

CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA SOUZA

Faculdade de Tecnologia da Praia Grande

Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

SCRIPT FORGE

SITE DE CRIAÇÃO DE ROTEIROS PARA JOGOS DIGITAIS

Guilherme Rodrigues

Diego Caravana Mangini

PRAIA GRANDE

2025

Guilherme Rodrigues
Diego Caravana Mangini

SCRIPT FORGE
SITE DE CRIAÇÃO DE ROTEIROS PARA JOGOS DIGITAIS

Trabalho de Conclusão de Curso apresentado à Faculdade de Tecnologia da Praia Grande, como exigência parcial para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Joseffe Barroso de Oliveira

PRAIA GRANDE

2024

AGRADECIMENTOS

Gostaríamos de dedicar esta seção a todos que, de alguma forma, contribuíram para a realização deste projeto.

Agradecemos à faculdade, por disponibilizar a infraestrutura necessária ao longo do curso, para o aprendizado e o desenvolvimento acadêmico, aos professores que ministraram disciplinas essenciais para a nossa formação, contribuindo diretamente para que este projeto se tornasse possível.

Em especial, agradecemos à professora Simone, responsável pela disciplina de Banco de Dados. Além de transmitir com excelência os conhecimentos técnicos da área, foi uma pessoa atenciosa, compreensiva e acolhedora, oferecendo apoio tanto acadêmico quanto pessoal em momentos importantes.

Ao professor Joseffe, que atuou como orientador deste trabalho. Seu comprometimento e dedicação foram fundamentais para o desenvolvimento do projeto. Em diversas ocasiões, o professor foi além do esperado, oferecendo suporte técnico, incentivo e orientação, contribuindo significativamente para que pudéssemos concluir esta etapa com sucesso.

Por fim, expressamos nossa gratidão aos familiares, amigos e colegas que nos incentivaram durante toda a jornada. O apoio, a compreensão e as palavras de motivação foram essenciais para que não desistíssemos diante das dificuldades.

RESUMO

O “Script Forge” é um site que procura com o uso da inteligência artificial generativa desenvolver um roteiro para um jogo digital, o site utiliza a API oferecida pela empresa “Open AI” para integrar o “chat GPT” ao site, coletando uma inspiração entregue pelo usuário, então a inteligência artificial processa a informação e devolve uma ideia de roteiro para jogos digitais para o usuário.

A ideia do site é permitir que qualquer um possa desenvolver uma ideia básica e simples sem precisar se forçar ao extremo tentando ter uma ideia para seu roteiro, o site entrega uma ideia simples que pode ser expandido para uma grande história.

Palavras-Chave: Roteiro; Jogos digitais; Inteligência artificial.

ABSTRACT

“Script Forge” is a web application that seeks to use the generative artificial intelligence to create a script for a digital game, the site uses the “API” offered by “Open AI” to integrate “ChatGPT” into the site, it collects the inspirations offered by the user, then the artificial intelligence processes it and returns an idea for the digital game for the user.

The idea of the site is to allow anyone to develop a basic and simple idea without the need to waste too much effort trying to have an idea for the script, the site offers a simple idea that can be expanded to a grand history

Keywords: Script; Digital Game; artificial intelligence.

LISTA DE ABREVIações E SIGLAS

API	Application Programming Interface
CD	Código
CSS	Cascading Style Sheets
DS	Descrição
DT	Data
FK	Foreign Key
GPT	Generative Pre-trained Transformer
HTML	HyperText Markup Language
IA	Inteligência Artificial
IDE	Integrated Development Environment
IM	Imagem
NM	Nome
PHP	PHP: Hypertext Preprocessor
PK	Primary Key
POO	Programação Orientada a Objeto
RF	Requisito Funcional
RNF	Requisito Não Funcional
SG	Sigla
SQL	Structured Query Language
UML	Unified Modeling Language

LISTA DE FIGURAS

Figura 1 - Protótipo de baixa fidelidade 1.....	27
Figura 2 - Protótipo de baixa fidelidade 2.....	27
Figura 3 - Protótipo de alta fidelidade 1.....	28
Figura 4 - Protótipo de alta fidelidade 2.....	29
Figura 5 - Protótipo de alta fidelidade 3.....	30
Figura 6 - Protótipo de alta fidelidade 4.....	31
Figura 7 - Protótipo de alta fidelidade 5.....	31
Figura 8 - Protótipo de alta fidelidade 6.....	32
Figura 9 - Protótipo de alta fidelidade 7.....	32
Figura 10 – Caso de uso do sistema.....	43
Figura 11 - Diagrama de Classes.....	45
Figura 12 - Diagrama de sequência - login.....	47
Figura 13 - Diagrama de sequência - desenvolver roteiro.....	48
Figura 14 - Diagrama de sequência - salvar roteiro.....	49
Figura 15 - Diagrama de Entidade e Relacionamento.....	50
Figura 16 - DRAW IO.....	55
Figura 17 - FIGMA.....	57
Figura 18 - GitHub.....	58
Figura 19 - VISUAL STUDIO CODE.....	60
Figura 20 - XAMPP.....	61
Figura 21 - phpMyAdmin.....	62
Figura 22 - HTML.....	63
Figura 23 - CSS.....	64
Figura 24 - Bootstrap.....	65
Figura 25 - PHP.....	66
Figura 26 - DRAW IO.....	67
Figura 27 - JAVASCRIPT.....	68
Figura 28 - Chat GPT.....	70
Figura 29 - Banco ScriptForge.....	73
Figura 30 - Tabela Usuário.....	73
Figura 31 - tabela histórico.....	74
Figura 32 - Paginas 1.....	86
Figura 33 - Paginas 2.....	87
Figura 34 - Paginas 3.....	88
Figura 35 - Paginas 4.....	89
Figura 36 - Paginas 5.....	90
Figura 37 - Paginas 6.....	91
Figura 38 - Paginas 7.....	91

LISTA DE TABELAS

Tabela 1 - Tabela de requisitos.....	35
Tabela 2 - Dicionário de Dados da Tabela de Usuários.....	51
Tabela 3 - Dicionário de Dados da Tabela de Histórico.....	52

SUMÁRIO

1 INTRODUÇÃO	11
2 PROBLEMATIZAÇÃO DE PESQUISA	14
3 SOLUÇÃO E HIPÓTESE	16
3.1 OBJETIVO GERAL.....	17
3.2 OBJETIVOS ESPECÍFICOS.....	17
4 JUSTIFICATIVA	19
4.1 RELEVÂNCIA SOCIAL.....	21
5 METODOLOGIA	22
6 PLANEJAMENTO DO PROJETO	24
6.1 PROTOTIPAÇÃO.....	25
6.2 PROTÓTIPO DE BAIXA FIDELIDADE.....	26
6.3 PROTÓTIPO DE ALTA FIDELIDADE.....	28
6.4 ENTIDADES DO SISTEMA.....	33
6.5 REQUISITOS DO SISTEMA.....	34
6.5.1 REQUISITOS FUNCIONAIS	36
6.5.2 REQUISITOS NÃO FUNCIONAIS.....	40
6.6 DIAGRAMAS UML (LINGUAGEM DE MODELAGEM UNIFICADA).....	41
6.6.1 DIAGRAMAS DE CASO DE USO.....	42
6.6.2 DIAGRAMA DE CLASSES.....	44
6.6.3 DIAGRAMA DE SEQUÊNCIA.....	46
6.7 DIAGRAMA DE ENTIDADE E RELACIONAMENTO.....	50
7 TECNOLOGIAS E FERRAMENTAS	54
7.1 DRAW.IO.....	54
7.2 FIGMA.....	56
7.3 GITHUB.....	57
7.4 VISUAL STUDIO CODE.....	59
7.5 XAMPP.....	60
7.6 PHPMYADMIN.....	61
7.7 HTML.....	62
7.8 CSS.....	63
7.9 BOOTSTRAP.....	65
7.10 PHP.....	66
7.11 MYSQL.....	67
7.12 JAVASCRIPT.....	68
7.13 CHAT GPT.....	69
8 DESENVOLVIMENTO	71
8.1 BANCO DE DADOS.....	72
8.2 BACK-END.....	77
8.2.1 FUNÇÕES COMUNS.....	77
8.2.2 FUNÇÕES DO USUÁRIO.....	79

8.2.3 FUNÇÕES DO ADMINISTRADOR.....	80
8.2.4 FUNÇÕES DO HISTÓRICO.....	82
8.3 FRONT-END.....	84
8.3.1 PÁGINAS.....	85
8.3.2 CONSUMO DA API.....	92
9 CONSIDERAÇÕES FINAIS.....	94
REFERÊNCIAS.....	95

1 INTRODUÇÃO

O projeto intitulado “Script Forge” foi desenvolvido com o objetivo de explorar a aplicação da inteligência artificial generativa no contexto do desenvolvimento web. A proposta surgiu a partir de uma sugestão apresentada em sala de aula, que incentivava a criação de um sistema que integrasse IA em sua estrutura funcional. Motivados por essa ideia e pela afinidade dos integrantes com o universo dos jogos digitais, optou-se por construir uma aplicação web capaz de gerar roteiros para jogos de forma automatizada, utilizando recursos de IA.

A escolha dessa temática está diretamente relacionada aos desafios enfrentados durante o processo de criação de jogos, que envolvem tanto aspectos técnicos, como programação, design e a sonorização, além de desafios criativos, com destaque especial para o roteiro. A criação de uma narrativa envolvente e coerente é essencial para garantir a imersão do jogador e a qualidade da experiência de jogo. No entanto, esse processo costuma ser complexo, demandando tempo, criatividade e planejamento. Diante disso, o “Script Forge” foi idealizado como uma ferramenta de apoio, voltada à geração de roteiros simples e funcionais que possam auxiliar desenvolvedores, principalmente iniciantes, na etapa de criação narrativa.

Embora o foco inicial do projeto seja o apoio ao desenvolvimento de jogos digitais, o sistema também possui potencial para ser adaptado a outras mídias, como jogos de RPG de mesa, histórias em quadrinhos e qualquer outro formato que dependa de uma base narrativa estruturada. A proposta parte do princípio de que a IA pode atuar como catalisadora da criatividade, oferecendo sugestões e estruturas narrativas que sirvam como ponto de partida para o desenvolvimento humano.

Contudo, ressalta-se a importância do uso ético e consciente da inteligência artificial. Apesar de sua utilidade, ainda existem limitações técnicas e discussões relevantes, especialmente no que diz respeito à originalidade dos conteúdos gerados. Como apontado pela EBAC (2024), modelos de IA podem infringir direitos autorais ao utilizar, sem consentimento, materiais previamente existentes para gerar

novos conteúdos, o que levanta preocupações sobre plágio e propriedade intelectual. Assim, o “Script Forge” não busca substituir o trabalho humano, mas sim atuar como uma ferramenta de apoio, contribuindo para o processo criativo de forma responsável e colaborativa.

“Pessoalmente, sou bastante cético em relação ao uso da IA em videogames. Como ex-desenvolvedor, fico um pouco desconfortável ao ouvir sobre esse recurso sendo usado para escrever diálogos de personagens. Preocupo-me com resultados terrivelmente artificiais e com a possibilidade de os roteiristas perderem seus empregos.” (Scott J Mulligan, 2024).

Dessa forma, o objetivo do projeto é utilizar a inteligência artificial de maneira ética e responsável, atuando como uma ferramenta de apoio no desenvolvimento de jogos, sem desvalorizar ou substituir o trabalho de profissionais da área. Para reforçar essa postura, o próprio site conta com uma seção dedicada à conscientização dos usuários sobre o uso adequado da IA. Como destaca Daniel, em matéria publicada pela EBAC (2024), “a questão não é copiar uma sugestão da IA, mas utilizá-la como degrau para criar as próprias soluções criativas para o jogo”.

Este documento tem como finalidade registrar todas as etapas do desenvolvimento do projeto, desde o planejamento inicial até sua implementação prática. A proposta é oferecer uma visão clara e detalhada do processo, servindo como base tanto para análise quanto para futuras melhorias ou expansões do sistema. Nele, estão contidas informações sobre o que foi desenvolvido e o que está previsto para ser implementado nas próximas fases.

A estrutura deste trabalho está organizada em duas partes principais, o planejamento e o desenvolvimento. Na primeira parte, são apresentados o escopo do projeto, seus objetivos, requisitos funcionais e não funcionais, além do plano de negócio e das regras que orientam a aplicação. Também serão expostas representações visuais das telas do sistema, bem como diagramas técnicos que ilustram sua arquitetura, facilitando a compreensão global da proposta.

Na segunda parte, são descritas as etapas executadas durante o desenvolvimento, com as funcionalidades implementadas. Esse segmento também aborda os principais desafios encontrados, as estratégias utilizadas para superá-los e a documentação técnica relacionada à construção do sistema.

Essa abordagem detalhada visa garantir que todo o processo de criação do projeto “Script Forge” seja compreendido com clareza, permitindo que outros desenvolvedores ou interessados possam acompanhar, colaborar e expandir o projeto de forma alinhada aos seus propósitos originais.

2 PROBLEMATIZAÇÃO DE PESQUISA

Durante a fase de desenvolvimento de um jogo digital, é fundamental considerar diversos elementos que compõem o produto final. Entre esses elementos, o roteiro ocupa um papel essencial, sendo muitas vezes o diferencial entre um jogo medíocre e uma experiência de alta qualidade. No entanto, criar um roteiro envolvente e bem estruturado nem sempre é uma tarefa simples, especialmente em equipes pequenas, onde, na maioria das vezes, não há um roteirista especializado, tornando o desafio ainda maior.

Até mesmo em equipes maiores que possuem roteiristas dedicados e competentes, a criação de uma narrativa consistente pode ser um obstáculo. Nesse contexto, o “Script Forge” propõe uma solução prática: permitir que, a partir de uma ideia simples fornecida pelo usuário, a inteligência artificial gere automaticamente uma estrutura inicial de roteiro. Essa base pode, então, ser aprimorada e adaptada conforme a necessidade, facilitando o trabalho da equipe e contribuindo para o desenvolvimento narrativo do jogo.

Mesmo que muitos jogos não apresentem um roteiro complexo, ou sequer contem com uma narrativa explícita, ter ao menos uma ideia central de história é algo de grande relevância. Essa ideia pode servir como guia para diversas decisões de design, influenciando a direção artística, a trilha sonora e até as mecânicas do jogo. Por exemplo, uma história com tom melancólico pode inspirar a criação de ambientes mais sombrios e músicas emocionais, enquanto uma narrativa conceitual pode levar ao desenvolvimento de mecânicas inovadoras diretamente ligadas ao enredo.

É importante destacar, contudo, que o uso de ferramentas baseadas em inteligência artificial requer responsabilidade e ética. Embora a IA, como o ChatGPT, seja uma aliada poderosa no processo criativo, ela é frequentemente criticada por gerar conteúdo com base em padrões já existentes, sem necessariamente criar algo verdadeiramente inédito, conseqüentemente plagiando obras já existentes. Dessa

forma, o uso do Script Forge deve ser orientado para a inspiração e o desenvolvimento criativo, e não como substituição completa da autoria humana. O valor do roteiro gerado dependerá, principalmente, da capacidade do usuário de interpretar, adaptar e refinar o conteúdo produzido pela IA.

3 SOLUÇÃO E HIPÓTESE

Dado o problema descrito acima, propõe-se como solução a criação de uma ferramenta baseada em inteligência artificial capaz de auxiliar nesse processo criativo. O objetivo do site “Script Forge” é justamente esse, oferecer ao usuário um meio prático e acessível de transformar ideias iniciais em roteiros estruturados.

A plataforma utiliza a inteligência artificial do modelo ChatGPT, fornecido pela OpenAI, para interpretar uma ideia básica fornecida pelo usuário, como, por exemplo, “um jogo baseado em guerra nas estrelas”, e a partir disso, gerar uma proposta de roteiro coerente. O sistema é projetado para compreender e expandir temas simples, entregando uma narrativa inicial que pode ser desenvolvida e refinada posteriormente.

Além disso, o site permitirá ao usuário complementar e detalhar a proposta gerada, favorecendo a construção de histórias mais complexas e personalizadas. Mesmo aqueles com pouca experiência em escrita criativa poderão se beneficiar da ferramenta, utilizando os textos sugeridos como ponto de partida para a elaboração de roteiros completos.

A proposta contribui não somente para a democratização do processo de criação de roteiros, mas também para a otimização do tempo necessário para o desenvolvimento de jogos. Isso está alinhado à função da inteligência artificial como tecnologia de apoio, cuja finalidade é ampliar as capacidades humanas, e não as substituir.

É importante destacar que, apesar de seu potencial, a inteligência artificial deve ser compreendida como uma ferramenta complementar. O processo de criação exige a atuação ativa do usuário, que deve interpretar, adaptar e, quando necessário, revisar os conteúdos gerados. Dessa forma, garante-se a originalidade, a qualidade e a adequação do roteiro às necessidades específicas do projeto.

3.1 OBJETIVO GERAL

O projeto visa desenvolver um site que auxilie roteiristas na geração de ideias para a criação de roteiros, especialmente no contexto de jogos digitais, utilizando a inteligência artificial do modelo ChatGPT. A proposta consiste em permitir que o usuário forneça uma sugestão simples, que será interpretada e expandida pela IA, resultando em uma proposta inicial de roteiro.

A ideia central é reduzir o tempo necessário para a elaboração de roteiros, otimizando o processo de desenvolvimento de jogos. Além disso, a ferramenta visa complementar as ideias dos usuários, oferecendo suporte criativo e facilitando, de forma geral, o processo de construção narrativa.

3.2 OBJETIVOS ESPECÍFICOS

O objetivo é contribuir para o suporte de desenvolvedores independentes, especialmente em momentos de bloqueio criativo, uma dificuldade comum enfrentada por equipes pequenas. Em projetos com recursos e pessoal limitados, é comum que os esforços sejam divididos entre múltiplas áreas do desenvolvimento, o que pode restringir o tempo e a energia dedicados ao processo criativo, particularmente na elaboração de roteiros.

Esse cenário contrasta com o de grandes empresas do setor, que geralmente contam com equipes especializadas e múltiplos profissionais criativos, facilitando a produção de conteúdos originais e atrativos. Já os chamados jogos “indie”, desenvolvidos por equipes menores, enfrentam o desafio adicional de se destacar no mercado, seja por meio de inovações significativas ou pela abordagem criativa de temáticas já existentes.

Assim, o projeto “Script Forge” visa auxiliar diretamente nesse contexto, oferecendo uma ferramenta capaz de fornecer ideias iniciais ou sugestões de roteiro

a partir de comandos simples. Dessa forma, mesmo em momentos de esgotamento criativo, os pequenos desenvolvedores poderão contar com um apoio que lhes permita continuar produzindo, mantendo-se ativos e competitivos no mercado de jogos digitais.

4 JUSTIFICATIVA

O projeto se justifica pela necessidade de apoiar o processo criativo na área de desenvolvimento de jogos digitais, especialmente no que se refere à elaboração de roteiros. Ao propor uma ferramenta capaz de auxiliar os desenvolvedores na criação de narrativas, o projeto visa ampliar o potencial criativo dos profissionais da área, possibilitando o desenvolvimento de jogos com histórias mais elaboradas e originais, mesmo por equipes com recursos limitados.

Não só isso, mas a indústria de jogos, assim como outras mídias, passa por muitos ciclos de saturação. Sempre que um determinado estilo ou gênero alcança popularidade, uma abundância de produções com propostas semelhantes tende a surgir, gerando um excesso de títulos com ideias repetitivas. Esse fenômeno compromete a diversidade criativa e dificulta a inovação, afetando tanto os consumidores, que perdem o interesse por conteúdos saturados, quanto os desenvolvedores, que se sentem pressionados a seguir tendências em vez de explorar novos caminhos.

