

**UNIVERSIDAD PERUANA UNIÓN**  
FACULTAD DE INGENIERÍA Y ARQUITECTURA  
Escuela Profesional de Ingeniería de Sistemas



*Una Institución Adventista*

**Análisis comparativo de patrones de diseño de software  
para el desarrollo de aplicaciones móviles de calidad:  
Una revisión sistemática de la literatura**

Por:

Jesús Alberto Abanto Cruz

Omar Fernando Gonzales Ramírez

Asesor:

Fernando Asín Gómez

**Lima. Diciembre del 2019**

## DECLARACIÓN JURADA DE AUTORÍA DEL TRABAJO DE INVESTIGACIÓN

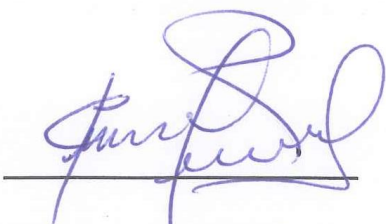
Fernando Asín Gómez, de la Facultad de Ingeniería y Arquitectura, Escuela Profesional de Ingeniería de Sistemas, de la Universidad Peruana Unión.

DECLARO:

Que el presente trabajo de investigación titulado: "ANÁLISIS COMPARATIVO DE PATRONES DE DISEÑO DE SOFTWARE PARA EL DESARROLLO DE APLICACIONES MÓVILES DE CALIDAD: UNA REVISIÓN SISTEMÁTICA DE LA LITERATURA" constituye la memoria que presentan los estudiantes Jesús Alberto Abanto Cruz y Omar Fernando Gonzales Ramírez para aspirar al grado de bachiller en Ingeniería de Sistemas, cuyo trabajo de investigación ha sido realizado en la Universidad Peruana Unión bajo mi dirección.

Las opiniones y declaraciones en este trabajo de investigación son de entera responsabilidad del autor, sin comprometer a la institución.

Y estando de acuerdo, firmo la presente declaración en Lima, a los 04 días de diciembre del año 2019



Fernando Asín Gómez

Análisis comparativo de patrones de diseño de software para el desarrollo de aplicaciones móviles de calidad: Una revisión sistemática de la literatura

# Trabajo de investigación

Presentado para optar al grado de bachiller en Ingeniería de Sistemas

## JURADO CALIFICADOR



Mg. Sergio Omar Valladares Castillo  
Presidente



Mg. Omar Leonel Loaiza Jara  
Secretario



Ing. Diana Lidia Sánchez Torpoco  
Vocal



Ing. María Antonieta Vidalón Aliaga  
Vocal



Mg. Fernando M. Asín Gómez  
Asesor

Lima, 02 de diciembre del 2019

# “Análisis comparativo de patrones de diseño de software para el desarrollo de aplicaciones móviles de calidad: Una revisión sistemática de la literatura”

Omar Gonzales Ramírez

Jesús Abanto Cruz

<sup>1</sup> Universidad Peruana Unión, PERU

<sup>2</sup> Lima, Perú

fernandogonzales@upeu.edu.pe

<sup>1</sup> Universidad Peruana Unión, PERU

<sup>2</sup> Lima, Perú

jesusabanto@upeu.edu.pe

**Resumen**— En este artículo de revisión se realizó un estudio cuyo propósito es el de encontrar los principales estudios sobre patrones de diseño de software para el desarrollo de aplicaciones móviles de calidad, para posteriormente determinar criterios de identificación que servirán como herramienta de selección de patrones de diseño de calidad. Para este estudio se realizó una revisión sistemática de la literatura en 3 bases de datos reconocidas (IEEE, Explorer, EBSCO). De un total de 3072 artículos encontrados, se identificaron 16, que debido a su gran relevancia con el tema de estudio se seleccionaron para su posterior revisión. Se utilizó como instrumento principal la estrategia PICO para hacer la búsqueda y selección de los artículos de interés según el propósito de este estudio. Los resultados indican que son 5 los principales patrones de diseño de software para el desarrollo móvil de calidad y que existen ciertos criterios de identificación para su análisis. Se puede concluir que gracias a estos criterios de identificación podemos obtener una herramienta de selección para comparar los principales patrones de diseño de software para el desarrollo de aplicaciones móviles de calidad, y finalmente poder implementar la más conveniente según las necesidades del proyecto.

