

Challenges of Migrating Legacies Web to Mobile: A Systematic Literature Review

V. Cajas, M. Urbieta, G. Rossi, and F. Domínguez

Abstract—The multi-device era comes thanks to mobile computing which gives remote access to resources from anywhere changing the people's life and opening new business opportunities. However, the legacy systems do not render the content properly in mobile devices because they were thought to be only compliant with Web browsers. Economic availability is often the reason why these have not been modernized. This work proposes a systematic literature review about the approaches used for the portabilization or modernization of web 1.0 business applications to mobile devices in the period 2006-2017, from SCOPUS, IEEE, and ACM. The search obtained 824 articles, where 37 were selected based on the inclusion criteria. The research questions were: P11. What have been the contribution to the portability of business applications to mobile devices? P12. What methods have been used? P13. What is the nature of the contribution found? P14. What are the objectives that are pursued in the research work?. The conclusions were obtained through the classification of selected papers with respect to focus, scope, type of research and type of contribution to give an answer to research questions. Then, the state of the art determines the main approaches such as model-driven development, content reorganization, augmentation, and specific migrations. However, certain research gaps were detected and detailed which represent exploration opportunities.

Index Terms—Evolution, Legacy application, Mobile applications, Migration, Systematic literature review.

I. INTRODUCCIÓN

EN la Web 1.0 [1] nacen los primeros sitios web: sencillos, sin interacción, de carácter divulgativo o publicitario para que las empresas pioneras ofrezcan servicios hacia la comunidad. La Web 2.0 surge bajo mejores condiciones tanto de hardware como de software [2] con el desarrollo de blogs, wikis, foros y redes sociales, siendo el usuario el protagonista, a través de la colaboración y conocimiento compartido. La Web 3.0 [3] corresponde a la web semántica, orientada a la construcción colaborativa de la información accesible por personas y máquinas. A partir de esta nueva evolución, se montan sobre una base de conocimiento nuevas aplicaciones tal como la inteligencia artificial y web semántica. Surge entonces la informática móvil que difiere de la informática de escritorio debido a la naturaleza dinámica del estado del sistema [4], provocando un cambio drástico en la vida de

las personas y todo lo referente a las comunicaciones [5]. Crea nuevos espacios para el internet de las cosas [6], el usuario puede acceder a la información desde cualquier lugar y desde cualquier tipo de dispositivo móvil [7]. Las aplicaciones empresariales empiezan a funcionar también en móviles como extensiones del core organizacional, ganando ventaja competitiva [8], junto con otros aplicativos de carácter personal como bases para nuevos modelos de negocios globalizados. Genera además nuevos requerimientos para el protocolo IEEE 802.11 [9] que se centra en las Wireless LAN.

Partiendo desde aplicaciones legacy Web 1.0 hasta Web 3.0 se vuelve un problema adaptar, migrar o portabilizar [10] de forma directa una aplicación Web empresarial que no fue visionada para su uso en dispositivos móviles. Este cambio puede demandar proyectos con duración de semanas o quizá años, y tener costos variables [11], en concordancia a su complejidad, entre otras características como su funcionalidad multiplataforma. Existen actualmente plataformas prácticamente incompatibles y cerradas (Android [12] e iOS [13]) donde se requiere desarrollos en bajo nivel utilizando lenguaje C para definir componentes reusables entre ellas. Es importante reflexionar también en los recursos limitados de estos dispositivos, procesadores poco potentes, pantallas de dimensiones reducidas, teclados pequeños o la inexistencia de ellos. Este hecho hace que la experiencia de los usuarios [14] no sea completamente satisfactoria, al encontrarse con problemas para acceder a los sitios web, generando de esta manera también una brecha tecnológica [15]. Entonces cada tipo de aplicación viene con desafíos, las empresas pueden optar por aplicaciones nativas o sacrificar parte de la funcionalidad para crear aplicaciones Web. Para destacar la relevancia del problema, se ha conducido una simple evaluación de los problemas que presentan sitios Web accedidos desde dispositivos móviles. Para ellos se aplicó la herramienta Google Search Console [16] para procesar 300 de los sitios web más buscados acorde al ranking de Amazon Alexa [17], lo que ha permitido identificar los principales problemas al correr aplicaciones web en dispositivos móviles. Es importante destacar que la evaluación tiene el propósito de explorar y no pretende reemplazar un estudio sistemático del problema. El resultado mostró que 54 sitios ¹ no se encuentran optimizados y tienen los siguientes problemas: (i) Los elementos en los que se puede hacer clic están demasiado cerca unos de otros. (ii) El contenido es más ancho que la pantalla. (iii) El texto es demasiado pequeño para leerlo. (iv) La ventana gráfica no se ha definido. (v) El sitio utiliza complementos incompatibles

V. E. Cajas, Universidad Tecnológica Indoamérica, Facultad de Ciencias Administrativas y Económicas, Quito, Machala y Sabanilla s/n, Ecuador, (e-mail: vivianacajas@uti.edu.ec).

M. M. Urbieta, Universidad Nacional de La Plata, Facultad de Informática, La Plata, 50 y 120, Argentina, (e-mail: matias.urbieta@lifa.info.unlp.edu.ar).

G. Rossi, Universidad Nacional de La Plata, Facultad de Informática, La Plata, 50 y 120, Argentina, (e-mail: gustavo.rossi@lifa.info.unlp.edu.ar).

F. J. Domínguez Mayo, Universidad de Sevilla en el Departamento de Lenguajes y Sistemas Informáticos, Avda. Reina Mercedes s/n. 41012, Sevilla, España, (email: fjdominguez@us.es).

¹<https://tinyurl.com/TestMobil-Cajas>

(Adobe Flash).

Por los inconvenientes detectados, existen varias iniciativas que tratan de solucionarlos, una de ellas es el diseño Web adaptable [18] el cual prioriza la correcta maquetación de una página en varios dispositivos [19]. Sin embargo los dispositivos antiguos no se adaptan correctamente a estos diseños, existen aplicaciones con pantallas de ancho fijo y al tratar de ocultar contenidos para facilitar la navegación se carga toda la página [20]. Además se debe considerar previamente en el diseño la no utilización de ciertos elementos: sliders, lightboxes, tabs que no permiten una buena experiencia para el usuario en móviles [21]. Sin embargo la maquetación correcta de las aplicaciones Web móviles no es el único problema existente a resolver, se debe también tomar en cuenta que en los negocios tradicionales el diseño de un sitio Web empresarial y un sitio móvil con características y opciones más livianas puede generar un abismo [22]. Adicionalmente estas aplicaciones pueden enmarcarse aún en arquitectura Web 1.0, por lo tanto se debe seleccionar cuidadosamente los contenidos adecuados para cada caso sin generar inconsistencias ni conflictos de requerimientos de negocio [23].

Este artículo se trata de una revisión sistemática de literatura [24] que extiende los resultados de un mapeo sistemático [25] realizado por los autores, en el cual se brindó una visión holística (mapa visual) del estado del arte en enfoques existentes para resolver los problemas asociados a la portabilización de aplicaciones legacies a móviles. Como resultado se identificó la necesidad de profundizar el estudio detallando y sintetizando cada uno de los trabajos seleccionados. Por lo tanto, para recalcar la diferencia entre el mapeo y la revisión sistemática, en este trabajo se presenta un estudio exhaustivo de la literatura pertinente a las preguntas de investigación siguiendo las directrices proporcionadas por Petersen [26], Kitchenham [24] y Brereton [27]. En segundo lugar, el período de tiempo corresponde a 11 años (2006-2017) determinado por la introducción del término web 2.0 [1]. El trabajo se encuentra organizado de la siguiente manera, la sección II explica la Formalización de la pregunta. A continuación, en la sección III de Revisión, se incluye el protocolo, fuentes y criterios de selección de artículos. En la sección IV de Ejecución de la Búsqueda, se describen los resultados preliminares. En la sección V de Extracción de la información se enumeran y sintetizan los trabajos seleccionados. En la Sección VI de Resultados y discusión, se muestra la evaluación realizada bajo los parámetros indicados anteriormente. En la sección VII constan las limitaciones del estudio. La sección VIII propone un enfoque para migrar y lo compara con los existentes. Por último, en la sección IX se pone en consideración las conclusiones y trabajo futuro de la investigación desarrollada.

II. FORMALIZACIÓN DE LA PREGUNTA DE INVESTIGACIÓN

Los conceptos de búsqueda mostrados en la Tabla I, se construyeron utilizando los pasos acorde a Brereton et al. [27] usando el criterio PICO (Problema, Intervención, Comparación, Resultado) [28]. Las preguntas de investigación (PI) [29] que responderá la revisión presentada en este documento

TABLA I
CONCEPTOS DE BÚSQUEDA Y SUS CONCEPTOS ALTERNATIVOS

PICO	Principales	Conceptos Alternativos
Problema	Portabilidad	Adaptación OR Modernización OR Migración OR Transformación
Intervención	Aplicaciones Legacy	Aplicaciones Web
Comparación Resultado	Móvil Enfoque	Multidispositivo OR Pantallas Pequeñas Marco

son: i) PI1. ¿Cuál ha sido la contribución a la portabilidad de las aplicaciones empresariales para dispositivos móviles? Permite conocer la tendencia en el área de móviles con los enfoques actuales e identificar de forma específica cómo se ha abordado el problema de la portabilización de las aplicaciones empresariales a móviles detectando propuestas en concreto. Para responder a esta pregunta se utilizará la categoría de tipo de contribución. ii) PI2. ¿Qué métodos se han usado? Determinar si los trabajos de investigación propuestos en este campo son prácticos o teóricos e identificar cómo han sido validados. Para responder a esta pregunta se determinará el ámbito de aplicación. iii) PI3 ¿Cuál es la naturaleza de la contribución encontrada? Identificar la naturaleza de los métodos encontrados para la portabilidad de aplicaciones empresariales a dispositivos móviles. Para responder a esta pregunta se clasificarán los enfoques encontrados en los estudios seleccionados. iv) PI4. ¿Cuáles son los objetivos que se persiguen en los trabajos de investigación? Señalar cuál es el principal punto de interés de la investigación y qué áreas han sido menos investigadas. De igual manera que con la pregunta anterior, para establecer la respuesta a esta pregunta se clasificarán las áreas de interés de los estudios seleccionados.

III. REVISIÓN

El método de revisión se basa en un protocolo de investigación [30], descrito a continuación.

A. Selección de las Fuentes

El periodo de búsqueda se ha centrado entre los años 2006, año en el que Tim O'Reilly [1] introduce el concepto de Web 2.0 y 2017 (inclusive), además para obtener los estudios se establece el uso de SCOPUS [31], IEEE Xplore Digital Library y ACM Digital Library.

B. Criterios de Inclusión y Exclusión

Los criterios explícitos de inclusión y exclusión para evaluar cada estudio, se muestran a continuación.

