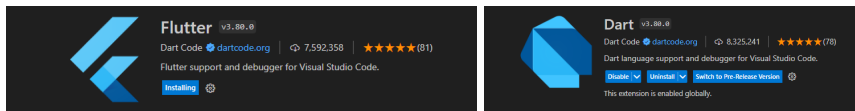


# Introducción a Flutter

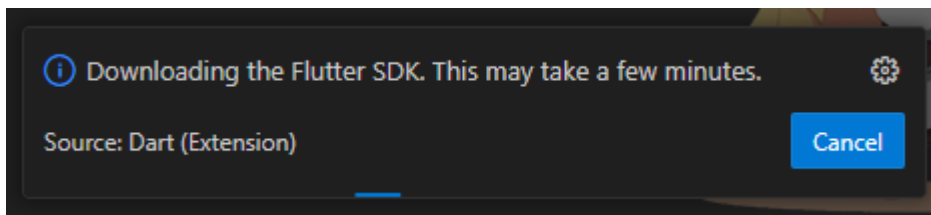
## Instalación y configuración VSCode

### Instalación de plugins



### Descargando e instalando el SDK Flutter

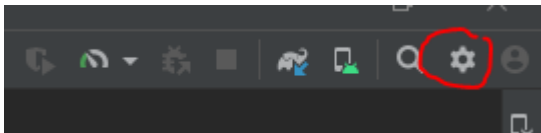
Cuando se crea un proyecto nuevo, con el comando `Ctrl+Mayús+P` → Flutter: Nuevo Proyecto, dará un mensaje de error indicando que el SDK de Flutter no está instalado. Recomendará descargarlo desde el propio VSCode, en el mismo mensaje de error. A continuación, pedirá que indiquemos la ruta donde se guardarán los recursos del sdk. Una vez indicado se genera la instalación.

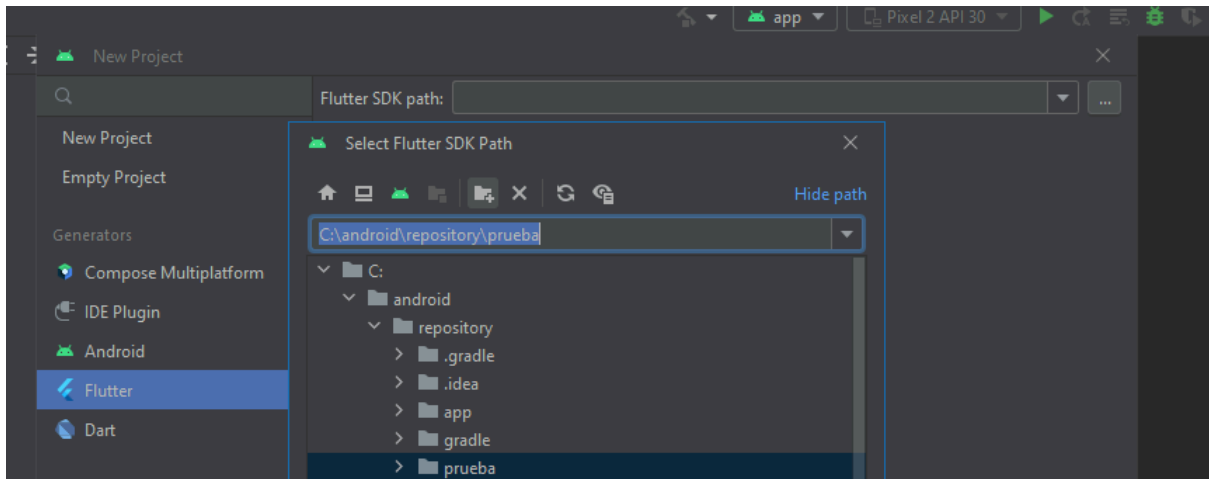


Una vez esperado unos minutos la aplicación lanza un código de estado y una serie de datos informativos como la versión de VSCode, etc.

