# Plano de Projeto OrangeHRM

Professora

Roberta de Souza Coelho

Equipe:

Candinho Luiz Dalla Brida Junior

Pedro Paulo Paiva de Medeiros

Victor Simplício Brandão de Lima

# Projeto OrangeHRM

link do site demo:

https://opensource-demo.orangehrmlive.com/

link do projeto no github:

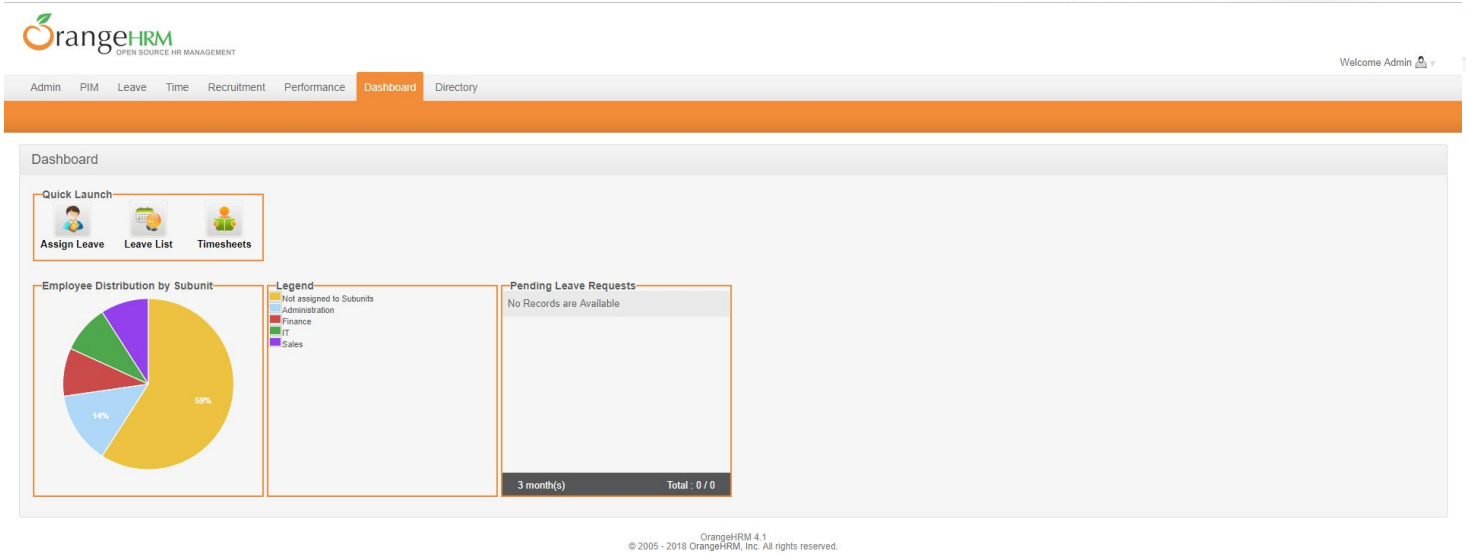
https://github.com/candinhojr/projeto-OrangeHRM

# Descrição breve do projeto a ser testado

O OrangeHRM é um sistema para Gestão de Recursos Humanos disponível como uma plataforma open-source.

Dividido em três releases (open-source, professional e enterprise), o sistema conta com inúmeros serviços que auxiliam o trabalho do RH de uma empresa oferecendo diversas funcionalidades tais como gerenciamento de funcionários, controle de horários de trabalho, recrutamento, entre outros.

# Algumas imagens do sistema...

# Escopo de testes

# Lista de funcionalidades a serem testadas

- Login
  - Verificar o login em diferentes casos (sucesso, insucesso, username correto, incorreto, password correto, incorreto)
- Formulários de cadastro de usuários
  - Validar os casos de sucesso e tentar o cadastro com dados inválidos
- Cadastrar Vaga de Emprego
  - Cadastrar um candidato para a vaga, conferir se os dados batem
  - Aceitar e marcar entrevista (verificar se o status altera em cada procedimento)

# Fluxo de Testes

- **Cadastrar** uma **vaga** X de **emprego**;
- **Inscrever** um **candidato** Y para a vaga X;
- **Aceitar** a **inscrição** do candidato Y e **marcar entrevista**;
- **Passar** o candidato Y na entrevista e **oferecer** a vaga de emprego;
- **Contratar** o candidato Y;
- **Criar** o **user** para o funcionário Y logar na plataforma;
- **Logar** com Y;

