**UNIVERSIDADE VIRTUAL DO ESTADO DE SÃO PAULO**

José Aparecido Gomes da Silva, 2215539

Matheus Marcelo Pereira Menezes, 2106424

**IMPLEMENTAÇÃO DE UM PROTÓTIPO DE SISTEMA DE GESTÃO DE VENDAS PARA PEQUENAS EMPRESAS**

|  |
| --- |
| **Vídeo de apresentação do Projeto Integrador**  https://www.youtube. |

Fartura e Manduri - SP

2024

**UNIVERSIDADE VIRTUAL DO ESTADO DE SÃO PAULO**

**IMPLEMENTAÇÃO DE UM PROTÓTIPO DE SISTEMA DE GESTÃO DE VENDAS PARA PEQUENAS EMPRESAS**

Relatório Técnico-Científico apresentado na disciplina de Projeto Integrador para o curso de Bacharel em Tecnologia da informação da Universidade Virtual do Estado de São Paulo (UNIVESP).

Fartura e Manduri - SP

2024

GOMES DA SILVA, José Aparecido; PEREIRA MENEZES, Matheus Marcelo. **Implementação de um protótipo de sistema de gestão de vendas para pequenas empresas.** 21f. Relatório Técnico-Científico. Tecnologia da informação – **Universidade Virtual do Estado de São Paulo**. Tutor: Suelen Santos Alves. Polo Fartura/Manduri 2024.

**RESUMO**

Este trabalho apresenta a implementação de um protótipo de Sistema de Gestão de Vendas para pequenas empresas, destacando a relevância da tecnologia da informação na otimização de processos comerciais. O objetivo foi desenvolver uma solução que melhorasse a eficiência operacional e a produtividade de uma pequena empresa em Fartura, São Paulo. A metodologia adotada envolveu a análise das necessidades da empresa, o desenvolvimento do diagrama conceitual do banco de dados e a implementação das classes (entidades) com o *framework* React. Incluiu a criação de um painel administrativo para facilitar o gerenciamento das principais operações comerciais e a realização de testes para identificar e corrigir erros. Os resultados indicam que o sistema traz o potencial de vários benefícios significativos, como a automatização de processos, redução de custos operacionais e aumento da produtividade. O protótipo mostra que é possível a centralização das operações e a integração de dados, facilitando a gestão e a tomada de decisões estratégicas. Conclui-se que a implementação do protótipo de Sistema de Gestão de Vendas é uma iniciativa útil para pequenas empresas. A tecnologia da informação mostrou ser uma ferramenta essencial para melhorar a eficiência e competitividade da empresa. Os resultados podem servir como base para futuras inovações para a empresa escolhida.

**PALAVRAS-CHAVE:** Pequenas Empresas; Gestão de Vendas; Eficiência Operacional; Tecnologia da Informação.

**LISTA DE ILUSTRAÇÕES**

**SUMÁRIO**

[**1 Introdução 6**](#_30j0zll)

[**2 Desenvolvimento 8**](#_lnxbz9)

[2.1 Objetivos 8](#_35nkun2)

[2.2 Justificativa e delimitação do problema 8](#_1ksv4uv)

[2.3 Fundamentação teórica 9](#_44sinio)

[2.3.1 Benefícios dos Sistemas de Gestão de Vendas para Pequenas Empresas: 10](#_2jxsxqh)

[2.3.2 Modelo conceitual de banco de dados 11](#_z337ya)

[2.3.3 *Framework* React 12](#_3j2qqm3)

[2.3.4 Git e github 14](#_1y810tw)

[2.4 Aplicação das disciplinas estudadas no projeto integrador 15](#_4i7ojhp)

[2.4.1 Algoritmos e Programação de Computadores I e II 15](#_2xcytpi)

[2.4.2 Formação Profissional em Computação 16](#_1ci93xb)

[2.4.3 Infraestrutura para Sistemas de Software 16](#_3whwml4)

[2.4.4 Banco de Dados 16](#_2bn6wsx)

[2.4.5 Estrutura de Dados 17](#_qsh70q)

[2.4.6 Fundamentos da Internet e Web 17](#_3as4poj)

[2.4.7 Programação Orientada a Objetos 17](#_1pxezwc)

[2.4.8 Gestão da Inovação e Desenvolvimento de Produtos 18](#_49x2ik5)

[2.4.9 Integração das Disciplinas 18](#_2p2csry)

[2.5 Metodologia 18](#_147n2zr)

[2.5.1 Ouvir e interpretar o contexto 18](#_3o7alnk)

[2.5.2 Prototipação 19](#_23ckvvd)

[2.5.3 Implementação e testes 20](#_ihv636)

[**3 Resultados: solução final 21**](#_32hioqz)

[3.1 Definição dos modelos (classes) no 22](#_1hmsyys)

[3.2 páginas do sistema 24](#_41mghml)

[3.3 Gerenciamento de versões com GitHub 28](#_2grqrue)

[**4 Considerações Finais 30**](#_vx1227)

[**Referências 32**](#_3fwokq0)

**1 Introdução**

No cenário empresarial contemporâneo, a eficiência e a otimização dos processos são fundamentais para a sobrevivência e o crescimento sustentável das organizações. Pequenas empresas, em particular, enfrentam desafios significativos, muitas vezes devido à falta de recursos e tecnologias adequadas para gerenciar suas operações de forma eficaz. Um dos aspectos cruciais para o sucesso de qualquer empreendimento é a gestão das vendas, que influencia diretamente a lucratividade e a sustentabilidade do negócio.

As pequenas empresas representam uma parte significativa da economia, contribuindo para a geração de empregos e o crescimento econômico local. No entanto, muitas vezes enfrentam desafios únicos, como recursos limitados, processos manuais e falta de acesso a tecnologias avançadas. Nesse sentido, a implementação de soluções tecnológicas adequadas pode ser um diferencial crucial para a sobrevivência e o sucesso dessas empresas no mercado atual.

A adoção de tecnologia pode trazer uma série de benefícios para as pequenas empresas, incluindo a automação de processos, a melhoria da eficiência operacional, a redução de custos, o aumento da produtividade e a expansão do alcance de mercado. Além disso, a tecnologia pode oferecer percepções valiosas por meio da análise de dados, permitindo uma tomada de decisão menos subjetiva e mais estratégica.

Considerando este contexto, este trabalho propõe abordar a implementação de um Sistema de Gestão de Vendas para Pequenas Empresas, destacando a importância da tecnologia da informação como uma importante ferramenta para resolver os desafios enfrentados por essas organizações.

A falta de um sistema integrado de vendas é um problema recorrente para muitas pequenas empresas, limitando sua capacidade de competir no mercado e atender às demandas dos clientes de forma eficiente. Muitas vezes, essas empresas dependem de processos manuais e desatualizados, resultando em erros, ineficiências e perda de oportunidades de negócio. Nesse contexto, o desenvolvimento e a implementação de um software de vendas eficiente tornam-se imperativos, visando não apenas otimizar o processo de vendas, mas também melhorar a gestão de estoque, pedidos e relacionamento com os clientes.

A tecnologia da informação desempenha um papel crucial na resolução desses problemas, oferecendo soluções inovadoras e personalizadas para as necessidades específicas de cada empresa. Sistemas de gestão de vendas automatizados e integrados permitem uma visão abrangente das operações comerciais, desde o acompanhamento do ciclo de vendas até a análise de dados para tomada de decisões estratégicas. Além disso, esses sistemas facilitam a comunicação e a colaboração entre os diferentes colaboradores da empresa, promovendo uma maior eficiência e produtividade.

Ao adotar tecnologias de informação para a gestão de vendas, as pequenas empresas podem alcançar diversos benefícios, como a redução de custos operacionais, o aumento da precisão e rapidez no processamento de pedidos, a melhoria da experiência do cliente e a identificação de oportunidades de mercado. Além disso, a tecnologia pode oferecer percepções valiosas por meio da análise de dados, permitindo uma tomada de decisão menos subjetiva e mais estratégica. Nesse sentido, a computação em nuvem tem sido uma tecnologia transformadora, permitindo que empresas de todos os portes acessem ferramentas e recursos avançados de forma escalável e com custos reduzidos. Para pequenas empresas, que frequentemente enfrentam limitações orçamentárias e operacionais, a computação em nuvem oferece uma alternativa eficiente e acessível para gerenciar e otimizar suas operações comerciais. Segundo Armstrong (2021, p. 35), 'a nuvem permite que pequenas empresas adotem sistemas de gestão e análise de dados sem a necessidade de investir em infraestrutura física local, facilitando o acesso a tecnologias antes restritas a grandes corporações'.

