

# ESTADO DEL ARTE: METODOLOGÍAS DE DESARROLLO DE APLICACIONES MÓVILES

## STATE OF ART: MOBILE SOFTWARE DEVELOPMENT METHODOLOGIES

---

### **Jimmy Rolando Molina Ríos**

Grupo de Investigación de Ingeniería de Sistemas, Universidad Técnica de Machala, (Ecuador).

E-mail: [j.molina1@udc.es](mailto:j.molina1@udc.es) ORCID: <https://orcid.org/0000-0003-3915-8270>

### **Joofre Antonio Honores Tapia**

Grupo de Investigación de Ingeniería de Sistemas, Universidad Técnica de Machala, (Ecuador).

E-mail: [jhonores@utmachala.edu.ec](mailto:jhonores@utmachala.edu.ec) ORCID: <https://orcid.org/0000-0001-8612-3025>

### **Nieves Pedreira-Souto**

Departamento de Ciencias da Computación e Tecnoloxías da Información. Facultad de Informática.

Universidade da Coruña, (España).

E-mail: [nieves.pedreira@udc.es](mailto:nieves.pedreira@udc.es) ORCID: <https://orcid.org/0000-0002-8122-0886>

### **Henry Paúl Pardo León**

Grupo de Investigación de Ingeniería de Sistemas, Universidad Técnica de Machala, (Ecuador).

E-mail: [hpardo1@utmachala.edu.ec](mailto:hpardo1@utmachala.edu.ec) ORCID: <https://orcid.org/0000-0002-3785-1402>

**Recepción:** 22/11/2020 **Aceptación:** 09/03/2021 **Publicación:** 14/06/2021

#### **Citación sugerida:**

Molina, J. R., Honores, J. A., Pedreira-Souto, N., y Pardo, H. P. (2021). Estado del arte: metodologías de desarrollo de aplicaciones móviles. *3C Tecnología. Glosas de innovación aplicadas a la pyme*, 10(2), 17-45. <https://doi.org/10.17993/3ctecno/2021.v10n2e38.17-45>

## RESUMEN

Desde el origen de la telefonía celular, el desarrollo de aplicaciones móviles ha crecido de manera exponencial abriendo un nuevo campo de investigación en la rama de la ingeniería de software. La necesidad de producir una aplicación con altos índices de calidad ha llevado al estudio y formulación de nuevas metodologías que abarquen todos los aspectos correspondientes y relacionados a la calidad y tiempo de producción. El objetivo de la presente investigación es realizar un estudio exhaustivo para extraer datos relevantes de cada marco de trabajo a través de la investigación bibliográfica y de campo para la construcción de un estado del arte que compruebe a través de un análisis la metodología indicada para el desarrollo de aplicaciones móviles. El resultado obtenido del análisis muestra que, a pesar de encontrar cierta similitud entre las metodologías con principios ágiles, Scrum es caracterizada como la metodología indicada para el desarrollo de aplicaciones móviles. Este hecho dio lugar al surgimiento de nuevas metodologías de desarrollo de software con enfoques a las denominadas prácticas ágiles cuyo objetivo es la producción de software de calidad.

## PALABRAS CLAVE

Aplicaciones Móviles, Desarrollo Móvil, Metodologías Ágiles, Ingeniería De Software, Soluciones De Software, Telefonía Celular.

## ABSTRACT

*Since the origin of cell phones, the development of mobile applications has grown exponentially, opening a new field of research in the field of software engineering. The need to produce an application with high quality indices has led to the study and formulation of new methodologies that cover all the corresponding aspects related to quality and production time. The objective of this research is to carry out an exhaustive study to extract relevant data from each framework through bibliographic and field research for the construction of a state of the art that verifies through an analysis the methodology indicated for the development of mobile applications. The result obtained from the analysis shows that, despite finding a certain similarity between the methodologies with agile principles, Scrum is characterized as the indicated methodology for the development of mobile applications. This fact gave rise to the emergence of new software development methodologies with approaches to the so-called agile practices whose objective is the production of quality software.*

## KEYWORDS

*Mobile Applications, Mobile Development, Agile Methodologies, Software Engineering, Software Solutions, Cellular Telephony.*

# 1. INTRODUCCIÓN

El desarrollo de aplicaciones móviles tiene como origen a final de los 90 durante la generación de teléfonos celulares, donde tenían funciones básicas como: almacenar contactos, gestión de perfiles, configuración de ringtones y otras más complejas que permitían la gestión de llamadas telefónicas.

Con el paso del tiempo, la transformación tecnológica en el área de la telefonía celular dio lugar a un nuevo mercado en el desarrollo de software. A finales del año 2007 y comienzos del 2008, surgieron las primeras tiendas de aplicaciones móviles y para Julio del 2015, de acuerdo con (Hassan *et al.*, 2017), existen más de 3.9 millones de aplicaciones publicadas en las tiendas de aplicaciones. Esto ha ocasionado una gran conmoción en el área de desarrollo de software, estimulando un crecimiento exponencial en la demanda de aplicaciones para distintas clases de dispositivos.

Al igual que la producción de aplicaciones web o de escritorio, las móviles ameritan de metodologías de desarrollo para llevar a cabo un producto con altos índices de calidad. En el presente, las aplicaciones móviles son desarrolladas en gran parte con metodologías ágiles donde posee características enfocadas al desarrollo iterativo, flexibilidad y pruebas (Batarseh y Gonzalez, 2018).

En la comparación que se realizó entre las metodologías ágiles dentro del marco del desarrollo del estado del arte tuvo como resultado la identificación de las características de Scrum presentes en otros marcos de trabajo utilizados en investigaciones similares enfocados al desarrollo de aplicaciones móviles.

## 2. ANTECEDENTES O ESTADO DEL ARTE

### 2.1. APLICACIONES MÓVILES

**Tabla 1.** Aplicaciones móviles.

TÍTULO	CONTENIDO	AÑO
Software assurance practices for mobile applications	“Los sistemas móviles llevan a cabo sus operaciones en una amplia gama de entornos, desde servicios tradicionales como comunicación de voz, mensajería de texto y juegos, hasta aplicaciones comerciales, servicios basados en la ubicación, realidad aumentada, herramientas de productividad y mucho más.” (Corral <i>et al.</i> , 2015)	2015
Choosing between responsive-design websites versus mobile apps for your mobile behavioral intervention: presenting four case studies	“Las aplicaciones móviles son programas que pueden realizar determinadas tareas o proporcionar determinadas funciones para un usuario” (Turner-McGrievy <i>et al.</i> , 2017)	2017
Analyzing and automatically labelling the types of user issues that are raised in mobile app reviews	“El mercado de aplicaciones móviles sigue creciendo a un ritmo muy rápido con miles de desarrolladores, miles de aplicaciones y millones de dólares en ingresos.” (McIlroy <i>et al.</i> , 2016)	2016

**Fuente:** elaboración propia.

De acuerdo con la síntesis de la Tabla 1, la demanda de software móviles se encuentra en un constante progreso a niveles exponenciales durante un breve periodo de tiempo. En el presente, existe un abrumador número de desarrolladores al igual que aplicaciones publicadas en las principales App Store como Google Play, iOS, Windows Phone, etc.

