

# Práctica 0

## Introducción

### Objetivos

- Instalar el entorno de trabajo
  - Familiarizarse con el entorno de trabajo
    - Editor de texto
    - Terminal
  - Aprender a ejecutar (no escribir) código Python
  - Intuyendo Python
  - Entregar
- 

### Instalación del entorno de trabajo

Para desarrollar en python necesitamos:

- Intérprete de Python: [Python](#)
  - Entorno de desarrollo: [Visual Studio Code](#)
- 

### Instalación de Python

- Descarga el instalador de Python desde la [página oficial](#).
  - Configura la instalación de Python en tu sistema.
  - Verifica la instalación.
-

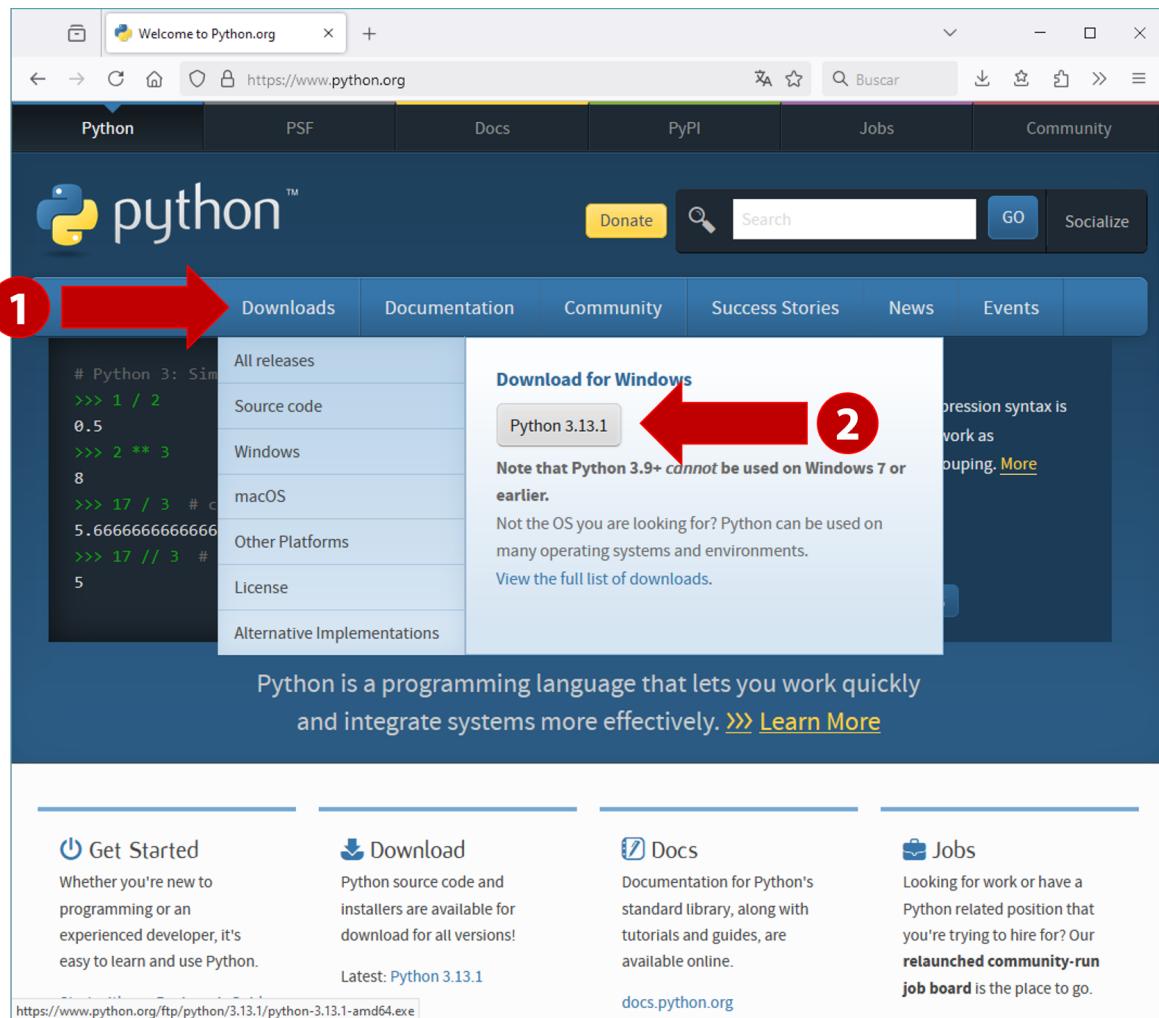


Figura 1: descarga-python

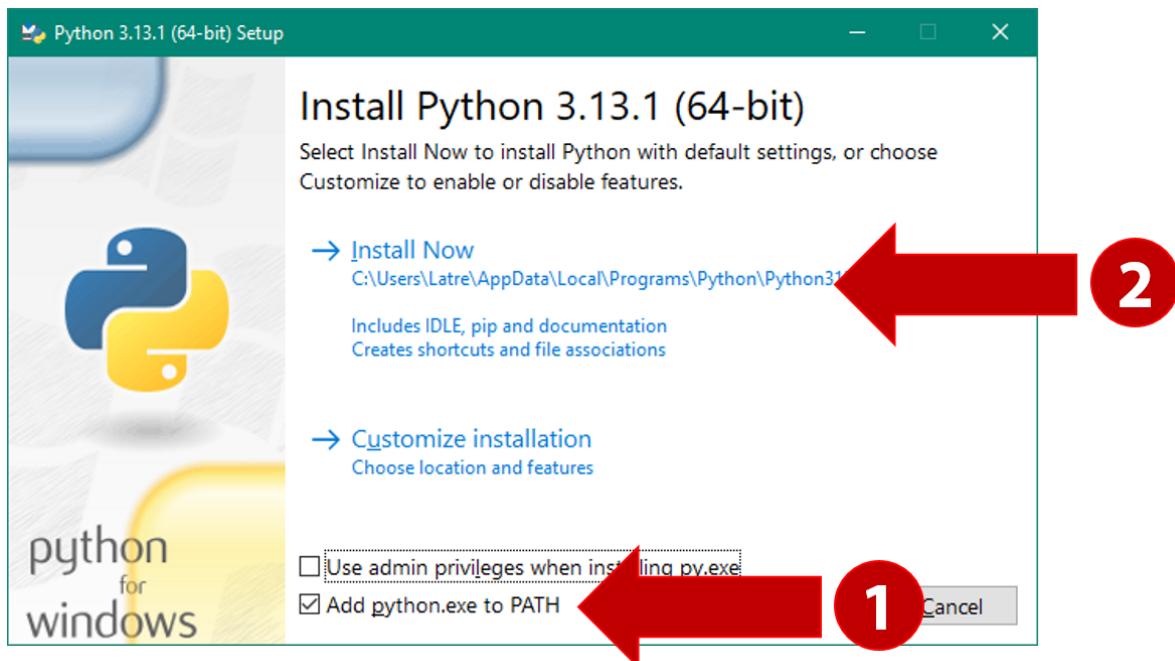


Figura 2: instalacion-python-conf-lab

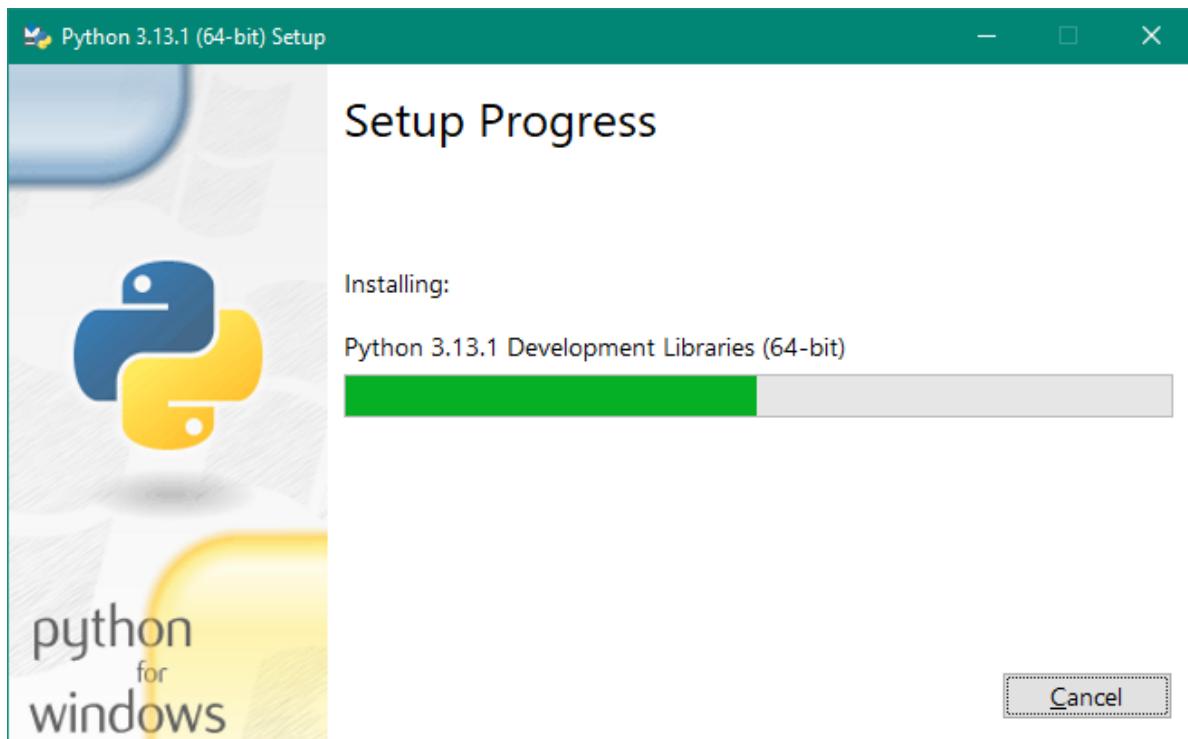


Figura 3: instalacion-python-progreso

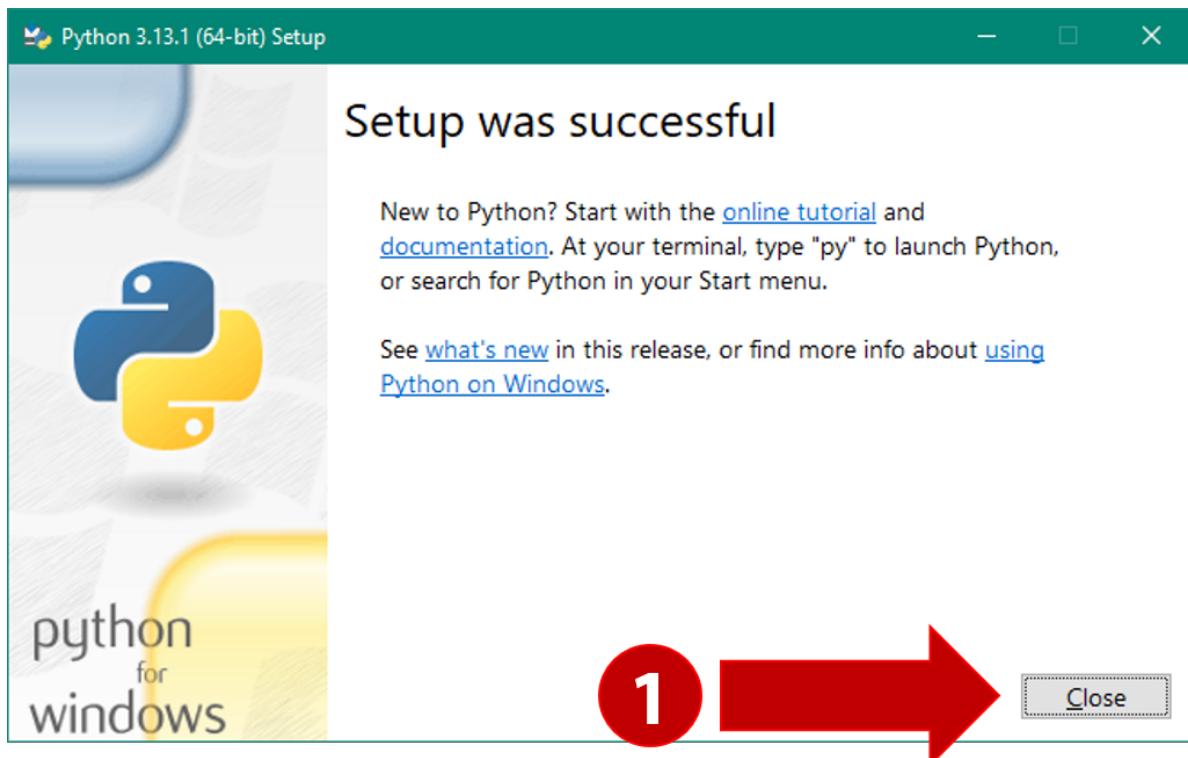


Figura 4: instalacion-python-fin

---

### Instalación de VSCode

- Descarga el instalador de Visual Studio Code desde la [página oficial](#).
  - Configura la instalación de Visual Studio Code y acepta los términos de licencia.
    - Asegúrate de seleccionar las opciones de integración con el sistema.
  - Configura el tema y las extensiones de Visual Studio Code.
-

