

# Bootcamp de Full Stack

Bienvenidos a la clase N°09

# Fundamentos

## GIT

### ¿Qué es Git?

Es un sistema de control de versiones distribuido, lo cual significa que se realiza una copia completa del repositorio que contiene la base de código en tu computadora (y en la de todos los demás).

Fuente: [https://developer.mozilla.org/es/docs/Learn/Tools\\_and\\_testing/GitHub](https://developer.mozilla.org/es/docs/Learn/Tools_and_testing/GitHub)



# Fundamentos

## GIT

### ¿Qué es GitHub?

Es un servicio web que permite alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de ordenador.

Fuente: <https://es.wikipedia.org/wiki/GitHub>



# Fundamentos

## GIT

### ¿Qué es un repositorio?

Un repositorio (en inglés Repository) es en donde Git almacena los archivos de tu proyecto y las distintas versiones de tus archivos. Un repositorio puede ser local o remoto.

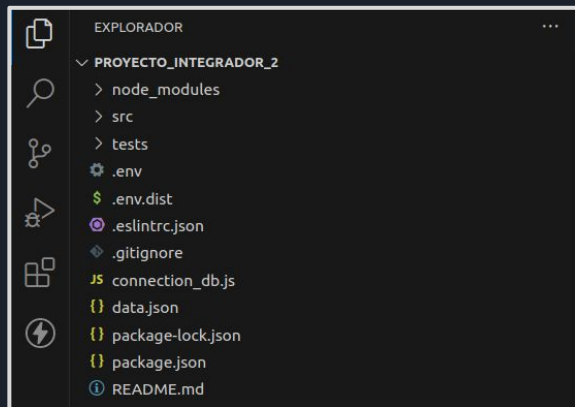
- Un repositorio local se guarda únicamente dentro de tu computadora.
- Un repositorio remoto se guarda en el servidor de un servicio web (Github, gitLab, etc)

# Fundamentos

## GIT

### ¿Qué es un directorio de trabajo?

Un directorio de trabajo (en inglés working directory) es el directorio del proyecto en donde están sus respectivos archivos y carpetas.

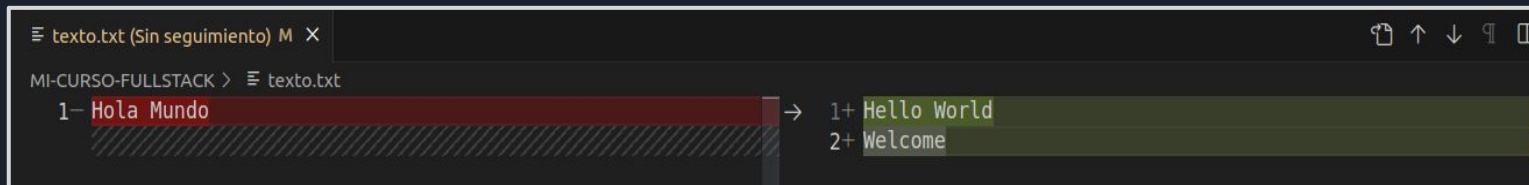


# Fundamentos

## GIT

### ¿Qué es el área de preparación?

El área de preparación (en inglés staging area) es el conjunto de nuevos archivos o archivos modificados que serán incluidos en un commit.



```
MI-CURSO-FULLSTACK > texto.txt
1- Hola Mundo
    → 1+ Hello World
      2+ Welcome
```

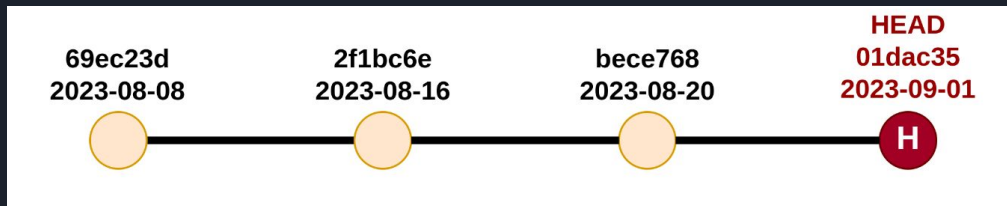
# Fundamentos

## GIT

### ¿Qué es una confirmación?

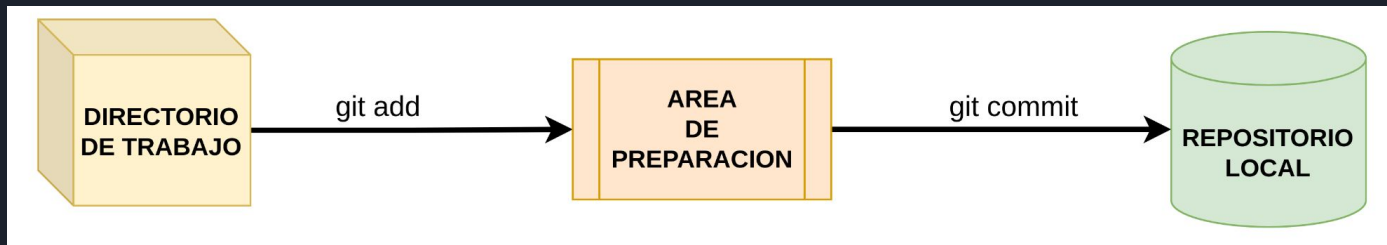
Una confirmación (en inglés commit) es como una foto (snapshot) de tu proyecto en un momento en particular.