Casos emblemáticos ilustram essa dinâmica como, por exemplo, com o sucesso de jogos como “Resident Evil” gerou uma onda de títulos com temática zumbi, muitos dos quais pouco relacionados ao conceito original, como “Flat Out 3”, um jogo de corrida que incluiu modos com zumbis somente para se adaptar à tendência. Da mesma forma, o jogo “Dino Crisis”, embora com proposta distinta, foi inspirado na fórmula de sucesso de jogos de ação da época, repetindo o padrão observado em outras franquias influenciadas por filmes como *Jurassic Park*. Mais recentemente, observa-se esse mesmo comportamento com a popularização de gêneros como “Battle Royale” e jogos de terror inspirados no estilo gráfico dos anos 1990–2000, popularizados por desenvolvedores como o estúdio “Chilla’s Art”.

“Jogos com o tema simplesmente “jogados” ali de qualquer jeito, temos aos montes e até agradam [...], mas quando jogamos algo que conseguimos notar o cuidado de criar uma sinergia entre o tema e alguma mecânica, é mais satisfatório e acaba por ganhar sempre os melhores feedbacks.” (jean, 2021).

Nesse contexto, o “Script Forge” propõe-se como uma alternativa para reduzir a dependência de tendências saturadas, oferecendo aos desenvolvedores uma ferramenta que estimule a criação de roteiros originais. Com isso, busca-se não somente facilitar o processo criativo, mas também promover maior diversidade temática na produção de jogos digitais, contribuindo para um mercado mais inovador e sustentável.

Além disso, ao desenvolver um site que estimule a criatividade na criação de jogos digitais, amplia-se o acesso de novas pessoas interessadas à área de desenvolvimento. Muitos jovens aspiram tornar-se desenvolvedores, mas acabam desistindo ao se deparar com as inúmeras dificuldades do processo, especialmente no contexto de desenvolvimento independente. Entre essas dificuldades está a necessidade de lidar com múltiplas responsabilidades simultaneamente, como programação, design, áudio, marketing e, especialmente, a criação de roteiros.

“Como acontece com tudo na vida que vale a pena, muitos desafios e dificuldades aparecem no meio do caminho. E é aí que nos damos conta de que todos nós, em algum momento, pensamos em desistir.” (Ferreira, 2021).

A criação de uma ferramenta que simplifique a elaboração de roteiros contribui diretamente para tornar esse processo mais acessível, reduzindo a carga de trabalho e permitindo que desenvolvedores foquem em outras áreas igualmente importantes do projeto. Dessa forma, o projeto não apenas facilita a entrada de novos talentos no mercado, como também contribui para o fortalecimento e crescimento da indústria de jogos digitais, promovendo um ambiente mais inclusivo e menos restrito, onde o desenvolvimento de jogos possa ser mais do que um simples passatempo, mas uma possibilidade real de carreira.

4.1 RELEVÂNCIA SOCIAL

É inegável a crescente relevância dos jogos digitais no cotidiano das pessoas. A indústria de jogos vem se consolidando como uma das principais formas de entretenimento da atualidade, ocupando espaço ao lado de mídias tradicionais como o cinema e as séries. Cada vez mais, os jogos fazem parte da rotina de milhões de indivíduos que optam por essa mídia interativa para ocupar seu tempo livre, aprender, socializar ou se divertir.

Além do impacto cultural e social, a indústria de jogos também apresenta uma relevância econômica significativa. O setor tem crescido de forma expressiva, tornando-se mais acessível tanto para consumidores quanto para novos criadores, o que amplia as oportunidades de participação no mercado. Esse crescimento favorece o surgimento de novos talentos e o fortalecimento do ecossistema de desenvolvimento, contribuindo diretamente para a geração de empregos, inovação tecnológica e movimentação da economia de diversos países, como foi levantado por BUENO (2024) “Atualmente o Brasil é o maior mercado de games da América Latina e o 13º no ranking mundial, responsável por movimentar cerca de R\$ 12 bilhões ao ano, segundo projeções da PwC”.

Dessa forma, iniciativas que promovam a democratização do acesso às ferramentas de criação de jogos, como o “Script Forge”, têm um papel relevante não apenas na formação de novos desenvolvedores, mas também no fortalecimento de uma indústria criativa cada vez mais presente e influente na sociedade contemporânea.

5 METODOLOGIA

Este projeto se caracteriza como uma pesquisa aplicada, tendo como objetivo o desenvolvimento de uma ferramenta prática, um site que utiliza inteligência artificial para auxiliar na criação de roteiros para jogos digitais. A proposta envolve o uso da API do ChatGPT, fornecida pela OpenAI, para gerar ideias de roteiros com base em sugestões simples fornecidas pelos usuários.

A metodologia adotada não prevê uma análise quantitativa rigorosa afinal dados numéricos quanto ao tema são escassos, ao invés disso a avaliação será feita em dois momentos distintos, anteriormente à criação, com base em pesquisas e análise de projetos e artigos semelhantes que permitam prever o potencial de sucesso da proposta, e após à implementação, avaliando a recepção dos usuários, a usabilidade da ferramenta e se os objetivos foram efetivamente atingidos.

Portanto, a abordagem utilizada para a coleta e análise dos dados será do tipo qualitativa, uma vez que não há, no momento do desenvolvimento deste projeto, uma ampla base de dados ou estudos consolidados sobre o uso de inteligência artificial na geração de roteiros, especialmente na área de jogos digitais. Ainda que existam algumas iniciativas no campo do cinema e das séries, esse tipo de aplicação na indústria de jogos é ainda mais escassa, o que reforça a originalidade e relevância da proposta.

A escassez de referências acadêmicas e técnicas sobre o uso de IA na roteirização de jogos apenas evidencia a necessidade de mais estudos e projetos na área. É inegável que a inteligência artificial vem crescendo e ganhando espaço tanto no cotidiano das pessoas quanto nos ambientes de trabalho. Assim, este projeto também se propõe como uma contribuição inicial para um campo ainda pouco explorado, mas com alto potencial de desenvolvimento.

“A IA tem a capacidade de realizar tarefas rotineiras e não rotineiras, de modo que ela traz a possibilidade de substituir trabalhos realizados

exclusivamente por profissionais de alta qualificação, sendo este o aspecto que diferencia estas tecnologias dos outros processos de automação observados na história.” (apud BUENO, 2024).

“No mercado audiovisual, a IA já traz grandes transformações as dinâmicas do trabalho e pode trazer ainda mais. Da criação de roteiros à edição dos vídeos, passando pela mudança de cores em uma filmagem e a prática do deep fake, a inteligência artificial cria possibilidades totalmente novas das quais não podemos fugir.” (42 Filmes, 2023).

Embora o projeto não exija muitas ferramentas, será feito o melhor uso possível dos recursos disponíveis, de modo a garantir eficiência, qualidade no desenvolvimento e organização do código.

A principal IDE utilizada será o Visual Studio Code, por sua leveza, flexibilidade e ampla compatibilidade com extensões úteis para desenvolvimento web. Para o controle de versão e colaboração, será utilizado o GitHub, permitindo o acompanhamento do progresso do projeto e facilitando a organização das alterações de código.

O desenvolvimento da aplicação será feito com o uso de linguagens de programação baseado em web, como HTML e CSS, para a estruturação e estilização das páginas, JavaScript, para interação e dinamismo no front-end, além da integração direta com a API do chat GPT, PHP e Node.js, para a lógica de back-end e integração com o banco de dados, já o bando de dados usado será o MySQL.

Será utilizada também a biblioteca Bootstrap no front-end, com o objetivo de facilitar a criação de um layout responsivo e visualmente agradável, otimizando o tempo de desenvolvimento.

Por fim, serão consultadas amplamente as documentações oficiais da OpenAI, especialmente no que diz respeito ao uso da API do ChatGPT, garantindo uma integração eficaz e segura com os serviços de inteligência artificial que constituem o núcleo funcional do sistema.

6 PLANEJAMENTO DO PROJETO

O projeto procura principalmente desenvolver um site prático e intuitivo, capaz de ser utilizado por qualquer pessoa, de qualquer lugar. Para que isso seja possível, é fundamental não apenas a escolha de ferramentas eficazes, mas também o domínio dessas ferramentas, a fim de garantir o melhor aproveitamento dos recursos disponíveis.

A escolha pela criação de uma aplicação web foi feita com base em sua flexibilidade e acessibilidade, visto que esse tipo de aplicação pode ser acessado de qualquer lugar com conexão à internet e funciona bem em diferentes dispositivos. Além disso, as tecnologias escolhidas para o desenvolvimento são amplamente utilizadas no mercado, o que garante estabilidade, suporte e facilidade de aprendizado.

O desenvolvimento será conduzido com foco principal nas linguagens PHP e JavaScript. O PHP será responsável pelo funcionamento do back-end do site, com destaque para a conexão com o banco de dados, possibilitando funcionalidades como o login do usuário e o armazenamento do histórico de roteiros criados. Já o JavaScript será utilizado no front-end, realizando a integração com a API da OpenAI para a geração automática dos roteiros, além de permitir modificações dinâmicas na interface do usuário durante a navegação.

Complementando essas linguagens, serão utilizadas também HTML e CSS, que desempenham um papel essencial na estruturação e estilização do conteúdo exibido na interface do site. O HTML será responsável por organizar os elementos da página, enquanto o CSS permitirá aplicar estilos personalizados, garantindo uma apresentação visual adequada, além disso, o HTML permite a utilização de frameworks com foco em front end, pensando nisso, o framework do Bootstrap foi escolhido, como forma de agilizar o processo de desenvolvimento e garantir responsividade, afinal ele fornece componentes pré-configurados de fácil customização.

Em relação ao banco de dados, será utilizada a linguagem SQL por meio do sistema MySQLi, em conjunto com o PHP. Essa integração será fundamental para armazenar informações dos usuários e dos roteiros gerados de forma segura e eficiente.

Durante o processo de desenvolvimento, o ambiente de programação adotado será o Visual Studio Code, uma IDE leve e moderna que oferece suporte a diversas extensões e facilita a integração com outras ferramentas. O controle de versão será feito por meio do GitHub, que permitirá acompanhar a evolução do projeto, registrar alterações e colaborar de forma organizada, especialmente considerando que o projeto passou e ainda passará por diversas modificações ao longo do seu ciclo de vida.

Para a fase de planejamento visual e prototipação, duas ferramentas principais serão utilizadas. O Draw.io será empregado na criação de diagramas e protótipos de baixa fidelidade, sendo ideal para representar fluxos de navegação e esboços iniciais da estrutura do site e o Figma que será utilizado na construção do protótipo de alta fidelidade, permitindo simular com maior precisão a interface final do site, com foco na usabilidade e experiência do usuário.

Esse conjunto de ferramentas e metodologias foi cuidadosamente selecionado para garantir que o projeto alcance seus objetivos de forma eficiente, mantendo a simplicidade de uso para o público-alvo e a qualidade técnica necessária para seu bom funcionamento.

6.1 PROTOTIPAÇÃO

Durante a fase inicial do desenvolvimento do projeto, foi elaborada a prototipação da aplicação, etapa essencial para a visualização do produto final e para orientar o planejamento e a implementação das funcionalidades. A criação dos protótipos permitiu antecipar possíveis ajustes, validar ideias e facilitar a comunicação da estrutura do sistema.

Foram desenvolvidos dois tipos de protótipos, um de baixa fidelidade e outro de alta fidelidade. O protótipo de baixa fidelidade, criado por meio da ferramenta Draw.io, teve como objetivo fornecer uma visão geral e simplificada da disposição básica dos elementos da interface. Este protótipo serviu como base inicial para orientar a estrutura do site, sendo uma representação rápida e acessível para os primeiros testes de usabilidade e revisão de funcionalidades.

Em seguida, foi desenvolvido o protótipo de alta fidelidade utilizando a ferramenta Figma. Este segundo protótipo buscou representar de forma mais próxima à aparência final da aplicação, com detalhes visuais mais refinados, cores, tipografias e interações entre as páginas. O protótipo de alta fidelidade foi fundamental para compreender como as páginas se conectam entre si e como a experiência do usuário seria na versão final do site.

Ambos os protótipos contribuíram significativamente para a organização e execução do projeto, fornecendo um guia visual claro que auxiliou tanto na parte técnica quanto nas decisões de design e usabilidade da aplicação.

6.2 PROTÓTIPO DE BAIXA FIDELIDADE

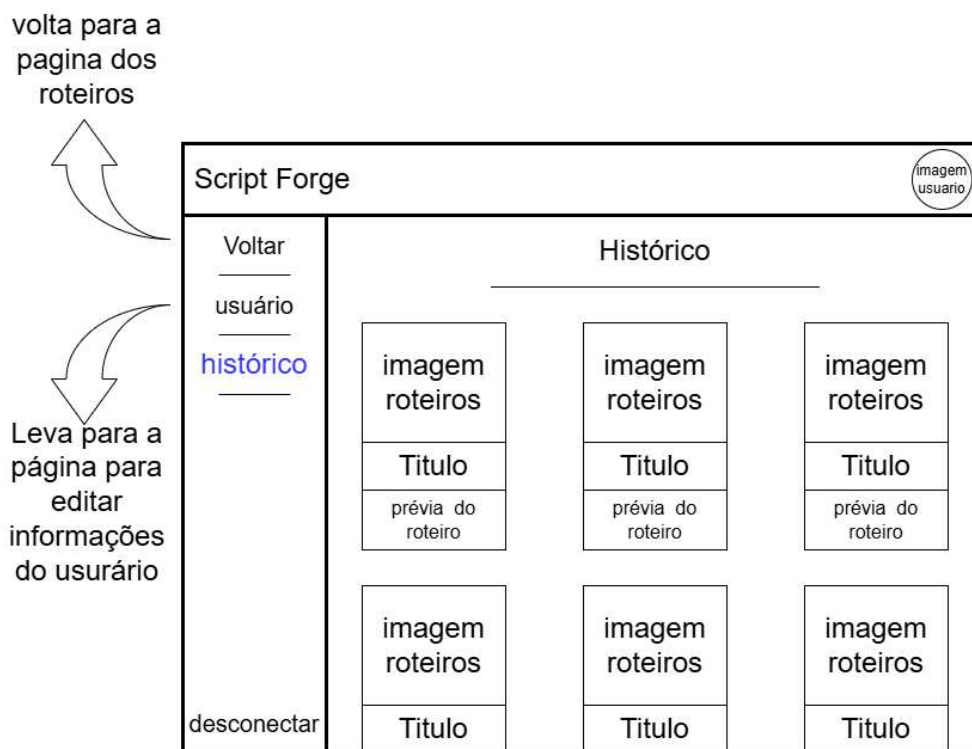
Esse protótipo foi desenvolvido utilizando a ferramenta Draw.io, que será empregada em outras etapas do projeto devido à sua versatilidade na criação de diagramas e fluxogramas. Nesta fase inicial, o objetivo do protótipo é manter-se simples, mas suficientemente claro para permitir o entendimento da estrutura básica da aplicação e servir como base para o direcionamento do desenvolvimento.

Figura 1 - Protótipo de baixa fidelidade 1



Fonte: Elaborado pelos autores, 2025.

Figura 2 - Protótipo de baixa fidelidade 2



Fonte: Elaborado pelos autores, 2025.

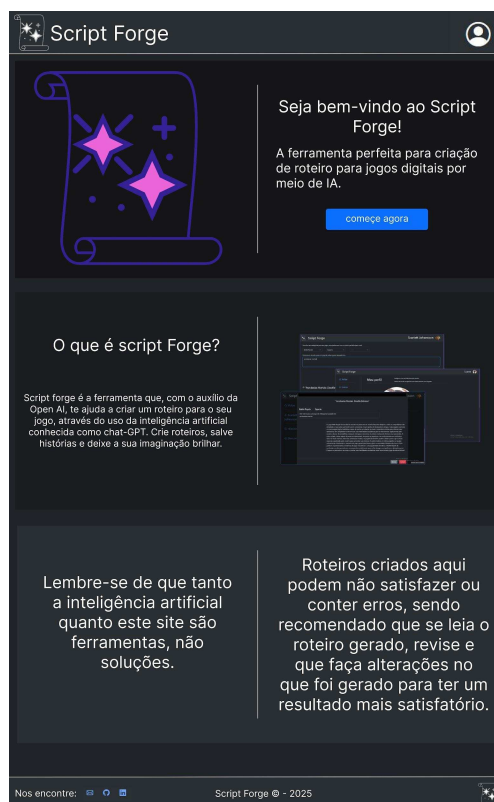
6.3 PROTÓTIPO DE ALTA FIDELIDADE

Assim como o protótipo de baixa fidelidade desenvolvido anteriormente, o protótipo de alta fidelidade também serve como um guia para o projeto, no entanto, este foi elaborado em paralelo à definição dos requisitos do sistema e à construção dos diagramas, sendo apresentado em sequência com o objetivo de facilitar a comparação entre os dois modelos.

O protótipo de alta fidelidade tem como propósito principal orientar a estilização da interface, além de representar as interações entre as páginas, como redirecionamentos e funcionalidades de botões. Também define elementos visuais importantes, como cores, textos e imagens, que serão utilizados na versão final do projeto.

Segue aqui as imagens dos protótipos de alta fidelidade:

Figura 3 - Protótipo de alta fidelidade 1.



Fonte: Elaborado pelos autores, 2025.

Figura 4 - Protótipo de alta fidelidade 2.

The image shows a dark-themed user interface for 'Script Forge'. At the top left is the logo, a scroll with stars and a plus sign, followed by the text 'Script Forge'. At the top right is a circular profile icon. The main content area is a large dark rectangle with a white border. Inside, the text 'Escolha uma categoria para criar seu roteiro' is followed by a dropdown menu showing '---' and a downward arrow. Below this is the text 'Escreva uma idéia para um roteiro de jogo:'. A large text input area contains placeholder text: 'Ut pharetra consectetur dapibus. Curabitur nec nunc nisl. Ut a purus sit amet dolor laoreet luctus. Integer lobortis augue ligula, ut ullamcorper turpis mollis sit amet. Suspendisse ut molestie odio.' Below the input area is a blue button with a pencil icon and the text 'Gerar roteiro'. Underneath is the heading 'Lorem ipsum' followed by another large text input area containing placeholder text: 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur ornare molestie interdum. Donec sed tristique eros. In cursus nec turpis sed varius. Curabitur mattis scelerisque massa, vel sodales dui euismod quis. Praesent accumsan ullamcorper augue. Cras enim lacus, dignissim vitae dignissim a, placerat semper ante. Duis scelerisque sapien sit amet turpis rutrum, vel convallis tellus vehicula. Duis sit amet lacus convallis, iaculis enim id, ornare erat. Maecenas ante erat, vehicula at egestas et, convallis ullamcorper neque. Aliquam erat volutpat. Praesent volutpat sed eros a mattis. Suspendisse erat mi, fringilla eget libero et, fringilla rhoncus arcu. Mauris condimentum venenatis nisi eu suscipit. Sed nec dolor nibh.' At the bottom of the main content area are two blue buttons: 'Elaborar' and 'Salvar'. The footer contains the text 'Nos encontre:' followed by icons for email, social media, and a logo, the text 'Script Forge © - 2025', and a small version of the 'Script Forge' logo on the right.

Script Forge

Escolha uma categoria para criar seu roteiro

Escreva uma idéia para um roteiro de jogo:

Ut pharetra consectetur dapibus. Curabitur nec nunc nisl. Ut a purus sit amet dolor laoreet luctus. Integer lobortis augue ligula, ut ullamcorper turpis mollis sit amet. Suspendisse ut molestie odio.

