

**COLÉGIO ESTADUAL DE EDUCAÇÃO PROFISSIONAL
PEDRO BOARETTO NETO
CURSO TÉCNICO EM INFORMÁTICA**

PEDRO JOSÉ CALDAS FREITAS

MEU PEQUENO QUADRO

**CASCADEL - PR
2024**

PEDRO JOSÉ CALDAS FREITAS

MEU PEQUENO QUADRO

Projeto de Desenvolvimento de Software
do Curso Técnico em Informática do
Colégio Estadual de Educação
Profissional Pedro Boaretto Neto –
Cascavel, Paraná.

Orientadores: Prof^a Aparecida S.Ferreira¹
Prof. Rony Peterson Cordeiro²
Prof. Cleonice Maria Pereira³

CASCADEL - PR
2024

¹Especialista em Educação Permanente: Saúde e educação pela FioCruz – Fundação Osvaldo Cruz. Especialista em tecnologias da Informação pela UNIVEL – União Educacional de Cascavel. Pedagoga formada pela UNIPAR – Universidade Paranaense. Professora do núcleo técnico do Estado do Paraná – Ensino médio técnico.

²

³Graduação em Especialização em PSICOPEDAGOGIA. (Carga Horária: 1200h). INSTITUTO DE EDUCAÇÃO SÉCULOS - FAVENI, FAVENI, Brasil. Especialização em Engenharia de Software com Métodos Ágeis. (Carga Horária: 480h). Universidade Cruzeiro do Sul, UNICSUL, Brasil. Título: Projeto Software. Orientador: Dr. Luís Nato Mendes Bezerra.

PEDRO JOSÉ CALDAS FREITAS**MEU PEQUENO QUADRO**

Este Projeto de Conclusão de Curso foi julgado e aprovado pelo Curso Técnico em Informática do Colégio Estadual de Educação Profissional Pedro Boaretto Neto.

Cascavel, Pr., xx de Xxxxx de 2024

COMISSÃO EXAMINADOR

<hr/> <div>Profª. Aparecida da S. Ferreira¹ Especialista em Tecnologia da Informação <i>Faculdade de Ciências Sociais</i> <i>Aplicadas de Cascavel</i> Orientadora</div> <hr/>	<hr/> <div>Prof. Rony Peterson Cordeiro Web Design</div> <hr/>
<div>Profª. Cleonice Maria Pereira Especialização em PSICOPEDAGOGIA. Banco de dados</div>	<div>Profª Márcio Alves dos Santos Especialista em Gestão e Docência no ensino superior, médio e técnico. Coordenador de curso</div>

Sumário

1	INTRODUÇÃO	5
1.1	Apresentação do Problema	6
2	OBJETIVOS	7
3	METODOLOGIA	8
4	REFERENCIAL TEÓRICO	10
5	DOCUMENTAÇÃO do projeto	12
5.1	Requisitos	16
5.1.1	Requisitos funcionais	16
5.1.2	Requisitos não funcionais	17
5.2	Diagrama de Contexto	18
5.3	Diagrama de Fluxo de dados	19
5.4	Diagrama de Entidade e relacionamento	20
5.5	Dicionário de Dados	22
5.6	Diagrama de Caso de Uso	24
5.7	Diagrama de Classe	25
5.8	Diagrama de Sequência	27
5.9	Diagrama de Atividade	28
6	Telas	29
7	Conclusão	33
8	REFERÊNCIAS	34

1 INTRODUÇÃO

GEMINI (2024), na era digital em constante evolução, a demanda por plataformas que facilitem a venda e a gestão de pedidos para quadros decorativos nunca foi tão alta. Diante desse cenário, surge a necessidade de um espaço virtual que não apenas permita aos criadores, designers e vendedores exporem e venderem seus produtos, mas também ofereça uma experiência integrada e interativa para a gestão de pedidos e relacionamento com os clientes. É nesse contexto que se insere nosso projeto: a criação de um site dedicado a auxiliar artistas na venda de seus produtos e no gerenciamento de pedidos, proporcionando uma plataforma completa e especializada para impulsionar seus negócios.

Para PRONEC@ (2019), a pintura refere-se genericamente à técnica de aplicar pigmento em forma pastosa, líquida ou em pó a uma superfície, a fim de colori-la, atribuindo-lhe matizes, tons e texturas.

Em um sentido mais específico, é a arte de pintar uma superfície, tais como papel, tela, ou uma parede (pintura mural ou afrescos). Devido ao fato de grandes obras de arte, tais como a Mona Lisa e A Última Ceia, do renascentista Leonardo Da Vinci, serem pinturas a óleo, a técnica é historicamente considerada uma das mais tradicionais das artes plásticas.

Conforme REYNOLDS (2020) com o crescimento da internet e do comércio eletrônico, ter uma presença online tornou-se essencial para qualquer negócio, incluindo uma loja de quadros. Um site de vendas permite que artistas alcancem um público mais amplo, ofereçam seus produtos de forma conveniente e aumentem suas vendas. Este documento explora a importância de sites de vendas para tais e como eles podem ser criados para maximizar o sucesso no mercado digital.

Além de aumentar a visibilidade, um site permite que segundos gerenciem melhor suas operações, como recebimento de pedidos, processamento de pagamentos, e até mesmo a coordenação de entregas. Com a integração de ferramentas de marketing digital, como SEO e redes sociais, podem atrair mais clientes e fidelizá-los ao longo do tempo.

Segundo a fala de SMITH (2019) A criação de um site de vendas é um passo crucial para artistas que desejam expandir seus negócios e alcançar novos clientes. Um site bem projetado, aliado a estratégias eficazes de marketing digital, pode

transformar uma pequena confeitaria em uma marca conhecida e respeitada. Portanto, investir em uma presença online não é mais uma opção, mas uma necessidade para quem quer prosperar no mercado competitivo de hoje.

1.1 Apresentação do Problema

O projeto Meu Pequeno Quadro tem como objetivo saciar a falta de alcance de confeitarias de casa, podendo atingir um número quase infinito de pessoas com uma loja online, onde poderá se encontrar grande variedade de obras de arte, incluindo quadro de intuíto didático, também podendo entrar em contato diretamente com o dono, assim garantindo uma alta facilidade de encontrar clientes, e os clientes encontrar sua loja.

2 OBJETIVOS

A criação de um site intuitivo e responsivo de uma loja de quadros.

Experiência Envolvente e Intuitiva: Garantir que o site seja fácil de usar e visualmente agradável, proporcionando uma navegação fluida e uma experiência de usuário (UX) positiva.

Destacar a Qualidade e Criatividade dos Produtos: Utilizar imagens de alta qualidade e descrições detalhadas para apresentar os produtos de confeitaria de forma atraente, destacando seus diferenciais e a criatividade dos itens oferecidos.

Facilitar a Navegação e a Compra Online: Implementar funcionalidades de e-commerce que permitam aos clientes explorar o catálogo de produtos, realizar pedidos e pagamentos de maneira simples e segura.

Promover a Marca e os Valores: Refletir os valores e a identidade da confeitaria através do design do site, conteúdos e campanhas de marketing, reforçando a imagem da marca e criando uma conexão com os clientes.

3 METODOLOGIA

O método comparativo compreende em estabelecer paralelos entre dois ou mais objetos de estudo, para analisar semelhanças e diferenças. Em outras palavras, é um método para comprovar ou refutar teorias e hipóteses que se baseia em comparações. Nas ideias de COELHO (2022), o método comparativo se baseia no método das semelhanças e das diferenças ou na observação de variações concomitantes“. Nessa perspectiva, no processo de comparação, identificar as semelhanças permite organizar e relacionar o novo conceito com o conhecimento que já existe. Por outro lado, estabelecer as diferenças permite discriminar o novo conceito de outros parecidos, para evitar confusão.

A Modelagem de Dados para DRUCKER (2024), é uma prática do campo da gestão e da análise de informações que busca representar visualmente a estrutura e as relações entre os dados em um sistema. Essencialmente, ela cria um mapa que descreve como as informações são organizadas e interagem dentro de um ambiente específico. Ao empregar técnicas e ferramentas adequadas, a modelagem de dados permite uma compreensão mais clara e sistemática deles, facilitando a interpretação, manipulação e tomada de decisões. Com isso ela desempenha um importante papel na integração de sistemas, no desenvolvimento de bancos de dados eficientes e na otimização de processos analíticos. Como funciona a modelagem de dados? A modelagem de dados é um processo que opera por meio de uma sequência de passos organizados, utilizando técnicas e ferramentas específicas para representar graficamente a estrutura, relacionamentos e características dos dados em um sistema. O primeiro passo é identificar os requisitos do sistema, ou seja, compreender as necessidades dos stakeholders em termos de dados. Os profissionais de modelagem de dados, em seguida, coletam informações relevantes para o contexto do sistema, incluindo dados sobre entidades, atributos, relacionamentos e restrições. Com base nos requisitos levantados são identificadas as entidades principais do sistema, em outras palavras, os objetos ou conceitos sobre os quais desejamos armazenar informações. Atributos, que são as características específicas das entidades, são associados a essas. Em seguida, estabelece-se relações entre diferentes entidades para representar como elas se conectam e interagem entre si. O processo de normalização é aplicado para eliminar redundâncias e melhorar a

eficiência do banco de dados, organizando as informações de maneira lógica e eficaz. Utilizando ferramentas gráficas, como diagramas de entidade-relacionamento (ER) ou diagramas UML com objetivo visualizar e comunicar a estrutura de dados de forma compreensível.

