



**INSTITUTO FEDERAL
SANTA CATARINA**

Ministério da Educação
Secretaria de Educação Profissional e Tecnológica
INSTITUTO FEDERAL DE SANTA CATARINA

ENGENHARIA DE SOFTWARE II

Plano de Teste

Profª. Milena
Nome: Matheus Santin Nazário

Data: 22/06/2023

Projeto Integrador 2023 - Software Lança/Consulta Notas

Checklist de Qualidade Software

| <i>Item avaliado</i> | <i>Objetivo de QA</i> | <i>Priorização</i> | <i>Avaliação: Aprovado (A),Reprovado (R), Não se aplica (N/A)</i> | <i>Item Avaliado</i> |
|---|-----------------------|--------------------|---|----------------------|
| Verificar se títulos e subtítulos estão de acordo com cada página | Usabilidade | Desejável | | |
| Verificar se os botões estão de acordo com cada ação de sua descrição | Usabilidade | Desejável | | |
| Verificar se os campos deixam digitar informações diferentes das permitidas por exemplo String não pode ser digitada em uma Int; Testar caracteres | Usabilidade | Desejável | | |
| Verificar se a apresentação de campos obrigatórios estão correta (‘*’ antes dos ‘:’);e se os dados não são gravados sem que todos os campos obrigatórios estejam preenchidos; | Usabilidade | Importante | | |
| Verificar se os processamentos associados aos botões estão ocorrendo corretamente; | Funcionalidade | Essencial | | |
| Verificar se a geração de código está funcionando corretamente; | Funcionalidade | Essencial | | |
| Verificar se a inclusão de notas e presença estão funcionando corretamente; | Funcionalidade | Essencial | | |
| Verificar se os campos existentes estão associados a outras telas e testar a consulta de notas; | Funcionalidade | Essencial | | |

Plano de Teste:

1. Teste da interface do usuário:

Verificar se a janela de início é exibida corretamente.

Testar se os botões "Professor" ,"Aluno" , “Voltar” , “Lançar notas” e “Consultar” são criados e exibidos corretamente na janela.

Clicar nos botões e verificar se as respectivas janelas (Professor e Aluno) vão ser abertas corretamente.

2. Teste das funcionalidades dos botões:

No botão "Professor", verificar se a janela do Professor é exibida corretamente.

No botão "Aluno", verificar se a janela do Aluno é exibida corretamente.

No botão "Lançar(Gravar)", verificar se os dados digitados vão ser gravados no banco de dados.

No botão "Consultar", verificar se o aluno vai conseguir consultar suas notas.

No botão “Voltar”, a página vai voltar ao início da tela.

3. Teste de usabilidade:

Avaliar a facilidade de uso da interface do usuário.

Testar se os botões estão posicionados corretamente na janela.

Verificar se as janelas do Professor e Aluno são intuitivas e fáceis de usar.

Verificar se os títulos e subtítulos estão corretos com cada página.

4. Teste de cobertura:

Verificar se todas as linhas de código foram testadas.

Garantir que todas as ações possíveis na interface do usuário foram testadas.

Plano de Teste:

1. Teste entre as janelas:

- Descrição: Verificar se é possível abrir e fechar corretamente as janelas do "Professor" e "Aluno".
- Clicar no botão "Professor" para abrir a janela do Professor e, em seguida, fechá-la ou voltar ao clicar no botão "Voltar". Em seguida, clique no botão "Aluno" para abrir a janela do Aluno e, em seguida, fechá-la ou voltar ao clicar no botão "Voltar".
- Na janela Professor ao clicar no botão "Lançar(Gravar)" deve ser lançado os dados inseridos no banco de dados ou voltar ao clicar no botão "Voltar".
- Na janela Aluno ao clicar no botão "Consultar" deve ser gerado um ArrayList, com a matrícula informada pelo aluno no campo Matrícula ou voltar ao clicar no botão "Voltar".
- Resultado esperado: As janelas do Professor e Aluno são abertas corretamente e podem ser fechadas sem erros.

