



**UNIVERSIDADE ESTADUAL DA PARAÍBA
CAMPUS I - CAMPINA GRANDE
CENTRO DE CIÊNCIAS E TECNOLOGIA
DEPARTAMENTO DE COMPUTAÇÃO
CURSO DE GRADUAÇÃO EM BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

RAFAEL DE MEDEIROS SILVA

**ROOMMATE: UMA APLICAÇÃO PARA GERENCIAMENTO DE
COMPROMISSOS DE UMA CASA COMPARTILHADA**

**CAMPINA GRANDE
2023**

RAFAEL DE MEDEIROS SILVA

**ROOMMATE: UMA APLICAÇÃO PARA GERENCIAMENTO DE
COMPROMISSOS DE UMA CASA COMPARTILHADA**

Trabalho de Conclusão de Curso apresentado ao Curso de Ciência da Computação da Universidade Estadual da Paraíba, como requisito parcial à obtenção do título de Bacharel em Ciência da Computação.

Orientador: Profa. Ma. Cheyenne Ribeiro Guedes Isidro.

CAMPINA GRANDE

É expressamente proibido a comercialização deste documento, tanto na forma impressa como eletrônica. Sua reprodução total ou parcial é permitida exclusivamente para fins acadêmicos e científicos, desde que na reprodução figure a identificação do autor, título, instituição e ano do trabalho.

S586r Silva, Rafael de Medeiros.
Roomate [manuscrito] : uma aplicação para gerenciamento de compromissos de uma casa compartilhada / Rafael de Medeiros Silva. - 2023.
54 p. : il. colorido.

Digitado.
Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) - Universidade Estadual da Paraíba, Centro de Ciências e Tecnologia, 2023.
"Orientação : Profa. Ma. Cheyenne Ribeiro Guedes Isidro, Coordenação do Curso de Computação - CCT. "
1. Desenvolvimento de software. 2. Aplicativo. 3. Solução computacional. I. Título

21. ed. CDD 005.3

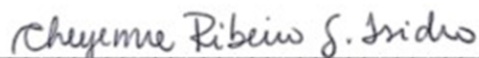
RAFAEL DE MEDEIROS SILVA

**ROOMMATE: UMA APLICAÇÃO PARA GERENCIAMENTO DE
COMPROMISSOS DE UMA CASA COMPARTILHADA**

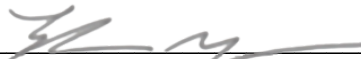
Trabalho de Conclusão de Curso apresentado ao Curso de Ciência da Computação da Universidade Estadual da Paraíba, como requisito parcial à obtenção do título de Bacharel em Ciência da Computação.

Aprovado em: 31/08/2023.

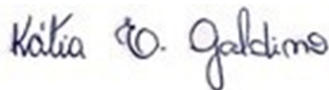
BANCA EXAMINADORA



Prof. Ma. Cheyenne Ribeiro Guedes Isidro (CCT/UEPB)
Orientador(a)



Prof. Me. Edson Holanda Cavalcante Júnior (CCT/UEPB)
Examinador(a)



Prof. Dra. Kátia Elizabete Galdino (CCT/UEPB)
Examinador(a)

AGRADECIMENTOS

Expresso profundos agradecimentos a Deus por Sua presença constante em minha jornada acadêmica. A fé inabalável iluminou cada passo e fortaleceu o meu percurso até este momento significativo do meu TCC. Reconheço a bênção de ter sido guiado por Sua graça, que me inspirou a superar desafios e a seguir acreditando na minha capacidade.

Minha trajetória acadêmica foi enriquecida por professores notáveis, e dentre eles, destaco Edson Holanda, cujas aulas de lógica no início do curso moldaram minha base, e Cheyenne Ribeiro, professora de linguagens formais e compiladores, cujo ensino excepcional foi um privilégio após meu processo de reingresso ao curso. Expresso meus sinceros agradecimentos por suas magistrais aulas e pela generosidade em compartilhar conhecimento. A dedicação que vocês demonstraram inspiraram profundamente meu entusiasmo em aprofundar meu entendimento na área da computação.

Também quero expressar minha gratidão a minha irmã Patrícia de Medeiros, matemática apaixonada por conhecimento, que tem sido minha inspiração desde a infância, e cujo apoio constante foi vital para minha jornada até aqui. Junto a ela, o suporte de minha família foi fundamental para trazer princípios e valores à minha profissão.

Minha mais profunda gratidão a você, mãe. Mesmo nos momentos em que me senti solitário em minhas conquistas, esteve sempre ao meu lado, sustentando a crença inabalável de que eu alcançaria meus objetivos. Este diploma traz consigo um simbolismo único, que é compartilhado contigo, pois as minhas vitórias também são, sem dúvida, frutos do seu apoio e dedicação incansável.

Agradeço também a Paulo Victor por seu amor e compreensão inestimáveis, um verdadeiro exemplo de disciplina e compromisso, e sua presença ao meu lado durante esta fase final do curso foi uma fonte constante de incentivo e apoio indispensáveis.

Durante o estágio, encontrei amigos que trago comigo até hoje. Em especial, desejo enfatizar Arthur Alves, um colega de estágio que se tornou um amigo para a vida. Além dele, meus outros amigos merecem meu caloroso agradecimento por tornarem a jornada mais leve do que parecia ser. Por fim, quero expressar minha profunda gratidão aos amigos que fiz durante o curso e que levo comigo para sempre. Vocês foram presentes inestimáveis da UEPB: Amanda Neves, Vanessa Portugal, Amaury Júnior, Joyce Mikaella, Pedro Ramos e Lindemberg.

“A conquista é um acaso que talvez dependa mais das falhas dos vencidos do que do gênio do vencedor.”

Madame de Staël

RESUMO

Esse trabalho consiste em um relatório técnico sobre o desenvolvimento da ferramenta ROOMMATE. Essa solução computacional surgiu da necessidade de organizar e concentrar as informações necessárias para manter uma agenda e os compromissos de uma casa compartilhada em dia, se tornando o principal guia de responsabilidades a ser seguido. Para planejamento e desenvolvimento do software, inicialmente foram definidos todos os requisitos a partir de um questionário que foi elaborado para capturar as complexidades inerentes às interações entre indivíduos que compartilham um mesmo espaço residencial. Depois, foram investigadas as tecnologias a serem utilizadas. Como este projeto é concebido para futuras versões como uma aplicação híbrida, buscou-se um framework que permitisse um desenvolvimento *mobile* através de linguagens nativas, como html e css. Dessa forma, optou-se por desenvolver o *front-end* através do *framework* Ionic. Já na estruturação do *back-end*, decidiu-se pelo Node.js pela sua eficiência de escalabilidade, permitindo lidar com várias operações concorrentes sem interromper a fluidez do programa. Para a criação e manipulação do banco de dados, recorreu-se ao Prisma ORM, uma técnica que estabelece uma relação entre objetos e os dados que eles representam. Para o sistema de gerenciamento de banco de dados, optou-se pelo PostgreSQL. Ao fim de seu desenvolvimento, verificou-se que a ferramenta ROOMMATE aborda diretamente as necessidades prementes de aprimoramento na administração da convivência em residências compartilhadas, atendendo as exigências essenciais para uma melhoria significativa na gestão e coexistência em contextos de convívio residencial compartilhado.

Palavras-Chave: desenvolvimento de software; aplicativo; mobile.

ABSTRACT

This work constitutes a technical report detailing the development of the ROOMMATE tool. This computational solution emerged from the need to systematize and consolidate the essential information required to maintain an up-to-date schedule and commitments of a shared household, effectively becoming the primary guide for adhering to responsibilities. The planning and software development process began by defining all requisites through a questionnaire, which was carefully crafted to capture the inherent complexities of interactions among individuals who share the same residential space. Subsequently, an exploration of suitable technologies was undertaken. As this project is envisaged for future iterations as a hybrid application, a framework enabling mobile development using native languages like HTML and CSS was sought. Hence, the decision was made to employ the Ionic framework for frontend development. For backend structuring, Node.js was chosen due to its scalability efficiency, enabling seamless handling of multiple concurrent operations without disrupting program fluidity. The Prisma ORM was employed for database creation and manipulation, establishing a relationship between objects and the data they represent. PostgreSQL was selected as the database management system. Upon its completion, it was evident that the ROOMMATE tool directly addresses the pressing needs for enhancing the administration of communal living within shared residences. It effectively fulfills essential requirements for substantial improvements in management and coexistence within shared residential contexts.

Keywords: software development; application; mobile.

LISTA DE ILUSTRAÇÕES

Figura 1 - Representação da quantidade de entrevistados que tiveram problemas devido ao não cumprimento das regras da casa.....	13
Figura 2 - Proporção das formas de acordo das regras da casa.....	14
Figura 3 - Apresentação da forma de como as contas são pagas na residência	15
Figura 4 - Apresentação do método de divisão das contas rateadas.....	15
Figura 5 - Apresentação do método de controle dos compromissos da casa.....	16
Figura 6 - Respostas sobre decisões tomadas em conjunto na residência.....	16
Figura 7 - Apresentação da conduta adotada para recebimento de visitas na residência.....	17
Figura 8 - Forma de disponibilização de documentos e regimento da residência.....	18
Figura 9 - Diagrama de Casos de Uso.....	22
Figura 10 - Diagrama de atividade da agenda.....	23
Figura 11 - Diagrama de atividade de contas.....	24
Figura 12 - Diagrama de Entidade-Relacionamento do aplicativo.....	25
Figura 13 - Protótipo de baixa fidelidade.....	27
Figura 14 - Logomarca da aplicação.....	28
Figura 15 - Tela de apresentação, de login e de cadastro.....	28
Figura 16 - Tela inicial (esq.) e o menu (dir.).....	29
Figura 17 - Tela de visualização (esq.) e inclusão (dir.) de contas.....	30
Figura 18 - Tela de visualização de agenda (esq.) e inclusão de novo evento (dir.).....	30
Figura 19 - Tela de visualização de enquetes (esq.) e inclusão de nova enquete (dir.).....	31
Figura 20 - Tela de visualização (esq.) e inclusão (dir.) de regras da casa.....	32
Figura 21 - Tela de visualização (esq.) e apresentação (dir.) do código da casa.....	32
Figura 22 - Tela de visualização dos visitantes (esq) e de inclusão de novo visitante (dir.).....	33
Figura 23 - Tela de configurações do aplicativo.....	34
Figura 24 - Componente ion-toggle do framework Ionic.....	37
Figura 25 - Trecho de código da classe de cadastro.....	37
Figura 26 - Tela de cadastro do host (esq.) e Tela de cadastro do hóspede (dir.)	38
Figura 27 - Tela de visualização da base de dados através do Prisma ORM	39
Figura 28 - Tela de cadastro de conta (esq.) e interface principal do menu de contas (dir.)....	39
Figura 29 - Tela de menu lateralizado da aplicação.....	40

Figura 30 -Tela de inclusão de conta com seleção de tipo (esq.) e seleção de data (dir.).....	40
Figura 31 - Tela de visualização de contas.....	41
Figura 32 - Tela de inclusão de novo compromisso.....	42
Figura 33 - Tela da agenda do dia 26/08 (esq.) e tela da agenda do dia 31/08 (dir.).....	43
Figura 34 -Tela de cadastro de conta (esq.) e interface principal do menu de contas (dir.).....	44
Figura 35 -Tela de cadastro de enquete (esq.) e interface principal de enquetes (dir.).....	45
Figura 36 - Tela de inclusão de nova regra.....	45
Figura 37 - Tela de apresentação de regras da casa.....	46
Figura 38 - Tela de apresentação de código da residência para o usuário <i>host</i>	47
Figura 39 - Tela de cadastro de visitante (esq.) e interface de visitantes cadastrados (dir.).....	48
Figura 40 - Tela inicial do aplicativo após efetuação de cadastros.....	49

LISTA DE QUADROS

Quadro 1 – Requisitos funcionais (RF).....	18
Quadro 2 – Requisitos não-funcionais (RNF).....	21

SUMÁRIO

1. INTRODUÇÃO.....	11
2. LEVANTAMENTO DE REQUISITOS.....	13
2.1. Requisitos funcionais.....	18
2.2. Requisitos não-funcionais.....	21
3. MODELAGEM DO SISTEMA.....	22
4. DESENVOLVIMENTO DO FRONT-END.....	26
4.1. Protótipo de baixa fidelidade da interface.....	26
4.2. Protótipo de alta fidelidade da interface.....	27
4.3. Programação do front-end.....	34
5. DESENVOLVIMENTO DO BACK-END.....	36
6. CONSIDERAÇÕES FINAIS.....	50
REFERÊNCIAS.....	51
APÊNDICE A – QUESTIONÁRIO APLICADO PARA LEVANTAMENTO DE REQUISITOS.....	52

1. INTRODUÇÃO

É cada vez mais comum o ingresso de alunos em Instituições de Ensino Superior localizadas em cidades vizinhas de onde os estudantes moram, ou até mesmo em outros estados. Muitas das vezes, isto é possível através de um acesso pendular pelo aluno, ou seja, do ato de ir e vir entre os dois municípios. Essa modalidade geralmente é possível quando o deslocamento é viável para manter a rotina diária de viagens e estudos. Porém, essa rotina é impraticável para alguns estudantes, seja pelo tempo de viagem, ou mesmo pelo alto custo do transporte. Dessa forma, passam a surgir muitas migrações territoriais, onde o indivíduo muda de sua cidade ou região, deixando a casa da família e passando a morar na cidade em que fará a sua graduação. Toda essa conjuntura gera a necessidade da definição de novos pontos de apoio social e financeiro, os quais se traduzem em novos vínculos de amizade e de convivência com pessoas de culturas distintas.

