

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

Antonio David de Souza
José Vinicius Tavares Aprigio

RELATÓRIO TÉCNICO
Projeto Integrador em Sistemas para Internet (PISI): KeyControl (Sistema de
Gerenciamento de Salas)

GUARABIRA – PB
2026

Antonio David de Souza
José Vinícius Tavares Aprigio

RELATÓRIO TÉCNICO
Projeto Integrador em Sistemas para Internet (PISI): KeyControl (Sistema de Gerenciamento de Salas)

Trabalho de Conclusão de Curso apresentado ao Curso de Tecnologia em 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(a): Rodrigo Leone Alves, Dr.
Coorientador(a): Rhavy Maia Guedes, Me.

GUARABIRA – PB
2026

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA DO IFPB - GUARABIRA

S729r

Souza, Antonio David de

Relatório técnico: Projeto Integrador em Sistemas para Internet (PISI): KeyControl (Sistema de Gerenciamento de Salas) / Antonio David de Souza; José Vinícius Tavares Aprigio.- Guarabira, 2026.
49f.; il.; color.

Trabalho de Conclusão de Curso (Tecnólogo em Sistemas para Internet). – Instituto Federal da Paraíba, Campus Guarabira. 2026.

“Orientação: Prof. Dr. Rodrigo Leone Alves
Coorientação: Prof. Me. Rhavy Maia Guedes”

Referências.

1.Sistema para internet 2. Gerenciamento de sala. 3. Sistema web. 4. Desenvolvimento ágil. 5.KeyControl. I. Título.

CDU 004.4(0.067)

Elaborada por Ana Carine da Costa Gonçalves - CRB 000676



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

ATA 19/2026 - 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 06 de agosto de 2026, às 10:00, no Laboratório de Informática, reuniram-se os membros da banca avaliadora, Rodrigo Leone Alves (Orientador), Rhavy Maia Guedes (Coorientador), Tannissa Luanna Cardoso de Araujo (Examinador Interno), Daniel Marques Targino Guimarães (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) **Antonio David de Souza, matrícula de nº 202323810011** intitulado "KeyControl (Sistema de Gerenciamento de Salas)", 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 92 (Noventa e Dois)**.

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 10 de agosto de 2026.

Documento assinado eletronicamente por:

- **Rodrigo Leone Alves, PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 10/08/2026 12:12:02.
- **Tannissa Luanna Cardoso de Araujo, PEDAGOGO-AREA**, em 10/08/2026 12:48:54.
- **Rhavy Maia Guedes, PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 10/08/2026 15:40:28.
- **Daniel Marques Targino Guimaraes, PROF ENS BAS TEC TECNOLOGICO-SUBSTITUTO**, em 10/08/2026 22:41:34.

Este documento foi emitido pelo SUAP em 10/08/2026. 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 919910
Verificador: b41f2d81df
Código de Autenticação:





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

ATA 20/2026 - 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 06 de agosto de 2026, às 10:00, no Laboratório de Informática, reuniram-se os membros da banca avaliadora, Rodrigo Leone Alves (Orientador), Rhavy Maia Guedes (Coorientador), Tannissa Luanna Cardoso de Araujo (Examinador Interno), Daniel Marques Targino Guimarães (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) **José Vinicius Tavares Aprigio, matrícula de nº 202323810003** intitulado "KeyControl (Sistema de Gerenciamento de Salas)", 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 92 (Noventa e Dois)**.

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 10 de agosto de 2026.

Documento assinado eletronicamente por:

- **Rodrigo Leone Alves, PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 10/08/2026 12:14:12.
- **Tannissa Luanna Cardoso de Araujo, PEDAGOGO-AREA**, em 10/08/2026 12:49:05.
- **Daniel Marques Targino Guimaraes, PROF ENS BAS TEC TECNOLOGICO-SUBSTITUTO**, em 10/08/2026 22:40:15.

Este documento foi emitido pelo SUAP em 10/08/2026. 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 919913
Verificador: 548e988e36
Código de Autenticação:



Por Antonio David de Souza

A Deus, À minha família e para todos que me desejaram o bem, o apoio de vocês foi indispensável durante minha caminhada!

Dedico!

Por José Vinícius Tavares Aprigio

A Deus, à minha família e a toda a comunidade escolar, pelo apoio, incentivo e ensinamentos ao longo desta caminhada.

Dedico!

AGRADECIMENTOS

Por Antonio David de Souza

Agradeço, primeiramente, a Deus, por me conceder saúde e força para perseguir meus objetivos.

À minha família, por acreditar em meu potencial.

À minha amada Bárbara Silva Francisco, cujo apoio e companhia foram indispensáveis ao longo da jornada.

Aos meus amigos, que sempre me encorajaram.

Aos meus professores, que contribuíram para a minha formação e busca pelo conhecimento.

E a mim mesmo, por não ter desistido.

Por José Vinícius Tavares Aprigio

Primeiramente, agradeço a Deus.

À minha família, pelo apoio, incentivo e contribuição ao longo de toda essa caminhada.

Aos meus amigos, pela parceria e auxílio durante o desenvolvimento deste trabalho.

Aos meus professores, pelos ensinamentos transmitidos, pela orientação e pela contribuição para minha formação acadêmica.

*“Se você vai tentar, vá até o fim.
Caso contrário, nem comece”.*

Charles Bukowski

RESUMO

O Projeto Integrador em Sistemas para Internet tem como objetivo desenvolver a capacidade de aplicação integrada dos conceitos e teorias adquiridos durante o curso. Nesse contexto, o presente relatório apresenta a documentação do sistema KeyControl, um software desenvolvido para auxiliar no gerenciamento de salas em ambientes educacionais, promovendo maior organização, controle e eficiência na utilização desses espaços. A metodologia adotada no desenvolvimento se baseia em uma pesquisa aplicada, utilizando práticas de desenvolvimento ágil, com levantamento de requisitos funcionais e não funcionais e implementação incremental da solução. Para o desenvolvimento do sistema, foram utilizadas as tecnologias Python, o framework Flask para o backend, React no frontend, PostgreSQL para o gerenciamento do banco de dados. O sistema oferece um conjunto de funcionalidades voltadas ao controle e gerenciamento de salas, visando facilitar a administração desses ambientes que optarem pela utilização da plataforma.

Palavras-chave: gerenciamento de sala; sistemas web; desenvolvimento ágil.

ABSTRACT

The Internet Systems Integrator Project aims to develop the ability to integrate the concepts and theories acquired during the course. In this context, this report presents the documentation of the KeyControl system, a software developed to assist in the management of rooms in educational environments, promoting greater organization, control and efficiency in the use of these spaces. The methodology adopted in development is based on applied research, using agile development practices, with functional and non-functional requirements gathering and incremental implementation of the solution. To develop the system, Python technologies were used, the Flask framework for the backend, React on the frontend, PostgreSQL for database management. The system offers a set of functionalities aimed at controlling and managing rooms, aiming to facilitate the administration of those environments that choose to use the platform.

Keywords: Room Management; Web Systems; Agile Development.

LISTA DE ILUSTRAÇÕES

Figura 1 - Diagrama de Casos de Uso do Sistema KeyControl.....	30
Figura 2 - Arquitetura do Sistema.....	36
Figura 3 - Modelo lógico da base de dados.....	37
Figura 5 - Tela de cadastro de usuários.....	38
Figura 6 - Tela de perfil.....	39
Figura 7 - Tela de edição de perfil.....	40
Figura 8 - Tela de início.....	40
Figura 9 - Tela de reservas (perfil Responsável).....	41
Figura 10 - Tela de cadastro de nova reserva (perfil Responsável).....	41
Figura 11 - Tela de retiradas (perfil Responsável).....	42
Figura 12 - Tela de cadastro de nova retirada (perfil Responsável).....	42
Figura 13 - Tela de histórico (perfil Responsável).....	43
Figura 14 - Tela de responsáveis (perfil Administrador).....	43
Figura 15 - Tela de salas (perfil Administrador).....	44
Figura 16 - Tela de cadastro de nova sala (perfil Administrador).....	44
Figura 17 - Tela de chaves (perfil Administrador).....	45
Figura 18 - Tela de criação de nova chave (perfil Administrador).....	45
Figura 19 - Tela de reservas (perfil Administrador).....	46
Figura 20 - Tela de cadastro de nova reserva (perfil Administrador).....	47
Figura 21 - Tela de retiradas (perfil Administrador).....	47
Figura 22 - Tela de cadastro de nova retirada (perfil Administrador).....	48

LISTA DE TABELAS

Tabela 1 - Cronograma de atividades do TCC.....	20
Tabela 2 - Tabela de Requisitos Funcionais.....	21
Tabela 3 - Tabela de requisitos não funcionais.....	27
Tabela 4 - Tabela de user stories relacionado aos perfis de responsáveis e administradores....	29
Tabela 5 - Tabela de cenários de carga.....	49

LISTA DE ABREVIATURA E SIGLAS

API	Application Programming Interface (Interface de Programação de Aplicações)
CI	Continuous Integration (Integração Contínua)
CPF	Cadastro de Pessoa Física
HTTP	Hypertext Transfer Protocol (Protocolo de Transferência de Hipertexto)
IFPB	Instituto Federal da Paraíba
JWT	JSON Web Token (Token Web JSON)
ORM	Object-Relational Mapping (Mapeamento Objeto-Relacional)
REST	Representational State Transfer (Transferência de Estado Representacional)
SGBD	Sistema Gerenciador de Banco de Dados
SIAPE	Sistema Integrado de Administração de Recursos Humanos
SQL	Structured Query Language (Linguagem de Consulta Estruturada)
UML	Unified Modeling Language (Linguagem de Modelagem Unificada)

SUMÁRIO

1 INTRODUÇÃO.....	14
1.1 Justificativa.....	14
1.2 Objetivos.....	15
1.2.1 Objetivo Geral.....	15
1.2.2 Objetivos Específicos.....	15
2 REFERENCIAL TEÓRICO.....	17
2.1 Gestão de espaços físicos em instituições de ensino.....	17
2.2 Sistemas de Informação na gestão administrativa.....	18
3 METODOLOGIA.....	20
3.1 Abordagem Metodológica.....	20
3.2 Cronograma.....	20
3.3 Procedimentos Técnicos.....	20
3.4 Requisitos.....	21
3.4.1 Requisitos Funcionais.....	21
3.4.2 Requisitos Não Funcionais.....	27
3.4.3 User Stories.....	29
3.4.4 Diagrama de Casos de Usos.....	30
3.5 Ferramentas e Tecnologias.....	31
3.5.1 Linguagem de programação.....	31
3.5.2 Backend.....	32
3.5.3 Frontend.....	32
3.5.4 Banco de Dados.....	33
3.5.5 Controle de Versão.....	33
3.5.6 Ambiente de desenvolvimento.....	34
4 RESULTADOS E DISCUSSÃO.....	35
4.1 Arquitetura do Sistema.....	35
4.2 Modelagem do Sistema.....	36
4.3 Telas do Sistema.....	37
4.4 Testes de Carga.....	48
4.4.1 Metodologia dos testes.....	48
4.4.2 Análise dos resultados.....	49
4.4.3 Conclusão dos testes.....	49
5 CONCLUSÃO.....	50
REFERÊNCIAS.....	51

1 INTRODUÇÃO

As instituições de ensino dependem de uma boa organização dos seus espaços físicos para garantir o funcionamento adequado das atividades acadêmicas. Nesse contexto, o gerenciamento de salas e o controle das chaves tornam-se processos essenciais para assegurar a utilização correta desses recursos, contribuindo para a organização administrativa, a segurança patrimonial e a disponibilidade dos ambientes.

Segundo Maximiano (2000), a administração envolve funções como planejamento, organização, direção e controle, sendo esta última responsável por acompanhar a execução das atividades e verificar se os resultados obtidos seguem o planejado. Dessa forma, o controle dos recursos físicos constitui uma atividade administrativa importante para garantir a utilização adequada dos espaços institucionais.

Entretanto, em muitas instituições, essas atividades ainda são realizadas manualmente ou por meio de ferramentas pouco integradas, o que pode ocasionar problemas como conflitos de reservas, perda de informações e falhas no controle de acesso aos ambientes. Além disso, a ausência de um sistema centralizado dificulta o acompanhamento das retiradas e devoluções de chaves, comprometendo a segurança e a organização institucional. A falta de registros confiáveis também dificulta a identificação de responsáveis pelo uso das salas, aumentando os riscos de extravios e utilização inadequada dos espaços.

Com base nisso, a utilização de sistemas de informação apresenta-se como uma alternativa para modernizar os processos administrativos relacionados à gestão de recursos físicos, proporcionando maior controle, rastreabilidade e confiabilidade das informações. Conforme destaca Maximiano (2000), mecanismos que produzem informações sobre o andamento das atividades são fundamentais para a função de controle, permitindo o acompanhamento de sua execução e contribuindo para a tomada de decisões.

Diante desse cenário, o sistema KeyControl foi desenvolvido com o objetivo de auxiliar instituições de ensino no gerenciamento de salas e no controle de chaves. A solução proposta permite o cadastro de usuários, a realização de reservas de salas e o registro das retiradas e devoluções de chaves, mantendo um histórico detalhado das operações realizadas e contribuindo para a organização, a segurança e a eficiência da gestão dos espaços físicos institucionais.