Las aplicaciones o software móvil desempeñan un sin número de funcionalidades y servicios disponibles para los usuarios o clientes que hacen uso recurrente o esporádico de uno o varios dispositivos móviles como los smartphones, tablets, Smart-tv u otros.

## 2.2. METODOLOGÍAS ENFOCADAS AL DESARROLLO MÓVIL

**Tabla 2.** Metodologías enfocadas al desarrollo de aplicaciones móviles.

TÍTULO	CONTENIDO	AÑO
Job-work fit as a determinant of the acceptance of large-scale agile methodology.	“Una metodología de desarrollo de sistemas o software (SDM) se puede definir como una colección documentada de políticas, procesos y procedimientos para mejorar el proceso de desarrollo de software con respecto al aumento de la productividad del personal de tecnología de la información (TI) y una mayor calidad de las soluciones de TI finales.” (Batra, 2020)	2020
A Comparative Analysis on Effort Estimation for Agile and Non-agile Software Projects Using DBN-ALO.	“Estas metodologías ágiles tienen ventajas como la flexibilidad, responden a los cambios incluso en etapas posteriores de desarrollo, son amigables para el cliente y entregan el software en funcionamiento rápidamente.” (Kaushik <i>et al.</i> , 2020)	2020
Using agile methodologies for adopting COBIT	“Al elegir una metodología ágil frente a una tradicional, los autores buscan facilitar la implementación, apoyándose en prácticas ágiles como habilitadores para lograr los objetivos propuestos. El principio ágil de ofrecer soluciones a menudo aumentará el apoyo de la alta dirección y disminuirá la resistencia al cambio.” (Amorim <i>et al.</i> , 2020)	2020

**Fuente:** elaboración propia.

En proporción a lo analizado en la Tabla 2, en el desarrollo de software móvil, al igual que en el desarrollo web o de escritorio, emplea metodologías para el seguimiento de procesos y procedimientos de acuerdo a políticas o directrices que son establecidas con el objetivo de optimizar recursos y garantizar que los procesos se hayan cumplido eficientemente para la obtención de un software o aplicación con calidad.

Por lo general, para el desarrollo de aplicaciones móviles, se opta por elegir metodologías ágiles en virtud de las características que ofrecen para adaptarse a nuevos contextos que emergen a lo largo del proyecto. Aquellas metodologías enfocadas al desarrollo de aplicaciones móviles, parten de la orientación del

desarrollo ágil, que, a diferencia del tradicional, presenta una mayor flexibilidad al cambio y el tiempo de entrega o producción es más corto.

## 2.3. MOBILE-D

**Tabla 3.** Metodología Mobile-D.

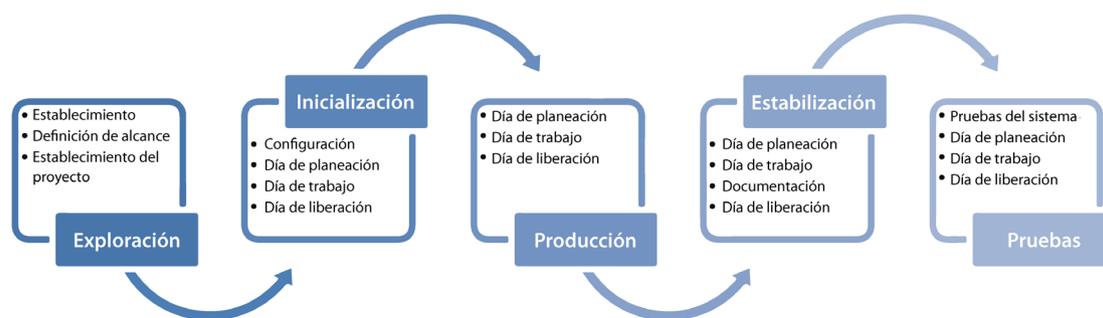
TÍTULO	CONTENIDO	AÑO
A Software Development Process for Super Agile Projects	“Compatible con equipos de menos de diez personas, trabajando en un mismo espacio y con el objetivo de entregar una aplicación completamente funcional en menos de diez semanas. Este enfoque se dividió en cinco iteraciones, y cada iteración tenía días para planificar, trabajar y entregar.” (Cerqueira <i>et al.</i> , 2018)	2018
Software engineering process models for mobile app development: A systematic literature review	“Mobile-D también se basa en los mismos métodos, incluidas las metodologías Crystal.” (Jabangwe <i>et al.</i> , 2018)	2018
An Agile-Based Integrated Framework for Mobile Application Development Considering Ilities	“presentan un enfoque de desarrollo ágil denominado Mobile-D, que considera las particularidades del entorno móvil tal como existían en 2004, como la limitada capacidad de memoria, variedad de estándares, variedad de plataformas y características especiales de los terminales móviles.” (Martinez <i>et al.</i> , 2020)	2020

**Fuente:** elaboración propia.

El análisis ejecutado en la Tabla 3, puntualiza las características de la metodología Mobile-D y su enfoque hacia los métodos o prácticas ágiles para la producción de aplicaciones móviles. En la Ilustración 1, explica el proceso de Mobile-D durante el ciclo de vida del software móvil; se divide cinco fases cada una

cuenta con subprocesos iterativos. Cabe resaltar que Mobile-D se fundamenta en los mismos principios de otras metodologías ágiles como las de Crystal.

El objetivo de Mobile-D es desarrollar aplicaciones móviles en grupos de trabajo reducidos a diez personas o menos, cada uno enfocado a un área específica del proceso. A partir de aquel planteamiento, el tiempo de entrega del producto completamente funcional estaría disponible dentro de diez semanas teóricamente.



