

Módulo Programación.

1º DAW.



PRÁCTICA 1: Instalación de Flutter.
UNIDAD DE TRABAJO 8.
Profesor: Pedro Antonio Santiago Santiago.

1. Introducción.

En esta práctica se instala el SDK de Flutter (incluido el compilador Dart), se configura el entorno, ya sea actualizando el PATH o configurando VSC y se describen y utilizan las herramientas más usadas de este SDK.

También se instala la extensión disponible para VSC, se crea una aplicación a partir de una plantilla y se trabaja con las principales opciones de la extensión.

Por último se modifica el programa para mostrar el famoso “Hola Mundo”.

2. Desarrollo de la práctica.

2.1. Parte 1. Descargando Flutter.

Ir a la página web de Flutter y descargar el archivo comprimido:

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

Descomprimir.

Es necesario tener Windows Power Shell y Git,

```
Developer PowerShell for VS 2022

*****
** Visual Studio 2022 Developer PowerShell v17.5.5
** Copyright (c) 2022 Microsoft Corporation
*****
PS C:\Users\Pedro\source\repos> git
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
      [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
      [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
      [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
      <command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
  clone      Clone a repository into a new directory
  init       Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
  add        Add file contents to the index
  mv         Move or rename a file, a directory, or a symlink
  restore    Restore working tree files
  rm         Remove files from the working tree and from the index
  sparse-checkout  Initialize and modify the sparse-checkout
```

Al ejecutar el comando flutter en la línea de comandos se obtiene error.

```
PS C:\Users\Pedro> flutter
flutter : El término 'flutter' no se reconoce como nombre de un cmdlet, función, archivo de script o programa
ejecutable. Compruebe si escribió correctamente el nombre o, si incluyó una ruta de acceso, compruebe que dicha ruta
es correcta e inténtelo de nuevo.
En línea: 1 Carácter: 1
+ flutter
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (flutter:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException
```

¿A qué es debido?

2.2. Parte 2. Actualizando el Path.

Cuando se ejecuta un comando en el terminal, pueden suceder 2 cosas:

- Que el comando o “programa” se encuentre en esa carpeta (normalmente se ha de colocar el ./, para referirse al directorio actual), con lo que se ejecutará.
- El comando está en otra carpeta y el sistema operativo a de encontrarlo, pero es imposible buscar en todo el sistema, por lo que posee una lista de “lugares” en los que buscar los comandos introducidos, a esta lista se le denomina PATH y es una de las variables de entorno más usadas en los sistemas operativos.

