

UNIVERSIDADE FEDERAL DO CEARÁ

CAMPUS DE RUSSAS

Sistema de Mercantil

**Adriano Kennedy ¹, Adriano Mendes ², João Pedro ³, Rian Rodrigues ⁴, Ruan Pablo ⁵, Vinícius Antonio ⁶.**

**Orientador: Profa. Dsc. Jacilane de Holanda Rabelo   
(Engenharia de Software)**

**¹Engenharia de Software – RUS0099  
Universidade Federal do Ceará (UFC) – Russas – CE – Brasil**

adrianomendes@alu.ufc.br

rianrodrigz@alu.ufc.br

filhokennedy@alu.ufc.br

ruanestacio@alu.ufc.br

jprodriguesmuniz27@hotmail.com

viniciusantonio@alu.ufc.br

jacilane.rabelo@uninorte.com.br

***Resumo.***

*O projeto em questão é um software de gestão de supermercados com o objetivo de otimizar processos internos e solucionar problemas encontrados na administração e controle de estoque, compras e vendas. Os benefícios esperados são a melhoria na gestão e controle de estoque, a redução de perdas e desperdícios, o aumento da eficiência operacional, a otimização do fluxo de caixa e a possibilidade de tomada de decisões mais assertivas. As funções do software incluem cadastro de usuários, pesquisa de produtos, adição de itens ao carrinho, pagamento, registro de compras, rastreamento de status dos pedidos, gerenciamento de inventário, entre outros.*

# Introdução do Projeto

O sistema a ser desenvolvido é um software que visa aprimorar a gestão de supermercados, solucionando os problemas encontrados na administração e controle de estoque, compras e vendas. O objetivo é otimizar os processos internos, facilitando o trabalho dos funcionários e proporcionando maior eficiência e segurança nas transações.

Os benefícios esperados do uso deste sistema são a melhoria na gestão e controle de estoque, a redução de perdas e desperdícios, o aumento da eficiência operacional, a otimização do fluxo de caixa e a possibilidade de tomada de decisões mais assertivas. Com isso, espera-se que o sistema contribua para o sucesso e crescimento dos supermercados que o utilizarem.

**1.1. Objetivo do Projeto**

* Melhoria na gestão e controle de estoque;
* Redução de perdas e desperdícios;
* Aumento da eficiência operacional;
* Otimização do fluxo de caixa;
* Possibilidade de tomada de decisões mais assertivas;
* Contribuição para o sucesso e crescimento dos supermercados que utilizarem o sistema.

Será desenvolvido um sistema de gestão de supermercados para solucionar problemas relacionados ao controle de estoque e vendas. O sistema será automatizado, para melhorar a gestão e controle de estoque, reduzir perdas e desperdícios, aumentar a eficiência operacional, otimizar o fluxo de caixa e permitir tomadas de decisão assertivas.

**1.2. Justificativa da Escolha do Tema**

Esse é um bom tema para ser abordado, pois os supermercados são estabelecimentos comerciais presentes em praticamente todas as cidades, e são importantes tanto do ponto de vista econômico quanto social. No entanto, muitos supermercados enfrentam problemas na gestão e controle de estoque, vendas e finanças, o que pode comprometer a eficiência operacional e a lucratividade do negócio.

**1.3. Organização do Trabalho**

[1. Introdução do Projeto](#)

[2. Descrição Geral do Sistema](#)

[3. Documentação de Casos de Uso](#)

[4. Diagrama de Classes](#)

[5. Matriz de Rastreabilidade](#)

[6. Planejamento do Scrum com Trello](#)

[7. Modificações](#)

[7.1 Sugestões de mudanças](#_5d6hv1iterbn)

[8. Como será feito](#)

[9. Conclusões](#)

[10. Ata de Reunião](#)

[11. Referências](#)

# Descrição Geral do Sistema

O objetivo principal do sistema é aprimorar a gestão de supermercados, solucionando problemas relacionados à administração e controle de estoque, compras e vendas. O sistema automatizado visa melhorar a eficiência operacional, otimizar o fluxo de caixa, reduzir perdas e desperdícios e permitir tomadas de decisão assertivas. A justificativa para escolher esse tema é que os supermercados são estabelecimentos comerciais importantes para a economia e a sociedade, mas muitos enfrentam problemas na gestão e controle de estoque, vendas e finanças. O sistema de gestão de supermercados desenvolvido pode contribuir para o sucesso e crescimento dos supermercados que o utilizarem.

As principais funções do aplicativo seriam:

**1.** Cadastro de usuários e criação de contas com informações pessoais e de endereço (RF01)

**2.** Pesquisa de produtos por nome, categoria ou palavra-chave (RF02)

**3.** Adição de itens ao carrinho de compras virtual (RF03)

**4.** Pagamento pelos itens no carrinho e escolha da opção de entrega (RF04)

**5.** Registro das compras do usuário para visualização do histórico (RF05)

**6.** Oferta de promoções e descontos especiais para incentivar as compras dos usuários (RF06)

**7.** Rastreamento do status dos pedidos e notificações sobre o andamento da entrega (RF07)

**8.** Catálogo de produtos atualizado com preços, descrições e imagens (RF08)

**9.** Adição e atualização de produtos em inventário pelos vendedores, incluindo informações como descrição, preço, quantidade e imagem (RF09)

**10.** Atualização das informações de perfil dos usuários, incluindo endereço, informações de pagamento e preferências (RF10)

**11.** Adição de itens à lista de desejos para compras futuras (RF11)

**12.** Gerenciamento de inventário e visualização de relatórios de vendas pelos vendedores (RF12)

**13.** Classificação e escrita de comentários sobre os produtos pelos usuários (RF13)

**14.** Criação de listas de compras personalizadas com base em categorias, histórico de compras e itens salvos (RF14)

**2.1. Descrição do Problema**

O problema identificado é que muitos supermercados enfrentam dificuldades na gestão e controle de seus estoques, vendas e finanças. A falta de controle efetivo pode levar a perda de vendas e comprometer a lucratividade do negócio, afetando diretamente o sucesso e o crescimento da empresa.

