

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO  
RIO GRANDE DO SUL  
CAMPUS CANOAS  
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE  
SISTEMAS

ARTUR FERNANDES GUIMARÃES

**Um Estudo sobre a Capacidade de Geração de Código do ChatGPT em  
Relação ao Desenvolvedor Humano na Criação de um Sistema.**

Canoas, RS  
2024.

ARTUR FERNANDES GUIMARÃES

**Um Estudo sobre a Capacidade de Geração de Código do ChatGPT em  
Relação ao Desenvolvedor Humano na Criação de um Sistema.**

Trabalho de Conclusão de Curso  
apresentado como requisito parcial para  
obtenção do título de Tecnólogo em  
Análise e Desenvolvimento de Sistemas  
pelo Instituto Federal de Educação,  
Ciência e Tecnologia do Rio Grande do  
Sul – Campus Canoas.

Professor Doutor Rafael Coimbra Pinto  
Orientador

Canoas, RS  
2024.



**Ministério da Educação**  
**Secretaria de Educação Profissional, Científica e Tecnológica**  
**Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul**  
**Campus Canoas**

**ATA DE DEFESA PÚBLICA DO TRABALHO DE CONCLUSÃO DE CURSO**

Aos 29 dias do mês de julho de 2024, às 21:00 horas, em sessão pública no Google Meet, na presença da Banca Examinadora presidida pelo(a) Professor(a):

Rafael Coimbra Pinto e

composta pelos examinadores:

1. Denise Regina Pechmann

2. Igor Lorenzato Almeida

3. \_\_\_\_\_,

o(a) aluno(a) Artur Fernandes Guimarães

apresentou o Trabalho de Conclusão de Curso intitulado:

Um Estudo sobre a Capacidade de Geração de Código do ChatGPT em Relação ao Desenvolvedor Humano na Criação de um Sistema

como requisito curricular indispensável para a integralização do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas. Após reunião em sessão reservada, a Banca Examinadora deliberou e decidiu pela Aprovação do referido trabalho, divulgando o resultado formalmente ao aluno e demais presentes e eu, na qualidade de Presidente da Banca, lavrei a presente ata que será assinada por mim, pelos demais examinadores e pelo aluno.



Documento assinado digitalmente  
**RAFAEL COIMBRA PINTO**  
Data: 29/07/2024 22:07:36-0300  
Verifique em <https://validar.iti.gov.br>

\_\_\_\_\_  
Presidente da Banca Examinadora

Documento assinado digitalmente

\_\_\_\_\_  
Examinador 01



**DENISE REGINA PECHMANN**  
Data: 29/07/2024 22:25:49-0300  
Verifique em <https://validar.iti.gov.br>

\_\_\_\_\_  
Examinador 02



Documento assinado digitalmente  
**IGOR LORENZATO ALMEIDA**  
Data: 29/07/2024 22:46:01-0300  
Verifique em <https://validar.iti.gov.br>

\_\_\_\_\_  
Examinador 03

Documento assinado digitalmente

\_\_\_\_\_  
Aluno



**ARTUR FERNANDES GUIMARAES**  
Data: 30/07/2024 17:40:20-0300  
Verifique em <https://validar.iti.gov.br>

## **AGRADECIMENTOS**

Deixo aqui meus agradecimentos aos meus pais, Marlise e Ricardo, pelo apoio, pelas caronas e pelo meu primeiro computador, é tudo graças a vocês. A Isadora, por todo o suporte e amor, além dos planos a serem cumpridos. Aos meus incontáveis amigos que me acompanharam até aqui, em especial ao Davi e ao Felipe, que me ajudaram durante todo o curso.

Gostaria de agradecer profundamente ao Prof. Rafael pelos ensinamentos, pela paciência, pelos puxões de orelha e, principalmente, por ter confiado em mim na conclusão deste trabalho.

Por fim, à minha avó, Cecília, que sempre me aplaudiu em tudo que fiz desde que eu era uma criança. Tenho certeza que desta vez não será diferente, de onde quer que ela esteja me cuidando.

## RESUMO

O ChatGPT, uma ferramenta de Processamento de Linguagem Natural, tem sido amplamente adotado no desenvolvimento de software devido à sua capacidade de gerar código a partir de requisitos em linguagem natural. Este estudo compara dois sistemas de loja virtual, um desenvolvido manualmente e outro inteiramente gerado pelo ChatGPT, focando na eficiência, qualidade do código e usabilidade. Ambos os sistemas utilizaram PHP, MySQL, HTML, CSS e JavaScript. As comparações técnicas utilizando SonarQube mostraram uma quantidade menor de falhas no sistema feito pelo ChatGPT, porém com muito mais redundâncias em relação ao código manualmente desenvolvido que, além disso, se mostrou mais aceito na pesquisa feita com usuários. Os resultados sugerem que a adaptação do desenvolvedor a essas novas tecnologias se mostra cada vez mais fundamental.

**Palavras-Chave:** Inteligência Artificial, ChatGPT, Desenvolvimento de Software, Qualidade de Código.

## ABSTRACT

ChatGPT, a Natural Language Processing tool, has been widely adopted in software development due to its ability to generate code from natural language requirements. This study compares two virtual store systems, one manually developed and the other entirely generated by ChatGPT, focusing on efficiency, code quality, and usability. Both systems utilized PHP, MySQL, HTML, CSS, and JavaScript. SonarQube technical comparisons revealed less issues on the ChatGPT generated code, but also a lot more duplications than the manually developed code, which also appeared to be much preferred among the users. The results suggest that the developer's adaptation to these new technologies is becoming increasingly essential.

**Keywords:** Artificial Intelligence, ChatGPT, Software Development, Code Quality.

## LISTA DE FIGURAS

Figura 1 - Diagrama de Casos de Uso do Sistema Manualmente Desenvolvido.....	16
Figura 2 - Diagrama de Atividades Manualmente Desenvolvido.....	18
Figura 3 - Diagrama de Entidade Relacionamento do Sistema Manualmente Desenvolvido	19
Figura 4 - PhpMyAdmin do Sistema Manualmente Desenvolvido.....	20
Figura 5 - Tela Inicial da Loja Ruydo Desenvolvida Manualmente.....	21
Figura 6 - Primeira interação com o ChatGPT.....	23
Figura 7 - Diagrama de Casos de Uso Gerado pelo ChatGPT.....	26
Figura 9 - Diagrama de Atividades Gerado pelo ChatGPT.....	28
Figura 10 - Diagrama Entidade-Relacionamento Gerado pelo ChatGPT.....	30
Figura 11 - Primeira Correção Solicitada a Ferramenta.....	31
Figura 12 - Resultado da Execução do Primeiro Código SQL Providenciado pelo ChatGPT....	32
Figura 13 - Primeiro Caso de Alucinação do ChatGPT.....	33
Figura 14 - Primeira Estética Gerada pelo ChatGPT.....	35
Figura 15 - Quarta Estética Gerada pelo ChatGPT.....	35
Figura 16 - Correção de Falha Funcional do Sistema.....	36
Figura 17 - Requisição de Adição de Produtos no Catálogo.....	37
Figura 17 - Adicionando Produtos Ao Catálogo via ChatGPT.....	38
Figura 19 - Resultado Final das Telas Principais de Ambos os Sistemas.....	39
Figura 20 - Análise Via SonarQube do Código Manualmente Desenvolvido.....	41
Figura 21 - Análise via SonarQube do Código Gerado pelo ChatGPT.....	41
Figura 22 - Modelo Likert das Respostas do SUS.....	43
Figura 23 - Exemplo de Pergunta Positiva do Questionário de Comparação dos Sistemas.	44
Figura 24 - Exemplo de Pergunta Negativa do Questionário de Comparação dos Sistemas...	44

## LISTA DE TABELAS

Tabela 1 - Descrição dos Casos de Uso do Sistema Manualmente Desenvolvido.....	16
Tabela 2 - Requisitos Funcionais e Não Funcionais Gerados pelo ChatGPT.....	23
Tabela 3 - Análise dos Dados Recolhidos via SonarQube.....	41
Tabela 4 - Pontuação do Questionário de Comparação dos Sistemas.....	44

## LISTA DE ABREVIATURAS E SIGLAS

PLN	Processamento de Linguagem Natural
IA	Inteligência Artificial
GPT	<i>Generative Pre-trained Transformer</i>
SQL	<i>Structured Query Language</i>
PHP	<i>PHP: Hypertext Preprocessor</i>
HTML	<i>HyperText Markup Language</i>
CSS	<i>Cascading Style Sheets</i>
SUS	<i>System Usability Scale</i>
SIS	Sistema



## SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>10</b>
1.1 OBJETIVO GERAL.....	11
1.2 OBJETIVOS ESPECÍFICOS.....	11
<b>2 FUNDAMENTAÇÃO TEÓRICA.....</b>	<b>12</b>
2.1 PROCESSAMENTO DE LINGUAGEM NATURAL (PLN).....	12
2.2 AUTOMAÇÃO E PRODUTIVIDADE NO DESENVOLVIMENTO DE SOFTWARE:...	13
2.3 IMPACTO ECONÔMICO E SOCIAL DA INTELIGÊNCIA ARTIFICIAL.....	14
<b>3 METODOLOGIA.....</b>	<b>16</b>
3.1 DESENVOLVIMENTO MANUAL.....	17
3.1.1 ANÁLISE DE REQUISITOS.....	17
3.1.2 IMPLEMENTAÇÃO DO SISTEMA MANUALMENTE DESENVOLVIDO.....	21
3.2 DESENVOLVIMENTO COM CHATGPT.....	22
3.2.1 DIAGRAMAS GERADOS.....	23
3.2.2 IMPLEMENTAÇÃO DO SISTEMA GERADO PELO CHATGPT.....	31
<b>4 COMPARAÇÃO TÉCNICA.....</b>	<b>41</b>
<b>5 COMPARAÇÃO DOS USUÁRIOS.....</b>	<b>44</b>
<b>6 CONCLUSÃO.....</b>	<b>47</b>
<b>APÊNDICE A - TABELAS DE OBJETIVOS PRINCIPAIS DO CHATGPT.....</b>	<b>52</b>
<b>APÊNDICE B - DEMAIS RESULTADOS E INTERAÇÕES DA CHATGPT.....</b>	<b>55</b>
<b>APÊNDICE C - FORMULÁRIO DE COMPARAÇÃO DOS SISTEMAS.....</b>	<b>66</b>

## 1 INTRODUÇÃO

A inteligência artificial (IA) tem sido um tema de grande interesse nas últimas décadas, com o potencial de revolucionar diversos setores da sociedade. Um dos principais ramos dessa área tão crescente é o Processamento de Linguagem Natural (PLN), que estuda a linguagem dos seres humanos e busca criar ferramentas para auxiliar na interação, comunicação e análise da mesma (PINTO, 2015). Entre essas ferramentas, está o ChatGPT, um sistema de diálogo processador de linguagem natural treinado com uma vasta quantidade de dados pré-existentes coletados da internet. Ele consegue assimilar a linguagem humana e interagir com o usuário, respondendo a perguntas sobre os mais diversos temas e áreas. Seu lançamento foi um marco no mundo da tecnologia, influenciando inúmeras áreas e, por se tratar de um sistema gratuito, atingiu grande parte da população, batendo o recorde de aplicativo mais rápido a alcançar a marca de 100 milhões de usuários ativos (REUTERS, 2023).

O uso de IA, especialmente do ChatGPT, tem ganhado destaque no desenvolvimento de *software*. Desenvolvedores adotam essa plataforma para auxiliar na criação de sistemas, devido à sua capacidade de interpretar e gerar código a partir de requisitos simples em linguagem natural. No entanto, isso também traz preocupações sobre o futuro do trabalho e a economia, com debates acalorados no meio acadêmico e crítico sobre os riscos e desafios associados ao uso da IA, além de suas potencialidades para aumentar a produtividade e a qualidade do código (ACEMOGLU, RESTREPO, 2018).

A comparação entre sistemas desenvolvidos pelo ChatGPT e aqueles criados manualmente é de extrema relevância para a indústria de *software*. Essa comparação não só avalia a velocidade e o custo de desenvolvimento, mas também a acessibilidade e a facilidade de manutenção dos sistemas gerados por IA. Além disso, as implicações éticas e sociais, como o impacto no emprego e a responsabilidade pelo código gerado, são considerações cruciais.

Este projeto teve como objetivo avaliar e comparar o desempenho de um sistema de loja virtual desenvolvido pelo ChatGPT com um sistema criado manualmente. Os resultados deste trabalho são fundamentais para aprofundar o entendimento sobre o potencial e as limitações da utilização da inteligência artificial no desenvolvimento de *software*.

## 1.1 OBJETIVO GERAL

Estudar, documentar e comparar a independência e efetividade do ChatGPT, em relação a um código feito por um humano, no desenvolvimento de um sistema web, desde a definição dos requisitos, passando pelo desenvolvimento, até os testes e usabilidade do usuário.

## 1.2 OBJETIVOS ESPECÍFICOS

Para se atingir o objetivo geral, pode-se citar os seguintes objetivos específicos para o projeto:

- Desenvolver um sistema *web* para servir de comparação com o ChatGPT;
- Avaliar a capacidade da IA em propor e listar requisitos de sistema apresentados em linguagem natural, identificando sua eficácia na tradução desses requisitos em código funcional;
- Extrair e analisar requisitos e diagramas teóricos;
- Integrar códigos gerados das diferentes partes do sistema e analisar sua compatibilidade;
- Documentar as perguntas, respostas e auto-correções desenvolvidas pela ferramenta;
- Analisar a capacidade da ferramenta em suprir os requisitos gerados;
- Comparar qualidade do código do sistema gerado com as do autor;
- Reunir dados de usuários em relação a usabilidade dos sistemas.

## 2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão explorados os pilares que sustentam a discussão sobre o papel e a adesão da inteligência artificial no âmbito do desenvolvimento de software, explicando o funcionamento das ferramentas de PLN, suas aplicações atuais na área da tecnologia e o debate social-econômico envolvido nesse processo.

### 2.1 PROCESSAMENTO DE LINGUAGEM NATURAL (PLN)

O Processamento de Linguagem Natural é uma subárea da inteligência artificial que se concentra na interação entre computadores e humanos por meio da linguagem natural. GONZALEZ e LIMA (2003) apontam que o PLN extrai o conjunto de informações por meio dos mais diversos aspectos da comunicação humana, como sons, palavras, sentenças e discursos, e que, separando esses aspectos em níveis sequenciais, pode-se chegar a fórmulas lógicas na comunicação.

O ChatGPT funciona com base na arquitetura GPT (Generative Pre-trained Transformer), desenvolvida pela OpenAI<sup>1</sup>. Essa arquitetura utiliza um método de aprendizado profundo chamado *Transformer*, que é particularmente eficaz para processar e gerar textos. Os modelos GPT são treinados em grandes conjuntos de dados textuais, coletados da internet, e aprendem a prever o próximo *token* em uma frase. Um *token* pode ser uma palavra, uma frase, um caractere ou até mesmo um subconjunto de caracteres, dependendo do contexto e da aplicação. Essas unidades menores são então utilizadas como entradas para modelos de IA, onde são processadas e analisadas para inferências ou geração de texto (SOUZA, 2024). Este processo, conhecido como "pré-treinamento", capacita o modelo a compreender contextos variados e a gerar texto coerente e relevante.

Após o pré-treinamento, a ferramenta passa por uma fase de "ajuste fino" (*fine-tuning*), onde o modelo é adaptado para tarefas específicas. No caso do ChatGPT, essa fase envolve o uso de Aprendizado por Reforço com Feedback Humano. Neste processo, os treinadores humanos fornecem demonstrações e classificações das respostas geradas pelo modelo, orientando-o a produzir

---

<sup>1</sup> <https://help.openai.com/en/articles/6783457-what-is-chatgpt>

respostas que sejam mais úteis e seguras. Esta técnica ajuda a alinhar o comportamento do modelo com as expectativas humanas, tornando suas interações mais naturais e satisfatórias (GUINNESS, 2024).

## **2.2 AUTOMAÇÃO E PRODUTIVIDADE NO DESENVOLVIMENTO DE SOFTWARE:**

A automação no desenvolvimento de software está transformando a forma como os programas são criados, oferecendo ferramentas que aumentam a produtividade e reduzem erros. Tecnologias como o ChatGPT prometem facilitar tarefas como geração de código e sugestão de melhorias, tornando o processo de desenvolvimento mais ágil. No entanto, embora essas ferramentas sejam valiosas, elas ainda enfrentam desafios significativos, como a criação de códigos otimizados e a capacidade de desenvolver sistemas complexos de forma autônoma. Portanto, a automação não elimina a necessidade de criatividade e julgamento humano no desenvolvimento de software.

Em MOTTA et. al., 2023, os autores discutem o potencial do ChatGPT como uma ferramenta para auxiliar no desenvolvimento de aplicações web. Eles argumentam que a ferramenta pode ajudar a aumentar a produtividade dos desenvolvedores, reduzir o tempo de desenvolvimento e melhorar a qualidade do código. Os autores apresentam uma série de exemplos de como o gerador de texto pode ser usado para auxiliar no desenvolvimento de aplicações *Web*. Ele pode, por exemplo, ser usado para responder a perguntas sobre sintaxe, semântica e padrões de design de programação. Também pode ser usado para gerar código, verificar erros e sugerir melhorias. Os autores concluem que é uma tecnologia promissora que tem o potencial de revolucionar o desenvolvimento de aplicações web. Eles recomendam que os desenvolvedores o explorem e considerem como ele pode ser usado para melhorar sua produtividade e qualidade de trabalho (MOTTA et al., 2023).

IGNÁCIO et al. (2024) destaca debates sobre a possibilidade de substituição dos programadores por IAs especializadas, observando que, embora as ferramentas de IA, como o GitHub Copilot e o ChatGPT, possam gerar código e acelerar o desenvolvimento, elas frequentemente produzem códigos mal otimizados e com falhas. O GitHub Copilot, por exemplo, ajuda na auto-completação de código e na

geração de sugestões, mas não consegue criar sistemas complexos de forma autônoma, enquanto o ChatGPT oferece respostas e exemplos de código, mas ainda não pode desenvolver aplicações completas. Apesar dessas limitações, as IAs são ferramentas valiosas para automatizar e melhorar processos em diversas áreas, como saúde e educação, e ainda não substituem a necessidade de julgamento e criatividade humanas no desenvolvimento de software.

Entretanto, é importante destacar que existem ferramentas que utilizam modelos de linguagem como um componente e conseguem desenvolver software de forma mais autônoma, como a Devin AI. No entanto, o presente trabalho avalia a geração de software pela interface padrão do ChatGPT, utilizando prompts manuais, e não de forma totalmente autônoma como essas outras ferramentas.

### **2.3 IMPACTO ECONÔMICO E SOCIAL DA INTELIGÊNCIA ARTIFICIAL**

A Inteligência Artificial é caracterizada por tentar resolver problemas que são complexos para algoritmos convencionais e envolve modelos e técnicas como os paradigmas simbólico, conexionista, evolutivo e probabilístico. O conceito de agentes inteligentes, que são sistemas autônomos capazes de interagir e tomar decisões, tem sido crucial para a integração de técnicas de IA. Contudo, a implementação da IA levanta questões éticas e práticas significativas, como a responsabilidade, transparência e impactos socioeconômicos. Garantir que a IA seja desenvolvida e utilizada de forma responsável é essencial para maximizar seus benefícios e minimizar riscos (SICHMAN, 2021).

TRABELSI (2024) examina o impacto da inteligência artificial (IA) no crescimento econômico, transformação das habilidades e revolução tecnológica. Destaca que, embora a IA possa aumentar a produtividade e criar novos mercados, também pode causar disrupções significativas, como a ampliação da desigualdade entre países e setores. A IA influencia profundamente o mercado de trabalho, substituindo tarefas e exigindo requalificação da força de trabalho. Além disso, a IA melhora a gestão de riscos e promove inovação, mas enfrenta desafios como questões éticas e a necessidade de regulamentação. Em conclusão, o artigo enfatiza que, para maximizar os benefícios da IA e minimizar seus impactos

negativos, é crucial abordar questões éticas, garantir acesso equitativo e implementar regulamentações adequadas.

Além disso, GARCIA (2020) alerta para o risco de vieses sociais e preconceitos embutidos nos dados utilizados por sistemas de IA, que podem ser perpetuados e amplificados por algoritmos. A autora destaca casos de discriminação racial e de gênero, enfatizando a necessidade de uma abordagem ética no desenvolvimento e aplicação desses sistemas, além de chamar a atenção para a importância da transparência e do escrutínio externo dos dados usados.

### 3 METODOLOGIA

Nesta pesquisa, um sistema web de loja virtual de artigos musicais chamado “Ruydo” foi desenvolvido. Primeiro, o sistema foi criado manualmente pelo autor, que seguiu um processo estruturado envolvendo a definição de requisitos, a elaboração de diagramas de casos de uso, a criação do modelo Entidade-Relacionamento e a implementação do código. Em seguida, o mesmo sistema foi solicitado ao ChatGPT, que realizou as mesmas etapas, desde a definição de requisitos até as configurações da interface da aplicação, documentando as interações e os objetivos específicos requisitados à ferramenta. A versão do ChatGPT utilizada foi a 4o, a última disponível no momento da pesquisa.

