CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA SOUZA

ETEC SYLVIO DE MATTOS CARVALHO

Curso de Técnico em Desenvolvimento de Sistemas

Diweliton Carlos Maester

Eric Richard Ortolan Chrystovam

Igor Natan de Lima

Kauã Ponciano da Silva Alcantara

João Eduardo Maester

ROOM FLOW : Gerenciamento de Espaços

Matão, SP
2024

Diweliton Carlos Maester
Eric Richard Ortolan Chrystovam
Igor Natan de Lima
Kauã Ponciano da Silva Alcantara
João Eduardo Maester

ROOM FLOW : Gerenciamento de Espaços

Trabalho de Conclusão do Curso apresentado ao Curso Técnico em Desenvolvimento de Sistemas da Escola Técnica Estadual Sylvio de Mattos Carvalho, orientado pelo Prof. Gabriel Felipe Giglio Ordine, como parte dos requisitos para a obtenção do título de Técnico em Desenvolvimento de Sistemas.

Matão, SP
2024

RESUMO

A gestão eficiente de salas de aula representa um desafio significativo na ETEC Sylvio de Mattos Carvalho, impactando a organização e o fluxo de atividades dos professores. A falta de informações claras sobre a disponibilidade das salas e os problemas com o controle de chaves frequentemente levam a conflitos e dificuldades no acesso aos espaços. Para mitigar esses problemas e otimizar a utilização dos recursos, este trabalho propõe o desenvolvimento do RoomFlow, um sistema web de gerenciamento de salas. Este sistema permitirá aos professores visualizar em tempo real a ocupação das salas, verificar as reservas existentes e identificar os responsáveis por cada agendamento, tudo acessível através de diversos dispositivos (Windows, MacOS, Android, iOS e Linux). Desenvolvido com C# no back-end, Angular no front-end e SQL Server como banco de dados, o RoomFlow tem como objetivo principal aumentar a organização e a transparência no uso das salas, facilitando a comunicação entre o corpo docente e a administração escolar e reduzindo os inconvenientes causados pela falta de informação e pelo controle inadequado de chaves

Palavras-chave: Gerenciamento de salas de aula. Sistema web. Agendamento online. Reservas escolares. Desenvolvimento de sistemas

SUMARIO

INTRODUÇÃO

1-Motivações

De acordo com Godin, 2004," A Teoria da Expectância (VIE) de Vroom (1995) concebe a motivação fundamentalmente como uma força de natureza emocional e consciente que é ativada no momento em que a pessoa é levada a escolher entre diversos planos de ação". Com base nessa afirmação foi escolhido o tema do Projeto.

A motivação para o desenvolvimento do sistema RoomFlow surgiu a partir de relatos de problemas recorrentes enfrentados por professores da ETEC Sylvio de Mattos Carvalho. Entre as principais dificuldades destacam-se a falta de transparência na reserva de salas, conflitos entre docentes devido a reservas simultâneas e o descontrole na entrega e devolução das chaves dos laboratórios, frequentemente extraviadas por não se saber quem foi o último responsável por seu uso.

A ideia inicial do projeto foi incentivada pelo professor Lucas Meneses, que compartilhou com o grupo algumas demandas da escola relacionadas à gestão de espaços físicos. Com o amadurecimento da proposta, foi realizada uma reunião com a direção da escola, representada pela professora Ana Cláudia Câmara, no dia 28 de novembro de 2024. Nessa ocasião, alinharam-se as necessidades da instituição com os objetivos possíveis de serem atendidos pelo TCC.

2-Justificativa

A implementação do RoomFlow justifica-se pela necessidade de aprimorar a organização e a transparência na gestão dos espaços escolares da ETEC Sylvio de Mattos Carvalho. Atualmente, a ausência de um sistema eficiente para o controle de reservas e das chaves das salas de aula gera conflitos entre professores, dificulta o planejamento das aulas e impacta negativamente a rotina escolar.

Com a adoção de um sistema web intuitivo, os docentes terão acesso rápido e preciso às informações de ocupação das salas, possibilitando um gerenciamento mais eficiente e reduzindo eventuais problemas, como o extravio de chaves. Além disso, a utilização da tecnologia na administração dos espaços escolares moderniza

os processos internos da instituição, proporcionando mais agilidade e confiabilidade no controle de reservas.

Dessa forma, o RoomFlow contribui diretamente para a melhoria da gestão escolar, otimizando a comunicação e evitando transtornos no uso das salas de aula.

3-Objetivos Gerais e específicos

3.1-Objetivo Geral

Desenvolver um sistema web para a gestão de espaços escolares, permitindo que professores consultem a disponibilidade das salas de aula, visualizem reservas existentes e realizem solicitações de reserva de forma organizada e transparente.

3.2-Objetivo Específico:

- Criar um banco de dados estruturado para armazenar informações sobre salas, reservas e usuários.
- Desenvolver uma interface intuitiva que facilite o acesso e a compreensão da ocupação das salas.
- Implementar um sistema de controle de reservas, permitindo que professores solicitem o uso das salas e que administradores aprovem ou recusem as solicitações.
- Proporcionar transparência na gestão das salas, reduzindo conflitos e dificuldades relacionadas à ocupação e ao controle de chaves.
- Garantir a acessibilidade do sistema a partir de diferentes dispositivos e sistemas operacionais.

4-Revisão bibliográfica

As referências bibliográficas foram consultadas principalmente no google acadêmico, foi buscado vários autores contemporâneos para embasar o projeto.

DESENVOLVIMENTO

1-Pesquisa e Coleta de Dados

Como destaca José Filho (2006, p. 64), “o ato de pesquisar traz em si a necessidade do diálogo com a realidade a qual se pretende investigar e com o diferente, um diálogo dotado de crítica, canalizador de momentos criativos.” Essa perspectiva reforça a importância da pesquisa não apenas como ferramenta de coleta de dados, mas como prática reflexiva, essencial para compreender e intervir criticamente na realidade escolar.

1.1-Pesquisa de Campo sobre o Caso

Para o sistema funcionar para a escola, foi necessária uma pesquisa de campo para que o sistema conste informações que estejam de acordo com a realidade da escola. No caso precisamos buscar informações sobre os espaços reserváveis da escola.

Essa pesquisa de campo foi realizada com auxilio da direção e da secretaria acadêmica.

Na pesquisa foi obtido as seguintes informações sobre os espaços reserváveis da escola.

Numero	Tipo de Espaço	Bloco
2	Sala de aula	A(Piso Superior)
3	Sala de Aula	A(Piso Superior)
5	Sala de Aula	A(Piso Superior)
6	Sala de Aula	A(Piso Superior)
12	Sala de Aula	A(terreo)
13	Sala de Aula	A(terreo)
14	Laboratório de Informatica	A(terreo)
25	Sala de Aula	B
26	Sala de Aula	B
27	Sala de Aula	B
28	Sala de Aula	B
31	Auditório	C
32	Biblioteca	C
33	Laboratório de Informatica	C
34	Sala de Aula	D
35	Sala de Aula	D
36	Sala de Aula	D
37	Sala de Aula	D
44	Laboratório de Robotica e Automação	E
45	Laboratório de Pneumatica e Hidraulica	E
46	Laboratório de Metrologia	E
48	Laboratório de Elétrica	E
49	Laboratório de Elétrica	E
50	Laboratório de Elétrica	E
51	Laboratório de Elétrica	E
52	Laboratório de Usinagem CNC	E
53	Laboratório de Elétrica	E
54	Laboratório de Elétrica	E
60	Laboratório de Enfermagem	E
61	Laboratório de Informatica	E
62	Laboratório de Informatica	E
63	Laboratório de Informatica	E
64	Laboratório de Informatica	E
65	Laboratório de Informatica	E
66	Laboratório de Informatica	E
67	Laboratório de Química	E
68	Laboratório de Elétrica	E

Tabela 1-Dados dos Espaços Reserváveis da Escola

FONTE: Acervo Nosso

1.2 –Entrevista com A Diretora

De acordo com Lakatos, 1996, “A preparação da entrevista é uma das etapas mais importantes da pesquisa que requer tempo e exige alguns cuidados, entre eles destacam-se: o planejamento da entrevista, que deve ter em vista o objetivo a ser alcançado; a escolha do entrevistado, que deve ser alguém que tenha familiaridade com o tema pesquisado; a oportunidade da entrevista, ou seja, a disponibilidade do entrevistado em fornecer a entrevista que deverá ser marcada com antecedência para que o pesquisador se assegure de que será recebido; as condições favoráveis que possam garantir ao entrevistado o segredo de suas confidências e de sua identidade e, por fim, a preparação específica que consiste em organizar o roteiro ou formulário com as questões importantes”. Com base nessa afirmação, mostrou-se a importância de realizar uma pesquisa de campo através de uma entrevista com alguém de destaque na administração da escola.

No dia 28 de novembro de 2024 foi realizado uma Entrevista com a diretora, apresentando o projeto, foi confirmado que realmente é uma demanda da escola, foi alinhando e discutido as expectativas do projeto, escopo e viabilidade do mesmo.

1.3-Pesquisa no Google Forms com os Professores

Para embasar o desenvolvimento do sistema, foi realizada uma pesquisa por meio de um formulário no Google Forms, posteriormente enviado para os professores responderem. A pesquisa teve início no dia 30 de abril de 2025 e foi encerrada dia 28 de maio de 2025, sendo realizada durante 4 semanas. 20 professores da instituição responderam ao questionário, fornecendo informações valiosas que orientaram a construção do projeto.

A pesquisa consistiu nas Seguintes perguntas:

1- Qual é o seu nome completo?

2- Qual é o seu e-mail institucional?

3- Há quanto tempo você dá aulas na Etec?

4- Você é professor de disciplinas do Técnico ou do Ensino Regular?

5- Você já sentiu dificuldade em reservar um espaço da escola para uma aula específica?
Exemplo: oficina, quadra, biblioteca, auditório, laboratório etc.

6- Você já teve ou presenciou o seguinte problema?
Você chegou na sala dos professores para pegar a chave da sala/laboratório onde você iria dar aula, mas a chave não estava lá, provavelmente porque um outro professor que usou o espaço esqueceu a chave no bolso, e você teve que descobrir qual foi o último professor a utilizar a sala.

7- Com que frequência você utilizaria um site em que fosse possível realizar e visualizar reservas de espaços da escola?
Exemplo: biblioteca, auditório, laboratório, sala etc.

