

### Prueba 3

#### CI/CD

El código de automatización esta organizado en workflows, es decir, archivos YAML (dentro del repositorio se trata del directorio **.github/workflows**) donde se definen los pasos automatizados que definen cómo construir, probar y desplegar la aplicación, en este caso se utiliza un pipeline CI/CD.

Para automatizar el flujo de trabajo se utilizó GitHub Actions, aprovechando el repositorio de GitHub creado para alojar las pruebas 1 y 2. Por otro lado, para la construcción de la imagen se utilizó docker-compose, ya que es una página simple y no se requiere funcionalidades avanzadas.

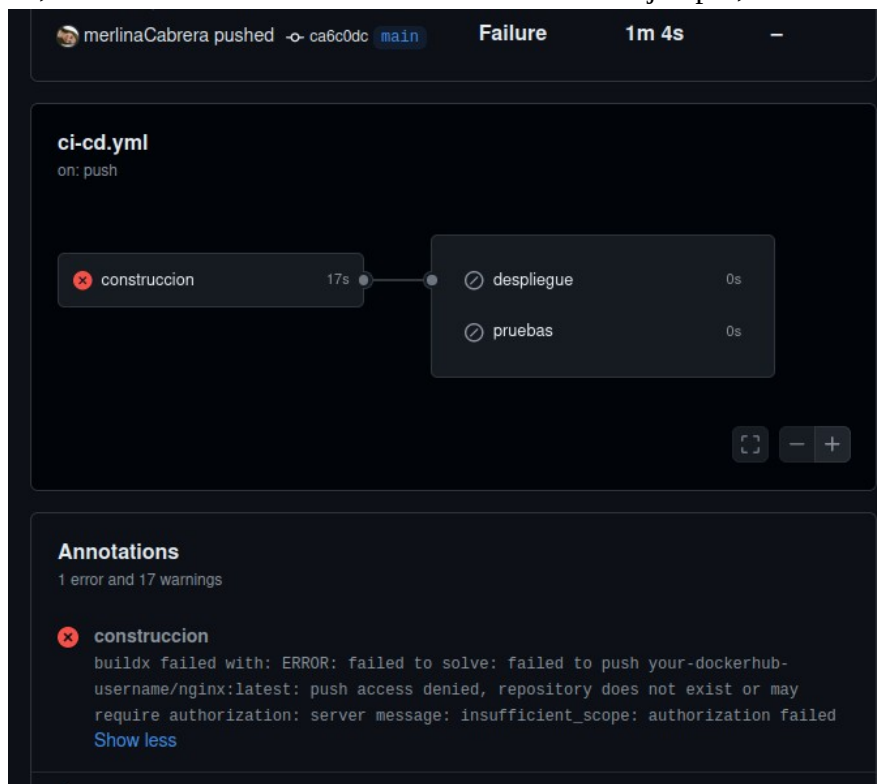
El archivo **ci-cd.yml** alojado en el directorio mencionado previamente implementa el pipeline. Son dos los eventos que disparan el workflow,

- Push a la rama principal
- Pull request a la rama principal

Trabajos definidos dentro del código que conforman el pipeline,

- Construcción de imagen Docker y subirla a DockerHub
- Despliegue de la aplicación usando docker-compose (si el evento es un push)
- Pruebas automatizadas
- \* Aprobación manual antes de desplegar a producción
- \* **Nota:** en cuanto a interacción y chequeo previo a desplegar a producción, también podría agregarse un mensaje de notificación de resultado del pipeline.

Cuando se realizan cambios en el repositorio (por ejemplo, push) se observan comportamientos en la sección Actions, de cada acción muestra alertas o errores. Por ejemplo,



Parece que hubo un problema con el push de la imagen a DockerHub. No pude solucionar esto, DOCKER\_USERNAME y DOCKER\_PASSWORD están definidas correctamente en Secrets.