2. Teste de usabilidade:

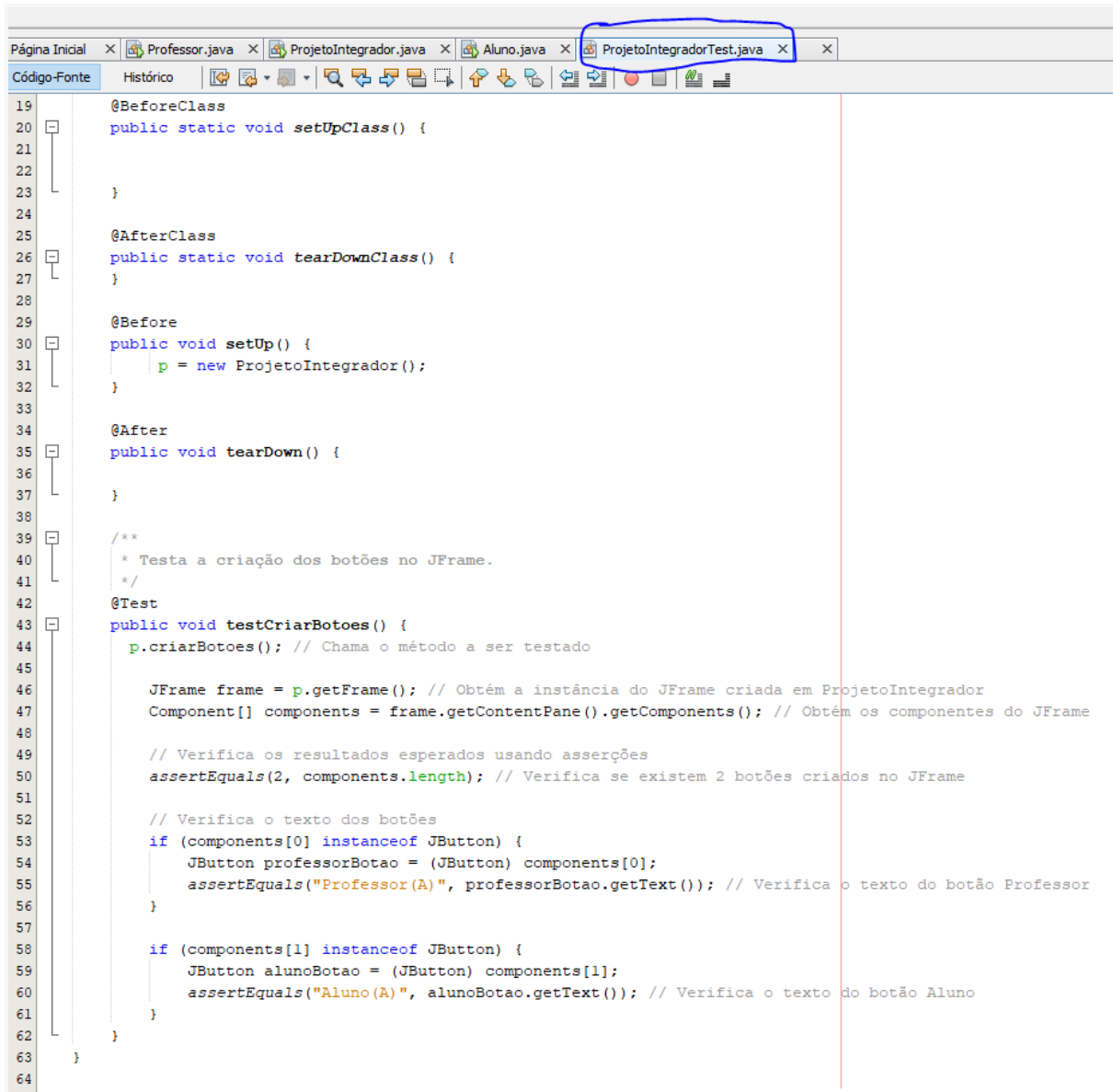
- Observar a disposição e descrição dos botões e interagir com as janelas do Professor e Aluno.
- Resultado esperado: A interface do usuário deve ser intuitiva, os botões estão posicionados corretamente e as janelas são fáceis de usar.

