



SÃO
PAULO
TECH
SCHOOL

Técnicas de Programação Web

Introdução a testes de Software

Automatizados

José Yoshiriro

Diego Brito

jose.yoshiriro@sptech.school

diego.lima@sptech.school

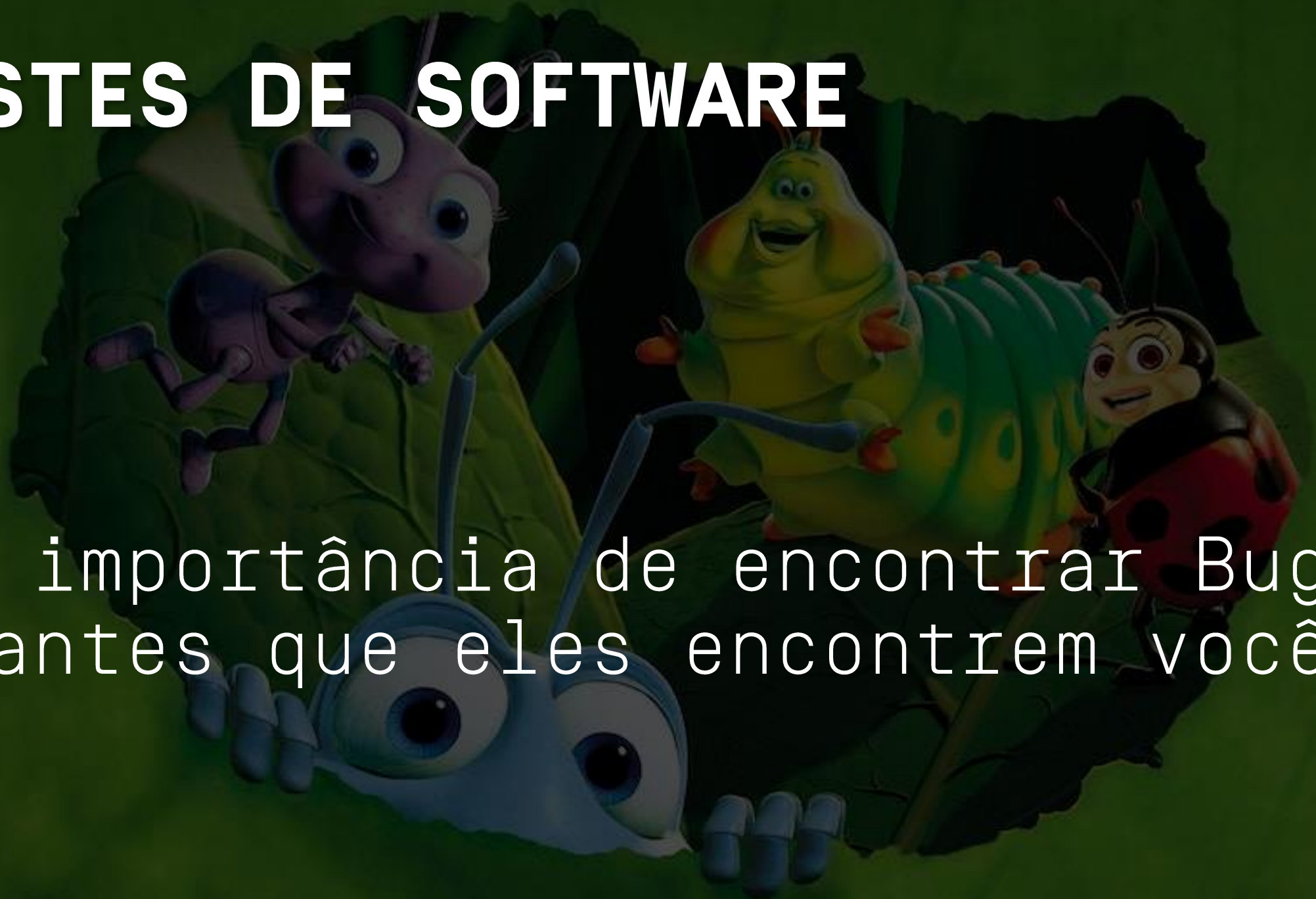
A photograph of a person's hand pointing at a laptop screen. The person is wearing a black smartwatch on their left wrist. The laptop is open, and the screen displays some content. The background is blurred, showing what appears to be a desk and some office equipment. A large white curved shape on the right side of the image contains the title and list.

