



**CENTRO DE CIÊNCIAS E TECNOLOGIAS
LABORATÓRIO DE CIÊNCIAS MATEMÁTICAS**

Davi Rodrigues Soares Machado

**Projeto da disciplina Paradigmas Orientados à Objetos para
Desenvolvimento de Software:
Sistema de Agenda Escolar**

Campos dos Goytacazes - RJ

14 de setembro de 2025

Sumário

1	Introdução	2
2	Objetivos	3
2.1	Público Alvo	3
2.2	Orçamento	4
3	Levantamento de Requisitos	5
3.1	Requisitos Funcionais	5
3.2	Requisitos Não Funcionais	6
4	Diagramas	7
4.1	Diagrama de Caso de Uso	7
4.2	Diagrama UML	8
4.3	Diagrama Entidade Relacionamento	9
5	Arquitetura do Sistema	10
6	Tecnologias Utilizadas	11
7	Cronograma	12

Capítulo 1

Introdução

A proposta deste projeto consiste no desenvolvimento de um sistema de agenda escolar voltado para atender às demandas de uma rede de escolas. O objetivo central é oferecer uma ferramenta que permita a alunos, professores e outros profissionais da instituição visualizar e acompanhar informações essenciais, como distribuição de salas, organização de turmas, docentes responsáveis, além de outras funcionalidades que serão apresentadas com maior clareza na seção de [Diagramas](#).

Esse projeto servirá como simulação prática de desenvolvimento de um software, para a disciplina Paradigmas Orientados à Objetos para Desenvolvimento de Software (POODev), possibilitando a aplicação de conceitos aprendidos em disciplinas como Programação Orientada à Objetos, Banco de Dados e Análise e Projeto de Sistemas. Além disso, o desenvolvimento será registrado e versionado dentro da plataforma GitHub, no [Repositório Github](#), onde será possível acompanhar a evolução do sistema, de acordo com o [Cronograma](#).

Capítulo 2

Objetivos

O projeto tem como objetivo central a organização da agenda escolar virtual de uma rede de escolas, integrando informações como alocação de salas, informações sobre disciplinas, professores e horários em uma única plataforma. O sistema também contará com recursos para os professores lançarem notas dos alunos, sendo possível gerar um relatório onde cada aluno poderá acompanhar suas notas de cada bimestre. O sistema de notas poderá gerar boletins escolares bimestrais e anuais para um melhor acompanhamento das respectivas notas.

Principais funcionalidades

Automatização da grade horária: Com o objetivo de reduzir os possíveis erros humanos na alocação de salas e professores, o sistema ficará responsável pela melhor distribuição dentro de cada unidade escolar, auxiliado do coordenador de cada unidade escolar.

Oferecer consultas personalizadas: Cada usuário poderá acessar o sistema com o intuito de visualizar a agenda escolar baseada em filtros como: por aluno, professor, turma ou sala. Será possível acompanhar o cronograma escolar e acessar boletins de alunos, que serão gerados automaticamente após o professor responsável lançar as notas das disciplinas.

Acesso disponível em multi plataformas: O sistema poderá ser acessado por qualquer dispositivo com acesso a internet via navegador web ou mobile.

2.1 Público Alvo

O público-alvo do sistema é formado por todos os membros que participam do processo acadêmico e administrativo de uma escola, cada um com diferentes níveis de uso e necessidades específicas. Os coordenadores de curso serão um dos principais perfis responsáveis por montar e gerenciar a grade horária, alocar professores e organizar a distribuição de salas. Os professores, utilizarão o sistema para consultar suas disciplinas, horários e salas atribuídas, garantindo maior clareza na sua rotina de trabalho. Já os alunos terão acesso a própria agenda, com a possibilidade

de acompanhar as disciplinas e horários de forma prática e acessível. Desse modo, o sistema visa atender de forma integrada os diferentes tipos de usuários, gerando uma maior organização e agilidade.

2.2 Orçamento

No orçamento abaixo foram levados em consideração o período de quatro meses de desenvolvimento do sistema, somado com o período de uso de um ano seguido.