A solução proposta é o desenvolvimento de um sistema de gestão de supermercados automatizado, que visa aprimorar a eficiência operacional, otimizar o fluxo de caixa, reduzir perdas e desperdícios e permitir tomadas de decisão assertivas. O sistema oferece funções como cadastro de usuários, pesquisa de produtos, adição de itens ao carrinho virtual, pagamento pelos itens, rastreamento do status dos pedidos, catálogo de produtos atualizado, gerenciamento de inventário e visualização de relatórios de vendas pelos vendedores, entre outras.

O sistema afetará diretamente os supermercados que o utilizarem, bem como seus funcionários e clientes. A utilização do sistema permitirá uma melhoria da gestão e controle de estoque, redução de perdas e desperdícios, aumento da eficiência operacional, otimização do fluxo de caixa e possibilidade de tomada de decisões mais assertivas. Isso pode contribuir significativamente para o sucesso e crescimento dos supermercados que o utilizarem, bem como para a economia e a sociedade em geral.

**2.2. Principais Envolvidos e suas Características**

**2.1.1. Usuários do Sistema**

O sistema de gestão de supermercados é destinado a empresas do setor varejista que operam com supermercados. Os principais usuários do sistema são os funcionários da empresa, incluindo vendedores, gerentes de loja, responsáveis pelo controle de estoque, compras e finanças.

Os vendedores serão os principais usuários do sistema durante o processo de venda, utilizando-o para acessar informações sobre os produtos, preços, promoções e descontos. Já os gerentes de loja e os responsáveis pelo controle de estoque poderão utilizar o sistema para gerenciar o inventário, verificar o estoque de produtos, gerar relatórios de vendas e visualizar o histórico de compras dos usuários.

Os usuários do sistema devem ter conhecimentos básicos de informática e treinamento específico para utilizar as funcionalidades do sistema. É importante que o sistema seja intuitivo e fácil de usar para garantir a efetividade na sua utilização pelos funcionários. Além disso, o sistema deve ser capaz de suportar um grande volume de informações e transações simultâneas, garantindo a agilidade e segurança nas operações.

**2.3. Regras de Negócio**

* Restrição de acesso: apenas usuários cadastrados e autenticados poderão acessar as funcionalidades do sistema.
* Restrição de estoque mínimo: o sistema deve emitir alertas e impedir a venda de produtos que estejam abaixo de um determinado nível mínimo de estoque, definido pela empresa.
* Restrição de pagamento: o sistema deve permitir apenas as formas de pagamento previamente acordadas entre a empresa e o cliente, evitando fraudes ou problemas de cobrança.
* Tolerância à falhas: o sistema deve ser capaz de lidar com eventuais falhas de conexão ou interrupções de energia elétrica, garantindo a integridade dos dados e das operações realizadas.
* Volume de informações: o sistema deve ser capaz de armazenar grandes quantidades de informações, como dados de cadastro de usuários, produtos, vendas e históricos de compras, de forma organizada e segura.
* Estimativa de crescimento: o sistema deve ser capaz de lidar com um aumento no volume de informações e transações à medida que a empresa cresce, sem comprometer a performance ou a segurança do sistema.
* Ferramentas de apoio: o sistema pode contar com ferramentas de apoio para auxiliar nas atividades de gestão, como relatórios gerenciais, análises de desempenho e indicadores de venda.
  1. **Protótipo do Sistema**

Fazer um protótipo de alta fidelidade do sistema pode trazer diversas vantagens para o processo de desenvolvimento do projeto, tais como:

1. Melhorar a comunicação: O protótipo de alta fidelidade é uma representação visual e funcional do sistema, que permite uma melhor comunicação entre a equipe de desenvolvimento e o cliente. Isso ajuda a garantir que todos estejam alinhados em relação às expectativas e requisitos do sistema.

2. Identificar problemas precocemente: Ao testar o protótipo de alta fidelidade, é possível identificar problemas e falhas no sistema antes mesmo de iniciar o desenvolvimento completo. Isso permite que a equipe de desenvolvimento faça os ajustes necessários, economizando tempo e recursos.

3. Obter feedback dos usuários: O protótipo de alta fidelidade permite que os usuários testem as funcionalidades do sistema e forneça feedback valioso para a equipe de desenvolvimento. Isso ajuda a garantir que o sistema seja desenvolvido de acordo com as necessidades e expectativas dos usuários.

4. Reduzir o tempo e custo de desenvolvimento: Ao identificar problemas e obter feedback dos usuários precocemente, a equipe de desenvolvimento pode fazer os ajustes necessários antes de iniciar o desenvolvimento completo. Isso reduz o tempo e custo do desenvolvimento, uma vez que os problemas são identificados e corrigidos antes que se tornem mais complexos e caros de resolver.

5. Facilitar a documentação: O protótipo de alta fidelidade é uma representação visual e funcional do sistema, que pode ser usado como uma fonte de documentação para o projeto. Isso ajuda a garantir que todas as funcionalidades e requisitos do sistema estejam documentados de forma clara e precisa.

