerando la separación del modelado de la navegación de los detalles de la capa de presentación y el modelado mismo de la navegación orientada a funciones, se distingue desde una perspectiva más general, el principio de la Separación de Intereses, el cual facilita la construcción de modelos, la comprensión y expresividad del lenguaje. En este sentido, como trabajo futuro, sería de interés hacer un estudio acerca de la separación de intereses en lenguajes de especificación de aplicaciones móviles, en vista de que los lenguajes que apliquen dicho principio podrían ser utilizados como insumo para un proceso basado en MDD y así mismo, podrían enriquecer dicho proceso.

También sería interesante confrontar y comparar las propuestas que adoptan un enfoque híbrido para el ambiente móvil contra las que generan implementaciones nativas. Se podría comparar por ejemplo, qué tan buena es la generación de código nativo generada directamente de los modelos en comparación a la implementación híbrida que se logra de la unión de MDD y Apache Cordova (propuesta de WebRatio). De esta manera, se podrá distinguir cual de las dos líneas es más adecuada para mejorar la portabilidad.

En cuanto al lado de la nube, el enfoque de desarrollo de código abierto es considerado como un enfoque de implementación más flexible para mejorar la portabilidad en relación con el problema del *vendor lock-in*⁶. Se podría considerar realizar un estudio que compare las combinaciones de MDD con implementación de código-abierto (enfoque que sigue WebRatio) contra la combinación de MDD con la implementación propietaria. Esto permitiría evaluar los aportes de estos dos tipos de combinaciones con relación al problema del *vendor lock-in* y, a partir de ello, obtener un mejor panorama de cual de las dos opciones es más razonable considerar (es decir, verificar si vale la pena consi-

6. Openshift, enlace: <https://goo.gl/UPRU9E>

derar MDD para la generación de funcionalidades en la nube sobre proveedores propietarios, o si sería más provechoso enfocarse directamente en la combinación de MDD y la implementación de código abierto).

Así mismo, en virtud de la consideración de una compensación entre desempeño y portabilidad, sería interesante realizar un estudio comparativo que verifique el desempeño entre una interfaz de comunicación REST (interfaz portable a través de las plataformas, o interfaz de diseño general) y una interfaz de comunicación nativa de cada plataforma (diseño especializado). Es decir, desde una perspectiva de implementación, ¿se justifica el tratamiento del problema de la portabilidad mediante la incorporación de REST a MDD, o sería más conveniente generar una implementación nativa en la comunicación, para lograr un mejor desempeño, tratando el problema de la portabilidad solo con MDD?

Como podemos observar, el mapeo sistemático realizado nos ha permitido identificar algunos aspectos positivos del uso de MDD para enfocar el problema de la portabilidad en las aplicaciones móviles con funcionalidades en la nube. Al mismo tiempo, un resultado positivo del estudio es que se han abierto diversos cuestionamientos que sugieren la necesidad de más estudios y evidencias empíricas para poder generalizar las consideraciones sobre la utilidad de MDD.

7. Limitaciones del Estudio

En esta sección se comentan algunas limitaciones relacionadas a la planificación y ejecución de este mapeo sistemático de la literatura.

En cuanto a la cadena de búsqueda, la misma podría ser enriquecida con términos adicionales, o más sinónimos de los términos ya considerados. En cuanto a la variación de las cadenas de búsqueda en las distintas fuentes de consulta, cabe mencionar que especialmente para los casos de Google Scholar y de DBLP, las cadenas sufrieron ciertos cambios, debido a que dichos servicios ofrecen formatos distintos de búsqueda. Las etapas de extracción y síntesis de datos no fueron planificadas con antelación, pero fueron llevadas a cabo de manera estructurada, recabando la información necesaria para poder responder las preguntas de investigación.

Finalmente, también cabe comentar que no se ha realizado una evaluación de calidad de los trabajos incluidos en el mapeo. En el trabajo de Genero et al. [17] se aclara que la inclusión de este punto no es esencial, ya que en un mapeo se pueden incluir tanto estudios teóricos como empíricos de cualquier tipo, y de esa forma puede ser difícil definir un mecanismo de evaluación común para todos los trabajos incluidos.

8. Conclusión

Este estudio presenta un mapeo sistemático de propuestas dirigidas por modelos para el desarrollo de aplicaciones móviles portables con funcionalidades en la nube. El proceso ha obtenido como resultado una limitada cantidad de propuestas, lo cual sugiere que es un área que debería seguir

madurando. Al mismo tiempo, las propuestas analizadas coinciden en señalar algunos aspectos positivos del uso de MDD para enfocar el problema de la portabilidad en las aplicaciones móviles con funcionalidades en la nube. Estas evidencias son todavía iniciales y requieren ser consolidadas a través de otros estudios. En este sentido, el análisis del mapeo sistemático realizado ha identificado diversos cuestionamientos que podrían orientar posteriores estudios y evidencias empíricas para poder generalizar las consideraciones sobre la utilidad de MDD al problema de la portabilidad en el contexto de aplicaciones aquí considerado.

