



CURSO DE BACHARELADO EM ENGENHARIA DE SOFTWARE

VITOR HUGO OLIVEIRA PALOCO

MARCELO YUKIO YAMAMOTO

PLANNER UNIFIL (SISTEMA DE HORÁRIOS)

Londrina
2025

SUMÁRIO

1	AGRADECIMENTOS	2
2	INTRODUÇÃO	3
3	DESCRIÇÃO DO SISTEMA	4
4	MODELAGEM DO SISTEMA	5
4.1	Diagrama de caso de uso	5
4.2	Diagrama de classe	5
4.3	Modelo Entidade-Relacionamento (MER)	6
4.4	Diagrama Entidade-Relacionamento (DER)	7
4.5	Diagrama de Implantação	7
5	DIAGRAMAS DE SEQUÊNCIA	8
5.1	Gerenciar Usuários	8
5.2	Gerenciar Matérias	9
5.3	Gerenciar Turmas	10
5.4	Gerenciar Aulas	11
5.5	Solicitar Mudança	12
5.6	Visualizar Aulas	12
6	CONCLUSÕES E TRABALHOS FUTUROS	14
	REFERÊNCIAS	15

1 AGRADECIMENTOS

Agradeço ao professor orientador Marcelo Yukio Yamamoto, pelo acompanhamento, conselhos e apoio constante durante o desenvolvimento do projeto.

Ao Núcleo de Prática de Informática (NPI) do Centro Universitário Filadélfia (UniFil), pela oportunidade de participar de um ambiente de aprendizado prático e contribuir com uma solução voltada às necessidades reais da instituição.

Aos professores envolvidos nas orientações do estágio, pelo suporte técnico e pelas contribuições valiosas que auxiliaram na evolução do projeto.

2 INTRODUÇÃO

O presente relato descreve as atividades realizadas durante o Estágio Supervisionado do curso de Bacharelado em Engenharia de Software, tendo como foco o desenvolvimento do Planner UniFil (Sistema de Horários), um projeto destinado à gestão e organização dos cronogramas acadêmicos dos cursos de Computação do Centro Universitário Filadélfia (UniFil).

O estágio foi conduzido no âmbito do Núcleo de Prática de Informática (NPI), com o objetivo de solucionar um problema recorrente na instituição: a dificuldade em gerenciar os horários de aulas utilizando planilhas e documentos manuais. Esse processo mostrava-se trabalhoso, sujeito a erros e pouco eficiente, especialmente diante da necessidade de constantes alterações de turmas e professores.

Diante desse contexto, foi proposta a criação de um sistema web que centralizasse a elaboração, manutenção e consulta dos horários acadêmicos. O sistema foi desenvolvido utilizando o framework *Laravel* no *backend* e o *Vue.js* no *frontend*, integrando-se a um banco de dados *PostgreSQL*. Essa combinação de tecnologias possibilitou uma aplicação moderna, escalável e de fácil manutenção.

Ao longo do estágio, foi possível aplicar os conhecimentos adquiridos durante o curso, envolvendo modelagem de sistemas, engenharia de software, versionamento de código e boas práticas de desenvolvimento. O projeto proporcionou uma experiência prática significativa, ao mesmo tempo em que ofereceu uma solução eficaz para uma necessidade real da universidade.

3 DESCRIÇÃO DO SISTEMA

O *Planner UniFil* é uma aplicação web voltada à gestão dos cronogramas acadêmicos, permitindo que professores, coordenadores e alunos acessem e manipulem informações de forma centralizada e segura. O principal objetivo é substituir o uso de planilhas, oferecendo uma plataforma automatizada, intuitiva e responsiva.

O sistema foi desenvolvido em arquitetura MVC (*Model-View-Controller*), garantindo uma separação clara das responsabilidades e maior escalabilidade. Em relação às linguagens e tecnologias, foram utilizadas as seguintes, no *backend*, o *PHP 8.4.5* (PHP, 2025) com o framework *Laravel 12* (LARAVEL, 2025), para o *frontend*, o *Vue.js 3.5* (VUEJS, 2025) na construção de interfaces dinâmicas e reativas; para o gerenciamento dos dados, o banco de dados *PostgreSQL 17.4* (POSTGRESQL, 2025), administrado por meio do *DBeaver*; além disso, utilizou-se *Git* e *GitHub* para controle de versão e o *Visual Studio Code* como ambiente de desenvolvimento integrado (IDE).

Entre as funcionalidades principais implementadas, destacam-se:

- Cadastro e gerenciamento de usuários, matérias, turmas e aulas;
- Associação de professores a disciplinas e horários específicos;
- Solicitação e registro de alterações de aula;
- Visualização de horários personalizados para cada turma ou professor;
- Controle de acesso com autenticação segura;
- Interface administrativa para edição de cronogramas.

A modelagem do sistema foi realizada por meio de diagramas de caso de uso, classes, entidade-relacionamento (MER/DER), sequência e implantação. Esses diagramas definiram a estrutura lógica e funcional do sistema e orientaram todo o processo de desenvolvimento.

As telas foram projetadas no Figma, com layout responsivo e navegação simples, priorizando a usabilidade. O workflow foi documentado em notação BPMN, comparando o processo anterior (“As Is”) e o novo processo automatizado (“To Be”).

