**Desenvolvimento de um Sistema de Automatização de Chamada**

Especificação de Requisitos de Software

Versão <1.3>

João A. Cordeiro

André L. F. Pieper

Mentor Willian A. H. Nascimento

**Agradecimentos**

Gostaríamos de prestar nossos mais sinceros agradecimentos a Willian Hübner e Leticia Pieper por nos orientar no decorrer da realização deste projeto e fazendo com que a conclusão deste fosse possível. Agradecemos por confiar em nosso potencial, por sempre estarem nos incentivando a sermos uma versão melhor de nós mesmos e nos capacitando e preparando para a nossa jornada.

Também agradecemos ao Anderson Haupt por nos auxiliar no desenvolvimento do código deste projeto.

Um agradecimento especial ao professor e coordenador Rafael F. Vieira que mesmo que não tenha nos auxiliado diretamente com a realização do nosso projeto, a sua paixão pelo ensino e a sua energia e disposição de tempo para com os alunos se torna contagiante nos auxiliando e influenciando a darmos o nosso melhor no que precisamos realizar.

Ao nosso amigo Hans Donner, um grande abraço e nossa gratidão por tê-lo em nossa caminhada, sendo um grande referencial com sua enorme dedicação e disposição. Obrigado por nos auxiliar em nossa jornada e incentivar a sempre seguirmos em frente.

A dedicação e disposição de todos em compartilhar seu tempo e experiência foram indispensáveis para superar as barreiras que encontramos.

Histórico da Revisão Documentação

| **Data** | **Versão** | **Descrição** | **Autor** |
| --- | --- | --- | --- |
| 01/10/2024 | 1.0 | Iniciando documentação | João A. Cordeiro |
| 12/10/2024 | 1.1 | Sequência da documentação | João A. Cordeiro |
| 05/11/2024 | 1.2 | Sequência da documentação | João A. Cordeiro, André L. F. Pieper |
| 11/11/2024 | 1.3 | Finalização da documentação | João A. Cordeiro |

Histórico da Revisão e Desenvolvimento do Projeto

| **Data Início** | **Data Final** | **Descrição** | **Autor** |
| --- | --- | --- | --- |
| 07/08/2024 | 07/08/2024 | Definição de Escopo | André, João, Mentor |
| 08/08/2024 | 22/08/2024 | Versão 1.0 API (Fazer um CRUD conectado com postgres) | João A. Cordeiro |
| 08/08/2024 | 22/08/2024 | Criar gerador de QR Code | André L. F. Pieper |
| 23/08/2024 | 06/09/2024 | Publicar API na WEB para receber post | João A. Cordeiro |
| 23/08/2024 | 30/08/2024 | Criar gerador funcional para Mobile | André L. F. Pieper |
| 31/08/2024 | 13/09/2024 | Criar leitor de QR Code funcional | André L. F. Pieper |
| 07/09/2024 | 20/09/2024 | Versão 2.0(Implementar Prisma na API + validação de tempo) | João A. Cordeiro |
| 14/09/2024 | 26/10/2024 | Fazer tela de login, tela home, tela do gerador e capturar IMEI (Mobile) | André L. F. Pieper |
| 21/09/2024 | 26/10/2024 | Versão 3.0 (Implementar todas as tabelas e rotas, prisma e suas validações específicas) | João A. Cordeiro |
| 27/10/2024 | 17/11/2024 | Fazer tela de login, tela home, gerar chamada (WEB professor) | André L. F. Pieper |
| 27/10/2024 | 08/11/2024 | Versão 4.0 (Implementar autenticação JWT em todas as rotas) | João A. Cordeiro |
| 18/11/2024 | 23/11/2024 | Testes finais do sistema | André, João |

Tabela de Conteúdo

[**1. Introdução 5**](#_30j0zll)

[1.1 Objetivo 5](#_1fob9te)

[1.2 Escopo 6](#_3znysh7)

[**2. Descrição Geral do Sistema 6**](#_tyjcwt)

[2.1 Visão Geral 6](#_bgstot4d6r80)

[**3. Arquitetura do Sistema 7**](#_3dy6vkm)

[3.1 Diagrama de Arquitetura 7](#_1t3h5sf)

[3.2 Casos de Uso 7](#_4d34og8)

[3.3 Componentes do Sistema 9](#_2s8eyo1)

[3.3.1 Módulo de Geração de QR Codes 9](#_mjs9c0yf5kso)

[3.3.2 Módulo Leitor de QR Codes 10](#_hk5qmibxb9c7)

[3.3.3 API 10](#_km3djyf80yqp)

[3.3.3.1 Rotas e Endpoints 10](#_tt89up3sagpi)

[3.4 Tecnologias Utilizadas 12](#_ysbmoo67j6d2)

[3.4.1 Do Projeto 12](#_1sbm9e5fujdl)

[3.4.2 Da Documentação 12](#_i6hlzyc321yy)

[**4. Requisitos do Sistema 13**](#_r4bw6l4bc2kx)

[4.1 Requisitos Funcionais 13](#_dn2g2tu3v9vu)

[4.2 Requisitos Não Funcionais 13](#_pbsv9cmtl1f)

[**5. Modelagem de Dados 14**](#_jcmojh1w0hkb)

[5.1 Diagrama de Entidade-Relacionamento (ER) 14](#_r2icr26bl9zt)

[5.2 Estrutura das Tabelas 15](#_qki13kgcgx4u)

[**6. Design da Interface do Usuário (UI/UX) 15**](#_6bnzugjf139d)

[6.1 Wireframes ou Mockups 15](#_wtpfp1vtbzm8)

[**7. Considerações Finais 17**](#_pkqtfxptsycj)

[7.1 Perspectivas Futuras 17](#_px417klfu088)

[**8. Conclusão 18**](#_4ajekve7h28l)

[**9. Referências 19**](#_ipwxqopnb9ph)