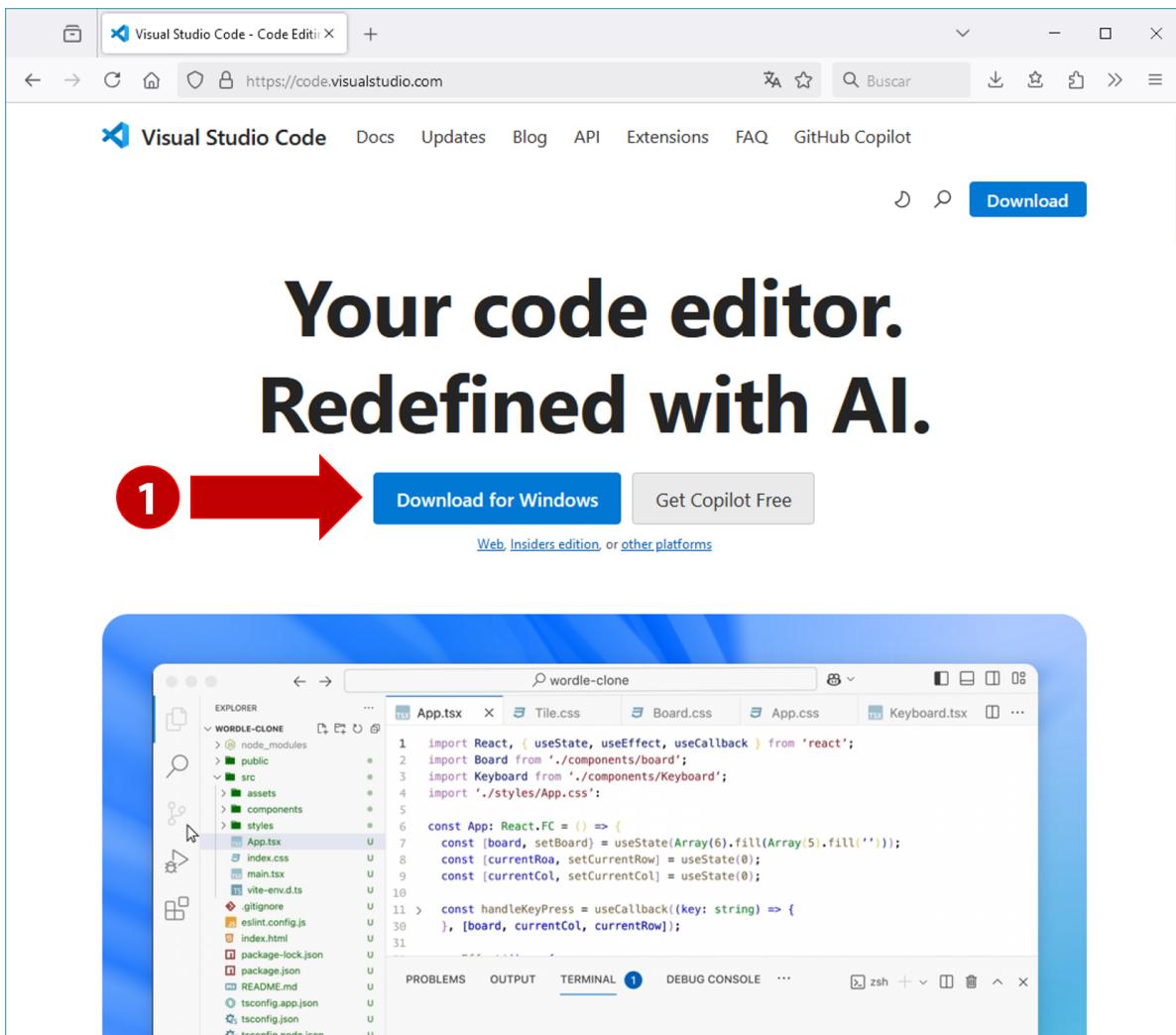


Figura 5: descarga-vsc

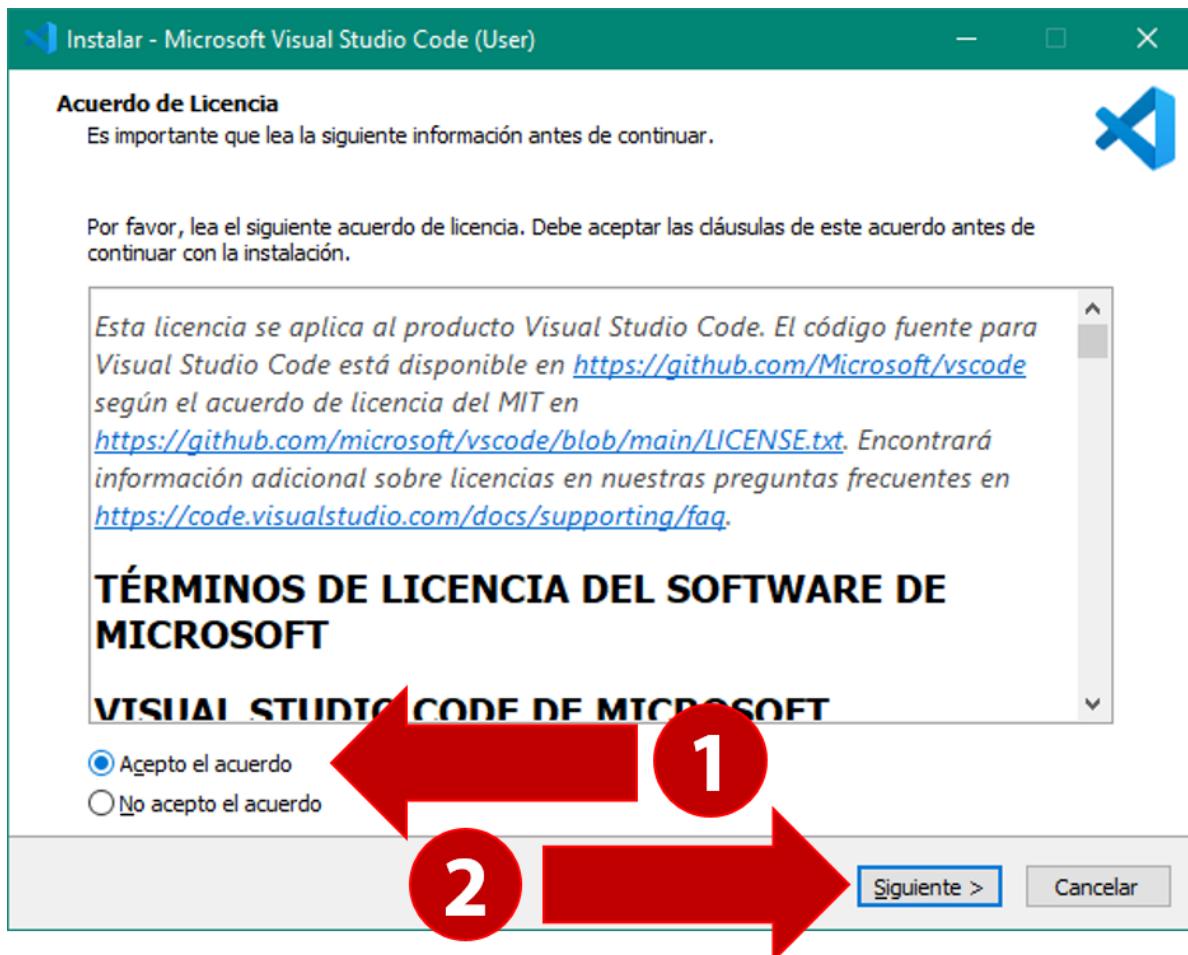


Figura 6: instalacion-vsc-licencia

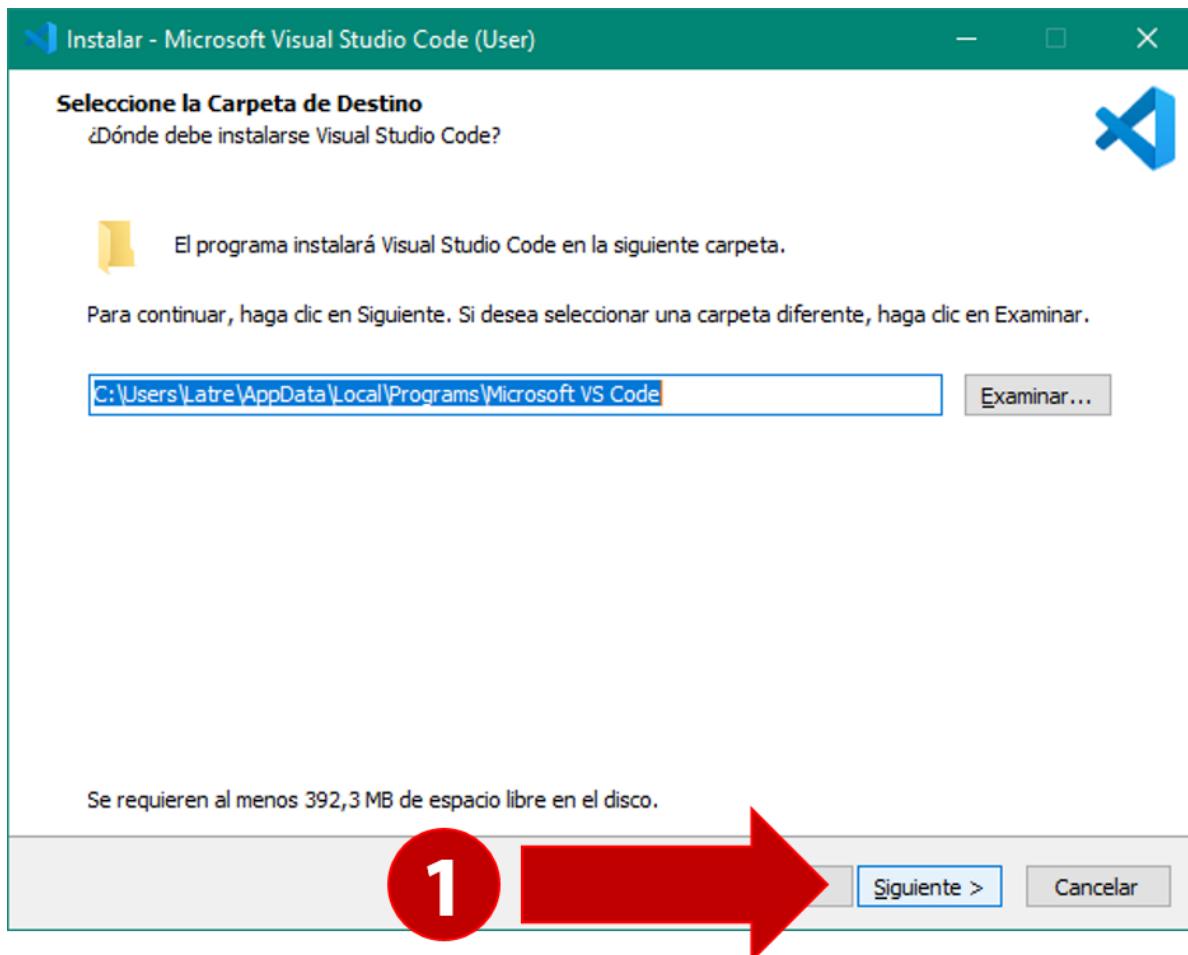


Figura 7: instalacion-vsc-dir

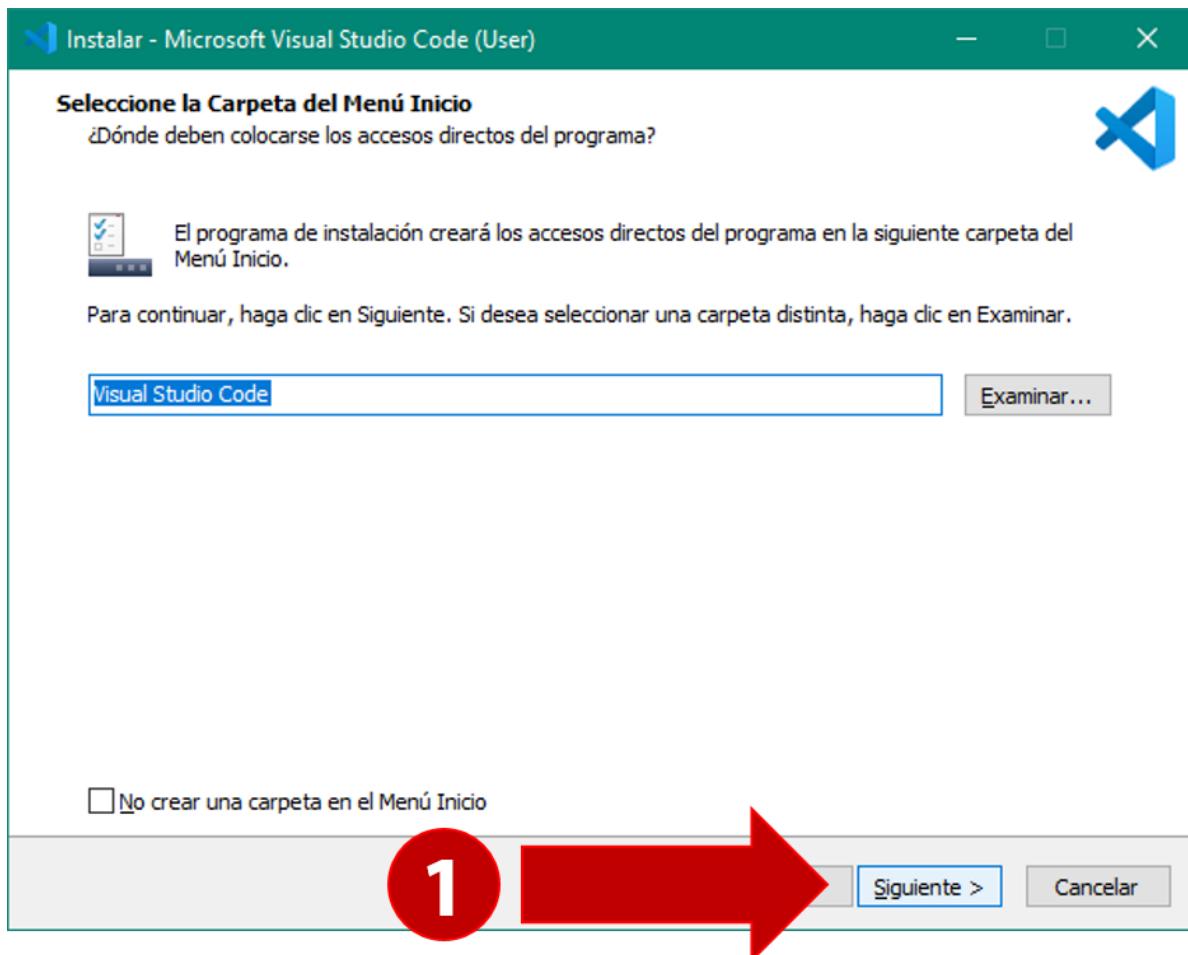


Figura 8: instalacion-vsc-menu

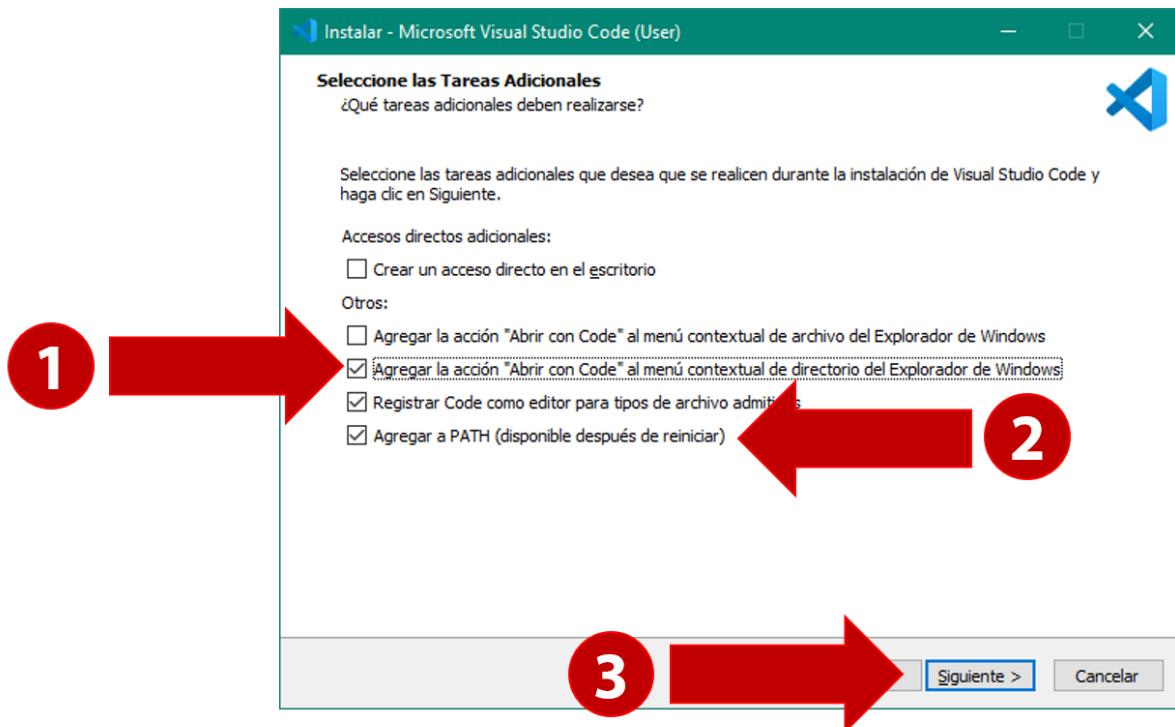


Figura 9: instalacion-vsc-extra

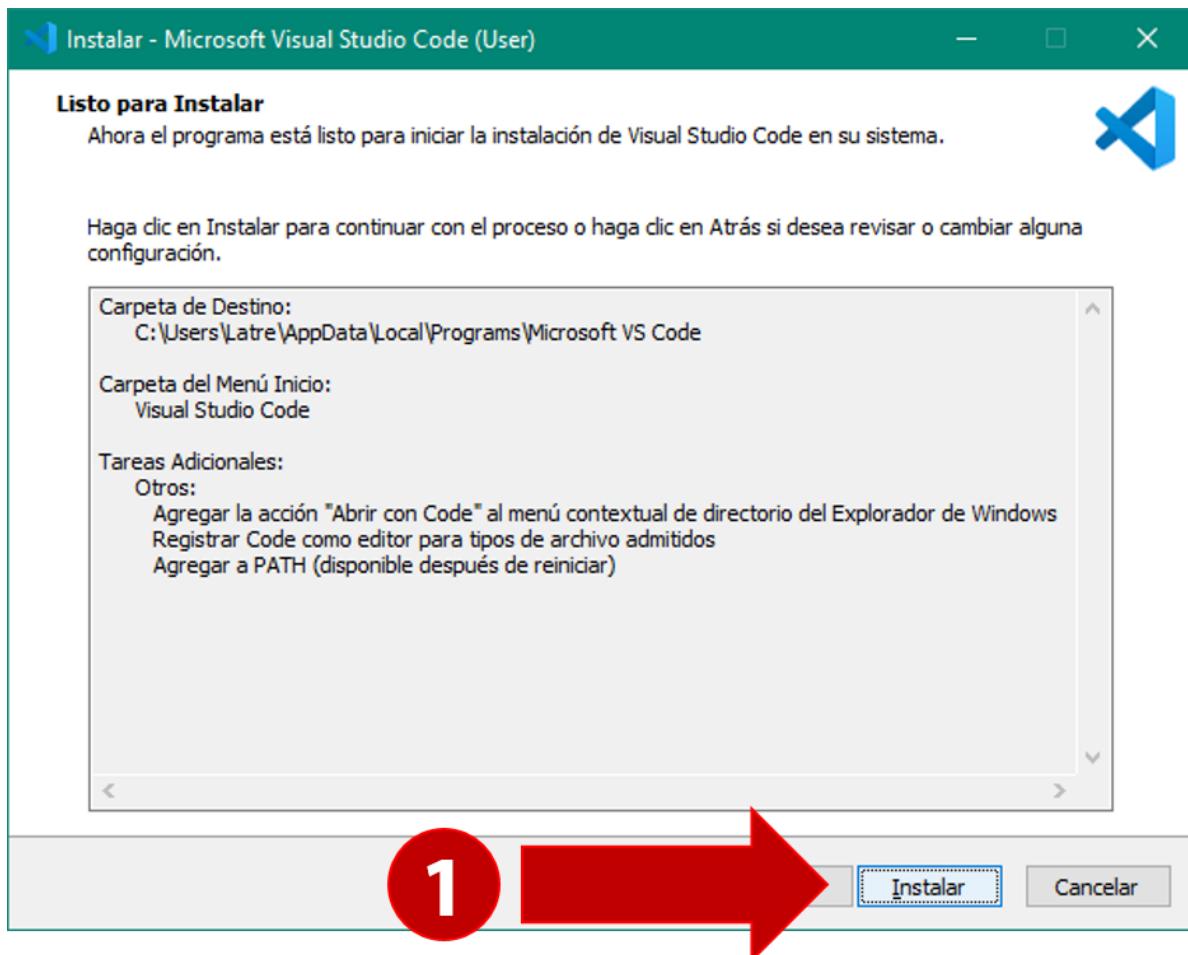


Figura 10: instalacion-vsc-ini

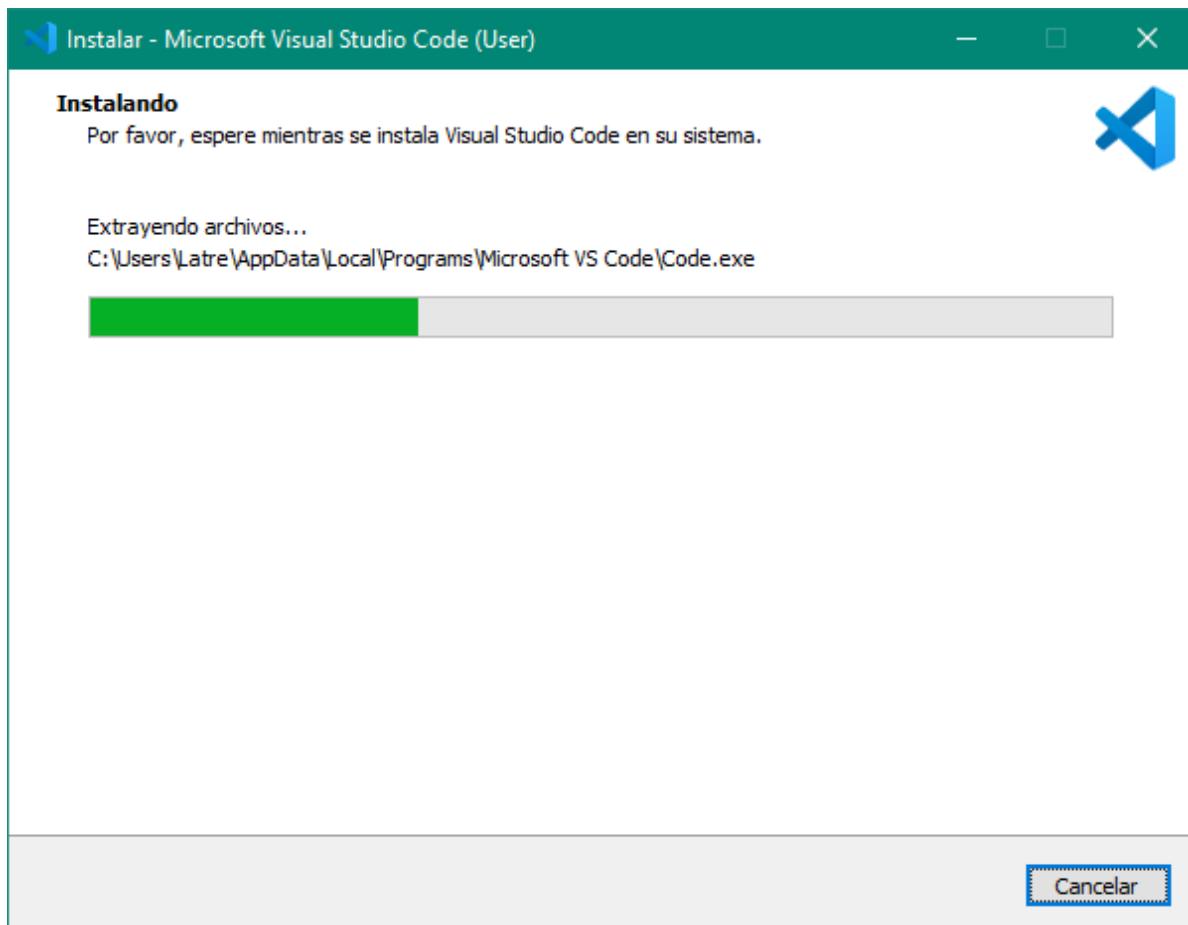


Figura 11: instalacion-vsc-progreso

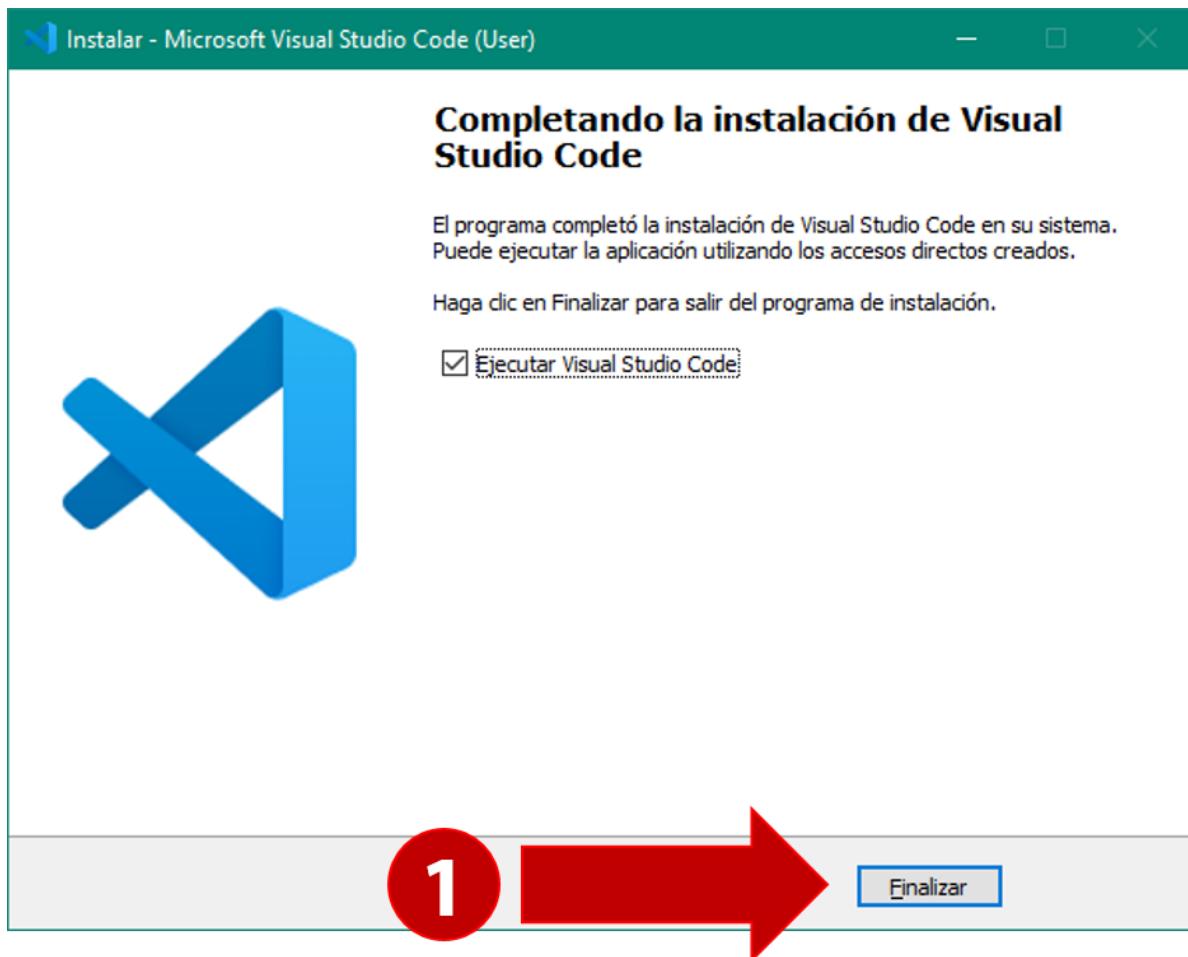


