

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
CAMPUS GUARABIRA
DIRETORIA DE DESENVOLVIMENTO DE ENSINO
COORDENAÇÃO DO CURSO DE TECNOLOGIA EM SISTEMAS PARA INTERNET

RELATÓRIO TÉCNICO

Projeto Integrador em Sistemas para Internet (PISI)

Documentação do Produto Implementado por Software: SafeKey GUARABIRA - PB

Integrante

CAIO RAFAEL DOS SANTOS SILVA

IAGO BRUNO SILVA FIGUEREDO

GUARABIRA – PB

2025

CAIO RAFAEL DOS SANTOS SILVA

IAGO BRUNO SILVA FIGUEREDO

RELATÓRIO TÉCNICO

Documentação de Produto Implementado por Software: SafeKey

Trabalho de Conclusão de Curso apresentado ao Curso Sistemas para Internet, do Instituto Federal da Paraíba – Campus Guarabira, em cumprimento às exigências parciais para a obtenção do título Tecnólogo em Sistemas para Internet.

ORIENTADOR: Rodrigo Leone Alves

COORIENTADOR: Nasson Paulo Sales Neves

**GUARABIRA – PB
2025**

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA DO IFPB - GUARABIRA

S586d Silva, Caio Rafael dos Santos
Documentação de Produto Implementado por Software: SafeKey / Caio
Rafael dos Santos Silva; Iago Bruno Silva Figueredo.- Guarabira, 2025.
29 f.: il; color.

Trabalho de Conclusão de Curso (Superior de Tecnologia em
Sistemas para Internet). – Instituto Federal da Paraíba, Campus Guarabira,
2025.

"Orientação: Prof. Dr. Rodrigo Leone Alves"

"Co-orientador: Prof. Dr. Nasson Paulo Sales Neves."

Referências.

1. Sistemas para Internet. 2. Gestão institucional. 3. Ficha técnica. 4.
Gerencialmente de salas. I. Título.

CDU 004.4(0.067)



MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
CAMPUS GUARABIRA

ATA 9/2025 - CCSTSI/DDE/DG/GB/REITORIA/IFPB

ATA DE APRESENTAÇÃO E DEFESA DE TRABALHO DE CONCLUSÃO DE CURSO
CCS de Tecnologia em Sistemas para Internet

Aos 13 de março de 2025, às 13:30, no Laboratório de Informática, reuniram-se os membros da banca avaliadora, Rodrigo Leone Alves (Orientador), Nasson Paulo Sales Neves (Coorientador), Rhavy Maia Guedes (Examinador Interno), Lucas Vieira de Souza (Examinador Interno), para avaliarem a apresentação do *Trabalho de Conclusão do Curso Superior de Tecnologia em Sistemas para Internet* (Relatório Final do Projeto Integrador em Sistemas para Internet - PISI) do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba (IFPB), *campus* Guarabira, desenvolvido pela aluno(a) **Caio Rafael dos Santos Silva**, matrícula de nº **202213810031**, intitulado "SafeKey", protocolado para apresentação de acordo com os requisitos expostos no Projeto Pedagógico de Curso de Tecnologia em Sistemas para Internet. Após a apresentação, a banca apresentou, por unanimidade, pareceres a favor da aprovação do trabalho. Desta forma, o Trabalho de Conclusão de Curso foi **aprovado** e definiu-se a **nota final 95 (Noventa e Cinco)**.

Nada mais havendo a tratar, às 16:45, encerraram-se os trabalhos, determinando a lavratura desta ata, que, após lida e considerada conforme, será assinada pelos presentes. Eu, Rodrigo Leone Alves, lavrei a presente ata.

Guarabira/PB, em 13 de março de 2025.

Documento assinado eletronicamente por:

- **Rodrigo Leone Alves**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 20/03/2025 13:25:13.
- **Lucas Vieira de Souza**, COORDENADOR(A) DE CURSO - FUCL - CCSTSI-GB em 20/03/2025 16:56:53.
- **Rhavy Maia Guedes**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 20/03/2025 19:20:58.



Documento assinado digitalmente

NASSON PAULO SALES NEVES
Data: 22/03/2025 16:26:54-0300
Verifique em <https://validar.itl.gov.br>

Este documento foi emitido pelo SUAP em 20/03/2025. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/autenticar-documento/> e forneça os dados abaixo:

Código 685268
Verificador: e979541986
Código de Autenticação:





MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
CAMPUS GUARABIRA

ATA 10/2025 - CCSTSI/DDE/DG/GB/REITORIA/IFPB

ATA DE APRESENTAÇÃO E DEFESA DE TRABALHO DE CONCLUSÃO DE CURSO
CCS de Tecnologia em Sistemas para Internet

Aos 13 de março de 2025, às 13:30, no Laboratório de Informática, reuniram-se os membros da banca avaliadora, Rodrigo Leone Alves (Orientador), Nasson Paulo Sales Neves (Coorientador), Rhavy Maia Guedes (Examinador Interno), Lucas Vieira de Souza (Examinador Interno), para avaliarem a apresentação do *Trabalho de Conclusão do Curso Superior de Tecnologia em Sistemas para Internet* (Relatório Final do Projeto Integrador em Sistemas para Internet - PISI) do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba (IFPB), *campus* Guarabira, desenvolvido pela aluno(a) **Iago Bruno Silva Figueredo**, matrícula de nº **202213810022**, intitulado "SafeKey", protocolado para apresentação de acordo com os requisitos expostos no Projeto Pedagógico de Curso de Tecnologia em Sistemas para Internet. Após a apresentação, a banca apresentou, por unanimidade, pareceres a favor da aprovação do trabalho. Desta forma, o Trabalho de Conclusão de Curso foi **aprovado** e definiu-se a **nota final 95 (Noventa e Cinco)**.

Nada mais havendo a tratar, às 16:45, encerraram-se os trabalhos, determinando a lavratura desta ata, que, após lida e considerada conforme, será assinada pelos presentes. Eu, Rodrigo Leone Alves, lavrei a presente ata.

Guarabira/PB, em 13 de março de 2025.

Documento assinado eletronicamente por:

- **Rodrigo Leone Alves**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 20/03/2025 13:28:02.
- **Lucas Vieira de Souza**, COORDENADOR(A) DE CURSO - FUCL - CCSTSI-GB em 20/03/2025 16:56:46.
- **Rhavy Maia Guedes**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 20/03/2025 19:20:27.

gov.br

Documento assinado digitalmente

NASSON PAULO SALES NEVES
Data: 22/03/2025 16:26:54-0300
Verifique em <https://validar.iti.gov.br>

Este documento foi emitido pelo SUAP em 20/03/2025. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/autenticar-documento/> e forneça os dados abaixo:

Código: 685274
Verificador: 299c860f35
Código de Autenticação:



RESUMO

Este projeto propõe o desenvolvimento de um sistema web para controle e gerenciamento de salas em instituições de ensino, garantindo um registro seguro de requisições e devoluções, além de um painel administrativo para monitoramento em tempo real. A falta de um controle eficiente pode resultar em acessos não autorizados, extravios e dificuldades na administração dos espaços. Para solucionar esse problema, o sistema foi desenvolvido utilizando tecnologias como React, Django e PostgreSQL. A proposta busca melhorar a segurança, organização e rastreabilidade no uso das salas, otimizando a gestão institucional.

