



CONSTRUCCIÓN Y EVOLUCIÓN DE SOFTWARE (ISWD633)

NOMBRES ESTUDIANTES: Sebastián Donoso

Ismael Toala

FECHA: 03-08-2025

TEMA: Proyecto 2 Bimestre

Nombre del Proyecto: Automatización de Integración y Despliegue para Aplicación Core-BankEC

Introducción:

El presente documento describe la implementación de un pipeline de integración y despliegue continuo (CI/CD) para una aplicación desarrollada en Python. El pipeline se construyó aplicando buenas prácticas de ingeniería de software, como análisis estático del código, construcción de imágenes Docker y ejecución automatizada al hacer cambios en el repositorio. La aplicación utilizada es un backend llamado *core-bankec-python*, que gestiona funciones bancarias como autenticación, validación y cifrado de datos.

Objetivos:

- Diseñar y construir un pipeline funcional para una aplicación Python.
- Aplicar herramientas de análisis estático de código (flake8).
- Automatizar la construcción de imágenes Docker.
- Asegurar que cada push a la rama main active el flujo de CI/CD.
- Consolidar prácticas de calidad y control de versiones aprendidas en clase.

1. Entregables del Proyecto

1.1. Estructura del Proyecto

Entregable	Descripción
app/	Código Python organizado por módulos
.github/workflows/ci.yml	Archivo de definición del pipeline CI/CD
Dockerfile	Define cómo construir la imagen Docker
requirements.txt	Lista de dependencias del proyecto



CONSTRUCCIÓN Y EVOLUCIÓN DE SOFTWARE (ISWD633)

.env (generado desde secrets)	Archivo de variables de entorno, generado en el pipeline
README.md	Documentación del proyecto

1.2. Construcción del Pipeline

El archivo de configuración ci.yml se ubicó en:

core-bankec-python/.github/workflows/ci.yml

El pipeline se activa automáticamente al hacer push a la rama main y realiza las siguientes etapas:

1. Checkout del código
2. Instalación de Python 3.10
3. Instalación de dependencias del proyecto
4. Ejecución del análisis estático con flake8
5. Construcción de la imagen Docker

ci.yml

```
name: CoreBankEC CI/CD

on:
  push:
    branches: [ main ]

jobs:
  build-test-lint:
    runs-on: ubuntu-latest

    steps:
      - name: 📁 Checkout del código
        uses: actions/checkout@v3

      - name: ⚙️ Configurar Python 3.10
        uses: actions/setup-python@v4
        with:
          python-version: '3.10'

      - name: 🛒 Instalar dependencias
        working-directory: core-bankec-python
```



CONSTRUCCIÓN Y EVOLUCIÓN DE SOFTWARE (ISWD633)

```
run: |
    python -m pip install --upgrade pip
    pip install -r requirements.txt
    pip install flake8

- name: ✅ Análisis estático con flake8
  working-directory: core-bankec-python
  continue-on-error: true
  run: |
    echo "Ejecutando flake8..."
    flake8 app/ --ignore=E501,E302,W293,E261

- name: 🔒 Crear archivo .env con secrets
  working-directory: core-bankec-python
  run: |
    echo "FERNET_KEY=${{ secrets.FERNET_KEY }}" >> .env
    echo "JWT_SECRET_KEY=${{ secrets.JWT_SECRET_KEY }}" >> .env
    echo "JWT_EXPIRATION_HOURS=0.25" >> .env

- name: 🏰 Construir imagen Docker
  working-directory: core-bankec-python
  run: |
    docker build -t core-bankec .
```

1.3. Ejecución Exitosa



CONSTRUCCIÓN Y EVOLUCIÓN DE SOFTWARE (ISWD633)

← CoreBankEC CI/CD

feat: generar .env desde secrets en pipeline CI/CD #4

Summary

Jobs

build-test-lint

Run details

Usage

Workflow file

build-test-lint

succeeded now in 31s

Set up job

Checkout del código

Configurar Python 3.10

Instalar dependencias

Análisis estático con flake8

Crear archivo .env con secrets

Construir imagen Docker

Post Configurar Python 3.10

Post Checkout del código

Complete job

1.4. URL GitHub

URL: <https://github.com/kevinsdonoso/Proyecto-2Bimestre.git>



CONSTRUCCIÓN Y EVOLUCIÓN DE SOFTWARE (ISWD633)

The screenshot shows a GitHub repository page for 'Proyecto-2Bimestre'. The repository has 1 branch and 0 tags. The main branch has 4 commits from 'kevinsdonoso' made 5 minutes ago. The commits are: 'feat: generar .env desde secrets en pipeline CI/CD', 'feat: generar .env desde secrets en pipeline CI/CD', 'feat: subir todos los archivos del Proyecto 2Bimestre con pip...', and 'feat: subir todos los archivos del Proyecto 2Bimestre con pip...'. The README file contains the text 'Examen de desarrollo software seguro by Grupo 1'. The repository has 0 stars, 0 forks, and 0 watching. It includes sections for About, Releases, Packages, and Languages, with Python being the primary language at 99.1%.

Conclusiones

- Se logró automatizar el flujo de integración continua con éxito.
- El pipeline facilita la detección temprana de errores y mejora la calidad del código.
- GitHub Actions resultó ser una herramienta eficaz y sencilla para orquestar tareas de construcción, pruebas y despliegue.
- El proyecto cumple con los objetivos propuestos y refuerza el uso de buenas prácticas de desarrollo profesional.

4. Recomendaciones

- Incluir pruebas unitarias automatizadas en futuras versiones del pipeline.
- Integrar herramientas como SonarCloud para análisis más avanzado.
- Automatizar también el despliegue hacia servicios como Railway o Render.
- Mantener la documentación técnica actualizada dentro del repositorio.