

En esta primera semana, se te presentarán los siguientes desafíos pre curso, para comenzar a codear de manera independiente.

El objetivo es que, en esta primera semana de curso, prepares el entorno de trabajo para iniciar la segunda semana de cursado sincrónico con el entorno listo. Si en el proceso se te dificulta avanzar, los profes te ayudarán en la primera clase. ¡Adelante!

### Desafío 1. Instalación de Visual Studio Code

Para instalar Visual Studio Code, sigue estos pasos:

1. Ir al sitio web oficial de Visual Studio Code en <https://code.visualstudio.com/>.
2. Haz clic en el botón de descarga para el sistema operativo que estés utilizando (Windows, macOS o Linux).
3. Una vez descargado el archivo de instalación, ábrelo para iniciar el proceso de instalación.
4. Sigue las instrucciones del instalador. Por lo general, simplemente tendrás que hacer clic en "Siguiente" o "Aceptar" en cada paso.
5. Durante la instalación, puedes elegir las opciones adicionales que deseas incluir, como la creación de accesos directos en el escritorio o el agregado de Visual Studio Code al menú contextual.
6. Espera a que se complete la instalación. Esto puede llevar algunos minutos.
7. Una vez finalizada la instalación, ejecuta Visual Studio Code desde el menú de inicio (en Windows) o desde la carpeta de aplicaciones (en macOS).
8. Visual Studio Code se abrirá con una interfaz de usuario. ¡Ahora estás listo para empezar a usarlo!

Recuerda que Visual Studio Code es altamente personalizable y admite extensiones para agregar funcionalidad adicional. Puedes explorar las extensiones disponibles desde la barra lateral izquierda en Visual Studio Code, haciendo clic en el ícono de extensiones (cuatro cuadros superpuestos).

### Desafío 2. Reconocimiento de entorno Visual Studio Code

La configuración personalizada de Visual Studio Code puede ayudarte a adaptar el editor a tu forma de trabajar y mejorar tu productividad. Aquí hay algunos pasos que puedes seguir para personalizar VS Code según tus preferencias:

- *Tema y apariencia:*

Para cambiar el tema, ve a la barra lateral izquierda y haz clic en el ícono de "Extensiones" (cuatro cuadros superpuestos) y busca "Tema". Instala el tema de tu elección y selecciona el nuevo tema en la barra de estado inferior.

Puedes ajustar la fuente, el tamaño y otros aspectos de la apariencia y la interfaz en la configuración de VS Code. Abre el menú "Archivo" en la barra de menú superior y selecciona "Preferencias" y luego "Configuración". Aquí puedes personalizar varios aspectos de la apariencia y la interfaz de usuario.

- *Atajos de teclado:*

-Visual Studio Code ofrece una amplia gama de atajos de teclado, pero también puedes personalizarlos según tus preferencias. Abre la configuración de VS Code como se mencionó anteriormente y busca "Atajos de teclado". Aquí puedes modificar o agregar atajos de teclado para comandos específicos. Archivo > Preferencias > Métodos abreviados de teclado.

- *Extensiones:*

Para instalar la extensión Live Server en Visual Studio Code, sigue estos pasos:

1. Abre Visual Studio Code en tu computadora.
2. Haz clic en el icono de extensiones en la barra lateral izquierda (o presiona Ctrl+Shift+X en Windows/Linux o Cmd+Shift+X en macOS) para abrir la vista de extensiones.
3. En el campo de búsqueda, escribe "Live Server".
4. Entre los resultados de búsqueda, deberías ver la extensión "Live Server" creada por "Ritwick Dey". Haz clic en "Instalar" para comenzar la instalación.
5. Después de instalar, abre el archivo HTML que deseas visualizar en el servidor en Visual Studio Code, haz clic con el botón derecho del ratón en el editor y selecciona "Abrir con Live Server" en el menú contextual. También puedes abrir el comando de paleta (presionando Ctrl+Shift+P en Windows/Linux o Cmd+Shift+P en macOS) y buscar "Live Server: Open with Live Server".
6. Se abrirá una nueva ventana del navegador con tu página web cargada y Live Server se encargará de actualizar automáticamente la página cuando realices cambios en el archivo.

-Explora el mercado de extensiones de Visual Studio Code y encuentra extensiones que se ajusten a tus necesidades. Puedes instalar y administrar extensiones desde la barra lateral izquierda en VS Code.

-Algunas extensiones populares incluyen linters para verificar la calidad del código, extensiones para lenguajes de programación específicos, herramientas de depuración adicionales, formateadores de código y mucho más. Instala las extensiones que te ayudarán en tu flujo de trabajo diario.