Palavras-chave: Ficha Técnica da Preparação, software, sistema web, gerenciamento de salas, gestão institucional

ABSTRACT

This project proposes the development of a web system for the control and management of rooms in educational institutions, ensuring secure records of requests and returns, as well as an administrative panel for real-time monitoring. The lack of efficient control can result in unauthorized access, losses, and difficulties in managing the spaces. To solve this problem, the system was developed using technologies such as React, Django, and PostgreSQL. The proposal aims to improve security, organization, and traceability in room usage, thereby optimizing institutional management.

Keywords: Technical Preparation Sheet, software, web system, room management, institutional management

SUMÁRIO

1. INTRODUÇÃO.....	9
1.2. DELIMITAÇÃO DO PROBLEMA.....	9
1.3. OBJETIVO GERAL.....	9
1.4. OBJETIVOS ESPECÍFICOS.....	10
1.5. JUSTIFICATIVA.....	10
1.6. MÉTODO DO TRABALHO.....	10
1.7. LEI GERAL DE PROTEÇÃO DE DADOS (LGPD).....	10
2. DESCRIÇÃO GERAL DO SISTEMA.....	11
2.1. DESCRIÇÃO DO PROBLEMA.....	11
2.2. PERFIS DE USUÁRIOS.....	11
2.3. REQUISITOS.....	12
2.3.1. REQUISITOS FUNCIONAIS.....	12
2.3.2. REQUISITOS NÃO FUNCIONAIS.....	14
3. ANÁLISE E DESIGN.....	15
3.1. DIAGRAMA DE CLASSES.....	16
3.2. DIAGRAMA DE CASOS DE USO.....	17
3.3. DIAGRAMA DE ATIVIDADES.....	18
3.4. DIAGRAMA DE OBJETOS.....	19
3.5. MODELO DE DADOS.....	20
3.5.1. MODELO LÓGICO DA BASE DE DADOS.....	20
3.5.2. DIAGRAMA DE RELACIONAMENTO DE ENTIDADES.....	21
3.5.3. DICIONÁRIO DE DADOS.....	22
3.6. FERRAMENTAS DE DESENVOLVIMENTO.....	24
4. IMPLEMENTAÇÃO.....	26
4.1. Backend.....	26
4.2. Frontend.....	28
5. CONCLUSÃO.....	29
6. REFERÊNCIAS.....	30

1. INTRODUÇÃO

O gerenciamento de chaves em ambientes institucionais é um desafio encontrado principalmente em instituições de ensino, onde existem vários usuários que necessitam acessar salas e laboratórios em diferentes horários. A falta de um sistema eficiente para realizar esse gerenciamento pode acabar ocasionando em problemas como a perda de chaves, dificuldades no controle de acessos e confusão em relação à disponibilidade das salas.

A cada dia que passa, a tecnologia tem se mostrado uma aliada poderosa na otimização de processos administrativos. Esse sistema web oferece uma solução inovadora para esse problema, permitindo o agendamento, controle e registro do uso de chaves de forma organizada e segura.

1.2. DELIMITAÇÃO DO PROBLEMA

O gerenciamento de chaves em instituições educacionais é fundamental para assegurar um acesso seguro e bem estruturado às salas e laboratórios. No entanto, diversas instituições continuam a usar registros manuais, como planilhas em papel ou anotações avulsas, o que pode resultar em perdas, acessos indevidos e complicações na gestão. A ausência de um sistema eficaz prejudica a rastreabilidade e a clareza do processo, tornando a administração mais difícil para os responsáveis.

1.3. OBJETIVO GERAL

O sistema web proposto tem como visão modernizar e aprimorar a gestão de salas em instituições de ensino, eliminando falhas comuns nos métodos convencionais de gerenciamento. Ao registrar de forma digital as requisições e devoluções, ele garante mais segurança e rastreabilidade. A solução minimiza falhas no controle manual, melhora a organização das salas e facilita a administração por parte dos responsáveis.

1.4. OBJETIVOS ESPECÍFICOS

- Criar um sistema para registrar requisições de uso das salas.
- Implementar dashboards para acompanhar o fluxo de uso das salas.
- Garantir um controle seguro do histórico de acessos.
- Facilitar a administração e o agendamento de salas por meio da tecnologia.

1.5. JUSTIFICATIVA

A ausência de um sistema para o gerenciamento das salas pode ocasionar dificuldades em sua gestão, resultando em problemas de segurança e eficiência. Um sistema web oferece um controle aprimorado, melhorando a gestão do acesso às salas e promovendo maior transparência na utilização dos recursos da instituição.

1.6. MÉTODO DO TRABALHO

Para garantir a organização e eficiência no desenvolvimento do sistema, foi implementada uma metodologia ágil, possibilitando entregas iterativas e ajustes com maior agilidade durante todo o processo. O Scrum foi escolhido como modelo principal, dividindo o desenvolvimento em ciclos curtos chamados sprints, que permitiram o monitoramento contínuo do progresso e a implementação de melhorias de forma ágil e flexível.

1.7. LEI GERAL DE PROTEÇÃO DE DADOS (LGPD)

Na implementação do software “SafeKey” e em conformidade com a Lei Geral de Proteção de Dados, o sistema adotará medidas específicas para assegurar a segurança de dados de privacidade de cada usuário administrador, professor ou aluno. A autenticação para acessar o sistema e as funcionalidades de cada usuário será somente através de um formulário de login com credenciais únicas onde somente usuários autorizados com verificações de acesso.

2. DESCRIÇÃO GERAL DO SISTEMA

O sistema “SafeKey” é uma plataforma que visa proporcionar uma gestão mais centralizada e eficiente para o uso das salas de uma instituição. O sistema contempla com três tipos de usuários - administrador, professor e aluno - nos quais compartilham as principais funcionalidades de gerenciamento de reservas das salas.

A interface do site foi desenvolvida de forma intuitiva com a finalidade de proporcionar uma experiência agradável aos usuários e usabilidade simples.

As funcionalidades principais do sistema foram:

- Cadastro de usuários
- Cadastro de salas
- Cadastro de reservas
- Listagem de usuários
- Listagem de salas
- Listagem de reservas
- Envio de notificações de avisos

2.1. DESCRIÇÃO DO PROBLEMA

A falta de um sistema eficiente para realizar o gerenciamento das salas e laboratórios utilizados pela instituição, podem ocasionar conflitos entre os professores. Atualmente, o processo é realizado de forma manual, o que pode levar a situações em que o professor planeja uma aula em um laboratório, mas ao chegar no horário da aula, descobre que outro professor pegou a chave e já está utilizando a sala. Essa falta de organização acaba impactando na dinâmica das aulas, forçando os professores a mudarem aquilo que tinham planejado de última hora.

2.2. PERFIS DE USUÁRIOS

UR_01: Administrador (Super usuário) - Os usuários finais do sistema são administradores responsáveis pela gestão principal do sistema de controle das chaves. Eles utilizam o sistema conforme necessário para realizar tarefas administrativas como criar, deletar, modificar e gerenciar salas, chaves e usuários, como também fazer requerimento e devolução das chaves às salas de aulas.

UR_02: Docente (Professor) - Os usuários finais são os professores docentes que ministram as aulas nas salas. Eles utilizam o sistema diariamente para gerenciar suas aulas nas salas desejáveis, podendo criar novos usuários e gerenciar as salas, chaves e usuários vinculados ao mesmo. Podem fazer requerimento e devolução das chaves às salas de aulas.

UR_03: Aluno - Os usuários finais são os alunos, gerenciados e criados pelos professores ou administradores. Eles utilizam o sistema somente quando necessário e for pedido pelo usuário, docente ou administrador, que está vinculado a ele podendo fazer o requerimento e devolução das chaves às salas de aulas.

