

MapMarket

# **VISÃO DO PRODUTO E PROJETO**

Versão 1.0

## Histórico de Revisão

<b>Data</b>	<b>Versão</b>	<b>Descrição</b>	<b>Autor</b>

## **Sumário**

<b>1</b>	<b><i>VISÃO GERAL DO PRODUTO</i></b>	<b>4</b>
1.1	<b>Declaração do Problema</b>	<b>4</b>
1.2	<b>Declaração de Posição do Produto</b>	<b>4</b>
1.3	<b>Objetivos do Produto</b>	<b>4</b>
<b>2</b>	<b><i>ABORDAGEM DE DESENVOLVIMENTO DE SOFTWARE</i></b>	<b>4</b>
2.1	<b>Metodologia</b>	<b>5</b>
2.2	<b>Processo</b>	<b>5</b>
2.3	<b>Procedimentos</b>	<b>5</b>
<b>3</b>	<b><i>LIÇÕES APRENDIDAS</i></b>	<b>5</b>
3.1	<b>Unidade 1</b>	<b>5</b>
3.2	<b>Unidade 2</b>	<b>5</b>
3.3	<b>Unidade 3</b>	<b>6</b>
3.4	<b>Unidade 4</b>	<b>6</b>
3.5	<b>Unidade 5</b>	<b>6</b>
<b>4</b>	<b><i>REFERÊNCIAS BIBLIOGRÁFICAS</i></b>	<b>6</b>

# VISÃO DO PRODUTO E PROJETO

## 1 VISÃO GERAL DO PRODUTO

### 1.1 Declaração do Problema

Usuários de serviços de entrega de supermercados têm dificuldade em escolher produtos como frutas e legumes, além da dificuldade na comunicação entre cliente e supermercado.

O problema	Dificuldade de comunicação entre mercado e cliente na confirmação dos produtos pedidos, dificuldade de validação da lista de compras pedida.
Afeta	Clientes, pois não há certeza dos estados dos produtos e se serão entregues.
Cujo impacto é	Aumento da insatisfação com serviço de entrega
Uma solução de sucesso seria	Uma aplicação web responsiva capaz de dar a possibilidade de escolha dos estados dos produtos, e produtos específicos.

### 1.2 Declaração de Posição do Produto

Será desenvolvido uma aplicação web responsiva para auxiliar serviços de entrega. A aplicação contará com possibilidade dos mercados e atacados disponibilizarem seus catálogos de produtos e serviços, será possível montar uma lista completa de compras, conectar mercado e cliente de forma a escolher produto específico ou informar eventuais problemas, será possível agendar uma compra.

Para	Solucionar a dificuldade de comunicação entre mercado e cliente e dar maior autonomia às escolhas.
Quem	Utiliza serviços de entrega
O (nome do produto)	MapMarket
Que	É capaz possibilitar a criação de listas de compras recorrentes, facilitar a comunicação entre cliente e mercado, e possibilitar escolher o estado de produtos como frutas e verduras.
Ao contrário	Supermercado Now, Rappi
Nosso produto	Será uma aplicação web responsiva que possibilita integrada com uma API do mercado para controle de estoque, criação de uma lista de compras recorrentes, é capaz de agendar uma compra, disponibiliza um chat para a comunicação entre cliente e mercado, contém sistema de pontuação e descontos.

### 1.3 Objetivos do Produto

1. *Facilitar a integração mercado-cliente no ato da compra.*
2. *Garantir a satisfação do cliente ao comprar produtos hortifruti.*
3. *Fidelizar o cliente ao mercado, por meio de programas de pontos e descontos.*

## 1.4 Escopo do Produto

### 1. Requisitos Funcionais

- a. Deve ser possível realizar CRUD de cliente;
- b. Deve ser possível realizar CRUD de lista de compras;
- c. Deve ser possível realizar CRUD de lista de compras recorrentes;
- d. Deve ser possível realizar CRUD de agendamento de compras;
- e. Deve ser possível pesquisar por categoria do produto (ex: limpeza, higiene...)
- f. Deve ser possível pesquisar por nome do produto;
- g. Deve ser possível adicionar comentários aos produtos (ex: escolher data de validade, melancia gigante);
- h. Aplicação deve possuir um conjunto de dados pré-cadastrados;
- i. Aplicação terá um sistema de pontuação e descontos;
- j. Aplicação terá um sistema de chat online para comunicação cliente-mercado;