Dessa forma, a implementação de um Sistema de Gestão de Vendas baseado em nuvem pode trazer vantagens importantes para pequenas empresas, como a mobilidade e o acesso remoto, permitindo que gestores acompanhem as operações em tempo real, de qualquer lugar e a qualquer momento. Além disso, a computação em nuvem facilita a integração a implementação de sistemas de vendas integrados permite que as empresas estejam mais bem preparadas para enfrentar os desafios do ambiente competitivo atual e se adaptar rapidamente às mudanças nas demandas do mercado, possibilitando uma visão abrangente e consolidada das atividades de vendas, estoque e relacionamento com clientes

Neste contexto, este projeto integrador se propõe a explorar as vantagens e desafios da implementação de um sistema de gestão de vendas para uma pequena empresa, destacando o papel crucial da tecnologia da informação na transformação e no aprimoramento dos processos comerciais.

**2 Desenvolvimento**

2.1 Objetivos

O objetivo deste projeto é desenvolver e implementar um protótipo de sistema de gestão de vendas para uma pequena empresa em Fartura, São Paulo, com o intuito de melhorar os processos comerciais e promover a inovação tecnológica na região. Os objetivos específicos são explorar as necessidades das empresas, descrever seus processos de vendas, analisar os impactos da falta de um sistema eficiente, e explicar como a adoção do protótipo desenvolvido pode beneficiar pequenas empresas na mesma situação.

2.2 Justificativa e delimitação do problema

O desenvolvimento e implementação do protótipo proposto nesse projeto integrador surgiu da necessidade de automatização do processo de vendas da empresa escolhida. A relevância social deste projeto reside na importância das pequenas empresas para a economia local, representando uma parte significativa do tecido empresarial e contribuindo para a geração de empregos e o desenvolvimento socioeconômico da região.

Ao enfrentarem desafios específicos, como recursos limitados e processos comerciais ultrapassados, as pequenas empresas enfrentam dificuldades para competir no mercado atual, o que ressalta a necessidade de soluções inovadoras e adaptadas às suas realidades.

Além disso, a implementação de um protótipo de sistema de gestão de vendas pode trazer uma série de benefícios que podem ser quantificados para as empresas e para a comunidade em geral. Isso inclui a melhoria da eficiência operacional, o aumento da produtividade, a redução de custos, a expansão do alcance de mercado e a geração de percepções valiosas para a tomada de decisões estratégicas. Portanto, ao contribuir para o fortalecimento das pequenas empresas locais, este projeto também promove o desenvolvimento econômico sustentável da região.

A delimitação do problema se concentra na falta de um sistema acessível de gestão de vendas nas pequenas empresas de Fartura, São Paulo. Esta lacuna tem impactos diretos na competitividade e sustentabilidade dessas empresas, limitando sua capacidade de atender às demandas dos clientes de forma eficiente e de se adaptar às mudanças do mercado. Ao abordar este problema específico, este projeto visa oferecer uma solução prática e personalizada que atenda às necessidades específicas das pequenas empresas locais, contribuindo para o seu crescimento e sucesso no cenário empresarial atual.

2.3 Fundamentação teórica

As pequenas empresas (PEs) são a base da economia brasileira, respondendo por grande parte dos negócios e da geração de empregos. No entanto, muitas PEs enfrentam desafios específicos, como recursos limitados, falta de acesso a tecnologias avançadas e a necessidade de se adaptar às constantes mudanças do mercado. Nesse contexto dinâmico, a gestão de vendas eficiente torna-se um fator crucial para o sucesso e a sustentabilidade desses negócios.

A gestão de vendas vai além da simples venda de produtos ou serviços, mas sim visa construir relacionamentos duradouros com os clientes, fidelizá-los e aumentar a rentabilidade da empresa. Através de um processo holístico que envolve todas as etapas do processo comercial, desde a prospecção de clientes até a pós-venda, a gestão de vendas eficiente contribui para:

* Aumento da produtividade da equipe de vendas: automatizando tarefas manuais, organizando informações e otimizando o tempo dos vendedores (Chiavenato, 2010).
* Melhoria da comunicação com os clientes: permitindo um atendimento mais personalizado e eficiente, além de fortalecer o relacionamento com os clientes (Kotler, 2000).
* Redução de custos operacionais: automatizando tarefas manuais, otimizando o processo de vendas e reduzindo erros (Laudon & Laudon, 2017).
* Aumento da lucratividade: através do aumento das vendas, fidelização de clientes e otimização dos recursos da empresa.

Além disso, as pequenas empresas precisam estar constantemente em busca de estratégias para que elas consigam se destacar no mercado, entre elas, destaca-se:

* Foco em nichos de mercado: identificar e direcionar seus esforços para um público específico com necessidades e desejos bem definidos, permitindo uma melhor segmentação e personalização da oferta (Kotler, 2000).
* Cultivo de relacionamentos próximos com os clientes: construir um relacionamento sólido e duradouro com os clientes, baseado na confiança e no respeito mútuo, fidelizando a base de clientes e aumentando a rentabilidade (Kotler, 2000).
* Utilização de ferramentas de marketing digital: implementar estratégias de marketing digital para alcançar novos clientes, aumentar a visibilidade da empresa e fortalecer a marca no mercado online (Kotler, 2000).

Dado o papel fundamental da tecnologia da informação (TI) para resolver os diversos problemas da sociedade. Uma das possíveis soluções que esta área pode trazer é referente aos chamados Sistemas de Gestão de Vendas (SGVs) como Ferramentas Estratégicas para Pequenas Empresas. Eles são software*s* que automatizam tarefas manuais, fornecem percepções valiosas sobre o processo de vendas e permitem a integração com outras áreas da empresa, como estoque, financeiro e marketing (Laudon & Laudon, 2017).

2.3.1 Benefícios dos Sistemas de Gestão de Vendas para Pequenas Empresas:

* Melhoria da eficiência operacional: Automatizando tarefas repetitivas, como lançamento de pedidos, controle de estoque e geração de relatórios, liberando tempo para atividades estratégicas (Martins, 2010).
* Aumento da visibilidade do processo de vendas: Fornecendo dados e insights sobre o desempenho da equipe, o comportamento dos clientes e as oportunidades de mercado, permitindo uma análise mais precisa e direcionada (Laudon & Laudon, 2017).
* Melhoria da tomada de decisões: Baseando as decisões estratégicas em dados concretos e confiáveis, ao invés de intuição ou achismos, aumentando a assertividade e o retorno sobre investimento (Laudon & Laudon, 2017).
* Aumento da competitividade: Permitindo que as PEs operem de forma mais eficiente e profissional, competindo com empresas maiores em pé de igualdade (Martins, 2010).

2.3.2 Modelo conceitual de banco de dados

Um Modelo Conceitual de Banco de Dados é uma representação abstrata e independente de qualquer sistema de gerenciamento de banco de dados específico. Ele descreve a estrutura e as relações entre os dados de uma organização de uma forma conceitual, sem entrar em detalhes de implementação. A utilização de um modelo conceitual é fundamental na fase inicial do desenvolvimento de sistemas de informação, pois ajuda a compreender e organizar os requisitos dos usuários, bem como a estrutura dos dados que serão armazenados (ELMASRI, 2019).

Um dos modelos conceituais mais amplamente adotados é o Modelo Entidade-Relacionamento (MER), proposto por Peter Chen em 1976. Neste modelo, as entidades são representadas como objetos (ou tabelas) e os relacionamentos entre essas entidades são representados por meio de linhas conectando-as. As entidades possuem atributos que descrevem suas características, e os relacionamentos podem ter cardinalidades que indicam a quantidade de instâncias de uma entidade que podem estar relacionadas a instâncias de outra entidade. O MER é amplamente utilizado devido à sua simplicidade e capacidade de representar de forma clara e intuitiva a estrutura dos dados e as relações entre eles (ELMASRI, 2019).