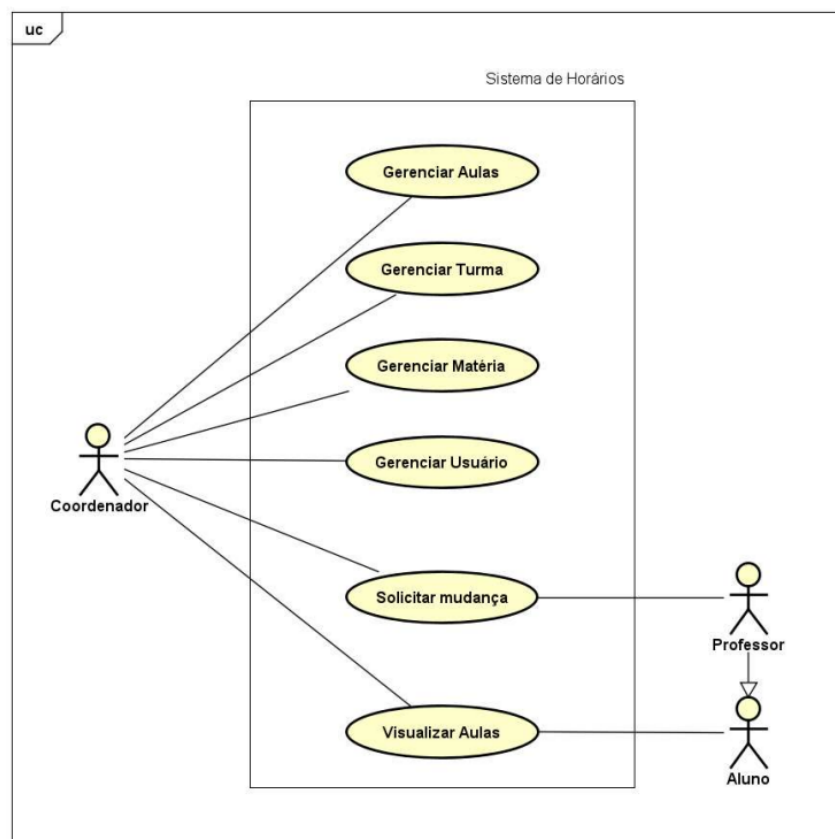
Durante o desenvolvimento, foram enfrentados desafios relacionados à integração entre frontend e backend e à estruturação do banco de dados. Esses obstáculos foram superados com estudo, testes e a aplicação de boas práticas de engenharia de software.

4 MODELAGEM DO SISTEMA

4.1 DIAGRAMA DE CASO DE USO

O diagrama de caso de uso é uma representação gráfica das interações entre atores externos e um sistema de software. Ele descreve as relações entre os diferentes casos e atores. Os elementos principais incluem atores, casos de uso e relacionamentos entre eles. É usado para capturar e comunicar os requisitos funcionais de um sistema de forma clara e compreensível, como pode ser verificado na Figura 1.

Figura 1 – Diagrama de Caso de Uso (Sistema de Horários)

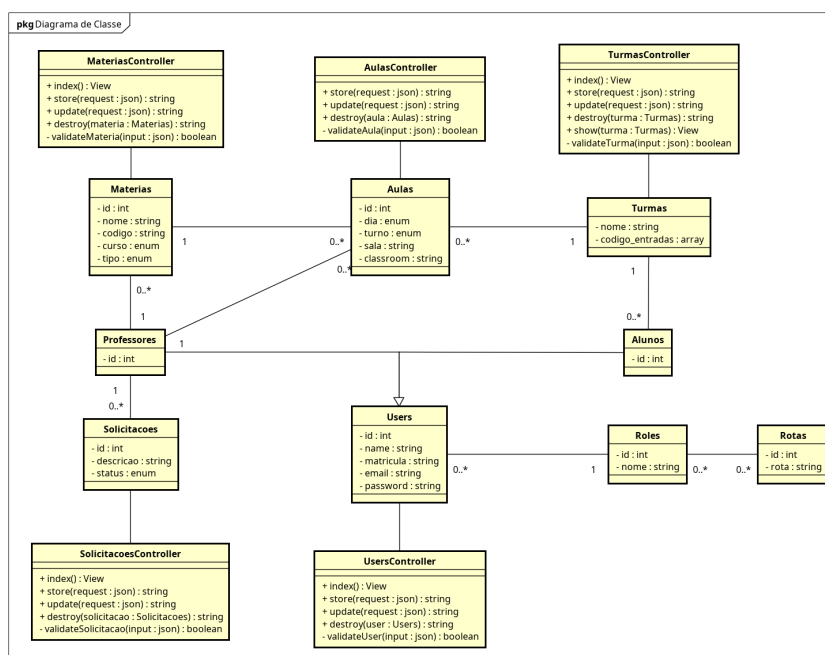


Fonte: Vitor Hugo Oliveira Paloco (2025)

4.2 DIAGRAMA DE CLASSE

O diagrama de classes descreve a estrutura estática do sistema, mostrando as classes que o compõem, seus atributos, métodos e os relacionamentos entre elas. Ele facilita a visualização da organização do código e o entendimento de como as entidades do sistema interagem. A Figura 2 apresenta o diagrama de classes do Sistema de Horários.