Orçamento para Desenvolvimento				
Descrição	Quantidade	Valor por mês	Quantidade de meses	Total
Equipe de Desenvolvimento - Valores Fixos				
Desenvolvedor Frontend	1	5000	3	15000
Desenvolvedor Backend	1	5000	3	15000
Designer UX/UI	1	4500	4	18000
QA / Tester	1	4000	2	8000
Gerente de Projeto	1	5500	4	22000
Despesas após lançamento - Valores Variáveis				
Servidor em nuvem	1	500	12	6000
Administração do Banco de Dados	1	200	12	2400
Serviços de notificação	1	100	12	1200
Manutenção do software	2	1000	12	12000
Custo total para Desenvolvimento	78000			
Custo anual para manter sistema por ano	21600			

Figura 2.1: Orçamento

Capítulo 3

Levantamento de Requisitos

Durante a primeira reunião, foram levantados alguns requisitos que foram listados abaixo, com a possibilidade do sistema de assinatura digital.

3.1 Requisitos Funcionais

- RF01: Cadastrar professores, alunos, salas e disciplinas e unidades escolares.
- RF02: O login deve ser através de email e senha.
- RF03: O sistema deve atribuir diferentes níveis de permissão baseado no tipo de usuário
- RF04: Cada sala deve possuir informações de localização, capacidade e tipo
- RF05: O sistema deve alertar quando houver alguma incompatibilidade
- RF06: Cada usuário deve ter a capacidade de visualizar os relatórios referentes ao seu nível (ex. alunos - boletins e horario escolar; professores - turmas e horários)
- RF07: O sistema deve permitir exportar os relatórios em pdf
- RF08: Consultar grade horária por aluno, professor ou sala.
- RF09: Alocar salas a disciplinas em horários específicos.
- RF10: Impedir conflitos de agendamento.
- RF11: Gerir notas dos alunos, sendo possivel gerar relatorios com situaao de aprovado ou reprovaado ou registro de notas.
- RF12: Sistema de assinatura digital dos professores e coordenadores.

3.2 Requisitos Não Funcionais

- RNF01: O sistema deve ser responsivo e rápido.
- RNF02: O sistema deve apresentar telas intuitivas com navegação simples.
- RNF03: O sistema deve suportar pelo menos 200 usuários simultâneos.
- RNF04: O sistema deve criptografar senhas dos usuários.
- RNF05: O sistema deve implementar níveis de permissão de acordo com o perfil do usuário.
- RNF06: O sistema deve ser capaz de aumentar a capacidade de usuários e dados sem degradação significativa de desempenho.

Capítulo 4

Diagramas

4.1 Diagrama de Caso de Uso

Seguindo o diagrama abaixo, temos os principais casos de uso do sistema de agenda escolar. Começamos com as entidades: Aluno, Professor, Coordenador, Secretário e Pró Reitor, em que cada uma terá acesso a funcionalidades específicas dentro do sistema. Por exemplo, somente os professores poderão lançar e editar as notas dos alunos.

