



UNIVERSIDADE NOVE DE JULHO - UNINOVE
PROJETO DE DESENVOLVIMENTO DE APLICAÇÕES MULTIPLATAFORMA

EXCLUÍDOS OS DADOS SOBRE OS AUTORES EM ATENDIMENTO A LGPD - LEI
GERAL DE PROTEÇÃO DE DADOS

CONDOMINUS - API PARA GERENCIAR CONDOMÍNIOS

**São
Paulo
2024**

EXCLUÍDOS OS DADOS SOBRE OS AUTORES EM ATENDIMENTO A LGPD - LEI
GERAL DE PROTEÇÃO DE DADOS

CONDOMINUS - API PARA GERENCIAR CONDOMÍNIOS

Trabalho apresentado à Universidade
Nove de Julho - UNINOVE, como parte
dos requisitos obrigatórios para
obtenção do título de Ciência da
Computação

Orientador: Edson Melo De Souza

São

Paulo

2024

Sumário

Codefy	5
História	6
Missão	7
Visão	7
Valores	8
Condominus	9
Arquitetura e Ferramentas utilizadas	9
Api	10
Java	10
Spring Boot	10
Banco de dados	11
Design	11
Postgres e H2	11
Autenticação e Autorização	12
Spring Security	12
JWT	12
Token JWT	13
Header	13
PayLoad	13
Signature	13
Senhas	14
Endpoints Livres	14
Requisições	14
Cadastro	14
Login	14
Documentação da api	15
Desenvolvimento do aplicativo móvel	15
Dart	15
Flutter	15
Provider	15
Consumo de Api	16
Autenticação	16
Telas	17
Login	17
Perfil Síndico	18
Home	18
Moradores	19
Visitantes	20
Perfil Morador	21
Home	22

Reservar	22
Visitantes	23

Codefy

A Codefy é uma empresa de pesquisa e implementação de softwares que está no mercado há precisamente 15 anos. Fundada em 2009 por um grupo de amigos que já estavam cansados de toda a burocracia que era para realizar simples tarefas.

Desde então, a empresa passou por inúmeras mudanças para que hoje em dia ela possa se chamar "Codefy" e ser uma das melhores desenvolvedoras de softwares personalizados.

Nossa empresa é conhecida por esta logo:



História

Com a pandemia, ocorreu uma grande mudança no cenário mundial que rapidamente influenciou na forma como as pessoas trabalham e se comunicam, impulsionando ainda mais a necessidade de soluções virtuais. E foi a partir daí que as portas se abriram para empresas como a Codefy expandirem e se destacarem no mercado, atendendo à crescente demanda por serviços online.

Os primeiros anos foram desafiadores, como costuma ser para qualquer empreendimento em sua fase inicial. A Codefy teve de enfrentar a árdua tarefa de conquistar confiança no mercado de um segmento altamente competitivo. No entanto, graças a uma abordagem centrada no cliente e um compromisso inabalável com a excelência, a empresa começou a se destacar.

Ao longo de uma década e meia que se seguiram, a Codefy experienciou um crescimento constante. Expandiu sua equipe, refinou seus processos e adotou tecnologias de ponta para atender de forma ainda mais eficiente e precisa às demandas em constante evolução.

Hoje, após 15 anos de dedicação incansável à sua missão, a Codefy é reconhecida como uma das principais referências em gestão de implementação de softwares não somente no mercado baiano, mas também por todo o Brasil. Seu compromisso inabalável com a segurança, transparência, inovação e responsabilidade continua a impulsionar seu sucesso e a solidificar sua reputação como um parceiro confiável e de confiança para seus clientes e comunidades atendidas.

O grupo possui duas sedes, uma localizada na grande São Paulo, mais precisamente no centro, e outra em Xique-Xique, Bahia (cidade natal do CEO da Codefy, Igor Alves).

Estamos sempre nos atualizando das novidades no mercado, então nossos equipamentos são da mais recente tecnologia para assim podermos oferecer o melhor atendimento aos nossos clientes.

Tudo isso só é possível ser feito graças a equipe preparada que temos, que está 100% envolvida com a causa e empenhada em solucionar problemas, construir esquemas elaborados, fazer pesquisas e saneamentos de dados e, acima de tudo, garantir a segurança dos usuários.

