



**Universidade do Minho**

Escola de Engenharia

Licenciatura em Engenharia informática

Mestrado Integrado em Engenharia Informática

## **Unidade Curricular de Laboratórios de Informática IV**

Ano Letivo de 2023/2024

### **Cantinho das Encomendas**

**Ana Oliveira (a104536), André Campos (a104618),  
Beatriz Peixoto (a104170), Tomás Pinto (a104448)**

Novembro, 2024

# **LI4**

Data de Recepção	
Responsável	
Avaliação	
Observações	

## Cantinho das Encomendas

**Ana Oliveira (a104536), André Campos (a104618),  
Beatriz Peixoto (a104170), Tomás Pinto (a104448)**

Novembro, 2024

# Resumo

Este projeto, desenvolvido no âmbito da Unidade Curricular de Laboratórios de Informática IV, intitulado “Cantinho das Encomendas”, tem como objetivo desenvolver uma aplicação de modo a otimizar a gestão da produção de encomendas para a pastelaria “Cantinho da Doçura”.

A primeira fase deste trabalho encontra-se dividida em cinco etapas.

Na primeira etapa, foi estruturada a definição do projeto, desde a contextualização à definição de objetivos para a aplicação e dos recursos necessários ao seu desenvolvimento. Esta etapa terminou com o desenvolvimento de um diagrama de Gantt, de modo a planear detalhadamente o cronograma de desenvolvimento deste trabalho.

De seguida, na etapa de levantamento de requisitos funcionais e não funcionais, foram definidas as funcionalidades do sistema a desenvolver, servindo estes como uma base essencial para as seguintes etapas de desenvolvimento. Estes mesmos requisitos delimitam o funcionamento do sistema e como cada tarefa é executada.

Posteriormente, a especificação e modelação do *software* visa detalhar os aspetos estruturais e comportamentais do sistema. Com esse mesmo propósito, foram definidos um modelo de domínio e um diagrama de componentes, referentes aos aspetos estruturais, que permitem definir os vários elementos do sistema e as suas interações. Na modelação dos aspetos comportamentais, foram definidos diagramas de atividades, que especificam o fluxo das funcionalidades do sistema.

Na quarta etapa, foi criado um modelo lógico relacional com o objetivo de estruturar o sistema de dados a implementar, aumentando a eficiência na gestão de encomendas, utilizadores e outros elementos relevantes do sistema. Este modelo foi construído tendo por base os requisitos previamente definidos e a modelação efetuada na etapa anterior.

Por fim, termina-se esta primeira fase do trabalho com a definição e caracterização das diferentes interfaces da aplicação, através da execução de esboços que caracterizam o seu aspeto visual.

Na segunda fase deste projeto, foi implementado o sistema, como modelado nas diversas etapas anteriores, no ecossistema .NET, através da *framework Blazor*. Além disso, a base de dados anteriormente modelada foi criada através de um *script* do SQL Server.

**Área de Aplicação:** Desenho e desenvolvimento de software de gestão de uma linha de montagem.

**Palavras-Chave:** Bases de Dados Relacionais, *Unified Modelling Language*, Engenharia de *Software*, Engenharia de Requisitos, .NET, Blazor, SQL Server

# Índice

Resumo	i
Índice	ii
Índice de Figuras	iv
Índice de Tabelas	vi
1. Introdução	1
1.1. Contextualização	1
1.2.1. Motivação	1
1.2.2. Objetivos	2
1.3. Justificação e utilidade do sistema	2
1.4. Estabelecimento da identidade do projeto	3
1.5. Identificação dos recursos necessários	3
1.6. Maquete do sistema	4
1.7. Definição de um conjunto de medidas de sucesso	5
1.8. Plano de desenvolvimento	5
2. Levantamento e Análise de Requisitos	7
2.1. Apresentação da estratégia e método	7
2.2. Descrição geral dos requisitos levantados	7
2.2.1. Requisitos Funcionais	7
2.2.2. Requisitos Não Funcionais	29
2.3. Validação dos requisitos estabelecidos	29
3. Especificação e Modelação do Software	30
3.1. Apresentação geral da especificação	30
3.2. Aspetos estruturais	30
3.2.1. Modelo de Domínio	30
3.2.2. Diagrama de Componentes	35
3.3. Aspetos Comportamentais	39
3.3.1. Gestão de Contas do Utilizador	40
3.3.2. Gestão de Stock	41
3.3.3. Gestão de Favoritos	42
3.3.4. Gestão de Encomendas – Cliente	43
3.3.5. Gestão de Encomendas – Funcionário	44
4. Conceção do Sistema de Dados	46
4.1. Apresentação geral da estrutura do sistema de dados	46
4.2. Descrição detalhada dos vários elementos de dados e os seus relacionamentos	47
4.2.1. Descrição das tabelas	47
4.2.2. Especificação das colunas das tabelas (atributos)	49
4.2.3. Especificação das chaves e relacionamentos	54
5. Esboço dos Interfaces do Sistema	56

5.1. Estrutura geral das interfaces do sistema	56
5.2. Caracterização das interfaces	56
6. Implementação da Aplicação	60
6.1. Apresentação e descrição do processo de implementação realizado	60
6.1.1. Apresentação das ferramentas e recursos utilizados	60
6.1.2. Estrutura da implementação	60
6.1.3. Apresentação e Descrição do Sistema	61
6.2. Apresentação da aplicação e explicação dos serviços implementados	68
6.2.1. MainLayout	68
6.2.2. Página Inicial	68
6.2.3. Tarte	69
6.2.4. Carrinho	70
6.2.5. Área Pessoal	70
6.2.6. Login	71
6.2.7. Registo	71
6.2.8. Recuperar Password	72
6.2.9. FAQ	72
6.2.10. Favoritos	73
6.2.11. Menu de Texto	73
6.3. Análise e avaliação da aplicação desenvolvida	75
7. Conclusões e Trabalho Futuro	76
7.1. Conclusões	76
7.2. Trabalho Futuro	77
Referências	78
Lista de Siglas	79
Anexos	80
Anexo 1 – Lista completa dos mockups do sistema	80

# Índice de Figuras

Figura 1 – Palete de cores do sistema	3
Figura 2 – Logótipo da pastelaria	3
Figura 3 – Maquete do sistema	4
Figura 4 – Diagrama de Gantt	6
Figura 5 – Modelo de Domínio	30
Figura 6 – Modelo de Domínio (Utilizadores)	31
Figura 7 – Modelo de Domínio (Cliente)	32
Figura 8 – Modelo de Domínio (Encomenda)	33
Figura 9 – Modelo de Domínio (Funcionário)	33
Figura 10 – Modelo de Domínio (Stock)	34
Figura 11 – Modelo de Domínio (Produto)	35
Figura 12- Diagrama de Componentes	36
Figura 13 – Diagrama de Componentes (Avaliação)	36
Figura 14 – Diagrama de Componentes (Utilizadores)	37
Figura 15 – Diagrama de Componentes (Stock)	38
Figura 16 – Diagrama de Componentes (Produtos)	38
Figura 17 – Diagrama de Componentes (Encomendas)	39
Figura 18 – Gestão de Contas do Utilizador	40
Figura 19 – Gestão de Stock	41
Figura 20 – Gestão de Favoritos	42
Figura 21 – Gestão de Encomendas (Cliente)	43
Figura 22 – Gestão de Encomendas (Funcionário)	44
Figura 23 – Modelo Lógico do Sistema	46
Figura 24 – Mockup de login	56
Figura 25 – Mockup de registo	57
Figura 26 – Mockup de menu principal	57
Figura 27 – Mockup de “Carrinho de Compras”	58
Figura 28 – Mockup de “As minhas Encomendas”	58
Figura 29 – Mockup “Menu Funcionário”	59
Figura 30 – Mockup de stock total	59
Figura 31 – Menu MainLayout	68
Figura 32 – Página Inicial	69
Figura 33 – Página de Tarte	69
Figura 34 – Página de Carrinho de Compras	70
Figura 35 – Página da Área Pessoal	70
Figura 36 – Página de Login	71
Figura 37 – Página de Registo	72
Figura 38 – Página de Recuperar Password	72

Figura 39 – Página de Perguntas Frequentes	73
Figura 40 – Página de Favoritos	73
Figura 41 – Menu de Texto: Página Inicial	74
Figura 42 – Menu de Texto: Menu Cliente	74
Figura 43 – Menu de Texto: Menu Funcionário	75
Figura 44 – Mockup de recuperar password (inserir email)	80
Figura 45 – Mockup de recuperar password (inserir código de confirmação)	80
Figura 46 – Mockup de recuperar password (inserir nova password)	81
Figura 47 – Mockup da página da tarte	81
Figura 48 – Mockup da tarte nos favoritos	82
Figura 49 – Mockup da tarte sem stock	82
Figura 50 – Mockup dos favoritos	83
Figura 51 – Mockup da aba de Perguntas Frequentes	83
Figura 52 – Mockup de ver avaliações	84
Figura 53 – Mockup de deixar avaliação	84
Figura 54 – Mockup de “A Minha Conta”	85
Figura 55 – Mockup da encomenda	85
Figura 56 – Mockup de terminar sessão	86
Figura 57 – Mockup de encomendas em fila de espera	86
Figura 58 – Mockup de encomenda em fila de espera	87
Figura 59 – Mockup de encomendas em preparação	87
Figura 60 – Mockup de encomenda em preparação	88
Figura 61 – Mockup de alterar progresso da tarte	88
Figura 62 – Mockup de encomendas enviadas	89
Figura 63 – Mockup de encomenda enviada	89
Figura 64 – Mockup de stock de um material	90
Figura 65 – Mockup de alterar stock	90
Figura 66 – Mockup de alterar capacidade de stock	91
Figura 67 – Mockup de adicionar tipo de material	91
Figura 68 – Mockup de ver tartes disponíveis - funcionário	92
Figura 69 – Mockup de adicionar tarte	92
Figura 70 – Mockup de remover tarte	93
Figura 71 - Mockup “A Minha Conta” - Funcionário	93

# Índice de Tabelas

Tabela 1	8
Tabela 2	8
Tabela 3	9
Tabela 4	9
Tabela 5	10
Tabela 6	11
Tabela 7	11
Tabela 8	11
Tabela 9	12
Tabela 10	13
Tabela 11	13
Tabela 12	14
Tabela 13	14
Tabela 14	15
Tabela 15	15
Tabela 16	16
Tabela 17	16
Tabela 18	17
Tabela 19	17
Tabela 20	18
Tabela 21	19
Tabela 22	19
Tabela 23	20
Tabela 24	20
Tabela 25	21
Tabela 26	21
Tabela 27	22
Tabela 28	22
Tabela 29	23
Tabela 30	23
Tabela 31	24
Tabela 32	25
Tabela 33	26
Tabela 34	26
Tabela 35	27
Tabela 36	27
Tabela 37	28
Tabela 38	28



Tabela 39 – Descrição das Tabelas do Modelo Lógico	49
Tabela 40 – Tabela Utilizadores	49
Tabela 41 – Tabela Cliente_Favoritos	50
Tabela 42 – Tabela Avaliacao	50
Tabela 43 – Tabela CarrinhoDeCompras	50
Tabela 44 – Tabela Carrinho_Tarte	51
Tabela 45 – Tabela Encomenda	51
Tabela 46 – Tabela Pagamento	52
Tabela 47 – Tabela Montagem	52
Tabela 48 - Tabela Tarte	53
Tabela 49 – Tabela Tarte_Material	53
Tabela 50 – Tabela Material	54
Tabela 51 – Tabela TarteEncomenda	54

# 1. Introdução

## 1.1. Contextualização

A pastelaria Cantinho da Doçura foi fundada em maio de 1962, pela gentil e habilidosa senhora D. Odete. Esta pastelaria localiza-se na freguesia de Monsanto, pertencente ao município de Idanha-a-Nova, no distrito de Castelo Branco.

Trata-se de um lugar pequeno, mas aconchegante com mobília de madeira escura. O pouco número de mesas que existem, reflete a simplicidade e familiaridade deste espaço. O papel de parede florido e os azulejos portugueses no chão, exaltam a tradição nacional.

Além disso, o maravilhoso e duradouro aroma a café e a montra de vidro que expõe as deliciosas e caseiras tartes da D. Odete são os aspetos que mais convidam quem passa pela porta. Trata-se, assim, do lugar preferido de convivência dos habitantes da pequena freguesia.

Na pastelaria, além da D. Odete, sempre esteve o seu neto, Mário, que cresceu aos cuidados da avó e, desde criança, a ajudava na preparação das receitas.

Após a morte de D. Odete, Mário, sentiu necessidade de nunca deixar cair no esquecimento não só os sabores maravilhosos criados pela idosa e estimados por toda a comunidade, mas também o lugar onde cresceu e partilhou bonitas memórias com a sua avó.

Nesse sentido, para homenagear a sua avó, Mário tomou a decisão de dar a conhecer a todo o país a pastelaria “Cantinho da Doçura” e, para isso, criou uma página nas redes sociais onde divulgou a história da pastelaria, publicou os produtos que lá confeccionava e, por mensagem privada, respondia aos pedidos dos clientes.

### 1.2.1. Motivação

Com o aumento do número de clientes e de pedidos, Mário sentia várias dificuldades na organização da pastelaria, tais como:

- Registrar todos os pedidos enviados, para não ocorrer nenhuma perda;
- Conseguir manter-se atualizado sobre o stock disponível de modo a evitar aceitar pedidos de clientes, quando não há o material suficiente para os realizar;
- Gerir os atrasos das entregas dos clientes;
- Não confundir pedidos, o que levava à criação desnecessária de encomendas;

Após analisar estes problemas, Mário decidiu que a melhor solução seria pedir auxílio de uma empresa informática para criar uma aplicação que resolvesse essas dificuldades e que melhorasse o rendimento da sua pastelaria.

### **1.2.2. Objetivos**

Animado por finalmente ter a oportunidade de usufruir de uma pastelaria funcional, Mário definiu uma lista de objetivos que gostava de alcançar com a implementação da aplicação para a gestão da linha de montagem. Para isso, a aplicação:

- Faria a gestão dos pedidos, certificando-se que não existam produções desnecessárias, pedidos esquecidos e que os pedidos correspondessem às necessidades do cliente, fazendo com que o mesmo escolhesse quais as tartes que gostaria de receber;
- Iria verificar que um pedido só pudesse ser aceite caso houvessem os materiais necessários para a sua produção;
- Iria permitir aos clientes visualizar a produção dos seus pedidos, de modo a diminuir as reclamações de “atraso nos pedidos”;
- Iria permitir aos clientes uma estimativa de quando poderão receber a encomenda, pois a aplicação faria a estimativa com base nos dados (tempo de produção, tempo de envio, etc);

### **1.3. Justificação e utilidade do sistema**

Para além dos problemas que a aplicação vai resolver, a equipa informática apresentou alguns benefícios que a aplicação irá fornecer à pastelaria:

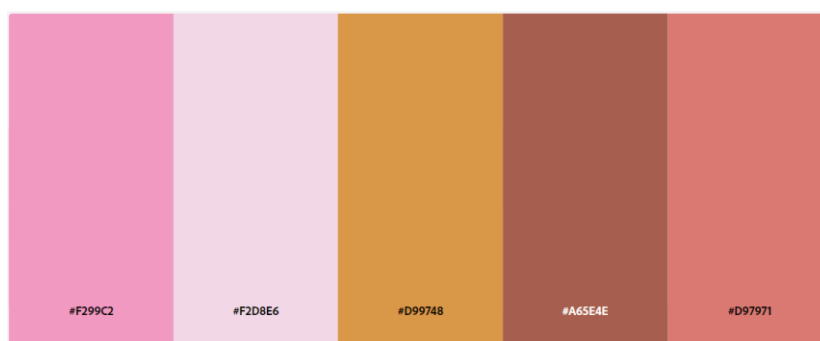
- Facilitar a produção de tartes, fazendo com que a gestão de pedidos fosse mais bem organizada
- Informar a quantidade existente no stock, indicando os produtos menos usados e indicando os produtos que faltam para a produção contínua de tartes. Isto faria com que houvesse uma melhor gestão na compra de material, minimizando custos em material pouco necessário;
- Clientes podem procurar informações da pastelaria e efetuar pedidos customizados de forma simples sem a intervenção do dono da pastelaria, Mário;
- Clientes conseguem ver os seus pedidos e as métricas dos seus pedidos (o tempo de confeição do pedido);
- Clientes mal-intencionados terão as suas ações bloqueadas pela aplicação, evitando que sejam produzidos bolos sem que seja realizado e confirmado o seu pagamento, evitando que os bolos sejam cancelados após estarem a ser processados e evitando que Mário tenha que lidar com esses clientes de forma direta;
- Mário receberia o feedback dos clientes sem precisar de perguntar diretamente;

## 1.4. Estabelecimento da identidade do projeto

O “Cantinho das Encomendas” é um sistema de software desenvolvido com o intuito de automatizar e otimizar a linha de montagem da produção de tartes nas confeitarias. Este projeto foi desenvolvido de forma a melhorar a eficiência e controlo da confeção, permitindo ao cliente um acompanhamento em tempo real da preparação da sua encomenda, desde o pedido à visualização da manufatura.

Este programa está a ser desenvolvido especificamente para a pastelaria “Cantinho da Doçura”, pela equipa de software da *Razen*, de modo a possibilitar a expansão da sua capacidade de produção e entrega, evitando o desperdício de recursos através da gestão de produção e inventário do sistema.

O “Cantinho das Encomendas” teve um tempo de produção de 5 meses, acumulando um custo total de 42.000 euros.



*Figura 1 – Paleta de cores do sistema*



*Figura 2 – Logótipo da pastelaria*

## 1.5. Identificação dos recursos necessários

Para garantir a realização correta e eficiente deste projeto, são necessários diversos recursos, tanto humanos como materiais.

Relativamente aos recursos humanos, é fundamental a colaboração do Mário e dos clientes e, também é essencial definir uma equipa de desenvolvimento de software capaz e motivada para desenvolver o projeto.

Em relação aos recursos materiais, a nível de hardware, é necessário 1 servidor, 4 máquinas para a equipa de desenvolvimento testar e desenvolver o sistema antes de o instalar e, por fim, uma máquina para o Mário utilizar na pastelaria. Já relativamente ao software, é necessário o Microsoft SQL Server, que é um sistema de gestão de base de dados para armazenar e gerir os dados, o *Microsoft .NET: ASP.NET Core*, especificamente as *frameworks Blazor, SignalR e Identity* para desenvolver e executar a aplicação em C#. Por fim, o *Visual Paradigm* foi utilizado para modelar os vários aspetos do sistema, através de diagramas.

## 1.6. Maquete do sistema

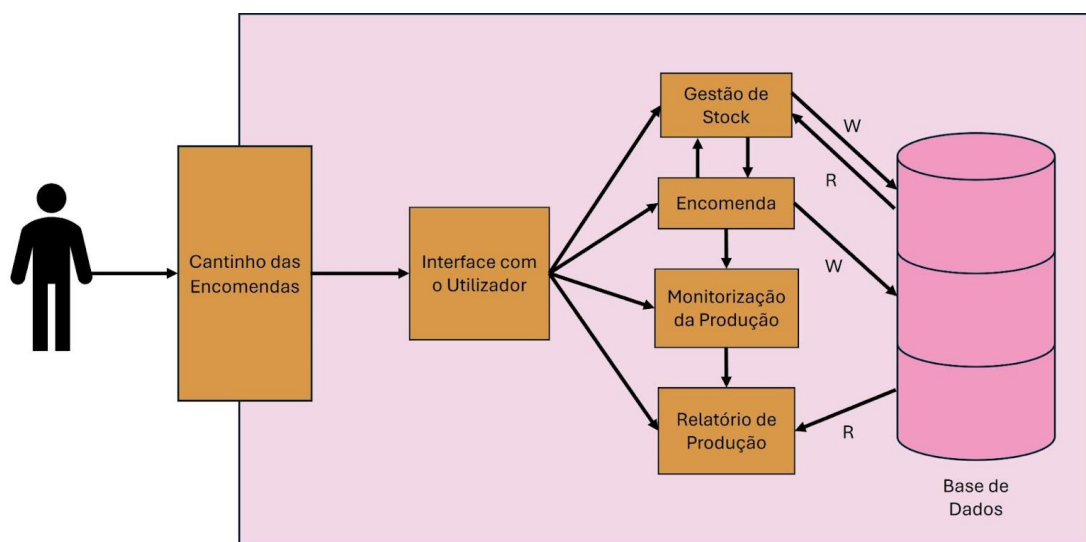


Figura 3 – Maquete do sistema

A maquete do sistema é representada pelos seguintes módulos:

- Interface com o utilizador
- Gestão de Stock
- Encomenda
- Monitorização da Produção
- Relatório de Produção

No software, o módulo de “Interface com o utilizador” gere o acesso à aplicação, de modo a garantir que diferentes utilizadores, de acordo com as suas autorizações, tenham acesso a diferentes funcionalidades do sistema.

A “Gestão de Stock” mantém atualizada a quantidade de material, neste caso de ingredientes, garantindo que existe stock suficiente para atender às encomendas e automatizando o reabastecimento de stock, quando necessário.

Do mesmo modo, no módulo da “Encomenda”, é possível coordenar a realização de pedidos, assim como definir a lista de espera para a produção destas mesmas encomendas.

Além disso, a “Monitorização da Produção” permite ao utilizador acompanhar, em tempo real, o estado da sua encomenda, podendo também visualizar o processo de confeção da encomenda. Finalmente, o módulo de “Relatório de Produção” fornece, no final da produção de cada encomenda, as estatísticas associadas ao tempo de fabrico e material utilizado.

## **1.7. Definição de um conjunto de medidas de sucesso**

Expansão da zona de venda → Aumentar a cobertura geográfica da venda de tartes a 100% do território nacional.

Capacidade de Atendimentos → Aumento do número de encomendas que podem ser atendidas num dia em 150%.

Eficiência da Linha → Proporção de tartes produzidas aumenta em 150%.

Qualidade do Produto → Melhoria na qualidade das tartes através do feedback dos clientes em 15%.

Tempo de Produção → Diminuição do tempo necessário para produzir cada tarte para metade.

Variedade de Produtos → Capacidade de introduzir novos sabores de tarte mais facilmente.

Controlo de Stock → Facilidade na gestão dos ingredientes, evitando desperdícios em 15% e garantir 95% de disponibilidades dos ingredientes em stock.

Plataforma Fácil e Acessível → Plataforma de fácil acesso para os clientes de forma a facilitar a encomenda de tartes em 50%, aumentando assim a taxa de encomendas realizadas com sucesso para 90% e utilizando um site intuitivo de modo a diminuir a taxa de abandono do processo da encomenda para menos de 5%.

Fidelidade do Cliente → devido à facilidade em realizar encomendas e à qualidade do produto, aumento de 70% da percentagem dos clientes que volta a realizar uma encomenda num prazo de 3 meses.

Tempo de entrega → redução em 30% do tempo médio de entrega das tartes após o pedido.

## **1.8. Plano de desenvolvimento**

Para organizar o desenvolvimento deste projeto, foi criado um diagrama de Gantt. Este diagrama ilustra com clareza as diversas tarefas a desenvolver e o seu cronograma. Além disso, permite uma melhor distribuição das tarefas pelo grupo de trabalho e é crucial para monitorar que o cronograma definido para cada etapa seja cumprido.

Este diagrama divide-se nos seis capítulos deste relatório, começando o mesmo pela definição do sistema, seguido da definição de requisitos e especificação do software. De seguida, procede-se à modelação do sistema de dados e à definição das interfaces do sistema, encerrando o relatório e trabalho a desenvolver nesta primeira fase com uma conclusão.

# Cantinho da Doçura

Laboratórios de Informática IV

LEI

Project day: **seg, 9/30/2024**

Display week: **1**

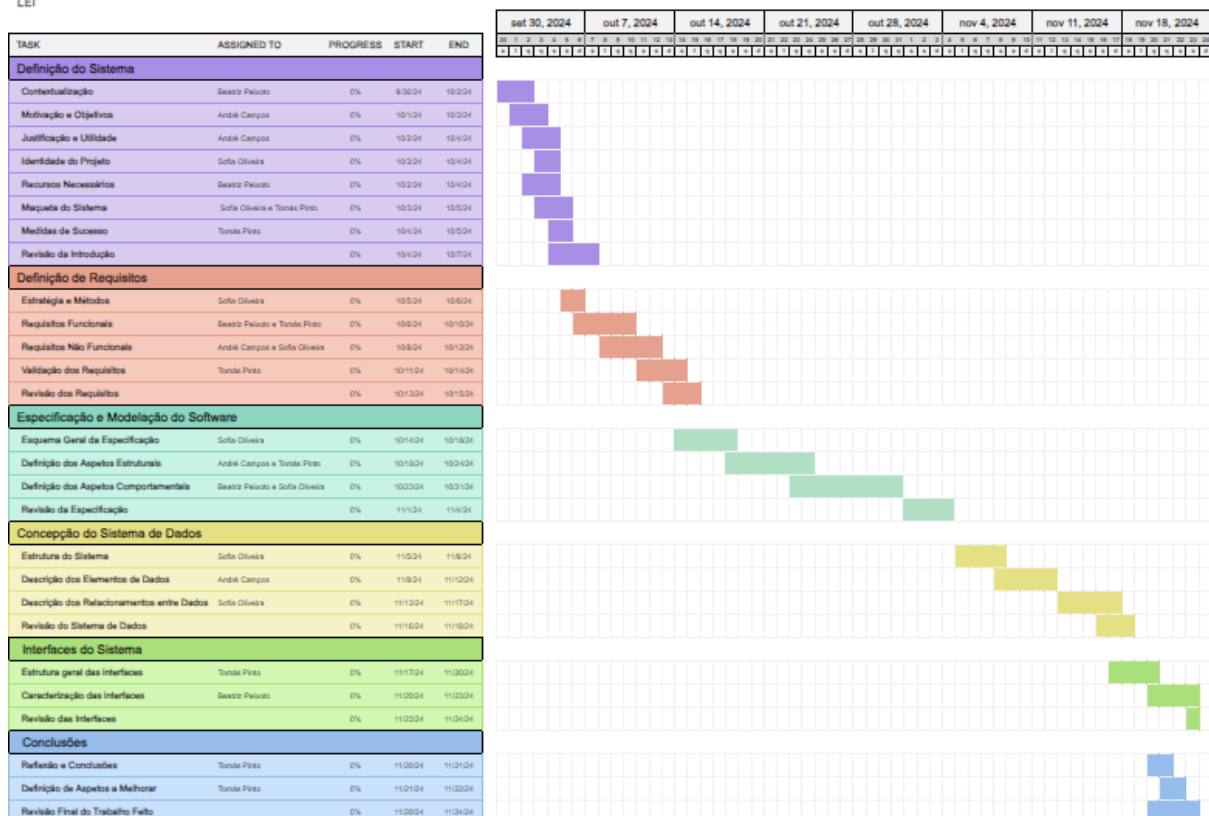


Figura 4 – Diagrama de Gantt

## 2. Levantamento e Análise de Requisitos

### 2.1. Apresentação da estratégia e método

Para a criação desta aplicação, a equipa de desenvolvimento de software necessitou de proceder ao levantamento de requisitos com os *stakeholders*, para poder garantir que as funcionalidades do programa atendessem aos objetivos previamente estabelecidos.

