

# COMPARATIVA DE METODOLOGÍAS DE DESARROLLO DE APLICACIONES MÓVILES

## COMPARISON OF MOBILE APPLICATION 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:** 25/01/2021 **Aceptación:** 26/05/2021 **Publicación:** 14/06/2021

#### **Citación sugerida:**

Molina, J. R., Honores, J. A., Pedreira-Souto, N., y Pardo, H. P. (2021). Comparativa de metodologías de desarrollo de aplicaciones móviles. *3C Tecnología. Glosas de innovación aplicadas a la pyme*, 10(2), 73-93. <https://doi.org/10.17993/3ctecno/2021.v10n2e38.73-93>

## RESUMEN

El desarrollo de aplicaciones móviles en la actualidad tiene una gran aceptación gracias al avance de la tecnología y producción de toda clase de dispositivos que permiten a los usuarios realizar tareas cotidianas ya sean de entretenimiento o laboral. Por ende, la necesidad de producir software de calidad y para ello se han desarrollado múltiples metodologías que buscan optimizar procesos a través de buenas prácticas y principios ágiles. El objetivo de la presente investigación es realizar una búsqueda exhaustiva de las metodologías de desarrollo enfocadas a la producción aplicaciones móviles para realizar una comparación de carácter analítica y de campo. Como resultado se obtuvo que Scrum abarca gran parte de los elementos y características que beneficiarían al desarrollo de aplicaciones móviles, de igual modo en el ámbito profesional, las empresas desarrolladoras de software además de usar Mobile-ID, emplean Scrum como un marco de trabajo completo que se adapta a toda clase de proyecto en cuanto al tamaño.

## PALABRAS CLAVE

Metodologías Ágiles, Scrum, Desarrollo Móvil, Aplicaciones Móviles, Comparativa De Metodologías.

## ABSTRACT

*The development of mobile applications is currently widely accepted thanks to the advancement of technology and the production of all kinds of devices that allow users to carry out daily tasks, whether they are entertainment or work. Therefore, the need to produce quality software and for this, multiple methodologies have been developed that seek to optimize processes through good practices and agile principles. The objective of this research is to carry out an exhaustive search of development methodologies focused on the production of mobile applications to carry out an analytical and field comparison. As a result, it was obtained that Scrum covers a large part of the elements and characteristics that would benefit the development of mobile applications, in the same way in the professional field, software development companies in addition to using Mobile-D, use Scrum as a complete framework that adapts to all kinds of projects in terms of size.*

## KEYWORDS

*Agile Methodologies, Scrum, Mobile Development, Mobile Applications, Methodologies Comparison.*

## 1. INTRODUCCIÓN

El desarrollo de software es una de las áreas de investigación con mayor influencia hoy en día, dado que constantemente emergen nuevas tendencias que buscan beneficiar a la producción de software con calidad. Las metodologías de desarrollo son responsables de lograr dicho objetivo por ende se observan cambios evidentes en su línea de tiempo. Actualmente, las nuevas metodologías de desarrollo se fundamentan en los principios ágiles, aplicando buenas prácticas y procesos o fases iterativas orientadas a mantener la flexibilidad frente a los cambios inesperados durante la ejecución de un proyecto.

Por otra parte, la producción de aplicaciones móviles ha tenido un progreso bastante significativo en los últimos años según Ho y Chung (2020). El incremento de aplicaciones móviles en el mercado oscila en un promedio de 4000 nuevas aplicaciones por día, lo cual refleja la alta demanda en la producción de aplicaciones para los distintos dispositivos móviles que existen hasta ahora. Por esa razón, las metodologías de desarrollo de software buscan abarcar la mayor parte de los aspectos considerados durante la ejecución de un proyecto de software con el objetivo de obtener un producto de software de calidad.

En el desarrollo de aplicaciones móviles existe una tendencia por el uso de las metodologías con principios ágiles por los beneficios que brinda durante el ciclo de vida del software. Sin embargo, no se pueden considerar todos los marcos de trabajo con base en los mismos principios como metodologías ideales para el desarrollo de aplicaciones móviles.

La presente investigación de carácter analítica, bibliográfica y de campo realiza una comparación entre las metodologías de desarrollo ágil cuyas directrices están orientadas a aspectos como el alcance, la estructura, modelado, el ciclo de vida e instrumentos de soporte.

Según la comparación que se llevó a cabo, deja en evidencia los distintos enfoques de cada metodología y su compromiso de resolver problemas detectados en otras, a fin de establecer un nuevo marco de referencia para el desarrollo de aplicaciones móviles que cumplan con los niveles de calidad esperados.

