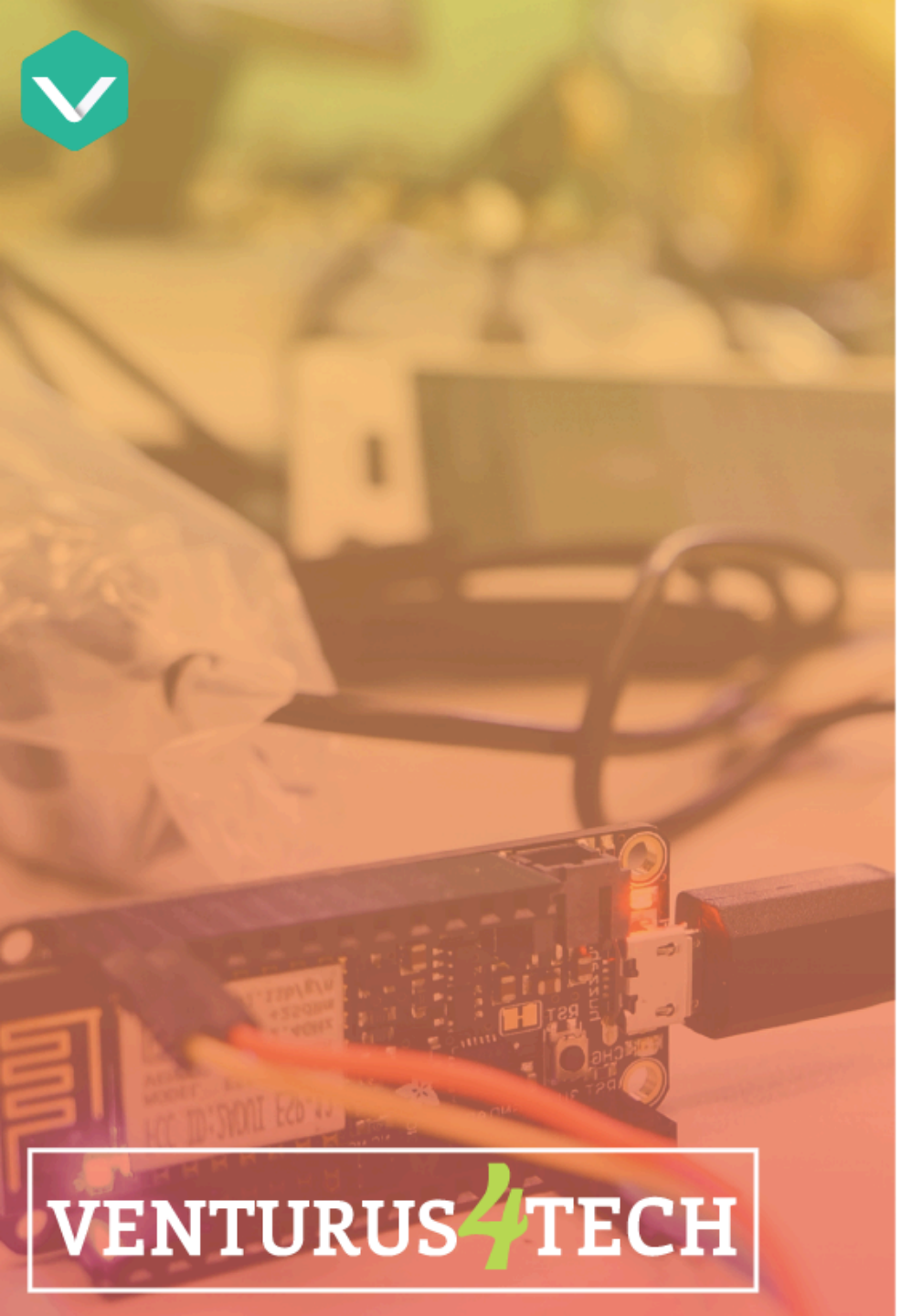

VENTURUS4TECH
UMA SALA DE AULA DIFERENTE

Teste de Software

Junior Toshiharu Saito
Marcio Cunha

VENTURUS4TECH

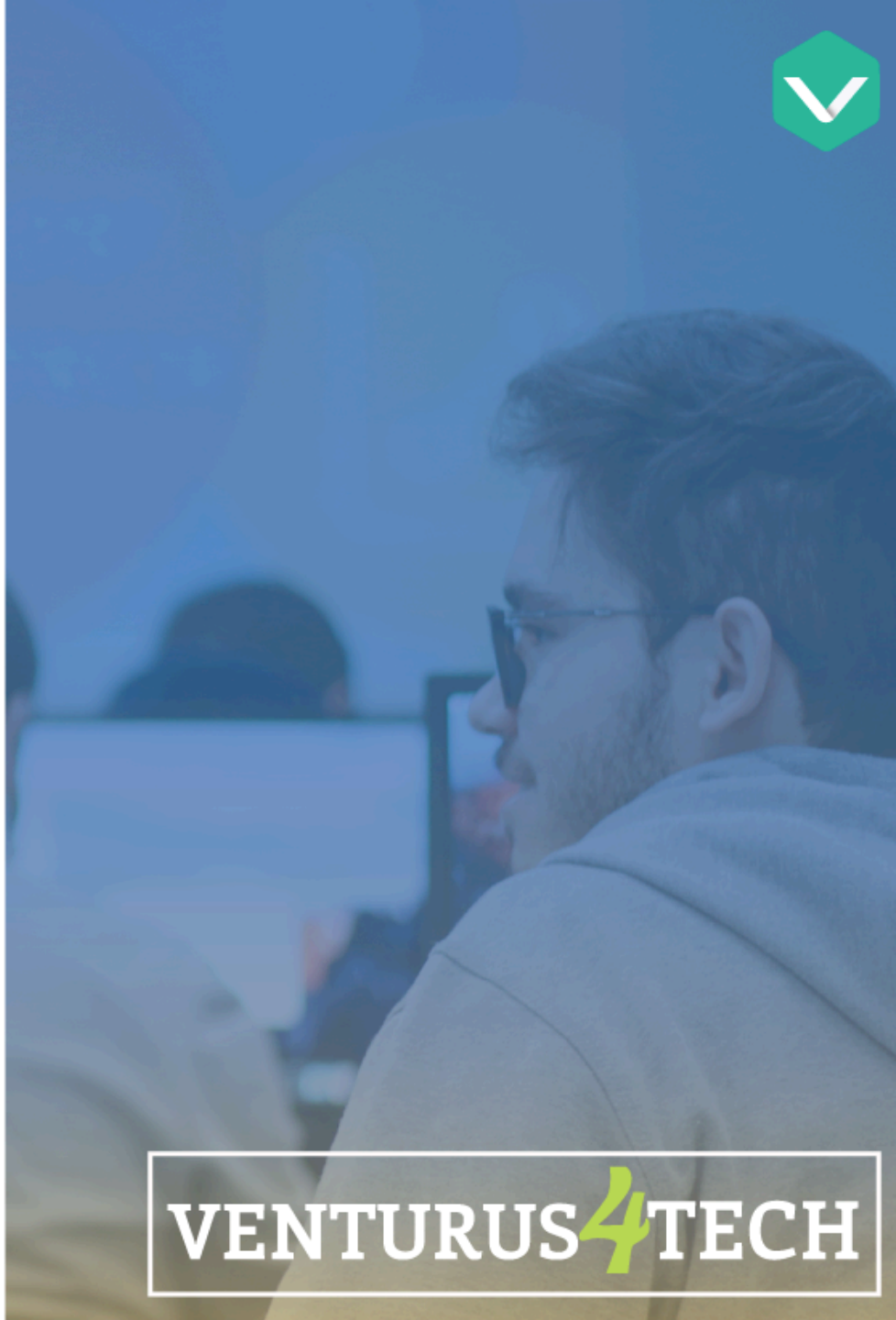


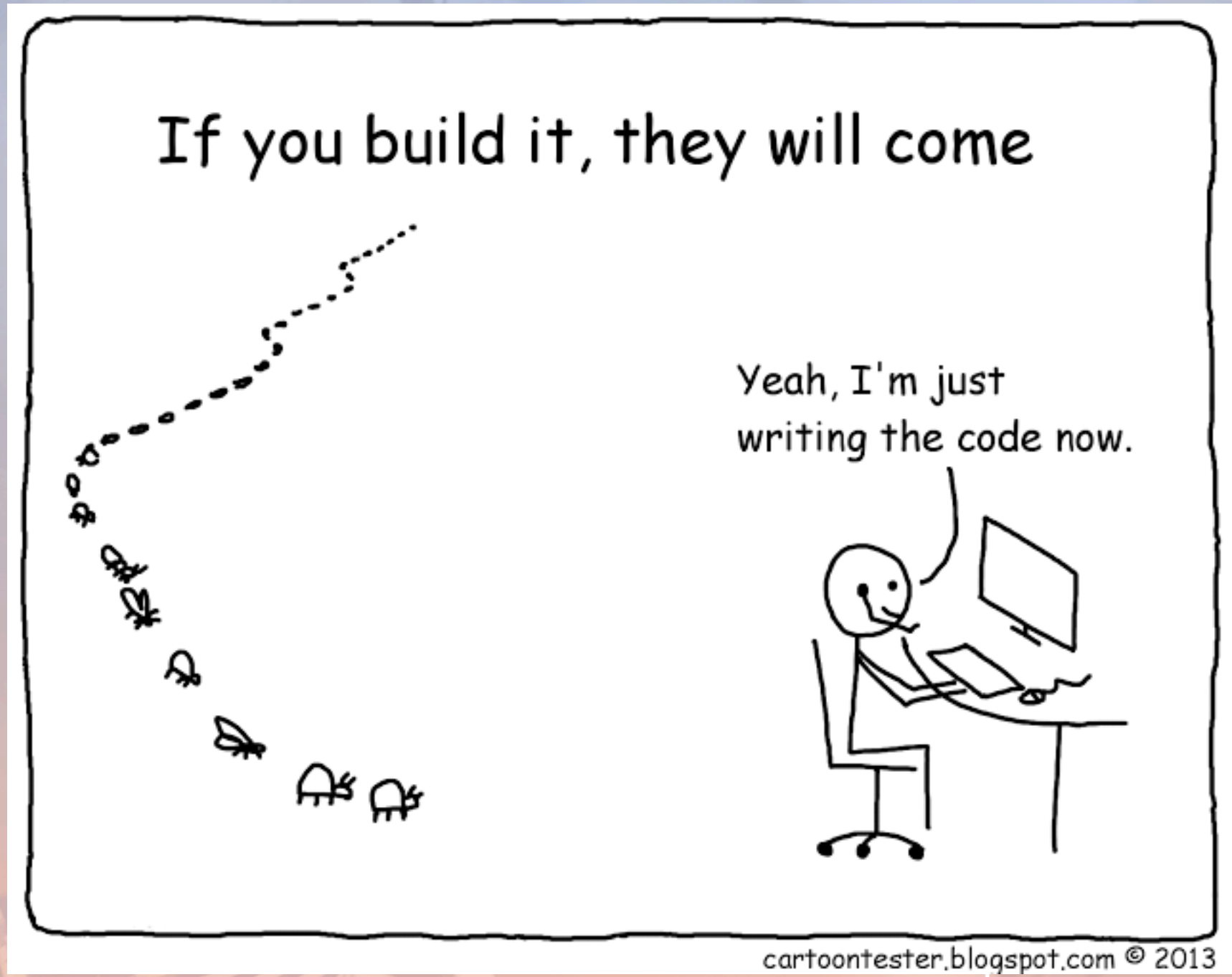
Junior Toshiharu Saito

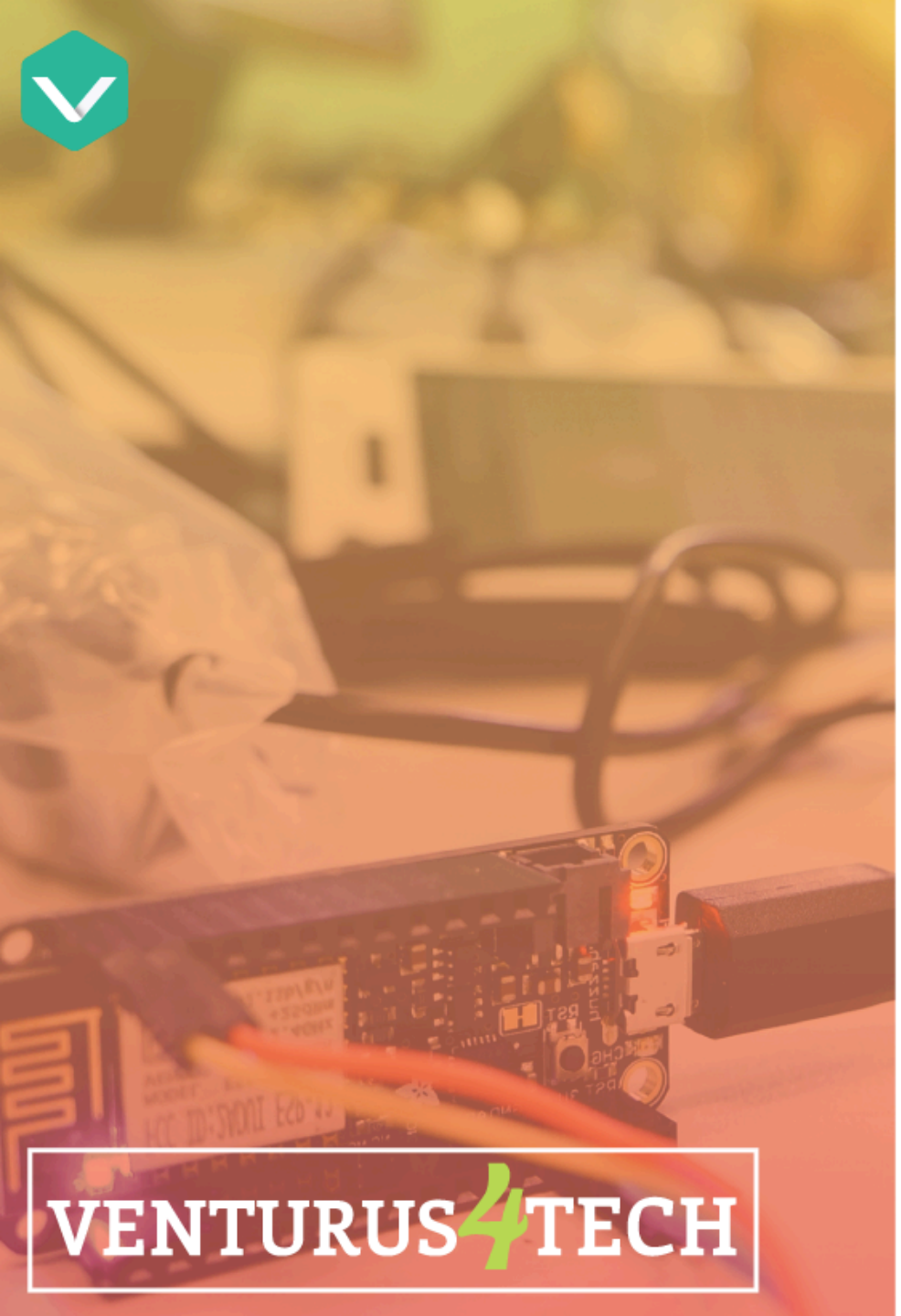
- Bacharel em Ciência da Computação pela UEM
- Mestre em Ciência da Computação pelo Instituto de Computação, UNICAMP
- 14 anos de experiência com Teste de Software
- Desenvolvimento e execução de casos de testes para aplicações mobile e web para clientes nacionais e internacionais
- Automação de testes para aplicações web com ferramentas como Selenium, JUnit, TestNG e Sikuli