**Palabras claves**— Patrones de diseño; Aplicaciones móviles; Eficiencia; Usabilidad; Desarrollo.

**Abstract**— In this review article a study was carried out whose purpose is to find the main studies and software design patterns for the development of quality mobile applications, then determine identification criteria that will serve as a tool for selecting design patterns for quality. For this study, a systematic review of the literature was carried out in 3 recognized databases (IEEE, Explorer, EBSCO). Of a total of 3072 articles found, 16 were identified, which due to their great relevance to the subject of study were selected for later review. The PICO strategy was used as the main instrument to search and select the articles of interest according to the purpose of this study. The results indicate that there are 5 main patterns of software design for quality mobile development and that there are certain identification criteria for analysis. It can be concluded that thanks to these identification criteria we can obtain a selection tool to compare the main software design patterns for the development of quality mobile applications, and finally be able to implement the most convenient one according to the needs of the project.

**Keywords**— Design patterns; Mobile apps; Efficiency; Usability; Developing.

## I. INTRODUCCIÓN

Los dispositivos móviles con sus funcionalidades sofisticadas y sus aplicaciones han cambiado la vida de las personas y la industria tecnológica, a consecuencia, hoy en día hay muchas organizaciones e individuos inclinándose hacia el desarrollo de aplicaciones móviles. Para incursionar en la implementación de estas novedosas aplicaciones es importante entender cómo funcionan los patrones de diseño de software y el paradigma POO (programación orientado a objetos). Los patrones de diseño orientados a objetos (OO) son un elemento tecnológico que representan soluciones comunes para problemas de diseño dados en el desarrollo de software[1]. Las buenas prácticas de diseño Orientado a Objetos son reutilizables, extensibles y mantenibles. Los patrones nos muestran cómo construir sistemas con buenas cualidades de diseño OO[2].

En el inmenso mundo de las aplicaciones móviles existen diversos patrones de diseño que nos brindan facilidades de crear un desarrollo más eficiente y organizado. El objetivo de la revisión es el de encontrar los principales estudios sobre patrones de diseño de software para el desarrollo de aplicaciones móviles de calidad, y posteriormente determinar criterios de identificación que servirán como herramienta de selección de patrones de diseño de calidad.

Este artículo está distribuido de la siguiente manera: la sección II presenta el marco conceptual; la sección III describe la revisión sistemática de la literatura; la sección IV presenta los resultados de la revisión sistemática de la literatura y la sección V describe las conclusiones.

## II. MARCO CONCEPTUAL

En esta sección se presentan algunas definiciones del contexto sobre el cual se realiza el estudio y el objeto de análisis.

### A. Paradigma POO (Programación orientado a objetos)

Los objetos existen en el mundo real. Los objetos están clasificados, especificados, organizados, acoplado, implementados y manipulados, dándose la facilidad de aplicar estas características de individualidad y facilidad de manipulación. Casi todos los principales lenguajes de programación están orientados a objetos, es por eso que es importante entender cómo funciona el paradigma POO [3].

POO es una manera de estructurar el código para una óptima legibilidad, uso y mantenimiento. POO se caracteriza con siguientes términos [4]:

- **Objeto:** Un objeto, en informática, es una unidad que consta de comportamientos y propiedades, y como tal, es un valor en memoria referenciado por un identificador.
- **Clases:** Es la implementación de un tipo de objeto. Determina la estructura de datos y los métodos operacionales aprobados que se aplican a cada uno de sus objetos.
- **Métodos:** Describe la forma en la que los datos de un objeto son controlados. Los métodos solo hacen referencia a la estructura de datos en un tipo de objeto. No deben acceder directamente a la estructura de datos de otro objeto.
- **Peticiones:** Solicita una operación específica y debe ser llamada usando uno o varios objetos como parámetros.

Una vez entendiéndose lo fundamental del paradigma Orientados a Objetos, es de real interés conocer que existen cinco conceptos que lo diferencian con respecto a la ingeniería de software convencional [1]:

- **Abstracción:** La abstracción es una las llaves clave de la programación orientado a objetos. Su principal objetivo es controlar la complejidad ocultados detalles innecesarios para el usuario.

- **Encapsulamiento:** Es el proceso de combinar datos y funciones en un solo concepto llamado clase. En la encapsulación no se accede directamente a un dato, sino a través de las funciones presentes dentro de la clase. En términos simples los atributos en la clase se mantienen privados y públicos dependiendo como se definan.
- **Modularidad:** La modularidad está relacionada con la encapsulación, y es una forma de mapear abstracciones encapsuladas en módulos físicos reales.
- **Polimorfismo:** Se refiere a procesar objetos de manera diferente según su tipo de datos, es decir, un método con implementación múltiple para una determinada a clase de acción.
- **Herencia:** Se refiere al mecanismo por el cual un objeto adquiere algunas o todas las propiedades de objeto.

### B. Patrones de Diseño

Los patrones de diseño tienen una gran variedad de usos y se han utilizado y probado en la práctica. Se ha demostrado que son eficaces en el desarrollo de software para simplificar el diseño general de las aplicaciones. Los patrones de diseño hacen que el software sea más reutilizable, lo que puede reducir el costo de producción y el tiempo de desarrollo. Los patrones de diseño son muy útiles para los desarrolladores y diseñadores, ya que encapsulan la experiencia, proporcionan un vocabulario común y mejoran la documentación de los diseños de software [1].