Gerar roteiro

Lorem ipsum

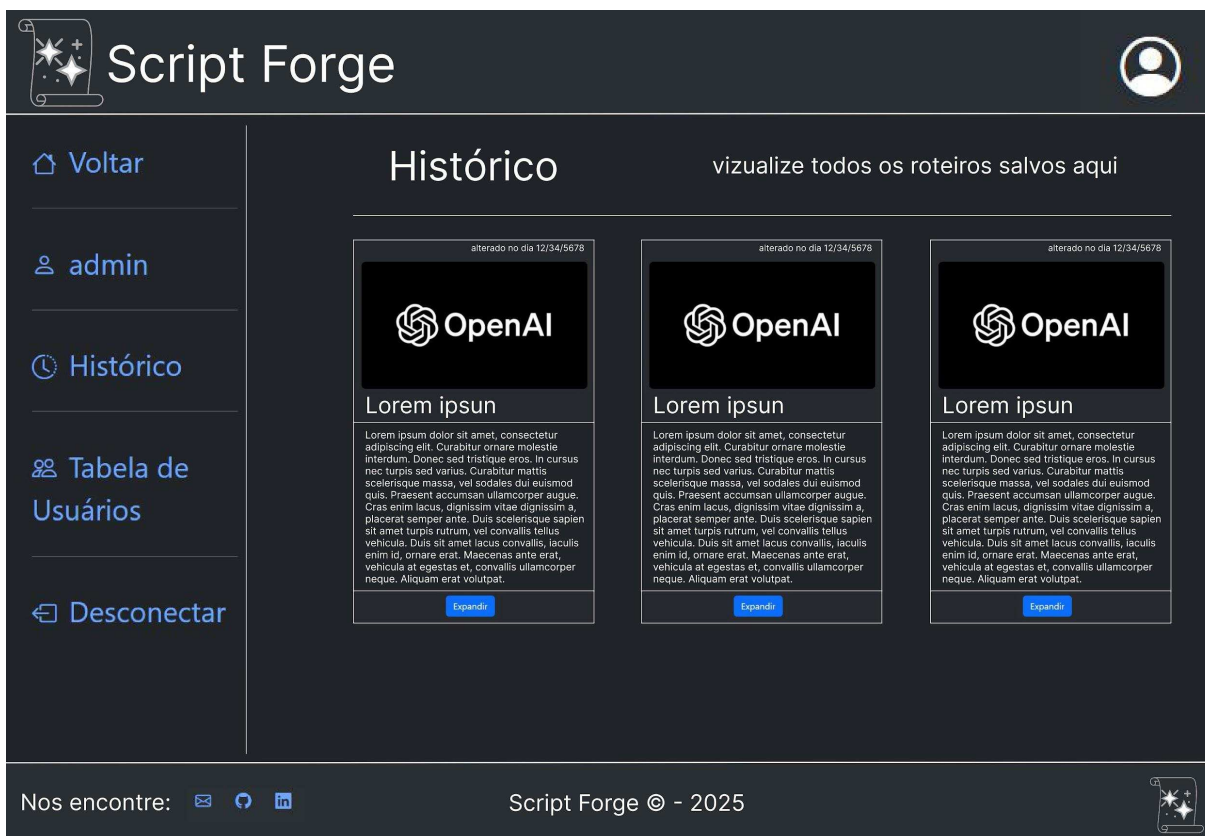
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur ornare molestie interdum. Donec sed tristique eros. In cursus nec turpis sed varius. Curabitur mattis scelerisque massa, vel sodales dui euismod quis. Praesent accumsan ullamcorper augue. Cras enim lacus, dignissim vitae dignissim a, placerat semper ante. Duis scelerisque sapien sit amet turpis rutrum, vel convallis tellus vehicula. Duis sit amet lacus convallis, iaculis enim id, ornare erat. Maecenas ante erat, vehicula at egestas et, convallis ullamcorper neque. Aliquam erat volutpat. Praesent volutpat sed eros a mattis. Suspendisse erat mi, fringilla eget libero et, fringilla rhoncus arcu. Mauris condimentum venenatis nisi eu suscipit. Sed nec dolor nibh.

Elaborar Salvar

Nos encontre: Script Forge © - 2025

Fonte: Elaborado pelos autores, 2025.

Figura 5 - Protótipo de alta fidelidade 3.



Fonte: Elaborado pelos autores, 2025.

Além disso, também foram planejadas as demais telas do projeto, como a tela de login, a tela de cadastro, a página do usuário e a tela onde o usuário poderá elaborar o texto em mais detalhes.

A seguir, são apresentadas as imagens das telas adicionais desenvolvidas:

Figura 6 - Protótipo de alta fidelidade 4.

The image shows a high-fidelity prototype of a login page for 'Script Forge'. The page has a dark theme. At the top left is the 'Script Forge' logo, which consists of a scroll icon with stars and a plus sign. At the top right is a user profile icon. The main content area is divided into two sections. On the left is a decorative graphic of a scroll with two pink stars and a plus sign. On the right is the login form, titled 'Faça seu login'. It includes an 'E-mail:' field with the placeholder 'nome@exemplo.com', a 'Senha:' field with the placeholder 'Senha', and a blue 'Entrar' button. Below the form, there is a link that says 'Não possui uma conta? [Cadastre-se aqui](#)'. At the bottom left, there are social media icons for email, WhatsApp, and LinkedIn, followed by the text 'Nos encontre:'. At the bottom center is the copyright notice 'Script Forge © - 2025'. At the bottom right is a small version of the scroll logo.

Fonte: Elaborado pelos autores, 2025.

Figura 7 - Protótipo de alta fidelidade 5.

The image shows a high-fidelity prototype of a registration page for 'Script Forge'. The page has a dark theme. At the top left is the 'Script Forge' logo, which consists of a scroll icon with stars and a plus sign. At the top right is a user profile icon. The main content area is divided into two sections. On the left is the registration form, titled 'Faça seu cadastro'. It includes a 'Username:' field with the placeholder 'Nome', an 'E-mail:' field with the placeholder 'nome@exemplo.com', a 'Senha:' field with the placeholder 'Senha', and a 'Repita sua senha:' field with the placeholder 'Senha'. Below the form is a blue 'Cadastrar' button. At the bottom of the form, there is a link that says 'Ja possui uma conta? [Conecte-se aqui](#)'. On the right is a decorative graphic of a scroll with two pink stars and a plus sign. At the bottom left, there are social media icons for email, WhatsApp, and LinkedIn, followed by the text 'Nos encontre:'. At the bottom center is the copyright notice 'Script Forge © - 2025'. At the bottom right is a small version of the scroll logo.

Fonte: Elaborado pelos autores, 2025.

Figura 8 - Protótipo de alta fidelidade 6.

Script Forge

Meu perfil Configure o seu perfil da forma que preferir.

Edite o seu perfil

Foto: Escolher arquivo Nenhum arquivo escolhido **Alterar**

Nome: admin **Alterar**

e-mail: admin@admin.com **Alterar**

senha: Senha Confirme sua senha **Alterar**
A senha deve ter ao menos 4 caracteres.

Excluir perfil

ATENÇÃO! Excluir o perfil fara com que você perca todas as suas informações, incluindo seus históricos salvos. Essa ação não pode ser desfeita. **Excluir**

Nos encontre: Script Forge © - 2025

Fonte: Elaborado pelos autores, 2025.

Figura 9 - Protótipo de alta fidelidade 7.

Script Forge

Painel do administrador visualize todos os roteiros salvos aqui

ID	Nome	Data Criado	Excluir
ID	Nome	Data Criado	Excluir
ID	Nome	Data Criado	Excluir
ID	Nome	Data Criado	Excluir
ID	Nome	Data Criado	Excluir
ID	Nome	Data Criado	Excluir
ID	Nome	Data Criado	Excluir
ID	Nome	Data Criado	Excluir

Nos encontre: Script Forge © - 2025

Fonte: Elaborado pelos autores, 2025.

É importante destacar que os protótipos não devem ser encarados como regras fixas a serem seguidas rigorosamente, mas sim como representações visuais que orientam o desenvolvimento do projeto. Durante a implementação, alguns elementos podem sofrer alterações, como o tamanho de componentes, textos e até mesmo cores, desde que o produto final mantenha uma semelhança coerente com o protótipo criado.

6.4 ENTIDADES DO SISTEMA

A definição das entidades de um sistema está diretamente relacionada aos papéis que cada tipo de usuário desempenha dentro da aplicação. Por exemplo, em um sistema hospitalar, um paciente não deve ter acesso às mesmas informações e funcionalidades que um médico ou enfermeiro. Por isso, é essencial planejar com clareza quem são as entidades e quais permissões cada uma delas terá.

No caso deste projeto, a definição das entidades é relativamente simples, pois o sistema não apresenta uma grande variedade de tipos de usuários ou uma hierarquia complexa. De modo geral, todos os usuários cadastrados compartilham o mesmo nível de permissão e funcionalidades básicas. No entanto, é necessário haver uma camada de moderação para manter a integridade da aplicação. Por isso, será criada uma entidade administrativa com permissões ampliadas.

Dessa forma, o sistema contará com duas entidades principais:

- **Usuário:** poderá utilizar todas as funcionalidades básicas da plataforma, como inserir ideias, gerar roteiros com auxílio da IA, salvar os roteiros em seu histórico, editar esse histórico e modificar informações do próprio perfil. Para isso, é necessário estar autenticado com uma conta.
- **Administrador:** terá todas as permissões do usuário comum, mas com funções adicionais voltadas à moderação e gestão do sistema. Isso inclui visualizar históricos de outros usuários, editar perfis de terceiros, apagar

contas que estejam utilizando o sistema de maneira indevida e manter o bom funcionamento da plataforma.

Essa estrutura simples e hierárquica torna a implementação mais direta, ao mesmo tempo que garante o controle necessário sobre a aplicação.

6.5 REQUISITOS DO SISTEMA

Os requisitos de um sistema representam, de forma geral, as regras, funcionalidades e objetivos que a aplicação deve atender para ser considerada completa. Eles são definidos durante as fases iniciais do projeto e têm como função descrever não apenas o que o sistema deve fazer, mas também como essas tarefas devem ser realizadas.

Sua definição é essencial, pois além de orientar o desenvolvimento, também garante que todos os envolvidos no projeto tenham uma visão clara e alinhada sobre as funcionalidades esperadas. Os requisitos são tradicionalmente divididos em duas categorias principais:

Requisitos funcionais que descrevem os comportamentos e funcionalidades que o sistema deve oferecer, como, por exemplo, permitir que o usuário crie, edite e salve roteiros.

Requisitos não funcionais que especificam restrições, padrões de qualidade ou critérios técnicos que devem ser seguidos durante o desenvolvimento, como desempenho, segurança, usabilidade e compatibilidade entre navegadores.

A correta definição e documentação desses requisitos é um passo fundamental para o sucesso de qualquer projeto de software, pois serve como guia e referência para toda a equipe de desenvolvimento.

“Requisitos de software são as descrições de serviços, funcionalidades e restrições que um sistema deve oferecer. [...] É no levantamento que elucidamos junto aos clientes e usuários do sistema quais são as necessidades da aplicação, portanto, é crucial que todos os envolvidos estejam alinhados sobre essas necessidades e que os requisitos estejam bem escritos e de fácil entendimento.” (FLORIANO, Jeronima, 2025).

A aplicação em desenvolvimento possui uma proposta simples, o que resulta em um número reduzido de requisitos levantados. No entanto, o cumprimento rigoroso desses requisitos é fundamental. Qualquer desvio ou falha na implementação pode comprometer o Resultado, a seguir os requisitos levantados:

Tabela 1 - Tabela de requisitos.

REQUISITOS FUNCIONAIS		
REQUISITO	NOME	DESCRIÇÃO
RF-01	Cadastrar Usuário	A aplicação deverá cadastrar um usuário.
RF-02	Apresentar usuário	A aplicação deverá apresentar os usuários cadastrados.
RF-03	Editar Usuário	A aplicação deverá editar os usuários cadastrados.
RF-04	Excluir Usuário	A aplicação deverá excluir os usuários que estão registrados no sistema.
RF-05	Enviar Prompt	A aplicação deverá enviar um prompt, através da API do chat GPT.
RF-06	Receber Resposta	A aplicação deverá receber uma resposta enviado através da API do chat GPT.
RF-07	Apresentar Resposta	A aplicação deverá apresentar a resposta recebida.
RF-08	Elaborar Resposta	A aplicação deverá enviar um prompt para elaborar a resposta recebida anteriormente.
RF-09	Salvar Resposta	A aplicação deverá armazenar a resposta recebida em um banco de dados.
RF-10	Apresentar Resposta	A aplicação deverá ser capaz de apresentar as respostas salvas no banco através do histórico.

	Salva	
RF-11	Editar Respostas Salvas	A aplicação deve ser capaz de editar uma resposta que foi salva anteriormente
RF-12	Apagar Resposta Salva	A aplicação deverá ser capaz de apagar a resposta salva do banco.
REQUISITOS NÃO FUNCIONAIS		
REQUISITO	NOME	DESCRIÇÃO
RNF-01	O Sistema deve ser Responsivo	A aplicação deverá ser capaz de manter suas características principais, independente de qual o dispositivo ou resolução utilizado pelo usuário.
RNF-02	Deve estar conectado ao Wi-fi	A aplicação precisa estar conectado a uma conexão Wi-Fi para poder funcionar corretamente.

6.5.1 REQUISITOS FUNCIONAIS

Como mencionado anteriormente, os requisitos funcionais são, em resumo, as regras e objetivos fundamentais que o sistema deve cumprir para poder ser considerado completo e satisfatório. Eles descrevem de forma clara as funcionalidades que a aplicação deve apresentar, ou seja, o que o sistema deve ser capaz de fazer. Esses requisitos orientam diretamente o processo de desenvolvimento, sendo essenciais para garantir que a solução atenda às necessidades do usuário e cumpra seu propósito principal. Cada requisito funcional representa uma ação, tarefa ou comportamento esperado da aplicação, compondo, assim, a base do funcionamento do sistema.

Aqui segue os requisitos funcionais de forma mais detalhada e concisa:

[RF001] CADASTRAR USUÁRIO

Descrição do caso de uso: O sistema deverá permitir o cadastro de novos usuários por meio de um formulário acessível na interface do site, esse cadastro deve coletar informações essenciais do usuário como email e senha, os dados deverão ser armazenados no banco de dados.

Prioridade: Média

Usabilidade: Normal

Desempenho: Médio

Situação: Incluído

Segurança: Alta

[RF002] APRESENTAR USUÁRIO

Descrição do caso de uso: O sistema deverá ser capaz de exibir uma listagem dos usuários cadastrados, apresentando informações relevantes como e-mail, o administrador poderá ver outros usuários cadastrados.

Prioridade: Média

Usabilidade: Normal

Desempenho: Médio

Situação: Incluído

Segurança: Alta

[RF003] EDITAR USUÁRIO

Descrição do caso de uso: O sistema deverá permitir a edição das informações dos usuários previamente cadastrados. A funcionalidade estará disponível para o próprio usuário editar seus dados. O administrador poderá editar informações sobre outros usuários.

Prioridade: Baixa

Usabilidade: Normal

Desempenho: Médio

Situação: Incluído

Segurança: Alta

[RF004] EXCLUIR USUÁRIO

Descrição do caso de uso: O sistema deverá permitir a exclusão de usuários previamente registrados. Essa ação poderá ser executada pelo próprio usuário, ao solicitar a exclusão de sua conta. O administrador poderá excluir outras contas.

Prioridade: Médio

Usabilidade: Baixa

Desempenho: Médio

Situação: Incluído

Segurança: Alta

[RF005] ENVIAR PROMPT

Descrição do caso de uso: O sistema deverá permitir que o usuário envie um prompt de texto para a API do ChatGPT.

Prioridade: Alta

Usabilidade: Alta

Desempenho: Alto

Situação: Incluído

Segurança: Média

[RF006] RECEBER RESPOSTA

Descrição do caso de uso: O sistema deverá ser capaz de receber uma resposta textual da API do ChatGPT em resposta ao prompt enviado pelo usuário.

Prioridade: Alta

Usabilidade: Alta

Desempenho: Alto

Situação: Incluído

Segurança: Alta

[RF007] APRESENTAR RESPOSTA

Descrição do caso de uso: O sistema deverá apresentar ao usuário, de forma clara e organizada, a resposta recebida da API do ChatGPT baseado no prompt enviado.

Prioridade: Alta

Usabilidade: Alta

Desempenho: Alto

Situação: Incluído

Segurança: Média

[RF008] ELABORAR RESPOSTA

Descrição do caso de uso: O sistema deverá permitir o envio de um novo prompt com base na resposta recebida anteriormente, elaborando ou expandindo o conteúdo.

Prioridade: Média

Usabilidade: Alta

Desempenho: Alto

Situação: Incluído

Segurança: Média

[RF009] SALVAR RESPOSTA

Descrição do caso de uso: O sistema deverá permitir que o usuário salve a resposta recebida da API do ChatGPT em seu histórico pessoal, armazenando essa informação de forma segura no banco de dados para consultas e edições futuras.

Prioridade: Média

Usabilidade: Normal

Desempenho: Médio

Situação: Incluído

Segurança: Alta

[RF010] APRESENTAR RESPOSTA SALVA

Descrição do caso de uso: O sistema deverá permitir que o usuário visualize as respostas previamente salvas em seu histórico pessoal, apresentando-as de forma organizada e acessível na área destinada ao histórico.

Prioridade: Média

Usabilidade: Normal

Desempenho: Médio

Situação: Incluído

Segurança: Média

[RF011] EDITAR RESPOSTAS SALVAS

Descrição do caso de uso: O sistema deverá permitir que o usuário edite uma resposta salva anteriormente no histórico, possibilitando elaboração do conteúdo com base em novas ideias e respostas enviadas pela API.

Prioridade: Baixa

Usabilidade: Normal

Desempenho: Alto

Situação: Incluído

Segurança: Média

[RF012] APAGAR RESPOSTA SALVA

Descrição do caso de uso: O sistema deverá permitir que o usuário exclua uma resposta salva do histórico.

Prioridade: Média

Usabilidade: Normal

Desempenho: Médio

Situação: Incluído

Segurança: Média

6.5.2 REQUISITOS NÃO FUNCIONAIS

Assim como os requisitos funcionais, os requisitos não funcionais também representam regras que o sistema deve cumprir para poder ser considerado completo e satisfatório. No entanto, sua principal função é estabelecer limitações, padrões e diretrizes que influenciam diretamente como os requisitos funcionais serão implementados. Esses requisitos abrangem aspectos como desempenho, usabilidade, escalabilidade, compatibilidade e, principalmente, segurança e responsividade. Por exemplo, um requisito não funcional pode estipular que a aplicação seja responsiva, ou seja, que mantenha a estrutura e usabilidade adequadas independentemente do dispositivo utilizado, seja ele um computador, tablet ou smartphone. Além disso, requisitos relacionados à segurança garantem que os dados dos usuários estejam protegidos contra acessos não autorizados ou vazamentos. Dessa forma, os requisitos não funcionais são fundamentais para garantir a qualidade e a confiabilidade do sistema todo.

A seguir a lista detalhada dos requisitos não funcionais:

[RNF001] O SISTEMA DEVE SER RESPONSIVO

Descrição do requisito: O sistema deverá ser responsivo, ou seja, deve manter sua estrutura visual e usabilidade independente do dispositivo ou resolução de tela utilizada pelo usuário (como computadores, tablets ou smartphones).

