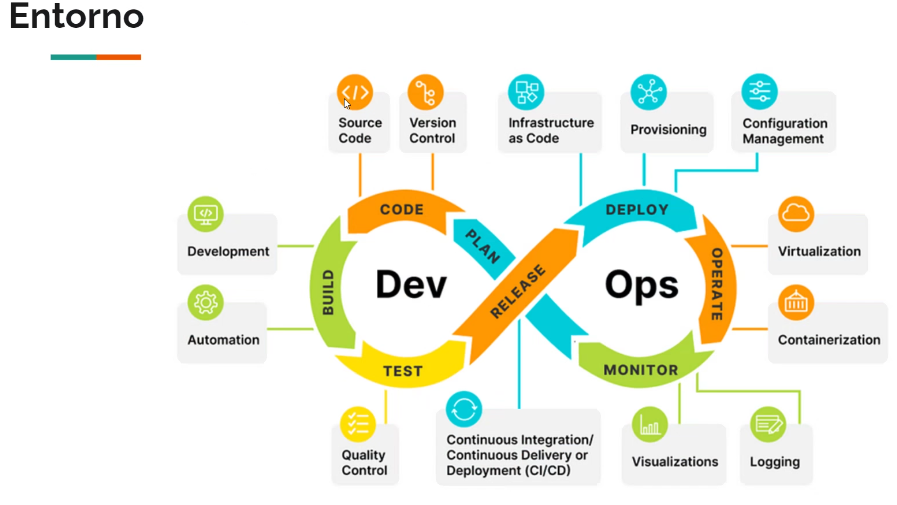
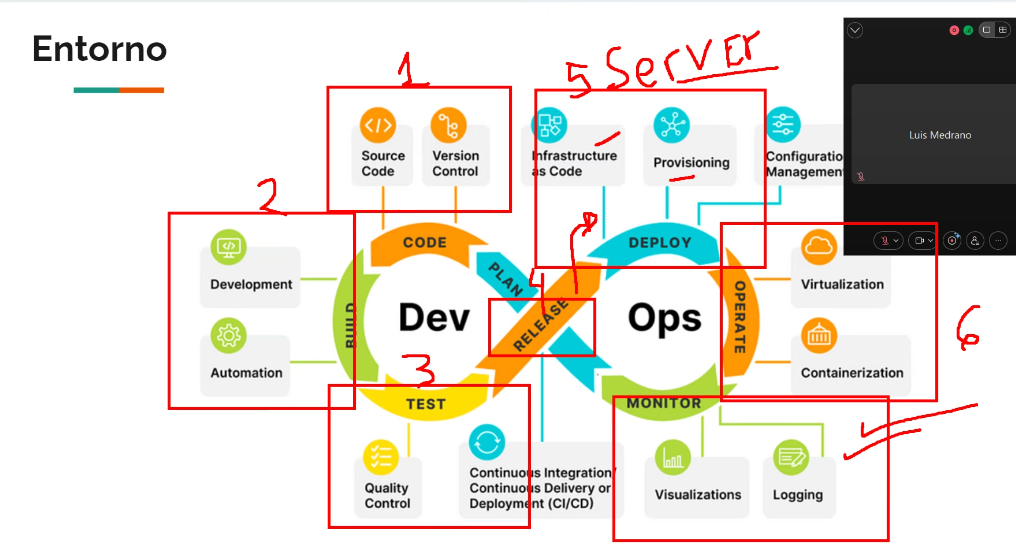
**Clase: 30/09/2024**



Estas etapas van ligadas a etapas de metodologías agiles.