Com este propósito em mente, foram realizadas entrevistas com o dono da pastelaria, Mário, e funcionários da mesma, desde o atendimento ao cliente até à produção. Estas entrevistas apresentaram um carácter semiestruturado, permitindo que além de responder às perguntas preparadas previamente, o utilizador poderia também oferecer livremente a sua perspetiva sobre as funcionalidades relevantes do programa.

Além disso, alguns requisitos também foram levantados através da observação direta do funcionamento da linha de montagem e produção das tartes da pastelaria, com o principal intuito de identificar alguns motivos de atraso e falhas na produção.

Por fim, de modo a complementar estes requisitos e garantir que os utilizadores fiquem satisfeitos com o software produzido, foram realizados alguns questionários a clientes do “Cantinho da Doçura”, com o fim de compreender não só as funcionalidades desejadas pelos funcionários da pastelaria, mas também pelos clientes, garantindo uma melhor usabilidade e experiência de utilização do sistema. Estes questionários foram distribuídos por uma amostra ampla de clientes, através das redes sociais da pastelaria, abordando diferentes demográficas.

### 2.2. Descrição geral dos requisitos levantados

#### 2.2.1. Requisitos Funcionais

Para especificar os requisitos funcionais levantados, estes foram estruturados em tabelas divididas em oito secções: descrição da função, dados de entrada, dados de saída, ação a ser tomada, pré-condição, pós-condição e, caso existam, consequências. Segundo Sommerville (2011, pp. 98), alguns dos problemas da especificação de requisitos através de linguagem natural são removidos se optarmos pela utilização de especificações estruturadas, tal como nos exemplos que se seguem.

É importante denotar que o *pedFiltro* seleciona pedidos que têm certas características mencionadas pelo ator. Abaixo vai ser possível atentar às especificações dos diferentes requisitos levantados.

**Requisito:** O cliente deve poder registar-se no sistema (Tabela 1).



<b>Função</b>	Registrar cliente.
<b>Descrição</b>	O sistema deve permitir que o cliente se registre no sistema.
<b>Entrada</b>	Nome, email, número de telefone, morada e senha pretendida.
<b>Saída</b>	Confirmação se o cliente foi ou não registado com sucesso.
<b>Ação</b>	Cliente indica os seus dados e a senha que pretende utilizar. O sistema valida se os dados se encontram no formato correto e o email não se encontra na base de dados. Caso os dados sejam validados com sucesso, regista o cliente na base de dados do sistema.
<b>Pré-condição</b>	Nenhuma.
<b>Pós-condição</b>	Cliente fica registado na base de dados.
<b>Consequências</b>	Os dados introduzidos são inválidos ou o email introduzido não é único.

*Tabela 1*

**Requisito:** O utilizador deve poder fazer a autenticação para aceder ao sistema (Tabela 2).

<b>Função</b>	Autenticação no sistema.
<b>Descrição</b>	O sistema deve permitir que os seus utilizadores se autentiquem.
<b>Entrada</b>	Email e senha do utilizador.
<b>Saída</b>	Confirmação se o utilizador foi autenticado com sucesso ou não.
<b>Ação</b>	O utilizador insere as suas credenciais. O sistema valida as credenciais indicadas, o utilizador fica autenticado e tem acesso às funcionalidades disponíveis para ele.
<b>Pré-condição</b>	O utilizador está registado no sistema.
<b>Pós-condição</b>	O utilizador fica autenticado.
<b>Consequências</b>	Credenciais inválidas e o utilizador não consegue autenticar-se.

*Tabela 2*

**Requisito:** O utilizador deve poder terminar a sessão (Tabela 3).

<b>Função</b>	Permitir que o utilizador termine a sua sessão no sistema.
---------------	--

<b>Descrição</b>	O sistema deve permitir ao utilizador encerrar a sua sessão ativa.
<b>Entrada</b>	Ação do utilizador ao selecionar a opção "Terminar Sessão".
<b>Saída</b>	Confirmação visual de que a sessão foi terminada com sucesso.
<b>Ação</b>	O sistema encerra a sessão do utilizador.
<b>Pré-Condição</b>	O utilizador deve estar autenticado e com uma sessão ativa.
<b>Pós-Condição</b>	A sessão do utilizador é encerrada e ele deixa de ter acesso à sua conta.
<b>Consequências</b>	O utilizador não pode aceder à sua conta sem efetuar novamente o login.

*Tabela 3*

**Requisito:** O utilizador deve poder alterar a palavra-passe, em caso de esquecimento (Tabela 4).

<b>Função</b>	Permitir que o utilizador redefina a sua palavra-passe quando não a consegue recordar.
<b>Descrição</b>	O sistema deve disponibilizar um processo para o utilizador alterar a sua palavra-passe, caso tenha esquecido a atual. Este processo deve assegurar a identidade do utilizador e fornecer um método seguro para redefinição da palavra-passe.
<b>Entrada</b>	Identificação do utilizador (email) e confirmação de identidade (código enviado por email)
<b>Saída</b>	Notificação de sucesso o erro na alteração da palavra-passe.
<b>Ação</b>	O sistema envia um código de verificação para o email/SMS registado pelo utilizador. Após confirmação do código, o sistema permite ao utilizador definir uma nova palavra-passe, que será armazenada de forma segura.
<b>Pré-Condição</b>	O utilizador deve estar registado no sistema e ter acesso ao meio de contato associado.
<b>Pós-Condição</b>	A palavra-passe do utilizador é atualizada com a nova definida e pode ser utilizada no próximo login.
<b>Consequências</b>	Nenhuma.

*Tabela 4*

**Requisito:** O cliente deve poder visualizar o estado da encomenda, isto é, se está na lista de espera, em preparação, interrompida ou a ser enviada (Tabela 5).

<b>Função</b>	Permitir que o cliente acompanhe o estado atual da sua encomenda.
<b>Descrição</b>	O sistema deve apresentar ao cliente o estado da sua encomenda, indicando se esta está na lista de espera, em preparação, interrompida ou a ser enviada.
<b>Entrada</b>	Identificação da encomenda.
<b>Saída</b>	Visualização do estado atual da encomenda.
<b>Ação</b>	O sistema verifica o estado da encomenda na base de dados e exibe a informação correspondente ao cliente numa interface visual.
<b>Pré-Condição</b>	A encomenda deve estar registada no sistema e o cliente autenticado para visualizar o estado.
<b>Pós-Condição</b>	O cliente obtém informação atualizada sobre o estado da encomenda e pode monitorizar o progresso.
<b>Consequências</b>	Nenhuma.

*Tabela 5*

**Requisito:** O utilizador deve poder ver as tartes da pastelaria (Tabela 6).

<b>Função</b>	Ver tartes existentes.
<b>Descrição</b>	O sistema deve permitir que o utilizador visualize todas as tartes do sistema e as suas informações.
<b>Entrada</b>	Nenhuma.
<b>Saída</b>	Lista de tartes, caracterizadas por nome e preço.
<b>Ação</b>	O sistema consulta na base de dados as informações de todas as tartes e exibe o seu nome e preço, numa interface apelativa.
<b>Pré-condição</b>	Utilizador autenticado.
<b>Pós-condição</b>	As tartes existentes são exibidas no ecrã.
<b>Consequências</b>	Nenhuma.

Tabela 6

**Requisito:** O cliente deve poder visualizar o seu carrinho de compras (Tabela 7).

<b>Função</b>	Ver carrinho de compras.
<b>Descrição</b>	O sistema deve permitir que o cliente visualize os produtos no seu carrinho de compras, a sua quantidade e o preço.
<b>Entrada</b>	Nenhuma.
<b>Saída</b>	Lista de tartes no carrinho e as suas quantidades e preço do carrinho.
<b>Ação</b>	O sistema procura na base de dados os registos referentes ao carrinho de compras do cliente, calcula o preço e exhibe-os no ecrã.
<b>Pré-condição</b>	Cliente autenticado.
<b>Pós-condição</b>	Carrinho de compras exibido no ecrã.
<b>Consequências</b>	Nenhuma.

Tabela 7

**Requisito:** O cliente deve poder adicionar um produto ao carrinho de compras (Tabela 8).

<b>Função</b>	Adicionar produto ao carrinho de compras.
<b>Descrição</b>	O sistema deve permitir que o cliente adicione os produtos que desejar ao carrinho de compras.
<b>Entrada</b>	Identificação do produto.
<b>Saída</b>	Carrinho de compras atualizado, com o produto selecionado.
<b>Ação</b>	O cliente indica o produto a adicionar. O sistema adiciona o produto ao carrinho de compras e calcula o valor total da compra.
<b>Pré-condição</b>	Cliente autenticado e o produto disponível em stock.
<b>Pós-condição</b>	O produto fica adicionado ao carrinho de compras.
<b>Consequências</b>	Nenhuma.

Tabela 8

**Requisito:** O cliente deve poder remover o produto do carrinho de compras (Tabela 9).

<b>Função</b>	Remover produto do carrinho de compras.
<b>Descrição</b>	O sistema deve permitir que o cliente retire produtos previamente adicionados ao carrinho de compras, antes de finalizarem a encomenda.
<b>Entrada</b>	Identificação do produto.
<b>Saída</b>	Carrinho de compras atualizado, sem o produto selecionado e o valor total é recalculado.
<b>Ação</b>	O sistema retira o produto selecionado do carrinho de compras, recalcula o valor total do carrinho e apresenta o carrinho de compras atualizado ao cliente.
<b>Pré-Condição</b>	Cliente autenticado e carrinho de compras possui produtos.
<b>Pós-Condição</b>	Carrinho de compras não possui o produto selecionado.
<b>Consequências</b>	Nenhuma.

*Tabela 9*

**Requisito:** O cliente deve poder pagar pela encomenda no seu carrinho de compras (Tabela 10).

<b>Função</b>	Pagar pela encomenda no carrinho de compras.
<b>Descrição</b>	O sistema deve permitir que o cliente efetue o pagamento da encomenda no seu carrinho de compras através de um dos métodos de pagamento disponíveis, finalizando assim a encomenda.
<b>Entrada</b>	Seleção do método de pagamento e informações de pagamento.
<b>Saída</b>	Confirmação do pagamento e recibo com o valor pago e os produtos encomendados.
<b>Ação</b>	O sistema processa o pagamento de acordo com o método escolhido. Se for possível processar o pagamento, a encomenda é colocada na lista de espera.
<b>Pré-Condição</b>	Cliente autenticado e carrinho de compras possui produtos.
<b>Pós-Condição</b>	Encomenda entra na lista de espera da produção.

<b>Consequências</b>	Nenhuma.
----------------------	----------

*Tabela 10*

**Requisito:** O cliente deve poder cancelar a encomenda depois de enviado e pago, se o mesmo ainda não tiver sido produzido (Tabela 11).

<b>Função</b>	Cancelar a encomenda.
<b>Descrição</b>	O sistema deve permitir que o cliente cancele uma encomenda que já foi paga, desde que esta ainda se encontre na lista de espera de produção.
<b>Entrada</b>	Identificação da encomenda.
<b>Saída</b>	Reembolso ao cliente.
<b>Ação</b>	O sistema verifica o estado da encomenda. Se esta ainda se encontrar “em espera”, procede-se ao cancelamento da mesma, retirando-a da lista de espera e processando o reembolso do cliente. Caso contrário, não é possível cancelar a encomenda.
<b>Pré-Condição</b>	Cliente autenticado no sistema e encomenda paga.
<b>Pós-Condição</b>	Encomenda cancelada e pagamento reembolsado ao cliente.
<b>Consequências</b>	Encomenda já se encontra em produção. Não é possível efetuar o seu cancelamento.

*Tabela 11*

**Requisito:** O cliente deve poder ver as suas encomendas usando um filtro (Tabela 12).

<b>Função</b>	O cliente vê as suas encomendas usando um filtro
<b>Descrição</b>	O cliente consegue ver quais as suas encomendas, usando um filtro especificado pelo mesmo
<b>Entrada</b>	ID do cliente e filtro
<b>Saída</b>	Lista com as encomendas do cliente filtradas pelo filtro definido pelo ator
<b>Ação</b>	O cliente indica que pretende ver as suas encomendas e indica qual filtro pretende usar (o filtro pode ser inexistente). O sistema procura as encomendas desse cliente e apresenta a lista de encomendas ordenada da mais recente para a mais antiga

<b>Pré-Condição</b>	Cliente deve estar registrado no sistema e deve ter a sessão iniciada
<b>Pós-Condição</b>	O cliente visualiza as suas encomendas
<b>Consequências</b>	Nenhuma

*Tabela 12*

**Requisito:** O cliente deve poder avaliar e comentar o desempenho da pastelaria (Tabela 13).

<b>Função</b>	Permitir que o cliente faça uma avaliação e deixe comentários sobre o desempenho da pastelaria.
<b>Descrição</b>	O sistema deve permitir ao cliente deixar uma avaliação, com uma classificação e comentários, sobre a sua experiência com a pastelaria e as encomendas realizadas.
<b>Entrada</b>	Classificação (escala de 1 a 5 estrelas) e comentário textual fornecidos pelo cliente.
<b>Saída</b>	Mensagem de confirmação de que a avaliação foi submetida com sucesso.
<b>Ação</b>	O sistema recebe a avaliação e o comentário do cliente e armazena as informações na base de dados.
<b>Pré-Condição</b>	O cliente deve estar autenticado e ter uma encomenda previamente concluída.
<b>Pós-Condição</b>	A avaliação e o comentário do cliente são armazenados e podem ser visualizados.
<b>Consequências</b>	Comentários negativos ou inapropriados podem impactar a reputação da pastelaria.

*Tabela 13*

**Requisito:** O utilizador deve poder visualizar as avaliações da pastelaria (Tabela 14).

<b>Função</b>	Visualizar avaliações da pastelaria.
<b>Descrição</b>	O sistema deve permitir que o utilizador, cliente ou funcionário, visualize as avaliações feitas pelos clientes sobre a pastelaria.
<b>Entrada</b>	Nenhuma.

<b>Saída</b>	Lista de avaliações, que incluem classificações e comentários.
<b>Ação</b>	O sistema procura na base de dados os registos das avaliações feitas pelos clientes e exibe-as no ecrã.
<b>Pré-Condição</b>	Utilizador autenticado.
<b>Pós-Condição</b>	Avaliações exibidas.
<b>Consequências</b>	Nenhuma.

*Tabela 14*

**Requisito:** O cliente deve poder adicionar produtos aos seus “favoritos” (Tabela 15).

<b>Função</b>	Adicionar produtos à lista de favoritos.
<b>Descrição</b>	O sistema deve permitir que o cliente adicione o produto que desejar à sua lista de favoritos.
<b>Entrada</b>	Identificação do produto.
<b>Saída</b>	Lista de favoritos atualizada, com o produto selecionado.
<b>Ação</b>	O cliente indica o produto a adicionar. O sistema adiciona o produto selecionado à lista de favoritos.
<b>Pré-Condição</b>	O cliente deve estar autenticado.
<b>Pós-Condição</b>	O produto fica na lista de favoritos.
<b>Consequências</b>	Nenhuma.

*Tabela 15*

**Requisito:** O cliente deve poder remover produtos dos seus “favoritos” (Tabela 16).

<b>Função</b>	Permitir que o cliente remova produtos da lista de “favoritos”.
<b>Descrição</b>	O sistema deve possibilitar que o cliente retire produtos previamente adicionados à sua lista de “favoritos”, ajudando-o a manter a lista atualizada com os itens de seu interesse.
<b>Entrada</b>	Identificação do produto que o cliente deseja remover da lista de favoritos.
<b>Saída</b>	Lista de favoritos atualizada, sem o produto selecionado.



<b>Ação</b>	O sistema remove o produto selecionado da lista de favoritos do cliente e exibe a lista atualizada.
<b>Pré-Condição</b>	Cliente autenticado e lista de favoritos contém produtos.
<b>Pós-Condição</b>	O produto selecionado deixa de estar na lista de favoritos do cliente.
<b>Consequências</b>	Nenhuma.

*Tabela 16*

**Requisito:** O cliente deve poder ver os seus “favoritos” (Tabela 17).

<b>Função</b>	Ver a lista de favoritos.
<b>Descrição</b>	O cliente deve poder visualizar a lista dos produtos que adicionou à lista de favoritos.
<b>Entrada</b>	Nenhuma.
<b>Saída</b>	Lista de produtos favoritos.
<b>Ação</b>	O cliente indica que pretende ver a sua lista de favoritos. O sistema apresenta uma lista com os produtos favoritos do cliente.
<b>Pré-Condição</b>	Cliente autenticado.
<b>Pós-Condição</b>	O cliente visualiza a sua lista de favoritos.
<b>Consequências</b>	Nenhuma.

*Tabela 17*

**Requisito:** O cliente deve poder indicar que pretende receber uma notificação quando for possível efetuar a sua encomenda, caso este tente efetuar uma encomenda que, por falta de stock, não pode ser produzida (Tabela 18).

<b>Função</b>	Permitir que o cliente solicite uma notificação quando o stock for repostado, tornando possível efetuar uma encomenda.
<b>Descrição</b>	O sistema deve permitir que o cliente indique que deseja receber uma notificação, via email, caso tente encomendar um produto que está fora de stock. Assim, o cliente será avisado quando o produto estiver disponível para encomenda.

<b>Entrada</b>	Identificação do produto fora de stock para o qual o cliente deseja receber uma notificação.
<b>Saída</b>	Confirmação de que a solicitação de notificação foi registada com sucesso.
<b>Ação</b>	O sistema regista o pedido do cliente para ser notificado sobre a disponibilidade do produto, monitoriza o stock e, ao repor o item, envia um email a notificar o cliente.
<b>Pré-Condção</b>	O produto desejado está fora de stock e o cliente autenticado indica interesse na notificação.
<b>Pós-Condção</b>	O cliente é notificado assim que o produto estiver disponível para encomenda.
<b>Consequências</b>	Nenhuma.

*Tabela 18*

**Requisito:** Nos dados do cliente, este deve poder modificar o seu nome, email, morada e número de telefone (Tabela 19).

<b>Função</b>	Modificar dados do cliente.
<b>Descrição</b>	O sistema deve permitir que o cliente modifique as informações do cliente, incluindo nome, email, morada e número de telefone.
<b>Entrada</b>	Identificação do campo a atualizar e novo valor para esse campo.
<b>Saída</b>	Exibição das informações atualizadas do cliente.
<b>Ação</b>	O sistema recebe os dados a alterar do cliente, valida-os verificando se se encontram no formato correto, no caso de alteração de email, que este ainda não se encontra na base de dados. Caso os dados estejam corretos, altera o registo do cliente na base de dados.
<b>Pré-Condção</b>	Cliente autenticado.
<b>Pós-Condção</b>	As informações do cliente encontram-se atualizadas.
<b>Consequências</b>	Os dados inseridos não são válidos.

*Tabela 19*

**Requisito:** O cliente deve poder ver a aba de “Perguntas Frequentes” do sistema (Tabela 20).

<b>Função</b>	Ver a aba de Perguntas Frequentes.
<b>Descrição</b>	O sistema deve permitir que o cliente aceda às Perguntas Frequentes para poder tirar algumas dúvidas.
<b>Entrada</b>	Nenhuma.
<b>Saída</b>	Lista com todas as perguntas frequentes do sistema.
<b>Ação</b>	O cliente indica que pretende ver as “Perguntas Frequentes”. O sistema apresenta uma lista com as perguntas frequentes.
<b>Pré-Condição</b>	Cliente autenticado.
<b>Pós-Condição</b>	O cliente consegue aceder às perguntas frequentes.
<b>Consequências</b>	Nenhuma.

*Tabela 20*

**Requisito:** O cliente deve poder visualizar graficamente as diversas fases da preparação da sua encomenda (Tabela 21).

<b>Função</b>	Permitir que o cliente acompanhe visualmente as várias fases da preparação da sua encomenda.
<b>Descrição</b>	O sistema deve exibir de forma gráfica o progresso da encomenda, apresentando as diferentes fases da sua preparação.
<b>Entrada</b>	Identificação da encomenda para a qual o cliente deseja visualizar o progresso.
<b>Saída</b>	Representação gráfica do progresso da encomenda nas diferentes fases de preparação.
<b>Ação</b>	O sistema obtém o estado atual da encomenda e apresenta um gráfico que reflete as fases de preparação concluídas e a fase atual.
<b>Pré-Condição</b>	O cliente deve estar autenticado e ter uma encomenda em andamento ou recentemente concluída.
<b>Pós-Condição</b>	O cliente visualiza uma representação gráfica atualizada das fases de preparação da sua encomenda.

<b>Consequências</b>	Nenhuma.
----------------------	----------

*Tabela 21*

**Requisito:** O funcionário deve poder ver as informações de uma determinada encomenda (Tabela 22).

<b>Função</b>	Visualizar informações sobre uma encomenda.
<b>Descrição</b>	O sistema deve permitir que o funcionário veja as informações referentes a uma determinada encomenda.
<b>Entrada</b>	Identificador da encomenda.
<b>Saída</b>	Identificador da encomenda, cliente associado, estado da encomenda, tartes e quantidades encomendadas e, se existir, relatório da encomenda.
<b>Ação</b>	O sistema recebe o identificador da encomenda, procura os registos a esta associados na base de dados e exibe-os no ecrã.
<b>Pré-Condição</b>	Funcionário autenticado.
<b>Pós-Condição</b>	Informações da encomenda exibidas.
<b>Consequências</b>	A encomenda não existe.

*Tabela 22*

**Requisito:** O funcionário deve poder ver as encomendas que estão a ser preparadas, usando um filtro (Tabela 23).

<b>Função</b>	Apresentar as encomendas que estão a ser preparadas.
<b>Descrição</b>	O funcionário vê quais as encomendas que estão a ser preparadas.
<b>Entrada</b>	Filtro para encomendas.
<b>Saída</b>	Lista de encomendas filtradas que estão a ser preparadas.
<b>Ação</b>	O funcionário indica que pretende ver as encomendas que estão a ser preparadas e indica o filtro que pretende usar (filtro pode ser inexistente). O sistema procura as encomendas em preparação usando o filtro indicado e apresenta ao funcionário as encomendas que estão a ser preparadas ordenadas da mais antiga para a mais recente.

<b>Pré-Condição</b>	Funcionário tem sessão iniciada.
<b>Pós-Condição</b>	Funcionário visualiza as encomendas que estão a ser preparadas.
<b>Consequências</b>	Nenhuma.

*Tabela 23*

**Requisito:** O funcionário deve poder ver as encomendas que já foram preparadas, usando um filtro (Tabela 24).

<b>Função</b>	Apresentar as encomendas feitas
<b>Descrição</b>	O funcionário vê quais as encomendas já preparadas
<b>Entrada</b>	Filtro para encomendas
<b>Saída</b>	Lista de encomendas filtradas que já foram preparadas
<b>Ação</b>	O funcionário indica que pretende ver as encomendas que já foram preparadas e indica um filtro que pretende utilizar (filtro pode ser inexistente). O sistema procura as encomendas no sistema e apresenta ao funcionário as encomendas filtradas que já foram preparadas ordenadas da mais recente para a mais antiga
<b>Pré-Condição</b>	Funcionário tem a sessão iniciada
<b>Pós-Condição</b>	Funcionário visualiza as encomendas que foram feitas
<b>Consequências</b>	Nenhuma

*Tabela 24*

**Requisito:** Funcionário deve poder ver as encomendas na fila de espera, usando um filtro (Tabela 25).

<b>Função</b>	Apresentar as encomendas na fila de espera
<b>Descrição</b>	O funcionário vê quais as encomendas que estão em fila de espera
<b>Entrada</b>	Filtro para encomendas
<b>Saída</b>	Lista de encomendas filtradas que estão em fila de espera

<b>Ação</b>	O funcionário indica que pretende ver as encomendas que estão na fila de espera e indica um filtro. O sistema procura as encomendas e apresenta ao funcionário as encomendas filtradas que estão na fila de espera
<b>Pré-Condição</b>	Funcionário tem sessão iniciada
<b>Pós-Condição</b>	Funcionário visualiza as encomendas que estão na fila de espera
<b>Consequências</b>	Nenhuma

*Tabela 25*

**Requisito:** O funcionário deve poder ver o stock atual de todos os materiais (Tabela 26).

<b>Função</b>	Visualizar o stock atual de todos os materiais
<b>Descrição</b>	O funcionário vê o stock atual de todos os materiais
<b>Entrada</b>	Nenhum
<b>Saída</b>	Lista com todos os itens presentes no stock juntamente com a sua quantidade no mesmo
<b>Ação</b>	O funcionário indica que pretende ver o stock atual de todos os materiais. O sistema apresenta ao funcionário uma lista contendo todos os materiais atuais e a quantidade existente desse material
<b>Pré-Condição</b>	Funcionário tem a sessão iniciada
<b>Pós-Condição</b>	Funcionário visualiza o stock atual de todos os materiais
<b>Consequências</b>	Nenhuma

*Tabela 26*

**Requisito:** O funcionário deve poder ver o relatório de produção de uma determinada encomenda (Tabela 27).

<b>Função</b>	Ver relatório de produção de uma determinada encomenda.
<b>Descrição</b>	O sistema deve permitir que o funcionário veja o relatório no final da produção de cada encomenda.
<b>Entrada</b>	Identificação da encomenda.