Criterios de Inclusión: (i) Documentos que cumplan la cadena de búsqueda. (ii) Documentos escritos en inglés. (iii) Journals, conferencias y documentos de talleres. (iv) Documentos publicados desde el 2006 hasta diciembre de 2017 (inclusive). (v) Documentos con propuesta de contribución relacionada. (vi) Documentos con enfoques de portabilización de aplicaciones web a móviles en software. Criterios de Exclusión: (i) Fuera de tópico. (ii) Documentos resumen de una presentación de taller. (iii) Documentos duplicados (misma investigación en diferentes bases de datos). (iv) Documentos en que el tema se mencione de manera introductoria o general, o sin (v) propuesta relacionada entre las contribuciones del documento. (vi) Documentos disponibles solo en forma de

TABLA II
DATOS EXTRAÍDOS

Metadatos básicos	Título, Autor y Fecha de Publicación
Faceta de Investigación	Opinión, Experiencia personal, Propuesta de solución, Validación de Propuesta y Evaluación.
Alcance de la Contribución	Modelado, Adaptabilidad, Verificación/Validación, Configuración/Despliegue, Evaluación de Proceso.
Ficha de datos.	Resumen, Herramientas, Origen, Destino, Tipo de Validación, Tipo de Migración, Conclusiones, Trabajo Futuro.
Tipo de contribución [33][34]	Técnica, Modelo, Proceso, Herramienta, Prototipo, Mapeo y Framework.
Ambito de Aplicación Enfoques	Academia, Industria, Mixto.
Area	Se determinarán con los estudios seleccionados.

resúmenes o presentaciones en PowerPoint. (vii) Migraciones de otro tipo de aplicaciones que no correspondan a sitios web y adaptaciones basadas solamente en hardware.

La selección de los estudios se realiza en dos etapas:

(i). Un investigador revisa el título y resumen de los estudios identificados por las búsquedas iniciales y los documentos irrelevantes son rechazados. La comprobación es realizada por otro investigador como inspector, si los investigadores no se ponen de acuerdo, el estudio debe ser incluido. (ii). Se obtienen las copias completas de los documentos no rechazados previamente. Se revisan por dos investigadores los criterios de inclusión/exclusión definidos, para obtener una lista final de los estudios. Si no resuelven sus desacuerdos se incluye la ayuda de un mediador.

IV. EJECUCIÓN DE LA SELECCIÓN

La búsqueda con la cadena resultante: (Portability OR Adaptation OR Modernization OR Migration OR Transformation) AND (Legacy Application OR Web Application) AND (Mobile OR Multi devices OR Small Screens) AND (Approach OR Framework), arroja 824 artículos ²: 354 Scopus, 160 IEEE y 300 ACM. Para la clasificación de los estudios, se utilizó el esquema de categorías sugerido por Wieringa [32]. Los datos extraídos de cada estudio se detallan en la Tabla II. De los estudios resultantes a continuación se puede observar las características:

Fuera del Tópico: 748 estudios que corresponden al 90,8%. Duplicados: 25 estudios que corresponden al 3,0%. Revisiones o mapeos: 7 estudios que corresponden al 0,8%. Migraciones de otros legacies: 7 estudios que corresponde al 0.81%. Incluido: 37 estudios que corresponden al 4,49%.

V. EXTRACCIÓN DE LA INFORMACIÓN

A continuación esta sección suministra la descripción de los estudios primarios seleccionados.

Lehtonen et al. "Towards user-friendly mobile browsing" [35] TutMobi prototipo que conserva el diseño de un sitio web en un dispositivo móvil. Preserva hasta el 90% del ancho de banda, a través de un proxy que contiene el 10% de la página original e incluye una imagen en miniatura de

ella con el contenido y coordenadas de los elementos más importantes. Del lado del cliente, se usa Java como mecanismo de implementación. En la evaluación demostró ser 35% más rápido que Opera Mini [36] y 85% más rápido que móvil Opera [37]. El inconveniente al conservar el diseño de la página, radica en que los usuarios tienen solo una interfaz visualmente orientada del sitio web. Los formularios HTML (HyperText Markup Language) no son compatibles con todo, falta refinar la función del zoom y la experiencia general del usuario debe mejorarse.

Di Santo et al. "Reversing GUIs to XIML descriptions for the adaptation to heterogeneous devices" [38] Proponen una herramienta de ingeniería inversa para interfaz gráfica de usuario (GUI) de Java. A través de transformaciones a descripciones abstractas basadas en lenguaje de marcado de interfaz extensible (XIML). Las descripciones resultantes son utilizadas por el framework TCPTTE [39] para ser procesadas en diferentes GUIs, que se adaptan dinámicamente a dispositivos heterogéneos en función de su perfil comunicado en el momento de la solicitud. Sin embargo hace falta proporcionar un soporte en tiempo de ejecución para permitir adaptación de widgets usando los perfiles y dominio en términos de actividades y tareas.

Cheng et al. "An adaptive and unified mobile application development framework for Java" [40] El framework Java GMA (aplicación móvil genérica) permite adaptar la interfaz de una aplicación a diferentes dispositivos. Considera su potencia informática y funcionalidades. Además, se propone una interfaz de servicio universal para programar aplicaciones (API) para acceder a diferentes servicios de back-end. Siendo el documento XML flexible y extensible, se puede ampliar fácilmente la base de datos para admitir más dispositivos finales. Está diseñado para ejecutarse en tres modos diferentes: navegador, cliente y maestro-esclavo. Sin embargo, hacen falta más pruebas que determinen resultados más específicos.

He et al. "A flexible content adaptation system using a rule-based approach" [41] Xadaptor permite la adaptación de contenido abstracto para dispositivos móviles a través de reglas con Prolog, que se invoca en función de la información del usuario a través de plantillas especificadas en un esquema XML. El analizador de contenido toma una página web, identifica objetos de acuerdo con las etiquetas HTML. Analiza sus propiedades, y genera hechos de página que se utilizan posteriormente para las inferencias de adaptación de contenido. Además usa CSS (Cascading Style Sheets) para páginas web reformateadas y finalmente lógica difusa para modelar el nivel de satisfacción del enfoque de adaptación. Sin embargo hace falta ampliar la base de reglas para admitir diseños extensos para todos los objetos de la estructura.

Ennai et al. "MobileSOA: A Service Oriented Web 2.0 Framework for Context-Aware, Lightweight and Flexible Mobile Applications" [42] MobileSOA, permite la interacción entre aplicaciones móviles y backends empresariales, con ejecución de servicios locales, ambientales y remotos a través de virtualización. Proporciona a los usuarios la capacidad de crear, aprovisionar y ejecutar aplicaciones dinámicas livianas. Funciona en Windows Mobile y en plataformas Linux. Utiliza un modelo de aplicación bilateral, un cliente ligero para mane-

²<https://tinyurl.com/y23qxjul>

jar detección de contexto, renderizado de interfaz de usuario (IU); y al lado del servidor un componente para el suministro de servicios basado en el contexto, política ejecución y generación de aplicaciones. Se encuentra planificado explorar la invocación asincrónica de las aplicaciones y construir un framework mashup.

Ahmadi et al. "Efficient web browsing on small screens" [43] Método que adapta automáticamente aplicaciones de escritorio a móvil, en dos fases principales: detección de contenido relacionado y adaptación de diseño, a través de un conjunto de reglas heurísticas, que se utilizan para detectar límites entre diferentes temas de la estructura y diseño visual. Se genera automáticamente una tabla de contenido para facilitar la navegación entre diferentes subpáginas. La evaluación muestra una mejora significativa en la usabilidad con respecto a Opera. Sin embargo hace falta refinar las reglas al ser comprobado que tienen menor rendimiento en ciertos sitios y la inclusión de criterios de importancia de bloque [44] para adaptación eficiente del diseño.

Kopf et al. "Adaptation of web pages and images for mobile applications" [45] Servicio de visualización para páginas web e imágenes que simplifica el diseño y la estructura para lograr una presentación compacta. Combina varias técnicas en dispositivos con diferentes resoluciones. Se utiliza un enfoque semiautomático, para identificar y recortar las fronteras irrelevantes, manejando regiones de interés. Se usa la interpolación lineal como último paso para escalar la imagen a su resolución final. Además, se ha desarrollado una herramienta de software para validar, agregar, eliminar o modificar los datos extraídos automáticamente. También simula diferentes dispositivos móviles, de modo que el usuario tenga la sensación de cómo la página web o imagen adaptada se verá. Para la evaluación se realizaron estudios con usuarios que confirmaron un gran beneficio que ofrece el sistema de adaptación, y solicitan que se simplifique la interfaz de la herramienta, bajando el esfuerzo manual, además falta adaptar audio y video.

Eap et al. "Personalised mobile learning content delivery: a learner centric approach" [40] Framework personalizado y consciente del contexto (PCAF), adapta automáticamente el contenido de aprendizaje diseñado para computadoras personales a móviles. A través de un complemento de aplicación web que captura patrones de acceso y los utiliza con configuraciones entrenadas, preferencias del alumno y perfiles de dispositivos para adaptar el contenido de aprendizaje a sus necesidades a través de un portal de proxy móvil (PPM). Se crean plantillas y se trata de copiar elementos seleccionados del contenido web y agregarlos al nuevo documento creado. La prueba de concepto del framework se realiza en la plataforma Java con Tomcat, se evidencia que es de bajo costo y proporciona una herramienta útil para desarrolladores de contenido de aprendizaje. A futuro se planea incluir un conjunto predeterminado de configuraciones para las pruebas de usabilidad.

Iñesta et al. "Framework and authoring tool for an extension of the UIML language" [41] Framework que genera interfaces de usuario para sitios de escritorio, sitios web dinámicos y aplicaciones móviles, a través de transformaciones de modelo.

Usa una adaptación de UIML (Lenguaje de marcado de interfaz de usuario) con el elemento vocabulario, donde los modelos de interfaz son transformados de niveles abstractos a niveles concretos, con definiciones independientes del renderizado. Se utiliza un meta-modelo central para representar enlaces entre elementos, así como widgets como contenedor superior. La herramienta se desarrolló con Eclipse, traduce el concepto de UIML directamente al editor y proporciona cuatro vistas: estructura del modelo, diseñador de interfaz de usuario, tabla de contenido y reglas de comportamiento. La aplicación sigue una arquitectura modular con tres capas (modelo-visor-controlador).

Xiao et al. "Mashup-Based Web Page Adaptation for Small Screen Mobile Device" [46] Nuevo esquema Mashup de adaptación de página web para dispositivo móvil. La página original se divide en pequeños bloques que son adecuados para la visualización móvil. Luego, los bloques se clasifican analizando capas de relación iterativa entre ellos en horizontal y vertical. Además, se brindan API de servicios web para que los usuarios de terminales móviles o proveedores de servicios de terceros pueden integrar los bloques de forma rápida y cómoda, estableciendo nuevas aplicaciones web personalizadas y variadas. Finalmente, el esquema propuesto se pone en práctica como prototipo, que verifica el esquema y muestra valores tanto técnicos como desde el punto de vista económico.

