



UNIVERSIDAD TÉCNICA DE BABAHOYO
FACULTAD DE ADMINISTRACIÓN, FINANZAS E INFORMÁTICA

PROCESO DE TITULACIÓN
NOVIEMBRE 2022 – ABRIL 2023

EXAMEN COMPLEXIVO DE GRADO O DE FIN DE CARRERA
PRUEBA PRÁCTICA
INGENIERÍA EN SISTEMAS

PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS

TEMA:

**ANÁLISIS COMPARATIVO DE LAS TECNOLOGÍAS SWIFT “IOS” Y FLUTTER
“ANDROID Y IOS” PARA EL DESARROLLO DE APLICACIONES MÓVILES**

EGRESADO:

RAMOS ALARCON ROBERTO ALEXANDER

TUTOR:

AGUIRRE RODRIGUEZ CARLOS GONZALO

AÑO 2023

SUMMARY

Currently, both are very popular technologies in the world of development for mobile applications, therefore, in a comparative study, information will be written about the Swift programming language and the Flutter framework, it can be seen that apparently they do not have the same function, through In this investigation that will be carried out, we will verify that these technologies have a main focus, which is to develop mobile applications. Swift is generally dedicated to designing within iOS, macOS, watchOS and tvOS devices. On the other hand, Flutter is made by the Google company, but it is focused on iOS and Android. This document will investigate the functions, characteristics, advantages, disadvantages, the beginnings of these technologies where it allows to know which has more ease and less slowness for the development of mobile applications, knowing the beginnings and their versions, it will also be indicated by tables Comparative the main approaches that both systems have, where when making a conclusion it becomes easier because that way the feasibility of each one of them will be observed. In general, Swift is one of the most popular programming languages, but it is only found for the Apple platform, creating high-quality applications itself, while the Flutter framework is coded by the Dart programming language, allowing it to be designed for its users. Platforms of the most used mobile operating systems in the commercial market of users.

Keywords

Technologies, Development, Platforms, Applications, Mobile.

RESUMEN

En la actualidad ambas son tecnologías muy populares en el mundo de desarrollo para aplicaciones móviles, por lo tanto en estudio comparativo se redactará información sobre el lenguaje de programación Swift y el framework de Flutter se puede observar que al parecer no tienen una misma función, mediante esta investigación que se realizará verificaremos que estas tecnologías tienen un enfoque principal, que es el de desarrollar aplicaciones móviles, por lo general Swift se dedica a diseñar dentro de los dispositivos de iOS, macOS, watchOS y tvOS. En cambio Flutter se encuentra elaborado por la compañía de Google, pero está enfocado para iOS y Android. En este documento se investigará las funciones, características, ventajas, desventajas, los inicios de estas tecnologías en donde permita conocer cuál cuenta con más facilidad y menos lentitud para el desarrollo de aplicaciones móviles, conociendo los inicios y sus versiones, también se indicará mediante tablas comparativas los principales enfoques que tienen ambos sistemas, en donde al realizar una conclusión se haga más fácil porque así se observará la factibilidad que tiene cada una de ellas. Por lo general Swift es uno de los lenguajes de programación más popular pero solo se encuentra para la plataforma de Apple creando en sí aplicaciones de alta calidad, mientras que el framework de Flutter se encuentra codificado por el lenguaje de programación Dart, permitiendo diseñarse para sus plataformas de los sistemas operativos móviles más utilizados en el mercado comercial de los usuarios.

Palabras claves

Tecnologías, Desarrollo, Plataformas, Aplicaciones, Móviles.

INTRODUCCION

En este estudio comparativo se realizará una investigación completa entre ambos sistemas, logrando obtener información que permita conocer cuál es el desarrollador móvil más eficaz para los dispositivos.

En este caso comparativo cuenta con un objetivo general analizar los sistemas móviles tanto Swift y Flutter para así conocer cuál es el lenguaje adecuado para desarrollar aplicaciones, en donde esto permitirá obtener una mejor organización y control al realizar un desarrollo de aplicaciones móviles, se conocerá un poco de cada sistema llegando a tener la finalidad de encontrar mucha información que sea de uso para la elaboración de una buena aplicación móvil, es por eso que mediante el desarrollo de esta investigación se detallará las características, ventajas, desventajas y muchas otras funciones que sean importante para así conocer el lenguaje que sería adaptado en un dispositivo móvil, en la cual tiene una problemática en este documento de desarrollo existen en todo el mundo dispositivos móviles tanto como Android y iOS, en conocer cómo se encuentran desarrollado las aplicaciones móviles, para que al programador o diseñador principiante se le haga fácil al querer diseñar una para muchas empresas o de uso personal, siempre y cuando obteniendo las documentaciones y guías que permita tener información acreditada y verificada para que las aplicaciones sean conocidas mediante alguna plataforma web.

Estos dos sistemas tanto como Swift que es una de las tecnologías que se encuentra desarrollada por Apple y solo se diseña dentro de iOS, macOS, watchOS y tvOS, mientras que por lo tanto el framework de Flutter su desarrollo móvil se encuentra en código abierto y es únicamente diseñado para iOS y Android, ambas tecnologías o desarrolladores de aplicaciones cuentan con sus propias funciones permitiendo al programador diseñarlas mediante las características y funciones necesarias que requiera cada aplicación o dispositivo que se esté utilizando.

