Centro de Educação Profissional em Tecnologia da Informação

# SAFELINKS: UMA PLATAFORMA INTEGRADA, SEGURA E INSTRUTIVA

Henry Lira,

Mateus Lago,

Emanoel Araujo.

Curso Técnico de Informática

Petrópolis, 28 novembro de 2025

## Resumo

O Safelinks surge como uma resposta prática e educativa aos crescentes riscos que enfrentamos online. Mais do que uma simples ferramenta, a plataforma foi pensada para unir proteção técnica e aprendizado, criando um ambiente mais seguro para a navegação e as compras na internet. Para isso, o Safelinks aproveita o poder de APIs consolidadas no mercado. O Google Safe Browsing e o VirusTotal atuam como nossos fiáveis verificadores, analisando URLs em tempo real para alertar sobre ameaças. Já a integração com o Google Shopping facilita a busca por produtos apenas em lojas verificadas, trazendo mais tranquilidade na hora das compras. E para garantir que a comunicação com cada usuário seja sempre clara e segura, utilizamos os serviços do Resend. O diferencial da plataforma, porém, vai além da tecnologia. Um sistema de autenticação robusto protege o acesso, enquanto um módulo educativo interativo transforma cada alerta em uma oportunidade de aprendizado. Com ferramentas de histórico e favoritos, os usuários podem acompanhar suas descobertas e sites seguros, construindo aos poucos hábitos digitais mais conscientes. Em resumo, o Safelinks representa uma abordagem proativa para a segurança na internet. Acreditamos que a defesa mais eficaz combina a prevenção técnica com a educação contínua do usuário, e é essa a missão que integramos em cada funcionalidade da nossa plataforma.

**Palavras-chave:** Cibersegurança. Segurança Digital. Educação Digital. Verificação de URLs. Google Safe Browsing. VirusTotal. Google Shopping.

Petrópolis, 28 novembro de 2025

*Projeto Final submetido à Fundação de Apoio à Escola Técnica – FAETEC, Centro de Educação Profissional em Tecnologia, como parte dos requisitos necessários para a obtenção do grau de Técnico em Informática. Sob a orientação da Professora Joseane Ferreira.*

Petrópolis, 28 novembro de 2025

# SUMÁRIO

# 1. INTRODUÇÃO

Com o avanço tecnológico e a popularização do comércio eletrônico, os ciberataques tornaram-se mais frequentes e sofisticados. Segundo dados, apenas em 2024, os ataques de phishing representaram 16% dos incidentes de segurança registrados, com um custo médio de R$7,75 milhões por violação, conforme dados da Sociedade Brasileira de Computação. Neste contexto, consumidores enfrentam desafios significativos para identificar lojas online confiáveis e proteger seus dados pessoais e financeiros.

## 1.1 Problema

A falta de conhecimento sobre cibersegurança por parte dos usuários finais, combinada com a proliferação de lojas virtuais fraudulentas, cria um ambiente propício para golpes online. Muitos consumidores não possuem ferramentas adequadas para verificar a autenticidade das lojas ou para reconhecer ameaças digitais, resultando em prejuízos financeiros e violação de dados.

## 1.2 Objetivos

### 1.2.1 Objetivo Geral

Desenvolver uma plataforma web integrada que una algumas funcionalidades de e-commerce seguro com educação em cibersegurança, proporcionando aos usuários um ambiente confiável para pesquisa de produtos e aprendizado sobre proteção digital.

### 1.2.2 Objetivos Específicos

- Implementar sistema de verificação de URLs utilizando Google Safe Browsing API e VirusTotal.
- Integrar API Google Shopping para busca segura de produtos (observando limitações devido à descontinuação parcial).
- Desenvolver módulo educativo sobre cibersegurança.
- Criar sistema de autenticação com recuperação de senha via Resend API.

- Implementar histórico de pesquisas e favoritos para usuários autenticados.

## 1.3 Metodologia

O desenvolvimento seguiu uma abordagem ágil, com as seguintes etapas:

1º Levantamento de requisitos funcionais e não-funcionais.

2º Projeto da arquitetura de tela

3º Implementação dos módulos principais

4º Integração das APIs externas

5º Testes de funcionalidade e segurança

6º Validação com usuários reais

## 1.4 Organização do texto

Este projeto está organizado em 7 capítulos. No capítulo 2 são apresentados os fundamentos teóricos. O capítulo 3 descreve a arquitetura do sistema. O capítulo 4 detalha a implementação. O capítulo 5 fala sobre o diagrama de sequência. O capítulo 6 é apresentado os diagramas de classes. Finalmente, o capítulo 7 traz as considerações finais.

# 2. FUNDAMENTAÇÃO TEÓRICA

## 2.1 Cibersegurança

A segurança, em qualquer ambiente de rede, envolve múltiplas camadas de proteção. Segundo Tanenbaum (2010), sistemas seguros devem garantir confidencialidade, integridade e disponibilidade dos dados. No contexto de um e-commerce, estes princípios são essenciais para proteger informações financeiras e pessoais dos usuários.

## 2.2 Verificação de urls

Para se proteger de fraudes online, verificar links suspeitos antes de clicar é um passo essencial. Ferramentas como a API do Google Safe Browsing são vitais nesse processo, pois consultam constantemente um enorme banco de dados global que cataloga sites maliciosos. Esse sistema é eficaz para bloquear em tempo real ameaças conhecidas, como páginas de phishing, focos de malware e golpes de engenharia social.

Já o VirusTotal complementa essa proteção ao funcionar como um "canivete suíço" da análise de segurança. Ele consolida a verificação de dezenas de mecanismos antivírus diferentes de uma só vez, o que aumenta significativamente as chances de flagrar uma ameaça. No entanto, é bom ficar atento: a versão pública da ferramenta impõe um limite de uso, permitindo apenas cerca de 4 consultas por minuto, o que pode ser uma restrição para alguns usos mais intensos.

## 2.3 Educação digital

A conscientização dos usuários é considerada a primeira linha de defesa contra ciberataques. Essa premissa parte do princípio, defendido por especialistas como Mitnick (2012), de que o elemento humano é frequentemente o elo mais fraco na segurança da informação. Nesse contexto, programas educativos que abordam temas como senhas seguras, identificação de phishing e navegação segura são comprovadamente eficazes na redução de incidentes de segurança.