Por otra parte, se corroboró que las metodologías mayormente aceptadas y aplicadas son aquellas que engloban un número significativo de componentes o normas contempladas en otros marcos de trabajo.

En base a la comparación y análisis en el contexto de elaboración del presente trabajo de investigación se obtuvo como resultado que Scrum es ideal para el desarrollo de aplicaciones móviles por sus características que permiten reducir el uso de recursos y maximiza la productividad durante el ciclo de vida del proyecto.

## 2. FUNDAMENTOS TEÓRICOS

### 2.1. APLICACIONES MÓVILES

Con la aparición de la telefonía celular surgió el desarrollo de aplicaciones móviles encargadas de la producción de programas relativamente cortos que desempeñaban funciones importantes para la intercomunicación entre los usuarios en la red (Bhardwaj *et al.*, 2019). Actualmente las aplicaciones móviles desempeñan un rol importante para la adaptación de los usuarios en la era de la digitalización donde gran parte de los procesos cotidianos se los puede realizar a través de un dispositivo móvil inteligente (McIlroy *et al.*, 2016).

### 2.2. DESARROLLO DE APLICACIONES MÓVILES

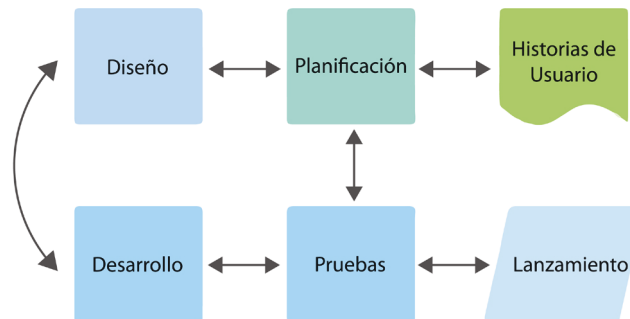
La producción de aplicaciones móviles necesita de un marco de trabajo que sirva de guía para el desarrollo de las aplicaciones que abarque la mayor cantidad de elementos relacionados con la calidad. Por ejemplo, Owoseni y Twinomurinzi (2018), menciona que: “Las aplicaciones móviles a menudo aprovechan la conectividad a Internet para mejorar la capacidad, como en los casos de las aplicaciones de redes sociales”. En algunos casos ciertas metodologías no consideran características que pueden contribuir con en el proceso de mejoramiento de la capacidad.

## 2.3. METODOLOGÍAS DE DESARROLLO ÁGILES

En el desarrollo de software móvil se hace énfasis por el uso de las metodologías de desarrollo basadas en el enfoque ágil debido a las ventajas que ofrecen ante los cambios, además de otras consideraciones. Según Kroener *et al.* (2019): “El núcleo del enfoque de desarrollo ágil es que las necesidades de los usuarios son el punto de partida para la entrega rápida de software funcional, que luego se revisa iterativamente —en escalas de tiempo breves— y se mejora en estrecha colaboración con el cliente y los usuarios finales”. De acuerdo con Albers *et al.* (2020), hace la siguiente afirmación sobre el enfoque que mantienen: “Los procedimientos de desarrollo ágiles para aumentar la agilidad de los equipos de desarrollo están particularmente establecidos en el campo del desarrollo de software.”

## 2.4. EXTREME PROGRAMMING

Extreme Programming también reconocida por sus siglas XP, es una metodología de desarrollo de software ligera fundamentada en los principios del manifiesto ágil (Sohaib *et al.*, 2019). Esta metodología se centraliza en cuatro aspectos importantes de los proyectos de desarrollo de software los cuales son: alcance, tiempo, costo y calidad. Además estimula el trabajo colaborativo en pareja (da Silva y Prikladnicki, 2015).

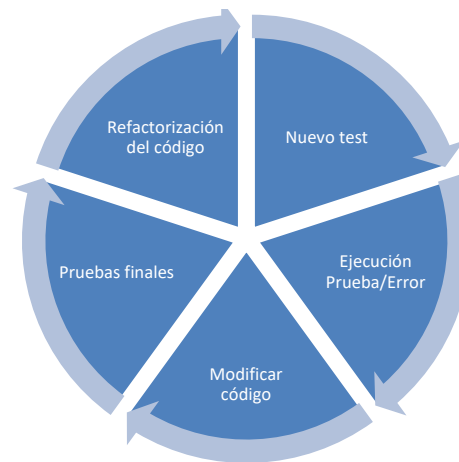


**Ilustración 1:** Fases de Extreme Programming.

**Fuente:** elaboración propia.

## 2.5. TEST-DRIVEN DEVELOPMENT

Test-Driven Development es una metodología orientada específicamente a la realización de pruebas y refactorización del código para obtener un software de calidad (Borle *et al.*, 2018). En el trabajo literario de Nanthaamornphong y Carver (2017) se menciona el impacto positivo de aplicar TDD en los proyectos de software donde alcanzaron productos con altos índices de calidad como resultado de la aplicación de pruebas iterativas durante el ciclo completo en el desarrollo del software.