Figura 2 – Diagrama de Classe (Sistema de Horários)

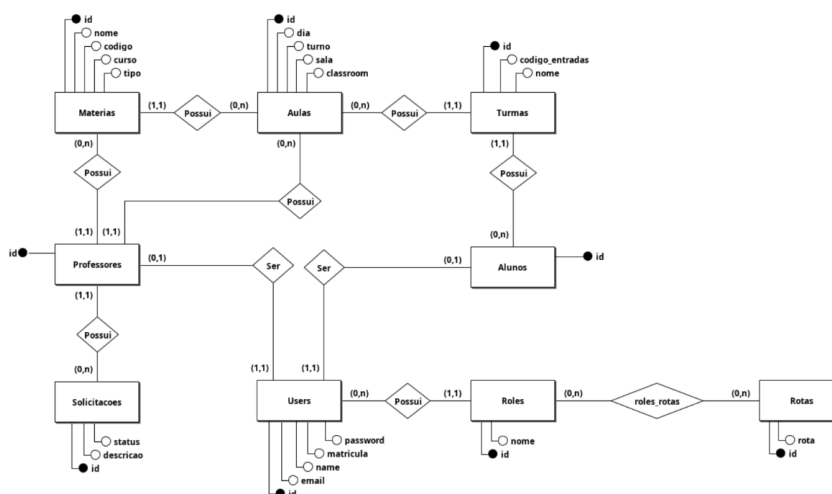


Fonte: Vitor Hugo Oliveira Paloco (2025)

4.3 MODELO ENTIDADE-RELACIONAMENTO (MER)

O Modelo Entidade-Relacionamento (MER) modela os dados do sistema, identificando as entidades principais, seus atributos e os relacionamentos entre elas. Ele serve como base para a criação do banco de dados relacional. A Figura 3 exibe o MER do Sistema de Horários.

Figura 3 – Modelo Entidade Relacionamento (Sistema de Horários)

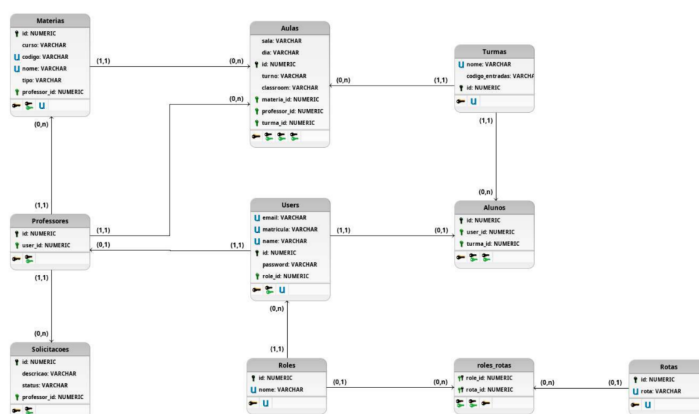


Fonte: Vitor Hugo Oliveira Paloco (2025)

4.4 DIAGRAMA ENTIDADE-RELACIONAMENTO (DER)

O Diagrama Entidade-Relacionamento (DER) representa a estrutura lógica do banco de dados, mostrando as tabelas, seus atributos e os relacionamentos entre elas. Ele define as chaves primárias e estrangeiras que garantem a integridade dos dados, além das cardinalidades que indicam como as entidades se conectam, servindo como base para a implementação do sistema. A Figura 4 exibe o DER do Sistema de Horários.

Figura 4 – Diagrama Entidade Relacionamento (Sistema de Horários)

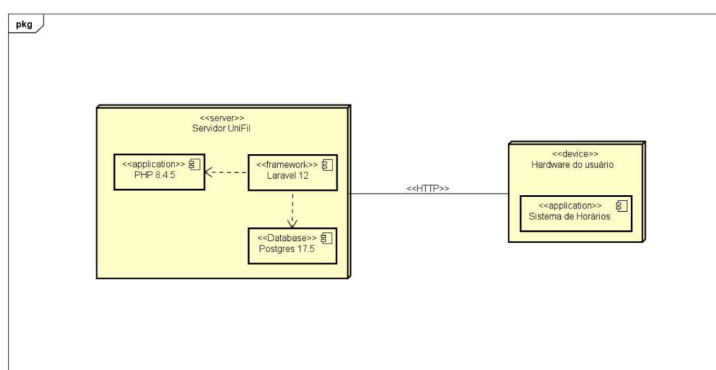


Fonte: Vitor Hugo Oliveira Paloco (2025)

4.5 DIAGRAMA DE IMPLANTAÇÃO

O diagrama de implantação mostra a distribuição física dos componentes de software em diferentes nós de hardware. Ele auxilia no planejamento da infraestrutura necessária para executar o sistema. A Figura 5 apresenta o diagrama de implantação do Sistema de Horários.