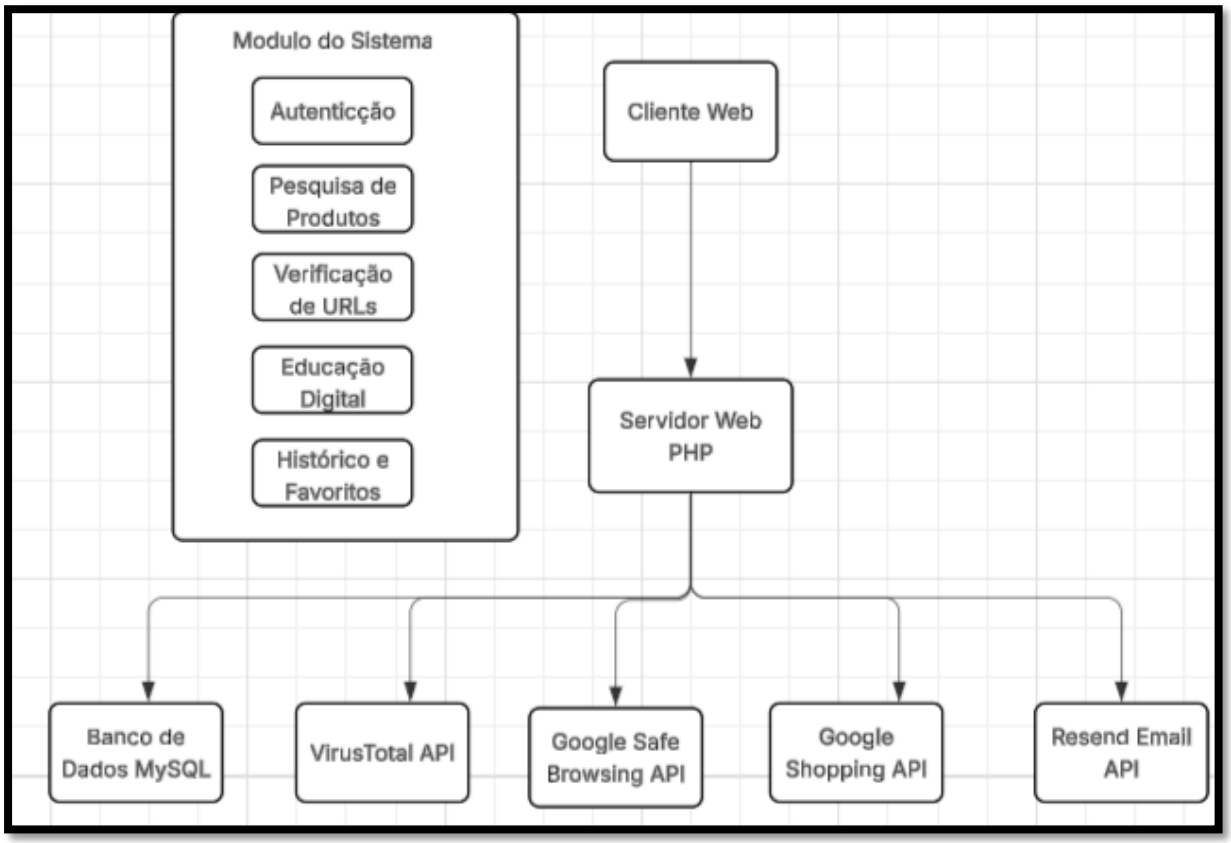
## 2.4 Arquiteturas seguras

Sistemas web modernos devem seguir princípios de segurança by design, incorporando proteções desde a fase de projeto. Isto inclui validação de entrada, autenticação robusta, criptografia de dados e princípio do menor privilégio, aspectos críticos em ambientes e verificação de segurança.

# 3. ARQUITETURA DO SISTEMA

## 3.1 Visão geral da arquitetura

Figura 1 - Arquitetura geral do sistema SafeLinks




## 3.2 Componentes principais

### 3.2.1 Camada de Apresentação

- Interface web responsiva.
- Formulários seguros com validação client-side e server-side.
- Feedback visual imediato sobre status de segurança de URLs.
- Design acessível com suporte a diferentes dispositivos

### 3.2.2 Camada de Aplicação

- Servidor web com PHP 7.4+
- Gerenciamento de sessões seguro com regeneração de ID.
- Controladores com sanitização de entrada e escape de saida.
- Implementação de CSRF tokens em formulários críticos.

### 3.2.3 Camada de Dados

- MySQL 8.0 com tabelas normalizadas e índices otimizados.
- Criptografia de senhas utilizando algoritmo bcrypt.
- Armazenamento seguro de tokens de recuperação.

## 3.3 Integração com APIs

### 3.3.1 Google Safe Browsing

- Verificação em tempo real de URLs contidas nas listas negras.
- Cache de resultados por 1h para otimização.
- Tratamento de erro para indisponibilidade de serviço.

### 3.3.2 Google Shopping

- Busca de produtos por categoria.
- Filtragem por loja.
- Exibição de informações das lojas.

### 3.3.3 VirusTotal

- API Análise multicamada complementar com mais de 65 motores (scanners e listas negras)
- Implementação de rate limiting (4 requisições/minuto)
- Processamento assíncrono para verificações em lote

### 3.3.4 Resend API

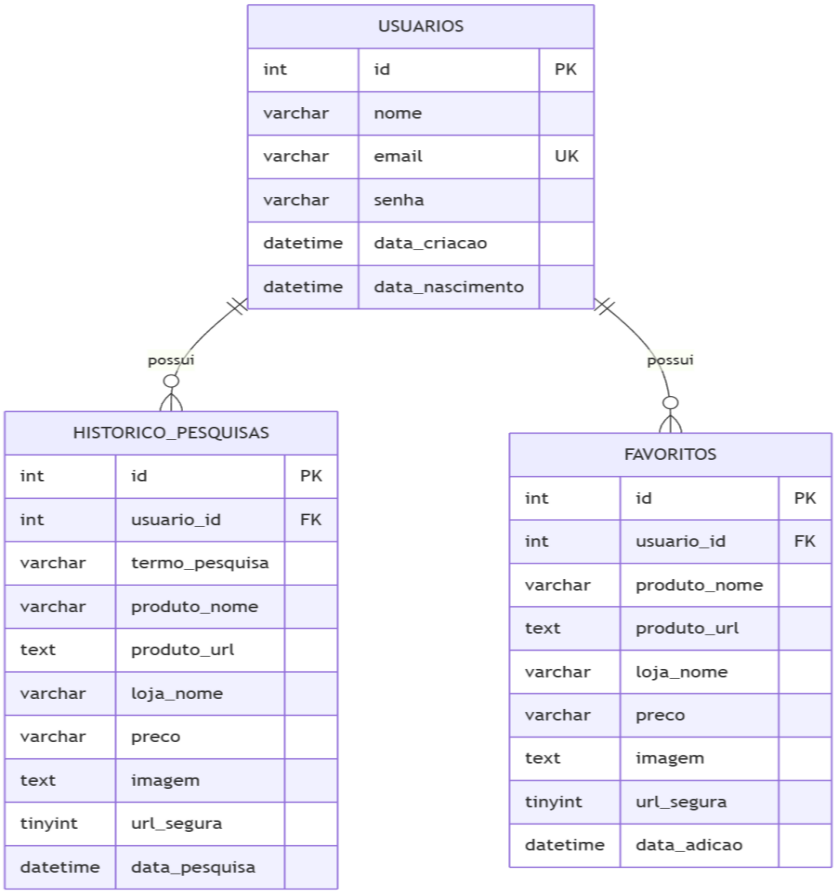
- Envio de emails transacionais para recuperação de senha.
- Templates responsivos com instruções claras.
- Monitoramento de taxa de entrega e possíveis abusos

# 4. IMPLEMENTAÇÃO

## 4.1 Modelo de dados

### 4.1.1 Diagrama Entidade-Relacionamento

Figura 2 - Diagrama entidade-relacionamento do banco de dados




### 4.1.2 Levantamento de Requisitos

Requisitos Funcionais

- RF01: Sistema de verificação multicamada de URLs.
  Desenvolvimento de um módulo central que realiza a verificação de URLs em tempo real, integrado às APIs Google Safe Browsing e VirusTotal. A implementação prevê a criação de um sistema de cache com validade de 60 minutos para otimização de performance e a aplicação de rate limiting, respeitando o limite de 4 requisições por minuto da API VirusTotal. O retorno ao usuário deve consolidar os resultados de ambas as APIs de forma clara.

- RF02: Autenticação de usuários com níveis de acesso.

Implementação de um sistema seguro de autenticação que inclui registro com validação de e-mail, login com proteção contra ataques de força bruta, recuperação de senha via e-mail utilizando a API Resend, e gerenciamento de sessões com regeneração de ID de sessão. O logout deve invalidar completamente a sessão do usuário.

- RF03: Recuperação segura de senha via email.
  A recuperação de senha deve ser feita de forma segura através do email utilizando a API Resend.

- RF04: Histórico e Favorito.
  Desenvolvimento de um sistema de armazenamento de histórico e lojas marcadas como favoritas, exclusivo para usuários autenticados.

- RF05: Sistema de tutoriais e conteúdo educativo
  Criação de uma área de aprendizado com tutoriais sobre criação de senhas seguras, identificação de tentativas de phishing e práticas de navegação segura. O módulo incluirá testes interativos que permitem ao usuário consolidar o conhecimento adquirido, com emissão de certificados após a conclusão de cada etapa.

- RF06: Integração com Google Shopping.
  Implementação de busca de produtos filtrada por categorias, com exibição de informações das lojas e redirecionamento seguro.
  **Observação técnica: Devido à descontinuação da API pública do Google Shopping em setembro de 2025, parte das URLs pode não funcionar conforme o esperado. A funcionalidade permanece como um recurso complementar, sujeito a revisão em versões futuras.*

Requisitos Não-Funcionais

- RNF01: Desempenho
  O sistema deve responder às solicitações de verificação de URLs em até 3 segundos, suportar 1000 usuários simultâneos e manter disponibilidade de 99% durante o horário comercial.

- RNF02: Segurança.

