

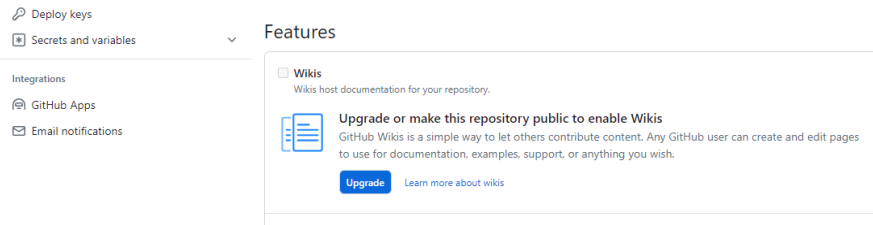
Dicas do professor para edição e formatação no Wiki do GitHub, com MarkDown e HTML

Profa. Monica Tissiani de Toni Pereira

Prof. Radamés Pereira

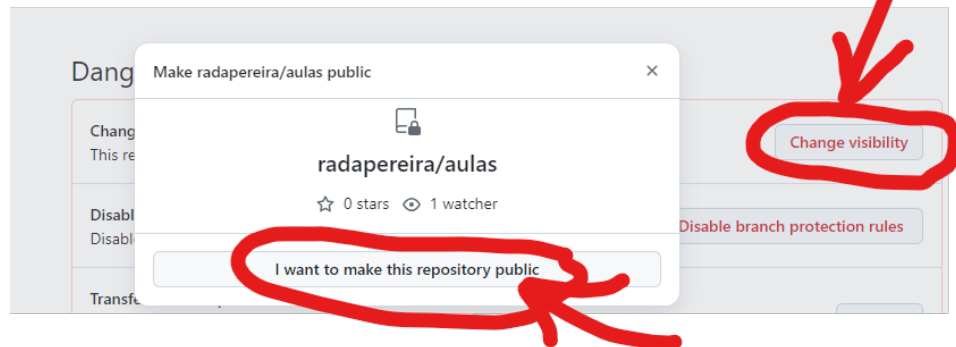
Habilitar uma wiki no GitHub para um projeto é um processo relativamente simples. As wikis no GitHub são úteis para criar documentação detalhada para seu projeto, facilitando a colaboração e o compartilhamento de informações entre os membros da equipe. Abaixo estão os passos detalhados para habilitar e configurar uma wiki no GitHub:

1. **Acesse o Repositório do Projeto:**
 - Vá para a página do repositório do projeto no GitHub.
2. **Navegue para a Aba de Configurações (Settings):**
 - No menu principal do repositório, clique na aba "Settings".
3. **Habilite a Wiki:**
 - Dentro das configurações, role para baixo até encontrar a seção "Features" ou "Options".
 - Marque a opção "Wikis" para habilitar a wiki para o repositório.



Danger Zone





4. Acesse a Wiki:

- Volte para a página principal do repositório.
- Agora, você verá uma nova aba chamada "Wiki" no menu principal do repositório. Clique nela para acessar a wiki.

5. Crie e Edite Páginas na Wiki:

- Na aba "Wiki", você pode começar a criar novas páginas clicando em "Create the first page" se a wiki estiver vazia ou "New Page" para adicionar novas páginas.
- Utilize a sintaxe Markdown para formatar o conteúdo da wiki. O GitHub oferece uma pré-visualização em tempo real enquanto você edita.

6. Gerencie a Wiki:

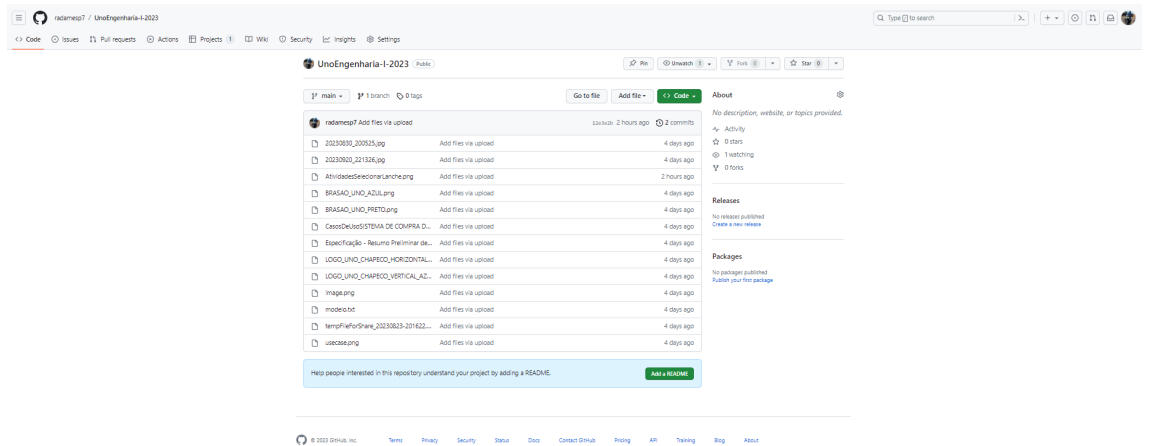
- As páginas da wiki podem ser organizadas em diferentes hierarquias, facilitando a navegação e o acesso à informação.
- Você também pode controlar as permissões da wiki nas configurações do repositório, definindo quem pode ler e editar as páginas.

Observação: No GitHub, o Wiki não suporta a aplicação direta de estilos CSS, como seria o caso em HTML puro.

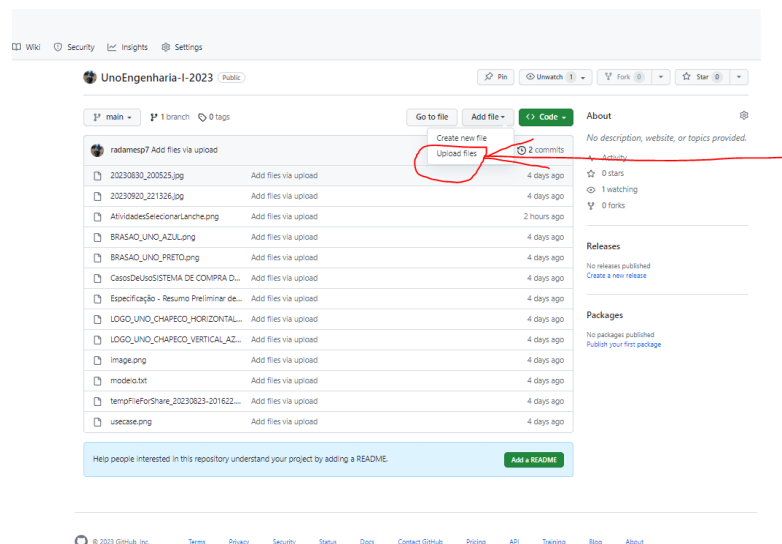
DICAS:

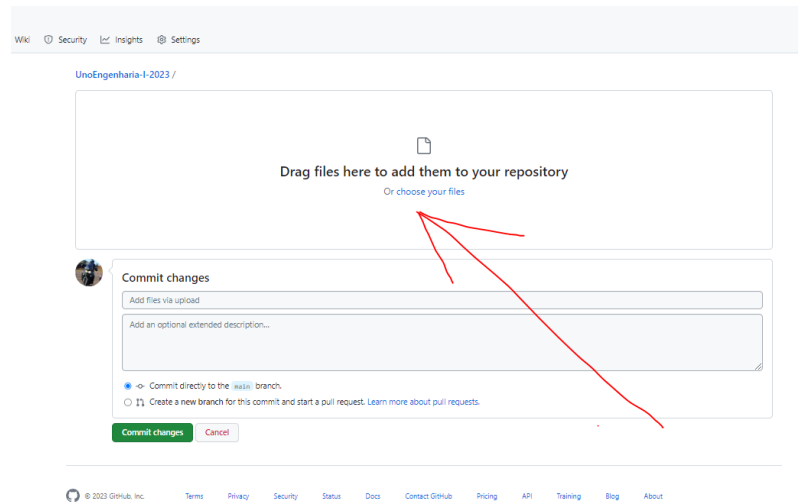
1. Para inserir figuras no Wiki/GitHub

a) Acesse o repositório do GitHub:

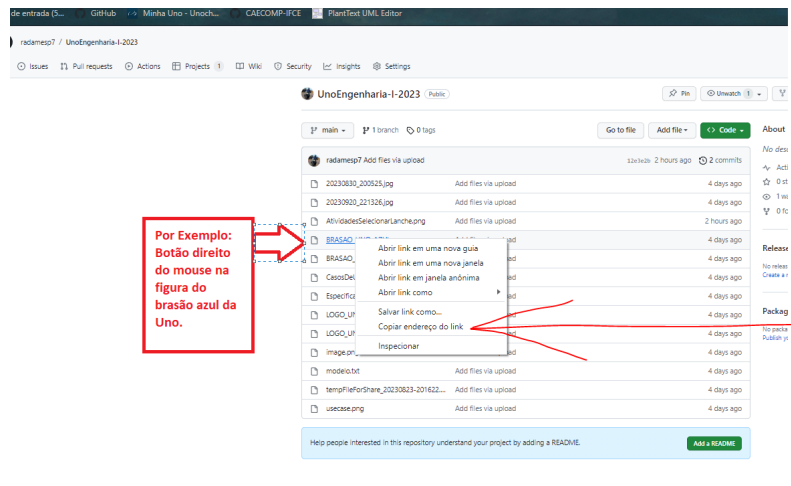


b) Carregue a figura (caso ela não esteja no repositório):





c) Copie o endereço de referência da figura:



d) Cópia do endereço de referência da figura:

https://github.com/radamesp7/UnoEngenharia-I-2023/blob/main/BRASAO_UNO_AZUL.png

e) Cola o endereço no Wiki/GitHub, como usei nesta tabela HTML:



```

**_EM CONSTRUÇÃO_**
<tbody style="width: 25%; border: 2px solid black;">
  <tr>
    <td>
      <img
        src=https://github.com/radamesp7/UnoEngenharia-I-2023/blob/main/BRASAO_UNO_AZUL.png align=right
        width="70"
        height="85" />
    </td>
  </tr>
</tbody>

```

```
# UNIVERSIDADE COMUNITÁRIA REGIONAL DE CHAPECÓ - UNOCHAPECÓ
# ESCOLA POLITÉCNICA
```

Resultado:

UNIVERSIDADE COMUNITÁRIA REGIONAL DE CHAPECÓ - UNOCHAPECÓ 
ESCOLA POLITÉCNICA 



2. Para destacar informações em vermelho, você pode usar a marcação para erro ou alerta em blocos de código, usando o seguinte formato:

```
```diff
```

```
- Este texto aparecerá em vermelho.
```

```
```
```

Ao colocar isso no Wiki do GitHub, o texto "Este texto aparecerá em vermelho." será renderizado com uma cor de fundo vermelha. Geralmente se usa para mostrar diferenças em código (por exemplo, linhas removidas), mas pode ser um truque para conseguir texto vermelho se você realmente precisar destacar algo no seu Wiki. Se você estiver procurando por opções de formatação mais avançadas e personalizáveis, talvez considere utilizar outro sistema ou plataforma que permita uma maior personalização da formatação e estilo.

3. Para usar tabelas em código html:

```
<table border="1">
  <tr>
    <td>Coluna 1, Linha 1</td>
    <td>Coluna 2, Linha 1</td>
  </tr>
  <tr>
    <td>Coluna 1, Linha 2</td>
    <td>Coluna 2, Linha 2</td>
  </tr>
  <!-- Aqui é onde usamos o atributo colspan -->
  <tr>
    <td colspan="2">Esta célula abrange duas colunas</td>
  </tr>
```

```
<tr>
  <td>Coluna 1, Linha 4</td>
  <td>Coluna 2, Linha 4</td>
</tr>
</table>
```

Esta é uma tabela HTML que possui várias linhas com duas colunas: No exemplo, a terceira linha tem uma única célula que abrange duas colunas graças ao uso do atributo `colspan="2"`. Isso resultará em uma tabela onde a terceira linha possui uma única célula que se estende por ambas as colunas, enquanto as outras linhas possuem duas células como esperado.

Descrição das TAGs HTML deste código:

a) ``<table border="1">``:

- A tag ``<table>`` define uma tabela HTML.
- O atributo `border` com valor `"1"` especifica que a tabela terá bordas, e o valor representa a largura da borda em pixels.

b) ``<tr>``:

- Esta tag define uma linha da tabela (table row). Cada linha da tabela é representada por uma tag ``<tr>``.
- No código fornecido, há quatro tags ``<tr>``, o que significa que a tabela tem quatro linhas.

c) ``<td>``:

- Esta tag define uma célula de dados da tabela (table data).
- As células são os componentes individuais dentro de uma linha.
- No código fornecido, cada linha ``<tr>`` tem duas células ``<td>``, representando as duas colunas da tabela.

d) ``colspan="2"``:

- O atributo `colspan` é usado para indicar quantas colunas uma célula deve ocupar.
- Neste caso, o valor `"2"` indica que a célula vai se estender por duas colunas.
- A linha que contém ``<td colspan="2">`` resultará em uma célula que ocupa toda a largura da tabela, abrangendo as duas colunas existentes.

e) ``<!-- ... -->``:

- Estas são tags de comentário. Texto entre estas tags não é renderizado no navegador, mas é útil para adicionar notas ou explicações no código para outros desenvolvedores (ou para você mesmo no futuro). No código fornecido, há um comentário explicando a utilização do atributo `colspan`.

Resumindo, o código HTML define uma tabela com quatro linhas e duas colunas. A terceira linha tem uma célula que se estende por duas colunas graças ao uso do atributo `colspan`.

4. Código do WIKI/GitHub do Professor em 14/10/2023:

```
                **_EM CONSTRUÇÃO_**
<tbody style="width: 25%; border: 2px solid black;">
  <tr>
    <td>
      <img
        src=https://github.com/radamesp7/UnoEngenharia-l-
2023/blob/main/BRASAO_UNO_AZUL.png align=right
        width="70"
        height="85" />
    </td>
  </tr>
  <tr>
  </tbody>

# UNIVERSIDADE COMUNITÁRIA REGIONAL DE CHAPECÓ - UNOCHAPECÓ
# ESCOLA POLITÉCNICA

# Título: _**"SISTEMA DE COMPRA DE MATÉRIA PRIMA - SCMP - Sistema de
Compra de Lanches - SCL"***_

## Equipe (engenheiro e desenvolvedores):

  **Prof. Radamés Pereira**

### Chapecó, 09 de outubro de 2023.

***

# 1. Introdução
## 1.1 Resumo do Projeto (Descrição textual)
```diff
- Descreve-se aqui o sistema a ser desenvolvido.
```

**1.1.1 Descrição Textual.**

SISTEMA DE COMPRA DE MATÉRIA PRIMA - SCMP e Sistema de compra de
```

lanche-scl:

O sistema inicia quando o cliente se aproxima do balcão e solicita um lanche que consta no cardápio. O caixa informa o valor. O cliente paga ao caixa o valor correspondente e aguarda a entrega. O caixa envia o pedido à cozinha que prepara o lanche e posteriormente entrega ao cliente chamando pelo seu nome.

****1.1.2 Características gerais do "Sistema de Compra de Matéria Prima - SCMP e Sistema de Compra de Lanches - SCL":****

O sistema procurará fornecer uma solução abrangente para otimizar o processo de compra, atendimento e entrega de lanches em um estabelecimento. Isso visa melhorar a experiência do cliente, bem como a eficiência operacional interna. O sistema será projetado para servir como uma ferramenta valiosa tanto para os clientes finais quanto para os funcionários do estabelecimento.

Alguns dos clientes envolvidos neste sistema são os próprios clientes finais que frequentam o estabelecimento. Eles desempenham um papel crucial no processo de compra. O sistema será projetado para atender às necessidades de uma ampla gama de clientes, desde aqueles com conhecimento básico de tecnologia até os mais proficientes. Isso garante que a experiência de compra seja acessível e agradável para todos.

Os objetivos estabelecidos para o SCMP são multifacetados e visam aprimorar diversos aspectos. Primeiramente, o sistema irá procurar melhoria na eficiência e precisão do atendimento ao cliente, reduzindo erros e diminuindo o tempo de espera. Além disso, facilitará na escolha de lanches por parte dos clientes e simplificará o processo de pedido e pagamento. A otimização da comunicação entre atendentes, caixa, cozinheiro e cliente será uma prioridade. O sistema também integrará perfeitamente a cozinha para garantir uma entrega rápida e eficiente e permitir que os clientes acompanhem o status de seus pedidos.

