Faculdade de Tecnologia do Estado de São Paulo

Olivier Silva Erhard

Phishing Tracker

Trabalho de Conclusão do Curso de Análise e Desenvolvimento de Sistemas.
Professor Orientador: Sergio Luiz Banin

São Paulo
2025

Sumario

1. Introdução

1.1. O que é Phishing

Phishing é um termo sem tradução literal para o português, mas que pode ser entendido como uma espécie de **“pescaria de dados”**. Trata-se de uma técnica de ataque cibernético em que o invasor tenta enganar a vítima, se passando por uma entidade confiável — como bancos, empresas conhecidas ou órgãos governamentais — com o objetivo de induzi-la a fornecer informações sensíveis, como senhas, dados bancários ou documentos pessoais.

Esses ataques geralmente são realizados por meio de **mensagens fraudulentas** (e-mails, SMS, ou outros canais digitais), que contêm links maliciosos. Esses links levam a sites falsos, visualmente semelhantes aos originais, onde a vítima pode ser induzida a inserir seus dados ou, pior, acabar permitindo a instalação de **malwares, vírus ou programas espiões** em seu dispositivo.

É importante destacar que o *phishing* muitas vezes é apenas o **primeiro passo** de um ataque mais complexo. Após o clique no link, o invasor pode assumir o controle da máquina da vítima, monitorar suas atividades, roubar dados em segundo plano ou até utilizar o dispositivo como parte de uma rede de ataques (*botnet*).

Portanto, o *phishing* representa uma ameaça significativa à segurança digital, funcionando como a "porta de entrada" para diversos tipos de invasões mais sofisticadas e danosas.

1.2. Como funciona no ambiente mobile

Os dispositivos móveis — como smartphones com sistemas Android, iOS e, anteriormente, Windows Phone — estão cada vez mais presentes no cotidiano das pessoas, superando até mesmo o uso de computadores tradicionais (como desktops e notebooks). Por serem portáteis e estarem constantemente conectados à internet, esses dispositivos enviam e recebem informações em tempo real, interagindo com serviços online, ambientes empresariais, lojas virtuais e até com dispositivos domésticos inteligentes (como lâmpadas, câmeras, assistentes de voz, entre outros).

Essa constante conectividade e a **alta concentração de dados sensíveis** — como informações bancárias, documentos, contatos, fotos, geolocalização e histórico de navegação — tornam os smartphones um alvo extremamente atrativo para

atacantes cibernéticos. Diferente de um computador que pode ser usado de forma mais segmentada, o celular tende a centralizar tudo em um único dispositivo, o que **aumenta o impacto de um ataque bem-sucedido**.

Os atacantes exploram diversos canais para alcançar suas vítimas, como redes sociais, aplicativos de mensagens, e-mails, SMS e até chamadas telefônicas. Ao se aproveitarem da confiança do usuário e da integração entre aplicativos e permissões do sistema, eles podem induzir cliques em links maliciosos ou a instalação de aplicativos comprometidos.

Essa combinação de fatores — **alta presença, constante conexão, centralização de dados e múltiplos vetores de ataque** — transforma o ambiente mobile em um dos principais alvos para campanhas de phishing, malwares e outros tipos de ameaças digitais.

1.3. O que pode ser feito?

O primeiro passo para se proteger de ataques como o phishing é **reconhecer os sinais mais comuns** dessas ameaças. Em geral, esse tipo de ataque explora **situações falsas de urgência**, com o objetivo de forçar a vítima a clicar rapidamente em um link sem pensar. Os invasores costumam se passar por entidades confiáveis — como bancos, órgãos governamentais ou até mesmo contatos conhecidos — para enganar o usuário.

O objetivo mais comum desses ataques é a **obtenção de dados bancários ou financeiros**, mas isso pode variar conforme o perfil da vítima. Por exemplo, executivos ou funcionários de grandes empresas podem ser alvos de ataques com fins de espionagem corporativa ou sabotagem. Apesar disso, o **ganho financeiro direto continua sendo a principal motivação** da maioria dos atacantes.

Os atacantes disfarçam suas armadilhas de formas muito convincentes. No caso de empresas, podem enviar mensagens informando sobre documentos pendentes ou faturas em atraso. Já em simulações de bancos, é comum o envio de notificações sobre "tentativas suspeitas de acesso" ou "transações não autorizadas", induzindo o usuário ao pânico e ao clique imediato.

Apesar do alto nível de sofisticação de muitos ataques, **existem sinais que ajudam a identificar um possível phishing**, como:

- **Endereço de e-mail do remetente**: atacantes muitas vezes tentam imitar o domínio oficial da empresa, mas não conseguem replicá-lo exatamente. Pequenas variações ou domínios genéricos (como @gmail.com) são sinais de

alerta.

- **Conteúdo genérico ou impessoal**: empresas legítimas normalmente se dirigem ao usuário pelo nome. Mensagens muito genéricas (como "Prezado cliente") podem indicar um golpe.

- **Erros gramaticais e ortográficos**: muitos e-mails de phishing contêm erros que não seriam aceitáveis em comunicações oficiais.

- **Tom alarmista**: frases como "sua conta será bloqueada em 24 horas" ou "ação imediata necessária" são comuns e têm o objetivo de pressionar a vítima a agir sem refletir.

Estar atento a esses detalhes é essencial para **evitar cair em golpes** e preservar a segurança dos dados pessoais e corporativos.

2. Justificativa

A segurança da informação é essencial em qualquer ambiente, seja físico ou virtual. Com o avanço da digitalização, empresas passaram a operar em ambos os contextos, o que aumenta a superfície de exposição a ameaças. Um dos ataques mais comuns atualmente é o **phishing**, incluindo sua variação mais sofisticada, o **spear phishing**, que tem como alvo profissionais de alto escalão. Nesse tipo de ataque, os criminosos se passam por entidades confiáveis — como prestadoras de serviço, instituições financeiras ou órgãos governamentais — com o objetivo de enganar o usuário e induzi-lo a clicar em links maliciosos.

Esses links, muitas vezes disfarçados em comunicações legítimas, podem direcionar o usuário a sites que contêm códigos maliciosos capazes de roubar dados sensíveis, monitorar atividades online e comprometer a integridade dos dispositivos. Em ambientes corporativos, isso representa um risco significativo tanto para os dados da empresa quanto para a privacidade dos funcionários.

Diante desse cenário, torna-se evidente a necessidade de mecanismos que permitam verificar a confiabilidade de links e informações recebidas, criando uma camada adicional de proteção contra esse tipo de ameaça. Considerando que grande parte da comunicação entre empresas e colaboradores ocorre por meio de dispositivos móveis, propõe-se o desenvolvimento de um aplicativo mobile focado na **verificação de links suspeitos e na proteção de dados sensíveis**, contribuindo para a prevenção de ataques e o fortalecimento da segurança digital nas organizações.

3. Objetivo do Sistema

3.1. Objetivo Geral:

Desenvolver um aplicativo mobile capaz de realizar a verificação de links suspeitos, informando ao usuário a **probabilidade de o link ser confiável ou malicioso**, com o objetivo de prevenir ataques de phishing e spear phishing em ambientes corporativos e pessoais.

3.2. Objetivos Específicos:

- Integrar fontes confiáveis (entidades de reputação digital) que realizam a verificação prévia de links, identificando indicadores de risco como coleta excessiva de dados, histórico de abuso ou alertas de segurança.

- Implementar um sistema de análise que agregue essas informações e gere uma **classificação de risco acessível ao usuário**, com base em critérios objetivos.

- Verificar a **localização geográfica do servidor de origem do link**, alertando o usuário caso haja inconsistência com a suposta identidade da entidade (ex: sites que se apresentam como nacionais, mas hospedados em países sem relação com o conteúdo).
- Oferecer uma interface simples e de fácil utilização para que o usuário se sinta mais confortável e faça essa verificação de forma rápida e simples

4. Objetivo da Aplicação

O objetivo da aplicação é tentar garantir uma nova barreira para o atacante, garantindo um pouco de segurança ao usuario, fazendo com que tenha uma opção a mais para casos de duvidas. Analizando o remetente, o conteudo da mensagem e o link para tentar identificar alguma inconsistência em algum desses pontos.

Identificar ameaças à segurança em mensagens(para phishing)

4.1. Estudo de viabilidade; Estado atual

Alternativas(nenhum com um aplicativo Android específico pra isso)

- Bitlock link defender, norton safeweb,

4.1.1. Descrição do Sistema Proposto

O sistema proposto consiste em uma **aplicação mobile para Android** voltada à **análise de mensagens suspeitas**, com o objetivo de ajudar o usuário a identificar possíveis tentativas de *phishing* ou outros tipos de golpes digitais. A aplicação funcionará como uma **barreira preventiva**, permitindo que o usuário envie uma mensagem suspeita para que ela seja verificada por múltiplos critérios de segurança.

O processo de funcionamento será dividido em três etapas principais:

1. **Recebimento da mensagem**:
   O usuário poderá encaminhar à aplicação o conteúdo de uma mensagem suspeita (por exemplo, recebida por e-mail, SMS, ou aplicativo de mensagens). O sistema irá extrair automaticamente o texto da mensagem e qualquer **link embutido** nela.

2. **Análise da mensagem**:
   Utilizando técnicas de **processamento de linguagem natural (IA)**, a aplicação analisará o conteúdo textual da mensagem em busca de padrões típicos de golpes — como tom de urgência, vocabulário genérico, erros gramaticais e expressões alarmistas. A partir disso, será gerado um **índice de risco textual**.

3. **Verificação do link**:
   O link extraído será submetido a diversas camadas de verificação:

- **Consultas a listas negras públicas** (como bases de dados de domínios maliciosos).
- **Verificação da reputação do domínio**, utilizando serviços de terceiros como Norton Safe Web, Google Safe Browsing, entre outros.
- **Geolocalização do servidor do link**, com o objetivo de identificar se ele está hospedado em países incomuns ou associados a fraudes, principalmente quando o domínio tenta simular uma entidade nacional.

Com base nessas verificações, a aplicação apresentará ao usuário um **relatório consolidado**, que indicará:

- O grau de confiabilidade da mensagem.
- O risco associado ao link.
- A origem geográfica do domínio.
- Sinais encontrados de possível fraude.

O foco principal da aplicação não é bloquear automaticamente os links ou impedir a navegação, mas sim **informar e conscientizar o usuário** sobre os riscos antes que ele interaja com conteúdos potencialmente maliciosos.

A proposta visa preencher uma lacuna no mercado de aplicativos mobile, já que **não há atualmente soluções integradas específicas para análise de mensagens suspeitas voltadas diretamente ao público comum no ambiente Android**, especialmente com abordagem centrada em IA e geolocalização como barreira adicional.

4.1.2. Benefícios do Sistema Proposto

A aplicação desenvolvida visa oferecer diversos benefícios tanto em termos de segurança digital quanto de usabilidade e prevenção de ataques. Abaixo, são listados os principais pontos positivos do sistema:

1. Aumento da segurança no ambiente mobile

O sistema atua como uma camada adicional de proteção contra golpes digitais, especialmente ataques de phishing, que são comuns em dispositivos móveis. Ao permitir a análise de links e mensagens suspeitas, contribui para a preservação de dados sensíveis do usuário, como informações bancárias e credenciais pessoais.

2. Detecção baseada em múltiplas fontes de verificação

A verificação dos links é realizada por meio de consultas a serviços externos de reputação, listas negras públicas e análise de geolocalização do servidor de destino. Essa abordagem torna a detecção mais precisa e confiável, reduzindo falsos positivos ou negativos.

3. Análise contextual com uso de inteligência artificial

A aplicação utiliza processamento de linguagem natural (NLP) para interpretar o conteúdo textual da mensagem, identificando padrões típicos de engenharia social, como tentativas de urgência, ambiguidade no destinatário ou erros ortográficos, que são característicos de ataques de phishing.

4. Geolocalização como fator de análise de risco

A localização do servidor onde o link está hospedado é usada como critério adicional de verificação. Isso permite identificar domínios que tentam simular entidades nacionais (como bancos ou órgãos públicos), mas que estão hospedados em países com histórico de fraudes, aumentando a precisão da análise.

5. Conscientização e orientação do usuário

O sistema não apenas informa sobre o risco de determinado link, mas também explica os fatores identificados na análise. Dessa forma, promove a conscientização e ajuda o usuário a desenvolver uma postura mais crítica em relação a mensagens suspeitas.

6. Interface acessível e fácil utilização

A solução foi desenvolvida com foco em usabilidade, permitindo que qualquer usuário, mesmo sem conhecimento técnico, possa analisar mensagens com poucos toques na tela. Isso amplia o alcance da solução e facilita sua adoção.

4.2. Facilidades do Sistema Proposto

O sistema apresenta diversas facilidades que o tornam uma solução prática e eficiente no combate a ameaças digitais, especialmente no contexto mobile. A seguir, são destacadas as principais:

1. Uso simplificado

A interface da aplicação foi projetada com foco na usabilidade, permitindo que qualquer pessoa, mesmo sem conhecimentos técnicos, possa utilizar a ferramenta com facilidade. O processo de verificação é intuitivo, exigindo apenas o envio da mensagem suspeita para análise.

2. Portabilidade

Por se tratar de um aplicativo desenvolvido para dispositivos móveis (inicialmente para Android), o sistema pode ser utilizado em qualquer lugar, a qualquer momento, sem necessidade de dispositivos adicionais, atendendo às necessidades de mobilidade dos usuários.

3. Instalação e atualização simples

A aplicação pode ser disponibilizada em lojas oficiais, como a Google Play Store, permitindo instalação segura e atualizações automáticas, o que facilita a manutenção e a distribuição de melhorias futuras.

4. Integração com serviços externos

O sistema foi projetado para consultar APIs e bancos de dados públicos de reputação de domínios, o que permite que a verificação seja feita com base em informações atualizadas e confiáveis, sem necessidade de infraestrutura local pesada.

5. Baixo consumo de recursos

A aplicação realiza suas análises com baixo uso de processamento e memória, o que a torna viável para execução em dispositivos com configurações modestas, ampliando sua acessibilidade.

6. Independência de plataforma para análise

Embora a aplicação seja mobile, as verificações de links e conteúdo podem ser feitas por meio de serviços externos, possibilitando eventual reaproveitamento da lógica em outras plataformas no futuro, como aplicações web ou extensões para navegadores.

4.3. Investimento e Retorno

Durante o desenvolvimento do projeto, não houve necessidade de investimento financeiro direto, uma vez que foram utilizadas ferramentas gratuitas e recursos próprios. No entanto, para viabilizar a aplicação em um ambiente de produção e escalar seu uso, é necessário considerar alguns custos operacionais.

4.3.1. Tabela 1- Investimento e retorno

Item	Descrição	Custo Estimado	Observações
Hospedagem em nuvem	Utilização de serviços como o Firebase ou Google Cloud Functions para Backend e API´s	Gratuito(Até Limite)	Plano gratuito suficiente para uso leve e testes; pode escalar futuramente
API do VirusTotal	Verificação da reputação de URLs	Gratuito (até 500 req/dia)	Como o uso será limitado, não haverá custo inicialmente
Geolocalização de IP/Domínio	Serviços de IP lookup gratuitos (ex: ip-api, ipinfo com plano grátis)	Gratuito	Limites diários suficientes para uso moderado
Banco de dados Firebase	Banco de dados não relacional	Gratuito(até certo limite)	Plano gratuito até certo limite, após isso paga-se por uso

4.4. Desfecho da viabilidade do sistema

A aplicação proposta configura-se como uma solução viável, eficaz e inovadora no enfrentamento das ameaças digitais que afetam o ambiente mobile, especialmente os ataques de phishing. Em um cenário onde dispositivos móveis são cada vez mais utilizados como meio principal de comunicação e gestão de informações pessoais e corporativas, garantir mecanismos de segurança específicos para esse contexto torna-se essencial.

A proposta se diferencia por integrar múltiplas camadas de verificação: análise textual por meio de inteligência artificial, consulta a bancos de dados públicos e serviços especializados de reputação de domínios, além da verificação geográfica da origem dos servidores. Essa combinação permite detectar não apenas links maliciosos, mas também padrões comportamentais típicos de engenharia social, aumentando significativamente a eficácia da ferramenta.

Além do aspecto técnico, a aplicação possui um forte componente educativo, ao apresentar ao usuário, de forma clara e acessível, os fatores de risco identificados em uma mensagem. Com isso, não apenas protege, mas também contribui para a conscientização e alfabetização digital da população em relação a ameaças cibernéticas.

A ausência de soluções consolidadas que integrem essa abordagem voltada especificamente ao público mobile Android reforça a relevância e a originalidade do projeto. Assim, o sistema proposto não só se apresenta como uma barreira adicional de segurança, mas também como um passo em direção ao fortalecimento da cultura de cibersegurança no uso cotidiano da tecnologia.

5. Plano de Projeto

A integração do LOC (Linhas de Código) com os pontos por função (Casos de Uso) é uma abordagem eficaz para estimar o tamanho e a complexidade de um projeto de software. Ao combinar esses dois métodos de medição, podemos obter uma estimativa mais precisa do esforço necessário para desenvolver o sistema.

No contexto do projeto em questão, foram atribuídos pontos de função aos casos de uso com base em seu grau de dificuldade funcional, variando de 1 (Baixa) a 5 (Alta), como mostrado abaixo:

5.1. Tabela 1 – Grau de dificuldade funcional e os respectivos pontos de função atribuídos

Grau de Dificuldade	Pontos de Função
Baixa	1
Média-Baixa	2
Média	3
Média-Alta	4
Alta	5

5.1.1. Tabela 2 – Número de linhas esperadas por ponto de função

Pontos de Função	Linhas de Código
1	500
2	800
3	1000
4	1300
5	1600

5.1.2. Tabela 3 – Casos de uso do sistema e grau de dificuldade funcional

Caso de Uso	Grau de Dificuldade	Pontos de Função
Consultar Link	Média	3
Gerar Relatório	Alta	5
Consultar Relatório	Média	3
Denunciar Link	Média-Baixa	2

5.1.3. Tabela 4- LOC + Pontos por função (Casos de uso)

Grau de Dificuldade	Linhas de Código por Caso de Uso	Quantidade de Casos de Uso	Total de Linhas de Código
2 (Média-Baixa)	800	1	800
3 (Média)	1000	2	2000
5 (Alta)	1600	1	1600
Total		4	4400

Essa abordagem quantitativa nos permite ter uma compreensão melhor do esforço de desenvolvimento necessário para implementar o sistema proposto, facilitando o planejamento do cronograma e a alocação de recursos. Ao considerar tanto a complexidade funcional quanto o tamanho do código, podemos realizar uma estimativa mais precisa do projeto e tomar decisões mais informadas durante o processo de desenvolvimento.

