

INSTITUTO FEDERAL DE SÃO PAULO
CAMPUS DE SÃO CARLOS
PÓS-GRADUAÇÃO *LATO SENSU* EM DESENVOLVIMENTO DE
SISTEMAS PARA DISPOSITIVOS MÓVEIS

LUCAS DE OLIVEIRA MACÊDO

**SHESAFE – APLICATIVO DE PEDIDO DE AJUDA PARA
MULHERES EM SITUAÇÃO DE RISCO**

SÃO CARLOS

2025

LUCAS DE OLIVEIRA MACÊDO

SHESAFE – APLICATIVO DE PEDIDO DE AJUDA PARA
MULHERES EM SITUAÇÃO DE RISCO

Trabalho de conclusão apresentado ao Instituto Federal de São Paulo para obtenção do título de Especialista em Desenvolvimento para Dispositivos Móveis.

Orientador: Silvana Maria Affonso de Lara

SÃO CARLOS

2025

Resumo

A violência contra a mulher constitui um problema de saúde pública global de proporções epidêmicas, afetando aproximadamente 770 milhões de mulheres anualmente em relacionamentos com parceiros ou ex-parceiros. A pandemia da COVID-19 agravou significativamente este cenário, uma vez que o isolamento social, principal medida de contenção adotada, criou condições propícias para o fortalecimento dos elementos de violência doméstica, mantendo as vítimas por períodos prolongados sob controle direto de seus agressores. Consequentemente, observou-se um aumento expressivo dos casos de violência contra a mulher durante o período pandêmico, intensificando uma tendência de crescimento já identificada anteriormente.

Palavras-chaves: Violência contra a mulher. Aplicação mobile. Geolocalização. Pedido de socorro. Proteção da mulher. Violência doméstica.

Abstract

Violence against women is a global public health problem of epidemic proportions, affecting approximately 770 million women annually in relationships with current or former partners. The COVID-19 pandemic has significantly worsened this situation, as social isolation, the primary containment measure imposed, created conditions conducive to the intensification of domestic violence, keeping victims under the direct control of their aggressors for prolonged periods. Consequently, there was a significant increase in cases of violence against women during the pandemic, intensifying a previously observed upward trend.

Key-words: Violence against women. Mobile application. Geolocation. Call for help. Women's protection. Domestic violence.

Lista de ilustrações

Figura 1 – Mapa de empatia inicial	9
--	---

Lista de tabelas

Tabela 1 – Cronograma Normal	21
Tabela 2 – Cronograma (IBGE)	21

Sumário

1	Introdução	8
1.1	Escopo do Projeto	8
1.2	Público-alvo	8
1.3	Mapa de empatia inicial	8
1.4	Ambiente de uso	8
1.5	Restrições de Uso/Circunstâncias	9
1.6	Diferencial	9
1.7	Funcionalidades mínimas para garantir a relevância do aplicativo	10
2	Referencial Teórico	11
2.1	Sistema operacional Android	11
2.1.1	Posicionamento no Mercado Global	11
2.1.2	Características de Licenciamento e Código Aberto	12
2.2	Android Studio	12
2.2.1	Caracterização e Fundamentação Técnica	12
2.2.2	Funcionalidades e Recursos Especializados	12
2.2.3	Recursos de Desenvolvimento Ágil e Qualidade de Código	13
2.3	Firebase	14
2.3.1	Caracterização e Contextualização Histórica	14
2.3.2	Interface de Gerenciamento e Implementação	14
2.3.3	Modelo de Comercialização e Estrutura Tarifária	15
2.3.4	Firebase Authentication	15
2.3.4.1	Caracterização do Serviço e Arquitetura Funcional	15
2.3.4.2	Modalidades de Autenticação e Provedores de Identidade	15
2.3.4.3	Integração Sistêmica e Conformidade com Padrões da Indústria	16
2.3.5	Firebase Firestore	16
2.3.5.1	Caracterização e Arquitetura do Sistema	16
2.3.5.2	Funcionalidades de Sincronização e Tempo Real	17
2.3.5.3	Escalabilidade e Modelo de Precificação	17
2.4	Compose Multiplatform	18
2.4.1	Fundamentação Conceitual e Arquitetural	18
2.4.2	Paradigma Declarativo e Gerenciamento de Estado	18
2.4.3	Interoperabilidade e Integração com Ecossistemas Nativos	18
2.4.4	Performance e Otimizações de Renderização	19
3	Trabalhos Relacionados	20
4	Metodologia e Cronograma	21

5 Resultados Esperados	22
Referências	23

1 Introdução

O isolamento social durante o período de pandemia acabou se tornando um fator primordial no aumento dos conflitos familiares, isso obrigou as mulheres, que convivem com seus agressores, a permanecerem por um período mais longo junto a eles. Embora haja visto um aumento no número de casos de violência contra a mulher, constata-se, segundo dados da Cartilha de Violência contra a mulher de maio de 2020, que o número de denúncias diminuiu. Tal fato se dá pela proximidade da vítima ao agressor, pelo sequestro de subjetividade, pela precariedade do serviço público e pelo medo de sair do isolamento, dentre outros fatores.