No que diz respeito às funções do sistema, ele desempenhará um papel central na realização dos objetivos mencionados anteriormente. Isso inclui a consulta do cardápio, a facilitação do pedido e pagamento, a comunicação interna, a integração com a cozinha e o acompanhamento do status do pedido. Além disso, o sistema registrará informações essenciais de compra de matéria-prima para os lanches, proporcionando uma visão abrangente das operações.

Por fim, os atributos não funcionais envolvem velocidade de resposta das interfaces para garantir que a experiência do usuário seja ágil. O modelo de persistência de dados, como sistemas de gerenciamento de banco de dados, armazenará informações críticas. Além disso, as restrições de design, incluindo regulamentações legais como a Lei Geral de Proteção de Dados serão respeitadas. A usabilidade, confiabilidade, segurança e eficiência são atributos de qualidade que devem ser mantidos para garantir a satisfação dos usuários e o sucesso geral do sistema.

O projeto "SISTEMA DE COMPRA DE MATÉRIA PRIMA - SCMP" é uma iniciativa abrangente que visa melhorar a experiência dos seus usuários e a eficiência

operacional por meio de funções bem definidas e requisitos funcionais e não funcionais cuidadosamente considerados. Com os objetivos estabelecidos e os usuários em mente, este sistema promete ser uma solução valiosa para a organização como um todo.

1.2 Plataforma de desenvolvimento (O equipamento dos desenvolvedores e ferramentas de software)

```diff

- Descreve-se aqui uma primeira visão das tecnologias para desenvolvimento do projeto de software.

```

Uma combinação de Python, JavaScript e Ruby on Rails para desenvolver aplicativo web abrangente e escalável.

****Hardware necessário:****

Um computador com capacidade de 16 a 32 GB de RAM, 256GB de SSD, I7, com Acesso a Internet e a virtualização AWS.

Python:

- > IDE: Visual Studio Code ou Jupyter Notebook.
- > Frameworks: Django, para desenvolvimento web.
- > Gerenciador de pacotes: pip e virtualenv para gerenciar dependências e ambientes virtuais.
- > Banco de dados: MySQL para produção.

****JavaScript:****

- > IDE: Visual Studio Code.
- > Frameworks: React para desenvolvimento de interfaces de usuário interativas.
- > Gerenciador de pacotes: npm (Node Package Manager) para gerenciar dependências.
- > Banco de dados: MySQL.

****Ruby on Rails:****

- > IDE: RubyMine e Visual Studio Code.
- > Framework: Ruby on Rails para desenvolvimento web.
- > Gerenciador de pacotes: Bundler para gerenciar as dependências do projeto.
- > Banco de dados: MySQL, para ambiente de produção.

****Desenvolvedor/es:****

Desenvolvedor Full Stack, Desenvolvedor Mobile, Desenvolvedor de Banco de Dados, Gerente de Projeto, Engenheiro-Arquiteto de Software, Analista de Testes, com único desenvolvedor assumindo estes múltiplos papéis.

1.3 Plataforma de operação (O equipamento do cliente/usuário do sistema)

```diff

- Descreve-se aqui uma primeira visão das tecnologias para operacionalização do sistema aos usuários.

...

A plataforma de operação refere-se ao ambiente em que o sistema ou aplicativo é executado no equipamento do cliente ou usuário.

Para o sistema de controle de lanches definido anteriormente, abaixo estão exemplos de tecnologias para a operacionalização em diferentes plataformas.

#### 1. Operação em Computadores:

> Hardware: Processador Intel Core i5, 8 GB de RAM, disco rígido de 500 GB.

> Sistema Operacional: Windows 10, macOSX, Ubuntu Linux.

> Navegador Web: Google Chrome, Mozilla Firefox, Microsoft Edge.

> Banco de Dados: MySQL.

> Linguagens de Programação: Python, JavaScript, Ruby.

> Frameworks e Bibliotecas: Django (Python), React (JavaScript), Ruby on Rails (Ruby).

#### 2. Operação para Dispositivos Móveis:

> Hardware: Processador Qualcomm Snapdragon, 4 GB de RAM, armazenamento interno de 64 GB.

> Sistema Operacional: Android, iOS.

> Navegador Web: Google Chrome, Safari.

> Banco de Dados: SQLite.

> Linguagens de Programação: Python, JavaScript, Ruby.

> Frameworks e Bibliotecas: Django (Python), React Native (JavaScript), RubyMotion (Ruby).

#### 3. Operação para Servidores Web:

> Hardware: Servidores em nuvem.

> Sistema Operacional: Linux Ubuntu Server.

> Servidor Web: Nginx, Apache.

> Banco de Dados: MySQL.

> Linguagens de Programação: Python, JavaScript, Ruby.

> Frameworks e Bibliotecas: Django (Python), Node.js (JavaScript), Ruby on Rails (Ruby).

A plataforma de operação deve ser suficiente para atender a uma variedade de dispositivos, garantindo a acessibilidade e a usabilidade do sistema para o maior número possível de usuários.

### ## 1.4 Definições e siglas (quaisquer siglas utilizadas no domínio, do vocabulário do usuário)

```diff

- Descreve-se aqui a definição de todas as siglas, abreviações e termos usados.

...

1.5 Perspectiva do produto

1.5.1 Modos de operação (Meios de acesso ao sistema/Arquitetura do sistema)

```diff

- Identificam-se aqui os modos requeridos de operação, tais como: Back-End-

Front-End, Móvel, Stand-Alone, ...  
...

**\*\*Modo de operação baseado em interface gráfica de usuário (GUI):\*\*** o sistema é acessado por meio de uma interface gráfica interativa, como botões, menus e janelas. Exemplo: selecionar itens do menu, fazer pedidos e visualizar o status do pedido.

**\*\*Modo de operação baseado em aplicativos móveis:\*\*** o sistema é acessado por meio de um aplicativo móvel instalado em um dispositivo móvel. Os usuários interagem com o sistema por meio da interface do aplicativo. Exemplo: o aplicativo móvel de controle de lanches permite aos usuários fazer pedidos, personalizar itens e rastrear o status da entrega.

**\*\*Modo de operação baseado em navegador web:\*\*** o sistema é acessado por meio de um navegador web em um dispositivo conectado à internet. Os usuários interagem com o sistema por meio de uma interface web. Exemplo: o sistema de controle de lanches é acessado por meio de um site, onde os usuários podem fazer pedidos, gerenciar suas preferências e acompanhar o status dos pedidos.

### 1.5.2 Requisitos de adaptação ao ambiente (Aspectos legais para aderência a legislação, ambiente de operação crítica como indústria, automação, protocolos de comunicação específicos)

```diff

- Definem-se aqui possíveis requisitos de adaptação do produto aos ambientes particulares onde ele será implantado. Por exemplo, parâmetros e métodos de configuração requeridos para ambientes específicos devem ser descritos aqui.

...

