

## **Licenciatura em Engenharia Informática**

Escola Superior de Tecnologia e Gestão

*Instituto Politécnico de Viana do Castelo*

Unidade curricular de

# **Projeto I**

Relatório de projeto

2021/2022

*Gestão de uma loja pronto-a-vestir*

*(Encomenda on-line + Gestão Stock)*

Carlos Macho nº22359

Bruno Gigante nº24584

## Sumário

<b>1.</b>	<b>Introdução.....</b>	<b>3</b>
1.1	Apresentação do tema .....	3
1.2	Objetivos do projeto .....	3
<b>2.</b>	<b>Apresentação do negócio.....</b>	<b>3</b>
2.1	Âmbito e enquadramento do projeto .....	3
2.2	Modelação dos Processos de negócio .....	4
2.2.1	Processo 1 - Gestão de Stock.....	4
2.2.2	Processo 2 – Venda a cliente .....	6
<b>3.</b>	<b>Levantamento de Requisitos.....</b>	<b>8</b>
3.1	Tipos de utilizador.....	8
3.1.1	Apresentação dos tipos de utilizador.....	8
3.2	Requisitos Funcionais .....	9
3.3	Requisitos não funcionais .....	10
<b>4.</b>	<b>Design e Modelação .....</b>	<b>11</b>
4.1	Modelo de casos de uso.....	11
4.1.1	Descrição dos casos de uso .....	12
4.2	Templates dos Casos de Uso .....	13
4.3	Modelo de classes .....	18
4.3.1	Descrição do modelo de classes .....	19
4.4	Diagrama Entidades Relacionamento/ Modelo relacional.....	20
4.5	Diagramas de atividades.....	22
4.6	Diagramas de sequência.....	28
4.7	Diagramas de transição de estados .....	30
<b>5.</b>	<b>Implementação da BD.....</b>	<b>32</b>
	Scripts de criação e triggers da base de dados .....	33
	Inserts na base de dados .....	54
	Consultas na base de dados .....	55
	Trigger na base de dados .....	57
<b>6.</b>	<b>Conclusão e Trabalho Futuro .....</b>	<b>58</b>

## 1. Introdução

### 1.1 Apresentação do tema

Este relatório foi desenvolvido no âmbito da Unidade Curricular (UC) de Projeto I, sendo proposta pelos docentes da mesma, António Cruz e Estrela Cruz.

O projeto fundamenta-se na elaboração de diferentes modelos, tais como o de negócio, SW e de dados, juntamente com a implementação da base de dados adequada, que permitirá um funcionamento eficiente do tema proposto.

Objetivo do nosso tema (Loja pronto-a-vestir) é fazer a sua **gestão de stock** de uma forma eficiente e eficaz, inclusive efetuar **encomendas aos fornecedores**, para poder atender aos ciclos de vida (moda) e atender às preferências de vestuário sazonais dos nossos clientes.

### 1.2 Objetivos do projeto

Nesta primeira fase, o objetivo do projeto é a identificação dos diferentes modelos de negócio presentes neste tema, sendo estes desenvolvidos por modelos BPMN.

## 2. Apresentação do negócio

Nesta secção iremos apresentar o modelo de negócio do tema e os diferentes processos de negócio identificados e necessários para o melhor funcionamento da gestão da loja pronto-a-vestir.

### 2.1 Âmbito e enquadramento do projeto

Para a nossa loja pronto-a-vestir poder atender aos seus ciclos de vida/moda, necessita de encomendar o vestuário sazonal para a loja, logo contacta os

fornecedores para efetuar essa encomenda. Se a encomenda é feita com sucesso, então os fornecedores enviam a encomenda para a loja. De seguida, stock da loja é atualizado com o novo vestuário obtido. Depois da atualização do stock, é possível informar clientes que previamente pediram para ser contactados de produtos que estavam indisponíveis. Por fim a loja efetua assim as vendas de todo o material existente/disponível até o mesmo entrar em pouco stock ou em rutura de stock, levando assim a loja ter que repetir todo este processo de aquisição de stock para venda a clientes.

