

**FATEC SÃO ROQUE - DR. BERNARDINO DE
CAMPOS**

**EMANUELE CRISTINA DA SILVA
JOÃO PAULO DE SOUSA**

**APLICATIVO DE APRENDIZADO EM
CIBERSEGURANÇA: PROJETO E
DESENVOLVIMENTO DO TATUPRO**

**SÃO ROQUE - SP
2025**

FATEC SÃO ROQUE - DR. BERNARDINO DE CAMPOS

APLICATIVO DE APRENDIZADO EM CIBERSEGURANÇA: PROJETO E DESENVOLVIMENTO DO TATUPRO

Trabalho de conclusão de curso
apresentado à FATEC SÃO ROQUE
– Dr Bernardino de Campos como
requisito para obtenção do diploma
do Curso **SISTEMAS PARA
INTERNET**.

Orientador(a): Professor
Especialista Marcos Costa de Sousa

**SÃO ROQUE - SP
2025**

FICHA CATALOGRÁFICA

Aplicativo de aprendizado em cibersegurança: projeto e desenvolvimento do TatuPro

SILVA, Emanuele Cristina da; SOUSA, João Paulo de.

Silva e Sousa. 2025

Orientador(a): Professor Especialista Marcos Costa de Sousa.

Páginas 000 f.: il.; 31cm

TCC de Graduação Tecnológica (**Sistemas para Internet**) -

FATEC SÃO ROQUE - SP

1. Cibersegurança. 2. Aplicativo móvel. 3. Educação digital.

4. Inteligência artificial. 5. Segurança da informação.

FATEC SÃO ROQUE - DR. BERNARDINO DE CAMPOS

AGRADECIMENTOS

Gostaríamos de agradecer, primeiramente, as nossas famílias por nos apoiarem nesses três anos e nos darem todo o suporte. Sem vocês nada disso seria possível!

Ao nosso orientador Professor Especialista Marcos Costa de Sousa por ter acolhido nossa ideia e dar-nos todo o suporte possível!

Aos amigos que fizeram parte desta jornada.

Aos professores, pela paciência e ensinamentos.

E, principalmente, um ao outro, por sermos parceiros de caminhada e superação, por mantermos a confiança no nosso trabalho mesmo nos momentos mais difíceis.

RESUMO

O presente trabalho descreve o projeto e desenvolvimento do TatuPro, um aplicativo mobile educativo voltado à conscientização sobre cibersegurança para usuários leigos. Com o crescente número de crimes cibernéticos no Brasil, onde mais de 4,5 mil pessoas sofrem tentativas de golpes por hora, identificou-se a necessidade de uma ferramenta acessível que promova educação digital preventiva. O objetivo geral consistiu em desenvolver uma aplicação que oferecesse conteúdos interativos, dicas práticas e suporte personalizado via chatbot. A metodologia envolveu pesquisa de aplicativos similares, definição de personas, levantamento de requisitos funcionais e não funcionais, prototipagem em Figma e teste de usabilidade com 27 participantes. O desenvolvimento utilizou Flutter para o *front end*, PHP (PHP: *Hypertext Preprocessor*) com arquitetura MVC (*Model View Controller*) e padrão RESTful para o *backend*, MySQL como banco de dados relacional e a API Gemini para o chatbot. Adicionalmente, foi implementado um módulo de geração automatizada de testes utilizando n8n e *Docker*, com técnicas de RAG para garantir conteúdos embasados em documentos oficiais. Os testes de usabilidade revelaram necessidade de ajustes na identidade visual e navegação, resultando em reformulações na paleta de cores e implementação de barra de navegação inferior com labels. Como resultado, obteve-se um aplicativo funcional, intuitivo e escalável que contribui para a formação de uma cultura de segurança digital mais consciente e preventiva no Brasil.

Palavras-chave: Cibersegurança. Aplicação Mobile. Educação Digital. Inteligência Artificial. Flutter.

ABSTRACT

This work describes the design and development of *TatuPro*, an educational mobile application focused on cybersecurity awareness for lay users. With the growing number of cybercrimes in Brazil, where more than 4,500 people suffer fraud attempts per hour, the need for an accessible tool promoting preventive digital education was identified. The general objective was to develop an application offering interactive content, practical tips, and personalized support via chatbot. The methodology involved research on similar applications, persona definition, functional and non-functional requirements gathering, prototyping in Figma, and usability testing with 27 participants. The development used Flutter for the frontend, PHP (PHP: Hypertext Preprocessor) with MVC (Model View Controller) architecture and RESTful pattern for the backend, MySQL as relational database, and Gemini API for the chatbot. Additionally, an automated test generation module was implemented using n8n and Docker, with RAG techniques to ensure content based on official documents. Usability tests revealed the need for adjustments in visual identity and navigation, resulting in color palette reformulations and implementation of a bottom navigation bar with labels. As a result, a functional, intuitive, and scalable application was obtained, contributing to the formation of a more conscious and preventive digital security culture in Brazil.

Keywords: Cybersecurity. Mobile application. Digital education. Artificial intelligence. Flutter.

SUMÁRIO

1. INTRODUÇÃO.....	10
2. REVISÃO DE LITERATURA.....	11
3. JUSTIFICATIVA.....	13
4. OBJETIVOS.....	14
4.1 Objetivo Geral.....	14
4.2 Objetivos Específicos.....	14
5. MATERIAIS E MÉTODOS.....	14
5.1. Benchmarking.....	15
5.1.1. HackerX - Learn Ethical Hacking.....	16
5.1.2. Cibersegurança e Antivírus.....	16
5.1.3. Learn Cyber Security.....	16
5.2. Personas e Público-alvo.....	16
5.3. Requisitos.....	18
5.3.1. Requisitos Funcionais.....	18
5.3.2. Requisitos Não Funcionais.....	19
5.4. Diagrama de Casos de Uso.....	19
5.5. Diagrama de Classes.....	20
5.6. Diagrama de Sequência.....	22
5.7. Fluxo de Navegação e Interação.....	25
5.8. Arquitetura da Informação.....	27
5.9. Teste de Usabilidade.....	27
5.9.1. Metodologia usada no teste.....	28
5.9.2. Perfil dos participantes.....	28
5.9.3. Problemas Identificados.....	30
5.9.4. Alterações Realizadas.....	33
5.9.4.1. Paleta de cores.....	33
5.9.4.2. Reformulação dos botões de navegação.....	34
5.9.4.3. Reorganização do layout das telas.....	36
5.9.5. Considerações sobre o Teste realizado.....	40
5.10. Tecnologias.....	40
5.10.1. Flutter.....	40
5.10.2. Dart.....	41
5.10.3. Gemini.....	41
5.10.4. Banco de Dados.....	42
5.10.5. Diagrama Entidade Relacionamento.....	43
5.10.6. PHP.....	44
6. DESENVOLVIMENTO.....	45
6.1. Backend.....	45
6.1.1. API RESTful.....	45

6.1.2. Modelo MVC (Model-View-Controller).....	46
6.1.3. Framework.....	46
6.2. Banco de Dados.....	49
6.3. FrontEnd.....	51
6.3.1. Chatbot.....	53
6.4. Módulo de Geração Automatizada de Testes.....	54
7. CONCLUSÃO DO TRABALHO.....	58
8. REFERÊNCIAS.....	60

LISTA DE FIGURAS

Figura 1 – Persona Josué Costa.....	17
Figura 2 – Persona Tatiane Santos.....	17
Figura 3 – Persona Maria Ferreira.....	18
Figura 4 – Diagrama de Casos de Uso.....	20
Figura 5 – Diagrama de Classes.....	21
Figura 6 – Diagrama de sequência - Login.....	22
Figura 7 – Diagrama de sequência - Edição de Usuário.....	23
Figura 8 – Diagrama de sequência - Correção de Testes.....	23
Figura 9 – Diagrama de sequência - Consultar Dicas.....	24
Figura 10 – Diagrama de sequência - Consultar Testes.....	24
Figura 11 – Diagrama de sequência - Exclusão de Usuário.....	25
Figura 12 – Fluxo de Navegação - Login.....	25
Figura 13 – Fluxo de Navegação - Chatbot.....	26
Figura 14 – Fluxo de Navegação - Testes.....	26
Figura 15 – Arquitetura da Informação.....	27
Figura 16 – Faixa Etária dos Participantes.....	28
Figura 17 – Gênero dos Participantes.....	29
Figura 18 – Ocupação dos participantes.....	29
Figura 19 – Dificuldades encontradas pelos participantes.....	30
Figura 20 – Entendimento sobre o objetivo do aplicativo.....	30
Figura 21 – Avaliação geral da experiência do usuário.....	31
Figura 22 – Pontos positivos e negativos.....	31
Figura 23 – Pontos positivos e negativos - cores do aplicativo.....	32
Figura 24 – Pontos positivos e negativos - dificuldades.....	32
Figura 25 – Pontos positivos e negativos - mudança geral.....	32
Figura 26 – Possível recomendação do aplicativo.....	33
Figura 27 – Paleta de cores inicial.....	33
Figura 28 – Paleta de cores alterada.....	34
Figura 29 – Appbar versão 1.....	34
Figura 30 – Appbar versão 2.....	34
Figura 31 – Botões na tela inicial do aplicativo.....	35
Figura 32 – Menu de navegação lateral.....	35
Figura 33 – Bottom Navigation.....	36
Figura 34 – Telas de navegação - login versão 1.....	36
Figura 35 – Telas de navegação - login versão 2.....	36
Figura 36 – Telas principais versão 1.....	37
Figura 37 – Telas principais versão 2.....	37
Figura 38 – Telas principais versão 1.....	38
Figura 39 – Telas principais versão 2.....	38
Figura 40 – Feedback versão 1.....	39

Figura 41 – Feedback versão 2.....	39
Figura 42 – Logotipo do Flutter.....	41
Figura 43 – Logotipo da Dart.....	41
Figura 44 – Logotipo do Gemini.....	42
Figura 45 – Logotipo do MySQL.....	43
Figura 46 – Diagrama de Entidade e Relacionamento.....	43
Figura 47 – Logotipo do PHP.....	44
Figura 48 – Estrutura Framework e classe “core/Routes”.....	48
Figura 49 – Logotipo do Docker.....	55
Figura 50 – Logotipo do n8n.....	55
Figura 51 – Nós n8n.....	55
Figura 52 – Modelo DEMO n8n.....	56
Figura 53 – Fluxo de automação de testes no n8n.....	57
Figura 54 – Dashboard Google IA Studio.....	58

1. INTRODUÇÃO

De acordo com o Instituto Brasileiro de Geografia e Estatística (IBGE), a digitalização crescente das relações sociais, comerciais e financeiras tem proporcionado inúmeros benefícios à sociedade, como agilidade, praticidade e conectividade (IBGE, 2024). No entanto, esse avanço também tem intensificado a exposição de usuários a riscos no ambiente virtual, tornando a cibersegurança um tema cada vez mais relevante e urgente. No Brasil, os dados são alarmantes. Uma pesquisa realizada pelo Instituto Datafolha, em parceria com o Fórum Brasileiro de Segurança Pública (FBSP), revelou que mais de 4,5 mil pessoas são alvo de tentativas de golpes financeiros por hora no país (CNN BRASIL, 2024). Tais golpes ocorrem, majoritariamente, por meio de aplicativos de mensagens ou ligações telefônicas, em que os criminosos se passam por representantes de instituições bancárias.

O mesmo levantamento identificou que cerca de 2.500 pessoas relataram não ter recebido produtos adquiridos em plataformas virtuais, e outras 1.680 foram vítimas de furto ou roubo de seus dispositivos móveis. Outro aspecto preocupante é a subnotificação dessas ocorrências: apenas 30% dos entrevistados que sofreram golpes com Pix ou boletos falsos informaram ter registrado boletim de ocorrência (CNN BRASIL, 2024). Esses dados evidenciam não apenas o crescimento dos crimes cibernéticos, mas também a carência de conhecimento, por parte da população, sobre práticas básicas de proteção digital. A falta de orientação adequada, sobretudo entre usuários leigos ou com pouca familiaridade com tecnologia, contribui para a manutenção de um cenário de vulnerabilidade digital generalizada.

Diante dessa realidade, o presente Trabalho de Graduação tem como objetivo apresentar o projeto e desenvolvimento do TatuPro, um aplicativo educativo voltado à promoção da cibersegurança, com foco na capacitação de usuários comuns. A proposta do TatuPro é disponibilizar, de forma acessível e interativa, conteúdos educativos, dicas práticas e simulações de golpes digitais, permitindo que o usuário reconheça ameaças e adote condutas seguras no ambiente virtual. O nome do aplicativo remete ao tatu, animal típico da fauna brasileira, cuja carapaça serve de proteção, simbolizando a missão do projeto: oferecer uma armadura digital ao cidadão comum. Ao integrar tecnologia, educação e usabilidade, o TatuPro visa contribuir para a formação de uma cultura de segurança digital mais sólida, consciente e preventiva, acessível a diferentes perfis de usuários.

2. REVISÃO DE LITERATURA

A tecnologia é repleta de benefícios como melhorias na telecomunicação, facilitação de processos, economia de espaço físico e assim por diante. No entanto, também existem experiências negativas, como os crimes ocorridos nesse meio devido a virtualização das riquezas e o nascimento de novos bens imateriais exclusivos do meio digital (NOLASCO, SILVA, 2022). De acordo com Nolasco e Silva (2022) “percebe-se, no entanto, que o arcabouço jurídico sobre o assunto é esparso, estando contido em diferentes instrumentos normativos, de maneira a dificultar a unificação da linguagem a comunicação entre os agentes multisectoriais” (NOLASCO, SILVA, 2023, p. 2376).

Santos e Araújo (2023) apontam que a falta de segurança cibernética pode ter impactos negativos significativos, como por exemplo o vazamento de informações sensíveis que pode levar à exposição de dados pessoais e financeiros, resultando em prejuízos financeiros. A respeito disso, Nolasco e Silva (2022) afirmam:

Em virtude disso, resta nítida a necessidade de uma atuação mais precisa, com um código e instituições que reúnam as atribuições e assuntos pertinentes à cibersegurança, a fim de atuar de modo mais eficaz e conciso na esteira do combate à criminalidade virtual e à proteção dos dados e informações. Uma vez comprovada a pungência dos prejuízos advindos de tal delinquência, pode-se captar a urgência de investimentos massivos na educação em cibersegurança e no suporte às equipes de combate ao crime virtual, assim como a majoração das penas de modo a desestimular a prática delituosa. (NOLASCO, SILVA, 2022, p.29)

Corroborando com as ideias de Nolasco e Silva (2023), Santos e Araújo (2023) mencionam que uma forma de atenuar os riscos acerca do uso da internet é por meio da educação em relação à segurança digital. Ações simples como criação de senhas fortes, não clicar em links de origem suspeita e atualizar os dispositivos com frequência são exemplos de boas práticas de uso da internet (SANTOS, ARAÚJO, 2023). Os autores também apontam que “a proteção dos dados pessoais também deve ser uma preocupação constante, seja através do uso de ferramentas de criptografia ou da adoção de políticas internas de proteção de dados” (SANTOS, ARAÚJO, 2023, p. 5). Ainda de acordo com os autores:

O aumento de ataques cibernéticos, impulsionado pelo crescimento exponencial do número de dispositivos conectados à internet, representa uma ameaça constante aos sistemas de segurança. Além disso, o avanço das tecnologias de hacking tem permitido que os criminosos virtuais desenvolvam métodos cada vez mais sofisticados para invadir sistemas e roubar informações sensíveis. Outro desafio é a falta de conscientização por parte dos usuários, que muitas vezes não adotam medidas básicas de segurança online. (SANTOS, ARAÚJO, 2023, p.21)

Na visão de Hoepers (2024), os fatores humanos deveriam estar presentes nas fases de um projeto e desenvolvimento de tecnologias, pois, para que sejam usadas por pessoas não especialistas, precisam ser compostas de boa usabilidade e acessibilidade. A autora aponta, porém, que “a realidade atual está longe de ser ideal, pois infelizmente a maior parte dos sistemas é complexa para a maior parte das pessoas, o que se reflete nesse baixo índice de habilidades por parte dos cidadãos” (HOEPERS, 2024, p. 63). Nesse sentido, a autora acrescenta:

Infelizmente, o baixo letramento digital da população brasileira é uma realidade que torna mais difícil o uso seguro da tecnologia. Os dados da TIC Domicílios mostram claramente que, apesar de 84% da população declarar que usa a Internet regularmente, menos da metade possui habilidades básicas como copiar e colar textos ou ativar configurações de segurança e privacidade (CETIC.BR, 2024, apud HOEPERS, 2024, p.63)

Para Hoepers (2024), os profissionais que atuarão na área de segurança e conscientização não precisarão se preocupar apenas com ensinar dicas de segurança, mas também o desenvolvimento do pensamento crítico do usuário. “Será necessário pensar em como proteger dados sensíveis de vazamentos via ferramentas de IA em nuvem, sejam elas chats, ferramentas de auxílio para programação (os “co-pilots”) ou quaisquer outras que surgirem.” (HOEPERS, 2024, p.65). Nesse mesmo contexto, a autora destaca a importância de preparar os usuários para lidar com essas novas ameaças:

Será necessário educar os usuários para esse cenário e esclarecer que os criminosos usaram os dados que eles mesmos colocam em redes sociais, pessoais ou de trabalho para treinar a IA, que poderá facilmente inferir relações de trabalho e de parentesco, e criar golpes personalizados sem interação dos fraudadores. E será necessário que os profissionais aprendam a usar a IA a seu favor, seja incrementando ferramentas de detecção ou automatizando tarefas como buscas por vulnerabilidades e análise de segurança código, por exemplo. (HOEPERS, 2024, p.65)

Cachioni et al. (2020, p. 47), ao citarem Building (2019), mencionam que “as Tecnologias de Informação e Comunicação (TIC) são capazes de transformar a realidade econômica e social das comunidades, tendo o potencial de mediar as aprendizagens nas atividades diárias, potencializando a participação ativa dos cidadãos na sociedade do conhecimento”. Além disso, segundo as autoras:

As TICs oferecem caminhos para compensar as dificuldades em relação ao tempo e à distância no contexto educacional, sendo facilmente adaptada para atender qualquer nível de aprendizagem do estudante. A realidade virtual permite interações on-line, por diferentes localizações geográficas, com transmissões de áudio, vídeo e textos. Os estudantes com alguma restrição física e/ou sensorial podem se beneficiar com o acesso às informações por meio das plataformas digitais. (CACHIONI et al., 2020, p.47)

Como demonstrado pelos trabalhos acima citados, existem uma série de lacunas no que diz respeito ao letramento digital da população, a usabilidade de ferramentas de segurança digital e o desenvolvimento tecnológico inclusivo. Dessa forma, o projeto em questão apresenta uma possível solução para superar essas barreiras e promover um acesso digital mais humanizado, seguro e integrado.

3. JUSTIFICATIVA

A crescente presença da tecnologia na vida cotidiana tem transformado profundamente a forma como as pessoas se comunicam, consomem e interagem com serviços bancários, governamentais e comerciais. Contudo, essa transformação também tem exposto milhões de usuários, em especial os menos familiarizados com práticas de segurança digital, a riscos cada vez maiores e frequentes no ambiente virtual. No Brasil, os índices de crimes cibernéticos são alarmantes fazendo com que 24% da população brasileira acima dos 16 anos sejam vítimas de golpes financeiros na internet (INSTITUTO DATASENADO, 2024). Segundo dados do Instituto Datafolha, mais de 4,5 mil pessoas sofreram tentativas de golpes financeiros por hora, sendo a maioria desses ataques realizados por meio de aplicativos de mensagens ou ligações fraudulentas (AGÊNCIA BRASIL, 2024).