Especificação de Requisitos de Software

# Introdução

A inovação tecnológica tem impactado diversas áreas, e a educação, com suas 2.574 instituições de ensino superior e mais de 22,6 milhões de vagas em cursos de graduação (INEP, 2022), é uma área que também não está fora dessa necessidade. Entre os desafios enfrentados, destaca-se o controle de presença dos alunos, ainda realizado de forma manual em muitas instituições. Para resolver esse problema, este projeto propõe um sistema automatizado que utiliza QR Codes gerados pelos alunos e lidos pelos professores, tornando o processo mais ágil e eficiente. Com o crescimento das matrículas e a expansão do ensino a distância (EAD), sendo que a rede privada concentrou 76,9% das matrículas em 2021 (Semesp, 2023), a implementação de tecnologias inovadoras é cada vez mais crucial para a eficiência das instituições. Além disso, o uso da Internet das Coisas (IoT) tem se mostrado promissor no contexto educacional, permitindo uma integração inteligente entre dispositivos e processos, o que pode ser aplicado para aprimorar a coleta e o gerenciamento de dados de presença em tempo real (Santos, 2019).

A justificativa surge da limitação dos métodos tradicionais de controle de presença, que consomem tempo valioso dos professores e estão sujeitos a falhas. Sendo possível solucionar esse problema, automatizando o processo e permitindo que as aulas sejam conduzidas de maneira mais fluida. A proposta de automação no controle de presença é uma resposta direta a essa realidade. Pesquisas indicam que o uso de tecnologia na educação pode melhorar significativamente a eficiência administrativa e pedagógica, além de proporcionar um acompanhamento mais preciso da participação dos alunos (Academia ERP, 2021).

## Objetivo

A finalidade deste projeto é desenvolver um sistema automatizado para o controle de presença de alunos em sala de aula, utilizando QR Codes e tecnologias de integração de sistemas. O objetivo é facilitar o processo de chamada, eliminando a necessidade de procedimentos manuais, otimizando o tempo dos professores e garantindo maior segurança e confiabilidade na verificação de presença acadêmica.

## Escopo

O sistema abrange quatro módulos principais:

1. Um aplicativo desenvolvido para dispositivos móveis para os alunos, responsável por gerar QR Codes;
2. Sistema de leitura de QR Codes para os professores, desenvolvido para funcionar via navegador;
3. Módulo administrativo para cadastro e gerenciamento de professores, disciplinas, alunos e horários de aula;
4. Servidor de aplicação, responsável pela integração de dados e pelo armazenamento das informações em um banco de dados PostgreSQL.

O sistema foi desenvolvido para funcionar em multiplataformas e está projetado para oferecer relatórios diários de presença e manutenção dos alunos, professores, disciplinas e cursos cadastrados.

# Descrição Geral do Sistema

O sistema proposto visa automatizar o processo de chamada de alunos em instituições de ensino, promovendo eficiência e reduzindo erros humanos. Através de uma aplicação multiplataforma, QR Codes são gerados e lidos em tempo real para registro de presença, garantindo segurança e integridade dos dados. O sistema também permite o gerenciamento de cadastros de alunos, professores e disciplinas, oferecendo relatórios automatizados para acompanhamento da frequência acadêmica.

## Visão Geral

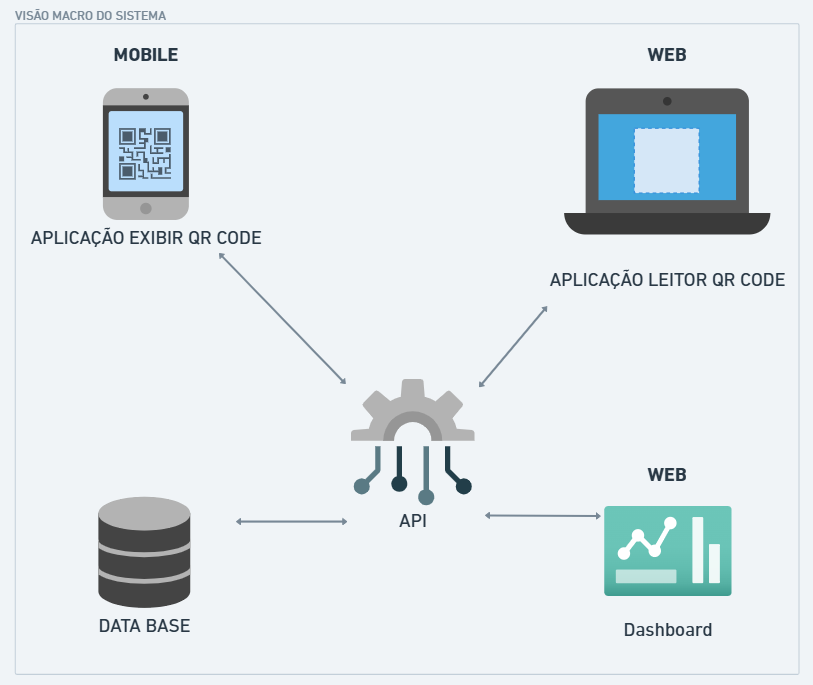
Este sistema representa uma inovação significativa no ambiente educacional, ao oferecer uma solução moderna para o controle de presença, fundamentada em tecnologias de automação, que têm se mostrado eficazes na melhoria da eficiência administrativa e na precisão dos registros (Srhir, 2024). A automação permite que instituições acadêmicas gerenciem a participação dos alunos de maneira mais eficaz. Além disso, com o uso de Internet das Coisas (IoT), é possível integrar dispositivos inteligentes para garantir maior confiabilidade no registro de frequência, assim como em cenários de ensino híbrido (IT Experts, 2023). Com isso, o sistema libera tempo dos professores para atividades de maior valor e fornece relatórios automáticos que possibilitam um acompanhamento acadêmico preciso.