Implementação de criptografia TLS 1.2+ para dados em trânsito, armazenamento de senhas com algoritmo bcrypt (custo 12), proteção CSRF em formulários críticos, sanitização e escape de dados de entrada, e configuração de headers de segurança (CSP, HSTS, X-Frame-Options).

- RNF03: Compatibilidade com principais navegadores
  Interface responsiva compatível com dispositivos móveis, suporte aos navegadores Chrome, Firefox e Safari...

## 4.2 Módulo de verificação de urls (PHP)

```php
class Safebrowsingchecker {
    public function checkUrl($url) {
        $safeBrowsingApiKey = 'sua_chave_safe_browsing';
        $virusTotalApiKey = 'sua_chave_virus_total';
        $safeBrowsingUrl = 'https://safebrowsing.googleapis.com/v4/threatMatches:find';
        $virusTotalUrl = 'https://www.virustotal.com/vtapi/v2/url/report';

        if ($this->canUseVirusTotal()) {
                        $safeBrowsingResult = $this->makeSafeBrowsingCall($safeBrowsingUrl, $url, $safeBrowsingApiKey);
            $virusTotalResult = $this->makeVirusTotalCall($virusTotalUrl, $url, $virusTotalApiKey);

            return [
                'safe_browsing' => $safeBrowsingResult,
                'virus_total' => $virusTotalResult,
                'used_both_apis' => true
            ];
        } else {
            $safeBrowsingResult = $this->makeSafeBrowsingCall($safeBrowsingUrl, $url, $safeBrowsingApiKey);
            return [
                'safe_browsing' => $safeBrowsingResult,
                'virus_total' => null,
                'used_both_apis' => false,
                'rate_limit_exceeded' => true
            ];
        }
    }
    private function canUseVirusTotal() {
        // Implementação do controle de rate limit (4 req/min)
        // Retorna true se pode usar VirusTotal, false se excedeu o limite
        return true; // Placeholder
    }

    private function makeSafeBrowsingCall($apiUrl, $url, $apiKey) {
        $requestData = [
            'client' => [
                'clientId' => 'safelinks',
                'clientVersion' => '1.0'
            ],
            'threatInfo' => [
                'threatTypes' => ['MALWARE', 'SOCIAL_ENGINEERING'],
                'platformTypes' => ['ANY_PLATFORM'],
                'threatEntryTypes' => ['URL'],
                'threatEntries' => [['url' => $url]]
            ]
        ];
        return $this->makeApiCall($apiUrl, $requestData, $apiKey);
    }

    private function makeVirusTotalCall($apiUrl, $url, $apiKey) {
        $params = http_build_query([
            'apikey' => $apiKey,
            'resource' => $url
        ]);
        return $this->makeApiCall($apiUrl . '?' . $params, [], $apiKey, 'GET');
    }
    private function makeApiCall($apiUrl, $requestData, $apiKey, $method = 'POST') {
        // Implementação da chamada API
        return ['success' => true, 'data' => []]; // Placeholder
    }
}
```

## 4.3 Módulo educativo

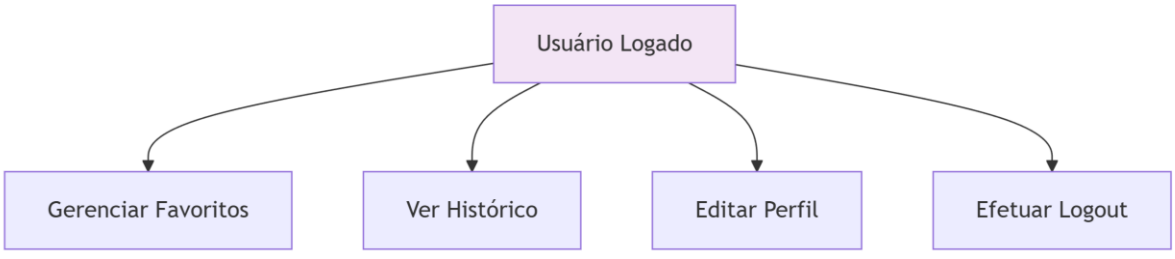
O modulo educativo inclui:

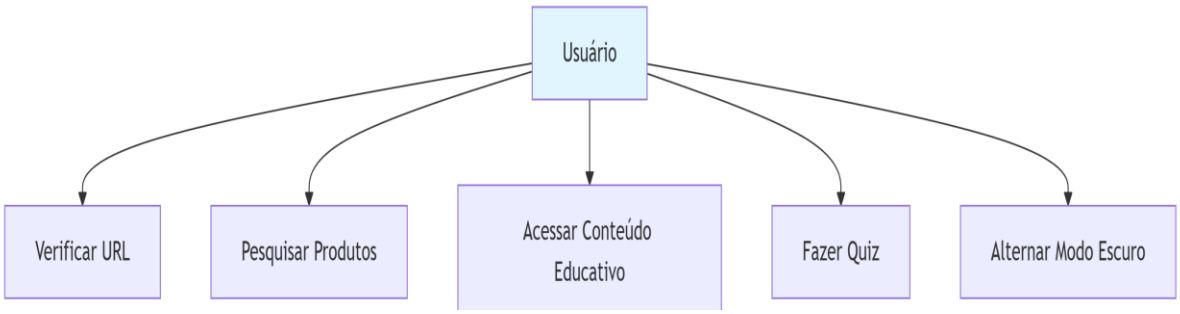
- Tutoriais sobre senhas seguras
- Guias de identificação de phishing
- Dicas de navegação segura
- Testes interativos de conhecimento

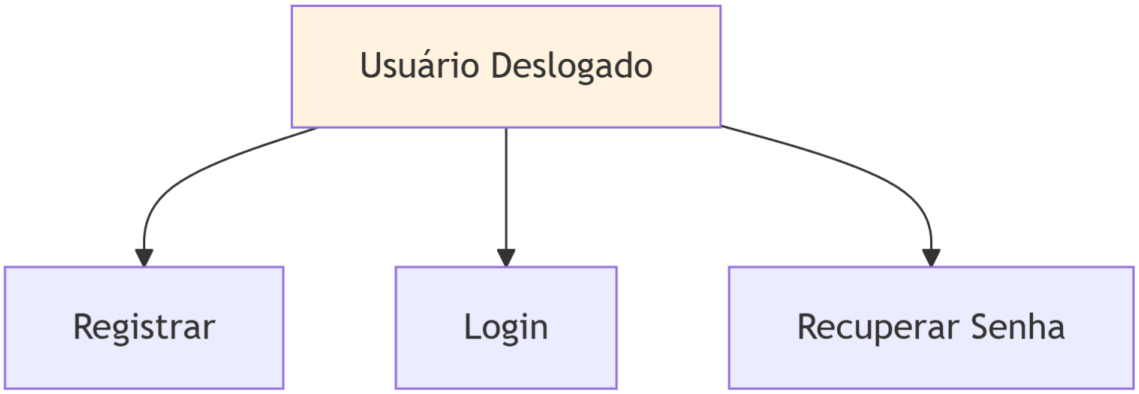
## 4.4 Sistema de autenticação(PHP)

```php
class Authentication {
    public function register($nome, $email, $senha) {
        // Validação de entrada
        $senhaHash = password_hash($senha, PASSWORD_BCRYPT);

        // Inserção no banco com tratamento de exceções
        return $this->createUser($nome, $email, $senhaHash);
    }

    public function recoverPassword($email) {
        // Geração de token seguro
        $token = bin2hex(random_bytes(32));

        // Envio de email via Resend API
        return $this->sendRecoveryEmail($email, $token);
    }
}
```

## 4.5 Caso de uso






**UC-01: Verificação de Segurança de URLs**

**Descrição:** Permite ao usuário verificar se uma URL é segura utilizando APIs de segurança.

**Fluxo Principal:**

Usuário insere URL no campo de pesquisa
Sistema identifica que se trata de uma URL
Sistema consulta Google Safe Browsing API
Sistema consulta VirusTotal API (se dentro do limite de requisições)
Sistema exibe resultado consolidado da verificação
[Extensão] Se usuário autenticado, registro é salvo no histórico