Figura 5 – Diagrama Implantação (Sistema de Horários)



Fonte: Vitor Hugo Oliveira Paloco (2025)

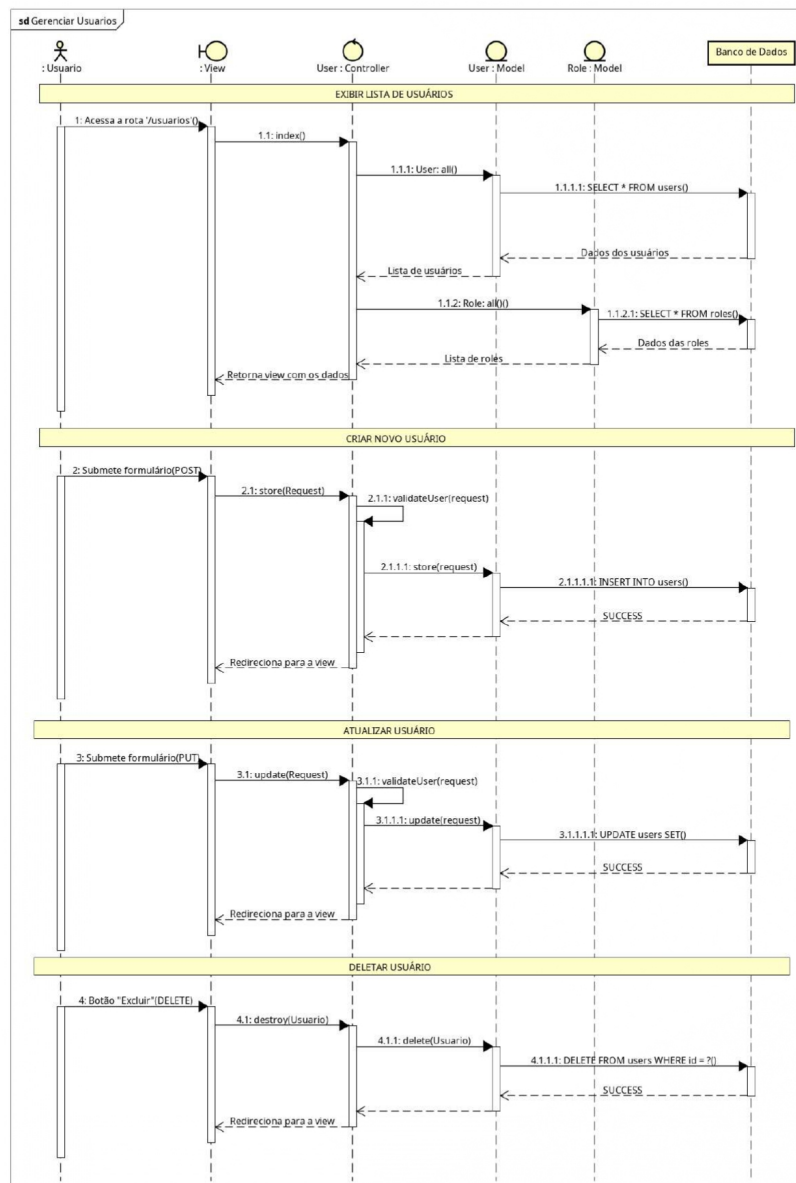
5 DIAGRAMAS DE SEQUÊNCIA

Os diagramas de sequência ilustram a interação temporal entre os objetos envolvidos em um determinado caso de uso. Ele mostra a ordem das mensagens trocadas entre os componentes do sistema.

5.1 GERENCIAR USUÁRIOS

A Figura 6 mostra o diagrama de sequência para o caso de uso "Gerenciar Usuários".

Figura 6 – Diagrama de Sequência - Gerenciar Usuários (Sistema de Horários)