1.1 Escopo do Projeto

O escopo do projeto tem como base oferecer uma aplicação simples, intuitiva e de fácil acesso ao nosso público-alvo para que estes sejam capazes de enviar um pedido de ajuda, para pessoas de sua confiança, em momentos que estiverem passando por uma situação de risco, agressão ou até mesmo violência. Estes pedidos de ajuda informarão a essas pessoas de confiança, na aplicação denominadas de “contatos seguros”, a localização atual do usuário juntamente com uma mensagem predefinida por ele.

1.2 Público-alvo

SheSafe tem como público-alvo pessoas do sexo feminino que desejam ter uma forma de comunicar a pessoas de sua confiança algum potencial risco; seja ele violência, agressão, insegurança na rua, perseguição por algum indivíduo ou outro risco qualquer, que estiver passando em tempo real

1.3 Mapa de empatia inicial

1.4 Ambiente de uso

O SheSafe poderá ser utilizado em qualquer ambiente desde que se tenha um plano de internet ativo para que os pedidos de ajuda possam ser enviados corretamente para a lista de contatos seguros. Uma outra restrição quanto ao uso é que os usuários devem possuir um smartphone com sistema operacional Android ou iOS para que possam instalar o aplicativo.



Figura 1 – Mapa de empatia inicial sobre sentimento das mulheres sobre violência

Fonte: Próprio Autor

1.5 Restrições de Uso/Circunstâncias

Não há restrições quanto ao uso do SheSafe. Caso o usuário sinta-se em situação de risco ele poderá acionar o aplicativo e enviar seu pedido de ajuda, desde que já tenha cadastrado sua lista de contatos seguros.

1.6 Diferencial

Dado que muito dos casos de agressão acontecem com mulheres em situação de risco e muitas das vezes elas são privadas de acesso a formas de comunicação mais tecnológicas, o intuito é ter uma aplicação onde as mulheres possam reportar crimes de forma simples, prática e disponível para todos os públicos. O principal diferencial está em não restringir o acesso a aplicação para nosso público-alvo e ter a opção de envio da localização no momento do acontecimento.

1.7 Funcionalidades mínimas para garantir a relevância do aplicativo

As funcionalidades mínimas elencadas para que o projeto tenha relevância são:

- a) Login: Para criação de usuário
- b) Pedido de socorro: Botão para acionar pedido de socorro
- c) Lista Segura: Gerenciamento de contatos seguros
- d) Perfil: Página para alteração da mensagem de socorro

2 Referencial Teórico

2.1 Sistema operacional Android

O Android constitui-se como um sistema operacional fundamentado no núcleo Linux, atualmente sob desenvolvimento e manutenção da empresa Google Inc. ([Google Inc., 2023](#)). Este sistema operacional caracteriza-se por uma arquitetura de interface baseada em manipulação direta, sendo especificamente projetado para atender às demandas de dispositivos móveis equipados com tecnologia de tela sensível ao toque, incluindo smartphones e tablets ([BURNETTE, 2021](#)).

A versatilidade da plataforma Android estende-se para além dos dispositivos móveis convencionais, abrangendo uma ampla gama de equipamentos tecnológicos. Neste contexto, destacam-se as adaptações específicas como o Android TV para televisores inteligentes, o Android Auto para sistemas automotivos e o Android Wear para dispositivos vestíveis, particularmente relógios inteligentes ([ABLESON; SEN; KING, 2022](#)). O sistema utiliza recursos de interface intuitiva, empregando a tecnologia de tela sensível ao toque para permitir que os usuários manipulem objetos virtuais através de gestos diretos, complementada por um teclado virtual integrado.

Embora o Android tenha sido concebido primordialmente para dispositivos com interface touchscreen, sua aplicabilidade transcende esta categoria, sendo implementado em consoles de videogames, câmeras digitais, computadores pessoais e diversos outros equipamentos eletrônicos ([MURPHY, 2023](#)). Esta flexibilidade arquitetural demonstra a robustez e adaptabilidade da plataforma para diferentes contextos de uso.

2.1.1 Posicionamento no Mercado Global

Dados estatísticos revelam a significativa penetração do sistema Android no mercado mundial de sistemas operacionais. Segundo levantamento conduzido pela StatCounter Global Stats ([StatCounter Global Stats, 2017](#)), empresa especializada em pesquisas e análises estatísticas de mercado tecnológico, em março de 2017, os usuários do Android representavam 37,93% da atividade global em redes, superando ligeiramente o sistema Windows, que registrou 37,91% no mesmo período. Estes números consolidaram o Android como o sistema operacional mais utilizado mundialmente, marcando um ponto de inflexão significativo no panorama tecnológico global.