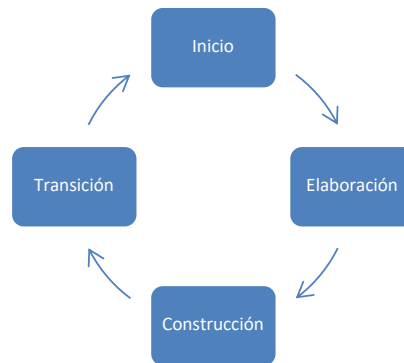


**Ilustración 2:** Fases de Test-Driven Development.

**Fuente:** elaboración propia.

## 2.6. RATIONAL UNIFIED PROCESS

RUP es una metodología de desarrollo orientada a la asignación de roles, responsabilidades y tareas dentro del equipo de trabajo. De acuerdo con Reyes-Delgado *et al.* (2016), la prioridad de RUP no se limita a desarrollar un producto de acuerdo a la planificación, también se enfatiza con entregar resultados que satisfagan las necesidades del cliente con índices altos de calidad con el presupuesto establecido.

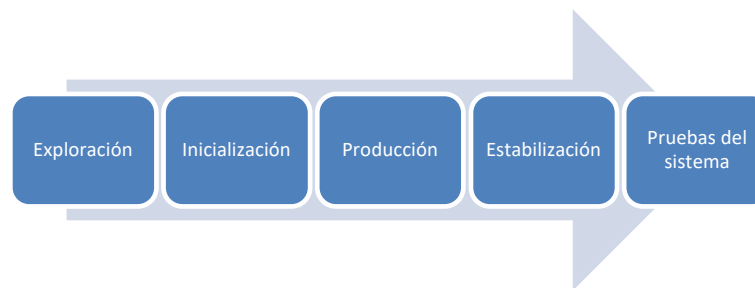


**Ilustración 3:** Fases de Rational Unified Process.

**Fuente:** elaboración propia.

## 2.7. MOBILE-D

“Fue introducido en 2004 como una metodología de desarrollo inspirada en la Programación Extrema, las Metodologías Crystal y el Proceso Unificado Racional” (Corral *et al.*, 2015). Mobile-D surgió de una combinación de otras soluciones ya conocidas que utilizan el mismo principio ágil que se enfoca más en la funcionalidad del software que en la documentación, opta por la participación directa del cliente antes que una negociación contractual y una de las más importante es la flexibilidad frente a los cambios antes que dar seguimiento a un plan (Pandey *et al.*, 2019).



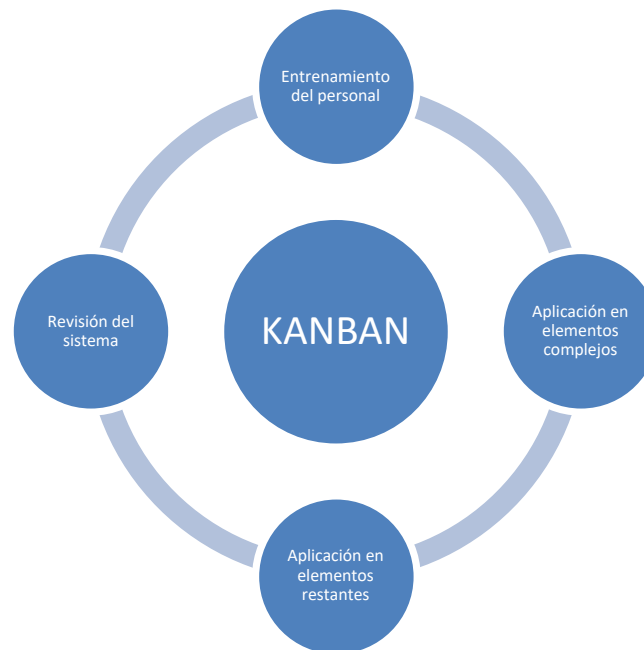
**Ilustración 4** Fases de mobile-D.

**Fuente:** elaboración propia.



## 2.8. KANBAN

Según Heikkilä *et al.* (2016) definen a Kanban de la siguiente manera: “Kanban es un método de gestión del flujo de trabajo especialmente adecuado para gestionar el trabajo continuo de ingeniería de software”. Pertenece al grupo de metodologías ágiles que busca optimizar el proceso de desarrollo del software con el afán de alcanzar resultados con altos índices de calidad. Entre las características de Kanban está el uso de tableros para realizar procesos y las historias de usuarios. Sin embargo, esta metodología no emplea los sprints y posee cuatro elementos orientados a la comunicación, producción, control centralizado y de proceso (Liskin *et al.*, 2014; Krishnaiyer y Chen, 2017).