Diante desse cenário preocupante, torna-se evidente a necessidade de promover a educação digital como uma ferramenta estratégica de prevenção. Muitos usuários, especialmente os leigos, desconhecem práticas básicas de proteção online, o que os torna alvos fáceis para criminosos. Além disso, a subnotificação dos crimes digitais, revelada pelo mesmo levantamento, com apenas 30% das vítimas formalizando ocorrência, indica não apenas um desamparo institucional, mas também um desconhecimento sobre como agir diante dessas situações.

Nesse contexto, justifica-se o desenvolvimento do aplicativo TatuPro, uma solução digital que visa oferecer conhecimento acessível sobre cibersegurança por meio de conteúdos educativos, dicas práticas e um sistema de chatbot para esclarecimento de dúvidas específicas. O aplicativo busca preencher a lacuna existente entre o avanço da tecnologia e a capacitação do usuário comum, promovendo uma cultura de prevenção e segurança no uso da internet. Ao combinar tecnologia, linguagem simples e recursos interativos, o TatuPro pretende contribuir para a construção de uma sociedade mais consciente, preparada e resiliente frente às ameaças digitais.

4. OBJETIVOS

4.1 Objetivo Geral

Desenvolver um aplicativo mobile educativo denominado TatuPro, com o intuito de promover a conscientização sobre cibersegurança entre usuários leigos, por meio de conteúdos interativos, dicas práticas e suporte personalizado, fornecido por um chatbot. O aplicativo visa capacitar os usuários de forma prática, ajudando-os a identificar e prevenir ameaças no ambiente digital.

4.2 Objetivos Específicos

- Identificar os principais tipos de golpes digitais que afetam usuários brasileiros, especialmente os mais recorrentes em plataformas de mensagens e comércio eletrônico;
- Elaborar conteúdos educativos acessíveis e didáticos sobre boas práticas de segurança digital, voltados a usuários com pouco conhecimento técnico;
- Desenvolver um chatbot integrado ao aplicativo, capaz de responder dúvidas frequentes e orientar os usuários em situações de risco digital de forma rápida e personalizada;
- Propor uma interface intuitiva e responsiva, adequada à experiência de diferentes perfis de usuários;
- Validar o protótipo do aplicativo por meio de testes com usuários reais, coletando feedback para futuras melhorias;
- Contribuir para a construção de uma cultura de prevenção digital no Brasil, alinhada aos princípios de cidadania digital e proteção de dados pessoais.

5. MATERIAIS E MÉTODOS

Esta seção do trabalho apresenta todos os materiais e métodos que foram utilizados no desenvolvimento do aplicativo TatuPro, detalhando as tecnologias utilizadas desde o levantamento de requisitos, passando pelos requisitos a serem atendidos com a prototipagem e sua implementação, bem como toda a pesquisa feita antes do desenvolvimento efetivo do aplicativo, o que envolve benchmark, criação de personas e descoberta do público-alvo,

listagem das funcionalidades do aplicativo, seus requisitos funcionais e não funcionais e os resultados do teste de usabilidade, tudo o que fundamenta a construção do sistema.

A seguir, são apresentados os principais módulos do sistema, a arquitetura adotada, os recursos implementados e os testes realizados para garantir a qualidade da aplicação.

5.1. Benchmarking

Após concebida a ideia inicial do projeto, foram feitas pesquisas referentes a aplicativos já existentes na área de estudo, aqueles que são referências no mercado, já que em um processo de desenvolvimento de produto ou serviço é preciso visualizar como os produtos existentes estão solucionando os problemas dos usuários. Isso é possível através do benchmarking: estudo da concorrência.

Durante a pesquisa, foram encontrados diversos aplicativos voltados a cibersegurança, abaixo está um recorte com três desses aplicativos:

Tabela 1 – Benchmark de aplicativos similares.

HackerX	Cibersegurança e Antivírus	Learn Cyber Security
Aprender noções básicas e avançadas de segurança cibernética e hackers	Proteger seu dispositivo com recursos projetados para mantê-lo seguro: Proteção contra Malware e Verificação de Violação de Email	Aprender sobre segurança cibernética e como os sistemas de segurança online.

Fonte: Elaborado pelos autores.

Os aplicativos sobre cibersegurança encontrados na pesquisa resolviam o problema do público-alvo de maneira paliativa, focados em proteger o dispositivo ou ensinar profissionais sobre cibersegurança. O que significa que há espaço para um novo aplicativo, agora focado em ensino de cibersegurança para o público não especializado. Abaixo mais detalhes sobre as principais características dos aplicativos encontrados.

5.1.1. HackerX - Learn Ethical Hacking

O aplicativo Learn Ethical Hacking - Ethical Hacking Tutorials é voltado para quem deseja iniciar uma carreira como hacker ético. Ensina desde os conceitos básicos até habilidades avançadas de cibersegurança e hacking. O foco é formar hackers éticos: profissionais que identificam falhas em sistemas e redes com o consentimento dos proprietários, ajudando a reforçar a segurança contra ataques maliciosos.

5.1.2. Cibersegurança e Antivírus

O aplicativo Cibersegurança e Antivírus tem o objetivo de fornecer proteção completa contra ameaças cibernéticas em dispositivos móveis, como vírus, malware, spyware e vazamentos de dados. Também visa proteger os dados pessoais dos usuários e fortalecer sua segurança digital. O aplicativo promove uma vida digital mais segura ao identificar vulnerabilidades, emitir alertas e sugerir melhorias de segurança.

5.1.3. Learn Cyber Security

O aplicativo Learn Cyber Security é focado em fornecer conhecimento a pessoas que desejam iniciar ou fortalecer a carreira na área de cibersegurança. Ele oferece tutoriais e lições que abrangem desde conceitos básicos até conteúdos mais avançados. O Aplicativo busca capacitar os usuários a entenderem e protegerem sistemas de computadores, com o objetivo de construir uma carreira sólida na área de segurança digital.

5.2. Personas e Público-alvo

O público-alvo é a definição ampla das pessoas que um produto ou serviço pretende atingir. Ele é descrito com base em características demográficas, geográficas, comportamentais e socioeconômicas. Já as personas são representações fictícias e detalhadas dos usuários ideais do sistema, baseadas em dados reais.

O público-alvo do aplicativo Tatupro são pessoas que têm pouco tempo, mas utilizam meios tecnológicos em diversos âmbitos da vida como: trabalho, entretenimento ou acadêmico. E que portanto podem ser atingidas por golpes cibernéticos.

A partir da ideia de um público-alvo foram criadas as seguintes personas:

Figura 1 – Persona Josué Costa.

Josué Costa - Estudante de TI e Operador de Telemarketing

Idade: 25 anos

Localização: São Roque/SP

Renda: R\$1.800,00



Com a correria do dia a dia, preciso de um jeito rápido e prático de aprender sem perder a qualidade do conteúdo.

- **Motivações:** Seu interesse por conhecer conceitos de ciber segurança começou com o trabalho, ao escutar reclamações dos cliente sobre golpes, e aumentou ao começar a estudar Tecnologia da Informação. Quer se aprofundar nesses âmbitos para aplicar no trabalho e conseguir novas oportunidades.
- **Rotina:** Sua rotina é corrida, portanto tem pouco tempo livre. Trabalha home office alguns dias da semana, utilizando artifícios tecnológicos tanto no trabalho como no seu dia a dia.
- **Comportamento:** Busca aprendizado rápido e fácil de entender. Costuma usar aplicativos, podcasts e vídeos como mecanismos de aprendizagem.
- **Dores:** Cansaço por conciliar trabalho e estudos. Dificuldade em encontrar conteúdo prático sobre ciber segurança, ou conteúdo didático e acessível.

Fonte: Elaborado pelos autores.

Figura 2 – Persona Tatiane Santos.

Tatiane Santos - Empreendedora da beleza

Idade: 44 anos

Localização: Rio Janeiro/RJ

Renda: R\$5.000,00



Hoje em dia, a gente faz tudo pelo celular... só queria entender como usar sem cair em cilada.

- **Motivações:** Tatiane usa o WhatsApp e o Facebook para divulgar seus serviços, marcar horários e conversar com as clientes. Mesmo estando sempre conectada, ela sente necessidade de aprender mais sobre como se proteger de golpes e fraudes, e quer usar a tecnologia de forma segura para fortalecer o seu negócio.
- **Rotina:** Ela tem uma rotina bem movimentada, dividida entre os atendimentos no seu próprio espaço, visitas a domicílio e a organização da vida pessoal.
- **Comportamento:** Costuma registrar os resultados dos atendimentos para postar nas redes sociais, acompanha vídeos e áudios com dicas e tendências da área da beleza, e participa de grupos no WhatsApp e páginas no Facebook sobre o tema.
- **Dores:** Tatiane tem medo de cair em golpes que possam prejudicar o relacionamento com suas clientes. Além disso, sente insegurança ao lidar com senhas, links duvidosos e configurações de privacidade nos aplicativos que usa no dia a dia.

Fonte: Elaborado pelos autores.

Figura 3 – Persona Maria Ferreira.

Maria Ferreira - Enfermeira Aposentada

Idade: 66 anos

Localização: Limoeiro de Anadia/AL

Renda: R\$3.500,00



• **Motivações:** Maria usa o WhatsApp e o Facebook para se manter próxima da família e socializar. Apesar de estar conectada, quer aprender a se proteger melhor online, pois tem pouco conhecimento sobre segurança digital.

• **Rotina:** Sua rotina é tranquila e gira em torno de atividades domésticas, lazer e participação em grupos virtuais. Ela se dedica ao tricô, assiste a vídeos no celular e conversa com amigos em grupos de redes sociais.

• **Comportamento:** Maria participa ativamente de grupos no WhatsApp e Facebook relacionados a hobbies, como tricô. Gosta de compartilhar conteúdos e se manter atualizada, mas costuma clicar em links e mensagens sem avaliar os riscos. Não usa aplicativos bancários e evita compras online por medo de cair em fraudes.

• **Dores:** Tem medo de cair em golpes e fraudes virtuais, principalmente em redes sociais e grupos. Sente insegurança ao lidar com senhas, links suspeitos e configurações de privacidade. Por não compreender bem os riscos, acaba ficando vulnerável e depende da ajuda de familiares para resolver problemas com o celular.

Fonte: Elaborado pelos autores.

5.3. Requisitos

Os requisitos do sistema se relacionam com tudo que o sistema deve fazer, ou seja, funcionalidades e comportamentos.

5.3.1. Requisitos Funcionais

Tabela 2 – Requisitos Funcionais.

Requisito funcional	Descrição
Login do usuário	O sistema deve permitir o login do usuário
Cadastro do usuário	O sistema deve permitir o cadastro de novos usuários com e-mail e senha.
Editar perfil de usuário	O sistema deve permitir a edição do perfil do usuário.
Dicas e lições sobre cibersegurança	O sistema deve fornecer dicas e lições sobre cibersegurança.
Chatbot.	O sistema deve fornecer um chatbot para o usuário tirar dúvidas.

Fonte: Elaborado pelos autores.

5.3.2. Requisitos Não Funcionais

Tabela 3 – Requisitos não funcionais.

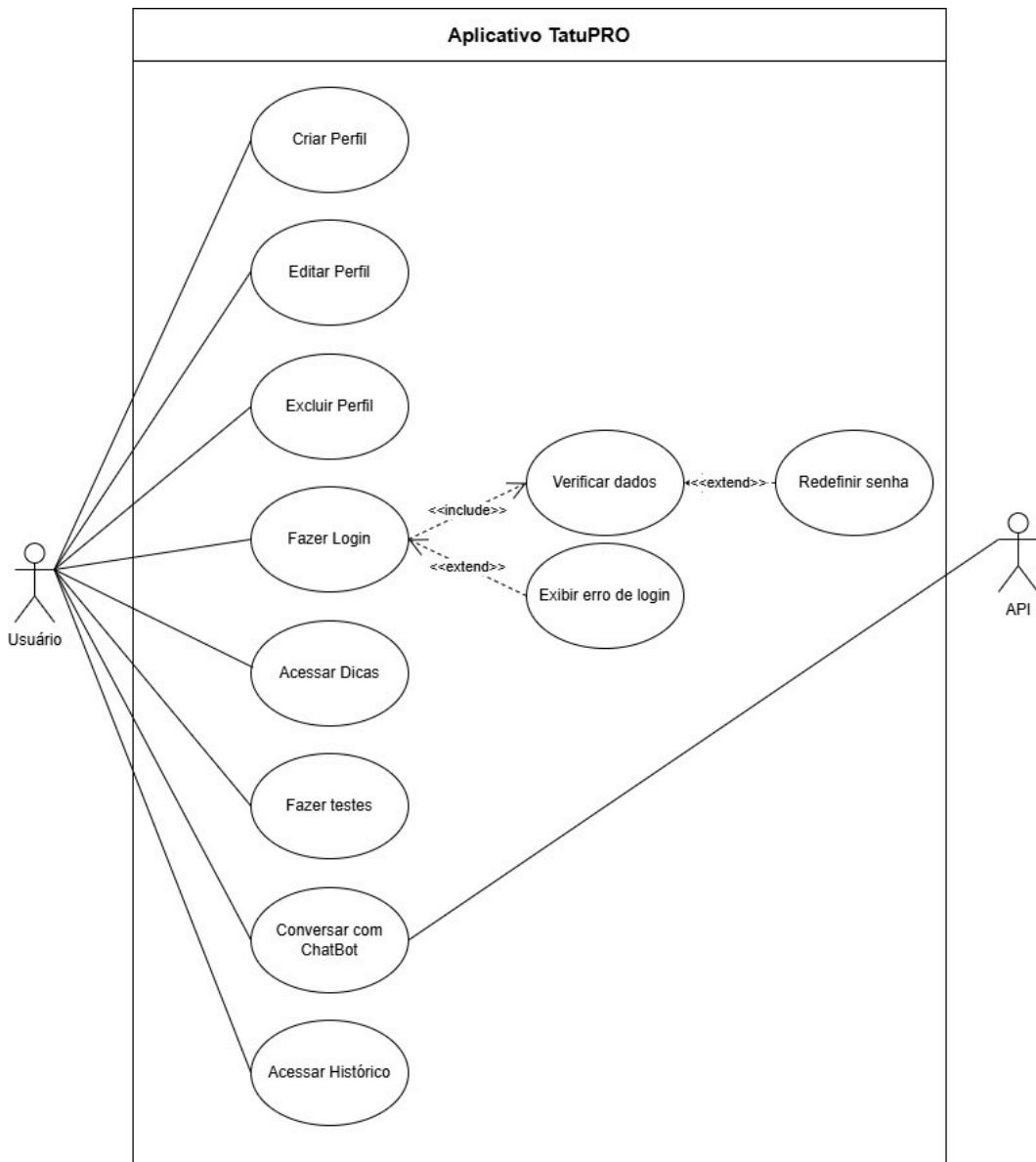
Requisito Não Funcional	Descrição
Usabilidade	A interface do aplicativo deve ser intuitiva e fácil de usar.
Interface amigável	O design da aplicação deve ser visualmente agradável e acessível.
Disponibilidade contínua	O sistema deve estar disponível para uso 24 horas por dia, 7 dias por semana.
Confiabilidade e segurança	O sistema deve funcionar de forma confiável e proteger os dados do usuário contra ameaças.
Escalabilidade	O sistema deve ser capaz de lidar com aumento de usuários e dados sem perda de desempenho.
Proteção de dados	Nenhum dado privado do usuário deve ser exposto ou compartilhado sem consentimento.
Conformidade legal	O sistema deve seguir as leis de proteção de dados vigentes, como a LGPD (Lei Geral de Proteção de Dados).

Fonte: Elaborado pelos autores.

5.4. Diagrama de Casos de Uso

Entre os diagramas da UML (Linguagem de Modelagem Unificada) existe o diagrama de casos de uso. O diagrama de casos de uso tem por objetivo apresentar as principais funcionalidades do sistema, resumindo os detalhes dos usuários do seu sistema (também conhecidos como atores) e as interações deles com o sistema, também servindo como forma de comunicação entre o desenvolvedor e uma pessoa não especialista, já que sua estrutura é simples de compreender, não precisando de um nível alto de conhecimento técnico. Na imagem abaixo está o Diagrama de Casos de Uso que representa o sistema Tatupro:

Figura 4 – Diagrama de Casos de Uso.

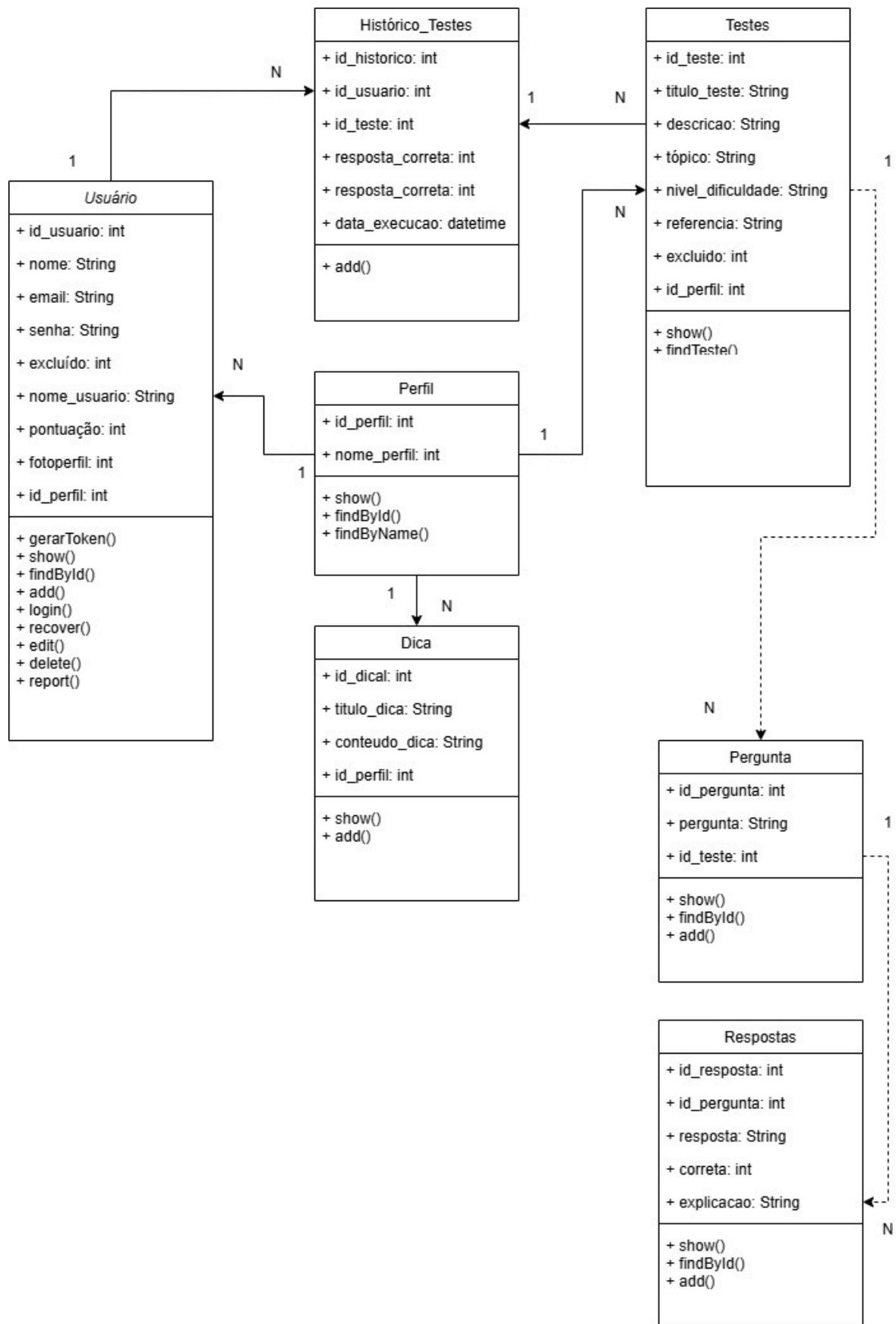


Fonte: Elaborado pelos autores.