No âmbito do desenvolvimento de aplicações móveis, pesquisa conduzida entre programadores durante o período de abril a maio de 2013 evidenciou que 71% dos desenvolvedores direcionavam seus esforços para a plataforma Android ([Developer Economics,](#)

2013). Esta preferência dos desenvolvedores reflete tanto a popularidade do sistema quanto as oportunidades de mercado que a plataforma oferece.

2.1.2 Características de Licenciamento e Código Aberto

Um aspecto fundamental que distingue o Android de outros sistemas operacionais móveis refere-se ao seu modelo de licenciamento. O sistema operacional distribuído pela Google opera sob licença de código aberto (open source), fundamentada na Apache License 2.0 ([Open Source Initiative, 2023](#)), garantindo aos desenvolvedores e programadores os direitos fundamentais de estudar, modificar e distribuir o software de forma gratuita.

Esta característica de código aberto proporciona liberdade significativa para qualquer indivíduo ou organização utilizar o sistema para qualquer finalidade, sem restrições de uso comercial ou não comercial. No entanto, é importante destacar que, embora o núcleo do sistema Android seja distribuído como software livre, a maioria dos dispositivos comerciais são lançados com uma combinação híbrida de software livre e software proprietário, incluindo aplicativos e serviços específicos do fabricante ou do Google ([LEE, 2022](#)).

2.2 Android Studio

2.2.1 Caracterização e Fundamentação Técnica

O Android Studio constitui-se como o ambiente de desenvolvimento integrado (IDE - Integrated Development Environment) oficial designado pela Google para o desenvolvimento de aplicações móveis destinadas à plataforma Android ([Google Developers, 2023](#)). Esta ferramenta de desenvolvimento fundamenta-se na arquitetura do IntelliJ IDEA, uma das principais plataformas de desenvolvimento Java desenvolvida pela JetBrains, incorporando suas funcionalidades centrais de edição de código e ferramentas avançadas de desenvolvimento ([JetBrains, 2023b](#)).

A escolha do IntelliJ IDEA como base arquitetural para o Android Studio justifica-se pela robustez e maturidade desta plataforma no ecossistema de desenvolvimento Java, linguagem predominante no desenvolvimento Android tradicional. Esta decisão estratégica permitiu à Google herdar um conjunto consolidado de funcionalidades de desenvolvimento, enquanto adiciona camadas especializadas voltadas especificamente para as particularidades do desenvolvimento móvel Android ([MEIER, 2023](#)).

2.2.2 Funcionalidades e Recursos Especializados

O Android Studio incorpora um conjunto abrangente de funcionalidades específicas para otimizar o processo de desenvolvimento de aplicações Android, transcendendo as

capacidades básicas de edição de código. Entre os recursos mais significativos para a produtividade do desenvolvedor, destacam-se elementos fundamentais que caracterizam a modernidade e eficiência da plataforma.

O sistema de compilação integrado baseia-se na ferramenta Gradle, um sistema de automação de build de código aberto que oferece flexibilidade superior aos sistemas tradicionais baseados em XML, como o Apache Maven ([Gradle Inc., 2023](#)). Esta integração proporciona gerenciamento automatizado de dependências, compilação incremental e configuração modular de projetos, elementos essenciais para o desenvolvimento escalável de aplicações complexas.

A plataforma incorpora um emulador Android de alta performance, desenvolvido com tecnologias de virtualização otimizadas que permitem a simulação eficiente de diversos dispositivos Android sem a necessidade de hardware físico ([Android Developers, 2023a](#)). Este recurso é fundamental para testes de compatibilidade e validação de funcionalidades em diferentes configurações de hardware e versões do sistema operacional.

O ambiente oferece suporte unificado para desenvolvimento direcionado a múltiplas categorias de dispositivos Android, incluindo smartphones, tablets, dispositivos vestíveis (wearables), sistemas automotivos e televisores inteligentes. Esta abordagem unificada simplifica significativamente o processo de desenvolvimento para ecossistemas heterogêneos de dispositivos ([ABLESON; KUMAR, 2023](#)).

2.2.3 Recursos de Desenvolvimento Ágil e Qualidade de Código

O Android Studio implementa funcionalidades avançadas voltadas para metodologias de desenvolvimento ágil e manutenção da qualidade de código. O recurso Instant Run representa uma inovação significativa no ciclo de desenvolvimento, permitindo a aplicação de alterações de código diretamente em aplicações em execução, eliminando a necessidade de recompilação completa do arquivo APK (Android Package) ([Android Development Team, 2022](#)). Esta funcionalidade reduz substancialmente o tempo de iteração durante o processo de desenvolvimento e depuração.

A integração nativa com sistemas de controle de versão, particularmente o GitHub, facilita a colaboração em equipes de desenvolvimento e o gerenciamento de código fonte ([GitHub Inc., 2023](#)). Adicionalmente, a plataforma disponibiliza modelos de código pré-configurados (code templates) que aceleram o desenvolvimento de funcionalidades comuns e promovem a padronização de práticas de codificação.

