

# Master Cloud Apps

## Integración y entrega continua

### Configuración de un pipeline de integración continua

#### Asunciones

- Se usa **GitFlow** como modelo de desarrollo.
- Se crea un sistema de integración continua usando las herramientas de CI **GitHub Actions**. <https://github.com/franco87/fj.franco.2019-CI>
- Se utiliza el repositorio **GitHub Packages** para almacenar artefactos java que se generan con cada **Pull Request**.
- El artefacto de entrega es una imagen **Docker** que se almacena en el Registry **DockerHub**. <https://hub.docker.com/r/franco87/practica-ci>
- Se utiliza **SonarCloud** para ejecutar tests de calidad. [https://sonarcloud.io/project/issues?id=franco87\\_fj.franco.2019-CI&resolved=false](https://sonarcloud.io/project/issues?id=franco87_fj.franco.2019-CI&resolved=false)

#### Workflows

- **build-develop-nightly.yml**: Ejecuta una build completa cada noche y publica la imagen Docker en el repositorio.
- **build-PR.yml**: Ejecuta tests unitarios en cada Pull Request de cualquier rama.
- **periodically-IT.yml**: Ejecuta tests de sistema una vez al día.
- **periodically-sonarcloud.yml**: Ejecuta test de calidad una vez al día.
- **release-master.yml**: Se ejecuta al detectar un tag en master, compila todo el proyecto, construye la imagen Docker y la sube al repositorio DockerHub taggeada con la versión del tag.