A implementação no banco de dados real ocorre a partir dos conceitos gerados durante o processo, traduzindo-os em esquemas físicos. Importante destacar que a modelagem de dados não é um processo estático, dessa forma, à medida que os requisitos evoluem, ajustes são realizados para garantir que ela continue a atender às necessidades em constante mudança do sistema.

4 REFERENCIAL TEÓRICO

HTML conforme EICH (2024), a base fundamental da web é o HTML (HyperText Markup Language). Essa linguagem define a estrutura e o significado do conteúdo que vemos nas páginas da internet. Imagine o HTML como os ossos de um corpo: ele dá forma e sustentação. Para entender melhor, podemos comparar uma página web a um livro. O HTML seria a estrutura do livro, definindo capítulos, parágrafos, títulos e imagens. Outras tecnologias, como o CSS, seriam responsáveis pela aparência do livro, como a fonte, as cores e o layout. Já o JavaScript adicionaria funcionalidades interativas, como menus suspensos ou animações. O termo "hipertexto" se refere à capacidade de conectar diferentes partes de um texto ou até mesmo páginas inteiras. Os links que você clica para navegar entre sites são exemplos de hipertexto em ação.

O **CSS** (Cascading Style Sheets ou Folhas de Estilo em Cascata) para EICH (2024), é como uma roupa para as páginas da internet. Ele define como os elementos de uma página, como títulos, parágrafos e imagens, serão exibidos na tela. Por exemplo, o CSS pode determinar a cor do texto, o tamanho das fontes, o espaçamento entre os elementos e até mesmo criar efeitos visuais como animações. Imagine uma página da web como um livro. O HTML (HyperText Markup Language) é como a estrutura do livro, definindo os capítulos, parágrafos e imagens. O CSS, por sua vez, é como a diagramação do livro, escolhendo as fontes, as cores e o layout das páginas. Para que serve o CSS? Estilização básica: Muda a cor e o tamanho dos títulos, links e outros elementos. Criação de layouts: Transforma uma simples coluna de texto em um layout mais complexo com colunas laterais e áreas de conteúdo. Efeitos visuais: Cria animações e outros efeitos para tornar a página mais interativa e interessante.

JavaScript para EICH (2024), é uma linguagem de programação que possibilita a criação de elementos sofisticados em sites. Sempre que uma página web vai além de exibir apenas informações estáticas — apresentando conteúdo que se renova periodicamente, mapas interativos ou gráficos animados em 2D/3D, por exemplo — é bem provável que o JavaScript esteja por trás disso. Essa linguagem faz parte da terceira camada da estrutura das tecnologias web, sendo que as duas primeiras (HTML e CSS) foram exploradas em mais profundidade em outras seções da Área de Aprendizado.

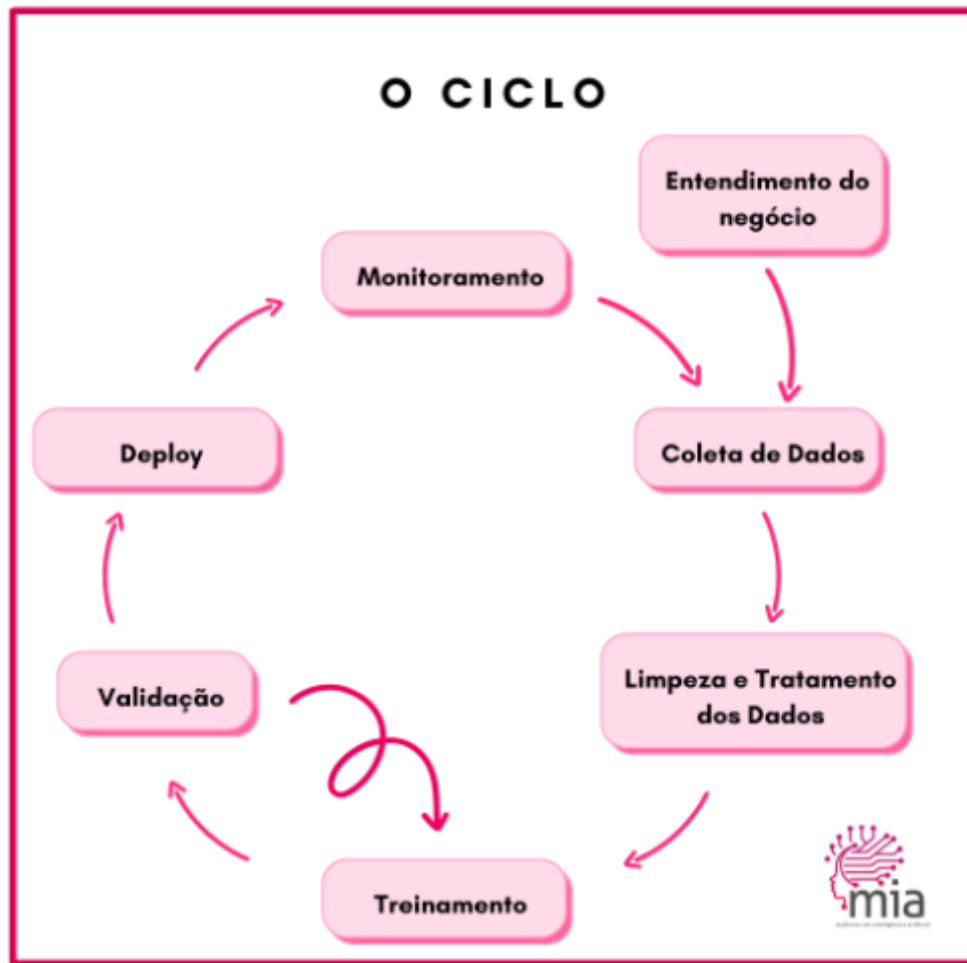
PHP, segundo CRISTO (2024), PHP: Hypertext Preprocessor, é uma linguagem de programação de código aberto muito popular, amplamente utilizada e ideal para criar aplicativos web, podendo ser integrada diretamente ao HTML. Ao invés de utilizar diversos comandos para exibir HTML (como em C ou Perl), as páginas PHP apresentam HTML integrado ao código, realizando determinada ação (como, por exemplo, exibir a mensagem "Olá, eu sou um script PHP!"). O código PHP é identificado pelas tags de início e fim das instruções de processamento, que possibilitam a transição para dentro e fora do "modo PHP". A diferença entre o PHP e o JavaScript no lado do cliente é que o PHP é executado no servidor, produzindo o HTML que é enviado para o navegador. O navegador apenas recebe os resultados da execução do script, sem ter acesso ao código fonte original. Uma opção é configurar o servidor para processar todos os arquivos HTML com PHP, o que impede que os usuários saibam o que está por trás das páginas. Uma das vantagens de adotar o PHP é a sua facilidade de uso para quem está começando garantindo ao mesmo tempo uma gama de funcionalidades avançadas para os programadores mais experientes. Não é preciso sentir receio ao se deparar com a extensa lista de recursos disponíveis no PHP. Aproveite ao máximo, aprenda rapidamente e comece a desenvolver scripts básicos em questão de poucas horas.

MySQL para CIRNE (2024), é um sistema de gerenciamento de banco de dados relacional de código aberto. O software opera em um modelo arquitetônico, permitindo que vários clientes se conectem ao servidor e acessem e manipulem os dados armazenados no banco de dados. Os dados são organizados em tabelas que consistem em linhas e colunas. Usando Structured Query Language (SQL), você pode realizar operações de criação, modificação e consulta de dados. A ferramenta oferece uma ampla gama de recursos e funcionalidades, incluindo suporte a transações, indexação eficiente, controle de acesso e segurança de dados. Além disso, o MySQL é conhecido por sua estabilidade, confiabilidade e facilidade de uso. Possui uma grande comunidade de desenvolvedores e usuários em todo o mundo, o que o ajuda a ser continuamente aprimorado e suportado.

