

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

```
1 name: Java CI
2
3 on:
4   push:
5     branches:
6       - main
7   pull_request:
8     branches:
9       - main
10
11 jobs:
12   build-ubuntu:
13     runs-on: ubuntu-latest
14
15     steps:
16     - name: Checkout code
17       uses: actions/checkout@v3
18
19     - name: Setup JDK
20       uses: actions/setup-java@v3
21       with:
22         java-version: '17'
23         distribution: 'temurin'
24
25     - name: Compile Java Code
26       run: |
27         mkdir -p bin
28         javac -d bin -cp lib/junit-platform-console-standalone-1.11.0.jar *.java
29
30     - name: Run Tests
31       run: |
32         java -cp "bin:lib/junit-platform-console-standalone-1.11.0.jar" org.junit.platform.console.Console
```

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;
}
```



erro proposital

main

Java CI #9: Commit 1925683 pushed by jGp34



4 minutes ago



42s

Demonstração de Falha

Triggered via push 4 minutes ago

jGp34 pushed 1925683 **main**

Status

Failure

Total duration

42s

Artifacts

-

ci.yml
on: push

build-ubuntu 9s

build-windows 30s

build-macos 8s

Annotations
3 errors and 1 warning

build-macos
Process completed with exit code 1.

build-ubuntu
Process completed with exit code 1.

build-windows
Process completed with exit code 1.

build-ubuntu
failed 6 minutes ago in 9s

> Set up job 2s

> Checkout code 2s

> Setup JDK 0s

> **Compile Java Code** 3s

```
1  ▶ Run mkdir -p bin
8  Tabuleiro.java:45: error: method setOcupante in class Casa cannot be applied to given types;
9      casa.setOcupante();
10             ^
11     required: char
12     found:    no arguments
13     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

build-ubuntu

failed 3 minutes ago in 10s

Search logs

Run Tests



1s

```
1  ▶ Run java -cp "bin:lib/junit-platform-console-standalone-1.11.0.jar"
2  org.junit.platform.console.ConsoleLauncher --scan-classpath
3
4  Thanks for using JUnit! Support its development at https://junit.org/sponsoring
5
6  Removido com sucesso
7  Bloqueio com sucesso. O jogador Y perdeu!
8
9  |
10 | └─ JUnit Platform Suite ✓
11 |   └─ JUnit Jupiter ✓
12 |     └─ TabuleiroTest ✓
13 |       └─ removerPeca() ✓
14 |       └─ getCasa() ✓
15 |       └─ getPeca() ✓
16 |       └─ verificarBloqueio() ✓
17 |       └─ verificarMoinho() ✓
18 |       └─ resetarTabuleiro() ✓
19 |       └─ colocarPeca() ✗ expected: <true> but was: <false>
20 |       └─ moverPeca() ✓
21 |     └─ CasaTest ✓
22 |       └─ Casa com peça branca ✓
23 |       └─ Casa e seu vizinho ✓
24 |       └─ Casa com peça preta ✓
25 |       └─ Casa vazia ✓
26 |     └─ Lista de vizinhos ✓
27 |   └─ JUnit Vintage ✓
```

```
32 Failures (1):
33   JUnit Jupiter:TabuleiroTest:colocarPeca()
34     MethodSource [className = 'TabuleiroTest', methodName = 'colocarPeca', methodParameterTypes =
35       '']
36     => org.opentest4j.AssertionFailedError: expected: <true> but was: <false>
37       org.junit.jupiter.api.AssertionFailureBuilder.build(AssertionFailureBuilder.java:151)
38     org.junit.jupiter.api.AssertionFailureBuilder.buildAndThrow(AssertionFailureBuilder.java:132)
39       org.junit.jupiter.api.AssertTrue.failNotTrue(AssertTrue.java:63)
40       org.junit.jupiter.api.AssertTrue.assertTrue(AssertTrue.java:36)
41       org.junit.jupiter.api.AssertTrue.assertTrue(AssertTrue.java:31)
42       org.junit.jupiter.api.Assertions.assertTrue(Assertions.java:183)
43       TabuleiroTest.colocarPeca(TabuleiroTest.java:14)
44       java.base/java.lang.reflect.Method.invoke(Method.java:569)
45       java.base/java.util.ArrayList.forEach(ArrayList.java:1511)
46       java.base/java.util.ArrayList.forEach(ArrayList.java:1511)
```


Tudo corrigido e correto

Triggered via push 4 minutes ago

 jGp34 pushed  19ed711 main

Status

Success

Total duration

27s

Artifacts

—

ci.yml

on: push

- ✓ build-ubuntu 13s
- ✓ build-windows 14s
- ✓ build-macos 8s



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.