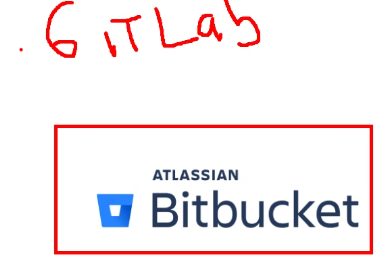
Formas DevOps:

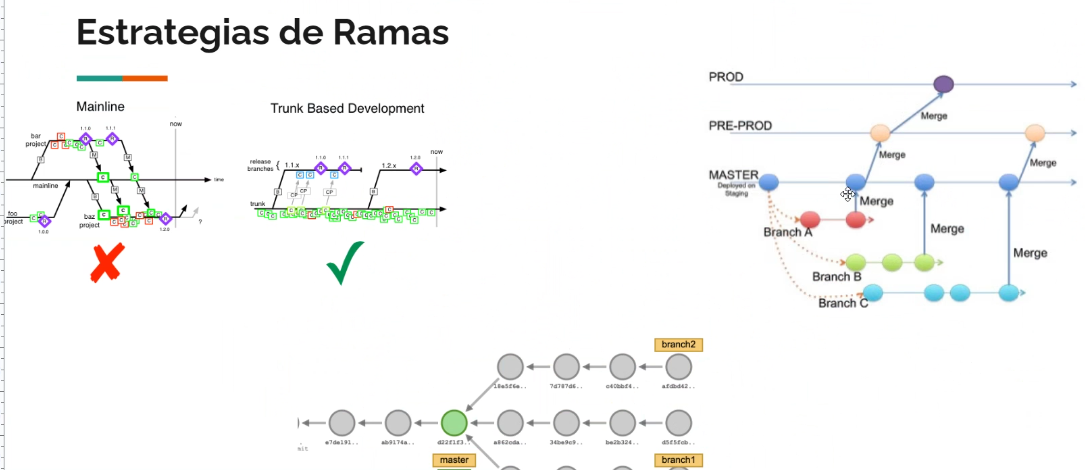
1. Me acoplo a estas etapas ya construidas.
2. Construyo cada etapa

**Etapa 01 – CÓDIGO**

* Elegir el lenguaje: JAVA, PHP, JS….
* Herramienta para gestionar el versionado del código: GIT

