



## ÍNDICE DE CONTENIDO

OBJETIVOS .....	1
¿QUÉ ES VISUAL STUDIO CODE? .....	1
INSTALACIÓN DE PYTHON .....	1
EJECUCIÓN DE INTÉRPRETE DE PYTHON Y NOTEBOOKS.....	2
INSTALACIÓN DE VISUAL STUDIO CODE.....	3
PROGRAMACIÓN EN PYTHON CON VISUAL STUDIO CODE .....	3
DEPURACIÓN EN PYTHON CON VISUAL STUDIO CODE .....	11

## OBJETIVOS

- Instalar todo lo necesario para programar en Python en la asignatura
- Aprender el uso básico de Visual Studio Code para el desarrollo de proyectos Python.

## ¿QUÉ ES VISUAL STUDIO CODE?

Según la propia web de la aplicación, Visual Studio Code (frecuentemente abreviado como VSC) es un editor de código fuente ligero pero potente que se ejecuta en el escritorio y está disponible para Windows, MacOS y Linux. Viene con soporte incorporado para JavaScript, TypeScript y Node.js y tiene un rico ecosistema de extensiones para otros lenguajes y entornos de ejecución (como C++, C#, Java, Python, PHP, Go, .NET). Por tanto, aunque nosotros lo usaremos para programar en Python, la misma aplicación permite trabajar en prácticamente todos los lenguajes utilizados actualmente.

Visual Studio Code dispone de muchas funcionalidades avanzadas muy útiles que facilitan la labor del programador. Además de dichas funcionalidades, existen multitud de extensiones de terceros (es decir, desarrolladas por programadores externos al propio equipo de desarrollo de la aplicación) que permiten ampliar aún más la funcionalidad.

En este documento, veremos cómo instalar y configurar la aplicación para poder trabajar con el lenguaje Python, cómo crear, ejecutar y depurar un proyecto, y cómo trabajar con repositorios de código Git y conectarlos con GitHub.

## INSTALACIÓN DE PYTHON

Antes de instalar Visual Studio Code, necesitamos instalar un entorno de ejecución para Python. Nosotros vamos a utilizar la distribución estándar de Python. Si ya dispones de otro entorno de ejecución de Python instalado en tu ordenador, puedes utilizarlo en lugar del que te proponemos, siempre que incluya una versión de Python igual o superior a la 3.8.

Para instalar Python, accede a esta url: <https://www.python.org/downloads/>. Pulsa el botón “Download Python 3.xx.yy” (los números que aparecerán en xx.yy corresponderán a la última versión disponible). Ejecuta el instalador, marca la opción “Add python.exe to PATH”, y pulsa en “Install Now”.

### Antes de continuar, reinicia el equipo.

Debes instalar un par de dependencias que utilizaremos en clase: matplotlib (librería para la generación de gráficas y diagramas) y jupyter (entorno de ejecución de Python desde navegadores web). Para ello, abre un terminal (“Símbolo del sistema” si estás en Windows, puedes ejecutarlo pulsando la tecla Windows y escribiendo “cmd”), y ejecuta estos dos comandos (pulsa la tecla *Enter* detrás de cada comando):

```
pip install matplotlib
pip install jupyter
```

En ambos casos, pulsa la letra “y” para confirmar la instalación, cuando se te pregunte.



## EJECUCIÓN DE INTÉRPRETE DE PYTHON Y NOTEBOOKS

Si la instalación anterior se llevó a cabo correctamente, deberías poder ejecutar el intérprete de Python, que te permite escribir código en Python y ejecutarlo instantáneamente. También podrás ejecutar Jupyter, que permite utilizar los notebooks que utilizamos en clase (y que podrás descargar desde la Enseñanza Virtual).

Abre el terminal del sistema (si estás en Windows, la aplicación se llama “Símbolo del sistema”, puedes pulsar la tecla de Windows y escribir el nombre anterior o “cmd”). Se abrirá una ventana en modo texto que permite interactuar con el sistema operativo mediante comandos, así como ejecutar aplicaciones. Si escribes “python” y pulsas *enter*, se ejecutará el intérprete de Python. Prueba a escribir alguna instrucción en Python, por ejemplo:

```
print("Hola, mundo!")
```

Para salir del intérprete, escribe:

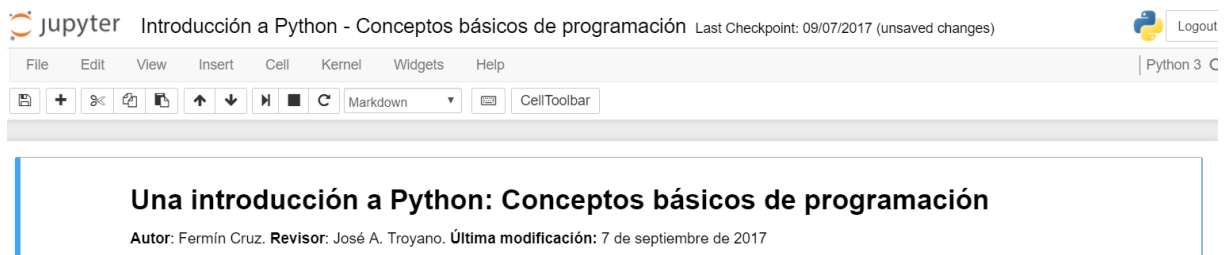
```
exit()
```

En lugar de “python”, podemos ejecutar el comando “ipython”, en cuyo caso el intérprete dispone de una mayor funcionalidad.

Sitúate ahora dentro de la carpeta en la que tengas almacenado los notebooks de la asignatura. Para ello, escribe “*cd ruta\_de\_la\_carpeta*” para acceder a una carpeta (puedes acceder a la carpeta en cuestión usando el Explorador de Archivos, y copiar la ruta de la carpeta que aparece al pulsar en la caja de texto que se muestra en la parte superior del explorador). Escribe “*jupyter notebook*” y pulsa *enter*. Se ejecutará la aplicación Jupyter, que permite trabajar con notebooks. Tras unos segundos, se abrirá un navegador web con la interfaz Jupyter.



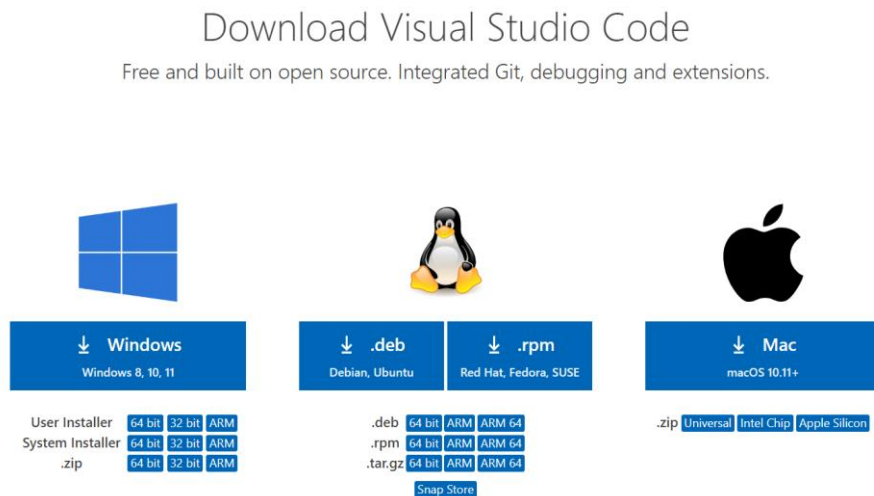
Puedes navegar por la estructura de carpetas y hacer clic sobre los notebooks para ejecutarlos. Por ejemplo, este es el aspecto que muestra el notebook “*Introducción a Python*”:



También podremos visualizar y ejecutar los *notebooks* desde Visual Studio Code, una vez que lo hayamos instalado.

