



Etec
Adolpho Berezin
Mongaguá



SÃO PAULO
GOVERNO DO ESTADO

Adryan Alexander Santana da Silva

Edkarllos Fernando Almeida da Silva

Lucas Pereira Batista

Nicholas Balbino Souza

Pedro Henrique Oliveira Lopes

SISTEMA DE GESTÃO DE CONDOMÍNIOS - CONDOVIEW

Equipe Indominus

Orientadores Profa. Graciete Henriques dos Santos

e Prof. Alexandre Marchiori de Almeida

Mongaguá

12/2024

Adryan Alexander Santana da Silva
Edkarllos Fernando Almeida da Silva
Lucas Pereira Batista
Nicholas Balbino Souza
Pedro Henrique Oliveira Lopes

SISTEMA DE GESTÃO DE CONDOMÍNIOS - CONDOVIEW
Equipe Indominus

Trabalho de Conclusão de Curso apresentado à Escola Técnica Adolpho Berezin, como parte dos requisitos para a obtenção do título de Técnico em Desenvolvimento de Sistemas.
Orientadores Profa.: Graciete Henriques dos Santos e Prof. Alexandre Marchiori de Almeida

Mongaguá
12/2024

Dedicamos este trabalho às nossas famílias e amigos, cujo apoio e incentivo foram fundamentais durante toda a jornada acadêmica.

AGRADECIMENTOS

Agradecemos aos nossos orientadores pela orientação e suporte ao longo deste projeto. Também gostaríamos de agradecer aos professores e colegas que nos ajudaram com feedback valioso e encorajamento.

*"Inovação é a habilidade de ver mudança
como uma oportunidade, não uma ameaça."*

Steve Jobs

RESUMO

O Sistema de Gestão de Condomínios proposto é uma plataforma destinada a facilitar a administração de condomínios residenciais. Nosso projeto busca oferecer várias funcionalidades, incluindo cadastro de moradores, gestão de áreas comuns, comunicação interna por meio de chats, gestão financeira e manutenção do prédio. Por meio de uma interface intuitiva, o sistema visa simplificar os processos de um condomínio, promovendo principalmente transparência e eficiência.

PALAVRAS-CHAVES: Condomínios, Gestão Residencial, Plataforma, Eficiência Administrativa

ABSTRACT

The proposed Condominium Management System is a platform designed to facilitate the administration of residential condominiums. Our project seeks to offer several functionalities, including resident registration, management of common areas, internal communication through chats, financial management and building maintenance. Through an intuitive interface, the system aims to simplify condominium processes, mainly promoting transparency and efficiency.

Key Words: Condominiums, Residential Management, Platform, Administrative Efficiency

LISTA DE FIGURAS

Figura 1 - Logo da Equipe	20
Figura 2 - Logo do Projeto	20
Figura 3 - Diagrama de Casos de Uso	40
Figura 4 - Diagrama de Classes	1
Figura 5 - Modelagem de Banco de Dados	1
Figura 6 - Wireframe Tela Cadastrar condomínio	6
Figura 7 - Wireframe Tela de alerta de e-mail não cadastrado	7
Figura 8 - Wireframe Tela menu	8
Figura 9 - Wireframe Tela avisos	9
Figura 10 - Wireframe Tela dados Reservas	10
Figura 11 - Wireframe Tela Ocorrência	11
Figura 12 - Wireframe Tela Assembleia	12
Figura 13 - Wireframe Tela Visitante	13
Figura 14 - Wireframe Tela de Despesas	14
Figura 15 - Wireframe Tela de Encomendas	15
Figura 16 - Wireframe Tela de Conversas	16
Figura 17 - Tela criar condomínio	17
Figura 18 - Tela de cadastro	18
Figura 19 - Tela login	19
Figura 20 - Tela perfil adm	20
Figura 21 - Tela configuração	21
Figura 22 - Menu adm	22
Figura 23 - Tela de despesas	23
Figura 24 - Tela chat geral	24
Figura 25 - Tela de ocorrência	25
Figura 26 - Tela de Manutenção	26
Figura 27 - Tela de reserva	27
Figura 28 - Tela de assembleias	28
Figura 29 - Tela encomendas detalhes	29
Figura 30 - Tela encomendas geral	30
Figura 31 - Tela inicial adm	31
Figura 32 - Menu segunda parte	32

Figura 33 - Tela conversas	33
Figura 34 - Informações do condomínio	34
Figura 35 – Histórico.....	35
Figura 36 – Notificações	36
Figura 37 - Tela visitante	37
Figura 38 – Avisos	38

SUMÁRIO

LISTA DE FIGURAS.....	7
INTRODUÇÃO.....	11
1. NICHOS DE MERCADO DO PROJETO.....	12
1.1 Público-alvo – Administradores de condomínios.....	12
1.2 Problema do Cliente.....	13
1.3 Solução Proposta.....	13
2. TECNOLOGIAS UTILIZADAS.....	15
2.1 HTML - HyperText Markup Language.....	15
2.2 CSS - Cascading Style Sheet.....	15
2.3 – JavaScript.....	15
2.4 – Vue JS.....	16
2.5 - C Sharp.....	16
2.6 – ASP.NET.....	16
2.7 – React Native.....	16
2.8 – Bootstrap.....	17
2.9 – Figma.....	17
2.10 – Canva.....	17
2.11 – MongoDB.....	17
3. EMPRESA.....	19
3.1 Missão.....	19
3.2 Visão.....	19
3.3 Valores.....	19
3.4 Logos.....	20
3.4.1 Logo da Equipe.....	20
3.4.2 Logo do Projeto.....	20
3.4.3 Slogan da Empresa e do Projeto.....	20

4.ANÁLISE.....	21
4.1. Documento Visão	21
4.2. Descrições de Casos de Uso	28
4.3. Diagrama de Casos de Uso	40
4.4. Diagrama de Classes	1
4.5. MER – Modelo do Banco de Dados	1
4.6. Create do Banco de Dados	1
4.7 Principais Selects do Banco de Dados.....	5
4.8. Wireframe das Telas	6
4.9 Prints das Telas.....	17
4.10 Trecho do Código Fonte.....	39
4. MANUAL DO USUÁRIO	45
CONCLUSÃO	49

INTRODUÇÃO

Nos últimos anos, o aumento da urbanização tem levado ao crescimento no número de condomínios residenciais.

A média de aumento entre os condomínios analisados foi de 8% em abril deste ano, em comparação ao mesmo mês de 2022, superando o IPCA do período, que foi 4,18%. Em média, os condomínios da cidade subiram quase o dobro do que subiu a inflação, mas há casos ainda mais gritantes. (TERRA, 2023)

No Brasil, os condomínios desempenham um papel fundamental na vida das pessoas, oferecendo opções variadas para moradia e negócios.

Com esse aumento surge uma necessidade de uma gestão mais eficiente desses ambientes, melhorando a qualidade de vida de moradores e permitindo uma gestão mais eficiente para síndicos e administradores.

Atualmente, a maioria dos condomínios ainda utilizam métodos tradicionais de gestão como planilhas e não tem um sistema de comunicação próprio, o que pode gerar lentidão nos processos e falta de clareza nas informações.

A proposta desse trabalho é desenvolver uma plataforma de Sistema de Gestão de Condomínios, uma plataforma que tem como objetivo integrar funcionalidades que contribuam para uma gestão mais eficiente e transparente.

Nosso objetivo ao desenvolver esse trabalho será oferecer uma solução capaz de: otimizar os processos de administração, melhorar a comunicação entre moradores e administradores, garantir a transparência na gestão financeira e facilitar o acompanhamento da manutenção do prédio.

1. NICHOS DE MERCADO DO PROJETO

O nicho de mercado do projeto se concentra no setor de gestão de condomínios residenciais no Brasil. Este segmento é escolhido por sua abrangência e representatividade, abarcando aproximadamente 80% do total de condomínios no país.

Os condomínios residenciais são os mais comuns no Brasil, representando cerca de 80% do total. Os condomínios comerciais representam cerca de 15%, e os condomínios mistos, que incluem unidades residenciais e comerciais, representam cerca de 5%. (ALCOWEB, 2024).

Condomínios residenciais são edificações planejadas e construídas para serem utilizadas como moradia, compostas por unidades autônomas devidamente identificadas, com áreas de uso comum, pertencentes a diversos proprietários.

O condomínio residencial é aquele tipo de condomínio planejado e construído para ser utilizado como moradia. Trata-se de uma edificação ou um conjunto de edificações destinado ao uso habitacional, construído sob forma de unidades autônomas devidamente identificadas, com áreas de uso comum, pertencentes a diversos proprietários (UNIFORT, 2016).

Os condomínios residenciais são caracterizados pela diversidade de perfis de moradores e pela necessidade de uma gestão eficiente para garantir o bem-estar e a qualidade de vida dos residentes.

Este segmento apresenta desafios e oportunidades significativas, incluindo a necessidade de otimização de processos administrativos, melhorias na comunicação entre moradores e administradores, e garantia de transparência na gestão financeira.

O número de condomínios e de pessoas que habitam esses lugares vem crescendo nos últimos anos. Nosso nicho consiste nesse tipo de mercado.

De acordo com Pedro Matos, diretor da Valore Imóveis, essa tendência de crescimento que vem sendo observada desde junho de 2020 ainda deve perdurar por um bom tempo" (Matos, 2021 apud Gomides, 2021).

1.1 Público-alvo – Administradores de condomínios

Público-alvo: Síndicos, membros do conselho administrativo, empresas especializadas em administração de condomínios e moradores de condomínios residenciais.

Nosso público-alvo são administradores de condomínios, este grupo engloba os profissionais especializados na gestão e administração de condomínios residenciais, como síndicos, membros do conselho administrativo e empresas

especializadas nesse tipo de serviço. Esses compartilham o interesse comum em simplificar e melhorar a administração condominial. Além deles, moradores do condomínio também são englobados no público-alvo que procuram ter uma experiência positiva e simplificada ao interagir com a administração do condomínio.

Os domicílios do tipo “casa de vila ou em condomínio”, que em 2010 abrigavam 1,6% das pessoas residentes no Brasil, passou a abrigar 2,4% em 2022. Dessa forma, em conjunto, os tipos “casa” e “casa de vila ou em condomínio” reuniam 87,2% da população. (AGÊNCIA IBGE NOTÍCIAS, 2022).

A gestão de condomínios residenciais tem se tornado uma preocupação crescente devido ao aumento dos condomínios, refletindo a necessidade de soluções eficazes para simplificar e otimizar esses processos.

Através do Aplicativo, buscamos proporcionar uma experiência positiva e simplificada para todos os usuários, promovendo uma administração condominial mais eficiente e colaborativa.

1.2 Problema do Cliente

As relações condominiais são fundamentalmente complexas. Isso porque a convivência, dentro de um mesmo espaço, entre pessoas com hábitos, valores, princípios e rotinas diferentes pode ser equiparado a um barril de pólvora, cuja explosão é fácil, rápida e catastrófica. (Neves ; Ferreira, 2023).

Por um condomínio abrigar diversos tipos de pessoas, é necessário estabelecer uma boa relação entre os moradores e os condôminos, pois qualquer brecha ou dupla interpretação dos comunicados podem gerar desentendimentos.

Por conta da possibilidade de haver desavenças, algumas pessoas se abstêm de criticarem a atual situação do condômino, pois "mesmo não concordando com os métodos e o modo como o edifício está sendo gerenciado, preferem evitar atritos, guardando para si suas opiniões e não participando ativamente das decisões". (SCHWARTZ, 2017). Com um sistema de comunicação anônima, os moradores poderão expressar suas opiniões e ideias para a gestão condominial sem se preocuparem em gerar brigas e intrigas.

1.3 Solução Proposta

A plataforma visa atender às necessidades da gestão condominial e dos moradores. Ao implementar um espaço para a comunicação entre gestores com os moradores. Além disso, a plataforma fornecerá todas as informações e ferramentas

essenciais para um controle financeiro mais preciso, garantindo uma administração mais eficiente dos recursos do condomínio, há também a possibilidade de publicar avisos referentes ao estado do condômino, alertando sobre os problemas.

Outro ponto relevante é a funcionalidade que permite a publicação de avisos e comunicados pertinentes ao estado do condomínio, proporcionando aos moradores informações importantes sobre eventuais problemas ou manutenções necessárias. Dessa forma, a plataforma não apenas facilita a gestão do condomínio, mas também promove uma maior transparência e participação por parte dos condôminos, fortalecendo o senso de comunidade e colaboração.

2. TECNOLOGIAS UTILIZADAS

2.1 HTML - HyperText Markup Language

HTML (HyperText Markup Language) é a linguagem de marcação padrão usada para criar páginas da web. Ela define a estrutura e o layout de uma página, permitindo que os desenvolvedores organizem o conteúdo de forma semântica. Composta por uma série de elementos e tags, o HTML permite a criação de textos, imagens, links e outros tipos de mídia.

O HTML não é considerado uma linguagem de programação, já que ele não pode criar funcionalidades dinâmicas. Ao invés disso, com o HTML, os usuários podem criar e estruturar seções, parágrafos e links usando elementos, tags e atributos. (Longen, 2023).

2.2 CSS - Cascading Style Sheet

CSS (Cascading Style Sheets) é uma linguagem de estilo utilizada em conjunto com o HTML para controlar a apresentação visual de páginas da web e fazer estilizações. Com o CSS, os desenvolvedores podem definir o layout, cores, fontes, espaçamentos e outros aspectos visuais de um site.

CSS é chamado de linguagem Cascading Style Sheet e é usado para estilizar elementos escritos em uma linguagem de marcação como HTML. O CSS separa o conteúdo da representação visual do site. Pense na decoração da sua página. Utilizando o CSS é possível alterar a cor do texto e do fundo, fonte e espaçamento entre parágrafos. Também pode criar tabelas, usar variações de layouts, ajustar imagens para suas respectivas telas e assim por diante. (Golçalves, 2022).

O Ambiente de Desenvolvimento Integrado (IDE) selecionado para a elaboração deste projeto é o Visual Studio Code. Esta escolha é baseada por ser uma ferramenta amplamente utilizada por desenvolvedores, por sua versatilidade e robustez e oferecer uma ampla gama de funcionalidades que atendem as necessidades desse projeto.

2.3 – JavaScript

JavaScript é uma linguagem de programação que permite que você implemente itens complexos em páginas web. Por exemplo, toda vez que uma página da web faz mais do que simplesmente mostrar a você informação estática, mostrando conteúdo que se atualiza em um intervalo de tempo, isso é uma das funções que o JavaScript exerce.

O JavaScript é uma linguagem de programação de alto nível voltada para o desenvolvimento web, criada originalmente para funcionar do lado do usuário, ou seja, nos navegadores. Junto do HTML e do CSS, é uma das principais tecnologias da web, permitindo a criação de páginas interativas com elementos dinâmicos e boa performance. Neste guia para iniciantes, saiba mais sobre o que é JavaScript e suas principais vantagens. (Melo, 2021).

2.4 – Vue JS

Vue JS é um framework JavaScript open source, para fazer a criação de aplicações web, possibilitando a criação de aplicações em forma reativa.

Vue JS é muito utilizado para criar aplicações single page (página única) e também para desenvolver vários tipos de interfaces, que possuem necessidades de maior interação e experiência mais valorosa para o usuário. (Picollo, 2020).

2.5 - C Sharp

O C# uma poderosa linguagem de programação orientada a objetos, que pode ser empregada no desenvolvimento de diversos tipos de projetos de software. Com ele é possível desenvolver aplicações desde aplicações embarcadas até aplicativos de área de trabalho, mobile e sistemas web.

O C# é uma linguagem de programação muito popular, sendo uma excelente escolha devido a sua baixa curva de aprendizado e simplicidade (mas sem deixar de ser uma linguagem poderosa). Além disso, ela é a linguagem principal do .NET Framework, o framework para desenvolvimento da Microsoft. (Guedes, 2018).

2.6 – ASP.NET

ASP.NET é um framework de desenvolvimento web, desenvolvido pela Microsoft, utilizado para a construção de aplicativos e páginas web dinâmicas e interativas que permite aos desenvolvedores criar sites e aplicativos web.