1.1 Justificativa

A necessidade de modernização dos processos administrativos nas instituições de ensino torna cada vez mais importante a utilização de sistemas que auxiliem na organização e

no controle de recursos físicos. O gerenciamento manual ou a utilização de sistemas excessivamente burocráticos para reservar salas e controlar chaves pode gerar problemas como conflitos de agendamento, demora nos registros, baixa adesão dos usuários ao sistema.

O desenvolvimento do sistema KeyControl surgiu a partir das dificuldades observadas no software utilizado pela própria instituição de ensino, IFPB Campus Guarabira, no qual foram identificadas oportunidades de melhoria relacionadas à simplificação do fluxo de registros e à otimização da experiência dos usuários. Esses problemas impactam diretamente a eficiência do controle das salas, aumentando o tempo necessário para cada operação e dificultando a rotina administrativa da instituição. Dessa forma, identificou-se a necessidade de uma solução mais simples, intuitiva e eficiente, capaz de reduzir etapas desnecessárias e otimizar o gerenciamento dos espaços físicos.

Nesse contexto, o desenvolvimento do sistema KeyControl justifica-se pela proposta de automatizar e centralizar o gerenciamento de salas e chaves, proporcionando maior controle, rastreabilidade e organização dos processos, além de melhorar a experiência dos usuários responsáveis pela utilização do sistema. Além disso, a utilização de uma solução web permite maior acessibilidade e praticidade no gerenciamento das informações, contribuindo para a modernização dos processos institucionais e para a melhoria da gestão das salas.

1.2 Objetivos

1.2.1 Objetivo Geral

Desenvolver um sistema web para gerenciamento de salas e controle de chaves em instituições de ensino, visando melhorar a organização, segurança e rastreabilidade na utilização dos espaços físicos.

1.2.2 Objetivos Específicos

- Implementar um módulo de cadastro de salas e chaves, permitindo o gerenciamento estruturado dessas informações.
- Desenvolver um módulo de reservas de salas permitindo o gerenciamento da utilização dos espaços por usuários autorizados.
- Implementar o controle de retirada e devolução de chaves, registrando data, horário e responsável pela operação.
- Criar um histórico de retiradas, possibilitando consultas e auditorias das ações realizadas.

- Desenvolver interfaces intuitivas, para rápida e fácil utilização, facilitando o uso do sistema pelos usuários da instituição.
- Centralizar as informações relacionadas à gestão de salas e chaves, reduzindo falhas manuais e melhorando a organização institucional.

2 REFERENCIAL TEÓRICO

2.1 Gestão de espaços físicos em instituições de ensino

A gestão de espaços e recursos físicos nas instituições possui uma relação intrínseca com as funções administrativas, uma vez que o planejamento, a organização e o controle são essenciais para que as organizações utilizem seus recursos de forma eficiente. Segundo Chiavenato (2003), o planejamento consiste em definir os objetivos e os caminhos necessários para atingi-los, enquanto a organização está relacionada à estruturação e à distribuição dos recursos, bem como à definição das responsabilidades relacionadas à sua administração, de modo a adequá-los às necessidades da organização. Nesse contexto, o controle de chaves pode ser compreendido como um procedimento administrativo que estabelece critérios para o acesso aos ambientes institucionais, atribui responsabilidades aos usuários e contribui para a proteção do patrimônio.

Outra função administrativa essencial é o controle, cuja finalidade é assegurar que o acesso aos espaços físicos ocorra conforme o planejamento estabelecido, prevenindo desvios, usos indevidos e perdas patrimoniais. Maximiano (2000, p. 27) define o controle como “o processo de assegurar a realização dos objetivos e de identificar a necessidade de modificá-los”. Dessa forma, o controle de chaves pode ser compreendido como um mecanismo administrativo de acompanhamento e verificação do uso dos espaços físicos, contribuindo para a preservação dos recursos da instituição.

Por fim, a eficiência administrativa está relacionada à padronização dos processos e à utilização adequada dos recursos. Para Chiavenato (2003, p. 63), “A padronização passa a ser vital[...] na melhoria da eficiência. A padronização conduz à simplicidade na medida em que a uniformidade reduz a variabilidade e as exceções que complicam o processo”. Complementando essa perspectiva, Maximiano (2000, p. 115) relaciona a eficiência à realização das atividades com o mínimo de esforço e o melhor aproveitamento possível dos recursos disponíveis. Na gestão de chaves, a adoção de normas para retirada, devolução e registro promove maior controle, transparência e rastreabilidade das operações, reduzindo falhas decorrentes de procedimentos informais e fortalecendo a gestão dos recursos físicos da instituição.

2.2 Sistemas de Informação na gestão administrativa

Os sistemas de informação possuem características importantes para a gestão administrativa, pois permitem organizar, processar e disponibilizar informações necessárias à gestão dos recursos organizacionais. Segundo Laudon e Laudon (2014, p. 13):

Um **sistema de informação (SI)** pode ser definido tecnicamente como um conjunto de componentes inter-relacionados que coletam (ou recuperam), processam, armazenam e distribuem informações destinadas a apoiar a tomada de decisões, a coordenação e o controle em uma organização.

Com isso, entende-se que um sistema de informação não se limita ao armazenamento de dados, mas envolve um processo de transformação dos dados em informações, de forma a alcançar uma melhor organização, controle dos recursos e processos administrativos.

Nesse contexto, é essencial entender a diferença entre dados e informações, uma vez que os dados representam os elementos que serão coletados e processados pelo sistema para posteriormente gerar informações úteis à organização.

Dados são compostos por fatos básicos, como o nome e a quantidade de horas trabalhadas em uma semana de um funcionário, números de peças em estoque ou pedidos. [...] Quando esses fatos são organizados ou arranjados de maneira significativa, eles se transformam em informações” (STAIR; REYNOLDS, 2006, p. 4)

Assim, dentro de um sistema de gestão, registros como identificação de usuários, salas e horários de movimentação de chaves representam dados que, após serem tratados, podem ser transformados em informações relevantes para a gestão da organização.

Os sistemas de informação também podem contribuir para a melhoria dos processos administrativos ao favorecer a organização e o gerenciamento das informações utilizadas pelas organizações. Segundo Stair e Reynolds (2006), os sistemas de informação modificam a forma como as organizações conduzem seus negócios, sendo utilizados para criar, armazenar e transferir informações. A partir desses recursos, é possível estruturar e organizar as informações, facilitando seu acompanhamento e consulta. Nesse sentido, o KeyControl aplica esses princípios ao gerenciamento de chaves institucionais, permitindo o cadastro de salas, usuários responsáveis, registros de retirada e devolução, reservas e histórico de movimentações.

Dessa maneira, o KeyControl deixa de ser apenas um mecanismo de armazenamento de registros e passa a atuar como uma ferramenta de apoio à gestão institucional. A partir das informações criadas pelo sistema, a instituição pode acompanhar movimentações, identificar ocorrências, aumentar a segurança do ambiente e tomar decisões coerentes com os dados estabelecidos. A rastreabilidade proporcionada pelo sistema contribui para substituir controles defasados, reduzir falhas operacionais e aumentar a eficiência da administração dos recursos físicos.

3 METODOLOGIA

3.1 Abordagem Metodológica

Em relação à abordagem metodológica, este trabalho caracteriza-se como uma pesquisa aplicada, pois visa ao desenvolvimento de uma solução tecnológica para auxiliar instituições de ensino no gerenciamento de salas e no controle de chaves.

3.2 Cronograma

Tabela 1 - Cronograma de atividades do TCC

Atividade	jun./25	jul./25	ago./25	set./25	out./25	dez./25	jan./26	fev./26	mar./26	abr./26	mai./26	jun./26
Pesquisa e Concepção	X											
Definição da Solução		X										
Fundamentação Teórica			X									
Análise de Requisitos				X								
Projeto de Software					X							
Desenvolvimento				X	X	X	X	X	X	X	X	
Testes											X	
Escrita do Documento					X	X	X	X	X	X	X	X

Fonte: elaborado pelos autores.

3.3 Procedimentos Técnicos

O desenvolvimento do sistema KeyControl foi conduzido com base na metodologia ágil Scrum, framework amplamente utilizado no desenvolvimento de software por possibilitar a organização do trabalho em ciclos curtos, favorecendo a adaptação às mudanças e a entrega contínua de funcionalidades (SCHWABER; SUTHERLAND, 2020).

Para a execução do projeto, foram adotadas sprints com duração de sete dias. No início de cada sprint, era realizado o planejamento das atividades a serem desenvolvidas, priorizando as funcionalidades previstas para o sistema, como o gerenciamento de salas, chaves, responsáveis, reservas e retiradas. Ao término de cada ciclo, as funcionalidades implementadas eram revisadas e testadas, permitindo identificar possíveis melhorias e validar se os requisitos definidos estavam sendo atendidos.

Essa abordagem possibilitou o desenvolvimento incremental do sistema, facilitando a correção de problemas ao longo do projeto e garantindo a evolução contínua da solução até sua versão final.

As etapas do desenvolvimento foram divididas da seguinte forma:

- **Levantamento de Requisitos:** identificação das necessidades do sistema a partir da análise dos processos de gerenciamento de salas e controle de chaves da instituição de ensino, considerando atividades como o cadastro de salas, chaves e responsáveis, a realização de reservas, o registro de retiradas e devoluções de chaves e a manutenção do histórico das operações.
- **Análise:** definição das regras de negócio, dos perfis de usuários e das funcionalidades do sistema. Nessa etapa, foram estabelecidas as permissões dos usuários, os fluxos de reservas, retiradas e devoluções de chaves, bem como a priorização das funcionalidades a serem implementadas.
- **Projeto:** definição da arquitetura do sistema e elaboração da modelagem da solução, incluindo diagramas UML, protótipos das interfaces e o modelo lógico do banco de dados, que serviram como base para a implementação do sistema.
- **Implementação:** desenvolvimento incremental do sistema utilizando sprints semanais, com a implementação, testes e validação contínua das funcionalidades previstas, como o gerenciamento de salas, chaves, responsáveis, reservas, retiradas e histórico de operações.

3.4 Requisitos

Esta seção é dedicada à apresentação dos requisitos que foram ponto de partida para o desenvolvimento da aplicação.

3.4.1 Requisitos Funcionais

Os requisitos funcionais descrevem os serviços que o sistema oferece ao usuário (SOMMERVILLE, 2011), desta forma, apresentamos os requisitos funcionais que foram implementados no sistema:

Tabela 2 - Tabela de Requisitos Funcionais

Requisitos	Descrição do Requisito
[RF001] Cadastro de Sala	O sistema deve permitir que o responsável com a função de administrador cadastre novas salas referentes ao espaço físico da instituição, fornecendo as informações do nome da sala e se ela está disponível para sua utilização. Sendo esses campos

	obrigatórios. Prioridade: ☒ Essencial ☐ Importante ☐ Desejável
[RF002] Listagem de Salas	O sistema deve permitir que o responsável com a função de administrador visualize as salas cadastradas, exibindo seus nomes e a sua disponibilidade. Prioridade: ☒ Essencial ☐ Importante ☐ Desejável
[RF003] Edição de Sala	O sistema deve permitir que o responsável com a função de administrador edite os dados de uma sala já cadastrada, possibilitando a atualização do nome e alterando a disponibilidade da sala, indicando que ela já não pode mais ser usada. Prioridade: ☒ Essencial ☐ Importante ☐ Desejável
[RF004] Exclusão de Sala	O sistema deve permitir que o responsável com a função de administrador exclua uma sala cadastrada. Prioridade: ☒ Essencial ☐ Importante ☐ Desejável
[RF005] Cadastro de Chave	O sistema deve permitir que o responsável com a função de administrador cadastre novas chaves, associando-as a salas previamente cadastradas e informando sua disponibilidade. O sistema deverá permitir o cadastro de múltiplas chaves para uma mesma sala. Além disso, ao cadastrar uma nova sala, uma chave deverá ser criada automaticamente e vinculada a ela. Prioridade: ☒ Essencial ☐ Importante ☐ Desejável
[RF006] Listagem de Chaves	O sistema deve permitir que o responsável com a função de administrador visualize as chaves cadastradas, apresentando o nome da chave, a sala à qual pertence e sua

	disponibilidade. Prioridade: ☒ Essencial ☐ Importante ☐ Desejável
[RF007] Edição de Chave	O sistema deve permitir que o responsável com a função de administrador edite os dados de uma chave já cadastrada, possibilitando a atualização do nome e alterando a disponibilidade da chave, indicando que ela já não pode mais ser usada. Prioridade: ☒ Essencial ☐ Importante ☐ Desejável
[RF008] Exclusão de Chave	O sistema deve permitir que o responsável com a função de administrador exclua uma chave cadastrada. Prioridade: ☒ Essencial ☐ Importante ☐ Desejável
[RF009] Cadastro de Responsável	O sistema deve permitir o cadastro de novos responsáveis, informando nome, SIAPE (quando servidor público federal), matrícula (quando aluno da instituição), CPF, data de nascimento, e-mail, senha e situação (ativo ou inativo). Regra de negócio: Caso não existam responsáveis cadastrados no sistema, o primeiro responsável deverá ser automaticamente cadastrado com a função de administrador. Prioridade: ☒ Essencial ☐ Importante ☐ Desejável
[RF010] Listagem de Responsáveis	O sistema deve permitir que o responsável com a função de administrador visualize os responsáveis cadastrados, apresentando o nome, SIAPE (quando servidor público federal), matrícula (quando aluno da instituição), CPF, data de nascimento, e-mail e situação (ativo ou inativo).

