

# **Automação de testes com Selenium WebDriver Page Objects e Arquitetura**

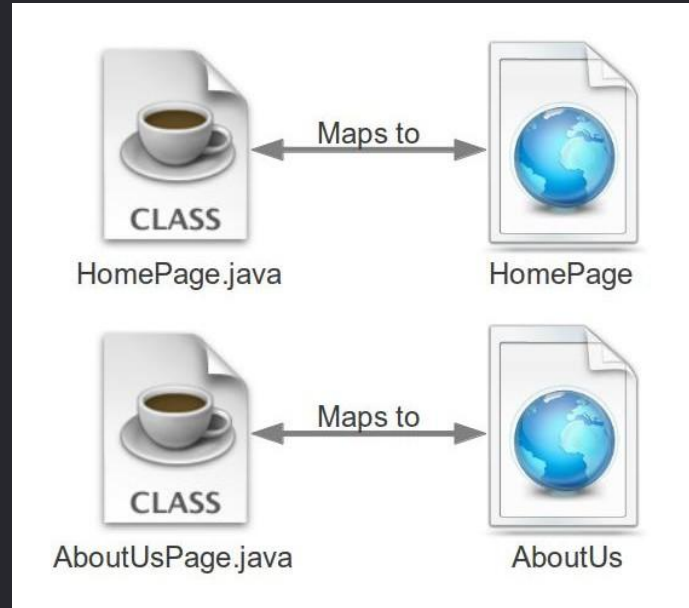
Antônio Moraes Trindade

<https://about.me/amtrindade>

# O que é Page Objects?

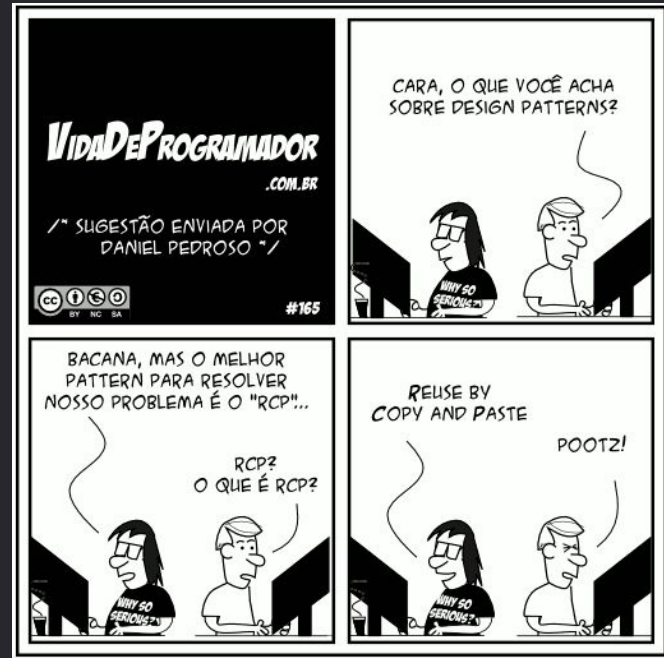
É um padrão de projeto (design pattern) muito utilizado para elaborar a arquitetura de um projeto de automação de testes.

O design pattern se consegue através da refatoração do código e generalização do problema.



# O que é Page Objects?

Design patterns ou Padrões de projeto são soluções de templates abstratas de alto nível. São como um desenho técnico ou documentação de uma arquitetura para soluções, e não uma solução por si própria. Não é um framework que simplesmente poder ser aplicado.



# Por que o Page Objects?

---

- Reduz a quantidade de código duplicado. Encapsula o estado interno de uma página em um único objeto.
- Alterações na UI afetam apenas a um único objeto da página e não os códigos de testes.
- Reutilização de código: Capaz de usar o mesmo objeto de uma página em uma variedade de casos de testes.

# Por que o Page Objects?

---

Cada funcionalidade de uma página vira um serviço.