Tópicos da aula

- Contexto Geral sobre Testes Automatizados
- Tipos de Testes
- Ferramentas
- TDD (Desenvolvimento Orientado a Testes)

TESTES DE SOFTWARE

A importância de encontrar Bugs
antes que eles encontrem você



O que são testes automatizados?

Testes automatizados são testes de software que são executados automaticamente por ferramentas especializadas, em vez de serem executados manualmente por uma pessoa, ou seja, são **programas** criados para **testar** outro **programa**.

Eles ajudam a garantir a qualidade do software, permitindo que os desenvolvedores detectem e corrijam erros de forma rápida e eficiente.

O que são testes automatizados?

Eles também ajudam a reduzir o tempo e os custos de teste, permitindo que os desenvolvedores executem testes com mais frequência e em menos tempo do que seria possível manualmente.

Os testes automatizados podem incluir testes unitários, testes de integração, testes de sistema e testes de aceitação do usuário, entre outros tipos.

Eles são essenciais para a prática de TDD (Desenvolvimento Orientado por Testes), que envolve escrever testes antes de escrever o código, para garantir que o código seja testado desde o início.

Por que escrever testes:

Qualidade **sem testes** é **pura SORTE!**

Os erros podem ser tão graves a ponto de comprometer a segurança do usuário ou a privacidade de dados, resultando em consequências legais e financeiras.

Sobre testes:

Testes manuais tem sua importância, **mas os automatizados são indispensáveis.**

Teste Automatizados

+ Velocidade;

- Chances de erro;

Evidências da existência e execução dos testes;

O programa sempre será executado, **independente de:**

- Cansaço;
- Ansiedade;
- Preguiça ou má fé;

Testes Manuais

- Velocidade;

+ Chances de erro;

Evidências ficam restritas ao profissional;

A execução manual de testes é **suscetível** a:

- Cansaço
- Ansiedade
- Preguiça ou má fé;

Por que escrever testes:

Testes Automatizados x Manuais

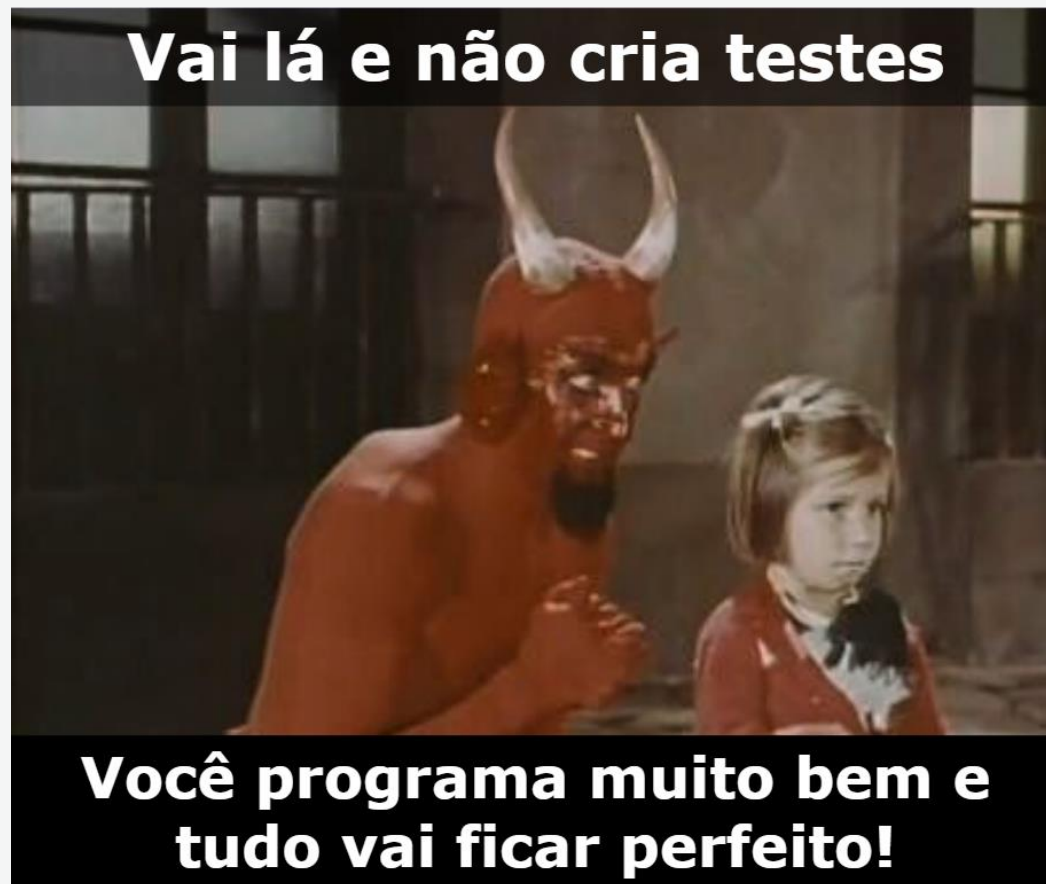
"Será que isso mexe afeta algo que já existia?"

- Executa os testes de todo o projeto!

"E agora para testar tudo o que foi desenvolvido no projeto?"

- Executa os testes de todo o projeto!

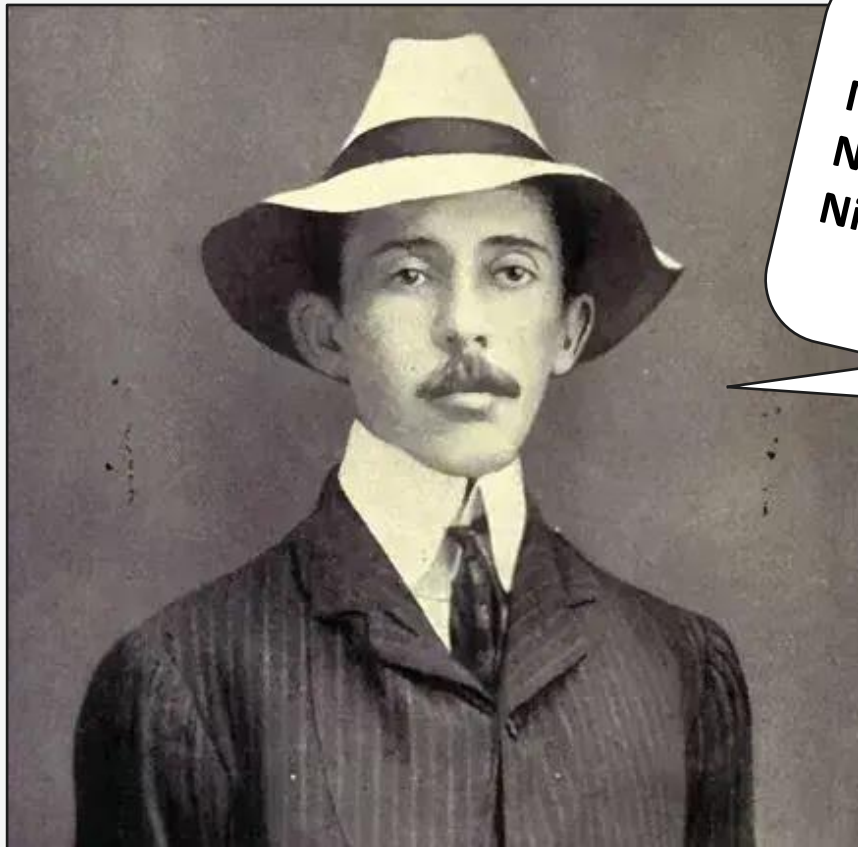
Por que escrever testes:



É por volta das **18h** das **sextas-feiras** que costumamos lembrar que não somos nenhum **Leonardo da Vinci**.

