



INSTITUTO FEDERAL
Rio Grande do Norte

Campus - Caicó

LABKEY

SISTEMA DE GERENCIAMENTO DE CHAVES

ATHIELY TAIANY • IGOR BRUNO • JOAQUIM MEDEIROS • LICURGO KEVEN • MARIA ANDRADE • MELISSA KAREN

GRUPO DOSCENTE: ANDERSSON • MAIRA ANDRADE • MAX MILLER



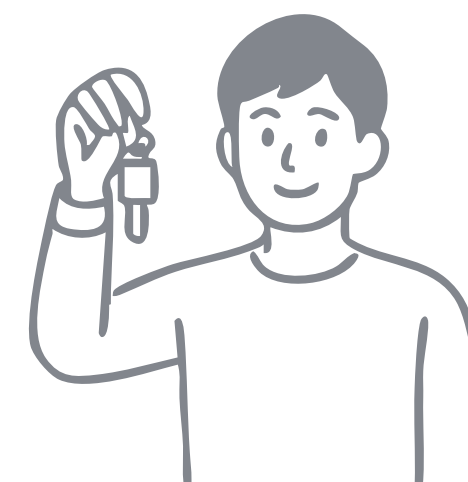
ÍNDICE



03	INTRODUÇÃO
04	JUSTIFICATIVA
05	OBJETIVOS
06	REQUISITOS
07	TECNOLOGIAS UTILIZADAS
13	RESULTADO
14	CONCLUSÃO

Introdução

O projeto LabKey tem como propósito disponibilizar uma aplicação web para auxiliar instituições educacionais e outras organizações no gerenciamento da utilização de laboratórios e salas por terceiros.



Justificativa

- Necessidade de centralizar e automatizar o processo de reserva de chaves.
- Solucionar problemas de organização e comunicação entre usuários.
- Melhorar a segurança no acesso aos laboratórios.
- Facilitar o trabalho de docentes, discentes e terceirizados.



Objetivos

Objetivo Geral

Desenvolver um sistema web de gerenciamento de reservas e controle de acesso a laboratórios e salas, oferecendo uma plataforma eficiente, intuitiva e segura para auxiliar instituições na administração de seus ambientes físicos.

Objetivos específicos

- Permitir o cadastro de laboratórios e salas, incluindo informações como nome, localização, capacidade e disponibilidade.
- Implementar o controle de reservas, possibilitando ao gestor visualizar a ocupação em tempo real e registrar agendamentos.
- Desenvolver uma interface amigável, responsiva e de fácil uso, alinhada às necessidades dos gestores.

Requisitos

Requisitos Funcionais

Código	Identificação	Classificação	Atores
[RF001]	Fazer Cadastro	Essencial	Aluno, Professor e Funcionário
[RF002]	Efetuar Login	Essencial	Aluno, Professor e Funcionário
[RF003]	Redefinir Senha	Importante	Aluno, Professor e Funcionário
[RF004]	Cadastrar Salas	Essencial	Administrador
[RF005]	Listar Salas	Essencial	Aluno, Professor, Funcionário e Administrador
[RF006]	Editar Sala	Importante	Administrador
[RF007]	Excluir Sala	Importante	Administrador
[RF008]	Realizar Reserva	Essencial	Aluno, Professor e Funcionário
[RF009]	Validar Disponibilidade	Essencial	Sistema
[RF010]	Visualizar Reservas	Importante	Aluno, Professor, Funcionário e Administrador
[RF011]	Cancelar Reserva	Importante	Aluno, Professor, Funcionário e Administrador
[RF012]	Notificar Confirmação	Desejável	Sistema
[RF013]	Evitar Conflitos	Desejável	Sistema
[RF014]	Reservar Recorrentemente	Desejável	Professor, Funcionário
[RF015]	Diferenciar Permissões	Essencial	Sistema
[RF016]	Gerenciar Reservas de Outros	Essencial	Administrador
[RF017]	Gerar Relatórios	Desejável	Administrador
[RF018]	Registrar Ações	Importante	Sistema

