

Actividad ED03_2 - Instalación y configuración de Visual Studio Code para diversos lenguajes informáticos.

Objetivo

El objetivo de esta práctica es investigar y realizar los pasos necesarios para instalar y configurar **Visual Studio Code (VS Code)**, para ser empleado utilizando diversos lenguajes de programación. Se deberá, además, instalar y usar las extensiones más útiles para dichos lenguajes y explorar características importantes del editor que faciliten el desarrollo de software.

Se deben realizar todas las actividades indicadas en el enunciado. Si algún software o plugin ya ha sido instalado, debe indicarse en la memoria a entregar la ausencia de dicha instalación por disponer de las herramientas indicadas.

Esta práctica puede realizarse tanto en SO Windows o Linux. A continuación, se detallan las tareas a realizar en esta práctica.

1. Instalación de Visual Studio Code

Si aún no lo tienes instalado, descarga e instala la última versión de Visual Studio Code desde <https://code.visualstudio.com>.

2. Instalación de Python y configuración de plugins

Descarga e instala la última versión de Python desde su sitio oficial: <https://www.python.org/downloads/>. Asegúrate de que la instalación incluya la opción "Añadir Python al PATH".

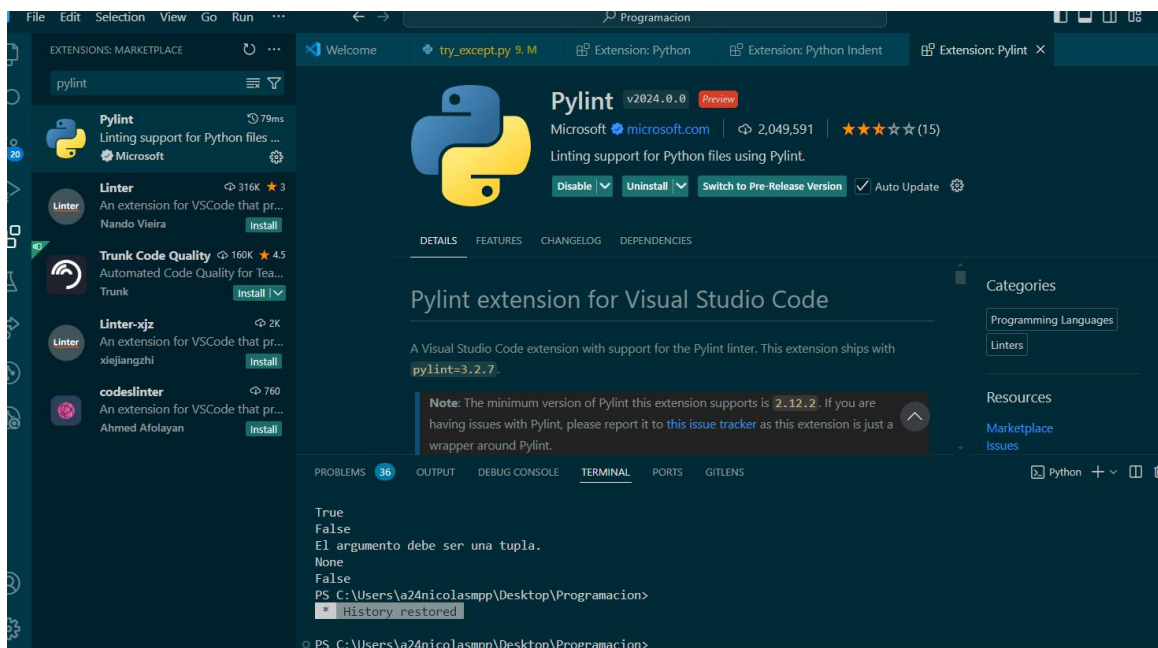
2.1. Instalación y configuración de extensiones en VS Code

Una vez instalado VS Code, investiga e instala las siguientes extensiones clave para trabajar con Python:

- **Python** (desarrollada por Microsoft): es la extensión principal que incluye soporte para ejecución, depuración y análisis de código Python. Investiga cómo realizar la configuración para que VS Code detecte el intérprete de Python en tu sistema.

