ROTEIRO DE TESTE PARA PROGRAMA

1. **Testes de menu.**

Não há pré-condições.

* + Fazer a navegação nos menus para ver sua fluidez.

Expectativa: Os menus funcionem de acordo com as opções inseridas.

1. **Teste de usabilidade.**

Não há pré-condições.

* + Pedir para uma pessoa que não conheça o sistema fazer uso dele em tarefas simples e demandar um feedback sobre a facilidade de uso e clareza as mensagens.

Expectativa: Coletar informações sobre a facilidade de uso do sistema, como posição dos menus e interação com mensagens recebidas do sistema.

1. **Teste de Inserção de Aluno com Dados Inválidos**

Não há pré-condições

* + **Entrada 1**: Entrada vazia
    1. **Na classe Student** deixar os campos ID, nome, idade, e-mail vazios.
    2. **Na classe Couse** deixar os campos Code, nome, descrição, valor vazios
  + **Entrada 2**: Entrada inválida
    1. **Estudante**:
       1. ID (char == 3 / não numérico / código existente)
       2. nome (char<3 && > 70 / contendo números)
          1. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin vitae justo nec nisl ultricies vehicula ut id metus
       3. idade (intervalo entre 4 e 120 / não numérico)
       4. e-mail (char < 8 && > 80 / fora de padrão [\*\*\*@\*\*.\*\*](mailto:***@**.**) ).
    2. **Curso**:
       1. CODE (char == 4 / não numérico)
       2. nome (char < 8 && char > 40 /contendo números)
       3. descrição (char < 30 && char > 300 / somente números)
          1. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur vel sem vitae velit fermentum sollicitudin at id ligula. Donec laoreet, lacus a ultrices vulputate, justo justo feugiat lorem, at feugiat ex magna ut mi. Cras suscipit, risus ut gravida tempus, augue lectus auctor lacus, id dignissim ligula arcu nec orci. Praesent blandit accumsan ligula, sed mollis neque viverra quis. Integer id massa nec tortor ullamcorper posuere. Sed at nunc ultricies, eleifend libero non, convallis orci. Suspendisse potenti
       4. preço (10,00 < valor > 1500,00 / não numérico)

**Expectativa**: O programa deve apresentar mensagens de erro apropriadas e solicitar a correção da entrada.

1. **Teste: Cadastro com Dados Válidos**

Pré-condições: Nenhum aluno ou curso cadastrado.

* + **Entradas alunos:**
    1. Legenda: ID, Name, Age; e-mail
    2. **Entrada 1**: 123; João Pedro; 15; [joao@gmail.com](mailto:joao@gmail.com)
    3. **Entrada 2:** 456; Maria Socorro; 26; [socorro@hotmail.com](mailto:socorro@hotmail.com)
    4. **Entrada 3:** 789; Antônio José; 55; [antonio@yhaoo.com.br](mailto:antonio@yhaoo.com.br)
  + **Entradas Cursos**
    1. Legenda: CODE; Name; description; Price.
    2. **Entrada 1:** 1234; Logica de Programação; Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque est tortor, molestie in odio at, venenatis posuere nisi. Fusce sodales condimentum vehicula. Ut quis ex id ante rhoncus tempus. Donec.; 156,25
    3. **Entrada 2:** 5678; Programação Orientada a Objetos; Vestibulum justo turpis, sagittis eget neque sit amet, fringilla lacinia dolor. Integer vitae sem risus. Fusce eget urna feugiat justo volutpat accumsan id sit amet sapien. Duis ante justo, ultrices.; 259,32
    4. **Entrada 3:** 9101; Curso C# Completo; 3.; 500,00

**Expectativa:** O programa não deve apresentar qualquer erro de entrada.

1. **Teste de Matrícula com Dados válidos.**
   * Legenda: ID do aluno => CODE do curso.
   * **Entrada 1**: 123 => 1234.
   * **Entrada 2**: 456 => 5678.
   * **Entrada 3**: 123 => 5678.
   * **Entrada 4**: 789 => 9101.

**Expectativa:** O programa não deve apresentar qualquer erro de entrada.

1. **Teste de Matrícula com Dados Inválidos**

Pré-condições: Somente os alunos, os cursos e as matrículas mencionados acima estejam inseridos.

* + Legenda: ID do aluno => CODE do curso; erro esperado.
  + **Entrada 1**: 999 => 1234; Matrícula de aluno não existente no sistema.
  + **Entrada 2**: 123 => 9999; Matrícula em curso inexistente.
  + **Entrada 3**: 123 => 1234; Matrícula duplicada (o mesmo aluno tentando se matricular duas vezes no mesmo curso).

**Expectativa**: O programa deve rejeitar as entradas com mensagens apropriadas.

1. T**este de Exclusão de Aluno ou Curso Não Existente (ainda em desenvolvimento)**

Pré-condições: Somente os alunos, os cursos e as matrículas mencionados acima estejam inseridos.

* + **Entrada 1:** ID de aluno 888; Tentativa de remover aluno com ID que não existe.
  + **Entrada 2:** CODE de curso 8888; Tentativa de remover curso que não foi cadastrado.

**Expectativa**: O sistema deve avisar que o aluno ou curso não foi encontrado.

1. **Teste de exclusão de aluno e curso existente**
   * Entrada 1: ID de aluno 123.
   * Entrada 2: CODE de curso 1234.

Expectativa: O sistema deve fazer as exclusões corretamente.

1. Teste de desmatricula de aluno
   * Entrada 1: 888 => 1234; Tentativa de desmatricular aluno inexistente.
   * Entrada 2: 123 => 8888; Tentativa de desmatricular aluno existente de curso não existente.
   * Entrada 3: 456 => 1234; Tentativa de desmatricular aluno existente de curso existente, mas que não esteja matriculado ele.

Expectativa: O sistema deve rejeitar as entradas apresentando as devidas mensagens de erro.

1. **Teste de Relatório (falta concluir o código)**
   * Testar relatórios vazios (sem alunos ou cursos cadastrados).
   * Testar relatórios com grande número de registros.
   * Verificar a formatação do relatório (está legível, fácil de interpretar?)