	Prioridade: ☒ Essencial ☐ Importante ☐ Desejável
[RF011] Edição de Responsável	O sistema deve permitir que o responsável edite seus dados pessoais, possibilitando a atualização do nome, da data de nascimento, do e-mail e da senha. Os campos CPF, SIAPE, matrícula, ativo e função só poderão ser alterados pelo responsável com a função de administrador. Prioridade: ☒ Essencial ☐ Importante ☐ Desejável
[RF012] Exclusão de Responsável	O sistema deve permitir que o responsável com a função de administrador exclua um responsável cadastrado. Regra de negócio: Antes da exclusão a situação do responsável precisa ser mudada para inativa. Prioridade: ☒ Essencial ☐ Importante ☐ Desejável
[RF013] Cadastro de Reserva	O sistema deve permitir que qualquer responsável cadastre novas reservas, informando a sala, a data, o horário de início e término, a frequência de utilização (única, semanal, quinzenal ou mensal) e, quando aplicável, os dias da semana em que a reserva será realizada. Regras de negócio: • Toda reserva deverá ser cadastrada inicialmente com o status ativa. Ao término do período reservado, seu status deverá ser alterado para finalizada ou cancelada, conforme a situação. • O sistema não deverá permitir cadastro de reservas com conflito de data, horário e sala em relação a outras reservas ativas. • O responsável que realizar o cadastro da reserva deverá ser automaticamente associado a ela.

	Prioridade: ☒ Essencial ☐ Importante ☐ Desejável
[RF014] Listagem de Reservas	O sistema deve permitir que qualquer responsável consulte as reservas cadastradas, exibindo informações como a sala reservada, o responsável, o período de utilização, a frequência, os dias da semana previstos para uso e o status da reserva (ativa, finalizada ou cancelada). Prioridade: ☒ Essencial ☐ Importante ☐ Desejável
[RF015] Edição de Reserva	O sistema deve permitir que o responsável edite as reservas por ele cadastradas, possibilitando a alteração da sala, da data, do horário, da frequência de utilização e dos dias da semana, quando aplicável. Prioridade: ☒ Essencial ☐ Importante ☐ Desejável
[RF016] Exclusão de Reserva	O sistema deve permitir que o responsável exclua as reservas cadastradas por ele. Regra de negócio: A exclusão de uma reserva somente poderá ser realizada caso ela não esteja com o status ativa. Prioridade: ☒ Essencial ☐ Importante ☐ Desejável
[RF017] Cadastro de Retirada	O sistema deve permitir que qualquer responsável registre novas retiradas de chaves, informando a chave a ser retirada, o horário previsto para devolução e, quando aplicável, a reserva à qual a retirada está vinculada. Regras de negócio: • O sistema deverá registrar automaticamente a data e o horário da retirada. • A data e o horário da devolução deverá ser registrado quando a chave for devolvida.

	• Toda retirada deverá ser cadastrada inicialmente com o status retirada, podendo ser alterada para devolvida ou atrasada, conforme a situação. • O responsável que realizar a retirada deverá ser automaticamente associado ao registro. • O sistema deverá marcar a chave e a sala como indisponíveis enquanto a retirada permanecer com o status retirada ou atrasada. • Após o registro da devolução, a chave e a sala deverão ser novamente marcadas como disponíveis. Prioridade: ☒ Essencial ☐ Importante ☐ Desejável
[RF018] Listagem de Retiradas	O sistema deve permitir que qualquer responsável consulte as retiradas cadastradas, exibindo informações como a sala reservada, o responsável, se é referente há uma reserva, o período de utilização e o status da retirada (retirada, devolvida ou atrasada). Prioridade: ☒ Essencial ☐ Importante ☐ Desejável
[RF019] Edição de Retirada	O sistema deve permitir que apenas o responsável com a função de administrador realize alterações nos registros de retirada. Prioridade: ☒ Essencial ☐ Importante ☐ Desejável
[RF020] Exclusão de Retirada	O sistema deve permitir que apenas o responsável com a função de administrador realize a deleção nos registros de retirada. Regra de negócio: A exclusão de uma retirada somente poderá ser realizada quando seu status for devolvida. Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

[RF021] Consulta ao Histórico de Retiradas	O sistema deve permitir que qualquer responsável consulte o histórico das retiradas, apresentando todas as informações relacionadas ao registro em uma única visualização. Prioridade: ☐ Essencial ☒ Importante ☐ Desejável
[RF022] Filtragem dos Dados	O sistema deve permitir que os responsáveis realizem consultas utilizando filtros de busca sobre histórico, facilitando a localização das informações desejadas. Prioridade: ☐ Essencial ☐ Importante ☒ Desejável

Fonte: elaborado pelos autores.

3.4.2 Requisitos Não Funcionais

Os requisitos não funcionais referem-se às características de qualidade e às restrições do sistema, não estando diretamente relacionados às funcionalidades disponibilizadas aos usuários (SOMMERVILLE, 2011). Dessa forma, a seguir são apresentados os requisitos não funcionais definidos para o sistema KeyControl.

Tabela 3 - Tabela de requisitos não funcionais

Requisitos	Descrição do Requisito
[RNF001] Usabilidade	O sistema deve possuir uma interface intuitiva e de fácil utilização, com fluxos simplificados para o cadastro, consulta e gerenciamento das informações, com poucas etapas necessárias para a execução das operações. Prioridade: ☒ Essencial ☐ Importante ☐ Desejável
[RNF002] Disponibilidade	O sistema deve permanecer disponível durante o horário de funcionamento da instituição, garantindo que os usuários possam acessar suas funcionalidades sempre que necessário. Em caso de

	indisponibilidade, o serviço deverá ser restabelecido no menor tempo possível. Prioridade: ☒ Essencial ☐ Importante ☐ Desejável
[RNF003] Desempenho	O sistema deve responder às solicitações dos usuários em até 2 segundos para operações de consulta e em até 5 segundos para operações de cadastro, edição e exclusão, considerando condições normais de uso. Segundo Nielsen (1993), tempos de resposta inferiores a 10 segundos tendem a manter a atenção do usuário, reduzindo a perda de foco durante a interação com o sistema. Prioridade: ☒ Essencial ☐ Importante ☐ Desejável
[RNF004] Controle de Permissões	O sistema deve restringir funcionalidades de acordo com o perfil do usuário, garantindo que apenas administradores possam alterar ou remover registros críticos. Prioridade: ☒ Essencial ☐ Importante ☐ Desejável
[RNF005] Integridade dos Dados	O sistema deve garantir a integridade e a consistência das informações armazenadas, impedindo o registro de dados inválidos, duplicados ou inconsistentes, bem como preservando o relacionamento entre salas, chaves, responsáveis, reservas e retiradas. Prioridade: ☒ Essencial ☐ Importante ☐ Desejável
[RNF006] Compatibilidade	O sistema deve ser compatível com as versões mais recentes dos navegadores Google Chrome, Microsoft Edge e Mozilla Firefox. Além disso, deverá possuir interface responsiva, permitindo sua utilização em computadores, tablets e smartphones. Prioridade: ☐ Essencial ☒ Importante ☐ Desejável

[RNF007] Persistência de Dados	O sistema deve utilizar um banco de dados relacional para o armazenamento persistente das informações, garantindo que os dados permaneçam íntegros e disponíveis após reinicializações do sistema. Prioridade: ☒ Essencial ☐ Importante ☐ Desejável
---------------------------------------	--

Fonte: elaborado pelos autores.

3.4.3 User Stories

Nesta seção são apresentadas as User Stories, que descrevem as funcionalidades do sistema a partir da perspectiva dos responsáveis e administradores que utilizarão o KeyControl.

Tabela 4 - Tabela de user stories relacionado aos perfis de responsáveis e administradores

Código	User Story
US001 - Cadastro de Sala	Como administrador, eu quero cadastrar salas, para que os espaços físicos da instituição possam ser gerenciados pelo sistema.
US002 - Gerenciamento de Salas	Como administrador, eu quero visualizar, editar e excluir salas cadastradas, para manter suas informações atualizadas.
US003 - Cadastro de Chaves	Como administrador, eu quero cadastrar chaves vinculadas às salas, para controlar os recursos físicos disponíveis.
US004 - Gerenciamento de Chaves	Como administrador, eu quero visualizar, editar e excluir chaves cadastradas, para manter o controle das chaves da instituição.
US005 - Cadastro de Responsável	Como administrador, eu quero cadastrar novos responsáveis, para que possam utilizar o sistema conforme suas permissões.
US006 - Gerenciamento de Responsáveis	Como administrador, eu quero visualizar, editar e excluir responsáveis, para manter os cadastros atualizados.
US007 - Cadastro de Reserva	Como responsável, eu quero reservar uma sala, para garantir sua disponibilidade

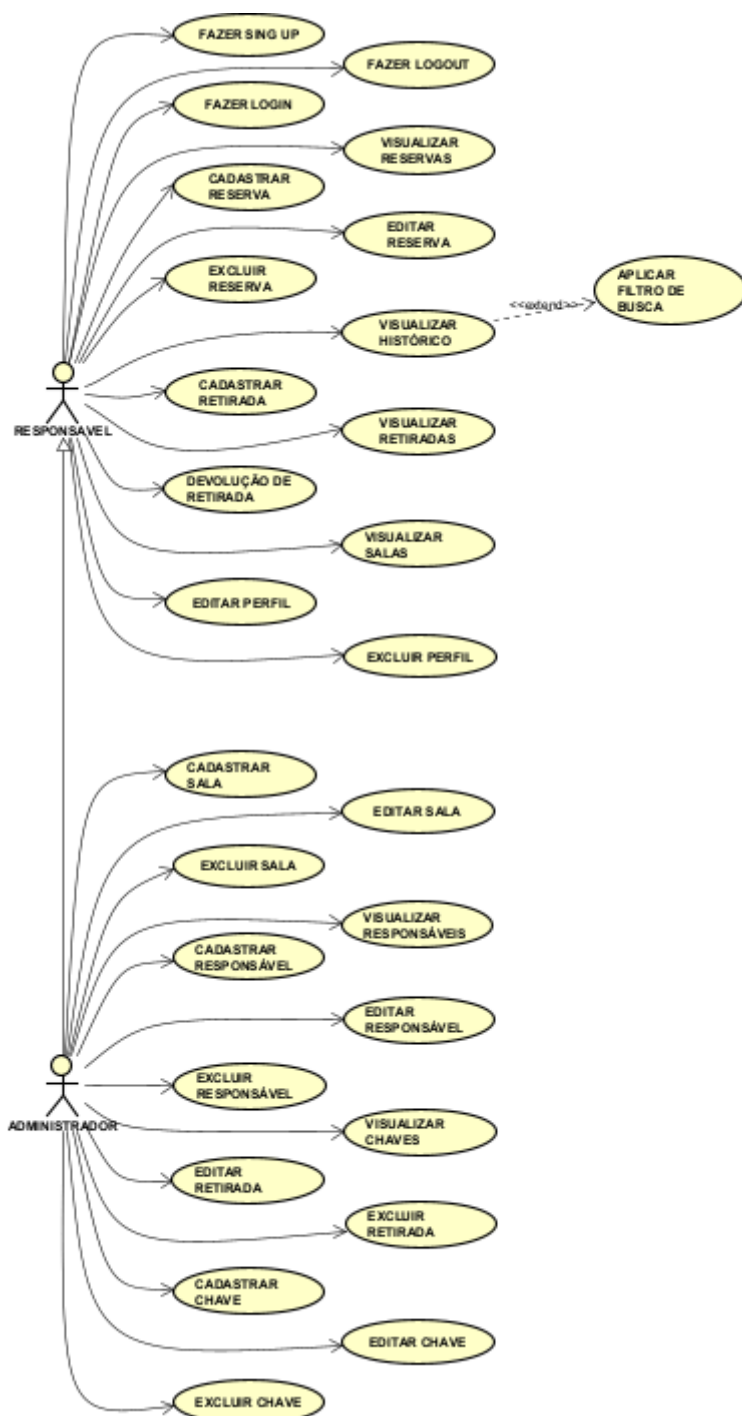
	durante o período desejado.
US008 - Gerenciamento de Reservas	Como responsável, eu quero consultar, editar e excluir minhas reservas, para administrar a utilização das salas.
US009 - Registro de Retirada	Como responsável, eu quero registrar a retirada de uma chave, para utilizar a sala correspondente.
US010 - Registro de Devolução	Como responsável, eu quero registrar a devolução da chave utilizada, para disponibilizá-la novamente para outros usuários.
US011 - Gerenciamento de Retiradas	Como administrador, eu quero editar ou excluir registros de retirada, para corrigir eventuais inconsistências nos dados.
US012 - Consulta ao Histórico	Como usuário, eu quero consultar o histórico de retiradas, para acompanhar as operações realizadas no sistema.
US013 - Filtragem de Dados	Como usuário, eu quero utilizar filtros de busca, para localizar rapidamente as informações desejadas.