Outro modelo conceitual comum é o Modelo de Dados Hierárquico, que organiza os dados em uma estrutura de árvore hierárquica. Este modelo é útil para representar relações de hierarquia, como uma organização com diferentes níveis de autoridade. No entanto, ele pode ser limitado em termos de flexibilidade, uma vez que impõe uma estrutura rígida de relacionamento entre os dados.

O Modelo de Dados de Rede é uma extensão do modelo hierárquico, permitindo que um nó tenha vários pais e, portanto, criando uma estrutura mais flexível. Ele é especialmente útil para representar relações complexas entre os dados, como associações de muitos-para-muitos. No entanto, sua complexidade pode tornar a modelagem e a implementação mais desafiadoras.

O Modelo de Dados Relacional, proposto por Edgar F. Codd em 1970, é atualmente o mais amplamente utilizado na prática. Nele, os dados são organizados em tabelas compostas por linhas e colunas, onde cada tabela representa uma entidade e cada linha representa uma instância dessa entidade. As relações entre as tabelas são estabelecidas por meio de chaves estrangeiras. Este modelo oferece uma estrutura simples e intuitiva para representar os dados, facilitando a consulta e manipulação deles (CODD, 1970).

2.3.3 *Framework*

**Figura 1**- Framework React



O React é uma biblioteca JavaScript amplamente utilizada para a criação de interfaces de usuário interativas. Desenvolvida pelo Facebook em 2013, ela tem revolucionado o desenvolvimento web com sua abordagem baseada em componentes reutilizáveis, permitindo a criação de interfaces mais modulares e escaláveis (SOUZA, 2020).

Uma das principais características do React é o uso do Virtual DOM. Ao invés de realizar atualizações diretas no DOM real, o React cria uma versão virtual que otimiza a performance da aplicação, pois apenas as partes alteradas da interface são re-renderizadas, o que aumenta a eficiência no gerenciamento de interfaces dinâmicas (SILVEIRA, 2019).

Além disso, o fluxo unidirecional de dados é outra vantagem significativa. No React, os dados fluem sempre de componentes pais para filhos, tornando a depuração e a manutenção do código mais simples e previsível (Brito, 2021). Com a introdução dos React Hooks na versão 16.8, tornou-se ainda mais prático desenvolver componentes funcionais, permitindo o uso de estado e outras funcionalidades, sem a necessidade de classes (SILVEIRA, 2020).

Outra ferramenta que complementa o React é o Next.js, um framework que adiciona funcionalidades como renderização no servidor e otimização para SEO. Ele facilita o desenvolvimento de aplicações web mais performáticas, principalmente para projetos que exigem melhor performance e otimização de busca (BRITO, 2021).

2.3.4 VLibras

O VLibras Widget é um recurso de acessibilidade desenvolvido para tornar páginas web acessíveis para pessoas surdas. Com a tradução automática de Português Brasileiro para a Língua Brasileira de Sinais (LIBRAS), essa ferramenta permite que usuários surdos sejam capazes de consumir conteúdo de texto em qualquer website.

O Front End WikiLibras foi construído utilizando o framework ReactJS. O React é uma biblioteca JavaScript de código aberto com foco em criar interfaces de usuário em páginas web. É mantido pelo Facebook, Instagram, outras empresas e uma comunidade de desenvolvedores individuais. Nota: A versão do ReactJS utilizada 16.9.0 já possui suporte aos hooks. Na parte do Back End, utilizamos o framework AdonisJS na sua versão 4.

2.3.5 Computação em Nuvem

A computação em nuvem, também chamada de cloud computing, é uma tecnologia que transformou profundamente a forma como dados e serviços digitais são armazenados, gerenciados e acessados. Surgida a partir das necessidades crescentes por flexibilidade e escalabilidade no uso de recursos computacionais, a nuvem permite que empresas e indivíduos acessem uma variedade de serviços e armazenem grandes volumes de dados pela internet, sem a necessidade de investir em infraestrutura física local.

Conforme destaca Armstrong (2021, p. 23), "a computação em nuvem democratizou o acesso a ferramentas avançadas de TI, antes restritas a grandes corporações". Este modelo de computação possibilita que recursos, como armazenamento e processamento de dados, sejam fornecidos como um serviço, onde usuários podem pagar apenas pelo que utilizam, adaptando-se às demandas específicas do negócio. Dessa forma, a nuvem se mostra como uma solução eficiente e econômica para as empresas que buscam aumentar a competitividade e inovar em um mercado cada vez mais digital.

2.3.6 Modelos de Serviço da Computação em Nuvem

Existem três principais modelos de serviço na computação em nuvem, cada um oferecendo um nível diferente de controle e responsabilidade:

* IaaS (Infrastructure as a Service): A infraestrutura como serviço fornece recursos básicos de TI, como servidores, armazenamento e redes, de forma virtualizada. Com o IaaS, as empresas podem construir sua própria infraestrutura sem a necessidade de investir em hardware físico. É ideal para empresas que desejam ter controle sobre o ambiente, podendo configurar e gerenciar os recursos conforme necessário.
* PaaS (Platform as a Service): A plataforma como serviço fornece um ambiente completo para desenvolvimento e implantação de aplicativos. Inclui infraestrutura, mas também fornece ferramentas de desenvolvimento, banco de dados e gerenciamento de aplicativos. O PaaS é ideal para desenvolvedores, pois permite criar e implantar aplicativos rapidamente, sem a necessidade de gerenciar a infraestrutura subjacente.
* SaaS (Software as a Service): O software como serviço é o nível mais alto da computação em nuvem, onde aplicativos completos são oferecidos como serviços para os usuários finais. No SaaS, o fornecedor da nuvem gerencia toda a infraestrutura e plataforma, oferecendo aos usuários finais acesso direto aos aplicativos, geralmente por meio de um navegador web. Exemplos populares de SaaS incluem e-mails online, ferramentas de produtividade e softwares de CRM.

. Modelos de Implantação da Computação em Nuvem

A forma como a nuvem é implantada também pode variar, dependendo das necessidades de privacidade, segurança e controle:

* Nuvem Pública: Oferecida por empresas que disponibilizam recursos de computação para o público geral. A infraestrutura é compartilhada entre diversos clientes, sendo acessada pela internet. É mais econômica e escalável, ideal para empresas que não necessitam de configurações personalizadas ou de um nível avançado de controle.
* Nuvem Privada: Exclusiva para uma organização, geralmente configurada dentro da própria empresa ou gerenciada por um provedor. Oferece mais controle e segurança, mas é mais cara e pode exigir manutenção dedicada. É ideal para empresas que trabalham com dados sensíveis e precisam de mais controle sobre a segurança e a personalização.
* Nuvem Híbrida: Combina a nuvem pública e privada, permitindo que dados e aplicativos sejam compartilhados entre ambas. Isso permite que as empresas mantenham dados mais críticos e sensíveis em uma nuvem privada enquanto aproveitam a escalabilidade e economia da nuvem pública para outros recursos. É uma opção popular para empresas que buscam flexibilidade.

2.3.6 Git e github

Git é um sistema de controle de versão distribuído amplamente utilizado para o gerenciamento de código-fonte durante o desenvolvimento de software. Criado por Linus Torvalds em 2005, Git revolucionou a forma como equipes de desenvolvimento colaboram e coordenam o trabalho em projetos de software (Chacon & Straub, 2014). Ele foi inicialmente concebido para o desenvolvimento do kernel do Linux, mas desde então se tornou uma ferramenta essencial para projetos de todos os tamanhos e domínios (Loeliger & McCullough, 2012).

A principal vantagem do Git reside na sua natureza distribuída, que permite que cada desenvolvedor mantenha uma cópia completa do repositório de código em sua máquina local. Isso possibilita que os desenvolvedores trabalhem offline e realizem commits locais, o que é fundamental para a flexibilidade e produtividade durante o desenvolvimento de software (Chacon & Straub, 2014).