5 DOCUMENTAÇÃO DO PROJETO

Para MESQUITA (2024) a documentação de projetos desempenha um papel fundamental no sucesso e na eficiência dos empreendimentos. Esses acompanhamentos, baseados na gestão empresarial, são o divisor de águas entre uma companhia amadora ou profissional. Independentemente do tamanho ou da natureza do projeto, a documentação abrange o registro de todas as informações relevantes, desde o planejamento inicial até a conclusão. Por isso, é como se resumisse e reunisse todas as informações importantes dele. Para entender melhor o conceito de documentação de projetos, continue lendo. Conheça a sua importância e confira orientações sobre como realizar esta documentação de maneira eficiente.

O que é a documentação de projetos? A documentação de projetos refere-se ao processo de registrar e organizar todas as informações relacionadas a um projeto. Isso inclui objetivos, escopo, cronograma, recursos, requisitos, estratégias, riscos, decisões e resultados alcançados. O objetivo principal é criar um histórico completo e estruturado do projeto, facilitando a compreensão, a colaboração e a referência futura. Importância da documentação de projetos. A documentação de projetos é essencial para o sucesso das atividades da empresa, produtividade e mais. Uma documentação clara e detalhada permite que todas as partes interessadas compreendam o projeto de maneira consistente. Ela serve como uma fonte confiável de informações, facilitando a comunicação entre a equipe do projeto, os clientes, os patrocinadores e outros envolvidos. Assim, com uma documentação adequada, evitam-se mal-entendidos e aumenta-se a eficiência das interações entre as partes. Aprendizado organizacional. A documentação de projetos permite que as organizações acumulem conhecimento e experiência ao longo do tempo.



Fonte: FREITAS, P. J. C., (2024).

Para OLIVEIRA (2024) tem está ideia sobre o ciclo de vida: Entendimento do negócio essa é uma das partes fundamentais do projeto. Afinal, para criar um modelo, é necessário entender o seu objetivo e um pouco do contexto no qual nos encontramos. No Projeto A, precisamos conversar com a área de negócios para entender, por exemplo: Qual o fluxo de vida do cliente dentro do negócio; O que, na visão deles, pode impactar esse cancelamento; Se existem variáveis (features, em inglês) sazonais — por exemplo, um período de reajuste de preço — ou regionais (como bairros em que existe um melhor ou pior atendimento). Além disso, precisamos entender quando e como esse modelo poderia ser usado. Se a predição for feita com um mês de antecedência é suficiente? As ações tomadas estariam de acordo?

Coleta de dados

Reunir os dados para criar a modelagem pode ser simples, bastando executar uma *query SQL* em uma base de dados. Mas, dependendo do negócio e da necessidade, pode ser uma tarefa bastante trabalhosa. Pode ser necessário capturar dados de fontes diversas e, em alguns casos, fazer raspagem de dados (web scraping), ou seja, ter que criar um script que faça download de dados da internet de maneira automatizada, como por exemplo, baixar tweets para criar um modelo que analise fake news. Então, tendo entendido quais as principais variáveis que precisamos para a criação do modelo, é hora de conversar com a equipe responsável pelos dados, que pode ser a equipe de Engenharia de Dados ou a própria equipe de Produtos, dependendo da estruturação da empresa.

Reunir os dados para criar a modelagem pode ser simples, bastando executar uma *query SQL* em uma base de dados. Mas, dependendo do negócio e da necessidade, pode ser uma tarefa bastante trabalhosa. Pode ser necessário capturar dados de fontes diversas e, em alguns casos, fazer raspagem de dados (web scraping), ou seja, ter que criar um script que faça download de dados da internet de maneira automatizada, como por exemplo, baixar tweets para criar um modelo que analise fake news. Então, tendo entendido quais as principais variáveis que precisamos para a criação do modelo, é hora de conversar com a equipe responsável pelos dados, que pode ser a equipe de Engenharia de Dados ou a própria equipe de Produtos, dependendo da estruturação da empresa.

Treinamento

Chegou a etapa do ciclo que leva toda a fama! Aqui, já com os dados organizados, limpos e tratados, os separamos em três grupos: treino, teste e validação. O treino e a validação são usados durante o treinamento, sendo o treino para treinar o algoritmo em si e a validação para testar os parâmetros do algoritmo. Devemos guardar os dados que ficaram em teste para a etapa seguinte. Existem outras formas de fazer essa separação entre treino e teste, como, por exemplo, com validação cruzada, separando os dados em vários folds. Tendo feito essa última organização, é hora de você brilhar com os seus conhecimentos de aprendizado da máquina. O ideal é escolher algoritmos para testes que se encaixem com o seu

problema, começando com os mais simples e, somente caso necessário, passando para algoritmos mais complexos.

Validação

Agora, com o seu modelo treinado, está na hora de avaliar os resultados!

É importante ter os dados separados para teste para garantir que o seu modelo não está com overfitting, ou seja, que ele não tenha se super ajustado aos dados de treino. Por isso, testamos nos dados de teste e calculamos as métricas necessárias de acordo com o nosso problema. Se for um problema de classificação, como é o caso dos nossos dois projetos, as métricas normalmente são: curva ROC; área sob a curva ROC (AUC – do inglês “area under the curve”); acurácia; precisão; sensibilidade (ou recall). É importante notar qual métrica é mais relevante de acordo com o seu problema. Se estou tratando de uma classificação de um problema de saúde, a sensibilidade é mais importante, pois é melhor capturar a maior quantidade de doentes corretamente, mesmo que algumas pessoas saudáveis sejam incorretamente classificadas como doentes. Porém, eu também não posso ter uma precisão muito ruim, para que pessoas saudáveis não passem pelo stress de acharem que estão doentes. Por isso, é preciso definir um bom limiar (threshold) de probabilidade que dê o equilíbrio que estamos procurando.

Deploy

O deploy consiste em deixar o seu modelo disponível para que outros serviços possam acessá-lo. Dessa forma, ele poderá de fato ser utilizado dentro do ecossistema da empresa ou diretamente pelos clientes do seu serviço. O deploy normalmente consiste na criação de uma API com um endpoint específico para a predição, mas também pode ser feito de algumas outras formas, como uma comunicação por mensageria ou até mesmo uma predição por batch em script.

Monitoramento

Avaliar o seu modelo enquanto em produção é uma etapa ainda pouco valorizada. Os dados não são fixos, portanto uma mudança de comportamento pode alterar todos os resultados. Por isso, é importante monitorar o andamento e entender quando é hora de capturar novos dados e retreinar o seu modelo. Independentemente do projeto, essa etapa consiste em capturar os dados e as predições do modelo que

estão em produção, e fazer as mesmas análises de validação do modelo. Se percebemos que o modelo está caindo em relação às métricas definidas — por exemplo, acurácia — e está muito pior do que com os dados utilizados na validação, é hora de pegar mais dados e retreinar o modelo.

5.1 Requisitos

Para HIGOR (2013), antigamente dizia-se que requisitos eram sinônimos de funções, ou seja, tudo que o software deveria fazer funcionalmente. No entanto, atualmente assumiu-se que requisitos de software é muito mais do que apenas funções. Requisitos são, além de funções, objetivos, propriedades, restrições que o sistema deve possuir para satisfazer contratos, padrões ou especificações de acordo com o(s) usuário(s). De forma mais geral um requisito é uma condição necessária para satisfazer um objetivo. Portanto, um requisito é um aspecto que o sistema proposto deve fazer ou uma restrição no desenvolvimento do sistema. Vale ressaltar que em ambos os casos devemos sempre contribuir para resolver os problemas do cliente e não o que o programador ou um arquiteto deseja. Dessa forma, o conjunto dos requisitos como um todo representa um acordo negociado entre todas as partes interessadas no sistema. Isso também não significa que o programador, arquiteto ou um analista bem entendido no assunto de tecnologia não possam contribuir com sugestões e propostas que levem em conta o desejo do cliente. Além disso, ainda temos um documento de requisitos que é uma coleção dos requisitos. Por fim, os requisitos possuem alguns objetivos centrais como estabelecer e manter uma concordância com os clientes e outros envolvidos sobre o que o sistema deve fazer, deve oferecer aos desenvolvedores, projetistas e testadores do sistema uma compreensão melhor dos requisitos do sistema, definir fronteiras do sistema definindo o que deve ser incluído e o que não deve fazer parte do sistema, fornece uma base para estimar o custo e o tempo de desenvolvimento do sistema e por fim definir uma interface de usuário para o sistema.

5.1.1 Requisitos funcionais

Os requisitos funcionais são aqueles que visam atingir a solução dos problemas do usuário. Desse modo, eles trabalham diretamente no objetivo para o qual uma solução foi escrita. Na

engenharia de software, um requisito funcional visa atender as necessidades dos usuários, materializando estes conceitos em forma de funções no novo sistema. (DESENVOLVEDOR, 2024).