Referencias

- [1] V. March, Y. Gu, E. Leonardi, G. Goh, M. Kirchberg, and B. S. Lee, "Mcloud: Towards a new paradigm of rich mobile applications," *Procedia CS*, vol. 5, pp. 618–624, 2011.
- [2] D. Sahu, S. Sharma, V. Dubey, and A. Tripathi, "Cloud computing in mobile applications," *International Journal of Scientific and Research Publications*, vol. 2, no. 8, pp. 1–9, 2012.
- [3] S. Abolfazli, Z. Sanaei, A. Gani, F. Xia, and L. T. Yang, "Rich mobile applications: genesis, taxonomy, and open issues," *Journal of Network and Computer Applications*, vol. 40, pp. 345–362, 2014.
- [4] M. Brambilla, A. Mauri, and E. Umuhoza, "Extending the interaction flow modeling language (IFML) for model driven development of mobile applications front end," in *Mobile Web Information Systems - 11th International Conference, MobiWIS 2014, Barcelona, Spain, August 27-29, 2014. Proceedings*. Springer International Publishing, 2014, pp. 176–191.
- [5] Z. Sanaei, S. Abolfazli, A. Gani, and R. H. Khokhar, "Tripod of requirements in horizontal heterogeneous mobile cloud computing," *CoRR*, 2012. [Online]. Available: <http://arxiv.org/abs/1205.3247>
- [6] P. Gupta and S. Gupta, "Mobile cloud computing: The future of cloud," *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, vol. 1, no. 3, pp. 134–145, 2012.
- [7] E. A. N. da Silva, R. P. Fortes, and D. Lucrédio, "A model-driven approach for promoting cloud paas portability," in *CASCON*, 2013, pp. 92–105.
- [8] C. Pons, R. Giandini, and G. Pérez, *Desarrollo de Software Dirigido por Modelos*, E. de la Universidad Nacional de La Plata, Ed. Editorial de la Universidad Nacional de La Plata (EDULP)/McGraw-Hill Educación, 2010.
- [9] M. Brambilla, J. Cabot, and M. Wimmer, *Model-Driven Software Engineering in Practice*, M. & Claypool, Ed. Morgan & Claypool, 2012.
- [10] M. González, L. Cernuzzi, and O. Pastor, "A navigational role-centric model oriented web approach - moweba," *Int. J. Web Eng. Technol.*, vol. 11, no. 1, pp. 29–67, 2016. [Online]. Available: <http://dx.doi.org/10.1504/IJWET.2016.075963>
- [11] N. Koch, A. Knapp, G. Zhang, and H. Baumeister, "Uml-based web engineering," in *Web Engineering: Modelling and Implementing Web Applications*. Springer, 2008, pp. 157–191.
- [12] 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.
- [13] J. Fons, V. Pelechano, O. Pastor, P. Valderas, and V. Torres, "Applying the oows model-driven approach for developing web applications. the internet movie database case study," in *Web Engineering: Modelling and Implementing Web Applications*. Springer, 2008, pp. 65–108.
- [14] C. Cachero, J. Gómez, and O. Pastor, "Ooh-method: un método de diseño de lugares web," in *Conference Proceedings, IDEAS 00. Cancún*, 2000, pp. 133–144.

- [15] B. Fling, *Mobile Design and Development: Practical concepts and techniques for creating mobile sites and web apps*. “O’Reilly Media, Inc.”, 2009.
- [16] T. Neil, *Mobile Design Pattern Gallery: UI Patterns for Smartphone Apps*, 2nd ed., I. O’Reilly Media, Ed. O’Reilly Media, Inc., 2014.
- [17] M. Genero, J. Cruz-Lemus, and M. Piattini, *Métodos de Investigación en Ingeniería del Software*, R.-M. Editorial, Ed. RA-MA, 2014.
- [18] H. Heitkötter, S. Hanschke, and T. A. Majchrzak, “Evaluating cross-platform development approaches for mobile applications,” in *Web information systems and technologies*. Springer, 2013, pp. 120–138.
- [19] A. Ribeiro and A. R. da Silva, “Survey on cross-platforms and languages for mobile apps,” in *Quality of Information and Communications Technology (QUATIC), 2012 Eighth International Conference on the*. IEEE, 2012, pp. 255–260.
- [20] B. Di Martino, G. Cretella, and A. Esposito, “Methodologies for cloud portability and interoperability,” in *Cloud Portability and Interoperability*. Springer, 2015, pp. 15–44.
- [21] B. Kitchenham and S. Charters, “Guidelines for performing systematic literature reviews in software engineering,” Keele University and University of Durham, Tech. Rep., 2007, version 2.3.
- [22] H. Heitkötter, T. A. Majchrzak, and H. Kuchen, “Cross-platform model-driven development of mobile applications with md 2,” in *Proceedings of the 28th Annual ACM Symposium on Applied Computing*. ACM, 2013, pp. 526–533.
- [23] A. H. Ranabahu, E. M. Maximilien, A. P. Sheth, and K. Thirunarayan, “A domain specific language for enterprise grade cloud-mobile hybrid applications,” in *Proceedings of the compilation of the co-located workshops on DSM’11, TMC’11, AGERE! 2011, AOPES’11, NEAT’11, & VMIL’11*. ACM, 2011, pp. 77–84.
- [24] D. Steiner, C. Turlea, C. Culea, and S. Selinger, “Model-driven development of cloud-connected mobile applications using dsls with xtext,” in *EUROCAST (2)*, ser. Lecture Notes in Computer Science, vol. 8112. Springer, 2013, pp. 409–416.
- [25] A. Ruokonen, L. Pajunen, and T. Systä, “On model-driven development of mobile business processes,” in *SERA*. IEEE Computer Society, 2008, pp. 59–66.
- [26] R. T. Fielding and R. N. Taylor, “Principled design of the modern web architecture,” *ACM Transactions on Internet Technology (TOIT)*, vol. 2, no. 2, pp. 115–150, 2002.
- [27] L. Richardson, M. Amundsen, and S. Ruby, *RESTful Web APIs*. “O’Reilly Media, Inc.”, 2013.