<DineSmart>

Visão (Projeto Pequeno)

Versão <2.0>

Histórico da Revisão

| **Data** | **Versão** | **Descrição** | **Autor** |
| --- | --- | --- | --- |
| <30/03/2023> | <1.0> | <Início da documentação> | <Felipe> |
| <18/05/2023> | <1.5> | <Revisão da documentação | <Felipe> |
| <12/06/2023 | <2.0> | <Término da documentação oficial> | <Felipe> |
|  |  |  |  |

Índice Analítico

[**1. Introdução 4**](#_heading=h.30j0zll)

[1.1 Referências 4](#_heading=h.1fob9te)

[**2. Posicionamento 4**](#_heading=h.3znysh7)

[2.1 Descrição do Problema 4](#_heading=h.2et92p0)

[2.2 Sentença de Posição do Produto 4](#_heading=h.tyjcwt)

[**3. Descrições dos Envolvidos e Usuários 4**](#_heading=h.3dy6vkm)

[3.1 Resumo dos Envolvidos 4](#_heading=h.1t3h5sf)

[3.2 Resumo dos Usuários 5](#_heading=h.4d34og8)

[3.3 Ambiente do Usuário 5](#_heading=h.2s8eyo1)

[3.4 Resumo das Principais Necessidades dos Envolvidos ou Usuários 5](#_heading=h.17dp8vu)

[3.5 Alternativas e Concorrência 6](#_heading=h.3rdcrjn)

[**4. Visão Geral do Produto 6**](#_heading=h.26in1rg)

[4.1 Perspectiva do Produto 6](#_heading=h.lnxbz9)

[4.2 Suposições e Dependências 7](#_heading=h.35nkun2)

[**5. Recursos do Produto 7**](#_heading=h.1ksv4uv)

[**6. Outros Requisitos do Produto 7**](#_heading=h.44sinio)

[6.1 Requisitos Funcionais 9](#_heading=h.6ys0aqnu5uru)

[6.2 Requisitos Não Funcionais 9](#_heading=h.o6lup6fyqumv)

[**7. Diagramas 9**](#_heading=h.h4j03fdcb53d)

[7.1 Diagrama de Caso de Usos 9](#_heading=h.s2zado4rqe7d)

[7.2 Diagrama de Classe 10](#_heading=h.fu7y105fwooj)

[7.3 Diagrama de atividade 11](#_heading=h.3uby1wwgwuzc)

[7.4 Diagrama de estado 12](#_heading=h.rqpjfv6n7g2x)

[7.5.0 Diagramas de Sequência 13](#_heading=h.yfva1pj77enq)

[7.5.1 Diagrama de Sequência Login/Cadastro 13](#_heading=h.sw0ocfl0g9da)

[7.5.2 Diagrama de Sequência Cadastro de itens 14](#_heading=h.f6ilyv1u66vs)

[7.5.3 Diagrama de Sequência Fazer Reserva 15](#_heading=h.jbfs6u2wap3)

[7.5.4 Diagrama de Sequência Check-in 16](#_heading=h.iywfq6ing2i7)

[7.5.5 Diagrama de Sequência Cancelar Reserva 17](#_heading=h.l4z3t7t33bkh)

[7.5.6 Diagrama de Sequência Check-out 18](#_heading=h.39toebir1zof)

[7.5.7 Diagrama de Sequência Consumo 19](#_heading=h.auvnszc66vp)

Visão (Projeto Pequeno)

# Introdução

A finalidade deste documento é coletar, analisar e definir as necessidades e características de nível superior do Restaurante App. Ele enfoca os recursos de que os envolvidos e usuários-alvo precisam e mostra **por que** essas necessidades existem. Os detalhes de como o Restaurante App atende a essas necessidades estão descritas nas especificações suplementares e de caso de uso.

## Referências

Link GitHub: <https://github.com/devmitz/DineSmart>

Prototipo Figma: <https://www.figma.com/file/CaX8BYtnLvXvpoDQf1XP5H/A3---Restaurante?type=design&node-id=0-1>

# Posicionamento

## Descrição do Problema

| O problema | A grande demora em filas de restaurantes durante a pós pandemia |
| --- | --- |
| afeta | Restaurantes e Clientes. |
| cujo impacto é | Por conta da grande demora em filas, a contaminação de vírus como o Covid-19 se torna mais fácil por conta dos clientes estarem próximos uns aos outros na fila.  Com o app as filas diminuirão e a contaminação e propagação dos vírus também. |
| uma boa solução seria | Sem filas, agilidade, praticidade, intuitivo. |