Fonte: elaborado pelos autores.

3.4.4 Diagrama de Casos de Usos

A Figura 1 apresenta as funcionalidades e os perfis de usuário do sistema em forma de diagrama de casos de uso.

Figura 1 - Diagrama de Casos de Uso do Sistema KeyControl



Fonte: elaborado pelos autores.

3.5 Ferramentas e Tecnologias

3.5.1 Linguagem de programação

No backend, foi utilizado a linguagem de programação Python, em conjunto com o framework Flask, por oferecer uma estrutura leve, flexível e de fácil manutenção para o desenvolvimento de aplicações web.

No frontend, foi utilizado a linguagem de programação Javascript, juntamente com a biblioteca React e o Material UI(MUI), que fornece um conjunto de componentes de interface baseados no conceito de Material Design. O MUI foi utilizado como biblioteca de componentes visuais, para obter uma padronização e melhor experiência do usuário.

3.5.2 Backend

O desenvolvimento do backend foi realizado utilizando a linguagem de programação Python em conjunto com o framework Flask. A escolha dessas tecnologias deve-se à sua simplicidade, flexibilidade e eficiência no desenvolvimento de aplicações web, permitindo a construção de uma API REST organizada, modular e de fácil manutenção. A arquitetura da aplicação foi desenvolvida seguindo os princípios da arquitetura REST, utilizando o protocolo HTTP para comunicação entre o frontend e o backend.

Para o acesso e manipulação dos dados, foi utilizado o SQLAlchemy como ferramenta de Mapeamento Objeto-Relacional (ORM), facilitando a comunicação entre a aplicação e o banco de dados relacional e reduzindo a necessidade de escrita direta de comandos SQL na maior parte do projeto. O gerenciamento das alterações no banco de dados foi realizado por meio do Flask-Migrate, permitindo o versionamento e a aplicação controlada das migrações do banco de dados.

Para garantir a segurança da aplicação, foi utilizado o padrão JWT (JSON Web Token) para o processo de autenticação e autorização dos usuários. O JWT permite a criação de tokens de acesso assinados pelo servidor, que são utilizados para identificar o usuário autenticado e controlar o acesso às funcionalidades protegidas do sistema. Após a autenticação, o usuário recebe um token que é enviado nas requisições subsequentes ao backend, permitindo a identificação do usuário.

Por fim foi adotado o mecanismo de armazenamento do token por meio de cookies com a flag HttpOnly, impedindo que o conteúdo do token seja acessado por scripts executados no navegador.

3.5.3 Frontend

O desenvolvimento do frontend foi realizado utilizando a biblioteca React, baseada na linguagem JavaScript, juntamente com o framework de construção Vite. O React foi escolhido por possibilitar o desenvolvimento da interface da aplicação por meio de componentes reutilizáveis, permitindo maior organização, modularidade, facilidade de manutenção do código e popularidade.

Para a construção dos elementos visuais da aplicação, foi utilizada a biblioteca Material UI (MUI), que disponibiliza componentes baseados no conceito de Material Design. A utilização do MUI contribuiu para a padronização da interface, proporcionando consistência visual e melhor experiência de utilização do sistema.

Também foram utilizados componentes complementares do ecossistema MUI, como MUI X Data Grid, para apresentar e manipular dados em tabelas, MUI X Date Pickers, para seleção de datas, MUI Icons para exibição de ícones.

Optamos por utilizar a biblioteca Sonner para exibição de notificações e mensagens de retorno ao usuário durante a interação com o sistema.

Dessa forma, o frontend ficou responsável pela camada de apresentação e interação do sistema, disponibilizando aos usuários as funcionalidades relacionadas ao gerenciamento de salas, chaves, reservas e registros de movimentações.

3.5.4 Banco de Dados

O banco de dados principal utilizado no desenvolvimento do sistema foi o PostgreSQL, um Sistema Gerenciador de Banco de Dados Relacional (SGBD) de código aberto, escolhido por sua robustez, confiabilidade e capacidade de garantir a integridade e a consistência das informações armazenadas. Sua utilização permitiu o armazenamento estruturado dos dados referentes às salas, chaves, responsáveis, reservas e retiradas do sistema.

Complementarmente, foi utilizado o Redis, um banco de dados em memória do tipo chave-valor (key-value), empregado como mecanismo de cache para armazenar temporariamente informações frequentemente consultadas. Sua utilização contribuiu para a redução do tempo de resposta da aplicação e para a diminuição da carga de processamento sobre o banco de dados principal.

Ademais, foi integrado o Apache Solr, uma plataforma de busca open-source baseada no Apache Lucene, responsável pela indexação e pesquisa dos dados do sistema. Sua utilização possibilitou a realização de consultas e filtros de forma mais rápida e eficiente, contribuindo para o desempenho das funcionalidades de pesquisa disponibilizadas pelo KeyControl.

3.5.5 Controle de Versão

Para o controle de versão do projeto, foram utilizados o Git e a plataforma GitHub, permitindo o gerenciamento do código-fonte, o registro do histórico de alterações, a

colaboração entre os desenvolvedores e o acompanhamento da evolução do sistema ao longo do desenvolvimento.

3.5.6 Ambiente de desenvolvimento

O desenvolvimento do sistema foi realizado utilizando o Visual Studio Code (VS Code), escolhido por sua interface intuitiva, ampla disponibilidade de extensões e suporte às tecnologias empregadas no projeto, como Python, Flask e React.

Para garantir a padronização e facilitar a configuração do ambiente de desenvolvimento, foi utilizado o Docker, permitindo a containerização dos serviços que compõem a aplicação. Dessa forma, o backend, o banco de dados PostgreSQL, o Redis e o Apache Solr foram executados em contêineres independentes, assegurando maior portabilidade, isolamento entre os serviços e consistência entre os ambientes de desenvolvimento.

A orquestração dos contêineres foi realizada por meio do Docker Compose, possibilitando a inicialização e o gerenciamento de todos os serviços necessários ao funcionamento do sistema a partir de um único arquivo de configuração. Essa abordagem simplifica a implantação da aplicação e reduz problemas relacionados à instalação de dependências e configuração do ambiente.

Para automatizar o processo de implantação da aplicação, foi utilizado o Jenkins como ferramenta de Integração Contínua (Continuous Integration – CI). O Jenkins foi configurado para monitorar o repositório hospedado no GitHub e, sempre que uma nova alteração era enviada ao repositório, executava automaticamente uma pipeline responsável por reconstruir a imagem Docker da aplicação, interromper a instância anteriormente em execução e iniciar uma nova versão contendo as alterações mais recentes. Essa abordagem reduziu a necessidade de intervenções manuais durante o processo de implantação, garantindo maior agilidade, padronização e confiabilidade na atualização do sistema.

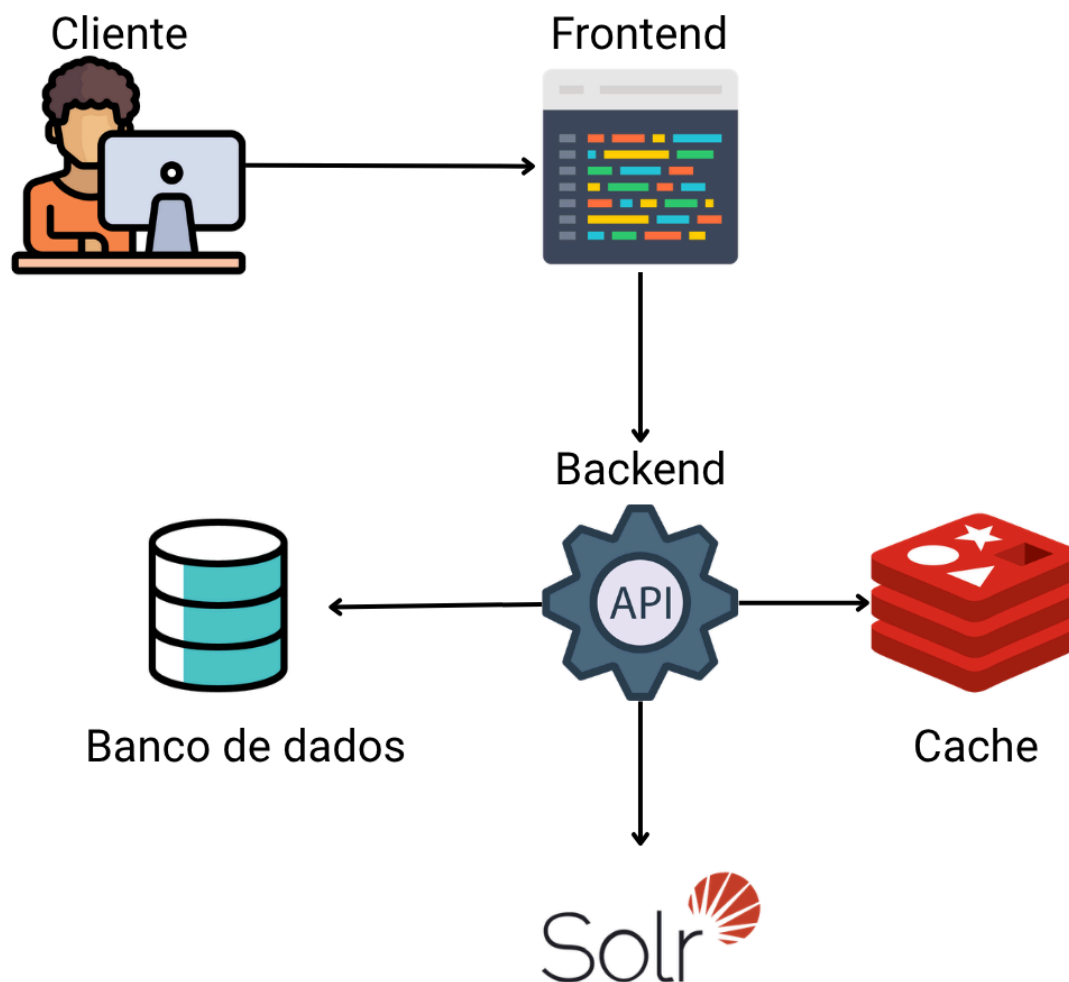
4 RESULTADOS E DISCUSSÃO

4.1 Arquitetura do Sistema

O KeyControl foi desenvolvido utilizando uma arquitetura em camadas baseada no modelo cliente-servidor. A solução é composta pelas camadas de frontend, backend, banco de dados, cache e mecanismo de busca especializada, que atuam de forma integrada para garantir organização, desempenho, escalabilidade e facilidade de manutenção. A Figura 2 apresenta a arquitetura geral do sistema.

A Figura 2 apresenta o fluxo de acesso ao sistema, começando do usuário entrando na aplicação através do frontend, enviando requisições para o backend e recebendo a resposta.

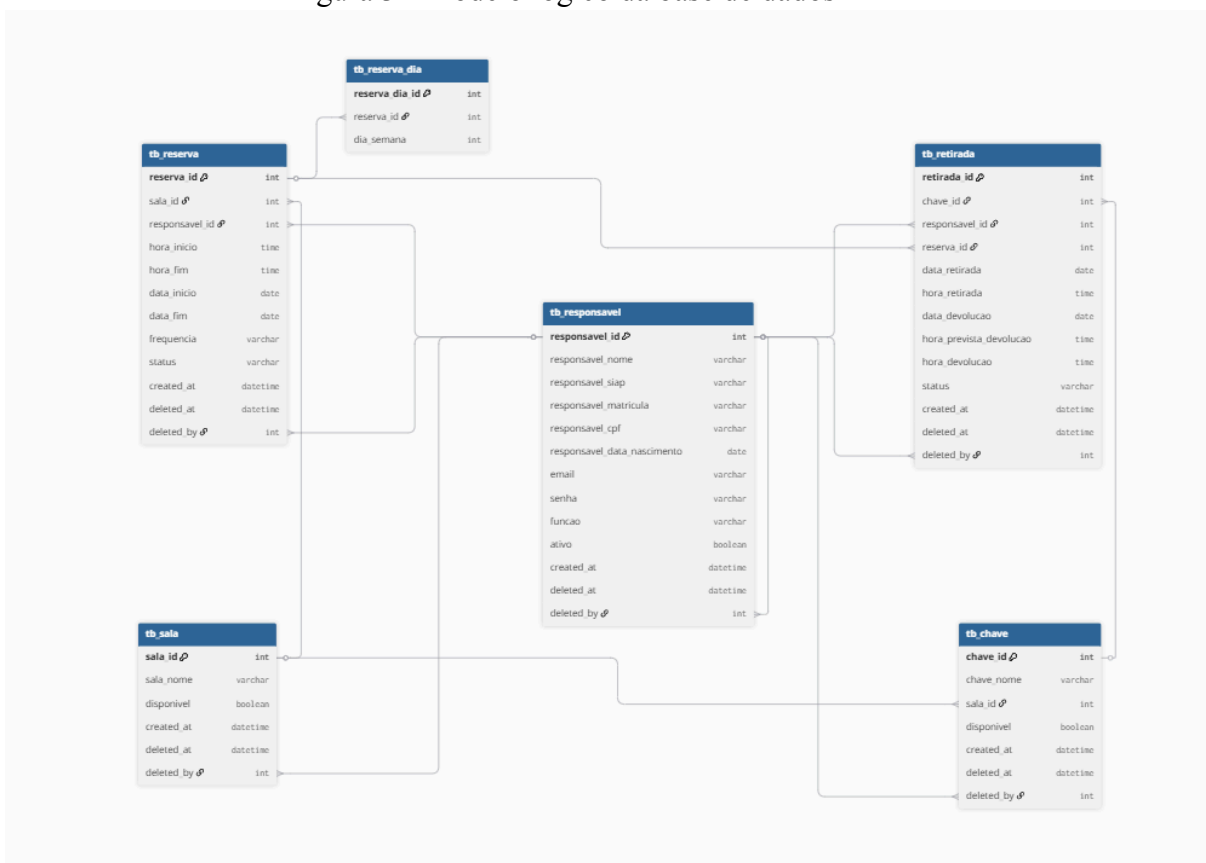
Figura 2 - Arquitetura do Sistema



4.2 Modelagem do Sistema

A Figura 3 apresenta a modelagem das tabelas do banco de dados, representando todas as entidades que o sistema utiliza.

