



Benedito TIZITO - Aplicação móvel para
Barbosa Ribeiro organizar compromissos de uma
Neto agenda social.

Trabalho de Projeto submetido como requisito parcial para obtenção do grau de **Mestre em Engenharia de Software**.

Júri

Presidente (José António Moinhos Cordeiro, ESTSetúbal/IPS)

Orientador (Cédric Claude Bernard Grueau, ESTSetúbal/IPS)

Arguente (Fausto José da Silva Valentim Mourato, ESCE/IPS)

Novembro de 2024

*Dedico este trabalho à minha mãe, cuja fé e
sabedoria sempre me guiaram.*

O seu incentivo trouxe-me de volta à universidade.

*O seu exemplo de determinação inspirou-me a
alcançar os meus objetivos.*

O seu empenho ajudou-me a ultrapassar obstáculos.

Este mestrado é tanto dela como meu.

Agradecimentos

Agradeço à minha mãe, Socorro Ferreira, e ao meu pai, João Barbosa, pelo apoio e incentivo ao longo de toda esta jornada. À minha irmã, Bruna Ribeiro, pelas sugestões que elevaram a qualidade deste trabalho. À minha amiga, Isabel Neves, pelos ajustes que tornaram o texto mais claro e fluido. Aos meus amigos, Patrícia Jesus, Pedro Mota e Edna Martins, pela validação do Tizito durante a fase de prototipagem. Ao meu orientador, Prof. Cédric Grueau, pelas sugestões que enriqueceram o conteúdo deste trabalho. Agradeço também a todos os meus amigos e colegas que, de forma direta ou indireta, contribuíram para a elaboração deste trabalho. Por fim, a minha gratidão especial à Daniela Domingues, cuja presença na minha vida foi essencial para que os meus sonhos se tornassem realidade.

Resumo

As mídias sociais assumem um papel notório na forma como as pessoas cooperam e realizam tarefas colaborativas. Uma análise da aplicação WhatsApp permitiu identificar padrões no comportamento dos utilizadores. Dessa análise, nasceu a oportunidade de conceber uma nova aplicação na área de programação e coordenação de grupos. Este trabalho apresenta a concepção de um novo produto de *software*, idealizado para marcar encontros pessoais entre amigos e guardar as recordações desses encontros. Trata-se de um produto de mídia social denominado Tizito, uma aplicação móvel para os sistemas operativos Android e iOS, cujo desenvolvimento utilizou a ferramenta Figma para a criação do protótipo de *interface*, a linguagem Dart para a escrita do código-fonte com suporte do *framework* Flutter, e os serviços de infraestrutura oferecidos pela plataforma Firebase. Após o desenvolvimento, um teste supervisionado permitiu validar a consistência dos dados e comportamento da aplicação. Por fim, um teste não supervisionado com 50 indivíduos mostrou pontos de melhoria no produto antes do seu lançamento. Por exemplo, criar mecanismos para que o utilizador possa adicionar um amigo; comunicar de forma eficiente as formas de adicionar um amigo; permitir a publicação de mais fotos, vídeos e comentários. A aplicação desenvolvida ilustra a possibilidade de criação de um novo mercado e apresenta um conjunto de tecnologias que podem servir de referência para o desenvolvimento de produtos com características semelhantes.

Palavras-chave: Mídia social, Aplicação móvel, Figma, Dart, Flutter, Firebase.

Abstract

Social media plays a significant role in the way people cooperate and perform collaborative tasks. An analysis of the WhatsApp application allowed us to identify patterns in user behavior. This analysis led to the opportunity to design a new application in the area of programming and group coordination. This paper presents the design of a new software product, designed to schedule personal meetings between friends and save memories of these meetings. This is a social media product called Tizito, a mobile application for the Android and iOS operating systems, whose development used the Figma tool to create the interface prototype, the Dart language to write the source code with support from the Flutter framework, and the infrastructure services offered by the Firebase platform. After development, a supervised test allowed us to validate the consistency of the data and behavior of the application. Finally, an unsupervised test with 50 individuals showed areas for improvement in the product before its launch. For example, creating mechanisms for the user to add a friend; efficiently communicating the ways to add a friend; allowing the publication of more photos, videos and comments. The developed application illustrates the possibility of creating a new market and presents a set of technologies that can serve as a reference for the development of products with similar characteristics.

Keywords: Social Media, Mobile App, Figma, Dart, Flutter, Firebase.

Índice

Agradecimentos	4
Resumo	5
Abstract.....	6
Índice	7
Lista de Figuras.....	10
Lista de Tabelas	13
Capítulo 1	14
Introdução	14
1.1. Contextualização	15
1.2. Problema	15
1.3. Objetivos	19
1.4. Justificação.....	20
1.4.1. Qual o problema a ser resolvido?	20
1.4.2. Para quem o problema será resolvido?	20
1.4.3. Qual o tamanho dessa oportunidade?	20
1.4.4. Que alternativas existem?	21
1.4.5. Que qualificações são necessárias?	21
1.4.6. Por que agora?	22
1.4.7. Como levar essa oportunidade ao mercado?	22
1.4.8. Como medir o sucesso e ganhar dinheiro?	22
1.4.9. Que fatores são críticos para o sucesso?	22
1.4.10. Diante do exposto, qual é a recomendação?	23
1.5. Estrutura do trabalho.....	23
Capítulo 2	24
Conceção da ideia.....	24
2.1. Origem	25
2.2. Marca	26
2.3. Conceito	27
Capítulo 3	30
Desenvolvimento do produto	30
3.1. Definição dos requisitos	31
3.1.1. 5W2H.....	31
3.1.2. Histórias do utilizador	32
3.2. Elaboração da interface.....	33
3.2.1. Sketches	33

3.2.2. Wireframes	34
3.2.3. Sitemaps.....	35
3.2.4. Referências.....	37
3.2.5. Padrões de design.....	38
3.2.6. Bibliotecas de padrões	39
3.2.7. Imagens	40
3.2.8. Protótipo.....	41
3.3. Construção da aplicação	42
3.3.1. Tecnologia	42
3.3.2. Arquitetura	42
3.3.3. Infraestrutura	48
Capítulo 4	50
Tizito	50
4.1. Acesso	51
4.1.1. Inicialização	52
4.1.2. Apresentação.....	53
4.1.3. Entrada	54
4.2. Conta.....	55
4.2.1. Registo.....	56
4.2.2. Palavra-passe.....	57
4.2.3. Principal	58
4.3. Amizade	59
4.3.1. Código QR	60
4.3.2. Lista de amigos.....	61
4.3.3. Amigo.....	62
4.3.4. Solicitação de amizade.....	63
4.4. Encontros	64
4.4.1. Lista de encontros	65
4.4.2. Título	66
4.4.3. Data e hora	67
4.4.4. Local	68
4.4.5. Tipo	69
4.4.6. Convidados.....	70
4.4.7. Encontro marcado	71
4.4.8. Encontro desmarcado.....	72
4.4.9. Encontro passado.....	73
4.5. Convites	74
4.5.1. Lista de convites	75
4.5.2. Convite aceite	76
4.5.3. Convite recusado.....	77

4.5.4. <i>Visibilidade</i>	78
4.5.5. <i>Encontro</i>	79
4.5.6. <i>Mensagens</i>	80
4.6. Recordações	81
4.6.1. <i>Fotos</i>	82
4.6.2. <i>Vídeos</i>	83
4.6.3. <i>Comentários</i>	84
4.7. Início	85
4.7.1. <i>Encontro público</i>	86
4.7.2. <i>Gostos</i>	87
4.7.3. <i>Comentários</i>	88
4.8. Notificações	89
4.8.1. <i>Notificação permanente</i>	90
4.8.2. <i>Notificação transitória</i>	91
4.9. Auxiliares	92
4.9.1. <i>Definições</i>	93
4.9.2. <i>Ajuda</i>	94
4.9.3. <i>Sair</i>	95
Capítulo 5	96
Análise dos resultados e discussão	96
5.1. Teste supervisionado	97
5.2. Teste não supervisionado	100
5.3. Sugestões e críticas	103
Capítulo 6	107
Conclusão	107
6.1. Considerações	108
6.2. Trabalhos futuros	109
Bibliografia	110

Lista de Figuras

Figura 1.1 - Exemplos de grupos temporários no WhatsApp.....	16
Figura 1.2 - Fluxo para divulgar ou decidir os detalhes de eventos sociais.....	17
Figura 1.3 - Resultados da sondagem.....	18
Figura 1.4 - Utilizadores da internet por cada tipo de dispositivo.	20
Figura 1.5 - Aplicações móveis por utilizadores ativos.	21
Figura 1.6 - Parcela do tráfego da web por sistema operativo.	21
Figura 1.7 - Utilizadores da internet por cada tipo de aplicação móvel.	22
Figura 2.1 - Detalhes do encontro.	25
Figura 2.2 - Exemplo de bilhete.	25
Figura 2.3 - Exemplos de álbuns de recortes.	26
Figura 2.4 - Marca Tizito.....	26
Figura 2.5 - Certificado de registo da marca Tizito.	27
Figura 2.6 - Comunicação entre o anfitrião e seus convidados.	28
Figura 2.7 - Linha temporal.	29
Figura 3.1 - Método 5W2H.....	31
Figura 3.2 - Perguntas determinantes sobre o Tizito.	31
Figura 3.3 - Histórias do utilizador do Tizito.....	32
Figura 3.4 - Funcionalidades do Tizito.	32
Figura 3.5 - <i>Sketches</i> do Tizito.....	33
Figura 3.6 - <i>Wireframes</i> do Tizito.....	34
Figura 3.7 - <i>Sitemaps</i> do Tizito.	35
Figura 3.8 - Estrutura do Tizito.	36
Figura 3.9 - Referências para o Tizito.....	37
Figura 3.10 - Padrões de <i>design</i>	38
Figura 3.11 - Biblioteca de padrões.....	39
Figura 3.12 - Banco de imagens.	40
Figura 3.13 - Modificação de imagens.....	40
Figura 3.14 - Protótipo do Tizito.	41
Figura 3.15 - Diagrama de classes do Tizito.....	43
Figura 3.16 - Classe que representa um ecrã.	44
Figura 3.17 - Classe que representa um controlador.	45
Figura 3.18 - Classe que representa um serviço.....	45
Figura 3.19 - Classe que representa uma tarefa específica.	46
Figura 3.20 - Classe que representa um <i>widget</i> personalizado.	47

Figura 3.21 - Classe que representa um teste de unidade.	48
Figura 3.22 - Painel do Firebase.	49
Figura 4.1 - <i>Sitemap</i> do acesso.	51
Figura 4.2 - Ecrã de inicialização.	52
Figura 4.3 - Ecrã de apresentação.	53
Figura 4.4 - Ecrã de entrada.	54
Figura 4.5 - <i>Sitemap</i> da conta.	55
Figura 4.6 - Ecrã de registo.	56
Figura 4.7 - Redefinição de palavra-passe.	57
Figura 4.8 - Ecrã principal.	58
Figura 4.9 - <i>Sitemap</i> da amizade.	59
Figura 4.10 - Código QR.	60
Figura 4.11 - Lista de amigos.	61
Figura 4.12 - Ecrã de amigo.	62
Figura 4.13 - Solicitação de amizade.	63
Figura 4.14 - <i>Sitemap</i> dos encontros.	64
Figura 4.15 - Lista de encontros.	65
Figura 4.16 - Título do encontro.	66
Figura 4.17 - Data e hora do encontro.	67
Figura 4.18 - Local do encontro.	68
Figura 4.19 - Tipo do encontro.	69
Figura 4.20 - Convidados do encontro.	70
Figura 4.21 - Encontro marcado.	71
Figura 4.22 - Encontro desmarcado.	72
Figura 4.23 - Encontro passado.	73
Figura 4.24 - <i>Sitemap</i> dos convites.	74
Figura 4.25 - Lista de convites.	75
Figura 4.26 - Convite aceite.	76
Figura 4.27 - Convite recusado.	77
Figura 4.28 - Visibilidade.	78
Figura 4.29 - Ecrã do encontro.	79
Figura 4.30 - Mensagens do encontro.	80
Figura 4.31 - <i>Sitemap</i> das recordações.	81
Figura 4.32 - Fotos do encontro.	82
Figura 4.33 - Vídeos do encontro.	83
Figura 4.34 - Comentários do encontro.	84
Figura 4.35 - <i>Sitemap</i> do início.	85
Figura 4.36 - Encontro público.	86
Figura 4.37 - Gostos do encontro.	87
Figura 4.38 - Comentários do encontro.	88

Figura 4.39 - <i>Sitemap</i> das notificações.	89
Figura 4.40 - Notificação permanente.	90
Figura 4.41 - Notificação transitória.	91
Figura 4.42 - <i>Sitemap</i> dos auxiliares.	92
Figura 4.43 - Opção “Definições”.	93
Figura 4.44 - Opção “Ajuda”	94
Figura 4.45 - Opção “Sair”	95
Figura 5.1 - Teste supervisionado.	97
Figura 5.2 - <i>Bug</i> detetado.	98
Figura 5.3 - <i>Log</i> da aplicação.	99
Figura 5.4 - Variável <i>nullable</i>	99
Figura 5.5 - Método <i>dispose</i>	99
Figura 5.6 - Biblioteca <i>flutter_google_places</i>	100
Figura 5.7 - Google Play Console.	100
Figura 5.8 - Utilizadores registados.	101
Figura 5.9 - Único encontro público.	102
Figura 5.10 - Formulário para sugestões e críticas.	103

Lista de Tabelas

Tabela 5.1 – Encontros registrados.	101
Tabela 5.2 – Números associados aos encontros.....	102
Tabela 5.3 – Sugestões.	104
Tabela 5.4 – Críticas.	106

Capítulo 1

Introdução

Este capítulo apresenta o conceito e contexto das mídias sociais, além de abordar o problema a ser resolvido e os objetivos pretendidos por este trabalho.

1.1. Contextualização

Embora a maioria das pessoas reconheça um serviço de mídia social quando o vê, isso não significa que seja fácil descrevê-lo. Muitas vezes, o termo “mídia social” é utilizado como sinónimo de “rede social”. Na verdade, uma rede social é um conceito que já existia muito antes do surgimento da internet. Refere-se a um grupo de pessoas que se conhecem ou que estão ligadas de alguma forma. Cada pessoa tem as suas próprias redes sociais, formadas por amigos, familiares e colegas. Essas redes sobrepõem-se e interagem com as redes sociais das outras pessoas. Resumindo, o termo “rede social” é um conceito sociológico, enquanto “mídia social” é um serviço tecnológico (Mc Mahon, 2021).

Durante a década de 1990, a internet tornou-se disponível publicamente nos Estados Unidos, e pouco depois no resto do mundo. No final do milénio, ocorreu uma revolução no conteúdo gerado para o utilizador, conhecida como “Web 2.0”, e os sites tornaram-se mais interativos e fáceis de utilizar. Por volta de 2010, conforme aumentou a capacidade de partilhar fotos e vídeos, o termo “mídia social” tornou-se mais popular. Alguns sites anunciavam-se como desenvolvidos explicitamente para partilhar formas específicas de mídia digital. Por exemplo, Last.FM para música, YouTube para vídeos e Flickr para fotos. Utilizar o termo “mídia social” era útil para os distinguir de sites como Friendster, Myspace e Bebo, que eram voltados principalmente para as redes sociais. No entanto, quase todos os sites que antes teriam sido classificados como um site de rede social agora oferecem aos seus utilizadores a possibilidade de partilhar a maioria das formas de mídia digital (Mc Mahon, 2021).

As mídias sociais são serviços *online* que incentivam os seus utilizadores a digitalizar e partilhar publicamente informações pessoais que antes eram privadas. Esses serviços tentam constantemente persuadir os utilizadores a revelarem algo sobre si próprios. Os mais famosos atualmente têm uma origem claramente norte-americana, como Facebook, YouTube, Twitter, Instagram, LinkedIn e Snapchat. Por algum tempo no início da década de 2010, parecia que um novo site de mídia social era lançado por semana, mas essa tendência desapareceu. Surgiram novos serviços, como Ello, Peach e Sararah, mas nenhum teve um impacto duradouro. Existe uma uniformidade considerável na mídia social de hoje (Mc Mahon, 2021).

1.2. Problema

A observação é um método que se baseia em ver as pessoas a enfrentar um problema no contexto em que ele ocorre. Para uma boa observação, é importante ter um conjunto de hipóteses. A cada ronda de observações, essas hipóteses são avaliadas e ajustadas até que um padrão seja encontrado.

Esse método foi utilizado para encontrar padrões no comportamento dos utilizadores do WhatsApp (<https://www.whatsapp.com/>). Além do envio instantâneo de mensagens de texto e áudio, esse produto permite a partilha de fotos, vídeos e documentos. Ele também permite a criação de grupos de conversa, que facilitam a comunicação entre vários contactos

simultaneamente. Desse modo, verifica-se que alguns utilizadores têm a prática de criar grupos temporários, com o intuito de reunir contactos específicos para divulgar ou decidir os detalhes de eventos sociais, como aniversários, almoços, *happy hours*, festas e outras formas de confraternização.



Figura 1.1 - Exemplos de grupos temporários no WhatsApp.

No total, 10 desses grupos foram observados e 2 papéis foram identificados:

- **Organizador:** utilizador do WhatsApp que cria o grupo e divulga os detalhes do evento. Esse papel também pode criar consultas ou sondagens para auxiliar na tomada de decisões, como a melhor data para o evento. Se necessário, ele pode cancelar o evento;
- **Participante:** contacto adicionado ao grupo pelo organizador, que visualiza os detalhes do evento. Esse papel também pode contribuir com sugestões ou preferências para auxiliar na tomada de decisões, como o local adequado para o evento.

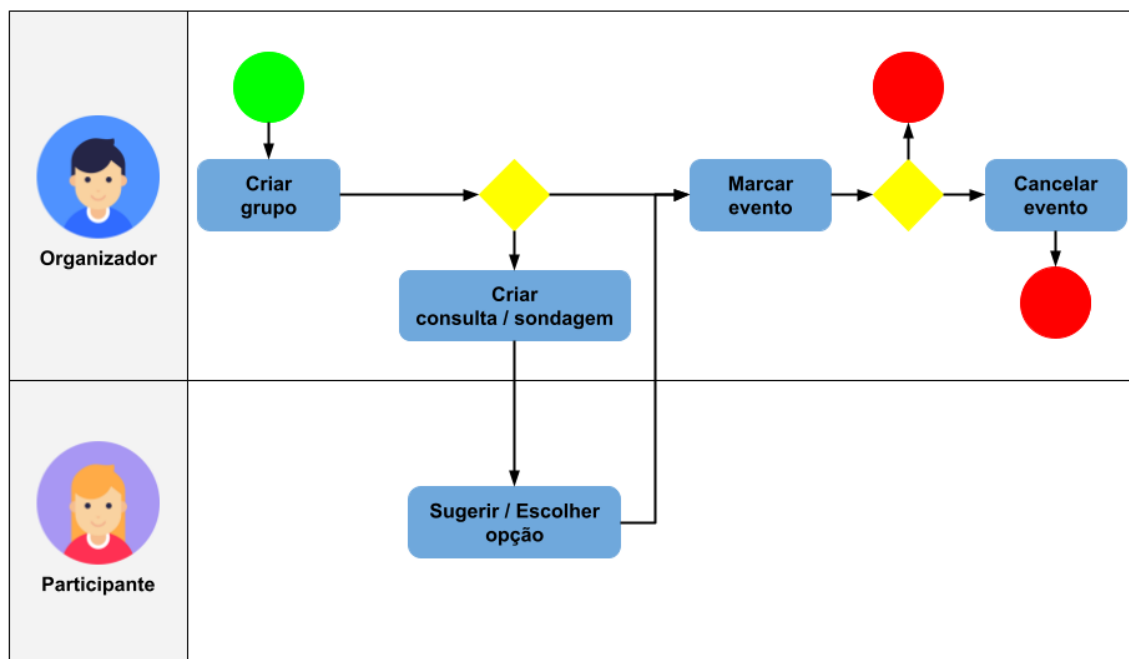


Figura 1.2 - Fluxo para divulgar ou decidir os detalhes de eventos sociais.

A partir da interação desses papéis, um padrão foi encontrado antes da realização dos eventos:

- Os detalhes do evento nem sempre são apresentados de forma clara e estruturada, o que dificulta a compreensão dos participantes;
- Quando o organizador cria consultas ou sondagens, inúmeras mensagens podem ser geradas, o que dificulta a recolha de respostas. Quanto mais participantes, mais complexo é decidir uma data ou local para o evento;
- Em alguns casos, mensagens relacionadas a temas secundários misturam-se com as que tratam do tema principal, o que dificulta o acompanhamento da conversa tanto para o organizador como para os participantes;
- Verifica-se que alguns participantes perdem o interesse e param de ler as mensagens à medida que o número de mensagens aumenta, o que dificulta a divulgação dos detalhes do evento;
- Quando o organizador cancela o evento, não há uma notificação específica e efetiva, o que dificulta a comunicação com os participantes. Normalmente, uma nova mensagem de texto ou áudio é enviada ao grupo, que pode se perder em meio a outras mensagens, sem o devido destaque e importância. Para exemplificar, alguns participantes podem comparecer ao local do evento sem saber que ele foi cancelado.

É importante realçar que essa observação decorreu sem a interferência do observador, para que o comportamento observado refletisse a realidade. Assim, outro padrão foi encontrado depois da realização dos eventos:

- Fotos e vídeos são frequentemente partilhados no grupo. No entanto, não existe um local específico na web para armazenar esses ficheiros. Desse modo, tanto o organizador como os participantes mantêm cópias individuais de cada foto e vídeo nos

seus dispositivos móveis. Contudo, em caso de dano ou perda do dispositivo, esses ficheiros podem tornar-se irrecuperáveis;

- Até mesmo durante a realização do evento, comentários são recorrentes, seja para enfatizar aspectos positivos ou negativos, seja para relatar momentos incomuns, ou simplesmente para agradecer a participação de todos. Esses comentários também são uma forma de recordação. No entanto, eles serão perdidos caso o utilizador opte por sair e apagar o grupo.

Embora os padrões encontrados apontem para um problema a ser resolvido, foi realizada uma sondagem com 50 indivíduos pertencentes aos grupos observados para avaliar o potencial interesse num novo produto de *software*. De forma anónima, eles responderam a seguinte pergunta: “Você utilizaria uma aplicação¹ para encontrar seus amigos e guardar as recordações desse encontro (fotos, vídeos e comentários)?” Para partilhar a pergunta e recolher as respostas, essa sondagem utilizou a ferramenta Google Forms.



Figura 1.3 - Resultados da sondagem.

Os resultados da sondagem revelaram que 44 indivíduos demonstraram disposição em utilizar a aplicação proposta, indicando a existência de um público potencialmente interessado no produto. Além disso, é importante destacar que 1 indivíduo apresentou uma condição para utilizar a aplicação, referindo que atualmente utiliza o WhatsApp e Instagram. Em conjunto, esses produtos oferecem funcionalidades semelhantes às da aplicação. Esse indivíduo sugere que a aplicação poderia substituir esses produtos caso ofereça algo inovador ou praticidade.

Por outro lado, 5 indivíduos não demonstraram disposição em utilizar a aplicação. Embora os motivos para essa rejeição não tenham sido detalhados, é possível que estejam

¹ [Portugal] [Informática] Programa informático que visa facilitar a realização de uma tarefa num computador ou num dispositivo móvel. [Equivalente no português do Brasil: aplicativo.]

"aplicação", in Dicionário Priberam da Língua Portuguesa [em linha], 2008-2024, <https://dicionario.priberam.org/aplicação>.

relacionados à capacidade de comunicação da proposta, exaustão no uso de aplicações, preocupações com privacidade, etc. Esses fatores podem ser explorados em pesquisas futuras para entender melhor essa rejeição e refinar a proposta da aplicação.

Como referido, o Instagram (<https://www.instagram.com/>) é um produto para partilha de fotos e vídeos, que podem ser editados, organizados por *tags* e associados a uma localização. Além disso, os utilizadores podem interagir por meio de comentários e gostos. No entanto, quando o organizador e participantes desejam que as fotos e vídeos não sejam visualizados por outras pessoas, mesmo em contas privadas, isso pode ser uma tarefa complexa, pois todos teriam que restringir quem pode visualizar cada publicação. Essa abordagem também exige que todos tenham uma conta no Instagram, o que dificulta o acesso para aqueles que, por qualquer motivo, preferem não utilizar esse produto.

1.3. Objetivos

“Inovação” é um termo utilizado com frequência por várias pessoas, mas cada uma tem uma definição diferente. Para algumas, inovação está associada a algo criativo, algo que não existia antes, algo diferente do que se encontra no dia a dia. Para outras, inovação está associada a algo tecnológico, como computadores quânticos, transmissão de eletricidade sem fios, edição de genoma, realidade virtual, realidade aumentada, nanotecnologia, etc.

No entanto, inovação não se resume apenas a algo criativo ou tecnológico. Para inovar, é preciso conhecer as tecnologias disponíveis e saber como utilizá-las para resolver um problema de um grupo de pessoas. Portanto, a criação de um novo produto de *software* é uma forma de inovação e deve começar pelo entendimento do problema, das pessoas que o enfrentam, do contexto em que ele ocorre e das motivações que essas pessoas têm para vê-lo resolvido (Torres, 2020).

Este trabalho tem como objetivo conceber e desenvolver um produto de *software* que permite a divulgação dos detalhes de eventos sociais e o armazenamento das recordações associadas a esses eventos. Para isso, este trabalho apresenta a ideia, bem como as funcionalidades e formas de interação entre o utilizador e o produto, além de descrever os métodos, ferramentas e boas práticas utilizados nas etapas de definição dos requisitos, elaboração da *interface* e construção da aplicação. Por fim, este trabalho visa realizar testes e recolher *feedback* inicial para orientar ajustes e melhorias no produto.

Embora a tomada de decisões sobre os detalhes de eventos sociais faça parte do problema apresentado, o produto não inclui a criação de consultas ou sondagens. Do mesmo modo, não inclui a edição de fotos e vídeos, a fim de reduzir o custo de desenvolvimento neste estágio inicial. Lançar no mercado e identificar uma forma de produzir receita não fazem parte do objetivo, pois exigem conhecimento em áreas que vão além do desenvolvimento do produto. Comparar e avaliar se o produto poderia substituir o WhatsApp e Instagram também não fazem parte do objetivo, pois considera-se necessário que o produto seja adotado por um grupo maior de utilizadores.

1.4. Justificação

Para descobrir se o problema é uma oportunidade a ser perseguida, este trabalho utilizou o **modelo de análise de oportunidade** proposto por Marty Cagan. As 10 perguntas do modelo foram respondidas com o suporte do relatório **DIGITAL 2021 (PORTUGAL)**, apresentado pela biblioteca **DataReportal** (<https://datareportal.com>), que oferece centenas de relatórios gratuitos repletos de dados, *insights* e tendências para ajudar pessoas e organizações ao redor do mundo a tomar decisões mais informadas.

1.4.1. Qual o problema a ser resolvido?

Reunir contactos específicos para divulgar ou decidir os detalhes de eventos sociais, como a data e local. Normalmente, esses eventos têm o propósito de reunir amigos, familiares, colegas de escola, universidade ou trabalho.

1.4.2. Para quem o problema será resolvido?

Utilizadores da internet com idade entre 16 e 64 anos que possuem *smartphone*.

1.4.3. Qual o tamanho dessa oportunidade?

De acordo com o relatório, 98% dos utilizadores da internet com idade entre 16 e 64 anos possuem *smartphone*.

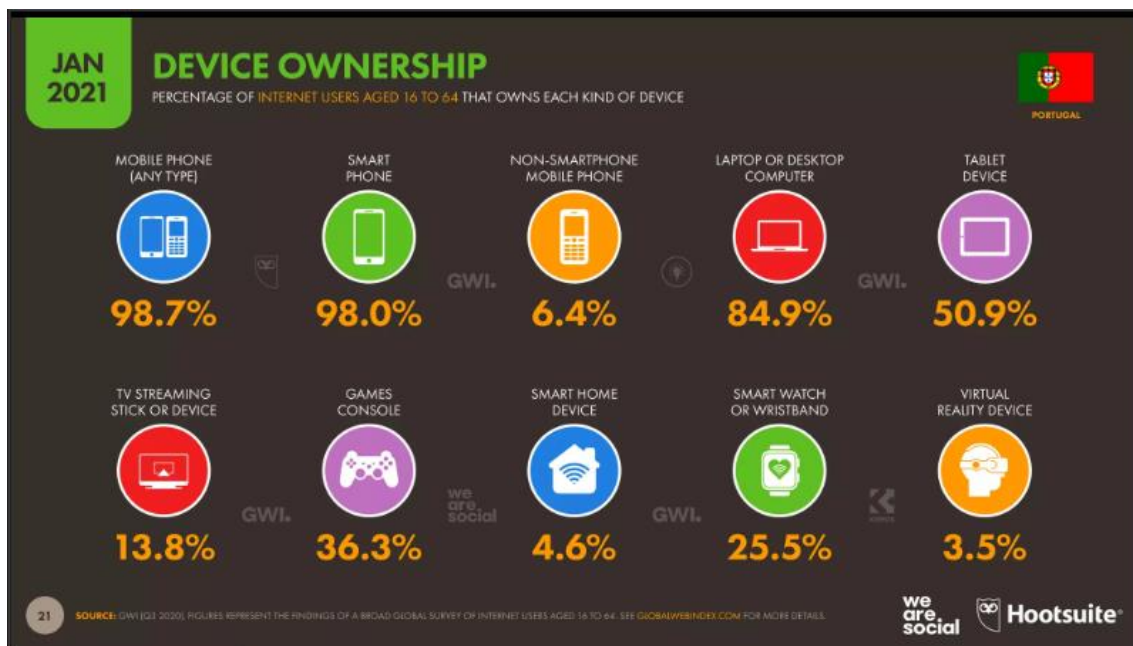


Figura 1.4 - Utilizadores da internet por cada tipo de dispositivo.

1.4.4. Que alternativas existem?

Em conjunto, o WhatsApp e Instagram oferecem funcionalidades que podem ser utilizadas para resolver o problema.



Figura 1.5 - Aplicações móveis por utilizadores ativos.

1.4.5. Que qualificações são necessárias?

Conhecimento em construção de aplicações móveis para Android e iOS.

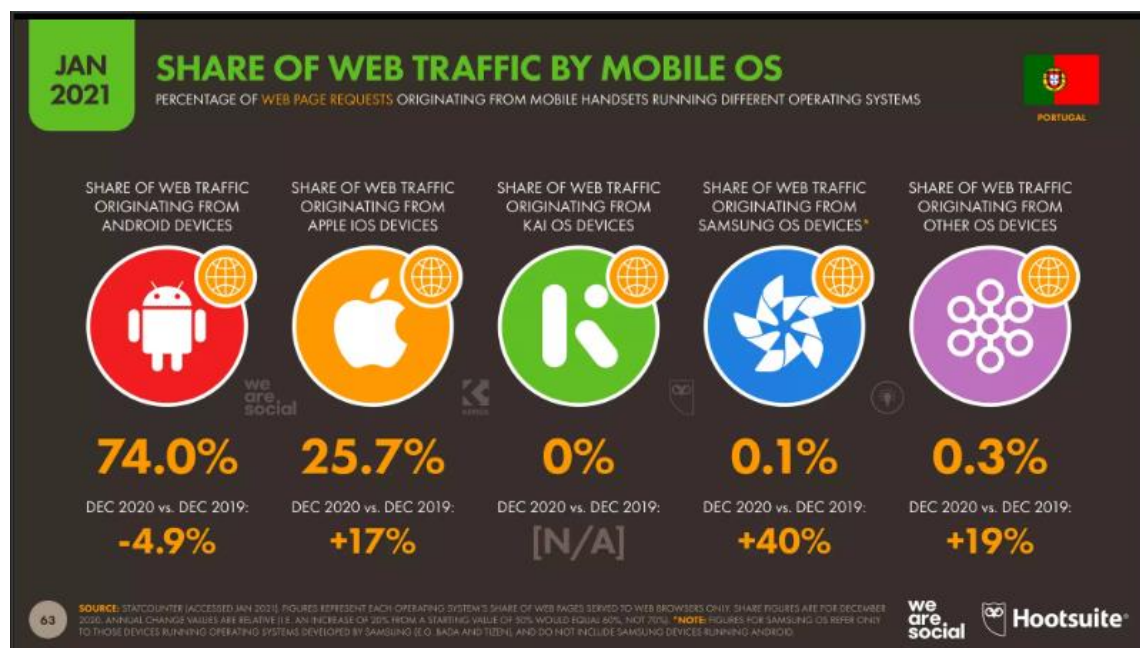


Figura 1.6 - Parcela do tráfego da web por sistema operativo.

1.4.6. Por que agora?

De acordo com o relatório, 95,8% dos utilizadores da internet com idade entre 16 e 64 anos utilizam aplicações móveis para troca de mensagens. Considerando que parte dessas mensagens visa organizar eventos sociais, existe a possibilidade de um novo mercado ser criado.

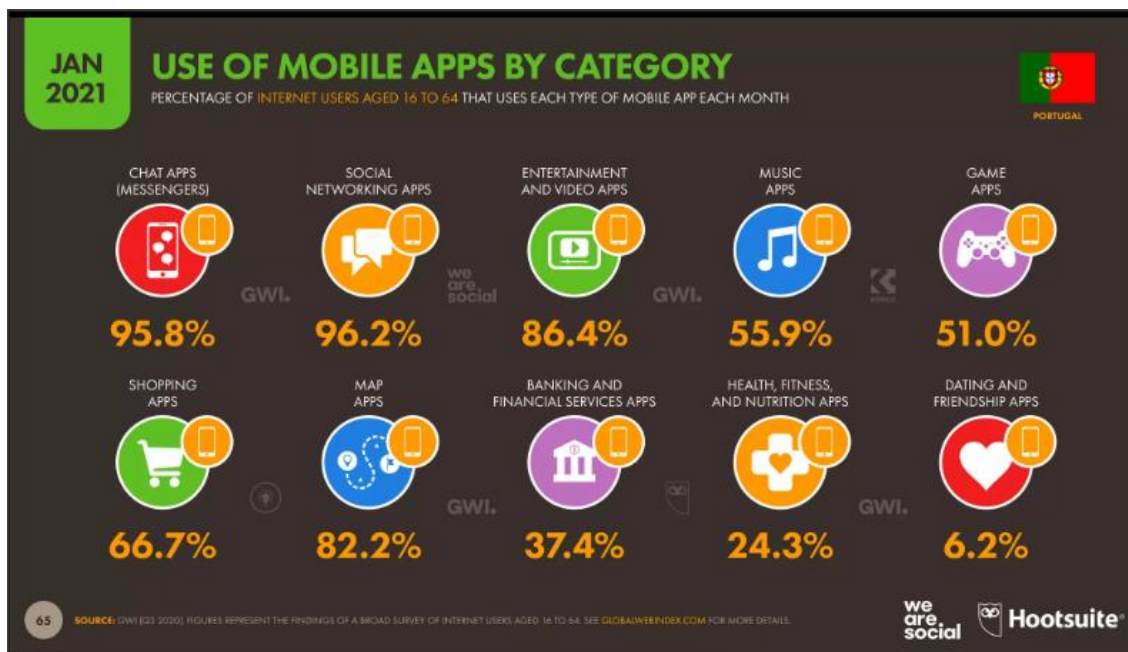


Figura 1.7 - Utilizadores da internet por cada tipo de aplicação móvel.

1.4.7. Como levar essa oportunidade ao mercado?

Este trabalho não pretende lançar o produto no mercado, pois considera-se necessário conhecimento em áreas como *marketing* de produtos.

1.4.8. Como medir o sucesso e ganhar dinheiro?

Métricas são essenciais para entender o produto com mais profundidade. No entanto, este trabalho pretende recolher apenas *feedback* inicial para orientar ajustes e melhorias no produto. Este trabalho não pretende identificar uma forma de produzir receita com o produto.

1.4.9. Que fatores são críticos para o sucesso?

De acordo com a sondagem realizada, 1 indivíduo sugere que o produto poderia substituir o WhatsApp e Instagram caso ofereça algo inovador ou praticidade. Outro fator é a construção do produto para Android e iOS. Em conjunto, esses sistemas operativos correspondem a 99,7% dos dispositivos móveis.

1.4.10. Diante do exposto, qual é a recomendação?

Após avaliar as respostas, conclui-se que vale a pena desenvolver o produto. Para isso, métodos e boas práticas para iniciantes serão utilizados em cada etapa do desenvolvimento. Ferramentas e bibliotecas gratuitas serão utilizadas para reduzir o custo de desenvolvimento, bem como serviços de infraestrutura que oferecem testes gratuitos e pagamento por utilização, com o intuito de minimizar tanto o custo de operação como o custo de armazenamento dos dados.

1.5. Estrutura do trabalho

Este trabalho está dividido em 6 capítulos:

- **Introdução:** este capítulo apresenta o conceito e contexto das mídias sociais, além de abordar o problema a ser resolvido e os objetivos pretendidos por este trabalho;
- **Conceção da ideia:** este capítulo apresenta a concepção de um novo produto de *software* chamado Tizito;
- **Desenvolvimento do produto:** este capítulo apresenta os métodos, ferramentas e boas práticas utilizados nas etapas de definição dos requisitos, elaboração da *interface* e construção da aplicação;
- **Tizito:** este capítulo apresenta as funcionalidades e formas de interação entre o utilizador e o produto;
- **Análise dos resultados e discussão:** este capítulo apresenta os testes realizados e o *feedback* inicial para orientar ajustes e melhorias no produto;
- **Conclusão:** este capítulo apresenta uma reflexão sobre os objetivos e resultados, resumindo os principais contributos e enumerando possíveis trabalhos futuros.

Capítulo 2

Conceção da ideia

Este capítulo apresenta a conceção de um novo produto de *software* chamado Tizito.