O ambiente incorpora um conjunto robusto de ferramentas e frameworks para implementação de testes automatizados, abrangendo testes unitários, testes de integração e testes de interface do usuário ([JUnit Team, 2023](#)). Estas funcionalidades são complementadas por ferramentas especializadas de análise estática de código, capazes de identificar

potenciais problemas relacionados a performance, usabilidade, compatibilidade entre versões do Android e outras questões críticas para a qualidade da aplicação final ([Android Static Analysis Team, 2023](#)).

2.3 Firebase

2.3.1 Caracterização e Contextualização Histórica

O Firebase constitui-se como uma plataforma abrangente de desenvolvimento para aplicações móveis e web, tendo sido incorporada ao portfólio de soluções da Google Inc. em outubro de 2014 através de processo de aquisição estratégica ([Google Inc., 2014](#)). Originalmente fundada em 2011 por James Tamplin e Andrew Lee, a plataforma Firebase foi concebida com o propósito de fornecer uma infraestrutura de backend-as-a-service (BaaS) completa e de alta usabilidade, direcionada especificamente para simplificar o processo de desenvolvimento e implementação de aplicações digitais ([TAMPLIN; LEE, 2021](#)).

A filosofia de desenvolvimento da plataforma fundamenta-se na disponibilização de um ecossistema integrado de serviços diversos, que abrangem desde funcionalidades básicas de armazenamento de dados até recursos avançados de análise comportamental e engajamento de usuários ([Firebase Team, 2023b](#)). Esta abordagem holística visa reduzir a complexidade técnica tradicionalmente associada ao desenvolvimento de backends customizados, permitindo que desenvolvedores concentrem seus esforços na criação de interfaces e experiências de usuário diferenciadas.

2.3.2 Interface de Gerenciamento e Implementação

A operacionalização da plataforma Firebase é mediada por um console web especializado, denominado Firebase Console, que representa a interface principal de gerenciamento e configuração de projetos ([Google Firebase, 2023b](#)). Este ambiente de desenvolvimento integrado foi projetado seguindo princípios de experiência do usuário (UX) orientados à simplicidade e intuitividade, proporcionando aos desenvolvedores um fluxo de trabalho otimizado para implementação de recursos.

O processo de utilização inicia-se com a criação de um projeto no console, seguido pela seleção e configuração dos serviços desejados a partir de um catálogo extenso de funcionalidades disponíveis. Cada serviço oferecido pela plataforma é acompanhado de documentação técnica detalhada, incluindo guias de implementação passo-a-passo, exemplos de código e melhores práticas de desenvolvimento ([Firebase Development Team, 2023b](#)).

2.3.3 Modelo de Comercialização e Estrutura Tarifária

O Firebase opera sob um modelo de negócios freemium, caracterizado pela disponibilização gratuita de funcionalidades básicas combinada com planos pagos para recursos avançados e maior capacidade de processamento ([Google Cloud Platform, 2023b](#)). Esta estrutura tarifária escalonável permite que desenvolvedores individuais e pequenas equipes iniciem projetos sem investimento inicial, enquanto organizações com demandas mais robustas podem optar por planos comerciais adequados às suas necessidades específicas.

A estratégia de precificação adotada segue uma lógica de pay-as-you-scale, onde os custos são proporcionais ao volume de utilização dos recursos, incluindo métricas como número de usuários ativos, volume de dados armazenados e quantidade de operações de leitura/escrita realizadas ([MORONEY, 2022](#)). Esta abordagem visa garantir a viabilidade econômica tanto para projetos em fase inicial quanto para aplicações consolidadas com grande base de usuários.

2.3.4 Firebase Authentication

2.3.4.1 Caracterização do Serviço e Arquitetura Funcional

O Firebase Authentication constitui-se como um serviço especializado de autenticação e autorização de usuários, integrado ao ecossistema da plataforma Firebase, oferecendo uma solução abrangente de gerenciamento de identidade digital para aplicações móveis e web ([Google Firebase, 2023a](#)). Este serviço fundamenta-se em uma arquitetura de backend-as-a-service (BaaS) que disponibiliza infraestrutura completa de autenticação, eliminando a necessidade de desenvolvimento e manutenção de sistemas proprietários de gerenciamento de credenciais por parte dos desenvolvedores.

A arquitetura do Firebase Authentication baseia-se na provisão de serviços de backend robustos, complementados por kits de desenvolvimento de software (SDKs) otimizados e bibliotecas de interface de usuário pré-desenvolvidas ([Firebase Development Team, 2023a](#)). Esta abordagem integrada visa simplificar significativamente o processo de implementação de funcionalidades de autenticação, permitindo que desenvolvedores foquem na lógica de negócio específica de suas aplicações, enquanto delegam a complexidade técnica da autenticação para a infraestrutura especializada do Firebase.