REQUISITOS FUNCIONAIS		
RF001	Cadastramento de usuário	Permite que o usuário faça o preenchimento dos seus dados para realizar o cadastramento.
RF002	Login de usuário	Permite que o login do usuário seja feito a partir do CPF e senha.
RF003	Pesquisa de produto	Permite o usuário procurar o produto por nome, ou outras especificações e também filtrar o conteúdo.
RF004	Catálogo de produto	Exibir um catálogo dos produtos separados por sessões.
RF005	Acesso administrador	Permite que o administrador insira, altere, e exclua informações de usuários e agendamentos.
RF007	Carrinho de compras	Adicionar e remover itens; mostrar o valor da soma de todos os produtos do carrinho;
RF008	Chekout e Pagamento	Checkout simplificado em várias etapas;
RF009	Gestão de Pedidos	O usuário conseguira ver o andamento do pedido.
RF010	Avaliação e Comentários	O cliente poderá comentar e avaliar os produtos, mas com moderação.

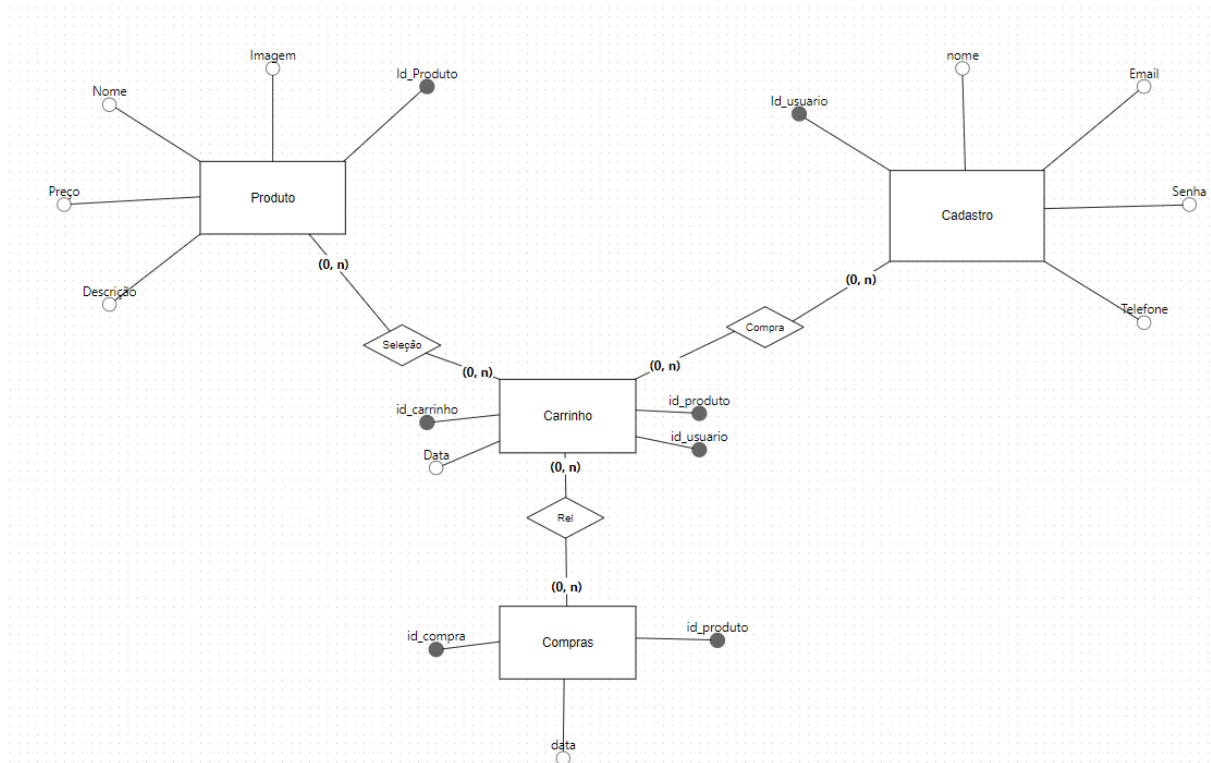
Fonte: FREITAS, P. J. C., (2024).

5.1.2 Requisitos não funcionais

Segundo o site TRTPR (2024) Requisitos não-funcionais são os requisitos relacionados ao uso da aplicação em termos de desempenho, usabilidade, confiabilidade, segurança, disponibilidade, manutenibilidade e tecnologias envolvidas. Não é preciso o cliente dizer sobre eles, pois eles são características mínimas de um software de qualidade, ficando a cargo do desenvolvedor optar por atender esses requisitos ou não.

REQUISITOS NÃO FUNCIONAIS		
RNF001	Velocidade	Que o usuário não precise esperar mais de 4 segundos para obter retorno.
RNF002	Facilidade de uso	O treinamento para utilização deve ser feito no máximo em 1 dia.
RNF003	Confiabilidade	Terá administrador tomando conta para caso ocorra falhas no sistema, e seja corrigido o mais breve possível.