Márcio Cunha

- Bacharel em Matemática pela FSA
- MBA em Gestão de TI
- Mais de 14 anos de experiência com Teste de Software em empresas como Ericsson e IBM
- Desenvolvimento e execução de testes para clientes nacionais e internacionais
- Automação de testes para aplicações web




If you build it, they will come
Yeah, I'm just writing the code now.
cartoontester.blogspot.com © 2013



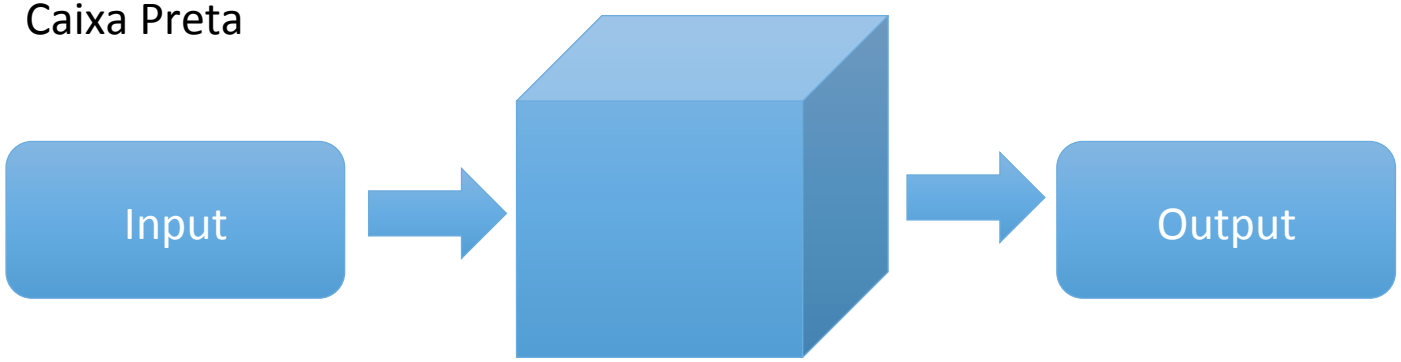
Definições

É o processo de análise de um item de software para detectar diferenças existentes entre os requisitos e as funcionalidades. (IEEE)

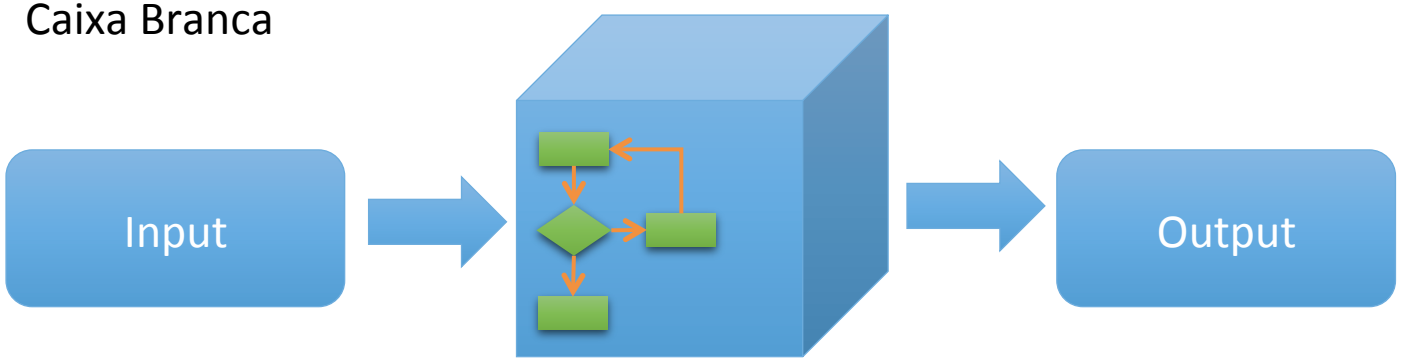
Investigação conduzida para fornecer informação sobre a qualidade do produto ou serviço em teste. (wikipedia)

Técnicas

Caixa Preta

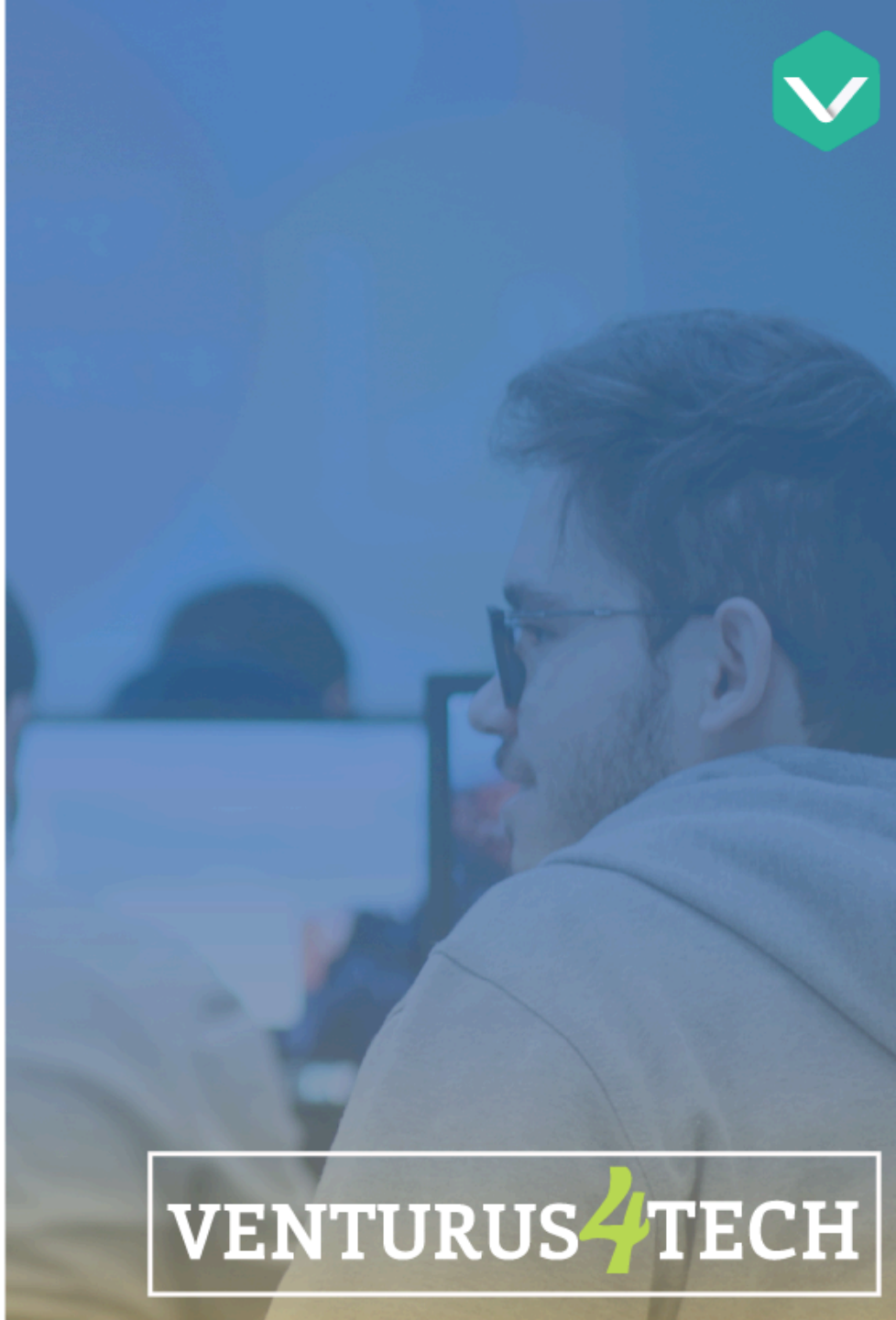


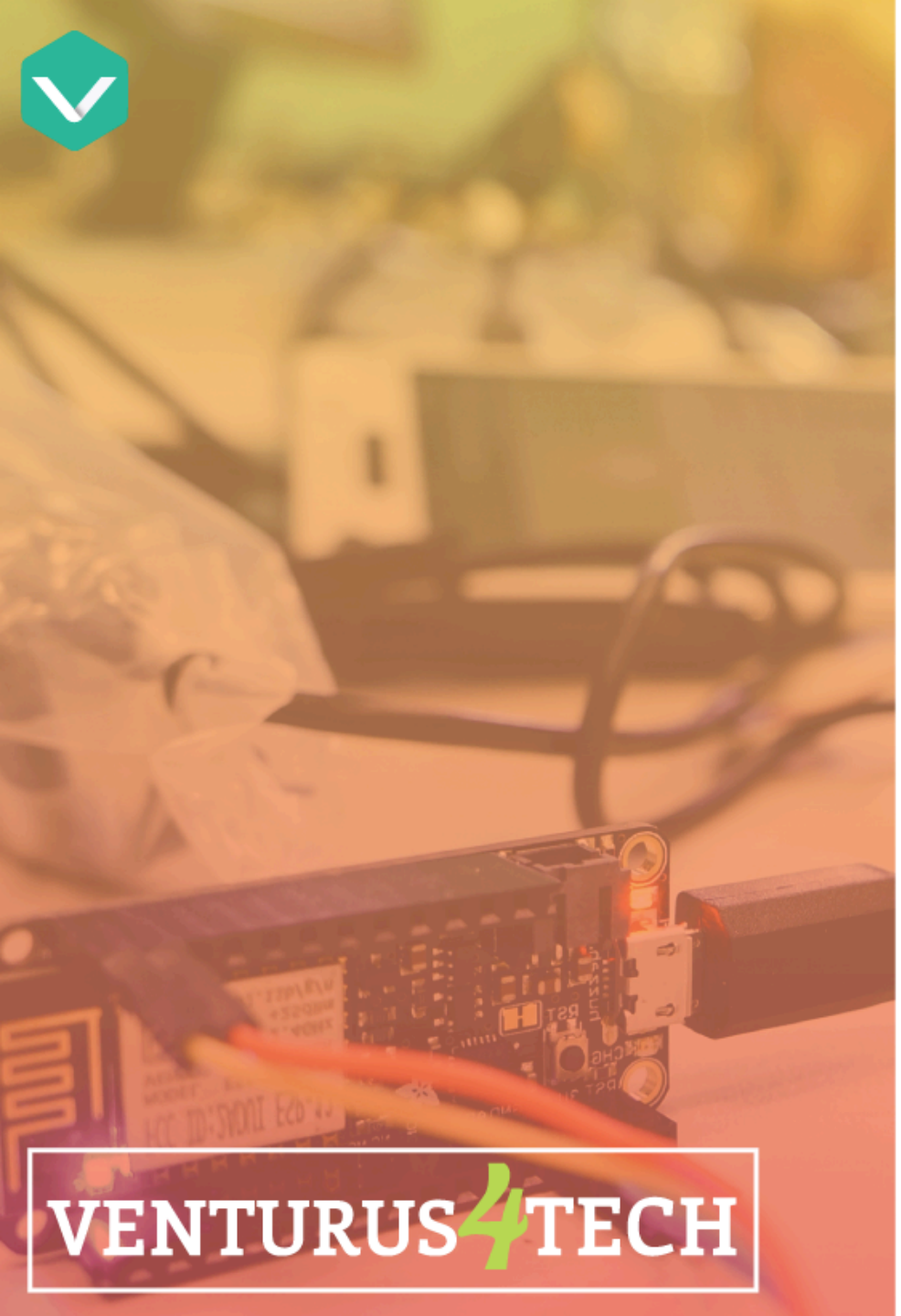
Caixa Branca



Categorias

- **Unitário**: testam a menor unidade de código possível
- **Integração**: testam o comportamento das unidades de código
- "***End-to-End***"(Aceitação): testam o sistema e verificam se estão de acordo com os requisitos





Caso de Teste

Documentar os cenários a serem testados

Geralmente possui os seguintes campos:

- Objetivo: o que esse caso de teste está validando
- Pré-condições: quais as condições que devem se encontrar antes de iniciar o teste
- Passos e Resultado: qual o procedimento a ser executado e seu respectivo resultado

Caso de Teste

- Exemplo de Caso de Teste
  - Cenário: O usuário ao selecionar “Esvaziar a Lixeira”, a notificação de confirmação deve ser mostrada.

Test Case	Summary	Steps	Expected Results
Teste-1: Lixeira	Pasta Lixeira	1. Esvaziar Lixeira	1. Veja a Notificação

VENTURUS4TECH

Caso de Teste

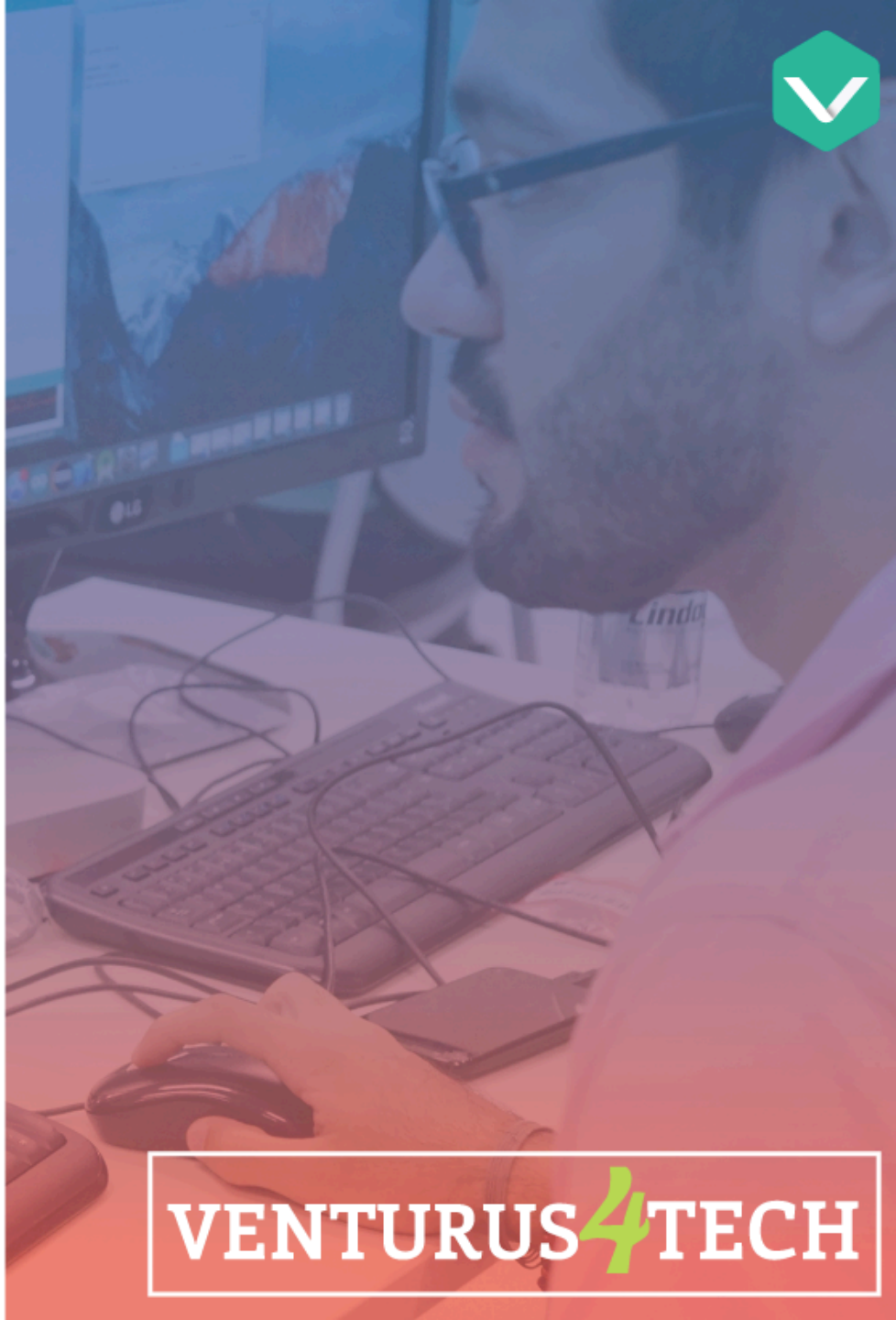
Test Case	Summary	Steps	Expected Results
Teste-1: Lixeira: Esvaziar a pasta	Verifique que o diálogo de confirmação é mostrado quando o usuário seleciona a opção de esvaziar a lixeira Pre-condições - A pasta lixeira possui um arquivo - O ícone Lixeira está visível na área de trabalho	1. Abra o menu de opções da pasta Lixeira 2. Selecione a opção “Esvaziar Lixeira”	1. O menu de opções é mostrado. As opções são Abrir e Esvaziar Lixeira 2. O diálogo de confirmação é mostrado. O texto “Deseja apagar os arquivos permanentemente?” As opções são Cancelar e Esvaziar Lixeira