Missão

O grupo de pesquisa e implementação de softwares Codefy oferece serviços voltados à otimização da gestão do dia-a-dia. Temos como foco, transformar tarefas demoradas e extensivas, em algo feito com apenas um clique, graças aos aplicativos que desenvolvemos. Obtendo maior produtividade aos nossos clientes.

Na Codefy, comprometemo-nos a fornecer soluções abrangentes e personalizadas para a gestão de qualquer convivência. Com um foco claro na transparência, qualidade e satisfação do cliente. Nosso objetivo é criar um ambiente seguro, harmonioso e bem administrado para o público de cada aplicativo que administramos, garantindo que todas as necessidades e preocupações sejam atendidas de maneira eficiente e profissional.

Além disso, buscamos promover o desenvolvimento profissional e pessoal de nossa equipe, capacitando-os para oferecer serviços de excelência e se adaptar às demandas em constante evolução do setor de implementação de softwares. Ao fazer isso, nossa meta não é ser apenas um provedor de serviços, mas um parceiro confiável e de confiança para nossos clientes, ajudando-os a alcançar seus objetivos e melhorar sua qualidade de vida em cada um de seus ambientes de convívio.

Visão

Nós aspiramos a consolidar nossa posição com o foco na desenvoltura de softwares, através da nossa dedicação contínua à excelência e inovação. Visualizamos um futuro onde cada software sob nossa administração se destaca não apenas pela eficiência operacional, mas também pelo alto padrão de qualidade oferecido aos nossos clientes.

Nosso objetivo é não apenas atender, mas superar as expectativas de nossos clientes, estabelecendo novos padrões de excelência na gestão, e construindo relacionamentos sólidos e duradouros com nossos parceiros e comunidades locais. Acreditamos que, ao alcançarmos nossa visão, estaremos contribuindo

significativamente para o desenvolvimento sustentável e o bem-estar das comunidades em que atuamos.

Valores

Na Codefy, nossas ações são guiadas por valores fundamentais que moldam nossa cultura e nossas operações.

Segurança é nossa prioridade máxima. Não apenas implementamos medidas físicas, mas também garantimos que cada decisão seja tomada com a segurança em mente. Investimos em tecnologias de ponta para monitoramento e controle de acesso, garantindo um ambiente seguro para todos os envolvidos.

Operamos em um ambiente de total transparência, onde todas as informações são compartilhadas abertamente e as comunicações são claras e acessíveis. Desde decisões administrativas até questões financeiras, comprometemo-nos a fornecer informações acessíveis a todos os envolvidos.

Abraçamos a inovação como um meio de melhorar continuamente nossos serviços e superar as expectativas dos clientes. Estamos constantemente explorando novas tecnologias e incentivando a criatividade em nossa equipe.

Assumimos total responsabilidade por nossas ações e decisões, honrando nossos compromissos e agindo com integridade em todas as situações. Para nós, responsabilidade é mais do que uma palavra - é um princípio fundamental que guia todas as nossas interações e operações. Estamos abertos ao feedback e comprometidos em aprender e crescer continuamente como organização.

Condominus

A Codefy foi abordada com o desafio de solucionar algumas problemáticas de um condomínio popular que possuía constantes conflitos por conta do atraso em que as notícias eram repassadas e da falta de pagamentos dentro do prazo. E com isso foi criada uma aplicação chamada “Condominus”.

Condominus é a nossa mais nova solução, que veio justamente para facilitar a comunicação existente entre Síndico (a) – Morador e Morador – Morador e, de quebra, ainda irá te informar das notícias mais importantes do dia. Com sua tecnologia inovadora, nós conseguimos simplificar a gestão de condomínios de forma rápida e eficiente.

Ademais, a interface intuitiva e as funcionalidades abrangentes do aplicativo, permitem que os moradores façam reservas das áreas livres do condomínio, com a antecedência necessária para não gerar mais conflitos. Ela também está encarregada de ajudar nas comunicações do síndico para os condôminos, como por exemplo, informar de possíveis manutenções dentro do condomínio e outras notícias. Agora, os moradores terão os seus visitantes cadastrados pelo síndico, garantindo uma maior segurança a todos.

