Tutorías IPC1

Clase 1- Introducción a Git y Herramientas visuales a Git

Javier Monjes

Agenda

01 Introducción

02 Introducción al control de versiones

03 Git
Inicio de repositorios

04 Ejemplos
DE uso de git

05 GitHub
Teórico

06 Herramientas visuales

Control de versiones

Existen herramientas permiten a los desarrolladores llevar un seguimiento de los cambios realizados en el código de un proyecto a lo largo del tiempo.

Hay muchas herramientas disponibles, pero una de las más populares es Git.

Git es un sistema de control de versiones distribuido que permite almacenar y gestionar diferentes versiones de un proyecto de manera eficiente.

Plataformas de alojamiento de repositorio (git)

- Existen distintas plataformas web para alojar proyectos de software y colaborar en el desarrollo de código.

Github

Lanzada en 2008. Es la plataforma de alojamiento de repositorio basada en git, más popular. Adquirida por Microsoft en 2018.

Gitlab

Lanzada en 2011. Desde el principio quería diferenciarse de Github, por lo que creó un producto para todo el ciclo de vida de DevOps

Bitbucket

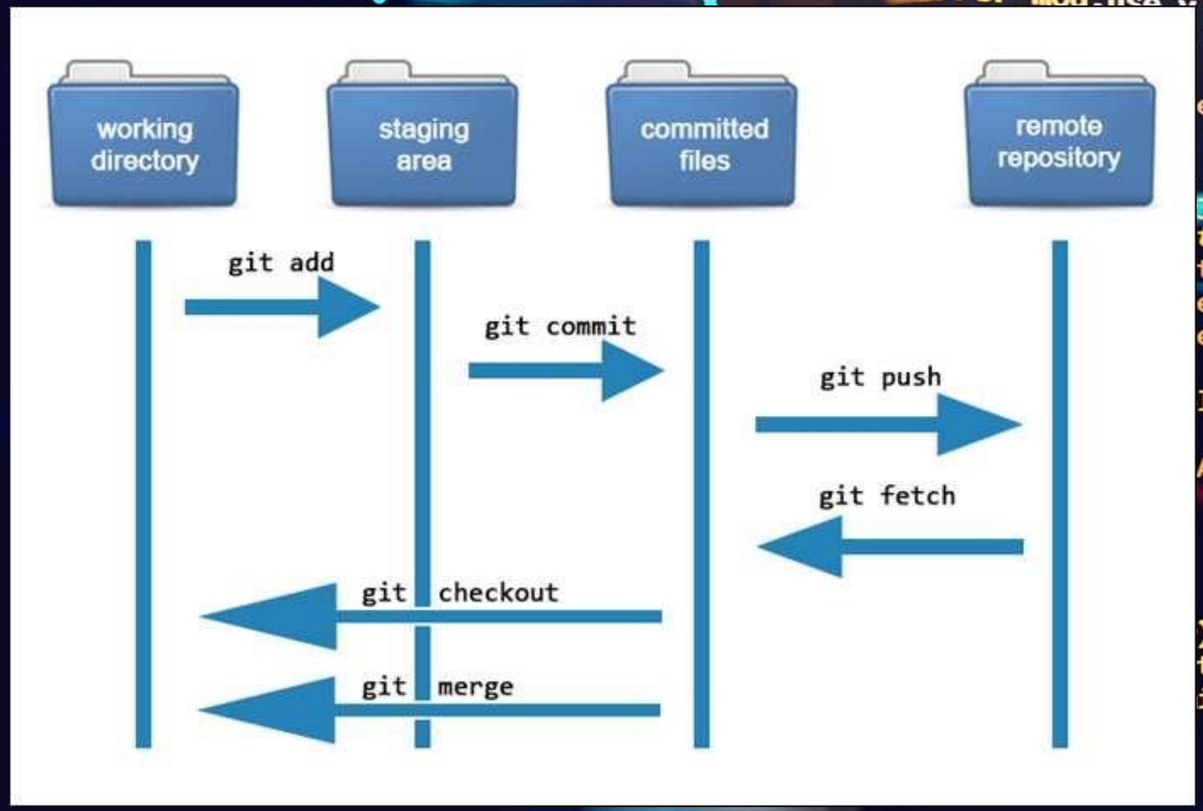
Diseñado para que lo utilicen equipos que aprovechan Jira de Atlassian.

