

FUNDAMENTOS DE FLUTTER

LUIS ENRIQUE PINEDO CHAUPIS



PRIMER MÓDULO

Descubriendo Flutter

Presentación del docente

Ahora, conoceremos algunos aspectos relevantes del experto y del contenido a abordar en este curso.

Los temas que abordaremos son:

- ¿Quién soy?
- ¿Qué hago? Mi experiencia
- ¿Por qué es importante llevar este curso?
- ¿Qué aprenderán en este curso?



ENRIQUE PINEDO

¿Quién soy y qué hago?

Desarrollador web con más de 5 años de experiencia. Actualmente, me desempeño como full stack developer en Jumper Agency. Asimismo, tengo experiencia realizando videotutoriales para explicar sobre el mundo de la programación.



¿Por qué es importante llevar este curso?

Flutter es un conjunto de herramientas que ayudan a programar para diferentes plataformas a la vez, de manera fácil y gratuita, donde conoceremos los primeros widgets y el proceso de instalación del SDK.

También aprenderemos sobre el framework y todo esto será importante para introducirnos en el ambiente de programación.



¿Qué aprenderás en este curso?

Flutter es un kit de desarrollo de software, que ayuda a crear aplicaciones hasta en seis plataformas simultáneamente, programadas en un solo proyecto.

Descubriendo Flutter

Ahora, conoceremos algunos aspectos relevantes del experto y del contenido a abordar en este curso.

Los temas que abordaremos son:

- ¿Qué es Flutter?
- Apps hechas con Flutter

Descubriendo Flutter

Este es un marco de trabajo de código abierto desarrollado por Google, que ayuda a crear una aplicación multiplataforma compilada de forma nativa con un solo código base.



Apps hechas con Flutter

1

Nu Bank, un banco digital, evaluaron muchas opciones en función a sus prioridades principales, donde los desarrolladores móviles más experimentados de Nubank, escogieron a Flutter.

2

Xianyu una aplicación creada en China por la compañía de comercio online más grande del mundo Alibaba, ha usado Flutter para crear una gran experiencia de app para iOS y Android.

3

Hamilton Musical, la aplicación oficial del exitoso musical de Broadway, fue la primera aplicación que mostró al mundo que Flutter es una opción ideal para aplicaciones de producción a gran escala.


Conclusiones del módulo 1

01

Flutter es una herramienta fundamental para aprender programación.

02

Existen muchas aplicaciones que facilitan su trabajo usando Flutter.



SEGUNDO MÓDULO

Instalación de Flutter

Instalando Flutter en Windows

Para instalar y ejecutar Flutter, el ambiente de trabajo debe de cumplir al menos con los siguientes requerimientos:

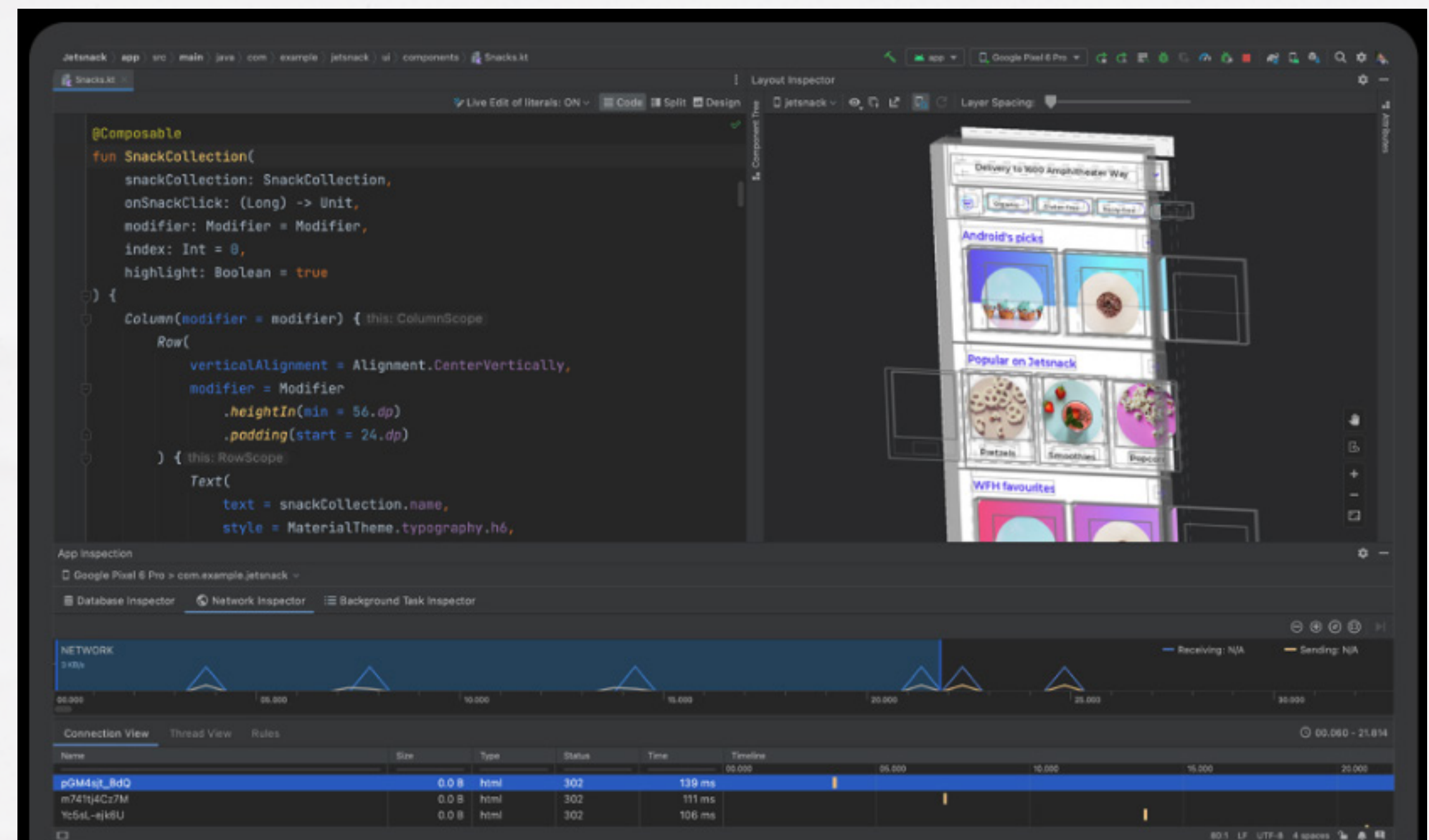
- Sistema Operativo: Windows 7 SP1 o superior (64-bit)
- Espacio en Disco: 400 MB (no incluye espacio en disco para IDE/herramientas).
- Herramientas: Flutter depende de que estas herramientas estén disponibles en tu entorno.
- Windows PowerShell 5.0 o superior (este es pre-instalado con Windows 10)

Si Git para Windows está ya instalado, asegúrate de poder correr comandos con git desde el Command Prompt o PowerShell.

Descarga Flutter para Windows: <https://esflutter.dev/docs/get-started/install/windows>

Instalando Android Studio en Windows

- Descarga e instala Android Studio.
- Inicia Android Studio, y sigue todo el 'Android Studio Setup Wizard'. Este instalará la versión más reciente de Android SDK, Android SDK Platform-Tools y Android SDK Build-Tools.
- Todas estas son necesarias ya que Flutter las necesita cuando se desarrolla para Android.



Instalando Android Studio en Windows

Visual Studio Code es un editor de código fuente ligero pero potente que se ejecuta en su escritorio y está disponible para Windows, macOS y Linux, en este caso aprenderemos a descargar para Windows

Viene con soporte integrado para JavaScript, TypeScript y Node.js y tiene un rico ecosistema de extensiones para otros lenguajes y tiempos de ejecución (como C++, C#, Java, Python, PHP, Go, .NET). Comience su viaje con VS Code con estos videos introductorios.