En este trabajo comparativo se inclinó a su línea investigativa desarrollo de sistemas de la información, comunicación y emprendimiento e innovación y una de sus sublínea investigativa por las redes y tecnologías inteligentes de software y hardware, esto ayudará a este trabajo comparativo a enfocarse en el desarrollo de ambas tecnologías que será beneficiado a nivel nacional e internacional, mediante la selección que se realizó para conocer la línea y sublínea, es de mucha importante porque así se conocerá el enfoque que tendrá este proyecto mediante su desarrollo.

La metodología a desarrollarse dentro de este estudio comparativo será la descriptiva, en la cual permitirá agrupar y organizar las ideas que se lograran receptor de las diferentes plataformas web, libros, documentos y articulo en donde permitirá obtener información actualizada, logrando conocer cuál sería la tecnología que se podría adaptar para el diseño de muchas aplicaciones, en este documento no se realizará encuestas por el hecho que es un tema completamente investigativo y no tiene un enfoque hacia un lugar en particular, por lo tanto se utilizará la aplicación de google trends que es de fácil uso y rápida en donde permitirá conocer mediante los últimos cinco años como se han encontrado desarrollándose la tecnología de Swift y Flutter.

DESARROLLO

En la actualidad los sistemas móviles de los más usados solo existen dos, son tanto los usuarios de Android y los usuarios iOS. Durante los últimos años, el internet ha tenido una gran lucha, el duopolio existió desde hace mucho tiempo atrás, pero no duró dentro del marketing tecnológico. Han existido muchos sistemas diferentes, unos más poderosos que otros, por lo cual dentro de la industria móvil salieron mejores sistemas lo que provoco que los otros ya no sean elegidos por los diferentes usuarios.

Se enlistará las características de los sistemas móviles que los hacen importantes, con cualquier sistema instalado en su dispositivo.

➤ **Portabilidad**

Los sistemas móviles son diseñados para desarrollarse dentro de los dispositivos móviles, lo que esto significa que son portátiles y ser usados de un lugar a otro.

➤ **Conectividad**

Estos sistemas se encuentran con su implementación redes móviles como las de Wi-Fi, 3G, 4G Y 5G, en donde esto permitirá tener comunicación con otros dispositivos.

➤ **Pantalla táctil**

La mayoría de estos sistemas vienen instalados el interfaz de usuario basado en pantalla táctil.

➤ **Batería**

Son desarrollados para consumir poca energía y obteniendo en sí una batería de larga duración, por lo que ellos están destinados a ser utilizados durante largos tiempos.

➤ **Sensores**

Tiene una variedad de sensores instalados, por ejemplo como el acelerómetro, giroscopio, sensor de proximidad, en la que permite a los dispositivos detectar su entorno y responder en consecuencia.

➤ **Personalización**

Es diseñada la opción para que los usuarios puedan personalizar su dispositivo móvil, con el tema o foto que deseen.

➤ **Seguridad**

Tienen integrado dentro del dispositivo medidas de seguridad para proteger las cuentas ingresadas o alguna información que los usuarios ingresen a su teléfono móvil.



Ilustración 1. Historia de los sistemas móviles

Fuente: (Estradas, 2020)

Según el autor del sitio web (Borja, 2018). Se indicarán mediante un cuadro algunos de los sistemas móviles que existen y existieron en el mercado de los dispositivos tecnológicos.

Historia de los sistemas móviles	PalmOS 1996 - 2007
	Symbian 1997 - 2012
	Windows Mobile 2000 – 2010
	Maemo 2005 – 2011
	Android 2003 – Actualidad
	iOS 2007 – Actualidad
	MeeGo 2010 – 2012
	Bada OS 2010 – 2013
	Windows Phone 2010 - 2015
	Ubuntu Touch 2011 – 2017
	Firefox 2012 – 2015
	Tizen 2012 – Actualidad
	BlackBerry OS 10 2013 – Actualidad

Tabla 1. Historia de los sistemas móviles

Creado por: Roberto Ramos

Según el autor del sitio web (Interactivo, 2018). La primera aplicación móvil se remonta a los años 90, es conocido por el calendario, juegos arcada, editor de sonido, etc. Realizaban trabajos básicos pero su plan era tan simple, la evolución de las aplicaciones es rápida debido a las nuevas tecnologías que iban surgiendo en aquel entonces como por ejemplo el de WAP y Transferencia de Datos (EDGE). Por lo tanto Apple lanzó el iPhone y luego de eso acabó con otros teléfonos inteligentes, incluyendo a Android el principal competidor de iPhone.

A continuación se enlistará algunas de las tiendas móviles que tienen los sistemas móviles de los diferentes dispositivos.