<b>Saída</b>	Relatório de produção da encomenda.
<b>Ação</b>	O cliente indica que pretende visualizar o relatório de produção de uma dada encomenda. O sistema apresenta o relatório de produção da encomenda indicada.
<b>Pré-Condição</b>	Funcionário autenticado.
<b>Pós-Condição</b>	O funcionário tem acesso ao relatório de produção da encomenda.
<b>Consequências</b>	Nenhuma

*Tabela 27*

**Requisito:** O funcionário deve poder interromper a produção de encomendas (Tabela 28).

<b>Função</b>	Interromper a produção.
<b>Descrição</b>	O sistema deve permitir que o funcionário interrompa temporariamente a produção de encomendas.
<b>Entrada</b>	Comando de interrupção da produção.
<b>Saída</b>	Atualização dos estados das encomendas em produção.
<b>Ação</b>	O sistema processa o comando do funcionário, verifica se existem encomendas em produção e, caso existam, altera o seu estado para “interrompida”, colocando-as na frente da lista de espera.
<b>Pré-Condição</b>	Funcionário autenticado e produção em funcionamento.
<b>Pós-Condição</b>	Encomendas previamente em produção têm o seu estado alterado para “interrompida”.
<b>Consequências</b>	Nenhuma.

*Tabela 28*

**Requisito:** O funcionário deve poder retomar a produção de encomendas (Tabela 29).

<b>Função</b>	Retomar a produção.
<b>Descrição</b>	O sistema deve permitir que o funcionário retome a produção de encomendas.

<b>Entrada</b>	Comando de retomada.
<b>Saída</b>	Atualização dos estados das encomendas de estado “interrompida” para o estado “em produção”.
<b>Ação</b>	O sistema processa o comando do funcionário, verifica se existem encomendas de estado “interrompida” e, caso existam, altera o seu estado para “em produção”.
<b>Pré-Condição</b>	Funcionário autenticado e produção interrompida.
<b>Pós-Condição</b>	Encomendas previamente de estado “interrompida” têm o seu estado alterado para “em produção”.
<b>Consequências</b>	Nenhuma.

*Tabela 29*

**Requisito:** Funcionário deve poder atualizar o estado de uma encomenda (Tabela 30).

<b>Função</b>	Atualizar o estado de uma encomenda.
<b>Descrição</b>	O funcionário atualiza qual o estado de uma encomenda selecionada pelo mesmo.
<b>Entrada</b>	Encomenda e novo estado.
<b>Saída</b>	Encomenda com o novo estado.
<b>Ação</b>	Funcionário indica que pretende atualizar o estado de uma encomenda. O sistema apresenta ao funcionário todas as encomendas. Funcionário seleciona a encomenda que pretende alterar, indica qual o novo estado dessa encomenda e o sistema atualiza o estado da encomenda selecionada pelo estado selecionado pelo funcionário.
<b>Pré-Condição</b>	Funcionário tem a sessão iniciada e existem encomendas em preparação.
<b>Pós-Condição</b>	A encomenda selecionada pelo funcionário tem um novo estado, selecionado pelo mesmo.
<b>Consequências</b>	Encomenda selecionada pode não existir ou pode não estar nas encomendas em preparação. Estado indicado pode não ser válido.

*Tabela 30*



**Requisito:** O funcionário deve poder atualizar o progresso da preparação da encomenda, fazendo com que o programa calcule as métricas da fase terminada (Tabela 31).

<b>Função</b>	Atualizar o progresso de uma tarte de uma encomenda.
<b>Descrição</b>	O funcionário atualiza qual o progresso de uma tarte de uma encomenda selecionada pelo mesmo. O sistema guarda métricas referentes à fase anterior, como tempo de produção, possíveis falhas e material gasto, armazenadas no relatório.
<b>Entrada</b>	Encomenda, a tarte da mesma e o seu novo progresso.
<b>Saída</b>	Encomenda com um novo progresso e novas métricas da fase anterior adicionadas ao relatório de produção.
<b>Ação</b>	Funcionário indica que pretende atualizar o progresso de uma encomenda. O sistema apresenta ao funcionário todas as encomendas que estão em preparação. Funcionário seleciona a encomenda que pretende alterar, indica em qual tarte pretende alterar o progresso e indica a nova fase de produção dessa tarte. O sistema atualiza o progresso da tarte selecionada pela fase selecionada pelo funcionário. De seguida, calcula o tempo que a encomenda passou na fase anterior, verifica se ocorreram interrupções na produção e calcula o material gasto nessa fase. O sistema coleta essas métricas na encomenda de produção, associando-as, no relatório, à fase de produção a que pertencem.
<b>Pré-Condição</b>	Funcionário tem a sessão iniciada e existem encomendas em preparação.
<b>Pós-Condição</b>	A tarte da encomenda selecionada pelo funcionário encontra-se numa nova fase de produção, selecionada pelo mesmo, e o relatório de produção da encomenda tem novas métricas associadas à fase anterior.
<b>Consequências</b>	Encomenda selecionada pode não existir ou pode não estar nas encomendas em preparação. Tarte selecionada pode não existir. Fase indicada pode não ser válida para aquela encomenda.

*Tabela 31*

**Requisito:** O funcionário deve poder, a qualquer momento, modificar a capacidade máxima de stock de qualquer material (Tabela 32).

<b>Função</b>	Modificar capacidade máxima de stock.
---------------	---------------------------------------

<b>Descrição</b>	O sistema deve permitir que o funcionário, a qualquer momento, altere a capacidade máxima de stock de qualquer material utilizado na produção das encomendas.
<b>Entrada</b>	Identificação do material e o valor correspondente à sua nova capacidade máxima.
<b>Saída</b>	Exibição do novo valor de capacidade máxima do material.
<b>Ação</b>	O sistema recebe a identificação do material, valida na base de dados que este é um material utilizado na produção, recebe o seu novo valor, valida que este mesmo valor corresponde a um número positivo e igual ou superior ao stock atual, e atualiza esse valor na base de dados do sistema.
<b>Pré-Condição</b>	Funcionário autenticado.
<b>Pós-Condição</b>	Capacidade máxima do stock do material é atualizada.
<b>Consequências</b>	O material indicado não existe.

*Tabela 32*

**Requisito:** O sistema deve notificar o funcionário quando a capacidade de stock de um qualquer material se encontrar a 10% da sua capacidade (Tabela 33).

<b>Função</b>	Notificar baixo stock de material.
<b>Descrição</b>	O sistema deve enviar um email a notificar o funcionário quando o stock atual de um qualquer material se encontrar abaixo de 10% da sua capacidade máxima.
<b>Entrada</b>	Nenhuma (O sistema monitoriza, em tempo real, o stock, de forma autónoma).
<b>Saída</b>	Notificação, via email, que indica o material cujo stock se encontra a 10% de capacidade.
<b>Ação</b>	O sistema monitoriza automaticamente o stock dos materiais em tempo real. Quando o stock de um qualquer material se encontra abaixo de 10% da sua capacidade máxima, envia um email de notificação ao funcionário.
<b>Pré-Condição</b>	Nenhuma.
<b>Pós-Condição</b>	O funcionário é notificado sobre os baixos níveis de stock.

<b>Consequências</b>	Nenhuma.
----------------------	----------

*Tabela 33*

**Requisito:** O funcionário deve poder, a qualquer momento, modificar a quantidade de um material do stock (Tabela 34).

<b>Função</b>	Modificar a quantidade de um material do stock.
<b>Descrição</b>	O sistema deve permitir que o funcionário, a qualquer momento, altere a quantidade de stock de qualquer material.
<b>Entrada</b>	Identificação do material e o valor correspondente à sua nova quantidade.
<b>Saída</b>	Exibição do novo valor de quantidade do material.
<b>Ação</b>	O sistema recebe a identificação do material, valida na base de dados que este é um material utilizado na produção, recebe o seu novo valor, valida que este mesmo valor corresponde a um número positivo e atualiza esse valor na base de dados do sistema.
<b>Pré-Condição</b>	Funcionário autenticado.
<b>Pós-Condição</b>	Quantidade de stock do material é atualizada.
<b>Consequências</b>	O material indicado não existe.

*Tabela 34*

**Requisito:** O funcionário deve poder, a qualquer momento, adicionar um novo material ao stock (Tabela 35).

<b>Função</b>	Adicionar um novo material ao stock
<b>Descrição</b>	O sistema deve permitir que o funcionário, a qualquer momento, adicione um novo material ao stock.
<b>Entrada</b>	Nome do material, quantidade inicial do material e capacidade máxima do material.
<b>Saída</b>	Exibição de confirmação com as informações do material.
<b>Ação</b>	O sistema recebe as informações do novo material, valida que o nome é único, a quantidade inicial e capacidade são números positivos e a

	capacidade é igual ou maior à quantidade inicial. De seguida, regista o novo material na base de dados.
<b>Pré-Condição</b>	Funcionário autenticado.
<b>Pós-Condição</b>	Novo material registado na base de dados do sistema.
<b>Consequências</b>	Já existe um material com esse nome ou os valores de capacidade e quantidade estão inválidos.

*Tabela 35*

**Requisito:** O funcionário deve poder remover um material do stock (Tabela 36).

<b>Função</b>	Remover material do stock.
<b>Descrição</b>	O sistema deve permitir que o funcionário remova um material do stock, desde que este não esteja associado a uma tarte.
<b>Entrada</b>	Identificação do material.
<b>Saída</b>	Exibição de confirmação da remoção.
<b>Ação</b>	O sistema recebe a identificação do material, valida na base de dados que este é um material utilizado na produção e verifica que este não esteja associado a nenhuma tarte. Se essas condições se verificarem, remove o registo do material da base de dados.
<b>Pré-Condição</b>	Funcionário autenticado.
<b>Pós-Condição</b>	Material removido da base de dados.
<b>Consequências</b>	O material indicado não existe ou encontra-se associado a uma tarte.

*Tabela 36*

**Requisito:** O funcionário deve poder, a qualquer momento, adicionar uma nova tarte no sistema (Tabela 37).

<b>Função</b>	Adicionar nova tarte.
<b>Descrição</b>	O sistema deve permitir que o funcionário adicione uma nova tarte no sistema, especificando os detalhes necessários.

<b>Entrada</b>	Nome, descrição, preço, materiais utilizados e a sua quantidade, assim como nome, descrição e imagem das etapas de preparação.
<b>Saída</b>	Confirmação de tarte adicionada com sucesso.
<b>Ação</b>	O sistema recebe os dados da nova tarte, valida que os campos se encontram preenchidos corretamente e se os materiais associados existem na base de dados. Se as condições se verificarem, regista a tarte na base de dados do sistema.
<b>Pré-Condição</b>	Funcionário autenticado.
<b>Pós-Condição</b>	As informações da tarte são adicionadas na base de dados.
<b>Consequências</b>	Algum material associado à tarte não existe.

*Tabela 37*

**Requisito:** O funcionário deve poder remover uma tarte do sistema (Tabela 38).

<b>Função</b>	Remover uma tarte.
<b>Descrição</b>	O sistema deve permitir que o funcionário remova uma tarte do sistema, desde que esta não esteja associada a encomendas “Em Preparação” ou “Em Espera”.
<b>Entrada</b>	Identificação da tarte.
<b>Saída</b>	Confirmação de remoção da tarte.
<b>Ação</b>	O sistema recebe o identificador da tarte, valida se existe no sistema e verifica se não está associada a encomendas “Em Espera” ou “Em Preparação”. Se estas condições se verificam, desassocia os materiais utilizados da tarte e remove o registo da mesma da base de dados.
<b>Pré-Condição</b>	Funcionário autenticado.
<b>Pós-Condição</b>	Tarte removida do sistema.
<b>Consequências</b>	Tarte pertence a encomendas “Em Espera” ou “Em Preparação”.

*Tabela 38*

### **2.2.2. Requisitos Não Funcionais**

- A produção dos pedidos só é realizada entre as 06:00 e as 18:30 todos os dias da semana, exceto às segundas-feiras. Fora deste horário, o sistema não permite o início de preparação de pedidos.
- O sistema deve permitir que sejam feitas encomendas a qualquer momento, incluindo fora do horário de produção. Estes pedidos são colocados na fila de espera e serão atendidos quando for retomada a produção.
- O sistema deve armazenar e tratar os dados dos clientes de acordo com o devido regulamento de proteção de dados atualmente em vigor.
- Deve ser possível adaptar facilmente o sistema a novas funcionalidades e receitas.
- No sistema, apenas funcionários autenticados devem ter acesso a funcionalidades do sistema exclusivamente para funcionários.
- Os funcionários devem conseguir manipular com facilidade todas as funcionalidades do sistema após 40 minutos de aprendizagem.
- Cada nova encomenda deve ser recebida e colocada na lista de espera em menos de 3 segundos.

### **2.3. Validação dos requisitos estabelecidos**

Foi necessário proceder à validação dos requisitos, de forma a garantir que as necessidades dos utilizadores foram retratadas corretamente nos mesmos. Nesse sentido, foram realizadas reuniões com o Mário, de modo a fazer a revisão destes requisitos, garantindo que estes, além de atenderem às necessidades expostas, eram consistentes, completos e claros.

Uma sessão de validação final resultou na aprovação dos requisitos estabelecidos, permitindo à equipa da *Razen* avançar para a próxima etapa de desenvolvimento de software.

### 3. Especificação e Modelação do Software

### 3.1. Apresentação geral da especificação

Este capítulo apresenta a especificação estrutural e comportamental do sistema, com base nos requisitos definidos previamente.

Para modelar os aspetos estruturais do sistema, foram elaborados um modelo de domínio e um diagrama de componentes.

O modelo de domínio permite entender e explicar o problema que a nossa aplicação terá de resolver, baseado nos conceitos da pastelaria e nos requisitos propostos pelo Mário.

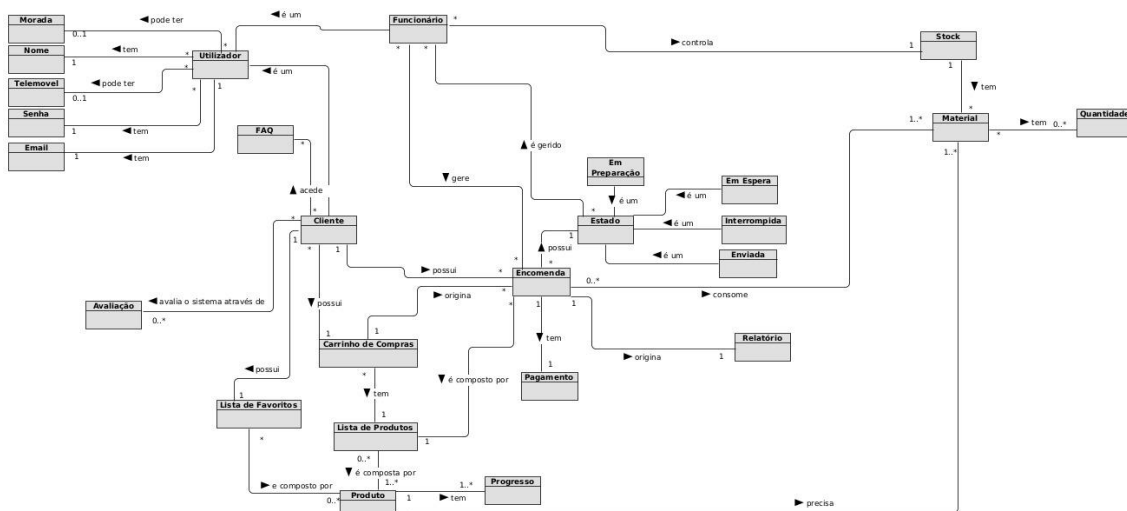
O diagrama de componentes permite identificar os principais componentes do sistema, assim como as dependências e interações entre eles. Deste modo, é facilitado o processo de divisão do sistema em vários módulos.

De seguida, para modelar os aspetos comportamentais, foi feito um diagrama de atividade, que representa o fluxo das ações executadas pelo sistema. Deste modo, é possível perceber melhor o funcionamento do sistema, assim como detalhar o trajeto de ações que definem as funcionalidades do mesmo.

Assim sendo, esta especificação permite um melhor estudo da estrutura do sistema a implementar, permitindo também uma melhor descrição das funcionalidades do mesmo, o que auxilia o desenvolvimento e implementação do sistema.

### 3.2. Aspetos estruturais

### 3.2.1. Modelo de Domínio



*Figura 5 – Modelo de Domínio*

Após o levantamento de requisitos, o modelo de domínio (Figura 5) foi criado com objetivo de representar, de forma clara, os conceitos e regras que a aplicação terá. Desta forma, identificamos quais as principais entidades, conceitos e as relações entre eles.

O modelo de domínio não indica como a aplicação irá funcionar, mas sim os conceitos que a aplicação terá de levar em consideração, com base nos requisitos do cliente. Ao analisar este modelo, podemos dividi-lo em 6 partes: utilizadores, cliente, encomenda, funcionário, stock e produto. Neste seguimento, podemos visualizar estas subsecções de forma individual nos próximos subcapítulos.

### 3.2.1.1. Utilizadores

Um utilizador (Figura 6) tem:

- Senha (obrigatório)
- Email (obrigatório e único para cada utilizador)
- Nome (obrigatório)
- Telemóvel (opcional)
- Morada (opcional)

Um Utilizador pode ser um funcionário ou um cliente.

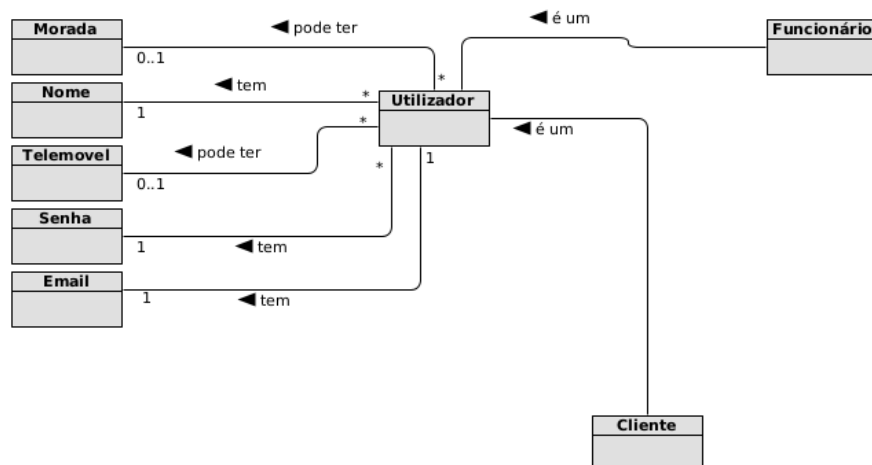


Figura 6 – Modelo de Domínio (Utilizadores)

### 3.2.1.2. Cliente

O cliente:

- É um utilizador.
- Consegue avaliar a pastelaria.
- Consegue aceder às perguntas frequentes (FAQ).
- Tem encomendas.
- Tem uma lista de favoritos, que é composta por produtos.



- Possui um carrinho de compras que tem uma lista de produtos, a qual é constituída por produtos que o cliente deseja comprar. Futuramente, o carrinho de compras originará uma encomenda.

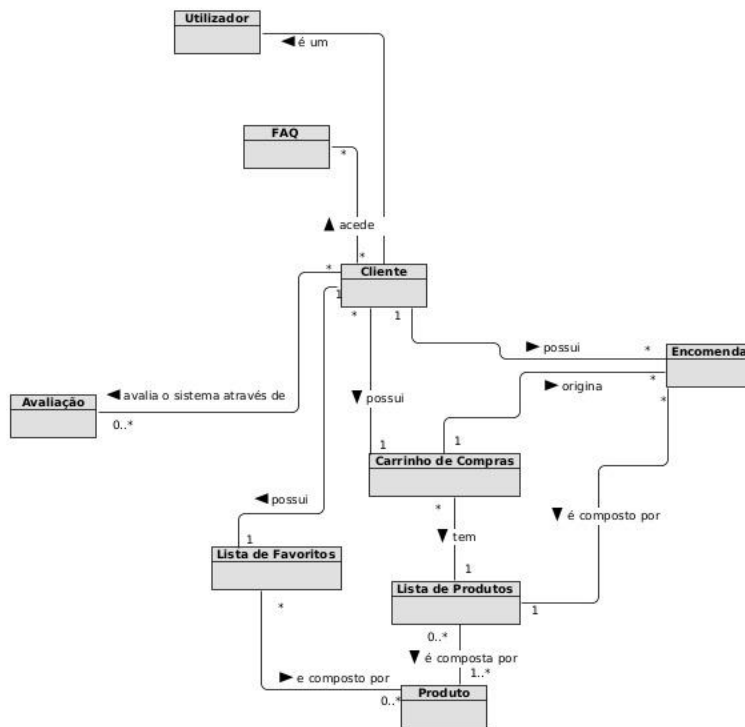


Figura 7 – Modelo de Domínio (Cliente)

### 3.2.1.3. Encomenda

Uma Encomenda:

- Pertence a um cliente.
- É originada de um carrinho de compras.
- Tem de um pagamento.
- Origina um relatório.
- Possui um estado.
- É gerida pelos funcionários.
- Possui uma lista de produtos.
- Consome material.

Uma encomenda é composta por produtos, que pertencem à lista de produtos.

O estado da encomenda pode ser:

- Espera - a encomenda ainda não foi atendida pelos funcionários.
- Preparação - a encomenda saiu da fila de espera e está a ser atendida pelos funcionários.
- Enviada - a encomenda foi processada pelos funcionários e será enviada para o respetivo cliente.

- Interrompida - a encomenda foi interrompida pelo sistema.