Descarga e instala **Visual Studio Code**.

Fuente: <https://code.visualstudio.com/docs/?dv=osx>

```
4  var server = express();
5  server.use(bodyParser.json);
6
7  server.g
8    get (property) Application.get: ((name: string)...
9    getMaxListeners
10   arguments
11   engine
12   length
13   merge
14   purge
15   settings
16   toString
17   defaultConfiguration
18
```

```
Launch + [server.js server]
VARIABLES
  Local
    __dirname: "/Users/wa...
    __filename: "/Users/wa...
    bodyParser: function ...
    database: Object
    exports: Object
    express: function cre...
    module: Module
    PORT: 3002
  WATCH
  CALL STACK PAUSED ON S...
    (anonymous function)
    Module._compile
    Module._extensions..js

43
44
45
46
47 var PORT = 3002;
48 server.listen(PORT, function() {
49   console.log('listening at %s', PORT);
50 });
51
52
53
```

```
29
30 server.get('/todos', function(req, res, next) {
31   database.getAll(function(todos) {
32
33     getAll(function (data) {
34       var todos = _.filter(data.todos, function (todo) {
35         return todo.id != id;
36       });
37     });
38     data.todos = todos;
39     commit(data);
40     resolve(data);
41   });
42 }
43
```


Instalando Flutter en macOS

Para instalar y ejecutar Flutter, tu entorno de trabajo debe cumplir los siguientes requisitos mínimos:

- Sistema Operativo: macOS (64-bit)
- Espacio en disco: 700 MB (Esto no incluye el espacio en disco para IDE/tools).
- Herramientas: Flutter necesita que estas herramientas de línea de comando estén disponibles en tu entorno.

- | | | | |
|---------|-------|---------|---------|
| □ bash | □ rm | □ curl | □ which |
| □ mkdir | □ git | □ unzip | |