Requisitos Não-Funcionais

Código	Identificação	Descrição
[NF001]	Desempenho	O sistema deve ser capaz de receber várias requisições simultâneas sem degradação significativa de desempenho.
[NF002]	Usabilidade	A interface do usuário deve ser intuitiva e permitir que os usuários realizem as principais tarefas em poucos cliques.
[NF003]	Segurança	O sistema deve garantir que todos os dados sensíveis sejam criptografados em trânsito e em repouso.
[NF004]	Compatibilidade	O sistema deve ser compatível com os principais navegadores (Chrome, Firefox, Safari e Edge) e dispositivos móveis. Além disso, deve funcionar em sistemas operacionais Windows, macOS e Linux.
[NF005]	Confiabilidade	O sistema deve garantir uma disponibilidade de 99,5% durante o horário de funcionamento.

Tecnologias utilizadas



- Linguagem de programação Python;
- Framework web Fastapi;
- Biblioteca Python moderna para trabalhar com bancos de dados chamada Sqlmodel;
- Framework Bootstrap e coreUI;
- UML usado para os diagramas.

API - Rotas

```
@app.put(
    "/api/v1/reservas/{reserva_id}/status",
    summary="Mudar o status da reserva (ADMIN)",
    dependencies=[Depends(verificar_admin)]
)
def mudar_status_reserva(
    reserva_id: int,
    status_input: dict,
    session: Session = Depends(get_session)
):
    """Endpoint para que o Administrador altere o status (Aprovada/Rejeitada/Cancelada) de uma reserva."""
    # Busca a reserva
    reserva = session.get(Reserva, reserva_id)
    if not reserva:
        raise HTTPException(status_code=404, detail="Reserva não encontrada.")

    # Extrai e valida o novo status
    novo_status_str = status_input.get("status")
    if not novo_status_str:
        raise HTTPException(status_code=400, detail="Status não fornecido.")

    try:
        novo_status = StatusReserva(novo_status_str)
    except ValueError:
        raise HTTPException(status_code=400, detail="Status inválido.")

    # Atualiza e salva o status
    reserva.status = novo_status

    session.add(reserva)
    session.commit()
    session.refresh(reserva)

    return {"mensagem": f"Status alterado para {novo_status_str} com sucesso!", "status": reserva.status.value}
```

```
@app.post(
    "/api/v1/reservas",
    summary="Solicitar reserva de uma sala",
    status_code=status.HTTP_201_CREATED
)
def solicitar_reserva(
    dados: ReservaInput,
    request: Request,
    session: Session = Depends(get_session)
):
    """Endpoint para solicitar uma nova reserva. Requer que o usuário esteja logado."""
    if "usuario_id" not in request.session:
        raise HTTPException(status_code=401, detail="Usuário não autenticado.")

    usuario_id = request.session["usuario_id"]

    # Verifica se a sala existe
    sala = session.get(Sala, dados.sala_id)
    if not sala:
        raise HTTPException(status_code=404, detail="Sala não encontrada.")

    # Cria a reserva com status PENDENTE
    reserva = Reserva(
        data=dados.data,
        hora_inicio=dados.hora_inicio,
        hora_fim=dados.hora_fim,
        sala_id=dados.sala_id,
        usuario_id=usuario_id,
        status=StatusReserva.PENDENTE
    )


    session.add(reserva)
    session.commit()
    session.refresh(reserva)

    return {"mensagem": "Solicitação de reserva enviada com sucesso!", "reserva_id": reserva.id, "status": reserva.status.value}
```


Reservas de laboratórios (user comum)

A pagina de reservas tem como objetivo exibir as reservas feitas pelos usuários, permitindo que eles façam as solicitações de reserva, edições em reservas já solicitadas bem como seu cancelamento .

