

[Spring Boot]

# [Testes unitários em API REST para gerenciar estoque de cervejas]

---

[Rodrigo Peleias]

[Desenvolvedor de Software Senior]

# Quero conhecer vocês :)

# Objetivos da Aula

1. Apresentar a pirâmide de testes de software e detalhar cada nível
2. Destacar a importância dos testes unitários durante o desenvolvimento
3. Apresentar frameworks referência para testes: JUnit, Mockito e Hamcrest
4. Codificar, compartilhar e aprender todos juntos :)

# O que vamos utilizar

- ✓ Java 14
- ✓ Maven 3.6.3
- ✓ Spring Boot (última versão estável lançada)
- ✓ GIT/ GITHUB para versionamento de código
- ✓ Frameworks JUnit, Mockito e Hamcrest

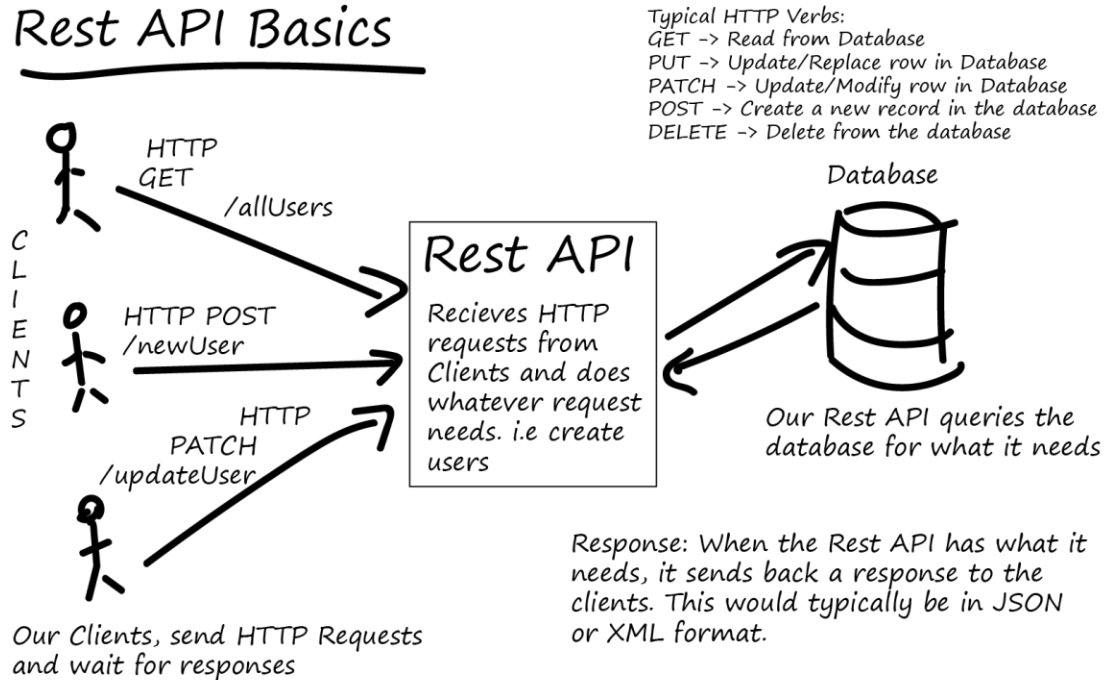
# Bora aprender e nos divertir :)

## [Spring Boot]



# Padrão arquitetural REST

## Rest API Basics





DIGITAL  
INNOVATION  
ONE

# API RESTful - Richardson



GLORY OF REST



LEVEL 3

HYPERMEDIA CONTROLS

LEVEL 2

HTTP VERBS

LEVEL 1

RESOURCES

LEVEL 0

THE SWAMP OF POX



# Verbos HTTP

Uri	Method	Description	Request Stream	Response Stream	Status Code Returned
/people	GET	All people in the system	n/a	<b>Person</b> collection	200/404
/people/{id}	GET	Get specific <b>Person</b>	n/a	<b>Person</b>	200/404
/people	POST	Creates a new entity <b>Person</b> entity in the system. Expects a representation of the person in the body	<b>Person</b> without the Id specified	<b>Person</b>	201/404
/people/{id}	PUT	Modifies a <b>Person</b> resource	Person	n/a	200/404
/people/{id}	DELETE	Deletes a <b>Person</b> resource	n/a	n/a	200/404