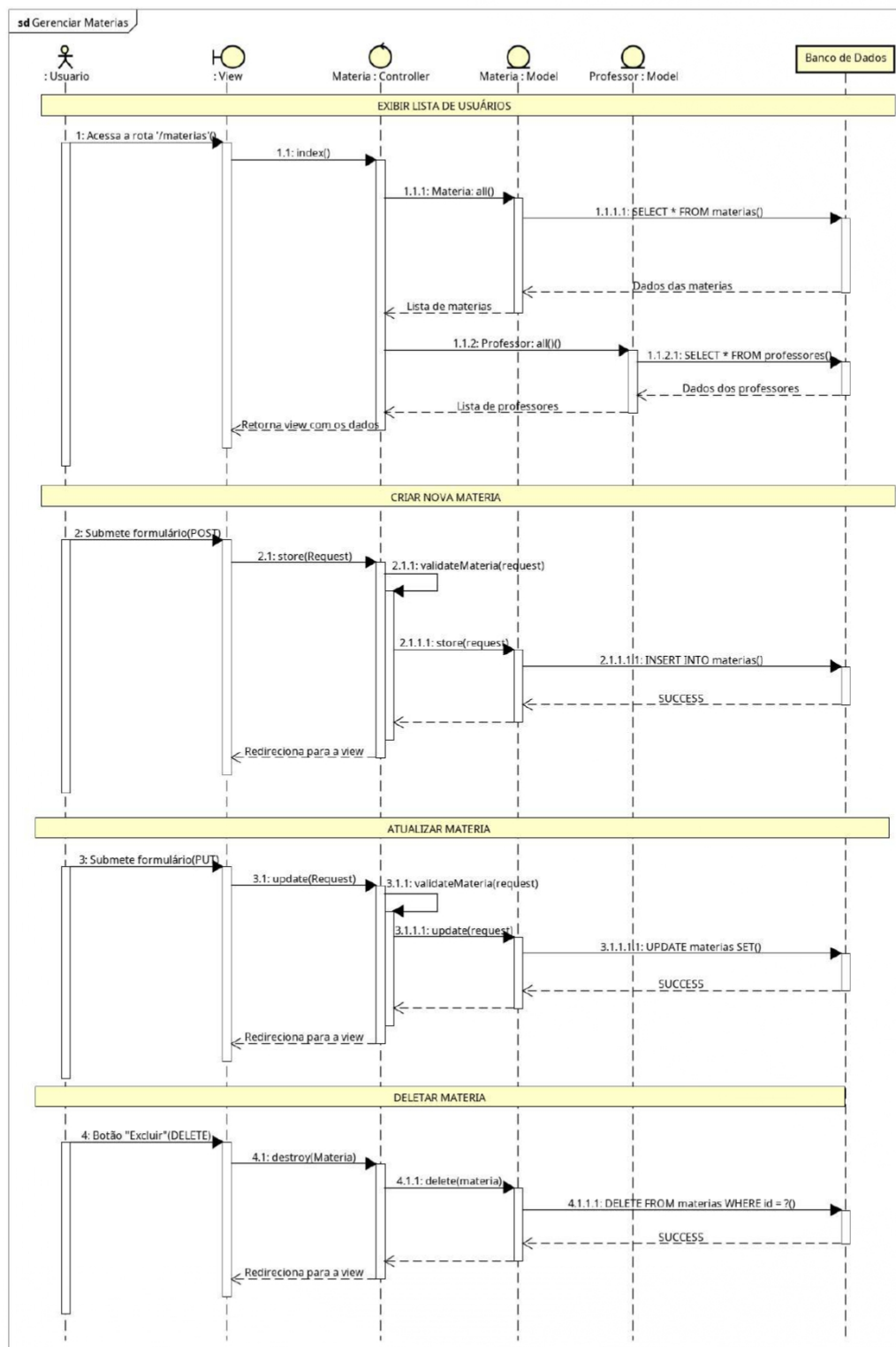


Fonte: Vitor Hugo Oliveira Paloco (2025)

5.2 GERENCIAR MATÉRIAS

A Figura 7 mostra o diagrama de sequência para o caso de uso "Gerenciar Matérias".

Figura 7 – Diagrama de Sequência - Gerenciar Matérias (Sistema de Horários)

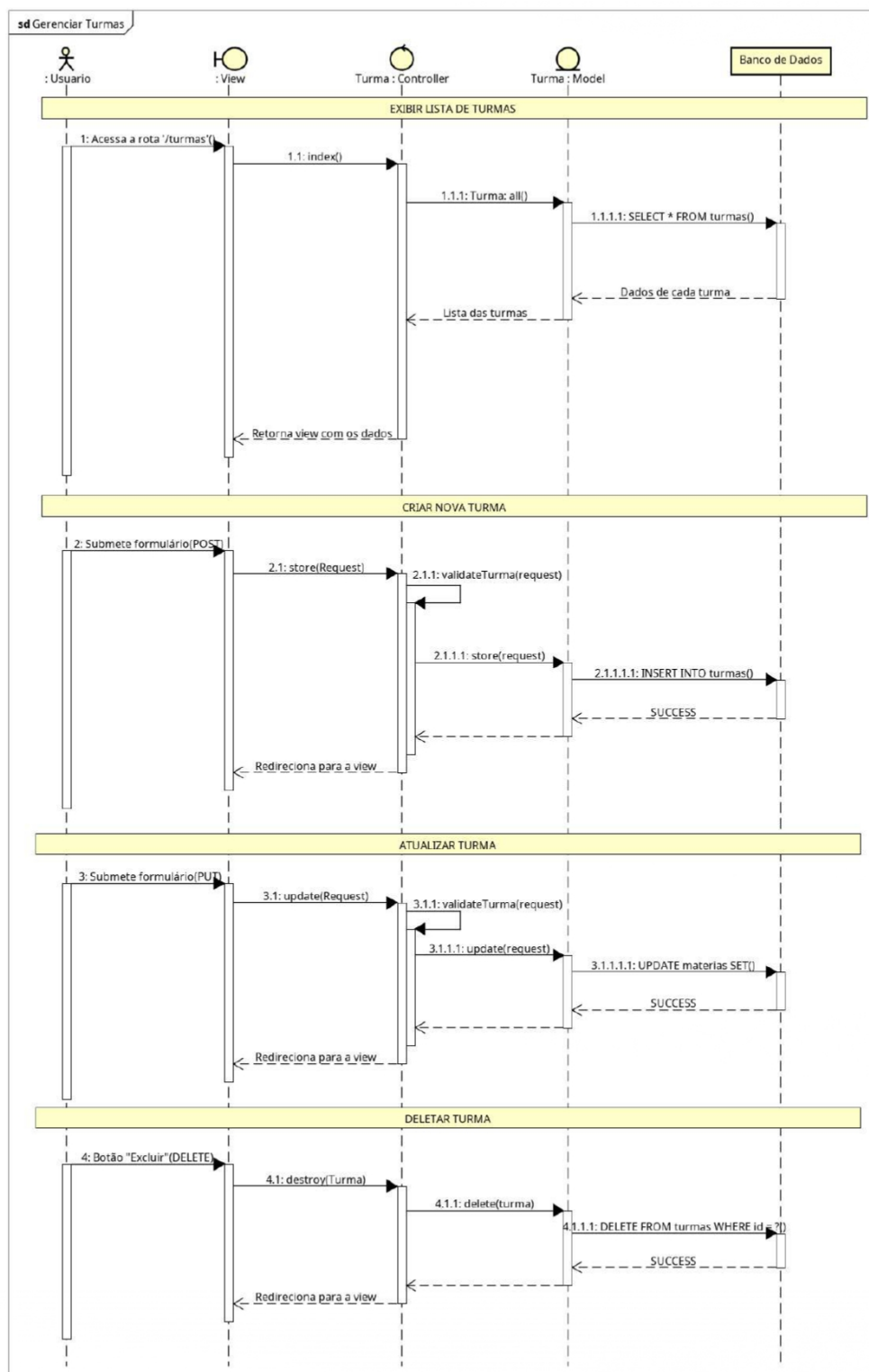


Fonte: Vitor Hugo Oliveira Paloco (2025)

5.3 GERENCIAR TURMAS

A Figura 8 mostra o diagrama de sequência para o caso de uso "Gerenciar Turmas".