2.1. Origem

Agenda é o conjunto dos compromissos a serem tratados por alguém. O tipo de agenda é definido pela natureza dos compromissos. Por exemplo, reuniões com fornecedores, palestras com funcionários e visitas a clientes são associados a uma agenda profissional; participação em aulas, realização de exames e presença em congressos são associados a uma agenda académica; cinema com os amigos, almoço com a família e *happy hour* com os colegas de trabalho são associados a uma agenda social.

Tizito é o nome do produto de *software* concebido e desenvolvido por este trabalho. Trata-se de uma aplicação móvel para organizar compromissos de uma agenda social, onde cada compromisso é representado por um “encontro”. O Tizito permite a marcação de encontros pessoais entre amigos, familiares, colegas de escola, universidade ou trabalho. Ele também permite a divulgação dos detalhes do encontro de forma clara e objetiva. Além disso, o Tizito permite o armazenamento das fotos e vídeos associados a cada encontro.

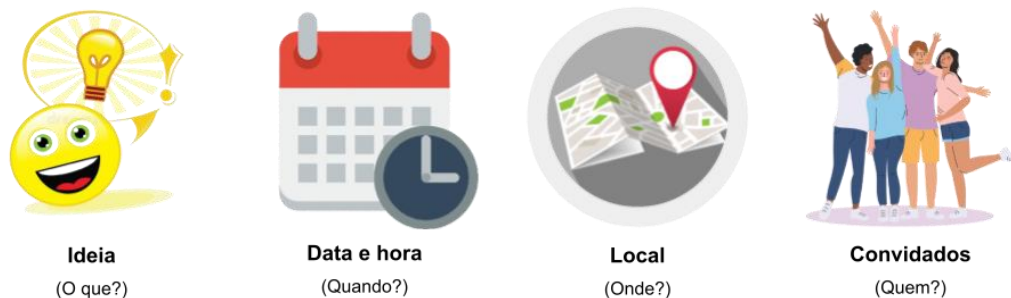


Figura 2.1 - Detalhes do encontro.

A ideia surgiu a partir de uma lembrança de infância: quando não existiam *smartphones*, a troca de bilhetes era um dos meios de comunicação utilizados pelas crianças e adolescentes durante as aulas. Normalmente, esses bilhetes eram convites para jogar à bola, andar de bicicleta ou fazer alguma outra atividade habitual naquela época. Bastava acenar com a cabeça para cima e para baixo num sinal de afirmativo para responder “Sim!”.



Figura 2.2 - Exemplo de bilhete.

Outra lembrança de infância contribuiu para a concepção da ideia: álbuns de recortes (*scrapbooks*) eram personalizados com textos, fotos, desenhos, excertos de jornais/revistas, embalagens, folhas de árvores, etc. Esses álbuns tinham a finalidade de registrar momentos especiais entre amigos.



Figura 2.3 - Exemplos de álbuns de recortes.

2.2. Marca

A marca Tizito retrata uma folha de papel arrancada de um caderno, com o nome escrito à mão (fonte *Satisfy*) e fita adesiva no topo, indicando que pode ser colada em qualquer superfície, seja um monitor de computador, um volante de carro ou uma porta de frigorífico. A inclinação para a direita (5°) remete à informalidade dos encontros, enquanto a cor laranja (#FF9800) remete à alegria. Essa marca é como um lembrete do que fazer num feriado ou fim de semana, depois de um dia de escola ou trabalho, durante o horário do almoço ou do café.



Figura 2.4 - Marca Tizito.

A marca Tizito foi registrada no Instituto Nacional da Propriedade Industrial sediado no Brasil (<https://www.gov.br/inpi/pt-br>), para garantia da propriedade e do uso exclusivo.



Figura 2.5 - Certificado de registro da marca Tizito.

2.3. Conceito

O principal objetivo do Tizito é facilitar a comunicação entre o anfitrião, aquele que marca o encontro, e seus convidados, aqueles que recebem o convite. Portanto, é um produto destinado ao consumidor final com foco na conectividade e interação entre os utilizadores; é um produto de mídia social que permite aos utilizadores criar perfis, gerir contactos, partilhar atualizações, publicar mídias e trocar mensagens; é um produto *online* que pode ser acedido por meio de dispositivos móveis, como *smartphones*. O Tizito é baseado em três pilares:

- **Convites:** oferecer maior objetividade na marcação de encontros pessoais entre amigos, reduzindo a troca de mensagens para esse fim;
- **Recordações:** permitir o armazenamento das fotos, vídeos e comentários de cada encontro, promovendo a preservação de momentos especiais;
- **Notificações:** manter o utilizador informado sobre as ações do anfitrião e/ou convidados, garantindo a comunicação de todos para todos.

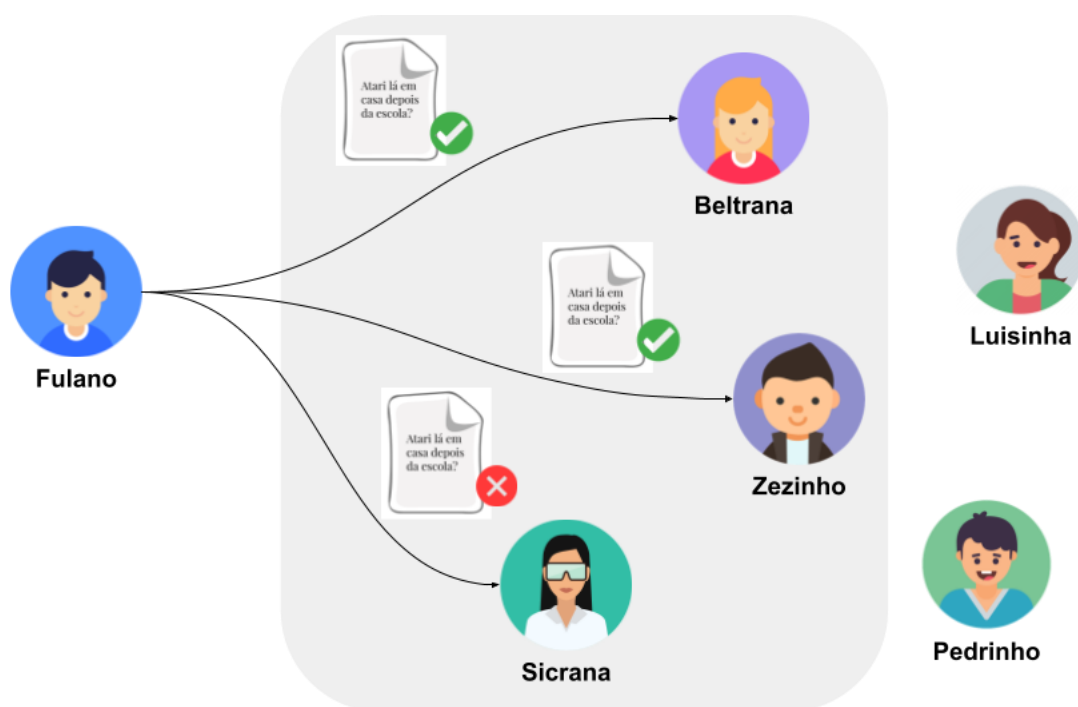


Figura 2.6 - Comunicação entre o anfitrião e seus convidados.

Para marcar um encontro, o anfitrião precisa decidir “o que” fazer, que pode ser um tema, *hobby*, desejo, comemoração ou cotidiano. Em seguida, ele precisa decidir “quando” e “onde” o encontro vai acontecer, assim como “quem” será convidado. Além disso, o anfitrião precisa decidir “qual” a visibilidade das fotos e vídeos com base no tipo de encontro:

- **Público:** tanto os amigos do anfitrião como os amigos de cada convidado podem visualizar as fotos e vídeos, bem como gostar e comentar o encontro;
- **Privado:** apenas o anfitrião e convidados podem visualizar as fotos e vídeos, bem como comentar o encontro.

Para cada encontro, o anfitrião pode publicar uma única foto e um único vídeo. Cada convidado que aceitou o convite também pode publicar uma única foto e um único vídeo. Essa abordagem visa promover um espírito de colaboração, no qual todos contribuem para o álbum do encontro. O Tizito oferece uma experiência completa ao utilizador, composta por ações que ocorrem antes, durante e depois do encontro:

- Um encontro pode ser marcado com até 30 minutos de antecedência. Metade desse prazo é destinada à moderação de possíveis imprevistos;
- Após marcar o encontro, não é permitido alterar a data e hora, nem remover convidados. No entanto, o anfitrião pode alterar o título e local do encontro, bem como adicionar novos convidados com até 15 minutos de antecedência;
- O anfitrião também pode desmarcar o encontro com até 15 minutos de antecedência. Esse prazo é para evitar que os convidados cheguem ao local do encontro sem saber que ele foi desmarcado;

- Cada convidado recebe um convite, que pode ser aceite ou recusado até à data e hora do encontro, pois considera-se perdoável que ele chegue ao local do encontro depois do horário marcado;
- Tanto o anfitrião como os convidados que aceitaram o convite podem efetuar publicações a partir de 15 minutos para o encontro.

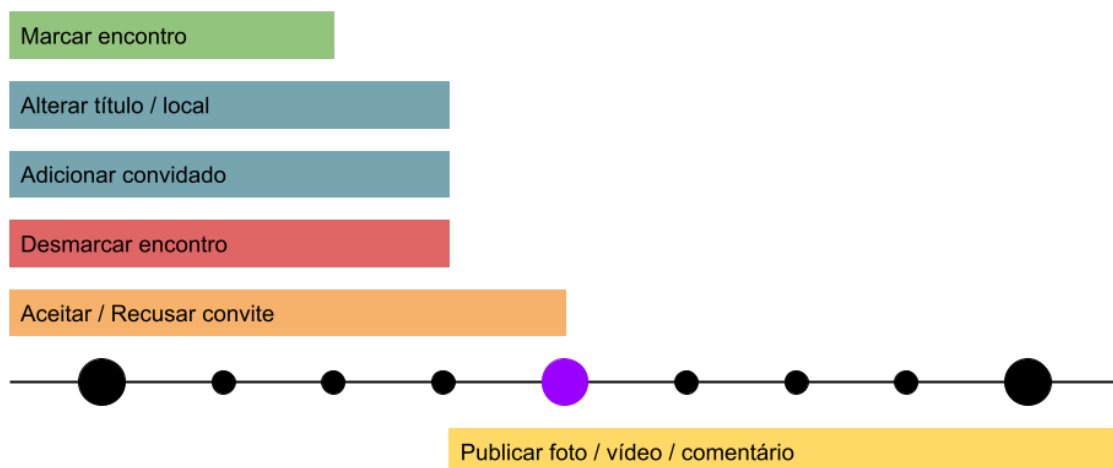


Figura 2.7 - Linha temporal.

Para cada ação, o Tizito notifica os respetivos utilizadores. Existem dois tipos de notificação:

- **Permanente:** é aquela que permanece acessível após a entrega. Ela é armazenada e o utilizador pode marcá-la como lida. Normalmente, ocorre antes do encontro. Por exemplo, quando o anfitrião desmarca o encontro;
- **Transitória:** é aquela que existe apenas durante a entrega. Ela é temporária e o utilizador pode visualizá-la apenas uma vez. Normalmente, ocorre depois do encontro. Por exemplo, quando o convidado publica uma foto.

Capítulo 3

Desenvolvimento do produto

Este capítulo apresenta os métodos, ferramentas e boas práticas utilizados nas etapas de definição dos requisitos, elaboração da *interface* e construção da aplicação.

3.1. Definição dos requisitos

3.1.1. 5W2H

5W2H é um método para responder perguntas determinantes sobre uma ação ou projeto (Zambelli, 2022). Dessa forma, é possível fazer um planeamento completo e eficiente, sempre com o intuito de garantir clareza e organização na execução de atividades. A sigla 5W2H nada mais é que as iniciais em inglês das seguintes perguntas:

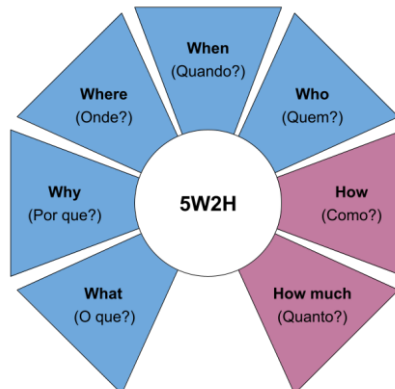


Figura 3.1 - Método 5W2H.

Esse método pode ser adotado em diferentes contextos, como na criação de novos produtos ou serviços, planeamento de projetos, gestão da qualidade, criação de planos de ação, implementação de mudanças organizacionais, gestão de crises, auditorias internas, produção e manutenção industrial.

5W2H	
What (O que será feito?)	Aplicação móvel para organizar os compromissos de uma agenda social.
Why (Por que será feito?)	Para oferecer maior objetividade na marcação de encontros pessoais entre amigos.
Where (Onde será feito?)	Será desenvolvido num portátil pessoal.
When (Quando será feito?)	Será desenvolvido entre os meses de julho e setembro de 2023.
Who (Quem será responsável?)	Uma única pessoa será responsável pela conceção da ideia, definição dos requisitos, elaboração da <i>interface</i> e construção da aplicação.
How (Como será feito?)	Métodos e boas práticas para iniciantes serão utilizados em cada etapa do desenvolvimento.
How much (Quanto vai custar?)	Ferramentas e bibliotecas gratuitas serão utilizadas para reduzir o custo de desenvolvimento, bem como serviços de infraestrutura que oferecem testes gratuitos e pagamento por utilização, com o intuito de minimizar tanto o custo de operação como o custo de armazenamento dos dados.

Figura 3.2 - Perguntas determinantes sobre o Tizito.

3.1.2. Histórias do utilizador

Histórias do utilizador são descrições simples e curtas de funcionalidades que serão implementadas no *software* (Gomes, 2013). Essas descrições são escritas sob o ponto de vista do utilizador e estruturadas no seguinte formato: “**Como um** [papel], **eu quero** [funcionalidade] **para que** [motivo].”

Como um anfitrião, eu quero marcar encontros para que eu possa reunir amigos, familiares, colegas de escola, universidade ou trabalho.
Como um convidado, eu quero aceitar ou recusar convites para que eu possa confirmar ou não minha presença em um encontro.

Figura 3.3 - Histórias do utilizador do Tizito.

A história de utilizador é apenas um lembrete de que mais detalhes serão necessários, e não deve ser considerada suficiente para a implementação da funcionalidade (Sabbagh, 2013). Esses detalhes são regras de negócio que podem ser anexadas às histórias do utilizador. A ferramenta Miro (<https://miro.com/pt/>) foi utilizada para listar as funcionalidades do Tizito.



Figura 3.4 - Funcionalidades do Tizito.

3.2. Elaboração da interface

3.2.1. Sketches

Sketches são uma forma rápida de rabiscar uma nova *interface* utilizando papel e caneta. Podem ser muito úteis para validar conceitos de produtos e ideias de *design* (Teixeira, 2017).

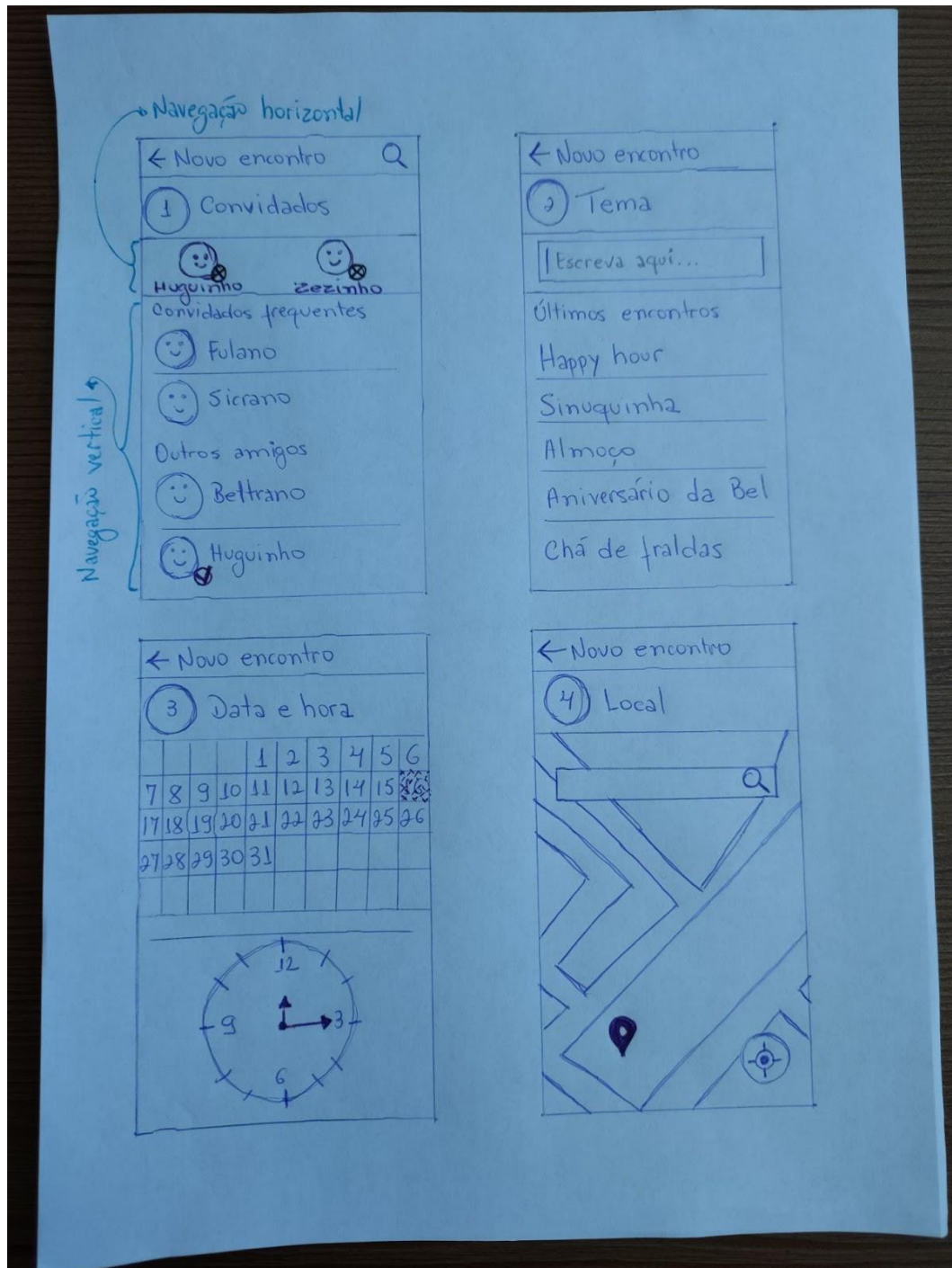


Figura 3.5 - Sketches do Tizito.

3.2.2. Wireframes

Wireframes são desenhos básicos da estrutura de determinada *interface*. Normalmente são feitos em tons de cinza, sem imagens, com o intuito de organizar os elementos que farão parte do *design* (Teixeira, 2017). A ferramenta Marvel App (<https://marvelapp.com/>) foi utilizada para criar os *wireframes*.

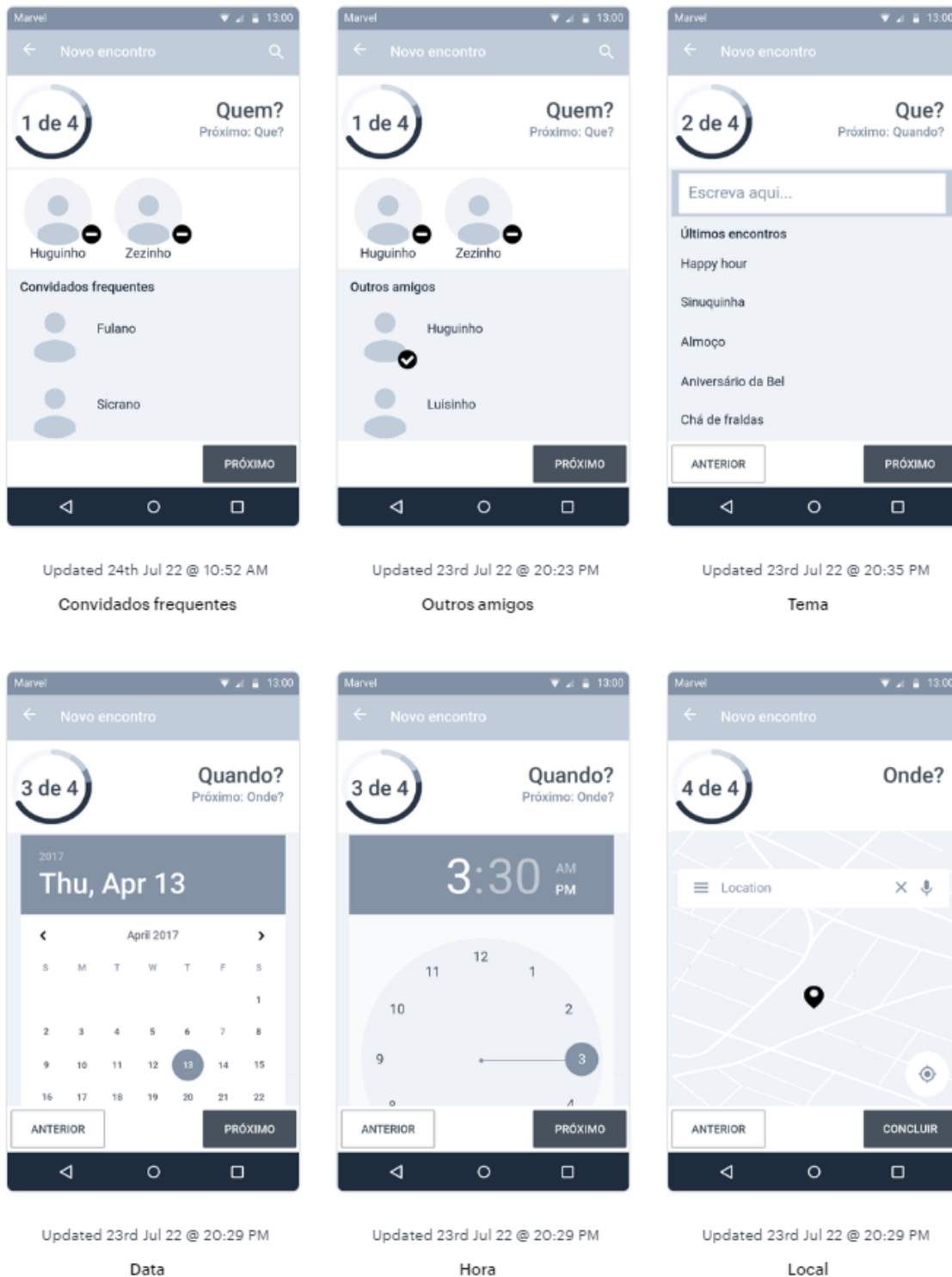


Figura 3.6 - Wireframes do Tizito.

3.2.3. Sitemaps

Sitemaps consistem em diagramas que organizam e mapeiam a estrutura de *sites* e aplicações para dispositivos móveis. Eles também ajudam na visualização da navegação e conexão entre *interfaces* (Teixeira, 2017).

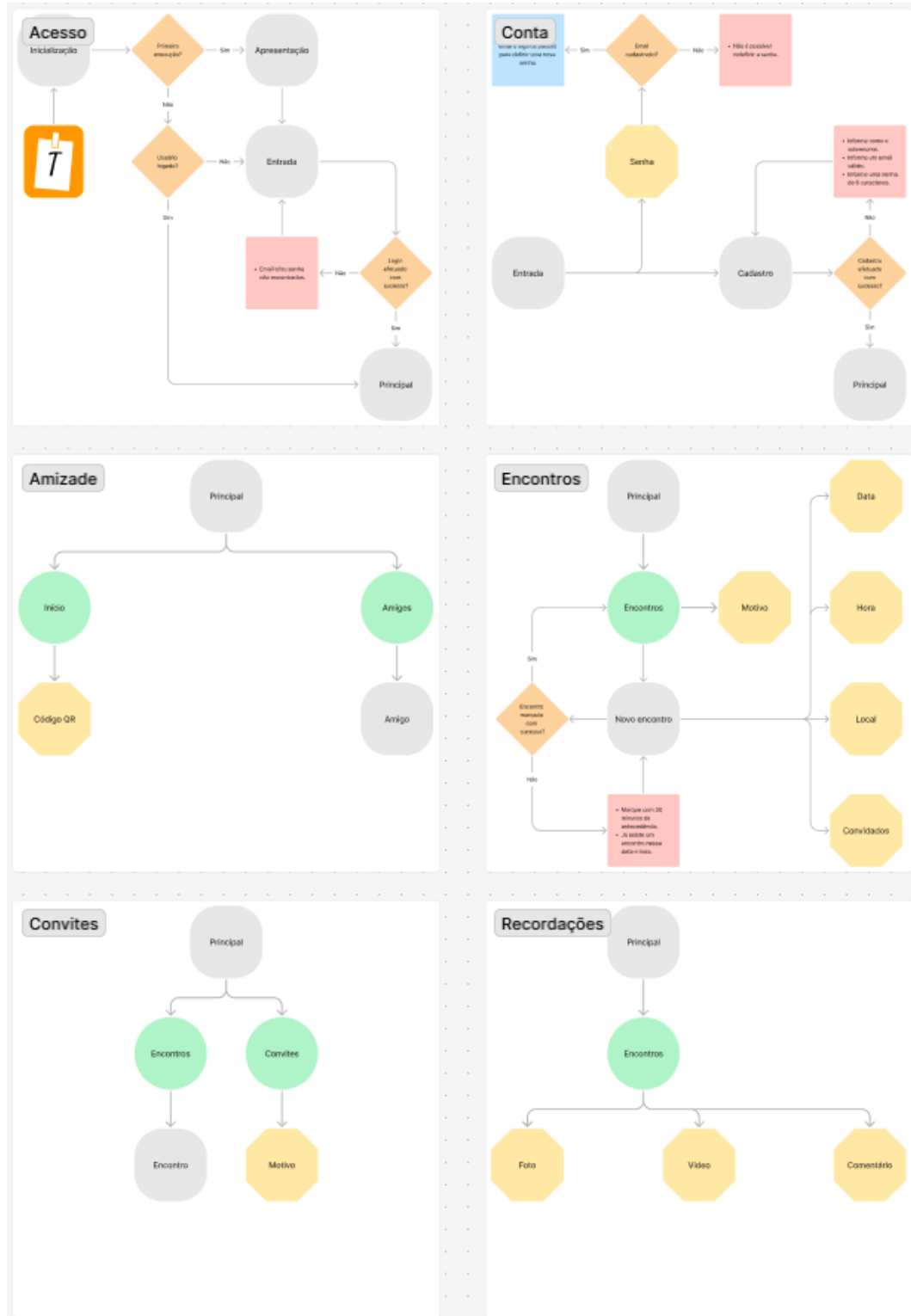


Figura 3.7 - Sitemaps do Tizito.

Este trabalho definiu sua própria notação para a estrutura do Tizito e utilizou a ferramenta FigJam (<https://www.figma.com/figjam/>) para criar os *sitemaps*.



Figura 3.8 - Estrutura do Tizito.

3.2.4. Referências

Algumas plataformas servem como fonte de inspiração e referências visuais, permitindo que *designers* e outros profissionais acompanhem tendências, explorem diferentes abordagens de *design* e identifiquem padrões de interação. As referências devem ser utilizadas como ponto de partida para criar algo novo (Lisboa, 2023). A ferramenta Dribbble (<https://dribbble.com/>) foi utilizada como fonte de referências para o Tizito.

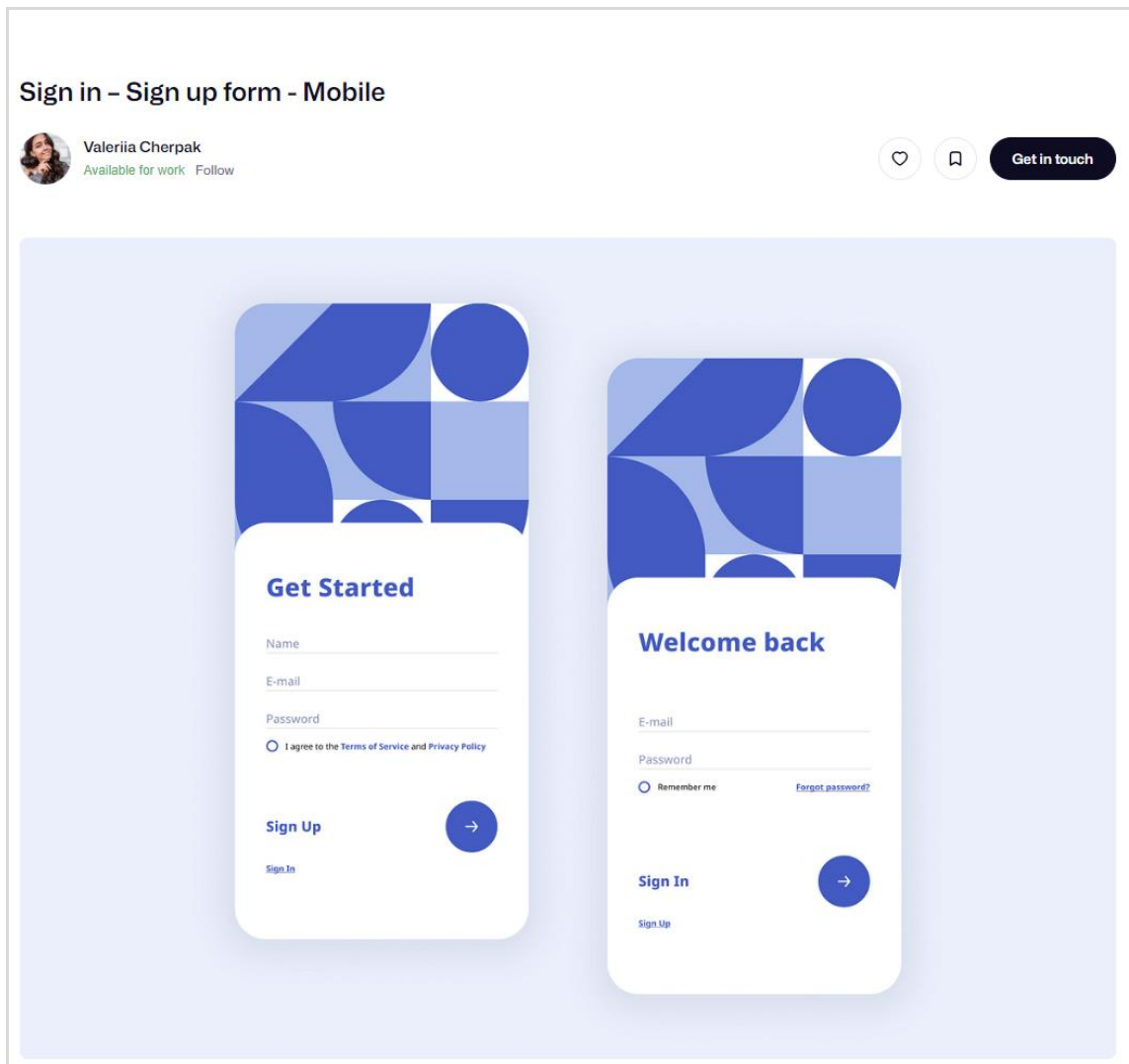


Figura 3.9 - Referências para o Tizito.

3.2.5. Padrões de design

Padrões de *design* são soluções recorrentes que resolvem problemas comuns de *interface* e melhoram a experiência do utilizador (Villain, 2023). Eles também fornecem uma linguagem comum que permite discussões mais eficientes sobre diferentes alternativas de *design*. A ferramenta *User Interface Design Patterns Library* (<https://ui-patterns.com/patterns>) foi utilizada para consulta de padrões.

User Interface Design Patterns		
Getting input	Navigation	Dealing with data
Forms Password Strength Meter Structured Format Keyboard Shortcuts Rule Builder Captcha Inplace Editor Preview Drag and drop Fill in the Blanks WYSIWYG Input Feedback Calendar Picker Morphing Controls Expandable Input Forgiving Format Settings Undo Input Prompt Good Defaults Autosave	Tabs Navigation Tabs Module Tabs Jumping in hierarchy Shortcut Dropdown Breadcrumbs Fat Footer Notifications Modal Home Link Menus Vertical Dropdown Menu Accordion Menu Horizontal Dropdown Menu Content Adaptable View Article List Favorites Cards Tag Cloud Carousel Progressive Disclosure Continuous Scrolling Event Calendar Archive Pagination Tagging	Tables Table Filter Alternating Row Colors Sort By Column Formatting data Frequently Asked Questions (FAQ) Dashboard Copy Box Images Gallery Slideshow Image Zoom Search Autocomplete Search Filters
Explaining the process Wizard Completeness meter Steps Left Inline Help Box		Onboarding
Community driven		Guidance Coachmarks Playthrough Guided Tour Inline Hints Walkthrough Blank Slate

Figura 3.10 - Padrões de *design*.

3.2.6. Bibliotecas de padrões

Biblioteca de padrões é uma lista prática que contém exemplos, e às vezes código, dos padrões de interação e principais elementos interativos que compõem as *interfaces* de um determinado produto (Teixeira, 2017). O Tizito utiliza o Material Design (<https://m3.material.io/>) como biblioteca de padrões.

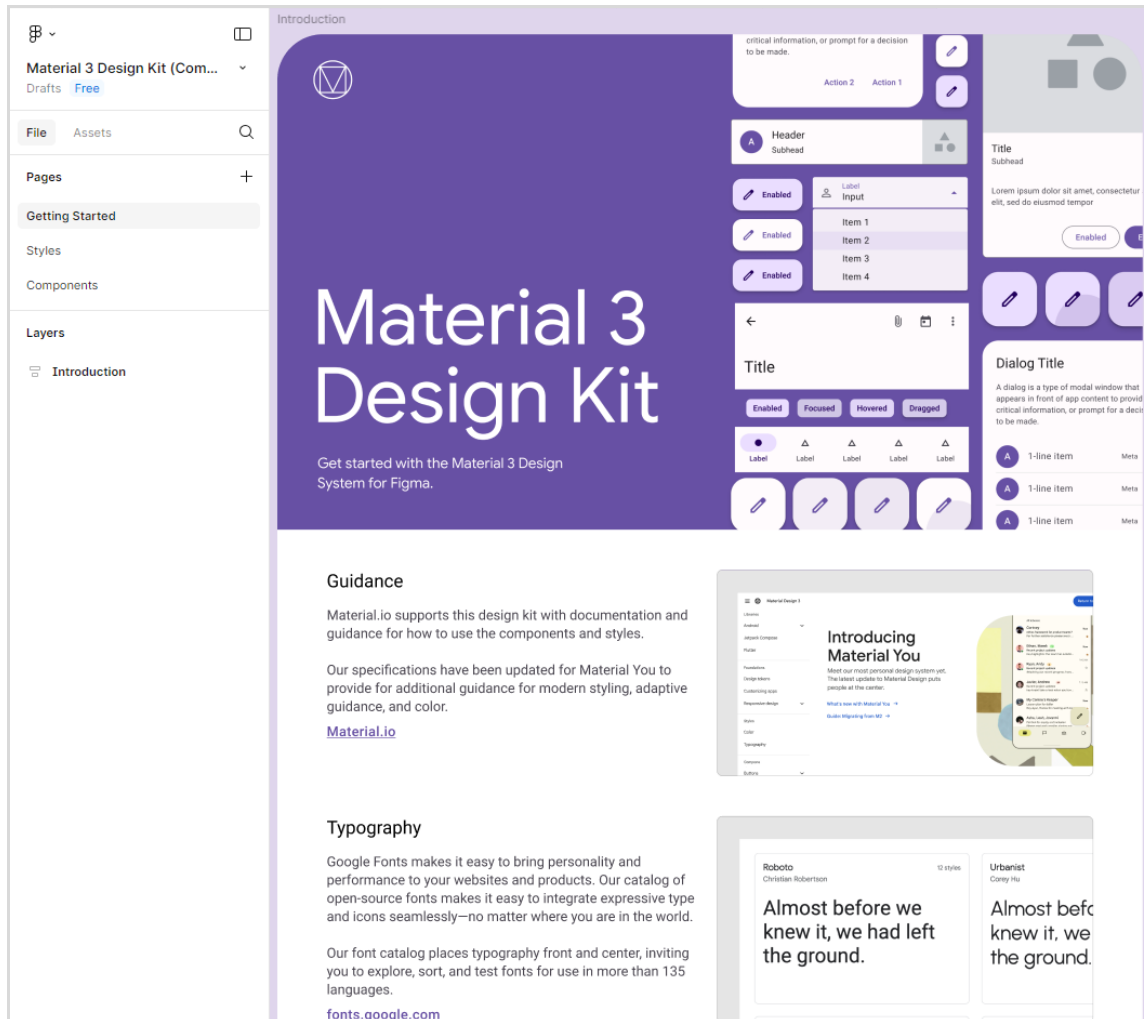


Figura 3.11 - Biblioteca de padrões.

3.2.7. Imagens

A ferramenta Freepik (<https://www.freepik.com/>) foi utilizada como banco de imagens para criar o protótipo do Tizito. Ela oferece fotografias, ilustrações e imagens vetoriais gratuitas para designers.