## Sentença de Posição do Produto

| Para | Consumidores de restaurantes (Principalmente em horário de pico) |
| --- | --- |
| Quem | Clientes |
| O DineSmart | é um aplicativo de reserva em diversos restaurantes |
| Que | A entrada e saída rápida dos estabelecimentos. |
| Diferente de | IFood, Rappi, UberEats… |
| Nosso produto | O nosso produto não é um delivery ou retirada do produto no restaurante, o nosso produto é uma organização de filas do estabelecimento de forma online, sem a necessidade do cliente estar no local em uma fila. |

# Descrições dos Envolvidos e Usuários

## Resumo dos Envolvidos

| **Nome** | **Descrição** | **Responsabilidades** |
| --- | --- | --- |
| Desenvolvedores | Irá fazer a parte do front-end e back-end. | Garante a produção e manutenção do aplicativo. |
| Analista de Dados | Irá realizar o banco de dados. | Responsável pelos bancos de dados necessários para o funcionamento do app. |
| Gerente de Projeto | Irá comandar (Desenvolvedores, Analistas de Dados) e monitorar o andamento do desenvolvimento e manutenção. | Monitora o andamento do projeto. |

## Resumo dos Usuários

| **Nome** | **Descrição** | **Responsabilidades** | **Envolvido** |
| --- | --- | --- | --- |
| Clientes | O cliente é o usuário que utiliza casualmente o aplicativo. | O mesmo tem a função de fazer reservas em restaurantes e comprar produtos por ele. | N/A |

## Ambiente do Usuário

Na visão do usuário, tudo será bastante intuitivo. Atualmente a quantidade de pessoas ideal para realizar o uso do sistema são 2, sendo uma para questão básica de uso e outra para a administração, há uma previsão de que possa aumentar esse número devido a novas utilidades do sistema.

O tempo de duração das tarefas do sistema acaba variando bastante, depende da pessoa e da situação em que ela se encontra, mas normalmente leva de 5 a 7 minutos. Já em questões de restrições, não restringimos nada atualmente, se no futuro for necessário, haverá atualizações sobre.

O aplicativo se encontra em plataformas como PlayStore e sua base de dados é localizada na FireBase. Pensamos no futuro em implantar o sistema para iOS também, deixando ele disponível na AppleStore.

## Resumo das Principais Necessidades dos Envolvidos ou Usuários

| **Necessidade** | **Prioridade** | **Preocupações** | **Solução Atual** | **Soluções Propostas** | |
| --- | --- | --- | --- | --- | --- |
| Interação entre usuário e restaurante | Alta | Problemas de conexão de chat com o aplicativo | Monitoramento e investimento em uma conexão estável, com acesso ao suporte pelo usuário. | |  |
| Segurança com “scam” e golpes de restaurantes | Alta | Perfis de restaurantes que coloquem links para sites externos para que possam roubar informações | O restaurante precisa confirmar seu comércio físico, e o aplicativo não deixará a inserção de links. | |  |
| Recomendações do app | Baixa | O aplicativo deve mostrar restaurantes com uma gastronomia igual a que o usuário frequenta, e dar desconto aos mesmos. | Utilizar um método de verificação para descobrir quais restaurantes são parecidos e dar desconto baseado nos pratos. | |  |

## Alternativas e Concorrência

IFood, é um app de pedidos, ele garante que o usuário peça um certo produto (Alimentício) e receba na porta de sua casa já pronto, ele garante a retirada do produto se o cliente desejar, informando o horário que o pedido ficará pronto, ele também garante o pagamento do cliente para o restaurantes e motoboys.

O principal ponto forte é a praticidade e segurança, você conseguir pagar online e receber o produto já pronto na porta de sua casa torna a vida do cliente mais prática com apenas alguns cliques.

# Visão Geral do Produto

## Perspectiva do Produto

Nossa proposta de solução para a retomada pós-pandemia tem por objetivo orientar estabelecimentos comerciais do segmento de restaurantes a organizar e controlar, por meio da ferramenta digital, a lotação e pedidos no estabelecimento, a fim de evitar a formação de filas e longo tempo de espera no local. Além disso, pensamos em um novo sistema em que se é possível, como cliente, escolher o pedido e gerar uma comanda mesmo que ainda não esteja no local, com o intuito de reduzir o tempo de espera nos estabelecimentos.

Através da aplicação, restaurantes, bares e outros estabelecimentos do tipo poderão cadastrar a marca – tendo, assim, a possibilidade de divulgá-la – e realizar um melhor controle de pedidos, atualização do cardápio (que será exibido em uma interface para a clientela) entre outros.