Los patrones de diseño han tenido influencia masiva en el desarrollo de software. Al igual que las aplicaciones web, la implementación de aplicaciones móviles también estableció algunos patrones probados y estándares para superar los retos y limitaciones del desarrollo móvil. La mayoría de las aplicaciones móviles se desarrollaron con código de baja calidad y no se basan en patrones de diseño arquitectónico. Los desarrollos de una aplicación móvil con el patrón de diseño correcto pueden sincronizar efectivamente la interfaz de usuario con modelos de datos y lógica empresarial, esto influirá en cómo se debe ver su código fuente. Existen diversos patrones de diseño de arquitectura para el desarrollo móvil. En la tabla I mostramos los siguientes patrones de diseño para aplicaciones móviles [5].

TABLA I  
PATRONES DE DISEÑOS USADO EN VARIAS PLATAFORMAS

Plataforma	Patrones de Diseño
IOS	Abstract Factory, Adapter, Factory Method, Template Method , Singleton, MVC
Android	MVC, Model View View-Model(MVVM), Model-View-Presenter(MVP), BLOC, Viper
Windows Phone	Patterns such as Factory Method, Template Method, Command, Observer, MVC

Fuente: H. J. La and S. D. Kim, "Balanced MVC architecture for developing Service-based Mobile Applications.

### C. Proceso de desarrollo de software

Es la descripción de una secuencia de actividades que deben ser seguida por un equipo de trabajadores para generar un conjunto coherente de productos, uno de los cuales es el programa del sistema deseado. El objetivo básico del proceso es hacer predecible el trabajo que se requiere [6]:

- Mantener un nivel de calidad.
- Predecir el tiempo de desarrollo.
- Predecir el costo.

### D. Naturaleza de las aplicaciones de software:

No existe un proceso de desarrollo universal. Debe configurarse de acuerdo con la naturaleza del producto y de la experiencia de la empresa. Tipos de aplicaciones [6]:

- Aplicaciones Monoprocesadoras: Se ejecutan en un solo computador. No se comunica con otras aplicaciones.

- Ej. Procesador de texto.
- Aplicaciones Embebidas: Se ejecuta en un entorno computarizado especial. Requiere co-diseño de hardware/software. Ejemplo: Teléfono móvil.
- Aplicaciones de Tiempo Real: Tiene entre sus especificaciones requerimiento temporales naturaleza reactiva. Ejemplo: Software de radar.
- Aplicaciones Distribuidas: Se ejecuta en múltiples procesadores. Requiere intercomunicación a través de la red. Ejemplo: Aplicaciones de red.

### III. DEFINICION DE LA REVISION SISTEMATICA DE LA LITERATURA

La metodología de investigación conocida como la Revisión Sistemática de la Literatura (RSL), ha sido desarrollada para recolectar y analizar toda la información y evidencia disponible acerca del objeto de estudio, se valida con el hecho de adquirir información relevante y actualizada con considerable grado de fiabilidad, no teniendo que leer gran cantidad de artículos, debido a que la metodología, brinda una serie de pasos que ayuda optimizar la investigación, haciéndolas muchas más prácticas [8].

#### A. *Identificar la necesidad de la realización*

La revisión sistemática de la literatura que se establece en este trabajo se origina mediante el requerimiento o necesidad de identificar qué patrones de diseño de arquitectura existen para implementar proyectos móviles seguros, de calidad. Así mismo se requiere identificar (i) ¿Qué criterios existen para la ejecución de análisis comparativo de patrones de diseño de software en aplicaciones móviles de calidad?, (ii) ¿Cuáles son los patrones más frecuentes que aportan alto valor de seguridad en datos sensibles que se gestionan en las aplicaciones móviles?, (iii) ¿Cuáles son los patrones de diseño más frecuentes en el desarrollo de aplicaciones móviles?

TABLA II  
ELABORACIÓN DEL OBJETIVO DE LA INVESTIGACIÓN

Campos/Criterios	Valor
Objeto de estudio	Análisis Comparativo.
Propósito	Identificar
Foco	Patrones
Involucrados	Industria de Software, seguridad informática, desarrollos de software, desarrollo móvil.
Factores del contexto	Ninguno.

#### B. *Preguntas para la revisión sistemática*

Para la definición y estructuración de las preguntas de investigación se tomó como referencia la finalidad de la investigación expuesta en la sección anterior. En la siguiente Tabla III se muestran las preguntas de investigación y en la tabla IV las preguntas bibliométricas.

