

UNI SENAI SANTA CATARINA
Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas
Projeto Aplicado I (AEx)

**DOCUMENTAÇÃO TÉCNICA DO PROJETO APLICADO: ORGANIZAÇÃO DA
GESTÃO DE PEDIDOS**

Equipe:

Amanda Marques Pardiniho
Daiane Olete da Silva Maniçoba
Fabíola Reginaldo Corrêa da Silva
Marta Sayuri Mukai

Professor(a) Responsável:

Iskailer Inaian Rodrigues

Florianópolis - Santa Catarina

2025

SUMÁRIO

1. EQUIPE E PLANEJAMENTO DAS ATIVIDADES	PÁG. 03
1.1. Nome da Equipe	
1.2. Integrantes e Funções	
1.3. Cronograma do Projeto	
2. PROBLEMA ESCOLHIDO E DESCRIÇÃO	PÁG. 04
2.1. Contextualização	
2.2. Descrição do Problema	
2.3. Relevância	
2.4. Impacto	
2.5. Fontes de Pesquisa	
2.6. Justificativa	
3. NECESSIDADES DO CLIENTE/USUÁRIO E VALIDAÇÃO	PÁG. 07
3.1. Perfil do Cliente/Usuário	
3.2. Necessidades Identificadas	
3.3. Método de Validação	
3.4. Conclusões Obtidas	
4. TECNOLOGIAS ESCOLHIDAS E JUSTIFICATIVA	PÁG. 09
4.1. Frontend	
4.2. Backend	
4.3. Banco de Dados	
4.4. Hospedagem/Deploy	
4.5. Outras Ferramentas	
5. DIAGRAMA E DESCRIÇÃO DA SOLUÇÃO PROPOSTA	PÁG. 11
5.1. Descrição Geral da Solução	
5.2. Diagrama de Arquitetura	
5.3. Protótipo da Interface	
6. REFERÊNCIAS.....	PÁG.10

1. Equipe e Planejamento das Atividades

Para o desenvolvimento deste projeto, montamos uma equipe preparada para atender as mais variadas demandas e necessidades que possam surgir. A seguir apresentamos um breve resumo da qualificação profissional de cada integrante desta equipe.

1.1. Apresentação da Equipe

Amanda, formada em Farmácia, trabalha atualmente como desenvolvedora de software júnior e atuará na gestão do projeto. Formada em Contabilidade, a Daiane organizará a comunicação com o cliente, refinando a demanda para produção. Já a Fabíola é administradora e atuará na parte de designer e apresentações no decorrer do projeto. Por fim, Marta conta com uma vasta experiência em gestão de fábrica alimentícia e ficará responsável por documentar todo o processo até a entrega final.

1.2. Integrantes e Funções

Para agilizar a organização e a comunicação entre os envolvidos, elaboramos um quadro identificando as funções de cada integrante.

Integrantes e Funções		
Nome	Função	Principais responsabilidades
Amanda Marques Pardinho	Desenvolvedora	Backend/Gestão do projeto
Daiane Olete da Silva Maniçoba	Contadora	Comunicação
Fabíola Reginaldo Corrêa da Silva	Administradora	Frontend/Designer
Marta Sayuri Mukai	Administradora	Compilação arquivos

Com a equipe e suas funções definidas, seguiremos para a proposta do cronograma a ser executado durante a elaboração do projeto.

1.3. Cronograma do Projeto

Cronograma do Projeto			
Responsável	Descrição	Data Limite	Finalizado
Equipe	Definir a equipe, funções e tema do projeto	12/08/2025	Sim
Equipe	Definir assunto, cronograma inicial e objetivos gerais do projeto	12/08/2025	Sim
Equipe	Analisar o projeto SAGA SENAI	19/08/2025	Não
Daiane/Marta	Entender a demanda com o cliente solicitante	30/08/2025	Não
Daiane/Marta	Entrevistas com stakeholders	02/09/2025	Não
Daiane/Marta	Validar requisitos e aprovar com a empresa	02/09/2025	Não
Amanda/Fabíola	Definir arquitetura, design e implementação	16/09/2025	Não
Daiane/Marta	Aprovar primeira versão	30/09/2025	Não
Amanda/Fabíola	Correções funcionais e técnicas	07/10/2025	Não
Equipe	Entrega versão final	05/12/2025	Não

2. Problema escolhido e Descrição

A gestão de pedidos é fundamental para o sucesso nas vendas. Atrasar o envio de um pedido ou, pior, enviar um pedido errado ao cliente compromete a imagem da empresa enquanto organização, gera custos adicionais considerando a

devolução de mercadorias enviadas erroneamente e compromete o relacionamento com o cliente.

Por outro lado, uma gestão eficiente gera dados valiosos para a direção, e pode direcionar corretamente esforços das áreas de vendas e marketing, entre outras.

Além de reduzir erros e atrasos, uma gestão de pedidos bem organizada permite o acompanhamento completo de cada etapa do processo, desde a entrada do pedido até a entrega final. Isso facilita a identificação de pontos críticos, melhora a precisão dos estoques e aumenta a satisfação do cliente, ao garantir que suas expectativas sejam atendidas de forma precisa.

2.1. Contextualização

Atualmente, muitas pequenas e médias empresas enfrentam dificuldades na organização e controle de seus pedidos, tanto em comércios físicos quanto virtuais. Esse cenário se agrava com o aumento das vendas em canais digitais e a falta de integração entre os meios de atendimento (loja, WhatsApp, redes sociais, e-commerce, entre outros). Em muitos casos, o controle é feito manualmente, em planilhas ou anotações, o que aumenta as chances de erros, atrasos e perda de informações.

A concorrência acirrada existente atualmente, não permite mais empresas que não tenham agilidade e efetividade nas suas atividades meio. Ou a empresa informatiza seus setores, ou as dificuldades vão surgir exponencialmente, comprometendo sua permanência no mercado.

Além disso, a crescente exigência dos clientes por agilidade e precisão torna indispensável a adoção de soluções tecnológicas que integrem canais de venda e centralizem informações. Sendo assim, a gestão de pedidos deixa de ser apenas operacional e passa a ser um diferencial competitivo.

2.2. Descrição do Problema

A ausência de um sistema centralizado e eficiente para a gestão de pedidos dificulta o acompanhamento em tempo real, atrasa o envio da mercadoria, compromete o atendimento ao cliente, e gera inconsistências nos processos de venda, separação e entrega. Isso leva à desorganização, custos adicionais, retrabalho e até à perda de vendas.

Sem controle automatizado, a empresa perde a visibilidade do fluxo de pedidos, o que dificulta a identificação de falhas, o planejamento de recursos e o atendimento ágil e preciso às necessidades dos clientes.

2.3 Relevância

Resolver esse problema é essencial para melhorar a produtividade, reduzir falhas operacionais e aumentar a satisfação do cliente.

Um sistema de gestão de pedidos bem estruturado proporciona maior controle sobre o fluxo de vendas, estoque e entregas, tornando o processo mais ágil e confiável. Os funcionários trabalham mais seguros e satisfeitos, pois confiam no processo e a sociedade ganha serviços mais eficientes, melhores experiências de compra e fortalecimento da economia local, com pequenas e médias empresas mais competitivas.

2.4. Impacto

De acordo com a pesquisa “Dores dos pequenos negócios” do Sebrae (2023), cerca de 42% dos pequenos negócios enfrentam dificuldades relacionadas à organização operacional, sendo a gestão de pedidos apontada como um dos principais gargalos. Esse cenário reforça a importância de soluções tecnológicas que integrem os processos e reduzam falhas manuais.

Além disso, observa-se que muitos microempreendedores ainda utilizam métodos manuais, como planilhas e anotações, para controle de pedidos, o que aumenta as chances de erro e retrabalho.

Com base nessas informações e em levantamentos preliminares sobre o comportamento do público-alvo, espera-se que a implantação de um sistema automatizado proporcione impactos positivos, como a redução de erros operacionais, o aumento da agilidade nas entregas e a melhoria da satisfação dos clientes.

Para validar esses dados de forma mais precisa, foram planejadas entrevistas com microempreendedores locais, mas não foi possível aplicá-las dentro do prazo do projeto. As necessidades foram inferidas com base em referências de mercado.

2.5 Fontes de Pesquisa

- Sebrae (2023) – Relatório “*Desafios da Gestão nos Pequenos Negócios*”, que apresenta dados sobre as principais dificuldades enfrentadas por micro e pequenas empresas em seus processos operacionais.
- Artigo da Endeavor Brasil (2022) – “*A importância da automação nos processos de venda*”, que aborda os benefícios da digitalização para a eficiência e o controle de operações comerciais.
- Entrevistas com microempreendedores locais – previstas para as próximas etapas do projeto, com o objetivo de levantar informações qualitativas sobre os principais desafios e expectativas em relação à gestão de pedidos.
- Feedbacks do cliente solicitante – coletados durante o processo de compreensão inicial da demanda, utilizados para direcionar o escopo e as funcionalidades esperadas na solução.

2.6. Justificativa

Com base nas informações levantadas, a equipe considera o problema relevante, recorrente e viável de ser solucionado com as tecnologias estudadas atualmente (por exemplo: desenvolvimento web, banco de dados, integração de APIs).

Acreditamos ser possível criar uma solução simples e acessível, com interface amigável, capaz de centralizar pedidos e automatizar tarefas, como controle de status, envio de notificações e geração de relatórios.

Essa proposta será validada e ajustada ao longo do desenvolvimento, mas já indica uma boa aderência às necessidades dos usuários mapeadas anteriormente.

3. Necessidades do Cliente/Usuário e Validação

A etapa de identificação das necessidades do cliente é essencial para o desenvolvimento de soluções eficazes, capazes de promover maior satisfação e fidelização. Para Kotler (2004), a solução de problemas consiste em saber ouvir e responder às questões dos clientes a fim de identificar necessidades e sugerir soluções seguras. Esse processo, se bem conduzido, contribui para a otimização de investimentos e recursos, prevenindo a criação de produtos que não atendam às expectativas do público-alvo.

A etapa de validação, por sua vez, tem como finalidade assegurar que as necessidades foram corretamente compreendidas tornando a solução proposta

adequada à demanda, permitindo ajustes no decorrer do desenvolvimento da solução, tornando-se portanto, um fator determinante para o sucesso do projeto.

Conforme explicado pelo cliente, a demanda visa solucionar problemas relacionados ao processo de gestão de pedidos, com ênfase em melhorar o controle e a eficiência desde o momento da entrada do pedido até a sua entrega ao cliente. Moraes (2015), afirma que “os sistemas devem ser projetados de maneira que as pessoas possam controlá-los, entendê-los e utilizá-los sem riscos de falhas ou de informações incorretas”. A afirmação corrobora com o objetivo de tornar as operações mais organizadas e reduzir a margem de erro que tem impactado a satisfação do cliente e a eficiência operacional.

3.1 Perfil do Cliente/Usuário

Identificar o perfil dos usuários ajuda a empresa a ser mais assertiva na solução e a se comunicar melhor com o seu público a fim de alinhar as reais demandas que a solução deve apresentar.

Neste projeto buscamos aprimorar a gestão de pedidos de pequenas e médias empresas, a fim de contribuir para uma comunicação mais eficiente entre setores e com os clientes.

Dentro da organização, os usuários mais impactados pelo processo de gestão de pedidos são os que atuam no estoque, expedição e vendas, uma vez que dependem diariamente de informações precisas e atualizadas sobre a disponibilidade de itens para o correto desempenho de suas atividades. O setor de compras também é diretamente influenciado, pois necessita de dados confiáveis sobre os saldos de estoque para planejar e executar a reposição de produtos de forma eficaz. Além disso, as áreas de gestão e financeira utilizam esses dados para elaborar relatórios gerenciais e definir estratégias de melhoria e investimento. Por fim, os clientes são beneficiados pela comunicação eficiente sobre o andamento de seus pedidos, o que contribui para maior transparência e satisfação no relacionamento com a empresa.

3.2 Necessidades Identificadas

- Necessidade 1 - agilidade no registro de pedidos: inserir, consultar e atualizar informações de forma ágil, otimizando o tempo de processamento e organização dos pedidos, validação realizada com base em análise

documental e referências teóricas (Sebrae, Endeavor) complementadas por discussões internas da equipe.

- Necessidade 2 - acompanhamento das vendas: a integração dos dados permitirá analisar as vendas no momento em que elas estão ocorrendo, validada por meio de estudos de casos semelhantes e análise de boas práticas de mercado, aliadas aos feedbacks internos da equipe.
- Necessidade 3 - redução de erros: um controle de estoque *realtime* evita erros e aumenta a confiabilidade nos dados movimentados, garantindo estoque para entrega e dando segurança aos funcionários na venda e separação dos itens, evitando erros e cancelamentos, validação baseada em referências técnicas e levantamento teórico de casos similares (Sebrae, Endeavor), além de observações discutidas pela equipe.
- Necessidade 4 - satisfação dos clientes: a automatização dos pedidos permite a notificação automática aos setores envolvidos na venda, mesmo os secundários como financeiro, além de notificar o cliente sobre o status da entrega do seu produto. Esses fatores asseguram os envolvidos no processo sobre cada etapa, além de fidelizar o cliente demonstrando agilidade e comprometimento com a demanda gerada, validação fundamentada em análise de literatura sobre gestão de atendimento ao cliente e reflexões internas da equipe sobre as práticas atuais do mercado.
- Necessidade 5 - organização: a integração dos dados no sistema facilita a organização e controle de todo o processo, validação obtida a partir da análise documental de relatórios institucionais (Sebrae) e da avaliação conjunta da equipe quanto às dificuldades observadas em processos manuais.

3.3 Método de Validação

Com o objetivo de agilizar a compreensão das reais necessidades dos usuários e reduzir os custos decorrentes de um deslocamento inviável neste momento do projeto, foi utilizado um roteiro de levantamento de requisitos elaborado internamente. Esse instrumento permitiu compreender e integrar as expectativas da contratante com sua percepção sobre as necessidades dos usuários.

Além de minimizar custos logísticos, a utilização desse roteiro interno possibilitou que a equipe refletisse e descrevesse com precisão as demandas

identificadas, além de integrar diferentes pontos de vista sobre o processo. Novas versões desse roteiro poderão ser aplicadas ao longo do desenvolvimento do sistema, a fim de alinhar e refinar continuamente as expectativas dos usuários

3.4 Conclusões Obtidas

Conclui-se que a identificação das necessidades dos clientes e a validação das informações obtidas constituem fatores essenciais para o sucesso do projeto. A utilização do roteiro interno de levantamento de requisitos contribuiu para a organização das prioridades e orientou as etapas necessárias à aprovação do primeiro protótipo.

Dessa forma, é possível reunir informações suficientes para definir o escopo inicial do sistema e dimensionar os principais elementos das funcionalidades requeridas para atender à demanda.

Entretanto, novos alinhamentos entre a equipe de desenvolvimento e o cliente serão realizados ao longo da construção do software, com o objetivo de garantir o ajuste contínuo das expectativas.

3.5 Revisão das etapas anteriores

O projeto Organização da Gestão de Pedidos surgiu da necessidade de aprimorar os processos de controle e acompanhamento de pedidos em pequenas e médias empresas, que em sua maioria ainda realizam essas atividades de forma manual, sem automatização ou controle eficiente das operações.

As etapas anteriores abordaram:

Etapas 1/4 – Planejamento: Foi formada uma equipe multidisciplinar, com integrantes responsáveis por desenvolvimento, design, comunicação e documentação. Nessa fase foram definidos o tema do projeto, os objetivos, as funções de cada membro e o cronograma de execução, estabelecendo uma base organizada para o andamento das próximas fases.

Etapas 2/4 – Problema e Contexto: O grupo identificou que muitas pequenas e médias empresas ainda realizam o controle de pedidos de forma manual, o que gera atrasos, erros e retrabalho. A ausência de um sistema centralizado dificulta o acompanhamento das etapas e a comunicação entre setores. Fontes como o Sebrae (2023) e a Endeavor Brasil (2022) reforçaram a importância de adotar

soluções tecnológicas acessíveis que otimizem o processo e aumentem a competitividade dessas empresas.

Etapas 3/4 – Necessidades do Cliente e Validação: Para compreender melhor as demandas do cliente, a equipe realizou análises e conversas diretas que possibilitaram identificar as principais necessidades operacionais e alinhar as expectativas do projeto. A partir desse entendimento foram destacados cinco pontos prioritários: agilidade no registro de pedidos, acompanhamento das vendas em tempo real, redução de erros, melhoria da satisfação do cliente e organização dos processos internos. Essas necessidades orientaram a definição do escopo funcional e dos objetivos da solução.

A partir dessas informações, foi estruturada uma proposta de solução tecnológica que centraliza as etapas do pedido em um único sistema digital.

4. Tecnologias Escolhidas e Justificativa

A tecnologia escolhida para o desenvolvimento de qualquer solução, precisa atender aos requisitos, objetivos e restrições específicas do projeto, mantendo-se viável economicamente e escalável para crescimento futuro.

As escolhas tecnológicas apresentadas a seguir consideram o conhecimento da equipe de desenvolvimento, a robustez das ferramentas, o custo-benefício e a capacidade de atender tanto à demanda atual quanto às projeções de crescimento do negócio

4.1 Frontend

Para a interface do usuário, optou-se pela stack moderna composta por TypeScript, React e Tailwind CSS. React é a biblioteca JavaScript mais popular para construção de interfaces, oferecendo componentização, reatividade eficiente através do Virtual DOM e um vasto ecossistema com grande comunidade ativa. A adição do TypeScript traz tipagem estática ao projeto, reduzindo significativamente erros em tempo de desenvolvimento e facilitando a manutenção do código, além de prevenir problemas de integração entre frontend e backend.

O Tailwind CSS complementa a stack como framework CSS utilitário, acelerando o desenvolvimento da interface e garantindo consistência visual e responsividade sem a necessidade de escrever CSS customizado extenso. Esta escolha também abre possibilidades futuras de expansão para aplicativos mobile

através do React Native, reduzindo custos e tempo de desenvolvimento nessa eventual migração.

4.2 Backend

O backend será desenvolvido utilizando Java com Spring Boot, uma das stacks mais consolidadas para aplicações corporativas. Esta escolha se justifica por múltiplas razões técnicas e estratégicas. Primeiramente, Java é multiplataforma e compatível com qualquer sistema operacional, oferecendo segurança robusta através de mecanismos de proteção integrados e amplamente testados pela indústria. A JVM (Java Virtual Machine) otimiza continuamente a execução do código, garantindo alta performance mesmo sob carga elevada.

A capacidade de escalabilidade do Java é outro fator decisivo, permitindo suportar grande volume de usuários e transações simultâneas conforme o negócio cresce. Ainda, por ser uma linguagem orientada a objetos, é fácil a reutilização de código e manutenção, enquanto a tipagem forte previne erros em integrações e lógica de negócio complexa. Já o Spring Boot oferece soluções prontas para autenticação, segurança, persistência e integração, fato que acelera o desenvolvimento. Por fim, a comunidade ampla e ativa garante documentação extensa e suporte facilitado para resolução de problemas.

Como tecnologia complementar, está prevista a utilização de Python com FastAPI ou Flask para funcionalidades específicas relacionadas ao processamento e análise de dados, uma vez que esta linguagem possui as principais ferramentas usadas nessa área que poderão ser aproveitadas conforme a complexidade das funcionalidades de análise se torne mais clara durante o desenvolvimento. Ambas as linguagens são de domínio da equipe, garantindo produtividade desde o início do projeto.

Outra tecnologia utilizada será Docker que servirá como ferramenta para containerização do backend, garantindo consistência entre os ambientes de desenvolvimento, teste e produção. Através de containers, é possível empacotar a aplicação Java/Spring Boot junto com todas suas dependências, bibliotecas e configurações necessárias, eliminando o problema de "na minha máquina funciona". Esta abordagem facilita significativamente o trabalho da equipe, pois todos os desenvolvedores trabalharão com ambientes idênticos independentemente de seus sistemas operacionais.

Durante o desenvolvimento local, o Docker Compose será utilizado para orquestrar múltiplos containers simultaneamente. Esta configuração simula um ambiente de produção, permitindo identificar e corrigir problemas de integração precocemente.

4.3 Banco de Dados

O PostgreSQL foi escolhido como sistema de gerenciamento de banco de dados principal por ser uma solução gratuita, open-source, robusta e confiável para dados transacionais críticos. O sistema também oferece excelente suporte a relacionamentos complexos, essencial para modelar adequadamente as entidades do negócio como clientes, pedidos, itens e produtos. A integridade referencial forte garante consistência dos dados e previne inconsistências como pedidos sem clientes ou itens sem produtos associados.

Além do desempenho escalável adequado para aplicações de médio a grande porte, o PostgreSQL oferece recursos avançados como suporte nativo a JSON para flexibilidade futura e consultas complexas, além de compatibilidade nativa com Azure através do Azure Database for PostgreSQL.

Por fim, durante a fase de desenvolvimento e testes, será utilizado o banco H2 em memória, que facilita a execução de testes automatizados e prototipagem rápida sem necessidade de infraestrutura externa.

4.4 Hospedagem/Deploy

A estratégia de hospedagem foi definida buscando simplicidade, custo inicial reduzido e aproveitamento do conhecimento pré existente da equipe. Para backend e banco de dados, o Microsoft Azure foi selecionado como provedor cloud principal devido ao conhecimento prévio da equipe com a plataforma e sua curva de aprendizado menos íngreme comparada à AWS. A Azure oferece tiers gratuitos generosos para desenvolvimento e testes, integração nativa com PostgreSQL através do Azure Database for PostgreSQL e suporte robusto para containerização com Azure Container Apps.

A estratégia de desenvolvimento seguirá uma abordagem gradual, iniciando com desenvolvimento local usando Docker Compose para simular ambiente de produção, seguido de testes de integração no Azure Container Apps e deploy em produção aproveitando o PostgreSQL Serverless. Esta abordagem permite utilizar

os tiers gratuitos para minimizar custos iniciais enquanto garante um ambiente de desenvolvimento próximo ao de produção.

Para a hospedagem da aplicação React/TypeScript, que gera conteúdo estático, serão avaliadas três plataformas especializadas em Jamstack: Cloudflare Pages, Vercel e Netlify. Todas oferecem recursos semelhantes, ideal para MVP e fase inicial:

- Todas as plataformas oferecem um plano gratuito generoso, o que é essencial para a fase de desenvolvimento do MVP e para a operação inicial de uma micro/pequena empresa.
- As três candidatas têm integração direta com o GitHub, o que permite que as atualizações do código sejam automaticamente construídas e implantadas (CI/CD), acelerando o ciclo de desenvolvimento da equipe.
- Todas utilizam uma extensa Rede de Distribuição de Conteúdo (CDN) global, garantindo que o site seja carregado rapidamente em qualquer localidade, o que melhora a experiência do usuário (UX) e a satisfação do cliente.
- SSL/HTTPS é incluído gratuitamente em todas as plataformas, protegendo a comunicação com os usuários.

A escolha final será baseada na preferência da equipe e melhor integração com o backend no Azure, embora todas sejam plenamente compatíveis.

4.5 Outras Ferramentas Essenciais

O desenvolvimento do projeto será apoiado por um conjunto de softwares para gerenciamento, comunicação e monitoramento. Concatenamos aqui, algumas opções sugeridas.

4.5.1 Gerenciamento de Projeto:

- Git: será usado para versionamento do código, sendo indispensável para a organização do projeto.
- Github: hospedagem do repositório do projeto que conterá o código-fonte, protótipo, referências que eventualmente serão usadas, fluxogramas, documentação, entre outros.
- Trello ou Notion: usados para organizar as tarefas de desenvolvimento.

- Figma: prototipagem do frontend.
- Drawio: criação dos fluxogramas.
- Vs Code: editor de código-fonte que se comporta como IDE (Ambiente de Desenvolvimento Integrado) será usado para o desenvolvimento tanto do frontend quanto do backend.

4.5.2 Autenticação:

- Spring Security com JWT: garante uma solução robusta com tokens para sessões stateless.

4.5.3 Notificações:

- JavaMail API: permitirá a comunicação com usuários através de e-mails transacionais (confirmação de pedidos e recuperação de senha) utilizando Gmail SMTP, além de ser integrada ao Spring Boot.
- Twilio ou WhatsApp Business API ou Evolution API (alternativa econômica): irá notificar clientes via SMS/WhatsApp (pode ser considerado para investimento futuro).

4.5.4 Monitoramento:

- Sentry: detecção de erros na produção, aproveitando uma opção free tier generosa.
- Google Analytics: entender o uso do sistema e comportamento dos usuários e diversas outras possibilidades.

4.5.5 Documentação da API:

- Swagger/OpenAPI: automatiza documentação da API, mantendo sincronização constante com o código-fonte e facilitando testes e integração.

4.5.6 Estimativa de Custos Mensais

Na fase inicial (de 0 a 6 meses), estimamos uma operação com volume de 5 a 30 pedidos diários, gerando um custo operacional mínimo. A hospedagem de backend e banco de dados no Azure, assim como a hospedagem do frontend nas plataformas Jamstack, utilizam tiers gratuitos suficientes para este volume. Quanto ao serviço de email através de Gmail SMTP, também é gratuito para este volume de

transações. O único custo fixo será o domínio personalizado, que custa aproximadamente R\$40,00 por ano, resultando em um investimento mensal entre R\$3,00 a R\$5,00.

Conforme o negócio for crescendo e se consolidado no mercado, é esperado um aumento na demanda. Estimamos que o volume de pedidos fique entre 50 e 150 pedidos diários. Assim, os custos se ajustarão proporcionalmente.

Nesta fase de crescimento para pequena empresa, estima-se que a hospedagem de backend e banco de dados no Azure demandará entre R\$ 80 e R\$ 150 mensais, enquanto a hospedagem do frontend poderá permanecer no tier gratuito ou migrar para plano pago conforme necessidade de recursos adicionais (cerca de R\$ 50-100 mensais). Já as ferramentas de monitoramento básico permanecerão gratuitas, com possibilidade de upgrade futuro.

Assim, o investimento total nesta fase de crescimento ficará entre R\$100,00 e R\$300,00 mensais, valores adequados ao faturamento esperado de uma pequena empresa em expansão. A arquitetura escolhida permite escalar gradualmente, pagando apenas pelos recursos efetivamente utilizados e mantendo os custos proporcionais ao crescimento real do negócio.

5. Diagrama e Descrição da Solução Proposta

5.1 Descrição Geral da Solução

A arquitetura do sistema foi projetada seguindo o modelo cliente-servidor, separando claramente as responsabilidades entre apresentação, lógica de negócio e persistência de dados. Esta separação garante a facilidade da manutenção, escalabilidade e facilita futuras expansões do sistema.

O frontend constitui a interface gráfica acessível através de navegadores web, permitindo que clientes realizem e acompanhem seus pedidos, enquanto funcionários e proprietários dos estabelecimentos gerenciam todas as operações do negócio. Desenvolvido em TypeScript com React e Tailwind CSS, a aplicação oferece interface totalmente responsiva, adaptando-se perfeitamente a diferentes tamanhos de tela e possibilitando acesso completo através de smartphones e tablets sem perda de funcionalidades. A escolha do React garante atualizações rápidas da interface e experiência fluida para o usuário, enquanto o TypeScript adiciona segurança através da tipagem estática.

O backend concentra toda a lógica de negócio e serviços essenciais do sistema, funcionando como uma API RESTful que recebe requisições do frontend e processa toda a inteligência da aplicação. Desenvolvido em Java com Spring Boot, gerencia autenticação e controle de permissões dos usuários, processamento completo de pedidos, gestão de produtos e estoque, integração com sistemas de pagamento, geração de notas fiscais, produção de relatórios gerenciais e administrativos, envio de notificações por email, e toda a comunicação com o banco de dados para operações de criação, leitura, atualização e exclusão de informações. A documentação da API é gerada automaticamente através do Swagger/OpenAPI, facilitando integrações e manutenção.

O sistema prevê integrações com serviços de terceiros para funcionalidades especializadas, incluindo gateways de pagamento para processamento seguro de transações financeiras e sistemas de emissão de notas fiscais eletrônicas conforme legislação vigente. Estas integrações seguem padrões de mercado e utilizam APIs REST documentadas, garantindo confiabilidade e conformidade legal.

Complementando a arquitetura principal, está previsto o desenvolvimento de um microserviço especializado em Python utilizando FastAPI para análise de dados e geração de insights estratégicos. Este serviço funcionará de forma independente, comunicando-se com o backend Java através de APIs REST, e aproveitará as bibliotecas Python para ciência de dados, incluindo Pandas para manipulação de dados, NumPy para cálculos numéricos, e Matplotlib ou Plotly para visualizações. O microserviço poderá processar grandes volumes de dados históricos de pedidos, identificar padrões de consumo, prever demandas futuras de produtos, analisar sazonalidades, calcular métricas de performance do negócio como ticket médio e taxa de recompra e gerar relatórios analíticos avançados que auxiliem na tomada de decisões estratégicas.

A arquitetura de microserviços permite que este componente escale independentemente conforme o volume de análises cresce, seja deployado separadamente no Azure Container Apps e não impacte a performance do sistema transacional principal. Esta separação de responsabilidades também facilita a manutenção, permitindo que a equipe especialize-se em diferentes tecnologias conforme necessário e possibilita futuras expansões.

O banco de dados PostgreSQL armazena de forma estruturada e segura todas as informações transacionais e cadastrais do sistema, incluindo dados de

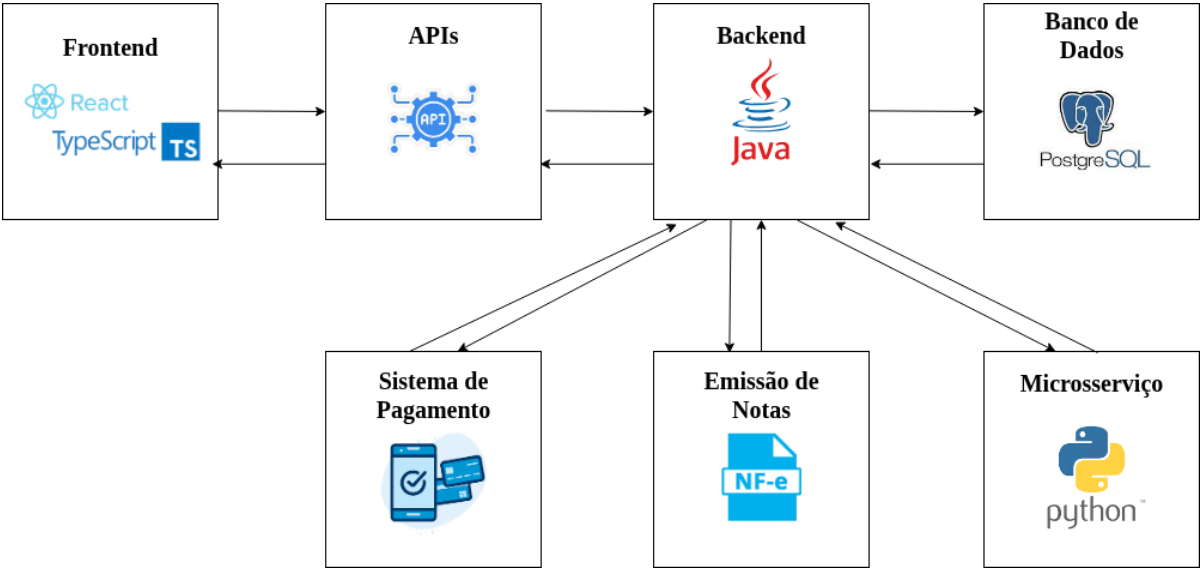
clientes, produtos, pedidos, usuários e configurações. A integridade referencial garante consistência entre as entidades relacionadas, prevenindo inconsistências como pedidos órfãos ou produtos sem categoria. Ainda, o PostgreSQL oferece recursos avançados de consulta, índices para otimização de performance e suporte a transações ACID, essenciais para operações comerciais confiáveis.

A infraestrutura de deploy distribui os componentes estrategicamente para otimizar performance e custos. O backend e banco de dados são hospedados no Microsoft Azure, aproveitando Azure Container Apps para containerização via Docker e Azure Database for PostgreSQL para gerenciamento do banco. O frontend é distribuído através de CDN global em plataforma Jamstack (Vercel, Netlify ou Cloudflare Pages), garantindo carregamento rápido em qualquer localidade geográfica e alta disponibilidade.

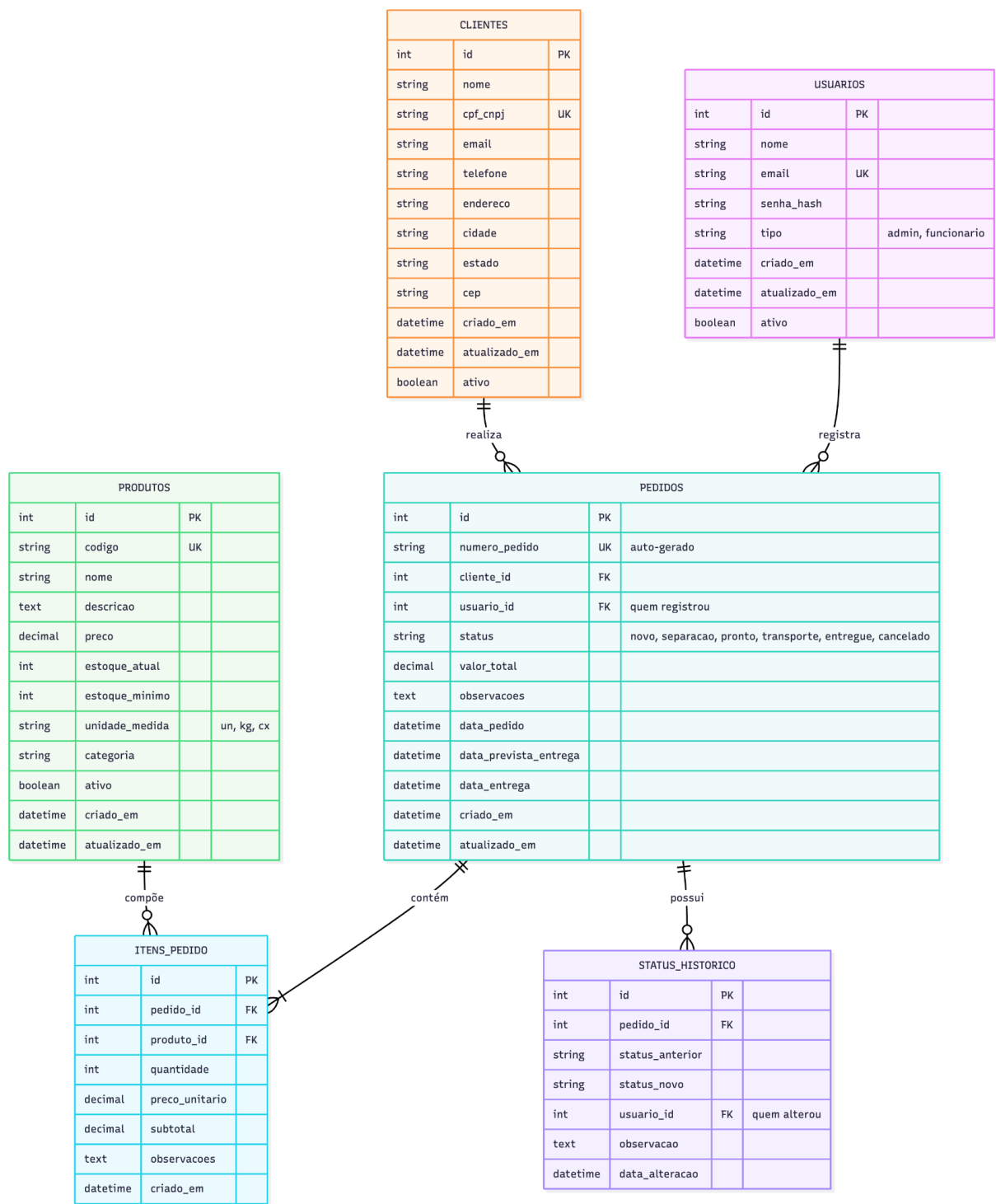
O monitoramento e a observabilidade são garantidos através de ferramentas especializadas integradas à stack. O Sentry, com seus SDKs nativos para Java/Spring Boot e React/TypeScript, realiza detecção e rastreamento automatizado de erros e exceções tanto no backend quanto no frontend. O Google Analytics, integrado à aplicação React, permite rastrear jornadas de usuários, páginas mais acessadas e métricas de engajamento essenciais para decisões estratégicas sobre o produto.

Esta arquitetura foi escolhida por equilibrar robustez técnica, os custos especialmente na fase inicial, a facilidade de manutenção através de tecnologias amplamente documentadas e capacidade de crescimento conforme a demanda do negócio. A separação clara de responsabilidades permite que cada componente evolua independentemente, facilitando atualizações e correções sem impactar todo o sistema.

5.2 Diagrama de Arquitetura



5.3 Protótipo da Interface



5. Conclusão, Resultados Alcançados, Dificuldades, Aprendizados e Perspectivas Futuras

5.1 Conclusão

O projeto Organização da Gestão de Pedidos teve como principal objetivo desenvolver uma proposta tecnológica capaz de otimizar o controle de pedidos em pequenas e médias empresas, substituindo processos manuais por um sistema automatizado e integrado.

Através da análise do problema, levantamento das necessidades do cliente e definição das tecnologias adequadas, foi possível estruturar uma solução que visa aumentar a eficiência operacional, reduzir erros e melhorar a satisfação do cliente.

A iniciativa demonstra que a digitalização dos processos é não apenas viável, mas essencial para a competitividade das empresas, sobretudo diante do crescimento das vendas digitais e da necessidade de integração entre canais.

5.2 Resultados Alcançados

Durante as etapas de levantamento e validação, foram alcançados resultados significativos, entre eles:

- Compreensão aprofundada das necessidades dos usuários por meio da análise documental, referências teóricas (Sebrae, Endeavor) e discussões internas da equipe, permitindo definir requisitos mais assertivos.
- Estruturação de uma proposta de sistema centralizado, com funcionalidades de controle de pedidos, acompanhamento de vendas em tempo real, notificações automáticas e integração com estoque.
- Seleção de tecnologias modernas e acessíveis, adequadas à realidade de micro e pequenas empresas, como React, Spring Boot e PostgreSQL, garantindo escalabilidade e baixo custo inicial.
- Planejamento detalhado de implantação e hospedagem, prevendo uso de tiers gratuitos para testes e migração futura para planos pagos conforme o crescimento da demanda.

Estes resultados consolidam uma base sólida para a construção do protótipo e futura implementação da solução.

5.3 Dificuldades Encontradas

Durante o desenvolvimento das etapas até o momento, algumas dificuldades foram observadas:

- Limitação de tempo e recursos para realizar entrevistas presenciais com empreendedores, o que levou a equipe a adaptar a coleta de dados por meio

de um roteiro interno de levantamento de requisitos, elaborado para substituir interações diretas neste estágio do projeto.

- Complexidade na definição da arquitetura tecnológica, considerando o equilíbrio entre robustez, custo e curva de aprendizado da equipe.
- Desafios na integração entre áreas de conhecimento distintas, exigindo colaboração entre membros com perfis técnicos e administrativos.

Apesar disso, tais dificuldades contribuíram para fortalecer o trabalho em equipe e aprimorar o planejamento das próximas fases.

5.4 Aprendizados

O desenvolvimento do projeto proporcionou aprendizados relevantes, como:

- A importância de compreender profundamente as necessidades do usuário antes de iniciar o desenvolvimento técnico, evitando retrabalho e desalinhamento de expectativas.
- O valor da validação contínua junto ao cliente, reforçando a ideia de que o desenvolvimento deve ser interativo e colaborativo.
- O reconhecimento de que a escolha de tecnologias deve equilibrar inovação e acessibilidade, priorizando soluções que possam ser mantidas e ampliadas no futuro.
- A percepção de que a gestão de projetos e comunicação entre equipes é tão importante quanto o domínio técnico para o sucesso de uma solução aplicada.

5.5 Perspectivas Futuras e Próximas Etapas

Para as próximas UCs e fases do projeto, estão previstas as seguintes ações:

- Desenvolvimento do protótipo funcional com base nas tecnologias definidas, priorizando a interface de cadastro e acompanhamento de pedidos.
- Integração do sistema com o banco de dados PostgreSQL, garantindo persistência e consistência das informações.
- Implementação de autenticação e notificações automáticas, ampliando a segurança e comunicação do sistema.
- Testes com usuários reais (microempreendedores locais) para avaliar a usabilidade e efetividade da solução.

- Aprimoramento do design e da experiência do usuário (UX/UI) utilizando ferramentas como Figma e Tailwind CSS.
- Documentação técnica completa e plano de implantação, para que o sistema possa ser facilmente escalado ou comercializado no futuro.

6. Referências

Endeavor Brasil (2022) – “A importância da automação nos processos de venda”.

KOTLER, Philip. Marketing essencial: conceitos, estratégias e casos. 2. ed. São Paulo, SP: Pearson, 2004. E-book. Disponível em: <https://plataforma.bvirtual.com.br>. Acesso em: 18 out 2025.

MORAIS, Roberto Ramos de. Logística empresarial. 1. ed. Curitiba: Intersaberes, 2015. E-book. Disponível em: <https://plataforma.bvirtual.com.br>. Acesso em: 18 out 2025.

Sebrae. Relatório “Dores dos pequenos negócios”. São Paulo, 2023.