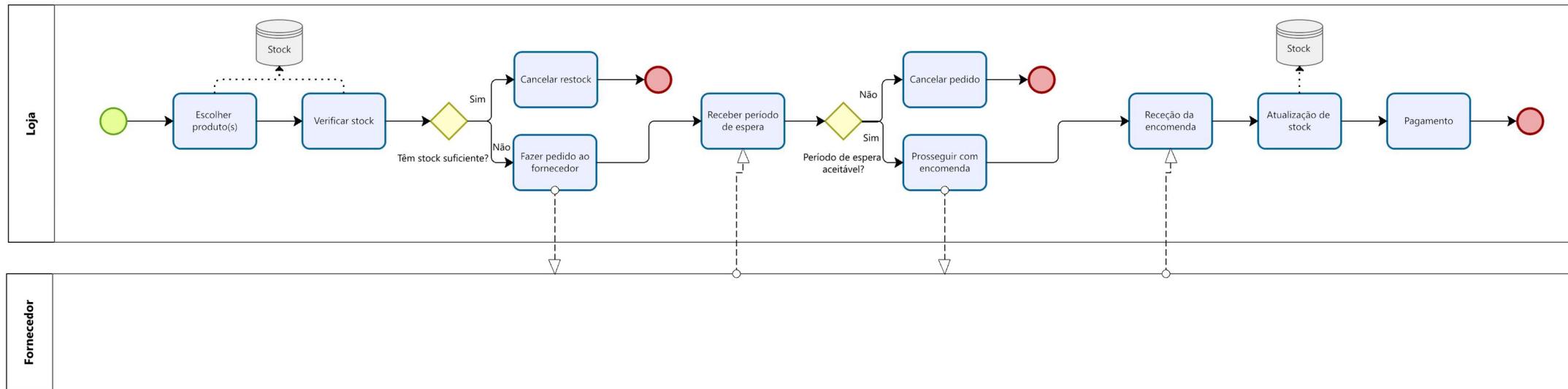
## 2.2 Modelação dos Processos de negócio

Nesta secção iremos apresentar detalhadamente os processos de negócio formulados.

### 2.2.1 Processo 1 - Gestão de Stock

Neste primeiro processo, tratamos de solucionar todo o processo que envolve a reposição de stock entre a loja e os fornecedores.

Inicialmente começamos por ver que produtos a loja precisa de repor e para tal a loja faz uma verificação do stock que possui atualmente e através dessa verificação conclui que produtos precisam de ser repostos. Assim que determinado os produtos a serem repostos, a loja faz um pedido de verificação de disponibilidade desses produtos com os fornecedores, se não estiverem disponíveis, a loja pode optar por cancelar o pedido de reposição. No entanto, se estiver disponível, a loja prossegue com o pedido de reposição de stock. Após finalizado o pedido, a loja aguarda que os produtos encomendados sejam entregues á loja. Assim que tudo foi entregue, a loja atualiza o stock com os novos produtos obtidos e de seguida prossegue ao pagamento de todos os produtos encomendados ao fornecedor, terminando aqui o processo da reposição de stock.



