INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO SUL
Campus Bento Gonçalves

Curso Superior de Análise e Desenvolvimento de Sistemas
Desenvolvimento de Aplicações Corporativas
Prof. Dr. Maurício Covolan Rosito
PROVA P2

INSTRUÇÕES GERAIS

- Esta é uma **prova prática, individual e com consulta permitida**.
- A nota desta atividade corresponde a **50% da avaliação final da disciplina**.

ENTREGA DA ATIVIDADE

- O prazo máximo para entrega é **03/07/2025, até às 23h59**, por meio da plataforma **Moodle**.
- A entrega deve ser feita em **um arquivo compactado (.zip)** contendo todos os arquivos necessários para execução da aplicação.
- **Apenas envios realizados dentro do prazo serão considerados**. Submissões fora do prazo não serão aceitas, salvo justificativa oficial validada previamente.
- O envio deve ser feito exclusivamente pelo Moodle, no campo específico destinado à prova.

CRITÉRIOS DE AVALIAÇÃO

- Atendimento aos **requisitos funcionais** propostos;
- Demonstração de **compreensão dos conceitos** abordados ao longo da disciplina;
- **Pontualidade** na entrega;
- **Clareza e organização** da solução apresentada.

OBJETIVO:

Aprimorar a aplicação desenvolvida na Prova P1, incorporando práticas modernas de desenvolvimento corporativo, como padronização de código, controle de versão, arquitetura escalável, autenticação e uso de ORM com Sequelize.

REQUISITOS OBRIGATÓRIOS:

Veja abaixo os requisitos obrigatórios para a aplicação front-end (React):

- **Autenticação e Rota Protegida:**
  - Implementar formulário de login no front-end.
  - Armazenar o token JWT no localStorage e enviá-lo nas requisições protegidas via Authorization.
  - Deixar pelo menos uma rota pública;
  - Proteger ao menos uma rota (ex: /dashboard) com verificação de autenticação.
- **Aprimoramento de Componentes:**
  - Refatorar componentes para manter a padronização e reutilização.
  - Validar formulários com biblioteca ou lógica personalizada.

Veja abaixo os requisitos obrigatórios para a aplicação back-end (Node.js + Express.js)

- **Padronização de Código:**
  - Configurar e aplicar o **ESLint** com uma configuração adequada
  - Configurar e aplicar o **Prettier** para formatação automática.
  - Configurar scripts no package.json para rodar lint e prettier com comandos como **npm run lint** e **npm run format**.

- **Controle de Versão:**
  - Criar um repositório no GitHub, com histórico de commits organizados e significativos.
  - Incluir um README.md com descrição do projeto, instruções de instalação e execução.
- **Arquitetura da Aplicação:**
  - Refatorar a API utilizando Arquitetura em Camadas:
    - Camadas obrigatórias: routes, controllers, services, models, middlewares, database, utils (se necessário).
      - qualquer alteração necessária, consultar o professor da disciplina
    - Separação clara de responsabilidades entre as camadas.
- **Autenticação:**
  - Implementar login de usuário com autenticação JWT:
  - Endpoint de login com verificação de credenciais.
  - Geração de token JWT.
  - Middleware para proteger pelo menos uma rota de cada entidade.
  - Criar uma entidade User com email, password (hash com bcrypt) e role.
- **ORM Sequelize:**
  - Refatorar a aplicação para usar o Sequelize como ORM.
    - deve consultar o professor sobre esta funcionalidade
  - Criar models Sequelize para as entidades.
- **Migrations e Seeders:**
  - Criar **migrations** para as tabelas.
  - Criar pelo menos um **seeder** para popular uma tabela com dados fictícios.

APRESENTAÇÃO DA APLICAÇÃO:

- A **apresentação da aplicação deve ser realizada presencialmente em sala de aula**
  - **Datas de apresentação:** 03/07/2025 e 10/07/2025
  - **OBS:** A pedido do aluno, ele pode antecipar sua apresentação mediante acordo com o professor.
- Criar um repositório no **GitHub** e compartilhar com o professor (user: mauriciorosito) até **29/05/2025**.
- Além disso, é **obrigatório o envio de um vídeo explicativo** sobre o funcionamento da aplicação.
  - O vídeo deve ter **no máximo 10 minutos** de duração.
  - O **link para o vídeo** (no YouTube, Google Drive, etc.) deve ser incluído em um **arquivo de texto (.txt)** e enviado junto com os demais arquivos no .zip.
- O vídeo deve conter:
  - Demonstração prática da aplicação em funcionamento;
  - Explicação dos principais trechos do **código-fonte**;
  - Orientações de uso (interface, navegação, funcionalidades);
  - Indicação clara de **limitações ou funcionalidades não implementadas**, caso existam — isso é importante para alinhar expectativas e justificar eventuais lacunas.