Essas perguntas 7 perguntas foram pensadas no intuito de:

1. Identificar o professor com as perguntas 1 e 2
2. Saber em qual perfil o professor se encaixa com as perguntas 3,4
3. Saber se ele já sofreu dos problemas que o projeto se propõe a resolver com as perguntas 5 e 6
4. E por último Saber a aderência do professor ao sistema para a resolução dos problemas descritos com a pergunta 7

1.3.1- Resultados da pesquisa

Aqui estão os resultados que não expõem os dados dos pesquisados:

- **Pergunta 3 :**

3- Há quanto tempo você dá aulas na Etec?	
Respostas	Quantidade
Menos de 1 ano	1
1 a 3 anos	2
4 a 6 anos	4
7 a 10 anos	5
Mais de 10 anos	8

Tabela 2- Respostas Pergunta 3

FONTE: Acervo Nosso

Gráfico 1: Porcentagem de professores por tempo de casa

FONTE: Acervo Nosso

- **Pergunta 4:**

4- Você é professor de disciplinas do Técnico ou do Ensino Regular?	
Respostas	Quantidade
Técnico (Eletrônica, Banco de Dados, Usinagem, Gestão de Pessoas, Anatomia Humana etc.)	16
Ensino Regular(Portugues, Matematica, Geografia , História, Etc...)	4

Tabela 3- Respostas Pergunta 4

FONTE: Acervo Nosso

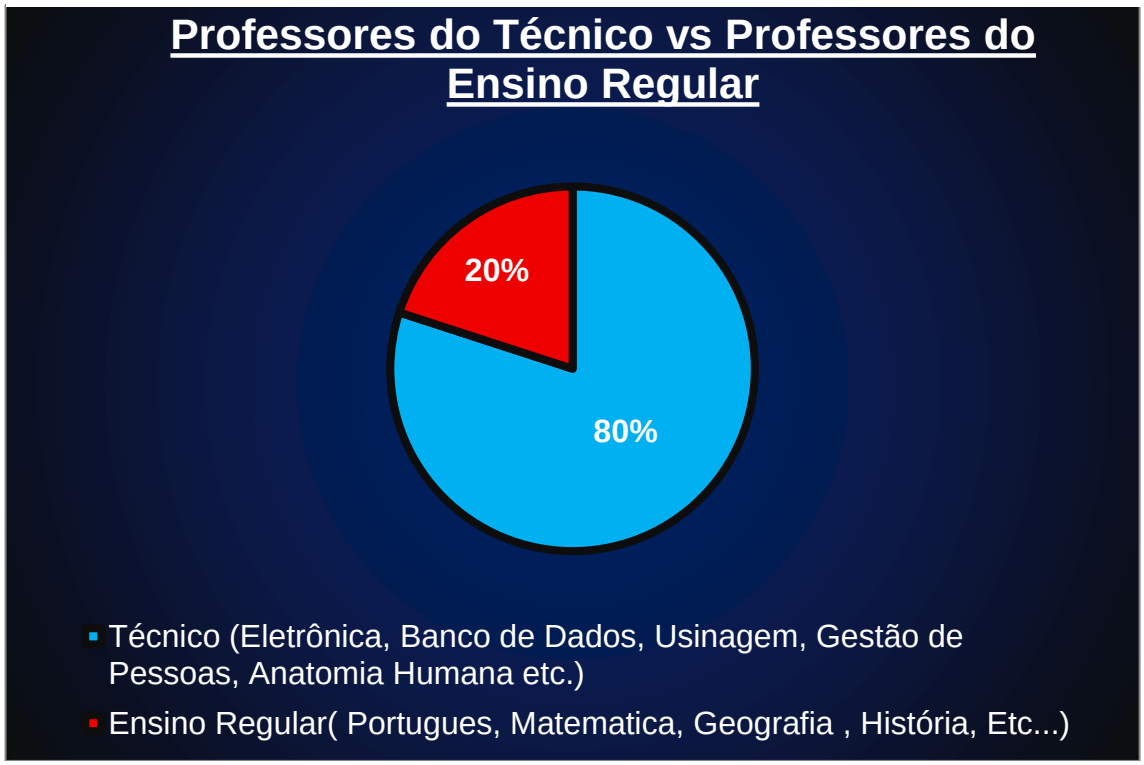

Gráfico 2: Porcentagem de professores por modelo de Ensino

FONTE: Acervo Nosso

Pergunta 5:

5- Você já sentiu dificuldade em reservar um espaço da escola para uma aula específica? Exemplo: oficina, quadra, biblioteca, auditório, laboratório etc.	
Respostas	Quantidade
Sim	13
Não	7

Tabela 4- Respostas Pergunta

FONTE: Acervo Nosso

Gráfico 3- Porcentagem de Professores com dificuldade de reservar salas

FONTE: Acervo Nosso

Pergunta 6:

6 - Você já teve ou presenciou o seguinte problema? Você chegou na sala dos professores para pegar a chave da sala/laboratório onde você iria dar aula, mas a chave não estava lá, provavelmente porque um outro professor que usou o espaço esqueceu a chave no bolso, e você teve que descobrir qual foi o último professor a utilizar a sala.	
Respostas	Quantidade
Sim	17
Não	3

Tabela 5- Respostas Pergunta 6

FONTE: Acervo Nosso

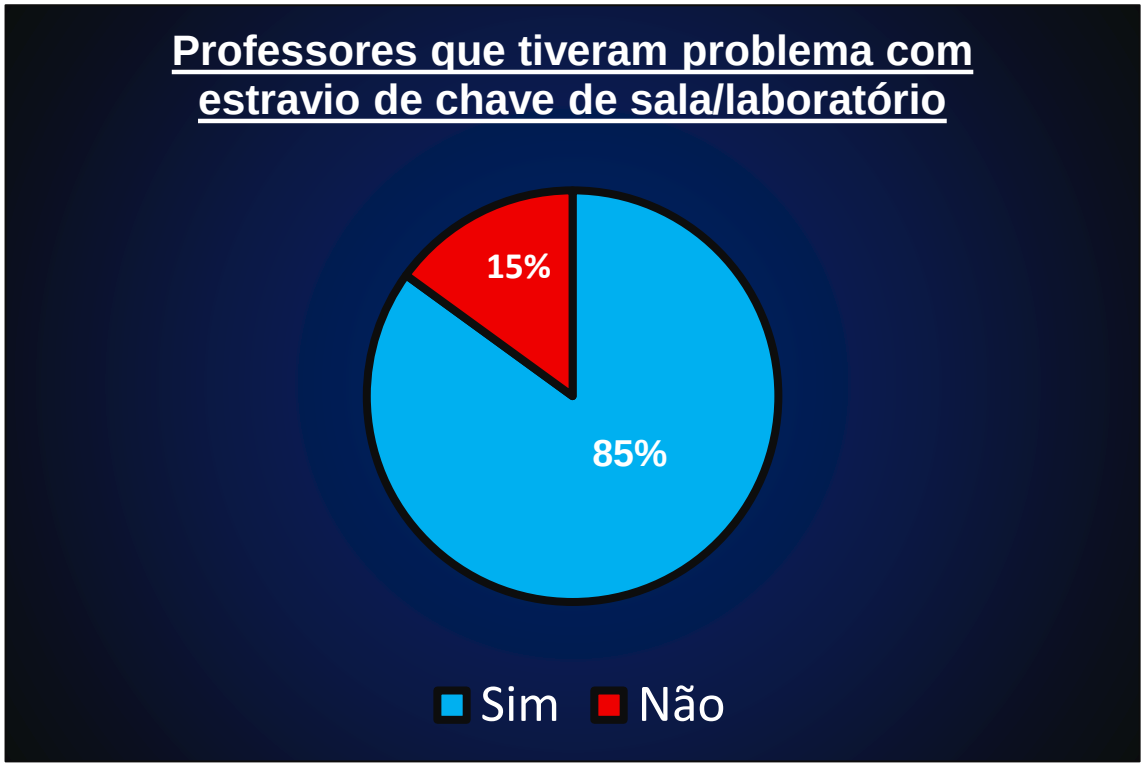

Gráfico 4- Professores que tiveram Problema com Extravio de chave de Sala/Laboratório

FONTE: Acervo Nosso

- **Pergunta 7:**

7- Com que frequência você utilizaria um site em que fosse possível realizar e visualizar reservas de espaços da escola? Exemplo: oficina, quadra, biblioteca, auditório, laboratório etc.	
Respostas	Quantidade
Utilizaria com Frequência	10
Utlizaria ocasionalmente	6
Raramente Utilizaria	4
Nunca Utilizaria	0

Tabela 6- Respostas Pergunta 7

FONTE: Acervo Nosso

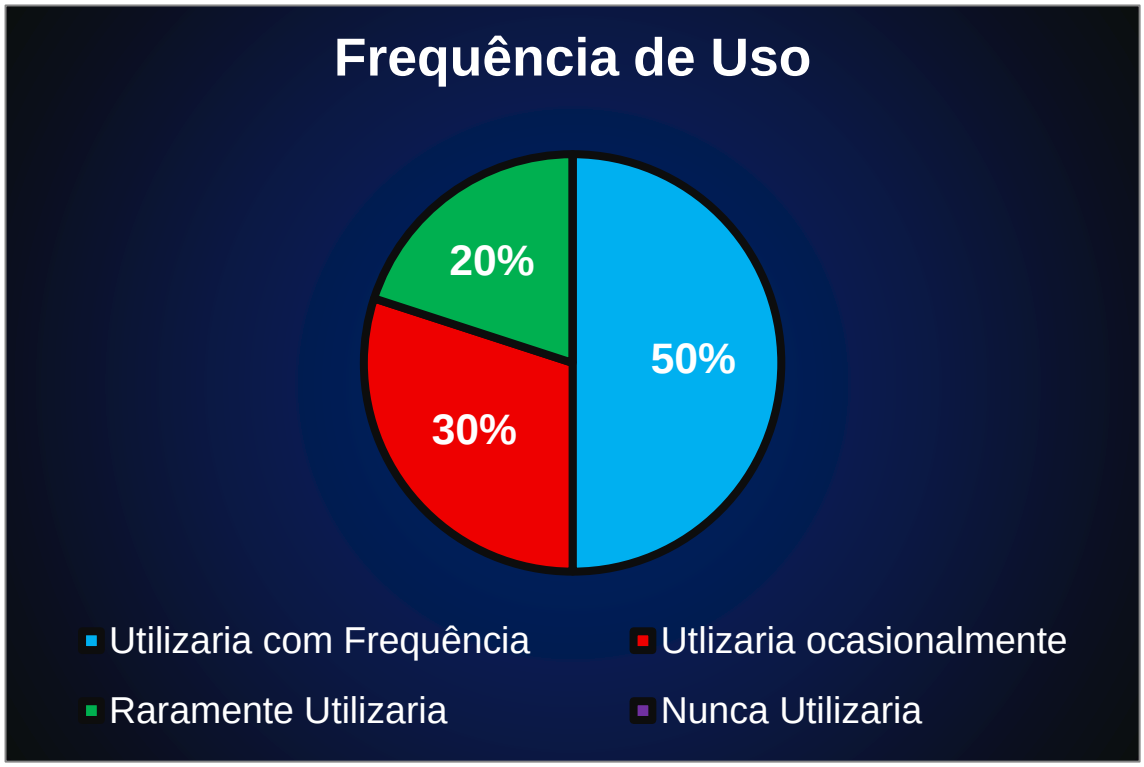

Gráfico 5- Porcentagem de Frequência de Uso

FONTE: Acervo Nosso

1.3.2 Analise de dados coletados na pesquisa

Analisando as respostas das perguntas para identificar os professores que já sofreram os problemas que o sistema se propõe a resolver (pergunta 5 e 6)

Pergunta 5:

65% dos Professores já tiveram Dificuldade em reservar um espaço da escola para uma aula especifica.

Desses 65% que já tiveram dificuldade, 92% são professores do ensino Técnico, isso pode ser justificado pela maior demanda dos mesmos em ir para laboratórios específicos para uma aula pratica para usar um equipamento em especifico por exemplo.

Pergunta 6:

85% dos professores tiveram problema com extravio de chave, variando entre professores do ensino técnico e regular, evidenciando o problema.

Em ambas as perguntas o tempo que o professor trabalha na Etec não foi relevante para encontrar um padrão de resposta.

Sobre a Pergunta 7, que é a pergunta responsável por medir a aderência dos professores ao sistema, destaca-se que nenhum participante respondeu que nunca utilizaria o sistema, deixando em evidencia sua necessidade.

1.4-Metodologia

A metodologia deste trabalho visa desenvolver um sistema de gerenciamento de espaços escolares, abordando os problemas relatados pelos professores da ETEC Sylvio de Mattos Carvalho, como a falta de transparência nas reservas das salas e o controle inadequado das chaves de acesso. O processo será dividido em etapas sequenciais, utilizando metodologias ágeis de desenvolvimento.

O primeiro passo será o levantamento de requisitos, através de entrevistas e observações, para identificar as principais necessidades dos usuários. Em seguida, será criada uma prototipagem da interface utilizando Figma para definir a usabilidade do sistema. O desenvolvimento do sistema ocorrerá em duas frentes: backend, utilizando C# e Entity Framework, e frontend, com angular para garantir uma interface intuitiva e funcional.