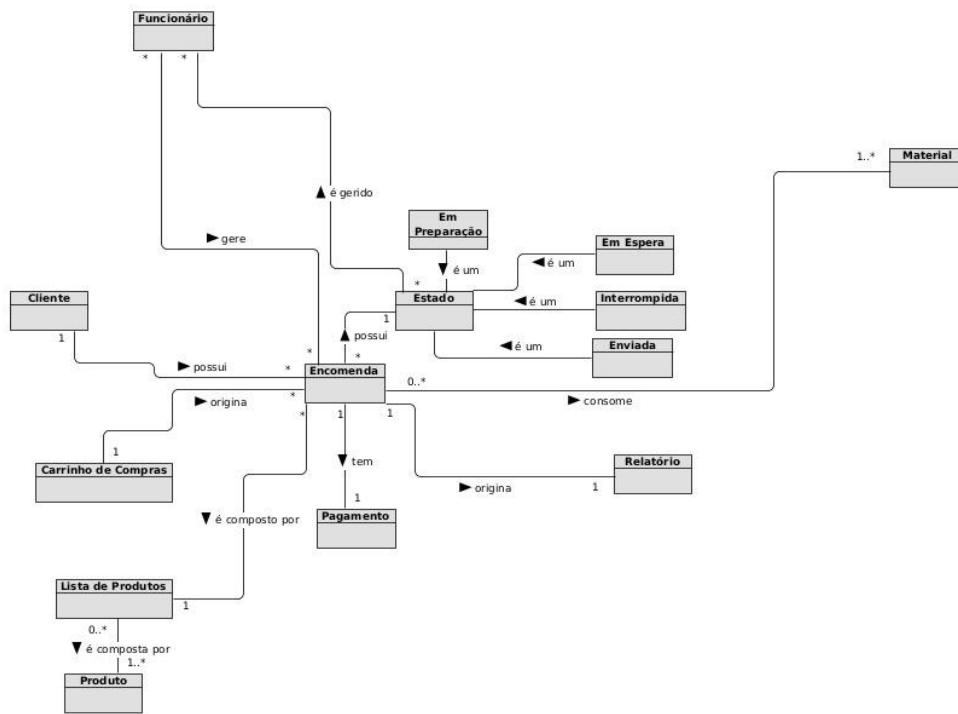


Figura 8 – Modelo de Domínio (Encomenda)

### 3.2.1.4. Funcionário

O funcionário:

- É um utilizador
- Gere as encomendas
- Altera o estado das encomendas
- Gere/visualiza o stock

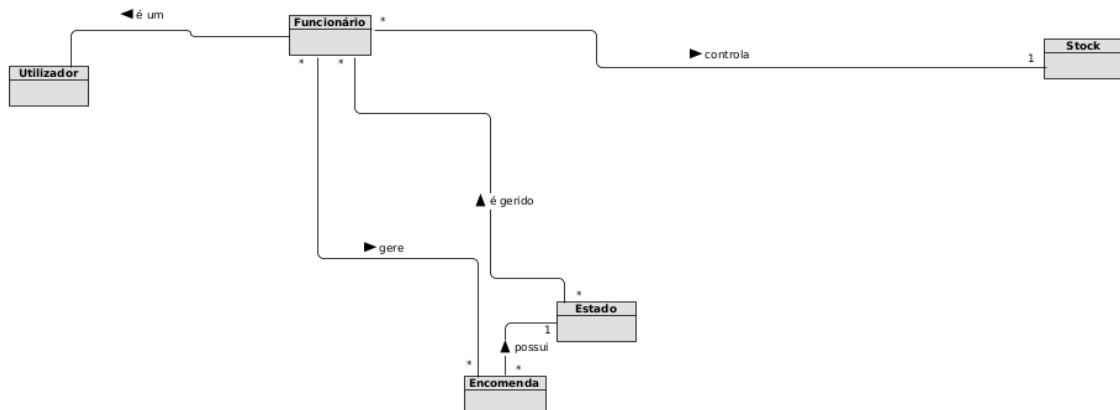


Figura 9 – Modelo de Domínio (Funcionário)

### 3.2.1.5. Stock

- O stock é controlado pelos funcionários
- O stock tem material, tendo estes uma quantidade associada
- Os produtos gastam material vindo do stock

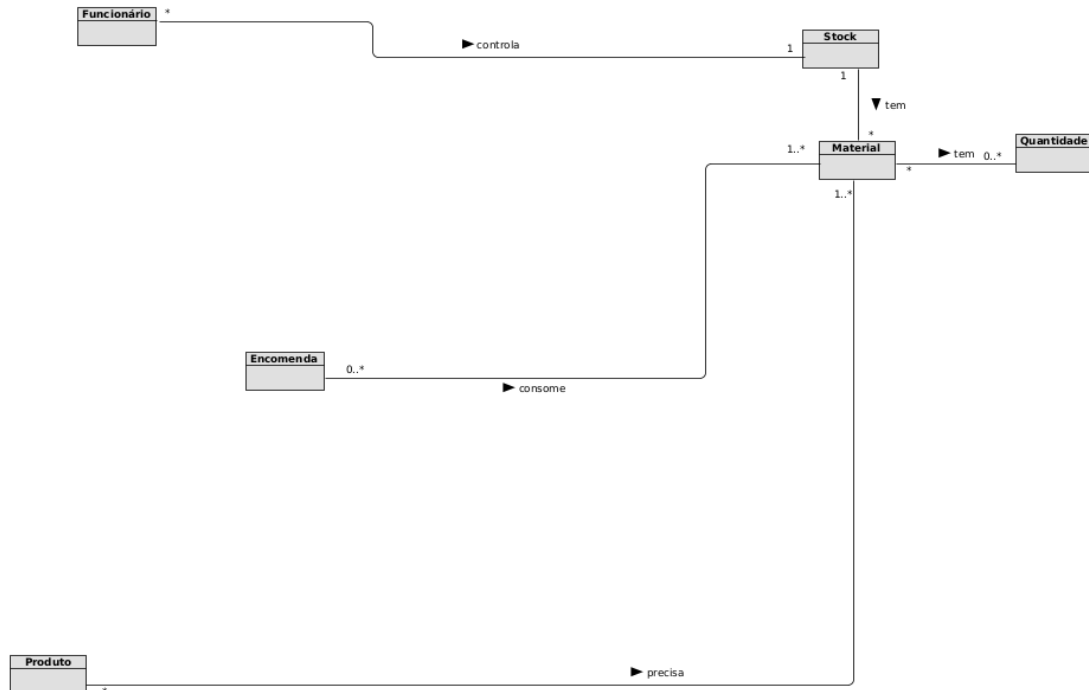


Figura 10 – Modelo de Domínio (Stock)

### 3.2.1.6. Produto

O produto:

- É utilizado nas listas de favoritos
- É utilizado nas listas de produtos
- Possui progressos
- Utiliza material do stock

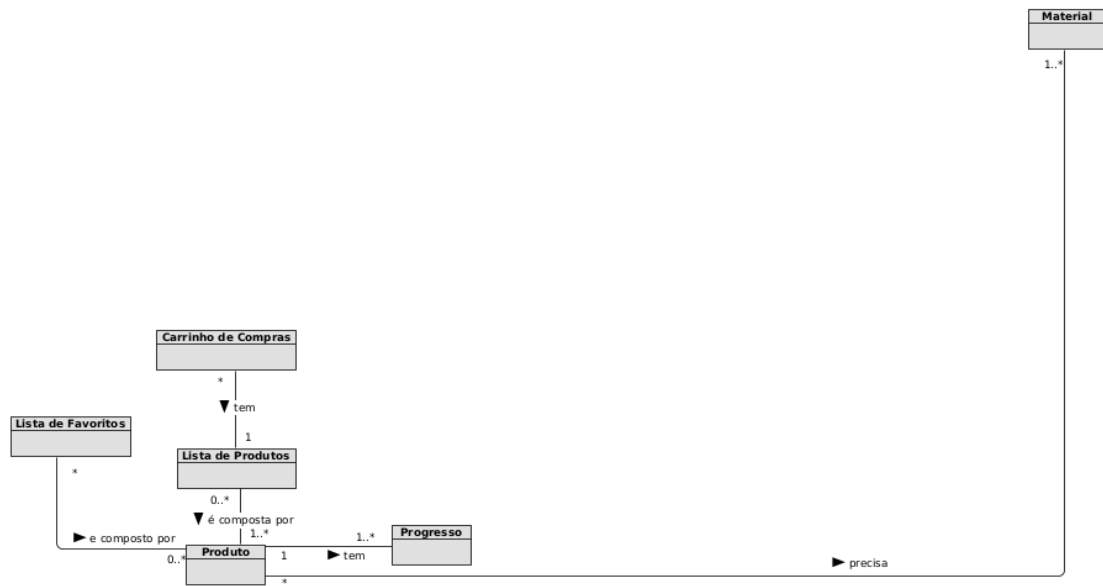


Figura 11 – Modelo de Domínio (Produto)

### 3.2.2. Diagrama de Componentes

Após uma análise do modelo de domínio, criamos um diagrama de componentes para representar o comportamento e funcionamento do nosso programa e como os componentes irão se relacionar entre si.

Este diagrama de componentes indica apenas os componentes e as interfaces que precisam/provem para a Lógica de Negócio. Certas interfaces que estes componentes oferecem/necessitam e que estão fora da lógica de negócio não foram consideradas neste diagrama. Por exemplo, o componente “Utilizadores” irá precisar de uma interface para poder guardar os utilizadores numa base de dados e irá fornecer uma interface para o UI (User Interface), porém estas interfaces não estão diretamente relacionadas à lógica de negócio, por isso não foram consideradas.

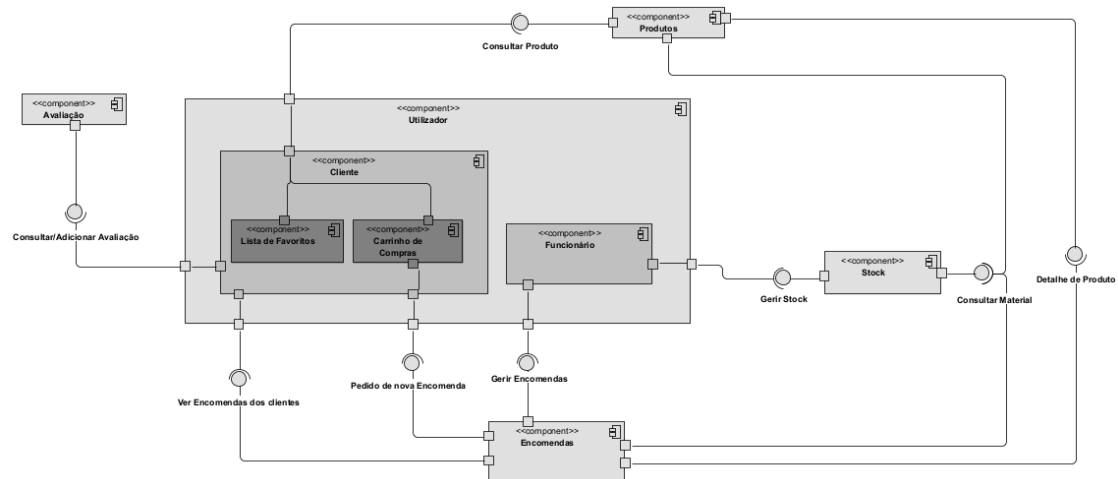


Figura 12- Diagrama de Componentes

No nosso diagrama de componentes existem cinco principais componentes, sendo estas *Avaliação*, *Utilizadores*, *Stock*, *Produtos* e *Encomendas*.

### 3.2.2.1. Avaliação



Figura 13 – Diagrama de Componentes (Avaliação)

Este componente (Figura 13) gere todas as avaliações da pastelaria. Interfaces que oferece:

- Métodos para consultar avaliações e para adicionar avaliações.

Interfaces que necessita: nenhuma.

### 3.2.2.2. Utilizadores

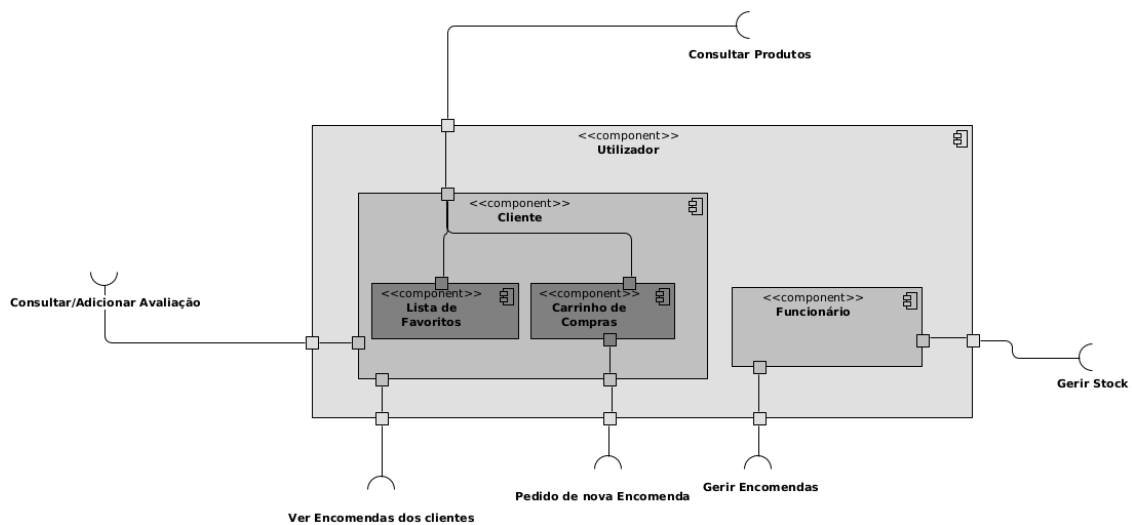


Figura 14 – Diagrama de Componentes (Utilizadores)

Este componente (Figura 14) gere todos os utilizadores registados no sistema, sejam eles clientes ou funcionários.

Este componente tem dois subcomponentes dentro dele:

- Cliente: este subcomponente gere todos os utilizadores do tipo “Cliente”,
- Funcionário: este subcomponente gere todos os utilizadores do tipo “Funcionário”.

Dentro do subcomponente “Cliente”, existem outros dois subcomponentes:

- Lista de Favoritos: este subcomponente gere todas as listas de favoritos dos clientes,
- Carrinho de Compras: este subcomponente gere todos os carrinhos de compras dos clientes.

Interfaces que oferece: nenhuma.

Interfaces que necessita:

- Métodos para consultar e adicionar avaliações (**clientes**),
- Métodos para ver encomendas dos clientes (**clientes**),
- Métodos para fazer um pedido para a criação de uma nova encomenda (**carrinho de compras**),
- Métodos para consultar Produtos existentes (**carrinho de compras & lista de favoritos**),
- Métodos para gerir as encomendas na linha de montagem (**funcionário**),

Métodos para gerir e consultar stock (**funcionário**).

### 3.2.2.3. Stock



Figura 15 – Diagrama de Componentes (Stock)

Esta componente (Figura 15) gere todos os materiais e as suas quantidades da pastelaria.

Interfaces que oferece:

- Métodos que gerem os materiais do stock, permitindo inserção e remoção de um, material do stock e modificar a quantidade de um material,
- Métodos de consulta ao stock, não permitindo a sua modificação.

Interfaces que necessita: Nenhuma.

### 3.2.2.4. Produtos

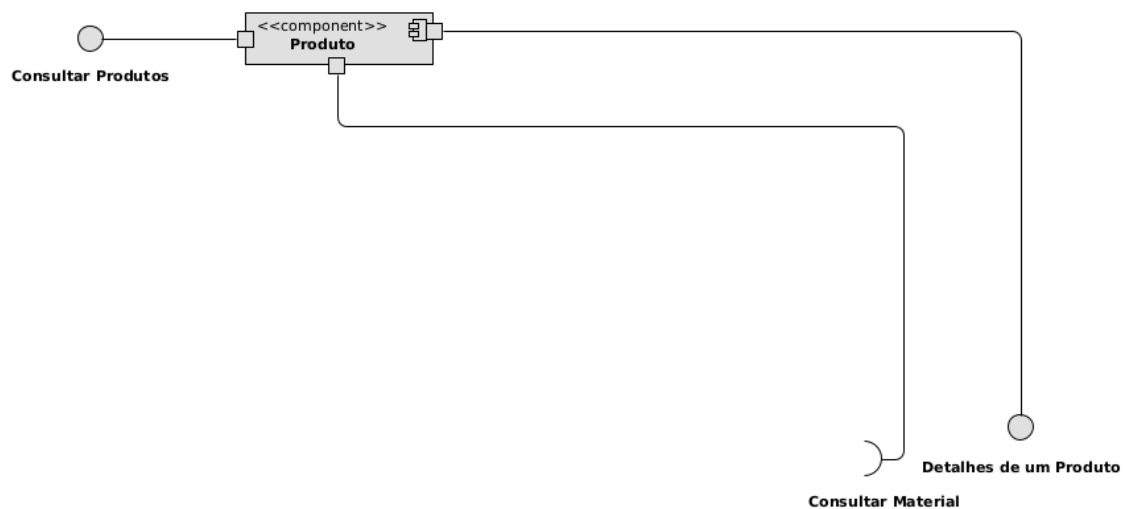


Figura 16 – Diagrama de Componentes (Produtos)

Esta componente gere todos os produtos existentes na pastelaria.

Interfaces que oferece:

- Métodos para consultar os produtos existentes,
- Métodos para obter detalhes de um produto.

Interfaces que necessita:

- Métodos para consultar os materiais existentes no stock.

### 3.2.2.5. Encomendas

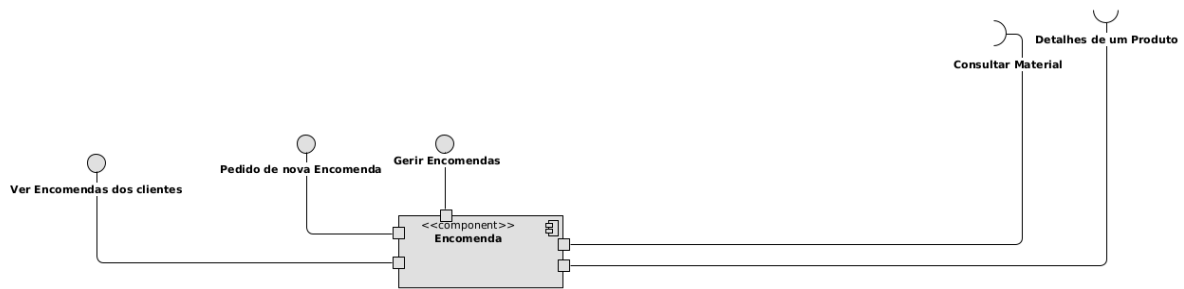


Figura 17 – Diagrama de Componentes (Encomendas)

Esta componente gere todas as encomendas.

Interfaces que oferece:

- Métodos para gerir/visualizar as encomendas,
- Métodos para ver as encomendas dos clientes,
- Métodos para a criação de novas encomendas (Os métodos fazem um pedido de encomenda, não é garantido que a encomenda será criada).

Interfaces que necessita:

- Métodos de consulta o material do stock para ver se a encomenda pode ser produzida,
- Métodos de consultar detalhes do produto para poder atualizar o progresso da encomenda.

## 3.3. Aspetos Comportamentais

Para conceituar os aspetos comportamentais, tínhamos à escolha a realização de um dos dois diagramas seguintes: diagrama de casos e diagrama de atividades.

Os diagramas de casos de uso identificam as interações entre os atores e o sistema, mostrando que funcionalidades se podem realizar, mas sem definir detalhadamente essas funcionalidades, ou seja, apresenta uma visão mais geral do sistema. Já os diagramas de atividade, descrevem detalhadamente o fluxo de trabalho num sistema. Por outras palavras, o diagrama de atividades apresenta os passos necessários para se executar uma dada tarefa por ordem sequencial e temporal das ações. Nesse sentido, este diagrama permite mostrar as funcionalidades do sistema de forma mais intuitiva e compreensível sobretudo para os stakeholders.

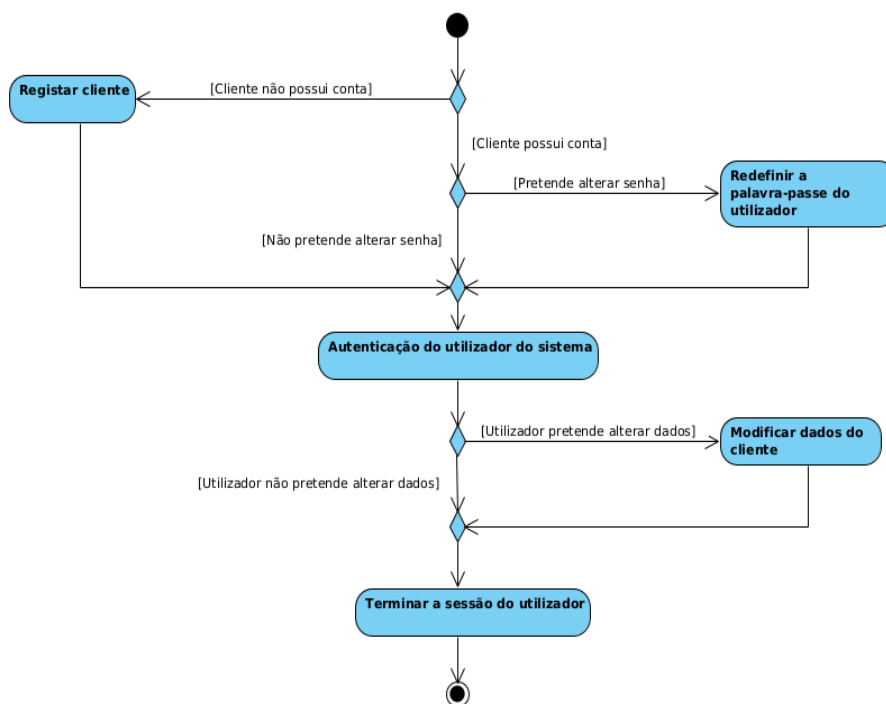
Posto isto, para modelar os aspetos comportamentais, optamos por realizar diagramas de atividades.

Para desenvolver os aspetos comportamentais, criamos mais do que um diagrama de atividades, isto é, analisamos e organizamos os requisitos definidos e, através deles, definimos várias funcionalidades do sistema. Para cada funcionalidade, fizemos o respetivo diagrama de atividades. As funcionalidades principais definidas são: gestão de contas do utilizador, gestão de



encomendas do cliente, gestão de encomendas do funcionário, gestão de favoritos e, por fim, gestão de stock.

### 3.3.1. Gestão de Contas do Utilizador



*Figura 18 – Gestão de Contas do Utilizador*

Neste diagrama de atividade (Figura 18), o primeiro nodo de decisão, testa se o cliente já tem ou não conta. Caso o cliente não possua uma conta, o fluxo segue para a atividade de “Registrar cliente”.

No nodo de decisão seguinte, caso o cliente pretenda mudar a sua palavra-passe, por exemplo, por esquecimento, o fluxo segue para a atividade de “Redefinir a sua palavra-passe”. De seguida, todos os caminhos de decisão são unidos num nó de junção e o fluxo prossegue para a ação de “Autenticação do utilizador no sistema”.

Depois disso, se o cliente pretender alterar os seus dados, o fluxo segue para a atividade de “Modificar dados do cliente”.

Assim, todos os caminhos são unidos no nodo de junção final e o fluxo segue para a atividade de “Terminar sessão” e, por fim, o fluxo avança para o nodo que termina a atividade.

### 3.3.2. Gestão de Stock

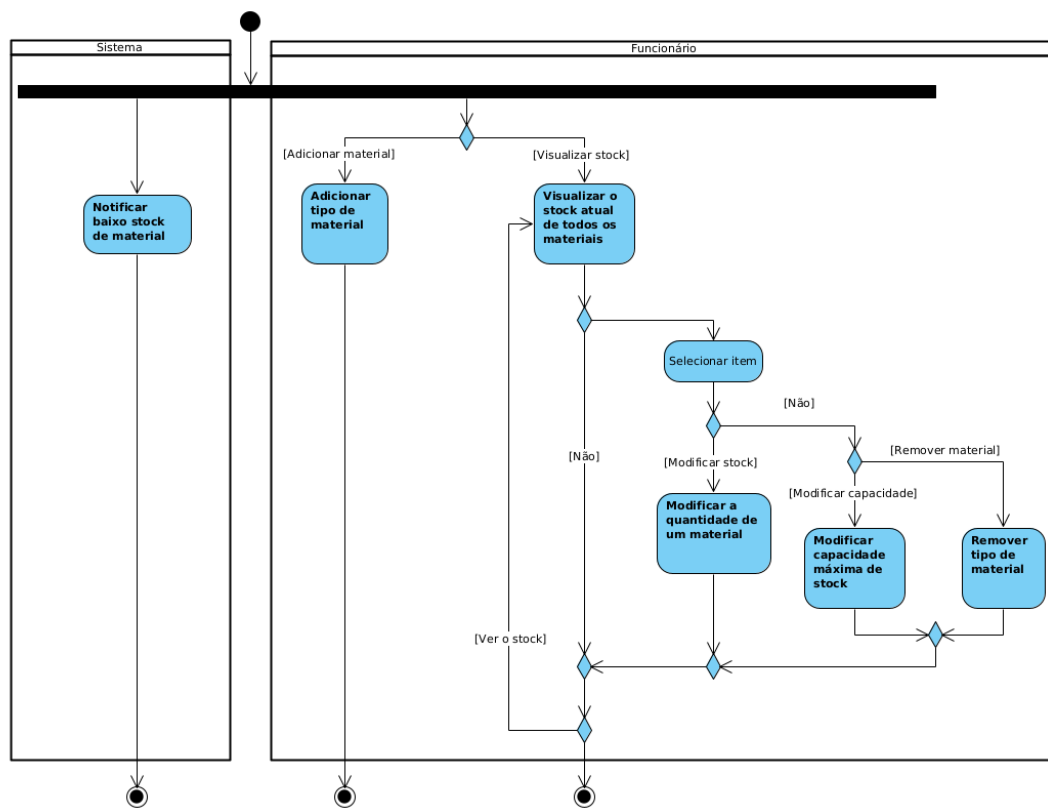


Figura 19 – Gestão de Stock

Neste diagrama (Figura 19), conseguimos ver que existem dois subsistemas chamados “Sistema” e “Funcionário”. O diagrama inicia no nodo inicial e o fluxo segue para um fork, indicando que estes dois subsistemas vão acontecer em simultâneo.

No subsistema “Sistema”, o fluxo segue para a atividade “Notificar baixo stock de material”, que notifica os funcionários quando há menos de 10% da capacidade de stock de um material.

Já no subsistema “Funcionário”, um nodo de decisão inicial apresenta ao funcionário duas alternativas: adicionar tipo de material (caso esse material não esteja registado no stock) e visualizar o stock atual de todos os materiais.

Caso o funcionário pretenda adicionar um tipo de material, o fluxo segue para a atividade “Adicionar tipo de material” e finaliza a atividade.

Caso o funcionário decida visualizar o stock, o fluxo segue para a atividade “Visualizar o stock atual de todos os materiais”. Após essa atividade, o funcionário pode selecionar um item do stock ou se só pretende visualizar o stock atual.

Se o funcionário selecionou um item do stock, utilizamos três nodos de decisão para modelar as possibilidades de escolha do funcionário. No primeiro nodo, o funcionário pode querer modificar a quantidade atual desse material em stock. Se não for essa a escolha, o funcionário pode modificar a capacidade máxima de stock deste item (útil para a notificação de baixo stock, como referido anteriormente) e, caso não queira modificar a capacidade máxima de stock, pode querer remover esse tipo de material do stock.

No final, estas três possibilidades mencionadas convergem para um nodo de decisão onde o funcionário pode escolher ver o stock novamente, desta forma o fluxo volta para a atividade “Visualizar o stock atual de todos os materiais”, ou pode escolher terminar a atividade.