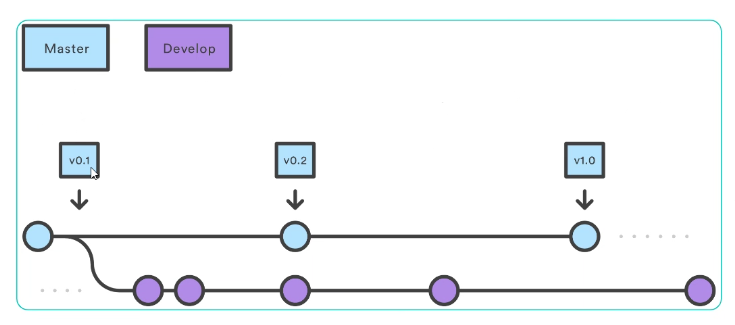


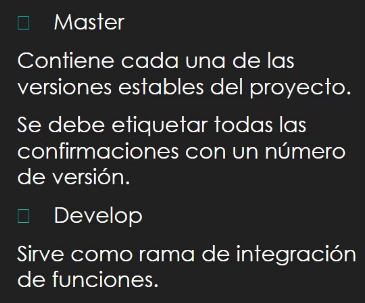


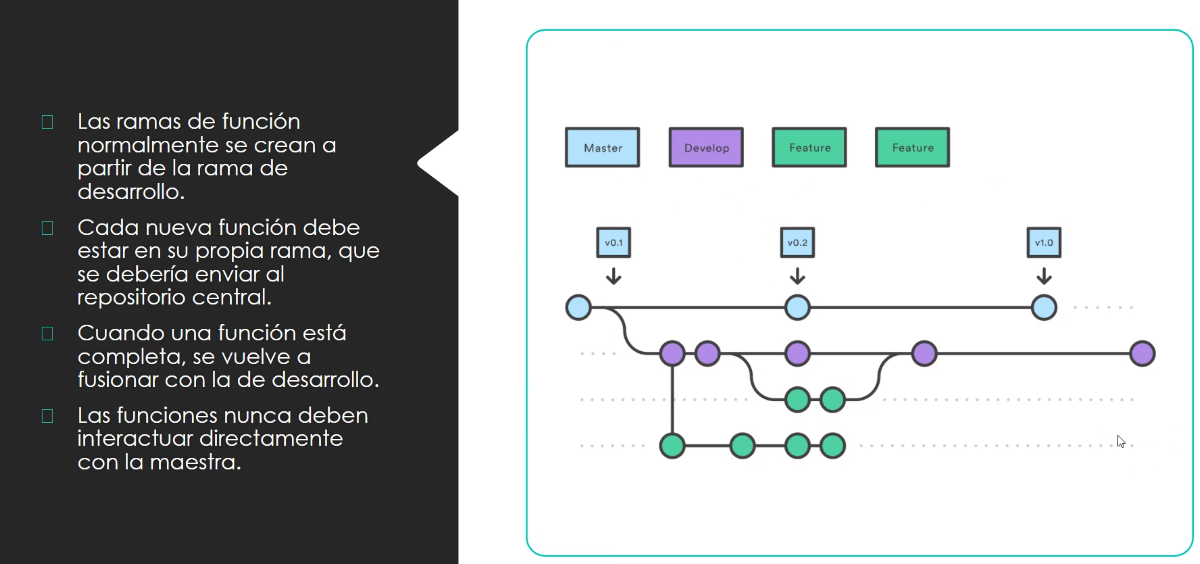


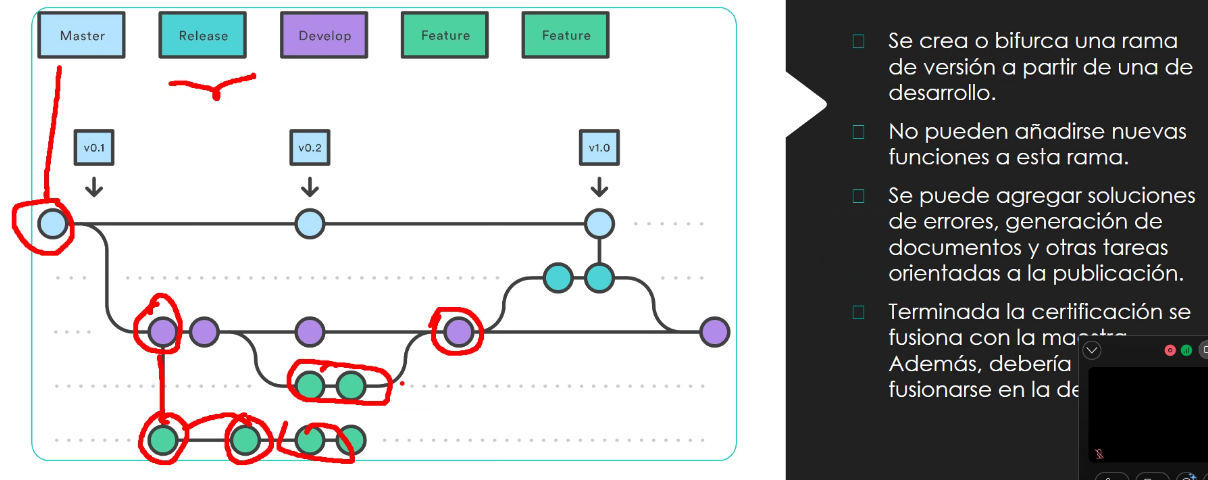
* Cada ambiente es el reflejo de una rama, o cada rama es un ambiente: Producción, Pre Producción, …
* Definir los aprobadores de cada rama.

