**FATEC – OURINHOS**

**ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**ENGENHARIA DE SOFTWARE**

**Papa’s Pizzaria**

**Versão1.0**

**FATEC – OURINHOS**

**ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**ENGENHARIA DE SOFTWARE**

**Papa’s Pizzaria**

**Versão1.0**

**ACADÊMICOS**

MATHEUS ESPOSTO SAGAE  
REBECA TODA BARBOSA

JOÃO VINÍCIUS WOLF AZEVEDO

RAFAEL FERREIRA FRANÇA

LEONARDO FRANÇA DE OLIVEIRA

Projeto Desenvolvido para as disciplinas de Engenharia de Software da Faculdade de Tecnologia de Ourinhos.

# Gerenciamento de Configuração

| **Data** | **Atividade** | **Responsável** |
| --- | --- | --- |
| 18/04/2024 | Formatação/Composição da Engenharia de Requisitos | Matheus Esposto Sagae |
| 28/05/2024 | Feito correção após primeira entrega | Rafael Ferreira França, João Vinícius Wolf Azevedo, Rebeca Toda Barbosa, Matheus Esposto Sagae e Leonardo França de Oliveira |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Sumário**

[Gerenciamento de Configuração 3](#_gjdgxs)

[1](#_30j0zll) INTRODUÇÃO AO DOCUMENTO 5

[1.1](#_1fob9te) Tema 5

[1.2](#_3znysh7) Objetivo do Projeto 5

[1.3](#_2et92p0) Delimitação do Problema 5

[1.4](#_tyjcwt) Justificativa da Escolha do Tema 5

[1.5](#_3dy6vkm) Método de Trabalho 5

[2](#_1t3h5sf) DESCRIÇÃO GERAL DO SISTEMA 6

[2.1](#_4d34og8) Descrição do Problema 6

[2.2](#_2s8eyo1) Objetivo do Sistema 6

[2.3](#_17dp8vu) Descrição do Sistema 6

[2.3.1](#_3rdcrjn) Principais Envolvidos e suas Características 6

[2.3.2](#_26in1rg) Usuários do Sistema 6

[2.3.3](#_lnxbz9) Desenvolvedores do Sistema 6

[2.4](#_35nkun2) Regras de Negócio 6

[2.5](#_1ksv4uv) Mapa Mental do sistema 7

[3](#_44sinio) REQUISITOS DO SISTEMA 8

[3.1](#_2jxsxqh) Requisitos Funcionais 8

[3.2](#_1y810tw) Requisitos Não Funcionais 13

[4](#_2xcytpi) MODELAGEM DO SISTEMA 15

[4.1.1](#_1ci93xb) Protótipos de telas 15

[4.1.2](#_3whwml4) Cadastro de Produto 15

[4.1.3](#_2bn6wsx) Processamento do pedido 17

[4.1.4](#_qsh70q) Relatório 18

[4.2.1](#_3as4poj) Diagramas de Use Case 21

[4.2.2](#_1pxezwc) Use Case Geral 21

[4.2.3](#_49x2ik5) Use Case Específico 21

# INTRODUÇÃO AO DOCUMENTO

Este documento especifica os requisitos do sistema Papa's Pizzaria, fornecendo aos desenvolvedores as informações necessárias para o projeto e implementação, assim como para a realização dos testes e homologação do sistema.

## Tema

Desenvolvimento de Sistema para Venda/Delivery.

## Objetivo do Projeto

O aplicativo de entrega da Papa's Pizzaria visa oferecer uma plataforma digital intuitiva e eficiente para facilitar o processo de pedido, pagamento e entrega de pizzas aos clientes.

## Delimitação do Problema

Dificuldade e inconveniência na realização de pedidos, que consequentemente afeta a eficiência operacional de uma pizzaria, além da falta de transparência e rastreabilidade comparado a outros tipos/sistemas de entrega.

## Justificativa da Escolha do Tema

Com o aumento do uso de dispositivos móveis e a preferência dos consumidores por soluções digitais, a criação de um sistema de entrega de pizza online se alinha às tendências tecnológicas atuais.

## Método de Trabalho

A escolha de um modelo híbrido que integra o Modelo Cascata, Prototipação e Metodologias Ágeis é justificada pela necessidade de combinar as vantagens distintas de cada abordagem para otimizar o processo de desenvolvimento de software. Cada modelo será aplicado em fases específicas do projeto para maximizar sua eficácia.

Modelo Cascata: Utilizado na fase inicial do projeto, o Modelo Cascata, descrito por Royce (1970), oferece uma estrutura sequencial clara e bem definida. É ideal quando os requisitos são estáveis e bem compreendidos, permitindo um planejamento detalhado e uma execução ordenada. Esta abordagem será aplicada nas etapas de definição e análise de requisitos, onde a clareza e a estrutura são cruciais para o sucesso do projeto.

Prototipação: Incorporada durante a fase de design e desenvolvimento, a Prototipação, conforme descrita por Boehm (1986), permite a criação de modelos funcionais do sistema para validar requisitos e obter feedback antecipado dos usuários. Esta abordagem facilitará a identificação precoce de problemas e ajustes necessários, aumentando a precisão no entendimento das necessidades do usuário.

Metodologias Ágeis: Aplicadas durante as fases de desenvolvimento e testes, as Metodologias Ágeis, como Scrum e Extreme Programming (XP), enfatizam a flexibilidade, a colaboração contínua com o cliente e entregas incrementais frequentes. Segundo Beck et al. (2001) e Schwaber e Sutherland (2013), essas metodologias são eficazes em ambientes dinâmicos e incertos, permitindo adaptações rápidas às mudanças nos requisitos e prioridades do projeto. Elas serão empregadas para gerenciar e adaptar as mudanças conforme o projeto evolui.

Ao combinar essas metodologias de maneira estratégica, buscamos criar um processo de desenvolvimento robusto e adaptável. O Modelo Cascata proporcionará a clareza e a estrutura necessárias no início do projeto, a Prototipação garantirá a validação iterativa das necessidades do usuário, e as Metodologias Ágeis oferecerão a flexibilidade e adaptabilidade necessárias para enfrentar mudanças ao longo do ciclo de desenvolvimento.

# DESCRIÇÃO GERAL DO SISTEMA

Este capítulo tem como objetivo descrever de forma geral o sistema, o escopo e as principais funções. A descrição geral do sistema deve abranger os itens a seguir.

## Descrição do Problema

O problema a ser resolvido é a falta de eficiência e conveniência no processo de entrega de pizzas. Os principais afetados pelo sistema são tanto os clientes quanto a própria pizzaria. Para os clientes, o sistema proporcionará uma experiência de compra mais conveniente e transparente, permitindo que eles façam pedidos de pizza de forma rápida e fácil, acompanhem o status de seus pedidos em tempo real e personalizem suas escolhas de acordo com suas preferências individuais. Para a pizzaria, o sistema irá melhorar a eficiência operacional, reduzir erros de comunicação, otimizar o tempo de entrega e fornecer dados valiosos para análise e tomada de decisões, como o histórico de pedidos e as preferências dos clientes.

## Objetivo do Sistema

O objetivo geral do sistema é proporcionar um gerenciamento eficiente de pedidos e entregas em uma pizzaria. Especificamente, o sistema visa permitir um controle eficaz dos pedidos feitos pelos clientes, tanto para entrega quanto para retirada no local, e uma manutenção do cardápio de forma conveniente para os funcionários. Além disso, o sistema busca oferecer um acompanhamento detalhado do status dos pedidos, facilitando a comunicação entre a cozinha e o pessoal de entrega, e fornecer análises de dados para auxiliar os proprietários na tomada de decisões estratégicas.

## Descrição do Sistema

O sistema de entrega deverá fornecer uma interface amigável e fácil de usar para os clientes navegarem pelo menu, selecionarem suas pizzas e personalizarem seus pedidos de acordo com suas preferências, facilitar o processo de pedido e permitir que os funcionários da pizzaria possam gerenciar facilmente esses pedidos.

### Principais Envolvidos e suas Características

Rebeca será responsável quanto ao design da interface de todo o aplicativo, enquanto quanto a parte do código será dividida entre Leonardo e João Vinícius. Já a parte de teste unitários e banco de dados, Matheus e Rafael.

### Usuários do Sistema

Inicialmente será disponibilizado acesso e cadastro temporário de usuários nos quais podem ser acessados por alguns alunos para fins de teste.

### Desenvolvedores do Sistema

Os desenvolvedores serão alunos da Faculdade de Tecnologia de Ourinhos, do curso de Análise e Desenvolvimento de Sistemas, do período Noturno. O desenvolvimento terá o auxílio e gerenciamento dos professores das disciplinas de Engenharia de Software.

## Regras de Negócio

*Restrições de Negócio:* A pizzaria pode ter políticas específicas de entrega, como uma área geográfica de cobertura ou horários de funcionamento restritos, que devem ser respeitadas pelo sistema.

*Restrições de Desempenho*: O sistema deve ser capaz de lidar com um grande volume de pedidos durante os horários de pico, garantindo que o tempo de resposta permaneça aceitável para os clientes.

*Tolerância à Falhas*: O sistema deve ser projetado com redundância e mecanismos de recuperação de falhas para garantir a disponibilidade contínua do serviço, mesmo em caso de problemas técnicos.

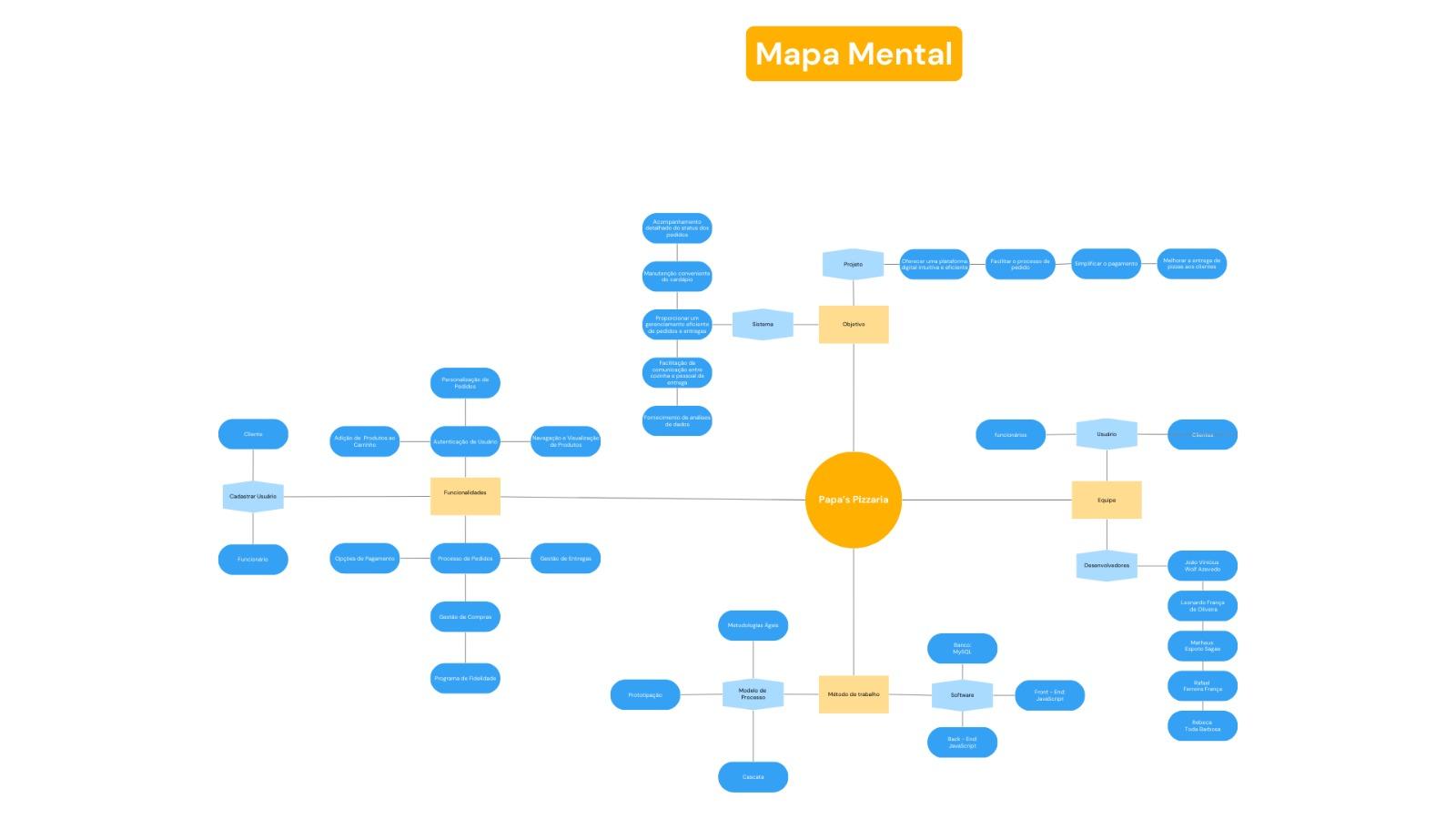
*Volume de Informação*: O sistema precisará armazenar informações sobre pedidos, clientes, menus e transações financeiras. Portanto, deve ser dimensionado para lidar com um grande volume de dados de forma eficiente.

*Estimativa de Crescimento de Volume*: O sistema deve ser escalável para lidar com um potencial aumento no volume de pedidos e dados à medida que a pizzaria cresce e conquista mais clientes.

*Ferramentas de Apoio*: O sistema pode se beneficiar de ferramentas de apoio, como sistemas de análise de dados para extrair insights valiosos dos dados coletados e ferramentas de gestão de relacionamento com o cliente (CRM) para melhorar a fidelidade do cliente e a personalização do serviço.

*Segurança da Informação*: Deve-se garantir que todas as transações financeiras sejam seguras e que os dados dos clientes sejam protegidos contra acessos não autorizados.

## Mapa Mental do sistema



# REQUISITOS DO SISTEMA

Tomando por base o contexto do sistema, foram identificados os seguintes requisitos:

## Requisitos Funcionais

Abaixo estão os requisitos funcionais do sistema, ou seja, as funções que o sistema deve exercer.

Observações:

* Dados marcados com \* (asterisco) serão considerados campos de preenchimento obrigatório.

| **Identificador** | **Descrição** | **Prioridade** | **Requisitos Relacionados** |
| --- | --- | --- | --- |
| RF01 | Cadastrar Usuário | Essencial |  |
| RF02 | Autenticação de Usuário | Essencial | RF01 |
| RF03 | Navegação e Visualização de Produtos | Essencial | RF02 |
| RF04 | Personalização de Pedidos | Essencial | RF03, RF09 |
| RF05 | Adição de Produtos ao Carrinho | Essencial | RF04 |
| RF06 | Processo de Pedido | Essencial | RF05 |
| RF07 | Gestão de Entregas | Essencial | RF06 |
| RF08 | Opções de Pagamento | Essencial | RF06, RF01 |
| RF09 | Gestão de Conta | Essencial | RF01 |
| RF10 | Programa de Fidelidade | Essencial | RF08, RF09 |

**RF01 – Cadastrar Usuário**

1. O sistema deve permitir que os usuários realizem o cadastro fornecendo informações pessoais.
2. O sistema deve permitir a inclusão dos seguintes dados no cadastro do usuário:
   1. Nome completo (obrigatório)
   2. Endereço (obrigatório)
   3. E-mail (obrigatório e deve ser único)
   4. Número de telefone (obrigatório)
3. O sistema deve validar os campos obrigatórios para garantir que:
   1. Todos os campos obrigatórios estão preenchidos.
   2. O e-mail fornecido tem um formato válido.
   3. O número de telefone fornecido tem um formato válido.
4. O sistema deve verificar a unicidade do e-mail para garantir que não haja dois usuários cadastrados com o mesmo e-mail.
5. O sistema deve permitir que o usuário execute operações de:
   1. Consulta dos seus dados de cadastro.
   2. Atualização dos seus dados de cadastro.
   3. Exclusão do seu cadastro.
6. O sistema deve enviar um e-mail de confirmação para o usuário após o cadastro ser realizado com sucesso, contendo um link para ativação da conta.
7. O sistema deve permitir que o usuário ative sua conta através do link enviado por e-mail.
8. O sistema deve garantir a segurança dos dados do usuário, utilizando criptografia para armazenar informações sensíveis como senhas.
9. O sistema deve fornecer feedback ao usuário durante o processo de cadastro, indicando se houve sucesso ou falha, e orientando sobre como corrigir eventuais erros.
10. O sistema deve permitir a recuperação da senha, caso o usuário a esqueça, enviando um e-mail com instruções para redefinição da senha.

**RF02 – Autenticação de Usuário**

1. O sistema deve fornecer um mecanismo de controle de acesso para os usuários.
2. O sistema deve permitir as seguintes operações de autenticação:
   1. Login utilizando e-mail e senha.
   2. Recuperação de senha em caso de esquecimento.
3. O sistema deve permitir a inclusão dos seguintes dados para autenticação:
   1. E-mail (obrigatório)
   2. Senha (obrigatório)
4. O sistema deve validar os campos obrigatórios durante a autenticação para garantir que:
   1. O e-mail tem um formato válido.
   2. A senha fornecida atende aos critérios de segurança estabelecidos (por exemplo, comprimento mínimo, combinação de caracteres).
5. O sistema deve permitir a recuperação de dados cadastrais em caso de esquecimento de senha ou login, enviando um e-mail com instruções para redefinição da senha ao e-mail cadastrado.
6. Após a validação bem-sucedida do login, o sistema deve apresentar ao usuário as funções disponíveis de acordo com seu perfil e permissões.
7. O sistema deve implementar medidas de segurança para a autenticação, incluindo:
   1. Criptografia das senhas armazenadas.
   2. Implementação de HTTPS para garantir a segurança da transmissão de dados.
   3. Bloqueio temporário de contas após múltiplas tentativas de login falhadas.
   4. Autenticação multifator (MFA) opcional para aumentar a segurança.
8. O sistema deve permitir que o usuário execute operações de:
   1. Consulta de suas informações de autenticação.
   2. Alteração de senha.
   3. Desativação da conta.
9. O sistema deve fornecer feedback ao usuário durante o processo de login, indicando se houve sucesso ou falha, e orientando sobre como corrigir eventuais erros.
10. O sistema deve registrar logs de atividades de autenticação, incluindo tentativas de login, mudanças de senha e recuperação de conta, para fins de auditoria e segurança.

**RF03 – Navegação e Visualização de Produtos**

1. O sistema deve permitir que os clientes naveguem facilmente pelas categorias de produtos disponíveis.
2. O sistema deve permitir a visualização dos seguintes detalhes de cada produto:
   1. Imagens do produto.
   2. Descrição detalhada do produto.
   3. Preço do produto.
3. O sistema deve garantir que a navegação e visualização dos produtos estejam acessíveis após a autenticação do usuário por meio de login e senha.
4. O sistema deve garantir que a interface do usuário para navegação e visualização de produtos seja:
   1. Amigável, proporcionando uma experiência de usuário intuitiva e fácil de usar.
   2. Responsiva, ajustando-se adequadamente a diferentes dispositivos e tamanhos de tela (desktop, tablets, smartphones).
5. O sistema deve permitir a navegação por:
   1. Categorias de pizzas.
   2. Filtros e critérios de pesquisa (por exemplo, ingredientes, preço, promoções).
6. O sistema deve fornecer feedback visual ao usuário durante a navegação, como:
   1. Indicação de carregamento de conteúdo.
   2. Destaque de produtos em promoção ou novos lançamentos.
7. O sistema deve permitir que os clientes executem operações de:
   1. Consulta de produtos por categoria ou por critérios de pesquisa específicos.
   2. Visualização de uma lista de produtos em formato de galeria ou lista.
   3. Acesso rápido aos detalhes do produto ao clicar em uma miniatura ou link.
8. O sistema deve registrar logs de navegação e visualização de produtos para fins de análise e melhoria contínua da experiência do usuário.
9. O sistema deve integrar mecanismos de acessibilidade para garantir que usuários com necessidades especiais possam navegar e visualizar produtos sem barreiras.
10. O sistema deve permitir que os clientes adicionem produtos visualizados ao carrinho de compras diretamente da interface de navegação e visualização.

**RF04 – Personalização de Pedidos**

1. O sistema deve permitir que os usuários personalizem seus pedidos de pizza.
2. O sistema deve permitir a inclusão das seguintes opções de personalização para as pizzas:
   1. Adição de ingredientes.
   2. Remoção de ingredientes.
   3. Escolha do tamanho da pizza.
   4. Escolha do tipo de borda.
3. O sistema deve calcular automaticamente o preço da pizza com base nas personalizações feitas, considerando:
   1. Custo adicional de ingredientes adicionados.
   2. Desconto ou ajuste de preço para ingredientes removidos.
   3. Diferenças de preço entre os tamanhos de pizza.
   4. Diferenças de preço entre os tipos de borda.
4. O sistema deve fornecer feedback em tempo real ao usuário durante o processo de personalização, incluindo:
   1. Atualização do preço total conforme as opções são selecionadas ou alteradas.
   2. Indicação de quais ingredientes estão disponíveis ou indisponíveis.
5. O sistema deve permitir que o usuário visualize um resumo das personalizações antes de adicionar a pizza ao carrinho de compras.
6. O sistema deve garantir que todas as opções de personalização estejam claramente apresentadas e sejam fáceis de selecionar, através de uma interface amigável e intuitiva.
7. O sistema deve permitir que o usuário salve combinações de personalização favoritas para pedidos futuros.
8. O sistema deve registrar logs de personalizações feitas pelos usuários para fins de análise e melhoria contínua do serviço.
9. O sistema deve validar as opções de personalização para garantir que:
   1. Ingredientes adicionados ou removidos sejam compatíveis com a base da pizza selecionada.
   2. O preço calculado reflete corretamente as personalizações feitas.
10. O sistema deve permitir que o usuário modifique ou cancele personalizações antes de finalizar o pedido.

**RF05 – Adição de Produtos ao Carrinho**

1. O sistema deve permitir que os clientes adicionem produtos ao carrinho de compras.
2. O sistema deve permitir a inclusão dos seguintes dados ao adicionar um produto ao carrinho:
   1. Identificação do produto.
   2. Quantidade do produto.
3. O sistema deve permitir que os clientes visualizem um resumo do pedido antes de finalizá-lo, incluindo:
   1. Lista de produtos adicionados.
   2. Quantidade de cada produto.
   3. Preço individual de cada produto.
   4. Preço total do pedido
4. O sistema deve permitir que os clientes executem operações de:
   1. Adição de produtos ao carrinho.
   2. Remoção de produtos do carrinho.
   3. Alteração das quantidades de produtos no carrinho.
5. O sistema deve atualizar automaticamente o preço total do pedido ao adicionar, remover ou alterar a quantidade de produtos no carrinho.
6. O sistema deve fornecer feedback visual ao usuário durante as operações de adição, remoção e alteração de produtos no carrinho, incluindo:
   1. Confirmação de que um produto foi adicionado ou removido.
   2. Atualização do preço total.
7. O sistema deve permitir que o usuário salve o carrinho de compras para continuar a compra posteriormente.
8. O sistema deve registrar logs das operações realizadas no carrinho de compras para fins de análise e melhoria contínua do serviço.
9. O sistema deve validar os dados do carrinho para garantir que:
   1. Todos os produtos adicionados estejam disponíveis em estoque.
   2. As quantidades solicitadas sejam permitidas.
10. O sistema deve permitir que o cliente finalize o pedido a partir do carrinho de compras, direcionando-o ao processo de pagamento e confirmação.
11. O sistema deve permitir que o cliente finalize o pedido a partir do carrinho de compras, direcionando-o ao processo de pagamento e confirmação.

**RF06 – Processo de Pedido**

1. O sistema deve permitir que os usuários revisem e confirmem seus pedidos antes da finalização.
2. O sistema deve fornecer feedback visual ao usuário durante o processo de revisão do pedido, indicando claramente os detalhes do pedido e oferecendo opções para ajustes, se necessário.
3. O sistema deve permitir que os usuários agendem pedidos para entrega em horários específicos, incluindo:
   1. Seleção da data e horário desejados para entrega.
   2. Opções de intervalo de horário de entrega.
4. O sistema deve fornecer confirmação ao usuário após a conclusão do agendamento do pedido, incluindo:
   1. Confirmação da data e horário agendados.
   2. Instruções adicionais, se necessário, para garantir uma entrega bem-sucedida.
5. O sistema deve permitir que os usuários cancelem ou modifiquem pedidos agendados dentro de um prazo determinado antes do horário de entrega agendado.
6. O sistema deve registrar logs das operações realizadas durante o processo de revisão e agendamento de pedidos para fins de análise e melhoria contínua do serviço.

**RF07 – Gestão de Entregas**

1. O sistema deve fornecer a funcionalidade de rastreamento em tempo real do status do pedido, desde o momento do preparo até a entrega.
2. O sistema deve permitir que os clientes acompanhem o progresso do pedido em tempo real, visualizando informações sobre:
   1. Status atual do pedido (por exemplo, em preparo, em rota de entrega, entregue).
   2. Atividades relacionadas ao pedido, como início do preparo, saída para entrega e entrega concluída.
   3. Localização do entregador, quando aplicável.
3. O sistema deve fornecer uma interface amigável e intuitiva para o rastreamento do pedido, acessível tanto via aplicativo quanto via website.
4. O sistema deve garantir a precisão das informações de rastreamento, atualizando o status do pedido em tempo real e refletindo quaisquer alterações ou eventos relevantes durante o processo de entrega.
5. O sistema deve permitir que os clientes recebam notificações automáticas sobre o status do pedido, incluindo:
   1. Confirmação do pedido após a conclusão da compra.
   2. Atualizações sobre o progresso do pedido, como início do preparo, saída para entrega e entrega concluída.
   3. Alertas em caso de atrasos ou problemas durante a entrega.
6. O sistema deve oferecer opções de comunicação em tempo real entre o cliente e o entregador, como mensagens de texto ou chat integrado, para facilitar a comunicação durante o processo de entrega.
7. O sistema deve permitir que os clientes forneçam feedback sobre a experiência de entrega, incluindo avaliações e comentários sobre o serviço prestado.
8. O sistema deve garantir a privacidade e segurança das informações de rastreamento e comunicação entre cliente e entregador, utilizando medidas de criptografia e proteção de dados adequadas.

**RF08 – Opções de Pagamento**

1. O sistema deve suportar diversas formas de pagamento para oferecer flexibilidade aos clientes.
2. O sistema deve permitir as seguintes opções de pagamento:
   1. Cartão de crédito.
   2. Cartão de débito.
   3. Carteiras digitais (por exemplo, Paypal, Apple Pay, Google Wallet).
   4. Pagamento na entrega (em dinheiro ou com cartão).
   5. Utilização de pontos do programa de fidelidade.
3. O sistema deve garantir que os clientes possam selecionar e utilizar a opção de pagamento desejada durante o processo de finalização do pedido.
4. O sistema deve calcular e exibir claramente o valor total a ser pago, incluindo qualquer desconto aplicado ao utilizar pontos do programa de fidelidade.
5. O sistema deve permitir que os clientes salvem suas informações de pagamento para uso em futuros pedidos, garantindo que essas informações sejam armazenadas de maneira segura e conforme as regulamentações de proteção de dados.
6. O sistema deve validar as informações de pagamento fornecidas pelos clientes, garantindo que:
   1. Os dados do cartão de crédito/débito são válidos e completos.
   2. As carteiras digitais são autenticadas corretamente.
   3. Os pontos do programa de fidelidade são suficientes para cobrir o valor do pedido, se essa opção for escolhida.
7. O sistema deve garantir a segurança das transações financeiras através de:
   1. Criptografia dos dados de pagamento durante a transmissão.
   2. Armazenamento seguro das informações de pagamento
   3. Conformidade com padrões de segurança, como PCI-DSS.
8. O sistema deve fornecer feedback em tempo real ao cliente sobre o status da transação de pagamento, incluindo:
   1. Confirmação de pagamento bem-sucedido.
   2. Notificação de falha na transação e orientações para correção.
9. O sistema deve permitir que os clientes visualizem um histórico de transações de pagamento em seu perfil, incluindo detalhes de cada pagamento realizado.
10. O sistema deve registrar logs de todas as transações financeiras para fins de auditoria, análise e melhoria contínua do serviço.
11. O sistema deve oferecer suporte ao cliente para resolução de problemas relacionados a pagamentos, como transações falhadas ou reembolsos.

**RF09 – Gestão de Conta**

1. O sistema deve permitir que os usuários visualizem e editem suas informações de perfil.
2. O sistema deve permitir a inclusão e edição dos seguintes dados no perfil do usuário:
   1. Nome completo.
   2. Endereço de entrega (com opção de adicionar múltiplos endereços).
   3. Número de telefone.
   4. E-mail.
   5. Preferências de comunicação.
3. O sistema deve permitir que os usuários executem operações de:
   1. Consulta dos seus dados de perfil.
   2. Atualização dos seus dados de perfil.
   3. Exclusão de endereços de entrega.
4. O sistema deve permitir que os usuários visualizem um histórico de pedidos anteriores, incluindo:
   1. Data e hora do pedido.
   2. Lista de produtos e quantidades.
   3. Preço total do pedido.
   4. Status do pedido (por exemplo, entregue, cancelado).
   5. Detalhes da entrega ou retirada.
5. O sistema deve fornecer feedback ao usuário durante o processo de atualização do perfil, indicando se houve sucesso ou falha, e orientando sobre como corrigir eventuais erros.
6. O sistema deve permitir que os usuários recuperem suas informações de login (por exemplo, redefinição de senha) através de um processo seguro, enviando um e-mail com instruções para o e-mail cadastrado.
7. O sistema deve permitir que os usuários desativem suas contas se desejarem, com uma opção para reativação futura.

**RF10 – Programa de Fidelidade**

1. O sistema deve implementar um programa de fidelidade que acumule pontos para os usuários após cada pedido finalizado.
2. O sistema deve calcular e adicionar pontos ao saldo do usuário com base nas seguintes condições:
   1. Valor gasto no pedido (pontos proporcionais ao valor total do pedido).
   2. Tipo de produto adicionado ao pedido (por exemplo, pontos extras para pizzas específicas).
3. O sistema deve permitir que os usuários visualizem seu saldo de pontos atual e histórico de pontos acumulados em sua conta.
4. O sistema deve permitir que os usuários utilizem seus pontos acumulados durante o processo de pagamento para obter descontos no valor total do pedido.
5. O sistema deve calcular automaticamente o desconto aplicado com base na quantidade de pontos que o usuário deseja utilizar, garantindo que:
   1. O usuário possa escolher a quantidade de pontos a ser utilizada.
   2. O desconto não exceda o valor total do pedido.
6. O sistema deve fornecer feedback em tempo real ao usuário sobre o saldo de pontos disponível e o desconto aplicado ao utilizar pontos no pagamento.
7. O sistema deve garantir a segurança e integridade dos pontos acumulados, evitando fraudes e garantindo que os pontos sejam corretamente contabilizados.
8. O sistema deve permitir que os usuários executem operações de:
   1. Consulta de saldo de pontos.
   2. Visualização de histórico de pontos acumulados e utilizados.
   3. Resgate de pontos para descontos nos pedidos.
9. O sistema deve notificar os usuários sobre a adição de novos pontos após cada pedido finalizado, incluindo detalhes sobre como os pontos foram calculados.
10. O sistema deve permitir que os usuários participem de promoções especiais do programa de fidelidade, onde pontos extras podem ser acumulados por atividades específicas ou em eventos promocionais.

## Requisitos Não Funcionais

Abaixo estão os requisitos não funcionais do sistema.

| **Identificador** | **Descrição** | **Categoria** | **Prioridade** | **Requisitos relacionados** |
| --- | --- | --- | --- | --- |
| RNF01 | O usuário deve ser capaz de alterar sua senha sem a intervenção de um administrador. | Usabilidade | Alta |  |
| RNF02 | O sistema deve ser compatível com uma variedade de dispositivos. | Compatibilidade | Alta | RNF03 |
| RNF03 | O sistema deve ser escalável para lidar com um aumento no número de clientes e pedidos. | Escalabilidade | Alta |  |
| RNF04 | O sistema deve ser seguro contra acesso não autorizado. | Segurança | Alta | RNF01 |
| RNF05 | O tempo de resposta para ações do usuário deve ser razoável. | Desempenho | Média | RNF03 |

**RNF01 – Usabilidade**

1. O usuário deve ser capaz de alterar sua senha sem a intervenção de um administrador.
2. O controle deve ser a partir da entrada de seu login e senha;
3. A interface do usuário deve ser intuitiva e de fácil utilização, permitindo que clientes de diferentes níveis de habilidade possam fazer pedidos sem dificuldades.

**RNF02 – Compatibilidade**

1. O sistema deve ser compatível com uma variedade de dispositivos, garantindo uma experiência consistente para todos os usuários.
2. Antes de atualizado o aplicativo, deve-se testar em diversas plataformas para garantir o quesito de compatibilidade.

**RNF03 – Escalabilidade**

1. O sistema deve ser escalável para lidar com um aumento no número de clientes e pedidos, sem comprometer o desempenho ou a qualidade do serviço

**RNF04 – Manutenibilidade**

1. O código-fonte do sistema deve ser bem estruturado, modular e devidamente documentado, facilitando a manutenção e a incorporação de novos recursos no futuro.
2. O sistema deve seguir padrões de codificação consistentes e práticas de desenvolvimento ágil.

# MODELAGEM DO SISTEMA

Este Tópico apresenta a modelagem referentes ao projeto apresentado neste documento.

Foram criados modelos baseados no paradigma Orientado a Objetos.

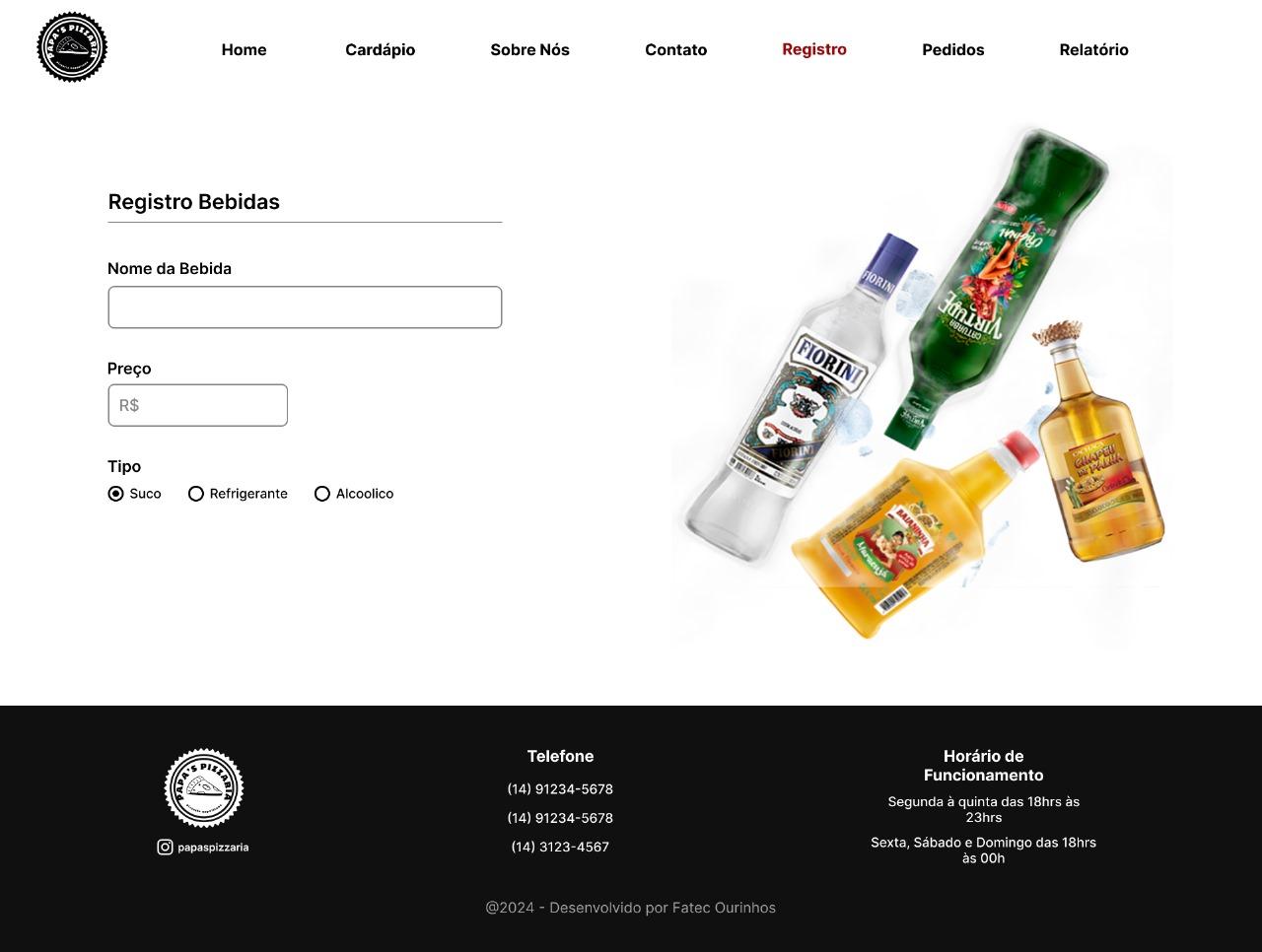
Para o desenvolvimento dos modelos do sistema a ser gerado foi utilizada a Linguagem de Unificada de Modelos (UML).

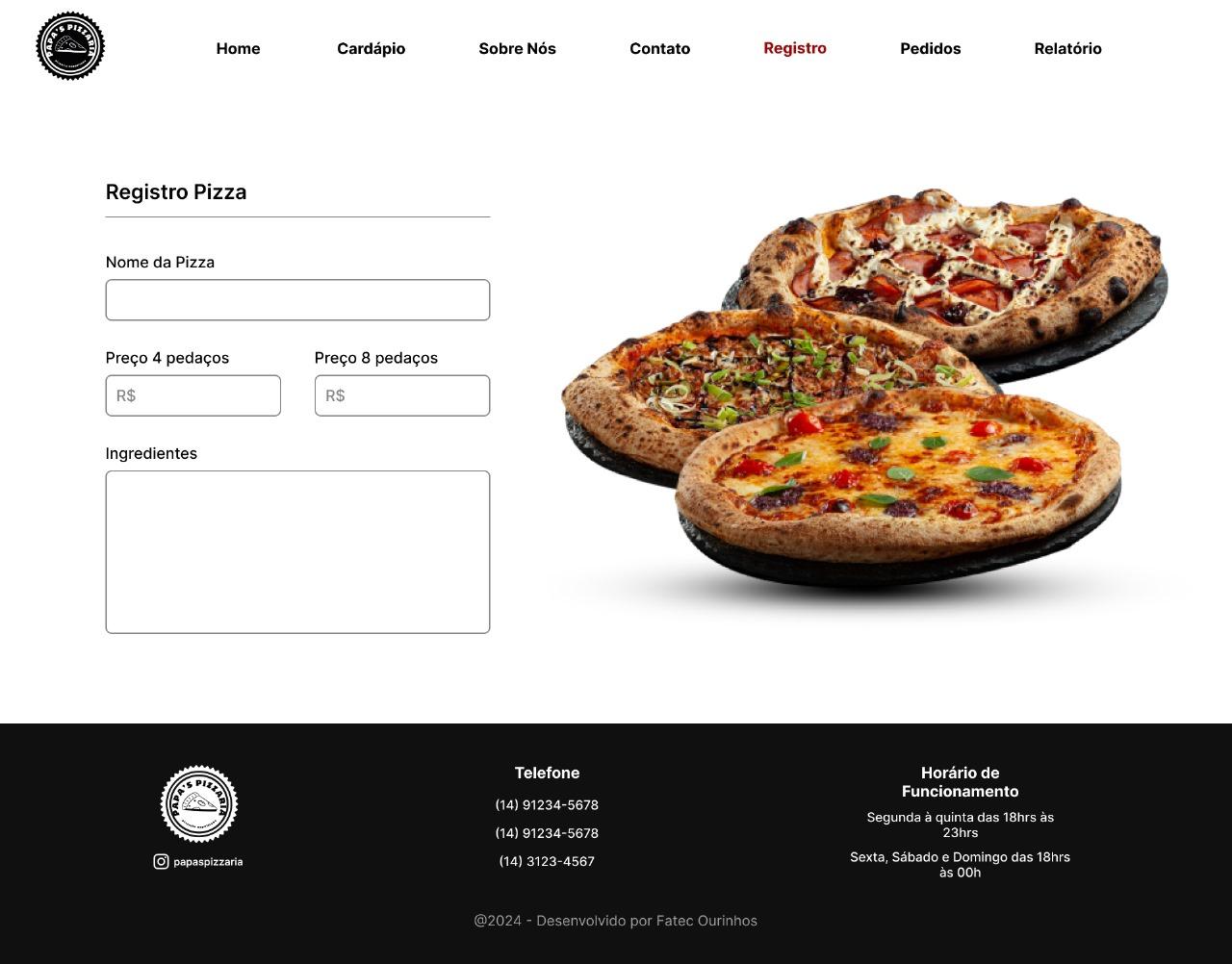
A criação dos modelos foi com auxilio das seguintes ferramentas CASE:

* VisualParadigm Online para modelagem dos processos do sistema (Diagramas de Use Case, Classes, Sequencia e Atividades).
* Figma para modelagem das telas do sistema.

### Protótipos de telas

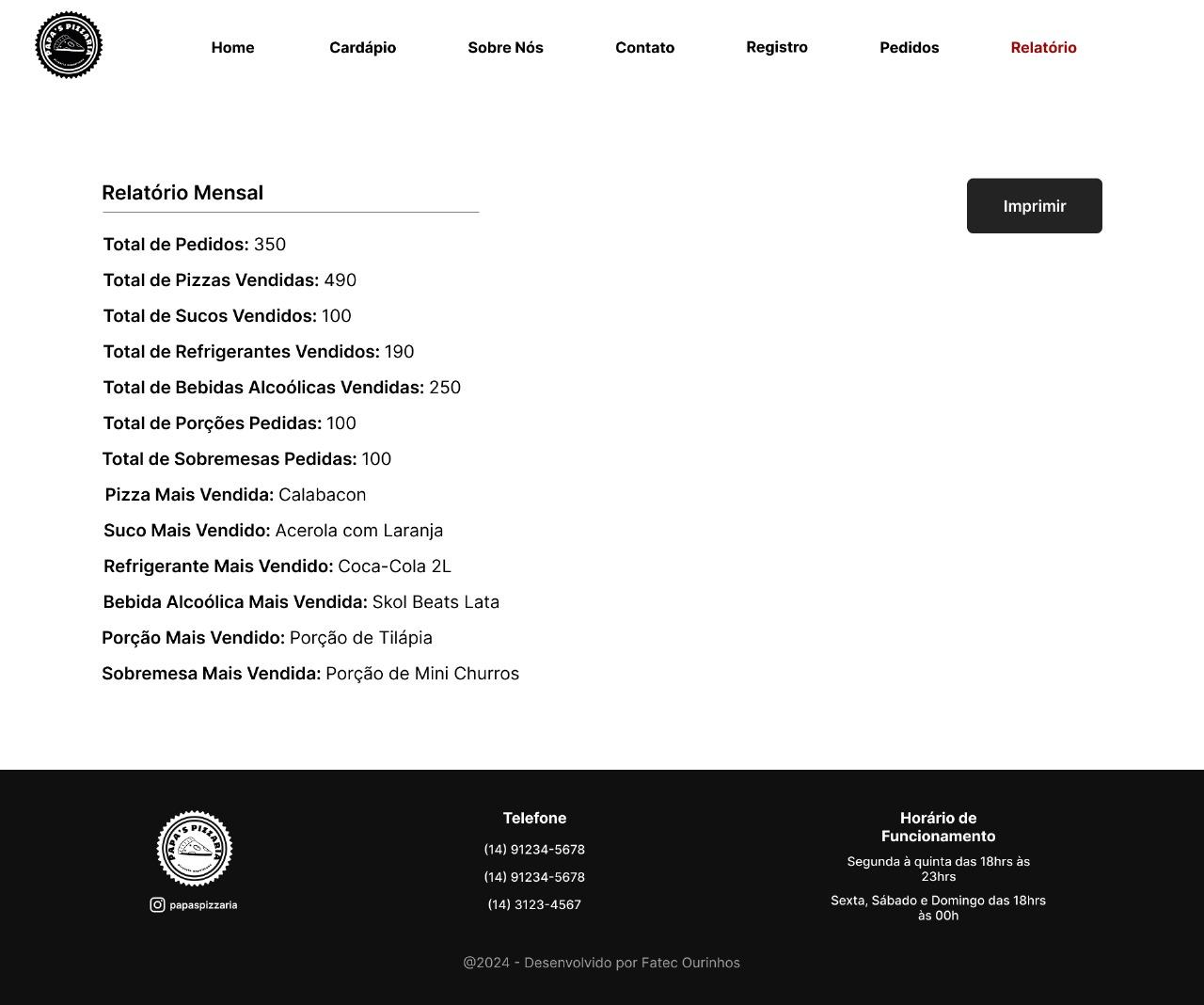
A seguir serão apresentados **protótipos** das telas do sistema. Elas poderão sofrer alterações.

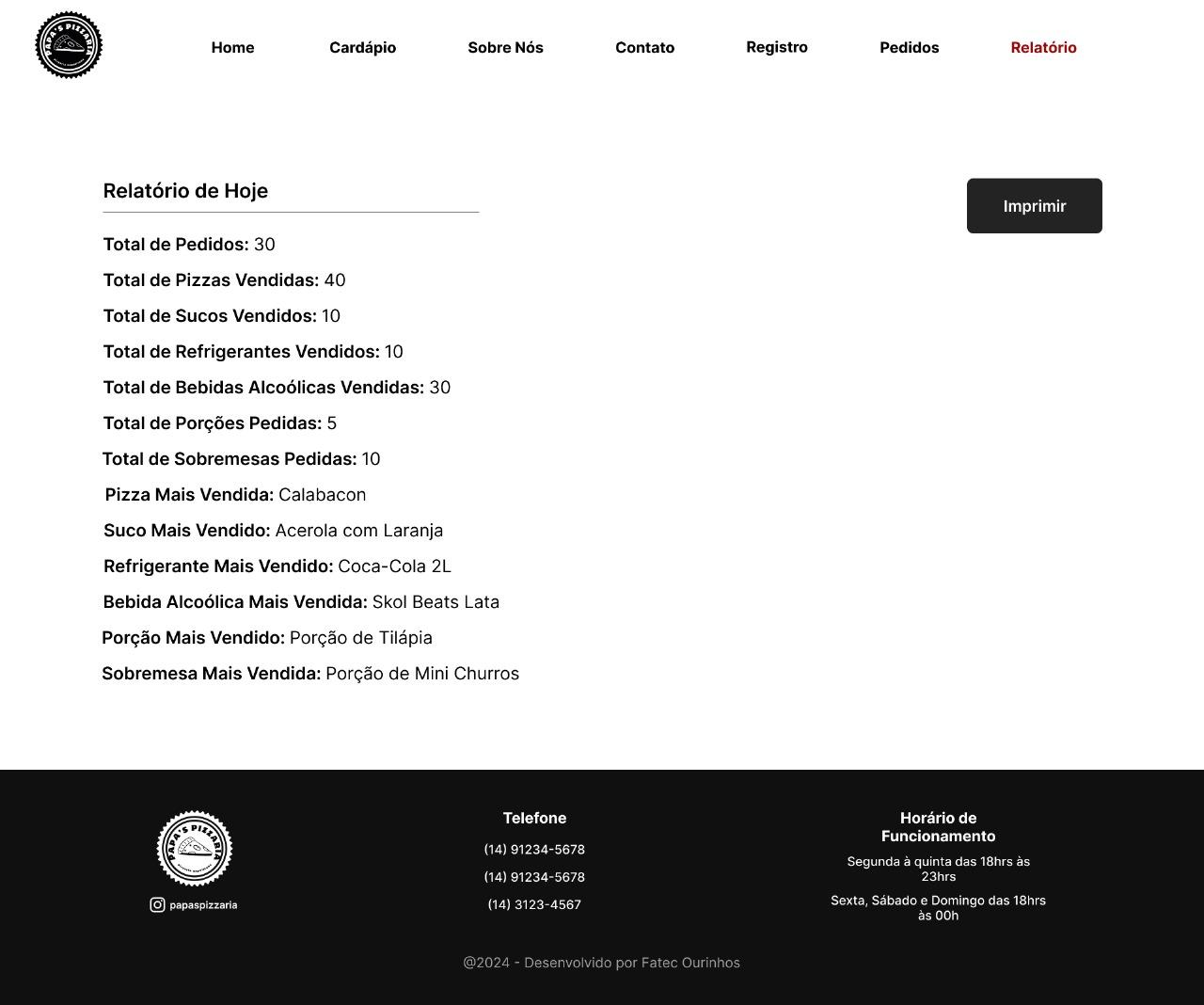
* + 1. **Cadastro de Produto**

****

### Processamento do pedidoInterface gráfica do usuário Descrição gerada automaticamente com confiança média

### Relatório

****



### Diagramas de Use Case

### Use Case Geral

Diagrama

Descrição gerada automaticamente

### Use Case Específico

| **Caso de Uso** | **Ator Principal** | **Pré-condições** | **Descrição** | **Pós-condições** |
| --- | --- | --- | --- | --- |
| Efetuar o Cadastro | Usuário | O usuário deve ter o aplicativo instalado no seu dispositivo. | 1. O usuário abre o aplicativo. 2. Seleciona "Cadastrar". 3. Preenche informações pessoais. 4. Clica em "Confirmar". 5. Sistema valida e cria a conta. 6. Sistema envia e-mail de confirmação. 7. Usuário confirma e-mail. | O usuário tem uma conta ativa no aplicativo. |
| Alterar Informações do Perfil | Usuário | O usuário deve estar logado no aplicativo. | 1. O usuário acessa a área de perfil. 2. Seleciona "Editar Perfil". 3. Sistema exibe as informações atuais. 4. Usuário modifica as informações. 5. Clica em "Salvar Alterações". 6. Sistema valida e atualiza. | As informações do perfil são atualizadas. |
| Visualizar Cardápio | Usuário | O usuário deve estar logado no aplicativo. | 1. O usuário navega até a página inicial. 2. Seleciona um restaurante. 3. Sistema exibe o cardápio. 4. Usuário visualiza os itens do cardápio. | O usuário visualiza o cardápio do restaurante. |
| Adicionar ao Carrinho | Usuário | O usuário deve estar logado e visualizando o cardápio. | 1. O usuário seleciona um item do cardápio. 2. Escolhe as opções adicionais. 3. Clica em "Adicionar ao Carrinho". 4. Sistema adiciona o item ao carrinho. | O item é adicionado ao carrinho do usuário. |
| Método de Pagamento | Usuário | O usuário deve ter itens no carrinho. | 1. O usuário navega até o carrinho. 2. Clica em "Finalizar Compra". 3. Sistema exibe opções de pagamento. 4. Usuário seleciona método de pagamento. 5. Insere informações de pagamento. 6. Confirma método de pagamento. | O método de pagamento é selecionado. |
| Validar Pedido | Sistema | O usuário deve ter selecionado o método de pagamento. | 1. Sistema revisa o pedido. 2. Verifica a disponibilidade dos itens. 3. Confirma a validade das informações de pagamento. | O pedido é validado. |
| Pedido em Preparação | Restaurante | O pedido deve ser validado pelo sistema. | 1. Restaurante recebe o pedido. 2. Começa a preparar os itens. 3. Sistema atualiza o status para "Em Preparação". | O pedido está sendo preparado. |
| Pedido Concluído | Restaurante | O pedido deve estar em preparação. | 1. Restaurante conclui a preparação. 2. Sistema atualiza o status para "Concluído". | O pedido está pronto para coleta. |
| Pedido Finalizado | Sistema | O pedido deve estar concluído. | 1. Sistema verifica que o pedido está pronto. 2. Prepara o pedido para o entregador. | O pedido está pronto para entrega. |
| Validar Pagamento | Sistema | O pedido deve estar pronto para ser entregue. | 1. Sistema verifica as informações de pagamento. 2. Confirma a transação financeira. | O pagamento é confirmado. |
| Emitir NFC | Sistema | O pagamento deve ser validado. | 1. Sistema gera a Nota Fiscal de Consumidor (NFC). 2. Envia a NFC para o e-mail do usuário e armazena no histórico de pedidos. | A NFC é emitida e enviada ao usuário. |
| Pedido em Entrega | Entregador | O pedido deve estar finalizado e pronto para entrega. | 1. Entregador coleta o pedido. 2. Sistema atualiza o status para "Em Entrega". 3. Entregador segue para o endereço. | O pedido está a caminho. |
| Confirmação de Entrega | Entregador | O pedido deve estar em entrega. | 1. Entregador chega ao endereço. 2. Entrega o pedido. 3. Confirma a entrega no aplicativo. 4. Sistema atualiza o status para "Entregue". | O pedido é marcado como entregue. |

Diagrama

Descrição gerada automaticamente

### Use Case Específico - Navegar e visualizar produtos

| **Definição** | **Detalhe** |
| --- | --- |
| UseCase | Navegar e visualizar produtos |
| Sumário | Permite ao usuário navegar e visualizar os produtos disponíveis. |
| Ator | Cliente |
| Pré-condição | Usuário acessa o site |
| Pós-condição | Produtos visualizados |
| Base Sequence | 1. Usuário acessa o site 2. Navega pelas categorias de produtos 3. Visualiza os detalhes de um produto |
| Exception Sequence | 1. Produtos indisponíveis 2. Erro no carregamento da página |

### Use Case Específico – Cadastrar usuário

| **Definição** | **Detalhe** |
| --- | --- |
| UseCase | Cadastrar usuário |
| Sumário | Permite ao usuário criar uma conta no sistema. |
| Ator | Clientes |
| Pré-condição | Usuário acessa a página de cadastro |
| Pós-condição | Conta de usuário criada |
| Base Sequence | 1. Usuário acessa a página de cadastro 2. Preenche os dados necessários 3. Submete o formulário 4. Recebe confirmação de cadastro |
| Exception Sequence | 1. Dados inválidos 2. Erro no servidor |

### Use Case Específico – Autenticar usuário

| **Definição** | **Detalhe** |
| --- | --- |
| UseCase | Autenticar usuário |
| Sumário | Permite ao usuário realizar login no sistema. |
| Ator | Cliente |
| Pré-condição | Usuário já cadastrado |
| Pós-condição | Usuário autenticado |
| Base Sequence | 1. Usuário acessa a página de login 2. Insere credenciais 3. Submete o formulário 4. Acesso concedido |
| Exception Sequence | 1. Credenciais inválidas 2. Conta bloqueada |

### Use Case Específico – Personalizar pedido

| **Definição** | **Detalhe** |
| --- | --- |
| UseCase | Personalizar pedido |
| Sumário | Permite ao usuário personalizar os detalhes do pedido. |
| Ator | Cliente |
| Pré-condição | Usuário autenticado e produto adicionado ao carrinho |
| Pós-condição | Pedido personalizado |
| Base Sequence | 1. Usuário acessa o carrinho 2. Seleciona personalização 3. Salva as alterações |
| Exception Sequence | 1. Personalização indisponível 2. Erro ao salvar as alterações |

### Use Case Específico – Adicionar produtos no carrinho

| **Definição** | **Detalhe** |
| --- | --- |
| UseCase | Adição de produtos ao carrinho |
| Sumário | Permite ao usuário adicionar produtos ao carrinho de compras. |
| Ator | Cliente |
| Pré-condição | Usuário navegando no site |
| Pós-condição | Produtos adicionados ao carrinho |
| Base Sequence | 1. Usuário visualiza um produto 2. Seleciona a quantidade 3. Adiciona ao carrinho |
| Exception Sequence | 1. Produto esgotado 2. Erro ao adicionar produto |

### Use Case Específico – Processo de pedido

| **Definição** | **Detalhe** |
| --- | --- |
| UseCase | Processo de pedido |
| Sumário | Permite ao usuário finalizar o pedido e realizar pagamento. |
| Ator | Cliente |
| Pré-condição | Produtos no carrinho e usuário autenticado |
| Pós-condição | Pedido processado e pagamento confirmado |
| Base Sequence | 1. Usuário acessa o carrinho 2. Confirma os produtos 3. Realiza o pagamento 4. Recebe confirmação do pedido |
| Exception Sequence | 1. Falha no pagamento 2. Erro no processamento do pedido |

### Use Case Específico – Gestão de pedidos

| **Definição** | **Detalhe** |
| --- | --- |
| UseCase | Gestão de entregas |
| Sumário | Permite ao cliente gerenciar as entregas dos pedidos. |
| Ator | Cliente |
| Pré-condição | Pedido confirmado |
| Pós-condição | Entrega informada |
| Base Sequence | 1. cliente acessa o painel de entregas 2. Visualiza os pedidos |
| Exception Sequence | 1. Erro ao visualizar pedido 2. Informações de entrega incompletas |

### Use Case Específico – Cadastrar produtos

| **Definição** | **Detalhe** |
| --- | --- |
| UseCase | Cadastrar produtos |
| Sumário | Permite ao administrador cadastrar novos produtos no sistema. |
| Ator | Administrador, Gerente |
| Pré-condição | Administrador autenticado |
| Pós-condição | Produto cadastrado |
| Base Sequence | 1. Administrador acessa a página de cadastro de produtos 2. Preenche os dados do produto 3. Submete o formulário |
| Exception Sequence | 1. Dados inválidos 2. Erro no servidor |

### Use Case Específico – Cadastrar usuários

| **Definição** | **Detalhe** |
| --- | --- |
| UseCase | Cadastrar usuários |
| Sumário | Permite ao administrador cadastrar novos usuários no sistema. |
| Ator | Administrador, Gerente |
| Pré-condição | Administrador autenticado |
| Pós-condição | Usuário cadastrado |
| Base Sequence | 1. Administrador acessa a página de cadastro de usuários 2. Preenche os dados do usuário 3. Submete o formulário |
| Exception Sequence | 1. Dados inválidos 2. Erro no servidor |

### Use Case Específico – Emitir relatórios

| **Definição** | **Detalhe** |
| --- | --- |
| UseCase | Emitir relatórios |
| Sumário | Permite ao administrador emitir relatórios sobre as operações. |
| Ator | Administrador, Gerente |
| Pré-condição | Administrador autenticado |
| Pós-condição | Relatório emitido |
| Base Sequence | 1. Administrador acessa a página de relatórios 2. Seleciona o tipo de relatório 3. Gera o relatório 4. Visualiza/baixa o relatório |
| Exception Sequence | 1. Falha ao gerar relatório 2. Dados insuficientes |

### Use Case Específico – Confirmar pedido

| **Definição** | **Detalhe** |
| --- | --- |
| UseCase | Confirmar pedido |
| Sumário | Permite ao usuário confirmar o pedido realizado. |
| Ator | Cozinheiro |
| Pré-condição | Pedido personalizado e pagamento realizado |
| Pós-condição | Pedido confirmado |
| Base Sequence | 1. Usuário visualiza o resumo do pedido 2. Confirma os detalhes do pedido 3. Recebe confirmação de pedido |
| Exception Sequence | 1. Falha na confirmação 2. Erro no servidor |

### Use Case Específico – Encerrar pedido

| **Definição** | **Detalhe** |
| --- | --- |
| UseCase | Encerrar pedido |
| Sumário | Permite ao administrador encerrar um pedido após a entrega. |
| Ator | Cozinheiro |
| Pré-condição | Pedido entregue |
| Pós-condição | Pedido encerrado |
| Base Sequence | 1. Administrador acessa o painel de pedidos 2. Seleciona o pedido entregue 3. Marca o pedido como encerrado |
| Exception Sequence | 1. Pedido não entregue 2. Erro ao encerrar o pedido |

### Use Case Específico – Confirmar entrega

| **Definição** | **Detalhe** |
| --- | --- |
| UseCase | Confirmar entrega |
| Sumário | Permite ao usuário confirmar a entrega do pedido. |
| Ator | Entregador |
| Pré-condição | Pedido entregue |
| Pós-condição | Entrega confirmada |
| Base Sequence | 1. Usuário recebe a entrega 2. Acessa o painel de pedidos 3. Confirma a entrega |
| Exception Sequence | 1. Entrega não recebida 2. Erro ao confirmar entrega |

### Use Case Específico – Encerrar pedido

| **Definição** | **Detalhe** |
| --- | --- |
| UseCase | Encerrar entrega |
| Sumário | Permite ao administrador encerrar a entrega após confirmação do usuário. |
| Ator | Entregador |
| Pré-condição | Entrega confirmada pelo usuário |
| Pós-condição | Entrega encerrada |
| Base Sequence | 1. Administrador acessa o painel de entregas 2. Seleciona a entrega confirmada 3. Marca a entrega como encerrada |
| Exception Sequence | 1. Falha na confirmação do usuário 2. Erro ao encerrar a entrega |

### Diagrama sequencial

