



**FACULDADE DE TECNOLOGIA DE GUARULHOS
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E
DESENVOLVIMENTO DE SISTEMAS**

**LUCAS BEZERRA DA SILVA
NETIVAN FLORENTINO DA ROCHA**

**HORNERO: APLICATIVO MOBILE PARA DOAÇÃO E/OU VENDA DE
RESÍDUOS DE MATERIAIS DE CONSTRUÇÃO.**

**Guarulhos
2023**

LUCAS BEZERRA DA SILVA
NETIVAN FLORENTINO DA ROCHA

**HORNERO: APLICATIVO MOBILE PARA DOAÇÃO E/OU VENDA DE
RESÍDUOS DE MATERIAIS DE CONSTRUÇÃO.**

Trabalho de Graduação do Curso de Análise e Desenvolvimento de Sistemas, apresentado como requisito parcial para obtenção do Título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Me. Rodrigo Vieira Campos.

Guarulhos
2023

**LUCAS BEZERRA DA SILVA
NETIVAN FLORENTINO DA ROCHA**

**HORNERO: APLICATIVO MOBILE PARA DOAÇÃO E/OU VENDA DE
RESÍDUOS DE MATERIAIS DE CONSTRUÇÃO.**

Trabalho de Graduação apresentado ao Curso de Análise e Desenvolvimento de Sistemas como requisito parcial para obtenção do **Título de Tecnólogo em Análise e Desenvolvimento de Sistemas.**

Banca Examinadora

Orientador: _____

Prof. Me Rodrigo Vieira Campos
Fatec Guarulhos

Banca: _____

Titulação + nome completo
Fatec Guarulhos

Banca: _____

Titulação + nome completo
Fatec Guarulhos

Guarulhos, 27/11/2023

RESUMO

SILVA, Lucas Bezerra da; ROCHA, Netivan Florentino da. **Hornero: aplicativo mobile para doação e/ou venda de resíduos de materiais de construção**. 2023. 44 p. Trabalho de Conclusão de Curso - Faculdade de Tecnologia de Guarulhos, Guarulhos.

A indústria da construção civil tem se desenvolvido amplamente, de modo que, atualmente, presenciamos considerável preocupação entre as pessoas por construírem ou reformarem espaços para que se tornem não apenas habitáveis, mas transmitam bem-estar e conforto. Ao tratarmos mais especificamente do nicho de reformas, é possível notar que muitos resíduos de materiais são gerados e que, por vezes, tais resíduos poderiam ser reutilizados ao invés de descartados, pois, há ocasiões em que, tal descarte é feito inadequadamente, gerando impactos no meio ambiente. Neste sentido, este estudo explorou os meios para conceber um aplicativo mobile para que os seus usuários possam realizar a doação ou venda de resíduos de materiais de construção, provenientes de obras ou reformas, que estão aptos para uso. A aplicação foi desenvolvida para Android e iOS utilizando a linguagem Dart e permite aos usuários realizarem negociações entre anunciantes e consumidores.

Palavras-chave: Aplicativo mobile, construção civil, reuso, sustentabilidade.

ABSTRACT

The construction industry has developed extensively, so that, currently, we see considerable concern among people about building or renovating spaces so that they become not only habitable, but convey well-being and comfort. When dealing more specifically with the renovation niche, it is possible to notice that a lot of material waste is generated and that, sometimes, such waste could be reused instead of discarded, as there are occasions when such disposal is done inappropriately, generating impacts in the environment. In this sense, this study explored the means of designing a mobile application so that its users can donate or sell waste construction materials, from works or renovations, that are suitable for use. The application was developed for Android and iOS using the Dart language and allows users to carry out negotiations between advertisers and consumers.

Keywords: Mobile application, construction industry, reuse, sustainability.

LISTA DE ABREVIATURAS

ABRELPE - Associação Brasileira de Empresas de Limpeza Pública e Resíduos Especiais

CONAMA - Conselho Nacional do Meio Ambiente

ECMA - *European Computer Manufacturers Association*

HTML - *HyperText Markup Language*

IDE - *Integrated Development Environment*

IPEA - Instituto de Pesquisa Econômica Aplicada

JSX - *JavaScript Syntax Extension*

SDK - *Software Development Kit*

XML - *Extensible Markup Language*

GLOSSÁRIO

ANDROID – Sistema operacional baseado no núcleo Linux, projetado principalmente para dispositivos eletrônicos móveis (como smartphones e tablets) com tela sensível ao toque ou interface de usuário baseada na manipulação direta; desenvolvido por um consórcio de desenvolvedores conhecido como *Open Handset Alliance*, sendo o principal colaborador o Google.

DART - Linguagem de *script* voltada à web desenvolvida pela Google.

FLUTTER – Kit de desenvolvimento de interface de usuário, de código aberto, criado pela empresa Google em 2015, baseado na linguagem de programação Dart, que possibilita a criação de aplicativos compilados nativamente, para os sistemas operacionais Android, iOS, Windows, Mac, Linux, Fuchsia e Web.

GARBAGE COLLECTION – Processo usado para a automação do gerenciamento de memória. Com ele é possível recuperar uma área de memória inutilizada por um programa, o que pode evitar problemas de vazamento de memória, resultando no esgotamento da memória livre para alocação.

IONIC – Kit para desenvolvimento de código aberto completo para desenvolvimento de aplicativo móvel híbrido criado por Max Lynch, Ben Sperry e Adam Bradley da Drifty Co. em 2013.

IOS – Sistema operacional móvel da Apple Inc.

JAVA - linguagem de programação orientada a objetos desenvolvida na década de 90 por uma equipe de programadores chefiada por James Gosling, na empresa Sun Microsystems, que em 2008 foi adquirida pela empresa Oracle Corporation.

KOTLIN – Linguagem de programação multiplataforma, orientada a objetos e funcional, concisa e estaticamente tipada, desenvolvida pela JetBrains em 2011, que compila para a Máquina virtual Java e que também pode ser traduzida para a linguagem JavaScript e compilada para código nativo.

NODE.JS - *Software* de código aberto, multiplataforma, baseado no interpretador V8 do Google e que permite a execução de códigos JavaScript fora de um navegador web.

OBJECTIVE-C - Linguagem de programação reflexiva orientada a objeto que adiciona transmissão de mensagens no estilo Smalltalk para o C.

PHOTOSHOP - *Software* caracterizado como editor de imagens bidimensionais do tipo raster desenvolvido pela Adobe Systems.

REACT - Biblioteca *front-end* JavaScript de código aberto com foco em criar interfaces de usuário em páginas web.

SWIFT – Linguagem de programação desenvolvida pela Apple para desenvolvimento no iOS, macOS, watchOS, tvOS, Windows e Linux.

XCODE – Ambiente de desenvolvimento integrado e *software* livre da Apple Inc. para gerenciamento de projetos relacionados com o sistema operacional macOS.

LISTA DE FIGURAS

Figura 1 - Mapa Mental Flutter	19
Figura 2 - Roteiro de Como Levantar História de Usuários.....	28
Figura 3 - Diagrama de Casos de Uso do Aplicativo Hornero	30
Figura 4 - Protótipo da tela de login	31
Figura 5 - Protótipo da tela inicial.....	32
Figura 6 - Protótipo Visualização da tela de consulta de anúncios	33
Figura 7 - Protótipo Visualização de um anúncio	33
Figura 8 - Tela de Login	34
Figura 9 - Tela de Cadastro de Usuário	35
Figura 10 - Tela Principal	35
Figura 11 - Menu Principal	36
Figura 12 - Tela de Cadastro de Anúncio.....	37
Figura 13 - Tela de Consulta de Anúncio	38
Figura 14 - Tela Resultado Pesquisa Anúncios.....	38
Figura 15 - Tela Finalizar Uma Compra	39
Figura 16 - Lista de Compras Efetuadas.....	39
Figura 17 - Tela Troca de Mensagens (Vendedor x Comprador).....	40
Figura 18 - Visão dos Dados Persistidos no Firebase.....	41
Figura 19 - Visão De Um Anúncio Gravado no Firebase	42

LISTA DE TABELAS

Tabela 1 - Levantamento das Principais Histórias de Usuários para o Aplicativo Hornero.	28
---	----

LISTA DE GRÁFICOS

Tabela 1 - Levantamento das Principais Histórias de Usuários para o Aplicativo Hornero.	28
---	----

SUMÁRIO

1.	INTRODUÇÃO.....	12
1.1.	Justificativa	13
1.2.	Objetivos.....	14
1.2.1.	Objetivo Geral.....	14
1.2.2.	Objetivos Específicos	14
2.	FUNDAMENTAÇÃO TEÓRICA	15
2.1.	Projeto do Aplicativo Hornero	15
2.2.	Desenvolvimento Multiplataforma.....	16
2.3.	Flutter.....	17
2.4.	Dart.....	19
2.5.	Ambiente de Desenvolvimento	20
2.6.	Banco de Dados	21
2.7.	Dispositivos Móveis	23
3.	METODOLOGIA	24
4.	MATERIAIS E MÉTODOS.....	25
4.1.	Escopo do Projeto.....	25
4.2.	Metodologia de Engenharia de <i>Software</i>	25
4.2.1.	Requisitos	27
4.2.1.1.	Histórias de Usuários.....	27
4.2.1.2.	Casos de Uso	29
4.2.1.3.	Diagrama de Classes	30
4.2.1.4.	Protótipos	31
5.	RESULTADOS E DISCUSSÕES.....	34
6.	CONSIDERAÇÕES FINAIS.....	43
	REFERÊNCIAS.....	44
	APÊNDICE A – Diagrama de Classes Hornero.....	47

1. INTRODUÇÃO

Resíduos são inevitáveis nas atividades industriais. SILVA (2020) aponta que o grande crescimento populacional, desde a Revolução Industrial, demandou cada vez mais uma cadeia produtiva maior e conseqüentemente mais geradora de resíduo. Por esse motivo, desde os anos 60, organizações mundiais, governos, dentre outros, têm se mobilizado em torno do tema a fim minimizar os efeitos negativos ao meio ambiente e a futuras gerações. Por meio desse esforço nasce o conceito de desenvolvimento sustentável e eventos como a Conferência Rio-92 fazem nascer compromissos mundiais em torno da gestão e redução dos resíduos advindos da produção.