Prioridade: Alta

Usabilidade: Alta

Desempenho: Baixo

Situação: Incluído

Segurança: Baixa

[RNF002] Deve estar conectado ao Wi-fi

Descrição do requisito: A aplicação deverá funcionar somente quando conectada a uma rede de internet estável. Para garantir o funcionamento correto da integração com a API do ChatGPT, é necessário que o sistema esteja conectado à internet, preferencialmente via Wi-Fi ou conexão estável equivalente.

Prioridade: Alta

Usabilidade: alta

Desempenho: Baixo

Situação: Incluído

Segurança: Baixa

6.6 DIAGRAMAS UML (LINGUAGEM DE MODELAGEM UNIFICADA)

Após levantar os requisitos e definir as entidades do sistema, o próximo passo para o desenvolvimento da aplicação é a definição dos diagramas UML, esses diagramas, mais uma vez, têm como principal objetivo servir como um guia para os desenvolvedores, mostrando de forma clara e visual como o sistema deverá funcionar. Eles ajudam a estruturar as funcionalidades, definir o que cada entidade pode ou não fazer, e até mesmo descrever os atributos e comportamentos dos objetos no sistema. Com esses diagramas bem definidos, torna-se muito mais fácil organizar o desenvolvimento e garantir que tudo esteja conforme o planejado.

Em resumo, os diagramas UML são ferramentas criadas com o intuito de oferecer uma visão clara e estruturada do funcionamento de um sistema, com o objetivo de auxiliar os desenvolvedores no cumprimento dos requisitos previamente definidos. Entre os principais diagramas utilizados estão o Diagrama de Casos de Uso, o Diagrama de Classes, o Diagrama de Sequência, entre outros. Cada um desses diagramas possui regras específicas para sua construção e interpretação, sendo essenciais para garantir o bom planejamento, entendimento e desenvolvimento de um projeto de software.

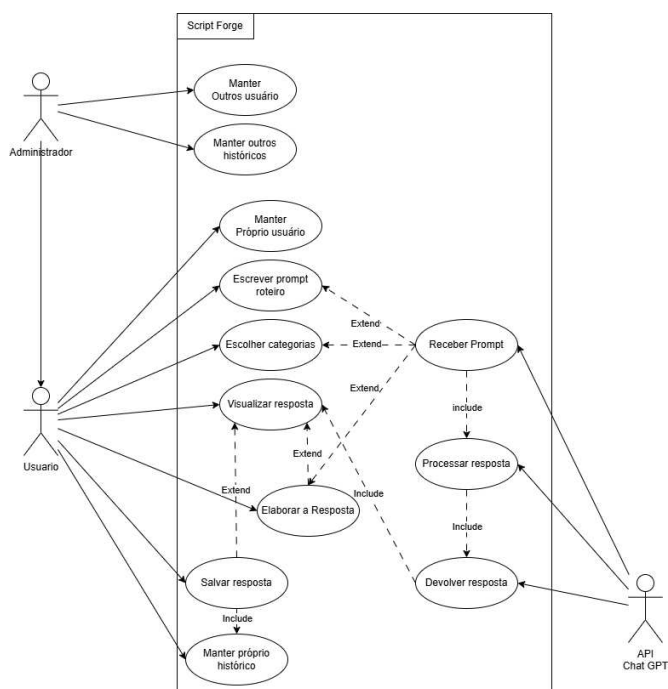
De acordo com informações do site LucidChart (2025), “A UML representa as melhores práticas para desenvolver e documentar aspectos diferentes da modelagem de software e sistemas de negócios”. Dessa forma, o sistema em desenvolvimento possui uma estrutura simples, o que torna desnecessária a criação de muitos diagramas UML. Por esse motivo, optou-se por utilizar apenas os diagramas mais essenciais para a representação das funcionalidades e da estrutura do sistema. Os principais escolhidos foram o Diagrama de Casos de Uso, o Diagrama de Classes e Diagrama de Sequências.

A seguir, serão apresentados os diagramas desenvolvidos, com o uso da ferramenta do DRAW.IO, acompanhados de explicações detalhadas sobre cada um deles, visando esclarecer sua função e como representam os aspectos do sistema.

6.6.1 DIAGRAMAS DE CASO DE USO

O diagrama de caso de uso, como mencionado anteriormente, é um dos diagramas UML essenciais para o desenvolvimento de um projeto. Ele tem como base as entidades previamente definidas, além de considerar determinadas partes do sistema como atores, a fim de orientar as funcionalidades que o sistema deve executar. Este diagrama será posteriormente complementado pelo diagrama de sequência, que detalhará a interação entre os elementos representados.

Figura 10 – Caso de uso do sistema.



Fonte: Elaborado pelos autores, 2025.

Como pode ser observado, o diagrama de caso de uso foi desenvolvido com base na definição prévia das entidades do sistema. Cada usuário realiza ações que correspondem às permissões descritas anteriormente, no entanto, a entidade chamada “API Chat GPT” aparece presente, embora não tenha sido listada entre os usuários do sistema, atua como uma entidade externa fundamental, representando a própria integração com o modelo de linguagem.

As tarefas descritas no diagrama são diretamente derivadas dos requisitos funcionais porém é possível notar o termo "manter", que se refere ao conjunto de ações que envolvem cadastrar, editar, visualizar e excluir, seja do próprio usuário, de outros usuários (no caso do administrador) ou do histórico pessoal de respostas.

As relações de “include” e “extend” também estão presentes no diagrama e ajudam a identificar ações obrigatórias e ações opcionais, respectivamente. Por exemplo:

- A ação “Receber prompt” atuada pela API é opcional, afinal o usuário pode escrever o texto, escolher a categoria e ainda decidir se deseja ou não o

enviar, por isso, trata-se de um extend, uma funcionalidade condicional.

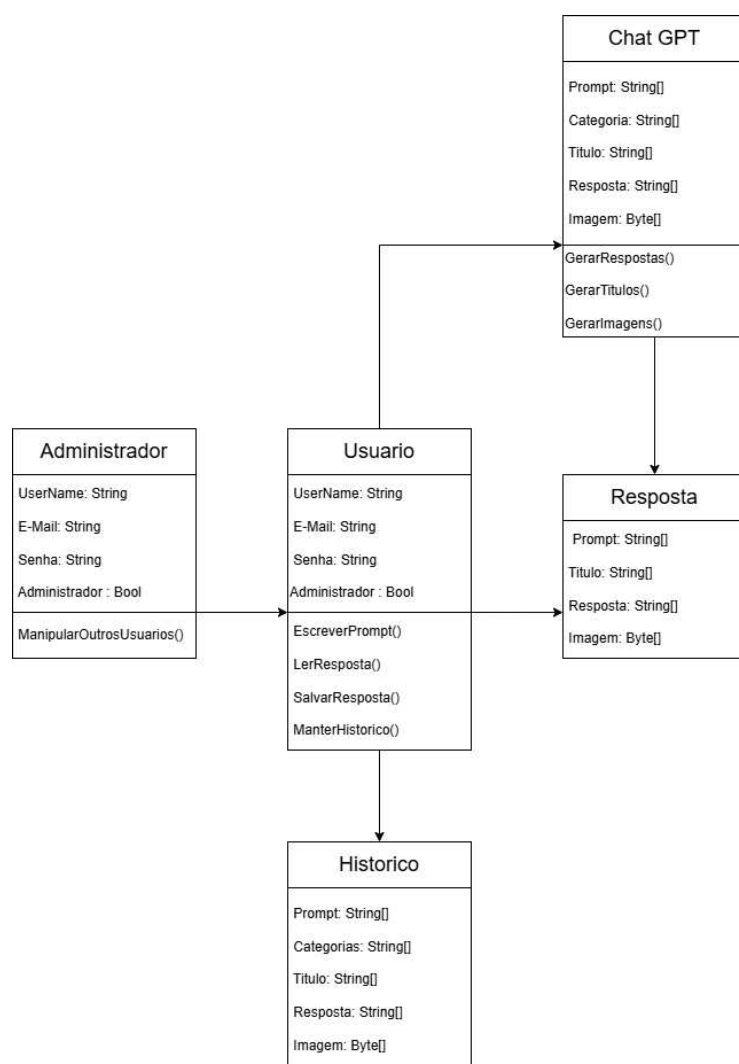
- Por outro lado, uma vez que o prompt é enviado, a API obrigatoriamente o processa e envia uma resposta (ainda que essa resposta seja uma mensagem de erro). Esse comportamento representa um include, pois ocorre sempre que o prompt é enviado.

Além disso, é possível identificar no diagrama uma sequência recursiva de ações, ou seja, um fluxo que pode se repetir indefinidamente. Essa sequência é composta pelas funções “Receber Prompt”, “Processar Prompt” “Enviar Resposta” “Visualizar Resposta” “Elaborar Resposta”, esse ciclo existe porque uma das funcionalidades principais do sistema é permitir que o usuário, ao visualizar a resposta recebida, possa enviar um novo prompt com base nela, refinando ou expandindo o conteúdo até atingir o resultado desejado.

6.6.2 DIAGRAMA DE CLASSES

O diagrama de classes possui uma estrutura visual semelhante a um esquema de banco de dados, no entanto, seu foco vai além da organização dos dados. Seu principal objetivo é representar as classes do sistema, seus atributos, métodos e as relações entre elas, servindo como um guia fundamental no desenvolvimento orientado a objetos.

Figura 11 - Diagrama de Classes



Fonte: Elaborado pelos autores, 2025.

Como pode ser observado, algumas classes no diagrama são separadas das entidades principais, como as classes Resposta, ChatGPT e Histórico, isso ocorre porque, embora não sejam entidades do sistema com funções autônomas ou interativas como o Usuário ou Administrador, elas ainda representam componentes importantes no funcionamento da aplicação, sendo objetos que precisam ser modelados e manipulados durante a execução do sistema.

Esse diagrama é construído com base em uma visão orientada a objetos, também conhecida como POO (Programação Orientada a Objetos). Através dele, é

possível compreender quais objetos o sistema manipula, quais atributos e métodos cada classe possui, e como essas classes se relacionam entre si.

Em resumo, o diagrama de classes fornece um guia essencial para a estruturação do código, auxiliando o desenvolvedor a entender quem interage com quem, quais dados estão disponíveis em cada parte do sistema e como esses dados são processados.

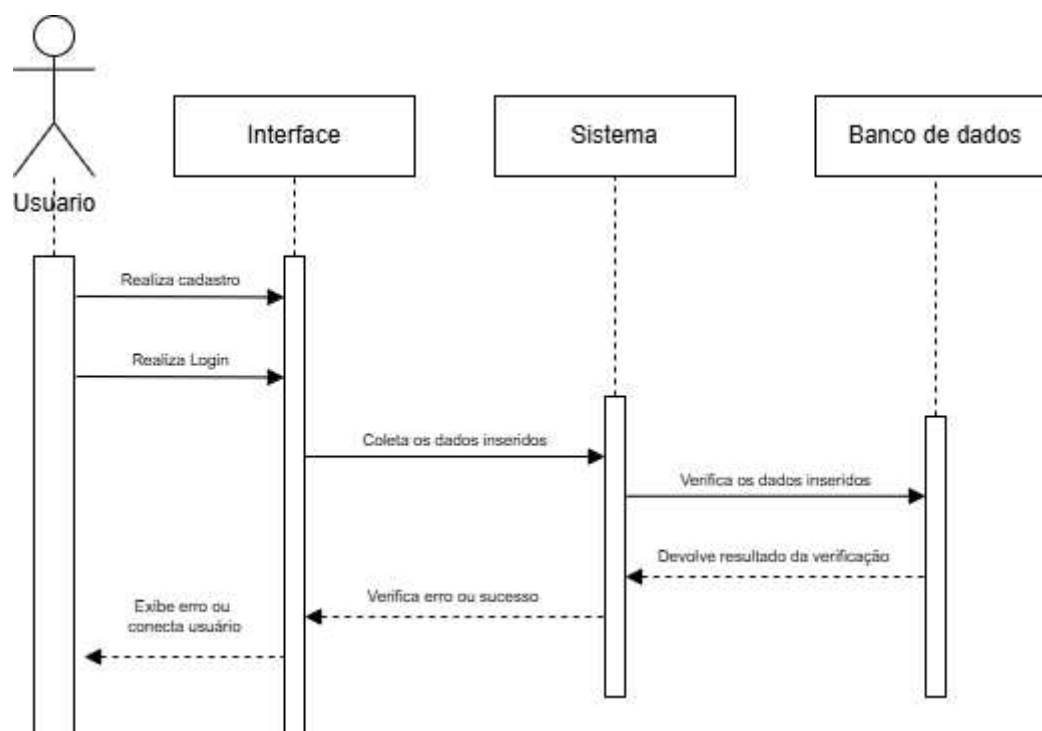
6.6.3 DIAGRAMA DE SEQUÊNCIA

O diagrama de sequência visa representar graficamente a ordem e o fluxo das interações entre os componentes do sistema ao longo da execução de um processo específico. Este tipo de diagrama é especialmente útil para complementar o diagrama de caso de uso, pois detalha a lógica por trás de cada funcionalidade apresentada.

No contexto deste projeto, foram elaborados diagramas de sequência para os principais processos da aplicação, a fim de evidenciar a dinâmica das operações e o papel de cada elemento envolvido, os processos representados incluem o fluxo completo de login do usuário, o processo de criação e elaboração de roteiros, e o procedimento para salvamento dos roteiros gerados.

Cada diagrama será acompanhado por uma legenda explicativa, a qual tem a finalidade de auxiliar na interpretação correta das ações descritas, garantindo assim uma compreensão clara e precisa do funcionamento interno do sistema.

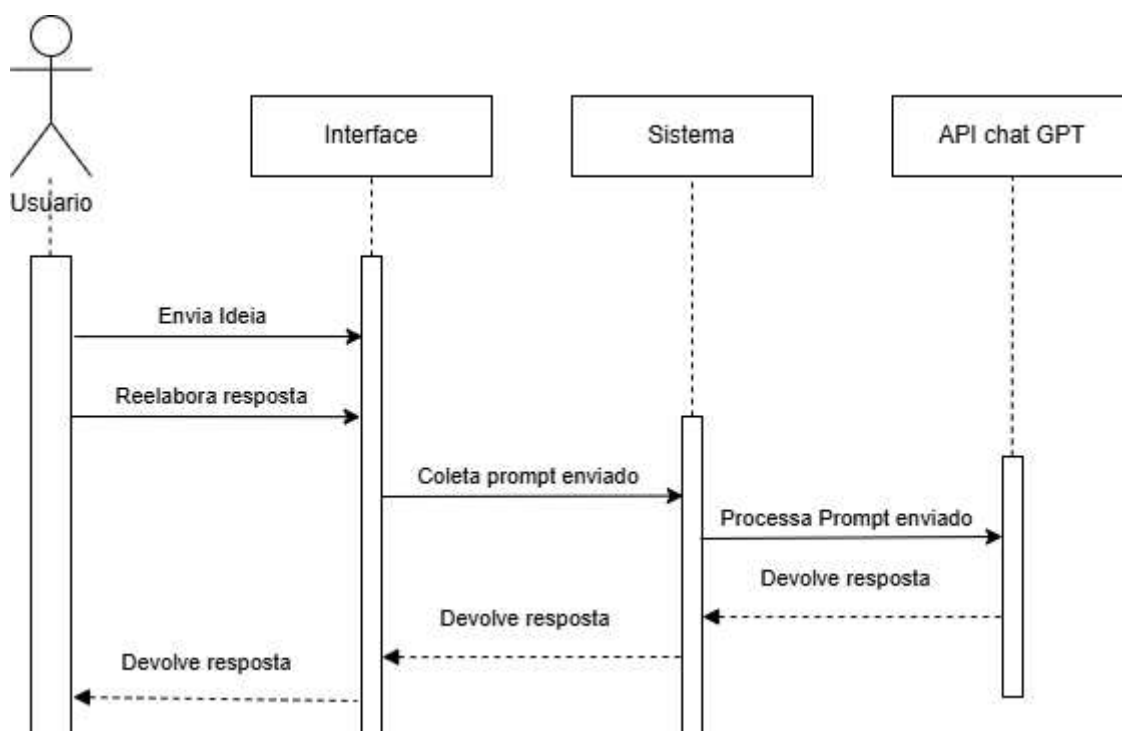
Figura 12 - Diagrama de sequência - login



Fonte: Elaborado pelos autores, 2025.

Como apresentado, o diagrama de sequência representa o fluxo completo para o processo de login do usuário. Inicialmente, o usuário insere suas credenciais, como e-mail e senha, ou realiza uma solicitação de cadastro. Esses dados são coletados pela interface do sistema e enviados ao servidor, que os repassa ao banco de dados para validação. No caso do login, o banco verifica se o e-mail está cadastrado e se a senha informada corresponde a registrada. Após a verificação, o banco de dados retorna uma resposta ao servidor, que, por sua vez, informa ao usuário se o acesso foi concedido ou se houve erro na autenticação, como credenciais inválidas ou e-mail não cadastrado.

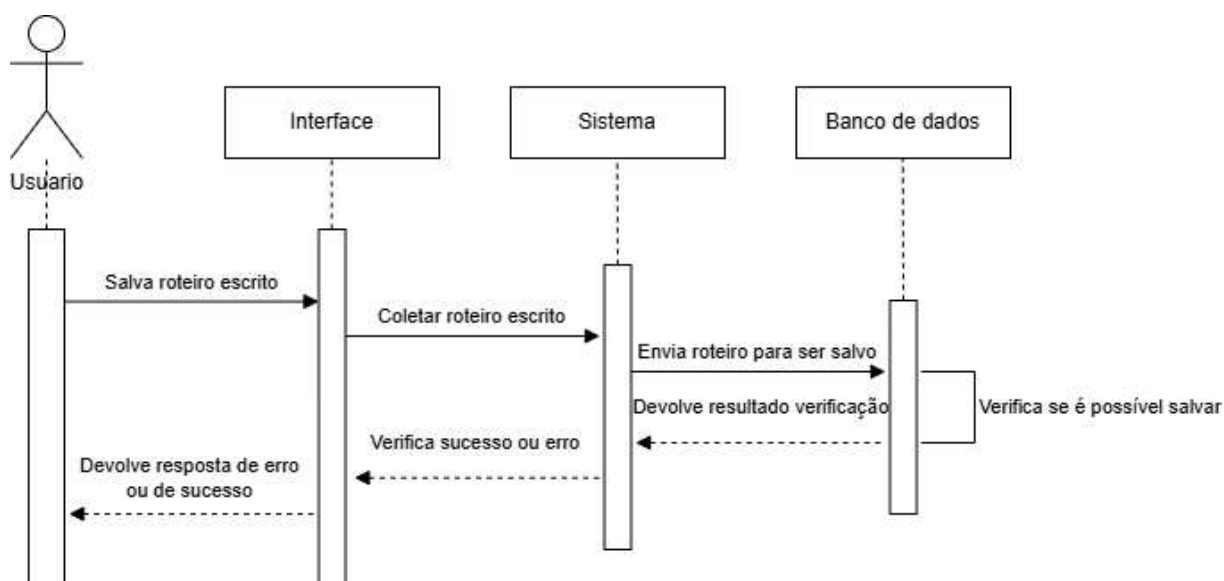
Figura 13 - Diagrama de sequência - desenvolver roteiro



Fonte: Elaborado pelos autores, 2025.

Conforme mencionado anteriormente, o diagrama apresentado representa a sequência de eventos envolvida no envio de um prompt pelo usuário à aplicação, seja para a geração de uma nova ideia ou para a elaboração de um texto previamente criado. O processo tem início quando o usuário insere e envia um prompt por meio da interface do sistema. O sistema, então, coleta os dados informados e os encaminha para a API do ChatGPT. A API realiza o processamento da solicitação, gera uma resposta com base no conteúdo enviado e a retorna ao sistema. Por fim, o sistema recebe essa resposta e a apresenta ao usuário. A resposta pode variar entre um roteiro elaborado, uma sugestão de conteúdo ou até uma mensagem de erro, caso haja falha no processamento. Essa mesma sequência se aplica ao caso em que o usuário opta por elaborar um texto já existente.

