

# Informe de Trabajo Práctico – Azure DevOps CI/CD

## Introducción

El presente trabajo práctico tuvo como objetivo poner en funcionamiento un flujo de **Integración Continua (CI)** en Azure DevOps, utilizando un **agente self-hosted**.

El propósito fue que, frente a cambios en el repositorio, se ejecutaran automáticamente los procesos de **build del frontend (React con Vite)** y del **backend (Go)**, y que se publicaran los artefactos generados para posterior despliegue.

## 1. Tareas a realizar

De acuerdo a las consignas, las tareas principales fueron:

- **Preparación del entorno**
  - Crear pool de agentes *SelfHosted* en Azure DevOps.
  - Registrar el agente local como servicio de Windows (Agent-Local).
- **Estructura del repositorio**
  - Organizar el monorepo con carpetas `/frontend` y `/backend`.
  - Incorporar archivo `azure-pipelines.yml` en la raíz.
- **Definición del pipeline**
  - Un único *stage* de CI, ejecutado sobre la rama `main`.
  - Build del frontend con `npm ci && npm run build`.
  - Build del backend con `go mod tidy`, `go test` y `go build`.
  - Publicación de artefactos (`dist/` y `bin/`).

## 2. ¿Cómo las realizamos?

### 2.1 Agente self-hosted

- Se descargó el instalador desde Azure DevOps.
- Se configuró el agente como servicio en Windows, asociado al pool **SelfHosted**
- Se verificó el estado **online** en la organización.

```
Administrador: Símbolo del sistema - vconfig.cmd
C:\Users\herna\OneDrive\Desktop\Ingesoft3\AgenteAD0>.vconfig.cmd

Azure Pipelines

agent v4.261.0 (commit 4513101)

Security warning! The group BUILTIN\Users has access to write/modify the agent folder. Please examine the log for more details.
>> Connect:
Enter server URL > https://dev.azure.com/hermandezsimon2122
Enter authentication type (press enter for PAT) >
Enter personal access token > *****
Connecting to server ...
>> Register Agent:
Enter agent pool (press enter for default) > SelfHosted
Enter agent name (press enter for LAPTOP-08HNNPLO) > AgenteSimon
Enter replicas? (Y/N) (press enter for N) > y
Scanning for tool capabilities.
Connecting to the server.
Successfully added the agent
Testing agent connection.
Enter work folder (press enter for _work) >
2025-09-22 18:09:10Z: Settings Saved.
Enter run agent as service? (Y/N) (press enter for N) > y
Enter enable SERVICE_SID_TYPE_UNRESTRICTED for agent service (Y/N) (press enter for N) > n
Enter User account to use for the service (press enter for NT AUTHORITY\NETWORK SERVICE) > n
Enter a valid value for User account to use for the service.
Enter User account to use for the service (press enter for NT AUTHORITY\NETWORK SERVICE) >
Agenting file permissions to 'NT AUTHORITY\NETWORK SERVICE'.
Service vstagent.hermandezsimon2122.SelfHosted.AgenteSimon successfully installed
Service vstagent.hermandezsimon2122.SelfHosted.AgenteSimon successfully set recovery option
Service vstagent.hermandezsimon2122.SelfHosted.AgenteSimon successfully set to delayed auto start
Service vstagent.hermandezsimon2122.SelfHosted.AgenteSimon successfully configured
Enter whether to prevent service starting immediately after configuration is finished? (Y/N) (press enter for N) >
```

```
C:\Users\herna\OneDrive\Desktop\Ingesoft3\AgenteAD0>.\run.cmd
Scanning for tool capabilities.
Connecting to the server.
2025-09-25 18:31:55Z: Listening for Jobs
2025-09-25 18:31:57Z: Running job: Build y Publicar Artefactos
2025-09-25 18:33:51Z: Job Build y Publicar Artefactos completed with result: Succeeded
|
```

## 2.2 Repositorio

- Se mantuvo el frontend React en la carpeta /frontend.
- Se organizó el backend (Go) en la carpeta /backend.
- Se agregó el archivo azure-pipelines.yml en la raíz del repo.