Segundo a Resolução 307/2002, do CONAMA, artigo 2º, define-se:

I – Resíduos da construção civil: são os provenientes de construções, reformas, reparos e demolições de obras de construção civil, e os resultantes da preparação e da escavação de terrenos, tais como: tijolos, blocos cerâmicos, concreto em geral, solos, rochas, metais, resinas, colas, tintas, madeiras e compensados, forros, argamassa, gesso, telhas, pavimento asfáltico, vidros, plásticos, tubulações, fiação elétrica etc., comumente chamados de entulhos de obras, caliça ou metralha;

A construção civil tem uma grande participação na geração de resíduos. Segundo o IPEA (2012), os materiais remanescentes da construção civil representam de 50% a 70% do resíduo sólido no ambiente urbano. Em 2014, o Brasil produziu 78,6 milhões de toneladas de resíduos sólidos urbanos, e constatou-se que 45 milhões de toneladas originaram na construção civil (GOMES et al., 2021). Já segundo constatou-se pela ABRELPE (2020 apud Ribeiro et al., 2021) o panorama dos resíduos sólidos neste país, no ano de 2019, levantado por aquela associação, apontava que em torno de 56% dos resíduos gerados, teve como origem as atividades de construção e de demolição. Em termos de toneladas isso representa aproximadamente, 44,5 milhões, de um total de 79 milhões de resíduos sólidos urbanos.

LOPES (2020), afirma que a disposição irregular dos resíduos pode gerar problemas ambientais e de saúde pública, além de sobrecarregar os sistemas de limpeza pública. RIBEIRO et al., (2021) constatam que a falta de comprometimento dos responsáveis por gerar e coletar tais resíduos e a falta de consciência ambiental por parte de todos podem ser as causas do aparecimento de depósitos clandestinos,

além de estarem diretamente relacionado a enchentes, assoreamento dos rios e córregos, aparecimento e aumento de vetores transmissores de doenças, etc., que acarreta na queda da qualidade da vida das pessoas

O que fazer com tamanha quantidade de resíduos e o correto manuseio desse material para que não gere impactos ambientais é a grande questão.

Com a publicação da Lei nº 12.305 de 02 de agosto de 2021 que trata da Política Nacional de Resíduos Sólidos, o gerenciamento de resíduos passa a priorizar a não geração e o controle de resíduos direto na fonte. Destaca-se ainda que esse documento apresenta princípios e importantes diretrizes voltados a regularização da gestão integrada dos resíduos quer seja no campo público como no privado (SILVA, 2020).

Esse trabalho visa contribuir para atender a premissa apontada acima da não geração de resíduos, buscando apresentar um aplicativo para dispositivos móveis que permita às pessoas disponibilizarem em uma plataforma específica, sobras de materiais utilizados, bem como, itens que foram trocados e que ainda estão em bom estado para utilização, venda ou doação.

1.1. Justificativa

Este projeto de pesquisa tem como motivação apresentar uma alternativa para diminuir os impactos negativos dos resíduos originados da construção civil. Sabe-se que a construção civil é responsável por 50 a 70% dos resíduos sólidos gerados e que esses resíduos, em sua grande parte, são descartados em terrenos baldios e/ou margens de rios e córregos, ou ficam armazenados em algum local inapropriado das residências podendo ocorrer acidentes domésticos ou até servir de alternativa para desenvolvimento de pragas, insetos e animais peçonhentos.

Nos últimos anos, o PIB da construção civil no Brasil se mostrou acima do PIB nacional o que implica em maior quantidade de resíduo gerado. Com esse ritmo, é urgente que se apresentem soluções rápidas para contornar esse problema.

Apresenta-se neste trabalho um aplicativo que pode contribuir para minimizar esse impacto. Pretende-se, em uma única plataforma, unir pessoas interessadas em adquirir produtos em quantidade customizada e pessoas interessadas em vender ou

doar produtos sobressalentes e com isso diminuir a quantidade de resíduos descartados e/ou armazenados incorretamente na construção civil.

1.2. Objetivos

1.2.1. Objetivo Geral

O objetivo geral deste trabalho é modelar, desenvolver e implementar um aplicativo móvel que funcione como uma plataforma de comercialização e/ou doação de sobras de materiais de construção, visando reduzir o desperdício e promover o reuso destes materiais.

1.2.2. Objetivos Específicos

Realizar uma revisão da literatura sobre os desafios do descarte inadequado de sobras de materiais de construção, considerando os aspectos ambientais, econômicos e sociais;

Identificar os principais atores envolvidos no setor da construção civil, desde pequenos projetos de reforma até grandes empreendimentos, e entender suas necessidades e motivações para participar de uma plataforma de reaproveitamento de materiais;

Definir os requisitos funcionais e não funcionais do aplicativo móvel, considerando aspectos técnicos, de usabilidade e segurança;

Desenvolver o aplicativo móvel de acordo com as especificações definidas, utilizando as tecnologias e ferramentas apropriadas para garantir um desempenho eficiente e uma experiência de usuário intuitiva;

Elaborar conclusões sobre a eficácia do aplicativo em alcançar os objetivos propostos, discutindo suas contribuições para a sustentabilidade ambiental, a eficiência econômica e a promoção de práticas mais responsáveis na construção civil;

Por meio da consecução desses objetivos específicos, almeja-se contribuir para o avanço do conhecimento no campo da tecnologia aplicada à sustentabilidade na construção civil, oferecendo uma solução concreta para mitigar os desafios enfrentados pelo setor em relação ao descarte de sobras de materiais.

2. FUNDAMENTAÇÃO TEÓRICA

O objetivo desta seção é apresentar os conceitos que ajudarão a compreender a proposta do projeto bem como o escopo que foi utilizado para o desenvolvimento do aplicativo.

2.1. Projeto do Aplicativo Hornero

De acordo com o que já foi explanado na introdução deste trabalho, na qual foi relatada a importância da construção civil para o país e a sua participação no PIB, vislumbrou-se a oportunidade de centralizar a solução proposta para o problema apresentado em um aplicativo específico para a construção civil.

De acordo com o TECHTUDO (2022) os oito melhores sites/app para vender e comprar produtos usados online são Mercado Livre, Enjoei, Shopee, OLX, Peguei Bode, Estante Virtual, Já Vendeu e Facebook Marketplace. Destaca-se que os mais populares, a saber, Mercado Livre, Enjoei e OLX não tem algo específico para este fim, ou seja, disponibilizar um espaço na plataforma para fomentar a doação e/ou venda de produtos de construção civil que sobraram de obras e reformas. Ao pesquisar sobre um site/aplicativo específico para esse fim, constatou-se a existência do site www.sobradaobra.com.br. Em que pese o site informar que há um aplicativo disponível para Android e IOS, não foi encontrado nas respectivas lojas, o *app*.

Não se pretende desenvolver um aplicativo que tenha todas as funcionalidades de um sistema de venda entre os usuários, tendo em vista a complexidade de aplicativo para *Marketplace*. Conforme escopo abaixo, buscou-se trazer para esse projeto as funcionalidades que permitam ao usuário seguir um fluxo de utilização do início ao fim, ou seja, desde o anúncio até uma negociação, porém não é objeto dessa solução gerenciar transações financeiras e logística entre produtos adquiridos pelos usuários.

2.2. Desenvolvimento Multiplataforma

Antigamente, os profissionais tinham a necessidade de desenvolver um aplicativo duas vezes, utilizando linguagens diferentes, a fim de assegurar a compatibilidade nos sistemas Android e iOS (SMART INNOVATION, 2021).

Madureira (2017), ressalta que os aplicativos nativos são desenvolvidos utilizando linguagens de programação exclusivas para um sistema operacional específico, o que resulta em incompatibilidade com outros sistemas. Nos dias de hoje, os sistemas operacionais dominantes no contexto de aplicativos móveis são o Android e o iOS. Para criar um aplicativo nativo, é necessário programá-lo na linguagem específica do sistema operacional, como Java e Kotlin para o Android (sendo o Kotlin a linguagem mais utilizada atualmente), e Objective-C ou Swift para o iOS. Essa abordagem garante um desempenho e integração otimizados com as funcionalidades e características de cada sistema, mas também implica em desenvolver versões separadas para cada plataforma.

Para Muller (2016), o desenvolvimento nativo é difícil e custoso. Outro ponto observado pelo autor, é que os usuários do iOS geralmente gastam mais dinheiro em compras dentro dos aplicativos e na loja da Apple em comparação com os usuários de dispositivos Android. Esse fator econômico muitas vezes incentivava desenvolvedores a priorizar o desenvolvimento para iOS e lançar aplicativos na plataforma da Apple primeiro. Como resultado, os usuários de dispositivos Android enfrentavam a necessidade de esperar vários dias ou até mesmo semanas para obter o mesmo *software* que já estava disponível na plataforma da Apple.

Com o desenvolvimento híbrido, por meio de uma ferramenta específica, um único modelo fica compatível em ambas as plataformas. Grosso modo, a redução com o custo pode chegar a 50% e por isso, *frameworks cross-platforms* possuem uma boa performance e segurança (SMART INNOVATION, 2021).

É importante entender o conceito de *framework* para prosseguir.

Segundo Sommerville (2018), um *framework* é uma estrutura genérica que fornece um conjunto de funcionalidades e abstrações comuns, projetadas para serem estendidas e especializadas em aplicações mais específicas. Os *frameworks* são projetados para economizar tempo e esforço, fornecendo suporte para características genéricas que são comuns a várias aplicações de um determinado tipo. Os

frameworks são uma ferramenta valiosa no desenvolvimento de *software*, pois permitem a reutilização de código, a padronização de práticas e a aceleração do processo de desenvolvimento, o que pode resultar em aplicativos mais robustos e eficientes.