Figura 12: instalacion-vsc-fin

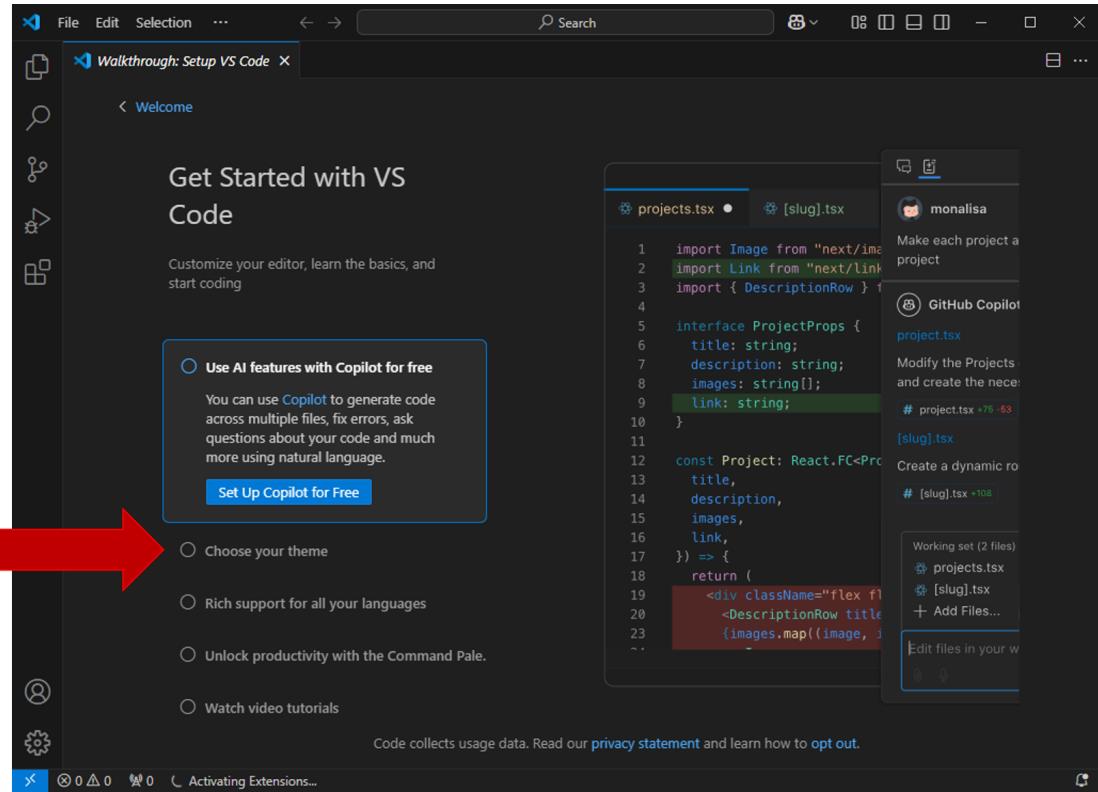


Figura 13: conf-vsc-tema

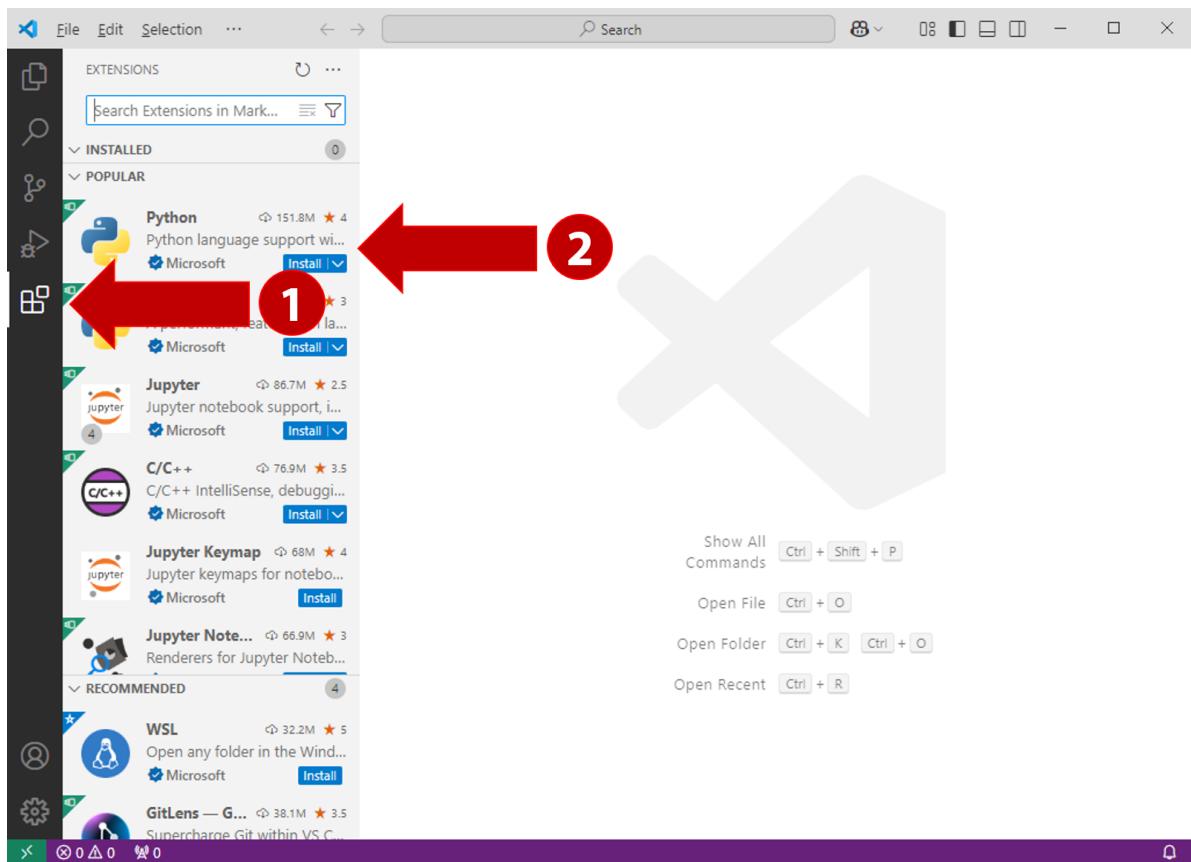


Figura 14: conf-vsc-extensiones

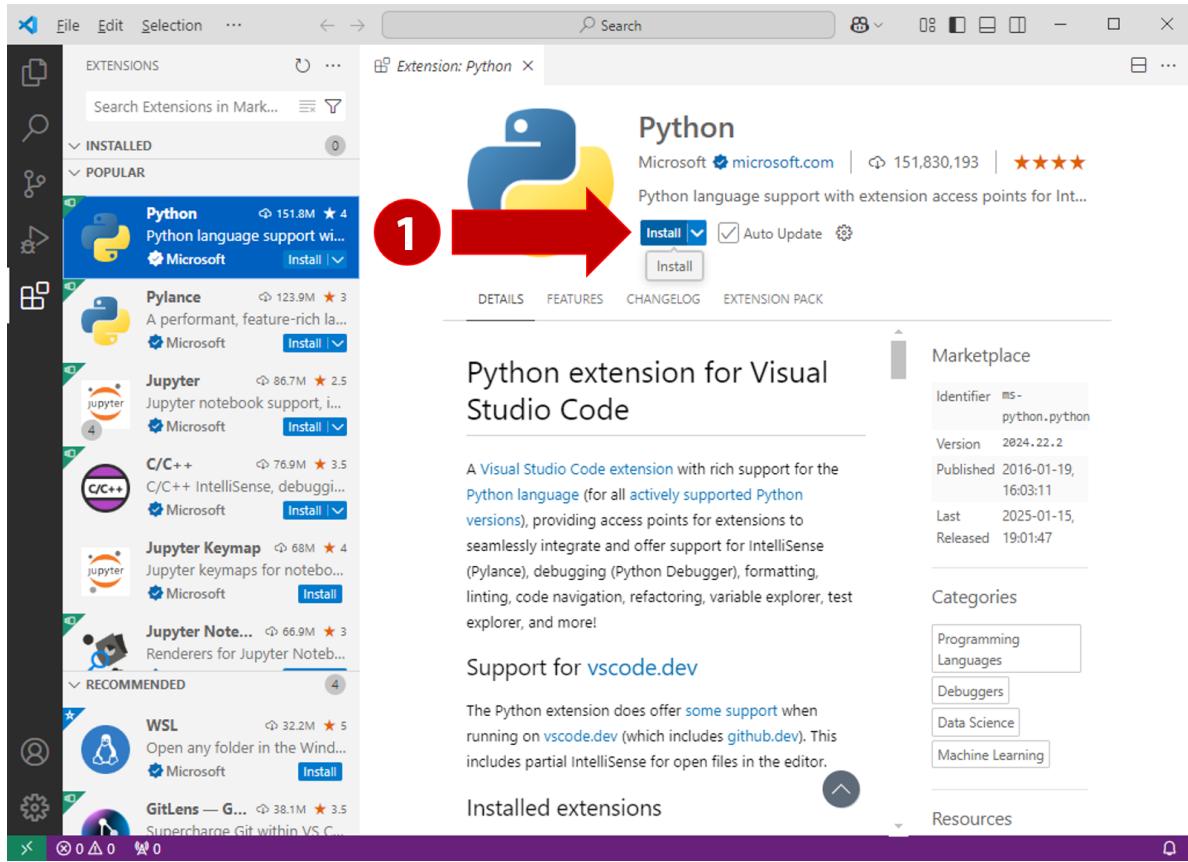


Figura 15: conf-vsc-ext-python

## Familiarizarse con el entorno de trabajo

Durante el proceso de desarrollo, siempre tendremos en cuenta:

- **Sistema ficheros:** Localización de los programas
- **Código fuente:** Contenido de los programas
- **Terminal:** Ejecución de programas python