- Google Play
- App Store
- Windows Phone Store
- BlackBerry World

En la actualidad existe la problemática que se desconoce cómo se encuentran distribuidos o desarrollados ciertos sistemas móviles, por lo tanto muchos dispositivos móviles tanto como Android y iOS, es importante conocer cómo se encuentran diseñados los sistemas a tratarse dentro de este estudio comparativo, para así conocer cuál es factible y fácil de usar para que el usuario al querer desarrollar una aplicación móvil poder elegir e inclinarse en el sistema móvil más eficaz.



Ilustración 2. Desarrollo de aplicaciones móviles

Fuente: (Benedí, 2019)

SWIFT “iOS”

Según el autor del sitio web (Saghatelyan, 2018). Este sistema móvil es un lenguaje de programación que fue creado por Apple, con el desarrollo de crear aplicaciones iOS y macOS, este sistema es de código abierto, ingresando desde el año 2014, tiene compatibilidad con el lenguaje de programación “Objective-C” y se encuentra enfocado con la instrucción orientada a objetos, una de sus ventajas principales es que requiere de una tasa de aprendizaje baja, esto quiere decir que puede ser uno de los principales lenguaje de desarrollo dispositivos móviles.



Ilustración 3. Lenguaje de programación móvil

Fuente: (Swift, 2017)

Este es en lenguaje de programación versátil y de muy alto nivel que admite lenguajes de programación, funcionales y que se encuentra orientado a objetos, la sintaxis de esta tecnología es moderna y fácil de leer y escribir, lo que para el programador lo hace accesible para los que se encuentran en su etapa de principiantes en el mundo de la programación. Swift tiene una gran biblioteca en la cual brinda a los desarrolladores muchas herramientas y características que facilitan a la creación de aplicaciones, también cuenta con el uso de un sistema de recolección de basura, lo que esto lo hace más importante que los desarrolladores no tienen que preocuparse por la gestión de un manual de la memoria.

A continuación se mostrará en un cuadro las versiones en la que este lenguaje de programación Swift ha hecho que sus codificaciones sean más fácil para sus programadores.

VERSIONES	Swift 1.0 “Versión inicial en junio del 2014”
	Swift 2.0 “Lanzamiento en septiembre de 2015”
	Swift 3.0 “Lanzamiento en septiembre de 2016”
	Swift 4.0 “Su versión fue lanzada en septiembre 2017”
	Swift 5.0 “ Lanzamiento en marzo de 2019”
	Swift 5.1 “Lanzamiento en septiembre de 2019”
	Swift 5.2 “Su versión fue lanzada en marzo de 2020”
	Swift 5.3 “Lanzamiento en septiembre 2020”
	Swift 5.4 “Lanzamiento en marzo de 2021”

Tabla 2. Versiones del lenguaje de programación Swift

Creado por: Roberto Ramos

Es importante reconocer que las versiones principales de lenguaje de programación para dispositivos móvil Swift, Apple también ha lanzado alguna actualizaciones que incluyen un sin números de correcciones de errores y de mejoras en su rendimiento.

Según el autor del sitio web (KeepCoding, 2018). Se enlistará las características que hacen importante este lenguaje de programación Swift.

- Comandos
- Funciones
- Ciclos
- Parámetros
- Código Condicional
- Variables
- Operadores
- Tipos
- Inicialización
- Corrección de errores

En el siguiente cuadro se desplegará las ventajas y desventajas del lenguaje de programación Swift, en la cual permite conocer como está desarrollado este sistema.

Ventajas	Fácil su aprendizaje y uso.
	Rapidez y eficiente.
	Interoperabilidad.
	Seguridad.
	Open Source.

Tabla 3. Ventajas del lenguaje de programación Swift

Creado por: Roberto Ramos

Desventajas	Existe falta de soporte de versiones antiguas iOS
	Poca estabilidad
	Curva de aprendizaje
	Actualizaciones frecuentes

Tabla 4. Desventajas del lenguaje de programación Swift

Creado por: Roberto Ramos

Swift es uno de los lenguajes de programación móviles más popular que tiene muchas ventajas, pero también con algunas desventajas, sin embargo cuenta con un enfoque en la eficiencia y la seguridad al servicio de la comunidad de desarrollo. Según el autor del sitio web (SwiftBeta, 2020). A continuación se mostrará las funciones que obtiene el lenguaje de programación Swift, permitiendo conocer su codificación, para que el programador tenga más facilidad a la hora de desarrollar las aplicaciones.

➤ **Permite el retorno o no de valores**

```
func createUser(name: String, age: Int) -> Bool {
    print("User name: \(name)")
    print("User age: \(age)")
    return true
}
```

Ilustración 4. Retorno o no de valores del lenguaje de programación móvil Swift.

➤ **Función de String, Int – Bool**

```
let createUserFunction = createUser(name:Age:)
```

Ilustración 5. Función String del lenguaje de programación móvil Swift.

➤ Tuplas en Swift

```
func getTwoValues() -> (String, String) {
    let name = "Swift"
    let surname = "Beta"
    return (name, surname)
}

let (name, surname) = getTwoValues()
```

Ilustración 6. Las tuplas del lenguaje de programación móvil Swift.

