




🧩 Escenario (actualizado)

Como líder técnico, debes:

1. Elegir una app (o crear una mínima) con ****Front + Back**** (stack libre: Angular/React/Vue + .NET/Node/Java, etc.).
2. Versionar todo en ****un único repo**** (mono-repo recomendado) con carpetas ``/front``, ``/back``.
3. Definir un ****pipeline en YAML**** con ****CI**** (build) que ****corra el pipeline en un agente Self-Hosted**** instalado en tu equipo.

que es yaml:

🧠 ¿Por qué usar YAML?

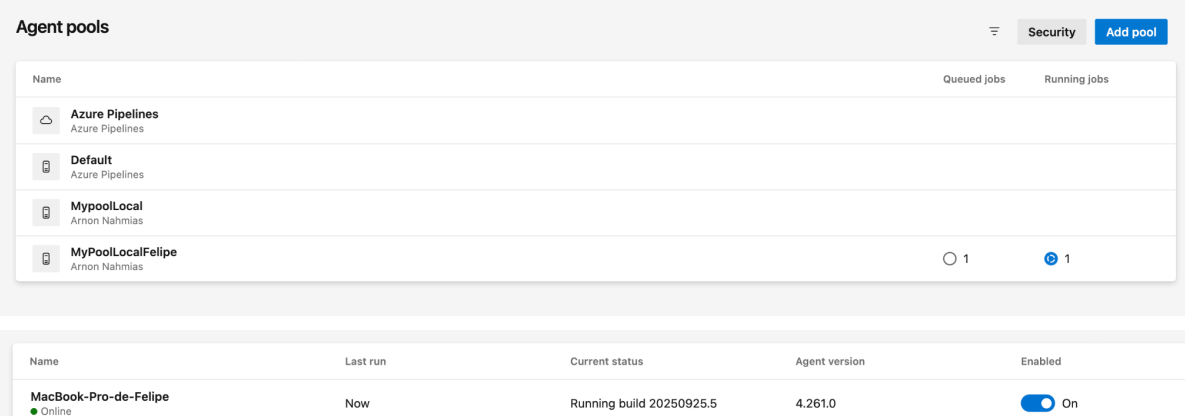
-  **Legible para humanos:** Más limpio que XML o JSON.
-  **Indentación por espacios:** No usa llaves ni corchetes.
-  **Muy usado en herramientas modernas:** GitHub Actions, Azure DevOps, Kubernetes, Docker Compose, etc.

basicamente es un codigo q ejecuta instaladores automáticamente y el yaml es para q sea legible?

📋 Tareas que debes cumplir

1. Preparación del entorno


- Crear ****Pool**** y ****Agente Self-Hosted**** en ADO (instalado como servicio en tu máquina).



Agent pools

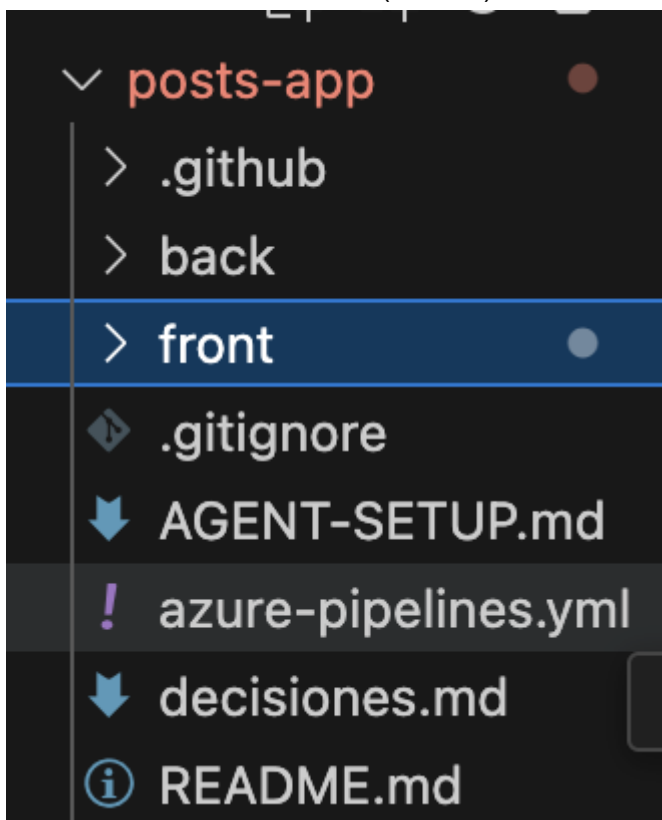
Name	Queued jobs	Running jobs
Azure Pipelines		
Default		
MypoolLocal		
MyPoolLocalFelipe	0	1

