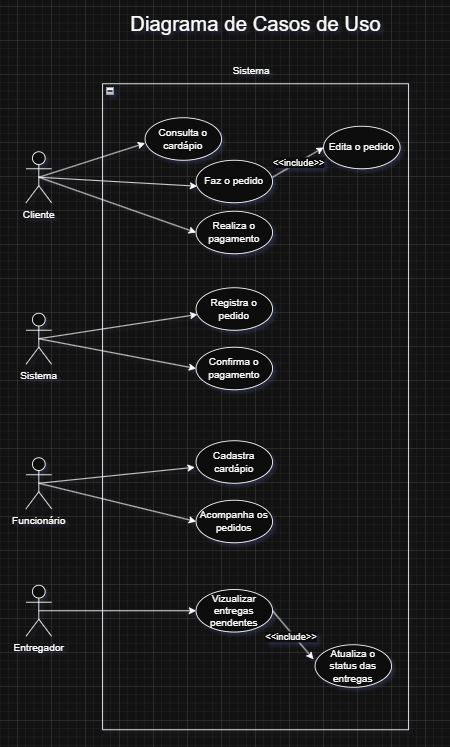
**Projeto da Arquitetura de Software – SliceHeaven**

**Integrantes:**  
Arthur Camargo  
Diego Polanski de Freitas Vieira  
Vitor Dias de Britto Militão  
Vitor Lucio de Oliveira