ASP.NET é um framework de código aberto para criar aplicativos da web na estrutura .NET (dotNET). Foi concebido pela Microsoft em 2002 na sua versão 1.0, dessa forma, as pessoas desenvolvedoras conseguiram criar aplicativos, serviços e sites dinâmicos utilizando essa tecnologia. Toda a estrutura é baseada no protocolo HTTP padrão, que é o protocolo padrão utilizado em todos os aplicativos. (Martin, 2021).

2.7 – React Native

React Native é um framework de desenvolvimento de aplicativos móveis criado pelo Facebook, que permite aos desenvolvedores criar aplicativos nativos para iOS e Android usando JavaScript e React.

O React Native é um framework de código aberto usado em vários setores e adotado com sucesso por centenas de empresas em todo o mundo, incluindo Uber, Microsoft e Facebook. (Cunha, 2023).

2.8 – Bootstrap

Bootstrap é um dos frameworks front-end que fornece um conjunto abrangente de ferramentas para criar interfaces de usuário responsivas e estilizadas de forma rápida e eficiente.

Bootstrap é um framework front-end que fornece estruturas de CSS para a criação de sites e aplicações responsivas de forma rápida e simples. Além disso, pode lidar com sites de desktop e páginas de dispositivos móveis da mesma forma. (GUILHERME, 2023).

2.9 – Figma

Figma é uma plataforma online de criação de interfaces, wireframes e protótipos. Seu papel é oferecer recursos de design de telas para aplicações variadas, permitindo que times de Design trabalhem em conjunto no mesmo projeto remotamente e simultaneamente. “De acordo com seu próprio site, o Figma é uma ferramenta de design que combina a acessibilidade da web com as funcionalidades de um aplicativo nativo.” (Lopes, 2023).

2.10 – Canva

Canva é uma plataforma online de design e comunicação visual que tem como missão colocar o poder do design ao alcance de todas as pessoas do mundo, para que elas possam criar o que quiserem e publicar suas criações onde quiserem. “O Canva é uma plataforma de design online e gratuita, criada em 2013 para “garantir que qualquer pessoa no mundo possa criar qualquer design para publicar em qualquer lugar”, segundo o próprio.” (Labouriau, 2023).

2.11 – MongoDB

MongoDB é um banco de dados NoSQL orientado a documentos, que armazena dados em formato binário JSON (BSON), Bancos de dados são repositórios essenciais para o armazenamento e a gestão de informações em praticamente todas as aplicações de software.

Em vez de utilizar tabelas e linhas como no modelo SQL, o MongoDB guarda as bases num único documento, que pode conter todas as informações

necessárias. Sendo assim, ele possibilita o acesso de forma eficiente, independentemente do esquema ou da estrutura utilizada. (Coutinho, 2023).

3. EMPRESA

Nós somos a equipe *Indominus Dev* Formada por 5 integrantes com habilidades diferentes e temos como objetivo oferecer soluções eficientes de software.

3.1 Missão

A missão da *Indominus Dev* é criar soluções acessíveis para os atuais problemas de nossa sociedade. Temos o compromisso de fornecer serviços de alta qualidade que atendam às necessidades de nossos clientes e superem suas expectativas. Utilizando as tecnologias disponíveis, buscamos ajudar as pessoas a lidarem com seus problemas.

3.2 Visão

Nosso objetivo é tornar-nos reconhecidos por fornecer soluções para os principais problemas e dificuldades das pessoas em nossa sociedade. Nossa ambição é expandir nossos negócios globalmente e nos tornar referência em inovação e qualidade. Vislumbramos um futuro em que nossos serviços serão utilizados em vários setores.

3.3 Valores

- **Inovação:** Buscamos constantemente novos métodos e tecnologias para criar soluções que superem as expectativas de nossos usuários.
- **Qualidade:** Estamos comprometidos em desenvolver serviços de alta qualidade, garantindo que nossos produtos atendam a todas as necessidades de nossos clientes.
- **Integridade:** Construimos relacionamentos de confiança com nossos clientes e parceiros, agindo de forma honesta, transparente e ética em todas as interações.
- **Aprendizagem Contínua:** Estamos sempre em busca de aprender e melhorar, acompanhando as últimas tendências e desenvolvimentos na área de tecnologia.
- **Respeito:** Valorizamos a diversidade e respeitamos as diferentes perspectivas e contribuições de cada membro de nossa equipe para com a comunidade.

3.4 Logos

3.4.1 Logo da Equipe



Figura 1 - Logo da Equipe

Fonte: Criado pela equipe

3.4.2 Logo do Projeto



Figura 2 - Logo do Projeto

Fonte: Criado pela equipe

3.4.3 Slogan da Empresa e do Projeto

Slogan da empresa: Indominus Dev: Avançando indomavelmente no futuro

Slogan do projeto: Inovação que Transforma Condomínios: Gestão Simplificada, Resultados Surpreendentes.

4. ANÁLISE

4.1. Documento Visão

DESCRIÇÃO GERAL:

Nosso aplicativo de gestão de condomínio é um aplicativo desenvolvido para simplificar a comunicação e resolver problemas dentro do condomínio, mantendo a transparência em todas as ações. Ele atua como um meio centralizado para os moradores se comunicarem e resolverem suas questões do cotidiano.

Caso você seja uma empresa de gestão de condomínios ou até mesmo um síndico particular, você pode cadastrar o seu conjunto habitacional no aplicativo, selecionando a quantidade de apartamentos e os espaços internos do local.

O síndico deve cadastrar um morador responsável por cada apartamento, para ter um controle de quem se responsabiliza pela moradia.

Todo morador pode se cadastrar, preenchendo as suas informações pessoais e selecionando o apartamento onde mora.

Em relação às finanças, o aplicativo permite que todos os moradores vejam claramente as entradas e saídas de dinheiro do condomínio. Isso significa que cada pagamento e recebimento pode ser acompanhado por todos, garantindo transparência e confiança na administração financeira.

Um recurso importante é o mural do condomínio. Nele, os moradores podem compartilhar fotos e até mesmo avisos. Isso promove uma comunicação aberta e interação completa entre os moradores do condomínio.

Além disso, a ferramenta conta com um sistema de denúncia anônima. Caso algum apartamento esteja causando distúrbios, como barulho excessivo ou descumprindo regras internas do condomínio, os moradores podem realizar denúncias anonimamente. O sistema permite que o motivo da denúncia seja selecionado ou digitado e enviada ao síndico. Com isso, o síndico pode aplicar um status negativo ao apartamento e, em casos mais graves, aplicar multa, dependendo do descumprimento das regras.

Para a administração do condomínio, o aplicativo oferece ferramentas para gerenciar a transição de síndicos, divulgar eventos e enviar comunicados importantes. Além disso, a administração pode acompanhar as interações dos moradores e agir rapidamente para resolver qualquer problema que surja.

Com o nosso aplicativo, a vida no condomínio se torna mais organizada e harmoniosa. Ele serve como um canal de comunicação eficiente e transparente, garantindo que todos os moradores se sintam ouvidos e participem ativamente da gestão do condomínio.

REQUISITOS FUNCIONAIS

RF 01 - Cadastrar Morador

Descrição: O sistema deve permitir o gerenciamento de moradores do condomínio, incluindo o cadastro de novos moradores e a concessão de acesso ao sistema. Isso envolve a coleta e armazenamento de informações relevantes sobre o morador, como nome completo, número de identificação (RG ou CPF), unidade habitacional, informações de contato (telefone e e-mail), data de entrada, entre outros.

Dados: Nome completo, número de identificação, unidade habitacional, informação de contato, data de entrada, método de autenticação.

RF 02 - Comunicar internamente

Descrição: Esse requisito funcional se refere à plataforma fornecer um sistema de comunicação interna, que premirá que os moradores, síndicos, administradores possam se comunicar entre si de forma eficiente facilitando a troca de informações importantes.

Dados: Notificações.

RF 03 - Publicar Avisos e Comunicados

Descrição: Este requisito funcional se refere a plataforma permitir a publicação de avisos e comunicados. Administradores poderão publicar e excluir avisos e comunicados que vão estar disponíveis para todos, facilitando o compartilhamento de informações importantes para os moradores.

Dados: Título do Aviso, descrição, data de publicação, autor, data de expiração, anexos.

RF 04 - Reservar Áreas do Condomínio

<p>Descrição: Este requisito se refere a permitir a gestão de reservas de áreas comuns do condomínio. Moradores poderão ver a disponibilidade de reserva de áreas de espaço compartilhamento comum como salão de festas, academia, churrasqueira, entre outros. Administradores e síndicos poderão gerenciar essas reservas, garantindo um uso organizado e justo dessas áreas.</p>
<p>Dados: espaço comum, data de reserva, horário de início, horário de término, nome do morador, status da reserva.</p>

<p>RF05 - Controlar Finanças</p>
<p>Descrição: Este requisito funcional se refere a necessidade de a plataforma fornecer funcionalidades para o controle financeiro do condomínio. O sistema deverá permitir que administradores e síndicos gerenciem as finanças do condomínio, incluindo relatório de boletos, controle de receitas e despesas, geração de relatórios financeiros e acompanhamento do orçamento.</p>
<p>Dados: Receitas, despesas, orçamento, boletos, relatórios financeiros, foto, nome do administrador, data, status.</p>

<p>RF06 – Cadastrar Visitantes</p>
<p>Descrição: Este requisito funcional se refere a plataforma permitir a gestão de visitantes do condomínio. A plataforma deverá possibilitar que moradores registrem visitantes com antecedência, permitindo um controle mais eficiente de entrada e saída e garantindo a segurança. Administradores e síndicos poderão visualizar e gerenciar esses registros, garantindo a segurança e o controle de acesso ao condomínio.</p>
<p>Dados: nome visita, morador responsável, data e hora da visita.</p>

<p>RF07 - Cadastrar ocorrência</p>
<p>Descrição: Este requisito funcional aborda a necessidade de o sistema ter um monitoramento de ocorrências na plataforma de gestão do condomínio, permitindo que os moradores relatem incidentes, problemas de segurança, e outros problemas que ocorreram no condomínio.</p>
<p>Dados: tipo de ocorrência, data e hora, descrição da ocorrência, local de ocorrência, status.</p>

RF08 - Visualizar Histórico

Descrição: Esse requisito funcional se refere a manter um registro de comunicações feitas na plataforma. Permitindo que os moradores tenham acesso a mensagens, avisos, comunicados e notificações anteriores, facilitando o acompanhamento dessas informações e proporcionando transparência e a rastreabilidade de informações do condomínio.

Dados: Categoria, data, remetente, assunto, conteúdo, anexos, status.

RF09 - Customizar dPerfil

Descrição: Esse requisito funcional aborda a funcionalidade de customização de perfis do usuário na plataforma. Possibilitando que cada morador, síndico ou administrador personalize as configurações.

Dados: foto, nome, cargo, informações de contato, tema, idioma, senha.

RF10 - Gerar Relatório Gerencial

Descrição: Esse requisito funcional se refere a plataforma fornecer funcionalidades de geração de relatórios gerenciais. Que permitirão que os administradores e síndicos tenham informações detalhadas de aspectos do condomínio, como finanças, segurança, manutenção, entre outros, melhorando assim a tomada de decisões e desempenho do condomínio.

Dados: tipo relatório, período, gráficos, exportação, foto.

RF11 - Inativar usuário

Descrição: Este requisito funcional se trata da capacidade de deixar um usuário da plataforma inativo, caso esse usuário não faça mais parte do condomínio. Essa funcionalidade permitirá que administradores e síndicos removam usuários que não têm mais nenhum vínculo com o condomínio ou que não são mais autorizados a acessar a plataforma, garantindo a segurança e a integridade dos dados.

Dados: identificação usuário, motivo.

RF12 - Cadastrar condomínio

<p>Descrição: Este caso de uso diz a respeito do cadastro de condomínios ao sistema permitindo que ele seja acessível para a adição de novos usuários e gerando dados específicos daquele condomínio. Ao cadastrar a unidade ao sistema ele concede acesso as funcionalidades que o sistema possui.</p>
<p>Dados: Nome do condomínio, unidade, nome do administrador.</p>

<p>RF13 - Solicitar manutenção</p>
<p>Descrição: Este requisito funcional se refere ao sistema permitir que os usuários solicitem manutenções para os problemas encontrados no condomínio. O morador poderá também visualizar as manutenções marcadas e pendentes no condomínio. Ao enviar uma solicitação o sistema deverá enviar essa solicitação ao administrador, que poderá visualizar a solicitação, aprovando ou desaprovando.</p>
<p>Dados: Título da manutenção, descrição, foto, prioridade, tipo de manutenção, status.</p>

<p>RF14 - Gerenciar correspondência</p>
<p>Descrição: O sistema deve permitir que os administradores, síndico ou funcionários do condomínio registrem novas correspondências recebidas para os moradores, preenchendo os dados da correspondência recebida.</p>
<p>Dados: Identificador, data de recebimento, destinatário, tipo de correspondências, observações, status.</p>

<p>RF15 - Cadastrar assembleia</p>
<p>Descrição: a aplicação deve permitir que o administrador ou síndico crie uma assembleia alertando os moradores e registrando informações relevantes sobre a reunião.</p>
<p>Dados: Título da assembleia, data da assembleia, horário, local da assembleia, pauta, observações adicionais, status.</p>

REQUISITO NÃO FUNCIONAL

<p>RNF 01 - Desempenho</p>

Descrição: O sistema deve ser capaz de processar e responder a solicitações de login, cadastro e comunicação interna em menos de 2 segundos, garantindo uma experiência de usuário eficiente. Além disso, a plataforma deve suportar até 500 usuários simultâneos sem degradação de desempenho.

RNF 02 - Segurança

Descrição: O sistema deve implementar protocolos de segurança robustos para proteger os dados dos usuários, incluindo criptografia de senhas e dados sensíveis, autenticação de dois fatores para acesso administrativo e auditoria de logs de acesso e operações críticas. Todas as comunicações entre o cliente e o servidor devem ser criptografadas usando TLS.

RNF 03 - Confiabilidade

Descrição: A plataforma deve garantir um tempo de atividade (*uptime*) de 99.9% ao longo do ano, com processos de backup automático diário e redundância de dados para evitar perda de informações em caso de falhas.

RNF 04 - Usabilidade

Descrição: O sistema deve ser intuitivo e fácil de usar, proporcionando uma experiência de usuário amigável e acessível. Deve haver suporte para diferentes idiomas e temas para atender às necessidades dos diversos usuários.

RNF 05 - Escalabilidade

Descrição: A plataforma deve ser escalável para suportar um aumento no número de usuários e dados sem perda de desempenho. Deve ser capaz de se adaptar a novas funcionalidades e integrações futuras sem necessitar de uma reestruturação completa.

RNF 06 - Compatibilidade

Descrição: A plataforma deve ser compatível com os principais navegadores da web (Chrome, Firefox, Safari, Edge) e deve ter uma versão mobile responsiva ou um aplicativo móvel para dispositivos iOS e Android, garantindo acesso em diferentes dispositivos.

RNF 07 - Manutenibilidade

Descrição: O sistema deve ser desenvolvido utilizando práticas de codificação e design modular que facilitem a manutenção e a atualização do software. Deve haver documentação completa e detalhada para desenvolvedores, incluindo instruções de configuração, uso e extensão da plataforma.

RNF 08 - Privacidade

Descrição: A plataforma deve assegurar a privacidade dos dados dos usuários, cumprindo todas as regulamentações aplicáveis de proteção de dados, como a LGPD (Lei Geral de Proteção de Dados). Os dados dos usuários devem ser utilizados exclusivamente para os fins especificados e não devem ser compartilhados com terceiros sem consentimento explícito.

RNF 09 - Interoperabilidade

Descrição: A plataforma deve ser capaz de integrar-se com sistemas externos, como serviços de pagamento e ferramentas de análise, por meio de APIs bem definidas e documentadas. Essa capacidade deve permitir a troca segura e eficiente de dados entre sistemas.