- Por exemplo: Uma tela de cadastro cada operação vira uma função:

- Adicionar
- Procurar
- Editar
- Excluir

# Regras e comportamentos

---

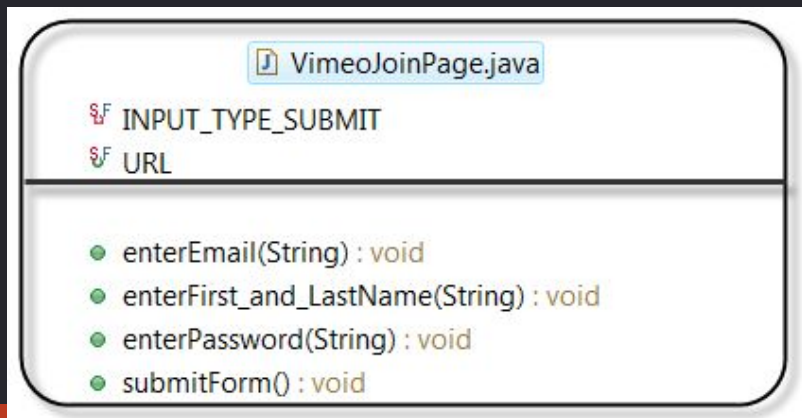
- Cada método tem como declaração de retorno a própria classe;
- Cada método retorna uma nova instância do driver;
- Todas as validações (asserts) devem ser feitas na classe de teste e não na classe da page;
- Para as validações o ideal é criar métodos que procurem por texto ou algum ponto de controle.

# Classes Page



# Classe Page

Exemplo de uma classe Page, cada função desta classe representa uma ação realizada pelo usuário na página.  
Por padrão tem o sufixo Page definido no nome da classe.





# Navegando Page a Page

Realizar uma operação em uma página que direciona para outra página.

Por exemplo: Clicar em um link ou enviar um formulário.

É responsabilidade desse objeto página retornar o próximo objeto página.

```
public class VimeoJoinPage

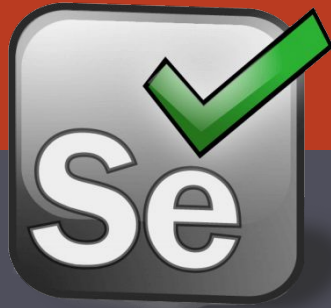
    public VimeoSignupPage submitForm(){
        driver.findElement(By.cssSelector(SUBMIT)).click();
        return new VimeoSignupPage(driver);
    }

public class VimeoSignupPage

    public VimeoProfilePage clickProfileLink(){
        //Clicking the profile link ("Me")
        driver.findElement(By.xpath("//*[@id='menu']/a")).click();

        return new VimeoProfilePage(driver);
    }
```

# Classes Test



# Classe Test

Classes que representam o fluxo de negócio por qual o teste irá passar para validar o resultado esperado.

Por padrão todas classes deste tipo tem um sufixo Test no nome.

```
@Test
public void testSignupSuccess() {
    VimeoJoinPage joinPage = new VimeoJoinPage(driver);
    //go to page
    joinPage.open();

    //enter data
    joinPage.enterFirst_and_LastName(first_and_LastName);
    joinPage.enterEmail(email);
    joinPage.enterPassword(password);

    //submit form
    VimeoSignupPage signupPage = joinPage.submitForm();

    //Verify signup success
    assertTrue("Sign up successful" , signupPage.isPageLoad());
}
```

# Exemplo de uma classe Test

---

Exemplo de uma classe de teste que realiza um login válido

```
@Test
public void testLogin(){
    HomePage home = new HomePage();
    CustomerPage customer = home.login("janipav", "extrasecretpassword");
    Assert.assertEquals(customer.getLoggedInUser(), "Pavel Janicek");
}
```

# Exemplos de classes Page

## Classes Page que implementam o login

```
public class HomePage{
    private WebDriver driver;
    private WebElement loginInput;
    private WebElement passwordInput;
    private WebElement loginSubmit;

    public WebDriver getDriver(){
        return driver;
    }

    public HomePage(){
        driver = new FirefoxDriver();
    }

    public CustomerPage login(String username, String password){
        driver.get("http://the-test-page.com");
        loginInput = driver.findElement(By.id("username"));
        loginInput.sendKeys(username);
        passwordInput = driver.findElement(By.id("password"));
        passwordInput.sendKeys(password);
        loginSubmit = driver.findElement(By.id("login"));
        loginSubmit.click();
        return new CustomerPage(this);
    }
}
```

```
public class CustomerPage{
    private HomePage homePage;
    private WebElement loggedInUserSpan;

    public CustomerPage(HomePage hp){
        this.homePage = hp;
    }

    public String getLoggedInUser(){
        loggedInUserSpan = homePage.getDriver().findElement(By.id("usrLongName"));
        return loggedInUserSpan.getText();
    }
}
```

# Hands on