➤ Nombre de los parámetros

```
func getTwoValues(paramOne name: String,
                  paramTwo surname: String) -> (String, String) {
    return (name, surname)
}

let (name, surname) = getTwoValues(paramOne: "Swift",
                                   paramTwo: "Beta")
```

Ilustración 7. Nombre de los parámetros del lenguaje de programación móvil Swift.

➤ getTwoValues

```
func getTwoValues(_ name: String,
                 _ surname: String) -> (String, String) {
    return (name, surname)
}

let (paramOne, paramTwo) = getTwoValues("Swift",
                                       "Beta")
```

Ilustración 8. getTwoValues del lenguaje de programación móvil Swift.

➤ Valores por defecto

```
func createDatabase(name: String, path: String = "/") {
    // Do something
}

createDatabase(name: "SwiftBetaDatabase")
```

Ilustración 9. Valores por defecto del lenguaje de programación móvil Swift.

➤ **Parámetros variadic en funciones**

```
func validate(names: String...) {  
    print("Names \(names)")  
}  
  
validate(names: "Swift", "Beta")
```

Ilustración 10. Parámetros variadic del lenguaje de programación móvil Swift.

➤ **Parámetros In-Out**

```
func validateNames(names: [String]) {  
    names = ["Beta", "Swift"]  
    print("Names \(names)")  
}  
  
validateNames(names: ["Swift", "Beta"])
```

Ilustración 11. Parámetros In-Out del lenguaje de programación móvil Swift.

El lenguaje de programación Swift es uno de los más populares para el desarrollo de aplicaciones móvil en las plataformas Apple iOS y iPadOS, es un lenguaje principal para los programadores de iOS juntándose así con la plataforma Cocoa Touch en las que permite crear aplicaciones nativas. Su sintaxis es corta y legible a la vez, obtiene una seguridad, su facilidad y el buen rendimiento lo hacen adecuado para un mejor desarrollo móvil. Esta tecnología es de código abierto, lo que esto lo hace aún más importante que se puede usar para crear aplicaciones en una variedad de plataformas y aplicaciones diferentes.

FLUTTER “ANDROID Y iOS”

El framework de Flutter es uno de los sistemas de códigos abiertos desarrollado por Google en el año 2017, su diseño fue para darle creación a aplicaciones móviles para Apple e iOS, en la actualidad esta tecnología se encuentra codificada con el lenguaje de programación Dart, normalmente este también se encuentra desarrollado por Google, su enfoque es de construir aplicaciones de alta calidad con una interfaz de usuario atractiva y alto rendimiento. Este grupo de programadores buscaban obtener algunas limitaciones que les permita el desarrollo de aplicaciones móviles tradicionales, por ejemplo como la inconsistencia en el diseño de aplicación, la visibilidad entre plataformas y dispositivos y se encuentra en la necesidad de escribir un código separado por cada plataforma.



Ilustración 12. Framework de Flutter

Fuente: (development, 2021)

El framework de Flutter es reconocido por ser uno de los desarrolladores de aplicaciones móviles, este se encuentra basado en generar muchas herramientas y bibliotecas para el desarrollo móvil para iOS y Android, lo que esto significa que las aplicaciones que están creadas desde Flutter son compiladas desde un código local de cada plataforma y se ejecutan correctamente.

Este framework se encuentra en constantes actualizaciones por lo que sus aplicaciones lo requieren es por eso que mediante un cuadro se indicarán las versiones que obtiene Flutter en donde los desarrollos a sus aplicaciones los hace más eficaz y factible a la vez.

Características	1.0 “En el 2018 se desarrolló la primera versión estable de Flutter”.
	1.5 “Su lanzamiento fue en mayo de 2019, obtiene mejoras en su rendimiento y estabilidad”.
	1.9 “Lanzada en septiembre de 2019, mejoró su calidad y estabilidad”.
	2.0 “Su lanzamiento fue en marzo de 2021, es una de las versiones más importantes”.
	2.5 “Lanzada en septiembre 2021, mejoró en el rendimiento de la aplicación”.

Tabla 5. Características de Framework Flutter

Creado por: Roberto Ramos

Es importante reconocer que este framework que es desarrollado para Android y iOS es una de las plataformas que comúnmente se encuentra en una evolución constante, y se logra esperar que Flutter pueda obtener más versiones para así mejorar la calidad, la estabilidad y las características que tiene la plataforma mencionada.

Según el autor del sitio web (Amazon, 2018). Se indicará en cuadros las ventajas y desventajas que tiene el framework de Flutter, en donde está destacado por ser un desarrollo en multiplataforma.

Ventajas	Obtiene un rendimiento casi nativo, por lo que su código es compilado desde una máquina.
	Tiene un rendimiento eficaz y a la vez se puede personalizar a gusto de los usuarios.
	Cuenta con muchas herramientas para sus desarrolladores permitiendo darles la facilidad a sus usuarios al momento de programar.

Tabla 6. Ventajas de Flutter

Creado por: Roberto Ramos

Desventajas	Curva de aprendizaje muy bajo, ya que para un programador con experiencia se le hará más fácil su desarrollo.
	Tamaño de la aplicación.
	Tienen muchos problemas de compatibilidad.
	Limitaciones de diseño en sus programas.
	Dependencia de un solo proveedor.

