

Configuración de los entornos de desarrollo: Visual Studio Code

En este apartado, vamos a preparar **Visual Studio Code** para que sea un potente editor de código y, especialmente, para que nos sirva como una herramienta eficaz para generar documentación en **Markdown**, para programar en **Javascript**, **Typescript** y **Python**.

Actividad

A medida que sigas este tutorial, realiza una memoria de todos los pasos, con capturas de pantalla, de cada una de las configuraciones y procesos que se explican a continuación.

- Configuración de los entornos de desarrollo: Visual Studio Code
 - Visual Studio Code. Instalación y configuración
 - 1. Instalación de Visual Studio Code
 - 2. Primeros ajustes y bienvenida
 - 3. Configuración de las extensiones para Markdown
 - 4. Configuración para Javascript en Visual Studio Code
 - 4.1. Uso de la consola del navegador (Chrome)
 - 4.2. Configuración y uso de Node.js
 - 4.3. Configuración y uso de Deno (admite Typescript de forma nativa)
 - 5. Python en VS Code
 - 5.1. Instalación de Python 3
 - 5.2. Extensiones interesantes de VS Code relacionadas con Python
 - 5.3. Seleccionar el Intérprete de Python en VS Code
 - Probando el mismo código en diferentes IDE y diferentes códigos en el mismo editor

Visual Studio Code. Instalación y configuración

Lo primero es tener Visual Studio Code instalado en tu sistema.

1. Instalación de Visual Studio Code

1. **Descarga el instalador:** Abre tu navegador web y ve a la página oficial de Visual Studio Code: <https://code.visualstudio.com/>.
2. **Selecciona tu sistema operativo:** El sitio web detectará automáticamente tu sistema operativo (Windows, macOS, Linux). Haz clic en el botón de descarga correspondiente.

3. **Ejecuta el instalador:** Una vez descargado, abre el archivo.

- **En Windows:** Sigue las instrucciones del asistente de instalación. Te recomiendo marcar la opción "**Agregar al PATH**" durante la instalación, ya que esto te permitirá abrir VS Code desde la línea de comandos en cualquier directorio.
- **En macOS:** Arrastra el icono de la aplicación a tu carpeta de "Aplicaciones".
- **En Linux:** Dependiendo de tu distribución, puedes usar el archivo `.deb` (Debian/Ubuntu) o `.rpm` (Fedora/RHEL), o seguir las instrucciones para tu gestor de paquetes.

4. **Inicia Visual Studio Code:** Una vez finalizada la instalación, abre la aplicación.

2. Primeros ajustes y bienvenida

Al abrir VS Code por primera vez, verás una pantalla de bienvenida. Puedes explorarla o cerrarla.

- **Idioma:** Si VS Code no está en tu idioma preferido, puedes instalar un **Language Pack**.
 - i. Ve a la vista de **Extensiones** (icono de los cuadrados en la barra lateral izquierda, o atajo `Ctrl+Shift+X`).
 - ii. Busca "Spanish Language Pack" e instálalo.
 - iii. Reinicia VS Code cuando te lo pida.
- **Tema:** Puedes cambiar el tema visual de VS Code desde Archivo > Preferencias > Tema de color (o Code > Preferencias > Tema de color en macOS). Experimenta con los temas "Dark Modern", "Light Modern", o descarga otros temas desde el Marketplace de extensiones.

Sin embargo, es recomendable que trabajes con VS Code en inglés, ya que la mayoría de la documentación está en ese idioma.

3. Configuración de las extensiones para Markdown

Las extensiones son el corazón de la personalización y funcionalidad de VS Code. Para instalarlas, usa la vista de **Extensiones** y busca la que necesites.

Si realizaste la actividad de ampliación de la unidad de programación 2, probablemente ya tengas preparado Visual Studio Code para usar Markdown. Si no, sigue estos pasos:

- **Para la documentación en Markdown:**
 - i. **Markdown All in One:**
 - En la barra de búsqueda de extensiones, escribe `Markdown All in One`.
 - Busca la extensión publicada por "Yuya Saito" y haz clic en **Instalar**.

- Esta extensión te dará atajos de teclado para formato, una tabla de contenidos automática, y previsualización mejorada.

ii. **Markdown Preview Enhanced:**

- En la barra de búsqueda de extensiones, escribe **Markdown Preview Enhanced**.
- Busca la extensión publicada por "WangKou" y haz clic en **Instalar**.
- Ofrece una previsualización aún más rica, con soporte para diagramas, matemáticas y más. Para activarla en un archivo Markdown, haz clic derecho y selecciona **"Open Preview"** o el icono de previsualización en la esquina superior derecha.