**Ilustración 1.** Proceso de Mobile-D.

**Fuente:** elaboración propia.

## 2.4. TEST-DRIVEN DEVELOPMENT (TDD)

**Tabla 4.** Metodología Test-Driven Development.

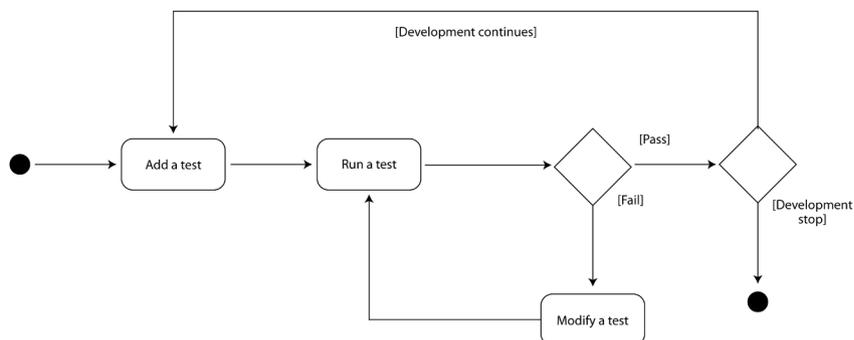
TÍTULO	CONTENIDO	AÑO
An Empirical Study on the Ability Relationships Between Programming and Testing	“Requiere que los desarrolladores escriban el código de prueba antes de escribir el código de producción para una determinada función. Entonces, sólo se escribe el código de función que pasa la prueba. Por lo tanto, los casos de prueba se utilizan para impulsar todo el desarrollo. Esto ayuda a los desarrolladores a escribir código conciso y de alta calidad y acelera el proceso de desarrollo.” (Yang <i>et al.</i> , 2020)	2020

An industry experiment on the effects of test-driven development on external quality and productivity	“TDD obliga a los desarrolladores a pensar en bits más pequeños, es decir, en términos de las subtareas de la funcionalidad, que se pueden probar, y a rediseñar el código después de implementar cada subtarea.” (Tosun <i>et al.</i> , 2017)	2017
---	--	------

**Fuente:** elaboración propia.

En el análisis de la Tabla 4, se obtienen los fundamentos en el cual se sustenta la metodología Test-Driven Development, los cuales consisten en realizar pequeños fragmentos de código para cada función o modulo a desarrollar con el objetivo de promover el desarrollo de códigos más limpios y de índices de calidad óptimos.

En la Ilustración 2, clarifica los procesos a ejecutarse durante la fase de producción de un código. Empieza con la entrada de un nuevo fragmento de prueba, el cuál pasa por la fase de testeo donde se define si continua con la pre producción en caso de no presentar errores de lo contrario se devuelve a la fase de modificar y se vuelve a correr la misma prueba. Ya en la etapa de producción o desarrollo se verifica si se ha realizado la refactorización o limpieza del código y termina.



**Ilustración 2.** Proceso de TDD.

**Fuente:** (Kerthyayana Manuaba, 2019)

## 2.5. EXTREME PROGRAMMING (XP)

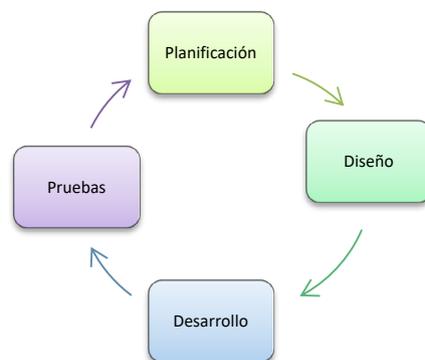
**Tabla 5.** Metodología Extreme Programming.

TÍTULO	CONTENIDO	AÑO
To what extent the medical device software regulations can be achieved with agile software development methods? XP—DSDM—Scrum	“XP fue desarrollado por Kent Beck en 1999 y proporciona una colección de prácticas de ingeniería de software. Aunque las prácticas no son novedosas, XP las reúne para facilitar el cambio y producir software de mayor calidad a un ritmo sostenible.” (Özcan-Top y McCaffery, 2019)	2019
Integrating design thinking into extreme programming	“El lema del principio de XP es “Abrazar el cambio” y los cuatro valores fundamentales que permiten a todo el equipo estar constantemente en contacto y responder al entorno cambiante son la simplicidad, la comunicación, la retroalimentación y el coraje” (Sohaib <i>et al.</i> , 2019)	2019

**Fuente:** elaboración propia.

Como se aprecia en la Tabla 5, Extreme Programming está enfocado a la aplicación de buenas prácticas para la producción ágil del software. Entre sus características más importantes son: la flexibilidad frente a los cambios, la producción en corto tiempo, fomenta el ambiente de trabajo en equipo, la producción sostenible y la calidad del software, entre otras.

Extreme Programming (XP), cuenta con cuatro fases: Planificación, Diseño, Desarrollo y Pruebas; cada etapa es co-dependiente de la fase anterior para determinar el curso que toma el proyecto durante su ciclo de vida como lo ejemplifica en la Ilustración 3.



**Ilustración 3.** Proceso Extreme Programming.

**Fuente:** elaboración propia.

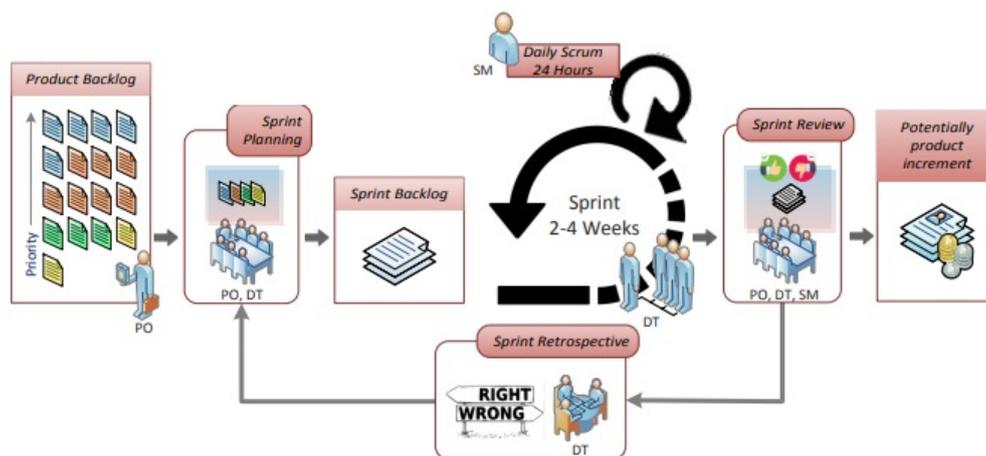
## 2.6. SCRUM

**Tabla 6.** Metodología Scrum.

TÍTULO	CONTENIDO	AÑO
Scrum Language Use in a Software Engineering Firm: An Exploratory Study	"Scrum, una implementación particular de los principios ágiles, es un enfoque ágil altamente estilizado para la gestión de proyectos en términos de personal, reuniones y tareas" (Friess, 2019)	2019
Targeted Scrum: Applying Mission Command to Agile Software Development	Procesos presentes en Scrum: " <ul style="list-style-type: none"> <li>• Product backlog</li> <li>• Sprint Planning</li> <li>• Sprint backlog</li> <li>• Sprint</li> <li>• Product increment</li> <li>• Sprint review</li> <li>• Sprint retrospective." (Harvie y Agah, 2016)</li> </ul>	2016

