



CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

KALEBE FURLAN DE OLIVEIRA

MARCELO YUKIO YAMAMOTO

**SISTEMA DE PRESENÇA DO NUCLEO DE PRÁTICA DE
INFORMATICA(NPI)**

**Londrina
2025**

SUMÁRIO

1	INTRODUÇÃO	2
2	DESCRIÇÃO DO SISTEMA	3
3	MODELAGEM DO SISTEMA	5
3.1	Diagrama de caso de uso	5
3.1.0.1	Caso de uso	6
3.2	Diagrama de classe	6
3.3	Diagrama de caso de entidade e relacionamento	7
3.4	DIAGRAMA DE IMPLANTAÇÃO	8
3.5	DIAGRAMA DE SEQUENCIA	9
3.6	DIAGRAMA DE ESTADO	12
3.7	TELAS	13
3.8	Interfaces do Sistema	13
3.8.1	Tela de Login	13
3.8.2	Tela Home	14
3.8.3	Tela de Lista de Alunos (Supervisor)	14
3.8.4	Tela de Lista de Alunos (Visualização Restrita)	14
3.8.5	Tela de Presenças (Aluno/Estágio)	14
3.8.6	Tela de Presenças Pendentes (Supervisor)	15
3.8.7	Resumo	15
4	CONCLUSÕES E TRABALHOS FUTUROS	17
4.1	TRABALHOS FUTUROS	17
	REFERÊNCIAS	18

1 INTRODUÇÃO

O Sistema de Presença NPI foi desenvolvido para suprir uma necessidade comum na instituição, a maneira de controlar a presença dos alunos no NPI. No contexto do Núcleo de Práticas de Informática (NPI), o controle tem sido feito manualmente, utilizando papel ou registros informais que demandam muito tempo dos supervisores, dificultando a verificação da presença.

A proposta do sistema surge da necessidade de modernizar o processo de registro de presença, tornando o registro mais ágil e seguro. Dessa forma, diferente de como era feito antes, o aluno não precisa mais buscar o supervisor para confirmar a presença, podendo fazer a solicitação diretamente pela plataforma, e assim que tiver tempo o supervisor vai conferir e aprovar o registro da sua presença, podendo visualizar a lista de presenças pendentes e escolher aceitar ou não.

O que será feito no sistema é uma organização dos dados, e a partir dele a presença é registrada junto da data e da hora, junto do status de pendente, confirmada ou negada, permitindo rápidas consultas e contribuindo, assim, para a transparência dos registros de presença.

O sistema oferece um ambiente simplificado e acessível, pensando na utilização de pessoas com níveis de domínio diversos em tecnologia. A expectativa é que ele possa facilitar o trabalho dos supervisores que pode ser muito repetitivo, proporcionando um ambiente mais justo e transparente.

O Sistema de Presença NPI serve muito bem de apoio para a rotina do NPI, modernizando a gestão de presenças, facilitando a comunicação com controle de frequência e futuras melhorias. Como apontado por Sommerville, “um software de qualidade deve ser usável, confiável e eficiente, e deve estar de acordo com os padrões de desenvolvimento estabelecidos” (SOMMERVILLE, 2011). Assim, não é apenas uma solução digital, mas uma forma de valorizar o tempo e a organização dos envolvidos.

2 DESCRIÇÃO DO SISTEMA

Neste capítulo iremos apresentar o desenvolvimento do sistema, detalhando todos os recursos que foram utilizados, funcionalidades implementadas e o aspecto técnico do funcionamento do sistema. O projeto foi pensado para atender a demanda do NPI quanto a seu gerenciamento das presenças dos alunos.

O sistema foi desenvolvido para ser uma aplicação web, acessível de qualquer navegador atual, com uma interface simples e intuitiva. O seu objetivo principal seria o de agilizar e simplificar os registros de presença, permitindo a solicitação da presença por parte dos alunos, com o controle dos supervisores para aprovar ou rejeitar essas solicitações.

Para garantir uma estrutura robusta, organizada e de fácil manutenção, o sistema está sendo desenvolvido utilizando tecnologias amplamente consolidadas no mercado. A linguagem de programação adotada foi o PHP 8.x, associada ao *framework* Laravel, que segue o padrão arquitetural MVC (*Model-View-Controller*), proporcionando uma separação clara entre a lógica de negócio, a camada de dados e a interface do usuário (OTWELL, 2024).

O banco de dados utilizado foi o MySQL, executado localmente por meio do ambiente de desenvolvimento XAMPP, que também forneceu suporte ao servidor Apache. O desenvolvimento do frontend foi realizado com HTML5, CSS3 e Bootstrap 5.3, garantindo responsividade e aderência à identidade visual da UniFil. Para renderização dinâmica das páginas, foi empregada a engine de templates Blade, integrada ao Laravel.

O sistema está estruturado com base em três tipos principais de usuários:

- **Aluno comum:** possui acesso a uma interface destinada exclusivamente à solicitação de registros de presença.
- **Aluno estagiário:** compartilha a mesma interface e funcionalidades do aluno comum, também voltadas à solicitação de presenças.
- **Supervisor:** conta com uma interface diferenciada, com funções administrativas adicionais, como:
 - Analisar as solicitações de presença enviadas por alunos e estagiários, podendo aprovar ou rejeitar cada uma.
 - Realizar o cadastro de novos alunos no sistema.

Todas as ações são controladas por permissões específicas, atribuídas conforme o perfil de cada usuário, garantindo segurança e integridade nos processos realizados.

As principais funcionalidades do sistema incluem: login (autenticação de usuários), com distinção de perfis (aluno, estagiário e supervisor); cadastro de novos alunos (função exclusiva dos supervisores); solicitação de presença por parte dos alunos e estagiários; e a visualização e gerenciamento das presenças aceitas e pendentes, função destinada aos supervisores. Essas funcionalidades foram desenvolvidas com foco na organização e no controle eficiente da frequência, de acordo com as permissões atribuídas a cada tipo de usuário.