Paternò, F. "MARIA: A universal, declarative, multiple abstraction-level language for service-oriented applications in ubiquitous environments" [47] MARIA, lenguaje basado en modelos para interfaces de usuario multidispositivos, con diferentes niveles de abstracción y una herramienta asociada que identifica 4 aspectos: nivel de abstracción de interfaz, granularidad, aspectos que se ven afectados por la composición de la interfaz y el momento en que la composición ocurre (tiempo de diseño / tiempo de ejecución). Además ofrece la posibilidad de personalizar las transformaciones de modelo a modelo (hacia adelante y hacia atrás) y las reglas, permitiendo pasar de un modelo a la implementación final. Se prueba con una aplicación del juego Pac-Man para transformarla a una versión adaptada para el móvil iPhone, con ciertas características como multitouch y acelerómetro, está pendiente continuar probando la usabilidad.

Paternò et al. "Desktop-to-mobile web adaptation through customizable two-dimensional semantic redesign" [48] Arquitectura basada en modelos, transforma aplicaciones web de escritorio a móviles, a través de un proxy. Se describe un algoritmo de reglas inteligentes y el soporte de herramientas (rediseño semántico bidimensional personalizable), explotando descripciones lógicas de la interfaz de usuario capaces de capturar información semántica de interacción que indica el propósito de los elementos de la interfaz. Se compara la solución con navegadores móviles existentes para objetivos similares como Mowser [49] y Skweezer [50], observando un mejor rendimiento. Esta solución no requiere que las aplicaciones se implementen utilizando un kit de herramientas particular para que puedan adaptarse y permite a los usuarios configurar el proceso y un mayor control sobre el cálculo de costos y los resultados de adaptación. Contenido como Flash,

los applets de Java no están actualmente adaptados. Además, en este trabajo no considera otros tipos de transformaciones como adaptación gráfica a vocal.

Armenise et al. "A tool for automatic adaptation of web pages to different screen size" [51] Herramienta con algoritmo genético evolutivo, capaz de entregar contenido adaptado a dispositivos móviles. A partir de una página determinada, busca diseños alternativos para ajustar mejor el contenido al tamaño reducido de la pantalla de destino, teniendo en cuenta las preferencias del diseñador, la visualización del objetivo y las limitaciones del dispositivo. Los resultados experimentales muestran que este enfoque es factible y puede competir con el diseño hecho por humanos, sin embargo, no es capaz de manejar múltiples preferencias y restricciones conflictivas. Además, se debe conservar el diseño dado, ya que la estructura no se modifica. Otras limitaciones se refieren al costo computacional para buscar y evaluar soluciones.

Guirguis et al. "A smart framework for web content and resources adaptation in mobile devices" [52] Framework para adaptación de contenido web y recursos en dispositivos móviles (WRAMD) a través de una herramienta para web masters que permite editar conjuntamente web móvil y web tradicional. Se podría integrar con cualquier sistema de gestor de contenido (CMS) o cualquier sistema basado en plantilla. Además los dispositivos móviles obtendrían un ancho de banda minimizado de acceso, sin embargo se han realizado pruebas experimentales con pocos web masters con Joomla CMS por lo que queda pendiente determinar la usabilidad con clientes reales.

Li et al. "Web page layout adaptation based on WebKit for e-paper device" [53] Método de adaptación de diseño de páginas web basado en el navegador móvil WebKit que a través de un agente de adaptación de diseño, preserva toda la estructura del HTML original modificando los estilos en el CSS, para cambiar las posiciones, eliminar el desplazamiento horizontal, ajustar tamaños de fuente, filtrar información inútil. Los resultados demuestran que la velocidad de carga mejora porque los estilos personalizados son más simples que los originales y el método es aceptable para dividir las páginas ordenadamente porque las subpáginas generadas son desplegadas en widgets. Queda a futuro extender estos principios a otros navegadores móviles y detectar la semántica de páginas web para proporcionar más significado a los títulos para subpáginas.

Koehl et al. "M.Site: Efficient content adaptation for mobile devices" [54] M.Site es un enfoque de adaptación de sitios web existentes al paradigma móvil. Un administrador del sitio selecciona visualmente objetos dentro de una página web y asigna atributos a objetos a partir de una colección de modificaciones predefinidas. El sistema propuesto genera código para una sesión múltiple, a través de un servidor proxy basado en PHP. Para proporcionar una interfaz simplificada, con herramientas de manipulación DOM, que no dependen de navegadores especiales o de servicios remotos de terceros. El almacenamiento en caché del lado del servidor permite amortizar los costos de procesamiento en el cliente. Construyen un prototipo y evalúan su eficacia en una aplicación web compleja cuyo resultado es una experiencia de navegación más

agradable en el dispositivo móvil.

Rajkumar et al. "Dynamic web page segmentation based on detecting reappearance and layout of tag patterns for small screen devices" [55] El algoritmo (DWS) mejora la segmentación dinámica de páginas web, basado en reconocimiento de patrones de reaparición de etiquetas en el DOM. Genera nodos implícitos para segmentar correctamente bloques anidados y luego el usuario selecciona hipervínculos basados en su área de interés. Los bloques informativos están evaluados por la cantidad de información, asignando un peso de importancia a cada nodo, teniendo en cuenta la cantidad de reapariciones de patrones del nodo. En los experimentos se compara con REPS y DWS y es más adecuado para entornos web dinámicos.

Challiol et al. "Crowdsourcing mobile web applications" [56] Enfoque de aumentación del lado del cliente en donde los usuarios pueden crear sus scripts personalizados en aplicaciones existentes a través del framework MOWA (Mobil web augmentation), para mejorar la experiencia a través de plugin. La evaluación demuestra que el enfoque es factible y permite desarrollar de una forma rápida y eficaz en comparación a otras técnicas de desarrollo web. Sin embargo los cambios que se pueden realizar son limitados y están trabajando en la creación de herramientas visuales para pre-construir una aplicación específica con el fin de simplificar el proceso de desarrollo.

Shaari et al. "Achieving "One-Web" through customization and prioritization" [57] Reestructuración del contenido para diferentes dispositivos a través de la priorización de elementos con clasificación de los div en un atributo rank (prioritario/no prioritario). Este motor diseñado para operar en el servidor, reordena, muestra y elimina elementos según la prioridad establecida por los usuarios o desarrolladores, reduciendo el tamaño de la página y los tiempos de descarga. Esta priorización se almacena en una base de datos del servidor para las configuraciones por default y las configuraciones definidas por el usuario. La evaluación con usuarios muestra que las tareas se cumplen más rápido con la versión priorizada, sin embargo, como limitación se observa que se debe tener una experiencia alta como usuario para poder priorizar los elementos y que es complicado interactuar en sitios con demasiados scripts en el cliente.

Amendola et al. "Adapting CRM systems for mobile platforms: An MDA perspective" [58] Migración de aplicaciones heredadas a móviles a través de la reingeniería que integra técnicas tradicionales de ingeniería inversa como análisis estático y dinámico con Model Driven Development. Usan como herramienta Eclipse hacia plataforma Android, se aplica como caso de estudio un CRM java. La primera etapa consiste en generar diagramas UML, recuperando diagramas de clases para detectar relaciones entre los diversos componentes de los módulos principales. Sin embargo como limitaciones del trabajo se presenta las diferencias existentes entre la versión de Java de Android con la estándar. Por otro lado debido a limitaciones tecnológicas y características de la plataforma es necesario crear un equivalente con diferentes widgets.

Albasir et al. "Smart mobile web browsing" [59] Sistema que adapta contenido web, controlando la cantidad de anuncios

a mostrarse en la página. Acorde al nivel actual de la batería del dispositivo para prolongar su vida útil, y el tipo de red para preservar el ancho de banda necesario para descargar las páginas. Equilibra la satisfacción de los editores de las páginas web y los usuarios finales a través de un navegador inteligente y un servidor que extrae la información necesaria para tomar la decisión, aplica determinada política. Al ser un borrador del algoritmo, en el trabajo futuro, se propone mejorarlo.

Yun et al. "MobiTran: tool support for refactoring PC websites to smart phones" [60] MobiTran, herramienta de desarrollo en línea para refactorizar visualmente sitios web de PC y adaptarlos a móviles como aplicación ligera; con la dirección URL del sitio web original, a través de un nuevo algoritmo para dividir la página en un conjunto de piezas de acuerdo con el tamaño de la pantalla, mientras se conservan la mayoría de los comportamientos CSS y JavaScript. Los desarrolladores deben ajustar las aplicaciones web semiacabadas con plantillas de diseño y estilo. La recuperación de páginas se hace a petición, mientras que los comportamientos táctiles y la integración se realizan con características de aplicaciones locales. Proporciona algunos widgets visuales útiles, además define un Lenguaje de generación de plantillas (TGL) para describir cómo generar la plantilla de destino para páginas web de la misma categoría. Al ser un paper corto no se evidencian los experimentos realizados ni sus estadísticas.

Sumit Pandey. "Responsive design for transaction banking - a responsible approach" [61] Estudio de caso del primer proceso de diseño impulsado por prototipos móviles. A través de una aplicación híbrida de teléfono inteligente y tableta para transacciones de banca, con estrategia web adaptativa. Reconocen que el modelo de desarrollo seleccionado para esta aplicación presenta desafíos únicos por la rápida tasa de crecimiento de los dispositivos habilitados para Internet capaces de mostrar la web completa y también se detallan las principales conclusiones del diseño para ese contexto.

Toile et al. "Adaptation of composite E-Learning contents for reusable in smartphone based learning system" [62] Método para adaptar contenido de aprendizaje virtual basado en PC hacia dispositivos móviles. Extrae la información, realiza la transcodificación y genera un nuevo contenido como un servicio web (SOA) para aplicaciones de teléfonos inteligentes. El método funciona al 100% con éxito al extraer el contenido de las muestras con Microsoft Producer y generar nuevo contenido para acceder en el teléfono inteligente. La prueba de usabilidad muestra el puntaje aceptado 94%. Sin embargo, existen varios sistemas educativos de aprendizaje virtual que han dejado de utilizarse por falta de adaptación y problemas tecnológicos, estaría pendiente evaluarlos con este enfoque.

Badam et al. "Polychrome: A cross-device framework for collaborative web visualization" [63] Framework genérico para construir visualizaciones de dispositivos interactuando con el DOM, construido en JavaScript, los dispositivos participantes no necesitan ningún software especial más allá de un navegador web moderno. El intercambio de interacciones y la sincronización entre dispositivos se realiza a través de una red segura punto a punto (P2P), mientras que los datos persistentes como detalles de sesión son administrados por

un servidor dedicado. Proporciona tres contribuciones: sitios web legados aumentados, visualizaciones web colaborativas, gestión de consistencia y sincronización. También módulos para almacenar la interacción del usuario (representado como operaciones). En combinación con el estado inicial de un sitio web, los registros de interacción son útiles para sincronizar dispositivos dentro del entorno de colaboración, administración de coherencia y reproducción de interacción.

Yang et al. "Panelrama: Enabling easy specification of cross-device web applications" [64] Panelrama es un framework de pantallas distribuidas basando en HTML5, para el desarrollo de aplicaciones complejas multiusuario. Mediante la combinación de distribución de UI, el posicionamiento de paneles de contenido condicionado por las características de los dispositivos y la posibilidad de ser extendido por desarrolladores. Permite a los usuarios interactuar con una sola aplicación desde diferentes dispositivos. A pesar de ser una solución potente, el framework está orientado al desarrollo de nuevas aplicaciones y portabilizar aplicaciones legacy.