**Fuente:** elaboración propia.

En el análisis realizado en la Tabla 6, se define a Scrum como un paradigma para el desarrollo de software que usa los fundamentos o principios ágiles y los enfoca para la estimulación del trabajo colaborativo con el fin de obtener el mejor resultado posible. En la Ilustración 4, se ejemplifica el procedimiento de cada fase y proceso dentro de la metodología Scrum.



**Ilustración 4.** Secuencia de procesos de Scrum.

**Fuente:** (Barcelos y Silva, 2020).

## 2.7. RATIONAL UNIFIED PROCESS (RUP)

**Tabla 7.** Metodología RUP.

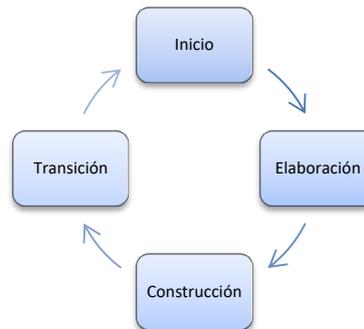
TÍTULO	CONTENIDO	AÑO
Scrum versus Rational Unified Process in facing the main challenges of product configuration systems development	“RUP es una metodología de desarrollo de software que no sigue el enfoque tradicional en cascada, pero prescribe varias iteraciones incrementales para obtener comentarios del usuario, lo cual es útil para alinear la solución de software con los requisitos del usuario.” (Shafiee <i>et al.</i> , 2020)	2020

Integrating the Rational Unified Process and participatory design for development of socio-technical systems: a user participative approach	<p>“Según RUP, la vida útil de un sistema se describe como un número finito de ciclos de desarrollo. Cada ciclo de desarrollo se divide en las cuatro fases del proyecto: Inicio, Elaboración, Construcción y Transición. Las fases, a su vez, se dividen en una serie de iteraciones, dependiendo de las necesidades y el tamaño del proyecto.” (Pilemalm <i>et al.</i>, 2007)</p>	2007
---	---	------

**Fuente:** elaboración propia.

Según el análisis de la Tabla 7, RUP a diferencia del enfoque del desarrollo tradicional, utiliza procesos incrementables con iteraciones que es proporcional a las necesidades del proyecto y dimensión del mismo.

El objetivo de RUP, es la adaptación a los requerimientos del cliente para el aseguramiento de la producción de un software de calidad que satisfaga las necesidades del usuario.



**Ilustración 5.** Metodología RUP.

**Fuente:** elaboración propia.

## 2.8. KANBAN

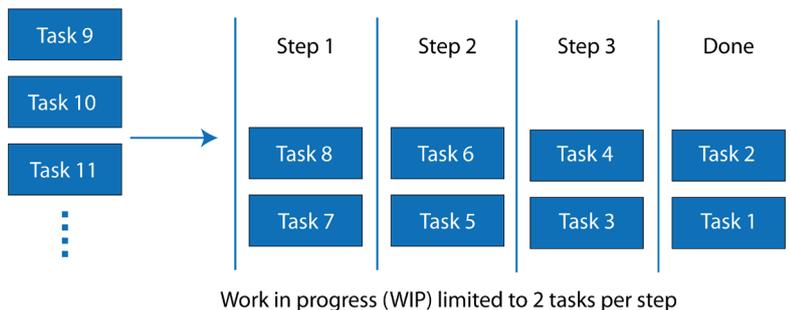
**Tabla 8.** Metodología Kanban.

TÍTULO	CONTENIDO	AÑO
Knowledge kanban system for virtual research and development	“Kanban es la palabra japonesa para tarjeta visual que proporciona información para regular el flujo de inventario y materiales. Kanban tiene tres reglas: (1) visualizar el flujo de trabajo, (2) limitar el trabajo en progreso (WIP) en cada estado del flujo de trabajo y (3) medir el tiempo de entrega (es decir, el tiempo promedio para completar un elemento)” (Jou Lin <i>et al.</i> , 2013)	2018
Kanban in software engineering: A systematic mapping study	“Kanban es descrito Kanban (K mayúscula) es un método de cambio evolutivo que utiliza un sistema de extracción de kanban (k pequeña), visualización y otras herramientas para catalizar la introducción de ideas Lean... el proceso es evolutivo e incremental” (Ahmad <i>et al.</i> , 2018)	2018
The kanban approach, between agility and leanness: a systematic review	“identifica cinco elementos para la implementación exitosa del método Kanban: 1) visualizar el flujo de trabajo, 2) limitar el trabajo en progreso (WIP), 3) administrar el flujo, 4) hacer que las políticas sean explícitas y 5) implementar ciclos de retroalimentación.” (Al-Baik y Miller, 2015)	2015

**Fuente:** elaboración propia.

De acuerdo con la Tabla 8, Kanban es una metodología con orientación al desarrollo ágil, su prioridad es estimular el rendimiento, la organización o colaboración y la distribución de trabajo. Se desenvuelve en torno a tres preceptos que buscan regular o normalizar los recursos del proyecto.

Kanban mantiene un planteamiento definido por fases o etapas como se muestra en la Ilustración 6. Entre las características de esta metodología es su capacidad para identificar los cuellos de botella durante la iteración de cada fase.

