Taller Práctico: Introducción al Control de Versiones

Julian F. Latorre
Bootcamp Desarrollo Web Full Stack

Objetivo del Taller

El objetivo de este taller es familiarizarte con los conceptos básicos del control de versiones utilizando Git. Aprenderás a crear un repositorio, hacer commits, trabajar con ramas y resolver conflictos simples.

1. Configuración Inicial

Antes de comenzar, asegúrate de tener Git instalado en tu sistema. Puedes verificarlo ejecutando el siguiente comando en tu terminal:

```
git --version
```

Si Git no está instalado, descárgalo e instálalo desde `https://git-scm.com/`.

2. Ejercicio 1: Creación de un Repositorio

Tarea

Crea un nuevo directorio para tu proyecto y inicializa un repositorio Git en él.

Pasos:

1. Abre tu terminal.
2. Crea un nuevo directorio llamado "mi_proyecto_git":

```
mkdir mi_proyecto_git
cd mi_proyecto_git
```

3. Inicializa un nuevo repositorio Git:

```
git init
```

Comprobación

Ejecuta el siguiente comando y asegúrate de que la salida indique que se ha inicializado un repositorio Git vacío:

```
git status
```

3. Ejercicio 2: Primer Commit

Tarea

Crea un archivo HTML simple y realiza tu primer commit.

Pasos:

1. Crea un archivo llamado ïndex.htmlçon el siguiente contenido:

```
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial
    -scale=1.0">
    <title>Mi Proyecto Git</title>
</head>
<body>
    <h1>Bienvenido a mi proyecto Git</h1>
    <p>Este es mi primer archivo en un repositorio Git.</p>
</body>
</html>
```

2. Añade el archivo al área de preparación:

```
git add index.html
```

3. Realiza tu primer commit:

```
git commit -m "Primer commit: Añadir archivo index.html"
```

Comprobación

Ejecuta el siguiente comando y asegúrate de que muestra información sobre tu commit:

```
git log
```

4. Ejercicio 3: Trabajando con Ramas

Tarea

Crea una nueva rama, realiza cambios en ella y luego fusiona estos cambios con la rama principal.

Pasos:

1. Crea una nueva rama llamada "feature-navbar":

```
git branch feature-navbar
```

2. Cambia a la nueva rama:

```
git checkout feature-navbar
```

3. Modifica el archivo ïndex.html.ªñadiendo una barra de navegación:

```
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial
    -scale=1.0">
    <title>Mi Proyecto Git</title>
</head>
<body>
    <nav>
        <ul>
            <li><a href="#home">Inicio</a></li>
            <li><a href="#about">Acerca de</a></li>
            <li><a href="#contact">Contacto</a></li>
        </ul>
    </nav>
    <h1>Bienvenido a mi proyecto Git</h1>
    <p>Este es mi primer archivo en un repositorio Git.</p>
</body>
</html>
```

4. Añade y realiza un commit de los cambios:

```
git add index.html
git commit -m "Añadir barra de navegación"
```

5. Vuelve a la rama principal:

```
git checkout main
```

6. Fusiona la rama "feature-navbarçon la rama principal:

```
git merge feature-navbar
```

Comprobación

Abre el archivo index.html y verifica que la barra de navegación está presente en la rama principal.

5. Ejercicio 4: Resolución de Conflictos

Tarea

Crea un conflicto intencionalmente y aprende a resolverlo.

Pasos:

1. Crea una nueva rama llamada çambio-titulo":

```
git checkout -b cambio-titulo
```

2. Modifica el título en el archivo index.html:

```
<h1>Bienvenido a mi increíble proyecto Git</h1>
```

3. Realiza un commit con este cambio:

```
git add index.html
git commit -m "Cambiar título de la página"
```

4. Vuelve a la rama principal:

```
git checkout main
```

5. Modifica el mismo título en el archivo index.html, pero de forma diferente:

```
<h1>Proyecto Git: Una introducción al control de versiones</h1
    >
```

6. Realiza un commit con este cambio:

```
git add index.html
git commit -m "Actualizar título con descripción"
```

7. Intenta fusionar la rama çambio-tituloçon la rama principal:

```
git merge cambio-titulo
```

Git detectará un conflicto. Abre el archivo index.html y verás algo como esto:

```
<<<<<<< HEAD
<h1>Proyecto Git: Una introducción al control de versiones</h1>
=======
<h1>Bienvenido a mi increíble proyecto Git</h1>
>>>>>>> cambio-titulo
```

Resolución del Conflicto

Edita el archivo manualmente para resolver el conflicto. Por ejemplo:

```
<h1>Bienvenido a mi increíble proyecto Git: Una introducción
    al control de versiones</h1>
```

Luego, añade el archivo y completa la fusión:

```
git add index.html
git commit -m "Resolver conflicto en el título"
```

6. Ejercicio 5: Trabajando con Repositorios Remotos

Tarea

Crea un repositorio en GitHub y sube tu proyecto local.

Pasos:

1. Crea una cuenta en GitHub si aún no tienes una.
2. Crea un nuevo repositorio en GitHub (no inicialices con README).
3. Conecta tu repositorio local con el remoto:

```
git remote add origin https://github.com/tu-usuario/tu-
    repositorio.git
```

4. Sube tus cambios al repositorio remoto:

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

Comprobación

Visita tu repositorio en GitHub y verifica que todos tus archivos y commits están allí.

7. Conclusión

¡Felicidades! Has completado el taller práctico de introducción al control de versiones con Git. Ahora tienes experiencia en:

- Crear y inicializar un repositorio
- Realizar commits
- Trabajar con ramas
- Resolver conflictos
- Conectar con repositorios remotos

Estos conceptos son fundamentales para el desarrollo de software colaborativo y te serán muy útiles en tus futuros proyectos.

8. Recursos Adicionales

Para seguir aprendiendo sobre Git y control de versiones, te recomendamos los siguientes recursos:

- Pro Git Book
- Atlassian Git Tutorial
- Learn Git Branching