## INSTALACIÓN DE VISUAL STUDIO CODE

Puedes descargar la aplicación desde <https://code.visualstudio.com/download>. Al acceder a dicha página, verás algo parecido a esto:

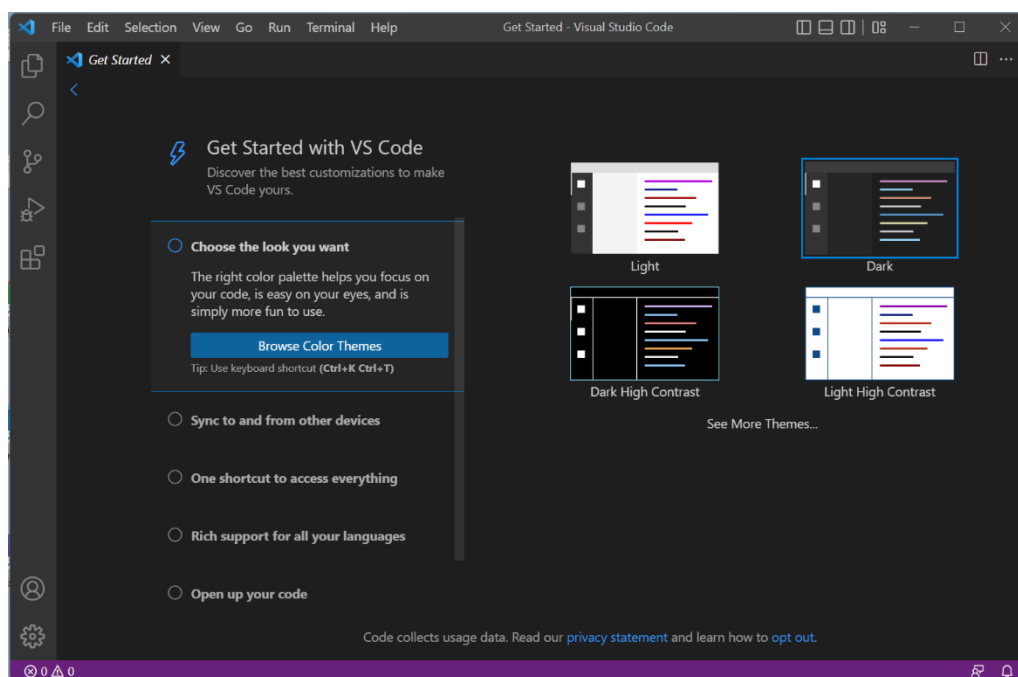


Escoge la opción correspondiente a tu sistema operativo. Si vas a trabajar en Windows, lo más recomendable es escoger la opción “User Installer”. Si tu ordenador es relativamente reciente, puedes escoger la versión “64 bit”. En caso de que al descargar esta opción y ejecutar el instalador tu sistema operativo te dijera que no puede ejecutarlo, deberías descargarte la versión “32 bit”.

Una vez descargado, ejecuta el instalador. Debes ir aceptando todas las opciones marcadas por defecto por el instalador: simplemente, ve pulsando siguiente hasta llegar al final del asistente.

## PROGRAMACIÓN EN PYTHON CON VISUAL STUDIO CODE

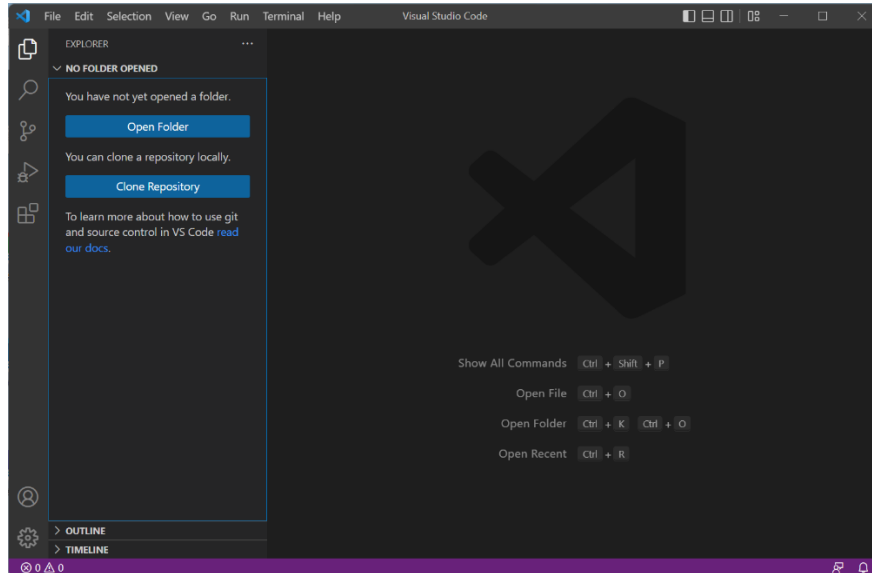
Al ejecutarse Visual Studio Code por primera vez, veremos la siguiente pantalla inicial:



Podemos aprovechar esta pantalla inicial para seleccionar el esquema de colores que prefiramos para la interfaz de la aplicación. Por defecto, el esquema seleccionado es “Dark”, que utiliza una combinación de colores con los fondos en

colores oscuros y los textos en colores claros. Una vez escogido el que prefiramos, podemos cerrar esta ventana de bienvenida, pulsando en el aspa que aparece en la pestaña a la derecha del texto “Get Started”<sup>1</sup>.

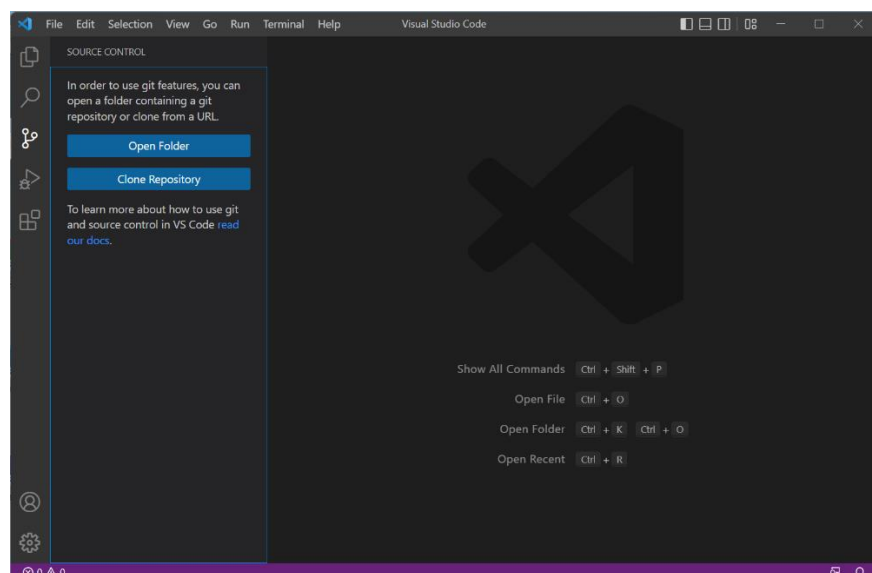
La barra de iconos que hay a la izquierda nos proporciona acceso directo a las funcionalidades más habituales de la aplicación. El primer botón sirve para abrir el explorador, donde podemos acceder a todos los archivos que conforman el proyecto en el que estemos trabajando. Como aún no hemos abierto ninguna carpeta, esta vista aparece vacía:



Volveremos a esta vista más adelante.

El siguiente botón (con el icono de una lupa), sirve para buscar en todos los archivos de la carpeta de trabajo cualquier cadena de texto y, en su caso, para sustituir todas las apariciones del texto buscado por un nuevo texto.