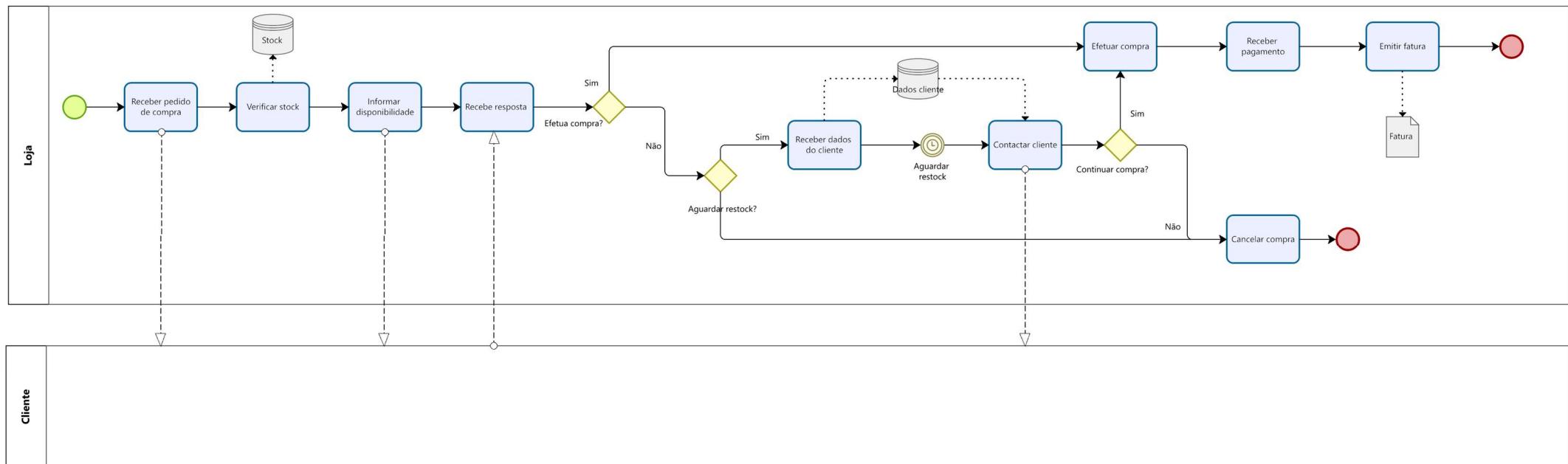
**FIGURA 1 - GESTÃO DE STOCK (MODELO BPMN)**

Powered by

## 2.2.2 Processo 2 – Venda a cliente

Neste segundo processo de negócio o objetivo é identificar todas as situações possíveis e como proceder face cada uma, na venda de um produto ao cliente.

Para tal, a loja começa por receber um pedido de compra do cliente que será informado da disponibilidade do produto que ele quer após ser verificado o stock. Se existir stock o cliente prossegue para o pagamento do produto e a loja emite fatura do mesmo, terminando assim esta interação. Caso não exista stock do produto o cliente pode cancelar a sua compra ou aguardar pela reposição do stock para que possa efetuar a compra mais tarde. Caso pretenda aguardar pela reposição do stock a loja fica com a informação de contacto do cliente e assim que chegar o novo stock o cliente é informado e tomará a decisão de comprar ou cancelar a compra. Caso decida prosseguir com a compra será efetuado o pagamento e a loja emitirá a fatura do produto, terminando assim o processo.



**FIGURA 2 - VENDA A CLIENTE (MODELO BPMN)**

Powered by

### 3. Levantamento de Requisitos

Nesta secção vão ser apresentados detalhadamente todos os requisitos levantados para o funcionamento do software.

#### 3.1 Tipos de utilizador

Para o levantamento dos requisitos do projeto forma identificados os seguintes tipos de utilizadores:

- Gestor;
- Funcionário.

##### 3.1.1 Apresentação dos tipos de utilizador

Antes de fazer o levantamento de requisitos do projeto, foram identificados dois tipos de utilizadores: O gestor e o funcionário.

O gestor será aquele que para além de fazer tudo o que um funcionário faz, realizará tarefas relacionadas com a gestão do stock, nomeadamente efetuar encomendas a fornecedores, e todo o tipo ações que estejam relacionadas com os fundos monetários da empresa.

O funcionário será o membro que estará presente em todas as interações com o cliente, efetuando assim ações relacionadas com vendas a clientes, verificação de stock, entre outras...

