

Professor: Fábio Kravetz

Leia atentamente as instruções:

- Um documento em PDF (**Deve constar o nome dos alunos em tal documento**) deverá ser entregue até o dia previsto com a explicação do(s) código(s) desenvolvido(s);
- **Haverá defesa do projeto por parte dos integrantes;**
- **O projeto poderá ser realizado em até 5 alunos (Equipe de Jornada), porém apenas 1 aluno deve realizar a entrega;**
- Todos os arquivos, inclusive o PDF, devem ser entregues em uma **pasta única**, tal pasta deve estar zipada;
- Trabalhos entregues com arquivos em formato word ou caso a pasta anexada não esteja zipada terão **ZERO** como nota;
- Pastas zipadas e vazias terão **ZERO** como nota;
- Padronize o nome do arquivo no seguinte formato: BE_aluno1_aluno2_aluno3_aluno4_aluno5;
- Apresentar todos os códigos desenvolvidos no trabalho e demais informações que se façam necessárias de maneira organizada no arquivo PDF mencionado acima;
- As informações devem ser apresentadas de forma clara e objetiva;
- Códigos mal indentados terão nota descontada;
- Caso seja verificado cópia entre códigos desenvolvidos pelos grupos a nota será **ZERO**;
- Caso seja verificado cópia integral de código de IAs generativas a nota será **ZERO**;
- **O trabalho escrito (arquivo PDF) deve possuir capa, contracapa, sumário, introdução, desenvolvimento e conclusão, ou seja, deve-se seguir as normas da ABNT.**

Projeto final – Tema “Livre”

Optar por temas relativos à indústria caso seja possível.

Data de entrega final: 25/11/2025

O tema “Gerenciador de Tarefas” não é um tema permitido.

Valor: 4,5 pontos

Desenvolva um projeto da sua escolha, **ele deve passar pela validação do professor**. Cada equipe terá a liberdade de escolha desde que atenda aos **requisitos mínimos** definidos. **O tema escolhido pela equipe deve ser apresentado até o dia 14/10/2025**. Após tal prazo, não será possível trocar o tema e as equipes que não apresentarem um tema, receberão um tema escolhido pelo Professor.

Requisitos Mínimos a serem atendidos

- 1. CRUD completo (CREATE, READ, UPDATE e DELETE):**
 - O sistema deve possuir no **MÍNIMO** três CRUD's diferentes, permitindo o cadastro, alteração, deleção e listagem de informações relevantes para o projeto;
 - Pode-se utilizar o framework Flask ou o framework Express.
- 2. No mínimo três páginas de navegação abertas:**
 - O projeto deverá conter pelo menos três páginas (**ABAS**) de navegação abertas (antes de fazer o login), sem contar com a página de login do usuário.
- 3. Sistema de Upload/Download:**
 - A aplicação deve conter no mínimo um sistema de *upload/download* de arquivos.

4. Sistema de Login:

- É necessário criar um sistema de login com um botão para se cadastrar e uma opção de “Lembrar senha”;
- O Hash é obrigatório.

5. Sessões:

- É obrigatório o uso de sessões.

6. MVC (Model, View, Controller):

- A organização do projeto deve seguir a **arquitetura MVC**, separando adequadamente a lógica de negócio (**Model**), a interface do usuário (**View**) e a camada de controle (**Controller**) com a implementação da validação dos dados de entrada.

7. Utilizar estruturas de controle e laços de repetição:

- Utilize laços de repetição (**for, foreach, while ou do-while**) e estruturas de controle (**if, elif, else**) de maneira apropriada no sistema, propiciando um fluxo lógico e eficiente;
- Realizar todas as validações necessárias para que, por exemplo, campos em branco não sejam “cadastrados” na(s) tabela(s) do banco de dados.

8. Banco de Dados:

- A escolha do banco de dados fica a escolha de cada equipe. Importante ressaltar que no dia da apresentação este ponto também será avaliado.

9. HTML:

- Não é permitido o uso de Tags depreciadas no HTML;

10. CSS

- Utilize o CSS, em todas as páginas, para deixar seu projeto com uma interface mais amigável para o usuário.

11. Javascript

- A utilização de Javascript é permitida desde que não ultrapasse 30% do código implementado, caso se utilize o *framework* Flask;

12. Banco de dados – Tabelas:

- Avaliação da qualidade das tabelas criadas no banco de dados;
- Avaliação das *queries* criadas e utilizadas no projeto.

13. Modularização

- É essencial que o código seja implementado de forma modularizada.

14. API

- A implementação de uma API no projeto em questão equivale a nota da segunda APS **(Nota: 0,5 pontos)**;

15. Documentação:

- Arquivo PDF com a explicação e comentários pertinentes relativos a páginas e/ou funções que sejam essenciais para o funcionamento do código **(Nota: 0,5 pontos)**;

16. Funcionamento:

- Código funcionando de maneira adequada e atendendo os requisitos mínimos descritos **(Nota: 2,0 pontos)**;

17. Defesa:

- A defesa equivale a 33% da nota do projeto **(Nota: 1,5 pontos)**;
- A nota de defesa é dada ao grupo, ou seja, se um ou mais integrantes não conseguir responder a um ou mais questionamentos sobre o código ou o trabalho em PDF a nota é retirada de todos os alunos do grupo.