4. Configuración para Javascript en Visual Studio Code

VS Code tiene un excelente soporte para JavaScript. Aquí vemos cómo complementarlo.

4.1. Uso de la consola del navegador (Chrome)

Aunque esto no es una configuración de VS Code per se, la consola del navegador es indispensable para el desarrollo frontend con JavaScript.

1. **Abre el navegador Chrome.**

2. **Abre las Herramientas de Desarrollador:**

- Haz clic derecho en cualquier parte de la página web y selecciona **"Inspeccionar"**.
- O usa el atajo de teclado: `Ctrl+Shift+I` (Windows/Linux) o `Cmd+Option+I` (macOS).

3. **Navega a la pestaña "Console":** Aquí es donde verás los errores de JavaScript, los mensajes de `console.log()`, y podrás ejecutar código JavaScript directamente.

4. **Prueba un comando:** Escribe `console.log("Hola desde la consola!");` y presiona `Enter`. Verás el mensaje impreso.

4.2. Configuración y uso de Node.js

Node.js te permite ejecutar JavaScript fuera del navegador, en el servidor o para tareas de desarrollo.

1. **Instalación de Node.js:**

- Ve a la página oficial de Node.js: <https://nodejs.org/>.
- Descarga la **versión LTS (Long Term Support)**, que es la más estable y recomendada.
- Ejecuta el instalador y sigue los pasos. Asegúrate de que las opciones de instalación de **npm (Node Package Manager)** y **agregar Node.js al PATH** estén seleccionadas.

2. **Verifica la instalación:**

- Abre el **Terminal integrado de VS Code** (`Ctrl+Ñ` o `View > Terminal`).
- Escribe `node -v` y presiona `Enter`. Debería mostrar la versión de Node.js instalada.
- Escribe `npm -v` y presiona `Enter`. Debería mostrar la versión de npm instalada.

3. Extensiones de VS Code para Node.js:

- **ESLint:** Para ayudarte a escribir código JavaScript sin errores y con un estilo consistente.
 - Instala la extensión ESLint (publicada por "Dirk Baeumer").
 - A menudo, necesitarás configurar un archivo `.eslintrc.js` en tu proyecto para definir las reglas.
- **Prettier - Code formatter:** Formatea tu código automáticamente al guardarlo.
 - Instala la extensión Prettier - Code formatter (publicada por "Prettier").
 - Para que formatee al guardar, ve a Archivo > Preferencias > Configuración (o Code > Preferencias > Configuración en macOS), busca `format on save` y marca la casilla.

4. Ejecutar un archivo Node.js en VS Code:

- Crea un archivo llamado `app.js` con el siguiente contenido:

```
console.log("¡Hola desde Node.js!");
```

- En el **Terminal integrado de VS Code**, navega hasta el directorio donde guardaste `app.js` (p.ej., `cd mi_proyecto`).
- Ejecuta el archivo con `node app.js`. Verás la salida en el terminal.

4.3. Configuración y uso de Deno (admite Typescript de forma nativa)

Deno es una alternativa moderna a Node.js, con soporte nativo para TypeScript.

1. Instalación de Deno:

- Abre tu terminal (o la línea de comandos).
- Sigue las instrucciones de instalación de la web oficial de Deno:
<https://deno.land/#installation>. Sigue el [getting started](#).

2. Verifica la instalación:

- En tu terminal (o el Terminal integrado de VS Code), escribe `deno --version`. Debería mostrar la versión de Deno.

3. Extensión de VS Code para Deno:

- Ve a la vista de **Extensiones** en VS Code.
- Busca `Deno` y instala la extensión oficial (publicada por "denoland").

4. Habilitar Deno para tu proyecto en VS Code:

- Abre la **Paleta de Comandos** (`Ctrl+Shift+P` o `Cmd+Shift+P`).
- Escribe `Deno: Initialize Workspace` y selecciona la opción. Esto creará un archivo `.vscode/settings.json` o actualizará el existente en tu proyecto, configurando VS Code para usar Deno.
- Verifica que el archivo `settings.json` contenga algo similar a:

```

{
  "deno.enable": true,
  "deno.lint": true,
  "deno.unstable": true,
  "[typescript]": {
    "editor.defaultFormatter": "denoland.vscode-deno"
  },
  "[typescriptreact]": {
    "editor.defaultFormatter": "denoland.vscode-deno"
  },
  "[javascript]": {
    "editor.defaultFormatter": "denoland.vscode-deno"
  },
  "[javascriptreact]": {
    "editor.defaultFormatter": "denoland.vscode-deno"
  }
}

```

5. Ejecutar un archivo Deno (TypeScript) en VS Code:

- Crea un archivo llamado `main.ts` con el siguiente contenido (TypeScript):

```

function greet(name: string): void {
  console.log(`Hola, ${name} desde Deno!`);
}

greet("Mundo");

```

- En el **Terminal integrado de VS Code**, navega hasta el directorio del archivo.
- Ejecuta el archivo con `deno run main.ts`. Si es la primera vez, Deno puede pedirte permiso para acceder a la red o al sistema de archivos (si tu código lo requiere).