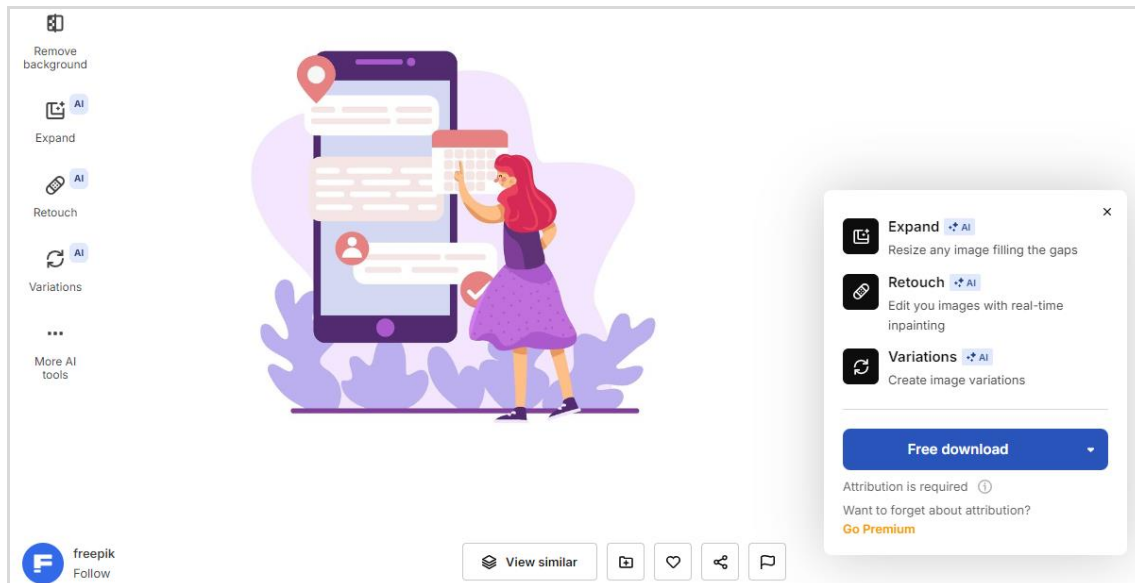


Figura 3.12 - Banco de imagens.

Pequenas modificações nas imagens também foram necessárias. Para isso, a ferramenta iLoveIMG (<https://www.iloveimg.com/>) foi utilizada de forma gratuita para cortar, redimensionar, comprimir e converter imagens.

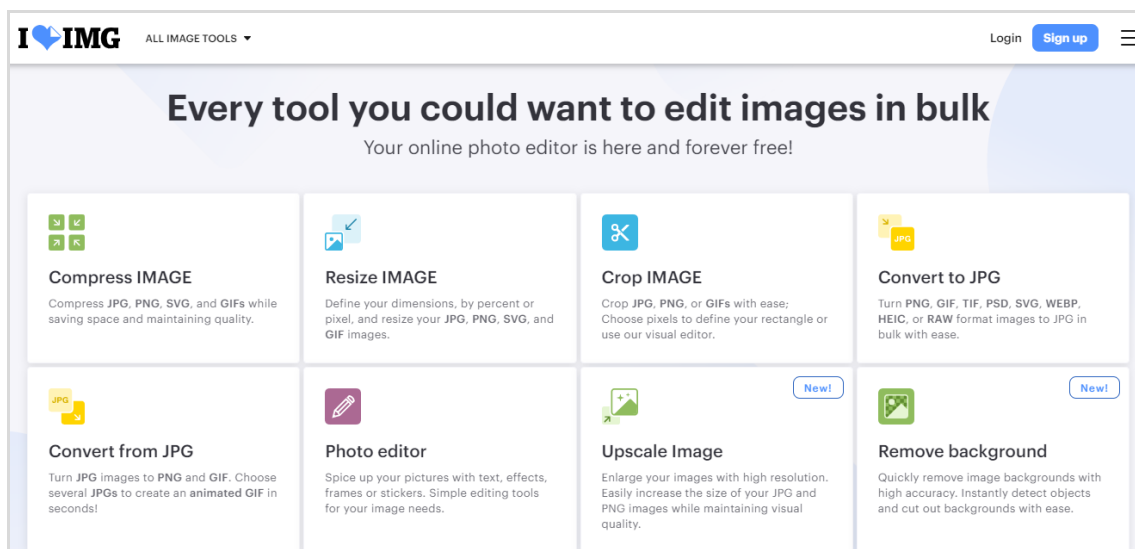


Figura 3.13 - Modificação de imagens.

3.2.8. Protótipo

Protótipo é uma simulação da navegação e funcionalidades de um *software*; é uma forma rápida de validar e testar um produto antes de construí-lo do começo ao fim (Teixeira, 2017). A ferramenta Figma (<https://www.figma.com/>) foi utilizada para criar o protótipo do Tizito.

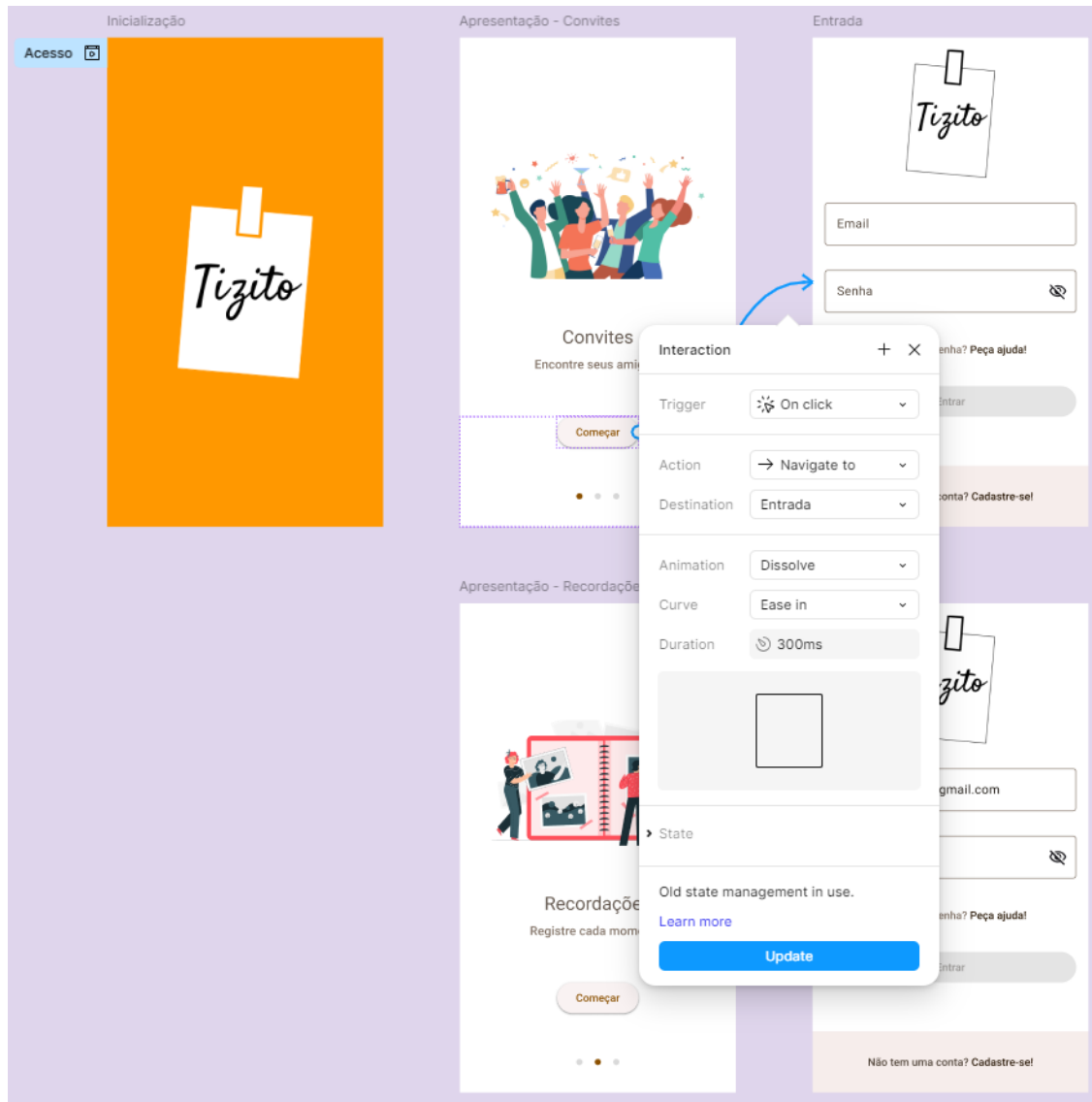


Figura 3.14 - Protótipo do Tizito.

3.3. Construção da aplicação

3.3.1. Tecnologia

Dart (<https://dart.dev/>) é uma linguagem de programação criada e mantida pelo Google. Sua primeira especificação foi aberta ao público em outubro de 2011. Após muito trabalho da equipa responsável, ela foi apresentada ao mundo em junho de 2012 pelos fundadores do projeto, Lars Bak e Kasper Lund, na conferência anual Google I/O. No entanto, foi somente em outubro de 2013 que a primeira versão estável do SDK (*Software Development Kit*) foi lançada. Em 2018, ocorreram os maiores avanços e crescimento de todo o ecossistema Dart, devido à popularidade do Flutter (<https://flutter.dev/>), framework de código aberto que utiliza a linguagem Dart (Bitencourt, 2022).

Desenvolvido pelo Google, o Flutter permite a construção de aplicações para Android e iOS a partir de um único código-fonte, proporcionando praticamente todo o desempenho de uma aplicação nativa. No Flutter, tudo é organizado como *widget*, ou seja, um componente com propriedades e eventos. Basicamente, os *widgets* são interpretados de tal forma que se tornam componentes nativos para Android e iOS assim que o código é executado. Portanto, o produto final não é algo em outra linguagem se comunicando com a plataforma nativa por meio de uma ponte, mas sim, um código nativo de alto desempenho. Existem dois tipos de *widgets*: básicos e personalizados. *Widgets* básicos são aqueles que vêm por padrão no Flutter SDK (botões, caixas de texto, espaçamento, *ícones*, listas, etc). *Widgets* personalizados são criados ou importados de acordo com a necessidade da aplicação (Marinho, 2021).

3.3.2. Arquitetura

Arquitetura de *software* refere-se à organização básica de um *software*. Seu principal objetivo é facilitar a compreensão, desenvolvimento, manutenção, expansão e implantação do *software*. O ideal é escolher uma arquitetura que mais se aproxime das necessidades do *software*. A partir disso, é possível decidir sobre a estrutura geral, a divisão em módulos, a comunicação entre eles e a padronização do processo de desenvolvimento (Rodrigues, 2023). A estrutura sobre a qual o Tizito foi construído é descrita a seguir.

MODELS

Este módulo agrupa as classes que representam entidades e objetos de valor. No Tizito, o modelo é centralizado nas classes *Meeting* e *User*. As demais abstrações devem indicar a que encontro estão associadas e qual o utilizador responsável. Por exemplo, todo comentário é publicado por um utilizador no contexto de um determinado encontro. O mesmo pode ser dito em relação às fotos, vídeos, mensagens e gostos.

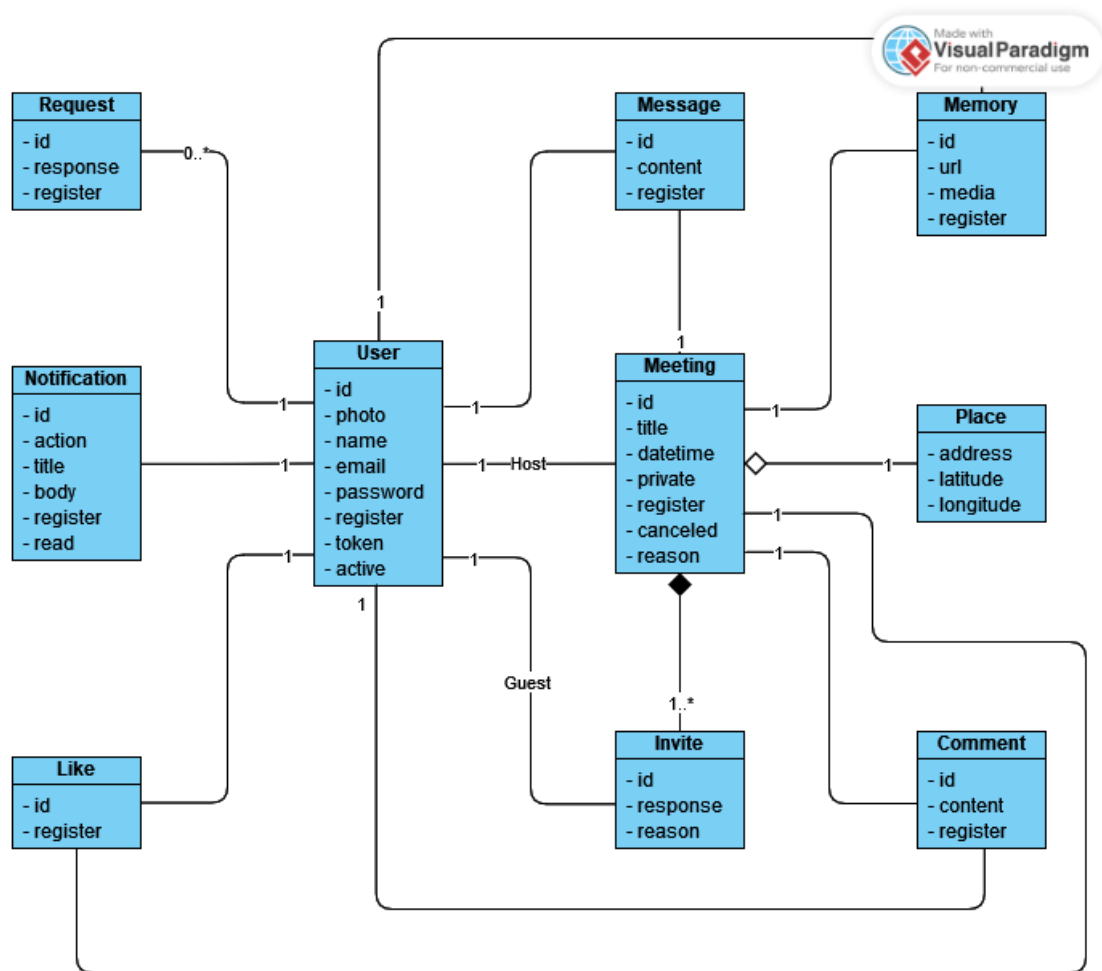
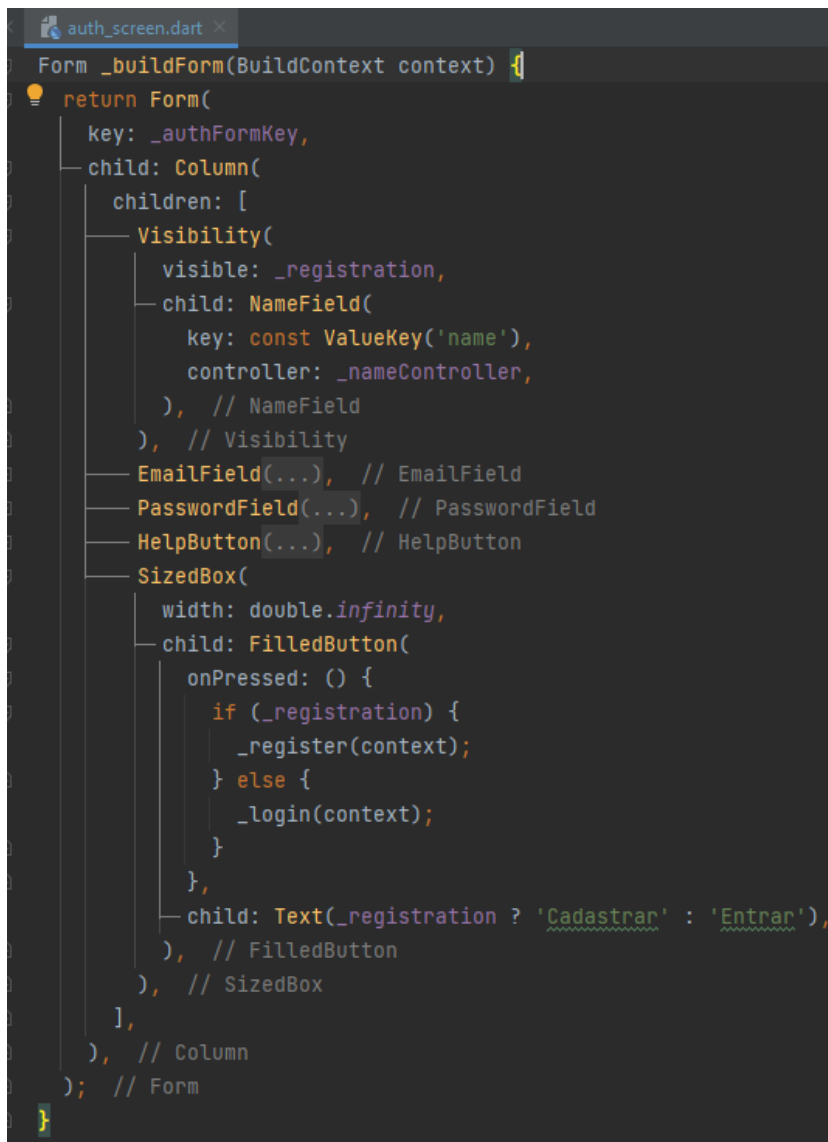


Figura 3.15 - Diagrama de classes do Tizito.

Para exemplificar, um objeto da classe *Meeting* é uma entidade, pois tem uma identidade única. Isso significa que nenhum encontro será igual a outro no Tizito. Já um objeto da classe *Place* é um objeto de valor, pois é caracterizado apenas por seu estado. Isso significa que dois locais serão idênticos se tiverem exatamente os mesmos valores para latitude e longitude.

SCREENS

Este módulo agrupa as classes que representam ecrãs, listas e diálogos. Além de *widgets* personalizados, elas são compostas por *widgets* básicos fornecidos pelo Flutter, cujo catálogo (<https://docs.flutter.dev/ui/widgets>) é dividido entre duas bibliotecas de padrões: Cupertino (*widgets* que se alinham com as Diretrizes de Interface Humana da Apple) e Material (*widgets* que implementam a especificação de design do Material 3). Como mencionado anteriormente, o Tizito utiliza o Material Design.



```

auth_screen.dart
Form _buildForm(BuildContext context) {
  return Form(
    key: _authFormKey,
    child: Column(
      children: [
        Visibility(
          visible: _registration,
          child: NameField(
            key: const ValueKey('name'),
            controller: _nameController,
          ), // NameField
        ), // Visibility
        EmailField(...), // EmailField
        PasswordField(...), // PasswordField
        HelpButton(...), // HelpButton
        SizedBox(
          width: double.infinity,
          child: FilledButton(
            onPressed: () {
              if (_registration) {
                _register(context);
              } else {
                _login(context);
              }
            },
            child: Text(_registration ? 'Cadastrar' : 'Entrar'),
          ), // FilledButton
        ), // SizedBox
      ],
    ), // Column
  ); // Form
}

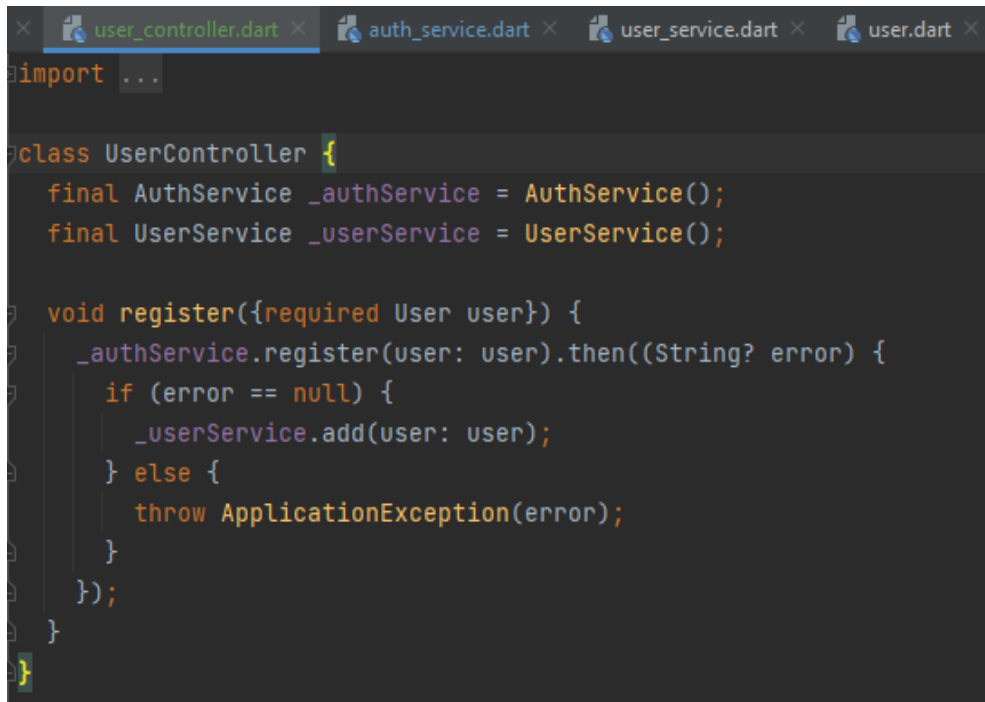
```

Figura 3.16 - Classe que representa um ecrã.

Para exemplificar, a classe *AuthScreen* representa o ecrã de entrada do Tizito. De acordo com a navegação do utilizador, ela também representa o ecrã de registo. Ela é composta por *widgets* básicos (*Form*, *Column*, *Visibility*, *SizedBox*, *FilledButton* e *Text*) e personalizados (*NameField*, *EmailField*, *PasswordField* e *HelpButton*).

CONTROLLERS

Este módulo agrupa as classes que representam controladores da aplicação. Por outras palavras, seus métodos encapsulam as funcionalidades e comportamento da aplicação. Por exemplo, quando um utilizador pressiona um botão, a *interface* pede ao controlador para executar a funcionalidade. Isso significa que ele verifica regras de negócio, persiste dados, envia e-mails, armazena ficheiros, etc.



```

import ...

class UserController {
  final AuthService _authService = AuthService();
  final UserService _userService = UserService();

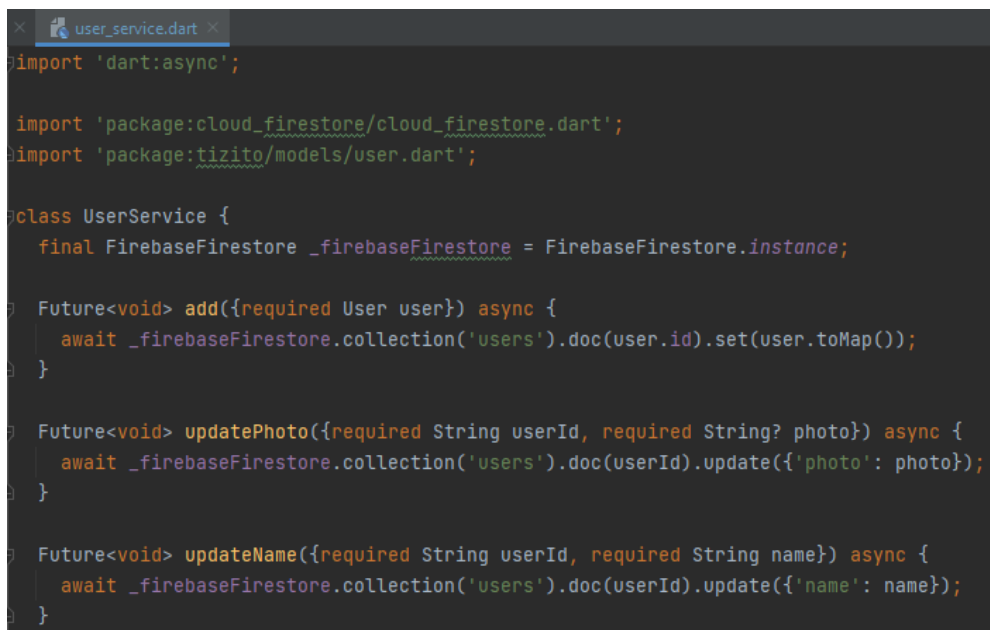
  void register({required User user}) {
    _authService.register(user: user).then((String? error) {
      if (error == null) {
        _userService.add(user: user);
      } else {
        throw ApplicationException(error);
      }
    });
  }
}

```

Figura 3.17 - Classe que representa um controlador.

SERVICES

Este módulo agrupa as classes que representam serviços de infraestrutura. Por exemplo, persistência de dados, envio de e-mails, armazenamento de arquivos, etc. Esses serviços podem ser internos, como persistência de dados no próprio dispositivo móvel, ou externos, como persistência de dados num servidor remoto.



```

import 'dart:async';

import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:tizito/models/user.dart';

class UserService {
  final FirebaseFirestore _firebaseFirestore = FirebaseFirestore.instance;

  Future<void> add({required User user}) async {
    await _firebaseFirestore.collection('users').doc(user.id).set(user.toMap());
  }

  Future<void> updatePhoto({required String userId, required String? photo}) async {
    await _firebaseFirestore.collection('users').doc(userId).update({'photo': photo});
  }

  Future<void> updateName({required String userId, required String name}) async {
    await _firebaseFirestore.collection('users').doc(userId).update({'name': name});
  }
}

```

Figura 3.18 - Classe que representa um serviço.

HELPERS

Este módulo agrupa as classes que representam tarefas específicas. Elas geralmente encapsulam bibliotecas de terceiros. Isso torna a manutenção mais fácil e até reduz o impacto na aplicação se uma biblioteca for substituída por outra. Para exemplificar, a classe *Encrypter* é chamada para criptografar ou descriptografar uma sequência de caracteres. Internamente, ela encapsula a biblioteca *encrypt* (<https://pub.dev/packages/encrypt>).



```
import 'package:encrypt/encrypt.dart' as lib;

class Encrypter {
  late lib.IV _iv;
  late lib.Encrypter _encrypter;

  Encrypter() {
    _iv = lib.IV.fromLength(16);

    lib.Key key = lib.Key.fromUtf8('KEY');

    _encrypter = lib.Encrypter(lib.AES(key));
  }

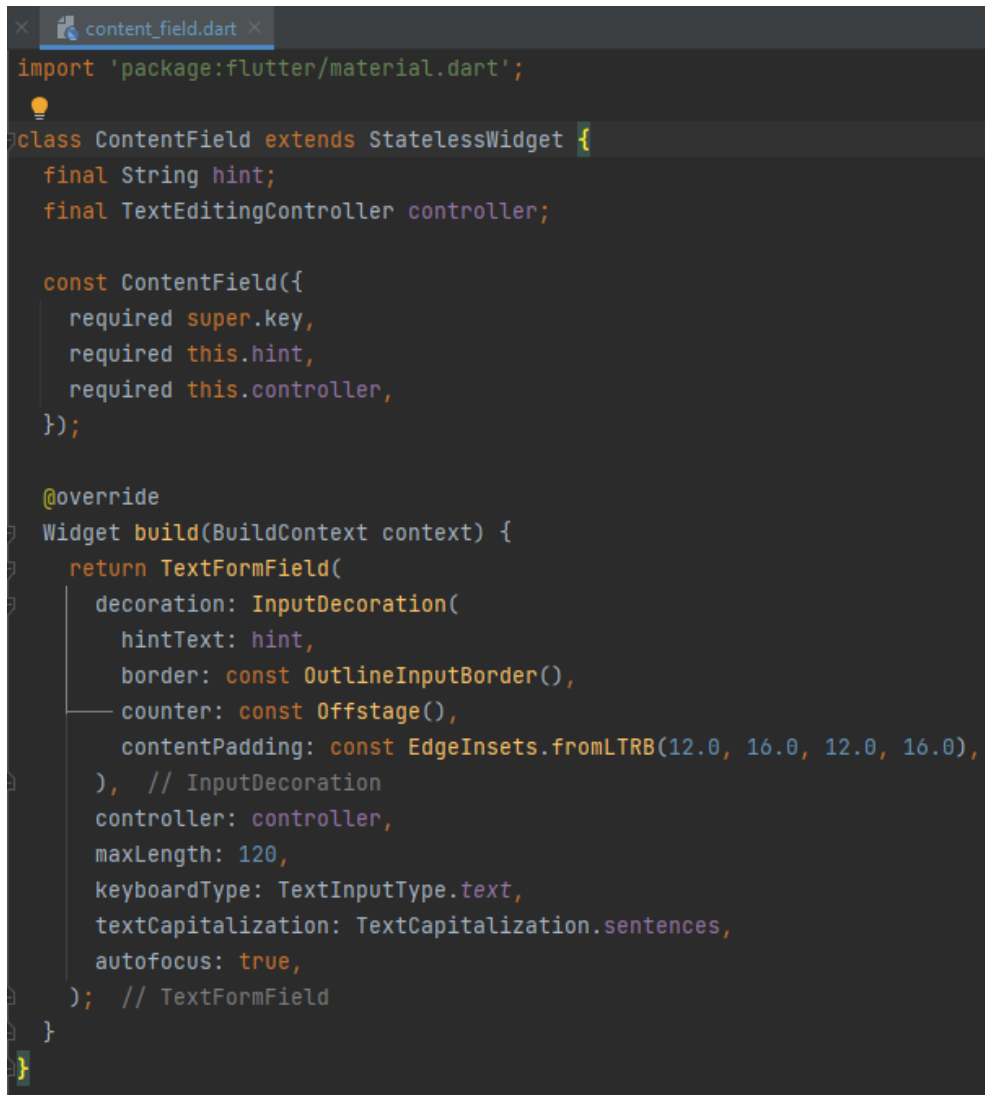
  String encrypt(String value) {
    return _encrypter.encrypt(value, iv: _iv).base64;
  }

  String decrypt(String value) {
    return _encrypter.decrypt64(value, iv: _iv);
  }
}
```

Figura 3.19 - Classe que representa uma tarefa específica.

COMPONENTS

Este módulo agrupa as classes que representam *widgets* personalizados. Desse modo, é possível utilizar um mesmo *widget* em vários ecrãs, listas e diálogos. Para exemplificar, um objeto da classe *ContentField* gera uma caixa de texto que permite uma sequência de até 120 caracteres. De acordo com o diagrama de classes do Tizito, tanto a classe *Message* como a classe *Comment* possuem um atributo chamado “*content*”. Isso significa que esse mesmo *widget* pode ser utilizado para enviar uma mensagem ou publicar um comentário.



```

import 'package:flutter/material.dart';

class ContentField extends StatelessWidget {
  final String hint;
  final TextEditingController controller;

  const ContentField({
    required super.key,
    required this.hint,
    required this.controller,
  });

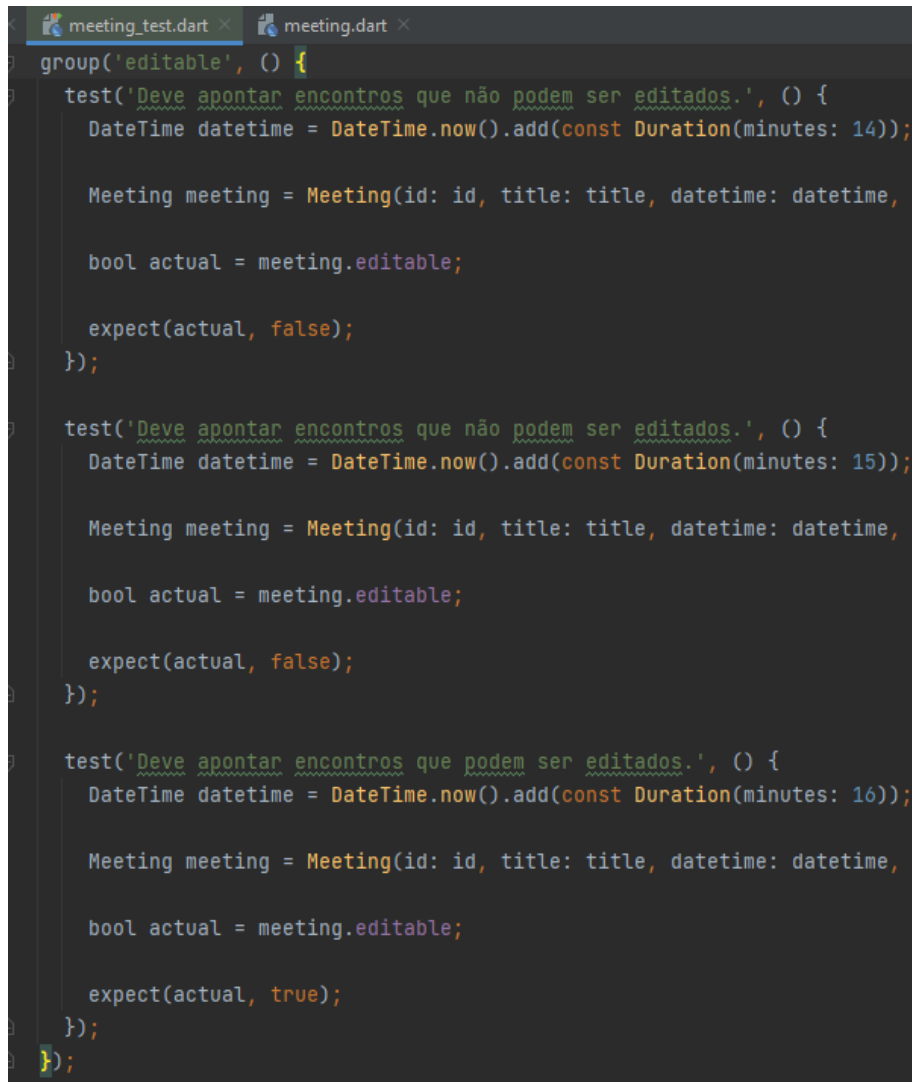
  @override
  Widget build(BuildContext context) {
    return TextFormField(
      decoration: InputDecoration(
        hintText: hint,
        border: const OutlineInputBorder(),
        counter: const Offstage(),
        contentPadding: const EdgeInsets.fromLTRB(12.0, 16.0, 12.0, 16.0),
      ), // InputDecoration
      controller: controller,
      maxLength: 120,
      keyboardType: TextInputType.text,
      textCapitalization: TextCapitalization.sentences,
      autofocus: true,
    ); // TextFormField
  }
}

```

Figura 3.20 - Classe que representa um *widget* personalizado.

TESTS

Uma boa prática na construção de aplicações é o TDD (*Test Driven Development*). Essa prática não somente garante a qualidade do *software* em desenvolvimento, mas também aumenta a segurança para futuras modificações nesse *software* (Torres, 2017). Assim, o Tizito tem um módulo que agrupa testes de unidade para as classes que representam entidades e objetos de valor.



```

group('editable', () {
  test('Deve apontar encontros que não podem ser editados.', () {
    DateTime datetime = DateTime.now().add(const Duration(minutes: 14));

    Meeting meeting = Meeting(id: id, title: title, datetime: datetime,

    bool actual = meeting.editable;

    expect(actual, false);
  });

  test('Deve apontar encontros que não podem ser editados.', () {
    DateTime datetime = DateTime.now().add(const Duration(minutes: 15));

    Meeting meeting = Meeting(id: id, title: title, datetime: datetime,

    bool actual = meeting.editable;

    expect(actual, false);
  });

  test('Deve apontar encontros que podem ser editados.', () {
    DateTime datetime = DateTime.now().add(const Duration(minutes: 16));

    Meeting meeting = Meeting(id: id, title: title, datetime: datetime,

    bool actual = meeting.editable;

    expect(actual, true);
  });
});

```

Figura 3.21 - Classe que representa um teste de unidade.

3.3.3. Infraestrutura

Existem serviços de infraestrutura que foram criados para facilitar a construção de aplicações, permitindo que o foco seja direcionado para a *interface*, funcionalidades e regras de negócio (Ribeiro, 2023). Devido à sua fácil integração com o Flutter e aos muitos serviços oferecidos, a ferramenta Firebase (<https://firebase.google.com/>) foi utilizada pelo Tizito.

FIREBASE AUTHENTICATION

Firebase Authentication é um serviço que facilita a implementação de autenticação em aplicações. Ele oferece suporte à autenticação por e-mail, palavra-passe e número de telefone, além de permitir o uso de contas Google, Facebook, etc. Ele também fornece ferramentas para gerenciar utilizadores, incluindo redefinição de palavra-passe e verificação de e-mail.

FIREBASE CLOUD FIRESTORE

Firebase Cloud Firestore é uma base de dados NoSQL, utilizada para armazenar e sincronizar dados com as aplicações, sejam elas Android, iOS ou web. Além disso, ela tem recursos como atualização em tempo real, suporte *offline*, consulta expressiva e escalabilidade.

FIREBASE STORAGE

Firebase Storage é um serviço de armazenamento, gerenciamento e partilha de ficheiros em nuvem. Ele suporta vários formatos (texto, áudio, vídeo, documentos, imagens, etc) e permite o armazenamento ilimitado de ficheiros em qualquer tamanho.

FIREBASE CLOUD FUNCTIONS

Firebase Cloud Functions é um serviço que permite executar código *backend* de forma automática em resposta a eventos disparados por funcionalidades da própria aplicação ou de outros serviços do Firebase. As funções são escritas em JavaScript ou TypeScript, armazenadas e executadas num ambiente isolado, seguro e escalável, sem a necessidade de gerenciar servidores.

FIREBASE CLOUD MESSAGING

Firebase Cloud Messaging é um serviço de envio de notificações *push* para aplicações móveis. Ele também possibilita o envio de notificações em *background*, que são notificações *push* enviadas para um dispositivo sem que a aplicação esteja aberta no momento. Normalmente, elas aparecem como uma notificação na barra de *status* do dispositivo. Desse modo, o utilizador pode abrir a aplicação pressionando a notificação ou removê-la, deslizando para o lado.



Figura 3.22 - Painel do Firebase.

Capítulo 4

Tizito

Este capítulo apresenta as funcionalidades e formas de interação entre o utilizador e o produto.

4.1. Acesso

Na primeira execução, o ecrã de apresentação das principais funcionalidades da aplicação permanece visível até o utilizador prosseguir para o ecrã de entrada. Nas execuções subsequentes, o utilizador é direcionado para o ecrã principal do Tizito, se o acesso anterior ainda estiver ativo; caso contrário, é necessário informar novamente as suas credenciais.