Na Constituição da República do Brasil de 1988, o art. 205 do capítulo III do título VII dispõe:

“A educação, direito de todos e dever do Estado e da família, será promovida incentivada com a colaboração da sociedade, visando ao pleno desenvolvimento da pessoa, seu preparo para o exercício da cidadania e sua qualificação para o trabalho”.
(BRASIL, 1988)

Dessa forma, o Estado, em teoria, possui os dispositivos legais que possibilitam a existência de programas sociais para o aporte financeiro no que tange às dificuldades de transporte intermunicipal de graduandos. Contudo, devido à escassez de recursos, esses programas não conseguem atender à demanda. A partir disso, com o objetivo de seguir com a graduação e conseguir uma ascensão social por meio do mérito, esses indivíduos se submetem à migração estudantil por conta própria muitas vezes financiados pelos pais - ou por recursos próprios - possibilitando sua manutenção na cidade universitária enquanto estudam em seu curso de graduação.

Tendo em mente essa conjuntura, são várias as formas de custear a nova moradia decorrente da situação forçada de migração. Uma maneira bastante usual é a divisão da residência com demais estudantes. Colegas de quarto (termo usado entre estudantes que dividem a mesma residência) passam a ser uma nova realidade nesse cenário. Essa

modalidade de convivência pode ser o primeiro contato desses indivíduos que vieram de lugares, culturas, costumes diferentes, o que pode ser uma experiência enriquecedora em troca de conhecimentos mas que, por outro lado, pode gerar desentendimentos. Relatos de alguns estudantes confirmam essa problemática, onde vários pontos podem acarretar divergências dentro da própria casa, tais como, o atraso de contas a pagar, o desconhecimento das regras de barulho e convivência da casa, o descumprimento de um quadro de tarefas acordado entre as pessoas, a presença de visitas inesperadas sem haver comunicado aos demais, como também a dificuldade de reunir todas as pessoas para que possam sanar os desentendimentos de convivência.

Nesse contexto, este trabalho consiste no desenvolvimento de uma aplicação mobile projetada em linguagem nativa e desenvolvida através do framework *Ionic*, visando torná-la uma ferramenta unificada de todas as demandas relativas ao gerenciamento de uma moradia compartilhada pelos seus próprios moradores (IONIC, 2023). A mesma deverá concentrar todas as informações necessárias para manter uma agenda e os compromissos da casa em dia, se tornando o principal guia de responsabilidades a serem seguidas, para assim dirimir os problemas, controlando-os antes que surjam conflitos entre os moradores.

Em consonância com a principal motivação do desenvolvimento desse projeto, o nome sugerido da aplicação é *RoomMate*. Esse termo é comumente usado em alojamentos de universidades de países anglófonos para se referir a colegas de quarto, os *roommates*.

Esse relatório está assim organizado. No Capítulo 2 apresentamos o levantamento de requisitos do software, baseado nas respostas a um questionário utilizado para auxiliar na definição desses requisitos. Em seguida, no Capítulo 3, abordamos as questões relativas ao processo de desenvolvimento, apresentando as ferramentas e diagramas úteis para entendimento das funcionalidades, bem como os protótipos de interface planejados. Já no Capítulo 4, são abordados os detalhes do desenvolvimento apresentando os resultados alcançados. Por fim, apresentamos as conclusões desse trabalho técnico.

2. LEVANTAMENTO DE REQUISITOS

Diante da necessidade do levantamento de requisitos para o desenvolvimento do software *RoomMate*, na primeira etapa foi realizada uma pesquisa para levantar os problemas de convivência mais frequentes de estudantes que possuem moradia compartilhada, por meio da aplicação de um questionário de pesquisa criado no *Google Forms*. A análise do questionário visou disponibilizar dados suficientes para o projeto de um software que auxilie estudantes que compartilham da mesma moradia a organizarem compromissos que surgem diante da convivência.

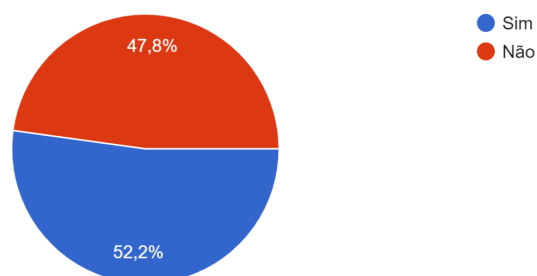
O questionário, disponível no Apêndice A deste trabalho, foi produzido baseado nas principais questões potencialmente geradoras de conflitos entre pessoas que dividem apartamento. A elaboração das perguntas se deu por meio de relatos informais de amigos do autor, assim como, a partir de relatos do próprio autor sobre a sua vivência em moradia compartilhada durante o período acadêmico.

Para obter um melhor aproveitamento do compartilhamento do formulário, o envio foi realizado via canais de conversa com estudantes, tais como, grupos de conversa via *Whatsapp*, formados por universitários e por condôminos próximos às universidades.

O questionário foi respondido por 23 pessoas, das quais 52,2% informaram já terem tido problemas devido ao não cumprimento das regras da casa, enquanto 47,8% não informaram problemas, conforme gráfico apresentado na Figura 1.

Figura 1 - Representação da quantidade de entrevistados que tiveram problemas devido ao não cumprimento das regras da casa.

Você já teve problemas devido ao não cumprimento das regras da casa?
23 respostas



Fonte: Elaborado pelo autor (2023)

Quando perguntados sobre como os moradores estabeleciam as regras da casa entre si, 78,3% dos entrevistados afirmaram que as regras da casa ficam alinhadas verbalmente, os demais informaram que ficam estabelecidas por contrato (8,7%) ou não conversam sobre as regras (8,7%). Há ainda o caso em que não há regras estabelecidas (4,3%), conforme Figura 2. O resultado obtido nesta questão demonstrou a necessidade de inclusão de uma ferramenta na aplicação a ser desenvolvida que ajudasse o *host* da residência a ter as regras da casa bem esclarecidas. Dessa forma, foi criado o #RF08 (Requisito funcional 08), que possibilita concentrar as regras da casa com a obrigatoriedade de leitura e confirmação de cada morador.

Figura 2 - Proporção das formas de acordo das regras da casa.

Como fica acordado as regras da casa entre as pessoas?
23 respostas



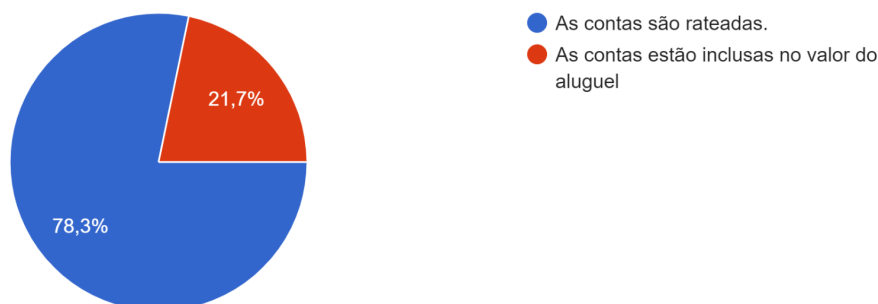
Fonte: Elaborado pelo autor (2023)

Outro quesito levantado no questionário foi sobre os débitos relacionados à residência. Na Figura 3, foi possível observar que 78,3% (18 de 23 pessoas) relataram que os débitos são rateados por igual pelos moradores.

Figura 3 - Apresentação da forma de como as contas são pagas na residência

Sobre as contas, elas são rateadas ou já estão inclusas no valor do aluguel?

23 respostas



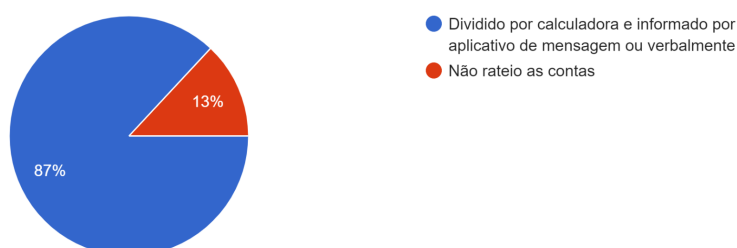
Fonte: Elaborado pelo autor (2023)

Por meio do resultado obtido na Figura 3, foi perguntada a forma que os entrevistados usam para calcular as contas quando há rateio. De acordo com a Figura 4, 87 % relataram que o rateio é feito por meio da divisão obtida por calculadora e informada por aplicativo de mensagem ou verbalmente. A partir dessa resposta, elaborou-se a função apresentada no #RF05 (Requisito Funcional 05), que tem o papel de incluir débitos, que serão rateados automaticamente para cada morador.

Figura 4 - Apresentação do método de divisão das contas rateadas.

Se há rateio nas contas, como ele é feito?

23 respostas



Fonte: Elaborado pelo autor (2023)

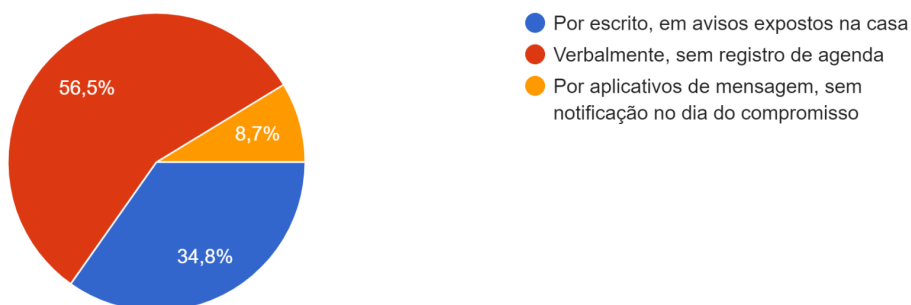
Quando perguntados sobre a agenda de compromissos da casa, de acordo com a Figura 5, 56,5 % dos entrevistados referem que não há registro dessas informações, sendo utilizada comunicação verbal. Já 34,8 % das respostas apontam que os avisos ficam expostos na casa, por escrito. 8,7 % informaram que as tarefas são passadas por aplicativos de mensagem, o que não notifica os usuários no dia do compromisso. Considerando esse resultado, foi elaborada a função disposta no #RF06 (Requisito Funcional 06) para cadastro de compromissos e o

#RF04 (Requisito Funcional 04), que será apresentado na tela inicial, na forma de um bloco de notas com o lembrete dos compromissos diários do usuário.

Figura 5 - Apresentação do método de controle dos compromissos da casa

Sobre a agenda de compromissos da casa, como é feito o controle?

23 respostas



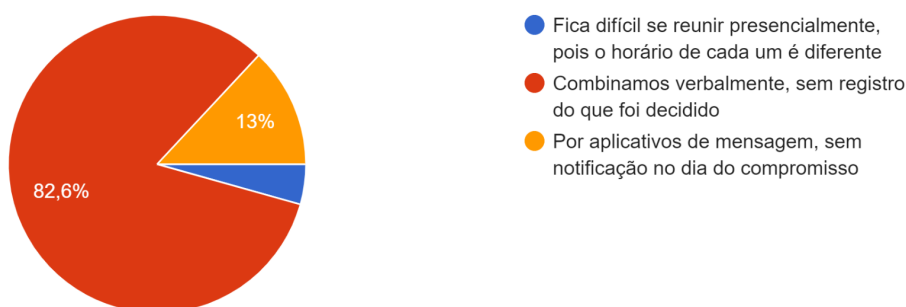
Fonte: Elaborado pelo autor (2023)

Quanto às decisões tomadas em conjunto pelos integrantes da residência, os resultados apresentados na Figura 6 apontam que 82,6 % decidem as questões de forma verbal, sem registro do que foi decidido. Nesse quesito, o #RF07 (Requisito Funcional 07) foi empregado no auxílio da criação de enquetes para registro dos resultados das decisões discutidas.

