

**UNIVERSIDAD DVENTISTA DE CENTRO AMÉRICA**  
**ESCUELA DE INGENIERÍA EN SISTEMAS**

Programación III

Investigación Flutter

Estudiantes

Jullyus Davis Williams

Ángel Fernando Maldonado Laínez

Profesor

Brian Funez

**2703/2019**

# Índice

Introducción .....	3
Objetivos .....	4
Objetivo general .....	4
Objetivo específico .....	4
Marco teórico .....	5
Antecedentes / historia .....	5
Instalación .....	6
Entorno y compatibilidad .....	14
Lenguaje de programación (DART) .....	15
Diseño de la app .....	17
Conclusiones .....	20
Bibliografía .....	21
ANEXOS .....	22

## Introducción

En este documento escrito nos vamos a dar cuenta que es Flutter y todas sus especificaciones. Al ser un framework muy reciente la mayoría de las personas no tienen una idea clara ni específica de lo que es este software. Al ser como antes se mencionaba un software nuevo para la sociedad y la informática en general suele dudar de su funcionamiento; en este documento vamos a dar a conocer el porque es un gran software que esta para revolucionar los años venideros y los idiomas nativos de programación. No se esta creando ninguna discordia contra ningún idioma de programación sino solo ver el ámbito de este framework y su utilidad y la revolución que esta tenga mas adelante.

## Objetivos

### Objetivo general

- Dar a conocer un nuevo framework llamado Flutter, como una nueva opción potente para el desarrollo de software.

### Objetivo específico

- Mostrar el entorno de desarrollo de Flutter
- Mostrar a Dart como lenguaje de programación especificado usado para este nuevo software.
- Ejemplificar una pequeña aplicación utilizando Flutter y Dart.
- Exponer ventajas y desventajas de Flutter como software de desarrollo multiplataforma.

## Marco teórico

### Antecedentes / historia

El desarrollo de aplicaciones móviles es un campo de esfuerzo relativamente reciente. Los desarrolladores de terceros han podido crear aplicaciones móviles durante menos de una década, por lo que no es sorprendente que las herramientas sigan evolucionando.

El Apple iOS SDK se lanzó en 2008 y el Google Android SDK en 2009. Estos dos SDK se basaron en diferentes idiomas: Objective-C y Java, respectivamente.

Su aplicación habla con la plataforma para crear widgets o acceder a servicios como la cámara. Los widgets se representan en un lienzo de pantalla y los eventos se devuelven a los widgets. Esta es una arquitectura simple, pero es casi necesario crear aplicaciones separadas para cada plataforma porque los widgets son diferentes, por no mencionar los idiomas nativos.

Los primeros frameworks multiplataforma se basaron en JavaScript y WebViews. Los ejemplos incluyen una familia de marcos relacionados: PhoneGap, Apache Cordova, Ionic y otros. Antes de que Apple lanzara su SDK de iOS, alentaban a los desarrolladores externos a crear aplicaciones web para el iPhone, por lo que la creación de aplicaciones multiplataforma utilizando tecnologías web era un paso obvio. Su aplicación crea HTML y lo muestra en una vista web en la plataforma. Tenga en cuenta que es difícil para los lenguajes como JavaScript hablar directamente con el código nativo (como los servicios), por lo que pasan por un "puente" que hace cambios de contexto entre el reino de JavaScript y el reino nativo. Debido a que los servicios de plataforma generalmente no se llaman con tanta frecuencia, esto no causó demasiados problemas de rendimiento.

Ya habiendo tomado en cuenta la historia sobre el transcurso del tiempo y los primeros frameworks multiplataforma entramos a la historia en si sobre Flutter, este entorno sale con el nombre de "Sky" en el 2015, los desarrolladores de Google que son los creadores de este medio trabajaron en el hasta llamarlo Flutter en mayo del 2017, ya en este momento el susodicho framework ya estaba a poco de salir a la luz. El 23 de marzo del

2018 se publica el anuncio del software ya mejorado y de una manera gratuita para todo el público. Esta fue la versión beta que estaría actualizándose cada vez que lo fuera necesario, pero esta fue la base y la versión original que salió a la luz. Flutter encontró el punto óptimo del desarrollo de experiencias móviles: integraciones de rendimiento y plataformas para elementos móviles nativos, con desarrollo de alta velocidad y alcance multiplataforma de kits de herramientas de IU portátiles.

Flutter está diseñado para programadores de dispositivos móviles nuevos y experimentados, y puede ayudar a crear apps finas y exitosas en tiempo récord con beneficios como los siguientes:

- **Desarrollo de alta velocidad** con funciones como recarga activa con estado, un nuevo marco de trabajo reactivo, un conjunto completo de widgets y herramientas integradas.
- **Diseños expresivos y flexibles** con conjuntos compatibles de widget, bibliotecas de animaciones completas, y una arquitectura segmentada y extensible.
- **Experiencias de alta calidad** en diferentes dispositivos y plataformas gracias a nuestro representador portátil con aceleración por GPU, tiempo de ejecución de código ARM nativo de alto rendimiento e interoperabilidad entre plataformas.