Las confirmaciones son el conjunto de fotos (snapshots) del proyecto. En el caso del cabezal (HEAD), siempre apunta a la última confirmación del proyecto.



Crear un repositorio local:

1. Inicializar: **git init**
2. Crear y/o modificar archivos y/o carpetas en el directorio de trabajo
3. Agregar los cambios en el área de staging: **git add** .
4. Confirmar los cambios en el repositorio local: **git commit -m "descripción de commit"**



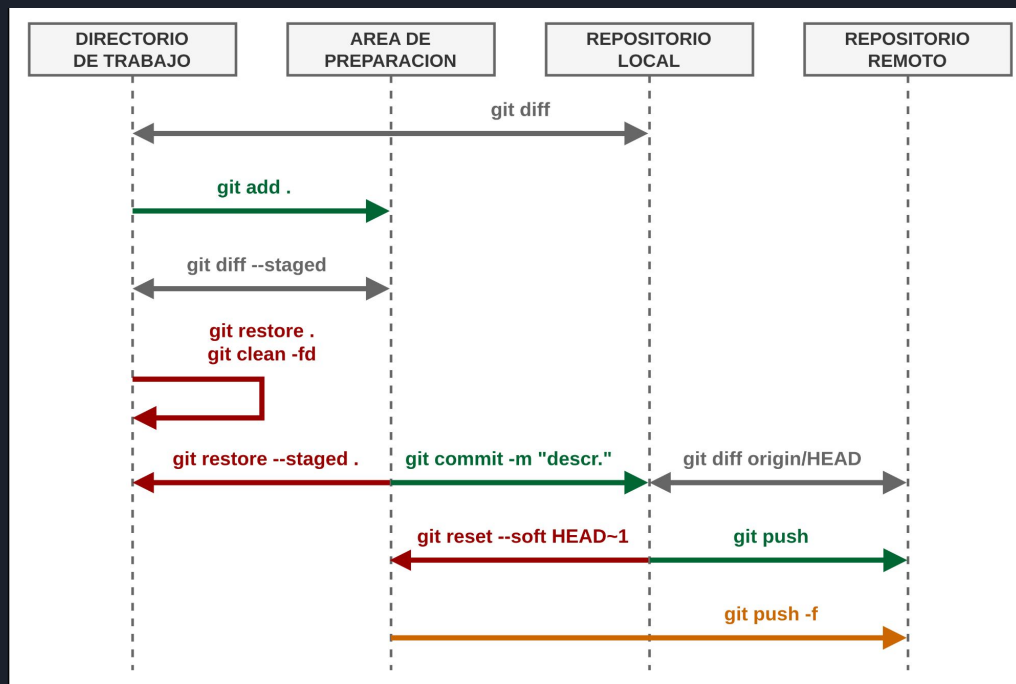


## Sincronizar Repositorios

1. Crear un repositorio en GitHub
  - a. Ir a la pestaña “Repositories”
  - b. Hacer clic en el botón “New”
2. Crear Personal Access Tokens
  - a. Ir a mi perfil -> Settings -> Developer Settings -> Personal access tokens (classic)
  - b. Hacer clic en el botón “Generate new token (classic)”
3. Agregar el token en el entorno local
  - a. `git config --global user.password "your token"`
  - b. `git config --global credential.helper store`
4. Agregar repo remoto: **`git remote add origin https://github.com/mi-repo.git`**
5. Subir confirmación al repositorio remoto por primera vez: **`git push -u origin main`**

1. Clonar localmente un repositorio remoto: `git clone https://github.com/mi-repo.git`
2. Traer última versión: `git pull`
3. Consultar historial de versiones del repositorio local: `git log --oneline`
4. Consultar el estado actual del repositorio local: `git status`
5. Observar cambios entre versiones: `git diff` | `git diff --staged` | `git diff origin/HEAD`
6. Agregar cambios en el área de staging: `git add .`
7. Quitar cambios del área de staging: `git restore .` | `git restore --staged .`
8. Confirmar los cambios en el repositorio local: `git commit -m "descripción de commit"`
9. Deshacer la última confirmación en el repositorio local: `git reset --soft HEAD~1`
10. Aplicar la confirmación desecha en el repositorio remoto: `git push -f`

## Comandos básicos



# BREAK

Descansemos 15 minutos



# Repaso

## HTML & CSS



### Temas

1. Elementos semánticos HTML
2. Selectores CSS
3. Imágenes
4. Formularios
5. Marcos Web
6. Audios & Videos
7. Unidades de medidas em rem vh y vw
8. Contenedores flexibles
9. Medias Queries

# CIERRE DE CLASE

Continuaremos en la próxima clase