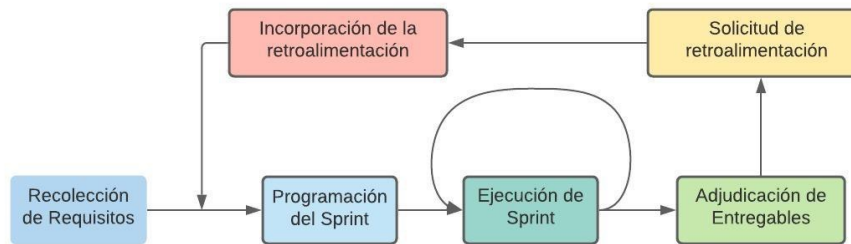


**Ilustración 5:** Fases de Kanban.

**Fuente:** elaboración propia.

## 2.9. SCRUM

Scrum es una metodología que se enfoca a estimular el trabajo colaborativo enfocado a cuatro elementos fundamentales considerados para el desarrollo de un proyecto de software, según Chaouch *et al.* (2019) y Aamir y Khan (2017), Scrum es uno de los framework más utilizados para los proyectos de desarrollo más complejos y con una mayor documentación para su implementación. En adición, Steghöfer *et al.* (2017), afirman lo siguiente: “Las iteraciones permiten el refinamiento continuo del producto desarrollado, donde el final de cada iteración (llamado Sprint) permite al cliente dar su opinión durante una reunión de Revisión de Sprint”.



**Ilustración 6:** Metodología Scrum.

**Fuente:** elaboración propia.

## 2.10. SCRUMBAN

“Al observar Scrum y Kanban por separado, podemos ver que ambos enfoques plantean desafíos para DevOps. Pero no tienen por qué representar alternativas; Se recomienda la conexión de Scrum y Kanban desde un punto de vista holístico” (Schaefer y Söllner, 2017). Sin embargo, Scrumban se encamina por el desarrollo del software ágil reuniendo los elementos y características de las metodologías Scrum y Kanban con el objetivo de mejorar la gestión del flujo de trabajo para aumentar la eficiencia y satisfacción del usuario final o cliente (Lunesu *et al.*, 2018).

### 3. METODOLOGÍA

La ejecución del presente estudio tiene como fundamento la investigación bibliográfica, analítica y de campo para el refinamiento del conjunto de datos recolectados (Snyder, 2019) con respecto a las metodologías de desarrollo de software orientadas a la producción de aplicaciones móviles. A fin de llevar a cabo el siguiente estudio se plantearon cuatro etapas enfocadas en realizar la comparación de las metodologías de desarrollo de aplicaciones móviles; las etapas de la investigación se distribuyen de la siguiente manera:

- La exploración bibliográfica de investigaciones relacionadas a las metodologías de desarrollo orientadas a la producción de aplicaciones móviles. A parte de la integración elementos como organizaciones y desarrolladores independientes de la provincia de El Oro relacionados a la construcción de aplicaciones móviles.
- Valoración y procesamiento del conjunto de información obtenida de la exploración bibliográfica.
- Planteamiento de un marco comparativo para las metodologías de desarrollo de aplicaciones móviles.
- Construcción de resultados y desarrollo de la conclusión referente a las metodologías de desarrollo de aplicaciones móviles determinando la metodología más óptima.

### 4. RESULTADOS

En esta sección se ejecuta una serie de comparaciones utilizando diferentes criterios generales, de calidad, del diseño estructural y requisitos a fin de puntualizar el alcance que tiene cada metodología en base a los parámetros antes ya mencionados.

**Tabla 1.** Identificación de metodologías y sus Acrónimos.

NOMBRE DE METODOLOGÍA	ACRÓNIMO	IDENTIFICADOR
Extreme Programming	XP	M1
Test-Driven Development	TDD	M2
Rational Unified Process	RUP	M3
Mobile-D	-	M4
KANBAN	-	M5
Scrum	-	M6
Scrumban	-	M7

**Fuente:** elaboración propia.

En la Tabla 1 se especifica los acrónimos de cada metodología al igual que el identificador que se le otorga con el objetivo de mantener una menor legibilidad durante la comparación que se realiza entre las metodologías.