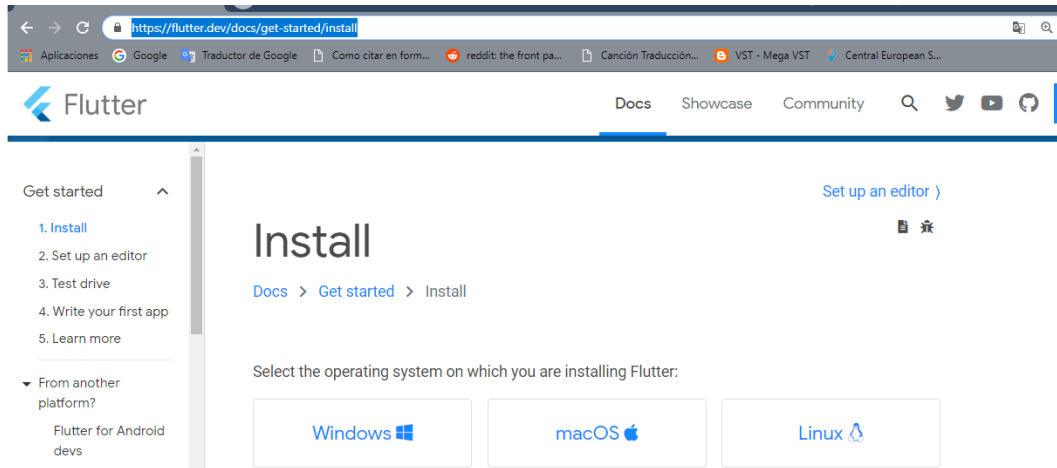
## Instalación

La instalación de este programa es compleja y se pueden producir varias complicaciones; por lo cual se ha preparado una guía rápida de la instalación del elemento.

El primer paso es dirigirse a la página web de flutter, específicamente en el siguiente link

<https://flutter.dev/docs/get-started/install>.

Luego seleccionamos el sistema operativo que poseamos en nuestra computadora (como se muestra en la imagen). Para esta pequeña guía utilizaremos el sistema operativo Windows.



*Ilustración 1. Tomada de <https://flutter.dev/docs/get-started/install>.*

Una vez seleccionado el sistema operativo nos aparecerá información y requerimientos del sistema para la instalación de flutter y algunas de sus dependencias como lo son Windows Powershell y Git. Pero lo que nos interesa es la descarga del SDK de flutter y para ello bajamos un poco en la página hasta que nos aparezca el botón de descarga. Le damos clic y nos aparecerá la descarga casi inmediatamente.

## Investigación Flutter Programación III

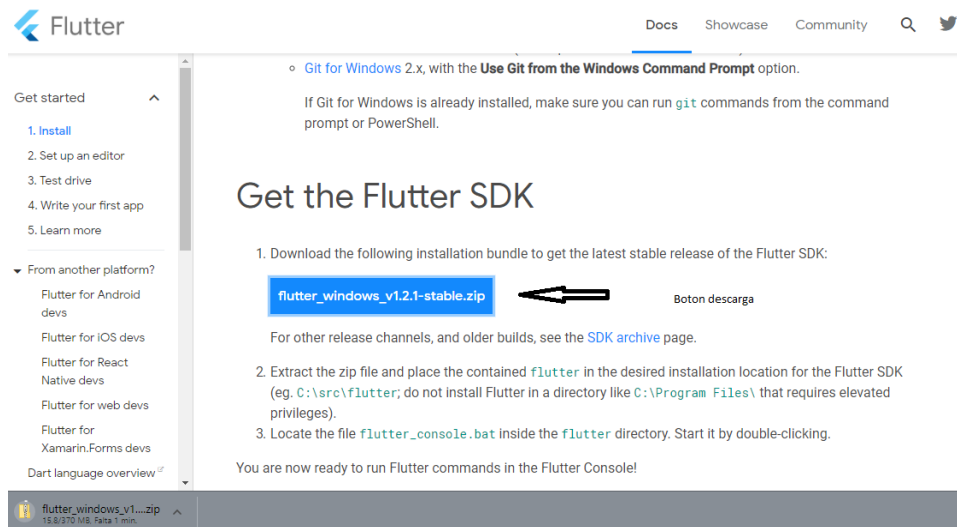


Ilustración 2. Tomada de <https://flutter.dev/docs/get-started/install>.

Una vez descargado el archivo, lo abrimos y lo descomprimos en la ruta deseada. En este caso yo lo descomprimí en Descargas.

Una vez extraído el archivo, nos introducimos en él y localizamos el archivo flutter\_console.bat y lo ejecutamos para comprobar de que los archivos descargados funcionan bien. Luego lo cerramos

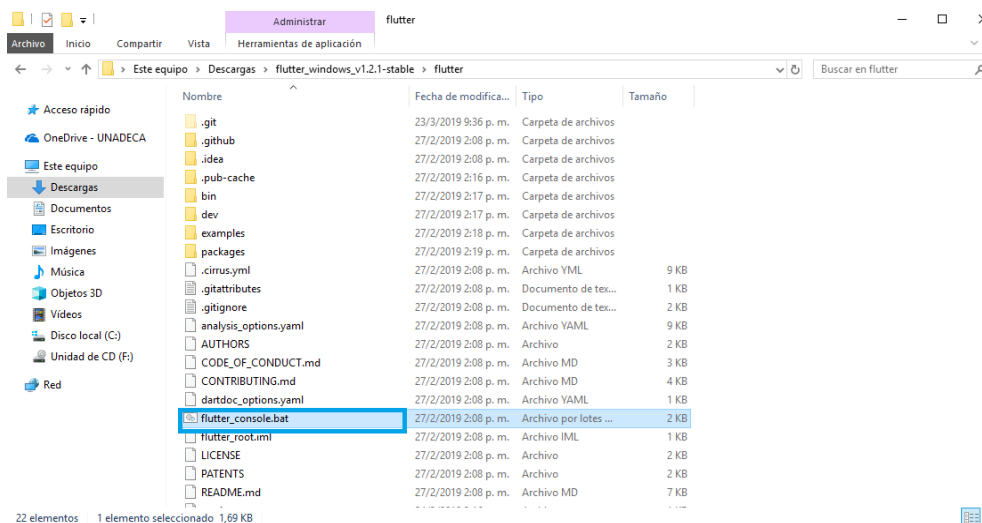
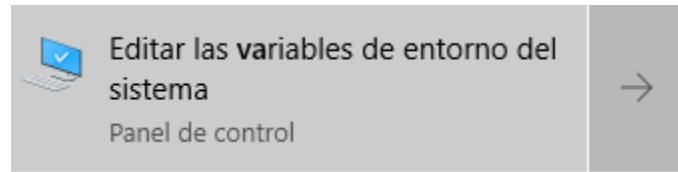


Ilustración 3 Instalación Flutter

Posteriormente, tendremos que agregar las variables al sistema.