Smart Innovation (2021), acrescenta que um *framework*, em desenvolvimento, pode ser visto como uma coleção de código comum que pode ser compartilhada entre vários projetos de *software*.

Dentre os *frameworks* disponíveis para desenvolvimento híbrido optou-se pelo Flutter para o desenvolvimento do Hornero.

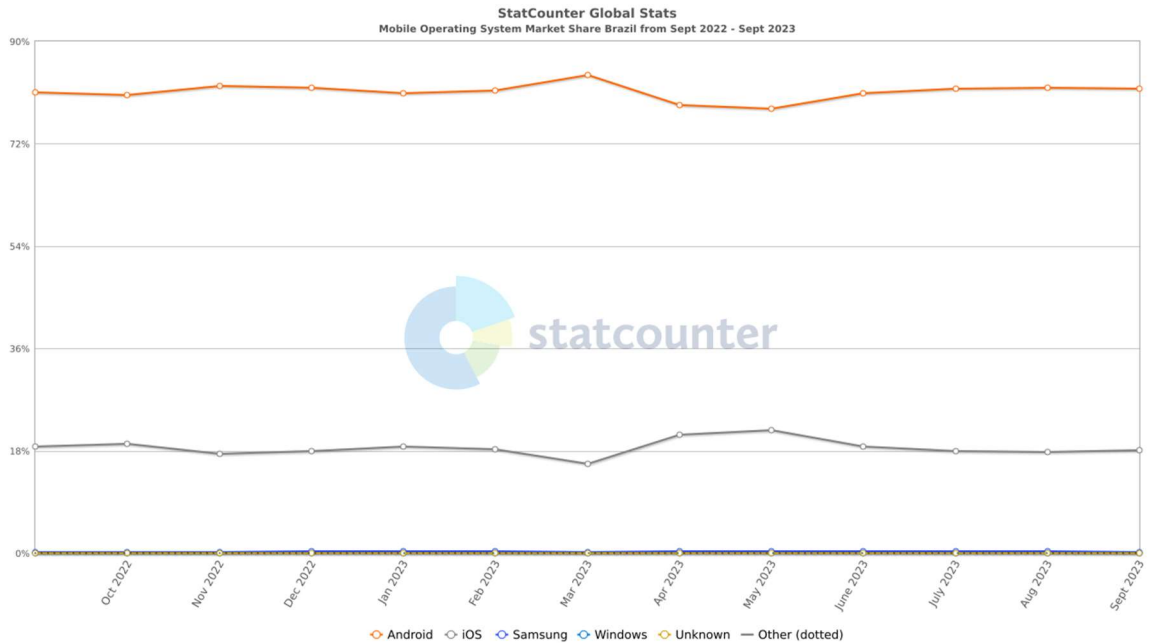
Pode-se afirmar que o Flutter é uma tecnologia recente e definitivamente considerado o preferido da comunidade. Além disso, é baseado em uma arquitetura de renderização 2D (processamento digital), utilizado nos jogos. Assim, o desempenho dos componentes e da animação apresenta um excelente resultado.

O *framework* Flutter, de fato, oferece uma abordagem diferente em relação aos *frameworks* tradicionais. Ele permite que os desenvolvedores escrevam o código em Dart e, em seguida, esse código é compilado para as linguagens nativas das plataformas-alvo, como Java e Kotlin para o Android e Objective-C e Swift para o iOS. Isso resulta em um desempenho que se assemelha a aplicativos nativos, proporcionando uma experiência de usuário fluída e de alta qualidade em múltiplas plataformas. Essa capacidade de compilação cruzada é uma das razões pelas quais o Flutter ganhou popularidade na comunidade de desenvolvimento de aplicativos (MADUREIRA, 2017).

2.3. Flutter

A partir da necessidade de desenvolver um aplicativo que atenda a dois tipos de usuários, que possuem em seus smartphones um dos sistemas operacionais mobile mais utilizados, com curva de aprendizado para o desenvolvimento, tecnologia empregada pelo *framework* e a linguagem de desenvolvimento, principalmente, tem-se o Flutter como boa opção, por atender aos requisitos mencionados acima.

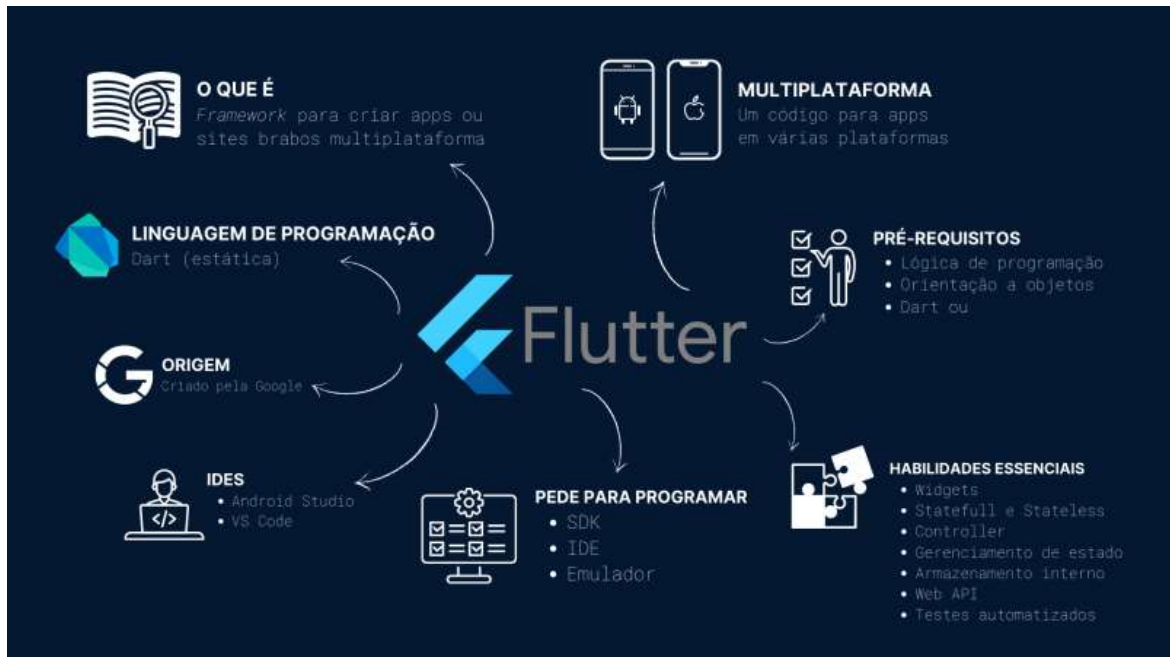
Gráfico 1 - Percentual de Participação dos Sistemas Operacionais para Dispositivos Móveis no Mercado Brasileiro



Segundo o gráfico 1, levantado pela StatCounter (2023), o Android e o iOS são os sistemas operacionais para smartphones mais utilizados no Brasil. Em setembro de 2023 esse percentual de utilização era de 81,66% e 18,06% respectivamente.

Flutter é um *framework* criado pela Google, seu uso já é bem disseminado no mercado e, permite também a criação de aplicações para *desktop* (Linux, Windows e macOS). Na prática, o Flutter é uma ferramenta que permite o desenvolvimento de aplicativos para as plataformas Android e iOS com um único código-base. Conforme verifica-se na figura 7, isso oferece uma série de benefícios, incluindo a versatilidade de atingir múltiplas plataformas, uma curva de aprendizado reduzida, e maior agilidade no processo de desenvolvimento, uma vez que as alterações no código podem ser refletidas em ambas as plataformas de forma consistente (ALBERTO, 2023).

Figura 1 - Mapa Mental Flutter



Fonte: ALBERTO (2023).

Para Marinho (2020), o Flutter é um *framework* que implementa o modelo de programação reativa inspirado no React. O autor complementa que tudo é organizado como Widget, independentemente de ser um botão, campo de texto ou espaçamento. Os widgets, no Flutter, são elementos de interface de usuário que são interpretados e renderizados diretamente como componentes nativos tanto no Android quanto no iOS. Isso significa que o código escrito em Flutter não se comunica com as plataformas nativas por meio de uma ponte, como em alguns outros *frameworks* multiplataforma, como o Ionic. Em vez disso, o Flutter gera código nativo de alto desempenho para cada plataforma alvo, resultando em aplicativos que se comportam e parecem nativos, sem a sobrecarga de uma camada de abstração. Isso é um dos pontos fortes do Flutter e uma razão pela qual ele é escolhido para desenvolver aplicativos multiplataforma com desempenho e aparência nativa.

2.4. Dart

Segundo Marinho (2020), Dart é uma linguagem otimizada para dispositivos clientes (ou *client-side*) multiplataforma, cujo objetivo é ser a mais produtiva e também

a mais flexível a fim de poder ser utilizada em diversos *frameworks* de desenvolvimento de aplicativos.

Tran (2023), aponta que Dart é utilizada por gigantes da tecnologia como o Google, Wix e eBay, é orientada a objetos baseada em classes que possui uma sintaxe no estilo de linguagem de programação “C”. Foi desenvolvida pelo Google em 2011 e posteriormente aprovado como um padrão pela ECMA. A linguagem Dart pode ser utilizada para o desenvolvimento de aplicativos web, aplicativos móveis e aplicativos de *desktop*. Além disso, ela pode ser usada para o desenvolvimento no lado do servidor com a ajuda do Node.js. Ela é uma linguagem rápida e escalável que suporta *ahead-of-time* (AOT) e é uma das linguagens mais fáceis de aprender. Com 0,95% de participação de mercado, o Dart é a 17ª linguagem de programação mais popular segundo aquele autor.