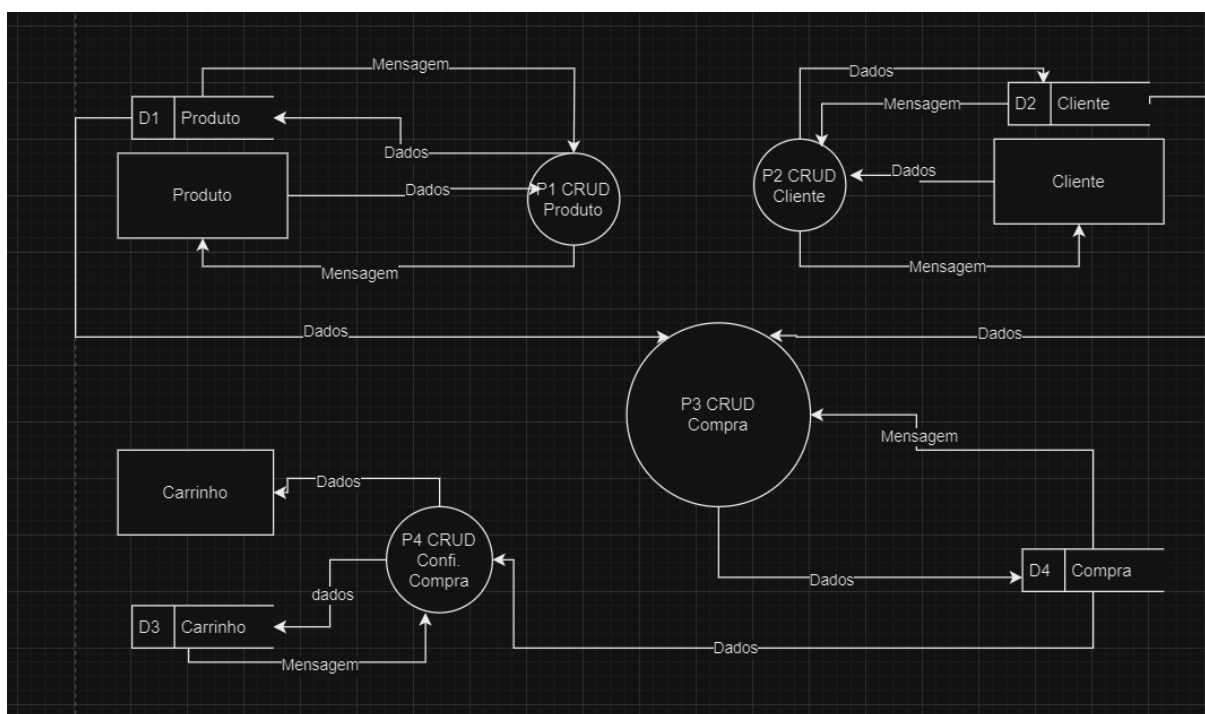
MODELO CONCEITUAL



Fonte: FREITAS, P. J. C., 2024

5.3 Diagrama de Fluxo de dados

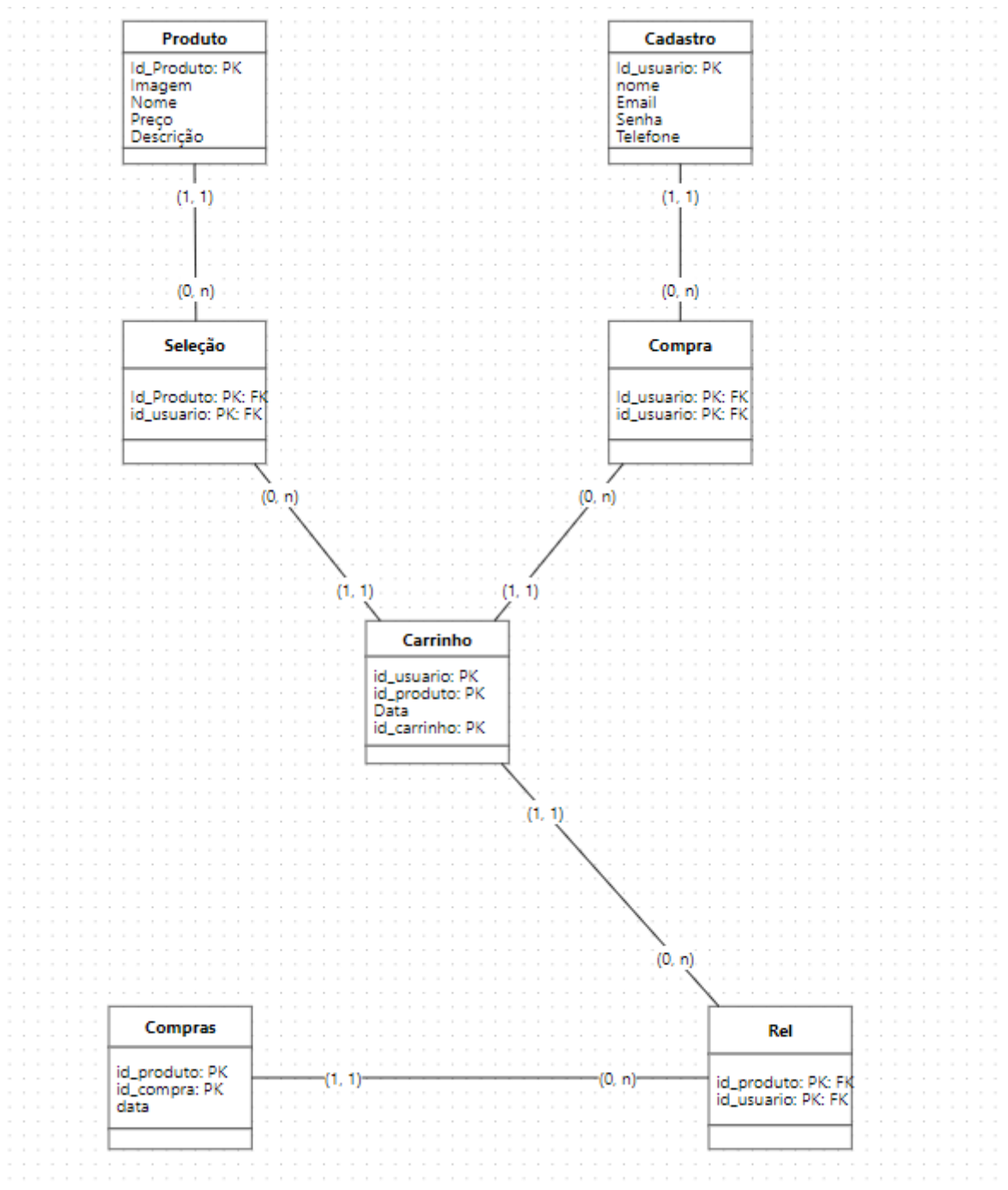
Para Khusid (2024), um diagrama de fluxo de dados mapeia a sequência de informações, atores e passos dentro de um processo ou sistema. Ele usa um conjunto de símbolos e formas padronizadas representando as diferentes etapas e pessoas que são necessárias para que um processo seja realizado corretamente. O diagrama pode ser tão simples ou tão complexo quanto o sistema que ele representa, mas a maneira mais fácil é usar uma ferramenta para fazer Diagramas de Fluxo de Dados Online. Normalmente, os diagramas de fluxo de dados são usados para representar visualmente o fluxo de dados de um sistema de uma empresa. Como mostrado no exemplo do diagrama de fluxo de dados abaixo, eles ilustram como os dados são processados dentro de um sistema usando um conjunto de entradas (inputs) e saídas (outputs).



Fonte: FREITAS, P. J. C., 2024

5.4 Diagrama de Entidade e relacionamento

Conforme diz KHUSID (2024) um diagrama de entidade e relacionamento (também conhecido como diagrama ER ou simplesmente DER) mostra como as entidades (pessoas, objetos e conceitos) interagem. Estes modelos de dados conceituais ajudam desenvolvedores e designers a visualizar as relações entre os elementos-chave do software. Criado pelo cientista da computação Peter Chen nos anos 70, os diagramas ERs são comumente usadas por empresas ao projetar e analisar bancos de dados. Entretanto, os DERs também podem mapear relações entre outros elementos empresariais, tais como funções (como a relação de funções de um gerente de produto com um desenvolvedor), objetos empresariais tangíveis (como um produto ou serviço), e objetos empresariais intangíveis (como um backlog de produtos). O diagrama em si é um tipo de fluxograma. É um modelo lógico que mostra como os dados fluem de uma entidade para a outra. Com este formato fácil de seguir, os desenvolvedores de software e designers podem visualizar claramente a estrutura de um sistema.



5.5 Dicionário de Dados

Um banco de dados é para ORACLE (2024), uma coleção organizada de informações - ou dados - estruturadas, normalmente armazenadas eletronicamente em um sistema de computador. Um banco de dados é geralmente controlado por um sistema de gerenciamento de banco de dados (DBMS). Juntos, os dados e o DBMS, juntamente com os aplicativos associados a eles, são chamados de sistema de banco de dados, geralmente abreviados para apenas banco de dados. Os dados nos tipos mais comuns de bancos de dados em operação atualmente são modelados em linhas e colunas em uma série de tabelas para tornar o processamento e a consulta de dados eficientes. Os dados podem ser facilmente acessados, gerenciados, modificados, atualizados, controlados e organizados. A maioria dos bancos de dados usa a linguagem de consulta estruturada (SQL) para escrever e consultar dados.

cadastros_usuarios

Coluna	Tipo
id (Primária)	int(11)
user_id	int(11)
full_name	varchar(100)
phone	varchar(15)
email	varchar(100)
password	varchar(255)
created_at	timestamp

Índices

Nome da chave	Tipo
PRIMARY	BTREE
user_id	BTREE

users

Coluna	Tipo
id (<i>Primária</i>)	int(11)
username	varchar(50)
password	varchar(255)
email	varchar(100)
created_at	timestamp

Índices

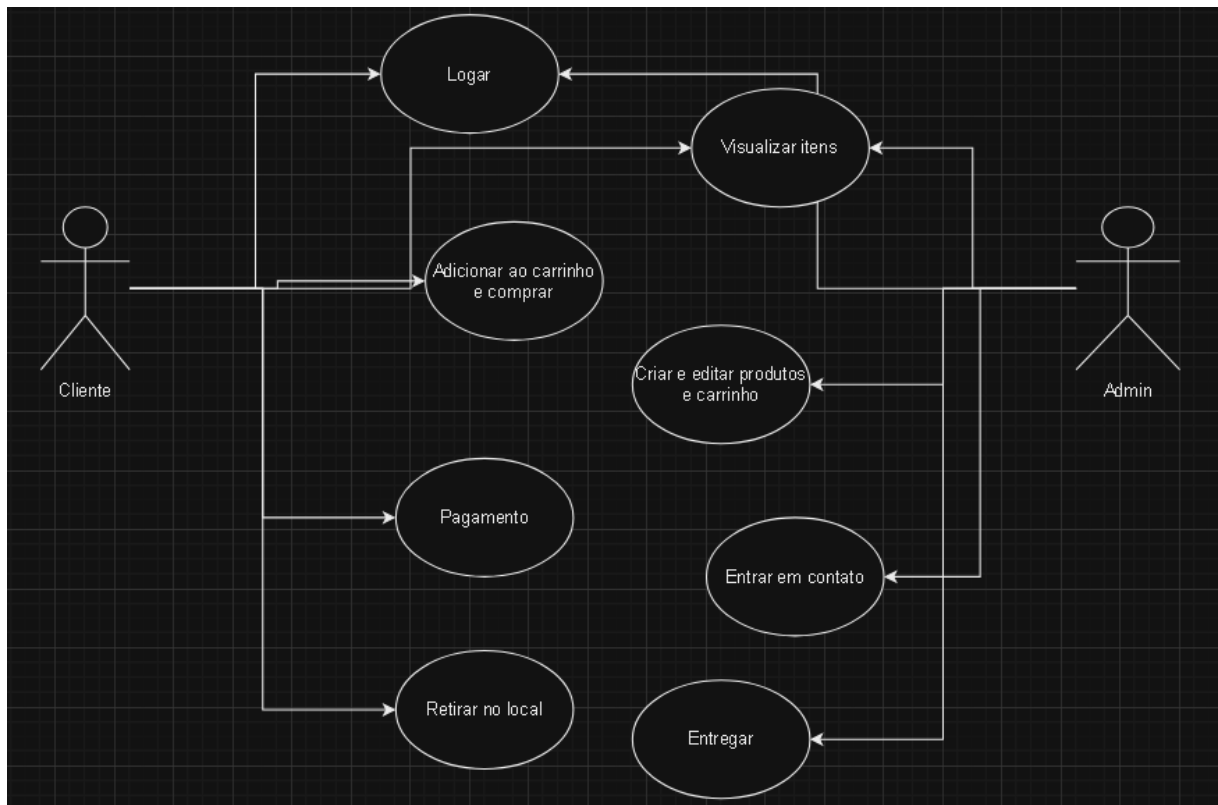
Nome da chave	Tipo
PRIMARY	BTREE

Fonte: FREITAS, P. J. C., 2024

5.6 Diagrama de Caso de Uso

Para LEANDRO (2012) esse diagrama documenta o que o sistema faz do ponto de vista do usuário. Em outras palavras, ele descreve as principais funcionalidades do sistema e a interação dessas funcionalidades com os usuários do mesmo sistema. Nesse diagrama não nos aprofundamos em detalhes técnicos que dizem como o sistema faz.