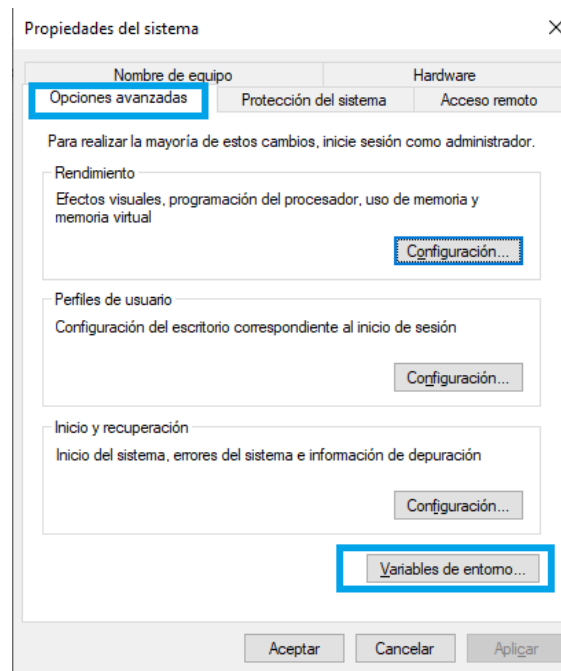
Escribimos en el buscador de Windows “Variables del sistema” y nos aparecerá una opción llamada “Editar Variables de Entorno del Sistema” y la damos click.





*Ilustración 4 Variables del Sistema Windows*

Una vez ejecutado, nos aparecerán las propiedades del sistema; en donde debemos de dirigirnos a las opciones avanzadas y seleccionar la opción “Variables de Entorno”



*Ilustración 5 Propiedades del Sistema*

Se abrirá la ventana de variables de entorno. Una vez en esa ventana, debemos estar seguros de que exista una variable llamada “JAVA\_HOME” y que apunte a dirección del JDK más reciente y previamente instalado. (Si no existe, crearla con el botón nueva y

añadir la ruta del SDK que por lo general se encuentra en C: \Archivos de Programa\Java\)

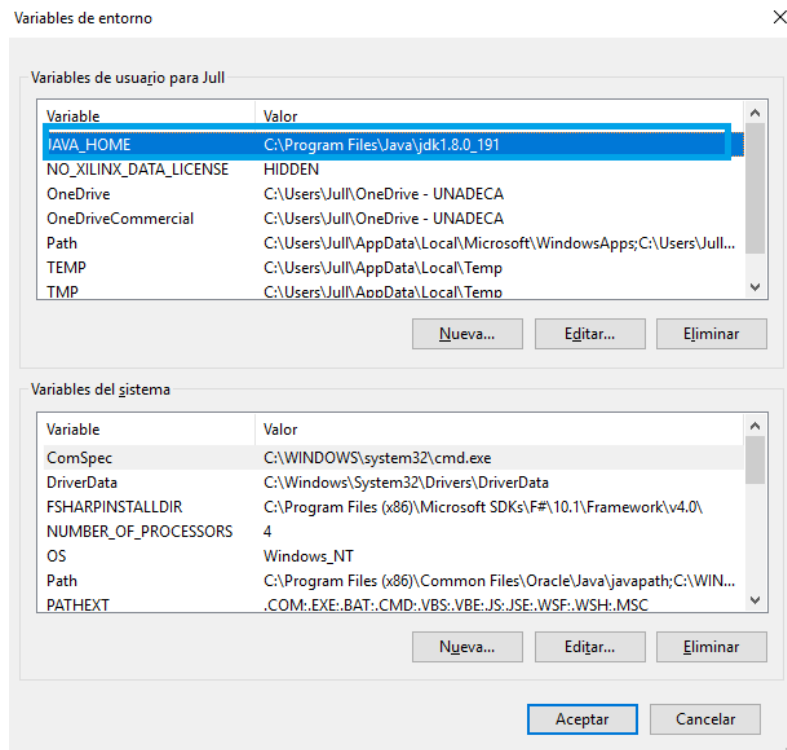


Ilustración 6 Variables del Sistema

También asegurarse de que la variable del sistema “Path”, esté creada y que se dirija exactamente en donde tenemos nuestros archivos de flutter, específicamente en la carpeta bin de este.