Por que escrever testes:

Santos Dumont – Pioneiro na aviação



Mano, acabei de fazer um avião. Rolezin?
Não testei, mas sou diferenciado, CONFIA!
Ninguém vai de comes e bebes, rlx!

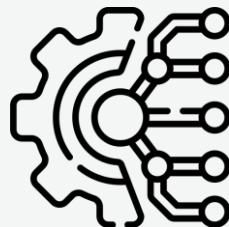
Tipos de testes automatizados

The background is a dark, almost black, space filled with intricate, glowing patterns. These patterns consist of numerous thin, white, curved lines that flow and swirl across the frame, creating a sense of dynamic movement. Interspersed among these lines are many small, bright white dots and some larger, fainter circular bokeh-like shapes, giving the impression of a starry field or a complex data visualization.



Unitários ou de unidade

Teste unitário é um tipo de teste de software em que cada parte individual do código é testada sozinha para verificar se funciona corretamente antes de ser combinada com outras partes.



Integração

Teste de integração é uma técnica de teste de software em que diferentes componentes do sistema são combinados e testados juntos para verificar se funcionam corretamente e se integram sem problemas.



Funcionais

Teste funcional é uma técnica de teste de software em que o sistema é testado em relação às especificações de requisitos funcionais, ou seja, os testes são realizados com base nas funcionalidades esperadas do software.

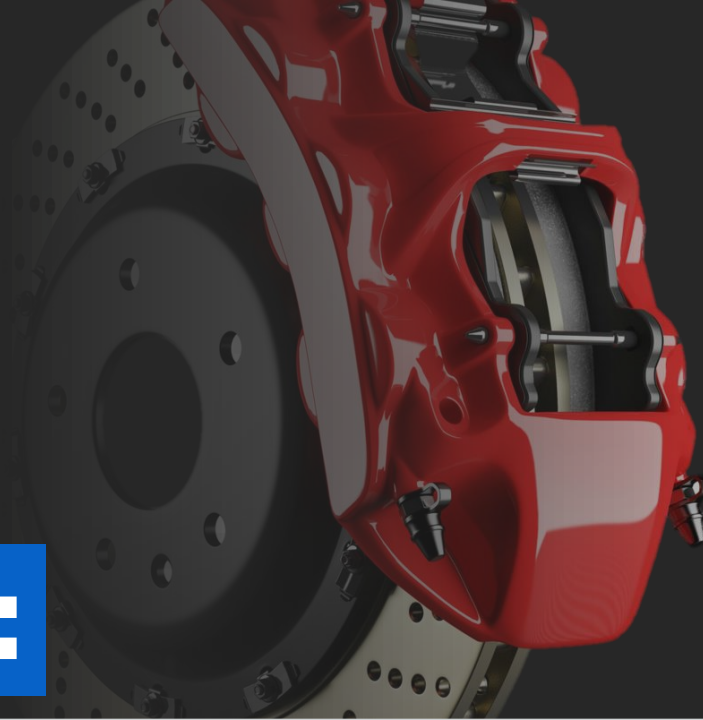
Testes unitários [ou de unidade] :

Equivalente a testar um método de uma classe.