Figura 14 - Diagrama de sequência - salvar roteiro



Fonte: Elaborado pelos autores, 2025.

O diagrama apresentado representa a sequência de eventos relacionados ao processo de salvamento de um roteiro gerado pela aplicação, conforme descrito anteriormente. Embora esse fluxo possa ser representado em conjunto com o diagrama que descreve a geração do roteiro, optou-se por sua separação com o objetivo de proporcionar uma visualização mais clara e didática.

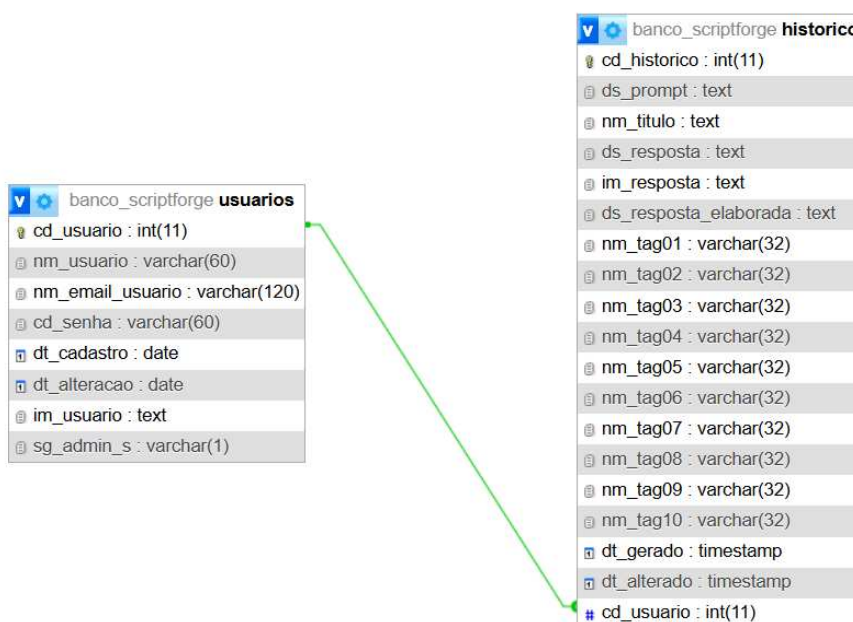
Neste processo, o usuário realiza uma solicitação para salvar a resposta obtida a partir do prompt enviado à API. O sistema, então, coleta as informações pertinentes, como o título, a resposta gerada, o prompt original e as categorias associadas. Em seguida, essas informações são encaminhadas ao banco de dados, que verifica a viabilidade da operação e realiza o salvamento. Após essa etapa, o banco de dados retorna uma resposta ao sistema, que a processa e notifica o usuário sobre o sucesso ou eventual falha da operação.

6.7 DIAGRAMA DE ENTIDADE E RELACIONAMENTO

Externo aos diagramas UML o Diagrama de Entidade e Relacionamento é um diagrama cujo objetivo é representar visualmente o relacionamento entre as entidades do sistema sob a perspectiva do banco de dados. Ele é fundamental para o planejamento da estrutura de armazenamento de dados, permitindo visualizar como as informações se conectam.

Para a criação deste diagrama, foi utilizada uma funcionalidade do phpMyAdmin, uma ferramenta de hospedagem de servidores local, que permite exportar o banco de dados desenvolvido em forma de diagrama. No entanto, vale destacar que também é possível desenvolvê-lo com o MySQL Workbench, uma ferramenta especializada e amplamente utilizada para a criação de diagramas de entidade e relacionamento, especialmente em projetos que utilizam bancos de dados relacionais.

Figura 15 - Diagrama de Entidade e Relacionamento



Fonte: Elaborado pelos autores, 2025.

O diagrama apresentado mostra a estrutura do banco de dados e utiliza a notação conhecida como “pé de galinha”. Nessa notação, os elementos ou entidades são conectados por linhas com marcações que representam o tipo de relacionamento entre eles.

No caso em questão, o relacionamento apresentado é do tipo um para muitos (1:N). Isso significa que um usuário pode ter vários históricos salvos, mas cada histórico pertence a apenas um usuário.

Apesar de simples, esse tipo de diagrama deve seguir algumas regras importantes que dizem respeito ao banco de dados. Por exemplo:

- Toda entidade deve possuir uma chave primária, que é um identificador único e exclusivo. É comum utilizar o Mnemônico CD (de "código") nesses casos.
- Em relacionamentos do tipo um para muitos, a entidade do lado "muitos" deve conter a chave primária da outra entidade como chave estrangeira.

No diagrama apresentado, isso é refletido da seguinte forma: a tabela histórico contém a chave estrangeira `cd_usuario`, que faz referência à chave primária da tabela usuário. Essa estrutura permite identificar facilmente a quem pertence cada histórico salvo.

A seguir será apresentado o dicionário dos dados, uma tabela que apresenta os dados e os explica de forma mais simples:

Tabela 2 - Dicionário de Dados da Tabela de Usuários

Tabela	Usuarios			
Descrição	Armazena todos os usuários cadastrados			
Observações				
CAMPOS				
NOME	DESCRIÇÃO	TIPO DE DADO	TAMANHO	RESTRIÇÕES
cd_usuario	Código de identificação de usuário	Int	11	PK / Identity

nm_usuario	Nome do usuário	Varchar	60	Not Null
nm_email_usuario	Email do Usuário	Varchar	120	Unique / Not Null
cd_senha	Senha do Usuário	Varchar	60	Not Null
dt_cadastro	Data que o usuário se cadastrou	Date		Not Null / Default
dt_cadastro	Data que o usuário alterou informações de cadastro	Date		Not Null / Default
im_usuario	Imagem apresentada no perfil do usuário	Text		Default
sg_admin_s	Define se o usuario é um administrador ou não	Bool	1	Default

Tabela 3 - Dicionário de Dados da Tabela de Histórico

Tabela	Histórico			
Descrição	Armazena todos os roteiros salvos pelos usuários			
Observações	Possui uma chave estrangeira da tabela usuário			
CAMPOS				
NOME	DESCRIÇÃO	TIPO DE DADO	TAMANHO	RESTRIÇÕES
cd_historico	Código de identificação de cada roteiro	Int	11	PK / Identity
ds_prompt	Registra o prompt escrito pelo usuario	Text		
nm_titulo	Nome do titulo criado para o roteiro	Text		
ds_resposta	Resposta enviada pela API	Text		
im_resposta	Imagem gerada pela API	Text		Default
nm_tag01	Categoria do roteiro	Varchar	32	

nm_tag02	Categoria do roteiro	Varchar	32	
nm_tag03	Categoria do roteiro	Varchar	32	
nm_tag04	Categoria do roteiro	Varchar	32	
nm_tag05	Categoria do roteiro	Varchar	32	
nm_tag06	Categoria do roteiro	Varchar	32	
nm_tag07	Categoria do roteiro	Varchar	32	
nm_tag08	Categoria do roteiro	Varchar	32	
nm_tag09	Categoria do roteiro	Varchar	32	
nm_tag10	Categoria do roteiro	Varchar	32	
dt_gerado	Data em que o roteiro foi gerado	Timestamp		Default
dt_alterado	Data em que o roteiro foi alterado	Timestamp		Default
cd_usuario	Codigo do usuario que salvou o roteiro	int	11	FK

7 TECNOLOGIAS E FERRAMENTAS

Neste tópico, serão abordadas as ferramentas utilizadas para a criação deste projeto, abrangendo tanto as ferramentas da fase de desenvolvimento quanto as utilizadas no planejamento e modelagem. É importante ressaltar que existem diversas outras ferramentas disponíveis que poderiam ser aplicadas para funções semelhantes, entretanto, serão destacadas aqui apenas aquelas que foram efetivamente utilizadas ao longo do desenvolvimento deste trabalho.

7.1 DRAW.IO

O Draw.io é uma ferramenta online voltada para a criação de diagramas, fluxogramas e desenhos técnicos. Sua interface intuitiva, combinada com uma ampla variedade de recursos e modelos prontos, facilita a elaboração de representações gráficas com qualidade e agilidade.

A escolha dessa ferramenta ocorreu, principalmente, devido à familiaridade prévia da equipe com sua utilização, além de sua simplicidade e eficiência. Outro fator relevante foi sua integração com o Google Drive, o que permitiu o salvamento direto dos arquivos na pasta do projeto em nuvem, contribuindo significativamente para a organização e o compartilhamento entre os integrantes do grupo.

Por ser uma solução disponibilizada por meio do Google Workspace, o Draw.io se mostrou uma alternativa acessível, estável e altamente funcional durante a fase de planejamento do projeto.

A ferramenta foi utilizada nas etapas iniciais do desenvolvimento, especialmente na criação dos diagramas UML e do protótipo de baixa fidelidade da aplicação.

“O Draw.io é um editor gráfico online no qual é possível desenvolver desenhos, gráficos e outros sem a necessidade de usar um software caro e pesado. Com um layout simples e fácil, o serviço permite ao usuário criar seus próprios processos, inserindo uma grande variedade de formas, textos, setas indicativas, e alterar cores e espessuras de traços. Para manter seus trabalhos a salvo, é possível conectá-lo com sua conta no Google Drive. Com isso, eles ficam disponíveis para serem alterados.[...]A interface do Draw.io é realmente muito simples e seus elementos são familiares, o que facilita muito seu uso.” (JGRAPH LTD., 2024).

Figura 16 - DRAW IO.



draw.io

Fonte: logowik, 2025.

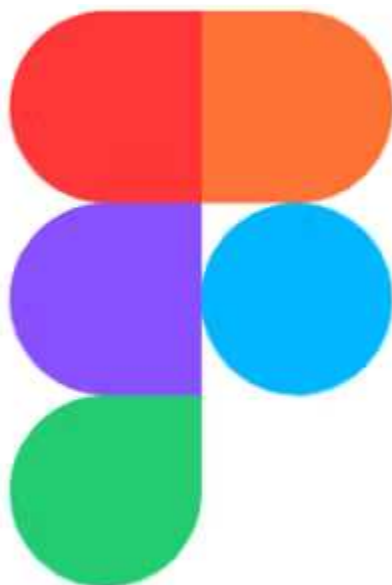
7.2 FIGMA

O Figma é uma ferramenta online utilizada para o desenvolvimento de protótipos, tanto de baixa quanto de alta fidelidade. A plataforma oferece uma ampla gama de recursos, incluindo modelos prontos, bibliotecas de componentes reutilizáveis e interações ilustrativas que permitem simular a navegação entre páginas, facilitando a visualização do fluxo da aplicação.

Uma das vantagens do Figma é sua interface colaborativa, que permite que vários membros da equipe trabalhem simultaneamente no mesmo projeto, promovendo agilidade e coesão no processo de design. Além disso, a versão paga da ferramenta oferece recursos adicionais, como a exportação de estilos e especificações diretamente para o código, o que auxilia na transição entre design e desenvolvimento.

No contexto deste projeto, o Figma foi utilizado após o levantamento dos requisitos e durante a criação da primeira versão da interface da aplicação. A ferramenta foi essencial para o desenvolvimento do protótipo de alta fidelidade, que serviu como guia visual para a implementação da estilização do site.

Figura 17 - FIGMA.



Fonte: Figma, 2025.

7.3 GITHUB

O GitHub é uma plataforma online que permite o armazenamento e o controle de versões de projetos em nuvem. A ferramenta oferece funcionalidades que auxiliam na organização do desenvolvimento, como o registro de tarefas pendentes, controle de alterações, gerenciamento de versões e compartilhamento de repositórios com equipes, permitindo que todos os colaboradores tenham acesso ao projeto e possam contribuir em tempo real.

Embora o GitHub ofereça integração com o Visual Studio Code e suporte a comandos por meio da linguagem Git, essas funcionalidades não foram utilizadas integralmente devido a dificuldades técnicas enfrentadas ao longo do desenvolvimento. Ainda assim, o uso da ferramenta foi fundamental para o armazenamento e o compartilhamento do projeto entre os membros da equipe, bem como para o controle dos requisitos definidos e das funcionalidades implementadas ou pendentes.

Além de sua utilidade prática durante o desenvolvimento, o GitHub também será utilizado futuramente como portfólio digital, reunindo este e outros projetos desenvolvidos em sala de aula, facilitando a apresentação das competências e experiências adquiridas ao longo do curso.

“O GitHub é uma plataforma de desenvolvimento colaborativo que aloja projetos na nuvem utilizando o sistema de controle de versões chamado Git. A plataforma ajuda os desenvolvedores a armazenar e administrar o código e faz o registro de mudanças. Geralmente o código é aberto, o que permite realizar projetos compartilhados e manter o acompanhamento detalhado de seu progresso. A plataforma GitHub também funciona como rede social, conectando os desenvolvedores com os usuários. Como usuário, você pode descarregar programas ou aplicativos, e da mesma maneira, pode colaborar com seu desenvolvimento oferecendo melhorias e discutindo as questões que interessam nos fóruns temáticos.” (EBAC., 2023).

Figura 18 - GitHub.



Fonte: logos-world, 2024.

7.4 VISUAL STUDIO CODE

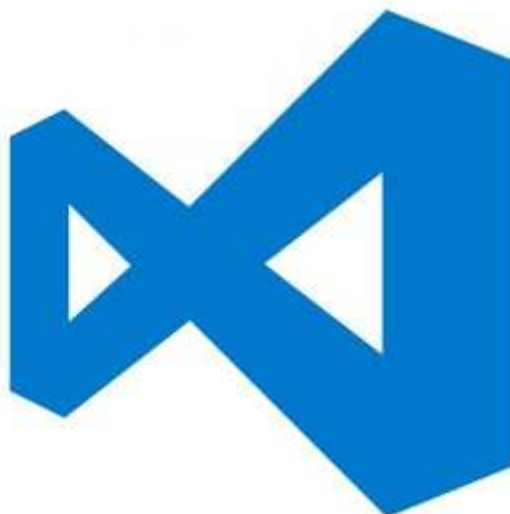
O Visual Studio Code ou VS Code é uma IDE (ambiente de desenvolvimento integrado) desenvolvido pela Microsoft. Ele permite a criação e edição de programas, aplicativos e aplicações em diversas linguagens de programação, como PHP, HTML, C#, C++, JavaScript, entre outras.

A ferramenta se destaca por sua leveza, facilidade de instalação em diferentes sistemas operacionais e ampla gama de extensões, que expandem suas funcionalidades, incluindo suporte a novas linguagens e ferramentas de análise de código.

No desenvolvimento deste projeto, o Visual Studio Code foi utilizado como editor principal para a criação do código-fonte do site, organização dos arquivos e identificação de possíveis erros por meio das extensões instaladas. Sua popularidade e suporte contínuo da comunidade contribuíram para a escolha da ferramenta como padrão durante todo o desenvolvimento.

“O Visual Studio Code é um editor de código-fonte muito popular entre os programadores. Disponibilizado pela Microsoft e Multiplataforma. Ele vem com suporte integrado para JavaScript, TypeScript e Node.JS e tem um rico ecossistema de extensões para outras linguagens (como C++, C#, Java, Python, PHP, Go).” (DIO, 2021).

Figura 19 - VISUAL STUDIO CODE.



Fonte: Logowik, 2025.

7.5 XAMPP

XAMPP é um software de hospedagem local que permite simular um servidor web diretamente no computador do desenvolvedor. Ele se destaca por sua facilidade de instalação e pela integração com o banco de dados MySQL (ou MariaDB), além de oferecer suporte a servidores como Apache e Tomcat. O XAMPP proporciona uma visualização realista do funcionamento de aplicações web e bancos de dados, sem a necessidade de um servidor externo.

No projeto, o XAMPP foi utilizado para testar e visualizar o funcionamento da aplicação, possibilitando ajustes e coleta de feedback sobre o comportamento dos elementos da página. A ferramenta também foi essencial na criação do banco de dados, definição de sua estrutura e atributos, por meio do phpMyAdmin, sua interface de administração gráfica. Por fim, o uso do XAMPP permitiu compreender o

funcionamento da aplicação em um ambiente que simula um servidor web real, facilitando a validação antes da publicação online.

“Para desenvolvedores web, o XAMPP é fundamental – e o ambiente de testes seguro e controlado proporcionado por ele é o principal motivo dele ser tão valorizado entre os profissionais da área [...] Isso não só agiliza o processo de desenvolvimento, como também ajuda a identificar e resolver erros com antecedência, garantindo que a versão final seja mais estável e eficiente.” (GARCIA, 2024).

Figura 20 - XAMPP.



Fonte: LogoWik, 2025.

7.6 PHPMYADMIN

O phpMyAdmin é uma ferramenta disponibilizada pelo XAMPP que permite visualizar, editar e gerenciar bancos de dados MySQL de forma prática e intuitiva. Ele possibilita que os usuários realizem diversas operações no banco de dados por meio de uma interface gráfica, sem a necessidade de utilizar comandos SQL diretamente.

Neste projeto, a ferramenta foi utilizada para criar o banco de dados, inserir dados personalizados durante as fases de teste e também para visualizar os dados registrados pela aplicação, possibilitando validar e testar as funcionalidades desenvolvidas ao longo do processo.

Figura 21 - phpMyAdmin.



Fonte: wikipedia, 2025.

7.7 HTML

HTML (HyperText Markup Language) é a principal linguagem utilizada no desenvolvimento de aplicações web. Sua função primordial é estruturar e apresentar os elementos de uma página, organizando o conteúdo em blocos e containers. A linguagem utiliza atributos como classes e IDs para identificar e organizar esses

blocos de maneira eficiente, seguindo princípios da programação orientada a objetos quando combinada com outras tecnologias.

Além disso, o HTML é altamente compatível com outras linguagens de desenvolvimento web, como CSS, JavaScript e PHP, podendo ser facilmente manipulado por elas. A linguagem também conta com o suporte de diversos frameworks e bibliotecas que otimizam o desenvolvimento, como o Bootstrap, que oferece componentes prontos e estilos pré-definidos para menus de navegação, formulários, botões e outros elementos de interface.

Neste projeto, o HTML foi adotado como a principal ferramenta de front-end, sendo utilizado em conjunto com o Bootstrap para a criação de uma interface responsiva, limpa e funcional.

Figura 22 - HTML.



Fonte: wikipedia, 2025.

7.8 CSS

CSS (Cascading Style Sheets) é uma linguagem complementar ao HTML que permite a aplicação de estilizações personalizadas aos elementos estruturados nas páginas web. Através do CSS, é possível definir cores, tamanhos, espaçamentos,

fontes, posicionamentos e diversos outros aspectos visuais. Além das estilizações básicas, o CSS também possibilita, quando integrado ao JavaScript, a criação de efeitos dinâmicos e animações que melhoram a experiência do usuário.

No desenvolvimento deste projeto, o uso do CSS foi pontual, uma vez que a maioria da estilização foi realizada com o auxílio do framework Bootstrap, que oferece estilos prontos e responsivos. No entanto, o CSS ainda foi utilizado para personalizações específicas, como a definição da cor de fundo do site, ajustes visuais nos elementos e detalhes adicionais que não foram contemplados pelo Bootstrap. Houve também uma tentativa de utilizar CSS em conjunto com JavaScript para criar uma barra de carregamento visual que representasse o tempo de resposta da API. No entanto, essa abordagem foi posteriormente descartada e substituída por outra solução mais prática e visualmente adequada.