Los ejecutables de flutter se encuentran en CARPETA_DESCOMPRIMIDA/bin, por ejemplo, si se va a la carpeta y se ejecuta flutter en el directorio bin, se obtiene:

```
PS C:\flutter\bin> pwd
Path
----
C:\flutter\bin
PS C:\flutter\bin> ls

Directorio: C:\flutter\bin

Mode                LastWriteTime         Length Name
----                -
d-----         04/05/2023   9:32             cache
d-----         04/05/2023   9:27            internal
d-----         04/05/2023   9:27             mingit
-a----         04/05/2023   9:26           2159 dart
-a----         04/05/2023   9:26           1488 dart.bat
-a----         04/05/2023   9:26           2391 flutter
-a----         04/05/2023   9:26           2544 flutter.bat

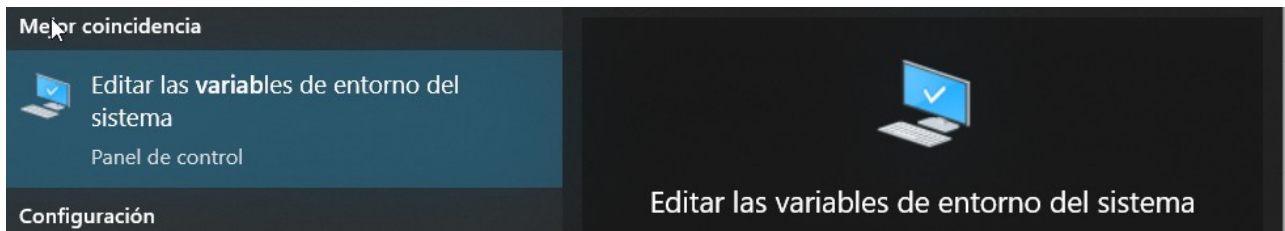
PS C:\flutter\bin> .\flutter.bat
Manage your Flutter app development.

Common commands:

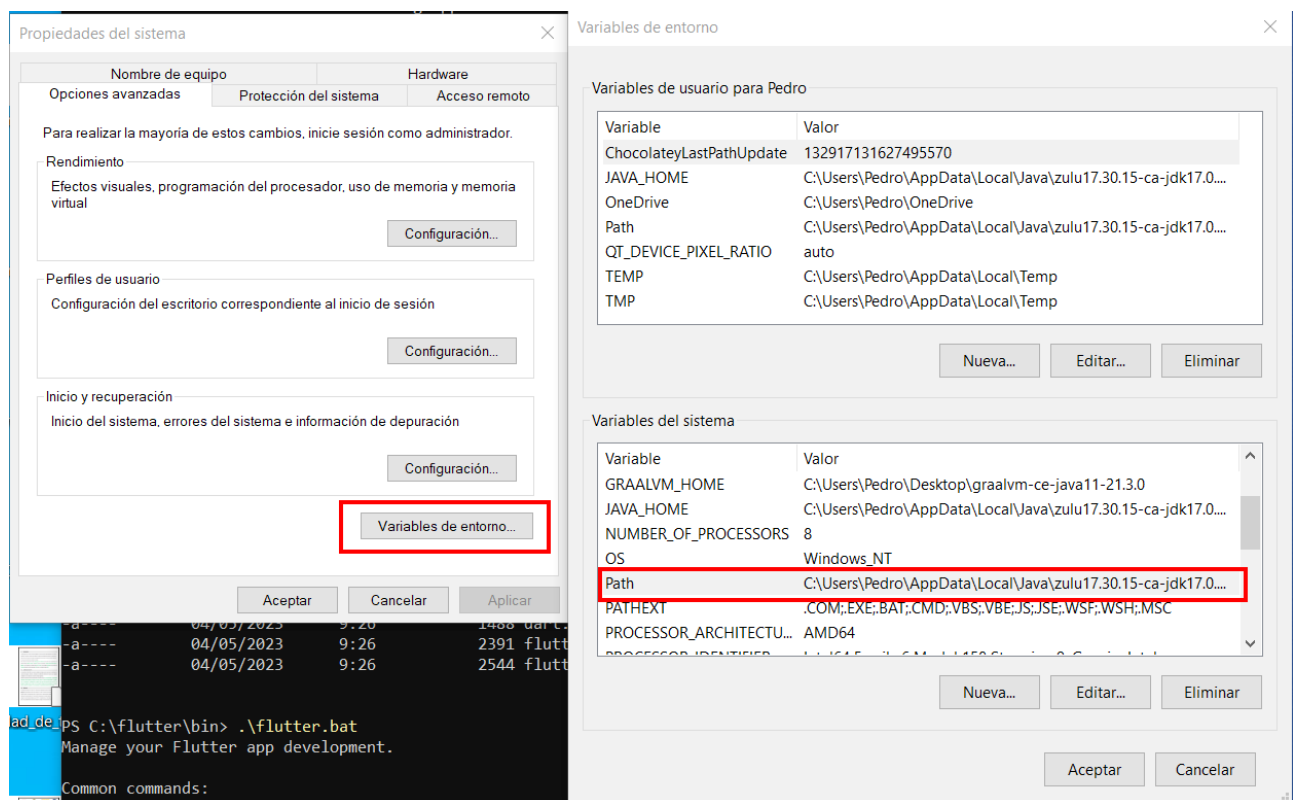
flutter create <output directory>
    Create a new Flutter project in the specified directory.
```

En Windows:

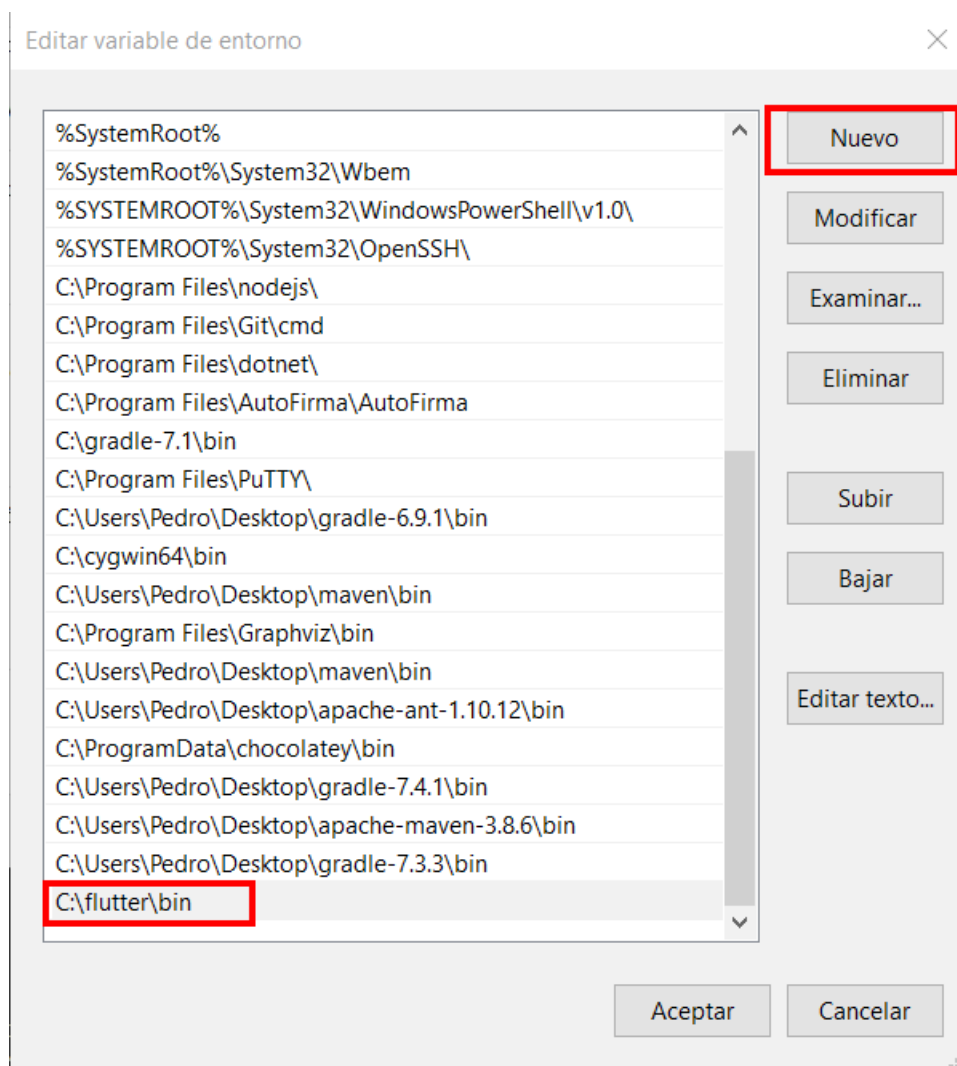
Buscar en el cuadro de búsqueda: Variables de entorno:



Pulsar botón **Variables de entorno**:



Buscar en variables del sistema la variable Path y pulsar editar, añadiendo la ruta en la que se encuentra la carpeta bin de Flutter.



Abrir **un nuevo terminal** y ejecutar el comando flutter.

```
C:\Users\Pedro>flutter
Manage your Flutter app development.

Common commands:

  flutter create <output directory>
    Create a new Flutter project in the specified directory.

  flutter run [options]
    Run your Flutter application on an attached device or in an emulator.

Usage: flutter <command> [arguments]

Global options:
-h, --help                Print this usage information.
-v, --verbose              Noisy logging, including all shell commands executed.
                           If used with "--help", shows hidden options. If used
                           with "--verbose", shows diagnostic information. (Use "-vv" to force verbose logging.)
-d, --device-id            Target device id or name (prefixes allowed).
--version                 Reports the version of this tool.
--suppress-analytics       Suppress analytics reporting when this command runs.

Available commands:
```

El comando Flutter ofrece diferentes opciones, desde crear un proyecto, hasta ejecutar una aplicación, pasando por la ejecución de los test. Los más usados:

- **flutter create <nombre_proyecto>**: Crea un nuevo proyecto de Flutter con el nombre especificado.
- **flutter run**: Ejecuta la aplicación de Flutter en el dispositivo o emulador conectado. Flutter detectará automáticamente si estás trabajando en una aplicación iOS o Android y lanzará la aplicación en consecuencia.
- **flutter build**: Compila la aplicación de Flutter para el despliegue. Puedes agregar opciones adicionales para compilar la aplicación para iOS o Android, como flutter build apk para compilar un archivo APK para Android.
- **flutter doctor**: Verifica la instalación de Flutter y muestra un informe de cualquier configuración faltante o problemas que puedan afectar el desarrollo.
- **flutter pub get**: Descarga las dependencias del proyecto desde el archivo pubspec.yaml y las instala en tu proyecto.
- **flutter clean**: Elimina los archivos generados en la última compilación. Puede ser útil cuando encuentras problemas de compilación o conflictos de dependencias.
- **flutter test**: Ejecuta los casos de prueba definidos en el proyecto.
- **flutter packages get**: Actualiza las dependencias del proyecto a las versiones más recientes, de acuerdo con el archivo pubspec.yaml.

Un comando interesante es flutter doctor, que muestra la información de la configuración y las plataformas para las que se puede crear una aplicación.

```
PS C:\Users\Pedro\source\repos> flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.7.12, on Microsoft Windows [Versi n 10.0.19045.2965], locale es-ES)
[X] Windows Version (Unable to confirm if installed Windows version is 10 or greater)
[!] Android toolchain - develop for Android devices (Android SDK version 30.0.3)
    X cmdline-tools component is missing
      Run `path/to/sdkmanager --install "cmdline-tools;latest"`
      See https://developer.android.com/studio/command-line for more details.
    X Android license status unknown.
      Run `flutter doctor --android-licenses` to accept the SDK licenses.
      See https://flutter.dev/docs/get-started/install/windows#android-setup for more details.
[✓] Chrome - develop for the web
[✓] Visual Studio - develop for Windows (Visual Studio Professional 2022 17.5.5)
[✓] Android Studio (version 2021.2)
[✓] VS Code (version 1.79.0)
[✓] Connected device (3 available)
[✓] HTTP Host Availability
```

En la imagen anterior indica, entre otra informaci n:

La versi n de Flutter.