GitHub, por sua vez, é uma plataforma de hospedagem de código-fonte baseada na web que utiliza o Git como sistema de controle de versão subjacente. Lançado em 2008, o GitHub revolucionou a forma como o código é compartilhado e colaborado na comunidade de desenvolvimento de software (Dabbish et al., 2012). Ele oferece uma ampla gama de recursos adicionais para colaboração e gerenciamento de projetos, incluindo controle de acesso baseado em permissões, rastreamento de problemas, gerenciamento de solicitações de *pull* e integração contínua (GitHub, 2024).

Um dos conceitos-chave do Git é o de ramificação (*branching*), que permite que os desenvolvedores trabalhem em novas funcionalidades ou correções de bugs sem interferir no código principal do projeto. Isso possibilita o desenvolvimento paralelo e iterativo, onde várias equipes podem trabalhar em diferentes partes do código simultaneamente, sem conflitos (Loeliger & McCullough, 2012; Spinellis, 2012).

Outro conceito fundamental do Git é o de *commit*, que registra uma mudança específica no código-fonte, juntamente com uma mensagem descritiva que explica o propósito da mudança. Os *commits* formam a base do histórico de versões do projeto e permitem que os desenvolvedores rastreiem e revertam alterações conforme necessário, garantindo a integridade e rastreabilidade do código ao longo do tempo (Chacon & Straub, 2014).

O GitHub oferece uma série de recursos sociais e de colaboração, como "*forking*" (clonagem) de repositórios de outros usuários para contribuição colaborativa, rastreamento de problemas (*issues*) para relatar e discutir bugs e solicitações de *pull* (*pull requests*) para propor e discutir mudanças no código-fonte. Esses recursos tornam o GitHub uma plataforma poderosa para colaboração e contribuição para projetos de código aberto (Dabbish et al., 2012).

2.4 Aplicação das disciplinas estudadas no projeto integrador

No desenvolvimento do projeto integrador, a aplicação prática dos conhecimentos adquiridos em diversas disciplinas foi fundamental para a construção de um sistema de gestão de vendas eficiente. As disciplinas de Banco de Dados, Estrutura de Dados, Fundamentos da Internet e Web, Programação Orientada a Objetos, e Gestão da Inovação e Desenvolvimento de Produtos desempenharam papéis cruciais na execução e no sucesso do projeto.

2.4.1 Algoritmos e Programação de Computadores I e II

Nestas disciplinas, foram abordados conceitos fundamentais que foram essenciais para o desenvolvimento inicial do sistema em termos de estruturas básicas de programação. A introdução a algoritmos e à linguagem Python pavimentou o caminho para o uso do *framework* React. O conhecimento sobre tipos de dados e listas foi importante na organização de dados listas de produtos e clientes. Além disso, a habilidade de depuração de programas foi indispensável para identificar e corrigir erros no código.

Os conceitos iniciais sobre programação orientada a objetos foram diretamente aplicados na estruturação do código, pois o *framework* Django se baseia nesse paradigma. Estruturas de dados como pilhas, filas e árvores certamente são usadas direta ou indiretamente pelo *framework*, embora isso fique geralmente transparente para que o utiliza. A utilização de GIT e testes automatizados garantiu o controle de versões do software durante o desenvolvimento e as atualizações.

2.4.2 Formação Profissional em Computação

Os conhecimentos adquiridos nesta disciplina foram aplicados em relação a compreensão de padrões importantes da Web, como HTML. A disciplina possibilitou compreender como deve ser feita a utilização de bibliotecas e *frameworks* para o desenvolvimento rápido e eficiente de sistema. O estudo sobre APIs e serviços web também contribuiu para pavimentar o conhecimento prévio necessário para a utilização do framework Django.

2.4.3 Infraestrutura para Sistemas de Software

Nesta disciplina, o aprendizado sobre computação em nuvem foi essencial para a criação da máquina virtual utilizada como o servidor do protótipo implementado nesse trabalho. Os estudos de caso de infraestrutura para sistemas de software forneceram insights úteis sobre as melhores práticas para a implementação e manutenção do sistema, visando sua robustez, eficiência operacional e confiabilidade.

2.4.4 Banco de Dados

A disciplina de Banco de Dados forneceu a base teórica e prática necessária para a modelagem, implementação e gerenciamento dos dados no sistema de gestão de vendas. Utilizamos conceitos de modelagem de dados para criar um modelo conceitual robusto, que foi posteriormente traduzido em um modelo lógico utilizando SQL. As habilidades adquiridas na disciplina permitiram a compreender como o Django cria tabelas, índices e consultas otimizadas para garantir um desempenho adequado do sistema.

2.4.5 Estrutura de Dados

A disciplina de Estrutura de Dados foi fundamental para a implementação de algoritmos eficientes e a escolha de estruturas de dados adequadas para o sistema de gestão de vendas. Internamente, o framework React utiliza as estruturas de dados estudadas na disciplina para gerenciar as diversas operações de inserção, exclusão e busca no sistema. A estrutura de dados lista foi visivelmente a mais utilizada nas páginas do protótipo, uma vez que cada uma das entidades possui potencialmente várias instâncias que são criadas e mantidas conforme a utilização da aplicação, essas instâncias são naturalmente exibidas em forma de lista.

2.4.6 Fundamentos da Internet e Web

Os conhecimentos adquiridos na disciplina de Fundamentos da Internet e Web foram essenciais para o desenvolvimento da interface do sistema de gestão de vendas e para a implementação do *front-end* da aplicação. Foram utilizadas tecnologias web como HTML e CSS para criar uma interface de usuário amigável e responsiva.

2.4.7 Programação Orientada a Objetos

A disciplina de Programação Orientada a Objetos (POO) foi aplicada no desenvolvimento do sistema de gestão de vendas diretamente na definição dos modelos requeridos pelo *framework* Django. Os conceitos teóricos da disciplina permitiram que as classes que refletem as entidades do domínio do problema fossem definidas. Por exemplo, classes como ‘Produto’, ‘Pedido’ e ‘Cliente’ foram criadas para modelar os dados e comportamentos associados a cada uma dessas entidades, promovendo uma estrutura de código clara e organizada. A aplicação dos princípios de design orientado a objetos contribuiu para a flexibilidade e a extensibilidade do sistema, facilitando futuras adições e modificações de funcionalidades.

2.4.8 Gestão da Inovação e Desenvolvimento de Produtos

A disciplina de Gestão da Inovação e Desenvolvimento de Produtos forneceu as estratégias e metodologias necessárias do ponto de vista de negócios para a condução do projeto integrador. Aplicamos conceitos de *design* *thinking* e metodologias ágeis para identificar as necessidades dos usuários, prototipar soluções e iterar sobre elas com base no feedback recebido. A gestão do ciclo de vida do produto foi planejada para garantir que cada etapa do desenvolvimento fosse alinhada com os objetivos do projeto e as expectativas dos *stakeholders*.

2.4.9 Integração das Disciplinas

A integração dos conhecimentos adquiridos em todas essas disciplinas permitiu a construção do protótipo de um sistema de gestão de vendas funcional. O projeto beneficiou-se da aplicação prática dos conceitos teóricos e práticos, resultando em um produto final que não apenas atende às necessidades imediatas da empresa, mas também possui potencial para escalabilidade e melhorias futuras. Cada disciplina contribuiu com um conjunto específico de habilidades e conhecimentos que, quando combinados, resultaram na solução proposta neste trabalho.

2.5 Metodologia

A metodologia empregada neste projeto foi elaborada visando a criação de um protótipo de sistema de gestão de vendas customizado às necessidades da pequena empresa atendida. A seguir são descritas as estratégias utilizadas em cada etapa do projeto.

2.5.1 Ouvir e interpretar o contexto

O projeto foi conduzido em Fartura, São Paulo, com foco nas pequenas empresas locais, que enfrentam desafios na gestão de vendas decorrentes da ausência de sistemas apropriados.

Os participantes foram selecionados entre proprietários e gestores de pequenas empresas em Fartura, bem como funcionários envolvidos nos processos de vendas e gestão.