Figura 8 – Diagrama de Sequência - Gerenciar Turmas (Sistema de Horários)

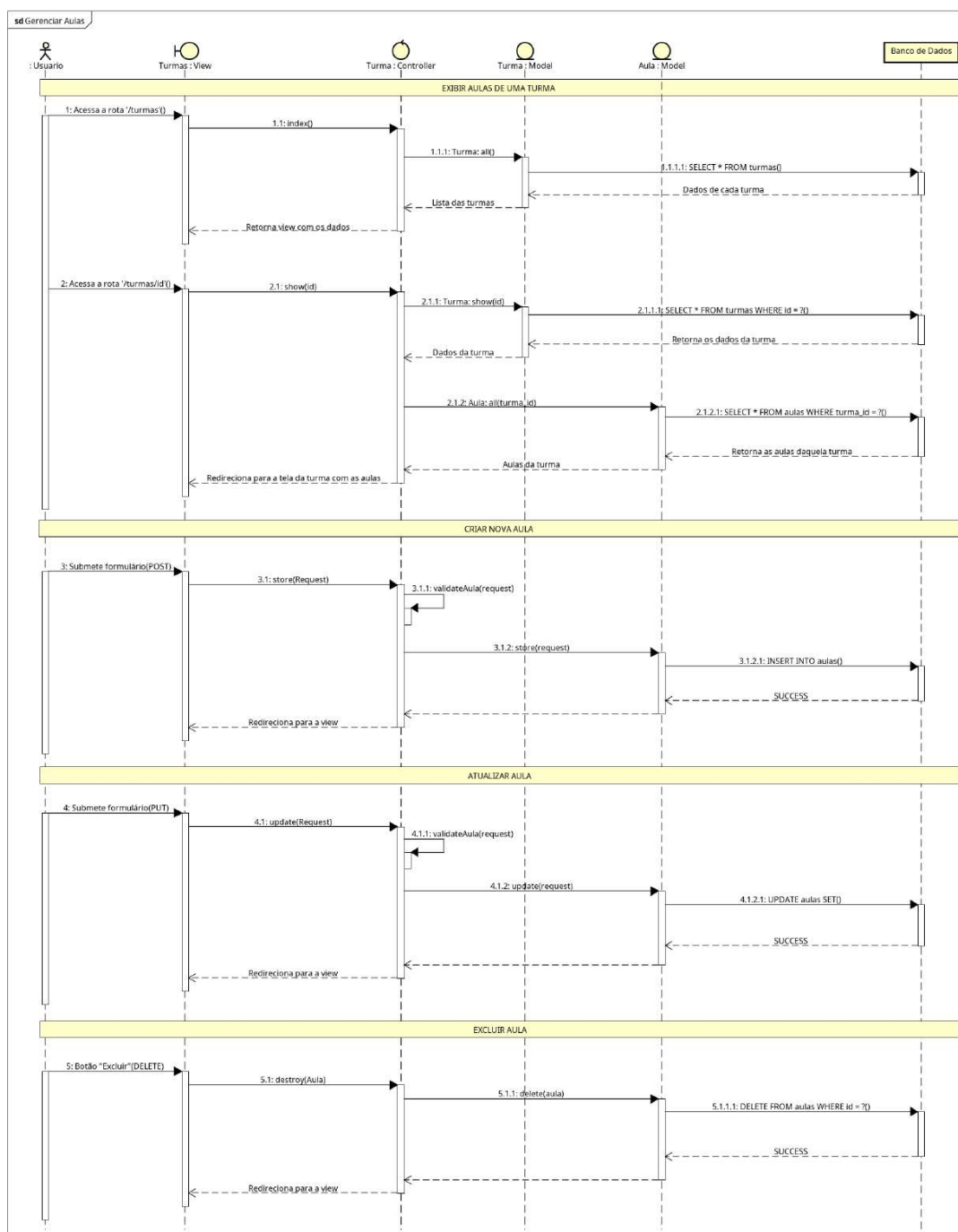


Fonte: Vitor Hugo Oliveira Paloco (2025)

5.4 GERENCIAR AULAS

A Figura 9 mostra o diagrama de sequência para o caso de uso "Gerenciar Aulas".