**Pré-condições:** URL deve ser fornecida

**Pós-condições:** Resultado da verificação é exibido

**UC-02: Pesquisa de Produtos**

**Descrição:** Permite pesquisar produtos e visualizar lojas verificadas.

**Fluxo Principal**

Usuário insere termos de produto na pesquisa
Sistema identifica como pesquisa de produtos
Sistema redireciona para página de resultados
Sistema busca produtos via Google Shopping API
Sistema verifica segurança das URLs das lojas
Sistema exibe produtos com indicadores de segurança

**Fluxos Alternativos:**

- **FA-01:** Usuário não autenticado tenta favoritar
  - Sistema redireciona para autenticação
  - Após login, retorna para resultados

**Pré-condições:** Termo de pesquisa deve ser fornecido **Pós-condições:** Lista de produtos é exibida

**UC-03: Autenticação de Usuário**

**Descrição:** Gerencia o registro, login e recuperação de conta.

**Fluxo Principal - Registro:**

Usuário preenche formulário de registro
Sistema valida email único
Sistema cria conta
Sistema envia email de confirmação
Usuário é redirecionado para login

**Fluxo Principal - Login:**

Usuário insere credenciais

Sistema valida autenticidade

[Extensão] Credenciais inválidas: mensagem de erro

Credenciais válidas: sessão é iniciada

**Fluxo Principal - Recuperação:**

Usuário solicita recuperação

Sistema gera token temporário

Sistema envia link via email

Usuário redefine senha

**UC-04: Gestão de Conteúdo Educativo**

**Descrição:** Fornece material educativo sobre segurança digital.

**Fluxo Principal:**

Usuário acessa página "Dicas"

Sistema exibe cards educativos

Usuário interage com conteúdo

Usuário realiza quiz educativo

Sistema fornece feedback imediato

Sistema exibe pontuação final

**Regras:**

- Conteúdo acessível a todos os usuários
- Pontuação do quiz não é persistida

**UC-05: Gestão de Preferências**

**Descrição:** Gerencia favoritos e histórico do usuário.

**Fluxo Principal - Favoritos:**

Usuário autenticado favorita loja.

Sistema armazena referência.

Usuário pode acessar lista de favoritos.

Usuário pode remover favoritos.

**Fluxo Principal - Histórico:**

Usuário autenticado acessa histórico.

Sistema exibe verificações anteriores.

Usuário pode filtrar resultados.

Pré-condições: Usuário deve estar autenticado.

**UC-06: Personalização de Interface**

Descrição: Gerencia preferências de interface do usuário.

Fluxo Principal**:**

Usuário alterna modo claro/escuro

Sistema aplica tema selecionado

Sistema persiste preferência

Tema é mantido em sessões futuras

## Tabela de Resumo

| ID Caso Uso | Nome | Atores | Complexidade |
|---|---|---|---|
| UC-01 | Verificação URLs | Todos | Alta |
| UC-02 | Pesquisa Produtos | Todos | Média |
| UC-03 | Autenticação | Todos | Alta |
| UC-04 | Conteúdo Educativo | Todos | Baixa |
| UC-05 | Gestão Preferências | Autenticados | Média |

| | | | |
|---|---|---|---|
| UC-06 | Personalização | Todos | Baixa |

### Regras de Negócio Principais

**RN-01:** Verificação de URLs utiliza duas APIs complementares

**RN-02:** Rate limiting de 4 requisições/minuto para VirusTotal

**RN-03:** Usuários não autenticados podem verificar URLs mas não salvam histórico

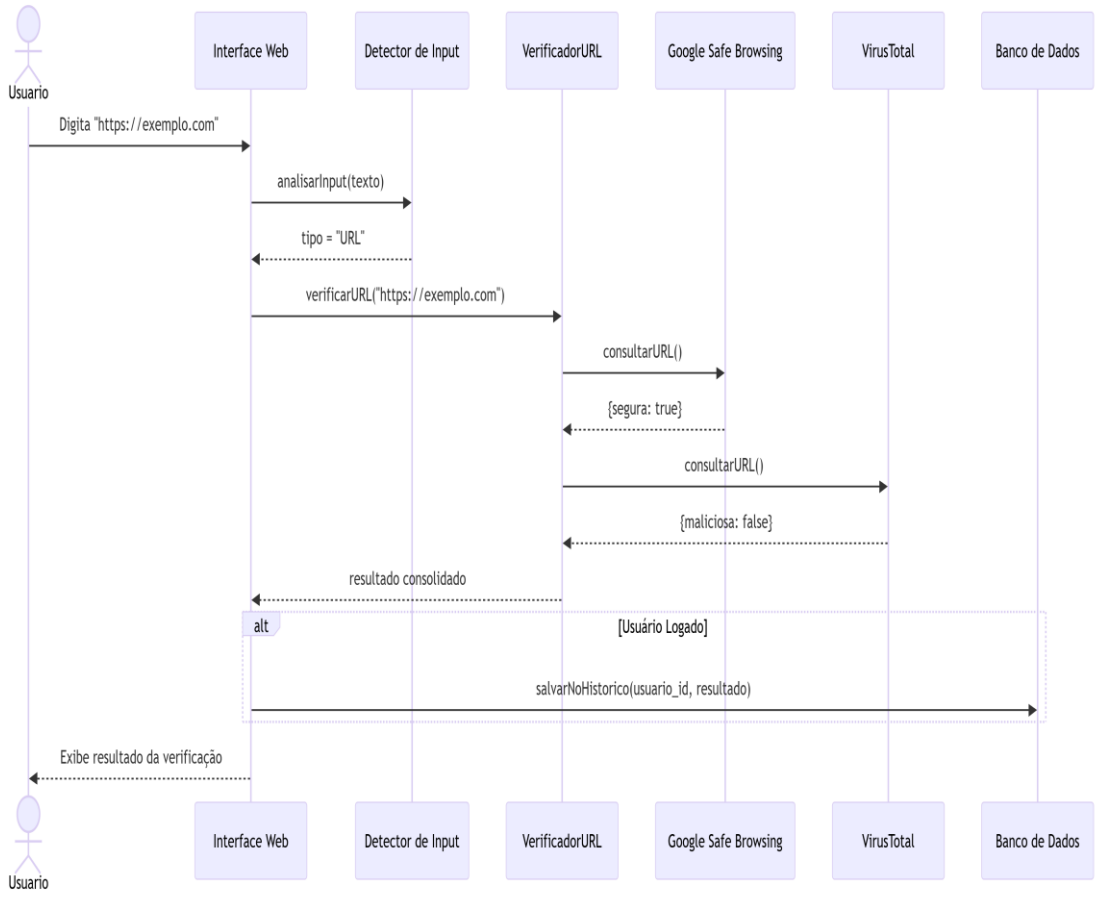
**RN-04:** Conteúdo educativo igual para todos os usuários