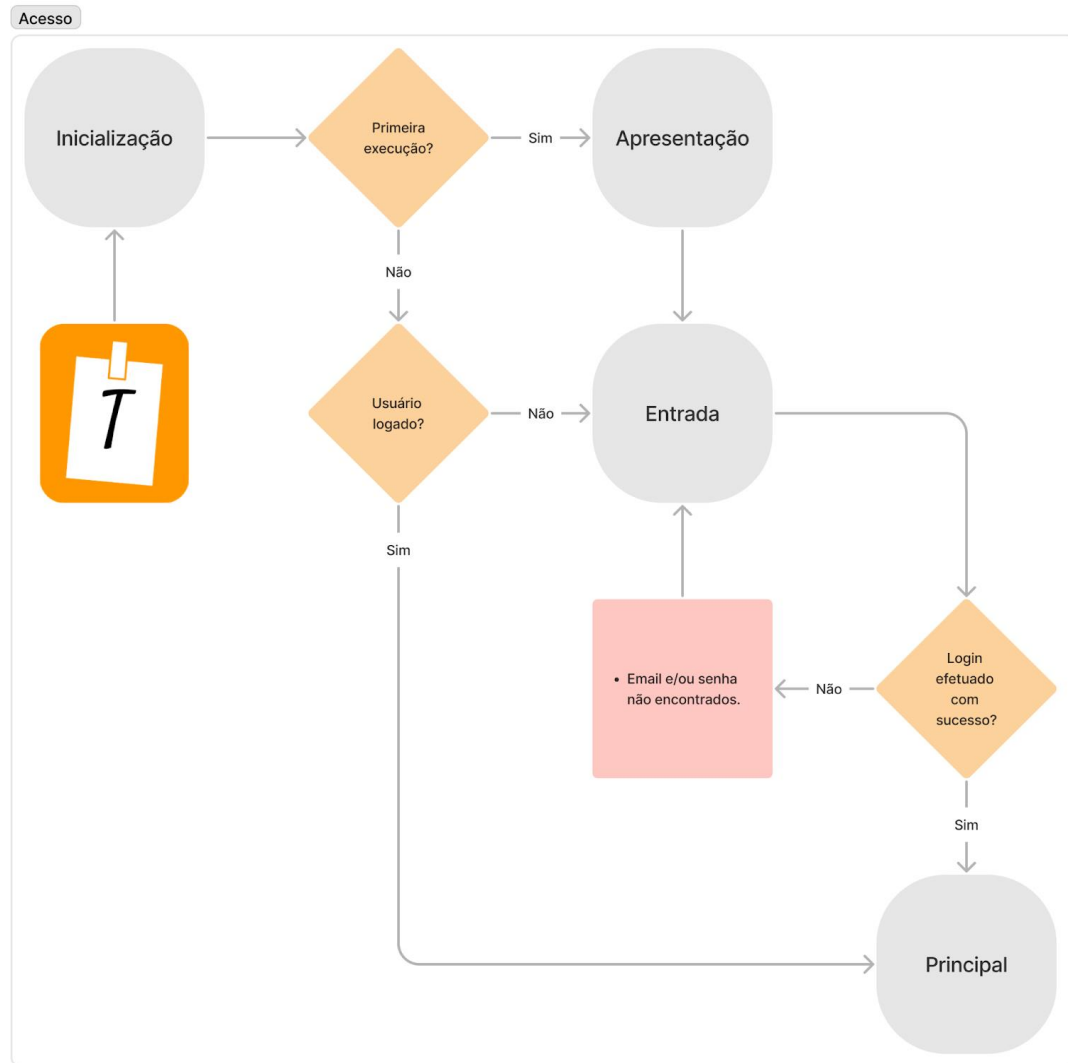


Figura 4.1 - Sitemap do acesso.

4.1.1. Inicialização

Ao pressionar o ícone do Tizito, o ecrã de inicialização permanece visível enquanto a aplicação está a carregar. A inicialização tem como finalidade atenuar a sensação de uma longa espera. Existem dois tipos de ecrãs de inicialização: *Placeholder UI*, que proporciona uma visualização não interativa da interface real da aplicação; *Branded*, que proporciona uma exposição momentânea à marca da aplicação. Esse último foi escolhido com o intuito de reforçar o reconhecimento da marca.



Figura 4.2 - Ecrã de inicialização.

4.1.2. Apresentação

As principais funcionalidades da aplicação são apresentadas por meio de um carrossel, a mostrar como o Tizito pode ser utilizado no dia a dia:

- **Convites:** Encontre os seus amigos.
- **Recordações:** Registe cada momento.
- **Notificações:** Fique a par de tudo.

Ao pressionar o botão "Iniciar", o utilizador prossegue para o ecrã de entrada da aplicação.

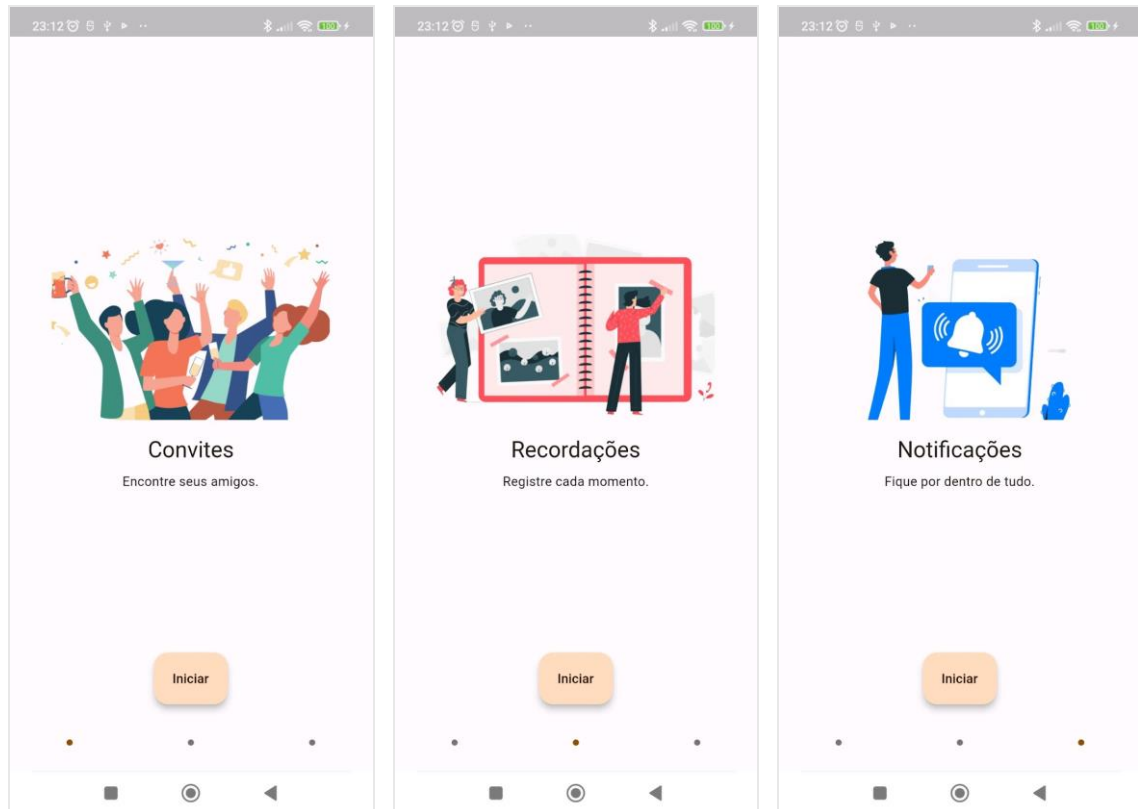


Figura 4.3 - Ecrã de apresentação.

4.1.3. Entrada

No ecrã de entrada, é necessário informar as credenciais do utilizador (e-mail e palavra-passe) para aceder ao ecrã principal do Tizito. Se elas não forem encontradas, uma mensagem de erro é apresentada.

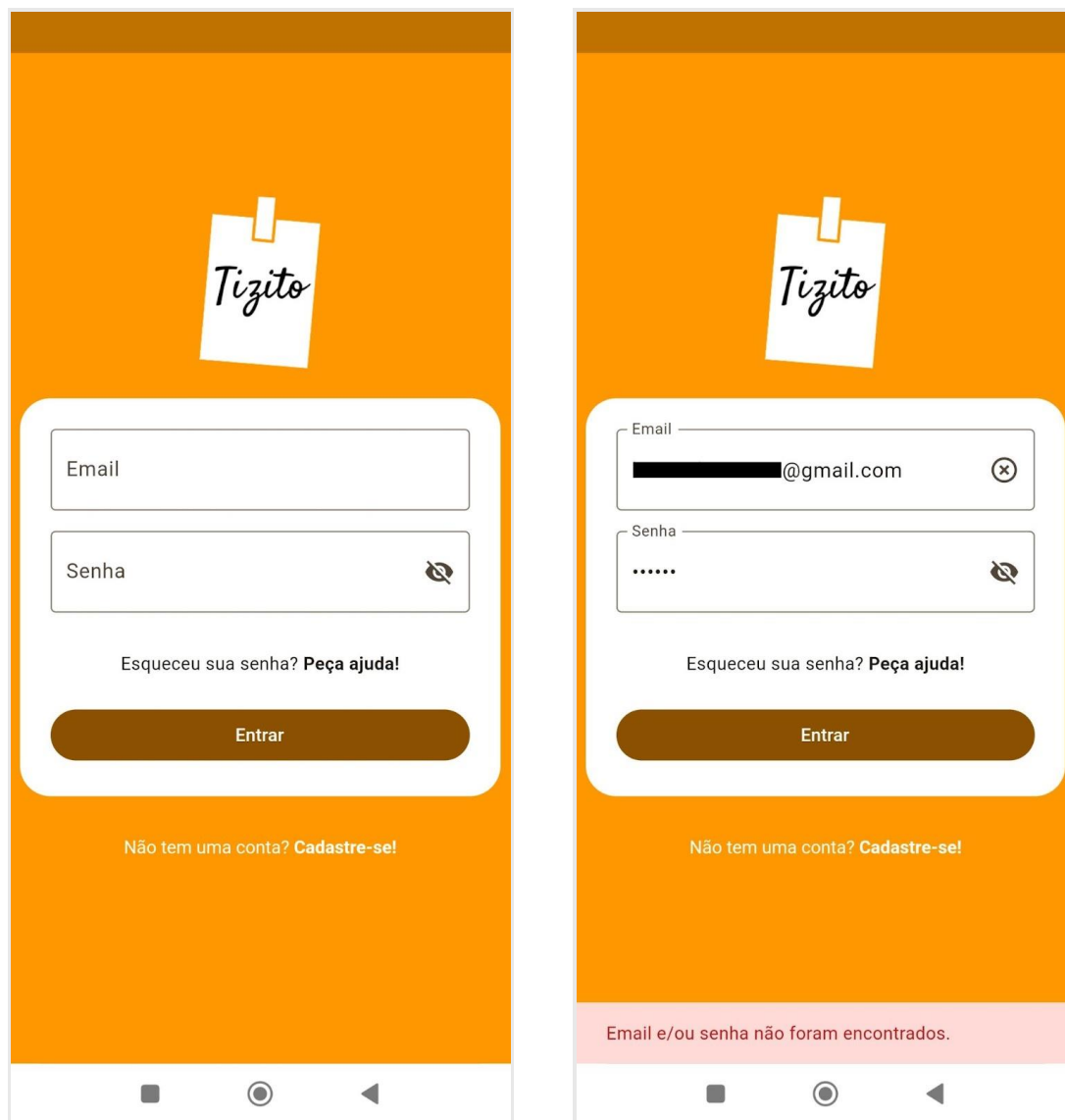


Figura 4.4 - Ecrã de entrada.

4.2. Conta

Para utilizar a aplicação, é necessário que o utilizador crie uma conta. Assim, é possível marcar encontros e receber convites, bem como identificar o responsável pela publicação de uma foto, vídeo ou comentário.

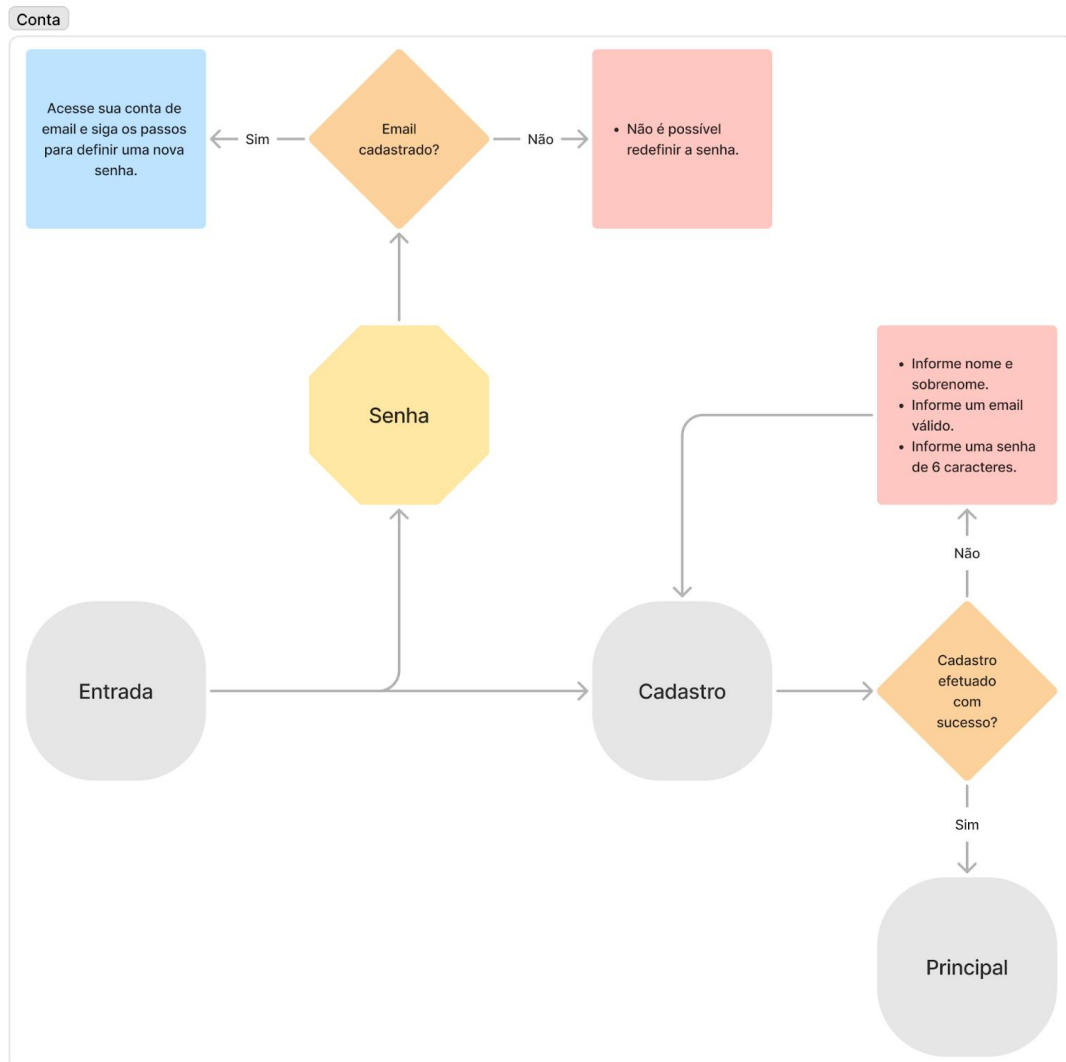


Figura 4.5 - Sitemap da conta.

4.2.1. Registo

Para criar uma conta, o utilizador deve informar o seu nome, *e-mail* e palavra-passe. Ao pressionar o botão “Cadastrar”, as seguintes regras de negócio são verificadas:

- O *e-mail* já está a ser utilizado;
- O *e-mail* não é válido;
- A palavra-passe não é forte o suficiente.

Essas regras são importantes para a consistência dos dados. Após efetuar o registo, o utilizador é direcionado para o ecrã principal do Tizito.

The image displays two versions of the Tizito registration screen. The left version is the initial state, showing three input fields: 'Nome', 'Email', and 'Senha' (with a toggle icon). Below the fields is a link to 'Termos de uso e Política de privacidade' and a 'Cadastrar' button. The right version shows the same screen after validation errors. The 'Nome' field has the message 'Informe um nome.', the 'Email' field has 'Informe um email válido.', and the 'Senha' field has 'Informe uma senha de 6 caracteres.'.

Figura 4.6 - Ecrã de registo.

4.2.2. Palavra-passe

No cenário em que o utilizador se esqueceu da palavra-passe, a aplicação oferece uma opção de redefinição a partir do ecrã de entrada. Para isso, o utilizador deve informar o *e-mail* registado e pressionar o botão “Redefinir”. Em seguida, deve aceder à sua conta de *e-mail*, abrir o *link* presente na mensagem enviada pelo Tizito e informar uma nova palavra-passe.

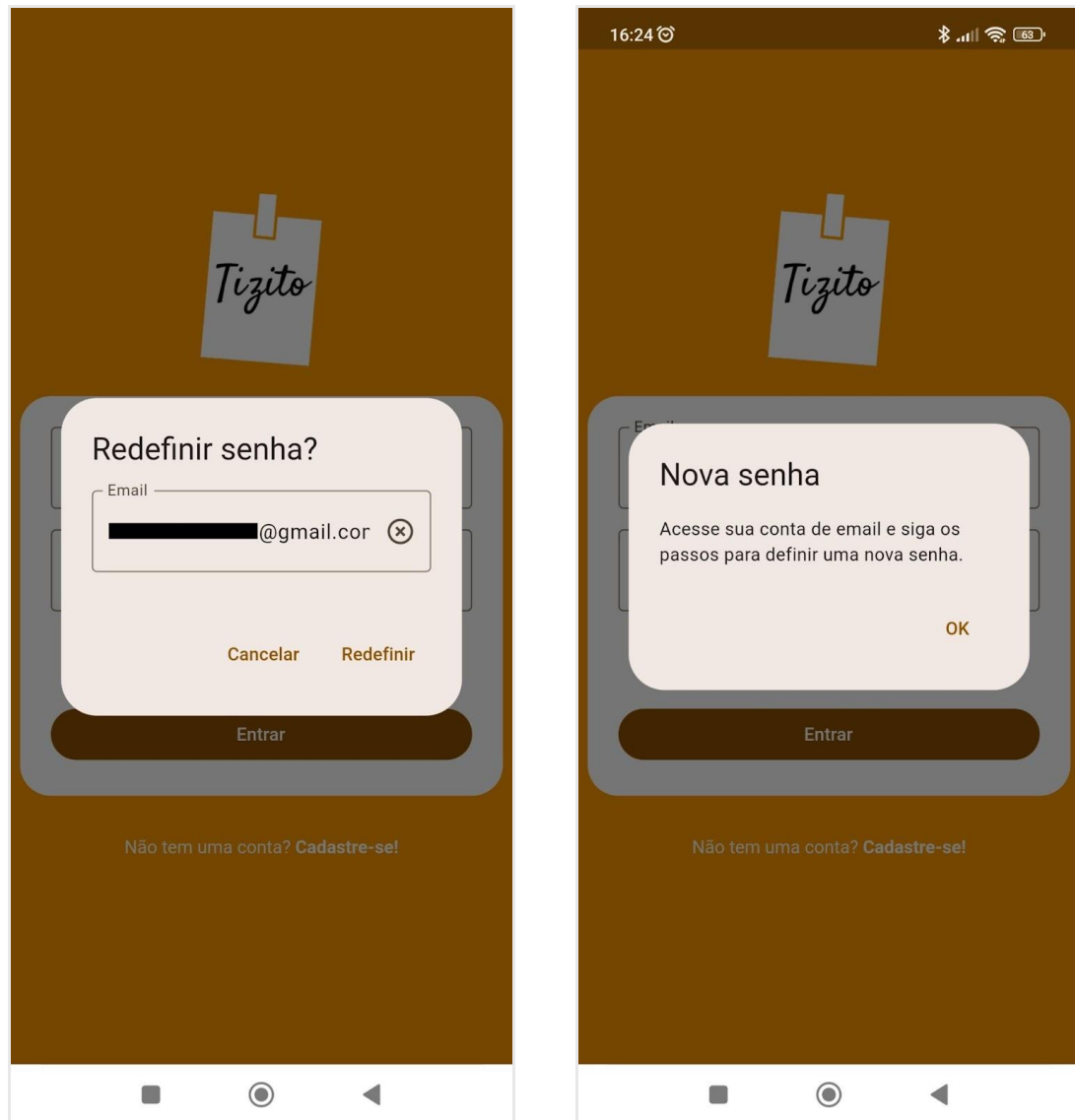


Figura 4.7 - Redefinição de palavra-passe.

4.2.3. Principal

O ecrã principal do Tizito é composto por quatro listas distintas:

- **Início**: encontros públicos dos amigos do utilizador;
- **Encontros**: encontros marcados pelo utilizador ou para os quais ele aceitou o convite, independentemente de serem públicos ou privados;
- **Convites**: convites recebidos pelo utilizador;
- **Amigos**: utilizadores da aplicação cuja solicitação de amizade foi aceite.

Além dessas listas, o ecrã principal também permite a navegação para ecrãs auxiliares por meio de um menu de opções, bem como para o ecrã de notificações do utilizador por meio de um ícone de sino. Esse arranjo contribui para uma organização intuitiva da *interface*.

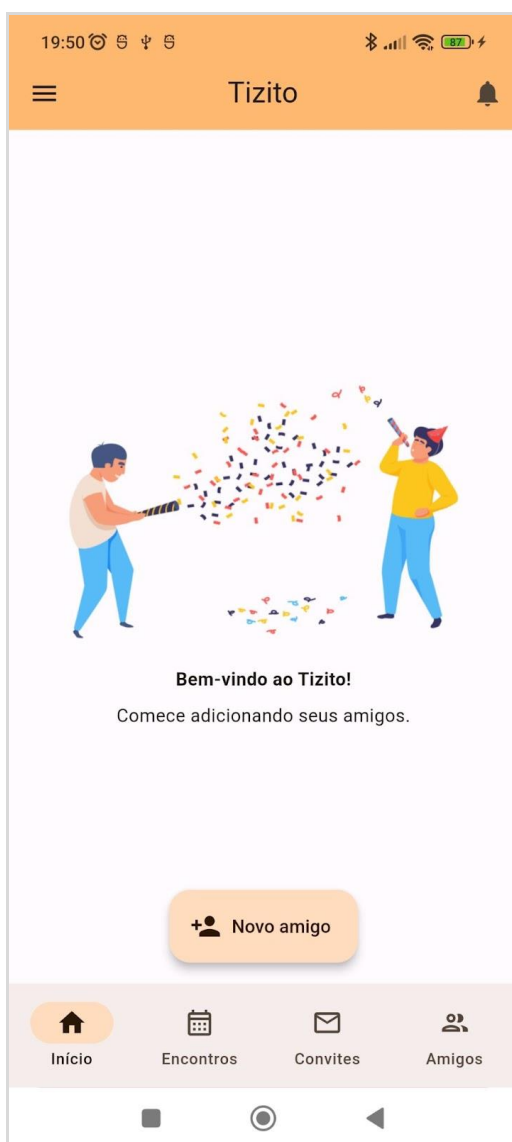


Figura 4.8 - Ecrã principal.

4.3. Amizade

Existem duas formas de adicionar um amigo no Tizito: por meio da leitura de um código QR (*Quick Response*) ou por meio de uma solicitação de amizade. Essa abordagem reflete o propósito de reunir pessoas que têm afinidade, convivência ou interesses em comum.

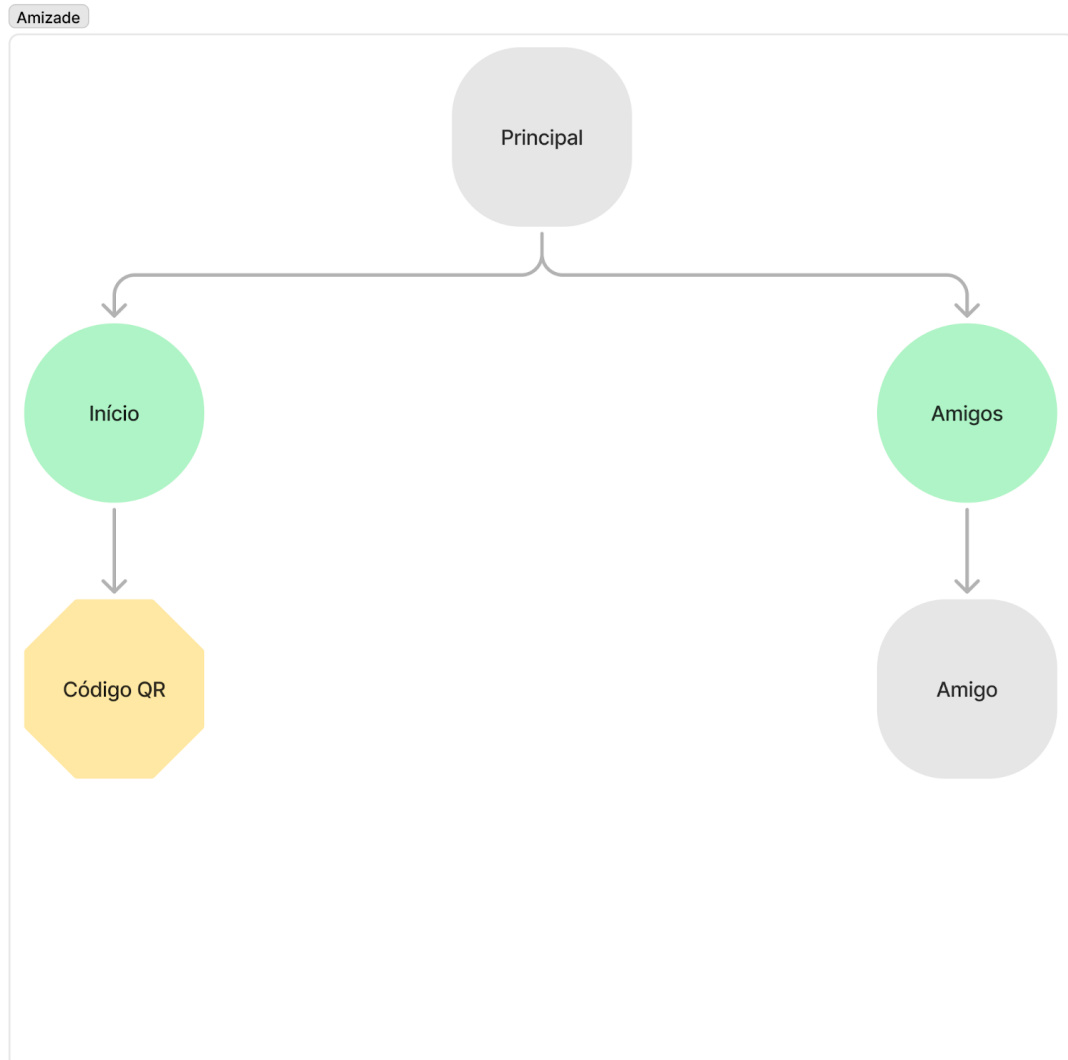


Figura 4.9 - Sitemap da amizade.

4.3.1. Código QR

A leitura de um código QR (código de barras bidimensional) é o único meio do utilizador adicionar o seu primeiro amigo. Por outras palavras, o utilizador deve pressionar o botão “Novo amigo” e escolher entre duas opções: digitalizar o código QR de outro utilizador da aplicação ou permitir que o seu próprio código QR seja digitalizado.

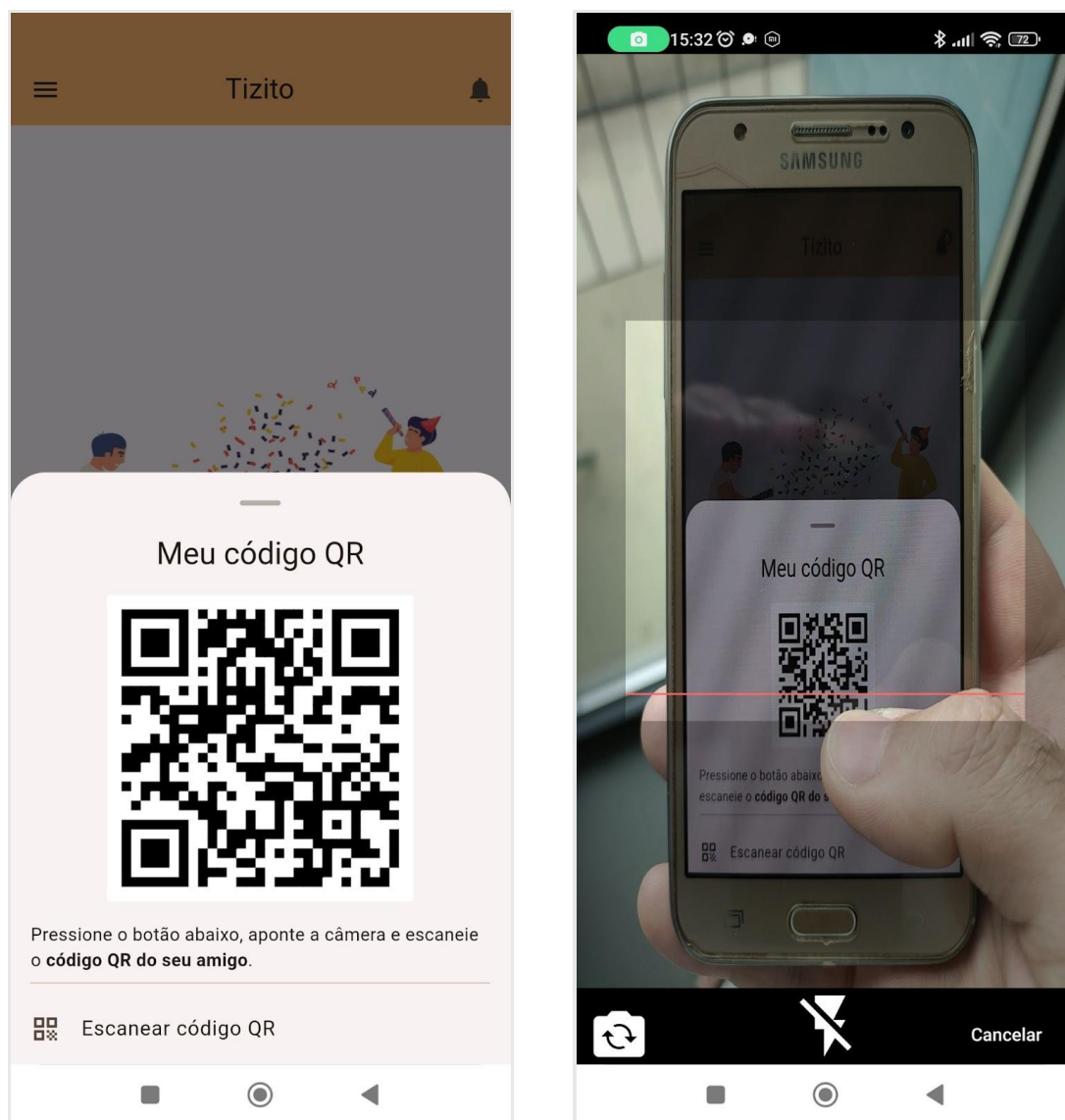


Figura 4.10 - Código QR.

4.3.2. Lista de amigos

Após adicionar um amigo, o utilizador pode visualizá-lo na lista de amigos. Ao pressionar a foto ou nome de um amigo, o utilizador navega até ao seu perfil e pode solicitar amizade de outros utilizadores da aplicação.

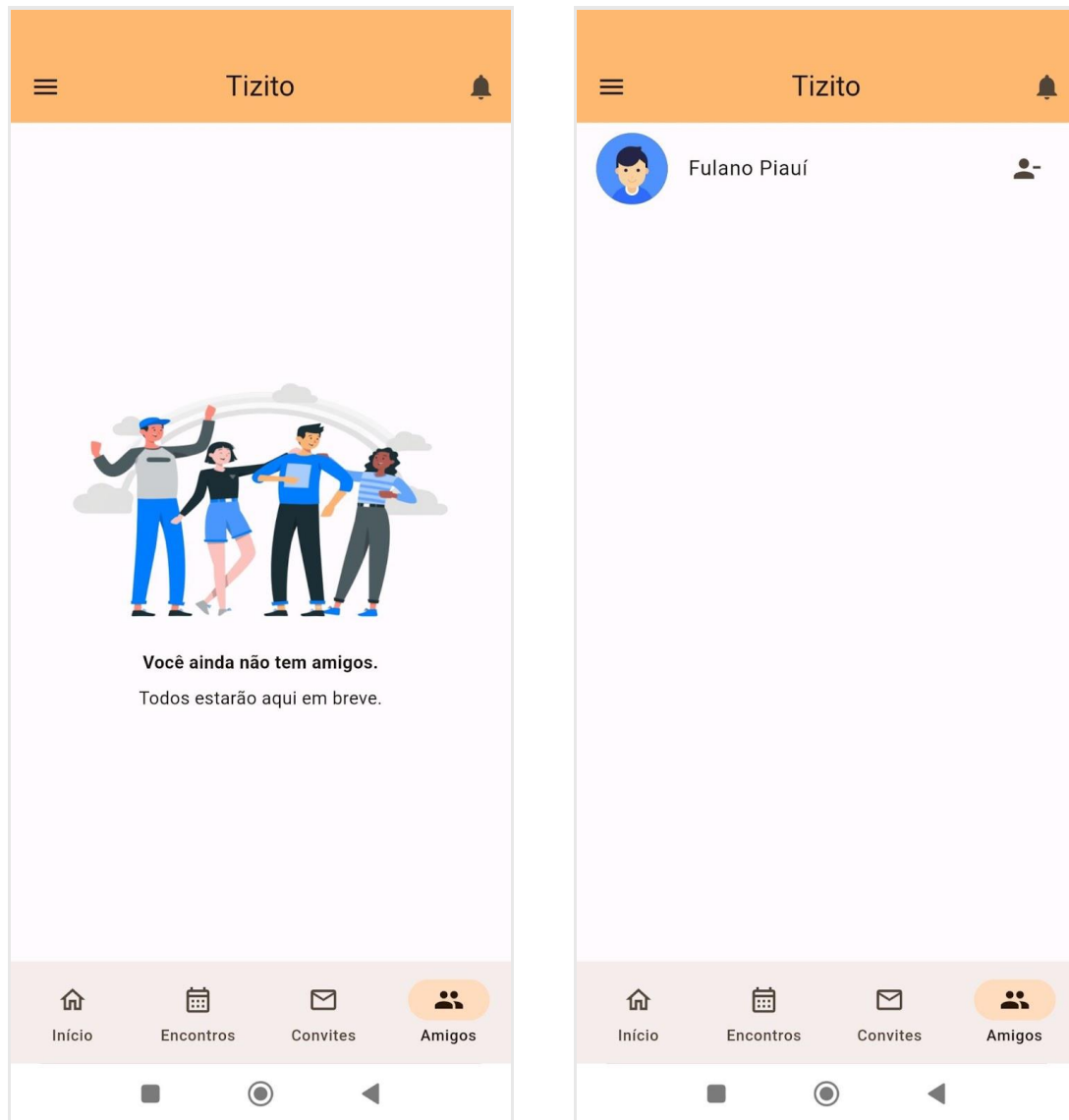


Figura 4.11 - Lista de amigos.

4.3.3. Amigo

Ao navegar até o perfil de um amigo, o utilizador visualiza a lista de amigos dele. Desse modo, basta adicionar os amigos em comum e cada um recebe uma solicitação de amizade.

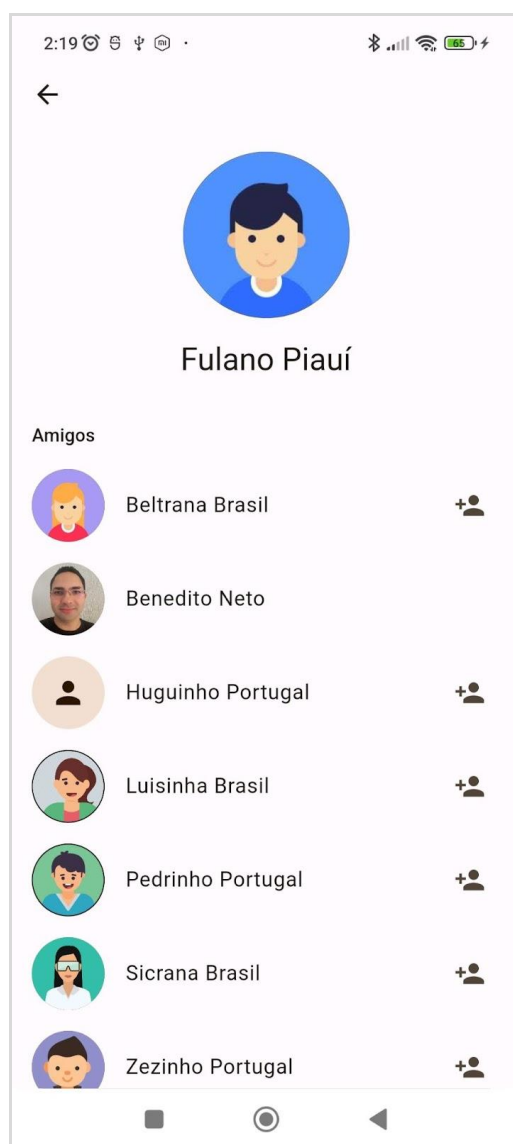


Figura 4.12 - Ecrã de amigo.

4.3.4. Solicitação de amizade

Após ser notificado, o utilizador pode aceitar ou recusar uma solicitação de amizade. Portanto, a lista de amigos tem como finalidade gerir as solicitações de amizade, bem como as amizades existentes.