# Arquitetura do Sistema

## Diagrama de Arquitetura

O diagrama de arquitetura do sistema ilustra como os diferentes módulos interagem entre si, mostrando as conexões entre o aplicativo móvel, o sistema de leitura, o módulo administrativo e o servidor de aplicação. A arquitetura adota uma abordagem cliente-servidor, com um banco de dados centralizado para garantir a persistência dos dados conforme é visível na figura 1.

Figura 1 - Representação Macro da Arquitetura do Sistema.

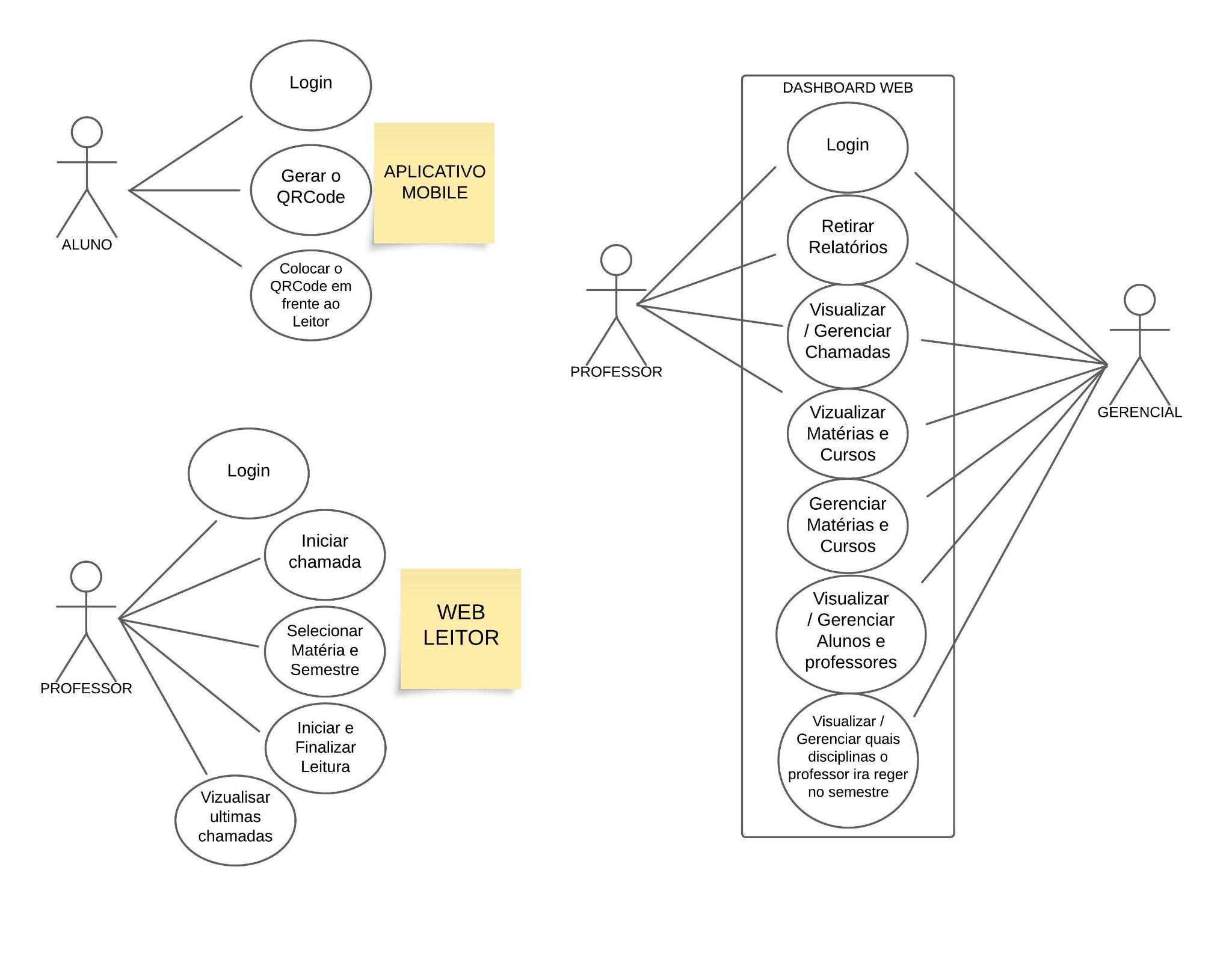


Fonte: Elaborado pelos autores (2024)

## Casos de Uso

Os casos de uso representam cenários específicos que descrevem como os usuários interagem com o sistema. Abaixo, detalha-se os principais casos de uso representados na figura 2.