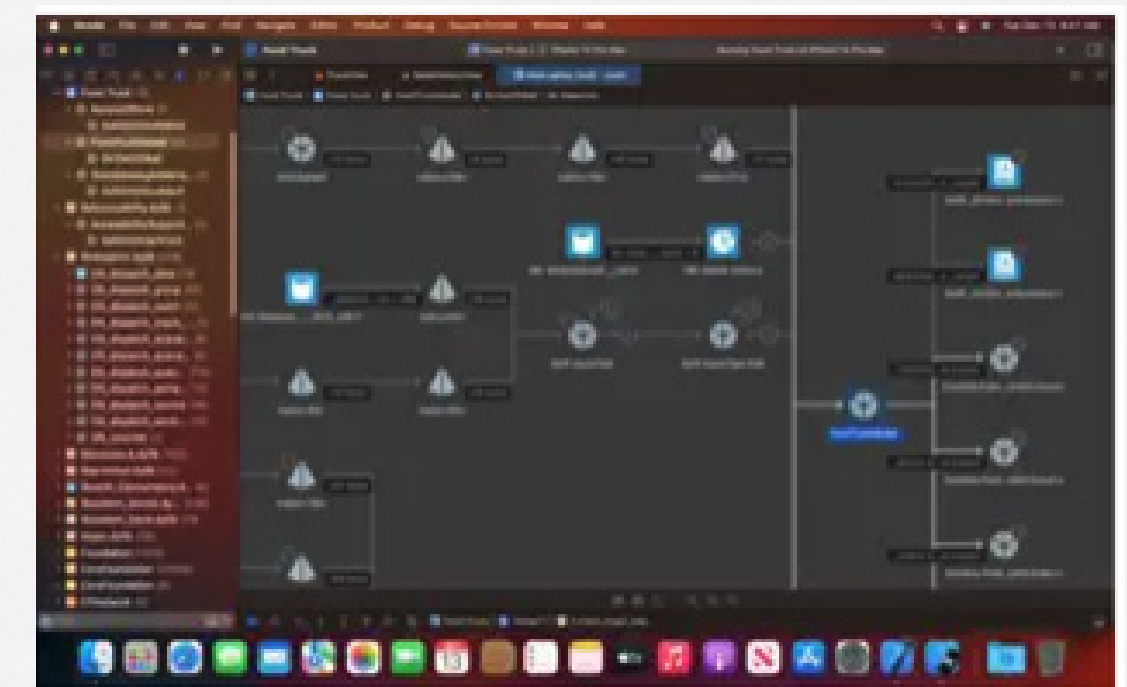
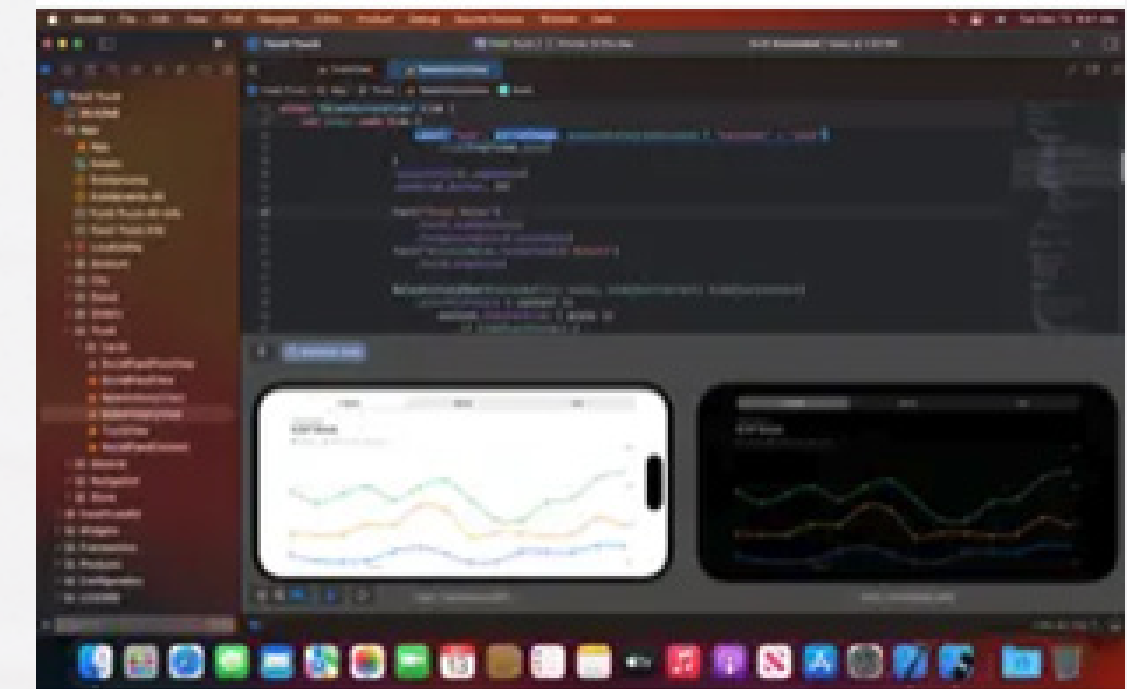
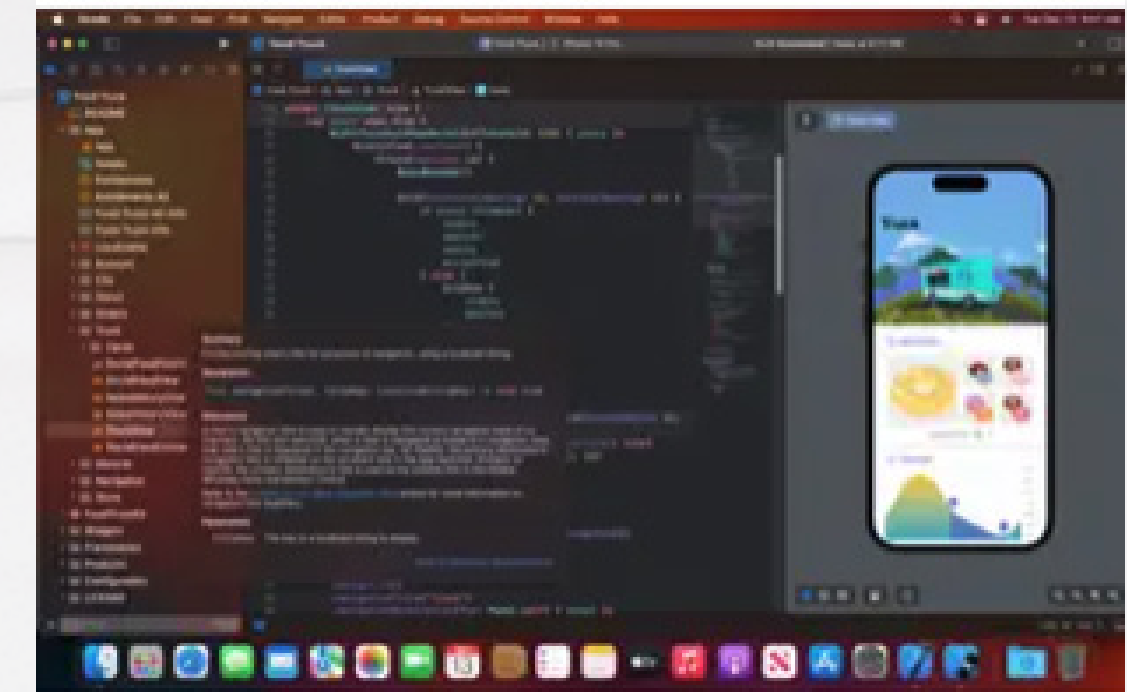
Descarga Flutter en Mac: <https://esflutter.dev/docs/get-started/install/macos>