RNF 10 - Disponibilidade

Descrição: A plataforma deve estar disponível 24/7, com suporte técnico acessível para resolver problemas críticos em tempo hábil. Deve haver notificações proativas para os usuários sobre manutenção programada e interrupções de serviço.

4.2. Descrições de Casos de Uso

Caso de uso	UC001
Nome	Cadastrar morador
Ator	Síndico, Administrador
Stackholder/Interessados	Morador, Síndico, Administrador
Pré-condições	Usuário não cadastrado no sistema
Pós-condição	Usuário ter acesso ao sistema
Breve descritivo	Usuário se cadastra no sistema com as informações
Ações do ator	Ações do sistema
Cenário principal	
• Solicita tela de cadastro de morador.	
	• Exibe tela de cadastro.
• Preencher os dados de cadastro.	
	• Verifica se dados estão completos e exibe: "cadastro realizado com sucesso".
Cenário alternativo (Secundário)	
3.1 Preenche dados incompletos ou inválidos	
	4.1 O sistema exibe tela de erro e retorna o usuário à tela de cadastro para correção dos dados.
Restrições/Validações:	
Dados utilizados:	Nível de acesso, nome completo, número de identificação, unidade habitacional, informação de contato, data de entrada

Caso de uso	UC002
Nome	Comunicar internamente
Ator	Morador, Síndico, Administrador
Stackholder/Interessados	Morador, Síndico, Administrador
Pré-condições	Funcionalidades de síndico e administrador não definidas no sistema
Pós-condição	Funcionalidades de síndico, administrador e morador definidas no sistema
Breve descritivo	Este caso de uso diz respeito à definição de funcionalidades do sistema e quem pode acessá-las, conforme definido pelo administrador.
Ações do ator	Ações do sistema
Cenário principal	

	<ul style="list-style-type: none"> O sistema disponibiliza uma plataforma de comunicação interna para os moradores, síndicos e administradores.
<ul style="list-style-type: none"> Os moradores podem enviar mensagens privadas uns aos outros ou moradores podem criar e participar de grupos de discussão sobre tópicos específicos. 	
	<ul style="list-style-type: none"> O sistema envia notificações aos moradores sobre novas mensagens, atividades em grupos de discussão e outras informações relevantes.
Cenário alternativo (Secundário)	
2.1 Usuário tenta enviar uma mensagem sem estar conectado à internet.	
	3.1 Sistema exibe mensagem: "Sem conexão com a internet. Tente novamente mais tarde."
Restrições/Validações:	
Dados utilizados:	Anexos, Data de Publicação, Notificação, Status da Mensagem, Identificação do Usuário, Mensagem.

Caso de uso	UC003
Nome	Publicar Avisos e Comunicados
Ator	Morador, Síndico, Administrador
Stackholder/Interessados	Morador, Síndico, Administrador
Pré-condições	Usuário cadastrado no sistema
Pós-condição	Morador, síndico ou administrador consegue realizar postagens, como avisos e comunicados
Breve descritivo	Neste caso de uso o síndico pode enviar avisos e comunicados para os moradores
Ações do ator	Ações do sistema
Cenário principal	
	1. O sistema disponibiliza uma aba para interação de postagens. Ações do Ator:
2. Todos os usuários cadastrados no condomínio podem fazer postagens, visualizar, curtir e comentar.	

	3. O sistema exibe uma notificação na aba de postagens para alertar sobre novas postagens.
Cenário alternativo (Secundário)	
• O usuário ultrapassa o limite de caracteres no post.	
	• O sistema não permite a postagem e exibe uma mensagem de erro indicando que o limite de caracteres foi ultrapassado.
Restrições/Validações:	
Dados utilizados:	Título do Aviso, descrição, data de publicação, autor, data de expiração, anexos.

Caso de uso	UC004
Nome	Reservar Áreas do Condomínio
Ator	Síndico, Morador, Administrador
Stackholder/Interessados	Síndico, Administrador, Morador
Pré-condições	Ser cadastrado no sistema
Pós-condição	Solicitar reserva e reservar área.
Breve descritivo	Se o condomínio possui alguma área interna para ser reservada, os moradores podem solicitar uma reserva. O síndico ou administrador pode aprovar ou recusar a solicitação.
Ações do ator	Ações do sistema
Cenário principal	
	• O sistema disponibiliza para os moradores uma aba para solicitar a reserva das áreas do condomínio que estão disponíveis.
• Os moradores podem solicitar a reserva do espaço interno com a data e hora da reserva.	
	• O sistema verifica a disponibilidade do local.
	• O sistema exibe uma notificação para o síndico e administrador sobre uma nova reserva solicitada.
• Síndico e administrador têm a opção de visualizar as reservas solicitadas e a opção de aceitar ou recusar a solicitação.	
	• O sistema envia uma notificação para o morador sobre o status da reserva solicitada.
Cenário alternativo (Secundário)	

• O usuário tenta reservar uma área indisponível.	
	• O sistema recusa automaticamente a solicitação da área e exibe uma mensagem de erro.
Restrições/Validações:	
Dados utilizados:	Notificações, Status da área, Solicitação da reserva, Status da solicitação, Data e hora da reserva

Caso de uso	UC005
Nome	Controlar Finanças
Ator	Síndico, Administrador
Stackholder/Interessados	Morador, Síndico, Administrador
Pré-condições	Estar cadastrado no sistema
Pós-condição	Permite a visualização e gestão do controle financeiro.
Breve descritivo	O sistema disponibiliza um menu para o controle financeiro do condomínio, onde a administração e o síndico podem realizar relatórios das finanças da copropriedade.
Ações do ator	Ações do sistema
Cenário principal	
	• O sistema disponibiliza uma aba para a visualização das finanças do condomínio.
• O morador visualiza a gestão financeira.	
	3. O sistema disponibiliza a opção de relatórios e a geração de boletos para o controle das finanças.
4. O síndico e o administrador realizam o controle financeiro.	
	5. A cada ação realizada dentro da gestão financeira, o sistema exibe uma notificação para os usuários visualizarem. Cenário Alternativo.
Cenário alternativo (Secundário)	
2.1 Morador tenta acessar a tela de relatório sem haver informações para o relatório.	
	3.2 O sistema exibe uma mensagem de erro indicando a falha na geração do relatório.
Restrições/Validações:	

Dados utilizados:	Boletos, Data e hora dos relatórios
--------------------------	-------------------------------------

Caso de uso	UC006
Nome	Cadastrar visitante
Ator	Síndico, Administrador, Morador
Stackholder/Interessados	Síndico, Administrador, Morador
Pré-condições	Usuário não estar cadastrado no sistema
Pós-condição	O administrador poderá saber visitantes que chegarão ao condomínio.
Breve descritivo	A aplicação irá possibilitar que o usuário solicite visitantes no sistema ao administrador ou síndico
Ações do ator	Ações do sistema
Cenário principal	
<ul style="list-style-type: none"> O morador acessa a seção de solicitar visita. 	<ul style="list-style-type: none"> O Aplicação exibi a tela de solicitar visitas.
<ul style="list-style-type: none"> Morador preenche dados do visitante e clica em compartilhar. 	<ul style="list-style-type: none"> A aplicação abre menu e exibi aplicativos em que ele pode compartilhar.
<ul style="list-style-type: none"> Morador escolhe aplicativo e pessoa para o convite ser compartilhado. 	<ul style="list-style-type: none"> A aplicação leva dados cadastrados na aplicação com a pessoa.
Cenário alternativo (Secundário)	
3.1 Morador tenta cadastrar um visitante para uma data passada.	
	4.1 Sistema exibe: "Local indisponível para o horário selecionado." e redireciona para a tela de seleção de data novamente.
Restrições/Validações:	
Dados utilizados:	Nome visitante, data da visita, horário, unidade da visita

Caso de uso	UC007
Nome	Cadastrar ocorrência
Ator	Síndico, administrador, morador
Stackholder/Interessados	Síndico, administrador, morador
Pré-condições	Haver alguma ocorrência
Pós-condição	Ocorrência
Breve descritivo	Monitoramento de ocorrências na plataforma de gestão do condomínio
Ações do ator	Ações do sistema
Cenário principal	
<ul style="list-style-type: none"> Morador solicita uma tela para registrar uma ocorrência 	<ul style="list-style-type: none"> Sistema libera tela para cadastro de ocorrência
<ul style="list-style-type: none"> Seleciona o tipo de ocorrência e explica como ela afeta os condôminos, colocando data e hora da ocorrência 	
	<ul style="list-style-type: none"> Verifica os dados e salva as informações registradas mostrando: "Ocorrência registrada com sucesso"
Cenário alternativo (Secundário)	
3.1 Não seleciona um tipo de ocorrência	
	4.1 Verifica os dados em falta e exibi: "Selecione um tipo de ocorrência"
3.2 Atinge um limite de caracteres	
	4.2 Analise os caracteres e exibi: "Limite de caracteres ultrapassado"
Restrições/Validações:	Cada postagem tem um limite de 250 caracteres
Dados utilizados:	Tipo ocorrência, descrição ocorrência, data ocorrência, nome usuário, local ocorrência, anexos

Caso de uso	UC008
Nome	Visualizar Histórico
Ator	Síndico, Administrador, Morador
Stackholder/Interessados	Síndico, Administrador, Morador
Pré-condições	Haver ocorrência, avisos ou comunicados no sistema
Pós-condição	Poder ter acesso a informações antigas
Breve descritivo	Manter um registro de comunicações feitas na plataforma

Ações do ator	Ações do sistema
Cenário principal	
<ul style="list-style-type: none"> • Usuário solicita a tela para ver histórico 	
	<ul style="list-style-type: none"> • Sistema libera tela onde o usuário escolhe o tipo de ocorrência que quer ver: Avisos, comunicados ou ocorrência
<ul style="list-style-type: none"> • Usuário seleciona o tipo de histórico que quer visualizar 	
	<ul style="list-style-type: none"> • Verifica as informações registradas sobre aquele tipo de ocorrência e mostra o histórico
Cenário alternativo (Secundário)	
3.1 Usuário seleciona um tópico que não possui conteúdo	
	4.1 Verifica os dados em falta e exibi: "Nenhum conteúdo disponível"
Restrições/Validações:	
Dados utilizados:	Tipo ocorrência, descrição ocorrência, data ocorrência, nome usuário, local ocorrência, anexos,

Caso de uso	UC009
Nome	Customizar Perfil
Ator	Síndico, Administrador
Stackholder/Interessados	Síndico, morador
Pré-condições	O morador deve estar logado no sistema
Pós-condição	O perfil do usuário é atualizado com as informações personalizadas
Breve descritivo	Síndico ou morador personaliza seu perfil dentro do sistema.
Ações do ator	
Ações do sistema	
Cenário principal	
1. O Morador/Síndico entra na seção de perfil.	
	2. Exibe a tela de edição de perfil
3. Edita os campos que deseja personalizar.	
	4. Valida e atualiza as informações de contato no perfil do usuário.
Cenário alternativo (Secundário)	

5. O Morador/Síndico coloca uma informação que já está no sistema.	
	6. Exibe uma mensagem de erro indicando a duplicidade da informação.
Restrições/Validações:	
Dados utilizados:	Foto, nome, cargo, informações de contato, tema, idioma

Caso de uso	UC010
Nome	Gerar Relatório Gerencial
Ator	Síndico, Administrador
Stackholder/Interessados	Síndico, Administrador
Pré-condições	O usuário deve estar logado no sistema
Pós-condição	Relatório financeiro ou de segurança detalhado é gerado e disponibilizado
Breve descritivo	Gerar relatórios detalhados como financeiros ou de segurança.
Ações do ator	Ações do sistema
Cenário principal	
1. Administrador/Síndico seleciona o tipo de relatório "Finanças" ou "Segurança".	
	2. Gera o relatório com gráficos ilustrativos conforme solicitado.
Cenário alternativo (Secundário)	
6. Administrador/Síndico seleciona um período que não possui informações.	
	7. Informa ao usuário que não há informações disponíveis para o período selecionado.
Restrições/Validações:	
Dados utilizados:	Tipo relatório, período, gráficos, exportação

Caso de uso	UC011
Nome	Inativar Usuário
Ator	Síndico, Administrador
Stackholder/Interessados	Síndico, Administrador
Pré-condições	O usuário deve existir no sistema
Pós-condição	O usuário é removido do sistema e perde o acesso

Breve descritivo	Administrador ou síndico exclui um usuário do sistema quando necessário.
Ações do ator	Ações do sistema
Cenário principal	
1. O administrador ou síndico entra na tela de gerenciamento de morador.	
	2. Exibe tela com as informações de moradores
3. Seleciona a opção para gerenciar o usuário específico.	
	4. Exibe informações do usuário em específico.
3. Seleciona a opção de inativar a conta do morador.	
	4. Solicita confirmação da inativação do usuário.
Cenário alternativo (Secundário)	
3.1 O Administrador ou síndico decide não inativar a conta.	
	4.1 Cancela o processo de exclusão.
Restrições/Validações:	
Dados utilizados:	Identificação usuário

Caso de uso	UC012
Nome	Cadastrar condomínio
Ator	Síndico, Administrador
Stackholder/Interessados	Síndico, Administrador
Pré-condições	Condomínio não estar cadastrado no sistema
Pós-condição	Condomínio poder estar cadastrado no sistema tendo acesso as funcionalidades e permitindo adicionar usuários ao sistema
Breve descritivo	O condomínio é cadastrado podendo estar disponível a sua utilização e funcionalidades do sistema
Ações do ator	Ações do sistema
Cenário principal	
<ul style="list-style-type: none"> Síndico/Administrador solicita tela de cadastro de condomínio 	
	<ul style="list-style-type: none"> Sistema solicita dados do condomínio

<ul style="list-style-type: none"> Síndico/Administrador preenche dados do condomínio 	
	<ul style="list-style-type: none"> Condomínio verifica dados e cadastra a nova unidade no sistema
Cenário alternativo (Secundário)	
<ul style="list-style-type: none"> O síndico ou administrador tenta cadastrar o condomínio, mas não preenche todos os campos obrigatórios. 	
	<p>4.1 O sistema exibe uma mensagem: "Por favor, preencha todos os campos obrigatórios antes de prosseguir."</p> <p>O ator é redirecionado para a tela de cadastro para corrigir as informações.</p>
Restrições/Validações:	
Dados utilizados:	Nome do condomínio, unidade, nome do administrador

Caso de uso	UC013
Nome	Solicitar manutenção
Ator	Síndico, Administrador, Morador
Stackholder/Interessados	Síndico, Administrador, Morador
Pré-condições	Usuário não estar cadastrado no sistema
Pós-condição	A solicitação de manutenção é registrada e o administrador/síndico poderá visualizar aprovando ou não.
Breve descritivo	O sistema deve permitir que os usuários registrem solicitações de manutenção.
Ações do ator	Ações do sistema
Cenário principal	
<ul style="list-style-type: none"> O usuário acessa a seção de solicitação de manutenção no aplicativo. 	
	<ul style="list-style-type: none"> O Aplicação exibi outras manutenções pendentes.
<ul style="list-style-type: none"> Usuário clica no botão "fazer manutenção". 	
	<ul style="list-style-type: none"> A aplicação exibi campos de manutenção para o usuário preencher

<ul style="list-style-type: none"> • Usuário preenche os dados e clica no botão “enviar” 	
	<ul style="list-style-type: none"> • A aplicação cadastra e envia a solicitação ao administrador/síndico
Cenário alternativo (Secundário)	
<ul style="list-style-type: none"> • Campos obrigatórios não preenchidos 	
	<ul style="list-style-type: none"> • Sistema não aprova solicitação
Restrições/Validações:	
Dados utilizados:	Título da manutenção, descrição, foto prioridade, tipo de manutenção.

