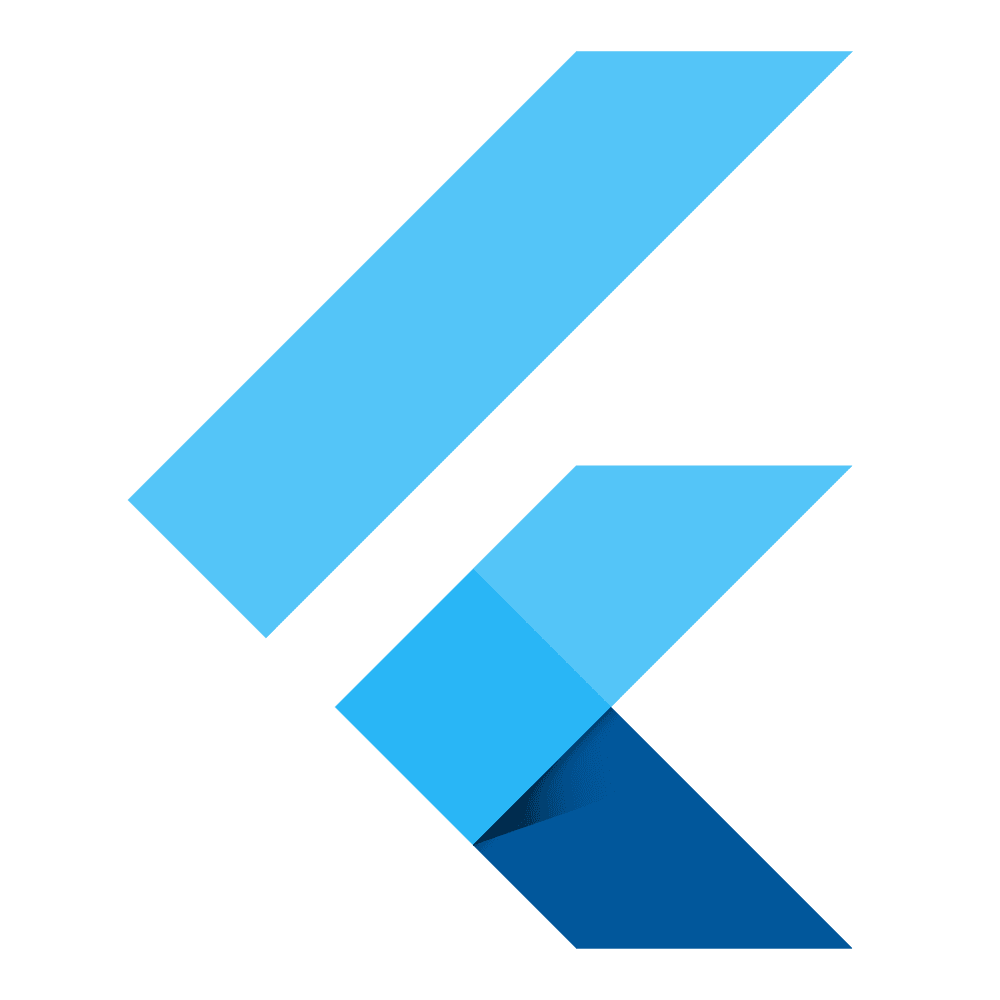
**Flutter**

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

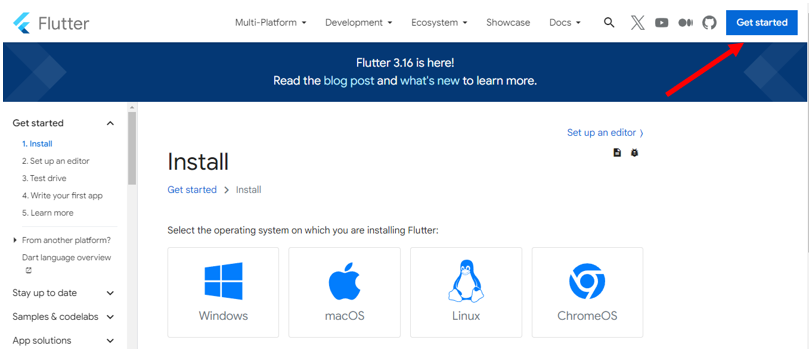
**Proceso de Instalación**

1. **Requisitos:**

Para instalar y ejecutar Flutter, su entorno de desarrollo debe cumplir con estos requisitos mínimos:

* *Sistemas operativos:* Windows 10 o posterior (64 bits), basado en x86-64.
* *Espacio en disco:* 2.5 GB (no incluye espacio en disco para IDE/herramientas).

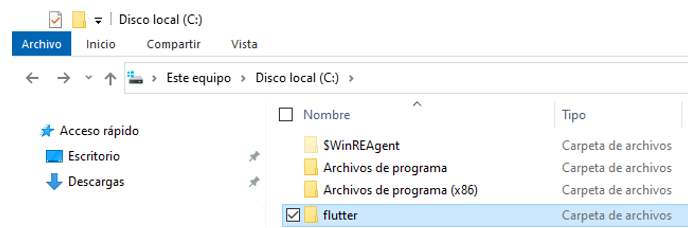
1. **Descargar el paquete de instalación (archivo comprimido):**

****

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

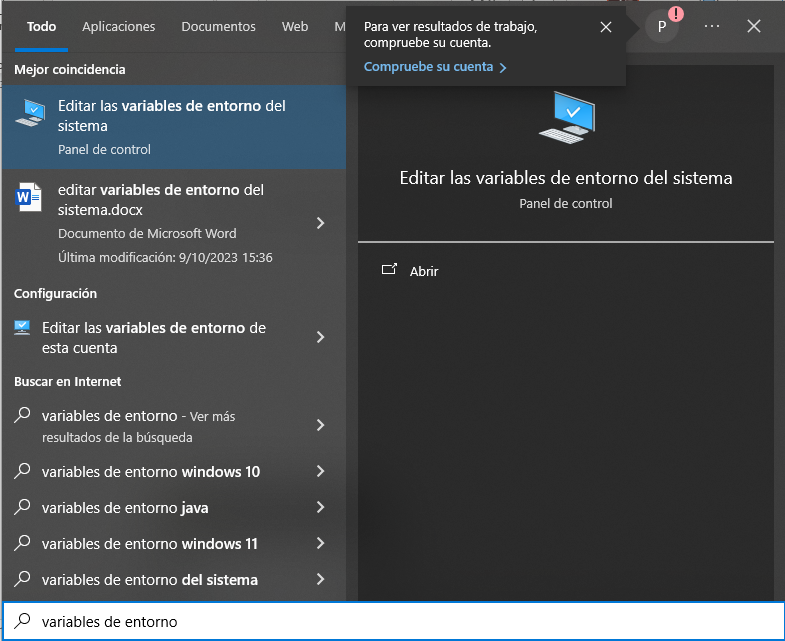
<https://docs.flutter.dev/get-started/install/windows>