# Critérios de teste

# Critérios de Teste Utilizados

- Análise do valor limite
- Divisão em classes de equivalência

# Casos de Teste Realizados

# Casos de Teste para Cadastrar Vaga de Emprego

- ➔ CreateVacancyTest();

# Casos de Teste para Cadastrar Candidato na Vaga

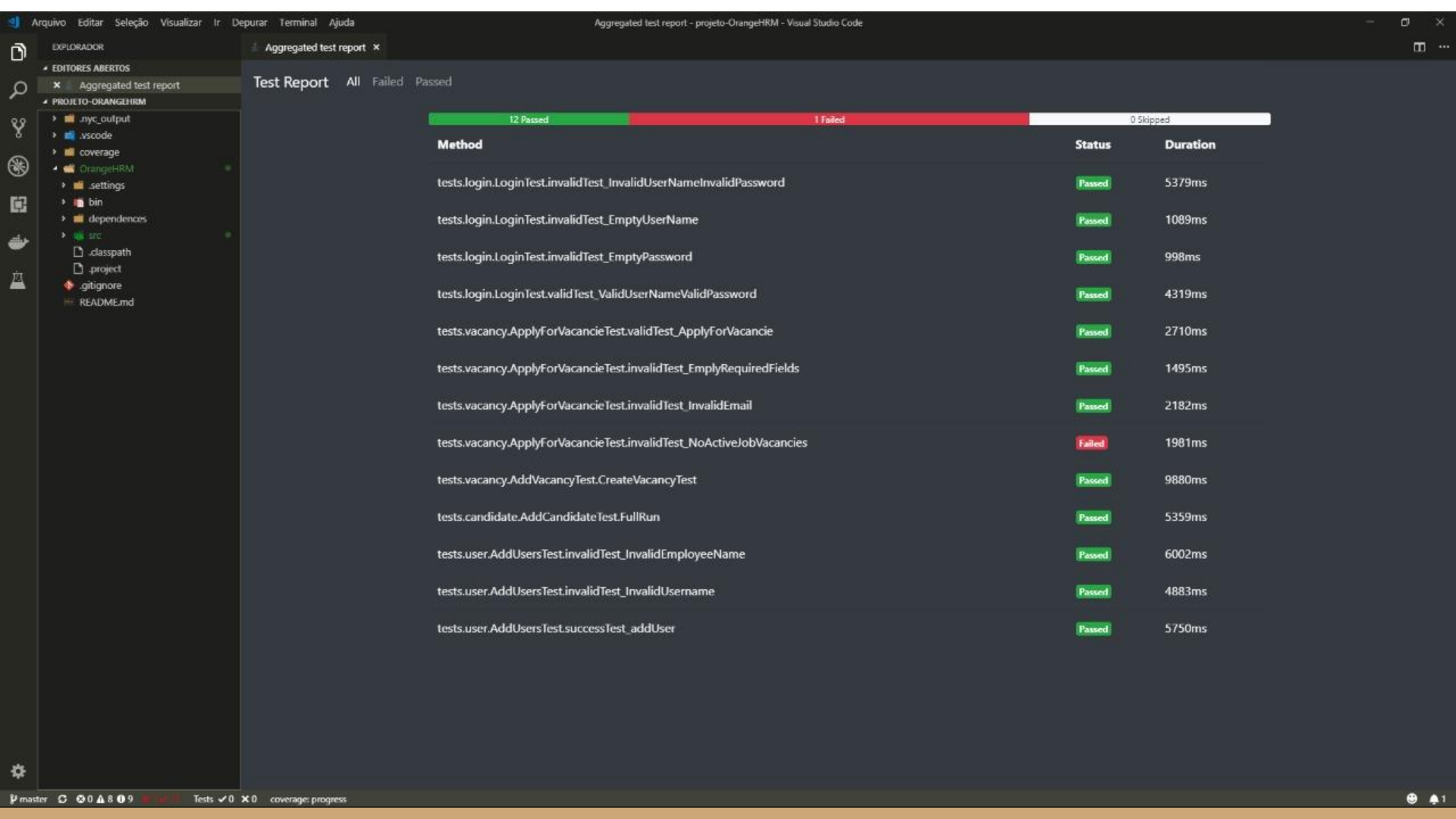
- ➔ validTest_ApplyForVacancieWithAllFieldsCompleted();
- ➔ validTest_ApplyForVacancieWithOnlyRequiredFieldsCompleted();
- ➔ invalidTest_EmplyRequiredFields();
- ➔ invalidTest_InvalidEmail();
- ➔ validTest_Document1M();
- ➔ validTest_InvalidDocument();
- ➔ invalidTest_DocumentGreaterThan1M();
- ➔ invalidTest_NoActiveJobVacancies();