Figura 6 - Respostas sobre decisões tomadas em conjunto na residência.

Quando há decisões a serem tomadas em conjunto pelos integrantes da residência

23 respostas



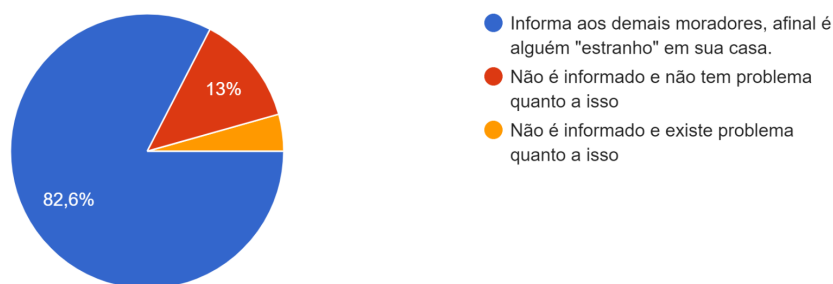
Fonte: Elaborado pelo autor (2023)

Na Figura 7, quando perguntado sobre a dinâmica de visitas da casa, 82,6 % alegaram que informam aos demais moradores sobre visitas. Uma pessoa entre vinte e três pessoas informou que não há comunicação sobre visitas que frequentam a casa e que há problemas quanto a isso. O #RF11 (Requisito Funcional 11) se trata de cadastro de visitantes, função onde os demais moradores serão comunicados a respeito de eventos de visita na residência. Essa mesma função dará também a possibilidade dos demais moradores aprovarem ou recusarem a visita agendada.

Figura 7 - Apresentação da conduta adotada para recebimento de visitas na residência

Quando alguém vai receber visita, como fica combinado?

23 respostas



Fonte: Elaborado pelo autor (2023)

Para 60,9 % dos entrevistados (Figura 8), os documentos e o regimento da residência precisam ser disponibilizados cada vez que chega um novo morador. Dessa forma, com a intenção de manter os documentos organizados e acessíveis automaticamente, foi criada a função que disponibiliza essas informações por meio do 8º Requisito Funcional.

Figura 8 - Forma de disponibilização de documentos e regimento da residência

Quanto ao regimento e demais documentos da residência. Hoje é disponibilizado em algum lugar fixo? Há a necessidade de enviar cada vez que chegar um novo morador?

23 respostas



Fonte: Elaborado pelo autor (2023)

2.1. Requisitos funcionais

Para elaborar o comportamento do software *RoomMate* foram definidos requisitos funcionais baseados no resultado do questionário cujas respostas foram apresentadas anteriormente e em conjunto com as experiências vividas pelo autor em situações de convivência com outros estudantes em um mesmo apartamento. Além disso, funções básicas de utilização de um sistema também foram incluídas nos requisitos como, por exemplo, uma criação de cadastro. A lista dos requisitos funcionais é apresentada no Quadro 1 a seguir.

Quadro 1 – Requisitos funcionais (RF)

Referência	Função	Descrição
#RF01	Efetuar login	O sistema deve efetuar o login do usuário no aplicativo
#RF02	Efetuar cadastro	O sistema deve efetuar o cadastro do usuário solicitando informações pessoais.. O cadastro não deve ser efetuado se faltar algum dado solicitado.

#RF03	Recuperar senha	O sistema deve solicitar o e-mail de cadastro para enviar o link com a opção de criar uma nova senha
#RF04	Lembretes contendo compromissos relacionados ao dia vigente	Todos os registros referentes ao dia atual devem constar numa janela de lembrete apresentada na tela inicial, tais como, débitos, visitas e enquetes.
#RF05	CRUD de contas	<ul style="list-style-type: none"> - Cadastrar uma nova conta da casa. Os valores da conta cadastrada devem ser divididos automaticamente pelos moradores registrados na casa; - Consultar as contas da casa cadastradas; - Editar contas cadastradas, desde que seja o autor do cadastro; - Excluir débitos cadastrados, desde que seja o autor do cadastro.
#RF06	CRUD de eventos	<ul style="list-style-type: none"> - Cadastrar um novo evento na agenda; - Consultar eventos agendados; - Editar eventos agendados, desde que seja o autor do evento; - Excluir eventos agendados, desde que seja o autor do evento. <p>OBS.: Usuário não deve conseguir cadastrar evento em datas passadas.</p>
#RF07	CRUD de enquetes	<ul style="list-style-type: none"> - Cadastrar uma nova enquete; - Consultar enquetes criadas; - Editar enquetes criadas (Só pode haver alteração pelo autor da enquete); - Excluir enquete criada, desde que seja o autor da enquete.

#RF08	CRUD de regras da casa	<ul style="list-style-type: none"> - Consultar regras da casa; - Funções disponíveis apenas para o host da casa: <li style="padding-left: 20px;">- Cadastrar nova regra da casa; <li style="padding-left: 20px;">- Editar regras criadas; <li style="padding-left: 20px;">- Excluir regras criadas. - As regras cadastradas obrigatoriamente precisam ser lidas e acordadas por cada morador da casa.
#RF09	Ingresso em uma casa	<p>O usuário deverá informar se ele será um host ainda na tela de cadastro:</p> <p>Caso sim, um código que irá representar a casa em que ele mora é criado automaticamente e será atribuído a ele pelo sistema.</p> <p>Caso o usuário não seja um host, deverá informar o código da casa em que ele irá ingressar.</p>
#RF10	CRUD de moradores	<p>Essa função aparece apenas para usuários host. Que poderá manter os moradores da residência em que ele é o responsável. Para novos moradores, deverá aparecer para o host, o pedido de ingresso feito por eles através do #RF10</p>
#RF11	CRUD de visitantes	<p>Informar que haverá um novo visitante em casa:</p> <ul style="list-style-type: none"> - Cadastrar um novo visitante que deverá ser aprovado pelos demais membros da casa; - Consultar visitantes cadastrados; - Editar visitante cadastrado, desde que seja o autor do cadastro; - Excluir o visitante cadastrado, desde que seja

		o autor do cadastro.
#RF12	Consultar informações dentro da conta	Trazer resultados de acordo com a informação passada numa barra de pesquisa que deverá estar disponível na tela inicial.
#RF13	Sair da conta	Efetuar o logoff da conta do aplicativo

Fonte: Elaborado pelo autor (2023)

2.2. Requisitos não-funcionais

Os requisitos não-funcionais definidos para a aplicação estão listados no Quadro 2 a seguir:

Quadro 2 – Requisitos não-funcionais (RNF)

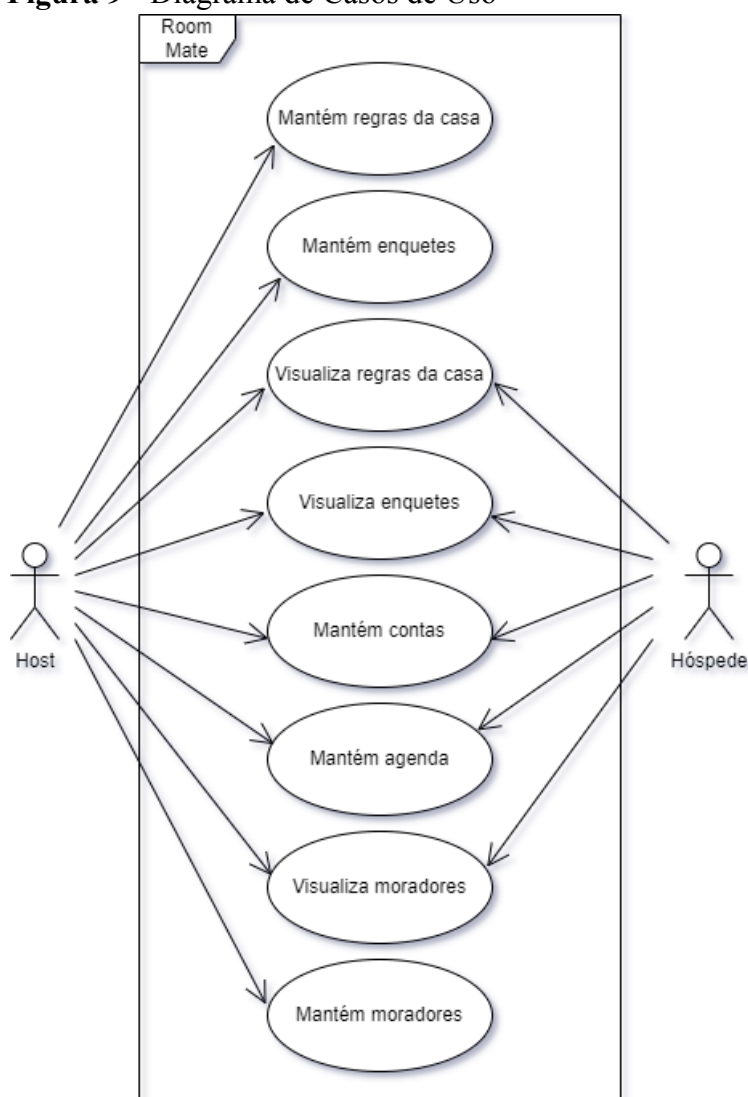
Referência	Função	Descrição
#RNF01	Segurança no acesso ao sistema	O sistema deve ser protegido contra acesso não autorizado
#RNF02	Versatilidade na execução nas principais plataformas mobile	O sistema deve ser executado nas plataformas mobile ANDROID e IOS.
#RNF03	Disponibilidade de execução Off-line	A execução do sistema deve acontecer mesmo sem acesso à rede.

Fonte: Elaborado pelo autor (2023)

3. MODELAGEM DO SISTEMA

Com alicerces nos requisitos previamente identificados, foram elaborados diversos diagramas a fim de fornecer diretrizes no processo de desenvolvimento. Esse percurso teve início com a concepção do diagrama de casos de uso, apresentado na Figura 9. Nesse diagrama, foram abordadas as autorizações conferidas aos dois perfis de usuários estipulados: "Host" e "Hóspede". Notadamente, certas funcionalidades, como a gestão das Regras da Casa, enquetes e a administração dos moradores, foram reservadas exclusivamente aos usuários *Hosts*. Por outro lado, funções como a visualização das Regras da Casa, a participação em enquetes, a exibição da lista de moradores, assim como a administração de contas e agenda, foram acessíveis tanto para os *Hosts* quanto para os *Hóspedes*.

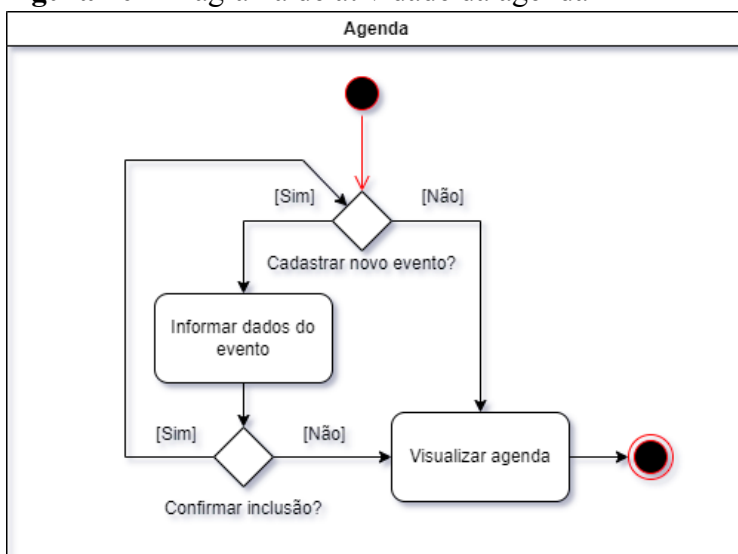
Figura 9 - Diagrama de Casos de Uso



Fonte: Elaborado pelo autor (2023)

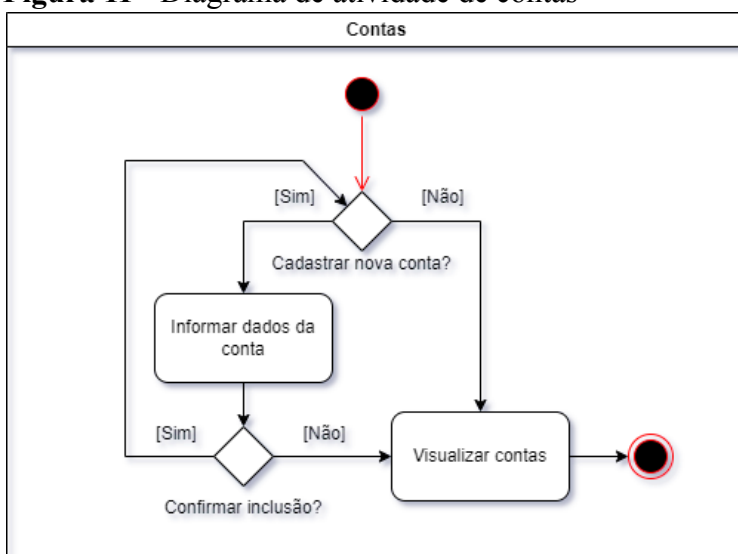
Para algumas funcionalidades, foi necessária a criação de diagramas de atividade para entender como devia ser feita a implementação do código. Na Figura 10, foi esboçada a funcionalidade da agenda do aplicativo, onde ficou possível visualizar o fluxo de acesso à agenda, onde o usuário pode escolher se deseja cadastrar um novo evento, abrindo um formulário para informar os dados desse evento ou se deseja apenas visualizar a agenda com os eventos já criados anteriormente.