1. **Descomprimir el archivo**

* Nombre original es similar a:  *flutter\_windows\_3.19.5-stable.zip* (Se sugiere revisar la última versión estable)
* La recomendación es cambiarle el nombre a la carpeta por *Flutter* únicamente
* Es importante usar nombres de carpetas sin espacio ni caracteres especiales
* Idealmente, ubicar la carpeta en una ruta corta, ejemplo colocarla directamente en C

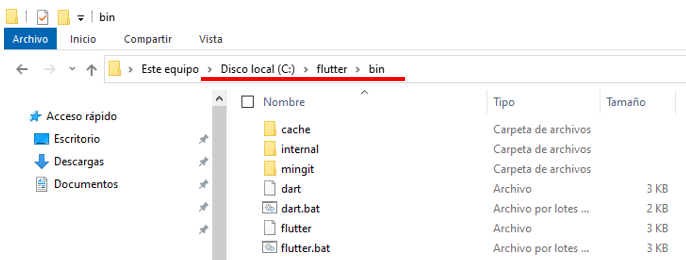
1. **Actualizar el path**

* Abrir **editar variables de entorno del sistema**



**Paso 1.** Copiar ruta de la carpeta *bin*:

C:\flutter\bin



**Paso 2.** Editar variables de entorno:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

1. Variables de usuario:

Una captura de pantalla de una computadora

Descripción generada automáticamente

* Path 🡪 Editar 🡪 Nuevo
* Pegar la ruta de la carpeta *bin*

Editar:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Nuevo:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

1. Variables del sistema:

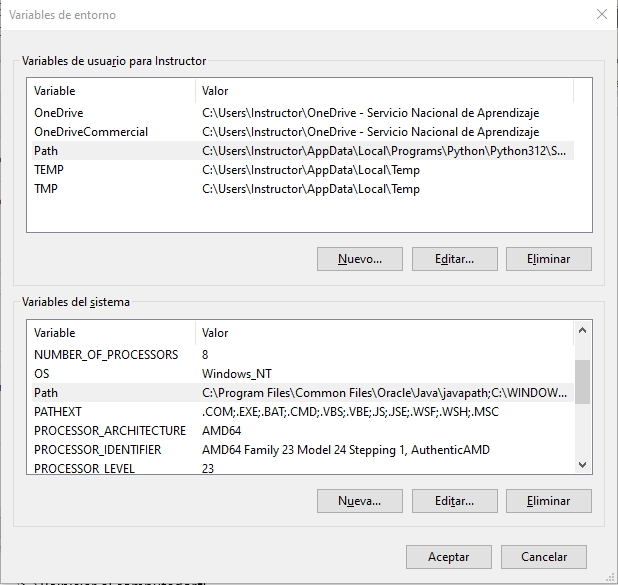
Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Se realiza de la misma manera que en el paso anterior:

* Path 🡪 Editar 🡪 Nuevo
* Pegar la ruta de la carpeta *bin*

Editar:



Nuevo:

Interfaz de usuario gráfica, Aplicación

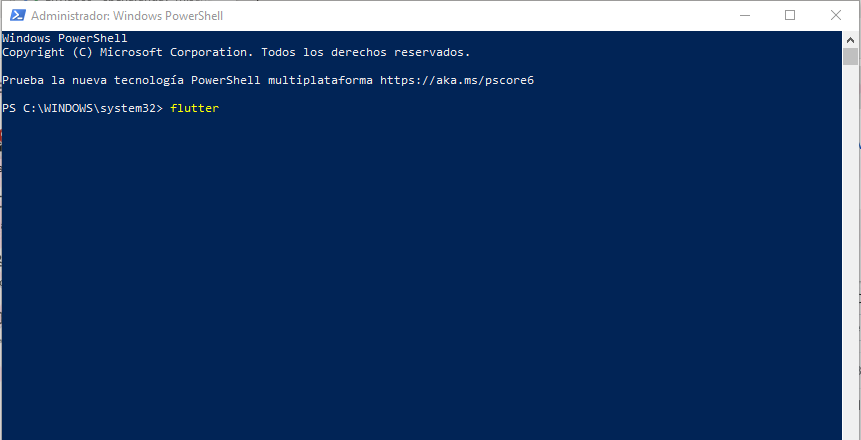
Descripción generada automáticamente

**PowerShell**

1. Abrir *PowerShell* como administrador:



1. Revisar la instalación realizada:



**Resultados esperados:**Texto

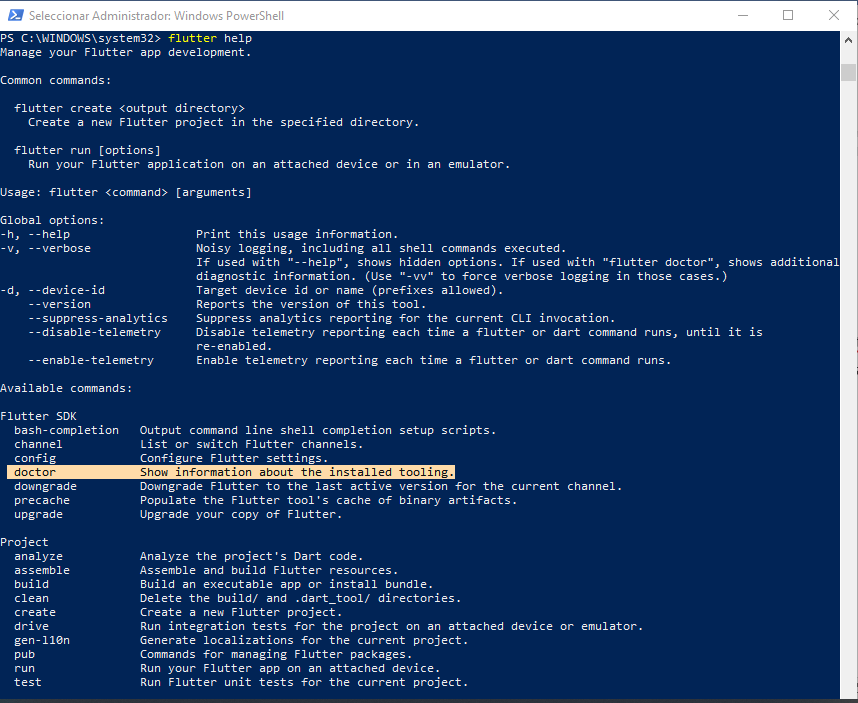
Descripción generada automáticamente

Texto

Descripción generada automáticamente con confianza media

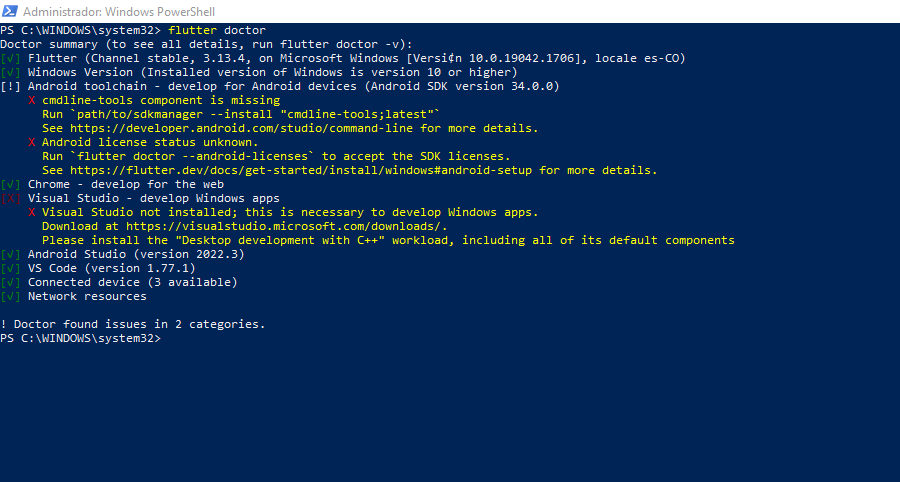
Ayuda:

Se puede ejecutar el comando "flutter help" para obtener más información sobre flutter. Así mismo, se ejecuta "**flutter help** -v" para obtener resultados de ayuda detallados, incluidas las opciones menos utilizadas.



1. Ejecutar *flutter doctor*:

* Mediante este comando se muestra información sobre las herramientas instaladas.
* Hace escaneo y permite identificar si hay alguna dependencia de plataforma que se necesite para completar la configuración para trabajar con Flutter.
* PS C:\WINDOWS\system32> flutter doctor
* Resultado:



Errores que se pueden presentar:

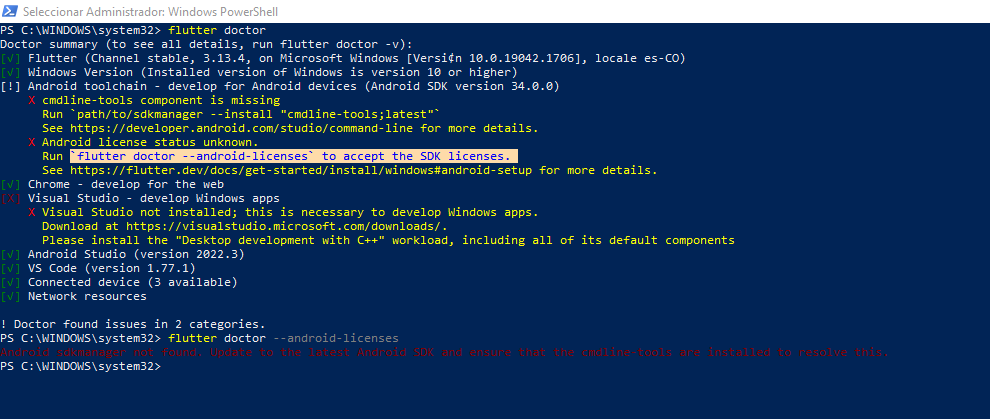
* No tiene el componente ***cmdline-tools component***
* No conoce la licencia de Android
* Visual Studio no está instalado

Soluciones de errores:

1. Ejecutar:

*Android license status unknown.*

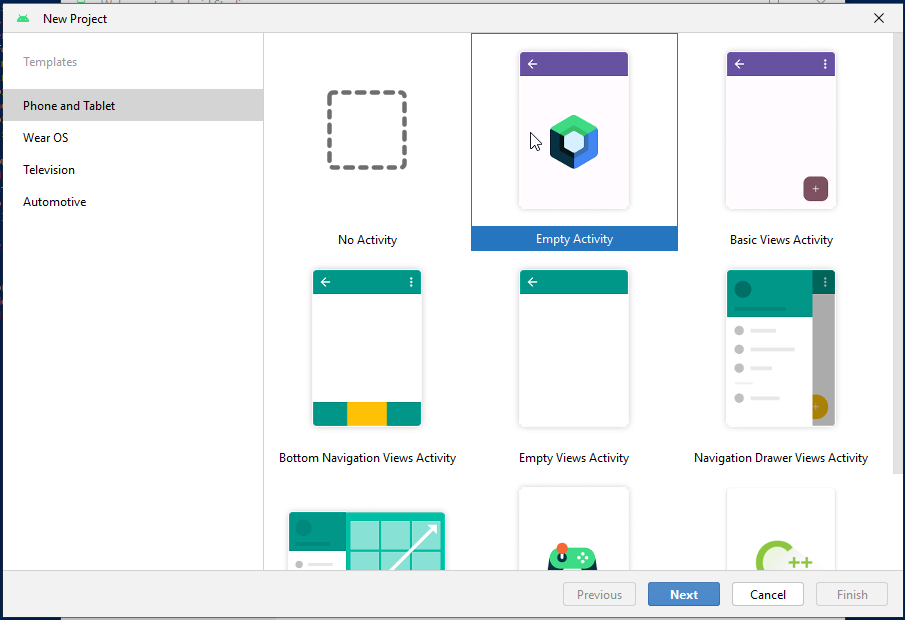
*Run `****flutter doctor --android-licenses****` to accept the SDK licenses.*

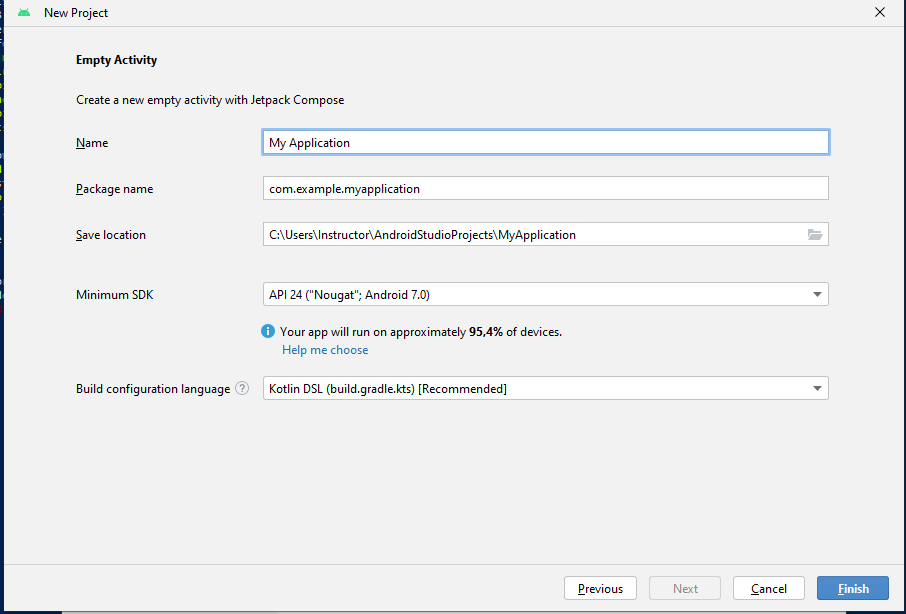


Sin embargo, indica error del SDK:

*“Android sdkmanager not found. Update to the latest Android SDK and ensure that the cmdline-tools are installed to resolve this.”*

Para solucionar creamos un proyecto en blanco en Android Studio para que se descargue e instale por defecto un SDK:

**

**

Instalación*:*

*Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

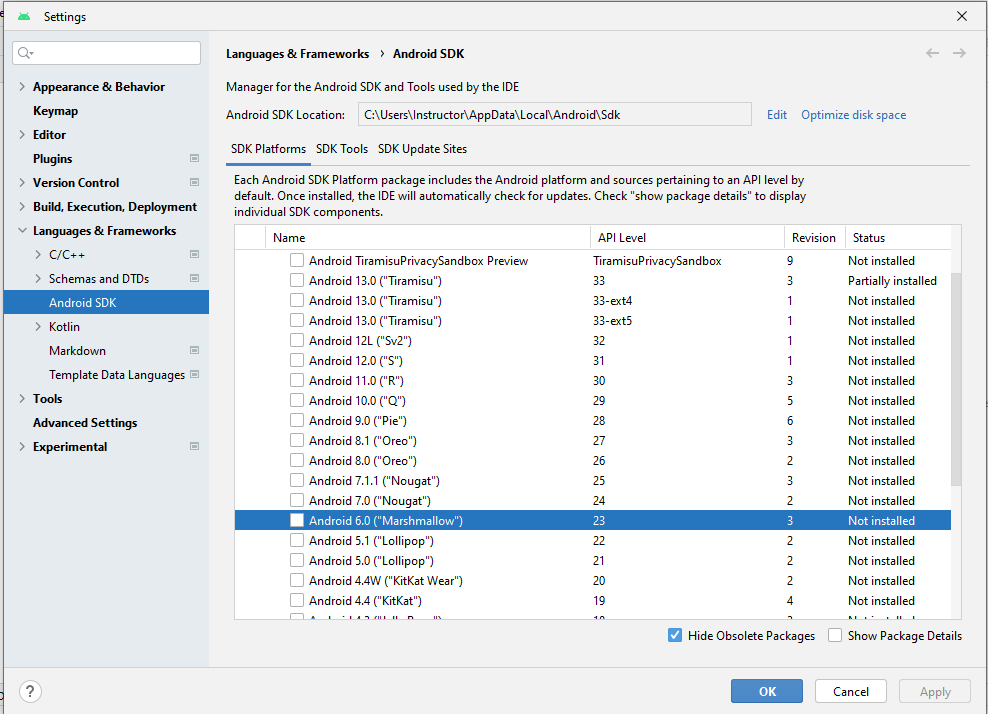
Descripción generada automáticamente*

Abrir el ***SDK Manager***:

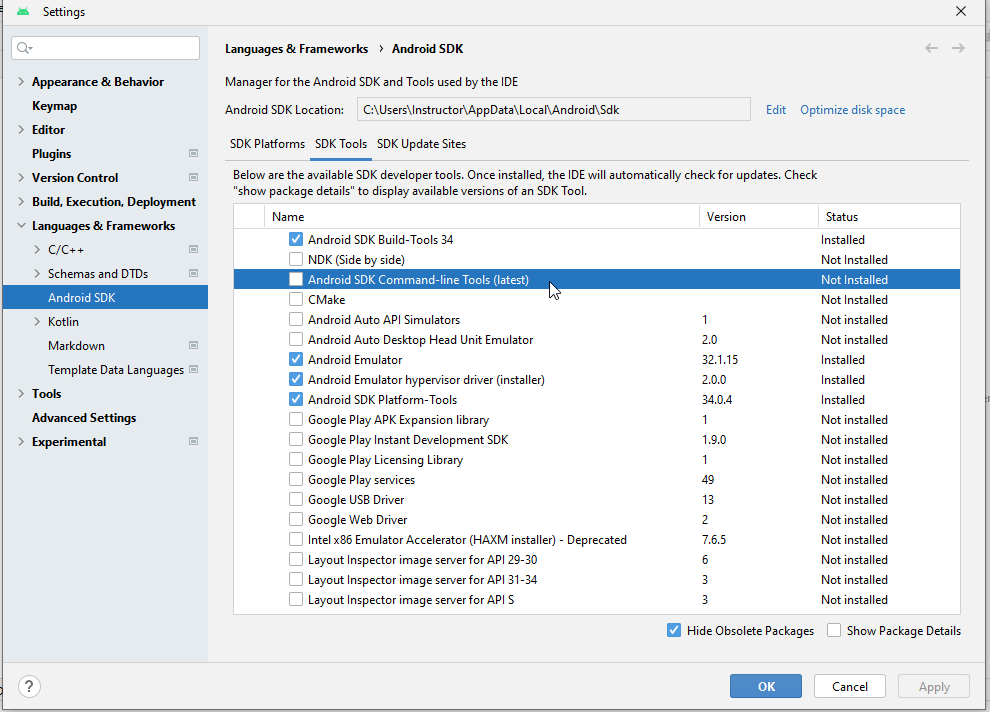
Interfaz de usuario gráfica, Texto, Aplicación, Word

Descripción generada automáticamente

* Instalar un SDK:



* Para corregir el error nos ubicamos en la pestaña *SDK Tools* y se realizan las siguientes instalaciones:

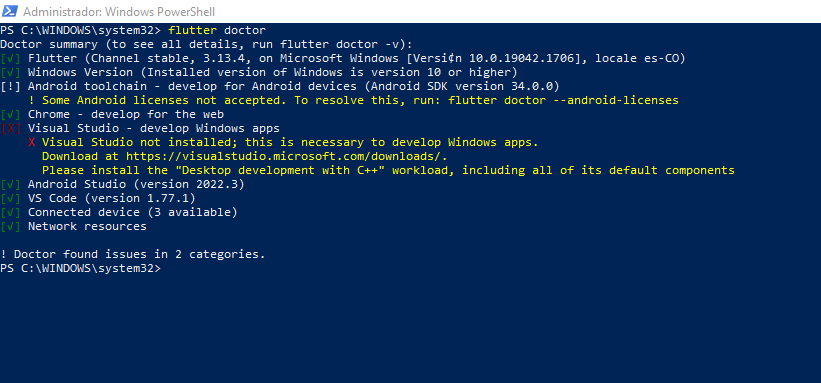


También instalar Google USB DRIVER:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