# Casos de Teste para Criar User para Empregado

- ➔ invalidTest_InvalidEmployeeName();
- ➔ invalidTest_InvalidUsername();
- ➔ successTest_addUser();

# Casos de Teste para Logar com Usuário

- ➔ validTest_ValidUserNameValidPassword();
- ➔ invalidTest_InvalidUserNameInvalidPassword();
- ➔ invalidTest_EmptyUserName();
- ➔ invalidTest_EmptyPassword();


Arquivo Editar Seleção Visualizar Ir Depurar Terminal Ajuda
Aggregated test report - projeto-OrangeHRM - Visual Studio Code
EXPLORADOR
EDITORES ABERTOS
Aggregated test report
PROJETO-ORANGEHRM
.nyc_output
.vscode
coverage
OrangeHRM
.settings
bin
dependences
src
.classpath
.project
.gitignore
README.md
Aggregated test report
Test Report All Failed Passed
12 Passed
1 Failed
0 Skipped
Method
Status
Duration
tests.login.LoginTest.invalidTest_InvalidUserNameInvalidPassword Passed 5379ms
tests.login.LoginTest.invalidTest_EmptyUserName Passed 1089ms
tests.login.LoginTest.invalidTest_EmptyPassword Passed 998ms
tests.login.LoginTest.validTest_ValidUserNameValidPassword Passed 4319ms
tests.vacancy.ApplyForVacancieTest.validTest_ApplyForVacancie Passed 2710ms
tests.vacancy.ApplyForVacancieTest.invalidTest_EmplyRequiredFields Passed 1495ms
tests.vacancy.ApplyForVacancieTest.invalidTest_InvalidEmail Passed 2182ms
tests.vacancy.ApplyForVacancieTest.invalidTest_NoActiveJobVacancies Failed 1981ms
tests.vacancy.AddVacancyTest.CreateVacancyTest Passed 9880ms
tests.candidate.AddCandidateTest.FullRun Passed 5359ms
tests.user.AddUsersTest.invalidTest_InvalidEmployeeName Passed 6002ms
tests.user.AddUsersTest.invalidTest_InvalidUsername Passed 4883ms
tests.user.AddUsersTest.successTest_addUser Passed 5750ms
master
Tests ✓0 ✕0
coverage: progress

# Exemplos de Caso de Teste utilizando Page Objetc

```java
public class BasePage {

    public WebDriver driver;
    public WebDriverWait wait;

    private final String BASE_PAGE_URL = "https://opensource-demo.orangehrmlive.com/";

    public BasePage() {
        this.driver = Driver.getInstance().getWebDriver();
        this.wait = Driver.getInstance().getWebDriverWait();
    }

    // Construtor
    public BasePage(WebDriver driver, WebDriverWait wait) {
        this.driver = driver;
        this.wait = wait;
    }

    // Método click
    public void click(WebElement elementLocation) {
        this.waitTime(1000);
        elementLocation.click();
    }

    // Método writeText
    public void writeText(WebElement elementLocation, String text) {
        this.waitTime(1000);
        // driver.findElement(elementLocation).sendKeys(text);
        elementLocation.sendKeys(text);
    }

    // Método readText
    public String readText(WebElement elementLocation) {
        this.waitTime(1000);
        return elementLocation.getText();
    }
```

# BasePage

```java
// Método findElement
public WebElement findElement(WebElement elementLocation) {
    this.waitTime(1000);
    return elementLocation;
}

// Método selectElement pra seleção de dropdowns
public Select selectElement(WebElement elementLocation) {
    this.waitTime(1000);
    return new Select(elementLocation);
}

public LoginPage goToLoginPage() {
    driver.get(this.BASE_PAGE_URL);

    return PageFactory.initElements(driver, LoginPage.class);
}

public HomePage logOut() {

    MainMenuPage mainMenuPage = PageFactory.initElements(driver, MainMenuPage.class);

    this.waitTime(1000);

    mainMenuPage.logOut();

    return PageFactory.initElements(driver, HomePage.class);
}

private void waitTime(long time) {
    driver.manage().timeouts().implicitlyWait(time, TimeUnit.MILLISECONDS);
}
```