Durante o desenvolvimento do sistema, ele foi testado utilizando o banco de dados MySQL com o uso do XAMPP, simulando o ambiente com servidor Apache e o banco de dados MySQL. As rotas, controladores e models foram criados de forma reutilizável, respeitando os padrões do framework Laravel, que adota o padrão arquitetural MVC (Model-View-Controller) (HANSEN, 2019). Com isso, o sistema foi preparado para a implantação online.

Além disso, foram utilizados diagramas UML para representar a lógica e a estrutura do sistema antes de sua implementação. Esses diagramas contribuíram para o planejamento e a validação da estrutura e dos fluxos do projeto, conforme recomendado por Booch et al. (1999) na modelagem de sistemas orientados a objetos (BOOCH; RUMBAUGH; JACOBSON, 1999).

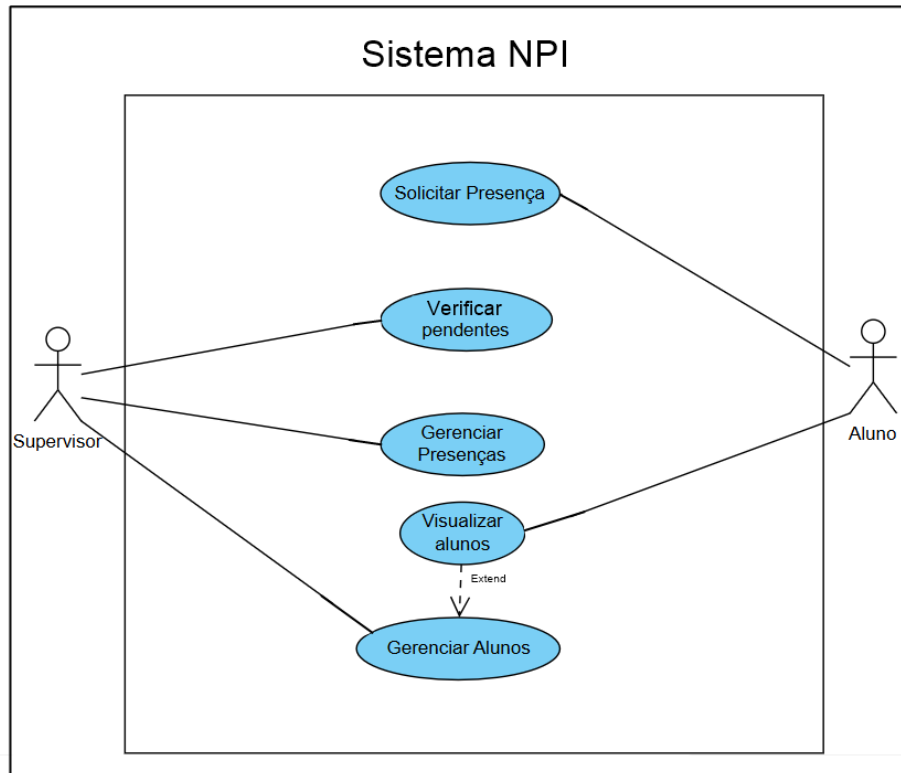
3 MODELAGEM DO SISTEMA

A modelagem do Sistema de Presença NPI apresenta uma visão completa da estrutura, funcionamento e fluxos do sistema, abrangendo diagramas de caso de uso, classe, entidade-relacionamento, implantação, sequência e estado. Essa modelagem descreve os tipos de usuários, as permissões e as interações realizadas, além de detalhar como os dados são organizados e processados. Também inclui a estrutura das interfaces, projetadas para garantir clareza, segurança e usabilidade no registro e controle de presenças.

3.1 DIAGRAMA DE CASO DE USO

O diagrama de caso de uso tem como objetivo auxiliar na compreensão do sistema e dos seus requisitos funcionais. Na Figura 1 é possível ter uma visão geral das interações entre os atores do tipo supervisor ou aluno com o sistema de presença NPI

Figura 1 – Diagrama de Caso de Uso no Sistema NPI



Fonte: O Autor (2025)

3.1.0.1 Caso de uso

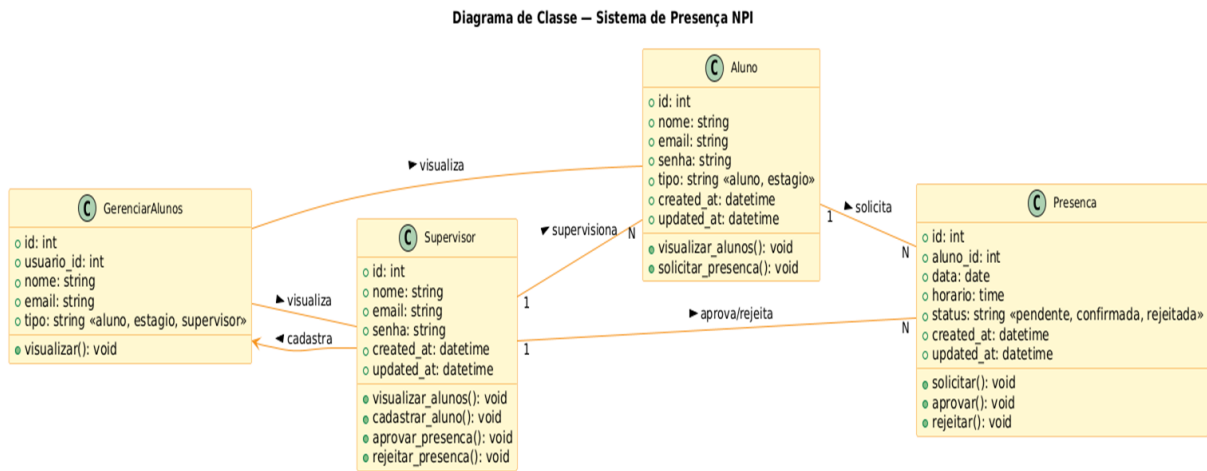
- **solicitar presenca:** Disponível para usuários do tipo aluno ou estagiário, permite enviar uma solicitação de presença que será avaliada por um supervisor.
- **aprovar solicitacao:** Permite ao supervisor confirmar uma presença previamente solicitada por um aluno ou estagiário.
- **rejeitar solicitacao:** Permite ao supervisor recusar uma solicitação de presença feita por um aluno ou estagiário.
- **visualizar alunos:** Permite que o supervisor visualize a lista completa de alunos cadastrados no sistema.
- **cadastrear aluno:** Permite que o supervisor cadastre um novo aluno no sistema, informando nome, email, tipo de usuário e outras informações básicas.