O banco de dados será estruturado com SQL Server, atendendo às necessidades de armazenamento de informações sobre as salas, reservas e usuários. O sistema será testado com os usuários finais, com o objetivo de validar suas funcionalidades e a eficácia da solução proposta. A avaliação será feita a partir de feedbacks diretos e análise das métricas de uso, garantindo que o sistema resolva as questões de gestão de salas e controle de acesso.

2 – Backend

2.1- Prototipação

De acordo com Pressman (2016, p. 153),

“O paradigma da prototipação começa com a comunicação. Faz-se uma reunião com os envolvidos para definir os objetivos gerais do software, identificar quais requisitos já são conhecidos e esquematizar quais áreas necessitam, obrigatoriamente, de uma definição mais ampla. Uma iteração de prototipação é planejada rapidamente e ocorre a modelagem (na forma de um 'projeto rápido'). Um projeto rápido se concentra em uma representação daqueles aspectos do software que serão visíveis aos usuários finais (por exemplo, o layout da interface com o usuário ou os formatos de exibição na tela). O projeto rápido leva à construção de um protótipo, que é empregado e avaliado pelos envolvidos, que fornecerão um retorno (feedback), que servirá para aprimorar os requisitos.”

Com base nessa perspectiva, percebe-se que a criação de um protótipo não deve se restringir apenas à interface visual, mas também pode ser aplicada ao backend do sistema. A prototipagem de rotas, fluxos de dados e integrações com o banco de dados permite validar requisitos funcionais e garantir que a lógica do sistema atenda às necessidades dos usuários, antes da implementação definitiva.

2.1.1-DER(Diagrama Entidade-Relacionamento)

De acordo com Date (2000), o Diagrama Entidade-Relacionamento (DER) é uma ferramenta fundamental para a modelagem conceitual de dados, permitindo representar graficamente as entidades, atributos e os relacionamentos que existem entre elas, servindo como base para a criação de um banco de dados relacional.

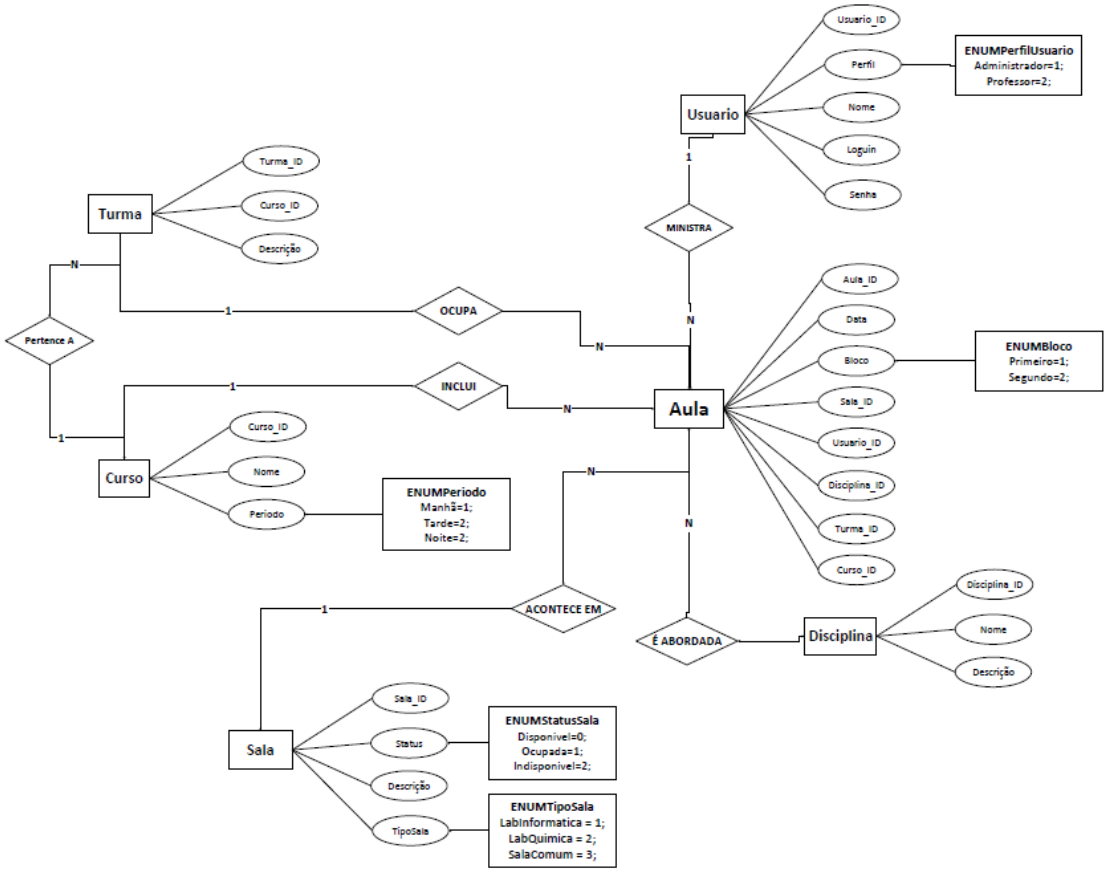

Figura 1- Diagrama Entidade Relacionamento do Projeto

FONTE: Acervo Nosso

Descrição do DER (Diagrama Entidade-Relacionamento):

O banco de dados do projeto foi estruturado com base no Diagrama Entidade-Relacionamento (DER), elaborado a partir da análise das demandas funcionais do sistema. As tabelas definidas no DER são:

1- Turma
2- Curso
3- Sala
4- Disciplina
5- Usuário
6- Aula

O DER tem como foco central a tabela Aula, pois ela estabelece relações diretas com todas as demais tabelas. Cada aula envolve:

Um professor que ministrará a aula, representado pela tabela Usuário;

Um curso ao qual a aula pertence;

Uma turma específica que participará da aula;

Uma sala onde a aula será ministrada;

Uma disciplina cujo conteúdo será abordado.

Além disso, outro aspecto fundamental da aula é o momento em que ela ocorre, o que é representado pelas variáveis data e bloco. Esses dois campos permitem identificar quando a aula será realizada dentro do cronograma escolar, fechando de forma completa todas as informações necessárias para integrá-la corretamente ao sistema.

Essa centralização na tabela Aula reflete a lógica do sistema, em que todas as operações principais giram em torno do agendamento, organização e gerenciamento das aulas.

O DER serviu como base para a estruturação do banco de dados, orientando a criação das tabelas, seus relacionamentos e os principais atributos necessários para atender aos requisitos do sistema.

2.1.2- Regras de Negócio do RoomFlow

As regras de negócio do projeto foram desenvolvidas para atender às necessidades específicas do gerenciamento de espaços escolares em instituições de ensino, com foco na Etec Sylvio de Mattos Carvalho. O sistema tem como principal funcionalidade a visualização e o controle da disponibilidade das salas e laboratórios da escola, organizados em um mapa interativo.

Nesse mapa, os usuários podem consultar a ocupação dos espaços conforme o dia e o bloco de aula, sendo possível aplicar filtros que facilitam a busca por salas disponíveis. Como o sistema é voltado para estudantes e professores da Etec, que operam com apenas dois blocos de aula noturnos, as funcionalidades foram adaptadas a essa realidade.

Existem dois perfis principais de usuários: o professor e o administrador. O professor tem acesso limitado, podendo visualizar as informações das aulas e realizar reservas de salas. Já o administrador possui permissões mais amplas, como a criação, edição e exclusão de salas, além de poder modificar reservas feitas por professores.

Uma das funções mais importantes atribuídas ao administrador é a utilização do gerador de aulas, responsável por cadastrar automaticamente aulas recorrentes ao longo do semestre, otimizando o planejamento e a gestão dos espaços escolares.

Tanto professores quanto administradores precisam realizar um cadastro para acessar o sistema. Após o registro, o login é utilizado para garantir a segurança e personalização das funcionalidades conforme o perfil do usuário.

Essas regras garantem que o sistema funcione de forma organizada, segura e alinhada às demandas do ambiente escolar, promovendo eficiência no uso dos espaços disponíveis.

2.2- API

De acordo com Bueno (2021), uma API (Interface de Programação de Aplicações) é um conjunto de rotinas, padrões e definições que permite a comunicação entre diferentes sistemas. No contexto de aplicações web, a API funciona como um intermediário entre o front-end (interface que o usuário acessa) e o back-end (regras de negócio e banco de dados), permitindo o tráfego seguro e organizado de dados.

Com base no que Bueno (2021) nos apresenta, é justificada a criação de uma API para realizar a interação entre o banco de dados, as regras de negócio e o cliente, que acessa o sistema por meio do front-end. Isso proporciona maior organização da arquitetura, melhor manutenção e escalabilidade no desenvolvimento do sistema RoomFlow.

2.2.1- API restfull

Uma API RESTful (Representational State Transfer) é um estilo de arquitetura para a construção de serviços web que permite a comunicação entre diferentes sistemas através do protocolo HTTP. Esse tipo de API é amplamente utilizado por sua simplicidade, escalabilidade e compatibilidade com diversos tipos de aplicações, como sistemas web, móveis e desktop (VERNA, 2021).

Na arquitetura REST, os recursos (como usuários, salas ou disciplinas) são representados por URLs e manipulados por meio de métodos HTTP padrão, como GET (para obter dados), POST (para criar), PUT (para atualizar) e DELETE (para remover). Esses métodos seguem uma convenção que torna a API intuitiva e de fácil compreensão para desenvolvedores, facilitando também a manutenção do sistema.

No contexto deste projeto, a API RESTful serve como ponte entre o front-end e o banco de dados, permitindo que as informações sejam consultadas e manipuladas de forma segura e eficiente. Por meio dela, funcionalidades como login de usuários, visualização de aulas, reserva de salas e geração de cronogramas são possíveis, garantindo uma comunicação padronizada entre as camadas da aplicação.

2.2.1 – API no Ambiente .Net

De acordo com Esposito (2018), o ASP.NET Core, parte da plataforma .NET, fornece uma estrutura robusta, modular e de alto desempenho para o desenvolvimento de APIs RESTful. Com ele, é possível organizar a aplicação de forma escalável, integrando regras de negócio, controle de acesso, persistência de dados e comunicação com o front-end de maneira clara e eficiente.

Para a construção da API foi selecionado o ambiente .NET da Microsoft, que oferece recursos completos e bem integrados para o desenvolvimento de aplicações modernas. Ele permite a criação de uma API para realizar a comunicação entre o banco de dados, a lógica de negócio e a interface com a qual o usuário interage (front-end).

