

GIT – GITHUB

Vamos a aprender a controlar las versiones de nuestro código, a guardarlo de forma segura en la nube y a **publicar nuestras webs para que cualquiera pueda verlas**.

1. Los Conceptos Básicos: El flujo de trabajo en Git

Git no guarda los archivos directamente en la nube al darle a "Guardar". Pasa por **cuatro estados o zonas principales**:

- **1. Directorio de Trabajo (Working Directory):** Es tu carpeta actual en el ordenador. Es tu "mesa de trabajo" donde creas, editas o borras archivos. Git sabe que estás haciendo cosas, pero aún no las está registrando oficialmente.
- **2. Área de Preparación (Staging Area):** Imagina que es una "caja de envíos". Aquí metes los archivos que ya has terminado de editar y que quieres incluir en la próxima "foto" de tu proyecto.
- **3. Repositorio Local (Local Repo):** Es el almacén en tu disco duro. Cuando haces un **Commit**, cierras esa "caja de envíos", le pones una etiqueta describiendo lo que has hecho (ej. *"Añadido el menú de navegación"*) y la guardas aquí. Ya tienes una versión guardada en tu equipo.
- **4. Repositorio Remoto (GitHub):** Es el almacén en la nube. Para llevar tus *commits* desde tu ordenador a los servidores de GitHub, hacemos un **Push** (empujar).



2. Cómo Crear un Repositorio en GitHub

El primer paso para subir nuestro proyecto es crearle un espacio en la nube.

1. Entrad en vuestra cuenta de [GitHub.com](https://github.com).
2. Arriba a la derecha, buscad el símbolo + y haced clic en **"New repository"**.
3. **Repository name:** Poned un nombre claro, por ejemplo prueba1 (Sin espacios, usad guiones).

4. **Public / Private:** Marcadlo como **Public** para que podamos publicar la página web más adelante y yo pueda evaluarla.
5. Marcad la casilla **"Add a README file"**. Es un archivo de texto donde explicaremos de qué va el proyecto.
6. Haced clic en el botón verde **"Create repository"**.

3. Instalar Git en Windows

Git es el motor que funciona por debajo de todo. Visual Studio Code solo es la interfaz gráfica que nos permite usarlo cómodamente.

1. Entrad en la página oficial de descargas de Git: git-scm.com/download/win.
2. Haced clic en **"64-bit Git for Windows Setup"** para descargar el archivo instalador.
3. Una vez descargado, ejecutad el archivo .exe.
4. Se abrirá el asistente de instalación. La regla de oro para esta primera vez es: **dejaremos todas las opciones por defecto**.
5. Pulsad el botón **"Next"** (Siguiendo) en todas y cada una de las pantallas sin cambiar ninguna casilla.
6. En la última pantalla, haced clic en **"Install"** y esperad a que la barra de progreso termine.
7. Desmarcad la casilla de "View Release Notes" y pulsad **"Finish"**.

Git es un sistema diseñado para trabajar en equipo, por lo que exige saber **quién** está haciendo los cambios. Si no le configuramos nuestra "identidad", nos dará un error al intentar hacer nuestro primer *Commit*.

1. Buscad en el menú de inicio de Windows la aplicación que acabamos de instalar llamada **"Git Bash"** y abridla. Aparecerá una ventana negra de terminal de comandos.
2. Vamos a configurar vuestro nombre. Escribid el siguiente comando (manteniendo las comillas, pero poniendo vuestro nombre real dentro) y pulsad Enter:

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

3. Ahora vamos a configurar el correo. Es **fundamental** que uséis el mismo correo con el que os habéis registrado en GitHub.

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

4. Sincronizar el Proyecto desde Visual Studio Code

Ahora vamos a conectar ese repositorio de la nube con nuestro editor de código, Visual Studio Code (VS Code).

Paso A: Clonar el repositorio (Traerlo a tu PC)

1. En vuestro repositorio de GitHub, haced clic en el botón verde "<> Code" y copiad la URL que aparece (termina en .git).
2. Abrid **VS Code**.
3. Id al tercer icono de la barra lateral izquierda (el de las tres bolitas conectadas por líneas, que es **Control de Código Fuente**).
4. Haced clic en "**Clonar repositorio**" (o *Clone Repository*).
5. Pegad la URL y pulsad Enter. Elegid en qué carpeta de vuestro ordenador queréis guardarlo y abridlo.

Paso B: Ciclo de trabajo (Guardar, Preparar, Commit y Push)

1. **Trabajar:** Cread un archivo llamado index.html y escribid vuestra estructura básica de HTML. Guardad el archivo (Ctrl + S).
2. **Preparar (Staging):** Id al icono de *Control de Código Fuente*. Veréis que vuestro archivo index.html aparece con una **U** (Untracked) o una **M** (Modified). Haced clic en el símbolo + al lado del archivo. Esto lo pasa al área de preparación.
3. **Commit:** En el cuadro de texto de arriba, escribid un mensaje claro, por ejemplo: "*Creación de la estructura base del index*". Luego, pulsad el botón "**Commit**" (o la marca de verificación ✓).
4. **Push (Sincronizar):** Finalmente, haced clic en el botón azul que dice "**Sincronizar cambios**" (Sync changes) o "Push". Esto subirá vuestro trabajo a GitHub.

Haced *commits* pequeños y frecuentes. No hagáis un solo commit al final de la semana que diga "*Todo hecho*". Si algo se rompe, no sabréis a qué punto volver.

5. Cómo Publicar Páginas (GitHub Pages)

Como estamos en Lenguaje de Marcas, no solo queremos guardar código; queremos ver la web funcionando. GitHub nos regala un hosting gratuito estático (ideal para HTML, CSS y JS) llamado **GitHub Pages**.

1. Aseguraos de que vuestro archivo principal se llama **exactamente index.html** y está en la raíz del repositorio.
2. Id a vuestro repositorio en la web de GitHub.
3. Haced clic en la pestaña "**Settings**" (Ajustes) que tiene un icono de engranaje.
4. En el menú lateral izquierdo, bajad hasta encontrar "**Pages**".
5. Bajo el apartado "**Build and deployment**", buscad "Branch" (Rama) y cambiad la opción de *None* a **main** (o master).
6. Pulsad "**Save**".

En unos minutos, en esa misma pantalla de Pages aparecerá un enlace diciendo: "*Your site is live at...*". **Ese es el enlace oficial de vuestra página web.**