5.5. Diagrama de Classes

O diagrama de classes é uma representação visual da estrutura estática do sistema, ilustrando as classes que compõem a aplicação, seus atributos, métodos e os relacionamentos entre elas. No contexto do TatuPro, este diagrama documenta as principais entidades do sistema, como Usuário, Teste, Pergunta, Resposta, Dica e Perfil, evidenciando como essas classes se relacionam através de associações, agregações e composições. A modelagem orientada a objetos apresentada no diagrama serve como base para a implementação tanto do backend quanto do frontend, garantindo que a estrutura de dados seja consistente em todas as camadas da aplicação e facilitando a manutenção e evolução do sistema.

Figura 5 – Diagrama de Classes.

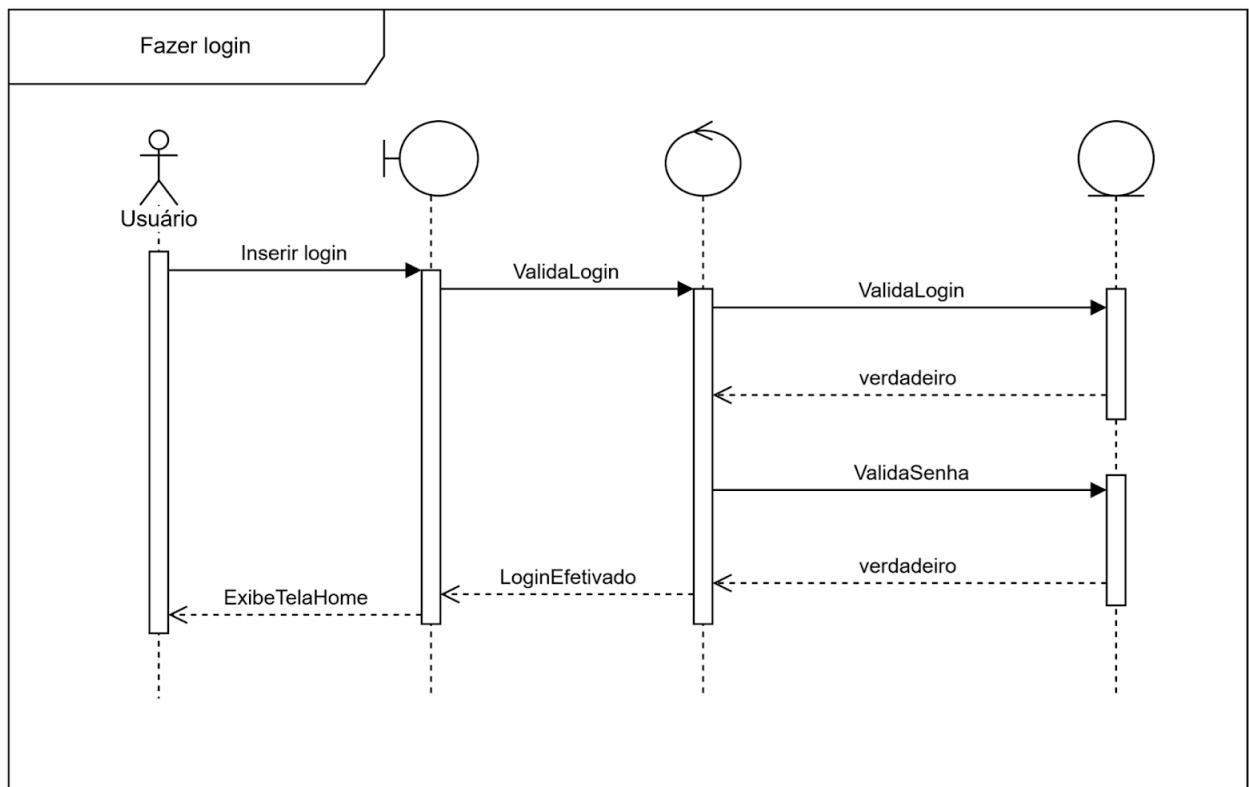


Fonte: Elaborado pelos autores.

5.6. Diagrama de Sequência

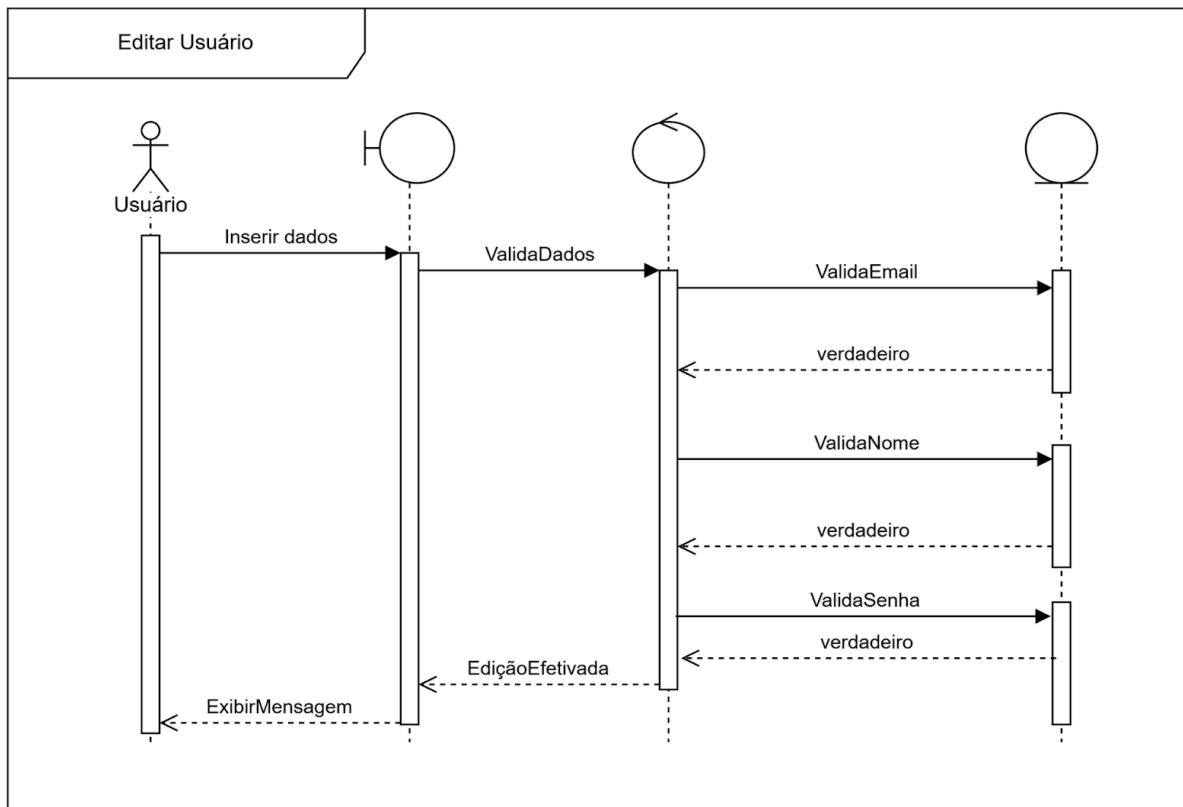
O diagrama de sequência representa a interação temporal entre os componentes do sistema durante a execução de funcionalidades específicas, mostrando a ordem das mensagens trocadas entre objetos ao longo do tempo. Para o TatuPro, foram modelados os principais fluxos de uso, como autenticação de usuário, realização de testes, consulta de dicas e interação com o chatbot, evidenciando como o frontend se comunica com a API (*Application Programming Interface*) RESTful (*Representational State Transfer*), que por sua vez interage com o banco de dados e, quando necessário, com serviços externos como a API Gemini. Esta documentação é fundamental para compreender o comportamento dinâmico da aplicação, auxiliando no desenvolvimento, debug e na identificação de possíveis gargalos de desempenho ou falhas na comunicação entre os componentes do sistema.

Figura 6 – Diagrama de sequência - Login.



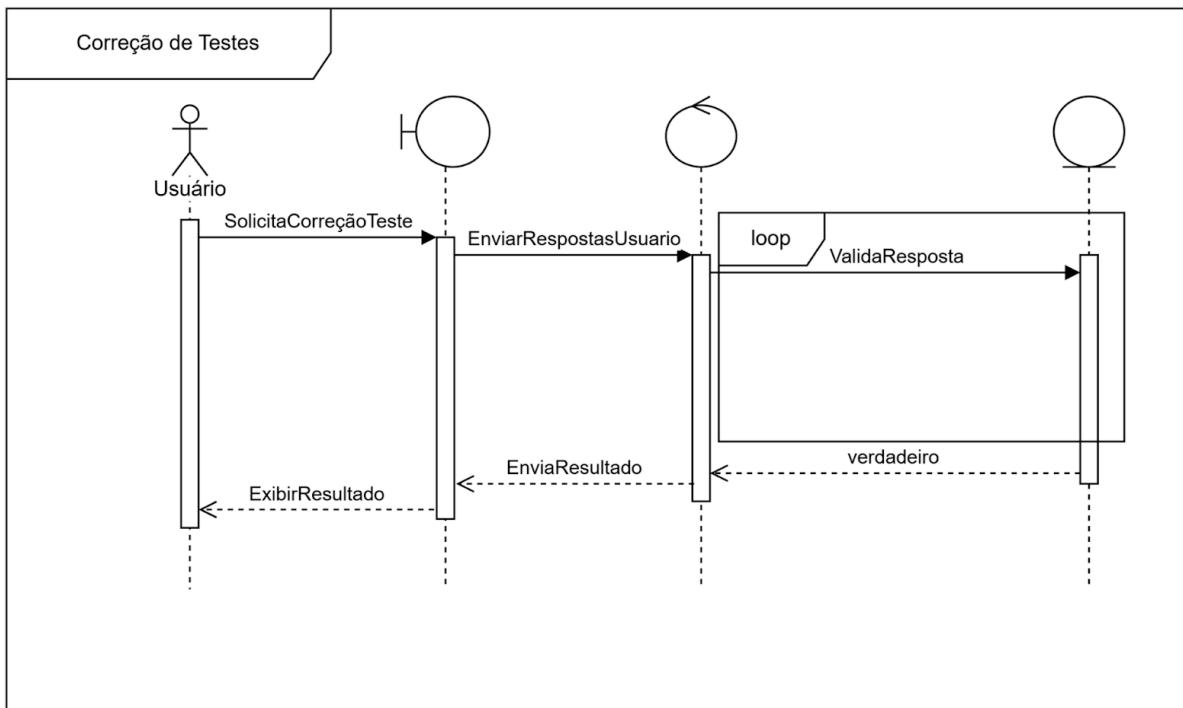
Fonte: Elaborado pelos autores.

Figura 7 – Diagrama de sequência - Edição de Usuário.



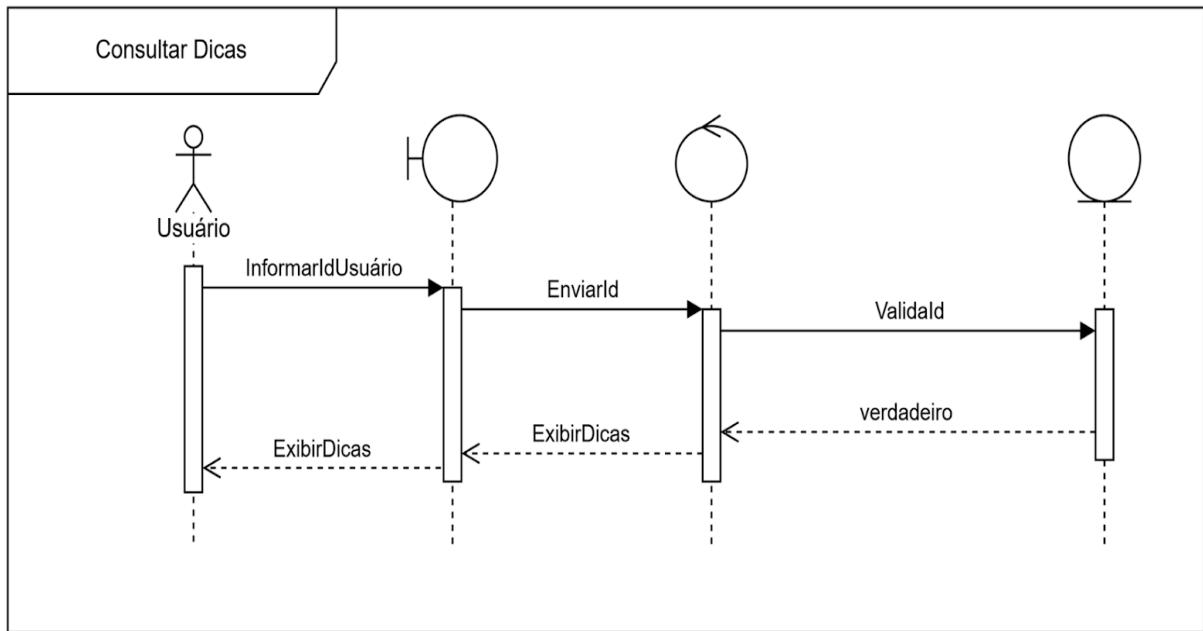
Fonte: Elaborado pelos autores.

Figura 8 – Diagrama de sequência - Correção de Testes.



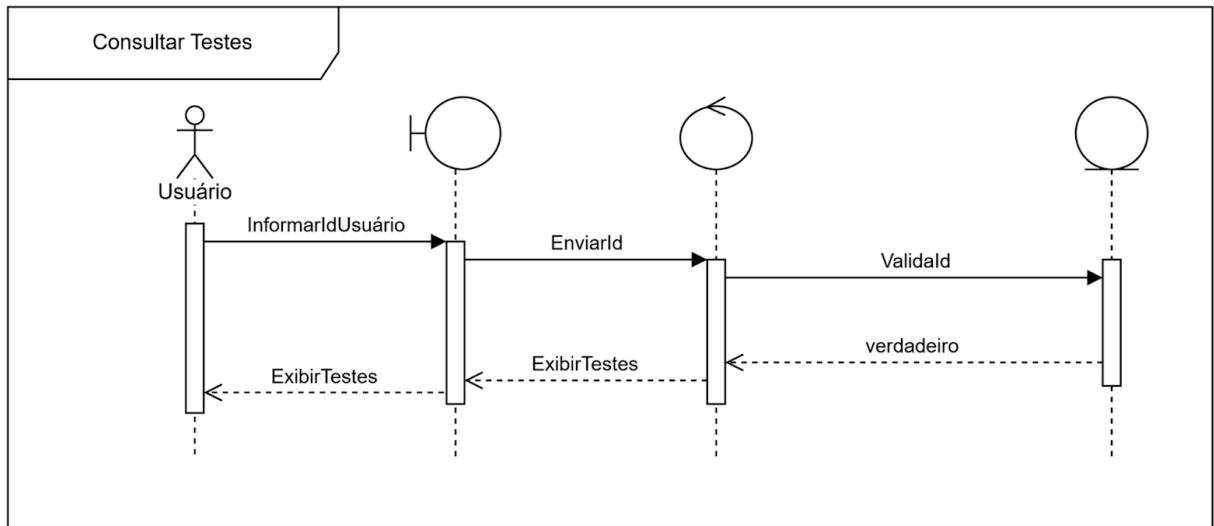
Fonte: Elaborado pelos autores.

Figura 9 – Diagrama de sequência - Consultar Dicas.



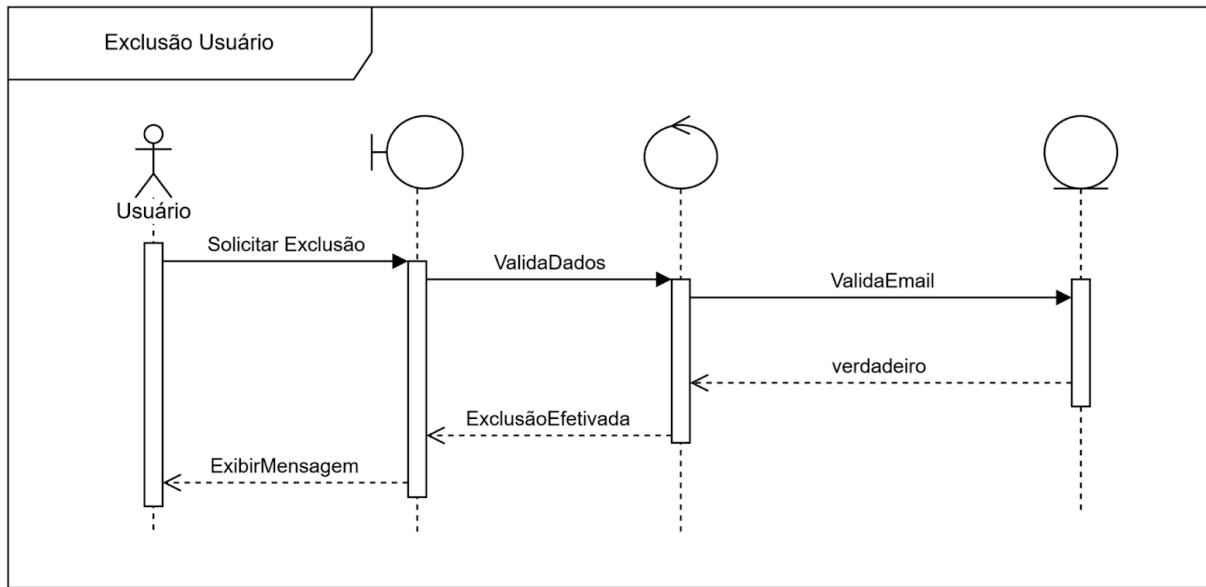
Fonte: Elaborado pelos autores.

Figura 10 – Diagrama de sequência - Consultar Testes.



Fonte: Elaborado pelos autores.

Figura 11 – Diagrama de sequência - Exclusão de Usuário.