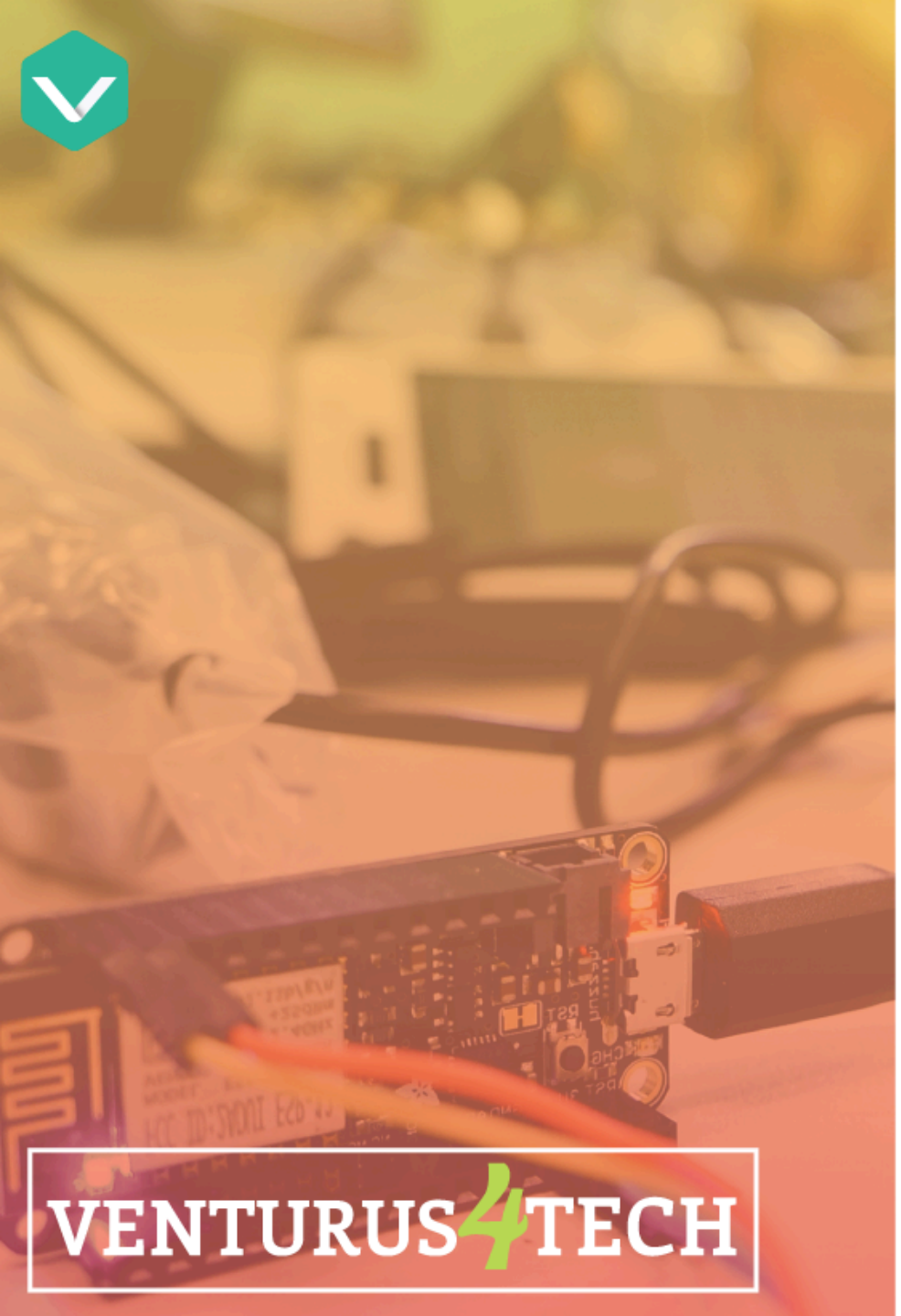
VENTURUS4TECH

Caso de Teste

Categorias

- Positivo
- Negativo
- Performance
- Stress
- Exploratório





Testes Automatizados

Por que automatizar?

- Tempo
- Economia
- Fator Humano
- Cobertura

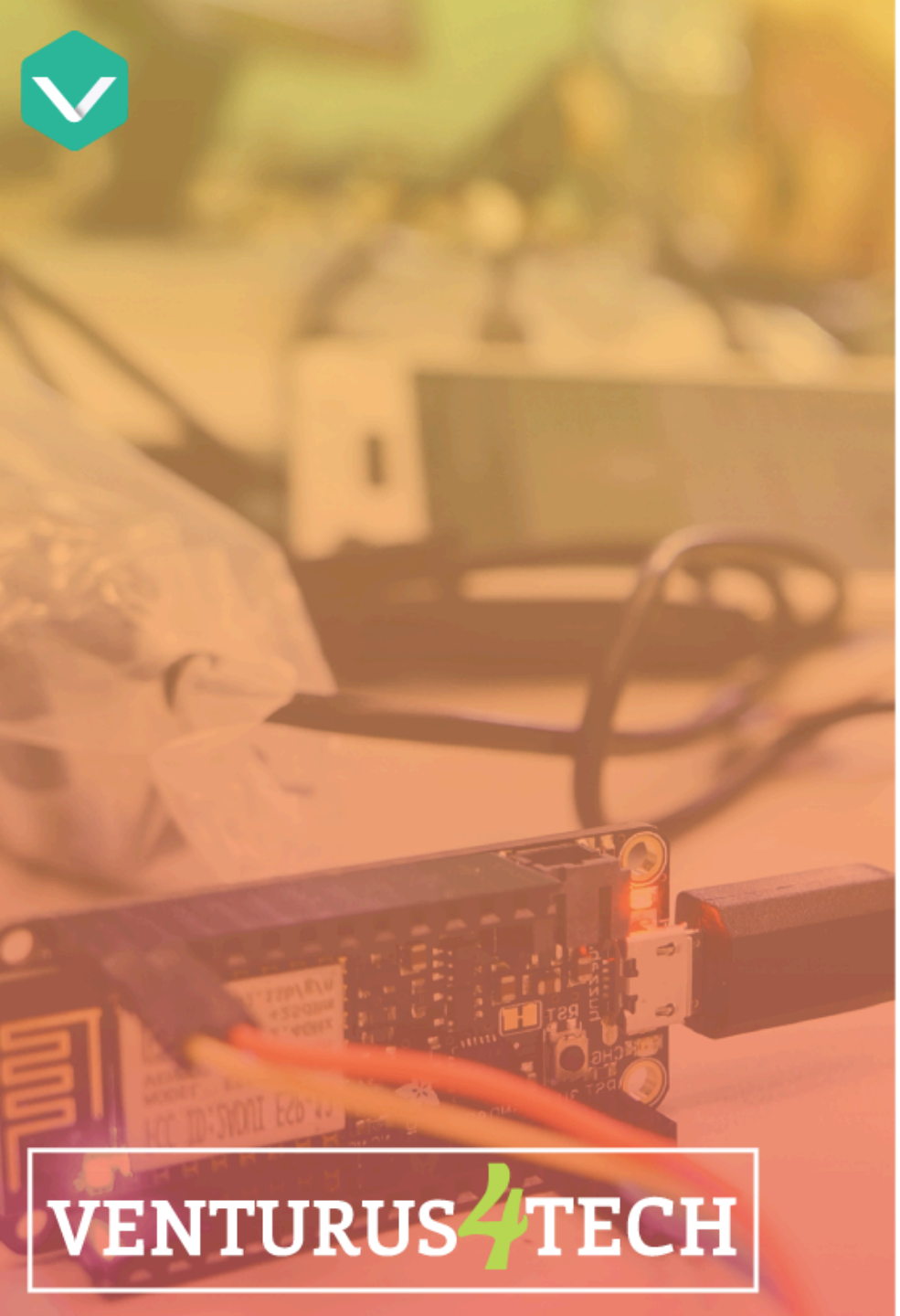
Por que não automatizar?

- Custo
- Manutenção
- Complexidade

Ferramentas de Automação

- Gerenciamento de Testes e Report: Testlink, Jira
- Testes Unitários: JUnit, testNG, AndroidJUnitRunner
- Testes “End-to-End”: Selenium, Sikuli, UI Automator
- Testes de stress e performance: JMeter

VENTURUS4TECH



JUnit

- Framework para automatizar testes unitários de código java
- Open-source
- Versão 4.X

JUnit

- Exemplo:
  - Código

```
public class Calculadora {

        public int soma(int x1, int x2){
                return x1+x2;
        }

        public int subtracao(int x1, int x2){
                return x2-x1;
        }
}
```

VENTURUS4TECH

JUnit

- Exemplo:
  - Teste

```
public class CalculadoraTest {
        Calculadora calc;

        @Before
        public void setUp(){
                calc = new Calculadora();
        }

        @Test
        public void testSoma() {
                assertEquals(3, calc.soma(1, 2));
        }

        @Test
        public void testSubtracao() {
                assertEquals(-1, calc.subtracao(1, 2));
        }
}
```

VENTURUS4TECH

JUnit

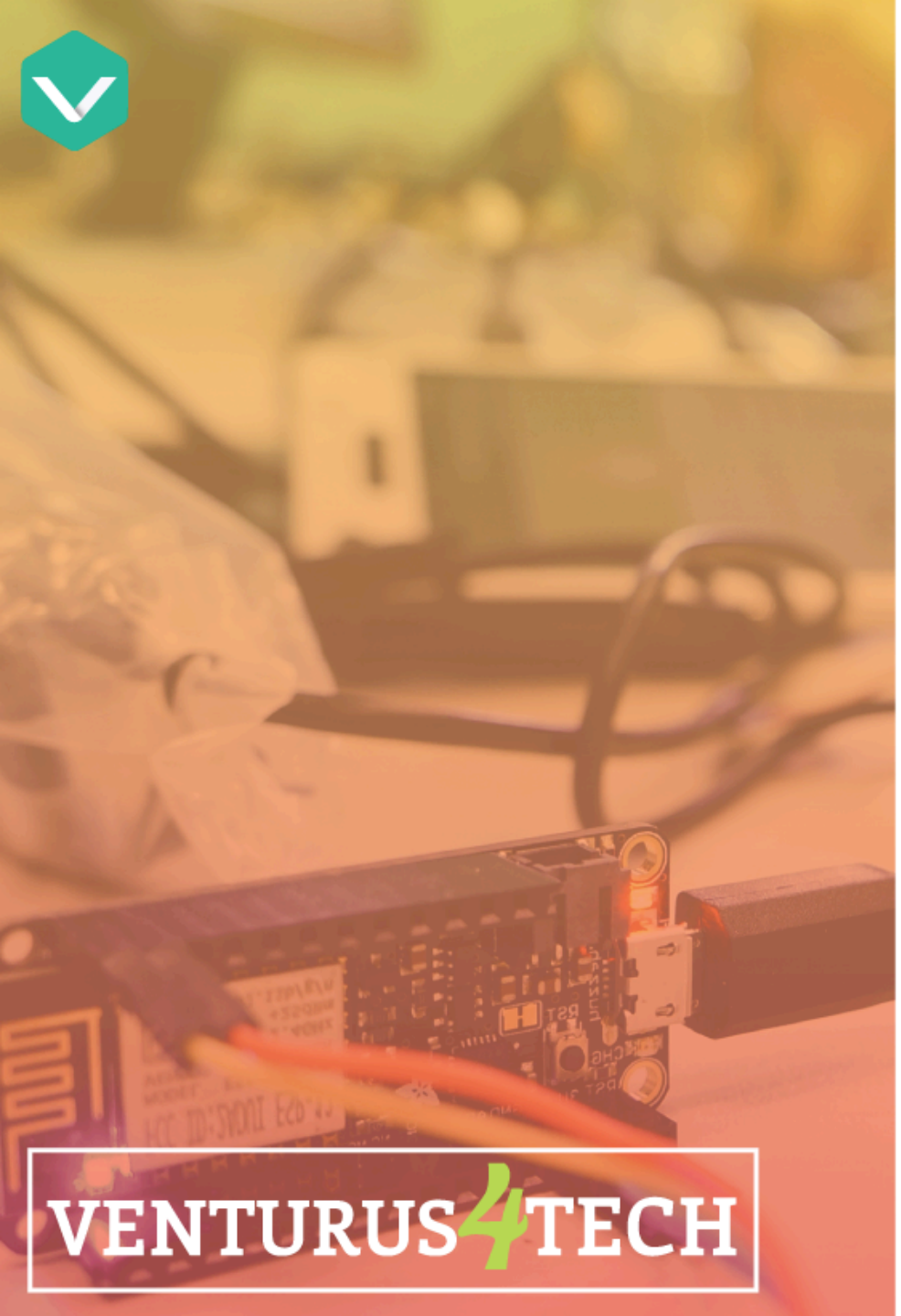
```
public class CalculadoraTest {
        Calculadora calc;

        @Before
        public void setup(){
                calc = new Calculadora();
        }

        @Test
        public void testSoma() {
                assertEquals(3, calc.soma(1, 2));
        }

}
```