Configurando Xcode

Xcode, es un conjunto de herramientas para desarrolladores que permite crear apps para Mac, iPhone, iPad, Apple Watch y Apple TV.

Xcode combina las funcionalidades de diseño de interfaz de usuario, programación, pruebas, depuración y envío a App Store en un flujo de trabajo unificado.

Descarga aquí [Xcode](#).



Android Studio en macOS

Para instalar Android Studio en tu Mac, sigue estos pasos:

- Ejecuta el archivo DMG de Android Studio.
- Arrastra y suelta Android Studio en la carpeta Aplicaciones y, luego, inícialo.
- Elige si deseas importar configuraciones anteriores de Android Studio y, luego, haz clic en OK.
- Completa el Setup Wizard de Android Studio, que incluye la descarga de los componentes del SDK de Android que se necesitan para el desarrollo.

Descarga e instala **Android Studio en Mac**:

<https://developer.android.com/studio/install?hl=es-419#mac>

Instalando Android Studio en macOS

Visual Studio Code es un editor de código fuente ligero pero potente que se ejecuta en su escritorio y está disponible para Windows, macOS y Linux. , en este caso aprenderemos a descargar para Mac

Viene con soporte integrado para JavaScript, TypeScript y Node.js y tiene un rico ecosistema de extensiones para otros lenguajes y tiempos de ejecución (como C++, C#, Java, Python, PHP, Go, .NET). Comience su viaje con VS Code con estos videos introductorios.

Descarga e instala **Visual Studio Code**.

Fuente: <https://code.visualstudio.com/docs/?dv=osx>

```
4  var server = express();
5  server.use(bodyParser.json);
6
7  server.g
8    get (property) Application.get: ((name: string)...
9    getMaxListeners
10   arguments
11   engine
12   length
13   merge
14   purge
15   settings
16   toString
17   defaultConfiguration
18
```

```
Launch  server.js server
VARIABLES
  Local
    __dirname: "/Users/wa...
    __filename: "/Users/wa...
    bodyParser: function ..
    database: Object
    exports: Object
    express: function cre...
    module: Module
    PORT: 3002
  WATCH
  CALL STACK  PAUSED ON S...
    (anonymous function)
    Module_compile
    Module_extensions.js
43
44
45
46
47  var PORT = 3002;
48  server.listen(PORT, function() {
49    console.log('listening at %s', PORT);
50  });
51
52
53
```

```
29
30  server.get('/todos', function(req, res, next) {
31    database.getAll(function(todos) {
32
33      getAll(function (data) {
34        var todos = _.filter(data.todos, function (todo) {
35          return todo.id != id;
36        });
37      });
38      data.todos = todos;
39      commit(data);
40      resolve(data);
41    });
42  });
43
44
```


Conclusiones del módulo 2

01

Vimos la importancia de las descargas de diferentes aplicaciones que funcionarán para poder iniciar el camino del desarrollo con Flutter.