Kovachev et al. "Direwolf: A framework for widget-based distributed user interfaces" [65] Componente de migración sincronizado a través del framework DireWolf, para aplicaciones ricas con interfaces de usuario distribuidas (DUI), en una federación de dispositivos básicos heterogéneos compatibles con navegadores web modernos como computadoras portátiles, teléfonos inteligentes y tabletas. Se basan en una tecnología integral combinada con la comunicación entre widgets multi-plataforma y la movilidad de sesión sin interrupciones. Permite aplicaciones de colaboración en tiempo real, durante la cual los widgets pueden volverse inactivos si detectan desconexión y puede recuperar el estado como una instantánea para continuar la tarea, se basa en el SDK1 ROLE de Java. Los resultados muestran que el framework facilita el caso de uso de la anotación de video semántica colaborativa. Sin embargo, hace falta extenderlo para que sea compatible con colaboración multiusuario de multidispositivos, e investigar qué combinaciones de dispositivos y tipos de widgets serían las más relevantes para su distribución.

Xiang et al. "Effective Page Segmentation Combining Pattern Analysis and Visual Separators for Browsing on Small Screens" [66] El método PAS (Análisis de Patrones y Separadores Visuales) se implementa en C++ y funciona en tres pasos: (i) Generar un árbol de etiquetas refinado con información de ubicación y diseño del árbol DOM original. (ii) Reconocer y fusionar patrones inexactos recursivamente. (iii) Segmentar a los bloques restantes con separadores visuales y regiones de color, similar a VIPS (visión basada en segmentación de páginas) [67]. En resultados supera a VIPS cuando la cantidad de bloques es grande, especialmente para páginas generadas automáticamente a partir de plantillas. Para lograr mejores resultados, se deben considerar más señales, como el uso del análisis semántico latente de texto y se requiere tener un framework flexible para incorporar diferentes técnicas.

Tseng et al. "Migratom.js: A JavaScript migration framework for distributed web computing and mobile devices" [68] Migratom.js, un framework Java-Script, que permite delegar la ejecución de operaciones en un servidor para disminuir la carga de procesamiento en dispositivos móviles. La descarga

de tareas y la migración de código con el paradigma de programación basado en flujo, se utilizó HTML5 WebSocket y Assistant Server para soportar los sistemas de comunicación clientserver y device-to-device. Primero permite a los desarrolladores dividir aplicaciones web en componentes con la biblioteca FBP.js, y luego la migración dinámica de componentes entre nodos por el administrador de migraciones. Está diseñado con portabilidad multiplataforma, peso ligero y API sencillas para simplificar el uso para dividir un programa en tareas de grano fino o grano grueso. Como resultados en los estudios de caso, las aplicaciones web se aceleraron descargando los componentes de cómputo intensivo a fuentes de cómputo superiores. Se pretende a futuro incrementar los escenarios emergentes.

Sarkis et al. "MSoS: A Multi-Screen-Oriented Web Page Segmentation Approach" [69] MSoS, un enfoque orientado a pantallas múltiples, para segmentar páginas web, automático y guiado, basado en: análisis visual, análisis DOM y análisis de contenido de funciones como en FOM [70]. Para lograr la distribución de la aplicación, actualiza el valor pG (parámetro de granularidad) en función del contenido y al tipo de interacción entre un usuario y un bloque de contenido. Define las funciones desde la perspectiva del usuario final y no desde el autor. Los experimentos en un conjunto de aplicaciones existentes contienen elementos multimedia. Se evaluó tres parámetros de rendimiento con respecto a BOM [71]: coherencia visual de los bloques, corrección de la función atribuida a cada bloque y que los bloques se mantengan separados. Es un método adaptativo, con una precisión del 81%, con mejores resultados de segmentación, especialmente en las regiones críticas de la página. Sin embargo se encuentra pendiente extender esta evaluación y validar de forma cualitativa y cuantitativa con diferentes métodos de segmentación.

Wang et al. "Towards Web Application Mobilization via Efficient Web Control Extraction" [72] Enfoque para extraer partes de una página como control web ejecutable. Crea un gráfico de dependencia de código ejecutado por el usuario y realiza el corte basado en el gráfico de dependencia, partiendo de que las aplicaciones web son impulsadas por eventos [73]. Extrae los elementos del DOM seleccionados para la presentación del control web específico e infiere controladores de eventos. Para probar el enfoque se implementa una herramienta prototipo para dos casos de estudio, sin embargo pese a demostrarse que es eficiente ahorrando del 26% al 98% de ancho de banda, se requiere un trabajo manual para adaptar los controles web extraídos a diferentes pantallas móviles, y se planea en el trabajo futuro que funcione automáticamente.

Bouzit et al. "Evanescent Adaptation on Small Screens" [74] Exponen el concepto de Adaptación Evanesciente, cuyo principio es una representación basada en dos capas: ítems pronosticados en la primera capa que se muestran arriba y desaparecen progresivamente bajo el control del usuario y abajo como segunda capa la lista de elementos completa. El enfoque es basado en ICD (desapareciendo en contexto)[75], mejorado con la integración del control de usuario. Para probar se usa teléfonos Android, con codificación en Java. De acuerdo a los resultados el trabajo requiere incluir ciertos grados de control del usuario aparte del botón salir, suavizar el efecto

de desaparición, ya que aún genera algunos errores. También puede mejorar la forma de mostrar la transición entre los elementos que aún pueden clicarse y los que simplemente se desactivan. Además la razón de ser de la predicción.

Miján et al. "Supporting personalization in legacy web sites through client-side adaptation" [76] Metodología para agregar la personalización en el Navegador Web a un sitio web existente el cual no es evolucionable. Enfoque orientado a diseñadores sin experiencia como desarrolladores o propietarios de sitios web. Se definen un conjunto de reglas de personalización que se aplicarán en el lado del cliente con las mínimas alteraciones en la aplicación de backend. Estos ajustes permiten mejorar la experiencia del usuario incorporando nuevos comportamientos, sin embargo no incorpora adaptaciones para que las aplicaciones funcionen adecuadamente en dispositivos móviles. El reporte es preliminar debe ser validado y presenta una herramienta como prueba de concepto.

G. Huang et al. "Programming Situational Mobile Web Applications with Cloud-Mobile Convergence: An Internetware-Oriented Approach" [77] Internetware, para aplicaciones web móviles situacionales [78] (contexto + datos sensoriales) con el paradigma Service-Model-View-Controller (SMVC), variante de MVC, que agrega un elemento denominado servicio junto al modelo. Adapta los dispositivos móviles sintetizando los recursos aprovechando la nube en máquinas virtuales Docker [79]. Considera la diversidad de hardware, acceso confiable a datos y flexibilidad en redes dinámicas, aún fuera de línea. Usa patrones de interacción de usuario, se implementa con servicios web en la nube. Sin embargo para un funcionamiento óptimo se espera conexión en una red estable de gran ancho de banda. Los usuarios necesitan cuenta en la nube para registrar la máquina virtual, se manejan plantillas y reorganización del DOM para aplicaciones web heredadas, con un algoritmo de particionamiento en un lenguaje específico de dominio (DSL) [80]. A futuro, se planea un IDE visual, tener un DSL más óptimo, eficiente, descriptivo, declarativo, con mejores capacidades de verificación de tipos, transformación, y más funciones integradas.

H. Li et al. "Extracting Main Content of Webpage to Enhance Adaptively Rendering for Small Screen Size Terminals" [81] Describen un algoritmo de transformación para adaptar páginas de acuerdo con la densidad del texto y características de los enlaces html de la página. Extraen el contenido principal y lo adaptan al tamaño de las terminales móviles. Los resultados demuestran que el método de transformación no depende de un software específico o plantillas de página web. En el trabajo futuro, se necesita manejar páginas web con imágenes y videos.

Bosetti et al. "An approach for building mobile web applications through web augmentation" [82] Incorporan conceptos de hipermedia física a aplicaciones web convencionales en dispositivos móviles. Mediante el cómputo de la posición del usuario, se identifican puntos de interés y su información relacionada tal como descripción para enriquecer el contenido de la página Web original utilizando aumentaciones del lado del cliente. Presentan la aplicación de una herramienta responsable de enriquecer las páginas considerando como caso de estudio un Museo. Sin embargo, debido a que se ha probado

a través de dos prototipos para dos escenarios en un primer experimento, se requieren más pruebas.

Sarkis et al. “A multi-screen refactoring system for video-centric web applications” [83] Sistema de refactoring automático que crea Sistemas de Multi-Pantallas (SMP). Resuelve la caracterización del ambiente, la división y distribución de la UI, y el funcionamiento de la aplicación en múltiples dispositivos. Analiza la estructura de la aplicación y su aspecto visual para segmentar la UI en bloques. Una arquitectura Maestro-Esclavo realiza la sincronización y actualización de los cambios entre los distintos dispositivos en tiempo real. Es capaz de rediseñar la interfaz dinámicamente para ajustarse al tipo de dispositivo, es decir, ser adaptativo. Se evalúa cuantitativamente el sistema en un conjunto de aplicaciones existentes. Sin embargo, el concepto de entorno debe extenderse para incluir no solo las características del dispositivo, sino también las preferencias del usuario en diferentes escenarios.

VI. RESULTADOS Y DISCUSIÓN

Acorde con Randolph [29] las preguntas de investigación en una revisión de literatura pueden ser únicamente respondidas mediante la lectura de estudios de investigación primaria, por lo tanto, a continuación se da respuesta a las preguntas de investigación formuladas en la sección II con base a los 37 trabajos seleccionados: (i) 8 trabajos de ACM siendo el 21.62%. (ii) 6 trabajos de IEEE siendo el 16.22%. (iii) 23 trabajos de SCOPUS siendo el 62.16%. Esto puede justificarse por el hecho de que SCOPUS es una base de datos que indexa el trabajo publicado por varios editores [84], incluidos IEEE y ACM. Por lo tanto, como respuesta a PI1, los métodos investigados en el periodo analizado corresponden a contribuciones a través de herramientas con 15 trabajos (P2, P3, P6, P7, P8, P9, P11, P12, P13, P14, P15, P22, P32, P33, P38), frameworks con 10 trabajos (P5, P16, P18, P19, P25, P26, P27, P29, P34, P36), prototipos (P4, P10, P21, P23, P28, P31) y técnicas con 6 trabajos (P1, P17, P20, P24, P30, P35). La línea cronológica acerca del periodo en donde se encuentran los estudios seleccionados se presenta en la Figura 1, en la cual se pueden observar los principales hitos que marcaron historia en la tecnología, tales como: la ingeniería web dirigida por modelos (MDWE) segunda generación, aplicaciones de página única (SPA). A continuación el lanzamiento de IOS. Luego el surgimiento de Android y Chrome. También se encuentra Apache Córdoba. Además nacen las aplicaciones para Smartwatches y el asistente de Amazon Alexa.

