



# BOOTCAMP DE INTRODUÇÃO AO MERCADO DE TRABALHO DE AUTOMAÇÃO DE TESTES

Com CypressJS e Cucumber (BDD)



# Eu sou Angelo F. Dias

- Cientista da computação (Bacharel);
- Engenheiro da Computação pela Escola Politécnica de Pernambuco da Universidade de Pernambuco – UPE (Mestre);
  - Inteligência Artificial e Mineração de Dados
- Mais de 14 anos inserido no mercado de trabalho;
- Professor da UNIESP (Graduação e Pós-Graduação);
- Analista QA Sênior;
  - Programador por opção.



# Bootcamp

O termo **Bootcamp** ou “Campo de Treinamento” nasceu nos *Campos Militares dos Estados Unidos*, onde acontece a preparação de soldados para a guerra.

Ele *acabou* se tornando comum em Educação para definir um tipo de aprendizagem **intensiva** e acelerada, nas mais diversas áreas.

## Intensivão do Angelão!



# BOOTCAMP DE INTRODUÇÃO AO MERCADO DE TRABALHO DE AUTOMAÇÃO DE TESTES

Com CypressJS e Cucumber (BDD)



# ENTRANDO NO BOOTCAMP



01

## DESCONECTE-SE

Colocar o  
celular no  
modo avião\*

02

## CONCENTRE-SE

Técnica/Método  
Pomodoro  
(4 x 25/5)

03

## SUA MISSÃO

BDD +  
Cucumber +  
CypressJS

04

## SOMOS UM TIME

Estamos todos  
aprendendo  
juntos.



# /SUMÁRIO



**/01**

## **/Práticas ágeis**

Spotify, Pirâmide de testes, BDD

**/02**

## **/BDD - BEHAVIOUR DRIVEN DEVELOPMENT**

Conceitos, Linguagem Gherkin

**/03**

## **/BDD - FERRAMENTAS**

Cucumber (Com Cypress), VSCode (Plugins)

**/04**

## **/AUTOMAÇÃO DE TESTES COM CYPRESS**

**/05**

## **/AUTOMAÇÃO DE TESTES COM CYPRESS E CUCUMBER**

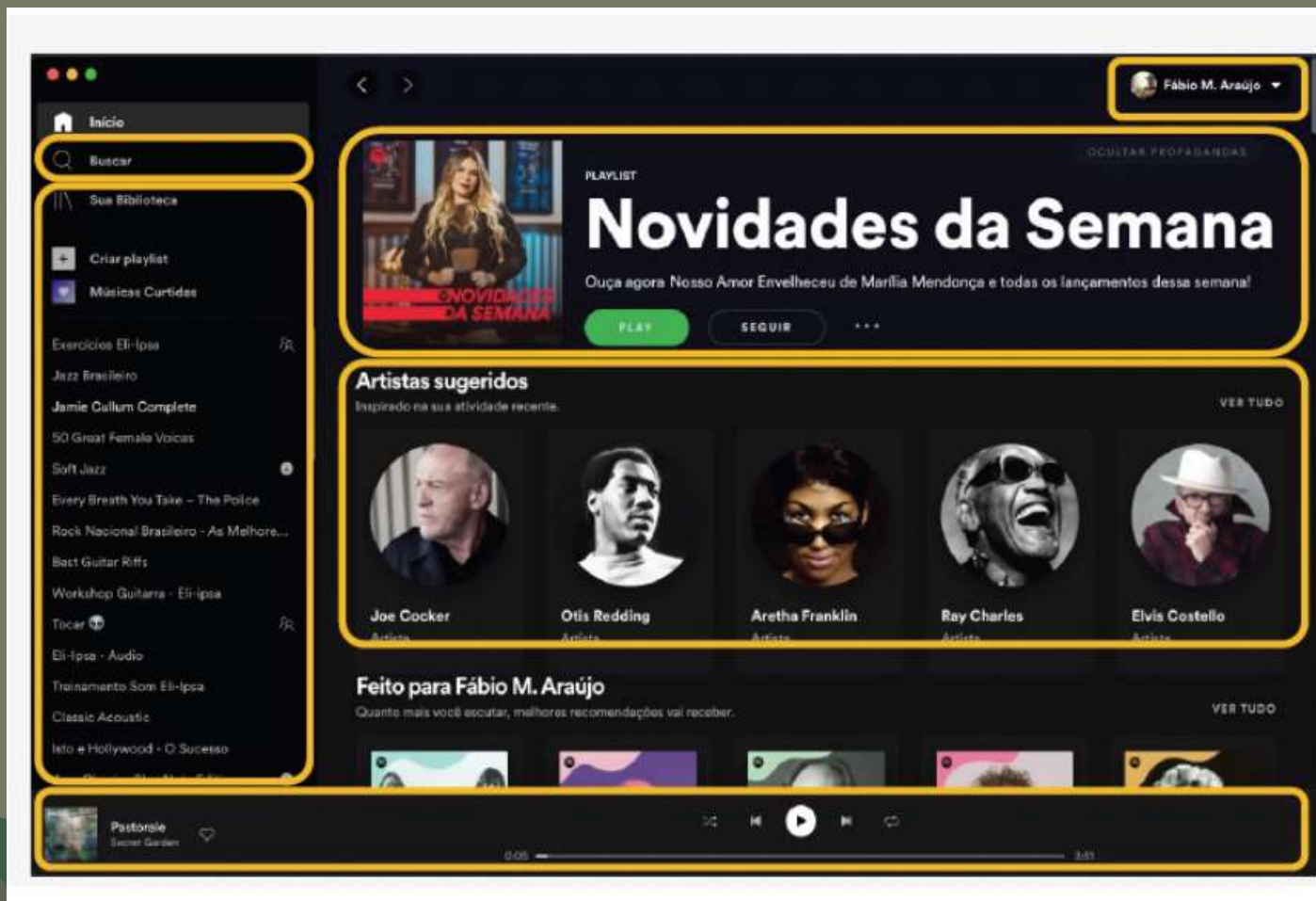
Com extras (se der tempo)