Figura 2 - Casos de Uso do Sistema

**

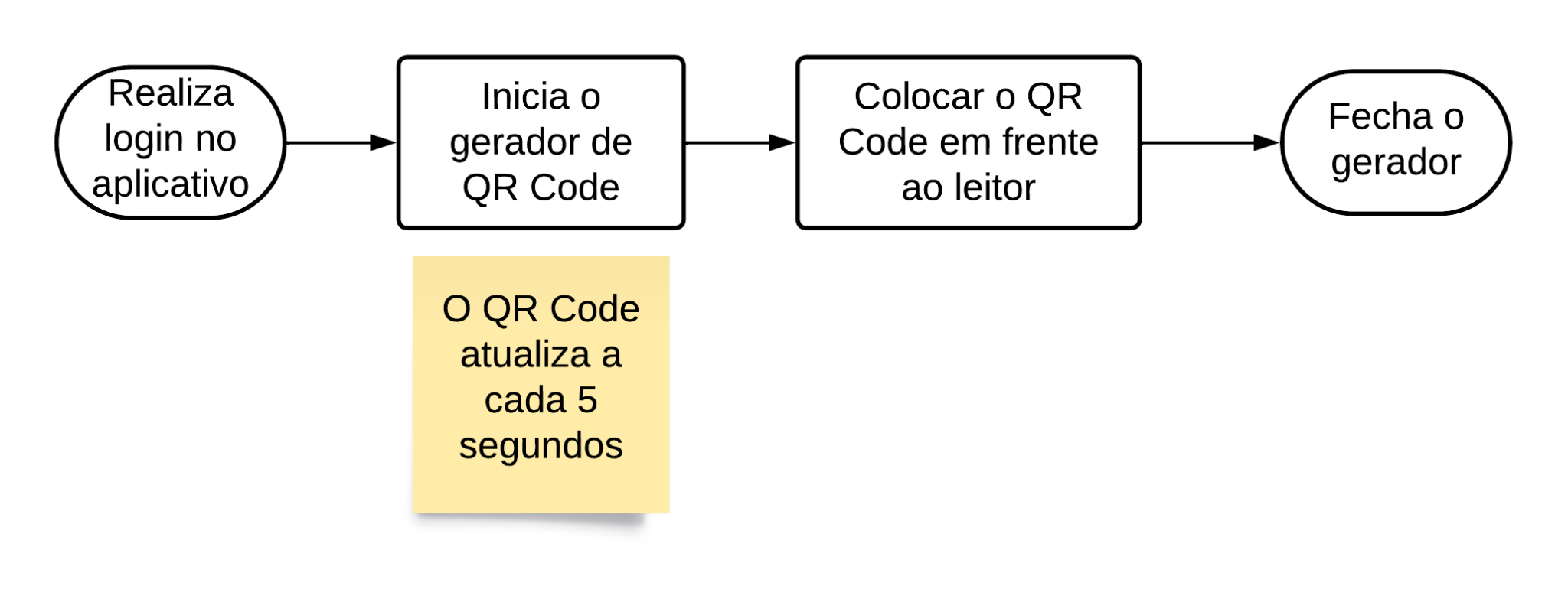
Fonte: Elaborado pelos autores (2024).

1. O aluno terá acesso ao aplicativo mobile onde irá realizar o login, gerar o qrcode e colocar o seu celular em frente ao leitor para registrar sua presença em aula.
2. O professor no acesso ao web leitor onde irá realizar o login, iniciar e finalizar as chamadas/leituras e também terá uma visualização breve de suas últimas chamadas iniciadas.
3. O professor no acesso a dashboard irá fazer o login, tirar relatórios de presenças de suas chamadas, poderá gerenciar as presenças adicionando ou removendo alunos da chamada e também pode visualizar a quais matérias e cursos ele está vinculado.
4. O gerencial terá o acesso geral a dashboard, fazendo o login, terá acesso às chamadas de todos os professores, poderá retirar relatórios das chamadas, gerencias os alunos e professores, gerenciar suas matérias e suas turmas entre outros.

## Componentes do Sistema

### **3.3.1 Módulo de Geração de QR Codes**

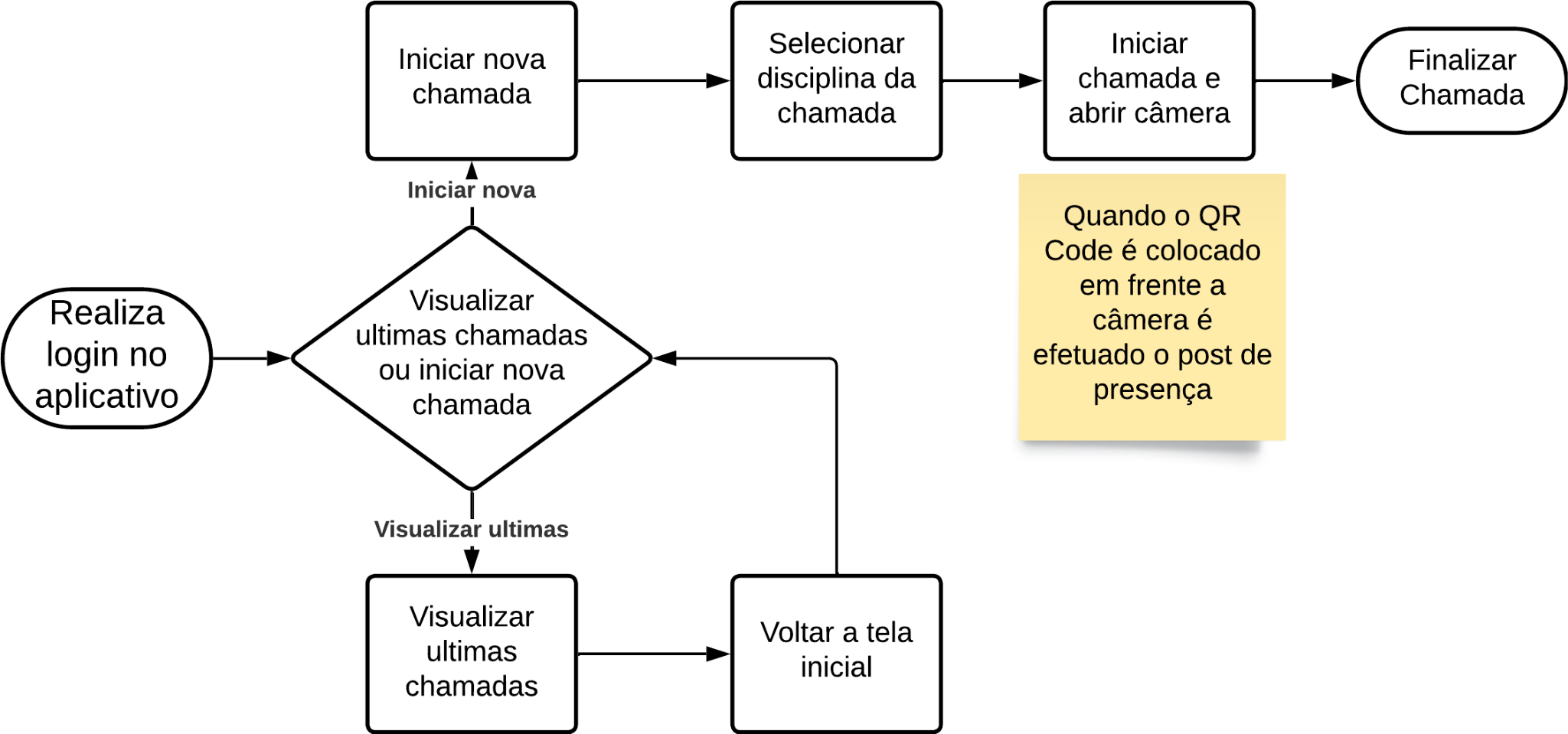
Permite que os alunos gerem QR Codes através de um aplicativo, onde o QR Code carrega informações de data e hora. O login no aplicativo é feito de maneira única onde será armazenado o código identificador único do dispositivo, impedindo que o aluno faça o login em outro celular.