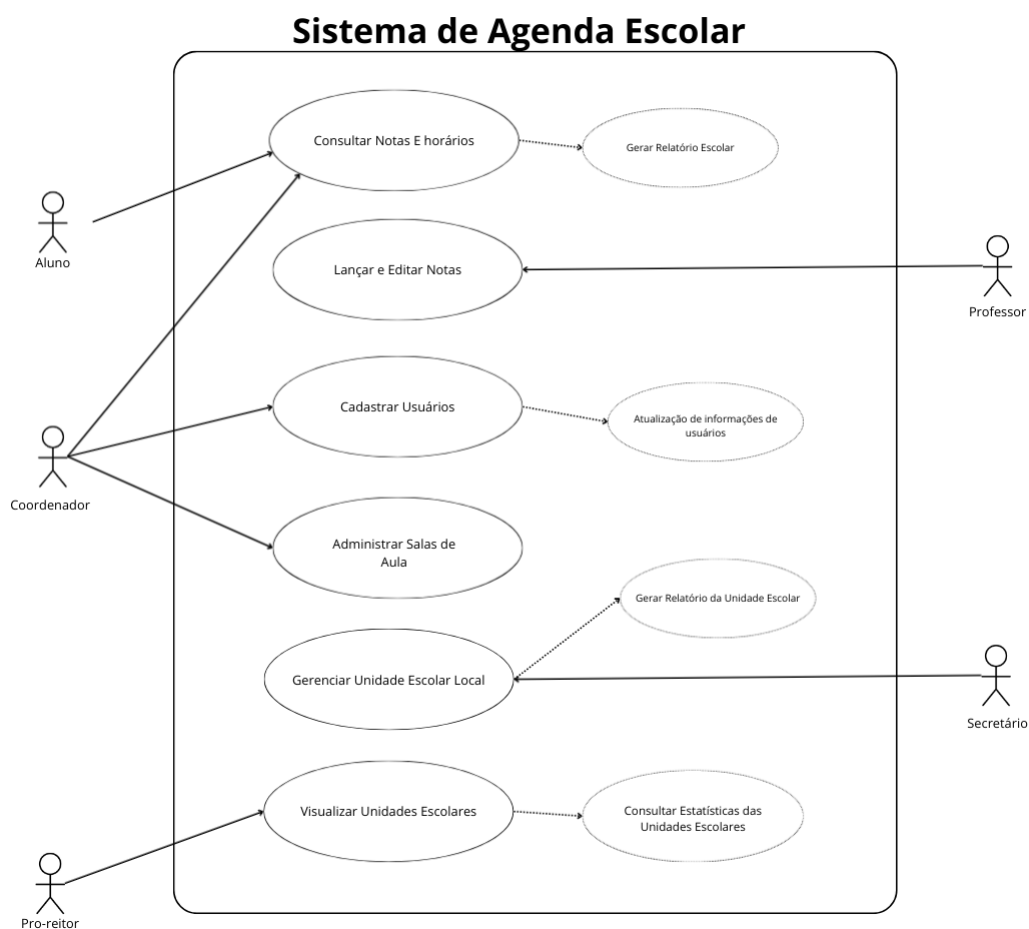


Figura 4.1: Diagrama de Casos de Uso

4.2 Diagrama UML

No diagrama UML, cada uma das classes é projetada com seus respectivos atributos e métodos.

