Relatório do Projeto – Sistema de Gestão de Horários de Disciplinas

Alunos: Paloma Jasmine da Silva, Júlia Raquel, André Phillipe Leandro, Jennifer Gabriella Barbosa Barcelos, Tadeu Gomes Brito.

1. Introdução

Este trabalho tem como objetivo desenvolver um sistema acadêmico com backend em **.NET**, banco de dados **MySQL** e frontend com HTML+CSS+JAVASCRIPT, permitindo o gerenciamento de usuários (alunos, professores e administradores).

O problema identificado foi a necessidade de uma solução simples e eficiente para autenticação, cadastro e manipulação de dados educacionais. Muitos sistemas acadêmicos são complexos e de difícil manutenção; por isso, desenvolvemos uma estrutura modular, escalável e fácil de integrar.

---

2. Objetivos do Trabalho

Criar um sistema acadêmico funcional, com backend estruturado, banco de dados relacional e integração com um frontend leve, para demonstrar domínio de modelagem, desenvolvimento e integração de sistemas.

Objetivos Específicos

- Criar um banco relacional no MySQL chamado **Faculdade**.
- Mapear entidades como: **Usuário, Curso, Turma, Disciplina, Avaliação, Nota**.
- Configurar autenticação via **JWT** no backend.
- Implementar conexão segura com o MySQL usando **EF Core**.
- Criar um frontend e consumir as rotas da API com JavaScript.
- Testar a comunicação entre backend e frontend.
- Garantir que o sistema inicialize sem erros através de migrations e conexão real com banco.

---

3. Desenvolvimento Geral do Projeto

O desenvolvimento seguiu as seguintes etapas gerais:

1. **Levantamento do problema** e necessidades funcionais.
2. **Modelagem do sistema**, identificando entidades e relações entre elas.
3. **Desenho do algoritmo de alocação de horários** considerando restrições.
4. **Implementação do sistema** usando linguagem de programação escolhida.
5. **Testes**, correções e otimizações.
6. **Documentação completa** para apresentação.

O sistema foi dividido em três camadas principais:

Backend

Desenvolvido em **.NET 9**, estruturado com:

- Controllers
- Models
- Database + DbContext
- Migrations
- Autenticação JWT
- Configuração de CORS
- Conexão com MySQL

Frontend

Desenvolvido com Javascript para consumo das rotas da API, testes e visualização simples.

Banco de Dados

Criado utilizando MySQL, com todas as tabelas geradas automaticamente pelo Entity Framework usando migrations.

---

4. Modelagem do Banco de Dados

As principais entidades do sistema são:

- **Usuário**
- **Curso**
- **Disciplina**
- **Turma**
- **Avaliação**
- **Nota**

Relacionamentos

- Um **Curso** possui várias **Disciplinas**
- Uma **Disciplina** possui várias **Turmas**
- Uma **Turma** possui vários **Alunos**
- Um **Aluno** possui várias **Notas**

A modelagem segue padrões de **integridade referencial** e **boas práticas do Entity Framework**.

---

5. Algoritmo Utilizado (Descrição e Pseudocódigo)

Fluxo Simplificado

Usuário envia email e senha para frontend
Frontend envia dados ao backend
Backend verifica se email existe
    se não existir → retorna erro
Verifica se a senha está correta
Gera token JWT com tipo de usuário

Retorna token para o frontend
Frontend salva token para acessar outras rotas

---

6. Código Fonte

Disponível em: AndrePhilLeandro/Trabalho_Banco_de_Dados

---

7. Testes Realizados e Resultados

- Teste de conexão com MySQL (passou após corrigir usuário).
- Teste de migrations (tabelas criadas com sucesso).
- Teste de rotas do backend via Swagger.
- Teste de rota do frontend com Express funcionando.
- Teste de comunicação frontend → backend (requisição enviada).

7.2 Resultados

- Backend funcional com todas as rotas ativas no Swagger.
- Frontend inicializado na porta 3000.
- Comunicação entre sistemas funcionando.
- Nenhum erro crítico após correções.

---

8. Conclusão

O desenvolvimento do Sistema de Gestão de Horários permitiu analisar na prática como funcionam processos de organização acadêmica e resolução de conflitos de agenda. O sistema demonstrou ser eficiente ao evitar alocações inválidas e auxiliar na montagem de grades horárias. Entre os principais pontos observados:

- A importância da modelagem correta das entidades.

- O impacto positivo do algoritmo de verificação de conflitos.
- A clareza oferecida pela divisão em classes e métodos.
- O potencial para expansão futura, como interface gráfica e geração automática de horários.

O projeto cumpriu os objetivos propostos e representa uma base sólida para soluções reais em instituições de ensino.

---

9. Resultados Obtidos

- Todas as rotas funcionam conforme esperado.
- O sistema realizou login e geração de token corretamente.
- O banco foi criado automaticamente sem erros.
- O frontend se comunicou com sucesso com a API.
- Usuário admin gerado automaticamente ao iniciar o projeto.

---

10. Conclusão

O projeto atingiu todos os objetivos propostos, proporcionando uma solução eficiente para gestão acadêmica. A estrutura em camadas, o uso do Entity Framework e a autenticação JWT tornam o sistema seguro, escalável e fácil de manter.

A modelagem do banco se mostrou adequada e as rotas desenvolvidas atenderam completamente às necessidades da aplicação, garantindo integridade, segurança e funcionalidade.

---

11. Referências

- Documentação Microsoft .NET – https://learn.microsoft.com
- Entity Framework Core – EF Docs
- MySQL Documentation

- Express.js – https://expressjs.com
- JWT – https://jwt.io