Adicionalmente, é fundamental reconhecer as limitações da metodologia deste estudo. Primeiramente, apenas um tipo de sistema foi comparado, desenvolvido por uma única pessoa, o que pode não refletir a ampla gama de aplicações possíveis. Em segundo lugar, a eficácia da construção do sistema pela IA depende da habilidade do pesquisador em utilizar a IA de forma eficiente. Por fim, há o viés do pesquisador, tanto no desenvolvimento quanto na análise dos resultados, que pode influenciar as conclusões do estudo.

Ambos os sistemas, após estarem prontos e disponibilizados online<sup>23</sup>, passaram por uma pesquisa com diversos entrevistados por meio de um questionário produzido na plataforma Google Forms<sup>4</sup> para testar suas principais funcionalidades, adaptação dos usuários, interface e demais opiniões sobre a utilização das aplicações. Os sistemas foram hospedados via ferramenta Git<sup>5</sup>, na plataforma GitHub<sup>67</sup> e, então, analisados pelo SonarQube, uma das ferramentas de análise de código mais utilizadas do mundo tanto no meio acadêmico quanto no meio industrial (LOMIO et. al, 2022)

---

<sup>2</sup> Site manualmente desenvolvido: <http://lojaa.infinityfreeapp.com/views/index.html>

<sup>3</sup> Site desenvolvido via ChatGPT: <http://lojabb.infinityfreeapp.com/?i=1>

<sup>4</sup> <https://www.google.com/intl/pt-BR/forms/about/>

<sup>5</sup> <https://git-scm.com/>

<sup>6</sup> Sistema feito via chatGPT: <https://github.com/ArturFernandes/harmoniaShop>

<sup>7</sup> Sistema desenvolvido manualmente: <https://github.com/ArturFernandes/Ruydo>



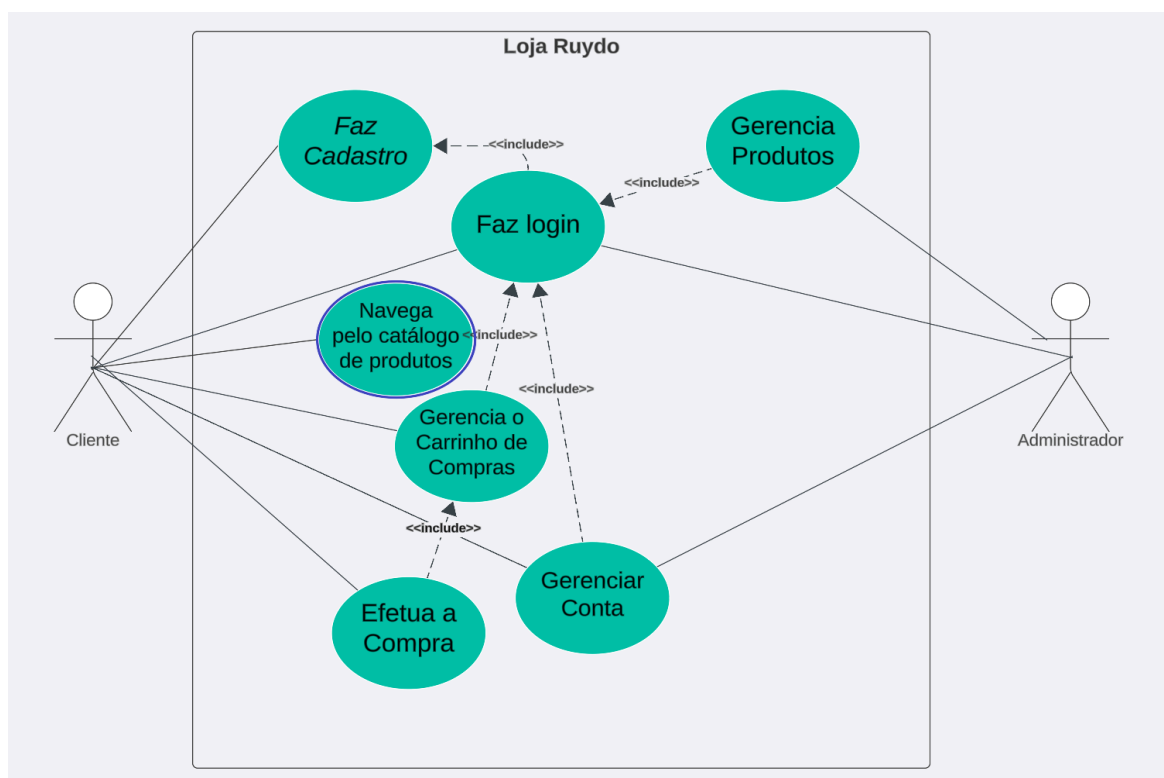
### 3.1 DESENVOLVIMENTO MANUAL

Nesta seção será documentado o processo de desenvolvimento manual do sistema por parte do autor, a análise de requisitos, a escolha das ferramentas e a implementação do código.

#### 3.1.1 ANÁLISE DE REQUISITOS

Para dar início ao desenvolvimento, foi estabelecido um diagrama de casos de uso (ver Figura 1), que inclui-se na fase de análise de requisitos. Essa fase é uma etapa crucial para o sucesso do projeto, pois define as bases sobre as quais o sistema será construído, minimizando ambiguidades e proporcionando uma visão clara das expectativas. Este processo envolve a colaboração entre desenvolvedores, analistas e usuários finais para assegurar que o sistema atenda às necessidades reais e específicas. Segundo Barros (2009), entender exatamente o que o sistema deve ser capaz de fazer é fundamental para a sua efetividade e usabilidade.

Figura 1 - Diagrama de Casos de Uso do Sistema Manualmente Desenvolvido



Fonte: Elaborado pelo autor, 2024.

O diagrama de casos de uso apresentado na Figura 1 ilustra as principais interações entre os usuários (clientes e administradores) e o sistema. Cada caso de uso representa uma funcionalidade distinta que o sistema deve oferecer. A seguir, a Tabela 1 detalha cada um desses casos de uso, fornecendo uma visão abrangente das operações e suas extensões.

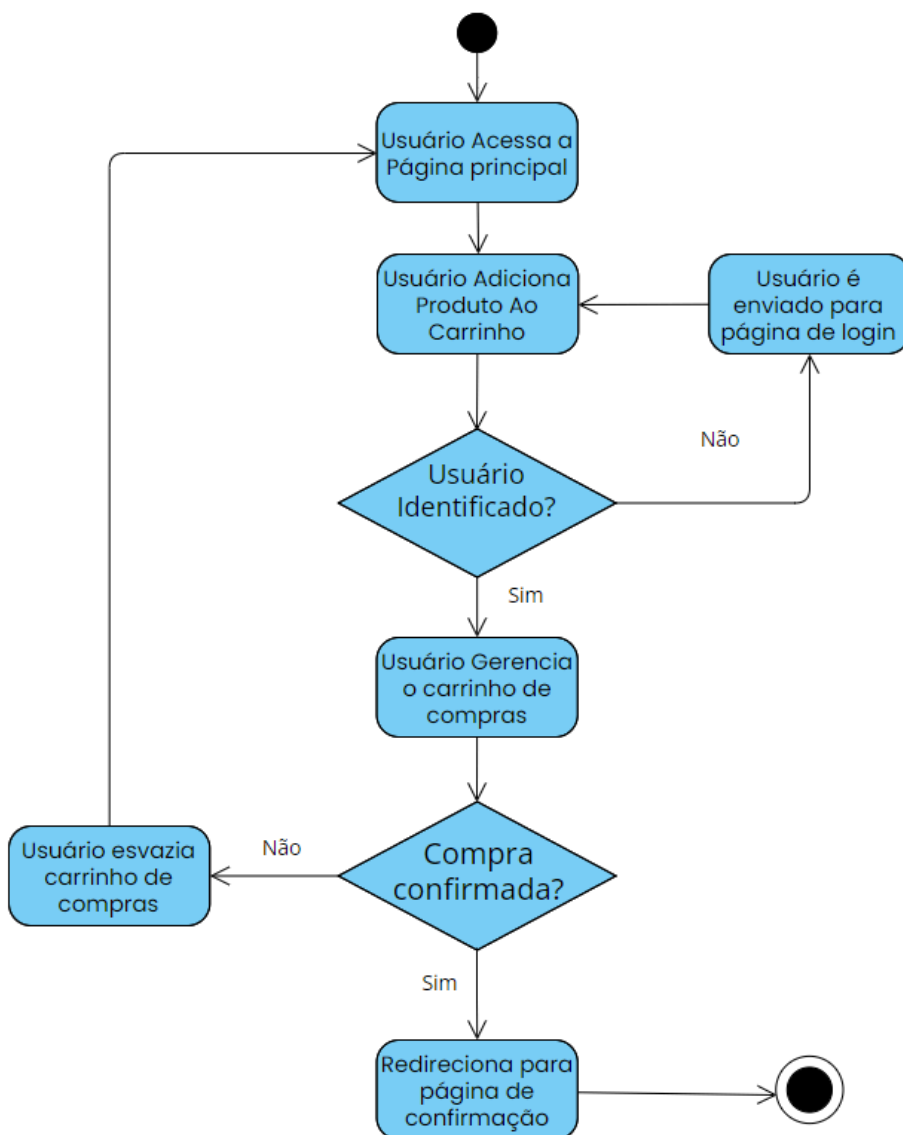
Tabela 1 - Descrição dos Casos de Uso do Sistema Manualmente Desenvolvido

Referência	Caso de Uso	Descrição	Extensão
UC-01	Fazer Cadastro	Permite que um novo usuário crie uma conta no sistema.	UC-02
UC-02	Fazer Login	Permite que um usuário registrado acesse sua conta no sistema	
UC-03	Navegar pelo Catálogo de Produtos	Permite que o usuário navegue pelo catálogo de produtos disponíveis no sistema	UC-02
UC-04	Gerenciar o Carrinho de Compras	Permite que o usuário adicione, remova ou modifique a quantidade de produtos no seu carrinho de compras.	UC-02, UC-03
UC-05	Efetuar a Compra	Permite que o usuário finalize a compra dos produtos adicionados ao carrinho.	UC-04
UC-06	Gerenciar Conta	Permite que o usuário altere informações da sua conta, como dados pessoais e senha.	UC-02
UC-07	Gerenciar Produtos	Permite que o administrador adicione, remova ou altere informações sobre os produtos disponíveis.	

Fonte: Elaborado pelo Autor, 2024.

Esses casos de uso formam a base do desenvolvimento do sistema, garantindo que todas as necessidades dos usuários sejam atendidas de maneira eficaz e eficiente. Com esse diagrama, é possível chegar a um Diagrama de Atividades, que é um modelo de diagramas utilizado pela indústria de desenvolvimento de software para representar a interação dentro das funcionalidades de forma simples e clara, mostrado na Figura 2.

Figura 2 - Diagrama de Atividades Manualmente Desenvolvido



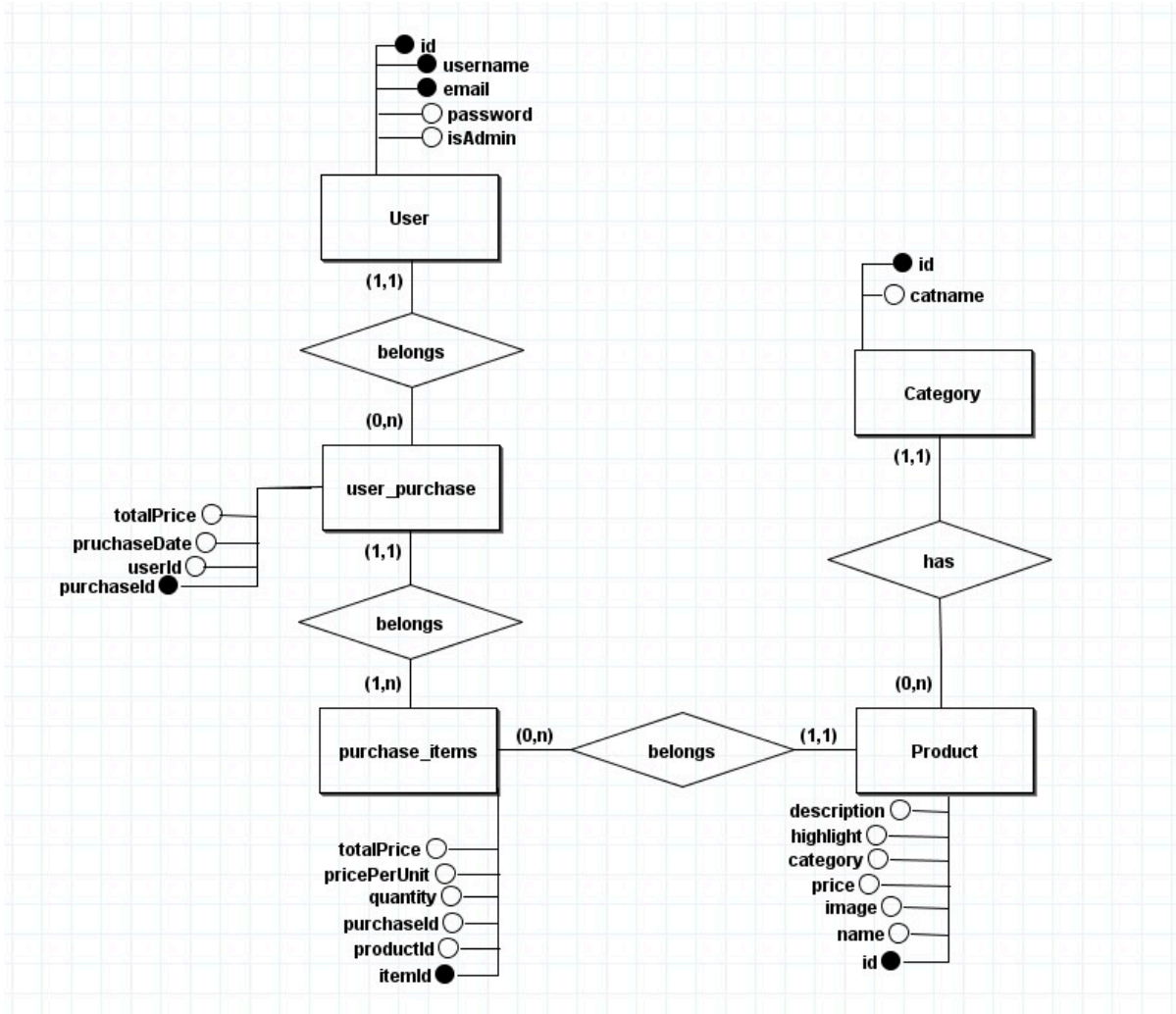
Fonte: Elaborado pelo Autor, 2024.

Esse diagrama serviu para estabelecer o fluxo de usabilidade do sistema, definindo o caminho pelo qual o usuário percorre, desde o acesso da página principal, passando por confirmações de autenticação e, finalmente, na finalização da compra.

Para estruturar o banco de dados, foi estabelecido um Diagrama Entidade Relacionamento (ER), representado na Figura 3. A modelagem ER, introduzida por Peter Chen em 1976, continua a ser popular devido à sua simplicidade e legibilidade, sendo utilizada tanto por desenvolvedores quanto por usuários finais para entender e representar o mundo real em um modelo de dados conceitual,

sendo amplamente ensinado nas instituições de ensino superior (FRANK et al, 2021).

Figura 3 - Diagrama de Entidade Relacionamento do Sistema Manualmente Desenvolvido



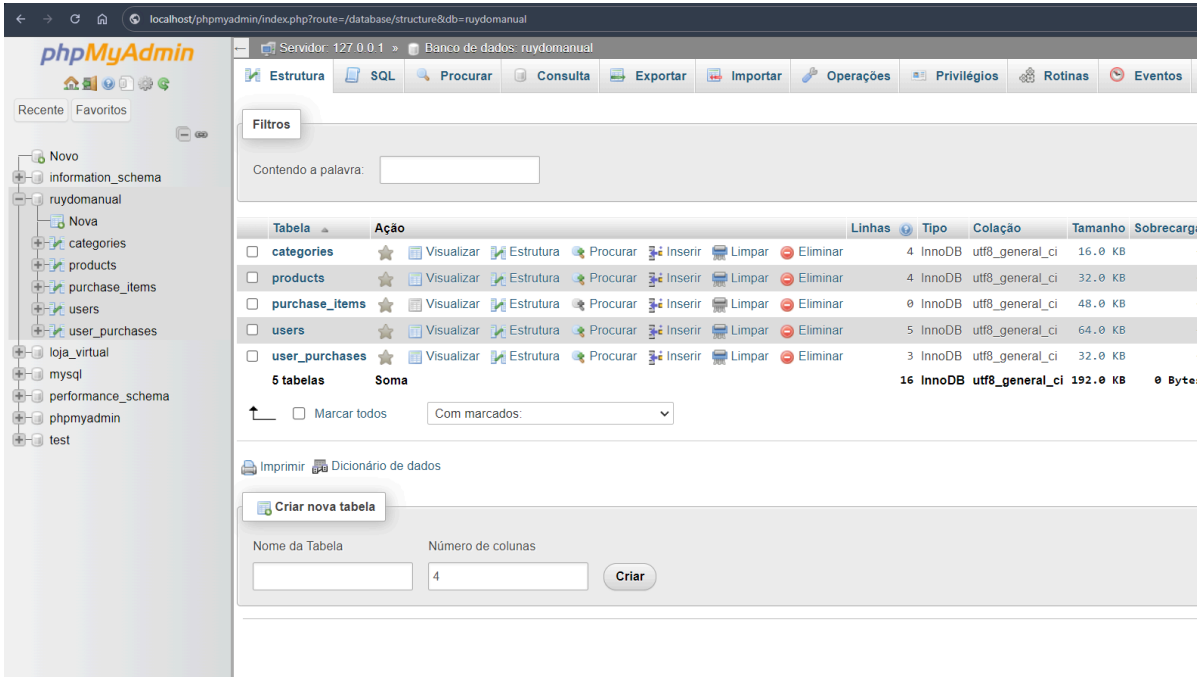
Fonte: Autor, 2024

O Diagrama de Entidade-Relacionamento (ER) é uma representação visual de como os dados são estruturados e inter-relacionados dentro do sistema de informação. Nele, foram definidas entidades como "User", "Category", "Product", "user\_purchase", e "purchase\_items", cada uma com seus respectivos atributos. Por exemplo, a entidade "User" possui atributos como "id", "username", e "email". Os relacionamentos entre as entidades mostram como um "User" pode fazer várias "user\_purchase", e como cada compra pode conter vários "purchase\_items", que se referem a diferentes "Product". Esse tipo de modelagem é crucial para garantir que o banco de dados represente fielmente as necessidades do usuário e permita uma gestão eficiente dos dados.

### 3.1.2 IMPLEMENTAÇÃO DO SISTEMA MANUALMENTE DESENVOLVIDO

O ambiente de desenvolvimento escolhido foi o XAMPP<sup>8</sup>, um pacote de software que inclui o servidor Apache, o banco de dados MySQL (gerenciado via PHPMyAdmin<sup>9</sup>, ver Figura 4) e interpretadores para scripts em PHP<sup>10</sup>. PHP (Hypertext Preprocessor) é uma linguagem de script de código aberto amplamente utilizada para o desenvolvimento de aplicações web. Ela permite a criação de páginas dinâmicas e interativas ao interagir diretamente com o banco de dados. O sistema de gerenciamento de banco de dados escolhido foi o MySQL, um sistema de gerenciamento de banco de dados relacional também de código aberto, que destaca-se por ser rápido e bem otimizado (MYSQL, 2001). Ambas tecnologias são compatíveis com uma ampla variedade de sistemas operacionais e muito utilizadas por conta da sua simplicidade, economia e escalabilidade (SPOTNIK et al, 2023).

Figura 4 - PhpMyAdmin do Sistema Manualmente Desenvolvido



The screenshot displays the PHPMyAdmin interface for a database named 'ruydomanual'. The left sidebar shows a tree view of databases, including 'information\_schema', 'ruydomanual', 'loja\_virtual', 'mysql', 'performance\_schema', 'phpmyadmin', and 'test'. The main area shows the 'Estrutura' (Structure) tab for the 'ruydomanual' database. A table list is displayed with columns: Tabela, Ação, Linhas, Tipo, Colaço, Tamanho, and Sobrecarga. The tables listed are: categories (4 lines, 16.0 KB), products (4 lines, 32.0 KB), purchase\_items (0 lines, 48.0 KB), users (5 lines, 64.0 KB), and user\_purchases (3 lines, 32.0 KB). A summary row shows 5 tables with a total of 16 lines and 192.0 KB. Below the table list, there is a 'Criar nova tabela' (Create new table) form with fields for 'Nome da Tabela' and 'Número de colunas' (set to 4).

Tabela	Ação	Linhas	Tipo	Colaço	Tamanho	Sobrecarga
categories	Visualizar, Estrutura, Procurar, Inserir, Limpar, Eliminar	4	InnoDB	utf8_general_ci	16.0 KB	-
products	Visualizar, Estrutura, Procurar, Inserir, Limpar, Eliminar	4	InnoDB	utf8_general_ci	32.0 KB	-
purchase_items	Visualizar, Estrutura, Procurar, Inserir, Limpar, Eliminar	0	InnoDB	utf8_general_ci	48.0 KB	-
users	Visualizar, Estrutura, Procurar, Inserir, Limpar, Eliminar	5	InnoDB	utf8_general_ci	64.0 KB	-
user_purchases	Visualizar, Estrutura, Procurar, Inserir, Limpar, Eliminar	3	InnoDB	utf8_general_ci	32.0 KB	-
<b>5 tabelas</b>	<b>Soma</b>	<b>16</b>	<b>InnoDB</b>	<b>utf8_general_ci</b>	<b>192.0 KB</b>	<b>0 Bytes</b>

Fonte: Autor, 2024.