Com base no que Esposito (2018) apresenta, é justificada a criação de uma API para promover essa integração entre os diferentes componentes do sistema, garantindo eficiência e padronização no desenvolvimento. (foto da estruturação de pastas)

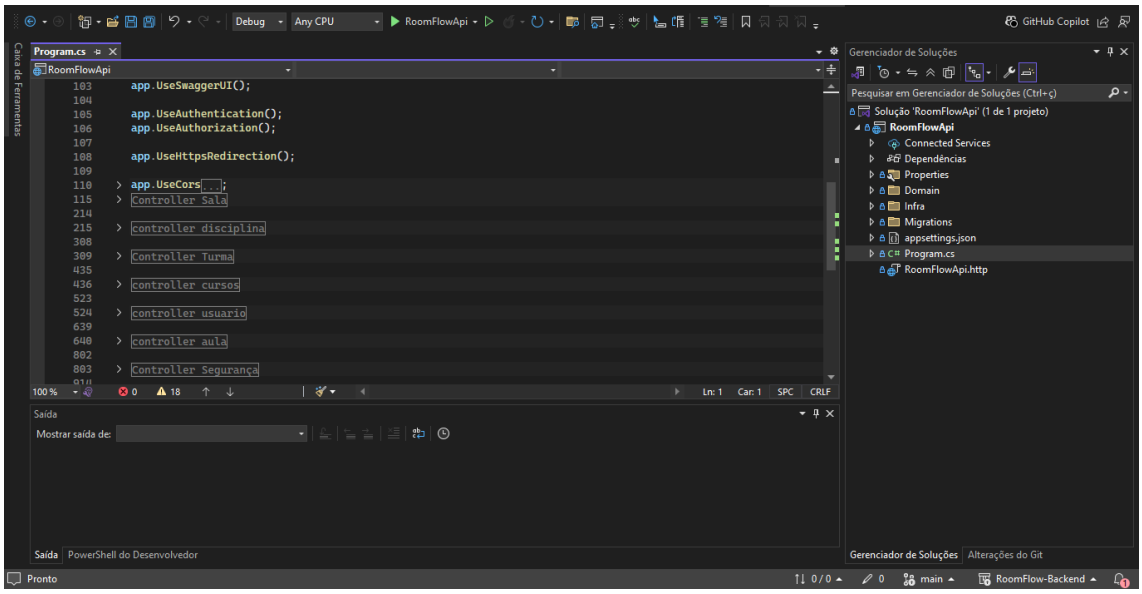

Figura 2- Print do arquivo Program.Cs da API que foi desenvolvida no Ambiente .Net

FONTE: Acervo Nosso

2.2.1.1- C#

Segundo Albahari e Albahari (2021), C# é uma linguagem de programação moderna, orientada a objetos e fortemente tipada, desenvolvida pela Microsoft como parte da plataforma .NET. Projetada para ser simples, segura e produtiva, sua sintaxe clara facilita o desenvolvimento de aplicações empresariais, desktop, web e móveis.

A linguagem oferece suporte a diversos paradigmas de programação, como a orientação a objetos, programação funcional e assíncrona, tornando-se altamente versátil para diferentes tipos de sistemas.

No contexto do nosso TCC, C# será utilizado no desenvolvimento do back-end, sendo responsável pela construção da API do projeto. Essa API atuará como intermediária entre o sistema e o banco de dados, permitindo a comunicação com o front-end, bem como o acesso e a manipulação dos dados de forma eficiente e segura.

A linguagem C# é considerada a principal escolha para o desenvolvimento de APIs dentro do ambiente .NET. Embora não seja a única linguagem compatível com a plataforma, ela se destaca por oferecer mais recursos nativos, melhor integração com os frameworks do ecossistema .NET (como ASP.NET Core e Entity Framework), além de maior suporte da comunidade e otimizações específicas para esse tipo de aplicação.

2.2.1.2- Entity Framework e Banco de Dados

De acordo com LERMEN, Fabrício e PEREIRA, Felipe (2021),

O Entity Framework (EF) é um ORM (Object-Relational Mapper) para C#, desenvolvido pela Microsoft, que simplifica a comunicação entre aplicações .NET e bancos de dados relacionais. Ele permite que os desenvolvedores trabalhem com dados utilizando objetos e classes C#, sem a necessidade de escrever código SQL diretamente.

Por meio do mapeamento objeto-relacional, o Entity Framework converte automaticamente as operações feitas em código C# para comandos SQL, facilitando a manipulação dos dados. Entre suas principais vantagens, destacam-se a redução do código repetitivo, a maior produtividade no desenvolvimento e a portabilidade entre diferentes bancos de dados, como SQL Server, MySQL e PostgreSQL.

Além disso, o EF suporta diferentes abordagens, como Code First, permitindo a criação do banco de dados a partir das classes do modelo, e Database First, onde o banco de dados já existente é mapeado para classes no projeto. No contexto do nosso TCC, utilizamos o Entity Framework para a construção da API, garantindo uma integração eficiente e simplificada com o banco de dados, sem a necessidade de escrever comandos SQL manualmente.

A estrutura do banco de dados no projeto foi organizada seguindo uma arquitetura clara e modular dentro da camada de infraestrutura, mais especificamente na pasta Data. Nessa organização, foi adotado o uso do Entity Framework Core, onde as configurações específicas de cada entidade (tabela) foram centralizadas na subpasta Configurations. Cada classe de configuração, como AulaConfiguration, CursoConfiguration, entre outras, implementa a interface IEntityTypeConfiguration<T>, permitindo descrever detalhadamente os atributos e regras de cada tabela — como Id, Descrição, campos obrigatórios e nomes de tabelas no banco. Já a subpasta Context armazena a classe RoomFlowContext, responsável por consolidar e aplicar todas essas configurações no método OnModelCreating, utilizando o método ApplyConfiguration. Essa abordagem permite uma separação clara entre entidades de domínio e regras de mapeamento com o banco, promovendo organização, escalabilidade e manutenibilidade no desenvolvimento da aplicação.

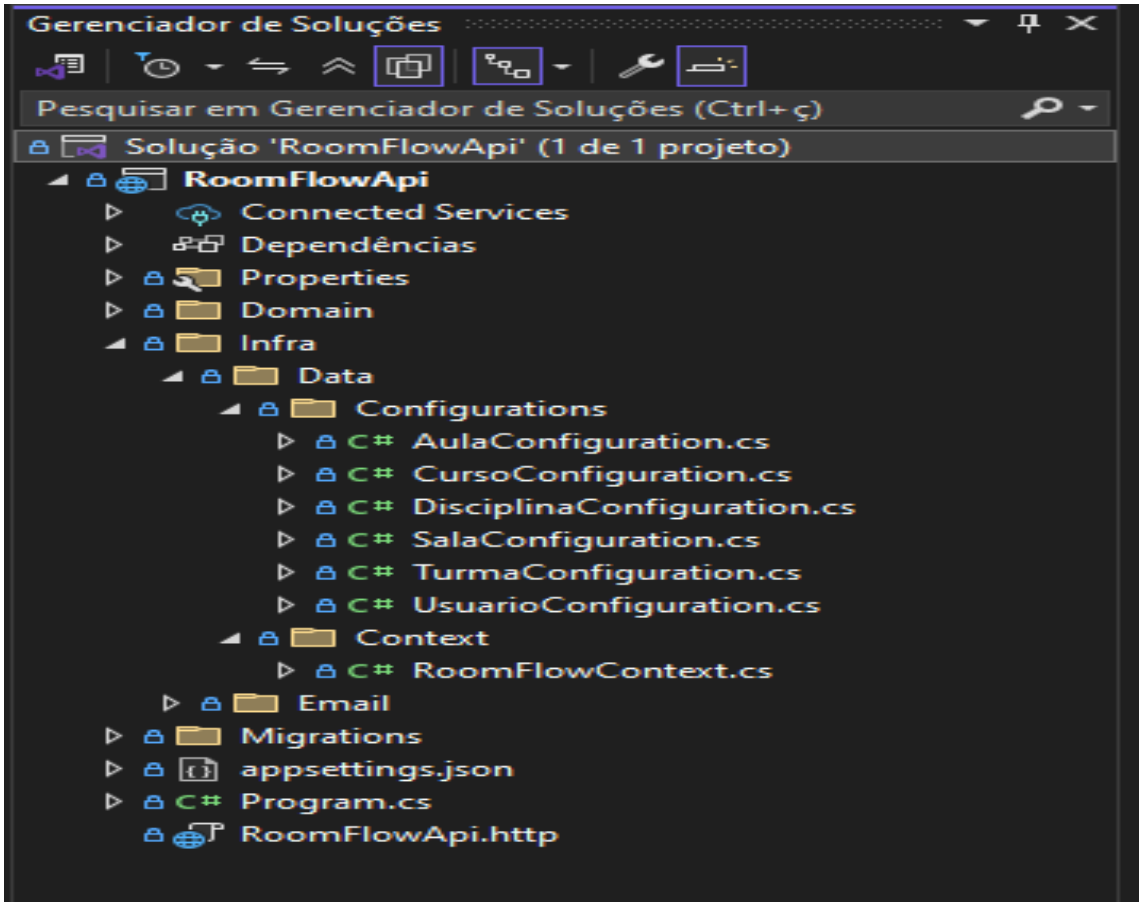

Figura 3- Pasta data, contendo as configurações para o Entity Framework

FONTE: Acervo Nosso

```
using Microsoft.EntityFrameworkCore.Metadata.Builders;
using RoomFlowApi.Domain.Entities;

namespace RoomFlowApi.Infra.Data.Configurations
{
    1 referência
    public class AulaConfiguration : IEntityTypeConfiguration<Aula>
    {
        0 referências
        public void Configure(EntityTypeBuilder<Aula> builder)
        {
            builder.HasKey(p => p.Id);
            builder.Property(p => p.Bloco)
                .IsRequired();
            builder.Property(p => p.SalaId)
                .IsRequired();
            builder.Property(p => p.DisciplinaId)
                .IsRequired();
            builder.Property(p => p.TurmaId)
                .IsRequired();
            builder.Property(p => p.Data)
                .IsRequired();
            builder.Property(p => p.ProfessorId)
                .IsRequired();
            builder.Property(p => p.CursoId)
                .IsRequired();

            builder.ToTable("TAB_Aula");
        }
    }
}
```

Figura 4- Exemplo de arquivo de configuração de tabela dentro da subpasta configurations

FONTE: Acervo Nosso

2.2.2-Swagger

De acordo com Souza e Silva (2021), o Swagger é uma ferramenta amplamente utilizada no desenvolvimento de APIs RESTful por proporcionar uma interface gráfica interativa que descreve os endpoints da API, suas rotas, parâmetros e métodos HTTP de forma clara e acessível. Ele não é exclusivo do ambiente .NET — podendo ser utilizado em diversas linguagens e frameworks — e tem como principal função facilitar a documentação, teste e integração das APIs durante o ciclo de desenvolvimento.

No contexto deste projeto, o Swagger funciona como uma "documentação viva" da API, permitindo que a equipe de backend disponibilize uma interface estruturada com todas as rotas e operações disponíveis, de forma que a equipe de frontend possa visualizar e interagir com esses endpoints, garantindo assim uma conexão eficaz entre as camadas do sistema.

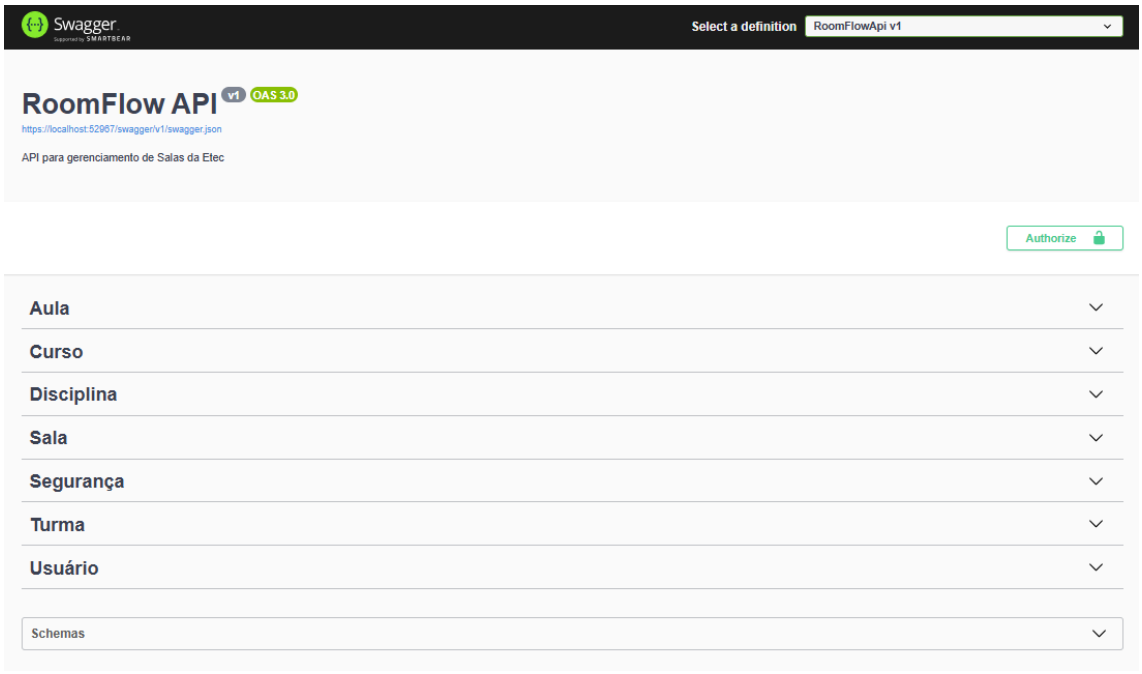

Figura 5- Swagger da API do RoomFlow

FONTE: Acervo Nosso

2.2.2.1-Controllers e Endpoints

Na estrutura de uma API RESTful em .NET, como a utilizada no sistema, os controllers são classes responsáveis por organizar os endpoints, que são os pontos de acesso à API. Cada controller agrupa as ações relacionadas a um recurso específico, como Curso, Aula ou Disciplina.