### 3.2 Requisitos Funcionais

Requisitos Funcionais		
Referência	Descrição	Prioridade
RF01	Como gestor, quero encomendar vestuário aos fornecedores, para gerir o stock.	Alta
RF02	Como gestor, quero fazer um pedido de cancelamento de uma encomenda, para gerir o stock.	Alta
RF03	Como gestor, quero fazer alterações editar produtos no, para gerir o stock.	Alta
RF04	Como gestor, quero remover produtos do sistema, para gerir o stock.	Alta
RF05	Como gestor, quero adicionar produtos ao sistema, para gerir o stock.	Alta
RF06	Como gestor, quero listar a faturação total de certo dia/mês, para gerir os fundos monetários da loja.	Média
RF07	Como gestor, quero fazer o fecho do dia/mês, para gerir a faturação.	Alta
RF08	Como gestor, quero fazer o pagamento da encomenda dos produtos, para gerir as dívidas.	Alta
RF09	Como funcionário, quero emitir uma fatura/recibo sempre que um cliente fizer uma compra, para gerir a faturação.	Alta
RF10	Como funcionário, ser notificado sempre que um produto ficar sem disponibilidade, para gerir o stock.	Baixa
RF11	Como funcionário, quero listar o stock existente, para verificar o stock disponível.	Alta
RF12	Como funcionário, quero guardar as informações de um cliente, para notificar o mesmo da disponibilidade de um produto pedido pelo cliente.	Média
RF13	O sistema deve alertar o funcionário quando uma peça desejada por um cliente voltou a entrar em stock.	Baixa