**Ilustración 6.** Proceso de Kanban.

**Fuente:** (Lei *et al.*, 2017).

## 2.9. SCRUMBAN

**Tabla 9.** Metodología Scrumban.

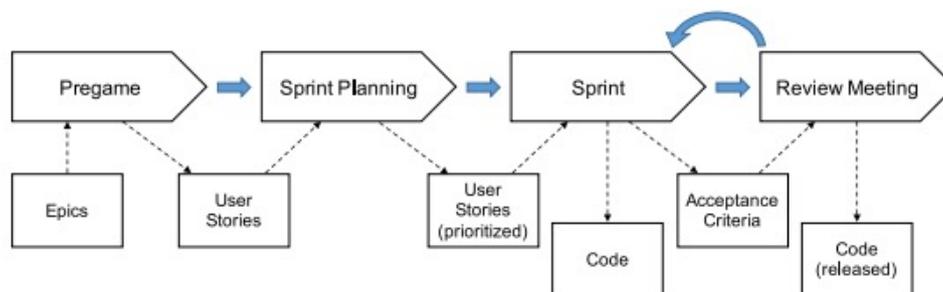
TÍTULO	CONTENIDO	AÑO
A procedure to detect problems of processes in software development projects using Bayesian networks	“Este enfoque se basa en procesos ligeros con un enfoque incremental para la especificación, el desarrollo y la entrega de software para maximizar la entrega de valor a los clientes. Tiene la intención de entregar software de trabajo rápidamente a los usuarios, quienes luego pueden proponer requisitos nuevos y modificados para ser incluidos en iteraciones posteriores del sistema.”(Perkusich <i>et al.</i> , 2015)	2015
Perceived importance of agile requirements engineering practices – A survey	“Los métodos ágiles más populares fueron Scrum y Kanban. Un poco menos populares fueron eXtreme Programming (XP) y Scrumban.”(Ochodek y Kopczyńska, 2018)	2018

**Fuente:** elaboración propia.

De acuerdo con el estudio planteado en la Tabla 9 sobre la metodología Scrumban, se identifican las características de mayor impacto sobre el desarrollo de software ágil. Scrumban, surgió a partir de la

combinación de Scrum y Kanban en busca del equilibrio entre la eficacia y eficiencia.

El proceso de Scrumban corresponde a un modelo firme enfocado a procesos iterativos e incrementales, la mejora continua del producto a través de la eficiencia y la entrega continua.



**Ilustración 7.** Proceso de Scrumban.

**Fuente:** (Lunesu *et al.*, 2018).

## 2.10. ANÁLISIS DE METODOLOGÍAS

Haciendo una retroalimentación a las características u objetivos principales de las metodologías utilizadas en el desarrollo de aplicaciones móviles Tabla 10, se establecen criterios de comparación para un análisis un poco más profundo en cuanto a las ventajas y desventajas entre los marcos de referencia para la producción de aplicaciones móviles.

**Tabla 10.** Características generales de las metodologías.

METODOLOGÍAS	ACRÓNIMO	CARACTERÍSTICAS U OBJETIVOS GENERALES
Mobile-D	Mobile-D	<ul style="list-style-type: none"> <li>• Busca resultados rápidos</li> <li>• Enfocado al desarrollo móvil</li> <li>• Sus etapas se dividen en ciclos de tres días</li> <li>• Es un modelo ágil</li> </ul>

Test-Driven Development	TDD	<ul style="list-style-type: none"> <li>• Evitar escribir código innecesario</li> <li>• Estimula la confianza del desarrollador en su código</li> <li>• Realiza la detección de todos los posibles errores antes del lanzamiento del producto.</li> </ul>
Extreme Programming	XP	<ul style="list-style-type: none"> <li>• Desarrollo incremental e iterativo</li> <li>• Estimula la relación cliente-desarrollador</li> <li>• Refactorización del código</li> <li>• Ejecución de pruebas continuas</li> </ul>
SCUM	SCRUM	<ul style="list-style-type: none"> <li>• Procesos iterativos e incrementales</li> <li>• Aplicación de entregables</li> <li>• Versátil a entornos cambiantes</li> <li>• Emplea principios ágiles</li> </ul>
Rational Unified Process	RUP	<ul style="list-style-type: none"> <li>• Se utiliza para el desarrollo de sistemas orientados a objetos</li> <li>• Orientado al desarrollo web y móvil</li> <li>• Utiliza casos de uso</li> </ul>
KANBAN	KANBAN	<ul style="list-style-type: none"> <li>• Manejo del riesgo</li> <li>• Enfocado a la producción de calidad</li> <li>• El ritmo de trabajo soportado y sustentable.</li> <li>• Optimización del trabajo</li> </ul>
Scrumban	Scrumban	<ul style="list-style-type: none"> <li>• Los procesos son iterativos e incrementales de manera continua</li> <li>• Relación estrecha entre los involucrados del proyecto</li> <li>• Orientado a la eficiencia y efectividad</li> </ul>

**Fuente:** elaboración propia.

En la Tabla 11 se plantea un marco de valorización para referenciar la escala de proyecto a través de un identificador utilizado en la Tabla 12. De acuerdo con las características pertenecientes a cada metodología con relación al tipo de proyecto que lo soporta.

De acuerdo con el análisis en la Tabla 12, las metodologías de desarrollo presentan cierta similitud en más de un criterio establecido para su estudio como en la escala o magnitud del proyecto en el que son empleadas, el tamaño del equipo de trabajo. De igual manera se observa la disparidad con otros criterios como los roles y fases.

**Tabla 11.** Valoración del tamaño de proyectos.

IDENTIFICADOR	ESCALA
1	Proyectos de desarrollo Grandes
2	Proyectos de desarrollo Medianos
3	Proyectos de desarrollo Pequeño

**Fuente:** elaboración propia.

**Tabla 12.** Análisis de metodologías.

	METODOLOGÍAS						
Parámetro	Mobile-D	TDD	XP	SCRUM	RUP	KANBAN	Scrumban
Fases	Exploración. Inicialización. Producción. Estabilización. Pruebas	Añadir test. Correr test. Modificar test. Despliegue.	Añadir test. Correr test. Modificar test. Despliegue.	Planificación de sprint. Etapa de desarrollo. Revisión de sprint. Retroalimentación.	Inicio. Elaboración. Construcción. Transición.	Petición de tareas. Selección de tareas. Desarrollo. Prueba. Finalización	Backlog. Selección de tareas. Análisis y diseño. Construcción. Integración y pruebas. Cierre.
Escala / Magnitud del proyecto	2-3	1-2	1-2-3	1-2-3	2-3	2-3	2-3
Tamaño de equipo (número de personas)	<11	No definido	>10	>7	>15	No definido	No definido
Roles	No definido	Cliente Analista de negocio Desarrolladores Arquitectos Administradores del sistema	Desarrollador Cliente Tester Tracker Coach Gestor	Scrum master Product Owner Team	Analistas Desarrolladores Gestores Apoyo Tester	No definido	No definido

**Fuente:** elaboración propia

### 3. METODOLOGÍA

El presente trabajo de investigación es de carácter descriptivo, explicativo y heurístico con el objetivo de seleccionar los datos adecuados de todo el material documental a fin de realizar el procedimiento de revisión y descripción (Shemberko y Sliva, 2016). El enfoque de este trabajo es el análisis de las metodologías utilizadas en la producción de software móvil, para ello se definieron cuatro etapas:

La investigación bibliográfica de investigaciones o trabajos relacionados a las metodologías de desarrollo de aplicaciones móviles.