Este artefato é comumente derivado da especificação de requisitos, que por sua vez não faz parte da UML. Pode ser utilizado também para criar o documento de requisitos.



Fonte: FREITAS, P. J. C., 2024

Visualizar Produtos:

O cliente acessa a página inicial e visualiza os produtos disponíveis, com imagens, descrição, preço e ingredientes principais.

Adicionar ao Carrinho:

O cliente seleciona um produto e o adiciona ao carrinho de compras.

Opcional: Adiciona observações como preferências de sabor, cor, etc.

Finalizar Pedido:

O cliente revisa os itens no carrinho e preenche o formulário com informações de contato.

Pagamento.

Envia o pedido, que é enviado por e-mail ou WhatsApp para o dono da loja.

Gerenciar Produtos (Admin):

O administrador faz login no painel de controle e adiciona/edita/remove produtos, definindo preço, descrição e estoque.

Gerenciar Pedidos (Admin):

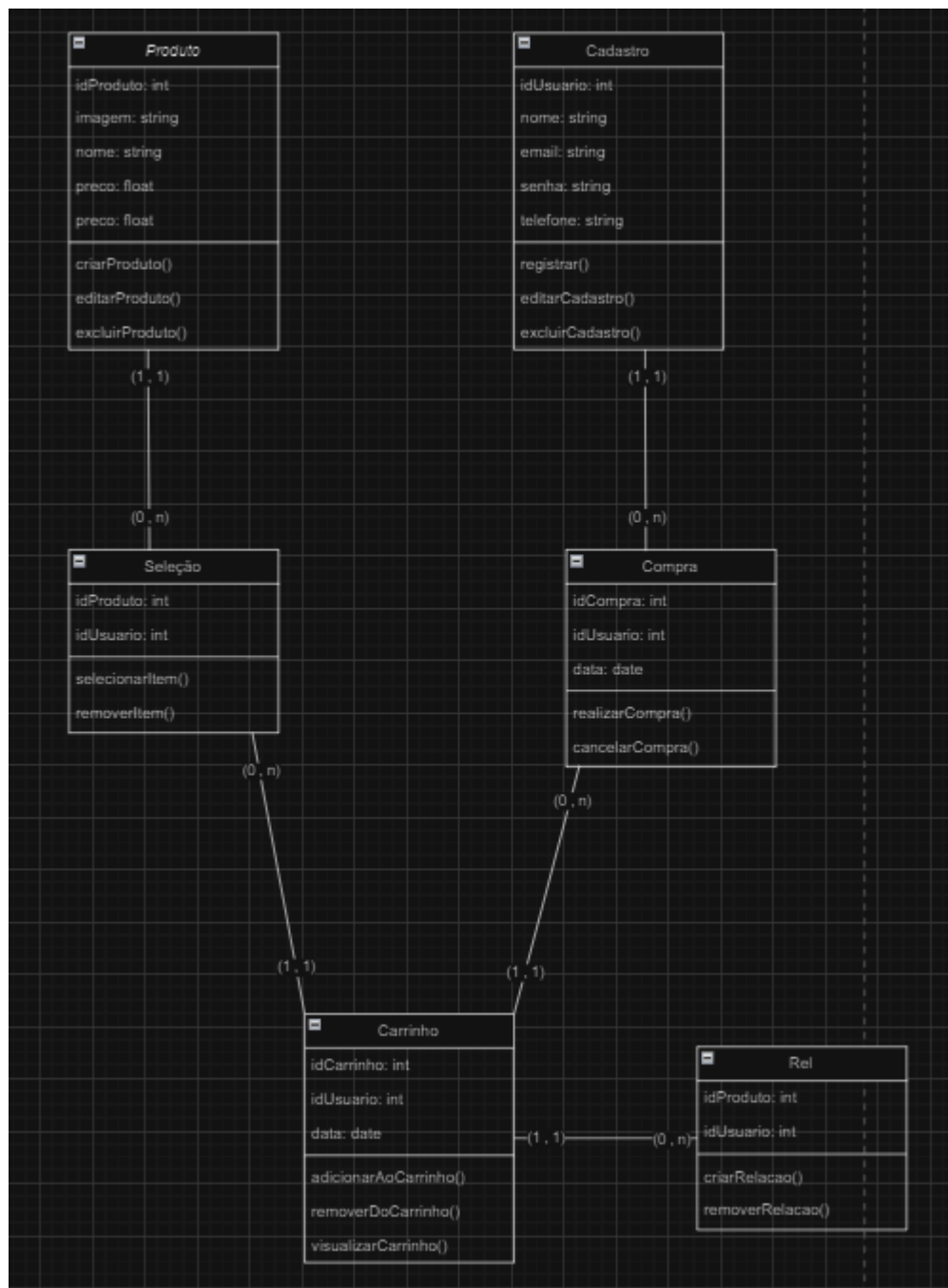
O administrador visualiza os pedidos realizados, confirma e informa o cliente o pagamento.

Consultar Informações de Contato:

O cliente acessa a página de "Contato" para entrar em contato direto com a confeitaria via e-mail.

5.7 Diagrama de Classe

Segundo Douglas (2016) na programação, um diagrama de classes é uma representação da estrutura e relações das classes que servem de modelo para objetos. Podemos afirmar de maneira mais simples que seria um conjunto de objetos com as mesmas características, assim saberemos identificar objetos e agrupá-los, de forma a encontrar suas respectivas classes. Na Unified Modeling Language (UML) em diagrama de classe, uma classe é representada por um retângulo com três divisões, são elas: O nome da classe, seus atributos e por fim os métodos.

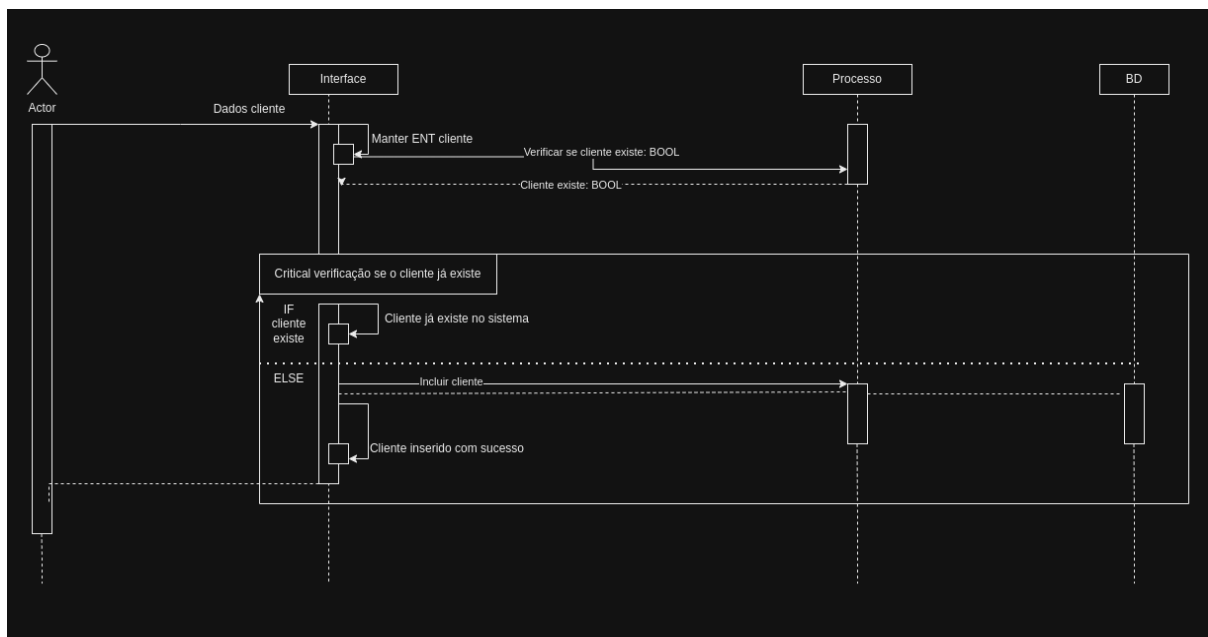


Fonte: FREITAS, P. J. C., 2024

5.8 Diagrama de Sequência

Para Paulo (2009) o diagrama de sequência preocupa-se com a ordem temporal em que as mensagens são trocadas entre os objetos envolvidos em determinado processo, ou seja, quais condições devem ser satisfeitas e quais métodos devem ser disparados entre os objetos envolvidos e em que ordem durante um processo. Dessa forma, determinar a ordem em que os eventos ocorrem, as mensagens que são enviadas, os métodos que são chamados e como os objetos interagem entre si dentro de um determinado processo é o principal objetivo deste diagrama.