2.3. REQUISITOS

2.3.1. REQUISITOS FUNCIONAIS

RF_01: Login - O sistema deve permitir acesso de usuários através de uma página de login.

Prioridade: Essencial Importante Desejável

RF_02: Cadastro de usuários - O sistema deve ser capaz de cadastrar novos acessos ao mesmo. Somente usuários já cadastrados no sistema, com permissão para cadastrar, podem dar acesso a novos usuários.

Prioridade: Essencial Importante Desejável

RF_03: Cadastro de novas salas - O sistema deve ter um formulário de cadastro para ter acesso a cadastrar novas salas que tenham uma chave para acessá-la.

Prioridade: Essencial Importante Desejável

RF_04: Requerimento da reserva - O sistema deve permitir o usuário fazer um requerimento de uma reserva para uma sala seleciona através de um formulário

Prioridade: Essencial Importante Desejável

RF_05: Listagem dos usuários cadastrados - O sistema deve ter uma página para listar todos os usuários cadastrados no sistema com suas devidas informações.

Prioridade: Essencial Importante Desejável

RF_06: Listagem de todas as salas cadastradas - O sistema deve manter ter uma página para listar todas as salas cadastradas com suas devidas informações.

Prioridade: Essencial Importante Desejável

RF_07: Listagem de todas as reservas cadastradas - O sistema deve manter ter uma página para listar todas as reservas cadastradas pelos usuários com suas devidas informações.

Prioridade: Essencial Importante Desejável

RF_08: Coleta e devolução das chaves - O sistema deve permitir que um usuário possa marcar que a chave foi retirada ou devolvida ao seu depósito.

Prioridade: Essencial Importante Desejável

RF_09: Indicar se a sala está em uso - O sistema deve ser capaz de mostrar se a sala está em uso na tela de listagem das salas.

Prioridade: Essencial Importante Desejável

RF_10: Indicar qual usuário está usando a sala - O sistema deve ser capaz de mostrar qual usuário está utilizando a sala na aba de detalhes.

Prioridade: Essencial Importante Desejável

RF_11: Indicar responsável pela sala no requerimento da reserva (Aluno) - Através do formulário de fazer o requerimento de uma reserva o sistema deve ser ter um campo para que o usuário do tipo aluno possa indicar um responsável pela sala.

Prioridade: Essencial Importante Desejável

RF_12: Envio de notificações de uma reserva (Professor ou Administrador) - O sistema deve ser capaz de enviar notificações para o email do usuário selecionado no requerimento de uma reserva feita por um usuário aluno

Prioridade: Essencial Importante Desejável

RF_13: Aprovar ou recusar um requerimento de reserva (Professor ou Administrador) - O sistema deve apresentar uma tela mostrando dados da reserva e do aluno que fez a reserva disponibilizando dois botões para o usuário aprovar ou recusar a reserva.

Prioridade: Essencial Importante Desejável

RF_14: Envio de notificações de reserva (Aluno) - O sistema deve ser capaz de enviar uma notificação por email para o usuário aluno indicando se a reserva feita pelo mesmo foi aprovada ou recusada.

Prioridade: Essencial Importante Desejável

RF_15: Atualização em tempo real do status da reserva - O sistema deve ser capaz de atualizar o status da reserva quando o horário da reserva for iniciado ou finalizado

Prioridade: Essencial Importante Desejável

RF_16: Dashboard do usuário - O sistema deve ter uma tela que liste todas as reservas feitas ou vinculadas ao usuário.

Prioridade: Essencial Importante Desejável

RF_17: Dashboard de histórico de uso - O sistema deve ter uma tela que liste os históricos de uso das salas, filtrando por tempo, mais usado, chaves não recolhidas.

Prioridade: Essencial Importante Desejável

2.3.2. REQUISITOS NÃO FUNCIONAIS

RNF_01: Disponibilidade - O sistema deve ficar disponível das 23:30 às 00:00 nos dias úteis e, aos sábados e domingos, 23:00 até as 00:30.

Prioridade: Essencial Importante Desejável

RNF_02: Segurança - O sistema deve fazer a verificação do email institucional que o usuário utilizar para realizar o cadastro e login.

Prioridade: Essencial Importante Desejável

RNF_03: Eficiência - O software deve ser eficiente, almejando consumir 270 MB de memória RAM e 0,4 GHz de processamento, almejando que tanto dispositivos com hardware inferior quanto superior possam rodar a aplicação.

Prioridade: Essencial Importante Desejável

RNF_04: Usabilidade - O usuário sem muitos conhecimentos, conseguirá ter uma experiência satisfatória ao utilizar o sistema com mínimo possível de erros cometidos pelo mesmo, portanto, todas as páginas devem ter atalhos associados para facilitar o uso.

Prioridade: Essencial Importante Desejável

RNF_05: Acessibilidade - O sistema deve possibilitar trocar para o alto-contraste, claro (light) e escuro (dark).

Prioridade: Essencial Importante Desejável

RNF_06: Integridade - O sistema deve garantir que os dados não sejam alterados ou corrompidos de maneira não autorizada.