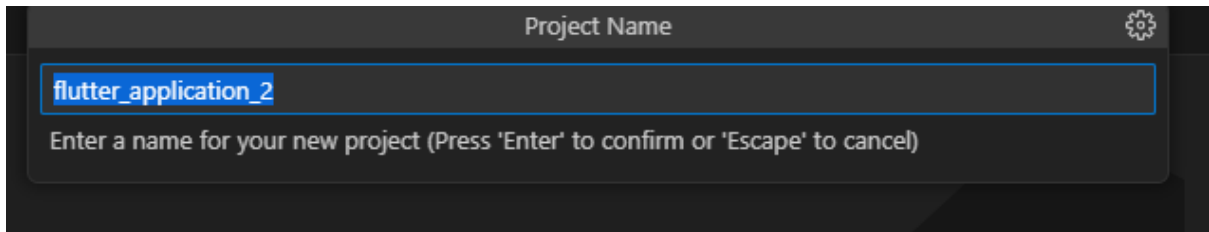
## Instalación y configuración Android Studio

### Instalación de plugins

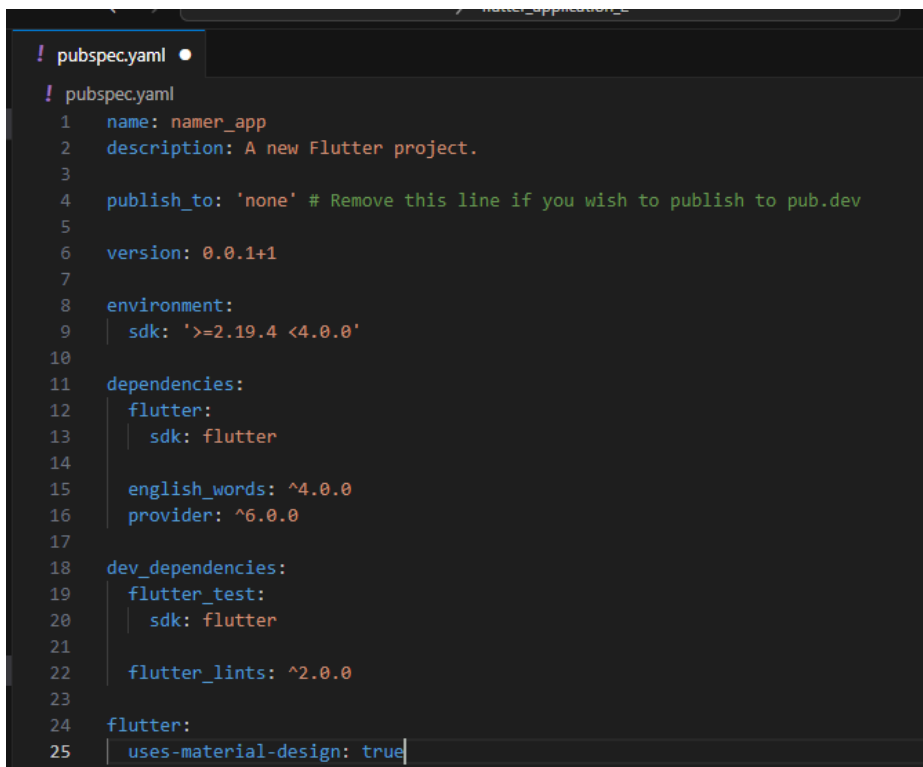




## Creando un nuevo proyecto en VSCode



## Realizando modificaciones en el pubspec.yaml:



Realizando modificaciones en analysis\_options.yaml:

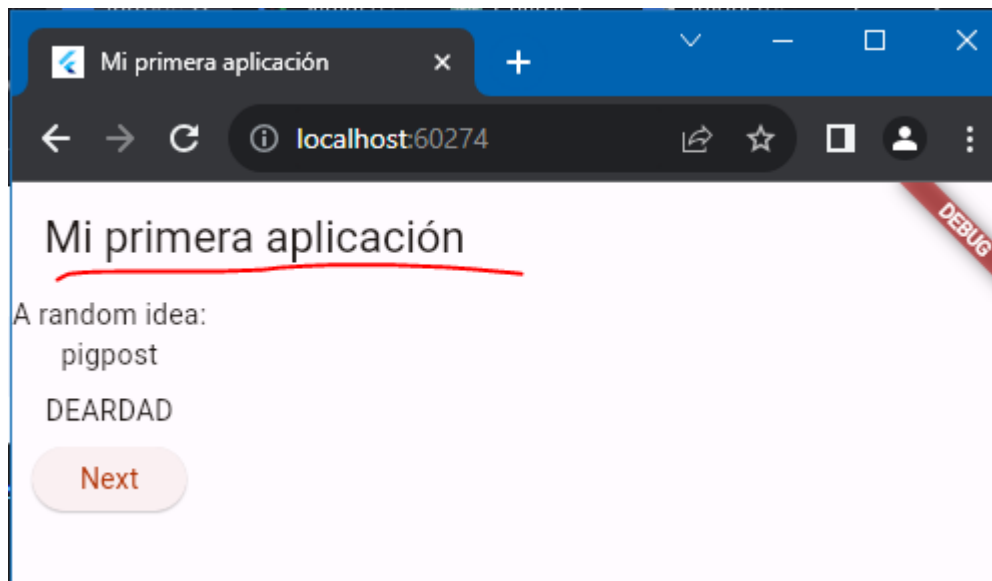
```
! analysis_options.yaml
1  include: package:flutter_lints/flutter.yaml
2
3  linter:
4    rules:
5      prefer_const_constructors: false
6      prefer_final_fields: false
7      use_key_in_widget_constructors: false
8      prefer_const_literals_to_create_immutables: false
9      prefer_const_constructors_in_immutables: false
10     avoid_print: false
11
```