- *Configuración del editor:*

-Puedes personalizar varios aspectos del editor de texto en la configuración de VS Code. Algunas opciones útiles incluyen el tamaño de tabulación, las reglas de sangría, el ajuste de línea, el resaltado de sintaxis, la sugerencia de código, entre otros.

-Explora la configuración del editor en la sección "Configuración" y realiza ajustes según tus preferencias.

- *Snippets personalizados:*

-Los snippets son fragmentos de código reutilizables que puedes insertar rápidamente mientras escribes. VS Code proporciona muchos snippets predefinidos, pero también puedes crear tus propios snippets personalizados. Para ello, ir a la configuración de VS Code y busca "Snippets". Aquí puedes agregar tus snippets personalizados para diferentes lenguajes de programación.

Recuerda guardar tus preferencias de configuración en un archivo "settings.json" para poder mantenerlas incluso después de actualizar o cambiar de dispositivo.

Estos son solo algunos ejemplos de cómo puedes personalizar Visual Studio Code. Experimenta con diferentes opciones y ajustes para encontrar la configuración que se adapte mejor a tu forma de trabajar y te ayude a ser más eficiente en tu flujo de desarrollo. En <https://code.visualstudio.com/docs> encontrarás documentación de lo antes descripto.

### Desafío 3. Instalar Git

Para instalar Git localmente en tu computadora, sigue estos pasos:

#### 1. Descarga el instalador:

Ir al sitio web oficial de Git en <https://git-scm.com/downloads>.

Dependiendo de tu sistema operativo, se mostrará automáticamente el instalador correspondiente. Haz clic en el enlace de descarga para descargar el instalador.

#### 2. Ejecuta el instalador:

Una vez que se complete la descarga, ejecuta el archivo de instalación de Git.

Si estás en Windows, es posible que debas confirmar permisos de administrador para ejecutar el instalador.

#### 3. Configura las opciones de instalación:

Durante el proceso de instalación, se te presentarán varias opciones. Asegúrate de leer y seleccionar las opciones que mejor se adapten a tus necesidades. Por lo general, las opciones predeterminadas son suficientes para la mayoría de los usuarios.

#### 4. Selecciona la ubicación de instalación:

El instalador te pedirá que selecciones la ubicación donde deseas instalar Git. Puedes optar por utilizar la ubicación predeterminada o elegir otra carpeta si lo deseas.

#### 5. Selecciona los componentes a instalar:

En este paso, puedes elegir qué componentes de Git deseas instalar. Por lo general, es recomendable dejar todas las opciones seleccionadas para obtener la funcionalidad completa de Git.

#### 6. Ajusta las variables de entorno (solo Windows):

Si estás instalando Git en Windows, se te pedirá que selecciones el editor de texto predeterminado y el terminal que deseas usar con Git. Puedes elegir el editor y el terminal de tu preferencia.

#### 7. Completa la instalación:

Haz clic en el botón "Instalar" para comenzar el proceso de instalación. Espera a que se complete la instalación.

#### 8. Verifica la instalación:

Una vez que la instalación haya finalizado, puedes verificar si Git se instaló correctamente abriendo una terminal (Command Prompt en Windows o Terminal en macOS/Linux) y ejecutando el comando **git --version**. Debería mostrarte la versión de Git instalada.

### Desafío 4. Creación de espacio en GitHub

Para crear un repositorio en GitHub y así tener un espacio para almacenar y gestionar tus proyectos, sigue los siguientes pasos:

Inicia sesión en tu cuenta de GitHub. Si no tienes una cuenta, crea una en <https://github.com/join>.

1. Una vez que hayas iniciado sesión, haz clic en el botón "New" (Nuevo) en la esquina superior derecha de la pantalla o ir a la página principal de tu perfil y haz clic en el botón "New repository" (Nuevo repositorio) en la parte superior.
2. En la página "Create a new repository" (Crear un nuevo repositorio), completa los siguientes campos:
3. Repository name (Nombre del repositorio): Ingresa un nombre descriptivo para tu repositorio.
4. Description (Descripción): Opcionalmente, proporciona una breve descripción de tu proyecto.
5. Public or private: Puedes elegir si deseas que tu repositorio sea público (visible para todos) o privado (visible solo para ti o colaboradores seleccionados).
6. Initialize this repository with a README: Si marcas esta opción, se creará automáticamente un archivo README.md en tu repositorio. Puedes optar por crearlo más tarde si lo prefieres.
7. Puedes agregar más opciones y personalizar la configuración de tu repositorio según tus necesidades, como agregar un archivo .gitignore, una licencia, etc.
8. Haz clic en el botón "Create repository" (Crear repositorio) para crear el repositorio.