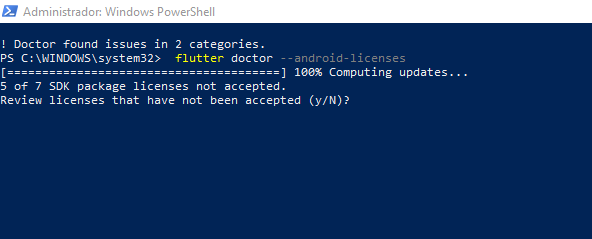
Ahora volvemos a verificar mediante el comando doctor:

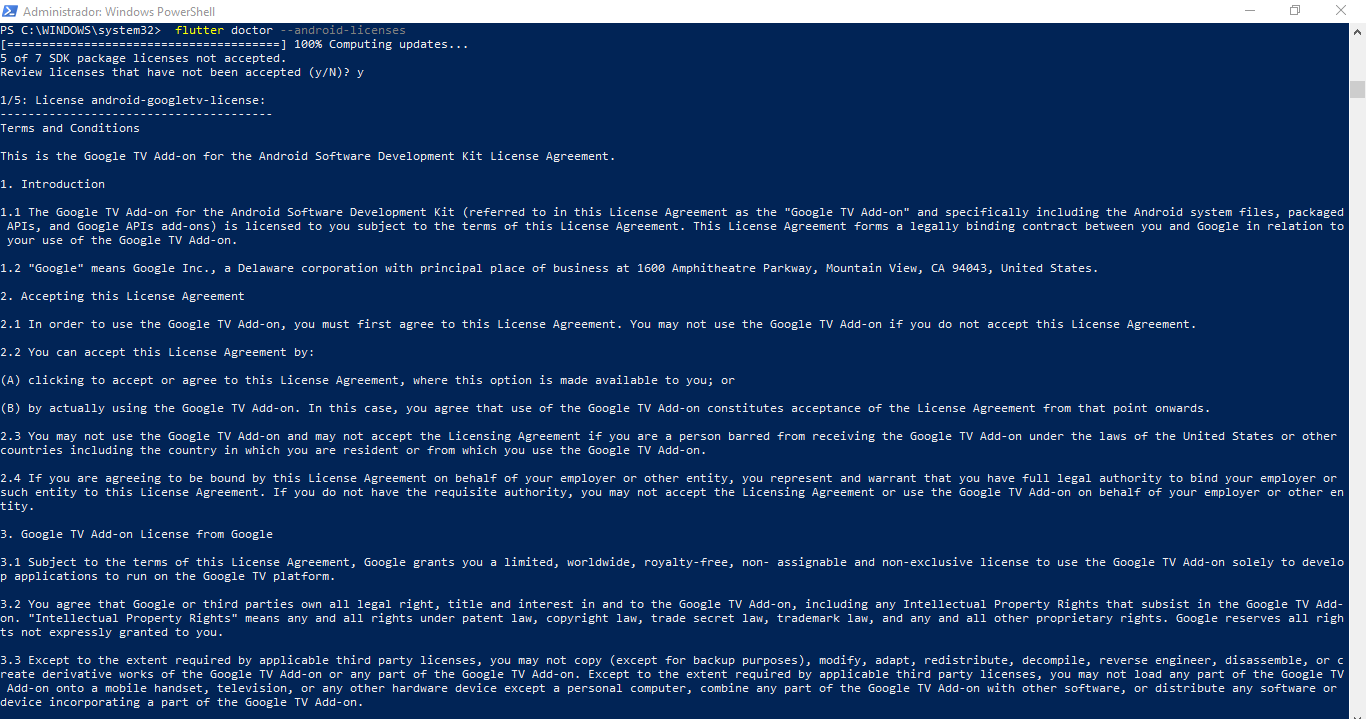


1. Notemos que se debe aceptar la licencia:

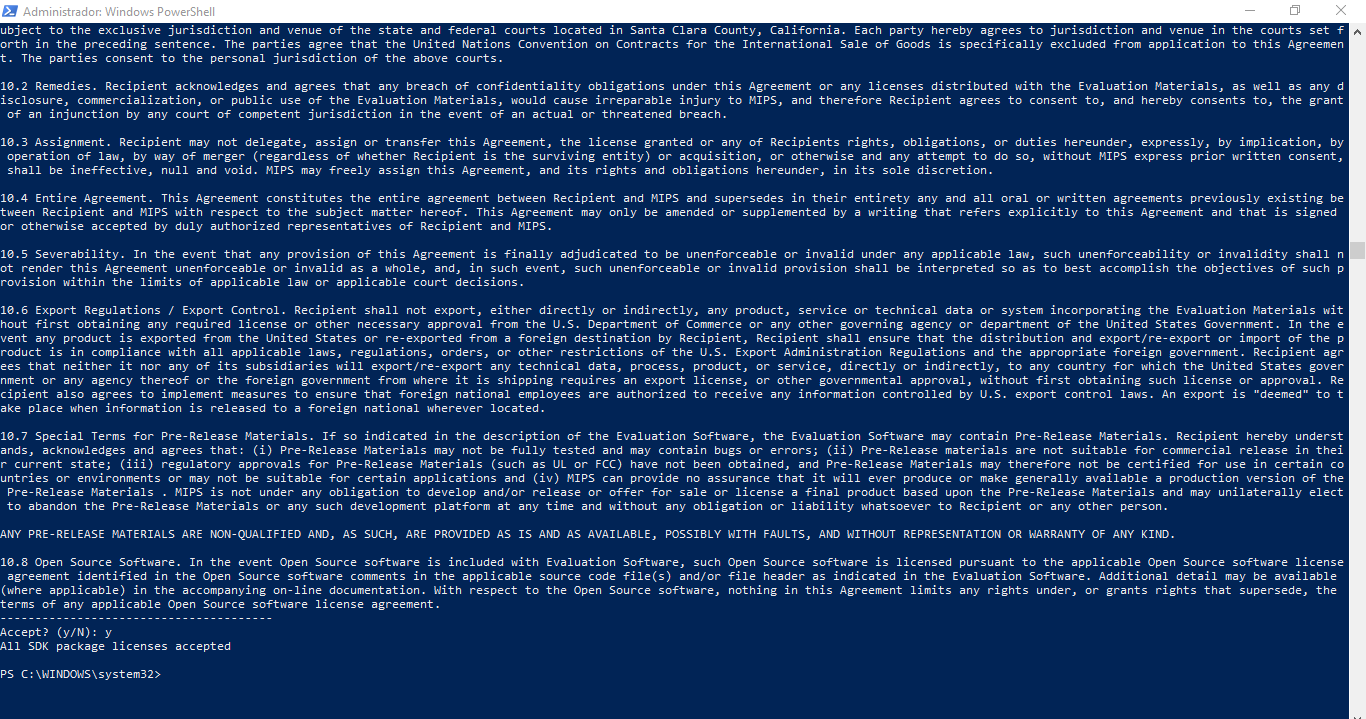
! Some Android licenses not accepted. To resolve this, run: flutter doctor --android-licenses