Figura 10 - Diagrama de atividade da agenda



Fonte: Elaborado pelo autor (2023)

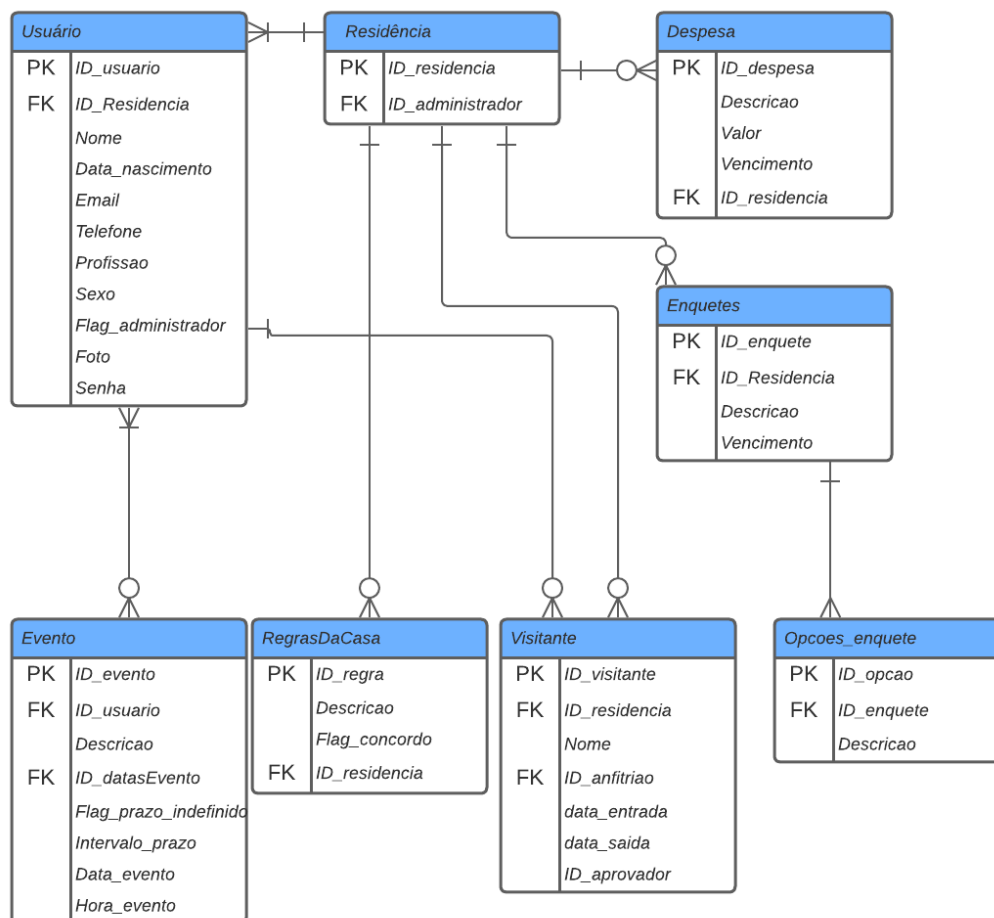
Foi desenvolvido um segundo diagrama de atividade com o objetivo de proporcionar uma compreensão mais abrangente sobre o fluxo de funcionalidades relacionadas às contas no aplicativo. Na Figura 11, esse diagrama apresenta de forma detalhada como o processo de acesso às contas foi estruturado. Ele oferece ao usuário a capacidade de tomar decisões entre cadastrar uma nova conta ou realizar a visualização das contas previamente registradas. Esse diagrama esclarece a lógica subjacente ao gerenciamento das contas no contexto da aplicação.

Figura 11 - Diagrama de atividade de contas

Fonte: Elaborado pelo autor (2023)

Com o intuito de aprofundar a compreensão das interações entre as entidades concebidas neste projeto, foi elaborado um diagrama de Entidade-Relacionamento, ilustrado na Figura 12. A finalidade primordial foi otimizar a arquitetura do banco de dados, essencial para o correto funcionamento do aplicativo. Por meio deste diagrama, as relações entre as diversas entidades foram elucidadas, destacando, por exemplo, a interligação entre a entidade "usuário" e a entidade "residência", bem como a relação entre "enquetes" e "residência". Essa visualização proporcionou clareza na representação das conexões fundamentais para o funcionamento coerente do sistema.

Figura 12 - Diagrama de Entidade-Relacionamento do aplicativo.



Fonte: Elaborado pelo autor (2023)

4. DESENVOLVIMENTO DO *FRONT-END*

Com base nos requisitos identificados, realizou-se uma análise de ferramentas visando a facilitar o desenvolvimento do projeto. Essas ferramentas, conhecidas como *frameworks*, buscam permitir que os desenvolvedores concentrem seus esforços na funcionalidade da aplicação, reduzindo o tempo gasto em configurações e arquitetura.

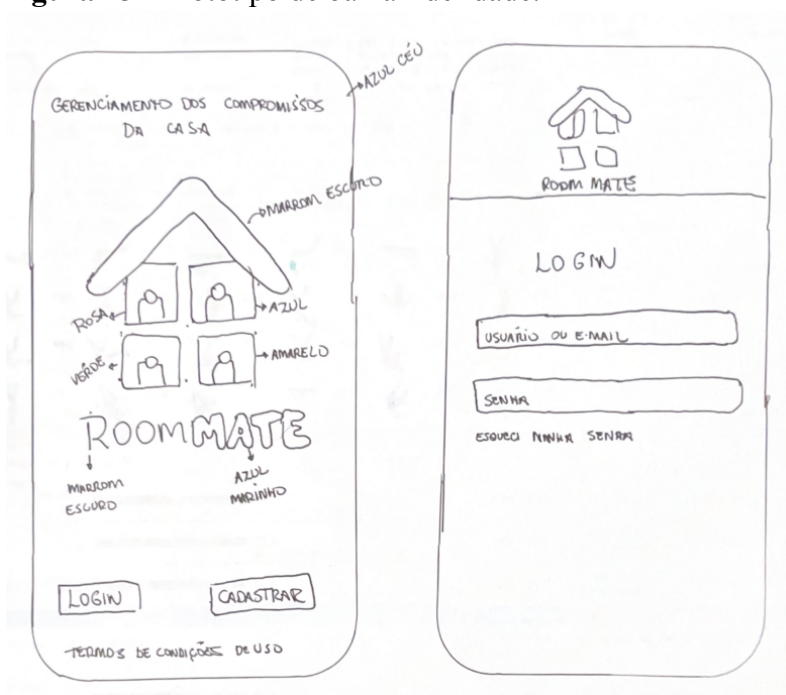
Para a concepção da aplicação, a proposta almejou um design limpo, simples e funcional, com componentes padronizados e uma tipografia coesa. Nesse contexto, o *framework* mais indicado foi o *Ionic*, adequado ao desenvolvimento de aplicações híbridas que atendem aos requisitos deste projeto (IONIC, 2023).

Visto que este projeto é concebido para futuras versões como uma aplicação híbrida, acessível tanto em dispositivos móveis quanto em *desktops*, buscou-se um *framework* que permitisse a construção da aplicação móvel como alternativa às abordagens nativas. Isso viabilizou o desenvolvimento da aplicação por meio de linguagens multiplataforma. Ademais, o *Ionic* oferece a possibilidade de distribuir a aplicação móvel para sistemas *Android* e *iOS*, tudo a partir do mesmo código desenvolvido, cumprindo o RNF02 - versatilidade na execução nas principais plataformas mobile. (IONIC, 2023)

4.1. Protótipo de baixa fidelidade da interface

Neste projeto, a concepção do design inicial da interface gráfica da aplicação envolveu a criação de um protótipo de baixa fidelidade. Esse protótipo foi esboçado manualmente com o uso de caneta e papel, dando forma à logomarca e à tela de apresentação do sistema, conforme ilustrado na Figura 13.

Figura 13 - Protótipo de baixa fidelidade.



Fonte: Elaborado pelo autor (2023)

4.2. Protótipo de alta fidelidade da interface

Após a conclusão desta etapa, a vetorização e criação da logomarca em alta fidelidade foram efetuadas por meio do software de edição gráfica *Corel DRAW*, seguindo as diretrizes do uso do software citadas por Mesick (2022). Com a intenção de assegurar uma estética visual equilibrada, optou-se por uma paleta de cores mais sóbria, tanto na composição da logomarca quanto no desenvolvimento global da aplicação. Isso representou uma modificação em relação à ideia de cores originalmente esboçadas na Figura 13, visando evitar uma sobrecarga de informações visuais. O resultado da logomarca finalizada é apresentado na Figura 14.

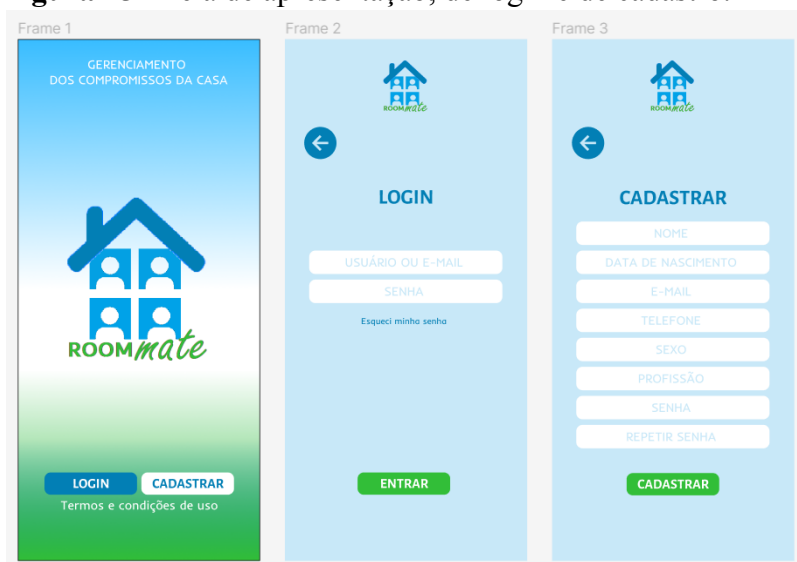
Figura 14 - Logomarca da aplicação.



Fonte: Elaborado pelo autor (2023)

Após a seleção da logomarca definitiva, foi empregado o software de edição gráfica vetorial *Figma* para materializar a concepção de telas e funcionalidades em alta fidelidade. O *Figma* possibilitou a criação dos protótipos das diversas telas do sistema, resultando em uma representação visual consistente e detalhada. A Figura 15 traz à tona os protótipos das telas de apresentação, login e cadastro, encapsulando assim a visualização das etapas de desenvolvimento, essa fase trouxe uma ideia de como seria elaborado o desenvolvimento do aplicativo, o que não significa que é a versão final do desenvolvimento.

Figura 15 - Tela de apresentação, de login e de cadastro.



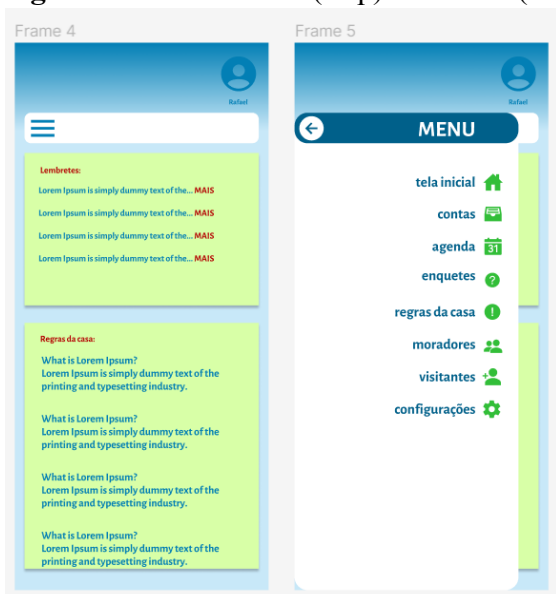
Fonte: Elaborado pelo autor (2023)

Dentro do ambiente do *Figma*, todas as demais telas que compõem o sistema também foram meticulosamente concebidas. A Figura 16, à esquerda, oferece uma visão da tela inicial que emerge imediatamente após a conclusão do processo de login. O propósito dessa tela é centralizar as informações primordiais, abrangendo tanto os lembretes do usuário

quanto às regras pertinentes à residência. Para apresentar essas informações de maneira eficaz, foram criados elementos que simulam *post-its*, proporcionando uma associação visual aos quadros de anotações frequentemente utilizados em ambientes compartilhados. Também foi idealizada uma barra de pesquisa que retorne informações disponíveis dentro da conta do usuário, atendendo ao #RF12 - Consultar informações dentro da conta..