3.2 DIAGRAMA DE CLASSE

O Diagrama de Classe tem como objetivo representar a estrutura interna do sistema, evidenciando as classes principais, seus atributos, métodos e os relacionamentos entre elas. Na Figura 2, é possível visualizar a organização das entidades que compõem o Sistema de Presença NPI, como os usuários (alunos, estagiários e supervisores) e os registros de presença.

Esse diagrama permite compreender como os dados são estruturados, como as funcionalidades se distribuem entre as classes e quais são as ligações lógicas entre os elementos. Além disso, ele serve como base para a implementação do sistema, garantindo maior clareza na modelagem e facilitando futuras manutenções ou expansões da aplicação.

Figura 2 – Diagrama de Classe no Sistema NPI



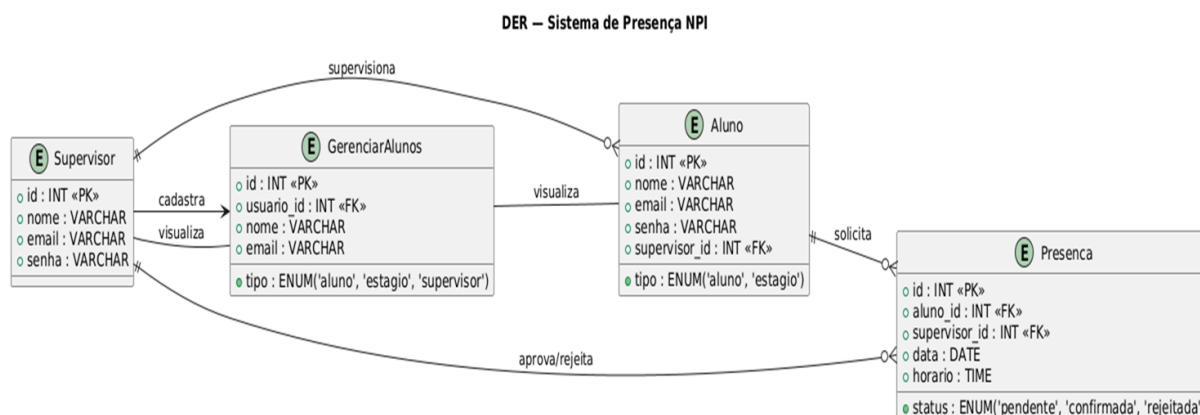
Fonte: O Autor (2025)

3.3 DIAGRAMA DE CASO DE ENTIDADE E RELACIONAMENTO

O Diagrama Entidade-Relacionamento (DER) tem como objetivo representar a estrutura lógica do banco de dados do sistema, evidenciando as entidades, seus atributos e os relacionamentos entre elas. Segundo Batini et al. (1992, p. 50), essa modelagem permite capturar os requisitos do domínio de forma estruturada e compreensível, facilitando tanto a comunicação entre analistas e desenvolvedores quanto a futura manutenção do sistema (BATINI; CERI; NAVATHE, 1992).

Na Figura 3, é possível observar a forma como os dados do Sistema de Presença NPI são organizados e conectados para garantir integridade, rastreabilidade e eficiência nas operações. A entidade Usuario centraliza as informações dos participantes do sistema, sejam eles alunos, estagiários ou supervisores, diferenciando-os por meio do campo tipo. A entidade Presenca está relacionada diretamente aos usuários e também aos projetos, registrando data, horário de entrada e saída, com a presença aparecendo nas solicitações pendentes para os supervisores. O sistema também contempla a entidade Projeto, permitindo o vínculo entre usuários e projetos específicos. Essa relação é intermediada pela tabela Associacao Projeto Aluno, que armazena a função do usuário dentro de determinado projeto. Para justificar ausências ou alterações no registro de presença, a entidade Justificativa foi criada. Ela está associada diretamente à presença e contém o motivo, a data de envio e o status da aprovação, proporcionando maior controle e transparência.

Figura 3 – Diagrama de caso de entidade e relacionamento no Sistema NPI

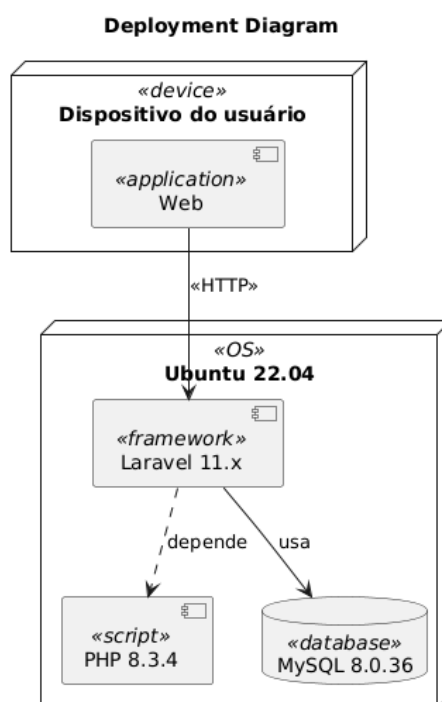


Fonte: O Autor (2025)