Os controllers no projeto estão organizados no arquivo Program.cs.

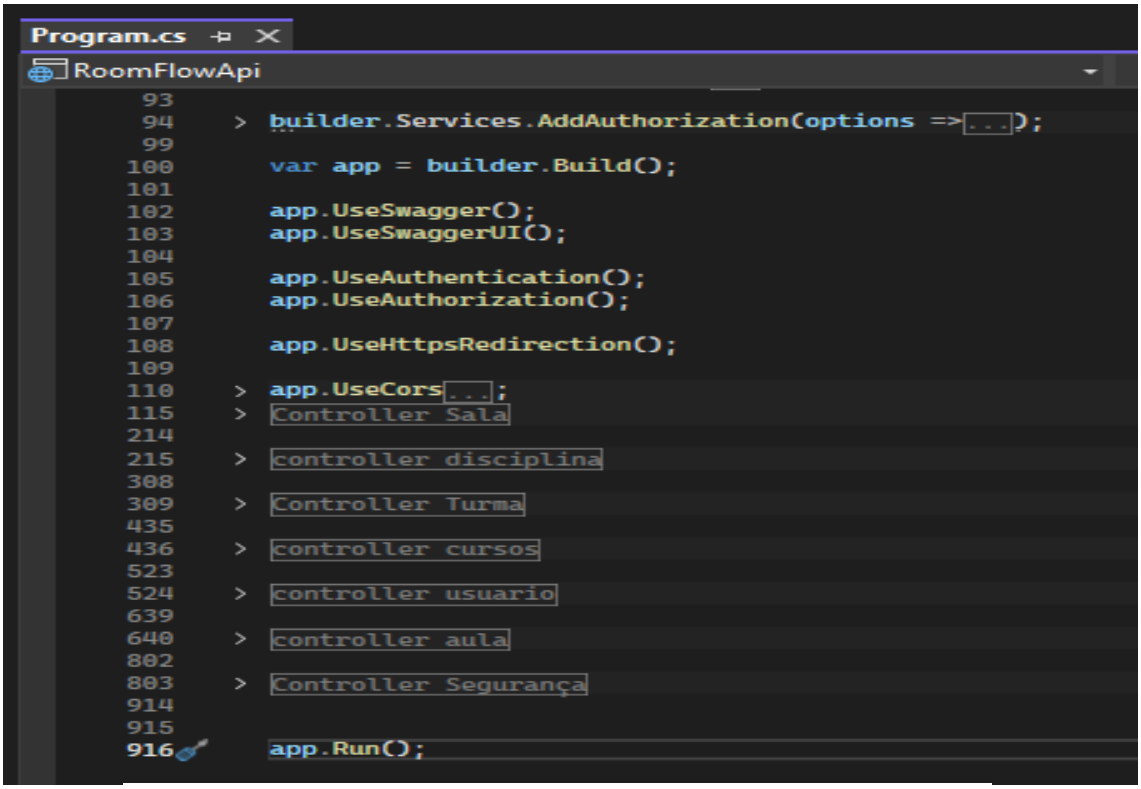

Figura 6- Print da localização dos controllers no projeto

FONTE: Acervo Nosso

Os endpoints correspondem aos métodos HTTP usados para interagir com os dados:

- GET para buscar informações,
- POST para criar novos registros,
- PUT para atualizar,
- DELETE para excluir.

Por exemplo, no CursoController, os endpoints permitem listar cursos (GET), adicionar um novo (POST), editar (PUT) ou remover (DELETE). Eles são essenciais para a comunicação entre o front-end e o back-end da aplicação.

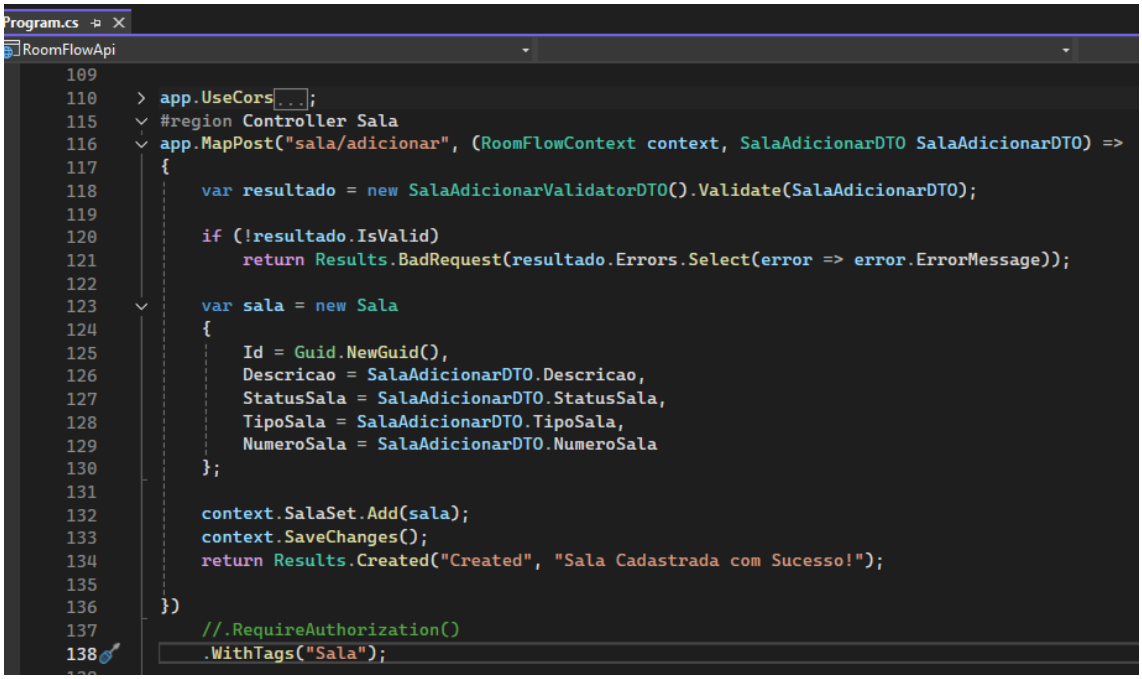

Figura 7- Exemplo de Endpoint dentro do projeto(sala/Adicionar)

FONTE: Acervo Nosso

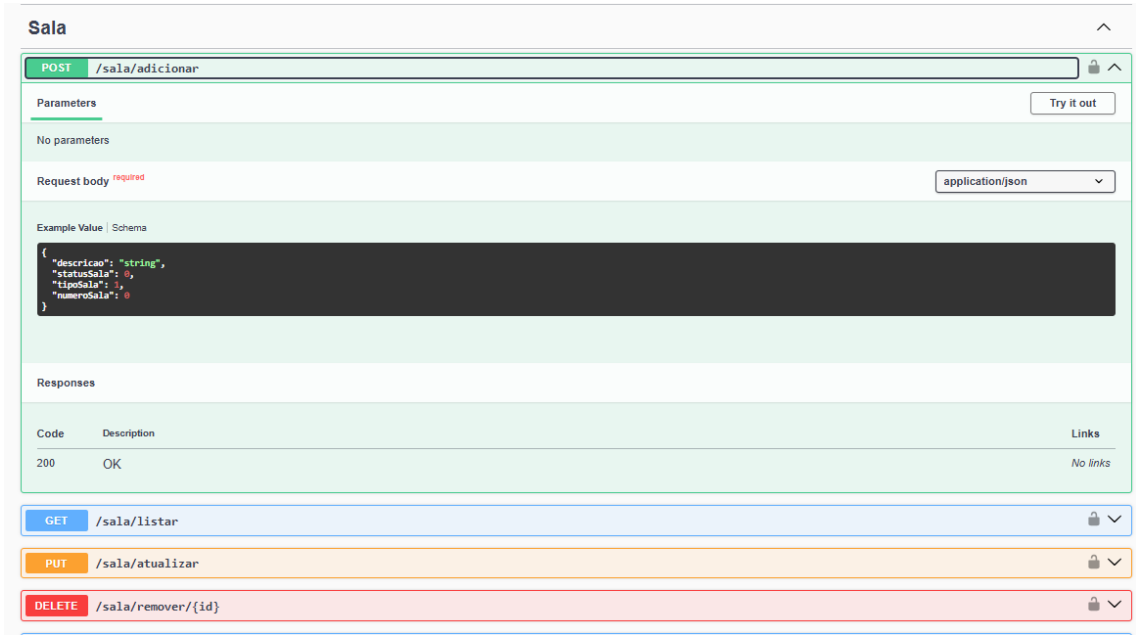

Figura 4-Exemplo de Endpoint visualizado no Swagger(Sala/Adicionar)

FONTE: Acervo Nosso

2.2.2.2- Squemas

Os **Schemas** representam os modelos de dados utilizados pelos métodos da API. Eles definem a **estrutura exata** que os dados devem seguir ao serem enviados ou recebidos por meio dos métodos CRUD de cada controller. Em outras palavras, os schemas funcionam como **“moldes”** que orientam o que deve ser informado — por exemplo, quais campos são obrigatórios, o tipo de dado esperado (texto, número, data) e o formato correto.

Esses dados são geralmente estruturados no formato **JSON (JavaScript Object Notation)**, um padrão leve e amplamente utilizado para a troca de informações entre sistemas. O JSON permite representar os dados de forma legível tanto para humanos quanto para máquinas, utilizando uma estrutura de chave e valor. Por exemplo, ao criar uma nova aula via API, o schema definirá quais propriedades devem estar no JSON enviado — como data, professorId, salaId, entre outras.

No Swagger, os schemas aparecem como uma documentação visual, facilitando a compreensão do que cada endpoint espera como entrada (POST, PUT) e o que retorna como saída (GET). Isso evita erros de comunicação entre as equipes de front-end e back-end, garantindo que os dados estejam no padrão correto.

Esses modelos são geralmente baseados nas **classes DTO (Data Transfer Object)**, que servem para transportar dados entre as camadas do sistema de forma organizada e segura.