Figura 9 – Diagrama de Sequência - Gerenciar Aulas (Sistema de Horários)

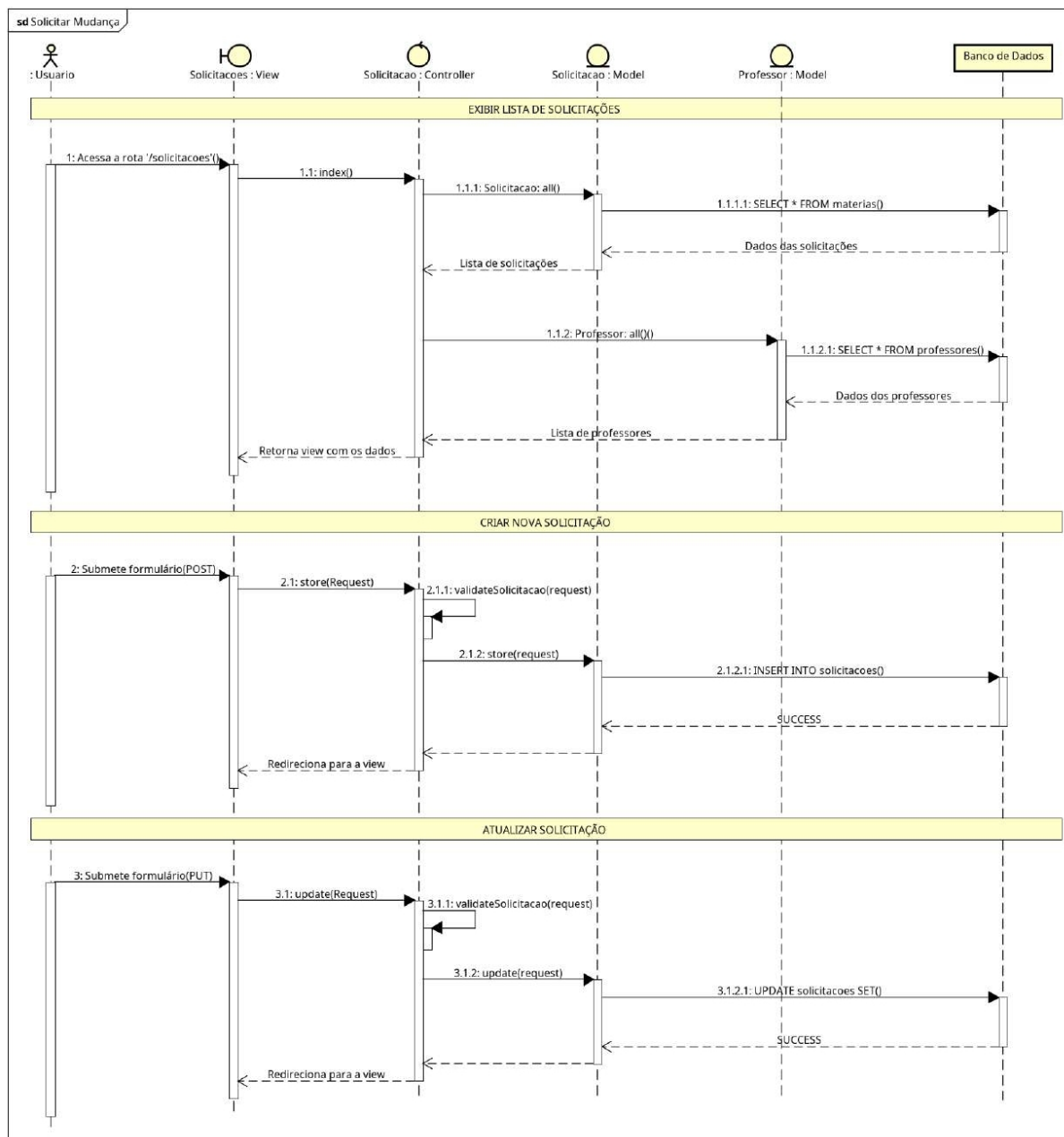


Fonte: Vitor Hugo Oliveira Paloco (2025)

5.5 SOLICITAR MUDANÇA

A Figura 10 mostra o diagrama de sequência para o caso de uso "Solicitar Mudança".

Figura 10 – Diagrama de Sequência - Solicitar Mudança (Sistema de Horários)