@Before: Execute este método antes de cada teste

@Test: Execute este método como um teste

Confirme se os valores são iguais

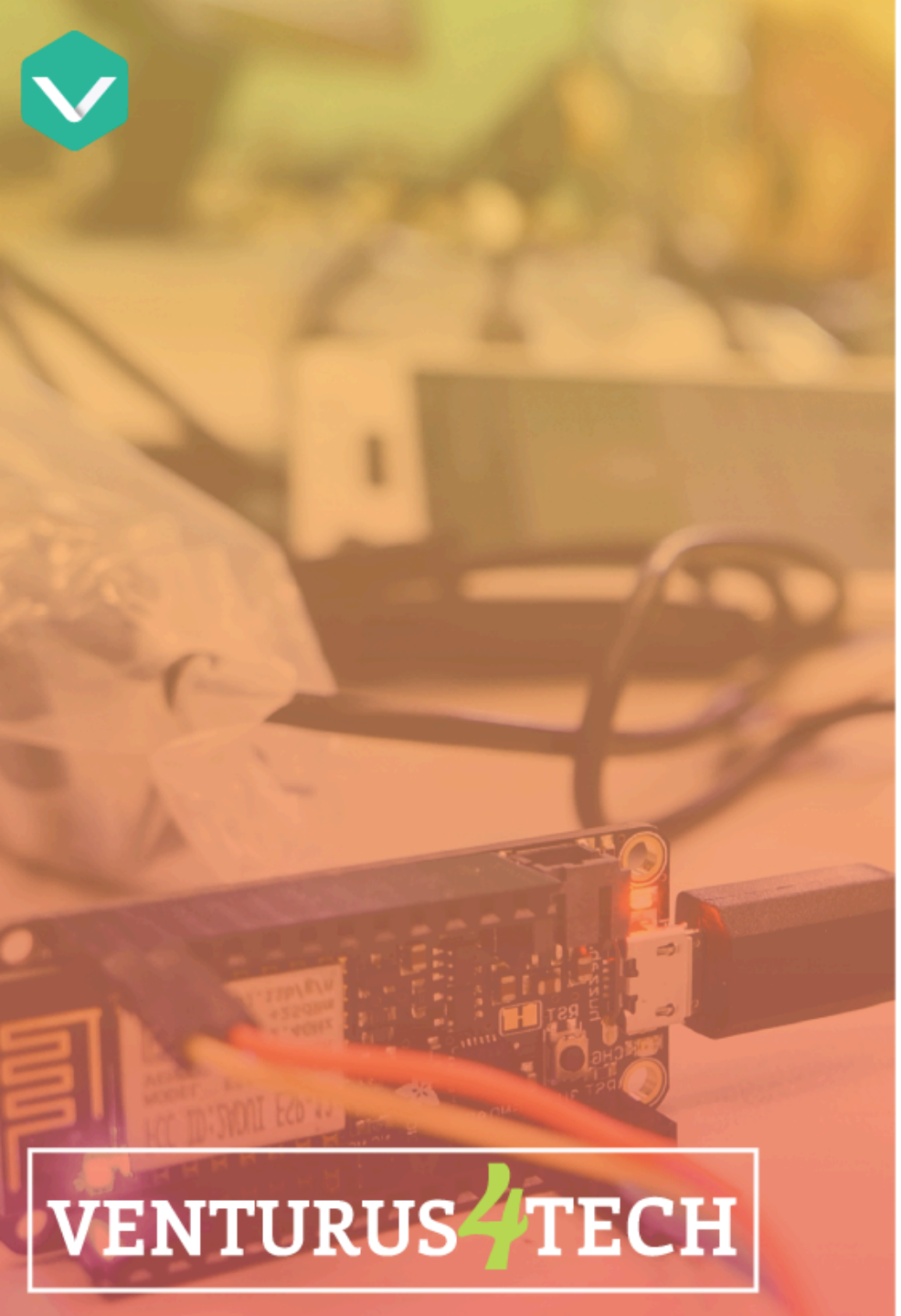
VENTURUS4TECH



JUnit

Anotações

- **@Test**: é o próprio teste a ser executado
- **@Before**: é o método que será executado antes de cada @Test
- **@BeforeClass**: é o método que será executado apenas uma vez, antes de todos os @Test
- **@After**: é o método que será executado depois de cada @Test
- **@AfterClass**: é o método que será executado ao final de todos os métodos @Test
- **@Ignore**: é o métdo será ignorado



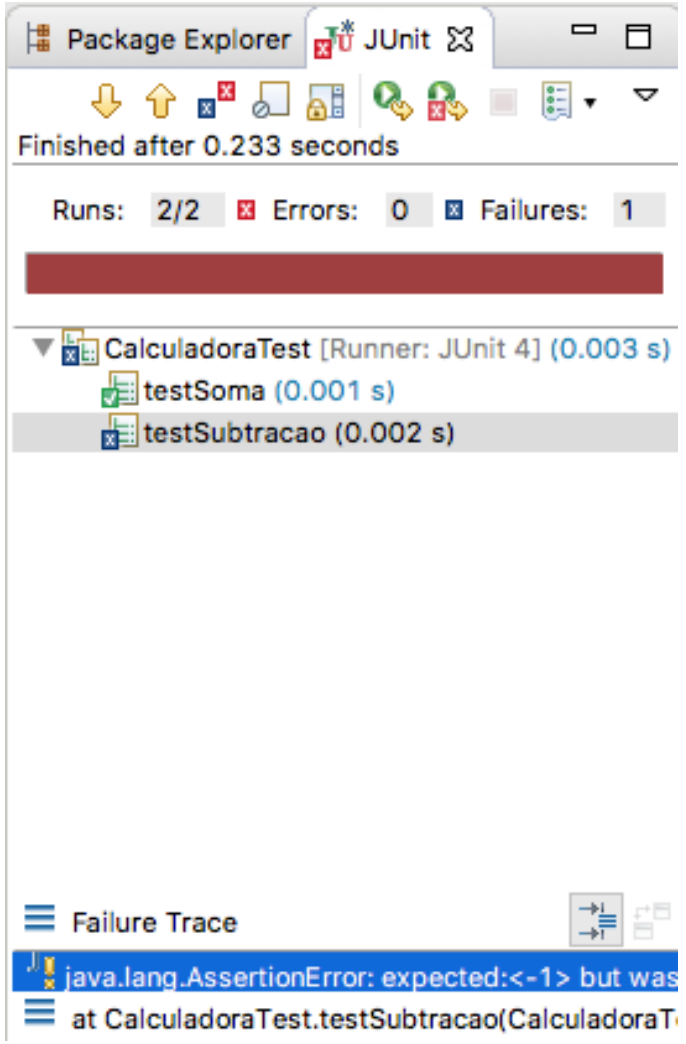
JUnit

Métodos

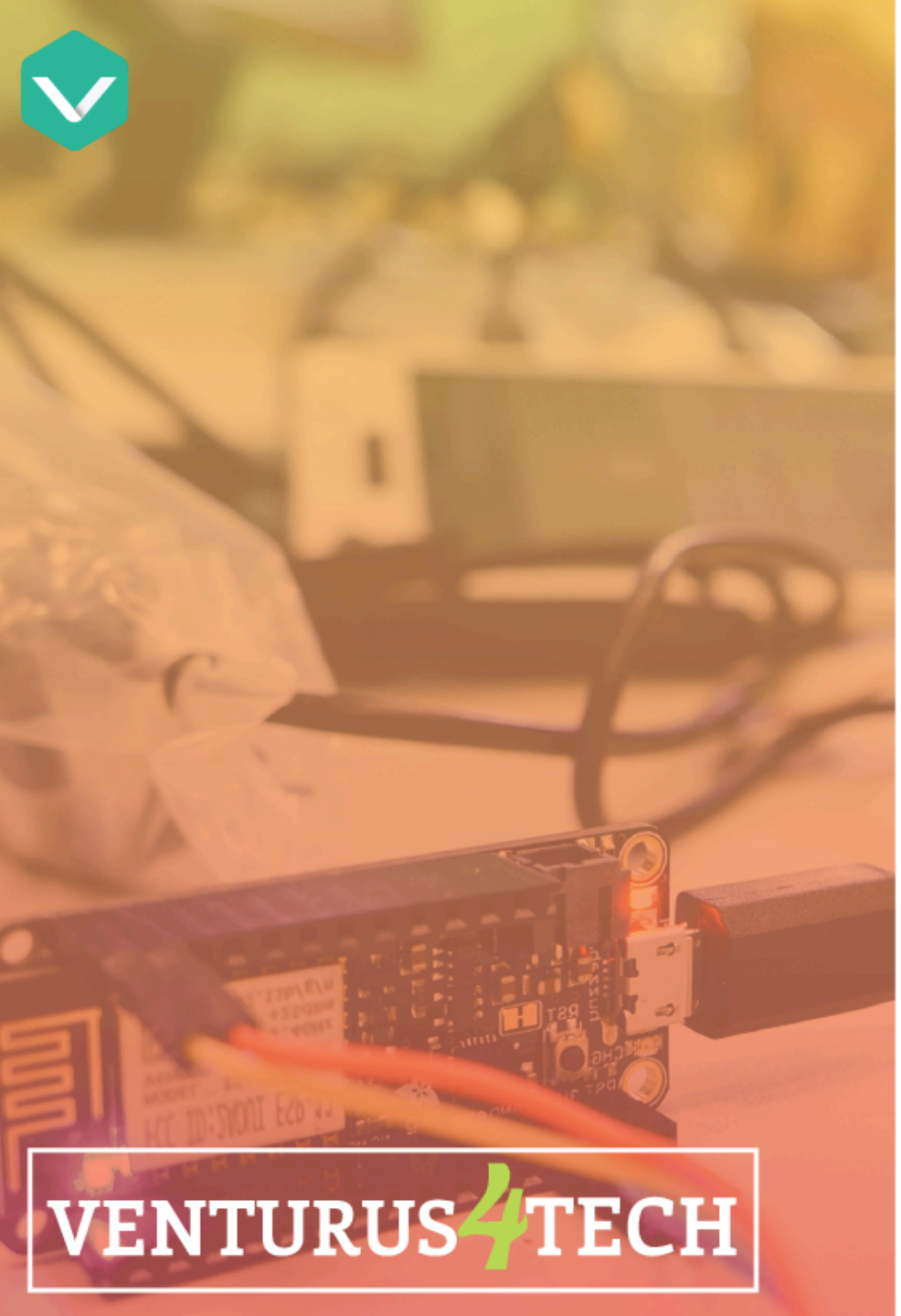
- assertEquals
- assertTrue
- assertFalse
- assertNull
- assertNotNull
- assertSame
- assertNotSame

JUnit

- Eclipse: Junit View
  - Forma mais amigável para acompanhar a execução dos testes e ver os resultados



VENTURUS4TECH



JUnit

Exercício 1

Crie a classe calculadora com o código

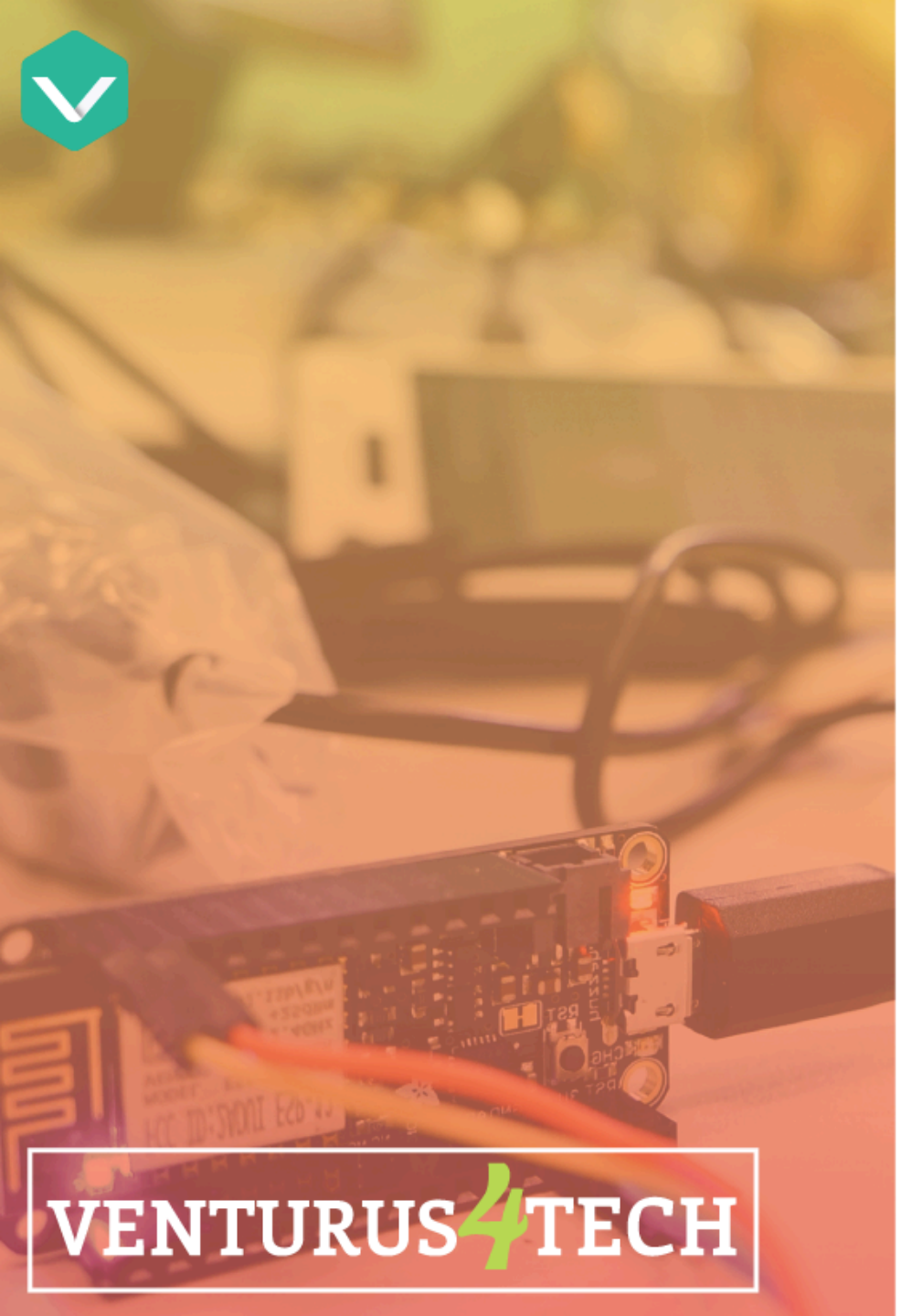
```
public class Calculadora {

        public int soma(int x1, int x2){
                return x1+x2;
        }
}
```

JUnit

- Exercício 1 – continuação
  - Crie um novo JUnit test case e coloque o código abaixo

```
public class CalculadoraTest {
        Calculadora calc;

        @Before
        public void setup(){
                calc = new Calculadora();
        }

        @Test
        public void testSoma() {
                assertEquals(3, calc.soma(1, 2));
        }
}
```

VENTURUS4TECH



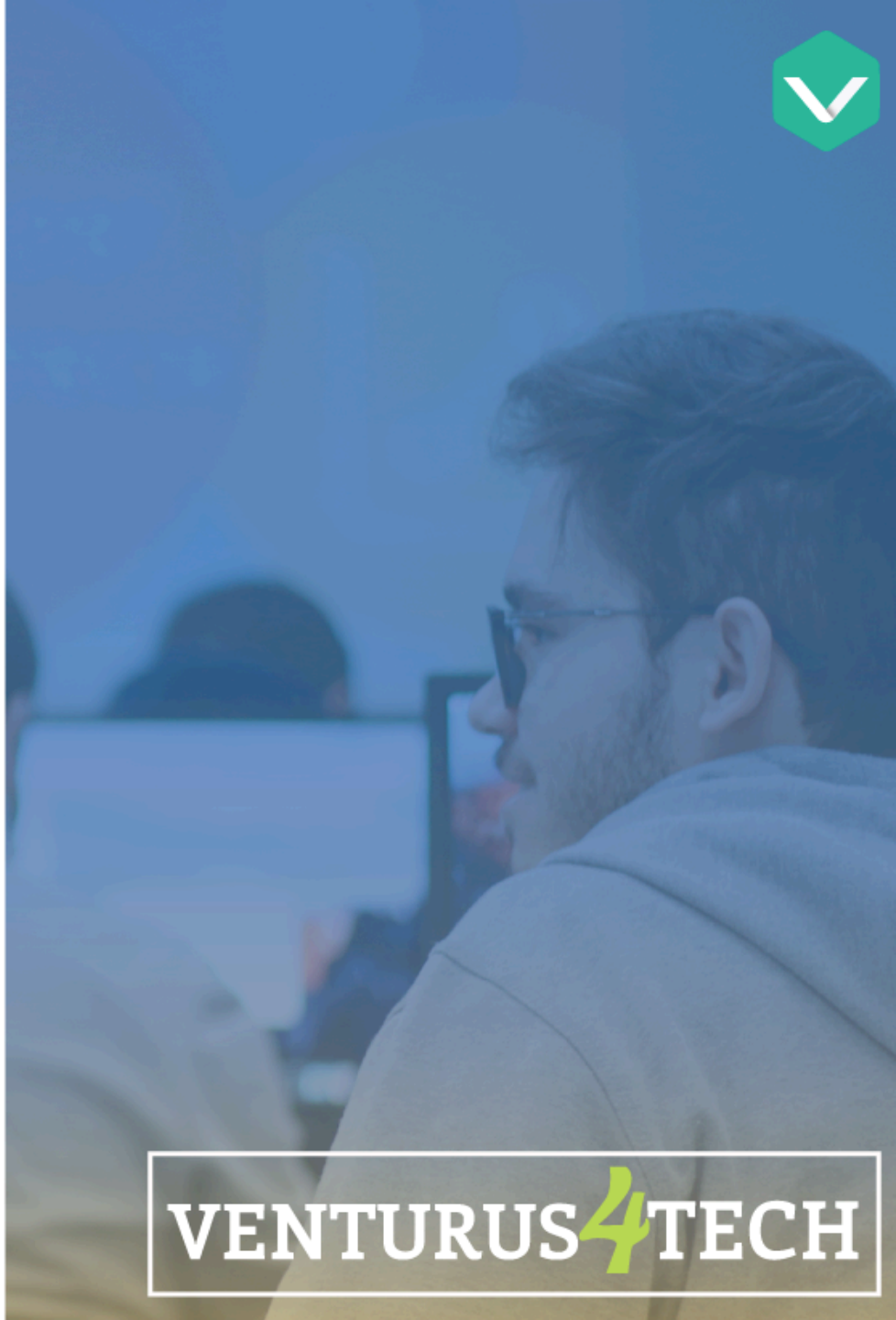
JUnit

Exercício 2

- Implemente outras funções da calculadora (Exemplo: subtração, multiplicação, divisão, X^Y)
- Para cada função, crie um teste