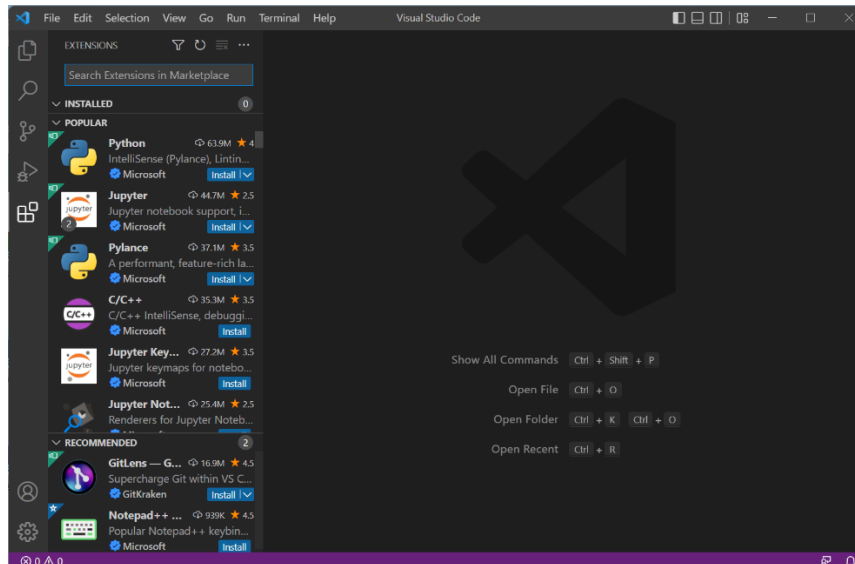
El tercer botón nos permitirá trabajar con el sistema de control de versiones Git, así como con los repositorios remotos de GitHub. Haremos uso de esta tecnología en el transcurso de la asignatura, especialmente para los proyectos entregables. Puedes ver un video sobre el uso de esta funcionalidad en Visual Studio Code aquí: <https://www.kaltura.com/tiny/yv925>.



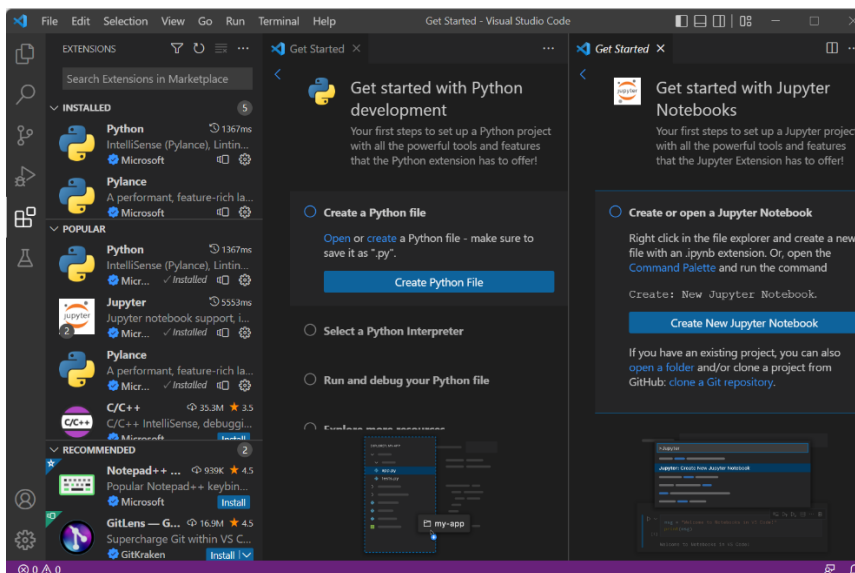
<sup>1</sup> Es posible que aparezca una ventana emergente abajo a la derecha, preguntando si se quiere instalar una extensión para el idioma español. Recomendamos no instalar esta extensión y mantener la interfaz en inglés, puesto que toda la documentación que nos servirá en el futuro para aprender nuevas funcionalidades de la aplicación estará en dicho idioma. Puedes indicar que no quieres volver a recibir la misma sugerencia pulsando en el botón de rueda dentada.

El cuarto botón nos da acceso a la vista de depuración, lo que nos permitirá ejecutar nuestros programas paso a paso y examinar el contenido de las distintas variables del entorno. Veremos un ejemplo de depuración más adelante en este mismo documento.

Por último, el quinto botón nos permite comprobar las extensiones que tenemos instaladas, así como buscar e instalar nuevas extensiones. Las extensiones permiten ampliar la funcionalidad proporcionada por Visual Studio Code. Vamos a acceder a esta vista para instalar la extensión que nos permitirá programar en lenguaje Python:

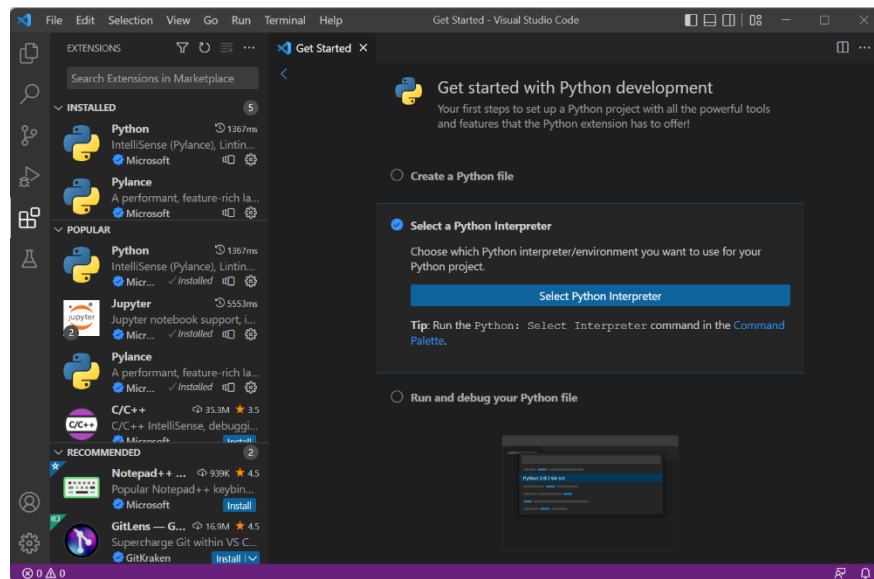


Pulsa el botón “Install” de la extensión “Python”. Esto desencadenará la instalación de dicha extensión, además de la de otras dos relacionadas que se instalarán automáticamente. Una vez acabada la instalación, debes tener una vista parecida a esta:



Son las páginas de bienvenida de dos de las extensiones que se han instalado. Puedes cerrar la de la derecha, y en la otra, pulsar “Select a Python Interpreter”<sup>2</sup>.

<sup>2</sup> Como puedes ver, se ha instalado una extensión para integrar Jupyter en VSC. Esto significa que podrás visualizar y ejecutar los notebooks también desde VSC, en lugar de en un navegador web.



Pulsa en el botón “Select Python Interpreter”. La aplicación te mostrará la barra o paleta de comandos (una barra de texto que aparece en la parte superior de la pantalla) y en ella se habrá desplegado una lista con todos los intérpretes de Python encontrados en tu ordenador. Si te salieran distintas opciones a escoger, selecciona la versión de Python que instalaste anteriormente; en cualquier momento puedes cambiar el que se va a utilizar para ejecutar nuestro proyecto, de dos maneras:

- Pulsando en el número de versión del intérprete que actualmente se está utilizando, el cual aparece en la barra inferior de la aplicación<sup>3</sup>. Al pulsar en ella, se nos vuelve a mostrar la lista de intérpretes disponibles en el sistema para seleccionar el que queremos utilizar.
- Accediendo a la paleta de comandos<sup>4</sup> (tecla F1). La paleta de comandos permite acceder a todas las funcionalidades de VSC, escribiendo en ella las palabras adecuadas (funciona de una manera similar al botón de inicio de Windows). Si escribimos “Python select interpreter” llegaremos a la misma lista desplegable de intérpretes anterior.

Ahora vamos a crear un nuevo proyecto para ver cómo editamos nuestro código Python y lo ejecutamos. En Visual Studio Code, un proyecto no es más que una carpeta cualquiera de nuestro disco duro que contenga tanto los archivos de código fuente (ficheros de texto con el código en el lenguaje de programación que estemos utilizando) como el resto de archivos utilizados por el programa que estemos desarrollando (ficheros de datos, imágenes, ficheros de configuración, etc.).