Arquitetura e Ferramentas utilizadas

O modelo de nosso projeto segue os padrões web, sendo dividido em duas partes distintas: o frontend e o backend. Para garantir uma arquitetura sólida e modular, implementamos padrões adicionais como MVC (Model-View-Controller) e SOLID. Estes padrões promovem uma separação clara de responsabilidades e facilitam a manutenção e evolução do código ao longo do tempo.

No backend, utilizamos o Spring Boot como framework principal, aproveitando sua robustez e eficiência para desenvolver uma API poderosa e escalável. O Spring Boot nos proporciona um ambiente de desenvolvimento produtivo, além de integrar-se perfeitamente com outras ferramentas e frameworks.

Já no frontend, optamos pelo Flutter, um framework de desenvolvimento de interfaces de usuário multiplataforma, que nos permite criar aplicativos para iOS, Android e web a partir de um único código-fonte. O Flutter oferece uma experiência de desenvolvimento fluida e consistente, além de proporcionar uma aparência visual atraente e responsiva para os nossos usuários.

Por fim, é importante ressaltar que nosso projeto se baseia em uma arquitetura de API (Application Programming Interface), o que significa que nossa aplicação oferece serviços e funcionalidades através de uma interface padronizada, permitindo a integração com outros sistemas e aplicativos de forma eficiente e segura.

Api

O backend consiste na integração de usuários com os dados da aplicação e claro de forma segura, restrita e totalmente monitorada. Para a criação de tal ferramenta foi escolhida uma das linguagens mais conhecidas e utilizadas no mundo “Java” juntamente com o Framework Spring boot.

Java

A robustez, segurança e vasta gama de bibliotecas oferecidas pelo Java foram os pilares fundamentais na implementação do back-end deste aplicativo. Ao optarmos pelo Java, priorizamos não apenas a solidez do sistema, mas também a sua proteção contra vulnerabilidades, garantindo assim uma experiência confiável e segura aos usuários. Além disso, a riqueza de bibliotecas disponíveis facilitou significativamente o desenvolvimento do aplicativo, proporcionando soluções eficazes e eficientes para os desafios enfrentados durante o processo de criação.

Spring Boot

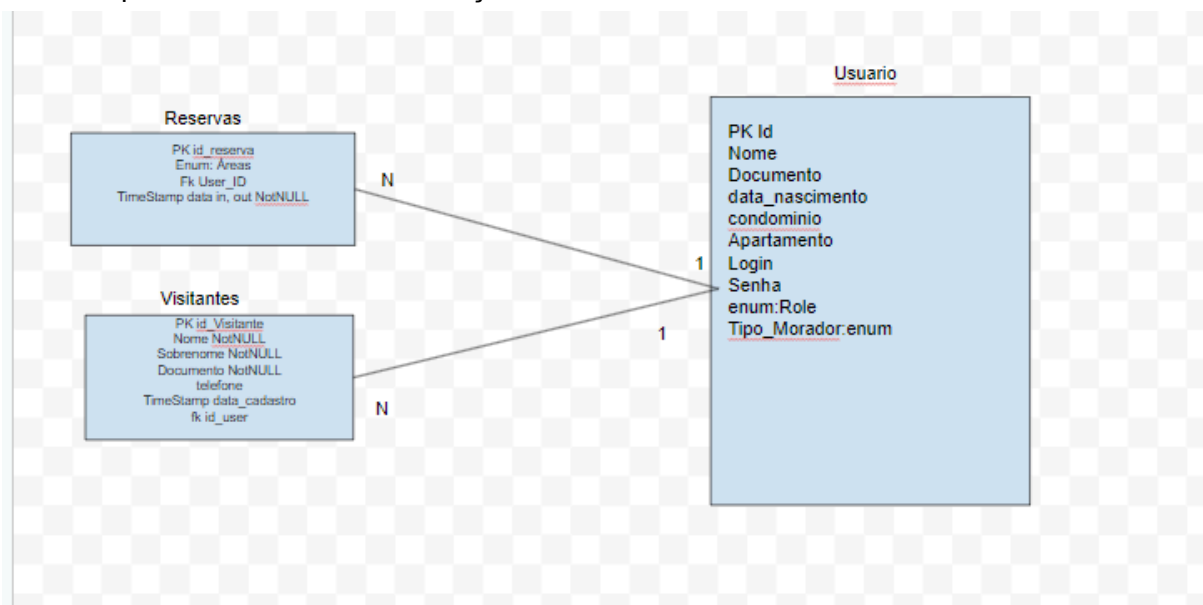
Utilizando o Spring Boot, simplificamos radicalmente o processo de criação e configuração deste aplicativo. Ele oferece um ambiente de desenvolvimento altamente produtivo, automatizando as configurações e fornecendo uma extensa variedade de funcionalidades integradas. Com isso, conseguimos focar nossa energia no desenvolvimento do próprio aplicativo, sem a necessidade de nos preocuparmos com detalhes de configuração tediosos. Essa abordagem não apenas acelerou o processo de desenvolvimento, mas também garantiu uma base sólida e confiável para a aplicação, permitindo que entregássemos um produto de alta qualidade de forma mais eficiente.

