

Taller 1: Construcción de pipelines en Cloud

En este ejercicio debe realizar la construcción de un pipeline para el código disponible en:

<https://github.com/bortizf/microservice-app-example>

Para este proyecto, debe trabajar tanto del lado de desarrollo como de operaciones, considerando los diferentes aspectos para que el proyecto pueda ser utilizado por un equipo ágil. Inicialmente, deben escoger la metodología ágil a utilizar.

Aspectos a considerar para el desarrollo del taller:

1. 2.5% Estrategia de branching para desarrolladores

R//

La estrategia que proponemos para el equipo de desarrollo es Git Flow, un enfoque estructurado y ampliamente utilizado en equipos ágiles.

Estructura de ramas:

Rama :	Propósito
main :	Contiene código estable y listo para producción.
develop :	Contiene el código en estado estable para pruebas, con nuevas features integradas.
feature/* :	Ramas para el desarrollo de nuevas funcionalidades.
release/* :	Preparación de nuevas versiones a producción (pruebas finales, ajustes).
hotfix/* :	Corrección rápida de bugs en producción directamente sobre main.

Flujo de trabajo:

- Los desarrolladores crean ramas feature/nombre-funcionalidad desde develop.
- Una vez completada la feature, se hace un pull request (PR) a develop.
- Cuando varias funcionalidades están listas, se crea una rama release/x.x.x desde develop.
- Tras pruebas, release/x.x.x se fusiona a main y también a develop.
- Si hay bugs críticos en producción, se crea una rama hotfix/x.x.x desde main.

2. 2.5% Estrategia de branching para operaciones

R//

Para el equipo de operaciones, se define una estrategia de branching que permita mantener la infraestructura como código organizada, trazable y sincronizada con el desarrollo del sistema.

Enfoque: Infraestructura como Código (IaC)

Se utilizará una rama dedicada y una estructura de carpetas claras dentro del repositorio para gestionar scripts de infraestructura, pipelines y configuraciones relacionadas con despliegue.

Ramas utilizadas:

Rama	Propósito
• infra	Rama dedicada para los archivos de infraestructura (IaC, Jenkins, Azure).
• infra/dev	Rama para pruebas y cambios preliminares en la infraestructura.
• infra/prod	Rama protegida para despliegue en producción.

3. 15.0% Patrones de diseño de nube (mínimo dos)

R//

4. 15.0% Diagrama de arquitectura

R//

5. 15.0% Pipelines de desarrollo (incluidos los scripts para las tareas que lo necesiten)

6. 5.0% Pipelines de infraestructura (incluidos los scripts para las tareas que lo necesiten)

7. 20.0% Implementación de la infraestructura

8. 15.0% Demostración en vivo de cambios en el pipeline

9. 10.0% Entrega de los resultados: debe incluir la documentación necesaria para todos los elementos desarrollados.

Presentación: 23 de abril. 8 minutos por grupo.