

**DISCIPLINA:** Backend Development.

**PROFESSOR:** Fábio I. Kravetz.

Observações:

- As equipes devem ter no máximo 5 integrantes, de preferência as equipes de jornada;
- **Devem ser entregues os códigos desenvolvidos. Os nomes dos alunos devem constar no arquivo app.py;**
- Deve ser realizado o *upload* de um arquivo .rar ou zip no link disponibilizado no AVA;
- Somente 1 (UM) integrante da equipe deve realizar a entrega.
- Após a finalização do desenvolvimento, deve-se realizar a apresentação da aplicação em funcionamento e a defesa do código para o Professor.
- O valor da APS é de 0,5 pontos, dividido da seguinte forma:
  - 0,25 pontos – Código em pleno funcionamento;
  - 0,25 pontos – Defesa do código desenvolvido. Os questionamentos serão realizados a todos os integrantes da equipe e, caso a(s) resposta(s) seja(m) inadequada(s), a pontuação é retirada da equipe.

## ATIVIDADE PRÁTICA SUPERVISIONADA 01

### 1) Requisitos Funcionais

- **Criação de Tarefas:** O usuário deve ser capaz de adicionar uma nova tarefa, fornecendo, no mínimo, um título, uma descrição, um status e o horário de cadastro da tarefa;
- **Visualização de Tarefas:** A página inicial da aplicação deve exibir a lista de tarefas existentes.
- **Atualização de Tarefas:** O usuário deve ter a opção de editar as informações de uma tarefa, como título, descrição e status;
- **Conclusão de Tarefas:** Deve ser possível marcar uma tarefa como "concluída". Visualmente, tarefas concluídas devem ser diferenciadas das pendentes;
- **Exclusão de Tarefas:** O usuário poderá remover uma tarefa da lista;
- **Priorização de Tarefas:** As tarefas devem possuir um nível de prioridade (ex: Baixa, Média, Alta), que deve ser visível na lista de tarefas.

## 2) Requisitos Não Funcionais

- **Framework:** A aplicação deve ser desenvolvida utilizando Flask;
- **Arquitetura:** A estrutura do projeto deve seguir os princípios da arquitetura MVC;
- **Banco de dados:** Deve ser utilizado o MySQL ou SQLAlchemy ou um banco de dados de sua preferência;
- **Interface do Usuário:** A interface deve ser limpa, intuitiva e responsiva;
- **Sem Autenticação:** O sistema não necessita de um mecanismo de login e autenticação de usuários. As tarefas podem ser armazenadas em sessão ou no armazenamento local do navegador.