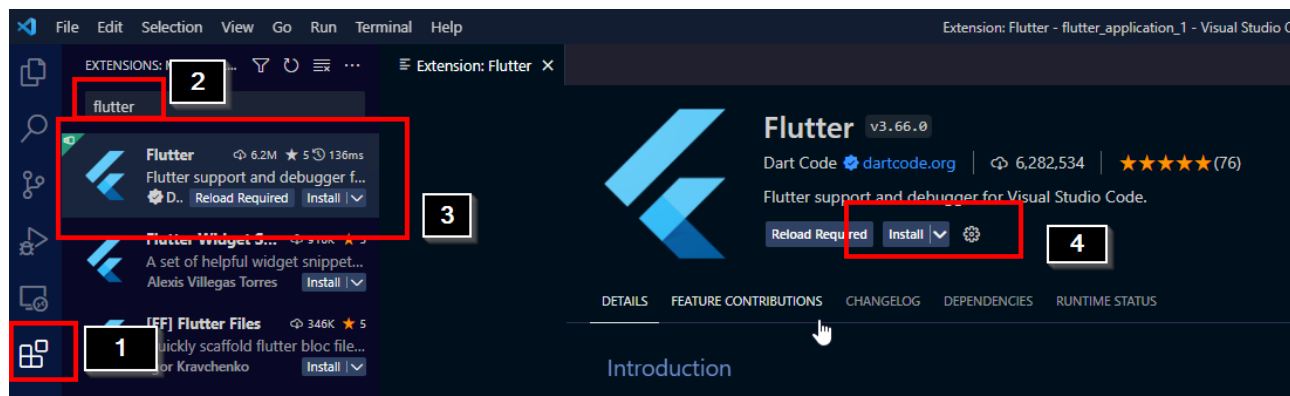
No se dispone de las herramientas para crear aplicaciones Andriod.

Puede generar y probar aplicaciones Web (Chrome).

Se encuentra instalado

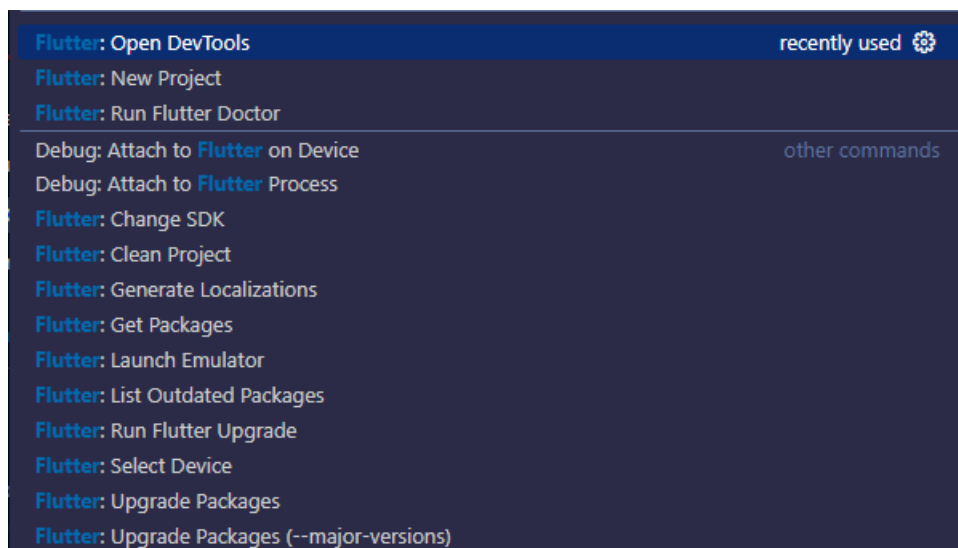
2.3. Parte 3. Configurando el IDE.

Se opta por usar VSC para el desarrollo, ya que posee una extensi n para Flutter, adem s en caso de no poder modificar la variable del entorno (por ejemplo equipos del aula donde no se tiene permisos de administrador):



VSC no utiliza los t picos men s de los entornos de desarrollo, sino que se basa en que las extensiones a aden comandos que se pueden ejecutar, incluyendo un buscador de

estos, para acceder al buscador pulsar : Ctrl+Shift+P apareciendo en la parte superior central un cuadro de búsqueda, al teclear flutter, aparecen las opciones disponibles:



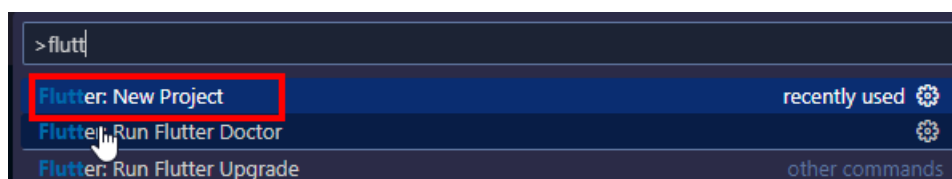
La documentación de la extensión se puede encontrar en:

[Visual Studio Code | Flutter](#)

Desde el entorno se puede acceder a un terminal, pero al ejecutar el comando flutter no lo encuentra:

```
PS C:\Users\Pedro\Desktop\ejemplo3> flutter
flutter : El término 'flutter' no se reconoce como nombre de un cmdlet, fu
acceso, compruebe que dicha ruta es correcta e inténtelo de nuevo.
En línea: 1 Carácter: 1
+ flutter
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (flutter:String) [], Command
+ FullyQualifiedErrorId : CommandNotFoundException
```

Es necesario por tanto indicar en que lugar se encuentra el SDK, se pueden modificar las propiedades de la extensión, pero una manera más sencilla es simular la creación de un proyecto Flutter, pulsar Ctrl+Ship+P y buscar Flutter, selecciondo “**New Project**”, apareciendo en la parte inferior un cuadro de diálogo para configurar el path del SDK.