Caso de uso	UC014
Nome	Cadastrar correspondência
Ator	Síndico, Administrador
Stackholder/Interessados	Síndico, Administrador
Pré-condições	Usuário deverá ser um síndico ou morador
Pós-condição	A correspondência é registrada no sistema com um id único
Breve descritivo	O sistema deve permitir que o administrador/síndico cadastrem as correspondências referentes a um morador.
Ações do ator	Ações do sistema
Cenário principal	
<ul style="list-style-type: none"> • Administrador/síndico acessa seção de cadastro de correspondências no sistema 	
	<ul style="list-style-type: none"> • O Aplicação exibi a tela de cadastros de correspondências.
<ul style="list-style-type: none"> • Administrador/síndico preenche campos de cadastro de correspondências e clica em cadastrar. 	
	<ul style="list-style-type: none"> • O sistema verifica os campos e cadastra no sistema.
Cenário alternativo (Secundário)	
3.1 Administrador tenta cadastrar uma correspondência para um morador inexistente no sistema.	
	4.1 Sistema exibe: "Morador não encontrado. Verifique os dados e tente novamente."

Restrições/Validações:	
Dados utilizados:	Identificador, data de recebimento, destinatário, tipo de correspondências, observações, status.

Caso de uso	UC015
Nome	Cadastrar assembleia
Ator	Síndico, Administrador, Morador
Stackholder/Interessados	Síndico, Administrador, Morador
Pré-condições	Usuário não estar cadastrado no sistema.
Pós-condição	Administrador poderá alertar moradores sobre reunião.
Breve descritivo	A aplicação irá possibilitar que o usuário solicite visitantes no sistema ao administrador ou síndico.
Ações do ator	Ações do sistema
Cenário principal	
<ul style="list-style-type: none"> O morador acessa a seção de cadastrar assembleia 	
	<ul style="list-style-type: none"> O Aplicação exibi a tela de solicitar cadastrar assembleia.
<ul style="list-style-type: none"> Morador preenche dados da assembleia e clica em “cadastrar”. 	
	<ul style="list-style-type: none"> A aplicação verifica os dados e cadastra uma nova assembleia, alertando os usuários.
Cenário alternativo (Secundário)	
<ul style="list-style-type: none"> Campos obrigatórios não preenchidos 	
	<ul style="list-style-type: none"> Sistema não aprova cadastro
Restrições/Validações:	
Dados utilizados:	Título da assembleia, data da assembleia, horário, local da assembleia, pauta, observações adicionais, status

4.3. Diagrama de Casos de Uso

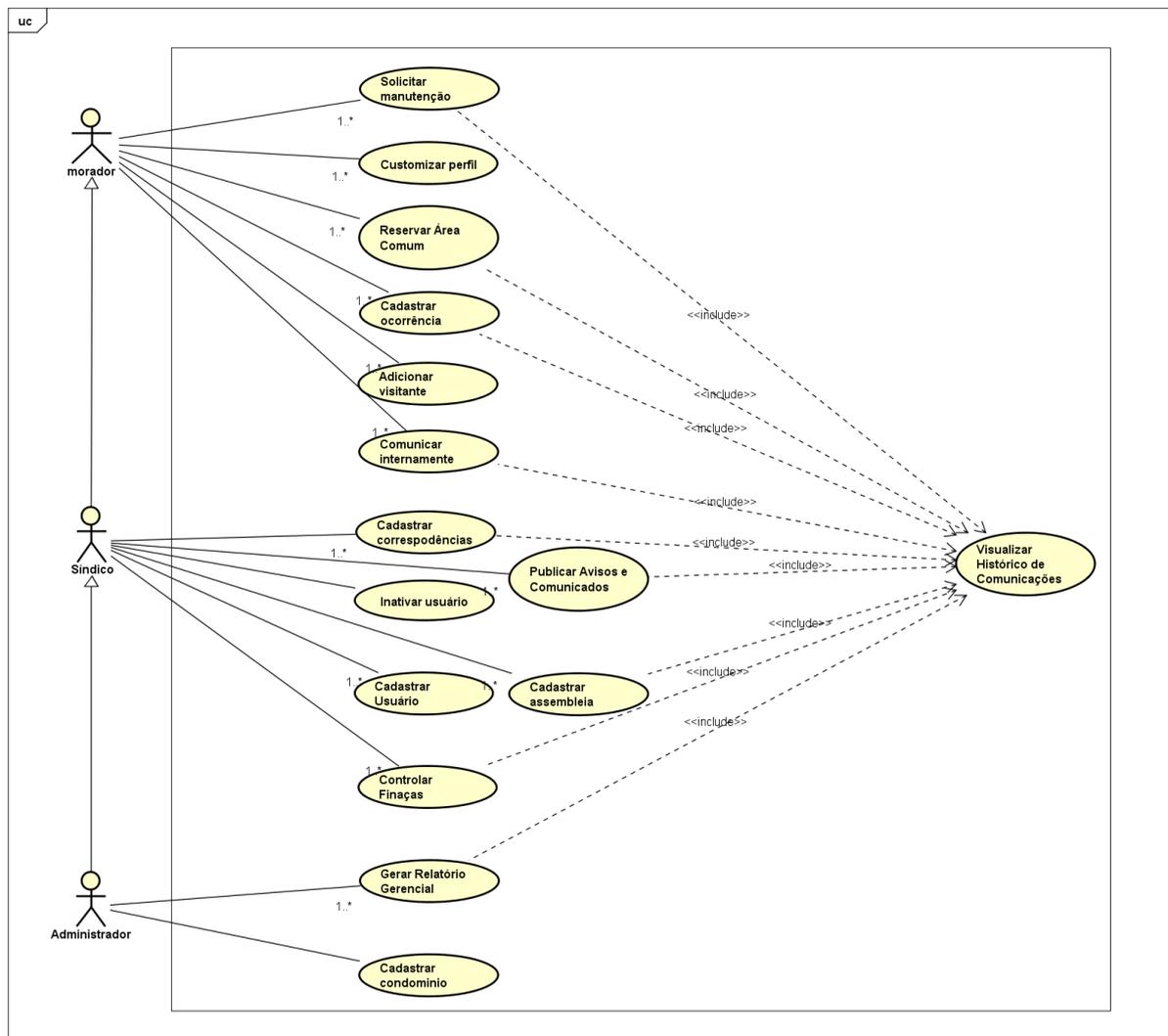


Figura 3 - Diagrama de Casos de Uso

4.4. Diagrama de Classes

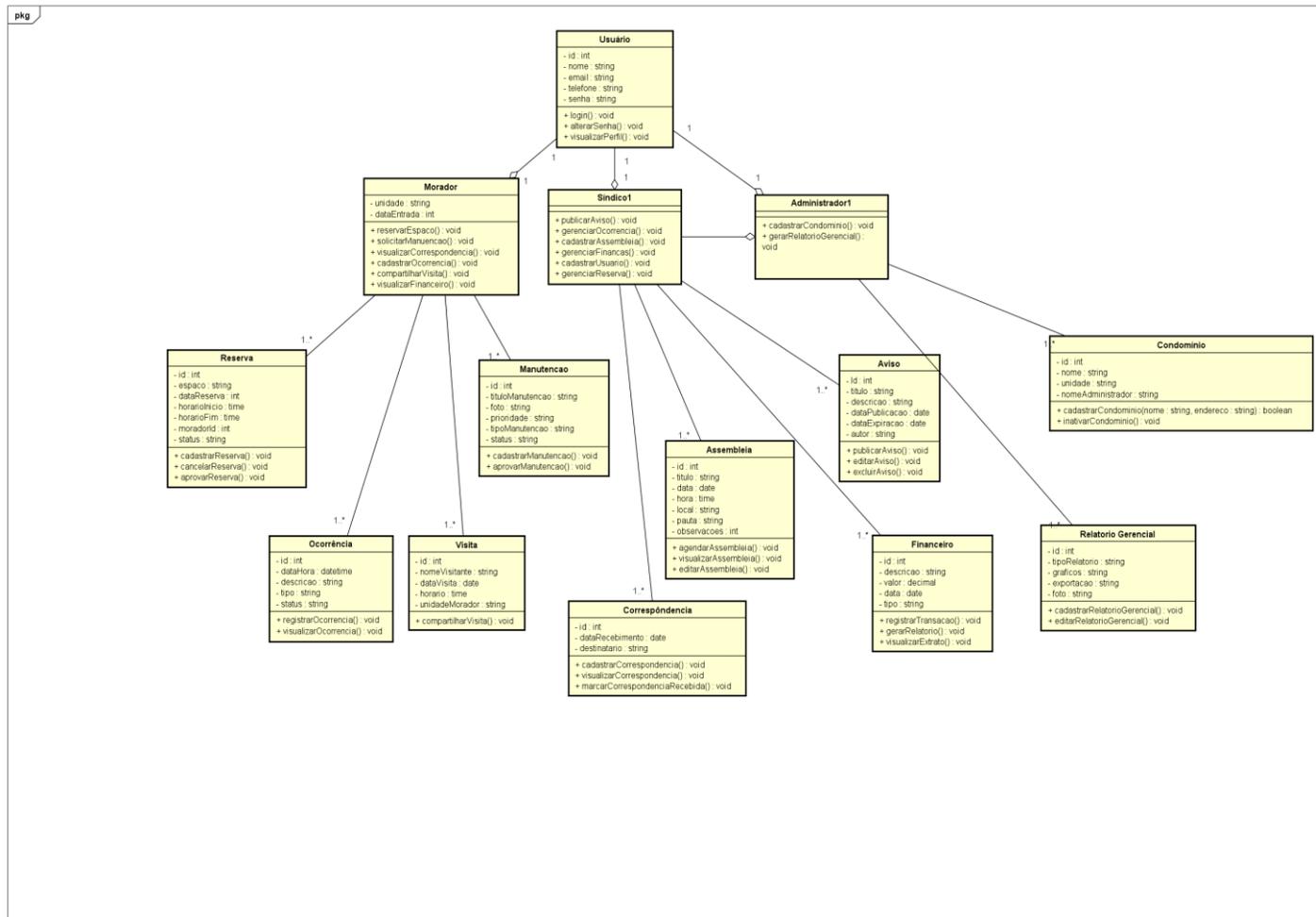


Figura 4 - Diagrama de Classes

4.5. MER – Modelo do Banco de Dados

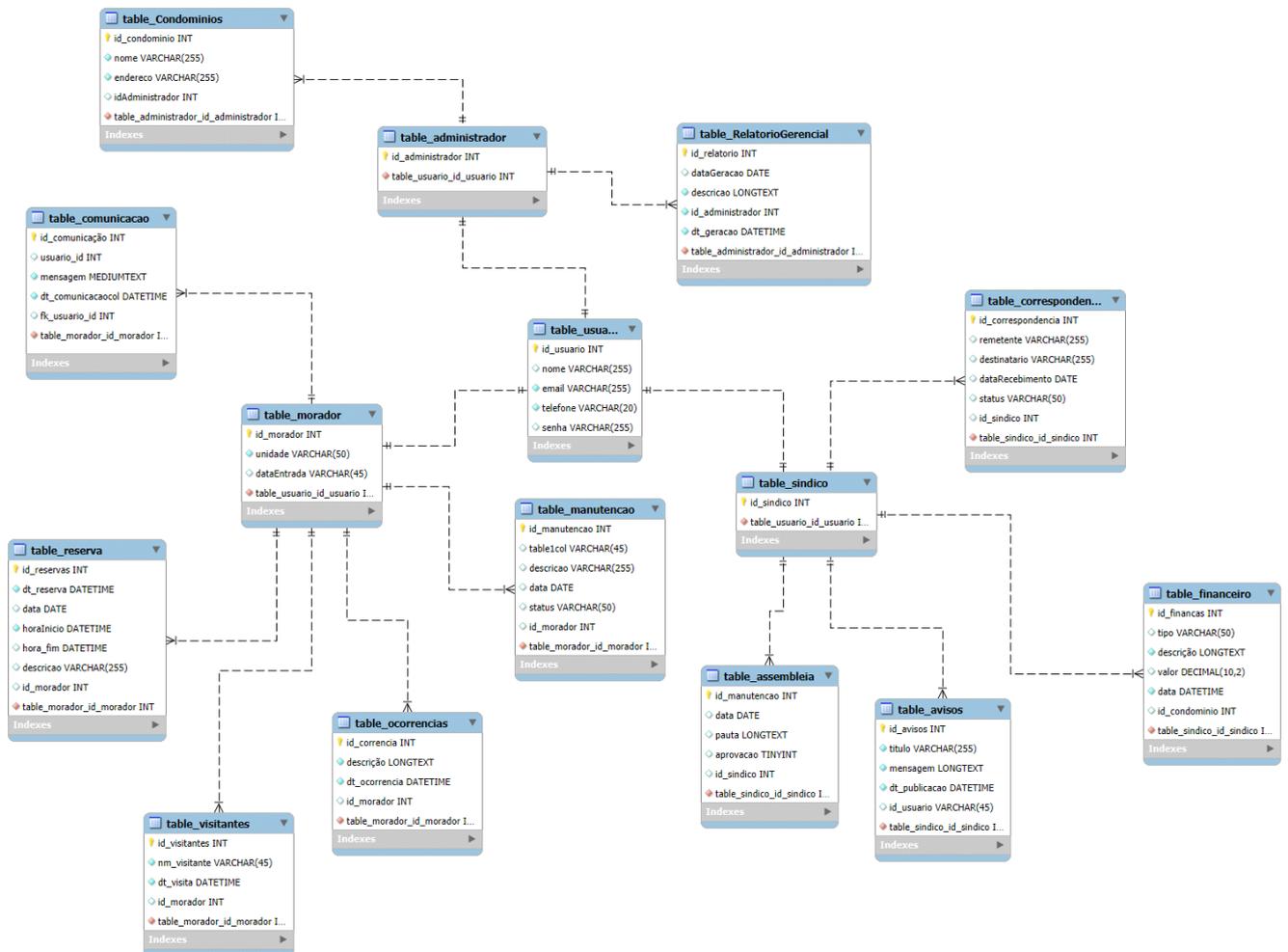


Figura 5 - Modelagem de Banco de Dados

4.6. Create do Banco de Dados

```
create database db_condoview;
use db_condoview;
```

```
CREATE TABLE tb_usuarios (
    id_usuario INT AUTO_INCREMENT PRIMARY KEY,
    nome VARCHAR(255) NOT NULL,
    email VARCHAR(255) NOT NULL UNIQUE,
    telefone VARCHAR(20),
    senha VARCHAR(255) NOT NULL
);
```

```

CREATE TABLE tb_sindico (
    id_sindico INT auto_increment primary key,
    tb_usuario_id_usuario INT NOT NULL,
    FOREIGN KEY (tb_usuario_id_usuario) REFERENCES
tb_usuarios(id_usuario)
);

```

```

CREATE TABLE tb_administrador (
    id_administrador INT AUTO_INCREMENT PRIMARY KEY,
    tb_usuario_id_usuario INT NOT NULL,
    FOREIGN KEY (tb_usuario_id_usuario) REFERENCES
tb_usuarios(id_usuario)
);

```

```

CREATE TABLE tb_condominios (
    id_condominio INT AUTO_INCREMENT PRIMARY KEY,
    nome VARCHAR(255) NOT NULL,
    endereco VARCHAR(255),
    id_administrador INT NOT NULL,
    FOREIGN KEY (id_administrador) REFERENCES
tb_administrador(id_administrador)
);

```

```

CREATE TABLE tb_reserva (
    id_reserva INT AUTO_INCREMENT PRIMARY KEY,
    data_reserva DATETIME NOT NULL,
    data DATE,
    hora_inicio TIME,
    hora_fim TIME,
    tb_morador_id_morador INT NOT NULL,
    FOREIGN KEY (tb_morador_id_morador) REFERENCES
tb_morador(id_morador)
);

```

```

CREATE TABLE tb_morador (
  id_morador INT AUTO_INCREMENT PRIMARY KEY,
  unidade VARCHAR(50),
  data_mudanca VARCHAR(45),
  tb_usuario_id_usuario INT NOT NULL,
  FOREIGN KEY (tb_usuario_id_usuario) REFERENCES
tb_usuarios(id_usuario)
  ON DELETE CASCADE ON UPDATE CASCADE
);