DIGITAL  
INNOVATION  
ONE

# Pirâmide de testes



High Integration  
Slow Execution  
Lesser in Number



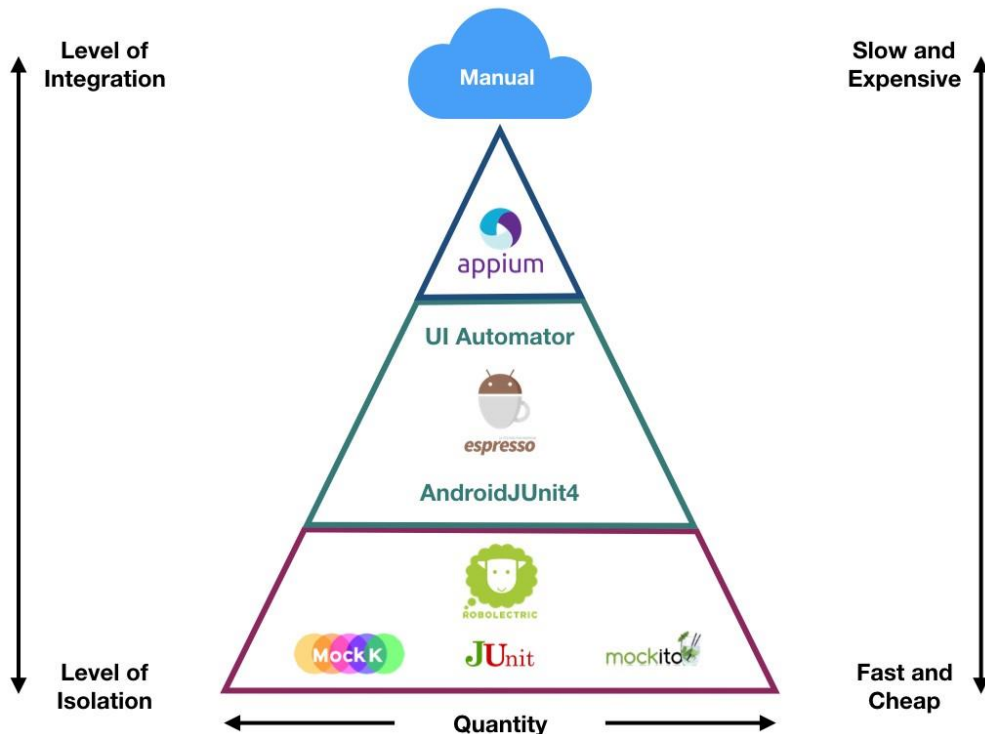
High Isolation  
Fast Execution  
Higher in Number

UI Tests

Integration Tests

Unit Tests

# Pirâmide de testes



# Pirâmide de testes

- ✓ Sistema testado de ponta a ponta!!
- ✓ Evolução segura: sem quebrar funcionalidades!
- ✓ Teste também é forma de documentação!
- ✓ Integração contínua (CI)
- ✓ Deploy contínuo: (CD)

# Nível 1: testes unitários

- ✓ Maior número de testes, menor custo e tempo
- ✓ Testes feito pelo próprio desenvolvedor
- ✓ Rápidos, com base em linhas de código
- ✓ Cobertura de vários cenários para as linhas
- ✓ Integração com outros códigos: através de mocks

# Principaux frameworks

## ✓ JUnit

```
import static org.junit.jupiter.api.Assertions.assertEquals;

import example.util.Calculator;

import org.junit.jupiter.api.Test;

class MyFirstJUnitJupiterTests {

    private final Calculator calculator = new Calculator();

    @Test
    void addition() {
        assertEquals(2, calculator.add(1, 1));
    }

}
```

# Principais frameworks

## ✓ Mockito

now you can verify interactions

```
import static org.mockito.Mockito.*;

// mock creation
List mockedList = mock(List.class);

// using mock object - it does not throw any "unexpected interaction" exception
mockedList.add("one");
mockedList.clear();

// selective, explicit, highly readable verification
verify(mockedList).add("one");
verify(mockedList).clear();
```

and stub method calls

```
// you can mock concrete classes, not only interfaces
LinkedList mockedList = mock(LinkedList.class);

// stubbing appears before the actual execution
when(mockedList.get(0)).thenReturn("first");

// the following prints "first"
System.out.println(mockedList.get(0));

// the following prints "null" because get(999) was not stubbed
System.out.println(mockedList.get(999));
```



# Principaux frameworks

## ✓ Hamcrest

```
import org.junit.jupiter.api.Test;
import static org.hamcrest.MatcherAssert.assertThat;
import static org.hamcrest.Matchers.*;

public class BiscuitTest {
    @Test
    public void testEquals() {
        Biscuit theBiscuit = new Biscuit("Ginger");
        Biscuit myBiscuit = new Biscuit("Ginger");
        assertThat(theBiscuit, equalTo(myBiscuit));
    }
}
```

# Principais frameworks

✓ Spring Boot Starter Test: todos os frameworks!!

```
<dependency>
<groupId>
    org.springframework.boot
</groupId>
<artifactId>
    spring-boot-starter-test
</artifactId>
<scope>
    test
</scope>
</dependency>
```

- ◀ Includes testing libraries
- ◀ JUnit
- ◀ Mockito
- ◀ Hamcrest
- ◀ Spring core
- ◀ Spring test



# Dúvidas?

[Spring Boot]

# Obrigado!!

[Spring Boot]