Adicionalmente, na mesma Figura 16, à direita, é possível identificar o Menu Principal, que serve como portal para acessar as telas cruciais da aplicação. Cada alternativa do menu é representada por um ícone ilustrativo, simplificando a navegação e facilitando o reconhecimento das funcionalidades disponíveis. Esse design intuitivo contribui para uma experiência de usuário mais acessível e amigável.

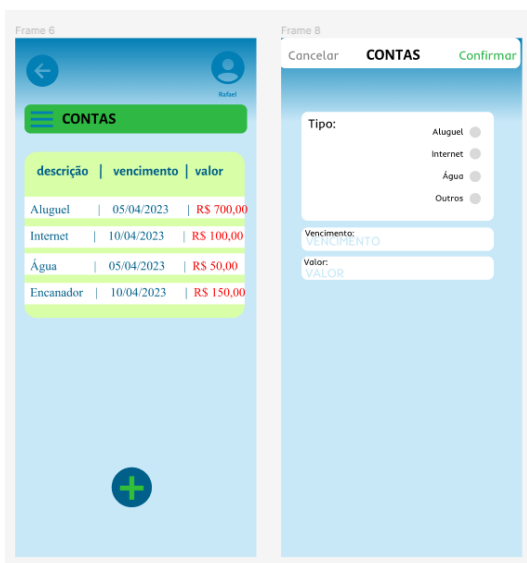
Figura 16 - Tela inicial (esq.) e o menu (dir.)



Fonte: Elaborado pelo autor (2023)

Na Figura 17, a tela de visualização de contas é apresentada à esquerda, exibindo as contas registradas para o usuário. Na tela à direita, observa-se a representação da tela destinada à inclusão de novas contas. Nesse contexto, o usuário é guiado pelo processo de seleção do tipo de conta, estipulação da data de vencimento e atribuição do valor correspondente, caracterizando assim um fluxo intuitivo e eficiente.

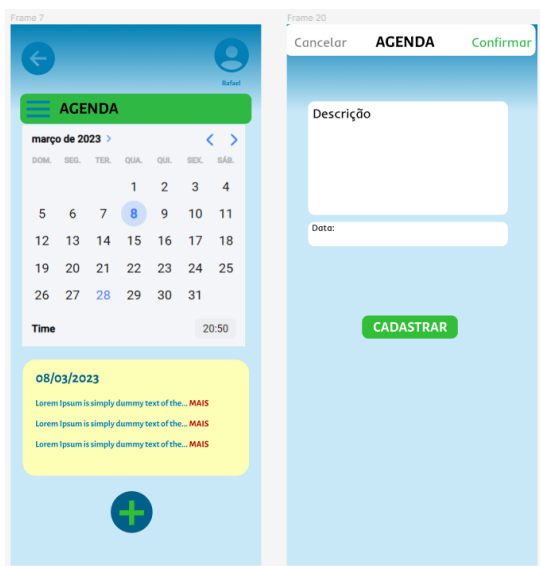
Figura 17 - Tela de visualização (esq.) e inclusão (dir.) de contas



Fonte: Elaborado pelo autor (2023)

Na Figura 18, à esquerda, um calendário que está sempre visível nessa tela é disponibilizado, permitindo a seleção da data desejada para visualizar os eventos correspondentes na agenda. À direita, é apresentada a tela destinada à inclusão de um novo evento, proporcionando uma abordagem interativa para o registro de novos compromissos.

Figura 18 - Tela de visualização de agenda (esq.) e inclusão de novo evento (dir.)

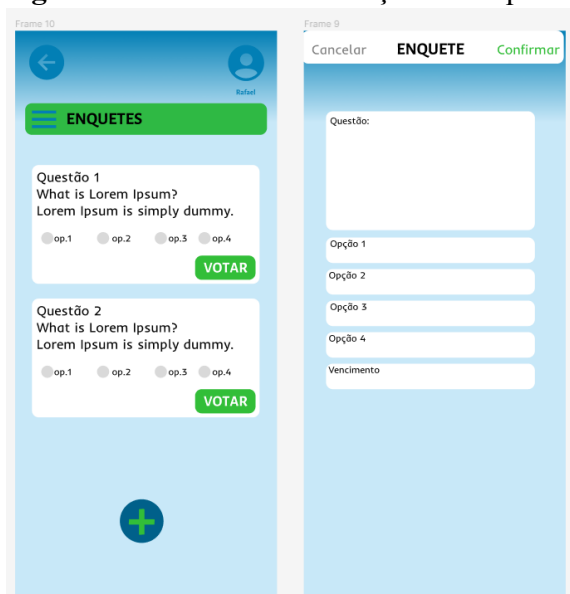


Fonte: Elaborado pelo autor (2023)

A representação do menu de enquetes, que faz parte integrante do aplicativo, é apresentada na Figura 19. À esquerda, há uma visão panorâmica das enquetes criadas,

juntamente com a capacidade de participar ao votar em suas opções. Na Figura 19 à direita, surge a tela que exhibe o processo de criação de uma nova enquete. Nesse contexto, os usuários podem adicionar a pergunta e suas respectivas alternativas, atualmente limitadas em quatro opções, além de determinar uma data de encerramento para a votação. Essa abordagem abrange de maneira coesa as funcionalidades relacionadas às enquetes dentro do aplicativo.

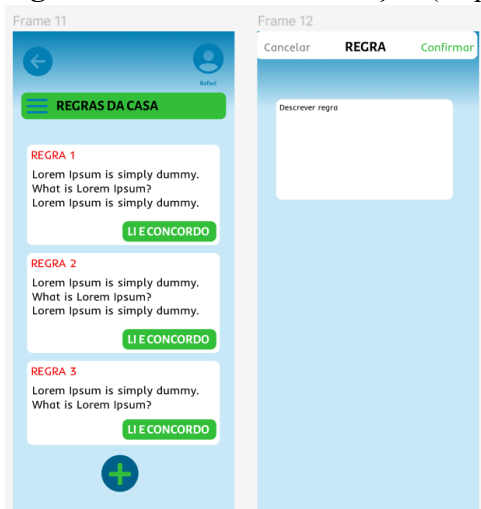
Figura 19 - Tela de visualização de enquetes (esq.) e inclusão de nova enquete (dir.).



Fonte: Elaborado pelo autor (2023)

Na Figura 20, observam-se com duas telas distintas: na seção "Regras da casa" à esquerda, é exibida uma tela que apresenta as diretrizes da residência. Aqui, o usuário é instado a ler atentamente e consentir com todas as normas ao entrar em uma habitação já estabelecida. Se o usuário ocupa a posição de administrador da casa, a tela mostrada na Figura 20 à direita, se torna acessível, permitindo-lhe adicionar as regras residenciais conforme necessário.

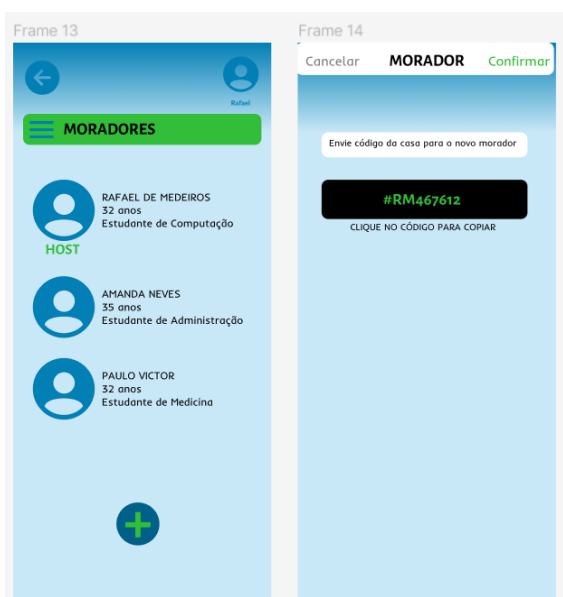
Figura 20 - Tela de visualização (esq.) e inclusão (dir.) de regras da casa



Fonte: Elaborado pelo autor (2023)

A Figura 21 oferece uma representação visual das telas associadas aos moradores. Na seção "Moradores", à esquerda, são apresentados os perfis de todos os residentes que compartilham a mesma residência do usuário. Em contrapartida, a tela à direita na Figura 21, proporciona uma funcionalidade que permite aos usuários já conectados à casa a partilha de um código, viabilizando a integração de novos moradores ao convívio residencial. Através dessa representação, o fluxo de interações no contexto dos moradores é elucidado de maneira concisa e intuitiva.

Figura 21 - Tela de visualização (esq.) e apresentação (dir.) do código da casa.

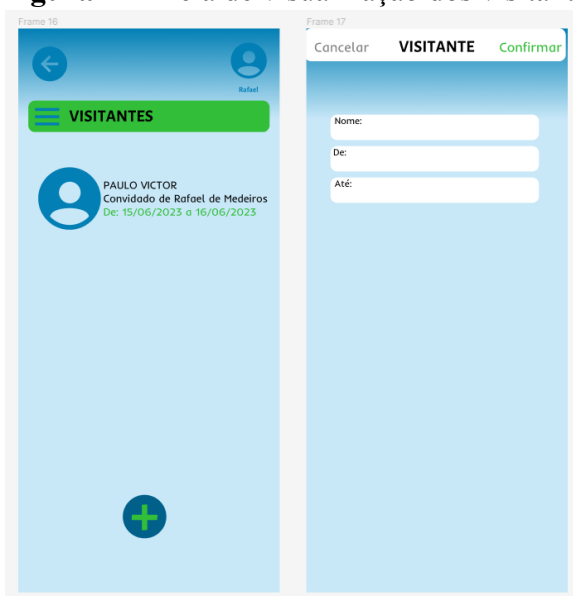


Fonte: Elaborado pelo autor (2023)

No âmbito da gestão dos visitantes da residência, uma tela foi concebida para fornecer a visualização dos visitantes previamente cadastrados, com a opção de inclusão de novos visitantes, caso necessário. A Figura 22 apresenta as telas relacionadas aos visitantes, proporcionando uma visão das funcionalidades pertinentes a essa categoria de interação.

Nesse contexto, a aplicação oferece uma interface que agiliza o controle dos visitantes da residência, permitindo tanto a revisão dos visitantes existentes quanto a adição de novas entradas, conforme necessário. Por meio dessa representação, a capacidade de gerenciamento dos visitantes é delineada de maneira clara e eficaz.

Figura 22 - Tela de visualização dos visitantes (esq.) e de inclusão de novo visitante (dir.)

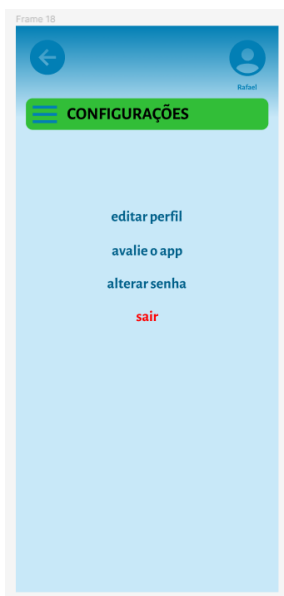


Fonte: Elaborado pelo autor (2023)

Concluindo, a última tela concebida está relacionada às configurações do aplicativo. Nessa seção, os usuários têm a capacidade de editar seu perfil, avaliar o aplicativo, modificar a senha e realizar o logout da conta atualmente logada. Essa tela é mostrada na Figura 23.

Por meio dessa interface, é oferecida uma gama de opções para a personalização e administração da experiência do usuário. Essas funcionalidades englobam desde a manutenção do perfil até a interação com a aplicação em termos de avaliação e gerenciamento de segurança. A representação visual na Figura 23 encapsula esse conjunto de possibilidades, oferecendo aos usuários o controle total sobre as configurações do aplicativo.

Figura 23 - Tela de configurações do aplicativo.



