MapMarket

**VISÃO DO PRODUTO E PROJETO**

Versão 2.1

**Histórico de Revisão**

| **Data** | **Versão** | **Descrição** | **Autor** |
| --- | --- | --- | --- |
| 03/02 | 1.0 | Visão inicial produto | Pablo Christiano |
| 10/02 | 2.0 | Atualização visão e Requisitos | Pedro Menezes |
| 23/02 | 2.1 | Atualização Requisitos | Samuel Sato |
| 14/03 | 3.0 | Arquitetura de Software e Diagrama de classes | Samuel Sato |

**Sumário**

[***1***](#_heading=h.30j0zll) ***VISÃO GERAL DO PRODUTO 4***

[**1.1**](#_heading=h.1fob9te) **Declaração do Problema 4**

[**1.2**](#_heading=h.tyjcwt) **Declaração de Posição do Produto 4**

[**1.3**](#_heading=h.1t3h5sf) **Objetivos do Produto 4**

[**1.4**](#_heading=h.4d34og8) **Escopo do Produto 4**

[1.4.1](#_heading=h.2s8eyo1) Requisitos Funcionais 4

[1.4.2](#_heading=h.17dp8vu) Requisitos Não-Funcionais 4

[**1.5**](#_heading=h.3rdcrjn) **Mínimo Produto Viável (MVP) 5**

[***2***](#_heading=h.26in1rg) ***ABORDAGEM DE DESENVOLVIMENTO DE SOFTWARE 5***

[**2.1**](#_heading=h.23ckvvd) **Metodologia 5**

[**2.2**](#_heading=h.35nkun2) **Processo 5**

[**2.3**](#_heading=h.44sinio) **Procedimentos 5**

[***3***](#_heading=h.z337ya) ***VISÃO GERAL DO PROJETO 6***

[**3.1**](#_heading=h.3j2qqm3) **Organização do Projeto 6**

[**3.2**](#_heading=h.1y810tw) **Planejamento das Fases e/ou Iterações do Projeto 6**

[**3.3**](#_heading=h.4i7ojhp) **Matriz de Comunicação 6**

[**3.4**](#_heading=h.2xcytpi) **Gerenciamento de Riscos 6**

[**3.5**](#_heading=h.1ci93xb) **Critérios de Replanejamento 7**

[***4***](#_heading=h.3whwml4) ***LIÇÕES APRENDIDAS 7***

[**4.1**](#_heading=h.2bn6wsx) **Unidade 1 7**

[**4.2**](#_heading=h.3as4poj) **Unidade 2 7**

[**4.3**](#_heading=h.1pxezwc) **Unidade 3 7**

[**4.4**](#_heading=h.49x2ik5) **Unidade 4 7**

[**4.5**](#_heading=h.2p2csry) **Unidade 5 7**

[***5***](#_heading=h.147n2zr) ***REFERÊNCIAS BIBLIOGRÁFICAS 7***

**VISÃO DO PRODUTO E PROJETO**

1. **VISÃO GERAL DO PRODUTO**
   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. **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 aplicacção 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. **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. **Escopo do Produto**
      1. **Requisitos Funcionais**

| **Nº** | **Requisito** | **Prioridade** |
| --- | --- | --- |
| RF1 | Deve ser possível realizar gestão de cliente | ALTA |
| RF2 | Deve ser possível realizar gestão de mercados | ALTA |
| RF3 | Deve ser possível realizar validação de acesso dos usuários | ALTA |
| RF4 | Deve ser possível realizar gestão de lista de compras | ALTA |
| RF5 | Deve ser possível realizar gestão de lista de compras recorrentes | MÉDIA |
| RF6 | Deve ser possível realizar agendamento de compras | BAIXA |
| RF7 | Deve ser possível realizar consulta de produtos (adicionar filtros”funcionalidades”) | ALTA |
| RF8 | Deve ser possível adicionar comentários aos produtos (ex: escolher data de validade, melancia gigante) | BAIXA |
| RF9 | Aplicação terá um sistema de pontuação e descontos; | MÉDIA |
| RF10 | Aplicação terá um sistema de chat online para comunicação cliente-mercado | ALTA |

*Obs:*Prioridade ALTA diz respeito a histórias com maior grau de importância, maior valor de mercado, para o projeto e devem ser implementadas primeiro. Prioridade MÉDIA é relacionada a histórias que apresentam bom valor de mercado, mas não fazem parte do MVP. Prioridade BAIXA apresentam menor grau de importância para o projeto.