2.4. Parte 4. Creando la primera aplicación con Flutter.

El comando flutter permite crear estructuras para un proyecto, en el directorio del proyecto insertar el comando:

flutter create nombre_aplicación

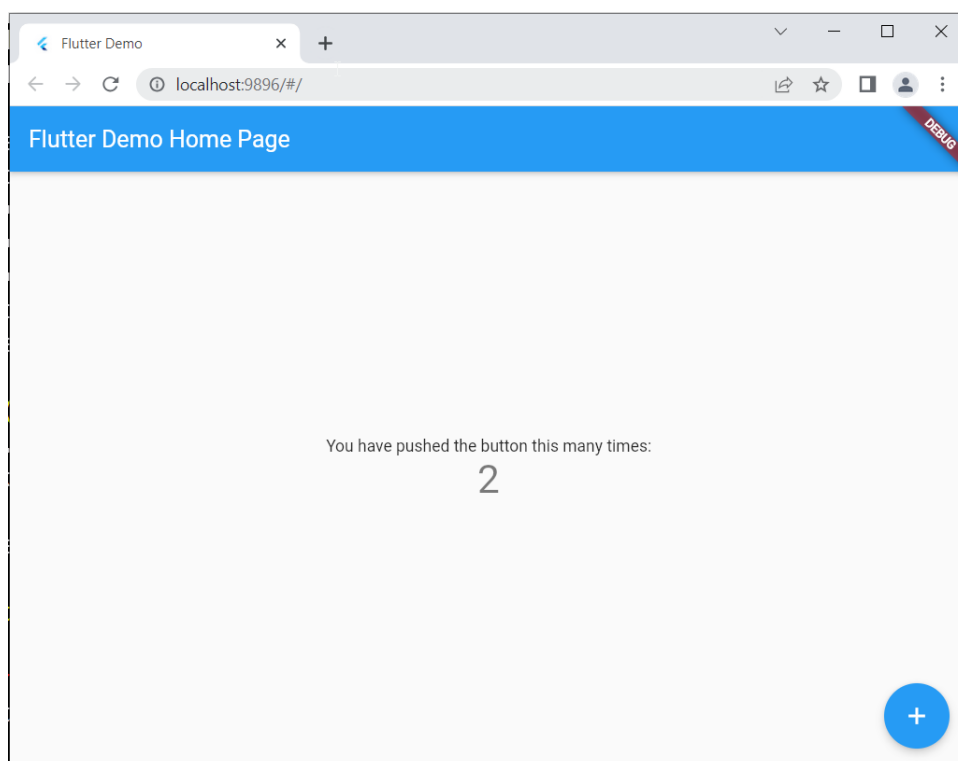
Entrar en el directorio y ejecutar:

flutter run

Preguntando, de las plataformas disponibles en cuales se quiere ejecutar:

```
PS C:\Users\Pedro\ejemplo> flutter run
Multiple devices found:
Windows (desktop) • windows • windows-x64 • Microsoft Windows [Version 10.0.19045.2965]
Chrome (web)      • chrome • web-javascript • Google Chrome 113.0.5672.127
Edge (web)        • edge   • web-javascript • Microsoft Edge 114.0.1823.43
[1]: Windows (windows)
[2]: Chrome (chrome)
[3]: Edge (edge)
Please choose one (To quit, press "q/Q"):
```

Al seleccionar, por ejemplo 2:



Para generar una aplicación usable:

flutter build <tipo>

Donde tipo indica la plataforma, por ejemplo para Windows:

flutter build windows

```
PS C:\Users\Pedro\ejemplo> flutter build windows

Building with sound null safety

Building Windows application...
```