Testes de integração:

Equivalente a testar uma funcionalidade que usou várias classes (não o sistema completo)



Testes funcionais [ou de sistema] :

Testar se a saída (ou resposta) de uma funcionalidade do sistema completo funciona como esperado



Ferramentas

Existem várias bibliotecas para escrever testes unitários na plataforma Java, alguns exemplos são:

- **JUnit – mais popular, utilizaremos essa**
- Spock
- Test NG

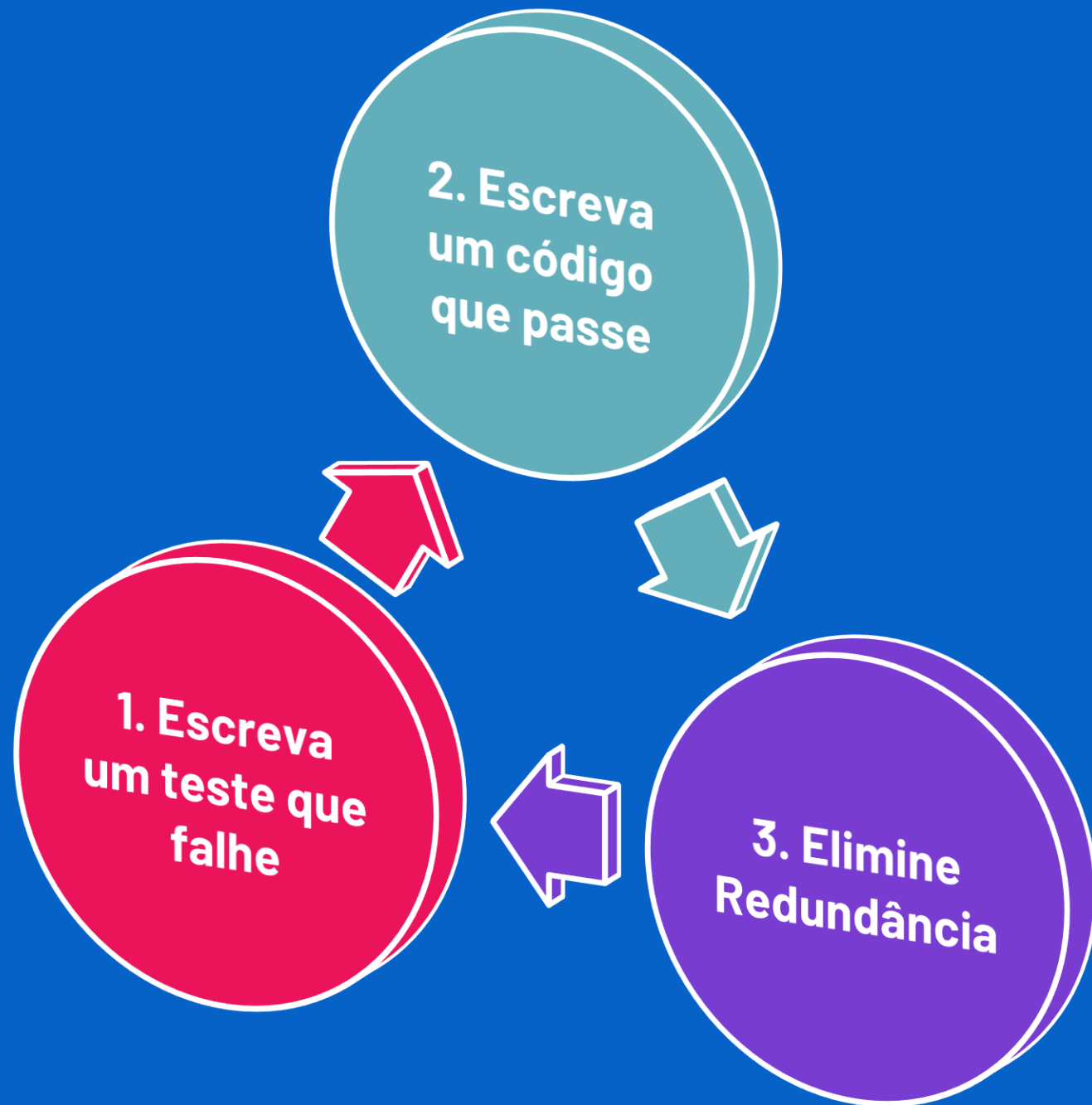
Vale lembrar que os testes podem ser escritos em outra linguagem da plataforma!