Figura 23 - CSS.



Fonte: Figma, 2025.

7.9 BOOTSTRAP

Bootstrap é um framework front-end desenvolvido para facilitar o desenvolvimento de interfaces web responsivas e modernas. Ele oferece um conjunto abrangente de componentes e classes CSS e JavaScript pré-definidos, que permitem aplicar rapidamente estilos padronizados e funcionais aos elementos HTML. Dentre os principais recursos oferecidos, destacam-se componentes como navbars, modais, botões, grids, entre outros. Um de seus diferenciais é a vasta documentação oficial, clara e acessível, que facilita o aprendizado e a implementação de suas funcionalidades mesmo para desenvolvedores com pouca experiência.

No projeto desenvolvido, o Bootstrap foi utilizado em praticamente toda a estrutura de front-end. Sua biblioteca de componentes permitiu a criação de uma interface organizada e responsiva, sem a necessidade de criar estilos do zero. Entre os elementos mais utilizados, destacam-se a Navbar, que foi empregada para permitir a navegação entre diferentes seções da aplicação, e os Modals, usados para exibir mensagens de aviso, confirmações e feedbacks ao usuário. Apesar da forte presença do Bootstrap, alguns estilos personalizados também foram aplicados via CSS para atender demandas específicas do projeto, como ajustes de cores e detalhes visuais não previstos pelos estilos padrão do framework.

Figura 24 - Bootstrap.



Fonte: Figma, 2025.

7.10 PHP

PHP (Hypertext Preprocessor) é uma linguagem de programação amplamente utilizada no desenvolvimento web, principalmente na camada back-end. Sua principal função é permitir a criação de páginas dinâmicas e a manipulação de dados entre a interface do usuário (front-end) e o banco de dados. Diferentemente do HTML, que apenas estrutura o conteúdo da página, o PHP é executado no servidor, o que o torna essencial para a lógica de funcionamento da aplicação, como autenticação de usuários, envio de formulários, e operações com o banco de dados.

No desenvolvimento deste projeto, o PHP foi utilizado como linguagem principal para o back-end, sendo responsável pela comunicação entre os formulários da interface web e o banco de dados MySQL. Por meio da extensão MySQLi (MySQL Improved), foi possível realizar ações como inserir, atualizar, consultar e excluir registros no banco de dados, garantindo assim a persistência das informações fornecidas pelos usuários. Além disso, o PHP foi utilizado para processar as requisições de login, cadastro, salvamento de dados, e gerenciamento de sessões, assegurando que cada usuário tivesse acesso somente às suas informações.

A escolha pelo PHP se deu tanto por sua ampla compatibilidade com o servidor local XAMPP quanto pela familiaridade da equipe com a linguagem, o que contribuiu para um desenvolvimento mais ágil e eficiente.

Figura 25 - PHP.



Fonte: logowik, 2025.

7.11 MYSQL

MySQL é um sistema de gerenciamento de banco de dados relacional amplamente utilizado em aplicações web. Ele utiliza a linguagem SQL (Structured Query Language) para armazenar, consultar e estruturar dados, sendo ideal para lidar com grandes volumes de informação de forma organizada e segura. Por ser compatível com diversas linguagens de programação, como PHP, é uma das escolhas mais comuns no desenvolvimento de sistemas web dinâmicos.

No contexto deste projeto, o MySQL foi utilizado como banco de dados principal, sendo responsável por armazenar dados essenciais como informações de usuários, prompts enviados, respostas geradas pela API e o histórico de interações. A criação e manutenção das tabelas, bem como a definição de seus relacionamentos, foram realizadas por meio do phpMyAdmin, uma ferramenta gráfica integrada ao XAMPP, que facilita a administração do banco de dados.

Essa combinação de ferramentas MySQL, PHP e XAMPP — proporcionou um ambiente de desenvolvimento estável, eficiente e de fácil visualização, permitindo que todas as interações do sistema com o banco de dados fossem testadas localmente de forma prática e segura.

Figura 26 - DRAW IO.



Fonte: Figma, 2025.

7.12 JAVASCRIPT

JavaScript é uma linguagem de programação que complementa o HTML, sendo amplamente utilizada no desenvolvimento web. É considerada a principal linguagem de programação voltada para o lado do cliente, permitindo a manipulação dinâmica de elementos HTML, além de possuir funcionalidades semelhantes às linguagens de back-end, como o próprio PHP. Além disso, é a principal tecnologia utilizada para a comunicação com APIs e pode ser empregada em ambientes e frameworks como Node.js e React, estendendo sua aplicação para além do navegador.

No projeto, o JavaScript foi utilizado principalmente como ponte entre a API do ChatGPT e a aplicação web. A linguagem também foi aplicada para manipular elementos da interface, como os modais de aviso de erro ou sucesso. Nesses casos, o JavaScript foi responsável por detectar o fechamento dos modais e reiniciar a página, corrigindo automaticamente os erros gerados pela aplicação. Além disso, foi implementada uma funcionalidade de texto dinâmico, que substituiu a imagem anteriormente utilizada para representar o tempo de carregamento da resposta da API, proporcionando uma experiência mais leve e fluida ao usuário.

Figura 27 - JAVASCRIPT.



Fonte: Logos-World, 2025.

7.13 CHAT GPT

O ChatGPT é um modelo de linguagem baseado em inteligência artificial desenvolvido pela OpenAI, que utiliza redes neurais avançadas. Ele é capaz de compreender linguagem natural e gerar respostas coerentes, criativas e contextualizadas a partir de comandos textuais e dados armazenados e coletados previamente. A tecnologia é amplamente utilizada em diversas aplicações, como assistentes virtuais, suporte ao cliente, geração de conteúdo e automatização de tarefas linguísticas.

Para integrar esse modelo em aplicações externas, a OpenAI disponibiliza uma API (Application Programming Interface), que permite o envio de comandos (prompts) para o modelo e o recebimento de respostas geradas em tempo real. A API do ChatGPT pode ser acessada via requisições HTTP utilizando linguagens como JavaScript ou linguagens de back-end como Python, PHP e Node.js. A comunicação é geralmente feita no formato JSON, e é necessário o uso de uma chave de autenticação (API Key) fornecida pela OpenAI.

No projeto desenvolvido, a API do ChatGPT teve papel essencial, pois foi a principal ferramenta usada para o funcionamento do programa. A aplicação envia o conteúdo inserido pelo usuário para a API, que retorna uma resposta processada, simulando uma comunicação inteligente e contextualizada. Essa resposta, por sua vez, é apresentada ao usuário em tempo real por meio da interface do site.

A API permitiu a implementação de recursos avançados, como a geração de roteiros com base em categorias e palavras-chave definidas, ampliando as possibilidades de interação e personalização dentro do sistema. Seu uso contribuiu para o funcionamento do projeto, ao mesmo tempo, em que demonstrou a aplicabilidade de modelos de linguagem natural em contextos reais de desenvolvimento web.

Figura 28 - Chat GPT.



Fonte: Pinterest, 2025.

8 DESENVOLVIMENTO

Neste tópico, será abordada a fase de desenvolvimento do projeto, detalhando os elementos que compõem o sistema e explicando como cada parte foi implementada. O desenvolvimento foi dividido em três fases principais, cada uma com foco em funcionalidades essenciais para o funcionamento do site. Apesar de cada fase ter um objetivo específico, houve também ajustes e melhorias no sistema todo ao longo do processo, sempre conforme o planejamento inicial.

A primeira fase teve como foco a integração da API do ChatGPT ao sistema. Esta funcionalidade representa o núcleo do projeto, já que todo o propósito do site gira em torno da geração de roteiros com o apoio da inteligência artificial. Nesta etapa, foi utilizada principalmente a linguagem JavaScript, responsável por se comunicar com a API e processar as respostas retornadas. Essa fase foi crucial para validar a viabilidade do projeto, onde, caso a API não funcionasse conforme o esperado, seria necessário reformular totalmente o objetivo da aplicação.

A segunda fase concentrou-se na implementação do banco de dados. Utilizando PHP e MySQL, foi desenvolvido um sistema de cadastro de usuários e armazenamento de histórico, permitindo que os roteiros criados pudessem ser salvos, visualizados e reutilizados. Ainda que essas funcionalidades já tivessem sido esboçadas na primeira fase, elas foram totalmente refeitas nesta etapa, com melhorias significativas em estrutura e desempenho. Essa fase foi essencial para permitir o início de uma fase de testes com usuários reais, que puderam avaliar as funcionalidades disponíveis e contribuir com feedbacks.

A terceira fase retomou o foco na API, desta vez para implementar a funcionalidade de elaboração de roteiros com base em textos já existentes. Diferente da geração de ideias novas, esta função exigiu um aproveitamento mais avançado da API, sendo necessário compreender melhor suas limitações e possibilidades. Além disso, esta etapa foi marcada por uma série de correções e ajustes com base no retorno dos usuários, como conflitos entre elementos da interface e sugestões sobre a estrutura da elaboração de roteiros.

A seguir, serão apresentados os componentes que integram o sistema de forma detalhada, permitindo uma compreensão clara e segmentada do desenvolvimento técnico do projeto.

8.1 BANCO DE DADOS

Como mencionado anteriormente, o banco de dados utilizado neste projeto foi o MySQL. A escolha dessa tecnologia se deu por diversos fatores: em primeiro lugar, trata-se de um dos bancos de dados mais compatíveis com a linguagem PHP, utilizada no desenvolvimento do back-end do sistema. Além disso, o MySQL foi amplamente explorado durante o curso, o que proporcionou maior familiaridade com seus conceitos e aplicações práticas. A linguagem SQL, utilizada para manipulação e consulta de dados no MySQL, também foi estudada em profundidade, o que reforçou a decisão de optar por esse sistema gerenciador de banco de dados.

Outro fator determinante foi o fato de o MySQL ser amplamente utilizado no mercado de trabalho, o que torna sua compreensão e aplicação um diferencial importante para qualquer desenvolvedor. Dessa forma, além de atender às necessidades do projeto, a utilização do MySQL também contribuiu para o preparo dos integrantes do grupo em relação ao cenário profissional.

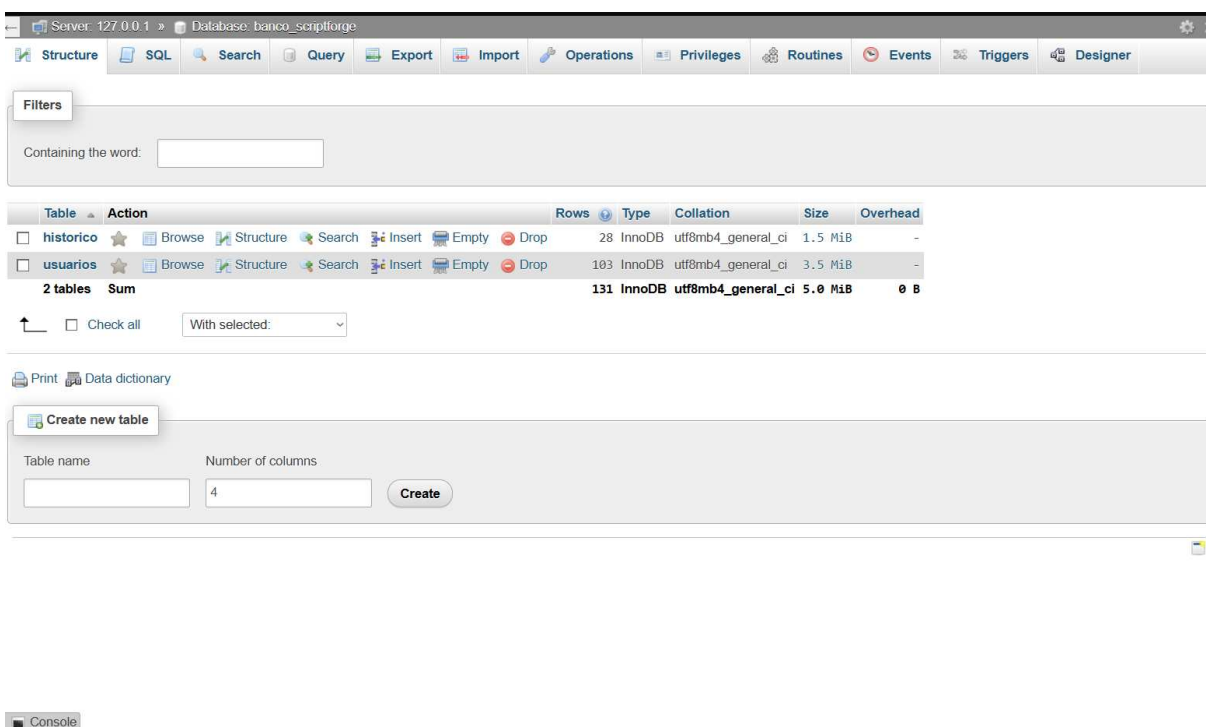
O banco de dados foi desenvolvido durante a segunda fase de desenvolvimento do projeto. Como nessa etapa já havia clareza sobre os dados que precisariam ser armazenados e sobre os relacionamentos entre eles, o processo de modelagem foi realizado de forma mais estruturada e eficiente. A primeira etapa consistiu na criação das tabelas.

Para a implementação prática do banco de dados, foi utilizado o XAMPP, que oferece um ambiente de desenvolvimento local com suporte ao MySQL e uma interface gráfica por meio do phpMyAdmin. Essa ferramenta facilitou a criação e

visualização das tabelas, seus campos e os relacionamentos entre elas, contribuindo para um processo mais didático e intuitivo.

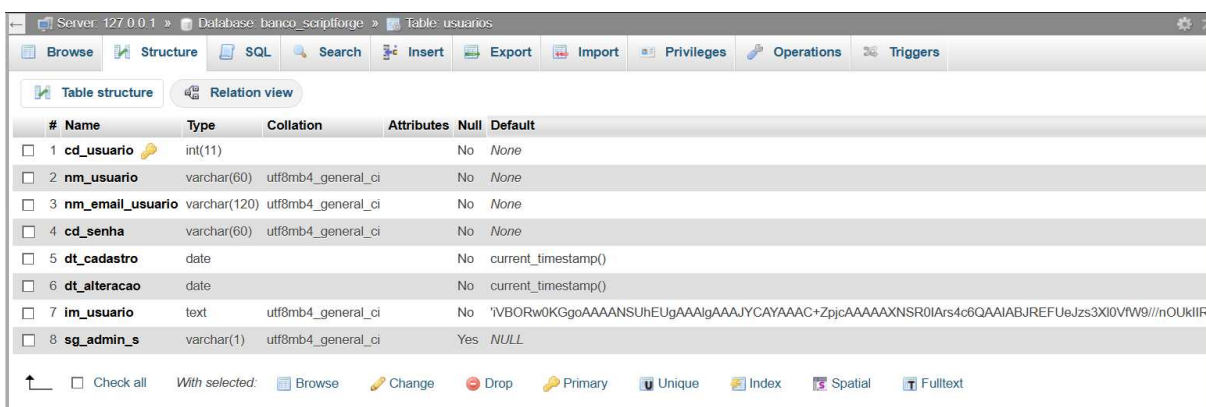
A seguir, será apresentada a estrutura do banco de dados a partir do phpMyAdmin, com suas respectivas tabelas, atributos e relacionamentos, a fim de ilustrar de forma clara a organização e o fluxo das informações dentro do sistema.

Figura 29 - Banco ScriptForge.



Fonte: autoral, 2025.

Figura 30 - Tabela Usuário.



Fonte: autoral, 2025.

Figura 31 - tabela histórico.

#	Name	Type	Collation	Attributes	Null	Default
<input type="checkbox"/>	1	cd_historico	int(11)		No	None
<input type="checkbox"/>	2	ds_prompt	text	utf8mb4_general_ci	No	None
<input type="checkbox"/>	3	nm_titulo	text	utf8mb4_general_ci	No	None
<input type="checkbox"/>	4	ds_resposta	text	utf8mb4_general_ci	No	None
<input type="checkbox"/>	5	im_resposta	text	utf8mb4_general_ci	No	'UkiGRu5CAABXRUIQVIA4IOJCAABQrwKdASqABzgEPmEwIkgklyIhrS4CIAMCWlu/Enu7Z8eXfklI8nrJIHGfCP5fXv7v+
<input type="checkbox"/>	6	ds_resposta_elaborada	text	utf8mb4_general_ci	No	None
<input type="checkbox"/>	7	nm_tag01	varchar(32)	utf8mb4_general_ci	No	None
<input type="checkbox"/>	8	nm_tag02	varchar(32)	utf8mb4_general_ci	No	None
<input type="checkbox"/>	9	nm_tag03	varchar(32)	utf8mb4_general_ci	No	None
<input type="checkbox"/>	10	nm_tag04	varchar(32)	utf8mb4_general_ci	No	None
<input type="checkbox"/>	11	nm_tag05	varchar(32)	utf8mb4_general_ci	No	None
<input type="checkbox"/>	12	nm_tag06	varchar(32)	utf8mb4_general_ci	No	None
<input type="checkbox"/>	13	nm_tag07	varchar(32)	utf8mb4_general_ci	No	None
<input type="checkbox"/>	14	nm_tag08	varchar(32)	utf8mb4_general_ci	No	None
<input type="checkbox"/>	15	nm_tag09	varchar(32)	utf8mb4_general_ci	No	None
<input type="checkbox"/>	16	nm_tag10	varchar(32)	utf8mb4_general_ci	No	None
<input type="checkbox"/>	17	dt_gerado	timestamp		No	current_timestamp()
<input type="checkbox"/>	18	dt_alterado	timestamp		No	current_timestamp()
<input type="checkbox"/>	19	cd_usuario	int(11)		No	None

Fonte: autoral, 2025.

A ferramenta phpMyAdmin, utilizada em conjunto com o XAMPP, permite a criação de tabelas de forma visual e intuitiva, facilitando o processo de modelagem do banco de dados. Essa abordagem gráfica possibilita a definição dos campos, tipos de dados, chaves primárias e estrangeiras, além dos relacionamentos entre as tabelas, sem a necessidade de codificação manual em um primeiro momento.

Contudo, após a modelagem visual, o phpMyAdmin permite a exportação do banco de dados em formato .sql, o que gera automaticamente os comandos em linguagem SQL responsáveis pela criação e estruturação das tabelas. Esse arquivo contém todas as instruções necessárias para recriar o banco de dados em qualquer outro ambiente compatível com MySQL, oferecendo flexibilidade e portabilidade ao projeto.