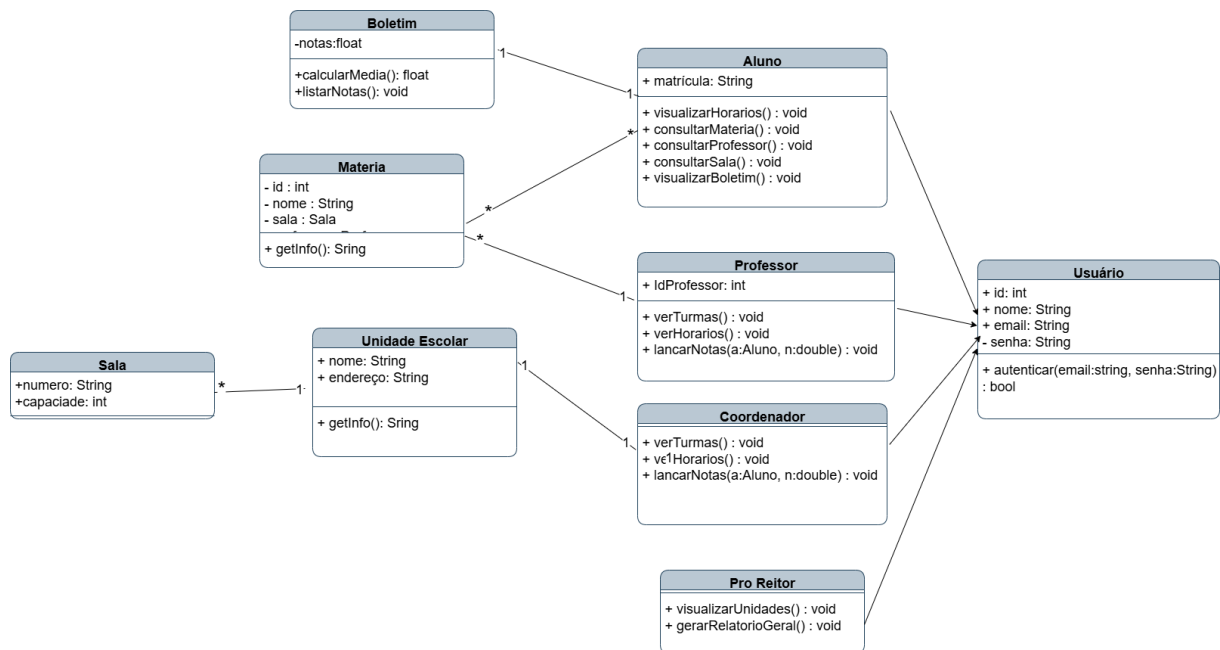


Figura 4.2: Diagrama UML

4.3 Diagrama Entidade Relacionamento

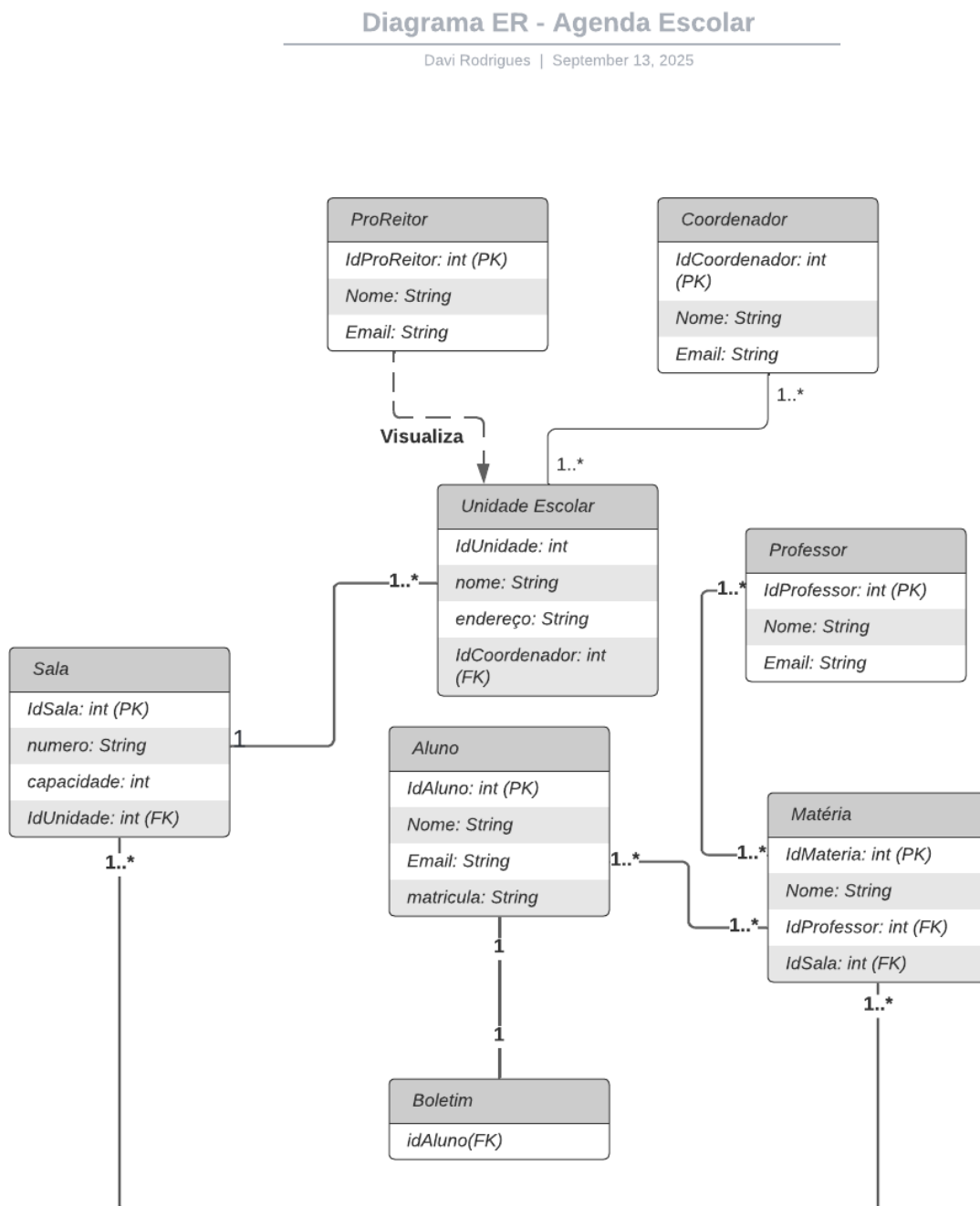


Figura 4.3: Diagrama Entidade-Relacionamento

Capítulo 5

Arquitetura do Sistema

Visão geral O sistema foi projetado com base na arquitetura cliente-servidor, onde a aplicação é dividida em dois grandes blocos: a interface de apresentação (front-end) e a lógica de negócio (back-end). Esses blocos se comunicam através de uma API REST, permitindo maior flexibilidade, escalabilidade e manutenção. O banco de dados relacional é utilizado para garantir a consistência das informações do sistema.

Fluxo de comunicação Basicamente, Quando um usuário estiver acessando o sistema, o front-end ficará responsável pelas interações com o usuário. Quando o usuário fizer uma solicitação, o front-end vai enviar requisições ao back-end por meio de protocolos http. O back-end vai processar essas informações, consultando o banco de dados se necessário, e retorna ao front-end um arquivo JSON, que será visualizado pelo usuário como forma de resposta a solicitação.

Capítulo 6

Tecnologias Utilizadas

- **Back-end:** A linguagem escolhida para o desenvolvimento foi a linguagem Python.
- **Front-end:** O front-end servirá como ferramenta de visualização para o sistema e será desenvolvido principalmente com React e Tailwind CSS.
- **Banco de Dados:** PostgreSQL local
- **Controle de Versão:** Todo o processo será documentado e versionado dentro da plataforma github, que poderá ser acessada através do [Repositório Github](#)

Capítulo 7

Cronograma