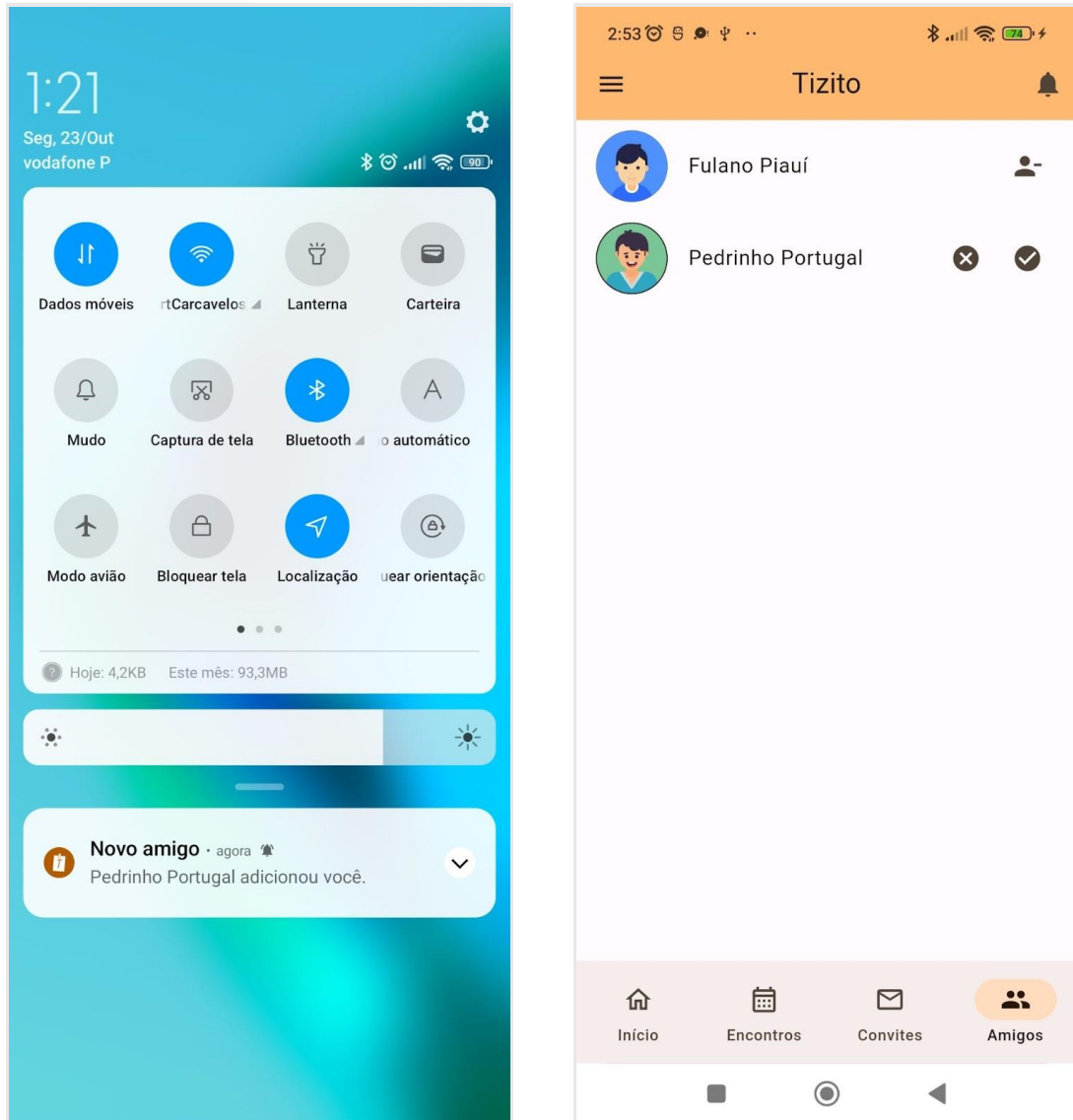


Figura 4.13 - Solicitação de amizade.

4.4. Encontros

Para marcar um encontro, o utilizador assume o papel de anfitrião. Isso significa que ele deve informar o título do encontro, escolher a data e hora, bem como o local e convidados.

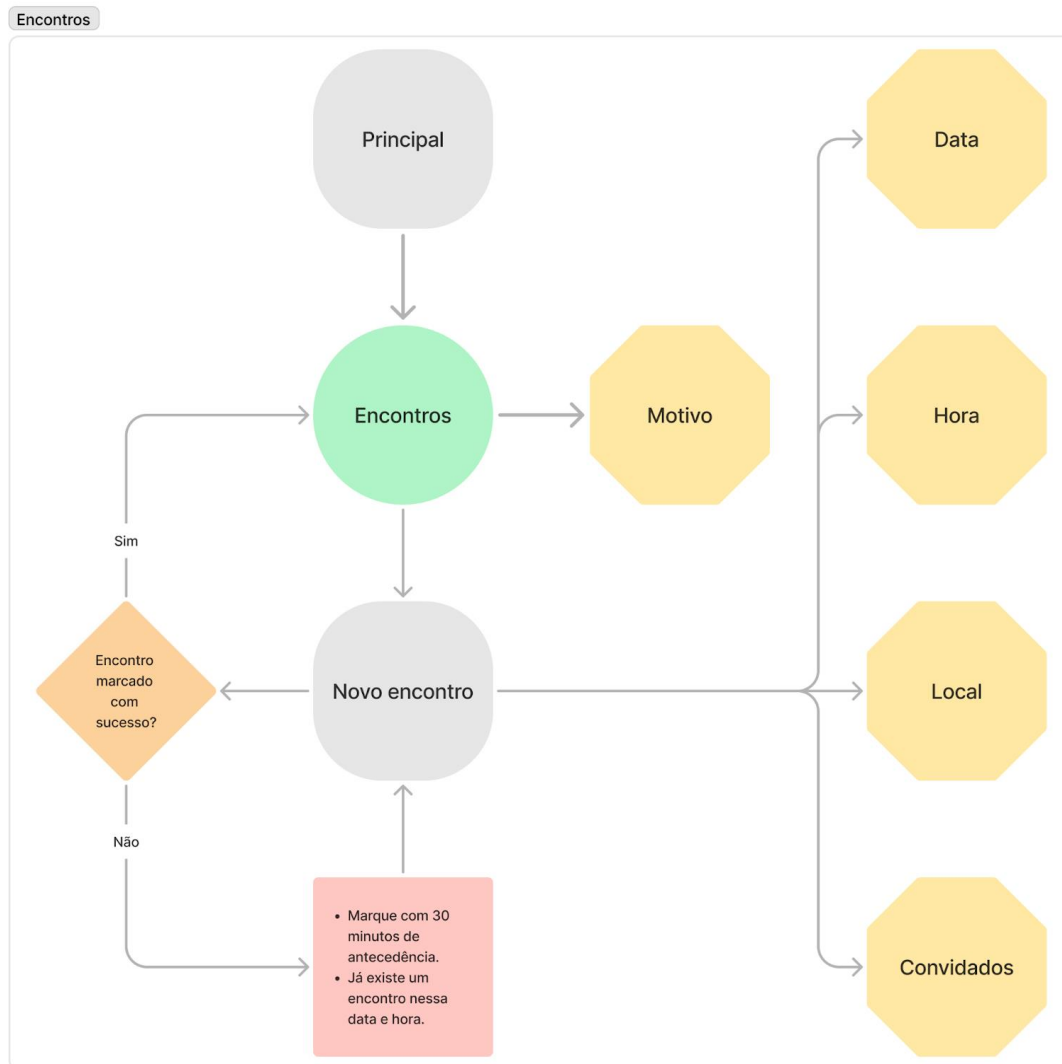


Figura 4.14 - Sitemap dos encontros.

4.4.1. Lista de encontros

O encontro deve ter pelo menos um convidado. Portanto, após adicionar o seu primeiro amigo, o utilizador pode marcar o primeiro encontro.

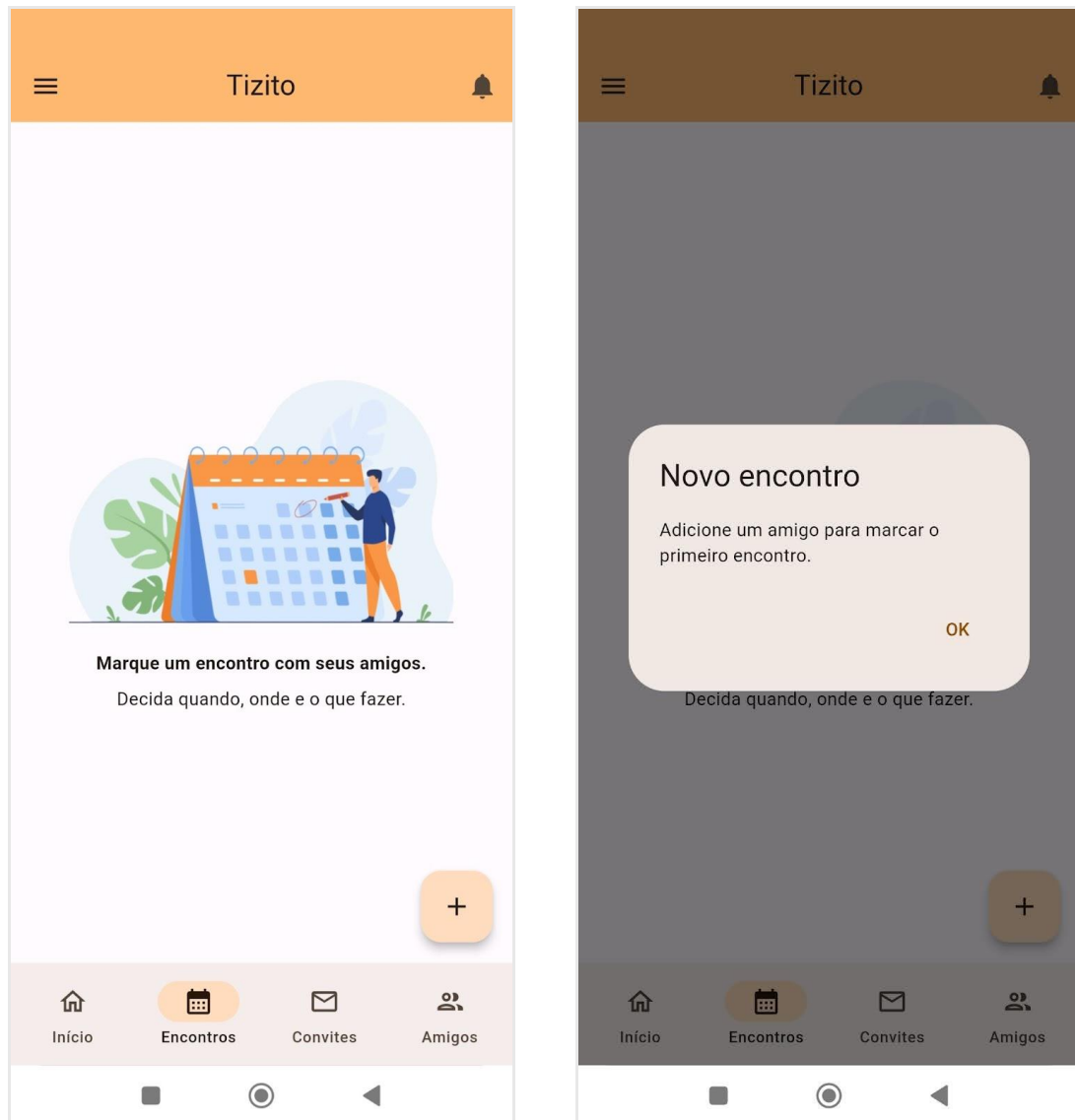


Figura 4.15 - Lista de encontros.

4.4.2. Título

O título do encontro desperta o interesse ou a curiosidade dos convidados. Portanto, é um detalhe que permite ao anfitrião total liberdade criativa, por exemplo: “Niver da Rê”, “Happy hour”, “Cinema”, “Exposição Portinari”, “Caminhada”, “Chá de bebê”, “Confra da equipa”, “Sinuquinha”, etc.



The screenshot shows a mobile application interface for creating a new event. At the top, there's a status bar with the time 17:28 and various icons. Below that, an orange header bar contains a close button (X), the title "Novo encontro", and a "Marcar" button. The main form area is white and contains several fields: a "Título" field with the text "Churrasco" and a clear button (X); a "Data" field showing "sáb, 21 de set" with a dropdown arrow; a "Hora" field showing "18:00" with a dropdown arrow; a "Local" field with a location pin icon; a "Privado" toggle switch that is turned on, with the text "Visível apenas para você e os convidados." below it; and a "Convidados" section with a "Selecionar" button. At the bottom, a keyboard is visible, and the app's navigation bar is at the very bottom.

Figura 4.16 - Título do encontro.

4.4.3. Data e hora

A data e hora são sempre no futuro. Como mencionado anteriormente, existe uma linha temporal com regras a serem seguidas. Além disso, encontros com a mesma data e hora não são permitidos.

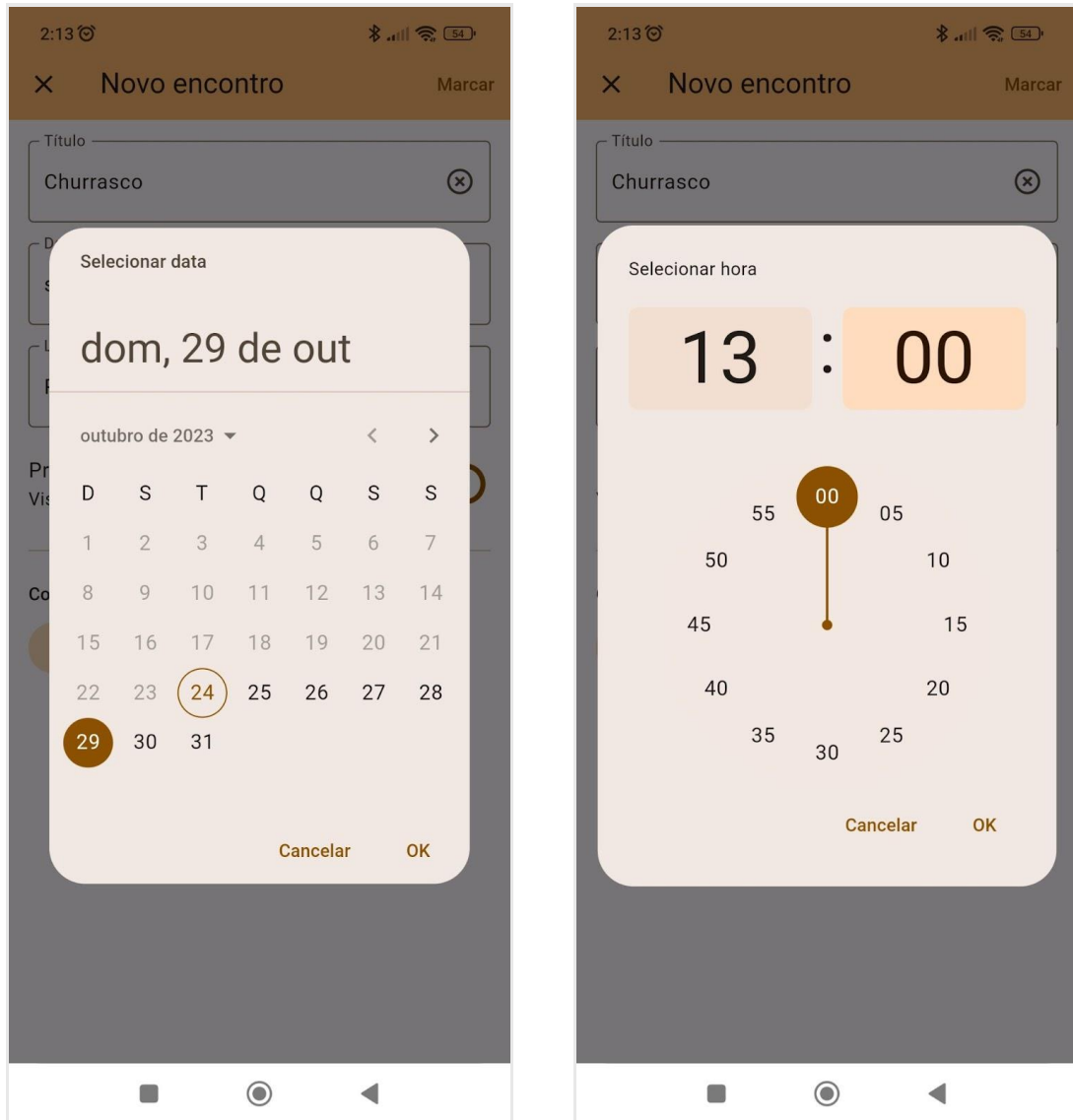


Figura 4.17 - Data e hora do encontro.

4.4.4. Local

Existem três maneiras de escolher o local do encontro:

- Localização atual do dispositivo móvel;
- Pesquisa por estabelecimentos;
- Marcação no mapa.

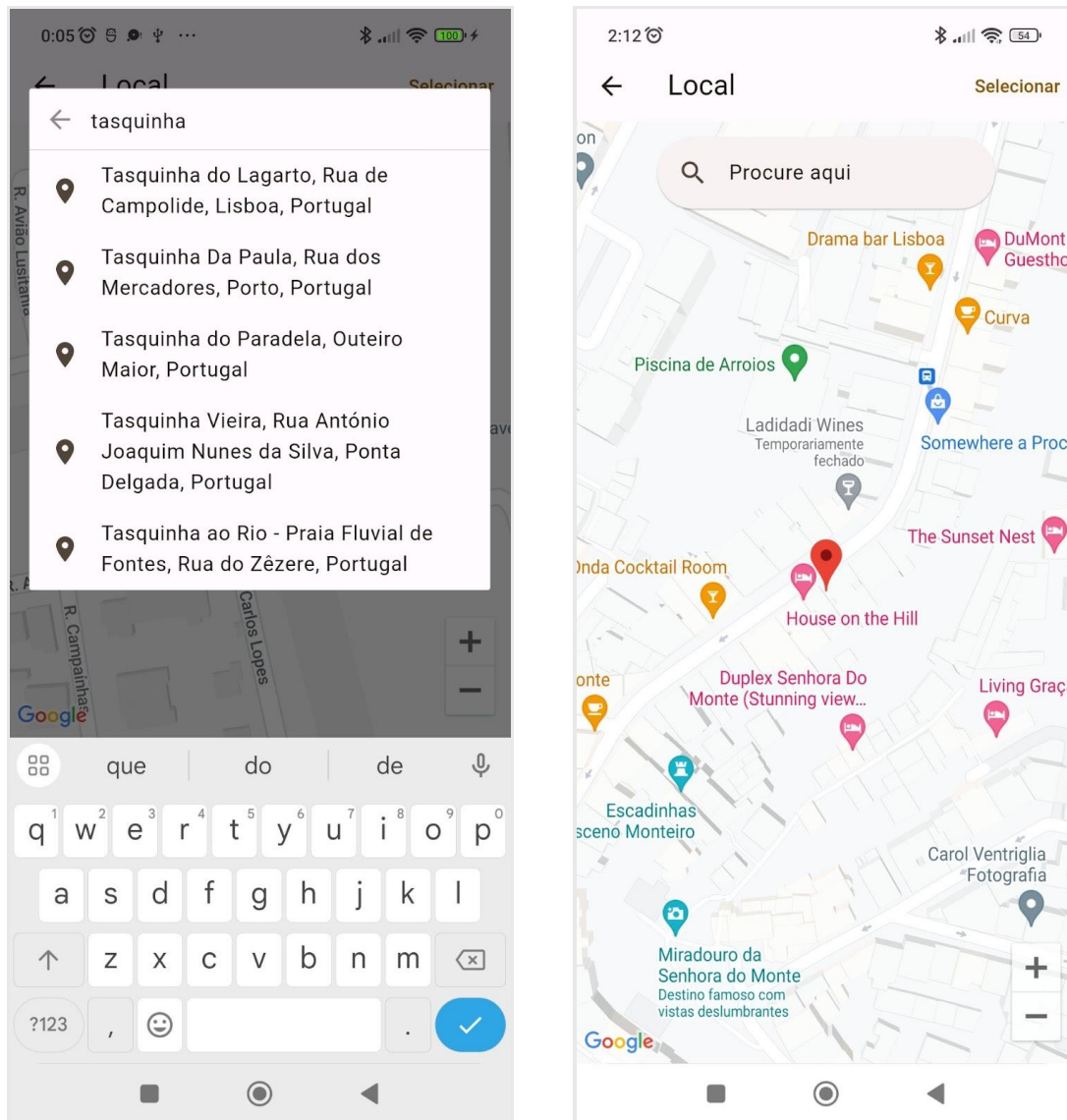


Figura 4.18 - Local do encontro.

4.4.5. Tipo

O tipo de encontro define a visibilidade das fotos e vídeos. Como mencionado anteriormente, os encontros podem ser públicos ou privados.

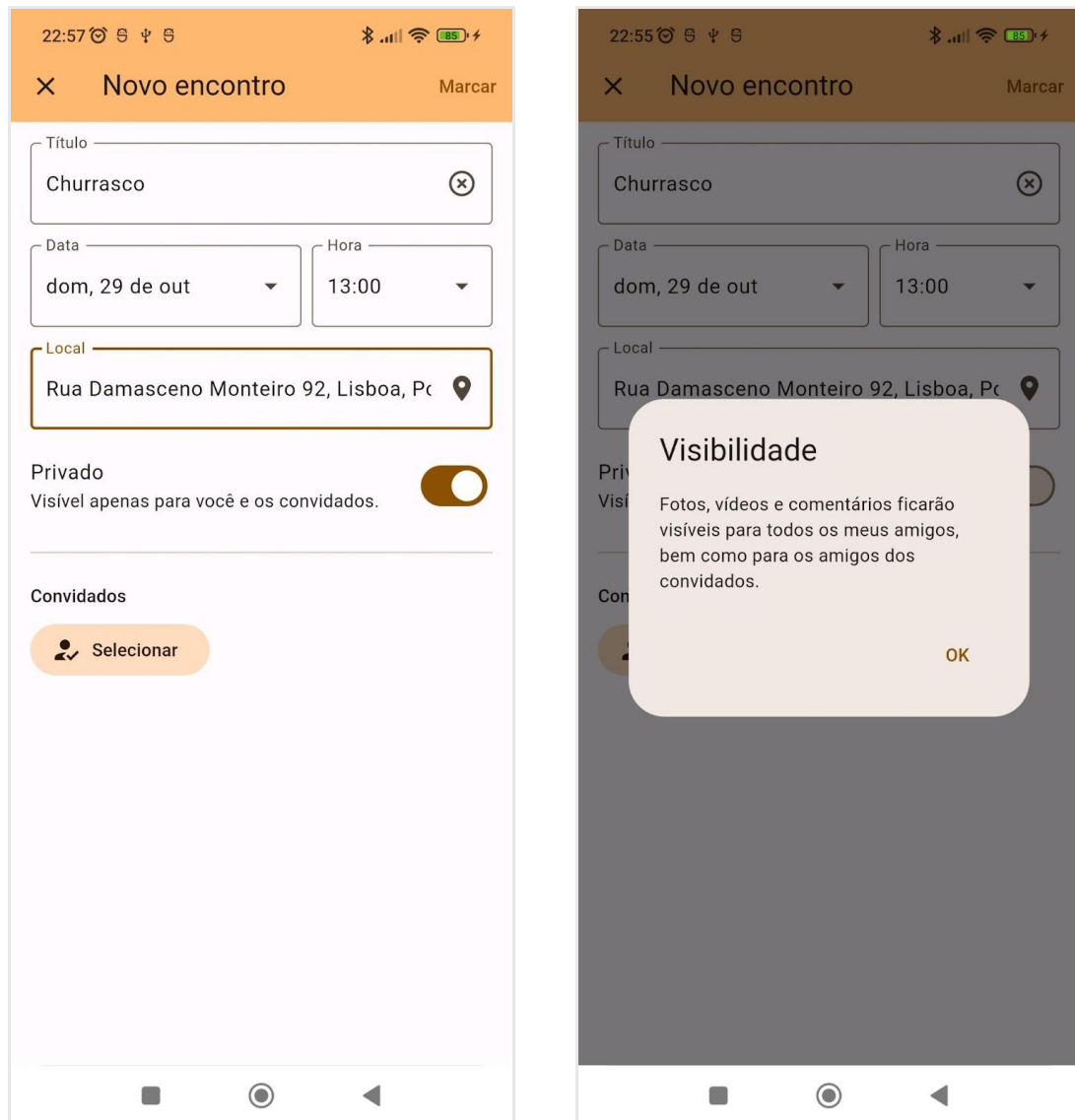


Figura 4.19 - Tipo do encontro.

4.4.6. Convidados

Os convidados pertencem exclusivamente à lista de amigos do utilizador. Por outras palavras, não é permitido convidar utilizadores da aplicação que não tenham uma solicitação de amizade aceite pelo anfitrião.

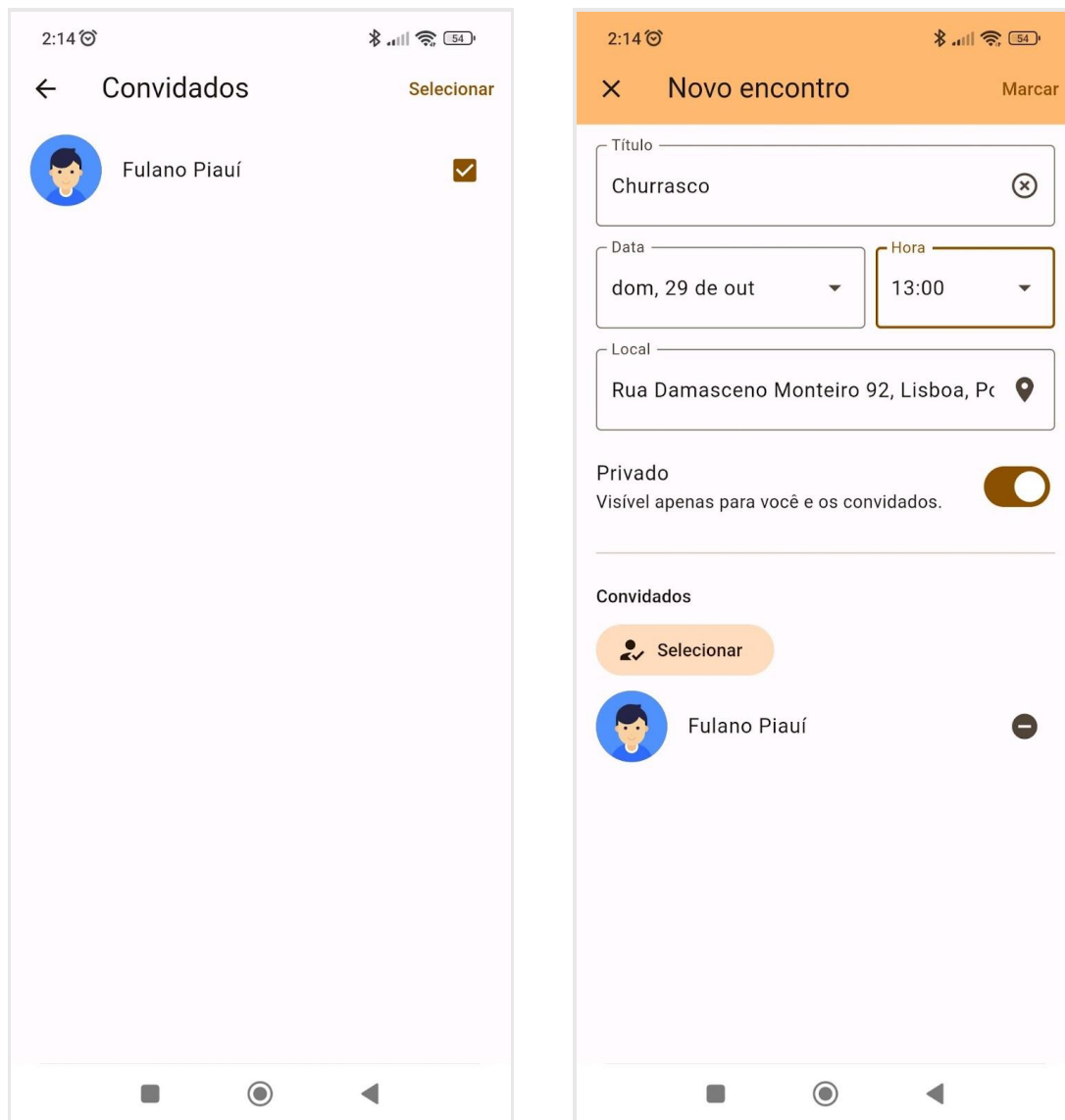


Figura 4.20 - Convidados do encontro.

4.4.7. Encontro marcado

Os encontros marcados são ordenados por data e hora. Além disso, ícones diferenciam o tipo de encontro: público (🌐) ou privado (🔒).

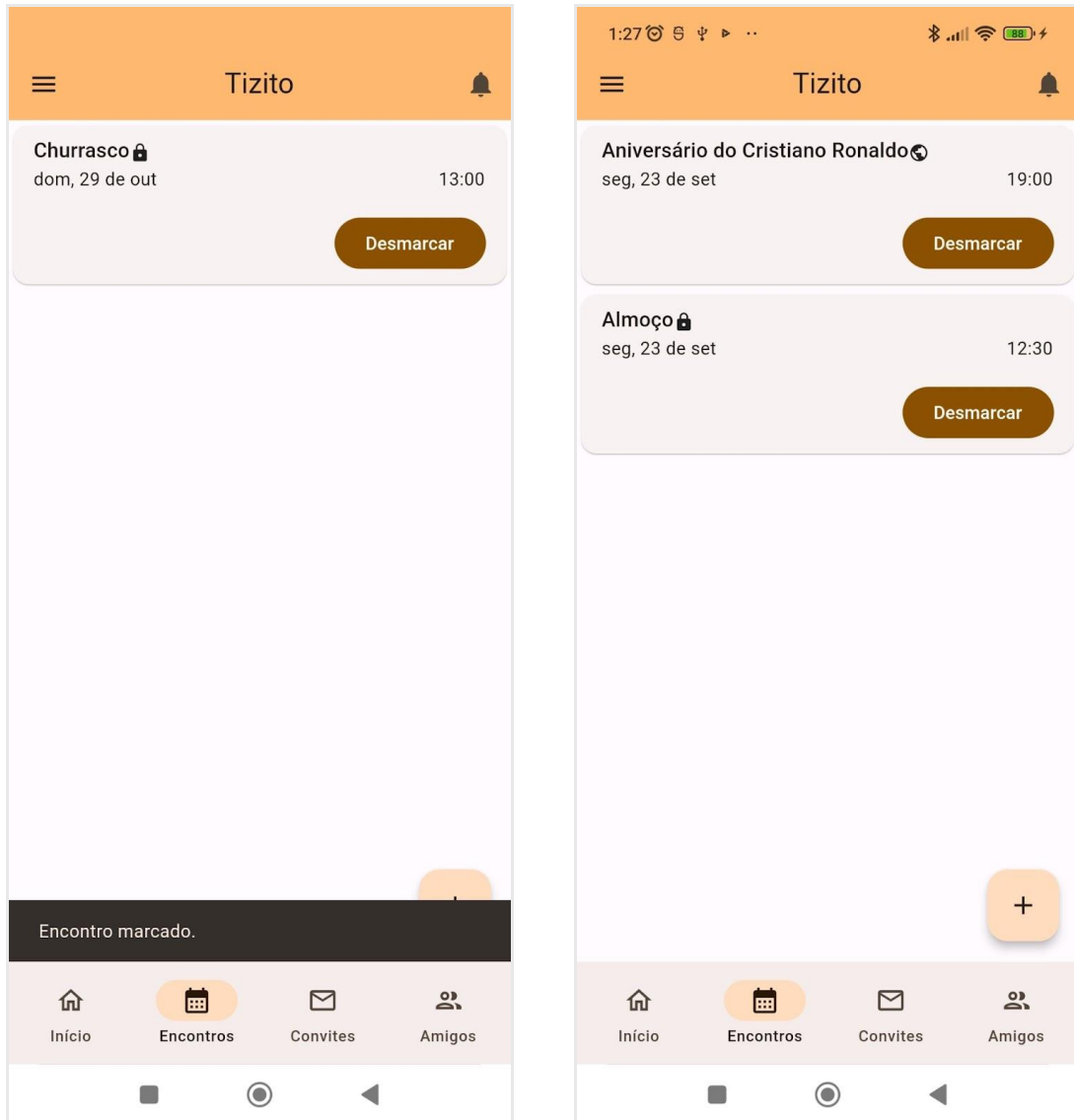


Figura 4.21 - Encontro marcado.

4.4.8. Encontro desmarcado

Para desmarcar um encontro, o anfitrião deve informar o motivo aos convidados que aceitaram o convite ou que ainda não responderam. Em seguida, os encontros desmarcados são removidos da lista de encontros.



Figura 4.22 - Encontro desmarcado.

4.4.9. Encontro passado

O Tizito utiliza cores distintas para diferenciar os encontros marcados que já aconteceram daqueles que ainda estão para acontecer.



Figura 4.23 - Encontro passado.

4.5. Convites

Quando um encontro é marcado, cada convidado é notificado e recebe um convite com a foto e nome do anfitrião, além dos detalhes do encontro.

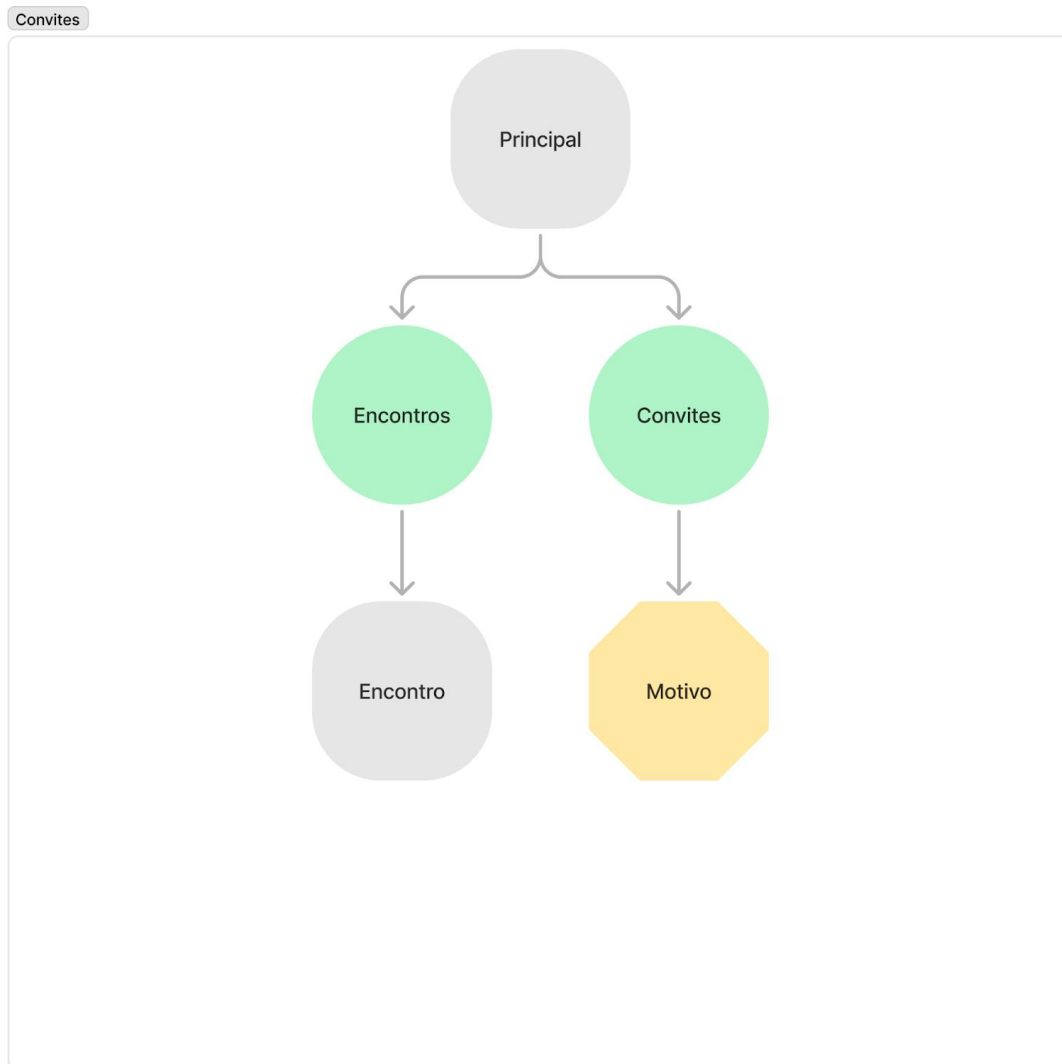


Figura 4.24 - *Sitemap* dos convites.

4.5.1. Lista de convites

Os convites são ordenados por data e hora do encontro. Desse modo, o convidado pode responder ao convite quando o encontro estiver mais próximo de acontecer. Além disso, uma seta para baixo expande o convite para mostrar mais detalhes do encontro e ajudar o convidado a decidir entre aceitar ou recusar. Se não houver resposta, o convite é removido da lista de convites após a data e hora do encontro.

