**Projeto Integrador Assistente de Desenvolvimento de Aplicativos Computacionais II**

Curso: Técnico em Informática

Turma: INF 13

Professor: Guilherme Eduardo Walter

Datas: Entrega: 25/08 Recuperação: 27/08 Apresentação: 01/09

**1. Objetivo**

Desenvolver um sistema de Gestão Escolar para controle de professores, alunos e cursos, incluindo funcionalidades de cadastro, edição, exclusão, vinculação e listagem com filtros.

**2. Critérios Obrigatórios**

* Repositório no GitHub:
  + Deve conter commits de todos os integrantes do grupo.
  + README explicativo do projeto.
* Modelagem de Banco de Dados (DER):
  + Diagrama mostrando tabelas e relacionamentos.
  + Pode ser feito em Draw.io, Drawdb.app, dbdiagram.io ou similar.
* Esboço de Telas:
  + Wireframes feitos no Figma.com, Wireframe.cc (ou similar) com as principais telas do sistema.
* Gerenciamento de Projeto:
  + Cada grupo deve usar uma ferramenta de gerenciamento para organizar o trabalho, rastrear tarefas e visibilidade de progresso.
  + Sugestões:
    - Trello: quadros com listas (Backlog, Em andamento, Revisão, Concluído).
    - Jira: (Kanban ou Scrum Board)

**3. Módulos do Sistema**

* Autenticação
  + Tela de login obrigatória.
  + Restrições de acesso para usuários autenticados.
* Professores
  + Cadastro com validação de e-mail.
  + Listagem com filtro por nome. Deve buscar tanto pelo nome quanto pelo sobre sobrenome e não usar diferenciar letras minúsculas e maiúsculas. Exemplo: Se buscar por "gui", deve listar "Guilherme".
  + Detalhamento completo.
  + Edição de e-mail, telefone, ativo e formação.
  + Exclusão.
  + Vinculação de professor a curso (apenas se ambos ativos).
* Alunos
  + Cadastro com validação de e-mail.
  + Listagem com filtro por nome. Deve buscar tanto pelo nome quanto pelo sobre sobrenome e não usar diferenciar letras minúsculas e maiúsculas. Exemplo: Se buscar por "gui", deve listar "Guilherme".
  + Detalhamento completo.
  + Edição de e-mail, telefone e ativo.
  + Exclusão.
  + Vinculação de aluno a curso (apenas se ambos ativos).
* Cursos
  + Cadastro.
  + Listagem com filtros por nome, categoria e valor mínimo.
  + Detalhamento completo.
  + Edição de descrição, categoria, valor, carga horária, ativo e professor.
  + Adicionar professor ou aluno (restrições de ativo).
  + Exclusão.

**4. Backend – API REST**

* Tecnologias: C# / .NET
* Rotas: serão definidas seguindo o padrão REST, com operações de criação, leitura, atualização e exclusão (CRUD) para as entidades do sistema.
* Validações:
  + Campos obrigatórios.
  + E-mail válido.
  + Entidades ativas para vinculação.
  + Tratamento de erros (entidade não encontrada, regras de negócio).
* Segurança:
  + Todos os endpoints privados e protegidos por autenticação.

**5. Frontend**

* Tecnologias: React / Angular / Vue / PHP
* Páginas:
  + Login.
  + Dashboard inicial com navegação para os módulos.
  + Listagens com filtros.
  + Cadastros.
  + Edições.
  + Detalhes com opções de exclusão e vinculação.

**6. Regras de Negócio**

* Professores e alunos só podem ser vinculados a cursos se estiverem ativos.
* Cursos precisam estar ativos para receber vínculos.
* Filtros de pesquisa devem ser case-insensitive usando LIKE.
* Retorno de dados deve obedecer os campos indicados para cada listagem.

**7. Entregas Mínimas (MVP)**

* Login com autenticação e proteção de todas as rotas.
* CRUD de professores, alunos e cursos (com filtros e regras de negócio).
* Wireframes das principais telas do sistema.
* Diagrama mostrando tabelas e relacionamentos.
* Banco de dados funcional.
* Interface navegável (mesmo que simples).
* Documentação no README com:
  + Descrição do projeto
  + Tecnologias usadas
  + Lista de integrantes e papéis
  + Fluxo de contribuição

**8. Prazos e Organização**

* 1ª Semana:
* Configuração do ambiente (.NET, banco de dados, repositório Git, frontend).
* Criação do DER.
* Criação dos wireframes das telas principais.
* Implementação do CRUD de Professores (backend + frontend).
* Implementação do CRUD de Alunos (backend + frontend).
* 2ª Semana:
* Implementação do CRUD de Cursos (backend + frontend).
* Funcionalidade de vinculação (professor/aluno ativo a curso ativo).
* Aplicação de regras de negócio e validações.
* Ajustes de filtros e tratamento de erros.
* 3ª Semana:
* Integração final entre módulos.
* Testes completos e correção de erros.
* Documentação do projeto.
* Preparação e ensaio da apresentação.
* Entrega final.

**9. Critérios de Avaliação**

* Cumprimento dos requisitos mínimos.
* Uso correto de validações e regras de negócio.
* Qualidade do código e organização do projeto no GitHub.
* Qualidade do DER e dos wireframes.
* Interface amigável e responsiva.
* Funcionalidade real do sistema.
* Participação de todos os integrantes.
* Apresentação do Projeto

**10. Grupos**

* **Grupo 1**
  + **Gabriel Halmenschlager Spall**
  + **Pedro Henrique Konradt**
  + **Everton Luiz Henrich Schneiders**
* **Grupo 2**
  + **João Vitor Rodrigues da Silva**
  + **Arthur Miguel Kowaslczuk de Carvalho**
  + **Alex dos Santos**
* **Grupo 3**
  + **Matheus Andre Gollmann**
  + **Samuel Borba Goulart**
  + **Pedro Agusto Bohnen dos Santos**
  + **Leonardo**
* **Grupo 4**
  + **Arthur Schena Giovanella**
  + **Bruno Rodrigues Zuge**
  + **Bruno Hensel Brites Junior**