

UNIDAD 0: DISEÑO DE INTERFACES DE USUARIO: Git y VSC



INTRO

[ARTICULO: Douglas Engelbart, el pionero](#)

[VIDEO: Douglas Engelbart y la madre de todas las demos](#)

[ARTICULO: Alan Kay, el padre conceptual del interfaz gráfico](#)

[VIDEO: XEROX y el mouse](#)

GIT

[Git Cheat Sheet](#)

[GitHub: Tareas Diarias](#)

Git usamos Visual Studio Code

VIDEO: [Creación Carpetas Github](#)

VIDEO: [Usar Git desde VSC](#)

DOC: [Tareas básicas GIT y VSC](#)

GIT trabajando en equipo.

DOC: GIT en equipo de trabajo

VIDEO: [Práctica hacer un Pull Request](#)

VIDEO: [Estrategias de Branching](#)

JUEGO: [GIT](#)

Mejorar Github

DOC: [Estructura Readme](#)

WEB: [Generador Readme](#)

VIDEO: [Recomendaciones Perfil Profesional](#)

Práctica 0: Creación y Gestión de un Repositorio Git desde Visual Studio Code

Objetivo:

El objetivo de esta práctica es familiarizarte con el uso de Git y Visual Studio Code (VS Code) para gestionar un repositorio local, organizar adecuadamente los archivos y carpetas de un proyecto, y poner en práctica comandos esenciales de Git. Además, aprenderás cómo realizar commits, crear ramas y manejar conflictos básicos. Todo esto te permitirá desarrollar un flujo de trabajo eficiente en control de versiones.

Requisitos previos:

- Tener instalado Git en tu ordenador.
- Tener instalado Visual Studio Code.
- Conocer los conceptos básicos de Git y de programación.
- Aceptar el trabajo <https://classroom.github.com/a/kqfd9km2>

Pasos a seguir:

1. Crear la carpeta del proyecto en tu ordenador

- Crea una nueva carpeta en tu ordenador llamada **MiProyectoGit**.
- Dentro de esta carpeta, organiza las siguientes subcarpetas y archivos:
 - **src/**: Para archivos fuente.
 - **docs/**: Para documentación.
 - **tests/**: Para pruebas o archivos de test.
 - Un archivo llamado **README.md** en la raíz del proyecto.
 - Un archivo de configuración, por ejemplo, **.gitignore**.

2. Inicializar el repositorio Git

- Abre Visual Studio Code y navega hasta la carpeta **MiProyectoGit**.
- Utiliza la terminal integrada de VS Code (ver pestaña **Terminal**) o Git Bash para inicializar un repositorio en esta carpeta con el comando:

```
git init
```

- Comprueba que se ha creado la carpeta **.git** en el directorio de tu proyecto.

3. Realizar el primer commit

- Agrega todos los archivos y carpetas creados al área de staging:

```
git add .
```

- Haz un commit inicial con el siguiente mensaje:

```
git commit -m "Commit inicial: estructura de carpetas y archivos"
```

4. Crear y cambiar a una nueva rama

- Crea una nueva rama llamada **desarrollo** y cambia a ella:

```
git checkout -b desarrollo
```

- Verifica que estás en la rama `desarrollo` utilizando:

```
git branch
```

5. Modificación y nuevos commits

- En la carpeta `src/`, crea un archivo llamado `principal.txt` que incluya un simple código de prueba, por ejemplo, un "Hello World":

```
print("Hello, Git!")
```

- Haz un commit con este nuevo archivo:

```
git add src/main.py
git commit -m "Añadido main.py con mensaje de bienvenida"
```

6. Subir los cambios a una plataforma remota (opcional)

- Si tienes una cuenta en GitHub, GitLab u otra plataforma, crea un repositorio vacío y sigue las instrucciones para enlazar tu repositorio local con el remoto.

```
git remote add origin <URL_del_repositorio_proporcionada_profesor>
git push -u origin desarrollo
```

7. Manejo de conflictos (simulación)

- Cambia a la rama `master`:

```
git checkout master
```

- Modifica el archivo `README.md` añadiendo una descripción breve del proyecto y haz un commit.

- Ejemplo de modificación:

```
# Mi Proyecto Git
Este es un proyecto para aprender a usar Git con Visual Studio
Code.
```

```
git add README.md
git commit -m "Añadido descripción en README.md"
```

- Vuelve a la rama `desarrollo` y modifica el archivo `README.md` añadiendo una lista de características:

```
## Características
- Uso de Git
- Organización de carpetas
- Comandos básicos de VS Code
```

```
git add README.md
git commit -m "Añadida lista de características al README"
```

- Intenta fusionar la rama `desarrollo` en `master` para simular un conflicto:

```
git checkout master
git merge desarrollo
```

- Resuelve el conflicto editando el archivo README .md manualmente desde VS Code, uniendo ambas versiones.

- Finaliza el proceso de fusión:

```
git add README.md  
git commit -m "Conflicto resuelto en README.md"
```

8.Exploración de otras herramientas de Git en VS Code

- Usa la sección de control de versiones de Visual Studio Code (icono de rama de Git en la barra lateral izquierda) para visualizar cambios, hacer commits, comparar archivos modificados y explorar el historial de cambios.

Entregables:

- Subir el proyecto a una plataforma como GitHub).

Tiempo estimado:

2 horas.