```

```

CREATE TABLE tb_ocorrencias (
  id_ocorrencia INT AUTO_INCREMENT PRIMARY KEY
);

```

```

CREATE TABLE tb_avisos (
  id_avisos INT AUTO_INCREMENT PRIMARY KEY
);

```

```

CREATE TABLE tb_comunicacao (
  id_comunicacao INT AUTO_INCREMENT PRIMARY KEY,
  usuario_id INT NOT NULL,
  mensagem MEDIUMTEXT,
  dt_comunicacao DATETIME,
  FOREIGN KEY (usuario_id) REFERENCES tb_usuarios(id_usuario)
);

```

```

CREATE TABLE tb_correspondencia (
  id_correspondencia INT AUTO_INCREMENT PRIMARY KEY,
  remetente VARCHAR(255),
  destinatario VARCHAR(255),
  data_Recebimento DATE,
  status VARCHAR(50),
  id_sindico INT NOT NULL,
  FOREIGN KEY (id_sindico) REFERENCES tb_sindico(id_sindico)
);

```

);

```
CREATE TABLE tb_manutencao (
    id_manutencao INT AUTO_INCREMENT PRIMARY KEY,
    local VARCHAR(45),
    descricao VARCHAR(255),
    dataabertura DATE,
    tb_morador_id_morador INT NOT NULL,
    FOREIGN KEY (tb_morador_id_morador) REFERENCES
tb_morador(id_morador)
);
```

```
CREATE TABLE tb_assembleia (
    id_assembleia INT AUTO_INCREMENT PRIMARY KEY,
    data DATE NOT NULL,
    pauta LONGTEXT
);
```

```
CREATE TABLE tb_relatoriogerencial (
    id_relatorio INT AUTO_INCREMENT PRIMARY KEY,
    dataGeracao DATE,
    descricao LONGTEXT,
    id_administrador INT NOT NULL,
    dt_geracao DATETIME,
    FOREIGN KEY (id_administrador) REFERENCES
tb_administrador(id_administrador)
);
```

```
CREATE TABLE tb_visitantes (
    id_visitantes INT AUTO_INCREMENT PRIMARY KEY,
    nm_visitante VARCHAR(45) NOT NULL,
    dt_visita DATETIME NOT NULL,
    id_morador INT NOT NULL,
    FOREIGN KEY (id_morador) REFERENCES tb_morador(id_morador)
```

);

```
CREATE TABLE tb_financeiro (
    id_financeiro INT AUTO_INCREMENT PRIMARY KEY,
    tipo Varchar(50),
    descricao LONGTEXT not null,
    valor DECIMAL(10,2),
    data DATETIME not null,
    id_condominio int not null,
    tb_sindico_id_sindico int not null,
    foreign key (id_condominio) references tb_condominios(id_condominio),
    foreign key (tb_sindico_id_sindico) references tb_sindico(id_sindico)
);
```

4.7 Principais Selects do Banco de Dados

SELECT

```
    c.nome AS nome_condominio,
    e.logradouro || ', ' || e.numero || ' - ' || e.cidade || '/' || e.estado AS
endereco_condominio,
    u.numero AS unidade_numero,
    a.nome AS anfitriao_nome,
    a.email AS anfitriao_email,
    ag.data AS data_agendada,
    ag.hora AS hora_agendada
```

FROM

```
    condominio c
```

INNER JOIN

```
    endereco e ON c.endereco_id = e.id
```

INNER JOIN

```
    unidade u ON u.condominio_id = c.id
```

INNER JOIN

anfitriao a ON u.anfitriao_id = a.id

INNER JOIN

agenda ag ON ag.unidade_id = u.id

4.8. Wireframe das Telas



Insira os dados e acesse sua conta

Email

Senha

[Esqueceu a
senha?](#)

[Criar conta](#)

ENTRAR

Criar Condomínio

Figura 6 - Wireframe Tela Cadastrar condomínio

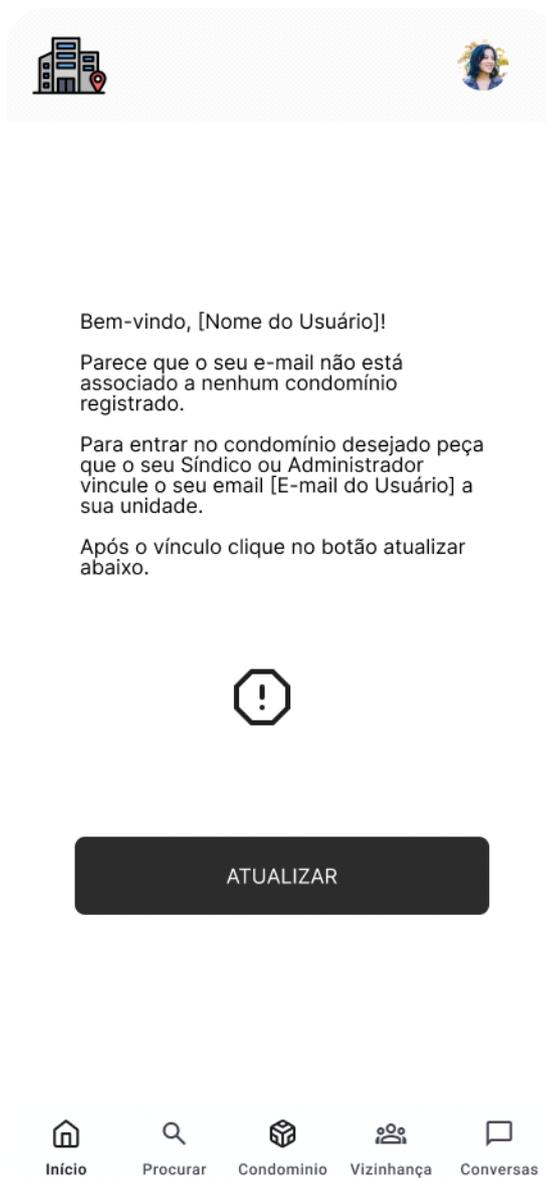


Figura 7 - Wireframe Tela de alerta de e-mail não cadastrado



Figura 8 - Wireframe Tela menu

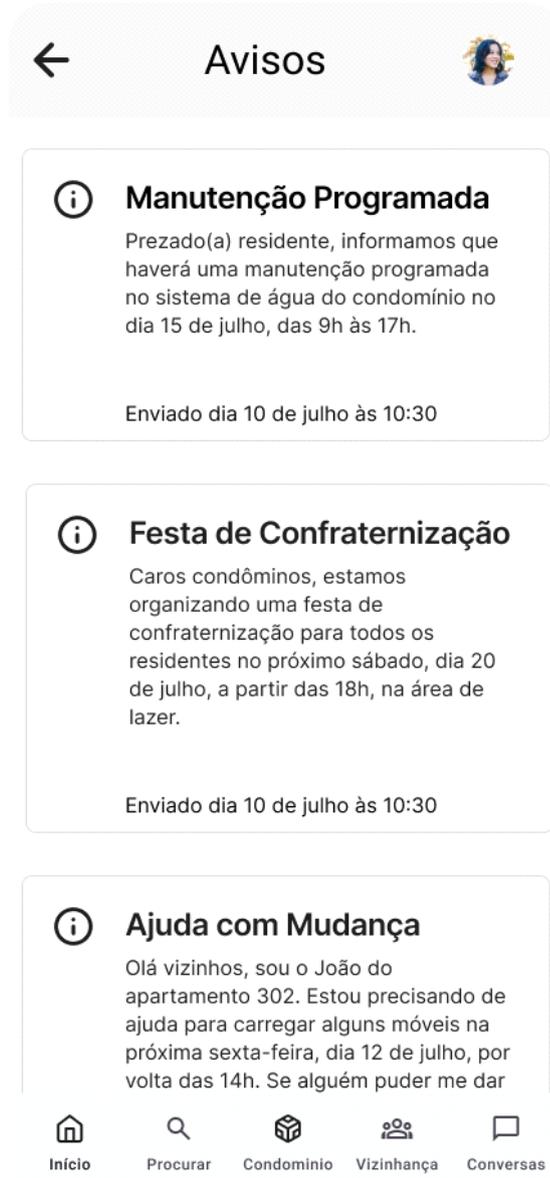


Figura 9 - Wireframe Tela avisos

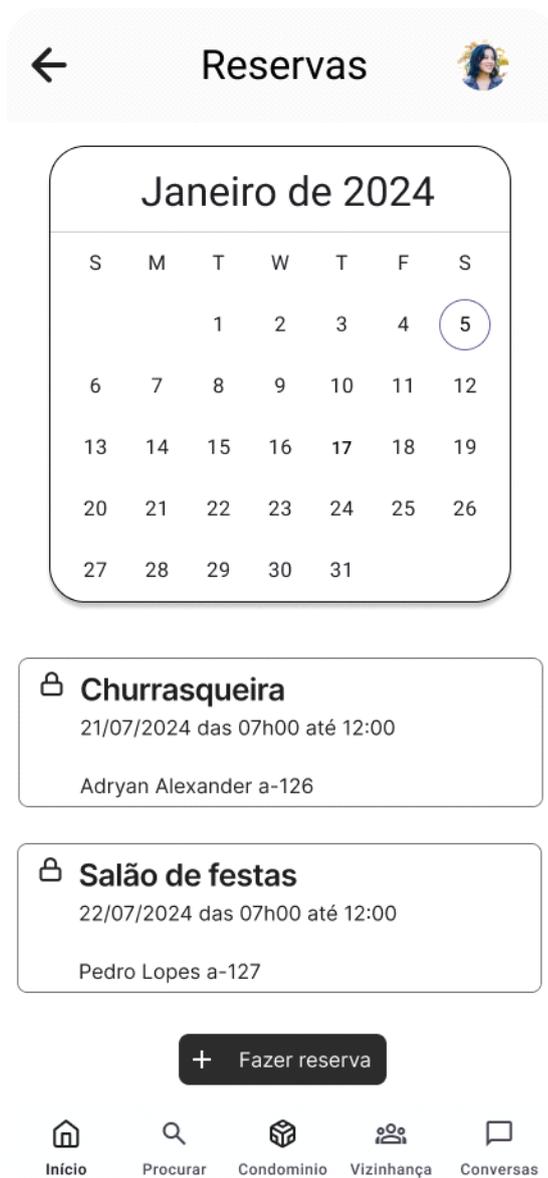
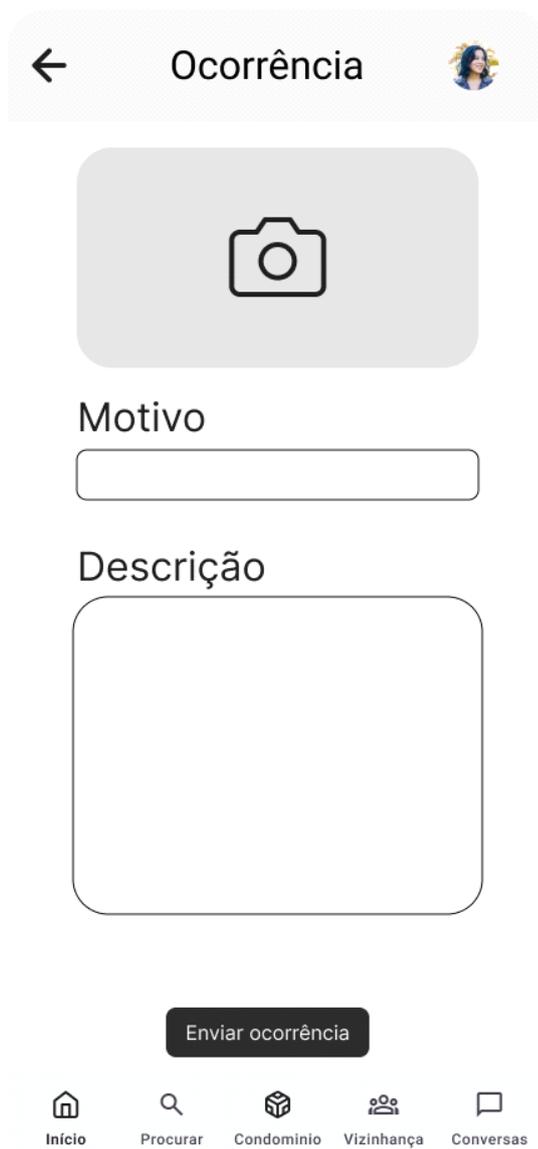


Figura 10 - Wireframe Tela dados Reservas



← Ocorrência 



Motivo

Descrição

Enviar ocorrência

 Início  Procurar  Condomínio  Vizinhança  Conversas

Figura 11 - Wireframe Tela Ocorrência



Figura 12 - Wireframe Tela Assembleia

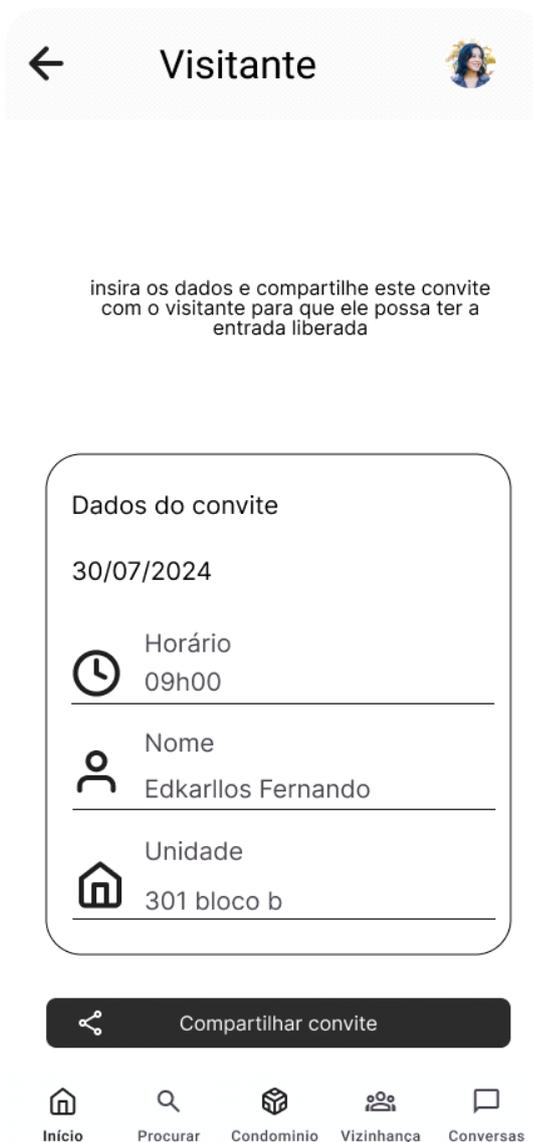


Figura 13 - Wireframe Tela Visitante



Figura 14 - Wireframe Tela de Despesas



Figura 15 - Wireframe Tela de Encomendas

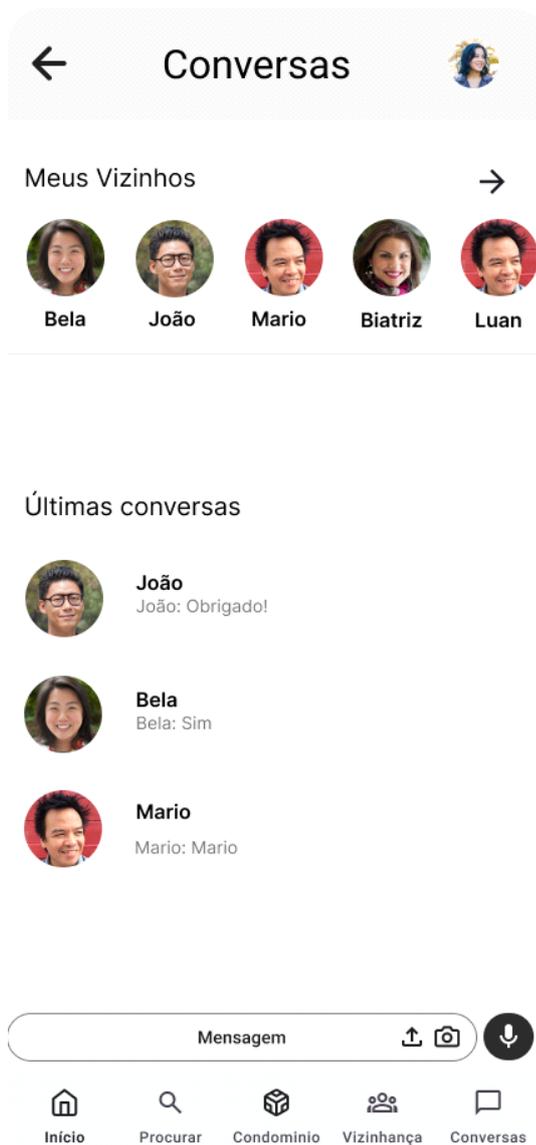


Figura 16 - Wireframe Tela de Conversas

4.9 Prints das Telas

14:29

← Criar condomínio

Dados do Condomínio

Nome do Condomínio

Endereço

CNPJ

E-mail

Telefone

Descrição

CONTINUAR

Figura 17 - Tela criar condomínio



Criar Conta

Já possui conta? Entrar

CRIAR CONTA

Criar Condomínio

[Veja Termos de uso e Política de privacidade](#)