GIT

Es una herramienta utilizada para administrar el control de las versiones de sus aplicaciones

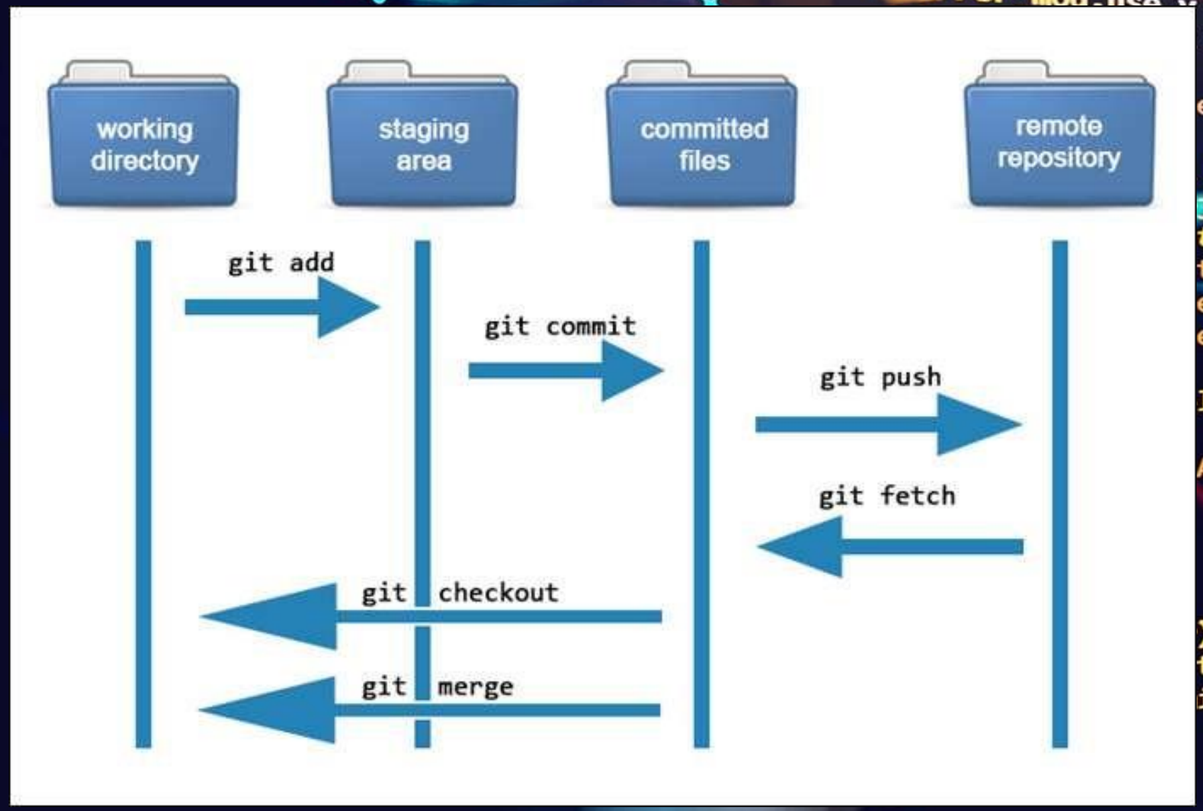
¿Cómo funciona Git?

Cada proyecto en Git tiene su propio repositorio, que es una colección de archivos y directorios que se controlan con Git.