### 2. Requisitos não funcionais

- a. Aplicação deve ser web responsiva;
- b. Aplicação deve ser suportada em: Chrome, Safari e Firefox;
- c. Aplicação deve seguir em conformidade com a LGPD;
- d. A interação com o usuário deverá ser feita por meio de interface gráfica;
- e. O backend será em Node.JS e o fronted será em React;

## 1.5 Mínimo Produto Viável (MVP)

### 1. Requisitos Funcionais

- a. Deve ser possível realizar CRUD de cliente;
- b. Deve ser possível realizar CRUD de lista de compras;
- c. Deve ser possível pesquisar por nome do produto;
- d. Aplicação deve possuir um conjunto de dados pré-cadastrados;
- e. Aplicação terá um sistema de chat online para comunicação cliente-mercado;

### 2. Requisitos não funcionais

- a. Aplicação deve ser web responsiva;
- b. O backend será em Node.JS e o fronted será em React;

## 2 ABORDAGEM DE DESENVOLVIMENTO DE SOFTWARE

### 2.1 Metodologia

#### 2.1.1 Questões Técnicas:

- 2.1.2 Qual é o tamanho do sistema que está sendo desenvolvido? Será um projeto relativamente grande. Teremos cliente, supermercado e estoque.
- 2.1.3 Que tipo de sistema está sendo desenvolvido? Aplicação Web responsiva.
- 2.1.4 Qual a vida útil prevista para o sistema? 4 meses
- 2.1.5 O sistema está sujeito a controle externo? Sim, o supermercado teria controle sobre os produtos cadastrados e estoque.

#### 2.2 Questões Humanas:

- 2.2.1 Qual é o nível de competência dos projetistas e programadores do time de desenvolvimento? Todos os integrantes possuem um conhecimento intermediário.
- 2.2.2 Como está organizado o time de desenvolvimento? Dividimos em PO, SM e desenvolvedores. Por ser um time pequeno, o PO e o SM também serão desenvolvedores.
- 2.2.3 Quais são as tecnologias disponíveis para apoiar o desenvolvimento do sistema? Node.JS e React para a parte de programação; Discord e WhatsApp para comunicação e GitHub para versionamento.

#### 2.3 Questões Organizacionais:

- 2.3.1 É importante ter uma especificação e um projeto (design) bem detalhados antes de passar para a implementação - talvez por motivos contratuais? Utilizando o MVP, não precisaremos fazer a especificação detalhada do design da aplicação. Criaremos o backlog que será atualizado ao longo das sprints.
- 2.3.2 É realista uma estratégia de entrega incremental, na qual o software é entregue aos clientes ou outros stakeholders e um rápido feedback é obtido? Sim, o PO será parte fundamental, uma vez que também será parte do time de desenvolvimento.
- 2.3.3 Os representantes do cliente estarão disponíveis e dispostos a participar do time de desenvolvimento? Sim, o grupo é formado por três participantes, logo, todos serão parte do time de desenvolvimento.
- 2.3.4 Existem questões culturais que possam afetar o desenvolvimento do sistema? Não.

O grupo optou por usar uma metodologia ágil, essa sendo o Scrum, para o desenvolvimento do software, pois os integrantes do grupo já têm uma maior familiaridade com o método. Por ser uma metodologia ágil os requisitos de software são mais flexíveis tornando mais prático caso o grupo precise fazer alguma modificação, também nos permitirá ter uma visão mais clara dos prazos para cada requisito segundo Ian Sommerville. A comunicação e o sistema menos rigorosos para se dividir cada parte do desenvolvimento dos requisitos, ajudará ao projeto, pois o conhecimento sobre as tecnologias que serão utilizadas está homogêneo entre os integrantes.

Para o projeto foi pensado em termos sprints de duração de uma semana, um prazo mais confortável para todos do grupo, e terá rotatividade de Scrum Master, Product Owner, e membros do desenvolvimento prático, mas apesar dessa rotatividade quem ficar com esses papéis ainda irá ter participação no desenvolvimento da sprint. O objetivo dessa rotação de papéis é que todos os membros tenham a possibilidade de se familiarizar mais com todas as etapas da metodologia Scrum e assim adquirir mais conhecimento sobre cada área, engajando também no aprimoramento de suas habilidades como líder.