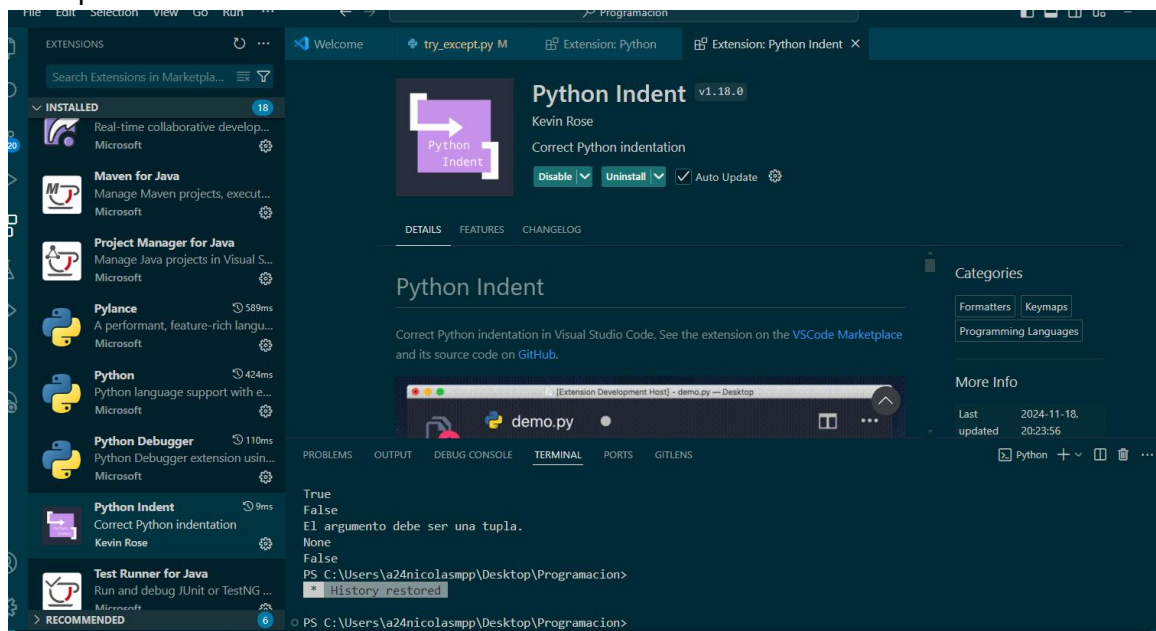


- **Pylint:** una herramienta de análisis estático de código para Python. Configura esta extensión para que VS Code realice la verificación automática de calidad de tu código.



*Nota: busca información adicional sobre el objetivo y las características de los **analizadores de código o linters**, y su funcionamiento integrados en los IDEs*

- **Python Indent:** mejora la experiencia de indentado, especialmente útil en Python por su sensibilidad a la indentación.