En la carpeta build\windows se ha generado un proyecto C++ para Windows (que es posible abrir con Visual Studio (el completo)), con código C++.

```
ejemplo (Ámbito global)
1  #include <flutter/dart_project.h>
2  #include <flutter/flutter_view_controller.h>
3  #include <windows.h>
4
5  #include "flutter_window.h"
6  #include "utils.h"
7
8  int APIENTRY wWinMain(_In_ HINSTANCE instance, _In_opt_ HINSTANCE prev,
9                      _In_ wchar_t *command_line, _In_ int show_command) {
10     // Attach to console when present (e.g., 'flutter run') or create a
11     // new console when running with a debugger.
12     if (!::AttachConsole(ATTACH_PARENT_PROCESS) && ::IsDebuggerPresent()) {
13         CreateAndAttachConsole();
14     }
15
16     // Initialize COM, so that it is available for use in the library and/or
17     // plugins.
18     ::CoInitializeEx(nullptr, COINIT_APARTMENTTHREADED);
19
20     flutter::DartProject project(L"data");
21
22     std::vector<std::string> command_line_arguments =
23         GetCommandLineArguments();
24
25     project.set_dart_entrypoint_arguments(std::move(command_line_arguments));
26
27     FlutterWindow window(project);
28     Win32Window::Point origin(10, 10);
29     Win32Window::Size size(1280, 720);
30     if (!window.Create(L"ejemplo", origin, size)) {
31         return EXIT_FAILURE;
32     }
33     window.SetQuitOnClose(true);
34
35     ::MSG msg;
36     while (::SendMessage(&msg, WM_QUIT, 0, 0)) {
37         ::TranslateMessage(&msg);
38         ::DispatchMessage(&msg);
39     }
40
41     ::CoUninitialize();
42     return EXIT_SUCCESS;
43 }
44
```

En el caso de web:

flutter build web

Y el código generado se encuentra en build/web:

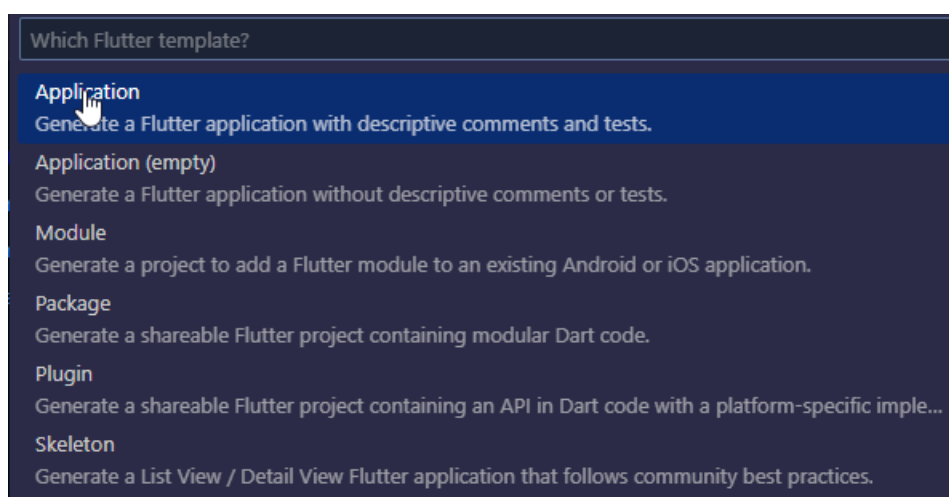
 assets	11/06/2023 20:16	Carpeta de archivos	
 canvaskit	11/06/2023 20:15	Carpeta de archivos	
 icons	11/06/2023 20:16	Carpeta de archivos	
 .last_build_id	11/06/2023 20:16	Archivo LAST_BUILD_ID	1 KB
 favicon	04/05/2023 9:30	Archivo PNG	1 KB
 flutter	11/06/2023 20:15	Archivo JavaScript	14 KB
 flutter_service_worker	11/06/2023 20:16	Archivo JavaScript	8 KB
 index	11/06/2023 20:16	Firefox HTML Document	2 KB
 main.dart	11/06/2023 20:15	Archivo JavaScript	1.648 KB
 manifest	11/06/2023 19:56	Archivo de origen JSON	1 KB
 version	11/06/2023 20:16	Archivo de origen JSON	1 KB

Identificar los tipos de archivos generados ¿A qué tecnologías perteneces?

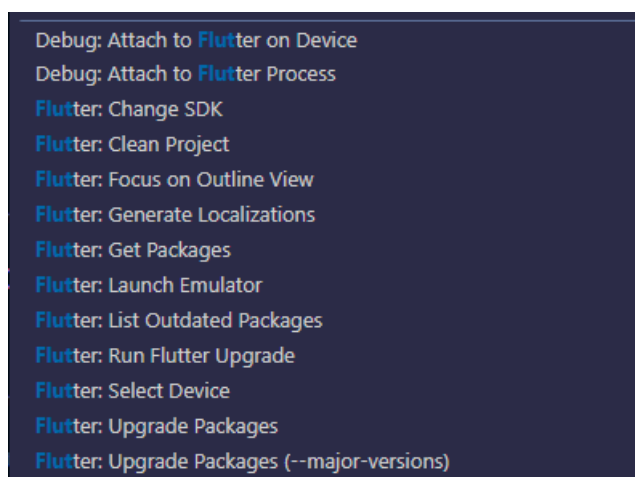
Estas mismas acciones se pueden realizar desde VSC, por ejemplo crear un nuevo proyecto:

- **Ctrl+Shift+P**
- **Flutter: New Project**

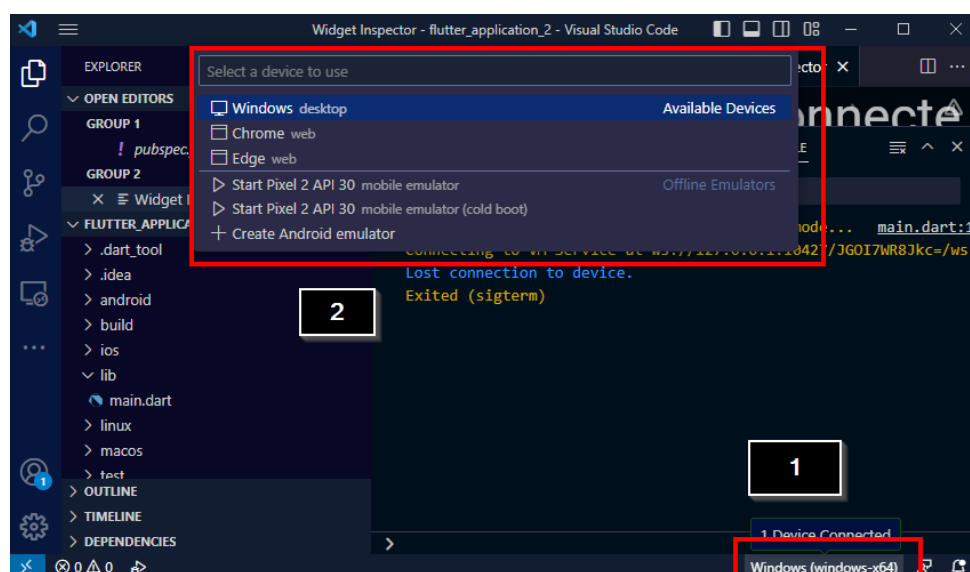
Permite ejecutar plantillas dependiendo del tipo de proyecto:



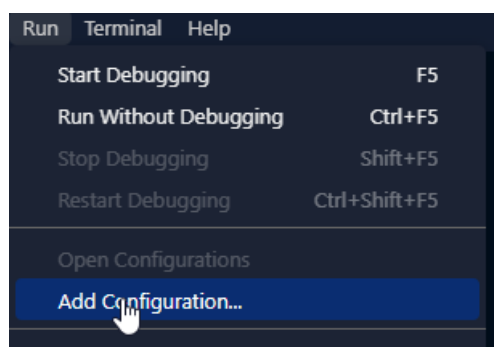
Seleccionado Application, que ejecuta el mismo que en la línea de comandos:

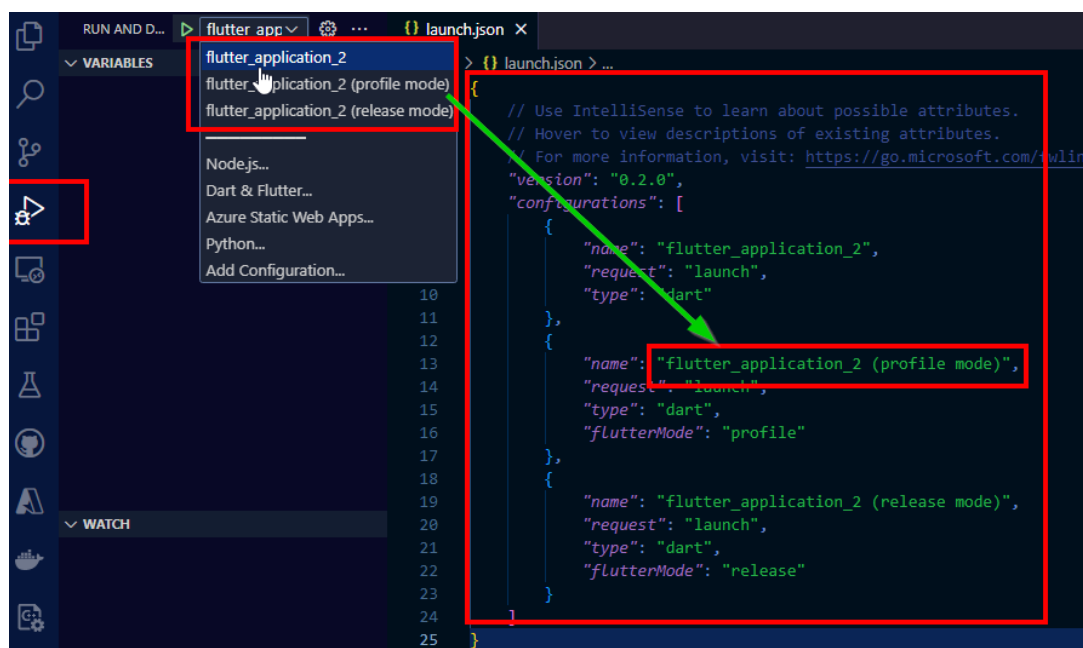


Para ejecutarlo se sigue los pasos usuales en cualquier IDE, en VSC F5 para ejecutar en modo depuración y Ctrl+F5, sin depuración, pudiendo seleccionar el dispositivo en el que se ejecutará en la parte inferior derecha del IDE.



Se puede configurar la ejecución y la construcción del proyecto de forma automática, para ello es necesario crear el fichero launch.json. Ir a **Run→Add Configuration**, creándose el fichero launch.json.