5.2. Viabilidade Técnica

A viabilidade técnica do sistema proposto foi avaliada com base nas tecnologias selecionadas, nos recursos disponíveis e nos conhecimentos da equipe de desenvolvimento. O projeto será desenvolvido utilizando a linguagem **Kotlin**, que

proporciona uma sintaxe moderna, segura e compatível com o ecossistema Android. A interface do usuário será construída com **Jetpack Compose**, uma ferramenta declarativa moderna para construção de telas, que simplifica o desenvolvimento visual e melhora a manutenção do código.

A aplicação será estruturada com base nos princípios da arquitetura **SOLID**, utilizando **orientação a objetos** e **injeção de dependências** para garantir modularidade, reutilização de código e facilidade de testes. Isso contribui diretamente para a escalabilidade e manutenibilidade do sistema ao longo do tempo.

Para autenticação de usuários, será utilizado o **Firebase Authentication**, serviço confiável e seguro, que facilita o gerenciamento de credenciais e o controle de acesso à aplicação. O **Firebase** também será utilizado como **banco de dados**, oferecendo escalabilidade, sincronização em tempo real e integração facilitada com o ecossistema Android.

A verificação de links será realizada por meio da integração com **APIs de terceiros**, que fornecem dados sobre reputação e potencial risco de URLs. Esse recurso é essencial para garantir a precisão na identificação de links suspeitos ou perigosos, objetivo central do sistema.

A API será hospedada na **Google Cloud Platform (GCP)**, aproveitando os recursos de escalabilidade, disponibilidade e segurança da nuvem do Google. Caso seja necessário implementar monitoramento e análise de desempenho da aplicação.

Considerando o conjunto de tecnologias adotadas — todas modernas, bem documentadas e amplamente utilizadas na indústria —, assim como a experiência da equipe e a infraestrutura disponível, conclui-se que o desenvolvimento do sistema é **totalmente viável do ponto de vista técnico**.

5.3. Arquitetura do sistema

A arquitetura de software define a estrutura e o comportamento dos componentes de um sistema, bem como a forma como eles interagem. A escolha de uma arquitetura adequada é crucial para garantir a escalabilidade, manutenibilidade, segurança e desempenho do aplicativo, alinhando-se diretamente com os requisitos funcionais e não funcionais estabelecidos.

Para o desenvolvimento do aplicativo verificador de links, foi adotada uma **Arquitetura Cliente-Servidor Híbrida**, combinando os pontos fortes de uma aplicação mobile nativa (cliente) com a flexibilidade e escalabilidade dos serviços em nuvem (servidor) fornecidos pelo Google Firebase e Google Cloud Platform. Além disso, no lado do

cliente (aplicativo mobile), o padrão de arquitetura **Model-View-ViewModel (MVVM)** foi implementado para promover a separação de responsabilidades e facilitar o desenvolvimento e a testabilidade.

5.3.1. Arquitetura Geral Cliente-Servidor Híbrida

Esta arquitetura é composta pelos seguintes componentes principais:

- **Aplicativo Mobile (Cliente):** Desenvolvido em Kotlin com Jetpack Compose, este é o ponto de interação direto com o usuário. É responsável por coletar a entrada (mensagens/links), exibir os relatórios de análise, gerenciar o histórico local e orquestrar as chamadas para o backend. Implementa o padrão MVVM internamente para gerenciar sua lógica de apresentação.
- **Backend em Nuvem (Servidor - Firebase/GCP):** Atua como o cérebro central do sistema, gerenciando a lógica de negócio principal e a integração com serviços externos. Utiliza:
    - **Firebase Authentication:** Para a gestão segura de usuários e controle de acesso.
    - **Firebase Firestore (ou Realtime Database):** Como banco de dados NoSQL para persistência dos dados (usuários, histórico de análises, links denunciados). A natureza NoSQL do Firebase oferece alta escalabilidade e sincronização de dados em tempo real, ideal para aplicações mobile dinâmicas.
    - **Google Cloud Functions (GCP Functions):** Embora não explicitamente detalhado para cada funcionalidade, a utilização de Cloud Functions é altamente recomendada para orquestrar as chamadas a múltiplas APIs externas (como Google Safe Browse, VirusTotal, APIs de geolocalização). Isso centraliza a lógica de integração, reduz a complexidade no cliente, permite maior controle sobre as chaves de API e melhora a segurança ao evitar que o aplicativo mobile se conecte diretamente a todas as APIs externas. As Functions atuam como um *middleware* serverless.

- **APIs Externas de Verificação:** São serviços de terceiros consultados pelo backend (via Cloud Functions) para obter informações críticas sobre links e domínios. Incluem:
    - **Google Safe Browse API:** Para verificação de URLs contra listas de sites perigosos.
    - **VirusTotal API:** Para análise abrangente de arquivos e URLs, incluindo reputação e detecções de antivírus.
    - **Serviços de Geolocalização de IP (ex: IP-API, IPinfo):** Para identificar a localização geográfica dos servidores de origem dos links.

    - **Serviços de Processamento de Linguagem Natural (NLP):** Caso a análise textual da mensagem seja sofisticada e exija um serviço específico para identificar padrões de phishing no conteúdo.

5.3.2. Padrão Model-View-ViewModel (MVVM) no Cliente Mobile

Dentro do aplicativo mobile, o padrão MVVM foi escolhido para estruturar o código. Este padrão arquitetural promove a separação de responsabilidades em três componentes principais:

- **View (Visão):** Representa a interface do usuário (UI). No contexto do Jetpack Compose, são as `Composables` que exibem os dados e capturam as interações do usuário. A View é passiva e não contém lógica de negócio ou de estado complexa; ela apenas "observa" o ViewModel.
- **ViewModel:** Atua como um intermediário entre a View e o Model. Ele expõe dados (observáveis) que a View pode exibir e processa as ações do usuário da View, convertendo-as em operações no Model. O ViewModel mantém o estado da UI e sobrevive a mudanças de configuração (rotações de tela), garantindo que os dados não sejam perdidos. Contém a lógica de apresentação e coordenação.
- **Model:** Representa a camada de dados e a lógica de negócio do aplicativo. Isso inclui:
    - **Repositórios:** Abstraem a origem dos dados (local, remoto – Firebase, APIs externas). São responsáveis por buscar, armazenar e gerenciar dados, fornecendo uma interface limpa para o ViewModel.
    - **Modelos de Dados (Entities):** Classes que representam as estruturas de dados do aplicativo (ex: `Link`, `Mensagem`, `Relatorio`), geralmente alinhadas com as entidades do modelo de dados.
    - **Lógica de Negócio:** Regras e processos que governam a funcionalidade do aplicativo, como a consolidação dos resultados das diferentes APIs de verificação para gerar um relatório final.

5.3.3. Benefícios da Arquitetura MVVM:

- **Separação de Responsabilidades:** Facilita a manutenção e o desenvolvimento paralelo.
- **Testabilidade:** O ViewModel e o Model podem ser testados independentemente da View, o que simplifica e agiliza os testes unitários.
- **Reusabilidade:** A lógica de negócio no Model e a lógica de apresentação no ViewModel podem ser reutilizadas em diferentes Views.

- **Gerenciamento de Estado:** O ViewModel lida com o estado da UI de forma robusta, melhorando a experiência do usuário.

A combinação da arquitetura Cliente-Servidor Híbrida com o Firebase/GCP e o padrão MVVM no cliente mobile permite um sistema robusto, escalável e com alta capacidade de manutenção, apto a lidar com as demandas de segurança e análise de links em ambientes dinâmicos.

6. Requisitos funcionais e não funcionais

6.1. Requisitos funcionais

1. Entrada e Captura de Dados

- RF 01: Permitir que o usuário insira manualmente um texto ou mensagem no aplicativo.
- RF 02: Permitir que o usuário cole uma mensagem recebida de outras fontes.

2. Identificação e Separação de Componentes

- RF 03: Permitir a análise independente do conteúdo textual e do link, caso ambos estejam presentes.

3. Análise do Conteúdo da Mensagem

- RF 04: Permitir a visualização do texto da mensagem sem os links.
- RF 05: Realizar análise textual simples, incluindo:
    - Detecção da língua da mensagem.
    - Extração de palavras-chave.

4. Análise de Links

- RF 06: Extrair e exibir o link completo presente na mensagem.
- RF 07: Identificar informações básicas sobre o domínio do link, como:
    - Domínio principal.
    - Extensão (.com, .org, etc.).
- RF 08: Verificar se o link está ativo ou inativo (status HTTP).
- RF 09: Obter a geolocalização básica do servidor onde o link está hospedado, informando país e cidade (se disponível).
- RF 10: Exibir o endereço IP do servidor do link.

5. Visualização de Dados

- RF 11: Exibir os dados coletados sobre o link e a mensagem de forma estruturada na interface.

6. Gerenciamento de Análises

- RF 12: Armazenar o histórico das análises realizadas localmente no dispositivo.
- RF 13: Permitir ao usuário visualizar e excluir análises anteriores.

7. Compartilhamento e Exportação

- RF14: Permitir ao usuário compartilhar os resultados da análise por meio de texto simples (ex.: copiar e colar em outros aplicativos).

6.2. Requisitos não funcionais

1. Usabilidade

- RNF 01: O sistema deve ter uma interface simples, intuitiva e adaptada para dispositivos móveis.
- RNF 02: Deve apresentar feedback claro ao usuário durante os processos, como carregamento ou erros.

3. Segurança

- RNF 03: As requisições feitas para APIs externas devem ser realizadas por meio de conexões seguras (HTTPS).
- RNF 04: O aplicativo não deve armazenar informações sensíveis dos usuários sem consentimento.

4. Compatibilidade e Portabilidade

- RNF 05: O aplicativo deve ser compatível com dispositivos Android versão 8.0 (Oreo) ou superior.
- RNF 06: O aplicativo deve funcionar em redes Wi-Fi e redes móveis (3G, 4G, 5G).

5. Manutenibilidade

- RNF 07: O sistema deve ter código organizado de forma modular, permitindo facilidade na manutenção e expansão futura.

6. Confiabilidade

- RNF 08: O sistema deve ser capaz de informar ao usuário quando ocorrerem falhas na análise ou na comunicação com APIs externas.

- RNF 09: As análises realizadas devem permanecer salvas no dispositivo, mesmo que haja falha de conexão no momento da consulta.

7. Principais Funcionalidades

7.1. Análise Abrangente de Links e Mensagens

A funcionalidade central do Phishing Tracker é a capacidade de analisar links e o conteúdo textual de mensagens para identificar potenciais ameaças de phishing. O usuário pode inserir ou colar mensagens suspeitas no aplicativo. O sistema então processa separadamente o texto e os links, utilizando:

- Processamento de Linguagem Natural (PNL): Para analisar o conteúdo textual da mensagem, buscando padrões típicos de golpes como tom de urgência, erros gramaticais e vocabulário genérico.
- Verificação de Reputação de Domínio: Consultas a serviços de terceiros como Google Safe Browse e VirusTotal são realizadas para verificar se o link está associado a atividades maliciosas ou a listas negras.
- Geolocalização do Servidor: O sistema verifica a localização geográfica do servidor onde o link está hospedado, alertando o usuário sobre inconsistências, como um site que se apresenta como nacional, mas está hospedado em um país com histórico de fraudes.
- Informações Técnicas do Link: Exibe o domínio principal, a extensão e o endereço IP do servidor do link, além de verificar se o link está ativo.

7.2. Geração de Relatórios Detalhados de Risco

Após a análise, o Phishing Tracker gera um relatório consolidado e de fácil compreensão para o usuário. Este relatório fornece uma avaliação clara do risco associado à mensagem e ao link, informando:

- O grau de confiabilidade geral da mensagem.
- O risco específico associado ao link.
- A origem geográfica do domínio do link.
- Os sinais encontrados que indicam uma possível fraude (ex: tom alarmista, erros ortográficos, inconsistência de localização do servidor). Essa funcionalidade não apenas informa sobre o risco, mas também atua como uma ferramenta de conscientização, ajudando o usuário a entender os indicadores de phishing.

7.3. Histórico e Gerenciamento de Análises

O aplicativo oferece a capacidade de armazenar e gerenciar um histórico das análises realizadas diretamente no dispositivo do usuário. Isso permite:

- Visualização de Análises Anteriores: O usuário pode consultar relatórios de análises passadas, o que é útil para referência ou para rever informações importantes.
- Exclusão de Análises: A funcionalidade de exclusão garante que o usuário tenha controle sobre seus dados e privacidade, podendo remover análises antigas conforme

desejar. Essa funcionalidade contribui para a usabilidade e a conveniência, permitindo que o usuário acompanhe suas verificações ao longo do tempo.

7.4. Compartilhamento de Resultados e Feedback

O Phishing Tracker permite que o usuário compartilhe os resultados da análise de forma simples. Embora o documento mencione apenas a opção de copiar e colar o texto da análise, a funcionalidade de "Denunciar Link" sugere um potencial para contribuição colaborativa, onde usuários podem reportar links maliciosos. Este aspecto é crucial para:

- Disseminação da Informação: Usuários podem alertar amigos, familiares ou colegas sobre links perigosos, aumentando a proteção na comunidade.
- Melhora Contínua do Sistema: A capacidade de denunciar links pode, no futuro, alimentar bases de dados e aprimorar a detecção de novas ameaças, tornando o sistema mais robusto e adaptável aos ataques emergentes.

8. Tecnologias Utilizadas

O Phishing Tracker é desenvolvido com um conjunto de tecnologias modernas e eficientes, garantindo um sistema robusto, escalável e seguro. A arquitetura é dividida entre o aplicativo mobile e o backend em nuvem, com integrações essenciais a APIs externas, e tudo isso orquestrado com Docker para um ambiente consistente.

1. Desenvolvimento Mobile (Cliente)

- Kotlin: A principal linguagem de programação para o desenvolvimento do aplicativo Android. Kotlin é reconhecida por sua sintaxe moderna, concisa e segura, além de ser totalmente interoperável com Java e a linguagem preferida do Google para o ecossistema Android.
- Jetpack Compose: Utilizado para a criação da interface do usuário (UI). Este toolkit UI declarativo simplifica e acelera o desenvolvimento de telas, resultando em um código mais limpo e reativo.
- Arquitetura MVVM (Model-View-ViewModel): Um padrão arquitetural implementado no cliente mobile para garantir a separação de responsabilidades. Essa abordagem melhora a manutenibilidade, a testabilidade e a escalabilidade do código.

2. Backend e Infraestrutura em Nuvem

- Python com FastAPI: O coração do backend é desenvolvido em Python utilizando o FastAPI. FastAPI é um framework web moderno, rápido (de alto desempenho) e que oferece documentação interativa automática (compatível com OpenAPI e JSON Schema), ideal para a criação de APIs robustas e fáceis de manter.
- Docker: Utilizado para conteinerização do backend. O Docker garante que o ambiente de execução do FastAPI seja consistente desde o desenvolvimento até a produção, facilitando o deploy, a escalabilidade e a gestão das dependências. Isso é crucial para a portabilidade e a agilidade da implantação do sistema.
- Google Firebase: Uma plataforma abrangente do Google que fornece diversos serviços para o backend da aplicação.
    - Firebase Authentication: Gerencia o processo de autenticação de usuários de forma segura e simplificada, controlando o acesso à aplicação.
    - Firebase Firestore (ou Realtime Database): Utilizado como banco de dados NoSQL para a persistência de dados, como informações de usuários, histórico de análises e links denunciados. Sua natureza NoSQL garante alta escalabilidade e sincronização de dados em tempo real, características cruciais para aplicações mobile dinâmicas.
- Google Cloud Platform (GCP): A infraestrutura de nuvem do Google que complementa o Firebase. As funções do backend, conteinerizadas com Docker, seriam implantadas e executadas na GCP, aproveitando seus recursos de escalabilidade e alta disponibilidade.

3. Integração com APIs Externas

O sistema se integra com diversas APIs de terceiros para obter informações cruciais sobre links e domínios. Essas integrações são gerenciadas pelo backend:

- Google Safe Browse API: Utilizada para verificar URLs contra listas de sites perigosos conhecidos por hospedar malware, phishing ou softwares indesejados.
- VirusTotal API: Para uma análise mais abrangente de URLs, incluindo relatórios de reputação, detecções de antivírus e outros indicadores de segurança.
- Serviços de Geolocalização de IP (ex: IP-API, IPinfo): Permitem identificar a localização geográfica dos servidores de origem dos links, um fator importante para a análise de risco, especialmente para detectar inconsistências.
- Serviços de Processamento de Linguagem Natural (PNL): Embora o serviço específico não tenha sido detalhado, a menção de PNL no documento indica a possível integração com ferramentas que auxiliem na análise textual do conteúdo da mensagem para identificar padrões de phishing.

9. Segurança

A segurança é um pilar fundamental no desenvolvimento do Phishing Tracker, garantindo a proteção tanto dos usuários quanto da integridade do sistema. Diversas camadas de segurança são implementadas, abrangendo desde a comunicação até o armazenamento de dados e a arquitetura do aplicativo.

1. Comunicação Segura (HTTPS)

Todas as requisições realizadas entre o aplicativo mobile, o backend (FastAPI em Python) e as APIs externas de verificação são feitas exclusivamente através de conexões seguras HTTPS. Isso criptografa os dados em trânsito, protegendo-os contra interceptação, adulteração e espionagem por terceiros mal-intencionados. Este requisito não funcional (RNF 03) é crucial para a confidencialidade das informações trocadas.

2. Gerenciamento de Autenticação e Autorização

O sistema utiliza o Firebase Authentication para o gerenciamento seguro de usuários. Isso garante que apenas usuários autenticados possam acessar funcionalidades específicas, como o armazenamento de histórico de análises. A autenticação robusta protege o acesso aos dados do usuário e evita o uso indevido da API do backend.