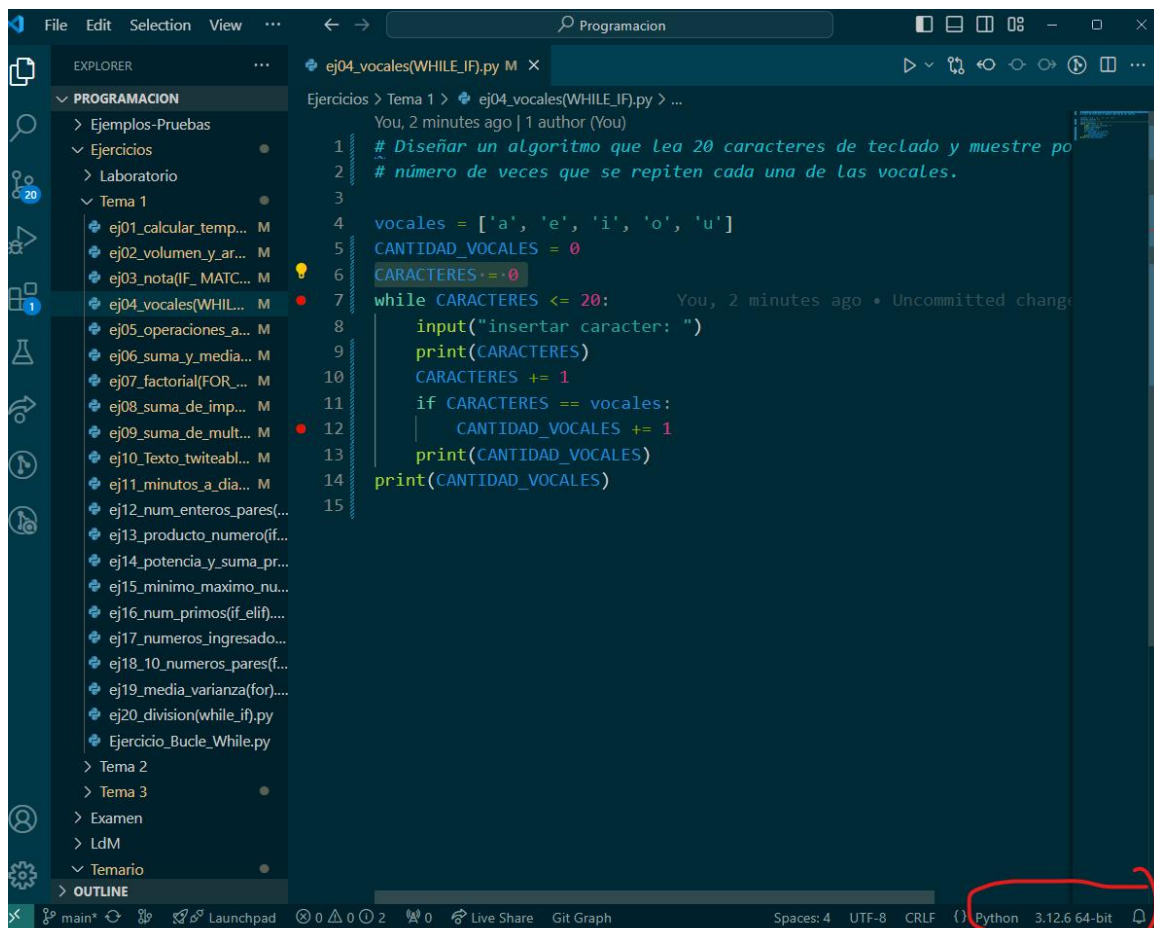
Debes indicar la utilidad de cada uno de los plugins, incluyendo una captura de código que demuestre dicha utilidad (por ejemplo, comparando el código antes y después de instalar dicho plugin; busca la forma que mejor demuestre dicha utilidad).

¿Conoces alguna otra extensión útil para programar en Python desde VSCode?

CodeSnap Saca screens del código que estas escribiendo, no es exclusivo de pyhton pero resulta util.

2.2. Configuración de entorno de trabajo

- **Configuración del intérprete de Python:** investiga cómo seleccionar el intérprete de Python adecuado en VS Code y asegúrate de que tu entorno de trabajo esté correctamente configurado.
- **Depuración:** explora cómo configurar el sistema de depuración en VS Code para poder poner puntos de interrupción (breakpoints) y depurar un script de Python.
-



2.3. Ejecución de código Python

Para hacer todas las pruebas relacionadas con Python, puedes hacer uso del código adjunto en la tarea (ed03_2.py). Dicho código contiene algunos errores que harán que no sea directamente ejecutable (recuerda que Python es un lenguaje interpretado, por lo que no hay un proceso de compilación como tal, y el código es traducido línea a línea en tiempo de ejecución).

Descarga el archivo, ábrelo en VS Code, y renómbralo desde el propio VS Code (busca cómo renombrar un archivo, así como el atajo para realizarlo sin utilizar el ratón). Debes llamar a tu archivo **ed03_2-apellidos_nombre.py**

F2

Con la ayuda de los plugins, deberás corregir los errores que tenga el código y ejecutarlo. Asegúrate de que todo el código esté correctamente anidados e indentados. Busca la forma de autoindentar todo el código (opción de VSCode y atajos del teclado).