**Tabla 2.** Equiparación global de las metodologías.

ID	VENTAJAS	DESVENTAJAS	AÑO DE APARICIÓN
M1	<ul style="list-style-type: none"> <li>• Fomenta la confianza de los desarrolladores.</li> <li>• Estimula la programación organizada.</li> <li>• Se enfoca en la eficiencia de sus procesos.</li> <li>• Facilita la aplicación de cambios.</li> <li>• Se ajusta fácilmente a nuevas tecnologías.</li> </ul>	<ul style="list-style-type: none"> <li>• Imposible predecir errores antes del desarrollo.</li> <li>• Aplicación de costos elevados.</li> <li>• Su utilización suele ser igual de compleja a una metodología tradicional.</li> </ul>	1996-1999
M2	<ul style="list-style-type: none"> <li>• Optimización del código.</li> <li>• Separación de interfaces e implementación.</li> <li>• Promueve el desarrollo lean.</li> </ul>	<ul style="list-style-type: none"> <li>• Las pruebas unitarias se limitan a probar todo el código.</li> <li>• Errores en la aplicación de pruebas.</li> <li>• Se necesitan pruebas de integración.</li> <li>• En ocasiones se obvian la aplicación de pruebas.</li> </ul>	1996-2002

M3	<ul style="list-style-type: none"> <li>• Disminución de riesgos en el desarrollo del proyecto.</li> <li>• Enfocada en obtener un producto de calidad.</li> <li>• Organización en la asignación de roles y tareas.</li> </ul>	<ul style="list-style-type: none"> <li>• Intenta predecir los errores durante todo el ciclo del proyecto.</li> <li>• Elevan el costo de producción.</li> <li>• No es aplicable a proyectos pequeños.</li> <li>• Es compleja su aplicación.</li> </ul>	1998
M4	<ul style="list-style-type: none"> <li>• Bajo costos de aplicación.</li> <li>• Garantiza la producción del producto en el tiempo establecido.</li> <li>• Producción de entregables o demos.</li> </ul>	<ul style="list-style-type: none"> <li>• No tiene un marco sólido de comunicación.</li> <li>• No es aplicable a proyectos grandes o fragmentados.</li> <li>• Carece de procesos que garanticen la calidad en los resultados.</li> </ul>	2005
M5	<ul style="list-style-type: none"> <li>• Responsabilidad con los tiempos de entrega.</li> <li>• Posee procesos enfocados a la calidad del producto.</li> <li>• Optimización en la gestión de tareas.</li> <li>• Previene la sobrecarga en tareas.</li> </ul>	<ul style="list-style-type: none"> <li>• Costos elevados en la aplicación de proyectos grandes</li> <li>• Mantiene una limitación en el número de tareas que puede realizar</li> <li>• No es escalable a proyectos grandes</li> </ul>	2003-2008
M6	<ul style="list-style-type: none"> <li>• Se incorpora al cliente de manera directa al desarrollo del proyecto.</li> <li>• Se optimizan los procesos al dividir las tareas complejas a pequeñas partes sencillas.</li> <li>• Estabilidad al implementar cambios inesperados.</li> <li>• Gestión sistemática de riesgos.</li> <li>• Generación de entregables.</li> <li>• Aplicable a cualquier tamaño de proyectos.</li> </ul>	<ul style="list-style-type: none"> <li>• Los procesos son dependientes del gestor de proyecto.</li> <li>• Puede generar incomodidad en la adaptación de los integrantes.</li> </ul>	1995

M7	<ul style="list-style-type: none"> <li>• Fomenta la mejora continua.</li> <li>• Priorización de tareas.</li> <li>• Control y seguimiento durante el ciclo del proyecto.</li> <li>• Se ejecutan decisiones y hechos en el momento indicado.</li> </ul>	<ul style="list-style-type: none"> <li>• La comunicación se ve limitada al uso de una aplicación móvil.</li> <li>• Fácil desvío del objetivo principal del proyecto.</li> <li>• Carente de una visión general para evaluar los resultados del proyecto.</li> <li>• Falencias en el control de la contribución de cada miembro.</li> <li>• Fomenta la generación de nuevos procesos independientes por cada individuo.</li> </ul>	2005-2009
----	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------

**Fuente:** elaboración propia..

En la Tabla 2, se plantea un análisis general en función de las características principales de las metodologías donde se corrobora la superioridad de Scrum en cuanto al costo y beneficio que conlleva su ejecución sobre un proyecto. A pesar de considerarse el año de aparición, las nuevas metodologías se enfocan más en resolver problemáticas específicas que observan en otras metodologías sin considerar el mejoramiento en otros elementos en el proceso de producción de una aplicación móvil.