En cuanto al ámbito de investigación que responde a PI2, la mayor parte de métodos han sido validados dentro de la academia 64.86%, luego en la industria 29.73% y la minoría son trabajos conjuntos 5.40%. Por otro lado los resultados muestran que 59,46% de los documentos incluidos corresponden a trabajos provenientes de conferencias, luego el 21.62% son artículos, el 10.81% simposios, corresponden a revistas el 5.40% y workshop 2,70%. Además, se puede mencionar que los métodos de investigación y validación utilizados corresponden a estudios de casos, que facilitan la validación de la solución propuesta cuando se cuenta con varias aplicaciones

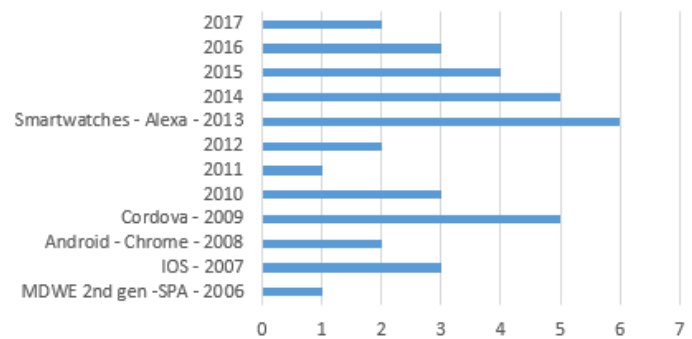


Fig. 1. Línea Cronológica.

en las que se realizó la implementación de la metodología. Experimentos con usuarios que generalmente corresponden a la validación de temas de usabilidad y determinación de cumplimiento de tareas. Los prototipos o simulaciones son adecuados cuando se desea realizar una prueba parcial de la metodología o no se cuenta con la disponibilidad del hardware requerido. Para responder a PI3, se detectaron varios enfoques que permiten dar solución parcial o total, al problema de la portabilidad de aplicaciones, los cuales se evidencian en los estudios de la Tabla III, además de las categorías correspondientes. Por otro lado, en la Figura 2 se puede observar a primera vista el resumen de enfoques obtenidos en esta revisión. Además la Tabla IV muestra una comparación de los enfoques pertenecientes a los trabajos incluidos en el estudio bajo parámetros que caracterizan la portabilización como enfoques evolutivos, automáticos, basados en el comportamiento del usuario, si son generales, la calidad de la interfaz obtenida, usabilidad, si abstrae el modelo de navegación del usuario, por último si el tipo de migración es solamente de interfaz o de lógica de negocio. A continuación se resumen las particularidades de los enfoques: (i). Desarrollo de Software Dirigido Por Modelos (MDD): Uso de modelos como principales artefactos del proceso de desarrollo, utilizados para dirigir las tareas de comprensión, diseño, construcción, pruebas, despliegue, operación, administración, mantenimiento y modificación de los sistemas. (ii). Manipulación Del DOM: Crea, elimina, actualiza, mueve elementos y contenido del árbol DOM de las páginas para lograr personalizarlas de acuerdo al dispositivo. (iii). Computación en la nube: Permite el acceso de red ubicuo bajo demanda a un grupo compartido de recursos informáticos configurables que pueden aprovisionarse y lanzarse rápidamente con un mínimo esfuerzo administrativo o la interacción del proveedor de servicios. (iv). Traductores: Programas que reciben como entrada código escrito en cierto lenguaje y producen como salida código en otro lenguaje. (v). Desarrollo Multipantalla: Surge debido a la infinidad de modelos y tamaños de pantallas de dispositivos móviles, PCs existentes actualmente en el mercado, lo cual es un aspecto de relevancia para considerar en el diseño de sistemas. (vi). Migraciones específicas: Instalar software en una nueva plataforma o en una versión mejorada, incluye todo, hasta considerar la mejor configuración de hardware para soportar las aplicaciones y su negocio, dentro de los límites como conjuntos de habilidades disponibles, presupuestos limitados y cumplimiento normativo. (vii). Aumentación: Proceso de

TABLA III
DISTRIBUCIÓN DE ESTUDIOS POR CATEGORÍA

No.	Año	Fuente	Enfoque	Ambito	Alcance	Faceta	Contribución	Documento
P1	2006	A	DOM [85]	A	EP	PS	T	W
P2	2007	S	Migraciones específicas [86]	A	EP	PS	H	C
P3	2007	S	Migraciones específicas	A	VV	PS	H	C
P4	2007	S	XML [87]	I	VV	PS	P	A
P5	2008	I	SOA [88]	I	VV	PS	F	C
P6	2008	A	DOM	A	VV	PS	H	C
P7	2009	S	Multipantalla [89]	A	PA	V	H	C
P8	2009	S	XML	A	VV	PS	H	A
P9	2009	S	Modelos [90]	A	PA	PS	H	A
P10	2009	I	Mashups [91]	A	VV	PS	P	C
P11	2009	A	Traductores [92]	I	EP	PS	H	A
P13	2010	S	Modelos	A	PM	V	H	C
P14	2010	S	Programación Genética [93]	A	VV	PS	H	C
P16	2010	S	Multipantalla	A	VV	PS	H	C
P17	2011	S	DOM	A	VV	PS	H	C
P18	2012	S	DOM	A	VV	PS	F	C
P20	2012	I	DOM	A	VV	PS	T	C
P21	2013	S	Aumentación [94]	A	VV	PS	F	C
P22	2013	S	Modelos	A	VV	PS	F	A
P23	2013	S	Modelos	A	PA	PS	T	C
P25	2013	S	DOM	A	VV	PS	P	C
P26	2013	A	Traductores	M	EP	PS	H	C
P27	2013	A	Prototipos	I	VV	PS	P	C
P29	2014	S	Migraciones específicas	I	PA	PS	T	C
P30	2014	S	DOM	I	EP	PS	F	C
P31	2014	S	Multipantalla	I	EP	E	F	C
P32	2014	S	Widgets [95]	A	EP	E	F	R
P33	2014	I	DOM	A	VV	PS	P	C
P35	2015	S	Arquitectura [96]	I	EP	PS	F	S
P36	2015	A	XML	I	VV	PS	T	S
P37	2015	A	DOM	A	EP	PS	P	S
P38	2015	A	Técnicas de Visualización	M	EP	PS	H	S
P39	2016	S	DOM	A	VV	PS	H	C
P41	2016	I	Computación en la Nube [97]	I	VV	PS	F	A
P42	2016	I	Algoritmos [98]	A	EP	PS	T	C
P43	2017	S	DOM	A	VV	PS	H	R
P44	2017	S	Multipantalla	I	VV	PS	H	A

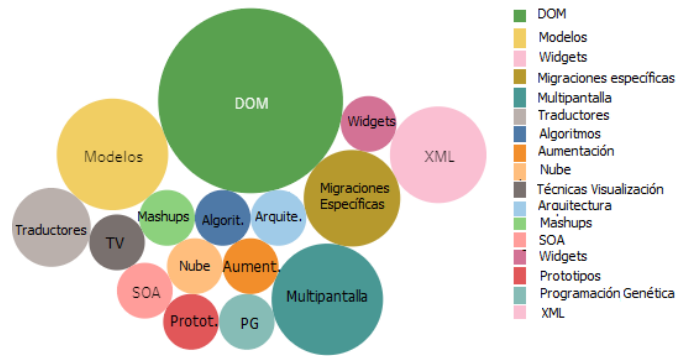


Fig. 2. Enfoques.

una considerable experiencia en programación. (x). Técnicas de Visualización: Son técnicas gráficas de análisis de datos que permiten mejorar la presentación. (xi). Algoritmos: Los algoritmos reciben una entrada y la transforman en una salida, debe ser definido, finito y eficiente para encontrar la solución en el menor tiempo posible. (xii). Arquitectura: Asocia los requerimientos con los componentes del sistema que la implementarán, esta descripción arquitectónica incluye componentes y conectores (en términos de estilos) y la definición de operadores que crean sistemas a partir de subsistemas o, en otros términos, componen estilos complejos a partir de estilos simples. (xiii). XML: El Lenguaje de marcado extensible, es un subconjunto de SGML (Standard Generalized Markup Language) diseñado para facilitar la implementación y la interoperabilidad con SGML y HTML. (xiv). Programación genética: Trabaja sobre una población de individuos, cada uno de ellos representa una posible solución al problema que se desea resolver. (xv). Widgets: Pequeñas aplicaciones que tienen como principal objetivo mostrar y dar fácil acceso a algunas de las principales funciones del móvil, ofreciendo informaciones de manera visual y sin requerir ser abiertos al estar ubicados en el escritorio. (xvi). Prototipo Adaptable Web: Aunque no es un sistema completo, posee las características del sistema final o parte de ellas.

Respondiendo a PI4, el tipo de migración que se abordó en los estudios seleccionados se encuentra clasificado de la siguiente manera: Adaptación de la Interfaz (Estudios P1, P4, P6, P7, P8, P10, P11, P12, P13, P14, P15, P16, P17, P18, P19, P20, P21, P22, P23, P24, P26, P27, P28, P30, P32, P33, P35). Migración con interfaz y lógica de negocio (Estudios P2, P3, P5, P9, P25, P29, P31, P34, P36). Por otro lado, la mayoría de los artículos proponen soluciones generales, sin ámbito específico de aplicación. Otros corresponden a soluciones orientadas a temas de aprendizaje móvil, aumentación, trabajo colaborativo, adaptaciones para aplicaciones de la salud, catálogos generales, juegos, turismo. Estos resultados evidencian que la mayoría de estudios han tratado de resolver problemas generales como por ejemplo el tipo de elementos que no deben ser utilizados en pantallas pequeñas tales como el proceso de la eliminación de combos y en su lugar el uso de checkbox, iconos o widgets para mejorar la experiencia del usuario. Además existen propuestas que incluyen técnicas de visualización que de cierta manera logran ampliar el espectro visual en pantalla, como la renderización geométrica para

Ambito	A	Academia	Fuente	A	ACM
	I	Industria		I	IEEE
Faceta de Investigación	O	Opinión	Tipo de contribución	S	SCOPUS
	Ex	Experiencia Personal		M	Modelo
	PS	Propuesta de Solución		T	Técnica
	V	Validación de Propuesta		P	Proceso
	E	Evaluación		H	Herramienta
Alcance de la Contribución	PM	Modelado	Documento	PP	Prototipo
	PA	Adaptabilidad		MP	Mapeo
	VV	Verificación/validación		F	Framework
	CD	Configuración/Despliegue		W	Workshop
	EP	Evaluación de Proceso		C	Conferencia
				A	Artículo
				R	Revista
				S	Simposio

aumentar, mejorar y optimizar las capacidades informáticas de dispositivos móviles aprovechando diversos enfoques viables, hardware y software y conservar energía. (viii). SOA: Una aplicación es estructurada como un conjunto de servicios orquestados por procesos de negocios, incluyendo los extraídos de varias aplicaciones existentes. (ix). Mashup: Aplicaciones de software que fusionan API y fuentes de datos separadas en una sola interfaz integrada y que aún requieren