Prioridade: Essencial Importante Desejável

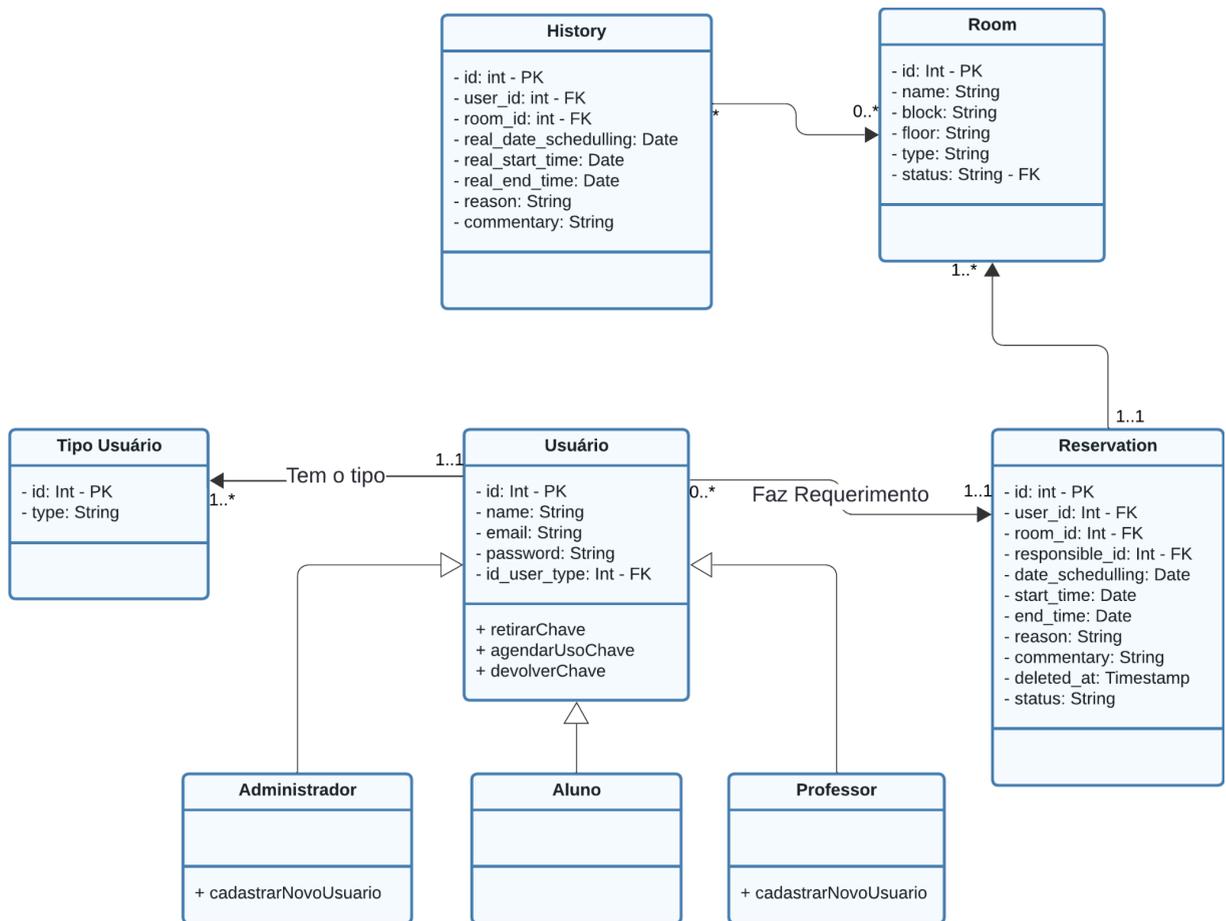
3. ANÁLISE E DESIGN

Nesta etapa do relatório será mostrado como foi construído o escopo do projeto anteriormente apresentado, através da criação de modelagens da arquitetura utilizando uma linguagem de modelagem visual UML (Unified Modelling Language - Linguagem de Modelagem Unificada) para gerar diagramas que servem como uma representação visual de um aspecto do sistema.

3.1. DIAGRAMA DE CLASSES

O Diagrama de Classes é uma representação visual estática que mapeia as estruturas de um sistema modelando suas classes, atributos, operações e relações entre os objetos.

Figura 1 - Diagrama de classes.

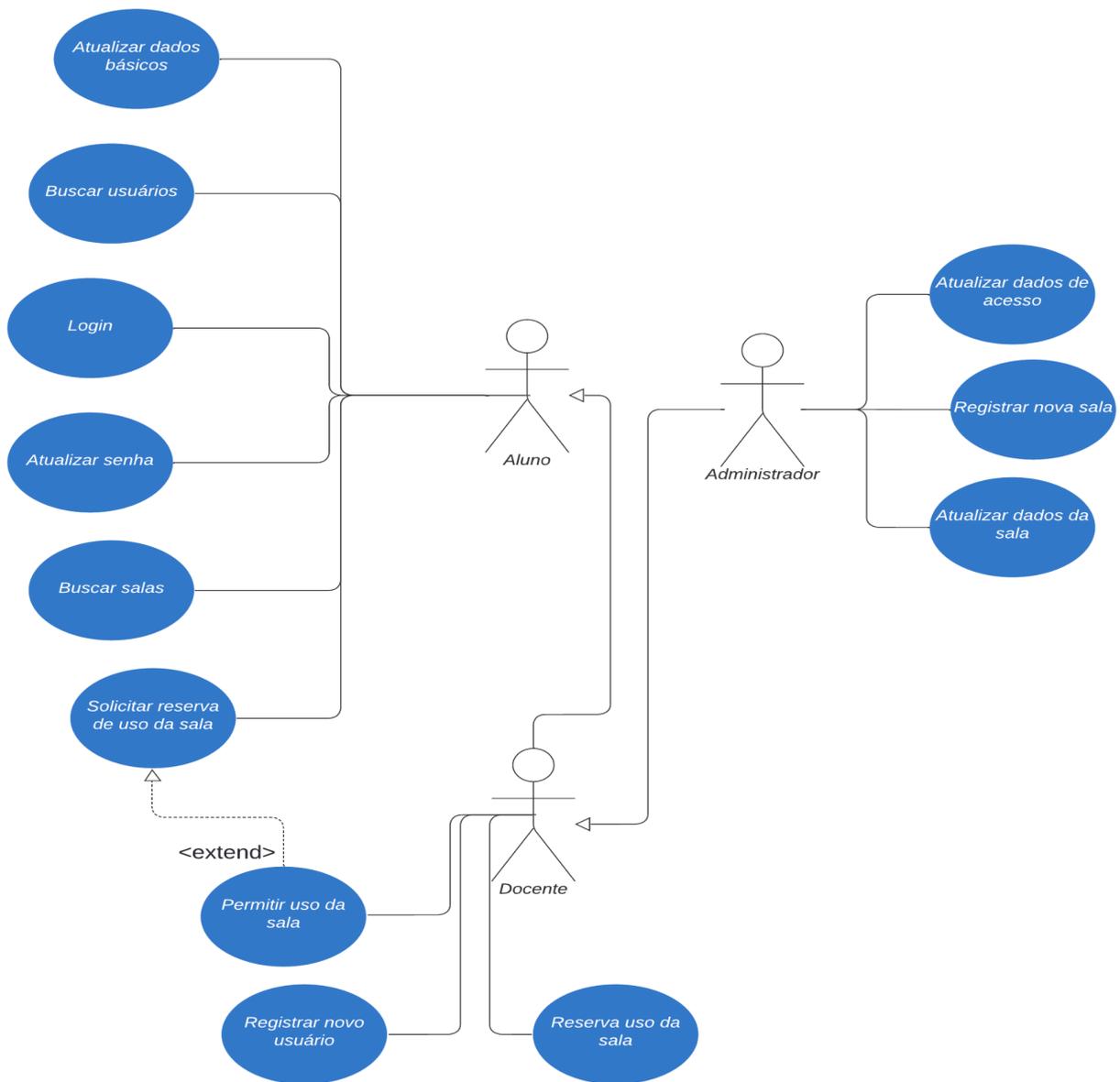


Fonte: Autoral.

3.2. DIAGRAMA DE CASOS DE USO

O Diagrama de Casos de Uso é uma representação visual que resume os detalhes e possíveis interações do usuário no sistema (atores).

Figura 2 - Diagrama de Casos de Uso.