Os arquivos baseados nas classes DTO estão localizados na pasta domain do projeto

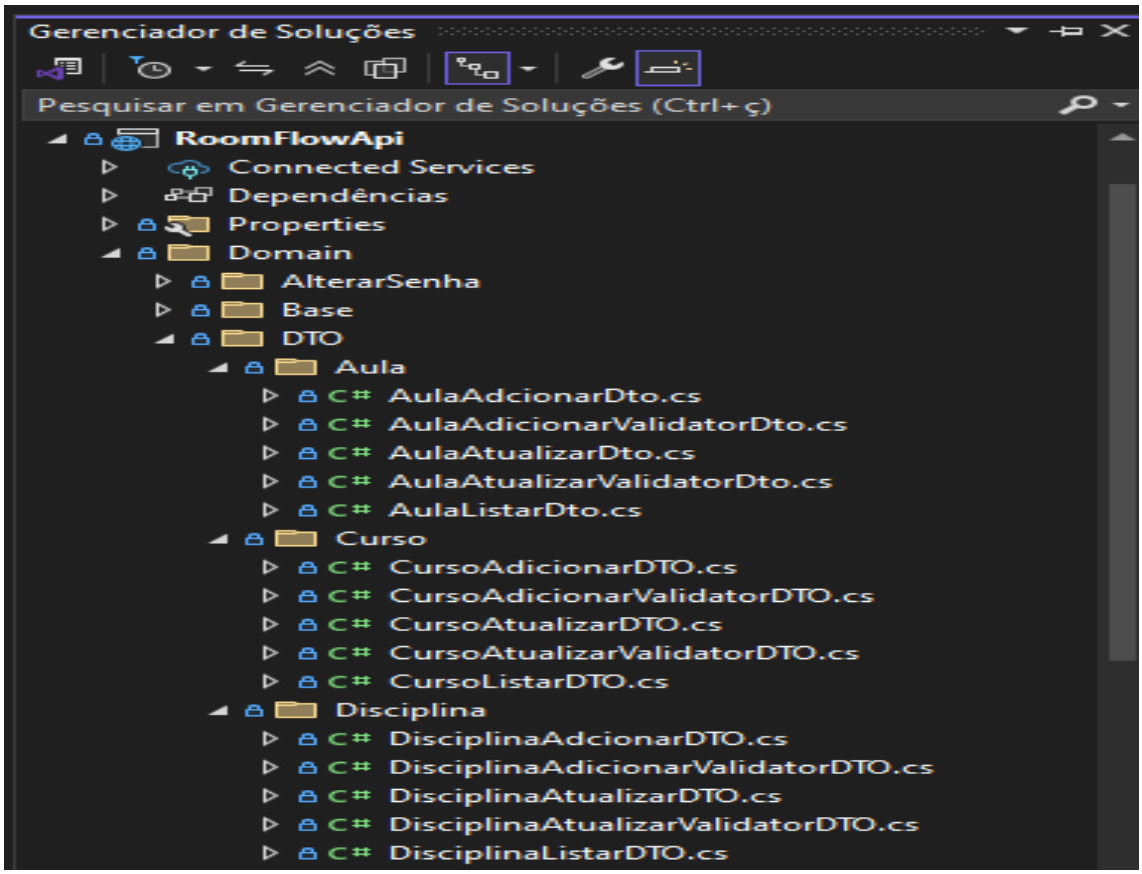

Figura 9 Localização dos DTO's no projeto

FONTE: Acervo Nosso

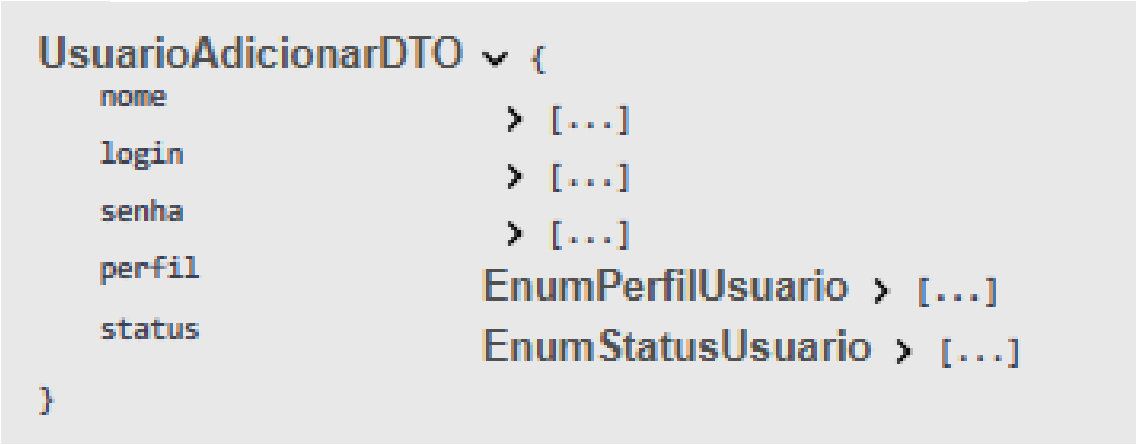

Figura 10 - Exemplo de visualização de Schema no Swagger(UsuarioAdicionarDTO)

FONTE: Acervo Nosso

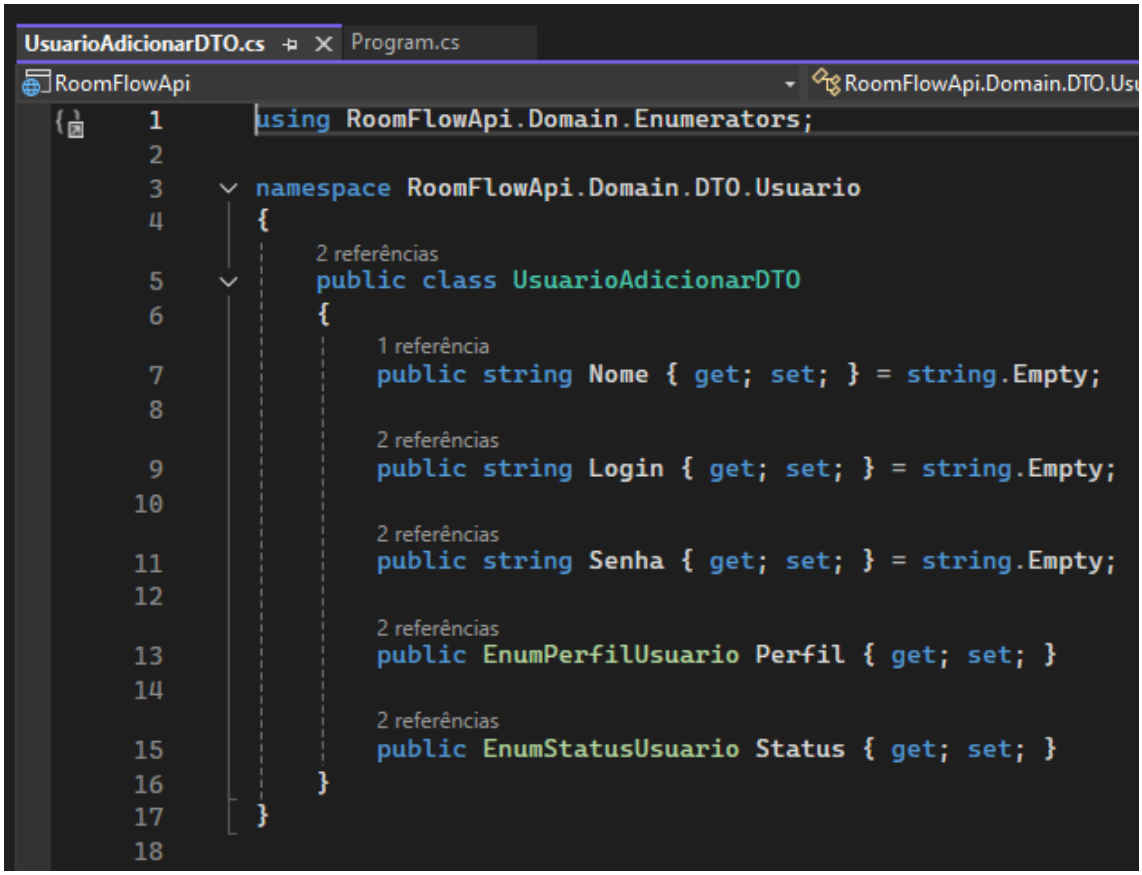

Figura 11- Exemplo de Schema dentro do projeto .net(UsuarioAdicionarDTO)

FONTE: Acervo Nosso

3-FrontEnd

3.1-Prototipagem

De acordo com Pressman (2016, p. 153),

“O paradigma da prototipação começa com a comunicação. Faz-se uma reunião com os envolvidos para definir os objetivos gerais do software, identificar quais requisitos já são conhecidos e esquematizar quais áreas necessitam, obrigatoriamente, de uma definição mais ampla. Uma iteração de prototipação é planejada rapidamente e ocorre a modelagem (na forma de um 'projeto rápido'). Um projeto rápido se concentra em uma representação daqueles aspectos do software que serão visíveis aos usuários finais (por exemplo, o layout da interface com o usuário ou os formatos de exibição na tela). O projeto rápido leva à construção de um protótipo, que é empregado e avaliado pelos envolvidos, que fornecerão um retorno (feedback), que servirá para aprimorar os requisitos.”

Com base nessa perspectiva, evidencia-se a importância da prototipagem no desenvolvimento do front-end do sistema. Ao representar visualmente telas, menus, formulários e elementos interativos, é possível validar precocemente a experiência do usuário e garantir que a interface esteja alinhada às expectativas dos usuários finais. Esse processo permite ajustes rápidos e eficazes, contribuindo para um sistema mais funcional, intuitivo e acessível

3.1.1- Figma

. De acordo com a Tech & Tunes (2023), o Figma tem se mostrado essencial no processo de prototipagem, pois permite a criação de protótipos interativos de alta fidelidade, que simulam com precisão o comportamento do produto final. Ele também oferece suporte à colaboração em tempo real, aos componentes reutilizáveis, além de bibliotecas e integração entre arquivos, promovendo eficiência e consistência no trabalho de design .

Por essas razões, utilizamos o Figma na prototipagem do nosso front-end, garantindo uma visualização precisa das telas e interação do sistema antes da implementação.

3.1.1.1 – Logos

Durante o processo de prototipagem com o Figma, foram desenvolvidas diferentes versões da logo do projeto, adaptadas para distintos contextos e ambientes do sistema, como telas iniciais, interfaces internas, ícones reduzidos e materiais de apresentação. A escolha por criar múltiplas variações visuais visa garantir **coerência estética**, **versatilidade de aplicação** e **adequação ao layout de cada tela**. O Figma facilitou esse processo ao permitir a criação de componentes reutilizáveis e o ajuste preciso de elementos gráficos, mantendo a identidade visual do projeto coesa em todas as versões.