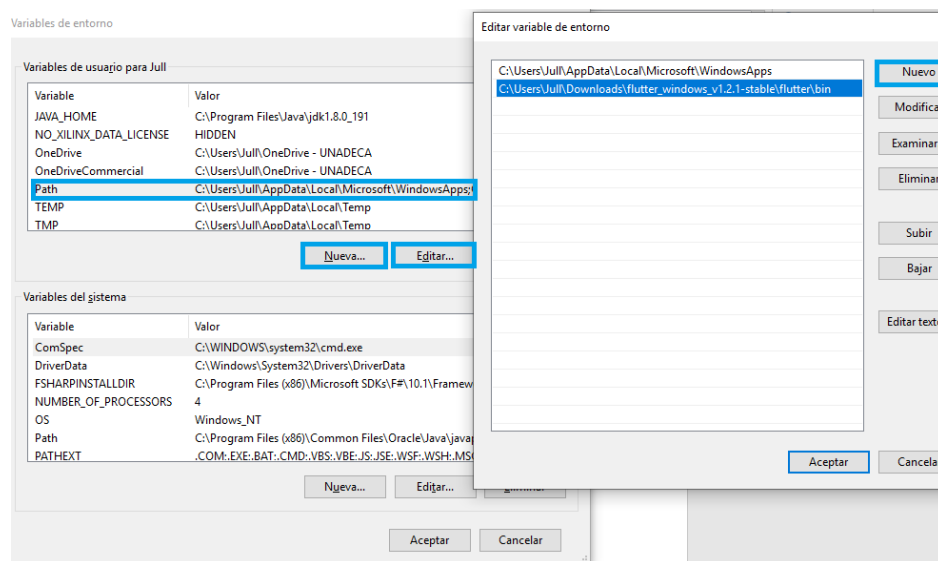
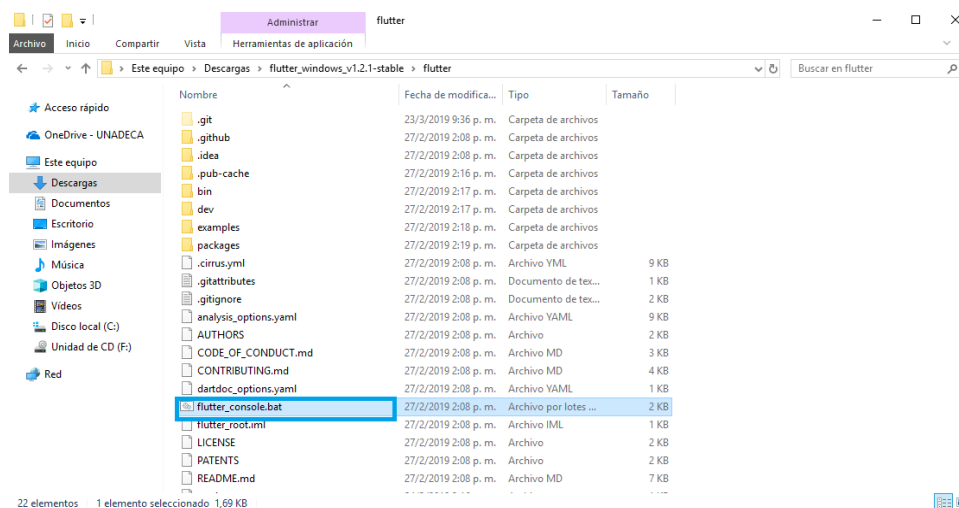


Ilustración 7 Variables del Sistema

Finalmente aceptamos y aceptamos y listo.

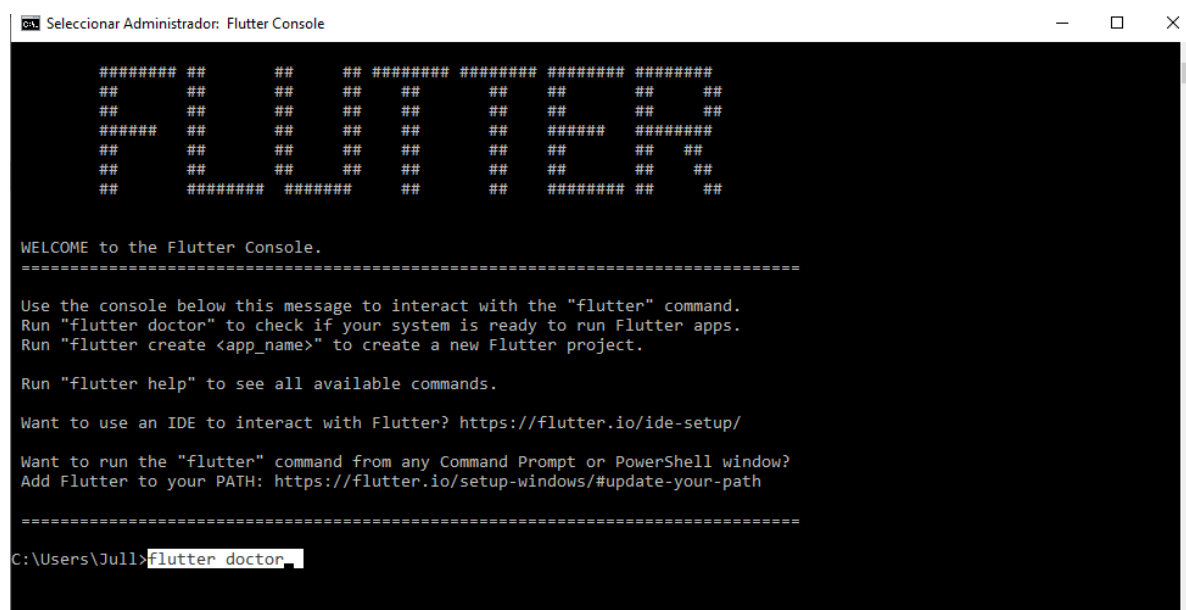
Para la siguiente fase, nos iremos al directorio donde se encuentran los archivos de flutter y localizamos el archivo el archivo “flutter\_console.bat.” Procedemos a ejecutarlo con derechos de administrador.



*Ilustración 8 Instalación Flutter*

Una vez ejecutado el archivo, nos aparecerá la consola de comandos de flutter.

En ella digitaremos el comando “flutter doctor” para ver el status de nuestros archivos y programa.



*Ilustración 9 Instalación Flutter (Consola)*

Una vez ejecutado el comando nos aparecerán los mensajes de estado del programa.

