INTRODUÇÃO:

Boa Tarde! Somos a equipe do Edifício Splendido, composta por Caio Barp S. Favetti (Analista de Testes), Henrique da Silva Cardoso (Programador Front-End), João Pedro Oechsler (Arquiteto de Banco de Dados), Luis Gabriel Lima Cunha (Arquiteto de Informação), Marcos Vinicius Borgert (Programador Back-End) e Mariana Gonçalves Barretto e nesse documento falaremos sobre nossa ideia sendo apresentada, o aplicativo Tenant Helper. Ele seria um aplicativo para auxiliar a vida do morador, automatizando processos repetitivos e facilitando o recebimento de entregas.

Cada integrante do grupo ficou encarregado de uma função principal, podendo auxiliar outros integrantes que precisassem de ajuda. Caio Barp S. Favetti e Henrique da Silva Cardoso escreveram a documentação, João Pedro Oechsler modelou o banco de dados, Luis Gabriel Lima Cunha e Mariana Gonçalves Barretto fizeram o design das telas e Marcos Vinicius Borgert criou os diagramas.

OBJETIVOS:

Objetivo Geral:

Possuímos dois objetivos com esse projeto: facilitar a comunicação porteiro-inquilino, aumentando sua eficiência, e automatizar o processo repetitivo de reservas (espaço de festas, academia e vagas de garagem).

Objetivos Específicos:

1. Facilitar o envio de avisos de entregas/visitas feitos por porteiros e o recebimento do mesmo pelos inquilinos.
2. Facilitar o envio de avisos gerais feitos por Síndicos e o recebimento do mesmo pelos inquilinos.
3. Auxiliar o agendamento de uso de espaço de festas, academia e vagas de garagem.
4. Acessar área restrita para cada inquilino logar, visualizar e baixar o seu boleto de serviços referente ao condomínio(receber lembretes para pagamento).
5. Mostrar lembretes gerais do condomínio, como lembrete de reuniões, dia do lixo, etc.

Materiais

Essa seção explica a função de todas as tecnologias as quais seriam utilizadas para desenvolver o aplicativo.

Git e Github: Nós usamos o Git para um versionamento de código, e o Github para armazenar nosso trabalho.

Git: é um aplicativo de versionamento de código onde podemos fazer cópias do nosso projeto, para que, se quisermos alterar o código dele, não afete o projeto principal.

Github: é uma plataforma de hospedagem de código-fonte e arquivos com controle de versão usando o **Git**. É o local onde podemos armazenar nosso projeto, atualizá-lo sempre que novos arquivos forem adicionados e recuperá-los em caso de perda dos arquivos originais.

Dart: o Dart é uma linguagem de programação fortemente tipada,a missão inicial do Dart era substituir o JavaScript para desenvolvimento de scripts em páginas web. Hoje, a linguagem é considerada multi-paradigma

Flutter: é uma biblioteca que utiliza a linguagem Dart da Google para criação de aplicações móveis Cross Platform. Isto significa que podemos criar aplicações para Android e iOS compartilhando parte do código escrito, inclusive há possibilidade de criar aplicações Web e Desktop com Flutter.

Intellij IDEA: A plataforma IntelliJ é uma plataforma OSS desenvolvida pela JetBrains para criar IDEs e ferramentas de desenvolvedor com reconhecimento de linguagem que foi construída em Java. Ela fornece uma abordagem entre plataformas à construção de ferramentas para qualquer linguagem, seja para a JVM ou não.

SQLite: SQLite é uma biblioteca em linguagem C que implementa um pequeno, rápido, auto-suficiente, de alta confiança, engine de banco de dados SQL com todas as funcionalidades. SQLite é a engine de banco de dados mais usada no mundo. SQLite está integrada em todos os telefones celulares e na maioria dos computadores e vem integrada em inúmeras outras aplicações que as pessoas usam todos os dias.

Photoshop: Adobe Photoshop é um software editor de imagens do tipo raster desenvolvido pela Adobe Systems. É possível usá-lo em conjunto a vários outros produtos Adobe como o InDesign (edição de texto), Illustrator (edição de gráficos vetoriais), Premiere (edição de vídeo não-linear), etc.

Figma: uma ferramenta de design de interface moderna. O Figma é uma ferramenta de UI online e gratuita, feita para criar, colaborar, prototipar e inspecionar.