SHIFT+ALT+F

Fijate bien en los subrayados que añade VSCode sobre el código, en algunos casos se corresponderá con errores (el código no se ejecutará con éxito) y en otro en problemas de estilo (aunque aparezcan los subrayados, el código se podrá ejecutar sin problema):

```

40         # por que aca no tira el error de %?
41
42     # Operaciones básicas
43
44     def sumar(self, a, b):
45         """Suma dos números"""
46         self.validar_numeros(a, b)
47         logger.info("Sumando {%d} + {%d}", a, b)
48         return a + b
49
50     def restar(self, a, b):
51         """Resta dos números"""
52         self.validar_numeros(a, b)
53         logger.info("Restando {%d} - {%d}", a, b)
54         return a - b
55
56     def multiplicar(self, a, b):
57         """Multiplica dos números"""
58         self.validar_numeros(a, b)
59         logger.info("Multiplicando {%d} * {%d}", a, b)
60         return a * b
61
62     def dividir(self, a, b):
63         """Divide dos números"""
64         self.validar_numeros(a, b)
65         if b == 0:
66             logger.error("Intento de división por cero")
67             raise ValueError("No se puede dividir por cero")
68         logger.info("Dividiendo {%d} / {%d}", a, b)
69         return a / b
70
71     # Operaciones avanzadas
72     def potencia(self, base, exponente):
73         """Calcula la potencia de un número"""
74         self.validar_numeros(base, exponente)
75         logger.info("Calculando {%d} ^ {%d}", base, exponente)
76         return math.pow(base, exponente)
77
78     def raiz_cuadrada(self, numero):
79         """Calcula la raíz cuadrada de un número"""
80         self.validar_numeros(numero)
81         if numero < 0:

```


Debes intentar que el código, con los plugins activados, no presente ningún tipo de subrayado.

A continuación, modifica el código provisto para que se pida al usuario los datos de entrada a través de la consola. Busca la información necesaria para realizar dichas modificaciones. Vuelve a ejecutar el código y confirma su correcto funcionamiento. Extrae todas las capturas de pantalla que creas convenientes para confirmar este correcto funcionamiento.

3. Edición de archivo HTML y CSS e instalación de plugins

Ahora vamos a trabajar con los lenguajes HTML y CSS, que sirven para definir la estructura y los estilos de una web estática.

3.1. Instalación y configuración de extensiones en VS Code

Investiga e instala las siguientes extensiones clave para trabajar con HTML y CSS:

- **Live Server:** lanza un servidor de desarrollo local con recarga en vivo para páginas estáticas y dinámicas.
- **CSS Peak:** permite navegar a los estilos CSS de un ID y una clase desde un archivo HTML.

Debes indicar la utilidad de cada uno de los plugins, incluyendo una captura de código que demuestre dicha utilidad.

3.2. Modificación del archivo HTML

Para hacer todas las pruebas relacionadas con HTML y CSS, puedes hacer uso del código adjunto en la tarea (ed03_2.html y ed03_2.css). Dicho código contiene algunos errores que harán que no sea directamente interpretable (HTML y CSS no son lenguajes de programación como tal, sino lenguajes de marcas y de estilos respectivamente, por lo que no podemos hablar de ejecución como tal).

Descarga los archivos, ábrelos en VS Code, y renómbralos desde el propio VS Code a **ed03_2-apellidos_nombre.html** y **ed03_2-apellidos_nombre.css**

Realiza las modificaciones necesarias y corrige **los errores sintácticos**:

- En el archivo **HTML**
 - Falta una etiqueta de cierre en el elemento `<meta charset="UTF-8">`.
 - Falta cerrar la lista de navegación `` y uno de los elementos de la lista ``.
 - Asegúrate de que todos los elementos HTML estén correctamente anidados e indentados. Busca la forma de autoindentar todo el código (opción de VSCode y atajos del teclado). **Shift + Alt + F**

```

<!DOCTYPE html>
<html>

<head>
  <title>Práctica HTML con VS Code</title>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="ed03_2.css">
</head>

<body>
  <header>
    <h1>Bienvenidos a la práctica de HTML</h1>
  </header>

  <nav>
    <ul>
      <li><a href="#">Inicio</a></li>
      <li><a href="#about">Sobre nosotros</a></li>
      <li><a href="#contacto">Contacto</a></li>
    </ul>
  </nav>

  <section id="about">
    <h2>Sobre nosotros</h2>
    <p>Esta es una página de prueba para practicar edición de HTML en Visual Studio Code.</p>
    
  </section>

  <section id="contacto">
    <h2>Contacto</h2>
    <form>
      <label for="name">Nombre</label>
      <input type="text" id="name" name="name">

      <label for="email">Email</label>
      <input type="email" id="email" name="email">

      <textarea name="mensaje" id="mensaje" cols="30" rows="5">Escribe tu mensaje aquí</textarea>

      <button type="submit">Enviar</button>
    </form>
  </section>

  <footer>
    <p>Entornos de Desarrollo. CFGS de Desarrollo de Aplicaciones Multiplataforma. Curso 2024/25.</p>
  </footer>
</body>

</html>

```

- Agrega una imagen real al documento HTML, utilizando una URL externa.