Tabla 7. Desventajas de Flutter

Creado por: Roberto Ramos

Según el autor del sitio web (Devs, 2019). Se desplegará las características principales que tiene el framework de Flutter.

- Las aplicaciones nativas en Flutter se desarrollan directamente para el sistema operativo, por lo general este framework aprovecha al máximo las aplicaciones nativas y así poder lograr resultados de calidad.
- El framework de Flutter incorpora el Material Design de Google y Cupertino desde Apple, donde bajo la experiencia esto permitirá al usuario obtener las herramientas desarrolladas por las propias empresas que han solicitado dicho diseño.
- Las razones principales para deshacerse de una aplicación es el tiempo que se requiere en ello instalando y viceversa, es por eso que el tiempo de carga del framework de Flutter es nada menos que de un segundo en cualquier dispositivo de iOS o Android, cuenta con esa gran compatibilidad.
- Obtiene una flexibilidad y un rápido crecimiento a la vez, ya que cuenta con la herramienta de recarga caliente, en donde permite su planificación y ver cambios en tiempo real desde el dispositivo o los diferentes simuladores que este framework trabaje.

Flutter es uno de los marcos de códigos abiertos que como ya se lo ha indicado es desarrollado por Google, para realizar la creación de aplicaciones nativas sea más fácil, rápida y sencilla, es por eso que framework su función principal es la de su codificación que es 100% nativo para cada plataforma que desempeñe su diseño, haciendo que UX sea muy similar a las aplicaciones tradicionales.

Según el autor del sitio web (Reséndiz, 2020). Se conocerá las funciones importante que tiene Dart que es uno de los lenguajes de programación en el que esta transcrito el framework de Flutter.

➤ Sintaxis

```
Dart
1 void nombreFuncion() {
2     //bloque de codigo
3 }
```

Ilustración 13. Sintaxis de Dart

➤ Tipo de dato

```
Dart
1 tipoDato nombreFuncion() {
2     //bloque de codigo
3     return valor;
4 }
```

Ilustración 14. Tipo de dato

➤ Funciones con parámetros

```
Dart
1 int suma(numeroUno, numeroDos) {
2     int resultado;
3     resultado = numeroUno + NumeroDos;
4     return resultado;
5 }
```

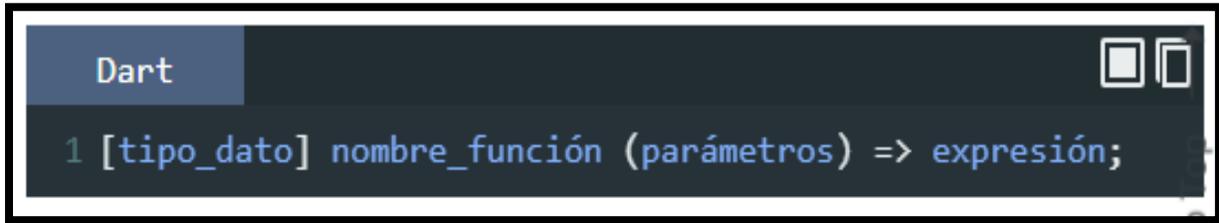
Ilustración 15. Funciones con parámetros

➤ Funciones recursivas

```
Dart
1 double factorial(numero) {
2     if (numero <= 0) {
3         // termina el caso
4         return 1;
5     } else {
6         return (numero * factorial(numero - 1));
7     } // function invokes itself
8 }
9 }
```

Ilustración 16. Funciones recursivas

➤ **Funciones de flecha o arrow**



```
Dart
1 [tipo_dato] nombre_función (parámetros) => expresión;
```

Ilustración 17. Funciones de fleca o arrow

A continuación se mostrará en un cuadro las funciones que tiene Flutter que lo hace aun cada vez más importante para desarrollar las aplicaciones móviles.

Funciones de Flutter	Hot Reload “Tiene permitido a los desarrolladores hacer cambios de códigos”.
	Widgets “Crea interfaces de usuario”.
	Soporte multiplataforma “Desarrollo de aplicaciones en iOS, Android y web”.
	Animaciones y gráficos “Crea animaciones y gráficos interactivos”.
	Personalización “Ofrece una gran flexibilidad en diseños y funcionalidad”.
	Herramientas de depuración “Gama amplia de depuración”.
	Integración con otras tecnologías “Se integra fácilmente”.

Tabla 8. Funciones del framework de Flutter

Creado por: Roberto Ramos

Tablas comparativas entre Swift y Flutter

LENGUAJE DE PROGRAMACIÓN	
SWIFT	FLUTTER
Es un lenguaje de programación diseñado para desarrollar aplicaciones móviles dentro Apple.	No es un lenguaje de programación sino un framework, que utiliza el lenguaje de programación Dart para desarrollar aplicaciones móviles.