* + 1. **Requisitos Não-Funcionais**

| **Nº** | **Requisitos não Funcionais (Classificação)** | **Requisitos Não Funcionais** |
| --- | --- | --- |
| RNF1 | Requisitos de portabilidade | Aplicação deve ser web responsiva |
| RNF2 | Requisitos de portabilidade | Aplicação deve ser suportada em: Chrome, Safari e Firefox |
| RNF3 | Requisitos legais | Aplicação deve seguir em conformidade com a LGPD (Consentimento do cidadão, Finalidade e necessidade) |
| RNF4 | Requisitos de facilidade de uso | A interação com o usuário deverá ser feita por meio de interface gráfica |
| RNF5 | Requisitos da organização | O backend será em Node.JS e o fronted será em React |
| RNF6 | Requisitos da organização | Aplicação deve possuir um conjunto de dados, produtos e mercados, pré-cadastrados |

* 1. **Mínimo Produto Viável (MVP)**
     1. *Requisitos Funcionais*
* RF1: Deve ser possível realizar gestão de cliente
* RF2: Deve ser possível realizar gestão de mercados
* RF3: Deve ser possível realizar validação de acesso dos usuários
* RF4: Deve ser possível realizar gestão de lista de compras
* RF7: Deve ser possível realizar consulta de produtos
* RF10: Aplicação terá um sistema de chat online para comunicação cliente-mercado
  + 1. *Requisitos não Funcionais*
* RNF1: Aplicação deve ser web responsiva
* RNF2: Aplicação deve ser suportada em: Chrome, Safari e Firefox
* RNF4: A interação com o usuário deverá ser feita por meio de interface gráfica
* RNF5: O backend será em Node.JS e o fronted será em React
* RNF6: Aplicação deve possuir um conjunto de dados pré-cadastrados

1. **ABORDAGEM DE DESENVOLVIMENTO DE SOFTWARE**

1. Metodologia
   1. **Questões Técnicas:**
   2. Qual é o tamanho do sistema que está sendo desenvolvido? Será um projeto relativamente grande. Teremos cliente, supermercado e estoque.
   3. Que tipo de sistema está sendo desenvolvido? Aplicação Web responsiva.
   4. Qual a vida útil prevista para o sistema? 4 meses
   5. O sistema está sujeito a controle externo? Sim, o supermercado teria controle sobre os produtos cadastrados e estoque.
2. **Questões Humanas:**
   1. Qual é o nível de competência dos projetistas e programadores do time de desenvolvimento? Todos os integrantes possuem um conhecimento intermediário.
   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.
   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.
3. **Questões Organizacionais:**
   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. 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 fundamenta, uma vez que também será parte do time de desenvolvimento.
   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.
   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.

* 1. **Processo**

| *Atividade* | *Objetivo* | *Papel* |
| --- | --- | --- |
| *Estabelecer os prazos de cada sprint* | *Escolher os prazos de cada sprint para o desenvolvimento de requisitos.* | *Scrum Master e Project Owner* |
| *Realizar escopo do produto* | *Escolhas de todas as características do produto* | *Product Owner e Equipe de Desenvolvimento* |
| *Realizar prototipagem* | *Fazer o protótipo das páginas que será seguido para o desenvolvimento do software* | *Desenvolvedor* |
| *Realizar estudo das Tecnologias* | *Estudar as tecnologias utilizadas para o desenvolvimento de software.* | *Desenvolvedor* |
| *Realizar o desenvolvimento dos requisitos* | *Desenvolver os requisitos do produto* | *Desenvolvedor* |
| *Realizar versionamento* | *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* | *Desenvolvedor* |
| *Realizar Refatoração* | *Melhorar o entendimento e qualidade do código do produto de software* | *Desenvolvedor* |
| *Realizar Revisão* | *Revisar o código do produto de software* | *Desenvolvedor* |

* 1. **Procedimentos**

| *Atividade* | *Método* | *Ferramenta* |
| --- | --- | --- |
| *Estabelecer dos prazos de cada sprint* | *planning poker* | *Planilha excel* |
| *Realizar escopo do produto* | *Lean inception* | *Mural* |
| *Realizar prototipagem* | *Criação das estruturas e design de cada página, assim como a relações entre elas* | *Figma* |
| *Realizar estudo das Tecnologias* | *Estudar e aprimorar as habilidades dos desenvolvedores por meio de cursos e material disponível como livros e documentação* | *Udemy, YouTube, Documentos* |
| *Realizar o desenvolvimento dos requisitos* | *Desenvolver os requisitos estabelecidos em cada sprint .* | *Vs Code, Node.js, Reactive e MongoDB* |
| *Realizar versionamento* | *Separar o progresso de desenvolvimento do produto de software* | *GitHub* |
| *Realizar Refatoração* | *Analisar como o código foi escrito e reescrevê-lo com o objetivo de tornar sua leitura mais fácil, renomeando variáveis, funções etc* | *GitHub/Vscode* |

