

Ciclo completo de CI/CD con Dagger y Kubernetes

Grado en Ingeniería Informática
Universidad de Santiago de Compostela

Autor: Daniel Vieites Torres

Tutor: Juan Carlos Pichel Campos
Co-tutor: Francisco Maseda Muiño

28 de agosto de 2025

Tabla de contenidos

Objetivos

Fundamentos teóricos

Conceptos y herramientas

Dagger

¿Cuál es el problema?

- Aplicaciones con un gran volumen de tecnologías.

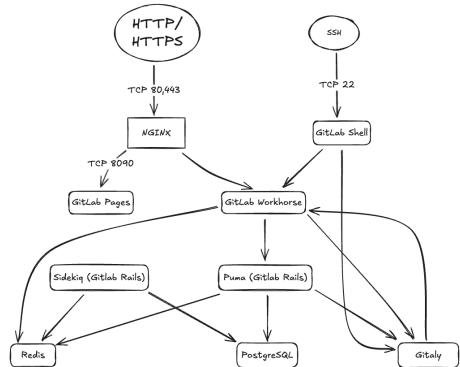


Figura: Arquitectura de GitLab.

¿Cuál es el problema?

- ▶ Aplicaciones con un gran volumen de tecnologías.
- ▶ Gran complejidad.

```
jobs:
  cicd:
    runs-on: ubuntu-24.04
    steps:
      - name: Checkout zoo repository
        uses: actions/checkout@v4
        with:
          path: zoo
```

```
dir: / -> +9000
dir: /.gitlab/ci -> ~70
```

Figura: Archivos YAML en GitLab.

¿Cuál es el problema?

- ▶ Aplicaciones con un gran volumen de tecnologías.
- ▶ Gran complejidad.
- ▶ Coste de mantenimiento elevado.

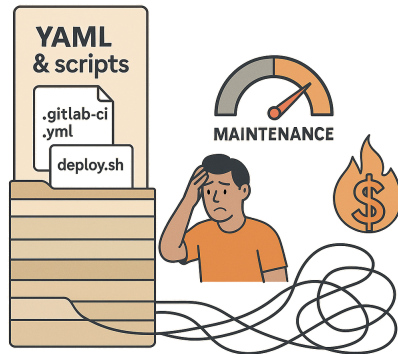


Figura: Imagen generada con IA.

¿Cuál es el problema?

- ▶ Aplicaciones con un gran volumen de tecnologías.
- ▶ Gran complejidad.
- ▶ Coste de mantenimiento elevado.
- ▶ Baja portabilidad.



¿Qué se propone?

Dagger para gestionar CI/CD.



CI/CD

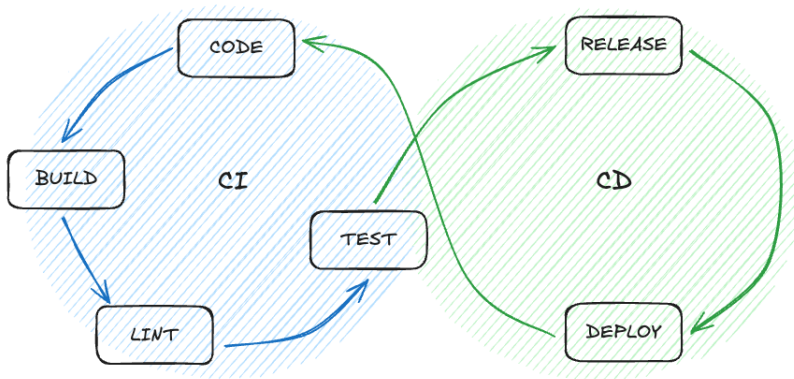


Figura: Ciclo de CI/CD de este TFG.

GitOps & ArgoCD

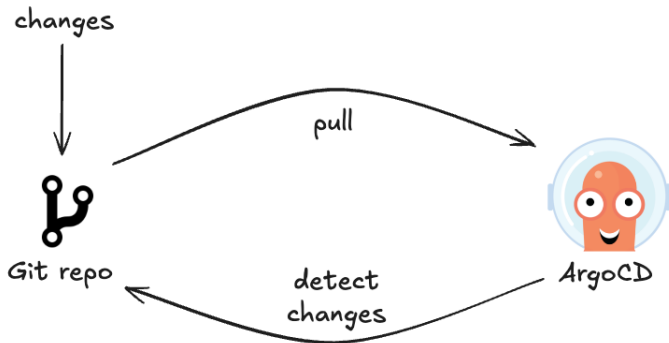


Figura: Funcionamiento de ArgoCD con GitOps.

Kubernetes & Helm

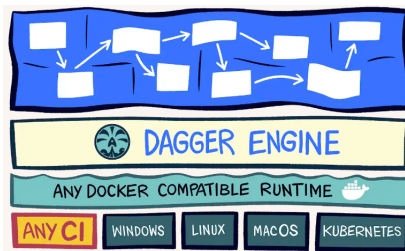


&



Dagger

- ▶ SDK de creación de *pipelines* de CI/CD.



Dagger

- ▶ SDK de creación de *pipelines* de CI/CD.
- ▶ Múltiples lenguajes.



Dagger

- ▶ SDK de creación de *pipelines* de CI/CD.
- ▶ Múltiples lenguajes.
- ▶ Módulos.



Dagger

- ▶ SDK de creación de *pipelines* de CI/CD.
- ▶ Múltiples lenguajes.
- ▶ Módulos.
- ▶ *Runtime* de OCI.



Dagger

- ▶ SDK de creación de *pipelines* de CI/CD.
- ▶ Múltiples lenguajes.
- ▶ Módulos.
- ▶ *Runtime* de OCI.
- ▶ Uso de caché.

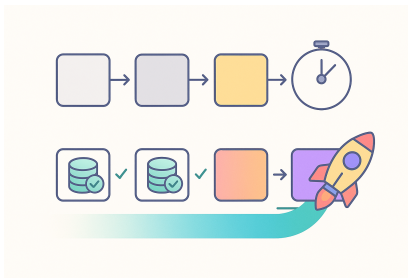


Figura: Imagen generada con IA.