TABLA III  
PREGUNTAS DE INVESTIGACIÓN Y MOTIVACIÓN

ID	Pregunta	Motivación
PI-01	¿Qué criterios existen para la ejecución de análisis comparativo de patrones de diseño de software en aplicaciones móviles de calidad?	Identificar criterios que permitan determinar las características de patrones de diseño confiables.
PI-02	¿Cuáles son los patrones más frecuentes que aportan alto valor de seguridad en datos sensibles que se gestionan en las aplicaciones móviles?	Identificar patrones de diseño, métodos y funciones que aporten al desarrollo de un sistema seguro.
PI-03	¿Cuáles son los patrones de diseño más frecuentes en el desarrollo de aplicaciones móviles?	Determinar los patrones de diseño actuales más usados en el ámbito de desarrollo de software móvil.

TABLA IV  
PREGUNTAS BIBLIOMÉTRICAS Y MOTIVACIÓN

ID	Pregunta	Motivación
PB-01	¿Qué cantidad de publicaciones existen por tipo de artículo?	Determinar la cantidad de estudios publicados por tipo de artículo para identificar la concentración de los mismos.
PB-02	¿Cómo ha evolucionado en el tiempo la frecuencia de las publicaciones sobre este tema?	Identificar la frecuencia de las publicaciones para poder establecer la relevancia del tema en el tiempo.

### C. Cadena de búsqueda.

La estrategia elegida para la elaboración de la cadena de búsqueda fue la estrategia PICO, a continuación se muestran las cadenas de búsqueda para la selección de resultados basado en esta estrategia [9].

#### *Población:*

##### Entidad:

- Término principal 1: Patrones de diseño, Calidad,
- Términos alternos: Arquitectura de software, Seguridad
- Término principal 2: análisis comparativo
- Términos alternos: comparación, estudio comparativo

#### *Intervención:*

##### Entidad:

- Término principal 1: Análisis comparativo.
- Términos alternos: Diferencias
- Término principal 2: Desarrollo de software móvil.
- Términos alternos: Aplicativo móvil.

#### *Comparación:*

- No aplica para este estudio.

#### *Resultados:*

##### Entidad:

- Término principal 1: Implementación.



- Términos alternos: Aplicación.

TABLA V  
CADENAS DE BÚSQUEDA

CONTEXTO	TÉRMINOS
Población	(DESIGN PATTERNS* OR SOFTWARE ARCHITECTURE*)AND(QUALITY* OR SECURE*)
Intervención	(COMPARATIVE ANALYSIS* OR DIFERENCES)AND(MOBILE SOFTWARE DEVELOPMENT* OR MOBILE ALPICATIONS*)
Comparación	No aplica.
Resultado	(IMPLEMENTACIÓN* OR APLICACION*)
Contexto	No aplica

#### D. Criterios de inclusión y exclusión

Luego de ejecutar la cadena de búsqueda en diferentes librerías indexadas, los resultados deben ser sometidos a evaluación para poder determinar cuáles son los estudios primarios que responden directamente las preguntas de investigación formuladas [10].

Se tomó en consideración los siguientes criterios para la evaluación de los estudios:

##### Criterios de Inclusión:

- C.I.1 Se consideran todos aquellos artículos provenientes de librerías digitales indexadas.
- C.I.2 Los artículos deben provenir del área de Ingeniería de Software.
- C.I.3 Se aceptarán artículos que contengan estudios o análisis comparativos de patrones de arquitectura de software.
- C.I.4 Se considerarán todos los artículos que se encuentren dentro del rango de temporalidad definido (2010-2019).
- C.I.5 Se aceptarán artículos provenientes de revistas científicas y conferencias.

##### Criterios de Exclusión:

- C.E.1 Serán excluidos los artículos duplicados.
- C.E.2 Serán rechazados los artículos que no se encuentren en idioma inglés.
- C.E.3 Serán rechazados los artículos de contenido similar, quedándose solo los que tengan el contenido más completo.
- C.E.4 Serán excluidos los estudios secundarios, estudios terciarios y resúmenes.
- C.E.5 Serán excluidos los artículos cuyo título no tenga relación con el objeto de estudio

TABLA VI  
PROCEDIMIENTOS Y CRITERIOS DE INCLUSIÓN

PROCEDIMIENTO	CRITERIO DE SELECCIÓN
Paso 1	C.I.1, C.I.4, C.E.2
Paso 2	C.I.1, C.I.3, C.E.2
Paso 3	C.I.5, C.I.1, C.I.3, C.E.5
Paso 4	C.I.5, C.I.2, C.E.2, C.E.3, C.E.4

#### IV. RESULTADOS

##### A. Resultados de la búsqueda

Para la búsqueda en las bases de datos se emplea la cadena de búsqueda realizada anteriormente:

(DESIGN PATTERNS\* OR SOFTWARE ARCHITECTURE\*) AND (QUALITY\* OR SECURE\*) AND (COMPARATIVE ANALYSIS\* OR DIFERENCES\*) AND (MOBILE SOFTWARE DEVELOPMENT\* OR MOBILE APLICACIONES\*) AND (IMPLEMENTATION\* OR APLICACION\*)

##### B. Resultados de filtros aplicados

Los resultados obtenidos por cada base de datos, pasaron por 4 pasos en las que se aplican los criterios de Inclusión y Exclusión para filtrar los artículos de interés en relación a los objetivos de esta investigación. En la tabla VII se muestran los resultados de la selección de estudio y en la tabla VIII se listan todos los artículos resultantes de dicha selección.

TABLA VII  
RESULTADOS DE LA BÚSQUEDA.

Base de datos	Artículos descubiertos	Paso 1	Paso 2	Paso 3	Paso 4
IEEE Xplore	634	42	23	12	2
SCIENCE DIRECT	895	425	63	21	8
EBSCO	1543	556	68	42	6
	3072	1023	154	75	16

TABLA VIII  
ARTÍCULOS RESULTANTES.

Id	Biblioteca	Título	Año	Tipo de documento	Autor(es)
1	IEEE	Design patterns in object-oriented analysis and design	2011	Conference	Jiang, Shuai Mu, Huaxin [11]
2	IEEE	Design patterns in software development	2011	Conference	Mu, Huaxin Jiang, Shuai [12]
3	SCIENCE DIRECT	Automated framework for classification and selection of software design patterns	2019	Conference	Hussain, Shahid Keung, Jacky Sohail, Muhammad Khalid Khan, Arif Ali Ilahi, Manzoor [13]
4	SCIENCE DIRECT	Software design patterns classification and selection using text categorization approach	2017	Conference	Hussain, Shahid Keung, Jacky Khan, Arif Ali [14]
5	SCIENCE DIRECT	How have we evaluated software pattern application? A systematic mapping study of research design practices	2015	Conference	Riaz, Maria Breaux, Travis Williams, Laurie [15]
6	SCIENCE DIRECT	An empirical investigation on the reusability of design patterns and software packages	2011	Journal	Ampatzoglou, Apostolos Kritikos, Apostolos Kakarontzas, George Stamelos, Ioannis [16]
7	SCIENCE DIRECT	Supporting multi-view development for mobile applications	2019	Journal	Barnett, Scott Avazpour, Iman Vasa, Rajesh Grundy, John [17]
8	SCIENCE DIRECT	Source code and design conformance, design pattern detection from source code by classification approach	2015	Journal	Chihada, Abdullah Jalili, Saeed

					Hasheminejad, Seyed Mohammad Hossein Zangoeei, Mohammad Hossein [18]
9	SCIENCE DIRECT	Research patterns and trends in software effort estimation	2017	Journal	Sehra, Sumeet Kaur Brar, Yadwinder Singh Kaur, Navdeep Sehra, Sukhjit Singh [19]
10	SCIENCE DIRECT	Software engineering process models for mobile app development: A systematic literature review	2018	Review articles	Jabangwe, Ronald Edison, Henry Duc, Anh Nguyen [20]
11	EBSCO	Usability Studies on Mobile User Interface Design Patterns: A Systematic Literature Review	2017	Conference	Punchoojit, Lumpapun Hongwarittorrn, Nuttanont [21]
12	EBSCO	A catalogue associating security patterns and attack steps to design secure applications	2019	Conference	Salva, Sébastien Regainia, Loukmen [22]
13	EBSCO	A general framework to detect behavioral design patterns	2018	Conference	Liu, Cong Van Dongen, Boudewijn Assy, Nour Van Der Aalst, Wil M.P. [23]
14	EBSCO	MMVMi: A validation model for MVC and MVVM design patterns in iOS applications	2018	Conference	Aljamea, Mariam Alkandari, Mohammad [24]
15	EBSCO	Investigating the effect of design patterns on energy consumption	2017	Conference	Feitosa, Daniel Alders, Rutger Ampatzoglou. [25]
16	EBSCO	An empirical basis for the use of design patterns by architects in parametric design	2016	Review articles	Yu, Rongrong Gero, John S. [26]

### C. Análisis Bibliométrico

#### 1. Pregunta de Bibliometría 1 (PB-01)

¿Qué cantidad de publicaciones existen por tipo de artículo?

