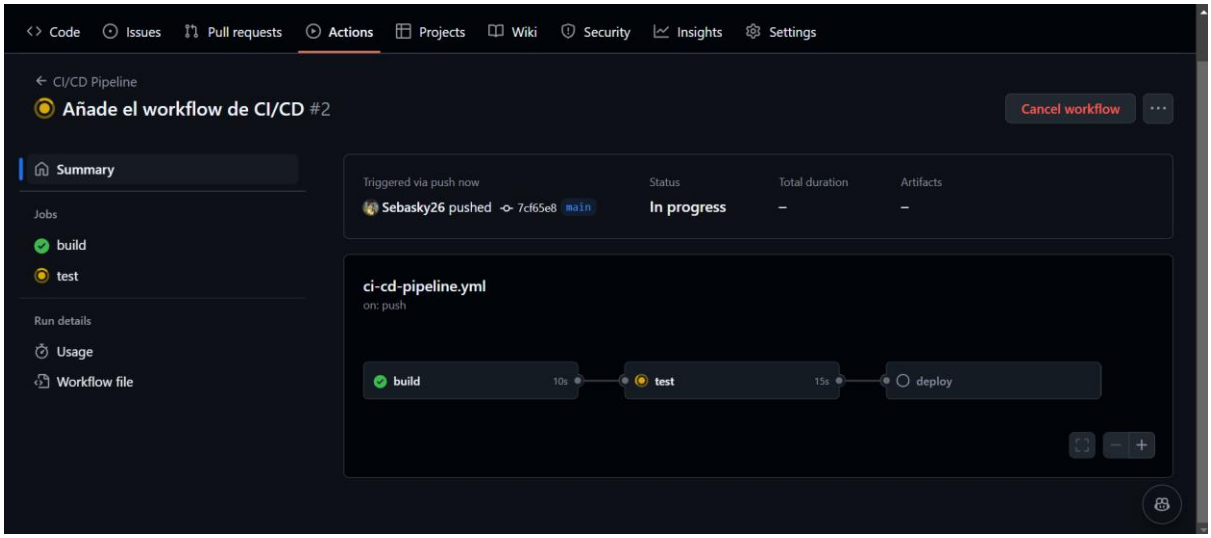
Examen B1

Nombre: Sebastian Aisalla

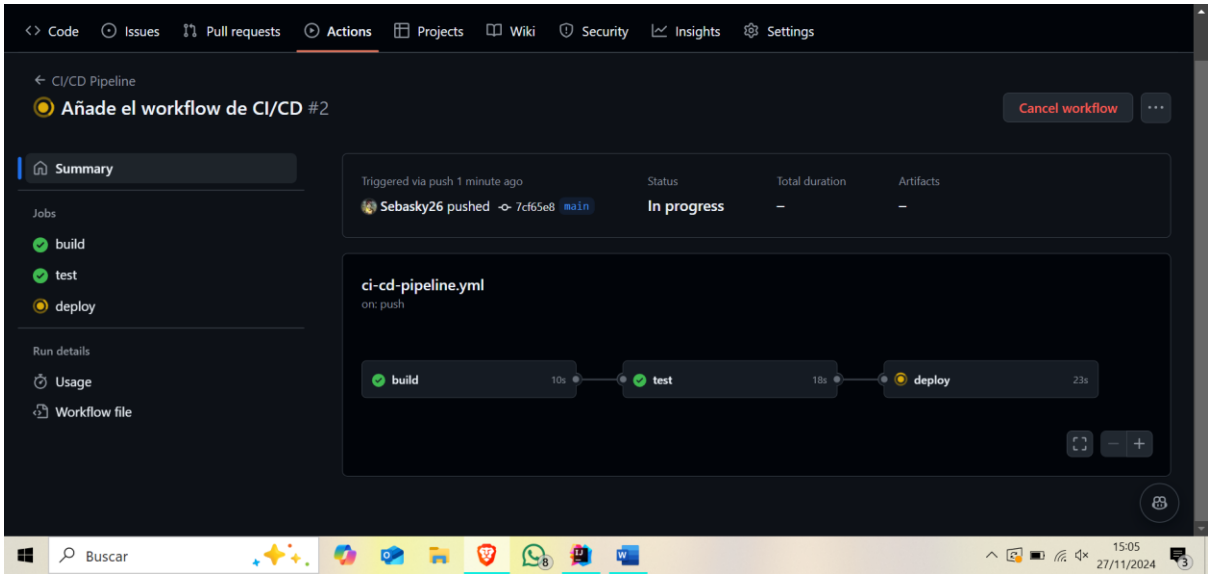
Fecha: 27/11/2024

Link GitHub: https://github.com/Sebasky26/Examen_Construccion_1B.git

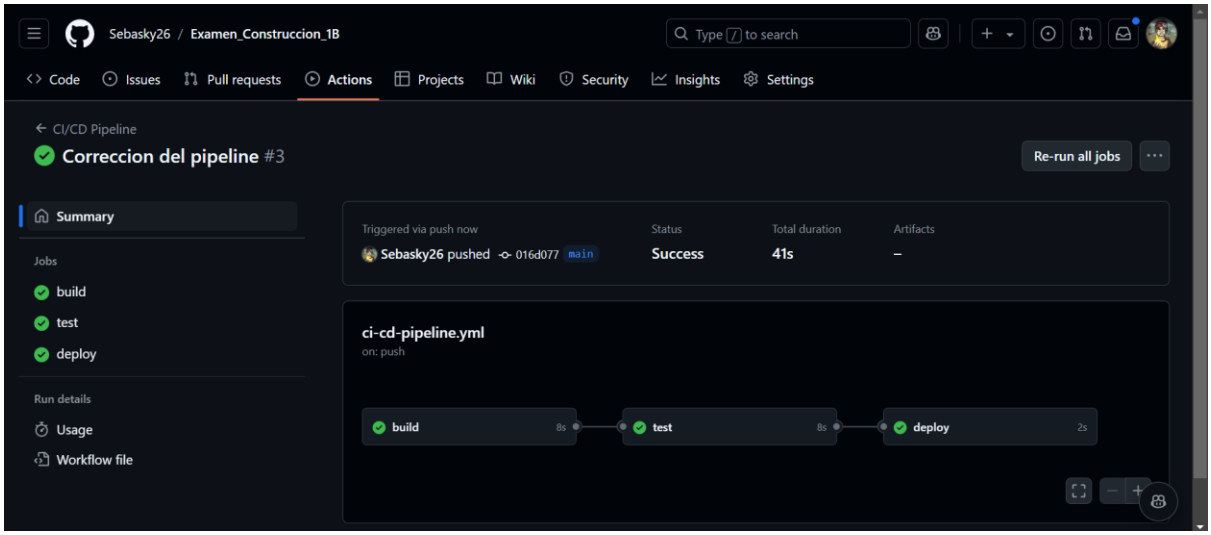
Build



Test



Deploy



Pasos para la creación del pipeline

1. **Creación de la Aplicación:**
    - Empecé desarrollando una aplicación sencilla en Java llamada **"Lista de Tareas"**. Esta aplicación permite agregar, listar y marcar tareas como completadas.
    - Organicé el proyecto siguiendo la estructura estándar de Maven, con los archivos fuente en src/main/java y los archivos de prueba en src/test/java. Definí las clases principales ListaDeTareas y Tarea, además de las pruebas unitarias ListaDeTareasTest y TareaTest.
    - Configuré el archivo pom.xml para incluir dependencias como **JUnit 5** y aseguré la compatibilidad con Java 17.
2. **Subida del Proyecto a GitHub:**
    - Creé un repositorio en GitHub llamado **"Examen_B1_Construccion"**.
    - Inicialicé Git en mi proyecto local, añadí los archivos y subí el proyecto al repositorio.
3. **Configuración del Workflow en GitHub Actions:**
    - Dentro del repositorio, creé una carpeta .github/workflows/ y añadí un archivo llamado ci-cd-pipeline.yml para definir el pipeline.

- Configuré el workflow para que se ejecute automáticamente en eventos como push y pull_request en la rama main.