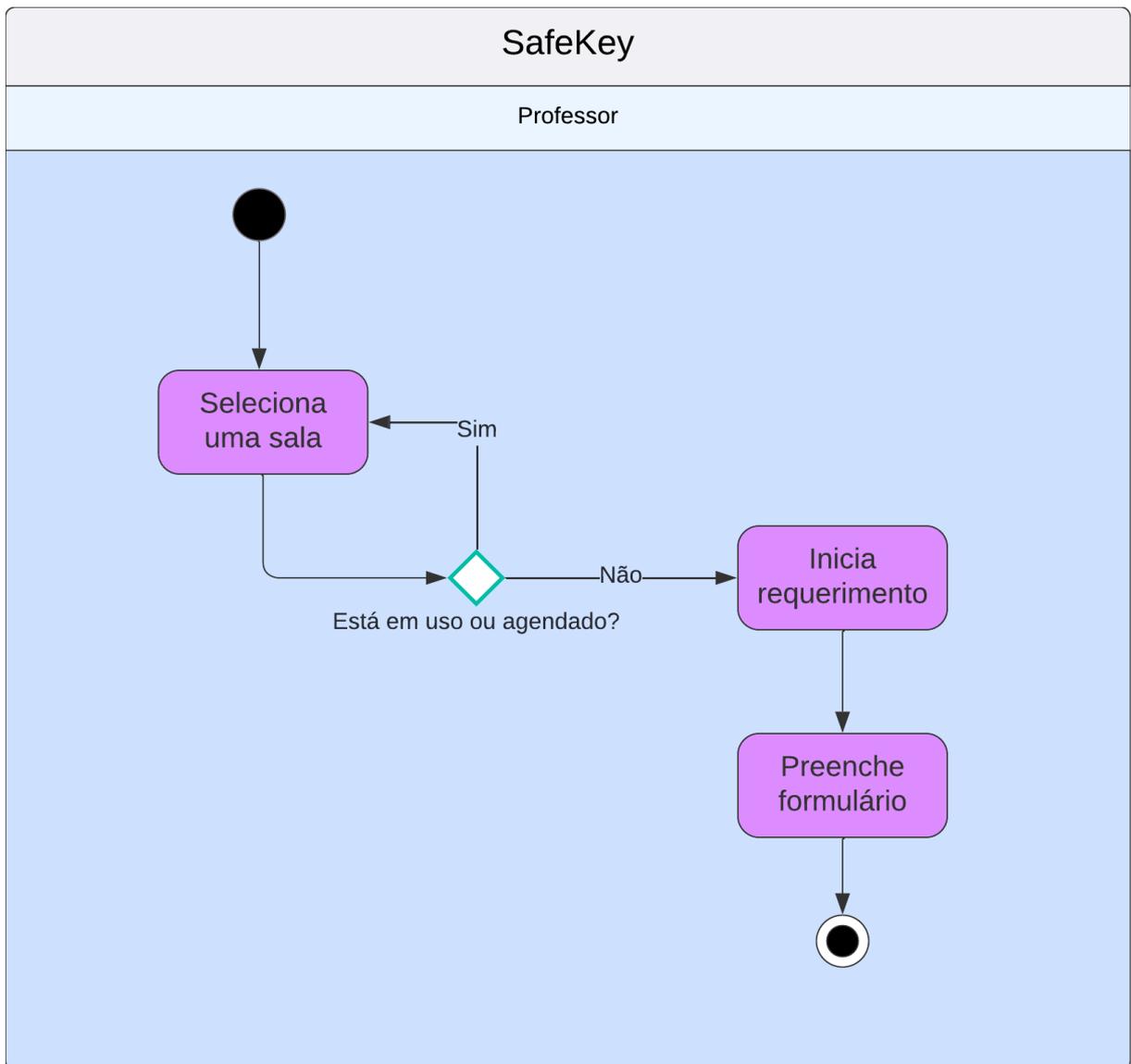


Fonte: Autoral.

3.3. DIAGRAMA DE ATIVIDADES

O Diagrama de Atividades é uma representação visual ou fluxograma que mostra as atividades executadas por um ou mais usuários no sistema.

Figura 3 - Diagrama de Atividades.

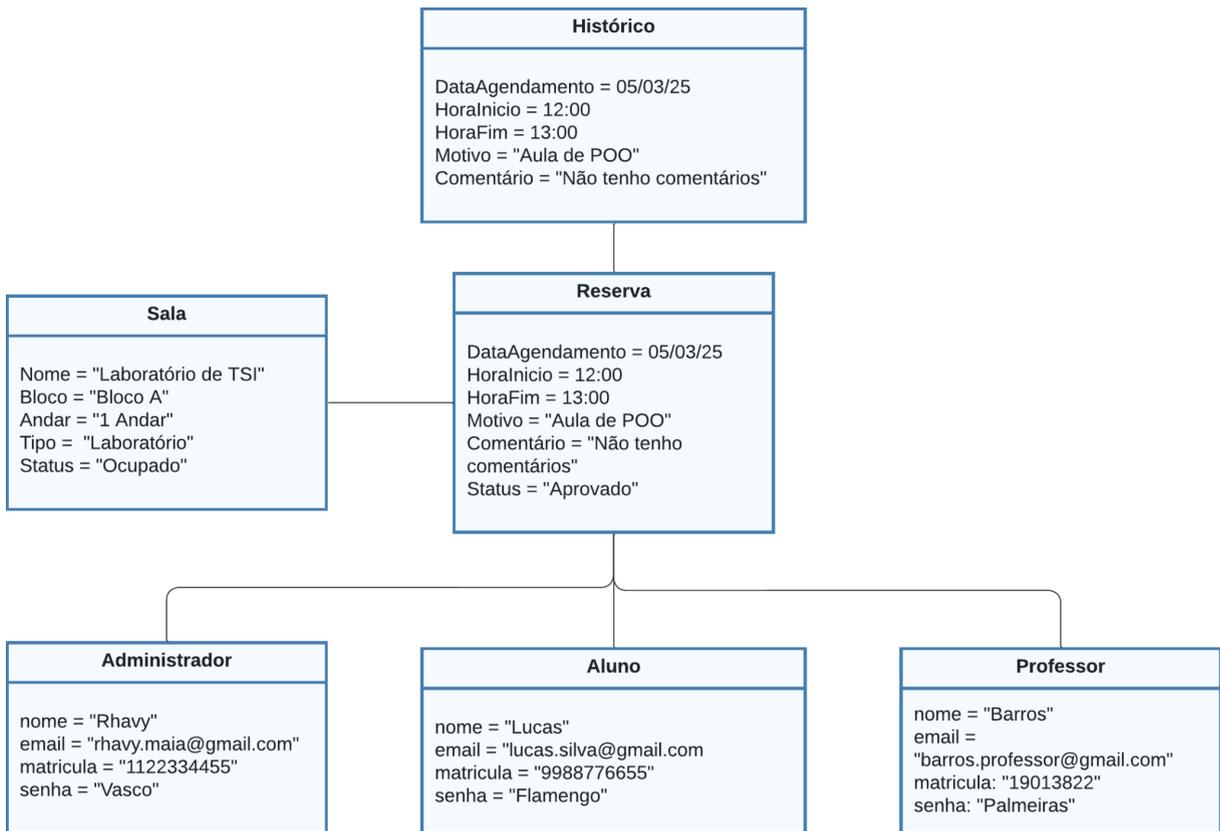


Fonte: Autoral.

3.4. DIAGRAMA DE OBJETOS

O Diagrama de Objetos oferece uma representação visual geral do sistema, mesmo que parecido com o diagrama de classes, este diagrama indica os atributos de um conjunto de objetos com valores de exemplo e como se relacionam no sistema.

Figura 4 - Diagrama de Objetos.