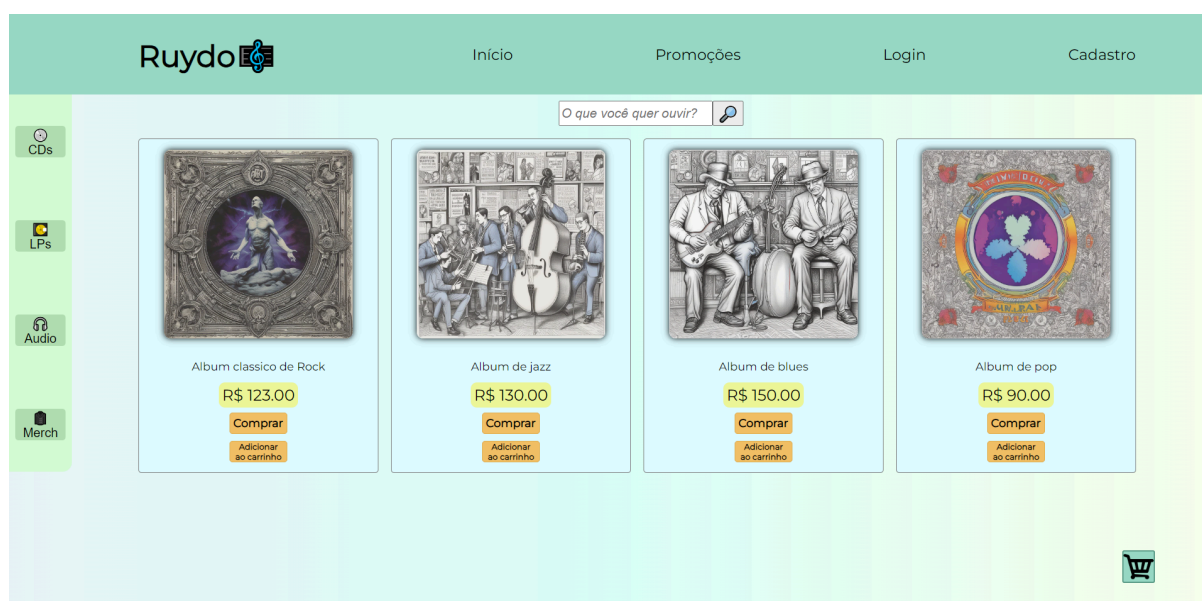
<sup>8</sup> [https://www.apachefriends.org/pt\\_br/index.html](https://www.apachefriends.org/pt_br/index.html)

<sup>9</sup> <https://www.phpmyadmin.net/>

<sup>10</sup> <https://www.php.net/>

O PHPMyAdmin, uma ferramenta baseada na web, facilitou a administração do MySQL, permitindo a criação, modificação e visualização de tabelas e dados. Para a construção das páginas web, HTML<sup>11</sup> e CSS<sup>12</sup> foram utilizados, chegando no resultado mostrado na Figura 5. O HTML estruturou o conteúdo das páginas, enquanto o CSS estilizou e organizou a apresentação visual, criando uma interface agradável e fácil de navegar. Além disso, JavaScript, em conjunto com a técnica AJAX<sup>13</sup>, foi empregado para permitir a atualização assíncrona dos conteúdos das páginas, melhorando a interatividade e a responsividade da aplicação (DISSANAYAKE e DIAS, 2021).

Figura 5 - Tela Inicial da Loja Ruydo Desenvolvida Manualmente.



Fonte: Autor, 2024.

A comunicação entre o front-end e o back-end foi gerida pelo JavaScript, que se comunicava com os scripts PHP para operações no banco de dados. Para a codificação do projeto, foi utilizado o editor de texto VSCode<sup>14</sup>, devido à sua simplicidade e funcionalidade intuitiva, suportando múltiplas linguagens de programação e facilitando a edição de códigos. Ao finalizar o desenvolvimento manual do sistema, foi iniciado o desenvolvimento por meio do ChatGPT.

### 3.2 DESENVOLVIMENTO COM CHATGPT

<sup>11</sup> <https://developer.mozilla.org/pt-BR/docs/Web/HTML>

<sup>12</sup> <https://developer.mozilla.org/pt-BR/docs/Web/CSS>

<sup>13</sup> [https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/Client-side\\_web\\_APIs/Fetching\\_data](https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/Client-side_web_APIs/Fetching_data)

<sup>14</sup> <https://code.visualstudio.com/>

WHITE, 2023, define a engenharia de prompt como uma habilidade cada vez mais importante para se comunicar de maneira eficaz com modelos de linguagem grandes (LLMs), como o ChatGPT. Essa prática envolve a formulação cuidadosa de perguntas e comandos para obter respostas mais precisas e relevantes dos LLMs. Com a capacidade desses modelos de interpretar e responder a uma ampla gama de solicitações, a maneira como as instruções são apresentadas pode influenciar significativamente a qualidade das respostas geradas. Portanto, dominar a engenharia de prompt é essencial para maximizar o potencial dessas tecnologias avançadas, seja em aplicações comerciais, acadêmicas ou pessoais.

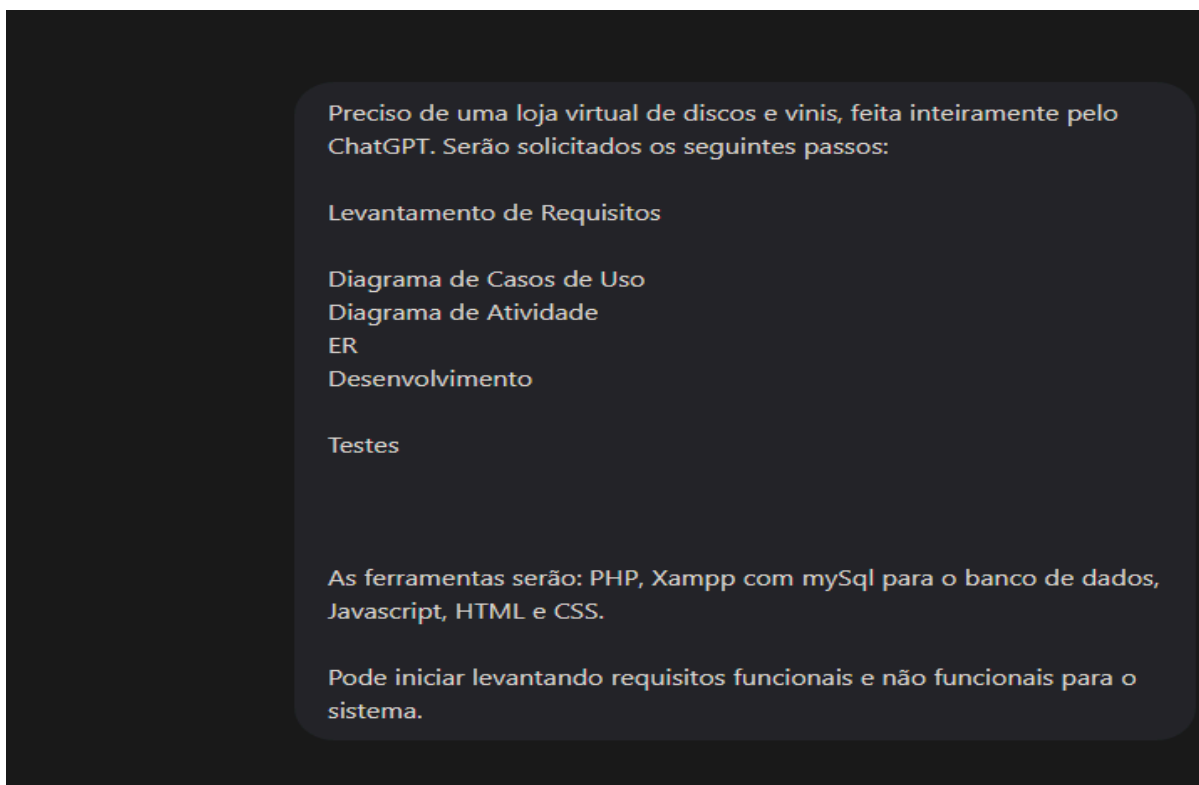
Nesta seção, serão documentadas as principais requisições e respostas do ChatGPT durante todo o desenvolvimento do sistema (para acessar as demais interações, ver apêndice C).

### **3.2.1 DIAGRAMAS GERADOS**

O primeiro passo do desenvolvimento via ChatGPT foi estabelecer uma requisição clara e objetiva, exigindo os seguintes passos para a ferramenta: levantamento de requisitos, diagrama de casos de uso, diagrama de atividades, diagrama entidade-relacionamento, desenvolvimento e testes. Nessa mesma requisição, foi solicitado que gerasse o levantamento de requisitos.

Na Figura 6, encontra-se a primeira interação com a ferramenta:

Figura 6 - Primeira interação com o ChatGPT



Fonte: Elaborado pelo Autor, 2024

*“Preciso de uma loja virtual de discos e vinis, feita inteiramente pelo ChatGPT. Serão solicitados os seguintes passos: Levantamento de Requisitos, Diagrama de Casos de Uso, Diagrama de Atividade, ER, Desenvolvimento, Testes As ferramentas serão: PHP, Xampp com mySql para o banco de dados, Javascript, HTML e CSS. Pode iniciar levantando requisitos funcionais e não funcionais para o sistema.”*

Então, a ferramenta apresentou os requisitos funcionais e não funcionais do sistema gerado pelo ChatGPT, organizados em duas colunas distintas, representados na Tabela 2. Os requisitos funcionais incluem cadastro de usuário, catálogo de produtos, carrinho de compras, checkout, gestão de pedidos e administração do sistema, enquanto os não funcionais abrangem desempenho, segurança, usabilidade, confiabilidade e compatibilidade.



Tabela 2 - Requisitos Funcionais e Não Funcionais Gerados pelo ChatGPT

<b>ID</b>	<b>Requisito</b>	<b>Descrição</b>
RF01	Cadastro de Usuário	Permitir que os usuários se registrem fornecendo nome, endereço, e-mail e senha.
RF02	Login de Usuário	Permitir que usuários cadastrados façam login no sistema.
RF03	Catálogo de Produtos	Exibir uma lista de discos e vinis disponíveis para compra, com filtros por categoria, artista, etc.
RF04	Detalhes do Produto	Mostrar informações detalhadas dos produtos, incluindo imagens, descrições, preço e disponibilidade.
RF05	Carrinho de Compras	Permitir que os usuários adicionem itens ao carrinho, editem a quantidade e vejam o subtotal e total do pedido.
RF06	Checkout	Oferecer um processo de checkout que solicita informações de pagamento e envio, e calcula o total da compra incluindo impostos e custos de envio.
RF07	Gestão de Pedidos	Permitir que os usuários visualizem o histórico de pedidos, enviem e-mails de confirmação e rastreiem o status dos pedidos.
RF08	Administração do Sistema	Permitir que administradores adicionem, editem e removam produtos, gerenciem pedidos e visualizem relatórios de vendas e estatísticas.
RNF01	Desempenho	O sistema deve ser responsivo e oferecer tempos de carregamento rápidos, mesmo durante períodos de tráfego intenso.
RNF02	Segurança	Implementar medidas de segurança robustas, como criptografia de senha e comunicação segura, e garantir conformidade com regulamentações de proteção de dados.
RNF03	Usabilidade	O sistema deve ser intuitivo e fácil de usar para usuários de todos os níveis de habilidade, oferecendo uma experiência consistente em diferentes dispositivos e navegadores.
RNF04	Confiabilidade	Minimizar o tempo de inatividade do sistema por meio de testes rigorosos e monitoramento contínuo, e implementar backups regulares dos dados do sistema.
RNF05	Compatibilidade	Garantir que o sistema seja compatível com diferentes navegadores da web e dispositivos, e com as versões mais recentes das tecnologias web.

Fonte: Elaborado pelo Autor, 2024.

Com os requisitos funcionais e não-funcionais estabelecidos, foi requisitado à ferramenta que gerasse o Diagrama de Casos de Uso no formato PlantUML<sup>15</sup>, que fornece diversos modelos de diagramas usados na engenharia de software por meio de um código de texto simples, utilizando o prompt “Continue com o Diagrama de Casos de Uso, usando a ferramenta PlantUML”. Então, o seguinte código foi gerado:

```
@startuml
  left to right direction
  actor "Usuário" as User
  actor "Administrador" as Admin

  rectangle "Loja Virtual de Discos e Vinis" {
    User --> (Visualizar Catálogo)
    User --> (Pesquisar Produtos)
    User --> (Visualizar Detalhes do Produto)
    User --> (Adicionar Item ao Carrinho)
    User --> (Remover Item do Carrinho)
    User --> (Finalizar Compra)

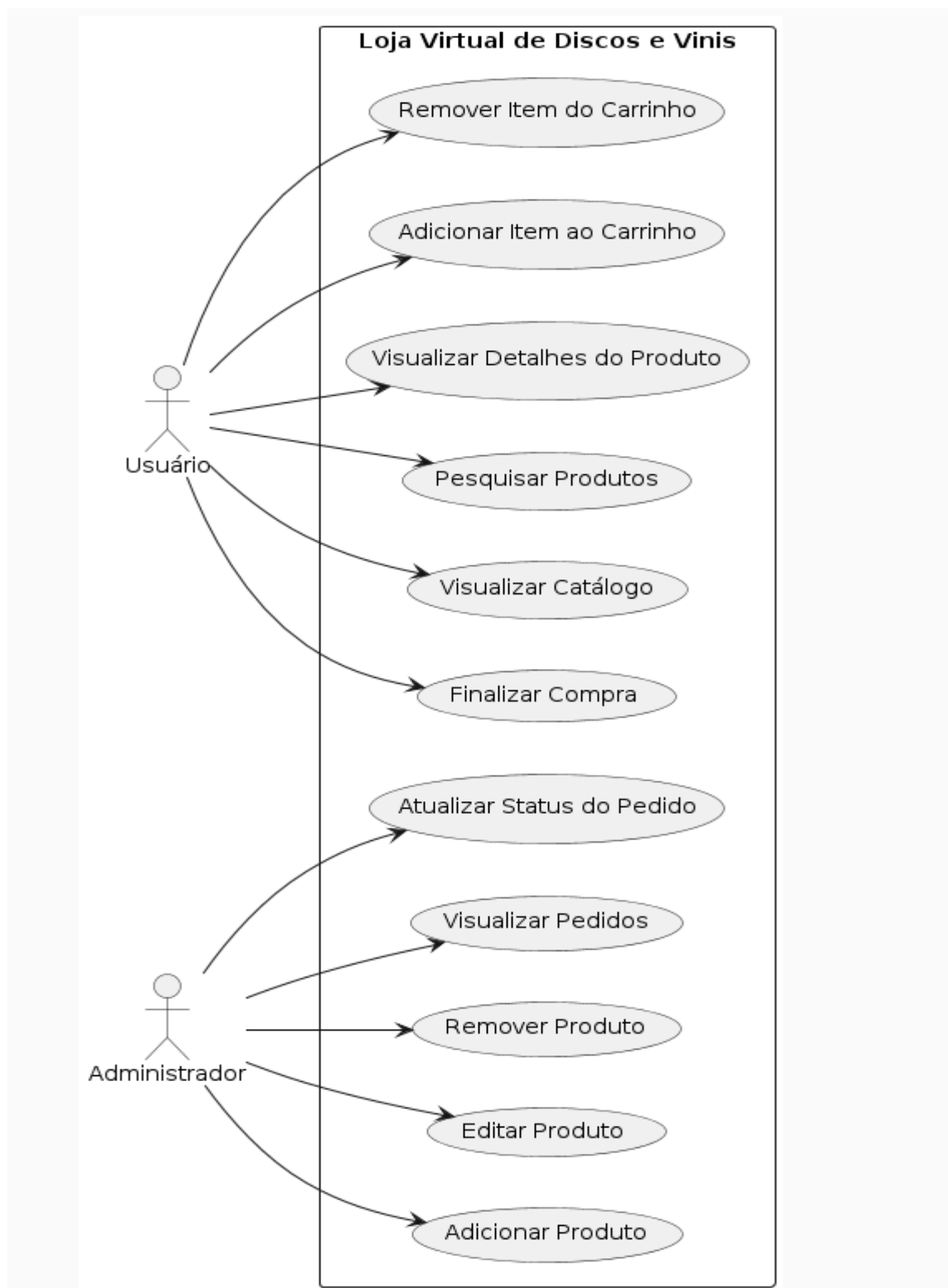
    Admin --> (Adicionar Produto)
    Admin --> (Editar Produto)
    Admin --> (Remover Produto)
    Admin --> (Visualizar Pedidos)
    Admin --> (Atualizar Status do Pedido)
  }
@enduml
```

Com esse código executado no PlantUML, foi gerado o diagrama de casos de uso que vemos na Figura 7.

---

<sup>15</sup> <https://plantuml.com/>

Figura 7 - Diagrama de Casos de Uso Gerado pelo ChatGPT



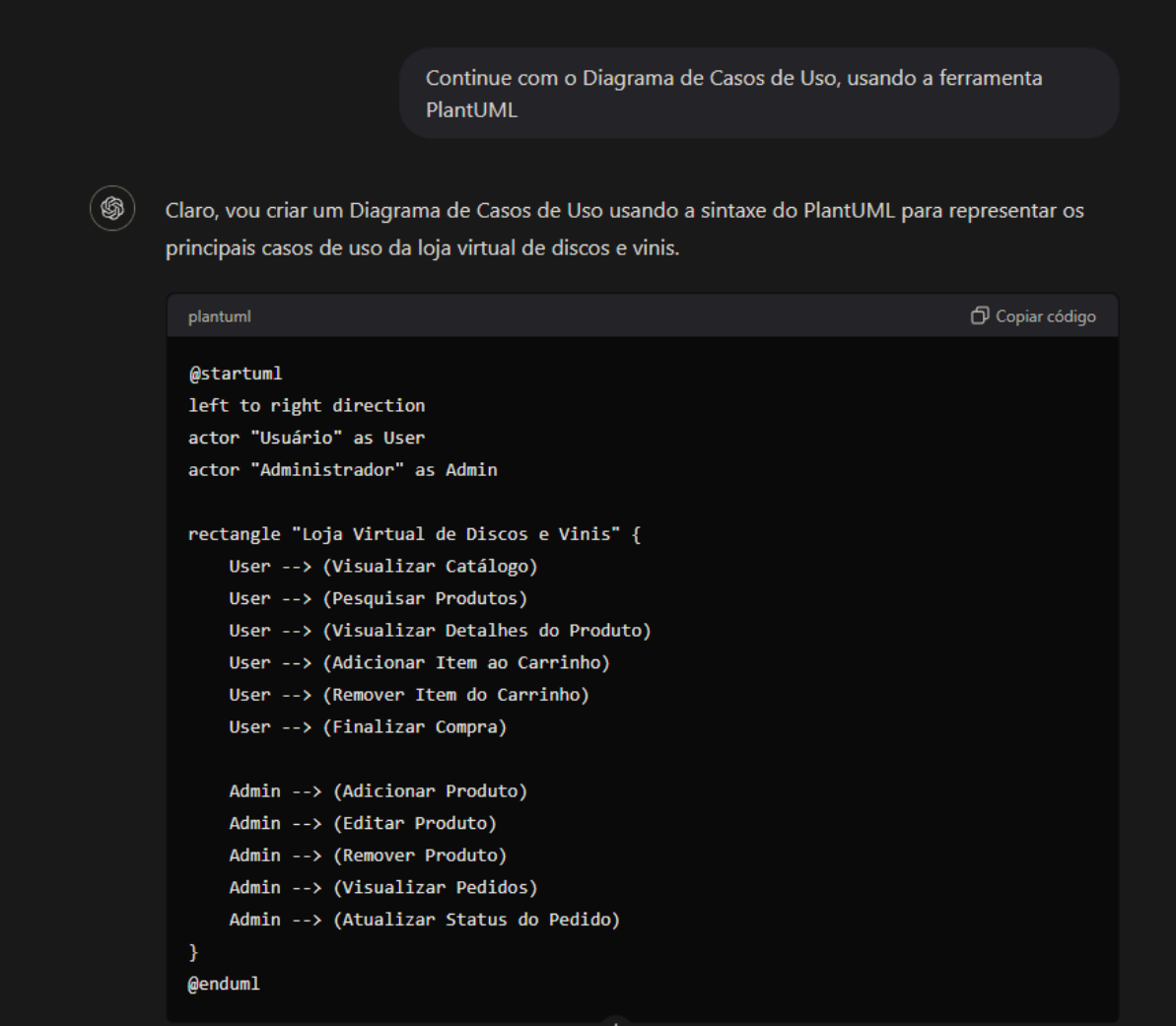
Fonte: Elaborado pelo autor, 2024.

O diagrama mostra as principais interações do usuário com o sistema, incluindo a adição e remoção de itens do carrinho, visualização de detalhes do produto, pesquisa de produtos, visualização do catálogo, finalização de compras e atualização do status dos pedidos. Diferente do diagrama feito manualmente, o

gerado pelo ChatGPT não conta com ações dependentes umas das outras, além de separar diretamente as atividades feitas pelos usuários e pelos administradores.

Na Figura 8, tem-se a próxima requisição, o diagrama de atividades, utilizando o prompt “Excelente, agora prossiga com o diagrama de atividades.”, onde se obteve por resposta um novo código PlantUML.