**RN-05:** Favoritos restritos a usuários autenticados

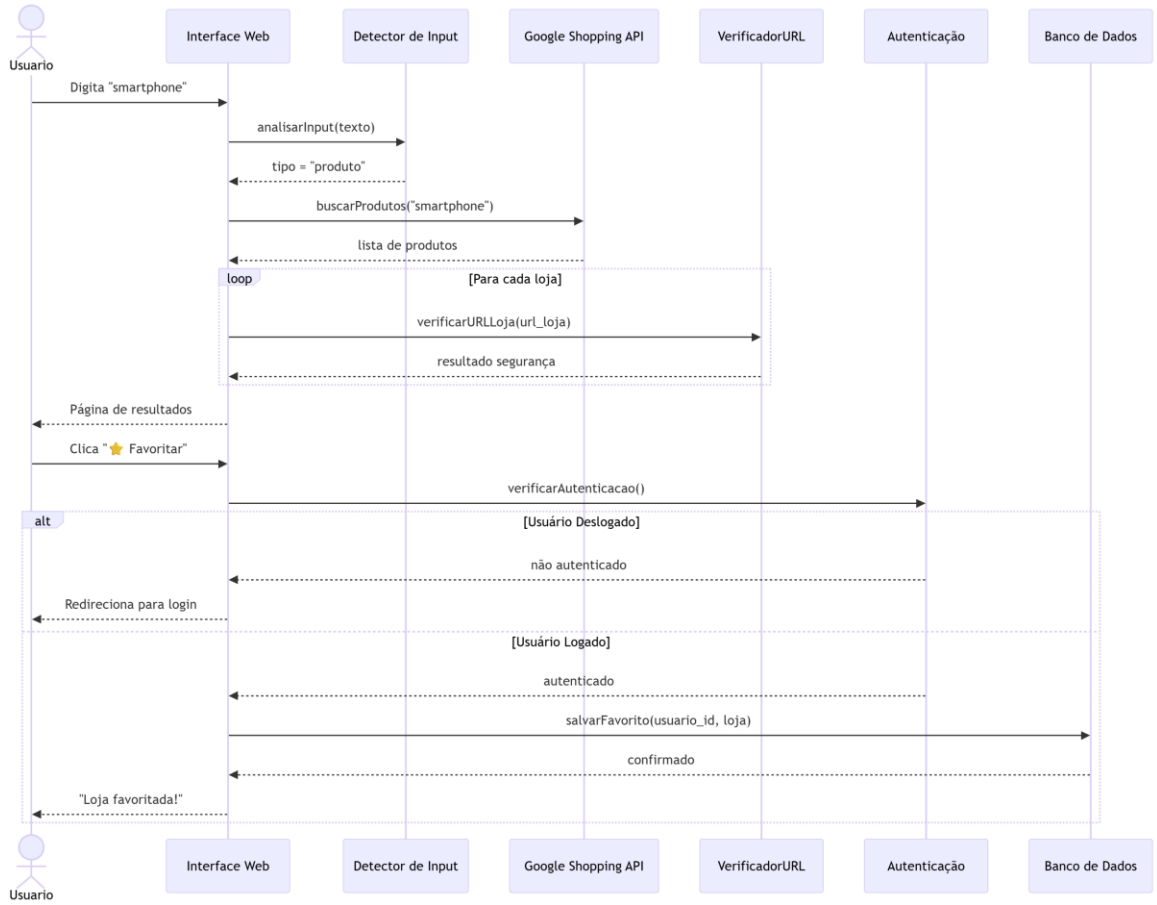
**RN-06:** Preferências de interface persistem no navegador

## 5. DIAGRAMA DE SEQUÊNCIA

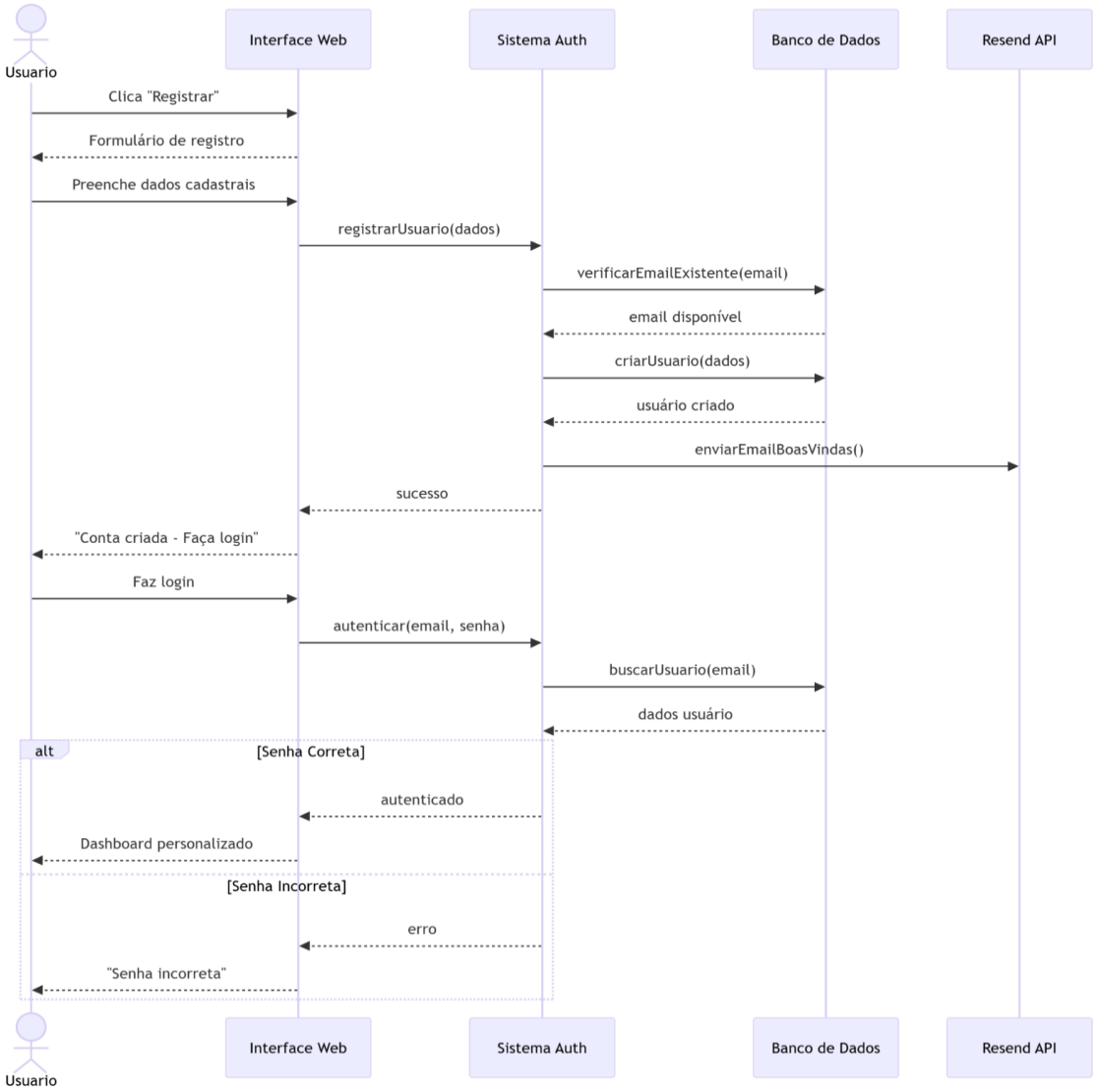
Os diagramas a seguir mostram, de maneira visual, como o usuário interage com o nosso sistema em cada funcionalidade principal. É como se estivéssemos contando uma história de como cada tarefa é realizada, mostrando quem conversa com quem e em que ordem.