Name	Last run	Current status	Agent version	Enabled
MacBook-Pro-de-Felipe ● Online	Now	Running build 20250925.5	4.261.0	On

MyPoolLocalFelipe						Update all agents	New agent
Jobs	Agents	Details	Security	Approvals and checks	Analytics		
Name	Project	Agent	Queued	Wait time	Duration		
 Job 66 20250925.5 TP4 (20)	TP4	MacBook-Pro-de-Felipe	Today at 5:21 PM	<1s	2m 23s		

2. Estructura del repo y definición del pipeline

- Organizar `/front`, `/back` y agregar `azure-pipelines.yml` en raíz.
- YAML requerido (multi-stage):
 - `Stage CI` (trigger en `main`):
 - Build front (por ejemplo `npm ci && npm run build`).
 - Build back (por ejemplo `dotnet restore/build/test` o `mvn package`/`gradle build`).
 - Publicación de artefactos (dist/bin).



3. Evidencias

- Capturas: creación del pool/agente, ejecuciones de CI y artefactos publicados.

🛠️ Pasos sugeridos (checklist)

1. `Self-Hosted Agent`
 - Crear Pool `SelfHosted` y registrar `Agent-Local` (como servicio).
2. `Repo`
 - Estructura `/front`, `/back`, `azure-pipelines.yml`.
3. `CI`

- Build+test front y back, publicar artefactos.

4. **Evidencias**

- Capturas y explicación en `decisiones.md`.

📄 Entregables

1. **Acceso al proyecto en Azure DevOps** con:

- Pipeline **YAML** de CI apuntando al **Self-Hosted agent**.
- Ejecuciones exitosas (logs visibles) y artefactos publicados.

```

Removing agent from the server
Enter authentication type (press enter for PAT) >
Enter personal access token > *****
Connecting to server ...
Succeeded: Removing agent from the server
Removing .credentials
Succeeded: Removing .credentials
Removing agent
Succeeded: Removing agent
PS C:\agent> .\config.cmd --unattended --url https://dev.azure.com/2222270 --auth pat --token 6ah2PTH2nbcZPd1YFvGygfUyq0Uo2hmFssiqBrCFisXsJ8CsgoJQq799BtACRAA886Q6ChA888ZD0Fy4l --pool HypoLocal --agent ARNON
--work _work --subservice

Azure Pipelines
agent v4.261.0 (commit 45f3f81)

>> Connect:
Connecting to server ...
>> Register Agent:
Scanning for tool capabilities.
Connecting to the server.
Successfully added the agent
Testing agent connection.
2025-09-24 01:03:22: Settings: Agent
Creating file permissions to 'NT AUTHORITY\NETWORK SERVICE'.
Service vsagent.2222270.HypoLocal.ARNON successfully installed
Service vsagent.2222270.HypoLocal.ARNON successfully set recovery option
Service vsagent.2222270.HypoLocal.ARNON successfully set to delayed auto start
Service vsagent.2222270.HypoLocal.ARNON successfully configured
Service vsagent.2222270.HypoLocal.ARNON started successfully
PS C:\agent>

```

2. **Repositorio en GitHub** con:

- **README.md**: cómo ejecutar local, cómo corre el pipeline, prerequisites del agente, puertos, URLs .
- **decisiones.md** con:
 - Stack elegido y estructura del repo.
 - Diseño del pipeline (stages, jobs, artefactos).
 - Evidencias (capturas).