A atualização do QR-Code acontece a cada 5 segundos, isso dificulta o compartilhamento do código QR com outras pessoas. Como o código do aluno é armazenado dentro do dispositivo, a única forma de se obter um Qr-Code válido é no dispositivo já cadastrado.

### 3.3.2 Módulo Leitor de QR Codes

Este módulo permite que os professores escaneem os códigos gerados pelos alunos através de um site web para que seja multiplataforma. A leitura instantânea atualiza automaticamente o banco de dados com a presença registrada.



Para que a presença seja registrada, a diferença dos tempos do QR Code e do servidor não seja maior que 5 segundos. Neste módulo o professor também terá uma visualização resumida de suas últimas chamadas realizadas.

### **3.3.3 API**

Centraliza a comunicação entre os módulos e o banco de dados, fazendo as validações dos dados, fazendo autenticação de acesso e manipulação do banco de dados com ORM.

#### **3.3.3.1 Rotas e Endpoints**

Todas as rotas possuem um CRUD básico (get, post, put e delete) e algumas contendo algumas verificações específicas para as suas operações.

Tabela 1: Rotas utilizadas pelos sistemas atualmente

| /usuarios | Informações referentes ao cadastro de usuários dos alunos, professores e gerencial | |
| --- | --- | --- |
| POST | /login | email: , senha: ,imei: |
| POST | /loginWeb | email: ,senha: |
| /semestre | Informações referentes ao vínculo de professor, disciplina e semestre | |
| GET | /professor/:id\_professor | Quais são as disciplinas do professor |
| /chamadas | Informações sobre as chamadas realizadas pelos professores | |
| POST | / | id\_professor: , id\_disciplina: , id\_semestre: , data\_hora\_inicio: |
| PUT | /finalizar | id: , data\_hora\_final: |
| GET | /professor/:id\_professor | Quais são as chamadas do professor |
| /chamada | Informações sobre as presenças dos alunos nas aulas | |
| POST | /alunos | hora\_post: ,id\_chamada: ,id\_aluno: |

Tabela 2: Rotas a serem utilizadas

| /usuarios | Informações referentes ao cadastro de usuários dos alunos, professores e gerencial | |
| --- | --- | --- |
| /semestres | Informações referentes ao cadastro dos semestres anuais | |
| /semestre | Informações referentes ao vínculo de professor, disciplina e semestre | |
| /chamadas | Informações sobre as chamadas realizadas pelos professores | |
| /chamada | Informações sobre as presenças dos alunos nas aulas | |
| /disciplinas | Informações referentes ao cadastro das disciplinas | |
| /cursos | Informações referentes ao cadastro dos cursos | |
| /turmas | Informações referentes ao cadastro das turmas | |
| /turma/alunos | Informações referentes ao vínculo que mantém o aluno pertencente a determinada turma | |
| /turma/disciplinas | Informações referentes ao vínculo que mantém a disciplina pertencente a determinada turma | |

## Tecnologias Utilizadas

### 3.4.1 Do Projeto

1. JavaScript - Uma linguagem de programação amplamente usada em diversos ambientes, do front-end web a servidores, aplicativos móveis e sistemas IoT.

(Mozilla, 2022)

1. React - Biblioteca JavaScript para criação de interfaces de usuário e também serve como base para gerenciar a lógica de componentes e estados complexos.

(React, 2023)

1. Node JS - Ambiente de execução que permite executar JavaScript no lado do servidor. (Node.JS, 2023)
2. Prisma - ORM para manipulação de banco de dados em Node.js.

(Prisma, 2023)

1. Expo - Ferramenta para desenvolvimento de aplicativos móveis com React Native. (Expo, 2023)
2. Postgres Vercel - Banco de Dados alocado e disponível na web a partir da Vercel. (Vercel, 2023)
3. Express - Framework para desenvolvimento de API. (Express.js, 2024)

### 3.4.2 Da Documentação

1. Whimsical - Plataforma para diagramas e wireframes. (Whimsical, 2024)
2. Dbdiagram - Ferramenta para modelagem de banco de dados. (Dbdiagram.io, 2024)
3. Figma - Ferramenta de design colaborativo para prototipagem. (Figma, 2024)
4. LucidChart - Software de criação de diagramas de fluxo e mapas mentais. (Lucidchart, 2024)

# Requisitos do Sistema

## Requisitos Funcionais

1. Registro de Usuários: O sistema deve permitir o cadastro de alunos e professores com dados pessoais e acadêmicos.
2. Controle de Presença: Implementação de uma verificação de presença com QR Codes, facilitando a confirmação de presença pelo professor.
3. Relatórios de Frequência e Chamadas: Geração automática de relatórios de frequência e chamadas, acessíveis aos administradores.
4. Gerenciamento de Turmas e Disciplinas: Gerenciais devem conseguir criar e gerenciar turmas e disciplinas.