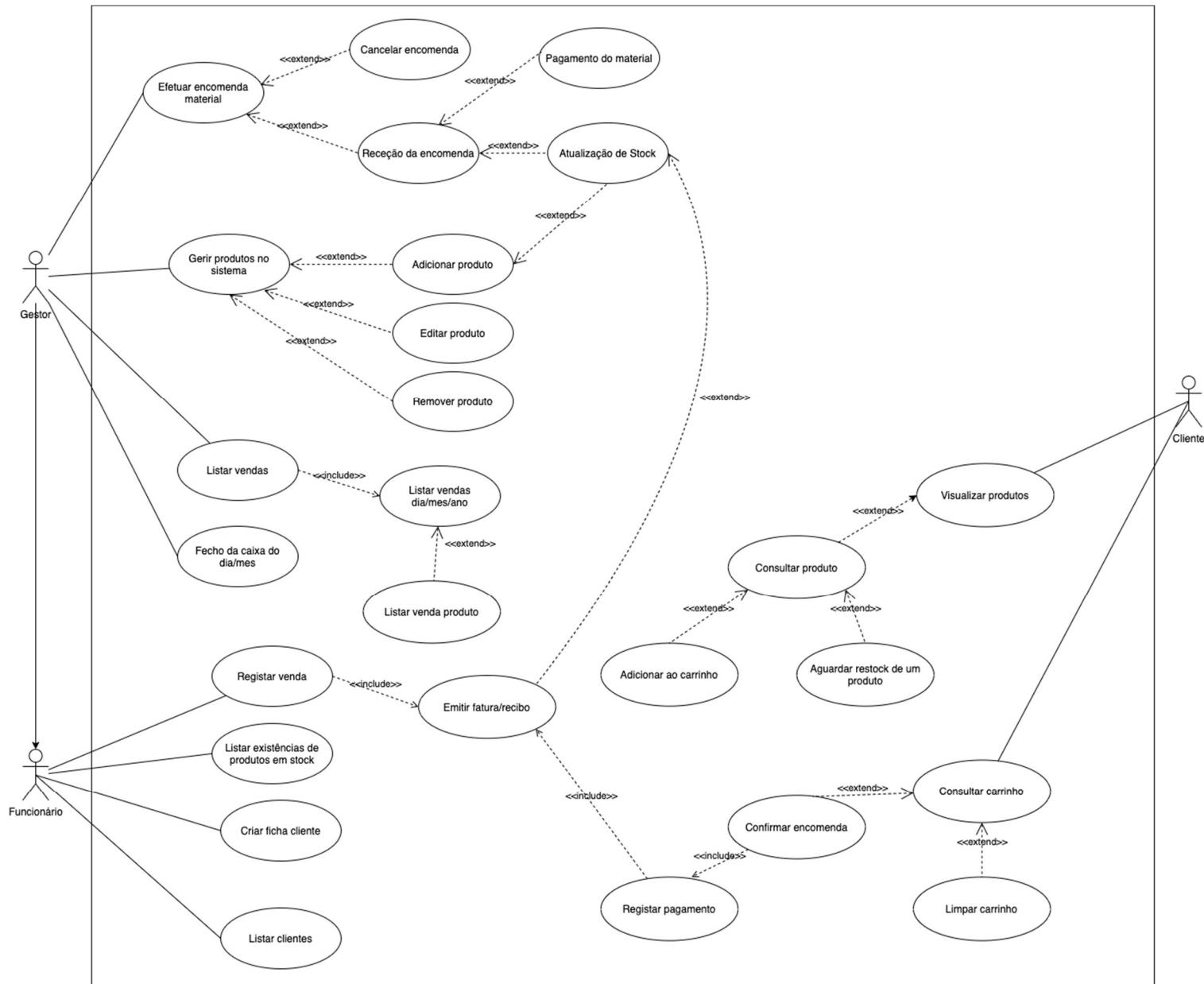
### 3.3 Requisitos não funcionais

Requisitos Não Funcionais	
Referência	Descrição
RNF01	O Software deverá ser desenvolvido na linguagem java.
RNF02	O Software deverá comunicar com a base de dados <i>Oracle Database</i> .

## 4. Design e Modelação

Nesta secção irão ser apresentados os diferentes modelos/ diagramas referentes ao trabalho proposto.

### 4.1 Modelo de casos de uso



**FIGURA 3 - DIAGRAMA DE CASOS DE USO**

#### 4.1.1 Descrição dos casos de uso

**Efetuar encomenda material:** Para efetuar uma encomenda a um fornecedor o gestor escolhe o(s) item(s) a encomendar e efetua a devida encomenda. Caso o mesmo decida pode ainda cancelar a encomenda. Após a receção da mesma o stock é atualizado.

**Gerir produtos no sistema:** O gestor pode gerir todos os produtos no sistema incluindo diversas ações como adicionar novos produtos, editar produtos existentes e remover produtos.

**Listar vendas:** A listagem de vendas permite visualizar todas as vendas realizadas assim como apenas as vendas de um determinado dia, mês ou ano. É ainda possível listar as vendas efetuadas de um certo produto.

**Fecho da caixa do dia/mês:** O gestor ter permissão para fechar a caixa daquele dia ou daquele mês, isto é, esta ação permite ao gestor decidir quando para de vender num certo dia e ainda ver o total faturado nesse dia ou nesse mês.

**Registrar venda:** Um funcionário que efetue uma venda de um ou mais produtos regista no sistema essa venda para que o stock possa ser atualizado.

**Emitir fatura/recibo:** Sempre que é efetuada uma venda quer online quer em loja física será emitida uma fatura/recibo para dar saída de stock dos produtos vendidos e para controlar a faturação.

**Listar existências de produtos em stock:** Um funcionário tem acesso a uma listagem de todos os produtos existentes em stock.

**Criar ficha cliente:** Um funcionário pode criar uma ficha de cliente onde serão inseridos dados pessoais do cliente.

**Listar clientes:** Um funcionário pode listar todos os clientes existentes no sistema, ou seja, verificar que clientes já têm ficha criada, para saber se é necessário ou não a criação de uma ficha para um certo cliente.

**Visualizar produtos:** Um cliente pode visualizar todos os produtos disponíveis online e não apenas em loja física sendo que pode também consultar detalhes daquele produto. Caso decida comprar um produto que esteja a ver pode sempre adicionar ao carrinho o mesmo e em caso de não haver stock desse produto pode aguardar pela reposição do stock.

**Consultar carrinho:** Um cliente pode consultar o seu carrinho de modo a ver todos os produtos existentes no mesmo. Caso queira pode também limpar o carrinho removendo assim todos os itens do mesmo. Caso contrário poderá confirmar a sua compra efetuando assim uma encomenda.

**Registrar pagamento:** Sempre que é efetuada uma venda é registado o pagamento para possibilitar a emissão de uma fatura/recibo.