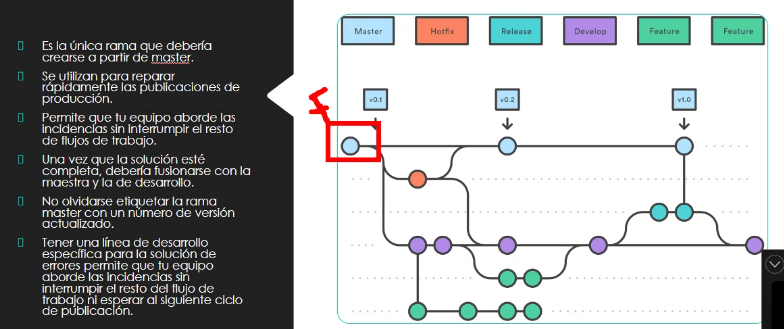
**Estrategia de ramas: Git-Flow**



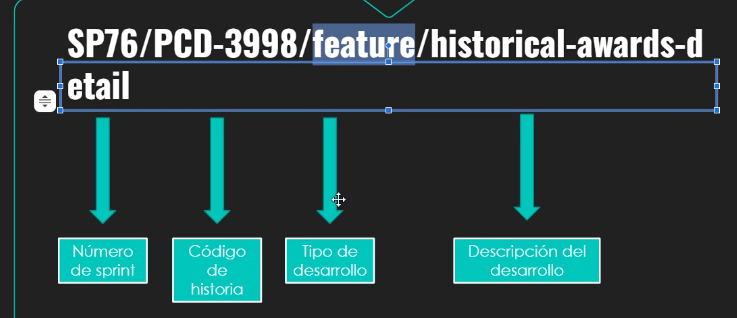


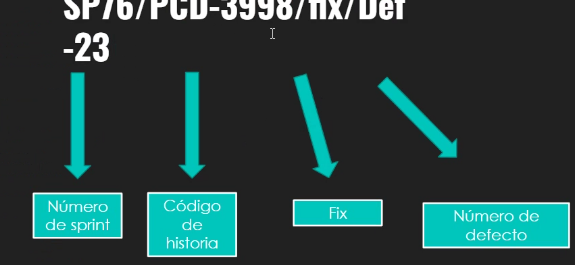


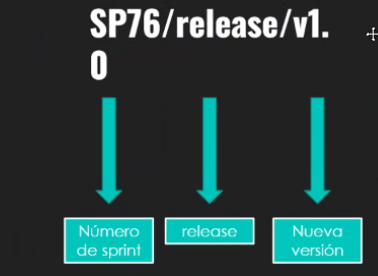




Como llamar a las ramas:







EJEMPLO DE GIT:

PARA CREAR RAMAS:

# Git checkout –b desarrollo 🡪 cambia a la rama, sino, la crea

# git branch 🡪 para ver las ramas

FUSIONAR LAS RAMAS (MERGUEO)

# git merge …

//////////// Enlazar GitHub con Local ///////////////////////////////////

git add .

git commit -m "Actualiza Paul Giraldo"

git branch -M main

git remote add origin https://github.com/paulgiraldo/Equipo\_Trabajo\_2.git

git push -u origin main

//////////////////////////////////////////////

ENLAZAR UNA RAMA REMOTA CON UN RAMA LOCAL

# git fetch origin Paul:Paul

# git checkout Paul

------ REALIZAR LOS CAMBIOS—

# git add

# git commit

# git push origin Paul -- con esto se actualiza

**Clase: 02/10/2024**

**Plan:** Entender el proyecto

**Code:** Involucra al equipo de desarrollo, lenguaje, tipo de solución, estrategia de brantcheo…

**Build:** Consideras si tu lenguaje se compila, dependencias externas… lo necesario para que funcione bien.

**Test:** Pruebas de testeo para el código… QA…

**Release:** Construcción de La versión estable, tener un repo de versiones estables, etiquetar, versionar, una repo.

**Deploy:** a donde enviamos el reléase: QA, Pre, Prod., que restricciones antes de producción, aprobaciones previas.

La Infraestrutura depende del ***Deploy*** (por los ambientes) y también de ***Operate*** (porque definimos a dónde va el artefacto)

**Operate:** La infraestrutura donde va a operar el software.

**Monitor:** Logs de todos, alarmas, alertas de fallos…herramientas visuales, para toma de decisiones. Se puede retroceder a ***Operate*** para incrementar algo. El monitoreo también puede ser de otros ambientes como QA, Pre..