En la Fig.1 en la cantidad de publicaciones por tipo de artículo en base a los 16 artículos resultantes, se puede observar que los artículos de conferencia (conference), con 10 artículos resultantes, representan el 62,5% del total de los artículos seleccionados, seguidamente los artículos de revista (Journal), con 4 artículos resultantes, representan el 25 % del total de los artículos; y por último el 12.5% restante, con 2 artículos resultantes, representan a los artículos de revisión. (review articles). Esta pregunta bibliométrica concluye que las conferencias y artículos de investigación son las más utilizadas para el análisis.

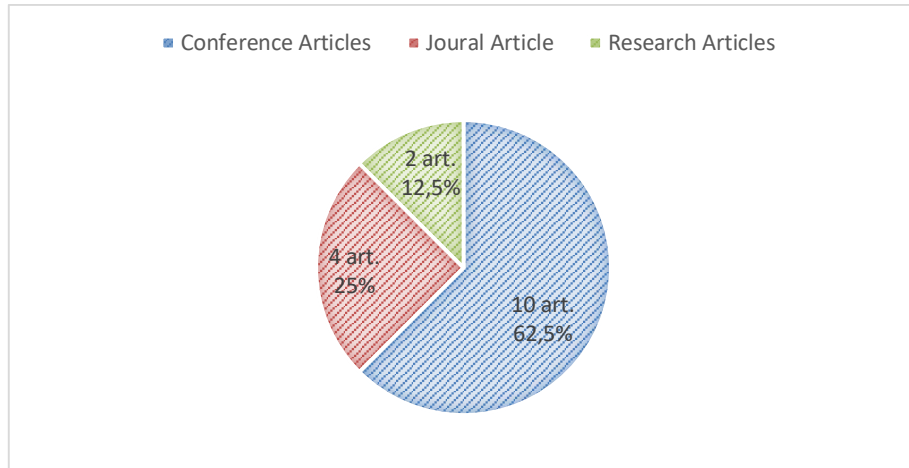


Fig. 1 Cantidad de publicaciones por tipo de artículos

## 2. Pregunta de Bibliometría 2 (PB-02)

¿Cómo ha evolucionado en el tiempo la frecuencia de publicaciones sobre este tema?

En la Fig.2 podemos observar que el total de 16 artículos, hubo un incremento en el número de publicaciones a partir del año 2016 en adelante; 62.5% del total ha sido publicado a lo largo de los 3 últimos años, y el 37.5% del total han sido publicado entre el 2011 y 2016.

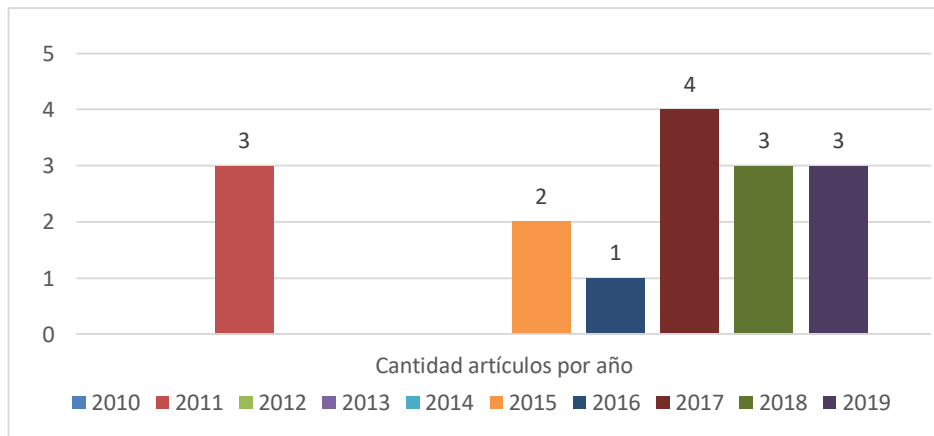


Fig. 2 Publicaciones de artículo por año

D. Sintetizar los datos extraídos

## 1. Pregunta de Investigación 1 (PI-01)

¿Qué criterios existen para la ejecución de análisis comparativo de patrones de diseño de software en aplicaciones móviles de calidad?

A través de la extracción de información de los artículos, siguiendo el enfoque del análisis orientado a objetos para estudiar los patrones de diseño existentes, se pudo encontrar diversas características que definen a un patrón de diseño confiable. El formato de los patrones de diseño que se muestra en la Tabla IX es una plantilla que describe las características de los patrones de diseño, propuesto por Gamma E. y Helm R. [4].

TABLA IX  
DISEÑO DE PATRONES Y FORMATOS

Partes	Descripción
Nombre de Patrón	Nombre del patrón de diseño
Intención	Los objetivos o propósitos del patrón y el problema que resuelve.
Motivación	Un escenario específico inconsistente puede resolverse mediante los patrones de diseños
Aplicación	Describe una situación compleja
Estructura	Representación o marco de la estructura del proyecto
Participaciones	Indica la clase y objetos
Colaboración	Indica la colaboración de los participantes
Implementación	Se desarrolló el patrón de diseño

Fuente: Gamma E, Helm R, Johnson R, Design Patterns: Elements of Reusable Object-Oriented Software.