[**Link Figma**](https://www.figma.com/file/71tYBfLcF5Ukg1EnWKz43c/Trabalho-de-MDS?node-id=1-1236)

* 1. **Aplicação das Técnicas**
* Especificação de requisitos (funcionais e não-funcionais): nesta técnica, foram levantados e documentados todos os requisitos do sistema, tanto funcionais quanto não-funcionais. Os artefatos gerados nesta técnica incluem a **Tabela 1 (Requisitos Funcionais do Sistema)** e a **Tabela 2 (Requisitos Não Funcionais do Sistema)**.
* Modelagem de casos de uso: nesta técnica, foi criado o diagrama de casos de uso da UML, que descreve os principais casos de uso do sistema e as interações entre os usuários e o sistema. Os artefatos gerados nesta técnica incluem o diagrama de casos de uso (**Figura 1**).
* Criação de diagrama de classes: nesta técnica, foi criado o diagrama de classes da UML, que descreve as principais classes do sistema e as relações entre elas. Os artefatos gerados nesta técnica incluem o diagrama de classes (**Figura 2**).
* Protótipo do sistema: nesta técnica, foi criado um protótipo do sistema, que permite validar e testar algumas das funcionalidades do sistema antes de sua implementação completa. Os artefatos gerados nesta técnica incluem o protótipo do sistema (**Link Figma**).
* Matriz de rastreabilidade: nesta técnica, foi criada a matriz de rastreabilidade, que relaciona os requisitos do sistema com as funcionalidades e artefatos correspondentes. Os artefatos gerados nesta técnica incluem a matriz de rastreabilidade (**Figura 3**).
* Planejamento do desenvolvimento do projeto utilizando Scrum+Kanban: nesta técnica, foi definido o processo de desenvolvimento do projeto, utilizando a metodologia ágil Scrum em conjunto com o método Kanban. Os artefatos gerados nesta técnica incluem o backlog do produto (lista de funcionalidades a serem implementadas), o backlog da sprint (lista de tarefas a serem realizadas em cada sprint), o quadro Kanban (para visualização e controle do progresso do projeto) na **Figura 4**.
  1. **Resultados Obtidos**

Inicialmente, foi realizada uma análise dos problemas enfrentados pelos supermercados em relação à gestão de estoque, compras e vendas. Com base nessas informações, foi elaborada uma lista inicial de requisitos funcionais e não funcionais. Para garantir que nenhum requisito importante fosse deixado de fora, foi criada uma matriz de rastreabilidade, que mapeava cada requisito funcional para sua respectiva modelagem e vice-versa.

A análise dos resultados também incluiu a criação de um protótipo do sistema, que foi utilizado para validar e refinar os requisitos. Além disso, a análise dos resultados também considerou fatores como restrições de negócio, restrições de desempenho e volume de informação a ser armazenada. Esses fatores foram levados em consideração na definição dos requisitos não funcionais, que foram incluídos no conjunto inicial de requisitos do sistema.

**Tabela 1. Requisitos Funcionais do Sistema**

| **Código** | **Descrição** | **Prioridade** |
| --- | --- | --- |
| RF01 | O aplicativo deve permitir que os usuários se cadastrem e criem uma conta com informações pessoais e de endereço. | Essencial |
| RF02 | Os usuários devem conseguir pesquisar produtos específicos por nome, categoria ou palavra-chave. | Essencial |
| RF03 | O aplicativo deve permitir que os usuários adicionem e deletem itens do seu carrinho de compras virtual. | Essencial |
| RF04 | O aplicativo deve permitir que os usuários paguem pelos itens em seu carrinho e escolham uma opção de entrega. | Essencial |
| RF05 | O aplicativo deve manter um registro das compras do usuário para que eles possam ver o que compraram anteriormente. | Importante |
| RF06 | O sistema deve oferecer promoções e descontos especiais para incentivar as compras dos usuários. | Desejável |
| RF07 | O sistema deve permitir que os usuários rastreiem o status dos pedidos e recebam notificações sobre o andamento da entrega. | Essencial |
| RF08 | O aplicativo deve oferecer um catálogo de produtos atualizado com preços, descrições e imagens. | Essencial |
| RF09 | O sistema deve permitir que os vendedores adicionem e atualizem produtos em seu inventário, incluindo informações como descrição, preço, quantidade e imagem. | Essencial |
| RF10 | Os usuários devem poder atualizar suas informações de perfil, incluindo endereço, informações de pagamento e preferências. | Importante |
| RF11 | O aplicativo deve permitir que os usuários adicionem itens à sua lista de desejos para compras futuras. | Desejável |
| RF12 | O sistema deve permitir que o vendedor visualize relatórios de vendas. | Desejável |
| RF13 | O sistema deve permitir que os usuários classifiquem e escrevam comentários sobre os produtos que compraram. | Importante |
| RF14 | O sistema deve verificar o endereço antes do pedido ser realizado pelo usuário. | Importante |
| RF15 | O sistema deve mostrar a previsão de chegada de produtos que acabaram em estoque. | Desejável |
| RF16 | O sistema deve possibilitar ao usuário deletar um ou mais itens no carrinho. | Importante |
| RF17 | O aplicativo deve fornecer um horário previsto para a entrega dos produtos antes e após a compra. | Desejável |
| RF18 | O sistema deve informar quando um produto estiver se esgotando. | Desejável |
| RF19 | O sistema deve mostrar o seu pedido indo para seu endereço em tempo real. | Desejável |
| RF20 | O sistema deve possibilitar a repetição de uma compra passada | Importante |

