INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SUL-RIO-GRANDENSE – CÂMPUS PELOTAS - VISCONDE DA GRAÇA
CURSO TÉCNICO EM DESENVOLVIMENTO DE SISTEMAS

CHAVE MESTRA: UM SISTEMA PARA CONTROLE DO USO DAS CHAVES DOS AMBIENTES FÍSICOS DO CAVG

Leonardo Oliveira Garcia
Thiago Pucinelli Aires da Silva

Pelotas, dezembro de 2025.

Leonardo Oliveira Garcia
Thiago Pucinelli Aires da Silva

CHAVE MESTRA: UM SISTEMA PARA CONTROLE DO USO DAS CHAVES DOS AMBIENTES FÍSICOS DO CAVG

Projeto de desenvolvimento de sistema apresentado como requisito para obtenção de diploma no curso Técnico em Desenvolvimento de Sistemas, do Instituto Federal Sul-Rio-Grandense – Câmpus Pelotas - Visconde da Graça.

Orientador: Prof. Dr. João Ladislau Barbará Lopes.

Pelotas, dezembro de 2025.

RESUMO

Este trabalho de conclusão de curso tem como tema central o desenvolvimento de um sistema denominado Chave Mestra, uma solução digital voltada ao controle do uso das chaves dos ambientes físicos do Câmpus Pelotas-Visconde da Graça, do Instituto Federal Sul-rio-grandense. Observando os desafios enfrentados na gestão das chaves, foi constatado que o processo manual apresentava limitações relacionadas à confiabilidade, rastreabilidade, agilidade, falhas de registro e dificuldades de auditoria. Assim, o projeto tem como objetivo geral informatizar o gerenciamento de chaves, oferecendo maior organização, segurança e transparência às rotinas administrativas da instituição. Para tanto, foram definidos objetivos específicos que abrangem: o desenvolvimento de uma interface responsiva e intuitiva; a implementação de funcionalidades de cadastro, reserva, empréstimo e devolução de chaves; o registro de logs de todas as ações realizadas; o uso de autenticação e criptografia de senhas para reforçar a segurança dos dados; possibilidade de acesso para usuários internos e externos, permitindo a consulta de disponibilidade e o acompanhamento de históricos de uso. A metodologia adotada foi a engenharia de software com abordagem ágil, especificamente o framework Scrum, que se destacou por sua flexibilidade e pelo desenvolvimento iterativo e incremental. O processo incluiu levantamento de requisitos funcionais e não funcionais por meio de entrevistas e questionários aplicados a funcionários e alunos. Com base nos requisitos levantados, foram modelados os casos de uso, os fluxos do sistema e o banco de dados. A implementação utilizou tecnologias consolidadas e compatíveis com as competências adquiridas ao longo do curso: HTML5, CSS e Tailwind CSS para a camada de apresentação, JavaScript para interatividade, PHP para o back-end e MySQL como sistema gerenciador de banco de dados. Essa arquitetura buscou garantir escalabilidade, desempenho e segurança, com suporte a múltiplos dispositivos e navegadores. O sistema desenvolvido contempla funcionalidades como: autenticação de usuários, cadastro e gerenciamento de chaves, registro de empréstimos e devoluções com data e hora, reservas antecipadas, relatórios de utilização e controle de atrasos, além da geração de logs para fins de auditoria. A interface responsiva assegura usabilidade tanto em computadores, quanto em dispositivos móveis. Os resultados do projeto demonstram que a informatização possibilita ganhos em organização e segurança, podendo reduzir falhas no gerenciamento das chaves dos ambientes físicos, bem como possibilitar maior praticidade com as reservas on-line.

Palavras chave - Controle de chaves; Segurança da informação; Gestão acadêmica.

LISTA DE ILUSTRAÇÕES

Quadro 1 - Cronograma TCC – Etapas por Mês 10
Quadro 2 - Levantamento de requisitos 12
Quadro 3 - Levantamento de requisitos não funcionais 14

LISTA DE FIGURAS

Figura 1 - Modelo de caso de uso 15
Figura 2 - Modelagem conceitual do banco de dados 16
Figura 3 - Modelagem conceitual do banco de dados 21
Figura 4 - Área de Login 25
Figura 5 - Tela inicial para Gerente 26
Figura 6 - Tela Inicial do usuário comum 27
Figura 7 - Tela inicial para Administrador 28
Figura 8 - Tela de cadastro de chaves 29
Figura 9 -Tela de cadastro de usuário 30
Figura 10 - Tela de edição de perfil do usuário 32
Figura 11 - Tela de redefinição de senha 32
Figura 12 - Ações: Promover usuário e excluir chave 34
Figura 13 - Histórico completo de empréstimos e gráficos 34
Figura 14 - Questões aplicadas a Gestão 38
Figura 15 - Questões aplicadas a Gestão 39
Figura 17 - Questões aplicadas a Gestão 41
Figura 18 - Questões aplicadas a Gestão 42
Figura 19 - Questões aplicadas aos Alunos e Servidores 44
Figura 20 - Questões aplicadas aos Alunos e Servidores 45
Figura 21 - Resultado das questões aplicadas aos funcionários e servidores 46
Figura 22 - Resultado das questões aplicadas aos funcionários e servidores 47
Figura 23 - Resultado das questões aplicadas aos funcionários e servidores 48
Figura 24 - Resultado das questões aplicadas aos funcionários e servidores 49
Figura 25 - Resultado das questões aplicadas aos funcionários e servidores 50
Figura 26 - Resultado das questões aplicadas aos funcionários e servidores 51
Figura 27 - Resultado das questões aplicadas aos funcionários e servidores 52

SUMÁRIO

1 INTRODUÇÃO **6**
1.1 Tema 6
1.2 Motivações 7
1.3 Objetivos 7
1.3.1 Objetivo Geral 7
1.3.2 Objetivos específicos 8
2 METODOLOGIA **9**
2.1 Modelo de desenvolvimento 9
2.2 CRONOGRAMA **10**
2.3 Levantamento de requisitos 11
2.5 Requisitos Funcionais 12
Fonte: autoria própria 12
2.6 Requisitos não funcionais 13
3 MODELAGEM **14**
3.1 Modelo de caso de uso **14**
3.2 Modelagem conceitual do banco de dados 16
3.3 Descrição do Modelo Conceitual do Banco de Dados 17
3.4 Relacionamentos (com cardinalidade) 19
3.5 Modelo lógico do banco de dados 20
3.6. Considerações sobre o Modelo 22
4. TECNOLOGIAS UTILIZADAS **23**
5. DESCRIÇÃO DO SISTEMA **25**
5.1 Acesso ao Sistema (Login) 25
5.1.1 Tela inicial para Gerente 26
5.1.2 Tela inicial para usuário comum 27
5.1.3 Tela inicial para Administrador 28
5.2 Cadastro de Chaves 29
5.3 Cadastro de Usuários 30
5.4 Registro de Empréstimo e Devolução 31
5.5 Reserva de Chave 31
5.6 Edição de Perfil do Usuário 32
5.7 Área do Gerente 33
6. CONSIDERAÇÕES FINAIS **35**
REFERÊNCIAS **36**
APÊNDICE 1: QUESTIONÁRIO APLICADO AOS FUNCIONÁRIOS E SERVIDORES **37**
APÊNDICE 2: QUESTIONÁRIO APLICADO AOS ALUNOS E FUNCIONÁRIOS **43**
APÊNDICE 3: RESULTADO DO QUESTIONÁRIO APLICADO PARA LEVANTAMENTO DE REQUISITOS DO SISTEMA DE GESTÃO DE CHAVES (FUNCIONÁRIOS E SERVIDORES DA GESTÃO ACADÊMICA) **46**
APÊNDICE 4 - INSTRUÇÕES SQL PARA CRIAÇÃO DA BASE DE DADOS **52**

1 INTRODUÇÃO

A gestão e o controle de chaves físicas são processos essenciais para garantir a segurança e a organização em instituições acadêmicas e administrativas. No Instituto Federal de Educação, Ciência e Tecnologia Sul Rio Grandense Câmpus Pelotas - Visconde da Graça (IFSUL CAVG), a administração desse controle ainda é realizada de forma manual, utilizando papel e caneta. Embora esse método seja funcional, ele apresenta diversas limitações que comprometem a eficiência e a confiabilidade do processo.

A ausência de um sistema digitalizado pode resultar em falhas no registro de entrada e saída de chaves, dificultando a rastreabilidade e aumentando o risco de extravios. Além disso, a dependência de registros físicos pode gerar inconsistências nos dados, dificultando a auditoria e a recuperação de informações quando necessário. Diante desse cenário, faz-se necessária a implementação de um sistema informatizado que possibilita um gerenciamento mais seguro e eficiente.

Para solucionar esse problema, este projeto propõe o desenvolvimento de um site gerenciador de chaves, que visa otimizar o controle e a rastreabilidade destas dentro da instituição acadêmica. A ferramenta permitirá o cadastro de chaves, o registro de empréstimos, devoluções e agendamentos.

1.1 Tema

Este trabalho tem como tema o desenvolvimento de um software para o controle do uso das chaves dos ambientes físicos do Instituto Federal Sul-Rio-Grandense – Câmpus Pelotas Visconde da Graça (IFSul CAVG). Através de observação constatamos que o gerenciamento manual, feito atualmente por meio de registros em papel, apresenta limitações quanto à segurança, confiabilidade e agilidade do processo. Assim, busca-se propor e implementar uma solução digital, que permita o cadastro, a reserva, o empréstimo e a devolução de chaves de forma mais organizada e segura, contribuindo para a otimização da gestão e para a melhoria da experiência de alunos, servidores e funcionários da instituição.

1.2 Motivações

A motivação para o projeto veio do desejo de contribuir com a instituição e realizar algo socialmente relevante, beneficiando alunos e servidores do campus ao proporcionar mais segurança e praticidade. Observando os desafios enfrentados na gestão das chaves, identificamos que o controle era realizado de forma manual, utilizando papel e caneta, o que frequentemente resultava em dificuldades na organização, falhas na segurança e erros no registro de entrada e saída das chaves.

A ausência de um sistema mais eficiente despertou em nós a motivação para desenvolver uma solução digital que tornasse esse processo mais prático e seguro, baseado no princípio da eficiência na administração pública (Brasil, 1998). Com o software de gerenciamento de chaves, denominado Chave Mestra, buscamos não apenas otimizar a gestão das chaves, reduzindo equívocos e melhorando os registros, mas também proporcionar uma melhor visualização da disponibilidade, garantindo maior controle e eficiência no uso desses recursos.

1.3 Objetivos

Os objetivos do projeto têm como finalidade orientar e definir, com clareza, o propósito do trabalho e os caminhos a serem seguidos para alcançar os resultados esperados. Servindo como guia para a execução de todas as etapas do desenvolvimento, assegurando que o produto final esteja alinhado com a proposta inicial e com a metodologia adotada.

Neste contexto é apresentado um objetivo geral, que expressa a meta principal do projeto, e objetivos específicos, que detalham as ações necessárias para que essa meta seja atingida de forma estruturada e eficiente.

1.3.1 Objetivo Geral

Desenvolver um sistema web com o objetivo de informatizar o processo de reserva, controle de retirada e devolução de chaves físicas no Instituto Federal Sul-rio-grandense - Câmpus Pelotas Visconde da Graça (IFSul CAVG). A proposta é substituir o método atual, feito manualmente, por uma solução digital mais prática, segura e confiável. O sistema permitirá o registro detalhado das movimentações de chaves, facilitando o acompanhamento, a rastreabilidade e a organização das informações, além de reduzir erros humanos e perdas de dados.

1.3.2 Objetivos específicos

Levando em conta o objetivo geral deste trabalho, os objetivos específicos foram definidos com a finalidade de apresentar as etapas necessárias para a concretização da proposta. Cada um deles representa uma ação prática e direcionada que contribui para o alcance do resultado final desejado.

- Desenvolver uma interface intuitiva e responsiva: Criar uma interface de usuário amigável, acessível e adaptada a diferentes dispositivos, garantindo que os usuários possam navegar com facilidade e utilizar todas as funcionalidades do sistema, independentemente do tamanho da tela. O layout responsivo busca oferecer uma experiência consistente e eficiente tanto em dispositivos móveis quanto em desktops.
- Otimizar o controle e a gestão das chaves: Desenvolver ferramentas que possibilitem o cadastro detalhado das chaves do campus, bem como o monitoramento, agendamento, retirada e devolução, organizando essas atividades de forma eficiente, reduzindo erros e assegurando maior controle e organização.
- Reforçar a segurança e o controle do sistema: Implementar autenticação de administradores por senha, registro das ações executadas e medidas de proteção de dados, assegurando integridade, confiabilidade e rastreabilidade das informações, em conformidade com as melhores práticas.
- Ampliar o acesso e a transparência para os usuários: Implementar funcionalidades que possibilitem o cadastro online de alunos e servidores, a consulta remota à disponibilidade das chaves, a realização de agendamentos e o acesso a um histórico pessoal de retiradas, por meio de uma área acessível e otimizada para dispositivos móveis.

2 METODOLOGIA

Buscando criar um sistema que facilitasse a organização das chaves do CAVG, a primeira etapa da metodologia se deu através da utilização do pensamento computacional, que, de acordo com Brackmann (2017, p. 18), é *"*um processo de resolução de problemas que envolve expressar soluções de forma que um computador – humano ou máquina – possa efetivamente realizá-las*"*. Dessa forma, decompomos o problema e, por meio do pensamento algorítmico, criamos os seguintes requisitos que o software deveria ter, assim formando os objetivos gerais e específicos supracitados.

Para garantir um desenvolvimento eficiente e alinhado às necessidades da instituição, o projeto adotou a metodologia Ágil, utilizando o framework Scrum. Esse modelo foi escolhido devido à sua flexibilidade e ao foco na colaboração contínua, permitindo ajustes durante o desenvolvimento do sistema conforme as necessidades identificadas (Schwaber; Sutherland, 2020) .

2.1 Modelo de desenvolvimento

O Scrum é um framework ágil que divide o desenvolvimento em ciclos chamados sprints, onde cada sprint possui um conjunto de tarefas bem definidas a serem concluídas dentro de um período estabelecido. Essa abordagem possibilita a entrega incremental do sistema, permitindo que funcionalidades sejam desenvolvidas, testadas e validadas continuamente, garantindo um maior alinhamento com as expectativas dos usuários.

2.2 CRONOGRAMA

Para orientar a execução das fases do projeto dentro do prazo acadêmico, foi elaborado um cronograma inicial detalhado. O Quadro 1, apresentado a seguir, ilustra a distribuição planejada das principais etapas do desenvolvimento ao longo dos meses, desde a concepção inicial do tema em abril até a defesa do trabalho em dezembro.

Este planejamento serviu como um guia norteador para o gerenciamento do tempo. Contudo, é importante ressaltar que ele não foi seguido à risca. Visto que o projeto adotou uma abordagem ágil (Scrum), o cronograma sofreu adaptações e ajustes conforme os desafios e necessidades que surgiram durante os ciclos de desenvolvimento, permitindo a flexibilidade necessária para a conclusão do sistema.

Quadro 1 - Cronograma TCC – Etapas por Mês

Datas	**Etapas**
Abril	Escolha do tema Modelagem conceitual Início da documentação
Maio	Continuação da documentação Início da prototipação
Junho	Prototipação Início da implementação
Julho	Prototipação Implementação
Agosto	Implementação Início dos testes
Setembro	Testes Início das correções de erros
Outubro	Correções de erros
Novembro	Correções de erros (etapa final)
Dezembro	Defesa do TCC

2.3 Levantamento de requisitos

O levantamento de requisitos foi realizado por meio de observação direta, e da aplicação de um questionário aos funcionários responsáveis pela guarda e controle de chaves do sistema manual atual.

Antes da aplicação dos questionários e entrevistas, os participantes foram informados sobre os objetivos da pesquisa, o caráter acadêmico do trabalho e o uso exclusivo dos dados coletados para fins de desenvolvimento do sistema e elaboração deste projeto.

As entrevistas foram realizadas presencialmente no mês de março de 2025. Os questionários foram aplicados por meio do Google Forms e permaneceram disponíveis entre os dias 4 e 20 de março.

Nesse mesmo período também foi aplicado um questionário aos alunos, com perguntas voltadas às dificuldades enfrentadas durante o processo de entrada nas salas de aula e laboratórios, especialmente relacionadas à demora na entrega ou retirada de chaves. O objetivo foi obter uma visão mais ampla dos impactos do atual sistema manual.

Esse processo permitiu identificar os principais desafios enfrentados no controle de chaves e definir as funcionalidades essenciais para a solução proposta. Dessa forma, o sistema foi planejado com base em necessidades reais, visando proporcionar maior eficiência e segurança na gestão das chaves.

Devido à baixa adesão nas respostas do corpo discente, a análise de dados apresentada no Apêndice 3 refere-se exclusivamente ao levantamento feito com servidores e funcionários da gestão. Os modelos completos dos questionários utilizados encontram-se nos Apêndices 1 e 2 deste trabalho, incluindo os links para os formulários originais, disponíveis para consulta.

2.5 Requisitos Funcionais

O Quadro 2 lista os principais requisitos funcionais (RF) implementados.