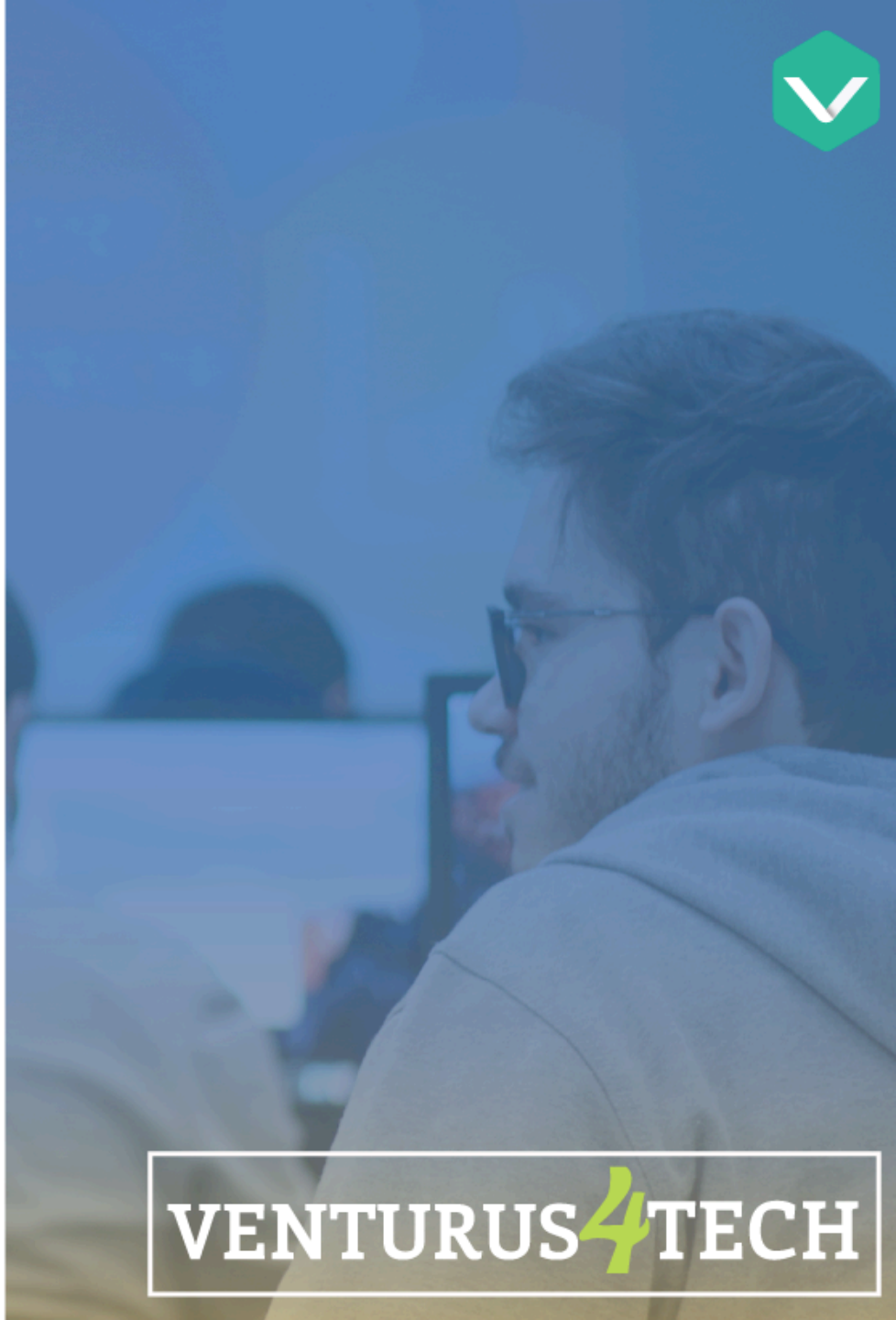
Selenium

- Automação de testes para aplicações web
- Pode executar ações utilizando o próprio navegador (Safari, Chrome, Firefox, Internet Explorer)
- Disponível para java, C#, Ruby, Python e JavaScript



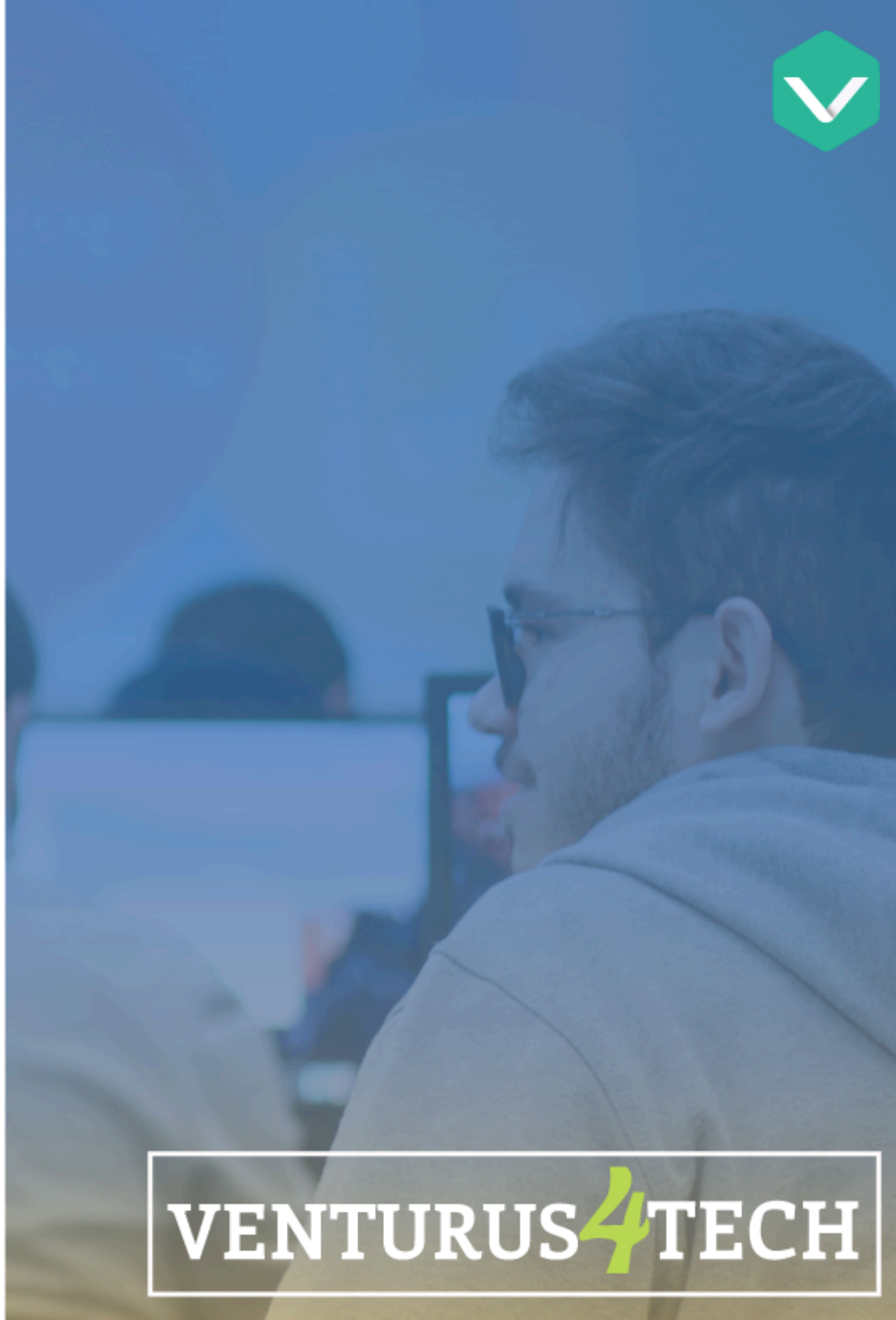
Selenium

- Selenium WebDriver
  - API para o desenvolvimento dos testes
- Selenium Grid
  - Execução em diferentes máquinas e navegadores
- Selenium IDE
  - Add-on para o Firefox



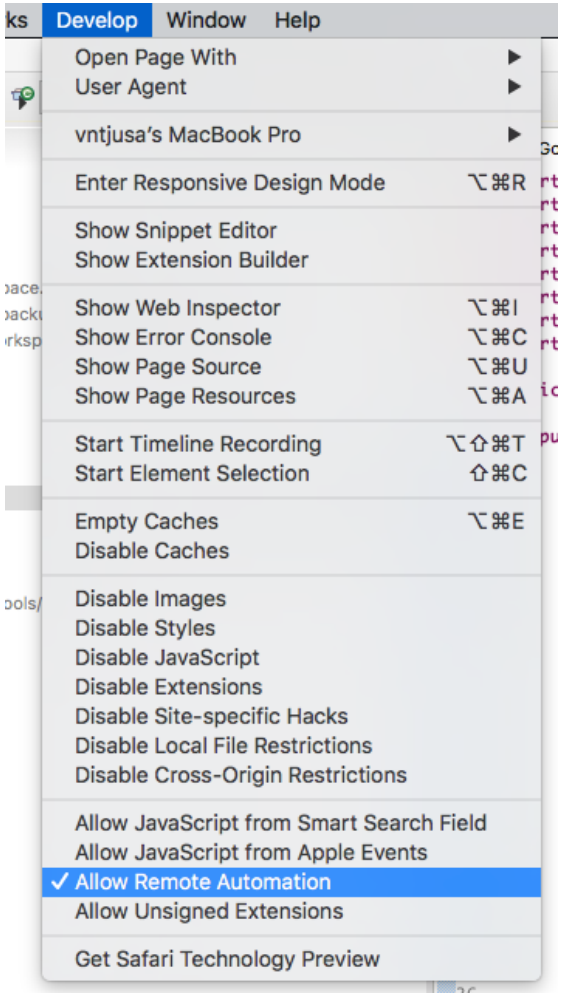
Selenium

- Setup Inicial
  - Baixar o Selenium
  - A partir da versão 3.0.1, existe a necessidade de baixar os drivers para Firefox (geckodriver) e Chrome (chromedriver)
    - geckoDriver: https://github.com/mozilla/geckodriver/releases
    - chromeDriver: https://sites.google.com/a/chromium.org/chromedriver/downloads



Selenium

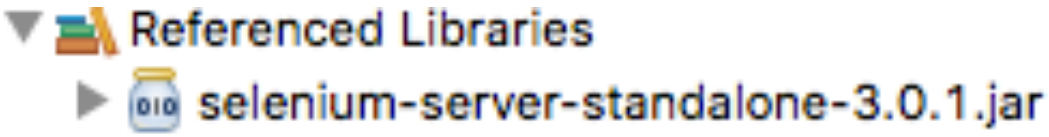
- No Safari, é necessário habilitar o Remote Automation