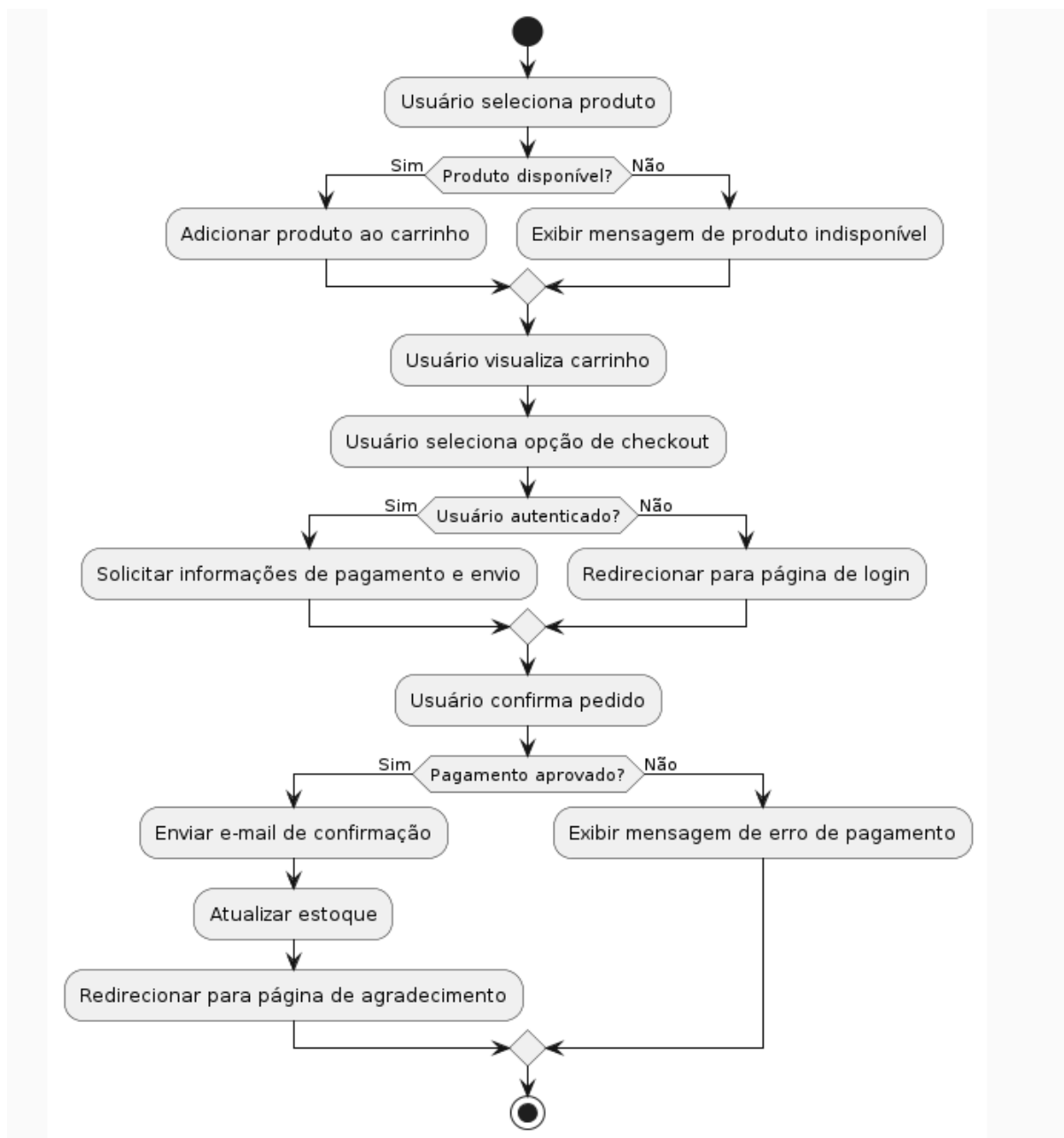
Figura 8 - Requisição do Diagrama de Atividades

The image shows a chat interface with a dark background. At the top, a light grey bubble contains the text: "Continue com o Diagrama de Casos de Uso, usando a ferramenta PlantUML". Below this, a circular icon with a stylized 'G' is followed by the text: "Claro, vou criar um Diagrama de Casos de Uso usando a sintaxe do PlantUML para representar os principais casos de uso da loja virtual de discos e vinis." Underneath the text is a code editor window with a dark background and light text. The code is PlantUML syntax for a Use Case Diagram. It starts with "@startuml", followed by "left to right direction", and two actor definitions: "actor 'Usuário' as User" and "actor 'Administrador' as Admin". A large rectangle represents the system boundary: "rectangle 'Loja Virtual de Discos e Vinis' {". Inside this rectangle, there are two groups of use cases. The first group is for the User: "User --> (Visualizar Catálogo)", "User --> (Pesquisar Produtos)", "User --> (Visualizar Detalhes do Produto)", "User --> (Adicionar Item ao Carrinho)", "User --> (Remover Item do Carrinho)", and "User --> (Finalizar Compra)". The second group is for the Admin: "Admin --> (Adicionar Produto)", "Admin --> (Editar Produto)", "Admin --> (Remover Produto)", "Admin --> (Visualizar Pedidos)", and "Admin --> (Atualizar Status do Pedido)". The code ends with "}" and "@enduml". A "Copiar código" button is visible in the top right corner of the code editor.

Fonte: Elaborado pelo Autor, 2024.

Executando esse código, foi gerado um fluxograma (ver figura 9), que possui correspondentes nos diagramas de atividades (MELLO, 2011).

Figura 9 - Diagrama de Atividades Gerado pelo ChatGPT.



Fonte:Elaborado pelo autor, 2024.

Este diagrama de atividades descreve as principais etapas do uso do sistema, desde a seleção de um produto pelo usuário, até a confirmação do pedido, processamento do pagamento e atualização do estoque. O diagrama mostra as diferentes decisões e ações do usuário ao longo do processo, como adicionar o produto ao carrinho, selecionar a opção de checkout, fornecer informações de pagamento, receber confirmação da compra, entre outras.

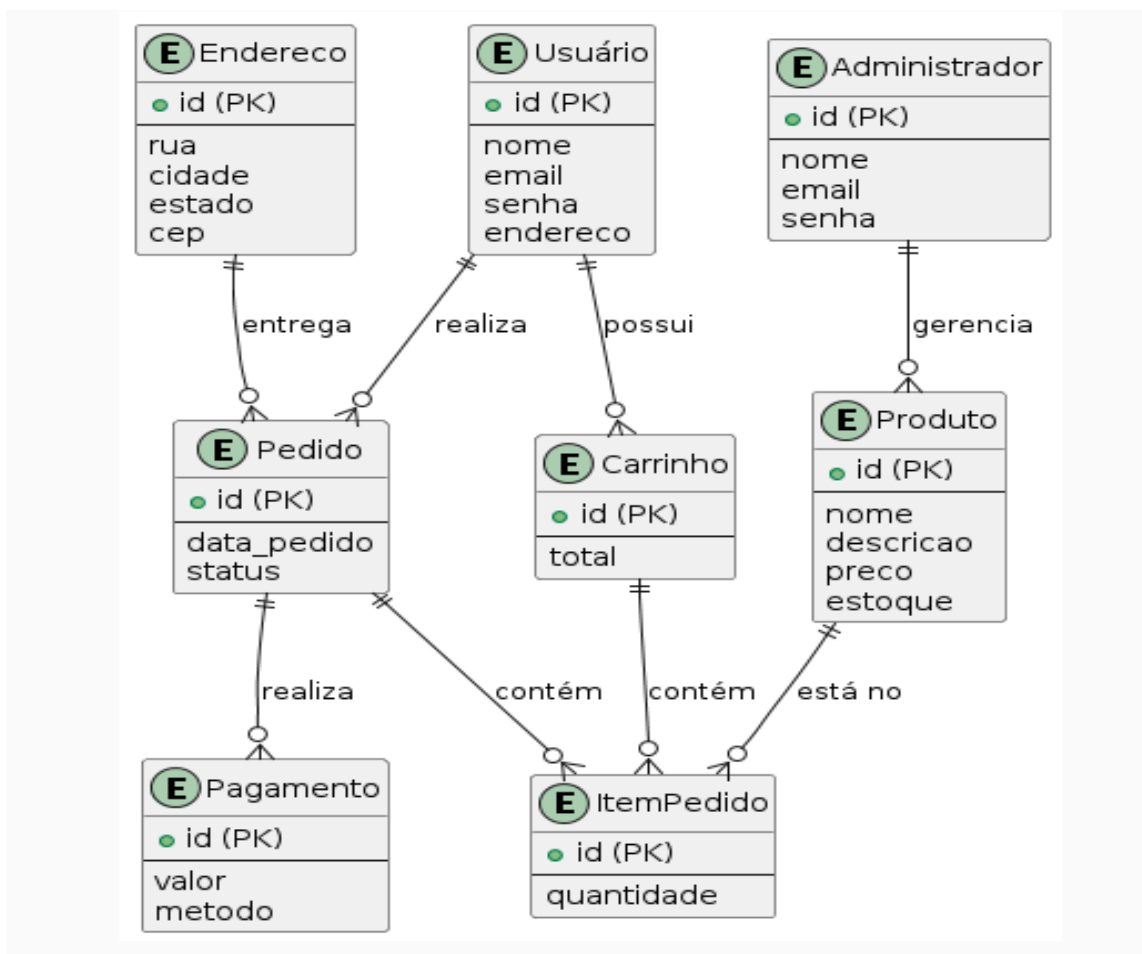
Por fim, foi requisitado o diagrama do modelo Entidade-Relacionamento, utilizando o prompt “Certo, agora prossiga com o diagrama Entidade Relacionamento, para o banco de dados.”, que retornou outro código PlantUML:

```
"@startuml  
entity "Usuário" {  
  + id (PK)  
  --  
  nome  
  email  
  senha  
  endereco  
}
```

```
entity "Produto" {  
  + id (PK)  
  --  
  nome  
  descricao  
  preco  
  estoque  
}  
[...]"
```

O código, então, gerou o diagrama entidade-relacionamento (ver Figura 10). Esse diagrama organiza a estrutura do banco de dados, com entidades como Usuário, Produto, Pedido e Pagamento, e seus respectivos atributos e relacionamentos, permitindo a visualização das interações do sistema.

Figura 10 - Diagrama Entidade-Relacionamento Gerado pelo ChatGPT



Fonte: Elaborado pelo Autor, 2024.

As entidades principais no diagrama incluem "Endereço" (com atributos como rua, cidade, estado, e cep), "Usuário" (nome, email, senha, e endereço), "Administrador" (nome, email, senha), "Pedido" (data\_pedido, status), "Pagamento" (valor, método), "Carrinho" (total), "Produto" (nome, descrição, preço, estoque), e "ItemPedido" (quantidade). Os relacionamentos indicam que um "Endereço" entrega um "Pedido", um "Usuário" realiza um "Pedido" e possui um "Carrinho", um "Administrador" gerencia "Produto", um "Pedido" realiza um "Pagamento" e contém "ItemPedido", e um "Carrinho" contém "ItemPedido", que está relacionado a um "Produto".

Esses diagramas serviram para a ferramenta se guiar ao implementar o desenvolvimento da aplicação.

### 3.2.2 IMPLEMENTAÇÃO DO SISTEMA GERADO PELO CHATGPT

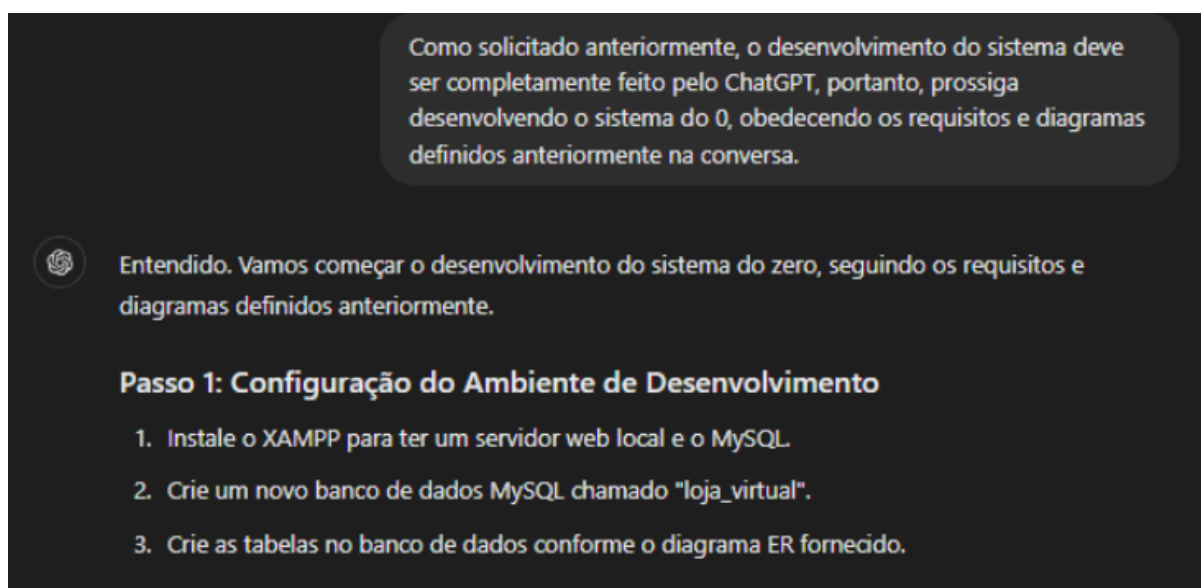
Nesta fase, foi iniciado o processo de solicitações de código ao ChatGPT. Utilizando a engenharia de prompt, foram formuladas perguntas e comandos

específicos para orientar o modelo na geração do código necessário para o desenvolvimento do sistema de loja virtual "Ruydo". Para manter a consistência, foi utilizado o mesmo ambiente de desenvolvimento que no desenvolvimento manual: o XAMPP, com PHP, MySQL e PhpMyAdmin, HTML, CSS e JavaScript. Esta escolha garantiu que as diferenças observadas fossem atribuídas principalmente à fonte do código (manual vs. ChatGPT) e não às ferramentas utilizadas.

Ao solicitar a ferramenta que desse início ao desenvolvimento, com o prompt "Certo, agora prossiga com o desenvolvimento do sistema.", teve-se o primeiro erro por parte do ChatGPT. Mesmo especificando anteriormente para a ferramenta que a aplicação deveria ser inteiramente desenvolvida pela própria, na primeira tentativa de dar sequência ao desenvolvimento ao sistema, o retorno não foi como esperado, na verdade, foi provido informações e exemplos de como o próprio usuário deveria seguir na produção da loja: "[...] Use PHP para processar os dados do formulário e inserir as informações do usuário no banco de dados MySQL. Consulte o banco de dados MySQL para obter os produtos disponíveis e exibi-los em uma página HTML.[...]"

Enganos como esse tornaram-se comuns ao longo da pesquisa, sendo necessária a correção em diversos momentos para que a resposta fosse mais adequada ao que era pedido. Na Figura 11, tem-se um exemplo de correção requisitada à ferramenta.

Figura 11 - Primeira Correção Solicitada a Ferramenta



Fonte: Elaborado pelo Autor, 2024.



Depois da correção, a ferramenta entendeu o que era requisitado e começou por enviar respostas mais adequadas, contendo código de fato e instruções de como aplicá-lo no desenvolvimento.

Após criar as principais páginas do sistema, foi solicitado o sistema de banco de dados, passando o código PlantUML do Diagrama Entidade-Relacionamento enviado anteriormente como contexto: “Envie o SQL responsável pelo diagrama Entidade Relacionamento enviado anteriormente:

```
@startuml
entity "Usuário" {
+ id (PK)
--
nome
email
senha
endereco
}
```

[...]”. A resposta foi, então, executada no sistema do PHPMyAdmin, onde ele permite inserir códigos completos de importação e alteração de bancos de dados via SQL, como mostrado na Figura 12.

Figura 12 - Resultado da Execução do Primeiro Código SQL Providenciado pelo ChatGPT.

The screenshot shows the PHPMyAdmin interface. The left pane contains the following SQL code:

```
1 CREATE TABLE Usuario (
2   id INT AUTO_INCREMENT PRIMARY KEY,
3   nome VARCHAR(100) NOT NULL,
4   email VARCHAR(100) NOT NULL,
5   senha VARCHAR(100) NOT NULL,
6   endereco VARCHAR(255) NOT NULL
7 );
8
9 CREATE TABLE Produto (
10  id INT AUTO_INCREMENT PRIMARY KEY,
11  nome VARCHAR(100) NOT NULL,
12  descricao TEXT,
13  preco DECIMAL(10, 2) NOT NULL,
14  estoque INT NOT NULL
15 );
16
17 CREATE TABLE Pedido (
18  id INT AUTO_INCREMENT PRIMARY KEY,
19  data_pedido DATE,
20  status VARCHAR(50)
21 );
22
23 CREATE TABLE ItemPedido (
24  id INT AUTO_INCREMENT PRIMARY KEY,
25  quantidade INT NOT NULL,
26  pedido_id INT,
27  produto_id INT,
28  FOREIGN KEY (pedido_id) REFERENCES Pedido(id),
29  FOREIGN KEY (produto_id) REFERENCES Produto(id)
30 );
31
32 CREATE TABLE Carrinho (
33  id INT AUTO_INCREMENT PRIMARY KEY,
34  total DECIMAL(10, 2) NOT NULL,
35  usuario_id INT,
36  FOREIGN KEY (usuario_id) REFERENCES Usuario(id)
37 );
```

The right pane shows the 'Estrutura' view of the database, listing the created tables with their columns and sizes:

Tabela	Ação	Linhas	Tipo	Colação	Tamanho	Sobrecarga
administrador	Visualizar	4	InnoDB	utf8mb4_general_ci	16.0 KB	-
carinho	Visualizar	4	InnoDB	utf8mb4_general_ci	32.0 KB	-
endereco	Visualizar	4	InnoDB	utf8mb4_general_ci	16.0 KB	-
itempedido	Visualizar	4	InnoDB	utf8mb4_general_ci	48.0 KB	-
pagamento	Visualizar	4	InnoDB	utf8mb4_general_ci	16.0 KB	-
pedido	Visualizar	4	InnoDB	utf8mb4_general_ci	16.0 KB	-
produto	Visualizar	4	InnoDB	utf8mb4_general_ci	16.0 KB	-
usuario	Visualizar	4	InnoDB	utf8mb4_general_ci	16.0 KB	-
<b>8 tabelas</b>	<b>Soma</b>				<b>176.0 KB</b>	<b>0 Bytes</b>

Fonte: Elaborado pelo Autor, 2024.

Após implementar todo o material fornecido até então e conectar ao servidor do XAMPP para analisar o resultado no navegador, um erro SQL era mostrado. Nesse ponto, após solicitar sua correção enviando juntamente todo o código como contexto para auxiliá-la utilizando o prompt

“[...] Esse código fornecido pela ferramenta gerou o seguinte erro:

```
Fatal error: Uncaught mysqli_sql_exception: Unknown database 'loja_virtual' in
C:\xampp\htdocs\repoGPT\harmoniaShop\index.php:21 Stack trace: #0
C:\xampp\htdocs\repoGPT\harmoniaShop\index.php(21):
mysqli->__construct('localhost', 'root', Object(SensitiveParameterValue),
'loja_virtual') #1 {main} thrown in
C:\xampp\htdocs\repoGPT\harmoniaShop\index.php on line 21.”
```

Na resposta, teve-se o primeiro caso de contradição da IA, também chamada de “alucinação”, que, para alguns autores, é quando a ferramenta gera informações imprecisas ou até mesmo falsas (CATALANO e LORENZI, 2023), representado na Figura 13:

Figura 13 - Primeiro Caso de Alucinação do ChatGPT

Esse código fornecido pela ferramenta gerou esse erro aqui

```
Fatal error: Uncaught mysqli_sql_exception: Unknown database
'loja_virtual' in
C:\xampp\htdocs\repoGPT\harmoniaShop\index.php:21 Stack trace:
#0 C:\xampp\htdocs\repoGPT\harmoniaShop\index.php(21): mysqli-
> __construct('localhost', 'root', Object(SensitiveParameterValue),
'loja_virtual') #1 {main} thrown in
C:\xampp\htdocs\repoGPT\harmoniaShop\index.php on line 21
```

Analise novamente a sintaxe do SQL fornecido anteriormente e do código que provido, está tudo de acordo?

Desculpe pelo erro anterior, parece que houve uma inconsistência entre o nome do banco de dados usado no código PHP e o nome especificado no script SQL. Vou corrigir isso:

No código PHP fornecido, o nome do banco de dados usado é "loja\_virtual". No entanto, no script SQL fornecido, o banco de dados é criado com o nome "loja\_virtual". Vamos ajustar o nome do banco de dados no script SQL para corresponder ao nome usado no código PHP.

