

Beatriz Soares de Melo

Camila Silva Romagnoli

Kesley Alexsandro Vieira Severino

Lucas Moura Pereira

Leonardo Augusto Moreira Lourenço

Letícia Amanda Franco Gonçalves

Octávio Tabai Ribeiro Lage

**Importante**

Este documento é um modelo do projeto a ser criado para a disciplina Gestão de Configuração e Evolução de Software do curso de Engenharia de Software da PUC Minas.

Belo Horizonte, \_\_\_\_\_\_\_\_\_\_\_\_\_

Sumário

[1.](#_3znysh7) Introdução 3

[Problema 3](#_2et92p0)

[Objetivos 3](#_tyjcwt)

[Justificativa 3](#_3dy6vkm)

[Público alvo 3](#_1t3h5sf)

[2.](#_2s8eyo1) Especificação do Projeto 4

[Personas 4](#_17dp8vu)

[Requisitos Funcionais 4](#_3rdcrjn)

[Requisitos não funcionais 4](#_26in1rg)

[3.](#_35nkun2) Metodologia 5

[Relação de Ambientes de Trabalho 5](#_1ksv4uv)

[Gerenciamento do Projeto (CCC) 5](#_44sinio)

[4.](#_z337ya) Projeto de Interface 6

[Diagrama de caso de uso e Diagrama de Classe 6](#_3j2qqm3)

[5.](#_1y810tw) Arquitetura da Solução 6

[Diagrama de Componentes 6](#_4i7ojhp)

[Hospedagem 6](#_2xcytpi)

[6.](#_1ci93xb) Funcionalidades do Sistema (Telas) 7

[7.](#_qsh70q) Plano de Testes de Software 8

[8.](#_3as4poj) Registro de Testes de Software 9

[Referências 10](#_49x2ik5)

# Introdução

Aqui o time CCC deverá introduzir o tema, sobretudo com aspectos relacionados às necessidade de se criar sistemas para gestão de negócios como o caso da Pizzaria ( o time deverá também criar uma identidade para o sistema – nome)

## Problema

Relatar de maneira breve e objetiva o problema, ou seja, a dor do cliente que por sua vez buscou a solução de software através deste Projeto.

## Objetivos

Relatar o principal objetivo do Projeto e desdobrá-lo em objetivos específicos.

Como objetivos específicos, podemos ressaltar:

* Fornecer funcionalidades que permitam.....

## Justificativa

Breve justificativa sobre a importância do Software para a empresa.

## Público alvo

Beneficiários do Software.

# 

# Especificação do Projeto

Levantar com o time de Dev as personas definidas por eles e inserir nesta Seção do Projeto.

## Personas

### Requisitos Funcionais

Devem ser inseridas as solicitações de mudanças/implementação aprovadas e refinadas pelos times de CCC e DEV identificando a situação em que as mesmas foram analisadas.

| **ID** | **Descrição** | **Situação** |
| --- | --- | --- |
| RF-01 | O Cliente faz pedido via Delivery | Aprovado |
| RF-02 | Atendente ou Garçom cadastram pedidos na fila | Reprovado |
| RF-03 | Caixa ou Entregador dão baixa na conta |  |
| RF-04 | Garçom gerencia pedidos presenciais prontos |  |
| RF-05 | Atendente gerencia pratos no cardápio |  |
| RF-06 | Entregador gerencia pedidos delivery prontos |  |
| RF-07 | Pizzaiolo gerencia pedidos pendentes |  |
| RF-08 |  |  |
| RF-09 |  |  |
| RF-10 |  |  |

### Requisitos não funcionais

A tabela a seguir deve apresentar os requisitos não funcionais que o projeto deverá atender.

| **ID** | **Descrição** | **Prioridade** |
| --- | --- | --- |
| RNF-01 | Os pedidos mais antigos devem ser priorizados na fila |  |
| RNF-02 | A plataforma deve mostrar o tempo decorrido até a entrega do pedido na mesa ou endereço |  |
| RNF-03 |  |  |
| RNF-04 |  |  |

# 

# Metodologia

A metodologia contempla as definições de ferramental utilizado pela equipe tanto para a manutenção dos códigos e demais artefatos quanto para a organização dos times na execução das tarefas do projeto.

## Relação de Ambientes de Trabalho

Os artefatos gerados por cada time do projeto são desenvolvidos a partir de algumas plataformas e a relação dos ambientes com seu respectivo propósito é apresentada na tabela que se segue.

| **Ambiente** | **Plataforma** | **Link de Acesso** |
| --- | --- | --- |
| Repositório de código fonte (DEV) |  |  |
| Documentação do projeto (DOC) |  |  |
| Testes (Testes) |  |  |
| Gerenciamento do Projeto (CCC) |  |  |

## Gerenciamento do Projeto (CCC)

A equipe utiliza metodologias ágeis, tendo escolhido o Scrum como base para definição do processo de desenvolvimento.

A equipe está organizada da seguinte maneira:

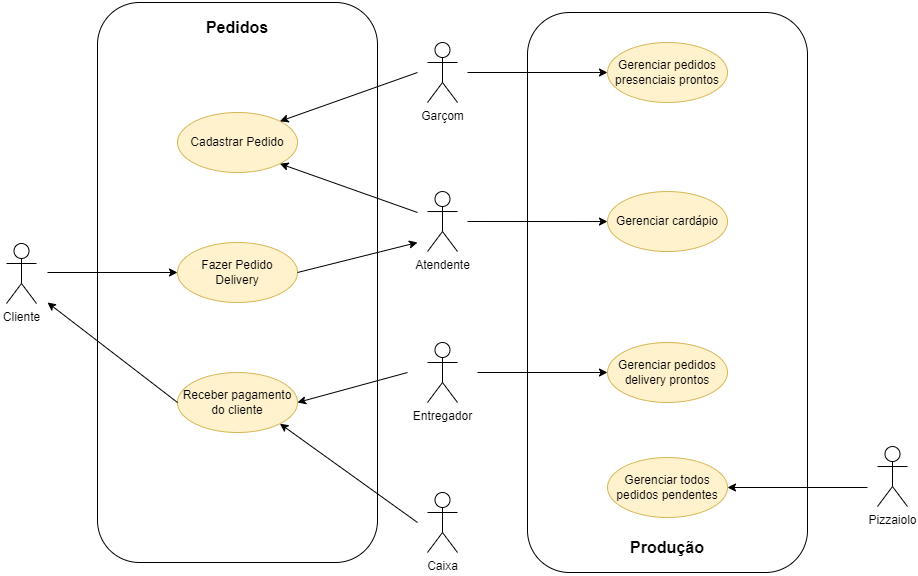
* Scrum Master: Nome do gestor do Projeto (o time CCC irá definir o nome a ser indicado aqui!)
* Product Owner: Simone de Assis Alves da Silva
* Equipe de Desenvolvimento: nome dos componentes DEV
* Equipe de CCC: nome dos componentes CCC
* Equipe de DOC: nome dos componentes DOC
* Equipe de Testes: nome dos componentes Testes

# 

# Projeto de Interface

Dentre as preocupações para a montagem da interface do sistema, estamos estabelecendo foco em questões como agilidade, acessibilidade e usabilidade. Desta forma, o projeto tem uma identidade visual padronizada em todas as telas que são projetadas para funcionamento via Web.

## Diagrama de caso de uso



## Caso de uso descritivo

**Caso de uso:** Cadastrar pedido

**Descrição:** Registrar o pedido do cliente no sistema

**Atores:** Atendente, garçom

**Pré-condições:** Possuir itens cadastrados no estoque

**Fluxo principal**:

1. Cliente realiza pedido pessoalmente ou por telefone.

2. Quem recebe verifica se existem os produtos no estoque.

3. Atendente registra o pedido do cliente no sistema e emite recibo **(FA01).**

4. Fim do fluxo

**Fluxo alternativo 01 -** Cliente cancela pedido**:**

1. Atendente muda status do pedido para cancelado.

2. Fim do fluxo

**Caso de uso:** Gerenciar cardápio

**Descrição:** O atendente registra os produtos da pizzaria

**Atores:** Atendente

**Pré-condições:** -

**Fluxo principal:**

1. Atendente lista produtos do cardápio.

2. Atendente clica cadastrar novo produto **(FA01, FA02).**

3. Atendente digita as informações do produto (nome, descrição, preço, tamanho).

4. Atendente atualiza o cardápio.

5. Fim do fluxo

**Fluxo alternativo 01 -** Editar produto existente**:**

1. O Atendente escolhe por editar/excluir um produto existente.

2. Ir para o passo 4 do Fluxo principal.

**Fluxo alternativo 02 -** Desistência**:**

1. O Atendente cancela a operação.

2. Fim do fluxo.

**Caso de uso:** Gerenciar pedidos pendentes

**Descrição:** Pizzaiolo visualiza o pedido recebido e prepara

**Atores:** Pizzaiolo

**Pré-condições:** Existir pedidos pendentes

**Fluxo principal:**

1. Pizzaiolo lista pedidos pendentes (ordenado por chegada).

2. Pizzaiolo prepara o pedido mais antigo.

3. Após pronto, pizzaiolo marca pedido como pronto.

4. Fim do fluxo.

**Caso de uso:** Gerenciar pedidos prontos

**Descrição:** Lista os pedidos prontos para entrega

**Atores:** Entregador e garçom

**Pré-condições:** Ter pedidos prontos

**Fluxo principal:**

1. Entregador ou garçom listam os pedidos prontos;

2. Coletar pedido pronto no balcão destinado.

3. Levar o pedido até a mesa do cliente **(FA01).**

4. Entregar pedido **(FA02, FA03).**

5. Fim do fluxo.

**Fluxo alternativo 01 -** Pedido delivery**:**

1. Levar pedido até o endereço de entrega.

2. Ir para o passo 4 do Fluxo principal.

**Fluxo alternativo 02 -**  Pedido incorreto**:**

1. Alterar o status do pedido para pendente e com prioridade.

2. Voltar para o passo 1 do Fluxo principal.

**Fluxo alternativo 02 -**  Cliente ausente**:**

1. Funcionário altera o status do pedido para cancelado.

2. Fim do fluxo.

**Caso de uso:** Receber pagamento

**Descrição:** Recebe o pagamento pelos pedidos e dá baixa na conta.

**Atores:** Caixa, entregador

**Pré-condições:** Pedido entregue

**Fluxo principal:**

1. Funcionário informa ao cliente o valor da conta.

2. Cliente paga valor informado e recebe troco.

3. Funcionário da baixa no pedido/conta do cliente.

**Caso de uso:** Fazer pedido delivery

**Descrição:** Cliente contata Atendente para fazer o pedido

**Atores:** Cliente, atendente

**Pré-condições:** Ter produtos no cardápio

**Fluxo principal:**

1. Cliente obtém cardápio.

2. Cliente informa ao atendente o seu pedido com produto e quantidade **(FA01).**

3. Cliente confirma pedido.

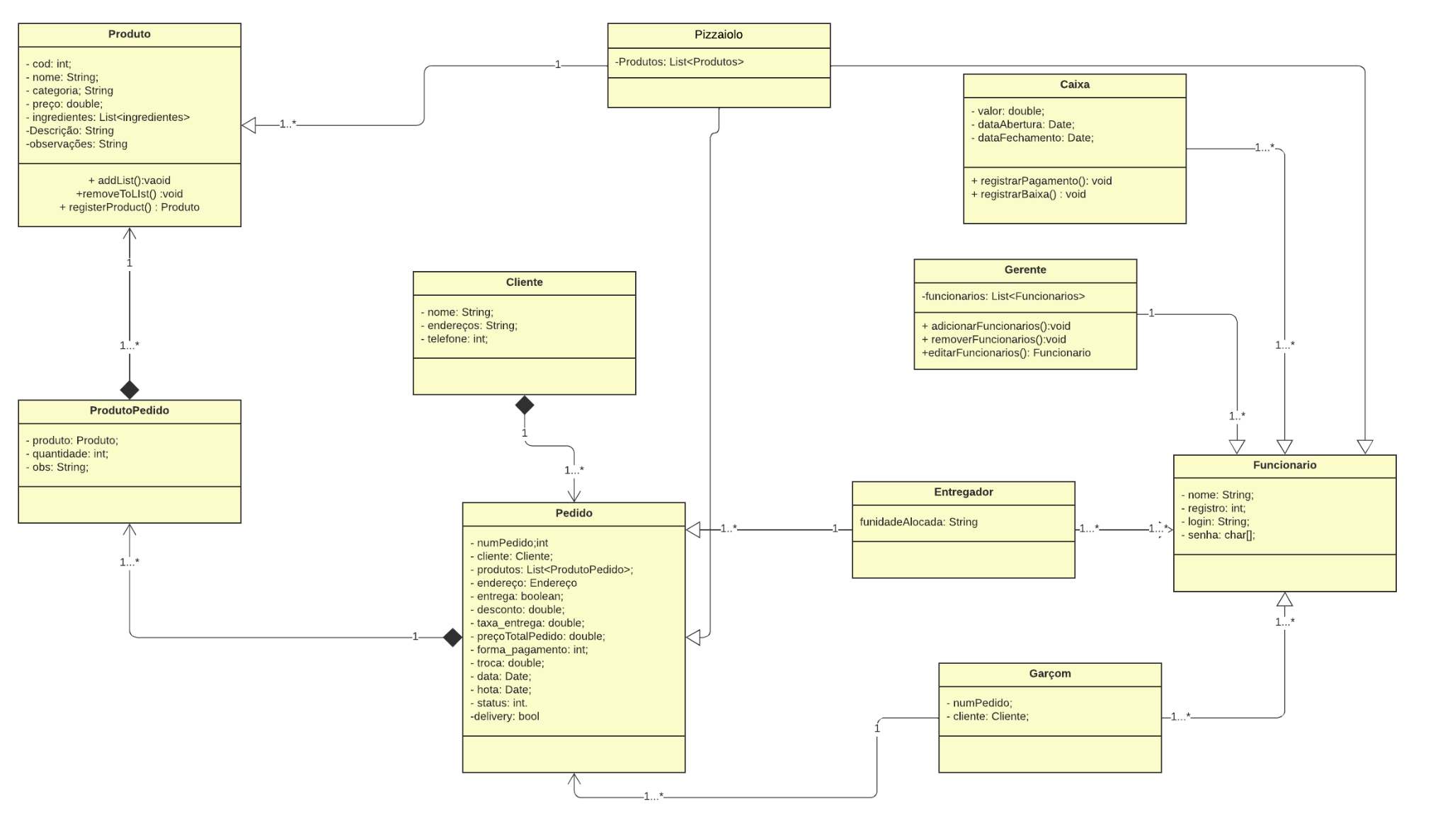
4. Fim do fluxo.

**Fluxo alternativo 01 -** Cliente cancela operação**:**

1. Atendente encerra atendimento.

2. Fim do fluxo

## Diagrama de Classe



# Arquitetura da Solução

Nesta seção são apresentados os detalhes técnicos da solução criada pela equipe, tratando dos componentes que fazem parte da solução e do ambiente de hospedagem da solução.

### Diagrama de Componentes

Os componentes que fazem parte da solução são apresentados na Figura que se segue.

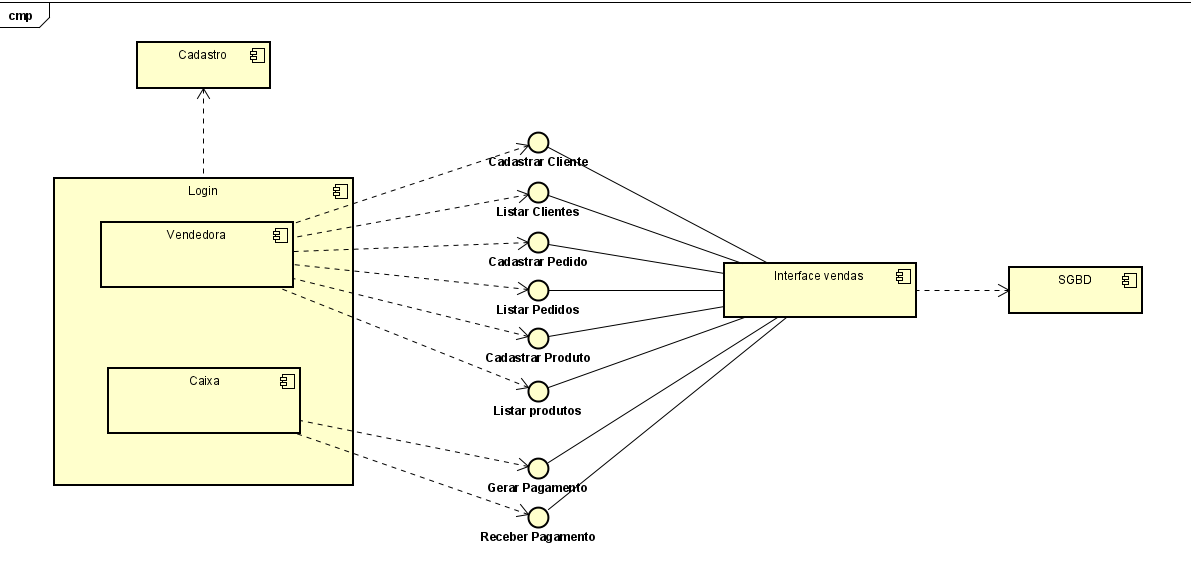


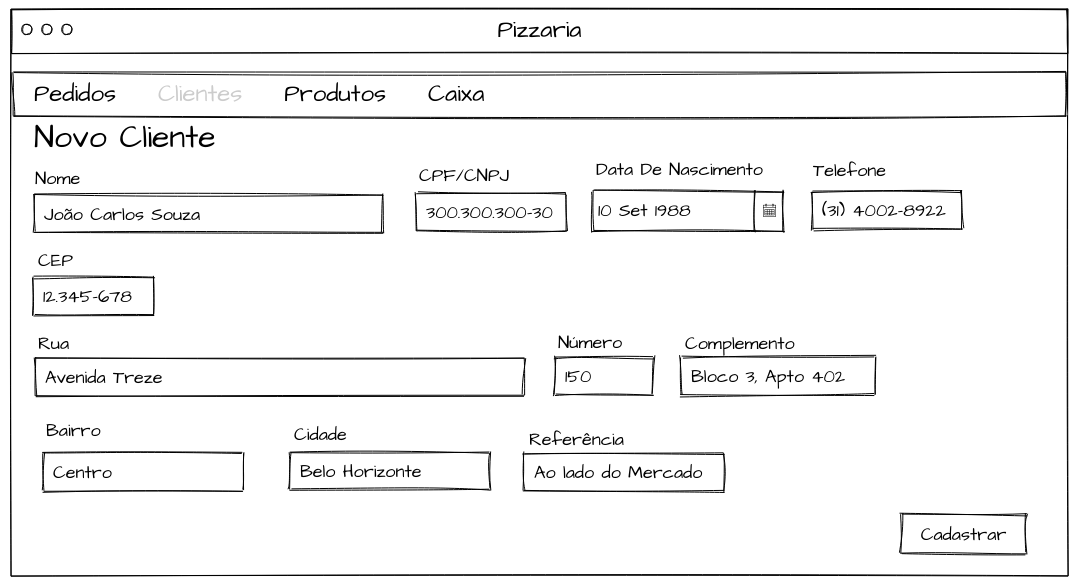
Figura 01 - Diagrama de Componentes

### Hospedagem

Por possuir suporter a todas as tecnologias envolvidas e uma plano gratuito, a hospedagem dos componentes da aplicação será toda na plataforma do Heroku. O Heroku é uma Plataforma como Serviço(PaaS) onde é permitido fazer hospedagem, configuração, testagem e publicação de aplicações escaláveis de maneira rápida e simples. Além disso, possui integração a um banco de dados PostgreSQL na plataforma, permitindo um acesso aos dados do sistema de baixa latência.

# Funcionalidades do Sistema (Telas)

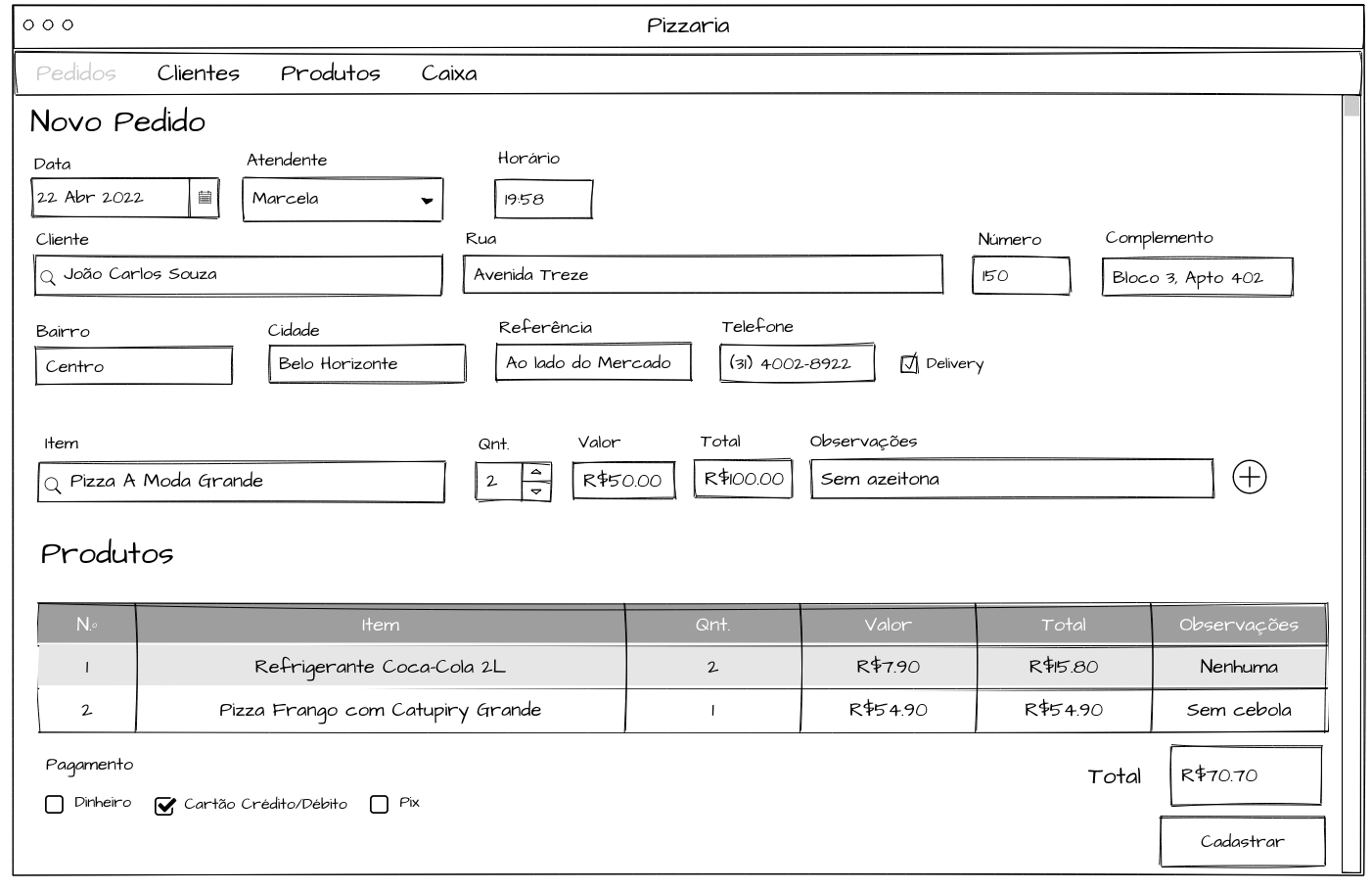
**1. Cadastro de Clientes**



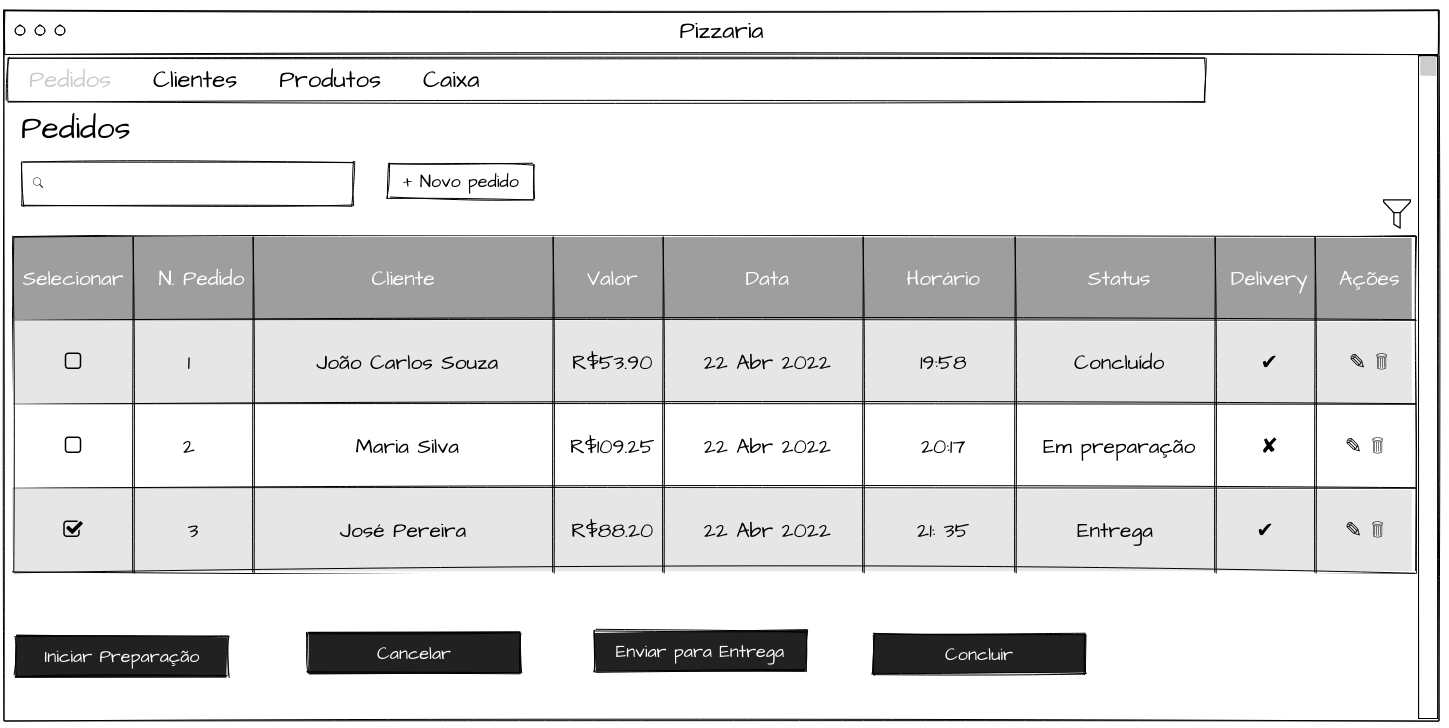
2. Lista de Clientes

# 

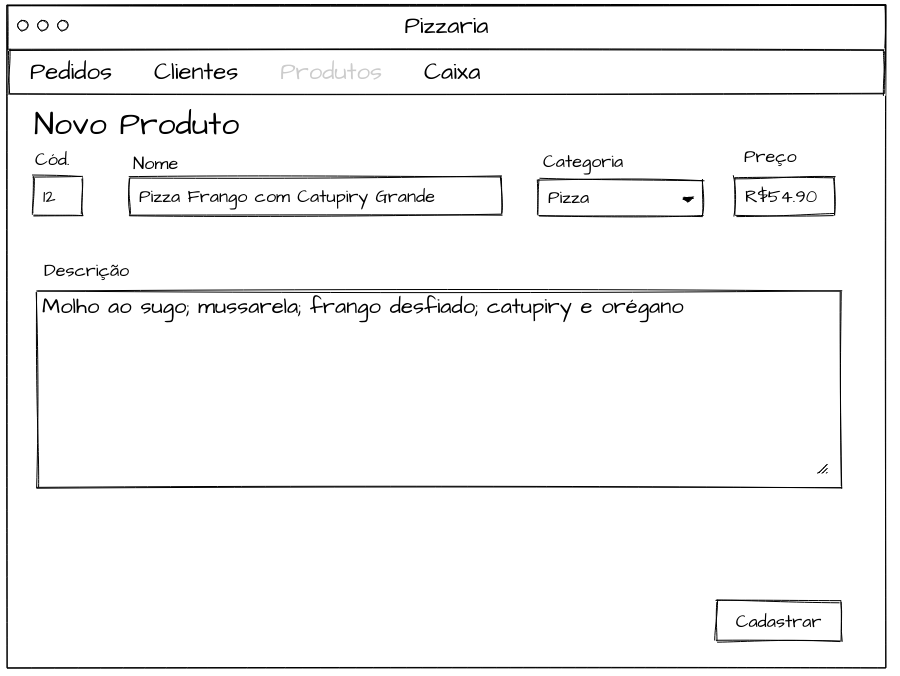
Cadastro de Pedido



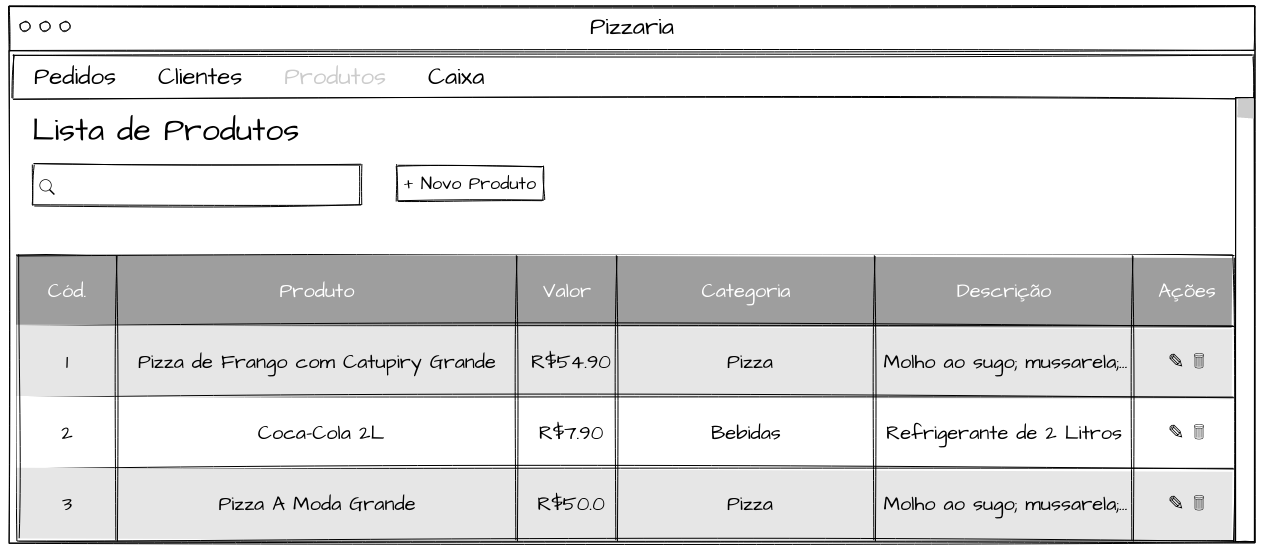
Lista de Pedidos



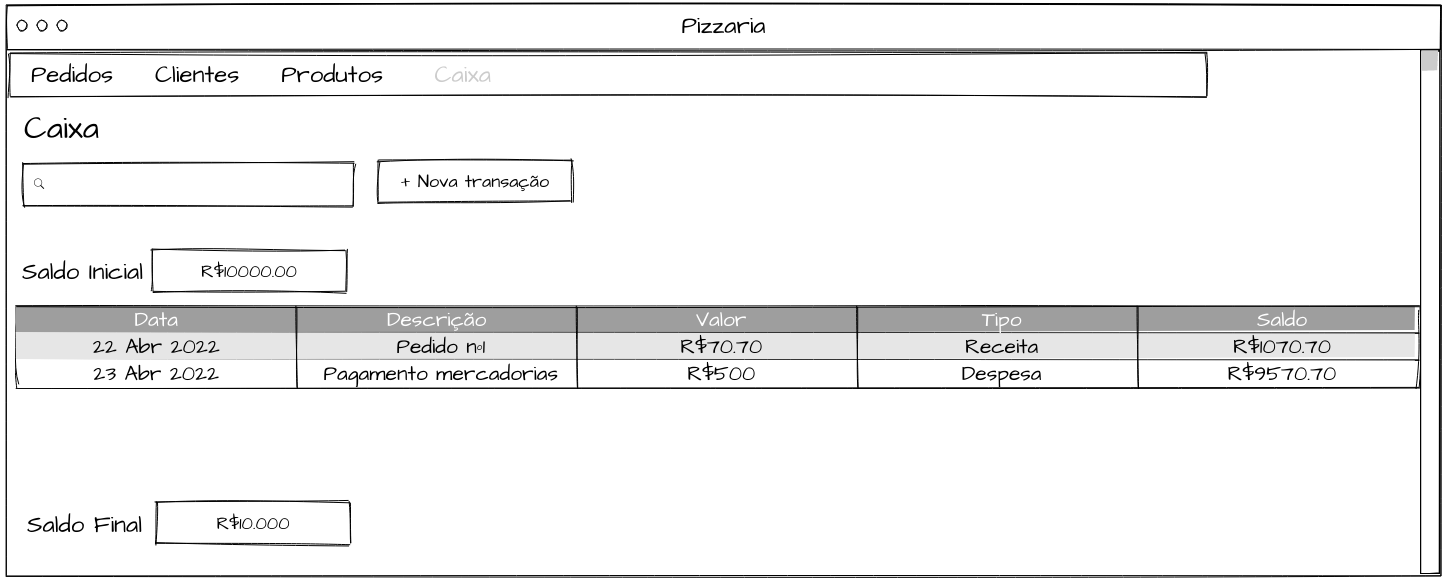
Cadastrar Produto



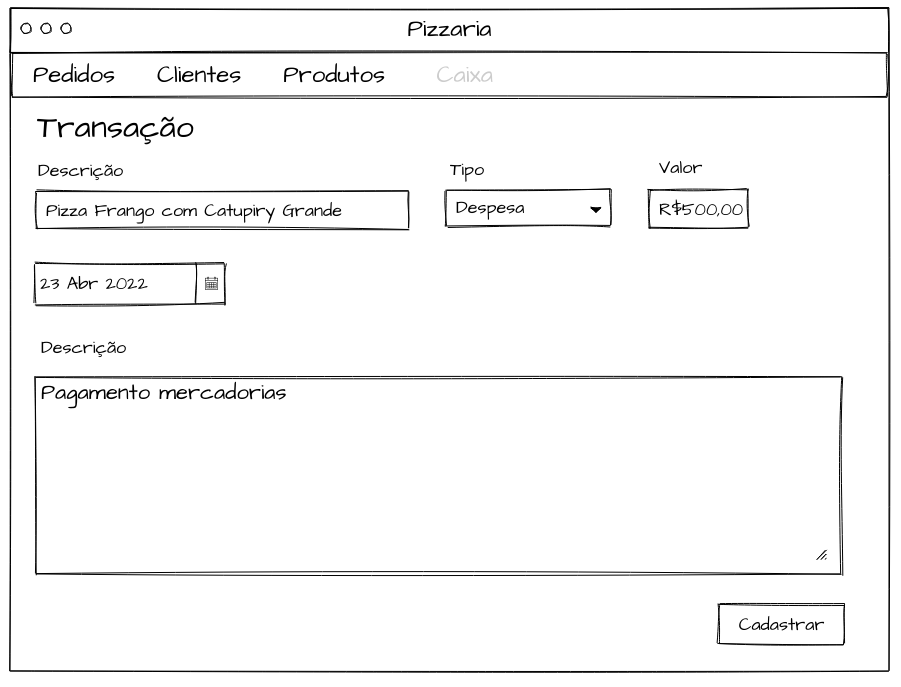
Lista

de Produtos

Caixa



Nova Transação



# Plano de Testes de Software

Os requisitos para realização dos testes de software são:

\*

\*

\*

Os testes funcionais a serem realizados no sistema são descritos a seguir:

| **Caso de Teste** | **CT-01 -** |
| --- | --- |
| **Requisitos Associados** |  |
| **Objetivo do Teste** |  |

**--------------- IMPORTANTE ---------------**  
Coloque todos os casos de testes planejados para o projeto.

**----------------------------------------------------**

# Registro de Testes de Software

Os resultados obtidos nos testes de software realizados são descritos abaixo.

**--------------- IMPORTANTE ---------------**  
Coloque o resultado de todos os casos de testes planejados para o projeto. Você pode mostrar o resultado através de um print de tela.

**----------------------------------------------------**

# Referências