2.3.4.2 Modalidades de Autenticação e Provedores de Identidade

O sistema oferece suporte abrangente a múltiplas modalidades de autenticação, atendendo às diversas preferências e contextos de uso dos usuários finais. A plataforma implementa autenticação tradicional baseada em credenciais de senha, permitindo que usuários criem contas através de endereços de email e senhas personalizadas ([STALLINGS, 2023](#)). Adicionalmente, o serviço incorpora funcionalidades de autenticação por

meio de números de telefone, utilizando protocolos de verificação por SMS para validação de identidade.

Uma característica distintiva do Firebase Authentication reside em sua capacidade de integração com provedores de identidade federados, implementando o conceito de Single Sign-On (SSO) através de plataformas estabelecidas no mercado ([JONES; SMITH; CHEN, 2022](#)). Entre os provedores suportados, destacam-se o Google, Facebook, Twitter, GitHub, Apple, Microsoft, Yahoo, além de suporte para provedores customizados através de protocolos padronizados. Esta diversidade de opções de autenticação visa atender às preferências individuais dos usuários e reduzir barreiras de entrada nas aplicações.

2.3.4.3 Integração Sistêmica e Conformidade com Padrões da Indústria

O Firebase Authentication foi projetado com foco na integração estreita com outros componentes do ecossistema Firebase, incluindo Firebase Firestore, Firebase Realtime Database, Firebase Cloud Functions e Firebase Analytics ([Firebase Team, 2023c](#)). Esta integração sistêmica permite a implementação de regras de segurança baseadas em identidade de usuário, controle de acesso granular a recursos e personalização de experiências com base no perfil de autenticação.

A arquitetura do serviço fundamenta-se na aderência rigorosa a padrões industriais reconhecidos para autenticação e autorização, particularmente os protocolos OAuth 2.0 e OpenID Connect ([HARDT, 2012](#)). O OAuth 2.0 constitui-se como o padrão de facto para autorização de aplicações web e móveis, proporcionando mecanismos seguros para concessão de acesso limitado a recursos sem exposição de credenciais. Complementarmente, o OpenID Connect representa uma camada de identidade construída sobre o OAuth 2.0, fornecendo informações de identidade de usuário de forma padronizada ([SAKIMURA et al., 2014](#)).

Esta conformidade com padrões estabelecidos facilita significativamente a integração do Firebase Authentication com sistemas de backend existentes, APIs de terceiros e infraestruturas corporativas, garantindo interoperabilidade e reduzindo a complexidade de implementação em arquiteturas híbridas ou migrações graduais de sistemas legados.

2.3.5 Firebase Firestore

2.3.5.1 Caracterização e Arquitetura do Sistema

O Firebase Firestore constitui-se como um banco de dados NoSQL (Not Only SQL) orientado a documentos, desenvolvido e mantido pela Google como componente central do ecossistema Firebase ([Google Cloud, 2023](#)). Esta solução de armazenamento fundamenta-se em uma arquitetura distribuída e escalável, projetada especificamente para atender às demandas de aplicações móveis e web modernas que requerem sincronização

de dados em tempo real entre múltiplos clientes e dispositivos (CHANG et al., 2008).

A arquitetura do Firestore baseia-se no modelo de dados orientado a documentos, onde as informações são organizadas em estruturas hierárquicas compostas por coleções e documentos (MongoDB Inc., 2023). Cada documento representa uma unidade de dados estruturada em formato JSON, contendo pares chave-valor que podem incluir tipos primitivos, arrays, objetos aninhados e referências a outros documentos. Esta flexibilidade estrutural permite a modelagem eficiente de dados complexos sem as restrições impostas pelos esquemas rígidos dos bancos de dados relacionais tradicionais.

2.3.5.2 Funcionalidades de Sincronização e Tempo Real

Uma característica distintiva do Firestore reside em suas capacidades de sincronização automática e atualizações em tempo real (Firebase Development Team, 2023c). O sistema implementa listeners de dados que permitem às aplicações cliente receber notificações instantâneas sempre que ocorrem alterações nos dados armazenados, eliminando a necessidade de polling manual ou requisições periódicas para verificação de atualizações. Esta funcionalidade é fundamental para o desenvolvimento de aplicações colaborativas, sistemas de mensagens instantâneas e interfaces que requerem refletir mudanças de estado imediatas.

O mecanismo de sincronização offline representa outro aspecto tecnológico significativo, permitindo que aplicações continuem operando mesmo durante períodos de conectividade intermitente ou ausente (TERRY, 2013). O Firestore mantém uma cache local dos dados utilizados pela aplicação, sincronizando automaticamente as alterações quando a conectividade é restabelecida, garantindo consistência eventual dos dados distribuídos.

2.3.5.3 Escalabilidade e Modelo de Precificação

O Firestore foi arquitetado para proporcionar escalabilidade horizontal automática, adaptando-se dinamicamente ao volume de dados e quantidade de operações sem intervenção manual dos desenvolvedores (Google Cloud Platform, 2023a). Esta característica é particularmente relevante para aplicações com padrões de uso imprevisíveis ou crescimento exponencial de usuários, onde a capacidade de processamento deve ajustar-se automaticamente à demanda.