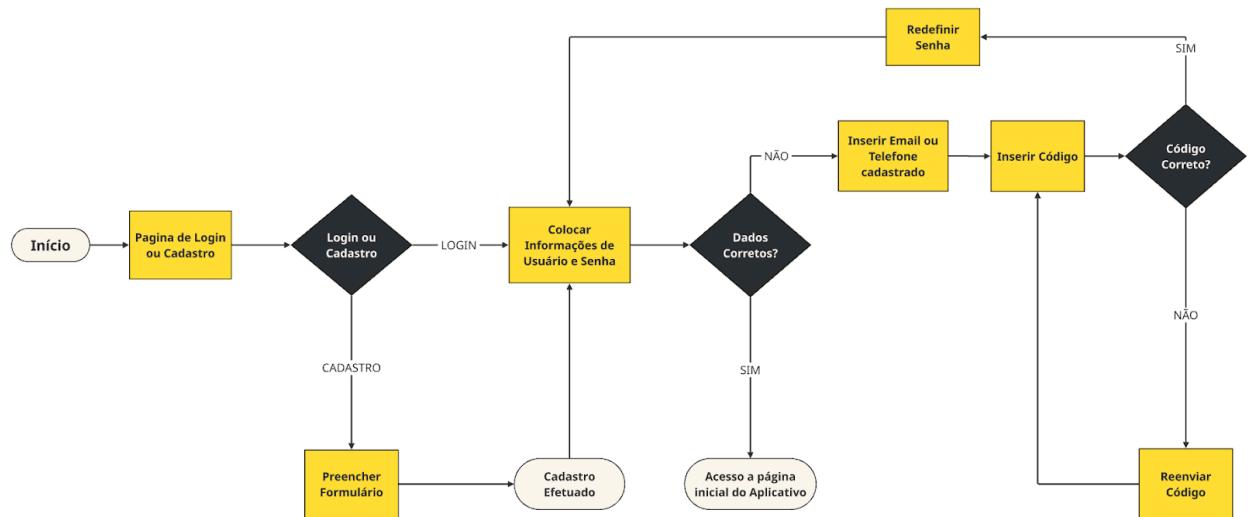


Fonte: Elaborado pelos autores.

5.7. Fluxo de Navegação e Interação

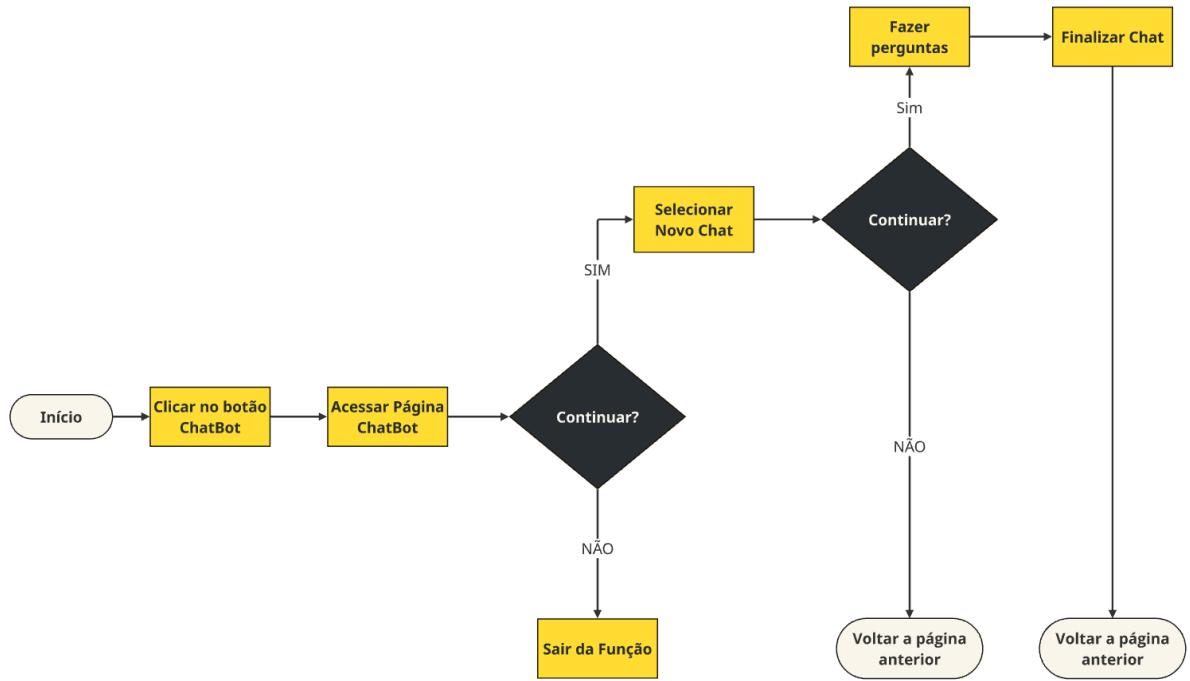
Fluxo de Navegação de interação é um conceito que se relaciona com a interação do usuário com um site ou aplicativo, ou seja, os caminhos que um usuário pode fazer dentro do sistema. Abaixo estão os fluxogramas do aplicativo TatuPro:

Figura 12 – Fluxo de Navegação - Login.



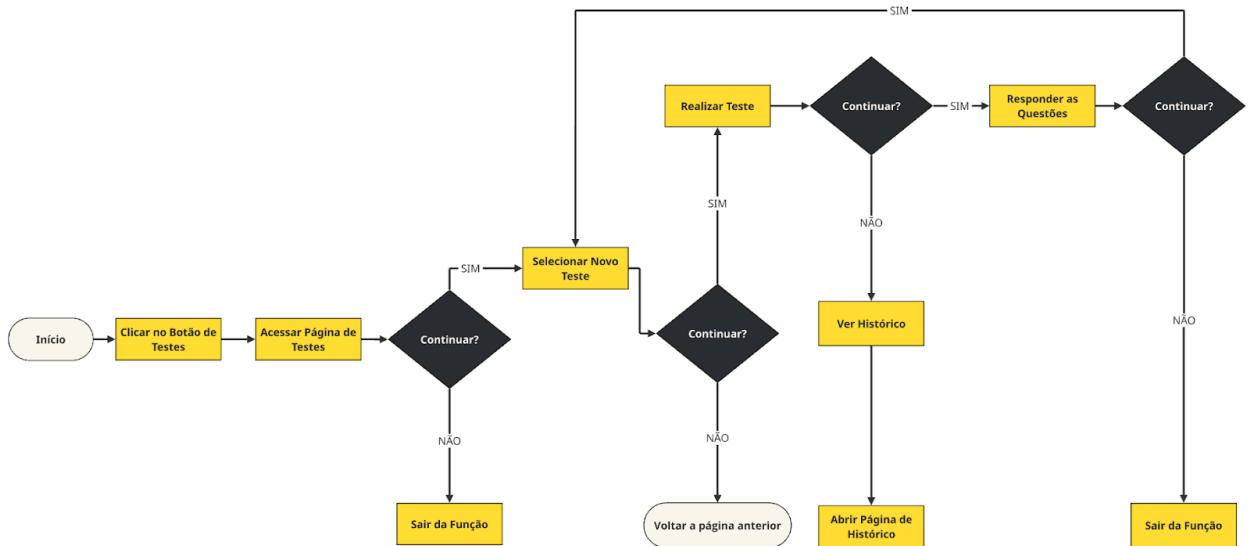
Fonte: Elaborado pelos autores.

Figura 13 – Fluxo de Navegação - Chatbot.



Fonte: Elaborado pelos autores.

Figura 14 – Fluxo de Navegação - Testes.

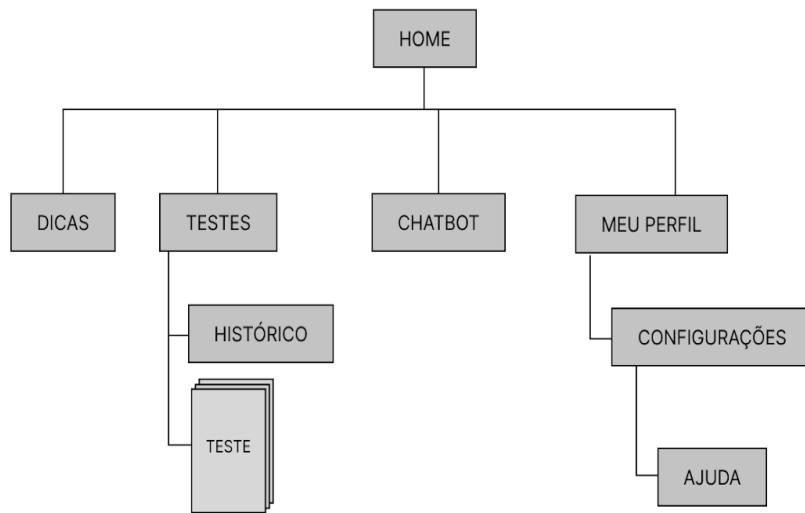


Fonte: Elaborado pelos autores.

5.8. Arquitetura da Informação

A Arquitetura da Informação tem como objetivo organizar o site, quais as páginas existirão e de que maneira elas se interligam, auxiliando o usuário a localizar-se dentro do aplicativo.

Figura 15 – Arquitetura da Informação.



Fonte: Elaborado pelos autores.

5.9. Teste de Usabilidade

O teste de usabilidade é uma técnica essencial no desenvolvimento de produtos digitais, utilizada para avaliar a facilidade de uso e a experiência do usuário ao interagir com uma interface. Ele permite identificar problemas de navegação, dificuldades na execução de tarefas e oportunidades de melhoria, garantindo que o produto atenda às expectativas do público-alvo. Ao observar usuários reais interagindo com o protótipo, foi possível obter insights valiosos que contribuem para um design mais intuitivo e eficiente, reduzindo retrabalho e aumentando a satisfação do usuário.

O teste teve como objetivo avaliar a usabilidade do aplicativo TatuPro, buscando identificar problemas e avaliar sua eficiência e usabilidade. Este relatório apresenta os principais insights retirados das respostas obtidas durante o teste, identificando os principais desafios e problemas do usuário ao interagir com o primeiro protótipo do sistema, desta forma obtendo ideias valiosas para a melhorar a usabilidade e interatividade do aplicativo.

5.9.1. Metodologia usada no teste

Para avaliar a experiência de uso do aplicativo TatuPro, foi conduzido um teste de usabilidade com 27 participantes representativos do público-alvo. Foi desenvolvido um protótipo do aplicativo no Figma, que possibilita uma simulação do que seria o produto final, ao acessar o link do formulário o usuário foi convidado a testar o protótipo e após a análise, responder o formulário para avaliar a experiência obtida no uso do aplicativo.

As respostas que foram analisadas abaixo, visam identificar dificuldades, analisar a eficiência da interface, facilidade em encontrar o objetivo do aplicativo pela navegação e propor melhorias para otimizar a experiência do usuário, auxiliando para o desenvolvimento de uma navegação fluida e eficaz.

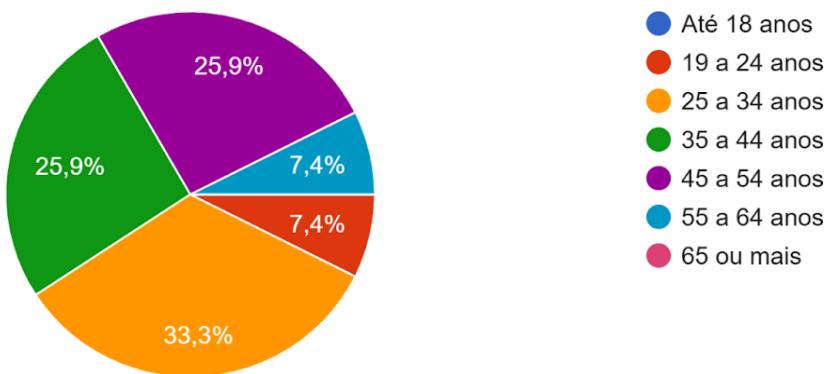
5.9.2. Perfil dos participantes

Abaixo estão os gráficos que representam a amostra da população que respondeu ao teste (Figura 16, Figura 17 e Figura 18).

Figura 16 – Faixa Etária dos Participantes.

Seleciona sua faixa etária

27 respostas



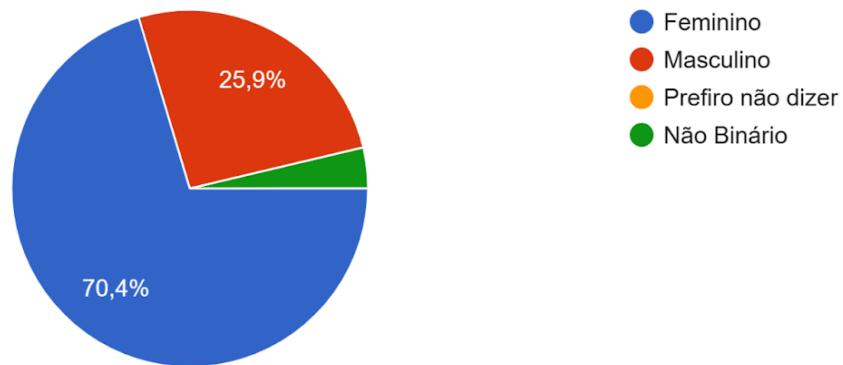
Fonte: Elaborado pelos autores.

Como podemos observar na figura acima, os usuários que nos ajudaram a validar a usabilidade do aplicativo, se concentram em sua maioria na faixa etária dos 25 aos 34 anos, totalizando 33,3% das respostas, entretanto temos uma quantidade equilibrada com as faixas etárias de 35 a 44 anos e também 45 a 54 anos, mostrando uma diversidade com relação à idade dos participantes.

Figura 17 – Gênero dos Participantes.

Qual o seu gênero?

27 respostas



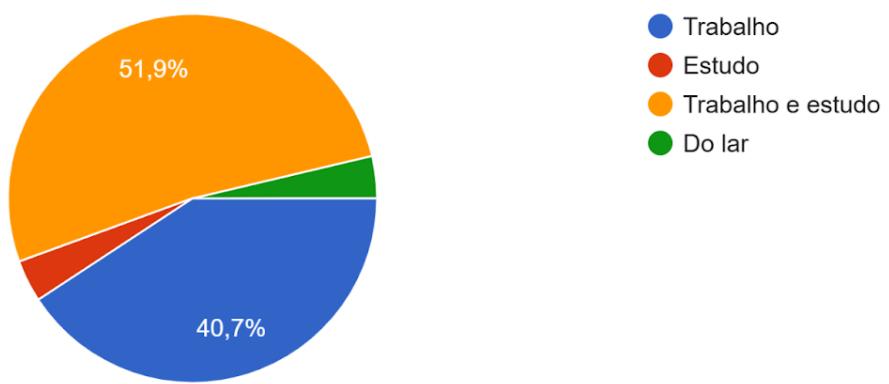
Fonte: Elaborado pelos autores.

A maioria das respostas foram provenientes do público feminino, totalizando 70,4% do total de respostas da nossa pesquisa (Figura 17).

Figura 18 – Ocupação dos participantes.

Qual a sua ocupação?

27 respostas



Fonte: Elaborado pelos autores.

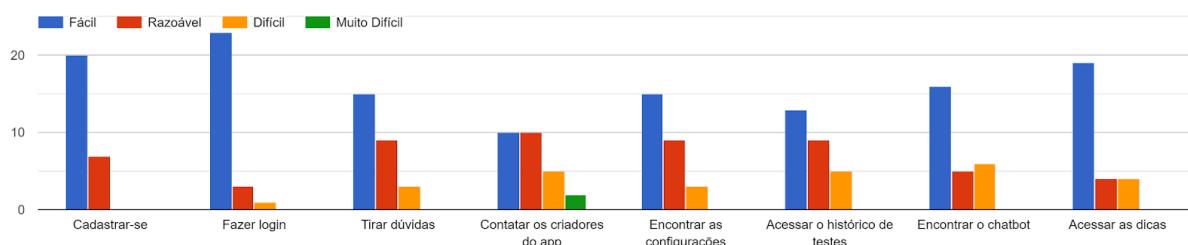
Referente a ocupação dos nossos usuários, a maioria das pessoas que ajudaram nas respostas, trabalha e estuda (Figura 18). A aplicação do teste foi realizada entre os dias 24 de março e 14 de abril do ano de 2025 conduzido remotamente.

5.9.3. Problemas Identificados

O gráfico abaixo mostra o nível de dificuldade para cada funcionalidade específica do aplicativo. Nota-se que os usuários tiveram uma certa dificuldade para algumas ações. Essa dificuldade (levando em consideração as ações com avaliações acima de razoável) sugere a necessidade de melhorias no visual do aplicativo, disposição de alguns botões e organização dos fluxos de navegação (Figura 19).

Figura 19 – Dificuldades encontradas pelos participantes.

Selecione o nível de dificuldade (a direita) para cada ação (a esquerda).



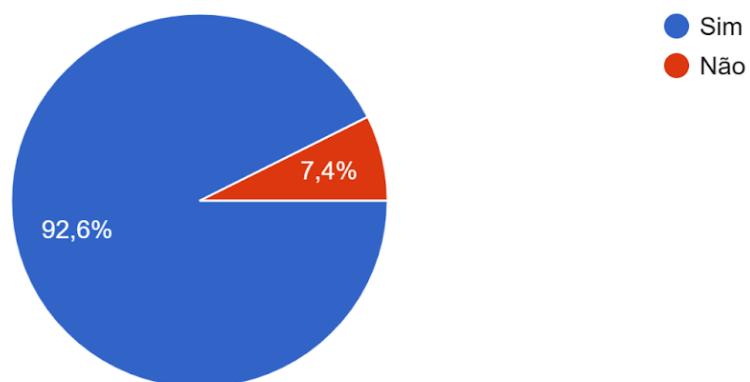
Fonte: Elaborado pelos autores.

O gráfico da figura 20 apresenta o entendimento do público sobre o objetivo do aplicativo. Obtivemos 92,6% de resposta positiva, no entanto 7,4% foram respostas negativas quanto a compreensão do intuito do aplicativo, o que sugere a mudança da identidade visual do aplicativo.

Figura 20 – Entendimento sobre o objetivo do aplicativo.

Foi possível entender o objetivo do aplicativo?

27 respostas



Fonte: Elaborado pelos autores.

As figuras 21 e 22 permitem a visualização de algumas respostas sobre a experiência do usuário. Com base nas respostas apresentadas, retiramos duas que sugerem as seguintes mudanças: revisão ortográfica, término do fluxo de perguntas e transições entre as telas.

Figura 21 – Avaliação geral da experiência do usuário.

Qual a sua avaliação geral sobre a experiência do usuário?

27 respostas

O app possui outras "pastas" para tirar dúvidas, mas quando clico nelas, não acontece nada. O teste apresenta ter 10 questões, mas só aparece a primeira. Quando clico em "enviar", o teste é finalizado e não consigo visualizar as outras questões. Como sou leiga no assunto, não consegui entender muito bem o funcionamento, mas espero ter ajudado!

As funções do app são muito claras e os botões são de boas visualização
Poderia colocar transições entre as telas ao clicar nos botões
No menu hambúrguer poderia ter os botões testes, dicas, chatbot e sair
Não apareceu a barra de rolagem, não sei se é necessário configurar
Sugiro revisão ortográfica: Tela 1 tutorial: utilizaR; tela tutorial leia dicas: repete verbo ficar; tela tutorial chatbot: frase "pergunte sem e medo"
Tutorial completo: verificar estrutura dos textos e tom que pretende usar

Fonte: Elaborado pelos autores.

Figura 22 – Pontos positivos e negativos.

Quais os pontos positivos e negativos do aplicativo?

27 respostas

Passa as informações com clareza, porém, como falei anteriormente, algumas "pastas" para tirar dúvidas não respondem ao clique. O teste apresenta ter 10 questões, mas só aparece uma e ao clicar em "enviar", o teste é finalizado sem mostrar as outras. Isso também acontece nas dicas, tem várias dicas, mas apenas a primeira abre ao clicar, já o restante não...

Fonte: Elaborado pelos autores.

A figura 23 referente a pergunta sobre pontos negativos e positivos do aplicativo também sugere mudança no visual do aplicativo, mais especificamente nos botões.