Para melhor entendimento, o cronograma de desenvolvimento foi dividido em dois bimestres, sendo o primeiro focado em backend e integração com banco de dados, e o segundo focado no frontend e nos testes de aplicação. Em ambas etapas, no final de cada semana do mês, uma reunião com cliente irá acontecer, onde será mostrado cada nova parte incluída no sistema.

Calendário de Entrega do Sistema							
Setembro				Outubro			
Semana 2 (07/09 - 13/09)	Semana 3 (14/09 - 20/09)	Semana 4 (21/09 - 27/09)	Semana 5 (28/09 - 04/10)	Semana 1 (05/10 - 11/10)	Semana 2 (12/10 - 18/10)	Semana 3 (19/10 - 25/10)	Semana 4 (26/10 - 01/11)
Planejamento e Levantamento de requisitos	Apresentação da Proposta e Arquitetura inicial	Apresentação da Documentação do Projeto	Ajustes na modelagem e na documentação do sistema	Implementação do Backend da aplicação			
Reunião com Cliente	Semana P1 Pesquisa Operacional		Semana P1 Compiladores				Reunião com Cliente
			Criação do Protótipo				
			Reunião com Cliente				

Figura 7.1: Cronograma do Primeiro Bimestre de Desenvolvimento

Calendário de Entrega do Sistema					
Novembro				Dezembro	
Semana 1 (02/11 - 08/11)	Semana 2 (09/11 - 15/11)	Semana 3 (16/11 - 22/11)	Semana 4 (23/11 - 29/11)	Semana 1 (30/12 - 05/12)	Semana 2 (07/12 - 13/12)
Implementação do Frontend da aplicação			Testes Funcionais e de uso	Demonstração do software e documentação do sistema	
	<i>Semana Projeto Compiladores</i>		<i>Semana P2 Compiladores</i>		
<i>Semana P2 Pesquisa Operacional</i>			Reunião com Cliente	<i>Semana P3 Pesquisa Operacional</i>	

Figura 7.2: Cronograma do Segundo Bimestre de Desenvolvimento