¡Listo! Ahora tienes un espacio en GitHub donde puedes cargar, gestionar y colaborar en tus proyectos.

### Desafío 5. Probando mi entorno

Aquí tienes las instrucciones para crear tu primer proyecto HTML en Visual Studio Code:

#### 1. *Instalar Visual Studio Code*

Descarga e instala Visual Studio Code desde el sitio web oficial (<https://code.visualstudio.com/>). Sigue las instrucciones de instalación para tu sistema *operativo*. (*Desafío 1. Instalación de Visual Studio Code*)

#### 2. *Crear una nueva carpeta de proyecto*

Crea una carpeta en tu computadora donde deseas guardar tu proyecto HTML. Puedes hacer clic derecho en cualquier ubicación y seleccionar "Nueva carpeta" para crearla.

#### 3. *Abrir la carpeta de proyecto en Visual Studio Code*

Abre Visual Studio Code.

Haz clic en "Archivo" en la barra de menú superior y selecciona "Abrir carpeta".

Navega hasta la carpeta de proyecto que creaste en el Paso 2 y haz clic en "Abrir".

#### 4. *Crear un nuevo archivo HTML*

En Visual Studio Code, haz clic derecho en la sección de explorador de archivos a la izquierda.

Selecciona "Nuevo archivo" y nómbralo como "index.html" (o cualquier otro nombre que desees seguido de la extensión ".html").

Esto creará un nuevo archivo HTML vacío.

### 5. Escribir código HTML básico

Abre el archivo "index.html" haciendo clic en él en la sección de explorador de archivos.

Escribe el siguiente código HTML básico en el archivo:

```
<!DOCTYPE html>
<html>
<head>
  <title>Mi primer proyecto HTML</title>
</head>
<body>
  <h1>Hola, mi nombre es...!</h1>
</body>
</html>
```

### 6. Guardar el archivo HTML

Haz clic en "Archivo" en la barra de menú superior y selecciona "Guardar" (o simplemente presiona Ctrl+S o Cmd+S).

Guarda el archivo en la ubicación de tu proyecto con el nombre "index.html".

### 7. Ver el resultado en el navegador web

Haz clic derecho en el archivo "index.html" en la sección de explorador de archivos.

Selecciona "Abrir con Live Server" para abrir el archivo en tu navegador web.

Si no tienes la extensión Live Server instalada, puedes instalarla desde la tienda de extensiones de Visual Studio Code. Después de la instalación, reinicia Visual Studio Code y repite el paso anterior.

¡Ahora deberías ver "Hola, mi nombre es Mirta Fernandez!" en tu navegador web. Has creado y ejecutado tu primer proyecto HTML en Visual Studio Code. A partir de aquí, puedes comenzar a agregar más elementos y estilos HTML a tu proyecto.

## Desafío 6. Configurar Token de Git con el espacio local

Aquí tienes los pasos para configurar las credenciales y el token de Git en tu espacio local:

1. Abre una terminal en tu computadora.
2. Verifica si ya tienes configurado el nombre de usuario y el correo electrónico global en Git. Puedes hacerlo ejecutando los siguientes comandos:

```
git config --global user.name
```

```
git config --global user.email
```

Si los valores de nombre de usuario y correo electrónico están configurados, puedes pasar al paso 3. Si los valores están vacíos, configúralos utilizando los comandos:

```
git config --global user.name "Tu Nombre"
```

```
git config --global user.email "tu_correo@ejemplo.com"
```

- Ejecuta el siguiente comando para configurar el token:

```
git config --global credential.helper store
```

Esto hará que Git almacene las credenciales, incluido el token, en el disco de manera persistente.