Así que aceptamos la licencia mediante el comando indicado***: PS C:\WINDOWS\system32> flutter doctor --android-licenses***

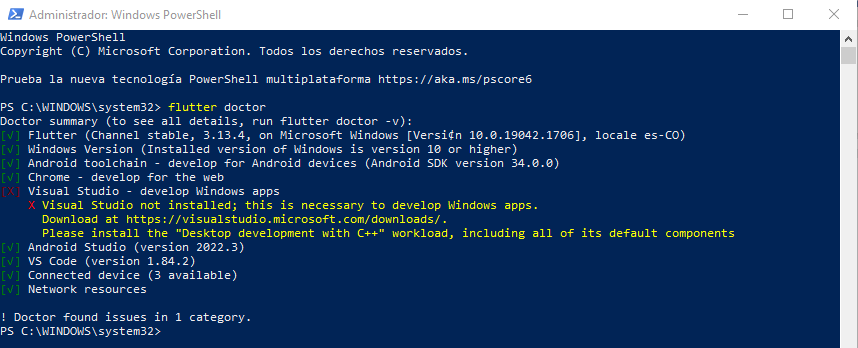




Finalmente:

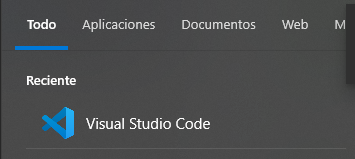


Ahora volvemos a verificar mediante el comando doctor:

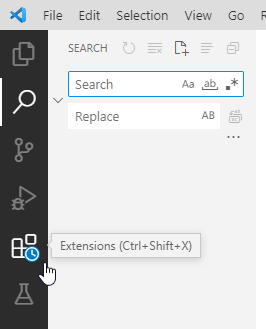


Una vez que haya terminado de configurar el entorno de trabajo, todo lo que tiene que hacer es iniciar un primer proyecto de Flutter. Para nuestro taller, utilizaremos el editor de código fuente ***Visual Studio Code***.

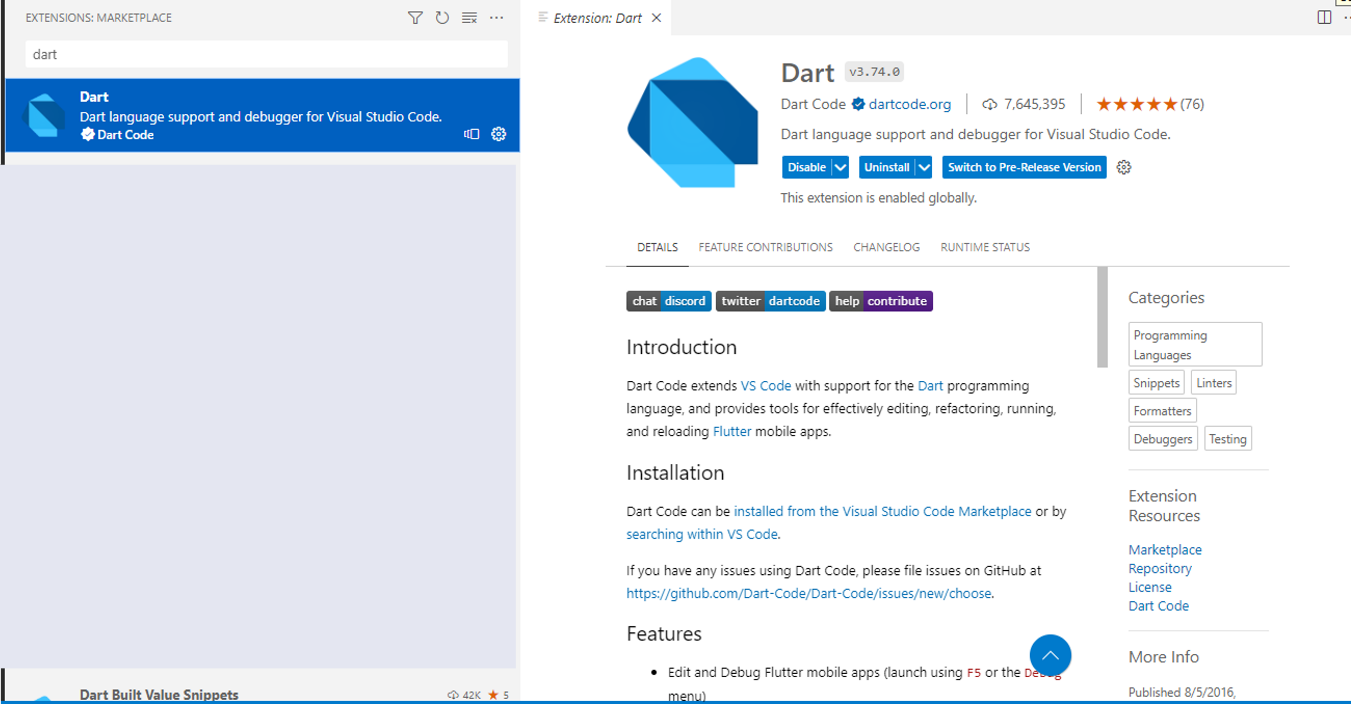
**VISUAL STUDIO CODE**



1. Instalar: <https://code.visualstudio.com/>
2. Abrimos la herramienta VISUAL CODE
3. Nos ubicamos en ***EXTENSIONES***



1. Instalar las siguientes extensiones:
   1. **Dart**

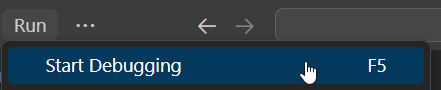


La extensión Dart nos ofrece soporte para el lenguaje de programación Dart y proporciona herramientas para editar,ejecutar y recargar de manera efectiva las aplicaciones móviles con el framework Flutter.