Métodos:

Requisitos Funcionais e não funcionais:

Identificação	Descrição
RF1	Possuir a função de envio de avisos por pessoas registradas como funcionários
RF2	Possuir a função de recebimento de avisos por pessoas registradas com inquilinos
RF3	Permitir aos usuários poderem reservar espaços de festa, academia e garagem remotamente
RF4	Possuir uma tela de usuário com informações da conta, funções para mudá-las e acesso a boletos bancários do inquilino
RF5	Mostrar lembretes de eventos que vão ocorrer no condomínio como reuniões ou dia do lixo
RNF1	Ser desenvolvido utilizando a linguagem de programação Dart e o framework Flutter
RNF2	Ter como gerenciador de banco de dados a plataforma SQLite
RNF3	Utilizar a IDE IntelliJ IDEA como ambiente de desenvolvimento

Modelo de arquitetura de desenvolvimento de software:

O modelo escolhido para seguir o desenvolvimento desse aplicativo foi o MVC. O nome do modelo é uma sigla que significa Modelo (Model), Visão (View) e Controlador (Controller). Esse modelo foi escolhido por causa da sua popularidade, o que significa mais suporte da comunidade caso algo dê errado e sugestões de implementação, por causa dos benefícios que ela oferece, como a otimização de velocidade entre as requisições feitas pelo comando de usuários e pela sua história, esse modelo de arquitetura de software já existe a mais de quarenta anos e já foi testado e aprovado por diversos aplicativos.

Vou explicar os três conceitos que englobam esse modelo:

Modelo:

Também é conhecido como Business Object Model, é utilizado para gerenciar e controlar a forma como os dados se comportam por meio das funções, lógica e regras de negócio estabelecidas. Ele recebe as informações do Controller, valida se ela está correta ou não e envia a resposta mais adequada.

Controlador:

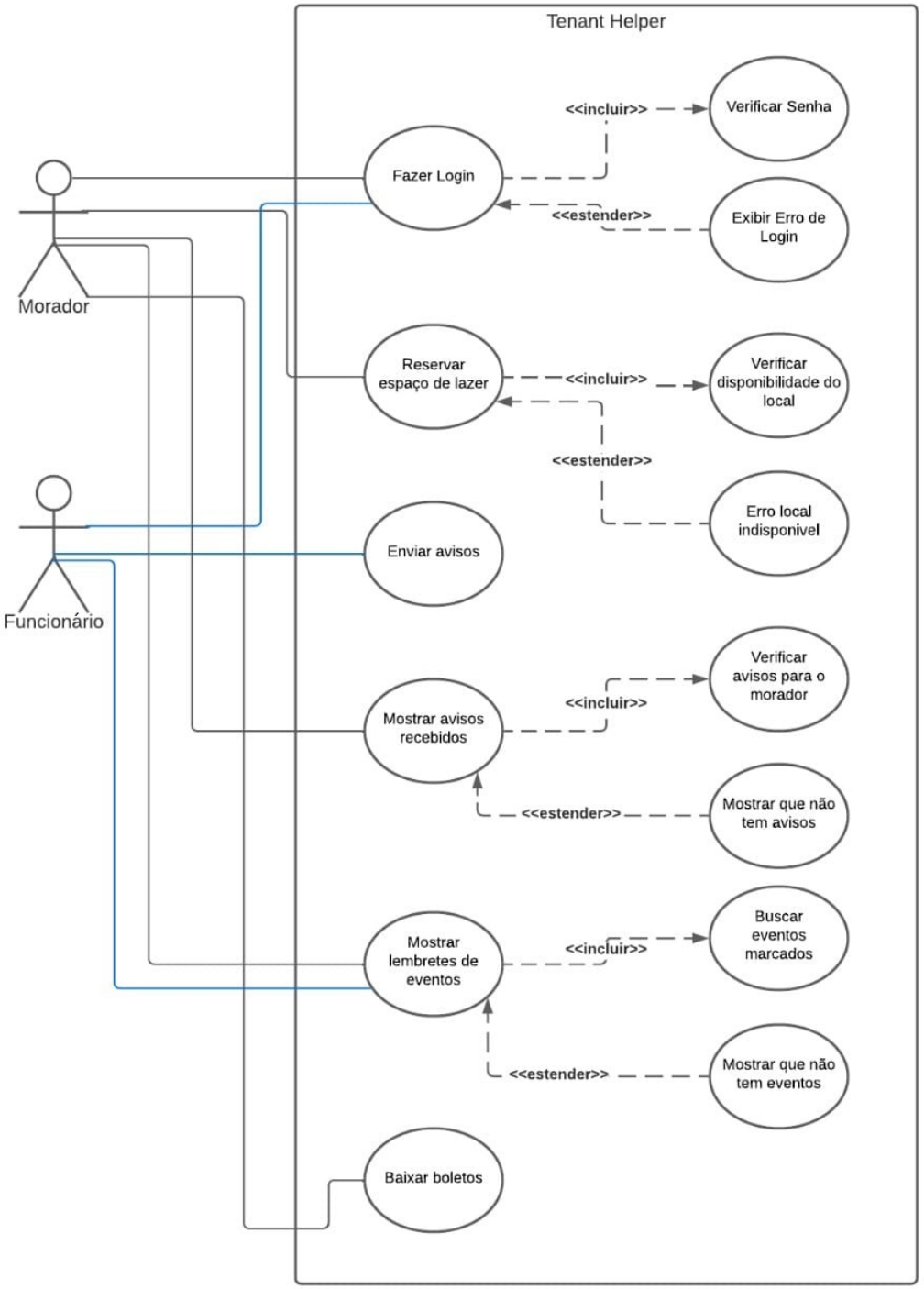
É responsável por receber e intermediar as informações requisitadas pela camada de Visão e as respostas obtidas da camada de Modelo, processando os dados que o usuário informou e os repassando para outras camadas.