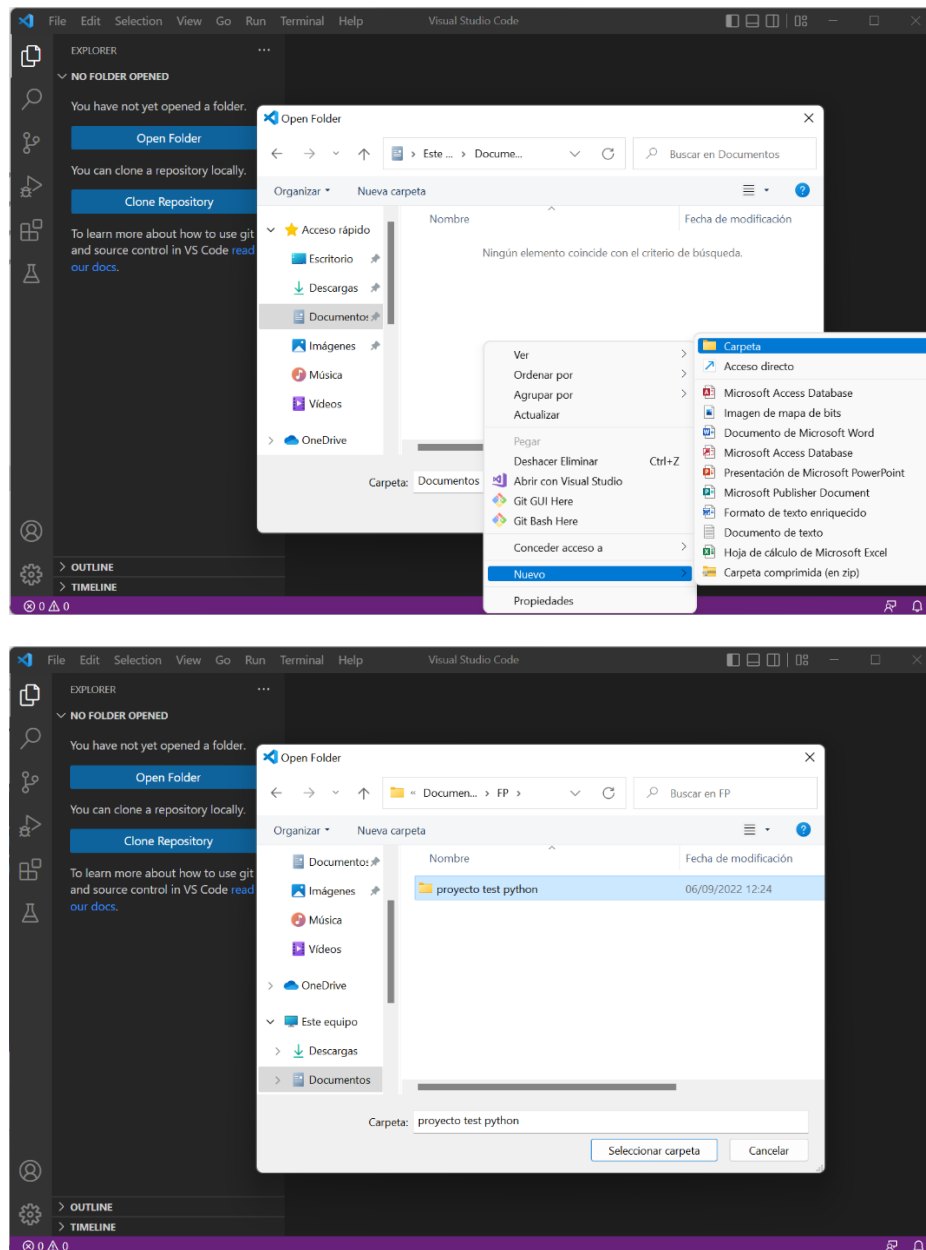
Vamos a abrir una carpeta nueva. Para ello, accede al explorador (primer botón de la barra de botones de la izquierda), y pulsa el botón “Open Folder”.

NOTA: También puedes abrir una carpeta desde el menú superior “File” → “Open Folder”. Incluso puedes arrastrar una carpeta ya existente de tu disco duro (por ejemplo, desde el Explorador de Archivos de Windows) encima de la ventana de la aplicación VSC, lo que desencadenará la misma acción anterior (abrir dicha carpeta).

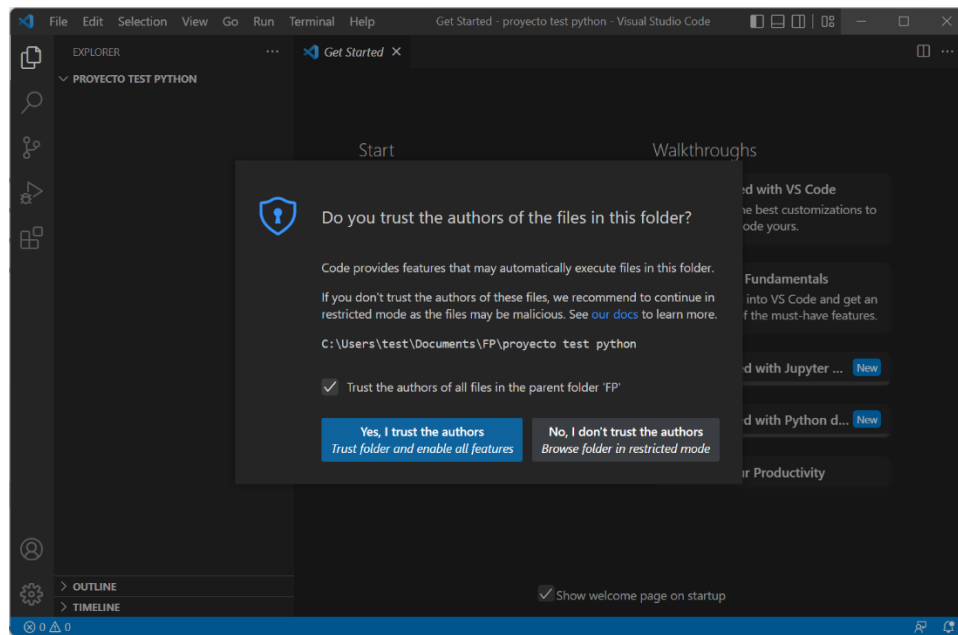
Es recomendable que crees alguna carpeta en tu disco duro donde mantengas todo el material relacionado con la asignatura. Por ejemplo, vamos a crearnos una carpeta “FP” dentro de “Documentos”, y dentro de ella una carpeta que se llame “proyecto test python”. Todo esto puedes hacerlo desde el Explorador de Archivos de Windows, o directamente desde la la ventana contextual que se abre al pulsar en “Open Folder”:

<sup>3</sup> Este botón sólo aparecerá cuando estemos editando un fichero de código fuente en Python.