4. **Definición de los Pasos del Pipeline:**

Etapa 1: Build

- Configuré el paso para compilar el proyecto usando Maven. Incluí los siguientes pasos:
  - Clonar el repositorio usando actions/checkout.
  - Configurar Java 17 en el entorno de ejecución con actions/setup-java.
  - Ejecutar el comando mvn compile para compilar los archivos fuente.
- Verifiqué que el directorio target/ se generara correctamente al listar su contenido.

Etapa 2: Test

- Configuré esta etapa para ejecutar pruebas unitarias con Maven. Los pasos incluyen:
  - Clonar el repositorio y configurar Java como en la etapa anterior.
  - Ejecutar el comando mvn test para validar las pruebas unitarias definidas en ListaDeTareasTest y TareaTest.

Etapa 3: Deploy

- Definí esta etapa para simular el despliegue del proyecto. Incluí:
  - Verificar si el directorio target/ existía.
  - Crear un archivo comprimido (deploy-package.zip) con el contenido de target/ si estaba presente.

5. **Verificación del Pipeline:**

- Realicé pruebas subiendo cambios al repositorio para activar el pipeline.

- En la pestaña **Actions** de GitHub, confirmé que las etapas Build, Test y Deploy se ejecutaban correctamente.
- Solucioné problemas, como conflictos y errores en la configuración de Maven o faltantes en el directorio target/, ajustando el workflow en consecuencia.