Definem-se aqui possíveis requisitos de adaptação do produto aos ambientes particulares onde ele será implantado. Por exemplo, parâmetros e métodos de configuração requeridos para ambientes específicos devem ser descritos aqui.

| Número de ordem | Requisito | Detalhes |
|-----------------|-----------|----------|
|-----------------|-----------|----------|

| | | |
|---|---|---|
| 1 | Configuração de ticket de venda e da Nota Fiscal Eletrônica | Configuração dos campos de formulário com interface responsiva. |
|---|---|---|

1.6 Funções do produto (Funções básicas, R1.1 ..., R1.2 ... No geral, o que o sistema deve fazer?)

```diff

- Identificam-se aqui as principais funções que o produto desempenhará, descrevendo de forma sintética o objetivo de cada uma.

...

No geral, o sistema deve fazer:

- \* R1.0 Obter o valor faturado no dia.

- \* R1.1 Melhorar a eficiência e a precisão do atendimento ao cliente e reduzir erros e agilizar o tempo de espera dos clientes.

- \* R1.2 Consultar o cardápio de lanches disponíveis, facilitar a escolha dos clientes

e agilizar o processo de solicitação.

- \* R1.3 Realizar o pedido e efetuar o pagamento de maneira rápida e conveniente.

- \* R1.4 Otimizar a comunicação entre os atendentes, caixa, cozinheiro e cliente, garantindo que todas as etapas do processo sejam realizadas de forma eficiente e coordenada.

- \* R1.5 Integrar com a cozinha permitindo que o pedido seja enviado e preparado de maneira ágil, garantindo uma entrega rápida e eficiente ao cliente.

- \* R1.6 Permitir que o cliente acompanhe o status do pedido, desde o momento da solicitação até a entrega do lanche.

- \* R1.7 Obter informações de compra de matéria prima para os lanches.

- \* R1.8 Proporcionar experiência de compra agradável e eficiente para os clientes.

- \* R1.9 Facilitar o gerenciamento interno do estabelecimento.

- \* R1.10 Controlar o consumo de matéria prima;

- \* R1.11 Conhecer o número de clientes atendidos;

- \* R1.12 Conhecer quantos e quais lanches e bebidas foram entregues.

## 1.7 Características dos usuários (identificação da formação dos usuários para suas especialidades no uso do sistema, tal como ensino-médio, graduação, especialista, engenheiro, doutor, etc...)

```diff

- Descrevem-se aqui as principais características dos grupos de usuários esperados para o produto, tais como cargo ou função, permissão de acesso, frequência de uso, nível de instrução, proficiência no processo de negócio e proficiência em informática.

```

## Principais características dos grupos de usuários esperados:

```
<table style="border-collapse: collapse; width: 100%; color: #000080;"> <!--
```

Aplicando Azul Marinho em todo o conteúdo da tabela -->

```
 <thead style="width: 25%; border: 2px solid black;">
```

```
 <th style="width: 25%; border: 2px solid black;">
```

```
 Usuários
```

```
 </th>
```

```
 <th style="width: 25%; border: 2px solid black;">
```

```
 Descrição
```

```
 </th>
```

```
 </thead>
```

```
 <tbody style="width: 25%; border: 2px solid black;">
```

```
 <tr>
```

```
 <td style="width: 25%; border: 2px solid black;">
```

```
 Clientes:
```

```
 </td>
```

```
 <td style="width: 25%; border: 2px solid black;">
```

Os usuários finais que acessam o sistema para fazer pedidos de lanches. Os clientes terão acesso a funcionalidades específicas do sistema, como visualizar o cardápio, fazer pedidos, acompanhar o status dos pedidos e fazer pagamentos. Os clientes podem usar o sistema com frequência variável,

dependendo de sua preferência e necessidade de fazer pedidos de lanches. Podem variar em nível de instrução, desde usuários com nível básico, até usuários com nível de instrução mais elevado. Conhecimento básico sobre o processo de fazer pedidos de lanches, como selecionar itens, personalizar pedidos e acompanhar o status dos pedidos.

Atendentes/Cozinheiro/as	Os funcionários responsáveis por receber os pedidos dos clientes e gerenciar o processo de preparação e entrega dos lanches. Os atendentes terão acesso a funcionalidades adicionais, como confirmar pedidos, atualizar o status dos pedidos e gerenciar o estoque. Os atendentes usarão o sistema de forma contínua durante o horário de trabalho para gerenciar os pedidos e o fluxo de trabalho. Nível de instrução médio ou equivalente. Bom conhecimento do processo de negócio de atendimento de pedidos de lanches e sejam proficientes em informática para realizar suas tarefas diárias.
Caixa(Gerentes)	Os responsáveis pela supervisão e administração do sistema, gerenciando o estoque, preços, relatórios e outras tarefas relacionadas. Os gerentes terão permissões mais amplas, podendo acessar todas as funcionalidades do sistema, incluindo a administração de usuários, relatórios e configurações gerais. Os gerentes usarão o sistema regularmente para monitorar as operações, gerar relatórios e tomar decisões relacionadas ao negócio. Nível de instrução mais elevado, como graduação ou experiência profissional relevante. Proficiência no processo de negócio e proficiência em informática. Amplo conhecimento do processo de negócio, incluindo operações, finanças e relatórios. Devem ter boa proficiência em informática para gerenciar efetivamente o sistema e suas configurações.

## ## 1.8 Restrições (Lei Geral de proteção de dados, etc...)

``diff

- Descrevem-se aqui aspectos técnicos e gerenciais que possam limitar as opções dos desenvolvedores, tais como restrições legais.

...

## 1.9 Hipóteses de trabalho (Sistema operacional, versão de ferramentas de software, licenças de bibliotecas e de sub-sistemas)

```diff

- Descrevem-se aqui fatores que não são restrições limitativas do desempenho, como na subseção anterior, mas fatores cuja alteração requer modificações a ER, como, por exemplo, versão a ser utilizada do ambiente operacional ou plataforma de desenvolvimento.

...

2 Requisitos específicos

2.1 Interfaces externas

2.1.1 Visão geral (Tipos de interface, relatórios, gráficos, dashboards)

```diff

- Descreve-se aqui, de forma detalhada, todas as entradas e saídas do produto.

...

### 2.1.2 Requisitos para interfaces gráficas de usuário (mokups/wireframes)

```diff

- Sugere-se, no caso de interfaces gráficas, a inclusão dos seguintes elementos:

- Um esboço do layout gráfico sugerido para a interface;
- Uma descrição dos relacionamentos com outras interfaces;
- Um diagrama de estados/atividades, caso necessário para melhor entender-se o comportamento requerido da interface;
- Uma lista dos campos de dados da interface;
- Uma lista dos comandos da interface;
- BPM;

...

2.2 Requisitos funcionais

2.2.1 Diagramas de casos de uso (Modelo UML de Casos de Uso)

```diff

- Incluir todos os casos de uso que se pretende implementar em uma liberação. Pode-se incluir ainda: um certo caso de uso e seus relacionamentos, todos os casos de uso para um certo ator.

...

<table style="border-collapse: collapse; width: 100%; color: #000080;"> <!--

Aplicando Azul Marinho em todo o conteúdo da tabela -->

<thead style="width: 25%; border: 2px solid black;">

<th style="width: 25%; border: 2px solid black;">

Diagrama de Casos de Uso

</th>

</thead>

```
<tbody style="width: 25%; border: 2px solid black;">
```

```
<tr>
```

```
<td colspan="2" class="bold-title">
```

```
DESCRIÇÃO TEXTUAL DOS CASOS DE USO (Ver: 1.1.1 Descrição
Textual.)
```

SISTEMA DE COMPRA DE MATÉRIA PRIMA - SCMP e Sistema de compra de  
lanche-scl:

O sistema inicia quando o cliente se aproxima do balcão e solicita um lanche que  
consta no cardápio. O caixa informa o valor. O cliente paga ao caixa o valor  
correspondente e aguarda a entrega. O caixa envia o pedido à cozinha que  
prepara o lanche e posteriormente entrega ao cliente chamando pelo seu nome.

```
</td>
```

```
</tr>
```

```
<tr>
```

```
<td style="width: 25%; border: 2px solid black;">
```

```
<img src=https://github.com/radamesp7/UnoEngenharia-I-
2023/blob/main/usecase.png?raw=true align=middle>
```

```
</td>
```

```
</tr>
```

```
<tr>
```

```
<td style="width: 25%; border: 2px solid black;">
```

```
@startuml

```

```
!theme cloudscape-design

```

```
top to bottom direction

```

```
!define BACKGROUND_COLOR white

```

```
skinparam backgroundColor BACKGROUND_COLOR

```

```
actor Atendente as Atend

```

```
actor Caixa as Caixa

```

```
actor Cliente as Cli

```

```
actor Cozinheiro as Cozin

```

```
Atend <|-- Caixa #orange;line:orange;line.bold;text:orange

```

```
rectangle "SISTEMA DE COMPRA DE MATERIA PRIMA - SCMP" as
Sistema #white;line:black;line.bold;text:black; {

```