A clientela, por outro lado, poderá realizar um cadastro próprio de usuário para:

1 - Pesquisar por bares, restaurantes etc. em qualquer região;

2 - Ter uma estimativa da lotação do estabelecimento no momento da pesquisa (ex.: cheio, não muito cheio);

3 - Realizar reserva para o local escolhido (bem como a data e o horário);

4 - Consultar o cardápio do estabelecimento;

5 - Realizar pedido (mesmo em casa ou a caminho do local);

6 - Comprovar a reserva, com o aplicativo em mãos, no local;

7 - Avaliar (com estrelas) o atendimento do estabelecimento;

## Suposições e Dependências

O aplicativo necessita da dependência de Android superior a versão 9.

# Recursos do Produto

-Capacidade de filtrar restaurantes próximos ao usuário - Este recurso listará para o usuário os restaurantes mais próximos

-Obter a estimativa da lotação do estabelecimento - Permite que o usuário saiba se o ambiente está com um fluxo alto ou baixo de pessoas no momento

-Acesso ao restaurante remotamente - Permite que o usuário faça seu pedido ou reserva mesmo estando longe do local, como objetivo de economizar tempo e diminuir filas.

-Sistema de avaliação - Recurso de avaliação(com estrelas) para obter um feedback do cliente ao restaurante.

# Outros Requisitos do Produto

O requisito mínimo é de um aparelho smartphone, com 1GB de memória RAM, e 1GB de memória HD e acesso a Internet.

Recomendamos para uma melhor utilização o uso de aparelhos com 4GB de memória RAM e uma internet estável.

Referente a isso temos alguns testes relacionados a capacidade de utilização, desempenho, compatibilidade, segurança e automação.

Teste de capacidade de utilização: Tem como objetivo sobrecarregar o uso do aplicativo em diferentes smartphones, para garantir que o site continue funcionando mesmo com um número excessivo de usuários.

Teste de desempenho: Tem como objetivo testar a otimização do aplicativo com a intenção de descobrir se o mesmo funciona/inicia de forma rápida e prática nos smartphones dos usuários.

Teste de compatibilidade: Tem como objetivo realizar downloads em dispositivos diversos com capacidades diferentes, com o intuito de conseguir uma maior acessibilidade para todos os usuários.

Teste de segurança: Tem o intuito de forçar ao máximo o sistema com inúmeros tipos de ataque, como; DDoS e ataques de força bruta. Dessa forma conseguimos entender a capacidade de que temos de aguentar algum tipo de hack.

Teste de automação: Tem como objetivo saber se o aplicativo consegue criar um ambiente confortável para os usuários e ser útil aos mesmos, os dando recomendações de acordo com o gosto e disponibilizando descontos e promoções baseado no que o usuário procura e consome.

A única dependência para a instalação do app é do sistema android superior a versão 9.

Ajuda on-line, o cliente tem suporte ao abrir um ticket enviando um email para gente, basta explicar o problema que o cliente teve e ele receberá toda ajuda necessária e possível para a resolução do problema.

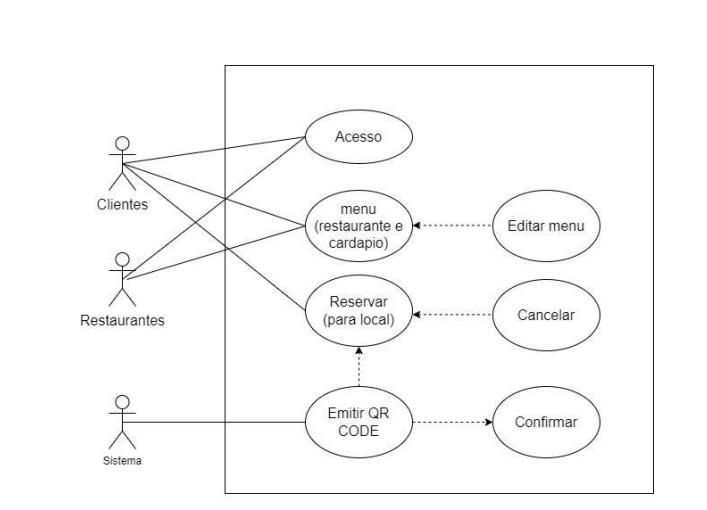
Instalação, basta o usuário abrir o instalador de aplicativos nativo do próprio smartphone e pesquisar pelo nome “Restaurante App” e verificar que o aplicativo selecionado garante a empresa x , após isso o cliente tem apenas que clicar no botão de instalação que o nosso app já instalado de forma automática todas a dependências necessárias.