3. Proteção e Armazenamento de Dados

A segurança dos dados do usuário é uma prioridade. O Phishing Tracker adota as seguintes práticas:

- Não Armazenamento de Informações Sensíveis Sem Consentimento (RNF 04): O aplicativo é projetado para não coletar ou armazenar dados pessoais sensíveis dos usuários sem autorização explícita. O foco é na análise de links e mensagens, e não na coleta de informações do usuário.
- Firebase Firestore/Realtime Database: O uso desses bancos de dados do Firebase proporciona um ambiente seguro para o armazenamento de informações de análises e históricos, aproveitando as robustas políticas de segurança e a infraestrutura do Google.
- Histórico Local de Análises (RNF 12 e RNF 09): As análises realizadas são armazenadas localmente no dispositivo para garantir a disponibilidade, mesmo sem conexão. O acesso a esses dados é controlado pelo próprio aplicativo, minimizando riscos de exposição indevida.

4. Isolamento e Consistência com Docker

A conteinerização do backend com Docker adiciona uma camada de segurança importante:

- Isolamento de Ambiente: Cada serviço (o backend FastAPI, por exemplo) é executado em seu próprio contêiner isolado, o que limita o impacto de uma vulnerabilidade em um serviço sobre os outros.
- Ambientes Consistentes: O Docker garante que o ambiente de execução do backend seja o mesmo em desenvolvimento, teste e produção, reduzindo problemas de configuração e "desvios de ambiente" que podem introduzir vulnerabilidades.
- Atualizações Seguras: Facilita a aplicação de patches e atualizações de segurança nas dependências do backend, pois as imagens Docker podem ser reconstruídas e implantadas de forma controlada.

5. Integração Segura com APIs Externas

O backend atua como um intermediário para as chamadas às APIs externas (Google Safe Browse, VirusTotal, APIs de geolocalização). Essa abordagem é mais segura porque:

- Chaves de API Centralizadas: As chaves de acesso a essas APIs são mantidas no backend (no ambiente seguro da GCP), e não expostas no aplicativo mobile, mitigando o risco de vazamento.
- Controle de Tráfego: O backend pode implementar lógicas de controle de taxa (rate limiting) e validação de requisições antes de encaminhá-las às APIs externas, protegendo contra abusos e ataques de negação de serviço.

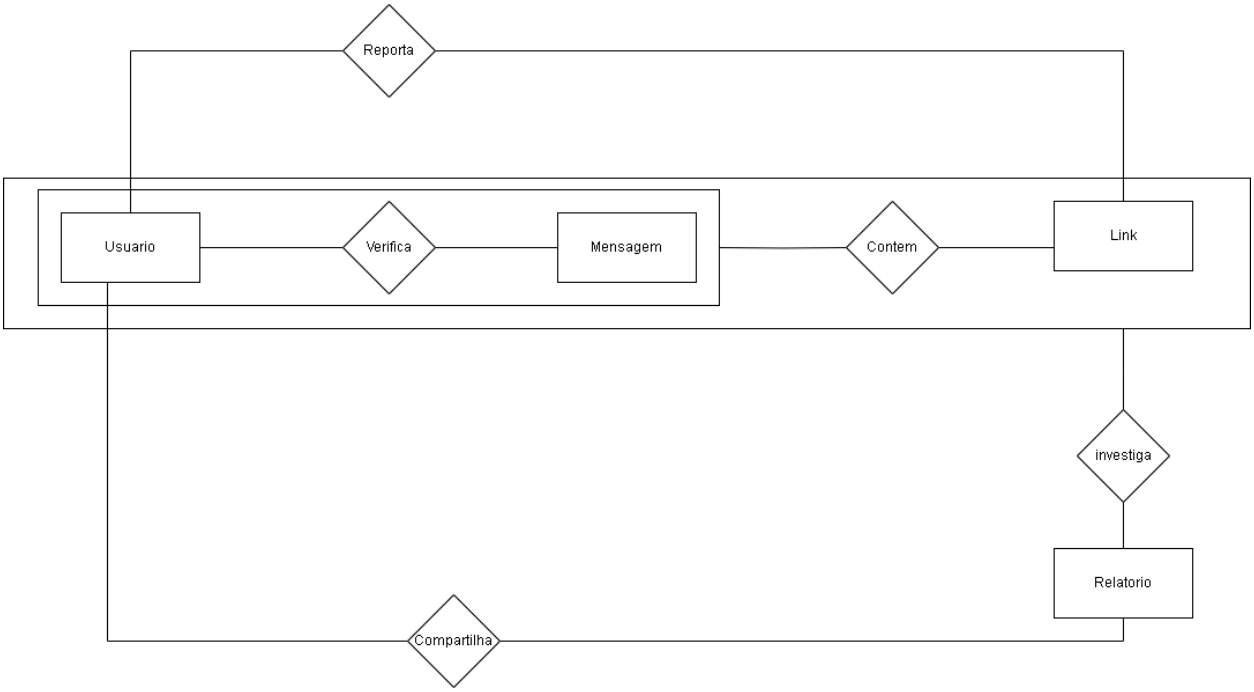
6. Arquitetura Robusta e Manutenibilidade

A adoção de uma Arquitetura Cliente-Servidor Híbrida e do padrão MVVM no cliente, combinada com os princípios SOLID e injeção de dependências, contribui indiretamente para a segurança. Um código bem-estruturado e modular (RNF 07) é mais fácil de auditar, manter e expandir, o que facilita a identificação e correção de potenciais falhas de segurança antes que se tornem vulnerabilidades.

A abordagem do Phishing Tracker à segurança é multifacetada, combinando boas práticas de desenvolvimento, o uso de tecnologias seguras e a conscientização do usuário para oferecer uma proteção eficaz contra as ameaças de phishing.