# BasePage

# AddUserPage

```java
public class AddUserPage extends BasePage {

  @FindBy(id = "systemUser_userType")
  private WebElement userRoleDropdown;

  @FindBy(id = "systemUser_employeeName_empName")
  private WebElement employeeInputId;

  @FindBy(id = "systemUser_userName")
  private WebElement usernameInputId;

  @FindBy(id = "systemUser_status")
  private WebElement statusDropdownId;

  @FindBy(id = "systemUser_password")
  private WebElement passwordInputId;

  @FindBy(id = "systemUser_confirmPassword")
  private WebElement confirmPasswordInputId;

  @FindBy(id = "btnSave")
  private WebElement saveButtonId;

  @FindBy(xpath = "//form[@id='frmSystemUser']/fieldset/ol/li[2]/span")
  private WebElement employeeNameMessage;
  @FindBy(xpath = "//form[@id='frmSystemUser']/fieldset/ol/li[3]/span")
  private WebElement usernameMessage;

  public AddUserPage() {
    super();
  }

  public AddUserPage(WebDriver driver, WebDriverWait wait) {
    super(driver, wait);
  }
```

# AddUserPage

```
public void fillEmployeeName(String employeeName) {
  writeText(this.employeeInputId, employeeName);
}

public void fillUsername(String username) {
  writeText(this.usernameInputId, username);
}

public void fillPassword(String password) {
  writeText(this.passwordInputId, password);
}

public void fillConfirmPassword(String confirmPassword) {
  writeText(this.confirmPasswordInputId, confirmPassword);
}

public void selectUserRole(String userRole) {
  selectElement(this.userRoleDropdown).selectByVisibleText(userRole);
}

public void selectStatus(String status) {
  selectElement(this.statusDropdownId).selectByVisibleText(status);
}

public void clickSave() {
  click(this.saveButtonId);
}

public boolean verifyEmployeeInvalid() {
    return readText(this.employeeNameMessage).equalsIgnoreCase("invalid");
}
```

# BaseTest

```java
public class BaseTest {
    protected WebDriver driver;
    protected WebDriverWait wait;

    protected HomePage home;

    public BaseTest() {
        this.driver = Driver.getInstance().getWebDriver();
        this.wait = Driver.getInstance().getWebDriverWait();

        this.home = PageFactory.initElements(driver, HomePage.class);
    }

    // Verificar o uso do BeforeClass
    @Before
    public void setUp() throws Exception {
        this.driver = Driver.getInstance().getWebDriver();
        this.wait = Driver.getInstance().getWebDriverWait();
    }
```

```java
public class AddUsersTest extends BaseTest {

  private AddUserPage addUserPage;

  @Before
  public void setUp() throws Exception {
    super.setUp();

    this.addUserPage = PageFactory.initElements(driver, AddUserPage.class);

    this.home.goToLoginPage().loginToOrangeHRM("Admin", "admin123").goToViewSystemUsers().goToAddUserPage();
  }

  @After
  public void after() throws Exception {
    this.home.logOut();
  }

  @Test
  public void invalidTest_InvalidEmployeeName() {
    this.addUserPage.fillEmployeeName("adwad");

    this.addUserPage.clickSave();

     assertTrue(this.addUserPage.verifyEmployeeInvalid());
  }

  @Test
  public void invalidTest_InvalidUsername() {
    this.addUserPage.fillUsername("awd");

    this.addUserPage.clickSave();

     assertTrue(this.addUserPage.verifyUsernameInvalid());
  }
```

# Bugs Encontrados

# Bugs Encontrados

- ➔ É possível um funcionário ter mais de um cadastro de usuário;
- ➔ É possível cadastrar o mesmo candidato para a mesma vaga;
- ➔ Não aceita documentos com exatamente 1M nos formulários, quando não deveria aceitar documentos com mais de 1M;

# Dificuldades e Lições Aprendidas

# Dificuldades e Lições Aprendidas

- Sem acesso a documentação do Sistema;
- Sistema remove dados cadastrados periodicamente;
- Organização;
- Sistema instável:
  - Dados cadastrados no sistema alterados de forma imprevisível
  - e.g. Cadastros de usuários apagados (incluindo os empregados default)
- Listas e Alertas;