### 3.3.3. Gestão de Favoritos

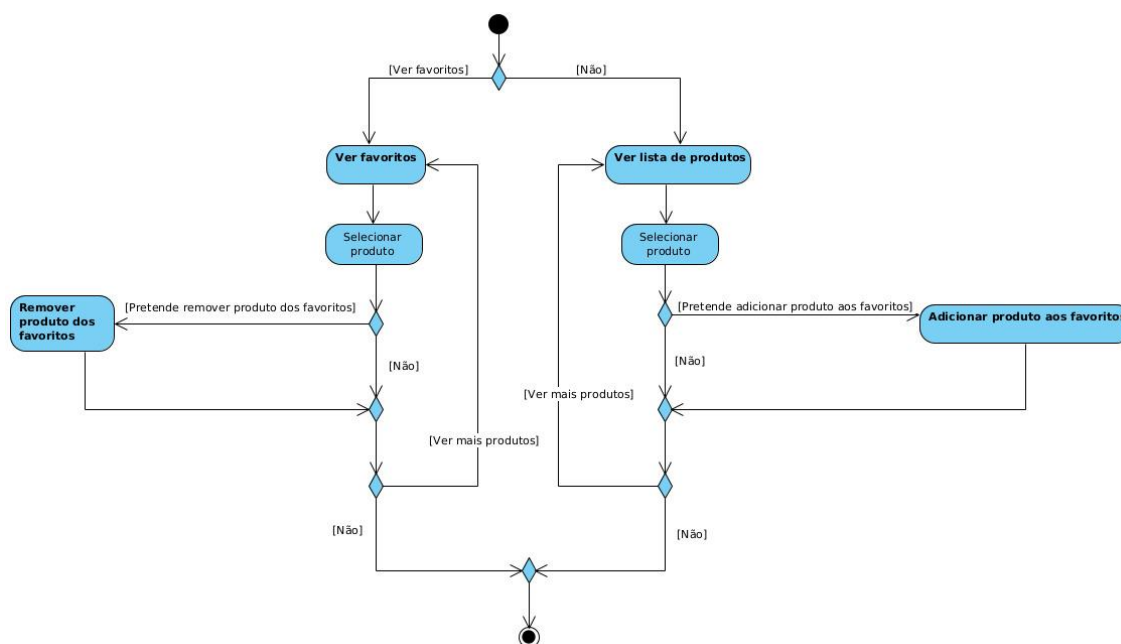


Figura 20 – Gestão de Favoritos

Neste diagrama (Figura 20) é visível um nodo de decisão inicial que apresenta ao cliente duas possíveis escolhas: ver a lista de favoritos, caso pretenda editar os itens já adicionados e ver a lista de produtos à venda, para procurar novos itens para adicionar aos favoritos.

Caso o cliente pretenda ver a sua lista de produtos favoritos então o fluxo segue para a atividade de “Ver favoritos”. Lá o cliente seleciona um item e decide, através de um nodo de decisão, se pretende ou não removê-lo da sua lista. Através do nodo de junção, ambos os caminhos de decisão são unidos pelo que, independentemente da escolha do cliente, este pode voltar a ver a sua lista de favoritos.

Se o cliente decidir não ver os seus favoritos e querer ver os produtos que estão à venda, então o fluxo segue para a atividade de “Ver lista de produtos”, onde o cliente seleciona um produto da lista e, através de um nó de decisão, decide se pretende ou não adicionar o produto aos favoritos. Independentemente de o cliente ter ou não adicionado um produto aos favoritos, pode voltar a ver a lista de produtos à venda.

Em ambas as decisões, os caminhos de decisão são unidos num nó de junção e o fluxo segue para o término da atividade.

### 3.3.4. Gestão de Encomendas – Cliente

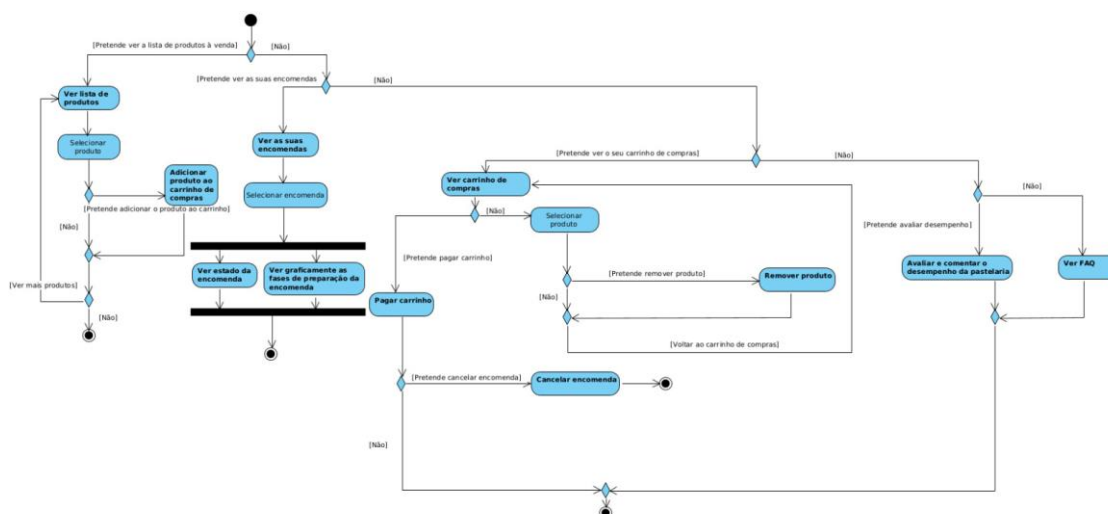


Figura 21 – Gestão de Encomendas (Cliente)

Neste diagrama (Figura 21), além da gestão das encomendas, agrupamos duas atividades complementares do cliente, que correspondem a avaliar e comentar o desempenho da pastelaria e a ver FAQ.

Neste sentido, utilizamos quatro nodos de decisão principais para modelar as possibilidades de escolha do cliente. No primeiro nodo, o cliente pode querer visualizar a lista de produtos disponíveis, caso pretenda adicionar um produto ao carrinho de compras. Se não for essa a escolha, o cliente pode ver as suas encomendas e, caso não queira ver as suas encomendas, pode querer visualizar o seu carrinho de compras, para o editar ou para pagar. Por fim, se não for nenhuma das outras decisões, o cliente pode avaliar o desempenho da pastelaria ou consultar as FAQ.

No nodo inicial, se o cliente escolher ver a lista de produtos à venda, seleciona um produto e, se pretender adicioná-lo ao carrinho de compras, o fluxo segue para a atividade de “Adicionar item ao carrinho de compras” e, independentemente de adicionar ou não o item, o fluxo segue para o nodo de decisão onde o cliente decide se quer ver mais produtos ou se pretende terminar a sua atividade.

Se o cliente não pretende ver os produtos à venda, o fluxo segue para um novo nodo de decisão onde o cliente pode escolher ver as suas encomendas. Se pretender visualizar as suas encomendas, seleciona uma encomenda, vê o estado em que a encomenda se encontra e assiste simultaneamente às suas fases de preparação graficamente. Se pretender ver o carrinho de compras, é utilizado um nodo de decisão onde, se pretender pagar o carrinho, o fluxo segue para a atividade de “Pagar carrinho” e, posteriormente, se pretender cancelar a sua encomenda, o fluxo segue para a atividade de “Cancelar encomenda” e, de seguida, para o término da atividade.

Se não pretender pagar o carrinho, então edita o carrinho, isto é, seleciona um item de lá e decide se pretende removê-lo. Independentemente de ter ou não removido o item, volta ao seu carrinho de compras.

Por fim, no último nó de decisão, o cliente, se pretender avaliar a pastelaria, o fluxo segue para a atividade de “Avaliar e comentar o desempenho da pastelaria”. Senão, segue para a visualização das perguntas frequentes (FAQ).

Em ambos os casos, o fluxo segue para o encerramento da atividade.

### 3.3.5. Gestão de Encomendas – Funcionário

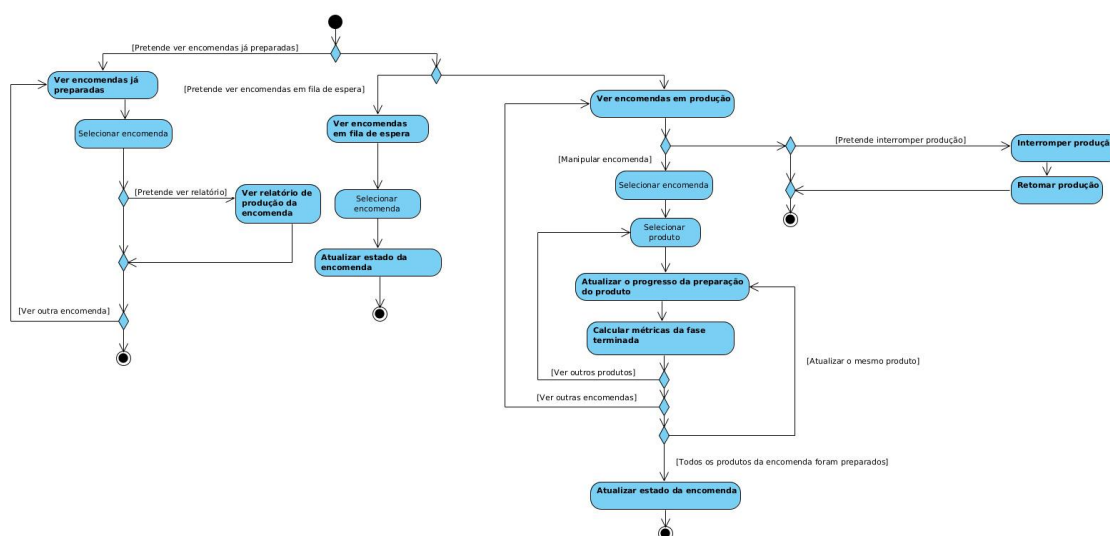


Figura 22 – Gestão de Encomendas (Funcionário)

À semelhança do diagrama anterior, utilizamos dois nodos de decisão principais para modelar as possibilidades de escolha do funcionário (Figura 22).

No primeiro nodo, o funcionário pode escolher ver as encomendas já preparadas, se pretender ver o relatório da produção. Caso contrário, o funcionário pode decidir ver as encomendas que estão em fila de espera. Senão, escolhe ver as encomendas que estão em produção.

Neste sentido, se o funcionário escolher ver as encomendas já preparadas, seleciona a encomenda e, se pretender ver o seu relatório de produção, o fluxo segue para a atividade de “Ver relatório de produção da encomenda” e, independentemente de ver ou não o relatório, pode procurar outras encomendas ou terminar a atividade.

Se não pretender ver as encomendas já preparadas, tem duas decisões disponíveis: ou ver as encomendas na fila de espera ou então ver as encomendas em produção.

Se pretender ver as encomendas em fila de espera, seleciona a encomenda e atualiza o seu estado, por exemplo, para “Em Preparação”.

Senão, vê as encomendas que estão em produção. Nesse sentido, o funcionário pode escolher manipular uma encomenda, com o objetivo de atualizar o seu progresso, ou controlar a produção

geral das encomendas. Ao escolher manipular uma encomenda, o funcionário seleciona uma encomenda, seleciona um produto que pertence a essa encomenda, atualiza o progresso de preparação do produto e o sistema calcula as métricas da fase terminada. Caso existam mais produtos naquela encomenda para atualizar o progresso, o funcionário pode voltar e selecionar outro produto. Se houver outras encomendas a precisar de atualização, o funcionário pode ver outras encomendas e proceder à sua manipulação.

No último nodo de decisão, se a produção do produto selecionado ainda não acabou, o fluxo volta a atualizar o progresso da preparação do produto. Quando todos os produtos da encomenda estão preparados, o fluxo segue para a atividade de atualizar o estado da encomenda e a atividade finaliza.

Em vez de decidir manipular uma encomenda, o funcionário pode escolher controlar a produção geral de encomendas, isto é, interromper a produção das encomendas e, de seguida, retomá-la. Além disso, em ambas situações, o nodo de término termina a atividade



## 4.2. Descrição detalhada dos vários elementos de dados e os seus relacionamentos

### 4.2.1. Descrição das tabelas

<u>Tabela</u>	<u>Descrição</u>
<b>Utilizador</b>	Representa os utilizadores do sistema, que podem ser clientes ou funcionários. Nesta tabela, armazenam-se as informações relevantes dos mesmos, sendo estas email (que corresponde à chave primária), nome e senha, relevantes para atender, por exemplo, às funcionalidades de registo e autenticação no sistema. Além disso, também se armazena a morada e número de telemóvel do cliente, relevante para a logística das entregas.
<b>Cliente_Favoritos</b>	Representa as tartes que o cliente marcou como favoritas, fazendo a conexão entre o email do utilizador e a identificação da tarte. Isto permite que o cliente facilmente acesse à funcionalidade de gerir os seus produtos favoritos, o que melhora a experiência do cliente durante a utilização do sistema.
<b>Avaliacao</b>	Representa as avaliações dos clientes ao desempenho da pastelaria. Assim, permite que o cliente possa deixar uma avaliação numérica e, opcionalmente, um comentário a avaliar a pastelaria. A avaliação está associada ao email do cliente que a fez.
<b>CarrinhoDeCompras</b>	Representa os carrinhos de compras dos clientes da pastelaria. Cada carrinho está associado a um cliente através do email. Esta tabela gere as compras antes de estas se concretizarem em encomendas.
<b>Carrinho_Tarte</b>	Representa os produtos que se encontram no carrinho de compras do cliente. Esta tabela associa as tartes ao carrinho, com uma quantidade associada a cada tarte. A tabela é essencial para a gestão de produtos associados a um carrinho de compras.



<b>Encomenda</b>	Representa os carrinhos de compras que se concretizaram em encomendas. Está, portanto, associada ao identificador do carrinho de compras e ao email do cliente que realizou a encomenda. A encomenda é também caracterizada pelo estado em que se encontra e por um relatório de produção, que é finalizado quando termina a preparação da encomenda.
<b>Pagamento</b>	Representa o pagamento da encomenda por parte do cliente. Cada pagamento está ligado a uma encomenda pelo seu identificador. É também armazenado o valor pago e a data de pagamento. O pagamento é essencial para proceder à produção da encomenda.
<b>Montagem</b>	Representa as diversas fases de preparação de cada tarte. Cada etapa é identificada por um par (idTarte, idFase), sendo também caracterizada por um nome, uma descrição, uma imagem que corresponde à sua representação gráfica e um indicador da sua próxima fase. Esta tabela é essencial para atender às funcionalidades correspondentes à visualização gráfica das encomendas por parte dos clientes, assim como à atualização do progresso da encomenda, por parte do funcionário.
<b>TarteEncomenda</b>	Representa as tartes pertencentes a uma encomenda. É caracterizada pelo identificador da encomenda à qual pertence, um identificador do tipo de tarte, e um atributo idFase, associado à tabela Montagem, que indica a fase atual da sua preparação no processo de produção. Esta tabela permite acompanhar o progresso da preparação de cada tarte na linha de montagem e associar a mesma à encomenda correspondente.
<b>Tarte</b>	Representa as tartes da pastelaria. Cada tarte é caracterizada por um identificador único, um nome, uma descrição e o seu preço. É um ponto central do sistema e atende funcionalidades de gestão de encomendas, carrinhos de compras e favoritos.
<b>Tarte_Material</b>	Representa a relação entre as tartes e os ingredientes necessários para a sua produção. Esta tabela está então associada ao identificador da tarte e ao identificador do material, sendo também caracterizada pela quantidade necessária de cada material. Esta tabela é essencial para a gestão de stock da pastelaria.

<b>Material</b>	Representa os ingredientes utilizados na produção de tartes. Cada material é caracterizado por um identificador, um nome, uma quantidade disponível atualmente e uma capacidade máxima de stock. Atende, assim, a funcionalidades relacionadas à gestão de stock, além de verificar se uma dada encomenda pode ser atendida.
-----------------	--

*Tabela 39 – Descrição das Tabelas do Modelo Lógico*

#### 4.2.2. Especificação das colunas das tabelas (atributos)

ATRIBUTO	DOMÍNIO/TIPO	DESCRIÇÃO	EXEMPLO	NULO
email	VARCHAR(45)	Endereço de email do utilizador	abcd@gmail.com	Não
senha	VARCHAR(45)	Palavra-passe de acesso do utilizador	password123!	Não
nome	VARCHAR(60)	Nome completo do utilizador	Rui Sousa Tavares	Não
tipo	ENUM('Funcionário', 'Cliente')	Indica se o utilizador é um funcionário ou cliente da pastelaria	Cliente	Não
morada	VARCHAR(255)	Endereço de morada do utilizador	Rua do Casal, 45, Felgueiras	Sim
telemóvel	VARCHAR(10)	Número de telemóvel do utilizador	912345678	Sim

*Tabela 40 – Tabela Utilizadores*

ATRIBUTO	DOMÍNIO	DESCRIÇÃO	EXEMPLO	NULO
idTarte	INT	Identificador único da tarte	2	Não

cliente	VARCHAR(45)	Email identificador do utilizador	abcd@gmail.com	Não
---------	-------------	-----------------------------------	----------------	-----

*Tabela 41 – Tabela Cliente\_Favoritos*

ATRIBUTO	DOMÍNIO	DESCRIÇÃO	EXEMPLO	NULO
cliente	VARCHAR(45)	Email identificador do utilizador	abcd@gmail.com	Não
rating	INT	Avaliação numérica dada pelo cliente, de 1 a 5	4	Não
comentario	VARCHAR(500)	Comentário do cliente sobre a pastelaria	adorei	Sim

*Tabela 42 – Tabela Avaliacao*

ATRIBUTO	DOMÍNIO	DESCRIÇÃO	EXEMPLO	NULO
id	INT	Identificador único do carrinho de compras	2	Não
proprietario	VARCHAR(45)	Email identificador do utilizador	abcd@gmail.com	Não

*Tabela 43 – Tabela CarrinhoDeCompras*

ATRIBUTO	DOMÍNIO	DESCRIÇÃO	EXEMPLO	NULO
idCarrinho	INT	Identificador único do carrinho de compras	12	Não
idTarte	INT	Identificador único da tarte	1	Não
quantidade	INT	Quantidade de unidades da tarte adicionadas ao carrinho	2	Não

Tabela 44 – Tabela Carrinho\_Tarte

ATRIBUTO	DOMÍNIO	DESCRIÇÃO	EXEMPLO	NULO
id	INT	Identificador único da encomenda	6	Não
estado	ENUM('Em espera', 'Interrompida', 'Em preparação', 'Enviada')	Identifica o estado atual da encomenda no sistema.	Enviada	Não
relatorio	VARCHAR(500)	Relatório associado à encomenda com informações como custo, tempo e materiais utilizados em cada fase da preparação.	Tarte de Mirtilo Fase 1: Demorou 12 minutos	Não
realizada_por	VARCHAR(45)	Email identificador do utilizador	abcd@gmail.com	Não
idCarrinho	INT	Identificador único do carrinho de compras	4	Não

Tabela 45 – Tabela Encomenda

ATRIBUTO	DOMÍNIO	DESCRIÇÃO	EXEMPLO	NULO
id	INT	Identificador único do pagamento	3	Não
valor	FLOAT	Montante a pagar pela encomenda	27.54	Não

dataPagamento	DATETIME	Data e hora em que o pagamento da encomenda foi efetuado	2024-11-17 11:20:00	Não
idEncomenda	INT	Identificador único da encomenda	4	Não

Tabela 46 – Tabela Pagamento

ATRIBUTO	DOMÍNIO	DESCRIÇÃO	EXEMPLO	NULO
idTarte	INT	Identificador único da tarte a ser preparada	3	Não
idFase	INT	Identificador da fase de preparação	2	Não
nomeFase	VARCHAR(45)	Nome da fase de preparação atual	Levar ao forno	Não
descricaoFase	VARCHAR(255)	Descrição detalhada da fase de preparação atual	Nesta fase, lavou-se a tarte de mirtilo ao forno a 180° durante 25 minutos	Não
imagemFase	LONGBLOB	Imagem que representa a fase atual de preparação	(Imagem Binária)	Não
proximaFase	INT	Identificador da próxima fase de preparação da tarte	3	Não

Tabela 47 – Tabela Montagem

ATRIBUTO	DOMÍNIO	DESCRIÇÃO	EXEMPLO	NULO
----------	---------	-----------	---------	------

id	INT	Identificador único da tarte	2	Não
nome	VARCHAR(45)	Nome da tarte	Tarte de Mirtilo	Não
descricao	VARCHAR(255)	Descrição da tarte	Tarte com base de bolacha e recheio de mirtilo	Não
preco	FLOAT	Preço da tarte	14.90	Não

*Tabela 48 - Tabela Tarte*

ATRIBUTO	DOMÍNIO	DESCRIÇÃO	EXEMPLO	NULO
idTarte	INT	Identificador único da tarte	2	Não
idMaterial	INT	Identificador único do material	4	Não
quantidade	FLOAT	Quantidade do material necessário para a tarte	0.5	Não

*Tabela 49 – Tabela Tarte\_Material*

ATRIBUTO	DOMÍNIO	DESCRIÇÃO	EXEMPLO	NULO
id	INT	Identificador único do material	3	Não
nome	VARCHAR(45)	Nome do material	Farinha	Não
quantidade	FLOAT	Quantidade de material disponível	57.5	Não
capacidade	FLOAT	Capacidade máxima de material	200.0	Não

ATRIBUTO	DOMÍNIO	DESCRIÇÃO	EXEMPLO	NULO
idTarte	INT	Identificador do tipo de tarte	3	Não
idEncomenda	INT	Identificador da encomenda	1	Não
idFase	INT	Identificador da fase de preparação da encomenda	2	Sim

Tabela 51 – Tabela TarteEncomenda

### 4.2.3. Especificação das chaves e relacionamentos

**Utilizador**(email, senha, nome, tipo, morada, telemovel)

**Chave Primária:** email

**Cliente Favoritos**(idTarte, cliente)

**Chave primária:** idTarte, cliente

**Chave estrangeira:** idTarte **REFERENCES** Tarte(id)

**Chave estrangeira:** cliente **REFERENCES** Utilizador(email)

**Avaliacao**(cliente, rating, comentario)

**Chave primária:** cliente

**Chave estrangeira:** cliente **REFERENCES** Utilizador(email)

**CarrinhoDeCompras**(id, proprietario)

**Chave primária:** id

**Chave estrangeira:** proprietario **REFERENCES** Utilizador(email)

**Carrinho Tarte**(idCarrinho, idTarte, quantidade)

**Chave primária:** idCarrinho, idTarte

**Chave estrangeira:** idCarrinho **REFERENCES** CarrinhoDeCompras(id)

**Chave estrangeira:** idTarte **REFERENCES** Tarte(id)

**Encomenda**(id, estado, relatorio, realizada\_por, idCarrinho)

**Chave primária:** id

**Chave estrangeira:** realizada\_por **REFERENCES** Utilizador(email)

**Chave estrangeira:** idCarrinho **REFERENCES** CarrinhoDeCompras(id)

**Pagamento**(id, valor, dataPagamento, idEncomenda)

**Chave primária:** id

**Chave estrangeira:** idEncomenda **REFERENCES** Encomenda(id)

**Montagem**(idTarte, idFase, nomeFase, descricaoFase, imagemFase, proximaFase)

**Chave primária:** idTarte, idFase

**Chave estrangeira:** idTarte **REFERENCES** Tarte(id)

**TarteEncomenda**(idTarte, idEncomenda, idFase)

**Chave primária:** idTarte, idEncomenda

**Chave estrangeira** idTarte **REFERENCES** Tarte(id)

**Chave estrangeira** idEncomenda **REFERENCES** Encomenda(id)

**Chave estrangeira** idFase **REFERENCES** Montagem(idFase)

**Tarte**(id, nome, descricao, preco)

**Chave primária:** id

**Tarte Material**(idTarte, idMaterial, quantidade)

**Chave primária:** idTarte, idMaterial

**Chave estrangeira:** idTarte **REFERENCES** Tarte(id)

**Chave estrangeira:** idMaterial **REFERENCES** Material(id)

**Material**(id, nome, quantidade, capacidade)

**Chave primária:** id



## 5. Esboço dos Interfaces do Sistema

### 5.1. Estrutura geral das interfaces do sistema

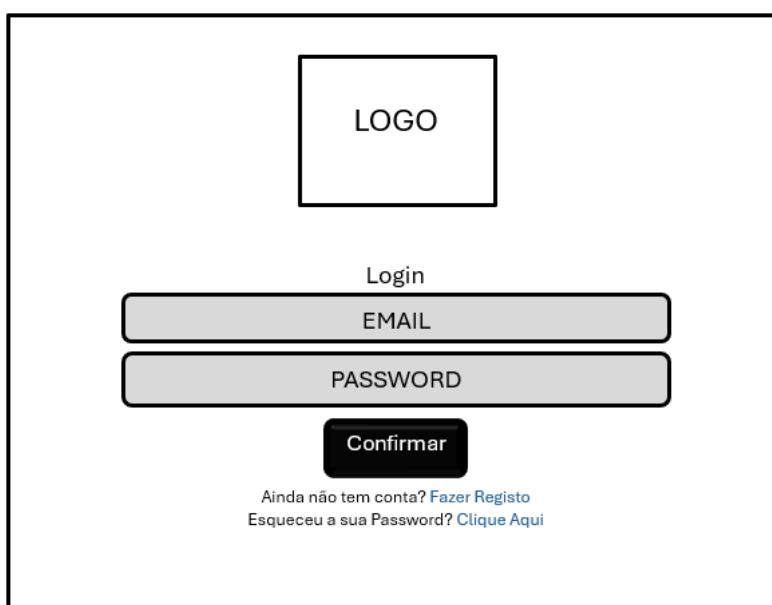
Após a definição completa do planeamento do sistema, desde o levantamento de requisitos até à definição da sua arquitetura, passámos para o planeamento das interfaces gráficas. Este processo teve como objetivo antever a vertente visual da aplicação e garantir que a interação com o utilizador fosse intuitiva e eficaz.

Para isso, criámos alguns mockups, que são esboços, focados em mostrar a estrutura e organização da aplicação, sem entrar em detalhes de design como cores ou tipografia. Estes esboços ajudam a planear o layout, hierarquia dos elementos e possíveis interações do utilizador com o sistema.