Conforme mencionado acima, Dart é AOT (*ahead-of-time*), ou seja, compila todo o código para linguagem de máquina antes da execução, trazendo estabilidade e previsibilidade ao comportamento do código. Este tipo de compilação oferece um tempo de inicialização mais ágil, especialmente quando a maior parte do código é executada durante o processo de inicialização do aplicativo, ou seja, ao abri-lo pela primeira vez. A linguagem Dart pode ser configurada para operar no modo JIT (*just-in-time*), interpretando o código em ciclos de desenvolvimento que exigem atualizações rápidas nos componentes da tela, sem a necessidade de recompilar todo o aplicativo. A linguagem simplifica a criação de animações e transições suaves que são executadas a uma taxa de 60 quadros por segundo. Além disso, a linguagem pode realizar alocação de objetos e coleta de lixo (*garbage collection*) sem bloqueios, o que alivia a pressão sobre a memória e o armazenamento de forma transparente e eficaz. Outro ponto de destaque é que o Flutter não necessita de uma linguagem de layout declarativo separado, como JSX, XML, HTML, ou muito menos de construtores de interface visual separados, porque o leiaute declarativo e programático da linguagem é fácil de compreender e visualizar sem precisar separar tudo em camadas (MARINHO, 2020).

2.5. Ambiente de Desenvolvimento

De acordo com Alberto (2023) para desenvolver em Flutter, é necessárias duas ferramentas: uma IDE (VSCode, VisualStudio, AndroidStudio, etc) ou um editor de

texto (bloco de notas, notepad++, Atom, etc) e além disso um dispositivo virtual ou físico que emula um celular funcionando em tempo real. Para Android é mais simples porque o emulador roda em várias máquinas, ao contrário, para o iOS precisa de um Mac com xCode. Aqui cabe uma observação a respeito do sistema onde vai ser desenvolvido o aplicativo: O conjunto *software* e hardware precisa ser robusto o suficiente para rodar o código e o emulador. Do contrário a experiência de desenvolver pode se tornar muito exaustiva e frustrante. Vencida essa parte, é necessário instalar, no computador, as seguintes ferramentas:

- 1) Flutter SDK: o arquivo que permite utilizar a ferramenta;
- 2) Android Studio e/ou xCode: contém as ferramentas de build para testar e subir a aplicação em dispositivos Android e iOS (além de servirem como IDE);
- 3) Em terceiro, a IDE que é o programa onde se escreve o código. Há diversas opções de IDE no mercado para escolher. No caso do aplicativo Hornero, optou-se por utilizar o Visual Studio Code.
- 4) Por fim, o emulador: necessário para rodar e testar o aplicativo. No caso será utilizado o Android Studio.

2.6. Banco de Dados

Não importa quem você seja ou o que faça, mais cedo ou mais tarde você precisará armazenar dados em seu aplicativo e recuperá-los posteriormente (CAP SISTEMA, 2022).

Conforme menção acima, não tem como pensar em um aplicativo nos moldes do que discorre esse trabalho, sem pensar em um banco de dados.

Os bancos de dados desempenham um papel crucial na gestão desse ativo valioso chamado "dados". Os dados, por si só, podem não conter muitas informações úteis, mas quando passam por processamento, formatação e correção de inconsistências, transformam-se em informações valiosas. O que geralmente é exibido nas interfaces dos sistemas são informações, que são, essencialmente, dados que foram devidamente tratados para se tornarem compreensíveis e úteis para os usuários das aplicações. Portanto, os bancos de dados desempenham um papel fundamental na coleta, armazenamento e recuperação desses dados, permitindo que sejam transformados em informações relevantes e de fácil compreensão, contribuindo

para o funcionamento eficiente de muitos aspectos de nossas vidas cotidianas (MARINHO, 2020).

Conforme relatado por Marinho (2020), quando se refere a dispositivos móveis, são inúmeras as opções do que se pode armazenar em bancos de dados locais, embora não tenham a robustez de memória interna necessária para armazenar um grande banco de dados. Dentro desse rol de opções pode-se apontar os dados de sessão, os quais garantem que o usuário não precise autenticar-se todas as vezes que acessa o aplicativo, dados de navegação, tokens de segurança, dados cadastrados pelo usuário e que se desejam manter localmente, caches de navegação no aplicativo com a ausência de conexão com a internet, igualmente como se tem nas redes sociais, etc .

Nesse caminho destaca-se o SQLite. Segundo o SQLite (2023), essa solução é uma biblioteca em linguagem C que fornece um banco de dados leve, baseado em disco. Ele é conhecido por ser autocontido, o que significa que requer poucas dependências externas ou configurações. O SQLite é projetado para ser incorporado em aplicativos, tornando-o uma escolha popular para aplicativos móveis e de *desktop*.

Como a essência da solução apresentada neste trabalho é a comunicação eletrônica entre pessoas, torna-se essencial que o banco de dados utilizado proporcione a disponibilização das informações para todos os usuários em tempo real. Por esse motivo, a opção escolhida para persistência dos dados foi utilizar os serviços do Firebase.

O Firebase é uma plataforma de *Backend-as-a-Service* (BaaS), o que significa que oferece uma infraestrutura de *back-end* pronta para desenvolvedores de aplicativos. Destaca-se que, com o Firebase, os desenvolvedores podem dedicar seu tempo à criação da aplicação em si, reduzindo a necessidade de lidar extensivamente com a configuração e manutenção de servidores (RIBEIRO, 2023).

Dentre os serviços abarcados nesta solução, destaca-se o Firebase Realtime Database, que é um banco de dados NoSQL hospedado na nuvem, por meio do qual, é possível armazenar e sincronizar dados entre os seus usuários em tempo real (GOOGLE, 2023).

Para Ribeiro (2023), as vantagens e desvantagens do Firebase são:

a) Facilidade de uso: Possui uma interface intuitiva e documentação abrangente, facilitando o processo de aprendizagem para os desenvolvedores.

b) Funcionalidades integradas: Os desenvolvedores podem escolher e utilizar os serviços que são mais relevantes para suas aplicações.

c) Tempo Real: Os serviços operam em tempo real, uma funcionalidade crucial para o desenvolvimento de *software* que requer atualizações instantâneas e sincronização em tempo real.

d) Autenticação e validação: Possui funcionalidades robustas de autenticação e validação de usuários, elementos essenciais para a segurança de aplicativos, garantindo a proteção adequada das informações e acesso autorizado.

Desvantagens do Firebase:

a) Limitações de personalização: Limitação de personalização em comparação com soluções mais flexíveis.

b) Custo escalonado: Embora ofereça um plano gratuito, os custos podem aumentar à medida que a aplicação escala ou faz uso intensivo de certos recursos.

2.7. Dispositivos Móveis

A proposta limita o desenvolvimento a um aplicativo para utilizar apenas em smartphones. Isso se dá em função da quantidade de aparelhos disponíveis e a praticidade e portabilidade de utilização do equipamento.

Essa grande quantidade de aparelhos se deve a uma combinação de funcionalidade versátil, conectividade, demanda por comunicação e dependência crescente da tecnologia móvel (FGV, 2023). Segundo pesquisa realizada anualmente pela FGV, o Brasil tem 464 milhões de dispositivos digitais (computador, notebook, tablet e smartphone) isso considerando o uso doméstico e corporativo. Ainda segundo a FGV (2023), estima-se um aumento de 10% nas vendas desses dispositivos esse ano.

Para efeito de conceito, o NIST (2023) define que dispositivo móvel pode ser entendido como um dispositivo computacional portátil cujo formato permite ser facilmente carregado por uma pessoa; pode operar sem uma conexão física para receber e transmitir informações; possui armazenamento de dados e contém uma fonte de energia. Pode-se citar como exemplo os smartphones, *tablets* e os *e-readers*.

A proposta deste desenvolvimento é ter um aplicativo que seja executado no maior número de smartphones possível e que tenham como sistema operacional tanto Android como iOS.

3. METODOLOGIA

Para a elaboração deste trabalho foi realizada uma pesquisa bibliográfica, de modo a revisar na literatura os textos publicados no setor da construção civil e de temas relacionados ao descarte de materiais de construção; bem como a execução de uma pesquisa-ação, para que, evidenciando uma dificuldade no setor estudado, se pudesse propor uma solução que corroborasse na tratativa.

4. MATERIAIS E MÉTODOS

Esse capítulo é destinado a apresentar conceitos relacionados às atividades relacionadas ao processo de desenvolvimento de *software*, a fim de proporcionar um melhor entendimento de como os artefatos resultantes do processo deste aplicativo foram obtidos. Neste capítulo, serão apresentados os principais artefatos resultantes desse processo.

4.1. Escopo do Projeto

Diante o que já foi apontado acima reforça-se que o objetivo desse sistema é facilitar a venda e doações de sobras de materiais da construção civil e considera-se como essencial algumas funcionalidades para que o sistema atenda ao proposto.

O aplicativo deverá conter uma área de login, cadastro de usuário, cadastro de anúncio de venda/doação, pagamento de anúncio e por fim um chat para negociação entre os interessados.

É de suma importância que os usuários sejam autenticados a fim de garantir a segurança de acesso ao sistema, e sua identificação diante de qualquer ação no sistema.

O cadastro de anúncio será o espaço utilizado para usuário inserir fotos, descrição do produto, local do anúncio a fim de facilitar que os interessados encontrem o produto por meio das informações disponíveis nos anúncios.

Uma das formas de manter o sistema, no sentido financeiro, é cobrar um valor por anúncio e para isso será necessário ter uma funcionalidade para que os usuários possam fazer pagamento utilizando os principais meios existentes no mercado.

A consulta do anúncio, por sua vez, fornecerá os filtros necessários para cada usuário encontrar o produto que deseja.

Por fim, o sistema deverá permitir que os usuários possam tirar as dúvidas sem necessitar migrarem para aplicativos de troca de mensagens e até mesmo finalizarem suas negociações.

4.2. Metodologia de Engenharia de Software

Acredita-se que a engenharia de *software* é de suma importância em qualquer processo de desenvolvimento de aplicações.