**Tabla 3.** Comparación de estructuras y requisitos entre las metodologías.

PROPIEDADES	METODOLOGÍAS						
	M1	M2	M3	M4	M5	M6	M7
Tamaño del grupo.	<20	-	<7	<11	Adaptable a cualquier tamaño	Adaptable a cualquier tamaño	Adaptable a cualquier tamaño
Enfoque	Procesos iterativos incrementales	Ejecución de pruebas	Procesos iterativos incrementales	Ciclos cortos y rápidos	Desarrollo incremental y gestión de tareas	Ejecución de sprints en tiempo fijo	Aplicación de sprint en gestión de tareas

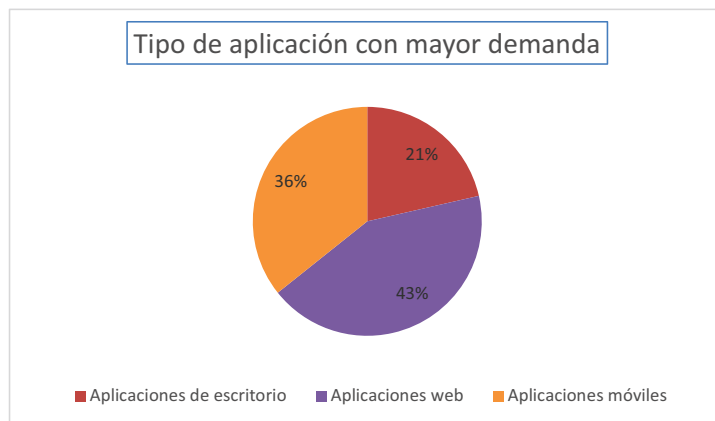
Alcance del proyecto	Proyectos grandes	Proyectos cortos	Proyectos cortos y medianos	Proyectos pequeños y medianos	Proyectos medianos y grandes	Aplicable a cualquier tamaño	Proyectos medianos y grandes
Relación cliente/proyecto	Directa	-	-	Directa	Discreta	Por medio del director de proyecto	Discreta
Ciclo de iteraciones	De 1 a 6 semanas	De 2 a 6 semanas	Sin definir	De 1 a 10 semanas	Sin definir	De 2 a 4 semanas	Sin definir

**Fuente:** elaboración propia.

La comparación ejecutada en la Tabla 3 se aplica sobre características relacionadas a la estructura y requisitos que consumen las metodologías. De forma breve, se observa cierta similitud entre las metodologías Extreme Programming, Scrum y Kanban. Sin embargo, también se encontraron disparidades en criterios del alcance como el tamaño del proyecto que se pueden aplicar y el tamaño del equipo encargado del proyecto. Esto tiene un impacto muy importante a la hora de elegir la metodología indicada puesto que en algunos casos la aplicación de un marco de trabajo puede resultar altamente costosa a pesar de ser un proyecto pequeño.

Por otra parte, se considera también los resultados de la investigación de campo aplicado a un grupo selecto de empresas desarrolladoras de software dentro de la provincia de El Oro según el último censo de actividad económica realizado por el INEC (Instituto Nacional de Estadística y Censos (INEC), s. f.).

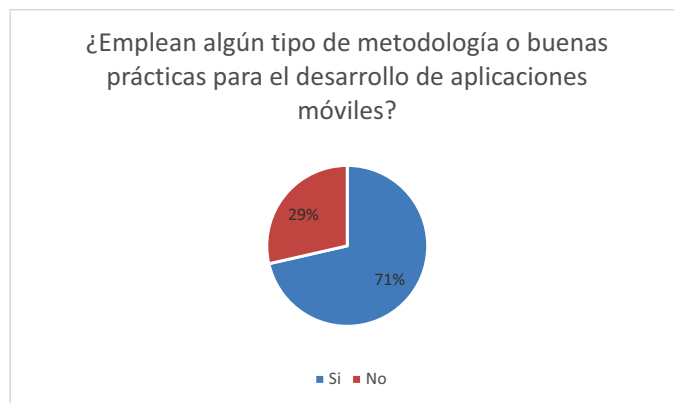
En la Ilustración 7 se evidencia la distribución del tipo de software requerido por usuarios finales a las empresas desarrolladoras. El 43% pertenece a la demanda de software web, mientras que el 36% pertenece a las solicitudes por aplicaciones móviles y por último el 21% corresponde a las peticiones por aplicaciones de escritorio.



**Ilustración 7.** Tipo de aplicación con mayor demanda en el mercado.

**Fuente:** elaboración propia.

En la Ilustración 8 se corrobora la tendencia por el uso de metodologías para el desarrollo de software, el 71% afirman que emplean marcos de trabajo o hacen uso de buenas prácticas para el desarrollo de software, mientras que el 29% mencionaron que no aplican ningún tipo de metodología para el desarrollo de proyectos de software.