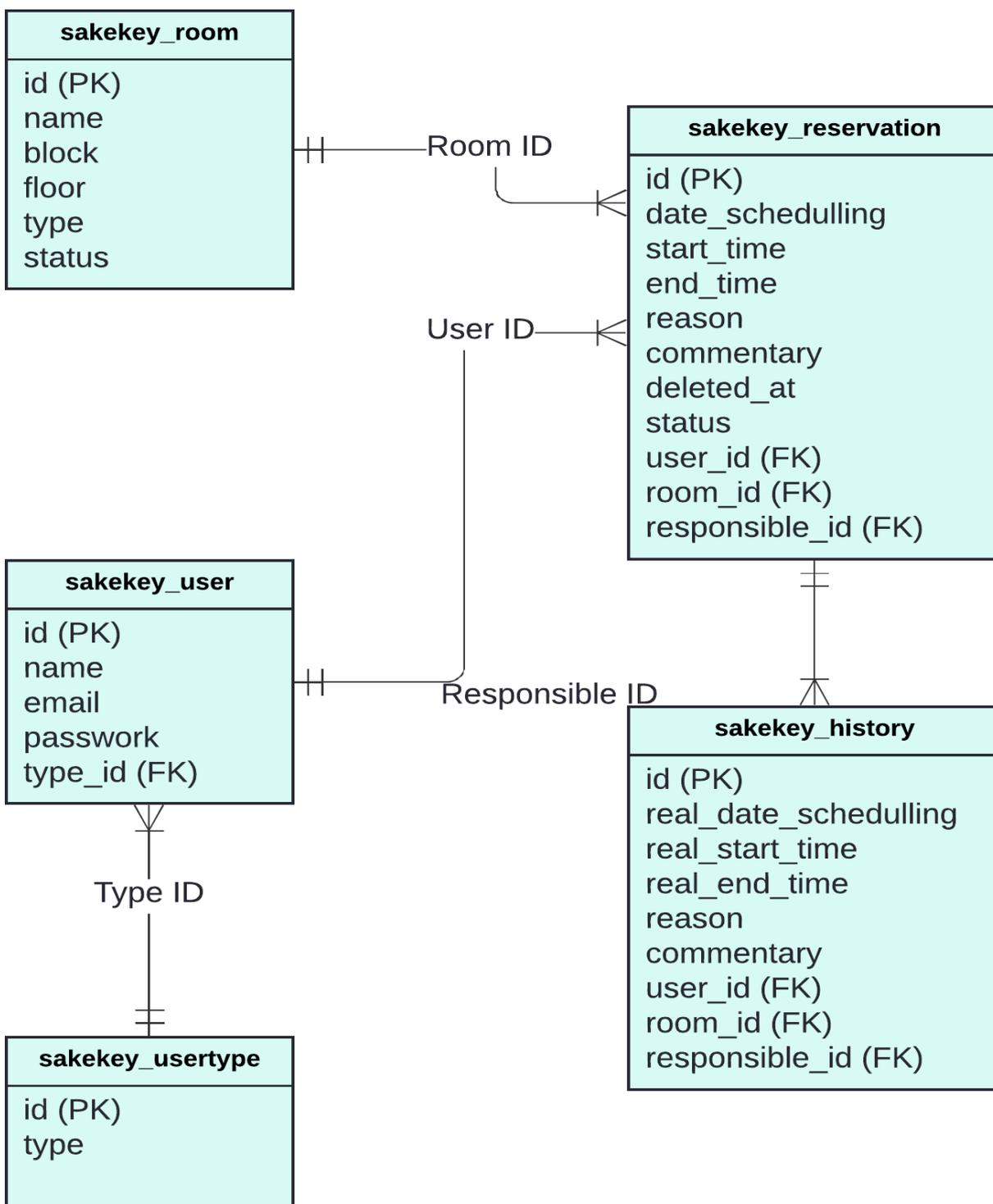


Fonte: Autoral.

3.5. MODELO DE DADOS

3.5.1. MODELO LÓGICO DA BASE DE DADOS

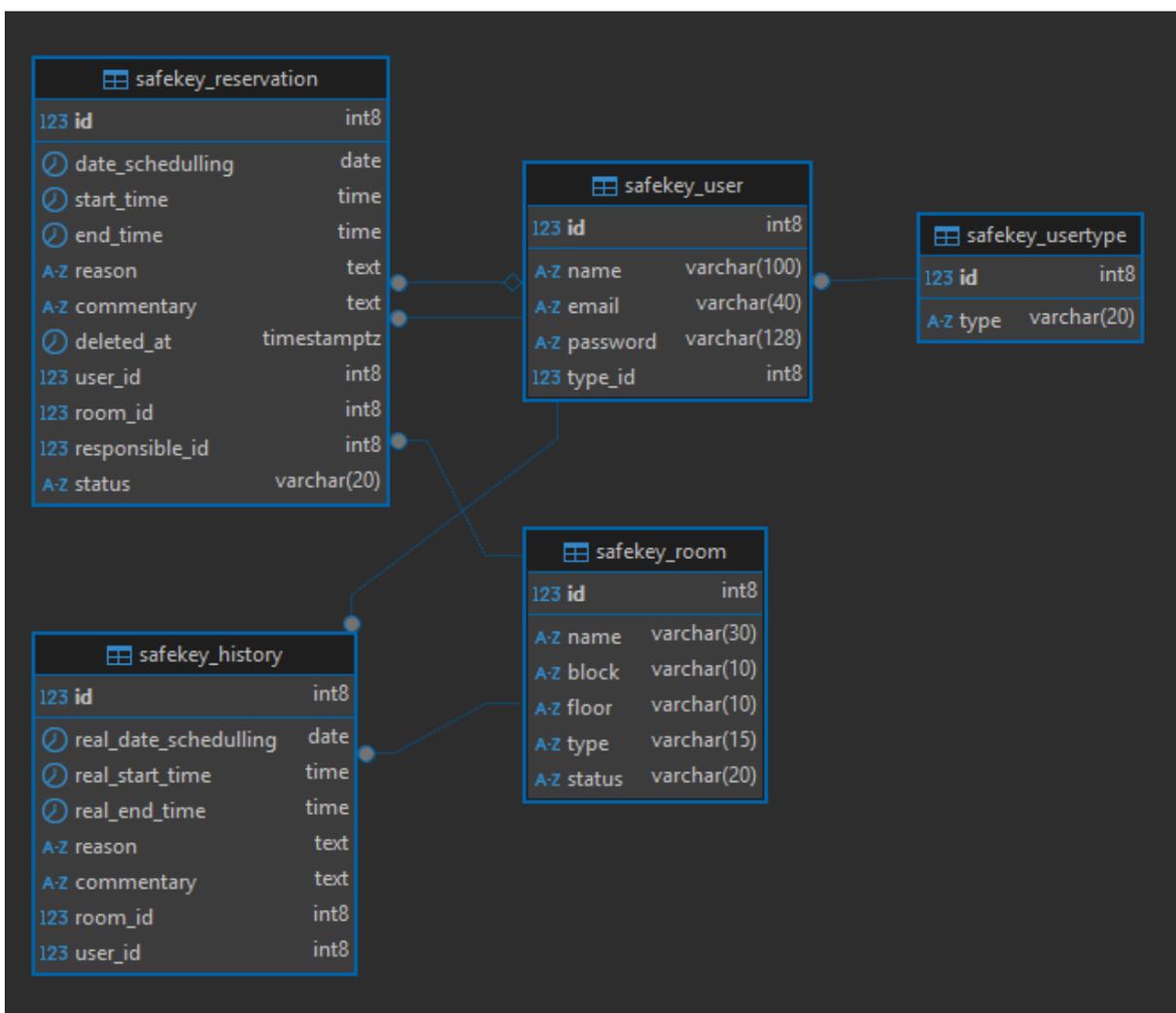
Figura 5 - Modelo Lógico da Base de Dados.



Fonte: Autoral.

3.5.2. DIAGRAMA DE RELACIONAMENTO DE ENTIDADES

Figura 6 - Diagrama de Relacionamento de Entidades.



Fonte: DBeaver.

