

# Charla GitHub avanzado 2024

Ramas, comunicación, workflows

---

Alejandro Barrachina Argudo

2024

Universidad Complutense de Madrid

# Tabla de contenidos

Introducción

Ramas

Main

Ramas secundarias

Pull request y Merge

# Tabla de contenidos

Introducción

Ramas

Main

Ramas secundarias

Pull request y Merge

# Introducción

Charla anterior: <https://github.com/ALK222/charla-git-principiantes-2023>

Se asume que:

- Tenéis cuenta de GitHub
- Sabéis hacer crear repositorios
- Sabéis hacer *commit*, *push*, *pull*

# Tabla de contenidos

Introducción

Ramas

Main

Ramas secundarias

Pull request y Merge

# Ramas

- Mantienen el desarrollo paralelo para distintas partes del programa
- Mantienen separadas las versiones estables de las de desarrollo
- Nos permiten proteger ciertas ramas

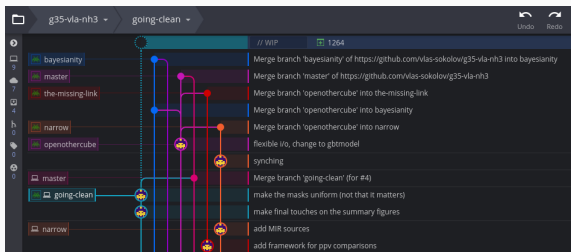


Figure 1: origen de la imagen: <https://stackoverflow.com/questions/46827112/toggling-branch-tree-view-in-gitkraken>

# Tabla de contenidos

Introducción

Ramas

Main

Ramas secundarias

Pull request y Merge



# Main

La rama Main (en proyectos antiguos también se puede encontrar como Master) es la rama principal del proyecto y la primera rama que se crea automáticamente al crear el repositorio.

- Rama (normalmente) protegida para evitar código malicioso (de manera intencional o no) sin supervisión.
- Generalmente es la rama estable del programa, solo se suben versiones definitivas

# Tabla de contenidos

Introducción

Ramas

Main

Ramas secundarias

Pull request y Merge

## Ramas secundarias

A estas ramas les daremos nosotros un nombre al crearlas. Son útiles para desarrollar nuevas características del programa o para hacer arreglos de bugs.

Estas ramas se pueden unir a la principal en cualquier momento del desarrollo para incorporar estos cambios a la rama principal.

# Tabla de contenidos

Introducción

Ramas

Main

Ramas secundarias

Pull request y Merge

# Merge

Un *Merge* nos permite unificar un conjunto de *commits* en un solo historial.

Normalmente esto se hace entre los extremos de dos ramas para fusionarlas en una sola.

Esto puede generar conflictos entre archivos que se pueden resolver con distintos editores de texto o herramientas de gestión de git.

# Pull Request

Un *pull request* es un mecanismo de GitHub para poder hacer *merge* en proyectos ajenos o para solicitar feedback antes de hacer un merge en tu propio proyecto.

En ciertos repositorios es obligatorio pasar una *pull request* con varios supervisores para poder contribuir si no eres parte del equipo.