O modelo de precificação adotado segue uma estrutura pay-per-operation, onde os custos são calculados com base no número de operações de leitura, escrita e exclusão realizadas, complementado por tarifas de armazenamento e transferência de dados (Firebase Team, 2023a). Esta abordagem granular de cobrança permite otimização de custos através do design eficiente de consultas e estratégias de cache, tornando-se economicamente viável tanto para projetos em escala reduzida quanto para aplicações empresariais de grande porte.

2.4 Compose Multiplatform

2.4.1 Fundamentação Conceitual e Arquitetural

O Compose Multiplatform constitui-se como um framework de desenvolvimento de interfaces de usuário fundamentado no paradigma declarativo, desenvolvido pela JetBrains como extensão multiplataforma do Jetpack Compose originalmente concebido pela Google para aplicações Android nativas ([JetBrains, 2023a](#)). Esta solução tecnológica representa uma evolução significativa na abordagem tradicional de desenvolvimento de interfaces, substituindo o modelo imperativo por uma metodologia declarativa onde os desenvolvedores descrevem o estado desejado da interface, delegando ao framework a responsabilidade de gerenciar as transições e atualizações necessárias.

A arquitetura do Compose Multiplatform fundamenta-se na filosofia de "escrever uma vez, executar em qualquer lugar" (write once, run anywhere), permitindo o compartilhamento substancial de código de interface de usuário entre diferentes plataformas, incluindo Android, iOS, desktop (Windows, macOS, Linux) e aplicações web ([Kotlin Foundation, 2023](#)). Esta abordagem unificada visa reduzir significativamente o esforço de desenvolvimento e manutenção de aplicações multiplataforma, eliminando a necessidade de implementação de interfaces específicas para cada sistema operacional de destino.

2.4.2 Paradigma Declarativo e Gerenciamento de Estado

O paradigma declarativo implementado pelo Compose Multiplatform representa uma mudança fundamental na conceptualização do desenvolvimento de interfaces de usuário. Diferentemente das abordagens imperativas tradicionais, onde desenvolvedores devem especificar explicitamente cada etapa de modificação da interface, o modelo declarativo permite a definição de funções composáveis que descrevem a aparência e comportamento da interface em função do estado atual da aplicação ([GAMMA; JOHNSON; LEE, 2023](#)).

O framework implementa um sistema sofisticado de gerenciamento de estado reativo, onde alterações nos dados da aplicação trigger automaticamente a recomposição seletiva dos componentes de interface afetados ([Android Developers, 2023b](#)). Este mecanismo de recomposição inteligente otimiza o desempenho através da identificação precisa dos elementos que necessitam atualização, evitando renderizações desnecessárias e garantindo eficiência computacional mesmo em interfaces complexas com grandes volumes de dados dinâmicos.

2.4.3 Interoperabilidade e Integração com Ecossistemas Nativos

Uma característica distintiva do Compose Multiplatform reside em sua capacidade de integração harmoniosa com componentes nativos das plataformas de destino,

permitindo a incorporação de funcionalidades específicas do sistema quando necessário ([JetBrains Development Team, 2023](#)). Esta flexibilidade arquitetural possibilita aos desenvolvedores aproveitar recursos únicos de cada plataforma, como APIs específicas do iOS ou componentes nativos do Android, sem comprometer os benefícios do desenvolvimento multiplataforma.

O framework suporta a criação de `expect/actual` declarations, um mecanismo que permite a definição de interfaces comuns esperadas em código compartilhado, com implementações específicas para cada plataforma ([Kotlin Team, 2023](#)). Esta funcionalidade é particularmente relevante para cenários que demandam acesso a recursos nativos específicos, como câmera, sensores, ou integrações profundas com o sistema operacional, mantendo a coesão arquitetural do código compartilhado.

2.4.4 Performance e Otimizações de Renderização

O Compose Multiplatform implementa um conjunto avançado de otimizações de performance destinadas a garantir experiências de usuário fluidas e responsivas em todas as plataformas suportadas. O sistema de renderização utiliza técnicas de virtualização para gerenciamento eficiente de listas extensas, lazy loading para componentes complexos, e algoritmos otimizados de diff para minimizar operações de layout e desenho ([Google Android Team, 2023](#)).

A arquitetura do framework incorpora conceitos de renderização baseada em canvas personalizado para cada plataforma, permitindo consistência visual absoluta entre diferentes sistemas operacionais, enquanto mantém performance nativa através da utilização de APIs de baixo nível específicas de cada plataforma ([Google Skia Team, 2023](#)). Esta abordagem híbrida garante tanto a uniformidade da experiência do usuário quanto a eficiência computacional necessária para aplicações demanding em recursos gráficos.

3 Trabalhos Relacionados

4 Metodologia e Cronograma

O abnT_EX introduziu o comando `IBGETab` para formatação de tabelas. Um exemplo de tabela convencional do L^AT_EX pode ser observado na Tabela 1 enquanto um exemplo usando o `IBGETab` é mostrado na Tabela 2.