Tabla 9. Tabla comparativa de lenguaje de programación entre Swift y Flutter

Creado por: Roberto Ramos

RENDIMIENTO	
SWIFT	FLUTTER
Es conocido por su velocidad y eficiencia, por lo tanto compila directamente las aplicaciones nativas.	Utiliza un motor renderizado personalizado para que las aplicaciones tengan un mejor rendimiento.

Tabla 10. Tabla comparativa de rendimiento entre Swift y Flutter

Creado por: Roberto Ramos

CURVA DE APRENDIZAJE	
SWIFT	FLUTTER
Requiere que los programadores tengan conocimiento profundo de programación.	Su aprendizaje es mucho más fácil para los programadores que tienen poca experiencia.

Tabla 11. Tabla comparativa de curva de aprendizaje entre Swift y Flutter

Creado por: Roberto Ramos

CONCLUSIONES

La tecnologías de Swift y Flutter son ambos sistemas diferentes funciones, ventajas y desventajas, por lo que hace en términos de desarrollo móvil, mediante la investigación se conoce que Swift es uno de los lenguajes de programación utilizado solo para la plataforma de Apple, y es reconocido por la velocidad y eficiencia que se obtiene al desarrollar las aplicaciones básica y nativas, mientras tanto Flutter no es un lenguaje de programación si no un framework que permite el desarrollo móvil con la codificación del lenguaje de programación Dart y normalmente las aplicaciones que han sido desarrolladas mediante este framework son de alta calidad y con una experiencia de usuario fluida.

En base a la investigación se llega a una conclusión que ya dependerá del uso de tecnología que se desea utilizar, revisando los factores y tipos de dispositivos en los que se desea desarrollar las aplicaciones móviles, en la cual también se señala que si el usuario es un programador principiante y quiere aplicar diferentes plataformas y con muchos tipos, diseños, personalizaciones el framework de Flutter es recomendado para ser una buena opción, mientras que si es un programador de gama alta se recomendará su inclinación por diseñar aplicaciones para dispositivos del grupo de Apple.

BIBLIOGRAFÍAS

- Amazon. (10 de marzo de 2018). *Amazon.com*. Obtenido de Amazon.com:
<https://aws.amazon.com/es/what-is/flutter/#:~:text=Flutter%20es%20un%20marco%20de,con%20un%20%C3%BAnico%20c%C3%B3digo%20base>.
- Benedí, Á. (14 de octubre de 2019). *LAS MEJORES PLATAFORMAS HÍBRIDAS PARA DESARROLLO DE APPs MÓVILES*. Obtenido de LAS MEJORES PLATAFORMAS HÍBRIDAS PARA DESARROLLO DE APPs MÓVILES:
<https://indexdesarrollo.com/las-mejores-plataformas-hibridas-para-desarrollo-de-apps-moviles/>
- Borja. (12 de agosto de 2018). *HISTORIA DE LOS SISTEMAS MOVILES* . Obtenido de HISTORIA DE LOS SISTEMAS MOVILES :
https://www.elespanol.com/elandroidelibre/20180812/historia-sistemas-moviles-toda-competencia-android/329717692_0.html
- development, F. a. (16 de abril de 2021). *why you should choose Flutter*. Obtenido de why you should choose Flutter: <https://www.appify.digital/post/flutter-app-development>
- Devs, Q. (5 de julio de 2019). *Qué es Flutter y por qué utilizarlo en la creación de tus apps*. Obtenido de Qué es Flutter y por qué utilizarlo en la creación de tus apps.:
<https://www.qualitydevs.com/2019/07/05/que-es-flutter/>

Estradas, V. M. (1 de octubre de 2020). *Evolución de los sistemas móviles*. Obtenido de Evolución de los sistemas móviles: <https://icolectiva.blog/2020/10/01/evolucion-de-los-sistemas-moviles/>

Interactivo, A. (22 de julio de 2018). *HISTORIA DE LAS APLICACIONES MOVILES* . Obtenido de HISTORIA DE LAS APLICACIONES MOVILES : <https://arteint.com/2018/07/22/historia-las-aplicaciones-moviles/>

KeepCoding. (30 de mayo de 2018). *¿Qué es Swift y cuáles son sus principales características?* Obtenido de ¿Qué es Swift y cuáles son sus principales características?: <https://keepcoding.io/blog/que-es-swift-y-sus-principales-caracteristicas/>