**Tabela 2. Requisitos Não-Funcionais do Sistema**

| **Código** | **Descrição** | **Categoria** | **Prioridade** |
| --- | --- | --- | --- |
| RNF01 | O aplicativo deve ser fácil de usar e intuitivo para os usuários poderem navegar e fazer compras facilmente. | Usabilidade | Desejável |
| RNF02 | O aplicativo deve ser confiável e estar disponível para uso durante as horas de funcionamento do supermercado | Disponibilidade e Confiabilidade | Importante |
| RNF03 | O aplicativo deve garantir a segurança das informações pessoais e de pagamento dos usuários. | Segurança | Essencial |
| RNF04 | O aplicativo deve ser compatível com vários dispositivos móveis e sistemas operacionais. | Compatibilidade | Desejável |
| RNF05 | O aplicativo deve ter bom desempenho para os usuários poderem pesquisar, adicionar ao carrinho e finalizar compras rapidamente. | Desempenho | Importante |
| RNF06 | O aplicativo deve estar disponível para uso 24 horas por dia, 7 dias por semana, com tempo de inatividade mínimo. | Disponibilidade | Desejável |
| RNF07 | O aplicativo deve ser capaz de lidar com um grande número de usuários e transações simultâneas. | Desempenho | Desejável |
| RNF08 | O aplicativo deve ser confiável e não apresentar falhas ou erros frequentes. | Confiabilidade | Desejável |
| RNF09 | O sistema deve ser fácil de atualizar e manter ao longo do tempo. | Manutenção | Desejável |
| RNF10 | O sistema é voltado para dispositivos mobiles. | Portabilidade | Essencial |
| RNF11 | O aplicativo deve ser compatível com diferentes idiomas. | Internacionalização | Desejável |
| RNF12 | O aplicativo deve ter um tempo de resposta rápido para reduzir a espera do usuário durante a navegação e finalização de compras. | Desempenho | Desejável |
| RNF14 | O aplicativo deve fornecer um sistema de suporte eficaz, com opções de contato para suporte ao cliente e um sistema de FAQ abrangente. | Usabilidade | Desejável |
| RNF15 | O aplicativo deve ser projetado para ser acessível para usuários com necessidades especiais, como usuários com deficiência visual ou auditiva. | Usabilidade | Desejável |
| RNF16 | O sistema deve ser escalável e capaz de lidar com aumentos inesperados no tráfego do aplicativo. | Desempenho | Importante |

# Documentação de Casos de Uso

* 1. **Diagrama de Caso de Uso**

Este diagrama de caso de uso visa representar a interação entre um sistema e seus atores, ou seja, as pessoas, organizações ou outros sistemas que interagem com o sistema em questão. Neste diagrama estão dispostas as representações das interações que estarão presentes entre sistema e atores.



**Figura 1. Diagrama de Caso de Uso**

* 1. **Atualização do diagrama de caso de uso**

Foi realizado a apresentação do documento e do protótipo do sistema. Os clientes tiveram algumas sugestões onde influenciaram no diagrama de caso de uso e com isso foi necessário fazer atualizações.

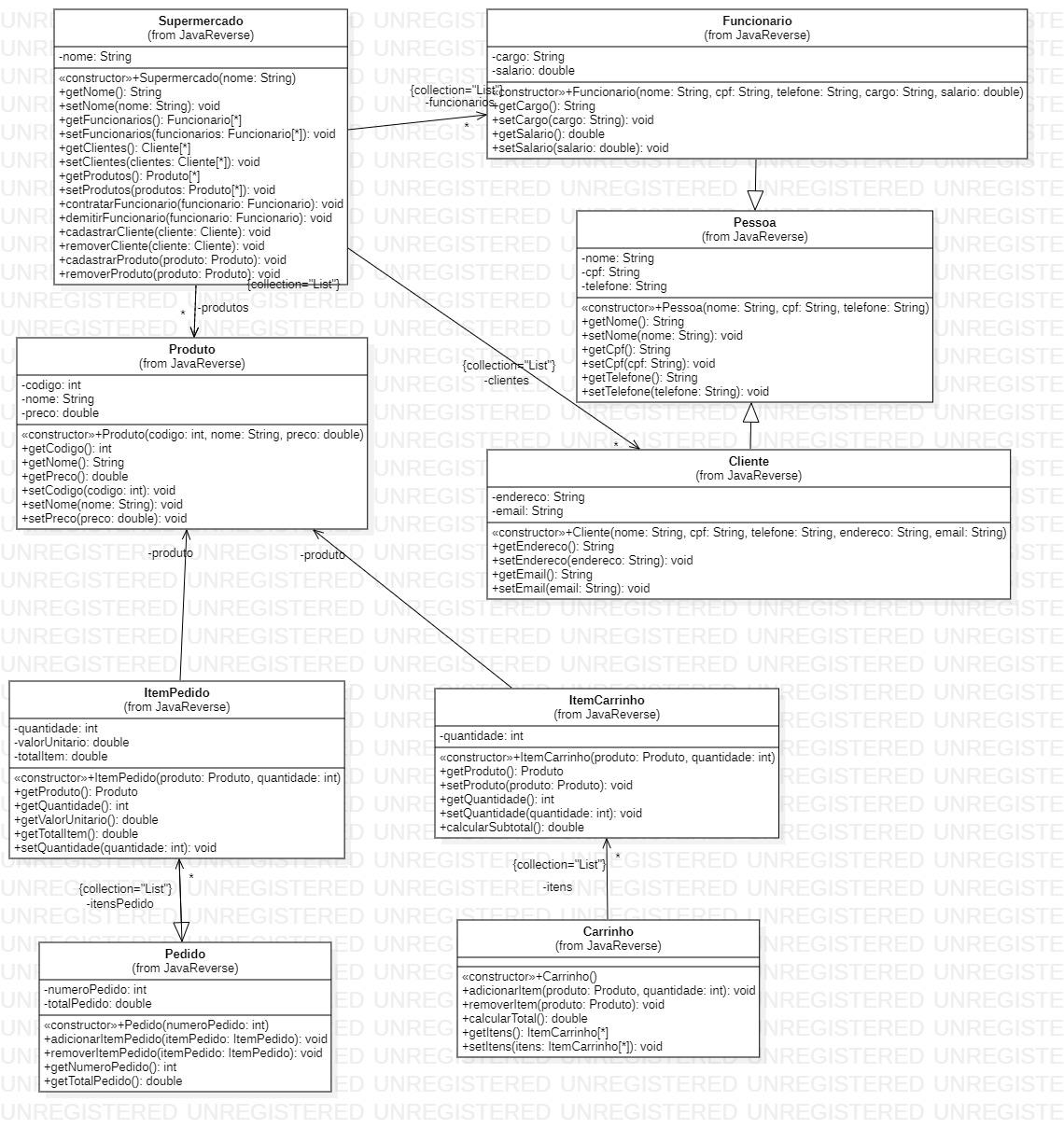


**Figura 2. Diagrama de Caso de Uso atualizado**

# Diagrama de Classes

O diagrama de classes é uma representação gráfica das classes e suas relações em um sistema. É uma ferramenta de modelagem usada para projetar, visualizar, documentar e comunicar a estrutura e o comportamento de um sistema.

