

Actividad 2.2: Trabajando con repositorios remotos y Github

Introducción

El objetivo de esta actividad es comenzar a trabajar con repositorios remotos, en este caso con Github. Es recomendable que se haya realizado la actividad 2.1 previamente, ya que el trabajo partirá de lo realizado en ella.

Debes documentar todas las actividades propuestas, haciendo capturas de pantalla de todas las acciones realizadas y los resultados obtenidos, creando un informe en formato pdf.

Crear una Cuenta en GitHub

Paso 1.1: Regístrate en GitHub

1. Ve a la página de GitHub en <https://github.com>.
2. Haz clic en Sign up (Regístrate) y sigue las instrucciones para crear una cuenta con tu correo electrónico.
3. Después de completar el registro, verifica tu correo electrónico si GitHub te lo solicita.

Paso 1.2: Configuración básica de tu cuenta de GitHub

1. Inicia sesión en tu cuenta de GitHub.
2. En la parte superior derecha, haz clic en tu perfil y selecciona Settings (Configuración).
3. Revisa y ajusta tus preferencias, como el idioma de la interfaz y las notificaciones de correo electrónico.

Configurar Git para Trabajar con GitHub

Paso 2.1: Configura tu nombre de usuario y correo electrónico en Git (si no lo has hecho ya). Deberías haberlo hecho en la actividad 2.1.

Paso 2.2: Configurar SSH o HTTPS para conectarte con GitHub.

GitHub permite dos métodos principales para conectarte a repositorios: HTTPS y SSH. Te recomendamos configurar SSH para mayor seguridad.

- Generar una clave SSH (si aún no tienes una): En tu terminal, ejecuta el siguiente comando para generar una nueva clave SSH:

```
ssh-keygen -t ed25519 -C "tuemail@ejemplo.com"
```

Sigue las instrucciones, presionando **Enter** cuando te pida el nombre del archivo y la ubicación. Luego, añade la clave pública a tu agente SSH:

```
eval "$(ssh-agent -s)"
```

```
ssh-add ~/.ssh/id_ed25519
```

- Añadir tu clave SSH a GitHub: Copia tu clave pública con el siguiente comando:

```
cat ~/.ssh/id_ed25519.pub
```

Entra en tu cuenta de GitHub y navega a Settings > SSH and GPG keys. Haz clic en New SSH key, ponle un nombre y pega tu clave pública en el campo correspondiente.

- Probar la conexión: Para verificar que la conexión con GitHub a través de SSH funciona correctamente, ejecuta:

```
ssh -T git@github.com
```

Si la configuración es correcta, deberías ver un mensaje de bienvenida.

Crear un Repositorio en GitHub

Paso 3.1: Crear un nuevo repositorio

1. En tu cuenta de GitHub, haz clic en el ícono de “+” en la esquina superior derecha y selecciona New repository.
2. Rellena los detalles:
 - Repository name: Ponle un nombre a tu repositorio (por ejemplo, `primer_proyecto_nombre_apellido`).
 - Description (opcional): Describe brevemente de qué se trata el repositorio.
 - Elige si el repositorio será público o privado.
 - No marques la opción de añadir README por ahora (esto lo haremos después).
3. Haz clic en Create repository.

Conectar el Repositorio Local con GitHub

Supongamos que ya tienes un repositorio local en tu máquina. Ahora lo conectaremos al repositorio remoto en GitHub.

Paso 4.1: Inicializa el repositorio local (si no lo has hecho ya). Deberías haberlo hecho en la actividad 2.1. Abre tu terminal de git y navega al directorio del proyecto local:

```
cd ruta/a/tu/proyecto
```

Si el repositorio aún no está inicializado en Git, inicialízalo:

```
git init
```

Paso 4.2: Vincular el repositorio local al repositorio remoto

Usa el siguiente comando para añadir el repositorio remoto en GitHub (sustituye la URL con la de tu repositorio):

```
git remote add origin git@github.com:tuusuario/primer_proyecto_nombre_apellido.git
```

Verifica que el remoto se haya añadido correctamente:

```
git remote -v
```

Crear una Rama, Hacer Cambios y Subirlos a GitHub

Trabajar con ramas te permite realizar cambios sin afectar la rama principal.

Paso 5.1: Crear una nueva rama

Crea una nueva rama llamada `nueva-funcionalidad`:

```
git checkout -b nueva-funcionalidad
```

Ahora estarás trabajando en esta nueva rama.

Paso 5.2: Hacer cambios en tu proyecto. Modifica o añade algún archivo en tu proyecto. Por ejemplo, crea un archivo `README.md` con una descripción del proyecto.

Añade el archivo al staging area y haz un commit:

```
git add README.md
```

```
git commit -m "Añadir archivo README.md"
```

Paso 5.3: Subir la nueva rama al repositorio remoto

Envía la nueva rama al repositorio remoto en GitHub:

```
git push origin nueva-funcionalidad
```

Fusionar la Rama con la Rama Principal

Una vez que tu trabajo en la rama `nueva-funcionalidad` está listo, es momento de fusionar los cambios con la rama principal (`main`).

Paso 6.1: Ir a la rama `main`. Cambia a la rama principal (`main` o `master`):

```
git checkout main
```

Paso 6.2: Fusionar los cambios de la nueva rama. Fusiona la rama `nueva-funcionalidad` con la rama principal:

```
git merge nueva-funcionalidad
```

Si no hay conflictos, los cambios se integrarán automáticamente.

Paso 6.3: Subir los cambios fusionados a GitHub. Finalmente, sube los cambios de la rama principal al repositorio remoto:

```
git push origin main
```

Conclusión

Al finalizar esta actividad habrás aprendido a:

- Crear una cuenta en GitHub y configurar una conexión SSH.
- Crear un repositorio remoto en GitHub y conectarlo con un repositorio local.
- Trabajar con ramas en Git para desarrollar nuevas funcionalidades de forma aislada.
- Subir ramas al repositorio remoto y fusionar los cambios con la rama principal.

Este es el flujo de trabajo básico pero poderoso que te permitirá colaborar eficientemente en proyectos de software y mantener tu trabajo organizado.