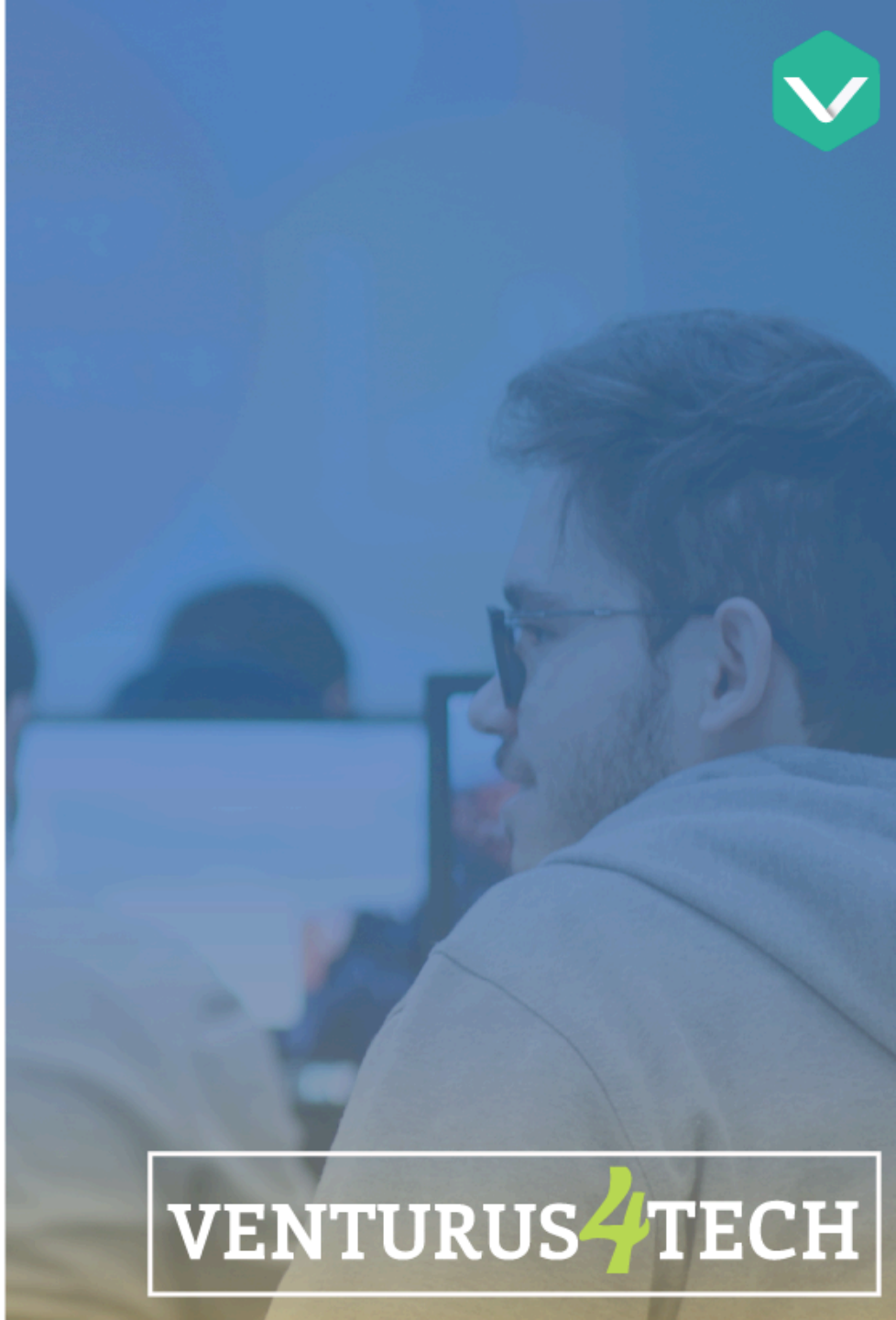


VENTURUS4TECH

Selenium

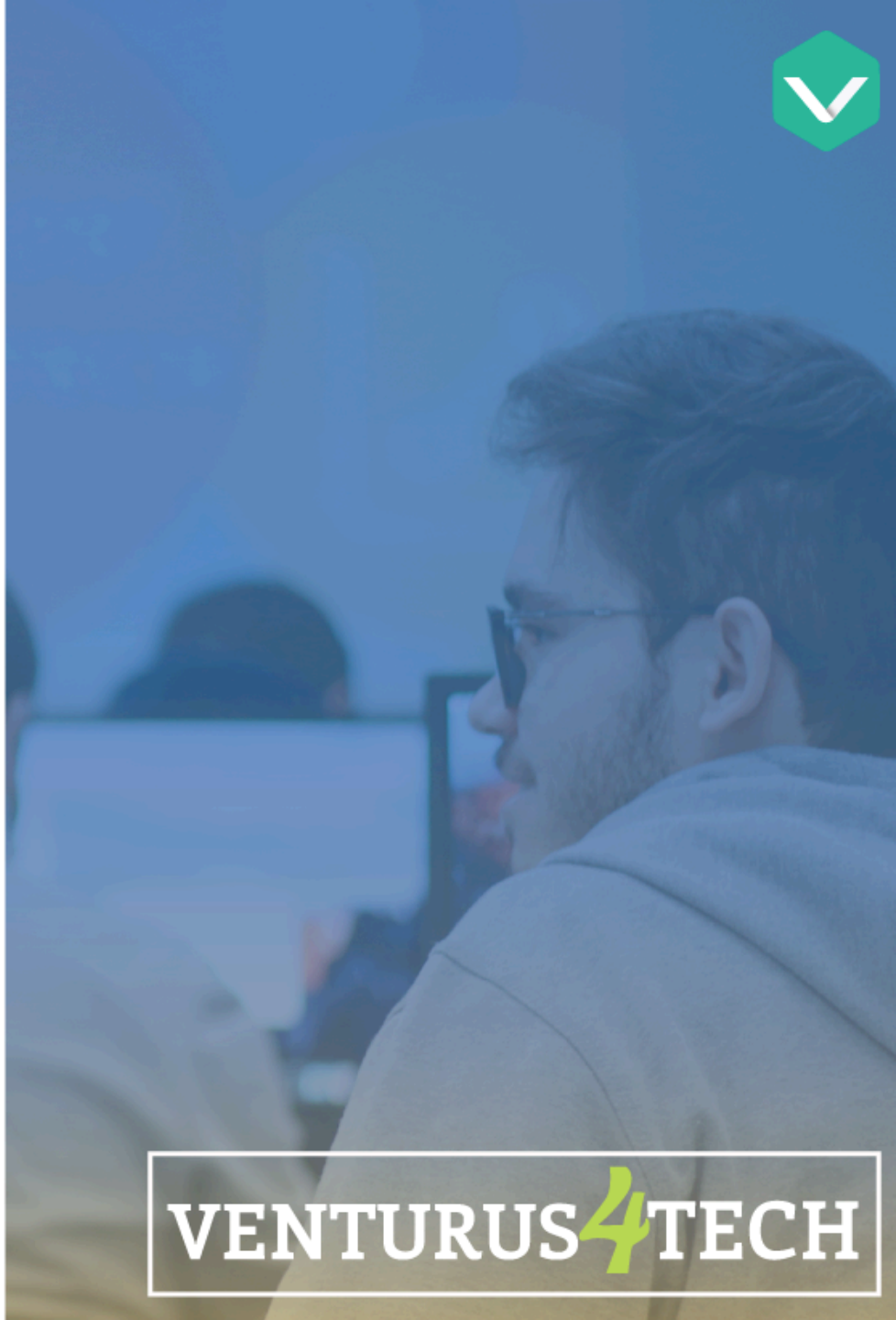
- Na ferramenta de desenvolvimento, crie um novo projeto e adicione a biblioteca do selenium





Selenium

- Componentes importantes
  - **WebDriver**: módulo que se conecta ao navegador
  - **WebElement**: módulo que representa um elemento HTML



Selenium

- Exemplo de Código

```
public class FindOnGoogle {

    public static void main(String[] args) throws InterruptedException {
        WebDriver driver;
        driver = new SafariDriver();
        driver.get("http://www.google.com");

        WebElement inputText;
        inputText = driver.findElement(By.name("q"));

        inputText.sendKeys("teste");
        inputText.sendKeys(Keys.RETURN);

        Thread.sleep(10000);
        driver.quit();
    }
}
```

VENTURUS4TECH

Selenium

- Entendendo o código

```
WebDriver driver;
driver = new SafariDriver();

driver.get("http://www.google.com");

WebElement inputText;
inputText = driver.findElement(By.name("q"));

inputText.sendKeys("teste");
inputText.sendKeys(Keys.RETURN);

Thread.sleep(10000);
driver.quit();
```

Instanciando o driver para o Safari

Carrega a página

Digita “teste”

Digita a tecla Enter

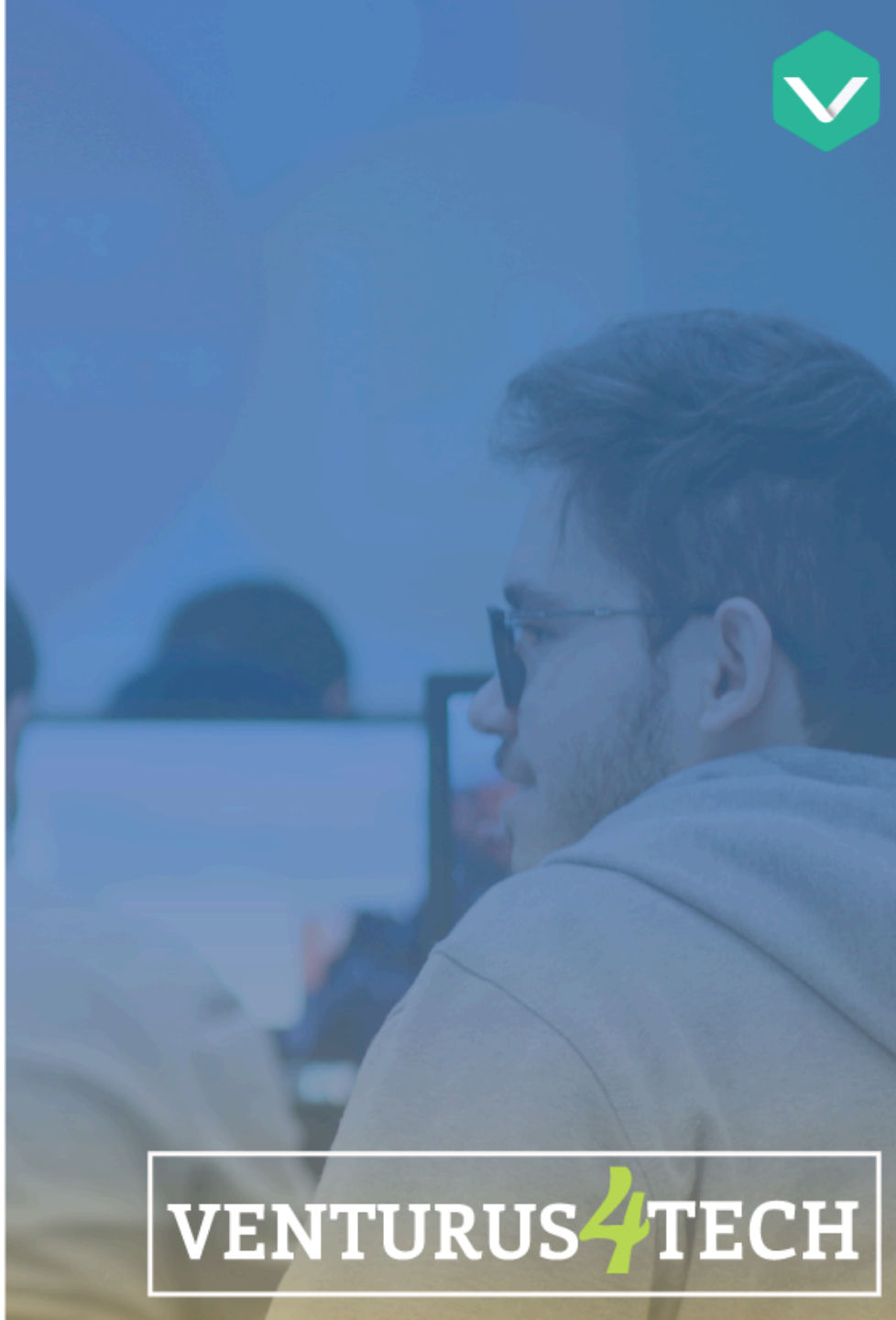
Localiza o text box

Finaliza a conexão com o navegador

VENTURUS4TECH

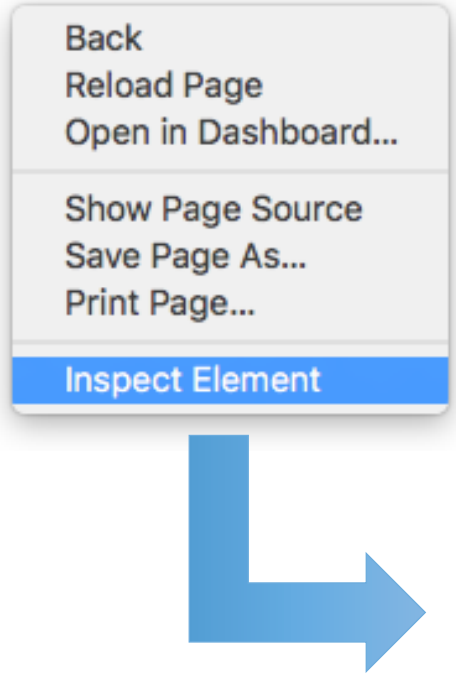
Selenium

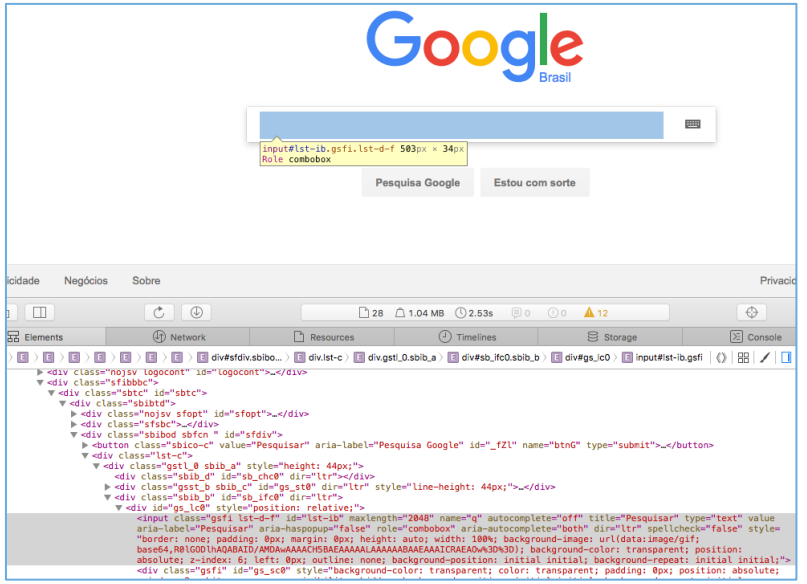
- findElement
  - By
    - id
    - name
    - className
    - xpath
    - tagName
    - cssSelector
    - linkText
    - partialLinkText



Selenium

- Como achar o elemento
  - A maioria dos navegadores tem o modo de desenvolvedor
  - Clique no botão direito do mouse e selecione “Inspect Element” para visualizar o código fonte da página





VENTURUS4TECH

Selenium

- Clique em Elements

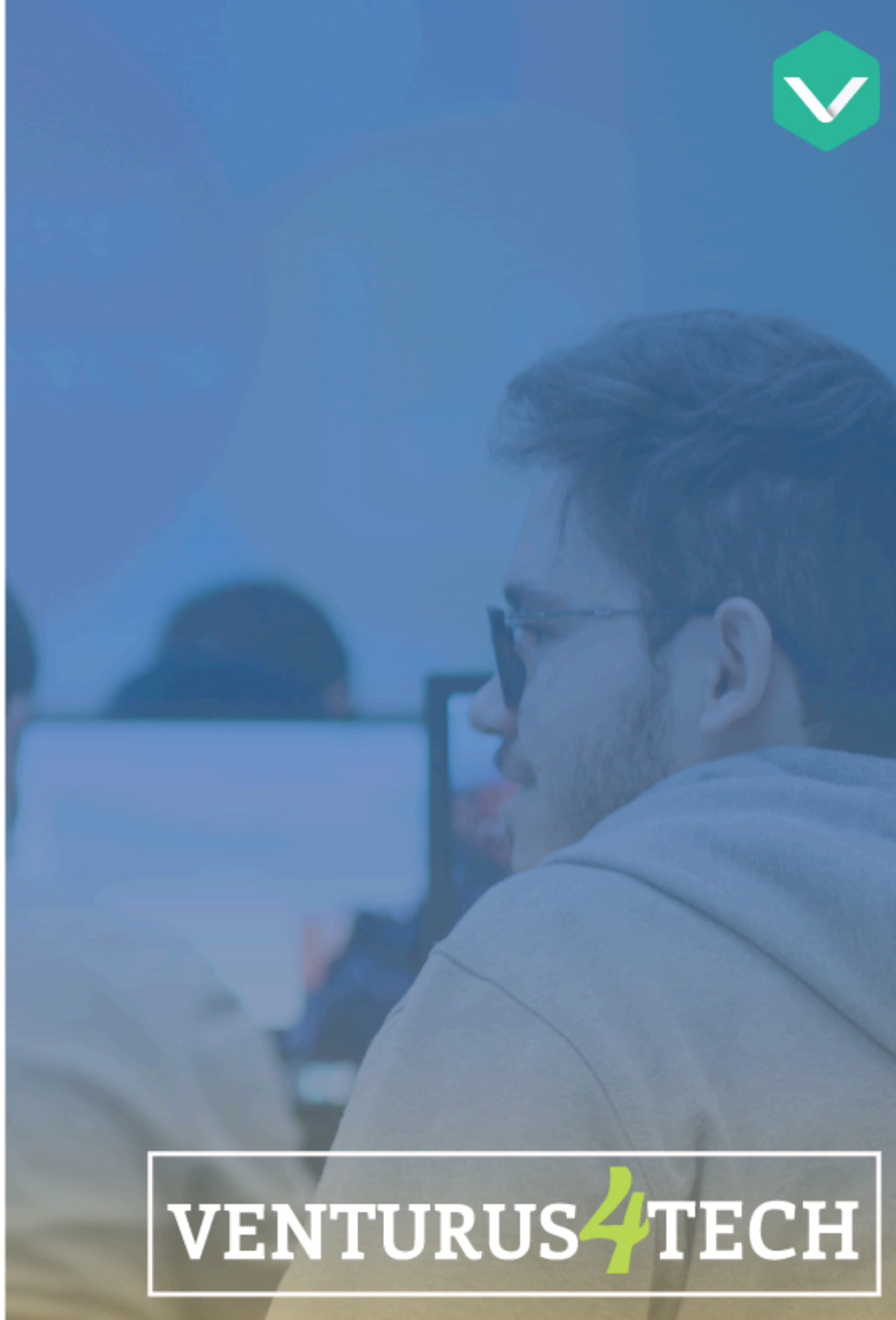
VENTURUS4TECH

Selenium

- Sincronia entre código e a aplicação web
  - Delay para carregar a aplicação web