1. **VISÃO GERAL DO PROJETO**
   1. **Organização do Projeto**

| **Papel** | **Atribuições** | **Responsável** | **Participantes** |
| --- | --- | --- | --- |
| Desenvolvedor | Codificar o produto, realizar refatoração, realizar teste dos códigos | Pedro | Pablo, Samuel |
| Dono do Produto | Atualizar o escopo do produto, organizar o escopo das sprints, validar as entregas | Samuel | Pablo, Pedro |
| Analista de Qualidade | Garantir a qualidade do produto, garantir o cumprimento do conceito de pronto, realizar inspeções de código | Pablo | Pedro, Samuel |
| Scrum master | Gerenciar a equipe de desenvolvimento, organizar reuniões entre a equipe de desenvolvimento e o dono do produto, garante que a estrutura do projeto seja seguido | Pedro | Pablo, Samuel |

* 1. **Planejamento das Fases e/ou Iterações do Projeto**

| **Sprint** | **Produto (Entrega)** | **Data Início** | **Data Fim** |
| --- | --- | --- | --- |
| Sprint 1 | Definição do Produto | 30/01/22 | 05/02/22 |
| Sprint 2 | MVP e Planejamento do Projeto | 20/02/22 | 26/02/22 |
| Sprint 3 | Cadastro de cliente, cadastro de mercado, editar informações de cliente e mercado | 27/02/22 | 05/03/22 |
| Sprint 4 | Cadastrar produto, editar informações de produto, buscar produto | 06/03/22 | 12/03/22 |
| Sprint 5 | Criar lista de compras, criar lista de compras recorrentes, | 13/03/22 | 19/03/22 |
| Sprint 6 | Finalização de compras, realizar login e logout | 20/03/22 | 26/03/22 |
| Sprint 7 | implementar chat online | 27/03/22 | 02/04/22 |
| Sprint 8 | Deletar lista de compras, editar lista de compras | 03/04/22 | 09/04/22 |
| Sprint 9 | atribuir pontos por compras ao cliente, Trocar pontos por descontos | 10/04/22 | 16/04/22 |
| Sprint 10 | Agendar compras, Adicionar comentário em um produto da lista de compras, excluir comentário | 17/04/22 | 23/04/22 |

* 1. **Matriz de Comunicação**

| **Descrição** | **Área/ Envolvidos** | **Periodicidade** | **Produtos Gerados** |
| --- | --- | --- | --- |
| *Reunião semanais para alinhamento da equipe e acompanhamento de atividades, acompanhamentos pelo zenhub.* | *Equipe do Projeto* | *Semanal* | *Relatórios da sprint.* |
| *Apresentação do progresso da equipe* | *Equipe e Professor* | *Mensal* | *Ao fim de cada Módulo* |

* 1. **Gerenciamento de Riscos**

| Risco | Ações |
| --- | --- |
| *Horários incompatíveis* | *Organizar agendas semanais em grupo para combater a perda de reuniões por incompatibilidade de horários* |
| *Não cumprir o MVP* | *Revisar entregas semanais dos requisitos a fim de cumprir o MVP* |

* 1. **Critérios de Replanejamento**

O replanejamento do produto ocorrerá caso seja verificado, ao decorrer do desenvolvimento, que o escopo do projeto está muito grande e a equipe de desenvolvimento não será capaz de entregar o necessário para compor o MVP até a data estipulada para a entrega do produto.

* 1. **Padrões de projeto**
* Padrão de criação:

Builder: Para criação de objetos complexos em etapas, não é necessário criar um objeto passando todos os atributos no construtor padrão

Singleton: Para criação de apenas uma instância com as credenciais para acessar o banco de dados

* Padrões estruturais:

Adapter: Para modularizar o código possibilitando maior facilidade em manutenção e melhorias

Proxy: Para controle de acesso dos usuários

* Padrões comportamentais:

Strategy: Para possível aplicação de cupons de desconto e resgate de pontuações, aplicar descontos.