3.4 DIAGRAMA DE IMPLANTAÇÃO

A Figura 4 apresenta o diagrama de implantação que ilustra a estrutura do sistema sendo essencial para entender como será implantado no ambiente

Figura 4 – Diagrama de Implantação



Fonte: O Autor (2025)

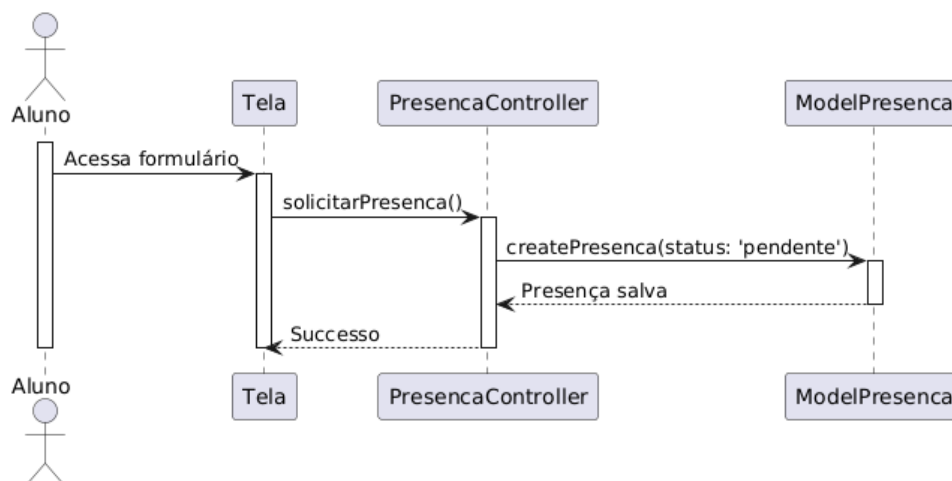
3.5 DIAGRAMA DE SEQUENCIA

Através desses diagramas de sequência, é possível compreender com clareza a forma como o Sistema de Presença NPI lida com os principais fluxos relacionados à autenticação, solicitação, aprovação, cadastro e exclusão de dados. As interações entre usuário → interface → controlador → modelo → banco de dados seguem o padrão arquitetural MVC (Model-View-Controller), oferecendo um sistema organizado, escalável e de fácil manutenção.

- **Solicitação de Presença (Aluno):** O primeiro diagrama de sequência representa o processo de solicitação de presença por um aluno ou estagiário. O fluxo inicia com o usuário acessando a tela onde é possível realizar a solicitação. Ao submeter o formulário, uma requisição POST /presencas/solicitar é enviada para o controlador PresencaController.

O controlador, por sua vez, chama o método responsável por criar uma nova instância do modelo Presenca, atribuindo automaticamente o status inicial como pendente. A presença é então salva no banco de dados por meio de um INSERT, e o sistema responde com um redirecionamento e uma mensagem de sucesso, informando que a solicitação foi registrada.

Figura 5 – Diagrama de Sequencia Solicitar presença

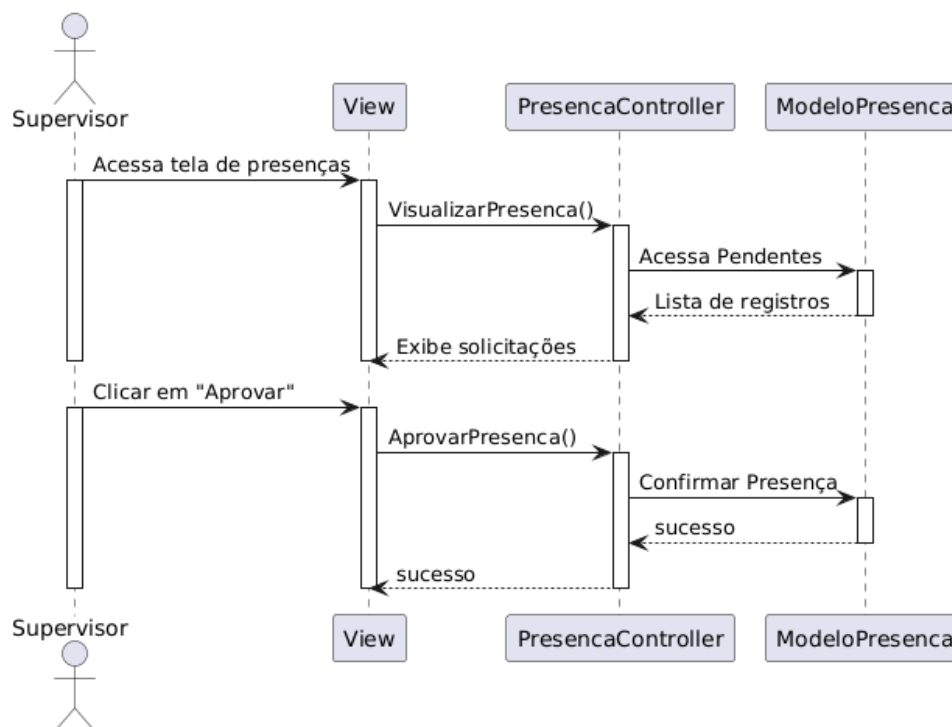


Fonte: O Autor (2025)

- **Aprovação de Presença (Supervisor):** O segundo diagrama ilustra a funcionalidade onde um supervisor acessa a lista de solicitações pendentes e aprova uma delas. Inicialmente, o supervisor acessa a rota /presencas, que realiza um GET e busca os registros pendentes no banco de dados, utilizando o relacionamento com o aluno para apresentar os dados de forma organizada.

Quando o supervisor clica em “Aprovar”, uma requisição POST /presencas/id/aprovar é enviada ao controlador. O PresencaController localiza o registro correspondente usando find(id), atualiza o campo status para confirmada, e salva a alteração no banco por meio de um UPDATE.