Quadro 2 - Levantamento de requisitos

Requisito	Tipo	Descrição
REF01	Autenticar Usuários	O sistema deve conter login para a realização do acesso.
REF02	Consultar Disponibilidade	O sistema deve permitir a busca e listagem de chaves (por nome, usuário ou data), exibindo visualmente o status atual de cada item (disponível, emprestada, reservada).
REF03	Gerenciar Reservas	O sistema deve permitir que usuários realizem o agendamento de chaves disponíveis, bem como o cancelamento de reservas efetuadas.
REF04	Visualizar Registros	O sistema deve permitir a visualização do registro de uso das chaves. (ex: Informações completas sobre retiradas e devoluções)
REF05	Registrar Empréstimo e Devolução de Chaves	O sistema deve permitir que o gerente ou administrador, registre a retirada e a devolução de chaves, contendo data e hora.
REF06	Cadastrar Usuários	O sistema deve permitir o cadastro de usuários com atributos: nome, cpf, senha, email e categoria.
REF07	Editar Perfil	O sistema deve permitir que o usuário altere suas informações pessoais. (Ex: Nome, email e senha)
REF08	Fornecer Histórico de Uso	O sistema deve exibir o histórico pessoal de uso das chaves para cada usuário.
REF09	Gerenciar Chaves	O sistema deve permitir o cadastro, a edição e a exclusão de chaves, gerenciando atributos como nome, número de identificação, descrição e quantidade.
REF10	Gerenciar Permissões	O sistema deve permitir que o gerente gerencie as permissões dos usuários. (O que o usuário pode ou não fazer)

Fonte: autoria própria

2.6 Requisitos não funcionais

O Quadro 3 lista os principais requisitos não funcionais (RNF) definidos para o sistema.

Quadro 3 - Levantamento de requisitos não funcionais

Requisito	Tipo	Descrição
RNF01	Desenvolvimento	Deve ter estrutura modular, com separação entre front-end, back-end e banco de dados.
RNF02	Desenvolvimento	O sistema deve ser desenvolvido utilizando as seguintes tecnologias: Back-end: PHP, Banco de dados: MySQL Front-end: HTML5, CSS, JavaScript Framework: Tailwind e ser compatível com os principais navegadores.
RNF03	Desempenho	O sistema deve responder às ações em até 2 segundos
RNF04	Escalabilidade	Deve suportar aumento de dados e usuários sem perda de desempenho.
RNF05	Disponibilidade	O sistema deve ter alta disponibilidade (mínimo de 99% online).
RNF06	Confiabilidade	O sistema deve garantir consistência e disponibilidade dos dados.
RNF07	Auditabilidade	O sistema deve registrar todas as ações realizadas no sistema.
RNF08	Segurança	Os dados devem estar protegidos contra acesso não autorizado.
RNF09	Segurança	O sistema deve permitir criação de backups periódicos.
RNF10	Segurança	O sistema deve restringir o acesso a funcionalidades e informações com base no perfil do usuário, garantindo que cada nível tenha acesso apenas às ações autorizadas.
RNF11	Segurança	Senhas devem ser criptografadas no banco de dados.
RNF12	Acessibilidade	O sistema deve seguir boas práticas de acessibilidade.
RNF13	Usabilidade	A interface deve ser intuitiva e fácil de usar, operando em celulares e tablets com layout adaptado.
RNF14	Usabilidade	As notificações devem ser exibidas ao usuário de forma clara e visível na interface.

Fonte: autoria própria.

3 MODELAGEM

3.1 Modelo de caso de uso

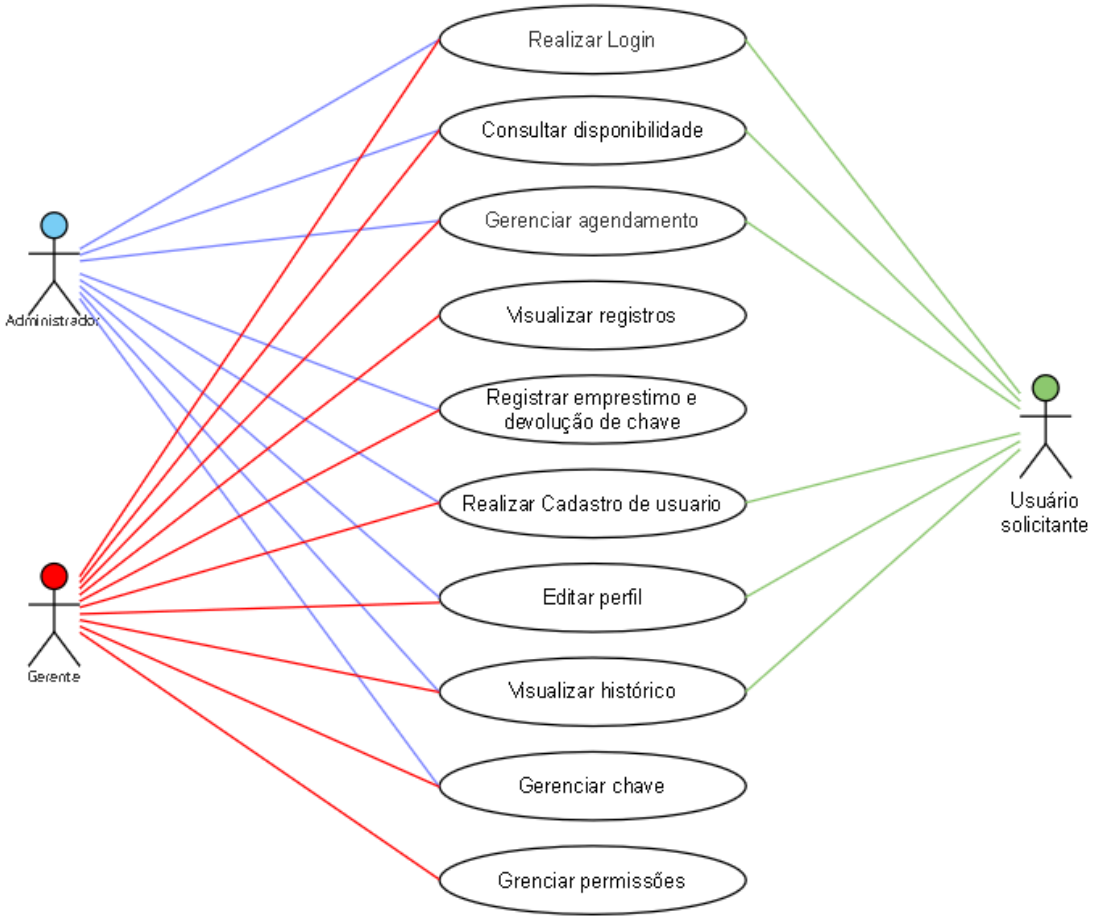
Buscando ajudar a identificar e documentar as funcionalidades oferecidas e como o usuário interage com o sistema, foi usado durante o desenvolvimento do software a ferramenta de Modelo de caso de uso, vista nas aulas de Análise e projetos de sistema do curso Técnico de desenvolvimento de sistemas..

Conforme a Figura 1, é possível visualizar três atores: Gerente, Administradores e Usuário comum (Externo). Sendo assim:

- **Gerente:** Nível máximo de acesso. Responsável pela configuração e auditoria do sistema.
- **Administrador:** Nível operacional de acesso. Responsável pela execução dos processos diários.
- **Usuário comum:** Nível de acesso externo e limitado. Pode ser aluno, servidor (sem acesso administrativo).

Figura 1 - Modelo de caso de uso.

Fonte: Própria (2025)

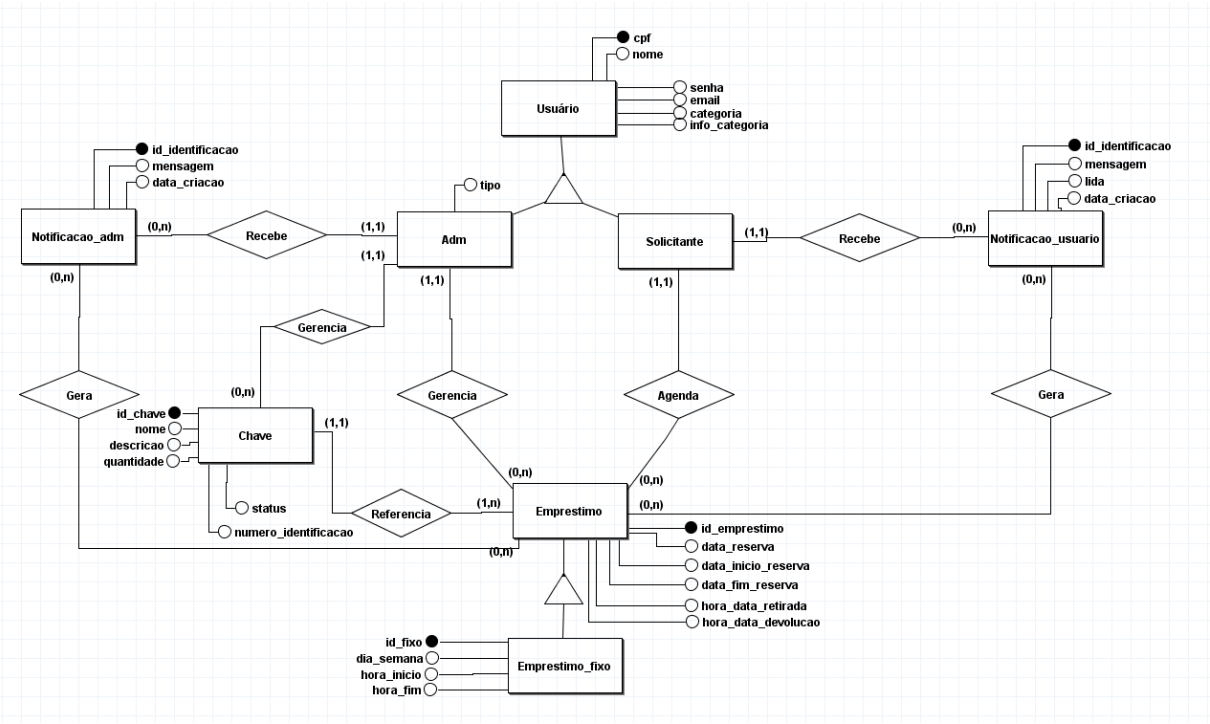


Fonte - Própria (2025)

3.2 Modelagem conceitual do banco de dados

Buscando uma forma visual de organizar as ideias, foi usado um mapa conceitual. Nele, é possível ver os conceitos e como eles se relacionam, o que facilita a compreensão do conteúdo. No caso do sistema desenvolvido, o modelo conceitual do banco de dados, com suas entidades e conexões, está mostrado na Figura 2.

Figura 2 - Modelagem conceitual do banco de dados.



Fonte - Própria (2025)

3.3 Descrição do Modelo Conceitual do Banco de Dados

A estrutura do banco de dados foi modelada para representar as entidades e seus relacionamentos com clareza e precisão. Abaixo estão descritas as principais entidades e seus atributos:

Este grupo define quem interage com o sistema e quais são seus papéis.

- Usuário
  - Propósito: Armazenar os dados de login e informações básicas de todas as pessoas cadastradas no sistema, sejam elas administradores ou solicitantes.
  - Atributos Principais:
    - cpf: Identificador único (chave primária) do usuário.
    - nome: Nome completo.
    - senha: Senha para autenticação.
    - email: E-mail para realização de login, contato e notificações.
    - categoria: Define o tipo de usuário (ex: "Aluno", "Professor",).
    - info_categoria: Informação adicional ligada à categoria.

- Adm
  - Propósito: Define quais usuários da tabela usuario têm privilégios administrativos.
  - Atributos Principais:
    - cpf: Chave estrangeira que se liga ao usuario.cpf.
    - tipo: Define o nível de administração (ex: "Administrador" ou "Gerente" dependendo de seu tipo o usuário possuí diferentes permissões de uso no sistema.).

- solicitante
  - Propósito: Define quais usuários da tabela usuario estão autorizados a solicitar chaves.
  - Atributos Principais:
    - cpf: Chave estrangeira que se liga ao usuario.cpf.

Entidade de Item (O Objeto Gerenciado) este é o item que o sistema controla.

- chave
  - Propósito: A entidade central do sistema. Representa o item físico (a chave) que será emprestado.
  - Atributos Principais:
    - id_chave: Identificador numérico único da chave.
    - nome: Nome descritivo (ex: "Sala 101", "Laboratório de Química").
    - numero_identificacao: Um código único de identificação.
    - descricao: Detalhes adicionais sobre a chave ou o local.
    - quantidade: Quantidade total de cópias desta chave que o sistema possui.
    - status: Status atual (ex: "Disponível", "Emprestada", "Manutenção").
    - cpf_adm: O CPF do administrador responsável por esta chave.

Entidades de Transação (O Processo de Empréstimo) este grupo define as ações que ligam os Usuários às Chaves.

- emprestimo
  - Propósito: Esta é a tabela principal de transações. Registra empréstimos pontuais ou agendados (reservas).
  - Atributos Principais:
    - id_emprestimo: Identificador único da transação.
    - id_chave: Chave estrangeira para a tabela chave.
    - cpf_solicitante: Chave estrangeira para o solicitante (ou usuario) que pegou a chave.
    - cpf_adm: Chave estrangeira para o usuario_adm que autorizou/registrou a retirada.
    - data_inicio_reserva / data_fim_reserva: Período em que a chave foi reservada.
    - hora_data_retirada / hora_data_devolucao: Registros factuais de quando a chave foi fisicamente retirada e devolvida.
    - categoria: Tipo de empréstimo (ex: "Pontual", "Evento").

- emprestimo_fixo
  - Propósito: Tabela especializada para registrar empréstimos recorrentes, como um professor que usa a mesma sala toda segunda-feira de manhã.
  - Atributos Principais:
    - id_fixo: Identificador único do empréstimo fixo.
    - id_chave: Chave estrangeira para a chave.
    - cpf_solicitante: Chave estrangeira para o solicitante.
    - cpf_adm: Chave estrangeira para o administrador.
    - dia_semana: O dia da semana em que ocorre (ex: 1 para Segunda, 2 para Terça).
    - hora_inicio / hora_fim: Horário fixo do empréstimo.

Entidades de Comunicação (Logs e Alertas)
Este grupo gerencia a comunicação com os usuários sobre o status dos empréstimos.

- notificacao_usuario
  - Propósito: Armazenar mensagens enviadas aos solicitantes (ex: "Lembrete de devolução").
  - Atributos Principais:
    - id_notificacao: Identificador único da notificação.
    - cpf_usuario: O destinatário da mensagem.
    - id_emprestimo: (Opcional) O empréstimo ao qual a notificação se refere.
    - mensagem: O texto da notificação.
    - lida: Flag (0 ou 1) para controlar se o usuário já visualizou.

- notificacoes_admin
  - Propósito: Armazenar notificações para os administradores (ex: "Nova solicitação de chave pendente").
  - Atributos Principais:
    - id_notificacao: Identificador único.
    - id_emprestimo: O empréstimo que gerou a notificação.
    - cpf_admin: O administrador que deve receber o alerta.
    - mensagem: O texto da notificação.

3.4 Relacionamentos (com cardinalidade)

usuario_adm (1) : (N) chave

- Leitura: Um usuario_adm pode ser responsável por várias chaves.
- Leitura Inversa: Uma chave é gerenciada por um usuario_adm.

chave (1) : (N) emprestimo

- Leitura: Uma chave pode estar em vários emprestimos (ao longo do tempo).
- Leitura Inversa: Um emprestimo específico refere-se a apenas uma chave.

solicitante (1) : (N) emprestimo

- Leitura: Um solicitante pode realizar vários emprestimos.
- Leitura Inversa: Um emprestimo é feito por apenas um solicitante.

usuario_adm (1) : (N) emprestimo

- Leitura: Um usuario_adm pode aprovar/registrar vários emprestimos.
- Leitura Inversa: Um emprestimo é aprovado por um usuario_adm

chave (1) : (N) emprestimo_fixo

- Leitura: Uma chave pode ter vários emprestimos_fixos (ex: um para segunda-feira, outro para terça).
- Leitura Inversa: Um emprestimo_fixo refere-se a apenas uma chave.

solicitante (1) : (N) emprestimo_fixo

- Leitura: Um solicitante pode ter vários emprestimos_fixos.
- Leitura Inversa: Um emprestimo_fixo pertence a apenas um solicitante.

emprestimo (1) : (N) notificacao_usuario

- Leitura: Um emprestimo pode gerar várias notificacoes_usuario.
- Leitura Inversa: Uma notificacao_usuario refere-se a um emprestimo (ou nenhum, pois id_emprestimo é nulo).

usuario (1) : (N) notificacao_usuario

- Leitura: Um usuario pode receber várias notificacoes_usuario.
- Leitura Inversa: Uma notificacao_usuario é destinada a apenas um usuario.

emprestimo (1) : (N) notificacoes_admin

- Leitura: Um emprestimo pode gerar várias notificacoes_admin.
- Leitura Inversa: Uma notificacao_admin refere-se a apenas um emprestimo.