Estos tres elementos pueden ser controlados desde **Visual Studio Code**.

## Sistema de ficheros

Se recomienda seguir la siguiente estructura para tener todos los programas organizados. Agruparemos el código de cada práctica en su propio directorio (carpeta).

```
./  
p0  
    hello.py  
    ej1.py  
    ej2.py  
p1  
...  
p6
```

---

**Tarea:** Crea la estructura de directorios en tu sistema de ficheros utilizando el **Explorador de Windows**.

1. Clic derecho > Nueva > Carpeta > p0
2. Accede a la carpeta p0
3. Clic derecho > Nuevo > Documento de texto > saludo.py
  1. Asegúrate de que la extensión del fichero es .py

*Nota:* Algunas versiones de Windows ocultan por defecto las extensiones de los ficheros. Puedes mostrarlas en Vista > Mostrar u ocultar > Extensiones de nombre de archivo.

---

## Visual Studio Code

**Tarea:** Abrid Visual Studio Code:

- Forma 1:
  1. En el explorador > Acede a la carpeta p0
  2. Clic derecho > Abrir directorio en VSCode
  3. Confirma que confías en el directorio
- Forma 2:
  1. Abre Visual Studio Code

- 
2. Abrir carpeta... > Selecciona la carpeta p0
  3. Confirma que confías en el directorio