Figura 23 – Pontos positivos e negativos - cores do aplicativo.

Quais os pontos positivos e negativos do aplicativo?

27 respostas

O aplicativo em si é ótimo, com cores satisfatórias e com um tutorial fácil e rápido.

Para ter algo negativo, somente o botão de criar conta que a cor poderia um pouquinho melhor

Fonte: Elaborado pelos autores.

As figuras 24 e 25 também se referem à pergunta sobre os pontos negativos do sistema sugerem melhorias no visual do aplicativo para deixar seu objetivo mais claro e organização dos fluxos de navegação.

Figura 24 – Pontos positivos e negativos - dificuldades.

Quais os pontos positivos e negativos do aplicativo?

27 respostas

A parte do chat bot é um pouco mais difícil de localizar

Fonte: Elaborado pelos autores.

Figura 25 – Pontos positivos e negativos - mudança geral.

Boa naveabilidade, fácil compreensão. Talvez o tema utilizado em um geral (imagem, formato, cores) não me passam a mensagem de qual é o objetivo do aplicativo, além de público alvo e benefícios (apesar de já saber), eu procuraria uma maneira de deixar isso mais claro.

Fonte: Elaborado pelos autores.

Foi feita uma análise da coluna sobre “Quem entre seus conhecidos gostaria desse produto?”, utilizando uma tabela dinâmica. É possível analisar a partir da figura 26 que das 21 respostas a essa pergunta, apenas 5 não souberam responder ou não conheciam alguém que usaria o aplicativo. Chegando a conclusão de que o aplicativo TatuPro teria um público que o utilizaria.

Figura 26 – Possível recomendação do aplicativo.

Quem entre seus conhecidos gostaria desse produto?	Qtde de respostas
Não sei	3
Minha filha	2
Turma da sala, amigos	1
Sim	1
Ninguém por enquanto	1
Não conheço	1
Não	1
Meus amigos e familiares	1
Fernanda Lima	1
Estudantes acadêmico	1
Empresa	1
Creio que meu sobrinho.	1
Amigos. Estudam ti	1
Amigos.	1
Amigos, gostam de informações sobre tecnologia....	1
amiga	1
alguns amigos da área	1
Ariane uma pessoa muito inteligente	1
Total geral	21

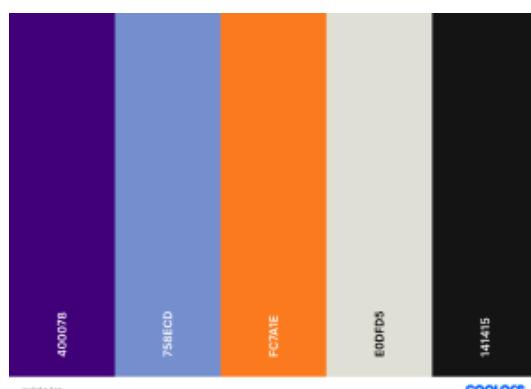
Fonte: Elaborado pelos autores.

5.9.4. Alterações Realizadas

5.9.4.1. Paleta de cores

Mudança na paleta de cores do aplicativo devido a falta de identificação do ponto de vista dos usuários sobre o objetivo do aplicativo. Trocamos o roxo que tínhamos como cor principal para um tom de roxo azulado, já que azul é comumente ligado a segurança. Com isso é esperado que o usuário entenda mais facilmente sobre o que é o aplicativo.

Figura 27 – Paleta de cores inicial.



Fonte: Elaborado pelos autores.

Figura 28 – Paleta de cores alterada.



Fonte: Elaborado pelos autores.

5.9.4.2. Reformulação dos botões de navegação

Reformulamos todos os botões de navegação já que houveram dificuldades para encontrar as funcionalidades principais do aplicativo. A *appbar* foi substituída por um modelo mais simples e clean, o botão de notificação continua no mesmo lugar, no entanto retiramos o menu hambúrguer.

Figura 29 – Appbar versão 1.



Fonte: Elaborado pelos autores.

Figura 30 – Appbar versão 2.



Fonte: Elaborado pelos autores.

Todos os botões antes encontrados no corpo do texto, agora estão no *bottom navigation* do aplicativo, demos destaque ao botão do chatbot já que no teste os usuários

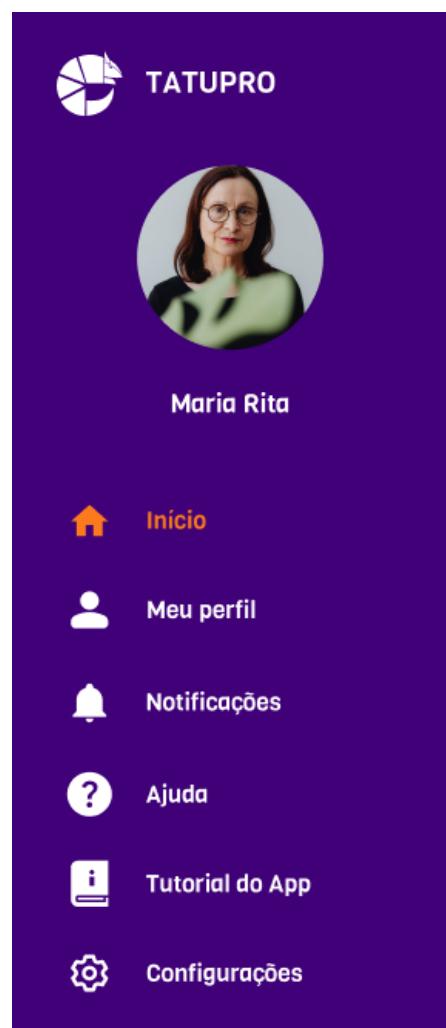
tiveram mais dificuldade para encontrá-lo. Além disso, todas as opções dispostas no menu hambúrguer podem ser acessadas pelo botão “Meu perfil” no *bottom navigation*.

Figura 31 – Botões na tela inicial do aplicativo.



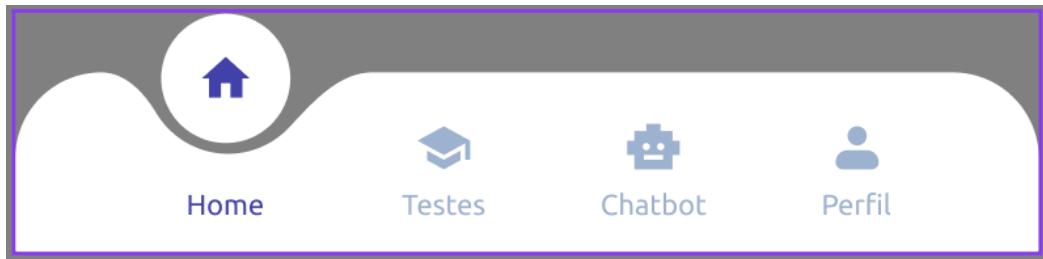
Fonte: Elaborado pelos autores.

Figura 32 – Menu de navegação lateral.



Fonte: Elaborado pelos autores.

Figura 33 – Bottom Navigation.

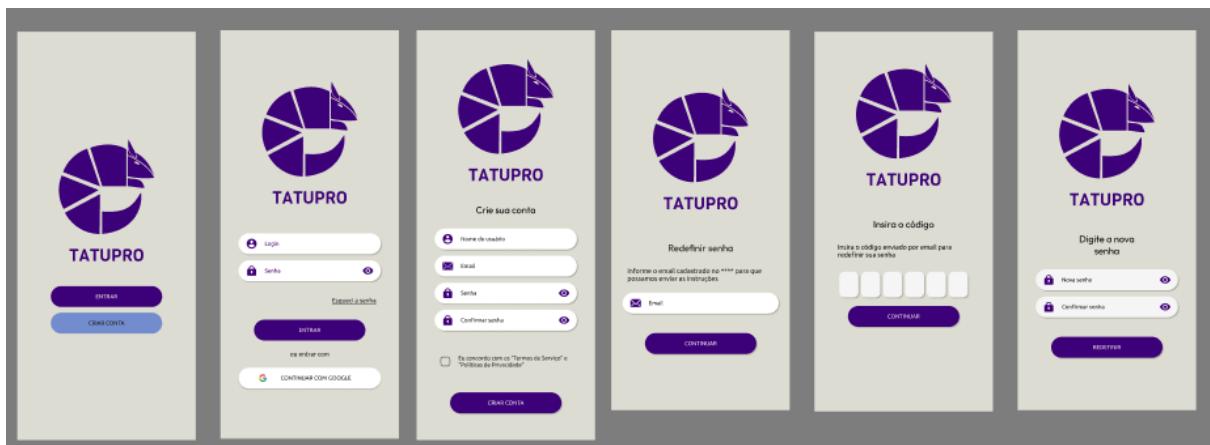


Fonte: Elaborado pelos autores.

5.9.4.3. Reorganização do layout das telas

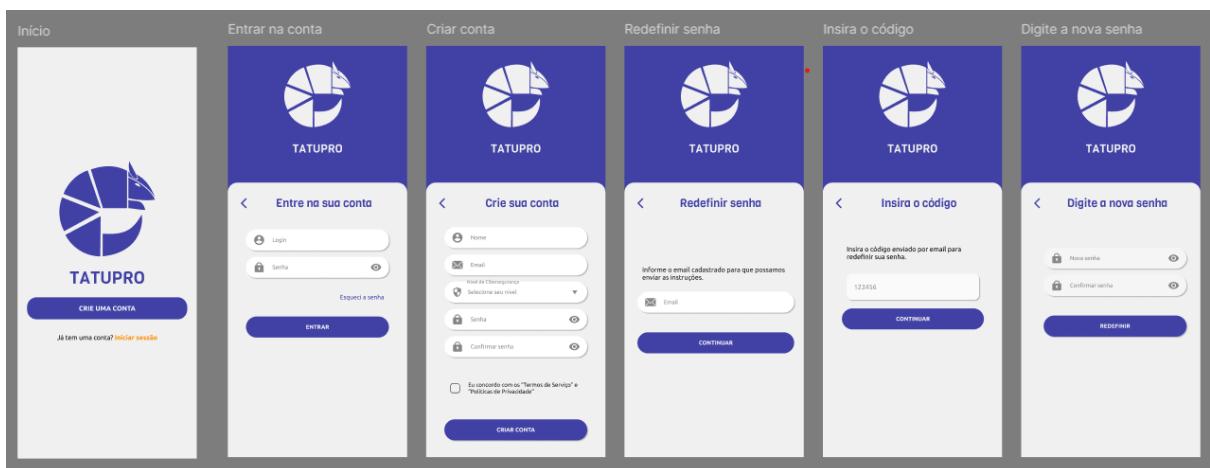
Alterações simples no fluxo de login visando melhor visualização dos campos de *input* e melhor fluxo de navegação.

Figura 34 – Telas de navegação - login versão 1.



Fonte: Elaborado pelos autores.

Figura 35 – Telas de navegação - login versão 2.



Fonte: Elaborado pelos autores.

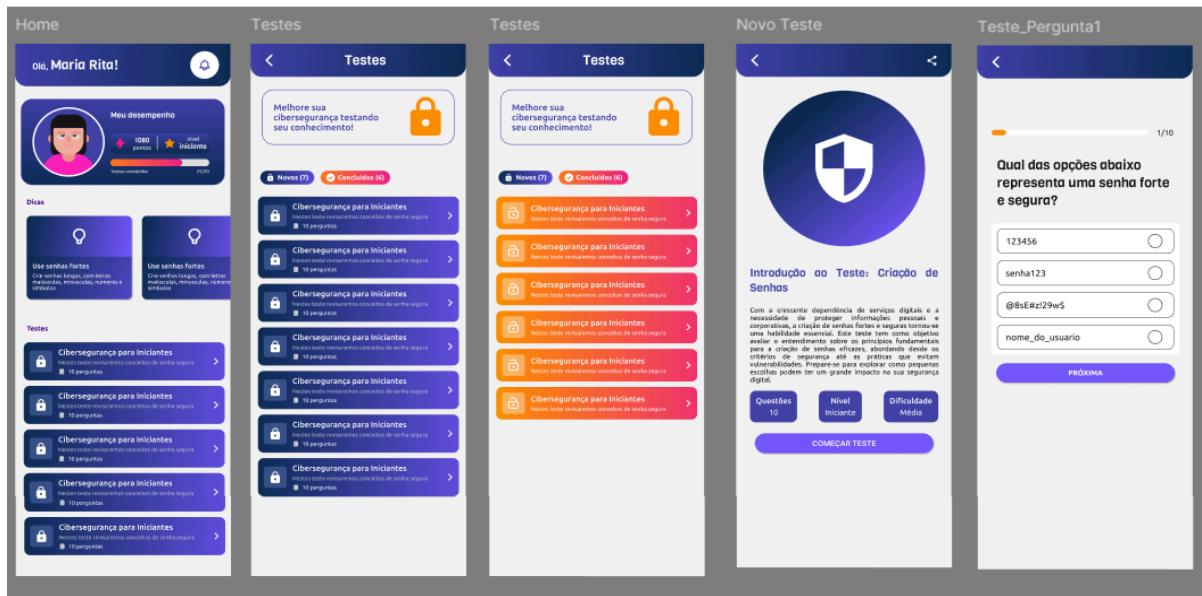
Melhor organização no layout das telas principais, inserindo pequenas introduções em texto, remanejamento dos widgets e componentes buscando melhorar a experiência do usuário.

Figura 36 – Telas principais versão 1



Fonte: Elaborado pelos autores.

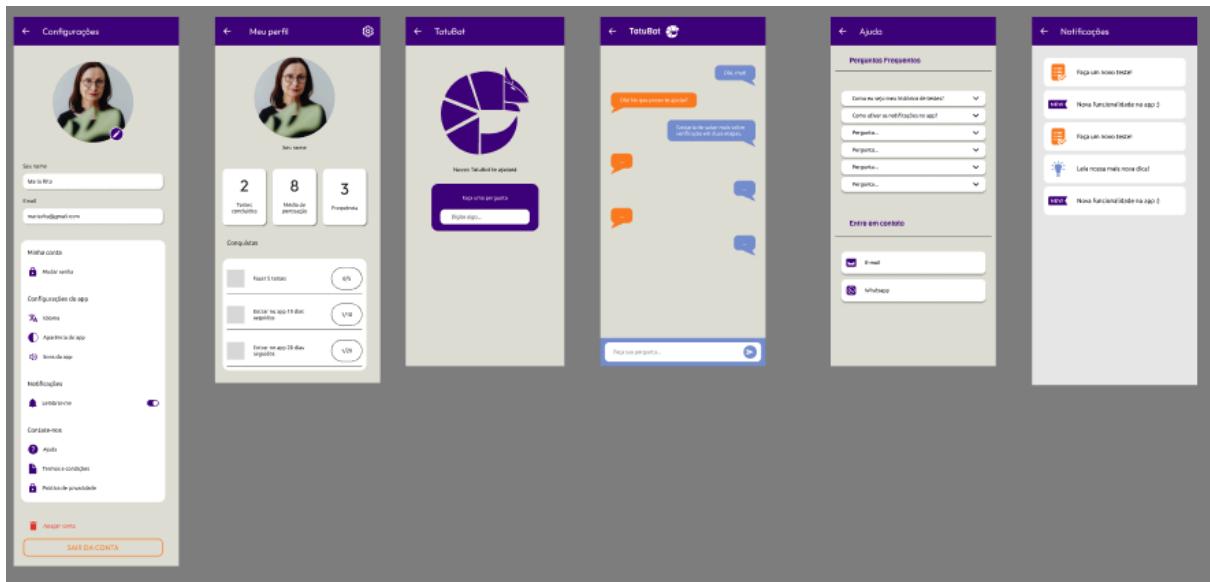
Figura 37 – Telas principais versão 2



Fonte: Elaborado pelos autores.

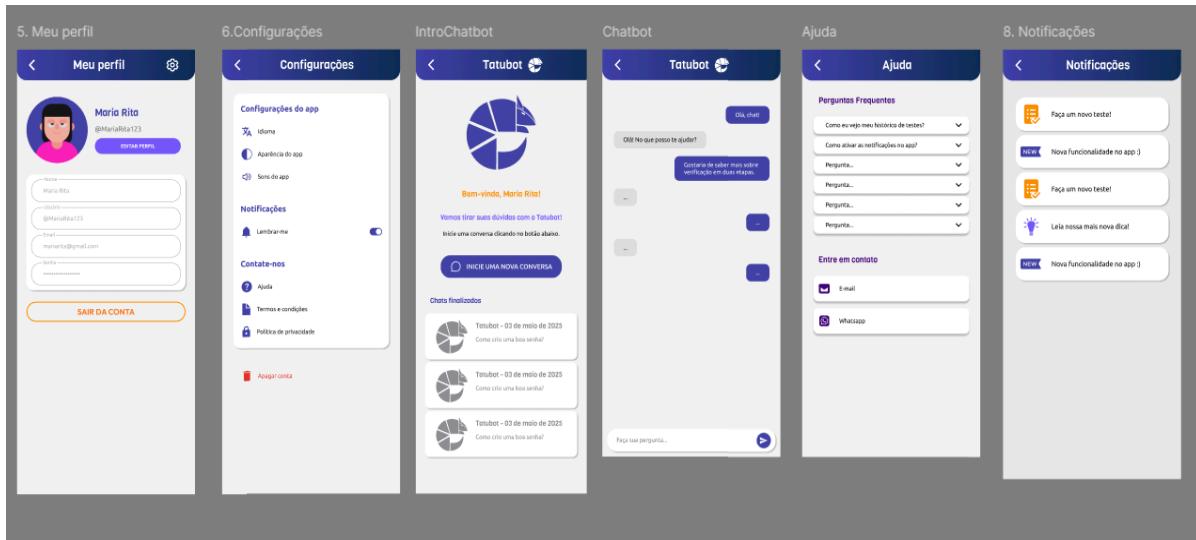
Melhor divisão das telas de “Meu perfil” e “Configurações”, pois observamos que a tela de “Meu perfil” não condizia com a ideia do aplicativo. Reestruturação da tela do chatbot adicionando o histórico do chat.

Figura 38 – Telas principais versão 1



Fonte: Elaborado pelos autores.