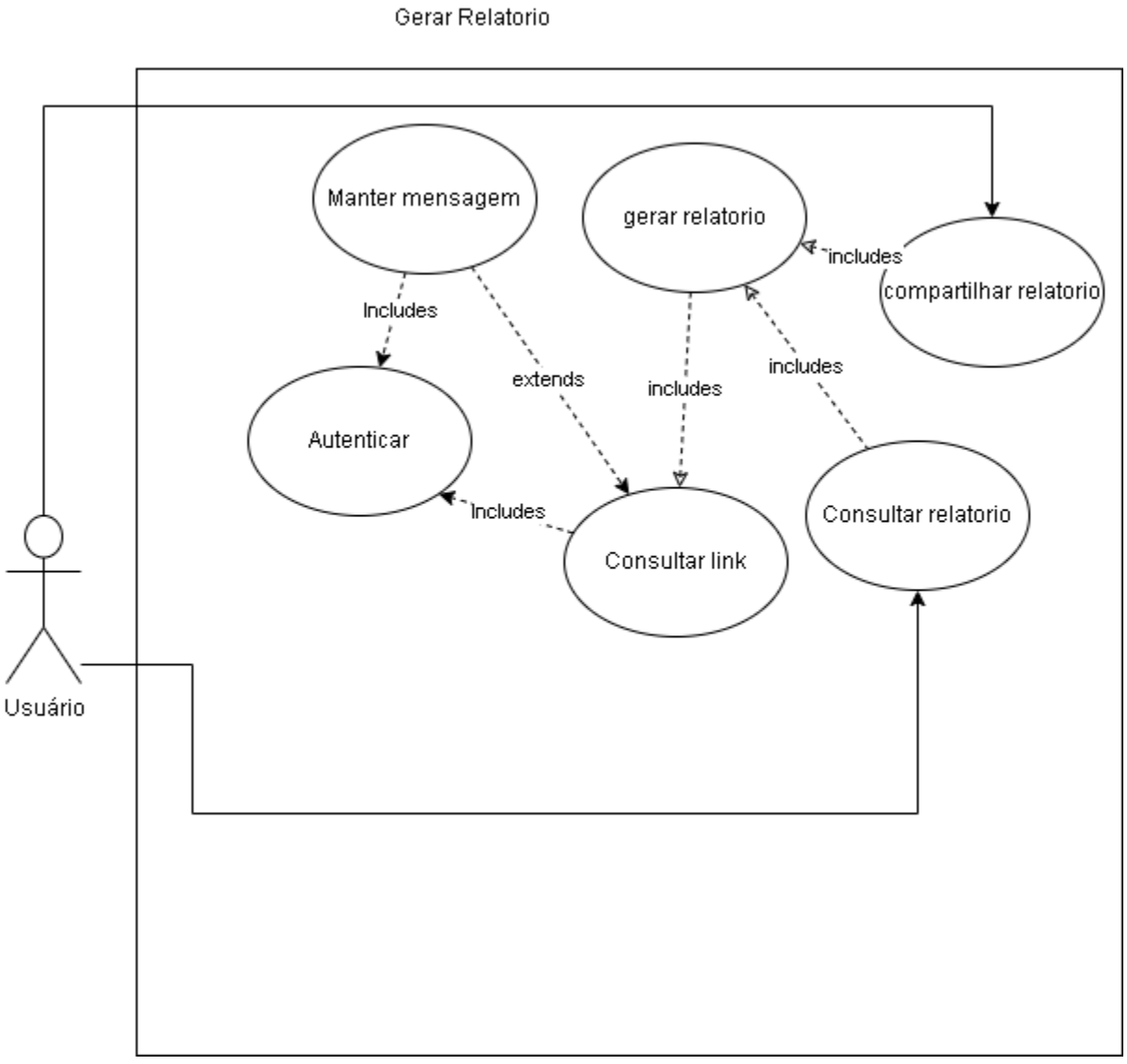
10. Modelo de Entidade e relacionamento

10.1. Diagrama 1

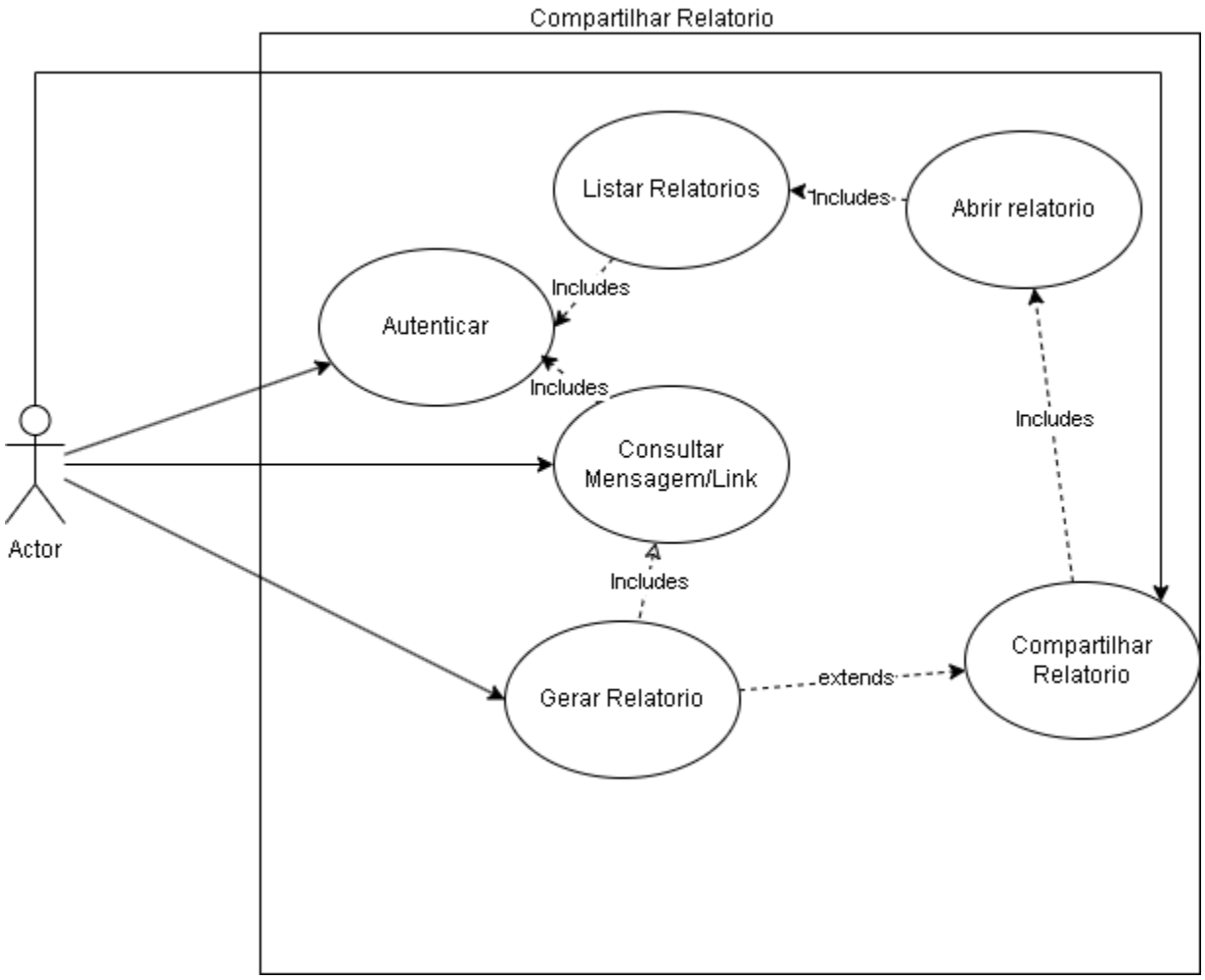


11. Caso de Uso

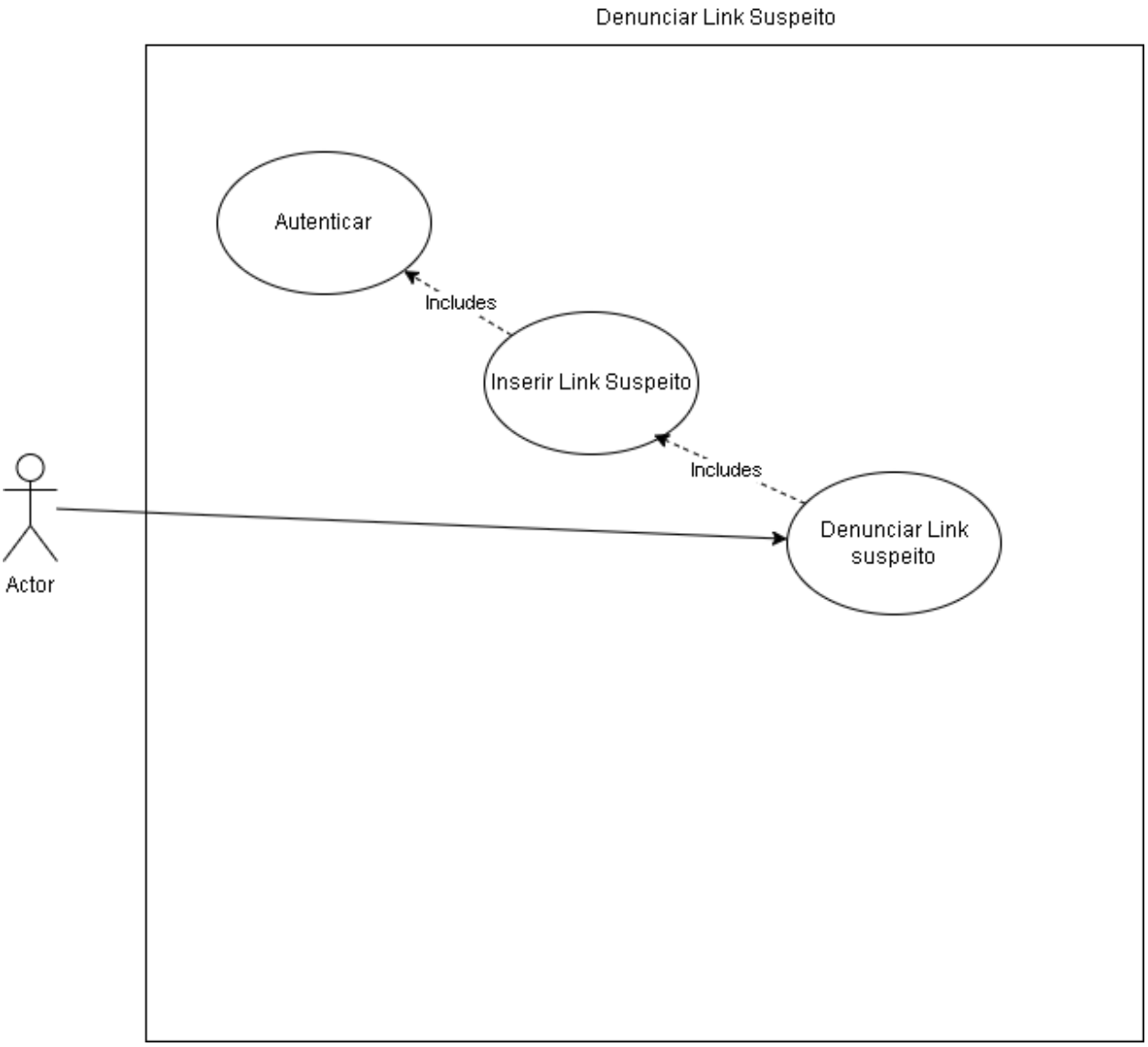
11.1. Diagrama 1



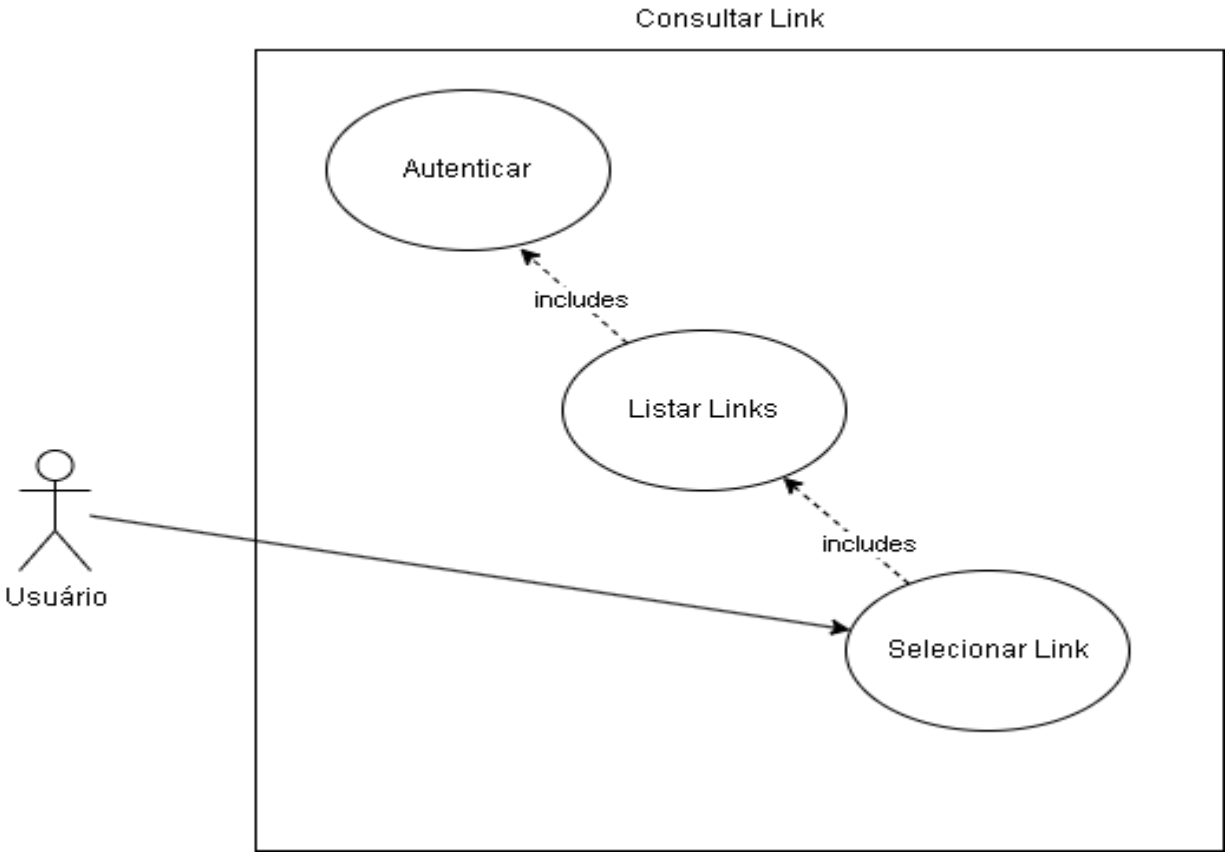
11.2. Diagrama 2



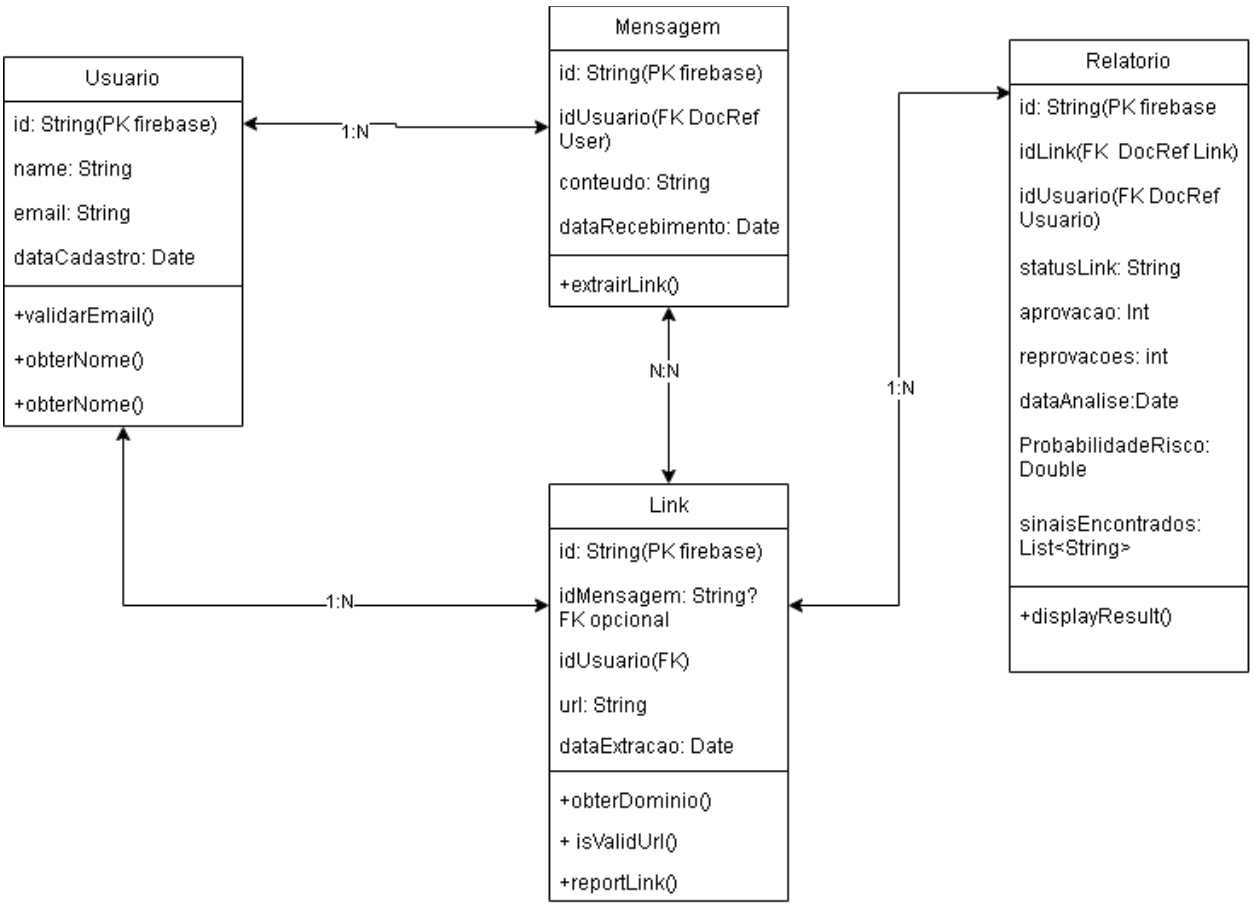
11.3. Diagrama 3



11.4. Diagrama 4

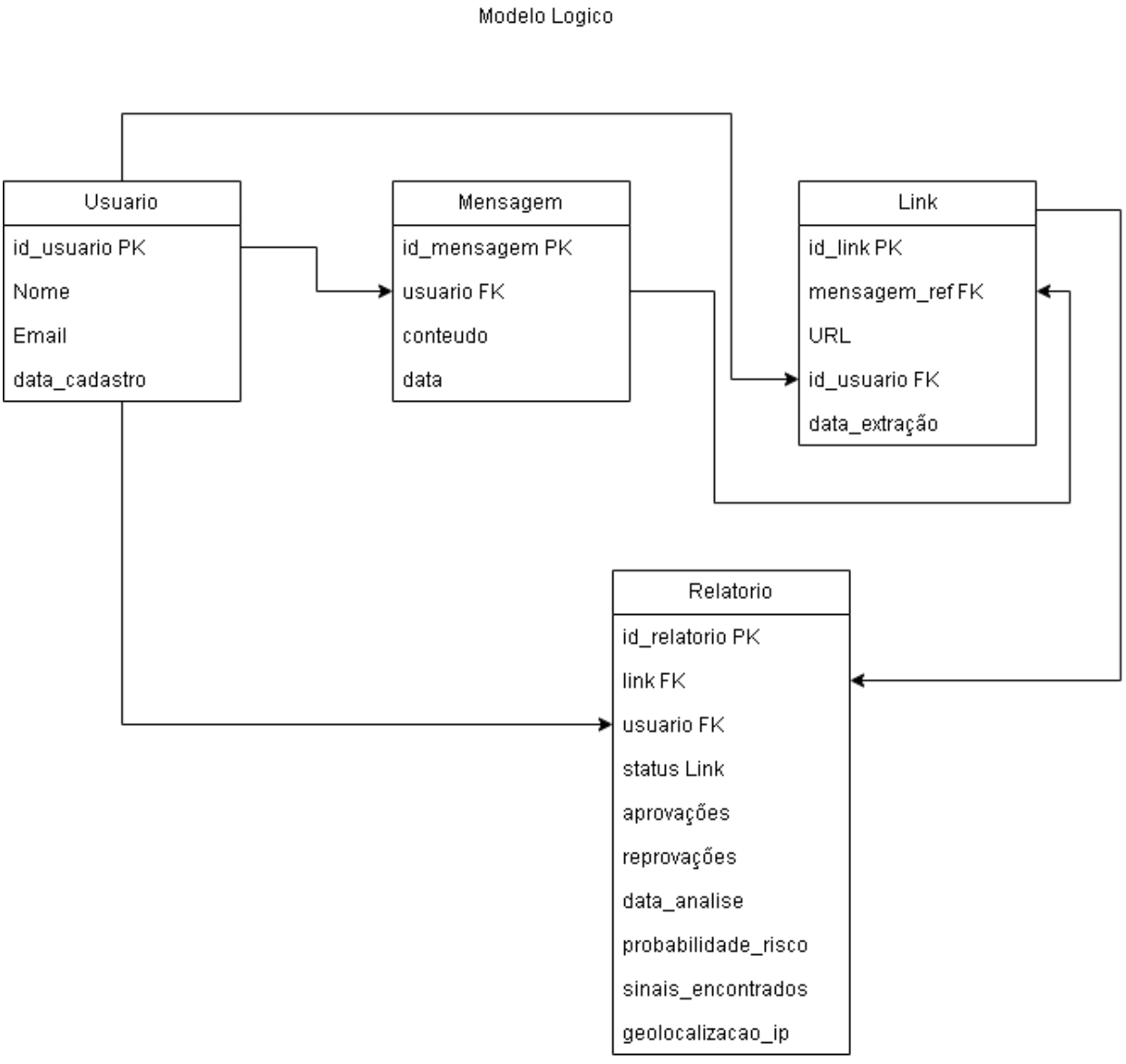


12. Diagrama de classes

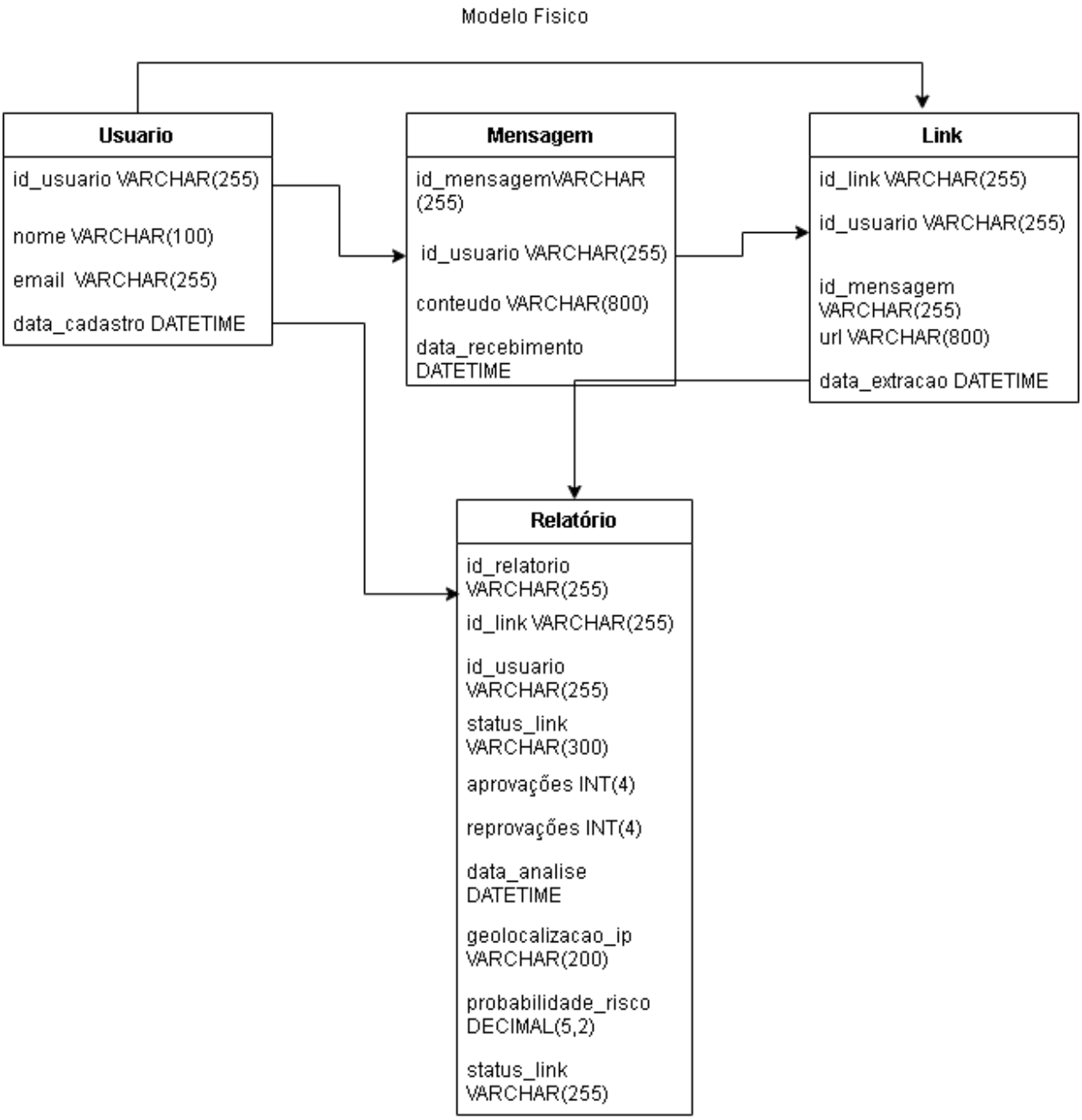


13. Modelagem de Dados

13.1. Modelo lógico

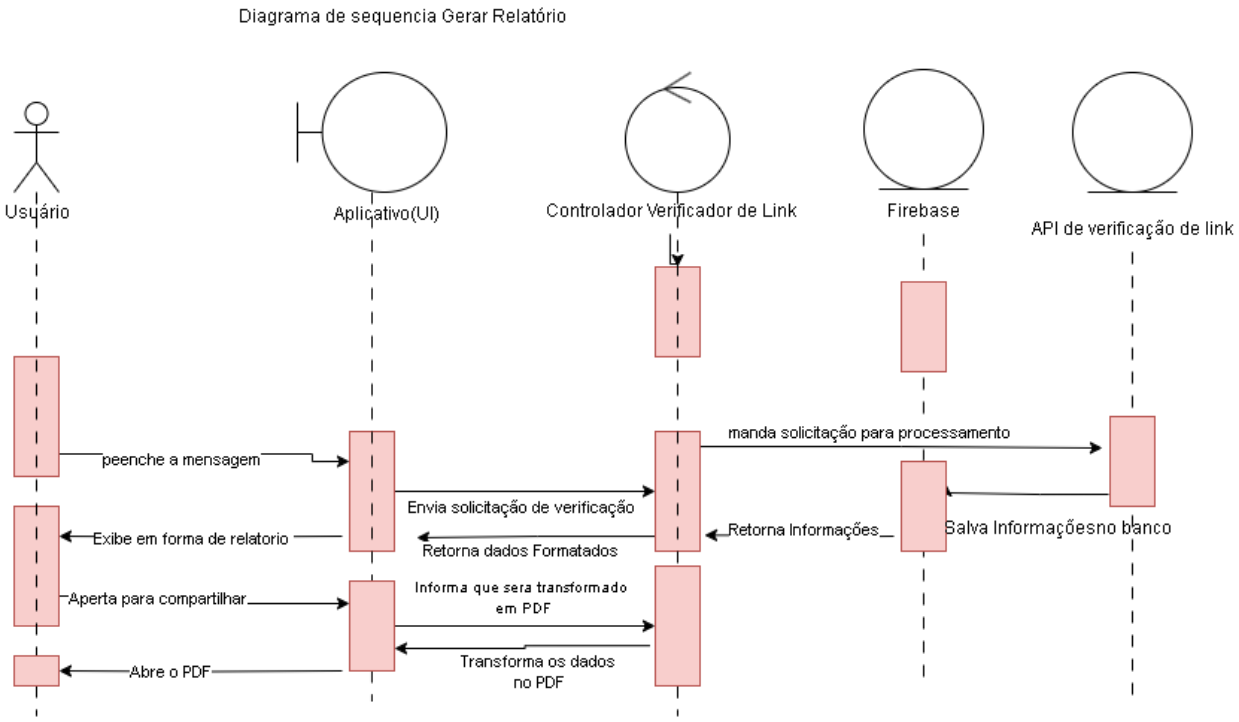


13.2. Modelo físico

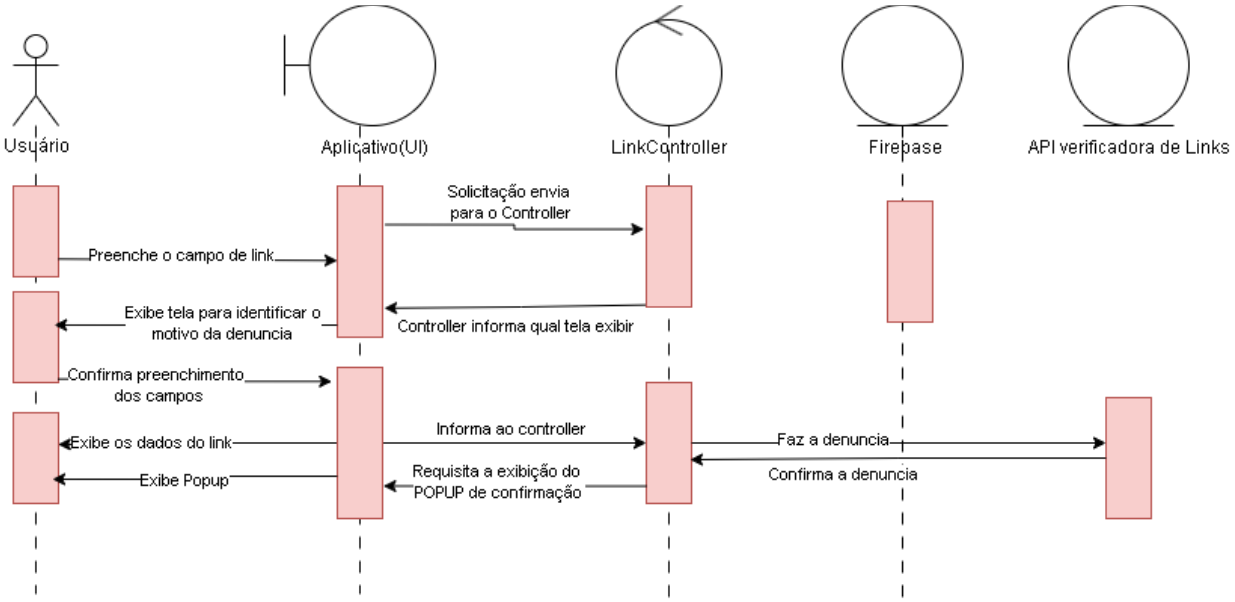


14. Diagramas de Sequência

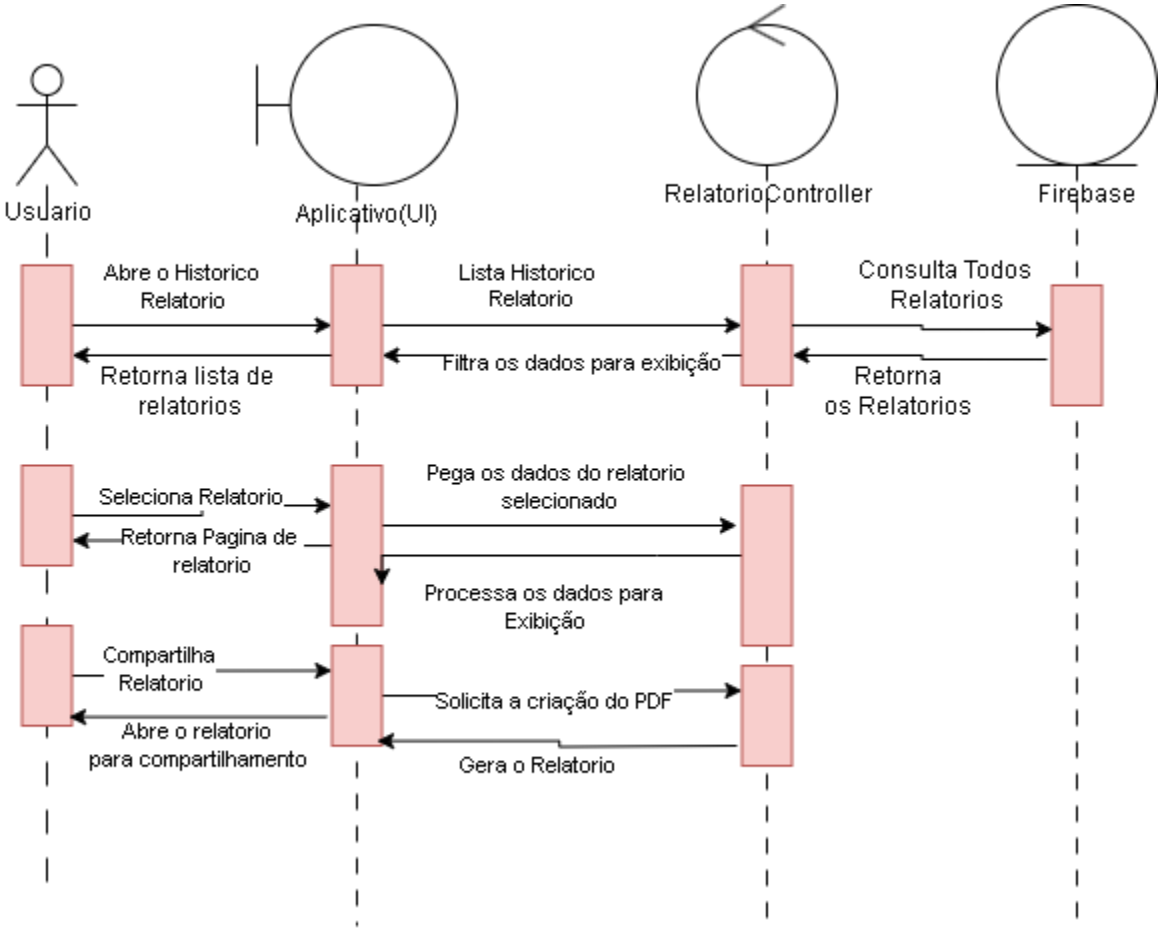
14.1. Gerar relatório



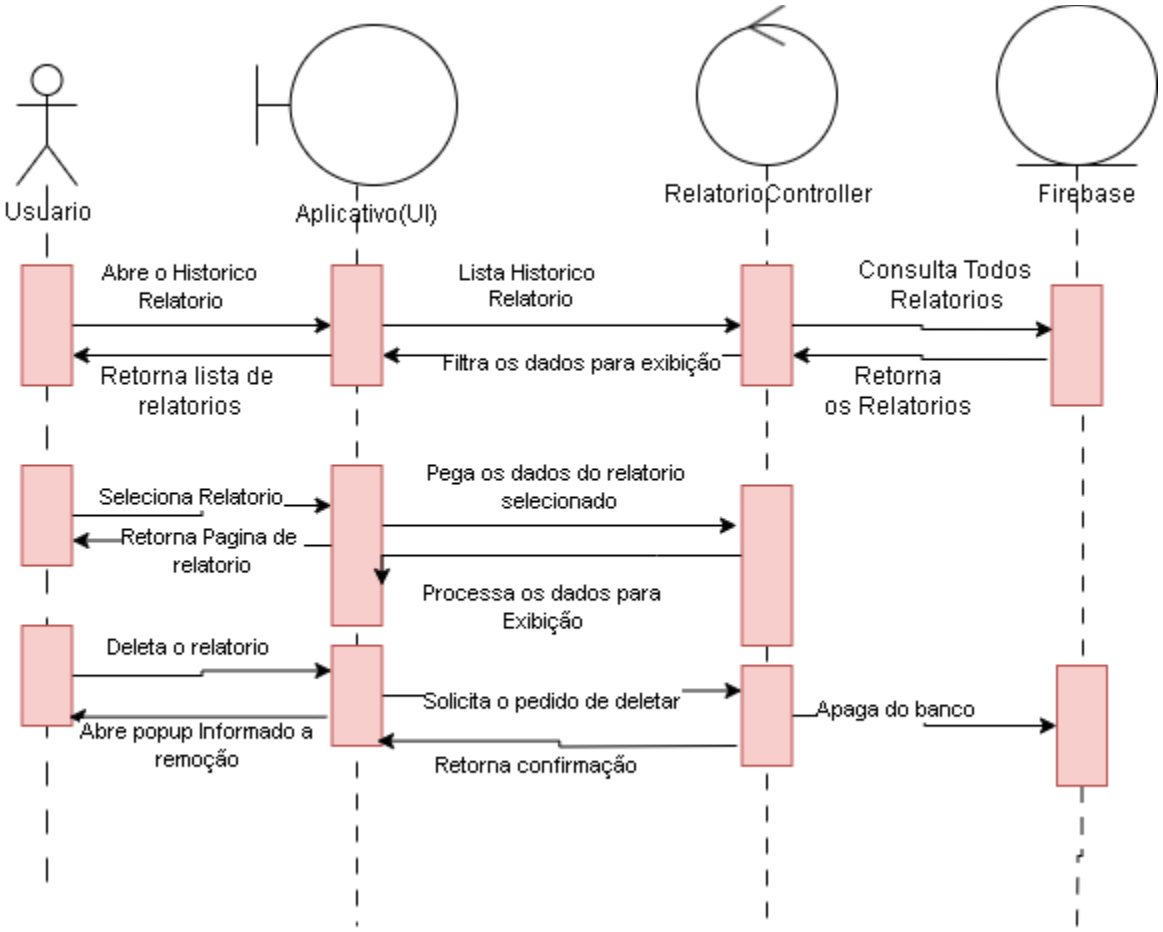
14.2. Denunciar Link



14.3. Compartilhar Relatório



14.4. Consultar Link



15. A Aplicação

O "Phishing Tracker" é um sistema de segurança digital meticulosamente projetado, que opera através de uma colaboração sinérgica entre sua API de Backend robusta e um Aplicativo Mobile intuitivo. Essa arquitetura dual, com responsabilidades bem definidas, não só otimiza o desempenho e a escalabilidade, mas também garante uma camada de proteção eficiente e acessível contra as crescentes ameaças de phishing no ambiente digital.

15.1. A API de Backend: O Centro de Inteligência e Análise

A API de Backend é o coração do sistema "Phishing Tracker", funcionando como um centro de inteligência que orquestra a coleta, análise e consolidação de dados de segurança provenientes de diversas fontes externas. Sua principal função é processar requisições complexas de verificação, garantindo que o aplicativo mobile permaneça leve e responsivo, concentrando-se na experiência do usuário.

Para atingir sua capacidade de detecção multifacetada, a API se integra com quatro serviços cruciais, cada um adicionando uma camada única de inteligência à análise:

Google Safe Browse: Agindo como a primeira e fundamental linha de defesa, este serviço verifica URLs em tempo real contra as extensas e constantemente atualizadas listas do Google. Essas listas incluem sites conhecidos por hospedar malware, software indesejado ou por serem plataformas de phishing. A integração com o Safe Browse permite que o sistema identifique rapidamente ameaças já catalogadas, protegendo o usuário de acessar conteúdos comprovadamente maliciosos.

VirusTotal API: Esta integração oferece uma análise de segurança aprofundada e colaborativa. Ao receber um link, a API o envia ao VirusTotal, que o submete a dezenas de motores antivírus e scanners de reputação. O retorno é um relatório detalhado que inclui a avaliação de múltiplas entidades de segurança, informações sobre o endereço IP do servidor e detecções de ferramentas de proteção. Um recurso distintivo do VirusTotal é sua abordagem comunitária, onde usuários e pesquisadores podem denunciar URLs suspeitas. Essa contribuição coletiva alimenta a base de dados global, acelerando a identificação e a mitigação de novas ameaças para toda a comunidade de segurança.

IPinfo (ou Serviço de Geolocalização Similar): Para adicionar uma dimensão geográfica vital à análise, a API utiliza serviços de geolocalização de IP. Ao determinar o endereço IP do servidor que hospeda o link, é possível identificar sua localização geográfica. Este dado é crucial para detectar inconsistências que podem indicar fraude, como um link que se apresenta como sendo de uma instituição financeira ou governamental brasileira, mas está hospedado em um servidor localizado em um país

com histórico de atividades cibernéticas ilícitas. Tal discrepância serve como um forte indicador de risco, alertando o usuário para uma possível tentativa de golpe.

Gemini AI (Google Gemini): A inteligência artificial de ponta do Google é fundamental para a análise contextual das mensagens e para a geração de relatórios compreensíveis. O Gemini processa o conteúdo textual da mensagem enviada pelo usuário, buscando padrões típicos de engenharia social, como o uso de linguagem alarmista, erros gramaticais incomuns em comunicações oficiais, solicitações de urgência irrealistas e uma abordagem genérica que não personaliza o destinatário. Além de auxiliar na validação e possível identificação de links ofuscados dentro do texto, o Gemini é o responsável por sintetizar todas as informações coletadas dos demais serviços em um relatório final que é claro, conciso e altamente contextualizado. Esse relatório é entregue em uma linguagem natural, explicando ao usuário os riscos detectados e os elementos que o tornam suspeito, embora sempre com a ressalva de que, como qualquer IA, pode haver casos de "alucinações" que exigem a atenção do usuário.