Aqui está o script SQL corrigido:

```
sql Copiar código

CREATE DATABASE loja_virtual;

USE loja_virtual;

CREATE TABLE Usuario (
  id INT AUTO_INCREMENT PRIMARY KEY,
  nome VARCHAR(100) NOT NULL,
  email VARCHAR(100) NOT NULL,
  senha VARCHAR(100) NOT NULL,
  endereco VARCHAR(255) NOT NULL
);

CREATE TABLE Produto (
  id INT AUTO_INCREMENT PRIMARY KEY,
  nome VARCHAR(100) NOT NULL,
  descricao TEXT,
  preco DECIMAL(10, 2) NOT NULL,
  estoque INT NOT NULL
);

CREATE TABLE Pedido (
  id INT AUTO_INCREMENT PRIMARY KEY,
```

Parece que o código está tentando se conectar a um banco de dados chamado "loja\_virtual", mas o erro indica que esse banco de dados não foi encontrado. Isso pode acontecer se o banco de dados não estiver configurado corretamente no servidor MySQL ou se o nome do banco de dados estiver incorreto no código.

Vamos resolver isso:

1. Verifique se você criou o banco de dados "loja\_virtual" no MySQL. Você pode fazer isso acessando o phpMyAdmin ou usando um cliente MySQL como o MySQL Workbench.
2. Se você confirmou que o banco de dados está configurado corretamente, verifique se as credenciais de conexão (nome de usuário e senha) estão corretas no código.
3. Certifique-se de que o usuário do MySQL tenha permissões adequadas para acessar o banco de dados. O usuário precisa ter permissões para SELECT, INSERT, UPDATE e DELETE nas tabelas do banco de dados.
4. Se você alterou o nome do banco de dados ou as credenciais de conexão, atualize o código PHP com as informações corretas.

Se você seguir essas etapas e ainda tiver problemas, me avise para que possamos investigar mais a fundo.

Fonte: Elaborado pelo Autor, 2024.

*“[...] o nome do banco de dados usado é ‘loja\_virtual’. No entanto, no script SQL fornecido, o banco de dados é criado com o nome “loja\_virtual”[...]”*

Como indicado na Figura 13, ao tentar consertar o erro, a ferramenta afirma que o nome do banco de dados utilizado é “loja\_virtual”, e então tenta corrigir o que está sendo utilizado no script SQL, quando, na verdade, está de acordo. Então, após intervir e especificar mais a fonte do erro, especificando que era uma falha de sintaxe no código PHP, a ferramenta corrigiu o erro e seguiu seu trabalho.

Uma vez que as páginas estavam funcionais, foi iniciada a construção dos layouts por parte da ferramenta, com a requisição:

*“Agora crie CSS para as páginas”.*

Essa resultou em um resultado com códigos CSS como resposta:

*“Aqui está o código CSS básico para estilizar as páginas da loja virtual*

*Harmonia Shop:*

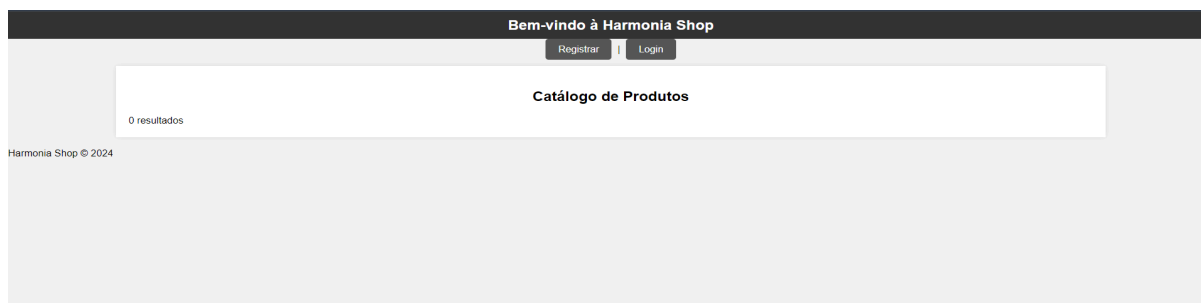
```
/* styles.css */
body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
    background-color: #f4f4f4;
}

header {
    background-color: #333;
    color: #fff;
    padding: 10px 0;
    text-align: center;
} [...]
```

Nesta fase, buscou-se criar algo relativamente parecido com o criado manualmente, sem passar argumentos diretos, para testar a capacidade da ferramenta de gerar uma interface limpa e que agradasse usuários a serem entrevistados uma vez que o sistema estivesse pronto.

Inicialmente, o ChatGPT gerou interfaces com problemas estéticos significativos, resultando em páginas visualmente pouco atraentes, com cores escuras e contrastes que tornavam a navegação desconfortável (ver Figura 14).

Figura 14 - Primeira Estética Gerada pelo ChatGPT.



Fonte: Elaborado pelo Autor, 2024.

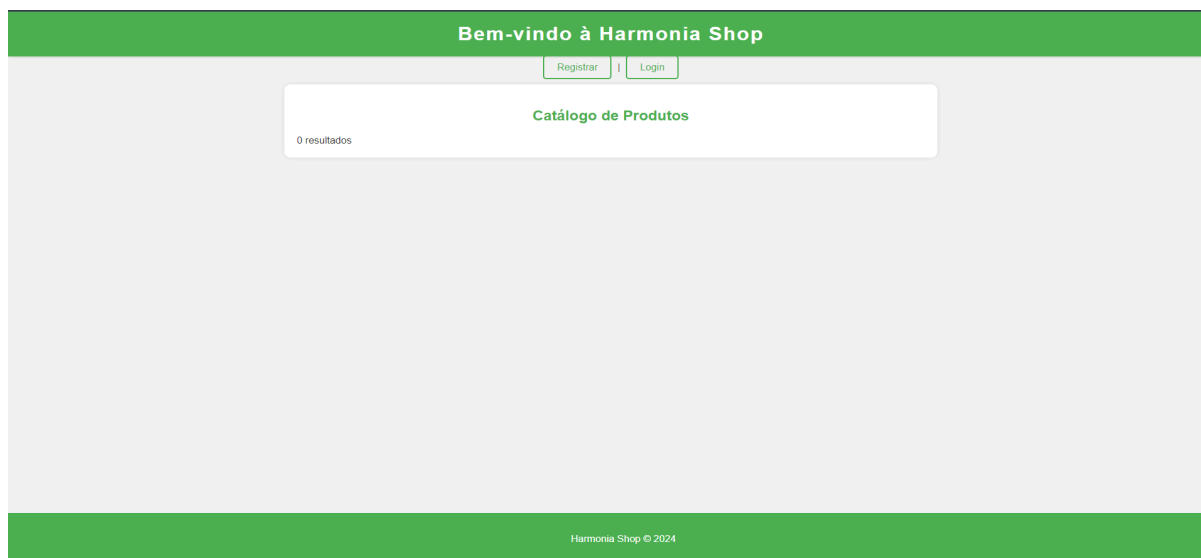
As cores escuras, combinadas com margens não otimizadas, criaram um ambiente pesado e pouco convidativo, evidenciando que a ferramenta, sem diretrizes específicas, tendia a gerar estilos que não atendiam aos padrões modernos de usabilidade e design centrado no usuário.

Reconhecendo esses problemas, novas tentativas foram feitas para tentar tornar a organização da interface mais clara e convidativa, como as seguintes requisições:

*“O CSS não está bom, deixe ele melhor em todas as páginas.”;*

*“O CSS não está mudando nada na página de registro, e alguns elementos estão desalinhados.”.* Essas tentativas resultaram em melhorias significativas (ver Figura 15). Contudo, o processo mostrou a necessidade de intervenção humana contínua para guiar a ferramenta em direção a um resultado esteticamente agradável e funcional.

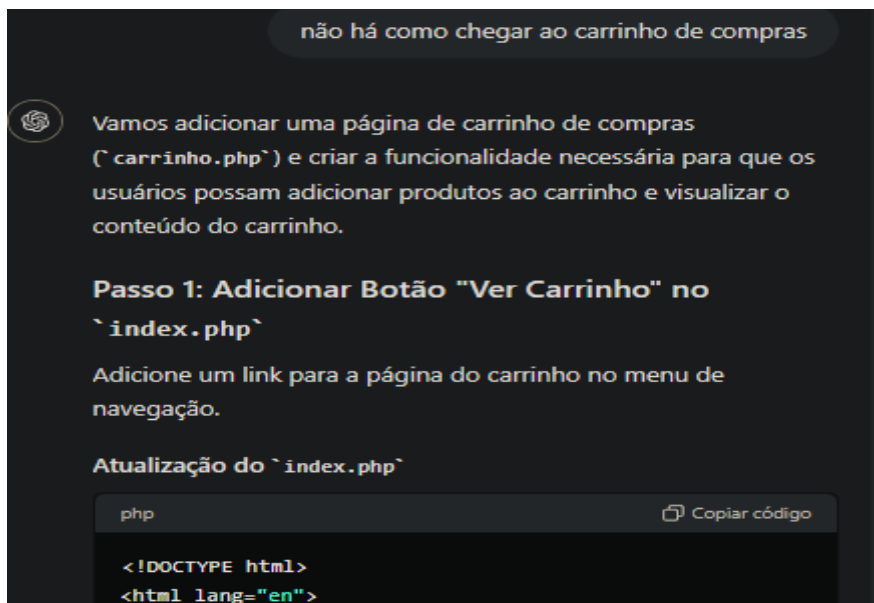
Figura 15 - Quarta Estética Gerada pelo ChatGPT



Fonte: Elaborado pelo Autor, 2024.

Além dos problemas estéticos, a interface gerada pelo ChatGPT apresentou falhas funcionais. Além disso, a interface permitia que os usuários tentassem fazer login repetidamente, mesmo após já estarem logados, em vez de oferecer uma opção de logout. Esses problemas de usabilidade demonstraram que, além da estética, a funcionalidade prática também precisava de ajustes significativos, que deveriam ser instruídos especificamente para a ferramenta, como representado na Figura 16:

Figura 16 - Correção de Falha Funcional do Sistema



Fonte: Elaborado pelo Autor, 2024.

Em resposta, a ferramenta retornava alterações em múltiplas partes do código se necessário, garantindo que o problema fosse resolvido, na maior parte das vezes, com apenas uma interação direta sendo necessária:

*"Passo 1: Adicionar Botão "Ver Carrinho" no index.php  
Adicione um link para a página do carrinho no menu de navegação.*

*Atualização do index.php*

*php*

*Copiar código*

*<!DOCTYPE html>*

*<html lang="en">*

*<head>*

*<meta charset="UTF-8">*

*<meta name="viewport" content="width=device-width, initial-scale=1.0">*

*<title>Harmonia Shop</title>*

*<link rel="stylesheet" href="styles.css">*

*</head>*

*<body>[...]*

## Passo 2: Criar Página de Carrinho (carrinho.php)

Crie uma página para visualizar os itens adicionados ao carrinho.

### Código de carrinho.php

php

Copiar código

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
  <title>Carrinho de Compras - Harmonia Shop</title>
```

```
[...]
```

Depois de consertar os erros funcionais da interface do sistema, foi requisitada a adição de produtos no catálogo, como representado na Figura 17.

Figura 17 - Requisição de Adição de Produtos no Catálogo

Agora preciso que tenha produtos no catálogo

🗨️

Para adicionar produtos ao catálogo, você pode inserir alguns registros na tabela `Produto` do seu banco de dados MySQL. Vou fornecer um script SQL para inserir produtos de exemplo no banco de dados.

### Passo 1: Inserir Produtos no Banco de Dados

Execute o seguinte script SQL no seu banco de dados MySQL para adicionar produtos de exemplo à tabela `Produto`:

sql
Copiar código

```

USE loja_virtual;

INSERT INTO Produto (nome, descricao, preco, estoque) VALUES
('Vinil A', 'Álbum clássico de rock', 120.00, 10),
('Vinil B', 'Álbum de jazz', 100.00, 8),
('Vinil C', 'Álbum de blues', 130.00, 5),
('Vinil D', 'Álbum de música clássica', 150.00, 7),
('Vinil E', 'Álbum de pop', 90.00, 12);
          