As informações iniciais foram coletadas por meio de entrevistas com os proprietários e gestores das empresas, além da observação direta dos processos de vendas sempre que viável. Com essa abordagem, os integrantes do grupo adquiriram uma compreensão holística das necessidades e desafios enfrentados pelas empresas em relação à gestão de vendas.

2.5.2 Prototipação

Os dados coletados foram analisados tanto qualitativamente quanto quantitativamente. A análise qualitativa envolveu a identificação de padrões, temas e desafios comuns emergentes das entrevistas e observações. Paralelamente, a análise quantitativa dos dados obtidos permitiu uma compreensão mais ampla das tendências e métricas relacionadas aos processos de vendas das pequenas empresas.

Com base na análise dos dados, foram concebidas as versões iniciais do esquema do banco de dados e das janelas da interface gráfica da aplicação. Para o esquema do banco de dados, foi desenvolvido um diagrama conceitual utilizando a plataforma draw.io, que representa as principais entidades levantadas no sistema, bem como seus atributos e relacionamentos iniciais. Esse diagrama proporcionou uma visão conceitual, estruturada e organizada das informações que serão armazenadas e manipuladas pelo sistema.

A partir do diagrama, os conceitos ali representados podem ser implementados fazendo uso do framework React. Esse framework é o ponto central do projeto porque permite criar toda a estrutura de programação, banco de dados e interface gráfica do projeto. As informações descritas no modelo conceitual foram interpretadas e convertidas para código ------ e assim cada uma das janelas do protótipo foram geradas devidamente configuradas para serem exibidas pelo framework. Essas janelas são acessíveis por meio da interface gráfica web proporcionada pelo próprio framework React. Isso permite que o protótipo possa se tornar facilmente uma aplicação final, uma vez que as informações representadas já são consistentes e fiéis ao que se espera de uma aplicação real.

2.5.3 Implementação e testes

Conforme apresentado na subseção anterior, a escolha do framework React permite que a diferença do protótipo para a aplicação final seja mínima. Dessa forma, os esforços de prototipação acabam sendo aproveitados para a fase de implementação.

Em sua versão inicial, o protótipo foi apenas testado no ambiente computacional local do próprio computador do integrante responsável. Por não se tratar de uma aplicação completa, os testes consistiram apenas em verificar a funcionalidade básica das principais características da aplicação, como a inserção de dados, a navegação entre telas e a correta exibição das informações. Esses testes foram realizados manualmente, simulando o uso do sistema por um usuário real, a fim de identificar possíveis erros ou inconsistências na interface e na lógica de funcionamento.

Desde a sua criação até a finalização, as versões do protótipo foram todas gerenciadas através do GitHub, de modo que cada alteração significativa consistiu numa operação *commit* no repositório do projeto. Nesse sentido, o GitHub serviu como uma ferramenta importante para o controle das versões e atualizações, isto é, o progresso do protótipo.

Na sua versão final, o protótipo foi configurado em uma máquina virtual na nuvem, permitindo que qualquer usuário com conexão à Internet possa acessá-lo. Nesta versão, uma página inicial foi criada para o usuário ter o primeiro contato com o sistema. Ao clicar no botão de acessar sistema, o usuário é direcionado à página de login para poder ter acesso ao sistema de fato.

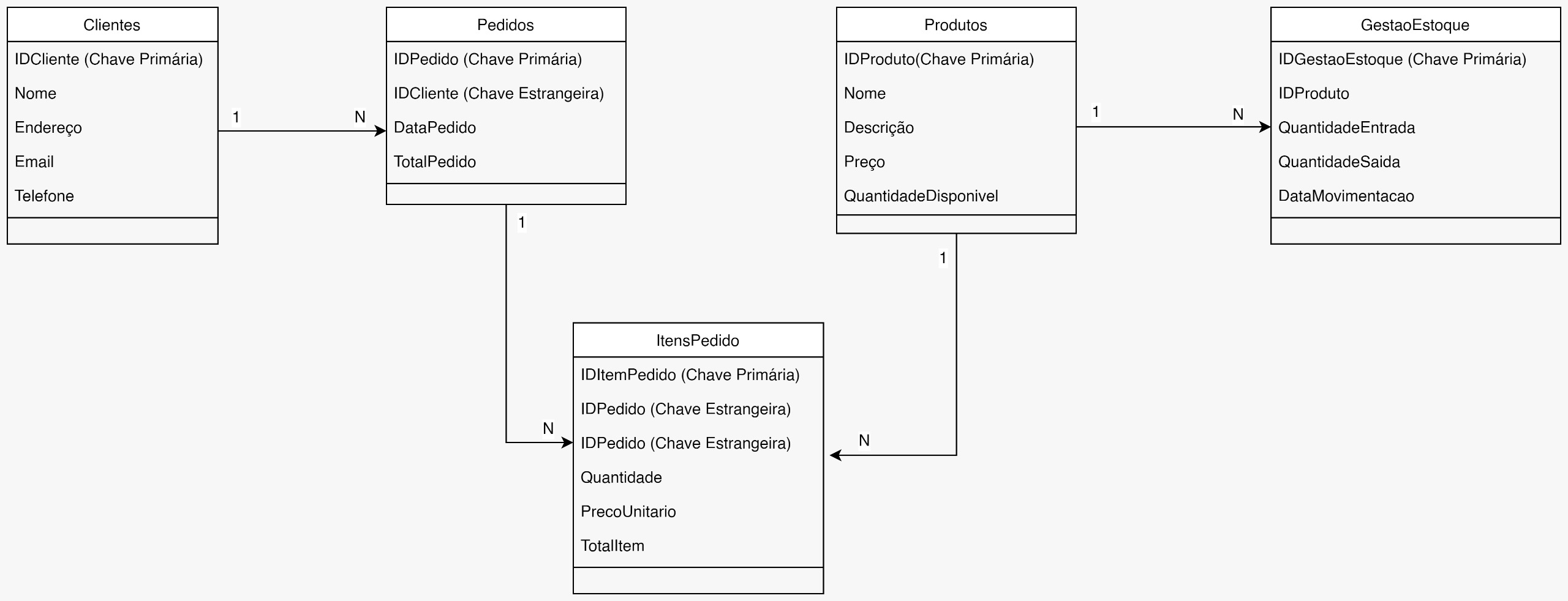
**3 Resultados: solução final**

O protótipo desenvolvido tem por objetivo automatizar o processo de vendas da empresa selecionada. Para isso, por meio da última entrevista realizada com o proprietário da empresa, concluiu-se que são necessárias as seguintes entidades, bem como seus respectivos atributos:

* Cliente, com o registro do código identificador, nome, endereço, e-mail e telefone;
* Produto, com o registro do código identificador, nome, descrição, preço e quantidade disponível;
* Pedido, com o registro do código identificador, código do cliente associado ao pedido, data do pedido e valor total do pedido;
* Itens do pedido, com o registro do código identificador do item pedido, códigos do pedido e do produto associado, quantidade, preço unitário e total do item;
* Gestão de estoque, com o registro do código identificador, código do produto, quantidade de entrada, quantidade de saída e data da movimentação.

Dessa forma, ao ouvir a necessidade da empresa atendida, o diagrama conceitual do banco de dados foi gerado utilizando a plataforma draw.io, de modo a representar visualmente as entidades e seus relacionamentos. O diagrama pode ser visto na Figura 2.

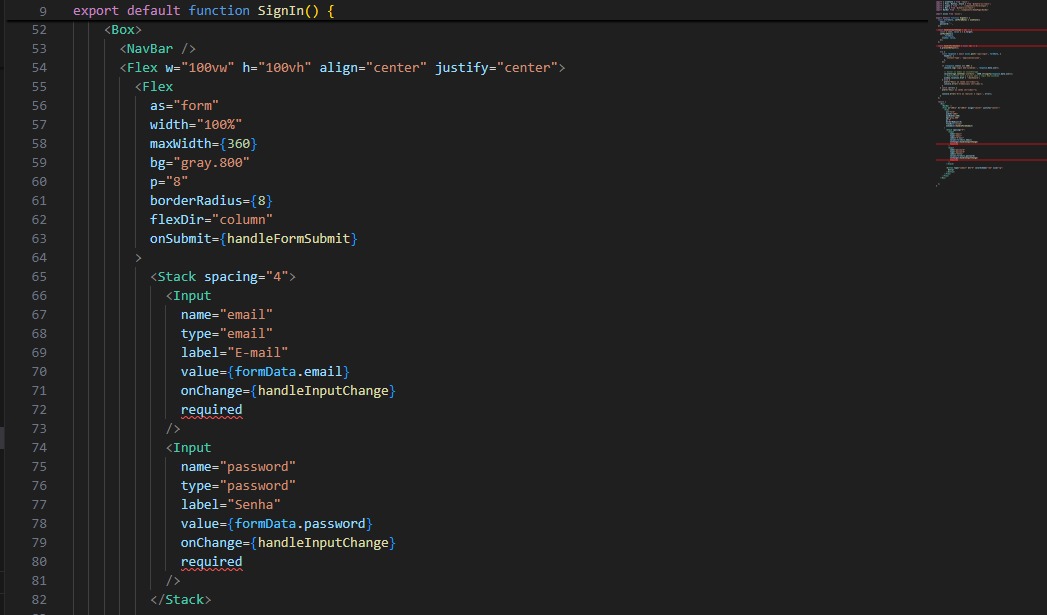
**Figura 2** - Diagrama conceitual do banco de dados

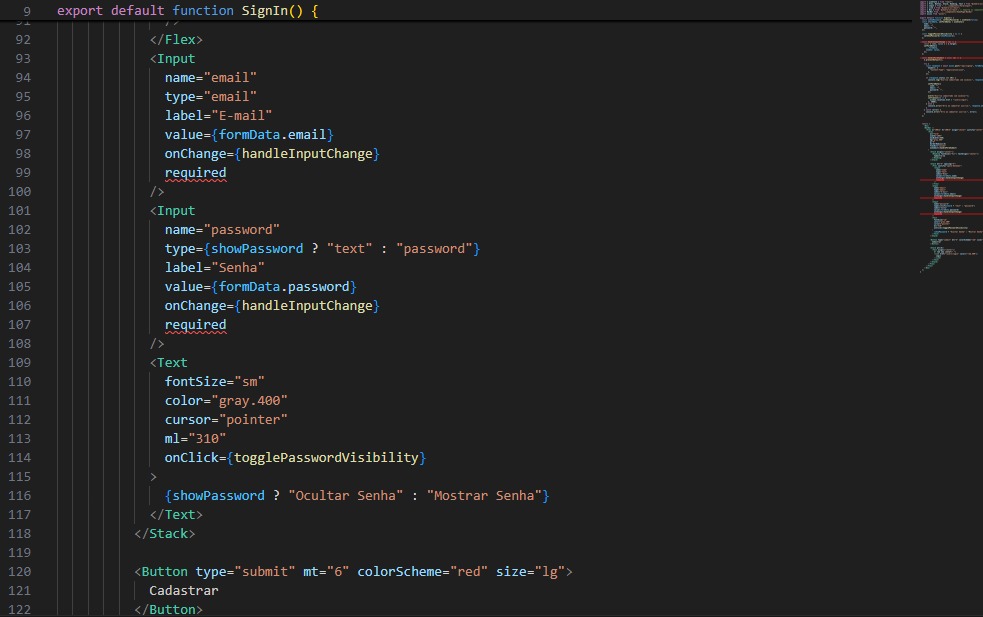
Fonte: elaborada pelos autores, 2024.

3.1 Definição dos modelos (classes)

A elaboração do diagrama permitiu a definição das classes no *framework* Django. A definição deve ser feita no arquivo models.py conforme orientação da documentação oficial. Dessa forma, o resultado da tradução do diagrama da Figura 2 resultou na elaboração dos modelos apresentados na Figura 3.

**Figura 3** - Modelos das entidades do projeto





Fonte: elaborada pelos autores, 2024.

Os relacionamentos entre as entidades também são pontos fundamentais para a lógica de funcionamento do protótipo. Com base nas Figuras 2 e 3, as seguintes regras são observadas:

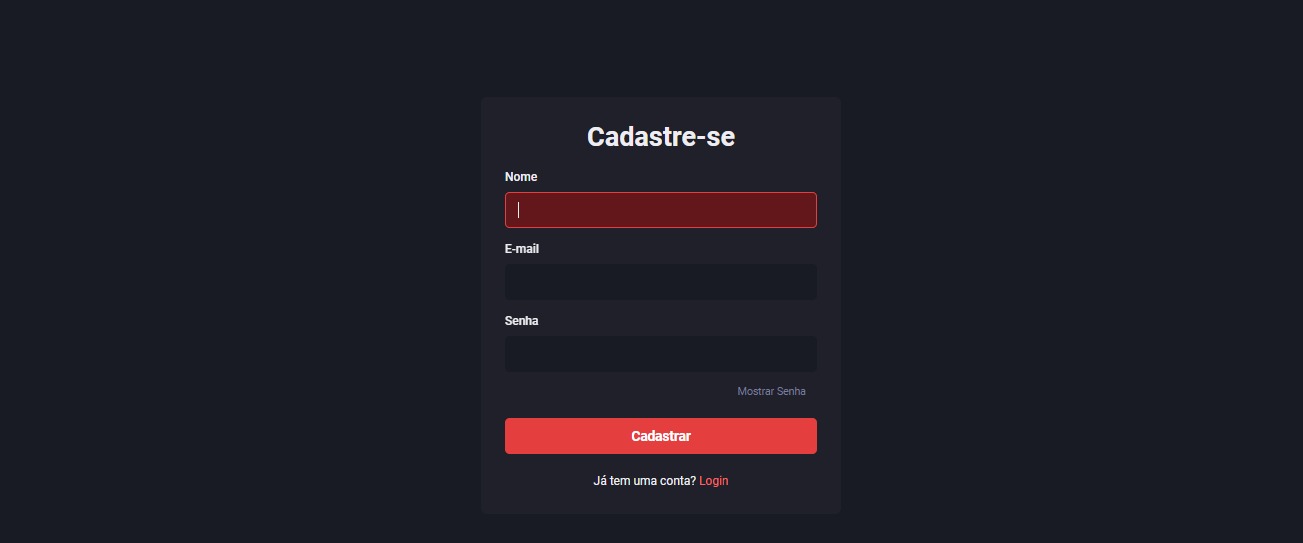
* Um cliente pode fazer muitos pedidos, mas cada pedido é feito por apenas um cliente. Portanto, há um relacionamento de "um para muitos" entre a tabela Clientes e a tabela Pedidos.
* Um pedido pode conter muitos itens de pedido, e um item de pedido só pode estar associado a um único pedido. Portanto, há um relacionamento de "um para muitos" entre a tabela Pedidos e a tabela ItensPedido.
* Cada produto pode ter muitos itens de pedido associados a ele, mas cada item de pedido está associado a apenas um produto. Portanto, há um relacionamento de "um para muitos" entre a tabela Produtos e a tabela ItensPedido.
* A tabela GestaoEstoque mantém o registro de movimentações de estoque para cada produto. Cada movimentação está associada a apenas um produto, mas um produto pode ter muitas movimentações registradas. Portanto, há um relacionamento de "um para muitos" entre a tabela Produtos e a tabela GestaoEstoque.

Além das definições da Figura 3, cada entidade também possui por padrão um código identificador controlado pelo próprio Django. Para fins de visualização, o método \_\_str\_\_ de cada classe foi sobrescrito para a correta representação textual das instâncias.

3.2 páginas do sistema

A Figura 4 ilustra as entidades criadas e habilitadas para serem manipuladas no painel administrativo do React.

**Figura 4** - Classes das entidades habilitadas no painel do React



Fonte: elaborada pelos autores, 2024.

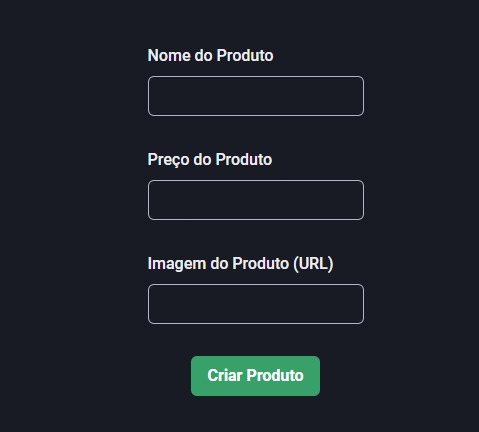
Cada uma das classes criadas permite as operações básicas de criação, listagem, alteração e remoção de seus objetos. A Figura 5 ilustra como um objeto da classe cliente pode ser cadastrado usando o painel administrativo.

**Figura 5** - Cadastro de um novo cliente usando o painel administrativo

Fonte: elaborada pelos autores, 2024.

A Figura 6 ilustra como os pedidos cadastrados no sistema são exibidos. Cada pedido possui um código identificador, um vendedor associado, a data da efetivação do pedido e o seu valor total. Todas essas informações são mostradas de maneira resumida na listagem.

**Figura 6 -** Listagem dos pedidos

  
Fonte: elaborada pelos autores, 2024.

Desde a versão inicial, não foram observados erros conceituais no esquema de banco de dados, de modo que a modelagem inicial se manteve para a versão final do protótipo. A apresentação do protótipo final para o proprietário não resultou em nenhum pedido significativo de alteração, uma vez que se trata de um protótipo com limitações naturalmente esperadas.

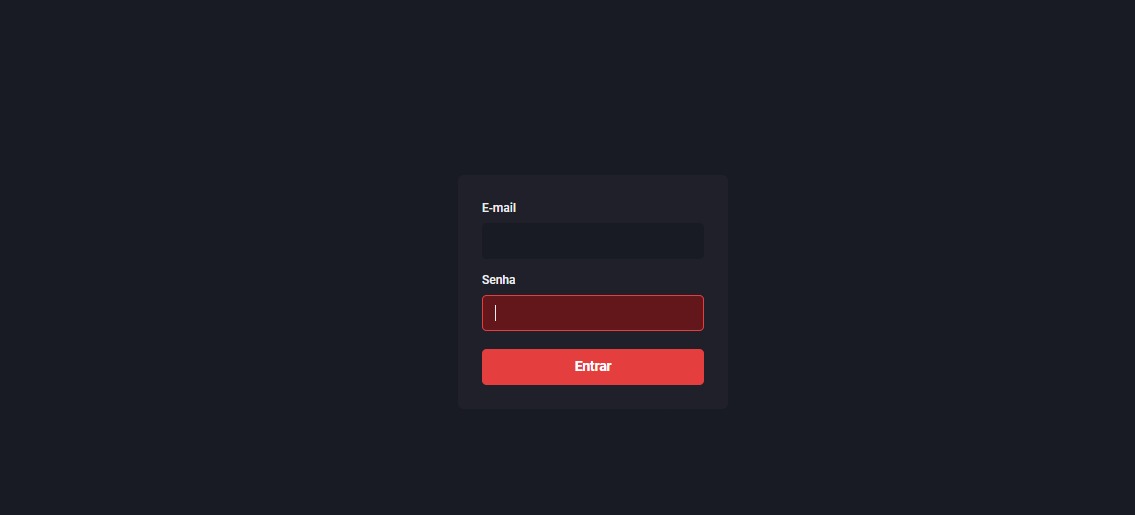
Uma página inicial simples foi desenvolvida para os usuários que acessarem o domínio brainsales.com.br. Ela foi projetada para dar as boas-vindas a um visitante ou direcionar um usuário credenciado para a tela de login do sistema. A Figura 7 ilustra a página inicial desenvolvida.

**Figura 7 -** Página inicial do protótipo

  
Fonte: elaborada pelos autores, 2024.

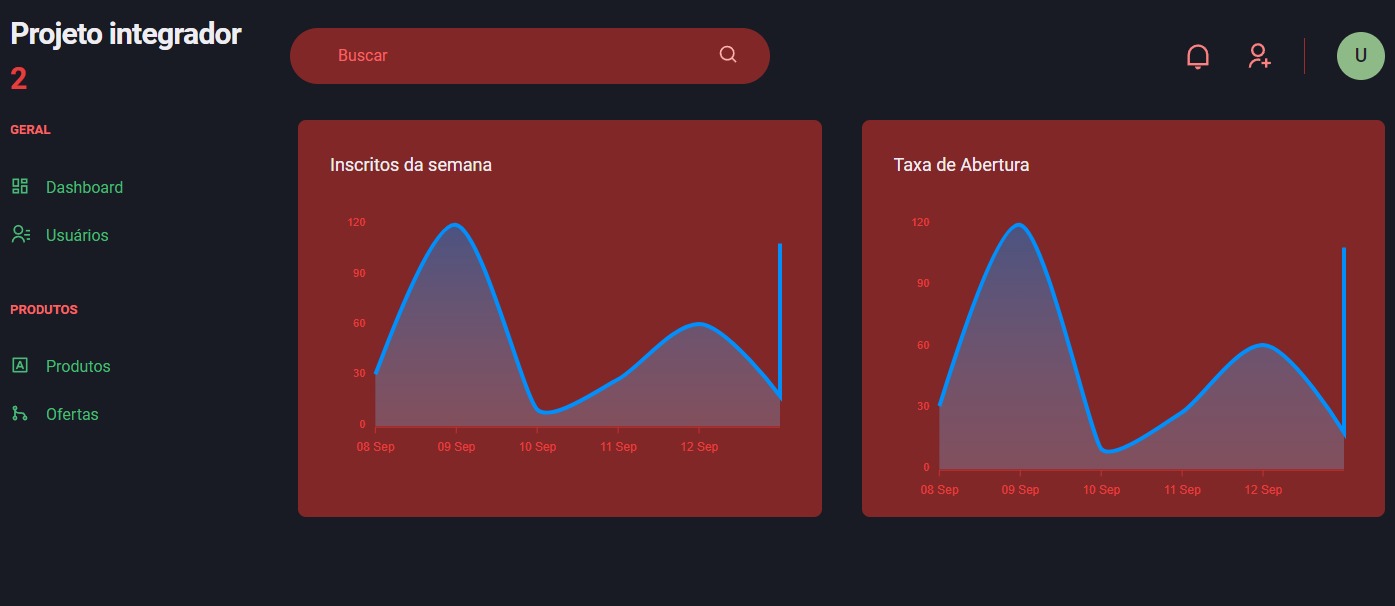
Ao clicar no botão “criar nova conta”, o usuário é levado para a página de login, que pode ser vista na Figura 8.

**Figura 8 -** Página de login

  
Fonte: elaborada pelos autores, 2024.

Após essa etapa, o usuário devidamente credenciado é redirecionado para a página central da aplicação. Por meio dessa página é que todos os recursos do protótipo proposto podem ser acessados. A interface permite que para cada entidade definida na seção vendas, as operações de adicionar, modificar, alterar, excluir e listar sejam realizadas de forma fácil e intuitiva a partir da interação com as páginas da aplicação. A Figura 9 ilustra a página principal do protótipo.

**Figura 9 -** Página principal do protótipo



Fonte: elaborada pelos autores, 2024.

Algumas configurações nos arquivos internos do React foram necessárias para um melhor ajuste no sistema. Essas configurações incluem:

* Alterar as configurações de internacionalização do servidor para que a data e hora, bem como os textos sejam exibidos em português;
* Modificar alguns arquivos internos das páginas do administrador para que o nome do protótipo fosse exibido. Isso exigiu que as mensagens do sistema fossem recompiladas.

3.3 Gerenciamento de versões com GitHub

Por fim, a plataforma GitHub foi importante para o versionamento dos arquivos que compõem este protótipo. Um repositório chamado “PI-I-Brain-sales-platform” foi criado para isso, assim como seu devido arquivo readme para a descrição das informações do repositório. Uma parte do que é mostrado nesse arquivo pode ser visto na Figura 10.

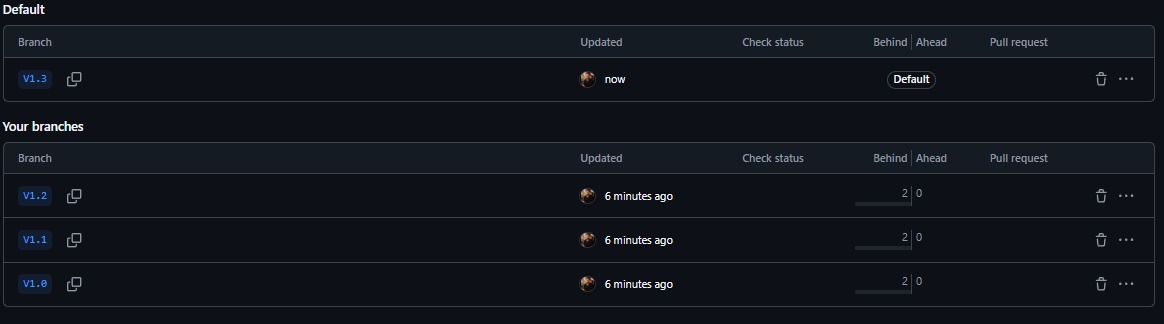
**Figura 10 -** Parte do arquivo readme



Fonte: elaborada pelos autores, 2024.

Cada *commit*, isto é, a operação que faz alterações no repositório, foi realizado de modo a marcar os principais avanços realizados no projeto. Esses avanços incluem desde a criação e definição inicial do projeto, até a última modificação referente a versão final. O histórico dos *commits* feitos no repositório pode ser visto na Figura 11.

**Figura 11 -** Histórico



Fonte: elaborada pelos autores, 2024.

**4 Considerações Finais**

Este projeto abordou a implementação de um protótipo de sistema de gestão de vendas para pequenas empresas, destacando a importância da tecnologia da informação como uma ferramenta vital para otimizar processos comerciais e aumentar a competitividade no mercado.

A adoção do sistema de gestão de vendas proposto trouxe benefícios claros e observáveis, evidenciando o impacto positivo da tecnologia nas pequenas empresas:

1. **Melhoria da Eficiência Operacional**: A automatização de processos, como o cadastro de clientes e o gerenciamento de pedidos, resultou em uma operação mais fluida e eficiente, reduzindo o tempo gasto em tarefas manuais e propensas a erros.
2. **Aumento da Produtividade**: Com a implementação do sistema, os funcionários agora podem focar em atividades estratégicas, melhorando a produtividade geral da empresa.
3. **Redução de Custos**: A integração de dados e a centralização das operações permitem uma gestão mais eficaz dos recursos, resultando em uma redução significativa dos custos operacionais.
4. **Expansão do Alcance de Mercado**: O sistema tem o potencial de facilitar a análise de dados e a identificação de novas oportunidades de mercado, contribuindo para a expansão dos negócios.

Todos esses benefícios são possíveis em um sistema que de fato seja posto em produção e amplamente utilizado e aceito pela empresa e seus vendedores. O protótipo criado não é evidentemente um sistema final, mas permite vislumbrar esses benefícios.

Apesar dos benefícios, a implementação do sistema também apresentou alguns desafios:

1. **Ajustes e Correções**: Durante o desenvolvimento, foi necessário realizar diversos ajustes para corrigir erros conceituais e melhorar a funcionalidade do sistema, o que demandou tempo e recursos adicionais.
2. **Domínio das tecnologias empregadas**: O conhecimento da maneira correta de utilizar cada tecnologia empregada na realização do protótipo, bem como a integração entre elas foi um desafio constante enfrentado por todos os membros do grupo. Com o devido esforço e persistência, as barreiras de curva de aprendizado e problemas pontuais foram sendo eliminadas em grupo, possibilitando a finalização do projeto.

O protótipo desenvolvido neste projeto representa um passo importante para a digitalização das pequenas empresas, mas há espaço para melhorias e expansões futuras:

1. **Integração com Outros Sistemas**: Futuras versões do sistema poderiam integrar-se com outras ferramentas de gestão empresarial, como sistemas de contabilidade e CRM, oferecendo uma solução mais completa.
2. **Personalização e Escalabilidade**: Desenvolver funcionalidades adicionais e permitir maior personalização do sistema para atender às necessidades específicas de diferentes tipos de negócios será crucial.
3. **Análise de Dados Avançada**: Implementar ferramentas de análise de dados mais avançadas pode proporcionar insights ainda mais valiosos para a tomada de decisões estratégicas.
4. **Versão mobile**: Embora o React seja relativamente responsivo, uma versão dedicada da aplicação dispositivos móveis seria interessante para o contexto da empresa, visto que muitos vendedores dependem do uso de celular para escreverem os pedidos.

Por fim, a implementação do protótipo Brain Sales Platform mostrou-se adequado e útil para a empresa escolhida, com potencial significativo para transformar a operação de pequenas empresas como ela. As tecnologias e ferramentas utilizadas foram fundamentais para a correta produção do protótipo. Espera-se que o protótipo possa ser de fato útil para o cotidiano dos colaboradores da empresa.

**Referências**

ABNT – ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. ***NBR 14724****: Informação e documentação. Trabalhos Acadêmicos - Apresentação*. Rio de Janeiro: ABNT, 2002.

ARMSTRONG, Michael. ***Computação em Nuvem****: Teoria e Prática*. 3. ed. Rio de Janeiro: Editora Tecno, 2021.

BROWN, T. ***Design Thinking: Uma Metodologia*** *Poderosa para Decretar o Fim das Velhas Ideias*. Rio de Janeiro: Elsevier, 2010.

COSTA, Mariana; ALMEIDA, Carlos**. *Serviços em Nuvem: IaaS, PaaS e SaaS para Negócios Modernos*.** São Paulo: Editora Novatec, 2022.

CHIAVENATO, Idalberto. ***Administração de Vendas***. São Paulo: Elsevier, 2010.

CHACON, Scott; STRAUB, Ben. *Pro Git*. Springer Nature, 2014.

CHEN, P. P. *The Entity-Relationship* ***Model—Toward a Unified View of Data*. ACM Transactions on Database Systems** (TODS), v. 1, n. 1, p. 9-36, 1976.

CODD, E. F. ***A Relational Model of Data for Large Shared Data Banks***. Communications of the ACM, v. 13, n. 6, p. 377-387, 1970.

DABBISH, L.; STUART, C.; TSAY, J.; HERBSLEB, J. D. Social coding in **GitHub:** Transparency and collaboration in an open software repository. *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work* (CSCW '12), p. 1277-1286, 2012. <https://doi.org/10.1145/2145204.2145396>. The Web framework for

DJANGO. **perfectionists with deadlines |** Django. Disponível em: <https://www.djangoproject.com/>. Acesso em: 01 out. 2024.

ELMASRI, Ramez et al. ***Sistemas de Banco de Dados***. 6. ed. Pearson, 2019.

FERREIRA, Ricardo; SANTOS, Paula. ***Modelos de Implantação da Computação em Nuvem****: Uma Abordagem Prática*. Brasília: Editora Acadêmica, 2020.

GOMES, Lúcia. ***Desafios e Benefícios da Computação em Nuvem na Indústria*** *4.0*. Porto Alegre: Editora Ciência e Tecnologia, 2021.

GITHUB. Let’s build from here. Disponível em: <https://github.com/>. Acesso em: 30 out. 2024.

KOTLER, Philip; ARMSTRONG, **Gary. *Administração de Marketing***. São Paulo: Atlas, 2000.

LAUDON, Kenneth C.; LAUDON, Jane P. ***Sistemas de Informação Gerenciais: Administrando a Empresa Digital***. 12. ed. Bookman Editora, 2017.

LOELIGER, Jon; MCCULLOUGH, Matthew. ***Version Control with Git: Powerful Tools and Techniques for Collaborative Software Development*.** O'Reilly Media, Inc., 2012.

MACHADO, Fernando. ***Nuvem Híbrida e Segurança da Informação****: Estratégias para Negócios*. Salvador: Editora Soluções Empresariais, 2019.