

Grupo vega - Mesa Fácil

VISÃO DO PRODUTO E DO PROJETO

Versão [\[1.1\]](#)

Tabela - Integrantes do Grupo:

Mat.	Nome	Função (responsabilidade)
211031468	Pedro Victor Salerno Martins	Desenvolvimento Back-end
211030620	Patrick Anderson Carvalho dos Santos	Scrum Master
170146243	João Lucas Costa Vale	Desenvolvimento Front-end
180100271	Emivalto da costa tavares junior	Gerente de Testes

Histórico de Revisões

Data	Versão	Descrição	Autor
10/05/2024	1.0	Revisão de informações, inserção de diagramas, junção dos textos dos colaboradores de equipe. Junção de informações para validação na próxima reunião.	Patrick Anderson Carvalho dos Santos
11/05/2024	1.1	Atualização de informações e inserção de recomendações do professor de acordo com a reunião.	

Sumário

1	<i>VISÃO GERAL DO PRODUTO</i>	3
1.1	Problema	3
1.2	Declaração de Posição do Produto	3
1.3	Objetivos do Produto	4
1.4	Tecnologias a Serem Utilizadas	4
2	<i>VISÃO GERAL DO PROJETO</i>	4
2.1	Ciclo de vida do projeto de desenvolvimento de software	4
2.2	Organização do Projeto	4
2.3	Planejamento das Fases e/ou Iterações do Projeto	4
2.4	Matriz de Comunicação	5
2.5	Gerenciamento de Riscos	5
2.6	Critérios de Replanejamento	5
3	<i>PROCESSO DE DESENVOLVIMENTO DE SOFTWARE</i>	6
4	<i>DECLARAÇÃO DE ESCOPO DO PROJETO</i>	6
4.1	Backlog do produto	6
4.2	Perfis	6

4.3	Cenários.....	6
4.4	Tabela de Backlog do produto.....	7
5	<i>REFERÊNCIAS BIBLIOGRÁFICAS</i>.....	8

VISÃO DO PRODUTO E PROJETO4

1 VISÃO GERAL DO PRODUTO

1.1 Problema

PARA MELHOR VISUALIZAÇÃO: <https://mm.tt/app/map/3230938191?t=uhP9sTWNgl>



Tendo em vista que o mercado de restaurantes e estabelecimentos alimentícios em geral costumam ter requisitos funcionais muito variáveis, a metodologia ágil se demonstra muito compatível com as necessidades do mercado de software no ramo alimentício. Dessa forma, foi desenvolvido um sistema capaz de suprir as necessidades gerais de um restaurante, tendo em mente o potencial adaptativo da metodologia ágil.

O sistema se propõe a apresentar o cardápio do restaurante, realizar pedidos, que são agrupados pela entidade garçom e repassados para a cozinha, onde se recebe o pedido. Assim, o pedido pronto é retornado pela cozinha, onde são agrupados pela entidade garçom e entregue as mesas.

Logo, o sistema tem como objetivo resolver o problema de gestão de fluxo das mesas, e por consequência, seus pedidos e comandas.

1.2 Declaração de Posição do Produto

O sistema se difere dos demais do mercado por evitar uma aproximação genérica e uniforme para abordar os problemas apresentados por um ramo tão diverso como o ramo de restaurantes. A metodologia ágil possibilita que seja possível confeccionar um produto com alta customização. Assim, ao adquirir o produto, o cliente pode solicitar mudanças estruturais para fazer com que seus clientes tenham uma experiência melhor, e seus funcionários consigam operar de forma mais eficiente e eficaz.

Para:	Clientes e gestores de restaurantes
Necessidade:	Gerir e integrar a logística de controle das mesas
O (nome do produto):	Mesa fácil
Que:	É de fácil implementação, e altamente customizável
Ao contrário:	Aplicativos genéricos e uniformes
Nosso produto:	Apresenta uma experiência única para clientes e simples para funcionários.

1.3 Objetivos do Produto

O produto é capaz de realizar o registro dos pedidos do restaurante, além de integrar as partes necessárias para que a experiência do cliente seja entregue. Ao utilizar a metodologia ágil, esse sistema abre a possibilidade de alterações para se adequar melhor às necessidades de cada cliente, sendo um possível apresentar um produto único para cada cliente.

