

## HERRAMIENTAS DE DESARROLLO

### Laboratorio: Manejo de Repositorios, Ramas (Branching) y Commits en Git

#### Objetivo:

El objetivo de este laboratorio es enseñar cómo manejar repositorios, trabajar con ramas (branching), y realizar commits en **Git**. Los estudiantes aprenderán a crear y gestionar ramas para el desarrollo paralelo, hacer commits efectivos, y cómo manejar situaciones comunes al trabajar con repositorios.

---

#### Procedimientos Básicos:

##### 1. Crear un Repositorio Local:

Para crear un nuevo repositorio Git en tu máquina local:

```
git init
```

##### 2. Clonar un Repositorio Remoto:

Para clonar un repositorio remoto a tu máquina local:

```
git clone <URL_del_repositorio>
```

##### 3. Crear una Nueva Rama:

Para crear una nueva rama:

```
git branch <nombre_rama>
```

##### 4. Cambiar de Rama:

Para cambiar a una rama existente:

```
git checkout <nombre_rama>
```

También puedes crear y cambiar de rama al mismo tiempo con:

```
git checkout -b <nombre_rama>
```

##### 5. Ver las Ramas Existentes:

Para listar todas las ramas del repositorio:

```
git branch
```

## 6. Realizar un Commit:

Después de modificar archivos y añadirlos al Staging Area, realiza un commit:

```
git commit -m "Mensaje descriptivo sobre los cambios"
```

## 7. Fusionar una Rama:

Para fusionar una rama en la rama actual (por ejemplo, fusionar `feature` en `main`):

```
git merge <nombre_rama>
```

## 8. Eliminar una Rama:

Para eliminar una rama local que ya no es necesaria:

```
git branch -d <nombre_rama>
```

## 9. Subir Cambios al Repositorio Remoto:

Para subir los cambios de la rama actual al repositorio remoto:

```
git push origin <nombre_rama>
```

## 10. Obtener Cambios desde el Repositorio Remoto:

Para actualizar tu repositorio local con los últimos cambios desde el repositorio remoto:

```
git pull origin <nombre_rama>
```

---

## Preguntas Teóricas:

1. **¿Qué es una rama (branch) en Git?**
  - **Respuesta:** Una rama es una versión del proyecto donde puedes realizar cambios de forma independiente sin afectar el código en la rama principal (usualmente `main` o `master`).
2. **¿Qué es un commit en Git y qué información contiene?**
  - **Respuesta:** Un commit es un registro de los cambios realizados en el proyecto. Contiene la información sobre qué archivos fueron modificados, quién hizo el cambio, cuándo se hizo y un mensaje que describe los cambios realizados.
3. **¿Por qué usar ramas (branching) en Git?**
  - **Respuesta:** Las ramas permiten realizar trabajos paralelos, como desarrollar nuevas características o corregir errores, sin afectar el código en la rama principal. Esto mejora la organización y la seguridad del código.
4. **¿Qué es el `git merge`?**
  - **Respuesta:**
5. **¿Cómo puedes ver las ramas en un repositorio Git?**
  - **Respuesta:**

6. **¿Cuál es la diferencia entre `git pull` y `git fetch`?**
  - **Respuesta:**
7. **Crea un repositorio local en tu máquina**
  - **Respuesta:**
8. **Clona un repositorio remoto a tu máquina local**
  - **Respuesta:**
9. **Crea una nueva rama llamada `prueba/login`**
  - **Respuesta:**
10. **Cambia a la rama `prueba/login` que acabas de crear**
  - **Respuesta:**