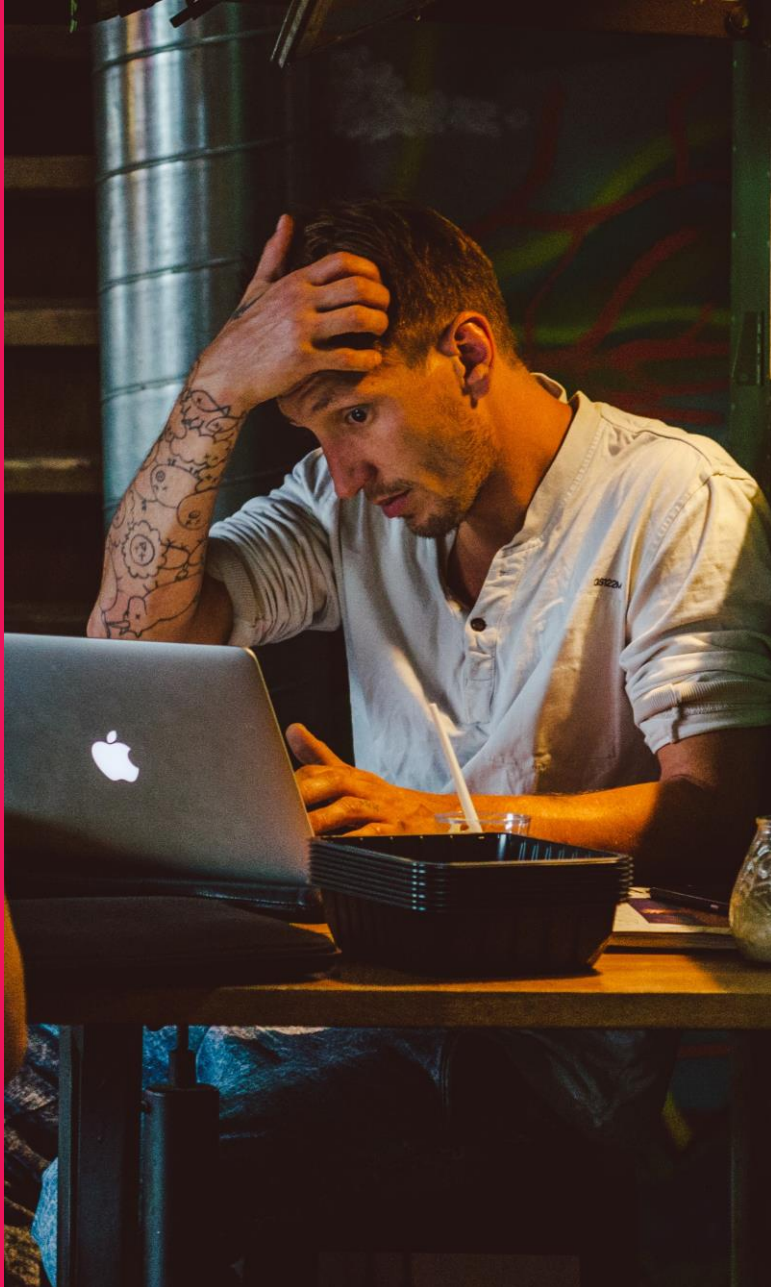


Test Driven Development

- O TDD é uma prática de desenvolvimento de software em que o programador escreve testes automatizados **antes mesmo de começar a escrever o código do programa.**
- O TDD é uma abordagem eficaz para desenvolver software mais confiável, com menos erros e que possa ser facilmente mantido no futuro.

Como funciona?





Implicações

- Mudança de mentalidade
- Aprendizado da técnica
- Integração com outros processos
- Investimento em ferramentas
- Resistência à mudança

Prazo/Orçamento

Agradeço a sua atenção!

Por:

- **José Yoshiriro**
- **Diego Brito**

diego.lima@sptech.school



SÃO
PAULO
TECH
SCHOOL