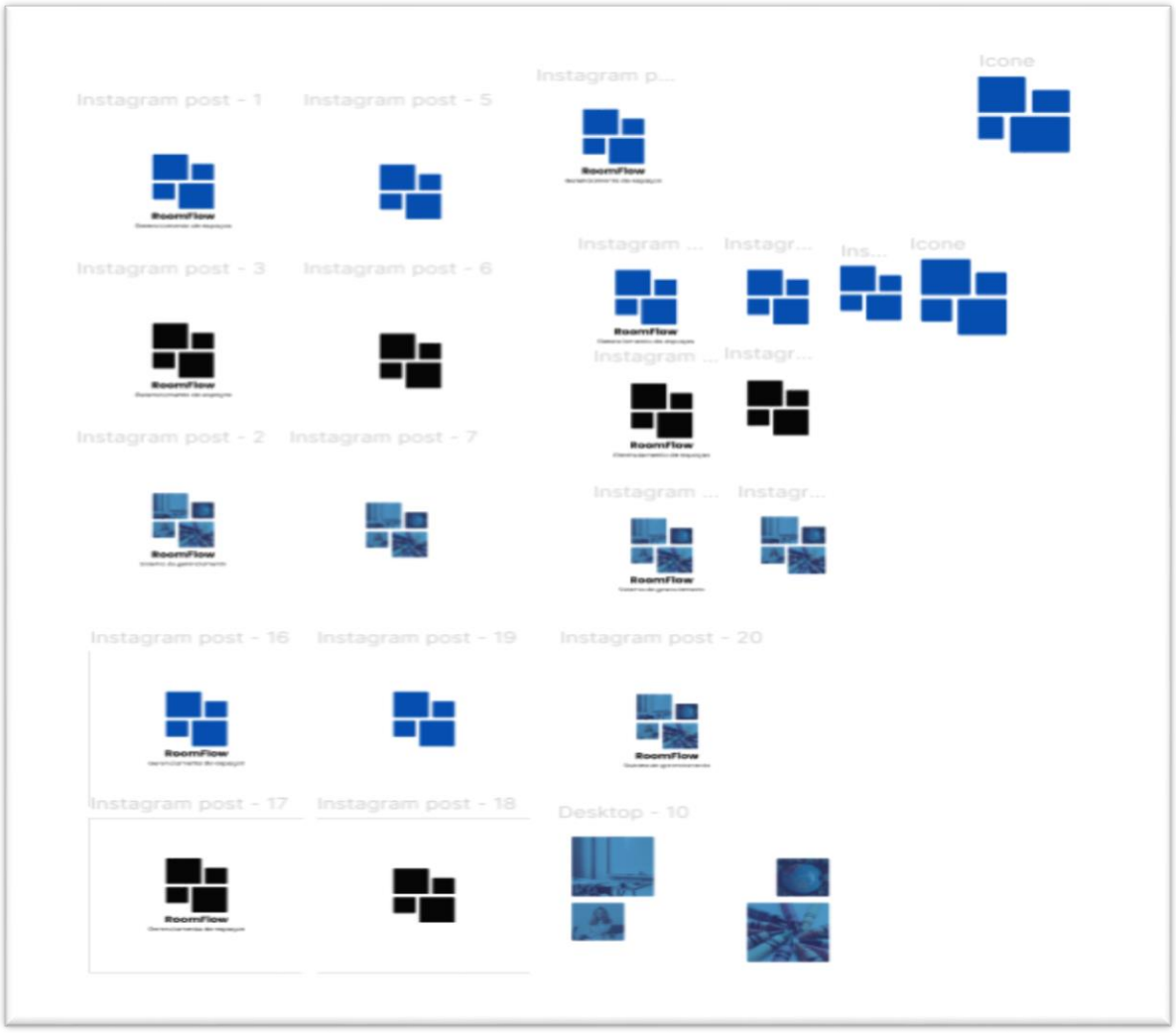

Figura 12- Logos do Roomflow Feitas no Figma

FONTE:Acervo Nosso

3.1.1.2- Telas

No processo de prototipagem realizado no Figma, foram desenvolvidas todas as telas que compõem a estrutura do sistema, desde o login até as interfaces principais de gerenciamento. O objetivo dessa etapa foi planejar visualmente a navegação, disposição dos elementos e funcionalidades, antes da implementação efetiva no front-end.

A imagem geral com todas as telas prototipadas permite uma visão completa do fluxo do sistema, evidenciando como as interfaces se conectam e interagem entre si. Além disso, algumas telas principais foram destacadas para apresentar em maior detalhe a organização visual, os elementos de usabilidade e a aplicação da identidade visual do projeto.

O uso do Figma para essa finalidade proporcionou maior agilidade, colaboração entre os membros da equipe e a possibilidade de realizar ajustes rápidos conforme as necessidades do projeto.

Figura 13-Telas Prototipadas no Figma

FONTE: Acervo Nosso

Dentre as telas desenvolvidas, destaca-se a **tela do Mapa**, que representa a interface principal visualizada pelo professor ao acessar o sistema. Essa tela centraliza as informações relacionadas à alocação das aulas, facilitando a visualização e o acompanhamento da grade de horários. Além dela, uma tela de grande relevância é a **tela do Gerador**, responsável por realizar a geração automática das aulas para o semestre, com base nas regras e dados previamente definidos no sistema.

Figura 14- Tela do Mapa de Salas no Figma

FONTE: Acervo Nosso

3.2- Aplicação Web

Segundo Pressman (2016), uma aplicação web é um tipo de software que é acessado por meio de navegadores e executado em servidores, oferecendo uma interface rica e interativa ao usuário, com foco em acessibilidade, atualização em tempo real e distribuição centralizada. Esse tipo de aplicação permite que o usuário interaja com os dados de forma dinâmica, sem a necessidade de instalar nenhum programa localmente.

No contexto deste projeto, a aplicação web desenvolvida em Angular se comunica com uma API RESTful escrita em C#, responsável pelo gerenciamento das regras de negócio e acesso ao banco de dados. Essa separação em camadas garante escalabilidade, manutenção facilitada e uma melhor divisão de responsabilidades no sistema.

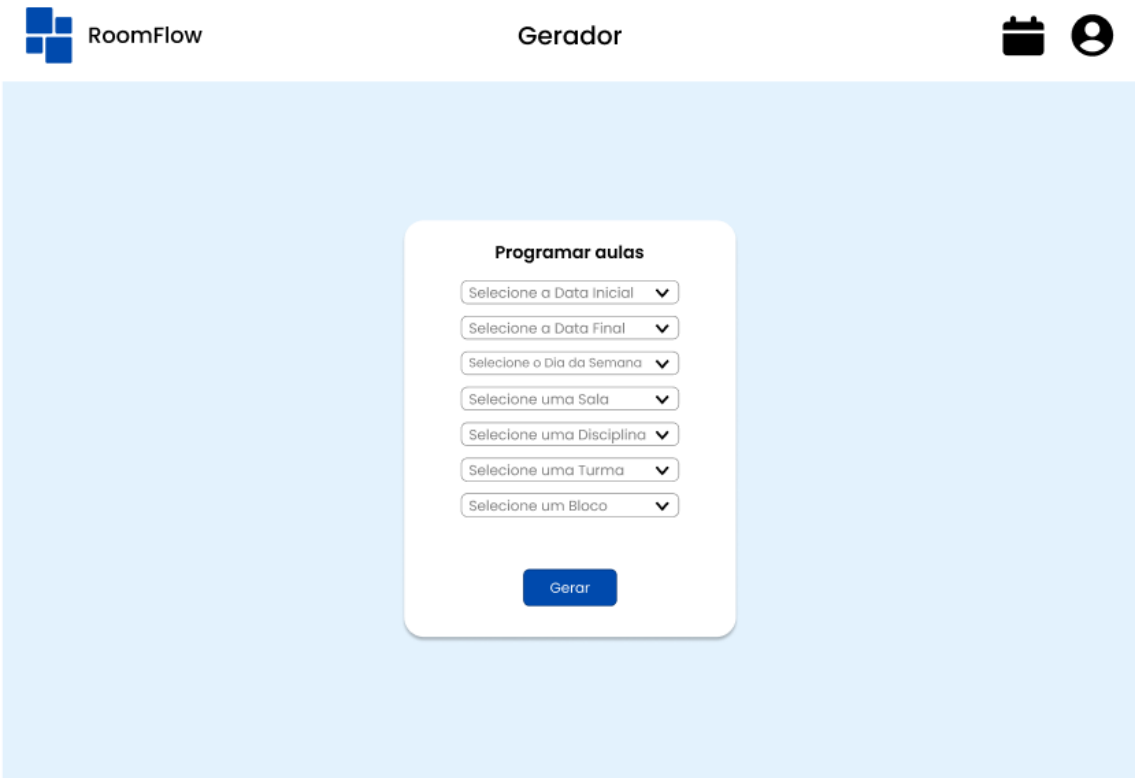

Figura 15- Tela do Gerador no Figma

FONTE: Acervo Nosso

3.2.1 HTML

Segundo Castro (2013), o **HTML (HyperText Markup Language)** é a linguagem de marcação utilizada para estruturar o conteúdo das páginas web. Com ele, é possível organizar informações por meio de elementos como títulos, parágrafos, listas, tabelas, imagens, links e formulários, estabelecendo a base da interface visual de qualquer site ou aplicação web.

No projeto desenvolvido, o HTML foi utilizado na construção das telas da aplicação web, servindo como alicerce para a disposição dos componentes visuais. Cada elemento da interface, como campos de formulário, botões e seções informativas, foi definido por meio de marcações HTML, proporcionando clareza na estruturação do conteúdo.

Contudo, sem a utilização do CSS, a tela permanece com uma aparência crua e desorganizada, o que compromete a usabilidade e a experiência do usuário. O HTML define a estrutura, mas é o CSS que estiliza e posiciona os elementos na tela de forma adequada, tornando a interface visualmente atrativa e funcional.

src > app > components > pages > principal > principal.component.html > app-nav-bar

```
<app-nav-bar
  [textoNav]="'Painel de Monitoramento'">
</app-nav-bar>
<div class="corpo">
  <mat-card>
    <mat-card-header>
      <div class="status">
        <div class="status-item">
          <span class="status-icon disponível"></span> Disponível
        </div>
        <div class="status-item">
          <span class="status-icon reservado"></span> Ocupada
        </div>
        <div class="status-item">
          <span class="status-icon indisponível"></span> Indisponível
        </div>
      </div>
    </mat-card-header>

    <mat-card-content>
      <div class="cards-container">
  <div *ngFor="let map of mapa">
    <app-cards-sala
      [mapa]="map"
      [saladesc]="map.descricao"
      [mostrarEditarBotao]="false"
      [mostrarExcluirBotao]="false"
      [botaoReservarSala]="authService.usuarioEhProfessor() ? true : false"
      >
    </app-cards-sala>
  </div>
</div>
```

Figura 16- Exemplo de Código em HTML do Mapa de salas

FONTE: Acervo Nosso

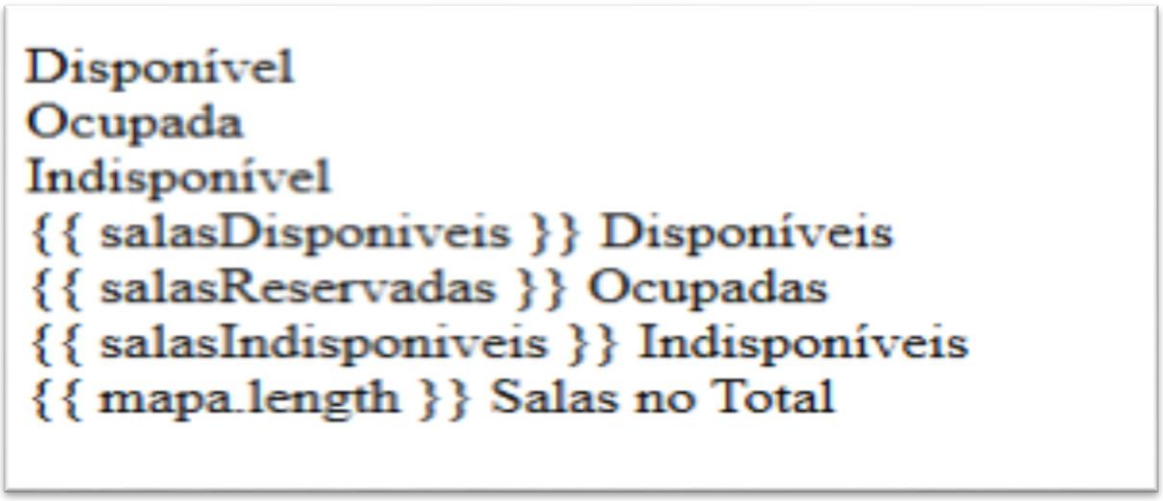

Figura 17- Visualização do codigo HTML do Mapa de salas sem a formatação de CSS

FONTE: Acervo Nosso

3.2.2- CSS

Segundo Castro (2013), **CSS (Cascading Style Sheets)** é a linguagem responsável pela estilização das páginas web, permitindo definir a aparência dos elementos estruturados em HTML. Com o CSS, é possível aplicar cores, margens, fontes, tamanhos, posicionamentos e diversos outros estilos visuais, oferecendo uma experiência visual agradável ao usuário.