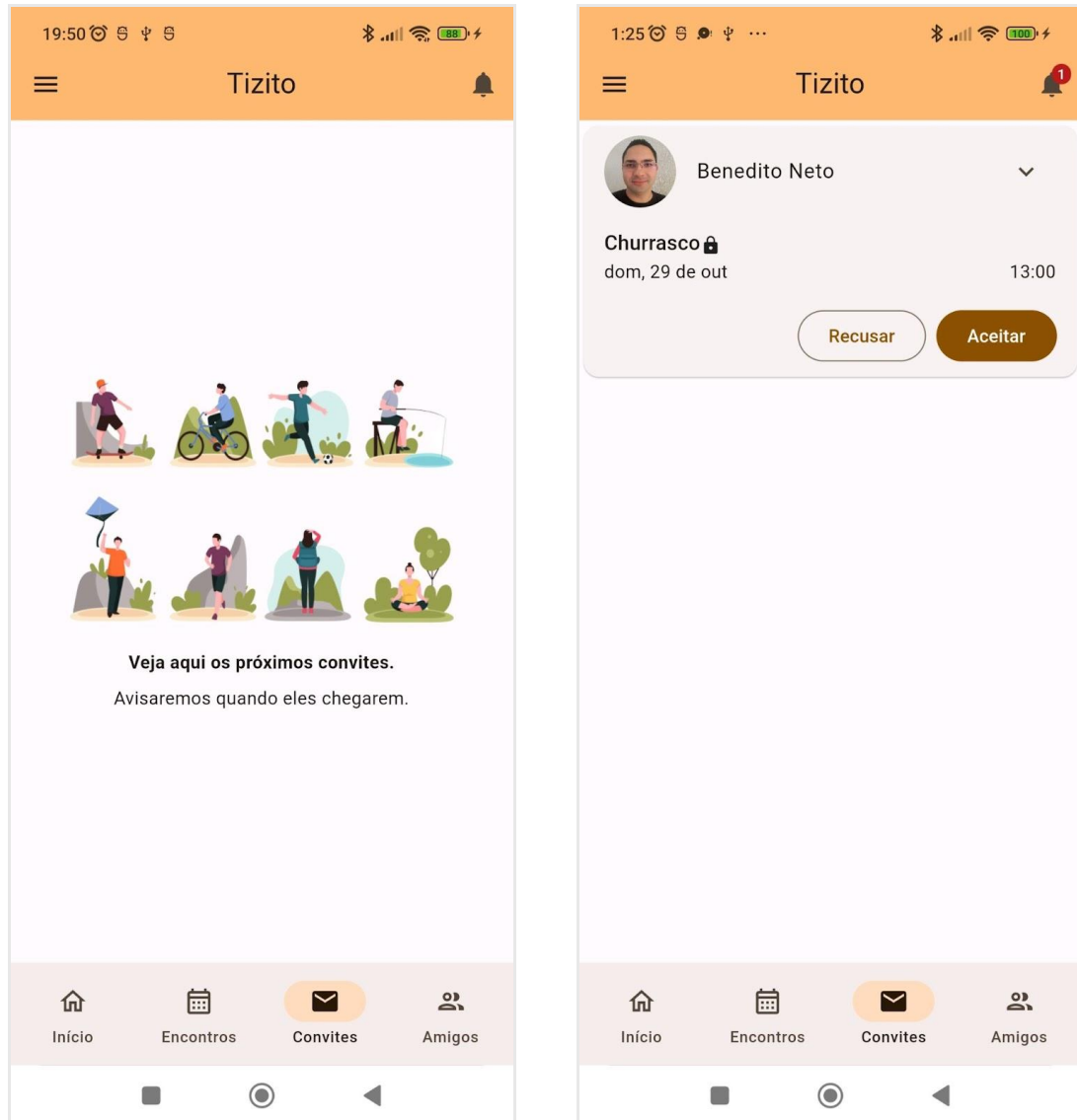


Figura 4.25 - Lista de convites.

4.5.2. Convite aceite

Se o convite for aceite, o convidado pode visualizar posteriormente os detalhes do encontro na lista de encontros.

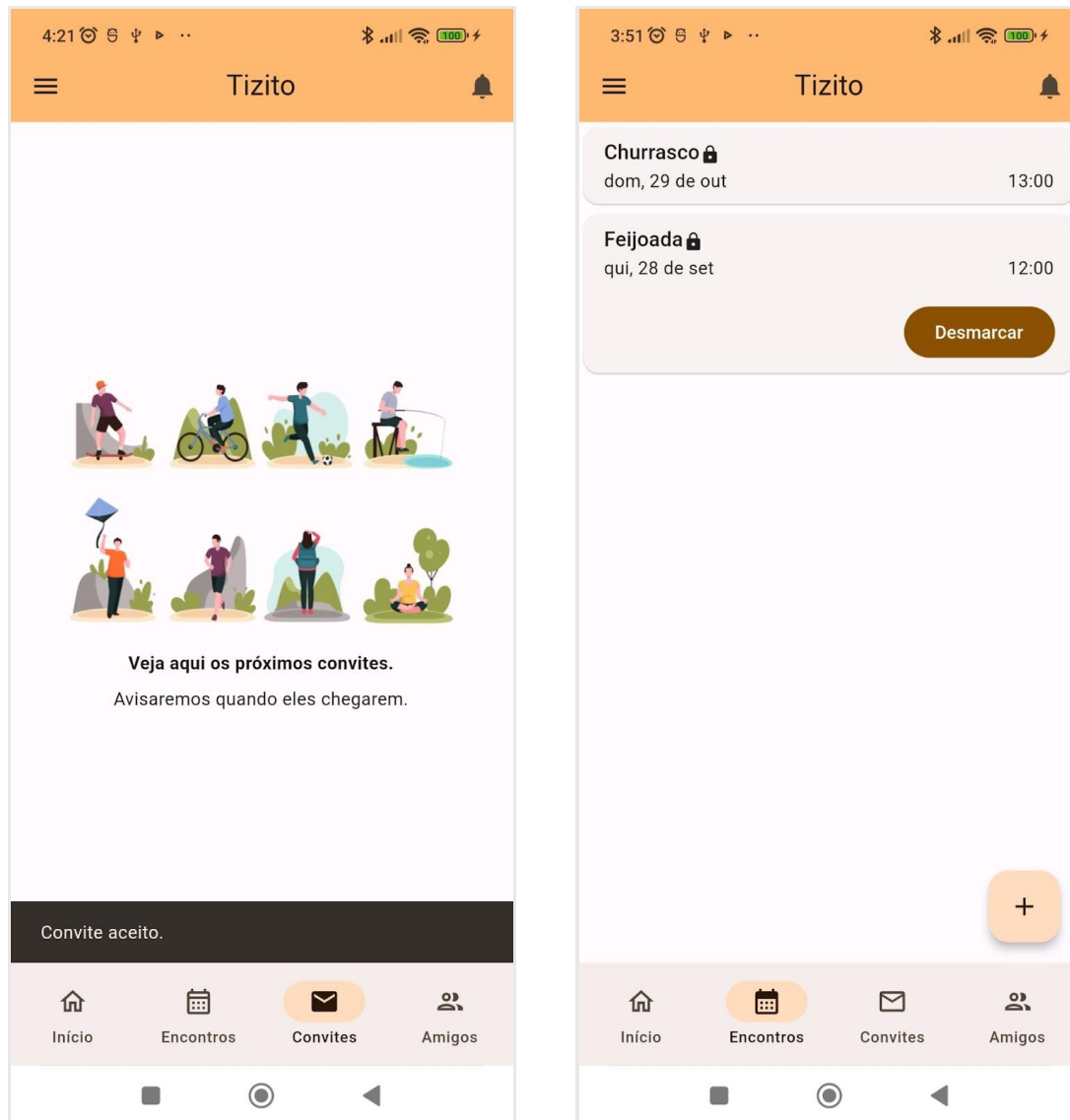


Figura 4.26 - Convite aceite.

4.5.3. Convite recusado

Se o convite for recusado, o convidado pode informar o motivo para explicar a sua ausência. É importante salientar que apenas o anfitrião pode visualizar o motivo.

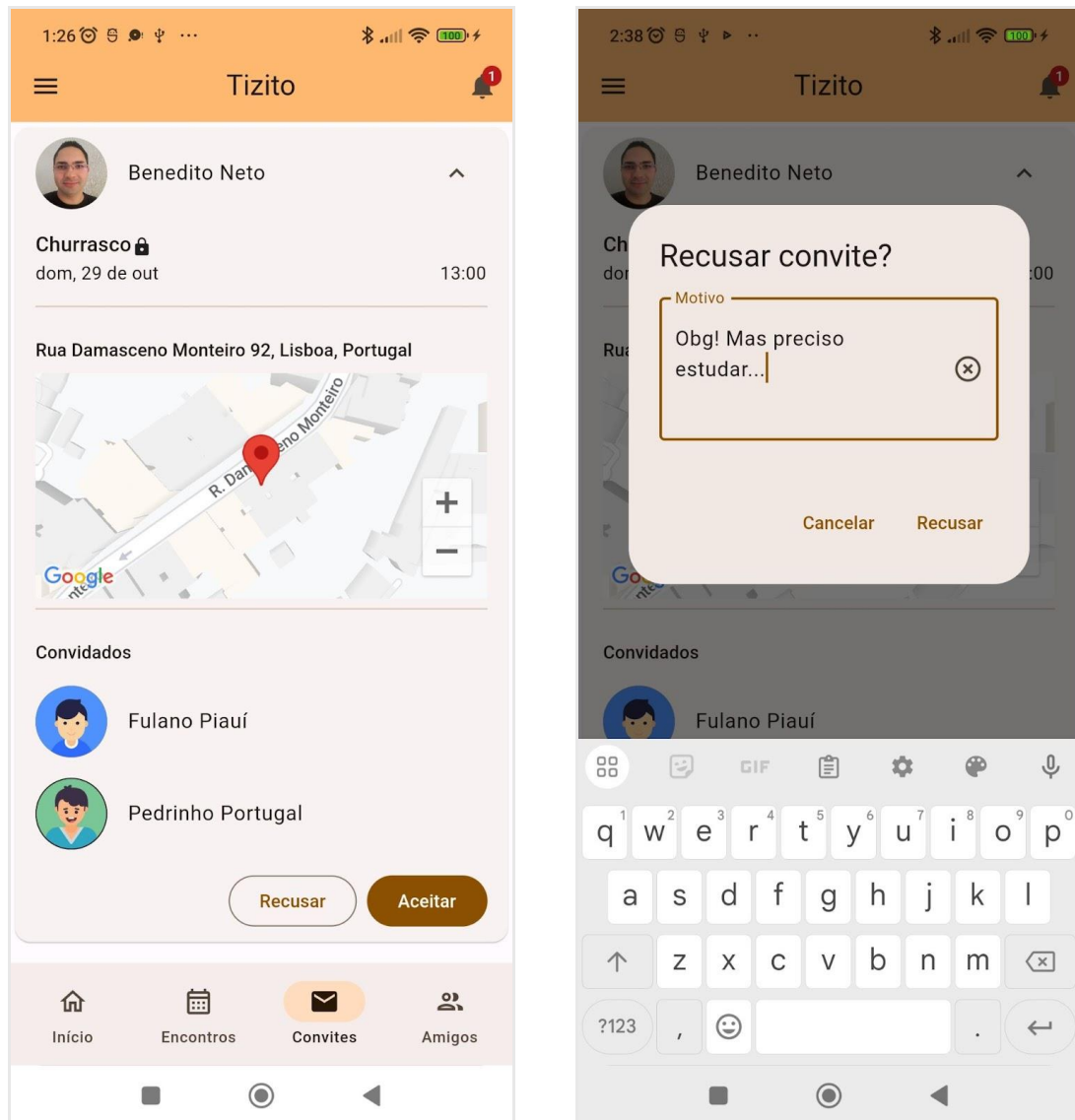


Figura 4.27 - Convite recusado.

4.5.4. Visibilidade

Se o convite para um encontro público for aceite, o convidado deve confirmar a sua decisão. Nesse cenário, a aplicação destaca a visibilidade das fotos e vídeos para que o convidado possa refletir se deve aceitar realmente o convite.



Figura 4.28 - Visibilidade.

4.5.5. Encontro

A perspetiva do anfitrião é diferente da perspetiva do convidado:

- O anfitrião pode visualizar a resposta de cada convite, ou seja, quem aceitou ou recusou, bem como quem ainda não respondeu. No entanto, o convidado apenas visualiza quem são os restantes convidados;
- O anfitrião pode alterar o título e local do encontro, bem como convidar mais amigos. No entanto, o convidado apenas envia mensagens de texto.

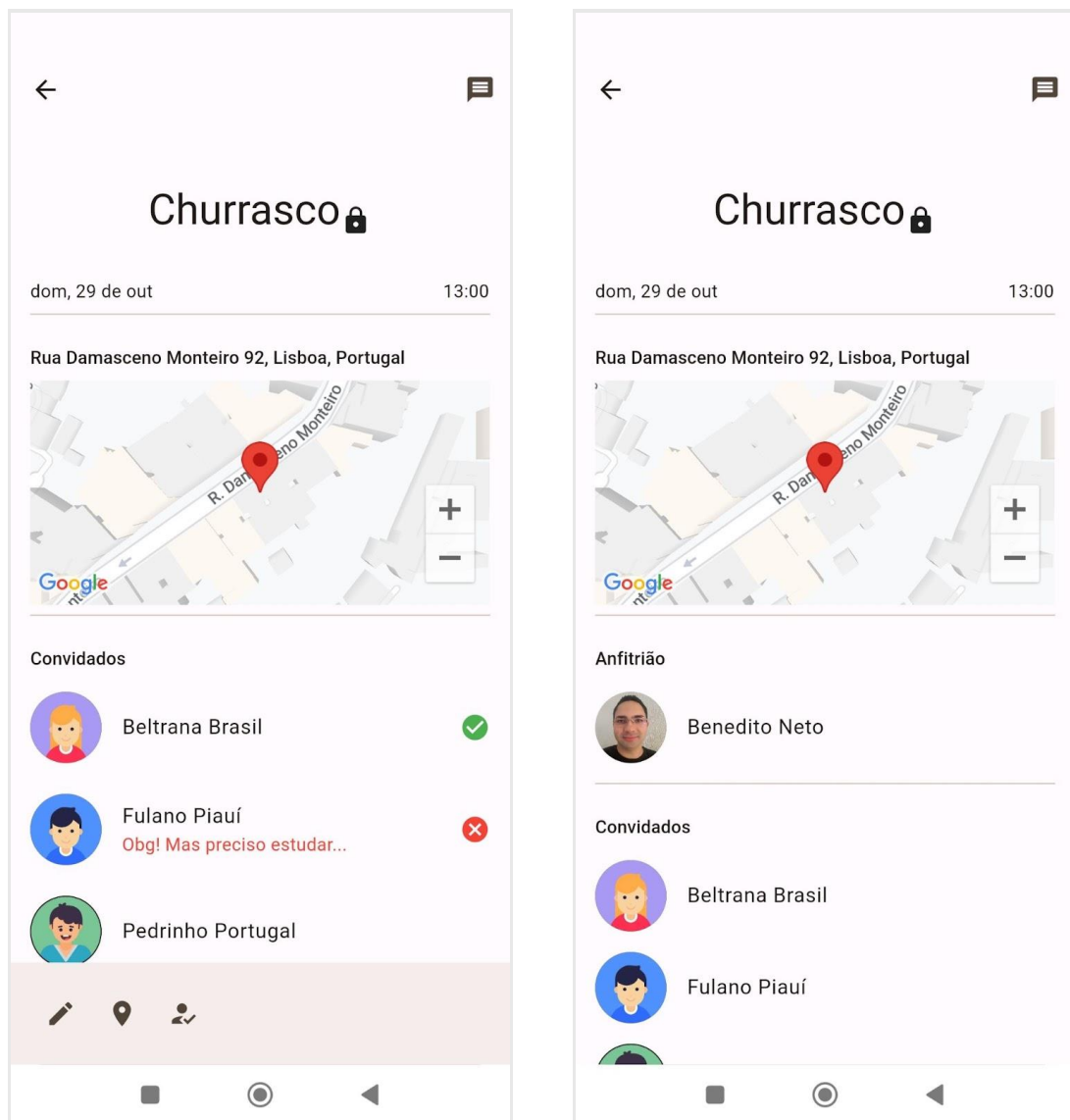


Figura 4.29 - Ecrã do encontro.

4.5.6. Mensagens

O Tizito permite o envio instantâneo de mensagens de texto. Desse modo, o anfitrião e convidados podem comunicar detalhes adicionais sobre o encontro. Normalmente, essas mensagens são avisos ou dúvidas relacionadas com o encontro.

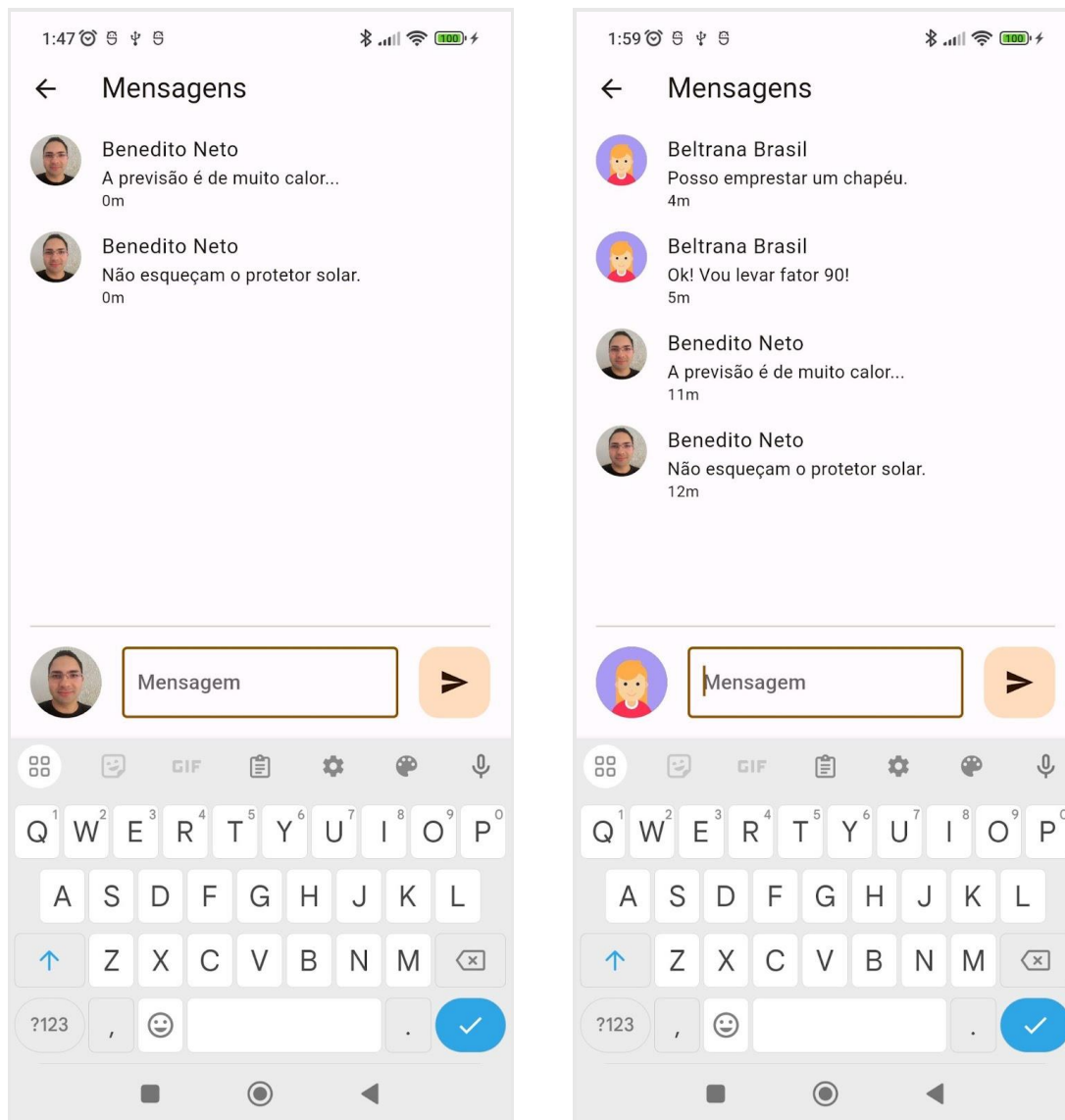


Figura 4.30 - Mensagens do encontro.

4.6. Recordações

Tanto o anfitrião como os convidados que aceitaram o convite podem efetuar publicações de fotos, vídeos e comentários. Existe um botão distinto para cada tipo de publicação. As fotos e vídeos são apresentados por meio de um carrossel, enquanto os comentários são apresentados por meio de um diálogo. O responsável por cada publicação é identificado pela foto e nome.

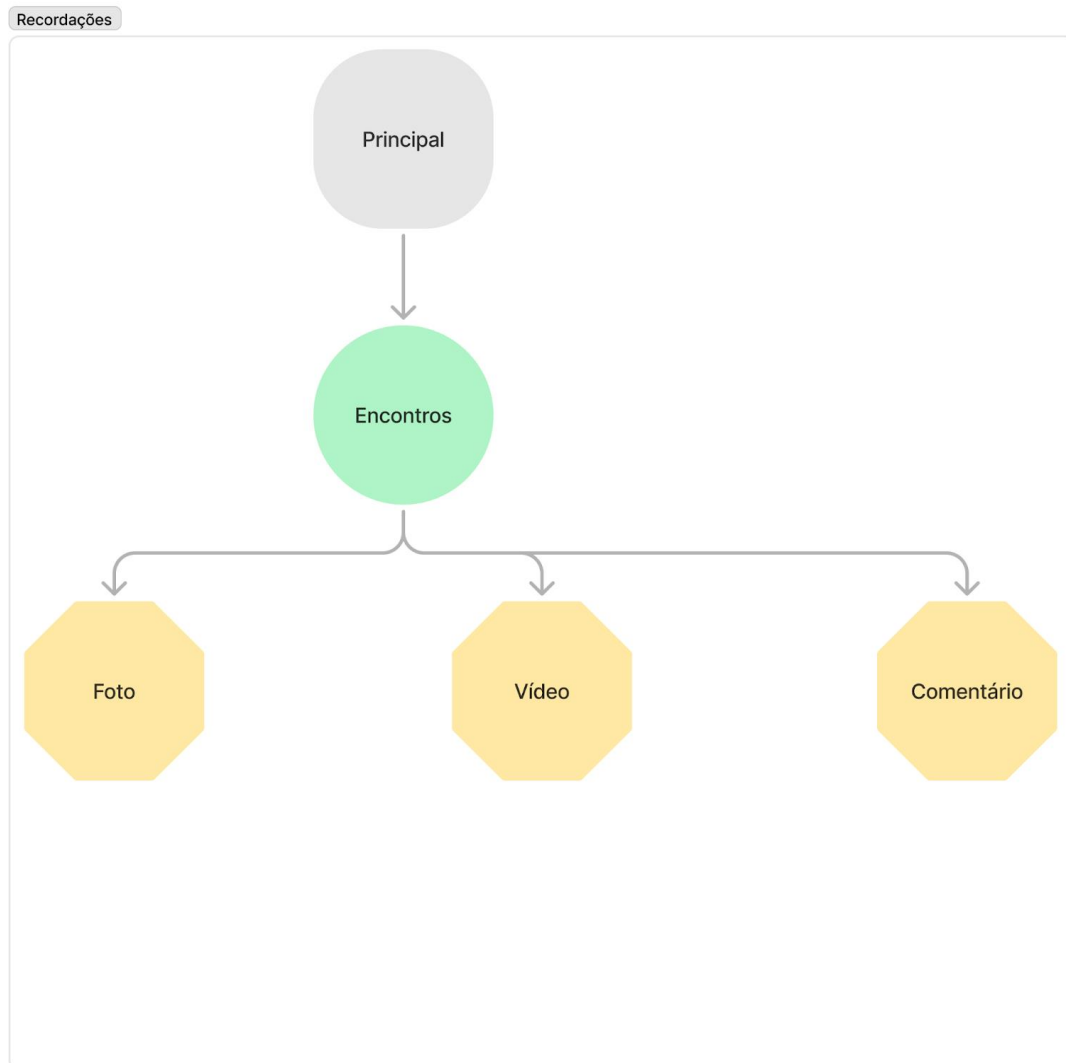


Figura 4.31 - Sitemap das recordações.

4.6.1. Fotos

Para publicar uma foto, basta utilizar a câmara do dispositivo móvel para tirar uma nova foto ou selecionar uma foto existente.

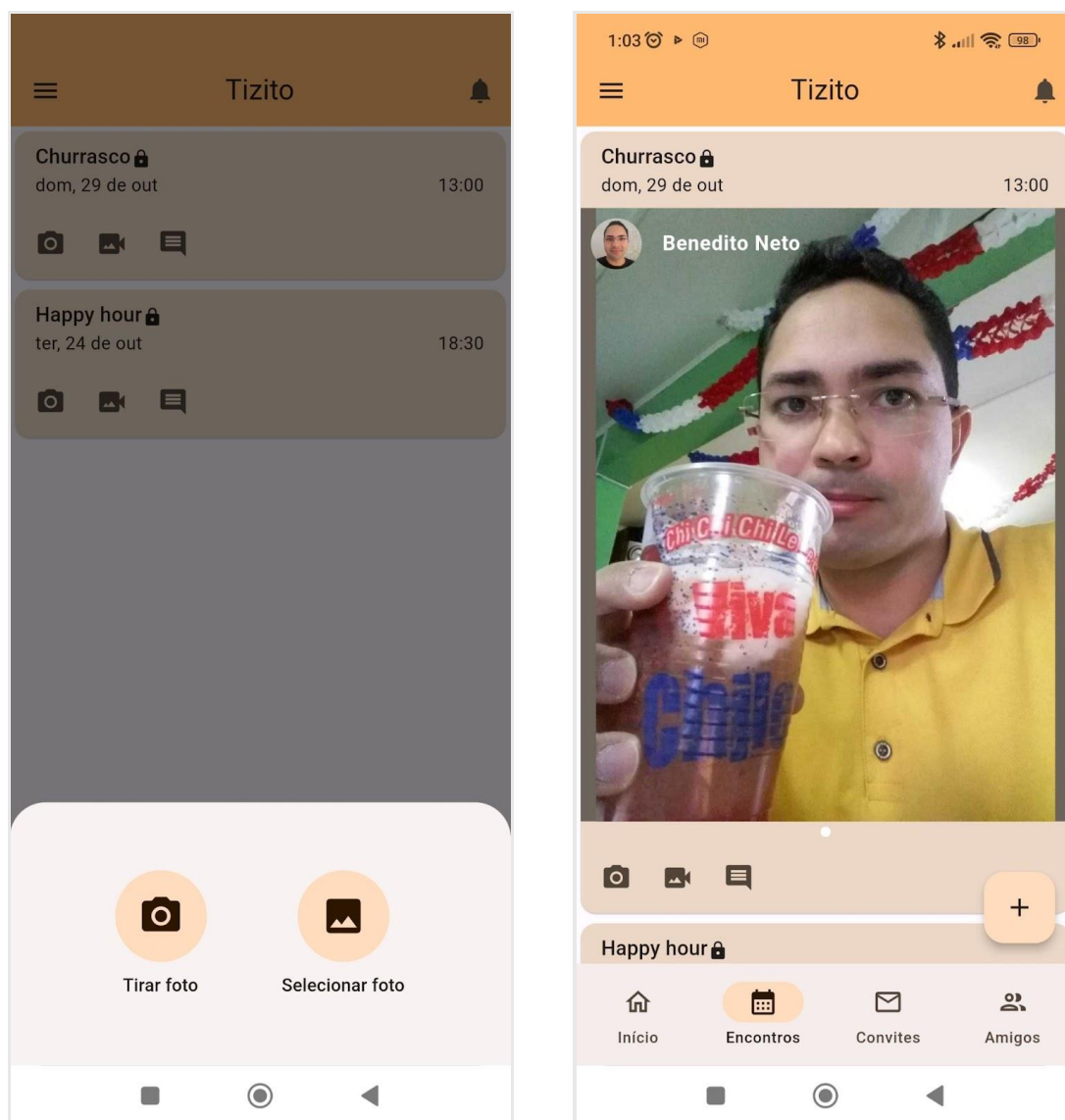


Figura 4.32 - Fotos do encontro.

4.6.2. Vídeos

Para publicar um vídeo, basta utilizar a câmara do dispositivo móvel para gravar um novo vídeo ou selecionar um vídeo existente.

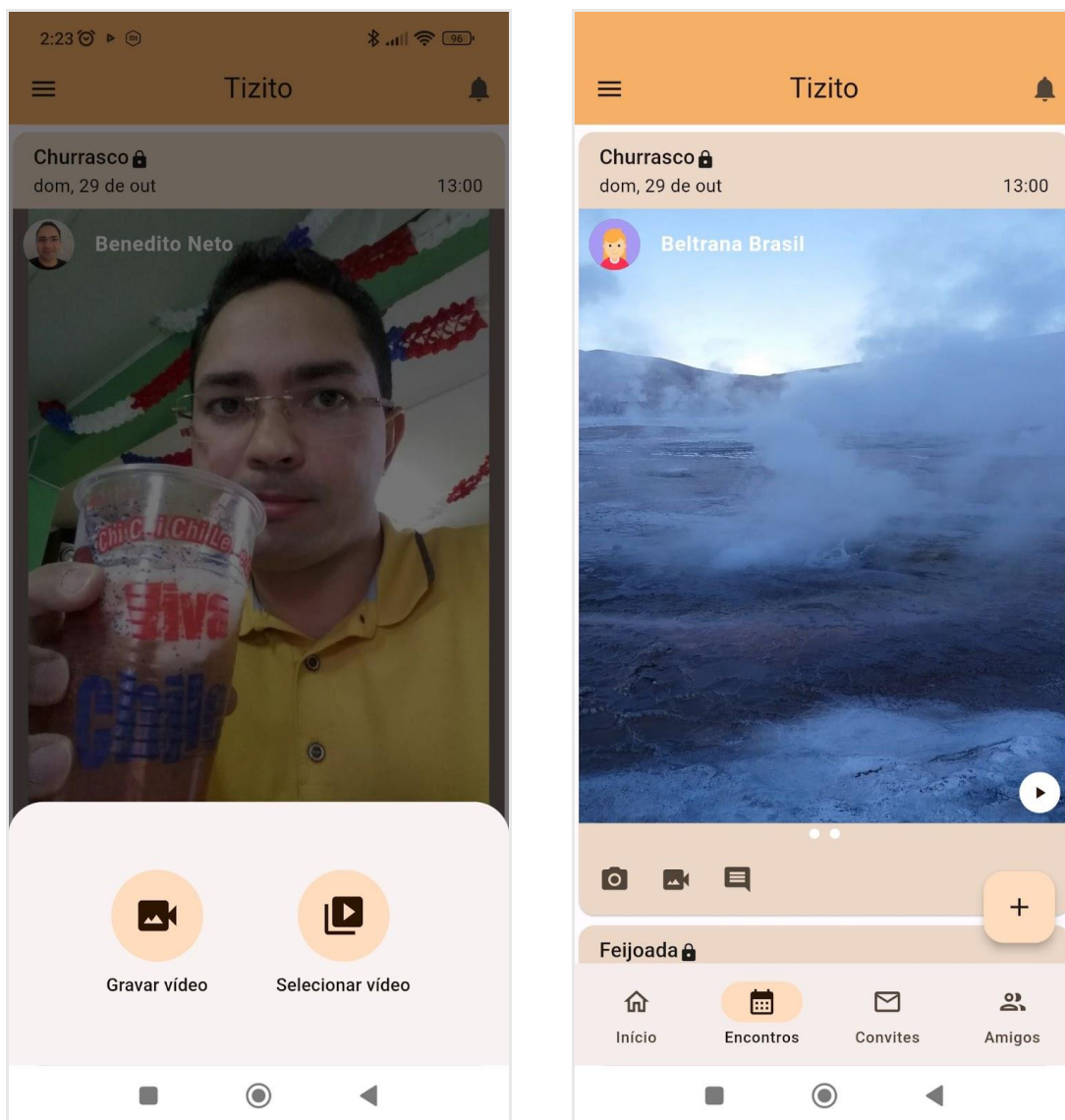


Figura 4.33 - Vídeos do encontro.

4.6.3. Comentários

Para publicar um comentário, basta pressionar o botão “Enviar” após informar o texto desejado.

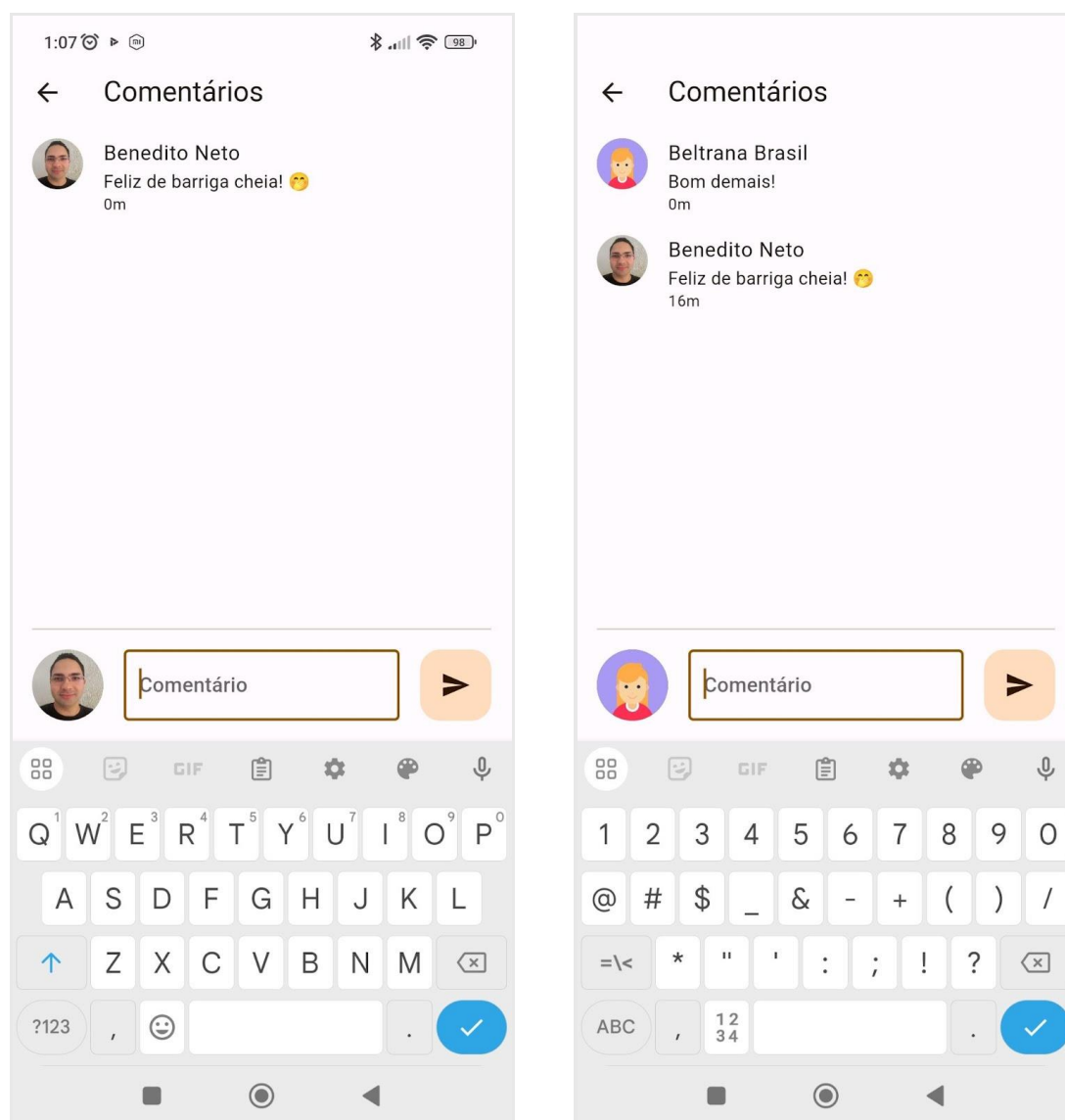


Figura 4.34 - Comentários do encontro.

4.7. Início

O utilizador pode visualizar as fotos e vídeos dos encontros públicos dos seus amigos, bem como gostar e comentar. Por outras palavras, ele não terá conhecimento dos encontros privados dos seus amigos.

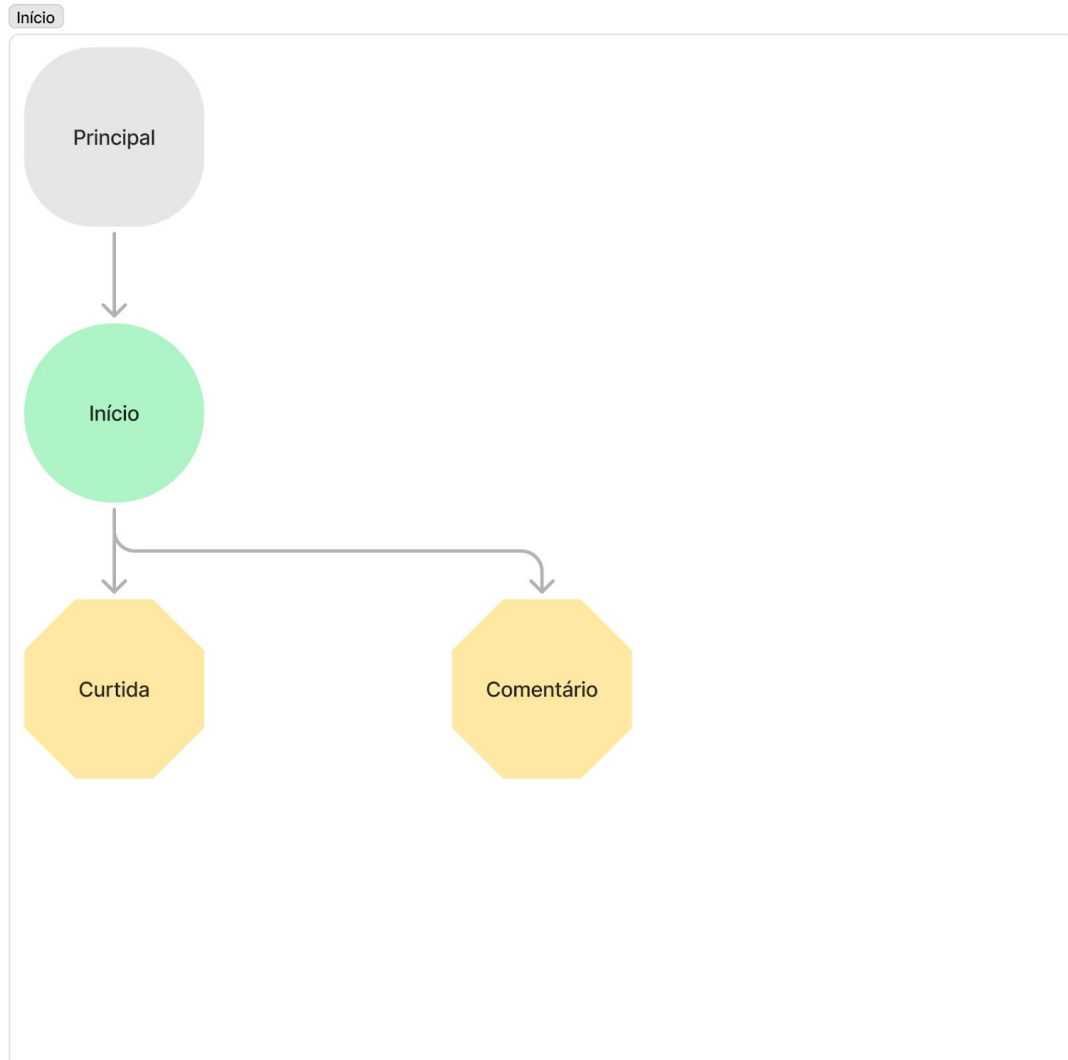


Figura 4.35 - *Sitemap* do início.