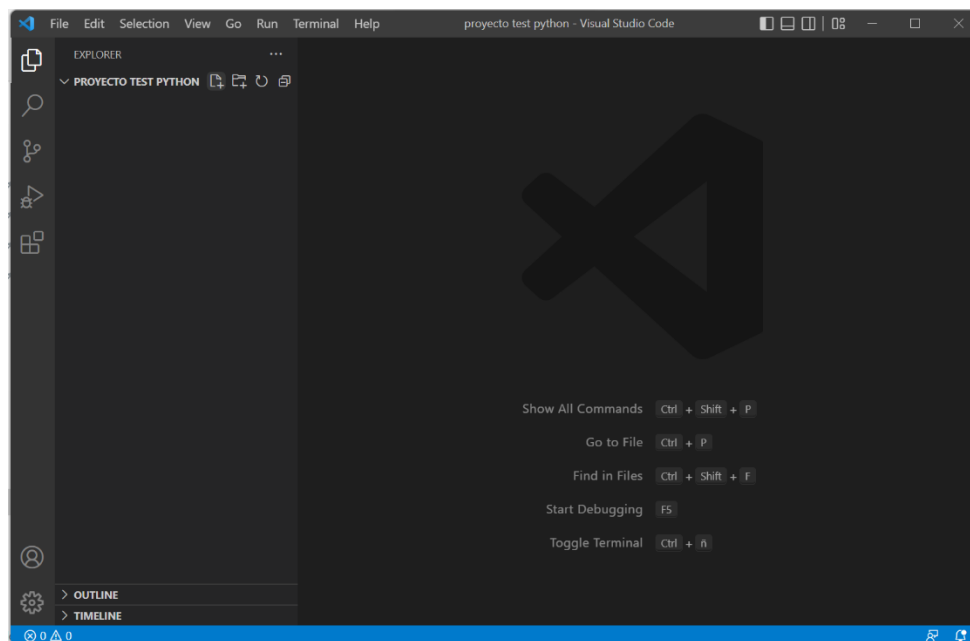
<sup>4</sup> Aunque las funcionalidades más comunes tienen accesos directos bien a través de botones de la interfaz, bien a través de los menús superiores, en ocasiones resulta más cómodo usar la paleta de comandos. Prueba por ejemplo a escribir en la paleta “color theme”, lo que nos permitirá cambiar el esquema de colores de la interfaz.



Siempre que abrimos una carpeta por primera vez, VSC nos preguntará si confiamos en los autores de los archivos contenidos en dicha carpeta. Debemos responder que sí para que nos permita ejecutar nuestros programas sin ninguna limitación. Marcando la opción “Trust the authors of all files in the parent folder ‘FP’”, conseguiremos que no nos vuelva a preguntar nada cuando abramos otras carpetas que ubiquemos allí.

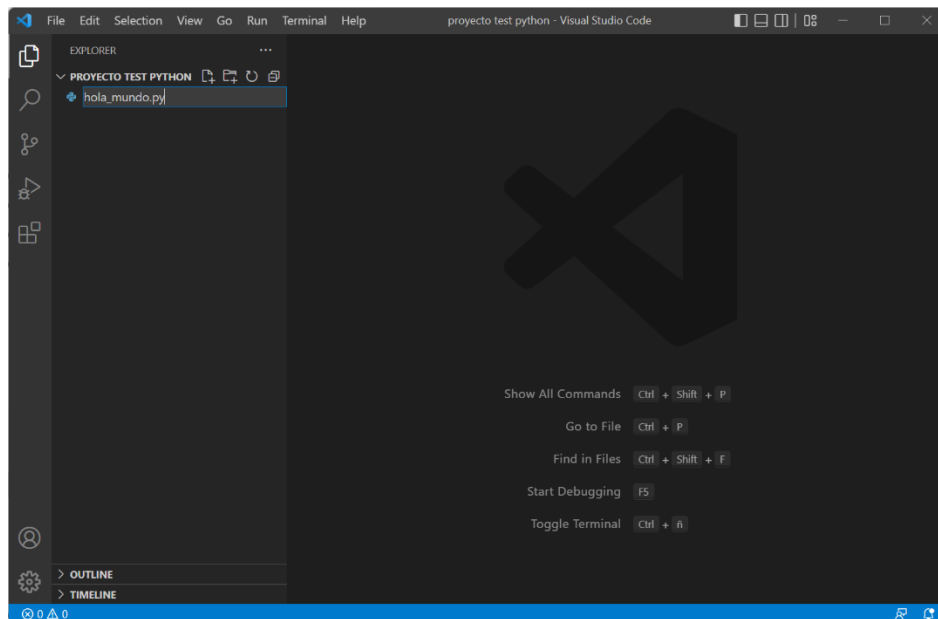


Se nos mostrará el contenido de la carpeta en el explorador. Dado que acabamos de crear la carpeta, el proyecto está vacío. Pon el cursor del ratón encima del nombre de la carpeta en el explorador, y pulsa en el primer botón que aparece, para crear un nuevo archivo:

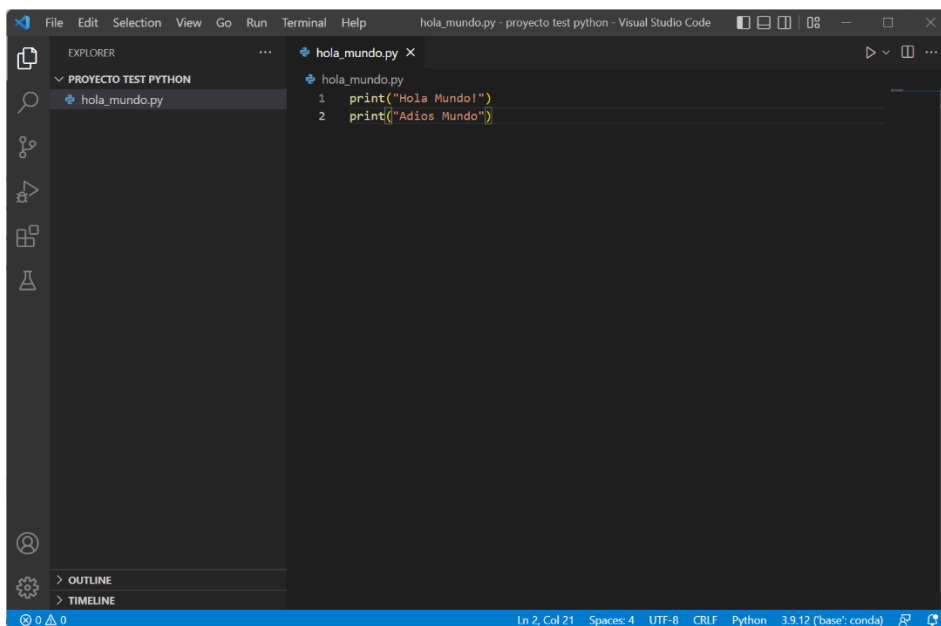


Escribe un nombre para el nuevo archivo y pulsa enter. Los archivos de código fuente en Python siempre acaban con la extensión ".py":





Ahora se nos mostrará en la parte central de la ventana un editor que nos permitirá escribir código en dicho archivo. Escribe el siguiente código de prueba:



Veamos ahora cómo ejecutar el código del archivo que estamos editando. Puedes hacerlo desde el menú superior “Run”, y luego “Run Without Debugging”. Tal como indica la interfaz, la tecla rápida para desencadenar dicha ejecución es Ctrl + F5. De cualquiera de las dos maneras, se abrirá un nuevo terminal en la parte inferior de la aplicación, y en él se ejecutará nuestro programa:

```

File Edit Selection View Go Run Terminal Help
hola_mundo.py - proyecto test python - Visual Studio Code

EXPLORER
PROYECTO TEST PYTHON
hola_mundo.py

hola_mundo.py
1 print("Hola Mundo!")
2 print("Adios Mundo!")

TERMINAL
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras.
https://aka.ms/PSWindows

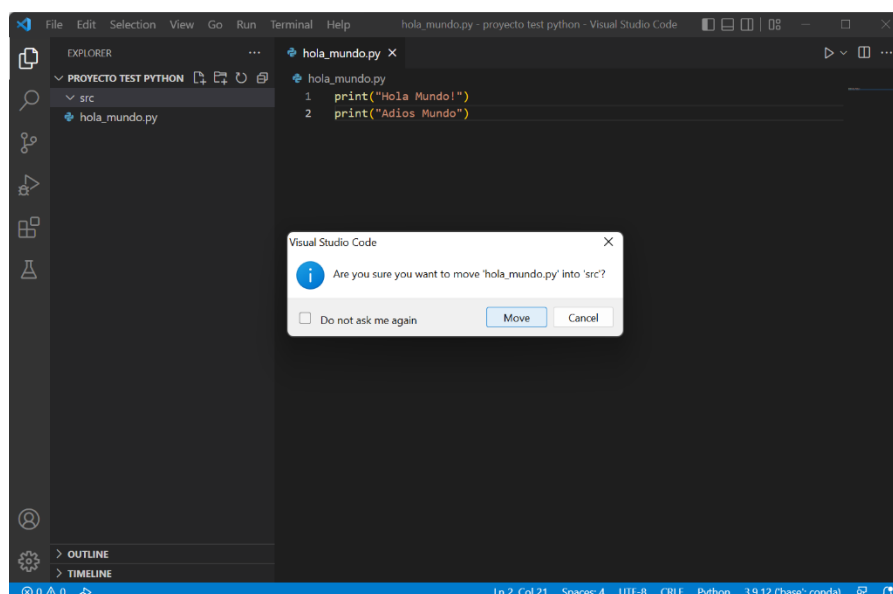
PS C:\Users\test\Documents\FP\proyecto test python> conda activate base
PS C:\Users\test\Documents\FP\proyecto test python> & 'C:\Users\test\miniconda3\python.exe'
'c:\Users\test\vscode\extensions\ms-python.python-2022.14.0\pythonFiles\python\debugpy
\adapter\..\..\debugpy\launcher' '49947' '--' 'c:\Users\test\Documents\FP\proyecto test pyth
on\hola_mundo.py'
Hola Mundo!
Adios Mundo
PS C:\Users\test\Documents\FP\proyecto test python>

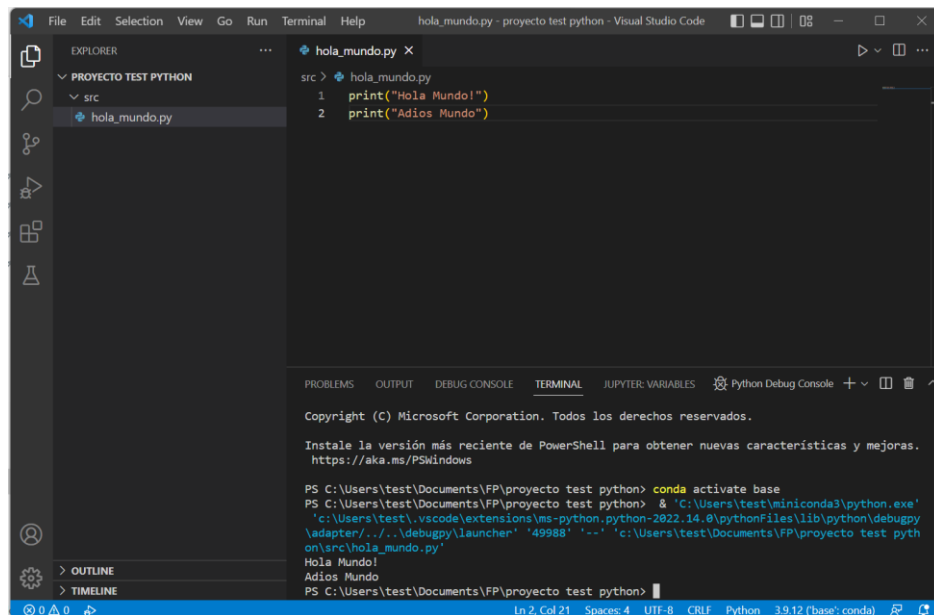
```

Observa que el terminal se parece a la aplicación “Símbolo del sistema” de Windows. Se trata de una interfaz textual que permite ejecutar aplicaciones en nuestro equipo. Para llevar a cabo la ejecución, VSC ha abierto este terminal, se ha ubicado en la carpeta que tenemos abierta (en el ejemplo mostrado, “C:\Users\test\Documents\FP\proyecto test python”), y ha ejecutado nuestro programa `hola_mundo.py` mediante el intérprete de Python que seleccionamos anteriormente. La ejecución del pequeño código que hemos escrito ocasiona la aparición de los mensajes “Hola Mundo!” y “Adios Mundo!”.

Una vez acabada la ejecución, el terminal se mantiene abierto, y podríamos escribir en él cualquier comando del sistema. Si volvemos a ejecutar el programa, la ejecución se llevará a cabo en el mismo terminal. Podemos cerrar en cualquier momento el terminal pulsando en el botón que tiene un icono de cubo de basura; en este caso, cuando se produzca una nueva ejecución, un nuevo terminal será creado; esto puede ser útil para eliminar el texto de ejecuciones anteriores y obtener una salida de nuestro programa más limpia y fácil de interpretar.

La estructura que vamos a seguir en nuestros proyectos difiere un poco de la que acabamos de ver. Para que los archivos del proyecto estén más organizados, crearemos una carpeta de nombre `src` dentro de la carpeta de nuestro proyecto (usando el segundo botón que aparece a la derecha del nombre del proyecto en el explorador), y ubicaremos en dicha carpeta los ficheros de código fuente (aquellos con la extensión “.py”). Podemos mover el archivo que creamos anteriormente arrastrándolo con el ratón encima de la carpeta `src`:

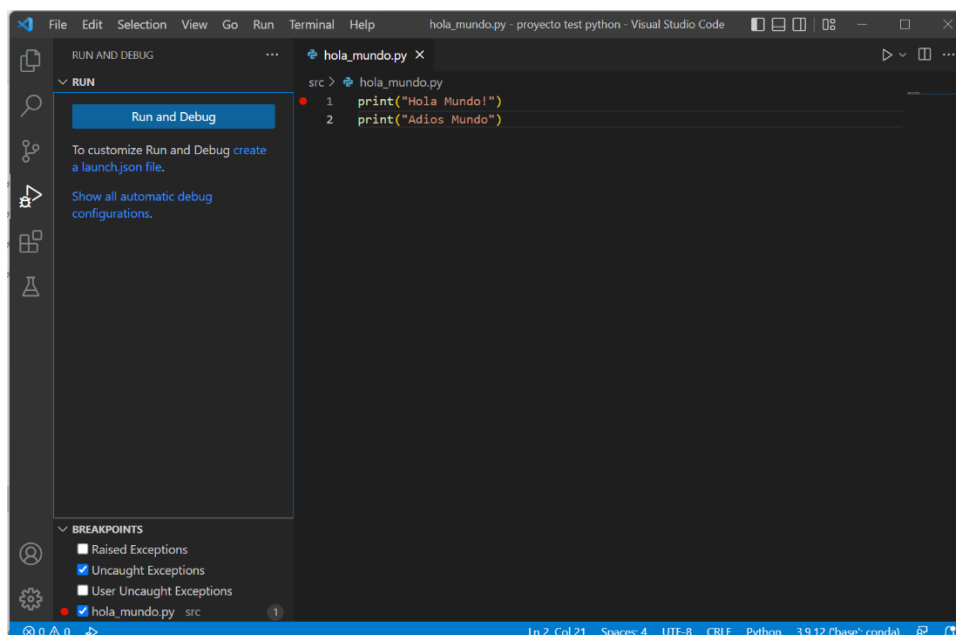




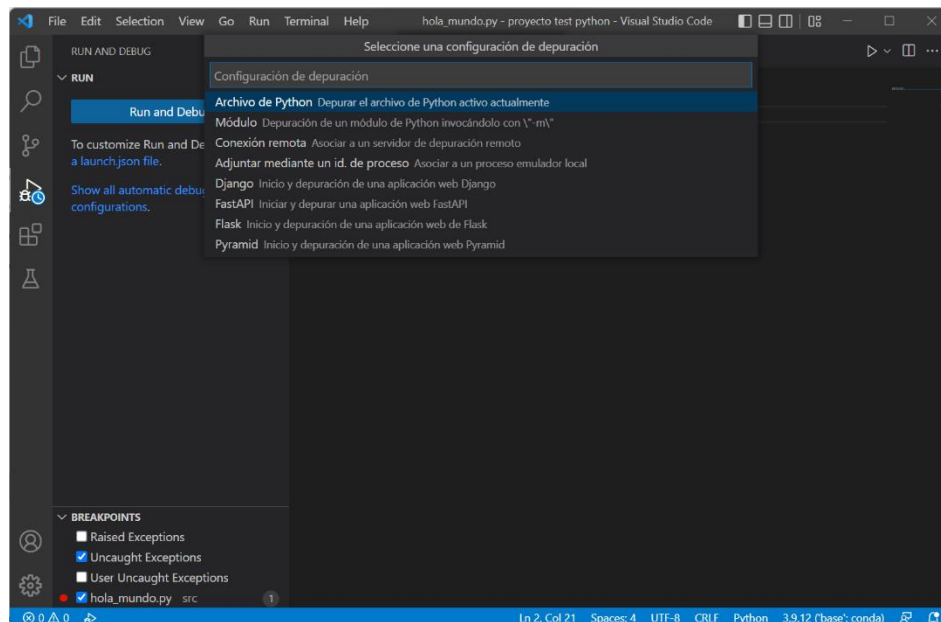
## DEPURACIÓN EN PYTHON CON VISUAL STUDIO CODE