Fonte: Elaborado pelo autor (2023)

4.3. Programação do *front-end*

Após a conclusão da fase de prototipagem do aplicativo, avançamos para a programação do *front-end*, adotando a linguagem *TypeScript* em conjunto com o *framework Ionic*. Esta escolha não apenas encaixa-se perfeitamente na visão de futura hibridização do aplicativo, mas também proporciona uma série de benefícios. O *Ionic* oferece uma extensa gama de componentes predefinidos que enriquecem a interface, tornando-a mais intuitiva e sofisticada, ao mesmo tempo em que simplifica o processo de desenvolvimento. (TYPESCRIPT, 2023; IONIC, 2023)

Na estruturação do *back-end*, voltamos nossa atenção ao *Node.js*, uma plataforma de desenvolvimento de código aberto. A seleção do *Node.js* foi guiada pela sua eficiência e escalabilidade, permitindo lidar com várias operações concorrentes sem interromper a fluidez do programa. Aproveitando o mecanismo de gerenciamento de pacotes, *NPM*, otimizou-se o desenvolvimento, promovendo a reutilização do código (NODE, 2023)

Para agilizar a criação e manipulação das tabelas do banco de dados, recorreremos ao *Prisma ORM (Object Relational Mapper)*, uma técnica que estabelece uma relação entre objetos e os dados que eles representam. Essa abordagem proporcionou um desenvolvimento

mais fluido e alinhado com as entidades do código *TypeScript* (PRISMA, 2023; TYPESCRIPT, 2023)

Na criação da *API REST*, adotamos o *framework Fastify*, uma ferramenta contemporânea e ágil para o desenvolvimento de aplicações web em Node.js (FASTIFY, 2023; NODE, 2023). Dando continuidade à nossa infraestrutura, optamos pelo sólido sistema de gerenciamento de banco de dados, o *PostgreSQL* (POSTGRESQL, 2023). Sua robustez se revelou particularmente útil para lidar com consultas complexas e volumes substanciais de dados. Para simplificar o processo de instalação e minimizar possíveis desafios, implementamos a plataforma *Docker* (DOCKER, 2023). Isso nos permitiu empacotar a base de dados necessária no aplicativo em um contêiner, acelerando consideravelmente o ciclo de desenvolvimento e implantação.

Para a autenticação e autorização dos usuários da aplicação, empregamos a tecnologia *JSON Web Token (JWT)*. O *JWT* desempenha um papel crucial ao estabelecer um formato compacto e seguro para a troca de informações entre o cliente e o servidor, garantindo assim a integridade e a confidencialidade dos dados transmitidos (JWT.IO, 2023).

5. DESENVOLVIMENTO DO *BACK-END*

Neste Capítulo, abordamos os aspectos cruciais do planejamento que demandaram um cuidado especial durante o desenvolvimento. Essa abordagem foi necessária para garantir o atendimento aos requisitos identificados e para suprir a necessidade inicial do público-alvo deste projeto. O objetivo foi assegurar que a solução proposta cumprisse a abordagem sugerida, visando resolver de forma eficaz o problema que foi identificado. É bastante comum realizar modificações de layout durante o desenvolvimento de um aplicativo para aprimorar a experiência do usuário, ajustar a aparência ou lidar com limitações tecnológicas. Essas modificações podem ser necessárias para garantir que a aplicação seja esteticamente agradável e funcional. No entanto, é importante ressaltar que as modificações realizadas durante o desenvolvimento não comprometeram a qualidade do resultado final.

Visando atender ao primeiro Requisito Não Funcional apresentado no Quadro 2, ao implementar o login no sistema, optou-se pela tecnologia *JSON Web Token (JWT)* para fazer a comunicação entre cliente e servidor, garantindo a confidencialidade e segurança no acesso ao sistema (JWT.OI, 2023).

Para a tela de efetuar o cadastro no sistema, foi incluído uma tratativa que possibilitou o usuário informar se ele é host ou hóspede, incluindo na base de dados essa informação que foi necessária para permissões dentro do sistema que dependem do tipo de usuário.

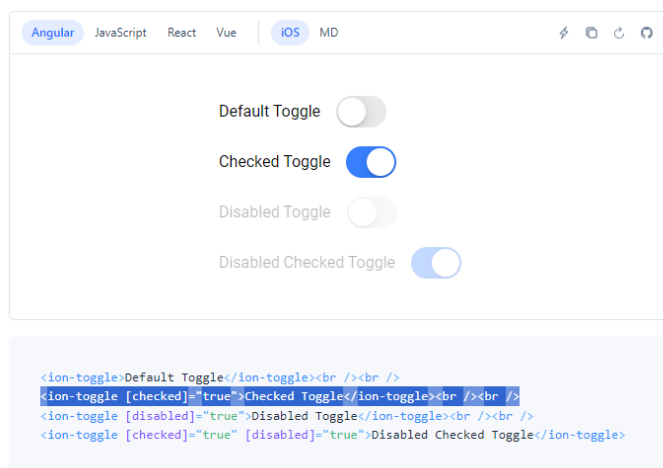
Por meio do framework Ionic, foi integrado o elemento "*ion-toggle*", como ilustrado na figura 24, com o propósito de aprimorar a interface de forma intuitiva. Este componente permite que o usuário alterne entre dois estados, os quais, neste caso, são representados pelas opções "host" ou "hóspede".

Figura 24 - Componente ion-toggle do *framework Ionic*.

ion-toggle

Toggles are switches that change the state of a single option. They can be switched on or off by pressing or swiping them. Toggles can also be checked programmatically by setting the `checked` property.

Basic Usage



Fonte: Adaptado de ionicframework.com. URL: <https://ionicframework.com/docs/api/toggle>

No código da tela de cadastro, foi incluída uma condição para que, caso o usuário não seja um host, apresente um formulário para que ele informe o código da casa que ele deseja ingressar. Dessa forma, foi incluída uma condição na linha 27, conforme Figura 25, que se comunica com o arquivo *typescript* para que houvesse a manipulação do componente de entrada de dados.

Figura 25 - Trecho de código da classe de cadastro.

```

19 <form [formGroup]="form" >
20   <ion-row class="form">
21     <ion-item lines="none">
22       <ion-label>Host?</ion-label>
23       <ion-toggle formControlName="isAdminControl">Host</ion-toggle>
24     </ion-item>
25   </ion-row>
26
27   <ion-row *ngIf="!isAdmin" class="form" >
28     <ion-item lines="none" color="success">
29       <ion-label position="floating">Código da casa</ion-label>
30       <ion-input type="text"></ion-input>
31     </ion-item>
32   </ion-row>

```

Fonte: Elaborado pelo autor (2023)

Assim, ao usuário ativar o recurso do componente *ion-toggle* para indicar sua função como *host*, conforme ilustrado na Figura 26, à esquerda, ele se isenta da responsabilidade de inserir o código da residência, uma vez que será automaticamente associado a um código recém-gerado pelo sistema. Isso assegura que suas atividades fiquem vinculadas à condição de *host*, além de permitir o compartilhamento do código da residência com seus hóspedes. No cenário em que o usuário opta por não ativar o componente *ion-toggle* (Figura 26 à direita), sinalizando que não atuará como *host*, uma nova seção é destacada, requerendo a inserção do código da residência à qual deseja se conectar. Essa tela passou por alterações quando comparada a seu protótipo apresentado na figura 15, para melhorar a funcionalidade para o usuário.

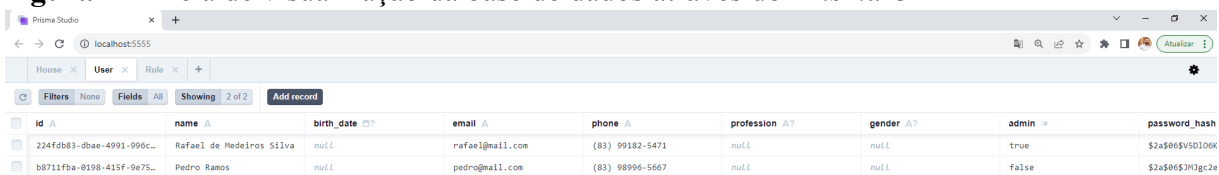
Figura 26 - Tela de cadastro do host (esq.) e Tela de cadastro do hóspede (dir.)

Tela	Função	Administrador?	Código da casa	Nome Completo	E-mail	Telefone	Sexo	Senha	Repetir senha
Esquerda	Host	Ativado	Não aplicável	Rafael de Medeiros Silva	rafael@mail.com	(83) 99182-5471	M
Direita	Hóspede	Desativado	3349a02e-0e74-43e4-9a50-164214a8572c	Pedro Ramos	pedro@mail.com	(83) 98996-5667	M

Fonte: Elaborado pelo autor (2023)

Após a conclusão do processo de cadastro, tornou-se evidente a eficácia do *Prisma ORM (Object Relational Mapper)* na gestão das tabelas do banco de dados, proporcionando uma visualização instantânea das alterações ocorridas, mostradas na Figura 27 (PRISMA, 2023). Essa agilidade destacou-se na inserção de usuários, regras e demais informações que compõem o aplicativo, que foram imediatamente registrados no banco de dados, alinhando-se de forma precisa com as informações fornecidas durante o cadastro.

Figura 27 - Tela de visualização da base de dados através do *Prisma ORM*



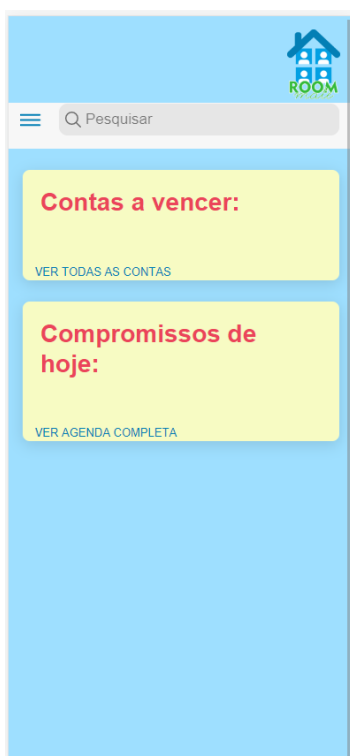
The screenshot shows the Prisma Studio interface with a table of user records. The table has columns for id, name, birth_date, email, phone, profession, gender, admin, and password_hash. Two records are visible:

id	name	birth_date	email	phone	profession	gender	admin	password_hash
224fdb83-dbae-4991-996c...	Rafael de Medeiros Silva	null	rafael@mail.com	(83) 99182-5471	null	null	true	\$2a\$86\$V5D106K
b8711fba-8198-415f-9e75...	Pedro Ramos	null	pedro@mail.com	(83) 98996-5667	null	null	false	\$2a\$86\$3HJgc2e

Fonte: Elaborado pelo autor (2023)

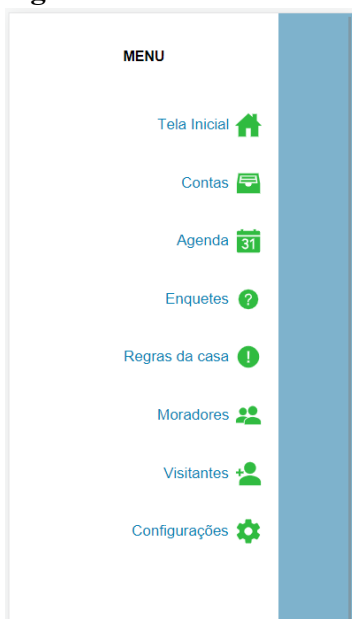
Feito o cadastro do usuário, o seu primeiro acesso retorna uma tela inicial sem contas ou compromissos a serem listados, como apresentado na figura 28.

Figura 28 - Tela inicial do aplicativo sem registros



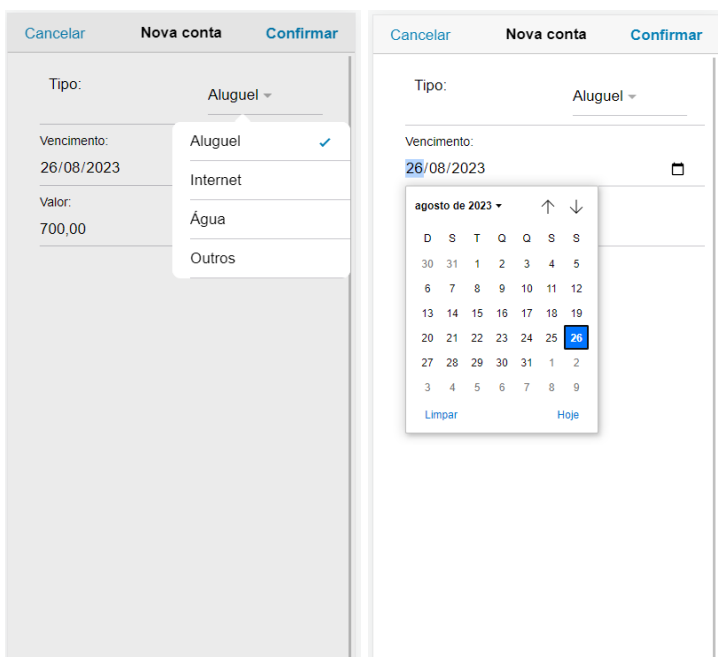
Fonte: Elaborado pelo autor (2023)

Ao clicar no ícone que representa o menu, na lateral da barra de pesquisa, são disponibilizadas as opções de acesso com ícones representativos para facilitar a identificação através de um *pop-up* lateralizado, representado na figura 29. A primeira opção desse menu dá acesso à tela de “Contas”.