- Espera Explícita
  - WebDriverWait
  - Aguarda por condições ou elementos na aplicação web

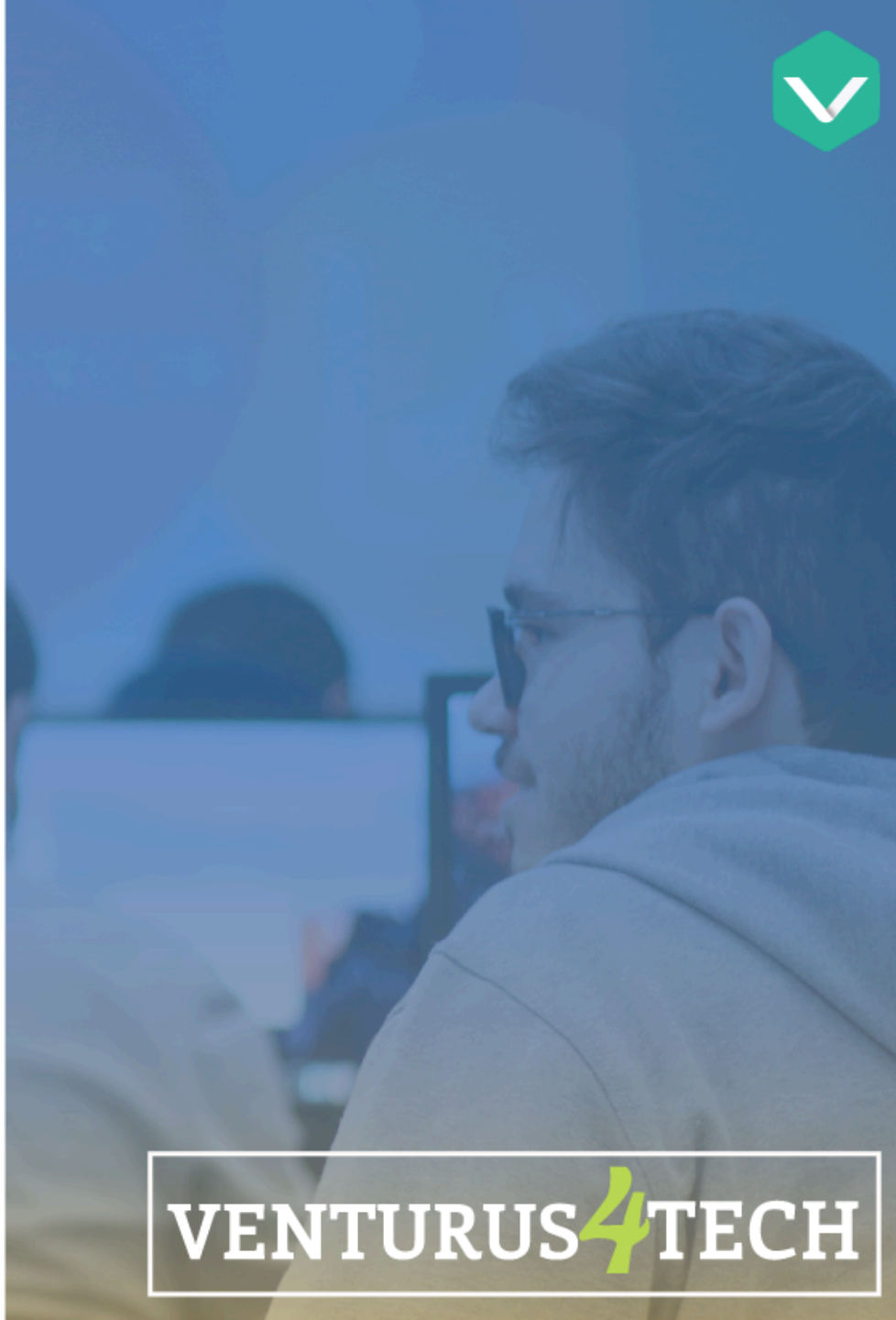
```
WebDriverWait wait = new WebDriverWait(webDriver, timeoutInSeconds);

wait.until(ExpectedConditions.visibilityOfElementLocated(By.id<locator>));
```



Selenium

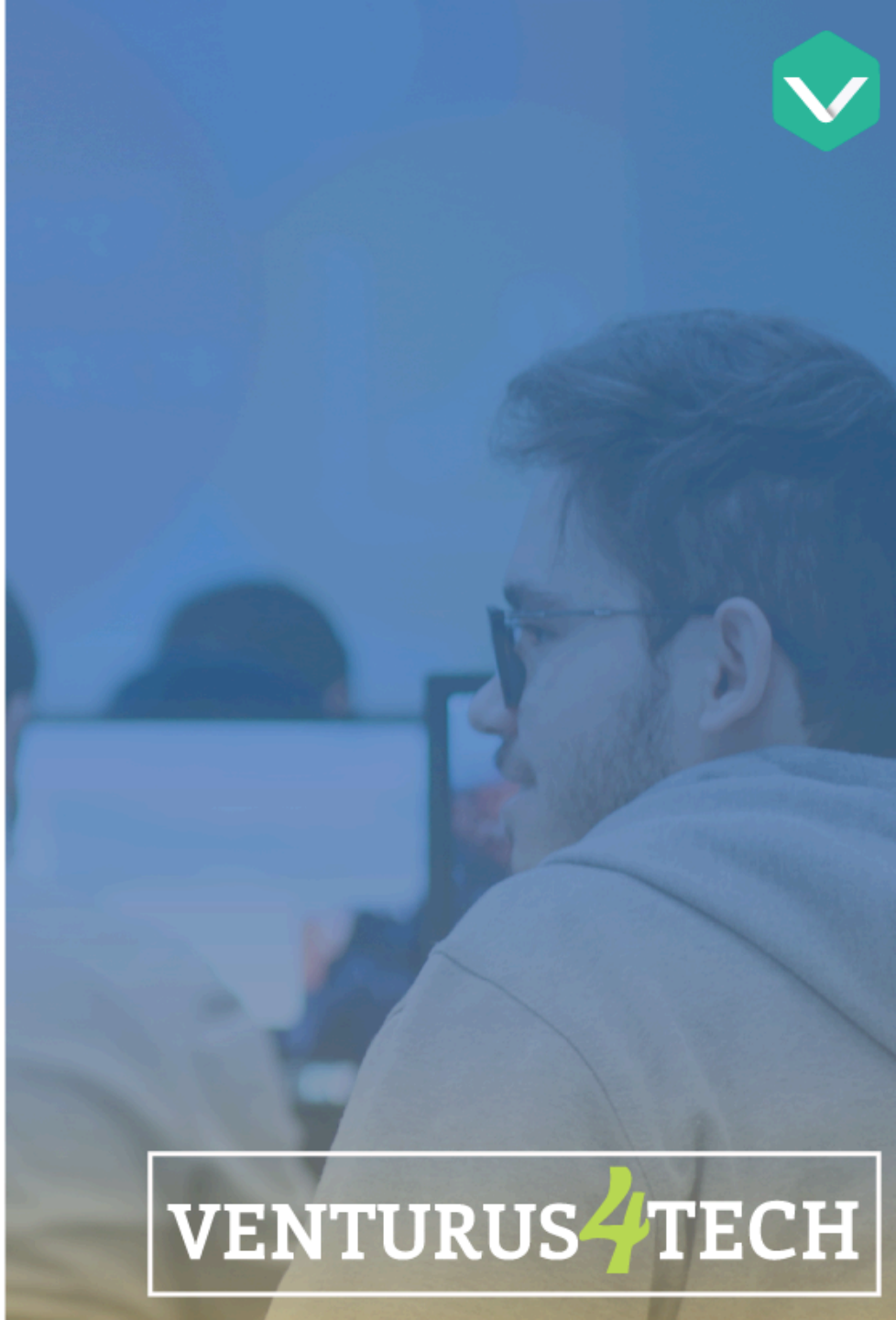
- Espera Implícita
  - WebDriver deve esperar um determinado tempo para achar os elementos

```
driver = new SafariDriver();
driver.manage().timeouts().pageLoadTimeout(10, TimeUnit.SECONDS);
driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

driver.get("http://www.google.com");
```



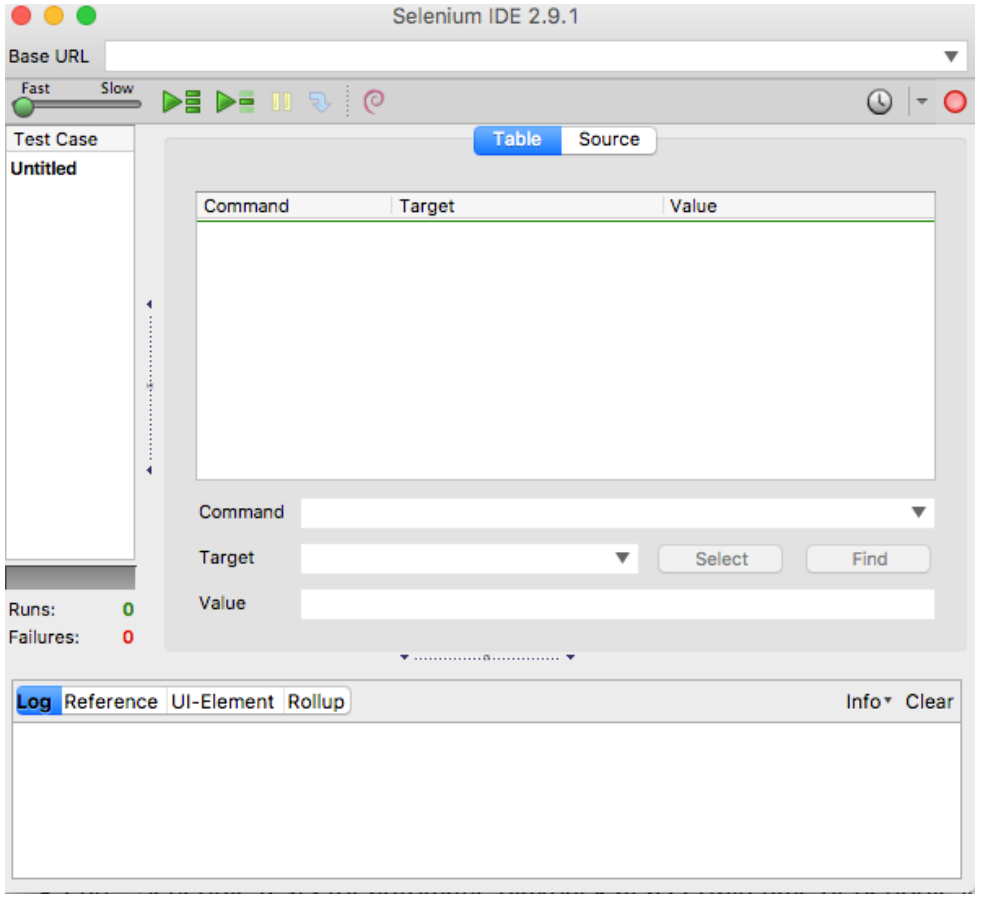
Selenium

- Exercício 1
  - Acessar o google e fazer uma busca
- Exercício 2
  - Acessar o site do webmotors e faça uma consulta
- Exercício 3
  - Usando JUnit e Selenium, valide sua aplicação
  - crie um teste para ligar a lâmpada e verifique o status



Selenium

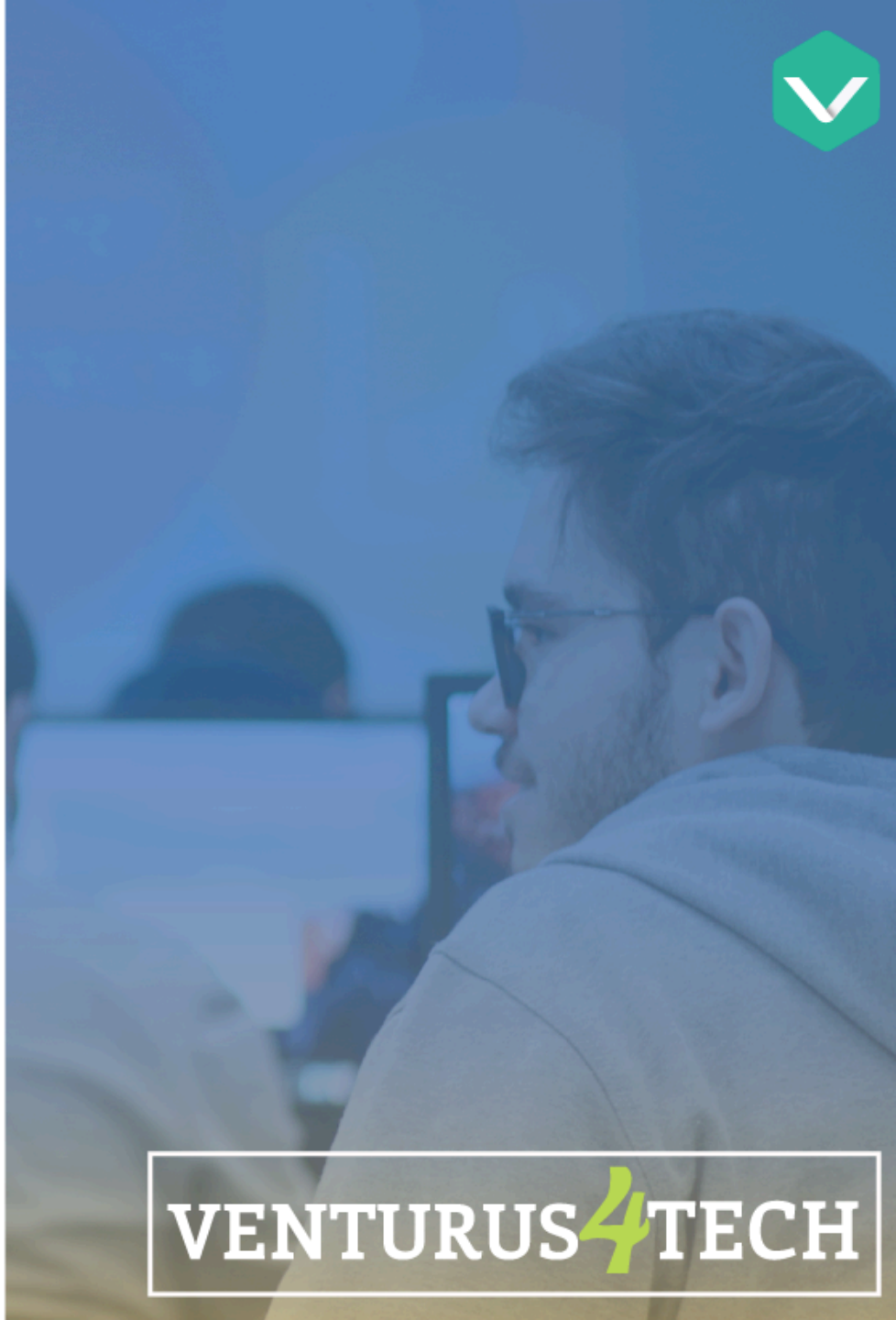
- IDE
  - Add-on para Firefox
  - Grava os passos executados diretamente do navegador