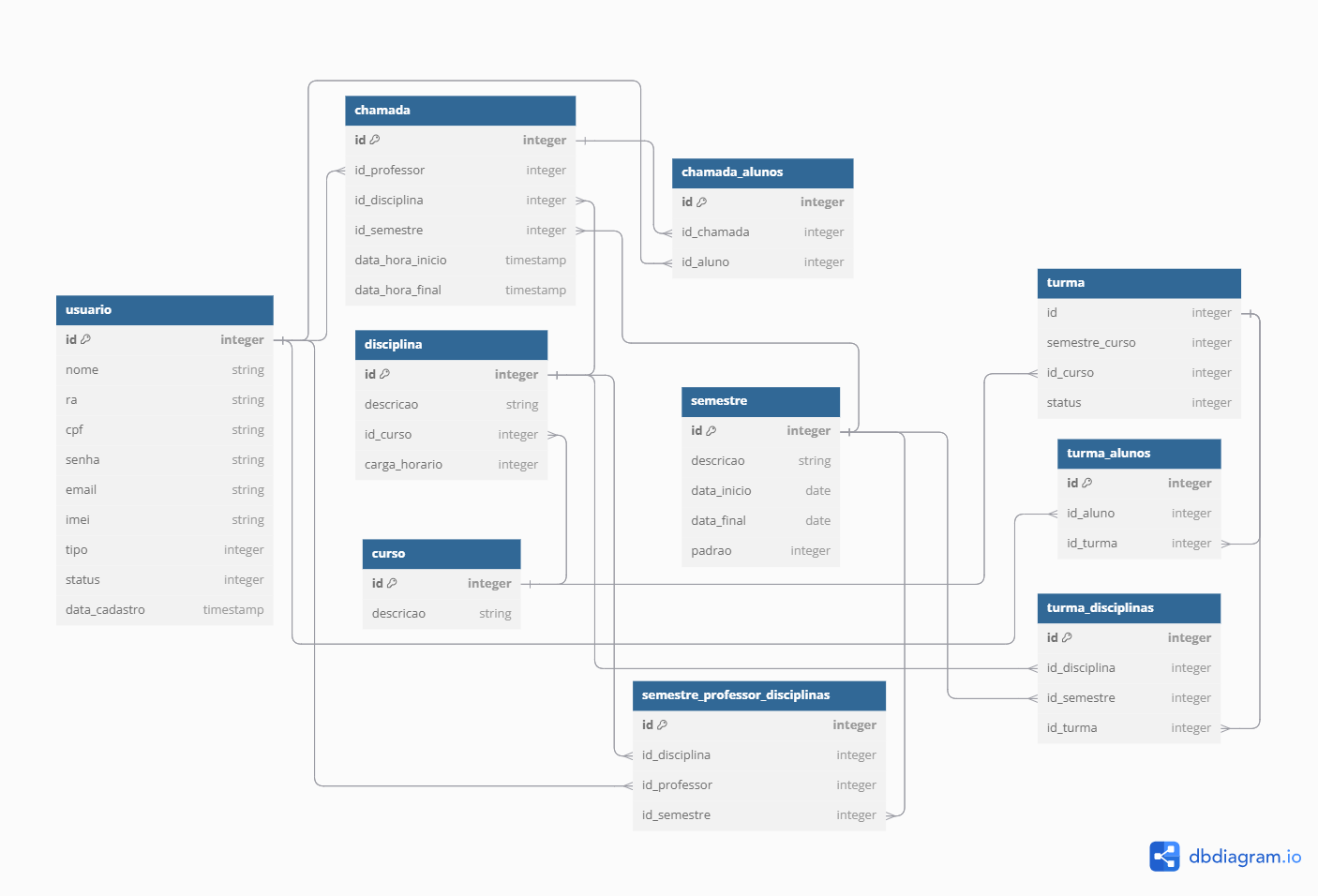
## Requisitos Não Funcionais

1. Segurança: Autenticação para proteger dados de alunos e professores, garantindo o sigilo das informações.
2. Usabilidade: Interfaces intuitivas e de fácil navegação, especialmente em dispositivos móveis.
3. Compatibilidade e Leveza: O sistema deve ser leve e acessível em dispositivos de diferentes capacidades, permitindo uma experiência fluida independentemente da configuração do dispositivo.
4. Escalabilidade: O sistema deve suportar aumento no número de usuários sem perder desempenho.

# Modelagem de Dados

## Diagrama de Entidade-Relacionamento (ER)

Figura 3 - Diagrama de tabelas e relacionamentos



Fonte: Elaborado pelos autores (2024).

## Estrutura das Tabelas

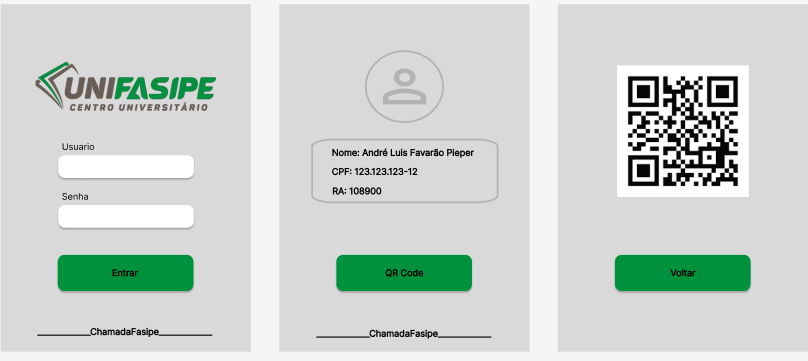
O banco de dados é composto por tabelas principais que representam os elementos-chave do sistema, como usuário, semestre, curso, disciplina, turma, e chamada. Essas tabelas armazenam informações essenciais, como dados de usuários, turmas, disciplinas e registros de chamada. As tabelas de vínculo, como turma\_alunos e turma\_disciplinas, associam turmas a alunos e disciplinas, respectivamente, enquanto semestre\_professor\_disciplinas vincula professores a disciplinas em um semestre específico para que seja possível visualizar a matéria que o professor irá reger naquele determinado semestre. A tabela chamada\_alunos registra a presença dos alunos nas chamadas iniciadas pelos professores, permitindo monitorar a frequência acadêmica de maneira precisa.

