

TECNICATURA EN TELECOMUNICACIONES ISPC 2024

TRABAJO PRACTICO #1

Profesor: Gonzalo Vera
Catedra: Proyecto Integrador 1
Alumno: Jose Marquez



Conociendo GIT



Git es un sistema de control de versiones distribuido que permite a los desarrolladores realizar un seguimiento de los cambios en su código a lo largo del tiempo, mientras que GitHub es una plataforma basada en la web que utiliza Git para alojar y administrar repositorios de código. Git permite a los desarrolladores colaborar en proyectos, crear ramas para diferentes funciones o correcciones de errores, y fusionar los cambios en la base de código principal. GitHub proporciona una interfaz fácil de usar para administrar repositorios de Git, incluidas características como el seguimiento de problemas, las solicitudes de incorporación de cambios y las revisiones de código. En resumen, Git es una herramienta para administrar repositorios de código y realizar un seguimiento de los cambios, mientras que GitHub es una plataforma para alojar y colaborar en repositorios de Git.

- Luego de esta breve introducción, procedemos a bajar el software y se instala en nuestra computadora desde su página oficial (<https://git-scm.com/downloads>).
- Una vez instalado en la máquina, podemos hacer una primera aproximación con algunos comandos iniciales:

Comandos iniciales de GIT

En la carpeta que vamos a destinar para el proyecto, podemos hacer click derecho y en el menú contextual tendremos las opciones, **Open Git GUI here (interfaz gráfica)** u **Open Git Bash here (Consola o línea de comandos)**.

Cuando hacemos click en la opción de Bash, se abre la consola para que podamos comenzar a utilizar GIT.

- Configuración inicial
 - **git --version** *“Muestra si se tiene correctamente instalado GIT y la versión instalada de GIT en nuestro equipo, puede ser WIN, MacOS o Linux.*
 - **git config --global** *“Configuración global y no por proyecto.*
 - **git config --global user.name “ nombre”** *Configura en nombre de usuario.*
 - **Git config --global user.email example@example.com** *“configura nuestro correo electrónico” Esto nos permite enlazar nuestros repositorios locales con Git hub.*

Comandos iniciales de GIT

- - **ls:** listado de directorios
- - **cd+tab** movernos en los directorios
- - **cd..** sube un nivel de directorios
- - **pwd** muestra en qué directorio nos encontramos
- - **mkdir "Ejemplo"** Crea una carpeta
- **git --version** “Muestra si se tiene correctamente instalado GIT y la versión instalada de GIT en nuestro equipo, puede ser WIN, MacOS o Linux.
- **git config --global** “Configuración global y no por proyecto”.
- **git config --global user.name** “ nombre” Configura en nombre de usuario.
- **git init** – inicializa git en nuestro repositorio.
- **git status** - Ver el estado de los ficheros
- **git branch -m main** Cambia el nombre de la rama MASTER a MAIN.
- **git add .** - agrega todos los ficheros que están pendientes.

Comandos iniciales de GIT



- **git commit -m "MENSAJE"** Asocia un mensaje al commit, cada commit genera un hash. *Registra los cambios en el índice de Git como un commit permanente en el historial del proyecto.*
- **git log** Muestra el historial de los cambios y el hash, autor y el día y hora.
- **git remote add origin main:** Sincroniza el repositorio local con el repositorio remoto.
- **git push:** Envía los cambios locales al repositorio remoto.
- **git pull:** Descarga los cambios del repositorio remoto y los fusiona con la rama local actual.
- **git branch** Listar todas las ramas locales
- **git branch -a** Listar todas las ramas (locales y remotas)
- **git branch nombre-nueva-rama** Crear una nueva rama
- **git branch -d** Nombre de la rama a borrar
- **git checkout nombre_de_la_rama** Cambia a la rama especificada.
- **git checkout -b nombre_de_la_nueva_rama** Crea nueva rama a partir de la rama actual.
- **git merge nombre_de_la_rama** Fusiona los cambios de otra rama en la rama actual.

Comandos iniciales de GIT



- `git clone [url remota]` *comando para clonar un repositorio remoto en tu máquina local*