¿Cómo Instalar Git?

https://education.github.com/pack



Comandos

- git config --list
- git config --global user.name "mi_nombre"
- git config --global user.email "mi_correo"

Configuración

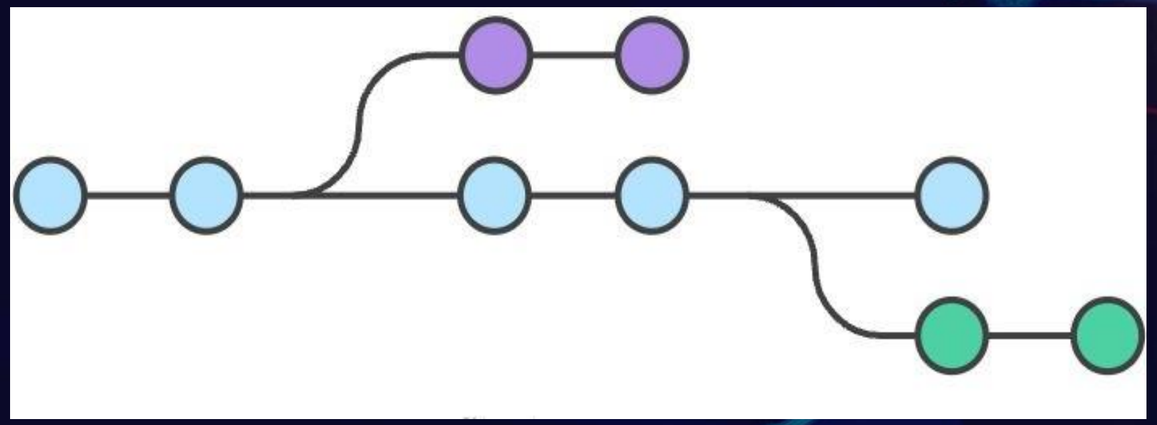
Primeros pasos

Commit

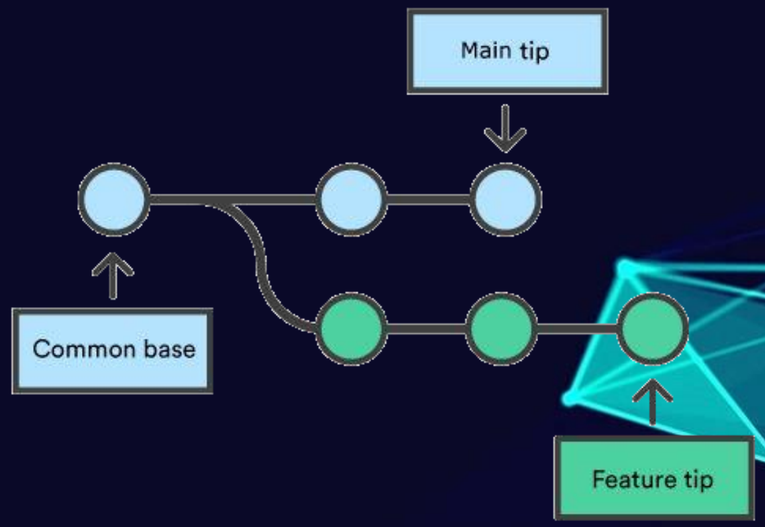
- git init
- git add .
- git commit -m “Mensaje de commit”

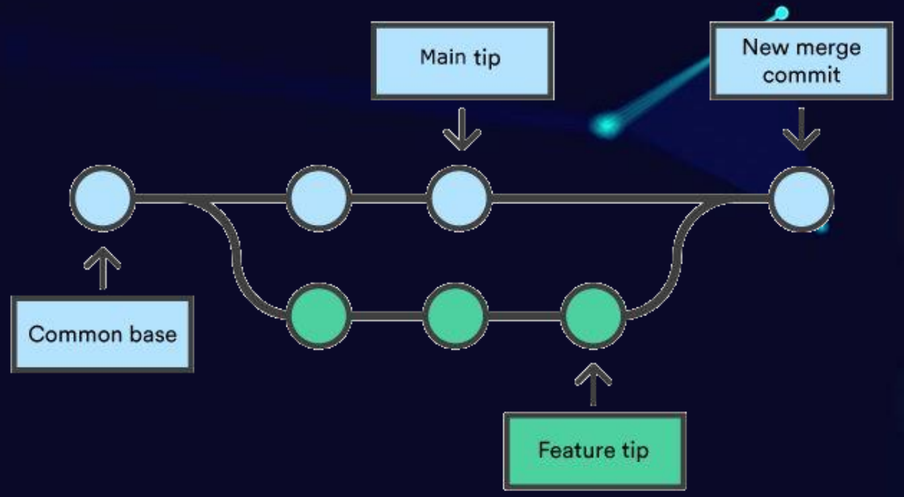
Es una instantánea de los cambios realizados en ese momento en particular.