Banco de dados

Para o armazenamento de dados, contamos com as poderosas ferramentas oferecidas pelo Spring Boot, como o Spring Data JPA. Essa tecnologia não apenas simplifica a persistência de dados, mas também facilita a consulta e a atualização dos mesmos. Com o Spring Data JPA, pudemos abstrair boa parte do código relacionado ao acesso ao banco de dados, o que resultou em um desenvolvimento mais ágil e eficiente. Além disso, o uso do JPA nos permitiu trabalhar com uma camada de persistência de alto nível, proporcionando maior flexibilidade e facilidade de manutenção ao lidar com o armazenamento e recuperação de dados. Essas ferramentas foram essenciais para garantir a robustez e a confiabilidade do sistema de armazenamento de nosso aplicativo.

Design

Abaixo é possível observar um esboço visual da estrutura do nosso banco de dados.



Postgres e H2

Para o banco de dados em ambiente de produção, optamos pelo PostgreSQL devido à sua robustez e facilidade de armazenamento. O PostgreSQL é uma escolha confiável para ambientes de produção, oferecendo recursos avançados de segurança, escalabilidade e desempenho. Sua capacidade de lidar com cargas de trabalho intensas e sua ampla compatibilidade com padrões SQL tornam-no uma escolha ideal para aplicativos de missão crítica. Além disso, sua comunidade ativa e

o suporte contínuo garantem que possamos contar com atualizações regulares e correções de segurança. Esses fatores combinados nos proporcionaram um ambiente de banco de dados sólido e confiável para suportar as necessidades do nosso aplicativo em produção.

O H2, por outro lado, foi utilizado em ambiente de desenvolvimento e testes devido à sua natureza leve e à facilidade de configuração. É uma opção excelente para cenários em que é necessário um banco de dados temporários.

Autenticação e Autorização

Spring Security

Através da integração do Spring Boot com o Spring Security, implementamos uma camada robusta de autenticação e autorização para nossas requisições. Utilizando filtros, conseguimos capturar todas as requisições e aplicar políticas de acesso de forma granular, garantindo que apenas usuários autorizados possam acessar e manipular as informações em nosso banco de dados.

O Spring Security nos oferece uma ampla gama de recursos para lidar com autenticação e autorização, desde a configuração de regras de acesso baseadas em papéis de usuário até a implementação de autenticação multifator avançada. Isso nos permite personalizar totalmente o comportamento de segurança de nossa aplicação de acordo com nossos requisitos específicos.

Com essa abordagem, podemos garantir a integridade e a confidencialidade dos dados em nosso sistema, protegendo-os contra acessos não autorizados e mantendo a conformidade com os requisitos de segurança da nossa aplicação.

JWT

A implementação do JWT (JSON Web Token) com o Auth0 representa uma medida crucial de segurança em nosso aplicativo. O objetivo principal do JWT é proteger os dados de autenticação durante a comunicação entre o cliente e o servidor. Ao utilizar o JWT, podemos criptografar os dados utilizando um segredo armazenado no servidor, gerando assim um token de 3 partes.

Este token é composto por um cabeçalho, um payload e uma assinatura, garantindo a integridade e a autenticidade dos dados transmitidos. Mesmo que uma pessoa mal-intencionada intercepte o token, ela não será capaz de decifrar ou manipular as informações contidas nele, devido à sua natureza criptografada e à necessidade do segredo para verificar sua autenticidade.