```
 usecase "Atender Cliente" as At

```

```
 usecase "Consultar Cardápio" as Card

```

```
 usecase "Solicitar Lanche" as Sol

```

```
 usecase "Aguardar Entrega" as Ag

```

```
 usecase "Enviar Pedido à Cozinha" as Env

```

```
 usecase "Preparar Lanche" as Prep

```

```
 usecase "Entregar Lanche" as Ent

```

```
 }

```

```
 Cli -right- Sol #blue;line:blue;line.bold;text:blue

```

```
 Cli -right- Card #blue;line:blue;line.bold;text:blue
```

```


```

```
 Cli -right- Pag #blue;line:blue;line.bold;text:blue

```

```
 Cli -right- Ent #blue;line:blue;line.bold;text:blue

```

```
 Cli -right- Ag #blue;line:blue;line.bold;text:blue

```

```
 (At) <|-- (Ent)
```

#orange;line:orange;line.bold;text:orange  	Atend -- At	#blue;line:blue;line.bold;text:blue  
	Caixa -- Pag	#blue;line:blue;line.bold;text:blue
	Caixa -- Env	#blue;line:blue;line.bold;text:blue
	Cozin -left- Prep	#blue;line:blue;line.bold;text:blue
	Cozin -left- Ent	#blue;line:blue;line.bold;text:blue
	Env ..> Pag	#red;line:red;text:red : <<include>>
	Prep ..> Env	#red;line:red;text:red : <<include>>
	Ent ..> Prep	#red;line:red;text:red : <<include>>
	Card ..> Sol	#red;line:red;text:red : <<extend>>
	@enduml  	
	</td>	
	</tr>	
	</tbody>	
	</table>	

### ### 2.2.2 Fluxos dos casos de uso (Casos de Uso Expandidos e Diagramas de Atividades)

```diff

- Expansão de Casos de Uso, Atividades e Contratos:

- o Pré-condições para a realização do caso de uso;
- o Fluxo principal do caso de uso (sucesso), descrito na forma de uma sequência de passos;
- o Fluxos alternativos do caso de uso;
- o Descrições mais formais, como diagramas de estado ou de atividade, se a complexidade do caso de uso exigir;
- o Observações.

```

#### ### 2.2.2.1 Caso de Uso expandido "Solicitar Lanche":

<thead style="width: 25%; border: 2px solid black;">

<th style="width: 25%; border: 2px solid black;">

Caso de Uso: Solicitar Lanche <br>

Atores: Cliente <br>

Propósito: Registrar um pedido de alimento e/ou bebida. <br>

Descrição: O Cliente chega ao balcão de atendimento e solicita um pedido do cardápio e efetua o pagamento. <br>

Tipo: Primário, essencial. <br>

Referência (as funções básicas): R1.1, R1.2, R1.3, R1.6, R1.8, R1.10, R1.11,



R1.12

<br>

<br>

Sequências Típicas de Eventos:

</th>

</thead>

<table style="border-collapse: collapse; width: 100%; color: #000080;"> <!--  
Aplicando Azul Marinho em todo o conteúdo da tabela -->

<thead style="width: 25%; border: 2px solid black;">

<th style="width: 25%; border: 2px solid black;">

Ação do Ator

</th>

<th style="width: 25%; border: 2px solid black;">

Resposta do Sistema

</th>

</thead>

<tbody style="width: 25%; border: 2px solid black;">

<tr>

<td style="width: 25%; border: 2px solid black;">

1 - Chegar no balcão para atendimento

</td>

<td style="width: 25%; border: 2px solid black;">

</td>

</tr>

<tr>

<td style="width: 25%; border: 2px solid black;">

2 - Solicitar lanche do cardápio e, ou bebida

</td>

<td style="width: 25%; border: 2px solid black;">

3 - Informar valor do pedido

</td>

</tr>

<tr>

<td style="width: 25%; border: 2px solid black;">

4 - Pagar o pedido

</td>

<td style="width: 25%; border: 2px solid black;">

5 - Fornecer recibo

</td>

</tr>

<tr>

<td colspan="2" class="bold-title">

Exceções:

</td>

</tr>

<tr>

<td style="width: 25%; border: 2px solid black;">

6 - No passo 2 e 4 o cliente desiste do pedido.

```

 </td>
 <td style="width: 25%; border: 2px solid black;">

 </td>
 </tr>
 <tr>
 <td style="width: 25%; border: 2px solid black;">
 7 - No passo 4 não consegue pagar
 </td>
 <td style="width: 25%; border: 2px solid black;">

 </td>
 </tr>
 <tr>
 <td style="width: 25%; border: 2px solid black;">
 8 - No passo 2 não tem o lanche
 </td>
 <td style="width: 25%; border: 2px solid black;">

 </td>
 </tr>
</tbody>
</table>

```

### 2.2.2.2 Modelagem com o diagrama de Atividades UML do Caso de Uso expandido Solicitar Lanche, Item 2.2.2.1:

```

<table style="border-collapse: collapse; width: 100%; color: #000080;"> <!--
Aplicando Azul Marinho em todo o conteúdo da tabela -->
 <thead style="width: 25%; border: 2px solid black;">
 <th style="width: 25%; border: 2px solid black;">
 Diagrama de Atividades - UML
 </th>
 </thead>
 <tbody style="width: 25%; border: 2px solid black;">
 <tr>
 <td style="width: 25%; border: 2px solid black;">
 <img src=https://github.com/radamesp7/UnoEngenharia-I-
2023/blob/main/AtividadesSelecionarLanche.png?raw=true align=middle>
 </td>
 </tr>
 <tr>
 <td style="width: 25%; border: 2px solid black;">
 @startuml

 !theme cloudscape-design

 <style>

 activityDiagram {

 FontName arial


```

```

 FontSize 18

 diamond {

 BackgroundColor #007FFF

 FontColor White

 FontName arial

 FontSize 25

 }

 arrow {

 FontColor #FF0000

 FontName arial

 FontSize 15

 }

 partition {

 FontSize 25

 RoundCorner 10

 BackgroundColor #F8F8FF

 }

 note {

 FontColor #000000

 LineColor #20B2AA

 BackgroundColor transparent

 }

}

document {

 LineColor #000000

 RoundCorner 30

}

</style>


```

' --- Diagrama de Atividades --- <br>

|Cliente(ator)| <br>

start <br>

partition SCL-SCMP{ <br>

:1 - Chegar no balcão para atendimento; <br>

:2 - Solicitar lanche do cardápio e, ou bebida; <br>

|Cliente(ator)| <br>

if (lanche disponível?) then ([Sim]) <br>

|Sistema| <br>

:3 - Informar valor do pedido; <br>

|Cliente(ator)| <br>

:4 - Pagar o pedido; <br>

if (consegue pagar?) then ([Sim]) <br>

|Sistema| <br>

:5 - Fornecer recibo; <br>

else ([Não]) <br>

|Cliente(ator)| <br>

stop <br>

note right: Cliente desiste do pedido <br>

endif <br>

else ([Não]) <br>

```
|Cliente(ator)|

stop

note right: Cliente desiste do pedido

endif

}

|Cliente(ator)|

stop

@enduml

</td>

</tr>

</tbody>

</table>

```

```diff

- ATENÇÃO: No exercício "TO" da semana da EFAPI é para desenvolver 3 expansões de Casos de Uso, ou seja, mais dois Casos de Usos expandidos do diagrama geral de Casos de Uso, Item 2.2.1. com seus respectivos diagramas de atividades.

```

## ## 2.3 Requisitos não-funcionais

### 2.3.1 Requisitos de desempenho (Velocidade de banda, tempo de resposta das interfaces e/ou impressão de relatórios)

```diff

- Requisitos de desempenho devem ser especificados de forma quantitativa e mensurável.

```

### 2.3.2 Requisitos de dados persistentes (Sistemas de gerenciamento de banco de dados e modelo de persistência)

```diff

- Descrevem-se aqui estruturas lógicas de dados persistentes (que mantêm seu valor após a execução do programa) que sejam usadas pelo produto. Cada estrutura de dados pode ser, por exemplo, um arquivo convencional ou uma tabela em um banco de dados.

```

## INCLUIR AQUI O MODELO DE BANCO DE DADOS

### 2.3.3 Restrições ao desenho

```diff

- Restrições de projeto impostas por padrões externos, com influência da legislação, tipo legislação fiscal municipal, estadual e federal.

```

### 2.3.4 Atributos de Qualidade

```diff

- Indica os atributos de qualidade, seguindo as características e subcaracterísticas recomendadas pela norma ISO-9126, ou mais atual.