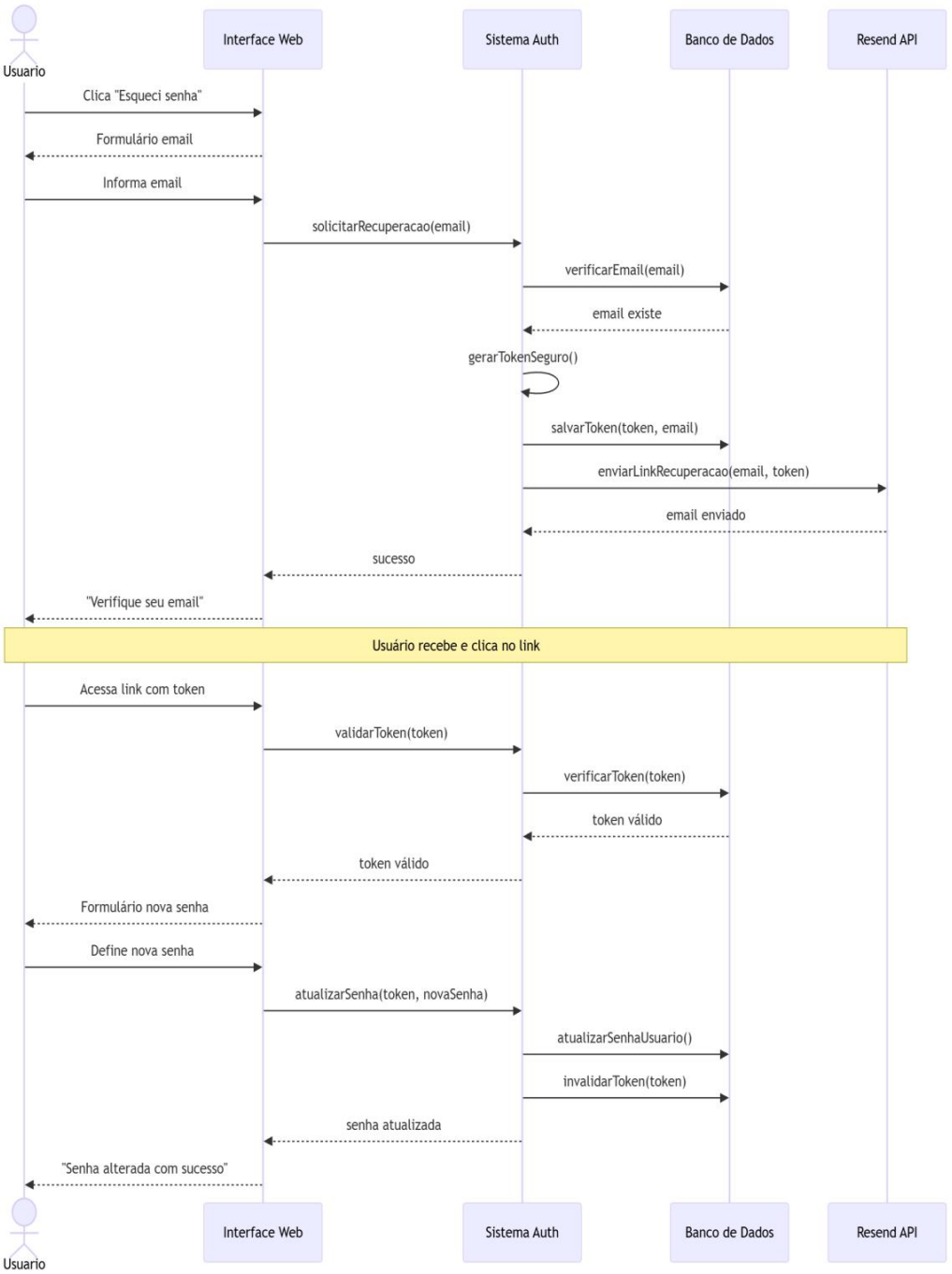

**Verificação de URL:** Mostra o processo completo desde que o usuário digita uma URL até receber o resultado da verificação de segurança, incluindo as consultas às APIs externas.



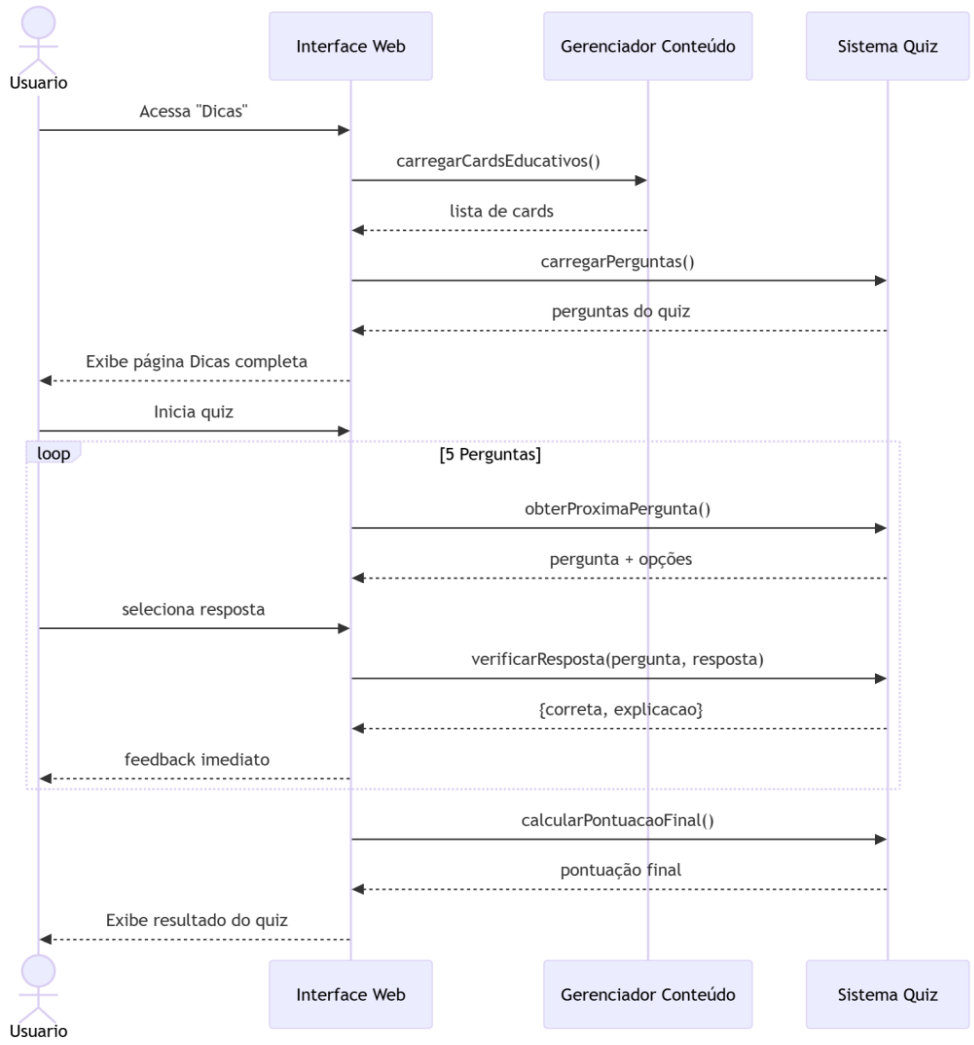

**Pesquisa de Produtos:** Ilustra como o sistema lida com pesquisas de produtos, desde a busca nas lojas até o processo de favoritar, incluindo o redirecionamento para login quando necessário.



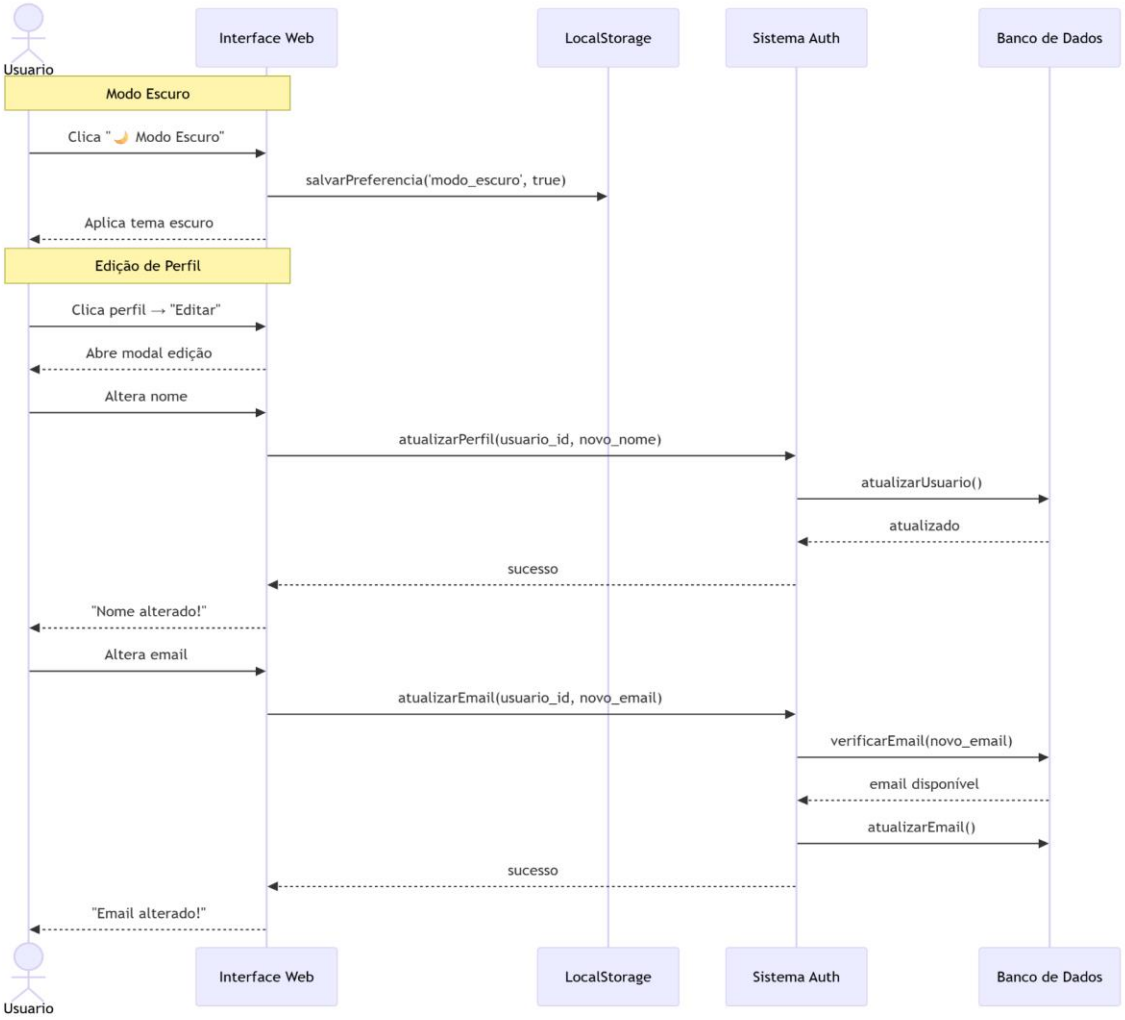

**Registro e Login:** Demonstra o fluxo de criação de conta e autenticação do usuário, com tratamento de erros como senha incorreta.