A aplicação dos conceitos dessa área do saber, no desenvolvimento de *software*, torna-se essencial pois ela se preocupa com os aspectos da produção de

software, desde sua concepção inicial até sua operação e manutenção (SOMMERVILLE, 2018).

Schaefer (2012), defende que sob as diretrizes da engenharia de *software*, o problema a ser tratado deve ser analisado e decomposto em partes menores, tratando-se de “dividir para conquistar” e que para cada parte resultante dessa divisão, deve-se elaborar uma solução.

Para Pressman (2011), independente da forma do *software* ou o seu campo de aplicação, todos devem passar pelos processos de engenharia. O autor justifica que os métodos da engenharia de *software* proporcionam obter as informações técnicas para desenvolver *softwares*.

Portanto, a fim de obter ganhos no desenvolvimento, torna-se necessário dividir todo o projeto em fases. Pressman (2011) afirma que uma metodologia de processo genérica estabelece cinco atividades: comunicação, planejamento, modelagem, construção e entrega. O autor menciona que pode ser utilizada em qualquer tipo de *software*, independentemente da complexidade e do tamanho.

Abaixo, há uma breve descrição das cinco atividades básicas propostas por Pressman (2011):

a) Comunicação: A primeira parte do trabalho é compreender os objetivos e levantar as necessidades das partes interessadas e somente depois disso inicia-se o trabalho técnico.

b) Planejamento: Essa fase pode ser explicada como uma analogia a ajuda de um mapa em uma jornada complexa. O final dessa fase seria um plano de projeto de *software* o qual, apresenta tarefas técnicas, os riscos prováveis, e os recursos que serão necessários, os produtos resultantes a ser produzidos e um cronograma de trabalho.

c) Modelagem: Aqui, desenvolve-se um “esboço”, de modo que se possa ter uma ideia do todo. Os modelos criados nessa fase ajudam a entender melhor as necessidades do *software* e o projeto.

d) Construção: Caracterizada pela produção do código e os testes que revelem erros na codificação.

e) Entrega: Como o próprio nome diz, o *software* é entregue ao cliente. Aqui recebe-se o *feedback* do cliente e avalia os resultados em relação a sua expectativa.

Interessante destacar a ressalva do autor, que menciona sobre os detalhes do processo de *software* serem bem diferentes em cada um dos casos, mas as atividades mencionadas acima permanecerão as mesmas.

Nessa linha, Gava (2009 apud Pfleeger e Atlee, 2006) expande o entendimento acima dizendo que os processos do desenvolvimento de *software* de maior interesse são:

4.2.1. Requisitos

Segundo Sommerville (2018), requisitos de um sistema representam as descrições dos serviços que o sistema deve prestar e as respectivas restrições para sua operação. Os requisitos refletem as necessidades dos clientes.

Para Pressman (2011), o objetivo do levantamento de requisitos é assimilar o que os vários interessados esperam do *software* que será desenvolvido. O modelo de requisitos deve alcançar três objetivos iniciais: (1) descrever o solicitado pelo cliente, (2) o estabelecimento de uma base para criar um projeto de *software* e (3) obter um conjunto de requisitos que possa ser validado assim que o *software* for desenvolvido.

Dentre os vários artefatos possíveis para essa fase, foram resultados, no levantamento de requisitos do aplicativo Hornero, as Histórias de Usuários, os Casos de Uso, o Diagrama de Classes e os Protótipos explicados e apresentados logo abaixo.

4.2.1.1. Histórias de Usuários

Segundo Rehkopf (2023), histórias de usuários são trechos que apresentam uma explicação informal e geral sobre uma função do *software* a qual é escrita a partir da perspectiva do usuário final. O principal objetivo é articular como um recurso de *software* pode gerar valor para o cliente.

Pode ser entendida, também, como uma metáfora, a qual pode ser explicada como “uma história que todos, desde clientes a gerentes, podem contar sobre como um recurso do sistema funciona” PRESSMAN (2011).

Soares (2021), aponta que para escrever uma história de usuário basta seguir o guia abaixo, o qual traz as informações:

- Como – autor da ação (persona);
- eu quero – trata-se da funcionalidade desejada;

- para – qual valor agregado à funcionalidade desejada.

Conforme apresentado na Figura 2, pode-se ter ideia de um exemplo aplicando o guia acima.

Figura 2 - Roteiro de Como Levantar História de Usuários



Fonte: SOARES (2021).

Para o aplicativo Hornero, pontua-se que as histórias de usuários são fictícias uma vez que o sistema não foi demandado por um cliente específico, mas o recurso ajuda no desenvolvimento do *software* servindo como um elo para as próximas fases. Na tabela 1 estão as histórias de usuários aplicadas ao desenvolvimento do aplicativo.

Tabela 1 - Levantamento das Principais Histórias de Usuários para o Aplicativo Hornero.

ID.	DESCRIÇÃO
01	Como anunciante gostaria de publicar meu produto para venda ou doação.
02	Como comprador gostaria de consultar anúncios cadastrados de acordo com a minha necessidade a fim de otimizar meu tempo de busca no sistema.
03	Como comprador gostaria de “favoritar” anúncios que oferecem produtos do meu interesse para facilitar uma compra em um momento mais oportuno.
04	Como comprador gostaria de ter um espaço para negociar os produtos disponíveis com o anunciante a fim de facilitar e agilizar a comunicação.
05	Como anunciante gostaria de corrigir erros em um anúncio que foi publicado para que não tenha prejuízo na venda com um anúncio incorreto.
06	Como anunciante gostaria de excluir um anúncio que foi publicado quando não for possível corrigir o erro para manter o meu espaço de venda sempre atualizado.

07	Como anunciante gostaria de concluir um anúncio cujo produto já foi vendido para manter os meus anúncios atualizados.
08	Como administrador gostaria de excluir anúncios que não atendem a proposta do aplicativo para que não se disperse o propósito da ideia principal do aplicativo.
09	Como administrador gostaria de receber do anunciante por cada anúncio publicado a fim de custear o sistema nas lojas oficiais de aplicativos, para todos os usuários, sem o uso de propagandas.

Fonte: Dos Autores

4.2.1.2. Casos de Uso

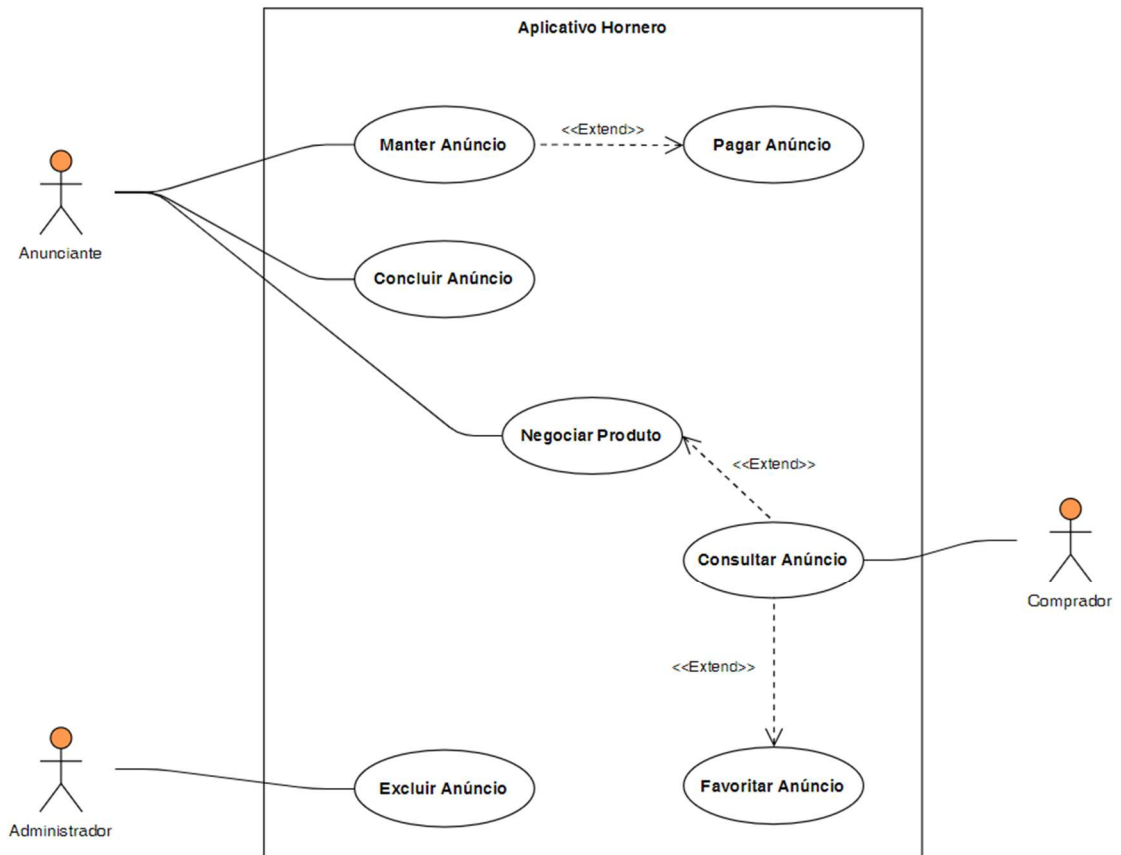
Os casos de uso do aplicativo Hornero, derivaram-se das histórias de usuários. Mas o que seria um caso de uso?

Para Pressman (2011) os casos de uso são definidos sob o ponto de vista de um ator. No caso, ator representa um papel que as pessoas (usuários) ou dispositivos desempenham quando estão interagindo com o *software*.

Segundo Booch et al. (2012), os casos de uso fornecem, aos desenvolvedores, uma maneira de compreensão em comum com os usuários finais do sistema e com os especialistas do domínio. Esses artefatos especificam o comportamento desejado; porém não determinam como esse comportamento será executado.