1.4 Tecnologias a Serem Utilizadas

A escolha de tecnologias adequadas é fundamental para o sucesso de qualquer projeto de software. Para o desenvolvimento do nosso sistema de gestão para restaurantes, utilizaremos um conjunto de tecnologias modernas e eficazes para as partes de front-end e back-end, juntamente com metodologias ágeis para garantir um processo de desenvolvimento adaptativo e iterativo.

Linguagens de Programação

Front-End:

As tecnologias de front-end, como HTML, CSS e JavaScript, são essenciais para a construção de interfaces de usuário interativas ([MEYER, 2020](#); [FLANAGAN, 2020](#)).

HTML: Será utilizado para construir a estrutura básica das páginas web do nosso sistema. HTML é essencial para organizar o conteúdo e os elementos visuais que os usuários interagem diretamente.

CSS: Com CSS, utilizaremos as páginas web para oferecer uma interface atraente e responsiva. CSS é crucial para garantir que nosso aplicativo seja visualmente apelativo e fácil de usar em diferentes dispositivos e tamanhos de tela.

JavaScript: JavaScript será usado para criar interações dinâmicas nas páginas web, melhorando a experiência do usuário ao permitir respostas rápidas a suas ações sem necessidade de recarregar a página.

Flutter: Embora tradicionalmente usado para desenvolvimento móvel, optamos por utilizar Flutter

também para o front-end para aproveitar sua capacidade de criar interfaces de usuário ricas e customizáveis tanto para web quanto para aplicativos móveis, garantindo uma experiência consistente em todas as plataformas.

Back-End:

MySQL: Escolhemos MySQL como nosso sistema de gerenciamento de banco de dados devido à sua robustez, confiabilidade e facilidade de uso. MySQL será utilizado para armazenar e gerenciar dados relacionados a pedidos, reservas, gestão de usuários e outros dados operacionais críticos.

Python: Python será utilizado para o desenvolvimento do servidor devido à sua clara e poderosa biblioteca. Python facilita a escrita de código limpo e legível, o que acelera o desenvolvimento e a manutenção. Além disso, a integração com frameworks web como Django permitirá construir aplicações seguras e escaláveis

Metodologias de Desenvolvimento:

Scrum (SCRUM): Scrum será adotado para gerenciar o projeto através de sprints curtos e reuniões regulares, como as dailies, sprint reviews e retrospectivas. Esta metodologia favorece a adaptabilidade e o feedback contínuo, essenciais para atender às necessidades em constante mudança de um ambiente de restaurante.

Extreme Programming (XP): Em conjunto com Scrum, o XP será utilizado para aprimorar as práticas de desenvolvimento de software. XP enfatiza a comunicação constante, feedback, simplicidade e coragem. Práticas como programação em pares e desenvolvimento orientado a testes garantirão a qualidade do código e a rapidez na entrega de funcionalidades.

2 VISÃO GERAL DO PROJETO

2.1 Ciclo de vida do projeto de desenvolvimento de software

O desenvolvimento e lançamento do aplicativo irá envolver várias etapas de entrega, como por exemplo, versão beta, versão de teste de interface e versão final.

Versão Beta

- **Testes de Usabilidade:** a versão beta terá que explorar o aplicativo para que o cliente possa fornecer feedback sobre de usabilidade e trazer experiência do usuário para aprimoramento.
- **Testes Funcionais:** Verificação das funcionalidades e correção de bugs.
- **Otimização de Desempenho:** Melhoria da velocidade e eficiência do aplicativo.