## 6.1 Requisitos Funcionais

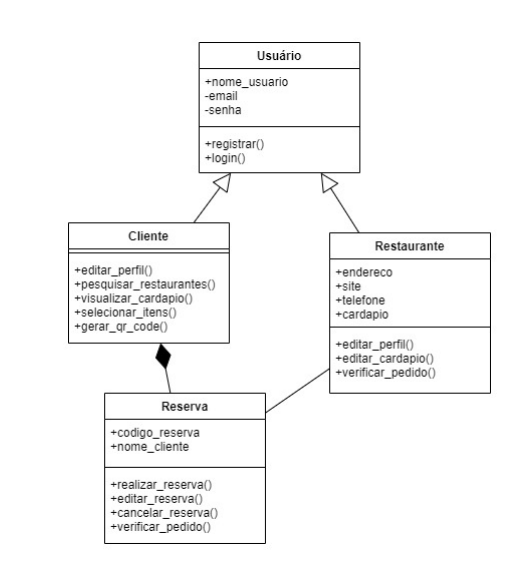
## 6.2 Requisitos Não Funcionais

# 7. Diagramas

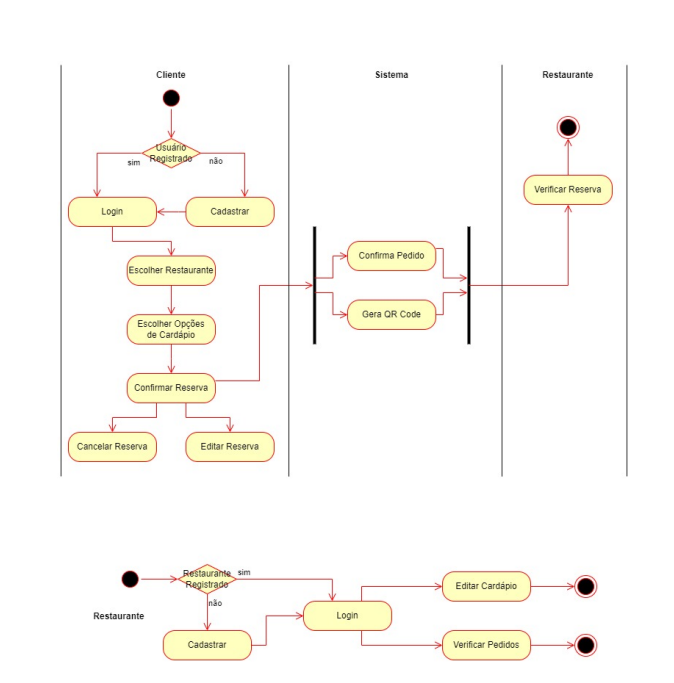
## 7.1 Diagrama de Caso de Usos



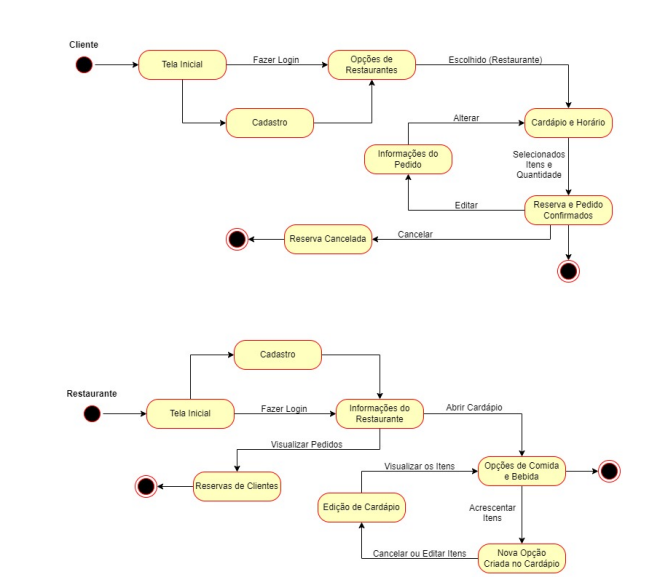
## 7.2 Diagrama de Classe



## 7.3 Diagrama de atividade

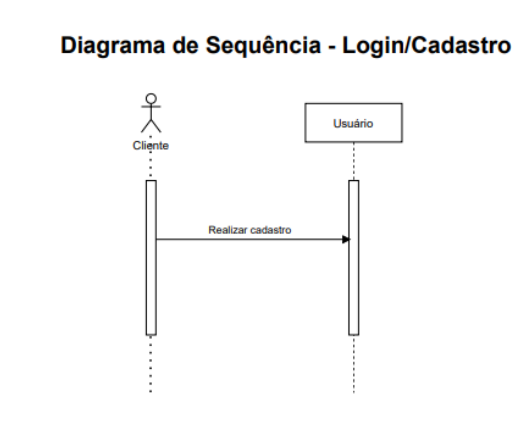


## 7.4 Diagrama de estado

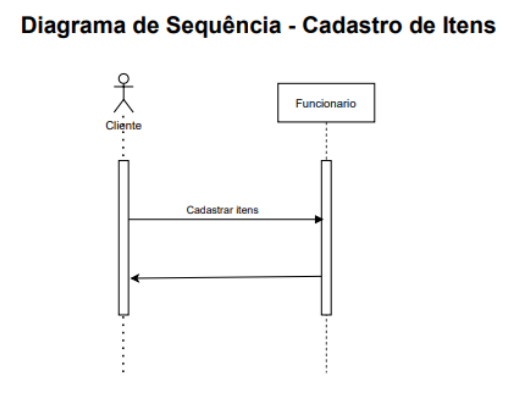


## 7.5.0 Diagramas de Sequência

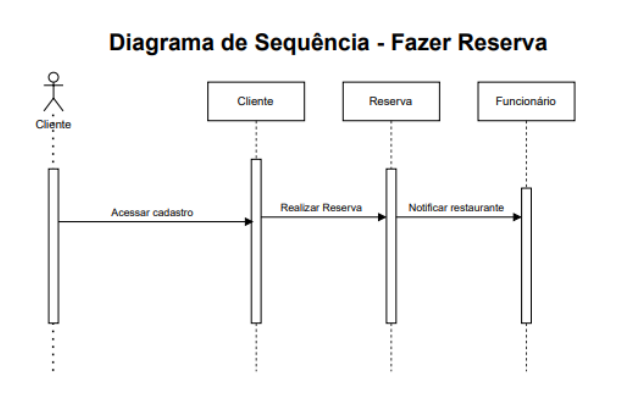