Tabela 1 – Cronograma do Projeto em Meses

Atividade	1	2	3	4	5	6	7	8	9	10	11	12
Revisão Bibliográfica	•	•										
Métodos			•	•								
Testes				•	•	•						
Resultados							•	•				
Conclusão							•	•	•			
Banca										•	•	•

Fonte: Próprio Autor

Tabela 2 – Cronograma do Projeto em Meses usando o comando `IBGETab` para a formatação da tabela

Atividade	1	2	3	4	5	6	7	8	9	10	11	12
Revisão Bibliográfica	•	•										
Métodos			•	•								
Testes				•	•	•						
Resultados							•	•				
Conclusão							•	•	•			
Banca										•	•	•

Fonte: Próprio Autor

5 Resultados Esperados

Referências

ABLESON, W. F.; KUMAR, R. S. **Android Multi-Device Development: Building Apps for Phones, Tablets, and Wearables**. Shelter Island, NY: Manning Publications, 2023. ISBN 978-1633439870.

ABLESON, W. F.; SEN, R.; KING, C. **Android in Action**. 3. ed. Shelter Island, NY: Manning Publications, 2022.

Android Developers. **Android Emulator: Run Android Apps on Your Computer**. 2023. Acesso em: 24 ago. 2025. Disponível em: <<https://developer.android.com/studio/run/emulator>>.

Android Developers. **State and Jetpack Compose: Managing UI State in Declarative Applications**. 2023. Acesso em: 29 ago. 2025. Disponível em: <<https://developer.android.com/jetpack/compose/state>>.

Android Development Team. Instant run: Faster android development with live code updates. **Google Developers Blog**, jun 2022. Acesso em: 24 ago. 2025. Disponível em: <<https://android-developers.googleblog.com/2022/06/instant-run-faster-android-development.html>>.

Android Static Analysis Team. **Android Lint: Static Code Analysis for Android Applications**. [S.l.], 2023. Acesso em: 24 ago. 2025. Disponível em: <<https://developer.android.com/studio/write/lint>>.

BURNETTE, E. **Hello, Android: Introducing Google's Mobile Development Platform**. 4. ed. Dallas, TX: Pragmatic Bookshelf, 2021.

CHANG, F. et al. Bigtable: A distributed storage system for structured data. **ACM Transactions on Computer Systems**, ACM, v. 26, n. 2, p. 1–26, 2008.

Developer Economics. **Developer Economics Q2 2013: State of the Developer Nation**. [S.l.], 2013.

Firebase Development Team. **Firebase Authentication SDK Documentation**. 2023. Acesso em: 24 ago. 2025. Disponível em: <<https://firebase.google.com/docs/auth>>.

Firebase Development Team. **Firebase Implementation Guide: Getting Started with Firebase Services**. [S.l.], 2023. Acesso em: 24 ago. 2025. Disponível em: <<https://firebase.google.com/docs/guides>>.

Firebase Development Team. **Firestore Real-time Updates Documentation**. 2023. Acesso em: 24 ago. 2025. Disponível em: <<https://firebase.google.com/docs/firestore/query-data/listen>>.

Firebase Team. **Cloud Firestore Pricing**. 2023. Acesso em: 24 ago. 2025. Disponível em: <<https://firebase.google.com/pricing>>.

Firebase Team. **Firebase Documentation: Build Better Apps**. 2023. Acesso em: 24 ago. 2025. Disponível em: <<https://firebase.google.com/docs>>.

Firebase Team. **Firestore Platform: Build and Run Apps Users Love**. 2023. Acesso em: 24 ago. 2025. Disponível em: <<https://firebase.google.com/>>.

GAMMA, R.; JOHNSON, P.; LEE, S. **Reactive Programming: Building Responsive User Interfaces**. 3. ed. Shelter Island, NY: Manning Publications, 2023. ISBN 978-1617298765.

GitHub Inc. **GitHub Integration with Android Studio**. 2023. Acesso em: 24 ago. 2025. Disponível em: <<https://docs.github.com/en/get-started/using-git/about-git>>.

Google Android Team. **Performance Best Practices for Jetpack Compose**. 2023. Acesso em: 29 ago. 2025. Disponível em: <<https://developer.android.com/jetpack/compose/performance>>.

Google Cloud. **Cloud Firestore: NoSQL Document Database**. 2023. Acesso em: 24 ago. 2025. Disponível em: <<https://firebase.google.com/products/firestore>>.

Google Cloud Platform. **Cloud Firestore Scaling and Performance**. 2023. Acesso em: 24 ago. 2025. Disponível em: <<https://firebase.google.com/docs/firestore/best-practices>>.

Google Cloud Platform. **Firebase Pricing Plans and Usage Limits**. 2023. Acesso em: 24 ago. 2025. Disponível em: <<https://firebase.google.com/pricing>>.

Google Developers. **Android Studio: The Official IDE for Android Development**. 2023. Acesso em: 24 ago. 2025. Disponível em: <<https://developer.android.com/studio>>.