Identificación y procesamiento de datos recolectados de fuentes con factor de impacto en el campo de la investigación.

Estudio comparativo de la información o datos recolectados con la verificación correspondiente.

Construcción de resultados y conclusiones sobre las metodologías de desarrollo enfocadas a la producción de aplicaciones móviles.

### 4. RESULTADOS

El crecimiento exponencial del desarrollo de aplicaciones móviles es el producto de la evolución tecnológica y la necesidad de mantenerse conectados a través de la red para realizar distintas actividades tanto laborales como de entretenimiento. Este hecho dio lugar al surgimiento de nuevas metodologías de desarrollo de software con enfoques a las denominadas prácticas ágiles cuyo objetivo es la producción de software de calidad. La selección de metodologías para la construcción de aplicaciones móviles se inclina por aquellos marcos de trabajo enfocados al desarrollo ágil; incluso metodologías creadas específicamente para la producción de aplicaciones móviles como Mobile-D se basan en los mismos principios y buenas prácticas del enfoque ágil. Sin embargo, no todas las metodologías de desarrollo ágil se las puede considerar viables para la producción de aplicaciones móviles debido al contexto de desarrollo para

el que fueron creadas. Es decir, existen metodologías con enfoque ágil, pero son estructuradas para la producción de aplicaciones o sistemas web o de escritorio.

**Tabla 13.** Investigaciones relacionadas.

# DOCUMENTO	AUTOR / ES	TÍTULO	LUGAR
1	Sukhpreet Kaur y Kanwalvir Singh Dhindsa	Design and Development of Android Based Mobile Application for Specially Abled People. (Kaur y Dhindsa, 2020)	India
2	Mamta Pandey, Ratnesh Litoriya y Prateek Pandey	Application of Fuzzy DEMATEL Approach in Analyzing Mobile App Issues. (Pandey <i>et al.</i> , 2019)	India
3	Alisa Ananjeva, John Stouby Persson, y Anders Bruun	Integrating UX work with agile development through user stories: An action research study in a small software company. (Ananjeva <i>et al.</i> , 2020)	Dinamarca
4	Anna Zaitsev, Uri Gal, y Barney Tan	Coordination artifacts in Agile Software Development. (Zaitsev <i>et al.</i> , 2020)	Australia
5	Pilar Mata, Austin Chamney, Gary Viner, Douglas Archibald y Liam Peyton	A development framework for mobile healthcare monitoring apps. (Mata <i>et al.</i> , 2015)	Londres
6	Steffen Vaupel, Gabriele Taentzer, René Gerlach y Michael Guckert	Model-driven development of mobile applications for Android and iOS supporting role-based app variability.(Vaupel <i>et al.</i> , 2018)	Alemania
7	Karina Curcio, Rodolfo Santana, Sheila Reinehr, Andreia Malucelli	Usability in agile software development: A tertiary study. (Curcio <i>et al.</i> , 2019)	Brazil

8	S. Sivaprakash y M. Venkatesan	A design and development of an intelligent jammer and jamming detection methodologies using machine learning approach. (Sivaprakash y Venkatesan, 2019)	India
9	Luis Valente, Bruno Feijó y Julio Cesar Sampaio do Prado Leite	Mapping quality requirements for pervasive mobile games. (Valente <i>et al.</i> , 2017)	Londres
10	Porfirio Tramontana, Domenico Amalfitano, Nicola Amatucci y Anna Rita Fasolino	Automated functional testing of mobile applications: a systematic mapping study. (Tramontana <i>et al.</i> , 2019)	Italia
11	Cinzia Bernardeschi, Luca Cassano y Andrea Domenici	SRAM-Based FPGA Systems for Safety-Critical Applications: A Survey on Design Standards and Proposed Methodologies. (Bernardeschi <i>et al.</i> , 2015)	Italia
12	Ainhoa Aldave, Juan M. Vara, David Granada, y Esperanza Marcos	Leveraging creativity in requirements elicitation within agile software development: A systematic literature review. (Aldave <i>et al.</i> , 2019)	España

**Fuente:** elaboración propia.

## 5. DISCUSIÓN

El punto de partida de esta investigación fue la necesidad de conocer la metodología de desarrollo de software adecuada para la producción de aplicaciones móviles. Para ello se llevó a cabo una investigación exhaustiva de las metodologías más populares en el ámbito del desarrollo de aplicaciones móviles para contestar a los enigmas que surgieron a lo largo de la investigación.

El manifiesto ágil a través de sus principios y buenas prácticas ha tenido un gran impacto sobre las metodologías de desarrollo indistintamente si se trata de desarrollo web, móvil o de escritorio. Marcos de trabajo como Extreme Programming, Scrum, Kanban, TDD, entre otras; utilizan principios del manifiesto ágil que se caracterizan por ser iterativos, flexibles al cambio, mantener la prioridad de la funcionalidad sobre la documentación, etc. Sin embargo, las metodologías que sobresalen de aquellas que emplean los mismos principios son: Extreme Programming, Kanban y Scrum (Matharu *et al.*, 2015) por diferentes factores que toman en consideración durante el ciclo de vida del software.

Sin embargo, Taibi *et al.* (2017), en su investigación, demuestra la disparidad que existe en medio de Extreme Programming, Scrum y Kanban en aspectos relacionados a la comunicación interna entre cada integrante en sus grupos de proyecto. Una de las causas de este hecho es la forma en como manejan el flujo de información en cada grupo de trabajo. Scrum y Kanban al estar estructurados y enfocados a la comunicación general, opta por realizar reuniones entre varios miembros del equipo para dar soporte y seguimiento a los procesos en curso a diferencia de Extreme Programming que fomenta la participación uno a uno.

La escalabilidad es otro factor importante para el análisis, puesto que se considera la aplicación de la metodología a distintos tipos de proyecto en cuanto a la magnitud o alcance del mismo durante su ejecución. Por lo tanto, Scrum mantiene la ventaja sobre Kanban y otras metodologías, debido a su adaptación a todo tipo de proyecto, posee un mejor sistema de comunicación, la iteración que realiza entre sus fases, el uso de herramientas que promueven la participación en conjunto, etc. Todas estas características han sido propicias para considerar a Scrum como la metodología más óptima dentro del desarrollo de aplicaciones móviles.

