Pipeline de CI/CD - Projeto Semáforos Inteligentes

Este pipeline foi criado utilizando GitHub Actions para automatizar o processo de integração contínua (CI) e entrega contínua (CD) de um projeto Java com Spring Boot.

Etapas do Pipeline:

1. Gatilhos de Execução (Triggers)

- O pipeline é acionado automaticamente em push ou pull request nas branches "main" e "develop".

2. Build do Projeto

- Checkout do código-fonte do repositório.
- Instalação do Java 21 (usando Temurin).
- Concessão de permissão de execução para o Maven Wrapper.
- Execução do Maven para compilar o projeto (sem testes): ./mvnw clean install -DskipTests

3. Testes Automatizados

- Execução dos testes da aplicação (temporariamente desativado se necessário): ./mvnw test

4. Geração do Artefato (JAR)

- Geração do pacote JAR com: ./mvnw package -DskipTests

5. Upload do Artefato (opcional)

- A etapa de upload pode ser reativada utilizando a action: actions/upload-artifact@v3.1.2

Componentes Utilizados:

- GitHub Actions: plataforma de automação de CI/CD integrada ao GitHub.

- Java 21 (Temurin): versão da JVM utilizada no ambiente de execução.

- Maven Wrapper: ferramenta de build para projetos Java.

- Spring Boot: framework Java para criação de aplicações modernas.

- Git: versionamento de código-fonte.

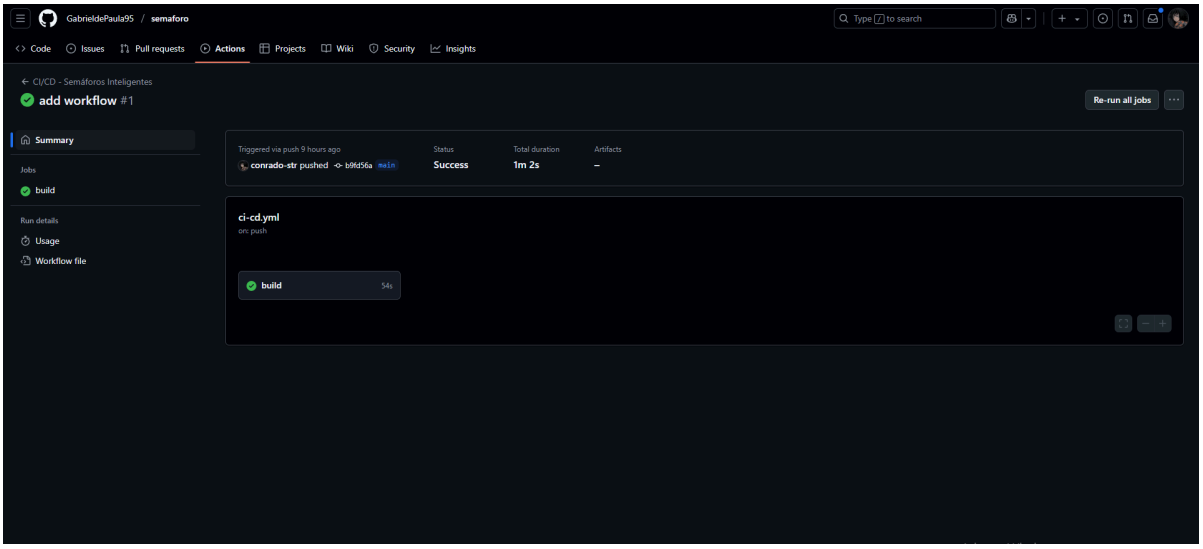
- YAML: linguagem de configuração do workflow.

Trecho do arquivo ci-cd.yml:

```yaml
name: CI/CD - Semáforos Inteligentes

on:
 push:
   branches:
     - main
     - develop
 pull_request:
   branches:
     - main
     - develop

jobs:
 build:
   runs-on: ubuntu-latest

   steps:
     - name: Checkout do código
       uses: actions/checkout@v3

     - name: Instala o Java 21
       uses: actions/setup-java@v3
       with:
         distribution: 'temurin'
```

```
        java-version: '21'

    - name: Dá permissão ao Maven Wrapper
      run: chmod +x ./mvnw

    - name: Build com Maven
      run: ./mvnw clean install -DskipTests=true

    - name: Rodar testes
      run: ./mvnw test

    - name: Gerar artefato
      run: ./mvnw package -DskipTests
```

Bild pelo Github Actions:



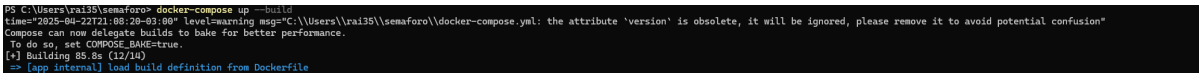
6. 🐳 Rodando com Docker

- Docker Desktop (instalado)
- Git instalado
- [Docker Compose](https://docs.docker.com/compose/)

Inicie o Docker Desktop

Navegue até o diretório da aplicação.

 # 1. Buildar a aplicação



docker-compose up --build
- Criar o container do banco Oracle XE
- Fazer o build da aplicação Spring Boot
- Iniciar a aplicação na porta 8080

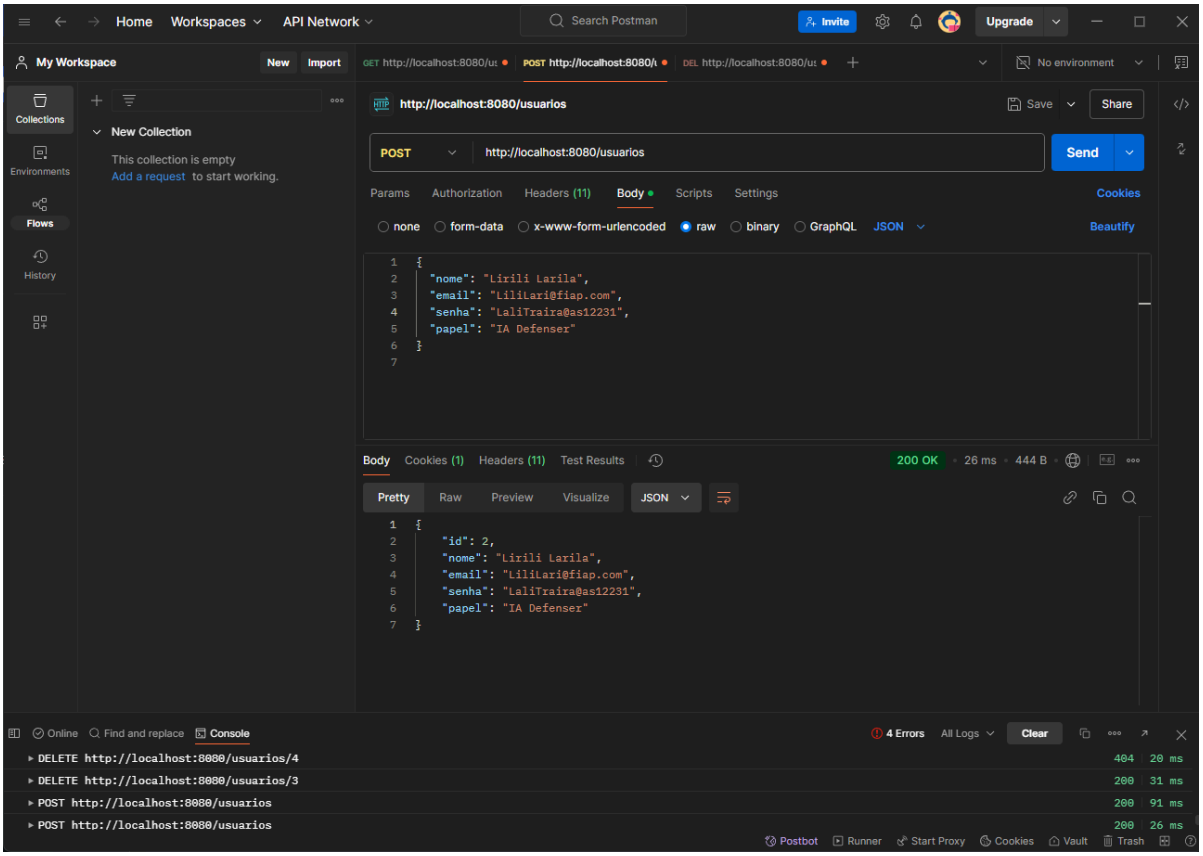
 # 2. Acesse a aplicação
exemplo:
http://localhost:8080/usuarios

 # 3. login: user
senha: (gerada no terminal)

Teste com Postman (caso necessário)
- Aba Authorization
- Tipo: No Auth **(colocado apenas para testar, é possivel dar um Get, Post e Delete.)**

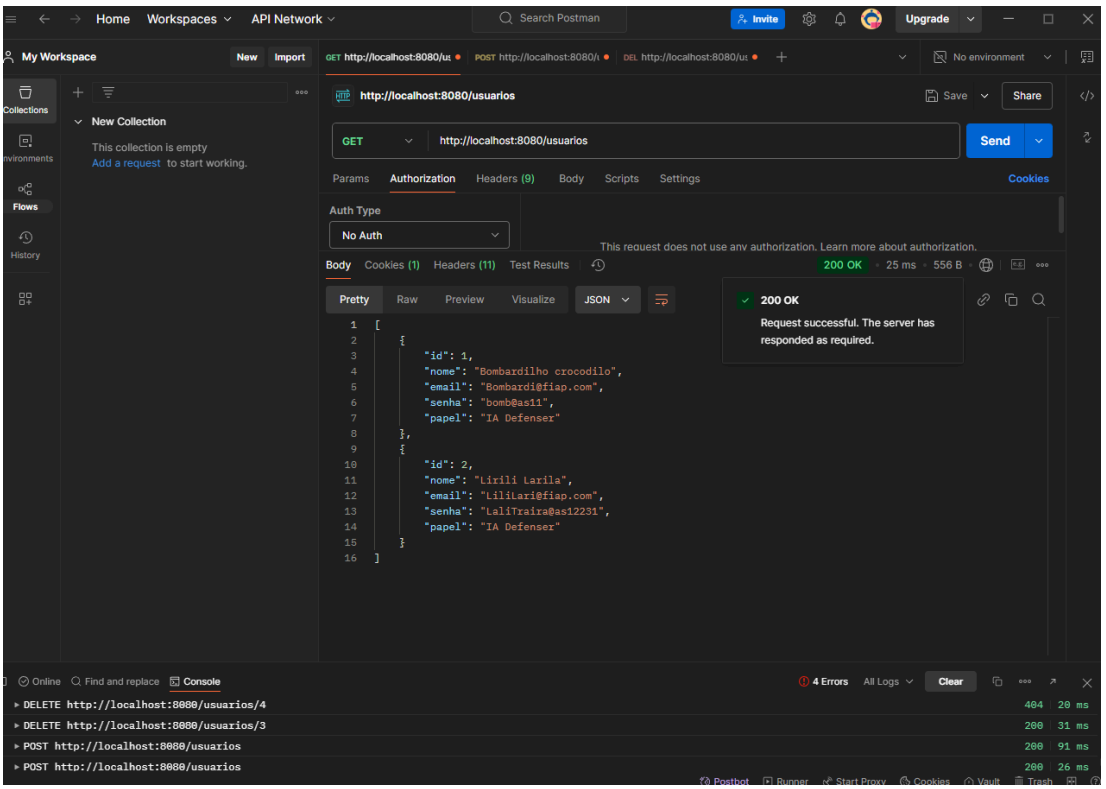
POST
POST http://localhost:8080/usuarios

Content-Type: application/json

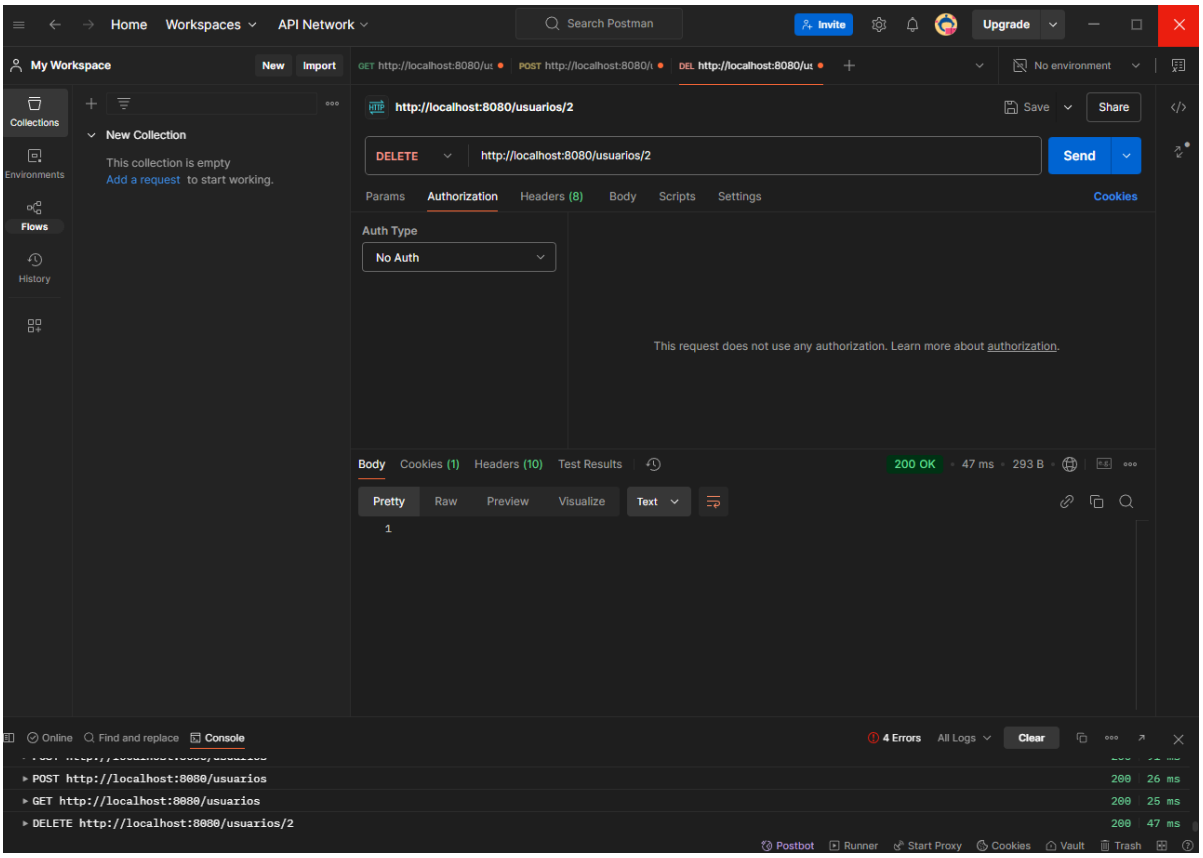



```
{
  "nome": "Lirili Larila",
  "email": "LiliLari@fiap.com",
  "senha": "LaliTraira@as12231",
  "papel": "IA Defenser"
}
```

GET http://localhost:8080/usuarios


Home
Workspaces
API Network
Search Postman
Invite
Upgrade
My Workspace
New
Import
No environment
Collections
Environments
Flows
History
New Collection
This collection is empty
Add a request to start working.
http://localhost:8080/usuarios
Save
Share
GET
http://localhost:8080/usuarios
Send
Params
Authorization
Headers (9)
Body
Scripts
Settings
Cookies
Auth Type
No Auth
This request does not use any authorization. Learn more about authorization.
Body
Cookies (1)
Headers (11)
Test Results
200 OK
25 ms
556 B
Pretty
Raw
Preview
Visualize
JSON
200 OK
Request successful. The server has responded as required.
[
{
"id": 1,
"nome": "Bombardilho crocodilo",
"email": "Bombardi@fiap.com",
"senha": "bomb@as11",
"papel": "IA Defenser"
},
{
"id": 2,
"nome": "Lirili Larila",
"email": "LiliLari@fiap.com",
"senha": "LaliTraira@as12231",
"papel": "IA Defenser"
}
]
Online
Find and replace
Console
4 Errors
All Logs
Clear
DELETE http://localhost:8080/usuarios/4 404 20 ms
DELETE http://localhost:8080/usuarios/3 200 31 ms
POST http://localhost:8080/usuarios 200 91 ms
POST http://localhost:8080/usuarios 200 26 ms
Postbot
Runner
Start Proxy
Cookies
Vault
Trash

Delete


Home
Workspaces
API Network
Search Postman
Invite
Upgrade
My Workspace
New
Import
No environment
Collections
Environments
Flows
History
New Collection
This collection is empty
Add a request to start working.
http://localhost:8080/usuarios/2
Save
Share
DELETE
http://localhost:8080/usuarios/2
Send
Params
Authorization
Headers (8)
Body
Scripts
Settings
Cookies
Auth Type
No Auth
This request does not use any authorization. Learn more about authorization.
Body
Cookies (1)
Headers (10)
Test Results
200 OK
47 ms
293 B
Pretty
Raw
Preview
Visualize
Text
1
Online
Find and replace
Console
4 Errors
All Logs
Clear
POST http://localhost:8080/usuarios 200 26 ms
GET http://localhost:8080/usuarios 200 25 ms
DELETE http://localhost:8080/usuarios/2 200 47 ms
Postbot
Runner
Start Proxy
Cookies
Vault
Trash

Para intorremper os containers

docker-compose down