Figura 6 – Diagrama de Sequencia Aceitar presença

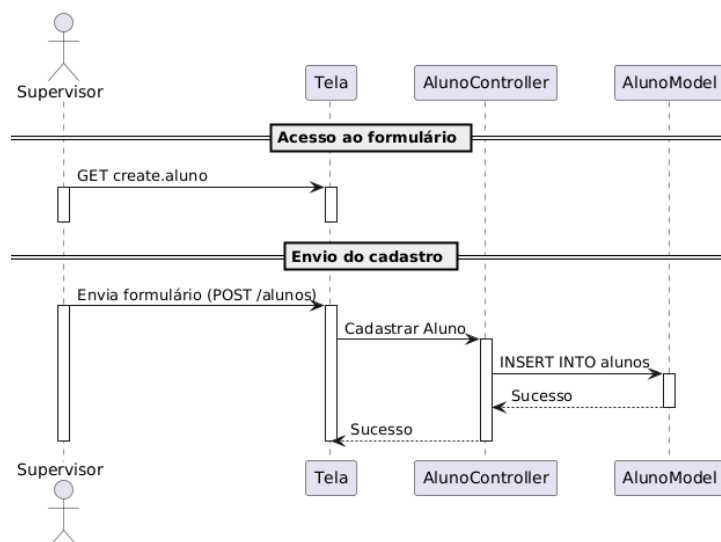


Fonte: O Autor (2025)

- **Cadastro de Novo Aluno (Supervisor):** O terceiro diagrama representa o fluxo de cadastro de um novo aluno no sistema, ação que é exclusiva dos usuários do tipo supervisor. Inicialmente, o supervisor acessa o formulário por meio da rota GET /alunos/create, o qual retorna uma interface de preenchimento HTML.

Após inserir os dados e submeter o formulário via POST /alunos, o controlador AlunoController recebe os dados e realiza a validação. Em seguida, é criado um novo objeto do modelo Aluno, que é salvo no banco de dados através de um INSERT. Uma vez concluída a operação, o sistema redireciona o usuário de volta à tela de listagem com uma mensagem indicando o sucesso do cadastro.

Figura 7 – Diagrama de Sequencia Cadastro Aluno

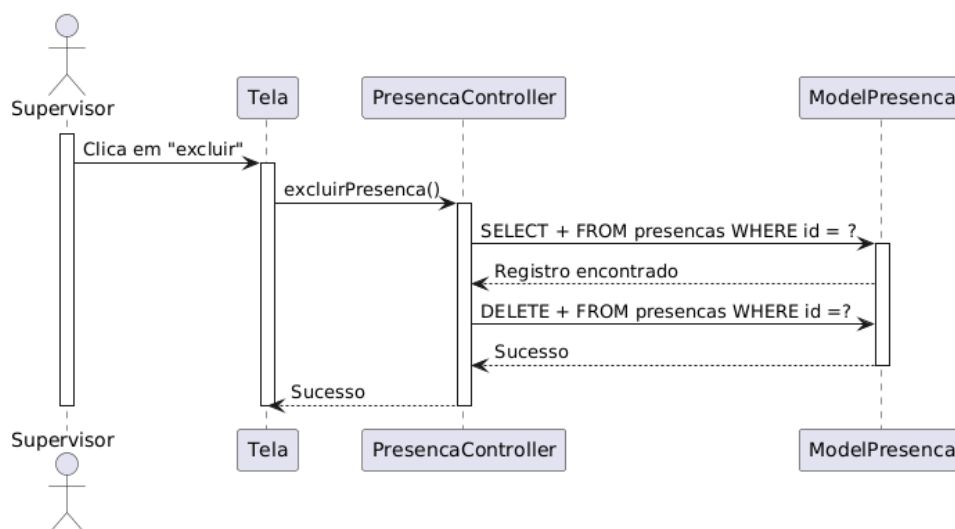


Fonte: O Autor (2025)

- **Exclusão de Presença:** Este diagrama de sequência descreve a funcionalidade de exclusão de um registro de presença, que pode ser acessada por usuários autenticados. Ao clicar no botão "Excluir", é enviada uma requisição DELETE /presencas/id ao controlador PresencaController.

O controlador localiza o registro correspondente utilizando `findById(id)`, confirmando a existência da presença. Em seguida, realiza a exclusão definitiva do registro com um comando DELETE no banco de dados.

Figura 8 – Diagrama de Sequencia Excluir Presença

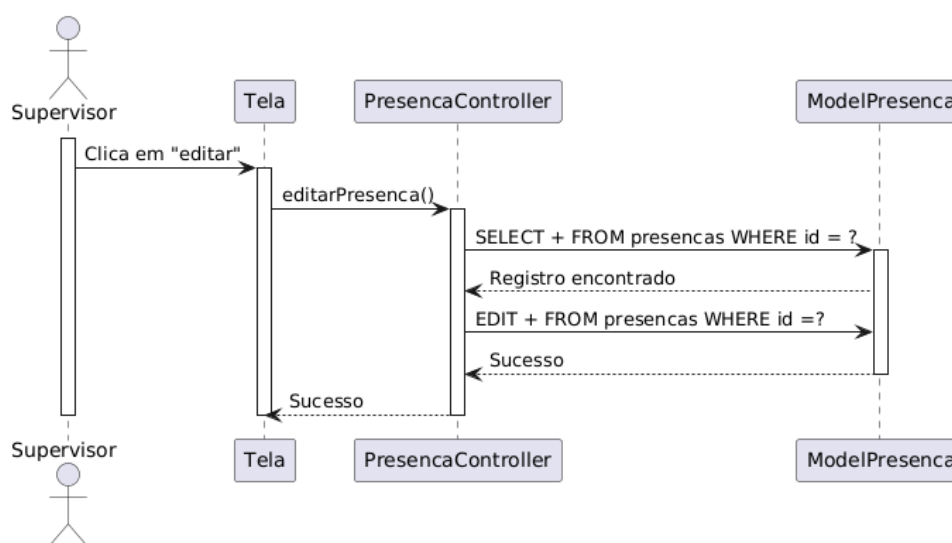


Fonte: O Autor (2025)

- **Edição de Presença:** O último diagrama de sequência descreve a funcionalidade de edição de um registro de presença, que pode ser acessada por usuários autenticados. Ao clicar no botão "Editar", é enviada uma requisição `EDIT /presencas/id` ao controlador `PresencaController`.