02

La importancia de Android para nuestro simulador.

03

Tener un dispositivo con las diferentes características ya descritas será necesario para empezar nuestro proyecto.



TERCER MÓDULO

Primeros pasos con Flutter

Todo es un widget

Un widget, es un elemento básico de la interfaz de usuario de toda app en Flutter, este nos muestra o proporciona una forma en que el usuario pueda interactuar con la aplicación, estos incluyen iconos, menús, botones, ventanas, formularios, entre otros para poder mostrar información o responder a una acción.



Widgets Básicos

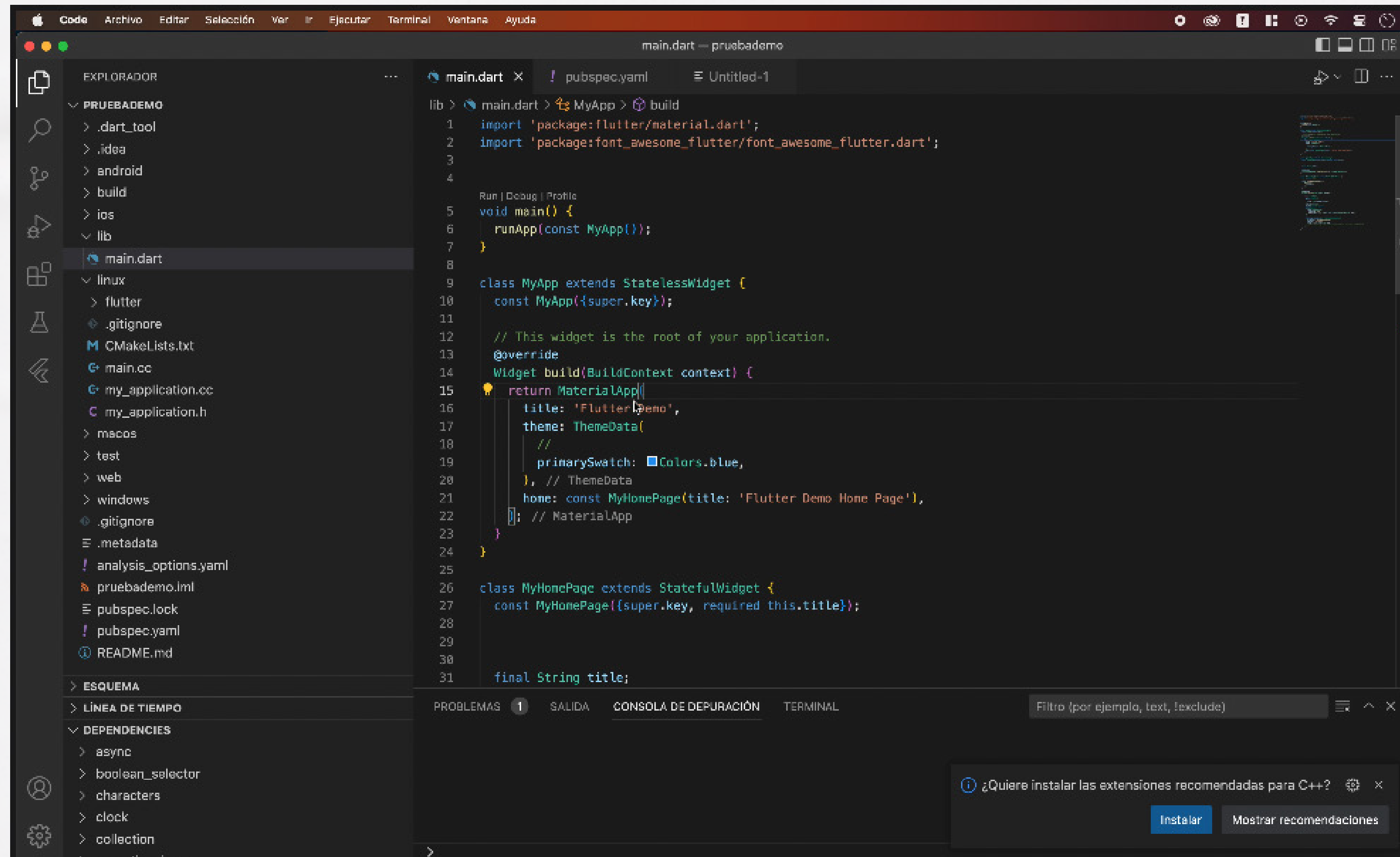
Flutter viene con un conjunto de potentes widgets básicos que son:

- **Text:** Crea una cadena de texto con estilo, dentro de tu aplicación.
- **Row, Column:** Son widgets flexibles te permiten crear layouts flexibles tanto en la dirección horizontal (Row) como en la vertical (Column).

Widgets Básicos

- **Stack:** Te permite apilar los widgets uno encima del otro en el orden como se pintan. Los Stacks se basan en el modelo de layout de posicionamiento absoluto en la web.
- **Container:** Te permite crear un elemento visual rectangular, el cual puede tener un fondo, borde, sombra, también márgenes, relleno interno y restricciones aplicadas a su tamaño. Asimismo, se puede transformar en un espacio tridimensional utilizando una matriz.

Estructura de un proyecto en Flutter



Instalando Android Studio en Windows

- ¿Qué es yaml?, es un lenguaje de serialización similar a XML o Json, este lenguaje es fácil de entender y de fácil lectura como una de las características más notables que tiene.
- Además puede manejar datos de tipo string, números y booleanos, podemos crear variables, objetos y también listas, y es importante respetar las mayúsculas, minúsculas, espacios, tabs y los puntos.
- Usamos pubspec.yaml para dos propósitos diferentes:
 - El primero, es para especificar archivos como imágenes, audio, video y fuentes utilizadas en el proyecto.
 - El segundo, es para especificar las bibliotecas que son las dependencias que hemos utilizado dentro del proyecto, antes de comenzar a explicar el contenido de este archivo.

Conclusiones del módulo 3

01


Vimos la importancia de los widgets para realizar programación y aprendimos de estos.

02

Es necesario conocer la estructura de un proyecto base , para tener la forma predeterminada y podamos realizar nuestro proyecto.

03

Pubspect.yaml cumple un rol importante para realizar nuestro proyecto.



CUARTO MÓDULO

Primeros Widgets

El Widget Container

Este widget te permite crear un elemento visual rectangular.

Un container puede estar decorado con un boxdecoration, con lo que podríamos definir un fondo, un borde o una sombra, también puede tener márgenes, rellenos internos y restricciones aplicadas a su tamaño.

El widget container intenta respetar la alineación, adaptarse al tamaño de su hijo(o child) , te ayuda a componer, decorar y colocar widgets hijos (o child).

El Widget Text

TEXT

Se refiere a una cadena de texto o string que cuenta con un solo estilo, cabe resaltar que la cadena, puede dividirse en múltiples líneas o también podremos mostrarla en una misma línea, dependerá de las restricciones que tenga nuestro diseño.

TEXTSTYLE

Técnicamente no es un widget, este se combina con widgets, es una clase que estiliza el texto desde el contorno, el color y el tamaño de la cadena de texto.

El Widget Icon

El widget de icono gráfico se dibuja con un glyph de una fuente que se describe en la opción de IconData.

No son interactivos, pero en caso que se requiera la interactividad se puede hacer uso del IconButton.

Es de formato cuadrado.

Widget Row & Column

Son uno de los elementos más básicos sobre el posicionamiento de widgets en un layout.