**/PRÁTICAS ÁGEIS**

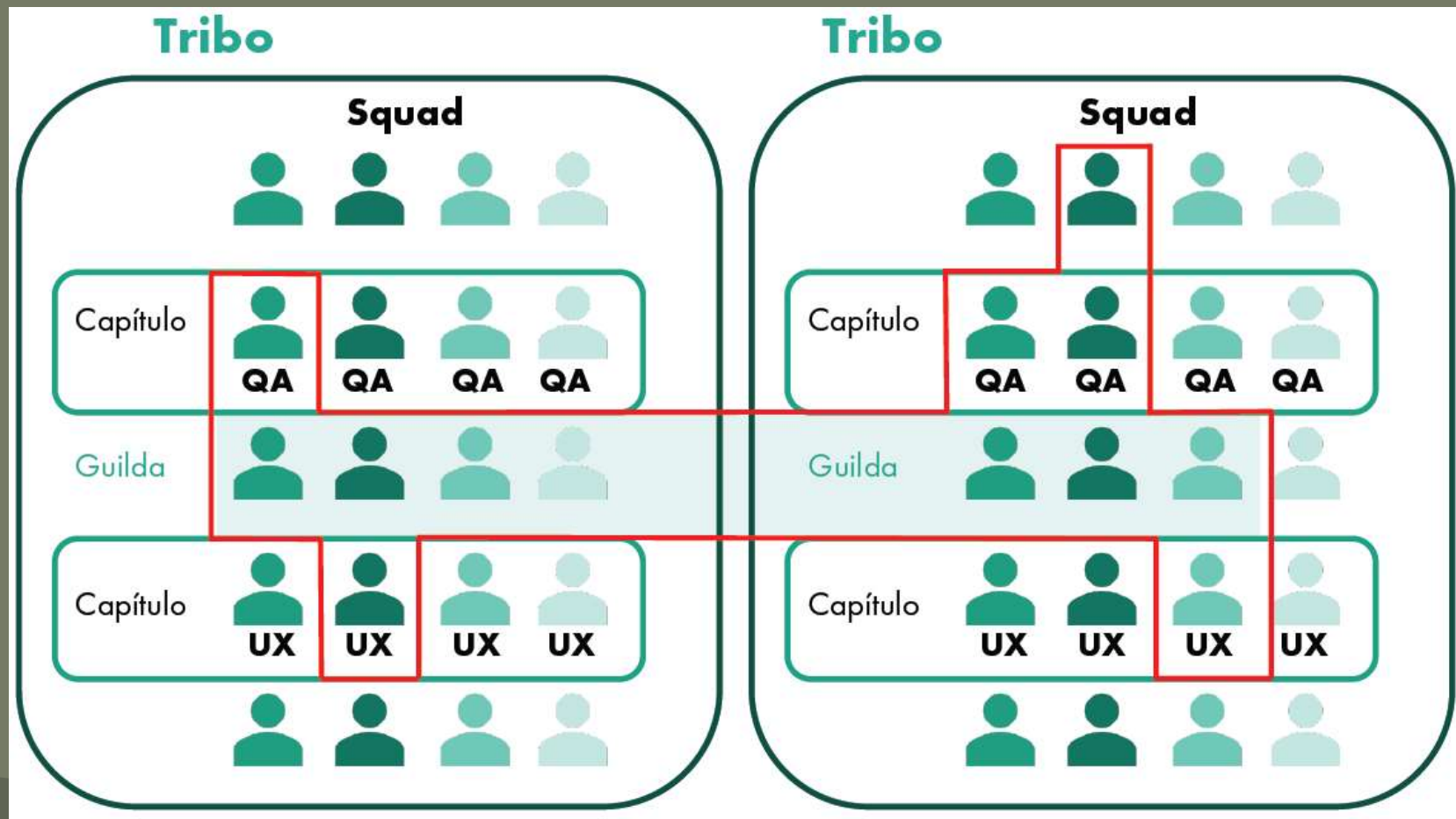


# /MODELO SPOTIFY





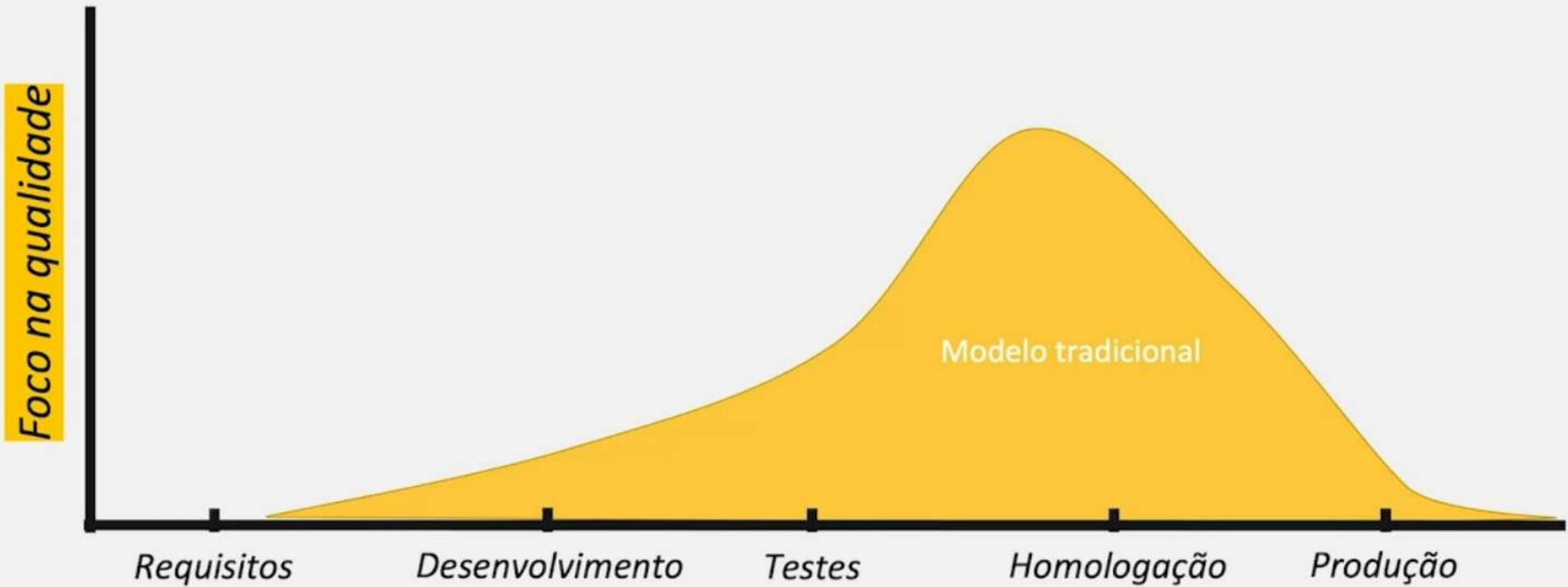
# /MODELO SPOTIFY



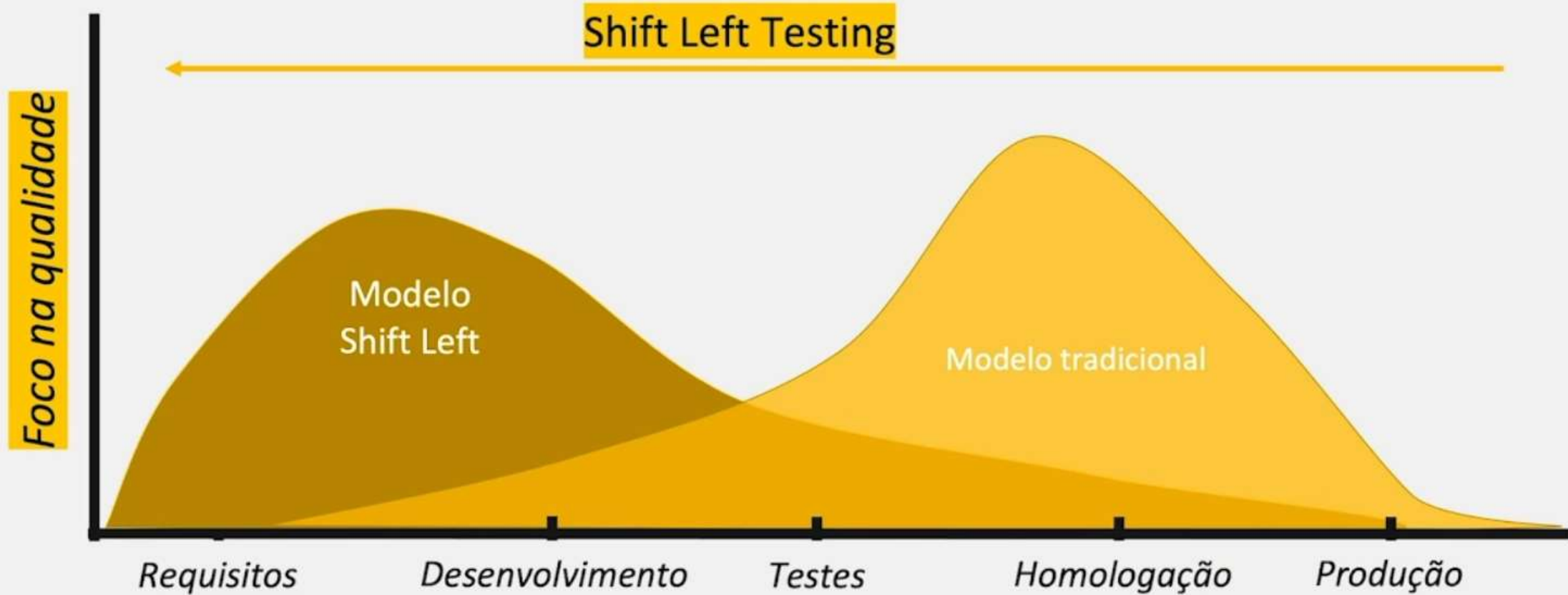
# /CAPÍTULO

- Desenvolver pessoas;
- Fomentar a cultura;
- Buscar apoios entre as Squads;
- Buscar capacitação de pessoas;
- Equalizar o conhecimento;
- Promover ações de engajamento;

