



# unopar

UNIVERSIDADE PITÁGORAS UNOPAR ANHANGUERA - MARAVILHA  
ENGENHARIA DE SOFTWARE

NATAN OGLIARI - 34466876

PROGRAMAÇÃO WEB

Maravilha/SC  
2024

NATAN OGLIARI - 34466876

## PROGRAMAÇÃO WEB

Trabalho de portfólio apresentado como requisito parcial para a obtenção de pontos para a média semestral.

Orientador: Anderson Emidio de Macedo Gonçalves .

Maravilha/SC  
2024

# Sumário

|                                    | Páginas  |
|------------------------------------|----------|
| <b>1   Introdução</b>              | <b>4</b> |
| <b>2   Métodos</b>                 | <b>4</b> |
| <b>3   Resultados</b>              | <b>7</b> |
| 3.1   Erro: novo usuário . . . . . | 7        |
| <b>4   Conclusões</b>              | <b>9</b> |

# 1 Introdução

A programação web vem ganhando cada vez mais espaço no mundo globalizado, nos dias atuais torna-se praticamente inviável a instituições dos mais variados tipos, que não possuem um web site e neste aspecto que se capacita profissionais para este leque.

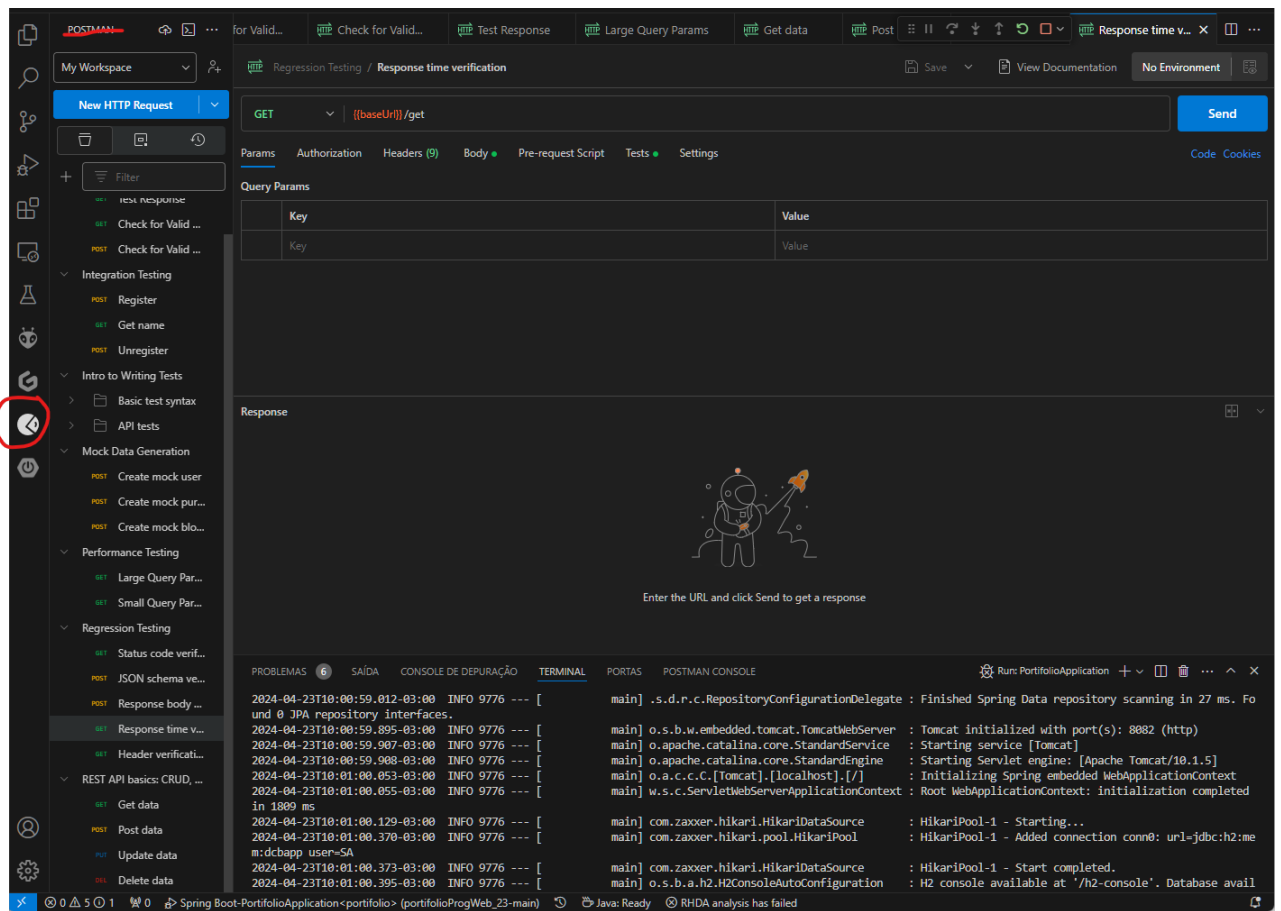
Segundo alvaron (2024), empresas que não possuem site perdem clientes para concorrentes, pois sem um site os sites de pesquisas não irão encontrar a empresa para aquisição de algum ou alguns produtos ou serviços.