A infraestrutura da API é construída sobre a Google Cloud Platform (GCP), especificamente utilizando o Cloud Run. Essa escolha estratégica confere múltiplos benefícios:

Escalabilidade e Estabilidade Inerentes: O Cloud Run é um serviço serverless que se autoescala dinamicamente, ajustando automaticamente os recursos conforme a demanda. Isso garante que a API consiga lidar com picos súbitos de requisições sem comprometer o desempenho ou a disponibilidade, proporcionando uma operação estável e confiável.

Eficiência de Custos Otimizada: Operando sob um modelo de pagamento por uso, o Cloud Run maximiza a eficiência financeira. Os containers da API permanecem em um estado "inativo" quando não há requisições, sendo ativados rapidamente apenas quando uma nova consulta é recebida. Isso elimina custos associados à manutenção de infraestrutura ociosa, tornando a solução economicamente viável para diferentes volumes de uso.

Segurança Reforçada: A API implementa um rigoroso controle de acesso para garantir que apenas usuários legítimos e autenticados possam interagir com o sistema. Através da integração com o Firebase Admin SDK, a API verifica a validade de JSON Web Tokens (JWTs) provenientes do aplicativo mobile. Essa validação assegura que cada requisição seja autenticada, adicionando uma camada crucial de segurança e integridade ao fluxo de dados.

15.2. O Aplicativo Mobile (Cliente): A Interface do Usuário e Ferramenta de Conscientização

O Aplicativo Mobile, desenvolvido em Kotlin com Jetpack Compose para o ecossistema Android, é a porta de entrada para a poderosa capacidade de detecção do Phishing Tracker. Ele foi concebido para ser a interface intuitiva e amigável que conecta o usuário diretamente à inteligência analítica fornecida pela API de backend.

Suas funcionalidades principais são cuidadosamente desenhadas para proporcionar uma experiência fluida e segura:

Interação Simplificada e Acessível: A interface do aplicativo é projetada com foco na usabilidade intuitiva. Ela permite que os usuários insiram ou compartilhem mensagens e links suspeitos de forma rápida e descomplicada. Essa facilidade de uso é fundamental para encorajar a adoção por um público amplo, independentemente do seu conhecimento técnico em cibersegurança.

Comunicação Segura com a API: O aplicativo é o responsável por coletar os dados fornecidos pelo usuário e, após a autenticação segura via Firebase Authentication, enviá-los à API de backend para processamento. Essa comunicação é criptografada e autenticada, protegendo a privacidade das informações transmitidas.

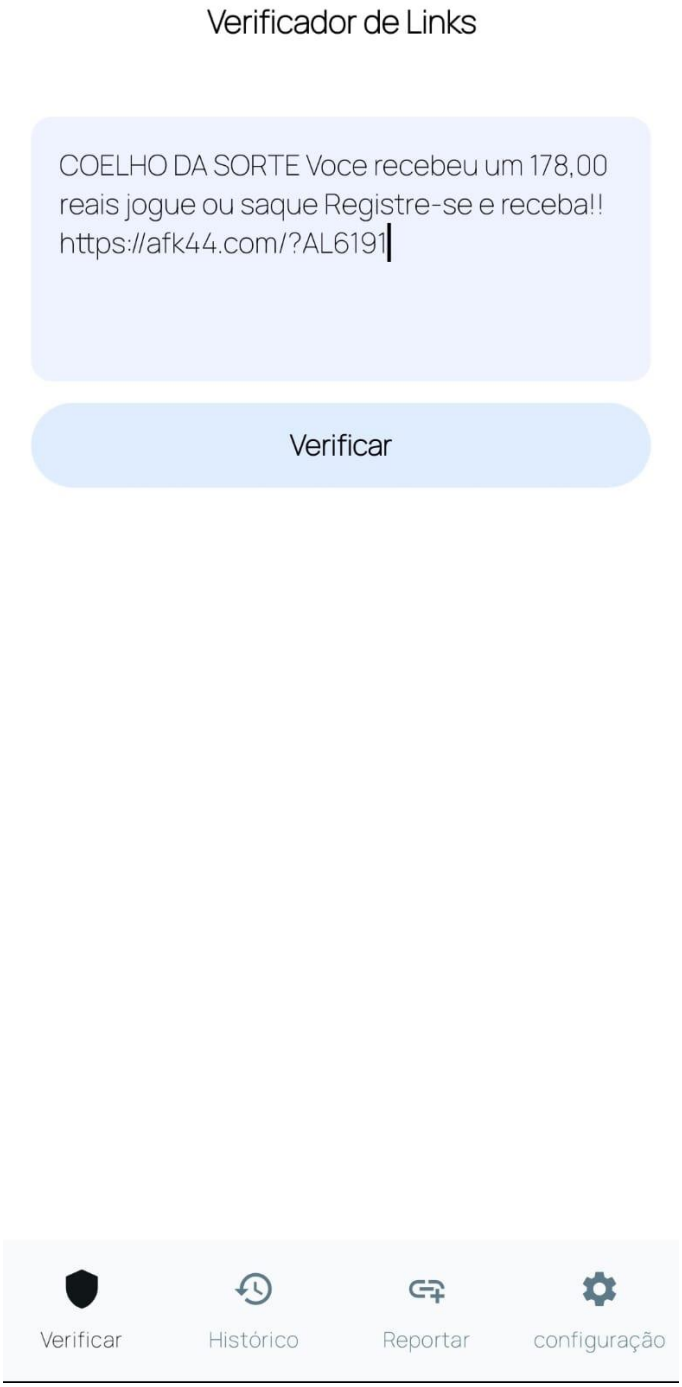
Geração e Exibição Dinâmica de Relatórios: Uma vez que a API retorna o resultado da análise, o aplicativo assume o papel de intérprete. Graças à flexibilidade proporcionada pela análise contextual do Gemini AI, o aplicativo é capaz de montar layouts de relatório que são específicos e personalizados para cada mensagem ou URL analisada. Isso significa que a apresentação visual do risco é adaptada aos elementos suspeitos detectados, tornando as informações mais claras, relevantes e compreensíveis para o usuário. Essa dinamicidade evita relatórios genéricos e aumenta a eficácia da comunicação do risco.

Compartilhamento Facilitado com Geração de PDF: Uma funcionalidade de grande valor agregado é a capacidade de formatar e gerar um arquivo PDF contendo o relatório completo da análise diretamente no aplicativo. Essa funcionalidade é essencial para a disseminação da informação e conscientização. Permite que o usuário compartilhe facilmente os resultados da verificação com amigos, familiares ou colegas, mesmo que eles não tenham o aplicativo instalado. Isso não só amplia o alcance da ferramenta, mas também promove uma cultura de cibersegurança coletiva, onde o conhecimento sobre ameaças é compartilhado de forma prática e eficaz.