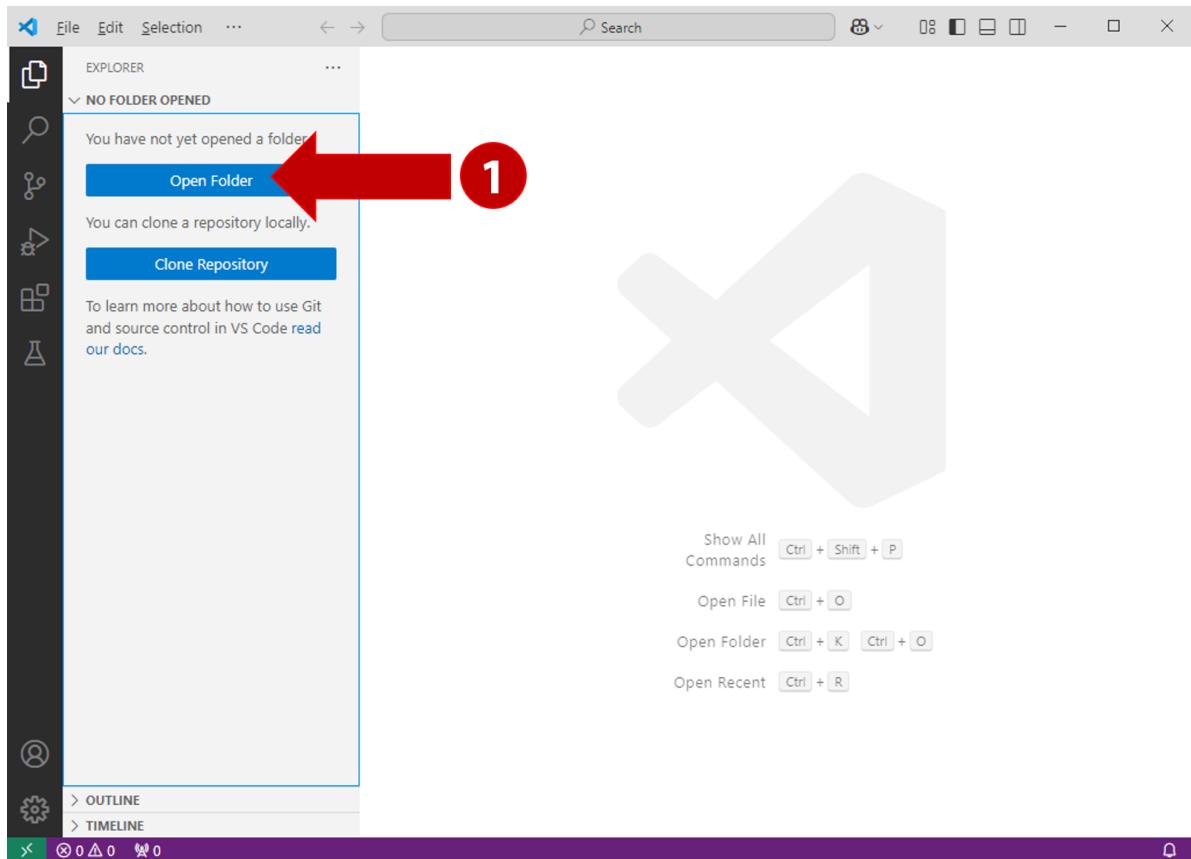


Figura 16: trabajo-vsc-abrir-carpeta

---

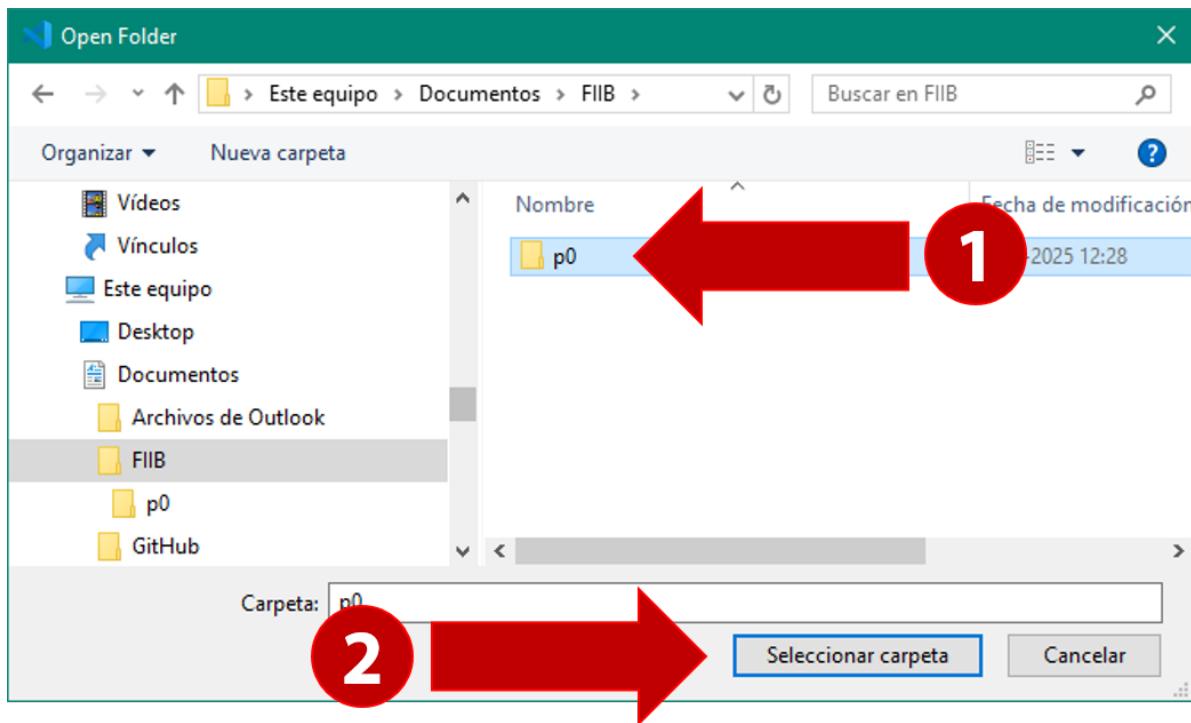


Figura 17: trabajo-vsc-sel-dir

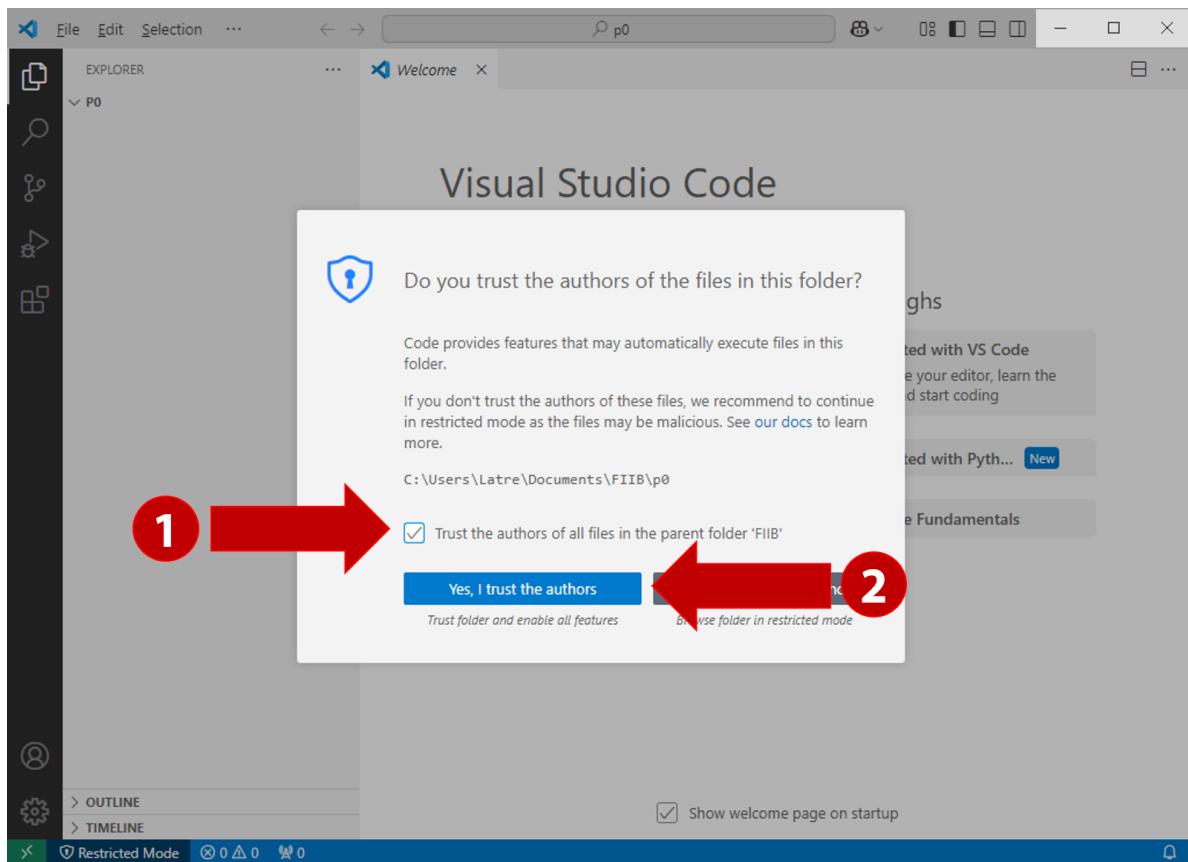


Figura 18: trabajo-vsc-trust

### VSCode: Árbol de ficheros

El árbol de ficheros muestra la estructura de directorios y ficheros del proyecto.