Visão:

É responsável por apresentar as informações de forma visual para o usuário. Durante seu desenvolvimento devem ser usados somente elementos gráficos como mensagens, botões ou telas.

Diagramas:

Diagrama de caso de uso:



Para um maior entendimento por parte do público sobre o nosso aplicativo, foi desenvolvido um diagrama de caso de uso, dentro do retângulo, estão as funcionalidades do aplicativo, e fora dele estão os atores, que fariam a interação com o sistema, no nosso caso seriam os moradores e os funcionários.

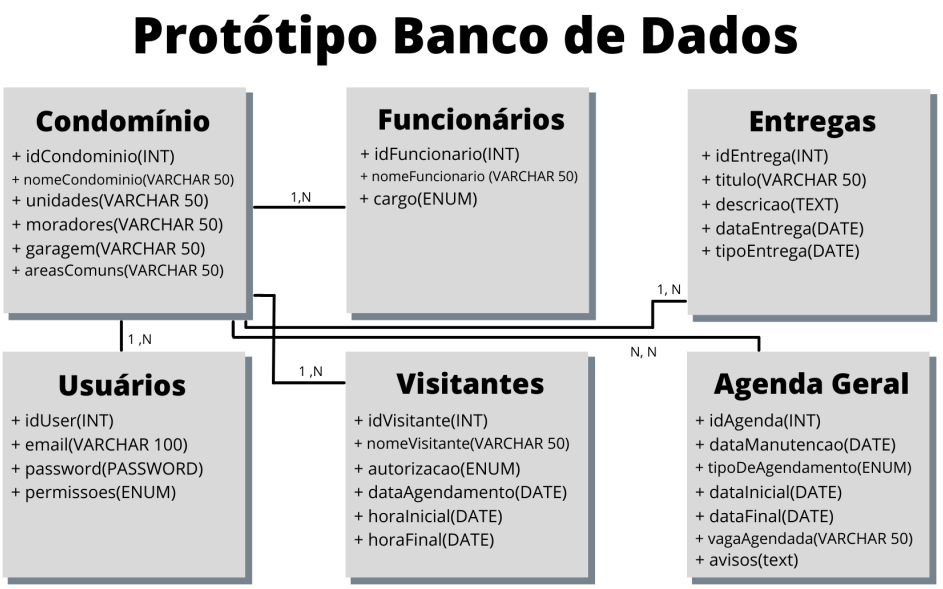
Teste de Software

Parte do processo de desenvolvimento de um software, é o Teste de Software, esses testes podem ser feitos pelo próprio desenvolvedor ou por um profissional especializado. Pensando no planejamento de nosso aplicativo, percebemos que testes serão necessários. Com uma breve pesquisa, notamos de imediato que o teste de caixa branca, o teste de caixa preta, o teste de regressão entre outros serão necessários. Conforme evoluímos nosso conhecimento, a automatização dos teste será feita, assim facilitando a vida do programador