Una herramienta fundamental en cualquier entorno de programación es el depurador. El depurador nos permite ejecutar nuestros programas paso a paso, de manera controlada, e inspeccionar el contenido de las variables del entorno en cualquier punto de la ejecución. De esta forma, podemos comprobar si los algoritmos que estamos implementando funcionan de la manera que deberían, detectando más fácilmente posibles errores en la codificación.

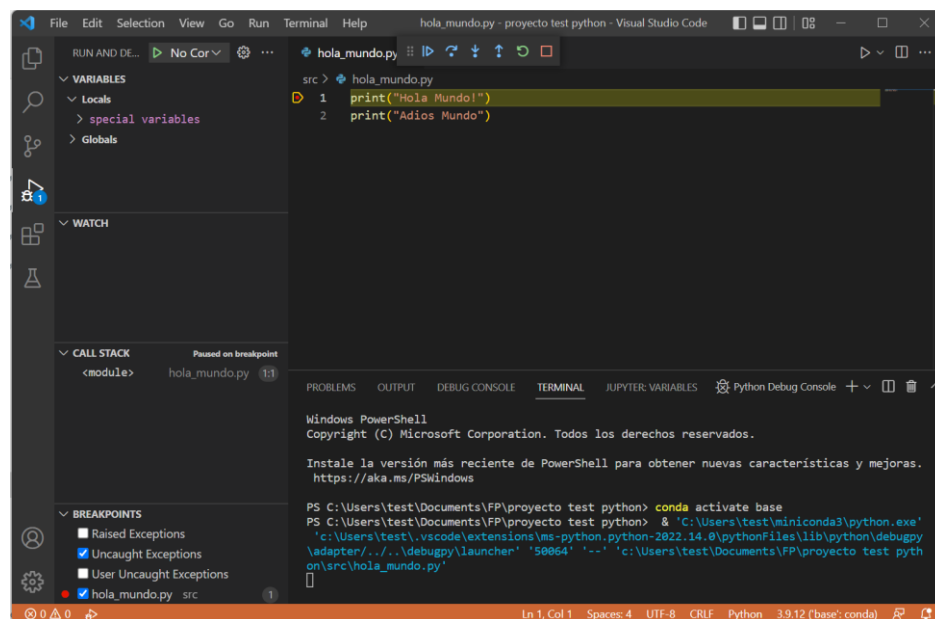
Para depurar, debemos llevar a cabo una ejecución en modo depuración de nuestro programa. Podemos hacerlo pulsando en el botón de depuración (barra izquierda, cuarto botón). Antes de pulsar el botón “Run and Debug”, vamos a seleccionar en el código un punto de interrupción, de manera que la ejecución se pausará al llegar a dicha instrucción. Para activar un punto de interrupción, lleva el cursor del ratón sobre la línea de código que desees (por ejemplo, la línea 1), sitúalo a la izquierda del número y haz clic. Debe aparecer un círculo rojo indicando que el depurador debe detenerse al llegar a este punto:



Ahora pulsa el botón “Run and Debug”, y a continuación pulsa enter para seleccionar la primera de las opciones que aparece en la paleta de comandos (“Archivo de Python...”):

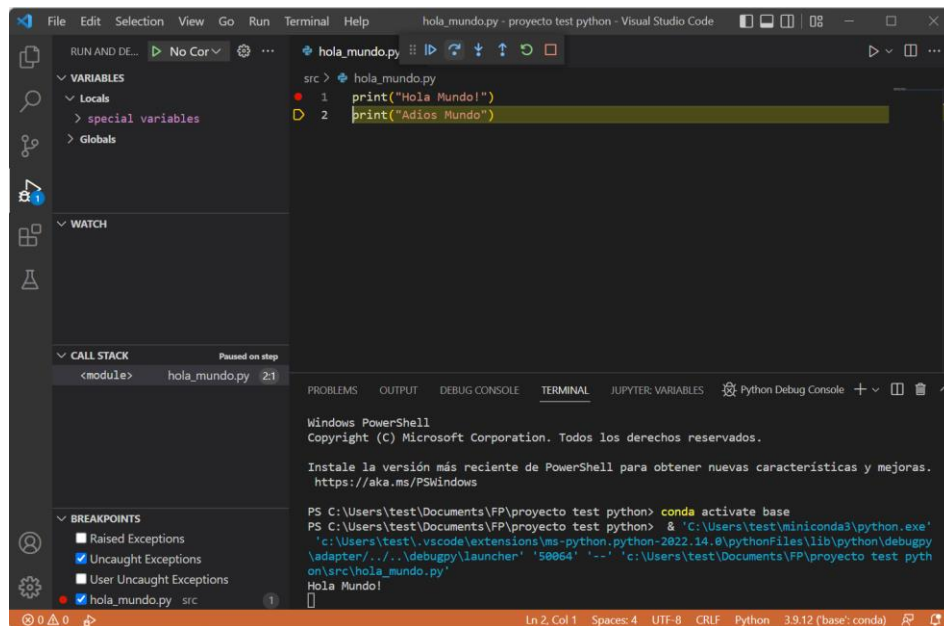


La ejecución del programa empezará, mostrando el terminal abajo como vimos antes. Pero ahora aparecerán más elementos en la interfaz, como puedes ver en la siguiente captura:



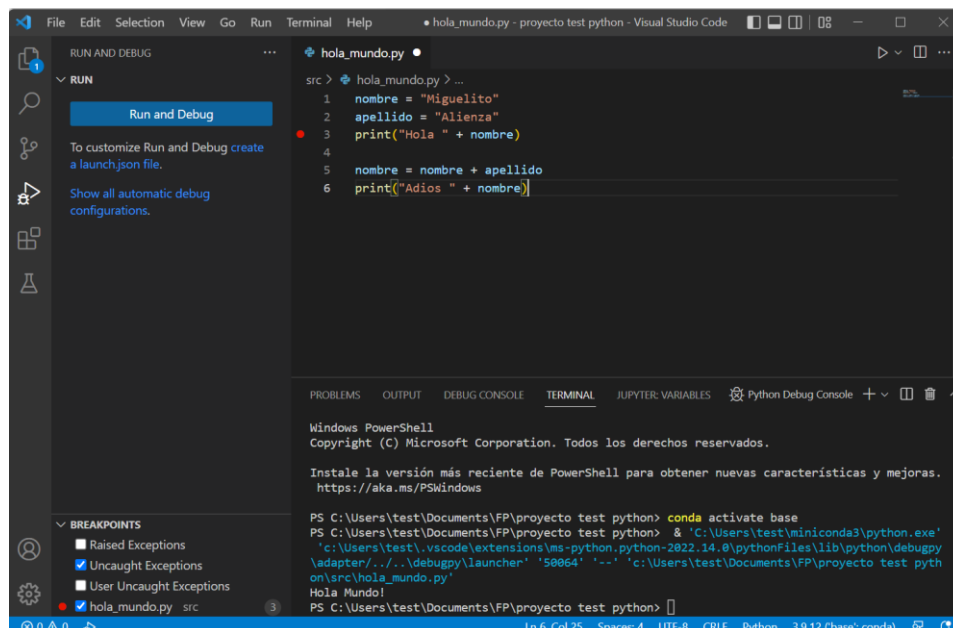
La flecha amarilla sobre la línea 1 de nuestro código indica que la ejecución se ha pausado justo antes de ejecutar dicha instrucción. La ventana VARIABLES que aparece a la izquierda nos muestra todas las variables que existen en este momento de la ejecución, así como sus valores. Como nuestro código es muy sencillo, no existe ninguna variable actualmente (en realidad, existen variables “especiales”, pero no entraremos en eso ahora).