Los patrones de diseño identificados deben ser aplicables en el desarrollo de un software móvil basadas en tres atributos de calidad, los cuales, según F. E. Shahbudin [3], son:

- 1) Eficiencia: la capacidad de la aplicación para exhibir el rendimiento requerido con respecto a la cantidad de recursos necesarios.
- 2) Usabilidad: las propiedades de usabilidad que muestran la facilidad de uso de la aplicación.
- 3) Reutilización: la medida en que los patrones de diseño podrían reutilizarse para una nueva aplicación dentro de un dominio similar.

Bajo estos criterios podemos identificar las características de cada patrón de diseño para luego poder compararlas y poder así implementar la más conveniente según las necesidades del proyecto.

## 2. Pregunta de Investigación 2 (PI-02)

¿Cuáles son los patrones más frecuentes que aportan alto valor de seguridad en datos sensibles que se gestionan en las aplicaciones móviles?

Se pudo reconocer que, en los artículos de investigación resultantes, se mencionan principalmente a estos 5 patrones de diseño: MVC extendido, Model View View-Model(MVVM), Model-View-Presenter(MVP), BLOC, Viper, como los patrones más confiables y con mayor valor de seguridad respecto al manejo de datos sensibles, debido a su estrecha convivencia con la filosofía CLEAN, un conjunto de buenas prácticas que propone un desarrollo limpio, código legible y muy bien estructurado [27].

### 3. Pregunta de Investigación 3 (PI-03)

¿Cuáles son los patrones de diseño más frecuentes en el desarrollo de aplicaciones móviles?

En la tabla X mostramos los siguientes patrones de diseño más frecuentes para aplicaciones móviles según H. Jung y Kim S. D. [5].

TABLA X  
PATRONES DE DISEÑO MÁS FRECUENTES EN EL DESARROLLO DE APLICACIONES MÓVILES.

Plataforma	Patrones de Diseño
IOS	Abstract Factory, Adapter, Factory Method, Template Method, Chain of Responsibility, Command, Observer, Singleton, MVC extendido
Android	MVC extendido, Model View View-Model(MVVM), Model-View-Presenter(MVP), BLOC, Viper
Windows Phone	Patterns such as Factory Method, Template Method, Command, Observer, MVC

### V. CONCLUSIÓN

Los patrones de diseño han mostrado un mayor impacto en el desarrollo de software clásico. Dado que los patrones de diseño proporcionan soluciones comprobadas, se implementan en el desarrollo de aplicaciones móviles para superar las limitaciones y los desafíos que esta supone. El desarrollo de aplicaciones móviles puede ser un proceso tedioso, ya que cada aplicación debe pasar por los ciclos de desarrollo para garantizar que la aplicación se ajuste a los atributos de calidad estándar. Por ello, este artículo de revisión concluye que los principales patrones de diseño de software para el desarrollo móvil de calidad son: MVC extendido, Model View View-Model(MVVM), Model-View-Presenter(MVP), BLOC, Viper; y que existen ciertos criterios de identificación para su análisis, estos criterios son atributos y características que definen a un patrón de diseño de calidad; gracias a estos criterios de identificación, el desarrollador o jefe de proyecto, puede obtener una herramienta de selección para comparar los principales patrones de diseño de software para el desarrollo de aplicaciones móviles de calidad, y finalmente poder implementar la más conveniente según las necesidades del proyecto.

### REFERENCIAS

- [1] P. Trento, G. Antonioli, R. Fiutem, and L. Cristoforetti, "Using Metrics to Identify Design Pattern in OO Software."
- [2] H. Mu and S. Jiang, "Design patterns in software development," *ICSESS 2011 - Proc. 2011 IEEE 2nd Int. Conf. Softw. Eng. Serv. Sci.*, pp. 322–325, 2011.
- [3] F. E. Shahbudin, "Design Patterns for Developing High Efficiency Mobile Application," *J. Inf. Technol. Softw. Eng.*, vol. 03, no. 03, 2013.
- [4] V. J. (2000) Gamma E, Helm R, Johnson R, *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison Wesley, USA. Addison Wesley, USA, 2000.
- [5] H. J. La and S. D. Kim, "Balanced MVC architecture for developing Service-based Mobile Applications," *Proc. - IEEE Int. Conf. E-bus. Eng. ICEBE 2010*, pp. 292–299, 2010.
- [6] J. M. Drake, "Proceso de desarrollo de aplicaciones software." pp. 0–36, 2008.

