

**Objetivo**

Aprender a instalar y configurar GitHub en Visual Studio Code, clonar un repositorio desde GitHub, manejar las tareas en Jira, y utilizar los comandos básicos de Git (push y pull) para el control de versiones en un proyecto de programación.

Práctica 1.2: Integración de control de versiones con Git y Jira

**PASOS**

Paso 1: ¿Qué es Git y GitHub?

- Git es una herramienta que se usa para llevar el control de los cambios en los archivos de un proyecto. A esto se le llama control de versiones. Con Git puedes guardar "versiones" de tu trabajo, regresar a versiones anteriores, y colaborar con otras personas.
- GitHub es una página web donde puedes guardar tus proyectos con Git. Es como una nube para tus códigos, además puedes compartir tus proyectos con otras personas o colaborar juntos en un mismo repositorio.
- ¿Para qué sirve?
 - ➔ Guardar versiones de tu código.
 - ➔ Recuperar versiones anteriores si algo falla.
 - ➔ Compartir tu código con otras personas.
 - ➔ Trabajar en equipo sin sobrescribir el trabajo de otros.

Paso 2: Instalar Git.

● Instrucciones:

- 1 Ve a la página oficial de Git: <https://git-scm.com>
- 2 Da clic en el botón "Download for Windows" (u otro sistema operativo que tengas).
- 3 Ejecuta el archivo que se descargue. Sigue los pasos del instalador. Solo presiona "Siguiente" hasta terminar, sin cambiar nada.
- 4 Una vez instalado, abre el programa "Git Bash" para comprobar que funciona.

● Verificación:

- ➔ Abre el menú de inicio de Windows.
- ➔ Escribe "Git Bash" y ábrelo.
- ➔ Escribe este comando y presiona ENTER:

● Instalar la extensión de GitHub en Visual Studio Code:

- 1 Abre Visual Studio Code.
- 2 Ve al ícono de los cuadritos del lado izquierdo (Extensiones).
- 3 En la barra de búsqueda escribe: `GitHub Pull Requests and Issues`
- 4 Da clic en Instalar.

Paso 3: Clonar un repositorio en Visual Studio Code.

● ¿Qué es clonar?

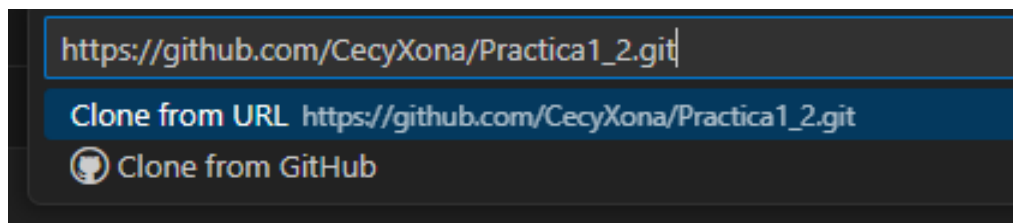
- Clonar significa copiar un proyecto desde GitHub a tu computadora para poder modificarlo.

- **Pasos:**

- 1 Entra a este repositorio que hemos preparado:
https://github.com/CecyXona/Practica1_2/tree/CecyXona-rama-frontend
- 2 Da clic en el botón verde "Code" y copia el enlace que aparece (termina en `.git`).
- 3 En el explorador de archivos de tu computadora, crea una nueva carpeta llamada "[Practica1_2](#)".
- 4 Abre Visual Studio Code y abre la carpeta [practica1_2](#) que anteriormente creamos.
- 5 Presiona **Ctrl + Shift + P** para abrir la barra de comandos.
- 6 Escribe: **Git: Clone** y selecciona esa opción.



- 7 Pega el enlace copiado y presiona ENTER.