```

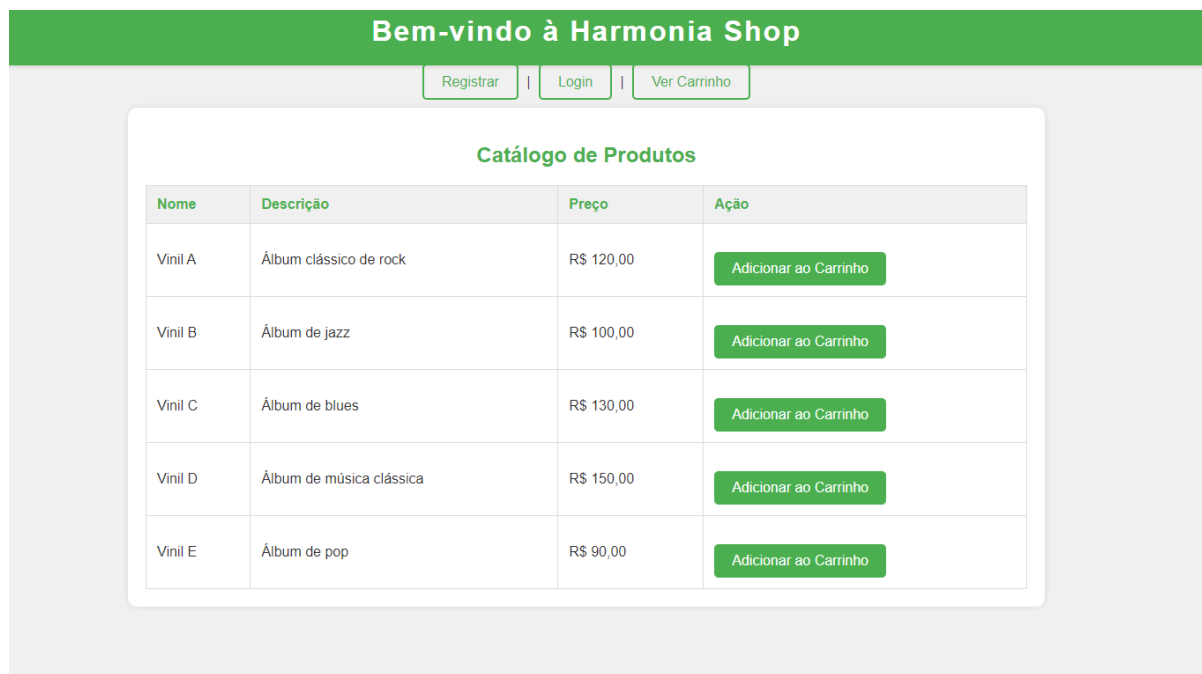
### Passo 2: Verificar o Catálogo de Produtos

Agora que os produtos foram adicionados ao banco de dados, você deve ser capaz de vê-los no catálogo de produtos ao acessar a página `index.php` da Harmonia Shop. O `index.php` foi configurado para consultar e exibir produtos da tabela `Produto`.

Fonte: Elaborado pelo Autor, 2024.

Similar ao processo de criação das tabelas do banco de dados, o código de inserção de dados na tabela de produtos é solicitado ao ChatGPT e, então, executado na plataforma do PHPMyAdmin. Após a execução, os produtos apareceram normalmente na tela principal, conforme a Figura 18.

Figura 17 - Adicionando Produtos Ao Catálogo via ChatGPT

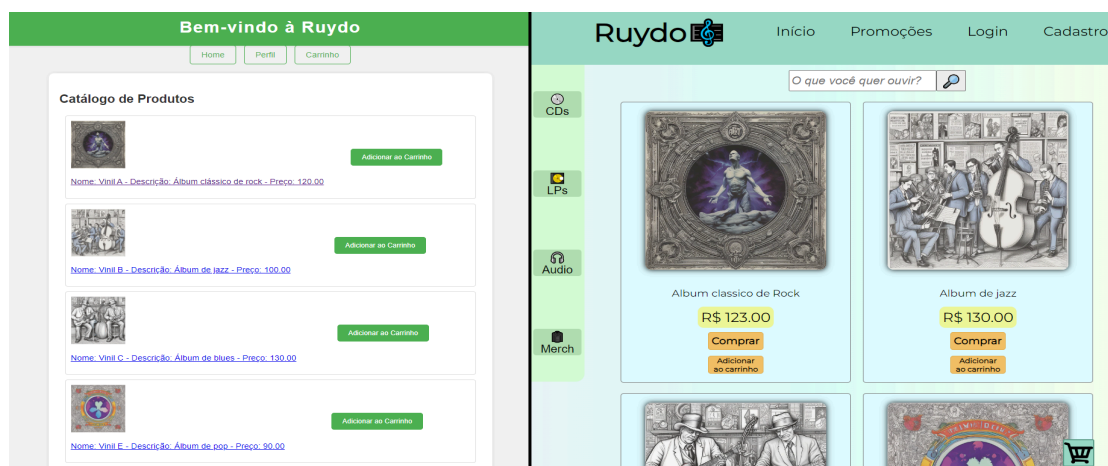


Fonte: Elaborado pelo Autor, 2024.

Após corrigir os demais erros do sistema, imagens foram adicionadas aos produtos do catálogo. Essas imagens foram produzidas via Artbreeder<sup>16</sup>, uma plataforma online que permite criar imagens através de requisições feitas com linguagem natural. Como representado na Figura 19, os produtos disponibilizados em ambas as plataformas usaram os mesmos nomes e imagens, para não influenciar usuários na identificação de qual sistema era feito com inteligência artificial ou não.

<sup>16</sup> <https://www.artbreeder.com/>

Figura 19 - Resultado Final das Telas Principais de Ambos os Sistemas



Fonte: Elaborado pelo Autor, 2024.

Uma vez que ambos os sistemas cumprissem os papéis de compra, cadastro, login e navegação, foram, então, submetidos a comparações técnicas e de usuários.



## 4 COMPARAÇÃO TÉCNICA

Nesta parte, ambos os sistemas foram submetidos à análise no SonarQube. Essa ferramenta ajuda a avaliar a qualidade do código, identificando possíveis problemas como bugs, vulnerabilidades e “code smells”, um termo comumente usado para descrever potenciais problemas no desenvolvimento do código (SANTOS et al., 2018). A análise no SonarQube fornece uma visão detalhada das áreas que necessitam de melhorias, permitindo uma comparação objetiva entre os dois sistemas em termos de qualidade e eficiência do código.

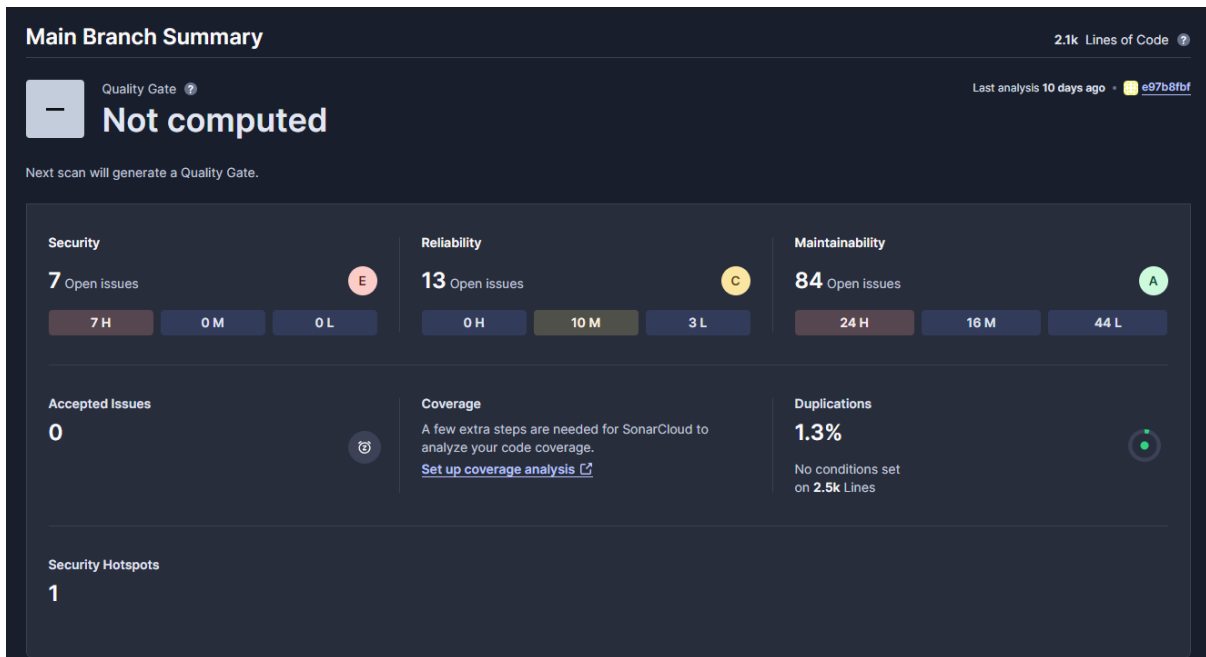
Por meio da ferramenta web SonarCloud.io<sup>17</sup>, que permite que os usuários adicionem seus repositórios do GitHub e demais plataformas no sistema de análise de código do SonarQube, foi possível reunir informações bem estruturadas sobre os erros e demais dados dos códigos desenvolvidos.

Como é possível ver nas Figuras 20 e 21, o código desenvolvido manualmente obteve um total de 7 erros altamente severos de segurança, 13 erros regularmente severos de confiabilidade e 84 erros altamente severos de manutenibilidade, totalizando 104 problemas identificados. Além disso, foi identificado uma redundância de 1.3% em aproximadamente 2500 linhas de código. O código desenvolvido via ChatGPT mostrou um total de 14 erros altamente severos de segurança, nenhum erro de confiabilidade, 1 altamente severo e 7 levemente severos de manutenibilidade, totalizando 22 erros. Já no aspecto de duplicações, foi observado um total de 13.6% de redundância num total de 865 linhas de código.

---

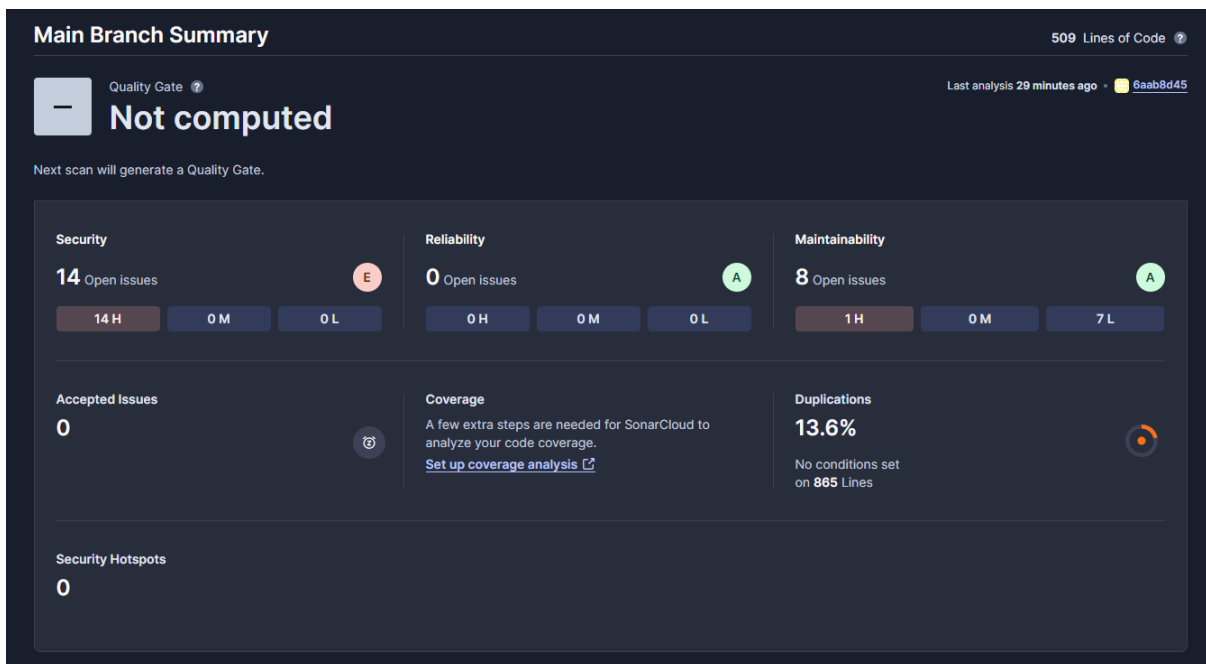
<sup>17</sup> <https://sonarcloud.io/>

Figura 20 - Análise Via SonarQube do Código Manualmente Desenvolvido



Fonte: Elaborado pelo Autor, 2024.

Figura 21 - Análise via SonarQube do Código Gerado pelo ChatGPT



Fonte: Elaborado pelo Autor, 2024.

Quando observamos a relação quantidade de linhas escritas x número de erros e repetições, é possível calcular a porcentagem de equívocos de ambas as aplicações, utilizando a seguinte fórmula:

$$\text{Erros por 1000 Linhas} = \left( \frac{\text{Total de Erros}}{\text{Total de Linhas de Código}} \right) \times 1000$$

Utilizando os dados mostrados anteriormente, chegamos nos seguintes cálculos:

- Código Manualmente Desenvolvido:

$$\text{Erros por 1000 Linhas} = \left( \frac{104}{2500} \right) \times 1000 = 41.6$$

- Código Gerado pelo ChatGPT

$$\text{Erros por 1000 Linhas} = \left( \frac{22}{865} \right) \times 1000 \approx 25.4$$

Desse modo, é possível observar os dados na seguinte tabela avaliativa:

Tabela 3 - Análise dos Dados Recolhidos via SonarQube

Critério	Código Desenvolvido Manualmente	Código Desenvolvido via ChatGPT
Total de Linhas de Código	2.500	865
Erros de Segurança	<b>7</b>	14
Erros de Confiabilidade	13	<b>0</b>
Erros de Manutenibilidade	84	<b>8</b>
Erros por 1000 Linhas de Código	41.6	<b>25.4</b>
Redundância por Linha	<b>0.013</b>	0.136

Fonte: Elaborado pelo Autor, 2024.

Esses cálculos ajudam a comparar a densidade de erros e redundância entre os dois sistemas, mostrando que, apesar de cometer menos erros diretos e gerar um código expressivamente menor do que o manualmente desenvolvido, o ChatGPT demonstra uma grande quantidade de repetições no código. Segundo JUERGENS et al., 2009, repetições inconsistentes são uma fonte significativa de falhas, indicando que a clonagem de código pode ser um problema substancial

durante o desenvolvimento e manutenção, a menos que se tome cuidado especial para identificar e acompanhar suas mudanças.

## 5 COMPARAÇÃO DOS USUÁRIOS

Neste capítulo, serão apresentados os resultados do questionário (ver Apêndice [D](#)) aplicado a diversos usuários, que realizaram as mesmas tarefas pré-estabelecidas em ambos os sites. Em seguida, será avaliada a experiência dos usuários utilizando uma adaptação do sistema de pesquisa System Usability Scale (SUS), criado por John Brooke em 1986 e amplamente utilizado no mundo inteiro (SAURO, 2011).

Neste projeto, foi utilizada uma métrica baseada no SUS, porém adaptada para a comparação de dois sistemas distintos. Diferente da metodologia tradicional da SUS, que avalia a usabilidade de um único sistema, essa abordagem permitiu uma comparação direta entre os sistemas avaliados. No método tradicional, as respostas definidas num modelo Likert, como vemos na figura 21, e então passam por uma fórmula para fornecer uma pontuação única entre 0 e 100 para cada sistema.

Figura 22 - Modelo Likert das Respostas do SUS

<b>Strongly Disagree 1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>Strongly Agree 5</b>
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Fonte: SAURO, 2024.

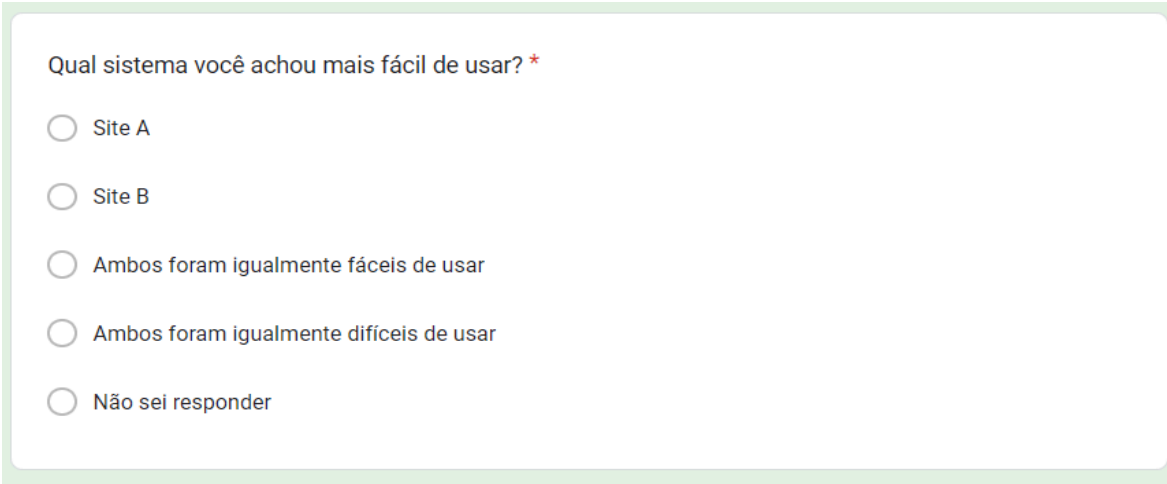
No entanto, a métrica adaptada simplifica os pontos coletados, permitindo uma comparação direta das pontuações dos dois sistemas. Esta adaptação é recomendada pois proporciona uma visão clara das diferenças relativas entre os sistemas, facilitando a identificação de qual deles oferece melhor usabilidade em termos comparativos.

Após a tentativa de execução das tarefas solicitadas, o usuário era perguntado se havia sucedido em ambos os sites. Se não conseguisse, um campo de texto era disponível ao usuário para explicar o motivo. Se conseguisse, 11 novas

perguntas eram disponibilizadas, separadas em perguntas positivas e negativas, para auxiliar na comparação dos sistemas.

Para calcular uma pontuação única para cada sistema, como na abordagem tradicional do SUS, a metodologia envolveu atribuir pontos com base nas respostas dos usuários para perguntas específicas. Para as perguntas consideradas positivas (ver Figura 23), um ponto era adicionado ao sistema escolhido pelo usuário, enquanto para as perguntas negativas (ver Figura 24), um ponto era subtraído do sistema selecionado. Se o usuário marcasse opções de que não sabiam responder ou avaliavam igualmente os dois sites, nenhum ponto era atribuído.

Figura 23 - Exemplo de Pergunta Positiva do Questionário de Comparação dos Sistemas

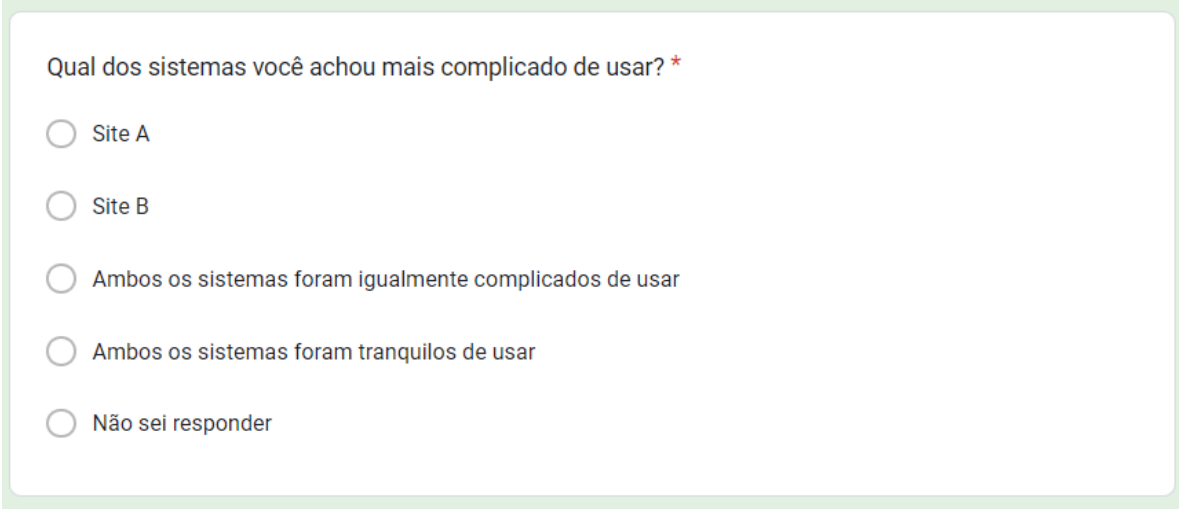


Qual sistema você achou mais fácil de usar? \*

- Site A
- Site B
- Ambos foram igualmente fáceis de usar
- Ambos foram igualmente difíceis de usar
- Não sei responder

Fonte: Elaborado pelo Autor, 2024.

Figura 24 - Exemplo de Pergunta Negativa do Questionário de Comparação dos Sistemas



Qual dos sistemas você achou mais complicado de usar? \*

- Site A
- Site B
- Ambos os sistemas foram igualmente complicados de usar
- Ambos os sistemas foram tranquilos de usar
- Não sei responder

Fonte: Elaborado pelo Autor, 2024.

O método aplicado gerou um total de 2 pontos para o sistema gerado pelo ChatGPT e 51 pontos para o desenvolvimento manualmente, mostrando não só uma preferência pelo de autoria humana, mas sim um descontentamento da vasta maioria dos usuários com o sistema gerado pelo ChatGPT. Por outro lado, 25% dos entrevistados não conseguiram efetuar as tarefas no site manualmente desenvolvido. Além disso, curiosamente, apenas um dos 16 usuários que responderam à pesquisa soube afirmar qual dos sites tinha sido feito com ajuda da IA, diferente do restante que não soube responder.

Tabela 4 - Pontuação do Questionário de Comparação dos Sistemas

Pergunta	Sis. manualmente desenvolvido	Sis. Desenvolvido via ChatGPT
Você achou algum sistema desnecessariamente complexo?	0	<b>1</b>
Qual sistema você achou mais fácil de usar?	<b>6</b>	1
Você acha que precisaria de ajuda de uma pessoa com conhecimento técnico para utilizar um dos sites?	0	0
Qual dos sites as pessoas conseguirão utilizar mais rapidamente?	<b>7</b>	1
Qual dos sistemas você achou mais complicado de usar?	1	<b>2</b>
Qual sistema você se sentiu mais confiante de usar?	<b>9</b>	0
Você precisou aprender um monte de coisas antes de utilizar um dos sistemas?	0	0
Com qual sistema você se sentiu mais confortável?	<b>9</b>	1
Qual interface você mais gostou?	<b>11</b>	0
A organização de informações é mais clara em qual dos sistemas?	<b>10</b>	1

Fonte: Elaborado pelo Autor, 2024.

## 6 CONCLUSÃO

Os avanços na inteligência artificial têm gerado grande expectativa em diversas áreas, e o desenvolvimento de software não é exceção. A IA promete revolucionar a forma como os programas são escritos, testados e mantidos, tornando o processo mais eficiente e acessível. Aspira-se que, no futuro, sistemas de IA como o ChatGPT possam assumir tarefas complexas de codificação com mínima intervenção humana, permitindo que desenvolvedores se concentrem em aspectos mais criativos e estratégicos do desenvolvimento.

A integração crescente da IA no desenvolvimento de software também levanta questões filosóficas sobre a substituição de seres humanos em tarefas repetitivas e braçais. Embora existam preocupações legítimas sobre a automação de empregos, os resultados do estudo indicam que a substituição completa dos desenvolvedores humanos ainda está distante. Existe, de fato, uma superioridade da ferramenta em gerar códigos sem tantas falhas diretas. Porém, como foi mostrado, essa habilidade também vem carregada com uma série de potenciais problemas, caso mal guiada. Além disso, o descontentamento dos usuários com a interface gerada via ChatGPT também demonstra uma deficiência no quesito, justificável, claro, por se tratar de um modelo de linguagem e não necessariamente de imagem.

Contudo, é importante reforçar que esse estudo se manteve limitado apenas à plataforma desenvolvida pela OpenAI, deixando de analisar várias ferramentas que complementam o próprio ChatGPT no quesito de desenvolvimento. Ainda, aplicações como a DevinAI<sup>18</sup>, por exemplo, que se propõe a ser um engenheiro de software feito 100% com inteligência artificial, já estão em fases bem mais avançadas no que diz respeito à substituição do desenvolvedor.

Em última análise, a IA se mostrou uma ferramenta valiosa no desenvolvimento de software, especialmente se explorada em sintonia com o desenvolvedor humano. A capacidade de resolver problemas complexos, inovar e tomar decisões informadas é algo que tende a ser cada vez mais automatizável e, com isso, fica nítida a necessidade crescente do desenvolvedor se adaptar a esse

---

<sup>18</sup> <https://www.cognition.ai/blog/introducing-devin>

meio de trabalho tão volátil, somada a importância de uma garantia social e econômica da divisão ética do trabalho.



## REFERÊNCIAS

ACEMOGLU, Daron; RESTREPO, Pascual. **Artificial intelligence, automation, and work**. In: The economics of artificial intelligence: An agenda. University of Chicago Press, 2018. p. 197-236.

PINTO, Sara Catarina Silva. **Processamento de linguagem natural e extração de conhecimento**. 2015. Dissertação de Mestrado.

<https://www.reuters.com/technology/chatgpt-sets-record-fastest-growing-user-base-analyst-note-2023-02-01/>

SOUZA, Eduardo. **Entendendo a Tokenização nos Modelos de Inteligência Artificial**. Disponível em:

<https://www.dio.me/articles/entendendo-a-tokenizacao-nos-modelos-de-inteligencia-artificial>. Acesso em: 20 de jul. de 2024.

BARROS, João Paulo. **Casos de uso e respectivos diagramas**. Escola Superior de Tecnologia e Gestão Instituto Politécnico de Beja, 2009.

GONZALEZ, Marco; LIMA, Vera Lúcia Strube. **Recuperação de informação e processamento da linguagem natural**. In: XXIII Congresso da Sociedade Brasileira de Computação. sn, 2003. p. 347-395.

GUINNESS, Harry. **Como o ChatGPT funciona**. Zapier, 2024. Disponível em: <<https://zapier.com/blog/how-does-chatgpt-work/>>. Acesso em: 05 de jul. de 2024.

IGNÁCIO, Andrew Carvalho; DA SILVA OLIVEIRA, Lucas; FRANCEZ, Marcos Paulo Moraes. **Eficiência do Uso da Inteligência Artificial no Desenvolvimento de Software**. Advances in Global Innovation & Technology, v. 2, n. 2, p. 06-16, 2024.

TRABELSI, Mohamed Ali. **The impact of artificial intelligence on economic development**. Journal of Electronic Business & Digital Economics, [S.l.], v. 2, n. 1, p. 1-20, 2024. Disponível em: <https://www.emerald.com/insight/content/doi/10.1108/JEBDE-10-2023-0022/full/html>. Acesso em: 16 jul. 2024.

SICHMAN, Jaime Simão. **Inteligência Artificial e sociedade: avanços e riscos**. Estudos Avançados, v. 35, p. 37-50, 2021.

GARCIA, Ana Cristina. **Ética e inteligência artificial**. Computação Brasil, n. 43, p. 14-22, 2020.

LOMIO, Francesco; MORESCHINI, Sergio; LENARDUZZI, Valentina. **A machine and deep learning analysis among SonarQube rules, product, and process metrics for fault prediction**. Empirical Software Engineering, v. 27, n. 7, p. 189, 2022.

FRANCK, Kewry Mariobo; PEREIRA, Robson Fernandes; DANTAS FILHO, Jerônimo Vieira. **Diagrama Entidade-Relacionamento: uma ferramenta para modelagem de dados conceituais em Engenharia de Software**. Research, Society and Development, v. 10, n. 8, p. e49510817776-e49510817776, 2021.

SOTNIK, Svitlana; MANAKOV, Volodymyr; LYASHENKO, Vyacheslav. **Overview: PHP and MySQL features for creating modern web projects**. 2023.

DISSANAYAKE, Nalaka R.; DIAS, G. K. A. **Best practices for rapid application development of AJAX based Rich Internet Applications**. In: 2014 14th International Conference on Advances in ICT for Emerging Regions (ICTer). IEEE, 2014. p. 63-66.

MYSQL, A. B. **MySQL**. 2001.

WHITE, Jules et al. **A prompt pattern catalog to enhance prompt engineering with chatgpt**. arXiv preprint arXiv:2302.11382, 2023.

DE MELLO, Rafael Maiani. **Técnica para Inspeção de Diagramas de Atividades**. 2011. Tese de Doutorado. Universidade Federal do Rio de Janeiro.

CATALANO, José Victor Rodrigues; LORENZI, Bruno Rossi. **Sem Referências: o ChatGPT sob a perspectiva latouriana e a armadilha do Duplo Clique**. Revista Faz Ciência, v. 25, n. 41, 2023.

NIELSEN, J. **10 Usability Heuristics for User Interface Design**. 1995. Disponível em:

<http://www.nngroup.com/articles/ten-usability-heuristics/>. Acesso em 18/07/2024.

SANTOS, José Amancio M. et al. **A systematic review on the code smell effect**. Journal of Systems and Software, v. 144, p. 450-477, 2018.

JUERGENS, Elmar et al. **Do code clones matter?**. In: 2009 IEEE 31st International Conference on Software Engineering. IEEE, 2009. p. 485-495.

SAURO, Jeff. **Measuring Usability With The System Usability Scale (SUS)**. 2011. Disponível em: <http://www.measuringu.com/sus>. Acesso em 18 jul. 2024.

## APÊNDICE A - TABELAS DE OBJETIVOS PRINCIPAIS DO CHATGPT

Objetivo	Iniciar desenvolvimento do sistema
Prompts	<p>“Preciso de uma loja virtual de discos e vinis, feita inteiramente pelo ChatGPT. Serão solicitados os seguintes passos:            Levantamento de Requisitos            Diagrama de Casos de Uso            Diagrama de Atividade            ER            Desenvolvimento            Testes            As ferramentas serão: PHP, Xampp com mySql para o banco de dados, Javascript, HTML e CSS. Pode iniciar levantando requisitos funcionais e não funcionais para o sistema. “;</p>
Tentativas	1

Objetivo	Criar o Diagrama de Casos de Uso com a ferramenta PlantUML
Prompts	“Continue com o Diagrama de Casos de Uso, usando a ferramenta PlantUML”
Tentativas	1

Objetivo	Criar o Diagrama de Atividade
Prompts	“Excelente, agora prossiga com o diagrama de atividade”
Tentativas	1

Objetivo	Criar o diagrama de Entidade e Relacionamento
Prompts	“Certo, agora prossiga com o diagrama Entidade Relacionamento, para o banco de dados.”
Tentativas	1

Objetivo	Prosseguir com o desenvolvimento do sistema
Prompts	<p>“Certo, agora prossiga com o desenvolvimento do sistema.”;            “Como solicitado anteriormente, o desenvolvimento do sistema deve ser completamente feito pelo ChatGPT, portanto, prossiga desenvolvendo o sistema do 0, obedecendo os requisitos e diagramas definidos anteriormente na conversa.”.</p>
Tentativas	2
Observações	Na primeira tentativa, a resposta mostrava o passo a passo do desenvolvimento, e não oferecia os códigos por conta própria.

Objetivo	Prosseguir com o desenvolvimento do sistema
Prompts	“Certo, agora prossiga com o desenvolvimento do sistema.”; “Como solicitado anteriormente, o desenvolvimento do sistema deve ser completamente feito pelo ChatGPT, portanto, prossiga desenvolvendo o sistema do 0, obedecendo os requisitos e diagramas definidos anteriormente na conversa.”.
Tentativas	2
Observações	Na primeira tentativa, a resposta mostrava o passo a passo do desenvolvimento, e não oferecia os códigos por conta própria.

Objetivo	Criar estilos em css para as páginas
Prompts	“Crie CSS para as páginas”; “O CSS não está bom, deixe ele melhor em todas as páginas.”; “O CSS não está mudando nada na página de registro, e alguns elementos estão desalinhados.” “as cores, fontes e distribuição do css ainda estão muito sem vida, consegue fazer algo melhor?” “o título e o header estão na mesma cor, e as distribuições dos elementos ainda estão muito espalhados.” “O título da página segue na mesma cor do header” “o login nao tem header nem footer” “Conserte o CSS”
Tentativas	8

Objetivo	Adicionar botão de redirecionamento para o carrinho de compras no menu principal
Prompts	“Não há como chegar ao carrinho de compras”
Tentativas	1

Objetivo	Adicionar produtos ao catálogo
Prompts	“Preciso adicionar produtos ao catálogo”; “Agora, preciso que tenha produtos no catálogo”.
Tentativas	2
Observações	O primeiro input foi mal inserido, pois ao dizer que “preciso adicionar”, a ferramenta entendeu que o ato de inserir os produtos de fato partiria de um usuário, assim, criando uma página nova de adição de produtos por acesso via administrador.

Objetivo	Adicionar função de fechar a compra
Prompts	“Não consigo fechar a compra”
Tentativas	1

## APÊNDICE B - DEMAIS RESULTADOS E INTERAÇÕES DA CHATGPT

Figura 1 - Análise de Requisitos Gerada pelo ChatGPT

Requisitos Funcionais:	Requisitos Não Funcionais:
<b>1. Cadastro de Usuário:</b> <ul style="list-style-type: none"> <li>Os usuários devem poder se registrar na plataforma fornecendo informações básicas como nome, endereço, e-mail e senha.</li> <li>Deve haver a opção de login para usuários já cadastrados.</li> </ul>	<b>1. Desempenho:</b> <ul style="list-style-type: none"> <li>O sistema deve ser responsivo e oferecer tempos de carregamento rápidos, mesmo durante períodos de tráfego intenso.</li> <li>Garantir que o sistema possa lidar com um grande volume de transações simultâneas.</li> </ul>
<b>2. Catálogo de Produtos:</b> <ul style="list-style-type: none"> <li>Exibir uma lista de discos e vinis disponíveis para compra.</li> <li>Permitir aos usuários filtrar os produtos por categoria, artista, gênero, etc.</li> <li>Mostrar detalhes dos produtos, incluindo imagens, descrições, preço e disponibilidade.</li> </ul>	<b>2. Segurança:</b> <ul style="list-style-type: none"> <li>Implementar medidas de segurança robustas para proteger os dados dos usuários, como criptografia de senha e comunicação segura.</li> <li>Garantir conformidade com regulamentações de proteção de dados, como GDPR.</li> </ul>
<b>3. Carrinho de Compras:</b> <ul style="list-style-type: none"> <li>Permitir aos usuários adicionar itens ao carrinho de compras.</li> <li>Permitir aos usuários editar a quantidade de itens no carrinho.</li> <li>Calcular o subtotal e o total do pedido no carrinho de compras.</li> </ul>	<b>3. Usabilidade:</b> <ul style="list-style-type: none"> <li>O sistema deve ser intuitivo e fácil de usar para usuários de todos os níveis de habilidade.</li> <li>Oferecer uma experiência de usuário consistente em diferentes dispositivos e navegadores.</li> </ul>
<b>4. Checkout:</b> <ul style="list-style-type: none"> <li>Oferecer um processo de checkout simplificado.</li> <li>Solicitar informações de pagamento e envio.</li> <li>Calcular o total da compra, incluindo impostos e custos de envio.</li> </ul>	<b>4. Confiabilidade:</b> <ul style="list-style-type: none"> <li>Minimizar o tempo de inatividade do sistema por meio de testes rigorosos e monitoramento contínuo.</li> <li>Implementar backups regulares dos dados do sistema para evitar perda de informações importantes.</li> </ul>
<b>5. Gestão de Pedidos:</b> <ul style="list-style-type: none"> <li>Permitir aos usuários visualizar o histórico de pedidos.</li> <li>Enviar e-mails de confirmação de pedidos para os usuários.</li> <li>Permitir aos usuários rastrear o status de seus pedidos.</li> </ul>	<b>5. Compatibilidade:</b> <ul style="list-style-type: none"> <li>Garantir que o sistema seja compatível com diferentes navegadores da web e dispositivos, como desktops, tablets e smartphones.</li> <li>Certificar-se de que o sistema seja compatível com as versões mais recentes das tecnologias web utilizadas.</li> </ul>
<b>6. Administração do Sistema:</b> <ul style="list-style-type: none"> <li>Permitir aos administradores adicionar, editar e remover produtos.</li> <li>Gerenciar pedidos e atualizar o status do pedido.</li> <li>Visualizar relatórios de vendas e estatísticas.</li> </ul>	

Fonte: Elaborado pelo Autor, 2024.

Figura 2 - Requisição de Código PlantUML do Diagrama de Atividade e a Resposta Provida

Excelente, agora prossiga com o diagrama de atividade

Vamos criar um Diagrama de Atividade para representar o processo de compra de um produto na loja virtual de discos e vinis.

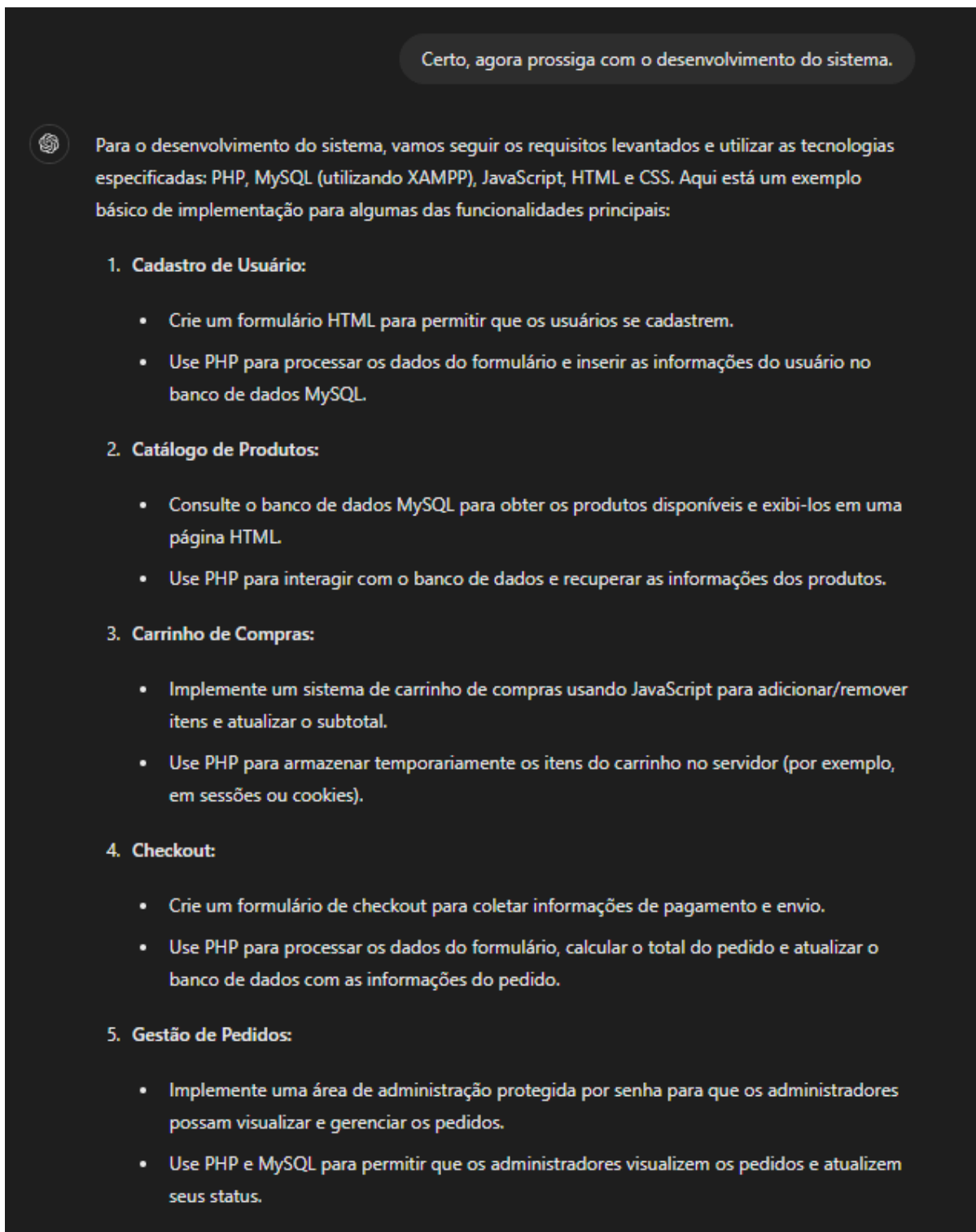
```