A figura 3 apresenta o diagrama de Casos de Uso do aplicativo Hornero, com base nas histórias de usuários.

Figura 3 - Diagrama de Casos de Uso do Aplicativo Hornero



Fonte: Dos Autores

4.2.1.3. Diagrama de Classes

Para Booch et al. (2012), um diagrama de classes mostra um conjunto de classes, interfaces e colaborações e seus relacionamentos. Os autores definem que os diagramas de classes são os artefatos que se encontram com maior frequência na modelagem de sistemas orientados a objetos.

De maneira geral, uma classe pode ser entendida como uma definição geral de um tipo de objeto de sistema. Os objetos representam alguma coisa no mundo real, por exemplo. Uma associação seria um vínculo entre as classes, que indica a existência de um relacionamento entre elas. Como consequência, cada classe pode precisar de algum conhecimento a respeito de sua classe associada. É normal que o primeiro estágio seja olhar para o mundo, identificar os objetos essenciais e representá-los como classes. SOMMERVILLE (2018).

Este diagrama destaca os dados essenciais para a criação do sistema de informação, em contraposição às suas funcionalidades. Dentro de todos os diagramas

da UML, o de Classes é aquele que apresenta o maior conjunto de símbolos para a sua representação gráfica (GÓES, 2014). O Digrama de Classes utilizado como base para o desenvolvimento do aplicativo Hornero encontra-se no apêndice A.

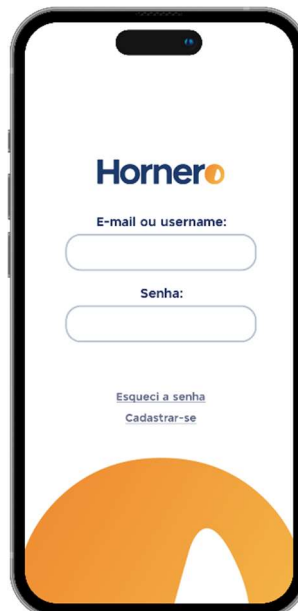
4.2.1.4. Protótipos

Pfleeger e Atlee (2010), defendem que o protótipo é um caminho para mostrar detalhes do sistema proposto e solicitar feedback dos usuários em potencial a fim de saber o que deveria ser melhorado, que funcionalidade estão faltando e se as existentes são úteis etc.

O protótipo seria uma versão inicial de um sistema. É útil para demonstrar conceitos, experimentar opções de projeto e descobrir mais sobre o problema e suas possíveis soluções (SOMMERVILLE, 2018).

Os protótipos do aplicativo Hornero foram feitos utilizando o Photoshop e estão apresentadas abaixo. No caso a ideia foi verificar leiaute das principais funcionalidades além de apresentar o padrão visual das telas.

Figura 4 - Protótipo da tela de login

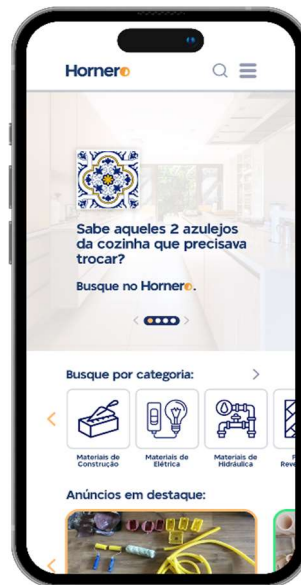


Fonte: Dos autores.

O protótipo acima demonstra a tela de login onde é exibido um campo para que o usuário digite o seu e-mail ou “*username*”; e um segundo campo para que insira a

sua senha (ambas informações já cadastradas previamente). Apresenta ainda o link “Esqueci minha senha” para que o usuário possa redefini-la e o *link* “Cadastre-se” para que novos usuários possam criar uma conta na aplicação.

Figura 5 - Protótipo da tela inicial



Fonte: Dos autores.

O protótipo acima demonstra a tela principal da aplicação que possui um menu superior principal contendo o logotipo, o ícone para pesquisa de anúncios e o ícone para navegar dentre as opções disponíveis; seguido de *banners* que destacam anúncios e a publicidade do próprio aplicativo; as categorias de anúncios disponíveis e; anúncios que estão em destaque na aplicação.

Figura 6 - Protótipo Visualização da tela de consulta de anúncios



Fonte: Dos Autores

O protótipo acima demonstra a tela de anúncios, que exibe todos os anúncios já cadastrados na aplicação, disponíveis para venda ou doação.

Figura 7 - Protótipo Visualização de um anúncio



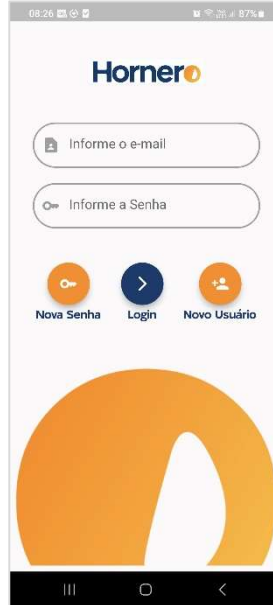
Fonte: Dos Autores

O protótipo acima demonstra o layout de um anúncio publicado dentro da aplicação, com todas as informações cadastradas pelo anunciante (usuário).

5. RESULTADOS E DISCUSSÕES

Os resultados apresentados nesta seção são as telas das principais funcionalidades do aplicativo desenvolvido e implementado em um aparelho móvel, bem como do sistema utilizado para a persistência dos dados.

Figura 8 - Tela de Login



Fonte: Dos Autores

A imagem acima demonstra a tela de login da aplicação, contendo o campo para inserção do e-mail do usuário e outro campo para inserção da senha de acesso, ambos de preenchimento obrigatórios. A tela apresenta o botão “Nova Senha” para que o usuário possa redefinir sua senha, caso necessário; o botão “Login” para confirmar os dados digitados e acessar a aplicação e; o botão “Novo Usuário” para que usuários que ainda não possuam uma conta cadastrada, possa efetuar seu cadastro.

Figura 9 - Tela de Cadastro de Usuário

Fonte: Dos Autores

A imagem acima demonstra a tela de cadastro de um novo usuário, contendo os campos “Nome”, “E-mail”, “CPF”, “Senha” e “Confirmação de Senha”. O preenchimento de todos os campos desta tela é obrigatório. A tela ainda possui o botão “Enviar Solicitação” para validar os dados digitados e criar uma nova conta para o respectivo usuário.

Figura 10 - Tela Principal

Fonte: Dos Autores

A imagem acima demonstra a tela principal da aplicação, contendo um menu superior principal contendo o logotipo, o ícone para pesquisa de anúncios e o ícone

para navegar dentre as opções disponíveis; seguido de *banners* que destacam anúncios e a publicidade do próprio aplicativo e; as categorias de anúncios disponíveis.

Figura 11 - Menu Principal



Fonte: Dos Autores

A imagem acima demonstra o menu principal da aplicação, que exibe os dados do usuário ativo e os botões “Home”, “Meus Anúncios”, “Minhas Vendas”, “Meus Pedidos”, “Favoritos”, “Minhas Negociações” e “Administrador” (quando o usuário ativo é administrador da aplicação), para que o usuário possa navegar na aplicação. Há botões cujas funções ainda não foram implementadas, nesta primeira versão da aplicação.

Figura 12 - Tela de Cadastro de Anúncio

09:03 84%

← Horner 🔍 ☰

Doação Venda

Valor do Produto 14.68

Marca Seleccionada: Duratex

Categoria Seleccionada: Revestim...

Descrição Produto: Rodapé MDF ...

Modelo do Produto: 15cm passa ...

Unidade de Medida: metros

Quantidade do Produto: 10

Confirmar

Fonte: Dos Autores

A imagem acima demonstra a tela de cadastro de novos anúncios dentro da aplicação. A primeira opção exibida é um componente que permite ao usuário informar se o anúncio que está sendo cadastrado é de um produto que será doado ou vendido (se a modalidade escolhida for venda, é solicitado que obrigatoriamente um valor seja cadastrado, caso contrário, o campo “Valor do Produto” é desativado). A tela ainda exibe os demais campos referentes às características do produto que está sendo anunciado.

Figura 13 - Tela de Consulta de Anúncio

Fonte: Dos Autores

A imagem acima demonstra a visualização de um anúncio publicado, onde é exibida todas as características do produto e a modalidade do anúncio (venda ou doação).

Figura 14 - Tela Resultado Pesquisa Anúncios

Fonte: Dos Autores

A imagem acima demonstra a visualização dos resultados obtidos após efetuar uma busca pelos produtos anunciados na aplicação atualmente.

Figura 15 - Tela Finalizar Uma Compra

Fonte: Dos Autores

A imagem acima demonstra a tela que é exibida quando o usuário deseja concluir uma compra dentro da aplicação, onde é solicitado que informe a quantidade que deseja comprar do respectivo produto e é exibido um botão para que confirme a transação.

Figura 16 - Lista de Compras Efetuadas

Fonte: Dos Autores

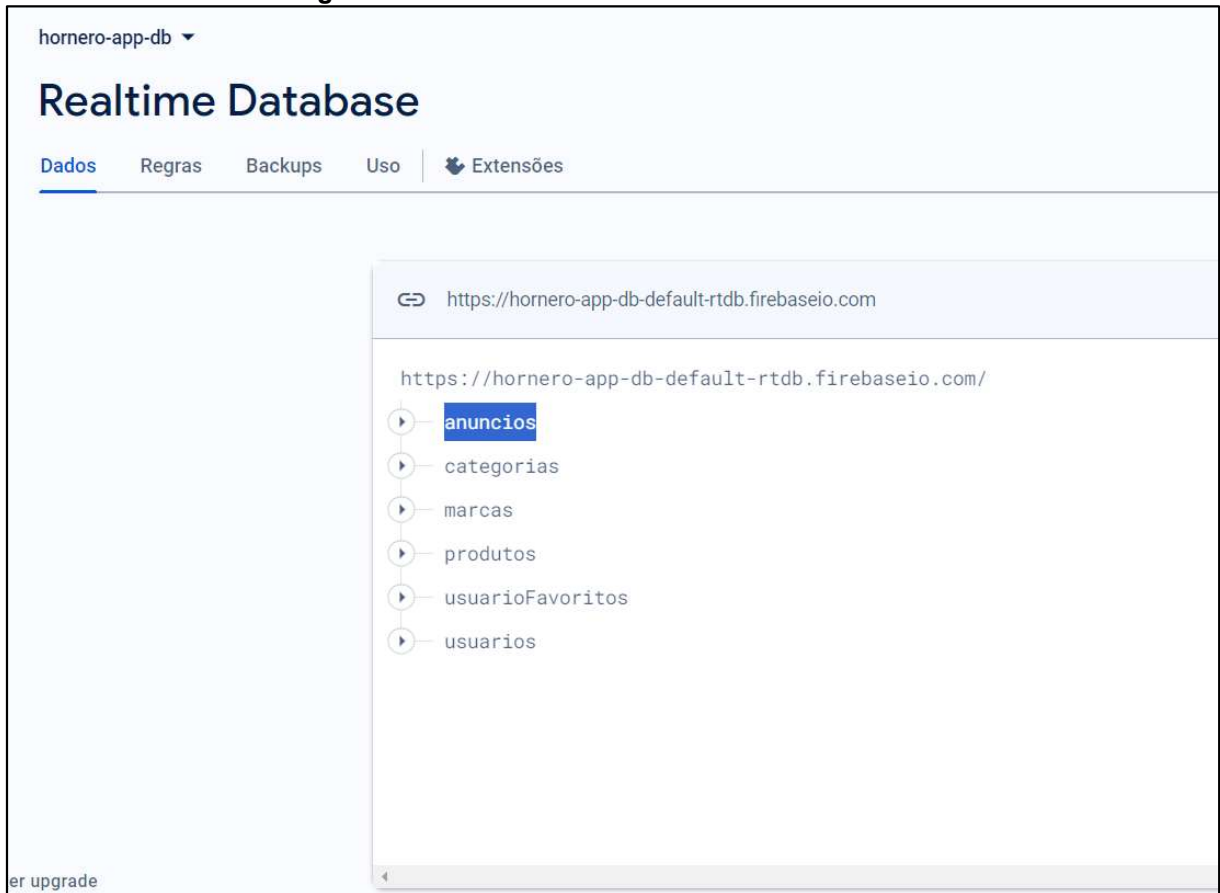
A imagem acima demonstra a lista de todos os produtos que um usuário já comprou utilizando a aplicação (histórico).

Figura 17 - Tela Troca de Mensagens (Vendedor x Comprador)



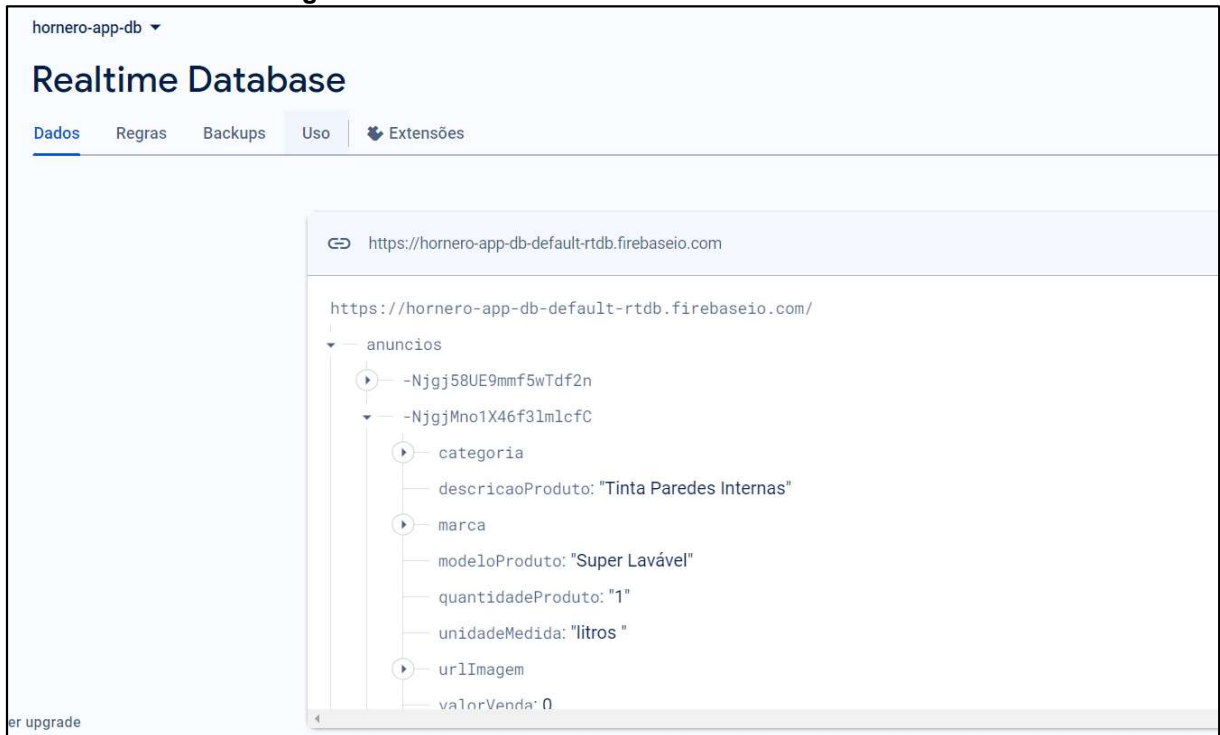
Fonte: Dos Autores

A imagem acima demonstra a forma de comunicação entre anunciantes e compradores dentro da aplicação. O formato utilizado é de um chat onde as respectivas partes são inteiramente responsáveis pelas informações fornecidas.

Figura 18 - Visão dos Dados Persistidos no Firebase

Fonte: Dos Autores

A imagem acima demonstra como os dados estão persistidos no Firebase. Diferente de banco de dados relacional no qual os dados são organizados em tabelas, neste tipo de banco de dados os dados são armazenados em documentos e coleções.

Figura 19 - Visão De Um Anúncio Gravado no Firebase

Fonte: Dos Autores

A imagem acima demonstra a estrutura de armazenamento do Firebase. Na estrutura acima, “anuncios” representa uma coleção e cada “filho” de “anuncio” é um documento.

6. CONSIDERAÇÕES FINAIS

Este trabalho teve como propósito principal a concepção, desenvolvimento e execução de um aplicativo móvel que atendesse às demandas dos usuários interessados em transações envolvendo resíduos de materiais de construção provenientes de obras e reformas. O aplicativo foi projetado para operar de maneira eficiente nos sistemas operacionais móveis mais amplamente utilizados, iOS e Android, com o intuito de proporcionar uma plataforma acessível e inclusiva.

Ao enfrentarmos o desafio de desenvolver o aplicativo, partimos de um conhecimento inicial quase nulo em relação à tecnologia específica exigida para sua implementação. No entanto, superamos esse obstáculo, possibilitando o desenvolvimento e a implementação dentro de um prazo relativamente curto. Um destaque relevante é que conseguimos alcançar esse marco sem custos significativos, demonstrando a viabilidade e eficácia do padrão adotado no processo.

Cabe observar que a impossibilidade de testar o aplicativo em dispositivos iOS é resultante das restrições impostas pela Apple à simulação em hardware de outras marcas. A limitação de testes ao sistema Android, devido a investimentos em hardware específico, representa um aspecto a ser considerado para futuras melhorias e expansões.

Apesar de não termos conduzidos testes "beta" com usuários reais, percebemos, por meio de avaliações internas, que o aplicativo exibe um notável potencial de utilização. Sua proposta única e facilidade de interação sugerem uma aceitação favorável por parte dos usuários, evidenciando sua relevância no contexto do gerenciamento sustentável de resíduos de construção.

Como perspectiva para futuros trabalhos, sugerimos a elaboração de cenários de teste mais abrangentes, envolvendo usuários em situações do cotidiano, a fim de aprimorar a usabilidade e identificar possíveis melhorias. Além disso, a implementação de opções para transações financeiras dentro da aplicação surge como uma evolução natural, embora reconheçamos a necessidade de considerar cuidadosamente as implicações e desafios associados à integração de APIs de terceiros para esse propósito. Este aspecto, embora não tenha sido explorado neste momento, permanece como uma oportunidade para futuras investigações visando enriquecer ainda mais a funcionalidade do aplicativo.

REFERÊNCIAS

ALBERTO, Matheus. **Flutter: o que é e tudo sobre o framework**. [S.l.], 2023. Disponível em: <<https://www.alura.com.br/artigos/flutter>>. Acesso em 14/10/2023.

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. **UML guia do usuário**. Rio de Janeiro: Elsevier, 2012.

CAP SISTEMA. **Escolhendo o banco de dados certo para o seu aplicativo Flutter. [S.l.]**, 2022. Disponível em: <<https://capsistema.com.br/index.php/2022/09/09/escolhendo-o-banco-de-dados-certo-para-o-seu-aplicativo-flutter/>>. Acesso em 14/10/2023.

CONAMA. Ministério do Meio Ambiente. Conselho Nacional do Meio Ambiente. **Resolução nº 307, de 05 de julho de 2002. Estabelece diretrizes, critérios e procedimentos para a gestão dos resíduos da construção civil**. Diário Oficial da República Federativa do Brasil, Brasília, DF, nº 136, 17 de julho de 2002, Seção1, p. 95-96.

FGV. **Uso de TI no Brasil: País tem mais de dois dispositivos digitais por habitante, revela pesquisa**. Brasil, 2023. Disponível em: <<https://portal.fgv.br/noticias/uso-ti-brasil-pais-tem-mais-dois-dispositivos-digitais-habitante-revela-pesquisa>>. Acesso em 08/10/2023.