5. Python en VS Code

VS Code es un entorno extremadamente popular y potente para el desarrollo en Python.

5.1. Instalación de Python 3

Para trabajar con Python, primero debes tenerlo instalado en tu sistema.

1. Descarga Python 3:

- Ve a la página oficial de descargas de Python: <https://www.python.org/downloads/>.
- Descarga la última versión estable de **Python 3**.

- **Ejecuta el instalador.**
 - **MUY IMPORTANTE (Windows):** Durante la instalación, asegúrate de marcar la casilla **"Add Python to PATH"** (Agregar Python al PATH). Esto es fundamental para que puedas ejecutar Python desde cualquier ubicación en tu terminal.
 - Sigue los pasos restantes del instalador.

2. Verifica la instalación:

- Abre el **Terminal integrado de VS Code** (`Ctrl+Ñ`).
- Escribe `python --version` y presiona `Enter` . Debería mostrar la versión de Python 3 que acabas de instalar (p.ej., `Python 3.10.x`).
- En algunos sistemas, puede que necesites usar `python3 --version` .

5.2. Extensiones interesantes de VS Code relacionadas con Python

Estas extensiones transformarán tu experiencia de desarrollo Python en VS Code.

1. **Python (Microsoft):** Esta es la extensión fundamental y obligatoria.

- Ve a la vista de **Extensiones** (`Ctrl+Shift+X`).
- Busca `Python` y busca la extensión publicada por **"Microsoft"**. Haz clic en **Instalar**.
- Esta extensión proporciona IntelliSense, depuración, formateo, linting, soporte para entornos virtuales, y mucho más.

2. **Pylance (Microsoft):** Complementa y mejora enormemente la extensión Python.

- Busca `Pylance` y haz clic en **Instalar**.
- Ofrece un análisis de código estático más rápido, sugerencias de autocompletado más precisas y una mejor experiencia con el tipado de Python.

3. **Jupyter:** Si trabajas con ciencia de datos, machine learning o notebooks, esta extensión es crucial.

- Busca `Jupyter` y haz clic en **Instalar**.
- Te permitirá abrir y ejecutar archivos `.ipynb` (Jupyter Notebooks) directamente en VS Code.

4. **Black Formatter:** Un formateador de código Python muy popular que aplica un estilo consistente automáticamente.

- Busca `Black Formatter` y haz clic en **Instalar**.
- Para usarlo, necesitas instalar `black` en tu entorno Python: `pip install black` .
- Luego, en VS Code, puedes configurarlo como tu formateador predeterminado en `Archivo > Preferencias > Configuración` y buscando `python.formatting.provider` y seleccionando `black` . También puedes habilitar `Editor: Format On Save` .

5.3. Seleccionar el Intérprete de Python en VS Code

Es importante que VS Code sepa qué versión de Python (o entorno virtual) debe usar para tu proyecto.

1. **Abre la Paleta de Comandos:** `Ctrl+Shift+P` (Windows/Linux) o `Cmd+Shift+P` (macOS).
2. **Escribe Python: Select Interpreter** y selecciona la opción.
3. **Elige tu intérprete:** Verás una lista de los intérpretes de Python que VS Code ha detectado en tu sistema. Selecciona la versión de Python 3 que instalaste.
 - Si tienes varios proyectos o entornos virtuales, este paso es crucial para asegurar que VS Code use el intérprete correcto para cada uno.

Probando el mismo código en diferentes IDE y diferentes códigos en el mismo editor

Actividad

Entra en la página web mycompiler.io. Se trata de un emulador virtual de entornos de programación. Selecciona un lenguaje, por ejemplo Python. Aparecerá el editor con un código para el programa `Hola Mundo`. Dale a ejecutar. Prueba ese mismo código en:

- pycharm (instálalo a través de JetBrains Toolbox, con la licencia educativa que conseguiste en el apartado anterior)
- vscode usando python3

Haz lo mismo con otros lenguajes de programación, pruébalos en diversos editores.

Actividad

Usando VS Code y las extensiones correspondientes, ejecuta un `Hola Mundo` en diferentes lenguajes (python, typescript y javascript, por ejemplo). Extrae el código de mycompiler.io.