Figura 29 - Menu lateralizado da aplicação

Fonte: Elaborado pelo autor (2023)

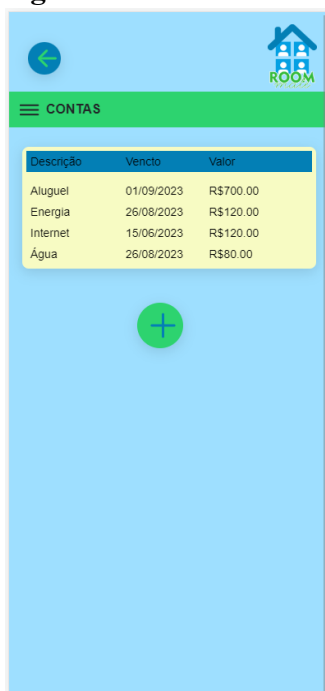
Quando o usuário clica na opção “Contas”, tem acesso à tela que permite visualizar as contas cadastradas e também cadastrar uma conta nova, através do ícone de inclusão, acessando a tela apresentada na figura 30 para preencher os dados da nova conta a ser cadastrada.

Figura 30 - Tela de inclusão de conta com seleção de tipo (esq.) e seleção de data (dir.)

Fonte: Elaborado pelo autor (2023)

Ao retornar à tela principal de “Contas” é possível consultar as contas cadastradas, onde aparecem os dados informados: Descrição, Vencimento e Valor. Na figura 31 é apresentado um conjunto de registros feitos pelo usuário através da tela da figura 30.

Figura 31 - Tela de visualização de contas



Descrição	Vencido	Valor
Aluguel	01/09/2023	R\$700.00
Energia	26/08/2023	R\$120.00
Internet	15/06/2023	R\$120.00
Água	26/08/2023	R\$80.00

Fonte: Elaborado pelo autor (2023)

A segunda tela apresentada no menu, refere-se a agenda de compromissos, onde é possível adicionar um novo compromisso através da tela de inclusão. Na figura 32, representando uma inclusão de compromisso, trata-se de um evento de apresentação de TCC onde a data selecionada no calendário para o acontecimento é 31 de agosto de 2023.

Figura 32 - Tela de inclusão de novo compromisso

Cancelado Novo compromisso Confirmado

Evento:
Apresentação do TCC

Período:

agosto de 2023 < >

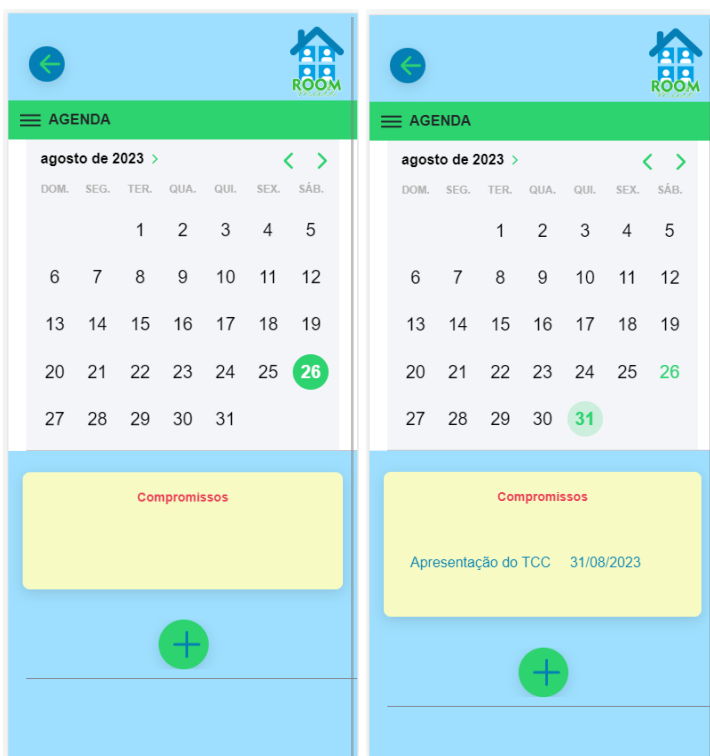
DOM.	SEG.	TER.	QUA.	QUI.	SEX.	SÁB.
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

Fonte: Elaborado pelo autor (2023)

Após retornar à tela de visualização da agenda, você poderá notar que o compromisso de apresentação do TCC, que foi previamente agendado, não está visível. Isso ocorre porque a data selecionada no calendário é o dia 26 de agosto, a qual difere da data em que o compromisso foi originalmente agendado, conforme exemplificado na figura 33, à esquerda.

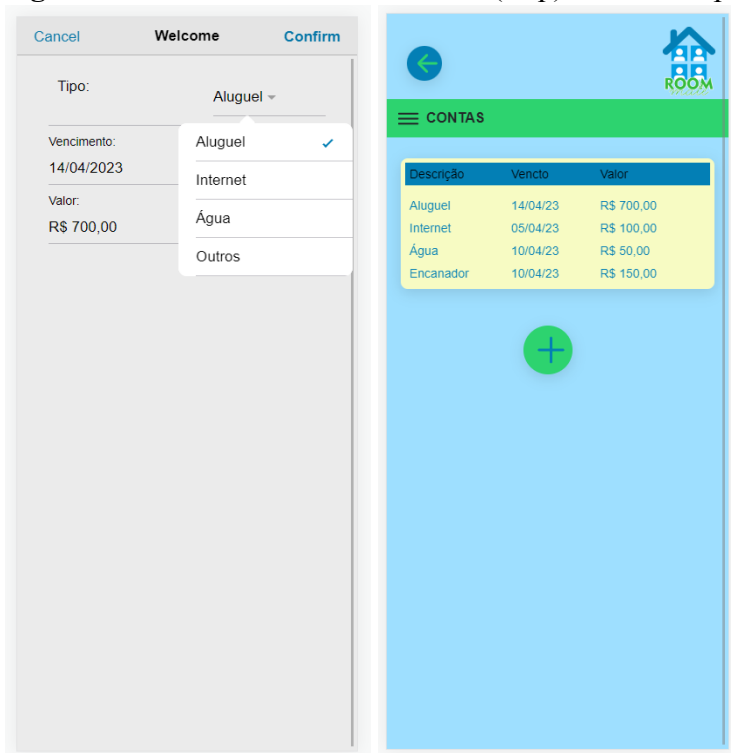
Para visualizar esse evento específico, é crucial que você selecione a data em que o compromisso foi agendado, ou seja, o dia 31 de agosto, conforme ilustrado na figura 33, à direita. Essa funcionalidade busca emular a experiência de utilizar uma agenda convencional, onde os compromissos são consultados com base nas datas em que foram marcados.

O objetivo é oferecer uma abordagem intuitiva semelhante à de um calendário tradicional, permitindo que você navegue pelos compromissos de forma eficaz, selecionando as datas correspondentes para acessar os eventos associados. Dessa maneira, a funcionalidade de visualização de compromissos é harmonizada com o contexto de um calendário, proporcionando uma experiência prática e familiar aos usuários.

Figura 33 - Tela da agenda do dia 26/08 (esq.) e tela da agenda do dia 31/08 (dir.)

Fonte: Elaborado pelo autor (2023)

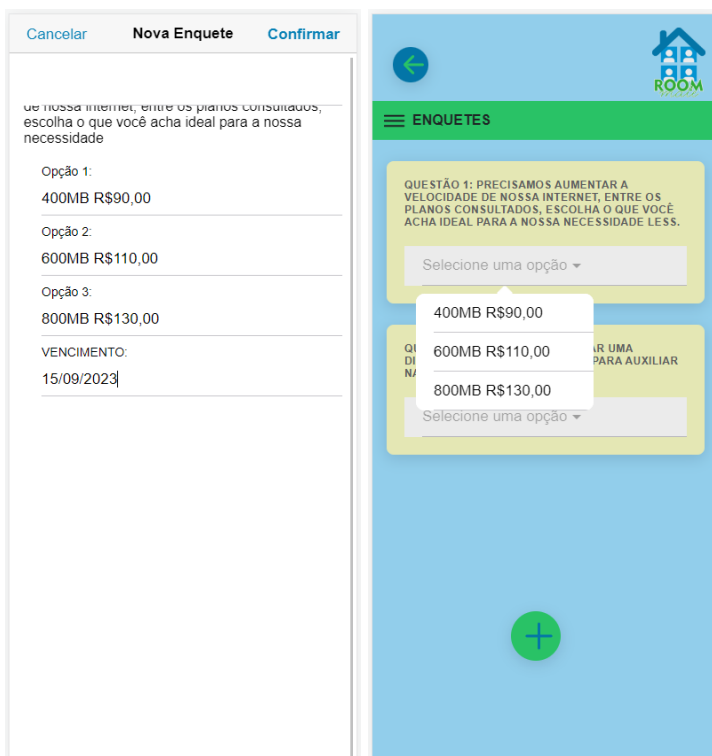
Dentre as operações de *CRUD* (*Create, Read, Update, Delete*) implementadas no aplicativo, destaca-se o *CRUD* relacionado às contas. Nesse contexto, a tela de cadastro de contas assume um papel central, demandando a inserção de informações como o tipo de conta, a data de vencimento e o valor correspondente (Figura 34 à esquerda). Por sua vez, a Figura 34, à direita, ilustra a interface principal do menu de contas, onde todas as contas registradas na residência são apresentadas de maneira organizada, oferecendo uma visão panorâmica das informações contábeis. No contexto do 5º Requisito Funcional, foi pontuado que a edição e exclusão de uma conta estavam restritas ao autor da mesma. Entretanto, essa particularidade foi reservada para ser abordada em uma futura etapa de implementação.

Figura 34 - Tela de cadastro de conta (esq.) e interface principal do menu de contas (dir.)

Fonte: Elaborado pelo autor (2023)

No contexto das enquetes, o aplicativo proporciona uma plataforma de votação entre os membros de uma mesma residência. Esta funcionalidade permite que apenas usuários com o perfil de *host* possam criar uma nova enquete, como ilustrado na figura 29, à esquerda. No entanto, a participação na votação não é restrita a *hosts*; qualquer usuário que seja parte da mesma residência tem a prerrogativa de participar, expressando suas opiniões por meio das enquetes disponíveis. A tela que reúne as enquetes para votação está representada na figura 29, à direita.

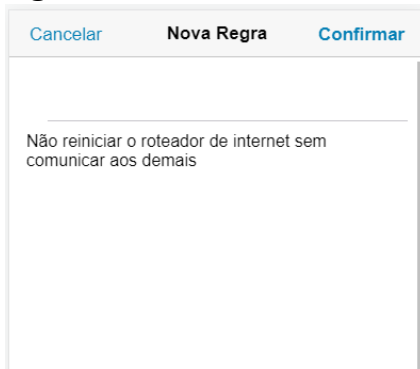
Figura 35 - Tela de cadastro de enquete (esq.) e interface principal de enquetes (dir.)



Fonte: Elaborado pelo autor (2023)

Outro marco fundamental na aplicação, em consonância com o planejamento, é a centralização das regras da casa. Essa capacidade de adicionar novas regras é exclusivamente designada ao *host* da casa, enquanto as regras em si podem ser acessadas por todos os residentes. Como ilustrado na figura 36, uma regra recém-criada foi inserida, destacando a simplicidade do processo, que requer apenas a descrição da nova diretriz.

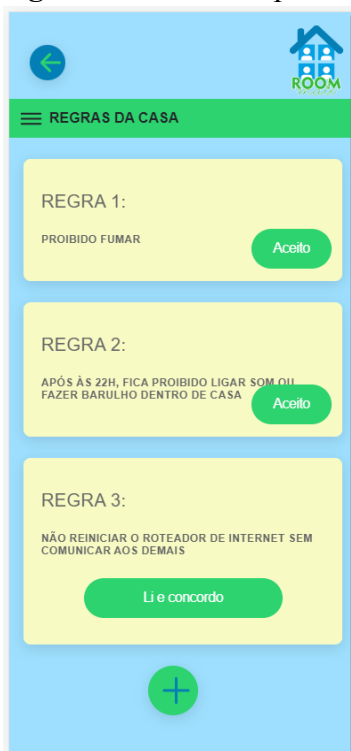
Figura 36 - Tela de inclusão de nova regra



Fonte: Elaborado pelo autor (2023)

Após a inclusão de uma nova regra, esta passa a integrar o painel de exibição na tela principal da seção "Regras da Casa", como exemplificado na figura 37. Na visualização compartilhada por todos os usuários, as regras são apresentadas, requerendo a aceitação por parte dos residentes. Na figura 37 é ilustrado o painel de regras para um usuário, onde as regras 1 e 2 já foram aceitas, enquanto a regra 3, recentemente adicionada, ainda se encontra pendente de aceitação.