...

2.4 Objetos/Classes

2.4.1 Modelo Conceitual/Classes de Análise/Modelo de Domínio (Classes, Associações, nomes das associações, Multiplicidades e Atributos)

2.4.2 DSS – Diagramas de Sequência do Sistema (Eventos e Operações de Casos de Uso (denotando as mensagens entre os objetos do domínio para atender ao Caso de Uso)).

2.4.3 Contratos (das Operações do DSS)

2.4.4 Classes de Implementação - Diagrama de Classes (Classes, Associações, nomes das associações, Multiplicidades, Atributos e Métodos).

```diff

- Atribuição de responsabilidades com GRASP (General Responsibility Assignment Software Patterns) que são um conjunto de princípios e diretrizes para atribuição de responsabilidades em projetos de software orientados a objetos.

...

Atribuição de responsabilidades com GRASP (General Responsibility Assignment Software Patterns) que são um conjunto de princípios e diretrizes para atribuição de responsabilidades em projetos de software orientados a objetos.

\*\*\*

## # 3 Análise de UCP

```diff

- Com as tabelas de escopo de valor do produto e tempo de desenvolvimento com Use Case Points - UCP. (COCOMO, Planning Poker e APF).Descreve-se aqui uma primeira visão das tecnologias para operacionalização.

...

#

Referências:

IEEE Std. 830 – 1993. IEEE Recommended Practice for Software Requirements Specifications.

IEEE ISO/IEC/IEEE 29148 – 2011. IEEE Systems and software engineering — Life cycle processes — Requirements engineering

```diff

- OBSERVAÇÃO: Os itens deste modelo de especificação, recomendado pela IEEE, poderão ser complementados com novos itens caso sejam justificáveis.

...



## 5. Exemplos de Códigos:

### Tabela:

#### 1.7 Características dos usuários (identificação da formação dos usuários para suas especialidades no uso do sistema, tal como ensino-médio, graduação, especialista, engenheiro, doutor, etc...)

- Descrevem-se aqui as principais características dos grupos de usuários esperados para o produto, tais como cargo ou função, permissão de acesso, frequência de uso, nível de instrução, proficiência no processo de negócio e proficiência em informática.

#### Principais características dos grupos de usuários esperados:

| Usuários                 | Descrição                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Clientes:                | Os usuários finais que acessam o sistema para fazer pedidos de lanches. Os clientes terão acesso a funcionalidades específicas do sistema, como visualizar o cardápio, fazer pedidos, acompanhar o status dos pedidos e fazer pagamentos. Os clientes podem usar o sistema com frequência variável, dependendo de sua preferência e necessidade de fazer pedidos de lanches. Podem variar em nível de instrução, desde usuários com nível básico, até usuários com nível de instrução mais elevado. Conhecimento básico sobre o processo de fazer pedidos de lanches, como selecionar itens, personalizar pedidos e acompanhar o status dos pedidos.                                                                                                                                                   |
| Atendentes/Cozinheiro/as | Os funcionários responsáveis por receber os pedidos dos clientes e gerenciar o processo de preparação e entrega dos lanches. Os atendentes terão acesso a funcionalidades adicionais, como confirmar pedidos, atualizar o status dos pedidos e gerenciar o estoque. Os atendentes usarão o sistema de forma contínua durante o horário de trabalho para gerenciar os pedidos e o fluxo de trabalho. Nível de instrução médio ou equivalente. Bom conhecimento do processo de negócio de atendimento de pedidos de lanches e sejam proficientes em informática para realizar suas tarefas diárias.                                                                                                                                                                                                      |
| Caixa(Gerentes)          | Os responsáveis pela supervisão e administração do sistema, gerenciando o estoque, preços, relatórios e outras tarefas relacionadas. Os gerentes terão permissões mais amplas, podendo acessar todas as funcionalidades do sistema, incluindo a administração de usuários, relatórios e configurações gerais. Os gerentes usarão o sistema regularmente para monitorar as operações, gerar relatórios e tomar decisões relacionadas ao negócio. Nível de instrução mais elevado, como graduação ou experiência profissional relevante. Proficiência no processo de negócio e proficiência em informática. Amplo conhecimento do processo de negócio, incluindo operações, finanças e relatórios. Devem ter boa proficiência em informática para gerenciar efetivamente o sistema e suas configurações. |

### Código:

#### ## 1.7 Características dos usuários (identificação da formação dos usuários para suas especialidades no uso do sistema, tal como ensino-médio, graduação, especialista, engenheiro, doutor, etc...)

```diff

- Descrevem-se aqui as principais características dos grupos de usuários esperados para o produto, tais como cargo ou função, permissão de acesso, frequência de uso, nível de instrução, proficiência no processo de negócio e proficiência em informática.

```

#### ## Principais características dos grupos de usuários esperados:

```
<table style="border-collapse: collapse; width: 100%; color: #000080;"> <!--
```