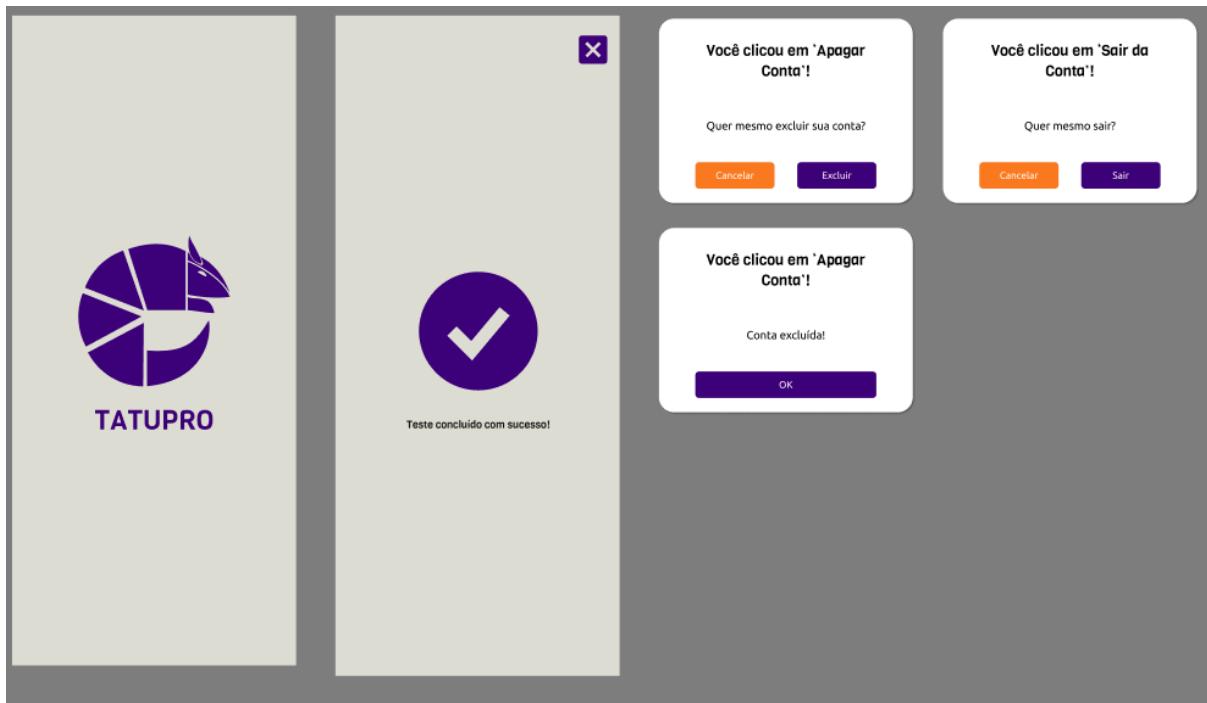
Figura 39 – Telas principais versão 2



Fonte: Elaborado pelos autores.

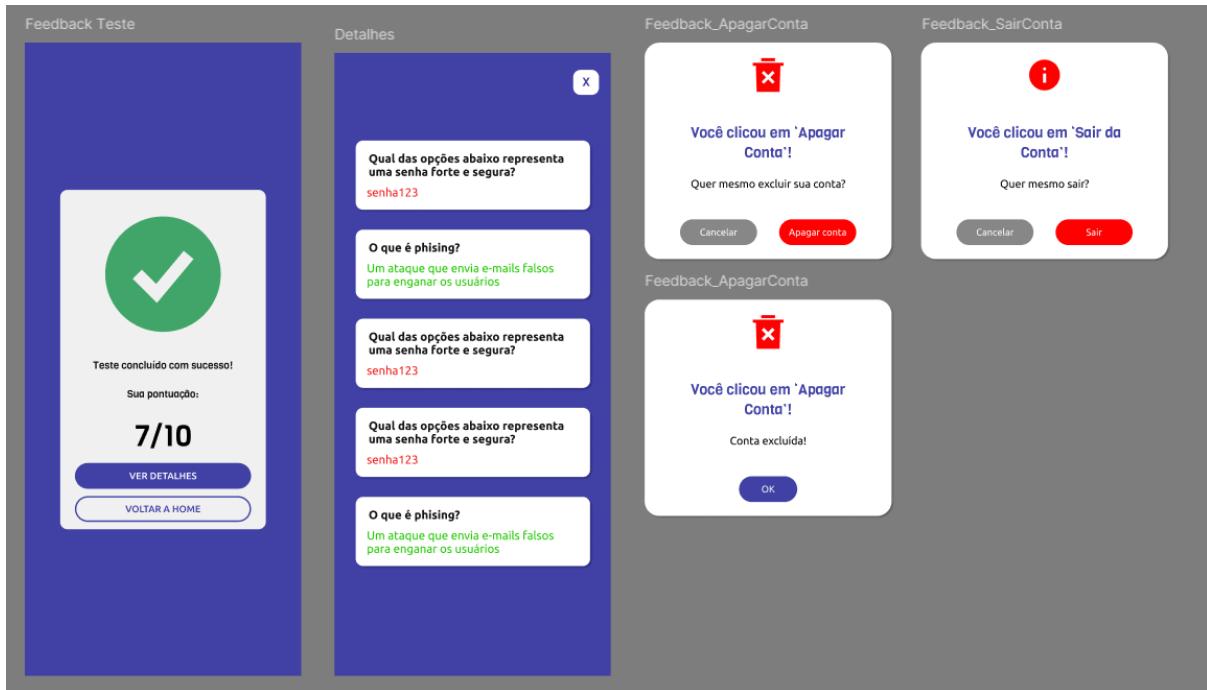
Mudança nas cores das telas de feedback e modais, com o objetivo de priorizar a experiência e melhor entendimento do usuário, diminuindo a chance de erro.

Figura 40 – Feedback versão 1



Fonte: Elaborado pelos autores.

Figura 41 – Feedback versão 2



Fonte: Elaborado pelos autores.

5.9.5. Considerações sobre o Teste realizado

Após a finalização do trabalho de pesquisa e feedback de possíveis usuários, foi possível identificar, avaliar e desenvolver ou reestruturar itens da interface para facilitar a navegação e melhorar o fluxo do aplicativo.

Foi observado que a faixa etária da maioria das respostas se encontra entre 25 e 54 anos, que majoritariamente trabalham e estudam, o principal problema encontrado foi a identificação do objetivo do aplicativo, 92,6 % dos participantes do teste demonstraram algum tipo de dificuldade em entender o objetivo do aplicativo, o que pode ser interpretado como uma falha grave no protótipo, apontando para a necessidade de mudanças na interface: paleta de cores, navegação na parte inferior do aplicativo, botões com rótulos para facilitar a identificação, inclusão de dicas no topo das páginas para que ficasse mais claro o objetivo do aplicativo.

As alterações visam facilitar a navegação e interação dos usuários, com cores mais agradáveis e condizentes com o objetivo final do aplicativo, facilitando assim a usabilidade do mesmo, com feedbacks claros e fáceis de identificar, fazendo com que a jornada do usuário se torne mais fluida e eficaz.

5.10. Tecnologias

5.10.1. Flutter

O desenvolvimento do aplicativo TatuPro será realizado utilizando o framework Flutter, em conjunto com a linguagem de programação Dart. O Flutter é uma tecnologia criada pela Google que permite a construção de aplicativos multiplataforma a partir de uma única base de código. Inicialmente, o TatuPro será desenvolvido primariamente para o sistema operacional Android, por ser o mais utilizado no Brasil. No entanto, a escolha do Flutter garante a escalabilidade futura para dispositivos iOS sem a necessidade de reescrita completa do projeto. A escolha do Flutter/Dart justifica-se principalmente por três fatores: a possibilidade de desenvolvimento multiplataforma, a ampla comunidade ativa que oferece suporte e bibliotecas já consolidadas, e a facilidade de integração com outros serviços, como APIs e bancos de dados. Dessa forma, assegura-se que o TatuPro seja um aplicativo moderno, escalável e de fácil manutenção.

Figura 42 – Logotipo do Flutter



Fonte: GOOGLE, [2025].

5.10.2. Dart

A linguagem Dart, também desenvolvida pela Google, é a base de programação utilizada no Flutter. Ela apresenta uma curva de aprendizado acessível, tipagem segura e alto desempenho, tornando-se uma solução eficiente para aplicações que exigem responsividade e boa experiência de usuário. O Dart funcionará como a parte lógica do aplicativo, desde a integração com banco até a comunicação com API's.

Figura 43 – Logotipo da Dart



Fonte: GOOGLE, [2025].

5.10.3. Gemini

Gemini é um modelo de IA (Inteligência Artificial) generativa do Google, conhecido por suas capacidades avançadas em compreensão e geração de texto, além de poder lidar com imagens e áudio. Durante o desenvolvimento do aplicativo, a API do Gemini será utilizada para

responder perguntas dos usuários, interpretar mensagens e gerar conteúdos relacionados à cibersegurança de forma dinâmica.

Os modelos de linguagem do Gemini destacam-se pela capacidade de compreender e gerar texto de forma contextualizada, possibilitando interações mais próximas de uma conversa humana. Isso favorece o engajamento do usuário e torna a experiência de aprendizado mais intuitiva, já que o aplicativo não se limita apenas a conteúdos estáticos, mas promove uma interação contínua. A escolha dessa tecnologia se justifica pela necessidade de oferecer um recurso que vá além de testes e dicas gerais, permitindo que cada usuário tenha acesso a informações adaptadas às suas demandas específicas. Além disso, o uso de um modelo robusto garante maior precisão nas respostas e a escalabilidade para evoluir o atendimento, podendo futuramente incorporar novas funcionalidades, como suporte em diferentes idiomas ou integração com bases de conhecimento ampliadas.

Figura 44 – Logotipo do Gemini



Fonte: GOOGLE, [2025].

5.10.4. Banco de Dados

O MySQL é um banco de dados relacional bastante conhecido entre os desenvolvedores. Testes, dicas e dados dos usuários serão armazenados no MySQL.

A escolha do MySQL justifica-se por sua ampla documentação, facilidade de integração com diferentes linguagens de programação e compatibilidade com diversas plataformas. Ademais, sua natureza de código aberto possibilita a utilização sem custos adicionais, o que é relevante no desenvolvimento de um projeto acadêmico. Dessa forma, assegura-se que o TatuPro possua uma base de dados estruturada, segura e escalável.

Figura 45 – Logotipo do MySQL

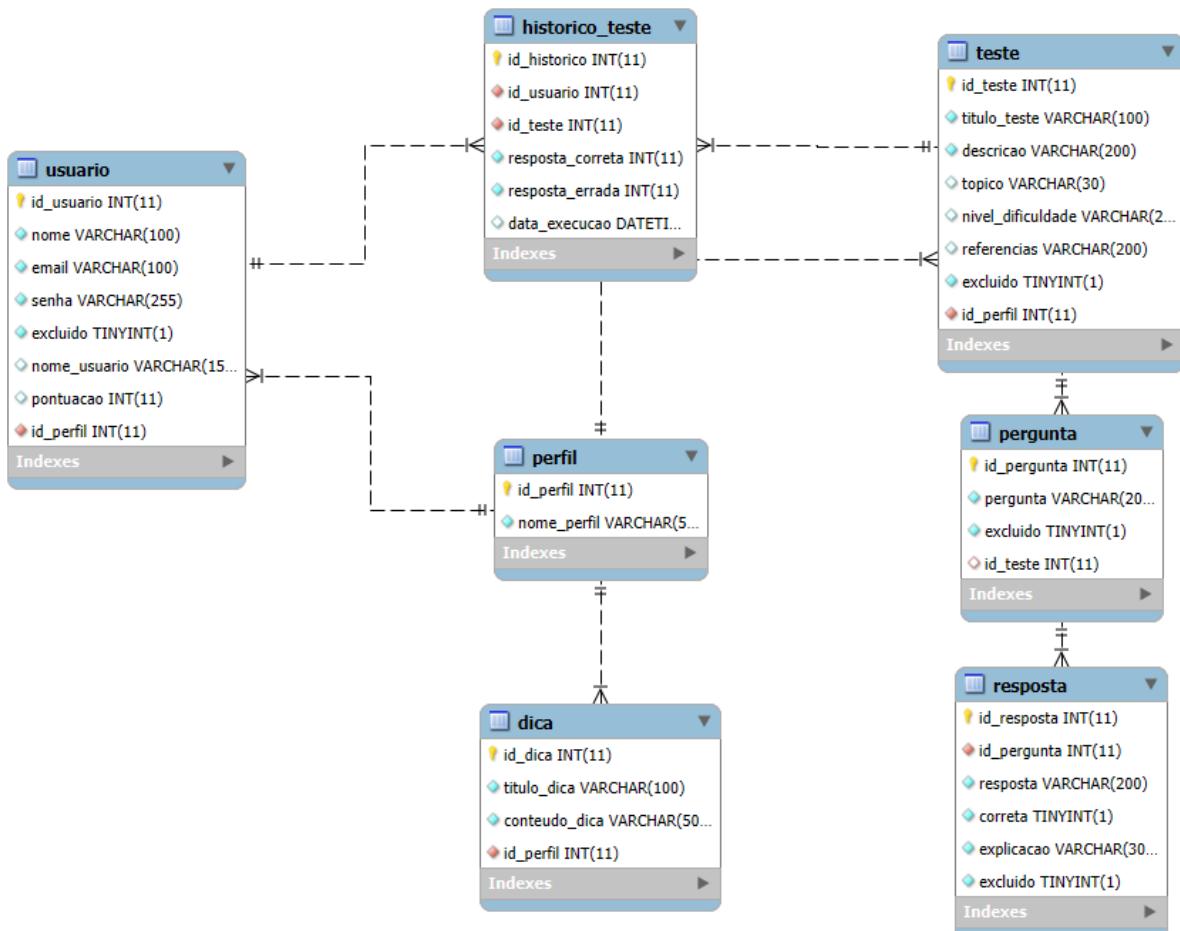


Fonte: GOOGLE, [2025].

5.10.5. Diagrama Entidade Relacionamento

Um Diagrama de Entidade e Relacionamento (DER), é uma representação visual que ilustra como as entidades em um sistema se relacionam entre si. Abaixo está o DER do aplicativo TatuPro.

Figura 46 – Diagrama de Entidade e Relacionamento



Fonte: Elaborado pelos autores.

5.10.6. PHP

O backend do TatuPro será desenvolvido em PHP (*PHP: Hypertext Preprocessor*), uma das linguagens de programação mais utilizadas para aplicações web devido à sua versatilidade, simplicidade de uso e ampla compatibilidade com diferentes servidores e bancos de dados. O PHP será responsável pela lógica da regra de negócio da aplicação, garantindo a comunicação entre o aplicativo desenvolvido em Flutter, o banco de dados relacional MySQL e a integração com o modelo de inteligência artificial do Gemini.

No contexto do TatuPro, o PHP terá como funções principais: gerenciar autenticação e autorização de usuários, processar e validar dados enviados pelo aplicativo, intermediar o acesso seguro aos bancos de dados, além de prover APIs RESTful que permitirão a integração com os diferentes módulos do sistema. Essa arquitetura assegura a separação entre a camada de apresentação (aplicativo) e a camada de dados, promovendo maior segurança.

A escolha pelo PHP justifica-se não apenas por ser uma tecnologia consolidada, com vasta comunidade ativa e ampla documentação, mas também por ter sido uma das linguagens mais trabalhadas ao longo do curso. Isso torna seu uso ainda mais relevante para o desenvolvimento do projeto, uma vez que aproveita competências já adquiridas e coloca em prática conhecimentos consolidados. Além disso, sua compatibilidade nativa com o MySQL, somada à possibilidade de integração com soluções modernas, reforça o PHP como a opção ideal para um projeto que demanda confiabilidade e escalabilidade, como o TatuPro.

Figura 47 – Logotipo do PHP



Fonte: GOOGLE, [2025]

6. DESENVOLVIMENTO

6.1. Backend

6.1.1. API RESTful

O desenvolvimento da arquitetura *backend* deste trabalho de graduação foi fundamentado no estilo REST (*Representational State Transfer*), esse modelo de arquitetura não é uma tecnologia, mas sim um conjunto de restrições e diretrizes que visam aproveitar as capacidades nativas do protocolo HTTP (*Hypertext Transfer Protocol*) e da própria Web, estabelecendo uma comunicação padronizada e leve entre sistemas distribuídos (FIELDING, 2000). O princípio central do REST é a manipulação dos recursos, proporcionando que o cliente interaja com os recursos contidos no backend por meio de requisições utilizando os verbos HTTP (GET, PUT, POST, PATCH, DELETE) padronizados para indicar uma ação desejada.

A estrutura REST serviu para a construção de uma API (Interface de Programação de Aplicações) que será o intermédio entre o cliente e o servidor. Nossa API possui um conceito RESTful, utilizando os métodos HTTP conseguimos comunicar através de requisições entre o cliente e o servidor, neste caso o cliente será representado pela nossa interface de *front end* desenvolvida em flutter, e o nosso servidor, responsável pela comunicação com o backend e pela regra de negócio sendo a API RESTful desenvolvida em PHP. Nessa estrutura também temos a característica de as requisições serem “sem estado”, cada requisição enviada ao servidor deverá conter todas as informações necessárias para que sejam entendidas e processadas sem depender de um estado de sessão previamente armazenado no servidor, simplificando o design do servidor e facilitando a escalabilidade.

Outra importante característica da API RESTful é a interface uniforme, a restrição fundamental que padroniza as interações, ela exige a identificação dos recursos por meio das URLs, indicadas na pasta de rotas da api, nela estão definidos os caminhos e os verbos que serão utilizados para realizar as consultas. A adoção dessa arquitetura foi essencial para que o aplicativo fosse desenvolvido, pois permitiu um desenvolvimento da API desacoplada do *frontend*, com isso a aplicação se tornou leve e padronizada com requisições em JSON, tornando a API escalável, pensando no possível crescimento de usuários e volume de conteúdo.

6.1.2. Modelo MVC (*Model-View-Controller*)

O modelo escolhido para a arquitetura da API foi o MVC, esse modelo de arquitetura define que a aplicação deve ser dividida em três partes interconectadas, cada uma sendo responsável por algum aspecto específico das funcionalidades implementadas no aplicativo.

A primeira parte a ser apresentada é a camada *Model*, responsável pela lógica de negócio e pela manipulação dos dados, a *model* faz a interação entre a aplicação e o banco de dados MySQL, realizando as operações que irão nos trazer as informações solicitadas pela camada controller.

A camada *controller* serve como uma espécie de intermediário entre o frontend e a camada de *model*. É nessa camada que a requisição HTTP recebida do frontend é traduzida para a ação a ser tomada na camada de *model*. Dependendo do verbo que for recebida a solicitação a controller acionará uma função específica dentro da camada *model*, como se traduzir-se a ação solicitada pelo cliente, após isso seleciona a saída apropriada, o formato que os dados devem ser entregues ao cliente.

A terceira camada é a camada da *View* (visão), é a camada que apresenta os dados ou a informação para o cliente/usuário, no caso da API do TatuPro, a camada de view está no aplicativo em flutter. Essa camada é responsável por acessar as rotas de envio de solicitações para que a controller possa saber quais dados necessitam ser enviados.

A adoção desse método para o desenvolvimento da API em PHP trouxe algumas vantagens cruciais para a qualidade e a sustentabilidade do nosso projeto. A principal vantagem é a separação de interesses, quando isolamos a lógica de dados, da lógica da requisição entre *model* e controller, o código nos mostra forma organizada as responsabilidades de cada função dentro das camadas. Outra vantagem é a reusabilidade das lógicas de negócio, sem a necessidade de reescrever o código podemos utilizar as lógicas da *model* em outras controllers. Fazendo a separação em diferentes camadas, outra facilidade encontrada é o fato de que a manutenção do código é facilitada, a localização e a correção de bugs é realizada de uma forma mais precisa. Além desses pontos, podemos observar uma grande facilidade no desenvolvimento distribuído, pois ao mesmo tempo podemos ter diferentes membros da equipe, trabalhando em partes diferentes do código.

