

Git y GitHub

Nota: para subir archivos pesado toca por el protocolo “ssh”

github: Servicio de hosting que nos permite almacenar proyectos de desarrollo de software y control de versiones usando git.

git: Un sistema de control de versiones, donde se rastrea o se realiza una minuta del trabajo colaborativo de manera bidireccional, es decir, enviar y recibir actualizaciones de proyectos.

Conceptos Basicos

Control de versiones: Sistema encargado de administrar los cambios realizados en programas.

Repositorios: Coleccion de archivos de distintas versiones de un proyecto; el cual puede ser local o remoto, local esta en el computador y remoto esta en los servidores de GitHub.

Commit: Es el principal componente de la linea de tiempo en un proyecto que se encuentra en git, el commit es un registro o un sanpshot del estado de un proyecto en un determinado momento.

Linea del tiempo: Esta representada por punto, cada punto es un commit, se puede tener una linea de tiempo en local y otra en remoto. Se puede fucionar.

GitBash: Herramienta para ejecutar comandos de Git en Windows.

Tres areas en git

Directorio de trabajo: Carpeta de proyecto que contiene todos los archivos y el directorio .git del repositorio.

Area de preparacion: Conjunto de archivos y camvios que seran incluidos en el proximo commit.

Repositorio: Directorio que contiene los metadatos y versiones de tu proyecto.

Estados de los archivos

Modified: Si la version del archivo contiene cambios no que son parte del repositorio y no se han agregado al area de trabajo

Staged: Si la version del archivo contiene cambios que no son parte del repositorio y se han agregado al area de trabajo

Committed: La version del archivo ya se encuentra en el directorio de git

Estructura del Commit

id: Es un identificador unico llamado SHA

Mensaje: Mensaje que se ha escrito en el commit

SHA (Secure Hash Algorithm): Identifica los cambios realizados, quien los hizo y donde los hizo.

Ramas en Git y la rama main

Las ramas permiten realizar los cambios sin afectar la version original, cuando se finalice los cambios en las ramas se puede combinar con la rama principal.

Crear una rama: `git branch version-start`

Como cambiar de rama

`git checkout version-start` (cambia la rama)

`git checkout -b version-js` (Crea y cambia la rama)

Cambiar de nombre a una rama en Git

Se debe estar ubicado en la rama a la cual se le va a cambiar el nombre

`git branch -m version-start version-inicio` (Cambia el nombre de la rama desde la principal)

Eliminar una rama

Solo se pueden eliminar ramas locales, no se debe estar ubicado en la rama que se esta eliminando

```
git branch -d version-inicio
```

Crear commit en una rama

1. **Se cambia a la rama:** `git checkout version-javascript`
2. **se agrega al area de preparacion:** `git add .`
3. `git commit`

GitLog para ramas

1. **Se cambia a la rama:** `git checkout version-javascript`
2. Se usa `git log`
3. `git log --oneline` (Muestra todos los commits de manera resumida)

Fusionar Ramas en git

Merge, el proceso que permite fusionar varias ramas de desarrollo en una sola, para fusionar las ramas debes estar parado en la rama que recibira la fusion.

```
git checkout main (Pararse en la rama main)
git checkout -b text-expandido (Crear una nueva rama y parase
git branch (mostrar ramas)
git status (revisar el estatus en la rama text-exptandido)
git add . (seleccionar todos los archivos de la rama text exp
git commit (Realizar un commit en la ramma text-exptandido)
git checkout main (pararse en la rama main)
git merge text-expandido (Fusionar main con text-expandido)
```

Conflictos al fusionar

Herramientas

CLI = Command Line Interface

Comandos

git —version (version de git)

winget update git (Actualizar git en Windows)

git status (Muestra el estado actual del repositorio)

git add nombreArchivo (Para agregar un archivo en el area de preparacion)

git rm (Se utiliza para eliminar un archivo en el repositorio y en local)

git log (Muestra el historial de los commits)

git commit —amend (modificar el ultimo commit)

git checkout — (descartar los cambios no confirmados en el archivo)

git reset (Para desaser los cambio en el repositorio)

.gitkeep (para mantener un repositorio v)

Configurar credenciales

git config —global user.name "Usuario de GitHub"

git config —global user.email "Correo de GitHub"