Este é o diagrama de classes considerando o sistema desenvolvido, onde são apresentadas as classes que fazem parte do sistema, juntamente com suas relações.

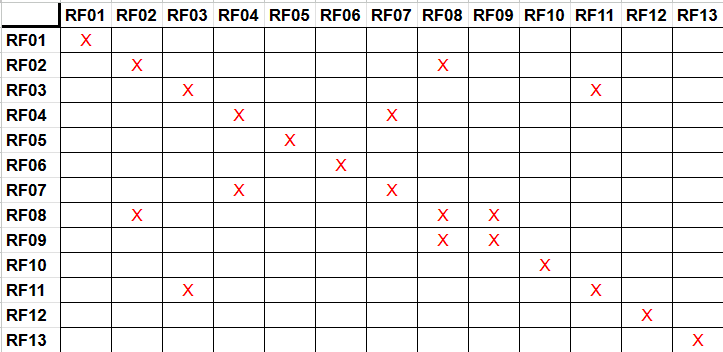
****

**Figura 2. Diagrama de Classes**

# Matriz de Rastreabilidade

A Matriz de rastreabilidade (RFxRF) é uma ferramenta usada no gerenciamento de requisitos de software que ajuda a garantir que todos os requisitos de um sistema estejam documentados, implementados e testados. Ela permite identificar a relação entre os requisitos funcionais (RF).

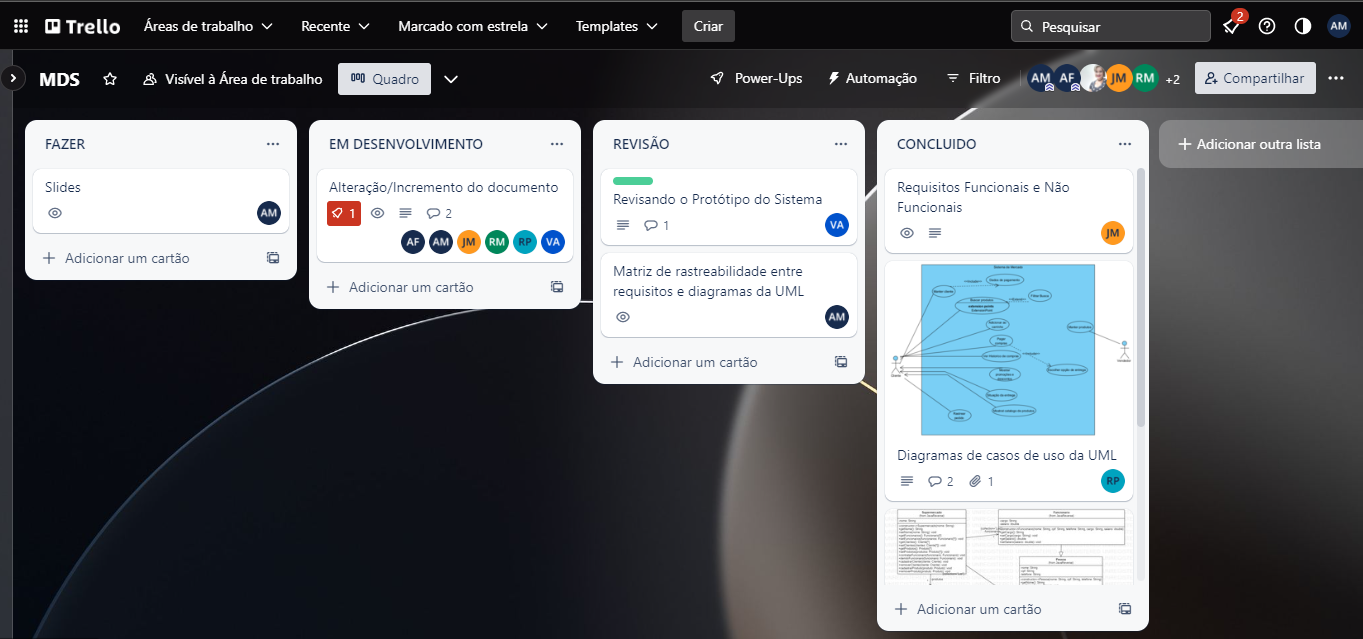
A matriz é preenchida com os requisitos identificados no projeto, com as relações entre eles indicadas nas interseções entre as linhas e colunas correspondentes. No exemplo apresentado, cada coluna e linha representa um requisito identificado no projeto (RF01 a RF20). Os "X" na diagonal principal indicam que cada requisito está relacionado consigo mesmo. Por exemplo, o RF02 está relacionado com o RF08, indicando que esses requisitos estão interdependentes. A matriz pode ser usada para verificar se todos os requisitos foram implementados, testados e validados, além de permitir a identificação de requisitos que precisam ser modificados ou atualizados ao longo do projeto.



**Figura 3. Matriz de Rastreabilidade**

# Planejamento do Scrum com Trello

Existem diversas formas de se organizar um trabalho em equipe e uma das opções disponíveis é utilizar ferramentas de gerenciamento de projetos. No caso descrito, a equipe optou pelo uso do Trello, uma plataforma que permite criar cards para cada tarefa, atribuir responsáveis e acompanhar o andamento de cada etapa. O processo foi dividido em etapas e, após a definição das tarefas de cada membro, os cards foram criados e os comentários utilizados para deixar informações importantes e relevantes. O momento de revisão permitiu que a equipe corrigisse possíveis defeitos nas tarefas atribuídas a cada membro.