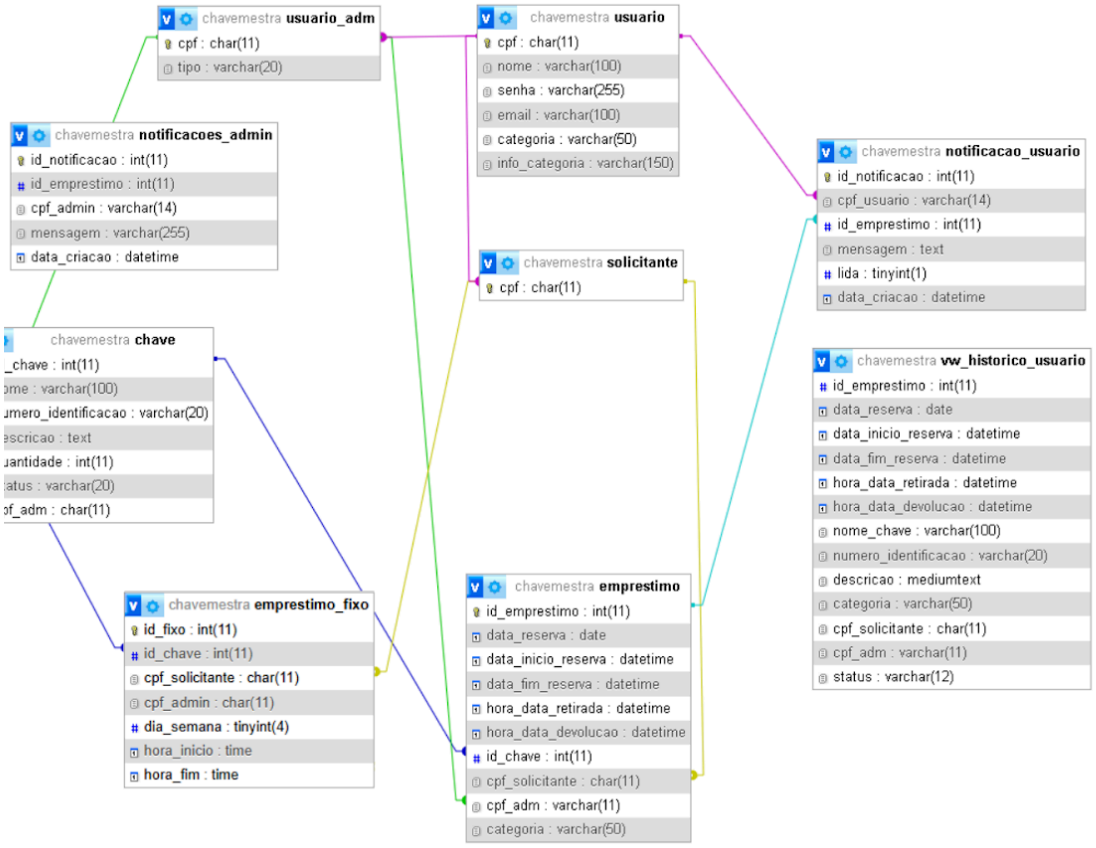
usuario_adm (1) : (N) notificacoes_admin

- Leitura: Um usuario_adm pode receber várias notificacoes_admin.
- Leitura Inversa: Uma notificacao_admin é destinada a apenas um usuario_adm.

3.5 Modelo lógico do banco de dados

Enquanto o modelo conceitual foca no "o que" (quais são as entidades e seus relacionamentos de forma abstrata), o modelo lógico descreve o "como" esses dados serão estruturados.

Figura 3 - Modelagem conceitual do banco de dados.



Fonte - Própria (2025)

3.6. Considerações sobre o Modelo

O fluxo do sistema e a estrutura de dados foram pensados para garantir rastreabilidade, controle de acesso e facilidade no uso por parte dos funcionários e usuários externos. O Gerente é responsável pela gestão do sistema, incluindo o controle de chaves e permissões dos demais usuários. Já os administradores registram as retiradas e devoluções. Usuários que podem ser estudantes, servidores e funcionários que não trabalham no setor, apenas realizam reservas,. O modelo conceitual garante integridade dos dados e espelha os processos realizados no dia a dia da instituição.

4. TECNOLOGIAS UTILIZADAS

Para o desenvolvimento do sistema Chave Mestra, foram selecionadas as tecnologias vistas no curso, que já são consolidadas, de fácil acesso e alinhadas às necessidades do projeto. A escolha buscou equilibrar segurança, desempenho, escalabilidade e simplicidade de manutenção, garantindo que o sistema possa ser utilizado de forma eficiente no contexto da Instituição.

- HTML (Hypertext Markup Language): Linguagem de marcação utilizada para estruturar e exibir o conteúdo das páginas web, como textos, imagens, links e outros elementos. A ferramenta foi usada para organizar informações em um formato compreensível pelos navegadores.

- CSS (Cascading Style Sheets): Responsável por definir o estilo e a aparência das páginas web. Tornou possível personalizar cores, fontes, tamanhos, espaçamentos e criar layouts mais elaborados, atribuindo identidade visual ao conteúdo estruturado pelo HTML.Framework de CSS

- Tailwind CSS**:** Framework de CSS, utilitário que facilitou a criação de uma interface moderna, responsiva e com alto nível de consistência visual. Diferentemente de frameworks baseados em componentes, como o Bootstrap, o Tailwind disponibiliza classes utilitárias pré-definidas que permitem aplicar estilos diretamente no HTML, reduzindo significativamente a necessidade de escrever CSS customizado. Essa abordagem acelerou o desenvolvimento, garantiu padronização entre as páginas e proporcionou maior controle sobre o design, tornando o processo mais flexível e eficiente.

- JavaScript: Linguagem de programação utilizada para tornar as páginas dinâmicas e interativas. Possibilitou ações como validação de formulários, efeitos visuais e interação em tempo real com o usuário, complementando o HTML e o CSS.

- MySQL: Sistema de gerenciamento de banco de dados relacional que organizou e armazenou informações em tabelas, permitindo realizar consultas, inserções e atualizações de dados de forma eficiente. Foi utilizado em conjunto com PHP para compor um sistema dinâmico.

- PHP: Linguagem de programação voltada para o desenvolvimento web no lado do servidor. Permitiu a criação de páginas dinâmicas, o processamento de formulários, a manipulação de bancos de dados e o gerenciamento de informações enviadas pelos usuários, integrando-se a HTML, CSS, JavaScript e MySQL.

  Em conjunto, essas tecnologias proporcionam a base necessária para o desenvolvimento do sistema, garantindo tanto a estrutura e a apresentação visual quanto a interatividade e o processamento de dados.

5. DESCRIÇÃO DO SISTEMA

O sistema "Chave Mestra" foi projetado para organizar e controlar o uso das chaves dos ambientes físicos do CAVG de forma digital, segura e eficiente. Seguindo uma lógica centrada nos diferentes perfis de usuários, nos processos de empréstimo, devolução, reserva e consulta, bem como no registro de todas as movimentações realizadas. Este capítulo mostra…

5.1 Acesso ao Sistema (Login)

O acesso ao sistema, independente do tipo de usuário, é realizado por meio de autenticação com usuário e senha.

Após o usuário preencher os campos ao clicar no botão “Entrar” ele é direcionado a página inicial referente a sua classificação. Além da página também contar com links para as páginas de cadastro e recuperação de senha.

Figura 4 - Área de Login.

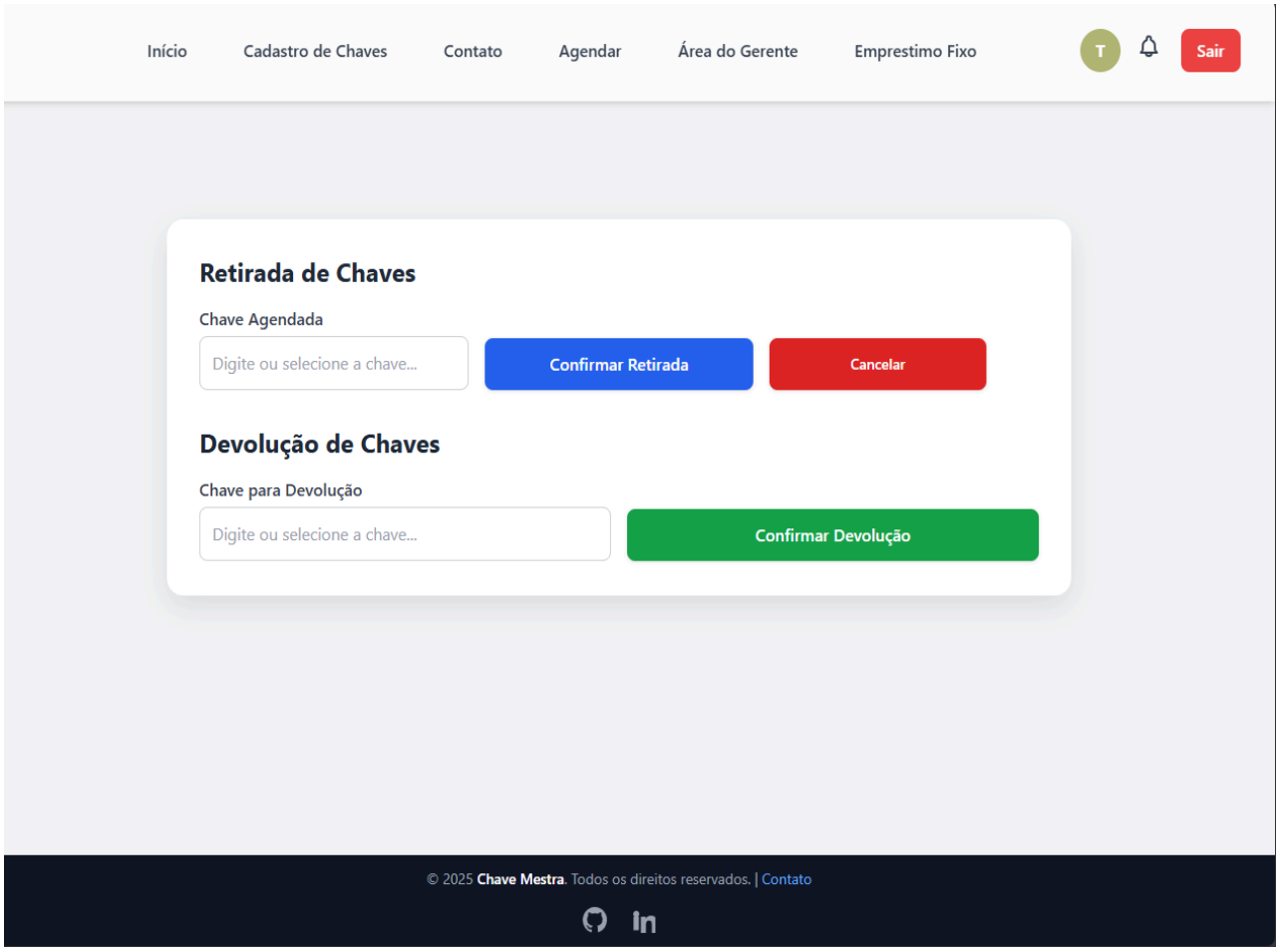
Fonte - Própria (2025)

5.1.1 Tela inicial para Gerente

A tela inicial para o Gerente apresenta as opções de registro de confirmação de retirada e devolução de chaves. além do cabeçalho que apresenta as seguintes opções através dos botões:

- Logo do software - Usado como link para a página inicial ao clicar.
- Início - Redireciona o usuário à página inicial.
- Contato - Direciona o usuário a página de contato.
- Cadastro de Chaves - Direciona o usuário a página de cadastro de chaves.
- Agendar - Direciona o usuário a página de agendamento de chaves.
- Ícone com letra inicial do nome - Direciona o usuário para a página de perfil pessoal do usuário.
- Sair - Desloga o usuário e o redireciona a página de login.

Figura 5 - Tela inicial para Gerente.

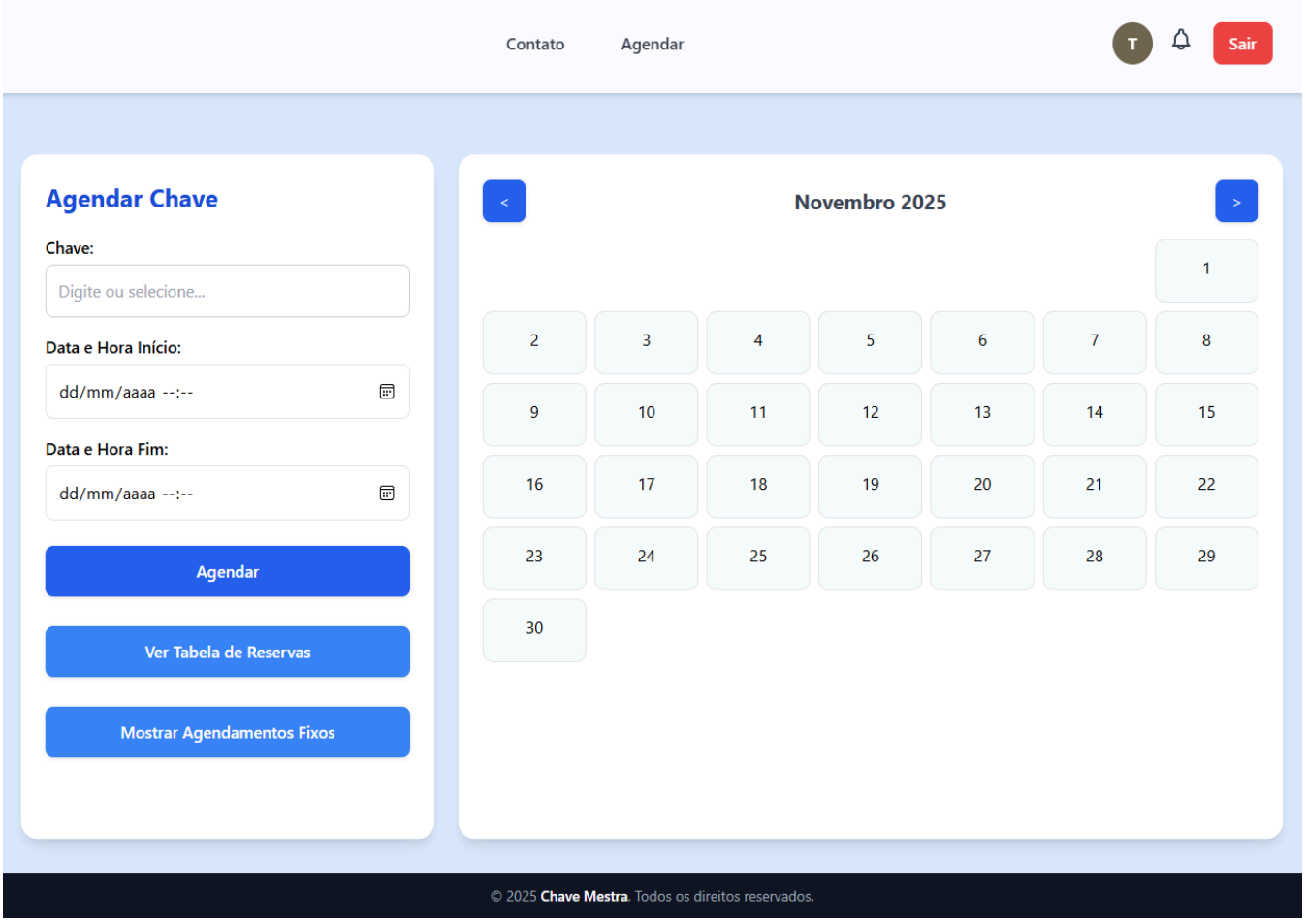


Fonte - Própria (2025)

5.1.2 Tela inicial para usuário comum

A tela inicial para o usuário comum é limitada a suas restrições, por isso neste caso o menu do cabeçalho não conta com os botões “Cadastro de chaves”, “Agendar”, “Área do Gerente” e “Empréstimo Fixo” pois diferente da tela inicial do Gerente e do Administrador, a página inicial desse usuário é a própria página de agendamento.

Figura 6 - Tela Inicial do usuário comum.

