DIGITAL INNOVATION ONE

[Spring Boot]

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

[Rodrigo Peleias]
[Desenvolvedor de Software Senior]

DIGITAL INNOVATION ONE

Quero conhecer vocês :)

DIGITAL INNOVATION ONE

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 :)

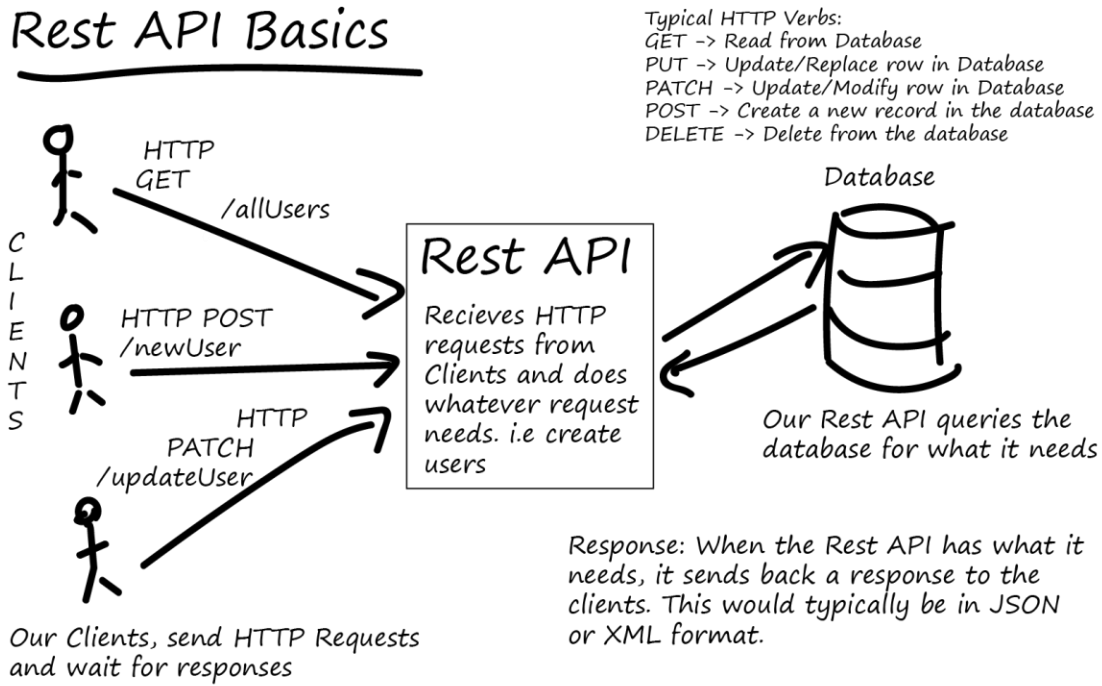
O que vamos utilizar

- ✓ 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 :)

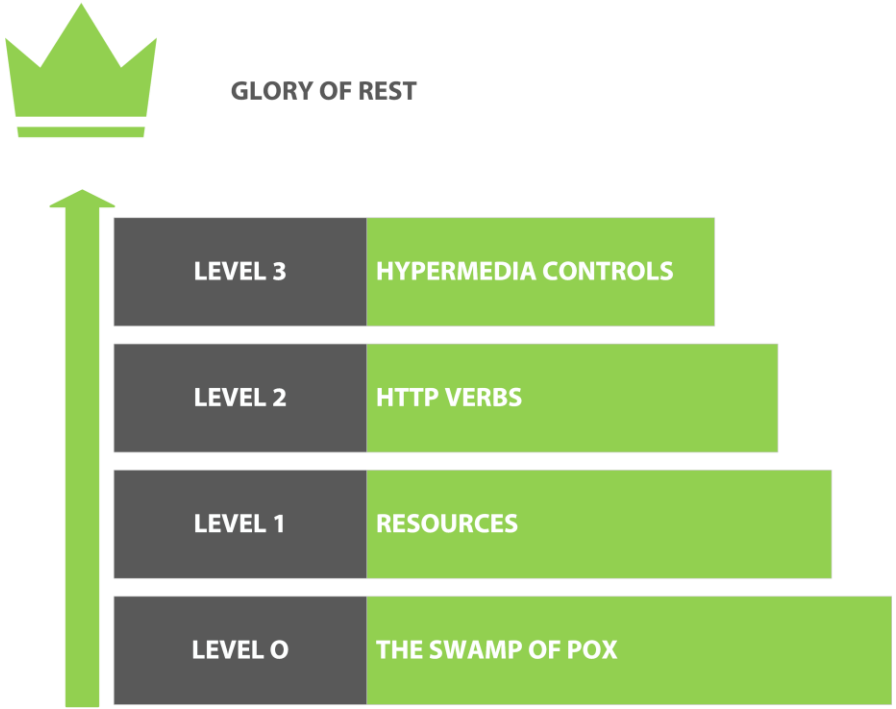
[Spring Boot]

