

EPET 20

TRABAJO PRACTICO DE PROGRAMACION CON GIT.

5to 2da

Barrientos Nahuel



Tp 3: programación con git.

1- Es un sistema de control de versiones distribuido, sirve para rastrear cambios en el código fuente, colaborar en proyectos y gestionar las versiones de manera eficiente. Se usa comúnmente en desarrollos de softwares.

2- Nos sirve para gestionar de manera eficiente los cambios en el código fuente, podemos trabajar con otros desarrolladores sin tener conflictos, mantener un historial de versiones para rastrear y revertir modificaciones, permite trabajar en diferentes funcionalidades simultáneamente mediante ramas para así facilitar la integración del trabajo en equipo en un proyecto.

3- La estructura básica de un repositorio de Git consiste en:

- Directorio de trabajo: tiene los archivos del proyecto en su estado actual.
- Area de preparación: es donde se almacenan los cambios seleccionados para que sean confirmados con el “git commit”.
- Repositorio: almacena el historial de versiones y la configuración del proyecto.
- Ramas: son líneas independientes de desarrollo. Las ramas por defecto son “main” o “master”.
- Commits: registran los cambios en el repositorio, se identifican cada uno con un hash único.

4- Para crear un repositorio desde cero, tenemos que ejecutar “git init” en el directorio del proyecto, después para agregar los archivos usamos “git add” y hacer el primer “git commit”

Para clonar un repositorio ya existente, usamos “git clone <>” junto con la URL del repositorio a clonar, con esto hacemos una copia completa junto con su historial de versiones.

5- Para crear un repositorio local primero necesitamos navegar al directorio de nuestro proyecto con “cd ruta/de/proyecto”. Luego iniciamos el repositorio con “git init”. Para agregar los archivos al área de preparación hacemos “git add .”. Y, luego hacemos el primer “commit”.



Para crear un repositorio remoto, conectamos el repositorio local al remoto con “git remote add origin <https://github.com/usuario/nombre>”. Para enviar los cambios al remoto hacemos “git push –u origin main”

6- Los “commit” son los registros de los cambios hechos en el proyecto. Son un punto de control que guarda archivos agregados al área de preparación junto a un breve mensaje. Su estructura principal es “git commit –m ‘mensaje’”.

7- Son líneas de desarrollo en las que podemos trabajar distintas funcionalidades o versiones de nuestro proyecto sin tocar a nuestra rama principal. Cada una es una copia del proyecto y la mas común es “main” o “master”. Estas ramas las podemos crear con “git branch nombre-rama”.

8- Se usan para integrar los cambios de una rama a otra, pero definen en como afectan el historial. El comando “merge” combina los cambios creando un “merge commit”, esto preserva el historial completo. El comando “rebase” replica los comits que hay en una rama sobre otra, reescribe el historial.

9- El comando “merge” combina los cambios creando un “merge commit”, así preservando el historial completo. El comando “rebase” reaplica los commits de una rama sobre otra, esto reescribe el historial.

10- El uso del comando “merge” es útil en el contexto de un proyecto en el cual nosotros necesitamos ver el historial de nuestro proyecto, para asi tener un mejor desarrollo colaborativo frente a un proyecto grande hecho en grupo. Si deseamos un trabajo sencillo podemos usar el comando “rebase”