As relações entre as tabelas garantem o gerenciamento integrado de alunos, professores, disciplinas, turmas e presença, facilitando a automação, organização dos dados acadêmicos e permitindo a realização de relatórios de forma eficiente e específica.

# Design da Interface do Usuário (UI/UX)

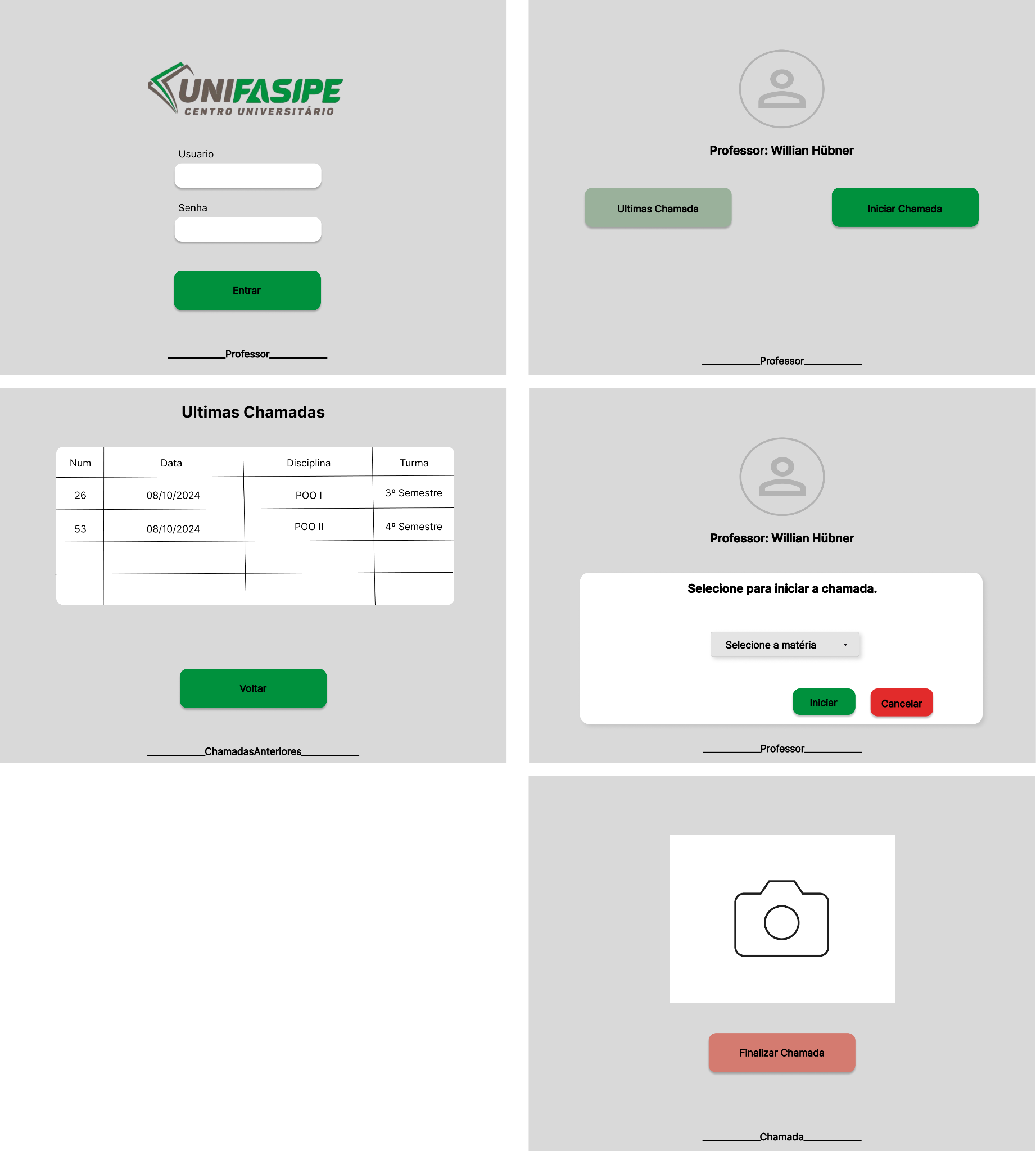
## Wireframes ou Mockups

Figura 4 - Protótipo das telas do aplicativo gerador de QR Code

**

Fonte: Elaborado pelos autores (2024).

Figura 5 - Protótipo das telas do site leitor de QR Code



Fonte: Elaborado pelos autores (2024).

# Considerações Finais

O desenvolvimento deste projeto busca criar uma solução inovadora e prática para a gestão de presença acadêmica, visando integrar tecnologia de forma prática e segura no ambiente educacional. Utilizando recursos como validação de tempo, o sistema visa melhorar o acompanhamento da presença dos alunos, tornando o processo mais eficiente e menos suscetível a falhas.