TABLA IV
COMPARACIÓN DE ENFOQUES

Enfoque de Portabilidad	Evolutivo	Automático	Comp. Usuario	General	Interfaz	Usabilidad	Modelo Navegación	Tipo Migración
DOM	S	A	N	G	B	M	N	IU
Migraciones específicas	N	S	N	E	R	A	N	IU+Lógica
XML	S	S	N	G	B	M	N	IU
SOA	S	S	N	E	B	M	N	IU+Lógica
Multipantalla	S	A	S	G	R	M	N	IU
Modelos	N	S	N	E	B	M	N	IU
Mashups	S	A	S	E	R	M	N	IU
Traductores	N	A	N	E	B	M	N	IU
Programación Genética	S	A	N	E	B	M	N	IU
Aumentación	S	S	S	G	R	A	N	IU
Prototipos	S	S	N	E	B	B	N	IU
Widgets	N	A	N	E	R	A	N	IU
Arquitectura	S	S	S	E	B	M	N	IU+Lógica
Técnicas de Visualización	S	A	N	E	R	A	N	IU
Computación en la Nube	S	A	S	E	B	M	N	IU+Lógica
Algoritmos	S	A	N	E	B	M	N	IU

Automático	A	Automático		A Alta
	S	Semiautomático		
Evolutivo	S	SI	Usabilidad	B Baja
	N	NO		
Comportamiento Usuario	S	SI	Modelo Navegación	S Si
	N	NO		N No
General	G	General	Tipo Migración	IU Interfaz usuario
	E	Específico		IU+Lógica con Lógica negocio
Interfaz	R	Rica		
	B	Básica		

reducir proporcionalmente los tamaños de los elementos, entre otras, por lo que se puede manifestar que la portabilización ha sido un punto de arranque para utilizar técnicas o paradigmas de otras áreas no relacionadas directamente con el desarrollo de software como el diseño gráfico y la arquitectura. Otros incluyen mejoras a la tecnología actual como realidad aumentada. Además, prima la adaptación de aplicaciones web de escritorio hacia Html, CSS, JavaScript o JQuery, y hacia HTML5 con las correspondientes variantes en los tipos de archivos. Una presentación totalmente móvil usando widgets e íconos, su ventaja radica en la usabilidad. Proponen mejoras a tecnologías existentes como navegadores y modelos. Se plantean mejoras de rendimiento usando cloud computing para reducir las limitaciones de los dispositivos. Uso de estrategias de personalización en el cliente o adaptaciones del DOM. Además SOA, mashups, componentes web también siguen en vigencia. Sin embargo un diseño con la correcta visualización en el dispositivo móvil no soluciona todo el problema que conlleva la adaptación, se debe considerar factores tales como la conectividad, rapidez de respuesta, navegación, información sensible, tipo de controles soportados por la plataforma, puesto que si existe incompatibilidad de controles se afecta directamente a la eficiencia.

VII. LIMITACIONES

La principal limitación radica en obtener resultados dependientes de herramientas de búsqueda integradas al sistema de indexación SCOPUS y a las bibliotecas digitales exploradas: ACM, IEEE de acuerdo a reglas booleanas con los conceptos

principales determinados para la revisión. Además, no se consideraron los informes, tesis doctorales o resultados no publicados. Ni la literatura gris (también denominada como literatura no convencional, literatura semipublicada o literatura invisible) que proporciona Google Scholar.

VIII. CONCLUSIONES Y TRABAJO FUTURO

En los últimos años, las aplicaciones móviles han crecido exponencialmente en todo tipo de área a nivel mundial, actualmente forman parte de la vida empresarial y cotidiana, por esta razón la portabilidad de aplicaciones a dispositivos móviles ha sido ampliamente estudiado. Sin embargo no se han realizado revisiones sistemáticas sobre este tema, por ese motivo surge este estudio, que abarca esta portabilización en un periodo de 2006 a 2017: Inicialmente con 824 estudios obtenidos a través de la cadena de búsqueda ejecutada en bibliotecas digitales SCOPUS, IEEE y ACM, seleccionados de acuerdo a la evaluación y parámetros de calidad 37: 23 de SCOPUS, 6 de IEEE y 8 de ACM, provenientes de conferencias, simposios, workshops, artículos y revistas. Desarrollados en la academia, en la industria y trabajos conjuntos. A diferencia del mapeo sistemático, este trabajo permitió relevar en profundidad, de forma estructurada y reproducible la literatura dando como resultado un resumen exhaustivo de los trabajos. Los estudios considerados acorde a los criterios de inclusión evidencian diferentes estrategias para abordar esta problemática de portabilización de aplicaciones legacies web a móviles, en primer lugar se puede observar estrategias basadas en una migración o adaptación con prioridad en la interfaz del usuario. El enfoque predominante en este caso corresponde a la manipulación del DOM porque permite eliminar, modificar, añadir los controles que generen una interfaz inadecuada. Luego el desarrollo de software basado en modelos (MDD), en el cual las propuestas son parcialmente exitosas, ya que no pueden garantizar un proceso completamente automático (siempre se requiere un programador para codificar las modificaciones o detalles que no se han considerado o situaciones que aparecen). A continuación, se han utilizado configuraciones a través de XML. Otra estrategia corresponde a desarrollos que enfatizan el funcionamiento en multipantallas. Para casos específicos se ha utilizado un enfoque a través del uso de traductores, programación genética y algoritmos. Entre los enfoques contemporáneos que más se han abordado se encuentra la aumentación, los mashups que permiten darle cierta personalización al usuario en los dispositivos de pantallas pequeñas y además brindarle información del contexto. Los prototipos tradicionalmente han sido utilizados para obtener retroalimentación temprana, puesto que son desarrollos rápidos. Se han utilizado además otras técnicas de visualización y el uso de widgets para representar las funcionalidades. En segundo lugar existen trabajos que hacen este traspaso a dispositivos móviles con una migración más integral de las aplicaciones, puesto que además de trabajar en la interfaz deben proporcionar en este proceso un cambio en la lógica de negocio. Las migraciones específicas son un ejemplo de este tipo de enfoques. Las soluciones a través de los servicios SOA también deben integrar las dos capas. Otro enfoque interesante corresponde a la construcción de una

arquitectura que permita integrar tanto la interfaz de usuario como la lógica de negocio para obtener mejores resultados. La computación en la nube debe también reestructurar la aplicación para optimizar recursos tanto de hardware como de software. Sin embargo, un diseño con una correcta visualización en el dispositivo móvil no resuelve todo el problema que conlleva la adaptación a este tipo de aplicaciones. Estos documentos incluyen asesoramiento y propuestas para mejorar el proceso de adaptación. En ellos se han identificado una serie de problemas comunes experimentados por los desarrolladores en la adaptación tales como el tamaño de pantalla, duración de la batería, conectividad, plataforma (IOS y Android). De ellos surgen propuestas de solución como prototipos, modelos, herramientas y frameworks. Además de acuerdo a los resultados obtenidos se determina que aún no se ha logrado dar soluciones totales a esta problemática. Siendo así, se pueden extraer las siguientes conclusiones del estudio presentado: (i) Se requieren investigaciones en áreas como la obtención de interfaces ricas que contrarresten al enfoque de hacer reformateos con resultados básicos. (ii) Profundizar en herramientas que apliquen estrategias de visualización de grandes volúmenes de datos, porque no se enfoca el manejo de reportes. (iii) Aprovechar el hardware con los recursos intrínsecos de los dispositivos móviles como cámara, geolocalización, comandos táctiles y no táctiles, reconocimiento de voz, voz a texto, movilidad, utilización de interfaces hápticas, hologramas, accesibilidad, de acuerdo con el concepto de aplicaciones progresivas. (iv) No se evidencian estudios acerca de las metodologías para la portabilización o migración de aplicaciones desarrolladas en herramientas licenciadas. (v) Hacen falta directrices sobre estándares de tipos de letra y otros aspectos de visualización. (vi) También se requiere fortalecer la seguridad de aplicaciones móviles que manejen core empresarial con datos sensibles. (vii) Tampoco se evidencia la utilización de herramientas de análisis de sentimiento de usuarios. (viii) Ni la adaptabilidad de workflows empresariales a workflows móviles. (ix) Soluciones a través de modelos o cadenas ocultas de Markov [99][100] tampoco son propuestas. (x) Por otro lado se recomienda hacer un seguimiento a los trabajos que proponen pautas o directrices de accesibilidad [101], usabilidad y marcos conceptuales para determinar si efectivamente fueron utilizadas o se encuentran pendientes de implementar. (xi) Así como pensar en protocolos que puedan ser emergentes y logren conectividad sin Internet. Finalmente, como trabajo futuro se ha planificado proponer un enfoque de portabilización hacia aplicaciones progresivas con características de comportamiento de usuario [102].