Banco de dados

Para o banco de dados temos um protótipo de tablas junto com suas respectivas relações, colunas e tipos de variáveis usadas no planejamento, neste protótipo buscamos fazer algo simples e de fácil entendimento para que os programadores no futuro pudessem realizar manutenções sem problemas de entendimento, contudo o banco de dados seria indispensável para a aplicação já que nele ficariam armazenadas informações como os funcionários, entregas e agendamentos.

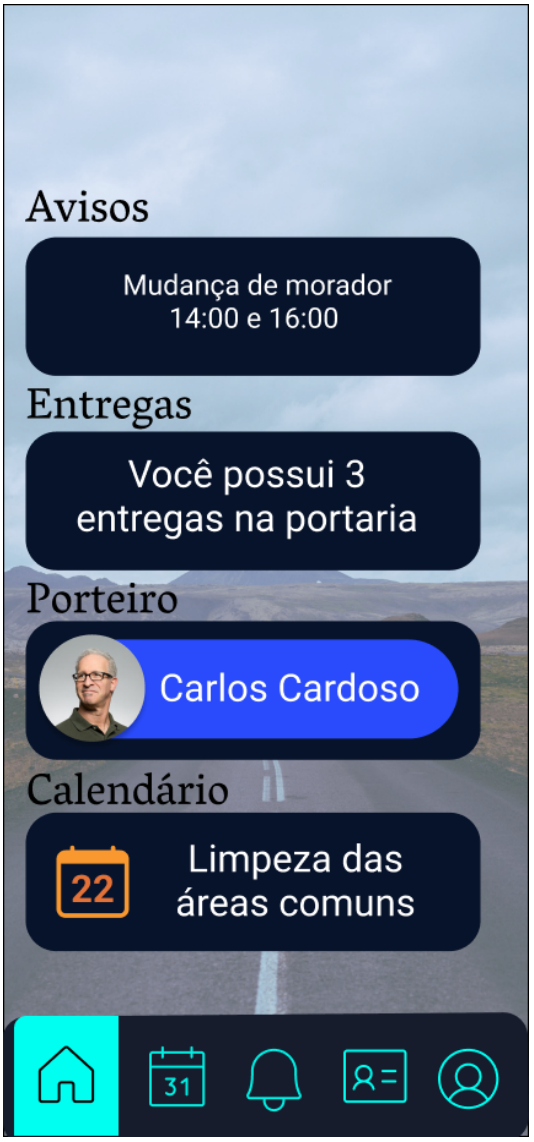
Na imagem a seguir podemos ver como funcionaria:



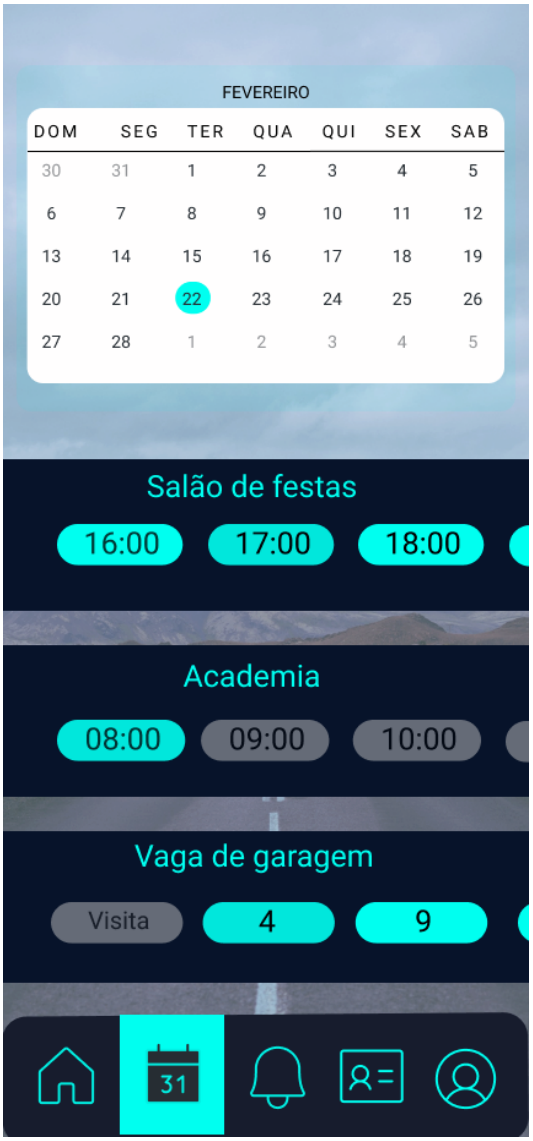
Telas

O aplicativo possui cinco telas principais: a tela Inicial, tela de Agenda, tela de Entregas, tela de Funcionários e tela de Usuário.

Tela Inicial: essa tela apresenta os avisos gerais, as entregas pendentes, o porteiro presente naquele horário de acesso e o calendário que marca os eventos ocorrendo pelo condomínio.

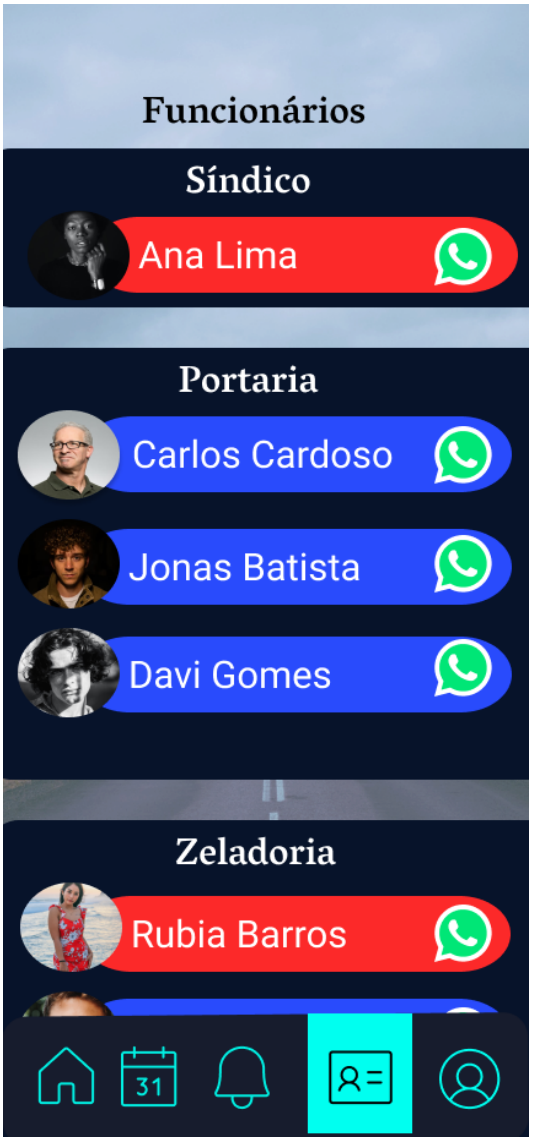


Tela Agenda: essa tela marca o dia atual em um calendário e apresenta os horários disponíveis e indisponíveis de reserva de serviços com salão de festas, academia e vaga de garagem.



Tela de Entregas: essa tela apresenta as encomendas e cartas presentes na portaria e através dela podem ser adicionados avisos de visita por parte dos moradores para o porteiro.

Tela de Funcionários: essa tela apresenta todos os funcionários do condomínio e ao pressionar o símbolo do aplicativo Whatsapp, o usuário será redirecionado à uma tela de chat de conversa com o funcionário.



Tela de Usuário: essa tela apresenta a foto do inquilino, seu nome, e-mail, senha, residencial, apartamento e vaga de garagem pertencente ao inquilino. Além disso, é possível consultar boletos já pagos, verificar boletos ainda não pagos, liberar sua vaga para outro inquilino e cancelar essa liberação.

Link de acesso ao protótipo feito através da plataforma: protótipo.

Conclusão

Todos na equipe concordaram que a produção deste trabalho foi um processo que ocorreu suavemente. Todos cumpriram seus papéis de forma excepcional e mesmo um dos integrantes chegando um pouco mais tarde, foi bem recebido e integrado na equipe rapidamente. Esperamos que você tenha gostado de

avaliar esse projeto tanto quanto gostamos de produzi-lo, obrigado pela oportunidade!