

# Bootcamp de Full Stack

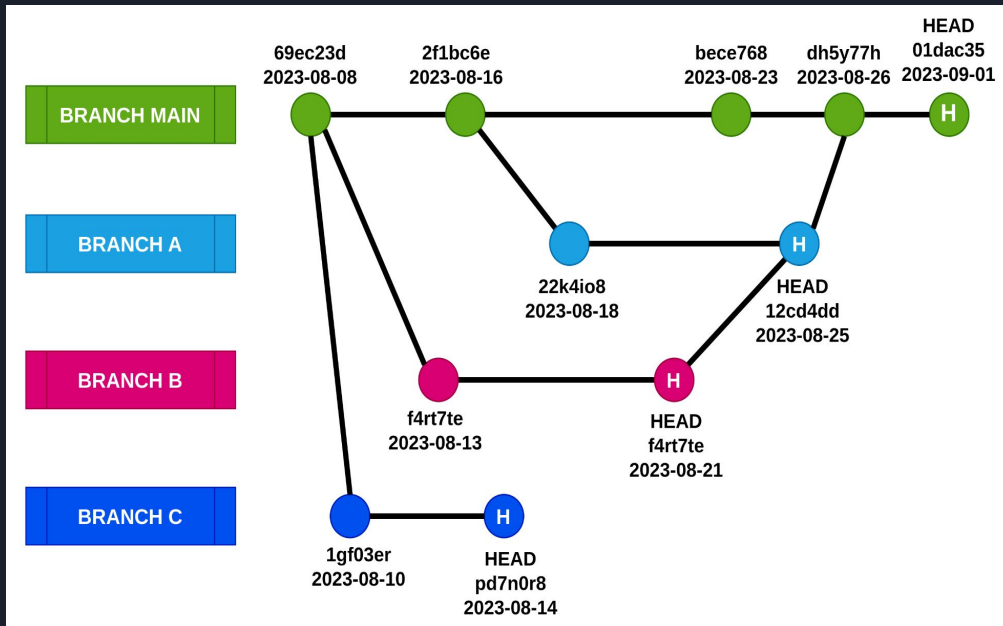
Bienvenidos a la clase N°10

- Branches en Git
- CI/CD
- Deploy en Vercel
- Espacio de consultas

# Fundamentos GIT

## ¿Qué es una rama?

Una rama representa una línea independiente de desarrollo, es como crear un nuevo área de trabajo con su área de ensayo y su historial. Los nuevos commits se registran de forma independiente en el historial la rama, lo que se conoce como forks en el historial del proyecto.



## Comandos básicos

1. Listar ramas (rama actual figura con un asterisco): **git branch**
2. Crear una rama en el repositorio local: **git branch nombre-rama**
3. Ir a la rama creada en el repositorio local: **git checkout nombre-rama**
4. Sincronizar la rama creada en el repositorio remoto: **git push -u origin nombre-rama**
5. Ir a la rama principal en el repositorio local: **git checkout main**
6. Fusionar la rama principal con una creada en el repositorio local: **git merge nombre-rama**
7. Sincronizar la rama fusionada en el repositorio remoto: **git push**
8. Eliminar rama en el repositorio local: **git branch -D nombre-rama**
9. Eliminar rama en el repositorio remoto: **git push origin --delete nombre-rama**

# Fundamentos

## Deployment

### ¿Qué es deploy de software ?

Un despliegue (deploy en inglés) se refiere al proceso de colocar un producto de software que se ha desarrollado en un entorno de producción.

- **Entorno de desarrollo:** Es en donde se construye el código fuente del software. Es accesible para los desarrolladores.
- **Entorno de prueba:** Es en donde se prueban las nuevas características del software. Es accesible para los desarrolladores y testadores.
- **Entorno de producción:** Es en donde se aloja el código fuente del software. Es accesible para los usuarios del sistema.

# Fundamentos

## Deployment

### ¿Qué es CI/CD?

generalmente refiere a las prácticas combinadas de integración continua y entrega continua (también conocida como despliegue continuo).

- La **integración continua** (continuous integration en inglés) es una práctica de ingeniería de software que consiste en hacer integraciones automáticas de un proyecto lo más a menudo posible para así poder detectar fallos cuanto antes.
- La **entrega continua** (continuous delivery en inglés) es un enfoque de la ingeniería del software en que los equipos de desarrollo producen software en ciclos cortos, asegurando que el software puede ser liberado en cualquier momento (...) Apunta a la construcción, prueba, y liberación del software de forma más rápida y más frecuente.

# Deployment

## Git & Vercel

1. Utilizar el repositorio creado en la clase anterior para el sitio web "The Devs Bands".
2. Crear un repositorio en GitHub <https://github.com/new>
3. Clonar el repositorio localmente: git clone <https://github.com/usuario/nombre-del-repo.git>
4. Acceder a la carpeta del proyecto para verificar su contenido: cd nombre-del-repo
5. Configurar Vercel siguiendo la guía:  
[https://docs.google.com/document/d/1ufktlxfPfinl95wD6-UAbic1AylyEV97zTngDJ5X\\_BQ/edit?usp=sharing](https://docs.google.com/document/d/1ufktlxfPfinl95wD6-UAbic1AylyEV97zTngDJ5X_BQ/edit?usp=sharing)
6. Modificar el proyecto según sea necesario y subirlos a GitHub.
7. Desplegar el proyecto en Vercel

# BREAK

Descansemos 10 minutos



# Repaso

## Espacio de Consultas



### 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