Por ejemplo:

`` (sustituye el link por el de una imagen real).

```

</ul>
</nav>

<section id="about">
  <h2>Sobre nosotros</h2>
  <p>Esta es una página de prueba para practicar edición de HTML en Visual Studio Code.</p>
  
</section>

<section id="contacto">
  <h2>Contacto</h2>
  <form>
    <label for="name">Nombre</label>
    <input type="text" id="name" name="name">

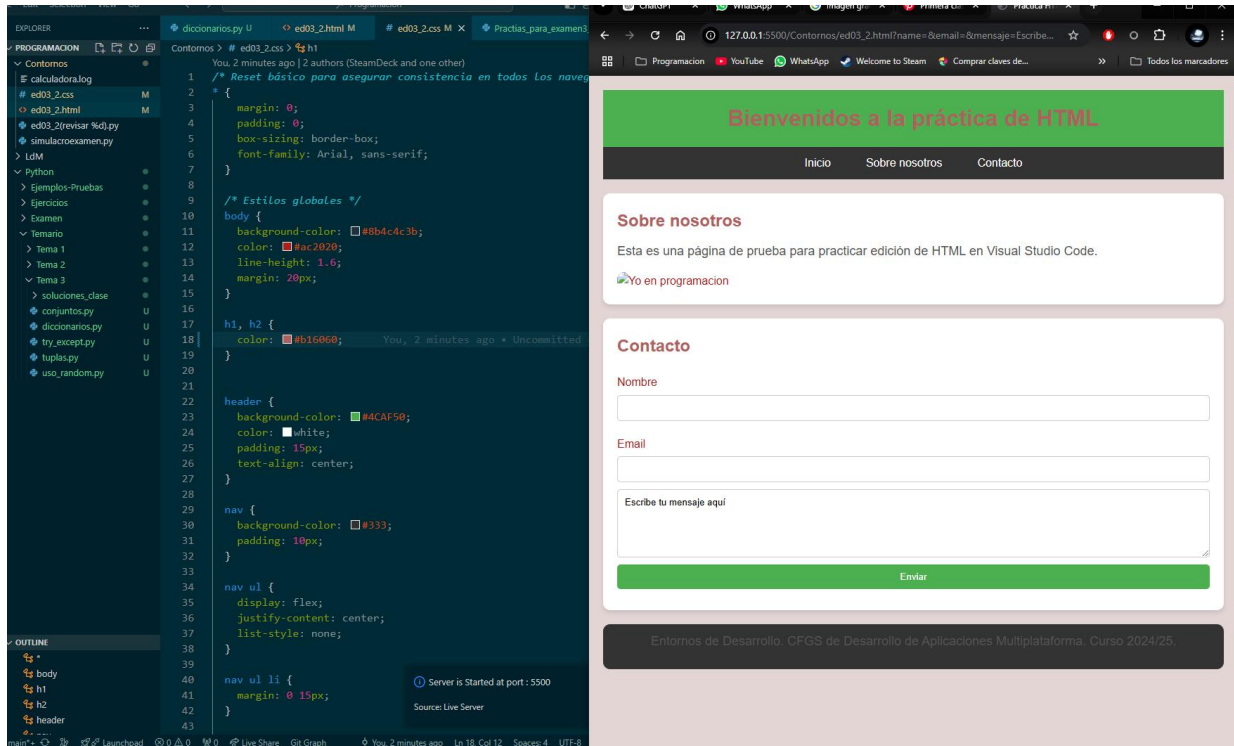
    <label for="email">Email</label>
    <input type="email" id="email" name="email">

    <textarea name="mensaje" id="mensaje" cols="30" rows="5">Escribe tu mensaje aquí</textarea>
  </form>
</section>

```

- Añade la referencia al archivo css en la sección head, de manera que se apliquen los estilos definidos en el archivo ed03_2.css:

```
<link rel="stylesheet" href="ed03_2.css">
```
- En el archivo **CSS**
 - Cambia el color del encabezado `<h1>` utilizando la funcionalidad de VSCode para seleccionar el color gráficamente. Ubica el cursor sobre el cuadrado del color y te aparecerá una ventana como esta:



3.3. Visualización en tiempo real:

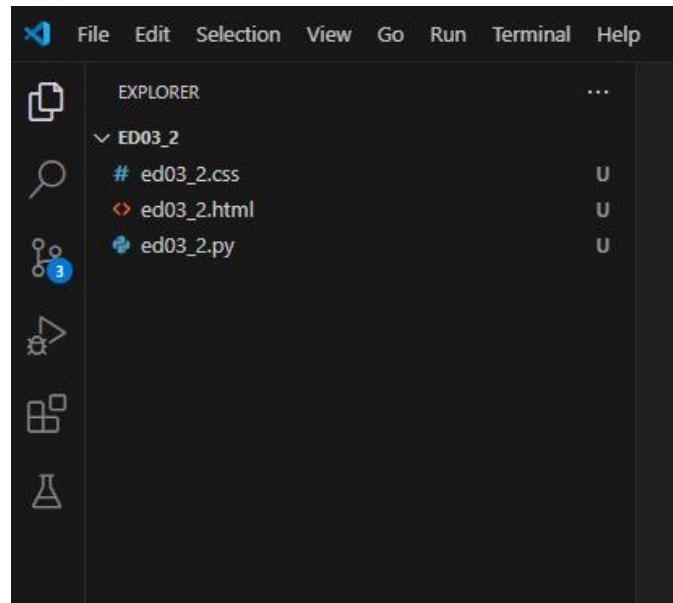
Utiliza la extensión **Live Server** para abrir el archivo HTML en tu navegador web y visualizar los cambios en tiempo real mientras editas el código en VS Code. Busca la información necesaria e indica cómo abrir el archivo HTML en el navegador.

4. Uso de Git desde VSCode

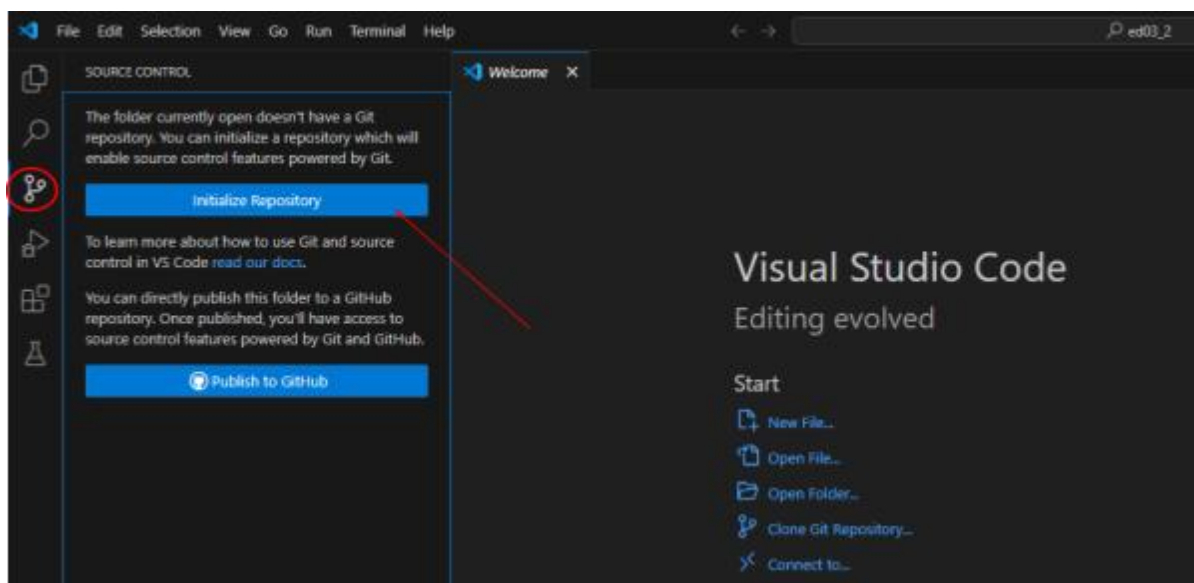
Una de las funcionalidades que proveen los IDEs, incluyendo VSCode, es la posibilidad de trabajar con Git y Github de manera interactiva y gráfica, sin tener que abrir una terminal de Git y escribir los comandos uno a uno.