**Figura 4. Trello**

# Modificações

# Sugestões de mudanças

|  | **Sugestão 1** | **Sugestão 2** |
| --- | --- | --- |
| **equipe 1** | Categoria na busca. | Verificação do endereço antes do pedido. |
| **equipe 2** | Separar por categoria. | Clientes antigos, fidelidade. |
| **equipe 3** | Repetir compra. | Programa de fidelidade. |
| **equipe 4** | Tempo de alteração do pedido. | Notificação previa para falta de produtos. |
| **equipe 5** | Entrega em tempo real. |  |
| **equipe 6** | Tempo de entrega. | Estimativa antes do pedido. |
| **equipe 7** | Deletar um ou mais itens. | Troca de produto em caso de entrega errado. |
| **equipe 8** | Em caso de acabar o produto, mostrar a previsão de chegada do produto. |  |

* 1. **Analisar as manutenções solicitadas**

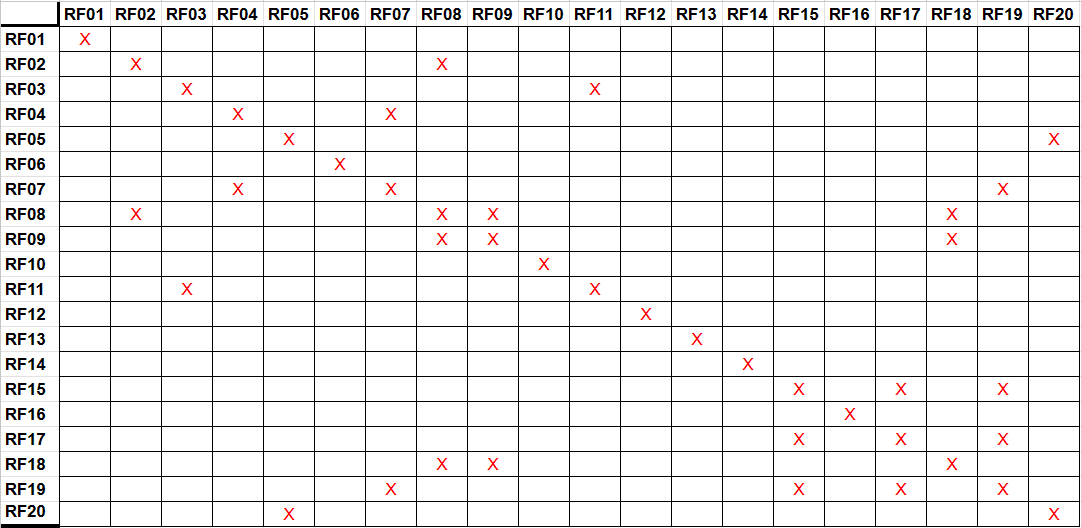
| **Descrição** | **Revisão inicial** | **Ação** | **Responsável** | **Comentários** |
| --- | --- | --- | --- | --- |
| Categoria na busca | 10/05/23 | Rejeitada | equipe 1 | Já existe no sistema |
| Verificação do endereço antes do pedido. | 10/05/23 | Aprovada | equipe 1 | Uma medida ótima para a otimização dos pedidos, evitando problemas maiores. |
| Separar por categoria. | 10/05/23 | Aprovada | equipe 2 | Melhora a organização do site, o que visualmente ajuda bastante o cliente. |
| Clientes antigos, fidelidade. | 10/05/23 | Aprovada | equipe 2 | Um bom método de manter os clientes. |
| Repetir compra. | 10/05/23 | Aprovada | equipe 3 | Funcionalidade que pode ser implementada, visto que pode facilmente agilizar bastante o fluxo do app. |
| Programa de fidelidade. | 10/05/23 | Rejeitada | equipe 3 | Requisito repetido |
| Tempo de alteração do pedido. | 10/05/23 | Rejeitada | equipe 4 | Vários métodos de segurança estão sendo aplicados, como a própria verificação da compra, para evitar a alteração, depois de entrar no sistema |
| Notificação previa para falta de produtos*.* | 10/05/23 | Aprovada | equipe 4 | Um método muito útil no trabalho visto a grande necessidade de controle de estoque |
| Entrega em tempo real. | 10/05/23 | Aprovada | equipe 5 | Um pouco fora do que pode ser realmente implementado no momento |
| Tempo de entrega. | 10/05/23 | Aprovada | equipe 6 | Um requisito totalmente necessário, visto o que o nosso app almeja entregar |
| Estimativa antes do pedido. | 10/05/23 | Aprovada | equipe 6 | Juntamente com o tempo de entrega, dá um possível tempo para a entrega é algo que ajuda bastante a organização do cliente |
| Deletar um ou mais itens. | 10/05/23 | Aprovada | equipe 7 | Outro requisito muito necessário, que se torna até algo básico que todo sistema deve entregar |
| Troca de produto em caso de entrega errado. | 10/05/23 | Rejeitado | equipe 7 | Esta sugestão não cabe ao sistema |
| Em caso de acabar o produto, mostrar a previsão de chegada do produto. | 10/05/23 | Aprovada | equipe 8 | Complementa o requisito de controle de estoque entra na mesma resolução |

* 1. **Impacto das modificações**

Diante das recentes solicitações de modificação da aplicação, tornou-se imprescindível a inclusão de novos requisitos para abranger todas as alterações efetuadas. Essas modificações têm um impacto significativo em vários aspectos, tanto na funcionalidade quanto na usabilidade da aplicação.

O impacto das modificações reflete-se em diversos aspectos. Primeiramente, pode-se observar que os novos requisitos estão relacionados com os antigos, impactando diretamente na funcionalidade.