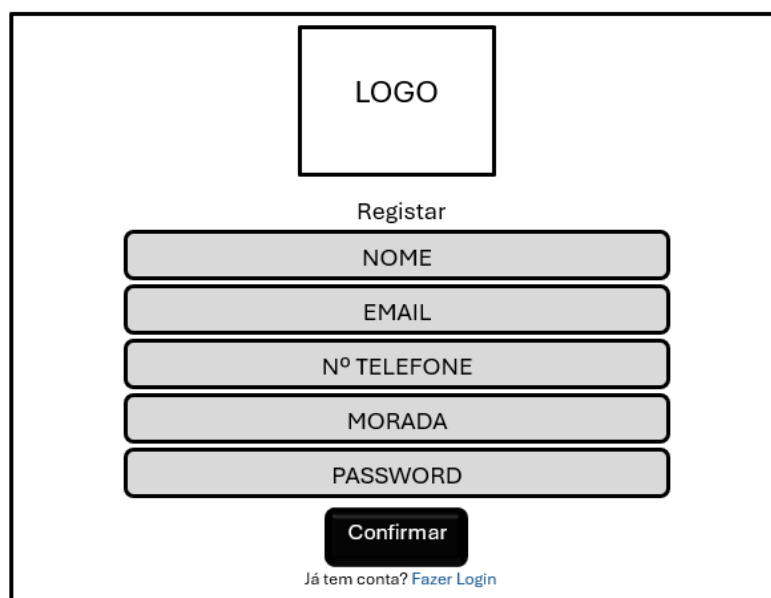
Além de facilitarem a comunicação e o alinhamento entre a equipa de desenvolvimento e o cliente, os mockups desempenham um papel crucial na construção de interfaces acessíveis, simples e intuitivas, promovendo uma experiência de utilização positiva e inclusiva para todos os públicos.

### 5.2. Caracterização das interfaces

Neste subcapítulo, serão apresentados alguns dos esboços elaborados para o sistema, cuja continuação se encontra no Anexo 1.



*Figura 24 – Mockup de login*



Logo

Registrar

NOME

EMAIL

Nº TELEFONE

MORADA

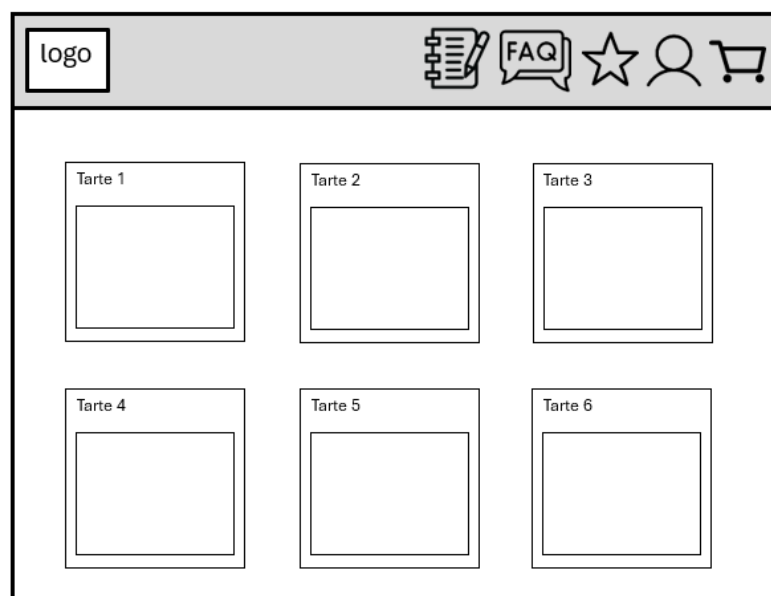
PASSWORD

Confirmar

Já tem conta? [Fazer Login](#)

This mockup shows a registration form layout. At the top is a box labeled 'Logo'. Below it is the text 'Registrar'. The form consists of six stacked input fields: 'NOME', 'EMAIL', 'Nº TELEFONE', 'MORADA', and 'PASSWORD'. Below these fields is a dark 'Confirmar' button. At the bottom, there is a link that says 'Já tem conta? Fazer Login'.

Figura 25 – Mockup de registo



logo

FAQ

Tarte 1

Tarte 2

Tarte 3

Tarte 4

Tarte 5

Tarte 6

This mockup shows a main menu layout. At the top is a header bar containing a 'logo' box, a 'FAQ' link with a speech bubble icon, and icons for a star, a user profile, and a shopping cart. Below the header is a grid of six items, each labeled 'Tarte 1' through 'Tarte 6', with a corresponding image placeholder box.

Figura 26 – Mockup de menu principal

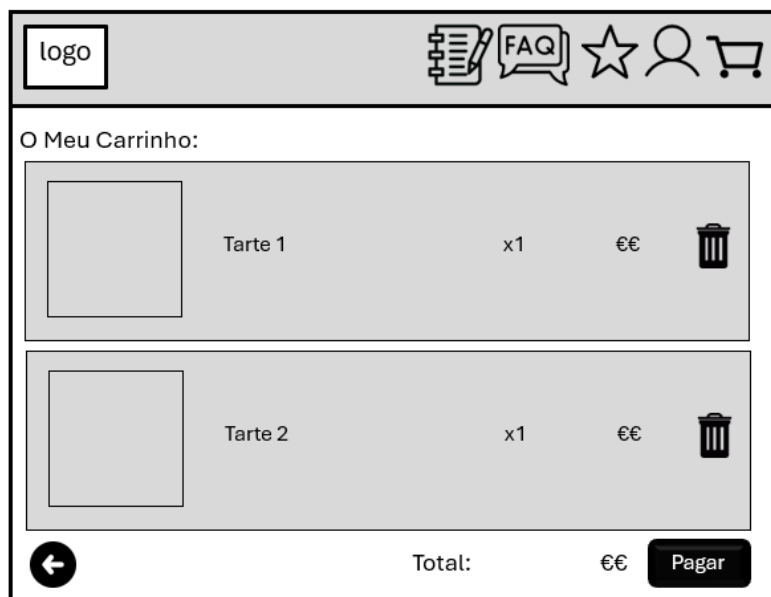


Figura 27 – Mockup de “Carrinho de Compras”

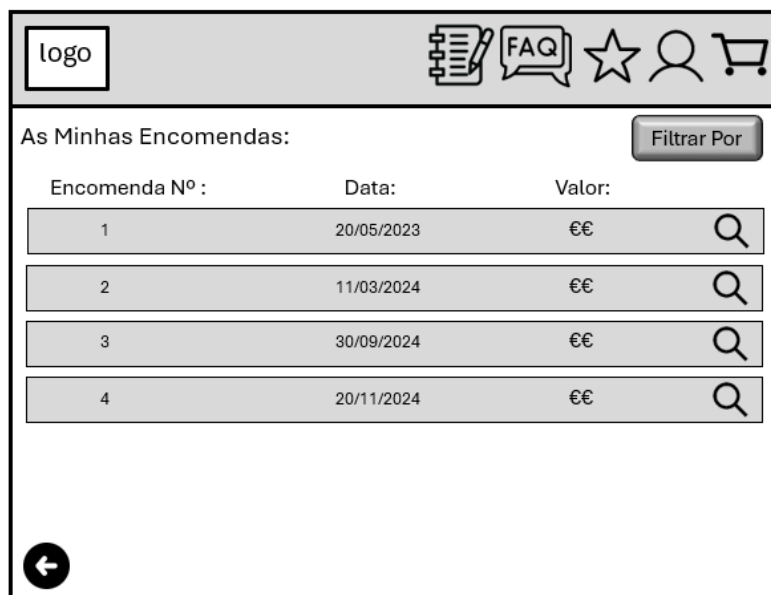


Figura 28 – Mockup de “As minhas Encomendas”

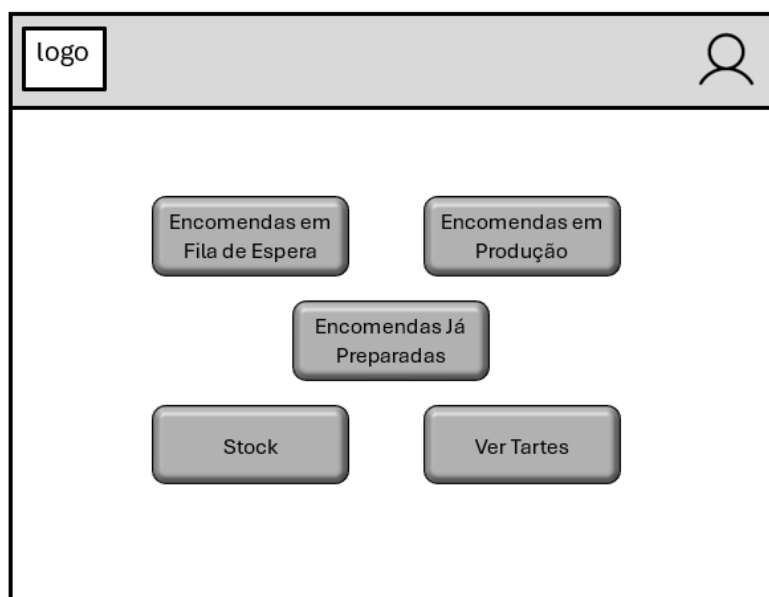


Figura 29 – Mockup “Menu Funcionário”

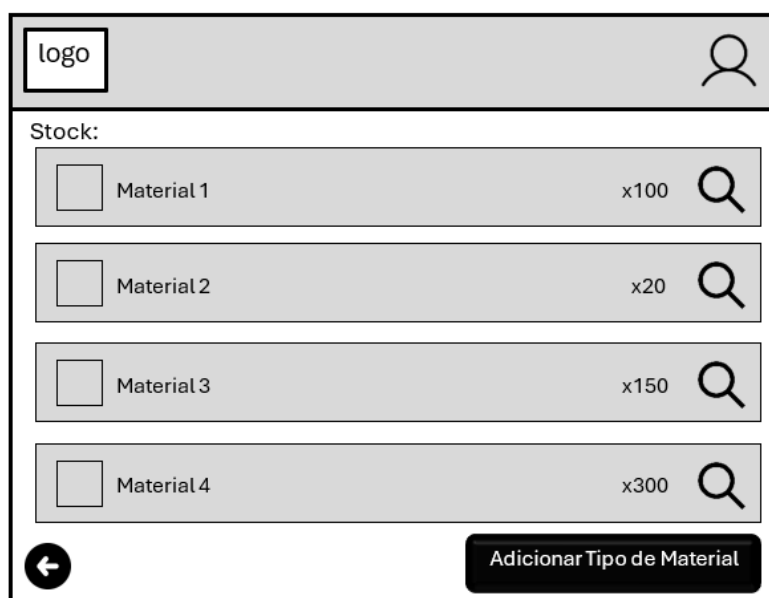


Figura 30 – Mockup de stock total

## 6. Implementação da Aplicação

### 6.1. Apresentação e descrição do processo de implementação realizado

Para a implementação deste sistema, foram usadas diversas ferramentas e lógicas de estrutura de desenvolvimento de projetos, que serão analisadas nestas subseqüentes secções do relatório.

#### 6.1.1. Apresentação das ferramentas e recursos utilizados

Tendo concluído a modelação do sistema a desenvolver, foi-nos possível começar a sua implementação. Neste sentido, a aplicação foi implementada utilizando várias ferramentas da *framework* ASP.NET e a plataforma de desenvolvimento *Visual Studio*. Além disso, a base de dados foi desenvolvida no *SQL Server*.

O *Blazor* é um *framework* que permite o desenvolvimento de aplicações *web* interativas e modernas, sem necessidade de utilizar *JavaScript*, substituindo as suas funções por C#. Faz parte do ecossistema *.NET* e facilita o desenvolvimento de interfaces para o utilizador. Assim, com recurso a esta ferramenta e, tendo como ponto de partida os esboços das interfaces, foi-nos possível implementar várias páginas interativas para a nossa aplicação.

#### 6.1.2. Estrutura da implementação

Relativamente à estrutura da implementação, este projeto contém várias pastas. No interior da diretoria *wwwroot*, encontra-se a pasta *css* e a pasta *images*. A pasta *css* contém todos os ficheiros *.css* desenvolvidos, necessários para embelezar, isto é, para dar estilo às diversas páginas desta aplicação. Estes ficheiros estão organizados de forma a facilitar a sua reutilização e são utilizados nos arquivos *.razor* para atribuir estilos específicos a cada página desta aplicação. Por outro lado, a pasta *images* contém todas as imagens utilizadas na aplicação.

Este projeto contém também uma pasta *Features*, onde se encontram as várias funcionalidades da aplicação. Por outras palavras, todos os ficheiros relacionados com uma dada funcionalidade, são colocados nessa *feature*. Por exemplo, a *feature* de Autenticação contém vários arquivos *.razor* relacionados, como o *Login* e o *Registo*. A pasta *Pages* contém apenas os arquivos *\_Host.cshtml*, *Error.cshtml* e *Error.razor*.

O arquivo *\_Host.cshtml* serve como ponto de entrada/inicialização para a aplicação e o ficheiro *Error.cshtml* permite que, sempre que ocorrer alguma falha, seja exibida uma página de erro personalizada com informação sobre o erro.

Na pasta *src* está todo o código correspondente à implementação do sistema, usando C# como linguagem principal.

Dentro desta pasta, existem 3 subpastas: **business**, **data** e **ui** (*user interface*). Na pasta **business**, tem toda a implementação da Lógica de Negócio. Na pasta **data**, tem toda a implementação da Base de Dados e na pasta **ui** tem toda a implementação da Interface do Utilizador, em modo texto.

O programa principal encontra-se no ficheiro **Program.cs**, onde possui dois modos. Se não tiver nenhum argumento, inicia em modo *Blazor*. Se o 1º argumento for “txt”, inicia em modo texto.

### 6.1.3. Apresentação e Descrição do Sistema

Neste capítulo vamos apresentar a implementação do sistema desenvolvido.

O sistema utiliza a arquitetura de “**3-Tier Application Architecture**”, por isso vamos apresentar as três camadas de forma detalhada.

#### 6.1.3.1. Lógica de Negócio

A lógica de negócio é onde toda a funcionalidade do programa e onde toda a implementação dos requisitos funcionais se encontram. A lógica de negócio está concentrada num objeto chamado “**Facade**”, onde nele se encontram os principais métodos para realizar a funcionalidade do sistema “Cantinho Das Encomendas”.

A seguir apresentamos todos os métodos, quais as suas funcionalidades e a quais requisitos funcionais estão associados.

##### **Registar Cliente** (requisito funcional 1)

Para permitir que os clientes consigam aceder ao serviço, como, por exemplo, proceder ao pagamento de uma encomenda, devem encontrar-se registados no sistema. O método **RegistarCliente** recebe os dados fundamentais do cliente e cria o cliente na base de dados.

##### **Autenticação no Sistema** (requisito funcional 2)

Para permitir o cliente/funcionário acessar o serviço, devem autenticar-se no sistema. O método **IniciarSessao** recebe o email e a palavra-passe do utilizador. Caso as credenciais estejam válidas, ou seja, sejam iguais às que existem no sistema, o utilizador fica autenticado e pode utilizar o sistema.

Existem outros métodos associados à autenticação, como:

- **isFuncionario**: indica se o utilizador autenticado é um funcionário.
- **SessaoIniciada**: indica se existe algum utilizador autenticado no sistema.
- **EmailLogin**: indica o email do utilizador que está a usar o sistema.

##### **Terminar Sessão** (requisito funcional 3)

Para permitir que o utilizador saia do sistema, evitando que outro utilizador possa aceder às suas informações, o utilizador deverá terminar a sua sessão com o método **TerminarSessao**.

#### **Alterar a palavra-passe** (requisito funcional 4)

Em caso de esquecimento da palavra-passe, o utilizador deverá usar o método **AlterarSenha**, indicando o seu email e a nova palavra-passe.

Inicialmente, este método enviaria para o email do utilizador um código, para confirmar a mudança de palavra-passe. Como este serviço não foi implementado, a alteração de palavra-passe acontece sem a necessidade de código de confirmação.

#### **Visualizar o estado da encomenda** (requisito funcional 5)

O cliente indica ao sistema qual a encomenda que deseja visualizar e o sistema, através do método **GetEncomenda**, devolve a encomenda indicada. Desta forma, usando os métodos de encomendas é possível ver os detalhes da encomenda, como, por exemplo, o seu estado.

#### **Ver Tartes da Pastelaria** (requisito funcional 6)

Através do método **ObterTartes**, são devolvidas todas as tartes registadas no sistema. Desta forma, o cliente consegue ver que tartes existem na pastelaria.

#### **Ver Carrinho de Compras** (requisito funcional 7)

Caso o cliente deseje visualizar o seu carrinho de compras, o método **ObterCarrinhoCompras** devolve o carrinho de compras de um utilizador. Desta forma, usando os métodos de **CarrinhoCompras**, é possível ver quais produtos estão dentro dele.

#### **Adicionar um Produto ao Carrinho de Compras** (requisito funcional 8)

Para que o cliente consiga comprar produtos da pastelaria, deverá adicionar os produtos ao seu carrinho de compras. Isso é possível através do método **AddProdutoCarrinhoCompras**, indicando o produto e o email do utilizador ao qual o carrinho de compras está associado.

Se tentarmos adicionar um produto que já se encontra no carrinho, o carrinho de compras indica quantas vezes esse produto foi adicionado.

#### **Remover um Produto do Carrinho de Compras** (requisito funcional 9)

Caso um cliente deseje retirar um produto do seu carrinho de compras, recorreremos ao método **RemoveProdutoCarrinhoCompras**, indicando o produto e o email do utilizador associado ao carrinho de compras.

#### **Cliente paga pela Encomenda no seu Carrinho de Compras** (requisito funcional 10)

Caso um cliente esteja satisfeito com os produtos do carrinho de compras, o cliente poderá pagar o carrinho de compras, transformando o seu carrinho de compras numa encomenda. Este processo recorre ao método **PagarEncomenda**.

Neste método, é feita a verificação de se existe material suficiente para produzir a encomenda, recorrendo ao método **TemMaterialSuficiente**. Após a verificação, o Carrinho de Compras é transformado numa Encomenda (usando o método **AddEncomendaCarrinhoCompras**),

adicionando a mesma ao cliente (usando o método **AddEncomenda**) e esse carrinho de compras é limpo, para que possa ser usado novamente (usando o método **EsvaziaCarrinhoCompras**).

Neste projeto assumimos que o pagamento foi efetuado, obviamente não verificando se houve um pagamento físico/real, para que tal método consiga ser executado.

#### **Cancelar a Encomenda** (requisito funcional 11)

Caso o cliente deseje cancelar uma encomenda, deverá indicá-la ao sistema e o método **CancelarEncomenda** tentará efetuar o cancelamento.

Neste método, se a Encomenda estiver em estado de “Em espera”, o cancelamento acontece com sucesso. Caso contrário, o cancelamento falha.

#### **Cliente deve poder ver as suas Encomendas** (requisito funcional 12)

O cliente consegue aceder às suas encomendas com o método **GetEncomendasCliente**, que devolve uma lista com todas as suas encomendas.

#### **Cliente deve poder Avaliar e Comentar o desempenho da Pastelaria** (requisito funcional 13)

O cliente pode avaliar a pastelaria com o método **AddAvaliacao**, indicando qual a sua classificação e podendo também indicar um comentário. Após esse método, a sua avaliação fica guardada no sistema.

#### **Cliente pode visualizar as avaliações da Pastelaria** (requisito funcional 14)

Caso um cliente deseje ver as avaliações da Pastelaria, pode fazê-lo recorrendo ao método **GetAvaliacoes**, que dará todas as avaliações registadas no sistema.

#### **Cliente pode adicionar Produtos aos seus Favoritos** (requisito funcional 15)

O cliente poderá adicionar um produto aos seus favoritos com o método **AddProdutoFavoritos**, indicando qual o produto que pretende adicionar.

#### **Cliente pode remover Produtos dos seus Favoritos** (requisito funcional 16)

O cliente poderá remover um produto dos seus favoritos recorrendo ao método **RemoveProdutoFavoritos**, indicando qual o produto que pretende retirar.

#### **Cliente pode ver os seus Favoritos** (requisito funcional 17)

Caso o cliente deseje ver os seus favoritos, recorre ao método **GetFavoritos**, que dará uma lista com todos os seus produtos favoritos.

#### **Cliente recebe uma notificação sempre que é possível efetuar a sua Encomenda** (requisito funcional 18)



Este requisito não foi 100% implementado, pois não implementamos o serviço de enviar email ao cliente. Porém, existe um método chamado **ConseguePagarEncomenda**, que indica se um cliente consegue efetuar o pagamento ou não, tendo em consideração se há material suficiente para produzir tal encomenda.

**Cliente pode modificar o seu nome, email, morada e número de telefone** (requisito funcional 19).

O cliente poderá modificar os seus dados usando os seguintes métodos:

- **ModifyClienteNome**: modifica o nome do cliente
- **ModifyClienteMorada**: modifica a morada do cliente
- **ModifyClienteTelefone**: modifica o número de telemóvel do cliente
- **ModifyClienteEmail**: modifica, se possível, o email do cliente. Se o email não foi registado, a mudança pode ser efetuada, porém, se o email pretendido já se encontrar registado no sistema, não é possível efetuar a mudança de email.

**Cliente deve poder ver as “Perguntas Frequentes”** (requisito funcional 20)

O cliente consegue aceder às “Perguntas Frequentes” com o método **GetFAQ**, que dá uma lista com todas as perguntas, assim como as respostas associadas às mesmas, que se encontram registadas no sistema.

**O cliente deve poder visualizar graficamente as diversas fases da preparação da sua Encomenda** (requisito funcional 21)

O cliente poderá ver as diversas fases da preparação da sua encomenda. Para isso, recorre ao método **GetEncomenda** para obter a Encomenda que pretende visualizar. Obtendo a Encomenda, podemos usar os *getters* de encomenda para obter quais os produtos da encomenda, e as fases dos mesmos.

Infelizmente a parte gráfica não foi implementada.

**Funcionário deve poder ver as informações de uma determinada encomenda** (requisito funcional 22)

Como mencionado acima, usamos o método **GetEncomenda** para obter uma Encomenda. Com essa encomenda, o funcionário pode usar os métodos de Encomenda para obter as informações referentes à mesma.

**Funcionário deve poder ver as Encomendas que estão a ser preparadas** (requisito funcional 23)

O funcionário consegue ver as Encomendas que estão a ser preparadas com o método **GetEncomendasDoing**, que dá uma lista de todas as encomendas cujo estado é “Em preparação”.

**Funcionário deve poder ver as Encomendas que já foram preparadas** (requisito funcional 24)

O funcionário consegue ver as Encomendas que já foram preparadas com o método **GetEncomendasDone**, que dá uma lista de todas as encomendas cujo estado é “Feita” ou “Enviada”.

**Funcionário deve poder ver as Encomendas que estão na fila de espera** (requisito funcional 25)

O funcionário consegue ver as Encomendas que se encontram na fila de espera com o método **GetEncomendasQueue**, que dá uma lista de todas as encomendas cujo estado é “Em Espera”.

**Funcionário deve poder ver o Stock** (requisito funcional 26)

Caso o funcionário deseje ver o stock atual do sistema, usa o método **VerStock**, que dá uma lista de todos os Materiais registados no sistema.

**Funcionário deve poder ver o relatório de produção de uma Encomenda** (requisito funcional 27)

O funcionário consegue obter o relatório de uma determinada encomenda usando o método **GetEncomendaRelatorio**.

Infelizmente, o Relatório não foi implementado.

**Funcionário deve poder interromper a produção de Encomendas** (requisito funcional 28)

Para que um funcionário consiga interromper a produção de Encomendas, deverá recorrer ao método **InterromperEncomendas**.

**Funcionário deve poder retomar a produção de Encomendas** (requisito funcional 29)

Caso um funcionário queira retomar a produção de Encomendas, deverá fazê-lo recorrendo ao método **RetomarEncomendas**.

**Funcionário deve poder atualizar o Estado de uma Encomenda** (requisito funcional 30)

Durante a confecção de produtos das encomendas, o funcionário poderá atualizar o estado da encomenda usando o método **AtualizarEstadoEncomenda**, indicando o id da encomenda.

Este método atualiza o estado das encomendas, que podem ir de “Em Espera” para “Em preparação” ou de “Em preparação” para “Enviado”/“Feito”.

**Funcionário deve poder atualizar o Progresso da preparação de uma Encomenda** (requisito funcional 31)

Durante a confecção de produtos das encomendas, o funcionário poderá atualizar o progresso de um certo produto de uma certa encomenda recorrendo ao método

**AtualizarProgressoEncomenda**, indicando o id da encomenda e qual a posição do produto que pretendemos atualizar.

Neste método, os produtos das encomendas vão progredindo, baseando-se nos procedimentos indicados pelos seus respetivos produtos, e quando não houver mais progressos, esse produto da encomenda é marcado como “Feito”.

**Funcionário deve poder modificar a capacidade máxima de Stock de qualquer Material** (requisito funcional 32)

Caso o Funcionário pretenda mudar a capacidade máxima de Stock de um certo material, deverá usar o método **SetCapMaxMaterial**, indicando qual o Material e a sua nova capacidade máxima.

**Sistema deve notificar o funcionário quando a capacidade de Stock de qualquer Material se encontrar a 10% da sua capacidade máxima** (requisito funcional 33)

Como dito acima, a funcionalidade de notificar ou enviar certas informações por email não foram implementadas, porém o funcionário consegue obter quais os Materiais que se encontram abaixo de 10% da sua capacidade máxima com o método **GetMaterialBaixoStock**.