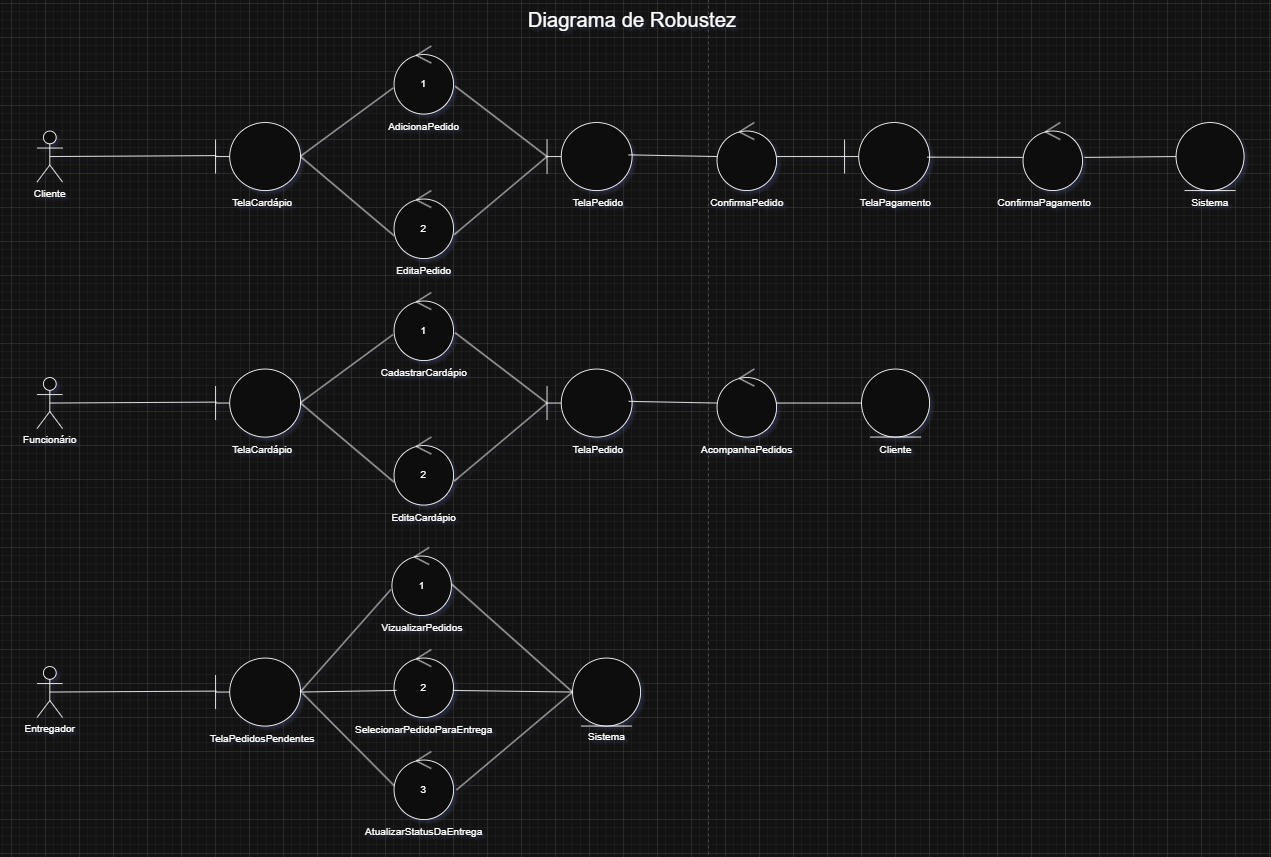
1. **Diagramas:**

**Todos os Diagramas podem ser visualizados também através desse link:**<https://drive.google.com/file/d/1OyoCAbErnHYUiud6h9NRrVp53v2k3ane/view?usp=sharing>

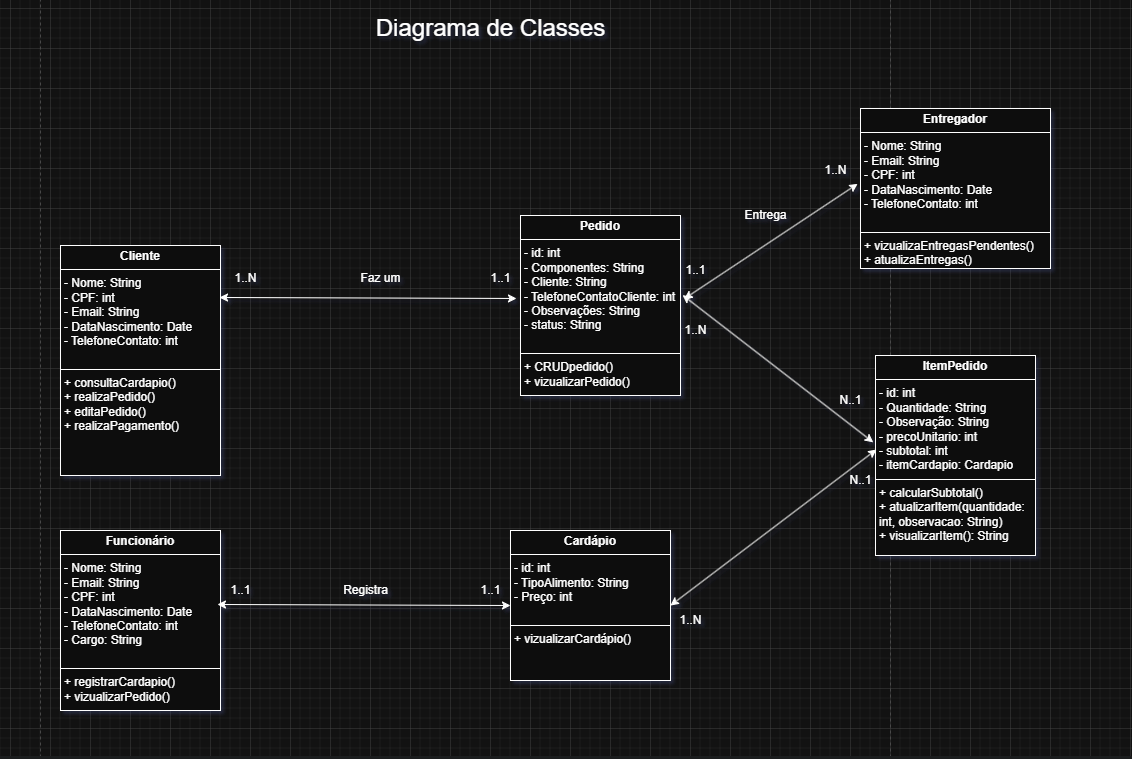
* 1. **– Diagrama de Casos de Uso**



* 1. **- Diagrama de Robustez**



* 1. **- Diagrama de Classes e pacotes**



**DIAGRAMA DE PACOTES AQUI**

**Modelo de Dados e Script SQL para SGBD**

*-- Tabela de pizzas  
CREATE TABLE* IF *NOT EXISTS* pizzas (  
 id BIGINT AUTO\_INCREMENT *PRIMARY KEY*,  
 name *VARCHAR*(100) *NOT NULL*,  
 description TEXT,  
 price *DECIMAL*(10,2) *NOT NULL*,  
 image\_url TEXT,  
 category *VARCHAR*(50) *DEFAULT* 'pizza',  
 sub\_category *VARCHAR*(50),  
 featured BOOLEAN *DEFAULT FALSE*,  
 rating *DECIMAL*(2,1),  
 discount *INTEGER*,  
 created\_at *TIMESTAMP DEFAULT CURRENT\_TIMESTAMP*) ENGINE=InnoDB;  
  