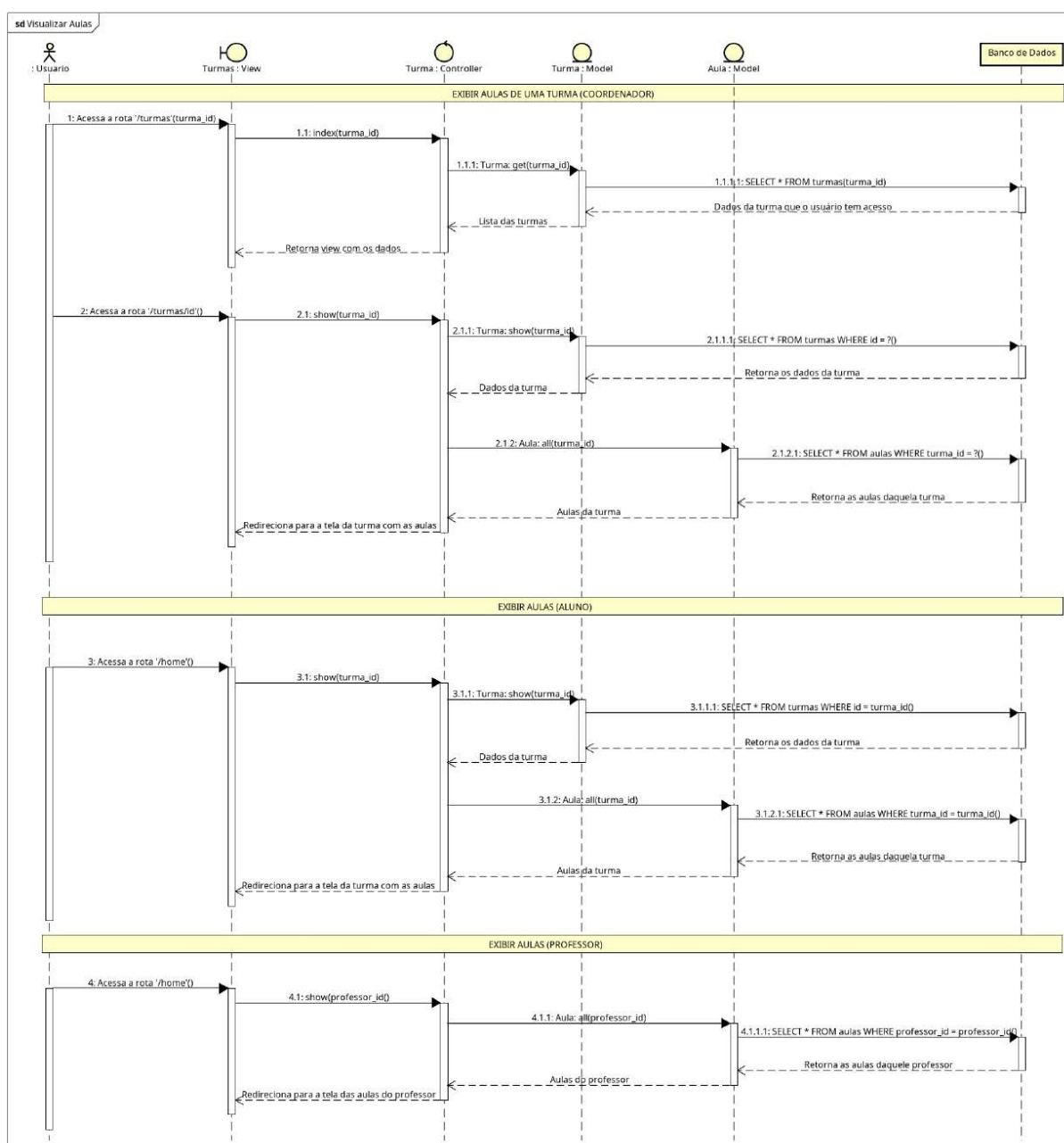


Fonte: Vitor Hugo Oliveira Paloco (2025)

5.6 VISUALIZAR AULAS

A Figura 11 mostra o diagrama de sequência para o caso de uso "Visualizar Aulas".

Figura 11 – Diagrama de Sequência - Visualizar Aulas (Sistema de Horários)



Fonte: Vitor Hugo Oliveira Paloco (2025)

6 CONCLUSÕES E TRABALHOS FUTUROS

O estágio supervisionado resultou na construção de um sistema funcional e eficiente, que cumpre os objetivos propostos. O Planner UniFil apresentou-se como uma solução prática para centralizar o gerenciamento acadêmico, reduzindo falhas, otimizando o tempo de atualização dos horários e melhorando a comunicação entre professores e alunos.

A experiência proporcionou grande aprendizado técnico e profissional, consolidando conhecimentos em programação web, modelagem de dados, controle de versões e integração de sistemas. O projeto reforçou a importância da engenharia de software na resolução de problemas reais e no desenvolvimento de soluções escaláveis.

Entretanto, devido ao tempo escasso e ao uso de tecnologias e linguagens que ainda não eram totalmente dominadas, o sistema permaneceu aberto a diversas melhorias. A principal delas refere-se ao polimento das regras de negócio, que em alguns casos específicos ainda podem apresentar lacunas, como nas situações de troca de turma de alunos. Além disso, funcionalidades complementares, como a exportação de dados em PDF e outras ferramentas de automação e relatório, ainda podem ser aprimoradas em versões futuras.

Como trabalhos futuros, sugerem-se:

- A implementação de inteligência artificial para geração automática e otimização de horários;
- A ampliação do uso do sistema para outros cursos da instituição;
- A criação de módulos de relatórios e notificações em tempo real;
- O aprimoramento das regras de negócio e validações, visando maior robustez e confiabilidade.

Conclui-se que o estágio foi uma oportunidade enriquecedora de aprendizado prático, contribuindo tanto para o desenvolvimento profissional quanto para a melhoria dos processos institucionais da UniFil. O projeto permitiu vivenciar os desafios reais do ciclo de desenvolvimento de software e demonstrou o potencial de evolução contínua da aplicação proposta.

REFERÊNCIAS

LARAVEL. *Laravel Documentation*. [S.l.], 2025. Acesso dia 27 de março de 2025. Disponível em: <<https://laravel.com/docs/12.x>>. 4

PHP. *PHP Documentation*. [S.l.], 2025. Acesso dia 27 de março de 2025. Disponível em: <<https://www.php.net/docs.php>>. 4

POSTGRESQL. *Sobre PostgreSQL*. [S.l.], 2025. Acesso dia 27 de março de 2025. Disponível em: <<https://www.postgresql.org/about/>>. 4

VUEJS. *Vue.js Documentation*. [S.l.], 2025. Acesso dia 27 de março de 2025. Disponível em: <<https://vuejs.org/guide/introduction>>. 4