Este método devolve uma lista de todos os Materiais que se encontram abaixo de 10% da sua capacidade máxima. Se este método devolver uma lista vazia, significa que todos os materiais do stock disponível se encontram acima de 10% da sua capacidade máxima.

**Funcionário deve poder modificar a quantidade de Material do Stock** (requisito funcional 34)

O funcionário pode modificar a quantidade de um Material do Stock com o método **SetMaterialQuantidade**, indicando qual o Material e a sua nova quantidade.

**Funcionário deve poder adicionar um novo Material ao Stock** (requisito funcional 35)

Caso na Pastelaria existam novos materiais, eles devem ser adicionados e registados no sistema. Para isso, o funcionário usa o método **AddMaterialStock**, indicando qual o nome do novo material e a sua capacidade máxima.

**Funcionário deve poder remover um novo Material do Stock** (requisito funcional 36)

Caso um material seja removido da pastelaria, o funcionário deverá removê-lo do sistema usando o método **RemoveMaterialStock**, indicando o nome do mesmo.

**Funcionário deve poder adicionar uma nova Tarte ao sistema** (requisito funcional 37)

A Pastelaria pode criar tartes/produtos. Para registá-los no sistema, o funcionário deve adicioná-lo com o método **AddTarte**, indicando o seu id, nome, preço, imagem, descrição, lista de materiais necessários para a sua confeção e uma lista de procedimentos para o produzir.

**Funcionário deve poder remover uma Tarte do sistema** (requisito funcional 38)

Caso a pastelaria deseje remover tartes/produtos do sistema, o funcionário deverá removê-lo usando o método **RemoveTarte**, indicando o seu id.

### Funcionalidades extras

Apesar de não estarem mencionados nos requisitos funcionais, a Lógica de Negócios tem outros métodos que poderão ser úteis:

1. **RegistarFuncionario**: método que permite a criação de um funcionário.
2. **ObterTarte**: método que dá uma tarte/produto específico indicando o seu ID.
3. **AddFAQ**: método que adiciona uma pergunta e a sua resposta nas “Perguntas Frequentes”.
4. **RemoveFAQ**: método que remove uma pergunta das “Perguntas Frequentes”, indicando qual a sua posição.
5. **MaterialExiste**: método que indica se um material existe no Stock do sistema, indicando o seu nome.
6. **GetUtilizador**: método que dado um email, indica qual o Utilizador associado a esse email, sendo possível depois usar métodos de Utilizador para aceder aos seus dados.
7. **ProducaoInterrompida**: método que indica se a produção está interrompida.
8. **FazerEncomendaProduto**: método que inicia a produção de um produto específico de uma encomenda, dado o ID da encomenda e a posição do produto. Este método indica que a produção deste produto está a ser tratada pelos funcionários.

### 6.1.3.2. Base de Dados

A base de dados é onde todos os dados do sistema são guardados de forma persistente numa base de dados. Esta camada é usada pela Lógica de Negócio para procurar ou guardar dados relevantes para o sistema.

Para tal efeito, usaríamos a base de dados *SQL Server* e iríamos usar vários DAO (*Data Access Objects*) como objetos que permitem fazer a conexão entre a Lógica de Negócio e a Base de Dados. Tendo isso em mente, foi criado um *script* SQL (*Structured Query Language*) que cria a base de dados modelada anteriormente. Esse *script* encontra-se na pasta *DataBase*.

Infelizmente a implementação dos DAOs não foi realizada, fazendo com que os dados sejam guardados em memória, ou seja, os dados não são guardados de forma permanente.

### 6.1.3.3. Interface do Utilizador

A interface do utilizador é onde a Lógica de Negócio e as suas funcionalidades são apresentadas, de forma visual, ao utilizador do sistema, neste caso, clientes e funcionários da pastelaria.

A interface foi dividida em duas: **Blazor** e **Menu de Texto**.

No *Blazor*, a interface do utilizador é muito melhor, esteticamente falando, e é o que desejávamos ter. Infelizmente a conexão com a Lógica de Negócio não aconteceu como queríamos, fazendo com que a manipulação do *Facade* não acontecesse.

No Menu de Texto, a interface do utilizador é simples, ou seja, texto no terminal. Porém, nela existe conexão com a Lógica de Negócio, permitindo manipulá-la, tal como pretendido.

Assim, através do Menu de Texto foi-nos possível mostrar as funcionalidades que eram pretendidas para esta aplicação Web.

## 6.2. Apresentação da aplicação e explicação dos serviços implementados

### 6.2.1. MainLayout

Todas as páginas desta aplicação contêm um *layout* comum, que corresponde a um menu horizontal localizado no topo da página. Este menu contém o logotipo da pastelaria e vários separadores que se podem visitar. Assim, cada página desta aplicação exibe este *layout*.

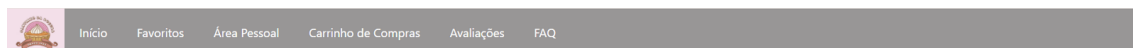


Figura 31 – Menu MainLayout

### 6.2.2. Página Inicial

A página inicial corresponde ao menu da pastelaria, isto é, aos produtos que estão à venda na aplicação. É possível aceder a esta página sempre que se carrega no separador “Início” do menu horizontal no topo da página. Além disso, é possível selecionar qualquer produto para ver a sua informação. Esta página utiliza um estilo *css* específico e este menu é gerado dinamicamente, na medida em que os dados do produto como o ID, nome e imagem são carregados através do *backend* do sistema.

## Bem-vindo ao Cantinho da Doçura!

"Na nossa pastelaria, cada doce é um pedaço de felicidade que derrete na boca e aquece o coração!"



Figura 32 – Página Inicial

### 6.2.3. Tarte

Nesta página é exibida a informação de cada produto individualmente. Por outras palavras, depois de se seleccionar um produto do Menu de tartes, apresentado na página inicial, é exibida a sua informação como o nome, a imagem, o preço, uma breve descrição, o botão para adicionar esse produto ao carrinho de compras e o botão para regressar ao menu.

De modo semelhante à página inicial, esta página utiliza também um estilo css específico.

Os detalhes dos produtos como o nome, a imagem, o preço e a descrição são carregados dinamicamente através do *backend*, de acordo com o parâmetro no URL (*Uniform Resource Locator*). Isto permite que a aplicação apresente a informação específica do produto seleccionado.



Figura 33 – Página de Tarte

## 6.2.4. Carrinho

Esta página corresponde ao carrinho de compras. Se um utilizador aceder ao carrinho de compras sem a sessão iniciada irá ver uma mensagem que indica que deve iniciar sessão para aceder o carrinho, junto com um botão que o redireciona para a página de login. É possível aceder a esta página sempre que se carrega no separador “Carrinho de Compras” do menu horizontal no topo da página.



Figura 34 – Página de Carrinho de Compras

## 6.2.5. Área Pessoal

Esta página corresponde à área pessoal do cliente, na qual este pode visualizar os seus dados e até alterar os mesmos. Além disso, pode ver as suas encomendas e terminar a sessão na sua conta. Caso a sessão não esteja iniciada, o utilizador pode ainda fazer *login* na sua conta e, se não possuir conta, pode proceder à sua criação. É possível aceder a esta página sempre que se carrega no separador “Área Pessoal” do menu horizontal no topo da página.



Figura 35 – Página da Área Pessoal

## 6.2.6. Login

A página de *login* permite que os utilizadores já registados na plataforma se autenticuem, para aceder a todas as funcionalidades que necessitam de autenticação, tais como o seu carrinho e as suas encomendas. Caso o utilizador não possua conta na plataforma, tem um *link* onde será redirecionado para a página do registo. O utilizador também pode aceder à página de reposição de password caso se tenha esquecido da sua *password*. Os utilizadores podem aceder a esta página de login a partir da área pessoal, através do carrinho de compras ou dos favoritos caso a sua sessão não esteja iniciada.

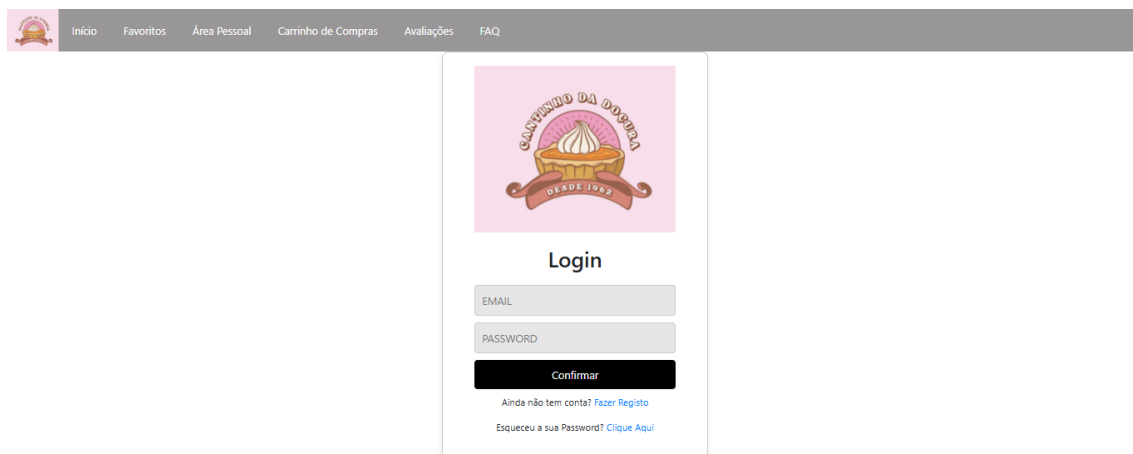


Figura 36 – Página de Login

## 6.2.7. Registo

A página de registo permite que os utilizadores que não estejam registados possam fazer o seu registo. Aqui o utilizador preenche os seus dados (Nome, Email, Telefone, Morada e *Password*) e no fim confirma que quer fazer o registo. Depois de efetuado o registo o utilizador pode fazer o *login*. Caso o utilizador tenha conta no sistema, pode voltar para a página de *login* clicando em “Fazer Login”. A página de registo pode ser acedida a partir da “Área Pessoal” ou então a partir da página de *login*.



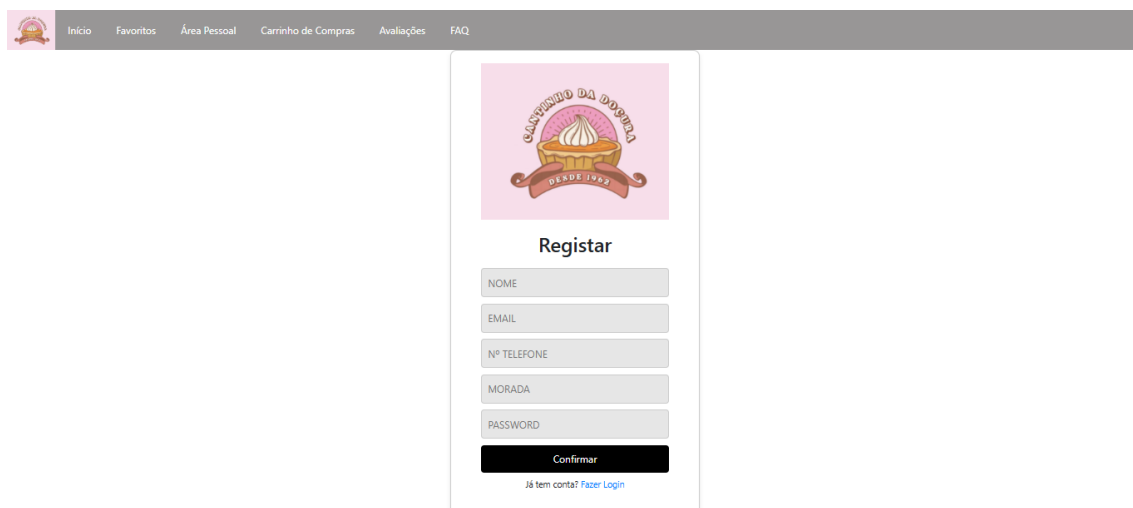


Figura 37 – Página de Registo

### 6.2.8. Recuperar Password

A página de Recuperar Password permite a um utilizador que se esqueceu da sua *password* definir uma nova a partir do seu email. A página de recuperar senha pode ser acedida a partir da página de *login*.

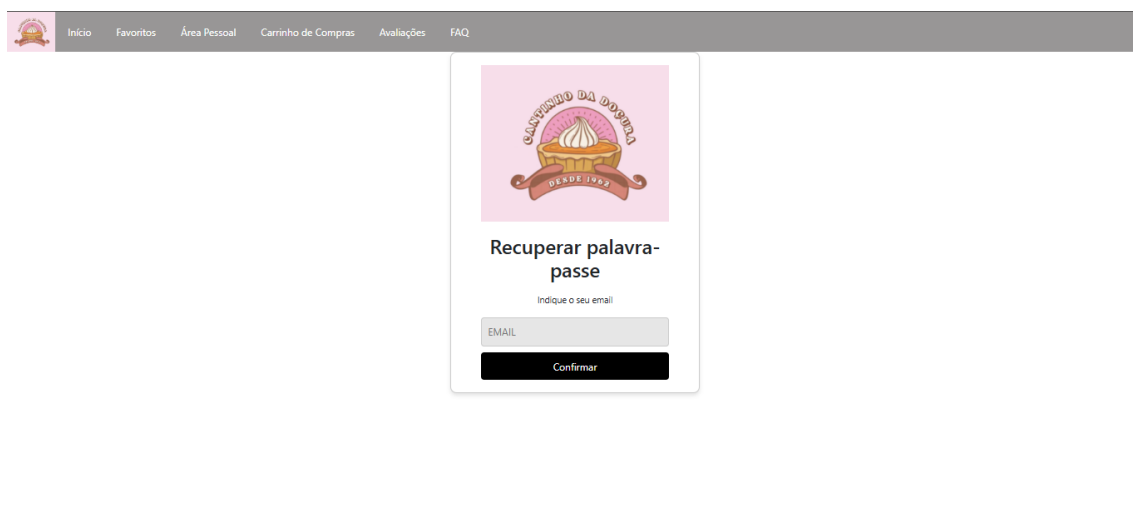
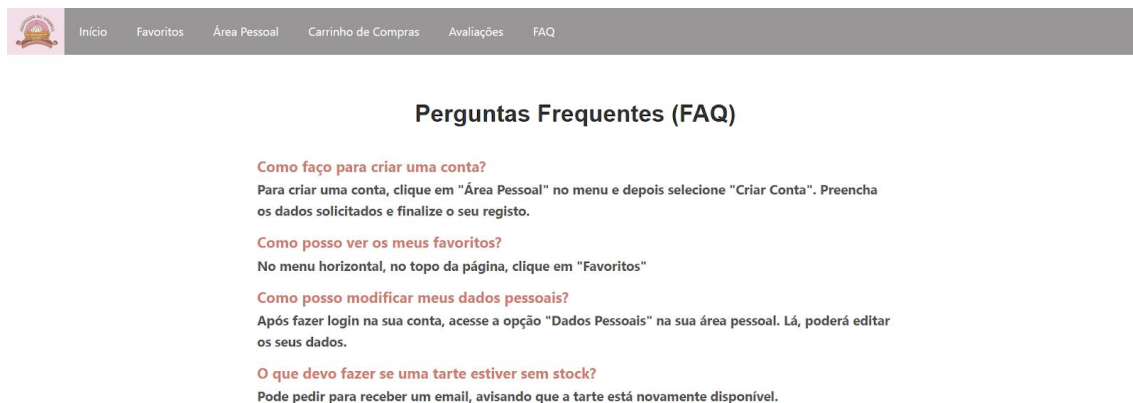


Figura 38 – Página de Recuperar Password

### 6.2.9. FAQ

A página de FAQ (*Frequently Asked Questions*) exhibe todas as perguntas e respostas mais comuns dos utilizadores sobre o funcionamento da aplicação. Utiliza também um estilo *css* específico. Além disso, tanto as perguntas como as respostas são carregadas para a aplicação dinamicamente através do *backend*.



*Figura 39 – Página de Perguntas Frequentes*

### 6.2.10. Favoritos

Esta página corresponde aos Favoritos. Se um utilizador aceder aos favoritos sem a sessão iniciada irá ver uma mensagem que indica que deve iniciar sessão para aceder os favoritos, junto com um botão que o redireciona para a página de login. É possível aceder a esta página sempre que se carrega no separador "Favoritos" do menu horizontal no topo da página.



*Figura 40 – Página de Favoritos*

### 6.2.11. Menu de Texto

Como referido anteriormente, o nosso programa possui 2 modos, o Blazor e o Menu de Texto. Nesta secção vamos apresentar o modo "Menu de Texto":

```
Building...
Bem vindo ao Cantinho das Encomendas!

*** Menu ***
1 - Iniciar Sessão
2 - Registrar
3 - Recuperar Palavra-Passe
0 - Sair
Opção: |
```

Figura 41 – Menu de Texto: Página Inicial

```
*** Menu ***
1 - Ver Produtos
2 - Ver Carrinho de Compras
3 - Adicionar Produto ao Carrinho
4 - Remover Produto do Carrinho
5 - Pagar Carrinho
6 - ---
7 - Ver Favoritos
8 - Adicionar Produto aos Favoritos
9 - Remover produto dos Favoritos
10 - Ver as minhas encomendas
11 - Cancelar uma encomenda
12 - Ver Preparação da Encomenda
13 - Ver Dados Pessoais
14 - Modificar Dados Pessoais
15 - Ver Avaliações
16 - Fazer Avaliação
17 - Ver FAQ
18 - Terminar Sessão
0 - Sair
Opção: |
```

Figura 42 – Menu de Texto: Menu Cliente

```
*** Menu ***
1 - Ver Stock
2 - ---
3 - Adicionar novo Material
4 - Remover Material existente
5 - Definir Quantidade de um Material
6 - Definir Quantidade Máxima de um Material
7 - ---
8 - Ver Produtos
9 - Adicionar Produto
10 - Remover Produtos
11 - ---
12 - Interromper Produção
13 - Ver uma Encomenda
14 - Ver Encomendas em espera
15 - Ver Encomendas a serem preparadas
16 - Ver Encomendas feitas
17 - ---
18 - Iniciar Produção de um Produto de uma Encomenda
19 - Atualizar Estado de Encomendas
20 - Atualizar Progresso de Encomendas
21 - ---
22 - Ver FAQ
23 - Adicionar FAQ
24 - Remover FAQ
25 - Terminar Sessão
0 - Sair
Opção: |
```

*Figura 43 – Menu de Texto: Menu Funcionário*

### 6.3. Análise e avaliação da aplicação desenvolvida

Tendo em conta as funcionalidades que tínhamos planeado para esta aplicação, consideramos que a aplicação desenvolvida não cumpre esses objetivos, principalmente na conexão da parte interativa à lógica de negócio. Contudo, acreditamos que a parte visual que realizamos da aplicação, ficou muito semelhante aos esboços feitos para as interfaces gráficas. Além disso, como já referido acima, em algumas páginas da aplicação fez-se o carregamento dinâmico de dados. À medida que novos dados são adicionados, como, por exemplo, uma nova FAQ ou um produto novo, os dados podem ser carregados sem ser necessário modificar a estrutura da página, o que contribui para uma maior escalabilidade e flexibilidade. Além disso, se for necessário alterar algum dado, o código da interface não é alterado, o que permite uma maior facilidade de manutenção.

Assim, apesar de termos o Menu de Texto, acreditamos que a aplicação não esteja a ideal para atender às necessidades do cliente.

## 7. Conclusões e Trabalho Futuro

### 7.1. Conclusões

Com a realização da primeira fase deste projeto, conseguimos aplicar os conhecimentos obtidos sobre desenvolvimento de software, seguindo a bibliografia da Unidade Curricular, neste projeto fictício chamado “Cantinho das Encomendas”.

Aprendemos como fazer a contextualização do nosso projeto, como elaborar e criar os requisitos funcionais e não funcionais para os nossos programas. Também conseguimos entender e realizar a especificação e modelação do software, indicando os aspetos estruturais e comportamentais do mesmo. Por fim, indicamos como o Sistema de Dados será usado na nossa aplicação e indicamos esboços das interfaces que o mesmo terá.

Acima de tudo, buscamos aplicar os conhecimentos obtidos nas outras Unidades Curriculares neste projeto, solidificando assim o nosso conhecimento.

Apesar de conseguirmos concluir todos os pontos, acreditamos que o nosso trabalho não correu de acordo com o planeado, tanto por divergências do funcionamento da aplicação, como por falta de conhecimento profundo sobre especificação e modelação de software, o que levou à sua constante alteração.

Além disso, este projeto forneceu-nos uma oportunidade de experienciar o planeamento de um projeto, algo que se revelará útil nas nossas futuras experiências profissionais.

Na segunda fase deste projeto, procedemos à implementação do sistema modelado anteriormente. Infelizmente, devido a diversos desafios, esta implementação ficou bastante aquém das nossas expectativas. As causas deste desempenho devem-se maioritariamente a um desleixo, da nossa parte, na organização do desenvolvimento do projeto e gestão do tempo que nos foi fornecido para a sua realização. No entanto, apesar destas dificuldades, esta segunda fase do projeto permitiu-nos a exploração de um novo ecossistema de desenvolvimento e novas ferramentas cruciais para o nosso futuro na área da programação e desenvolvimento de projetos de software. Assim sendo, planeamos continuar a explorar estas mesmas ferramentas em novos projetos.

Deste modo, apesar do resultado deste projeto, no que toca à sua implementação, não corresponder ao idealizado, este projeto e os conhecimentos nele adquiridos provaram-se cruciais para a nossa formação como desenvolvedores de projetos de *software*.

## 7.2. Trabalho Futuro

No futuro planeamos melhorar o sistema descrito neste documento, originando um software útil e totalmente funcional, mesmo sendo arquitetado para uma pastelaria fictícia.

Esperamos que o esforço gasto neste projeto não seja em vão pois com a sua realização pudemos aprender novos conceitos que irão ser uteis no nosso futuro.

## Referências

Sommerville, I. (2011). Software engineering (9th ed.). Addison-Wesley.

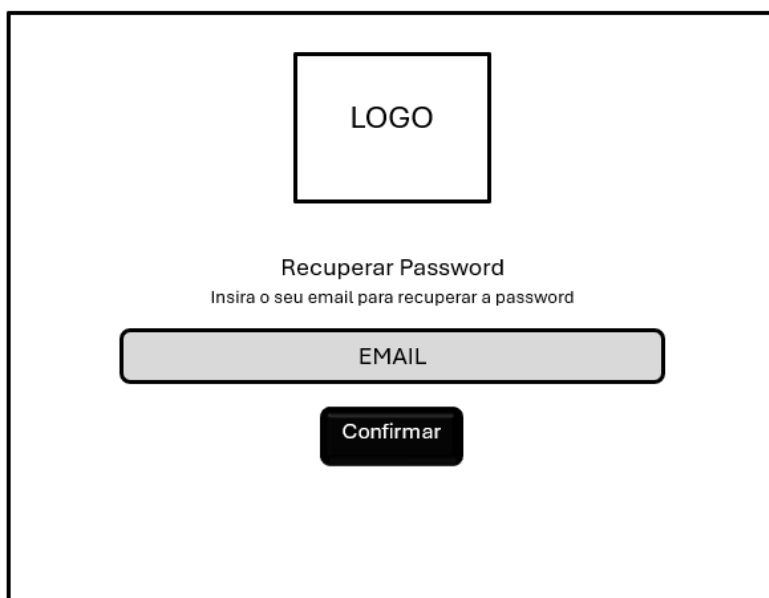
## Lista de Siglas

FAQ	<i>Frequently Asked Questions</i>
SQL	<i>Structured Query Language</i>
DAO	<i>Data Access Objects</i>
UI	<i>User Interface</i>



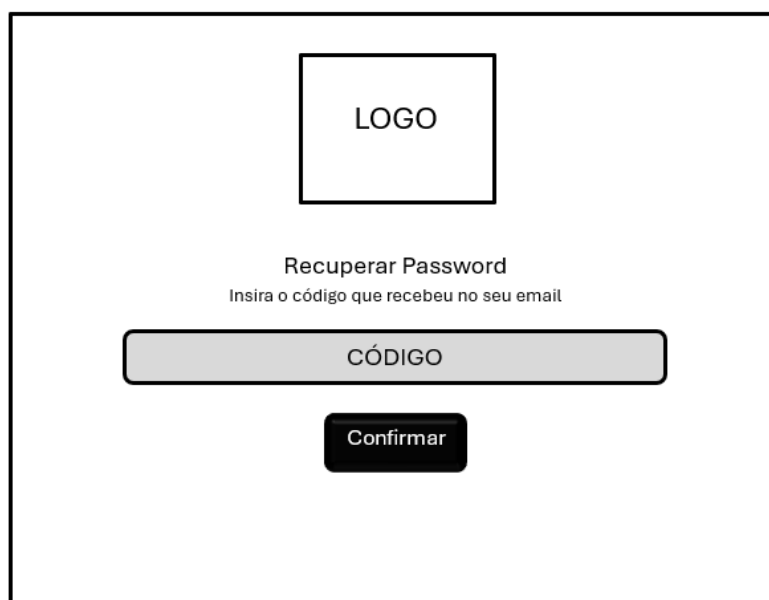
## Anexos

### Anexo 1 – Lista completa dos mockups do sistema



A rectangular frame containing a centered layout. At the top is a square box labeled "LOGO". Below it is the text "Recuperar Password" followed by the instruction "Insira o seu email para recuperar a password". Underneath is a wide, light-gray rounded rectangular input field labeled "EMAIL". At the bottom is a dark gray rounded rectangular button labeled "Confirmar".

*Figura 44 – Mockup de recuperar password (inserir email)*



A rectangular frame containing a centered layout. At the top is a square box labeled "LOGO". Below it is the text "Recuperar Password" followed by the instruction "Insira o código que recebeu no seu email". Underneath is a wide, light-gray rounded rectangular input field labeled "CÓDIGO". At the bottom is a dark gray rounded rectangular button labeled "Confirmar".