- [7] Y. Liu, "Analysis and Application of Interface Design Elements for Mobile Platform," in *Proceedings - 2016 International Conference on Smart City and Systems Engineering, ICSCSE 2016*, 2017, pp. 171–174.
- [8] C. Manterola, P. Astudillo, E. Arias, and N. Claros, "Revisiões sistemáticas de la literatura. Qué se debe saber acerca de ellas," *Cir. Esp.*, vol. 91, no. 3, pp. 149–155, 2013.
- [9] C. M. D. C. Santos, C. A. D. M. Pimenta, and M. R. C. Nobre, "A estratégia PICO para a construção da pergunta de pesquisa e busca de evidências," *Rev. Lat. Am. Enfermagem*, vol. 15, no. 3, pp. 508–511, 2007.
- [10] B. Kitchenham and S. Charters, "Guidelines for performing Systematic Literature Reviews in SE," pp. 1–44, 2007.
- [11] S. Jiang and H. Mu, "Design patterns in object oriented analysis and design," in *ICSESS 2011 - Proceedings: 2011 IEEE 2nd International Conference on Software Engineering and Service Science*, 2011, pp. 326–329.
- [12] H. Mu and S. Jiang, "Design patterns in software development," in *ICSESS 2011 - Proceedings: 2011 IEEE 2nd International Conference on Software Engineering and Service Science*, 2011, pp. 322–325.
- [13] S. Hussain, J. Keung, M. K. Sohail, A. A. Khan, and M. Ilahi, "Automated framework for classification and selection of software design patterns," *Appl. Soft Comput. J.*, vol. 75, pp. 1–20, 2019.
- [14] S. Hussain, J. Keung, and A. A. Khan, "Software design patterns classification and selection using text categorization approach," *Appl. Soft Comput. J.*, vol. 58, pp. 225–244, 2017.
- [15] M. Riaz, T. Breaux, and L. Williams, "How have we evaluated software pattern application? A systematic mapping study of research design practices," *Inf. Softw. Technol.*, vol. 65, pp. 14–38, 2015.
- [16] A. Ampatzoglou, A. Kritikos, G. Kakarontzas, and I. Stamelos, "An empirical investigation on the reusability of design patterns and software packages," *J. Syst. Softw.*, vol. 84, no. 12, pp. 2265–2283, 2011.
- [17] S. Barnett, I. Avazpour, R. Vasa, and J. Grundy, "Supporting multi-view development for mobile applications," *J. Comput. Lang.*, vol. 51, no. November 2015, pp. 88–96, 2019.
- [18] A. Chihada, S. Jalili, S. M. H. Hasheminejad, and M. H. Zangooei, "Source code and design conformance, design pattern detection from source code by classification approach," *Appl. Soft Comput. J.*, vol. 26, pp. 357–367, 2015.
- [19] S. K. Sehra, Y. S. Brar, N. Kaur, and S. S. Sehra, "Research patterns and trends in software effort estimation," *Inf. Softw. Technol.*, vol. 91, pp. 1–21, 2017.
- [20] R. Jabangwe, H. Edison, and A. N. Duc, "Software engineering process models for mobile app development: A systematic literature review," *J. Syst. Softw.*, vol. 145, no. May, pp. 98–111, 2018.
- [21] L. Punchoojit and N. Hongwarittorn, "Usability Studies on Mobile User Interface Design Patterns: A Systematic Literature Review," *Adv. Human-Computer Interact.*, vol. 2017, 2017.
- [22] S. Salva and L. Regainia, "A catalogue associating security patterns and attack steps to design secure applications," *J. Comput. Secur.*, vol. 27, no. 1, pp. 49–74, 2019.
- [23] C. Liu, B. Van Dongen, N. Assy, and W. M. P. Van Der Aalst, "A general framework to detect behavioral design patterns," *Proc. - Int. Conf. Softw. Eng.*, no. 1, pp. 234–235, 2018.
- [24] M. Aljamea and M. Alkandari, "MMVMi: A validation model for MVC and MVVM design patterns in iOS applications," *IAENG Int. J. Comput. Sci.*, vol. 45, no. 3, pp. 377–389, 2018.
- [25] D. Feitosa, R. Alders, A. Ampatzoglou, P. Avgeriou, and E. Y. Nakagawa, "Investigating the effect of design patterns on energy consumption," *J. Softw. Evol. Process*, vol. 29, no. 2, pp. 1–21, 2017.
- [26] R. Yu and J. S. Gero, "An empirical basis for the use of design patterns by architects in parametric design," *Int. J. Archit. Comput.*, vol. 14, no. 3, pp. 289–302, 2016.
- [27] R. C. Martin, *Praise for The Clean Coder*, vol. 96, no. 3. 2011.