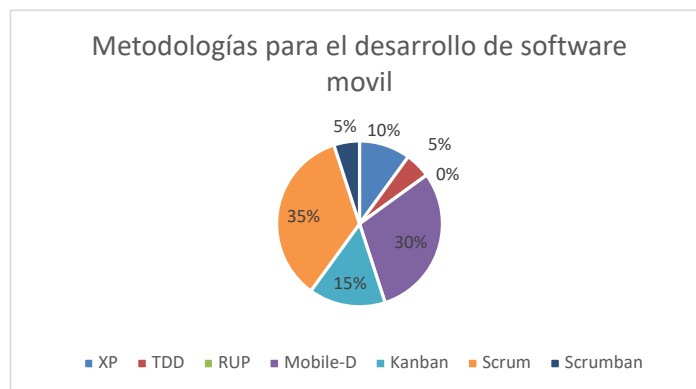


**Ilustración 8.** Aplicación de metodologías en proyectos de software.

**Fuente:** elaboración propia.



En cuanto a la Ilustración 9, el 35% del objeto de estudio se inclinan por el uso de la metodología Scrum para el desarrollo de aplicaciones móviles, el 30% prefiere emplear Mobile-D, el 15% emplea Kanban, el 10% utiliza XP, mientras que solo el 5% respectivamente optan por usar TDD y Scrumban.



**Ilustración 9.** Uso de metodologías.

**Fuente:** elaboración propia.

## 5. DISCUSIÓN

En base a la investigación bibliográfica y comparación de las características y directrices entre las metodologías como XP, TDD, RUP, Mobile-D, KANBAN, Scrum y Scrumban; se puede inferir que la metodología Scrum podría ser el marco de trabajo ideal para llevar a cabo los proyectos de desarrollo de aplicaciones móviles porque contribuye con elementos y procesos que maximizan la productividad y reduce el coste de ejecución sobre el proyecto. A diferencia de otras metodologías como XP que emplea el mismo principio ágil y abarca características similares, Scrum se enfoca más a la gestión del proyecto por ende su estructura es sólida y fiable.

Sin embargo, en el estudio de campo se observó una variación en cuanto a la preferencia de las empresas desarrolladoras de software a seleccionar la metodología para la producción de aplicaciones móviles. En este caso Scrum y Mobile-D son mayormente empleadas por las empresas, esto podría ser debido a las

características de cada una en cuanto a la adaptabilidad y enfoque sobre los procesos. Cabe resaltar que Mobile-D es un marco de trabajo orientado específicamente a la construcción de aplicaciones móviles por tal motivo genera cierta confianza para su ejecución en proyecto de este tipo.

## 6. CONCLUSIONES

El desarrollo de software de forma general necesita de directrices que orienten a los desarrolladores durante el ciclo de un proyecto para garantizar la calidad del producto final. En el caso de la producción de aplicaciones móviles, existen escasas metodologías enfocadas a este ámbito por diferentes razones, generalmente son proyectos pequeños, no siempre requiere de una cantidad considerable de integrantes, son de corto plazo, entre otras. Sin embargo, las metodologías híbridas satisfacen la necesidad de encontrar un marco de trabajo para el desarrollo de aplicaciones móviles.

En el estudio comparativo desarrollado se pudo cristalizar las diferencias entre las metodologías que a pesar de sustentarse en los principios ágiles mantienen un enfoque distinto en busca de resolver los enigmas que presentan otras metodologías. Esto no demuestra que sean incorrectas, al contrario, son bastante útiles para escenarios específicos para la que han sido desarrolladas.

Por otra parte, Scrum es una metodología versátil a diferentes contextos de aplicación, sus características iterativas, los elementos que emplea para dar soporte a sus procesos internos, estructura y enfoque la hacen ideal para la producción de aplicaciones móviles y su ciclo de vida.

## 7. REFERENCIAS BIBLIOGRÁFICAS

**Aamir, M., y Khan, M. N. A.** (2017). Incorporating quality control activities in scrum in relation to the concept of test backlog. *Sādhanā*, 42(7), 1051-1061. <https://doi.org/10.1007/s12046-017-0688-7>