Figura 3 - Modelo lógico da base de dados



Fonte: elaborado pelos autores.

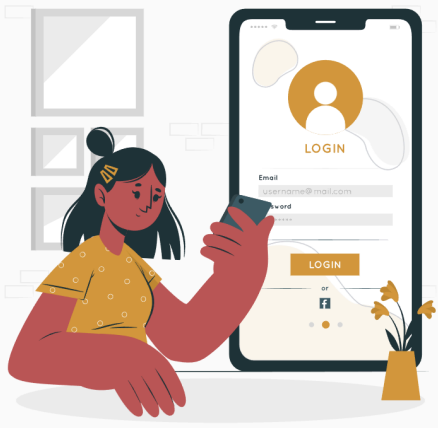
4.3 Telas do Sistema

As telas apresentadas nesta seção correspondem às principais funcionalidades desenvolvidas para o sistema KeyControl. A interface foi estruturada de modo a possibilitar o gerenciamento de usuários, salas, reservas e movimentações de chaves, contribuindo para a organização e o controle dos recursos institucionais relacionados. A seguir, são apresentadas as principais telas da aplicação e suas respectivas funcionalidades.

A Figura 4 apresenta a tela de login, responsável pela autenticação dos usuários e pelo controle de acesso às funcionalidades do sistema.

Figura 4 - Tela de login

KeyControl



Bem-Vindo

Email de usuário

Senha

Entrar

[Novo aqui?](#) [Cadastre-se](#)

localhost:5173

Fonte: elaborado pelos autores.

A Figura 5 apresenta a tela de cadastro de usuários, destinada ao registro das informações necessárias para a identificação e o gerenciamento dos usuários do sistema.

Figura 5 - Tela de cadastro de usuários

KeyControl

Já possui uma conta? [Clique aqui](#)

Cadastre-se

Nome de usuário

Email

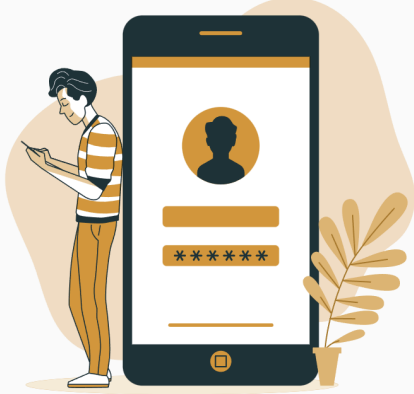
CPF

Data de nascimento

SIAP Ou

Senha

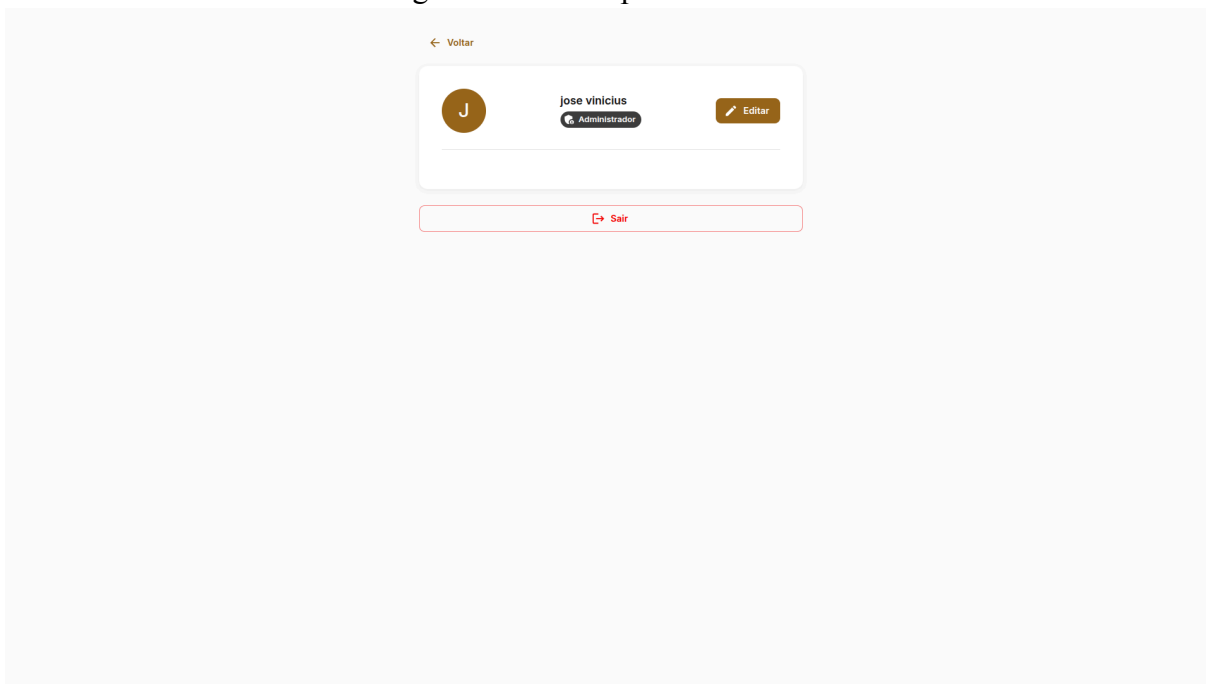
Cadastrar



Fonte: elaborado pelos autores.

Figura 6 apresenta a tela de perfil do usuário, desenvolvida para exibir as informações do autenticado e permitir o acesso às funcionalidades relacionadas ao gerenciamento de seus dados.

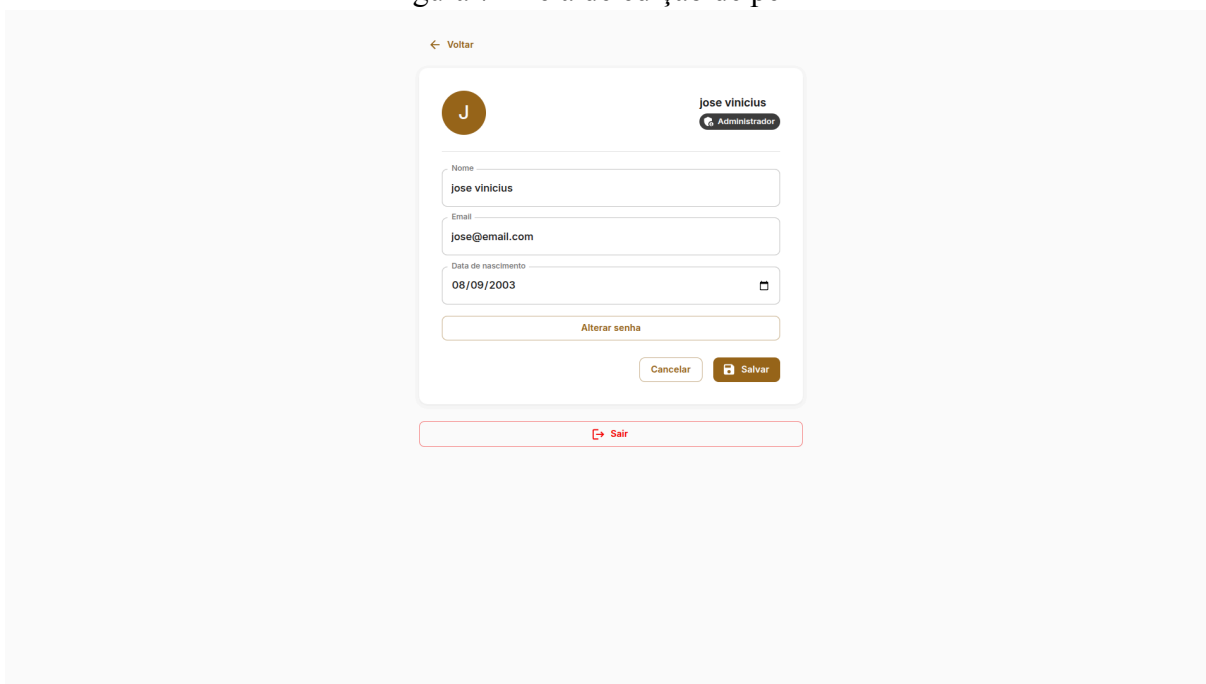
Figura 6 - Tela de perfil



Fonte: elaborado pelos autores.

Figura 7 apresenta a tela de edição de perfil, utilizada pelo usuário autenticado para realizar alterações em seus dados cadastrais.

Figura 7 - Tela de edição de perfil

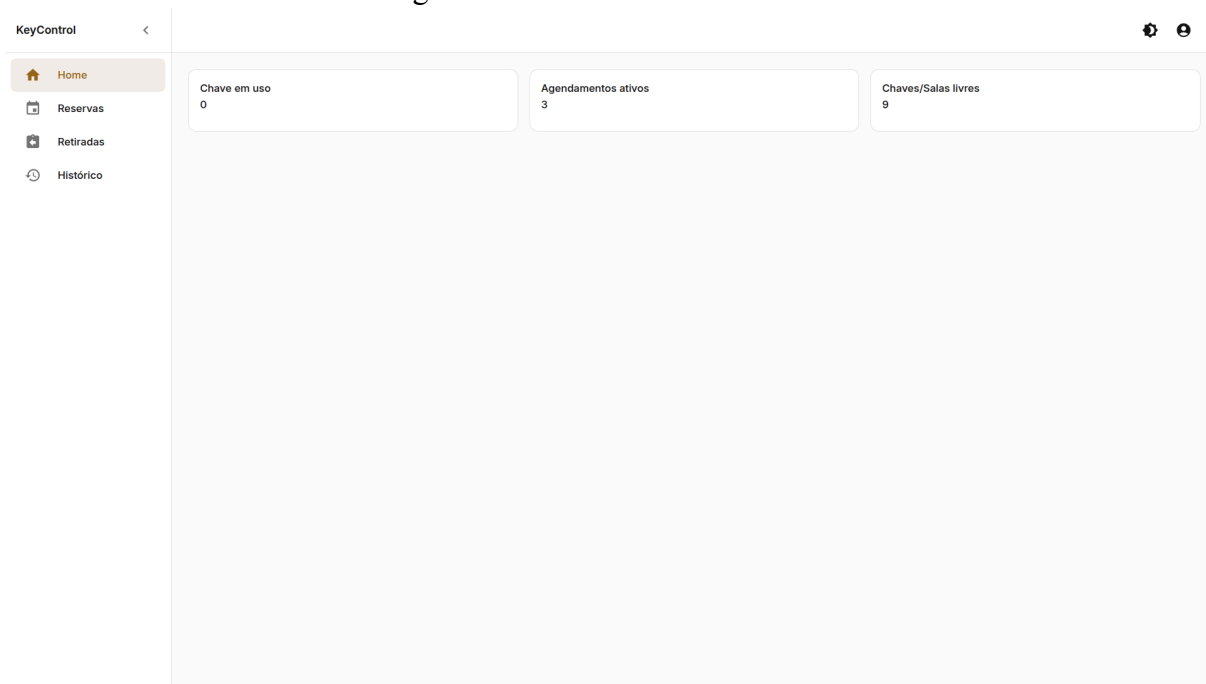


A tela de edição de perfil apresenta um formulário centralizado. No topo, há um link "Voltar" com uma seta para trás. O formulário contém: um avatar com a letra "J"; o nome de usuário "jose vinicius" com o cargo "Administrador"; campos de texto para "Nome" (preenchido com "jose vinicius"), "Email" (preenchido com "jose@email.com") e "Data de nascimento" (preenchido com "08/09/2003" e com ícone de calendário); um botão "Alterar senha"; e botões "Cancelar" e "Salvar" com ícone de disco. Abaixo do formulário, há um botão "Sair" com ícone de porta e seta.