REFERENCIAS

- [1] T. O'Reilly, "What is web 2.0?: design patterns and business models for the next generation of software. O'Reilly." no. 65, pp. 17–37, 2005.
- [2] W. J. Goralski, *ADSL and DSL technologies*. McGraw-Hill, Inc. New York, NY, USA ©2000, 2001.
- [3] G. B. Shelly and J. T. Campbell, *Microsoft Expression Web 3: Comprehensive*. Cengage Learning Inc., 2010.
- [4] B. N. Schilit, M. M. Theimer, and B. B. Welch, "Customizing Mobile Applications," *USENIX Symp osium on Mobile & Location-independent Computing*, pp. 1–9, 1993.
- [5] N. Mallat and V. K. Tuunainen, "Exploring Merchant Adoption of Mobile Payment Systems : An Empirical Study 1," *e-Service Journal*, vol. 6, no. 2, pp. 24–57, 2008.
- [6] P. Giner, C. Cetina, J. Fons, and V. Pelechano, "Developing Mobile Workflow Support in the Internet of Things." *IEEE Pervasive Computing*, pp. 18–26, 2010.
- [7] M. E. Ito, M. E., Okabe, D. E., & Matsuda, *Personal, portable, pedestrian: Mobile phones in Japanese life*. MIT press., 2005.
- [8] H. Sabat, "Emerging business models and trends in the mobile wireless industry," *International Journal of Information Technology and Management*, vol. 6, no. 2–4, pp. 299–328, 2007.
- [9] F. Wang, S. Li, Z. Dou, and D. Peng, "Markov Modeling Methods for Performance Analysis of IEEE 802 . 11 Protocol," in *2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, no. 1, 2018, pp. 1–5.
- [10] G. Salvatierra, C. Mateos, M. Crasso, A. Zunino, and M. Campo, "Legacy System Migration Approaches," *IEEE Latin America Transactions*, vol. 11, no. 2, pp. 840–851, 2013.
- [11] B. Kitchenham, O. Pearl Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, "Systematic literature reviews in software engineering - A systematic literature review," *Information and Software Technology*, vol. 51, no. 1, pp. 7–15, 2009.
- [12] Google, "Android," 2019. [Online]. Available: <https://www.android.com/>
- [13] TechTarget, "Apple IOS," 2019. [Online]. Available: <https://searchmobilecomputing.techtarget.com/definition/iOS>
- [14] R. Beauregard and P. Corriveau, "User Experience Quality : A Conceptual Framework for Goal Setting and Measurement," pp. 325–332, 2007.
- [15] D. Zhang and J. Lai, "Can convenience and effectiveness converge in mobile web? a critique of the state-of-the-art adaptation techniques for web navigation on mobile handheld devices," 2011.
- [16] Google, "Mobile-Friendly Test - Google Search Console," 2018. [Online]. Available: <https://search.google.com/test/mobile-friendly>
- [17] AmazonAlexa, "The top 500 sites on the web," 2017. [Online]. Available: <https://www.alexa.com/topsites>
- [18] E. Marcotte, "Responsive-web-design," 2010.
- [19] S. Weinzettelova, "Traditional Type in the Digital Era," *Bulletin-Prague College Centre for Research and Interdisciplinary Studies.*, pp. 5–24, 2012. [Online]. Available: <http://invenio.nucl.cz/record/124108>
- [20] J. Duckett, *HTML and CSS: design and build websites*. Indianapolis: John Wiley & Sons., 2011.
- [21] D. Zimmerman and T. Yohon, "Small-screen interface design: Where are we? Where do we go?" in *IEEE International Professional Communication Conference*, 2009.
- [22] J. Nielsen, "Mobile Site vs. Full Site," 2012. [Online]. Available: <https://www.nngroup.com/articles/mobile-site-vs-full-site/>
- [23] M. Urbietta, M. Escalona, M. Luna, and G. Rossi, "Detecting Conflicts and Inconsistencies in Web Application Requirements," in *Current Trends in Web Engineering, Workshops, Doctoral Symposium, and Tutorials Held at ICWE 2011 Paphos, Cyprus, June 20-21, 2011 Revised Selected Papers*, 2011.
- [24] B. Kitchenham and S. Charters, "Guidelines for performing Systematic Literature reviews in Software Engineering Version 2.3," *Engineering*, vol. 45, no. 4ve, p. 1051, 2007.
- [25] V. Cajas, M. Urbietta, Y. Rybarczyk, G. Rossi, and C. Guevara, "Portability Approaches for Business Web Applications to Mobile Devices: A Systematic Mapping," 2018.
- [26] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic mapping studies in software engineering," *EASE'08 Proceedings of the 12th international conference on Evaluation and Assessment in Software Engineering*, pp. 68–77, 2008.
- [27] P. Brereton, B. A. Kitchenham, D. Budgen, M. Turner, and M. Khalil, "Lessons from applying the systematic literature review process within the software engineering domain," *Journal of Systems and Software*, vol. 80, no. 4, pp. 571–583, 2007.
- [28] M. Petticrew and H. Roberts, *Systematic reviews in the social sciences: A practical guide*. John Wiley & Sons, 2008.
- [29] J. J. Randolph, "A Guide to Writing the Dissertation Literature Review in Software Engineering," *Practical Assessment, Research & Evaluation*, vol. 14, no. 13, pp. 1–13, 2009.
- [30] D. Budgen and P. Brereton, "Performing systematic literature reviews in software engineering," *Proceeding of the 28th international conference on Software engineering - ICSE '06*, p. 1051, 2006.
- [31] L. Chen, M. A. Babar, and H. Zhang, "Towards Evidence-Based Understanding of Electronic Data Sources," *EASE'10 Proceedings of the 14th international conference on Evaluation and Assessment in Software Engineering*, pp. 135–138.

- [32] R. Wieringa, N. Maiden, N. Mead, and C. Rolland, "Requirements engineering paper classification and evaluation criteria: A proposal and a discussion," *Requirements Engineering*, vol. 11, no. 1, pp. 102–107, 2006.
- [33] T. Kosar, S. Bohra, and M. Mernik, "Domain-Specific Languages: A Systematic Mapping Study," *Information and Software Technology*, vol. 71, pp. 77–91, 2016.
- [34] T. Mey Eap, D. Gašević, and F. Lin, "Personalised mobile learning content delivery: a learner centric approach," *Int. J. Mobile Learning and Organisation J. Mobile Learning and Organisation*, vol. 3, no. 1, pp. 84–101, 2009.
- [35] T. Lehtonen, S. Benamar, V. Laamanen, I. Luoma, O. Ruotsalainen, J. Salonen, and T. Mikkonen, "Towards User-Friendly Mobile Browsing," in *Proceeding AAA-IDEA '06 Proceedings of the 2nd international workshop on Advanced architectures and algorithms for internet delivery and applications Article No. 6*, 2006.
- [36] OperaMini, "Opera Mini Browser," 2018. [Online]. Available: <http://www.opera.com/products/mobile/operamini/>
- [37] OperaMobile, "Opera Mobile Browser," 2018. [Online]. Available: <http://www.opera.com/products/mobile/products/>
- [38] G. D. Santo and E. Zimeo, "Reversing GUIs to XIML Descriptions for the Adaptation to Heterogeneous Devices," in *Proceedings of the 2007 ACM symposium on Applied computing Pages 1456-1460*, 2007.
- [39] D. S. C. G. Giuseppe, "TCPTe. Thin-Client Application For Limited Devices," 2004. [Online]. Available: <http://tcpte.rcost.unisannio.it/>
- [40] M. C. Cheng and S. M. Yuan, "An adaptive and unified mobile application development framework for java," in *Journal of Information Science and Engineering*, 2007.
- [41] J. He, T. Gao, W. Hao, I.-L. Yen, and F. Bastani, "A Flexible Content Adaptation System Using a Rule-Based Approach," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 1, pp. 127–140, 2007.
- [42] A. Ennai and S. Bose, "MobileSOA: A service oriented web 2.0 framework for context-aware, lightweight and flexible mobile applications," in *Proceedings - IEEE International Enterprise Distributed Object Computing Workshop, EDOC*, 2008.
- [43] H. Ahmadi and J. Kong, "Efficient web browsing on small screens," in *Proceedings of the working conference on Advanced visual interfaces ACM pp 23-30*, 2008.
- [44] R. Song, H. Liu, J.-R. Wen, and W.-Y. Ma, "Learning Block Importance Models for Web Pages," in *Proceedings of the 13th international conference on World Wide Web pp 203-211*, 2004.
- [45] S. Kopf, B. Guthier, H. Lemelson, and W. Effelsberg, "Adaptation of Web Pages and Images for Mobile Applications," in *Proceedings Volume 7256, Multimedia on Mobile Devices 2009; 72560C*, 2009.
- [46] Y. Xiao, Y. Tao, and Q. Li, "A New Wireless Web Access Mode Based on Cloud Computing," 2008.
- [47] F. Paterno, C. Santoro, and L. D. Spano, "MARIA: A Universal, Declarative, Multiple Abstraction-Level Language for Service-Oriented Applications in Ubiquitous Environments," *Comput.-Hum. Interact. Article*, vol. 16, no. 19, 2009.
- [48] F. Paternò and G. Zichittella, "Desktop-to-mobile web adaptation through customizable two-dimensional semantic redesign," in *International Conference on Human-Centred Software Engineering pp 79-94*. Springer, 2010.
- [49] S. Erick, "Mowser," 2008. [Online]. Available: <https://techcrunch.com/tag/mowser/>
- [50] Skweezer, "skweezer," 2018. [Online]. Available: <https://skweezer.net/>
- [51] R. Armenise, C. Birtolo, and L. Troiano, "A tool for automatic adaptation of web pages to different screen size," in *ICEIS 2010 - Proceedings of the 12th International Conference on Enterprise Information Systems*, vol. 5 HCI, 2010, pp. 91–98.
- [52] S. K. Guirguis and M. A. Hassan, "A Smart Framework for Web Content and Resources Adaptation in Mobile Devices," in *2010 The 12th International Conference on Advanced Communication Technology (ICACT)*, 2009.
- [53] Q.-C. Li, Z.-Y. Zhang, J. Ma, and J. Zhang, "Web Page Layout Adaptation Based on WebKit for E-paper Device," in *2011 14th IEEE International Conference on Computational Science and Engineering*, 2011.
- [54] A. Koehl and H. Wang, "M.Site: Efficient content adaptation for mobile devices - Middleware 2012," in *Proceeding Middleware '12 Proceedings of the 13th International Middleware Conference*, pp. 41–60, 2012.
- [55] K. Rajkumar and V. Kalavani, "Dynamic web page segmentation based on detecting reappearance and layout of tag patterns for small screen devices," in *International Conference on Recent Trends in Information Technology, ICRTIT 2012*, 2012.
- [56] C. Challiol, S. Firmenich, G. A. Bosetti, S. E. Gordillo, and G. Rossi, "Crowdsourcing Mobile Web Applications," in *International Conference on Web Engineering ICWE 2013: Current Trends in Web Engineering*, 2013, pp. 223–237.
- [57] N. Shaari, S. Charters, and C. Churcher, "International Journal of Web Information Systems Achieving "One-Web" through customization and prioritization," *International Journal of Web Information Systems*, vol. 9, no. 3, pp. 279–316, 2013.
- [58] F. Amendola and L. Favre, "Adapting CRM systems for mobile platforms: An MDA perspective," in *SNPD 2013 - 14th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, 2013.
- [59] A. Albasir, K. Naik, and T. Abdunabi, "Smart Mobile Web Browsing," in *2013 International Joint Conference on Awareness Science and Technology & Ubi-Media Computing (iCAST 2013 & UMEDIA 2013)*, 2013.
- [60] M. Yun, Y. Fang, X. Zhu, X. Liu, and G. Huang, "MobiTran: Tool Support for Refactoring PC Websites to Smart Phones," in *Proceeding MiddlewareDPT '13 Proceedings Demo & Poster Track of ACM/IFIP/USENIX International Middleware Conference Article No. 6*, 2013.
- [61] S. Pandey, "Responsive design for transaction banking - a responsible approach," in *Proceeding APCHI '13 Proceedings of the 11th Asia Pacific Conference on Computer Human Interaction*, 2013, pp. 291–295.
- [62] Toile H, "Adaptation of composite E-Learning contents for reusable in smartphone based learning system," in *2014 International Conference on Advanced Computer Science and Information System*, 2014.
- [63] S. K. Badam and N. Elmqvist, "Polychrome: A cross-device framework for collaborative web visualization," *ITS 2014 - Proceedings of the 2014 ACM International Conference on Interactive Tabletops and Surfaces*, pp. 109–118, 2014.
- [64] J. Yang and D. Wigdor, "Panelrama: Enabling easy specification of cross-device web applications," in *Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI '14*, pp. 2783-2792, 2014.
- [65] D. Kovachev, D. Renzel, P. Nicolaescu, and R. Klamma, "DireWolf - Distributing and migrating user interfaces for widget-based web applications," in *Lecture Notes in Computer Science*, 2013.
- [66] P. Xiang, X. Yang, and Y. Shi, "Effective page segmentation combining pattern analysis and visual separators for browsing on small screens," in *Proceedings - 2006 IEEE/WIC/ACM International Conference on Web Intelligence, WI'06*, 2007.
- [67] D. Cai, S. Yu, J.-R. Wen, and W.-Y. Ma, "VIPS: a Vision-based Page Segmentation Algorithm," 2013.
- [68] T.-L. Tseng, S.-H. Hung, and C.-H. Tu, "Migratom.js: A JavaScript Migration Framework for Distributed Web Computing and Mobile Devices," in *Proceeding SAC '15 Proceedings of the 30th Annual ACM Symposium on Applied Computing*, pp. 798-801, 2015.
- [69] M. Sarkis, C. Concolato, and J.-C. Dufourd, "MSoS: A Multi-Screen-Oriented Web Page Segmentation Approach," in *Proceeding DocEng '15 Proceedings of the 2015 ACM Symposium on Document Engineering*, pp. 85-88, 2015.
- [70] J. Chen, B. Zhou, J. Shi, and H. Zhang, "Function-Based Object Model Towards Website Adaptation," in *Proceeding WWW '01 Proceedings of the 10th international conference on World Wide Web*, pp. 587-596, 2001.
- [71] A. Sanoja and S. Gancarski, "Block-o-Matic: A web page segmentation framework," in *International Conference on Multimedia Computing and Systems -Proceedings*, 2014.
- [72] S. Wang, W. Dou, G. Wu, J. Wang, C. Gao, J. Wei, and T. Huang, "Towards Web Application Mobilization via Efficient Web Control Extraction," in *Proceeding Internetware '15 Proceedings of the 7th Asia-Pacific Symposium on Internetware*, pp. 21-29, 2015.
- [73] G. Li, E. Andreasen, and I. Ghosh, "SymJS: automatic symbolic testing of JavaScript web applications," in *Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering - FSE 2014*, 2014.
- [74] S. Bouzit, D. Chêne, and G. Calvary, "Evanescent Adaptation on Small Screens," in *Proceeding OzCHI '15 Proceedings of the Annual Meeting of the Australian Special Interest Group for Computer Human Interaction*, pp. 62-68, 2015.
- [75] S. Bouzit, D. E. N. I. S. Chêne, and G. Calvary, "From Appearing to Disappearing Ephemeral Adaptation for Small Screens," in *Proceeding OzCHI '14 Proceedings of the 26th Australian Computer-Human*