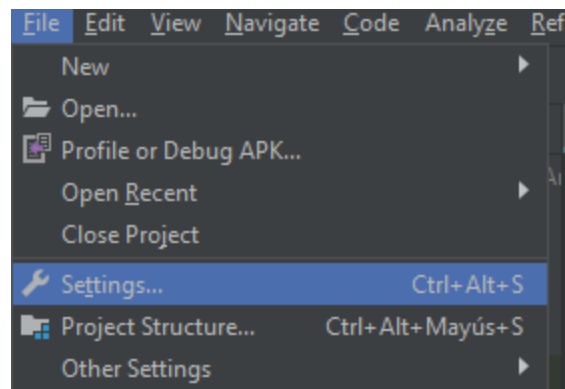
```
C:\Users\Jull>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, v1.2.1, on Microsoft Windows [VersiÃ³n 10.0.17763.379], locale es-CR)
[✓] Android toolchain - develop for Android devices (Android SDK version 28.0.3)
[✓] Android Studio (version 3.3)
[!] Connected device
    ! No devices available

! Doctor found issues in 1 category.
```

*Ilustración 10 Instalación Flutter. Verificación*

Cabe resaltar, que como yo ya poseo el programa solo me aparece que no tengo dispositivos conectados (dispositivos emulables activos). Pero la primera vez que ejecuten el comando les aparecerán checks en rojo que indicarán lo siguiente:

- ***La falta de licencia validada por el programa:*** Se soluciona escribiendo en la consola el comando “flutter doctor --android-licenses” y aceptando las licencias requeridas.
- ***Instalación de los plugin de flutter y dart:*** Para solucionarlo, nos dirigimos a Android Studio. En la parte de File > Settings > Plugins y estando en la barra de búsqueda buscamos flutter y luego dart. Procedemos a seguir los pasos para la instalación de los plugins



*Ilustración 11 Instalación Flutter*

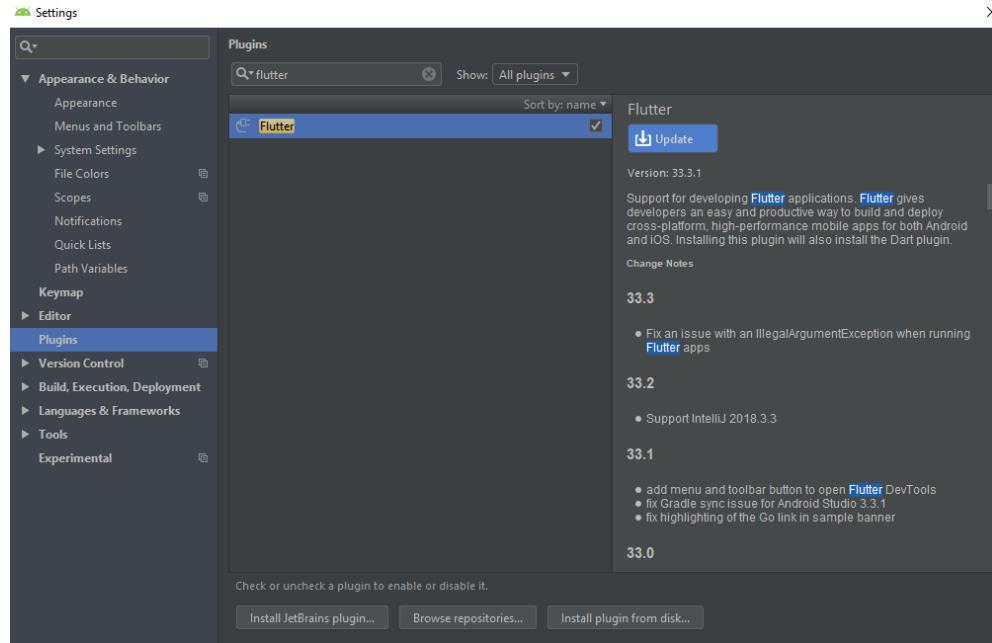


Ilustración 12 Instalación Flutter

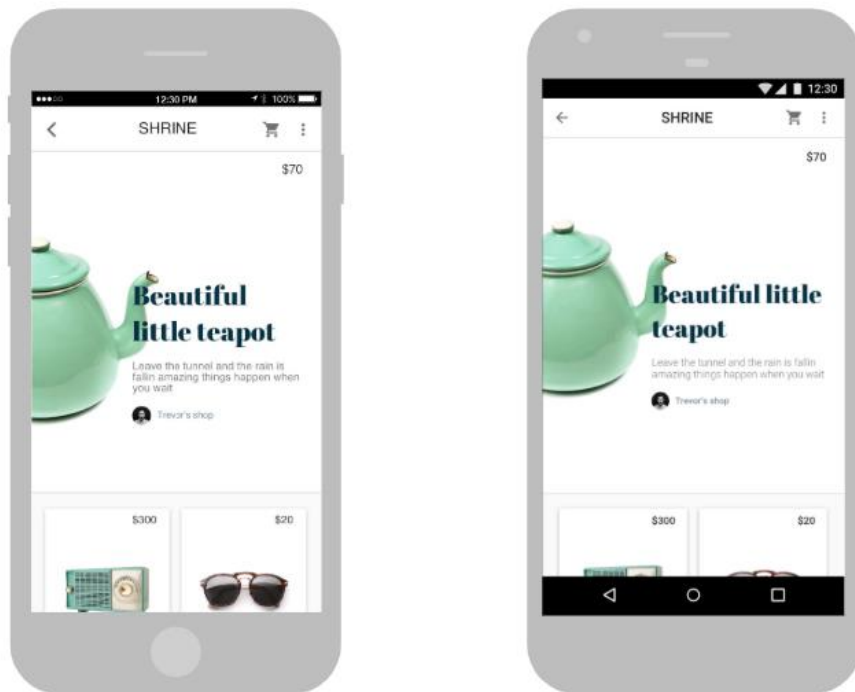
- **Versión del SDK desactualizada (en algunas ocasiones):** Simplemente ejecutamos en la consola el comando “Flutter doctor update” y luego “Flutter upgrade” (actualización del Flutter) y se nos actualizará la versión del SDK. Cabe resaltar que habría que actualizarlo en las variables del sistema.
- **Versión de Android Studio:** Esta aparecerá en rojo si no contamos con el Android Studio.