Figura 18 - Tela de cadastro



Login

[Esqueceu a senha?](#)

[Criar conta](#)

ENTRAR

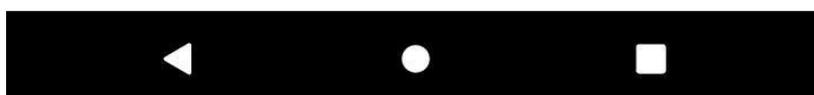


Figura 19 - Tela login

14:31

← Editar Perfil

N

Nome

Nicholas

Telefone

Salvar Alterações

Figura 20 - Tela perfil adm



Figura 21- Tela configuração

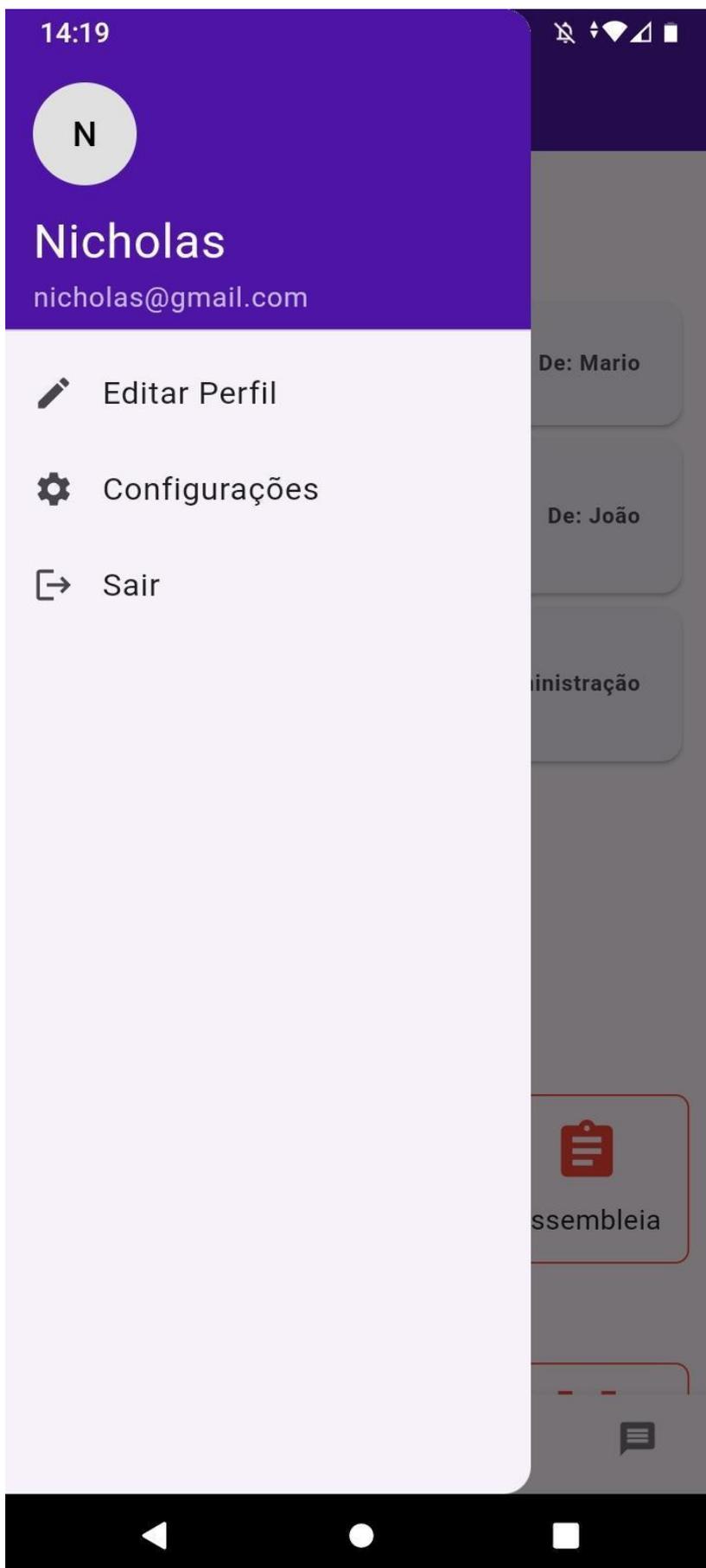


Figura 22 - Menu adm



Figura 23 - Tela de despesas

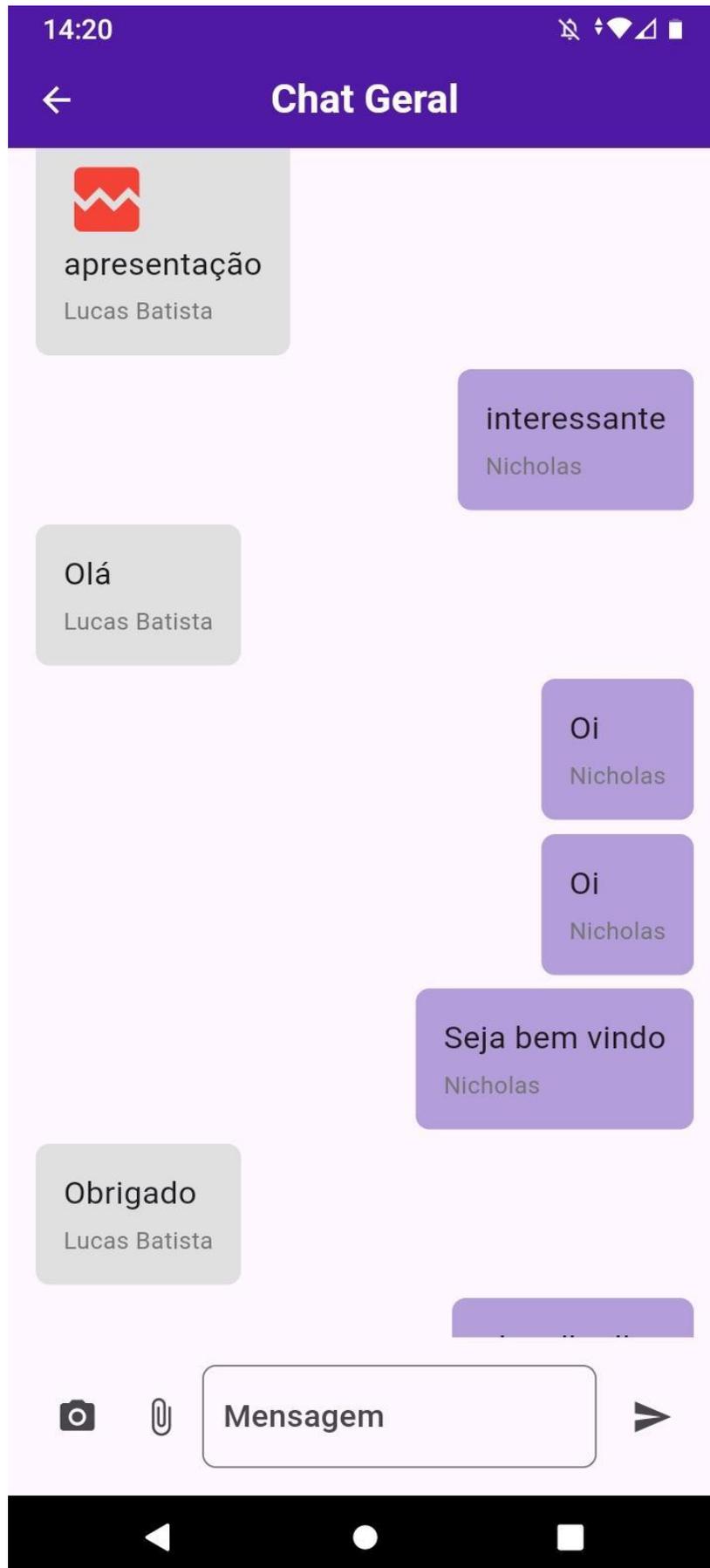


Figura 24 - Tela chat geral

14:20

← **Ocorrências**

Relatar Nova Ocorrência

Motivo

Descrição

Selecionar Data 

Selecionar Imagem 

➤ Enviar Ocorrência

Figura 25 - Tela de ocorrência

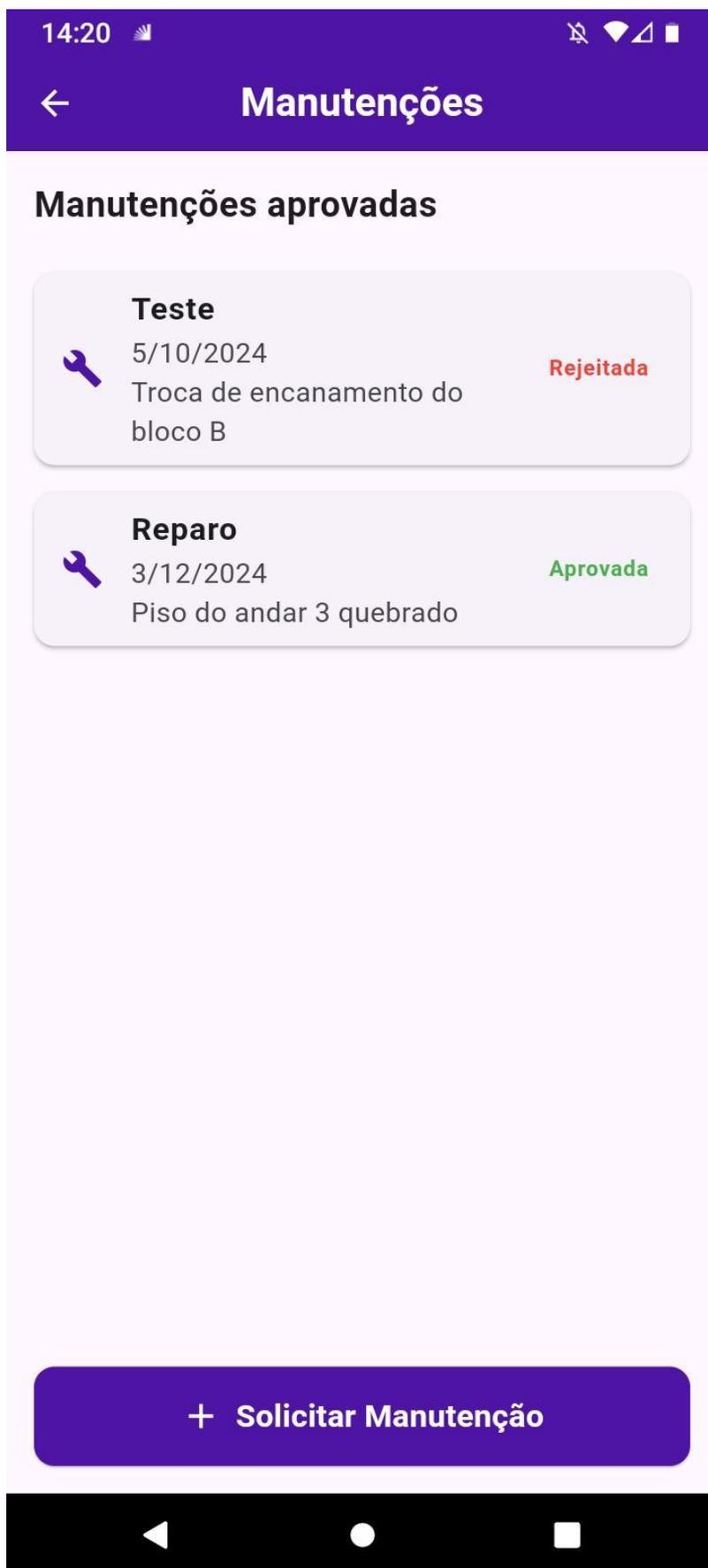


Figura 26 - Tela de Manutenção



Figura 27 - Tela de reserva

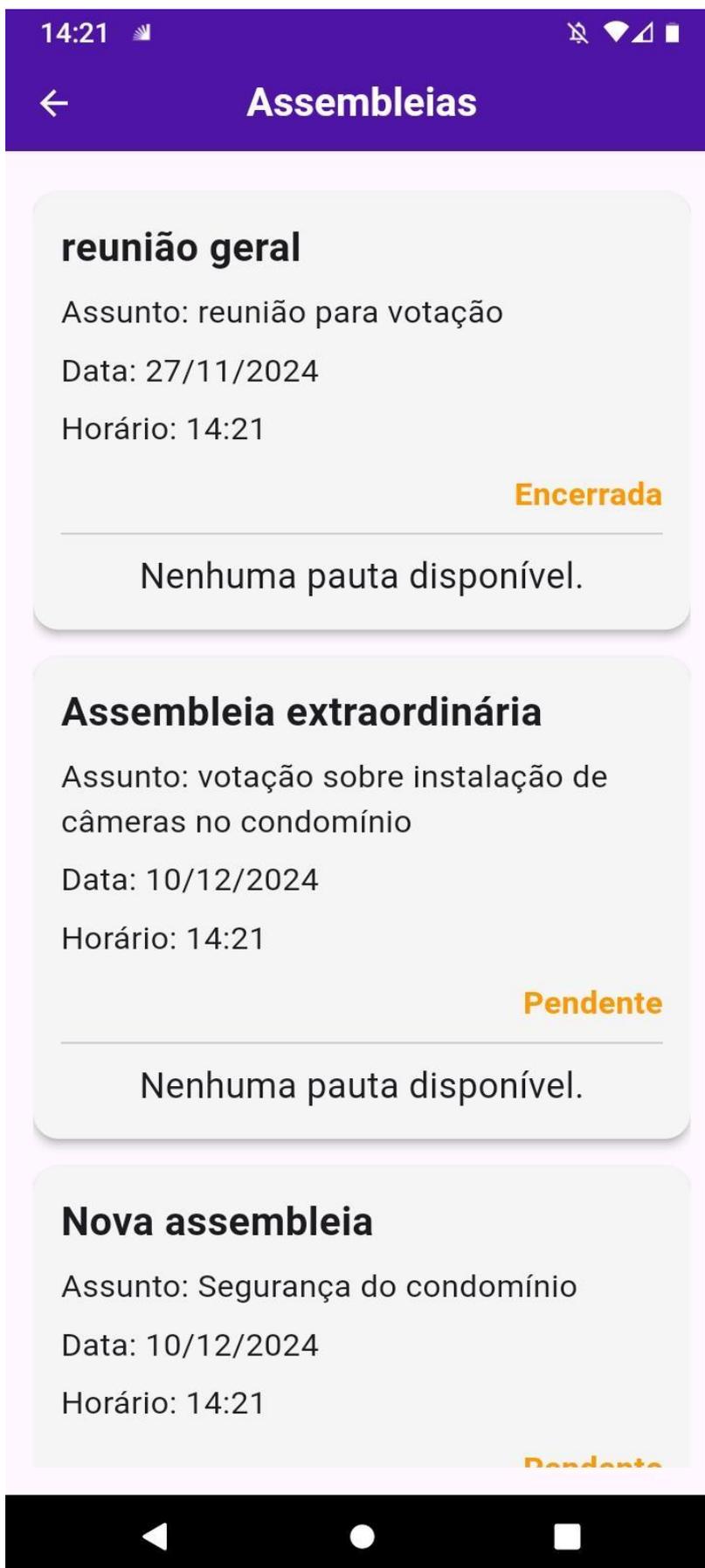


Figura 28 - Tela de assembleias



Figura 29 - Tela encomendas detalhes

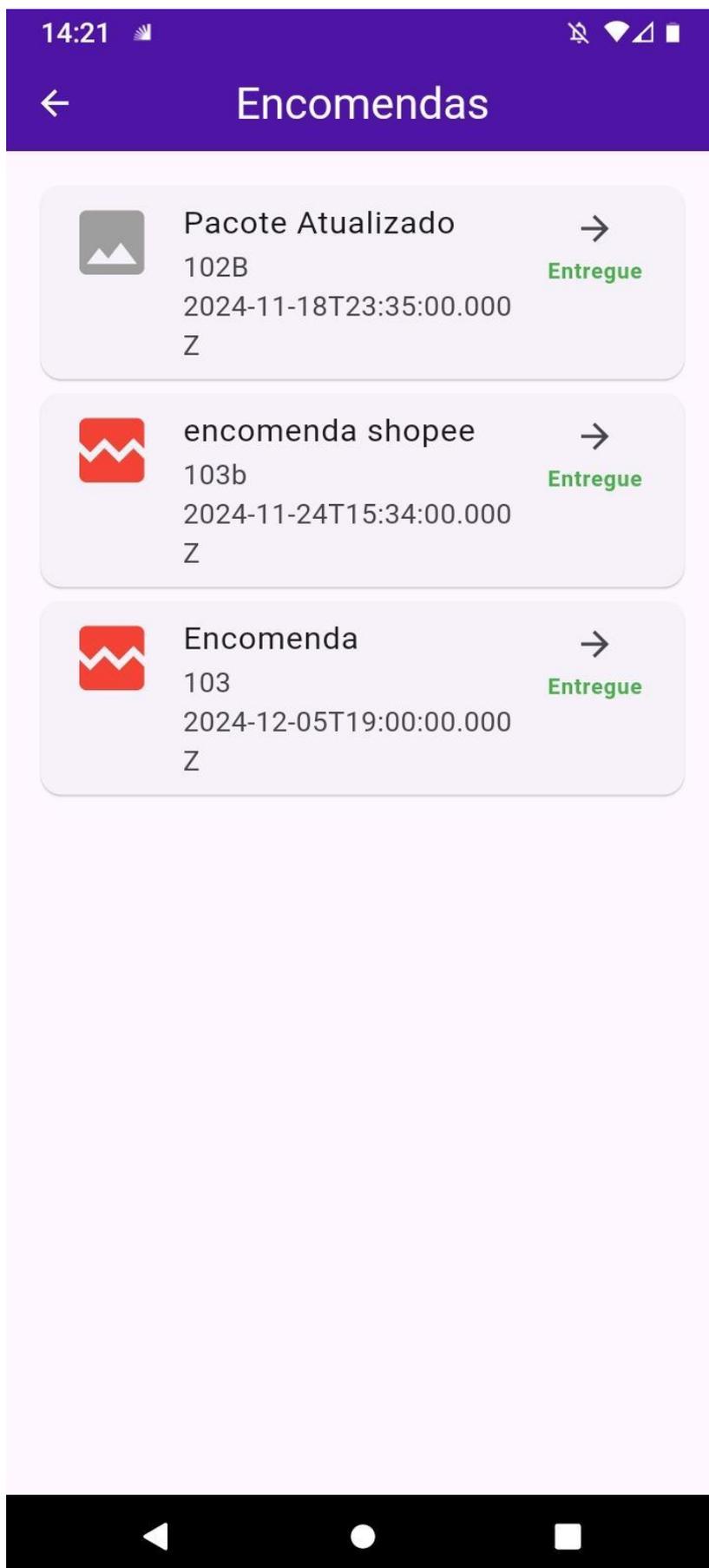


Figura 30 - Tela encomendas geral



Figura 31 - Tela inicial adm

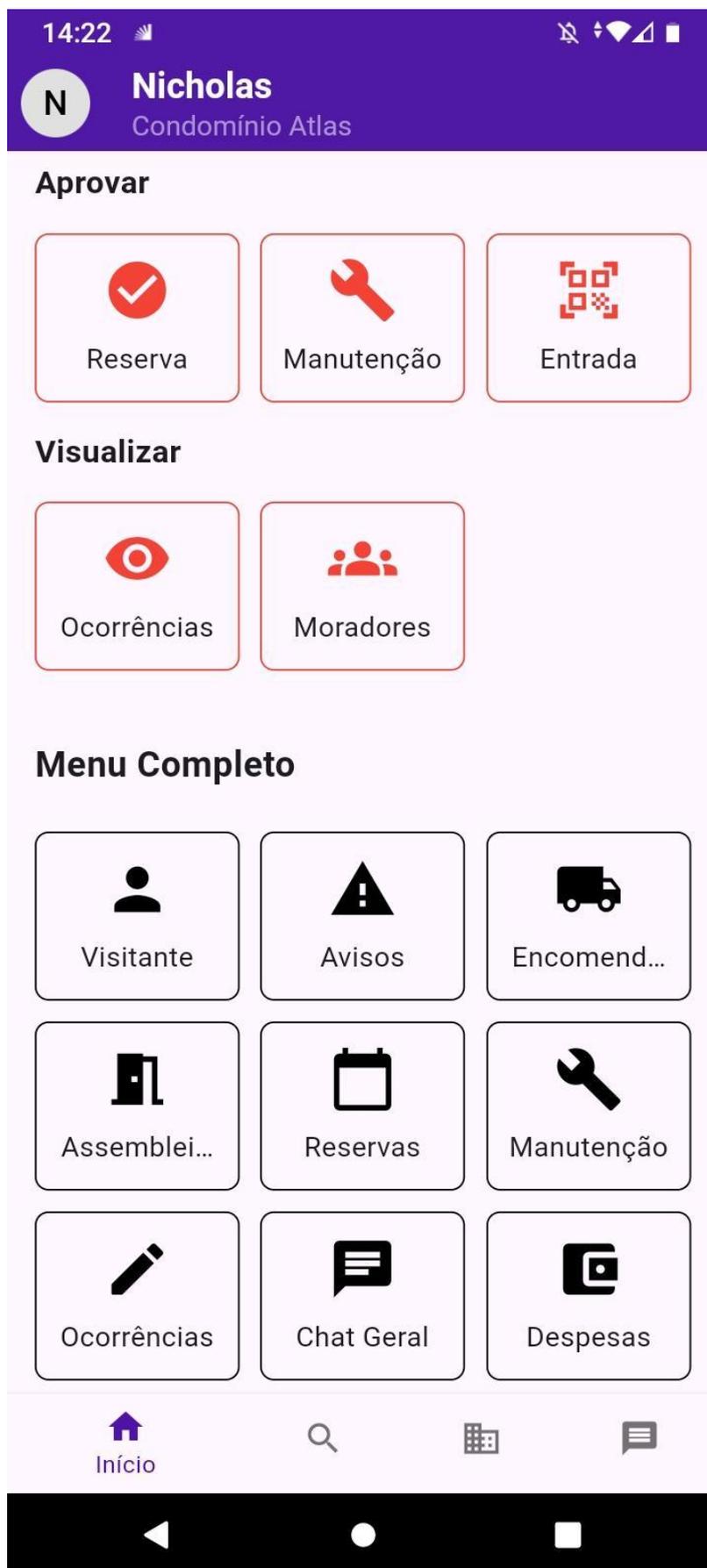


Figura 32 - Menu segunda parte



Figura 33 - Tela conversas



Figura 34 - Informações do condomínio

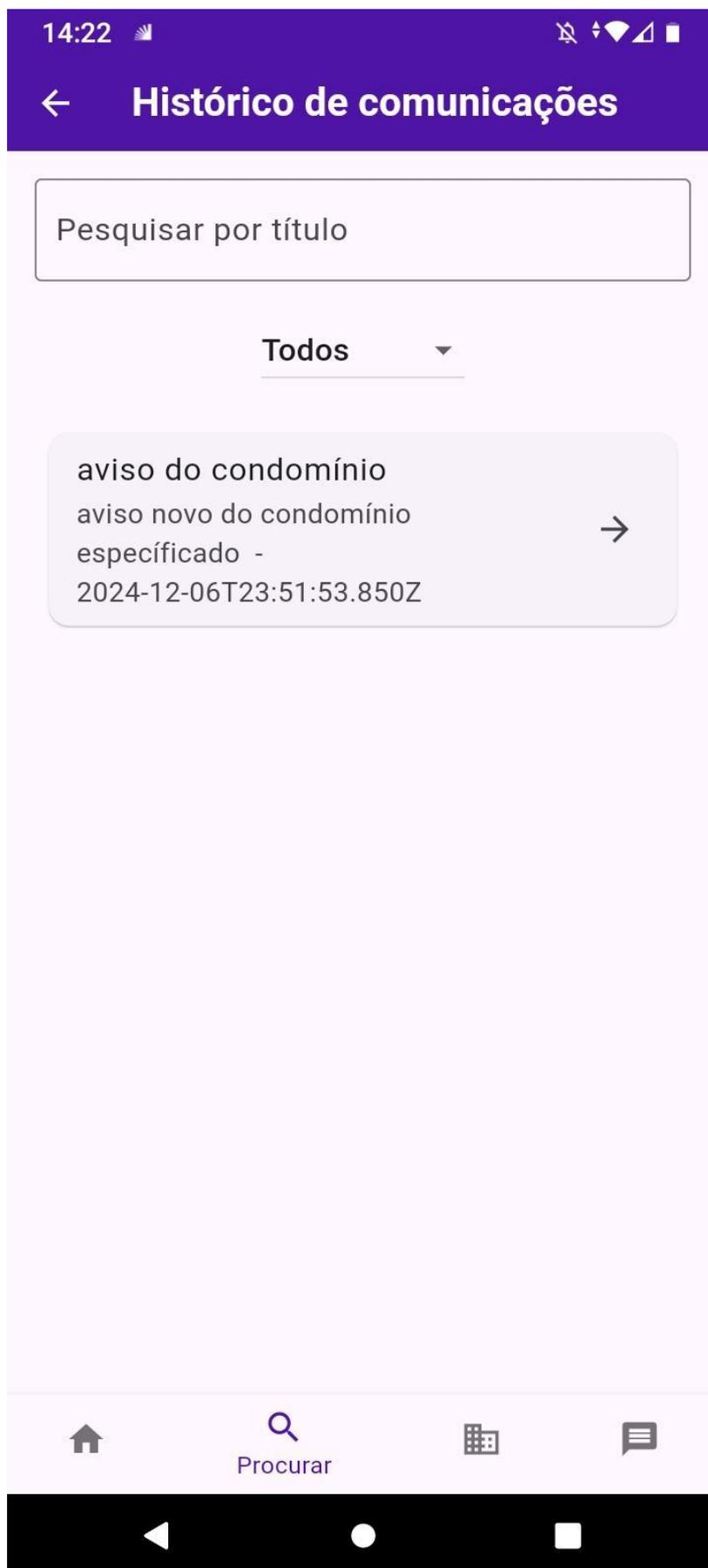


Figura 35 – Histórico

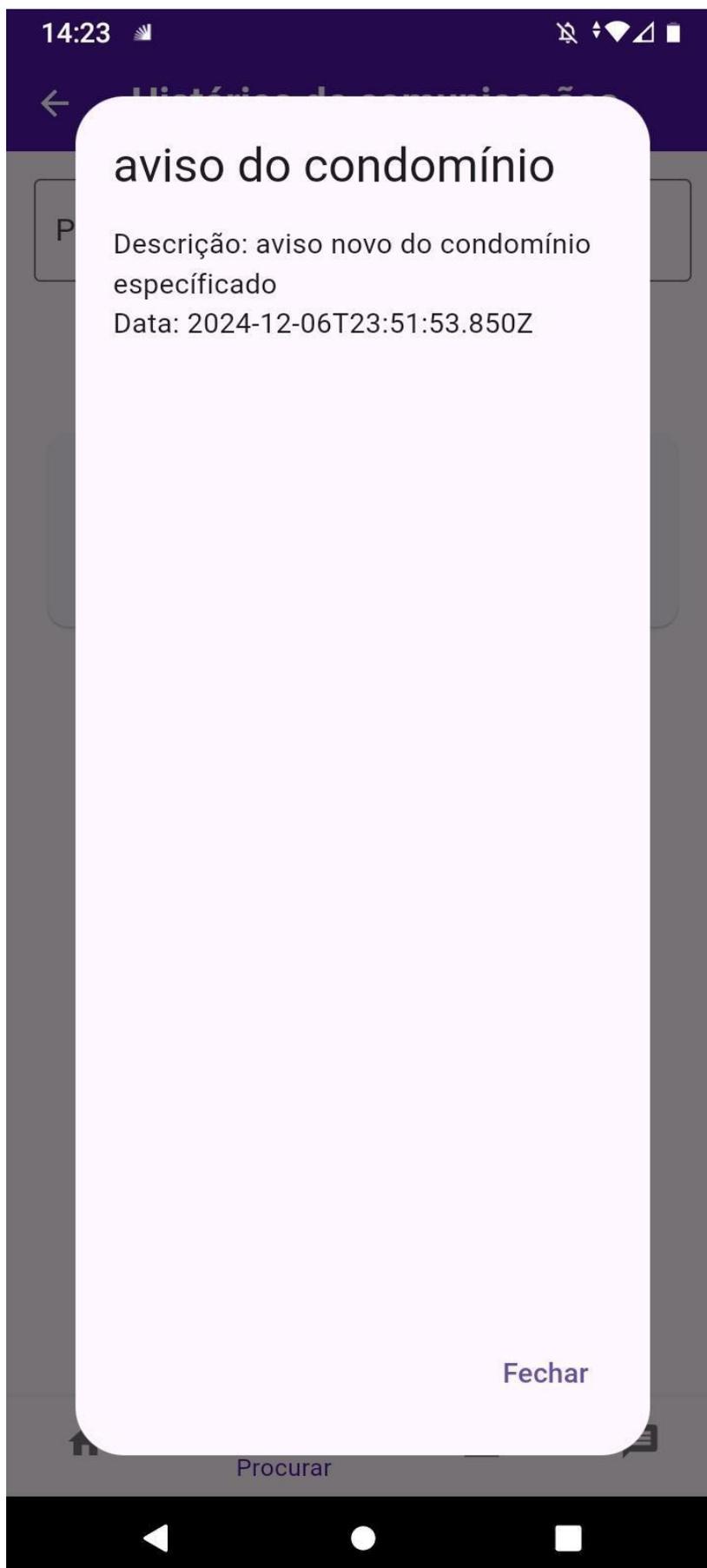


Figura 36 – Notificações

14:23

← Visitante

Insira os dados e compartilhe este convite com o visitante para que ele possa ter a entrada liberada

Dados do convite

📅 Data

🕒 Hora

👤 Nome do Visitante

🏠 Unidade

🔗 **Compartilhar Convite**

Figura 37 - Tela visitante



Figura 38 – Avisos