# /SHIFT LEFT TESTING



# /SHIFT LEFT TESTING



# /PAREAMENTO

- Revisar testes em conjunto com o desenvolvedor para aumentar a cobertura de testes e disseminar técnicas de teste;
- Técnica do XP (Extreme Program) também chamado de “Pair Program”.

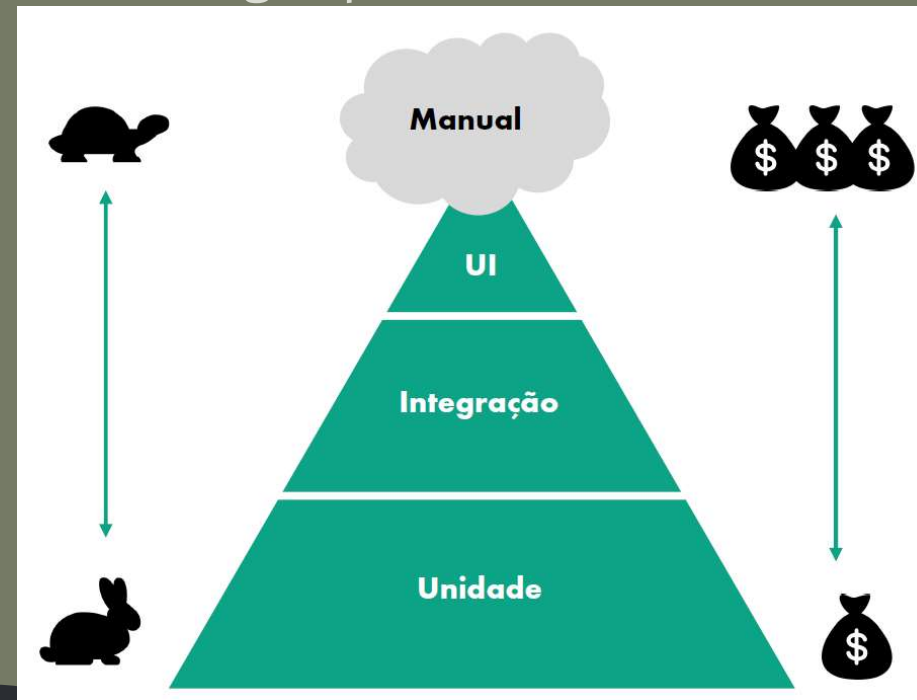


# **/PIRÂMIDE DE TESTES**

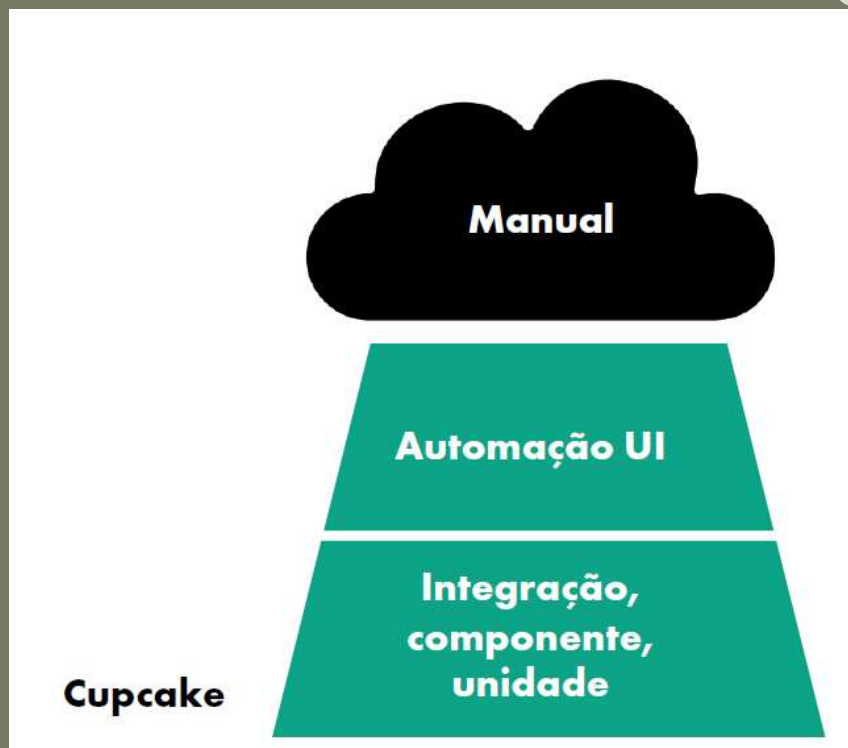


# /PIRÂMIDE DE TESTES

- A "Pirâmide de Testes" dá uma ideia de quantos testes devemos ter em cada um desses grupos ou camadas.

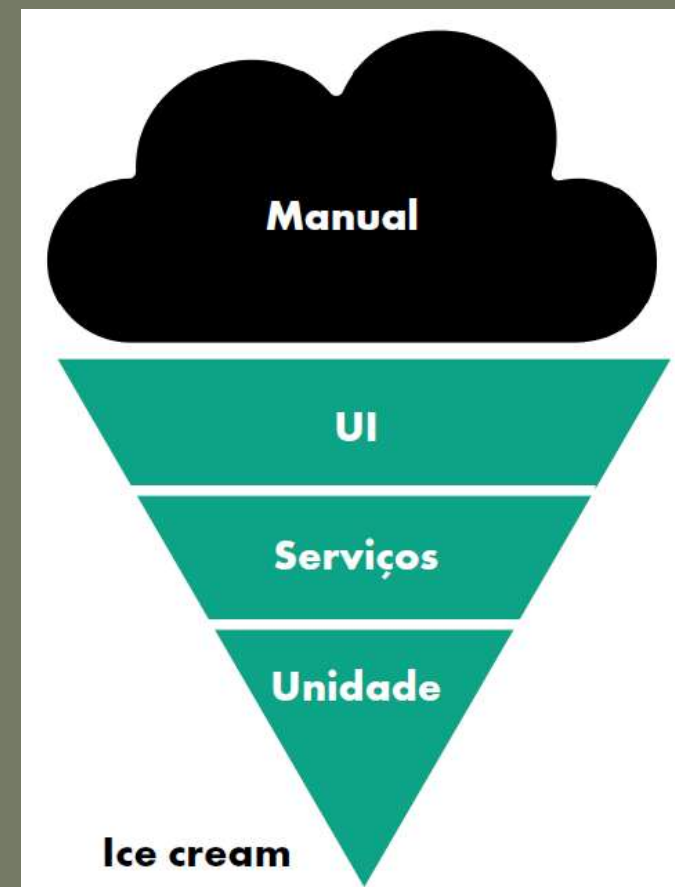
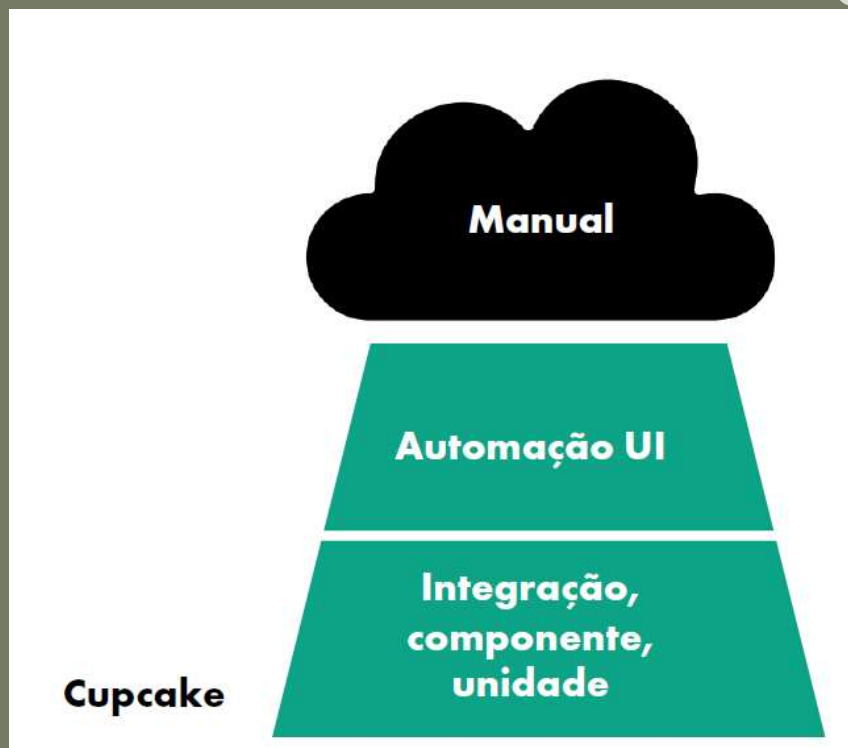


# /ANTI - PIRÂMIDE DE TESTES (EVITAR)





# /ANTI - PIRÂMIDE DE TESTES (EVITAR)



# **/BDD - BEHAVIOUR DRIVEN DEVELOPMENT**



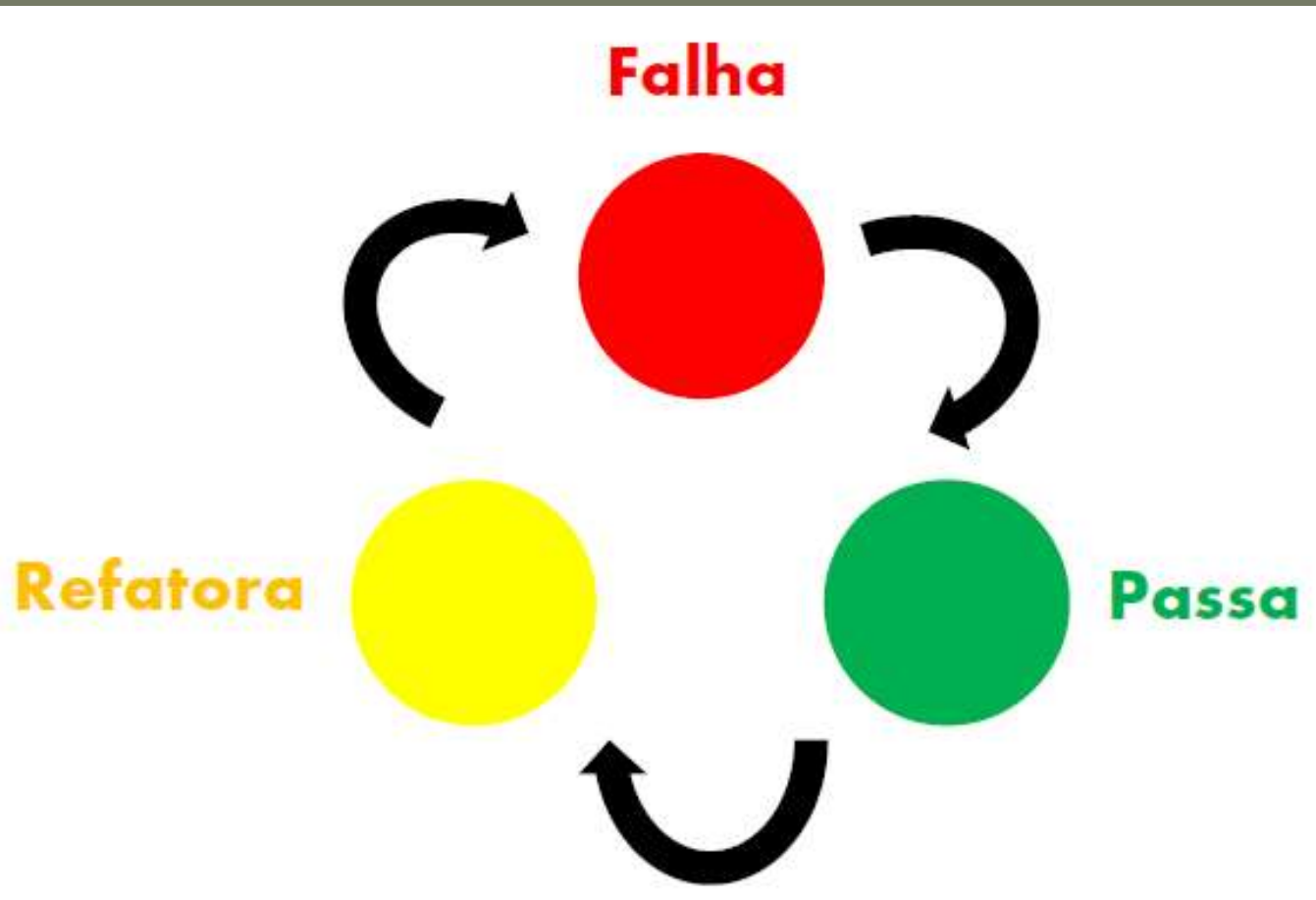
# /BDD – O QUE É?

- Behaviour Driven Development (Desenvolvimento orientado a comportamento);
- Criado por Dan North;
- É uma abordagem para desenvolvimento de software que se apoia em testes e comportamentos do usuário ou sistema;
- O BDD é uma evolução feita a partir da implantação de técnicas TDD (Test Driven Development);

<https://dannorth.net/>

# /E O TDD?

- Beha
- orie
- Cria
- É un
- soft
- do u
- O BD
- de t

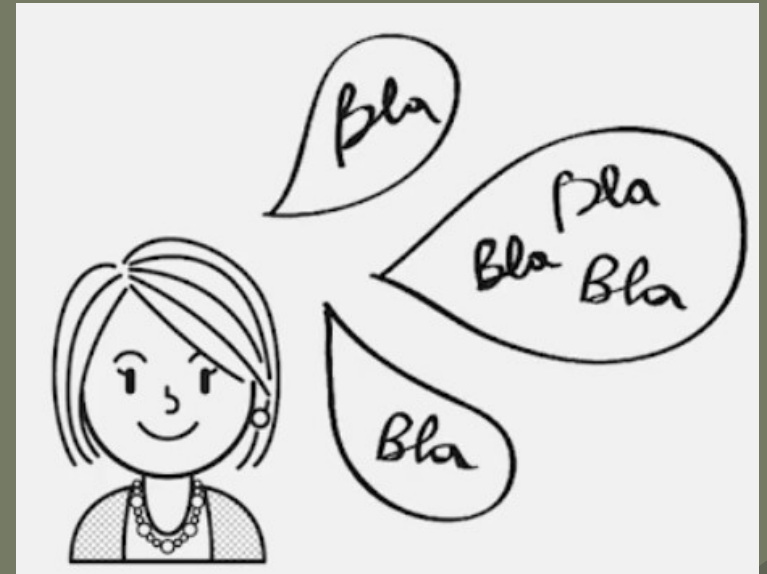


# /BDD – POR QUE?

- Melhora a comunicação entre desenvolvedores e profissionais de negócios;
- Assegura que toda a equipe esteja focada em entregar o necessário para alcançar seus objetivos;
- DevTeam como dono da solução, pois o time prover a solução técnica;
- Acelera a criação dos testes automatizados;
- Documentação viva.

# /FOCO NO USUÁRIO

- Conversa com usuário:
  - “Se eu fizer isso e isso, eu espero esse resultado”
  - Dado uma condição
  - Quando eu executo uma ação
  - Então o resultado é esse



# /BDD – COMO É?

1- **P.O** e **Usuário(s)**  
tem uma conversa sobre  
o que desejam  
desenvolver.



2- **P.O** junto com o **DevTeam**  
elaboram o requisito e  
definem a estrutura dos  
cenários.



3- Os cenários guiam o  
desenvolvimento...



4- **Usuários** usam os  
cenários para  
homologar.

# /TÉCNICA “OS TRÊS AMIGOS”

- Os Três Amigos refere-se a uma reunião dos três principais papéis envolvidos na produção de software:





# /USOS DO BDD

- Podemos dividir o uso do BDD em duas grandes partes:
  - • 1 – Como uma linguagem ubíqua para ilustrar o comportamento do usuário na aplicação;
  - • 2 – Como base para automação de testes, garantindo que a aplicação funcione como previsto por todos;

## /USOS DO BDD

- Juntando essas duas partes temos uma documentação viva, que não só explica o comportamento do software, mas também executa testes, documenta código e auxilia na homologação.

# /BDD – GHERKIN

# /BDD - GHERKIN

- O vocabulário comum em BDD é o Gherkin.
- Gherkin é uma linguagem que foi criada especialmente para descrições de comportamento.
- Remove detalhes da lógica de programação e foca no comportamento que uma funcionalidade deve ter.

# /BDD - GHERKIN

- Um arquivo Gherkin contém:
    - Título da funcionalidade;
    - Descrição da funcionalidade;
    - Cenários, que descrevem uma interação do usuário;
    - Pré requisitos (Dado);
    - Ações (Quando)
    - Resultado esperado (Então).
- Referencia: <https://cucumber.io/>

# /IMPORTANTE!

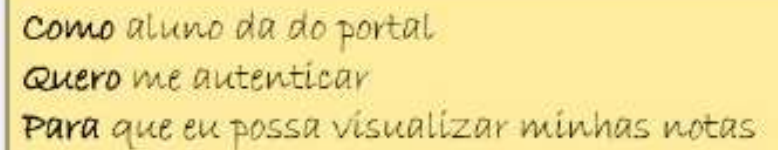
- Gherkin não é caso de teste!
- O Gherkin deve prever uma ação de comportamento de forma declarativa e não imperativa como no caso de teste;
- Deve dar uma visão de cenários, conciso e direto ao ponto:

# /IMPORTANTE!

- Ex.: Cenário: Buscar por “Qualidade” no Google
- Imperativo:
  - Dado que eu abra o navegador
  - E acesse o google com
  - E clicar no “search”
  - Quando eu buscar pela palavra “Qualidade”
  - E clicar no botão “buscar”
  - Então deve exibir a lista com os resultados
- Declarativo:
  - Dado que eu acesse o google
  - Quando eu buscar pela palavra “Qualidade”
  - Então deve exibir a lista com os resultados

# /EXEMPLO DE ESCRITA!

- Isso já pode ser uma documentação viva dentro do BDD, mas podemos melhorar explicando esse recurso, dividindo em cenários por exemplo:
  - • Autenticação válida;
  - • Usuário inexistente;
  - • Usuário com senha inválida;
  - • Etc...



Como aluno da do portal  
Quero me autenticar  
Para que eu possa visualizar minhas notas



# /CENÁRIOS

- Cenário 1 – Autenticação Válida
  - Dado que eu acesse a página de autenticação do portal TECH\*
  - Quando eu digitar o usuário angelofdiasg@angelofdiasg.tech
  - E a senha “abc@123”
  - Então deve ser exibido a mensagem de boas vindas: “Olá Professor”.

# /CENÁRIOS

- Cenário 2 – Usuário Inexistente
  - Dado que eu acesse a página de autenticação do portal TECH\*
  - Quando eu digitar o usuário `gdiasangelo@angelofdiasg.tech`
  - E a senha “abc@123”
  - Então deve ser exibido a mensagem “Usuário não cadastrado em nossa base”

# /CENÁRIOS

- Cenário 3 – Senha Inválida
  - Dado que eu acesse a página de autenticação do portal TECH\*
  - Quando eu digitar o usuário angelofdiasg@angelofdiasg.tech
  - E a senha “123456”
  - Então deve ser exibido a mensagem “Usuário ou senha inválidos”

# /CENÁRIOS USANDO CONTEXTO

- Contexto: Dado que eu acesse a página de autenticação do portal TECH\*
  - Cenário: 1 - Autenticação válida
  - Quando eu digitar o usuário angelofdiasg@angelofdiasg.tech
  - E a senha “abc@123”
  - Então deve ser exibido a mensagem de boas vindas: “Olá Professor”.

# /CENÁRIOS USANDO CONTEXTO

- Esquema do Cenário: Autenticação válida
  - Quando eu digitar o <usuario>
  - E a <senha>
  - Então deve exibir a <mensagem> de boas vindas

## **Exemplos:**

usuario	senha	mensagem
"joao@ebac.com.br"	"123@abc"	"Olá João"
"maria@ebac.com.br"	"123@abc"	"Olá Maria"
"felipe@ebac.com.br"	"123@abc"	"Olá Felipe"
"flavio@ebac.com.br"	"123@abc"	"Olá Flávio"
"ana@ebac.com.br"	"123@abc"	"Olá Ana"

# /CENÁRIOS USANDO TABELA DE DADOS

- Funcionalidade: Edição de alunos na plataforma
  - Cenário: Editar dados do aluno
  - Dado que eu acesse o portal com os seguintes usuários:

usuario	senha
"joao@ebac.com.br"	"123@abc"
"maria@ebac.com.br"	"123@abc"
"felipe@ebac.com.br"	"123@abc"
"flavio@ebac.com.br"	"123@abc"
"ana@ebac.com.br"	"123@abc"

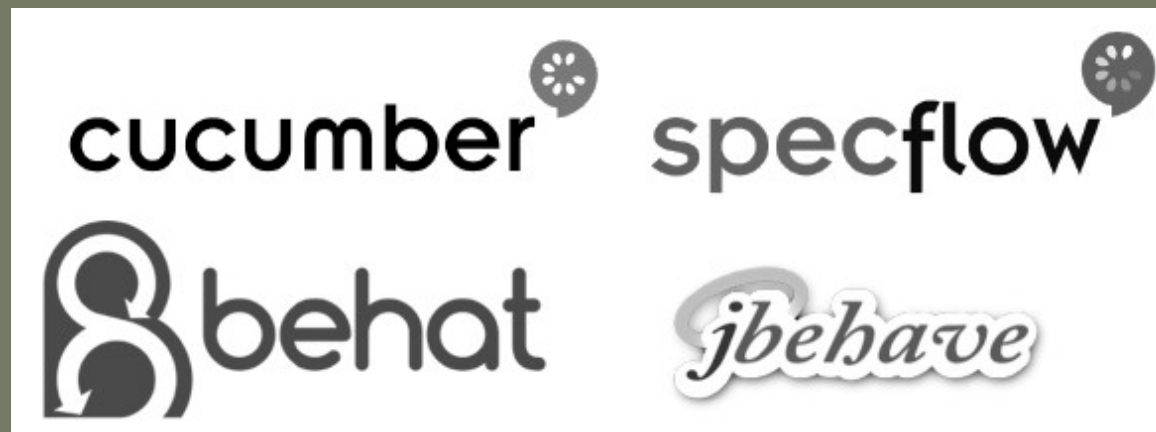
- Quando eu editar o cadastro com os dados obrigatórios
- Então então deve exibir uma mensagem: "Cadastro alterado com sucesso"

# /PALAVRAS RESERVADAS DO GHERKIN

#language: pt	
Feature	Funcionalidade
Scenario	Cenário
Given	Dado
When	Quando
Then	Então
And	E
Or	Ou
But	Mas
Scenario Outline	Esquema do Cenário
Background	Contexto
Examples	Exemplos
...	...

# /FERRAMENTAS PARA O GHERKIN

- Dependendo da linguagem de programação ou framework de automação de testes, algumas ferramentas podem auxiliar no padrão de escrita e até gerando steps automaticamente.





# /BDD – CUCUMBER



# /FERRAMENTAS PARA O GHERKIN

- VSCode
- Cucumber



Visual Studio Code

cucumber 



Cucumber (Gherkin) Full Support

Alexander Krechik | 346.031 | ★★★★★ | Re

VSCode Cucumber (Gherkin) Full Language Support + Formattir

Reload Required

Install



Please reload Visual Studio Code to comp



# BANG!

Vamos ATACAR!

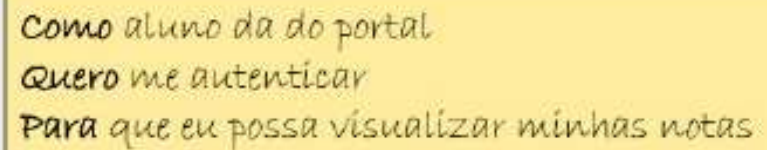
# /FERRAMENTAS PARA O GHERKIN

- login.feature

- <https://www.saucedemo.com/>

- Critérios de avaliação:

- Autenticação válida;
  - Usuário inexistente e com senha inválida;
  - Usuário em branco e senha em branco;



Como aluno da do portal  
Quero me autenticar  
Para que eu possa visualizar minhas notas

# /AUTOMAÇÃO DE TESTES COM BDD

- #language: pt
- Funcionalidade: Login no Swag Labs
- Contexto: Dado que eu acesse a página de autenticação do saucedemo
  - Cenário: Login válido
  - Esquema do Cenário: Logins inválidos com tabela

# /CENÁRIOS

- Cenário 1 – Autenticação Válida
  - Dado que eu acesse a página de autenticação do saucedemo
  - Quando eu digitar o usuário “standard\_user”
  - E a senha “secret\_sauce”
  - Então deve ser exibida a logo do “Swag Labs”.

# /CENÁRIOS

- Cenário 2 – Usuário Inexistente
  - Dado que eu acesse a página de autenticação do saucedemo
  - Quando eu digitar o usuário “standard\_angelo”
  - E a senha “secret\_sauce”
  - Então deve ser exibida a mensagem “Epic sadface: Username and password do not match any user in this service” .

# /CENÁRIOS

- Cenário 3 – Senha Inválida
  - Dado que eu acesse a página de autenticação do saucedemo
  - Quando eu digitar o usuário “standard\_user”
  - E a senha “secret\_noway”
  - Então deve ser exibida a mensagem “Epic sadface: Username and password do not match any user in this service” .



# /CENÁRIOS

- Cenário 4 – Dados e login em branco
  - Dado que eu acesse a página de autenticação do saucedemo
  - Quando eu digitar o usuário “”
  - E a senha “”
  - Então deve ser exibida a mensagem “Epic sadface: Username is required”.

# /CENÁRIOS USANDO CONTEXTO

- Contexto: Dado que eu acesse a página de autenticação do saucedemo
- Cenário: 1 - Autenticação válida
  - Quando eu digitar o usuário “standard\_user”
  - E a senha “secret\_sauce”
  - Então deve ser exibida a logo do “Swag Labs”.

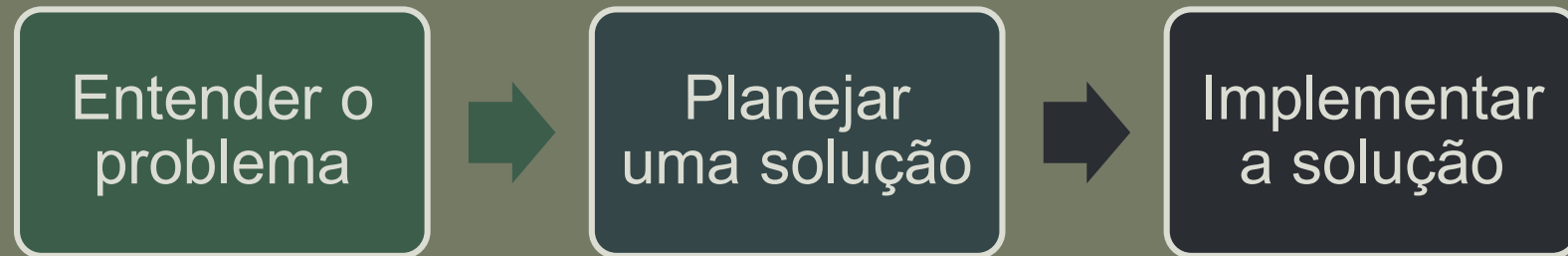
# /AUTOMAÇÃO DE TESTES COM BDD

- #language: pt
- Funcionalidade: Login no Swag Labs
- Contexto: Dado que eu acesse a página de autenticação do saucedemo
- Esquema do Cenário: Logins inválidos com tabela
- Quando eu digitar o <usuário>
- E a senha <senha>
- Então deve ser exibida a mensagem <mensagem>

- Exemplos:

●	usuario	senha	mensagem	
●	"user"	"secret_sauce"	"m1"	
●	"standard_user"	"12345"	"m2"	
●	""	""	"m3"	

# /APLICABILIDADE



# /DESAFIO – AUTOMAÇÃO DE TESTES COM BDD

- #language: pt
- Funcionalidade: Criar conta - Casas Bahia
- Como um cliente
- Quero criar conta no site das Casas Bahia
- Para poder fazer compras online
  
- Cenário 1: Cadastro com campos obrigatórios inválidos
- Cenário 2: Cadastro com campos obrigatórios válidos
- Cenário 3: Cadastro com CPF duplicado
- Cenário 4: Cadastro com e-mails diferentes

# /REFERÊNCIAS

- <https://www.slideshare.net/elias.nogueira/papel-do-qa-na-transformao-gil>
- <https://www.slideshare.net/elias.nogueira/papel-do-qa-na-transformao-gil>
- <https://cucumber.io/>



# AULA PRÁTICA



# /AUTOMAÇÃO DE TESTES COM CYPRESS

Aula prática, com pouco material teórico



# /POR QUE CYPRESS?

- Fácil instalação;
- Fácil configuração;
- Baixa curva de aprendizado;
- Documentação completa;
- Suporte a multi-browsers;
- Dashboard amigável;
- Fácil Depuração (debug);
- Tipos de testes: Unidade, Componentes, Integração & Teste de Interface Do usuário (UI)



# /PREPARAÇÃO DE AMBIENTE

- Instalar Node.JS
- Verificar a instalação:
- `node --version`
- Instalar Visual Studio Code
- Plugins recomendados:
  - ES6 Mocha Snippets
  - Cypress Snippets
  - Visual Studio IntelliCode



# BANG!

Iniciar PROJETO!

# /INICIAR PROJETO

- Criar o projeto node (package.json):
  - `npm init -y`
- Instalar o Cypress (na pasta do projeto)
  - `npm install cypress -D`
- Suporte em: <https://docs.cypress.io/guides/getting-started/installing-cypres>

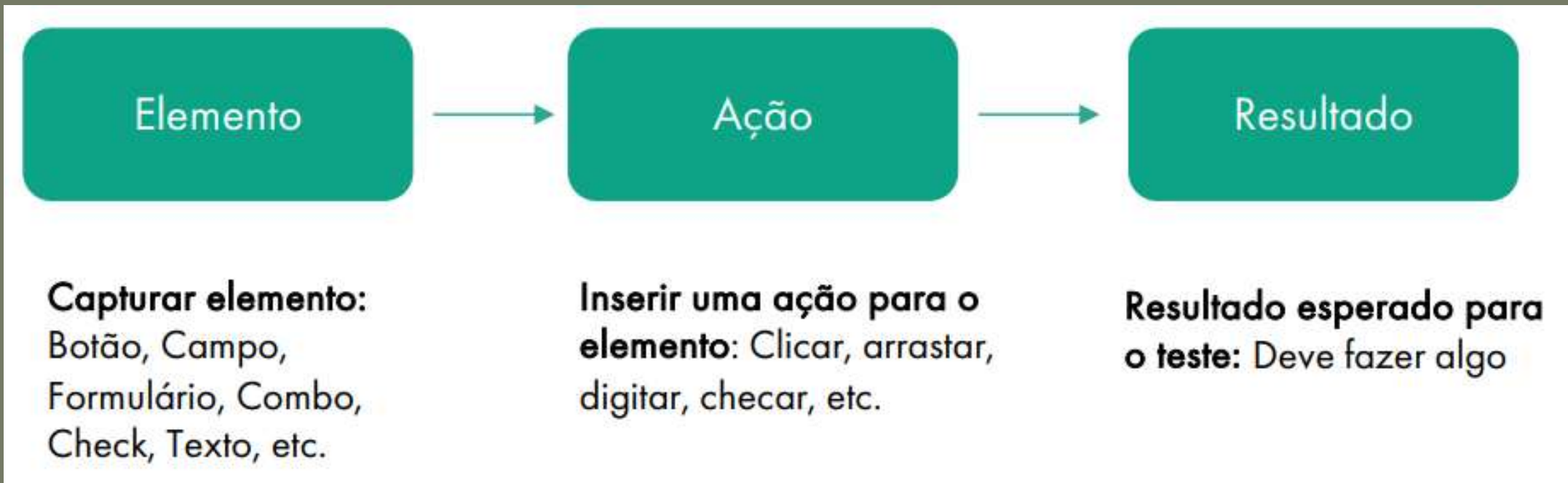
# /INICIAR PROJETO

- Após a instalação execute:
  - `npx cypress open`

# /INICIAR PROJETO

- Scaffolding também utilizado em programação para indicar que o código a que se refere é apenas **um esqueleto usado para tornar a aplicação funcional**, e se espera que seja substituído por algoritmos mais complexos à medida que o desenvolvimento da aplicação progride.

# /ESTRUTURA DO TESTE AUTOMATIZADO





# BANG!

Primeira automação!





# PRATIQUE



# /Hora de praticar

- Cenário 2 – Usuário Inexistente
- Cenário 3 – Senha Inválida
- Cenário 4 – Dados e login em branco
- Extra: <https://www.npmjs.com/package/@faker-js/faker>
  - <https://demo.automationtesting.in/Register.html>

# /HOOKS

before

Teste 1

Teste 2

Teste 3

Teste N

beforeEach

Teste 1

beforeEach

Teste 2

beforeEach

Teste 3

beforeEach

Teste N

Teste 1

Teste 2

Teste 3

Teste N

after

Teste 1

afterEach

Teste 2

afterEach

Teste 3

afterEach

Teste N

afterEach

# /HOOKS

```
before(() => {  
  //trecho de código aqui  
});
```

Exemplo: Carregar uma base de dados para os testes

```
beforeEach(() => {  
  //trecho de código aqui  
});
```

Exemplo: Visitar a mesma URL antes de cada teste

```
after(() => {  
  //trecho de código aqui  
});
```

Exemplo: Limpar uma base de dados depois de executar os testes

```
afterEach(() => {  
  //trecho de código aqui  
});
```

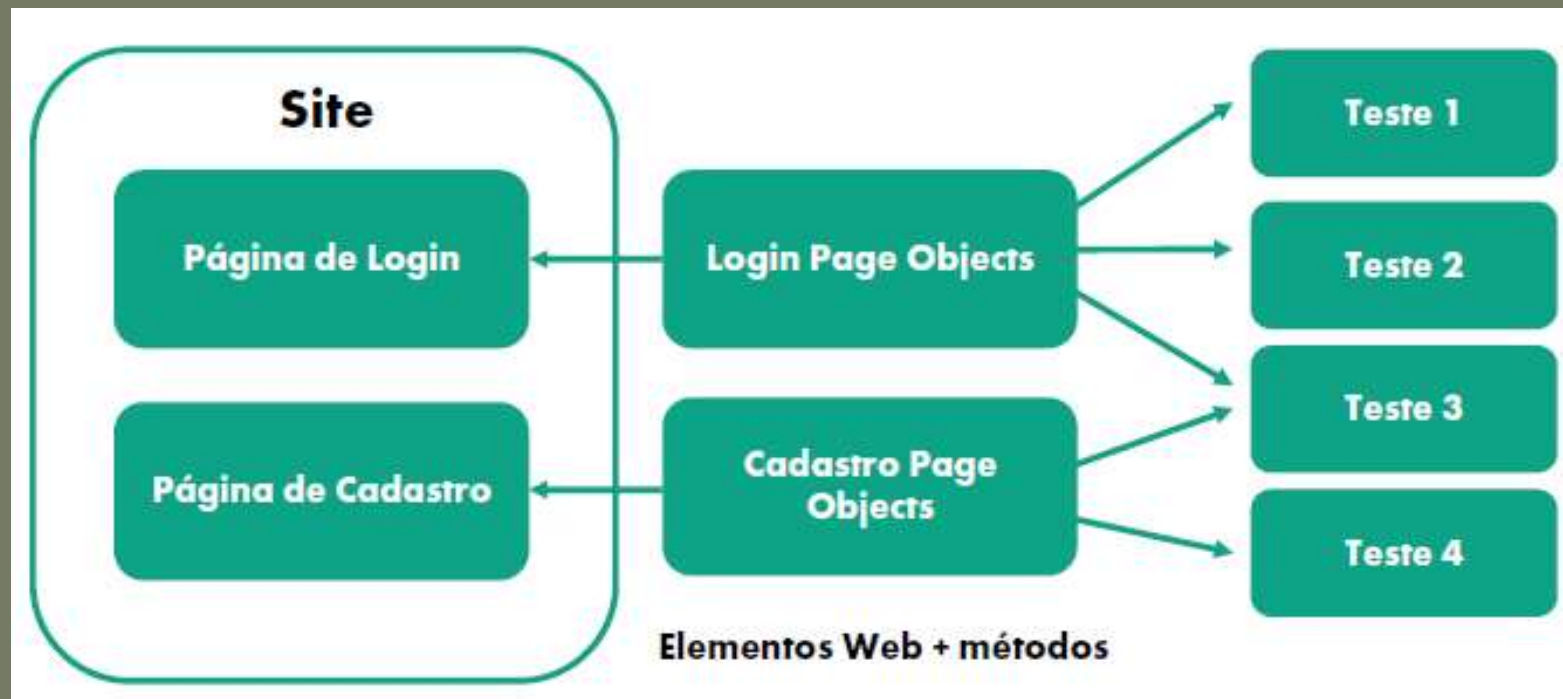
Exemplo: Limpar algum dado comum depois de cada teste. Não é comum o uso, porque o Cypress já limpa os dados depois de cada teste.



**EXTRA**



# /PAGE OBJECTS MODEL



# /REFERÊNCIAS

- <https://docs.cypress.io/guides/getting-started/installing-cypress>
- <https://docs.cypress.io/>
- <https://www.youtube.com/agilizei>

# /AUTOMAÇÃO COM BDD

Behavior Driven Development  
Com menos material teórico ainda



# /REFERÊNCIAS

- Introducing BDD - <http://dannorth.net/introducing-bdd/>
- CucumberJS - <https://cucumber.io/docs/installation/javascript/>
- Cypress Cucumber Preprocessor - <https://www.npmjs.com/package/cypress-cucumber-preprocessor>



# THANK YOU



Você tem alguma dúvida?  
prof2132@iesp.edu.br  
+55 81 98808 1008  
angelofdiasg.tech (em dev)



@oanjoeducador



[https://www.linkedin.com  
/in/angelofdiasg/](https://www.linkedin.com/in/angelofdiasg/)

CREDITS: This presentation template was created by **Slidesgo**, including icons  
by **Flaticon**, infographics & images by **Freepik** and illustrations by **Stories**

Please keep this slide for attribution



# SPECIAL AWARDS



## PRÁTICAS ÁGEIS

Spotify, Pirâmide de testes, BDD  
(Teoria)



## BDD

Conceitos, Linguagem Gherkin  
(Teoria e Prática)



## BDD - FERRAMENTAS

Cucumber (Com Cypress), VSCode (Plugins)  
(Prática)



## CYPRESS

Automação de testes com Cypress  
(Prática)



## BDD - CUCUMBER

Automação de testes com Cypress e Cucumber  
(Prática)



## PLATINA

O mercado está a sua espera!



# ARMY RANKS



CORONEL					
TENENTE					
CABO					
SOLDADO					





“Temos o mundo aqui e é a nossa vez.  
Temos que fazer algo de bom com ele.  
E Se você precisar de mim, estarei lá.

—STEVE ROGERS



# THANK YOU



Você tem alguma dúvida?  
prof2132@iesp.edu.br  
+55 81 98808 1008  
angelofdiasg.tech (em dev)



@oanjoeducador



[https://www.linkedin.com  
/in/angelofdiasg/](https://www.linkedin.com/in/angelofdiasg/)

CREDITS: This presentation template was created by **Slidesgo**, including icons  
by **Flaticon**, infographics & images by **Freepik** and illustrations by **Stories**

Please keep this slide for attribution