Una vez solucionados los errores, ejecutamos el comando **Flutter doctor** para ver el estado de la instalación, y veremos al completo funcionado.

## Entorno y compatibilidad

Flutter es un SDK de aplicaciones móviles para la creación de aplicaciones de alto rendimiento y alta fidelidad para iOS y Android, a partir de un único código base.

El objetivo es permitir a los desarrolladores que lancen aplicaciones de alto rendimiento que se adapten de forma natural a diferentes plataformas. Abarcamos las diferencias en los comportamientos de scrolling, tipografía, iconos, y más.



*Ilustración 13 Tomada de: <https://hackernoon.com/why-flutter-uses-dart-dd635a054ebf>*

Esta es una aplicación de demostración de la Galería, una colección de aplicaciones de ejemplo de Flutter que puedes ejecutar después de instalar Flutter y de configurar tu entorno. Shrine tiene scrolling de imágenes de alta calidad, cards interactivas, botones, listas desplegadas, y una página de carrito de compras. No se requiere experiencia en desarrollo móvil para empezar. Las aplicaciones están escritas en Dart, lo que resulta familiar si has utilizado un lenguaje como Java o JavaScript. La experiencia con lenguajes orientados a objetos es definitivamente útil, pero, incluso los no programadores han desarrollado aplicaciones Flutter.

## Lenguaje de programación (DART)

El primer equipo de Flutter evaluó más de una docena de idiomas y eligió Dart porque coincidía con la forma en que estaban creando interfaces de usuario.

Dart es una gran razón por la que los desarrolladores aman Flutter. Como dice un tweet:



*Ilustración 14 Tomada de twitter.com*

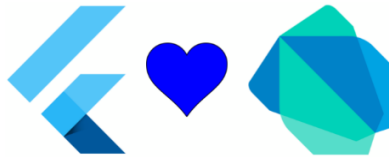
Aquí hay una lista rápida de las características de Dart que, en conjunto, lo hacen indispensable para Flutter:

- Dart es AOT (Ahead Of Time) compilado en código nativo rápido y predecible, que permite que casi todo Flutter se escriba en Dart. Esto no solo hace que Flutter sea rápido, se puede personalizar prácticamente todo (incluidos todos los widgets).
- Dart también se puede compilar con JIT (Just In Time) para ciclos de desarrollo excepcionalmente rápidos y flujos de trabajo que cambian el juego.
- Dart facilita la creación de animaciones y transiciones suaves que se ejecutan a 60 FPS. Dart puede hacer la asignación de objetos y la recolección de basura sin bloqueos. Y como JavaScript, Dart evita la programación preventiva y la memoria compartida (y por lo tanto se bloquea). Debido a que las aplicaciones Flutter se compilan a código nativo, no requieren un puente lento entre reinos (por ejemplo, JavaScript a nativo). También se inician mucho más rápido.
- Dart permite a Flutter evitar la necesidad de un lenguaje de diseño declarativo separado como JSX o XML, o constructores de interfaz visual separados, porque el

diseño declarativo y programático de Dart es fácil de leer y visualizar. Y con todo el diseño en un solo idioma y en un solo lugar, es fácil para Flutter proporcionar herramientas avanzadas que facilitan el diseño.

- Los desarrolladores han descubierto que Dart es particularmente fácil de aprender porque tiene características que son familiares para los usuarios de lenguajes estáticos y dinámicos.

No todas estas características son exclusivas de Dart, pero la combinación de ellas llega a un punto ideal que hace que Dart sea excepcionalmente poderoso para implementar Flutter. Tanto es así, es difícil imaginar que Flutter sea tan poderoso como lo es sin Dart.