## 6. CONCLUSIONES

De acuerdo con los resultados alcanzados en la presente investigación se puede concluir que las metodologías de desarrollo de software con base en las buenas prácticas ágiles en la actualidad son

mayormente utilizadas por los beneficios que aportan para la construcción de soluciones de software. Sin embargo, en el ámbito del desarrollo de aplicaciones móviles, no es posible hacer uso de cualquier metodología del género ágil debido a la parte estructural de algunos marcos de trabajo que están enfocados al desarrollo de software o sistemas web.

La metodología Scrum ha sido caracterizada por los beneficios que aporta para la realización de un proyecto de software. En la comparación ejecutada entre las metodologías más populares del enfoque ágil se logró evidenciar cierta similitud que mantienen estos marcos de trabajo fundamentados en los mismos principios. No obstante, Scrum hace énfasis sobre un modelo de desarrollo incremental en bloques de ciclo corto que beneficia la sincronización de la información fomentando la flexibilidad frente a los nuevos cambios que se requieran ejecutar durante la realización del proyecto sin restricciones en cuanto al tamaño del proyecto. Por lo tanto, esta metodología es considerada con mayor frecuencia para el desarrollo de aplicaciones móviles.

## REFERENCIAS BIBLIORÁFICAS

- Ahmad, M. O., Dennehy, D., Conboy, K., y Oivo, M.** (2018). Kanban in software engineering: A systematic mapping study. *Journal of Systems and Software*, 137, 96-113. <https://doi.org/10.1016/j.jss.2017.11.045>
- Al-Baik, O., y Miller, J.** (2015). The kanban approach, between agility and leanness: A systematic review. *Empirical Software Engineering*, 20(6), 1861-1897. <https://doi.org/10.1007/s10664-014-9340-x>
- Aldave, A., Vara, J. M., Granada, D., y Marcos, E.** (2019). Leveraging creativity in requirements elicitation within agile software development: A systematic literature review. *Journal of Systems and Software*, 157, 110396. <https://doi.org/10.1016/j.jss.2019.110396>

- Amorim, A. C., Mira, M., Pereira, R., y Gonçalves, M.** (2020). Using agile methodologies for adopting COBIT. *Information Systems*, 101496. <https://doi.org/10.1016/j.is.2020.101496>
- Ananjeva, A., Persson, J. S., y Bruun, A.** (2020). Integrating UX work with agile development through user stories: An action research study in a small software company. *Journal of Systems and Software*, 170, 110785. <https://doi.org/10.1016/j.jss.2020.110785>
- Barcelos Bica, D. A., y Silva, C. A. G. da.** (2020). Learning Process of Agile Scrum Methodology With Lego Blocks in Interactive Academic Games: Viewpoint of Students. *IEEE Revista Iberoamericana de Tecnologías del Aprendizaje*, 15(2), 95-104. <https://doi.org/10.1109/RITA.2020.2987721>
- Batarseh, F. A., y Gonzalez, A. J.** (2018). Predicting failures in agile software development through data analytics. *Software Quality Journal*, 26(1), 49-66. <https://doi.org/10.1007/s11219-015-9285-3>
- Batra, D.** (2020). Job-work fit as a determinant of the acceptance of large-scale agile methodology. *Journal of Systems and Software*, 168, 110577. <https://doi.org/10.1016/j.jss.2020.110577>
- Bernardeschi, C., Cassano, L., y Domenici, A.** (2015). SRAM-Based FPGA Systems for Safety-Critical Applications: A Survey on Design Standards and Proposed Methodologies. *Journal of Computer Science and Technology*, 30(2), 373-390. <https://doi.org/10.1007/s11390-015-1530-5>
- Cerqueira, M. M. B., Magalhães, A. P., Saba, H., y Jorge, E. M. F.** (2018). A Software Development Process for Super Agile Projects. *Proceedings of the XIV Brazilian Symposium on Information Systems*, 1–8. <https://doi.org/10.1145/3229345.3229399>
- Corral, L., Sillitti, A., y Succi, G.** (2015). Software assurance practices for mobile applications. *Computing*, 97(10), 1001-1022. <https://doi.org/10.1007/s00607-014-0395-8>
- Curcio, K., Santana, R., Reinehr, S., y Malucelli, A.** (2019). Usability in agile software development: A tertiary study. *Computer Standards & Interfaces*, 64, 61-77. <https://doi.org/10.1016/j.csi.2018.12.003>

- Friess, E.** (2019). Scrum Language Use in a Software Engineering Firm: An Exploratory Study. *IEEE Transactions on Professional Communication*, 62(2), 130-147. <https://doi.org/10.1109/TPC.2019.2911461>
- Harvie, D. P., y Agah, A.** (2016). Targeted Scrum: Applying Mission Command to Agile Software Development. *IEEE Transactions on Software Engineering*, 42(5), 476-489. <https://doi.org/10.1109/TSE.2015.2489654>
- Hassan, S., Shang, W., y Hassan, A. E.** (2017). An empirical study of emergency updates for top android mobile apps. *Empirical Software Engineering*, 22(1), 505-546. <https://doi.org/10.1007/s10664-016-9435-7>
- Jabangwe, R., Edison, H., y Duc, A. N.** (2018). Software engineering process models for mobile app development: A systematic literature review. *Journal of Systems and Software*, 145, 98-111. <https://doi.org/10.1016/j.jss.2018.08.028>
- Jou, C., Frank, F., y Min, Y.** (2013). Knowledge kanban system for virtual research and development. *Robotics and Computer-Integrated Manufacturing*, 29(3), 119-134. <https://doi.org/10.1016/j.rcim.2012.04.020>
- Kaur, S., y Dhindsa, K. S.** (2020). Design and Development of Android Based Mobile Application for Specially Abled People. *Wireless Personal Communications*, 111(4), 2353-2367. <https://doi.org/10.1007/s11277-019-06990-y>
- Kaushik, A., Tayal, D. Kr., y Yadav, K.** (2020). A Comparative Analysis on Effort Estimation for Agile and Non-agile Software Projects Using DBN-ALO. *Arabian Journal for Science and Engineering*, 45(4), 2605-2618. <https://doi.org/10.1007/s13369-019-04250-6>
- Kerthyayana, I. B.** (2019). Combination of Test-Driven Development and Behavior-Driven Development for Improving Backend Testing Performance. *Procedia Computer Science*, 157, 79-86. <https://doi.org/10.1016/j.procs.2019.08.144>

