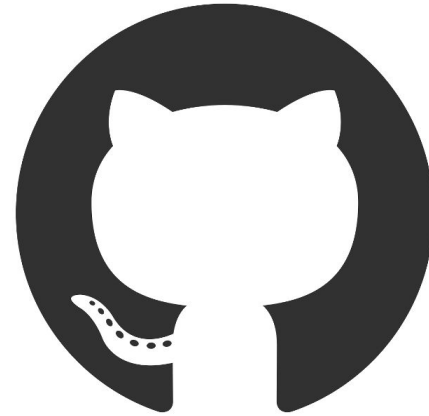


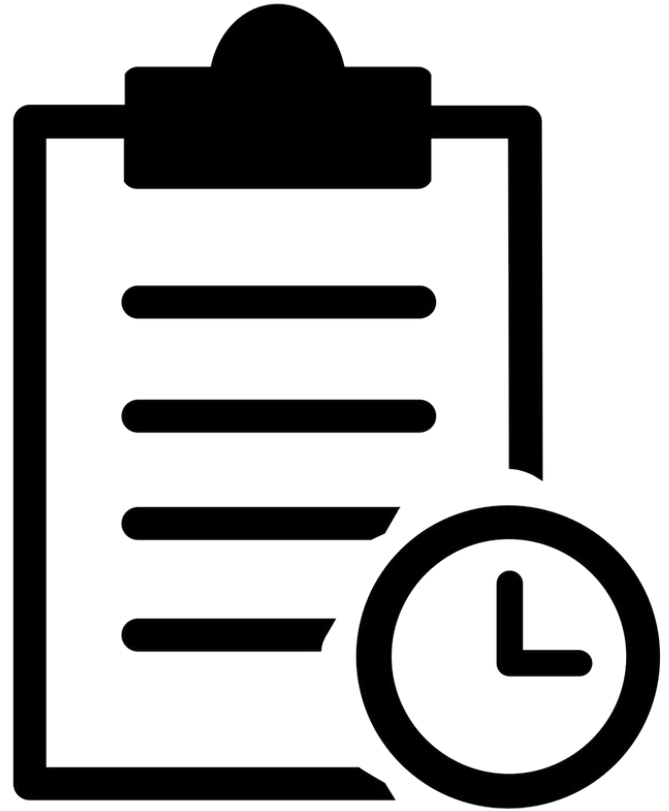
GITHUB



**Universidad Distrital Francisco José de Caldas**  
**Yohan Almonacid Ortiz**

# AGENDA

- ¿Que es GITHUB?
- Sistema de control de versiones
- Primeros Pasos
- Comandos básicos
- Comandos ubuntu
- Más Comandos
- Comandos Ramas
- Comandos Remote
- Entorno gráfico y registro



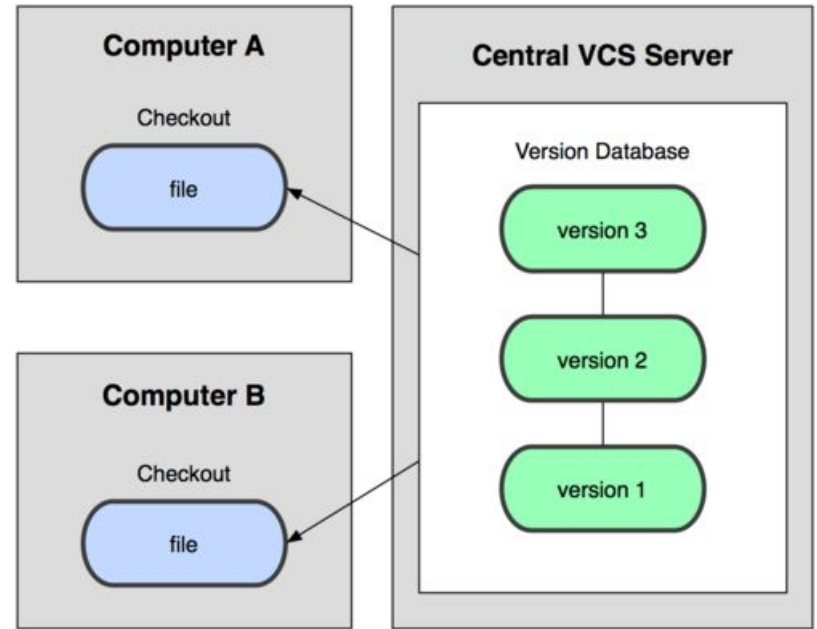
# ¿QUÉ ES GITHUB?