## 7.5.1 Diagrama de Sequência Login/Cadastro



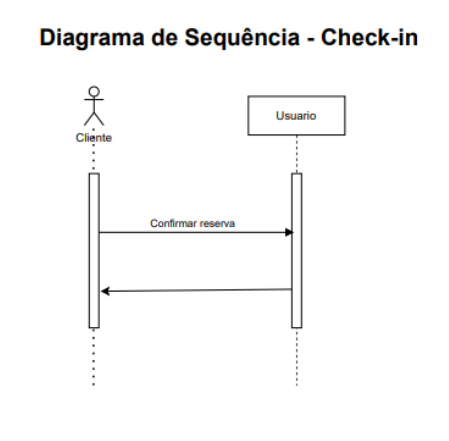
## 7.5.2 Diagrama de Sequência Cadastro de itens



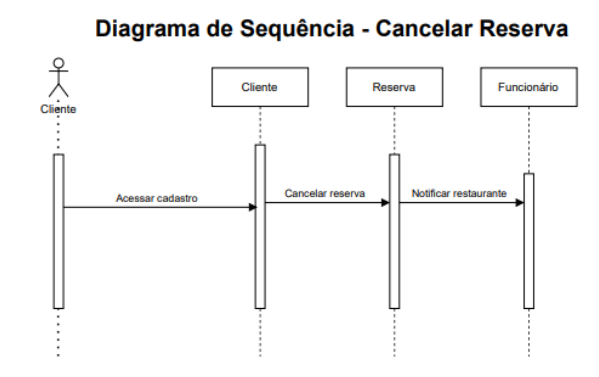
## 7.5.3 Diagrama de Sequência Fazer Reserva



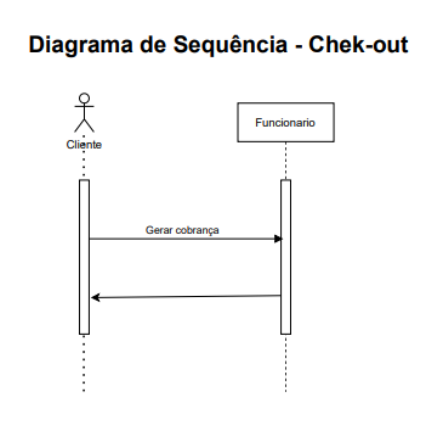
## 7.5.4 Diagrama de Sequência Check-in



## 7.5.5 Diagrama de Sequência Cancelar Reserva



## 7.5.6 Diagrama de Sequência Check-out



## 7.5.7 Diagrama de Sequência Consumo