## Perspectivas Futuras

Para estudos futuros propõe-se aprimoramentos que podem tornar o sistema ainda mais completo e eficiente:

1. Implantação do sistema em campo para testes e maturação.
2. Adicionar validação de geolocalização: A geolocalização pode adicionar uma camada de segurança à validação já existente de tempo, incrementando ainda mais na segurança e mantendo a transparência nas presenças apenas daqueles que estão em sala.
3. Implementar a dashboard para a administração de cadastros, vínculos e gerenciamento relatorial: Isso permitirá ao professor ter uma visualização detalhada de suas chamadas realizadas, das presenças de cada aluno em suas chamadas possibilitando alterações para inclusão ou exclusão de presença para casos específicos. Também sendo o módulo para o usuário gerencial fazer todos os vínculos necessários entre aluno, professor, disciplina, turma, semestre e curso.
4. Integração com Google Account: Login com contas Google, método já popularizado que permite facilitar e centralizar o login utilizando as contas registradas na plataforma.
5. Criptografia com base em chave de segurança: Implementando uma nova camada de segurança onde com a criptografia se torna mais seguro o tráfego de dados pelo sistema de forma que caso um pacote seja interferido ele não será aberto sem que o interceptor saiba a criptografia e a chave de segurança.

# Conclusão

Este trabalho documentou e desenvolveu um sistema inovador para o controle de presença acadêmica, promovendo automação, precisão e acessibilidade para o ambiente educacional. O objetivo foi integrar tecnologias modernas, como validações por tempo e, futuramente, geolocalização, para garantir a autenticidade dos registros e aumentar a eficiência do processo de gestão de dados acadêmicos. Durante o projeto, adotaram-se práticas de engenharia de software e requisitos vistos durante as aulas decorridas no 4º e 3º semestre (2024/1 e 2024/2) para estruturar uma solução robusta e funcional.

A criação do sistema foi pautada e idealizada na análise de necessidades reais dos usuários, como professores e administradores, garantindo que as funcionalidades atendam às demandas do ensino presencial e se necessário também do híbrido. A interação com os principais envolvidos e realização do projeto já estando no meio em que ele será implementado possibilitou identificar desafios específicos e elaborar uma ferramenta adaptável, simples e escalável, que poderá ser expandida para incluir recursos gerenciais e relatórios detalhados a partir da dashboard.

Com o uso desse sistema, espera-se não apenas otimizar o processo de verificação de presença, mas também criar um ambiente acadêmico mais organizado e transparente e também fomentar a realização de grandes projetos por parte dos acadêmicos do curso de Análise e Desenvolvimento de Sistemas.

# Referências

Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (INEP). Notas Estatísticas do Ensino Superior. Revista eletrônica 2022.

<https://download.inep.gov.br/publicacoes/institucionais/estatisticas_e_indicadores/notas_estatisticas_censo_da_educacao_superior_2021.pdf> Acesso em: 17 set. 2024

Santos, J. et al. "Automação e Internet das Coisas (IoT) no Contexto Educacional". Revista de Tecnologia e Inovação, 2019.

<https://www.doi.org/10.34117/bjdv5n11-292> Acesso em: 17 set. 2024

Instituto Semesp. "Evolução do Número de Matrículas no Ensino Superior Brasileiro", com base no INEP, 2021. Revista eletrônica 2023.

<https://www.semesp.org.br/mapa/edicao-13/brasil/matriculas/> Acesso em: 02 out. 2024

Academia ERP. “Technology and Automation in Education”. Revista eletrônica 2021: <https://www.academiaerp.com/blog/automation-in-education/> Acesso em: 06 out. 2024

Srhir, A. et al. "Smart Education in the IoT: Issues, Architecture, and Challenges". Revista eletrônica 2024.

<https://link.springer.com/chapter/10.1007/978-3-031-54376-0_35> Acesso em: 19 out. 2024

ITExperts. “O que é automação escolar e como fazer?” Revista eletrônica 2023.

[https://www.itexperts.com.br/blog/tecnologia-e-educacao-blog/automacao-escolar/#:~:text=No%20quesito%20economia%2C%20a%20automa%C3%A7%C3%A3o,dentro%20da%20sala%20de%20aula.](https://www.itexperts.com.br/blog/tecnologia-e-educacao-blog/automacao-escolar/#:~:text=No%20quesito%20economia%2C%20a%20automa%C3%A7%C3%A3o,dentro%20da%20sala%20de%20aula) Acesso em: 19 out. 2024

REACT. *Getting Started - Legacy React Documentation*. ReactJS.org, 2023. Disponível em:<https://pt-br.legacy.reactjs.org/docs/getting-started.html>. Acesso em: 22 nov. 2024.

NODE.JS. *Node.js Documentation*. Nodejs.org, 2023. Disponível em:<https://nodejs.org/docs/latest/api/documentation.html>. Acesso em: 22 nov. 2024.

PRISMA. *Prisma Documentation*. Prisma.io, 2023. Disponível em:<https://www.prisma.io/docs>. Acesso em: 22 nov. 2024.

EXPO. *Introduction - Expo Documentation*. Expo.dev, 2023. Disponível em:<https://docs.expo.dev/tutorial/introduction/>. Acesso em: 22 nov. 2024.

VERCEL. *Quickstart - Vercel Postgres*. Vercel.com, 2023. Disponível em:<https://vercel.com/docs/storage/vercel-postgres/quickstart>. Acesso em: 22 nov. 2024.

EXPRESS.JS. *Express - Framework para Node.js*. Expressjs.com, 2024. Disponível em:<https://expressjs.com/pt-br/>. Acesso em: 22 nov. 2024.

WHIMSICAL. *Whimsical Documentation*. Whimsical.com, 2024. Disponível em:<https://whimsical.com/docs>. Acesso em: 22 nov. 2024.

DBDIAGRAM.IO. *Documentation*. dbdiagram.io, 2024. Disponível em:<https://docs.dbdiagram.io/>. Acesso em: 22 nov. 2024.

FIGMA. *Central de ajuda*. Figma, 2024. Disponível em:<https://help.figma.com/hc/pt-br>. Acesso em: 22 nov. 2024.

LUCIDCHART. *Documentation and implementation: Level 1*. Disponível em:<https://www.lucidchart.com/pages/templates/documentation-and-implementation-level-1>. Acesso em: 22 nov. 2024.