O controlador localiza o registro correspondente utilizando `findById(id)`, confirmando a existência da presença. Em seguida, realiza a edição definitiva do registro de hora e data com um comando `UPDATE` no banco de dados, ajustando os valores informados.

Figura 9 – Diagrama de Sequencia Editar Presença



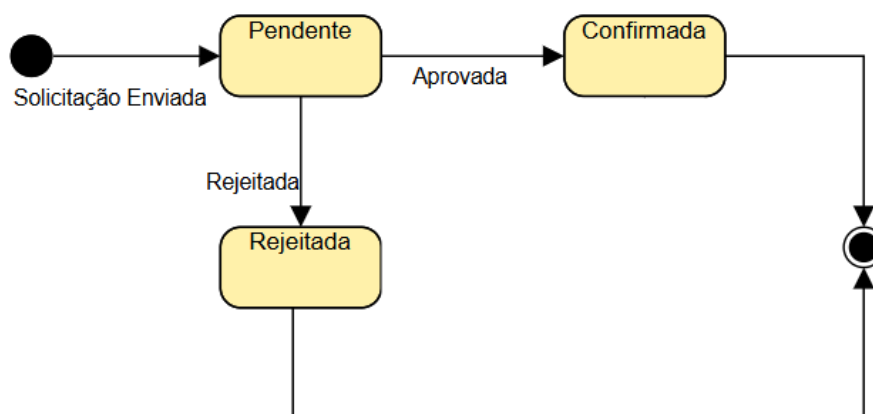
Fonte: O Autor (2025)

3.6 DIAGRAMA DE ESTADO

O Diagrama de Estado da entidade Presença representa o ciclo de vida de um registro de presença dentro do sistema. Ele descreve os possíveis estados pelos quais uma presença pode passar, desde o momento em que é solicitada pelo aluno ou estagiário até sua conclusão, seja por aprovação ou rejeição com ou sem justificativa, conforme recomendado ao descrever ciclos de vida de objetos em sistemas orientados a objetos (BOOCH; RUMBAUGH; JACOBSON, 1999).

O fluxo inicia no estado **Pendente**, onde a solicitação aguarda análise. A partir daí, a presença pode ser **Confirmada** ou **Rejeitada** por um supervisor. No caso de rejeição, o aluno tem a opção de enviar uma **Justificativa**, que será reavaliada. Se aceita, a presença retorna ao estado de **Confirmada**; caso contrário, permanece como **Rejeitada**. Esse diagrama reforça a lógica de validação e controle aplicada pelo

Figura 10 – Diagrama de Estado



Fonte: O Autor (2025)

sistema, garantindo clareza e rastreabilidade nas decisões tomadas sobre cada presença registrada.

3.7 TELAS

A seguir são apresentadas as telas do sistema, abaixo temos a figura 11 sendo a home junto com o link para o drive com as outras telas e abaixo as informações referentes as tela, para visualização abaixo está a homem do sistema, com os dois caminhos, o gerenciar alunos e gerenciar presenças.

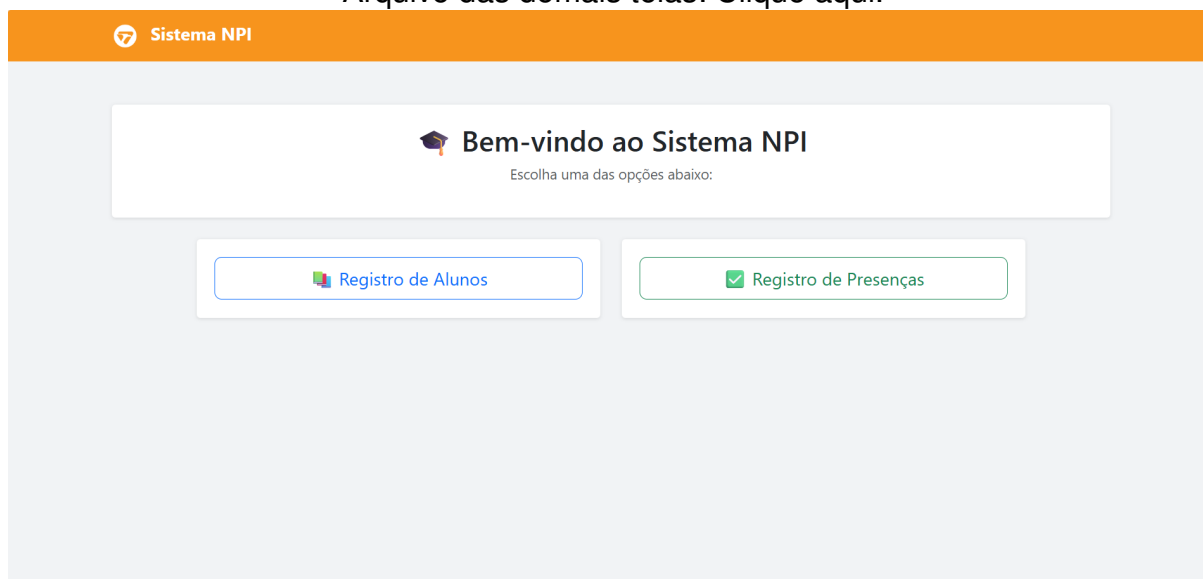
3.8 INTERFACES DO SISTEMA

As interfaces do **Sistema de Presença NPI** foram projetadas com base em uma arquitetura limpa e responsiva, utilizando o framework Bootstrap com adaptações inspiradas na identidade visual da UniFil. A navegação é adaptada conforme o tipo de usuário autenticado, promovendo acessibilidade e controle sobre as permissões de cada perfil.