Branch



Una rama es una bifurcación del estado del código que crea un nuevo camino para la evolución o cambios del mismo





Merge

Es una operación que combina los cambios de dos ramas diferentes en una sola, siendo una, la rama donde nos encontramos cuando ejecutamos el comando y la segunda, la rama que indiquemos después del comando.

Comandos

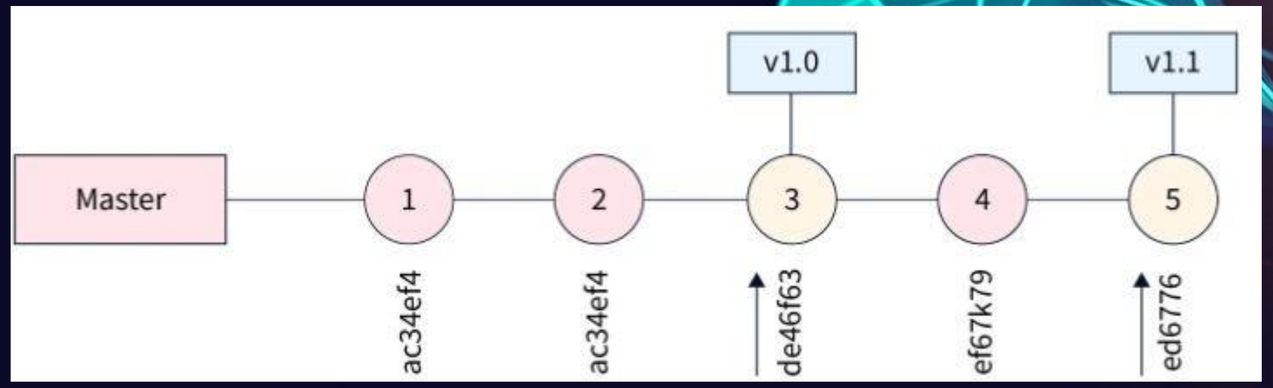
Merge

Nos movemos a la rama que aceptara los nuevos cambios

- git checkout [rama]

Rama de la cual obtenemos los cambios

- git merge [rama2]



Tag

Es una referencia específica a un punto concreto en la historia del repositorio. Se utiliza principalmente para marcar versiones importantes, cómo lanzamientos

Comandos

Tag

Listar los tags

- git tag

Crea tag y lo asigna a un commit

- git tag -a nombre id_commit -m “mensaje”

Pública el tag en nuestro repositorio

- git push origin –tags

Comandos más usados

- git clone
- git branch
- git checkout
- git status
- git add
- git commit
- git push
- git pull
- git revert
- git merge