Essa abordagem nos permite implementar um sistema de autenticação seguro e confiável, protegendo as credenciais dos usuários e garantindo que apenas usuários autorizados tenham acesso aos recursos protegidos de nossa aplicação. Além disso, ao utilizar o Auth0 como provedor de identidade, podemos aproveitar seus recursos avançados de gerenciamento de identidade e autenticação, simplificando ainda mais o processo de desenvolvimento e garantindo uma experiência segura para nossos usuários.

Token JWT

O token JWT é gerado por 3 partes:

Header

Que é responsável por especificar o tipo de token no caso JWT e o algoritmo de geração utilizado que no nosso caso foi utilizado o HS256

PayLoad

Aqui irá se encontrar o conteúdo transitado no nosso caso um Json com as credenciais do usuário.

Signature

A Parte final do token é responsável por identificar quem foi o responsável por gerar esse token para evitar tokens duplicados. A assinatura é feita com uma chave gerada pelo servidor.

Senhas

Por boas práticas as senhas estão sendo salvas no nosso banco de dados de forma encriptadas pelo algoritmo Bcrypt que gera um hash e esse hash de fato é salvo no nosso banco, além do próprio Bcrypt fazer o hash e comparar senhas em solicitações.

Endpoints Livres

Para que um usuário possa se cadastrar e logar os endpoints de /register e /login foram registrados como livres de autenticação ou autorização.

Requisições

O servidor agora com o devidamente configurado para cada requisição que deverá retornar dados que precisem de autenticação ele irá esperar receber no header da requisição um campo chamado Authorization com o valor sendo o token gerado pelo servidor.

Cadastro

Como dito anteriormente o /register estará livre logo qualquer um sem autenticação poderá se cadastrar em nossa aplicação com isso ele irá obter um login e senha caso não possua cadastro e assim futuramente irá utilizar estes dados para se logar na aplicação.

Login

Com login e senha adquiridos agora o usuário irá poder se logar, sendo assim nosso servidor tem a função de validar se o login e a senha encriptada dele se encontra em nosso banco de dados caso seja verdadeiro o servidor irá retornar um Token JWT.

Documentação da api

A documentação do nosso aplicativo backend foi elaborada utilizando o framework Swagger, cujo propósito é documentar e identificar os endpoints disponíveis em nossa aplicação.

Desenvolvimento do aplicativo móvel

A construção do nosso aplicativo móvel, também conhecido como o frontend do projeto, foi realizada utilizando o framework Flutter, que utiliza a linguagem de programação Dart. Neste processo, foram aplicados diversos padrões, frameworks e bibliotecas para a criação das interfaces visuais.

Dart

O aplicativo móvel foi todo escrito na linguagem dart pois ela oferece uma grande versatilidade e muitas ferramentas para o desenvolvimento de requisições através de suas futuras e funções assíncronas com o dar todo o comportamento do nosso aplicativo desenvolvido. um dos pontos que destacam o dart é a utilização de parâmetros nomeados que possibilitam a construção de de frameworks como flutter.

Flutter

A escolha pelo flutter foi quase unanimidade por questões como ferramentas prontas e a facilidade de desenvolvimento de telas. Esse framework mantido pela empresa google é incrivelmente eficiente quando se trata de performance e usabilidades, com o flutter todas nossas telas foram desenvolvidas, testadas e aplicadas.

Provider

O flutter entrega seus widgets para construção de telas e um widget específico o stateful garante serve como a principal ferramenta para o controle de estado de uma tela porém ela apresenta uma falha, conforme mais telas vão sendo adicionadas a complexidade começa a aumentar até certo momento onde não é mais possível entender o código, é uma péssima prática para isso frameworks como o provider existem.

O provider implementa um padrão muito conhecido na web “Observer” e com isso ele notifica as telas que precisam ser atualizadas e reconstruídas a simplicidade e a performance que o código é desenvolvido com o provider se sobressai muito em relação ao stateful e por isso ele foi escolhido para ser o nosso controlador de estado.

Consumo de Api

A construção do nosso aplicativo móvel, também conhecido como o frontend do projeto, foi realizada utilizando o framework Flutter, que utiliza a linguagem Utilizamos a biblioteca "http" do Dart para consumir todos os endpoints fornecidos pelo nosso servidor via API em Spring Boot. Os dados foram fornecidos no formato JSON, o que exigiu o uso adicional da biblioteca "convert" para decodificar os dados JSON e transformá-los em objetos Dart utilizáveis.