3.8.1 Tela de Login

A tela inicial do sistema é a interface de **Login**, onde todos os usuários devem se autenticar com e-mail e senha. Conforme o tipo de usuário (aluno, estágio ou supervisor), o sistema redireciona para áreas distintas com permissões apropriadas.

Figura 11 – Home do Sistema
Arquivo das demais telas: Clique aqui.



Fonte: O Autor (2025)

3.8.2 Tela Home

Após o login, o usuário visualiza a **Home**, que apresenta atalhos para as seções principais como: cadastro de alunos e registros de presença. Essa tela é ajustada conforme o tipo de usuário, exibindo opções como “Cadastrar aluno novo” apenas para supervisores.

3.8.3 Tela de Lista de Alunos (Supervisor)

A Figura 12 apresenta a visualização dos alunos cadastrados, acessada apenas por supervisores. Essa interface permite ao supervisor cadastrar novos alunos, visualizar detalhes como tipo, e-mail, idade e vínculo com um supervisor responsável.

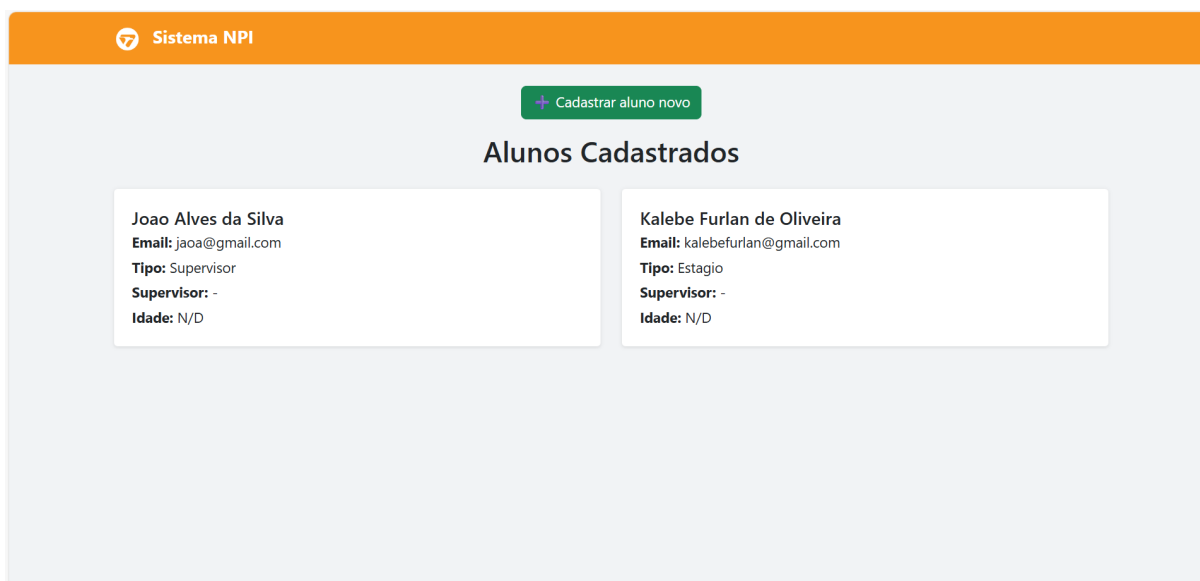
3.8.4 Tela de Lista de Alunos (Visualização Restrita)

A tela de Lista de Alunos representa a versão pública da listagem de alunos, visualizada por usuários com permissões reduzidas. Nela, o botão de cadastro é ocultado, garantindo que apenas supervisores possam gerenciar usuários.

3.8.5 Tela de Presenças (Aluno/Estágio)

A tela de presenças apresenta uma interface especial para alunos e estagiários. Nela, é possível realizar uma solicitação de presença para o dia atual, e visualizar

Figura 12 – Home do Sistema



Fonte: O Autor (2025)

a tabela com os registros confirmados.

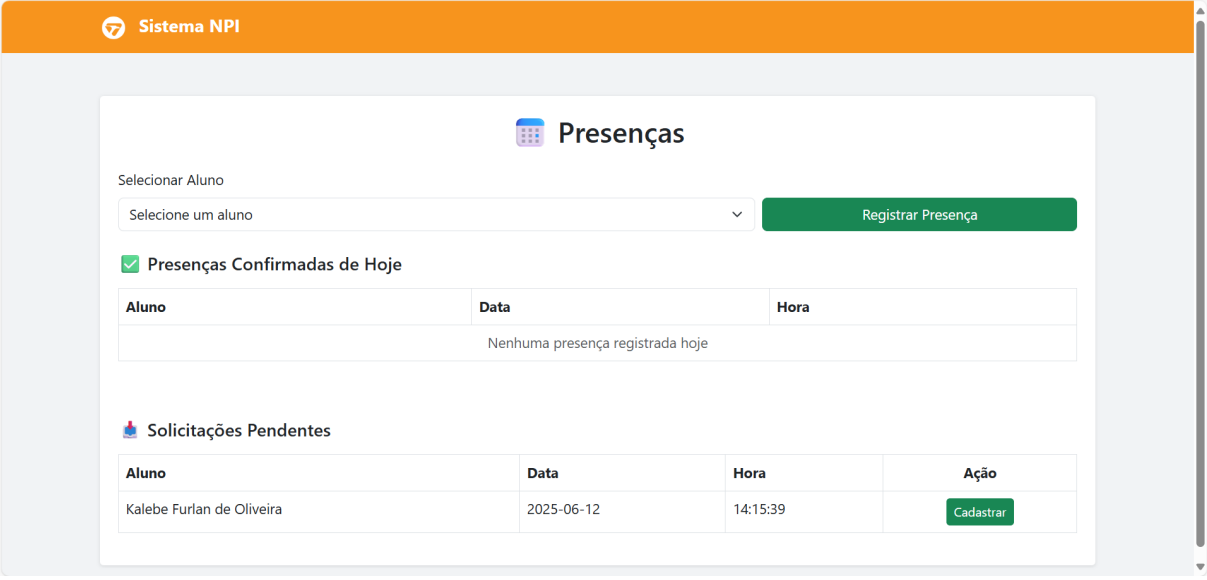
3.8.6 Tela de Presenças Pendentes (Supervisor)