4.7.1. Encontro público

A primeira lista do ecrã principal do Tizito reúne os encontros públicos dos amigos do utilizador, que também são ordenados por data e hora. Neste caso, os encontros mais recentes ficam no topo da lista.

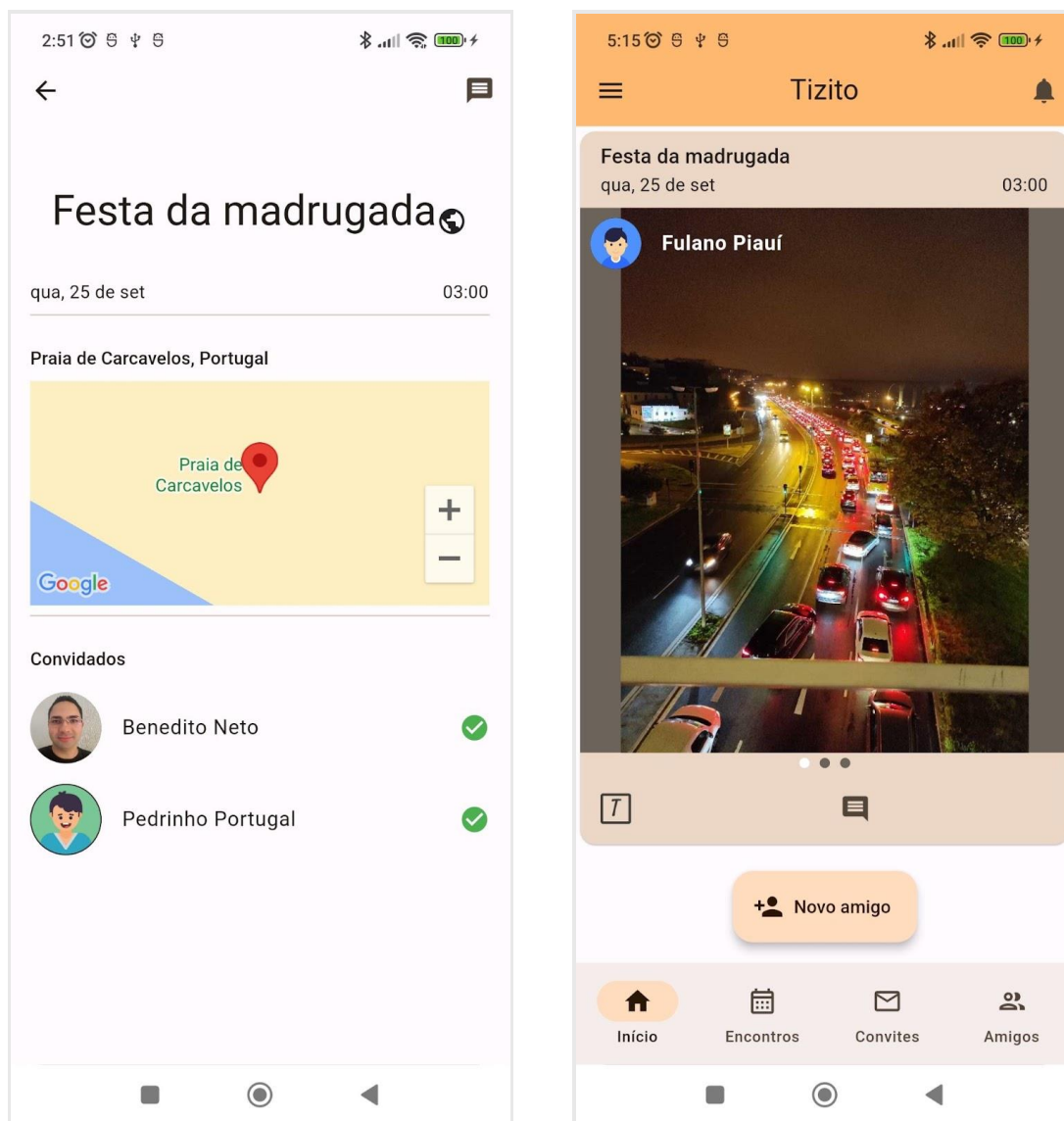


Figura 4.36 - Encontro público.

4.7.2. Gostos

O Tizito tem o seu próprio botão “Gostar” (*Like button*) com o intuito de reforçar o reconhecimento da marca. Ao lado desse botão, é possível visualizar o número de gostos, ou seja, o número de utilizadores que gostaram do encontro. Ao pressionar esse número, é possível visualizar esses utilizadores.

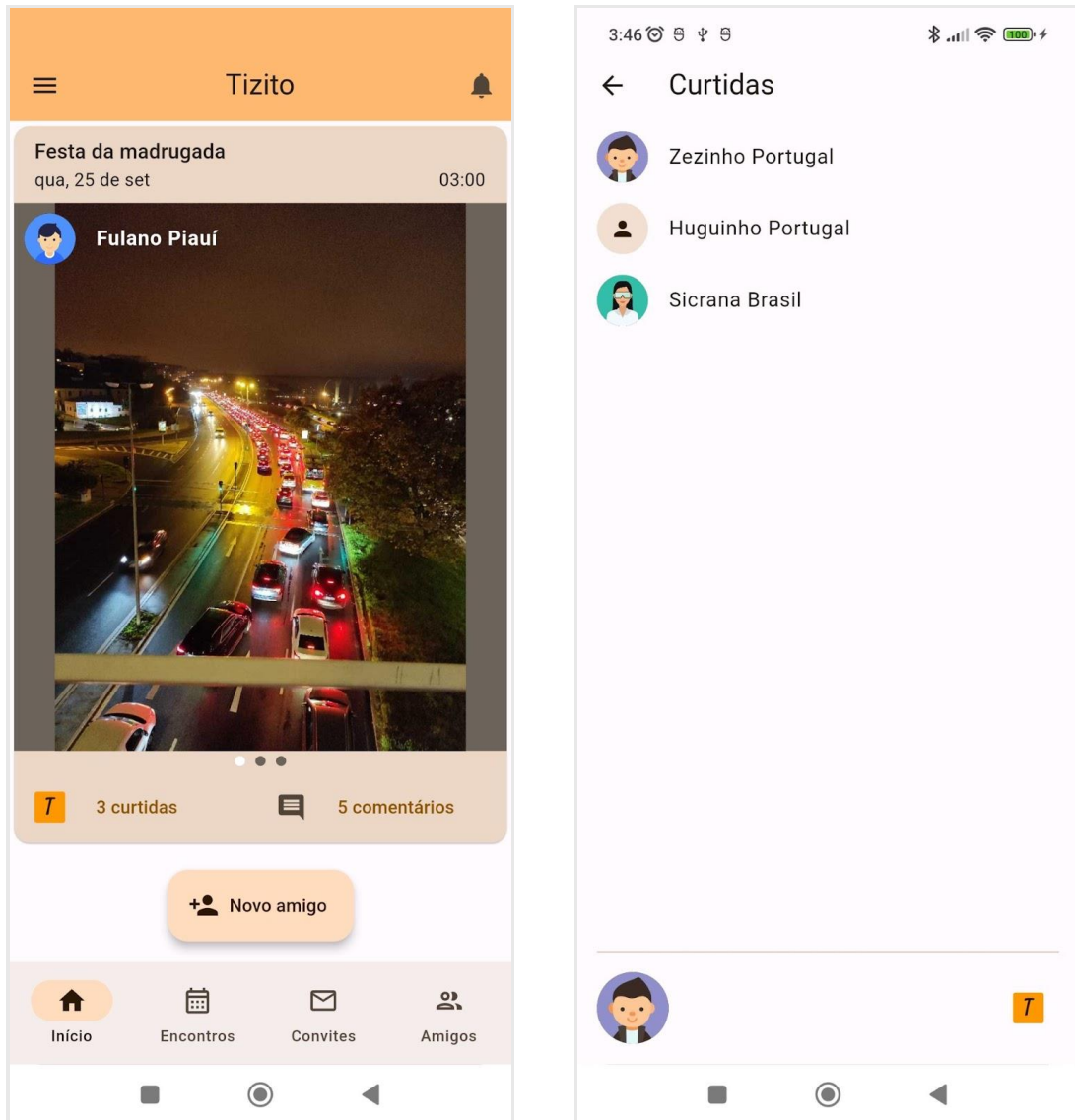


Figura 4.37 - Gostos do encontro.

4.7.3. Comentários

O utilizador também pode comentar um encontro público. Os comentários são adicionados aos comentários do anfitrião e convidados.

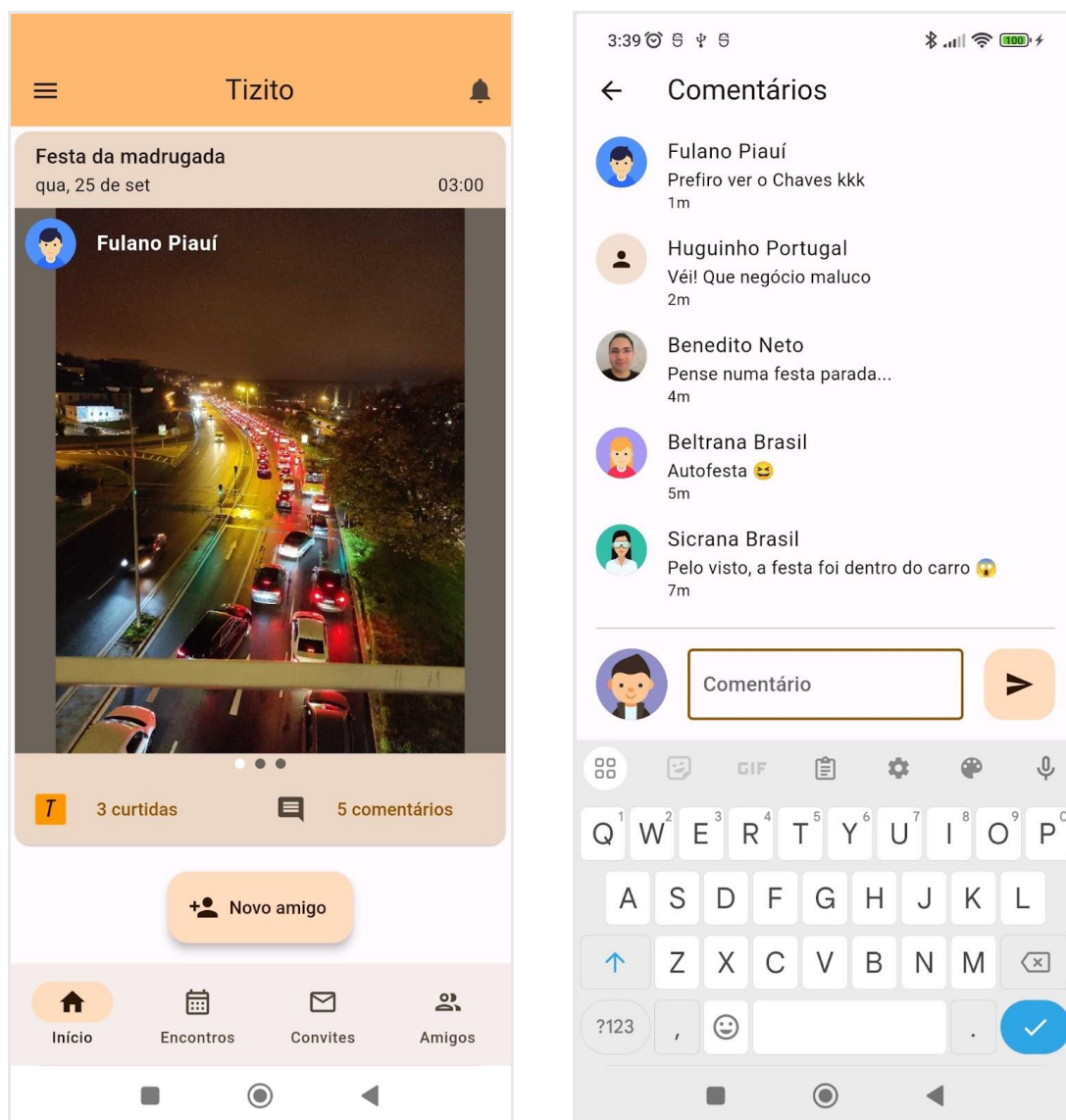


Figura 4.38 - Comentários do encontro.

4.8. Notificações

O Tizito envia uma notificação a cada utilizador envolvido nas funcionalidades apresentadas anteriormente.

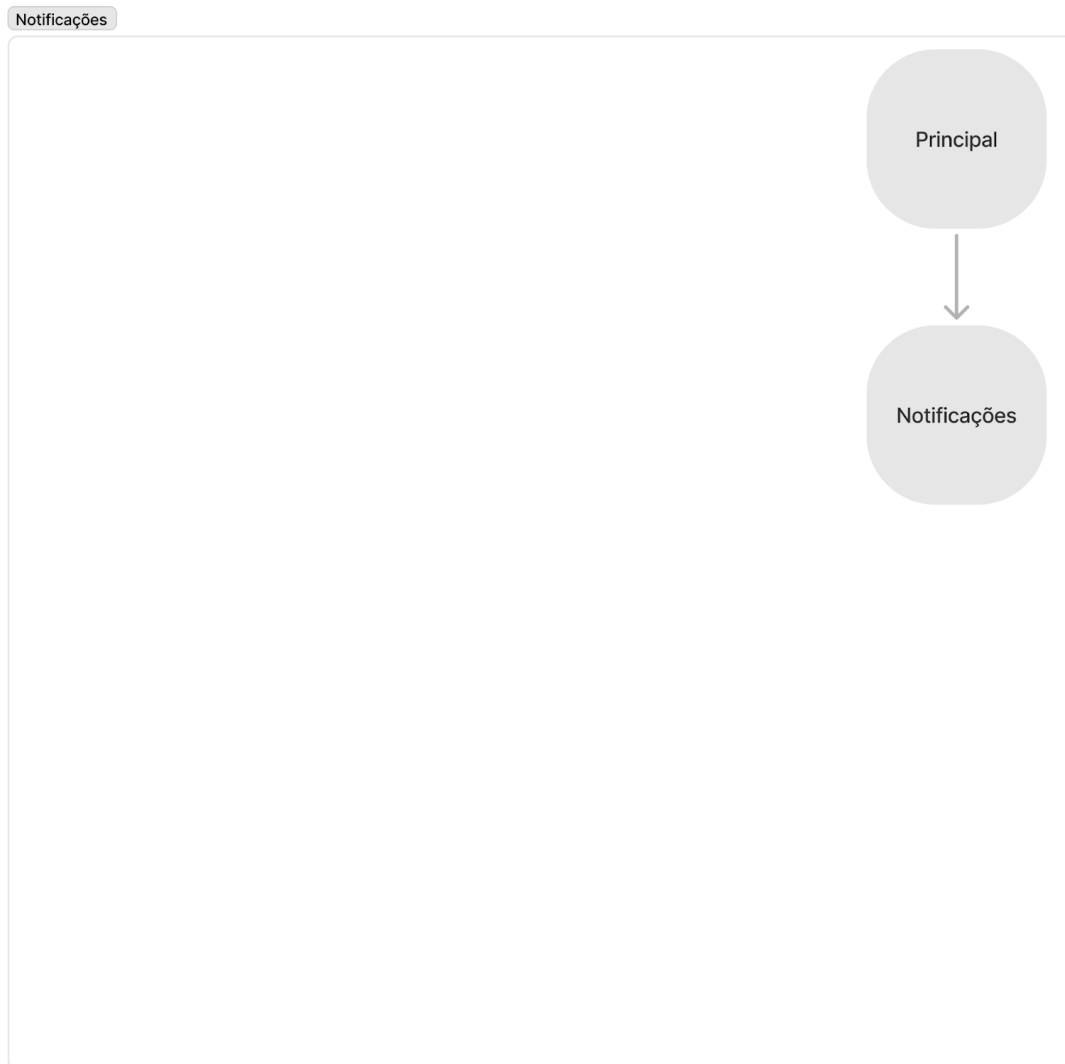


Figura 4.39 - *Sitemap* das notificações.

4.8.1. Notificação permanente

Como mencionado anteriormente, uma notificação permanente é armazenada e o utilizador pode marcá-la como lida. Por exemplo, quando o anfitrião altera o local do encontro.

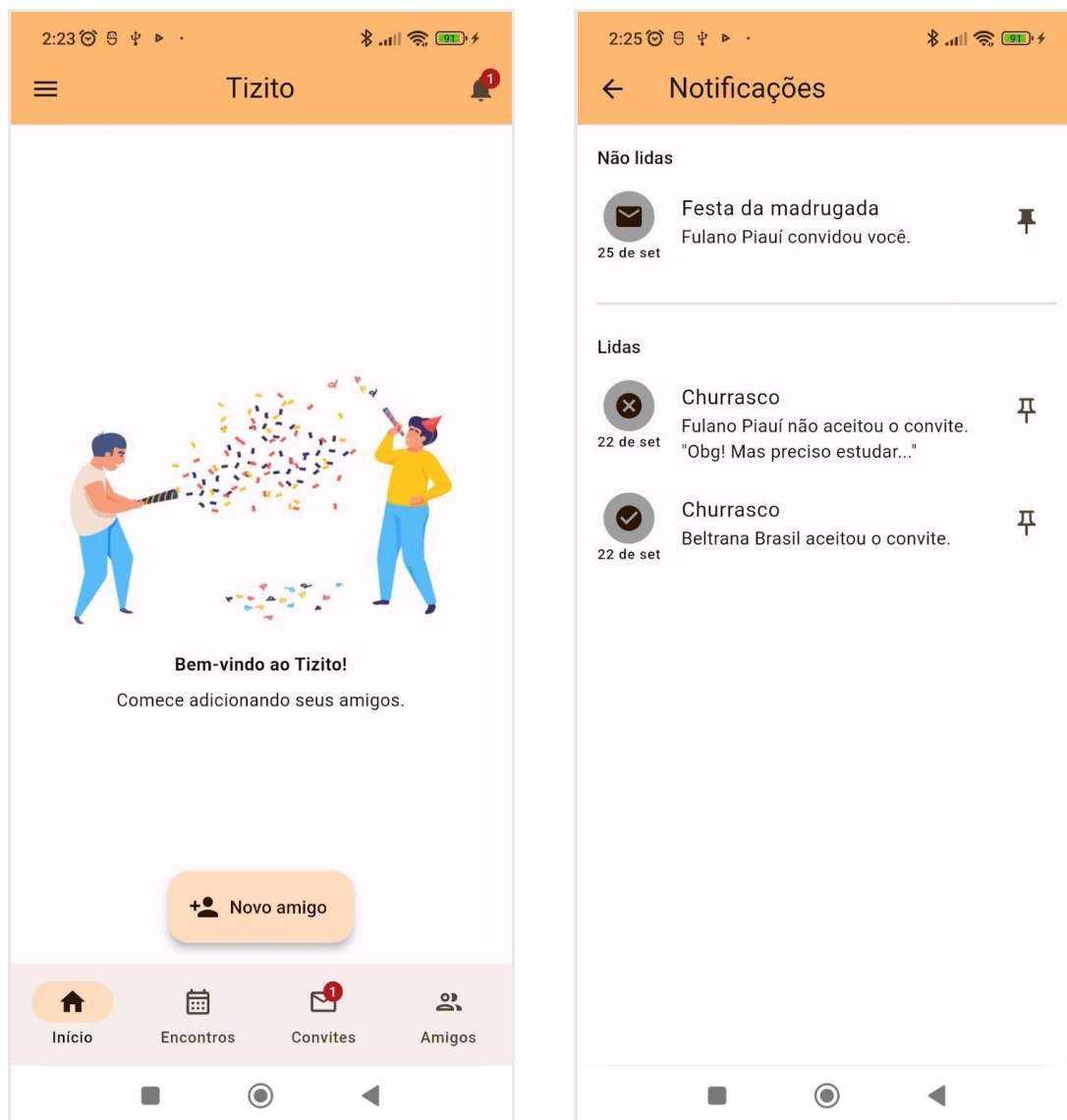


Figura 4.40 - Notificação permanente.

4.8.2. Notificação transitória

Como mencionado anteriormente, uma notificação transitória é temporária e o utilizador pode visualizá-la apenas uma vez. Por exemplo, quando o convidado publica um comentário.

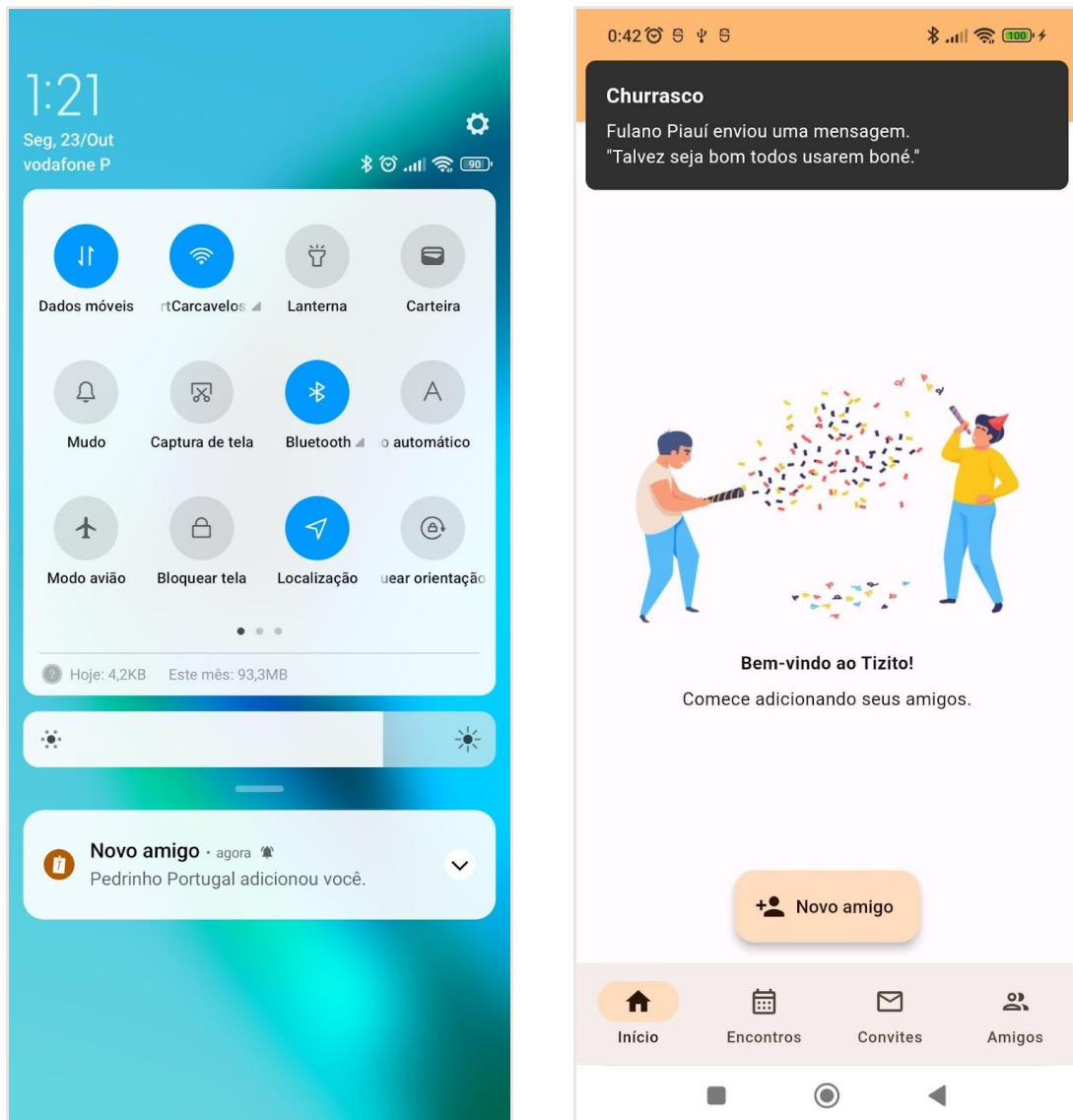


Figura 4.41 - Notificação transitória.

4.9. Auxiliares

Os ecrãs auxiliares da aplicação oferecem suporte ao utilizador e orientam a sua interação com o Tizito.

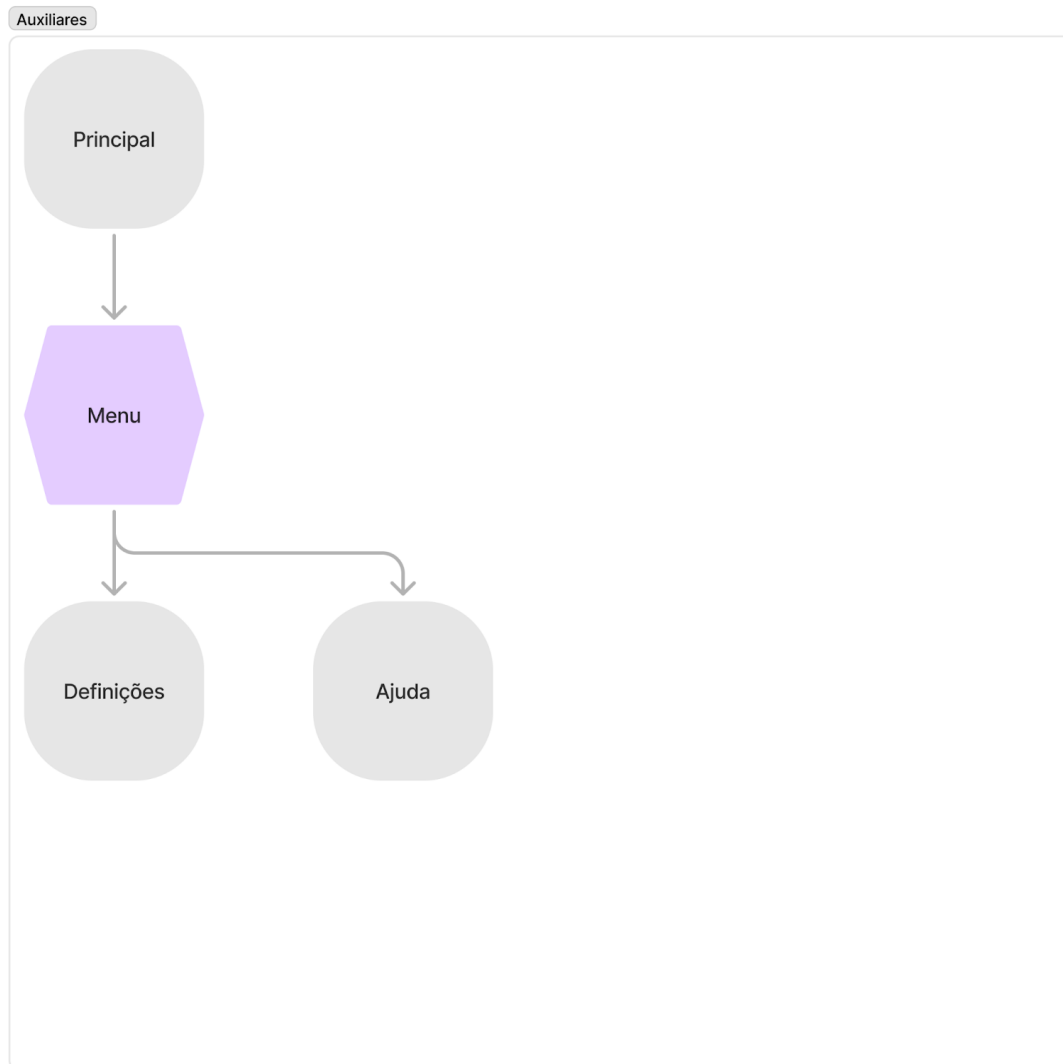


Figura 4.42 - *Sitemap* dos auxiliares.

4.9.1. Definições

Ao pressionar a opção “Definições”, o utilizador pode visualizar e alterar os dados da conta. Também pode incluir uma foto para facilitar a identificação pelos outros utilizadores. Além disso, existe o botão “Apagar conta” destacado a vermelho, que remove os dados da conta e impede o acesso ao ecrã principal do Tizito, bem como a visualização de todas as fotos, vídeos e comentários publicados pelo utilizador.

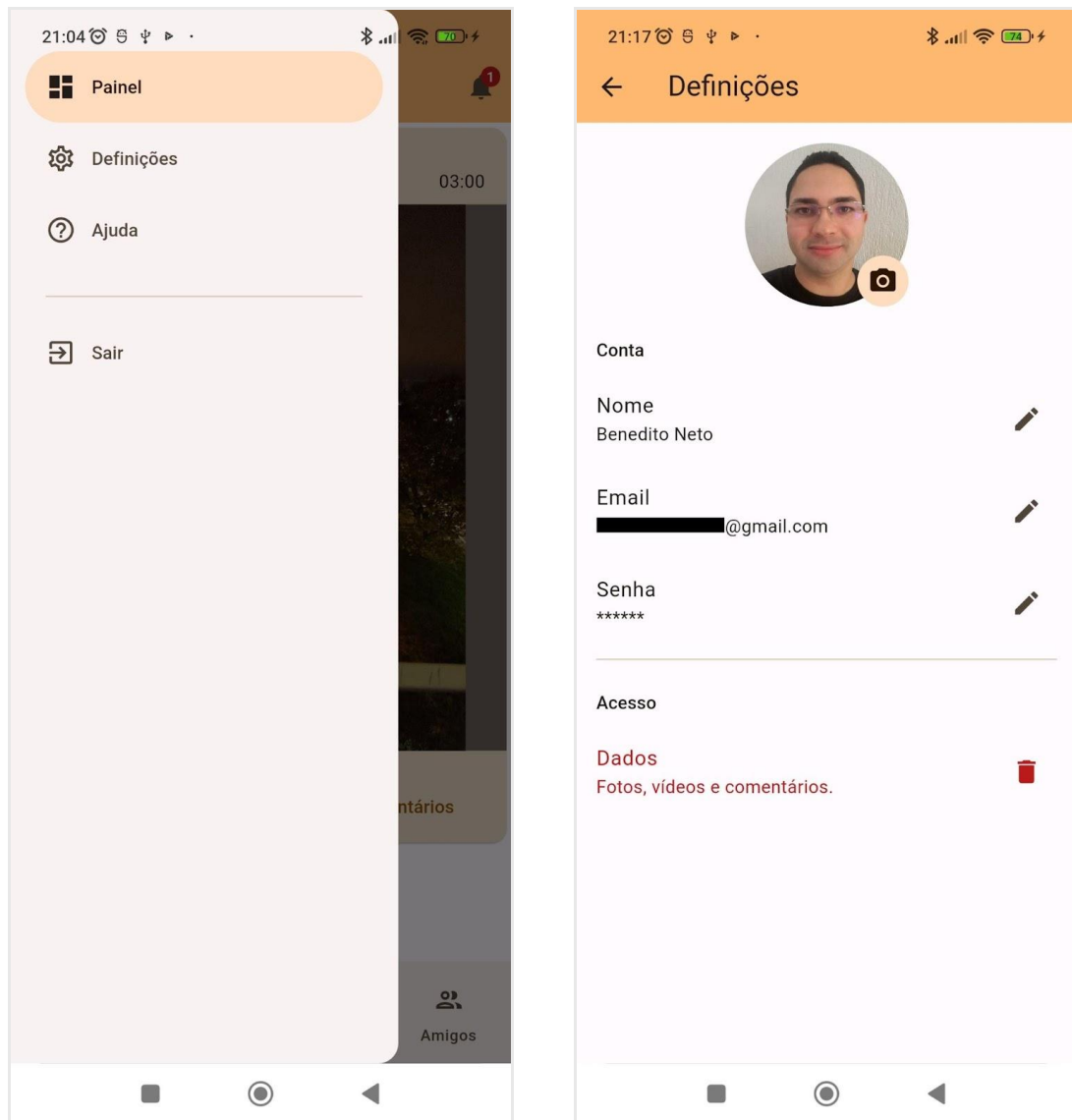


Figura 4.43 - Opção “Definições”.

4.9.2. Ajuda

Ao pressionar a opção “Ajuda”, o utilizador visualiza uma breve explicação em relação ao Tizito.

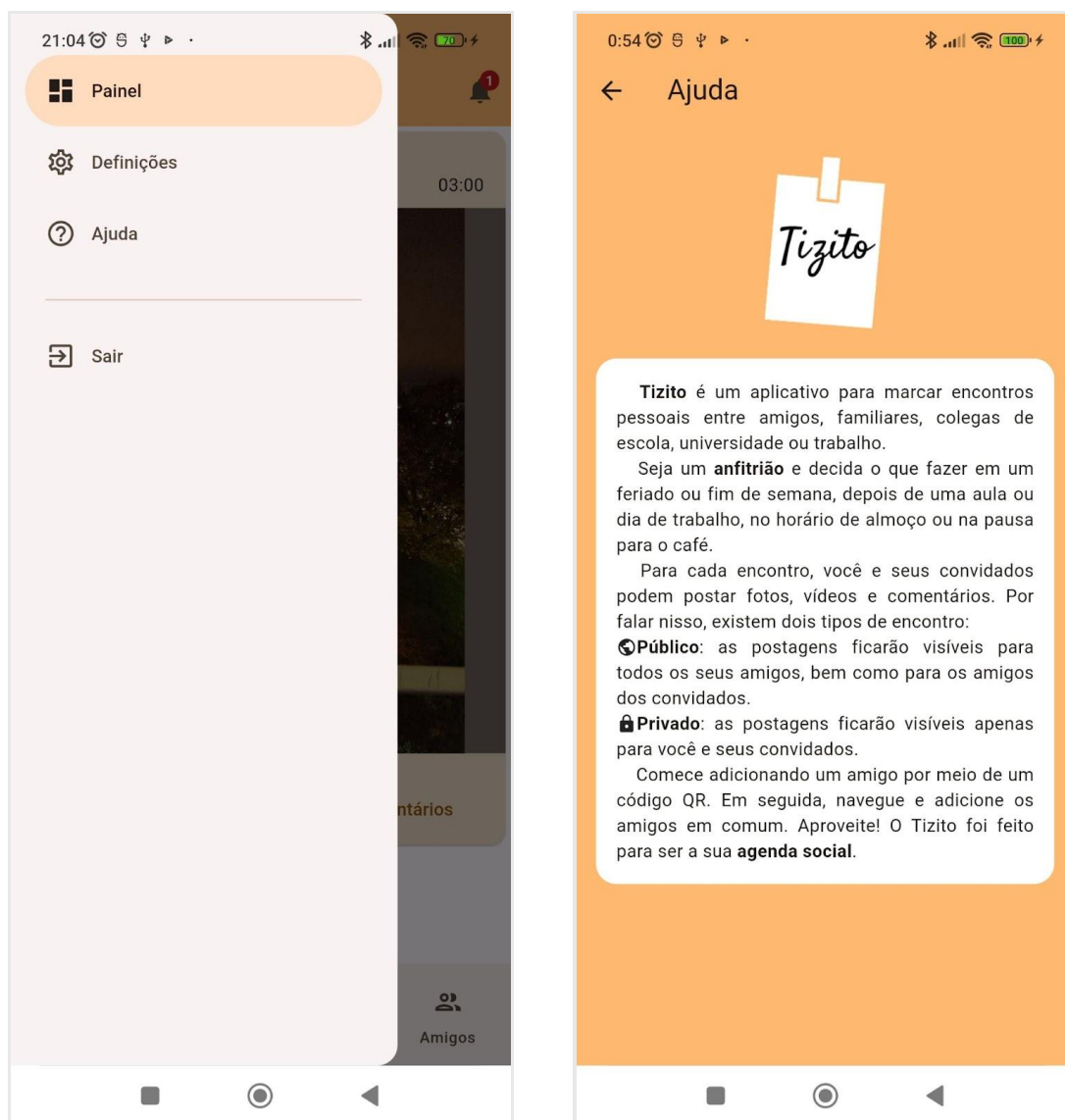


Figura 4.44 - Opção “Ajuda”.

4.9.3. Sair

Ao pressionar a opção “Sair”, o utilizador torna o acesso inativo, ou seja, é necessário informar novamente as suas credenciais para aceder ao ecrã principal do Tizito.