```
Aplicando Azul Marinho em todo o conteúdo da tabela -->
```

```
 <thead style="width: 25%; border: 2px solid black;">
```

```
 <th style="width: 25%; border: 2px solid black;">
```

```
 Usuários
```

```
 </th>
```

```
 <th style="width: 25%; border: 2px solid black;">
```

```
 Descrição
```

```
 </th>
```

</thead>

<tbody style="width: 25%; border: 2px solid black;">

<tr>

<td style="width: 25%; border: 2px solid black;">

Clientes:

</td>

<td style="width: 25%; border: 2px solid black;">

Os usuários finais que acessam o sistema para fazer pedidos de lanches. Os clientes terão acesso a funcionalidades específicas do sistema, como visualizar o cardápio, fazer pedidos, acompanhar o status dos pedidos e fazer pagamentos. Os clientes podem usar o sistema com frequência variável, dependendo de sua preferência e necessidade de fazer pedidos de lanches. Podem variar em nível de instrução, desde usuários com nível básico, até usuários com nível de instrução mais elevado. Conhecimento básico sobre o processo de fazer pedidos de lanches, como selecionar itens, personalizar pedidos e acompanhar o status dos pedidos.

</td>

</tr>

<tr>

<td style="width: 25%; border: 2px solid black;">

Atendentes/Cozinheiro/as

</td>

<td style="width: 25%; border: 2px solid black;">

Os funcionários responsáveis por receber os pedidos dos clientes e gerenciar o processo de preparação e entrega dos lanches. Os atendentes terão acesso a funcionalidades adicionais, como confirmar pedidos, atualizar o status dos pedidos e gerenciar o estoque. Os atendentes usarão o sistema de forma contínua durante o horário de trabalho para gerenciar os pedidos e o fluxo de trabalho. Nível de instrução médio ou equivalente. Bom conhecimento do processo de negócio de atendimento de pedidos de lanches e sejam proficientes em informática para realizar suas tarefas diárias.

</td>

</tr>

<tr>

<td style="width: 25%; border: 2px solid black;">

Caixa(Gerentes)

</td>

<td style="width: 25%; border: 2px solid black;">

Os responsáveis pela supervisão e administração do sistema, gerenciando o estoque, preços, relatórios e outras tarefas relacionadas. Os gerentes terão permissões mais amplas, podendo acessar todas as funcionalidades do sistema, incluindo a administração de usuários, relatórios e configurações gerais. Os gerentes usarão o sistema regularmente para monitorar as operações, gerar relatórios e tomar decisões relacionadas ao negócio. Nível de instrução mais elevado, como graduação ou experiência profissional relevante. Proficiência no processo de negócio e proficiência em informática. Amplo conhecimento do processo de negócio, incluindo operações, finanças e relatórios. Devem ter boa proficiência em informática para gerenciar efetivamente o sistema e suas configurações.



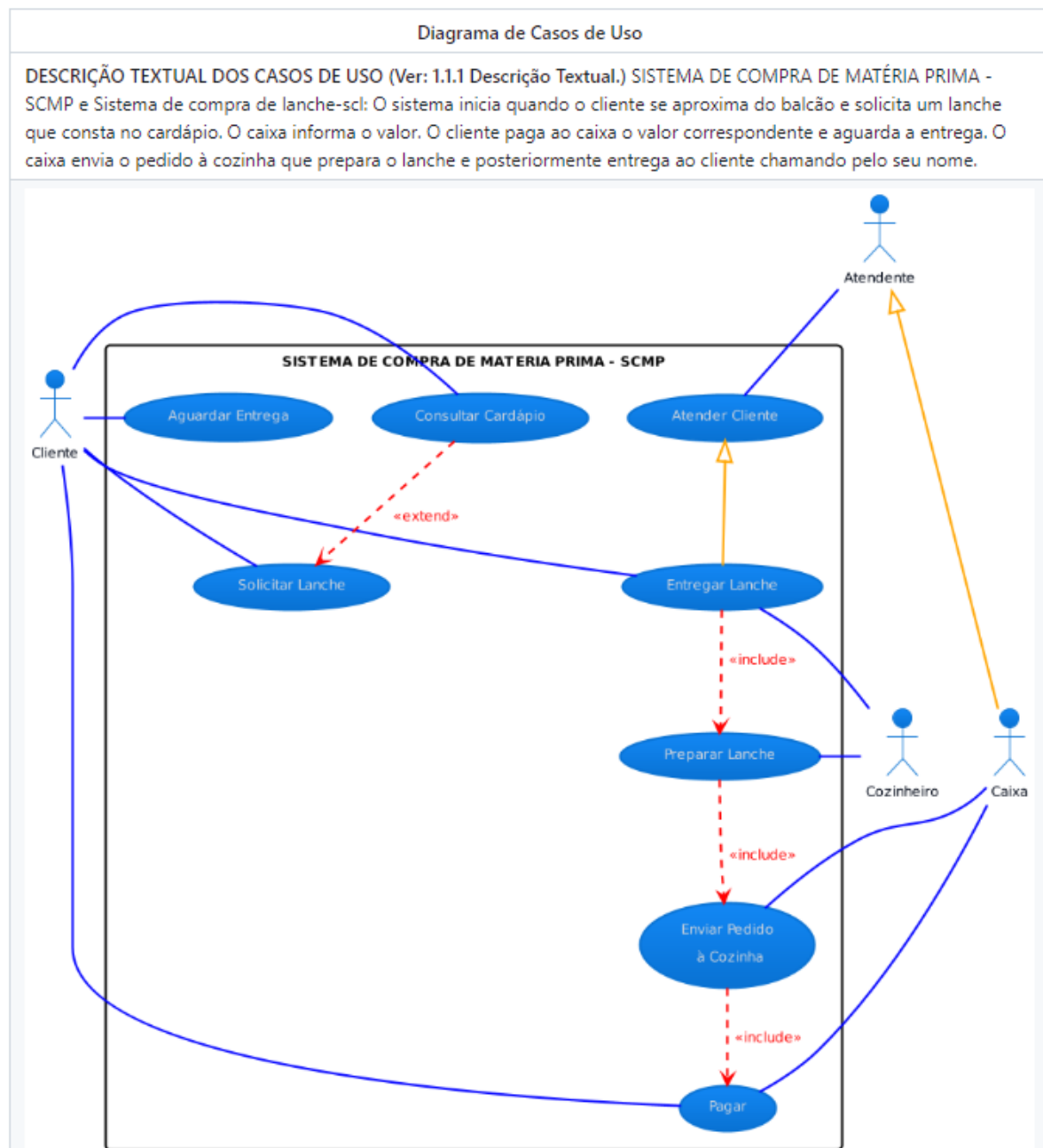
```
 </td>
 </tr>
 </tbody>
 </table>
```

Figura:

## 2.2 Requisitos funcionais

### 2.2.1 Diagramas de casos de uso (Modelo UML de Casos de Uso)

- Incluir todos os casos de uso que se pretende implementar em uma liberação. Pode-se incluir ainda: um certo ca



@startuml

!theme cloudscape-design

- 
- 
- 

## Código:

### ## 2.2 Requisitos funcionais

#### ### 2.2.1 Diagramas de casos de uso (Modelo UML de Casos de Uso)

```
```diff
```

- Incluir todos os casos de uso que se pretende implementar em uma liberação. Pode-se incluir ainda: um certo caso de uso e seus relacionamentos, todos os casos de uso para um certo ator.

```
```
```

```
<table style="border-collapse: collapse; width: 100%; color: #000080;"> <!-- Aplicando Azul Marinho em todo o conteúdo da tabela -->
```

```
<thead style="width: 25%; border: 2px solid black;">
```

```
<th style="width: 25%; border: 2px solid black;">
```

```
Diagrama de Casos de Uso
```

```
</th>
```

```
</thead>
```

```
<tbody style="width: 25%; border: 2px solid black;">
```

```
<tr>
```

```
<td colspan="2" class="bold-title">
```

```
DESCRIÇÃO TEXTUAL DOS CASOS DE USO (Ver: 1.1.1 Descrição Textual.)
```

SISTEMA DE COMPRA DE MATÉRIA PRIMA - SCMP e Sistema de compra de lanche-scl:

O sistema inicia quando o cliente se aproxima do balcão e solicita um lanche que consta no cardápio. O caixa informa o valor. O cliente paga ao caixa o valor correspondente e aguarda a entrega. O caixa envia o pedido à cozinha que prepara o lanche e posteriormente entrega ao cliente chamando pelo seu nome.

```
</td>
```

```
</tr>
```

```
<tr>
```

```
<td style="width: 25%; border: 2px solid black;">
```

```

```

```
</td>
```

```
</tr>
```

```
<tr>
```

```
<td style="width: 25%; border: 2px solid black;">
```

```
@startuml

```

```
!theme cloudscape-design

```

```
top to bottom direction

```

```
!define BACKGROUND_COLOR white

```

```
skinparam backgroundColor BACKGROUND_COLOR

```

```
actor Atendente as Atend

```

```
actor Caixa as Caixa

```

```
actor Cliente as Cli

```

```
actor Cozinheiro as Cozin

```

```
Atend <|-- Caixa #orange;line:orange;line.bold;text:orange

```

```
rectangle "SISTEMA DE COMPRA DE MATERIA PRIMA - SCMP" as Sistema
```

```
#white;line:black;line.bold;text:black; {

```

```
usecase "Atender Cliente" as At

```

```
usecase "Consultar Cardápio" as Card

```

```
usecase "Solicitar Lanche" as Sol

```

```
usecase "Aguardar Entrega" as Ag

```

```
usecase "Enviar Pedido à Cozinha" as Env

```

```
usecase "Preparar Lanche" as Prep

```

```
usecase "Entregar Lanche" as Ent

```

```
}

```

```
Cli -right- Sol #blue;line:blue;line.bold;text:blue

```

```
Cli -right- Card #blue;line:blue;line.bold;text:blue

```

```
Cli -right- Pag #blue;line:blue;line.bold;text:blue

```

```
Cli -right- Ent #blue;line:blue;line.bold;text:blue

```

```
Cli -right- Ag #blue;line:blue;line.bold;text:blue

```

```
(At) <|-- (Ent) #orange;line:orange;line.bold;text:orange

```

```
Atend -- At #blue;line:blue;line.bold;text:blue

```

```
Caixa -- Pag #blue;line:blue;line.bold;text:blue

```

```
Caixa -- Env #blue;line:blue;line.bold;text:blue

```

Cozin -left- Prep	#blue;line:blue;line.bold;text:blue  
Cozin -left- Ent	#blue;line:blue;line.bold;text:blue  
Env ..> Pag	#red;line:red;text:red : <<include>>  
Prep ..> Env	#red;line:red;text:red : <<include>>  
Ent ..> Prep	#red;line:red;text:red : <<include>>  
Card ..> Sol	#red;line:red;text:red : <<extend>>  
@enduml  	
</td>	
</tr>	
</tbody>	
</table>	

Tabela:

2.2.2.1 Caso de Uso expandido "Solicitar Lanche":

Caso de Uso: Solicitar Lanche  
Atores: Cliente  
Propósito: Registrar um pedido de alimento e/ou bebida.  
Descrição: O Cliente chega ao balcão de atendimento e solicita um pedido do cardápio e efetua o pagamento.  
Tipo: Primário, essencial.  
Referência (as funções básicas): R1.1, R1.2, R1.3, R1.6, R1.8, R1.10, R1.11, R1.12

Sequências Típicas de Eventos:

Ação do Ator	Resposta do Sistema
1 - Chegar no balcão para atendimento	-
2 - Solicitar lanche do cardápio e, ou bebida	3 - Informar valor do pedido
4 - Pagar o pedido	5 - Fornecer recibo
Exceções:	
6 - No passo 2 e 4 o cliente desiste do pedido.	-
7 - No passo 4 não consegue pagar	-
8 - No passo 2 não tem o lanche	-

Código:

```
2.2.2.1 Caso de Uso expandido "Solicitar Lanche":

<thead style="width: 25%; border: 2px solid black;">
<th style="width: 25%; border: 2px solid black;">
 Caso de Uso: Solicitar Lanche

 Atores: Cliente

 Propósito: Registrar um pedido de alimento e/ou bebida.