Padrão arquitetural REST



DIGITAL INNOVATION ONE

API RESTful - Richardson



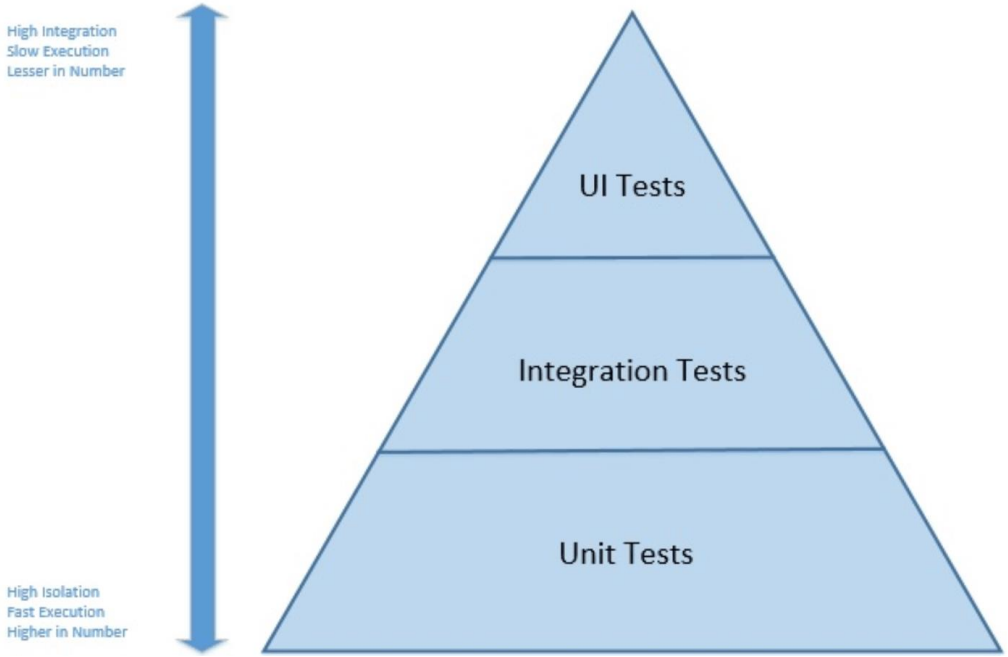
DIGITAL INNOVATION ONE

Verbos HTTP

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

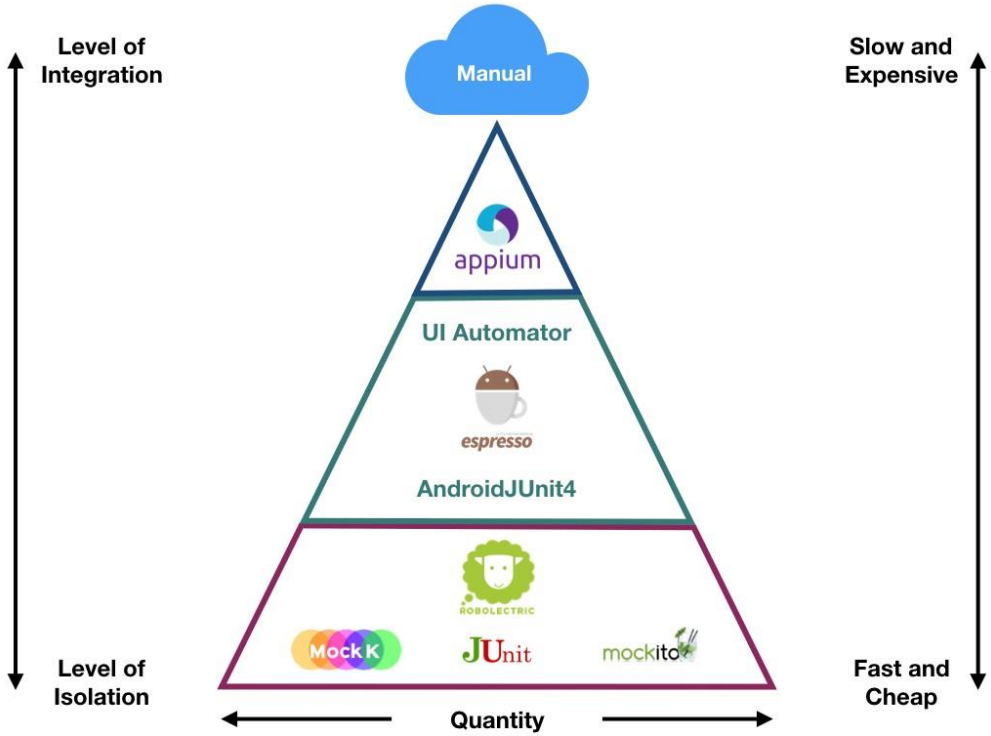
DIGITAL INNOVATION ONE

Pirâmide de testes



DIGITAL INNOVATION ONE

Pirâmide de testes



DIGITAL INNOVATION ONE

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)

DIGITAL INNOVATION ONE

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

DIGITAL INNOVATION ONE

Principais frameworks

- ✓ JUnit

```java
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));
```

Principais frameworks

- ✓ Hamcrest

```java
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]

DIGITAL
INNOVATION
ONE

Obrigado!!

[Spring Boot]