

Instruções: Projeto de Integração de Competências Profissionais

Curso: Programador Back-End – SENAI Maracanã **Instrutor:** Daniel Alves **Unidade Curricular:** Integração de Competências (20h)

1. O Desafio

O objetivo deste projeto é integrar os conhecimentos adquiridos em Lógica de Programação, Modelagem de Sistemas, Banco de Dados e Desenvolvimento de APIs. Vocês devem construir uma API RESTful utilizando **Python + Flask + MySQL**, garantindo a segurança, robustez e qualidade do código.

2. Formato de Trabalho

- **Grupos:** Máximo de 04 integrantes.
- **Apresentação:** Obrigatória no último dia de aula (ao vivo), demonstrando as rotas funcionando e o código.
- **Entrega:** Via Google Forms (um envio por grupo) contendo:
 1. Nomes completos dos integrantes.
 2. Link do repositório no **GitHub**.
 3. Arquivo **PDF** contendo Documento de Requisitos e Diagrama de Classes UML.

3. Temas Disponíveis (Escolha 1)

Os grupos podem repetir os temas, mas o desenvolvimento deve ser original para cada grupo.

1. Pilar Sério (Corporativo/Gestão)

Tema: API de Sistema de Service Desk (Chamados Técnicos) Este projeto foca em processos rígidos e conformidade com o perfil de Programador de Internet.

- **A proposta:** Criar uma API para gerenciar tickets de suporte de uma empresa.
- **Regras de Negócio:**
 - O usuário abre um chamado com prioridade (Baixa, Média, Alta).
 - A API deve permitir atualizar o status (Aberto, Em Atendimento, Concluído).
 - Deve haver uma rota para listar chamados por setor ou prioridade.
- **Destaque Técnico:** Foco total em **CRUD robusto** e **documentação de requisitos**.

2. Pilar Gamificado (Engajamento/RPG)

Tema: API "HabitQuest" (Gerenciador de Hábitos com XP) Ideal para alunos que gostam de lógica de jogos e interatividade.

- **A proposta:** Uma API onde cada tarefa concluída no banco de dados gera "XP" (experiência) para o usuário.
- **Regras de Negócio:**
 - Ao criar uma tarefa (POST), o usuário define a dificuldade.
 - Ao concluir (PATCH/PUT), a API calcula o bônus de XP e atualiza o nível do usuário no MySQL.
 - Rota de "Ranking" (Leaderboard) que ordena os usuários por nível.
- **Destaque Técnico:** Foco em **lógica booleana** e **estruturas de controle** para cálculo de recompensas.

3. Pilar Criativo/Lifestyle (Utilitário Pessoal)

Tema: API "VinoVault" ou "GameShelf" (Colecionador Digital) Focado em personalização e facilidade de uso (UX/UI mental).

- **A proposta:** Uma API para catalogar coleções pessoais (vinhos, jogos físicos, livros ou discos).
- **Regras de Negócio:**
 - Cadastro com campos específicos: data de aquisição, avaliação (estrelas) e fotos (links).
 - Funcionalidade de "Empréstimo": marcar qual item da coleção está emprestado e para quem.
 - Filtros de busca por categoria ou avaliação.
- **Destaque Técnico:** Foco em **modelagem de banco de dados** (entidade-relacionamento) e integridade referencial.

4. Requisitos Técnicos Obrigatórios

1. **Ambiente Virtual:** O projeto deve conter um venv configurado (não esqueça de incluir o arquivo requirements.txt).
2. **Banco de Dados:** MySQL com pelo menos 3 tabelas relacionadas (Foreign Keys).
3. **CRUD Completo:** As rotas devem permitir Criar, Ler, Atualizar e Deletar informações.
4. **Tratamento de Erros:** Uso correto de status codes (ex: 201 Created para sucessos, 404 Not Found para buscas vazias, 400 Bad Request para erros de validação).

Bom desenvolvimento a todos!