Google Firebase. **Firebase Authentication: Simple, Free Multi-Platform Sign-In**. 2023. Acesso em: 24 ago. 2025. Disponível em: <<https://firebase.google.com/products/auth>>.

Google Firebase. **Firebase Console: Project Management Interface**. 2023. Acesso em: 24 ago. 2025. Disponível em: <<https://console.firebase.google.com>>.

Google Inc. **Google Acquires Firebase to Help Developers Build Better Realtime Apps**. 2014. Acesso em: 24 ago. 2025. Disponível em: <<https://firebase.googleblog.com/2014/10/google-acquires-firebase-to-help.html>>.

Google Inc. **Android Developer Documentation**. 2023. Acesso em: 24 ago. 2025. Disponível em: <<https://developer.android.com/docs>>.

Google Skia Team. **Skia Graphics Engine: Cross-Platform 2D Graphics API**. 2023. Acesso em: 29 ago. 2025. Disponível em: <<https://skia.org/>>.

Gradle Inc. **Gradle Build Tool: Build Anything with Gradle**. 2023. Acesso em: 24 ago. 2025. Disponível em: <<https://gradle.org/>>.

HARDT, D. **The OAuth 2.0 Authorization Framework**. 2012. RFC 6749. Acesso em: 24 ago. 2025. Disponível em: <<https://tools.ietf.org/html/rfc6749>>.

JetBrains. **Compose Multiplatform: Declarative Framework for Sharing UIs Across Multiple Platforms**. 2023. Acesso em: 29 ago. 2025. Disponível em: <<https://www.jetbrains.com/lp/compose-multiplatform/>>.

JetBrains. **IntelliJ IDEA: The Java IDE for Professional Developers**. 2023. Acesso em: 24 ago. 2025. Disponível em: <<https://www.jetbrains.com/idea/>>.

JetBrains Development Team. **Compose Multiplatform Interoperability Guide**. 2023. Acesso em: 29 ago. 2025. Disponível em: <https://github.com/JetBrains/compose-multiplatform/blob/master/tutorials/Native_API_integration/README.md>.

JONES, M.; SMITH, S.; CHEN, W. Federated identity management in cloud computing: A comprehensive survey. **IEEE Transactions on Cloud Computing**, IEEE Computer Society, v. 10, n. 2, p. 234–251, 2022.

JUnit Team. **JUnit Testing Framework for Android Applications**. 2023. Acesso em: 24 ago. 2025. Disponível em: <<https://junit.org/junit5/>>.

Kotlin Foundation. **Kotlin Multiplatform: Share Code Between Mobile and Desktop**. 2023. Acesso em: 29 ago. 2025. Disponível em: <<https://kotlinlang.org/docs/multiplatform.html>>.

Kotlin Team. **Expected and Actual Declarations in Kotlin Multiplatform**. 2023. Acesso em: 29 ago. 2025. Disponível em: <<https://kotlinlang.org/docs/multiplatform-expect-actual.html>>.

LEE, M. J. Understanding android's open source architecture and commercial implementations. **Journal of Mobile Computing**, IEEE Computer Society, v. 15, n. 3, p. 45–62, 2022.

MEIER, R. **Professional Android Application Development**. 5. ed. Indianapolis, IN: Wrox Press, 2023. ISBN 978-1119793205.

MongoDB Inc. **What is NoSQL? NoSQL Databases Explained**. 2023. Acesso em: 24 ago. 2025. Disponível em: <<https://www.mongodb.com/nosql-explained>>.

MORONEY, L. **Firebase for Mobile Development: Build Scalable Real-time Apps**. Shelter Island, NY: Manning Publications, 2022. ISBN 978-1617299544.

MURPHY, M. L. **The Busy Coder's Guide to Android Development**. Ann Arbor, MI: CommonsWare, 2023.

Open Source Initiative. **The Apache License, Version 2.0**. 2023. Acesso em: 24 ago. 2025. Disponível em: <<https://opensource.org/licenses/Apache-2.0>>.

SAKIMURA, N. et al. **OpenID Connect Core 1.0 incorporating errata set 1**. 2014. Acesso em: 24 ago. 2025. Disponível em: <https://openid.net/specs/openid-connect-core-1_0.html>.

STALLINGS, W. **Cryptography and Network Security: Principles and Practice**. 8. ed. Boston, MA: Pearson, 2023. ISBN 978-0134444284.

StatCounter Global Stats. **Desktop vs Mobile vs Tablet Market Share Worldwide**. 2017. Acesso em: 24 ago. 2025. Disponível em: <<https://gs.statcounter.com/platform-market-share/desktop-mobile-tablet/worldwide>>.

TAMPLIN, J.; LEE, A. **Firebase: The Complete Guide to Building Real-time Applications**. 2. ed. Sebastopol, CA: O'Reilly Media, 2021.

TERRY, D. B. Replicated data consistency explained through baseball. **Communications of the ACM**, ACM, v. 56, n. 12, p. 82–89, 2013.