Resultados



SalasReservasOlá, teste ▾

Minhas Reservas

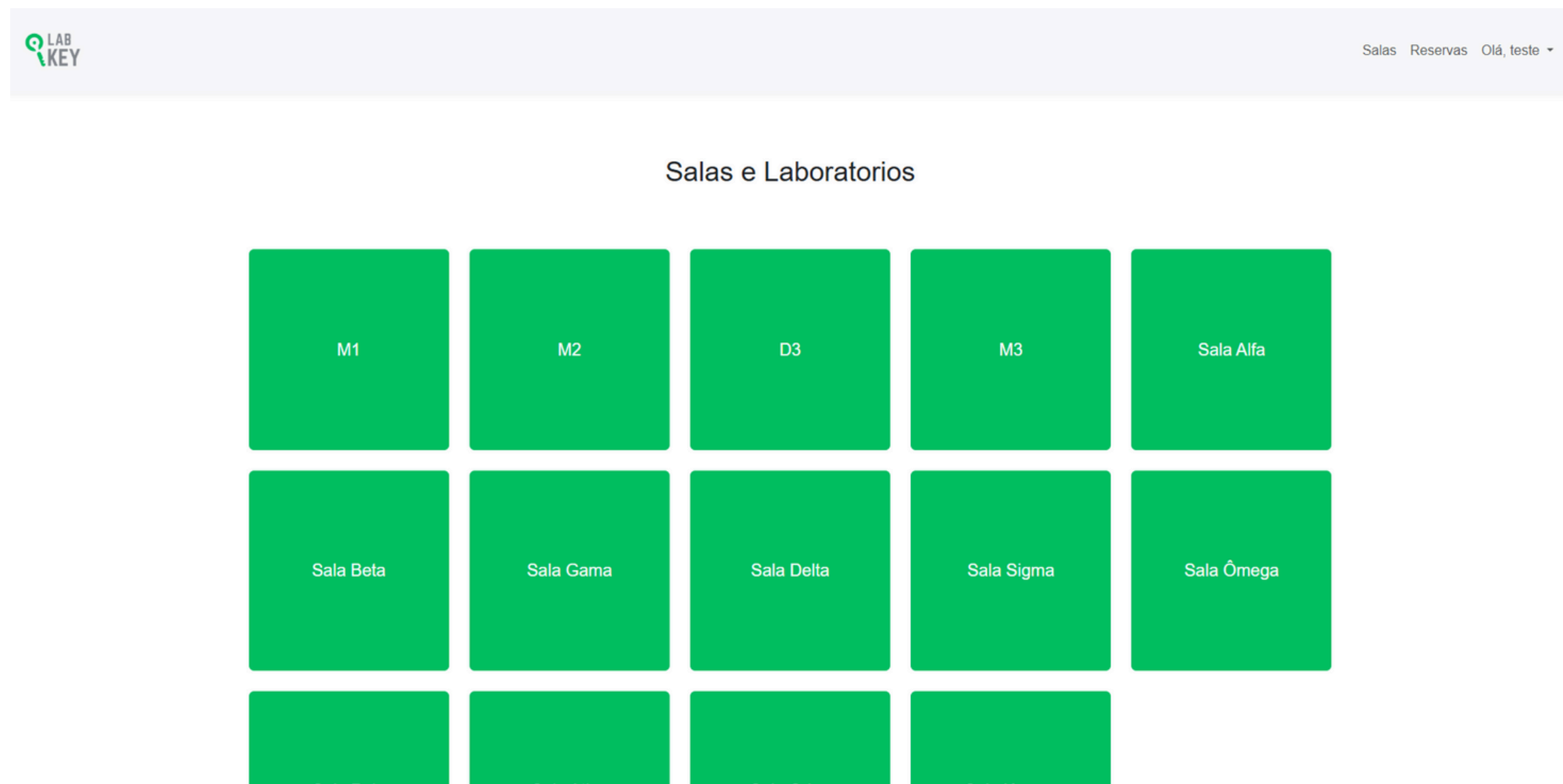
Nova Reserva

Data	Sala	Horário	Situação	Opções
02/01/2026	D3	10:30:00 - 12:00:00	REJEITADA	Ação indisponível
10/12/2025	M1	08:50:00 - 10:20:00	REJEITADA	Ação indisponível
27/01/2025	Sala Delta	07:00:00 - 08:00:00	APROVADA	<div>EditarCancelar</div>
20/01/2025	Sala Alfa	09:00:00 - 11:00:00	PENDENTE	<div>EditarCancelar</div>
12/01/2025	M2	14:00:00 - 15:00:00	APROVADA	<div>EditarCancelar</div>