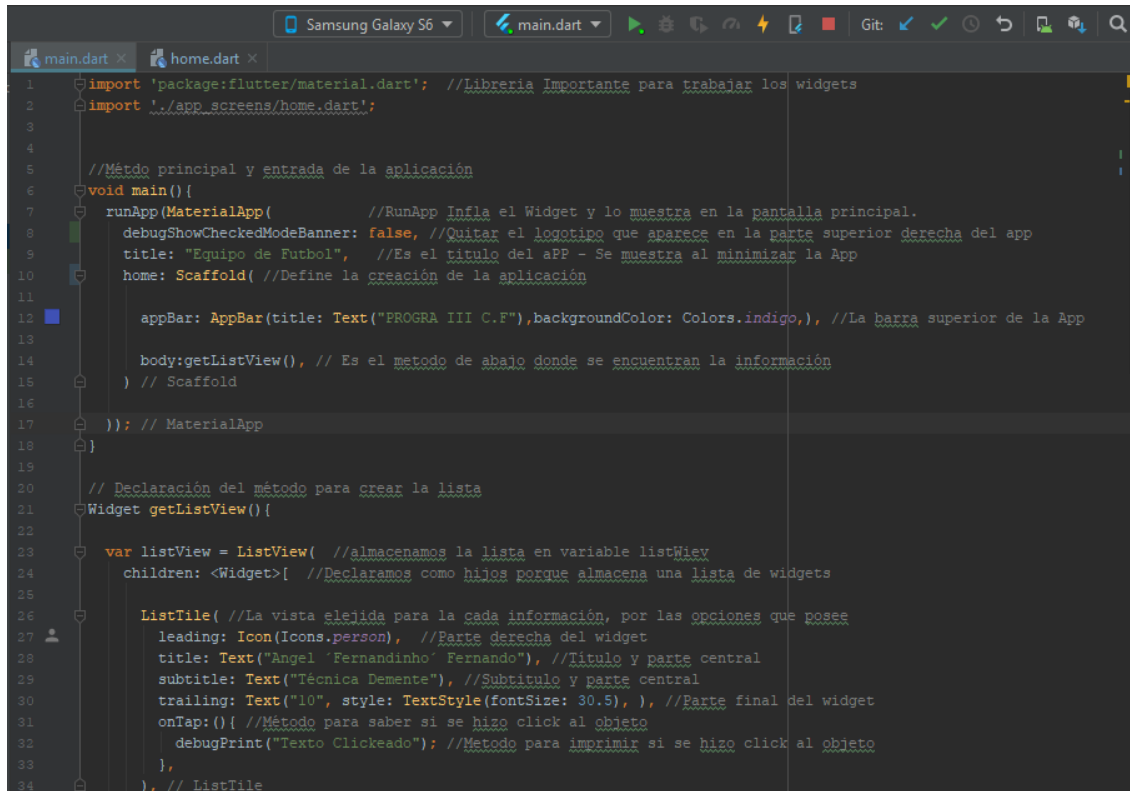


*Ilustración 15 Tomada de Google.co.cr*



## Diseño de la app

En esta sección veremos el diseño de la aplicación y el código utilizado para lograr el objetivo.

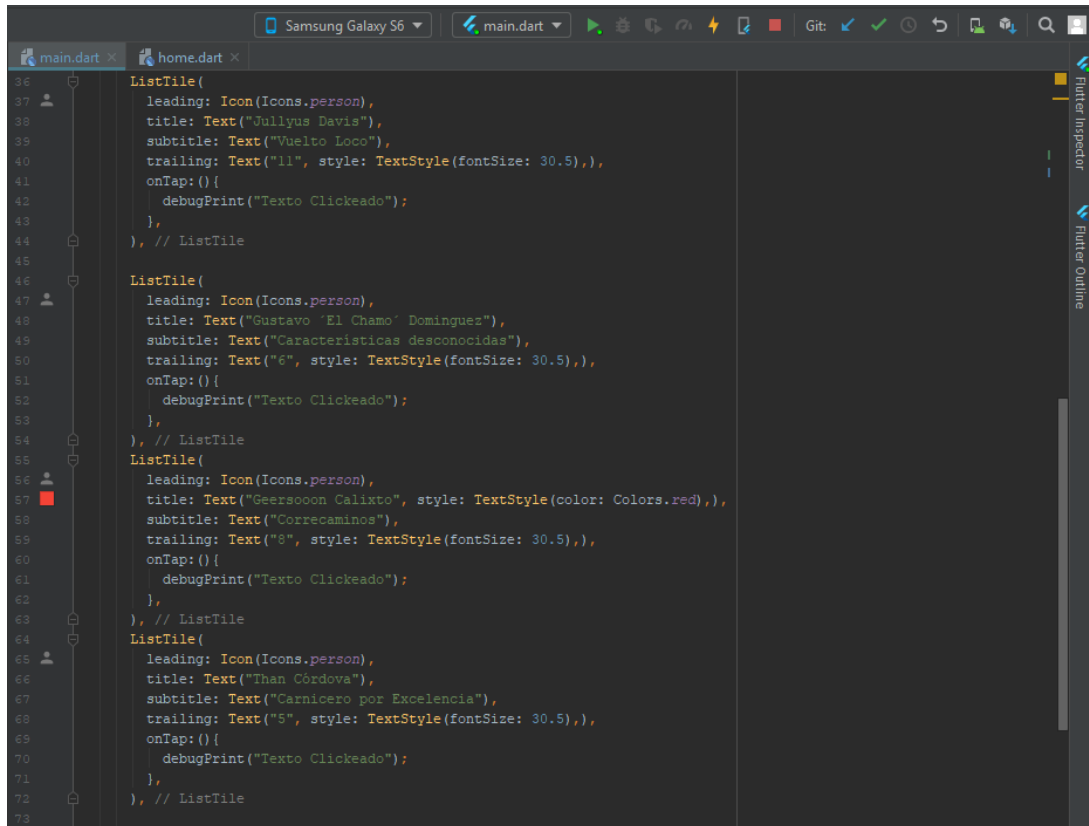


```
1 import 'package:flutter/material.dart'; //Libreria Importante para trabajar los widgets
2 import './app_screens/home.dart';
3
4
5 //Método principal y entrada de la aplicación
6 void main() {
7   runApp(MaterialApp( //RunApp Inicia el Widget y lo muestra en la pantalla principal.
8     debugShowCheckedModeBanner: false, //Quitar el logotipo que aparece en la parte superior derecha del app
9     title: "Equipo de Futbol", //Es el título del aPP - Se muestra al minimizar la App
10    home: Scaffold( //Define la creación de la aplicación
11
12      appBar: AppBar(title: Text("PROGRA III C.F"), backgroundColor: Colors.indigo, ), //La barra superior de la App
13
14      body: getListView(), // Es el método de abajo donde se encuentran la información
15    ) // Scaffold
16  )); // MaterialApp
17 }
18
19 // Declaración del método para crear la lista
20 Widget getListView() {
21   var listView = ListView( //almacenamos la lista en variable listView
22     children: <Widget>[ //Declaramos como hijos porque almacena una lista de widgets
23
24       ListTile( //La vista elegida para la cada información, por las opciones que posee
25         leading: Icon(Icons.person), //Parte derecha del widget
26         title: Text("Angel 'Fernandinho' Fernando"), //Título y parte central
27         subtitle: Text("Técnica Demente"), //Subtítulo y parte central
28         trailing: Text("10", style: TextStyle(fontSize: 30.5), ), //Parte final del widget
29         onTap: () { //Método para saber si se hizo click al objeto
30           debugPrint("Texto Clickeado"); //Método para imprimir si se hizo click al objeto
31         },
32       ), // ListTile
33     ],
34   );
```