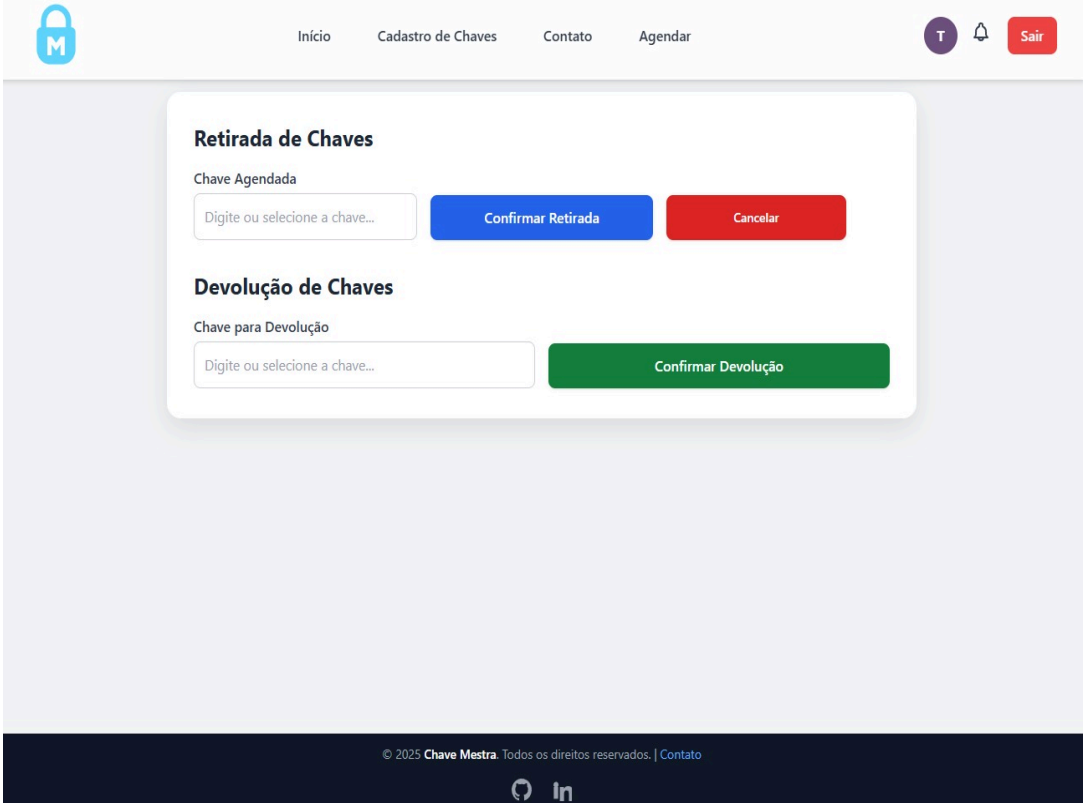


Fonte - Própria (2025)

5.1.3 Tela inicial para Administrador

A tela inicial para o Administrador também é limitada a suas restrições, neste caso o menu do cabeçalho não conta com os botões “Empréstimo Fixo” e “Área do Gerente”.

Figura 7 - Tela inicial para Administrador.

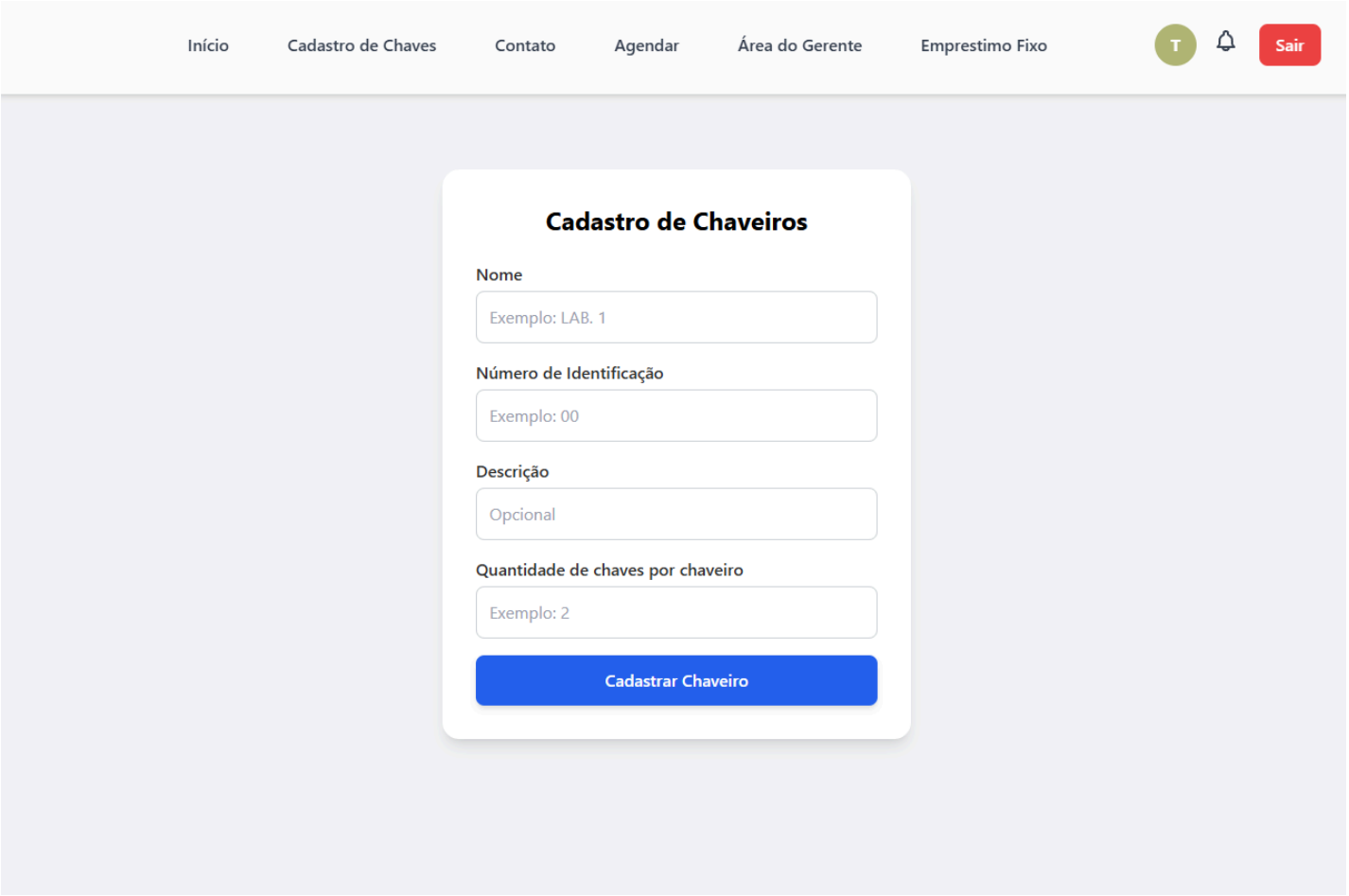


Fonte - Própria (2025)

5.2 Cadastro de Chaves

A funcionalidade de cadastro de chaves está restrita aos Administradores e ao Gerente. Eles devem informar o número da chave, sua identificação (nome), o local correspondente (sala) e, opcionalmente, uma descrição. O sistema verifica se já existe uma chave com o mesmo número para evitar duplicidade. Caso a chave não exista, os dados são gravados no banco de dados com status inicial “disponível”.

Figura 8 - Tela de cadastro de chaves.



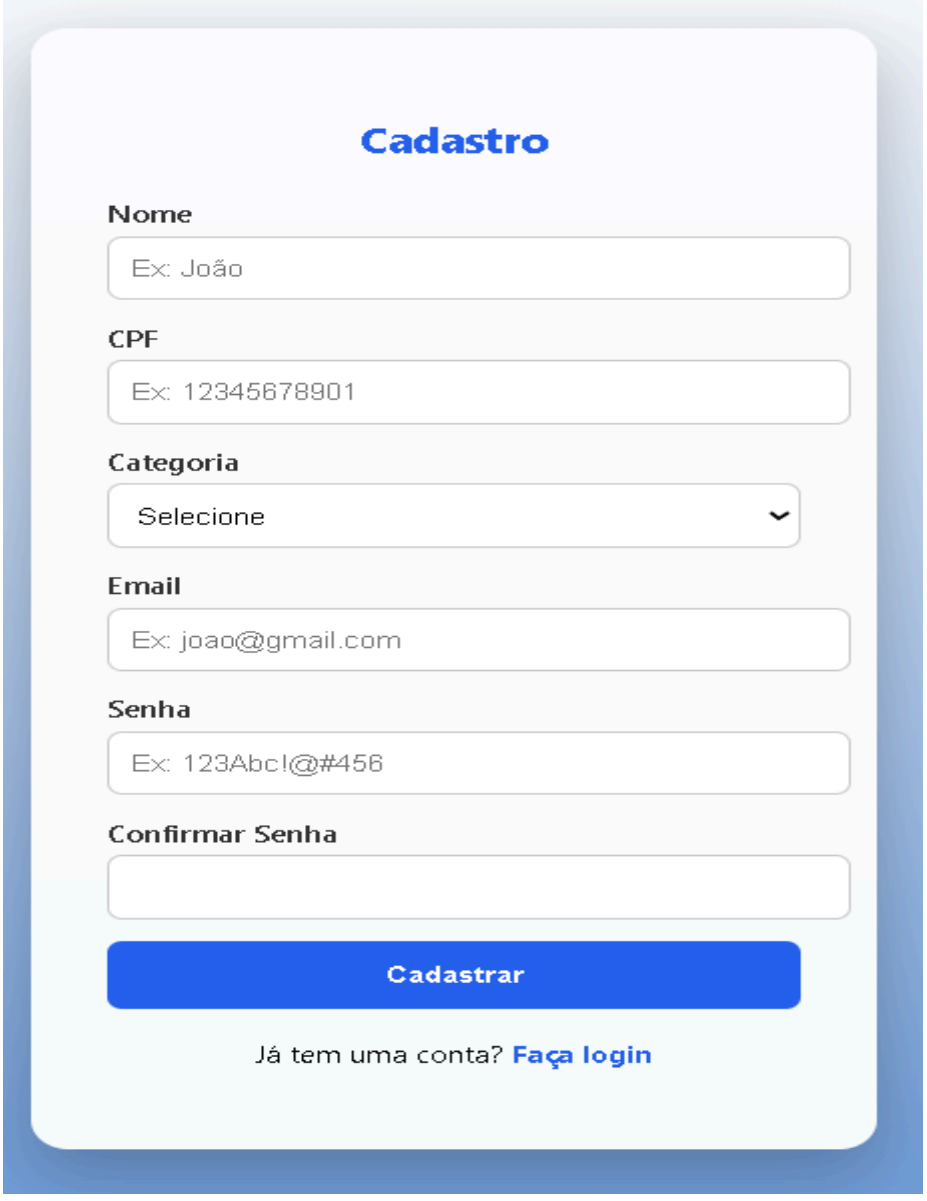
Fonte - Própria (2025)

5.3 Cadastro de Usuários

O cadastro de usuários ocorre de duas formas:

- **Administradores:** são cadastrados pelo Gerente, com permissões definidas conforme o tipo de acesso necessário.
- **Usuários externos (solicitantes):** realizam autocadastro via sistema, informando CPF, nome completo, e-mail e criando uma senha. Após o cadastro, esses usuários podem realizar agendamentos de chave e acompanhar seu histórico de uso.

Figura 9 -Tela de cadastro de usuário



Fonte - Própria (2025)

5.4 Registro de Empréstimo e Devolução

Gerente e Administrador acessam a página inicial de empréstimo de chaves, que pode ser vista na Figura 4.

Nesse processo o funcionário seleciona a chave reservada e registra a retirada, gravando data e hora. O sistema atualiza o status da chave para “emprestada”.

A devolução é realizada pelos funcionários, que acessam a lista de chaves atualmente emprestadas. Ao selecionar a chave correspondente, confirmam sua devolução. O sistema registra data e hora e atualiza o status da chave para “disponível”.

5.5 Reserva de Chave

Todos os usuários podem realizar a reserva de chave, informando o dia e os horários de início e término desejados. O sistema verifica se há conflito com outros empréstimos ou reservas e, se disponível, confirmar a reserva. Durante o período agendado, a chave permanece bloqueada para novos empréstimos, a pagina de reserva de chaves pode ser vista na Figura 5 que também corresponde a tela inicial do usuário.

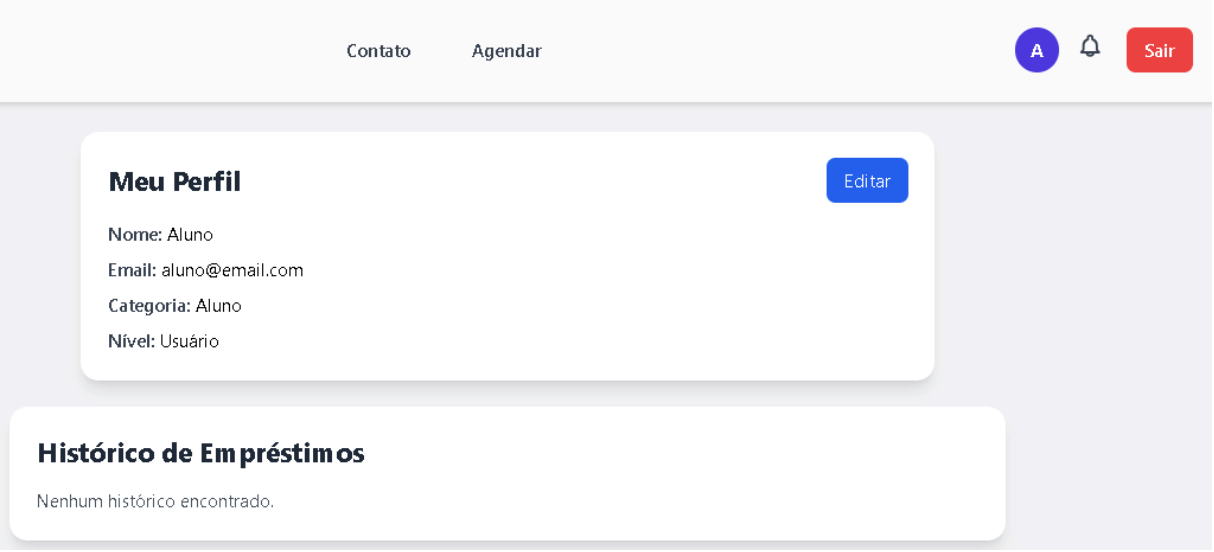
5.6 Edição de Perfil do Usuário

O acesso à funcionalidade de perfil é iniciado quando o usuário clica no ícone de identificação (representado pela sua inicial), localizado no cabeçalho da aplicação. Esta ação redireciona o usuário para a sua página de perfil.

Nesta tela, o usuário pode consultar seu histórico pessoal de atividades e visualizar seus dados cadastrais atuais. Para editar estas informações, o usuário deve acionar o botão "Editar".

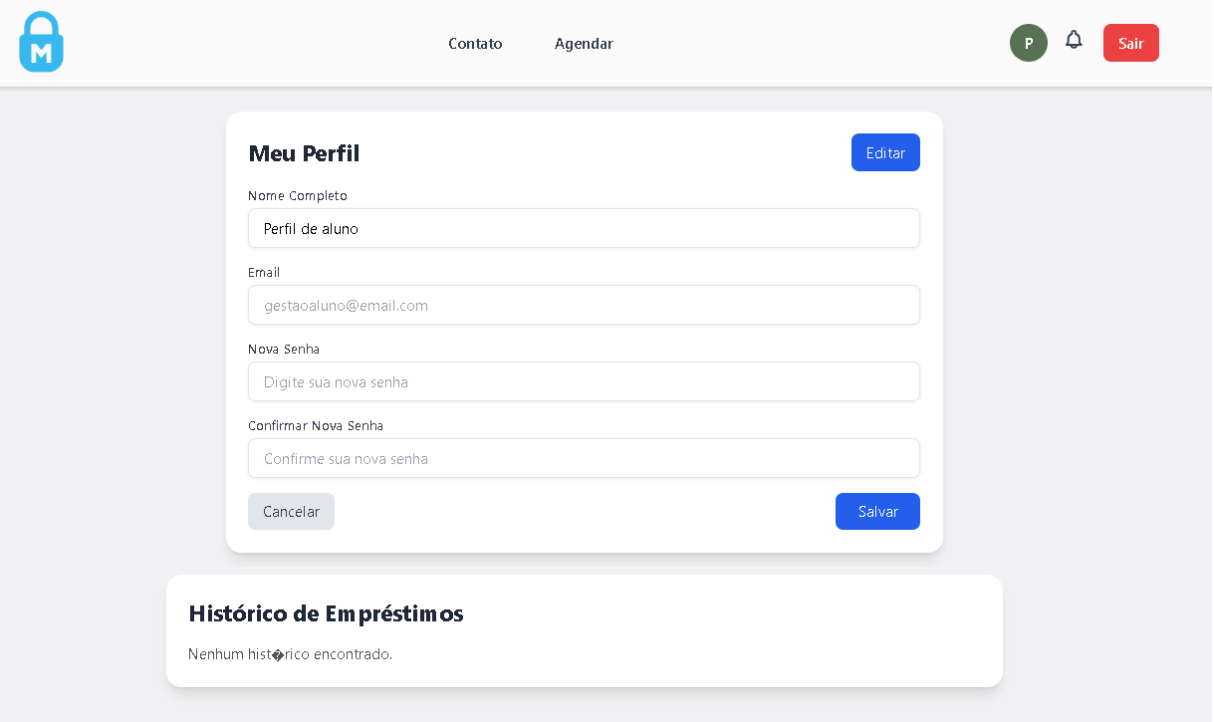
Ao ser clicado, o sistema apresenta uma caixa de validação, solicitando a confirmação do usuário para prosseguir. Após a confirmação ser fornecida, a interface habilita os campos do formulário para a edição dos dados pessoais. Esta seção também disponibiliza a funcionalidade de redefinição de senha.

Figura 10 - Tela de edição de perfil do usuário.



Fonte - Própria (2025)

Figura 11 - Tela de redefinição de senha.



Fonte - Própria (2025)

5.7 Área do Gerente

A Área do Gerente é uma seção de acesso restrito, disponível apenas para usuários autenticados com o perfil de "Gerente". Esta tela centraliza funcionalidades administrativas e ferramentas de análise de dados.