plantuml
@startuml
start
:Usuário seleciona produto;
if (Produto disponível?) then (Sim)
:Adicionar produto ao carrinho;
else (Não)
:Exibir mensagem de produto indisponível;
endif
:Usuário visualiza carrinho;
:Usuário seleciona opção de checkout;
if (Usuário autenticado?) then (Sim)
:Solicitar informações de pagamento e envio;
else (Não)
:Redirecionar para página de login;
endif
:Usuário confirma pedido;
if (Pagamento aprovado?) then (Sim)
:Enviar e-mail de confirmação;
:Atualizar estoque;
:Redirecionar para página de agradecimento;
else (Não)
:Exibir mensagem de erro de pagamento;
endif
stop
@enduml

```

Fonte: Elaborado pelo Autor, 2024.

Figura 3 - Primeira Tentativa Falha de Gerar Código com a ChatGPT



Certo, agora prossiga com o desenvolvimento do sistema.

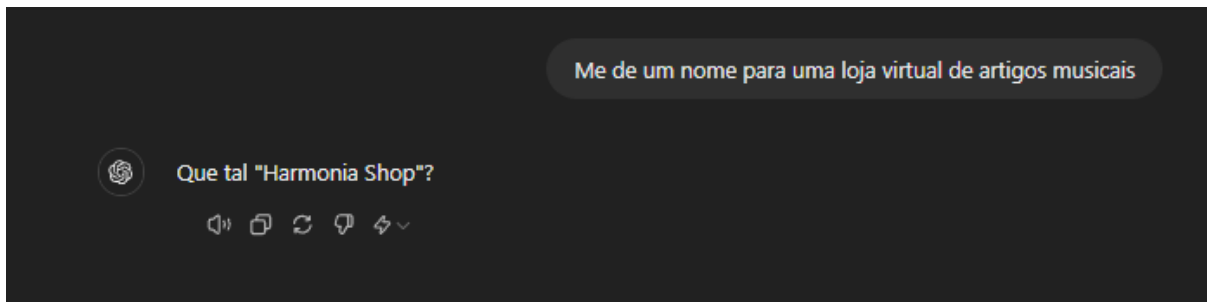
Para o desenvolvimento do sistema, vamos seguir os requisitos levantados e utilizar as tecnologias especificadas: PHP, MySQL (utilizando XAMPP), JavaScript, HTML e CSS. Aqui está um exemplo básico de implementação para algumas das funcionalidades principais:

- Cadastro de Usuário:**
  - Crie um formulário HTML para permitir que os usuários se cadastrem.
  - Use PHP para processar os dados do formulário e inserir as informações do usuário no banco de dados MySQL.
- Catálogo de Produtos:**
  - Consulte o banco de dados MySQL para obter os produtos disponíveis e exibí-los em uma página HTML.
  - Use PHP para interagir com o banco de dados e recuperar as informações dos produtos.
- Carrinho de Compras:**
  - Implemente um sistema de carrinho de compras usando JavaScript para adicionar/remover itens e atualizar o subtotal.
  - Use PHP para armazenar temporariamente os itens do carrinho no servidor (por exemplo, em sessões ou cookies).
- Checkout:**
  - Crie um formulário de checkout para coletar informações de pagamento e envio.
  - Use PHP para processar os dados do formulário, calcular o total do pedido e atualizar o banco de dados com as informações do pedido.
- Gestão de Pedidos:**
  - Implemente uma área de administração protegida por senha para que os administradores possam visualizar e gerenciar os pedidos.
  - Use PHP e MySQL para permitir que os administradores visualizem os pedidos e atualizem seus status.

Fonte: Elaborado pelo Autor, 2024.

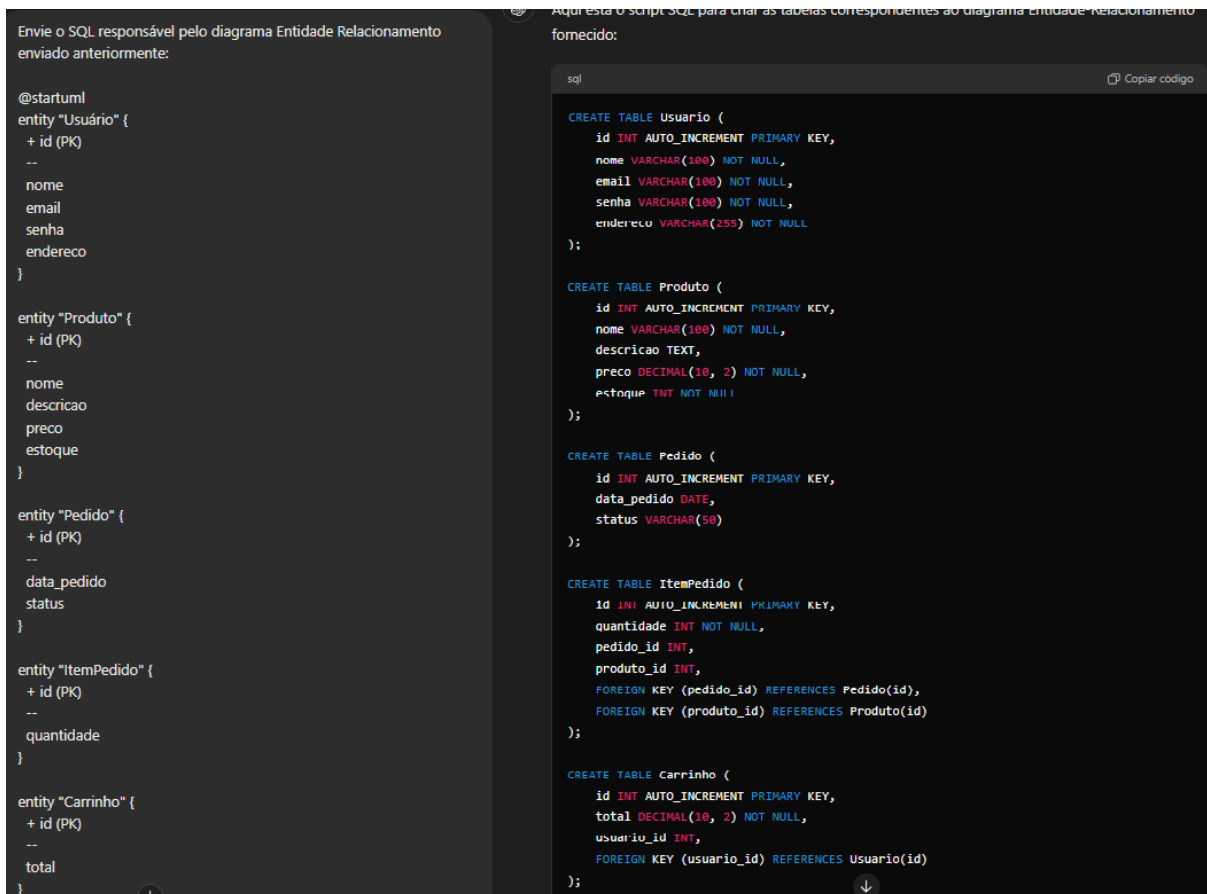


Figura 4 - Solicitação de Nome para Loja de Artigos Musicais.



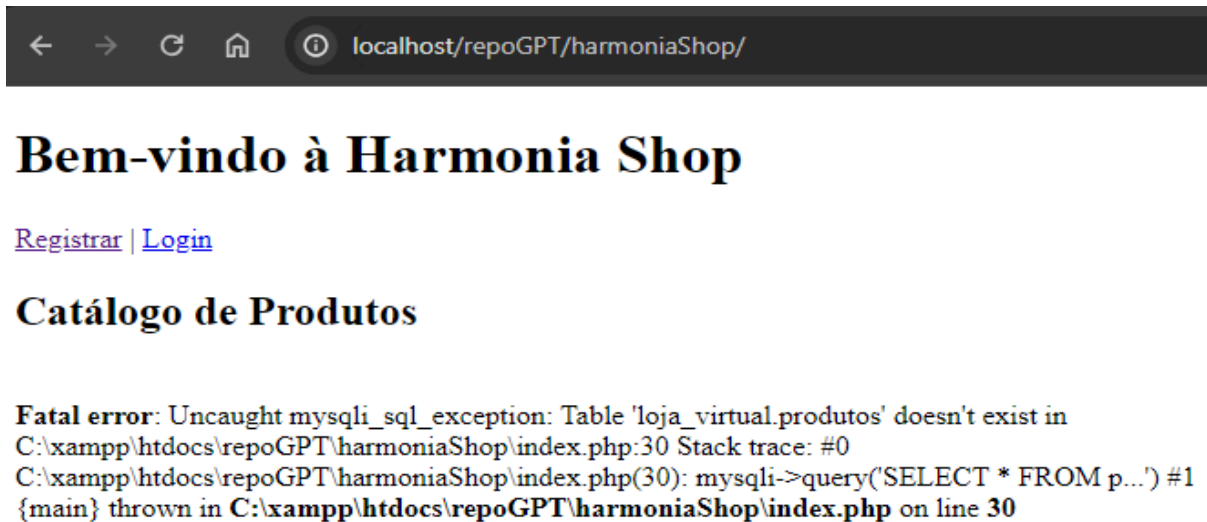
Fonte: Elaborado pelo Autor, 2024.

Figura 5 - Solicitação de Código SQL para Criação do Banco de Dados



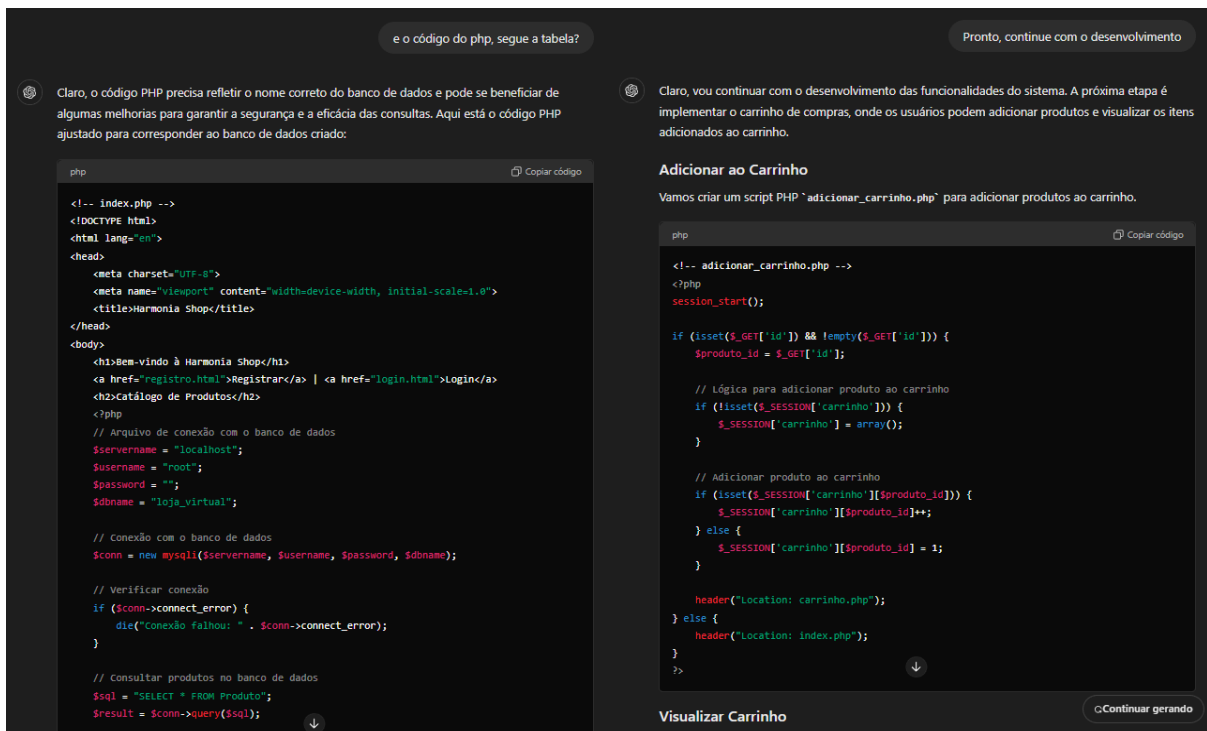
Fonte: Elaborado pelo Autor, 2024.

Figura 6 - Erro SQL na Primeira Página de Login Gerada



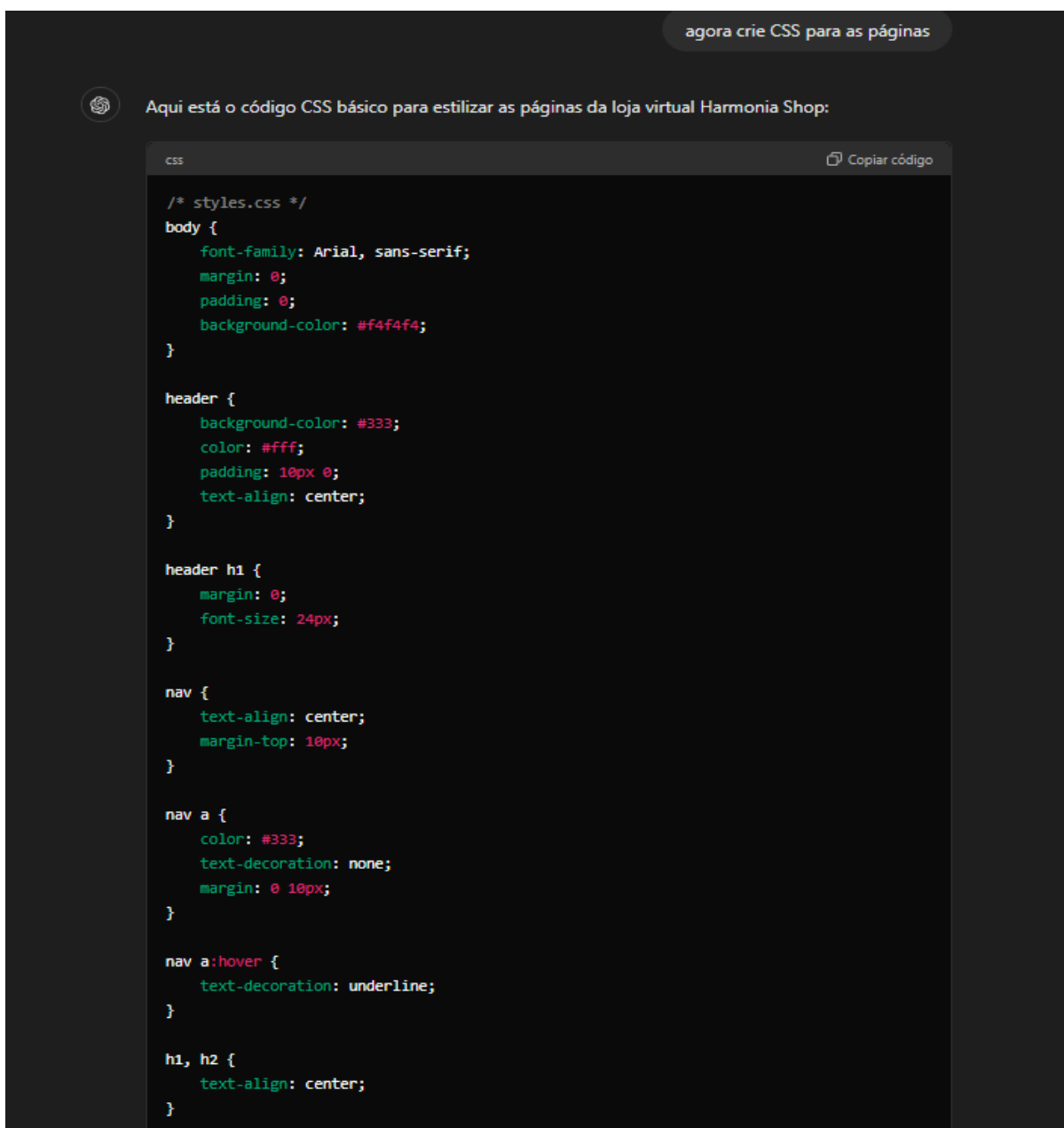
Fonte: Elaborado pelo Autor, 2024.

Figura 7 - Intervenção para Correção de Erro por Alucinação



Fonte: Elaborado pelo Autor, 2024.

Figura 8 - Exemplo de Solicitação de Código CSS para Interface da Aplicação



The image shows a dark-themed code editor window. At the top right, there is a button that says "agora crie CSS para as páginas". Below it, a text prompt reads "Aqui está o código CSS básico para estilizar as páginas da loja virtual Harmonia Shop:". The main area of the editor contains CSS code for a file named "styles.css". The code defines styles for the body, header, header h1, navigation (nav), navigation links (nav a), navigation link hover state (nav a:hover), and h1/h2 elements. The code is as follows:

```
css Copiar código

/* styles.css */
body {
  font-family: Arial, sans-serif;
  margin: 0;
  padding: 0;
  background-color: #f4f4f4;
}

header {
  background-color: #333;
  color: #fff;
  padding: 10px 0;
  text-align: center;
}

header h1 {
  margin: 0;
  font-size: 24px;
}

nav {
  text-align: center;
  margin-top: 10px;
}

nav a {
  color: #333;
  text-decoration: none;
  margin: 0 10px;
}

nav a:hover {
  text-decoration: underline;
}

h1, h2 {
  text-align: center;
}
```

Fonte: Elaborado pelo Autor, 2024.

Figura 9 - Solicitação de Adição de Produtos no Catálogo

Agora preciso que tenha produtos no catálogo

Para adicionar produtos ao catálogo, você pode inserir alguns registros na tabela `Produto` do seu banco de dados MySQL. Vou fornecer um script SQL para inserir produtos de exemplo no banco de dados.

**Passo 1: Inserir Produtos no Banco de Dados**

Execute o seguinte script SQL no seu banco de dados MySQL para adicionar produtos de exemplo à tabela `Produto`:

```
sql Copiar código
USE loja_virtual;

INSERT INTO Produto (nome, descricao, preco, estoque) VALUES
('Vinil A', 'Álbum clássico de rock', 120.00, 10),
('Vinil B', 'Álbum de jazz', 100.00, 8),
('Vinil C', 'Álbum de blues', 130.00, 5),
('Vinil D', 'Álbum de música clássica', 150.00, 7),
('Vinil E', 'Álbum de pop', 90.00, 12);
```

**Passo 2: Verificar o Catálogo de Produtos**

Agora que os produtos foram adicionados ao banco de dados, você deve ser capaz de vê-los no catálogo de produtos ao acessar a página `index.php` da Harmonia Shop. O `index.php` foi configurado para consultar e exibir produtos da tabela `Produto`.

Fonte: Elaborado pelo Autor, 2024.

Figura 10 - Produtos Adicionados ao Catálogo

**Bem-vindo à Harmonia Shop**

[Registrar](#) | [Login](#) | [Ver Carrinho](#)

**Catálogo de Produtos**

Nome	Descrição	Preço	Ação
Vinil A	Álbum clássico de rock	R\$ 120,00	<a href="#">Adicionar ao Carrinho</a>
Vinil B	Álbum de jazz	R\$ 100,00	<a href="#">Adicionar ao Carrinho</a>
Vinil C	Álbum de blues	R\$ 130,00	<a href="#">Adicionar ao Carrinho</a>
Vinil D	Álbum de música clássica	R\$ 150,00	<a href="#">Adicionar ao Carrinho</a>
Vinil E	Álbum de pop	R\$ 90,00	<a href="#">Adicionar ao Carrinho</a>

Fonte: Elaborado pelo Autor, 2024.

Figura 11 - Solicitação de Correção de Problema ao Fechar a Compra

Não consigo fechar a compra

Para permitir que os usuários fechem a compra, precisamos adicionar uma funcionalidade de checkout que permita aos usuários finalizar o pedido e processar o pagamento. Aqui estão os passos para implementar essa funcionalidade:

### Passo 1: Adicionar Botão "Fechar Compra" no Carrinho

Primeiro, vamos adicionar um botão "Fechar Compra" na página do carrinho.

Atualização do `carrinho.php`

```
php Copiar código
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Carrinho de Compras - Harmonia Shop</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <header>
    <h1>Carrinho de Compras</h1>
  </header>
  <nav>
    <a href="index.php">Home</a>
  </nav>
```

Fonte: Elaborado pelo Autor, 2024.

Figura 12 - Solicitação de Correção de Erro SQL ao Fechar a Compra

```
trace: #0 C:\xampp\htdocs\repoGPT\harmoniaShop\checkout.php(55):
mysqli->query('INSERT INTO Ite...') #1 {main} thrown in
C:\xampp\htdocs\repoGPT\harmoniaShop\checkout.php on line 55
```

Esse erro começou a aparecer ao fechar a compra

Esse erro indica que a coluna `preco` não existe na tabela `ItemPedido`. Vamos corrigir isso certificando-se de que a coluna `preco` seja adicionada à tabela `ItemPedido`. Se a coluna já existir e o erro persistir, pode ser necessário verificar a estrutura do banco de dados.

### Passo 1: Verificar e Corrigir a Estrutura da Tabela `ItemPedido`

Certifique-se de que a tabela `ItemPedido` tenha a coluna `preco`. Vamos usar `ALTER TABLE` para garantir que ela esteja presente.

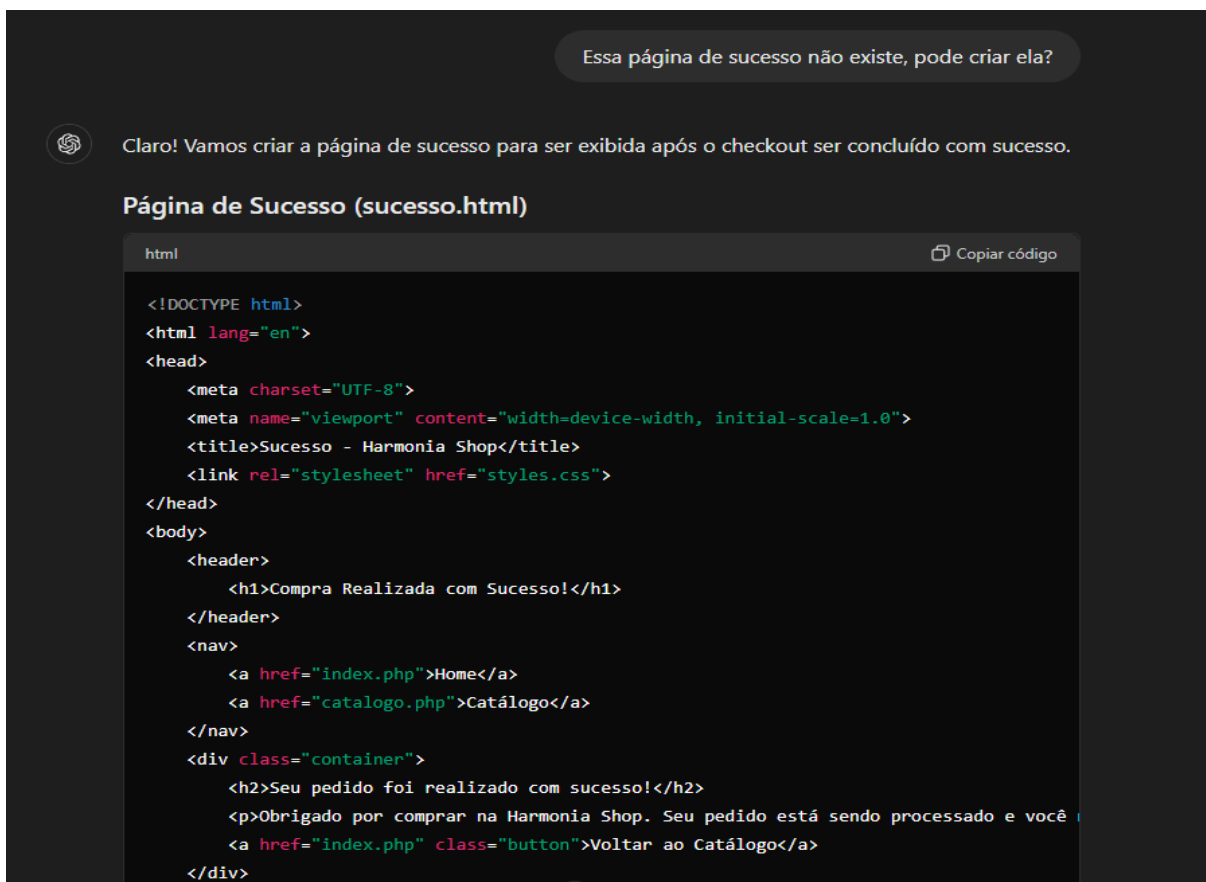
```
sql Copiar código
ALTER TABLE ItemPedido
ADD COLUMN preco DECIMAL(10, 2);
```

### Passo 2: Atualizar o Código PHP

Aqui está o código atualizado para `checkout.php`, com algumas verificações adicionais para garantir que o processo de inserção esteja correto:

Fonte: Elaborado pelo Autor, 2024.

Figura 13 - Problema de Redirecionamento para Página Inexistente ao Fechar Compra



Essa página de sucesso não existe, pode criar ela?

Claro! Vamos criar a página de sucesso para ser exibida após o checkout ser concluído com sucesso.

### Página de Sucesso (sucesso.html)

```
html Copiar código<!DOCTYPE html><html lang="en"><head><meta charset="UTF-8"><meta name="viewport" content="width=device-width, initial-scale=1.0"><title>Sucesso - Harmonia Shop</title><link rel="stylesheet" href="styles.css"></head><body><header><h1>Compra Realizada com Sucesso!</h1></header><nav><a href="index.php">Home</a><a href="catalogo.php">Catálogo</a></nav><div class="container"><h2>Seu pedido foi realizado com sucesso!</h2><p>Obrigado por comprar na Harmonia Shop. Seu pedido está sendo processado e você receberá um e-mail com os detalhes da compra.</p><a href="index.php" class="button">Voltar ao Catálogo</a></div>
```

Fonte: Elaborado pelo Autor, 2024.

Figura 14 - Página de Sucesso Criada



**Compra Realizada com Sucesso!**

[Home](#) [Catálogo](#)

**Seu pedido foi realizado com sucesso!**

Obrigado por comprar na Harmonia Shop. Seu pedido está sendo processado e você receberá um e-mail com os detalhes da compra.

[Voltar ao Catálogo](#)

Fonte: Elaborado pelo Autor, 2024.

Figura 15 - Problema Funcional de Fluxo de Login

Ao logar, ainda aparecem as opções de fazer login e registrar, mesmo já estando em uma conta

Para resolver isso, precisamos ajustar o menu de navegação para exibir diferentes opções dependendo se o usuário está logado ou não. Vamos modificar o `index.php` e outros arquivos PHP que contêm a navegação para verificar se o usuário está logado e alterar as opções de menu adequadamente.

**Modificação do `index.php`**

Primeiro, vamos garantir que haja uma sessão iniciada para verificar se o usuário está logado:

Fonte: Elaborado pelo Autor, 2024.

Figura 16 - Fluxo de Login do Usuário Corrigido.

## Ajuste no HTML para verificar se o usuário está logado

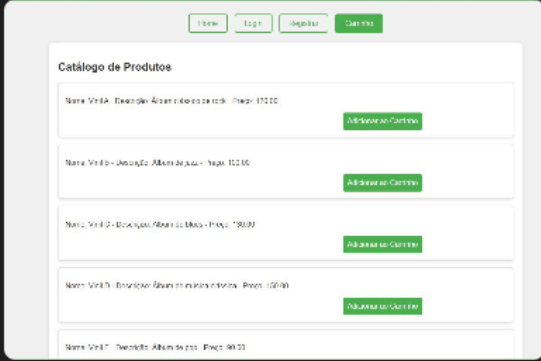
Vamos adicionar lógica condicional no arquivo `index.php` para exibir os links de "Login" e "Registrar" apenas se o usuário não estiver logado.

```
php Copiar código  
  
<!-- index.php -->  
<?php  
session_start();  
$isLoggedIn = isset($_SESSION['email']);  
>  
  
<!DOCTYPE html>  
<html lang="en">  
<head>  
  <meta charset="UTF-8">  
  <meta name="viewport" content="width=device-width, initial-scale=1.0">  
  <title>Harmonia Shop</title>  
  <link rel="stylesheet" href="styles.css">  
</head>  
<body>  
  <header>  
    <h1>Bem-vindo à Harmonia Shop</h1>  
  </header>  
  <nav>  
    <a href="index.php">Home</a>  
    <?php if (!$isLoggedIn): ?>  
      <a href="login.html">Login</a>  
      <a href="register.html">Registrar</a>  
    <?php endif; ?>  
    <a href="carrinho.php">Carrinho</a>  
  </nav>  
  <div class="container">  
    <h2>Catálogo de Produtos</h2>  
    <!-- Exemplo de produto -->  
    <?php  
    // Conexão com o banco de dados  
    $servername = "localhost";  
    $username = "root";  
    $password = "";  
    $dbname = "loja_virtual";  
  
    $conn = new mysqli($servername, $username, $password, $dbname);  
  
    // Verificar conexão  
    if ($conn->connect_error) {  
      die("Conexão falhou: " . $conn->connect_error);  
    }  
  }  
</div>  
</body>  
</html>
```

Fonte: Elaborado pelo Autor, 2024.



Figura 17 - Última Correção Utilizando a Funcionalidade de Adicionar Imagens a Requisição



O CSS está deixando os botões escondidos. Além disso, voltou o problema do login e registro aparecerem mesmo com o usuário logado.

Para corrigir o problema dos botões que estão "escondidos" e garantir que os links de "Login" e "Registrar" não apareçam quando o usuário está logado, vou fazer os seguintes ajustes:

1. **Ajustar a exibição dos botões no catálogo de produtos.**
2. **Adicionar lógica condicional no HTML para verificar se o usuário está logado.**

Fonte: Elaborado pelo Autor, 2024.

## APÊNDICE C - FORMULÁRIO DE COMPARAÇÃO DOS SISTEMAS

# Comparação de Sistemas

### LINKS PARA OS SITES:

[Site A](#)

[Site B](#)

**ATENÇÃO:** Devido ao sistema de hospedagem dos sites, talvez seja necessário atualizar a página e aguardar alguns segundos até as funcionalidades aparecerem por completo.

Acesse os sites acima e complete as seguintes tarefas em **ambos**:

- Registrar-se no site;
- Fazer login utilizando nome de usuário e senha;
- Acessar o perfil;
- Adicionar dois ou mais produtos ao carrinho;
- Acessar o carrinho;
- Fechar a compra;
- Sair da conta.

Após completar as tarefas em ambos os sites, responda o questionário de comparação.

**Lembre-se de responder todas as perguntas com honestidade, afim de evitar erros de coleta de dados na pesquisa.**

Você conseguiu efetuar as tarefas nos dois sistemas? \*

- Sim
- Não

Você achou algum sistema desnecessariamente complexo? \*

- Site A
- Site B
- Ambos os sites são igualmente desnecessariamente complexos
- Nenhum dos sites é desnecessariamente complexo
- Não sei responder

Qual sistema você achou mais fácil de usar? \*

- Site A
- Site B
- Ambos foram igualmente fáceis de usar
- Ambos foram igualmente difíceis de usar
- Não sei responder

Você acha que precisaria de ajuda de uma pessoa com conhecimento técnico para utilizar um dos sites? \*

- Site A
- Site B
- Precisaria de ajuda em ambos os sites
- Não precisaria de ajuda em nenhum dos sites
- Não sei responder

Qual dos sites as pessoas conseguirão utilizar mais rapidamente? \*

- Site A
- Site B
- O uso de ambos os sites será aprendido rapidamente
- O uso de ambos os sites será aprendido lentamente
- Não sei responder

Qual dos sistemas você achou mais complicado de usar? \*

- Site A
- Site B
- Ambos os sistemas foram igualmente complicados de usar
- Ambos os sistemas foram tranquilos de usar
- Não sei responder

Qual sistema você se sentiu mais confiante de usar? \*

- Site A
- Site B
- Me senti igualmente confiante usando ambos os sistemas
- Me senti igualmente inseguro usando ambos os sistemas
- Não sei responder

Você precisou aprender um monte de coisas antes de utilizar um dos sistemas? \*

- Site A
- Site B
- Precisei aprender um monte de coisas antes de usar os dois sites
- Não precisei aprender nada novo antes de usar nenhum dos sistemas
- Não sei responder

Com qual sistema você se sentiu mais confortável? \*

- Site A
- Site B
- Me senti igualmente confortável com ambos os sistemas
- Me senti igualmente desconfortável com ambos os sistemas
- Não sei responder

Qual interface você mais gostou? \*

- Site A
- Site B
- Gostei igualmente da interface dos dois sistemas
- Não gostei igualmente da interface dos dois sistemas
- Não sei responder

A organização de informações é mais clara em qual dos sistemas? \*

- Site A
- Site B
- As informações de ambas interfaces são igualmente claras
- As informações de ambas interfaces são igualmente confusas
- Não sei responder

Você diria que algum desses sistemas foi desenvolvido via inteligência artificial? \*

- Site A
- Site B
- Ambos os sites
- Nenhum dos sites
- Não sei dizer

Alguma observação sobre os sistemas?

Sua resposta

---