GAVA, Vagner Luiz. **Processo para especificação de requisitos de software com foco na aplicação em trabalho cooperativo**. 2009. 288f. Tese (Doutorado) - Escola Politécnica da Universidade de São Paulo.

GÓES, Wilson Moraes. **Aprenda UML por meio de estudos de caso**. 1 ed. São Paulo, 2014.

GOOGLE. **Firestore Realtime Database**. Armazene e sincronize dados em tempo real. [S.l.], 2023. Disponível em: <<https://firebase.google.com/products/realtime-database?hl=pt-br>>. Acesso em 12/11/2023.

GOMES, Carla Pinheiro; LEITE, Guilherme Urquiza; SENA, Rafael Wandson Rocha; ANDRADE, Elysson Marcks Gonçalves de. **Impacto Ambiental e Gerenciamento de Resíduos Sólidos Advindos da Construção Civil no Brasil: Uma Revisão de Literatura**. Id on Line Ver. Mult. Psci., Maio/2021, vol. 15, n.55, p. 729-742, ISSN: 1981-1179.

IPEA – Instituto de Pesquisa Econômica Aplicada. **Diagnóstico dos Resíduos Sólidos da Construção Civil**. Brasília, 2012. Disponível em <https://portalantigo.ipea.gov.br/agencia/images/stories/PDFs/relatoriopesquisa/120911_relatorio_construcao_civil.pdf>. Acesso em 15 de Agosto de 2023.

LOPES, Thayana Azevedo. **Tecnossolos elaborados com resíduos da construção civil: potencial para a recuperação de áreas degradadas**. Dissertação (Mestrado) – USP/Escola Superior de Agricultura “Luiz de Queiroz”. Piracicaba, 2020.

MARINHO, Leonardo H. **Flutter Framework: desenvolva aplicações móveis do Dart Side!** Casa do Código: [S.l.], 2020.

MADUREIRA, Daniel. **Aplicativo nativo, web App ou aplicativo híbrido?** [S.l.], [2017?]. Disponível em: <<https://usemobile.com.br/aplicativo-nativo-web-hibrido/>>. Acesso em 10/10/2023.

MULLER, Léo. **Plataforma da Intel permite desenvolver para Android e iOS ao mesmo tempo**. [S.l.], 2016. Disponível em: <<https://www.tecmundo.com.br/programacao/107831-plataforma-intel-permite-desenvolver-android-ios-mesmo-tempo.htm>>. Acesso em 10/10/2023.

NIST. **Glossary**. Maryland, Estados Unidos, [2023?]. Disponível em: <https://csrc.nist.gov/glossary/term/mobile_device>. Acesso em 08/10/2023.

PFLEEGER, Shari Lawrence e ATLEE, Joanne M.. **Software engineering: theory and practice**. 4th ed. [USA?]: Pearson, 2010.

PRESSMAN, Roger S. **Engenharia de software[recurso eletrônico]: uma abordagem profissional**. 7 ed. Porto Alegre: AMGH, 2011.

REHKOPF, Max. **Histórias de usuários com exemplos e um template: Histórias de usuários, ou user stories, são tarefas de desenvolvimento expressas, em geral, como "persona + necessidade + propósito"**. S.l, [2023?]. Disponível em: <<https://www.atlassian.com/br/agile/project-management/user-stories>>. Acesso em 08/10/2023.

RIBEIRO, Andre Louis Souza. **O que é Firebase? Para que serve, principais características e um Guia dessa ferramenta Google**. [S.l.], 2023. Disponível em: <<https://www.alura.com.br/artigos/firebase>>. Acesso em 12/11/2023.

RIBEIRO, Angélica Kelly dos S. Pimentel; MARQUES, Sheyla Karolina Justino, RIBEIRO, Igor Bruno Gomes; MAIA, Stoecio Malta Ferreira. **Quantificação e Classificação dos Resíduos Procedentes da Construção Civil no Município de Maceió-AL**. R. Gest. Sust. Ambien., Florianópolis, Maio/2021, v. 10, n.1, p.363-384.

SCHAEFER, Vagner Savegnago. **Desenvolvimento de um sistema de mapeamento de estradas utilizando tecnologia móvel**. 2012. 63f. Dissertação (Mestrado do Programa de Pós-Graduação em Geomática) - Universidade Federal de Santa Maria.

SILVA, Mariana Valério da. **Diagnóstico e propostas de melhoria para o gerenciamento de resíduos sólidos em estabelecimento comercial do ramo da construção civil.** São Carlos, 2020.

SMART INNOVATION. **Aplicativo híbrido: O que é e por que você deveria conhecer.** Indaiatuba, [2021?]. Disponível em: < <https://www.smartinnovation.com.br/blog/aplicativo-hibrido-o-que-e-e-por-que-voce-deveria-conhecer/>>. Acesso em 12/10/2023.

SOARES, Felipe. **Histórias de usuário: Desmistificando a escrita de User Story.** Belo Horizonte, 2021. Disponível em: < <https://www.dtidigital.com.br/blog/historias-de-usuario>>. Acesso em 08/10/2023.

SOMMERVILLE, Ian. **Engenharia de software.** 10 ed. São Paulo: Pearson Education do Brasil, 2018.

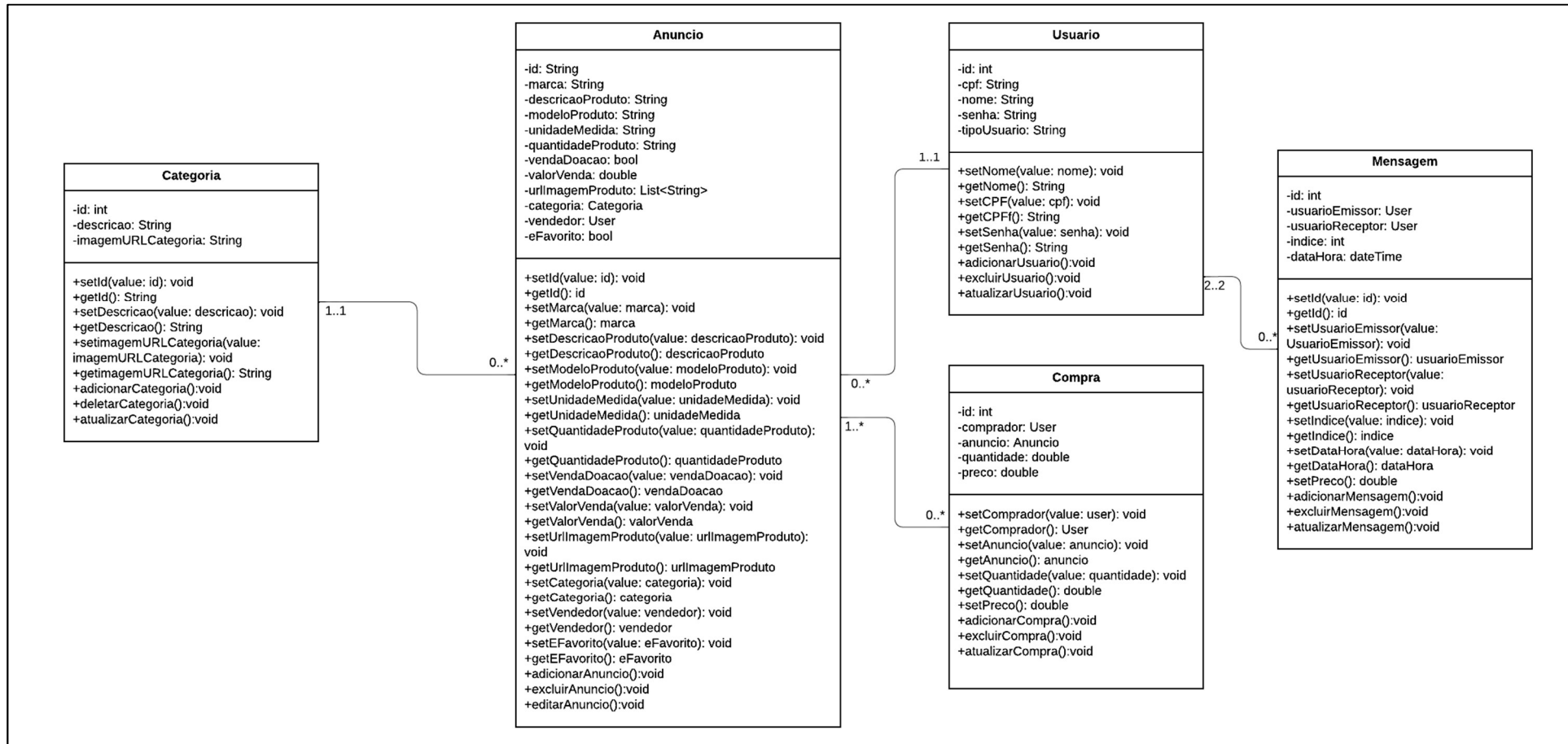
SQLite. **What Is SQLite?** [S.I.], 2023. Disponível em: <<https://www.sqlite.org/index.html>>. Acesso em 12/11/2023.

STATCOUNTER. **Operating System Market Share Brazil.** [S.I.], 2023. Disponível em: <<https://gs.statcounter.com/os-market-share/all/brazil>>. Acesso em 12/10/2023.

TECHTUDO. **Oito melhores sites para vender e comprar produtos usados online.** São Paulo, 2022. Disponível em <https://www.techtudo.com.br/listas/2022/10/oito-melhores-sites-para-vender-e-comprar-produtos-usados-online.ghtml>>. Acessado em 08/10/2023.

TRAN, Trung. **Top 20 Most Popular Programming Languages in 2023 & Beyond.** Vietnam, 2023. Disponível em: < <https://www.orientsoftware.com/blog/most-popular-programming-languages/>>. Acesso em 14/10/2023.

APÊNDICE A – DIAGRAMA DE CLASSES HORNERO



Fonte: Dos autores.