Es un software o sistema abierto distribuido a los usuarios de forma libre, diseñado para para **tener un control de versiones, desde pequeños proyectos hasta los más grandes con rapidez y eficacia.**

Diseñado por Linus Torvalds, creador de Linux, pensado para el mantenimiento de software.

# VCS (SISTEMA DE CONTROL DE VERSIONES)

Software que administra el acceso a un conjunto de ficheros, y mantiene un historial de cambios realizados. El control de versiones es útil para guardar cualquier documento que cambie con frecuencia, como código fuente, documentación o ficheros de configuración.



# CARACTERÍSTICAS

- Fuerte apoyo en el avance del desarrollo de software debido al uso de ramas.
- Gestión eficiente para grandes proyectos donde se requiere el trabajo simultáneo de los integrantes.
- Es fácil de aprender y su uso es sencillo.
- Rendimiento increíblemente rápido.



# COMANDOS BÁSICOS

- Configuración local de git:

**git config -global user.name “nombre”**

**git config -global user.email “email”**

- Comando para volver un fichero un repositorio:

**git init**

- Comando para clonar (descargar) un repositorio:

**git clone [dirección internet, dirección local]**

# COMANDOS BÁSICOS

- Comando para agregar un archivo al staging area:

**git add [archivo]**

- Comando para agregar todos los archivos en el fichero en el que se encuentra, al staging area:

**git add .**

- Comando para verificar el estado del repositorio y la información en el staging area:

**git status**

# COMANDOS UBUNTU

- Comando para crear un archivo:

**touch [nombre con extensión del archivo]**

- Comando para borrar un archivo o fichero:

**rm [nombre con extensión del archivo]**

- Comando para mover un archivo o fichero:

**mv [nombre con extensión del archivo]**



Working  
Directory

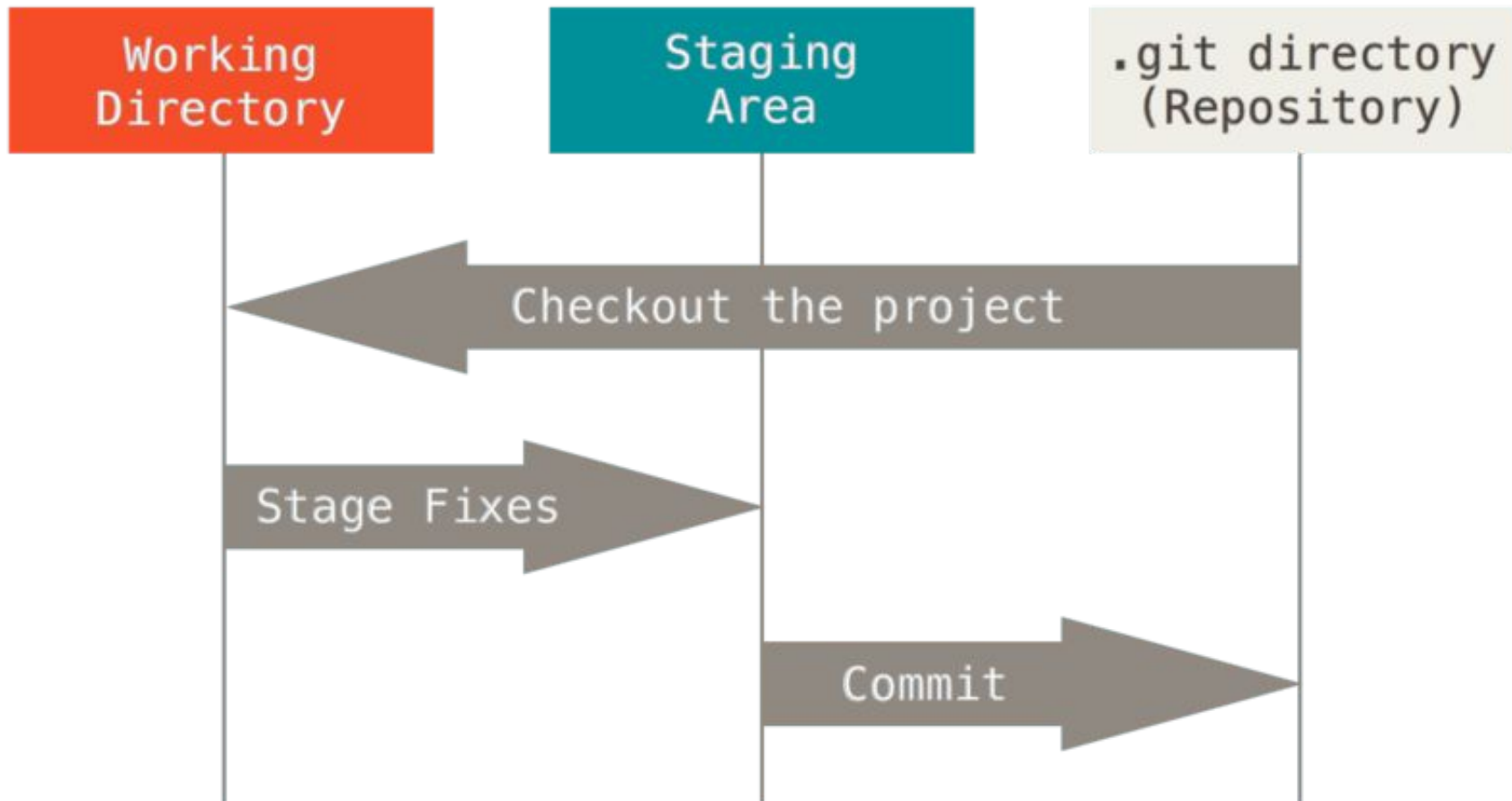
Staging  
Area

.git directory  
(Repository)

Checkout the project

Stage Fixes

Commit



# MÁS COMANDOS

- Comando para enviar una modificación con un comentario o etiqueta, definida commit:  
**git commit**
- Comando para definir qué archivo se quiere borrar:  
**git rm [nombre con extensión del archivo]**
- Comando para borrar un archivo del staging area:  
**git reset HEAD [nombre con extensión del archivo]**
- Comando para borrar un cambio en un archivo, volver a la versión anterior:  
**git checkout [nombre con extensión del archivo]**

# MÁS COMANDOS

- Comando para mostrar todos los commit realizados:

**git log**

- Comando para mostrar todos los commit realizados, sin detalles, solo una línea (título):

**git log --oneline**

# COMANDOS RAMAS

- Comando para mostrar todas las ramas en un repositorio:

**git branch**

- Comando para crear una nueva rama:

**git branch [nombre-rama]**

- Comando para crear una rama y cambiar a esta:

**git checkout -b [nombre-rama]**

- Comando para cambiar a una rama:

**git checkout [nombre-rama]**

# COMANDOS RAMAS

- Comandos para mezclar los cambios en dos ramas:

**git checkout [rama]**      - > posicionarnos en la rama que queremos mezclar otra

**git merge [rama]**              - > rama a mezclar

- Comandos para eliminar una rama:

**git branch -d [nombre-rama]**

- Comandos para ver un commit en particular y volver a esta versión:

**git checkout [# commit]**

# COMANDOS RAMAS

- Comandos para asignar un TAG a un commit:

**git tag [nombre-tag] [# commit]**

Con esto se puede asignar un TAG “Versión 2.0” por ejemplo, a un commit después de haber generado un cambio importante en el proyecto y acceder a él si se desea volver a un punto anterior de estabilidad.

# COMANDOS REMOTE

- Comandos para asociar un repositorio remoto a nuestro directorio local:

**git remote add [nom-repo-local] [dir-repo-remoto]**

- Comando para definir que rama tendrá acceso a dicho repositorio:

**git push [nom-repo-local] [nom-rama]**

- Comando para actualizar el repositorio remoto:

**git pull [nom-repo-local] [nom-rama]**

# ENTORNO GRÁFICO

Principalmente GitHub es un entorno gráfico que nos permite realizar las mismas acciones que realizamos en consola solo que por medio de una interfaz gráfica de usuario. Podemos realizar el control de versiones, creación, clonación y edición de repositorios.



[Pull requests](#)[Issues](#)[Marketplace](#)[Explore](#)

## Repositories

[oscarTopinOne/Gestion-Tecnologica](#)[Yoh-Ortiz/Ciencias3](#)[Yoh-Ortiz/pila\\_colas](#)[danielvargas97/Ejercicios-Haskell](#)[danielvargas97/Modelos-II](#)[danielvargas97/RepasoPOO](#)[Yoh-Ortiz/Modelos2](#)

## Learn Git and GitHub without any code!

Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.

[Read the guide](#)[Start a project](#)

## Discover interesting projects and people to populate your personal news feed.

Your news feed helps you keep up with recent activity on repositories you [watch](#) and people you [follow](#).