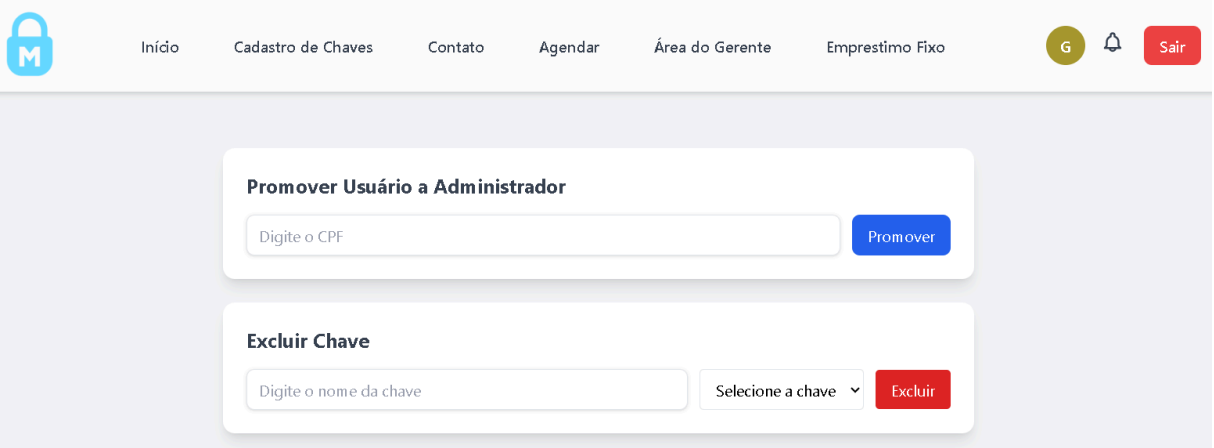
Nesta área, o Gerente pode executar ações administrativas, como promover um "Usuário" comum à função de "Administrador" e realizar a "exclusão de chaves" do sistema.

A página também apresenta um painel de análise com três gráficos focados nas reservas do mês atual, detalhando:

- As reservas por chave;
- Os horários mais comuns de solicitação;
- O total de reservas por dia.

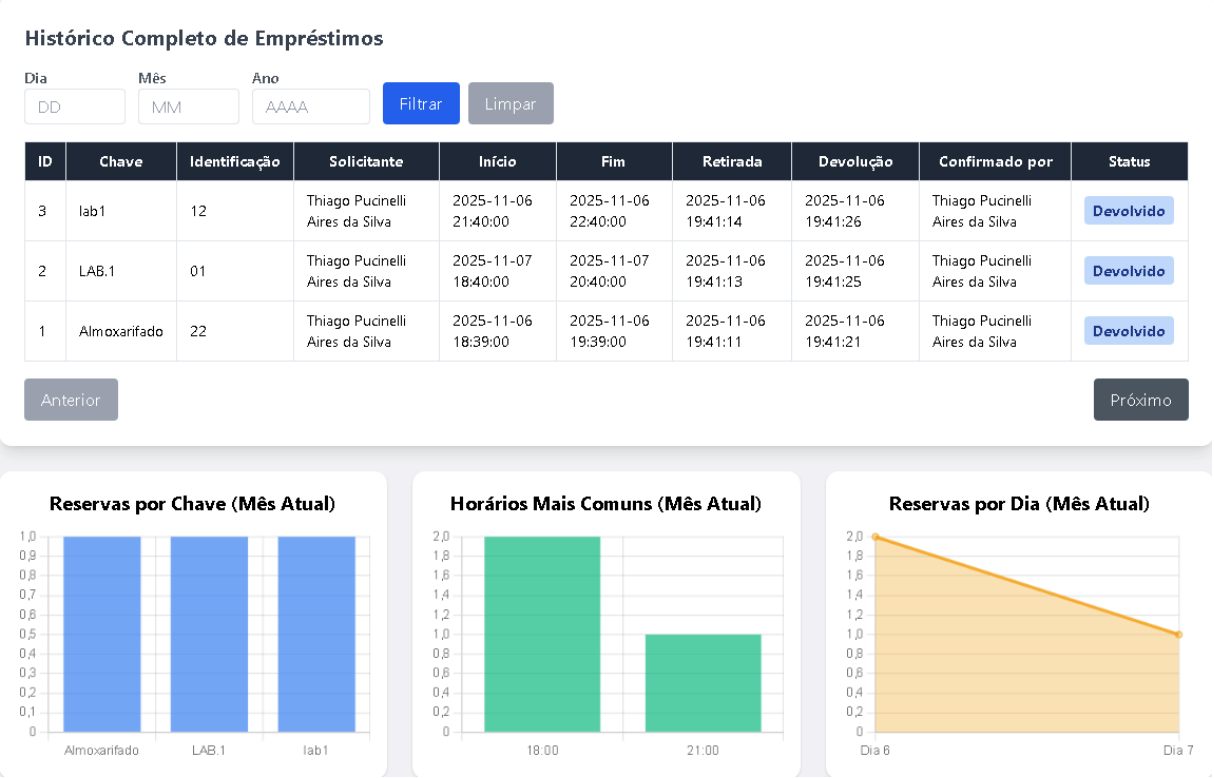
Adicionalmente, a seção disponibiliza o "histórico completo do uso das chaves", permitindo a consulta e auditoria de todas as movimentações registradas no sistema.

Figura 12 - Ações: Promover usuário e excluir chave.



Fonte - Própria (2025)

Figura 13 - Histórico completo de empréstimos e gráficos.



ID	Chave	Identificação	Solicitante	Início	Fim	Retirada	Devolução	Confirmado por	Status
3	lab1	12	Thiago Pucinelli Aires da Silva	2025-11-06 21:40:00	2025-11-06 22:40:00	2025-11-06 19:41:14	2025-11-06 19:41:26	Thiago Pucinelli Aires da Silva	Devolvido
2	LAB.1	01	Thiago Pucinelli Aires da Silva	2025-11-07 18:40:00	2025-11-07 20:40:00	2025-11-06 19:41:13	2025-11-06 19:41:25	Thiago Pucinelli Aires da Silva	Devolvido
1	Almoxarifado	22	Thiago Pucinelli Aires da Silva	2025-11-06 18:39:00	2025-11-06 19:39:00	2025-11-06 19:41:11	2025-11-06 19:41:21	Thiago Pucinelli Aires da Silva	Devolvido

Fonte - Própria (2025)

6. CONSIDERAÇÕES FINAIS

Tendo como objetivo a contribuição com a instituição de ensino IFSul/CaVG, no ano de 2025, no Curso Técnico em Desenvolvimento de Sistemas foi dado início ao projeto Chave Mestra, cujo propósito foi buscar informatizar o trabalho dos funcionários do setor de Gestão Acadêmica.

Considerando o objetivo geral deste trabalho de desenvolver um sistema web para substituir o controle manual de empréstimo e devolução de chaves, pode-se afirmar que este objetivo foi parcialmente alcançado. A aplicação web foi desenvolvida e mostrou-se funcional. Contudo, ao apresentar o projeto finalizado à gestão, concluímos, em conjunto com os responsáveis pelo setor, que o sistema ainda precisa passar por adaptações.

Notamos um receio natural por parte dos usuários em adaptar-se a uma nova forma de trabalho, mais moderna e informatizada, o que reforça a necessidade de ajustes para que o sistema se encaixe perfeitamente ao fluxo de trabalho da instituição. Em contrapartida, sob a ótica dos alunos, o objetivo foi alcançado com êxito, validando a utilidade da reserva on-line e da consulta de disponibilidade.

Um dos principais desafios encontrados durante o processo de criação do software foi em relação a saber ler e interpretar o código alheio, visto que o projeto foi desenvolvido em dupla. Essa experiência, embora complexa, serviu como um aprendizado fundamental sobre colaboração, versionamento e manutenção de código em equipe. Além disso, o projeto consolidou as competências técnicas adquiridas no curso, como a modelagem de banco de dados MySQL, programação de servidor em PHP e o desenvolvimento de interfaces com HTML, CSS e Tailwind CSS.

Como sugestão para trabalhos futuros, vislumbramos a adição de novas funcionalidades que aumentem a automação, como o registro de entregas via QR Code, o que traria ainda mais agilidade ao processo.

É inegável a contribuição que este trabalho pode proporcionar à instituição. Acreditamos que, por mais que o sistema não seja implementado de imediato, ele abre amplas possibilidades futuras de implementação. Mais do que isso, esperamos que este projeto sirva de inspiração para que outros alunos também se sintam motivados a construir projetos que beneficiem a comunidade acadêmica, visando o bem-estar coletivo.

REFERÊNCIAS

BRACKMANN, Christian Peter. *Desenvolvimento do pensamento computacional por meio de atividades desplugadas na educação básica.* 2017. 252 f. Tese (Doutorado em Informática na Educação) – Universidade Federal do Rio Grande do Sul, Porto Alegre, 2017.

BRASIL, **Constituição da República Federativa do Brasil.** Brasília - DF, 1988.

PRESSMAN, Roger S. *Engenharia de Software*. 7. ed. São Paulo: McGraw Hill, 2016.

SCHWABER, Ken; SUTHERLAND, Jeff. *Guia do Scrum: o guia definitivo do Scrum: as regras do jogo*. 2020. Disponível em: https://scrumguides.org. Acesso em: 08 abr. 2025.

SOMMERVILLE, Ian. *Engenharia de Software*. 10. ed. São Paulo: Pearson, 2011.

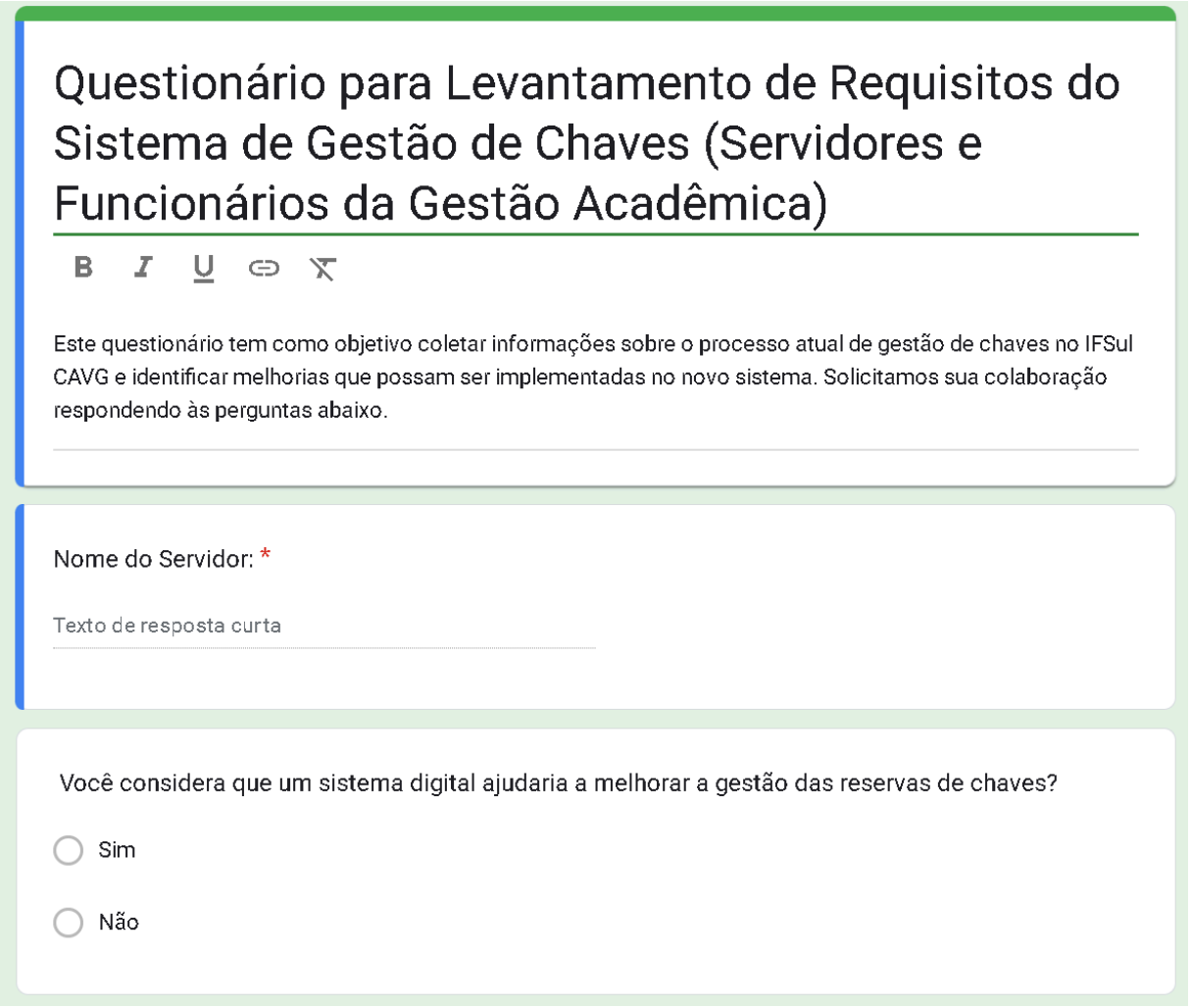
INSTITUTO FEDERAL SUL-RIO-GRANDENSE – CÂMPUS PELOTAS VISCONDE DA GRAÇA. Curso Técnico em Desenvolvimento de Sistemas. Pelotas: IFSul CAVG, 2025.

APÊNDICE 1: QUESTIONÁRIO APLICADO AOS FUNCIONÁRIOS E SERVIDORES

Perguntas:

1. Você considera que um sistema digital ajudaria a melhorar a gestão das reservas de chaves?
2. Quais registros são feitos atualmente?
3. Quais outros registros você acha que seriam úteis?
4. Como o sistema poderia tornar o processo mais eficiente para você?
5. Quais funcionalidades você considera essenciais para um sistema de controle de chaves?
6. Que tipos de relatórios sobre o uso das chaves seriam úteis para a administração?
7. Com que frequência são realizadas auditorias ou verificações das chaves?
8. Como você gostaria de visualizar esses relatórios?
9. Existe alguma preferência quanto ao design ou à forma de utilização do sistema?
10. Quais são os principais desafios enfrentados na gestão das chaves?
11. Com que frequência ocorrem problemas como perda ou extravio de chaves?
12. Há dificuldade na organização e busca por registros antigos?
13. Já houve casos de chaves não devolvidas ou mal registradas?
14. Como você lida com uma situação onde uma chave não foi devolvida?
15. Você teria alguma sugestão adicional para melhorar o sistema?

Figura 14 - Questões aplicadas a Gestão



Questionário para Levantamento de Requisitos do Sistema de Gestão de Chaves (Servidores e Funcionários da Gestão Acadêmica)

Este questionário tem como objetivo coletar informações sobre o processo atual de gestão de chaves no IFSul CAVG e identificar melhorias que possam ser implementadas no novo sistema. Solicitamos sua colaboração respondendo às perguntas abaixo.

Nome do Servidor: *

Texto de resposta curta

Você considera que um sistema digital ajudaria a melhorar a gestão das reservas de chaves?

- Sim
- Não

Fonte - Própria (2025)

Figura 15 - Questões aplicadas a Gestão

Quais registros são feitos atualmente?

- [ ] Entrada e saída das chaves
- [ ] Horário de empréstimo e devolução
- [ ] Nome do responsável pelo empréstimo
- [ ] Nome da sala
- [ ] Motivo do empréstimo
- [ ] Outro: _____

Quais outros registros você acha que seriam úteis?

Texto de resposta longa

Como o sistema poderia tornar o processo mais eficiente para você?

- [ ] Reduzindo erros no registro de empréstimos
- [ ] Agilizando a busca por informações
- [ ] Melhorando a rastreabilidade das chaves
- [ ] Enviando alertas automáticos para chaves não devolvidas
- [ ] Outro: _____

Fonte - Própria (2025)

Figura 16 - Questões aplicadas a Gestão

Quais funcionalidades você considera essenciais para um sistema de controle de chaves?

- [ ] Cadastro de chaves e usuários
- [ ] Registro digital de empréstimos e devoluções
- [ ] Notificação de chaves em atraso
- [ ] Geração de relatórios sobre uso das chaves
- [ ] Outro: _____

Que tipos de relatórios sobre o uso das chaves seriam úteis para a administração?

- [ ] Registro de empréstimos e devoluções
- [ ] Relatório de chaves mais utilizadas
- [ ] Relatório de chaves não devolvidas
- [ ] Relatório por período de uso (diário, semanal, mensal)
- [ ] Outro: _____

Com que frequência são realizadas auditorias ou verificações das chaves?

- [ ] Diariamente
- [ ] Semanalmente
- [ ] Mensalmente
- [ ] Raramente

Fonte - Própria (2025)

Figura 17 - Questões aplicadas a Gestão

Como você gostaria de visualizar esses relatórios?

- [ ] Tabelas e listas detalhadas
- [ ] Gráficos visuais
- [ ] Opção para exportar relatórios (PDF, Excel, etc.)
- [ ] Outro: _____

Existe alguma preferência quanto ao design ou à forma de utilização do sistema?