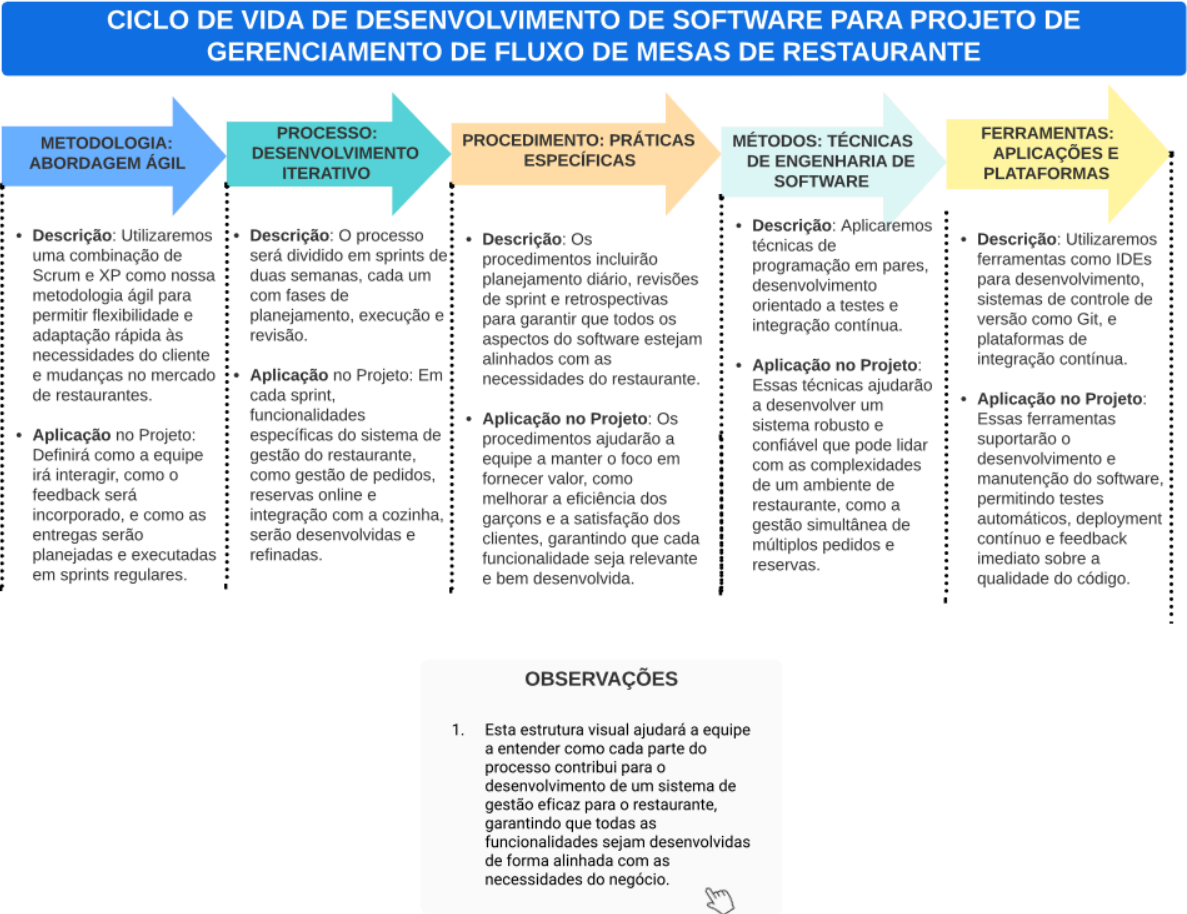
Segunda Versão de Teste

Com base no feedback da versão beta testers, a equipe faz ajustes e aprimoramentos. Uma segunda versão de teste é lançada para validar as mudanças. Além disso, o software contará com a implementação de algumas funcionalidades essenciais.

Entrega Final

A entrega final consiste em entregar o software pronto para ser validado pelo cliente e em seguida ser lançado

2.2 Organização do Projeto



Papel	Atribuições	Responsável	Participantes
Desenvolvedor	Codificar o produto, codificar testes unitários, realizar refatoração		
Dono do Produto	Atualizar o escopo do produto, organizar o escopo das sprints, validar as entregas		
Analista de Qualidade	Garantir a qualidade do produto, garantir o cumprimento do conceito de pronto, realizar inspeções de código		
Cliente			

2.3 Planejamento das Fases e/ou Iterações do Projeto

Sprint	Produto (Entrega)	Data Início	Data Fim	Entregável(eis)	Responsáveis	% conclusão
Sprint 1	Estrutura do site e página inicial funcional	01/07/2024	14/07/2024	Layout do site e navegação básica Página inicial com informações do restaurante e banner promocional	Patrick e Pedro	100%
Sprint 2	Implementação de	15/07/2024	28/07/2024	Sistema de	Pedro e	100%