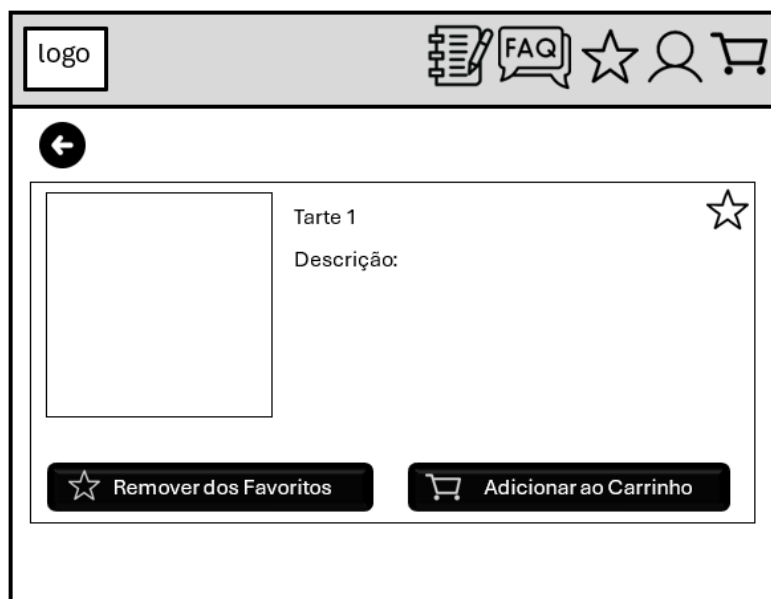
*Figura 45 – Mockup de recuperar password (inserir código de confirmação)*

A mockup of a password recovery screen. At the top center is a square box labeled "LOGO". Below it, the text "Recuperar Password" is centered, followed by the instruction "Insira a nova password". Underneath is a long, light gray rounded rectangular input field labeled "NOVA PASSWORD". At the bottom center is a dark gray rounded rectangular button labeled "Confirmar".

Figura 46 – Mockup de recuperar password (inserir nova password)

A mockup of a tart product page. The top header bar is gray and contains a "logo" box on the left and a row of icons (notepad, FAQ, star, person, shopping cart) on the right. Below the header is a white content area. On the left side of this area is a circular back arrow icon. To its right is a large rectangular placeholder for an image. To the right of the image placeholder, the text "Tarte 1" is displayed, followed by "Descrição:". At the bottom of the content area are two dark gray buttons: the first has a star icon and the text "Adicionar aos Favoritos", and the second has a shopping cart icon and the text "Adicionar ao Carrinho".

Figura 47 – Mockup da página da tarte



*Figura 48 – Mockup da tarte nos favoritos*



*Figura 49 – Mockup da tarte sem stock*

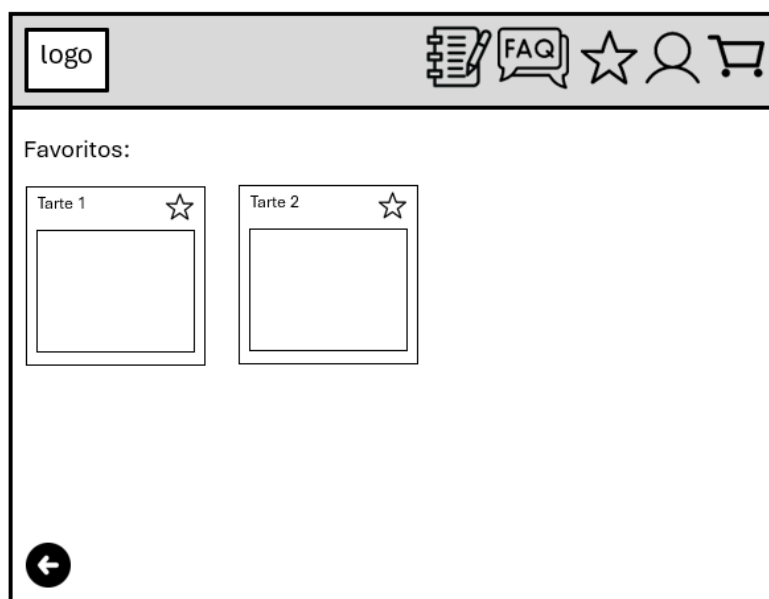


Figura 50 – Mockup dos favoritos

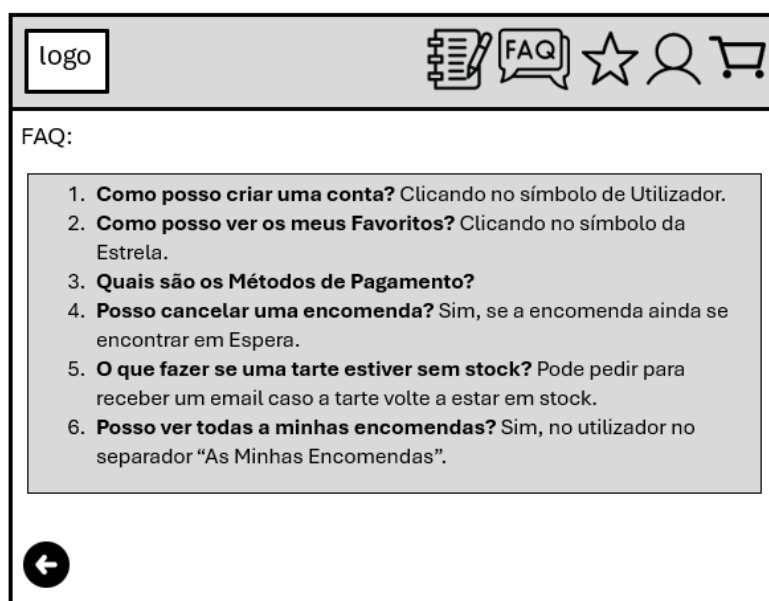


Figura 51 – Mockup da aba de Perguntas Frequentes

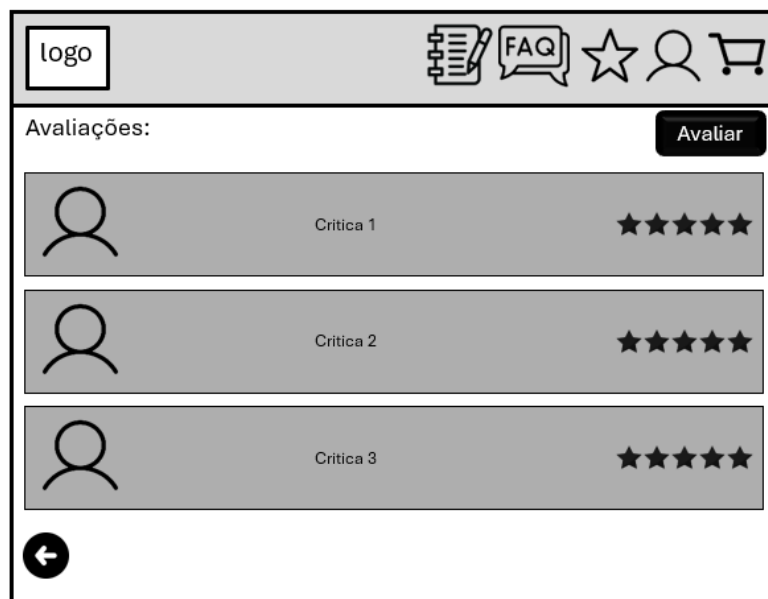


Figura 52 – Mockup de ver avaliações

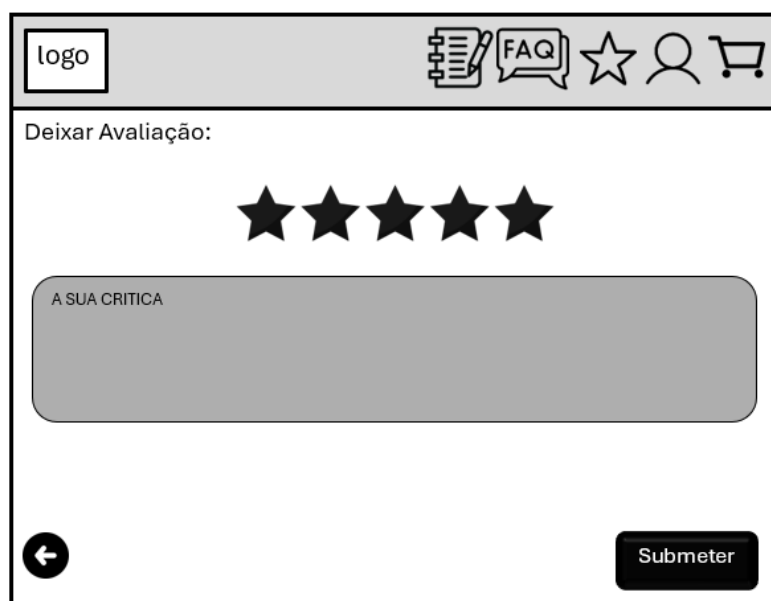


Figura 53 – Mockup de deixar avaliação

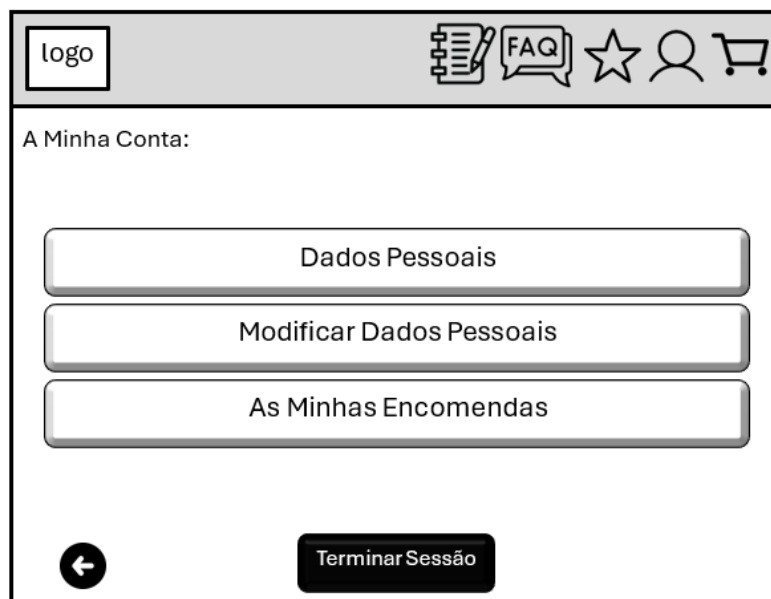


Figura 54 – Mockup de “A Minha Conta”

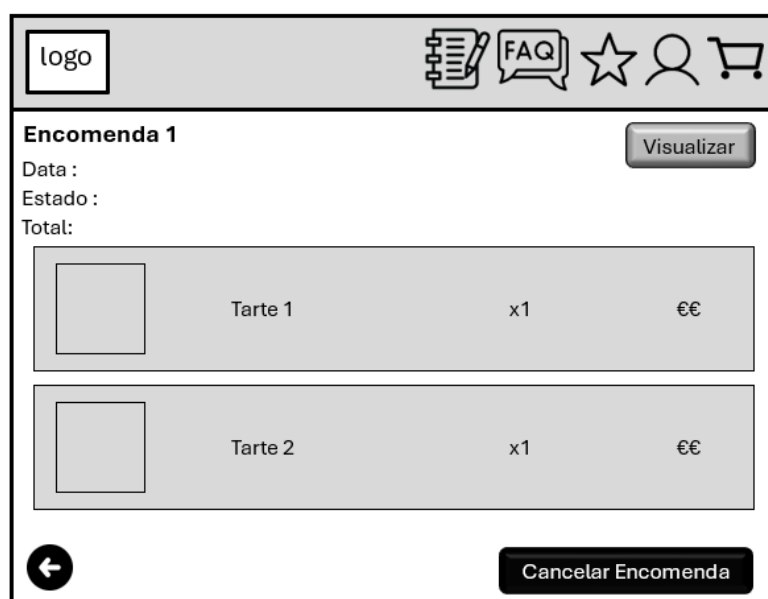


Figura 55 – Mockup da encomenda



Figura 56 – Mockup de terminar sessão



Figura 57 – Mockup de encomendas em fila de espera

logo

**Encomenda 1**

Atualizar Estado

Data :

Estado : Fila de Espera

Total:

Tarte 1

x1

€€

Tarte 2

x1

€€

←

Figura 58 – Mockup de encomenda em fila de espera

logo

**Encomendas em Produção:**

Filtrar Por

Encomenda Nº :	Data:	Estado:	
1	20/05/2023	Em Preparação	Q
2	11/03/2024	Em Preparação	Q
3	30/09/2024	Em Preparação	Q
4	20/11/2024	Em Preparação	Q

Interromper Produção

Retomar Produção

←

Figura 59 – Mockup de encomendas em preparação





Figura 60 – Mockup de encomenda em preparação

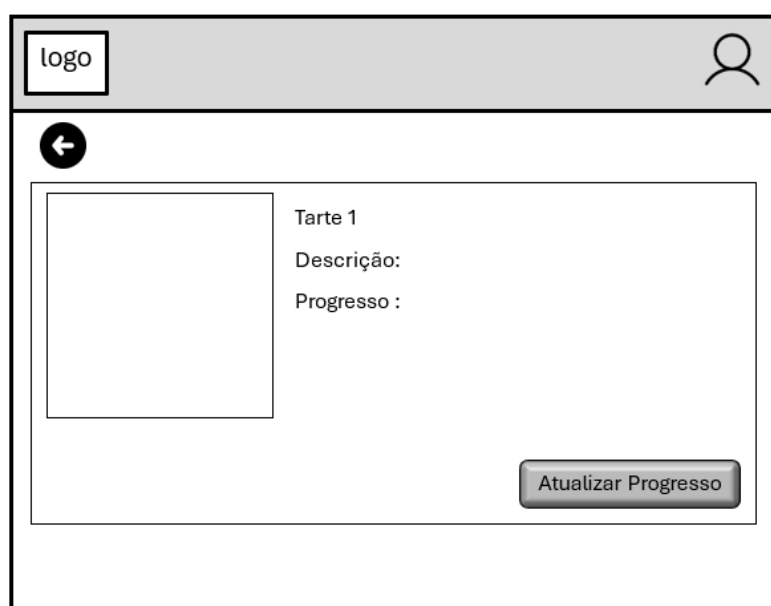


Figura 61 – Mockup de alterar progresso da tarte

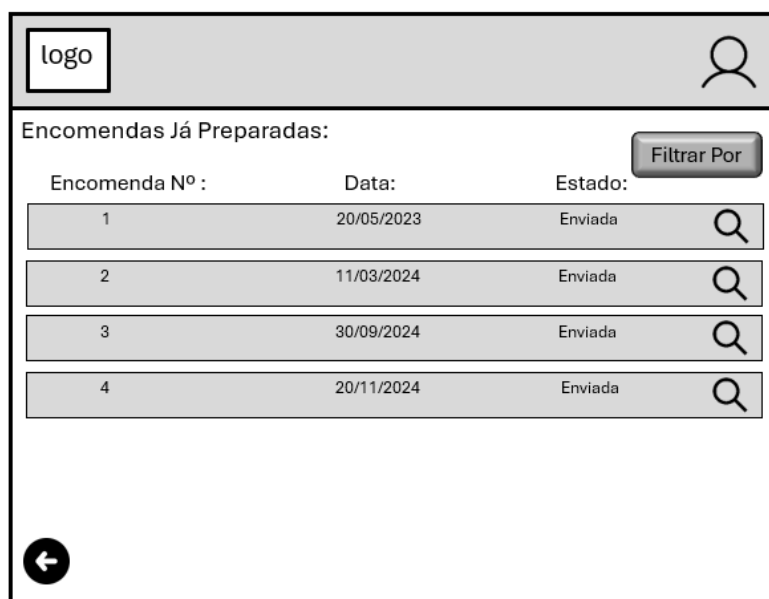


Figura 62 – Mockup de encomendas enviadas

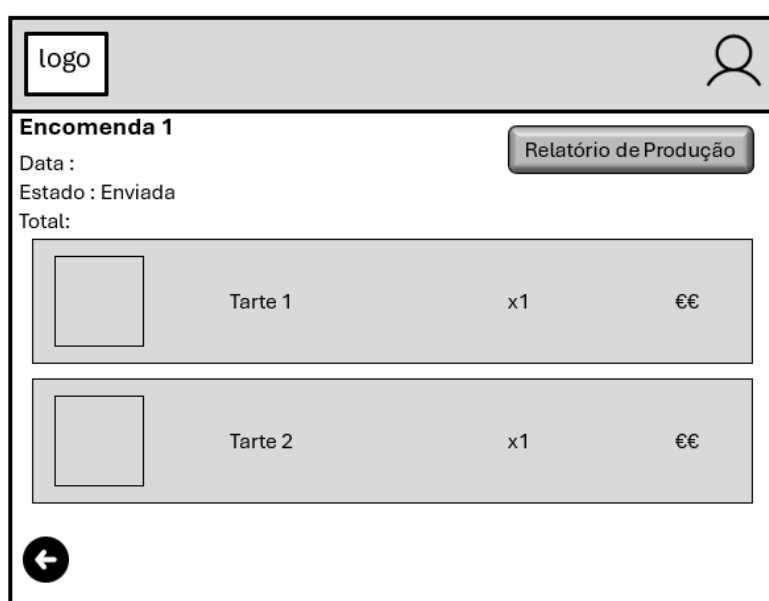
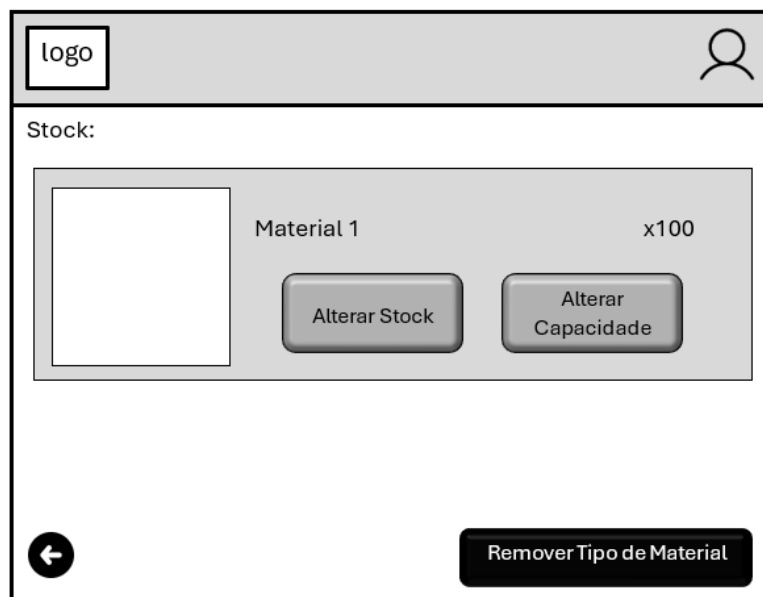


Figura 63 – Mockup de encomenda enviada



*Figura 64 – Mockup de stock de um material*



*Figura 65 – Mockup de alterar stock*

Alterar Capacidade:

Material 1 x100

NOVA CAPACIDADE

← Confirmar

*Figura 66 – Mockup de alterar capacidade de stock*

Adicionar Tipo de Material:

NOME

STOCK

CAPACIDADE

← Confirmar

*Figura 67 – Mockup de adicionar tipo de material*

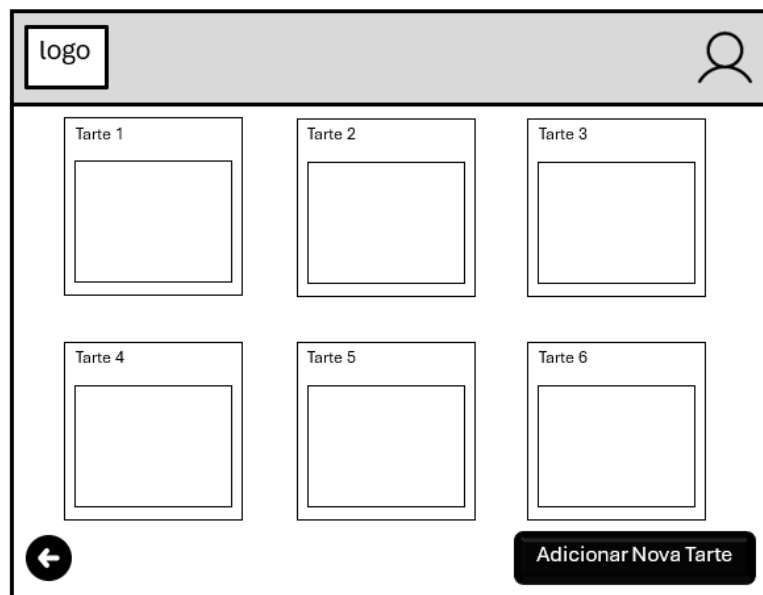


Figura 68 – Mockup de ver tartes disponíveis - funcionário

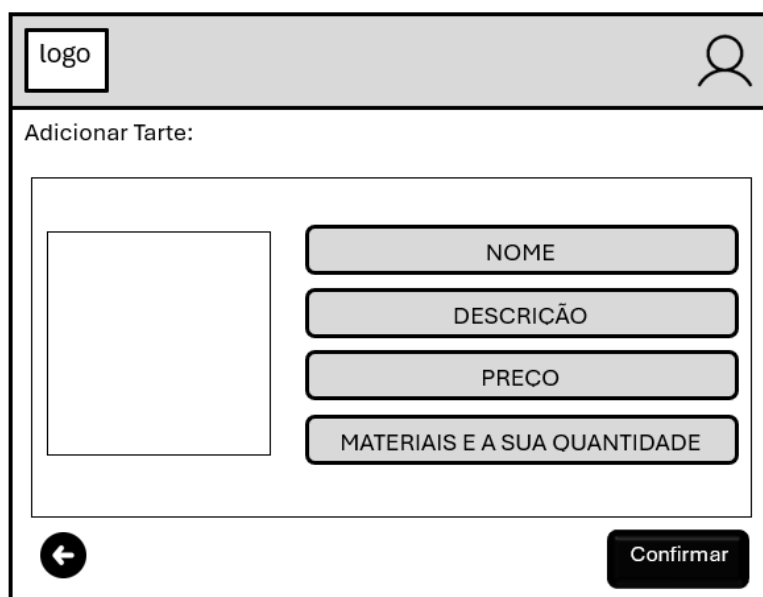


Figura 69 – Mockup de adicionar tarte

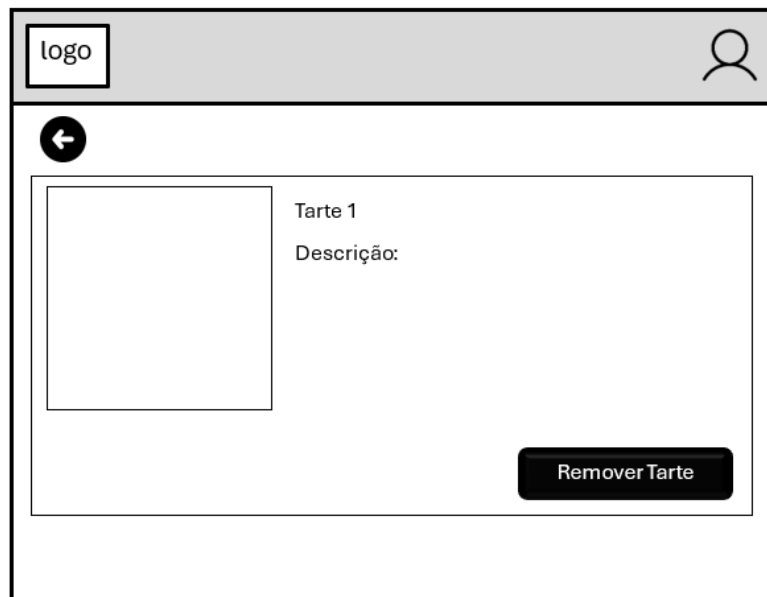


Figura 70 – Mockup de remover tarte

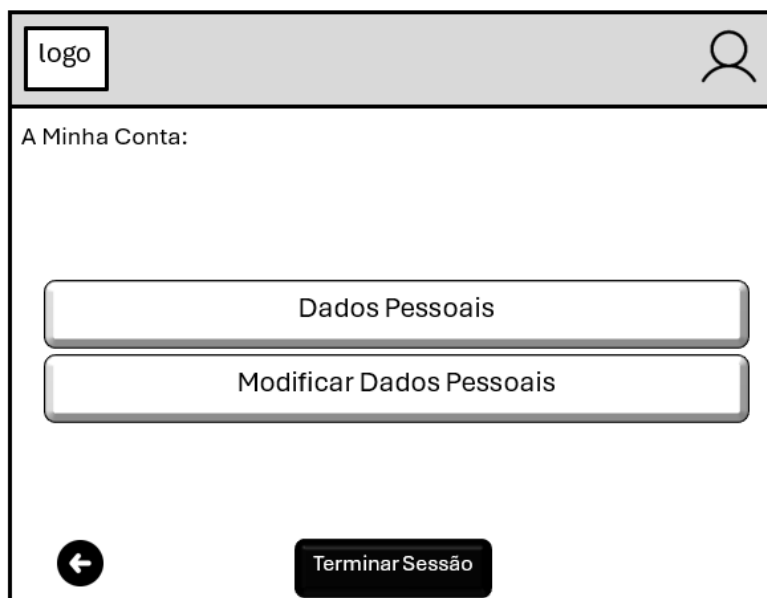


Figura 71 - Mockup “A Minha Conta” - Funcionário