



TRABAJO PRACTICO #1

Resumen del tutorial de Git/GitHub.

TELECOMUNICACIONES

LUCIANO LUJAN

Resumen

El tutorial "The Basics of GitHub" ofrece una introducción concisa, pero completa a Git y GitHub, dos herramientas fundamentales en el mundo del desarrollo de software.

Git, un Sistema de Control de Versiones Distribuido, facilita el seguimiento de cambios en el código, la colaboración y el intercambio entre equipos.

Por otro lado, GitHub amplifica el poder de Git al proporcionar una plataforma en Online que simplifica aún más la colaboración y la gestión de proyectos.

Git:

- Un Sistema de Control de Versiones (VCS) rastrea cambios en proyectos mientras colaboran personas y equipos.
- Los VCS permiten recuperar versiones anteriores del proyecto y proporcionan información sobre quién, cuándo y por qué se hicieron cambios.
- Git es un Sistema de Control de Versiones Distribuido (DVCS) popular que no requiere una conexión constante a un repositorio central.
- Con Git, los desarrolladores tienen acceso a un historial completo de cambios, lo que facilita la comprensión y contribución al proyecto.
- La colaboración con Git puede ocurrir en diferentes zonas horarias, y las ramas permiten proponer cambios de forma segura.
- Git promueve la colaboración entre equipos y empresas, permitiendo que expertos colaboren en proyectos importantes.

GitHub:

- GitHub aloja repositorios Git y ofrece herramientas para mejorar el código, como problemas, solicitudes de extracción y revisión de código.
- El flujo de GitHub organiza el trabajo en repositorios, permitiendo a los desarrolladores proponer, discutir y fusionar cambios.
- GitHub fomenta la colaboración en el desarrollo de software y proporciona una plataforma para compartir proyectos públicamente.

Conclusión

"The Basics of GitHub" es una puerta de entrada invaluable al mundo de la colaboración en línea y el desarrollo de software. Desde la comprensión de los conceptos básicos hasta la participación activa en la comunidad, este curso proporciona a los estudiantes las herramientas y el conocimiento necesarios para comenzar su viaje en GitHub con confianza y habilidad.

Comandos básicos:

- **git add:** Mueve los cambios del directorio de trabajo al área de preparación.
- **git branch:** Crea, cambia o elimina ramas.
- **git checkout:** Cambia a una rama o confirmación específica.
- **git clean:** Elimina archivos sin seguimiento del directorio de trabajo.
- **git clone:** Crea una copia de un repositorio remoto.
- **git commit:** Registra los cambios preparados en el historial del proyecto.
- **git fetch:** Descarga una rama remota sin fusionarla.
- **git init:** Inicializa un nuevo repositorio Git.
- **git log:** Muestra el historial de confirmaciones del proyecto.
- **git pull:** Descarga y fusiona una rama remota.
- **git push:** Envía cambios a un repositorio remoto.
- **git reset:** Deshace cambios en el directorio de trabajo o en el historial.
- **git status:** Muestra el estado del directorio de trabajo y la próxima confirmación.

Comandos avanzados:

- **git rebase:** Mueve las confirmaciones a otra rama.
- **git revert:** Deshace una confirmación.
- **git stash:** Guarda y restaura cambios sin confirmar.

Conceptos clave:

- **Rama:** Línea independiente de desarrollo.
- **HEAD:** Puntero a la confirmación actual.
- **Repositorio:** Conjunto de confirmaciones, ramas y etiquetas.
- **Etiqueta:** Marcador para un punto específico en el historial.
- **Control de versiones:** Sistema para registrar cambios en archivos.
- **Árbol de trabajo:** Archivos extraídos del repositorio.