LabKey © 2024

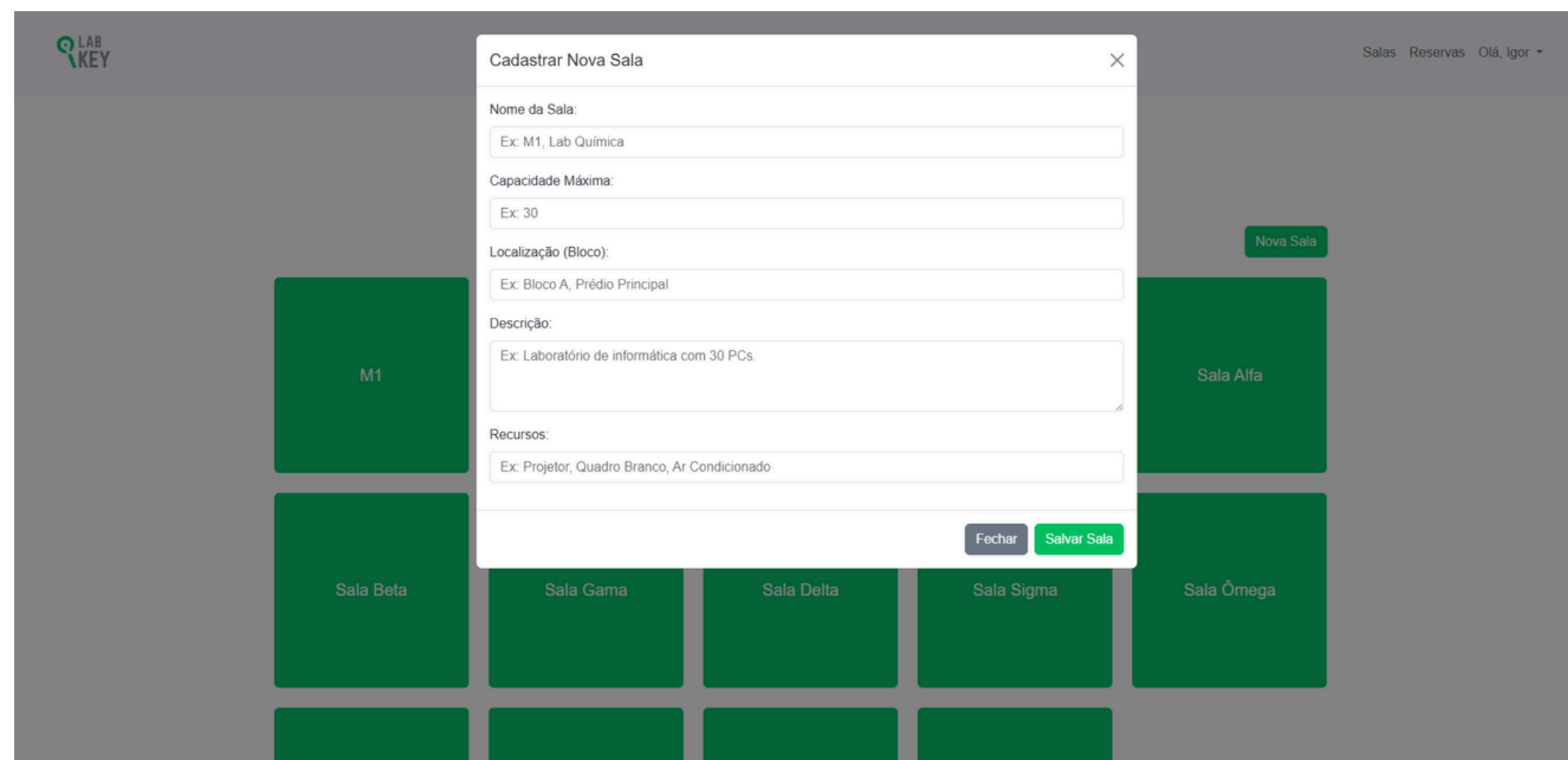
Salas e laboratório

A pagina de sala e laboratórios tem como objetivo exibir as salas cadastradas no sistema, permitindo aos usuários verificar as salas que estão disponíveis.



Cadastro de salas (adm)

A pagina de sala e laboratórios do administrador tem como objetivo exibir as salas para os administradores permitindo o gerenciamento das salas bem como seu cadastro, edição e ou exclusão.