No desenvolvimento da aplicação web do projeto, o CSS foi fundamental para tornar as interfaces mais atrativas e organizadas. Através dele, foram definidos padrões visuais para botões, menus, formulários e outros componentes da interface, garantindo uniformidade e coerência visual em todas as telas.O uso do CSS contribui diretamente para a usabilidade do sistema, pois sem ele, a interface permaneceria com aspecto cru e pouco intuitivo, dificultando a navegação e compreensão das funcionalidades. Além disso, técnicas como media queries permitiram a adaptação da aplicação a diferentes tamanhos de tela, tornando-a responsiva.

src > app > components > pages > principal > # principal.component.css > .cards-container

```css
.corpo {
  display: flex;
  flex-direction: column;
  min-height: 100vh;
  box-sizing: border-box;
  align-items: center;
  background-color: rgb(145, 166, 189);
  padding: 40px;
  overflow-x: hidden;
  overflow-y: auto;
}

.grid {
  display: grid;
  grid-template-columns: repeat(10, 1fr);
  grid-gap: 10px;
  width: 75%;
  border: 2px solid rgb(255, 255, 255);
  background-color: white;
  border-radius: 15px;
  padding: 15px;
  justify-items: center;
  margin-top: 20px;
}

.cards-container {
  display: grid;
  grid-template-columns: repeat(12, 1fr);
  gap: 16px;
  justify-items: center;
  width: 100%;
}
```

Figura 18- Exemplo do Código em CSS do Mapa de Salas

FONTE: Acervo Nosso

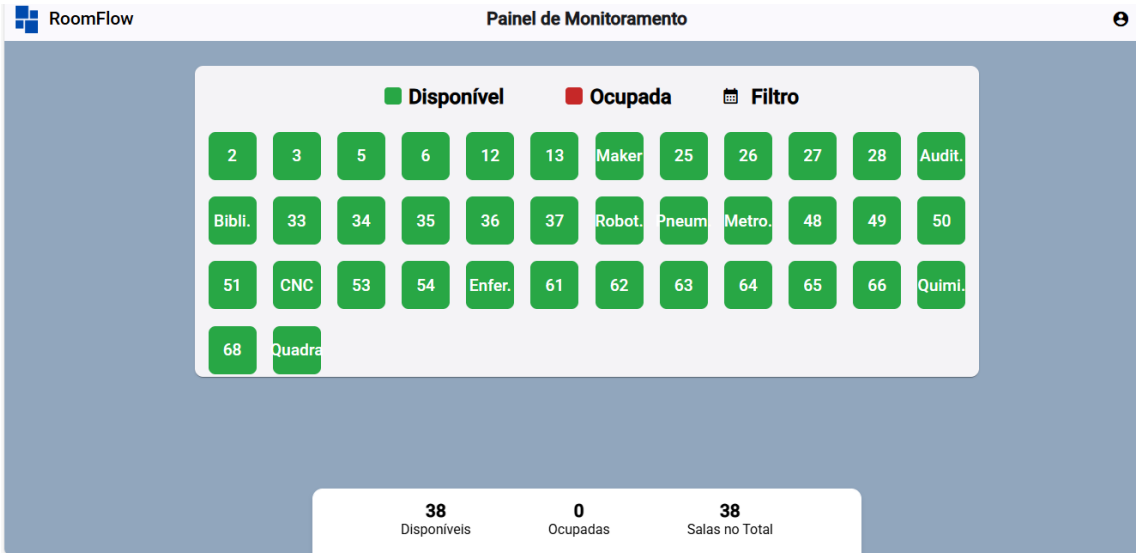

Figura 19-Exemplo do Mapa de salas com a estilização CSS

FONTE: Acervo Nosso

3.2.3 Javascript e typescript

Segundo Flanagan (2020), **JavaScript** é uma linguagem de programação essencial para o desenvolvimento web, responsável por adicionar interatividade e dinamismo às páginas. Ele permite manipular elementos da interface em tempo real, validar formulários, reagir a eventos do usuário e muito mais.

Já o **TypeScript** é um superset (ou superconjunto) do JavaScript desenvolvido pela Microsoft, que adiciona tipagem estática e recursos de orientação a objetos à linguagem. Isso permite maior robustez no desenvolvimento, facilitando a identificação de erros durante a codificação e melhorando a legibilidade e manutenção do código.

No projeto desenvolvido, ambos foram utilizados em conjunto dentro do framework Angular. O **TypeScript** foi predominante, oferecendo uma estrutura mais segura e organizada para o código da aplicação. Ele também facilitou a criação de componentes, serviços e integração com a API desenvolvida em C#.

O uso dessas linguagens foi crucial para tornar a aplicação web funcional, interativa e eficiente, promovendo uma boa experiência ao usuário final.

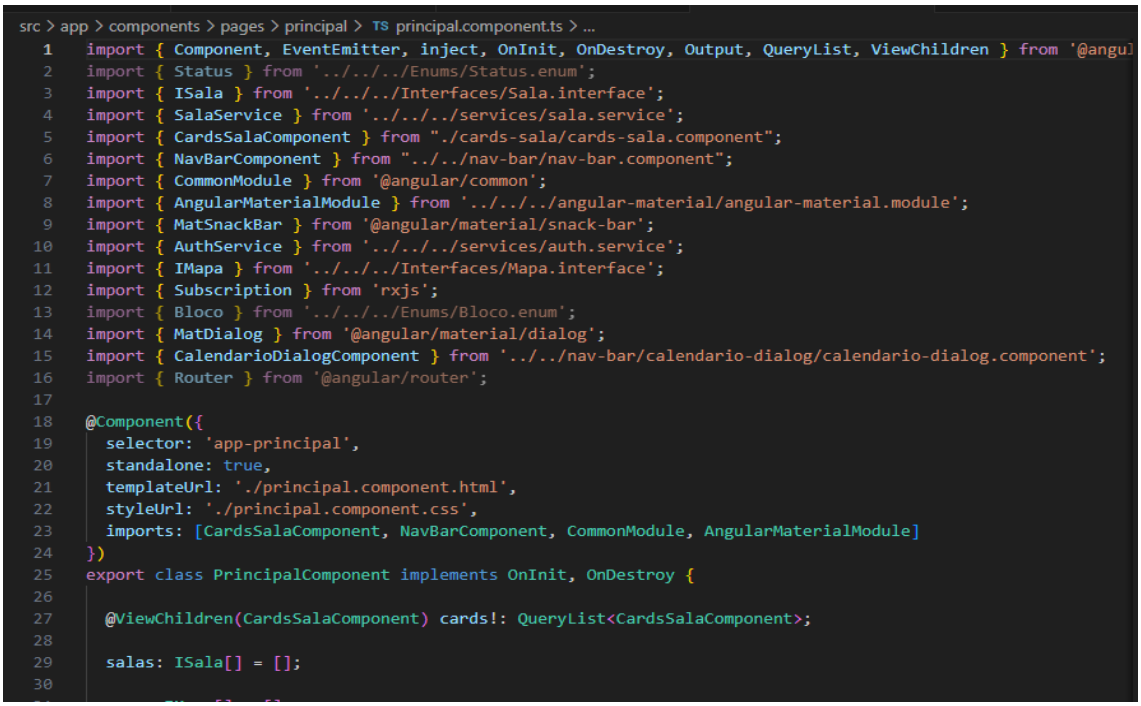

Figura 20- Exemplo de Código em Typescript da tela do Mapal

FONTE: Acervo Nosso

3.2.4 Utilização do Framework Angular no projeto

Segundo Freeman (2018), **Angular** é um framework front-end de código aberto mantido pelo Google, desenvolvido com base em TypeScript. Ele permite a criação de aplicações web dinâmicas e escaláveis com uma arquitetura baseada em componentes, promovendo a reutilização de código, modularidade e melhor organização do projeto.

No desenvolvimento da aplicação RoomFlow, o Angular foi essencial para estruturar a interface do usuário de forma reativa e eficiente. Por meio dele, foi possível criar componentes reutilizáveis para cada parte da interface — como o mapa de salas, o gerador de aulas e as telas de autenticação.

Além disso, o Angular facilita a integração com a API RESTful desenvolvida em C#, por meio do uso de serviços que consomem os dados e os apresentam ao usuário. Isso promove uma separação clara entre lógica de apresentação e regras de negócio, contribuindo diretamente para a manutenibilidade e escalabilidade da aplicação.

3.2.4.1 Angular-Material

O **Angular Material** é uma biblioteca de componentes visuais desenvolvida pela equipe do Angular, baseada nas diretrizes do Material Design criado pela Google. Seu objetivo é fornecer um conjunto coeso e moderno de elementos de interface, como botões, menus, caixas de diálogo, abas, formulários e tabelas, garantindo acessibilidade, responsividade e uma experiência consistente ao usuário (FREEMAN, 2018).

Durante o desenvolvimento da aplicação RoomFlow, o Angular Material foi adotado para padronizar e agilizar a criação da interface gráfica. Com ele, foi possível aplicar uma identidade visual profissional ao sistema, garantindo que todos os elementos seguissem o mesmo padrão estético e de usabilidade.

Além disso, o uso do Angular Material possibilitou a rápida implementação de funcionalidades interativas e responsivas, contribuindo para que a aplicação se adaptasse bem a diferentes tamanhos de tela, como desktops e tablets, e oferecendo uma experiência fluida ao usuário final.

REFERÊNCIAS

ALBAHARI, Joseph; ALBAHARI, Ben. C# 9.0 e .NET 5: Guia de bolso. 8. ed. São Paulo: Novatec, p. 5-18, 2021.

CASTRO, Elizabeth. HTML e CSS: design e construção de sites. São Paulo: Novatec, 2013.

DATE, C. J. Introdução a sistemas de banco de dados. 7. ed. Rio de Janeiro: Campus, 2000.

ESPOSITO, Dino. Programação com ASP.NET Core: desenvolvendo aplicações web modernas com C# e .NET 5. São Paulo: Novatec, p. 55-80, 2021.

FILHO, M. José. Pesquisas: contornos no processo educativo. Franca: Unesp-FHDSS, 2006.

FLANAGAN, David. JavaScript: o guia definitivo. 7. ed. São Paulo: Bookman, 2020.

FREEMAN, Adam. Angular: desenvolvimento de aplicações web. 2. ed. São Paulo: Novatec, 2018.

GONDIM, Sônia Maria Guedes; SILVA, Narbal. Motivação no trabalho. Psicologia, organizações e trabalho no Brasil. Porto Alegre: Artmed, p. 145-176, 2004.

LAKATOS, Eva Maria; MARCONI, Marina de Andrade. Técnicas de pesquisa. 3ª edição. São Paulo: Editora Atlas, 1996.

LERMEN, Fabrício; PEREIRA, Felipe. Entity Framework Core Essencial. São Paulo: Casa do Código, 2021.

PRESSMAN, Roger S. Engenharia de software: uma abordagem profissional. 8. ed. Porto Alegre: AMGH, p. 135-140, 2016.

PRESSMAN, Roger S. Engenharia de software: uma abordagem profissional. 8. ed. Porto Alegre: AMGH, p. 153-156, 2016.

SOUZA, A. F.; SILVA, R. M. Documentação de APIs com Swagger: teoria e prática no desenvolvimento de sistemas. São Paulo: Novatec, p. 45-62, 2021.

TECH & TUNES. A importância do FIGMA na prototipagem de designs: conheça suas vantagens. Disponível em: https://techtunes.com.br/a-importancia-do-figma-na-prototipagem. Acesso em: 07 jun. 2025.

VERNA, Sébastien. RESTful Web APIs. 2. ed. O'Reilly Media, 2021