

Guía de Estudio para Examen de Git y GitHub

Introducción a Git

Git es un sistema de control de versiones distribuido, diseñado para manejar proyectos grandes y pequeños con velocidad y eficiencia. Facilita el trabajo colaborativo entre desarrolladores, permitiendo que múltiples versiones de un proyecto sean mantenidas simultáneamente.

Comandos Básicos de Git

- **git init**: Inicializa un nuevo repositorio de Git local.
- **git clone [url]**: Crea una copia local de un repositorio remoto.
- **git add [archivo]**: Añade archivos al área de staging, preparándolos para ser comprometidos.
- **git commit -m "[mensaje]"**: Guarda los cambios en el repositorio local, acompañado de un mensaje descriptivo.
- **git status**: Muestra el estado de los archivos en el área de trabajo y el área de staging.
- **git log**: Muestra el historial de commits, ayudando a entender la secuencia de cambios.

Ramas en Git

Las ramas permiten a los desarrolladores trabajar en características o correcciones independientemente sin afectar la línea principal del desarrollo, conocida como 'master' o 'main'.

- **git branch [nombre-rama]**: Crea una nueva rama.
- **git checkout [nombre-rama]**: Cambia de la rama actual a otra rama.
- **git merge [nombre-rama]**: Fusiona los cambios de una rama especificada con la rama actual.

Manejo de Conflictos

Un conflicto de fusión ocurre cuando dos ramas han hecho cambios incompatibles en el mismo archivo. Git requerirá que se resuelvan manualmente estos conflictos antes de poder completar la fusión.

Introducción a GitHub

GitHub es una plataforma de alojamiento de código que utiliza Git para el control de versiones. Facilita la colaboración entre desarrolladores y ofrece características como el seguimiento de problemas, solicitudes de extracción (pull requests) y una wiki para cada proyecto.

Trabajo Colaborativo en GitHub

- **Fork:** Hacer un 'fork' de un repositorio crea una copia personal en tu cuenta de GitHub, permitiéndote experimentar y proponer cambios sin afectar el repositorio original.
- **Pull Request:** Es una solicitud para que el mantenedor de un repositorio considere integrar tus cambios. Es una parte esencial del flujo de trabajo colaborativo en GitHub.

.gitignore

El archivo .gitignore se utiliza para excluir archivos y directorios del control de versiones. Esto es útil para archivos que no deseas compartir, como archivos de configuración locales o archivos temporales.

Revertir Cambios

- `git revert [hash-del-commit]`: Crea un nuevo commit que invierte los cambios especificados, permitiendo deshacer errores de forma segura.
- `git reset`: Revoca commits o cambios en el área de staging, teniendo diferentes modos como `--soft`, `--mixed`, y `--hard`, cada uno afectando de manera diferente el índice, el área de trabajo y el historial.