Poniendo título

MyHomePage es una clase donde se crea el cuerpo de la aplicación. Dentro del método build es donde se construye y devuelve, en este caso, un Scaffold. Como argumentos pasamos el appBar que es la superior de nuestra app, donde pondremos el título con el nombre que aparece:

```
class MyHomePage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    var appState = context.watch<MyAppState>();
    var appStateCurrent = appState.current;
    var appStateCurrentDos = appState.currentDos;

    return Scaffold(
      appBar: AppBar(
        title: Text("Mi primera aplicación"),
      ), // AppBar
    );
  }
}
```



### Cambiando título con botón

Para cambiar el título agregaremos un botón. Para ello primero creamos un objeto de la clase `MyAppState` que nos servirá para escuchar todos los cambios de estado que se realicen en la app.

```
class MyHomePage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    var appState = context.watch<MyAppState>();
    var appStateCurrent = appState.current;
    var appStateCurrentDos = appState.currentDos;

    return Scaffold(
      appBar: AppBar(
        title: Text(appState.tituloChange),
      ), // AppBar
    );
  }
}
```

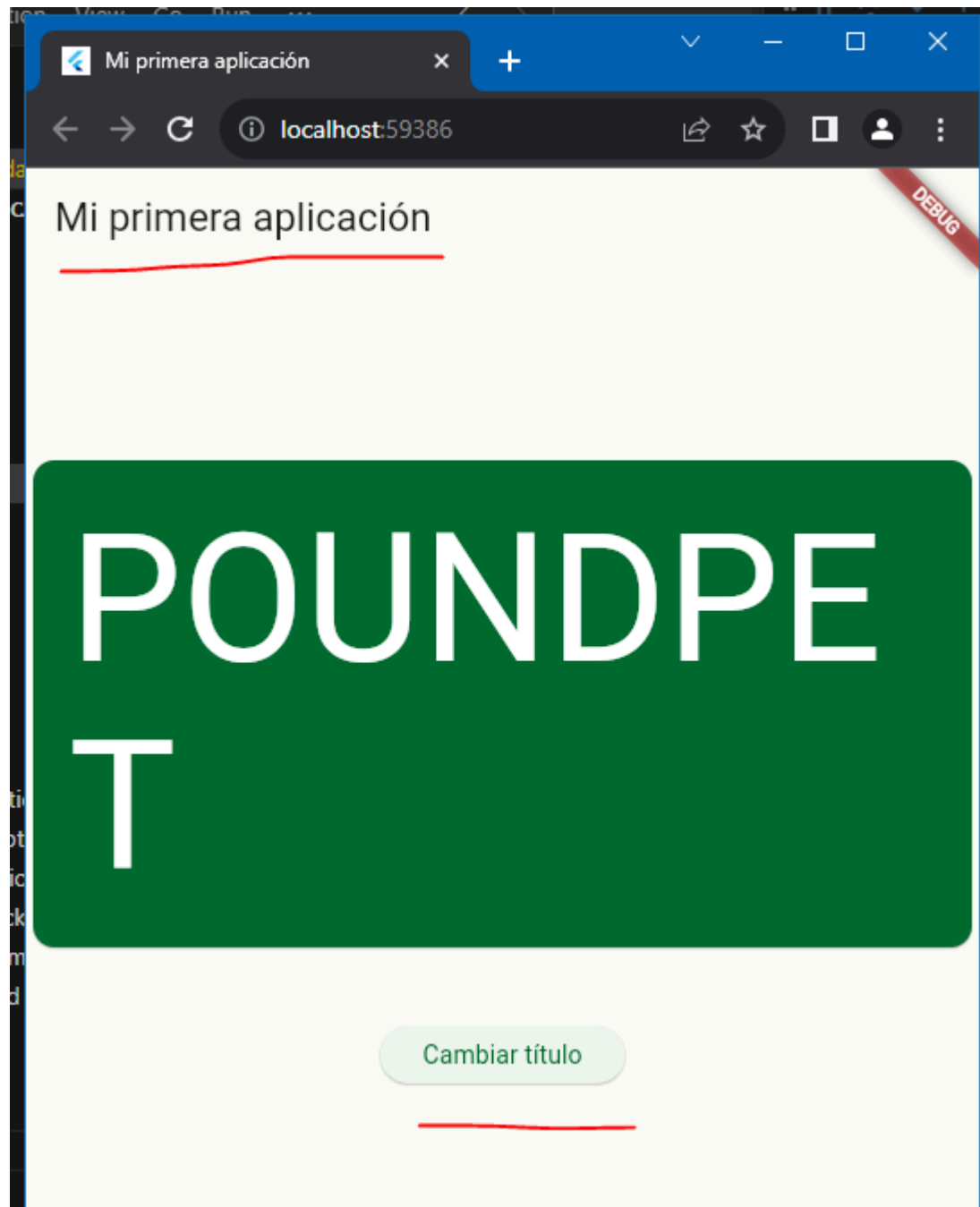
En la imagen se ve en la parte de `appBar` que modificamos el título. `appState` está llamando a una variable llamada `tituloChange`. Esta variable se crea en el método `changeTitle` de la clase `MyAppState`, que analizaremos más adelante.