## 2.4 Processo

<i>Atividade</i>	<i>Objetivo</i>	<i>Papel</i>
<i>Estabelecer os prazos de cada sprint</i>	<i>Escolher os prazos de cada sprint para o desenvolvimento de requisitos.</i>	<i>Scrum Master e Product Owner</i>
<i>Realizar escopo do produto</i>	<i>Escolhas de todas as características do produto</i>	<i>Product Owner e Equipe de Desenvolvimento</i>
<i>Realizar prototipagem</i>	<i>Fazer o protótipo das páginas que será seguido para o desenvolvimento do software</i>	<i>Desenvolvedor</i>

<i>Realizar estudo das Tecnologias</i>	<i>Estudar as tecnologias utilizadas para o desenvolvimento de software.</i>	<i>Desenvolvedor</i>
<i>Realizar o desenvolvimento dos requisitos</i>	<i>Desenvolver os requisitos do produto</i>	<i>Desenvolvedor</i>
<i>Realizar versionamento</i>	<i>Criar versões do projeto para o Product Owner e o Scrum Master, poderem ter uma visão mais próxima do desenvolvimento do produto, além de ter uma versão beta para poder testar</i>	<i>Desenvolvedor</i>
<i>Realizar Refatoração</i>	<i>Melhorar o entendimento e qualidade do código do produto de software</i>	<i>Desenvolvedor</i>
<i>Realizar Revisão</i>	<i>Revisar o código do produto de software</i>	<i>Desenvolvedor</i>

## 2.5 Procedimentos

<i>Atividade</i>	<i>Método</i>	<i>Ferramenta</i>
<i>Estabelecer dos prazos de cada sprint</i>	<i>planning poker</i>	<i>Planilha excel</i>
<i>Realizar escopo do produto</i>	<i>Lean inception</i>	<i>Mural</i>
<i>Realizar prototipagem</i>	<i>Criação das estruturas e design de cada página, assim como a relações entre elas</i>	<i>Figma</i>
<i>Realizar estudo das Tecnologias</i>	<i>Estudar e aprimorar as habilidades dos desenvolvedores por meio de cursos e material disponível como livros e documentação</i>	<i>Udemy, YouTube, Documentos</i>

<i>Realizar o desenvolvimento dos requisitos</i>	<i>Desenvolver os requisitos estabelecidos em cada sprint .</i>	<i>Vs Code, Node.js, Reactive e MongoDB</i>
<i>Realizar versionamento</i>	<i>Separar o progresso de desenvolvimento do produto de software</i>	<i>GitHub</i>
<i>Realizar Refatoração</i>	<i>Analisar como o código foi escrito e reescrevê-lo com o objetivo de tornar sua leitura mais fácil, renomeando variáveis, funções etc</i>	<i>GitHub/Vscode</i>

### 3 LIÇÕES APRENDIDAS

#### 3.1 Unidade 1

- Entendimento das diversas abordagens para se desenvolver um software ou produto.
- Swebok - Entendimento das disciplinas de desenvolvimento de software.
- SAFe

*[Liste as lições aprendidas na retrospectiva, com ênfase especial nas ações a serem tomadas para melhorar, por exemplo: o ambiente de desenvolvimento, o processo ou a colaboração da equipe.]*

#### 3.2 Unidade 2

*[Liste as lições aprendidas na retrospectiva, com ênfase especial nas ações a serem tomadas para melhorar, por exemplo: o ambiente de desenvolvimento, o processo ou a colaboração da equipe.]*

#### 3.3 Unidade 3

*[Liste as lições aprendidas na retrospectiva, com ênfase especial nas ações a serem tomadas para melhorar, por exemplo: o ambiente de desenvolvimento, o processo ou a colaboração da equipe.]*

#### 3.4 Unidade 4

*[Liste as lições aprendidas na retrospectiva, com ênfase especial nas ações a serem tomadas para melhorar, por exemplo: o ambiente de desenvolvimento, o processo ou a colaboração da equipe.]*

#### 3.5 Unidade 5

*[Liste as lições aprendidas na retrospectiva, com ênfase especial nas ações a serem tomadas para melhorar, por exemplo: o ambiente de desenvolvimento, o processo ou a colaboração da equipe.]*

### 4 REFERÊNCIAS BIBLIOGRÁFICAS

1. Swebok v3.0, IEE, GUIDE TO THE SOFTWARE ENGINEERING BODY OF KNOWLEDGE



*Sommerville, Ian Engenharia de software/ Ian Sommerville; tradução Luiz Cláudio Queiroz; revisão técnica Fábio Levy Siqueira. -- 10. ed. -- São Paulo: Pearson Education do Brasil, 2018*