Em resumo, as modificações solicitadas na aplicação exigiram a inclusão de novos requisitos para abranger todas as mudanças. Essas modificações têm um impacto significativo na funcionalidade, usabilidade, segurança e nos recursos necessários para o desenvolvimento da aplicação.



**Figura 5. Matriz de Rastreabilidade Nova**

* 1. **Modificações priorizadas**

1. Verificação do endereço antes do pedido: Essa funcionalidade é importante para garantir que o endereço de entrega esteja correto antes de confirmar o pedido.

2. Tempo de entrega, estimativa antes do pedido: Informar aos clientes uma estimativa do tempo de entrega antes de fazer o pedido é crucial para gerenciar as expectativas e fornecer uma boa experiência de compra.

3. Categoria na busca: Permitir que os clientes filtrem a busca por categorias é útil para facilitar a descoberta de produtos específicos.

4. Separar por categoria: Essa funcionalidade está relacionada à anterior, permitindo que os produtos sejam organizados em categorias distintas em todo o sistema.

5. Programa de fidelidade: Implementar um programa de fidelidade para incentivar a repetição de compras pode ser benéfico para manter os clientes engajados e satisfeitos.

8. Repetir compra: Facilitar a repetição de compras para clientes é conveniente.

9. Entrega em tempo real: Permitir que os clientes rastreie a entrega em tempo real é uma funcionalidade desejável, pois garante o feedback para o cliente sobre o onde estão suas compras.

10. Tempo de alteração do pedido: Permitir que os clientes alterem o pedido antes da entrega.

11. Notificação previa para falta de produtos: Notificar os clientes quando houver apenas um item disponível pode ser útil para criar um senso de urgência e evitar que eles percam a oportunidade de compra.

12. Deletar um ou mais itens: Permitir que os clientes excluam um ou mais itens do carrinho de compras é uma funcionalidade útil.

13. Horário previsto para entrega: Informar aos clientes um horário previsto para a entrega pode melhorar a experiência do cliente.

14. Em caso de acabar o produto, mostrar previsão de chegada: Informar aos clientes uma previsão de quando um produto esgotado estará disponível novamente pode ajudar a gerenciar as expectativas e evitar frustrações.

# Como será feito

* 1. **Planejamento da Manutenção — ciclo de manutenção**

Aqui está o passo a passo de como realizamos o ciclo de manutenção:

**Planejamento:** Começamos distinguindo as manutenções que seriam realizadas e como elas se comportariam no sistema, para evitar problemas entre os requisitos já feitos com os requisitos que foram atribuídos.

**Inspeção inicial:** Realizamos uma inspeção nos requisitos enviados pelos nossos colegas e vimos quais eram possíveis de serem aplicados.

**Programação da manutenção:** Programamos datas para que as manutenções fossem realizadas conforme foi programado pela equipe.

**Execução:** Realizamos a manutenção de acordo com o cronograma estabelecido. Isso pode envolver reparos ou qualquer outra funcionalidade necessária para garantir o bom funcionamento do sistema.

**Registro e documentação:** Registramos todas as atividades realizadas durante o ciclo de manutenção. Isso inclui a data, o tipo de manutenção realizada, os problemas encontrados e as soluções aplicadas.

* 1. **Descrever o tipo de manutenção prevista — tipo de manutenção**

| **Descrição** | **Tipo de manutenção** | **Comentários** |
| --- | --- | --- |
| Verificação do endereço antes do pedido. | Corretiva | A verificação estava sendo feita em momento indevido |
| Separar por categoria. | Perfectiva | Não havia divisão de categoria |
| Repetir compra. | Perfectiva | Uma nova funcionalidade foi implementada |
| Notificação previa para falta de produtos. | Perfectiva | Não estava prevista nos requisitos |
| Entrega em tempo real. | Corretiva | O cliente não possuía acompanhamento em tempo real |
| Tempo de entrega. | Perfectiva | Não estava sendo calculado o tempo de entrega |
| Estimativa antes do pedido. | Perfectiva | Uma nova funcionalidade foi implementada |
| Deletar um ou mais itens. | Corretiva | O cliente conseguia apenas colocar produtos |
| Em caso de acabar o produto, mostrar a previsão de chegada do produto. | Perfectiva | Não estava prevista nos requisitos |

# Conclusões

Tivemos a oportunidade de relembrar velhos conteúdos, logo, por termos visto a bastante tempo, tivemos que estudar para colocar em prática os métodos de cada metodologia.

Como profissionais entendemos que todas essas técnicas são essenciais no desenvolvimento de software, logo, foi de extrema importância rever todos esses conteúdos.

Ao realizar o trabalho com essas técnicas, o grupo aprendeu diversas lições valiosas que podem ser aplicadas na vida profissional de um engenheiro de computação. Algumas dessas lições incluem:

1. A importância de uma boa comunicação: durante o trabalho, o grupo percebeu que uma comunicação clara e eficaz é fundamental para garantir o sucesso do projeto. Essa lição pode ser aplicada na vida profissional ao se trabalhar em equipes multidisciplinares, onde a comunicação é essencial para garantir que todos estejam alinhados e trabalhando juntos em direção ao mesmo objetivo.

2. A necessidade de uma boa documentação: O grupo percebeu que uma boa documentação é fundamental para garantir que todas as funcionalidades e requisitos do sistema estejam documentados de forma clara e precisa. Essa lição pode ser aplicada na vida profissional ao se trabalhar em projetos de desenvolvimento de software, onde uma documentação adequada pode ajudar a evitar problemas e garantir que o projeto seja entregue no prazo e orçamento.

* 1. **Percepção do Adriano Mendes:**

Foi até mais trabalhoso do que pensei, mas tive a oportunidade de estudar novamente os conceitos que aprendi no passado e colocar em prática experiências que adquiri em outros projetos. Ajuda a compreender que um sistema vai além da codificação.