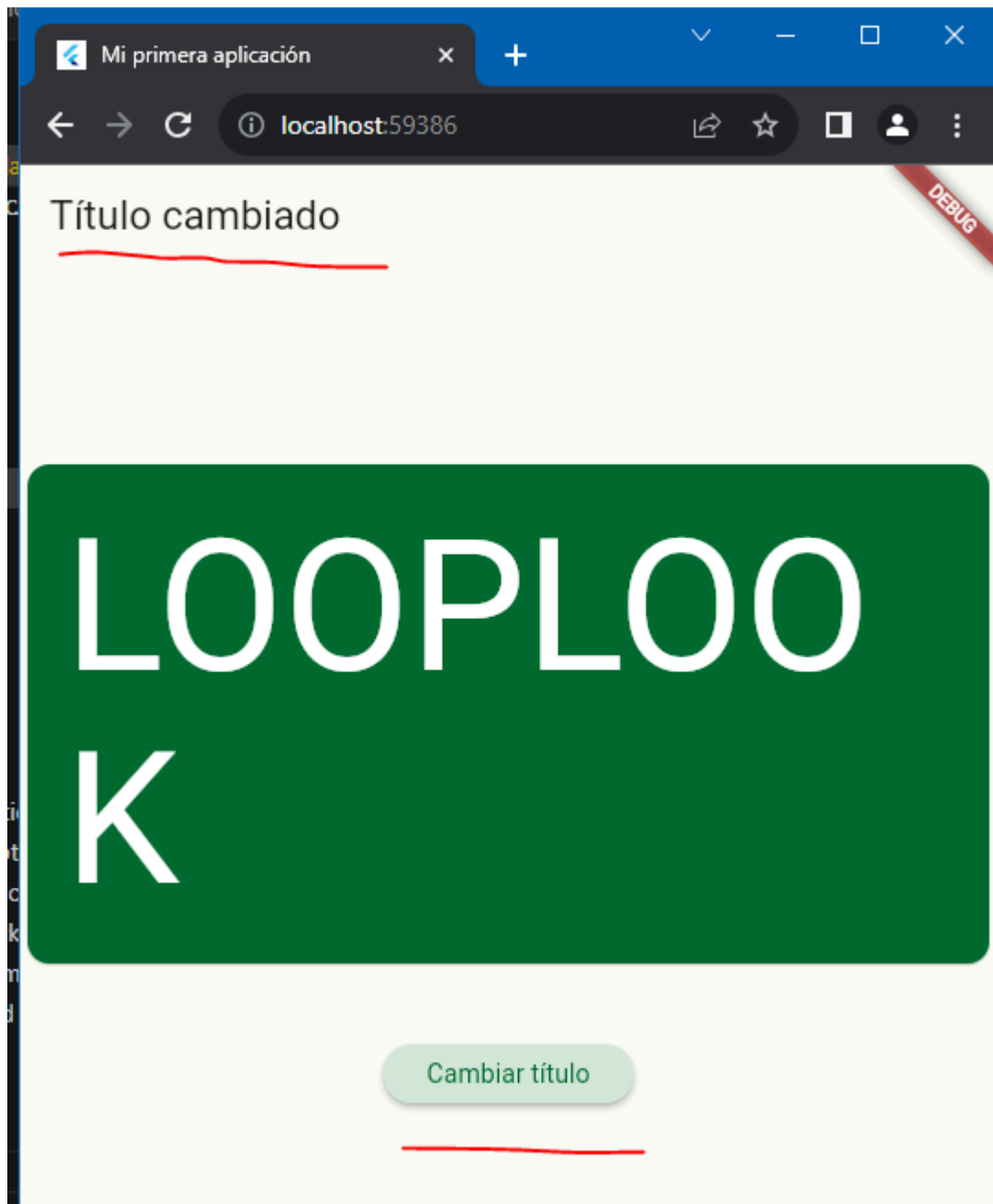
A continuación añadimos el botón usando ElevatedButton donde se le pasa por argumento el onPressed, que indica qué hacer cuando se presione el botón, y child que indica el elemento hijo, en este caso es un Text, que será el nombre del botón:

```
body: Center(  
  child: Column(  
    mainAxisAlignment: MainAxisAlignment.center,  
    children: [  
      BigCard(appStateCurrent: appStateCurrent),  
      SizedBox(height: 40), //Esto era un Text. Co  
      ElevatedButton(  
        onPressed: () {  
          appState.getNext();  
          appState.changeTitule();  
        },  
        child: Text("Cambiar título")  
      ), // ElevatedButton
```