3. **URL del proyecto** en la planilla:

- [Planilla de

TPs](<https://docs.google.com/spreadsheets/d/1mZKJ8FH390QHjwkABokh3Ys6kMOFZGzZJ3-kg5ziELc/edit?gid=0#gid=0>)

🗣️ Defensa Oral Obligatoria

Preguntas típicas:

- ¿Por qué YAML y no Classic para este caso?
- ¿Qué ventajas concretas tiene usar un agente Self-Hosted frente a uno Microsoft-Hosted en tu escenario?
- ¿Cómo estructurarías el pipeline para que el build del front y del back sean independientes pero parte de la misma integración continua?

✅ Evaluación

Criterio	Peso
Pipeline YAML (CI) funcionando	35%
Claridad y justificación en `decisiones.md`	15%
Defensa oral: comprensión y argumentación	50%


⚠️ Uso de IA

Podés usar IA (ChatGPT, Copilot), pero ****deberás declarar qué parte fue generada con IA**** y justificar cómo la verificaste.

Si no podés defenderlo, ****no se aprueba****.

tp4 azure: en mac os

COMO SE EJECUTA EL PIPELINE CON LA APLICACION FUNCIONANDO DE MANERA NORMAL

 Jobs in run #20250925.1 TP4 (19)		
Continuous Integration		
▼	✓ Build Frontend (Next...	1m 11s
	✓ Initialize job	<1s
	✓ Checkout TP4@main t...	2s
	✓ Verify Frontend Direc...	<1s
	✓ Cache npm (frontend)	3s
	✓ Setup Node.js 18.x	<1s
	✓ Install Frontend Depe...	5s
	✓ Build Frontend	11s
	✓ Lint Frontend	<1s
	✓ Publish Frontend Arti...	20s
	✓ Post-job: Cache np...	26s
	✓ Post-job: Checkout T...	<1s
	✓ Finalize Job	<1s

Database Configuration

✓ ✓ Verify Database Connec... 4s

✓ Initialize job <1s

✓ Checkout TP4@main t... 2s

✓ Database Connectio... <1s

✓ Post-job: Checkout T... <1s

✓ Finalize Job <1s

Application Health Verification

▼	✓	Verify Application Health	3s
	✓	Initialize job	<1s
	✓	Checkout TP4@main t...	2s
	✓	Health Status Summ...	<1s
	✓	Post-job: Checkout T...	<1s
	✓	Finalize Job	<1s

←

Jobs in run #20250925.1

TP4 (19)

Continuous Integration

>

✓

Build Frontend (Next...

1m 11s

>

✓

Build Backend (Express...

28s

Database Configuration

>

✓

Verify Database Connec...

4s

Application Health Verification

>

✓

Verify Application Health

3s

✓

Build Frontend (Next.js)

1 Pool: [MyPoolLocalFelipe](#)

2 Queued: Today at 10:04 AM [[manage parallel jobs](#)]

3 Agent: MacBook-Pro-de-Felipe

4 Started: Today at 10:04 AM

5 Duration: 1m 11s

6

7 The agent request is already running or has already completed.

8 ▶ Job preparation parameters

33 1 artifact produced

✓ #20250925.1 • Set up CI with Azure Pipelines

TP4 (19)

Run new

This run is being retained as one of 3 recent runs by main (Branch).

View retention leases

Summary

Code Coverage

Individual CI by felipe.ganame.ucc@hotmail.com

View 24 changes

Repository and version

Time started and elapsed

Related

Tests and coverage

TP4

Today at 10:04 AM

0 work items

[Get started](#)

main

2m 50s

2 published

Stages

Jobs

✓ Continuous Integrat...

2 jobs completed

2m 2s

2 artifacts

✓ Database Configura...