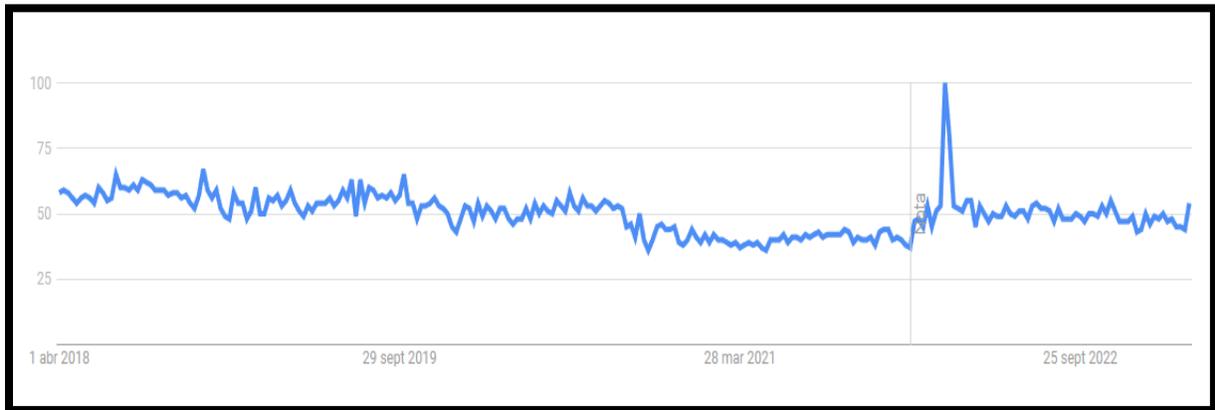
Reséndiz, D. (2 de mayo de 2020). *Funciones en dart / Flutter funciones - Blog de Informática*. Obtenido de Funciones en dart / Flutter funciones - Blog de Informática: <https://deiniresendiz.com/dart-funciones/>

Saghatelian, H. (2018). *DESARROLLO DE UNA APLICACION BASADA EN SISTEMA iOS*. Obtenido de DESARROLLO DE UNA APLICACION BASADA EN SISTEMA iOS: <https://riunet.upv.es/bitstream/handle/10251/124877/Saghatelian%20-%20Desarrollo%20de%20una%20aplicaci%3b3n%20basada%20en%20sistemas%20iOS%20con%20Xcode%20y%20Swift%20para%20deportis....pdf?sequence=1&isAllowed=y>

Swift. (29 de agosto de 2017). *Lenguajes de programación*. Obtenido de Lenguajes de programación: <https://lenguajesdeprogramacion.net/swift/>

SwiftBeta. (25 de agosto de 2020). *FUNCIONES EN SWIFT EN ESPAÑOL* . Obtenido de
FUNCIONES EN SWIFT EN ESPAÑOL : <https://www.swiftbeta.com/funciones-en-swift/>

ANEXOS



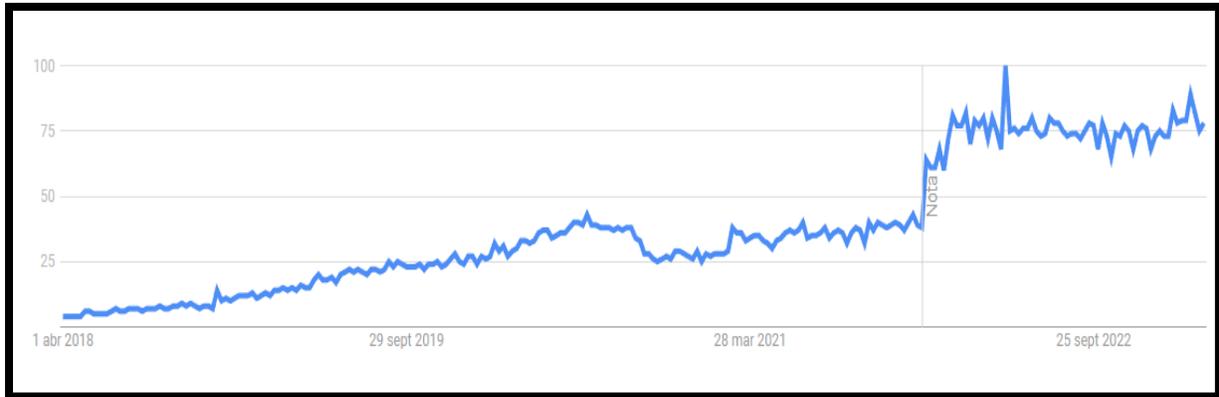
Fuente: Google Trends

Análisis: Se puede observar que el lenguaje de programación Swift desde los últimos años a nivel mundial en el 2021 obtuvo una subida de uso de creación de aplicación utilizando dicho lenguaje en mención.



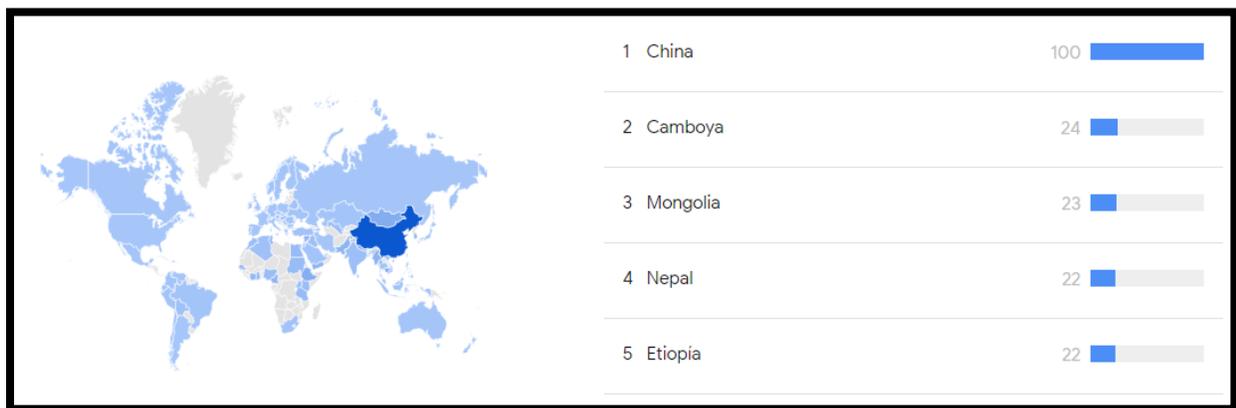
Fuente: Google Trends

Análisis: En la gráfica se observa que Swift, señala los cinco primeros lugares entre países y ciudades indicando en China como uno de los sitios donde más se utilizó dicho sistema.