Signed in as **Yoh-Ortiz**

[Set status](#)[Your profile](#)[Your repositories](#)[Your projects](#)[Your stars](#)[Your gists](#)[Help](#)[Settings](#)[Sign out](#)**GitHub Packages**

Sign up for the new experience

Welcome to the new GitHub  
to the stuff you care about

### Discover repositories

[aws/aws-sdk-java-v2](#)

The official AWS SDK for Java

Java ★ 508

[GTNewHorizons/NewHorizons](#)

New Modpack with Gregtech and Thaumcraft

JavaScript ★ 122

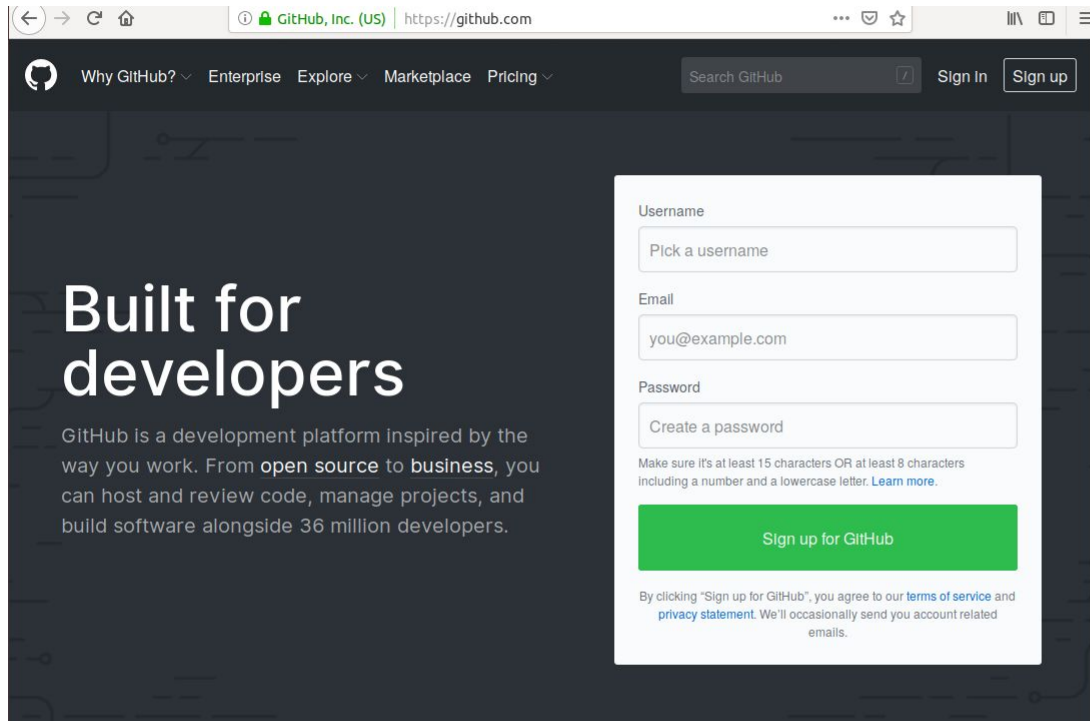
[MariaDB/server](#)

MariaDB server is a community developed fork of MySQL server. Started by core members of the original MySQL team, MariaDB actively works with outside developers to deliver the most featureful, stable...

C++ ★ 2.7k

[Go to Explore](#)

# REGISTRO GITHUB



A screenshot of the GitHub website's sign-up page. The browser address bar shows 'https://github.com'. The navigation bar includes links for 'Why GitHub?', 'Enterprise', 'Explore', 'Marketplace', and 'Pricing', along with a search bar and 'Sign in' and 'Sign up' buttons. The main content area features the GitHub logo and the heading 'Built for developers'. Below this, a paragraph describes GitHub as a development platform. On the right, a white sign-up form is overlaid on the dark background. The form contains fields for 'Username' (with placeholder 'Pick a username'), 'Email' (with placeholder 'you@example.com'), and 'Password' (with placeholder 'Create a password'). Below the password field, there is a note about password requirements and a link to 'Learn more'. A large green button labeled 'Sign up for GitHub' is positioned below the form. At the bottom of the form, a disclaimer states that by signing up, the user agrees to the terms of service and privacy statement.

Why GitHub? ▾ Enterprise Explore ▾ Marketplace Pricing ▾ Search GitHub / Sign in Sign up

## Built for developers

GitHub is a development platform inspired by the way you work. From **open source** to **business**, you can host and review code, manage projects, and build software alongside 36 million developers.

Username

Email

Password

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more.](#)

[Sign up for GitHub](#)

By clicking "Sign up for GitHub", you agree to our [terms of service](#) and [privacy statement](#). We'll occasionally send you account related emails.

# REFERENCIAS

- Notas en clase Modelos de programación 2, Docente Alejandro Daza.
- <http://conociendogithub.readthedocs.io/en/latest/data/introduccion/>
- Video explicativo “*GitHub - Crear un repositorio, validar cambios y sincronizar*” : <https://youtu.be/E3usf4UdkAI>