- Interaction Conference on Designing Futures: the Future of Design*, pp. 41-48, 2014.
- [76] J. L. Miján, I. Garrigós, and S. Firmenich, "Supporting personalization in legacy web sites through client-side adaptation," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9671, pp. 588-592, 2016.
- [77] G. Huang, X. Liu, X. Lu, Y. Ma, Y. Zhang, and Y. Xiong, "Programming Situational Mobile Web Applications with Cloud-Mobile Convergence: An Internetware-Oriented Approach," 2015.
- [78] S. Balasubramaniam, G. A. Lewis, S. Simanta, and D. B. Smith, "Situating software: Concepts, motivation, technology, and the future," *IEEE Software*, vol. 25, no. 6, 2008.
- [79] Docker Inc., "Docker," 2008. [Online]. Available: <https://www.docker.com/>
- [80] X. Liu, M. Xu, T. Teng, G. Huang, and H. M. Fellow, "MUIT: A Middleware for Adaptive Mobile Web-based User Interfaces in WS-BPEL," *arXiv preprint arXiv:1602.09125*, 2015.
- [81] H. Li, M. Hu, X. Du, and X. Zhu, "Extracting Main Content of Webpage to Enhance Adaptively Rendering for Small Screen Size Terminals," in *Proceedings - 2015 International Conference of Educational Innovation Through Technology, EITT 2015*, 2016.
- [82] G. A. Bosetti, S. Firmenich, S. E. Gordillo, G. Rossi, G.-J. Houben, and M. Bielikova, "An approach for building mobile web applications through web augmentation," *Journal of Web Engineering*, vol. 16, no. 2, pp. 75-102, 2017.
- [83] M. Sarkis, C. Concolato, and J. C. Dufourd, "A multi-screen refactoring system for video-centric web applications," *Multimedia Tools and Applications*, pp. 1-28, 2017.
- [84] L. Chen, M. A. Babar, and H. Zhang, "Towards Evidence-Based Understanding of Electronic Data Sources," *EASE'10 Proceedings of the 14th international conference on Evaluation and Assessment in Software Engineering*, pp. 135-138, 2010.
- [85] R. Nicholas, *Beyond jQuery*. Switzerland: Springer Link, 2016.
- [86] M. Technet, "Virtualización: Migración del software," 2013. [Online]. Available: <https://technet.microsoft.com/es-es/library/dn283989.aspx>
- [87] W3c, "XML Path Language (XPath)," *Engineering*, no. January, pp. 297-318, 2010.
- [88] E. S. Sánchez, P. J. Clemente, A. E. Prieto, J. M. Conejero, and R. R. Echeverría, "MigraSOA : Migration of Legacy Web Applications to Service Oriented Architectures (SOA)," *IEEE Latin America Transactions*, vol. 15, no. 7, pp. 1306-1311, 2017.
- [89] L. Bally, J. Brittan, and K. H. Wagner, "A prototype approach to information system design and development," *Information & Management*, vol. 1, no. 1, pp. 21-26, 1977.
- [90] C. Pons, R. Giandini, and G. Pérez, *Desarrollo de Software Dirigido Por Modelos*. Editorial de la Universidad Nacional de La Plata, 2010.
- [91] N. Zang, M. B. Rosson, and V. Nasser, "Mashups," *Proceeding of the twenty-sixth annual CHI conference extended abstracts on Human factors in computing systems - CHI '08*, p. 3171, 2008.
- [92] N. Kumar, J. Misurda, and B. R. Childers, "Instrumentation in Software Dynamic Translators for Self-Managed Systems Department of Computer Science University of Pittsburgh Mary Lou Soffa University of Virginia," in *Proceeding WOSS '04 Proceedings of the 1st ACM SIGSOFT workshop on Self-managed systems*, pp. 90-94, 2004.
- [93] U. Bodenhofer, "Genetic Algorithms : Theory and Applications by," 2002.
- [94] S. Abolfazli, Z. Sanaei, E. Ahmed, A. Gani, and R. Buyya, "Cloud-based augmentation for mobile devices: Motivation, taxonomies, and open challenges," *IEEE Communications Surveys and Tutorials*, vol. 16, no. 1, pp. 337-368, 2014.
- [95] U. Anind K. Dey College of Computing & Gvu Center, Georgia Institute of Technology, Atlanta, GA, "Understanding and Using Context," *Journal Personal and Ubiquitous Computing archive*, vol. 5, no. 1, pp. 4-7, 2001.
- [96] P. Kruchten, H. Obbink, and J. Stafford, "The Past, Present, and Future for Software Architecture," *IEEE Software*, vol. 23, no. 2, pp. 22-30, 2006.
- [97] P. Mell and T. Grance, "The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology," *National Institute of Standards and Technology, Information Technology Laboratory*, vol. 145, p. 7, 2011.
- [98] Y. Gurevich, "What is an algorithm?" in *Mathematics Unlimited — 2001 and Beyond pp 919-936*, no. July, 2001.
- [99] M. Horný, "Bayesian networks: A Technical report," no. 5, p. 15, 2014.
- [100] J. K. Deters and Y. Rybarczyk, "Hidden Markov model approach for the assessment of tele-rehabilitation exercises," *International Journal of Artificial Intelligence*, vol. 16, no. 1, pp. 1-19, 2018.
- [101] C. Siebra, T. B. Gouveia, J. Macedo, F. Q. B. Da Silva, A. L. M. Santos, W. Correia, M. Penha, M. Anjos, and F. Florentin, "Toward Accessibility with Usability: Understanding the Requirements of Impaired Uses in the Mobile Context," 2017.
- [102] V. Cajas, M. Urbietta, Y. Rybarczyk, and G. Rossi, "An Approach for Migrating Legacy Applications to Mobile Interfaces," in *New Knowledge in Information Systems and Technologies. WorldCIST'19 2019. Advances in Intelligent Systems and Computing.*, C. S. Rocha A., Adeli H., Reis L., Ed., vol. 930. La Toja: Springer, Cham, 2019.



Viviana Elizabeth Cajas Se graduó de Ingeniería en Sistemas e Informática en el año 2004 en la ESPE (Ecuador), desde el 2012 es Magíster en Administración de Empresas y Marketing de la Universidad Tecnológica Indoamérica (UTI) de Ecuador, es estudiante del Doctorado en Ciencias Informáticas en la Universidad Nacional de La Plata (Argentina). Es autora de algunos capítulos de libros y artículos de conferencias. Sus intereses comprenden las aplicaciones legacy y su portabilización a móviles desde el punto de vista del usuario, software ecológico, proactivo y evolutivo. Además el análisis del comportamiento del consumidor con base generacional. Actualmente es Docente de la UTI.



Mario Matias Urbietta nació en Formosa, Argentina, en 1981. Se recibió en B.S. y Ph.D en Ciencias Informáticas de la Universidad Nacional de La Plata (UNLP). Desde 2008, es asistente de investigación en el Laboratorio de Investigación y Formación en Informática Avanzada (LIFIA) en la UNLP. Es autor de 1 capítulo de libro, 10 artículos de revistas y 20 artículos de conferencia. Sus intereses investigativos incluyen la separación de inquietudes en aplicaciones Web, ingeniería de requisitos Agile y aumento Web. Enseña conceptos de programación orientada a objetos e ingeniería Web a nivel de licenciatura y doctorado en UNLP y Universidad Abierta Interamericana.



Gustavo Rossi recibió su doctorado en PUC-Rio, Brasil en 1996, cuya tesis fue el desarrollo del enfoque de diseño OOHDm, uno de los métodos maduros para el desarrollo de aplicaciones Web. Él es actualmente profesor titular en la Facultad de Informática, Universidad Nacional de La Plata. Ha sido profesor visitante en las Universidades de Lyon y Montpellier en Francia, recibió el Habilitation pour Diriger Recherches (HDR) en INSA-Lyon. Ha estado como parte del comité de PC de las conferencias más importantes de su campo de investigación como ACM, WWW, ICWE y ACM hypertexto. Miembro del consejo de redacción de IEEE, Internet Computing, IEEE IT Professional, WWW Journal, Journal of Web Ingeniería, Revista de Servicios y Aplicaciones de Internet y la Revista Internacional de Sistemas de Información Cooperativa.



Francisco José Domínguez Mayo recibió su PhD (con mención internacional) en ingeniería y tecnología de software en la Universidad de Sevilla en 2013. Es investigador e imparte asignaturas de teoría y práctica relacionadas con la ingeniería de software en el Departamento de Lenguajes y Sistemas Informáticos (LSI) de la misma universidad. Su área de investigación principal es la calidad del software dentro de la ingeniería Web y las metodologías. En los últimos años ha colaborado con universidades nacionales e internacionales realizando más de 40 publicaciones en revistas, conferencias y capítulos de libros. Revisor de revistas indexadas en el "Journal Citation Report", organizador de congresos, y pertenece al comité del programa de varios de ellos.