A seguir, será apresentada a estrutura do banco de dados em linguagem SQL, destacando as tabelas criadas, os campos utilizados e os relacionamentos estabelecidos entre elas.

```

CREATE TABLE `historico` (
  `cd_historico` int(11) NOT NULL,
  `ds_prompt` text NOT NULL,
  `nm_titulo` text NOT NULL,
  `ds_resposta` text NOT NULL,
  `im_resposta` text NOT NULL DEFAULT 'base64exemplo',
  `ds_resposta_elaborada` text NOT NULL,
  `nm_tag01` varchar(32) NOT NULL,
  `nm_tag02` varchar(32) NOT NULL,
  `nm_tag03` varchar(32) NOT NULL,
  `nm_tag04` varchar(32) NOT NULL,
  `nm_tag05` varchar(32) NOT NULL,
  `nm_tag06` varchar(32) NOT NULL,
  `nm_tag07` varchar(32) NOT NULL,
  `nm_tag08` varchar(32) NOT NULL,
  `nm_tag09` varchar(32) NOT NULL,
  `nm_tag10` varchar(32) NOT NULL,
  `dt_gerado` timestamp NOT NULL DEFAULT current_timestamp(),
  `dt_alterado` timestamp NOT NULL DEFAULT current_timestamp(),
  `cd_usuario` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

CREATE TABLE `usuarios` (
  `cd_usuario` int(11) NOT NULL,
  `nm_usuario` varchar(60) NOT NULL,
  `nm_email_usuario` varchar(120) NOT NULL,
  `cd_senha` varchar(60) NOT NULL,
  `dt_cadastro` date NOT NULL DEFAULT current_timestamp(),
  `dt_alteracao` date NOT NULL DEFAULT current_timestamp(),
  `im_usuario` text NOT NULL DEFAULT 'base64exemplo',
  `sg_admin_s` varchar(1) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

ALTER TABLE `historico`
  ADD PRIMARY KEY (`cd_historico`),
  ADD KEY `cd_usuario` (`cd_usuario`);

ALTER TABLE `usuarios`
  ADD PRIMARY KEY (`cd_usuario`);

ALTER TABLE `historico`
  MODIFY `cd_historico` int(11) NOT NULL AUTO_INCREMENT,
  AUTO_INCREMENT=124;

ALTER TABLE `usuarios`
  MODIFY `cd_usuario` int(11) NOT NULL AUTO_INCREMENT,
  AUTO_INCREMENT=130;

ALTER TABLE `historico`

```

```
ADD CONSTRAINT `historico_ibfk_1` FOREIGN KEY (`cd_usuario`)
REFERENCES `usuarios` (`cd_usuario`);
COMMIT;
```

O código gerado em SQL foi adaptado para atender às necessidades específicas do projeto. Um exemplo dessa adaptação está nas colunas `im_usuario` e `im_resposta`, que armazenam imagens convertidas para o formato Base64. Devido ao tamanho dessas informações, optou-se por não as exibir diretamente nas representações gráficas das tabelas, mas sim inseri-las como valores padrão no código SQL. Essas imagens servem como ilustração inicial, a do usuário, para ser personalizada posteriormente, e a da resposta, planejada para representar visualmente os roteiros gerados.

A imagem associada à resposta do ChatGPT foi incluída como parte de uma funcionalidade inicialmente planejada que era a geração de imagens ilustrativas dos roteiros. No entanto, devido a limitações técnicas e de tempo, essa funcionalidade não foi concluída nesta fase do projeto, planejada para futuras etapas de desenvolvimento.

Após a criação das tabelas, o próximo passo foi o estabelecimento dos relacionamentos entre elas, garantindo a integridade dos dados. Com a modelagem do banco finalizada, prosseguiu-se para a etapa de desenvolvimento em PHP, onde o banco de dados foi integrado ao site. Essa integração, embora tecnicamente simples, foi essencial para o funcionamento das principais funcionalidades do sistema.

A seguir, será apresentado o funcionamento do Back-End, detalhando como a conexão com o banco de dados foi realizada e de que forma o site interage com as informações armazenadas.

8.2 BACK-END

O back-end do site foi responsável por realizar a integração entre a aplicação web e o banco de dados, utilizando a linguagem PHP em conjunto com funções MySQLi. Essa integração permitiu criar uma ponte eficiente entre o conteúdo apresentado ao usuário e os dados armazenados no sistema. Cada arquivo PHP foi estruturado com base em tipos distintos de funcionalidade: funções básicas do usuário, funções do administrador, funções do histórico e funções comuns da aplicação. Essas funcionalidades serão detalhadas nos próximos tópicos.

Além disso, o back-end também é responsável por armazenar a chave da API utilizada no front-end. Essa prática visa a proteção da segurança do sistema, uma vez que a chave de autenticação da API não deve ser exposta no código acessível ao usuário final. Caso contrário, pessoas mal-intencionadas poderiam utilizá-la de maneira indevida, comprometendo o bom funcionamento do projeto e podendo gerar custos ou violações de uso.

O back-end também está presente em diversas áreas do site, mesmo naquelas que não se relacionam diretamente com o banco de dados. Nesses trechos, ele atua no gerenciamento da navegação entre páginas, na exibição de mensagens ou textos dinâmicos, e no controle de ações específicas com base no tipo de usuário, seja ele comum ou administrador. Outra função importante do back-end é a recuperação de roteiros salvos no banco de dados, permitindo que esses roteiros sejam reutilizados para elaborar novos textos, oferecendo continuidade e aproveitamento de conteúdo anterior.

8.2.1 FUNÇÕES COMUNS

O arquivo denominado `comun_function.php` é responsável por reunir as funções comuns utilizadas em diferentes partes do site. Ao contrário das funções

específicas para usuários ou administradores, as funções presentes neste arquivo não se restringem a um tipo específico de usuário, podendo ser executadas em praticamente todas as páginas da aplicação.

Entre as principais funcionalidades implementadas estão os mecanismos de login, logout, cadastro de usuários e consultas básicas ao banco de dados. Além disso, este arquivo também é responsável por estabelecer a conexão entre o site e o banco de dados, o que o torna essencial para qualquer operação que envolva manipulação ou leitura de dados persistentes.

Dessa forma, é fundamental que o `comun_function.php` seja incluído sempre que for necessário acessar ou modificar informações armazenadas no banco de dados, garantindo o funcionamento correto e padronizado da aplicação, é importante ressaltar que outros arquivos também possuem funções similares as que serão descritas aqui.

A seguir, serão detalhadas as principais funções implementadas nesse arquivo, com o objetivo de demonstrar seu papel dentro da arquitetura do sistema.

- **Login:** A função de login é responsável por autenticar o usuário no sistema. Ela coleta o e-mail e a senha inseridos no formulário e verifica se ambos os campos foram preenchidos corretamente. Caso falte alguma informação ou ocorra um erro, uma mensagem explicativa é exibida ao usuário. Se os dados estiverem corretos, a função realiza uma busca no banco de dados utilizando a senha criptografada. Se o usuário for encontrado, suas informações são recuperadas e armazenadas na sessão, conectando-o ao site com os dados correspondentes, como nome, foto e tipo de acesso.
- **Select:** A função select realiza a consulta de um único usuário no banco de dados, com base no seu identificador (`CD_usuario`). Ela retorna todas as informações armazenadas daquele usuário. É utilizada principalmente em situações onde é necessário acessar os

dados de outro usuário, como em áreas administrativas ou exibições públicas de perfil.

- **Insert:** A função de insert de usuário é responsável pelo cadastro de novos usuários. Ela coleta os dados enviados pelo formulário e realiza diversas validações. Se todas as condições forem atendidas, o usuário é cadastrado com sucesso no banco de dados, e uma mensagem de confirmação é exibida. Caso haja erro, uma mensagem com a descrição específica do problema é apresentada ao usuário.
- **Logout:** A função logout é responsável por encerrar a sessão do usuário. Ao ser chamada, ela limpa todas as variáveis da sessão ativa e redireciona o usuário para a página inicial do site. Essa ação é essencial para garantir a segurança e o controle de acesso, especialmente em ambientes compartilhados.

8.2.2 FUNÇÕES DO USUÁRIO

As funções descritas nesta seção são responsáveis por manipular dados diretamente relacionados ao próprio usuário. Embora algumas se assemelhem às funções comuns, elas são chamadas apenas em páginas específicas voltadas para a área do usuário autenticado.

Essas funções permitem que o próprio usuário realize ações como consultar informações detalhadas do seu perfil, atualizar seus dados pessoais ou alterar sua imagem de perfil.

Por serem ações que exigem autenticação e afetam diretamente os dados pessoais do usuário, essas funções possuem verificações específicas e são projetadas para garantir a integridade e a segurança das informações armazenadas.

A seguir, serão apresentadas em detalhes as principais funções implementadas para o usuário:

- **Select:** A função select permite consultar os dados pessoais de um usuário específico. Ao ser acionada, ela realiza uma busca no banco de dados com base no identificador do usuário e retorna as informações armazenadas, como nome, e-mail e imagem de perfil. Essa função é utilizada em páginas onde o usuário visualiza seus próprios dados ou quando é necessário apresentar informações de outro usuário (por exemplo, em interações entre usuários).
- **Update:** A função update é responsável por atualizar informações do perfil do usuário, essa função é personalizada com base nos diferentes tipos de dados que podem ser alterados, como senha, E-mail e imagem de perfil. O funcionamento segue a mesma lógica, os dados fornecidos pelo usuário são validados, e, caso estejam corretos, a atualização é realizada no banco de dados. Se houver inconsistências ou dados inválidos, a função retorna mensagens de erro específicas, garantindo clareza no feedback ao usuário.
- **Delete:** A função delete permite ao usuário excluir permanentemente sua conta do sistema. Quando acionada, ela remove o registro do usuário no banco de dados, apagando todas as informações associadas. Essa ação é irreversível, portanto, é geralmente acompanhada de uma confirmação adicional por parte do usuário para evitar exclusões acidentais.

8.2.3 FUNÇÕES DO ADMINISTRADOR

As funções administrativas são acessíveis exclusivamente por usuários com permissões de administrador e são acionadas apenas quando há necessidade de visualizar, cadastrar, alterar ou excluir informações de outros usuários no sistema.

Essas funcionalidades não podem ser acessadas por usuários comuns, garantindo a separação entre os níveis de acesso e mantendo a integridade da aplicação.

Devido ao seu potencial de impacto direto sobre os dados de terceiros, é fundamental que essas funções sejam protegidas por verificações de segurança rigorosas. A verificação do tipo de usuário deve ser realizada antes da execução de qualquer função desse grupo. Isso previne que usuários não autorizados acessem funcionalidades críticas por meio de manipulações no sistema.

Além disso, é importante ressaltar que administradores não possuem permissão para modificar ou excluir outros administradores, tampouco podem alterar seus próprios privilégios ou apagar suas próprias contas através dessas rotinas. Esse controle é intencional, visando evitar abusos e garantir que ações administrativas sejam feitas com responsabilidade e rastreabilidade.

Embora muitas dessas funções compartilhem a mesma estrutura lógica de funções comuns ou de usuário como insert, select, update e delete, elas operam sobre outros usuários, o que amplia seu impacto e importância no contexto da aplicação.

A seguir, cada uma dessas funções será apresentada em maior detalhe técnico.

- **Select:** A função select tem como objetivo consultar dados de outros usuários. Ela é implementada de duas formas, um select individual, que retorna as informações de um único usuário com base no seu identificador único e um select em massa, que retorna todos os registros da tabela de usuários, sendo útil para exibição em listas do painel de usuários.
- **Insert:** A função insert permite ao administrador cadastrar novos usuários diretamente pelo painel administrativo. Essa função funciona de forma semelhante ao cadastro comum feito pelo usuário, com a diferença de que não redireciona a página após a conclusão, o que

facilita a inserção de múltiplos usuários de forma contínua. Ela valida os dados fornecidos e exibe uma mensagem de sucesso ou erro, dependendo do resultado da operação.

- **Delete:** A função delete permite excluir contas de usuários do sistema. Por questões de segurança e integridade, essa função não permite a exclusão de outros administradores nem do próprio administrador logado. Caso um administrador deseje excluir sua própria conta, ele deverá realizar essa ação através das funções destinadas ao usuário.
- **Update:** A função update permite a modificação de dados de qualquer usuário comum, com base nas informações fornecidas pelo administrador. Assim como na exclusão, não é possível alterar dados de outros administradores nem do próprio administrador. Para modificar as próprias informações, o administrador deve utilizar as funções da área de usuário.

8.2.4 FUNÇÕES DO HISTÓRICO

Como mencionado anteriormente, estas funções são responsáveis por manipular os dados de histórico e os roteiros salvos pelos usuários, sendo distintas das funções relacionadas diretamente ao gerenciamento de usuários. Seu foco está em registrar, exibir e remover os roteiros gerados ou salvos pelo usuário.

Diferentemente das funções de cadastro ou administração, estas não alteram dados pessoais dos usuários, mas sim os conteúdos gerados dentro do sistema. Isso inclui, o roteiro criado a partir da interação com a API ou os registros armazenados no histórico de uso.

Essas funções são chamadas sempre que o usuário interage com o sistema de roteiros, seja ao gerar um novo roteiro, ao visualizar um roteiro salvo

anteriormente ou ao excluir algum item do histórico. Elas também podem ser acionadas pela própria API do sistema, quando necessário registrar automaticamente uma nova entrada.

Além disso, estas funções são independentes do tipo de usuário. Tanto usuários comuns quanto administradores utilizam o mesmo conjunto de funções para manipular o histórico pessoal. O sistema garante que cada usuário só tenha acesso ao seu próprio histórico, mantendo a privacidade e organização dos dados.

A seguir, será apresentada uma descrição técnica detalhada de cada uma dessas funções.

- **Select:** As funções de seleção são responsáveis por consultar os roteiros salvos no banco de dados. Elas foram divididas em três categorias específicas, um select individual, que seleciona um único roteiro com base no seu identificador (cd_historico) e é utilizada quando o usuário deseja visualizar um roteiro específico em detalhes, um select geral, que retorna todos os roteiros salvos pelo usuário, essa função é usada para exibir uma lista completa de roteiros em páginas de gerenciamento ou consulta e por fim um select dos últimos roteiros salvos, essa função retorna os cinco roteiros mais recentes salvos pelo usuário, essa função é utilizada na navbar do site, possibilitando um acesso rápido aos conteúdos mais recentes gerados.
- **Insert:** A função insert é responsável por inserir novos roteiros no banco de dados. Ela é chamada sempre que o usuário opta por salvar um roteiro gerado. Após a coleta dos dados, a função realiza uma validação básica e, caso os dados estejam corretos, envia-os para o banco. Em caso de inconsistência ou erro, o sistema retorna uma mensagem explicativa ao usuário.
- **Update:** A função update permite que um roteiro salvo anteriormente seja substituído por uma versão atualizada. Essa funcionalidade é acionada quando o usuário edita um roteiro já existente e opta por

salvar as alterações sobre o mesmo registro. A função coleta os novos dados e os grava no banco de dados, substituindo os anteriores.

- **Delete:** A função delete é responsável por remover permanentemente um roteiro salvo no banco de dados. Ela é chamada quando o usuário decide apagar um roteiro do seu histórico, garantindo que o mesmo não esteja mais disponível no sistema.

8.3 FRONT-END

O Front-End representa uma das partes principais do projeto, sendo responsável tanto pela apresentação visual da aplicação quanto pela interação direta com o usuário. Embora boa parte dessa camada esteja voltada à exibição dos elementos gráficos e à usabilidade do sistema, ela também contém funcionalidades fundamentais para o funcionamento do projeto como um todo.

A principal funcionalidade implementada no Front-End é a chamada à API, realizada por meio do JavaScript utilizando a função fetch. Essa comunicação permite que os dados gerados a partir da API sejam processados e apresentados de forma dinâmica ao usuário.

O desenvolvimento do Front-End ocorreu ao longo de todo o projeto, mas as funcionalidades centrais foram implementadas, principalmente, durante a primeira e a terceira fases do processo de desenvolvimento, focadas diretamente na integração com a API. Essas funcionalidades serão detalhadas nos tópicos seguintes.

Além disso, o Front-End também foi responsável pela estilização visual do site, utilizando arquivos CSS personalizados em conjunto com a biblioteca Bootstrap, que facilitou a construção de uma interface moderna, responsiva e

amigável ao usuário. A estrutura visual e os elementos de design também serão explicados com mais profundidade nos próximos tópicos.

Por fim também será apresentado em detalhes as páginas e como elas estão estruturadas no projeto, além do fluxo entre elas e os tipos de arquivos

8.3.1 PÁGINAS

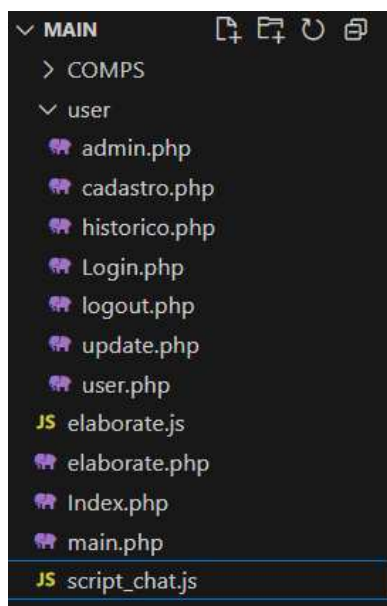
As páginas do sistema têm, em sua maioria, a função de apresentar a interface do sistema ao usuário. Essa apresentação é realizada por meio da linguagem HTML, que pode ser utilizada tanto em arquivos com extensão .html quanto em arquivos .php. No desenvolvimento deste projeto, optou-se pelo uso de arquivos .php, pois além de permitir a escrita em HTML, essa extensão possibilita a inclusão de trechos de código que interagem com o back-end, oferecendo maior flexibilidade no desenvolvimento.

Essa abordagem foi especialmente útil em componentes como na navbar, que exibe diferentes itens dependendo do status de login do usuário. Essa verificação é feita por meio de scripts escritos em PHP, permitindo personalizar a experiência de navegação com base na sessão ativa.

Neste tópico, serão abordadas detalhadamente as páginas implementadas no sistema, suas respectivas funcionalidades e como estão organizadas dentro da estrutura de arquivos do projeto.

A seguir, apresenta-se a disposição das páginas, seguidos de legendas explicando em detalhes:

Figura 32 - Páginas 1.



Fonte: autoral, 2025.

A imagem apresentada ilustra a estrutura das principais páginas do sistema. A página “Index.php” funciona como a página inicial do site. Nela, o usuário encontra uma breve explicação sobre o funcionamento da plataforma, além de artigos relacionados ao uso de inteligência artificial na geração de textos, a página também apresenta um botão que redireciona o usuário para a página “Main.php”, considerada a principal interface do sistema.

Na “Main.php”, o usuário pode interagir diretamente com o chat para criar roteiros. Essa interação é viabilizada por meio do arquivo “Script_chat.js”, responsável por conectar a interface com a API do ChatGPT e gerenciar as funcionalidades de envio e exibição de mensagens, essa interação será apresentada em mais detalhes quando for explicado o consumo da API.

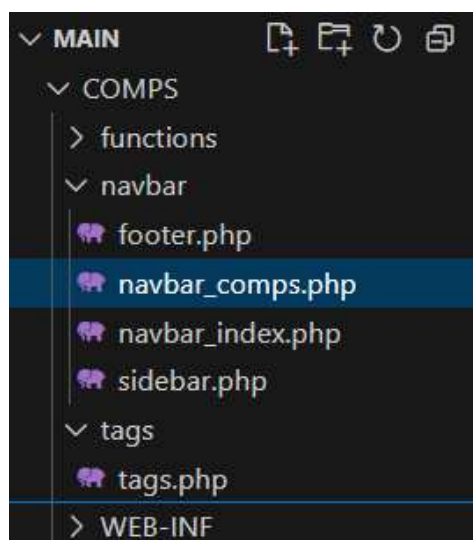
A partir da página principal, o usuário também pode acessar a página “Elaborate.php”, cuja função é refinar e elaborar o roteiro desenvolvido na etapa anterior. Assim como a “Main.php”, essa página conta com um JavaScript específico,

o “Elaborate.js”, que possui funcionalidades similares as do “Script_chat.js”, porém voltadas à etapa de finalização do roteiro, também interagindo com a API.

Além dessas páginas centrais, o sistema conta com a pasta “users”, que armazena páginas relacionados ao gerenciamento de usuários e administradores. Nela, encontram-se as páginas de “login.php” e “cadastro.php”, responsáveis por autenticar e registrar usuários no sistema.

As páginas “users.php” e “historico.php” são acessíveis apenas após o login e exibem, respectivamente, informações pessoais do usuário e os roteiros salvos anteriormente. Por fim, há as páginas “admin.php” e “update.php”, de acesso restrito ao administrador. A primeira permite a visualização dos usuários cadastrados, enquanto a segunda oferece ferramentas para atualização dos dados desses usuários.