Para hacer avanzar la ejecución, podemos pulsar el botón superior cuyo icono es una pequeña flecha curvada sobre un punto:



Al pulsar dicho botón, vemos que la flecha amarilla pasa a la siguiente línea, y en la consola ya se puede observar el efecto de la ejecución de la instrucción de la línea 1 (ha aparecido impreso el texto “Hola Mundo!”). Si pulsamos de nuevo el mismo botón anterior, la ejecución concluirá, puesto que ya no quedan más instrucciones en el programa. El mismo efecto tendría pulsar el botón que está a la izquierda del anterior (con un icono formado por una línea vertical y un triángulo), ya que este botón ejecutará nuestro programa hasta encontrar un nuevo punto de interrupción o hasta que la ejecución finalice por completo.

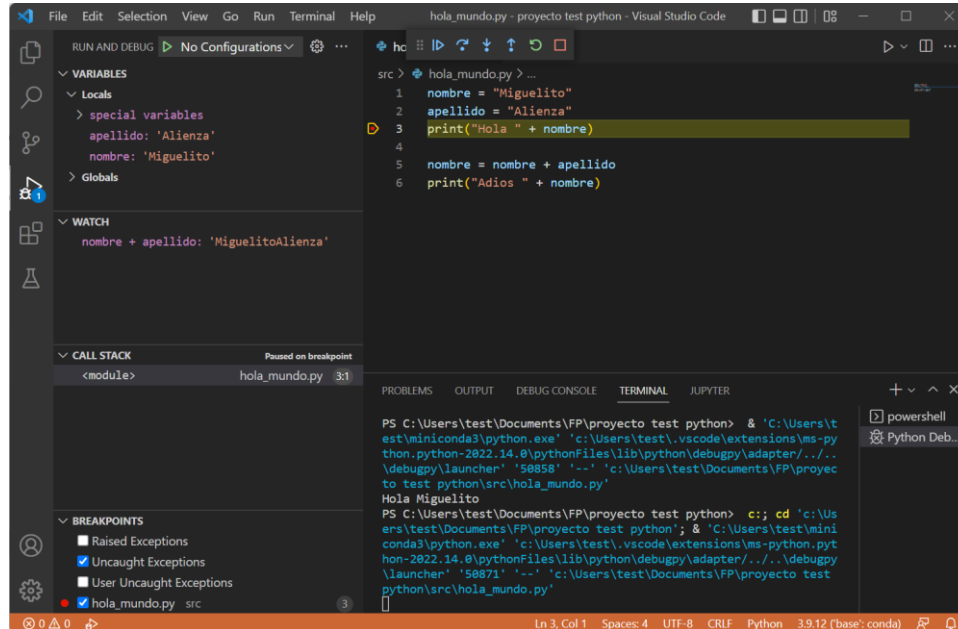
Para ejemplificar un caso de uso del depurador algo más realista, vamos a modificar nuestro programa:



Ahora estamos usando dos variables para almacenar el nombre y el apellido de una persona. Después mostramos un mensaje de saludo, usando el nombre de la persona. Posteriormente, estamos sobrescribiendo el contenido de la variable nombre, para que almacene tanto el nombre como el apellido. Por último, mostraremos una despedida, usando el nombre y el apellido que hemos almacenado en la variable nombre. Sin embargo, hemos cometido algún error en nuestro código, que ocasiona que la salida del programa no sea la que esperábamos (falta un espacio entre el nombre y el apellido mostrados):

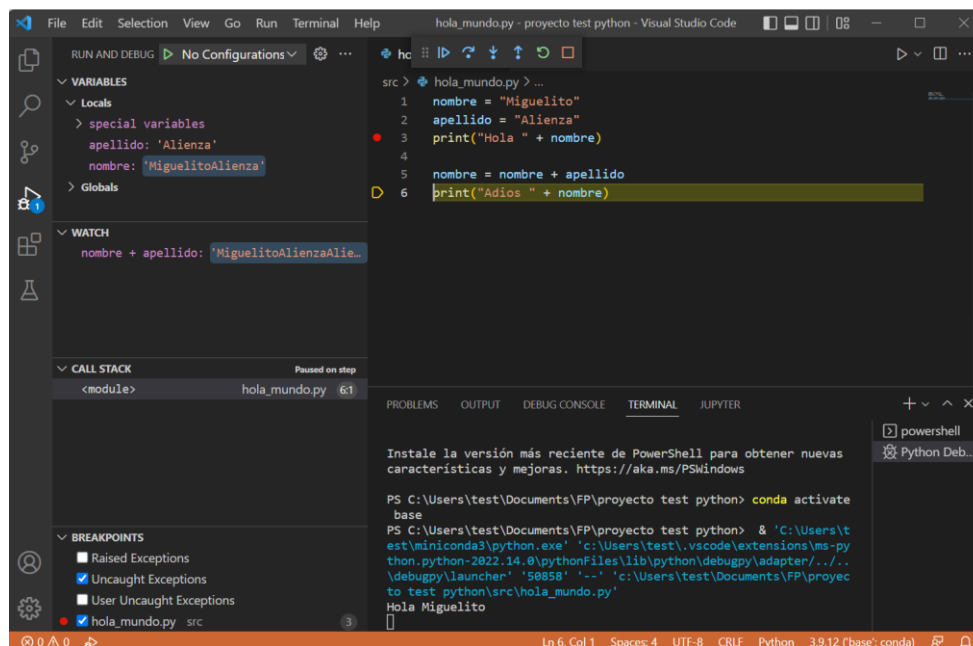
```
Hola Miguelito
Adiós MiguelitoAlienza
```

Si ponemos un punto de interrupción en la línea 3 y ejecutamos en modo depuración, podemos inspeccionar el valor de las variables justo en ese punto de la ejecución, que aparecerán en la ventana VARIABLES. Por otra parte, podemos añadir la expresión que estamos usando para concatenar el nombre y los apellidos (nombre + apellido), en la ventana WATCH. De esta forma, se visualizará cuál es el resultado de dicha expresión para el valor concreto que tienen las variables en cada momento de la ejecución. Para añadir dicha expresión en WATCH, colocamos el ratón a la derecha de WATCH y pulsamos el botón cuyo icono es un signo más, para a continuación escribir la expresión y pulsar enter:



Haciendo uso de estas herramientas, podemos tomar consciencia de errores en el código que no sean tan evidentes cuando simplemente leemos el código que hemos escrito. En nuestro ejemplo, observamos claramente que hay un error en la expresión nombre + apellido, pues hemos olvidado concatenar un carácter espacio entre el nombre y el apellido.

Si pulsamos dos veces el botón de ejecución por pasos, podemos observar que tras la ejecución de la línea 5, la variable nombre ha cambiado de valor. Esto queda resaltado en la interfaz mediante el coloreado de la línea que muestra el valor de la variable en la ventana VARIABLES:



Si ya hemos encontrado el error que buscábamos y no queremos continuar con la ejecución del resto del programa, podemos pulsar el botón más a la derecha de los mostrados arriba, cuyo icono es un cuadrado rojo.