3. Teste de Usabilidade:

- Verificar se a funcionalidade de consulta de dados dos alunos funciona corretamente.
- Digitar uma matrícula que tenha sido cadastrada; Caso uma que não tenha sido deve aparecer código de erro.
- Clicar no botão "Consultar"
- Resultado esperado deve ser um campo abaixo onde vai ser mostrado as informações inseridas pelo usuário "Professor".

Teste Unitário com printScreen:

Aqui abaixo tem o código onde criei uma nova classe chamada ProjetoIntegradorTest



```
19 @BeforeClass
20 public static void setUpClass() {
21
22
23 }
24
25 @AfterClass
26 public static void tearDownClass() {
27
28 }
29
30 @Before
31 public void setUp() {
32     p = new ProjetoIntegrador();
33 }
34
35 @After
36 public void tearDown() {
37
38 }
39
40 /**
41  * Testa a criação dos botões no JFrame.
42  */
43 @Test
44 public void testCriarBotoes() {
45     p.criarBotoes(); // Chama o método a ser testado
46
47     JFrame frame = p.getFrame(); // Obtém a instância do JFrame criada em ProjetoIntegrador
48     Component[] components = frame.getContentPane().getComponents(); // Obtém os componentes do JFrame
49
50     // Verifica os resultados esperados usando asserções
51     assertEquals(2, components.length); // Verifica se existem 2 botões criados no JFrame
52
53     // Verifica o texto dos botões
54     if (components[0] instanceof JButton) {
55         JButton professorBotao = (JButton) components[0];
56         assertEquals("Professor(A)", professorBotao.getText()); // Verifica o texto do botão Professor
57     }
58
59     if (components[1] instanceof JButton) {
60         JButton alunoBotao = (JButton) components[1];
61         assertEquals("Aluno(A)", alunoBotao.getText()); // Verifica o texto do botão Aluno
62     }
63 }
64 }
```

Ao clicar com botão direito e em Executar arquivo ele gera o teste unitário:

The screenshot displays an IDE interface with a Java test file and its execution results. The top pane shows the source code of `ProjetoIntegradorTest`, which includes assertions for the number of components and the text of two buttons, "Professor (A)" and "Aluno (A)". The bottom pane shows the test results, indicating that the test failed due to a `NullPointerException` at line 50 of `ProjetoIntegrador.criarBotoes`.

```
50 // Verifica os resultados esperados usando asserts
51 assertEquals(2, components.length); // Verifica se existem 2 botões criados no JFrame
52
53 // Verifica o texto dos botões
54 if (components[0] instanceof JButton) {
55     JButton professorBotao = (JButton) components[0];
56     assertEquals("Professor (A)", professorBotao.getText()); // Verifica o texto do botão Professor
57 }
58
59 if (components[1] instanceof JButton) {
60     JButton alunoBotao = (JButton) components[1];
61     assertEquals("Aluno (A)", alunoBotao.getText()); // Verifica o texto do botão Aluno
62 }
63 }
```

ProjetoIntegradorTest

Resultados do Teste X

ProjetoIntegradorTest X

Tests passed: 0,00 %

Nenhum teste aprovado(s), 1 teste causou (causaram) erro. (0,171 s)

ProjetoIntegradorTest **Falhou**

- testCriarBotoes **causou um ERRO: java.lang.NullPointerException**
 - java.lang.NullPointerException
 - at projetointegrador.ProjetoIntegrador.criarBotoes(ProjetoIntegrador.java:50)
 - at ProjetoIntegradorTest.testCriarBotoes(ProjetoIntegradorTest.java:44)

projetoIntegrador (run) #3 | running...