* 1. **Percepção do Adriano Kennedy:**

Foi muito bom rever conceitos e ferramentas com que eu já havia trabalhado antes, tanto em disciplinas quanto em projetos, ajuda no aprofundamento e concretização do conhecimento estabelecido, por mais que seja uma documentação bem extensa, vale o aprendizado.

* 1. **Percepção do João:**

Como estou fazendo Manutenção de Software junto com Requisitos de Software, consegui por meus aprendizados de requisitos em prática. Além disso, consegui me aprofundar mais em algumas concepções já vistas antes em outras matérias. De longe foi a maior criação de um projeto que eu participei, então é legal ver como funciona na prática.

* 1. **Percepção do Rian:**

Tive a experiência de estudar novamente sobre diagramas de classes e POO, coisas que havíamos visto em semestres passados, além de outros diagramas da UML, o que foi uma ótima forma de lembrar conteúdos passados.

* 1. **Percepção do Ruan:**

Mesmo com todo esse tempo na formação em engenharia de software, essa foi a primeira vez que tive contato com a ferramenta Trello, de fato ela auxilia na produtividade e principalmente na organização, tornando tudo visível para os demais integrantes da equipe. Além disso, foi interessante que mesmo com a divisão de tarefas as pessoas se ajudavam tentando tirar as dúvidas umas das outras.

* 1. **Percepção do Vinícius:**

Trabalhei com o desenvolvimento do protótipo no figma, apesar de saber usar a ferramenta, tive um bom exercício de criatividade na hora de pensar em como seria o modelo para um supermercado, desde a paleta de cores, até as fontes, logo, e todas as partes envolvendo o mesmo. No final coloquei um produto que gosto muito, em especial o café, apesar de não ser um protótipo grande, ficou bem elaborado.

# Ata de Reunião

| N. da Reunião: 1ª. Reunião | |
| --- | --- |
| Data e horário | 10/04/2023. Tarde (15h) |
| Tipo | Presencialmente |
| Participantes: | Kennedy, Adriano, Pedro, Rian, Ruan, Vinícius. |
| Atividade: | A equipe se reuniu pela primeira vez para definir que aplicativo seria feito. |

| N. da Reunião: 2ª. Reunião | |
| --- | --- |
| Data e horário | 23/04/2023. Manhã (11h) |
| Tipo | chamada no Whatsapp |
| Participantes: | Kennedy, Adriano, Pedro, Rian, Ruan, Vinícius. |
| Atividade: | A equipe reuniu-se para decidir as divisões de tarefas entre os integrante do grupo, visando agilizar e não sobrecarregar algum integrante do grupo. |

| N. da Reunião: 3ª. Reunião | |
| --- | --- |
| Data e horário | 25/04/2023. Noite(18h) |
| Tipo | Google Meet |
| Participantes: | Kennedy, Adriano, Pedro, Rian, Ruan, Vinícius. |
| Atividade: | Debater sobre como era o uso do Trello, que por meio da criação de listas de tarefas e atribuição de responsabilidades aos membros da equipe. Com isso, ajuda a garantir que o processo de desenvolvimento ocorra de maneira organizada e eficiente. |

| N. da Reunião: 4ª. Reunião | |
| --- | --- |
| Data e horário | 26/04/2023. Manhã (10h) |
| Tipo | Whatsapp |
| Participantes: | Kennedy, Adriano, Pedro, Rian, Ruan, Vinícius. |
| Atividade: | Utilizamos o horário que seria da aula para prosseguir no processo e continuar a trabalhar no projeto para que avançássemos no processo e atingir nosso objetivo de concluir o trabalho no prazo estabelecido. |

| N. da Reunião: 5ª. Reunião | |
| --- | --- |
| Data e horário | 02/05/2023. Manhã (10h) |
| Tipo | Google Meet |
| Participantes: | Kennedy, Adriano, Pedro, Rian, Ruan, Vinícius. |
| Atividade: | Nos juntamos para realizar a revisão do documento e discutir os preparativos finais da apresentação |

| N. da Reunião: 6ª. Reunião | |
| --- | --- |
| Data e horário | 20/05/2023. Manhã (10h) |
| Tipo | Google Meet |
| Participantes: | Kennedy, Adriano, Pedro, Rian, Ruan, Vinícius. |
| Atividade: | Nos juntamos para realizar a revisão das alterações solicitadas e quais seriam escolhidas, foram separada as atividades |

| N. da Reunião: 7ª. Reunião | |
| --- | --- |
| Data e horário | 27/05/2023. Manhã (10h) |
| Tipo | Google Meet |
| Participantes: | Kennedy, Adriano, Pedro, Rian, Ruan, Vinícius. |
| Atividade: | Nos juntamos para realizar avaliar o documento atualizado com as novas alterações, além do protótipo |

| N. da Reunião: 8ª. Reunião | |
| --- | --- |
| Data e horário | 01/06/2023. Manhã (10h) |
| Tipo | Google Meet |
| Participantes: | Kennedy, Adriano, Pedro, Rian, Ruan, Vinícius. |
| Atividade: | Nos juntamos para realizar a revisão do documento e discutir sobre a apresentação |

# Referências

* Sommerville, I. (2015). Engenharia de software (9a ed.). Pearson.
* Pressman, R. S., & Maxim, B. R. (2016). Engenharia de software: uma abordagem profissional (8a ed.). McGraw Hill.
* Larman, C. (2007). Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento interativo (3a ed.). Bookman.
* Santos, R. C. (2016). Prototipagem rápida de interfaces de usuário para sistemas interativos. Dissertação de mestrado, Universidade Federal do Rio Grande do Sul.
* Borges, M. R., & Amaral, R. L. (2019). Uma revisão sistemática sobre a aplicação de matriz de rastreabilidade em desenvolvimento de software. Revista de Sistemas e Computação, 9(2), 63-72.
* Schwaber, K., & Sutherland, J. (2017). The Scrum guide: The definitive guide to Scrum: The rules of the game. Scrum.org.