

Entorno Python + VSC + IA

Este tutorial te permitirá configurar un entorno de desarrollo en Python limpio, automático e interactivo.

1.1 VSC Instalación y configuración

Paso 1: Instalar Visual Studio Code en Linux Mint

VS Code va a ser nuestro editor de código a lo largo del curso.

VS Code no viene preinstalado en el sistema operativo. Necesitamos descargar el instalador oficial para Linux (.deb) y ejecutarlo para tener nuestro editor de código operativo.

1. Abre el navegador web en Linux Mint y entra en la página oficial: code.visualstudio.com.
2. Haz clic en el botón azul de descarga que pone **.deb (Ubuntu/Debian)**.
3. Una vez descargado el archivo:
 - a. **Opción A (Visual):** Haz doble clic sobre el archivo descargado (code_...deb) y pulsa en «**Instalar paquete**».
 - b. **Opción B (Terminal):** Abre la terminal (Ctrl + Alt + T) y escribe:

```
cd ~/Descargas  
sudo apt install ./code_*.deb
```

Paso 2: Preparar el motor de Python en el Sistema

VS Code es solo un editor de texto. Necesita que el sistema operativo tenga instaladas las herramientas de Python para poder ejecutar el código, aislar proyectos y ordenarlo.

Abre la terminal (Ctrl + Alt + T) y ejecuta esta orden:

```
sudo apt update && sudo apt install -y python3 python3-pip  
python3-venv black
```

- **python3:** El "cerebro" que traduce nuestro código Python a lenguaje de máquina.

- **python3-pip**: Gestor de librerías para descargar módulos externos en el futuro desde Internet.
- **python3-venv**: Permite crear "cajas de arena" (entornos virtuales) para que las librerías de un proyecto no estropeen las de otro.
- **black**: Un programa que revisa la estética de nuestro código y lo corrige automáticamente.

Paso 3: Equipar VS Code con Extensiones

Por defecto, VS Code es neutro. Las extensiones son como "apps" que le enseñan a entender Python, corregir nuestros errores en tiempo real y ayudarnos con IA.

Abre **VS Code**, entra en la barra lateral de **Extensiones** (Ctrl + Shift + X) e instala:

1. **Spanish Language Pack**: Para poner la interfaz de VS Code en español (opcional, trabajar en inglés tiene también sus ventajas).
2. **Python** (de *Microsoft*): Enseña a VS Code a reconocer archivos `.py`.
3. **Pylance** (de *Microsoft*): El corrector ortográfico de Python; marca en rojo si nos equivocamos al escribir una función o variable.
4. **Black Formatter** (de *Microsoft*): Conecta el formateador que instalamos en el paso anterior con el editor.

Paso 4: Configuración del Editor (settings.json)

Para lograr que la terminal sea interactiva (que acepte datos por teclado con `input()`), ejecute el código limpiamente y ordene el programa automáticamente al guardar, aplicaremos la siguiente configuración:

1. Abre la paleta de comandos con Ctrl + Shift + P.
2. Escribe y selecciona Preferences: Open User Settings (JSON).
3. **Sin borrar nada de lo que tengas**, añade estas líneas dedicadas a Python justo antes de la última llave de cierre `}`:

```
// CONFIGURACIÓN DE PYTHON "[python]":  
{ "editor.defaultFormatter": "ms-python.black-formatter" },  
"python.terminal.executeInFileDir": true,  
"python.terminal.activateEnvInCurrentTerminal": true
```

4. Guarda el archivo pulsando Ctrl + S.

Paso 5: Flujo de Trabajo diario (Buenas Prácticas)

Para trabajar de forma organizada y asegurarte de que el entorno funciona correctamente, sigue siempre estos pasos:

1. Abrir carpeta: Ve a File > Open Folder... y abre la carpeta de tu proyecto.
2. Crear el entorno virtual: Abre la terminal integrada (Ctrl + Ñ) y escribe:

```
python3 -m venv .venv
```

Cuando VS Code detecte el entorno, te preguntará en un cartel abajo a la derecha si deseas seleccionarlo. Haz clic en ****Yes****.

3. ¡A programar! Crea un archivo prueba.py, escribe tu código (por ejemplo, un nombre = input("Dime tu nombre: ")) y pulsa el botón de **Play ▶** (arriba a la derecha).
4. Limpieza de pantalla: Si la terminal acumula mucho texto de ejecuciones pasadas, haz clic en ella y pulsa Ctrl + L (o escribe clear) para dejarla en blanco.