*-- Tabela de sobremesas  
CREATE TABLE* IF *NOT EXISTS* desserts (  
 id BIGINT AUTO\_INCREMENT *PRIMARY KEY*,  
 name *VARCHAR*(100) *NOT NULL*,  
 description TEXT,  
 price *DECIMAL*(10,2) *NOT NULL*,  
 image\_url TEXT,  
 category *VARCHAR*(50) *DEFAULT* 'dessert',  
 featured BOOLEAN *DEFAULT FALSE*,  
 discount *INTEGER*,  
 created\_at *TIMESTAMP DEFAULT CURRENT\_TIMESTAMP*) ENGINE=InnoDB;  
  
*-- Tabela de bebidas  
CREATE TABLE* IF *NOT EXISTS* drinks (  
 id BIGINT AUTO\_INCREMENT *PRIMARY KEY*,  
 name *VARCHAR*(100) *NOT NULL*,  
 description TEXT,  
 price *DECIMAL*(10,2) *NOT NULL*,  
 image\_url TEXT,  
 category *VARCHAR*(50) *DEFAULT* 'drink',  
 featured BOOLEAN *DEFAULT FALSE*,  
 discount *INTEGER*,  
 created\_at *TIMESTAMP DEFAULT CURRENT\_TIMESTAMP*) ENGINE=InnoDB;

*-- Tabela de usuarios*

CREATE TABLE users (

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(100) NOT NULL,

email VARCHAR(100) UNIQUE NOT NULL,

password\_hash VARCHAR(255) NOT NULL,

address TEXT,

phone VARCHAR(20),

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

**Documentação da Stack de Tecnologias – PizzaToday**

Este documento detalha a stack de tecnologias escolhida para o desenvolvimento do site de entrega de pizza, para a pizzaria PizzaToday. A escolha das tecnologias foi feita com o objetivo de criar um sistema robusto, escalável e de alta performance, adequado às necessidades do cliente e do negócio.

**1. Visão Geral**

O sistema será desenvolvido utilizando uma arquitetura de microsserviços, com o front-end separado do back-end, permitindo maior flexibilidade e escalabilidade. O front-end será responsável pela interface e interação com o usuário, enquanto o back-end cuidará da lógica de negócios e do gerenciamento de dados.

**2. Tecnologias Utilizadas**

**2.1. Front-end**

* **Linguagem:** JavaScript (ES6+)
  + Utilização de recursos modernos da linguagem para maior eficiência e legibilidade do código.
* **Framework:** React
  + Criação de interfaces de usuário dinâmicas e responsivas, proporcionando uma excelente experiência ao cliente.
* **Gerenciamento de Estado:** Redux ou Context API //Vai depender do que a Fernanda usa na videoaula
  + Gerenciamento eficiente do estado da aplicação, garantindo a consistência dos dados e facilitando a manutenção.
* **Estilização:** CSS Modules, Styled Components
  + Criação de componentes reutilizáveis e estilização consistente, garantindo uma aparência moderna e profissional.
* **Ferramentas de Build:** Vite
  + Otimização do processo de build, garantindo a performance da aplicação.
* **Gerenciador de Pacotes:** npm
  + Gerenciamento eficiente das dependências do projeto.

**2.2. Back-end**

* **Linguagem:** Java (JDK 17 ou superior)
  + Linguagem robusta e escalável, ideal para o desenvolvimento de sistemas complexos.
* **Framework:** Spring Boot
  + Desenvolvimento rápido e eficiente de APIs RESTful, facilitando a comunicação com o front-end.
* **ORM:** Spring Data JPA
  + Mapeamento objeto-relacional, simplificando o acesso e a manipulação dos dados no banco de dados.
* **API:** RESTful API
  + Comunicação eficiente entre o front-end e o back-end, utilizando o padrão REST.
* **Autenticação:** Spring Security (JWT)
  + Implementação de um sistema de autenticação seguro, utilizando tokens JWT.
* **Servidor de Aplicação:** Tomcat
  + Servidor web java para implantação do back-end.

**2.3. Banco de Dados**

* **SGBD:** MySQL
  + Banco de dados relacional robusto e escalável, ideal para armazenar dados de pedidos, clientes e produtos.