Fonte: elaborado pelos autores.

Figura 8 apresenta a tela de início do sistema, aqui são exibidos informações sobre o gerenciamento de salas, chaves e agendamentos.

Figura 8 - Tela de início

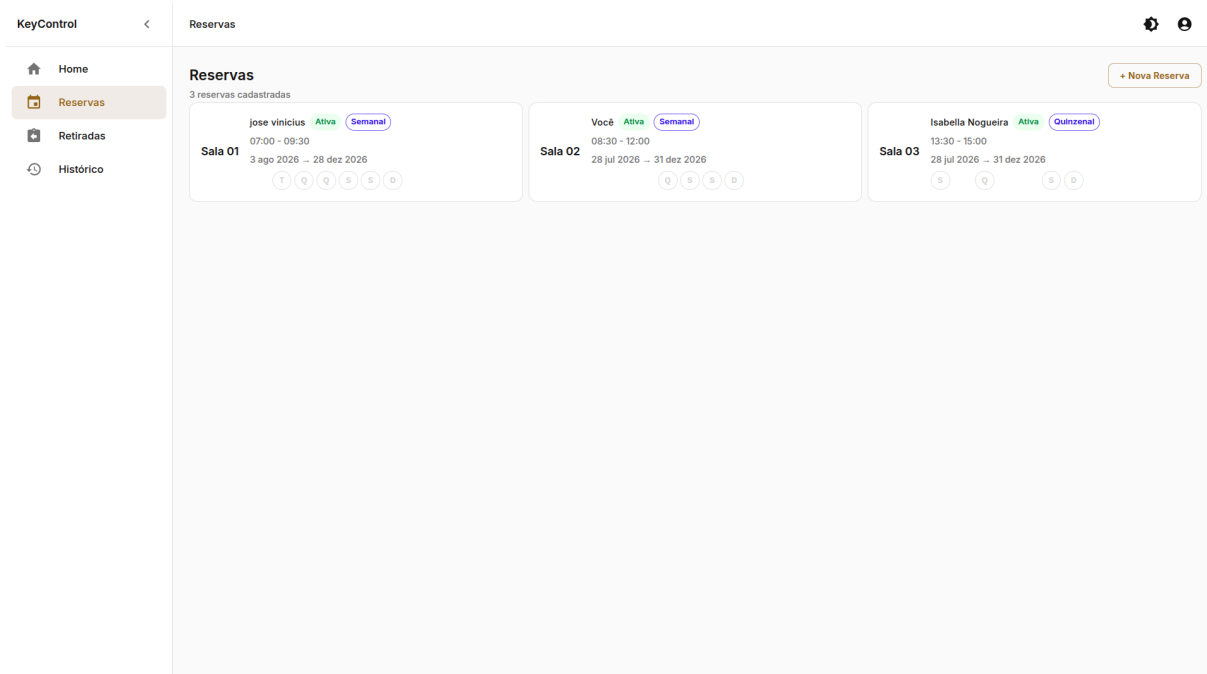


A tela de início do sistema possui uma barra lateral esquerda com o título "KeyControl" e um menu com as opções: "Home" (destacada), "Reservas", "Retiradas" e "Histórico". No topo direito, há ícones de configurações e perfil. O conteúdo principal exibe três cartões de resumo: "Chave em uso" com o valor "0", "Agendamentos ativos" com o valor "3" e "Chaves/Salas livres" com o valor "9".

Fonte: elaborado pelos autores.

Figura 9 apresenta a tela de reservas de salas e chaves de um usuário do tipo responsável ou usuário comum, o responsável pode visualizar todas as reservas e realizar a criação e gerenciar suas próprias reservas.

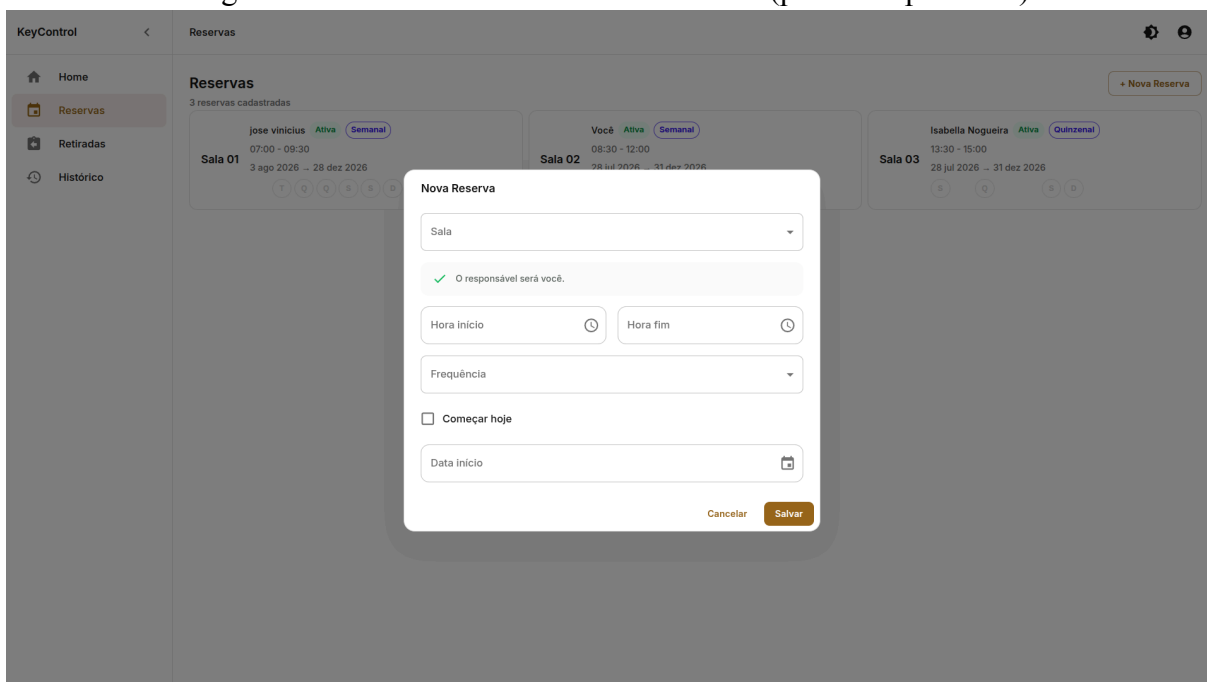
Figura 9 - Tela de reservas (perfil Responsável)



Fonte: elaborado pelos autores

Figura 10 apresenta a tela de cadastro de nova reserva de um perfil de responsável, o usuário insere os dados para registrar sua reserva.

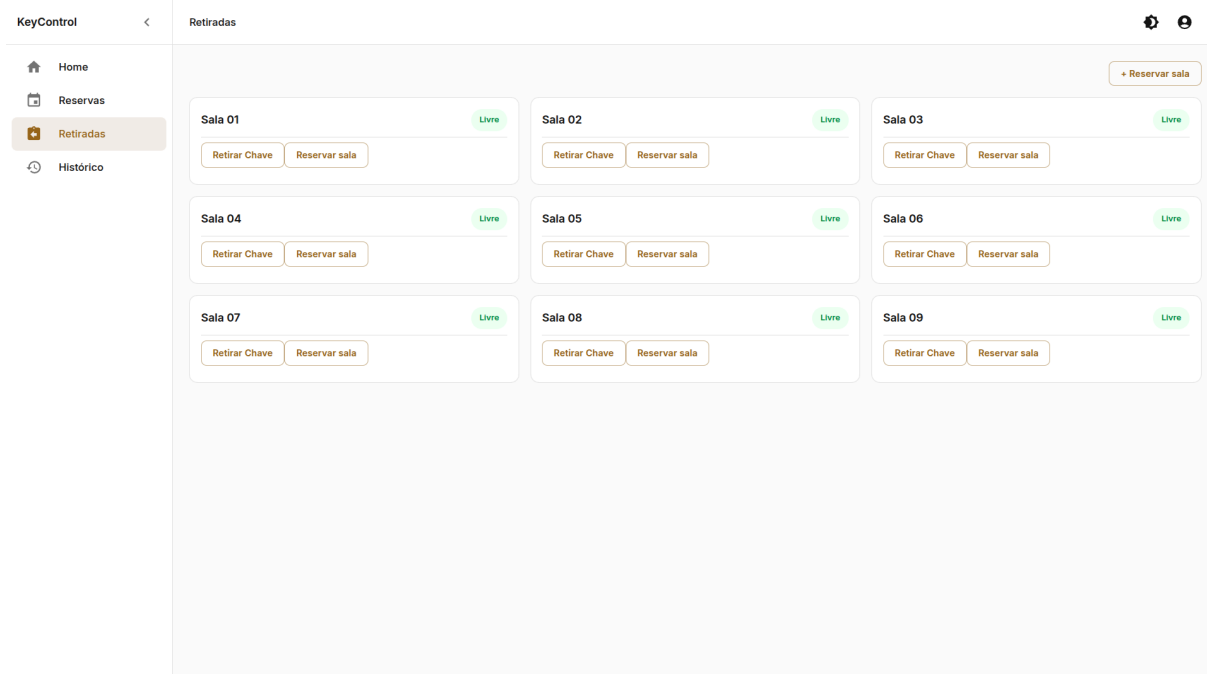
Figura 10 - Tela de cadastro de nova reserva (perfil Responsável)



Fonte: elaborado pelos autores.

Figura 11 apresenta a tela de retiradas do perfil de um responsável, listagem das salas do sistema e a possibilidade de retirar ou reservar uma sala.

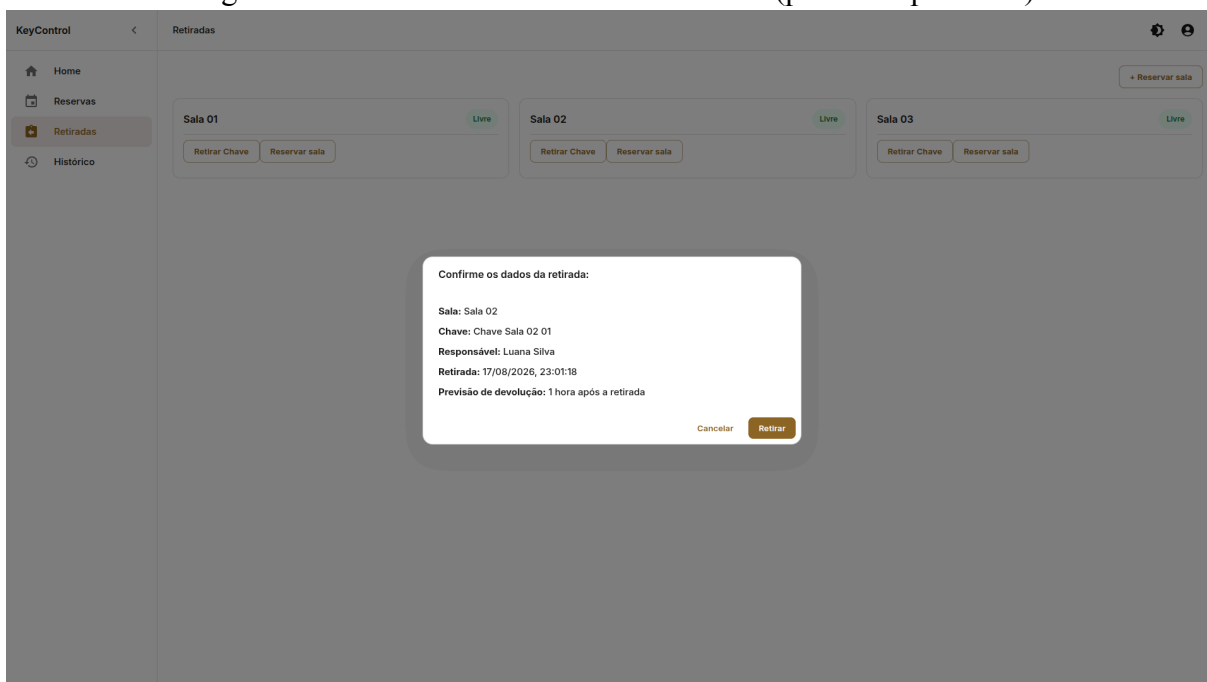
Figura 11 - Tela de retiradas (perfil Responsável)



Fonte: elaborado pelos autores.

Figura 12 apresenta a tela de cadastro de uma retirada, onde é feita a exibição dos dados e a confirmação de retirada de uma chave.

Figura 12 - Tela de cadastro de nova retirada (perfil Responsável)



Fonte: elaborado pelos autores.

Figura 13 apresenta a tela de histórico, nela é exibido uma listagem das transações das chaves exibindo dados como responsável, sala, chave, data e hora.