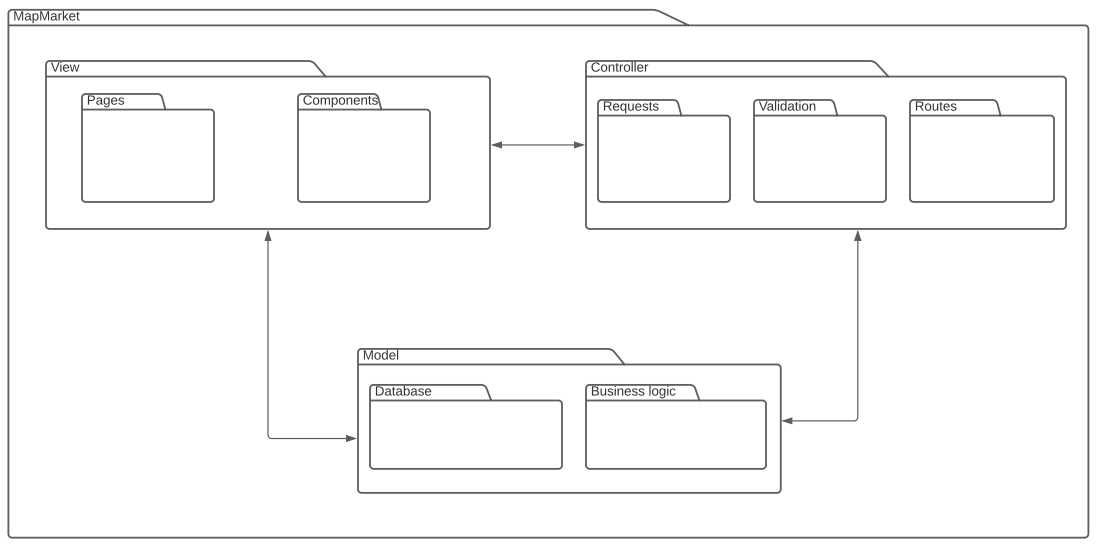
* 1. **Arquitetura de Projeto**

Arquitetura definida: MVC

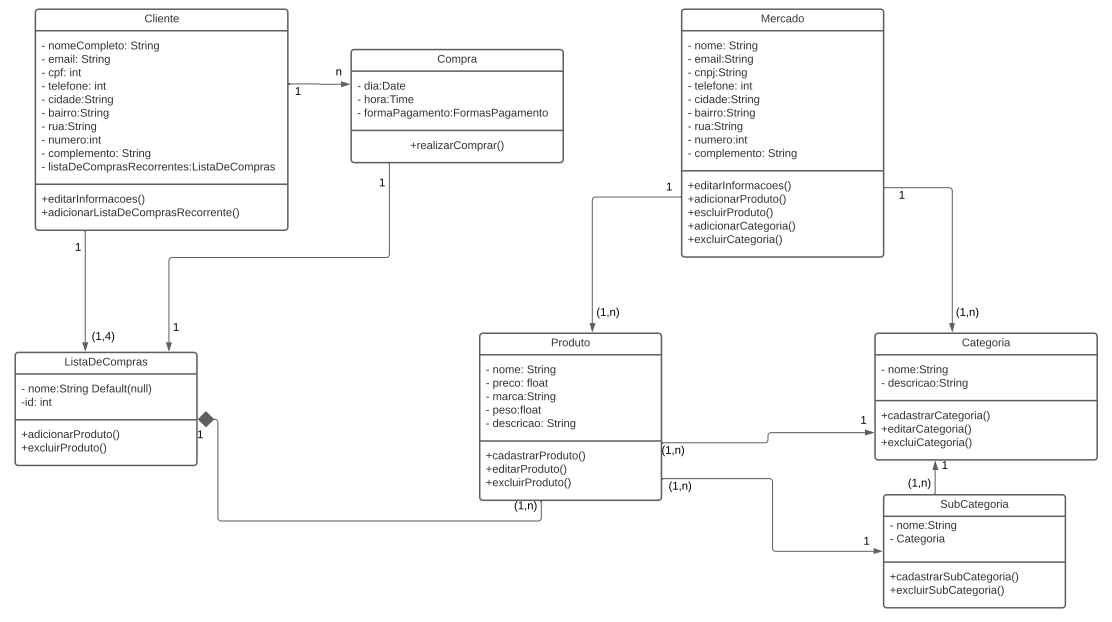
Model: Camada responsável pela lógica e regras de negócio, relacionamento com o banco de dados.

View: Camada de apresentação visual das informações (front-end).

Controller: Camada de intermediação entre Model e View, validar algumas operações.



* 1. **Diagrama de Classes**

****

1. **LIÇÕES APRENDIDAS**
   1. **Unidade 1**

*Unidade 1*

* Entendimento das diversas abordagens para se desenvolver um software ou produto.
* Swebok - Entendimento das disciplinas de desenvolvimento de software.
* SAFe
  1. **Unidade 2**
* Definição de requisitos funcionais e não funcionais, e quais destes compõem meu MVP.
* Estruturação do backlog de acordo com a abordagem ágil SAFe.
* Identificação de épicos, features, histórias e tasks.
* Refinamento de histórias de usuário.
  1. **Unidade 3**
* Estilos de Arquitetura de Software
* Protótipo de Interface
* Entendimento e aplicação sobre padrões de projeto
  1. **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.]*

* 1. **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.]*

1. **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* |
| --- | --- |
| *2.* | Scaled Agile Framework [*https://www.scaledagileframework.com/*](https://www.scaledagileframework.com/)*#* |