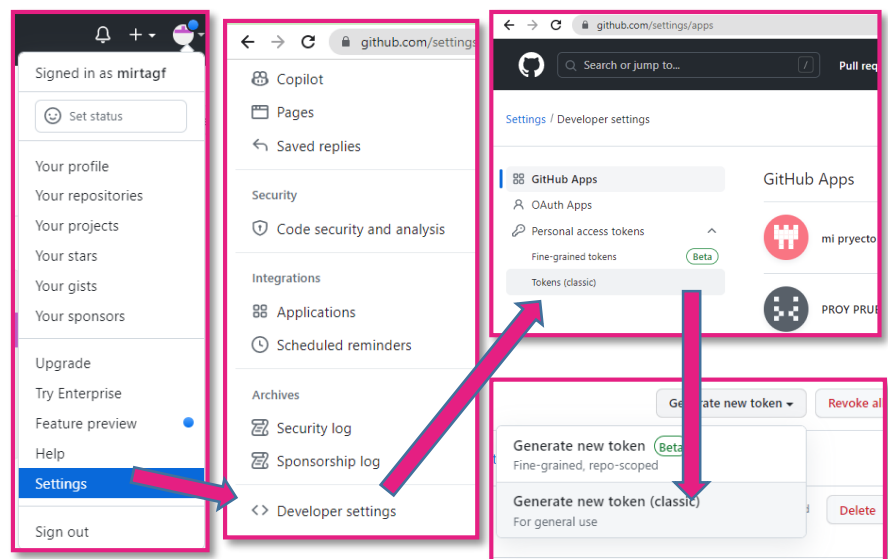
- Verifica si ya has almacenado un token en la configuración de Git. Puedes hacerlo ejecutando el siguiente comando:

```
git config --global credential.https://github.com.username
```

Si se muestra tu nombre de usuario, significa que ya tienes almacenado un token y puedes pasar al siguiente paso.

Si no se muestra nada, significa que aún no tienes un token almacenado y debes continuar con los siguientes pasos.

- Genera un Personal Access Token (PAT) en la plataforma Git que estés utilizando (por ejemplo, GitHub, GitLab, Bitbucket, etc.). Sigue las instrucciones proporcionadas en la plataforma para generar el token. Asegúrate de otorgar los permisos necesarios según tus necesidades.



- Copia el token generado.
- Ejecuta el siguiente comando para almacenar el token en la configuración de Git:

```
git config --global credential.https://github.com.username tu_usuario
```

- Reemplaza "https://github.com" con la URL de tu repositorio remoto (por ejemplo, si estás utilizando GitLab, sería "https://gitlab.com"). Además, reemplaza "tu\_usuario" con tu nombre de usuario en la plataforma Git.

Git utilizará el token almacenado en la configuración para autenticarte cuando realices operaciones que requieran autenticación, como clonar, hacer push o pull.

¡Listo! Has configurado las credenciales y el token de Git en tu espacio local. Recuerda mantener seguro tu entorno y tu sistema operativo para proteger tus credenciales.

### Desafío 7. Sincronizar GitHub

Para sincronizar tu primer proyecto HTML a GitHub directamente desde Visual Studio Code:

- Crear un repositorio en GitHub (realizado en Desafío 3. Creación de espacio en Git Hub)



-Asegúrate de dejar desmarcada la opción "Initialize this repository with a README" (Inicializar este repositorio con un README).

### 2. Inicializar el repositorio local

-Abre Visual Studio Code.

-Abre la carpeta de tu proyecto HTML en Visual Studio Code (si no lo has hecho todavía).

-Haz clic en "Terminal" en la barra de menú superior y selecciona "Nueva terminal" (o usa el atajo de teclado Ctrl+`).

-En la terminal, ejecuta el siguiente comando para inicializar el repositorio local de Git:

```
git init
```

### 3. Agregar y confirmar los cambios

-En Visual Studio Code, haz clic en la sección de control de código fuente en el panel izquierdo (tercer ícono en la barra lateral izquierda).

-Aparecerán los cambios no rastreados en tu proyecto. Haz clic en el botón "+" para agregar todos los archivos al área de preparación.

-A continuación, escribe un mensaje de confirmación en la parte superior del panel de control de código fuente que describa los cambios que realizaste.

-Haz clic en el botón de verificación (✓) para confirmar los cambios.

### 4. Conectar el repositorio remoto

-Copia la URL del repositorio remoto de GitHub. Debe tener el formato `https://github.com/usuario/repo.git`.

### 5. Configurar el origen remoto

En la terminal de Visual Studio Code, ejecuta el siguiente comando para agregar el origen remoto:

```
git remote add origin <URL_del_repositorio_remoto>
```

-Reemplaza `<URL_del_repositorio_remoto>` con la URL que copiaste en el paso anterior.

### 6. Subir los cambios al repositorio remoto

-En la terminal de Visual Studio Code, ejecuta el siguiente comando para subir los cambios al repositorio remoto en GitHub:

```
git push -u origin main
```



¡Listo! Ahora tu proyecto HTML debería estar subido a GitHub desde Visual Studio Code. Puedes visitar tu repositorio en GitHub para verificar que los cambios se hayan reflejado correctamente. Compártelo tu dirección de espacio Git en [Desafíos Semana 1](#) en el aula virtual del curso.