O diagrama de sequência baseia-se no diagrama de casos de uso. É importante deixarmos claro que o fato de existir apenas um diagrama de casos de uso não implica na existência de apenas um único diagrama de sequência. Segundo Guedes em seu livro “UML – Uma Abordagem Prática”, normalmente existem diversos diagramas de sequência em um projeto, um para cada processo específico do sistema

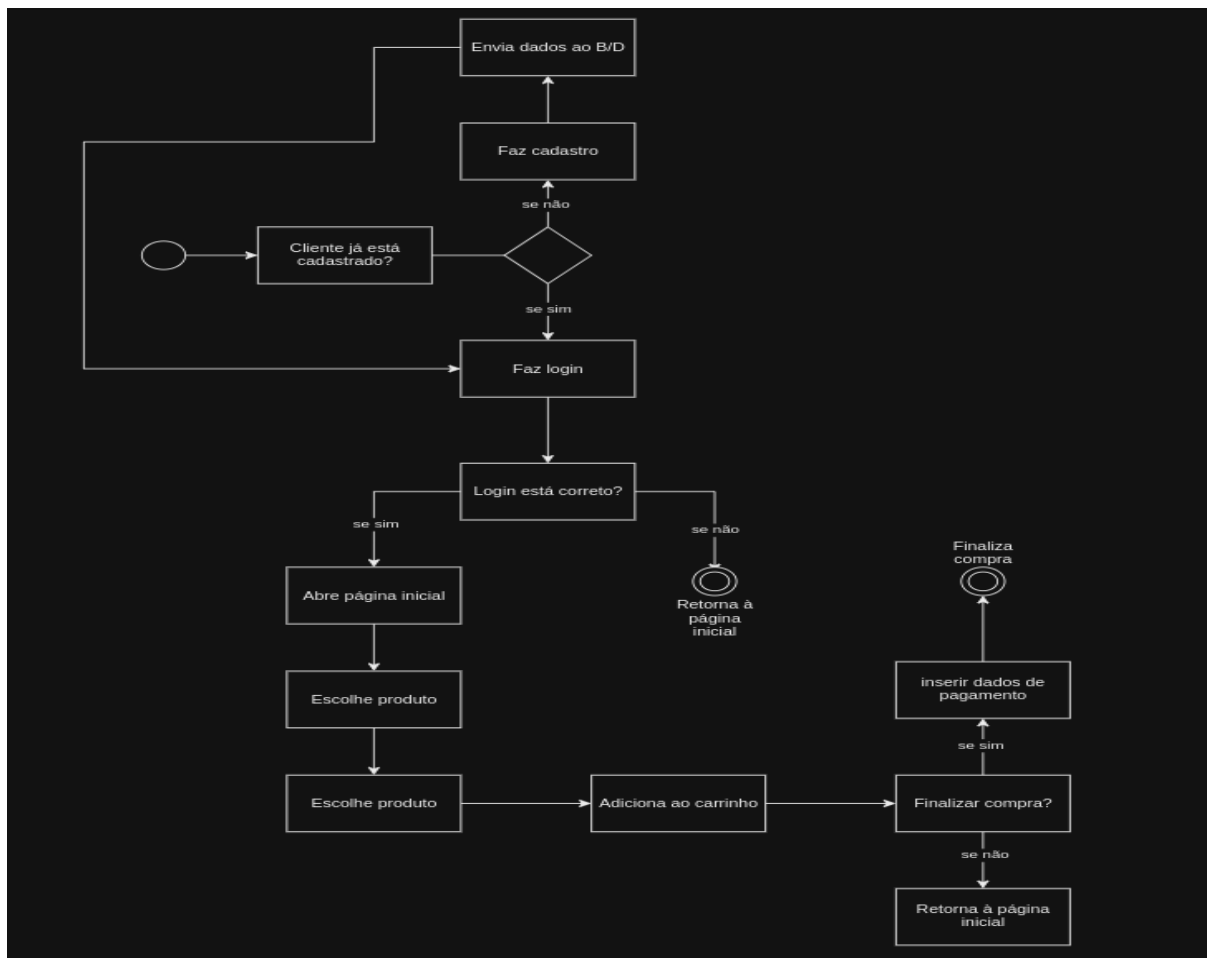


Fonte: FREITAS, P. J. C., 2024

5.9 Diagrama de Atividade

Segundo a citação de Geraldo (2010) aparentados com os diagramas de transição de estados, diagramas de atividades compartilham com aqueles uma parte significativa da notação. Estados de ação e estados de atividade são representados por um retângulo de cantos arredondados. De forma coerente com os diagramas de estados, ações são atômicas e atividades são conjuntos de ações e/ou atividades encadeadas.

Um diagrama de atividades pode constituir uma grande atividade (não atômica) formada por diversas ações (atômicas) e/ou sub-atividades. Entre duas ações e/ou atividades, uma seta simboliza simplesmente o término de uma atividade e o início da outra. Em princípio não é necessário rotular esse elemento, mas uma condição de guarda poderá estar eventualmente associada a ele. Os símbolos de início e final são idênticos àqueles utilizados em diagramas de estados.



Fonte: FREITAS, P. J. C., 2024

6 TELAS



Seja bem vindo a nossa loja online de quadros!

Loja online especializada em quadros para pintura e decoração. Além disso, oferece frete rápido, pagamento facilitado e promoções exclusivas. Ideal para quem busca praticidade na sua casa, e expressão artística para a vida.



Quadro Van Gogh



Quadro Paisagem




Quadro: O Susto



Sobre Nós



Somos apaixonados por obras de arte e decorações verdadeiras. Com uma equipe livre, oferecemos produtos e serviços de alta qualidade, garantindo a satisfação dos nossos clientes. Nosso objetivo é tornar a arte acessível, entregando soluções baratas e boas para as necessidades de cada cliente



Cadastro de usuário

Usuário:

Usuário:

Senha:

Usuário:

Entrar

Cadastrar



Login

Usuário:

Senha:

Entrar

Cadastrar



The image shows a mobile application screen for client registration. On the left is a vertical sidebar with icons for home, information, a box, a person, and a shopping cart. The main area has a black background with white text and rounded input fields. The title 'Cadastro de cliente' is at the top. Below it are labels and input fields for 'Nome:', 'E-mail:', 'Telefone:', and 'Data de Nascimento:'. At the bottom is a grey button labeled 'Cadastrar cliente'.

Cadastro de cliente

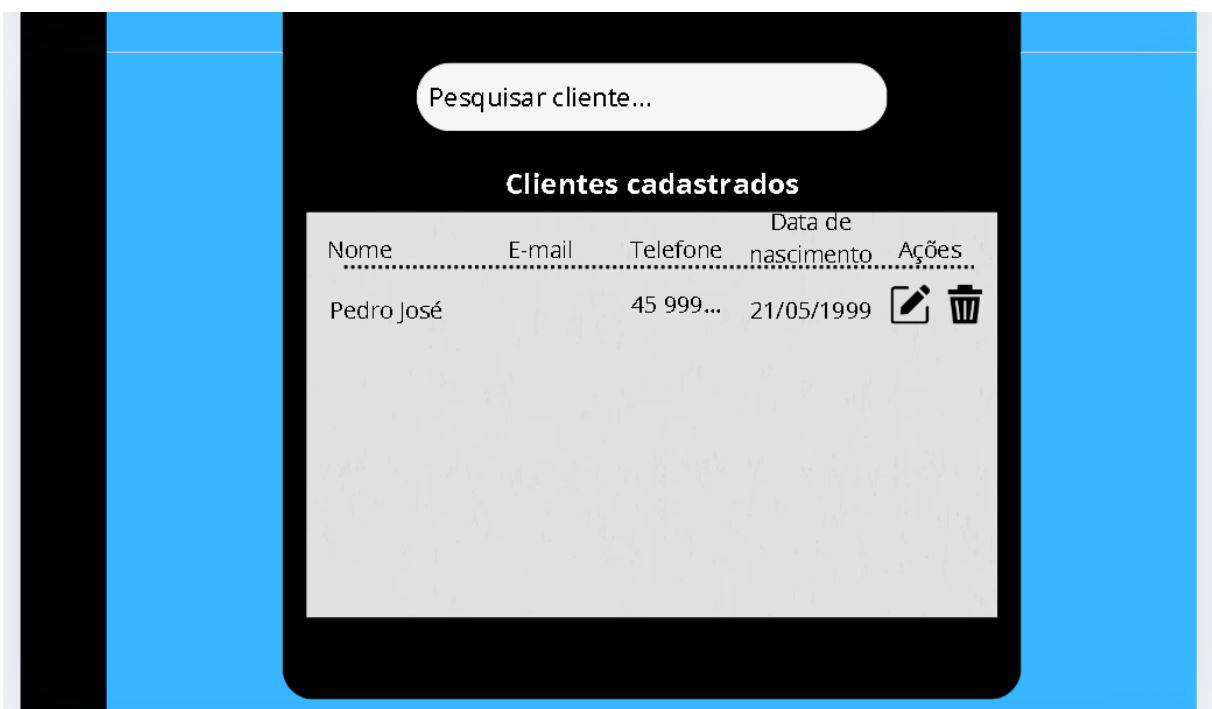
Nome:

E-mail:

Telefone:

Data de Nascimento:



Cadastrar cliente




The image shows a mobile application screen for viewing registered clients. At the top is a search bar with the placeholder text 'Pesquisar cliente...'. Below it is the title 'Clientes cadastrados'. Under the title is a table with columns for 'Nome', 'E-mail', 'Telefone', 'Data de nascimento', and 'Ações'. The first row of data shows 'Pedro José', '45 999...', and '21/05/1999'. The 'Ações' column contains edit and delete icons. The table is overlaid on a light grey background.

Pesquisar cliente...

Clientes cadastrados

Nome	E-mail	Telefone	Data de nascimento	Ações
Pedro José		45 999...	21/05/1999	 



Cadastro de pedido

Número do pedido:

Data do pedido:



Cliente

Itens do pedido:

Produto	Quantidade	Valor item	Ações
quadro simples	1	20,00	Ad. item

Cadastrar pedido

Pedidos cadastrados

Pedido	Cliente	Data	Valor Total	Ações
nº2	Pedro José	21/05/2024	R\$20,00	 

7 CONCLUSÃO

Este projeto teve como objetivo principal desenvolver um sistema eficiente para gerenciar produtos e compras, integrando funcionalidades como cadastro de usuários, seleção de itens, carrinho de compras e finalização de pedidos. A ideia surgiu da necessidade de atender demandas reais, criando uma plataforma prática que auxilie pequenos negócios, promovendo organização e acessibilidade.

Durante o desenvolvimento, abordamos desde a modelagem do banco de dados até a implementação prática das funcionalidades. Aplicamos conceitos de normalização, relacionamentos entre tabelas e princípios de programação orientada a objetos. Cada etapa trouxe desafios importantes, como criar tabelas intermediárias e estabelecer regras de negócio para manter a integridade dos dados. Esses desafios resultaram em aprendizados valiosos, enriquecendo minha trajetória como programador.

Os resultados confirmaram a viabilidade do projeto e demonstraram que, com dedicação e planejamento, é possível criar uma solução eficaz para atender às necessidades iniciais. O sistema desenvolvido proporciona não apenas uma gestão simplificada de produtos e usuários, mas também abre possibilidades para melhorias futuras, como integrar métodos de pagamento e realizar análises de vendas.

As implicações deste projeto são significativas tanto para o campo acadêmico quanto para a prática. Ele demonstra como tecnologias acessíveis e bem planejadas podem digitalizar processos e melhorar a experiência dos usuários. Além disso, o trabalho reforça a importância de aliar teoria e prática para solucionar problemas reais. No futuro, pretendo expandir as funcionalidades deste sistema, incluindo relatórios avançados de vendas e integração com redes sociais. Esses são caminhos promissores que podem fortalecer o impacto positivo da solução.

Agradeço imensamente a todos que contribuíram para o sucesso deste projeto: professores, colegas e familiares que ofereceram apoio e incentivo em todas as etapas. Este trabalho não teria sido possível sem dedicação e esforço coletivo. Concluo com a certeza de que este projeto é apenas o início de uma jornada. Ele reflete não apenas o conhecimento adquirido, mas também a vontade de continuar aprendendo, inovando e criando. Espero que esta iniciativa inspire outros a desenvolver soluções criativas para desafios do cotidiano.

8 REFERÊNCIAS

(GEMINI), Openia. Na era digital. 2021. Disponível em: <https://mindthegraph.com/blog/pt/chatgpt-citacoes/>. Acesso em: 08 ago. 2024.
CIRNE, Fernando. MySQL: guia completo de como utilizar o sistema. 2024. disponível em: <https://www.locaweb.com.br/blog/temas/codigo-aberto/mysql-guia-completo-de-como-utilizar-o-sistema>. Acesso em: 20 set. 2024.

CRISTO, Adriel. O que é o PHP? 2024. Disponível em: https://www.php.net/manual/pt_BR/intro-what-is.php. Acesso em: 20 set. 2024. LOCAWEB,

DOUGLAS. Orientações básicas na elaboração de um diagrama de classes: o que são os diagramas de classe. O que são os Diagramas de Classe. 2016. Disponível em: <https://www.devmedia.com.br/orientacoes-basicas-na-elaboracao-de-um-diagrama-de-classes/37224>. Acesso em: 25 nov. 2024.

EICH, Brendan. HTML: Linguagem de Marcação de Hipertexto. 2024. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/HTML>. Acesso em: 20 set. 2024.

EICH, Brendan. O que é CSS? . 2024. Disponível em: https://developer.mozilla.org/pt-BR/docs/Learn/CSS/First_steps/What_is_CSS. Acesso em: 20 set. 2024.

EICH, Brendan. O que é JavaScript?. 2024. Disponível em: https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/First_steps/What_is_JavaScript. Acesso em: 20 set. 2024.

GERALDO. Diagramas de Atividades - Engenharia de Software 31. Definições básicas e notação. 2010. Disponível em: <https://www.devmedia.com.br/diagramas-de-atividades-engenharia-de-software-31/18744>. Acesso em: 25 nov. 2024.

HIGOR. Introdução a Requisitos de Software: requisitos de software. Requisitos de Software. 2013. Disponível em: <https://www.devmedia.com.br/introducao-a-requisitos-de-software/29580>. Acesso em: 21 out. 2024

KHUSID, Andrey. Diagrama de Entidade e Relacionamento: o que é um diagrama de entidade e relacionamento (er)?. O que é um diagrama de entidade e relacionamento (ER)?. 2024. Disponível em: <https://miro.com/pt/diagrama/o-que-e-diagrama-entidade-relacionamento/>. Acesso em: 25 out. 2024.

KHUSID, Andrey. Diagrama de fluxo de dados: o que é um diagrama de fluxo de dados?. O que é um diagrama de fluxo de dados?. 2024. Disponível em: <https://miro.com/pt/diagrama/o-que-e-diagrama-fluxo-dados/>. Acesso em: 25 out. 2024.

LEANDRO. O que é UML e Diagramas de Caso de Uso: Introdução Prática à UML:

diagrama de casos de uso. Diagrama de Casos de Uso. 2012. Disponível em: <https://www.devmedia.com.br/o-que-e-uml-e-diagramas-de-caso-de-uso-introducao-pratica-a-uml/23408>. Acesso em: 25 nov. 2024.

MESQUITA, Aline. Documentação de projetos: o que é, por que e como fazer? 2024. Disponível em: <https://www.oitchau.com.br/blog/documentacao-de->

OLIVEIRA, Jéssica dos Santos de. O ciclo de vida de um projeto de dados. 2022. Disponível em: <https://www.programaria.org/o-ciclo-de-vida-de-um-projeto-de-dados/>. Acesso em: 07 out. 2024.

PAULO. Artigo SQL Magazine 64 - Utilizando UML. 2009. Disponível em: <https://www.devmedia.com.br/artigo-sql-magazine-64-utilizando-uml/12665>. Acesso em: 25 nov. 2024.

PRONEC. Qual seria a diferença de um artista plástico de um pintor? 2019. Disponível em: <https://pronec.org/certificado/qual-seria-a-diferenca-de-um-artista-plastico-de-um-pintor/>. Acesso em: 04 dez. 2024.

TRTPR. Conceito: Requisitos Não-Funcionais: definição. Definição. 2024. Disponível em: https://www.trt9.jus.br/pds/pdstrt9/guidances/concepts/supporting_requirements_B2C4D610.html. Acesso em: 23 out. 2024.