Graças a essa abordagem, conseguimos efetivamente consumir todos os dados necessários para nosso aplicativo Flutter diretamente da API. Isso garantiu uma integração suave entre o frontend e o backend, permitindo que nossa aplicação oferecesse uma experiência de usuário coesa e dinâmica.e programação Dart. Neste processo, foram aplicados diversos padrões, frameworks e bibliotecas para a criação das interfaces visuais.

Autenticação

No Dart assim que o usuário fizer o login o token será salvo dentro do provider e quando necessário fazer algum tipo de requisição ele será enviado junto do header da requisição para que o servidor identifique o token e valide a caso seja um recurso que pode ser acessado pelo nível daquele usuário o servidor irá responder com as informações.

Telas

Login

A tela de login tem a funcionalidade de autenticar o usuário e armazenar o token gerado dentro do “Provider”.



The image shows a login screen for an application named "CONDOMINUS". The background is a solid blue color. At the top center, there is a white icon of a building with a circular arrow around it, representing a community or a cycle. Below the icon, the word "CONDOMINUS" is written in a bold, white, sans-serif font. In the center of the screen, there is a white rounded rectangle containing the login form. Inside this rectangle, the word "Login" is written in a small, white, sans-serif font, followed by a horizontal line. Below this, the word "Senha" is written in a small, white, sans-serif font, followed by another horizontal line. At the bottom of the rounded rectangle, there is a dark blue button with the word "Entrar" written in white, sans-serif font.