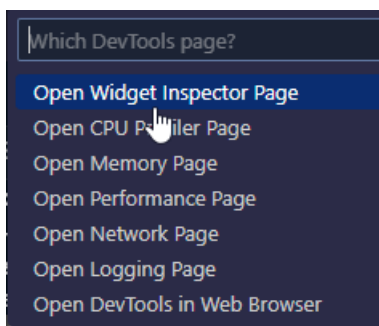


2.5. Parte 5. Herramientas de depuración.

La depuración en la fase de desarrollo es una herramienta clave independientemente del lenguaje. La extensión de Flutter para VSC posee las características típicas de los depuradores:

- BreakPoints.
- Variables.
- Ejecución paso a paso.
- ...

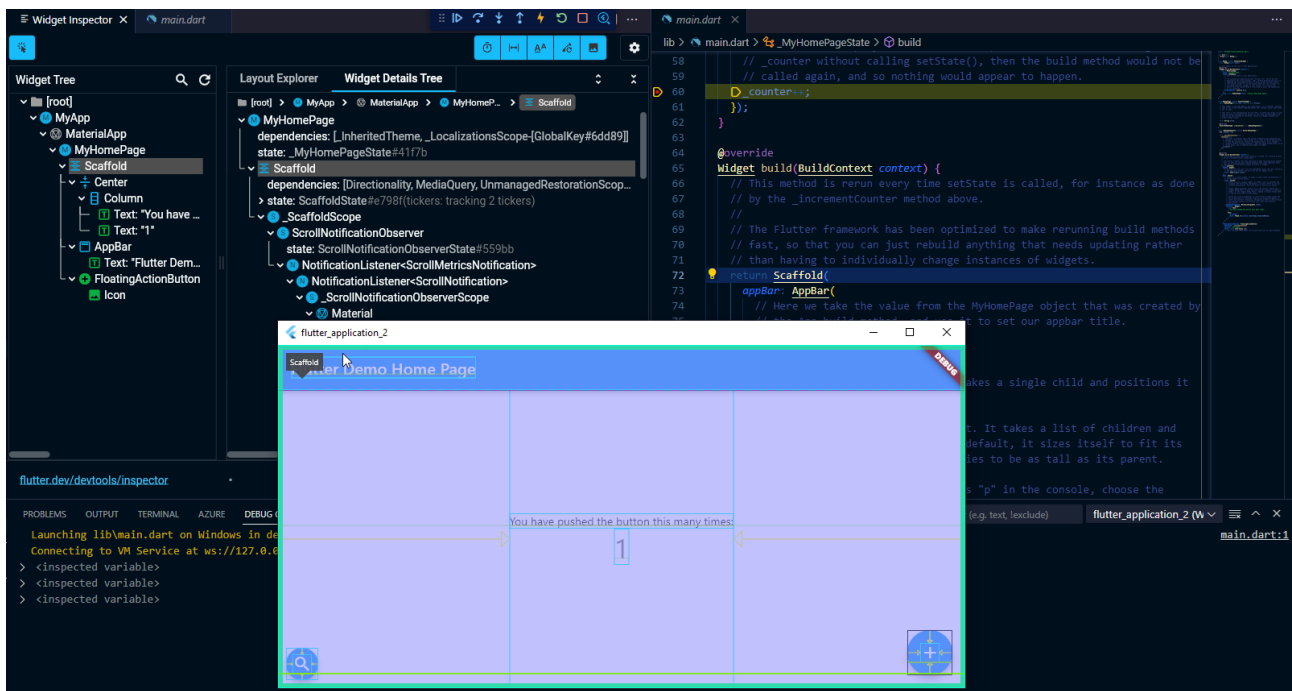
Pero además proporciona otras propias del desarrollo multiplataforma como es el inspector de componentes(similar a los que ofrecen los navegadores) que permite ver la estructura de los elementos, denominados “Widgets”, Performance que mide el rendimiento de la aplicación, o el uso de la .



Para activarlo, arrancar en modo depuración y buscar el comando Flutter: Open DevTools:



Y seleccionar el componente que se desee, aunque el uso mayoritario será el inspector.



2.6. Parte 6. Hola mundo.

La plantilla del proyecto crea un fichero en lib llamado main.dart que contiene:

Una función llamada main, que es el punto de entrada del programa.

```
void main() {  
  runApp(const MyApp());  
}
```

3 clases, que normalmente en las interfaces gráficas representan componentes.

- Clase MyApp que hereda de StatelessWidget, es la pantalla principal, internamente posee un elemento de tipo MyHomePage
- Clase MyHomePage que hereda de StatefulWidget

- `_MyHomePageState` que hereda de `State` y que posee un “Generic” con parámetro `<MyHomePage>`, posee un método `build` en el que se puede ver como define otros componentes como barra de aplicación (`AppBar`), columnas, textos...

Cambiar el texto : “You have pushed the button this many times” por “Hola Mundo, se ha pulsado el botón n veces”, con n el número de pulsaciones.