Supervisores acessam uma versão estendida da tela de presença, onde além das presenças confirmadas, também visualizam todas as **solicitações pendentes**. Essa interface oferece botões de ação como *Aprovar* ou *Rejeitar*, refletindo diretamente no status da presença.

3.8.7 Resumo

As interfaces do sistema foram desenvolvidas com foco na usabilidade e clareza. A separação de permissões por tipo de usuário garante que cada funcionalidade esteja acessível somente a quem possui a devida autorização. Isso promove segurança e confiabilidade na gestão das presenças e no controle de acessos do sistema. Link para visualizar as demais telas: [Clique aqui](#).

Figura 13 – Home do Sistema



The screenshot displays the 'Sistema NPI' interface. At the top, an orange header bar contains the system name. The main content area is titled 'Presenças' and includes a 'Selecionar Aluno' dropdown menu with the placeholder 'Selecione um aluno' and a green 'Registrar Presença' button. Below this, a section titled 'Presenças Confirmadas de Hoje' with a green checkmark icon shows a table with columns 'Aluno', 'Data', and 'Hora'. The table is currently empty, displaying the message 'Nenhuma presença registrada hoje'. Further down, a section titled 'Solicitações Pendentes' with a calendar icon shows a table with columns 'Aluno', 'Data', 'Hora', and 'Ação'. This table contains one entry for 'Kalebe Furlan de Oliveira' on '2025-06-12' at '14:15:39', with a green 'Cadastrar' button in the 'Ação' column.

Sistema NPI

Presenças

Selecionar Aluno

Selecione um aluno

Registrar Presença

✓ Presenças Confirmadas de Hoje

Aluno	Data	Hora
Nenhuma presença registrada hoje		

📅 Solicitações Pendentes

Aluno	Data	Hora	Ação
Kalebe Furlan de Oliveira	2025-06-12	14:15:39	Cadastrar

Fonte: O Autor (2025)

4 CONCLUSÕES E TRABALHOS FUTUROS

O desenvolvimento do sistema, permitiu que alcançássemos os objetivos estabelecidos, projetar e implementar uma aplicação que facilitasse o controle de presença, promovendo autonomia, transparência e praticidade para alunos e supervisores.

Ao longo do desenvolvimento, foram aplicados diversos conhecimentos adquiridos ao longo do curso, de forma prática, nas áreas de programação, modelagem de dados, autenticação de usuários, manipulação de banco de dados e utilização do framework Laravel. Foram também empregadas ferramentas de modelagem, como os diagramas UML, que contribuíram significativamente para o planejamento e a organização do sistema, proporcionando uma visualização clara dos componentes e fluxos envolvidos.

Entre os principais aprendizados obtidos neste projeto, destaca-se a importância da validação de requisitos diretamente com os usuários finais, gestão adequada de permissões de acesso dependendo do perfil do usuário e integração do frontend e backend da aplicação. Proporcionou uma experiência prática no desenvolvimento de software, incluindo tomada de decisões, testes, correções e refinamento da interface.

O sistema oferece uma contribuição simples, porém eficaz, no registro das presenças. Os supervisores passam a ter maior controle e segurança sobre os registros, enquanto os alunos podem acompanhar suas presenças de forma transparente. Além disso, a segmentação das funcionalidades por tipo de usuário contribui significativamente para a segurança e integridade dos dados.

4.1 TRABALHOS FUTUROS

Como sugestão para trabalhos futuros, recomenda-se a ampliação do sistema com a implementação de novos módulos, que possibilitem a expansão do uso além do contexto acadêmico. Por exemplo, seria viável adaptar o sistema para empresas, de forma que pudesse ser utilizado no controle de ponto e presença de colaboradores. Essa abordagem está alinhada com os princípios de modularidade e escalabilidade em sistemas de software, conforme destacado por (SANTOS, 2022), que defende a construção de soluções modulares e adaptáveis a diferentes domínios, facilitando a evolução e reutilização do sistema em novos cenários.

O estágio supervisionado proporcionou não apenas a aplicação dos conhecimentos mas também a experiência de lidar com os desafios do desenvolvimento, com crescimento pessoal, profissional e acadêmico.

REFERÊNCIAS

BATINI, C.; CERI, S.; NAVATHE, S. B. *Conceptual Database Design: an Entity-Relationship Approach*. [S.l.]: Benjamin/Cummings, 1992. 7

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. *The Unified Modeling Language User Guide*. [S.l.]: Addison-Wesley, 1999. 4, 12

HANSEN, M. S. *Laravel: Up and Running*. [S.l.]: O'Reilly Media, 2019. 4

OTWELL, T. *Laravel Documentation*. [S.l.], 2024. <<https://laravel.com/docs>>. 3

SANTOS, G. Engenharia de software escalável através de domínios e modularidade em negócio. In: *Anais Estendidos do Congresso Brasileiro de Software: Teoria e Prática (CBSOFT)*. Uberlândia, MG: Sociedade Brasileira de Computação, 2022. p. 20–23. 17

SOMMERVILLE, I. *Software Engineering*. 9. ed. [S.l.]: Addison-Wesley, 2011. 2