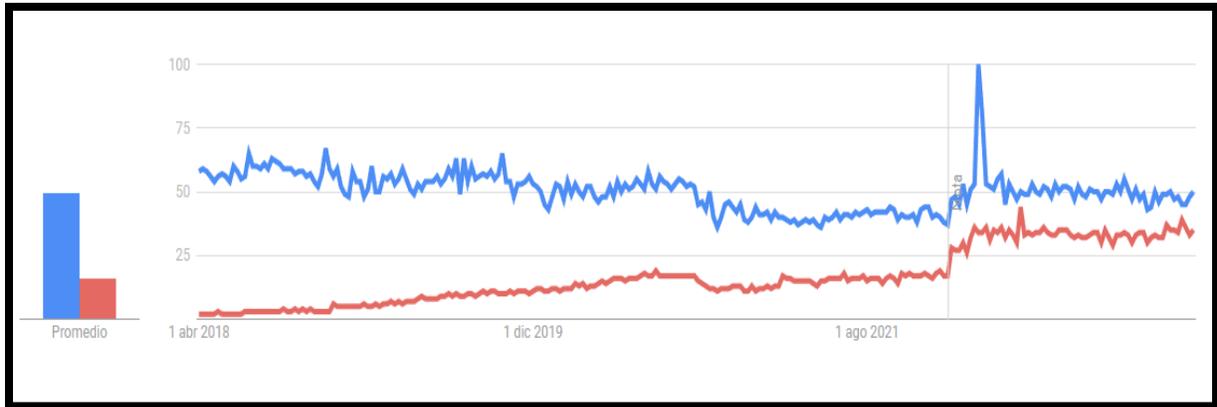
Fuente: Google Trends

Análisis: Se puede observar en la imagen que el framework de Flutter en los últimos cinco años, ha tenido un crecimiento notorio, en donde se indica que en el 2022 se ha mantenido hasta el año actual.



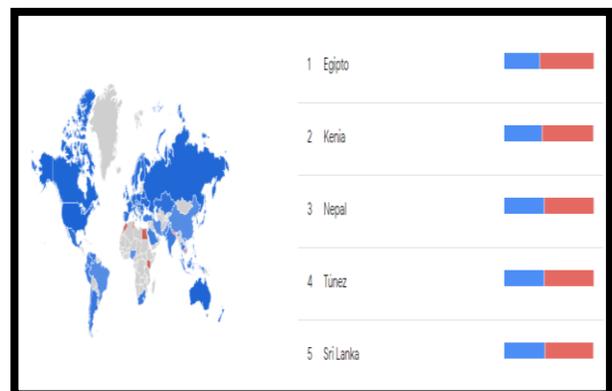
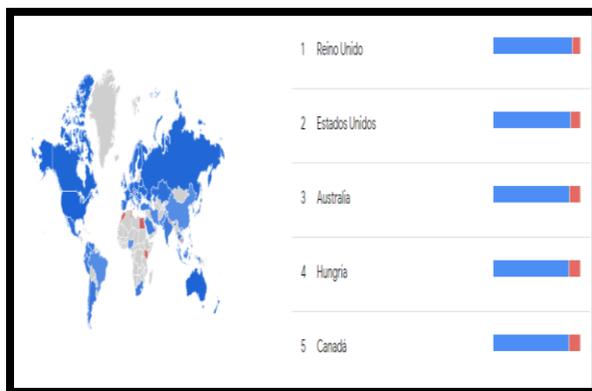
Fuente: Google Trends

Análisis: Se puede observar casi igual que a la anterior imagen de muestra en países y ciudades como China se mantiene en el uso de tecnologías, siendo así el framework de Flutter para algunas ciudades dentro de esa localidad la favorita para ellos.



Fuente: Google Trends

Análisis: Se puede observar en esta imagen comparativa como el lenguaje de programación Swift es uno de los más utilizados durante los últimos cinco años a nivel mundial.



Fuente: Google Trends

Análisis: Se observa en la primera imagen a nivel mundial realizando la comparación entre ambas tecnologías como Swift ocupa todo el recuadro siendo uno de los sistemas más utilizados para diseñar aplicaciones móviles, mientras que en el otro cuadro se observa el framework de Flutter que mantiene dentro de su comunidad un parcial, esto quiere decir que su diseño y el de Swift en esos países y ciudades son compartidos.