Em resumo, o aplicativo mobile é mais do que uma ferramenta de detecção; é um agente de capacitação e conscientização. Ele coloca o poder da análise de segurança avançada na palma da mão do usuário, transformando a complexa tarefa de identificar phishing em um processo simples e acessível, contribuindo diretamente para a criação de um ambiente digital mais seguro

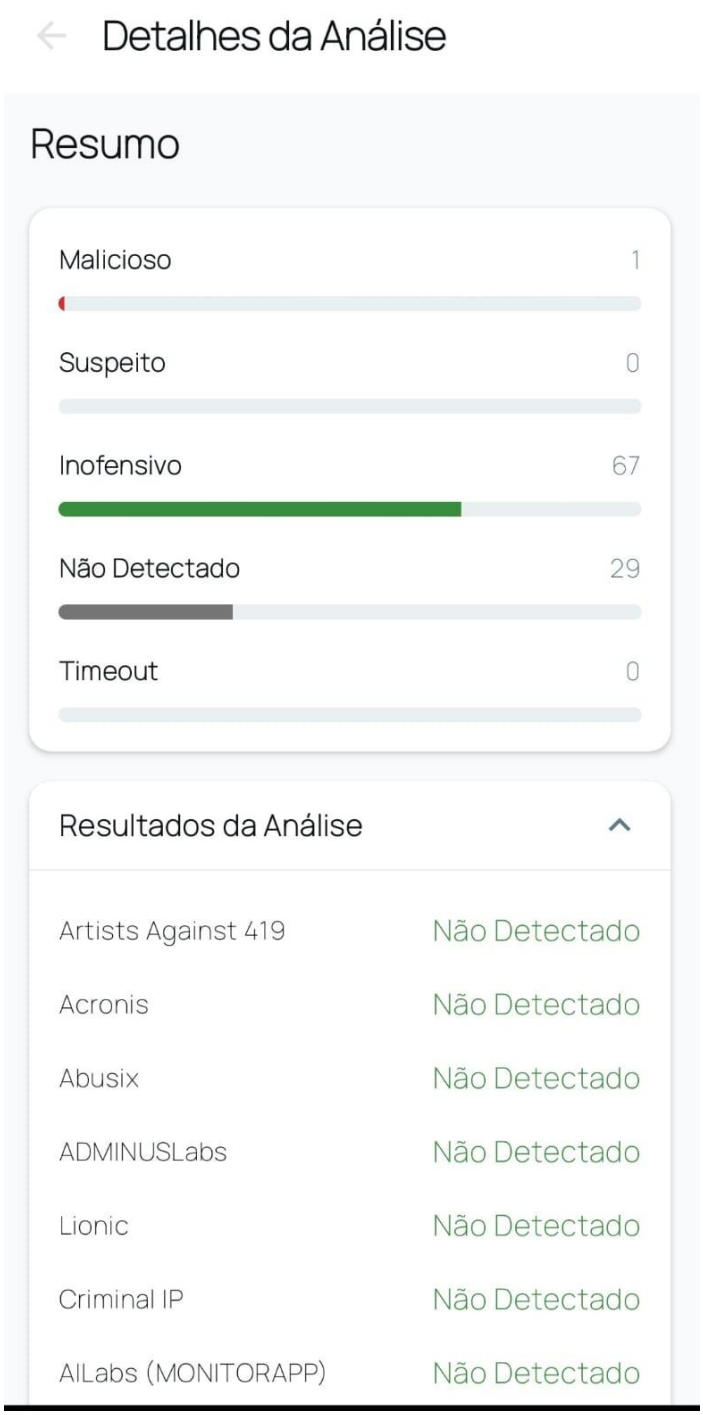
16. Processo

16.1. Analisar Mensagem passo 1



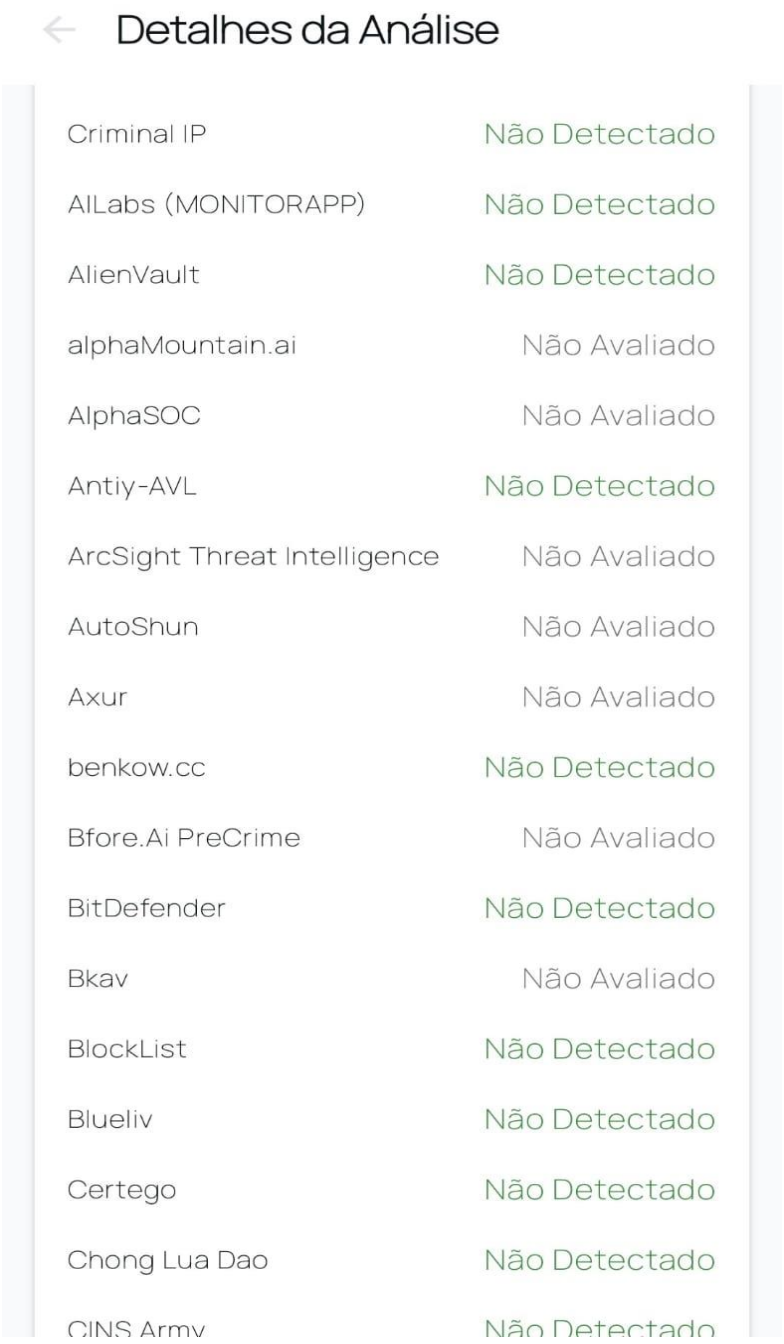
Adicionei um link suspeito para analisar

16.2. Passo 2



Apos um tempo esse foi meu resultado, é possivel ver resumidamente os que definiram como malicioso suspeito e assim por diante

16.3. Passo 3



1.3 Lista dos que testaram e viram tentaram identificar alguma coisa suspeita no site

16.4. Passo 4

1.4 Análise estrutural da mensagem, onde a IA consegue analisar o conteudo em si

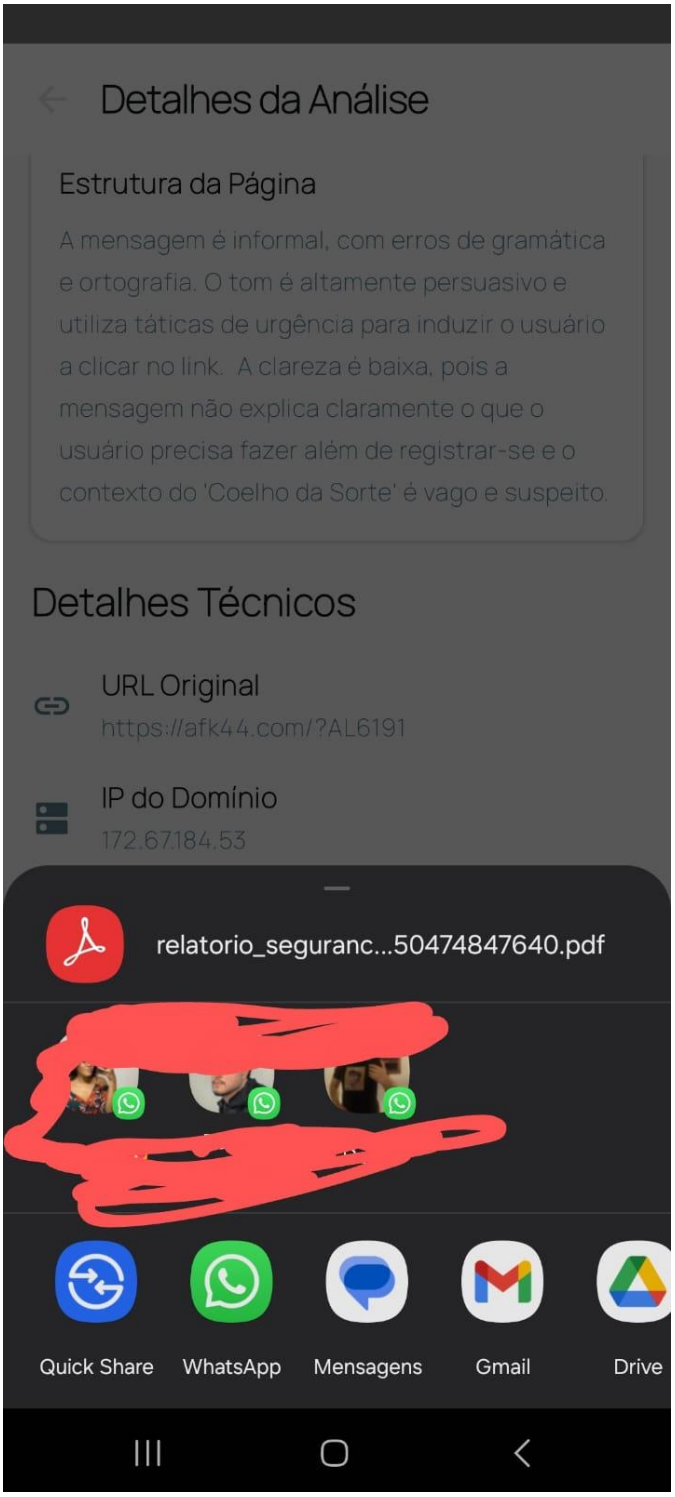
16.5. Passo 5

1.5 Detalhes identificados do link: dominio, provedor IP, geolocalização do servidor, e data

16.6. Passo 6

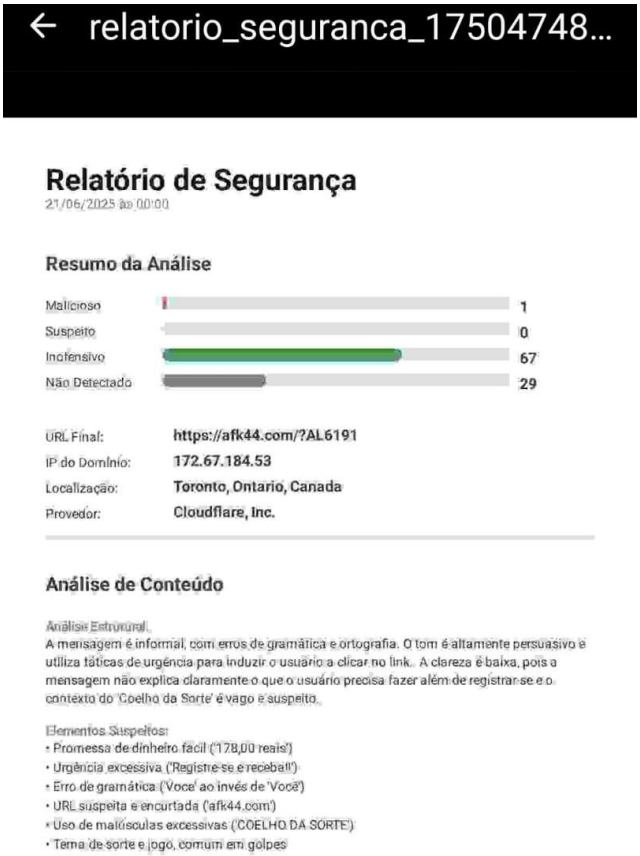
1.6 Botões que geram o relatório e que atualizam(reenviam para analise)