	<i>funcionalidades essenciais</i>			reservas online com integração ao calendário do restaurante Menu digital com fotos e descrições dos pratos Página de "Sobre Nós" com a história do restaurante e equipe	Emivalto	
<i>Sprint 3</i>	<i>Adição de funcionalidades e integrações para aprimorar a experiência do usuário</i>	<i>29/07/2024</i>	<i>11/08/2024</i>	Sistema de pedidos online para garçons e chefs Página de blog com notícias e promoções do restaurante	<i>Patrick e João Lucas</i>	100%
<i>Sprint 4</i>	<i>Versão Beta completa do site do restaurante</i>	<i>12/08/2024</i>	<i>31/08/2024</i>	Testes de usabilidade e correção de bugs Treinamento da equipe do restaurante para utilização do site Lançamento da Versão Beta do site	<i>João Lucas e Emivalto</i>	100%

2.4 Matriz de Comunicação

<i>Descrição</i>	<i>Área/Envolvidos</i>	<i>Periodicidade</i>	<i>Produtos Gerados</i>
• Acompanhamento das Atividades em Andamento	• Equipe do Projeto	• Duas vezes por semana	• Relatório de situação do projeto
• Acompanhamento dos Riscos, Compromissos, Ações Pendentes, Indicadores	• Equipe do Projeto	• Quinzenal	
- Comunicar situação do projeto	• Equipe • Prof/Monitor	• Semanal	•

2.5 Gerenciamento de Riscos

Risco de falhas de comunicação entre a equipe de desenvolvimento e os stakeholders:

- **Grau de exposição:** Alto
- **Mitigação:**
 - Estabelecer canais de comunicação claros e frequentes com os stakeholders.
 - Definir um responsável pela comunicação entre a equipe e os stakeholders.
 - Realizar reuniões periódicas para apresentar o progresso do projeto e coletar feedback.
 - Utilizar ferramentas de comunicação online, como chats e videoconferências.
- **Plano de contingência:**
 - Em caso de falhas de comunicação, realizar uma reunião imediata para identificar o problema e buscar soluções.

- Documentar todas as decisões e comunicá-las aos stakeholders.
- Treinar a equipe em comunicação eficaz.

Previsões de adaptações frequentes às necessidades do usuário e do ambiente do restaurante:

- **Grau de exposição:** Médio
- **Mitigação:**
 - Adotar uma metodologia de desenvolvimento ágil que permita adaptações rápidas às mudanças.
 - Desenvolver um sistema modular que facilite a implementação de novas funcionalidades.
 - Realizar testes de usabilidade frequentes para garantir que o sistema atenda às necessidades dos usuários.
 - Manter um bom relacionamento com os stakeholders para entender suas necessidades e expectativas.
- **Plano de contingência:**
 - Se as adaptações forem muito frequentes, reavaliar o escopo do projeto e negociar novas entregas com os stakeholders.
 - Priorizar as adaptações mais importantes e adiar as menos urgentes.
 - Comunicar as mudanças aos stakeholders de forma clara e transparente.

Capacidade produtiva da equipe de desenvolvimento perante o prazo:

- **Grau de exposição:** Médio
- **Mitigação:**
 - Planejar o projeto de forma realista e levar em consideração a capacidade da equipe.
 - Monitorar o progresso do projeto e identificar possíveis atrasos.
 - Comunicar os riscos aos stakeholders e buscar soluções conjuntas.
 - Implementar ferramentas de gerenciamento de projetos para auxiliar no acompanhamento das tarefas.
- **Plano de contingência:**
 - Se a equipe estiver atrasada, reavaliar o cronograma do projeto e negociar novos prazos com os stakeholders.
 - Contratar recursos adicionais, se necessário.
 - Reduzir o escopo do projeto.

Conhecimento técnico da equipe de desenvolvimento:

- **Grau de exposição:** Baixo
- **Mitigação:**
 - Contratar profissionais qualificados para o projeto.
 - Oferecer treinamento à equipe sobre as tecnologias que serão utilizadas.
 - Incentivar a participação da equipe em eventos e workshops da área.
 - Criar um ambiente de aprendizado contínuo na empresa.
- **Plano de contingência:**
 - Se a equipe não tiver o conhecimento técnico necessário, buscar ajuda de especialistas externos.
 - Treinar a equipe sobre as novas tecnologias.
 - Reduzir o escopo do projeto.

2.6 Critérios de Replanejamento

O processo de replanejamento deve seguir as seguintes etapas:

1. Identificar a necessidade de replanejamento:

- A equipe de projeto deve monitorar o andamento do projeto e identificar se algum dos critérios de replanejamento foi atingido para atingir o êxito na execução de todas as etapas do projeto.

2. Reunir a equipe de stakeholders:

- *Uma vez que a necessidade de replanejamento tenha sido identificada, a equipe de projeto deve reunir-se com todos os stakeholders para discutir os motivos do replanejamento e as possíveis soluções, para resolver o quanto antes.*

3. Analisar as opções:

- *A equipe de projeto deve analisar as diferentes opções de replanejamento e selecionar a melhor opção para o projeto para ser realizado a melhor planejamento.*

4. Atualizar o plano de projeto:

- *O plano de projeto deve ser atualizado para refletir as mudanças feitas durante o processo de replanejamento.*

5. Comunicar as mudanças:

- *As mudanças no plano de projeto devem ser comunicadas a todos os stakeholders.*

6. Monitorar e controlar o projeto:

- *A equipe de projeto deve continuar a monitorar e controlar o projeto após o replanejamento para garantir que o projeto esteja no caminho certo para atingir seus objetivos.*

¹ Lembrando: plano de mitigação existe para tentar evitar que um risco ocorra. O plano de contingência é acionado se um risco efetivamente ocorre.

3 PROCESSO DE DESENVOLVIMENTO DE SOFTWARE

Metodologias Escolhidas

A equipe decidiu integrar as metodologias Scrum e XP para tirar proveito de suas respectivas forças em um ambiente que exige adaptabilidade e eficiência ([KRUG, 2014](#)) :

Scrum: Será utilizado para estruturar o desenvolvimento em sprints, facilitar o planejamento iterativo, a revisão e a adaptação contínua dos produtos desenvolvidos. Scrum ajuda a manter o projeto alinhado com as necessidades do cliente e permite uma gestão eficaz do fluxo de trabalho através de reuniões regulares e feedbacks constantes.

Extreme Programming (XP): Complementará o Scrum com suas práticas de desenvolvimento técnico, como programação em pares, desenvolvimento orientado a testes e integração contínua. XP é escolhido para melhorar a qualidade do software e a produtividade da equipe ao minimizar o código defeituoso e incentivar uma cultura de feedback técnico detalhado.

- Integração Contínua
 - A integração contínua é uma prática de desenvolvimento de software que envolve a automação de testes e a integração de alterações de código em um repositório compartilhado várias vezes ao dia. Benefícios incluem:
 - Detecção Precoce de Problemas: Permite que equipes identifiquem e corrijam problemas rapidamente, bem antes da entrega do produto.
 - Feedback Imediato: As alterações são testadas assim que são integradas, fornecendo feedback instantâneo sobre a saúde do aplicativo.
 - Facilitação de Entregas Frequentes: Com a integração contínua, é possível fazer entregas frequentes de partes do software que estão prontas para produção.
- Dailies
 - Reuniões semanais para compartilhar atualizações rápidas sobre o progresso do trabalho, identificar obstáculos e sincronizar as atividades.
- Adequação Funcional e Armazenamento de Dados
 - Considerando a qualidade de 'adequação funcional', essas práticas asseguram que o software não apenas funciona corretamente sob condições normais, mas também continua a funcionar de forma robusta à medida que novos requisitos são introduzidos ou modificados.

A integração das metodologias Scrum e Extreme Programming (XP) foi adotada para aumentar a adaptabilidade e a eficiência do processo de desenvolvimento ([SCHWABER; SUTHERLAND, 2020; BECK; ANDRES, 2004](#)).

Fluxo de Trabalho

O fluxo de trabalho será documentado através de um diagrama que ilustra como as atividades são organizadas e interconectadas desde o início até a conclusão de cada sprint. Este diagrama incluirá:

Planejamento de Sprint (Scrum): Definição de metas, seleção de histórias de usuário e atribuição de tarefas.

Desenvolvimento Diário (XP): Aplicação de técnicas de XP no dia a dia, como programação em pares e testes contínuos.

Revisões de Sprint e Retrospectivas (Scrum): Avaliação do progresso e adaptação de estratégias conforme necessário para melhorar os resultados dos próximos ciclos.

4 DECLARAÇÃO DE ESCOPO DO PROJETO

4.1 Backlog do produto

Épico	Feature	US	ID	Nome	Descrição
E01 - Gestão	F01	US01	1	Cadastro	O sistema deve ser capaz de cadastrar novos clientes e

de Usuários				de usuários	vendedores, fornecendo informações como nome, e-mail, senha e tipo de usuário (cliente ou vendedor).
		US02	2	Registro de usuários	O sistema deve permitir que os clientes se registrem utilizando seu e-mail e senha para acessar as funcionalidades da plataforma.
		US03	3	Gerenciamento de Acessos	O sistema deve fornecer uma interface para que o administrador possa gerenciar os acessos de usuários, concedendo ou revogando privilégios conforme necessário.
E02 - Gestão de Atendimento ao Cliente	F02	US04	4	Cadastro de produtos	O sistema deve permitir que os vendedores adicionem novos produtos ao estoque, fornecendo informações como nome, descrição, preço e quantidade disponível.
		US05	5	Edição de produtos	O sistema deve permitir que os vendedores atualizem as informações de produtos existentes, como preço, descrição e quantidade disponível.
		US06	6	Adição de Produtos ao Carrinho	O sistema deve permitir que os clientes adicionem produtos ao seu carrinho de compras, selecionando a quantidade desejada.
	F03	US07	7	Remoção de Produtos do Carrinho	O sistema deve permitir que os clientes removam produtos do carrinho antes de finalizar a compra.
		US08	8	Finalizar Pedidos	O sistema deve apresentar um resumo do pedido ao cliente ao finalizar a compra, exibindo os itens selecionados, quantidades, preços e valor total.
		US09	9	Processamento de Pagamento	O sistema deve permitir que os operadores de sistema processem os pagamentos de forma segura, utilizando diferentes métodos de pagamento (cartão de crédito, boleto, Pix).
		US10	10	Opções de Pagamento	O sistema deve oferecer aos clientes a opção de escolher entre diferentes formas de pagamento, como cartão de crédito, boleto bancário ou Pix.
	F04	US11	11	Relatório de Vendas	O sistema deve fornecer aos gerentes acesso a relatórios de vendas diárias, contendo informações como valor total das vendas, produtos mais vendidos e melhores clientes.
		US12	12	Painel	O sistema deve disponibilizar um painel administrativo

				Administrativo	para que os administradores possam acessar todos os relatórios e ferramentas de análise disponíveis no sistema.
--	--	--	--	----------------	-----------------------------------------------------------------------------------------------------------------

4.2 Perfis

Tabela : Perfis de acesso

U S	Descrição	Funcionalidade associada		Código	Features
U S 0 1	Como gerente do sistema, Eu quero poder cadastrar novos clientes e vendedores, Para que eles possam acessar suas funcionalidades específicas na plataforma.	F01		F01	Sistema Login
U S 0 2	Como cliente, Eu quero me registrar com meu email e senha, Para que eu possa realizar compras no site.	F01		F02	Gerência de Produtos
U S 0 3	Como administrador do sistema, Eu quero gerenciar os acessos de usuários, Para que possa garantir que cada usuário tenha os privilégios corretos.	F01		F03	Gerência de Pedidos
U S 0 4	Como vendedor, Eu quero adicionar novos produtos ao estoque, Para que clientes possam visualizar e comprar meus produtos.	F02		F04	Geração de Estatísticas

U S 0 5	Como vendedor, Eu quero atualizar informações de produtos existentes, Para que os detalhes e quantidades estejam sempre corretos.	F02			
U S 0 6	Como cliente, Eu quero adicionar produtos ao meu carrinho de compras, Para que eu possa gerenciar minha compra antes de finalizar o pedido.	F03			
U S 0 7	Como cliente, Eu quero remover produtos do meu carrinho, Para que eu possa ajustar meu pedido antes da compra final.	F03			
U S 0 8	Como cliente, Eu quero ver um resumo do pedido ao finalizar a compra, Para que possa conferir e confirmar os itens antes de pagar.	F03			
U S 0 9	Como operador de sistema, Eu quero processar pagamentos de forma segura, Para que os clientes tenham suas compras efetuadas com segurança.	F03			
U S 1 0	Como cliente, Eu quero escolher entre diferentes formas de pagamento, Para que eu possa utilizar o método que for mais conveniente para mim.	F03			
U S 1 1	Como gerente, Eu quero acessar relatórios de vendas diárias, Para que possa fazer planejamento estratégico baseado em dados reais de vendas.	F04			
U S 1 2	Como administrador, Eu quero ter um painel administrativo para acessar todos os relatórios e	F04			

	ferramentas de análise, Para que eu possa tomar				
--	-------------------------------------------------	--	--	--	--

4.3 Cenários

Tabela : Cenários funcionais

Código	Casos de Uso	SPRINTS
UC01	Cadastro do funcionário	SPRINT 1
UC02	Login no sistema	SPRINT 2
UC03	Registro de Comidas	SPRINT 3
UC04	Registro de Pedidos	SPRINT 4
UC05	Gerenciamento de Mesas e Horários	SPRINT 5
UC06	Consulta de Pedidos	SPRINT 6
UC07	Gerar relatório de Vendas	SPRINT 7
UC08	Gerar relatório de comidas mais vendidas	SPRINT 8

4.4 Tabela de Backlog do produto

Numeração do Cenário / Requisito	Sprint	Nome do Requisito	Tipo de Requisito (Funcional / não funcional)	Priorização do Requisito (Must, Should, Could)	Descrição Sucinta do Requisito	User Histories (U.S.) associadas
1	Sprint 1	Cadastro de Clientes e Vendedores	Funcional	Must	Permitir o cadastro de clientes e vendedores no sistema.	US01, US02
2	Sprint 1	Visualização do Cardápio e Realização de Pedidos	Funcional	Must	Permitir aos clientes visualizar o cardápio e realizar pedidos.	US03, US04
3	Sprint 2	Gerenciamento de Pedidos na Cozinha e Notificações de Pedido Pronto	Funcional	Should	Permitir à cozinha gerenciar os pedidos e notificar quando estiverem prontos.	US05, US06

4	Sprint 2	Gestão de Mesas	Funcional	Should	Gerenciar a alocação e disponibilidade de mesas.	US07
5	Sprint 3	Sistema de Reservas Online	Funcional	Could	Permitir aos clientes fazer reservas de mesas online.	US08
6	Sprint 3	Integração com Sistemas de Pagamento	Funcional	Must	Integrar diferentes métodos de pagamento ao sistema.	US09
7	Sprint 4	Feedback e Avaliações dos Clientes	Funcional	Should	Coletar e gerenciar feedbacks e avaliações dos clientes.	US10
8	Sprint 4	Emissão de Faturas e Relatórios Gerenciais	Funcional	Could	Gerar faturas e relatórios gerenciais automaticamente.	US11

5 REFERÊNCIAS BIBLIOGRÁFICAS

BECK, Kent. Extreme Programming Explained: Embrace Change. 2nd ed. Addison-Wesley, 2004.

SCHWABER, Ken; SUTHERLAND, Jeff. The Scrum Guide. [S.l.]: Scrum.org, 2020.

BECK, Kent; ANDRES, Cynthia. Extreme Programming Explained: Embrace Change. 2nd ed. Boston: Addison-Wesley, 2004.

MEYER, Eric A. CSS: The Definitive Guide. 4th ed. Sebastopol: O'Reilly Media, 2020.

FLANAGAN, David. JavaScript: The Definitive Guide. 7th ed. Sebastopol: O'Reilly Media, 2020.