3.5.3. DICIONÁRIO DE DADOS

User

Campo	Descrição	Tipo
id (PK)	Identificador único do usuário	Inteiro
name	Nome do usuário	Texto
email	Email do usuário	Texto
password	Senha do usuário	Texto
type_id (FK)	Identificador referenciado ao tipo do usuário	Inteiro

User_type

Campo	Descrição	Tipo
id (PK)	Identificador único do tipo de usuário	Inteiro
type	Tipo do usuário	Texto

Room

Campo	Descrição	Tipo
id (PK)	Identificador único da sala	Inteiro
name	Nome da sala	Texto
block	Bloco que a sala se encontra	Texto
floor	Andar que a sala se encontra	Texto
type	Tipo de sala	Texto
status	Status da sala	Texto

Reservation

Campo	Descrição	Tipo
id (PK)	Identificador único da reserva	Inteiro
date_schedulling	Data da reserva	Data
start_time	Horário de início da reserva	Hora
end_time	Horário de encerramento da reserva	Hora
reason	Motivo da reserva	Texto
commentary	Comentário adicional da reserva	Texto
deleted_at	Data e hora que a reserva foi excluída	Data e hora
status	Status da reserva	Texto
user_id (FK)	Identificador referenciado ao usuário	Inteiro
room_id (FK)	Identificador referenciado a sala	Inteiro
responsible_id (FK)	Identificador referenciado ao usuário responsável	Inteiro

History

Campo	Descrição	Tipo
id (PK)	Identificador único da reserva	Inteiro
real_date_schedulling	Data real da reserva	Data
real_start_time	Horário de início real da reserva	Hora
real_end_time	Horário de encerramento real da reserva	Hora
reason	Motivo da reserva	Texto
commentary	Comentário adicional da reserva	Texto
user_id (FK)	Identificador referenciado ao usuário	Inteiro
room_id (FK)	Identificador referenciado a sala	Inteiro

3.6. FERRAMENTAS DE DESENVOLVIMENTO

O Ambiente de desenvolvimento é um ambiente composto por ferramentas, tecnologias, recursos e metodologias que os desenvolvedores utilizam no seu dia a dia.

Para o desenvolvimento do sistema, foram utilizadas ferramentas que garantem escalabilidade, segurança e desempenho aprimorado. As principais tecnologias adotadas foram:

Backend:

IDE(Ambiente de Desenvolvimento Integrado): Visual Studio Code - Oferece uma interface leve e intuitiva, suporta várias linguagens, além de outras ferramentas e extensões que ajudam no desenvolvimento.

Linguagem de Programação: Python, uma linguagem amplamente utilizada no universo do desenvolvimento web graças à sua simplicidade e à sua facilidade na construção de APIs eficientes, resultando em uma maior produtividade.

Framework: Django, um framework web baseado em Python que oferece uma estrutura já pronta, simplificando o desenvolvimento da aplicação.

Banco de Dados: PostgreSQL, um sistema de gerenciamento de banco de dados relacionais, disponibiliza escalabilidade e segurança, garantindo a integridade dos dados armazenados no banco.

Autenticação: JWT (JSON Web Token), um padrão utilizado para garantir uma comunicação segura entre o frontend e o backend, compartilhando os dados no formato JSON.

Hospedagem e Deploy: Docker e Docker Compose, um software de código aberto usado para implantar sistemas dentro de containers virtuais, facilitando a criação e manutenção.

Gerenciamento de Versionamento: Git, um sistema de controle de versão amplamente utilizado para rastrear mudanças no código fonte durante o desenvolvimento do software, garantindo controle sobre o código-fonte e facilitando a colaboração.

Frontend:

IDE(Ambiente de Desenvolvimento Integrado): Visual Studio Code - Oferece uma interface leve e intuitiva, suporta várias linguagens, além de outras ferramentas e extensões que ajudam no desenvolvimento.

Linguagem de Programação: JavaScript - É uma linguagem de programação de alto nível, dinâmica, multiparadigma e orientada à objetos, amplamente utilizada para desenvolvimento web.

Linguagem de Estilo: CSS (Cascading Style Sheets) - É uma linguagem de estilo utilizada para definir a aparência de páginas web, manipulando cores, fontes, layouts e efeitos visuais

Biblioteca: ReactJs - É uma biblioteca de javascript que permite a construção e renderização de componentes reutilizáveis nas quais representam partes do código.

Framework: Tailwind CSS - É um framework CSS utilitário para construção de interfaces web fornecendo a criação de classes utilitárias de baixo nível que podem ser combinadas para criar estilos personalizados, flexíveis e utilitários

Plataformas externas: Shadcn/ui - É uma plataforma que fornece componentes UI projetados e estilizados com javascript e Tailwind CSS. Essa plataforma permite que seus componentes possam ser importados como um componente pré-definido oferecendo o código fonte completo do componente permitindo personalizações específicas.

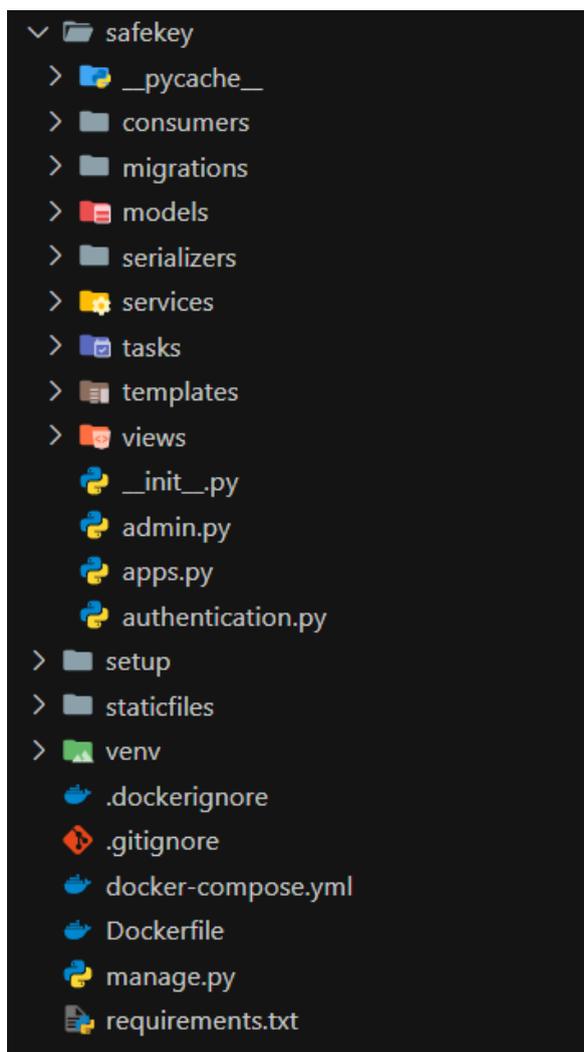
Gerenciamento de Versionamento: Git, um sistema de controle de versão amplamente utilizado para rastrear mudanças no código fonte durante o desenvolvimento do software, garantindo controle sobre o código-fonte e facilitando a colaboração.

4. IMPLEMENTAÇÃO

A implementação foi efetuada utilizando duas estruturas de projeto, sendo o backend, na qual foi concretizado com os padrões do framework Django para a linguagem de programação Python, e o frontend sendo implementado com a biblioteca ReactJs para a linguagem de programação Javascript.

4.1. Backend

Figura 7 - Estrutura de arquivos backend.



Fonte: Autoral.

O diretório do backend contém:

__pycache__/: Diretório criado automaticamente para armazenar arquivos compilados.

consumers/: Contém configurações relacionadas ao websocket.

migrations/: Contém os arquivos de migrações(cada alteração) do banco de dados.

models/: Contém os arquivos que definem os modelos do banco de dados.

serializers: Contém os serializers usados pelo Django REST Framework para converter instâncias dos modelos em dicionários para o python.

services: Contém a lógica utilizada para enviar os emails.

tasks: Contém a tarefa assíncrona utilizada com o celery.

templates: Contém estrutura html do conteúdo enviado no email.

views: Contém as views que processam as requisições HTTP e retornam suas respostas.

admin: Arquivo que contém a configuração do painel de administração do Django.

apps: Arquivo que contém a configuração da aplicação dentro do Django.

authentication: Arquivo que contém a lógica de autenticação do login.

setup: Contém as configurações do projeto do projeto.

staticfiles: Contém os arquivos estáticos do projeto.

venv: Ambiente virtual que isola as dependências do projeto.

.dockerignore: Arquivo que contém a lista de arquivos e diretórios que devem ser ignorados pelo Docker.