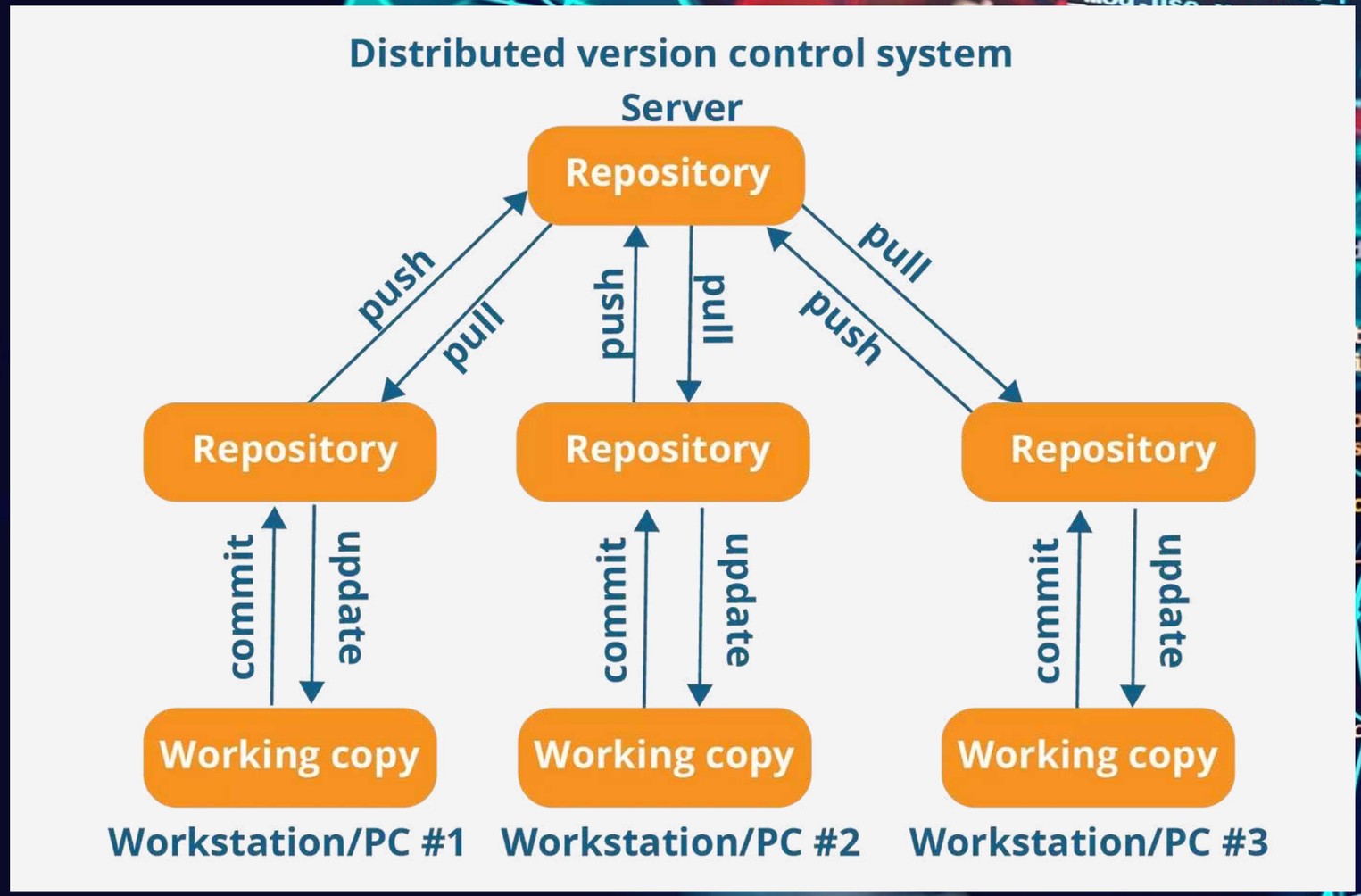
https://www.freecodecamp.org/espanol/news/10-comandos-de-git-que-todo-desarrollador-deberia-saber/

Comandos más usados

- git clone se usa para copiar un repositorio remoto en tu computadora1.
- git branch se usa para listar, crear o eliminar ramas en tu repositorio local.
- git checkout se usa para cambiar entre ramas o restaurar archivos en tu repositorio local1.
- git status muestra el estado de los cambios en tu repositorio local.
- git add agrega cambios en el directorio de trabajo al área de preparación (staging area).
- git commit guarda los cambios agregados al área de preparación en el repositorio local.
- git push envía los cambios guardados en el repositorio local a un repositorio remoto.
- git pull se usa para descargar contenido de un repositorio remoto y actualizar al instante el repositorio local para reflejar ese contenido.
- git revert deshace un commit específico creando un nuevo commit que invierte los cambios.
- git merge fusiona una rama en otra rama activa.