4.10 Trecho do Código Fonte

```
const User = require('../models/User');
const Assembly = require('../models/Assembly');
const mongoose = require("mongoose");

const createAssembly = async (req, res) => {
  const { title, description, date } = req.body;

  try {
    const currentDateTime = new Date();
    let status = "Pendente";

    if (new Date(date) < currentDateTime) {
      status = "Encerrada";
    } else if (new Date(date) === currentDateTime) {
      status = "Em Andamento";
    }

    const newAssembly = await Assembly.create({
      title,
      description,
      date,
      status,
    });

    res.status(201).json(newAssembly);
  } catch (error) {
    console.error("Erro ao criar a assembleia:", error);
    res.status(500).json({ errors: ["Houve um erro ao criar a assembleia.",
error.message] });
  }
};
```

```

const getAssemblies = async (req, res) => {
  try {
    const assemblies = await Assembly.find();

    const currentDateTime = new Date();
    assemblies.forEach((assembly) => {
      if (new Date(assembly.date) < currentDateTime) {
        assembly.status = "Encerrada";
      } else if (new Date(assembly.date) > currentDateTime) {
        assembly.status = "Pendente";
      } else {
        assembly.status = "Em Andamento";
      }
    });

    res.status(200).json(assemblies);
  } catch (error) {
    console.error("Erro ao buscar assembleias:", error);
    res.status(500).json({ errors: ["Houve um erro ao buscar assembleias.",
error.message] });
  }
};