# 2 Métodos

Para esta aula prática foi proposto um roteiro, está disposto em: roteiro da aula prática. De igual modo cria-se um repositório no GitHub para o versionamento da referida aula prática, e que pode ser acessado através deste link.

Neste vies foi eleito o modelo de relatórios em **LaTeX**, pois o mesmo acaba automatizando alguns aspectos. Nos aspectos da realização de atividade foi sugerido a utilização da IDE **Eclipse**, a instalação do Postman e como opcional o **Git**. Porém opto pela utilização da IDE **VS Code**, pois a mesma já vem integrado *plugin* do **Postman**, conforme demonstra a figura 1.

**Figura 1.** *Plugin do Postman na IDE vscode.*





Fonte: O autor (2024)

Sugere a criação de um modelo pré configurado através da ferramenta *Spring initializr*, e a pré configuração foi definida conforme demonstrada na figura 2. a versão do *Spring* definida no roteiro é a **3.0.0**, no entanto não esta mais disponível e optei pela versão **3.2.5**.

**Figura 2.** *Configuração Spring initializr*

22/04/2024, 09:21Spring Initializr



---

**Project**  
☐ Gradle - Groovy ☐ Gradle - Kotlin ☒ Maven

**Language**  
☒ Java ☐ Kotlin ☐ Groovy

**Spring Boot**  
☐ 3.3.0 (SNAPSHOT) ☐ 3.3.0 (RC1) ☐ 3.2.6 (SNAPSHOT) ☒ 3.2.5  
☐ 3.1.12 (SNAPSHOT) ☐ 3.1.11

**Project Metadata**  

Group  
com.example

Artifact  
demo

Name  
demo

Description  
Demo project for Spring Boot

Package name  
com.example.demo

Packaging  
☒ Jar ☐ War

Java  
☐ 22 ☐ 21 ☒ 17

**Dependencies**

ADD ...

GENERATEEXPLORESHARE...

<https://start.spring.io>1/2

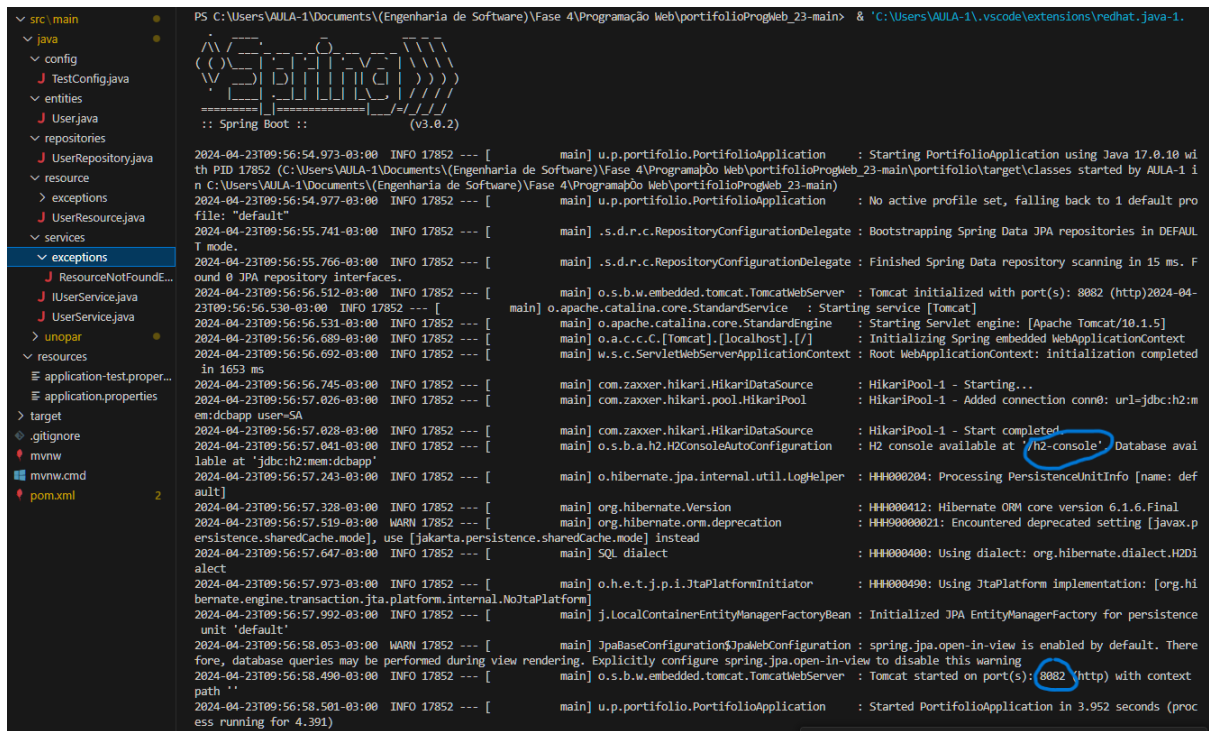
Fonte: O autor (2024)