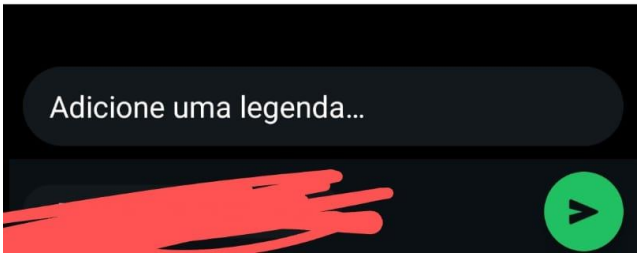
16.7.Passo 7



1.7 Para qual aplicativo irá compartilhar

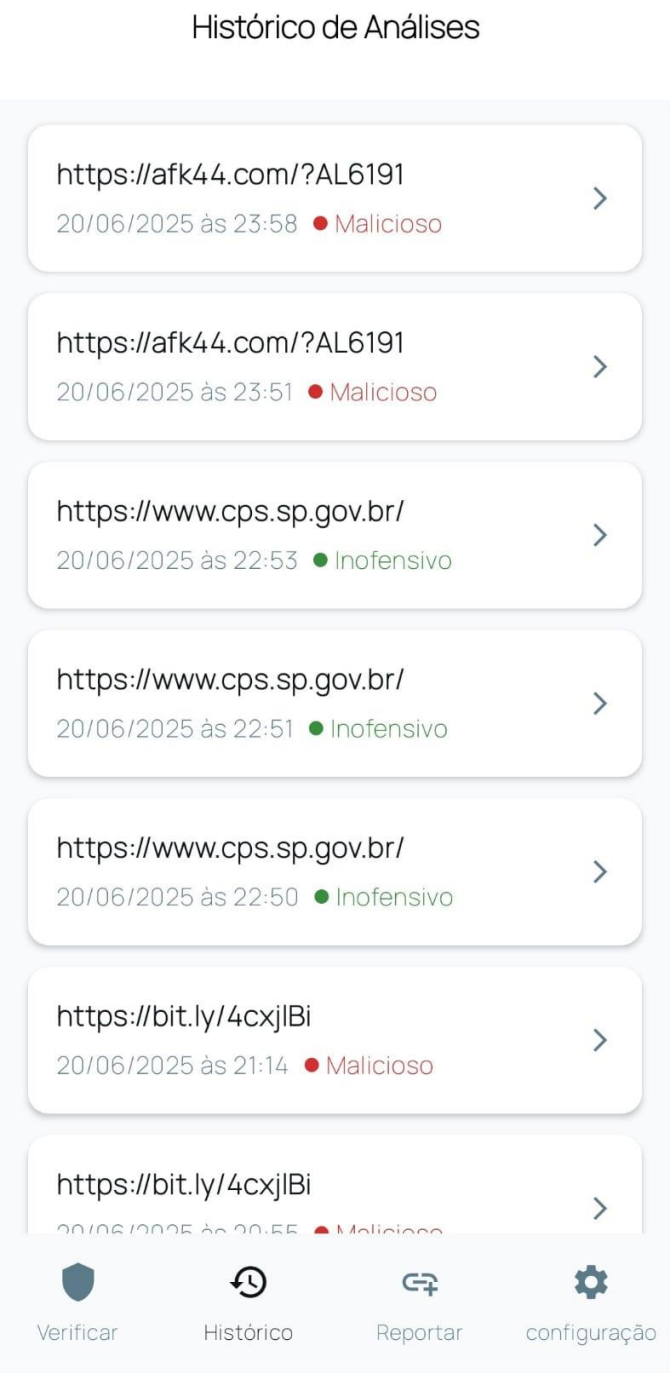
16.8. Passo 8



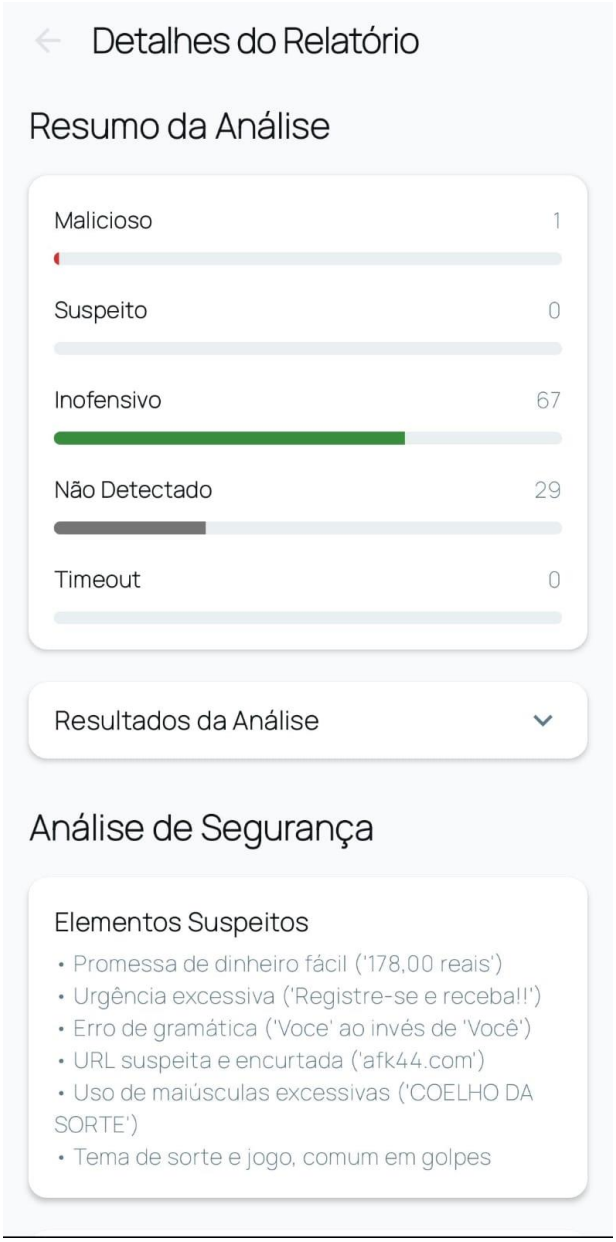


1.8 Envio do relatório para o destinatário desejado em PDF

16.9. Histórico de mensagens/links

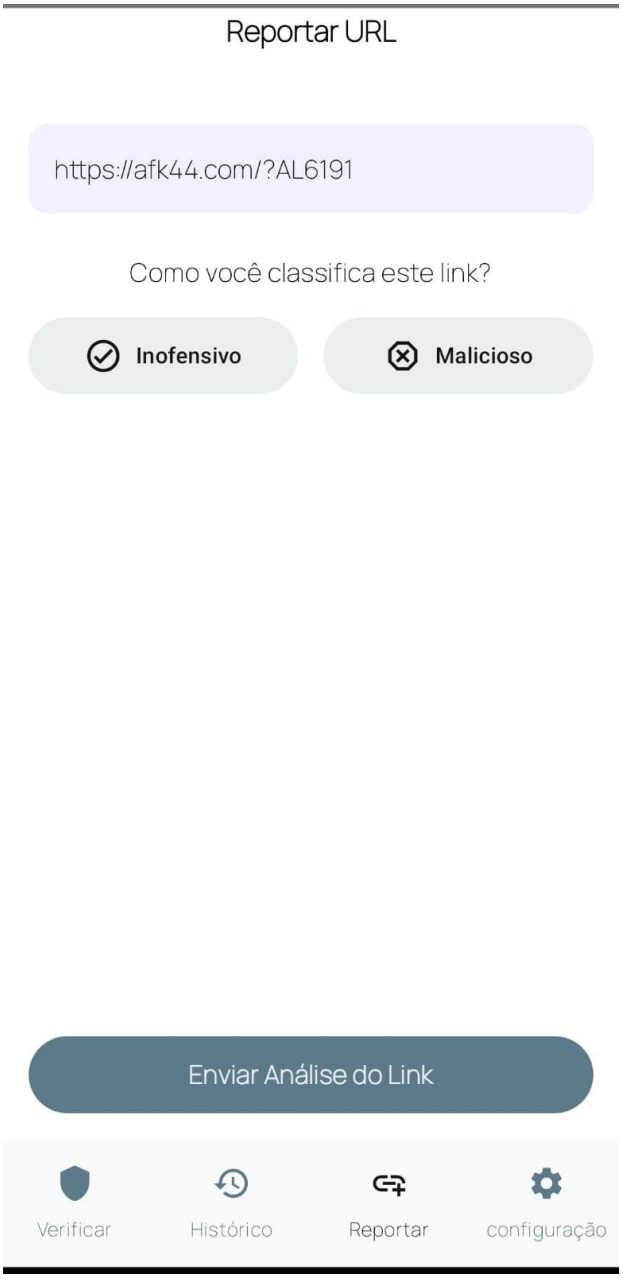


2.1 Página de histórico de análise

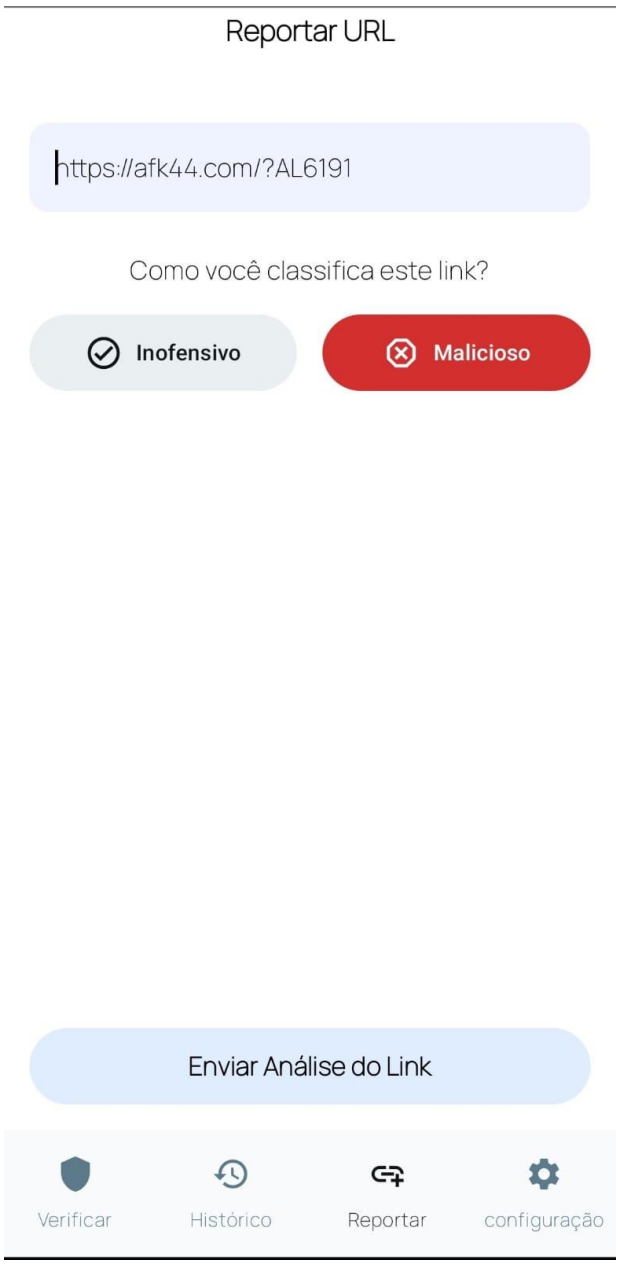

Detalhes do Relatório
Resumo da Análise
Malicioso 1
Suspeito 0
Inofensivo 67
Não Detectado 29
Timeout 0
Resultados da Análise
Análise de Segurança
Elementos Suspeitos
• Promessa de dinheiro fácil ('178,00 reais')
• Urgência excessiva ('Registre-se e receba!!')
• Erro de gramática ('Voce' ao invés de 'Você')
• URL suspeita e encurtada ('afk44.com')
• Uso de maiúsculas excessivas ('COELHO DA SORTE')
• Tema de sorte e jogo, comum em golpes

2.2, 2.3Mesma tela da análise contendo o grafico, resultados da análise, elementos suspeitos, análise estrutural e os dados sobre o servidor e a opção de compartilhar o pdf