https://www.freecodecamp.org/espanol/news/10-comandos-de-git-que-todo-desarrollador-deberia-saber/

Colaboración entre varios usuarios



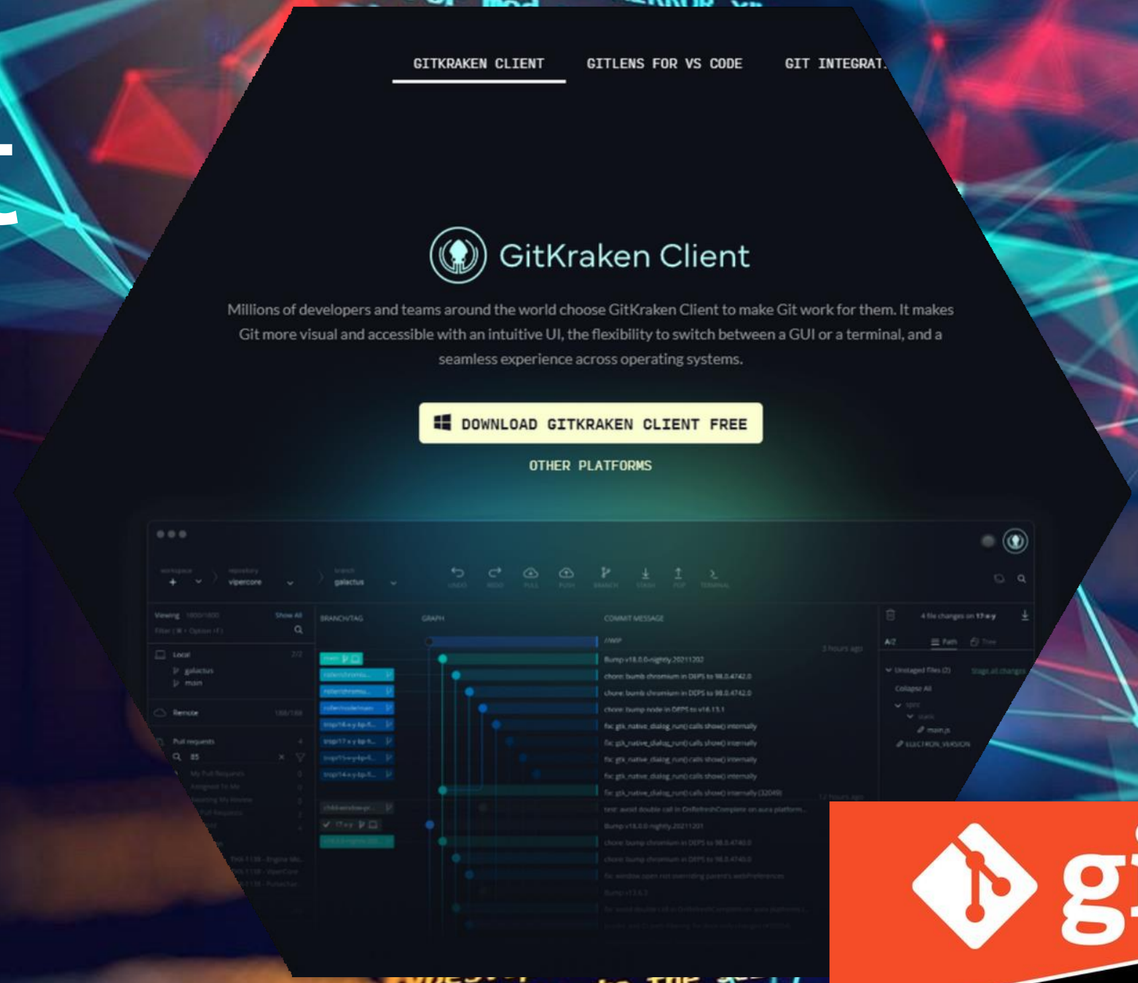
Herramientas visuales para git

Existen múltiples herramientas visuales que pueden ayudar a trabajar con Git de manera más intuitiva y fácil de entender.