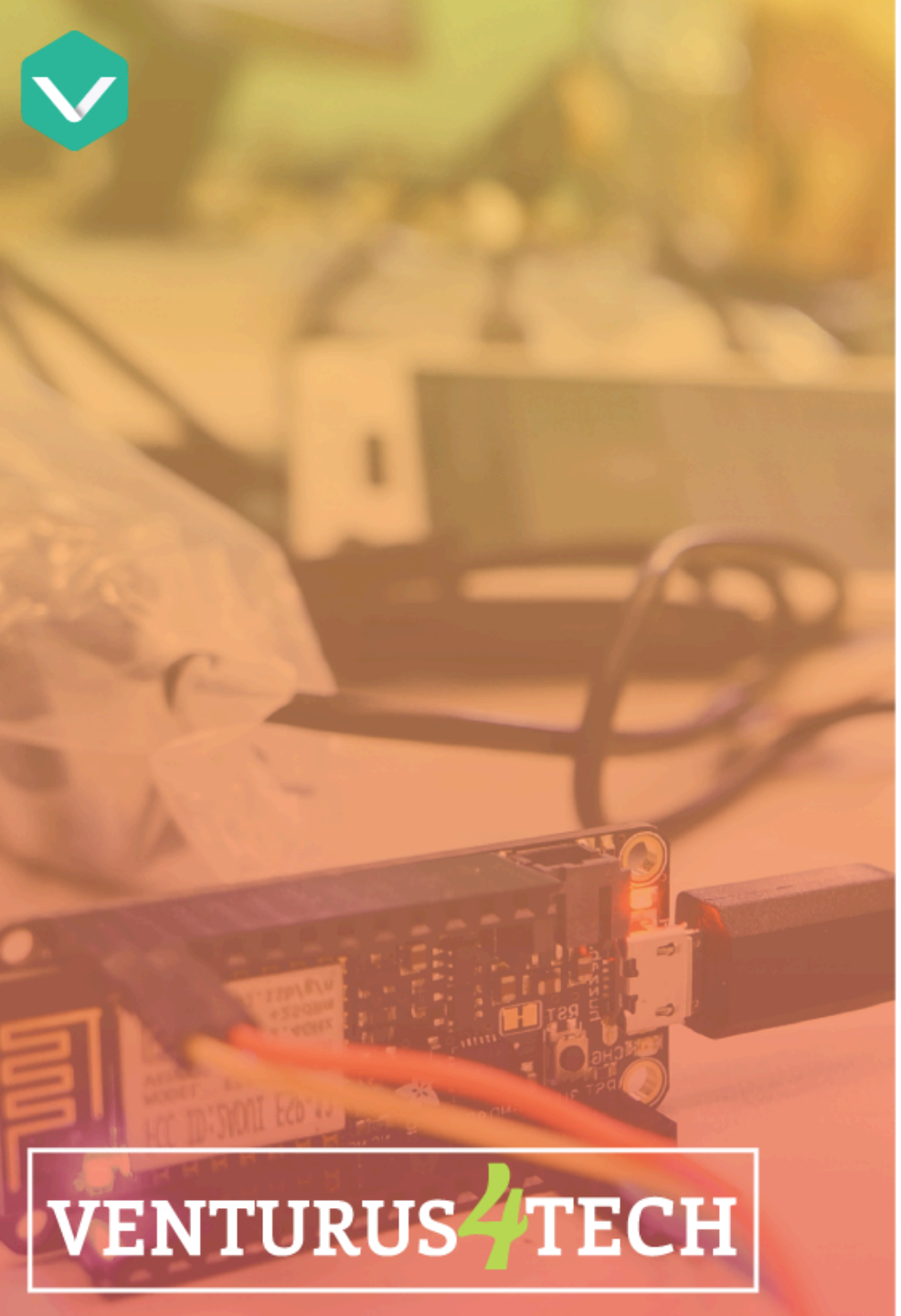


VENTURUS4TECH

Selenium

- Observações
  - Algumas funções podem não funcionar em alguns navegadores
  - Por exemplo, no exercício do site webmotors, o preenchimento dos campos de valores não funcionam corretamente usando o Safari, ao contrário do Firefox.





Page Object Pattern

- Separar o código do teste da implementação da página
- Facilitar a manutenção do código
- Facilitar a criação de testes
- PageFactory

Page Object Pattern

- Código do Objeto

```
public class GoogleSearchPage {
        private WebDriver driver;

  @FindBy(name = "q")
   private WebElement searchBox;

   public GoogleSearchPage(WebDriver driver){
        this.driver = driver;
        PageFactory.initElements(driver, this);
   }

   public void searchFor(String text) {
      // We continue using the element just as before
      searchBox.sendKeys(text);
      searchBox.submit();
   }
}
```

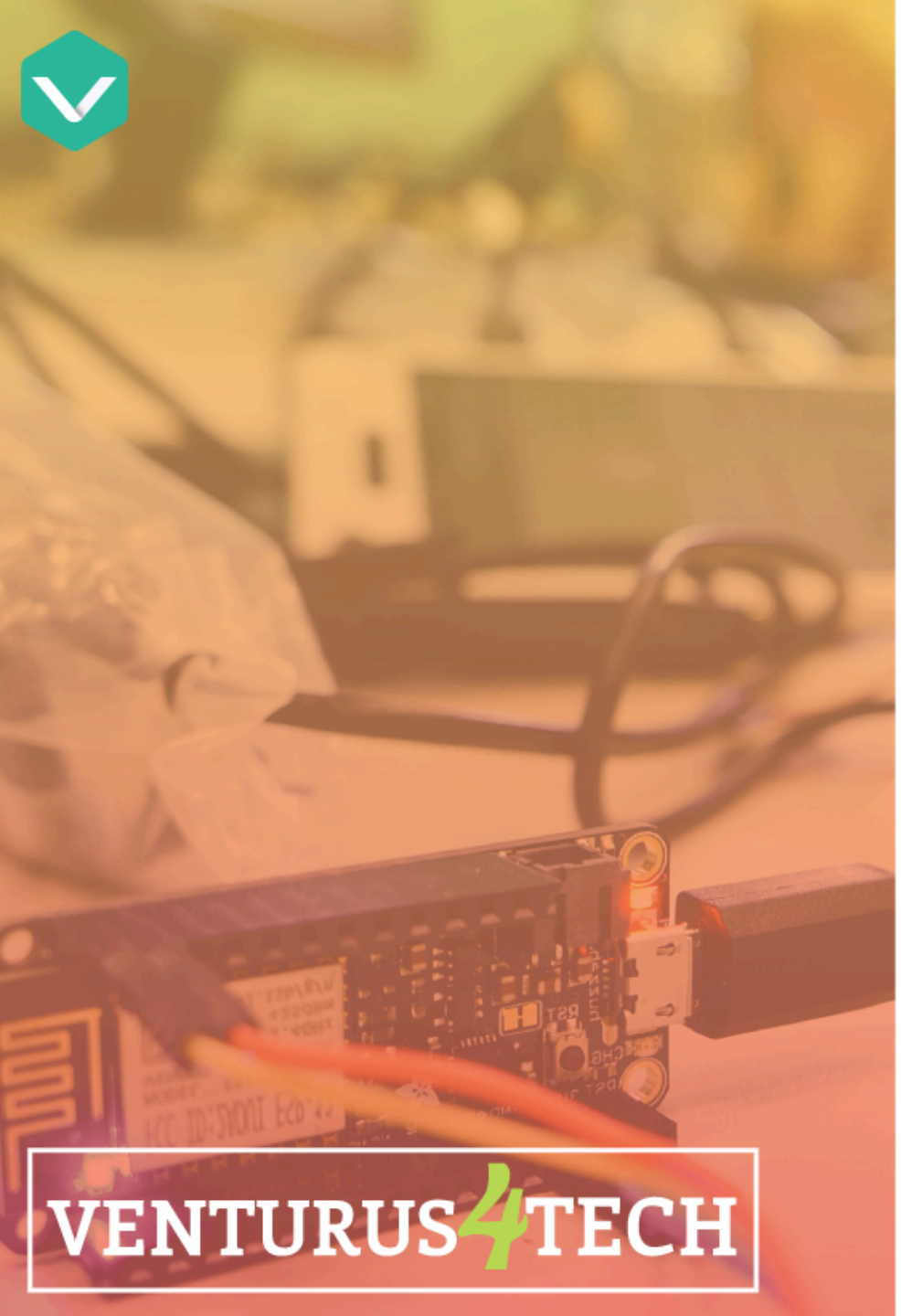
VENTURUS4TECH

Page Object Pattern

- Código do Teste

```
public class TestingGoogleSearch {
        WebDriver driver;
        String googleAddress = "http://www.google.com/";
        GoogleSearchPage googleSearchPage;

        @Before
        public void setUp() throws Exception {
                driver = new SafariDriver();
                driver.get(googleAddress);
                googleSearchPage = new GoogleSearchPage(driver);
        }

        @Test
        public void test() throws InterruptedException {
                googleSearchPage.searchFor("teste");
                Thread.sleep(5000);
        }
        …
```

VENTURUS4TECH



Referências

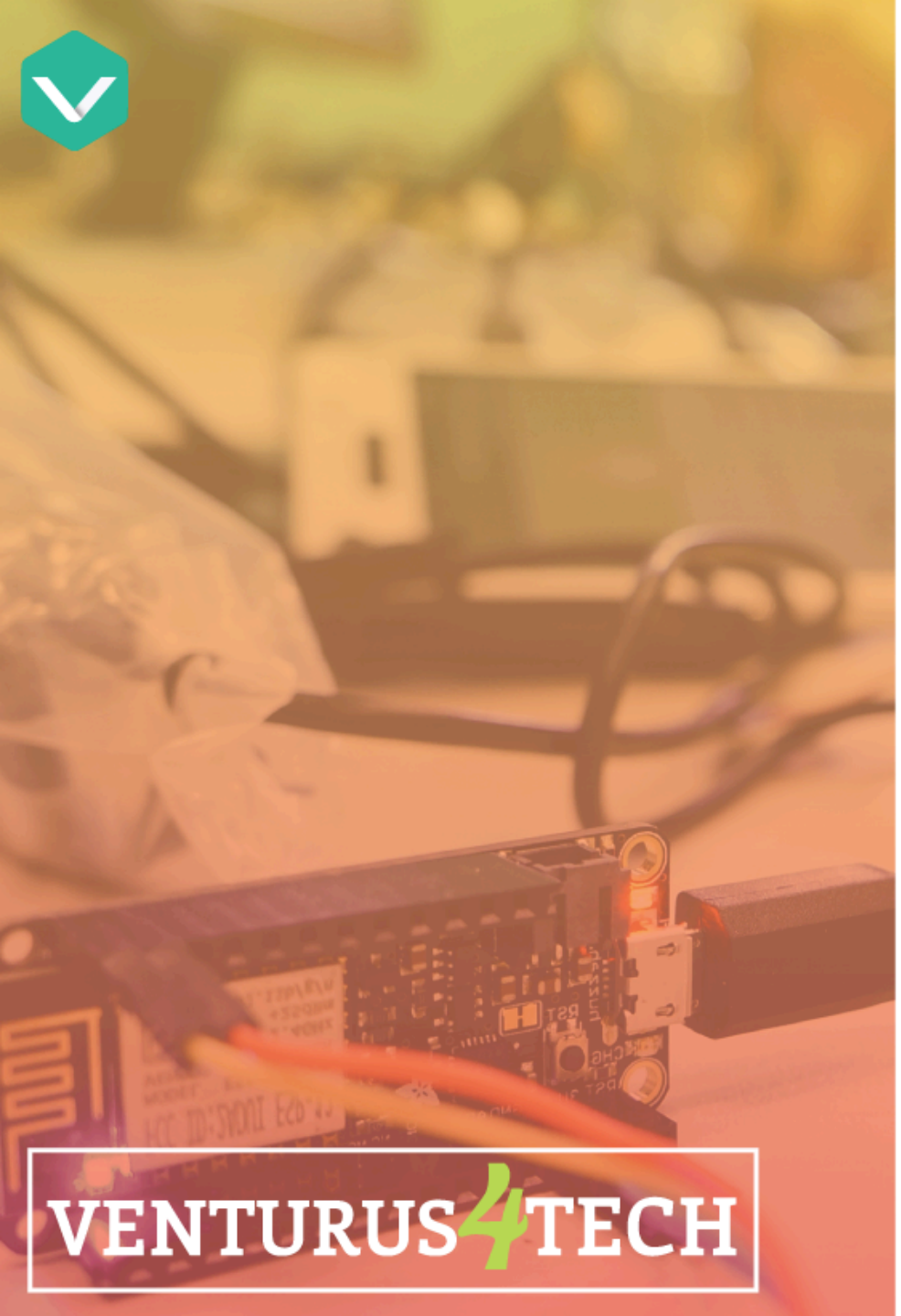
Teste de software - pt.wikipedia.org/wiki/Teste_de_software

Selenium – www.seleniumhq.org

JavaDocs do Selenium - seleniumhq.github.io/selenium/docs/api/java/

JUnit – www.junit.org

Google Testing Blog - testing.googleblog.com



Referências

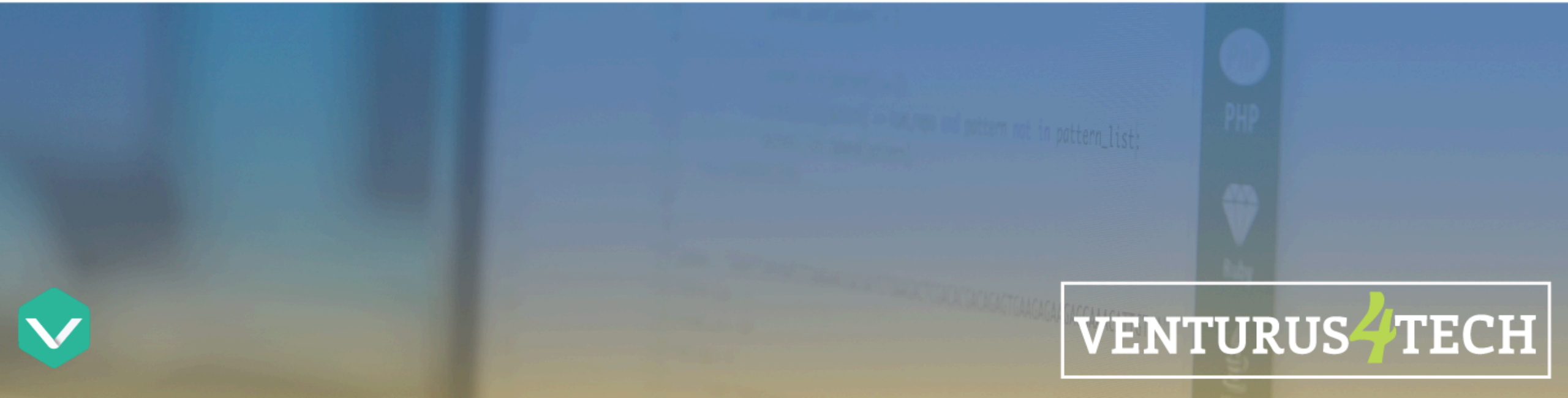
ISO/IEC/IEEE 29119: Software and systems engineering - Software testing

IEEE 829: IEEE Standard for Software and System Test Documentation

James Bach – www.satisfice.com

Contato

Junior Toshiharu Saito – junior.saito@venturus.org.br
Marcio Cunha – marcio.cunha@venturus.org.br



Obrigado!

VENTURUS4TECH