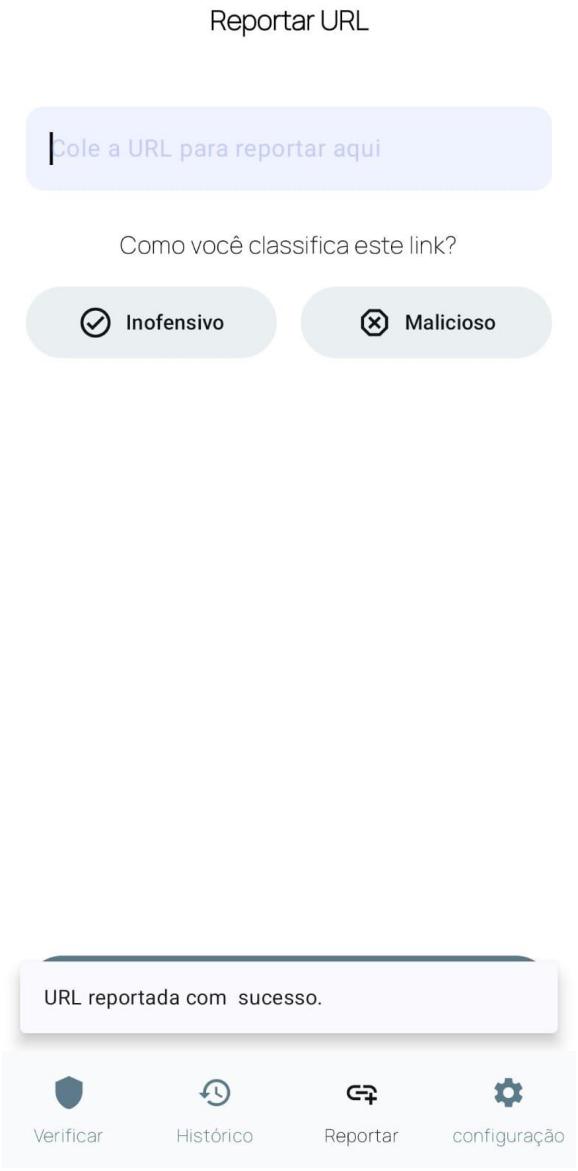
16.10. Reportar URL



3.1 Tela para reportar link, basta colar o link avaliar se é inofensivo ou malicioso

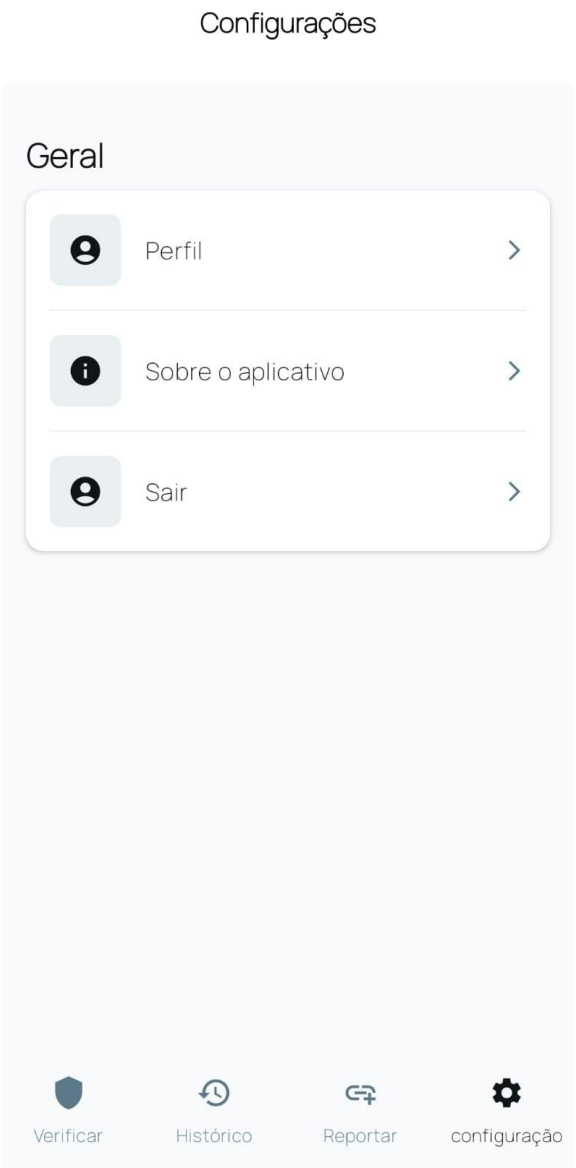


3.2 Avaliei o URL como malicioso



3.3 Conclui a avaliação

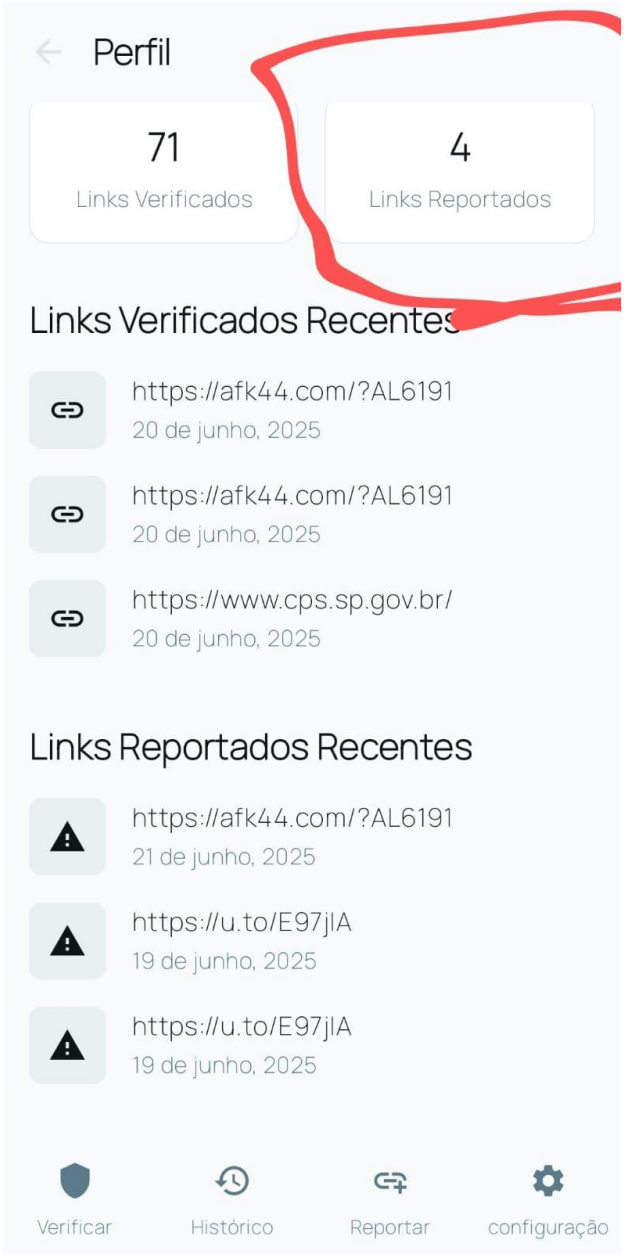
16.11. Configuração e Perfil



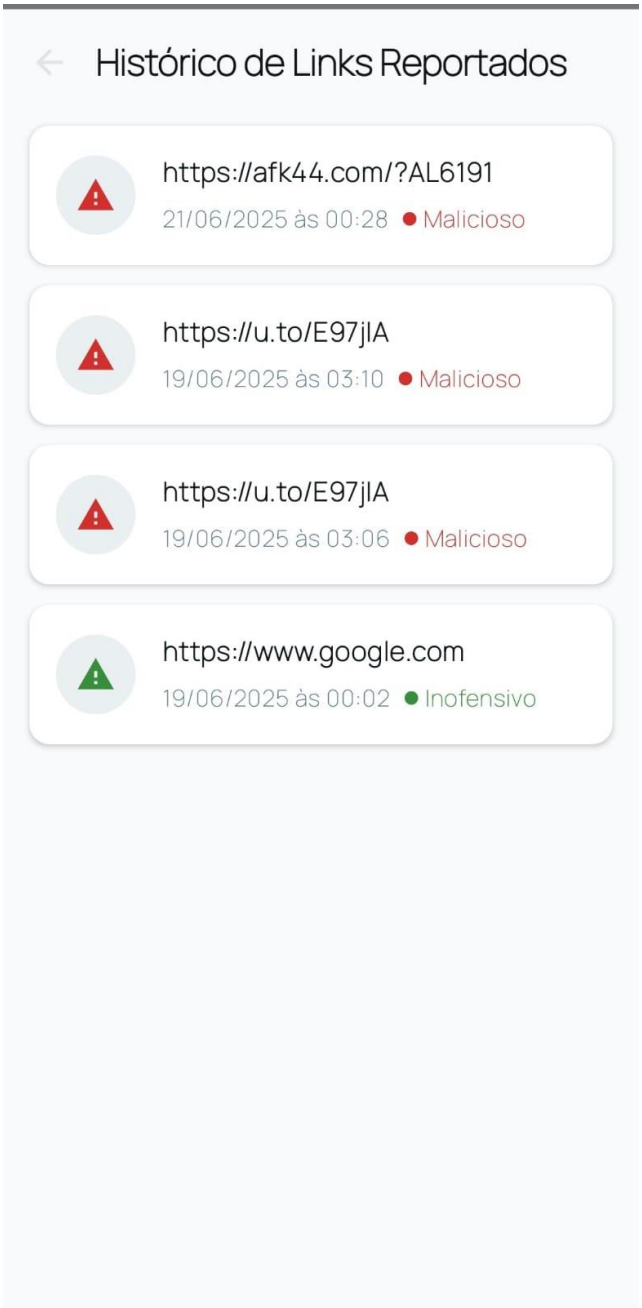
4.1 Configurações gerais

4.2 Tela com os detalhes e atividades da conta

4.3 Exibição das ultimas 3 atividades do usuario em cada tipo(relatorios e links reportados)

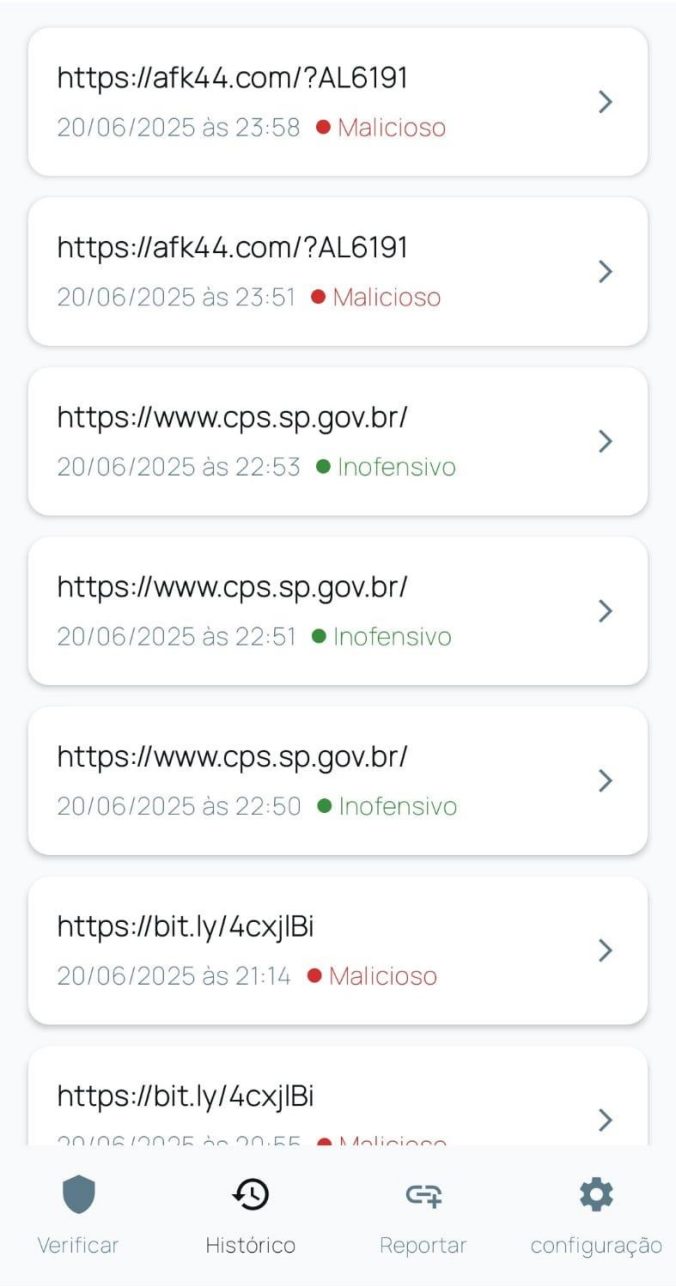


4.4 Para exibir o histórico de links reportados basta apertar o botão com número total total de links reportados



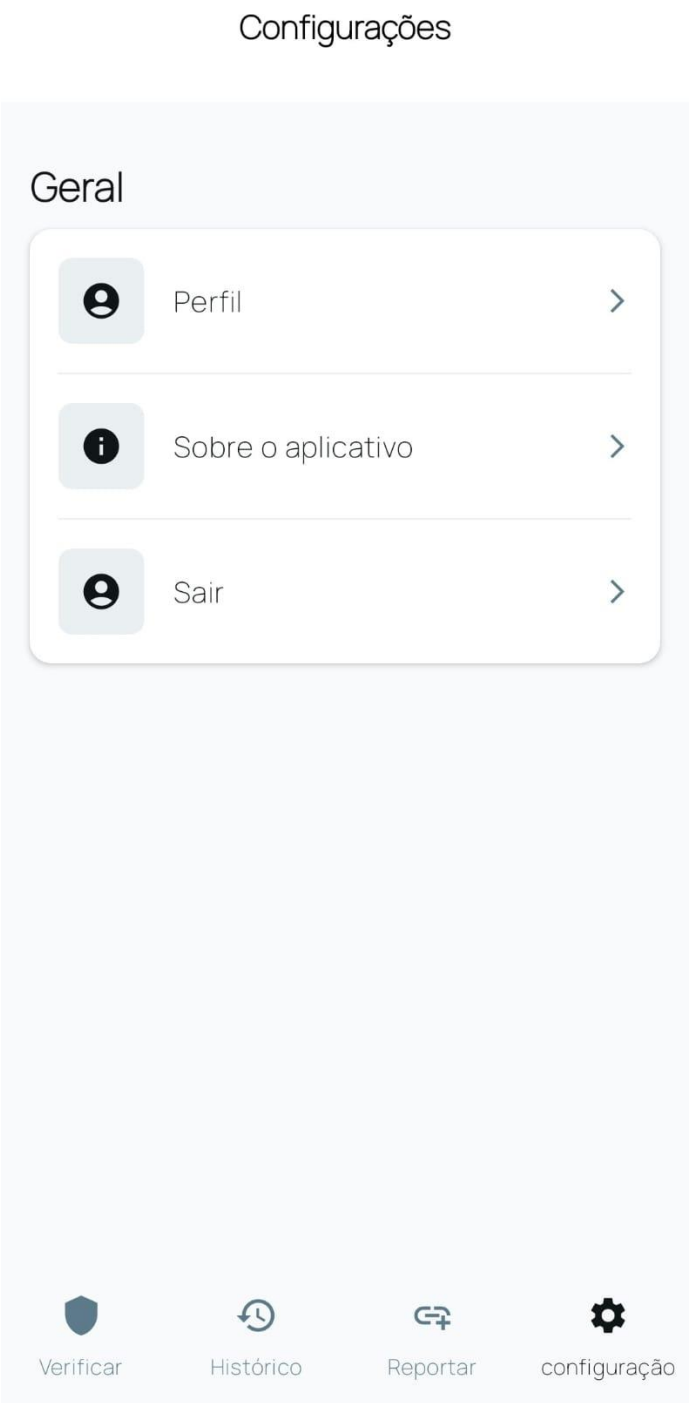
4.5 Histórico dos links reportados com o status de malicioso ou inofensivo

4.6 Ao clicar no número total de links verificados Abre a pagina de histórico



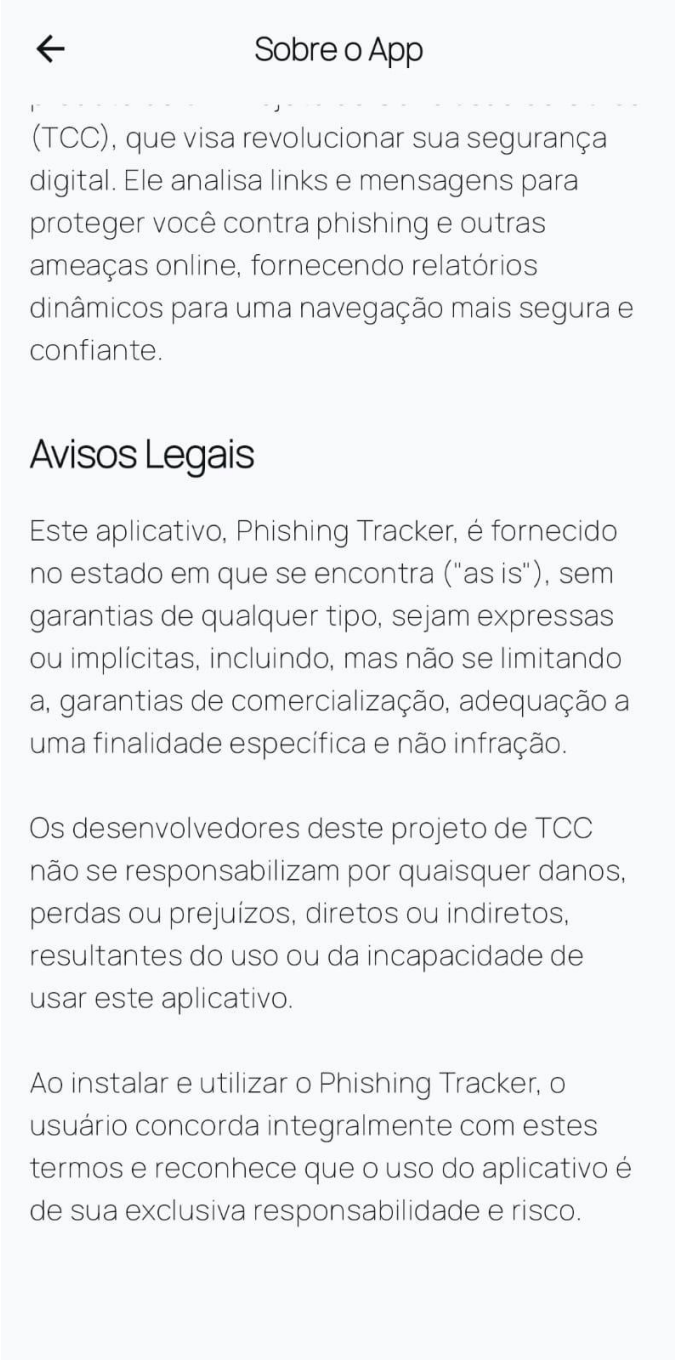
4.7 Tela de historico aberta pela pagina de perfil

16.12. Sobre o Aplicativo



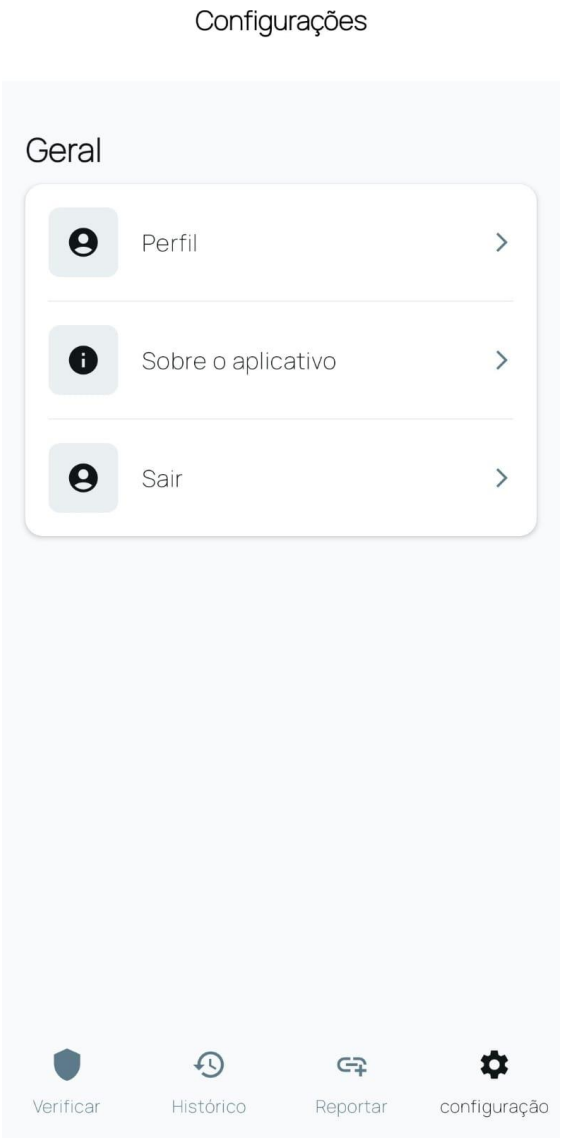
6.1 Ao Clicar em configuração/sobre o aplicativo vai abrir a pagina com a descrição contendo numero de versão e informando que é para um projeto de tcc

6.2 Detalhes “Sobre” o aplicativo, numero de versão e o desenvolvedor



6.3 Detalhes legais sobre o app isentando a responsabilidade sobre qualquer uso

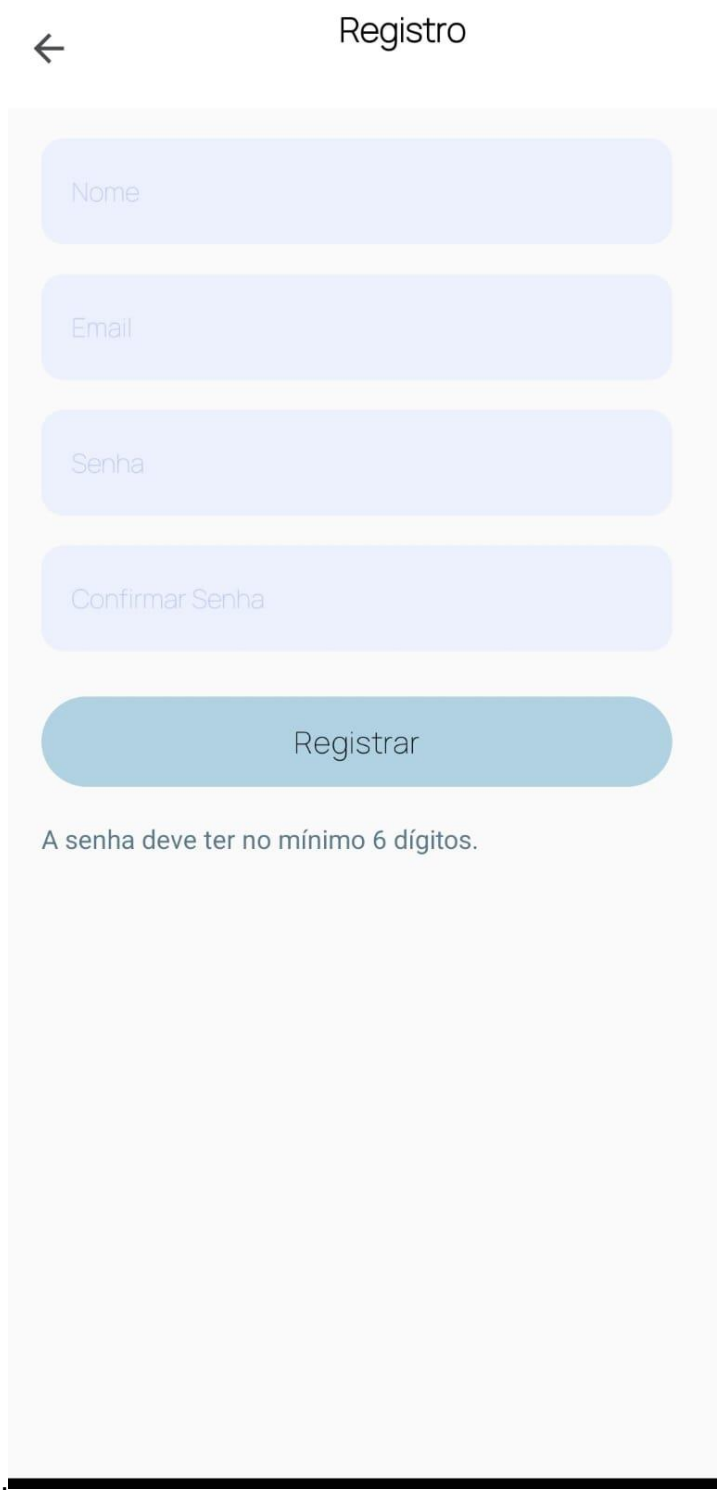
16.13. Logout



7.1 Ao apertar o botão de sair faz o logout da sessão

16.14. Tela principal(quando não está logado) Login

8.1 Tela principal com login e texto que leva a tela de registr



8.2 Tela de registro simplifica

17. Conclusão

17.1. Pontos que poderiam melhorar:

O desenvolvimento do Phishing Tracker representou um avanço significativo na criação de uma ferramenta de segurança digital acessível e eficaz. Contudo, como em todo projeto, identifiquei pontos-chave que, se aprimorados, poderiam elevar ainda mais o potencial e a robustez da aplicação.
Os principais aspectos que poderiam ter sido melhorados são:

- Ampliação da Base de Dados de Segurança: A integração com outros serviços públicos e privados de segurança digital, além dos já utilizados Google Safe Browse e VirusTotal, seria crucial. Isso permitiria ao Phishing Tracker consultar uma gama ainda mais vasta de listas de bloqueio e bases de dados de inteligência de ameaças, aumentando a precisão na detecção de phishing e outros ataques de nível similar. Adicionar mais fontes diversifica a análise e reduz as chances de falsos negativos.

- Implementação Abrangente de Testes: A falta de conhecimento técnico aprofundado na área de testes resultou em uma lacuna importante no ciclo de desenvolvimento. A introdução de testes unitários, de integração e de sistema seria fundamental para garantir a estabilidade, confiabilidade e a correção das funcionalidades em todas as camadas da aplicação (mobile, backend e integrações com APIs externas).

- Testes em Dispositivos Físicos e Variedade de Usuários: A dependência exclusiva de emuladores para testes limitou a validação em cenários reais de uso. Testar o aplicativo em uma ampla gama de dispositivos físicos, com diferentes especificações de hardware e versões de Android, é essencial para identificar e corrigir problemas de compatibilidade, desempenho e usabilidade que não são evidentes em ambientes virtuais genéricos. Além disso, a participação de um número maior de usuários de teste com perfis diversos forneceria um feedback mais rico e abrangente, permitindo identificar pontos de melhoria na interface, na experiência do usuário e na clareza dos relatórios, resultando em um produto mais intuitivo e eficaz para o público final

O Phishing Tracker se consolida como uma ideia promissora e um conceito robusto no cenário da segurança digital, especialmente no combate a ataques de phishing. A grande força da nossa abordagem reside na combinação estratégica da análise de conteúdo com a identificação da localização geográfica do servidor, em adição à tradicional verificação de links. Essa metodologia multifacetada não só eleva a capacidade de detecção de ameaças, mas também agrega um valor educacional inestimável ao processo. Ao apresentar ao usuário não apenas um veredito de "seguro" ou "malicioso", mas também os indícios específicos (como a inconsistência da localização do servidor ou padrões suspeitos no texto da mensagem), o Phishing Tracker incentiva uma dúvida salutar e um senso crítico.

Essa camada adicional de proteção capacita o usuário, transformando-o de um alvo potencial em um agente mais consciente e preparado. Consequentemente, as chances reais de um clique inadvertido em um link incerto diminuem significativamente, fortalecendo a segurança pessoal e corporativa em um ambiente digital cada vez mais propenso a golpes. O Phishing Tracker, portanto, não é apenas uma ferramenta, mas um passo adiante na construção de uma cultura de cibersegurança mais resiliente e informada.

18. REFERENCIAS:

https://www.checkpoint.com/pt/cyber-hub/threat-prevention/what-is-mobile-security/top-6-mobile-security-threats-and-how-to-prevent-them/#:~:text=phishing,-phishing%20%C3%A9%20um&text=Em%20dispositivos%20m%C3%B3veis%2C%20os%20ataques,m%C3%ADdia%20social%20e%20outros%20aplicativos.

https://www.kaspersky.com.br/resource-center/threats/top-seven-mobile-security-threats-smart-phones-tablets-and-mobile-internet-devices-what-the-future-has-in-store

https://www.checkpoint.com/pt/cyber-hub/threat-prevention/what-is-phishing/smishing-vs-phishing/

https://www.ibm.com/br-pt/topics/mobile-security

https://neweseguros.com.br/phishing-dispositivos-dados-seguro-cyber/

https://www.kaspersky.com.br/resource-center/threats/handling-phishing-attacks

https://www.kaspersky.com.br/resource-center/preemptive-safety/phishing-prevention-tips

https://nordvpn.com/pt-br/blog/o-que-e-phishing/

- https://www.avast.com/pt-br/c-phishing ,

- https://www.checkpoint.com/pt/cyber-hub/threat-prevention/what-is-mobile-security/top-6-mobile-security-threats-and-how-to-prevent-them/,

- https://www.oficinadanet.com.br/post/12991-diferenca-entre-virus-spam-spyware-worm-phishing-botnet-rootkit ,

- https://prodest.es.gov.br/entenda-o-que-e-phishing-e-adote-medidas-para-evita-lo#:~:text=Ele%20consiste%20em%20tentativas%20de,e%2Dmail%20com%20conte%C3%BAdo%20duvidoso. ,

- https://www.cnnbrasil.com.br/economia/negocios/brasil-e-vice-campeao-em-ataques-ciberneticos-com-1-379-golpes-por-minuto-aponta-estudo/