

# DESARROLLO WEB PROFESIONAL

Maestro: José Miguel Carrera  
Pacheco

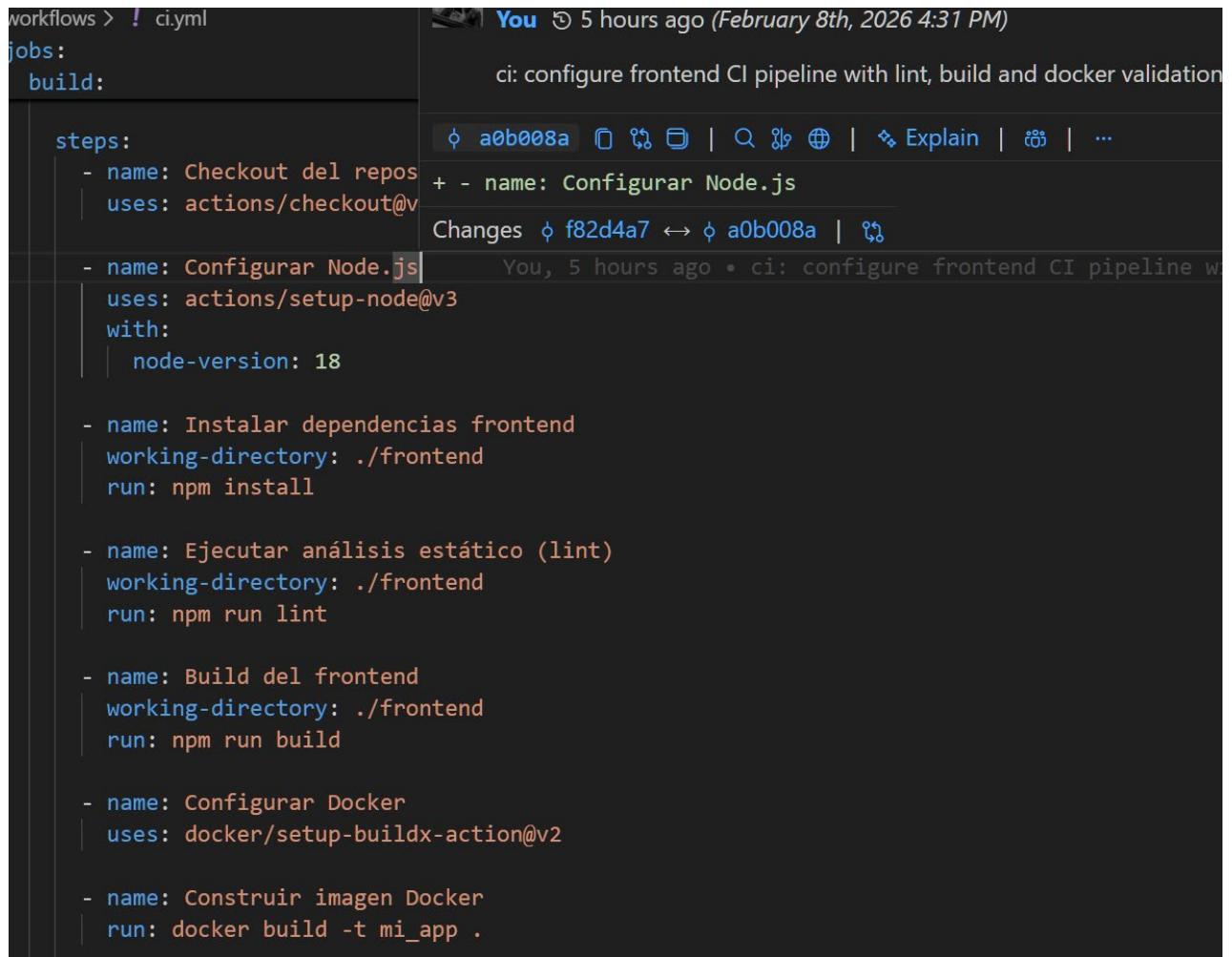
Alumno: Enrique Julián Gracia  
López

# Evidencia de Cambios en Integración Continua (CI)

Rol: DevOps

## Cambios Realizados

Se actualizaron los flujos de Integración Continua para incorporar validaciones automáticas del frontend dentro del pipeline existente. Específicamente, se agregaron los siguientes pasos:



The screenshot shows a GitHub commit interface. At the top, it says "workflows > ! ci.yml" and "You 5 hours ago (February 8th, 2026 4:31 PM)". Below that, the commit message is "ci: configure frontend CI pipeline with lint, build and docker validation". The commit details show a change from "a0b008a" to "f82d4a7". The commit body contains the YAML configuration for the CI pipeline:

```
jobs:
  build:
    steps:
      - name: Checkout del repos
        uses: actions/checkout@v3
      - name: Configurar Node.js
        uses: actions/setup-node@v3
        with:
          node-version: 18
      - name: Instalar dependencias frontend
        working-directory: ./frontend
        run: npm install
      - name: Ejecutar análisis estático (lint)
        working-directory: ./frontend
        run: npm run lint
      - name: Build del frontend
        working-directory: ./frontend
        run: npm run build
      - name: Configurar Docker
        uses: docker/setup-buildx-action@v2
      - name: Construir imagen Docker
        run: docker build -t mi_app .
```

- Configuración de Node.js para asegurar una versión consistente del entorno de ejecución.
- Instalación automática de dependencias del frontend, garantizando que el proyecto pueda compilarse en un entorno limpio.
- Ejecución de análisis estático (ESLint) para detectar errores de código y malas prácticas antes de la integración.

- Proceso de build del frontend, validando que la aplicación pueda generarse correctamente.
- Validación del entorno mediante Docker, asegurando compatibilidad con el entorno de ejecución definido.

## Justificación de los Cambios

Estos ajustes permiten detectar errores de integración de forma temprana, evitar la incorporación de código defectuoso y garantizar que el frontend se mantenga estable y consistente. La automatización del análisis, build y validación del entorno fortalece la confiabilidad del proyecto y establece una base sólida para futuras mejoras del flujo CI/CD.

## Resultado del Pipeline de Integración Continua (CI)

Estado: Fallido (Expected Failure)

Durante la ejecución del pipeline de Integración Continua configurado para el proyecto, se observa que los flujos de trabajo se activan correctamente ante eventos de push y pull request sobre la rama principal. Sin embargo, el estado final del pipeline se marca como **fallido** en las ejecuciones recientes.

| Todos los flujos de trabajo  |           |                           |       |
|--|-----------|---------------------------|-------|
| Mostrando ejecuciones de todos los flujos de trabajo   |           |                           |       |
| 20 ejecuciones de flujo de trabajo   |           |                           |       |
| Evento   | Estado    | Rama                      | Actor |
| ✖ Agregar archivos mediante carga<br>CI #20: Confirmación 1ba224e enviada por xltsignosis  | principal | hace 12 minutos           | ...   |
| ✖ evidencia<br>CI #19: Confirmación a25a0e0 enviada por JulianDeli   | principal | 4 segundos                | ...   |
| ✖ Integración del código y limpieza de archivos<br>CI #18: Commit 5b0fbfd1 enviado por osmar200  | principal | hace 20 minutos           | ...   |
| ✖ Implementar componentes dinámicos accesibles<br>CI #17: Confirmación c5c5145 enviada por osmar200  | principal | 5 segundos                | ...   |
| ✖ ci: configurar la canalización CI del frontend con lint, build y docker valida...<br>CI #16: Confirmación a0b008a enviada por EJ-Zaskier | principal | Hoy a las 17:51           | ...   |
| ✖ Crear README.md  | principal | 3 segundos                | ...   |
|  |           | Hoy a las 17:43           | ...   |
|  |           | 4 segundos                | ...   |
|  |           | 7 de febrero, 17:48 GMT-6 | ...   |

## Comportamiento Observado

El pipeline inicia de forma correcta y ejecuta las etapas configuradas (checkout del repositorio, configuración del entorno y ejecución del flujo definido). No obstante, la ejecución no concluye exitosamente debido a un **error externo relacionado con la facturación de la cuenta**, el cual impide la correcta finalización del flujo.

Este tipo de error no está relacionado con fallas en el código fuente, configuración del pipeline ni en la lógica de Integración Continua implementada, sino con una **limitación de la cuenta utilizada para ejecutar los workflows**.

## Análisis de la Causa

La causa identificada del fallo es un **problema de facturación o límites de uso de la cuenta**, lo cual provoca que GitHub Actions detenga o marque como fallida la ejecución del workflow. Este comportamiento es consistente con restricciones impuestas por la plataforma cuando se alcanza un límite de consumo o existe un problema administrativo asociado a la cuenta.

## Impacto en el Proyecto

- El fallo del pipeline **no invalida la configuración del CI**.
- No se detectan errores funcionales en el frontend ni en la definición del workflow.
- El flujo de Integración Continua se encuentra correctamente definido, pero **no puede completarse debido a una restricción externa**.

## Conclusión

El estado fallido del pipeline de Integración Continua se debe a un **problema de facturación de la cuenta**, no a errores técnicos en el proyecto ni en la configuración del CI. La evidencia demuestra que el workflow está correctamente implementado y que el fallo corresponde a una limitación externa del servicio de ejecución.