Figura 37 - Tela de apresentação de regras da casa

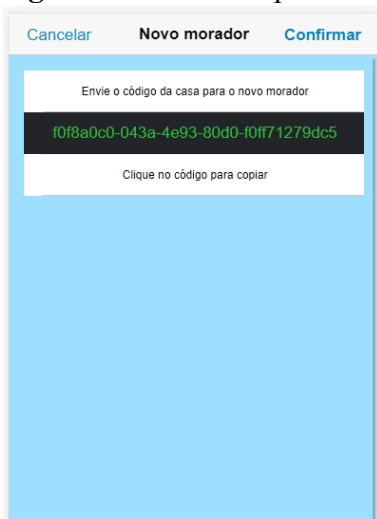


Fonte: Elaborado pelo autor (2023)

Ao abordar a inclusão de novos moradores na residência, a interação é facilitada pelo compartilhamento do código da residência, uma etapa importante para viabilizar a inscrição dos novos membros no aplicativo e a sua integração à comunidade residencial. Como exemplo apresentado anteriormente, na figura 26, à direita, observamos o usuário Pedro Ramos, que informou não possuir um perfil do tipo *host*. Como consequência, ele é solicitado a inserir o código específico da residência à qual deseja se vincular. Nesse cenário, a colaboração entre os usuários é evidenciada, sendo que o usuário *host* Rafael precisou fornecer o código da residência a Pedro Ramos, permitindo assim que ele se tornasse um hóspede na residência.

Esse processo é exemplificado na figura 38, onde o usuário *host* acessa o menu de moradores e, de forma conveniente, obtém acesso ao código da residência que foi automaticamente gerado após o cadastro inicial. É justamente esse código que o usuário *host* deve compartilhar com o novo morador. Ao fornecer o código, o novo morador pode inseri-lo durante o processo de registro, o que o conecta instantaneamente à residência.

Figura 38 - Tela de apresentação de código da residência para o usuário *host*.



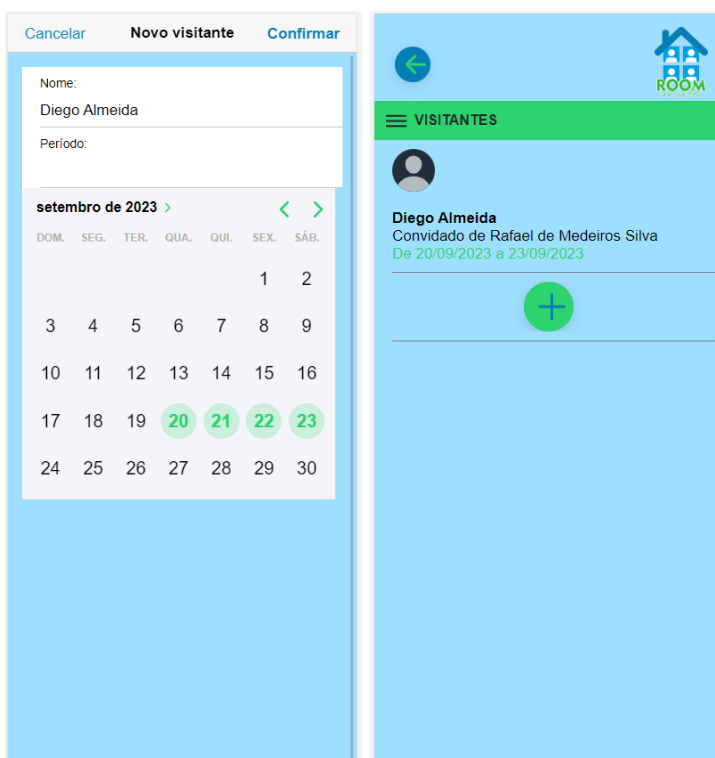
Fonte: Elaborado pelo autor (2023)

É importante destacar uma distinção: enquanto os usuários com perfil de *host* podem acessar e compartilhar o código da residência, o mesmo privilégio não é estendido aos usuários do tipo hóspede.

Essa abordagem reforça a dinâmica da plataforma, enfatizando a importância da privacidade e da segurança das informações. Os usuários do tipo *host* têm acesso à ferramenta de compartilhamento do código, permitindo-lhes facilitar a entrada de novos moradores na comunidade residencial.

Essa estrutura reafirma o compromisso do aplicativo em proporcionar um ambiente seguro e bem coordenado para todos os usuários, mantendo uma harmonia equilibrada entre a conveniência e a privacidade dentro da comunidade residencial.

Sobre o menu de visitantes, como função comum para todos os usuários, a tela de inclusão de um novo visitante engloba o nome do visitante e o período em que ele ficará na casa, no exemplo da figura 33, à esquerda, o visitante Diego Almeida ficará de visita no período de 20/09/2023 até 23/09/2023. Ao retornar à tela principal de visitantes é possível observar na figura 39, à direita, que o visitante é apresentado com o nome do usuário que o convidou.

Figura 39 - Tela de cadastro de visitante (esq.) e interface de visitantes cadastrados (dir.)

Fonte: Elaborado pelo autor (2023)

Concluindo a apresentação dos resultados, ao voltar para a tela inicial, é evidente que, em virtude dos cadastros efetuados durante a apresentação, surge um panorama resumido, ilustrado na figura 40. Esse resumo abrange tanto as contas registradas quanto os compromissos agendados para o dia, atuando como um instantâneo das obrigações mais importantes registradas no aplicativo. Por conseguinte, esse panorama cumpre um papel primordial, indo de encontro à tarefa essencial que foi atribuída a esse desenvolvimento: facilitar o gerenciamento dos compromissos associados à convivência em uma mesma residência.

Figura 40 - Tela inicial do aplicativo após efetuação de cadastros

Fonte: Elaborado pelo autor (2023)

6. CONSIDERAÇÕES FINAIS

O cerne deste projeto foi a concepção de uma aplicação móvel ancorada nas principais dificuldades identificadas no convívio entre indivíduos compartilhando um mesmo lar. A partir desse contexto, surgiu a *RoomMate*, uma aplicação com a missão de simplificar a administração das obrigações fundamentais nesse cenário compartilhado.

Apesar da abrangência das funcionalidades oferecidas pela aplicação, que abordaram de forma eficaz os desafios ligados à gestão de compromissos de quem divide a moradia, o processo de desenvolvimento deve ser continuado para atender integralmente os requisitos levantados nesse projeto, como também implementar novas ideias que surgiram durante seu desenvolvimento e que devem ser consideradas para futuras implementações.

No âmbito dos próximos passos para que todos os requisitos sejam contemplados, Para o #RF05 - CRUD de contas, deve-se permitir que haja a divisão automática dos valores para os moradores registrados na casa. Nos requisitos funcionais #RF05 - CRUD de contas, #RF06 - CRUD de eventos, e #RF07 - CRUD de eventos, deve-se permitir apenas que o autor de seu registro o edite e o exclua deve ser concluído. Na tela inicial, após efetuado o login, a barra de pesquisa ainda se encontra sem funcionalidade, devendo ser implementada para atender ao 12º Requisito Funcional.

Sobre as novas ideias que deverão ser implementadas, inclui-se: a criação de perfis individuais com avaliações de convivência feitas por aqueles que já compartilharam o espaço, criando uma rede de referências para comportamentos de convivência futuros; a inclusão de uma página dedicada a interações sociais, fornecendo recomendações de filmes, eventos locais e interesses compartilhados entre os moradores; a introdução de um novo tipo de usuário voltado para empreendedores, permitindo a promoção de serviços, sendo sugeridos aos moradores locais com base em proximidade, simplificando o acesso a serviços que frequentemente são desafiadores de encontrar; a publicação da aplicação nas lojas virtuais, possibilitando downloads.

Em última análise, a aplicação resultante do desenvolvimento aborda diretamente as necessidades prementes de aprimoramento na administração e convivência em ambientes residenciais compartilhados. Ao centralizar as tarefas dos moradores em um único ponto e fornecer informações sintetizadas para novos integrantes, a aplicação consegue atender às exigências essenciais para uma melhoria significativa na gestão e coexistência em contextos de convívio residencial compartilhado.

REFERÊNCIAS

BRASIL. Constituição (1988). **Constituição da República Federativa do Brasil**. Brasília, DF: Senado Federal: Centro Gráfico, 1988.

DOCKER. **Referência do Docker**, 2023 Disponível em: <https://docs.docker.com/reference/>. Acesso em: 01 jul. 2023.

FASTIFY. **Documentação do Fastify**, 2023 Disponível em: <https://fastify.dev/docs/latest/Reference/>. Acesso em: 15 jun. 2023.

IONIC. **Documentação do Ionic**, 2023. Disponível em: <https://ionicframework.com/docs>. Acesso em: 10 fev. 2023.

JWT.IO. **Introdução ao JSON Web Token (JWT)**, 2023. Disponível em: <https://jwt.io/introduction>. Acesso em: 01 jul. 2023.

MESICK, Jacob. **Logo Design in CorelDRAW Part 1**. Disponível em: <https://kb.corel.com/en/128038>. Acesso em 01 dez. 2022.

NODE.JS. **Documentação do Node.js**. 2023. Disponível em: <https://nodejs.org/pt-br/docs>. Acesso em: 03 jun. 2023.

POSTGRESQL. **Documentação do PostgreSQL**, 2023. Disponível em: <https://www.postgresql.org/docs/current/>. Acesso em: 01 jul. 2023.

PRISMA. **Documentação do Prisma ORM**, 2023. Disponível em: <https://www.prisma.io/docs>. Acesso em: 15 jun. 2023.

TYPESCRIPT. **Handbook - TypeScript em 5 minutos**. 2023. Disponível em: <https://www.typescriptlang.org/pt/docs/handbook/typescript-in-5-minutes.html>. Acesso em: 05 jan. 2023.

APÊNDICE A – QUESTIONÁRIO APLICADO PARA LEVANTAMENTO DE REQUISITOS

Problemas de gerenciamento de compromissos na convivência entre pessoas que dividem residência.

* Indica uma pergunta obrigatória

1. E-mail *

2. Como fica acordado as regras da casa entre as pessoas? *

Marcar apenas uma oval.

- Verbalmente, sem registros das regras
- Por contrato
- Não é conversado sobre as regras da casa
- Não existem regras da casa

3. Você já teve problemas devido ao não cumprimento das regras da casa? *

Marcar apenas uma oval.

- Sim
- Não

4. Sobre as contas, elas são rateadas ou já estão inclusas no valor do aluguel? *

Marcar apenas uma oval.

- As contas são rateadas.
- As contas estão inclusas no valor do aluguel

5. Se há rateio nas contas, como ele é feito? *

Marcar apenas uma oval.

- Dividido por calculadora e informado por aplicativo de mensagem ou verbalmente
- Não rateio as contas

6. Sobre a agenda de compromissos da casa, como é feito o controle? *

Marcar apenas uma oval.

- Por escrito, em avisos expostos na casa
- Verbalmente, sem registro de agenda
- Por aplicativos de mensagem, sem notificação no dia do compromisso

7. Quando há decisões a serem tomadas em conjunto pelos integrantes da residência *

Marcar apenas uma oval.

- Fica difícil se reunir presencialmente, pois o horário de cada um é diferente
- Combinamos verbalmente, sem registro do que foi decidido
- Por aplicativos de mensagem, sem notificação no dia do compromisso

8. Quando alguém vai receber visita, como fica combinado? *

Marcar apenas uma oval.

- Informa aos demais moradores, afinal é alguém "estranho" em sua casa.
- Não é informado e não tem problema quanto a isso
- Não é informado e existe problema quanto a isso

9. Quanto ao regimento e demais documentos da residência. Hoje é disponibilizado em algum lugar fixo? Há a necessidade de enviar cada vez que chegar um novo morador? *

Marcar apenas uma oval.

- É disponibilizado em um lugar fixo
- Há a necessidade de enviar cada vez que chega um novo morador

10. Todas essas informações são importantes para um bom convívio entre as pessoas que dividem a residência. Você acrescentaria mais alguma questão que não foi falada nas perguntas anteriores? *

Este conteúdo não foi criado nem aprovado pelo Google.

Google Formulários