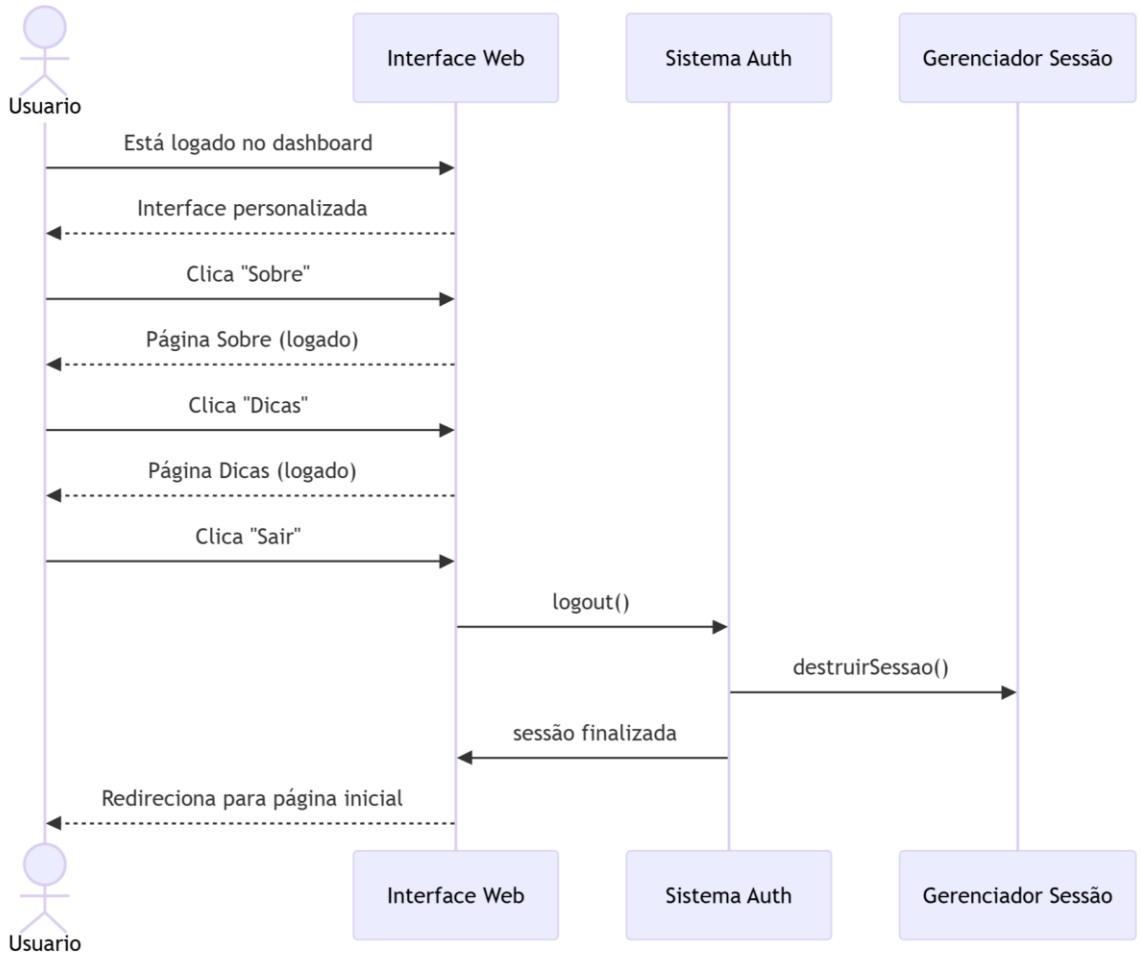

**Recuperação de Senha:** Mostra o processo seguro de recuperação de conta usando tokens temporários e envio de e-mail.




**Conteúdo Educativo:** Apresenta como o sistema carrega e gerencia o conteúdo educativo e o quiz interativo.

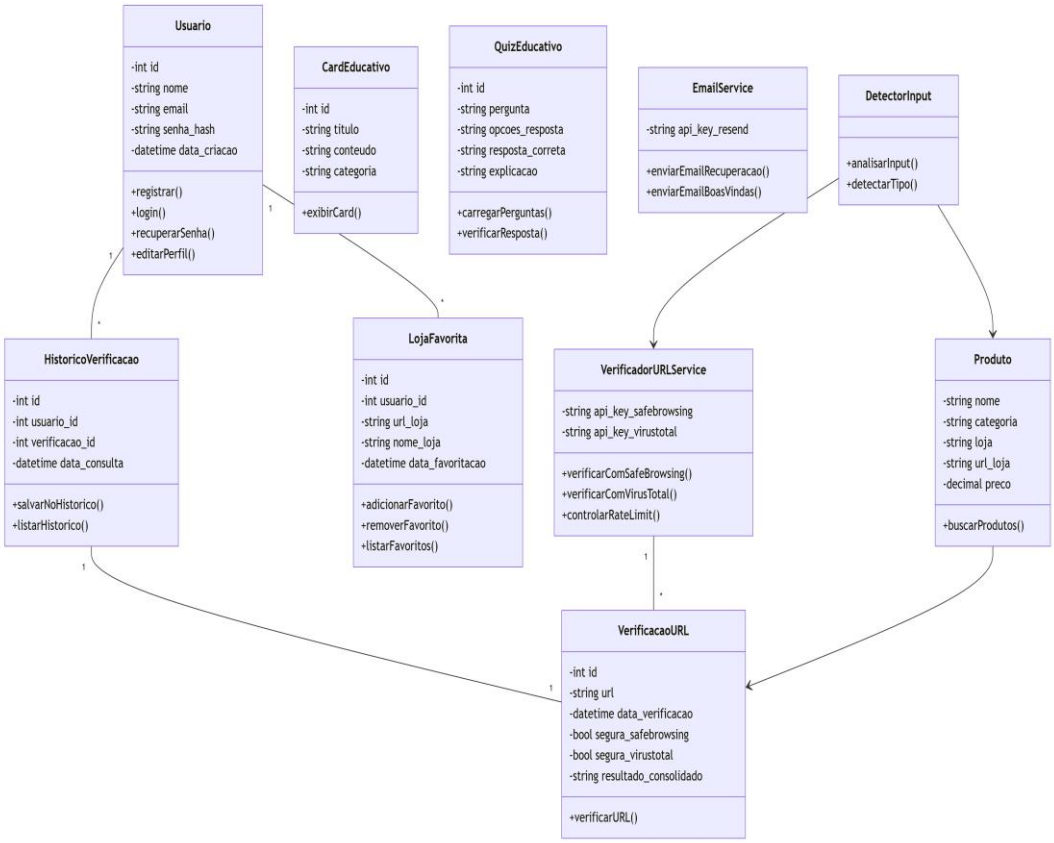



**Modo Escuro e Perfil:** Ilustra a funcionalidade de personalização da interface e edição do perfil do usuário.




**Logout e Navegação:** Mostra o fluxo de encerramento de sessão e navegação entre páginas.

## 6. DIAGRAMA DE CLASSES




O diagrama de classes do SafeLinks mostra como o sistema é organizado por dentro, apresentando as principais "peças" que fazem tudo funcionar. É como um mapa que explica como cada parte do sistema se conecta com as outras.