Figura 13 - Tela de histórico (perfil Responsável)

SALA	CHAVE	RESPONSÁVEL	RETIRADA	DEVOLUÇÃO PREVISTA	DEVOLUÇÃO REAL	STATUS
Sala 06	Chave Sala 06 01	jose vinicius	28/07/2026 às 22:26	23:26	29/07/2026 às 22:27	Devolvida
Sala 04	Chave Sala 04 01	jose vinicius	28/07/2026 às 22:26	23:26	29/07/2026 às 22:27	Devolvida
Sala 03	Chave Sala 03 01	jose vinicius	28/07/2026 às 22:18	23:18	29/07/2026 às 22:19	Devolvida
Sala 01	Chave Sala 01 01	jose vinicius	28/07/2026 às 21:40	22:40	29/07/2026 às 22:18	Devolvida

Itens por página 10 1-4 of 4

Fonte: elaborado pelos autores.

Figura 14 apresenta a tela de responsáveis exclusiva do administrador do sistema, é feita uma listagem dos usuários do sistema e acesso ao gerenciamento deles.

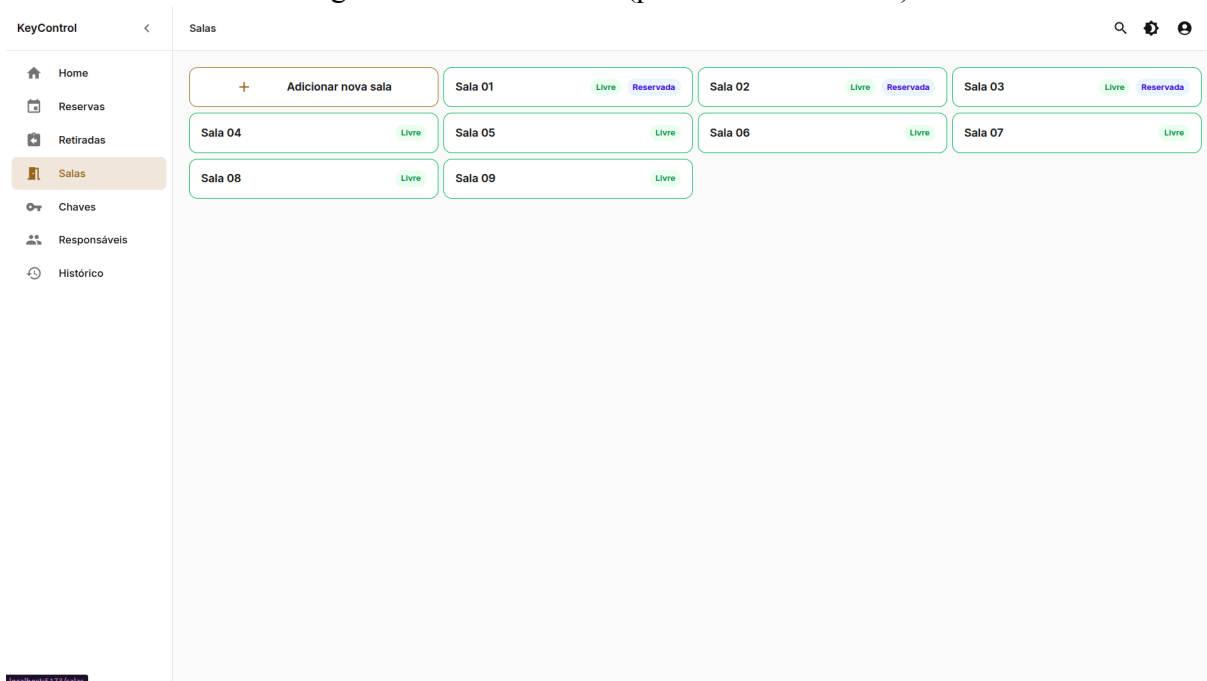
Figura 14 - Tela de responsáveis (perfil Administrador)

Nome	Email
Antonio David	david@gmail.com
Isabella Nogueira	isanogue@gmail.com

Fonte: elaborado pelos autores.

Figura 15 apresenta a tela de salas exclusiva do administrador, nela é feita a listagem das salas além do acesso a gestão das salas.

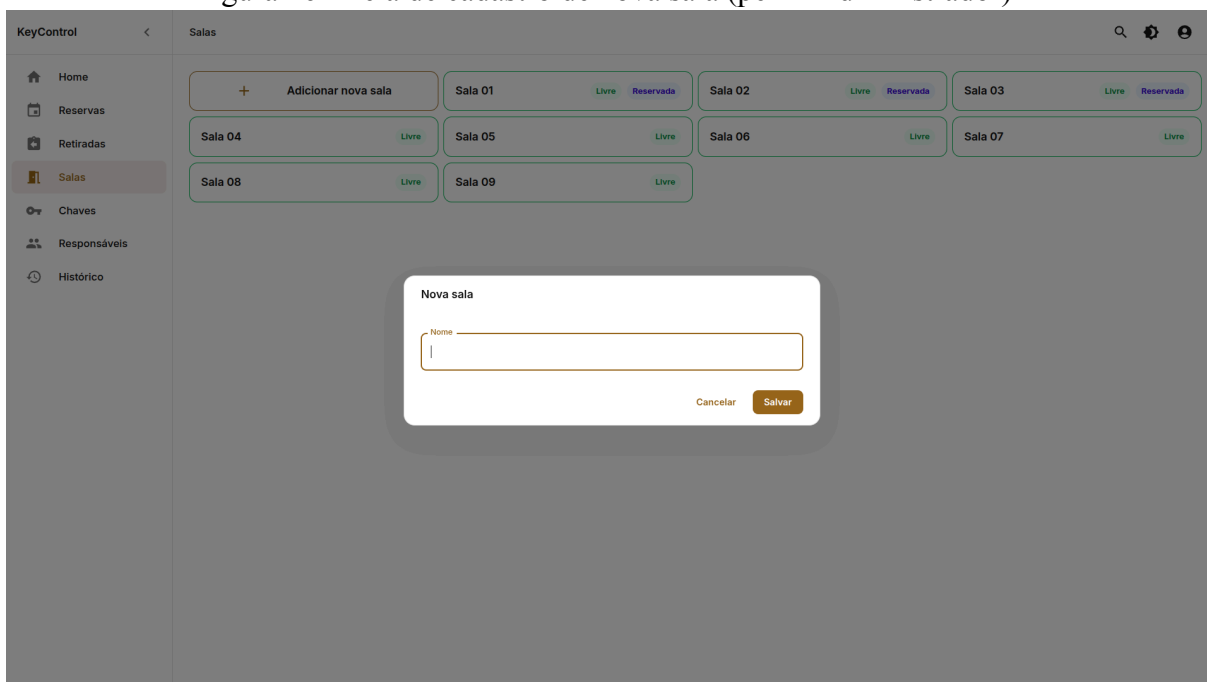
Figura 15 - Tela de salas (perfil Administrador)



Fonte: elaborado pelos autores.

Figura 16 apresenta a tela de cadastro de nova sala, ela é acessada através da tela de salas, Figura 15, e espera dados para cadastrar uma nova sala.

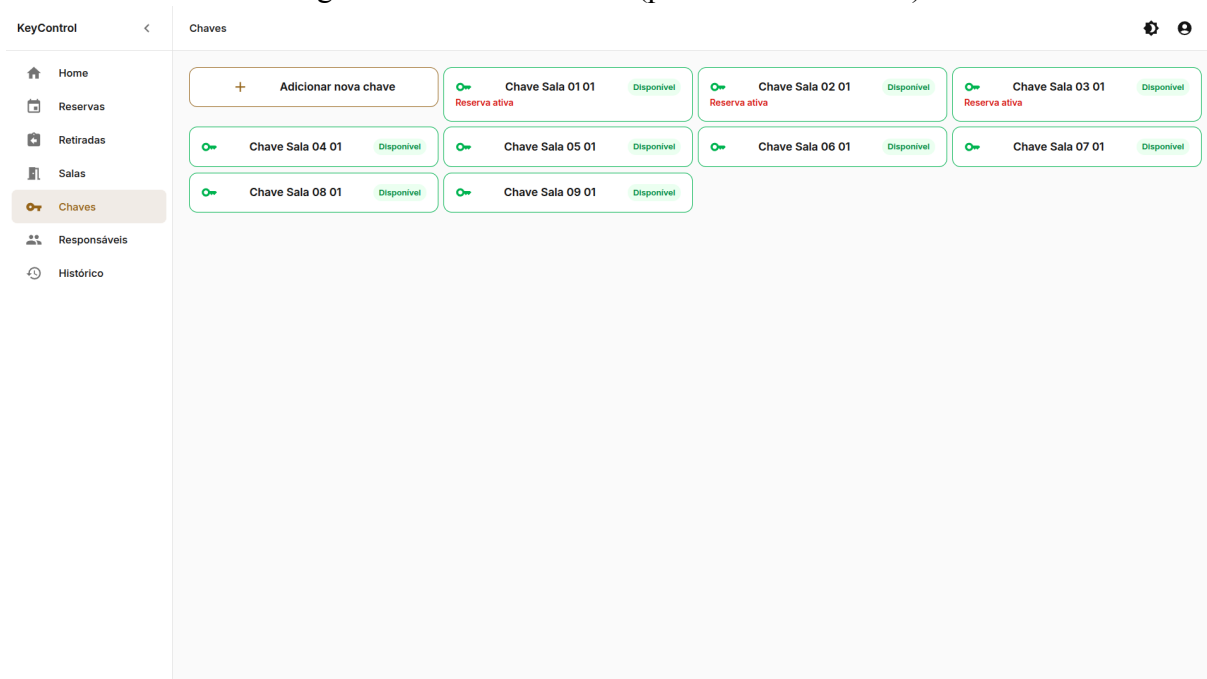
Figura 16 - Tela de cadastro de nova sala (perfil Administrador)



Fonte: elaborado pelos autores.

Figura 17 apresenta a tela de chaves exclusiva do administrador, nela é feita a listagem e gerenciamento das chaves

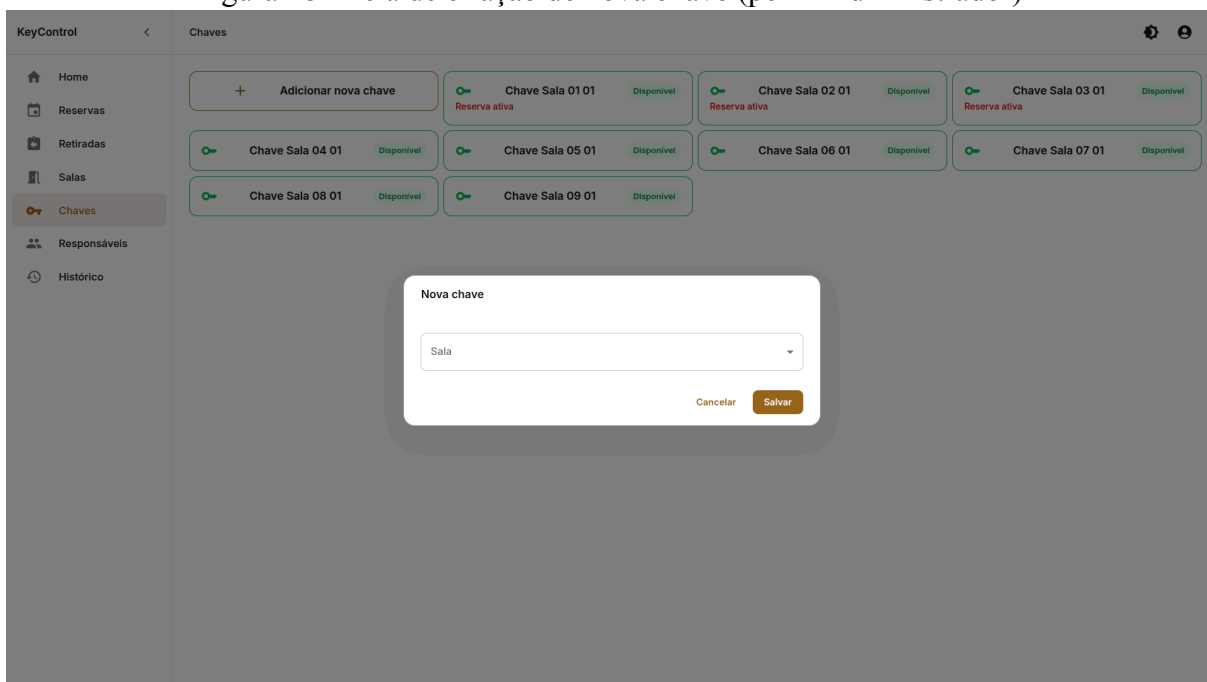
Figura 17 - Tela de chaves (perfil Administrador)



Fonte: elaborado pelos autores.

Figura 18 apresenta a tela de criação de uma nova chave, acessada através da tela de chaves, Figura 17, espera que o usuário informe a sala para registrar a nova chave.