- [ ] Interface simples e intuitiva
- [ ] Suporte para dispositivos móveis
- [ ] Tema escuro para melhor leitura
- [ ] Outro: _____

Quais são os principais desafios enfrentados na gestão das chaves?

- [ ] Falta de organização nos registros
- [ ] Perda ou extravio de chaves
- [ ] Falta de controle sobre quem está com a chave
- [ ] Demora na busca por registros antigos
- [ ] Falta de um sistema de notificação para chaves não devolvidas
- [ ] Outro: _____

Fonte - Própria (2025)

Figura 18 - Questões aplicadas a Gestão

Com que frequência ocorrem problemas como perda ou extravio de chaves?

- Diariamente
- Semanalmente
- Mensalmente
- Raramente
- Nunca

Há dificuldade na organização e busca por registros antigos?

- Sim, frequentemente
- Sim, às vezes
- Não, os registros são bem organizados

Já houve casos de chaves não devolvidas ou mal registradas?

- Sim, com frequência
- Sim, mas raramente
- Não

Como você lida com uma situação onde uma chave não foi devolvida?

Texto de resposta longa

Você teria alguma sugestão adicional para melhorar o sistema?

Texto de resposta longa

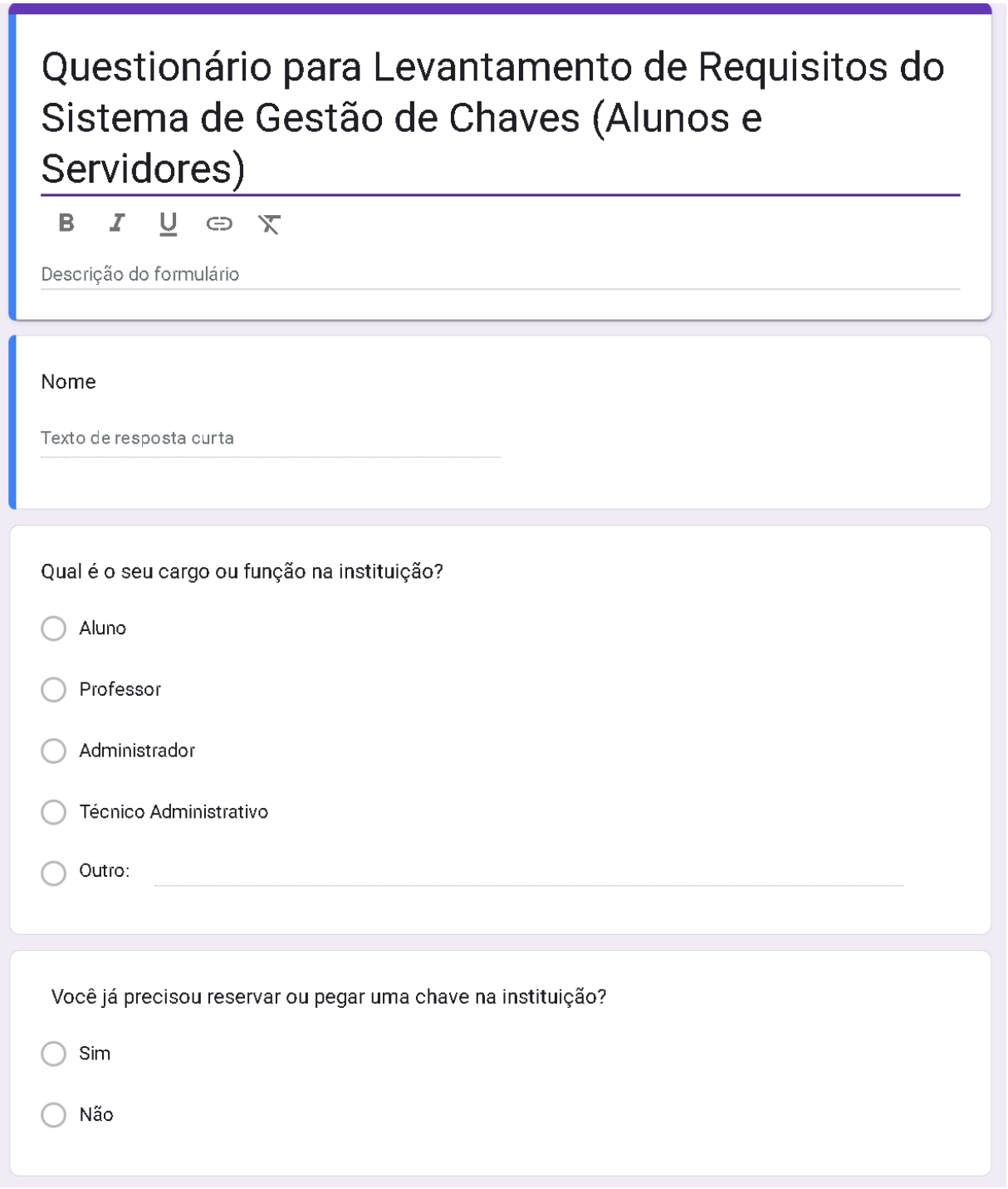
Fonte - Própria (2025)

APÊNDICE 2: QUESTIONÁRIO APLICADO AOS ALUNOS E FUNCIONÁRIOS

Perguntas:

1. Você já precisou reservar ou pegar uma chave na instituição?
2. Você já teve dificuldades ao solicitar uma chave? Se sim, quais?
3. Como você avalia o sistema atual de controle de chaves?
4. Você considera importante ter um sistema online para reserva de chaves?
5. Você tem alguma sugestão para melhorar o processo de reservas de chaves na instituição?

Figura 19 - Questões aplicadas aos Alunos e Servidores



Questionário para Levantamento de Requisitos do Sistema de Gestão de Chaves (Alunos e Servidores)

Descrição do formulário

Nome

Texto de resposta curta

Qual é o seu cargo ou função na instituição?

- Aluno
- Professor
- Administrador
- Técnico Administrativo
- Outro:

Você já precisou reservar ou pegar uma chave na instituição?

- Sim
- Não

Fonte - Própria (2025)

Figura 20 - Questões aplicadas aos Alunos e Servidores

Você já teve dificuldades ao solicitar uma chave? Se sim, quais?

- [ ] Chave não estava disponível
- [ ] Longa espera para retirada
- [ ] Falta de controle sobre quem pegou a chave
- [ ] Falha na reserva de chaves

Como você avalia o sistema atual de controle de chaves?

- ( ) Muito eficiente
- ( ) Eficiente
- ( ) Pouco eficiente
- ( ) Ineficiente

Você considera importante ter um sistema online para reserva de chaves?

- ( ) Sim
- ( ) Não
- ( ) Indiferente

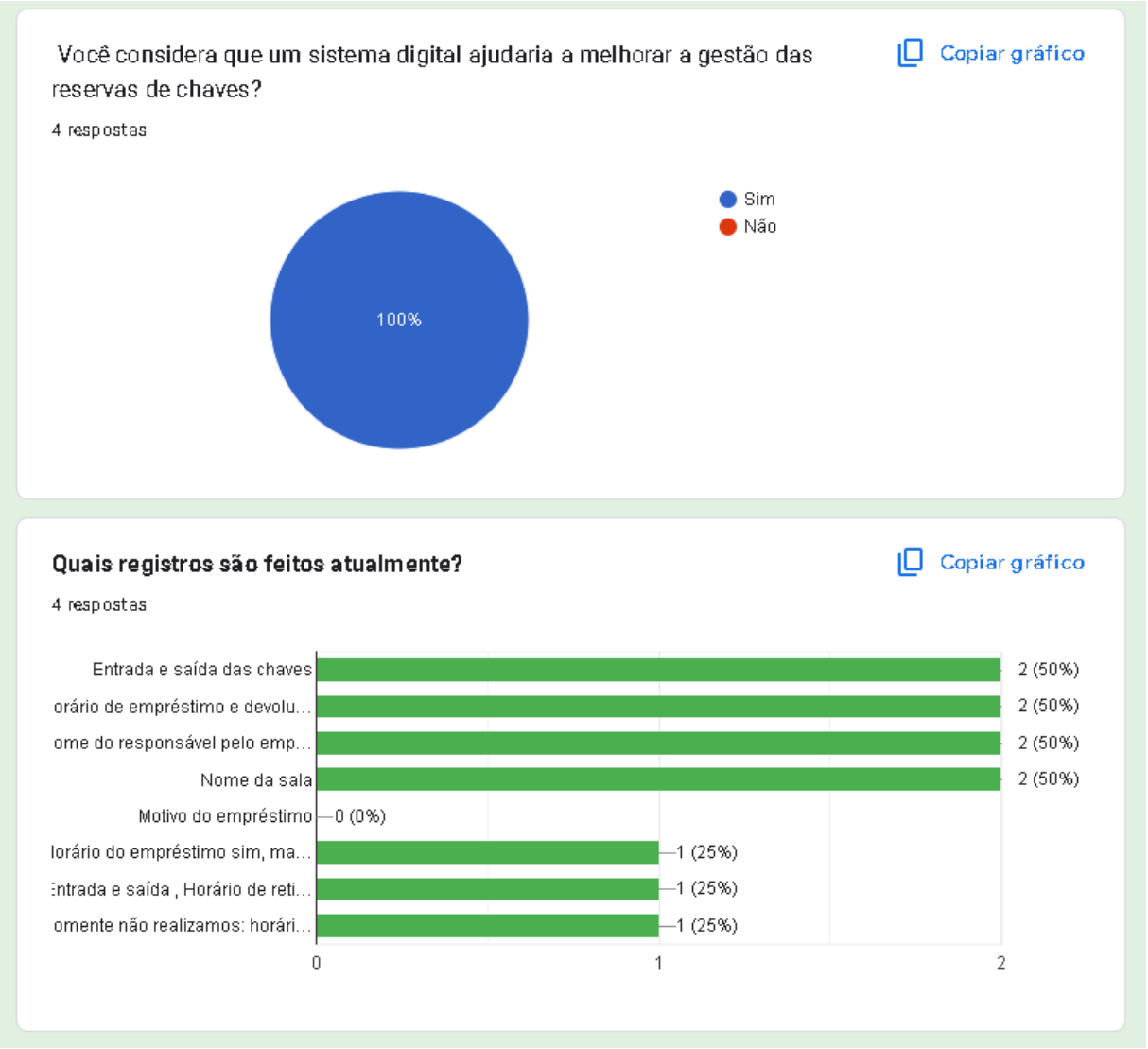
Você tem alguma sugestão para melhorar o processo de reservas de chaves na instituição?

Texto de resposta longa

Fonte - Própria (2025)

APÊNDICE 3: RESULTADO DO QUESTIONÁRIO APLICADO PARA LEVANTAMENTO DE REQUISITOS DO SISTEMA DE GESTÃO DE CHAVES (FUNCIONÁRIOS E SERVIDORES DA GESTÃO ACADÊMICA)

Figura 21 - Resultado das questões aplicadas aos funcionários e servidores

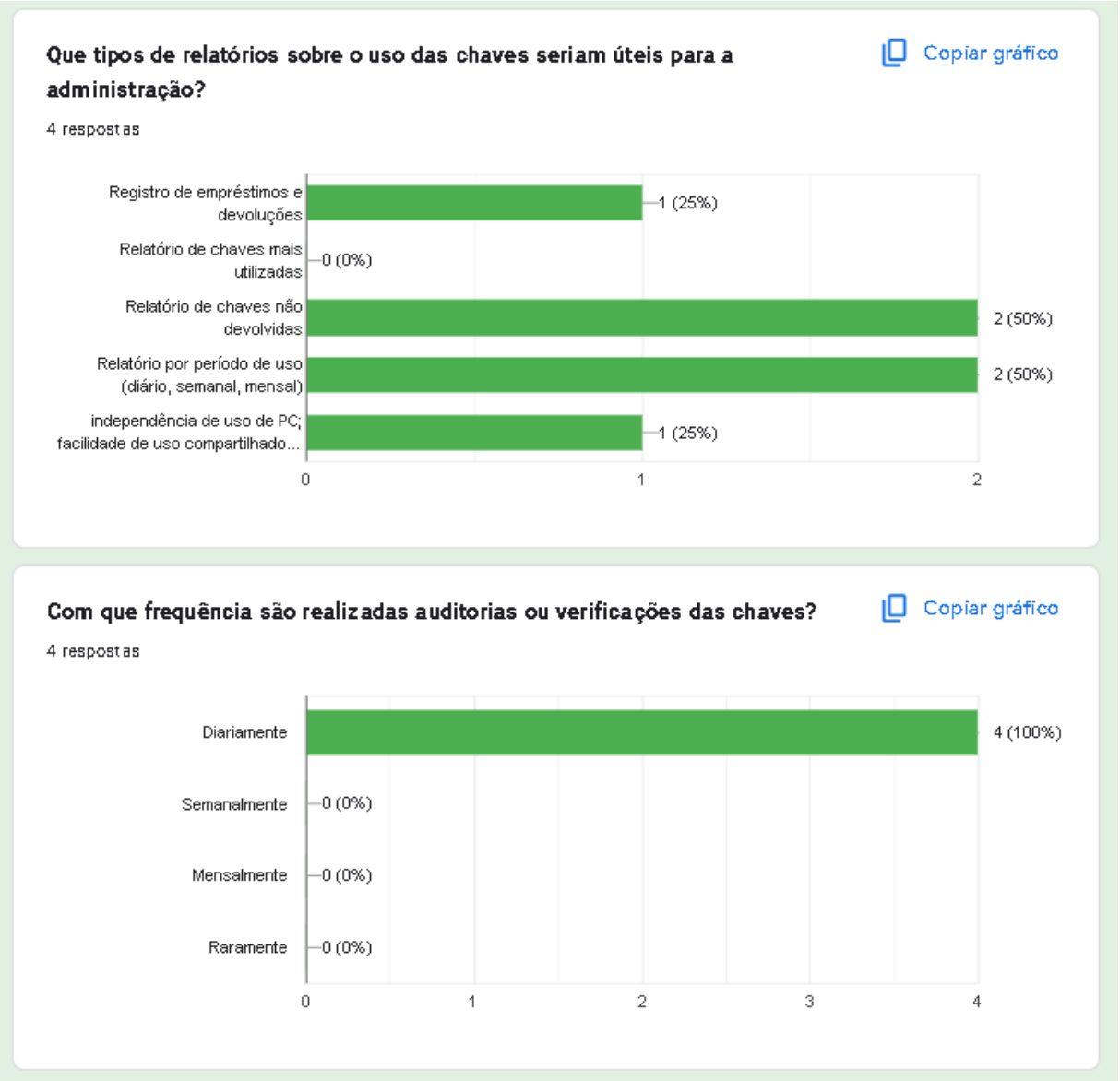


Fonte - Própria (2025)

Figura 22 - Resultado das questões aplicadas aos funcionários e servidores

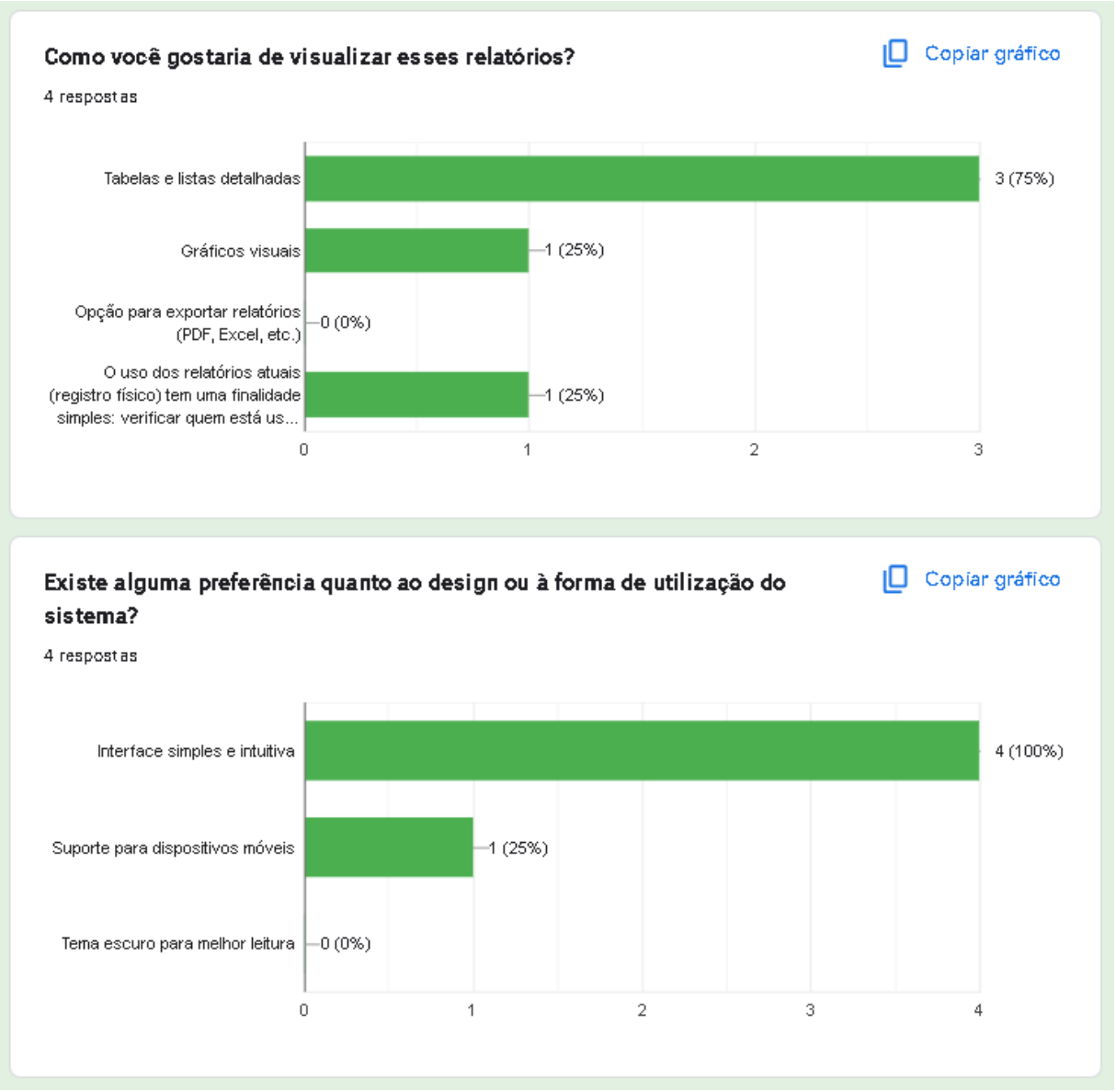
Fonte - Própria (2025)

