Git: Repositorios, Commits y Branches

Bootcamp Desarrollo Web Full Stack

28 de octubre de 2024

Introducción a Git

- Sistema de control de versiones distribuido
- Creado por Linus Torvalds en 2005
- Diseñado para manejar proyectos de cualquier tamaño
- Beneficios:
  - Trabajo colaborativo
  - Historial completo de cambios
  - Experimentación con nuevas ideas
  - Resolución de conflictos entre versiones

Repositorios en Git

- Un repositorio es un espacio para almacenar un proyecto y su historial
- Creación de un repositorio:

```
git init
```

- Estructura de un repositorio:
  - Directorio de trabajo
  - Área de preparación (staging area)
  - Directorio .git

Commits en Git

- Un commit es una instantánea de los cambios en un momento específico
- Creación de un commit:

```
git add <archivo>
git commit -m "Mensaje descriptivo"
```

- Buenas prácticas para mensajes de commit:
  - Usar el imperativo presente
  - Ser conciso pero descriptivo
  - Separar título y cuerpo del mensaje
  - Limitar el título a 50 caracteres

Branches en Git

- Las ramas son líneas independientes de desarrollo
- Creación de una nueva rama:

```
git branch <nombre-de-la-rama>
git checkout <nombre-de-la-rama>
```

- O en un solo comando:

```
git checkout -b <nombre-de-la-rama>
```

- Trabajando con ramas:
  - Listar ramas: `git branch`
  - Cambiar entre ramas: `git checkout <nombre-de-la-rama>`
  - Eliminar una rama: `git branch -d <nombre-de-la-rama>`

Fusión de Ramas

- Para incorporar cambios de una rama a otra:

```
git checkout rama-destino
git merge rama-origen
```

- Resolver conflictos si es necesario
- Confirmar la fusión con un nuevo commit

Ejercicios Prácticos

1. Crea un nuevo repositorio Git y añade algunos archivos
2. Realiza varios commits con cambios en los archivos
3. Crea una nueva rama y haz algunos cambios en ella
4. Vuelve a la rama principal y fusiona los cambios de la nueva rama

Conclusión

- Git es una herramienta poderosa para el control de versiones
- Facilita el trabajo colaborativo y el desarrollo de software
- Dominar repositorios, commits y branches es fundamental
- Práctica constante para mejorar habilidades en Git