**Classes Principais do Sistema:**

- **Usuario:** Cuida de tudo relacionado aos usuários - cadastro, login e perfil
- **VerificacaoURL:** Guarda os resultados das verificações de links que fazemos
- **HistoricoVerificacao:** Mantém o histórico das verificações para usuários cadastrados

- **LojaFavorita:** Armazena as lojas que os usuários marcaram como favoritas

**Classes que Fazem o Trabalho Pesado:**

- **VerificadorURLService:** É o responsável por conversar com as APIs do Google Safe Browsing e VirusTotal
- **EmailService:** Envia os e-mails de recuperação de senha usando o Resend
- **DetectorInput:** Decide se o que o usuário digitou é uma URL ou produto para pesquisa

**Classes de Apoio:**

- **Produto:** Representa os produtos que buscamos no Google Shopping
- **CardEducativo/QuizEducativo:** Cuidam dos conteúdos educativos que mostramos aos usuários

**Como tudo se conecta:**

- Cada usuário pode ter várias verificações no histórico (1 para muitos)
- O verificador de URLs cria e gerencia as verificações
- O detector de input trabalha junto com os serviços de verificação e produtos

Organização que facilita a manutenção:
A forma como separamos as classes faz com que o sistema seja mais fácil de entender e melhorar. Se precisarmos mudar algo nas verificações, por exemplo, só mexemos na parte responsável por isso sem afetar o resto do sistema.

Essa estrutura também nos permite adicionar novas funcionalidades no futuro sem precisar refazer tudo do zero.

# 7. CONSIDERAÇÕES FINAIS

## 7.1 Conclusões

O SafeLinks demonstra ser uma solução eficaz, combinando prevenção técnica com educação do usuário. A integração com APIs consolidadas permitiu criar um sistema robusto e confiável, enquanto o módulo educativo aborda a raiz do problema através da conscientização.

## 7.2 Contribuições

- Plataforma integrada
- Design limpo
- Metodologia de verificação em tempo real
- Conteúdo educativo acessível e prático
- Arquitetura escalável e segura

## 7.3 Limitações

- Dependência de APIs externas
- Cobertura limitada para URLs muito recentes
- Necessidade de atualização constante do conteúdo educativo
- Só podemos garantir a segurança dentro do nosso site
- Dependência de APIs externas, com impacto direto na funcionalidade de busca de produtos após a descontinuação do acesso público ao Google Shopping em setembro de 2025.

## 7.4 Trabalhos futuros

- Expansão para aplicativo móvel
- Integração com mais APIs de verificação

- Sistema de reputação de lojas baseado em comunidade
- Conteúdo educativo mais interativos
- Análise preditiva de novas ameaças
- Migração para a nova API do Google Shopping (Merchant Center) ou adoção de alternativas como APIs de comércio eletrônico de outros provedores, visando restabelecer a busca completa de produtos.

## 7.5 Orçamento

| Item | Quant. | Cust. Unidade | Cust. Total |
|---|---|---|---|
| Domínio | 1 ano | R$40,00 | R$40,00 |
| Hospedagem (Hostinger Single) | 48 meses | R$287,52 | R$287,52 |
| Desvl. BackEnd | 640hrs. | R$50,00 | R$32.000,00 |
| Desvl. FrontEnd | 640hrs. | R$50,00 | R$32.000,00 |
| Safe Browsing API | - | - | Gratuito |
| Virus Total API | - | - | Gratuito |
| Resend API | - | - | Gratuito |
| Google Shopping API | - | - | Gratuito |
| Total | | | R$64.327,52 |

*Alguns valores podem sofrer ajustes dependendo da necessidade.

## 7.6 Cronograma

| Fase | Semana |
|---|---|
| Planejamento<br>• Levantamento de Requisitos<br>• Análise de Viabilidade | 1 a 3 |
| Projeto<br>• Arquitetura do Sistema<br>• Diagramas<br>• Prototipos de Interface | 4 a 6 |
| Implementação<br>• Desenvolvimento BackEnd<br>• Desenvolvimento FrontEnd<br>• Integração das APIs | 7 a 13 |
| Testes<br>• Testes Funcionais<br>• Testes de Segurança | 14 a 16 |

## 7.7 Impacto social

O SafeLinks contribui para uma internet mais segura, empoderando usuários com ferramentas e conhecimento para se protegerem. A redução de incidentes de segurança beneficia tanto consumidores quanto comerciantes legítimos, fortalecendo a confiança no e-commerce e na própria web como um todo.

## REFERÊNCIAS BIBLIOGRÁFICAS


ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR 6022: Informação e documentação: artigo em publicação periódica científica impressa: apresentação. Rio de Janeiro, 2003. <Acesso em: 01 outubro 2025.

BOVO, Alessandro Botelho. Um Modelo de descoberta de conhecimento inerente à evolução temporal dos relacionamentos entre elementos textuais. 2011. 155 p. Tese - Universidade Federal de Santa Catarina, Florianópolis, 2011 <Acesso em: 01 outubro 2025

GOOGLE. Safe Browsing API Documentation. 2025. Disponível em: https://developers.google.com/safe-browsing/?hl=pt_BR <Acesso em: 01 outubro 2025.

GOOGLE. Shopping API Documentation. 2025. Disponivel em: https://console.cloud.google.com/apis/library/shoppingcontent.googleapis.com?hl=pt-br&project=loyal-operation-472102-r5 < Acesso em: 30 outubro 2025.

MITNICK, Kevin; SIMON, William. **A Arte de Enganar**. Tradução de Ronaldo Sérgio de Biasi. 2. ed. Rio de Janeiro: Alta Books, 2012. < Acesso em: 27 do outubro 2025.

Por OPUS TECNOLOGIA, “Phishing: você está atento a golpes digitais?”, 16/06/2025, Disponivel em: https://g1.globo.com/pr/parana/especial-publicitario/opus-tech/noticia/2025/06/16/phishing-voce-esta-atento-a-golpes-digitais.ghtml <Acesso em: 01 outubro 2025

RESEND. Resend API Documentation. 2025. Disponível em: https://resend.com/docs/introduction <Acesso em: 01 outubro 2025

SILVA, Thales do Nascimento. Uma arquitetura para descoberta de conhecimento a partir de bases textuais. 2012. 78 f. Trabalho de Conclusão de Curso - Universidade Federal de Santa Catarina, Araranguá, 2012. <Acesso em: 01 outubro 2025

TANENBAUM, Andrew S.; BOS, Herbert. Sistemas Operacionais Modernos. 4. ed. São Paulo: Pearson Education do Brasil, 2016. Disponível em: https://www.kufunda.net/publicdocs/Sistemas%20Operacionais%20Modernos%20(Andrew%20S.%20Tanenbaum,%20Herbert%20Bos).pdf.<Acesso em: 01 outubro 2025.

TANENBAUM, A. S.; STEEN, M. V. Sistemas distribuídos: princípios e paradigmas. 2. ed. São Paulo: Pearson Prentice Hall, 2007. <Acesso em: 01 outubro 2025.

VIRUSTOTAL. VirusTotal API v3 Documentation. 2025. Disponível em: https://developers.virustotal.com/reference < Acesso em 28 outubro de 2025

## Glossário

Verificação Multicamada: Estratégia de segurança que utiliza múltiplas APIs e técnicas complementares para aumentar a eficácia na detecção de ameaças.

Rate Limiting: Mecanismo de controle que limita a quantidade de requisições que um usuário pode fazer a uma API em determinado período de tempo.

Security by Design: Princípio de desenvolvimento que incorpora considerações de segurança desde as fases iniciais do projeto.

API REST: Interface de programação de aplicações que segue os princípios REST para comunicação entre sistemas.

Modelo Conceitual: Representação abstrata da arquitetura do sistema para fins de documentação e compreensão, não necessariamente refletindo a implementação literal.

Implementação Distribuída: Arquitetura onde funcionalidades são divididas em múltiplos arquivos e scripts que trabalham em conjunto.

Dessa forma, a documentação mantém a clareza arquitetural do modelo conceitual enquanto deixa explícito que a implementação real segue uma abordagem diferente, evitando qualquer má interpretação pelos avaliadores.