

Taller Práctico: Colaborando con GitHub

Objetivo: Aprender el flujo de trabajo colaborativo básico en GitHub. Al final de este taller, entenderás cómo varios desarrolladores pueden trabajar juntos en un mismo proyecto, integrando sus cambios de forma ordenada y resolviendo conflictos simples.

Participantes:

- **Usuario A** (Líder del Proyecto)
- **Usuario B**
- **Usuario C**
- **Usuario D**

Requisitos:

- Cada participante debe tener una cuenta de GitHub.
- Tener Git instalado en su computadora.
- Saber abrir y usar la terminal o línea de comandos.

Paso 1: Configuración Inicial

En esta fase, el **Usuario A** creará el repositorio central y dará acceso a los demás colaboradores.

Acciones para el Usuario A:

1. **Crear el Repositorio:**
 - Ve a tu perfil de GitHub y haz clic en **"New repository"**.
 - **Nombre del repositorio:** taller-colaborativo
 - **Descripción:** Nuestro primer proyecto en equipo.
 - Déjalo como **Público**.
 - **Importante:** Marca la casilla **"Add a README file"**. Esto crea el archivo inicial del proyecto.
 - Haz clic en **"Create repository"**.
2. **Añadir Colaboradores:**
 - Dentro de tu nuevo repositorio, ve a **Settings > Collaborators > Add people**.
 - Busca por su nombre de usuario de GitHub a **Usuario B**, **Usuario C** y **Usuario D** y envíales una invitación.

Acciones para todos los usuarios (A, B, C y D):

El usuario A (líder) deberá hacer lo siguiente en el repositorio antes de clonarlo en su equipo:

Editar el archivo [README.md](#) directamente en **github**



Agregar lo siguiente:

Taller Colaborativo
Nuestro primer proyecto en equipo.

Participantes:

1. **Aceptar la Invitación (Usuarios B, C, D):** Recibirán un correo electrónico con la invitación.
2. **Clonar el Repositorio:**
 - Vayan a la página del repositorio `taller-colaborativo`.
 - Hagan clic en el botón verde `< > Code` y copien la URL HTTPS.
 - Abran su terminal y ejecuten el siguiente comando para crear una copia local del proyecto:

```
git clone [URL_COPIADA]
```

Finalmente, entren en la carpeta del proyecto:

```
cd taller-colaborativo
```

Paso 2: Aportes Individuales

Cada participante hará un pequeño cambio en el archivo `README.md` para añadir su nombre a una lista de participantes.

Acciones para cada usuario (uno por uno):

1. **Abre el archivo `README.md`** con cualquier editor de texto.
2. **Añade tu nombre** bajo la sección "Participantes". Por ejemplo, el Usuario A añadiría su nombre así:

Taller Colaborativo
Nuestro primer proyecto en equipo.

Participantes:

- Usuario A

3. **Sigue el ciclo de trabajo colaborativo:** Este es el corazón del taller.

Paso A: Preparar los cambios (git add)
Indica a Git qué archivos quieres guardar.

```
git add README.md
```

Paso B: Guardar los cambios (git commit)
Confirma los cambios con un mensaje claro que describa lo que hiciste.

```
git commit -m "Añade a [Tu Nombre] a la lista"
```

Paso C: Sincronizar (git pull)
Antes de subir tus cambios, debes bajar los cambios que otros hayan hecho mientras tú trabajabas. Esto evita problemas.

```
git pull origin main
```

Paso D: Subir tus cambios (git push)
Ahora sí, envía tus cambios al repositorio central en GitHub para que todos los vean.

```
git push origin main
```

Cada vez que un usuario complete estos pasos, los demás pueden refrescar la página de GitHub para ver cómo el archivo `README.md` se actualiza con el nuevo nombre.

Paso 3: Resolución de Conflictos ¿Qué Pasa si Chocamos?

Vamos a simular un escenario muy común: dos personas modifican la misma línea de un archivo al mismo tiempo.

Escenario:

El Usuario C y el Usuario D van a cambiar el título del proyecto en el archivo `README.md` simultáneamente.

- **El Aporte del Usuario C (El Rápido):**
 - **Usuario C:** Abre `README.md` y cambia el título a: `# El Gran Taller Colaborativo`.
 - Guarda el archivo y sigue el ciclo completo rápidamente:

```
git add README.md
git commit -m "Actualiza el título del proyecto"
git push origin main
```

- ¡Listo! El cambio del Usuario C ya está en GitHub.
- **El Aporte del Usuario D (El Conflictivo):**
 - **Usuario D:** (Sin saber lo que hizo C) Abre `README.md` y cambia el título a: `# Taller de Colaboración Definitivo`.
 - Guarda el archivo y sigue los primeros dos pasos del ciclo:

```
git add README.md
git commit -m "Mejora el título del taller"
```

- Ahora, intenta sincronizar con `git pull` antes de subir:

```
git pull origin main
```

- **¡Conflicto!** La terminal mostrará un error que dice `MERGE CONFLICT`. Git no sabe cuál de los dos títulos es el correcto y te pide que lo decidas.
- **Resolviendo el Conflicto (Usuario D):**
 - Abre el archivo `README.md`. Verás algo así:

```
<<<<<<< HEAD
# Taller de Colaboración Definitivo
=====
# El Gran Taller Colaborativo
>>>>>>> [código del commit de C]
```

¿Qué significa esto?

- Lo que está entre `<<<<<<< HEAD` y `=====` es **tu cambio**.
- Lo que está entre `=====` y `>>>>>>>` es el **cambio que vino de GitHub** (el del Usuario C).
- **La solución:** Edita el archivo para dejar la versión final que desees. Puedes elegir una, la otra, o combinarlas. Elimina las líneas `<<<<<<<`, `=====` y `>>>>>>>`. Por ejemplo:

```
# El Gran Taller de Colaboración Definitivo
```

- **Finaliza la resolución:** Una vez arreglado el archivo, debes completar el ciclo para guardar la solución del conflicto.

```
git add README.md
git commit -m "Resuelve conflicto en el título"
git push origin main
```