6.1.3. Framework

Utilizando os conceitos de API RESTful e modelo de arquitetura MVC, para o desenvolvimento do backend do TatuPro foi utilizado um framework PHP personalizado,

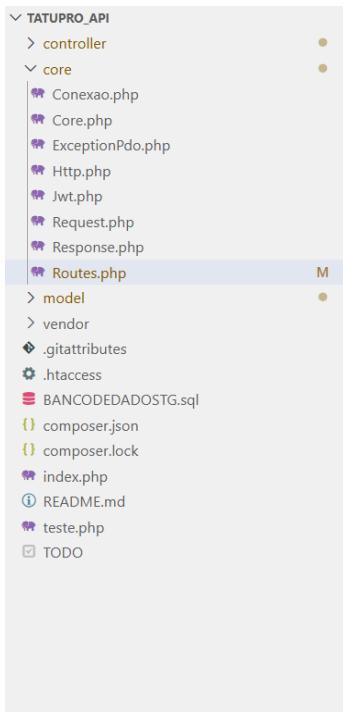
construído em ambiente acadêmico com o objetivo de proporcionar uma compreensão profunda dos conceitos fundamentais de desenvolvimento web e arquitetura de software. Este framework foi elaborado seguindo os princípios de microserviços e design patterns consolidados pela comunidade PHP, resultando em uma estrutura leve, modular e adequada às necessidades específicas do projeto. A organização estrutural do framework segue uma arquitetura bem definida e com separação clara de responsabilidades entre os componentes principais.

O primeiro ponto a ser destacado é o “core/”, a pasta que concentra os elementos essenciais para o funcionamento da aplicação. Dentro dela, o arquivo Conexao.php é responsável por gerenciar a conexão com o banco de dados MySQL, implementando o padrão de projeto Singleton, o que garante a existência de apenas uma instância de conexão durante todo o ciclo de vida da aplicação. O arquivo Core.php atua como o núcleo do framework, sendo encarregado de inicializar a aplicação e coordenar a interação entre os demais módulos. A classe ExceptionPdo.php foi desenvolvida para realizar o tratamento especializado de exceções relacionadas ao banco de dados, fornecendo mensagens de erro mais específicas e contribuindo para a etapa de depuração. O componente Http.php gerencia os elementos associados ao protocolo HTTP, incluindo o controle de headers, status codes e configurações de CORS. Já o arquivo Jwt.php implementa a lógica de autenticação baseada em JSON Web Tokens, permitindo a validação segura de usuários autenticados. Além disso, o módulo Request.php encapsula todas as informações provenientes da requisição HTTP, facilitando o acesso aos parâmetros, cabeçalhos e ao corpo da solicitação. Em contrapartida, o arquivo Response.php padroniza as respostas enviadas pela API, garantindo consistência no formato JSON (*JavaScript Object Notation*) retornado ao cliente. Por fim, o componente Routes.php é responsável pelo sistema de roteamento, realizando o mapeamento das URLs (*Uniform Resource Locator*) para seus respectivos controllers e métodos, com suporte aos principais verbos HTTP, como GET, POST, PUT, DELETE e PATCH.

A segunda parte da estrutura do framework é composta pelos diretórios *controller/*, *model/* e *vendor/*, cada um desempenhando um papel específico dentro da arquitetura proposta. O diretório *controller/*, reúne as classes responsáveis por receber as requisições encaminhadas pelo sistema de roteamento, processar a lógica de negócio correspondente e retornar as respostas apropriadas ao cliente, atuando como intermediário entre a camada de entrada e as demais camadas da aplicação. Em seguida, o diretório *model/* concentra as classes de modelo que realizam a interação direta com o banco de dados, seguindo o padrão *Active Record*, o qual associa cada classe a uma tabela específica e permite que operações de

consulta, inserção, atualização e remoção sejam executadas de forma encapsulada e orientada a objetos. Por fim, o diretório *vendor/* é gerenciado pelo Composer e contém todas as dependências externas utilizadas no projeto, garantindo que bibliotecas de terceiros sejam integradas de maneira padronizada e facilitando tanto a manutenção quanto a escalabilidade da aplicação.

Figura 48 – Estrutura Framework e classe “core/Routes”.



```

core > Routes.php
1  <?php
2
3  require_once 'Http.php';
4
5  Http::get('/usuarios', 'UsuarioController@show');
6  Http::get('/usuarios/relatorio', 'UsuarioController@report');
7  Http::get('/usuarios/busca-usuario', 'UsuarioController@findByID');
8  Http::post('/usuarios/busca-usuario-email', 'UsuarioController@findByEmail');
9  Http::post('/usuarios', 'UsuarioController@add');
10 Http::put('/usuarios', 'UsuarioController@edit');
11 Http::delete('/usuarios', 'UsuarioController@delete');
12 Http::post('/usuarios/login', 'UsuarioController@login');
13 Http::post('/usuarios/logout', 'UsuarioController@exit');
14 Http::patch('/usuarios/edita-foto-perfil/{id}', 'UsuarioController@editPhoto');
15
16 Http::patch('/usuarios/recupera-senha', 'UsuarioController@editPassword');
17
18 Http::patch('/usuarios/gerar-token', 'UsuarioController@gerarToken');
19 Http::patch('/usuarios/atualiza-pontuacao', 'UsuarioController@updatedPoint');
20
21 Http::get('/testes/{id}', 'TesteController@show');
22 Http::get('/testes/consulta-teste/{id}', 'TesteController@findTest');
23 Http::get('/testes/respostas-correctas/{id}', 'TesteController@findAllCorrectAnswer');
24 Http::post('/testes/correge-teste/{id}', 'TesteController@correctTestUser');
25 Http::post('/testes', 'TesteController@add');
26
27 Http::get('/perguntas', 'PerguntaController@show');
28 Http::get('/perguntas/{id}', 'PerguntaController@findByID');
29 Http::post('/perguntas', 'PerguntaController@add');
30 Http::put('/perguntas/{id}', 'PerguntaController@edit');
31 Http::delete('/perguntas/{id}', 'PerguntaController@delete');

```

Fonte: Elaborado pelos autores.

As características e a filosofia do framework desenvolvido refletem fortes influências de microserviços PHP amplamente reconhecidos no mercado, como o Slim Framework e o Lumen, versão minimalista do Laravel. Assim como essas soluções consolidadas, o framework prioriza leveza e performance, mantendo apenas os componentes essenciais para o funcionamento de uma API RESTful e evitando funcionalidades desnecessárias que possam comprometer sua eficiência. Além disso, apresenta elevada flexibilidade, permitindo customizações alinhadas às necessidades específicas do projeto, sem impor convenções rígidas ou acoplamentos excessivos. A separação de responsabilidades também se destaca como princípio fundamental, uma vez que cada componente cumpre uma função claramente definida, favorecendo a manutenção, a organização interna e a execução de testes. Paralelamente, o framework mantém foco exclusivo na construção de APIs, oferecendo uma

estrutura otimizada para o desenvolvimento RESTful, com suporte nativo à autenticação via JWT (*JSON Web Token*) e ao retorno de respostas em formato JSON.

A adoção de um framework desenvolvido em ambiente acadêmico se justifica pela oportunidade de proporcionar aos desenvolvedores uma compreensão aprofundada sobre o funcionamento interno de frameworks modernos, abrangendo desde o sistema de roteamento até o tratamento de requisições e respostas. Essa abordagem também garantiu total controle sobre a arquitetura e o comportamento da aplicação, permitindo a realização de adaptações específicas para atender aos requisitos do TatuPro. Embora ferramentas amplamente difundidas, como Laravel, Slim ou CodeIgniter, ofereçam maior variedade de recursos e comunidades mais robustas, o framework personalizado demonstrou ser plenamente capaz de atender às demandas do projeto, evidenciando que soluções educacionais bem estruturadas podem alcançar eficiência comparável à de ferramentas profissionais em aplicações de médio porte. Como resultado, a experiência contribuiu significativamente para o desenvolvimento das competências técnicas da equipe, fortalecendo conhecimentos que poderão ser aplicados em qualquer framework PHP profissional no futuro.

6.2. Banco de Dados

O Desenvolvimento do banco relacional aplicado ao TatuPro foi implementado utilizando o SGBD (Sistema Gerenciador de Banco de Dados) MySQL, a modelagem foi estrategicamente desenvolvida para refletir a lógica de aprendizado, onde o conteúdo de testes e dicas , é segmentado por níveis de proficiência, e o desempenho do usuário é rastreado a partir de um histórico detalhado. No modelo de Entidade-Relacionamento (MER) podemos observar que a estrutura de construção do banco gira em torno de três eixos principais: Usuário e Perfil, Conteúdo (Testes e Dicas) e Rastreamento de Progresso. Na tabela abaixo podemos observar o comportamento das principais tabelas e seus devidos relacionamentos chave:

Tabela 4 – Principais tabelas e relacionamentos.

Tabela	Propósito Principal	Relacionamentos Chave
“perfil”	Definir os níveis de proficiência do usuário (iniciante, intermediário e avançado).	1:N com “usuario”, “teste” e “dica”.
“usuario”	Armazena as informações de autenticação como: nome, email, senha e também o “id_perfil” do usuário.	N:1 com “perfil”; 1:N com “historico_teste”.
“teste”	Define os conjuntos de questões a serem respondidas, ligadas ao “id_perfil” específico.	N:1 com “perfil”; 1:N com “pergunta”.
“pergunta”	Armazena o enunciado, tópico e nível de dificuldade da questão, e está ligada a um teste.	N:1 com “teste”; 1:N com “resposta”.
“resposta”	Armazena as opções de resposta, indicando se é a correta e a sua explicação detalhada.	N:1 com “pergunta”.
“historico_teste”	Tabela de ligação (many-to-many) que armazena o desempenho do usuário em cada teste realizado.	N:1 com “usuario” e “teste”.
“dica”	Armazena conteúdo auxiliar de segurança, segmentado por “id_perfil”.	N:1 com “perfil”.

Fonte: Elaborado pelos autores.

Com o objetivo de assegurar a coerência dos dados e a segurança da aplicação, foram implementados diferentes mecanismos de controle e integridade. A integridade referencial é garantida por meio da utilização de chaves estrangeiras (FOREIGN KEY), assegurando que todas as relações entre as tabelas sejam devidamente mantidas. Desta forma, um registro presente na tabela “historico_teste” somente pode existir se estiver associado a um id_usuario e a um “id_teste” válidos e previamente cadastrados em suas respectivas tabelas.

No que se refere ao gerenciamento de usuários, a coluna senha da tabela “usuario” foi definida com o tipo de dado VARCHAR(255), reforçando a adoção de boas práticas de segurança da informação, uma vez que tal configuração possibilita o armazenamento de *hashes* criptográficos em substituição às senhas em texto puro. Adicionalmente, as colunas “status_usuario” e “excluido” implementam a técnica conhecida como *Soft Delete* (exclusão lógica), a qual permite que a aplicação oculte usuários inativos sem a necessidade de remover definitivamente seus registros e histórico do banco de dados.

O processo de rastreamento de progresso, representado pela tabela “historico _ teste”, desempenha papel essencial na lógica de aprendizado da aplicação. Essa tabela armazena o desempenho de cada tentativa realizada pelos usuários, registrando o “id_usuario”, o “id _ teste”, a contagem de respostas corretas e incorretas, além da data e hora de execução (“data_execucao”). Tais informações constituem a base de análise utilizada pelo Controller para determinar o progresso individual e o status de cada teste.

Por fim, a estrutura modular de conteúdo, composta pelas tabelas de perguntas e respostas, foi projetada de modo a proporcionar maior flexibilidade e capacidade de autoavaliação nos testes. A separação entre as entidades pergunta e resposta, juntamente com a presença da coluna correta (do tipo BOOL), contribui para uma organização mais eficiente das informações. Adicionalmente, a coluna “explicacao” na tabela resposta desempenha função pedagógica relevante, ao oferecer feedback imediato ao usuário, fortalecendo o caráter educativo e interativo da aplicação.

6.3. FrontEnd

O desenvolvimento da interface do aplicativo Tatupro foi realizado utilizando o framework Flutter, que possibilita a criação de aplicações multiplataforma a partir de uma única base de código. A escolha do Flutter justifica-se por diversos fatores que o tornam uma solução robusta e eficiente para o desenvolvimento mobile moderno. A ferramenta apresenta diversas vantagens para o processo de desenvolvimento como por exemplo a possibilidade de criar projetos para diversas plataformas: Android, iOS, Web e até mesmo Desktop; a principal diferença do Flutter, é que o framework compila o sistema desenvolvido na linguagem Dart para o código nativo dessas plataformas sem precisar de pontes(bridges) para comunicação.

Outra funcionalidade muito importante desse *framework* é o “Hot Reload”, com ela, em tempo real, podemos observar as alterações feitas no código, sem a necessidade de recompilar toda a aplicação, acelerando drasticamente o processo de desenvolvimento. A documentação do Flutter é robusta e bem estruturada, facilitando o aprendizado e a resolução de problemas, além de uma comunidade ativa e um ecossistema rico em contribuições, conta com um vasto repositório de pacotes e bibliotecas que auxiliam na implementação de funcionalidades complexas.

A estrutura de pastas escolhida para a implementação do aplicativo Tatupro foi planejada para separar os componentes e as funcionalidades de forma única e de fácil identificação:

- lib/model: Contém as classes e modelo que representam as entidades do sistema, dicas, testes, usuário, perguntas, facilitando a manipulação dos dados que serão recebidos via API.
- lib/screens: Armazena as telas do aplicativo, organizadas de forma modular para facilitar a navegação e a manutenção do código.
- lib/screens/core_widgets: Reúne os componentes reutilizáveis da interface, como por exemplo botões personalizados, cards e elementos visuais que aparecerão em várias telas.
- lib/themes: Define a identidade visual do aplicativo, incluindo a paleta de cores, a tipografia padronizada e os estilos globais da aplicação.

Para garantir que as funcionalidades mais complexas do aplicativo fossem completas e pensando na experiência do usuário final, foram integradas diversas bibliotecas especializadas no projeto:

- google_fonts: Facilita a integração de fontes personalizadas do Google Fonts, contribuindo para a identidade visual do TatuPro e melhorando a legibilidade do conteúdo educativo apresentado aos usuários.
- dio: Biblioteca poderosa para realizar requisições HTTP à API RESTful, escolhida por oferecer recursos avançados como interceptadores, tratamento de erros robusto, timeout configurável e melhor gerenciamento de requisições assíncronas. O Dio permitiu implementar uma camada de comunicação eficiente entre o frontend e o backend, facilitando o debug e a manutenção das chamadas à API.
- accordion: Implementa componentes de acordeão que permitem organizar o conteúdo educativo de forma expansível e colapsável, melhorando a naveabilidade e permitindo que o usuário acesse informações de maneira progressiva, sem sobrecarregar visualmente as telas de dicas e lições.
- curved_labeled_navigation_bar: Biblioteca utilizada para criar a barra de navegação inferior do aplicativo com design moderno e curvilíneo. A escolha desta biblioteca é especialmente relevante por incluir labels (rótulos) em cada botão de navegação, aspecto fundamental identificado durante o teste de usabilidade. Os labels garantem que os usuários, especialmente aqueles com menor familiaridade tecnológica, compreendam claramente a função de cada botão, reduzindo a curva de aprendizado e melhorando significativamente a experiência de navegação no aplicativo.

- `get_storage`: Solução leve e rápida para armazenamento local de dados, utilizada para persistir informações como tokens de autenticação, preferências do usuário e dados de sessão. Oferece melhor performance em comparação com outras soluções de armazenamento local, sendo ideal para dados que precisam ser acessados frequentemente.
- `jwt_decoder`: Biblioteca especializada em decodificar tokens JWT (JSON Web Tokens), essencial para o sistema de autenticação do aplicativo. Permite extrair e validar informações do usuário contidas no token, garantindo segurança na comunicação entre o frontend e o backend.
- `google_generative_ai`: Biblioteca oficial do Google para integração com a API Gemini, utilizada especificamente para implementar o chatbot do TatuPro. Esta biblioteca possibilita que o aplicativo se comunique diretamente com os modelos de IA generativa do Google, permitindo que os usuários façam perguntas sobre cibersegurança e recebam respostas contextualizadas e personalizadas em tempo real, cumprindo um dos objetivos específicos do projeto de oferecer suporte personalizado através de IA.

A escolha dessas bibliotecas justifica-se pela necessidade de criar uma aplicação responsiva, performática e com boa experiência de usuário, aproveitando soluções já consolidadas pela comunidade Flutter. A combinação dessas tecnologias permitiu implementar todas as funcionalidades previstas nos requisitos funcionais, desde a autenticação de usuários até a integração com o chatbot baseado em IA, mantendo o código organizado e escalável para futuras melhorias.

6.3.1. Chatbot

O aplicativo TatuPro conta com um chatbot, desenvolvido com um modelo do Gemini e desempenha um papel fundamental ao oferecer suporte personalizado em tempo real aos usuários. Sua criação envolveu a fusão da biblioteca `google_generative_ai` com o modelo `gemini-2.5-flash` do Google, garantindo respostas precisas sobre segurança cibernética de maneira contextualizada. A estrutura do chatbot foi planejada levando em consideração três elementos-chave: interação com a API Gemini, administração de diálogos e retenção local de informações. A biblioteca `google_generative_ai` tem a função de facilitar a comunicação com os modelos de IA do Google, permitindo a troca de mensagens estruturadas e respostas em linguagem natural. O modelo `gemini-2.5-flash` foi escolhido por sua combinação de velocidade e qualidade nas respostas, tornando-o ideal para interações em tempo real.

Para preservar o histórico de conversas mesmo após o fechamento do aplicativo, foi integrado um sistema de armazenamento local com SQLite utilizando a biblioteca sqflite versão 2.3.0. Essa abordagem foi adotada devido à necessidade de armazenar dados complexos relacionados às conversas de forma eficiente. O banco de dados possui duas tabelas principais: uma para armazenar detalhes de cada conversa, como identificador único, título e status, e outra para registrar todas as mensagens trocadas, identificando remetentes, conteúdo e horários de envio. A biblioteca path versão 1.8.3 foi integrada ao sqflite para gerenciar os caminhos dos arquivos de forma consistente em diferentes sistemas operacionais.

O sistema de gestão de conversas garante a persistência de todas as interações no banco de dados local, permitindo aos usuários retomar conversas anteriores a qualquer momento. A interface do chatbot foi desenvolvida para facilitar a visualização e interação, com recursos como indicações de chats ativos e organização clara do histórico de conversas concluídas. A classe DatabaseHelper foi criada para supervisionar as operações relacionadas ao SQLite, seguindo o padrão Singleton para manter a consistência. A exclusão de dados foi implementada de modo a preservar a integridade referencial e evitar a remoção acidental de informações cruciais.

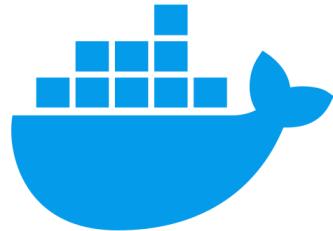
A utilização de modelos de dados específicos simplifica a representação das entidades do sistema, facilitando a interação com o banco de dados e abrindo possibilidades para futuras expansões. Essa implementação oferece diversas vantagens aos usuários, como salvamento automático, histórico persistente e continuidade de conversas ativas, resultando em uma experiência de uso mais fluida e natural. Em termos técnicos, o uso do SQLite garante um bom desempenho mesmo com grandes volumes de dados, e a arquitetura modular simplifica os testes e a manutenção do sistema. A escolha de bibliotecas consolidadas na comunidade Flutter garante estabilidade e suporte a longo prazo, atendendo com eficácia aos objetivos educacionais do TatuPro.

6.4. Módulo de Geração Automatizada de Testes

Esta seção descreve a infraestrutura responsável pela geração automatizada dos testes educativos no aplicativo. Essa infraestrutura possibilitou o uso de IA para criar testes adaptados ao nível de conhecimento de cada usuário.