É realizado uma sequência de passos disponibilizada no roteiro, e em relação ao passo 7, que indica a criação de um arquivo chamado de *application.properties*, já estava criado. com o conteúdo (**spring.application.name=demo**), e então apenas adicionei abaixo o conteúdo indicado no roteiro.

### 3 Resultados

Após as implementações e alguns problemas, foi prosseguido para os testes, na figura 3 de monstra o resultado da compilação do código em java.

**Figura 3.** Resultado da compilação do código em Java.



Fonte: O autor (2024)

Nesta compilação pode-se averiguar o nome para acesso e a porta de escuta, o nome para acesso é **h2-console**, conforme defenido no **pom.xml** e a porta de escuta é a **8082**.

#### 3.1 Erro: novo usuário

No roteiro da aula prática sugere a implementação do método **run**, da forma como esta exposto na listagem 1. Na linha 19 e 20 na criação de um novo usuário é indicado o primeiro campo como **null**, porém o mesmo não é aceito, conforme documentação.

```

1 package config;
2
3 import java.util.Arrays;
4 import org.springframework.beans.factory.annotation.Autowired;
5 import org.springframework.boot.CommandLineRunner;
6 import org.springframework.context.annotation.Configuration;
7
8 import entities.User;
9 import repositories.UserRepository;
10
11 @Configuration
12 public class TestConfig implements CommandLineRunner{
13
14     @Autowired
15     private UserRepository userRepository;
16
17     @Override
18     public void run(String... args) throws Exception {
19         User u1 = new User(null, "Maria Brown", "maria@gmail.com",
20             , "9888888888", "123456");
21         User u2 = new User(null, "Alex Green", "alex@gmail.com",
22             "977777777", "123456");
23         userRepository.saveAll(Arrays.asList(u1, u2));
24     }
25 }

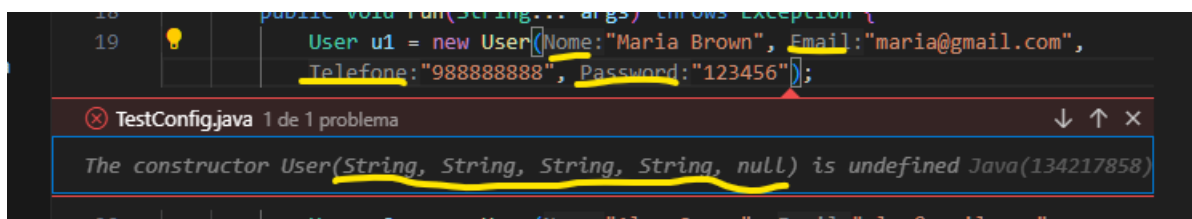
```

**Listagem 1.** Erro: novo usuário

Fonte: O autor (2024)

Na figura 4, a IDE expõe o que é esperado pelo construtor.

**Figura 4.** Argumentos demonstra que o construtor *User* espera.



Fonte: O autor (2024)



Desta forma, o código foi alterado para o que esta na listagem 2.

```
1      public void run(String... args) throws Exception {  
2          User u1 = new User("Maria Brown", "maria@gmail.com", "  
          9888888888", "123456");  
3          User u2 = new User("Alex Green", "alex@gmail.com", "  
          9777777777", "123456");
```

**Listagem 2.** *Construtor User, alterado*

Fonte: O autor (2024)

## 4 Conclusões

dsfdg

## Referências

ALVARON. **Empresas que não têm sites perdem muitos clientes para concorrentes**. 2024. Acessado em: 01 fev. 2024. Disponível em: <<https://alvaron.com.br/empresas-que-nao-tem-sites-perdem-muitos-clientes-para-concorrentes/>>.