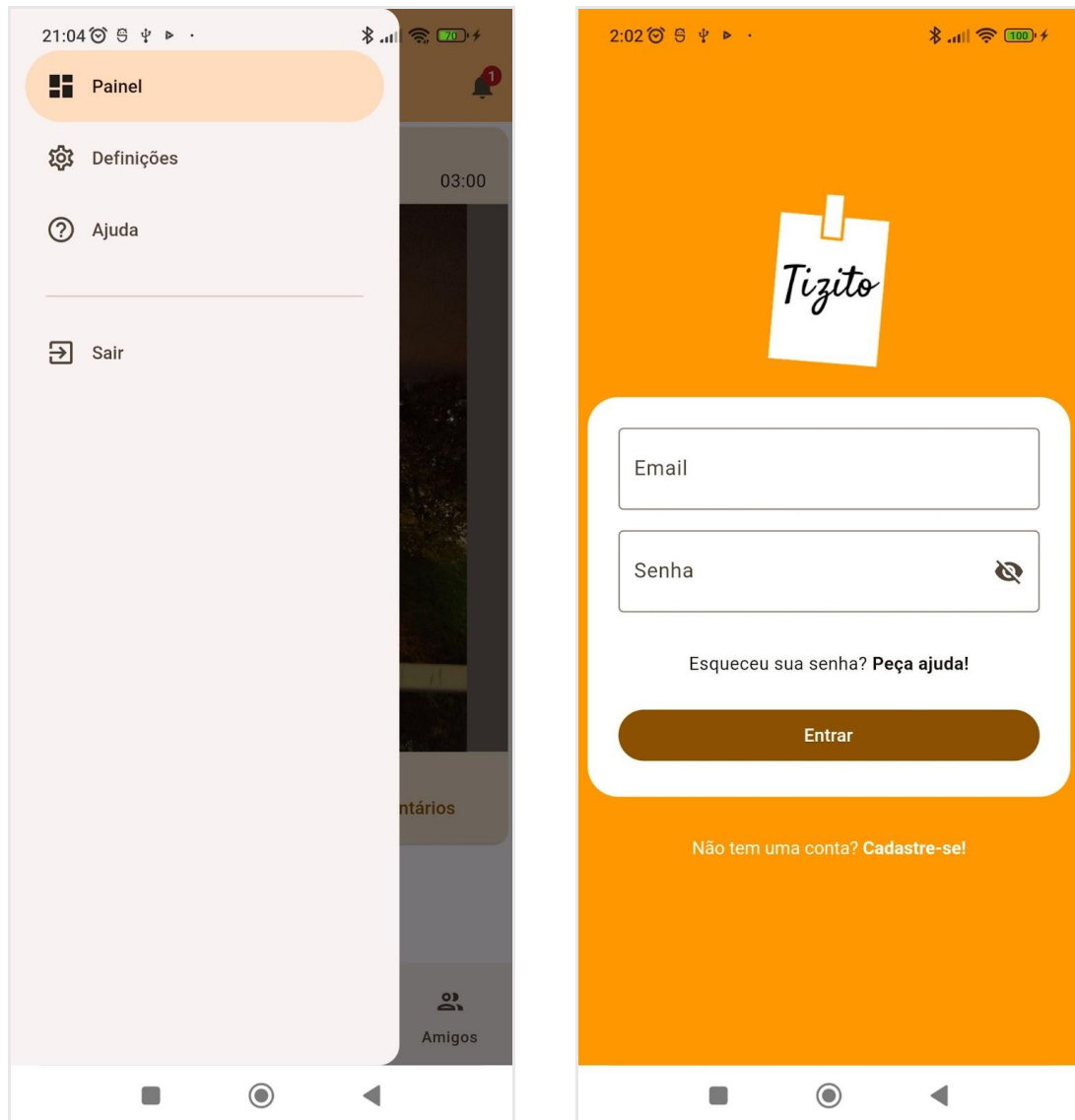


Figura 4.45 - Opção “Sair”.

Capítulo 5

Análise dos resultados e discussão

Este capítulo apresenta os testes realizados e o *feedback* inicial para orientar ajustes e melhorias no produto.

5.1. Teste supervisionado

O primeiro teste foi realizado com 5 indivíduos. Inicialmente, houve uma breve apresentação das principais funcionalidades da aplicação e instalação via USB nos respectivos *smartphones*. Em seguida, um indivíduo assumiu o papel de anfitrião, para marcar o encontro, enquanto os restantes assumiram o papel de convidados, para aceitar ou recusar o convite. Por fim, cada um publicou uma foto e um vídeo, além de comentários. O objetivo deste teste foi verificar o funcionamento geral da aplicação.



Figura 5.1 - Teste supervisionado.

O *log* da aplicação foi monitorizado durante o teste, bem como a integração da aplicação com bibliotecas de terceiros e serviços de infraestrutura, o que permitiu a validação da consistência dos dados e do comportamento da aplicação:

- **Autenticação:** registo e acesso dos indivíduos;
- **Persistência:** armazenamento dos dados gerados pelos indivíduos;
- **Publicações:** captura, *upload* e visualização de fotos e vídeos;
- **Notificações:** confirmação do envio e recebimento de mensagens para cada ação realizada pelos indivíduos.

No entanto, foi detetado um *bug* na biblioteca **flutter_google_places**, que é responsável por integrar a aplicação ao serviço Google Places API e oferece uma *interface* simples para pesquisa, *autocomplete* e seleção de locais. O *bug* ocorria sempre que o utilizador pressionava o botão “Procure aqui”, mas não realizava a pesquisa por estabelecimentos, seja ao pressionar a seta para voltar ou ao tocar na área cinza.

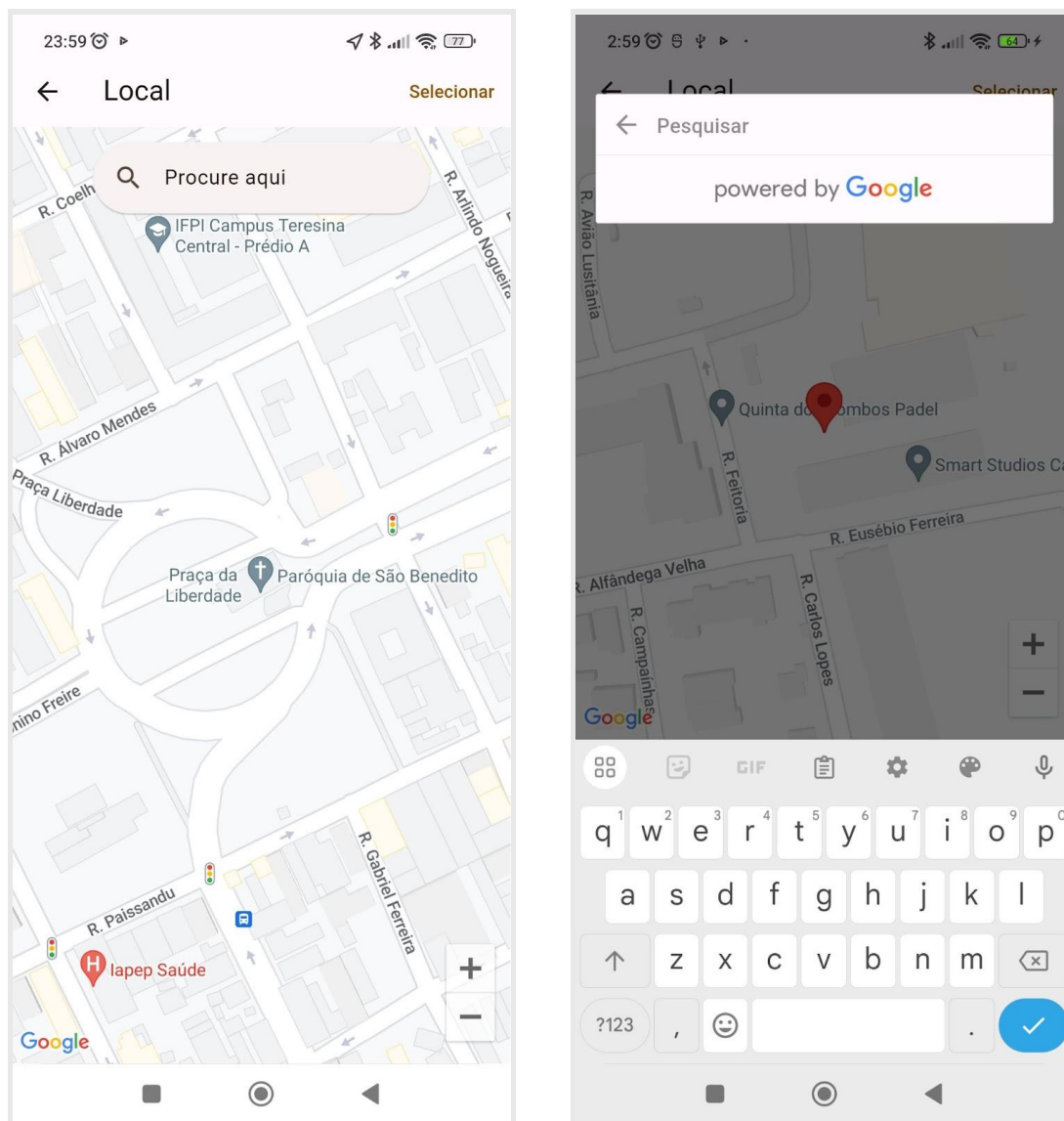


Figura 5.2 - Bug detetado.

Apesar disso, o *bug* não interfere no funcionamento da aplicação, não afeta a experiência do utilizador e não impede a biblioteca de cumprir o seu propósito. Ele sequer fica visível aos utilizadores, sendo detetado apenas pelo *log* da aplicação.

```
===== Exception caught by widgets library =====
The following _TypeError was thrown while finalizing the widget tree:
Null check operator used on a null value

When the exception was thrown, this was the stack:
#0    PlacesAutocompleteState.dispose (package:flutter_google_places/src/flutter_google_places.dart:467:14)
#1    StatefulElement.unmount (package:flutter/src/widgets/framework.dart:5297:11)
#2    _InactiveElements._unmount (package:flutter/src/widgets/framework.dart:1953:13)
#3    _InactiveElements._unmount.<anonymous closure> (package:flutter/src/widgets/framework.dart:1951:7)
#4    ComponentElement.visitChildren (package:flutter/src/widgets/framework.dart:5138:14)
#5    _InactiveElements._unmount (package:flutter/src/widgets/framework.dart:1949:13)
#6    _InactiveElements._unmount.<anonymous closure> (package:flutter/src/widgets/framework.dart:1951:7)
#7    ComponentElement.visitChildren (package:flutter/src/widgets/framework.dart:5138:14)
[112 more...]
=====
```

Figura 5.3 - Log da aplicação.

Após analisar o código da biblioteca, foi identificada uma variável *nullable* chamada “*_debounce*”. Isso significa que a variável permite valores dos tipos *Timer* e *Null*.

```
flutter_google_places.dart x
387 abstract class PlacesAutocompleteState extends State<PlacesAutocompleteWidget> {
388   TextEditingController? _queryTextController;
389   PlacesAutocompleteResponse? _response;
390   GoogleMapsPlaces? _places;
391   late bool _searching;
392   Timer? _debounce;
```

Figura 5.4 - Variável *nullable*.

Quando a caixa de pesquisa não é preenchida, essa variável tem um valor do tipo *Null*. Ao pressionar a seta para voltar ou ao tocar na área cinza, a caixa de pesquisa desaparece e a biblioteca executa um método chamado “*dispose*”.

```
flutter_google_places.dart x
462 @override
463 void dispose() {
464   super.dispose();
465   _places!.dispose();
466   _debounce!.cancel();
467   _queryBehavior.close();
468   _queryTextController!.removeListener(_onQueryChange);
469 }
470 }
```

Figura 5.5 - Método *dispose*.

Nesse momento, a biblioteca garante que a variável “*_debounce*” tem um valor do tipo *Timer*, o que não é verdadeiro. Portanto, a exceção “*Null check operator used on a null value*” é lançada. Em resumo, um possível ajuste seria usar uma versão da biblioteca com a correção do *bug*, mas o Tizito já usa a versão mais recente. Substituí-la pode ser um trabalho futuro, pois não impossibilita o lançamento do produto no mercado.

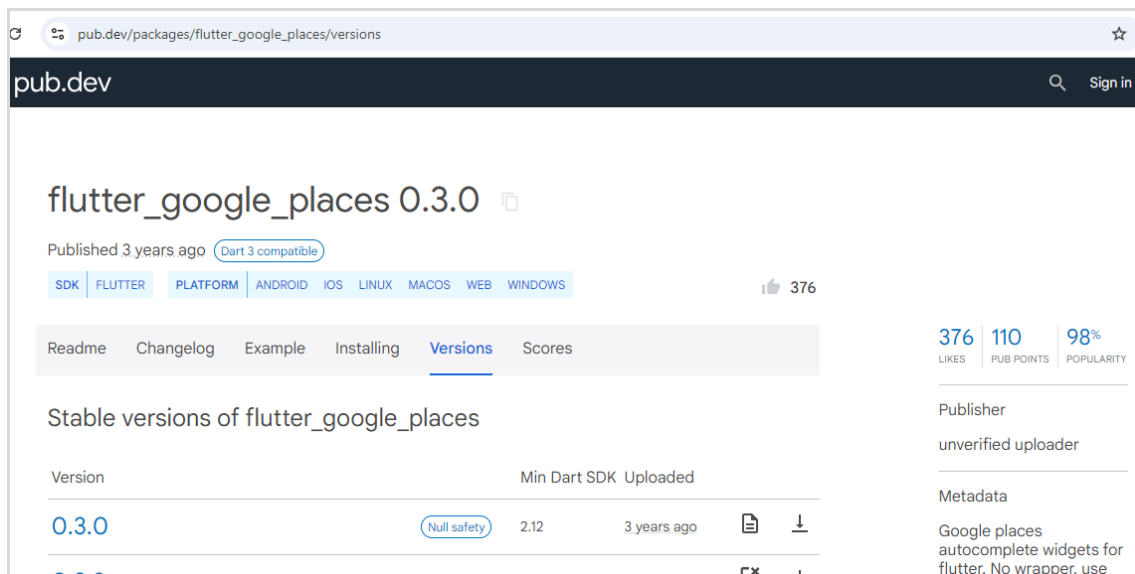


Figura 5.6 - Biblioteca flutter_google_places.

5.2. Teste não supervisionado

O segundo teste foi realizado com os 50 indivíduos que responderam à pergunta: “Você utilizaria uma aplicação para encontrar seus amigos e guardar as recordações desse encontro (fotos, vídeos e comentários)?”. Devido à abrangência e facilidade de publicação, foi gerado um executável compatível com o sistema operativo Android, sendo disponibilizado e divulgado pela ferramenta Google Play (<https://play.google.com/console>).

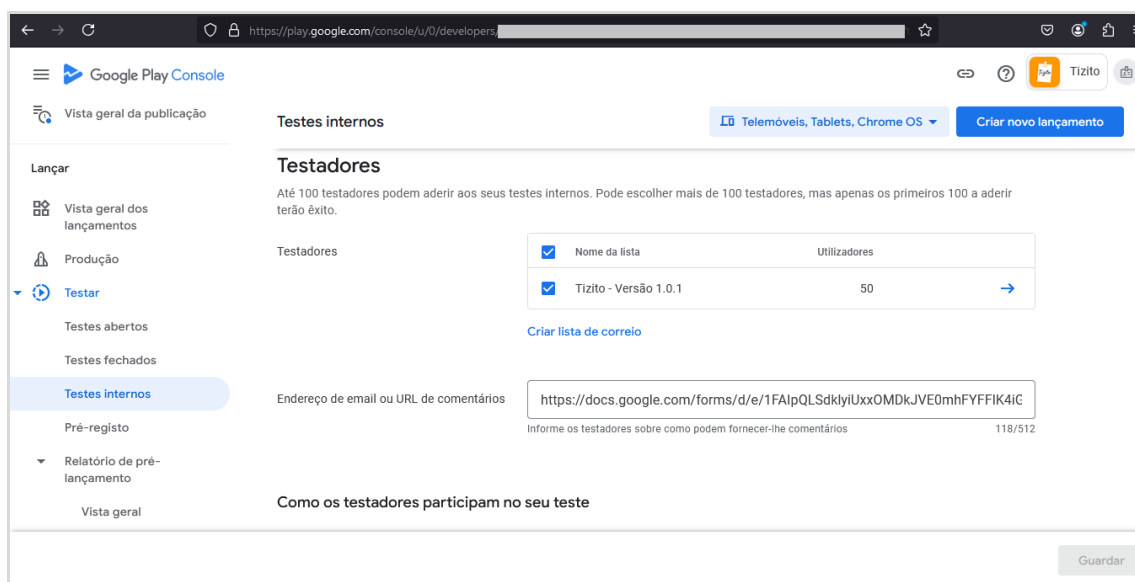


Figura 5.7 - Google Play Console.

Ao contrário do primeiro teste, não houve uma apresentação da aplicação, nem instalação nos respetivos *smartphones*. O teste teve a duração de 1 mês e não foi supervisionado. O objetivo deste teste foi analisar os dados gerados e verificar se a aplicação seria capaz de guiar os indivíduos por suas funcionalidades.

Dos 50 indivíduos que receberam o *link* para instalar e testar o Tizito, 38 registaram-se. Esse número é inferior aos 44 que demonstraram disposição em utilizar a aplicação proposta. Embora os motivos para não efetuarem o registo sejam desconhecidos, é possível que estejam relacionados à insegurança causada por um teste não supervisionado, dificuldade de instalação, incompatibilidade com o sistema operativo iOS, etc.

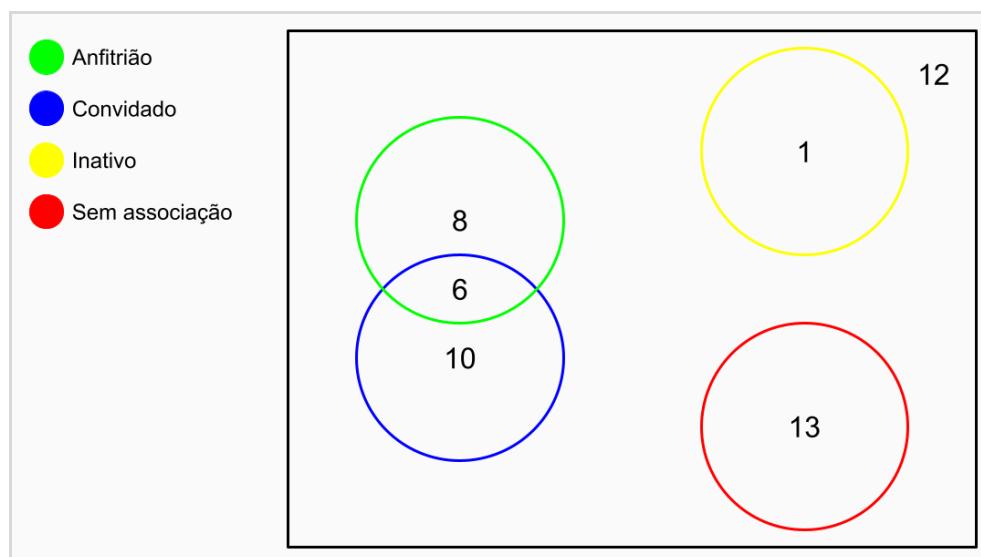


Figura 5.8 - Utilizadores registados.

Dos 38 indivíduos que se registaram, é importante destacar que 1 preferiu apagar a conta. Embora os motivos para essa desistência sejam desconhecidos, é possível que estejam relacionados à necessidade de adicionar o seu primeiro amigo para marcar o primeiro encontro. Como mencionado anteriormente, o único meio para isso é digitalizar o código QR de outro utilizador da aplicação ou permitir que o seu próprio código QR seja digitalizado.

O mesmo pode ser dito em relação aos 13 indivíduos que permaneceram registados, mas não estiveram associados a nenhum encontro, seja como anfitrião ou convidado. Esses números revelam que é preciso criar novos mecanismos para que os utilizadores possam adicionar novos amigos. Por exemplo, gerando um *link* com prazo limite para ser partilhado com outros utilizadores da aplicação. Logo, esse é o primeiro ponto de melhoria para a aplicação. Por fim, apenas 14 indivíduos foram responsáveis por 33 encontros registados num mês, entre públicos ou privados, marcados ou desmarcados.

Tabela 5.1 – Encontros registados.

Públicos		Privados	
1		32	
Marcados	Desmarcados	Marcados	Desmarcados
1	0	28	4

É notável que apenas um encontro público tenha sido marcado. Além disso, o convidado recusou o convite. É claro que a aplicação ainda está no estágio inicial, sendo utilizada por um grupo pequeno de utilizadores, mas esse dado levanta uma questão relevante: os encontros públicos realmente fazem parte do produto? Talvez o foco do Tizito seja marcar apenas encontros privados. No entanto, é necessário que a aplicação seja utilizada por mais utilizadores para avaliar se isso é uma tendência ou uma simples coincidência.

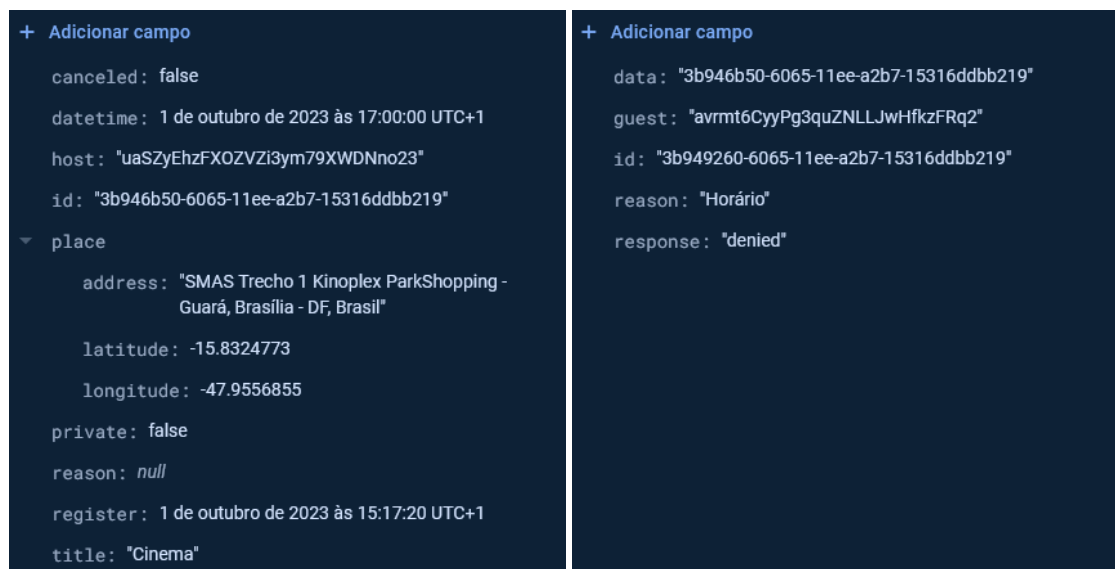


Figura 5.9 - Único encontro público.

O Tizito permite a marcação de encontros pessoais entre amigos. No entanto, os 33 encontros resultaram em apenas 49 convites (37 aceitos, 1 recusado e 11 sem resposta). Isso equivale a menos de um convidado e meio para cada encontro. Essa média reforça que é preciso criar novos mecanismos para que os utilizadores possam adicionar novos amigos. Desse modo, espera-se que um encontro tenha mais de 2 pessoas, e não apenas o anfitrião e um único convidado. Por exemplo, os grupos observados para o entendimento do problema têm uma média de 5 pessoas. Isso também reflete nos números associados aos encontros.

Tabela 5.2 – Números associados aos encontros.

Mensagens	Fotos	Vídeos	Comentários	Gostos
7	9	2	15	0

Para os 29 encontros marcados, 7 mensagens de texto foram enviadas. Esse número pode indicar que o Tizito realmente oferece maior objetividade na marcação de encontros pessoais entre amigos, reduzindo a troca de mensagens para esse fim. Por outro lado, ele pode indicar que os utilizadores não identificaram que o Tizito permite o envio instantâneo de mensagens de texto. Esse dado levanta uma questão relevante: é necessário um Teste A/B para o envio de mensagens? Esse teste implica oferecer duas versões diferentes do produto para diferentes utilizadores e ver qual delas tem melhor resultado (Teixeira, 2017).

Como mencionado anteriormente, o anfitrião tem permissão para publicar uma única foto e um único vídeo em cada encontro. O mesmo é permitido para cada convidado que aceitou o convite. Além de promover um espírito de colaboração, essa abordagem tem a intenção de oferecer algo novo para atender à sugestão do indivíduo que apresentou uma condição para utilizar a aplicação. Publicar fotos é mais comum do que publicar vídeos. Ainda assim, esse baixo número de fotos indica que a abordagem do Tizito pode não ter sido bem sucedida, pois existem encontros em que uma única pessoa tira todas as fotos. Portanto, remover esse limite para fotos e vídeos é outro ponto de melhoria para a aplicação.

Ao contrário das fotos e vídeos, não existe limite para os comentários do anfitrião e seus convidados. No entanto, os comentários são associados apenas ao encontro como um todo. Expandir essa funcionalidade para permitir comentários em cada foto e vídeo pode incentivar a interação entre os utilizadores e aumentar o número de comentários. Permitir que um comentário receba comentários pode aumentar ainda mais o número de comentários. Portanto, esse é mais um ponto de melhoria para a aplicação.

Para finalizar, é notável que não existem gostos. Essa é uma funcionalidade presente apenas nos encontros públicos. Porém, houve apenas um encontro público e o convite foi recusado. Portanto, não é possível realizar uma análise em relação ao botão “Gostar” do Tizito para verificar se o *ícone* reforça realmente o reconhecimento da marca. Os indivíduos sequer souberam da existência dele. Mais uma vez, é necessário que a aplicação seja utilizada por mais utilizadores.

5.3. Sugestões e críticas

Após a realização do segundo teste, os indivíduos foram incentivados a partilhar suas opiniões para a melhoria da aplicação, por meio de sugestões e críticas.

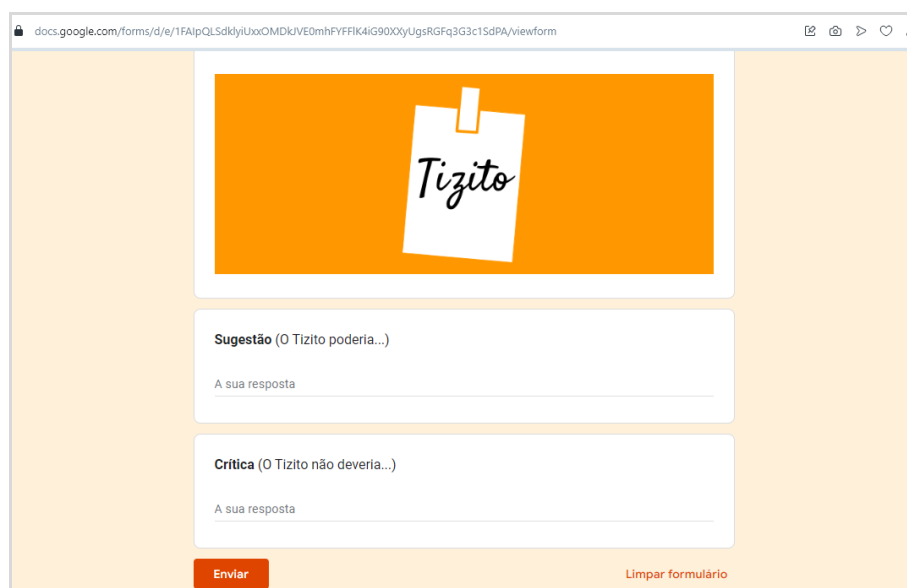


Figura 5.10 - Formulário para sugestões e críticas.

A seguir, as sugestões são confirmadas ou confrontadas com a análise dos dados gerados.

Tabela 5.3 – Sugestões.

Sugestão	Comentário
Muito bacana!! Parabéns!! Sugestões: Poderia ter mais uma forma de adicionar amigo, adicionar sem que precise do amigo scanear o QR code. Vai que o amigo esteja em outro país! Outra sugestão, ter status diferente ou cores diferentes para encontro passado, agendado ou cancelado. Como ficam todos na mesma lista a pessoa pode se confundir se tiver muitos eventos agendados.	Essa sugestão confirma a análise anterior: é preciso criar novos mecanismos para que os utilizadores possam adicionar novos amigos. O Tizito utiliza cores distintas para diferenciar os encontros marcados que já aconteceram daqueles que ainda estão para acontecer. Os encontros desmarcados são removidos da lista de encontros. No entanto, este pode ser mais um ponto de melhoria para a aplicação: diferenciar os encontros desmarcados e mantê-los na lista de encontros.
Explicar como adicionar amigo somente com a opção de leitura de QR code no celular de quem convida. Tenho que estar necessariamente ao lado do amigo a ser convidado? Mas o leitor não é no app que está instalado?	Essa sugestão demonstra mais um ponto de melhoria para a aplicação: o Tizito deve ser capaz de comunicar eficientemente as formas de adicionar um amigo.
Mostrar os encontros já na primeira página. Quando não há nada, está certo abrir nos amigos. Mas, depois que já tenho encontros agendados, acho que é melhor abrir nos encontros.	A primeira lista do ecrã principal da aplicação reúne os encontros públicos dos amigos do utilizador. Se for provado que o foco do Tizito é marcar apenas encontros privados, essa lista terá poucos ou nenhum encontro para o utilizador gostar ou comentar. Assim, o Tizito pode ser alterado e os encontros públicos deixam de existir, ou seja, o ecrã principal da aplicação terá apenas três listas: Encontros, Convites e Amigos.
Whatsapp também não possui um mecanismo de notificação para recordar os participantes meses ou anos depois daquele encontro. O Facebook faz esse tipo de notificação	Essa sugestão é uma ideia de funcionalidade. Por exemplo, o Tizito poderia notificar os utilizadores quando um encontro completar um ano e incentivá-los a marcar um novo.

Ser mesmo uma app futura e real, pois iria ajudar imenso na diminuição da criação de grupos do whatsapp e serviria como um bom calendário para encontros com amigos	Essa sugestão reflete o problema que deu origem ao Tizito e incentiva o seu lançamento no mercado.
Pra concluir seu trabalho coloca a importância da pesquisa para a sociedade, para você como pesquisador e para a área de estudo engenharia de software	Essa sugestão vai além de partilhar uma opinião sobre a aplicação e propõe uma ampliação do objetivo deste trabalho.
Receber notificações quando alguém te adiciona como amigo. Permitir partilhar um código (à semelhança do instagram), como alternativa ao código qr	Para cada ação, o Tizito notifica os respetivos utilizadores. No entanto, adicionar um amigo é um exemplo de notificação transitória. Isso significa que a notificação existe apenas durante a entrega.
	Essa sugestão também confirma que é preciso criar novos mecanismos para que os utilizadores possam adicionar novos amigos.
Rifa - ter a tabela dos números ser atualizada de forma automática	Essa sugestão é uma ideia de funcionalidade. Rifa é o sorteio de objetos por meio de bilhetes numerados. Por exemplo, o Tizito poderia produzir rifas para o entretenimento dos amigos durante um encontro.
Sorteio de amigo oculto, lista de presente	Essa sugestão é uma ideia de funcionalidade. “Amigo Oculto” ou “Amigo Secreto” é uma prática comum nalgumas culturas, na qual um amigo deve comprar um presente para outro. Por exemplo, o Tizito poderia indicar qual o amigo que receberá o presente antes do encontro acontecer.
Tradutor de português para outras línguas	Essa sugestão é um requisito funcional. A internacionalização do Tizito é um exemplo de trabalho futuro.
Mostrar eventos que meus amigos irão	Essa sugestão é uma ideia de funcionalidade. Se for provado que o foco do Tizito é marcar apenas encontros privados, essa sugestão pode não ser bem-aceita pelos utilizadores.

A seguir, as críticas são confirmadas ou confrontadas com a análise dos dados gerados.

Tabela 5.4 – Críticas.

Crítica	Comentário
Limitar o convite de amigos com leitura de QR code. A não ser que o nicho seja que você sempre encontre um amigo presencialmente para convidá-lo ao tizito.	Essa crítica também confirma que é preciso criar novos mecanismos para que os utilizadores possam adicionar novos amigos.
Falta maior interação entre anfitrião e convidados, mensagens, conversas	Essa crítica confirma a análise anterior: o baixo número de mensagens pode indicar que os utilizadores não identificaram que o Tizito permite o envio instantâneo de mensagens de texto.
	Essa crítica confirma a análise anterior: permitir comentários em cada foto e vídeo pode incentivar a interação entre os utilizadores e aumentar o número de comentários.
está ótimo, App leve	Na verdade, essa crítica é um elogio. Em formulários futuros, um campo para elogios será adicionado.
Nenhuma crítica	Na verdade, essa crítica é um elogio. Em formulários futuros, um campo para elogios será adicionado.

Capítulo 6

Conclusão

Este capítulo apresenta uma reflexão sobre os objetivos e resultados, resumindo os principais contributos e enumerando possíveis trabalhos futuros.

6.1. Considerações

Após encontrar padrões no comportamento dos utilizadores do WhatsApp, este trabalho apresentou a concepção de um novo produto de *software*, idealizado para marcar encontros pessoais entre amigos e guardar as recordações desses encontros. Trata-se de um produto de mídia social chamado Tizito, uma aplicação móvel para os sistemas operativos Android e iOS, cujo desenvolvimento utilizou a ferramenta Figma para a criação do protótipo de *interface*, a linguagem Dart para a escrita do código-fonte com o suporte do *framework* Flutter e os serviços de infraestrutura oferecidos pela plataforma Firebase. Essas tecnologias podem servir de referência para o desenvolvimento de produtos com características semelhantes.

Um teste supervisionado com 5 indivíduos permitiu a validação da consistência dos dados e do comportamento da aplicação, porém um *bug* associado à biblioteca de terceiros foi detetado. Apesar disso, o *bug* não interfere no funcionamento da aplicação, não afeta a experiência do utilizador e não impede a biblioteca de cumprir o seu propósito. Substituí-la pode ser considerado um ajuste e trabalho futuro, pois não impossibilita o lançamento do produto no mercado. As bibliotecas foram escolhidas por sua simplicidade, aceitação na comunidade e disponibilidade no repositório oficial de pacotes para aplicações Dart e Flutter.

Um teste não supervisionado com 50 indivíduos mostrou pontos de melhoria no produto antes do seu lançamento:

- Criar novos mecanismos para que o utilizador possa adicionar um amigo;
- Comunicar de forma eficiente as formas de adicionar um amigo;
- Permitir a publicação de mais fotos, vídeos e comentários.

Esses pontos de melhoria foram identificados a partir da análise dos dados gerados pela aplicação, além das sugestões e críticas expressas pelos indivíduos. Esse teste também serviu para levantar questões relevantes:

- Encontros públicos realmente fazem parte do produto?
- Um Teste A/B é necessário para o envio de mensagens?

É possível que o foco do Tizito seja apenas encontros privados; é possível que o botão para enviar mensagens não esteja bem posicionado na *interface*. De qualquer forma, é necessário que a aplicação seja utilizada por mais utilizadores para encontrar as respostas corretas para essas questões. Isso também se aplica à funcionalidade que não foi possível analisar: o botão “Gostar” do Tizito.

Certamente, as histórias do utilizador ajudaram na definição dos requisitos, a prototipagem ajudou na validação do produto antes de ser construído, enquanto o *framework* e as bibliotecas da comunidade ajudaram na construção da aplicação. No entanto, o fato de uma única pessoa ser responsável por todas as etapas do desenvolvimento do produto pode ter sido a razão para essas e outras questões que ainda podem surgir. O ideal seria ter uma pessoa para cada etapa e juntas formar uma equipa real, atuando como parceiros, cada um com sua especialidade e responsabilidade, colaborando para produzir o melhor produto possível.

6.2. Trabalhos futuros

O Tizito tentou promover um espírito de colaboração e oferecer algo novo com uma abordagem na qual tanto o anfitrião como os convidados podem publicar uma única foto e um único vídeo em cada encontro. No entanto, o baixo número de fotos e vídeos indica que essa abordagem pode não ter sido bem sucedida, pois existem encontros em que uma única pessoa tira todas as fotos. Portanto, o objetivo de oferecer algo novo continua a ser um item na lista de trabalhos futuros: criar um mecanismo para tomada de decisões sobre os detalhes de eventos sociais, internacionalizar a aplicação, aumentar a cobertura de testes automatizados, avaliar a usabilidade, identificar uma forma de produzir receita e lançar o produto no mercado.

Bibliografia

Bitencourt, J. (2022). O guia de Dart: Fundamentos, prática, conceitos avançados e tudo mais. Casa do Código.

Gomes, A. F. (2013). Agile: Desenvolvimento de software com entregas frequentes e foco no valor de negócio. Casa do Código.

Kemp, S. (2021, fevereiro 11). Digital in Portugal: All the statistics you need in 2021 — DataReportal – global digital insights. DataReportal – Global Digital Insights.
<https://datareportal.com/reports/digital-2021-portugal>

Lisboa, A. (2023, novembro 8). Figma: construindo o layout do seu site mobile. Alura.
<https://www.alura.com.br/curso-online-figma-construindo-layout-primeiro-site-mobile>

Marinho, L. (2021). Flutter Framework: Desenvolva aplicações móveis no Dart Side! Casa do Código.

Mc Mahon, C. (2021). A psicologia da mídia social. Edgard Blücher Ltda.

Ribeiro, A. (2023, janeiro 31). Entendendo o Firebase e suas principais funcionalidades. Alura.
<https://www.alura.com.br/artigos/entendendo-firebase-principais-funcionalidades>

Rodrigues, J. (2023, setembro 26). Flutter: escolhendo uma arquitetura para o seu projeto. Alura. <https://www.alura.com.br/artigos/flutter-arquitetura-seu-projeto>

Sabbagh, R. (2013). Scrum: Gestão ágil para projetos de sucesso. Casa do Código.

Teixeira, F. (2017). Introdução e boas práticas em UX Design. Casa do Código.

Torres, J. (2017). Guia da startup: Como startups e empresas estabelecidas podem criar produtos de software rentáveis (Atualizada). Casa do Código.

Torres, J. (2020). Gestão de produtos: Como aumentar as chances de sucesso do seu software (3a). Casa do Código.

Villain, M. (2023, junho 2). UI Design: melhorando a experiência de um app com UI Patterns. Alura. <https://www.alura.com.br/curso-online-ui-design-experiencia-app-ui-patterns>

Zambelli, R. (2022, maio 11). O que é 5W2H e como aplicá-lo na gestão de processos? Blog | Checklist Fácil. <https://blog-pt.checklistfacil.com/5w2h/>