4.1. Uso de herramientas nativas

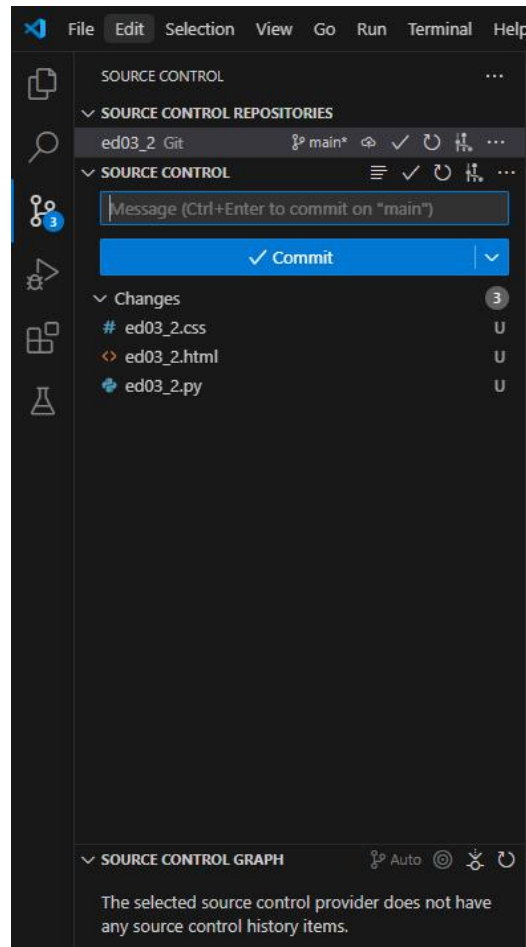
Vamos a crear un repositorio con los 3 archivos utilizados a lo largo de esta práctica. Por tanto, antes de nada, incluimos nuestros archivos en una carpeta llamada ed03_2.



Sin instalar ningún tipo de plugin, el IDE nos ofrece la capacidad de interactuar con un repositorio de Git

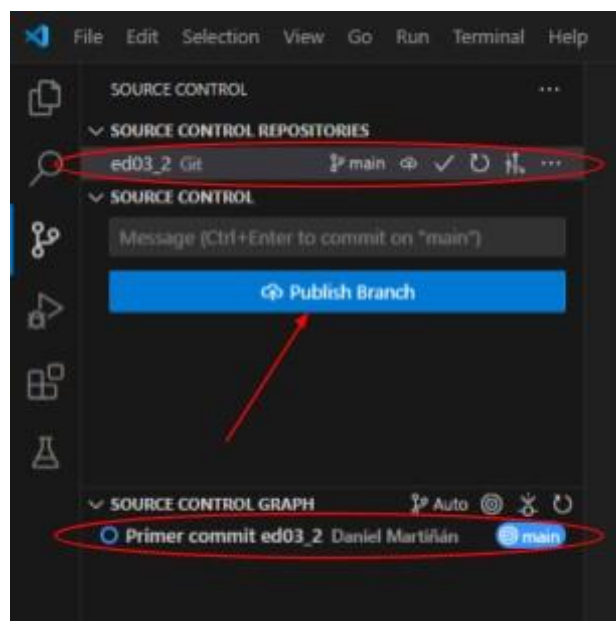


Si inicializamos un repositorio, nos aparecerá el siguiente menú, donde vemos los archivos con cambios (todos, ya que el repositorio acaba de ser creado) y la posibilidad de añadir los archivos y commitear:

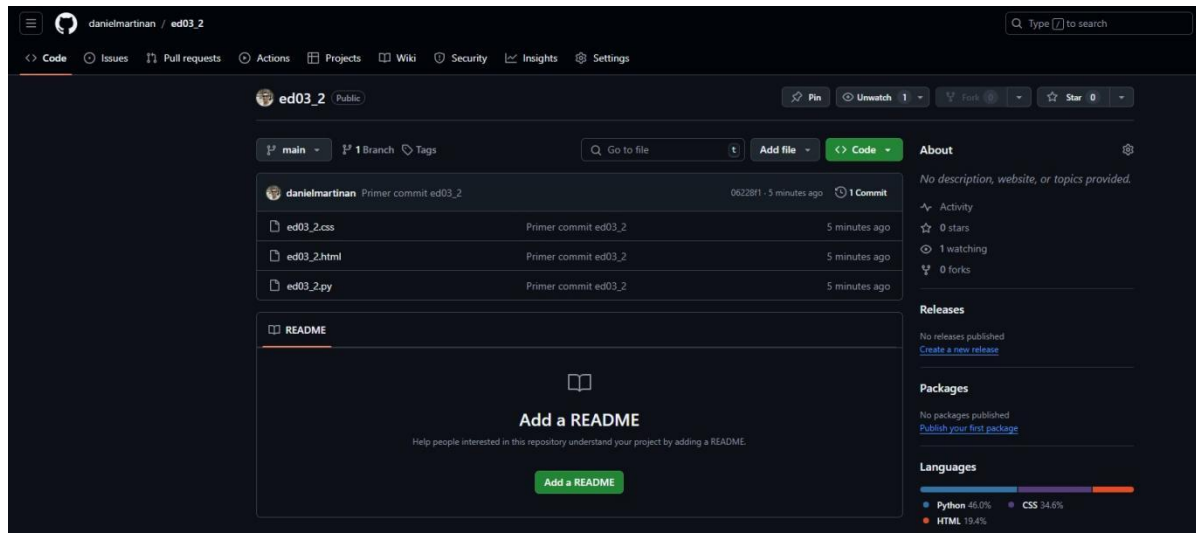


Debemos crear un commit con los 3 archivos y poner, como mensaje descriptivo “primer commit ed03_2”. **El commit lo hice en un repositorio que ya tenia, donde ya tenia todos los archivos y demas de la materia**

¿Qué comandos de git estaríamos aplicando con las acciones realizadas hasta el momento?



Ahora vemos que hay un commit, el que acabamos de crear, sobre la rama main. Vamos a publicar este commit en Github, haciendo click en “Publish Branch”. **Nos abrirá una ventana para que nos autentiquemos en Github. Completa el proceso.** Si todo ha ido bien, en tu github personal, se habrá creado un nuevo repositorio con el nombre que se haya asignado:



Asegúrate de que el repositorio creado sea público, para que sea accesible para su corrección.

4.2. Uso de plugins

Ahora debes instalar el plugin Git Graph. Explora las opciones que proporciona e indica las ventajas frente a las opciones nativas que ofrece VSCode.

5. Conclusiones

Tras realizar todas las tareas solicitadas, incluye en tu memoria un apartado de conclusiones donde indiques tus impresiones acerca del uso del IDE y de las extensiones:

- ¿Crees que el uso del IDE facilita la labor del desarrollo de software? Sí, **solo con los plugins facilita muchas cosas, ver errores, solucionar errores (indentación) y personalizar el IDE le da un toque de glam**
- ¿Crees que es ventajoso disponer de extensiones en los IDEs para extender su funcionalidad? **Si, de otra manera o el IDE seria incompleto o estaría repleto de funciones que probablemente no se utilizarían, al tener las extensiones se pueden instalar específicamente las que uno quiere o necesita.**
- ¿Crees que es intuitivo el uso de un IDE y de las extensiones? **Es fácil de utilizar pero no se si diría intuitivo, lleva un tiempo acomodarse a los atajos y a las ubicaciones de las funciones. No es algo que abriéndolo una vez ya se**

puede aprovechar al máximo.

- ¿Se te ocurren otras funcionalidades que pueda ofrecer un IDE y que puedas completar con la instalación de alguna extensión?

6. Entrega

Deberás entregar:

- un informe, en formato pdf, cuyo nombre sea **ed03_2-apellidos_nombre.pdf**, donde se incluyan:
 - Capturas de pantalla que los procesos de instalación realizados, si procede.
 - Capturas de la configuración de las extensiones en VS Code.

- Capturas de pantalla que evidencien el funcionamiento del depurador y de la ejecución de tu script en Python.
- Explicación de los problemas que hayas encontrado durante la instalación o configuración y cómo los has solucionado.
- Enlace al repositorio de Github creado.
- Carpeta que incluya:
 - El código fuente del script en Python con las modificaciones realizadas, renombrado como **ed03_2-apellidos_nombre.py**.
 - El código html y css con las modificaciones realizadas, renombrado como **ed03_2-apellidos_nombre.html** y **ed03_2-apellidos_nombre.css**.
 - Carpeta .git autogenerada al crear el repositorio a través de VScode

Todos estos archivos deberán estar comprimidos en un archivo .zip, .rar o .7z con el nombre ed03_2-apellidos_nombre, evitando como siempre tildes y eñes.