* 1. Flutter

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Esta extensión de Visual Studio Code agrega soporte para editar, ejecutar y recargar de manera efectiva las aplicaciones móviles de Flutter. Los proyectos deben ejecutarse usando **F5:**

* 1. **Flutter Color**

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Es un complemento fácil de usar que ayudará a trabajar con RGB Color en Visual Studio Code, para cualquier proyecto de forma automática.

Provee capacidad para permitir la categorización de colores, lo que garantiza una coherencia visual en toda la aplicación.

* 1. **Flutter Widget Snippets**

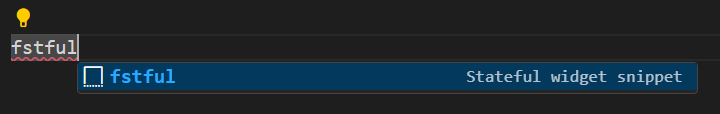
Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Snippets es un término utilizado en programación para referirse a fragmento de código​ fuente reutilizable.

Esta extensión nos provee de un conjunto de fragmentos útiles de Flutter y Dart para el desarrollo diario de Flutter.

Ejemplo:

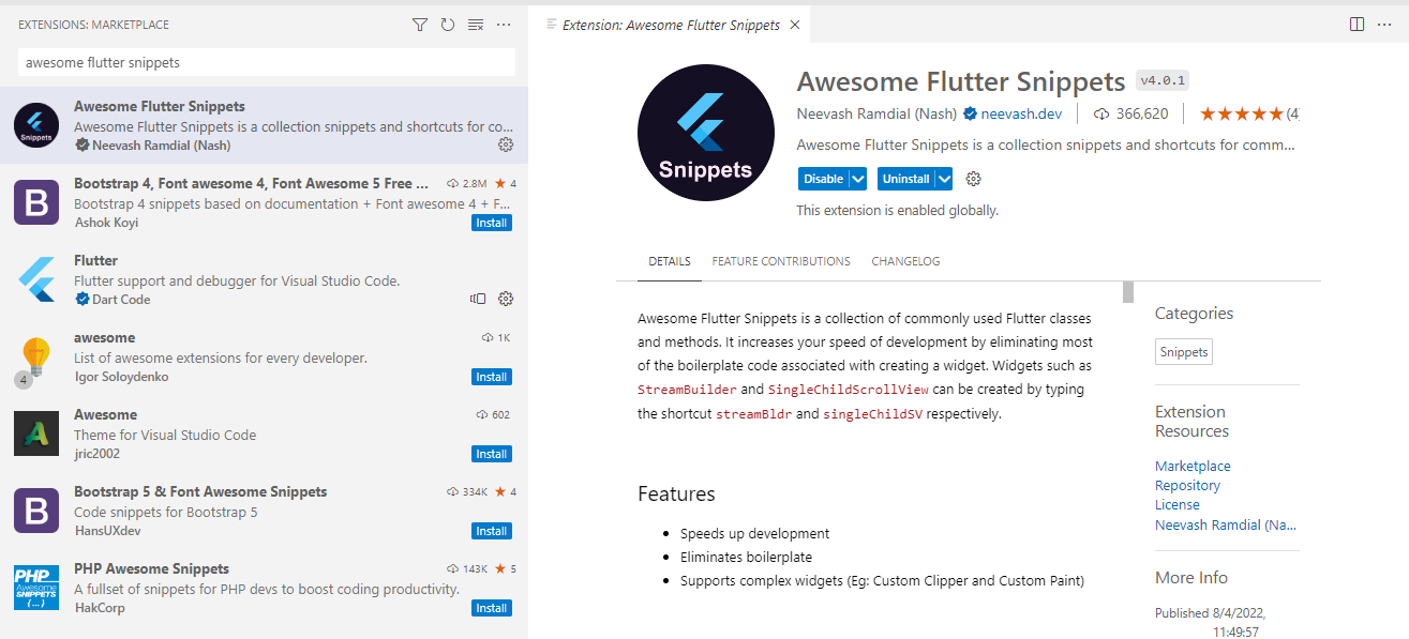


Snippets relacionados con Flutter

| **Snippet** | **Description** |
| --- | --- |
| fstful | StatefulWidget snippet. This is an alternative of stful |
| fstless | StatelessWidget snippet. This is an alternative of stless |
| fscaff | Scaffold widget snippet |
| fedgall | EdgeInsets widget snippet with named constructor all |
| fedgonly | EdgeInsets widget snippet with named constructor only |
| ftxt | Text widget snippet |
| finitlf | Flutter initState lifecycle method snippet |
| fic | Flutter Icon widget snippet |
| fcont | Flutter Container widget snippet |
| fcent | Flutter Center widget snippet |
| frow | Flutter Row widget snippet |
| fcol | Flutter Column widget snippet |
| fex | Expand widget snippet |
| fszbw | SizedBox widget snippet with just width argument |
| fszbh | SizedBox widget snippet with just height argument |
| fszb | SizedBox widget with width and height arguments |
| fedgsym | EdgeInsets widget with named constructor symmetric |
| fedgsymv | EdgeInsets widget with named constructor symmetric with vertical parameter |
| fedgsymh | EdgeInsets widget with named constructor symmetric with horizontal parameter |
| fimpmat | Add material's package import statement |
| fstream | Display a StreamBuilder widget |
| felbtn | Flutter ElevatedButton snippet |

Snippets relacionados con Dart:

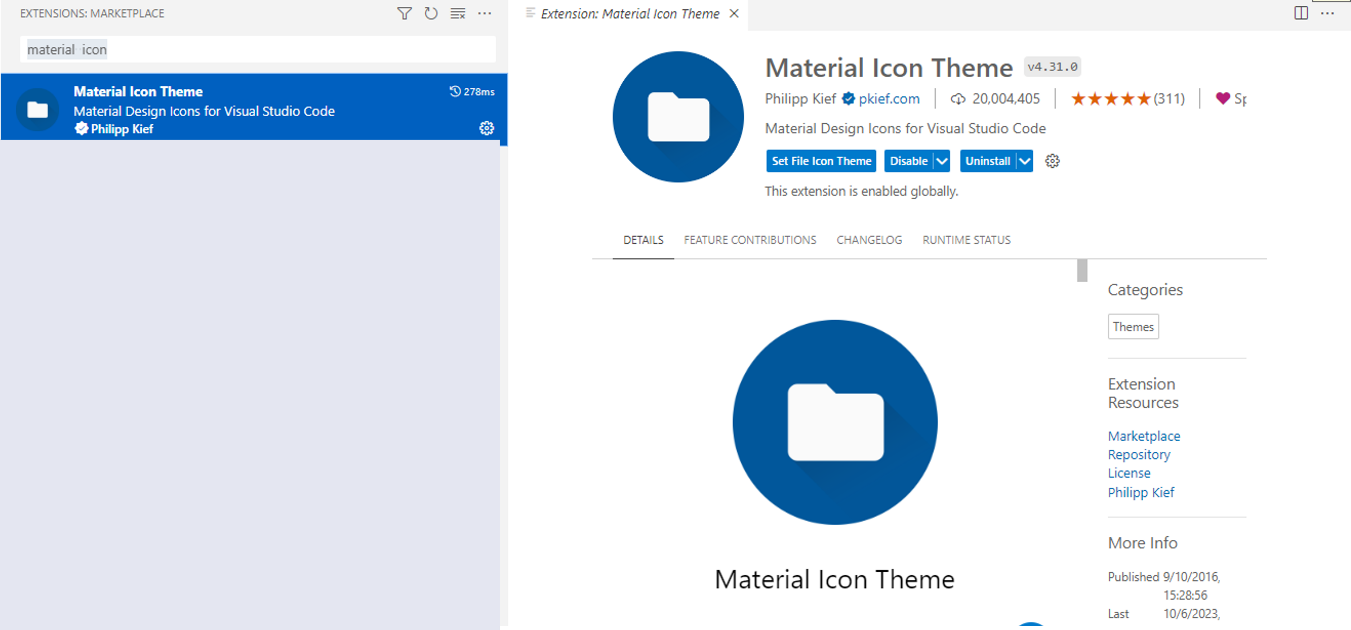
| **Snippet** | **Description** |
| --- | --- |
| dvar | Dart variable declaration using var |
| dfinal | Dart variable declaration using final |
| dconst | Dart variable declaration using const |
| dinvar | Dart Public Instance variable snippet |
| dprinvar | Dart Private instance variable snippet |
| dmt | Dart public method snippet |
| dprmt | Dart private method snippet |
| darr | Dart public arrow function snippet |
| dprarr | Dart private arrow function snippet |
| dopnctor | Dart optional named parameters constructor snippet |
| dlist | Dart List collection snippet |
| dmap | Dart Map collection snippet |
| dset | Dart Set collection snippet |
| dgetarr | Dart arrow function getter snippet |
| dimpas | Dart import as snippet |
| dimpshow | Dart import show snippet |
| dimplazy | Dart import deffered as snippet |
| dimphide | Dart import hide snippet |
| dexhide | Dart export hide snippet |
| dexshow | Dart export show snippet |
| dconvert | Dart convert lib import snippet |
| dimpmeta | Add meta package import statement |
| dan | Add a Dart anonymous function |
| dcla | Add Dart Class snippet |
| dclae | Add Dart Class snippet with extends keyword |

* 1. Awesome Flutter Snippets 

Awesome Flutter Snippets es una colección de clases y métodos de Flutter de uso común. Aumenta la velocidad de desarrollo al eliminar la mayor parte del código repetitivo asociado con la creación de un widget.

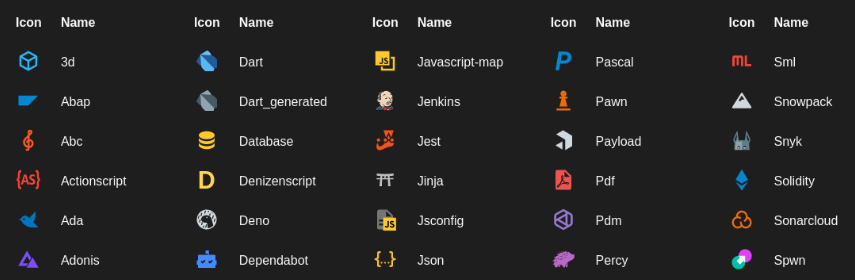
Por ejemplo, se pueden crear widgets como *StreamBuilder* y *SingleChildScrollView* escribiendo el shortcut *streamBldr* y *singleChildSV* respectivamente.

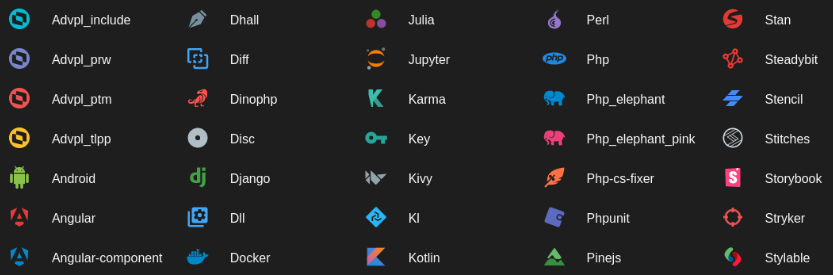
* 1. **Material Icon Theme**



Extensión para obtener los íconos de Material Design en Visual Studio Code.

Iconos:



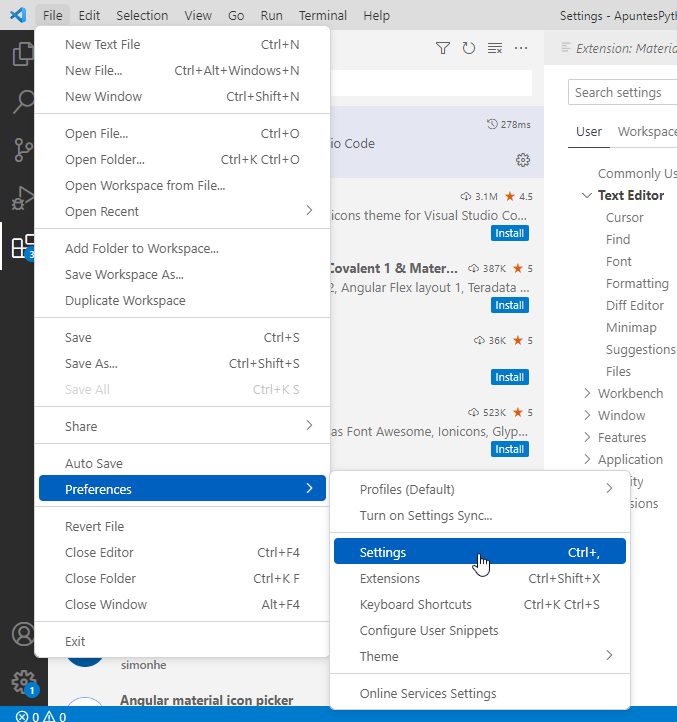


**Imagen que contiene dibujo, señal

Descripción generada automáticamente** *Es necesario habilitar las extensiones.*

1. Configurar ***Format on save*** de VISUAL STUDIO, si se desea:

El objetivo es aplicar un estilo consistente y ordenado a su código fuente lo cual permite mejorar la legibilidad del código fuente (Indentación de código).



**Opción 1.**

File 🡪 Preferences 🡪 Settings 🡪 Text editor 🡪 Format on save

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**Opción 2.** Mediante el engranaje:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente