# Documentação Técnica - Sistema da Pizzaria ‘Tratoria Rocca’

Lucas Terminiello, junho de 2025.  
Analise e desenvolvimento de sistemas – 1º DEV tarde  
Prof.º Bruno

SENAI  
Limeira-SP

**Índice**

1. [Introdução](#Xc2970a1a23df3ef3b34d1da3795ce3cec2678cb)
2. [Requisitos do Sistema](#Xe7e854738bc9a1ba8b8bb241637722b20abcff1)
   1. [Requisitos Funcionais (O que faz?)](#X36426f9f5eee1242d71bc8e71632e93425cd7cf)
   2. [Requisitos Não Funcionais (Como faz?)](#X3c9746c123c12f9deb18574315938dda4714b78)
3. [Arquitetura e Funcionamento](#Xbd1049d50d1a68f9e8163e7b32b71458c1f4720)
   1. [Visão Geral da Arquitetura](#X90b13d42250d2c59e49c3b7a467018a6c8e2cce)
   2. [Fluxo Principal de Operação](#Xbed978abbef2f15812d95b24fa6c507973bab49)
4. [Diagramas Visuais](#X3808b07f4c092a4bd105e46cbb571a813531037)
   1. [Fluxograma Geral](#X8443d80522fe261cbf85ce7000751345d75b28f)
   2. [Diagrama de Componentes](#X6cc55dbcdef93e5d05fb37b3551993aedca51f8)
   3. [Diagrama de Estado (Simplificado)](#X429f3e26ae0b511dcfd491e9a4afd8b2a9a5cb4)
   4. [Diagrama de Classe (Estruturas de Dados)](#X218360c7a4c72d079e82c6c047c10b590154d9e)
5. [Conclusão](#Xd01725661b91d9e97f8162e35c39665b7ae1f77)

## 1. Introdução

Bem-vindo à documentação do sistema de gerenciamento da Pizzaria Tratoria Rocca! Este documento foi criado pensando em estudantes, com o objetivo de explicar de forma clara e objetiva como o sistema funciona.

Trata-se de uma aplicação web de *front-end*, ou seja, tudo acontece diretamente no seu navegador (Chrome, Firefox, etc.). Não há um servidor complexo por trás. Ele simula as operações de uma pizzaria, permitindo que clientes vejam o cardápio e montem pedidos, e que administradores gerenciem as pizzas e opções.

Vamos explorar os requisitos (o que o sistema deve fazer), sua arquitetura (como ele é construído) e seu fluxo principal. Os diagramas visuais ajudarão a entender melhor a estrutura e o comportamento do sistema. O foco aqui é na visão geral, sem entrar em detalhes profundos de cada linha de código.

## 2. Requisitos do Sistema

Os requisitos definem o que o sistema precisa fazer e como ele deve se comportar. Eles foram levantados a partir da análise do código e das funcionalidades presentes.

### 2.1 Requisitos Funcionais (O que faz?)

* **Autenticação:** Permite login de Clientes e Administradores (com senhas fixas no código para teste) e redireciona para a tela correta.
* **Cliente:** Pode ver cardápio, filtrar pizzas, ver promoções, montar sua própria pizza (escolhendo tamanho, ingredientes, borda), adicionar itens ao carrinho, ver/editar o carrinho e “finalizar” o pedido (simulação).
* **Administrador:** Pode cadastrar, consultar, alterar e excluir pizzas do cardápio. Gerencia as opções disponíveis para montar pizza (tamanhos, ingredientes, bordas). Pode registrar vendas (simulação) e ver um relatório simples.
* **Geral:** Armazena dados (pizzas, carrinho, etc.) no navegador (localStorage) para que não se percam ao fechar e reabrir. Exibe mensagens (pop-ups) para confirmar ações ou informar erros.

*(Para uma lista detalhada de todos os requisitos funcionais, consulte o arquivo requisitos\_pizzaria.md gerado anteriormente)*

### 2.2 Requisitos Não Funcionais (Como faz?)

* **Usabilidade:** A interface busca ser simples e fácil de usar.
* **Tecnologia:** Construído apenas com HTML, CSS e JavaScript.
* **Persistência:** Usa o localStorage do navegador para guardar informações.
* **Segurança:** Baixa (próprio de um protótipo, com senhas no código).
* **Desempenho:** Rápido, pois tudo roda localmente no navegador.

## 3. Arquitetura e Funcionamento

Entender a arquitetura ajuda a visualizar como as partes do sistema se conectam.

### 3.1 Visão Geral da Arquitetura

O sistema é dividido em camadas simples que rodam no navegador:

1. **Apresentação (HTML/CSS):** Define a estrutura das páginas (login, cliente, admin) com HTML e a aparência (cores, fontes, layout) com CSS.
2. **Lógica (JavaScript):** Contém a “inteligência” do sistema. Os arquivos JS (PIZZARIA\_login.js, PIZZARIA\_cliente.js, PIZZARIA\_admin.js, popup\_function.js) cuidam das interações (cliques em botões), validações, cálculos (preço da pizza montada, total do carrinho) e manipulação dos dados.
3. **Armazenamento Local (LocalStorage):** Funciona como um pequeno “banco de dados” dentro do navegador. O JavaScript salva e lê informações aqui (como o cardápio, o carrinho, as opções de montagem) para que elas persistam entre as sessões.

### 3.2 Fluxo Principal de Operação

O caminho básico que um usuário percorre é:

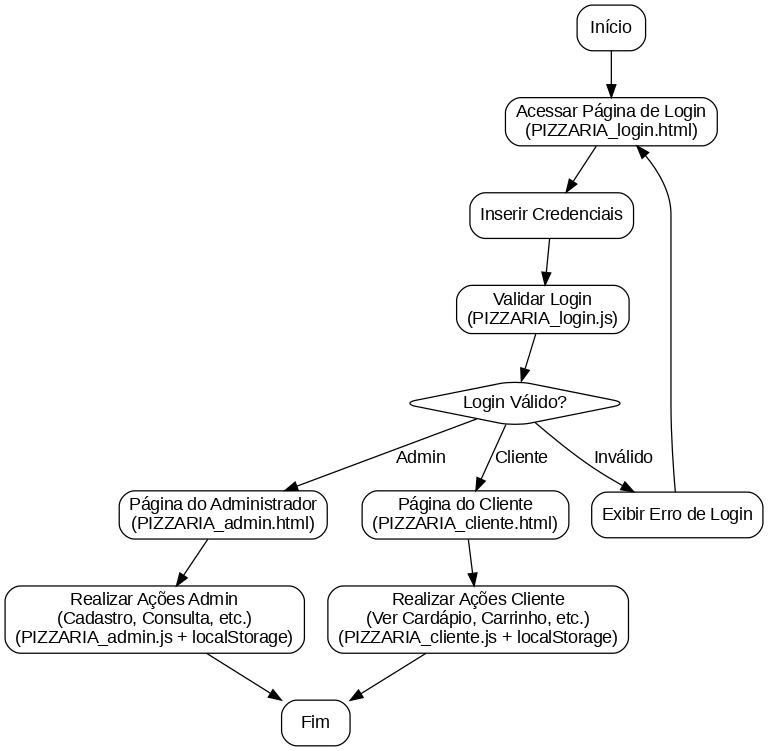
1. **Acesso:** Abre a página de login (PIZZARIA\_login.html).
2. **Login:** Insere usuário e senha (ou usa o botão Google simulado).
3. **Validação:** O PIZZARIA\_login.js verifica se os dados estão corretos.
4. **Redirecionamento:** Se o login for válido, vai para a página do Cliente (PIZZARIA\_cliente.html) ou do Administrador (PIZZARIA\_admin.html). Se inválido, mostra um erro.
5. **Interação:** O usuário realiza ações na sua respectiva página (ver cardápio, cadastrar pizza, etc.). O JavaScript correspondente executa a lógica e atualiza o localStorage.

## 4. Diagramas Visuais

Diagramas ajudam a visualizar a estrutura e o comportamento do sistema.

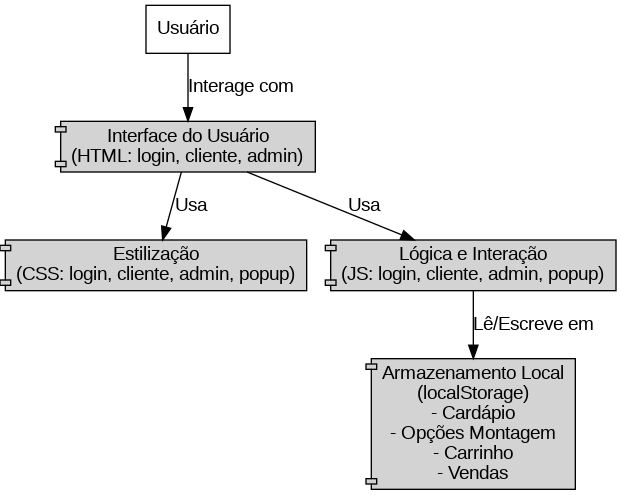
### 4.1 Fluxograma Geral

Mostra o fluxo principal de telas e decisões do usuário.

 *Figura 1: Fluxograma geral do sistema de pizzaria.*

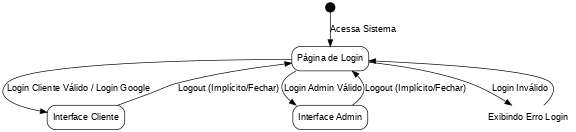
### 4.2 Diagrama de Componentes

Ilustra as principais partes do software e como elas se relacionam.

 *Figura 2: Diagrama de componentes do sistema de pizzaria.*

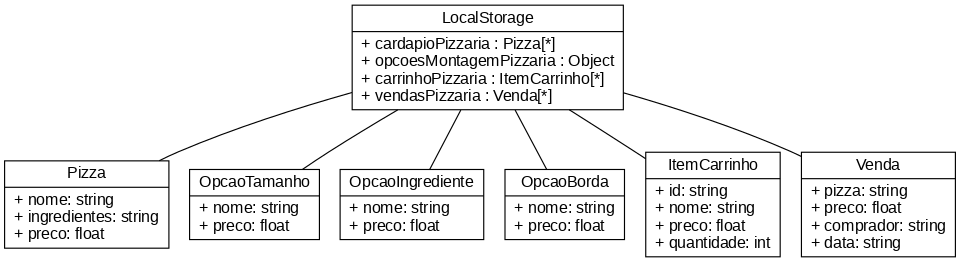
### 4.3 Diagrama de Estado (Simplificado)

Representa os principais estados em que a aplicação pode se encontrar.

 *Figura 3: Diagrama de estado simplificado da aplicação.*

### 4.4 Diagrama de Classe (Estruturas de Dados)

Mostra as principais estruturas de dados usadas e armazenadas pelo sistema (visão conceitual).

 *Figura 4: Diagrama de classe simplificado representando as estruturas de dados.*

## 5. Conclusão

Este documento apresentou uma visão geral do sistema de gerenciamento da Pizzaria Tratoria Rocca, focando nos requisitos, arquitetura, fluxo e representações visuais através de diagramas.

Como um protótipo front-end, ele demonstra as funcionalidades essenciais de interação do cliente e do administrador, utilizando o localStorage para simular a persistência de dados. Embora funcional para demonstração e estudo, é importante notar suas limitações para um uso em produção real.