```

```

const updateAssembly = async (req, res) => {
  const { id } = req.params;
  const { title, description, date } = req.body;

  try {
    const assembly = await Assembly.findByIdAndUpdate(
      id,
      { title, description, date },
      { new: true, runValidators: true }
    );
  }
};

```

```
    if (!assembly) {
      return res.status(404).json({ errors: ["Assembleia não encontrada."] });
    }

    res.status(200).json(assembly);
  } catch (error) {
    res.status(422).json({ errors: ["Erro ao atualizar a assembleia."] });
  }
};

const deleteAssembly = async (req, res) => {
  const { id } = req.params;

  try {
    const assembly = await Assembly.findByIdAndDelete(new
mongoose.Types.ObjectId(id));

    if (!assembly) {
      return res.status(404).json({ errors: ["Assembleia não encontrada."] });
    }

    res.status(200).json({ message: "Assembleia deletada com sucesso." });
  } catch (error) {
    res.status(500).json({ errors: ["Erro ao deletar a assembleia."] });
  }
};

module.exports = {
  createAssembly,
  getAssemblies,
  deleteAssembly,
  updateAssembly,
};

const mongoose = require("mongoose");
```

```
const { Schema } = mongoose;

const AssemblySchema = new Schema(
  {
    title: {
      type: String,
      required: true,
    },
    description: {
      type: String,
      required: true,
    },
    date: {
      type: Date,
      required: true,
    },
    status: {
      type: String,
      enum: ["Pendente", "Em Andamento", "Encerrada"],
      default: "Pendente",
    },
  },
  {
    timestamps: true,
  }
);

const Assembly = mongoose.model("Assembly", AssemblySchema);

module.exports = Assembly;

const express = require("express");
```

```
const router = express.Router();

const multer = require("multer");

const path = require("path");

const storage = multer.diskStorage({
  destination: function (req, file, cb) {
    cb(null, "uploads/assembly");
  },
  filename: function (req, file, cb) {
    cb(null, file.fieldname + "-" + Date.now() +
path.extname(file.originalname));
  }
});

const upload = multer({ storage: storage });

const {
  createAssembly,
  getAssemblies,
  deleteAssembly,
  updateAssembly,
} = require("../controllers/AssemblyController");
```

```
const validate = require("../middlewares/handleValidation");

const { assemblyValidation } =
require("../middlewares/assemblyValidation");

const authGuard = require("../middlewares/authGuard");

router.post("/assemblies",
upload.single("imagePath"),
assemblyValidation(), validate, createAssembly);

router.get("/admin/assemblies", getAssemblies);

router.delete("/admin/assemblies/:id", deleteAssembly);

router.put("/admin/assemblies/:id",
upload.single("imagePath"), validate,
updateAssembly);

module.exports = router;
```

4. MANUAL DO USUÁRIO

- Primeiro Acesso
 - a. Acesse o aplicativo Condoview.
 - b. Clique em "**Criar Conta**".
 - c. Insira suas informações pessoais.
 - d. Crie uma senha segura.
 - e. Confirme seu cadastro.
- Login
 - a. Insira seu e-mail e senha na tela de login.
 - b. Clique em "Entrar".

✓ . Funcionalidades Gerais

Cadastro e Gerenciamento de Moradores

Descrição: Permite cadastrar, editar ou excluir informações de moradores do sistema.

- Cadastrar Morador:
 - a. Acesse a aba "Gerenciar Moradores".
 - b. Clique em "Adicionar Morador".
 - c. Preencha os campos obrigatórios
 - d. Clique em "Salvar".
- Editar Informações de Moradores:
 - e. Acesse a aba "Gerenciar Moradores".
 - f. Clique no morador desejado.
 - g. Atualize os dados necessários e clique em "Salvar".
- Excluir Morador:
 - h. Acesse a aba "Gerenciar Moradores".
 - i. Selecione o morador e clique em "Excluir".
 - j. Confirme a exclusão.

Comunicação Interna

Descrição: Ferramenta para mensagens privadas ou em grupo.

- a. Acesse a aba "**Chat**".
- b. Escolha o destinatário ou acesso o chat geral.
- c. Digite sua mensagem e clique em "**Enviar**".

- d. Você também pode editar ou excluir mensagens enviadas, clicando no ícone correspondente.

Publicação de Avisos e Comunicados

Descrição: Permite criar, editar ou excluir avisos importantes.

1. Clique em "**Avisos**" no menu principal.
2. **Novo Aviso:**
 - a. Clique em "**Novo Aviso**", insira o título, descrição e data de expiração.
 - b. Adicione anexos, se necessário, e clique em "**Publicar**".
3. **Editar Aviso:** Selecione o aviso, edite as informações e salve.
4. **Excluir Aviso:** Selecione o aviso e clique em "**Excluir**".

Reserva de Áreas Comuns

Descrição: Gerencia as reservas, permitindo que administradores aceitem ou rejeitem solicitações.

1. Acesse a aba "**Reservas**".
2. Escolha a área desejada, data e hora e clique em "**Reservar**".
3. **Para Administradores:**
 - a. Revise as solicitações pendentes.
 - b. Aprove ou rejeite as reservas conforme necessário.

Gestão Financeira

Descrição: Administre receitas, despesas e boletos, e gere relatórios financeiros.

1. Acesse a aba "**Finanças**".
2. Consulte receitas, despesas ou relatórios detalhados.

Cadastro de Visitantes

Descrição: Cadastre visitantes e edite ou exclua informações.

1. Acesse a aba "**Visitantes**".
2. Clique em "**Cadastrar Visitante**", insira os dados e clique em "**Salvar**".
3. Para editar ou excluir, acesse o visitante cadastrado e selecione a opção desejada.

Registro de Ocorrências

Descrição: Permite registrar e gerenciar ocorrências.

1. Clique em "**Ocorrências**" no menu.
2. Insira os detalhes e clique em "**Enviar**".

3. Administre ocorrências com opções de edição ou exclusão.

Visualização de Histórico

Descrição: Consulte e gerencie registros passados.

1. Acesse a aba "**Histórico**".
2. Filtre por categoria (avisos, ocorrências, etc.).

Personalização de Perfil

Descrição: Atualize informações pessoais e preferências.

1. Clique em "Meu Perfil".
2. Atualize foto, nome, tema e outras informações.
3. Salve as alterações.

Geração de Relatórios Gerenciais

Descrição: Gera relatórios financeiros e operacionais.

1. Acesse a aba "**Relatórios**".
2. Escolha o tipo de relatório e o período.
3. Clique em "**Gerar**".
4. Relatórios podem ser exportados ou excluídos.

Inativação de Usuários

Descrição: Para usuários que não fazem mais parte do condomínio.

1. Acesse a aba "**Gerenciar Usuários**".
2. Selecione o usuário e clique em "**Inativar**".

Cadastro e Gerenciamento de Condomínios

Descrição: Adicione ou edite informações do condomínio.

1. Clique em "**Cadastrar Condomínio**".
2. Insira os dados e clique em "**Salvar**".
3. Edite ou exclua informações na aba de gestão.

Solicitação e Gerenciamento de Manutenções

Descrição: Gerencie solicitações de manutenção.

1. Moradores podem registrar pedidos na aba "**Manutenção**".
2. Administradores revisam, aceitam ou rejeitam as solicitações.

Gerenciamento de Correspondências

Descrição: Cadastre e gerencie correspondências recebidas.

1. Acesse a aba "**Correspondências**".
2. Registre as correspondências ou edite/exclua registros existentes.

Cadastro e Administração de Assembleias

Descrição: Registre reuniões e informe os moradores.

1. Clique em "**Cadastrar Assembleia**".
2. Insira os detalhes e salve.
3. Gerencie assembleias com opções de edição ou exclusão.

CONCLUSÃO

Com o crescente número de condomínios residenciais e comerciais no Brasil, a demanda por soluções que modernizem e otimizem a gestão condominial é evidente. O desenvolvimento do Sistema de Gestão de Condomínios proposto neste projeto apresenta uma resposta concreta a essa necessidade, abordando os desafios enfrentados por síndicos e moradores, como a falta de clareza nos processos, dificuldades de comunicação e métodos tradicionais de administração.

A plataforma desenvolvida cumpre o objetivo de integrar funcionalidades que promovem maior eficiência e transparência na gestão. Com recursos para otimizar processos administrativos, melhorar a comunicação, garantir a transparência financeira e facilitar o acompanhamento das manutenções, a solução apresentada contribui significativamente para a melhoria da qualidade de vida dos moradores e para a simplificação das atividades do síndico. Portanto, este projeto não apenas atende às demandas atuais dos condomínios, mas também posiciona a gestão condominial como um exemplo de inovação e modernidade, oferecendo uma ferramenta que é ao mesmo tempo prática, acessível e alinhada às transformações tecnológicas que marcam a sociedade contemporânea.

REFERÊNCIA BIBLIOGRÁFICA

BELANDI, Caio. Censo 2022: Cerca de oito a cada dez pessoas moravam em casas, mas cresce proporção de moradores em apartamentos. **Agência de Notícias - IBGE**. Disponível em: <https://agenciadenoticias.ibge.gov.br/agencia-noticias/2012-agencia-de-noticias/noticias/39239-censo-2022-cerca-de-oito-a-cada-dez-pessoas-moravam-em-casas-mas-cresce-proporcao-de-moradores-em-apartamentos#:~:text=Os%20domic%C3%ADlios%20do%20tipo%20%E2%80%9Ccasa,87%2C2%25%20da%20popula%C3%A7%C3%A3o>. Acesso em: 15 de Abril, 2024.

CASTRO, Bruna. Pesquisa mostra que reajuste de condomínios no Rio supera a inflação do período e preocupa. **Terra**. 2023. Disponível em:

CONDOMÍNIO RESIDENCIAL. **Grupo Unifort**, 2016. Disponível em: <https://unifortseguranca.com.br/condominio-residencial/>. Acesso em: 28 de Abril, 2024

COUTINHO, Thiago. Conheça o MongoDB e aproveite a flexibilidade e a escalabilidade dos bancos de dados NoSQL. **Voitto**, 2023. Disponível em: <https://voitto.com.br/blog/artigo/o-que-e-mongodb>. Acesso em: 30 de Abril, 2024

CUNHA, André. React Native: o que é e tudo sobre o Framework. **Alura**. 2023. Disponível em: <https://www.alura.com.br/artigos/react-native>. Acesso em: 29 de Abril, 2024.

GOLÇALVES, Ariane. O que é CSS? Guia básico para iniciantes. **Hostinger**, 2022. Disponível em: <https://www.hostinger.com.br/tutoriais/o-que-e-css-guia-basico-de-css>. Acesso em: 29 de Abril, 2024.

GUEDES, Marylene. O que é e como começar com C# (C Sharp)? **Treina Web**, 2018. Disponível em: <https://www.treinaweb.com.br/blog/o-que-e-e-como-comecar-com-c-sharp>. Acesso em: 29 de Abril, 2024.

https://www.terra.com.br/noticias/brasil/pesquisa-mostra-que-reajuste-de-condominios-no-rio-supera-a-inflacao-do-periodo-e-preocupa_6a9504c2eadbbc50bbf750ef07c0472frhboyf4o.html#:~:text=A%20m%C3%A9dia%20de%20aumento%20entre,h%C3%A1%20casos%20ainda%20mais%20gritantes. Acesso em: 13 de Abril, 2024.

LABOURIAU, Felipe. Canva: o que é a ferramenta, Design e uso. **Alura**, 2023. Disponível em: <https://www.alura.com.br/artigos/canva>. Acesso em: 30 de Abril, 2024

LIMA, Guilherme. Bootstrap: O que é, Documentação, como e quando usar. **Alura** 2023. Disponível em: https://www.alura.com.br/artigos/bootstrap?utm_term=&utm_campaign=%5BSearch%5D+%5BPerformance%5D++Dynamic+Search+Ads++Artigos+e+Conte%C3%BAdos&utm_source=adwords&utm_medium=ppc&hsa_acc=7964138385&hsa_cam=11384329873&hsa_grp=111087461203&hsa_ad=687448474447&hsa_src=q&hsa_tgt=aud-527303763294:dsa-2273097816642&hsa_kw=&hsa_mt=&hsa_net=adwords&hsa_ver=3&gad_source=1&gclid=EAlaIQobChMI79egllvohQMVv1RIAB26Sq_uEAAYASAAEgJS6PD_BwE. Acesso em: 29 de Abril, 2024.

LONGEN, Andrei O que é HTML? Guia Básico para Iniciantes. **Hostinger** 2023 Disponível em: <https://www.hostinger.com.br/tutoriais/o-que-e-html-conceitos-basicos>. Acesso em: 29 de Abril, 2024.

LOPES, MICHELE. O que é Figma e como USAR? **EBAC** 2023 Disponível em: <https://ebaonline.com.br/blog/o-que-e-figma-e-como-usar#:~:text=De%20acordo%20com%20seu%20pr%C3%B3prio,ou%20instala%C3%A7%C3%A3o%20de%20algum%20software>. Acesso em: 29 de Abril, 2024.

LOVIZON, Diego. O Crescimento dos Condomínios no Brasil: Uma Análise Abrangente. **Alcoweb**. 2024. Disponível em: <https://acolweb.com.br/blog/o-crescimento-dos-condominios-no-brasil-uma-analise-abrangente-24/#:~:text=Os%20condom%C3%ADnios%20residenciais%20s%C3%A3o%20os,%>

[HYPERLINK "https://acolweb.com.br/blog/o-crescimento-dos-condominios-no-brasil-uma-analise-abrangente-24/"2C%20representam%20cerca%20de%205%25](https://acolweb.com.br/blog/o-crescimento-dos-condominios-no-brasil-uma-analise-abrangente-24/). Acesso em: 12 de Abril, 2024

MARTINS, Vinicius. ASP.NET: o que é e como usar para criar páginas dinâmicas? **Blog Betrybe**, 2021. Disponível em: <https://blog.betrybe.com/framework-de-programacao/asp-net-o-que-e/>. Acesso em: 29 de Abril, 2024.

MATOS, Pedro. Pandemia faz explodir procura por casas em condomínios fechados. **Abreu Digital**, 2021. Disponível em: <https://abreu.digital/pandemia-faz-explodir-procura-por-casas-em-condominios-fechados/#:~:text=A%20necessidade%20de%20se%20manter,em%20rela%C3%A7%C3%A3o%20aos%20nossos%20lares>. Acesso em: 14 de Abril, 2024

MELO, Diego. O que é JavaScript? [Guia para iniciantes]. **Tecnoblog**, 2023. Disponível em: <https://tecnoblog.net/responde/o-que-e-javascript-guia-para-iniciantes/>. Acesso em: 29 de Abril, 2024.

NEVES, Cardoso; FERREIRA, Thiago. Condômino e Possuidor Antissocial: Teoria e Prática. Indaiatuba: Editora Foco, 2023.

PICOLLO, Lucas. Vue JS: O que é, como funciona e vantagens: **Geekhunter**, 2020. Disponível em: <https://blog.geekhunter.com.br/vue-js-so-veja-vantagens-e-voce/>. Acesso em: 29 de Abril, 2024.

SCHWARTZ, Rosely. **Revolucionando o condomínio**. 15ª edição. São Paulo: Saraiva, 2017.