**Tarea:** Crea un nuevo fichero `hello.py` en la carpeta `p0`.

1. Nuevo fichero (Icono del fichero con el signo +)
2. Escribe el nombre

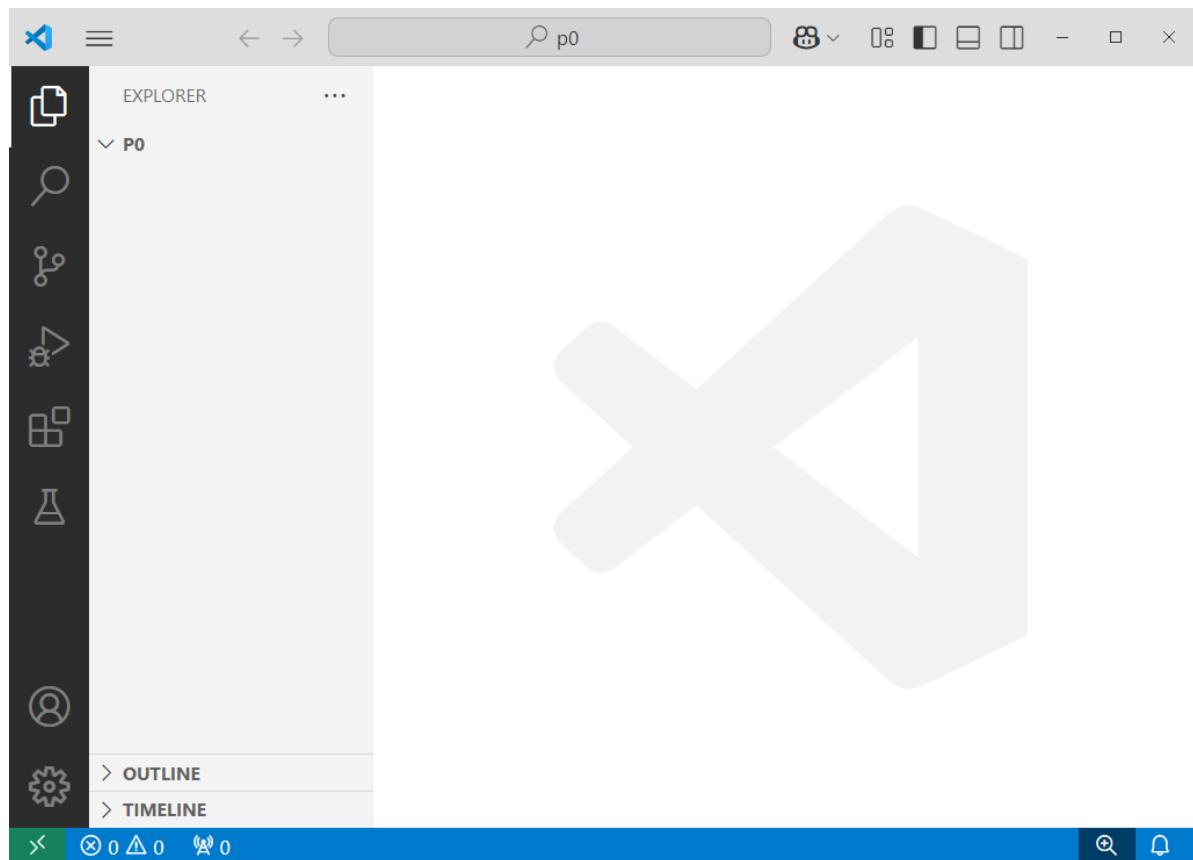


Figura 19: trabajo-vsc-entorno

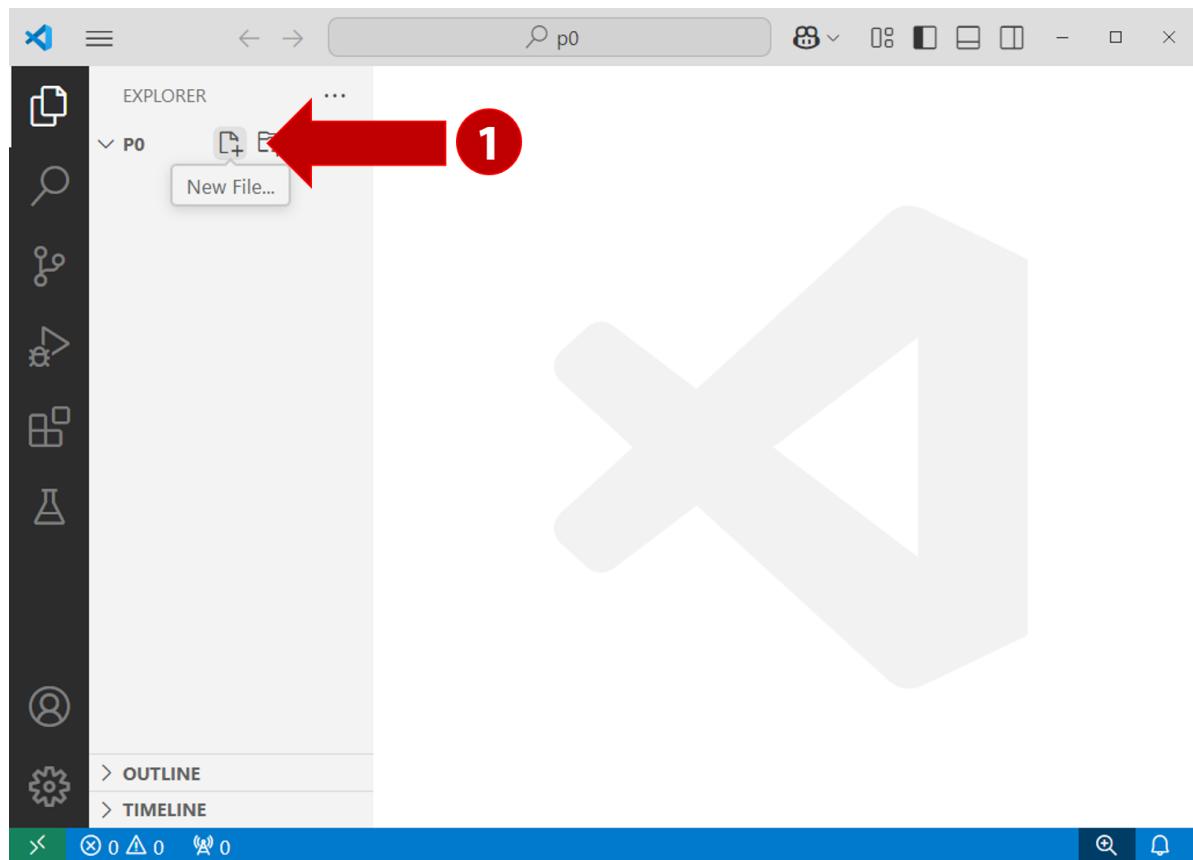


Figura 20: trabajo-vsc-crear-fichero

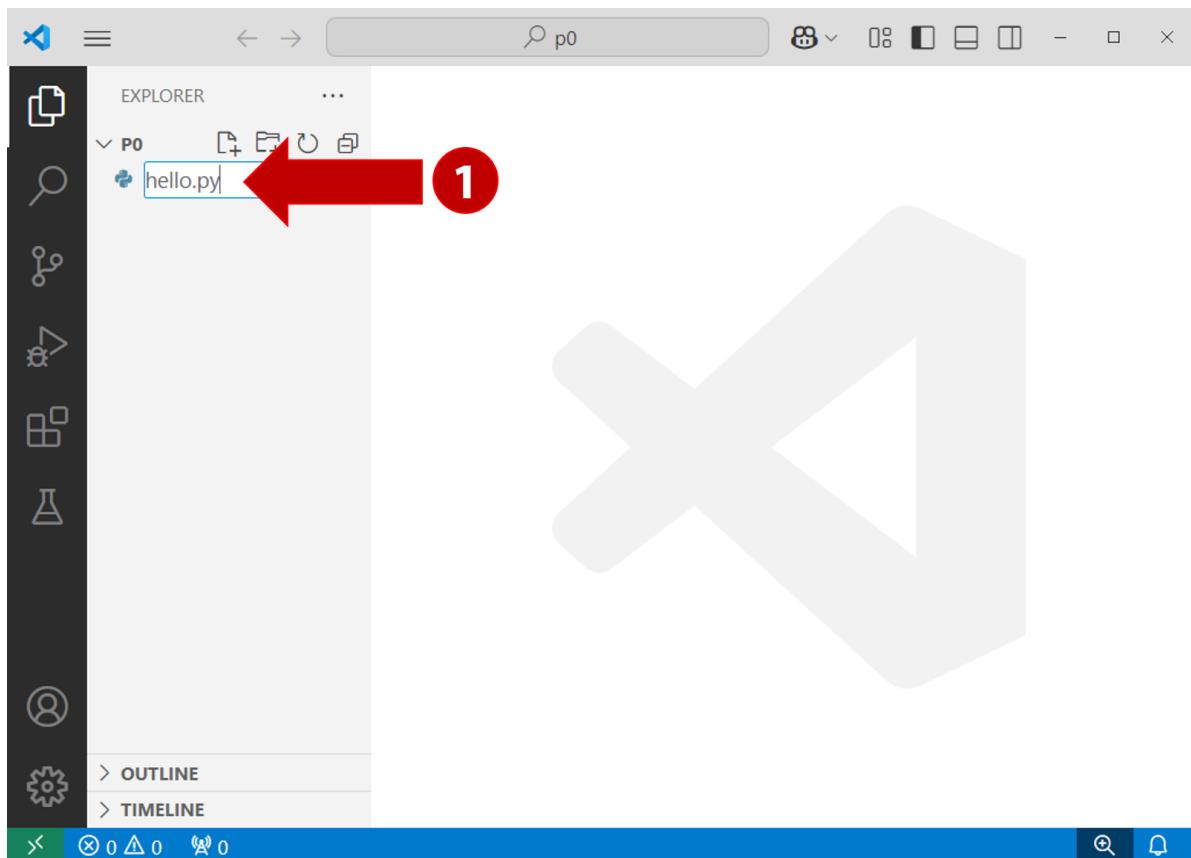


Figura 21: trabajo-vsc-nombrar-fichero

---

### VSCode: Editor de texto

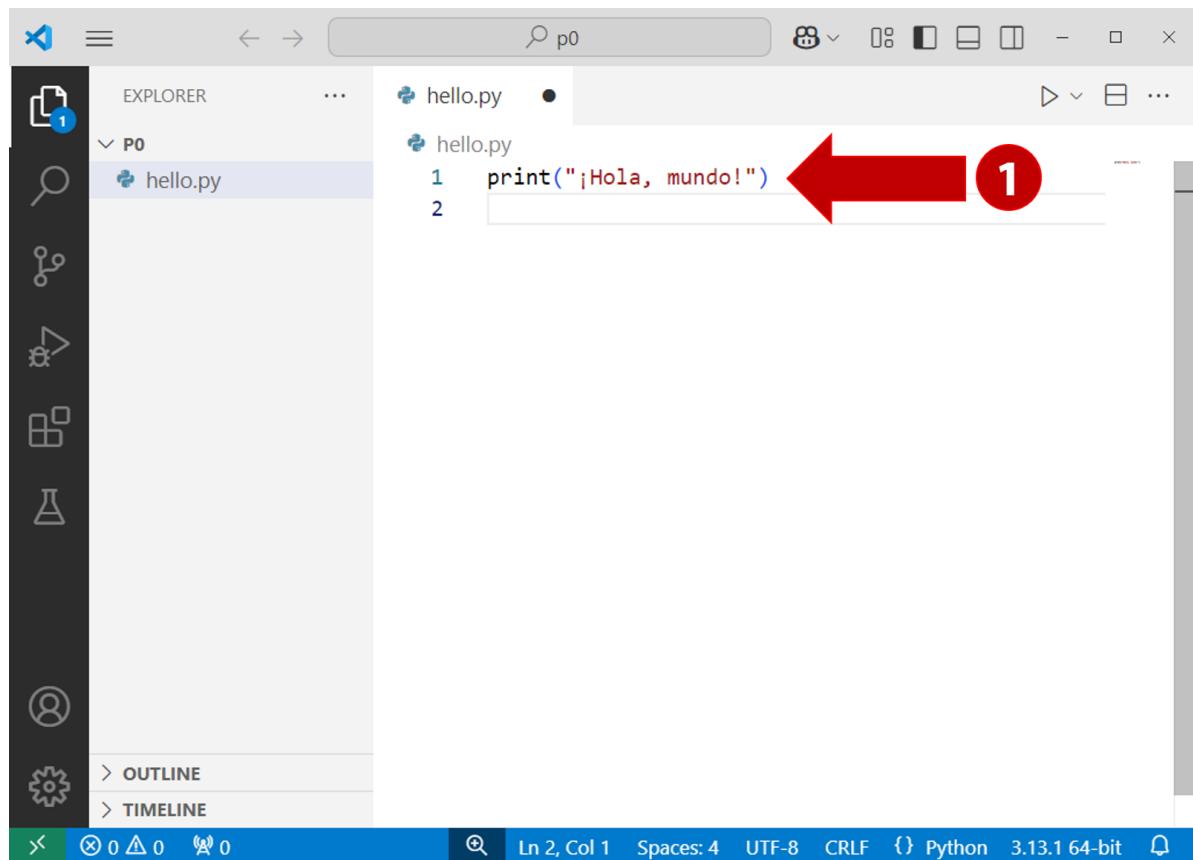
El editor de texto es donde escribiremos el código fuente de los programas.