1 job completed

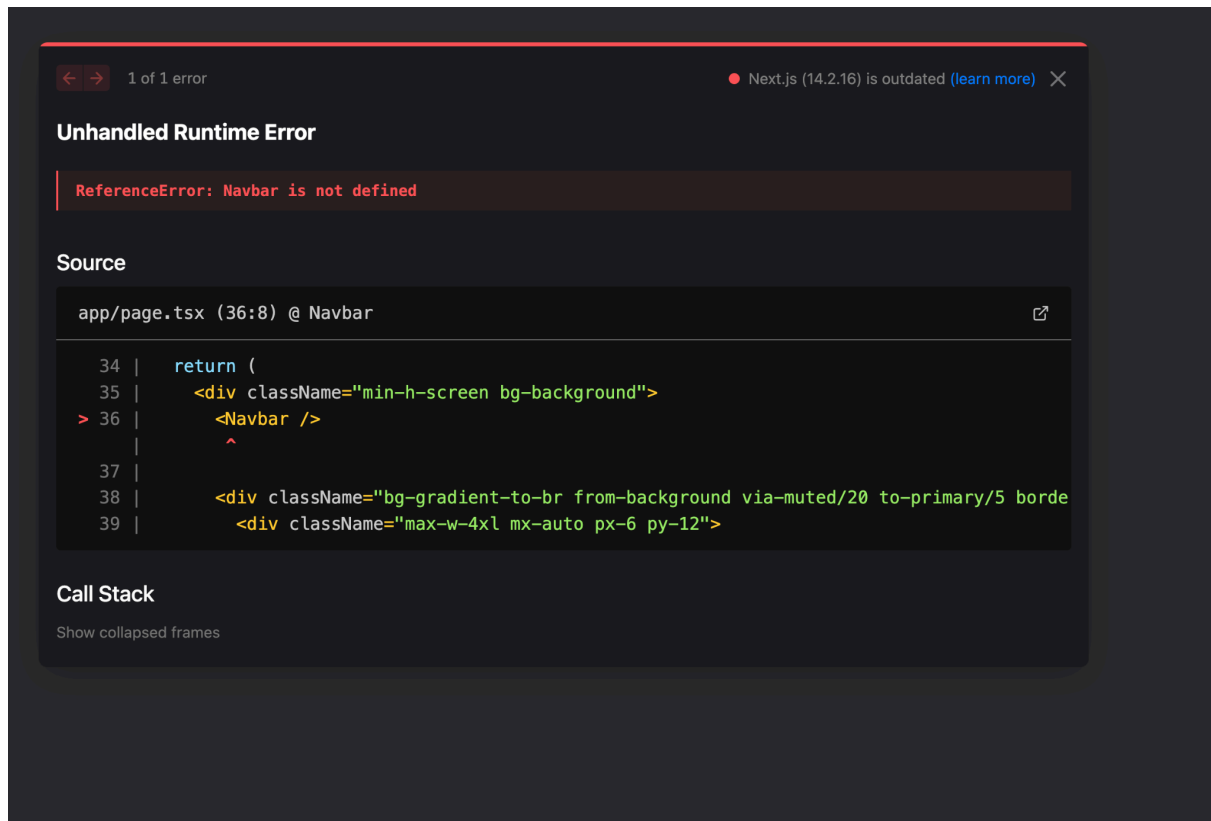
13s













✓ Application Health ...

1 job completed

15s

ROMPO EL FRONTEND INTENCIONALMENTE Y VEO COMO SE EJECUTA EL PIPELINE



▼		Build Frontend (Nex...	4m 31s
		Initialize job	<1s
		Checkout TP4@main t...	3s
		Verify Frontend Direc...	<1s
		Cache npm (fron...	3m 58s
		Setup Node.js 18.x	<1s
		Install Frontend Depe...	5s
		Build Frontend	9s
		Lint Frontend	<1s
		Publish Frontend Arti...	14s
		Post-job: Cache npm (fr...	
		Post-job: Checkout TP4...	

COMO SE VE EL PIPELINE CUANDO ROMPEMOS EL BACKEND:

✓	✗	Build Backend (Expres...	33s
✓		Initialize job	<1s
✓		Checkout TP4@main t...	6s
✓		Verify Backend Direc...	<1s
✓		Cache npm (backend)	16s
✓		Setup Node.js 18.x	<1s
✓		Install Backend Depen...	1s
✗		Build Backend (TypeS...	1s
⌵		Lint Backend	<1s
✓		Publish Backend Artif...	6s
✓		Post-job: Cache npm...	<1s
✓		Post-job: Checkout T...	<1s
✓		Finalize Job	<1s