Row

Permite crear una fila verticalmente en donde estarán nuestros widgets. Contiene sus elementos secundarios o sus elementos en la dirección horizontal desde la izquierda a la derecha.

Column

Permite crear una columna horizontalmente.

Contiene sus elementos secundarios o los elementos en dirección de arriba a abajo

Overflow y Widget Expanded

Overflow

Se denomina Overflow cuando se produce un problema de desbordamiento al colocar un widget en un lienzo que no cabe en el espacio disponible (ya sea a lo ancho, alto o ambos).

Widget Expanded

Le da la facilidad de expandirse para llenar el espacio disponible dentro del eje principal.

Se puede definir que tan estricta puede ser el tamaño, especificando un ajuste.

Conclusiones del módulo 4

01

Todos los widgets mencionados cumplen un rol de gran importancia y es necesario saber sus significados.

02

El widget Expanded será clave para nuestro diseño y fácil uso del simulador.



QUINTO MÓDULO

Más widgets

Widget Image

El Widget Image es el que nos permitirá agregar imágenes estáticas y dinámicas a una página o a nuestro proyecto.

Hay muchas formas en las que Flutter puede obtener una imagen para mostrar, la imagen puede provenir de los recursos incluidos en tu app.

Widget Image

Estos son los constructores que usaremos para las diversas formas en que se puede especificar una imagen:

- `Image.new`, para obtener una imagen de un `ImageProvider`
- `Image.asset`, para obtener una imagen de un `Asset`
- `Image.network`, para obtener una imagen de una URL
- `Image.file`, para obtener una imagen de un archivo
- `Image.memory`, para obtener una imagen de un `Uint8List`

Un `Image` admite los siguientes formatos de imagen: JPEG, PNG, GIF, GIF animado, WebP, WebP animado, BMP y WBMP.

Widget Stack

Nos permite crear una capa con varios widgets, colocándolos uno encima del otro.

Necesitamos una forma de superponer widgets uno sobre el otro, por ejemplo, es posible que deseemos mostrar texto sobre una imagen, por lo que para lograr este resultado haremos uso del widget Stack.

Widget Scaffold

Nos proporciona una estructura predeterminada basada en Material Design que nos sirve para organizar nuestras pantallas.

Dentro de éste widget podemos agregar todo tipo de elementos visuales de una manera más sencilla en lugar de ir acoplando un número de diferentes widgets.

El Scaffold es una estructura a manera de andamio el cual ya trae opciones como un Floating Action button, Appbar, drawer entre otros.

Widget SafeArea

El widget SafeArea maneja un child dando el relleno necesario para evitar que la barra del sistema o las esquinas redondeadas tapen a sus widgets.

Su uso se debe a que no en todos los dispositivos existen pantallas rectangulares, existen algunos que tienen las esquinas redondeadas e incluso algunos tienen pantalla curva por lo que generaría una mala experiencia de usuario si la aplicación que creaste tiene íconos o texto escrito fuera de esta curva o borde redondeado de la pantalla, lo que hará que no sea visible parte de tus elementos.

Widget ListView & LisTile

Para ocasiones en las que queramos desplazarnos horizontal o verticalmente haciendo scroll podemos hacer uso de este Listview.

Es un widget de diseño al igual que el Column y Row.

ListView.builder

Se usa cuando queremos crear una lista recursivamente sin escribir código una y otra vez. Crea una matriz lineal desplazable de widgets.

ListView.builder de forma predeterminada no admite la reordenación de elementos secundarios.

Conclusiones del módulo 5

01

Todos los widgets son importantes para realizar nuestro proyecto.

02

Mientras más uses los widgets y uses los correctos, será mucho más fácil desarrollar tu simulador.

03

El list.builder es necesario para crear una lista, sin escribir código una y otra vez.

Bibliografía

- <https://www.geeksforgeeks.org/listview-builder-in-flutter/>
- <https://barcelonageeks.com/flutter-widget-listtile>
- <https://flutter.dev/showcase/nubank>

NETZUN