The screenshot shows the 'Cadastrar Nova Sala' (Register New Room) form in the LAB KEY system. The form is overlaid on a dashboard that displays a grid of room cards. The dashboard includes a top navigation bar with the LAB KEY logo, a user profile 'Olá, Igor', and links for 'Salas' and 'Reservas'. The room cards in the background are labeled 'M1', 'Sala Alfa', 'Sala Beta', 'Sala Gama', 'Sala Delta', 'Sala Sigma', and 'Sala Ômega'. A 'Nova Sala' button is visible in the top right of the dashboard.

The 'Cadastrar Nova Sala' form contains the following fields:

- Nome da Sala:** Ex: M1, Lab Química
- Capacidade Máxima:** Ex: 30
- Localização (Bloco):** Ex: Bloco A, Prédio Principal
- Descrição:** Ex: Laboratório de informática com 30 PCs.
- Recursos:** Ex: Projetor, Quadro Branco, Ar Condicionado

At the bottom of the form, there are two buttons: 'Fechar' (Close) and 'Salvar Sala' (Save Room).

Aprovar ou reprovar solicitação (adm)

A pagina de reservas do administrador tem como objetivo exibir todas as reservas feitas pelos usuários, permitindo que eles gerenciem as solicitações feitas pelos usuários, podendo definir o status das reservas, aprovando ou rejeitando a solicitação.

Todas as Reservas

Data	Sala	Horário	Usuário	Situação	Opções
02/01/2026	D3	10:30:00 - 12:00:00	teste teste	REJEITADA	Status Finalizado
10/12/2025	M1	08:50:00 - 10:20:00	teste teste	REJEITADA	Status Finalizado
30/01/2025	Sala Ômega	10:30:00 - 12:00:00	Igor Bruno Arruda Soares Araújo	PENDENTE	Aprovar Rejeitar
28/01/2025	Sala Sigma	16:00:00 - 17:00:00	Administrador Geral	REJEITADA	Status Finalizado
27/01/2025	Sala Delta	07:00:00 - 08:00:00	teste teste	APROVADA	Rejeitar
25/01/2025	Sala Gama	13:00:00 - 15:00:00	Igor Bruno Arruda Soares Araújo	PENDENTE	Aprovar Rejeitar
22/01/2025	Sala Beta	15:00:00 - 16:00:00	Administrador Geral	APROVADA	Rejeitar
20/01/2025	Sala Alfa	09:00:00 - 11:00:00	teste teste	PENDENTE	Aprovar Rejeitar
15/01/2025	M3	10:00:00 - 11:00:00	Igor Bruno Arruda Soares Araújo	CANCELADA	Status Finalizado
14/01/2025	D3	08:00:00 - 09:00:00	Administrador Geral	REJEITADA	Status Finalizado
12/01/2025	M2	14:00:00 - 15:00:00	teste teste	APROVADA	Rejeitar
10/01/2025	M1	09:00:00 - 10:00:00	Igor Bruno Arruda Soares Araújo	PENDENTE	Aprovar Rejeitar

Resultado

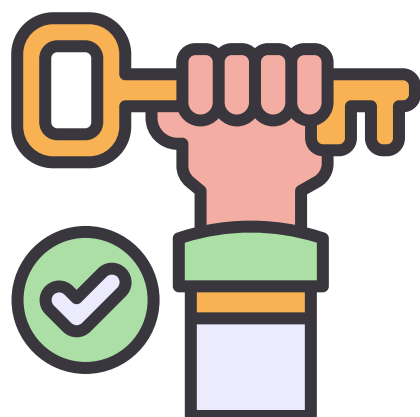
LabKey

Sua solução rápida e eficiente para reserva de salas, laboratórios e recursos. Chega de conflitos de agendamento!

[Comece Agora](#)[Ver Mais](#)

Conclusão

O desenvolvimento do LabKey teve como propósito criar um sistema web capaz de facilitar o gerenciamento de reservas e o controle de acesso a laboratórios e salas. Com o uso de metodologias ágeis e ferramentas atuais, foi possível construir uma solução prática, intuitiva e adequada às necessidades das instituições. O sistema contribui para organizar melhor os agendamentos, centralizar informações e tornar o uso dos espaços mais eficiente e transparente.



**OBRIGADO
PELA ATENÇÃO!**