Figura 23 - Resultado das questões aplicadas aos funcionários e servidores



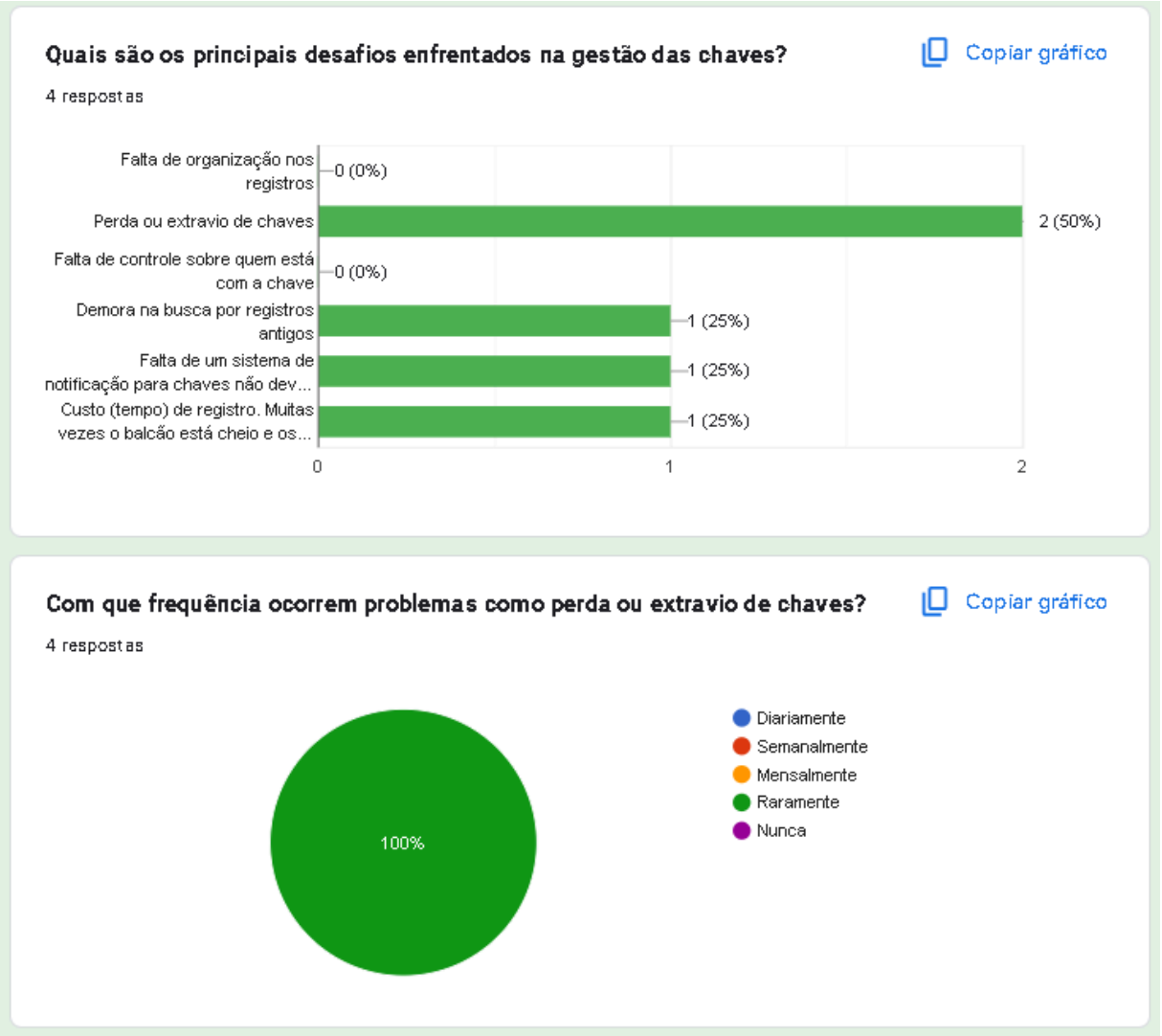
Fonte - Própria (2025)

Figura 24 - Resultado das questões aplicadas aos funcionários e servidores



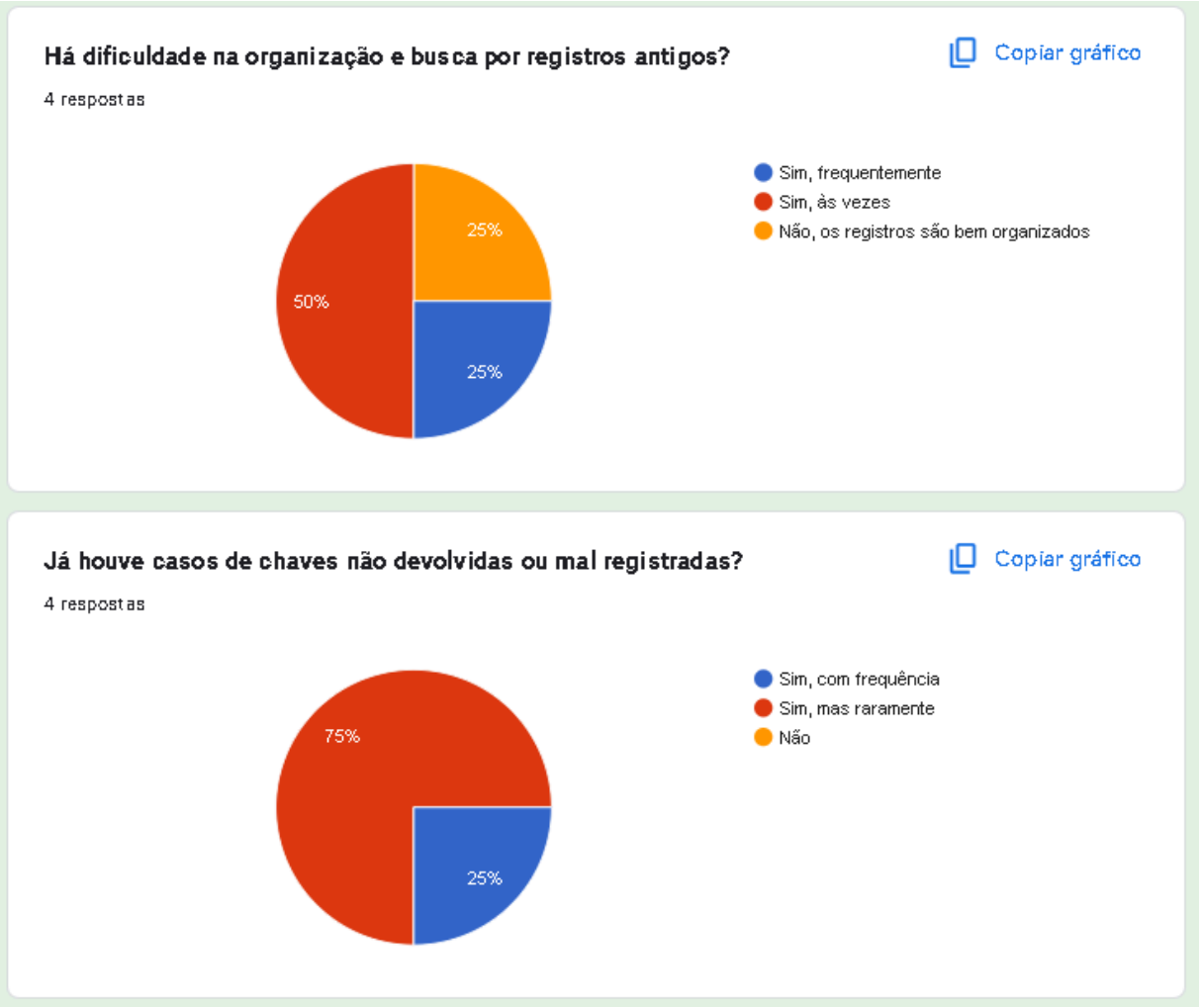
Fonte - Própria (2025)

Figura 25 - Resultado das questões aplicadas aos funcionários e servidores



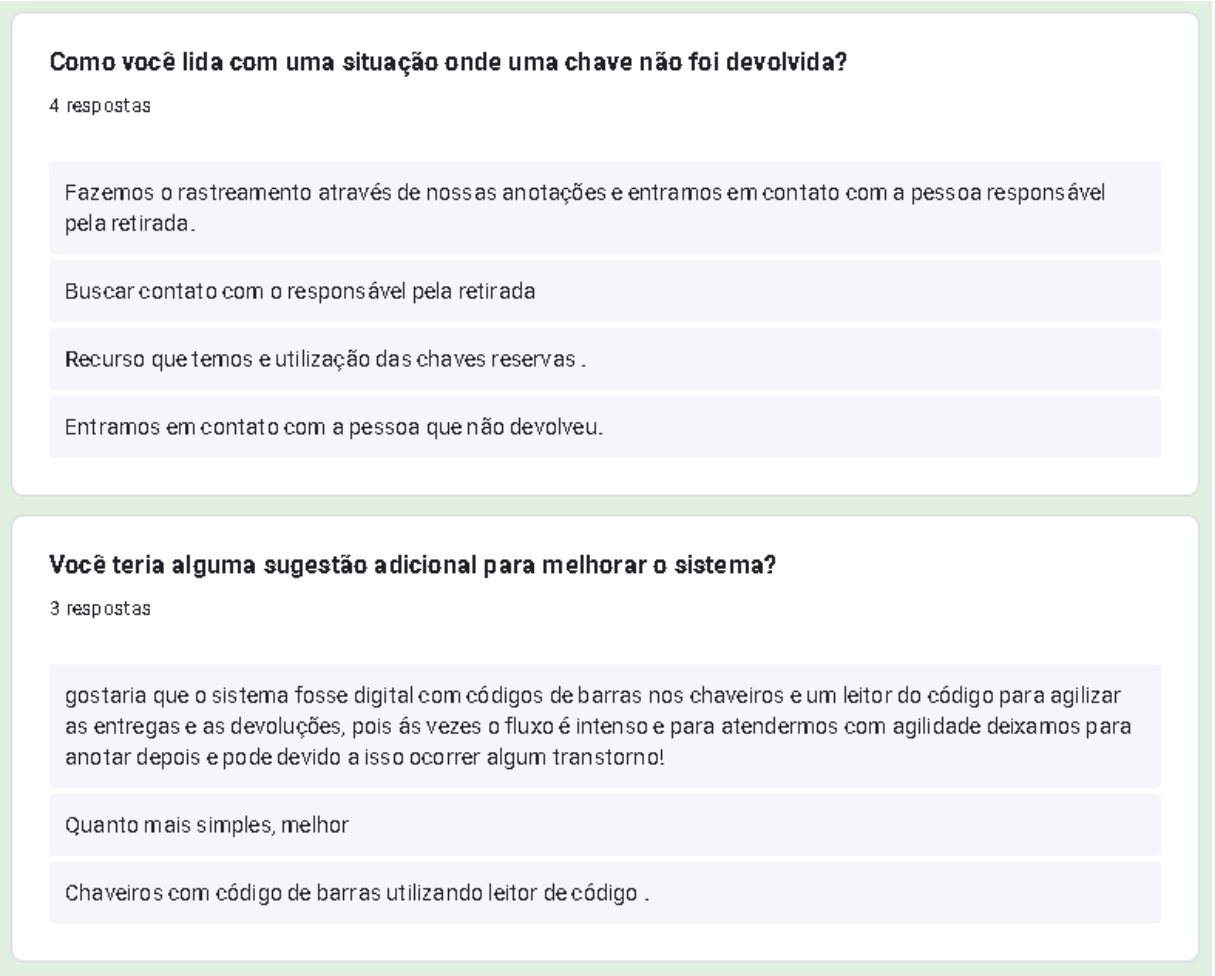
Fonte - Própria (2025)

Figura 26 - Resultado das questões aplicadas aos funcionários e servidores



Fonte - Própria (2025)

Figura 27 - Resultado das questões aplicadas aos funcionários e servidores



Como você lida com uma situação onde uma chave não foi devolvida?

4 respostas

Fazemos o rastreamento através de nossas anotações e entramos em contato com a pessoa responsável pela retirada.

Buscar contato com o responsável pela retirada

Recurso que temos e utilização das chaves reservas .

Entramos em contato com a pessoa que não devolveu.

Você teria alguma sugestão adicional para melhorar o sistema?

3 respostas

gostaria que o sistema fosse digital com códigos de barras nos chaveiros e um leitor do código para agilizar as entregas e as devoluções, pois ás vezes o fluxo é intenso e para atendermos com agilidade deixamos para anotar depois e pode devido a isso ocorrer algum transtorno!

Quanto mais simples, melhor

Chaveiros com código de barras utilizando leitor de código .

Fonte - Própria (2025)