- Lunesu, M. I., Münch, J., Marchesi, M., y Kuhrmann, M.** (2018). Using simulation for understanding and reproducing distributed software development processes in the cloud. *Information and Software Technology*, 103, 226-238. <https://doi.org/10.1016/j.infsof.2018.07.004>
- Martinez, D., Ferre, X., Guerrero, G., y Juristo, N.** (2020). An Agile-Based Integrated Framework for Mobile Application Development Considering Ilities. *IEEE Access*, 8, 72461-72470. <https://doi.org/10.1109/ACCESS.2020.2987882>
- Mata, P., Chamney, A., Viner, G., Archibald, D., y Peyton, L.** (2015). A development framework for mobile healthcare monitoring apps. *Personal and Ubiquitous Computing*, 19(3), 623-633. <https://doi.org/10.1007/s00779-015-0849-9>
- Matharu, G. S., Mishra, A., Singh, H., y Upadhyay, P.** (2015). Empirical Study of Agile Software Development Methodologies: A Comparative Analysis. *ACM SIGSOFT Software Engineering Notes*, 40(1), 1–6. <https://doi.org/10.1145/2693208.2693233>
- McIlroy, S., Ali, N., Khalid, H., y Hassan, A. E.** (2016). Analyzing and automatically labelling the types of user issues that are raised in mobile app reviews. *Empirical Software Engineering*, 21(3), 1067-1106. <https://doi.org/10.1007/s10664-015-9375-7>
- Ochodek, M., y Kopczyńska, S.** (2018). Perceived importance of agile requirements engineering practices – A survey. *Journal of Systems and Software*, 143, 29-43. <https://doi.org/10.1016/j.jss.2018.05.012>
- Özcan-Top, Ö., y McCaffery, F.** (2019). To what extent the medical device software regulations can be achieved with agile software development methods? XP—DSDM—Scrum. *The Journal of Supercomputing*, 75(8), 5227-5260. <https://doi.org/10.1007/s11227-019-02793-x>
- Pandey, M., Litoriya, R., y Pandey, P.** (2019). Application of Fuzzy DEMATEL Approach in Analyzing Mobile App Issues. *Programming and Computer Software*, 45(5), 268-287. <https://doi.org/10.1134/S0361768819050050>

- Perkusich, M., Soares, G., Almeida, H., y Perkusich, A.** (2015). A procedure to detect problems of processes in software development projects using Bayesian networks. *Expert Systems with Applications*, 42, 437–450. <https://doi.org/10.1016/j.eswa.2014.08.015>
- Pilemalm, S., Lindell, P.-O., Hallberg, N., y Eriksson, H.** (2007). Integrating the Rational Unified Process and participatory design for development of socio-technical systems: A user participative approach. *Design Studies*, 28(3), 263-288. <https://doi.org/10.1016/j.destud.2007.02.009>
- Shafiee, S., Wautelet, Y., Hvam, L., Sandrin, E., y Forza, C.** (2020). Scrum versus Rational Unified Process in facing the main challenges of product configuration systems development. *Journal of Systems and Software*, 170, 110732. <https://doi.org/10.1016/j.jss.2020.110732>
- Shemberko, L. V., y Sliva, A. I.** (2016). Literary studies: Information consumers, databases, principles of informal processing, and search strategies. *Scientific and Technical Information Processing*, 43(4), 217-228. <https://doi.org/10.3103/S0147688216040031>
- Sivaprakash, S., y Venkatesan, M.** (2019). A design and development of an intelligent jammer and jamming detection methodologies using machine learning approach. *Cluster Computing*, 22(1), 93-101. <https://doi.org/10.1007/s10586-018-2822-7>
- Sohaib, O., Solanki, H., Dhaliwa, N., Hussain, W., y Asif, M.** (2019). Integrating design thinking into extreme programming. *Journal of Ambient Intelligence and Humanized Computing*, 10(6), 2485-2492. <https://doi.org/10.1007/s12652-018-0932-y>
- Taibi, D., Lenarduzzi, V., Ahmad, M. O., y Liukkunen, K.** (2017). Comparing Communication Effort within the Scrum, Scrum with Kanban, XP, and Banana Development Processes. *Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering*, 258–263. <https://doi.org/10.1145/3084226.3084270>

- Tosun, A., Dieste, O., Fucci, D., Vegas, S., Turhan, B., Erdogmus, H., Santos, A., Oivo, M., Toro, K., Jarvinen, J., y Juristo, N.** (2017). An industry experiment on the effects of test-driven development on external quality and productivity. *Empirical Software Engineering*, 22(6), 2763-2805. <https://doi.org/10.1007/s10664-016-9490-0>
- Tramontana, P., Amalfitano, D., Amatucci, N., y Fasolino, A. R.** (2019). Automated functional testing of mobile applications: A systematic mapping study. *Software Quality Journal*, 27(1), 149-201. <https://doi.org/10.1007/s11219-018-9418-6>
- Turner-McGrievy, G. M., Hales, S. B., Schoffman, D. E., Valafar, H., Brazendale, K., Weaver, R. G., Beets, M. W., Wirth, M. D., Shivappa, N., Mandes, T., Hébert, J. R., Wilcox, S., Hester, A., y McGrievy, M. J.** (2017). Choosing between responsive-design websites versus mobile apps for your mobile behavioral intervention: Presenting four case studies. *Translational Behavioral Medicine*, 7(2), 224-232. <https://doi.org/10.1007/s13142-016-0448-y>
- Valente, L., Feijó, B., y Leite, J. C. S. do P.** (2017). Mapping quality requirements for pervasive mobile games. *Requirements Engineering*, 22(1), 137-165. <https://doi.org/10.1007/s00766-015-0238-y>
- Vaupel, S., Taentzer, G., Gerlach, R., y Guckert, M.** (2018). Model-driven development of mobile applications for Android and iOS supporting role-based app variability. *Software & Systems Modeling*, 17(1), 35-63. <https://doi.org/10.1007/s10270-016-0559-4>
- Yang, P., Liu, Z., Xu, J., Huang, Y., y Pan, Y.** (2020). An Empirical Study on the Ability Relationships Between Programming and Testing. *IEEE Access*, 8, 161438-161448. <https://doi.org/10.1109/ACCESS.2020.3018718>
- Zaitsev, A., Gal, U., y Tan, B.** (2020). Coordination artifacts in Agile Software Development. *Information and Organization*, 30(2), 100288. <https://doi.org/10.1016/j.infoandorg.2020.100288>