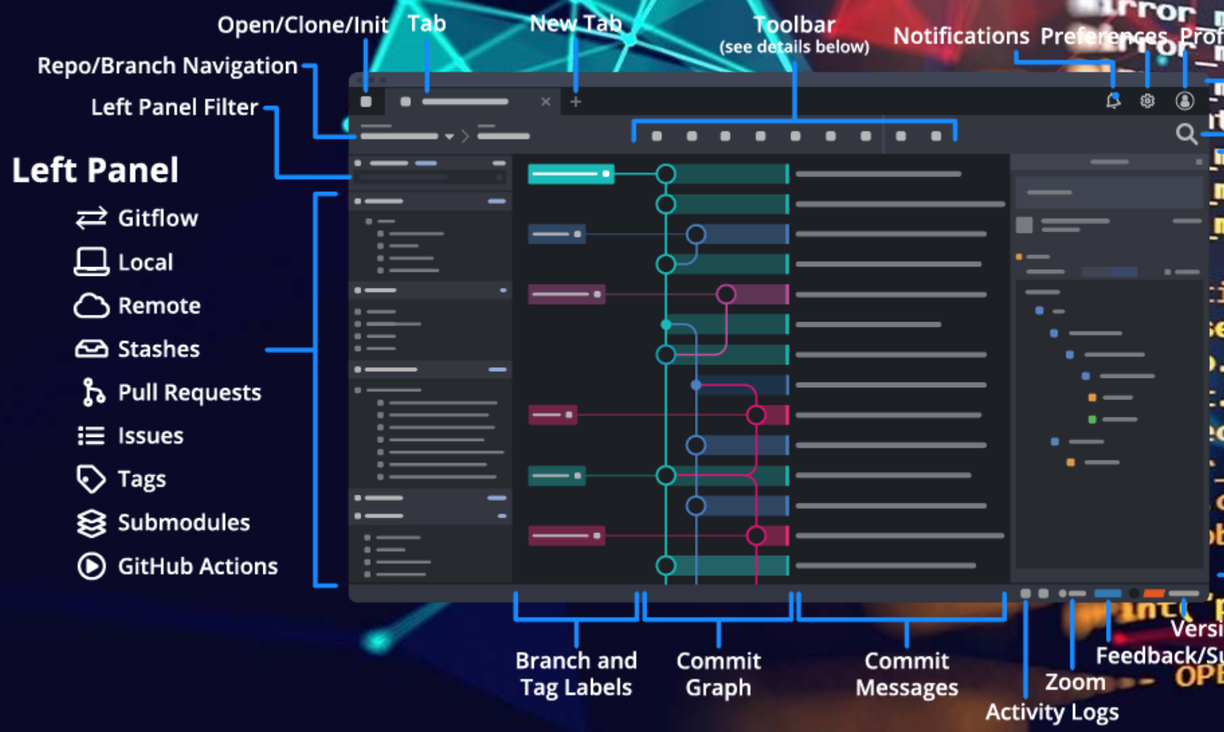
- https://git-scm.com/downloads/guis

Recomendación:

- https://www.gitkraken.com/git-client



Herramienta visual



Ejemplo Práctico

Utilizando git y realizando un programa de gestión de empleados

```java
no usages
public class Main {

    7 usages
    public static String[] empleados = new String[5];
    1 usage
    public static int[] horasTrabajadas = new int[5];

    no usages
    public static String[] nombreBonos = new String[5];
    no usages
    public static int[] porcentajesBonos = new int[5];

    1 usage
    public static void crearEmpleado(){
        Scanner input = new Scanner(System.in);
        System.out.println("Ingrese el nombre del nuevo empleado: ");
        String nombre = input.nextLine();
        System.out.println("Ingrese las horas trabajadas: ");
        int horas = input.nextInt();

        // validar que no existe ese empleado
```