Vamos con la función changeTitle. Nos dirigimos a MyAppState y creamos la variable tituloChange. Mas adelante creamos el método changeTitule y volvemos a cambiar su valor. Ahora con estos elementos podemos hacer que se cambie tituloChange, pero para que el botón funcione, necesitamos introducir notifyListeners() que será el encargado de notificar a los widgets que están escuchando para saber el estado de la App. Es decir, cada vez que se realice un cambio se notificará y MyHomePage, que es quien escucha, será quien modifique esa información.

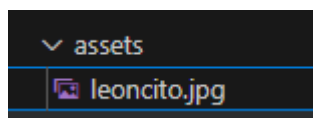
```
class MyAppState extends ChangeNotifier {  
  var current = WordPair.random();  
  var currentDos = WordPair.random();  
  String tituloChange = "Mi primera aplicación";  
  
  void getNext(){  
    current = WordPair.random();  
    currentDos = WordPair.random();  
    notifyListeners();  
  }  
  
  void changeTitule(){  
    tituloChange = "Título cambiado";  
    notifyListeners();  
  }  
}
```





## Añadiendo una imagen

Primero creamos un repositorio donde guardamos nuestras imágenes.



Modificamos el archivo pobspec.yaml. Le estaremos indicando la ruta de las imágenes.

```
flutter:  
  uses-material-design: true  
  assets:  
    - assets/
```

A continuación añadimos en el cuerpo Image.asset() y ponemos la ruta. Es importante haber realizado el paso anterior ya que de esta manera es capaz de encontrar los archivos.

```
body: Center(  
  child: Column(  
    mainAxisAlignment: MainAxisAlignment.center,  
    children: [  
      BigCard(appStateCurrent: appStateCurrent),  
      SizedBox(height: 40), //Esto era un Text. Cor  
      ElevatedButton(  
        onPressed: () {  
          appState.getNext();  
          appState.changeTitule();  
        },  
        child: Text("Cambiar título")  
      ), // ElevatedButton  
      Image.asset(  
        'assets/leoncito.jpg'), // Image.asset
```



Mi primera aplicación

DEBUG

# POINTZONE

Cambiar título