.gitignore: Arquivo que contém quais arquivos e diretórios que devem ser ignorados pelo Git.

docker-compose.yml: Arquivo que contém a configuração dos serviços como banco de dados e celery.

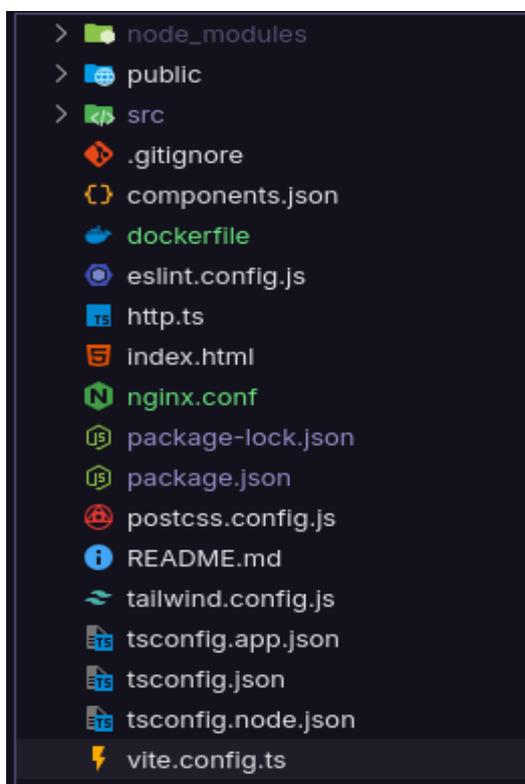
Dockerfile: Arquivo que contém a criação da imagem Docker do projeto.

manage: Arquivo principal do django, usado para rodar comandos.

requirements: Arquivo que contém a lista das dependências do projeto.

4.2. Frontend

Figura 8 - Estrutura de arquivos frontend.



Fonte: Autoral

O diretório do frontend contém:

node_modules/: Diretório que contém as dependências do projeto ReactJs.

public/: Pasta para armazenar arquivos incluídos em templates ou outras partes do aplicativo

src/: Contém o código-fonte da aplicação de componentes do React.

.gitignore: Arquivo que lista arquivos e diretórios a serem ignorados pelo Git.

.components.json: Arquivo que define os componentes e estilos padrões do shadcn/ui.

.dockerfile: Define a criação da imagem Docker do projeto.

.eslint.config.ts: Regras de linting do ESLint.

.http.ts: Arquivo que configura as requisições HTTP pelo axios.

.index.html: Página HTML principal onde irá ser mostrada toda a interface.

.nginx.conf: Contém a configuração do Nginx para servir a aplicação frontend.

.package-lock.json: Arquivo que faz o travamento de versões das dependências instaladas.

.package.json: Lista e versionaliza as dependências e script do projeto.

.postcss.config.js: Configuração para processamento de CSS com PostCSS.

.README.md: Arquivo de documentação e detalhes do projeto.

.tailwind.config.js: Arquivo de configuração do Tailwind CSS.

.tsconfig.app.json: Arquivo de configuração para o App do projeto.

.tsconfig.json: Arquivo de configuração do TypeScript.

.tsconfig.node.json: Arquivo de configuração para o Node.js.

.vite.config.ts: Arquivo de configuração do Vite, indicando os plugins do mesmo, configurações de build e rotas padrões de leitura de arquivos no projeto.

5. CONCLUSÃO

Ao final da implementação, a instituição terá um sistema web de gerenciamento de salas eficiente e de fácil utilização, que irá proporcionar um controle preciso sobre a disponibilidade de cada sala. O sistema permitirá gerenciar os usuários e salas registradas, realizar reservas, visualizar a disponibilidade das salas e de suas respectivas chaves em tempo real. Dessa forma, o sistema irá garantir uma gestão mais eficaz e melhor planejamento de atividades dentro da instituição.

6. REFERÊNCIAS

ALURA. Django: o que é e como funciona o framework? 2025. Disponível em: https://www.alura.com.br/artigos/django-framework?srsltid=AfmBOosp6WFDaej6dBDtjBk_0D1tZut1RJs3mRhXAGMFY0nUiyok3rT. Acesso em: 07 mar. 2025.

ALURA. O que é JSON Web Tokens (JWT)? 2025. Disponível em: https://www.alura.com.br/artigos/o-que-e-json-web-tokens?srsltid=AfmBOor90JvrhoWA8wV-k6i7EuImO4LfVvWnjbnSJhq7MxZS6BzY3_nW. Acesso em: 07 mar. 2025.

DBEAVER. DBEaver. 2025. Disponível em: <https://dbeaver.io>. Acesso em: 07 mar. 2025.

DJANGO REST FRAMEWORK. Django Rest Framework. 2025. Disponível em: <https://www.django-rest-framework.org>. Acesso em: 08 mar. 2025.

DOCKER. Get Started with Docker. 2025. Disponível em: <https://docs.docker.com/get-started/>. Acesso em: 07 mar. 2025.

GIT-SCM. Git. Disponível em: <https://git-scm.com/doc>. Acesso em: 7 mar. 2025.

LUCIDCHART. O que é um diagrama UML?. Disponível em: <https://www.lucidchart.com/pages/pt/o-que-e-diagrama-de-classe-uml>. Acesso em: 7 mar. 2025.

LUCIDCHART. O que é um diagrama de classe UML?. Disponível em: <https://www.lucidchart.com/pages/pt/o-que-e-diagrama-de-classe-uml>. Acesso em: 7 mar. 2025.

LUCIDCHART. Diagrama de caso de uso UML: O que é, como fazer e exemplos". Disponível em: <https://www.lucidchart.com/pages/pt/diagrama-de-caso-de-uso-uml>. Acesso em: 7 mar. 2025.

LUCIDCHART. O que é um diagrama de atividades UML?. Disponível em: <https://www.lucidchart.com/pages/pt/o-que-e-diagrama-de-atividades-uml>. Acesso em: 7 mar. 2025.

LUCIDCHART. O que é um diagrama de objetos UML?. Disponível em: <https://www.lucidchart.com/pages/pt/o-que-e-diagrama-de-objetos-uml>. Acesso em: 7 mar. 2025.

LUCID SOFTWARE INC. Lucidchart: Diagramação com inteligência. Disponível em: <https://www.lucidchart.com/pages/pt/landing>. Acesso em: 7 mar. 2025.

MICROSOFT. O que é PostgreSQL? 2025. Disponível em: <https://azure.microsoft.com/pt-br/resources/cloud-computing-dictionary/what-is-postgresql>.

Acesso em: 07 mar. 2025.

MICROSOFT. Visual Studio Code. Disponível em: <https://code.visualstudio.com/>. Acesso em: 7 mar. 2025.

MOZILLA DEVELOPER NETWORK. JavaScript. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>. Acesso em: 7 mar. 2025.

MOZILLA DEVELOPER NETWORK. CSS: Cascading Style Sheets. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/CSS>. Acesso em: 7 mar. 2025.

REACT. ReactJs. Disponível em: <https://react.dev/>. Acesso em: 7 mar. 2025.

SHADCN/UI. Construa sua biblioteca de componentes com shadui/cn. Disponível em: <https://ui.shadcn.com/>. Acesso em: 7 mar. 2025.

TAILWIND CSS. Framework CSS TailWindCSS. Disponível em: <https://tailwindcss.com/>. Acesso em: 7 mar. 2025.

VITE. Ferramenta de build rápida para projetos web modernos. Disponível em: <https://vite.dev/>. Acesso em: 7 mar. 2025.