

ENGENHARIA DE SOFTWARE - 3º PERÍODO

FINAL HAND-ON

Tema:

Validação de Dados e Autenticação de Usuários em Aplicações Desktop Distribuídas ou Web

ATENÇÃO: Trata-se da parte prática ref. à disciplina de POO II, onde o restante da nota será atribuída em avaliação objetiva e dissertativa sobre os tópicos da disciplina que envolvem o projeto.

Descrição do Projeto:

O aluno deverá desenvolver uma **aplicação desktop distribuída ou web completa**, contemplando tanto a interface (view) ou front-end quanto o back-end, com foco na **implementação de validações de dados, tratamento de exceções e controle de acesso por sessão**.

O sistema deverá permitir o **cadastro de usuários do sistema** em um **banco de dados relacional**, validando os seguintes campos obrigatórios:

- **Nome completo** (somente letras e espaços);
- **Idade** (somente maiores de 18 anos);
- **CPF** (formato e dígitos válidos);
- **Telefone** (com DDD e formato correto);
- **Endereço** (campos obrigatórios: rua, número, cidade e estado);
- **E-mail** (formato válido e não duplicado no banco de dados);
- **Senha** (com critérios mínimos de segurança, como tamanho e complexidade).

As **validações devem ser implementadas de forma combinada**, observando:

1. GUI (View) / Front-end:

- Uso de **estruturas de decisão (if/else, switch)**;
- Uso de **expressões regulares (regex)** para verificar formatos de entrada;
- Exibição de **mensagens de erro e sucesso** diretamente na interface.

2. Back-end:

- Revalidação dos dados recebidos;
- Uso de **tratamento de exceções (try/catch)** para lidar com erros;
- **Expressões regulares e estruturas condicionais** para reforçar a consistência dos dados;
- **Validação de credenciais e gestão de sessões** com controle de acesso restrito.

O aluno deverá ainda **escolher um contexto de aplicação comercial** (por exemplo: loja virtual, sistema de agendamento de serviços, controle de alunos, pedidos on-line, etc.) e adaptar a interface e funcionalidades ao cenário proposto.

Requisitos Técnicos Mínimos:

- **GUI (View) / Front-end:** Biblioteca AWT/Swing ou JavaFX se for implementado como Sistema Desktop Distribuído. Em caso de aplicação Web HTML5, CSS3 e JavaScript (ou framework moderno, como React ou Vue).
- **Back-end:** Linguagem à escolha (Node.js, PHP, Python, Java, etc.).
- **Banco de dados relacional:** MySQL, PostgreSQL, MariaDB ou similar.
- **Validações obrigatórias:**
 - Estruturas de decisão;
 - Expressões regulares;
 - Tratamento de exceções;
 - Controle de sessão e autenticação segura.

- **Mensagens claras** de validação na GUI (View) ou front-end e no back-end.
 - Testes das validações utilizando ao JUnit.
-

Entrega:

O projeto deve incluir:

1. **Código-fonte completo e comentado no Github.**
 2. **Relatório técnico** contendo:
 - Contexto comercial escolhido;
 - Descrição detalhada das validações implementadas, com exemplos de uso de estruturas de decisão, regex e exceções;
 - Estrutura do banco de dados (modelo lógico ou DER);
 - Descrição do fluxo de autenticação e controle de sessões;
 - Prints das principais telas do sistema;
 - Considerações sobre boas práticas e segurança.
-

Critérios de Avaliação (Total: 2000 pontos)

Critério	Descrição
1. Estrutura do Projeto Organização do Código	do Organização das pastas, nomenclatura de arquivos, clareza e legibilidade do código, uso de boas práticas e comentários.
2. Validações Front-end	no Implementação de validações com estruturas de decisão e expressões regulares, mensagens de erro e sucesso, prevenção de envio de dados inválidos.
3. Validações Back-end	no Revalidação dos dados recebidos, uso de estruturas condicionais e expressões regulares, tratamento de exceções (try/catch), proteção contra falhas de entrada.

- | | |
|--|--|
| 4. Autenticação e Controle de Sessões | Implementação de login, logout, restrição de acesso a páginas protegidas, persistência segura de sessão, e redirecionamentos corretos. |
| 5. Integração com Banco de Dados | Modelagem e implementação do banco de dados, criação das tabelas, inserção e recuperação de dados, verificação de unicidade e integridade. |
| 6. Interface e Usabilidade | Qualidade do layout, organização dos formulários, responsividade, clareza visual e feedback ao usuário durante interações. |
| 7. Relatório Técnico | Clareza, coerência, profundidade da explicação técnica, qualidade da redação, inclusão de prints e descrição das validações realizadas. |
| 8. Originalidade e Contexto Comercial | Adequação do sistema ao contexto escolhido, criatividade na proposta, coerência entre o problema e a solução desenvolvida. |

Objetivo de Aprendizagem:

Com este projeto, o aluno deverá demonstrar:

- Capacidade de **validar dados na GUI (View) ou front-end e no back-end** utilizando **estruturas de decisão, expressões regulares e tratamento de exceções**;
- Conhecimento sobre **autenticação, segurança e controle de acesso**;
- Habilidade para **integrar interface, lógica de negócio e banco de dados**;
- Competência para **documentar e justificar tecnicamente** suas implementações;
- Criatividade e senso prático na **adaptação de um contexto comercial real**.