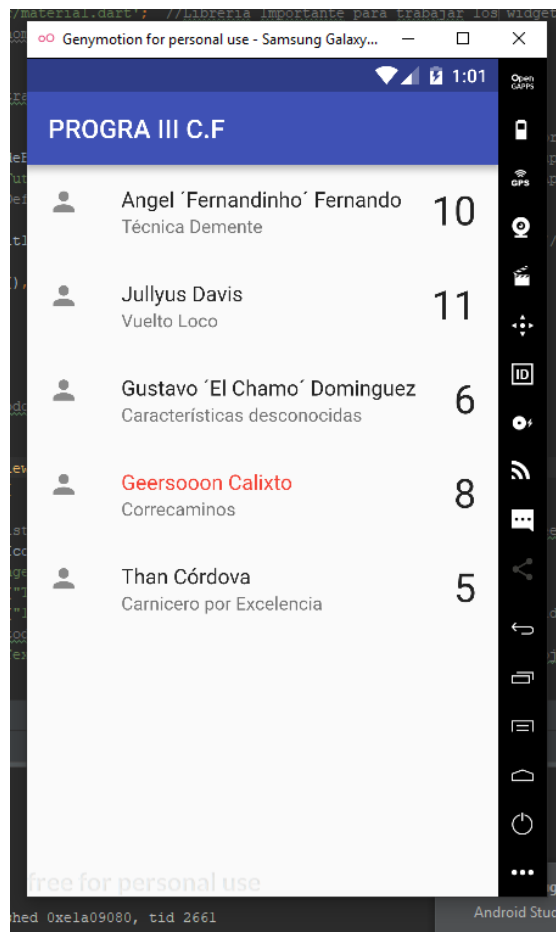
Ilustración 16. Código de la Aplicación

## Investigación Flutter

### Programación III



*Ilustración 17. Código de la Aplicación*



*Ilustración 18. Diseño de la Aplicación*

## Conclusiones

Para finalizar este documento dejamos en mente todo la información y el desarrollo de este nuevo y fabuloso framework. Trabajando e investigando sobre este llegamos a varios pensamientos concluyentes que este software en tan poco tiempo tiene un gran porvenir por el hecho de su maravilloso desarrollo y forma de poder realizar aplicaciones con grandes diseños que con poco trabajo se pueden llegar a realizar. Nos dimos cuenta de que este software poco a poco ira cambiando y revolucionando los idiomas nativos para dispositivos móviles, por su simpleza, rapidez, calidad y tecnología fácilmente personalizable.

Sentimos y llegamos a la conclusión que en par de años más este software revolucionará por completo y será uno de los lenguajes mas grande que existirán en talla global.

## Bibliografía

- Altexsoft. (Noviembre de 14 de 2018). *Altexsoft*. Obtenido de Altexsoft:  
<https://www.altexsoft.com/blog/engineering/pros-and-cons-of-flutter-app-development/>
- Baños, J. C. (28 de Febrero de 2018). *Frikipandi*. Obtenido de Frikipandi:  
<https://www.frikipandi.com/tecnologia/20180228/flutter-nuevo-framework-crear-aplicaciones-nativas-tanto-android-ios-disponible/>
- Dart. (s.f.). *Dart tutorials*. Obtenido de Dart tutorials: <https://www.dartlang.org/tutorials>
- Flutter. (s.f.). *Flutter*. Obtenido de Flutter: <https://flutter.dev/docs/get-started/install>
- Flutter. (s.f.). *Flutter docs*. Obtenido de Flutter doc: <https://flutter-es.io/docs/resources/technical-overview>
- Google. (23 de Marzo de 2018). *Google Developers*. Obtenido de Google Developers :  
<https://developers-latam.googleblog.com/2018/03/anuncio-de-flutter-beta-1-crea-bonitas.html>
- Leler, W. (Agosto de 25 de 2017). *HACKERNOON*. Obtenido de HACKERNOON:  
<https://hackernoon.com/whats-revolutionary-about-flutter-946915b09514>
- Leler, W. (26 de Febrero de 2018). *HACKERNOON*. Obtenido de HACKERNOON:  
<https://hackernoon.com/why-flutter-uses-dart-dd635a054ebf>

## ANEXOS

Ilustración 1. Tomada de <a href="https://flutter.dev/docs/get-started/install">https://flutter.dev/docs/get-started/install</a> .....	7
Ilustración 2. Tomada de <a href="https://flutter.dev/docs/get-started/install">https://flutter.dev/docs/get-started/install</a> .....	8
Ilustración 3 Instalación Flutter .....	8
Ilustración 4 Variables del Sistema Windows .....	9
Ilustración 5 Propiedades del Sistema .....	9
Ilustración 6 Variables del Sistema .....	10
Ilustración 7 Variables del Sistema .....	10
Ilustración 8 Instalación Flutter .....	11
Ilustración 9 Instalación Flutter (Consola) .....	11
Ilustración 10 Instalación Flutter. Verificación.....	12
Ilustración 11 Instalación Flutter .....	12
Ilustración 12 Instalación Flutter .....	13
Ilustración 13 Tomada de: <a href="https://hackernoon.com/why-flutter-uses-dart-dd635a054ebf">https://hackernoon.com/why-flutter-uses-dart-dd635a054ebf</a> ..	14
Ilustración 14 Tomada de <a href="https://twitter.com">twitter.com</a> .....	15
Ilustración 15 Tomada de <a href="https://www.google.co.cr">Google.co.cr</a> .....	16
Ilustración 16. Código de la Aplicación .....	17
Ilustración 17. Código de la Aplicación .....	18
Ilustración 18. Diseño de la Aplicación.....	19