- 8 Elige una carpeta de tu computadora donde quieres guardar el proyecto.
- 9 Cuando VSCode pregunte si deseas abrir el proyecto, da clic en "Sí".
- 10 Una vez clonado el proyecto, vamos a navegar a nuestra Rama de FrontEnd con el siguiente comando en la terminal de VS Code: **git checkout CecyXona-rama-frontend**

- **Modificación de archivo colaborativo**
- Una vez clonado nuestro proyecto, ingresaremos a nuestro index para registrar nuestro plantel o área de trabajo.
- Seleccionamos archivo que modificamos con el comando:
- `git add "index.html"`
- Agregamos commit para comentar los cambios realizados:
- `git commit -m "agregué mi plantel (nombre del plantel)"`
- Subimos nuestros cambios al repositorio (**NOTA:** Los cambios no se verán reflejados por razones de colaboración en github. Para que surtan efecto, deberán ser agregados al repositorio como colaboradores.).
- `git push`

Paso 4: Crear tu propio repositorio y subir tu proyecto (Push).

- **¿Qué es push?**
- Push significa enviar los cambios de tu computadora a GitHub.
- **Crea un nuevo proyecto simple:**
 - 1 Abre Visual Studio Code.
 - 2 Crea una nueva carpeta llamada mi-primer-repositorio.
 - 3 Dentro de la carpeta crea un archivo llamado `index.html`.
 - 4 Escribe este contenido:

```
<!DOCTYPE html>
<html>
<head>
  <title>Mi Primer Proyecto</title>
</head>
<body>
  <h1>¡Hola Mundo desde GitHub!</h1>
</body>
</html>
```

- 5 Guarda el archivo (Ctrl + S).

- **Publica tu proyecto en GitHub:**

- 1 Entra a <https://github.com>
- 2 Da clic en tu perfil > Repositories > New
- 3 Escribe el nombre del repositorio: mi-primer-repositorio
- 4 Marca la opción Add a README file.
- 5 Da clic en Create repository

- **Conecta tu proyecto local a GitHub:**

- 1 Abre Visual Studio Code en la carpeta de tu proyecto.
- 2 Abre una terminal de VS Code y escribe estos comandos uno por uno:

- **git init**

- **git remote add origin https://github.com/TU_USUARIO/mi-primer-repositorio.git**

Nota: Reemplaza TU_USUARIO por tu nombre de usuario de GitHub.



- `git add .`
- `git commit -m "Subiendo mi primer proyecto"`
- `git push -u origin main`
- Si te pide login, ingresa tu usuario y contraseña o usa el token de acceso personal (GitHub te lo genera si activas autenticación).

Paso 5: Descargar cambios (Pull).

- ¿Qué es pull?
- Pull es el comando que permite traer a tu computadora los últimos cambios hechos en GitHub por ti u otros colaboradores.
- Para usarlo:
 - ➔ Abre Visual Studio Code en el proyecto conectado a GitHub.
 - ➔ Abre la terminal de VS Code y escribe:
- `git pull origin main`
- Esto asegurará que tienes la versión más actual del proyecto.
- Carga cambios (Push)
- Modifica tu archivo index colocando el nombre de tu plantel o área de trabajo dentro del body y aplica un `push`.
- Seleccionamos archivo que modificamos con el comando:
- `git add "index.html"`
- Agregamos commit para comentar los cambios realizados:
- `git commit -m "agregué mi plantel (nombre del plantel)"`
- Subimos nuestros cambios al repositorio
- `git push`
- Una vez realizado esto, ingresamos a git hub para verificar que los cambios se hayan aplicado correctamente.

Paso 6: Crear y gestionar tareas en Jira.

- Pasos para comenzar en Jira:

- ➔ Crea un proyecto nuevo, selecciona tipo “Scrum”.
- ➔ Dentro del proyecto, da clic en "**Create**" para agregar una tarea.
- ➔ Llena los campos:
 - **Título:** Subir primer proyecto a GitHub
 - **Descripción:** Crear HTML básico y publicarlo en repositorio
 - **Asignado a:** (Tu nombre)
 - **Prioridad:** Media
- ➔ Puedes agregar más tareas como:
 - Clonar repositorio de prueba
 - Editar archivo HTML
 - Probar comandos Git

- Una vez realizado, finaliza las tareas dentro de Jira