Figura 33 - Páginas 2.



Fonte: autoral, 2025.

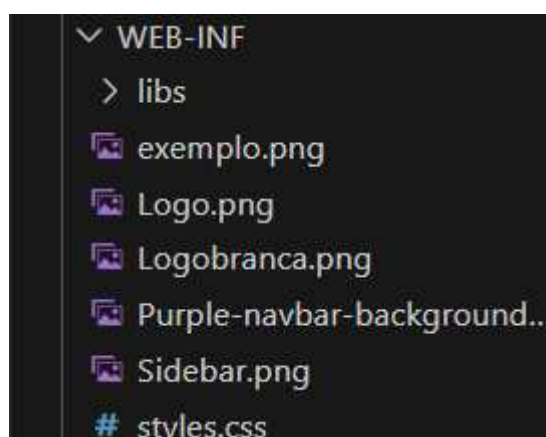
A imagem apresentada também evidencia alguns dos componentes reutilizáveis utilizados nas páginas do sistema. Para garantir uma melhor organização e manutenção do código, certos elementos foram separados em arquivos próprios, mesmo que sejam incorporados em outras páginas. Um exemplo disso é o arquivo “tags.php”, que, embora exibido na página “Main.php”, foi isolado

em um arquivo separado. Essa página é responsável por apresentar as categorias disponíveis para a criação de roteiros. Além disso, ela conta com um script JavaScript que organiza tanto a disposição visual quanto a forma de apresentação das categorias ao usuário.

Outros componentes reutilizáveis estão armazenados na pasta “NavBar”, que contém arquivos responsáveis pelos menus de navegação superiores. Os arquivos “navbar_comps.php” e “navbar_index.php” representam essas estruturas, sendo utilizados conforme o tipo de página. Essa separação se justifica pela necessidade de exibir elementos distintos, enquanto um deles contém opções específicas para usuários autenticados, o outro exibe elementos voltados para visitantes e páginas iniciais do sistema.

Outro elemento reutilizável presente em todas as páginas é o “footer.php”, que mantém um padrão de exibição ao final de cada página do site. Além disso, há também a “SideBar.php”, responsável pela exibição de uma barra lateral de navegação. Esta é apresentada exclusivamente nas páginas acessadas por usuários autenticados, contribuindo para uma navegação mais eficiente e organizada dentro do ambiente do usuário.

Figura 34 - Páginas 3.



Fonte: autoral, 2025.

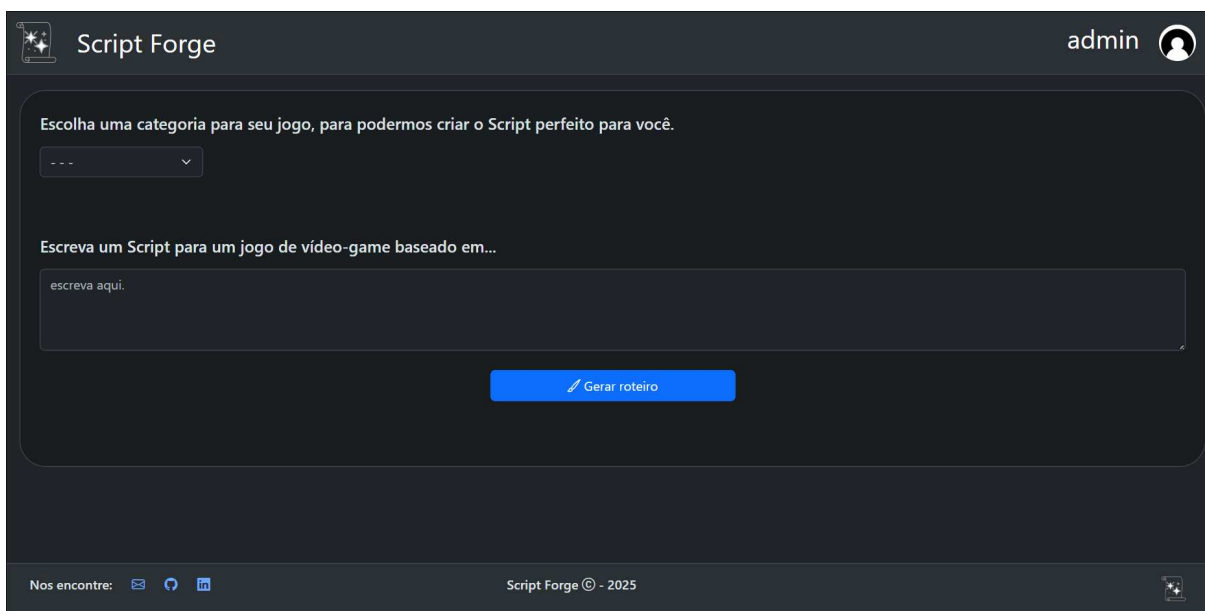
Por fim, temos a pasta “WEB-INF” apresentada na imagem acima, que, embora não contenha páginas propriamente ditas, armazena arquivos essenciais para a composição e o funcionamento das páginas do sistema. Dentro dessa pasta, encontram-se os recursos visuais, como imagens utilizadas ao longo do site, além de arquivos responsáveis pela estilização da interface.

Entre esses arquivos, destaca-se o “Styles.css”, que contém estilos personalizados desenvolvidos especificamente para o projeto. Esse arquivo é utilizado em conjunto com o framework “Bootstrap”, permitindo uma maior flexibilidade na construção do layout e no design responsivo das páginas.

Além disso, temos a subpasta “Libs” que armazena as bibliotecas do Bootstrap utilizadas por todo o sistema. Essa organização facilita a manutenção do código e garante que os estilos e funcionalidades estejam centralizados, promovendo maior consistência visual e estrutural em todas as páginas do site.

A seguir serão apresentadas as páginas em mais detalhes:

Figura 35 - Páginas 4.



Fonte: autoral, 2025.

Figura 36 - Páginas 5.

Script Forge

admin

[Voltar](#)

[admin](#)


[Histórico](#)

[Tabela de Usuários](#)

[Desconectar](#)

Meu perfil

Configure o seu perfil da forma que preferir.
Lembre-se de não compartilhar seus dados pessoais com ninguém.



admin - admin@admin.com

Edite o seu perfil:

Foto:

Nenhum arquivo escolhido

Nome:

e-mail:

senha:

A senha deve ter ao menos 4 caracteres.

Excluir o perfil

ATENÇÃO! Excluir o perfil fara com que você perca todas as suas informações, incluindo seus históricos salvos. Essa ação não pode ser desfeita.

Nos encontre: [✉](#) [🌐](#) [📺](#)

Script Forge © - 2025

Fonte: autoral, 2025.

Figura 37 - Páginas 6.



Script Forge admin 

[Voltar](#)

[admin](#)

[Histórico](#)

[Tabela de Usuários](#)

[Desconectar](#)

Histórico

Vizualize todos os seus roteiros salvos aqui.

Alterado pela ultima vez em 2025-05-19 00:24:40



"Love Blossoms: A Yuri Adventure"

O jogo se passa em um mundo de fantasia onde duas jovens guerreiras, Aria e Luna, são convocadas para proteger o reino de uma terrível ameaça. Aria é uma espadachim habilidosa e determinada, enquanto Luna é uma feiticeira poderosa e misteriosa. Conforme as

Alterado pela ultima vez em 2025-05-18 00:00:00



"A Tale of Pigeon Romance"

No jogo, o jogador assume o papel de um estudante que frequenta uma escola para pombos antropomórficos. A história se passa em um mundo onde os pombos são seres inteligentes e capazes de se comunicar com os humanos. O protagonista é novo na escola e logo descobre que

Alterado pela ultima vez em 2025-05-18 00:00:00

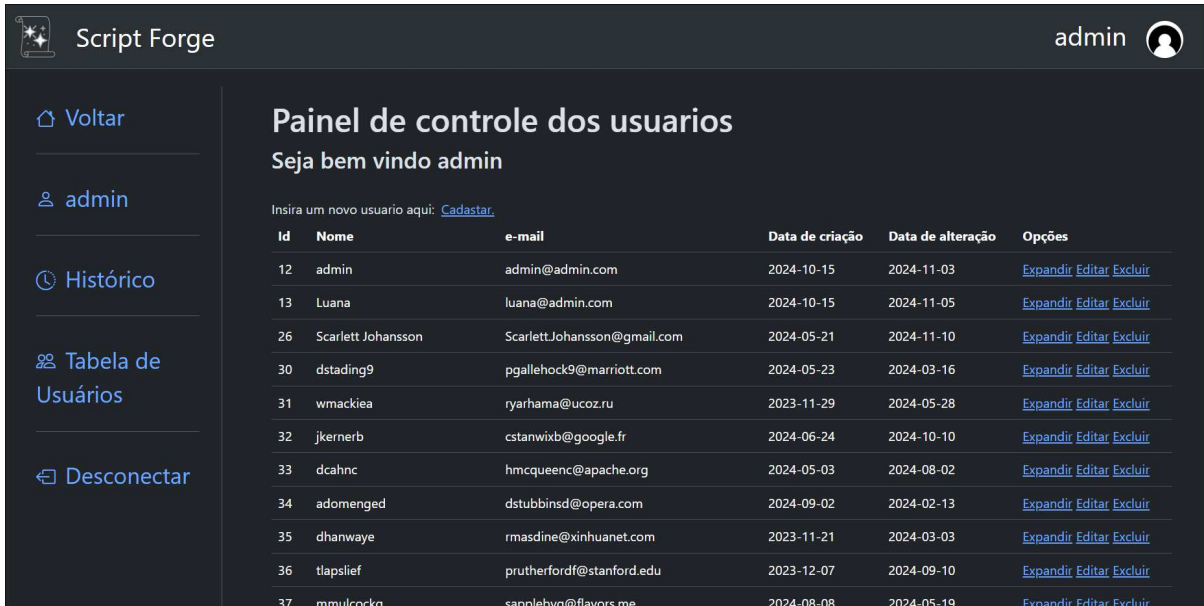



"Lovebirds: A Tale of Pigeon Romance"

No jogo, o jogador assume o papel de um estudante que frequenta uma escola para pombos antropomórficos. A história se passa em um mundo onde os pombos são seres inteligentes e capazes de se comunicar com os humanos. O protagonista é novo na escola e logo

Fonte: autoral, 2025.

Figura 38 - Páginas 7.



Script Forge admin 

[Voltar](#)

[admin](#)

[Histórico](#)

[Tabela de Usuários](#)

[Desconectar](#)

Painel de controle dos usuarios

Seja bem vindo admin

Insira um novo usuario aqui: [Cadastrar](#).

Id	Nome	e-mail	Data de criação	Data de alteração	Opções
12	admin	admin@admin.com	2024-10-15	2024-11-03	Expandir Editar Excluir
13	Luana	luana@admin.com	2024-10-15	2024-11-05	Expandir Editar Excluir
26	Scarlett Johansson	Scarlett.Johansson@gmail.com	2024-05-21	2024-11-10	Expandir Editar Excluir
30	dstading9	pgallehock9@marriott.com	2024-05-23	2024-03-16	Expandir Editar Excluir
31	wmackiea	ryarhama@ucoz.ru	2023-11-29	2024-05-28	Expandir Editar Excluir
32	jkernelb	cstanwixb@google.fr	2024-06-24	2024-10-10	Expandir Editar Excluir
33	dcahnc	hmqueenc@apache.org	2024-05-03	2024-08-02	Expandir Editar Excluir
34	adomenged	dstubbinsd@opera.com	2024-09-02	2024-02-13	Expandir Editar Excluir
35	dhanwaye	rmasdine@xinhuanet.com	2023-11-21	2024-03-03	Expandir Editar Excluir
36	tlapslif	prutherfordf@stanford.edu	2023-12-07	2024-09-10	Expandir Editar Excluir
37	mmulcockg	sapplebyg@flavors.me	2024-08-08	2024-05-19	Expandir Editar Excluir

Fonte: autoral, 2025.

8.3.2 CONSUMO DA API

Por fim, destaca-se a principal funcionalidade do sistema: a integração com a API do ChatGPT, oferecida pela OpenAI. Essa API é o componente central da aplicação, permitindo a comunicação entre o sistema e o modelo de inteligência artificial responsável pela geração de conteúdo textual. Por meio dessa integração, o sistema é capaz de criar roteiros, gerar títulos e elaborar textos de forma automatizada, conforme os parâmetros definidos pelo usuário e pelo próprio fluxo do site.

A API é acionada em três momentos principais do sistema, durante a geração inicial do roteiro, na criação do título correspondente e, por fim, na etapa de elaboração detalhada do conteúdo. Essas interações são realizadas por meio de scripts escritos em JavaScript, que enviam os dados do usuário à API, juntamente com uma chave de autenticação obrigatória para sua execução.

Por razões de segurança, essa chave de API é armazenada no back-end da aplicação, impedindo que seja exposta diretamente no código do front-end. A OpenAI, além de fornecer a API, disponibiliza uma documentação técnica completa que orienta os desenvolvedores quanto ao uso adequado da chave e às boas práticas de segurança. Essa documentação foi fundamental para a implementação segura e eficiente da API no projeto.

É importante destacar que a API possui algumas limitações impostas tanto pela própria OpenAI quanto pelos parâmetros definidos pelo desenvolvedor. Entre essas restrições, está o limite de “tokens”, uma métrica utilizada pela API que representa a quantidade de palavras ou caracteres processados em cada requisição. Esse limite afeta diretamente o tamanho dos textos que podem ser gerados em uma única chamada.

O funcionamento da API, de forma simplificada, consiste em coletar as entradas do usuário por meio do JavaScript, empacotar essas informações junto as

credenciais de autorização, e enviar a requisição para a API. O modelo de IA interpreta os dados e retorna uma resposta: em caso de sucesso, o sistema recebe o texto ou título solicitado; em caso de erro, o retorno traz informações sobre o problema ocorrido.

Vale ressaltar que a API da OpenAI possui outras funcionalidades que não foram exploradas neste projeto, como a geração de imagens, sons e até mesmo músicas. Embora o foco desta aplicação tenha sido exclusivamente na geração de texto, essas possibilidades demonstram o potencial de expansão da plataforma. Durante a fase de testes, o consumo da API manteve-se em níveis baixos. No entanto, é previsto que, com a disponibilização pública do site, seja necessário realizar uma análise mais detalhada dos custos de uso da API, a fim de garantir a viabilidade financeira do projeto em escala real.

9 CONSIDERAÇÕES FINAIS

Inicialmente, o projeto foi concebido como uma atividade prática para uma disciplina específica do curso. No entanto, após uma análise mais aprofundada, percebeu-se que o sistema possuía grande potencial e poderia ser expandido e aprimorado para ser apresentado como Trabalho de Conclusão de Curso (TCC). A partir desse momento, o projeto passou por diversas atualizações e melhorias até atingir sua forma atual.

Durante o processo de desenvolvimento, foram enfrentadas diversas dificuldades, como desafios técnicos, como a integração da API, e a conexão do banco de dados com o site. No entanto, com esforço, dedicação e persistência, foi possível superar esses obstáculos e conduzir o projeto até sua conclusão da melhor maneira possível.

Este trabalho representa a aplicação prática dos conhecimentos adquiridos ao longo do curso, mesmo que nem todas as disciplinas estivessem diretamente relacionadas ao projeto. Ainda que matérias como hardware ou microinformática não tenham tido impacto direto no desenvolvimento, disciplinas como Banco de Dados e Lógica de Programação mostraram-se essenciais para a construção do sistema. Assim, este projeto evidencia não apenas o domínio técnico adquirido, mas também a capacidade dos integrantes de aplicarem seus conhecimentos em um contexto real de desenvolvimento de software.

Embora o sistema tenha atingido seu objetivo principal dentro do escopo do TCC, há a intenção de continuar seu desenvolvimento futuramente, incorporando novas funcionalidades e aprimorando sua estrutura, a fim de transformá-lo em uma plataforma ainda mais completa e funcional.

REFERÊNCIAS

42FILMES. **Como a Inteligência Artificial vai mudar a produção audiovisual.** 42 Filmes. 2023. Disponível em: <https://42filmes.com.br/2023/10/11/como-a-inteligencia-artificial-vai-mudar-a-producao-audiovisual/>. Acesso em: 22 mai 2025.

BUENO, Felipe. **A expansão do mercado de games brasileiro se deve a mudanças no modo tradicional do trabalho.** Jornal da USP. 2024. Disponível em: <https://jornal.usp.br/radio-usp/a-expansao-do-mercado-de-games-brasileiro-se-deve-a-mudancas-no-modo-tradicional-do-trabalho/>. Acesso em: 12 mai 2025.

CHAHAD, José Paulo Zeetano. **Avanços recentes da inteligência artificial: impactos trabalhistas, sociais e regulatórios.** Jornal da USP. 2024. Disponível em: <https://jornal.usp.br/artigos/avancos-recentes-da-inteligencia-artificial-impactos-trabalhistas-sociais-e-regulatorios/>. Acesso em: 22 mai 2025.

DIO. **Visual Studio Code: Um excelente editor de código-fonte.** DIO, 2021 (aproximadamente). Disponível em: <https://www.dio.me/articles/visual-studio-code-um-excelente-editor-de-codigo-fonte>. Acesso em: 4 jun 2025.

EBAC. **O que é GitHub:** para que serve, como funciona e como utilizar. EBAC Online, s.d. 2023. Disponível em: <https://ebaonline.com.br/blog/o-que-e-github>. Acesso em: 4 jun 2025.

EBAC. **Os impactos da IA na criação de jogos.** EBAC Online, s.d. 2024. Disponível em: <https://ebaonline.com.br/blog/impactos-da-ia-seo>. Acesso em: 14 set 2024.

FERREIRA, Rafael. **Pensando em desistir de fazer jogos? Lições de 5 desenvolvedores que superaram todas as dificuldades.** Produção de Jogos. 2021. Disponível em: <https://producaodejogos.com/pensando-em-desistir-de-fazer-jogos-licoes-de-5-desenvolvedores-que-superaram-todas-as-dificuldades/>. Acesso em: 12 mai 2025.

FLORIANO, Jeronima. **O que são requisitos de software e por que é importante entendê-los.** DEV Community. 2023. Disponível em: <https://dev.to/jeronimafloriano/o-que-sao-requisitos-de-software-e-porque-e-importante-entende-los-bjn>. Acesso em: 22 mai 2025.

GARCIA, Guilherme. **XAMPP: O que é, como funciona, vantagens e instalação da ferramenta.** Mercado Online Digital, 22 maio 2024. Disponível em: <https://mercadoonlinedigital.com/blog/xampp/>. Acesso em: 4 jun 2025.

JEAN. **Os temas dos jogos estão saturados?** Covil dos Jogos. 2021. Disponível em: <https://covildosjogos.com.br/2021/05/10/os-temas-dos-jogos-estao-saturados/>. Acesso em: 12 mai 2025.

LUCIDCHART. **O que é UML?** Lucidchart. 2025 Disponível em: <https://www.lucidchart.com/pages/pt/o-que-e-uml>. Acesso em: 22 mai 2025.

MULLIGAN, Scott J. **Qual o impacto da IA no desenvolvimento de videogames?** MIT Technology Review Brasil. 2024. Disponível em: <https://mittechreview.com.br/ia-desenvolvimento-videogames/>. Acesso em: 14 set 2024.

TECHTUDO. **Draw.io:** editor de diagramas e fluxogramas online grátis. TechTudo. 2025. Disponível em: <https://www.techtudo.com.br/tudo-sobre/drawio/>. Acesso em: 22 mai 2025.