- Albers, A., Heimicke, J., Trost, S., y Spadinger, M.** (2020). Alignment of the change to agile through method-supported evaluation of agile principles in physical product development. *Procedia CIRP*, 91, 600-614. <https://doi.org/10.1016/j.procir.2020.02.218>
- Bhardwaj, K., Saunders, M., Juneja, N., y Gavrilovska, A.** (2019). Serving Mobile Apps: A Slice at a Time. *Proceedings of the Fourteenth EuroSys Conference 2019*, 1–15. <https://doi.org/10.1145/3302424.3303989>
- Borle, N. C., Fegghi, M., Stroulia, E., Greiner, R., y Hindle, A.** (2018). Analyzing the effects of test driven development in GitHub. *Empirical Software Engineering*, 23(4), 1931-1958. <https://doi.org/10.1007/s10664-017-9576-3>
- Chaouch, S., Mejri, A., y Ghannouchi, S. A.** (2019). A framework for risk management in Scrum development process. *Procedia Computer Science*, 164, 187-192. <https://doi.org/10.1016/j.procs.2019.12.171>
- 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>
- da Silva Estácio, B. J., y Prikladnicki, R.** (2015). Distributed Pair Programming: A Systematic Literature Review. *Information and Software Technology*, 63, 1-10. <https://doi.org/10.1016/j.infsof.2015.02.011>
- Heikkilä, V. T., Paasivaara, M., y Lassenius, C.** (2016). Teaching university students Kanban with a collaborative board game. *Proceedings of the 38th International Conference on Software Engineering Companion*, 471–480. <https://doi.org/10.1145/2889160.2889201>
- Ho, M. H.-W., y Chung, H. F. L.** (2020). Customer engagement, customer equity and repurchase intention in mobile apps. *Journal of Business Research*, 121, 13-21. <https://doi.org/10.1016/j.jbusres.2020.07.046>

- Instituto Nacional de Estadística y Censos (INEC).** (s. f.). *Censo Nacional Económico*. Instituto Nacional de Estadística y Censos. <https://www.ecuadorencifras.gob.ec/censo-nacional-economico/>
- Krishnaiyer, K., y Chen, F. F.** (2017). A Cloud-based Kanban Decision Support System for Resource Scheduling & Management. *Procedia Manufacturing*, 11, 1489-1494. <https://doi.org/10.1016/j.promfg.2017.07.280>
- Kroener, I., Barnard-Wills, D., y Muraszkievicz, J.** (2019). Agile ethics: An iterative and flexible approach to assessing ethical, legal and social issues in the agile development of crisis management information systems. *Ethics and Information Technology*. <https://doi.org/10.1007/s10676-019-09501-6>
- Liskin, O., Schneider, K., Fagerholm, F., y Münch, J.** (2014). Understanding the role of requirements artifacts in kanban. *Proceedings of the 7th International Workshop on Cooperative and Human Aspects of Software Engineering*, 56–63. <https://doi.org/10.1145/2593702.2593707>
- 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>
- McIlroy, S., Ali, N., y Hassan, A. E.** (2016). Fresh apps: An empirical study of frequently-updated mobile apps in the Google play store. *Empirical Software Engineering*, 21(3), 1346-1370. <https://doi.org/10.1007/s10664-015-9388-2>
- Nanthaamornphong, A., y Carver, J. C.** (2017). Test-Driven Development in scientific software: A survey. *Software Quality Journal*, 25(2), 343-372. <https://doi.org/10.1007/s11219-015-9292-4>
- Owoseni, A., y Twinomurinzi, H.** (2018). Mobile apps usage and dynamic capabilities: A structural equation model of SMEs in Lagos, Nigeria. *Telematics and Informatics*, 35(7), 2067-2081. <https://doi.org/10.1016/j.tele.2018.07.009>

- Pandey, M., Litoriya, R., y Pandey, P.** (2019). Novel Approach for Mobile Based App Development Incorporating MAAF. *Wireless Personal Communications*, 107(4), 1687-1708. <https://doi.org/10.1007/s11277-019-06351-9>
- Reyes-Delgado, P. Y., Mora, M., Duran-Limon, H. A., Rodríguez-Martínez, L. C., O'Connor, R. V., y Mendoza-Gonzalez, R.** (2016). The strengths and weaknesses of software architecture design in the RUP, MSF, MBASE and RUP-SOA methodologies: A conceptual review. *Computer Standards & Interfaces*, 47, 24-41. <https://doi.org/10.1016/j.csi.2016.02.005>
- Schaefer, P., y Söllner, D.** (2017). DevOps by Scrumban. *HMD Praxis der Wirtschaftsinformatik*, 54(2), 251-260. <https://doi.org/10.1365/s40702-017-0301-x>
- Snyder, H.** (2019). Literature review as a research methodology: An overview and guidelines. *Journal of Business Research*, 104, 333-339. <https://doi.org/10.1016/j.jbusres.2019.07.039>
- 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>
- Steghöfer, J.-P., Burden, H., Alahyari, H., y Haneberg, D.** (2017). No silver brick: Opportunities and limitations of teaching Scrum with Lego workshops. *Journal of Systems and Software*, 131, 230-247. <https://doi.org/10.1016/j.jss.2017.06.019>