Perfil Síndico

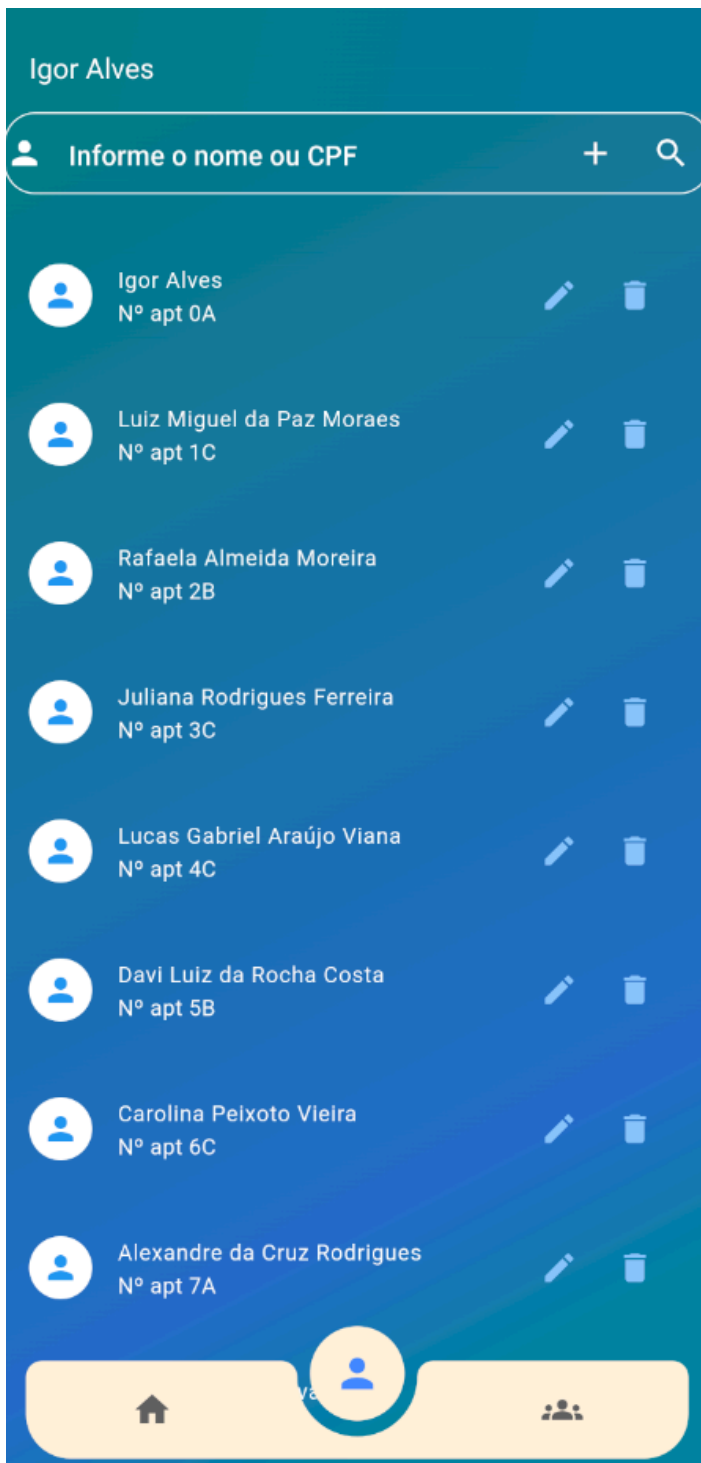
Home

Conta apenas com o logo do aplicativo



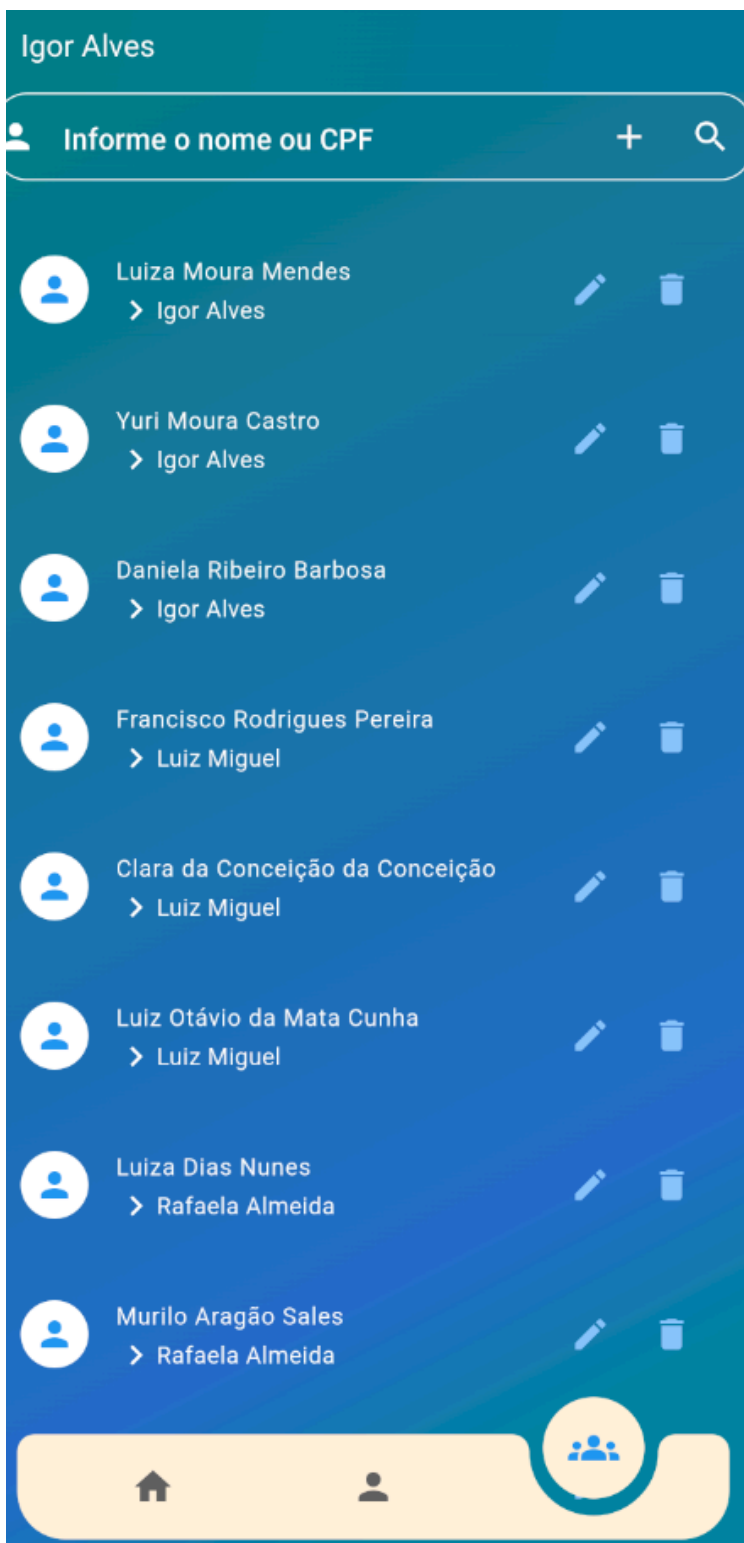
Moradores

Nesta tela o síndico será capaz de criar deletar e buscar moradores:



Visitantes

Nesta tela o síndico será capaz de criar deletar e buscar visitantes:



Perfil Morador

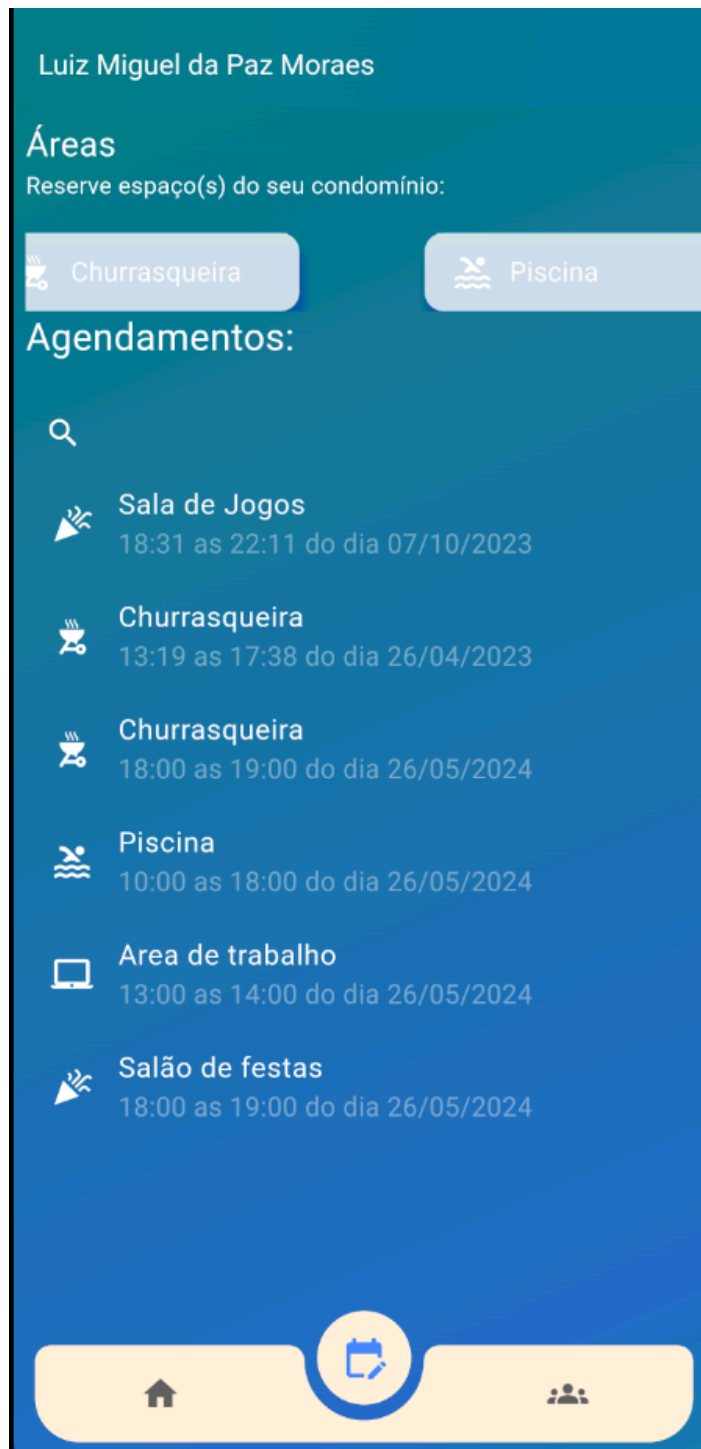
Home

A tela de home do morador conta com um carrossel onde tem como objetivo mostrar as principais informações do dia do condomínio como manutenções, reuniões etc.



Reservas

Nesta tela os moradores têm a possibilidade de agendar áreas comuns do condomínio:



Visitantes

Nesta tela o morador poderá visualizar seus visitantes:

