GRUPO 7 - JOGO TRILHA

Eric d'Arriaga, João Gabriel Pereira, João Vitor Baggio

GitHub Actions

A ferramenta utilizada foi o GitHub Actions para a criação de um pipeline de continuous integration.

A instalação foi simples, sendo necessário apenas configurar o arquivo .yml no repositório do trabalho.

Nesse .yml foi adicionado comandos para configurar o sistema operacional, configurar o ambiente JDK e compilar os arquivos .java presentes no trabalho. Depois disso, foi criado comandos de teste usando JUnit.

GitHub Actions

```
name: Java CI
 push:
   branches:
     - main
 pull_request:
   branches:
     - main
 build-ubuntu:
   runs-on: ubuntu-latest
    steps:
    - name: Checkout code
     uses: actions/checkout@v3
    - name: Setup JDK
      uses: actions/setup-java@v3
      with:
       java-version: '17'
       distribution: 'temurin'
    - name: Compile Java Code
      run:
       mkdir -p bin
       javac -d bin -cp lib/junit-platform-console-standalone-1.11.0.jar *.java
    - name: Run Tests
      run:
       java -cp "bin:lib/junit-platform-console-standalone-1.11.0.jar" org.junit.platform.console.Consol
```

GitHub Actions

Passo a passo:

Realizamos a configuração do GitHub Actions através da criação do arquivo .github/workflows/ci.yml

Foram definidos jobs para diferentes sistemas operacionais (windows, ubuntu e macOS).

Depois foi feita a configuração da versão do Java 17.

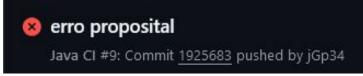
Logo após é realizado a configuração do JDK e compilação dos arquivos .java do jogo.

Por último está a realização de testes com JUnit.

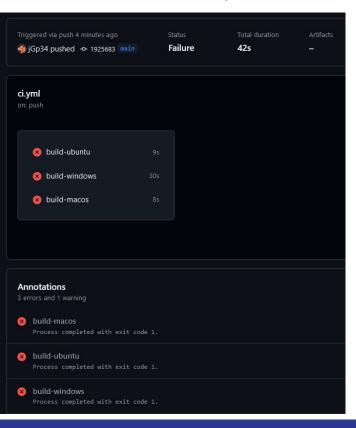
Demonstração de Falha

```
public boolean colocarPeca(int id, char simbolo) { 14 usages * jGp34
    Casa casa = casas.get(id);

if (casa.getOcupante() == '-') {
    casa.setOcupante();
    return true;
    }
    return false;
}
```



Demonstração de Falha



```
build-ubuntu
                                                 Q Search logs
failed 6 minutes ago in 9s
    Set up job
    Checkout code
       Setup JDK
        Compile Java Code
                                                                                                     3s
       ► Run mkdir -p bin
        Tabuleiro.java:45: error: method setOcupante in class Casa cannot be applied to given types;
                   casa.setOcupante();
          required: char
                   no arguments
          found:
          reason: actual and formal argument lists differ in length
    14 1 error
    15 Error: Process completed with exit code 1.
```

Uma falha mais interessante

```
public boolean colocarPeca(int id, char simbolo) { 14 usages
    Casa casa = casas.get(id);
    if (casa.getOcupante() == '-') {
        casa.setOcupante(simbolo);
        return false;
    return false;
```

Uma falha mais interessante

└ Lista de vizinhos ✓

JUnit Vintage √

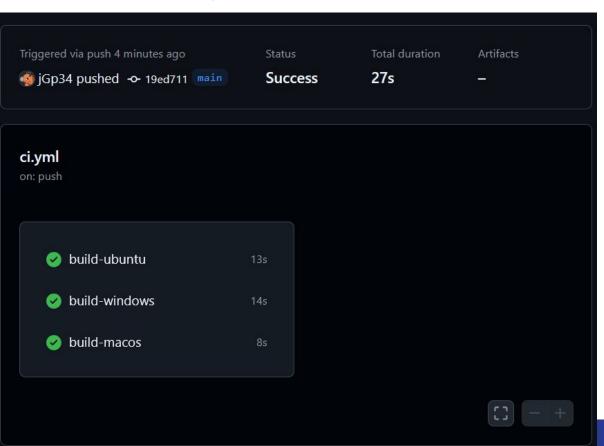
```
build-ubuntu
                                                                                                            32 Failures (1):
                                                 Q Search logs
                                                                                                                  JUnit Jupiter: TabuleiroTest: colocarPeca()

✓ ■ Run Tests

    1 ▶ Run java -cp "bin:lib/junit-platform-console-standalone-1.11.0.jar"
        org.junit.platform.console.ConsoleLauncher --scan-classpath
       Thanks for using JUnit! Support its development at https://junit.org/sponsoring
   10 Removido com sucesso
   11 Bloqueio com sucesso. O jogador Y perdeu!
                                                                                                                       TabuleiroTest.colocarPeca(TabuleiroTest.java:14)
           JUnit Platform Suite √
                                                                                                                       java.base/java.lang.reflect.Method.invoke(Method.java:569)
           JUnit Jupiter ✓
                                                                                                                       java.base/java.util.ArrayList.forEach(ArrayList.java:1511)
           — TabuleiroTest ✓
                                                                                                                       java.base/java.util.ArrayList.forEach(ArrayList.java:1511)
               — removerPeca() √
               — getCasa() √
              — getPeca() ✓
               - verificarBloqueio() ✓
              − verificarMoinho() √
              ⊢ resetarTabuleiro() ✓
              — colocarPeca() X expected: <true> but was: <false>
              └ moverPeca() ✓
           └ CasaTest ✓
              ─ Casa com peça branca ✓
              — Casa e seu vizinho ✓
              — Casa com peca preta ✓
              — Casa vazia √
```

```
MethodSource [className = 'TabuleiroTest', methodName = 'colocarPeca', methodParameterTypes =
    => org.opentest4i.AssertionFailedError: expected: <true> but was: <false>
       org.junit.jupiter.api.AssertionFailureBuilder.build(AssertionFailureBuilder.java:151)
org.junit.jupiter.api.AssertionFailureBuilder.buildAndThrow(AssertionFailureBuilder.java:132)
       org.junit.jupiter.api.AssertTrue.failNotTrue(AssertTrue.java:63)
       org.junit.jupiter.api.AssertTrue.assertTrue(AssertTrue.java:36)
       org.junit.jupiter.api.AssertTrue.assertTrue(AssertTrue.java:31)
       org.junit.jupiter.api.Assertions.assertTrue(Assertions.java:183)
```

Tudo corrigido e correto



build-ubuntu succeeded 5 minutes ago in 13s Set up job Checkout code Setup JDK Compile Java Code Run Tests Post Setup JDK Post Checkout code Complete job

Dificuldades encontradas

A criação desse pipeline foi, de maneira geral, bem simples. O único empecilho foi configurar o JUnit corretamente, já que o projeto não conta com ferramentas que configuram automaticamente dependências, como é o caso do Maven.

Para contornar o problema, foi necessário apenas digitar à mão o caminho do .jar que contém as bibliotecas e dependências necessárias para rodar os testes JUnit.

Para além disso, houveram apenas inconveniências simples, como descobrir os comandos para cada sistema operacional.