 Descrição: O Cliente chega ao balcão de atendimento e solicita um pedido do cardápio
e efetua o pagamento.

 Tipo: Primário, essencial.

 Referência (as funções básicas): R1.1, R1.2, R1.3, R1.6, R1.8, R1.10, R1.11, R1.12

 Sequências Típicas de Eventos:
</th>
</thead>
<table style="border-collapse: collapse; width: 100%; color: #000080;"> <!-- Aplicando
Azul Marinho em todo o conteúdo da tabela -->
 <thead style="width: 25%; border: 2px solid black;">
 <th style="width: 25%; border: 2px solid black;">
 Ação do Ator
 </th>
 <th style="width: 25%; border: 2px solid black;">
 Resposta do Sistema
```

```

</th>
</thead>
<tbody style="width: 25%; border: 2px solid black;">
 <tr>
 <td style="width: 25%; border: 2px solid black;">
 1 - Chegar no balcão para atendimento
 </td>
 <td style="width: 25%; border: 2px solid black;">
 —
 </td>
 </tr>
 <tr>
 <td style="width: 25%; border: 2px solid black;">
 2 - Solicitar lanche do cardápio e, ou bebida
 </td>
 <td style="width: 25%; border: 2px solid black;">
 3 - Informar valor do pedido
 </td>
 </tr>
 <tr>
 <td style="width: 25%; border: 2px solid black;">
 4 - Pagar o pedido
 </td>
 <td style="width: 25%; border: 2px solid black;">
 5 - Fornecer recibo
 </td>
 </tr>
 <tr>
 <td colspan="2" class="bold-title">
 Exceções:
 </td>
 </tr>
 <tr>
 <td style="width: 25%; border: 2px solid black;">
 6 - No passo 2 e 4 o cliente desiste do pedido.
 </td>
 <td style="width: 25%; border: 2px solid black;">
 —
 </td>
 </tr>
 <tr>
 <td style="width: 25%; border: 2px solid black;">
 7 - No passo 4 não consegue pagar
 </td>
 <td style="width: 25%; border: 2px solid black;">
 —
 </td>
 </tr>
 <tr>
 <td style="width: 25%; border: 2px solid black;">
 8 - No passo 2 não tem o lanche
 </td>
 <td style="width: 25%; border: 2px solid black;">
 —
 </td>
 </tr>

```

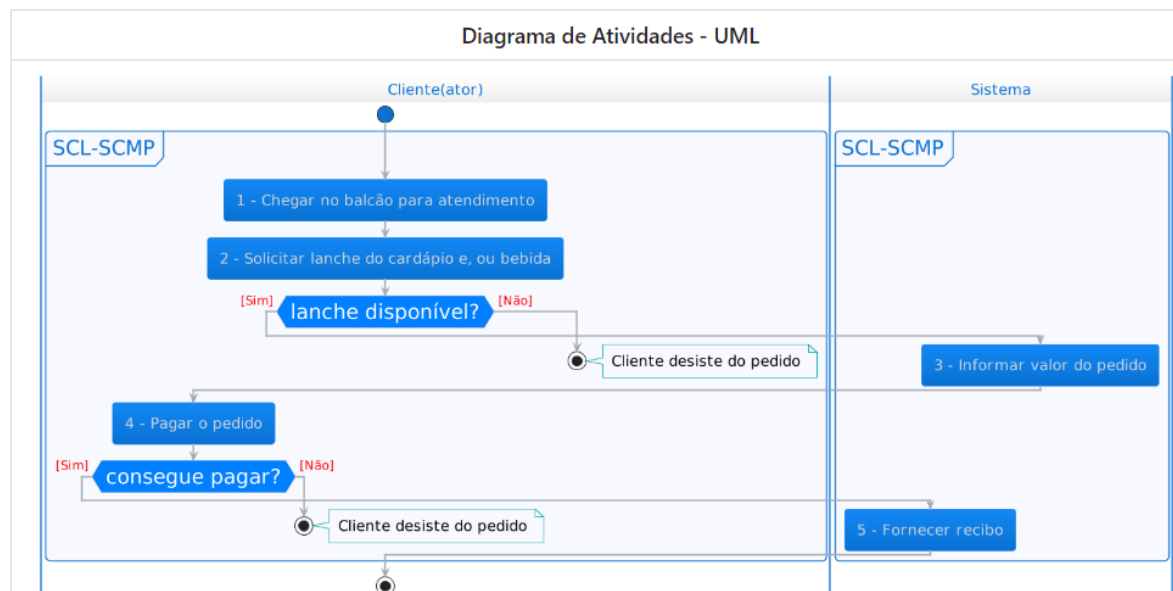
```

 </td>
 </tr>
</tbody>
</table>

```

## Figura:

2.2.2.2 Modelagem com o diagrama de Atividades UML do Caso de Uso expandido Solicitar Lanche, Item 2.2.2.1: [↗](#)



```

@startuml
!theme cloudscape-design
<style>
activityDiagram {
 FontName arial
 FontSize 18
 diamond {
 BackgroundColor #007FFF
 FontColor White
 FontName arial

```

- 
- 
- 

## Código:

### 2.2.2.2 Modelagem com o diagrama de Atividades UML do Caso de Uso expandido Solicitar Lanche, Item 2.2.2.1:

<table style="border-collapse: collapse; width: 100%; color: #000080;"> <!-- Aplicando Azul Marinho em todo o conteúdo da tabela -->

<thead style="width: 25%; border: 2px solid black;">

<th style="width: 25%; border: 2px solid black;">

Diagrama de Atividades - UML

</th>

</thead>

<tbody style="width: 25%; border: 2px solid black;">

<tr>

<td style="width: 25%; border: 2px solid black;">

<img src=https://github.com/radamesp7/UnoEngenharia-I-2023/blob/main/AtividadesSelecionarLanche.png?raw=true align=middle>

</td>

</tr>

<tr>

<td style="width: 25%; border: 2px solid black;">

@startuml <br>

!theme cloudscape-design <br>

<style> <br>

activityDiagram { <br>

FontName arial <br>

FontSize 18 <br>

diamond { <br>

BackgroundColor #007FFF <br>

FontColor White <br>

FontName arial <br>

FontSize 25 <br>

} <br>

arrow { <br>

FontColor #FF0000 <br>

FontName arial <br>

FontSize 15 <br>

} <br>

partition { <br>

FontSize 25 <br>

RoundCorner 10 <br>

BackgroundColor #F8F8FF <br>

} <br>

note { <br>

FontColor #000000 <br>

LineColor #20B2AA <br>

BackgroundColor transparent <br>

} <br>

} <br>

document { <br>

LineColor #000000 <br>

RoundCorner 30 <br>

} <br>

</style> <br>

' --- Diagrama de Atividades --- <br>

|Cliente(ator)| <br>

start <br>

partition SCL-SCMP{ <br>

:1 - Chegar no balcão para atendimento; <br>

:2 - Solicitar lanche do cardápio e, ou bebida; <br>

|Cliente(ator)| <br>



```
if (lanche disponível?) then ([Sim])

|Sistema|

:3 - Informar valor do pedido;

|Cliente(ator)|

:4 - Pagar o pedido;

if (consegue pagar?) then ([Sim])

|Sistema|

:5 - Fornecer recibo;

else ([Não])

|Cliente(ator)|

stop

note right: Cliente desiste do pedido

endif

else ([Não])

|Cliente(ator)|

stop

note right: Cliente desiste do pedido

endif

}

|Cliente(ator)|

stop

@enduml

</td>

</tr>

</tbody>

</table>

```

```
```diff
```
```