APÊNDICE 4 - INSTRUÇÕES SQL PARA CRIAÇÃO DA BASE DE DADOS

```
-- phpMyAdmin SQL Dump
-- version 5.2.1
-- https://www.phpmyadmin.net/
--
-- Host: 127.0.0.1
-- Tempo de geração: 04/11/2025 às 07:50
-- Versão do servidor: 10.4.32-MariaDB
-- Versão do PHP: 8.2.12

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";
```

```
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT
*/;
/*!40101 SET
@OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION
*/;
/*!40101 SET NAMES utf8mb4 */;

--
-- Banco de dados: `chave-mestra3`
--

-- -------------------------------------------------------

--
-- Estrutura para tabela `chave`
--

CREATE TABLE `chave` (
  `id_chave` int(11) NOT NULL,
  `nome` varchar(100) NOT NULL,
  `numero_identificacao` varchar(20) NOT NULL,
  `descricao` text DEFAULT NULL,
  `quantidade` int(11) NOT NULL,
  `status` varchar(20) NOT NULL,
  `cpf_adm` char(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--
-- Despejando dados para a tabela `chave`
--

INSERT INTO `chave` (`id_chave`, `nome`, `numero_identificacao`, `descricao`,
`quantidade`, `status`, `cpf_adm`) VALUES
(57, 'lab1', '12', 'laboratotio de insafasgfag', 2, '', '11111111111'),
(58, 'Almoxarifado', '22', 'Sala asgfasgag', 4, '', '11111111111'),
(59, 'LAB.1', '01', 'Chave laboratorio 1', 2, '', '11111111111');

-- -------------------------------------------------------

--
-- Estrutura para tabela `emprestimo`
--
```

```
CREATE TABLE `emprestimo` (
  `id_emprestimo` int(11) NOT NULL,
  `data_reserva` date DEFAULT NULL,
  `data_inicio_reserva` datetime NOT NULL,
  `data_fim_reserva` datetime NOT NULL,
  `hora_data_retirada` datetime DEFAULT NULL,
  `hora_data_devolucao` datetime DEFAULT NULL,
  `id_chave` int(11) NOT NULL,
  `cpf_solicitante` char(11) NOT NULL,
  `cpf_adm` varchar(11) DEFAULT NULL,
  `categoria` varchar(50) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--
-- Despejando dados para a tabela `emprestimo`
--

INSERT INTO `emprestimo` (`id_emprestimo`, `data_reserva`, `data_inicio_reserva`, `data_fim_reserva`, `hora_data_retirada`, `hora_data_devolucao`, `id_chave`, `cpf_solicitante`, `cpf_adm`, `categoria`) VALUES
(90, '2025-11-02', '2025-11-02 19:48:00', '2025-11-02 21:48:00', '2025-11-02 17:48:32', '2025-11-02 17:48:35', 58, '11111111111', '11111111111', NULL),
(91, '2025-11-02', '2025-11-02 20:50:00', '2025-11-02 21:50:00', '2025-11-02 17:50:45', '2025-11-02 17:50:52', 59, '11111111111', '11111111111', NULL),
(92, '2025-11-02', '2025-11-02 19:50:00', '2025-11-02 22:50:00', '2025-11-02 17:51:11', '2025-11-02 17:51:11', 59, '11111111111', '11111111111', 'Cancelado'),
(93, '2025-11-02', '2025-11-02 17:59:00', '2025-11-21 17:59:00', '2025-11-02 18:00:23', '2025-11-02 18:00:23', 57, '11111111111', '11111111111', 'Cancelado'),
(94, '2025-11-02', '2025-11-02 20:08:00', '2025-11-02 22:08:00', '2025-11-02 18:11:40', '2025-11-02 18:20:33', 58, '22222222222', '11111111111', NULL),
(95, '2025-11-02', '2025-11-02 22:13:00', '2025-11-02 23:09:00', '2025-11-02 18:11:49', '2025-11-02 18:20:35', 58, '22222222222', '11111111111', NULL),
(96, '2025-11-02', '2025-11-03 18:09:00', '2025-11-03 21:09:00', '2025-11-02 18:11:46', '2025-11-02 18:20:38', 59, '22222222222', '11111111111', NULL),
(97, '2025-11-02', '2025-11-04 20:11:00', '2025-11-04 23:11:00', '2025-11-02 18:11:44', '2025-11-02 18:20:41', 59, '22222222222', '11111111111', NULL),
(98, '2025-11-02', '2025-11-02 19:20:00', '2025-11-02 20:20:00', '2025-11-02 18:27:51', '2025-11-02 18:30:51', 58, '11111111111', '11111111111', NULL),
(102, '2025-11-02', '2025-11-02 20:03:00', '2025-11-02 22:03:00', '2025-11-02 19:03:45', '2025-11-02 19:03:45', 58, '22222222222', '11111111111', 'Cancelado'),
(105, '2025-11-02', '2025-11-02 21:26:00', '2025-11-02 22:26:00', '2025-11-03 22:34:42', '2025-11-03 22:37:50', 59, '22222222222', '11111111111', 'Cancelado'),
```

```
(106, '2025-11-02', '2025-11-02 21:31:00', '2025-11-02 22:31:00', '2025-11-03 22:34:39', '2025-11-03 22:37:47', 58, '22222222222', '11111111111', 'Cancelado'),
(107, '2025-11-03', '2025-11-04 10:30:00', '2025-11-04 14:30:00', '2025-11-03 22:34:34', '2025-11-03 22:37:33', 58, '11111111111', '11111111111', NULL),
(108, '2025-11-03', '2025-11-04 15:25:00', '2025-11-04 16:27:00', '2025-11-03 23:36:26', '2025-11-03 23:36:45', 58, 'ZD4WB0B4OPS', '11111111111', NULL),
(109, '2025-11-04', '2025-11-04 03:25:00', '2025-11-04 05:25:00', '2025-11-04 01:30:19', '2025-11-04 01:30:25', 59, '11111111111', '11111111111', NULL),
(110, '2025-11-04', '2025-11-04 05:31:00', '2025-11-04 06:31:00', '2025-11-04 01:35:42', '2025-11-04 01:35:50', 59, '11111111111', '11111111111', NULL),
(111, '2025-11-04', '2025-11-05 03:33:00', '2025-11-05 06:33:00', '2025-11-04 01:35:45', '2025-11-04 01:35:52', 57, '11111111111', '11111111111', NULL),
(112, '2025-11-04', '2025-11-11 15:34:00', '2025-11-11 18:35:00', '2025-11-04 01:35:39', '2025-11-04 01:35:47', 58, '11111111111', '11111111111', NULL),
(113, '2025-11-04', '2025-11-04 04:36:00', '2025-11-04 06:36:00', '2025-11-04 01:43:27', '2025-11-04 01:50:58', 59, '11111111111', '22222222222', NULL),
(114, '2025-11-04', '2025-11-04 01:45:00', '2025-11-04 01:51:00', '2025-11-04 01:43:23', '2025-11-04 01:50:53', 58, '11111111111', '22222222222', NULL),
(115, '2025-11-04', '2025-11-04 02:02:00', '2025-11-04 02:03:00', '2025-11-04 02:02:11', '2025-11-04 02:03:34', 59, '11111111111', '11111111111', NULL),
(116, '2025-11-04', '2025-11-04 02:03:00', '2025-11-04 02:04:00', '2025-11-04 02:04:56', '2025-11-04 02:06:03', 59, '11111111111', '11111111111', NULL),
(117, '2025-11-04', '2025-11-04 02:07:00', '2025-11-04 02:08:00', '2025-11-04 02:09:27', '2025-11-04 02:09:32', 59, '11111111111', '11111111111', NULL),
(118, '2025-11-04', '2025-11-04 02:10:00', '2025-11-04 02:11:00', '2025-11-04 02:10:07', '2025-11-04 02:23:12', 59, '11111111111', '11111111111', NULL),
(119, '2025-11-04', '2025-11-04 02:23:00', '2025-11-04 02:24:00', '2025-11-04 02:23:44', '2025-11-04 02:37:29', 57, '11111111111', '11111111111', NULL);

-- --------------------------------------------------------

--
-- Estrutura da tabela `emprestimo_fixo`
--

CREATE TABLE `emprestimo_fixo` (
  `id_fixo` int NOT NULL,
  `id_chave` int NOT NULL,
  `cpf_solicitante` char(11) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL,
  `cpf_admin` char(11) COLLATE utf8mb4_general_ci NOT NULL,
  `dia_semana` tinyint NOT NULL,
  `hora_inicio` time NOT NULL,
  `hora_fim` time NOT NULL
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--
-- Extraindo dados da tabela `emprestimo_fixo`
--

INSERT INTO `emprestimo_fixo` (`id_fixo`, `id_chave`, `cpf_solicitante`, `cpf_admin`,
`dia_semana`, `hora_inicio`, `hora_fim`) VALUES
(14, 58, '22222222222', '11111111111', 0, '18:24:00', '20:24:00'),
(15, 58, '22222222222', '11111111111', 1, '18:24:00', '20:24:00'),
(16, 58, '22222222222', '11111111111', 6, '18:24:00', '20:24:00'),
(17, 61, '22222222222', '11111111111', 0, '20:30:00', '23:30:00'),
(18, 61, '22222222222', '11111111111', 1, '20:30:00', '23:30:00'),
(19, 61, '22222222222', '11111111111', 2, '20:30:00', '23:30:00');

-- --------------------------------------------------------

--
-- Estrutura para tabela `notificacao_usuario`
--

CREATE TABLE `notificacao_usuario` (
  `id_notificacao` int(11) NOT NULL,
  `cpf_usuario` varchar(14) NOT NULL,
  `id_emprestimo` int(11) DEFAULT NULL,
  `mensagem` text NOT NULL,
  `lida` tinyint(1) DEFAULT 0,
  `data_criacao` datetime DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

-- --------------------------------------------------------

--
-- Estrutura para tabela `notificacoes_admin`
--

CREATE TABLE `notificacoes_admin` (
  `id_notificacao` int(11) NOT NULL,
  `id_emprestimo` int(11) NOT NULL,
  `cpf_admin` varchar(14) NOT NULL,
  `mensagem` varchar(255) NOT NULL,
  `data_criacao` datetime DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```
-- ----------------------------------------------------------

--
-- Estrutura para tabela `solicitante`
--

CREATE TABLE `solicitante` (
  `cpf` char(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--
-- Despejando dados para a tabela `solicitante`
--

INSERT INTO `solicitante` (`cpf`) VALUES
('11111111111'),
('22222222222'),
('ZD4WB0B4OPS');

-- ----------------------------------------------------------

--
-- Estrutura para tabela `usuario`
--

CREATE TABLE `usuario` (
  `cpf` char(11) NOT NULL,
  `nome` varchar(100) NOT NULL,
  `senha` varchar(255) NOT NULL,
  `email` varchar(100) NOT NULL,
  `categoria` varchar(50) DEFAULT NULL,
  `info_categoria` varchar(150) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--
-- Despejando dados para a tabela `usuario`
--

INSERT INTO `usuario` (`cpf`, `nome`, `senha`, `email`, `categoria`, `info_categoria`)
VALUES
('11111111111', 'Thiago Pucinelli Aires da Silva',
'$2y$10$5VV5060TF3NnPok8TQn2muRcYvLv0ONp7vMyxO10nhflBwFJyiJAO',
'thiago.pucinelli.177@gmail.com', 'Aluno', '123444444TDSCAVGs'),
```

```
('22222222222', 'Thiago Teste',
'$2y$10$dCufVzSbrKBYUF7FAxpjOOM8uAZntUh/Ahe/7hkQ.c194NpE0XxE2',
'thiagopucinellisenac@gmail.com', 'Aluno', 'Cavg1222sTDSTeste'),
('ZD4WB0B4OPS', 'Thiago usuario',
'$2y$10$3ZJWbwtYKlOlM5Q0noLrMu1Iv0gbo6/uy6dy8Yvn6rNV/.pw.ZVYO',
'thiagousuario@gmail.com', 'aluno', '20211VG.TDS_S0019');

-- --------------------------------------------------------

--
-- Estrutura para tabela `usuario_adm`
--

CREATE TABLE `usuario_adm` (
  `cpf` char(11) NOT NULL,
  `tipo` varchar(20) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--
-- Despejando dados para a tabela `usuario_adm`
--

INSERT INTO `usuario_adm` (`cpf`, `tipo`) VALUES
('11111111111', 'Gerente'),
('22222222222', 'Administrador');

-- --------------------------------------------------------

--
-- Estrutura stand-in para view `vw_historico_usuario`
-- (Veja abaixo para a visão atual)
--
CREATE TABLE `vw_historico_usuario` (
`id_emprestimo` int(11)
,`data_reserva` date
,`data_inicio_reserva` datetime
,`data_fim_reserva` datetime
,`hora_data_retirada` datetime
,`hora_data_devolucao` datetime
,`nome_chave` varchar(100)
,`numero_identificacao` varchar(20)
,`descricao` mediumtext
,`categoria` varchar(50)
,`cpf_solicitante` char(11)
```

```
,`cpf_adm` varchar(11)
,`status` varchar(12)
);

-- -------------------------------------------------------

--
-- Estrutura para view `vw_historico_usuario`
--
DROP TABLE IF EXISTS `vw_historico_usuario`;

CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY DEFINER VIEW `vw_historico_usuario` AS SELECT `e`.`id_emprestimo` AS `id_emprestimo`, `e`.`data_reserva` AS `data_reserva`, `e`.`data_inicio_reserva` AS `data_inicio_reserva`, `e`.`data_fim_reserva` AS `data_fim_reserva`, `e`.`hora_data_retirada` AS `hora_data_retirada`, `e`.`hora_data_devolucao` AS `hora_data_devolucao`, `c`.`nome` AS `nome_chave`, `c`.`numero_identificacao` AS `numero_identificacao`, `c`.`descricao` AS `descricao`, `e`.`categoria` AS `categoria`, `e`.`cpf_solicitante` AS `cpf_solicitante`, `e`.`cpf_adm` AS `cpf_adm`, CASE WHEN `e`.`categoria` = 'Cancelado' THEN 'Cancelado' WHEN `e`.`hora_data_retirada` is null THEN 'Reservado' WHEN `e`.`hora_data_retirada` is not null AND `e`.`hora_data_devolucao` is null THEN 'Em uso' WHEN `e`.`hora_data_devolucao` is not null THEN 'Devolvido' ELSE 'Desconhecido' END AS `status` FROM (`emprestimo` `e` join `chave` `c` on(`e`.`id_chave` = `c`.`id_chave`))union all select `f`.`id_fixo` AS `id_emprestimo`,NULL AS `data_reserva`,NULL AS `data_inicio_reserva`,NULL AS `data_fim_reserva`,NULL AS `hora_data_retirada`,NULL AS `hora_data_devolucao`,`c`.`nome` AS `nome_chave`,`c`.`numero_identificacao` AS `numero_identificacao`,`c`.`descricao` AS `descricao`,'Fixo' AS `categoria`,`f`.`cpf_solicitante` AS `cpf_solicitante`,NULL AS `cpf_adm`,'Fixo' AS `status` from (`emprestimo_fixo` `f` join `chave` `c` on(`f`.`id_chave` = `c`.`id_chave`)) ;

--
-- Índices para tabelas despejadas
--

--
-- Índices de tabela `chave`
--
ALTER TABLE `chave`
  ADD PRIMARY KEY (`id_chave`),
  ADD UNIQUE KEY `numero_identificacao` (`numero_identificacao`),
  ADD KEY `cpf_adm` (`cpf_adm`);

--
```

```
-- Índices de tabela `emprestimo`
--
ALTER TABLE `emprestimo`
  ADD PRIMARY KEY (`id_emprestimo`),
  ADD KEY `id_chave` (`id_chave`),
  ADD KEY `cpf_solicitante` (`cpf_solicitante`),
  ADD KEY `cpf_adm` (`cpf_adm`);

--
-- Índices de tabela `emprestimo_fixo`
--
ALTER TABLE `emprestimo_fixo`
  ADD PRIMARY KEY (`id_fixo`),
  ADD KEY `id_chave` (`id_chave`),
  ADD KEY `cpf_solicitante` (`cpf_solicitante`);

--
-- Índices de tabela `notificacao_usuario`
--
ALTER TABLE `notificacao_usuario`
  ADD PRIMARY KEY (`id_notificacao`),
  ADD KEY `cpf_usuario` (`cpf_usuario`),
  ADD KEY `id_emprestimo` (`id_emprestimo`);

--
-- Índices de tabela `notificacoes_admin`
--
ALTER TABLE `notificacoes_admin`
  ADD PRIMARY KEY (`id_notificacao`);

--
-- Índices de tabela `solicitante`
--
ALTER TABLE `solicitante`
  ADD PRIMARY KEY (`cpf`);

--
-- Índices de tabela `usuario`
--
ALTER TABLE `usuario`
  ADD PRIMARY KEY (`cpf`);

--
-- Índices de tabela `usuario_adm`
```

```
--
ALTER TABLE `usuario_adm`
  ADD PRIMARY KEY (`cpf`);

--
-- AUTO_INCREMENT para tabelas despejadas
--

--
-- AUTO_INCREMENT de tabela `chave`
--
ALTER TABLE `chave`
  MODIFY `id_chave` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=60;

--
-- AUTO_INCREMENT de tabela `emprestimo`
--
ALTER TABLE `emprestimo`
        MODIFY `id_emprestimo` int(11) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=120;

--
-- AUTO_INCREMENT de tabela `emprestimo_fixo`
--
ALTER TABLE `emprestimo_fixo`
  MODIFY `id_fixo` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=9;

--
-- AUTO_INCREMENT de tabela `notificacao_usuario`
--
ALTER TABLE `notificacao_usuario`
        MODIFY `id_notificacao` int(11) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=9;

--
-- AUTO_INCREMENT de tabela `notificacoes_admin`
--
ALTER TABLE `notificacoes_admin`
        MODIFY `id_notificacao` int(11) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=54;

--
-- Restrições para tabelas despejadas
--
```

```
--
-- Restrições para tabelas `chave`
--
ALTER TABLE `chave`
   ADD CONSTRAINT `chave_ibfk_1` FOREIGN KEY (`cpf_adm`) REFERENCES
`usuario_adm` (`cpf`);

--
-- Restrições para tabelas `emprestimo`
--
ALTER TABLE `emprestimo`
      ADD CONSTRAINT `emprestimo_ibfk_1` FOREIGN KEY (`id_chave`)
REFERENCES `chave` (`id_chave`),
     ADD CONSTRAINT `emprestimo_ibfk_2` FOREIGN KEY (`cpf_solicitante`)
REFERENCES `solicitante` (`cpf`),
  ADD CONSTRAINT `emprestimo_ibfk_3` FOREIGN KEY (`cpf_adm`) REFERENCES
`usuario_adm` (`cpf`);

--
-- Restrições para tabelas `emprestimo_fixo`
--
ALTER TABLE `emprestimo_fixo`
     ADD CONSTRAINT `emprestimo_fixo_ibfk_1` FOREIGN KEY (`id_chave`)
REFERENCES `chave` (`id_chave`),
   ADD CONSTRAINT `emprestimo_fixo_ibfk_2` FOREIGN KEY (`cpf_solicitante`)
REFERENCES `solicitante` (`cpf`);

--
-- Restrições para tabelas `notificacao_usuario`
--
ALTER TABLE `notificacao_usuario`
   ADD CONSTRAINT `notificacao_usuario_ibfk_1` FOREIGN KEY (`cpf_usuario`)
REFERENCES `usuario` (`cpf`) ON DELETE CASCADE,
  ADD CONSTRAINT `notificacao_usuario_ibfk_2` FOREIGN KEY (`id_emprestimo`)
REFERENCES `emprestimo` (`id_emprestimo`) ON DELETE CASCADE;

--
-- Restrições para tabelas `solicitante`
--
ALTER TABLE `solicitante`
   ADD CONSTRAINT `solicitante_ibfk_1` FOREIGN KEY (`cpf`) REFERENCES
`usuario` (`cpf`);
```

```
--
-- Restrições para tabelas `usuario_adm`
--
ALTER TABLE `usuario_adm`
   ADD CONSTRAINT `usuario_adm_ibfk_1` FOREIGN KEY (`cpf`) REFERENCES
`usuario` (`cpf`);
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET
CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```