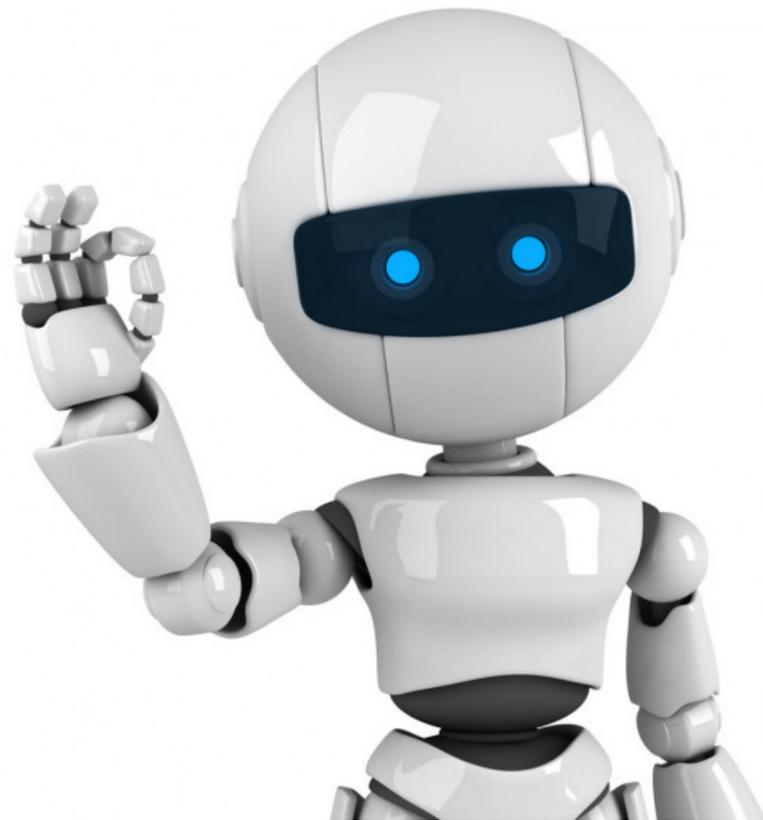




Flutter



Automação
de testes
para
aplicativos
em **Flutter**

Agenda

1. Apresentação
2. Entendendo a comunidade e desafios
3. Testes unitários
4. Testes de widget
5. Testes de UI
6. CODE

APRESENTAÇÃO



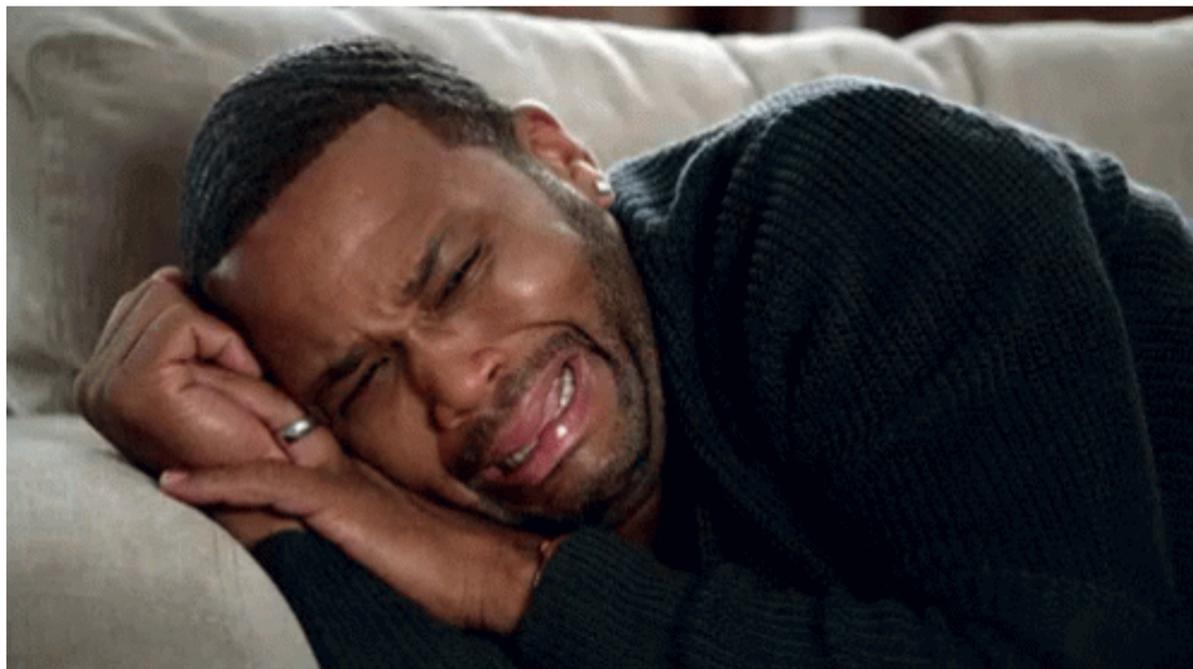
Jhonatas Santos Matos

- ⇒ QA Tech Leader - GoK. Digital
- ⇒ Formado em ADS
- ⇒ Pós Graduado em Engenharia de Software
- ⇒ Apaixonado por tecnologia e inovação
- ⇒ Entusiasta no desenvolvimento mobile
- ⇒ Amante de jogos online

Entendendo a comunidade e desafios na automação

"Testar ou não testar eis a questão"

Como assim ? Não tem testes ?

A screenshot of a live stream chat interface. The top part shows a video call grid with five participants. The bottom part shows a chat log with several messages. A red box highlights a message from Jhonatas Matos: "Como houve uma migração ai de tecnologia , provavelmente houveram muitos testes ...como voces estao lidando com os testes ? quais frameworks estao utilizando para unitarios, integrados ?". The chat log also includes messages from other users like NERFAL, Charles dos Santos, Rodrigo Branas, Dyago Matheus, Atila Augusto Dos Santos, Cristiano Lopes, Leo Moraes, and Matheus Costa.

Powered by StreamYard

Repetição das principais mensagens do chat

NERFAL JS e tao ruim que o node ta virando deno kkkk

Charles dos Santos @Dyago Matheus Odeio js

Jhonatas Matos Como houve uma migração ai de tecnologia , provavelmente houveram muitos testes ...como voces estao lidando com os testes ? quais frameworks estao utilizando para unitarios, integrados ?

Rodrigo Branas não tá fácil pra ninguém

Rodrigo Branas hauhuahau

Dyago Matheus Kkkkkk

Atila Augusto Dos Santos kkkkkkk

Rodrigo Branas nem o coitado do node se salvou

Cristiano Lopes kkkk

Leo Moraes O Flutter tem também testes, correto?

Rodrigo Branas alias, vai ter live de deno domingo eim (spoiler)

Dyago Matheus node virando deno e js virando TS

Matheus Costa todo mundo falando desse deno anora

Jhonatas Matos
ACREDITO SER O UNICO QA ASSISTINDO AQUI E VOCES NAO ESTAO FAZENDO TESTES

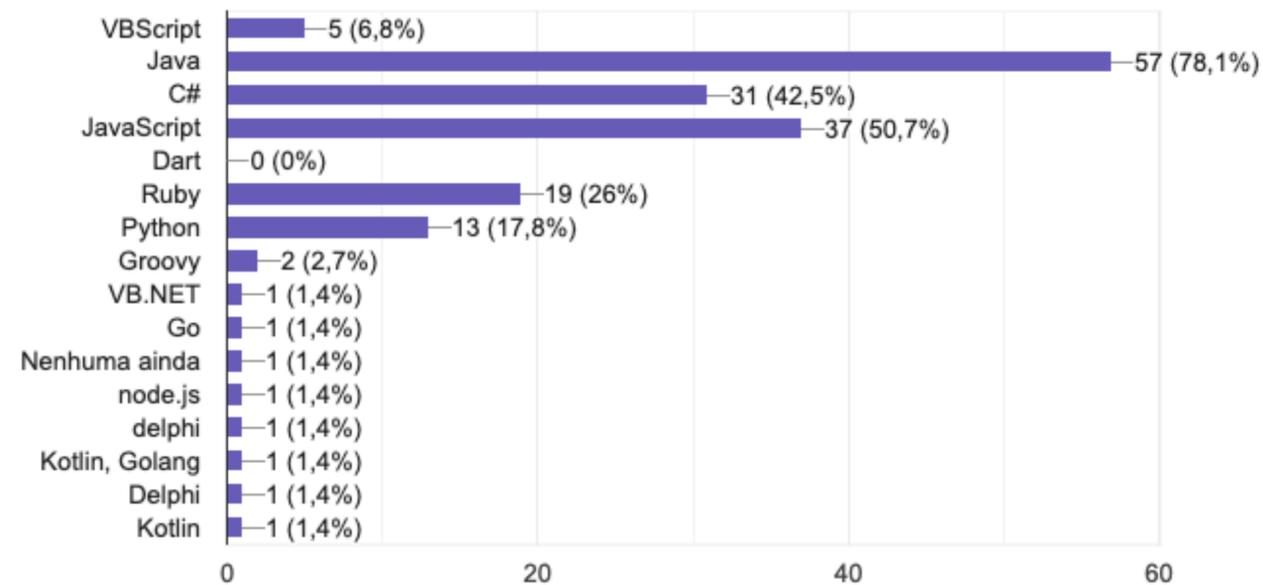
Também usamos o Flutter. (feat Pravalder)

Entendendo a comunidade e desafios na automação

Comunidade de testes

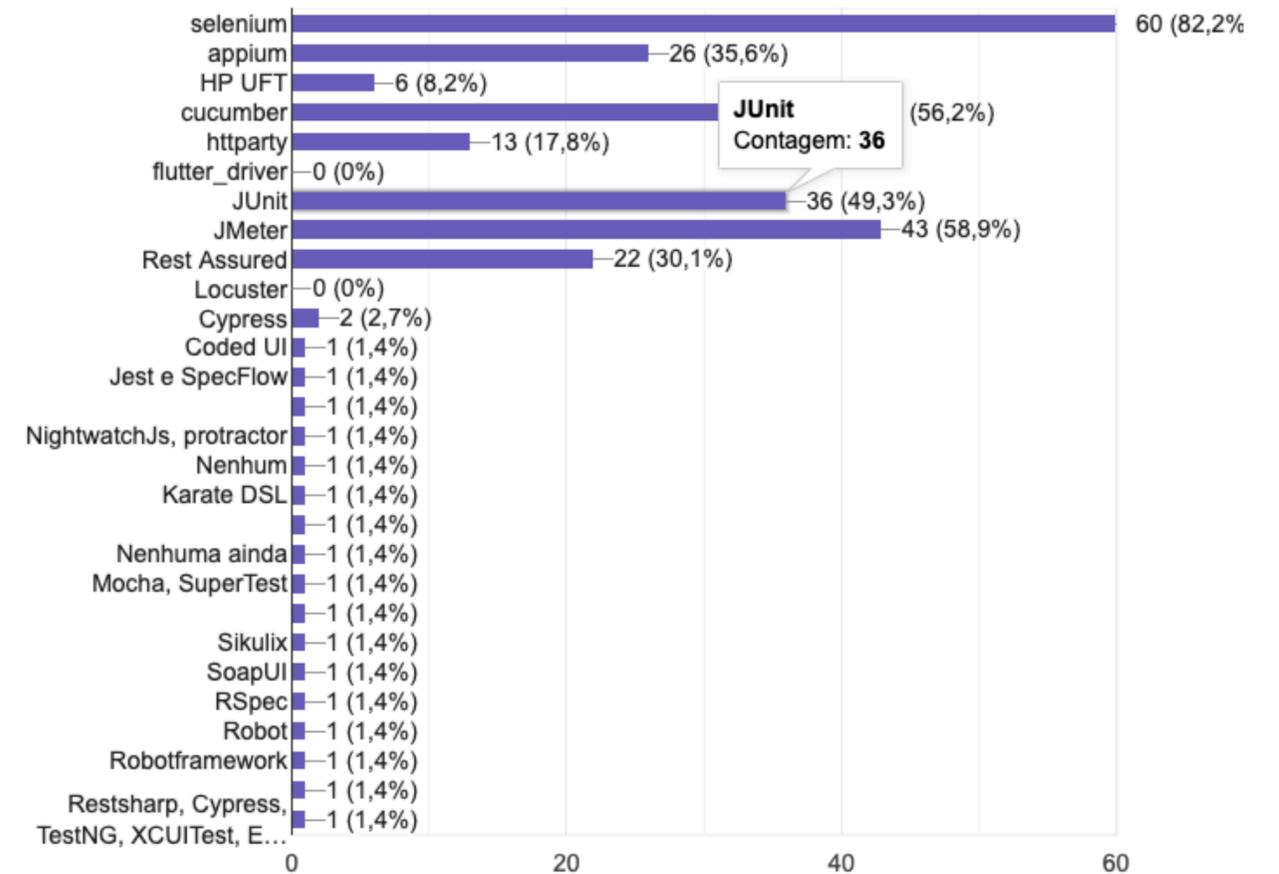
Quais linguagens de programação já trabalhou ?

73 respostas



Quais frameworks/ferramentas já trabalhou ?

73 respostas



Entendendo a comunidade e desafios na automação

Mapear elementos

Parar executar a automação funcional, é necessário mapear elementos para que o “robô” da automação possa executar o fluxo como se fosse o usuário utilizando o aplicativo.

```
44 child: RaisedButton(  
45   key: init_button, Jhonatas Santos, 2 months ago  
46   onPressed: () => _initApp(context),  
47   child: Text(  
48     'Iniciar Quiz',  
49     style: TextStyle(color: Colors.white, fontSize: 16),  
50   ), // Text  
51   color: Colors.green,  
52   elevation: 4,
```

The screenshot shows the UI Automator Viewer tool displaying the GO-K app interface. The app screen features the GO-K logo, the text "Seu quiz de qualidade", and a green button labeled "Iniciar Quiz". The tool's interface includes a toolbar with icons for zooming and a tree view of the UI hierarchy. The tree view shows the following structure:

- (0) FrameLayout [0,0][1080,1794]
 - (0) LinearLayout [0,0][1080,1794]
 - (0) FrameLayout [0,0][1080,1794]
 - (0) FrameLayout [0,0][1080,1794]
 - (0) View [0,0][1080,1794]
 - (0) View [0,0][1080,1794]
 - (0) View [0,0][1080,1794]
 - (0) ImageView [0,0][1080,1794]
 - (1) View:Seu quiz
 - (2) Button:Iniciar Quiz

The Node Detail panel on the right provides the following information for the selected button:

Node Detail	
index	2
text	Iniciar Quiz
resource-id	
class	android.widget.Button
package	br.com.goQa
content-desc	
checkable	false
checked	false
clickable	true
enabled	true
focusable	true
focused	false
scrollable	false

Entendendo a comunidade e desafios na automação

Comunidade Flutterando

██████ Oct '19

Olá pessoal, bom dia, tudo bom?

Gostaria de saber como faz para colocar ID nos elementos utilizando o flutter.

Gostaria de preencher a opção resource_id do android e o Accessibility_id do ios.

██████ Nov '19

Flutter não é nativo! Flutter é apenas um canvas onde todos os componentes são desenhados com uma biblioteca 2D chamada Skia.

Qualquer interação que você queira com componentes nativos simplesmente não existe (por isso também não é possível usar Roboscript em Flutter).

 1  Reply

Log In to Reply

⬆ Original Post

2 of 2 posts
November 2019

⬇ Now

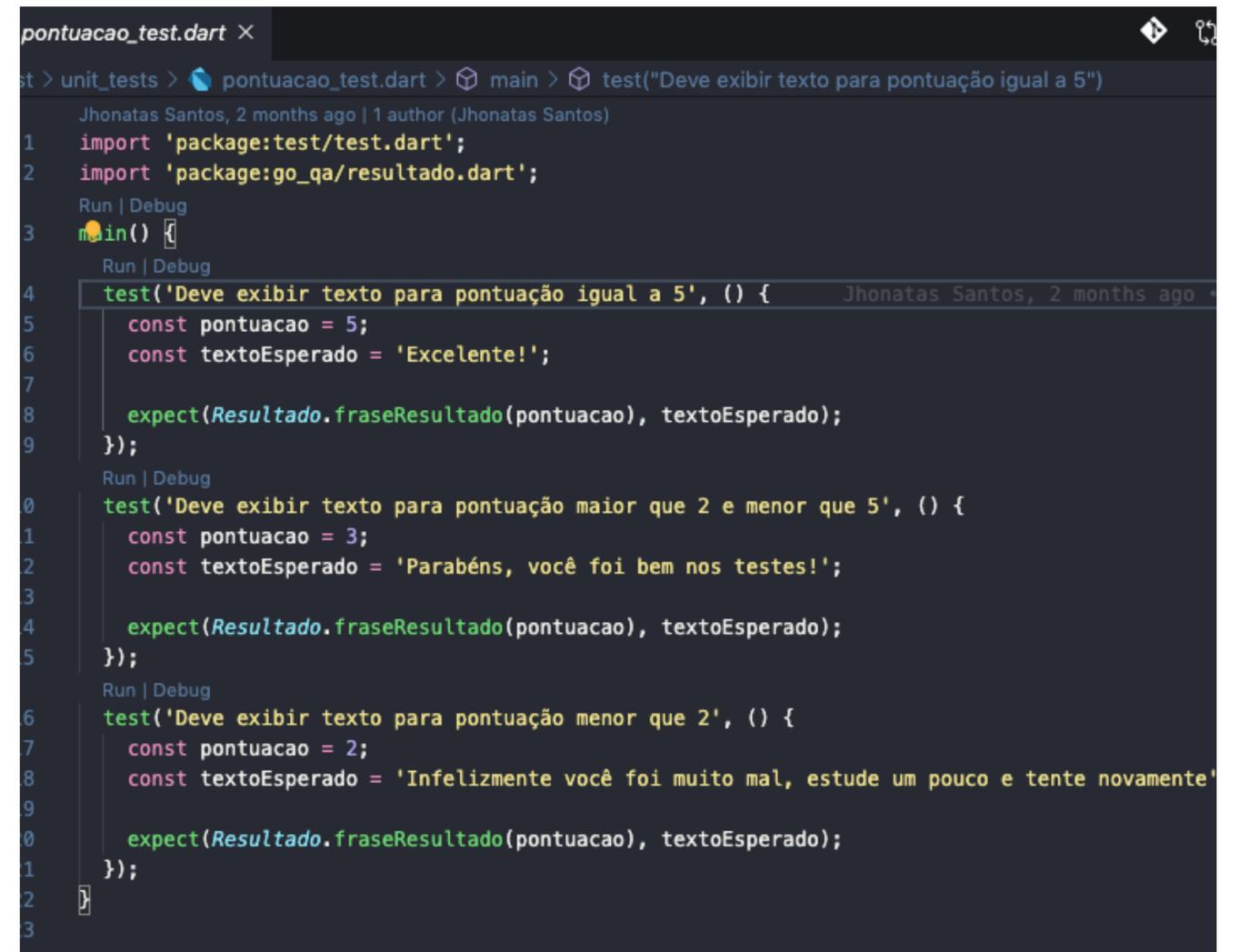
Testes unitários

Para testes unitários precisamos instalar apenas uma dependência.

```
dev_dependencies:  
  
test: ^1.6.1
```

Se você estiver utilizando o editor de texto vscode, ele “automagicamente” vai baixar e instalar o pacote, se não, basta você ir pelo terminal até a pasta onde se encontra o arquivo de dependências `pubspec.yaml`, e executar o comando:

```
pub get
```



```
pontuacao_test.dart ×  
st > unit_tests > pontuacao_test.dart > main > test("Debe exhibir texto para pontuação igual a 5")  
Jhonatas Santos, 2 months ago | 1 author (Jhonatas Santos)  
1 import 'package:test/test.dart';  
2 import 'package:go_qa/resultado.dart';  
Run | Debug  
3 main() {  
Run | Debug  
4 test('Debe exhibir texto para pontuação igual a 5', () {  
5   const pontuacao = 5;  
6   const textoEsperado = 'Excelente!';  
7  
8   expect(Resultado.fraseResultado(pontuacao), textoEsperado);  
9 });  
Run | Debug  
0 test('Debe exhibir texto para pontuação maior que 2 e menor que 5', () {  
1   const pontuacao = 3;  
2   const textoEsperado = 'Parabéns, você foi bem nos testes!';  
3  
4   expect(Resultado.fraseResultado(pontuacao), textoEsperado);  
5 });  
Run | Debug  
6 test('Debe exhibir texto para pontuação menor que 2', () {  
7   const pontuacao = 2;  
8   const textoEsperado = 'Infelizmente você foi muito mal, estude um pouco e tente novamente';  
9  
0   expect(Resultado.fraseResultado(pontuacao), textoEsperado);  
1 });  
2  
3
```

Testes de widget

Testes de widget, tem como objetivo verificar se ele agirá da forma esperada.

Além disso, o widget testado poderá receber e reagir a eventos do usuário, conseguirá instanciar widgets filhos e simulará a construção da UI.

Precisamos importar apenas o `flutter_test` como dependência de desenvolvimento.

Run | Debug

```
10 testWidgets('Iniciar quiz, navegar para a tela de identificação',
11   (WidgetTester tester) async {
12     NavigatorObserver mockObserver;
13     mockObserver = MockNavigatorObserver();
14
15     //Criando a tela Home com o mock
16     await tester.pumpWidget(
17       MaterialApp(home: Home(), navigatorObservers: [mockObserver]));
18
19     //Executando a ação de tap no botao iniciar quiz
20     await tester.tap(find.byKey(Home.init_button));
21     await tester.pumpAndSettle();
22
23     //Ao tocar no botão navegamos na tela Identificação
24     //O método didPush () deveria ter sido chamado ...
25     verify(mockObserver.didPush(any, any));
26
27     //Deve haver uma Identificacao presente na árvore de widgets ...
28     expect(find.byType(Identificacao), findsOneWidget);
29
30     //Com o botão que tem a Key start_button
31     // expect(find.byKey(Key('start_button')), findsOneWidget);
32     expect(find.byKey(Key('start_button')), findsOneWidget);
33   });
34
```

Testes de widget

WidgetTester: Responsável por toda a “simulação” do seu *widget*, construção, gestos, ***pumpWidget()***: Cria o *widget* e fala para o *WidgetTester* construí-lo.

pump(): Reconstrói o *widget* depois de um determinado período de tempo.

pumpAndSettle(): Reconstrói o *widget* até o último frame/estado, essencial quando se está

Finder: Como o nome entrega, ele é responsável por achar os *widgets*.

```
Run | Debug
10 testWidgets('Iniciar quiz, navegar para a tela de identificação',
11   (WidgetTester tester) async {
12     NavigatorObserver mockObserver;
13     mockObserver = MockNavigatorObserver();
14
15     //Criando a tela Home com o mock
16     await tester.pumpWidget(
17       MaterialApp(home: Home(), navigatorObservers: [mockObserver]));
18
19     //Executando a ação de tap no botao iniciar quiz
20     await tester.tap(find.byKey(Home.init_button));
21     await tester.pumpAndSettle();
22
23     //Ao tocar no botão navegamos na tela Identificação
24     //O método didPush () deveria ter sido chamado ...
25     verify(mockObserver.didPush(any, any));
26
27     //Deve haver uma Identificacao presente na árvore de widgets ...
28     expect(find.byType(Identificacao), findsOneWidget);
29
30     //Com o botão que tem a Key start_button
31     // expect(find.byKey(Key('start_button')), findsOneWidget);
32     expect(find.byKey(Key('start_button')), findsOneWidget);
33   });
34
```

Testes de UI

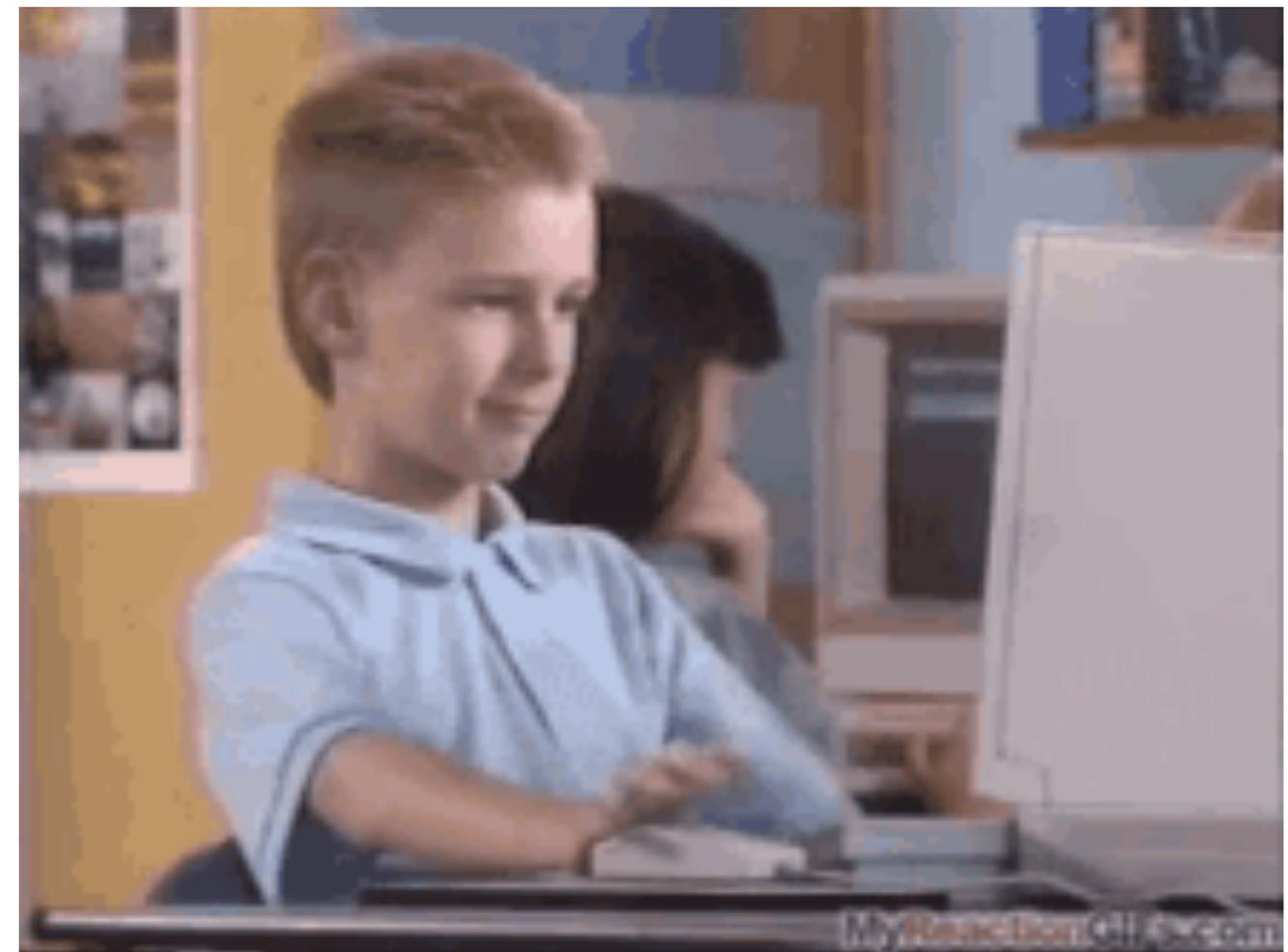
Precisamos adicionar a dependência `flutter_driver` que irá nos auxiliar nesse processo.

```
33 flutter_driver:  
34   sdk: flutter
```

Podemos adicionar mais uma dependência ao nosso projeto, `flutter_gherkin`.

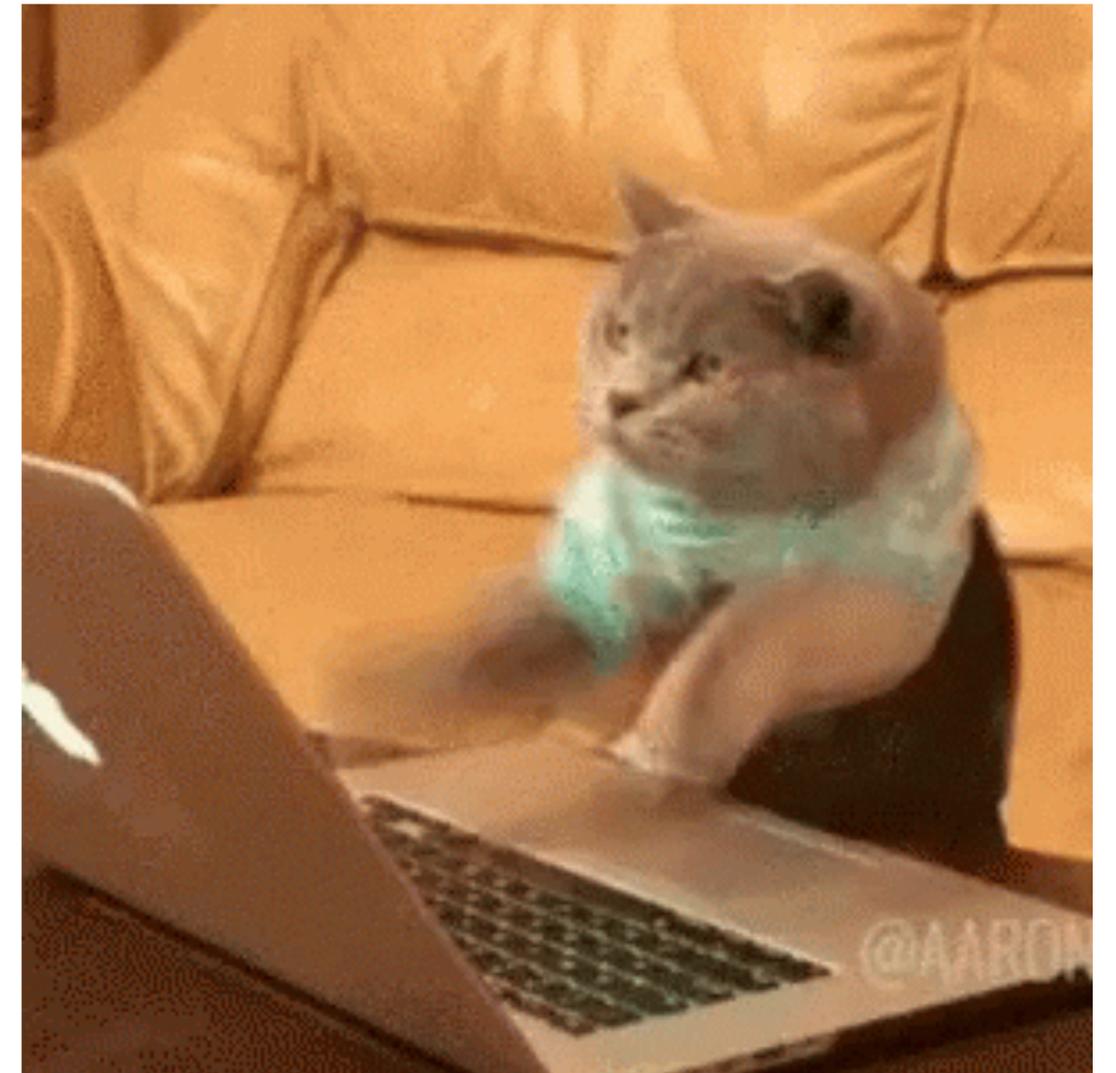
```
22 flutter_gherkin: ^1.1.8+2  
23
```

Após a instalação dessa dependência, podemos escrever nossos testes de forma ágil e trabalhar com documentação utilizando gherkin.



CODE

BORA CODAR !!!!



OBRIGADO

Jhonatas Santos Matos

 <https://www.linkedin.com/in/jhonatas-matos/>

 Bug in prod

 <https://github.com/Bug-In-Prod>

 <https://www.instagram.com/buginprod/>

VALEEEEU !!!!