**Tarea:** Escribe tu primer programa Python.

1. Abre el fichero `hello.py` desde el árbol de ficheros
2. Escribe el siguiente código:
3. Guarda los cambios (Ctrl+S)

```
print("Hello, World!")
```

---



A screenshot of the Visual Studio Code (VSC) interface. The left sidebar shows icons for Explorer, Search, Problems, Files, Terminal, and others. The main area displays a Python file named 'hello.py' with the following content:

```
1 print("¡Hola, mundo!")
```

A large red arrow points from the right towards the code line, and a red circle with the number '1' is placed over the arrow. The status bar at the bottom shows: ⌛ 0 △ 0 ⚡ 0, Line 2, Col 1, Spaces: 4, UTF-8, CRLF, Python 3.13.1 64-bit.

Figura 22: trabajo-vsc-escribir-codigo

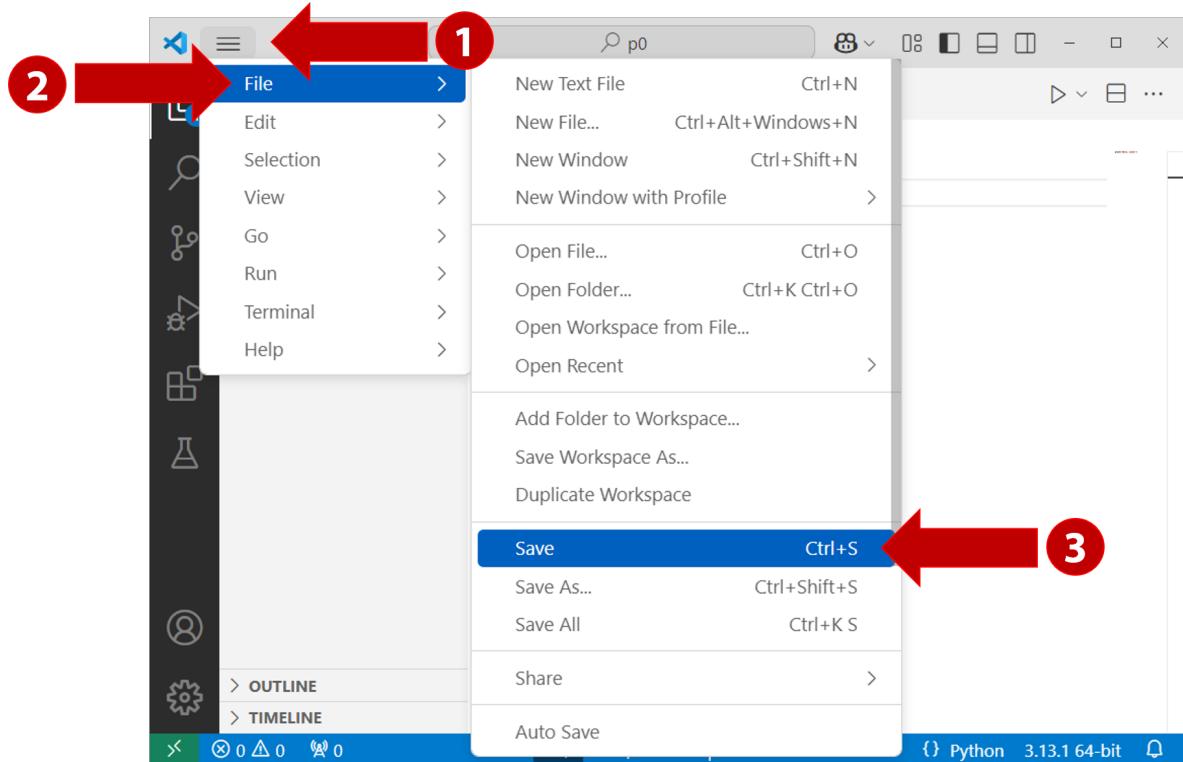


Figura 23: trabajo-vsc-guardar

---

### VSCode: Terminal

La terminal es donde ejecutaremos los programas.

**Tarea:** Ejecuta el programa `hello.py` desde la terminal de VSCode.

1. Abre la terminal de VSCode (`Terminal > Nueva terminal`)
2. Ejecuta el programa `hello.py` con el siguiente comando:

```
py hello.py
```

---

Otra forma más rápida de ejecutar el programa es pulsando el botón de Run que aparece en la parte superior derecha del editor.

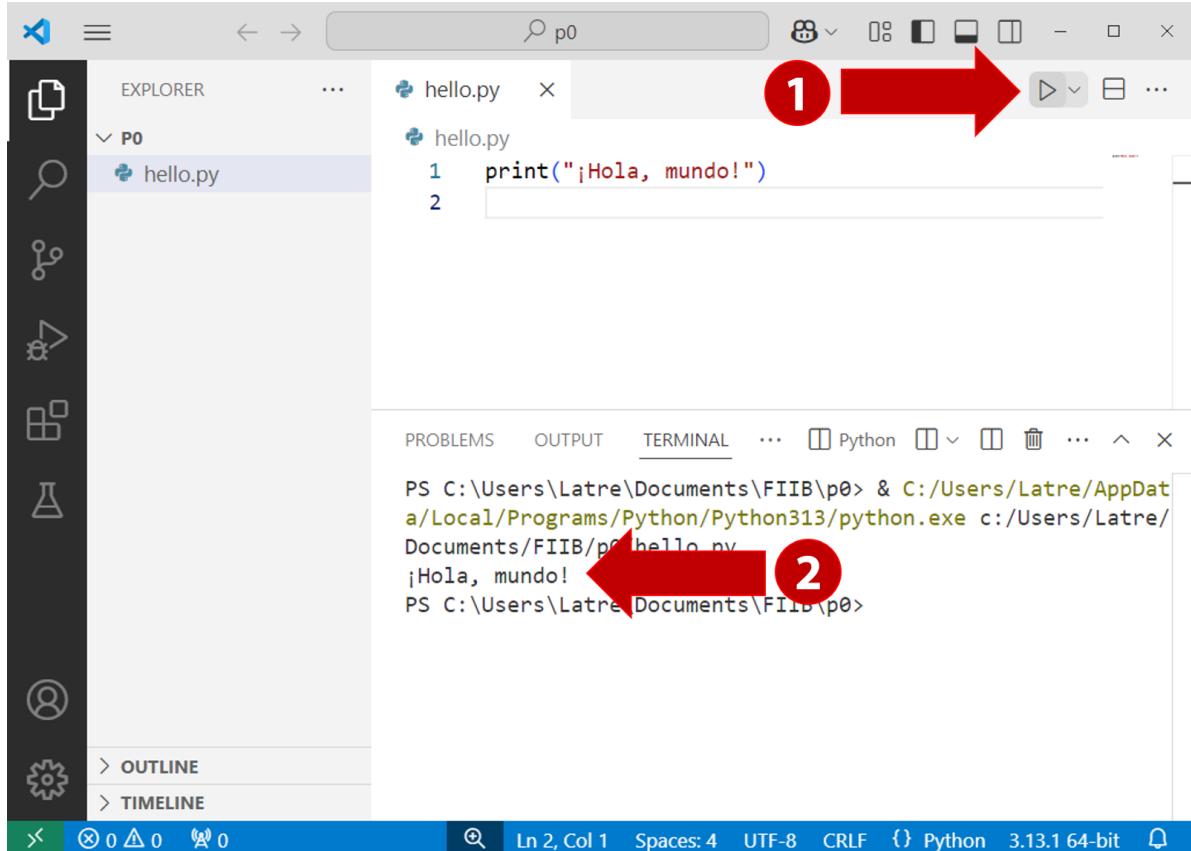


Figura 24: trabajo-vsc-ejecutar

---

## Programa interactivo

- Más adelante en el curso aprenderás a hacer programas que requieran de la **interacción** del usuario.
  - Dicha interacción se realiza también en la propia **terminal**.
  - Para familiarizarte ahora con este comportamiento, vamos a escribir y ejecutar un programa interactivo.
-

**Tarea:** Crea y ejecuta un programa interactivo.

1. Utiliza el árbol de ficheros para crear el fichero `saluda.py`
2. Escribe el **código** en `saluda.py`.
3. Ejecuta el programa `saluda.py` desde la terminal de VSCode o pulsando el botón de Run.

```
""" Sergio Martín Segura - 622612
El programa pide un nombre al usuario.
Luego escribe en la pantalla "Hola, NOMBRE".
"""

# Pide al usuario que escriba su nombre, lo lee de teclado y lo guarda en «nombre»
nombre = input("¿Cómo te llamas? ")

# Saluda al usuario escribiendo un mensaje personalizado en la pantalla
print("Hola,", nombre)
```

---

### Comentarios de código y docstring

Observa cómo en la primera línea del programa `saluda.py` se describe el funcionamiento del programa.

En un fichero python podemos escribir tres “tipos de texto” distintos:

- **Código:** Líneas de código que se ejecutan
  - Sigue la sintaxis de Python
- **Docstring:** Texto que sirve para documentar el código
  - Se escribe entre tres comillas-dobles """ ... """
- **Comentarios:** Cualquier otro texto que no se ejecuta ni se interpreta
  - Se escribe tras el símbolo #

## Intuyendo Python

A continuación evaluaremos nuestra capacidad de intuir y entender el funcionamiento de código python.

**Tarea:** Documenta los siguientes programas en un *docstring*.

1. Descarga o copia los ficheros en la carpeta p0.
2. Ábrelos con el editor de texto y lee su contenido.
3. Ejecuta los programas desde la terminal de VSCode.
4. Añade un *docstring* al principio de cada fichero incluyendo tu nombre y explicando qué hace el programa.

*Nota:* Puedes hacer Clic derecho > Guardar enlace como... en los nombres de cada fichero para descargarlos o hacer clic en el ícono del portapapeles para copiar todo el contenido.

---

### ej1.py

```
a = int(input("Introduce un número: "))
b = int(input("Introduce otro número: "))

print(a + b)
```

### ej2.py

```
pi = 3.14159265358979323846
x = float(input("Escribe un número real: "))

print(pi * (x ** 2))
```

### ej3.py

```
x = float(input("Introduce un número: "))

if x >= 0:
    print(x)
else:
    print(-x)
```

---

## **Entrega de la práctica**

Todas las prácticas se entregarán a través de **Moodle**. Esta Práctica 0 no es evaluable, pero sí se deberá entregar para familiarizarse con el proceso.

**Tarea:** Entrega el código de la práctica.

- Verifica que has nombrado correctamente los ficheros. Un fichero mal nombrado no será evaluado.
- Verifica que has añadido un *docstring* a cada fichero con tu nombre y NIP.
- Verifica que el funcionamiento de los programas es correcto.
- Accede a la entrega en Moodle y lee si hay instrucciones extra.
- Sube los ficheros \*.py individualmente (sin comprimir y sin crear carpetas adicionales).

*Nota:* No se aceptarán entregas fuera de plazo. En el caso de que se acerque la fecha límite de entrega, se recomienda al menos subir los ejercicios que se hayan terminado. **Se podrán resubir** tantas veces como se desee antes de la fecha límite de entrega.