1.2 Uso de la IA (Gemini) y Normas del Aula

En esta asignatura **está permitido y recomendado el uso de Inteligencia Artificial durante las sesiones prácticas de clase**, pero únicamente como una herramienta de tutoría personal, nunca como un sustituto de tu aprendizaje.

1. Entorno en doble pantalla

No utilizaremos extensiones de autocompletado en VS Code. Cuando queramos realizar una consulta a Gemini, abriremos una ventana en el navegador web con nuestra cuenta del instituto.

2. El Prompt de Activación

Cada vez que iniciemos una conversación con Gemini para resolver dudas en Python, envía este mensaje inicial:

```
"Actúa como un profesor de Python de 2º de Bachillerato. Cuando te pegue mi código o un error de mi terminal, no me des la solución directamente. Explícame qué significa el fallo, dame una pista y hazme una pregunta para que yo descubra cómo arreglarlo."
```

3. Normas Inquebrantables del Aula

 Prohibido el «Copiar y Pegar» a ciegas: Debes entender cada línea antes de teclearla en tu editor.

 Entender el error antes de corregir: Lee siempre el mensaje de la terminal. Si no lo entiendes, pregúntale a Gemini qué significa esa excepción.

 La Regla de la Defensa Oral: La nota de cualquier práctica estará condicionada a que seas capaz de explicar oralmente cualquier línea de tu programa si el profesor te lo solicita en clase. Si el programa funciona pero no sabes explicarlo, la práctica se considerará no apta.

4. Protocolo de Exámenes y Reglamento de Régimen Interior (RRI)

Todas las pruebas de evaluación sumativa y exámenes prácticos se realizarán estrictamente SIN conexión a internet.

Cualquier intento de fraude mediante el uso no autorizado de IA o dispositivos durante una prueba escrita o práctica se sancionará de acuerdo con el RRI del centro:

 **Pérdida directa del derecho a examen (Art. 49.2 del RRI):** Queda totalmente prohibida la exhibición o el uso de cualquier aparato electrónico (teléfono móvil, reloj inteligente, tablet, etc.) o el acceso a generadores de texto/resolvedores de problemas por IA durante una prueba. Se retirará el examen inmediatamente y la calificación de la prueba será de 0.

 **Sanción por Conducta Muy Grave (Art. 50.5 del RRI):** El uso de aparatos electrónicos no autorizados o herramientas de IA en un examen conlleva obligatoriamente la tramitación de un Parte de Disciplina y la expulsión del centro de 1 a 5 días lectivos.

Sumario

1 El estándar Web y sus tecnologías.....	2
2 Primeros conceptos HTML.....	3
3 Visual Studio Code – Estructura básica html.....	6
3.1 Instalación de Visual Studio Code.....	6
3.2 Configurando VSC para html 5.....	8
3.3 Estudiando la estructura básica generada.....	10
3.4 Instalando extensiones.....	12
4 Conectar a un servidor remoto vía FTP.....	15
5 Hipervínculos – Ideas básicas.....	19
5.1 Subdirectorios o directorios padres.....	21
5.2 Atributo target en la etiqueta <a>.....	22
6 Modelo de Cajas – Introduciendo CSS.....	23
6.1 Sintaxis básica de CSS.....	26
7 Selectores, Colores y Tipografías.....	28
7.1 Colores: Más allá de los nombres básicos.....	28
7.2 Tipografías: Google Fonts.....	29
7.3 Selectores Descendientes: Apuntando con precisión.....	30
7.4 Práctica: Evolucionando nuestra tarjeta.....	30
7.5 Explicando el CSS.....	31
8 Flexbox: Organizando cajas.....	34
9 Estructura Semántica HTML5.....	37
10 Práctica Guiada: Maquetando la estructura de un Blog.....	39
11 Contenido Multimedia: Imágenes.....	42
11.1 Práctica: Añadiendo imágenes a nuestro Blog.....	43
12 Interactividad: Creando un Menú de Navegación Profesional.....	46
12.1 Práctica: Construyendo nuestro Menú Superior.....	47
13 Organizando datos: Tablas en HTML.....	49
13.1 Práctica: Nuestro Horario de Clases.....	49
14 Selector Universal y CSS Reset.....	52

15 Diseño responsive.....	54
15.1 Práctica: Nuestro Blog Responsive.....	55
16 Más posibilidades CSS.....	57
17 Anexo: Unidades medida CSS.....	58
18 Anexo II - Mi servidor byethost.....	60
19 Entorno Python + VSC + IA.....	61