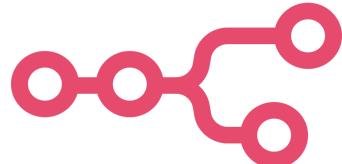
A infraestrutura do projeto foi composta por duas ferramentas principais: o Docker, responsável por fornecer um ambiente de execução isolado, e o n8n, utilizado para a automação dos fluxos de geração de testes.

Figura 49 – Logotipo do Docker



Fonte: GOOGLE, [2025].

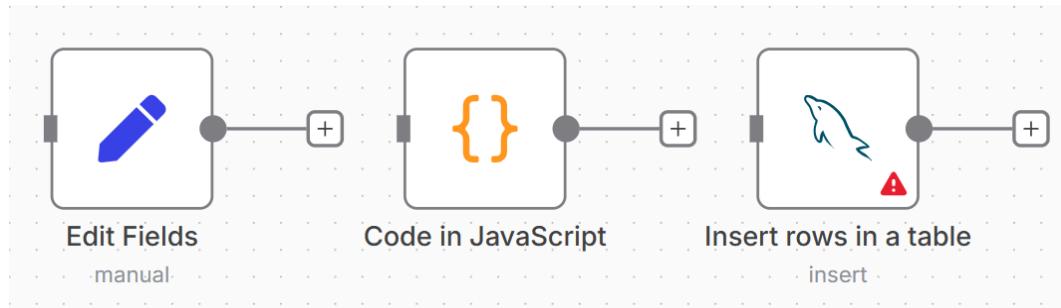
Figura 50 – Logotipo do n8n



Fonte: GOOGLE, [2025].

O n8n é uma plataforma de automação de fluxos de trabalho de código aberto e low-code que possibilita conectar diferentes aplicativos e serviços por meio de uma interface visual intuitiva. Por meio de blocos visuais, chamados *nós*, é possível criar conexões com bancos de dados e APIs RESTful. Assim, o n8n foi utilizado como facilitador dentro do contexto do aplicativo Tatupro, servindo como uma alternativa rápida e experimental para a utilização de IA na geração dos testes educativos.

Figura 51 – Nós n8n



Fonte: Elaborado pelos autores.

Para sustentar o uso do n8n e manter o ambiente de testes estável e seguro, utilizou-se o Docker, uma plataforma de código aberto que permite construir, empacotar e executar aplicações em unidades padronizadas chamadas contêineres. Os contêineres reúnem tudo o que o sistema precisa, como código, bibliotecas e configurações, em um único pacote. Por utilizarem os recursos do próprio sistema operacional, são mais leves e rápidos que as máquinas virtuais, garantindo que o aplicativo funcione da mesma forma em qualquer ambiente.

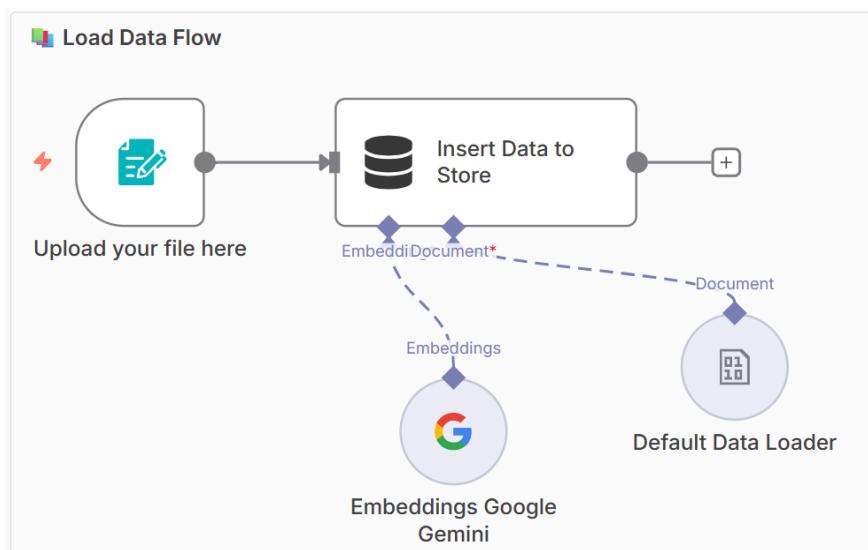
Atualmente, o n8n é executado via Docker, o que permite seu funcionamento local e acesso a todas as funcionalidades sem restrições. Outros benefícios dessa combinação incluem a criação de um ambiente autocontido, que evita conflitos com outras aplicações do dispositivo, e a possibilidade de migrar a instância do n8n para diferentes ambientes (como uma VPS) sem necessidade de reconfigurar dependências, garantindo maior estabilidade.

Um agente de IA possui autonomia, capacidade de raciocínio, planejamento e tomada de decisão. Com a infraestrutura de automação configurada, foi possível integrar um agente inteligente responsável por analisar o desempenho dos usuários e gerar automaticamente novos testes educativos.

Para aprimorar a precisão e a relevância das respostas, foi aplicada a técnica RAG (Retrieval-Augmented Generation), que conecta grandes modelos de linguagem (LLMs) a uma base de conhecimento externa. Em vez de “alucinar”, o sistema recupera informações confiáveis antes de gerar a resposta, produzindo conteúdos mais precisos, atualizados e contextualizados.

Para garantir uma base de conhecimento relevante e segura para o agente dentro do fluxo de automação, utilizou-se o modelo demo do n8n para experimentos com bancos de dados vetoriais. Os bancos de dados vetorizados são especializados em armazenar representações numéricas de textos, imagens ou áudios chamadas embeddings, permitindo que o sistema identifique informações conceitualmente semelhantes, e não apenas correspondências literais de palavras-chave. Dessa forma, o agente consegue adaptar-se e aprimorar suas respostas com base nas interações realizadas ao longo do tempo.

Figura 52 – Modelo DEMO n8n



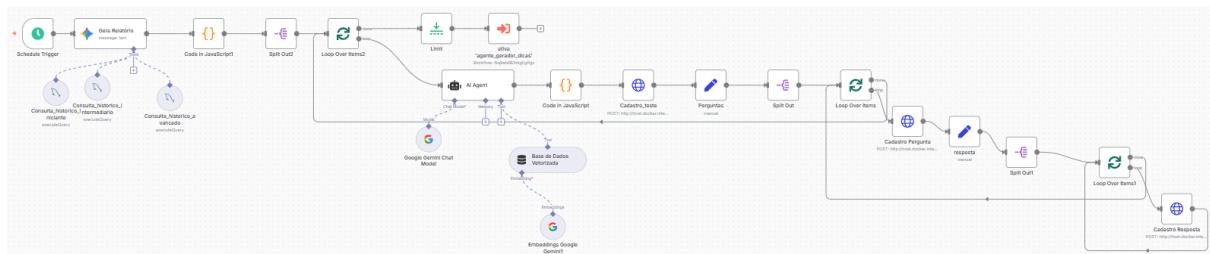
Fonte: Elaborado pelos autores.

Além disso, também foi utilizada a técnica de engenharia de prompt structured prompting, que consiste em fornecer instruções detalhadas e organizadas a um agente de IA. Diferente de um prompt simples, no structured prompting foram usadas estruturas XML (*eXtensible Markup Language*) para delimitar contexto, persona e regras ao Agente de IA e até mesmo um exemplo do que se espera da agente através de um exemplo em JSON.

O fluxo de geração de testes é acionado automaticamente uma vez por semana pelo aplicativo. Durante esse processo, o agente de IA analisa o desempenho dos usuários e cria novos testes adequados a cada nível. Os dados, como perguntas e respostas são cadastrados automaticamente no banco de dados MySQL por meio do n8n.

Para garantir que as perguntas tenham embasamento técnico e confiável, o agente de IA utiliza como referência conteúdos armazenados no banco vetorial, como a LGPD (Lei Geral de Proteção de Dados Pessoais) e a Cartilha de Cibersegurança do CERT.br.

Figura 53 – Fluxo de automação de testes no n8n



Fonte: Elaborado pelos autores.

Após a construção do fluxo de automação, o sistema foi executado para avaliar o possível custo da API do Gemini, tendo como resultado um teste para cada nível.

O painel exibe quatro métricas principais que permitem acompanhar o comportamento da integração entre o n8n e a API Gemini. A primeira métrica, Total API Requests per minute, mostra o número total de requisições feitas à API por minuto. É possível observar picos de atividade entre 4:40 PM e 5:10 PM, correspondendo aos momentos em que o fluxo de geração de testes foi acionado no n8n.

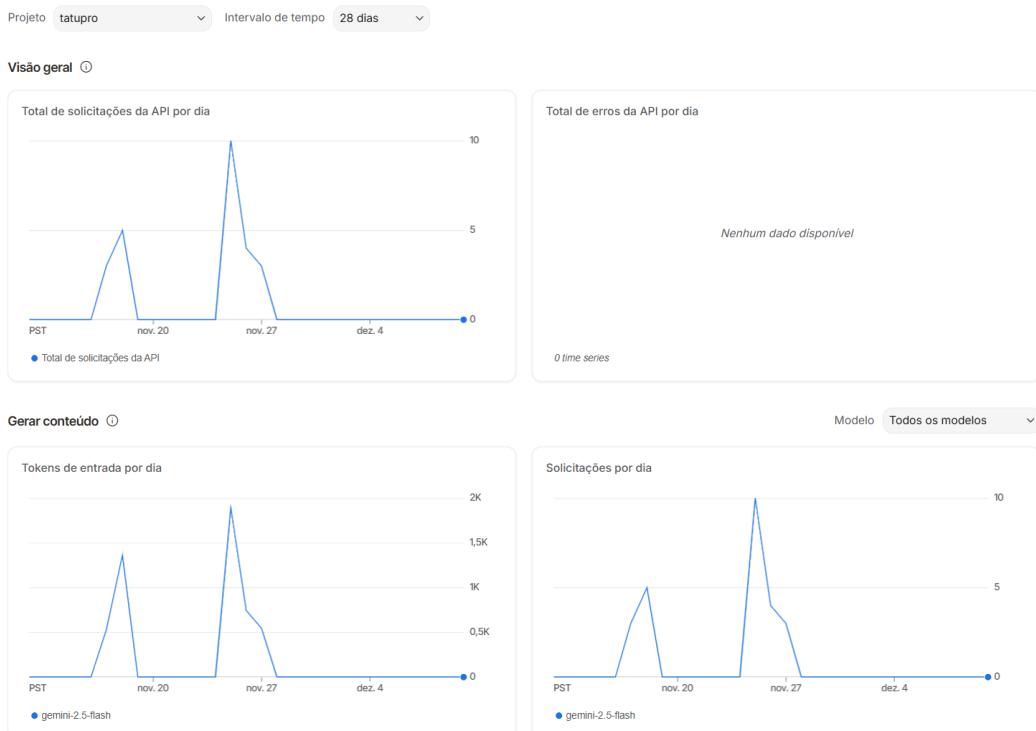
A segunda métrica, Total API Errors per minute, apresenta o número de erros retornados pela API durante o período analisado. Neste caso, não há dados registrados, o que demonstra que todas as requisições foram processadas com sucesso, sem falhas de comunicação ou autenticação.

Em seguida, a métrica Input Tokens per minute indica o volume de tokens de entrada, ou seja, a quantidade de texto enviada à API para processamento. Os picos observados nesse

gráficos revelam que, durante a execução dos fluxos, foram enviados prompts mais complexos ou extensos, resultando em um consumo maior de tokens.

Por fim, a métrica Requests per minute (por modelo) exibe o número de requisições por minuto filtradas pelo modelo utilizado neste caso, gemini-2.5-flash. É possível observar uma sequência de requisições espaçadas, o que evidencia o comportamento sequencial do fluxo de automação no n8n ao gerar os testes de forma dinâmica.

Figura 54 – Dashboard Google IA Studio



Fonte: Elaborado pelos autores.

7. CONCLUSÃO DO TRABALHO

O desenvolvimento do aplicativo TatuPro representou uma resposta concreta à crescente vulnerabilidade digital da população brasileira, especialmente entre usuários com pouco conhecimento técnico em cibersegurança. O projeto partiu da constatação de que 24% da população brasileira acima de 16 anos já foi vítima de golpes digitais, evidenciando uma lacuna crítica entre o avanço tecnológico e a capacitação dos usuários comuns para utilizar essas tecnologias de forma segura.

A metodologia adotada, que combinou pesquisa de similares, definição de personas, testes de usabilidade e prototipagem iterativa, mostrou-se fundamental para garantir que o

produto final atendesse às reais necessidades do público-alvo. Os testes realizados com 27 participantes foram especialmente valiosos, revelando problemas de identificação do objetivo do aplicativo e dificuldades de navegação que, uma vez corrigidos, resultaram em uma interface significativamente mais intuitiva e acessível.

Do ponto de vista técnico, a arquitetura desenvolvida demonstrou solidez e escalabilidade. A escolha do Flutter para o frontend possibilitou a criação de uma interface moderna e responsiva, enquanto a API RESTful desenvolvida em PHP, seguindo o padrão MVC, garantiu uma comunicação eficiente e organizada entre as camadas da aplicação. A utilização de um framework PHP personalizado, desenvolvido em ambiente acadêmico, proporcionou à equipe uma compreensão profunda dos fundamentos de desenvolvimento web, conhecimento que será aplicável em qualquer contexto profissional futuro.

A integração da API Gemini para o chatbot e a implementação do módulo de geração automatizada de testes utilizando n8n e Docker representaram inovações importantes no projeto. A aplicação da técnica RAG (Retrieval-Augmented Generation) com bancos de dados vetoriais contendo documentos oficiais como a LGPD e a Cartilha de Cibersegurança do CERT.br garantiu que os conteúdos gerados tenham embasamento técnico confiável, diferenciando o TatuPro de soluções genéricas.

Os objetivos específicos estabelecidos foram plenamente alcançados: identificaram-se os principais tipos de golpes que afetam usuários brasileiros, elaboraram-se conteúdos educativos acessíveis, desenvolveu-se um chatbot funcional capaz de responder dúvidas em tempo real, propôs-se uma interface intuitiva validada por usuários reais e contribuiu-se para a construção de uma cultura de prevenção digital alinhada aos princípios da LGPD.

Como trabalhos futuros, sugere-se a expansão do aplicativo para a plataforma iOS, a implementação de gamificação para aumentar o engajamento dos usuários, a criação de um sistema de notificações sobre novas ameaças digitais e a integração com órgãos de segurança pública para facilitar denúncias de crimes cibernéticos. Adicionalmente, a base de conhecimento do chatbot pode ser ampliada e o módulo de geração automatizada de testes pode ser refinado com base em dados de desempenho real dos usuários.

Conclui-se que o TatuPro cumpre sua missão de oferecer uma "armadura digital" ao cidadão comum, representando uma ferramenta educativa relevante no cenário nacional de combate aos crimes cibernéticos. Mais do que um aplicativo, o projeto materializa o compromisso com a democratização do conhecimento em segurança digital, contribuindo para uma sociedade mais consciente, preparada e resiliente frente às ameaças do ambiente virtual.

8. REFERÊNCIAS

AGÊNCIA BRASIL. 1 em cada 4 brasileiros caiu em golpe digital nos últimos 12 meses.

Agência Brasil, 7 set. 2024. Disponível em:

<https://agenciabrasil.ebc.com.br/radioagencia-nacional/seguranca/audio/2024-09/1-em-cada-4-brasileiros-caiu-em-golpe-digital-nos-ultimos-12-meses>. Acesso em: 4 dez. 2024.

INSTITUTO DATASENADO. Golpes digitais atingem 24% da população brasileira, revela DataSenado. *Senado Notícias*, Brasília, 1º out. 2024. Disponível em:

<https://www12.senado.leg.br/noticias/materias/2024/10/01/golpes-digitais-atingem-24-da-populacao-brasileira-revela-datasenado>. Acesso em: 11 dez. 2024.

CACHIONI, M.; FLAUZINO, K. de L.; BATISTONI, S. S. T.; CLIQUET, L. O. B. V.; ZAINE, I.; PIMENTEL, M. da G. C. **IDOSOS ON - LINE: TECNOLOGIA COMO RECURSO PARA A APRENDIZAGEM AO LONGO DA VIDA**. Estudos Interdisciplinares sobre o Envelhecimento, [S. l.], v. 25, 2021. DOI: 10.22456/2316-2171.118156. Disponível em: <https://seer.ufrgs.br/index.php/RevEnvelhecer/article/view/118156>. Acesso em: 07 maio. 2025.

CERT.BR. *Fascículos da Cartilha de Segurança*. CERT.br, [s.d.]. Disponível em:

<https://cartilha.cert.br/fasciculos/>. Acesso em: 2 dez. 2024.

CNN BRASIL. Datafolha: país tem mais de 4,5 mil tentativas de golpe financeiro por hora.

CNN Brasil, 13 ago. 2024. Disponível em:

<https://www.cnnbrasil.com.br/nacional/datafolha-pais-tem-mais-de-45-mil-tentativas-de-golpe-financeiro-por-hora/>. Acesso em: 11 dez. 2024.

FIELDING, Roy Thomas. *Architectural styles and the design of network-based software architectures*. 2000. Tese (Doutorado em Informática) – University of California, Irvine, 2000.

HOEPERS, Cristine. **A Importância dos Fatores Humanos para a Cibersegurança. Computação Brasil**, [S. l.], n. 52, p. 61–66, 2024. DOI: 10.5753/compbr.2024.52.4604 Disponível em: <https://journals-sol.sbc.org.br/index.php/comp-br/article/view/4604>. Acesso em: 02 maio. 2025.

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA (IBGE). Internet foi acessada em 72,5 milhões de domicílios do país em 2023. *Agência de Notícias do IBGE*, 16 ago. 2024. Disponível em:

<https://agenciadenoticias.ibge.gov.br/agencia-noticias/2012-agencia-de-noticias/noticias/41024-internet-foi-acessada-em-72-5-milhoes-de-domiciliros-do-pais-em-2023>. Acesso em: 12

mai. 2025.

NOLASCO, Loreci Gottschalk; MACIEL SILVA, Bruno Dutra. **Crimes cibernéticos, privacidade e cibersegurança.** REVISTA QUAESTIO IURIS, [S. l.], v. 15, n. 4, p. 2353–2389, 2022. DOI: 10.12957/rqi.2022.67976. Disponível em: <https://www.e-publicacoes.uerj.br/quaestioiuris/article/view/67976>. Acesso em: 05 maio. 2025.

OLHAR DIGITAL. *Gerenciador de senhas: os 5 melhores apps para você usar.* Olhar Digital, 16 jun. 2023. Disponível em:

<https://olhardigital.com.br/2023/06/16/internet-e-redes-sociais/gerenciador-de-senhas-os-5-melhores-apps-para-voce-usar/>. Acesso em: 5 dez. 2024.

SANTOS, J. V. F. dos; ARAÚJO, A. de. C. **CIBERSEGURANÇA E A IMPORTÂNCIA DO DIREITO DIGITAL.** Revista Multidisciplinar do Nordeste Mineiro, [S. l.], v. 12, n. 1, 2023. DOI: 10.61164/rmn.v12i1.1738. Disponível em: <https://revista.unipacto.com.br/index.php/multidisciplinar/article/view/1738>. Acesso em: 02 maio. 2025.

XPERTS ACADEMY. *Ferramentas open source para computação forense: impulsionando sua carreira.* Xperts Academy, [s.d.]. Disponível em:

<https://xpertsacademy.com.br/ferramentas-open-source-para-computacao-forense-impulsionando-sua-carreira/>. Acesso em: 11 dez. 2024.