

## **Projeto Final:**

### **Desenvolvimento de Interface Gráfica para o Sistema Persistente**

#### **Objetivo:**

Desenvolver um projeto Java que utilize JDBC e JPA para gerenciar o acesso a um banco de dados PostgreSQL. O projeto deve ser gerenciado com Maven, versionado no Github e conter pelo menos três classes relacionadas com mapeamento de relacionamento.

Além disso, o projeto deverá envolver interface gráfica (GUI) em Java, utilizando conceitos de programação orientada a objetos e integração com o banco de dados PostgreSQL (usando JPA). A interface deve permitir o gerenciamento das entidades, como visualização, cadastro, edição e exclusão de dados.

#### **Temática do trabalho:**

Escolha um tema livre para o seu sistema (ex: sistema de reservas, controle acadêmico, pet shop, gestão de biblioteca, etc.). O tema deve, obrigatoriamente, ter sido informado e validado no Classroom, pois deverá ser um tema único na turma.

#### **Instruções:**

##### **1. Configuração do Projeto:**

- Criar um repositório do projeto, ProjetoFinal\_LPOO\_[NomeAluno], para armazenar o código da GUI.
- Certifique-se de que todas as alterações são versionadas no GitHub. (presenças serão avaliadas de acordo com os commits realizados).
- Configure um novo projeto Maven.

##### **2. Banco de Dados:**

- Crie um banco de dados no PostgreSQL chamado ProjetoFinal\_[NomeAluno].
- As tabelas necessárias conforme o mapeamento das entidades deverão ser criadas automaticamente de acordo com as anotações JPA.

## 2. Desenvolvimento da Interface Gráfica:

- Crie uma interface gráfica utilizando Java Swing
- A GUI deve permitir as seguintes operações para cada entidade:
  - Visualizar todos os registros (e adição de filtros)
  - Adicionar novos registros.
  - Editar registros existentes.
  - Excluir registros.

Validações diferentes dessa proposta deverão ser validadas com a professora previamente, conforme cronograma de entregas semanais.

## 3. Funcionalidades Mínimas:

- Cada entidade deve ter uma tela dedicada ou seções específicas na GUI para suas operações CRUD (Create, Read, Update, Delete)
- Implemente validações básicas nos campos de entrada, como verificação de campos obrigatórios e formato de dados.
- Menu de acesso aos cadastros, lógica de negócio e um sobre – Ajuda, com a descrição do projeto idealizado.

## 4. Mapeamento JPA (Integração com o Banco de Dados):

- Certifique-se de que as operações realizadas na interface gráfica atualizem corretamente o banco de dados.
- Configure o persistence.xml para garantir que as operações de banco de dados sejam executadas corretamente.

## 5. Github:

- Faça commits regulares para documentar o progresso do desenvolvimento.
- Readme.md:
  - O diagrama do projeto (imagem) deverá estar disponível no Readme.md do projeto, assim como a descrição geral e uma breve explicação da modelagem e como usar o sistema. Visando facilitar a utilização por meio de usuários.

## Entrega e Apresentação:

- Envie o link do repositório do Github
  - a modelagem utilizada deverá ser incluída no arquivo Readme.md do Github.
- Ver exemplo no projeto do Estacionamento do IFSUL incluído no Github pela professora:
  - [https://github.com/ProfVanessa-IFSUL/LPOO\\_2024-2\\_EstacionamentoIFSUL](https://github.com/ProfVanessa-IFSUL/LPOO_2024-2_EstacionamentoIFSUL)
- O projeto deverá ser enviado até o dia 29-06-2025, via Classroom.
- O projeto deverá ser apresentado até a aula do dia 30-06 (Presencialmente) ou remoto, conforme agendamento disponibilizado pela Professora. Certifique-se que o projeto está funcionando e com todos os arquivos acessíveis via Github.
  - Projetos não apresentados terão sua nota desconsideradas.

### **Critérios de Avaliação:**

- **Qualidade da Interface:**
  - Usabilidade e design da interface gráfica.
  - Navegação e organização das telas.
- **Funcionalidade:**
  - Correção e eficiência das operações CRUD.
  - Integração correta com o banco de dados.
- **Versionamento:**
  - Uso adequado do GitHub para controlar a versão do projeto.
- **Documentação:**
  - Clareza e completude do Readme.md no repositório do GitHub.

---

### **Cronograma das Aulas Finais**

<b>Data</b>	<b>Atividade prevista</b>	<b>Envio</b>	<b>Formato</b>
16/06/2025	Validação da Modelagem proposta, conforme temática definida e informada no Classroom.	Diagrama de Classes UML no formato PDF	EAD
23/06/2025	Após validação da modelagem, realizar a persistência no BD, configurando o projeto adequadamente e envio do projeto para o repositório criado.	Link do repositório com commit realizado até o dia 23 com pelo menos a modelagem e persistência dos dados, conforme validação da modelagem.	EAD
30/06/2025	Apresentação	Link do repositório com projeto finalizado e commit realizado até o dia 29.	Presencial
07/07/2025	Reavaliação	Teórica e Prática	Presencial