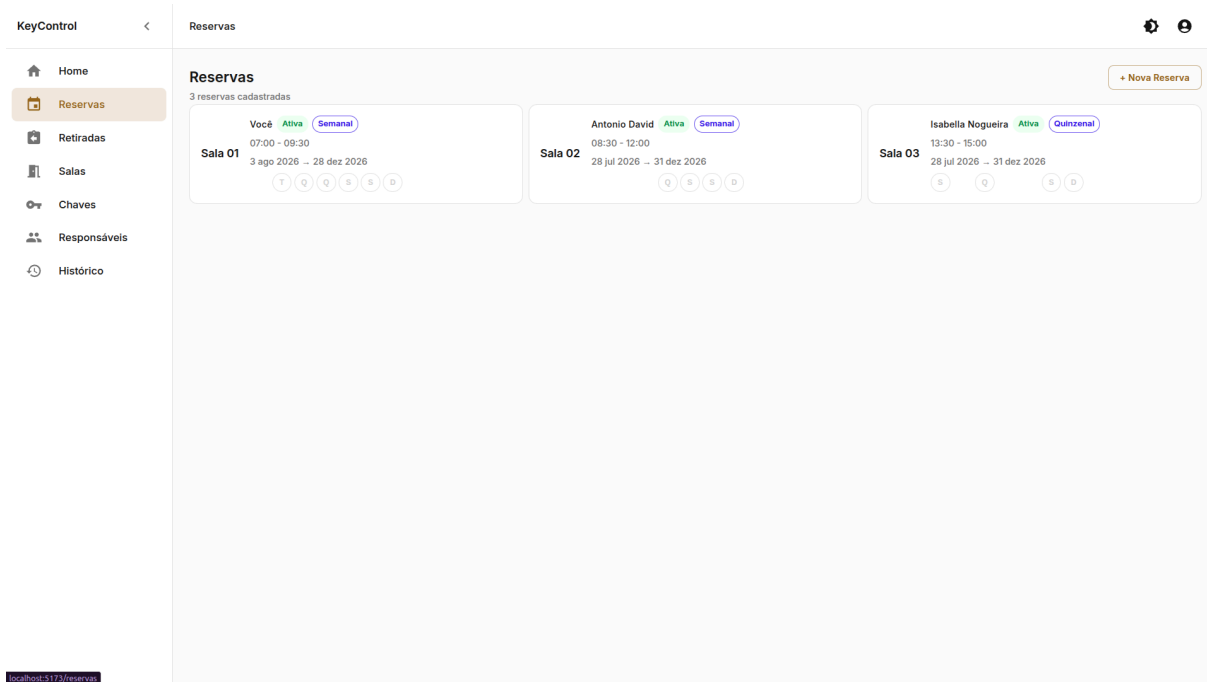
Figura 18 - Tela de criação de nova chave (perfil Administrador)



Fonte: elaborado pelos autores.

Figura 19 apresenta a tela de reservas, na visão do administrador, tem o acesso total à administração das reservas.

Figura 19 - Tela de reservas (perfil Administrador)



Fonte: elaborado pelos autores.

Figura 20 apresenta a tela de cadastro de nova reserva, acessado através da tela de reservas, figura 19, é esperado dados para realizar uma reserva.

Figura 20 - Tela de cadastro de nova reserva (perfil Administrador)

The screenshot shows the 'Nova Reserva' modal form. The form contains the following fields and options:

- Sala:** A dropdown menu.
- ☐ **Reservar para mim**
- Responsável:** A dropdown menu.
- Hora início:** A time selection field.
- Hora fim:** A time selection field.
- Frequência:** A dropdown menu.
- ☐ **Começar hoje**
- Data início:** A date selection field.
- Buttons:** 'Cancelar' and 'Salvar' at the bottom right.

The background interface shows a sidebar with 'Reservas' selected and a main area displaying a list of reservations for Sala 01, Sala 02, and Sala 03.

Fonte: elaborado pelos autores.

Figura 21 apresenta tela de retiradas, o administrador pode realizar suas próprias retiradas e também realizar as reservas.

Figura 21 - Tela de retiradas (perfil Administrador)

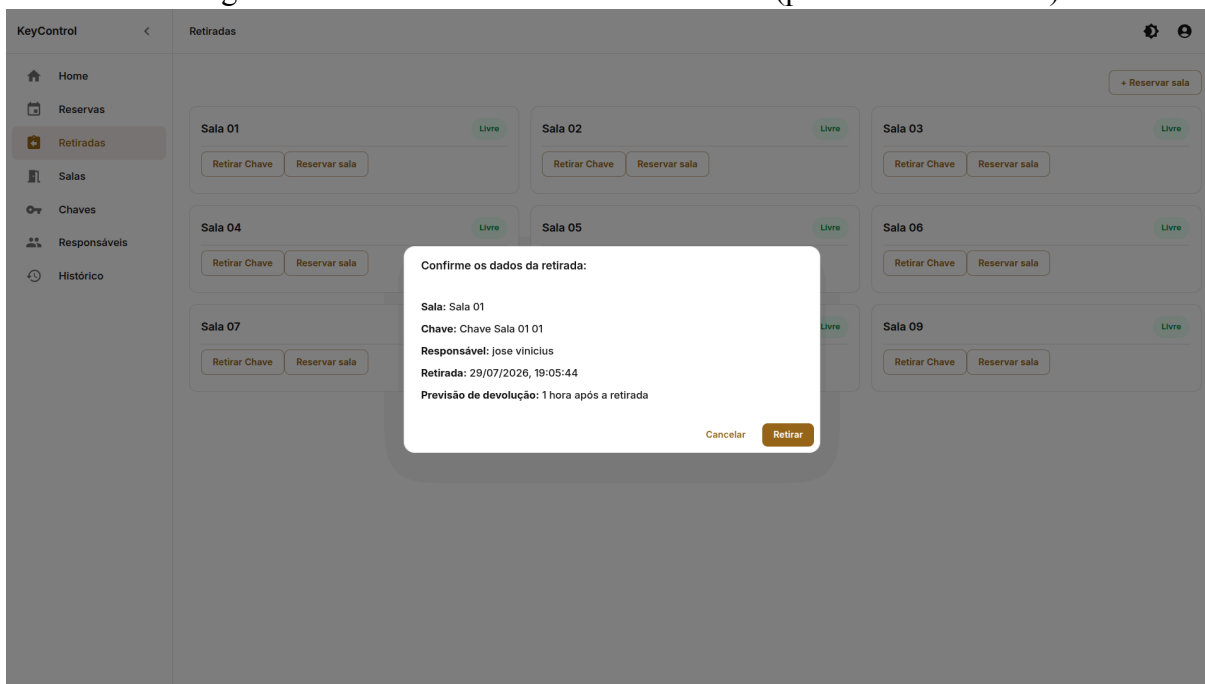
The screenshot shows the 'Retiradas' screen. The interface includes a sidebar with 'Retiradas' selected and a main area displaying a grid of rooms. Each room card shows the room number, a 'Livre' status indicator, and two buttons: 'Retirar Chave' and 'Reservar sala'.

Sala	Status	Retirar Chave	Reservar sala
Sala 01	Livre	Retirar Chave	Reservar sala
Sala 02	Livre	Retirar Chave	Reservar sala
Sala 03	Livre	Retirar Chave	Reservar sala
Sala 04	Livre	Retirar Chave	Reservar sala
Sala 05	Livre	Retirar Chave	Reservar sala
Sala 06	Livre	Retirar Chave	Reservar sala
Sala 07	Livre	Retirar Chave	Reservar sala
Sala 08	Livre	Retirar Chave	Reservar sala
Sala 09	Livre	Retirar Chave	Reservar sala

Fonte: elaborado pelos autores.

Figura 22 apresenta a tela de cadastro de nova retirada na visão de um administrador.

Figura 22 - Tela de cadastro de nova retirada (perfil Administrador)



Fonte: elaborado pelos autores.

4.4 Testes de Carga

Com o objetivo de avaliar o desempenho e a estabilidade da API desenvolvida, foram realizados testes de carga utilizando a ferramenta Apache JMeter. Os testes buscaram verificar o comportamento da aplicação diante de diferentes níveis de acesso simultâneo, analisando o tempo de resposta e a capacidade do sistema de processar requisições concorrentes sem ocorrência de falhas.

4.4.1 Metodologia dos testes

Foram definidos três cenários de carga progressiva, simulando 100, 500 e 1000 usuários simultâneos realizando requisições ao endpoint de listagem de reservas do sistema. Durante cada execução foram coletadas métricas de desempenho, incluindo:

- Tempo mínimo de resposta;
- Tempo máximo de resposta;
- Tempo médio de resposta;
- Quantidade de requisições processadas com sucesso;
- Quantidade de requisições com falha;

Os resultados obtidos são apresentados na Tabela 5.

Tabela 5 - Tabela de cenários de carga

Cenário	Usuários	Tempo Mín. (Ms)	Tempo Máx. (Ms)	Tempo Médio (Ms)	Req. Aceitas	Req. Falhas
1	100	2	33	3	100	0
2	500	2	39	3	500	0
3	1000	2	48	3	1000	0

Fonte: elaborado pelos autores.

4.4.2 Análise dos resultados

Os resultados demonstram que a API apresentou comportamento estável em todos os cenários avaliados. Mesmo com a simulação de 1000 usuários simultâneos, todas as requisições foram processadas com sucesso, não sendo registrada nenhuma falha durante os testes.

Observa-se ainda que o tempo médio de resposta permaneceu constante em aproximadamente 3 ms, enquanto o maior tempo registrado foi de 48 ms, valor significativamente inferior ao requisito não funcional de desempenho definido para o sistema.

À medida que a carga aumentou, houve apenas um pequeno crescimento no tempo máximo de resposta, passando de 33 ms para 48 ms, comportamento esperado em aplicações submetidas a maior concorrência e que não comprometeu o desempenho da aplicação.

4.4.3 Conclusão dos testes

Os testes realizados demonstraram que a arquitetura adotada foi capaz de suportar os cenários avaliados, mantendo 100% de sucesso nas requisições e tempos de resposta reduzidos mesmo sob elevada concorrência.

Os resultados indicam que o sistema atende ao requisito de desempenho estabelecido, apresentando boa capacidade de processamento e estabilidade para os níveis de carga simulados. Além disso, os testes evidenciam que a utilização do Redis como mecanismo de cache contribuiu para o baixo tempo de resposta observado durante as avaliações.

5 CONCLUSÃO

Este trabalho apresentou o desenvolvimento do KeyControl, um sistema voltado ao gerenciamento de salas, chaves e reservas em instituições de ensino, com o objetivo de modernizar e simplificar os processos de controle desses recursos. A solução busca reduzir a burocracia presente nos métodos tradicionais de gerenciamento, proporcionando maior agilidade na realização de reservas, retirada de chaves e acompanhamento da disponibilidade dos ambientes.

O sistema atendeu ao objetivo geral proposto, que consistiu em desenvolver um software capaz de centralizar o gerenciamento de salas, chaves, responsáveis, reservas e retiradas, oferecendo mecanismos para evitar conflitos de agendamento, controlar a disponibilidade das chaves e manter um histórico das operações realizadas. Além disso, a utilização de cache para otimização de consultas e de um mecanismo de busca especializada contribuiu para melhorar o desempenho e a experiência de uso do sistema.

A adoção de uma arquitetura em camadas, seguindo o modelo cliente-servidor, aliada ao uso de tecnologias como Python, Flask, React, PostgreSQL, Redis e Apache Solr, possibilitou o desenvolvimento de uma aplicação organizada, escalável e de fácil manutenção, seguindo boas práticas de Engenharia de Software e separação de responsabilidades.

Do ponto de vista acadêmico, o trabalho permitiu aplicar, de forma integrada, conhecimentos relacionados ao desenvolvimento de software, banco de dados, arquitetura de sistemas, APIs REST e engenharia de software, demonstrando a importância da utilização de tecnologias modernas para solucionar problemas presentes no cotidiano das instituições de ensino. Sob a perspectiva prática, a solução contribui para tornar o gerenciamento dos espaços físicos mais eficiente, reduzindo o tempo gasto em processos administrativos e minimizando erros decorrentes do controle manual.

Como trabalhos futuros, propõe-se a implementação de um sistema de notificações por e-mail para informar os responsáveis sobre confirmações de reservas, atrasos na devolução de chaves e cancelamentos. Também pretende-se desenvolver um módulo de autenticação integrado ao SUAP, disponibilizar dashboards com indicadores de utilização das salas e ampliar as opções de relatórios gerenciais, oferecendo aos administradores uma visão mais detalhada sobre a ocupação dos ambientes e a utilização dos recursos da instituição.

REFERÊNCIAS

CHIAVENATO, Idalberto. *Introdução à teoria geral da administração*. 7. ed. rev. atual. Rio de Janeiro: Elsevier, 2003.

LAUDON, Kenneth C.; LAUDON, Jane P. *Sistemas de informação gerenciais*. 11. ed. São Paulo: Pearson Education do Brasil, 2014.

MAXIMIANO, Antonio Cesar Amaru. *Introdução à administração*. 5. ed. rev. e ampl. São Paulo: Atlas, 2000.

NIELSEN, Jakob. *Usability engineering*. Boston: Academic Press, 1993.

SOMMERVILLE, Ian. *Engenharia de Software*. 9. ed. São Paulo: Pearson Prentice Hall, 2011.

STAIR, Ralph M.; REYNOLDS, George W. *Princípios de sistemas de informação: uma abordagem gerencial*. Tradução da 6. ed. norte-americana. São Paulo: Pioneira Thomson Learning, 2006.