## 2.3 Pipeline

El pipeline definido fue:

stages:

- stage: CI

displayName: 'Integración Continua'

jobs:

- job: BuildAndPublish

displayName: 'Build y Publicar Artefactos'

steps:

- task: NodeTool@0

inputs:

versionSpec: '20.19.0'

displayName: 'Instalar Node.js para Front'

- script: |

npm ci && npm run build --verbose

displayName: 'Build Frontend (React)'

workingDirectory: 'frontend'

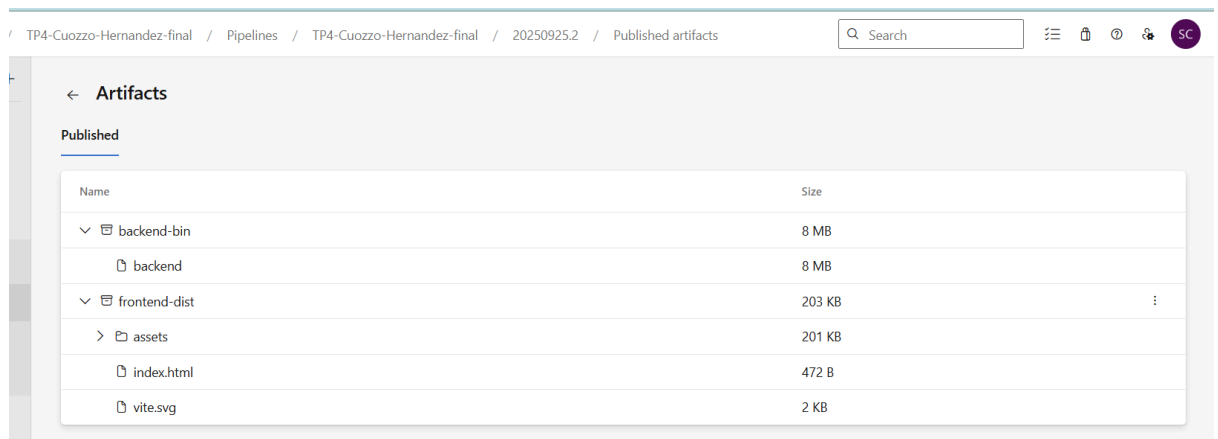
```

- task: PublishBuildArtifacts@1
  inputs:
    PathToPublish: '$(System.DefaultWorkingDirectory)/frontend/dist'
    ArtifactName: 'frontend-dist'
- task: GoTool@0
  inputs:
    version: '1.22.5'
    displayName: 'Instalar Golang'
- script: |
    go mod init backend && go mod tidy && go test ./... && go build -o ./bin/backend
  workingDirectory: 'backend'
  displayName: 'Build Backend (Go)'
- task: PublishBuildArtifacts@1
  inputs:
    PathToPublish: '$(System.DefaultWorkingDirectory)/backend/bin'
    ArtifactName: 'backend-bin'

```

## 2.4 Publicación de artefactos

- Para el frontend: se publicó el contenido de dist/ como artefacto frontend-dist.
- Para el backend: se publicó el binario generado en bin/ como artefacto backend-bin.



## 3. Problemas encontrados y soluciones

- **Error en publicación de artefactos (frontend)**
  - Apareció:  
*“Publishing build artifacts failed with an error: Not found PathToPublish: C:\Users\sofic\Documents\AgenteADO\_work\2\s\frontend\dist”*
  - Esto ocurrió porque, en la primera ejecución, la carpeta dist/ aún no existía.
  - Solución: volver a correr el pipeline tras el build correcto, verificando que vite.config.js genere dist/. Luego el artefacto se publicó sin problemas.