#### 4.2 Templates dos Casos de Uso

##### Casos de uso referentes apenas ao gestor

Id:	Caso de Uso: Efetuar Encomenda Material
Actor Principal:	Gestor
Pré-Condições:	O gestor deve estar autenticado no sistema
Pós-Condições:	
Cenário Principal	<ol style="list-style-type: none"> <li>1. O gestor seleciona os produtos a encomendar.</li> <li>2. Verifica disponibilidade do(s) fornecedor(es).</li> <li>3. O gestor confirma a encomenda.</li> </ol>
Cenários Alternativos	<ol style="list-style-type: none"> <li>1. O gestor pretende cancelar a encomenda.</li> <li>2. O gestor confirma a encomenda.</li> <li>3. O gestor inicia <b>Cancelar encomenda</b>.</li> </ol>
Excepções	

Id:	Caso de Uso: Gerir produtos no sistema
Actor Principal:	Gestor
Pré-Condições:	O gestor deve estar autenticado no sistema
Pós-Condições:	
Cenário Principal	<ol style="list-style-type: none"> <li>1. O gestor lista os produtos no sistema.</li> <li>2. O gestor gere os produtos no sistema.</li> </ol>
Cenários Alternativos	<ol style="list-style-type: none"> <li>1. O gestor pretende adicionar um produto.</li> <li>2. O gestor inicia <b>Adicionar produto</b>.</li> <li>3. É iniciado <b>Atualização Stock</b>.</li> </ol> <ol style="list-style-type: none"> <li>1. O gestor pretende editar um produto.</li> <li>2. O gestor inicia <b>Editar produto</b>.</li> </ol> <ol style="list-style-type: none"> <li>1. O gestor pretende remover um produto.</li> </ol>

	2. O gestor inicia <b>Remover Produto</b> . 3. É iniciado <b>Atualização Stock</b> .
Excepções	

Id:	Caso de Uso: Listar vendas
Actor Principal:	Gestor
Pré-Condições:	O gestor deve estar autenticado no sistema
Pós-Condições:	
Cenário Principal	1. O gestor lista as vendas. 2. O gestor filtra a listagem das vendas por dia/mes/ano.
Cenários Alternativos	1. O gestor pretende visualizar as vendas de um produto. 2. O gestor inicia <b>Listar venda produto</b> .
Excepções	

Id:	Caso de Uso: Fecho da caixa do dia/mês
Actor Principal:	Gestor
Pré-Condições:	O gestor deve estar autenticado no sistema
Pós-Condições:	
Cenário Principal	1. O gestor efetua o fecho da caixa daquele dia/mês. 2. É impresso o total de faturação desse período.
Cenários Alternativos	
Excepções	

## Casos de uso referentes ao funcionário e gestor

Id:	Caso de Uso: Registar Venda
Actor Principal:	Gestor, Funcionário
Pré-Condições:	O utilizador deve estar autenticado no sistema
Pós-Condições:	É emitida uma fatura ou recibo da venda e o stock é atualizado.
Cenário Principal	<ol style="list-style-type: none"> <li>1. O funcionário/gestor seleciona o(s) produto(s) a vender.</li> <li>2. O funcionário/gestor efetua o registo da venda no sistema.</li> <li>3. É iniciado <b>Registar pagamento</b>.</li> </ol>
Cenários Alternativos	
Excepções	

Id:	Caso de Uso: Listar existências de produtos em stock
Actor Principal:	Gestor, Funcionário
Pré-Condições:	O utilizador deve estar autenticado no sistema
Pós-Condições:	
Cenário Principal	<ol style="list-style-type: none"> <li>1. O sistema apresenta uma lista de produtos e o seu stock.</li> </ol>
Cenários Alternativos	
Excepções	

Id:	Caso de Uso: Criar ficha cliente
Actor Principal:	Gestor, Funcionário
Pré-Condições:	O utilizador deve estar autenticado no sistema
Pós-Condições:	É criada uma nova ficha de cliente.
Cenário Principal	<ol style="list-style-type: none"> <li>1. O utilizador abre o formulário da ficha de cliente.</li> <li>2. O utilizador preenche o mesmo com os dados do cliente.</li> </ol>
Cenários Alternativos	
Excepções	

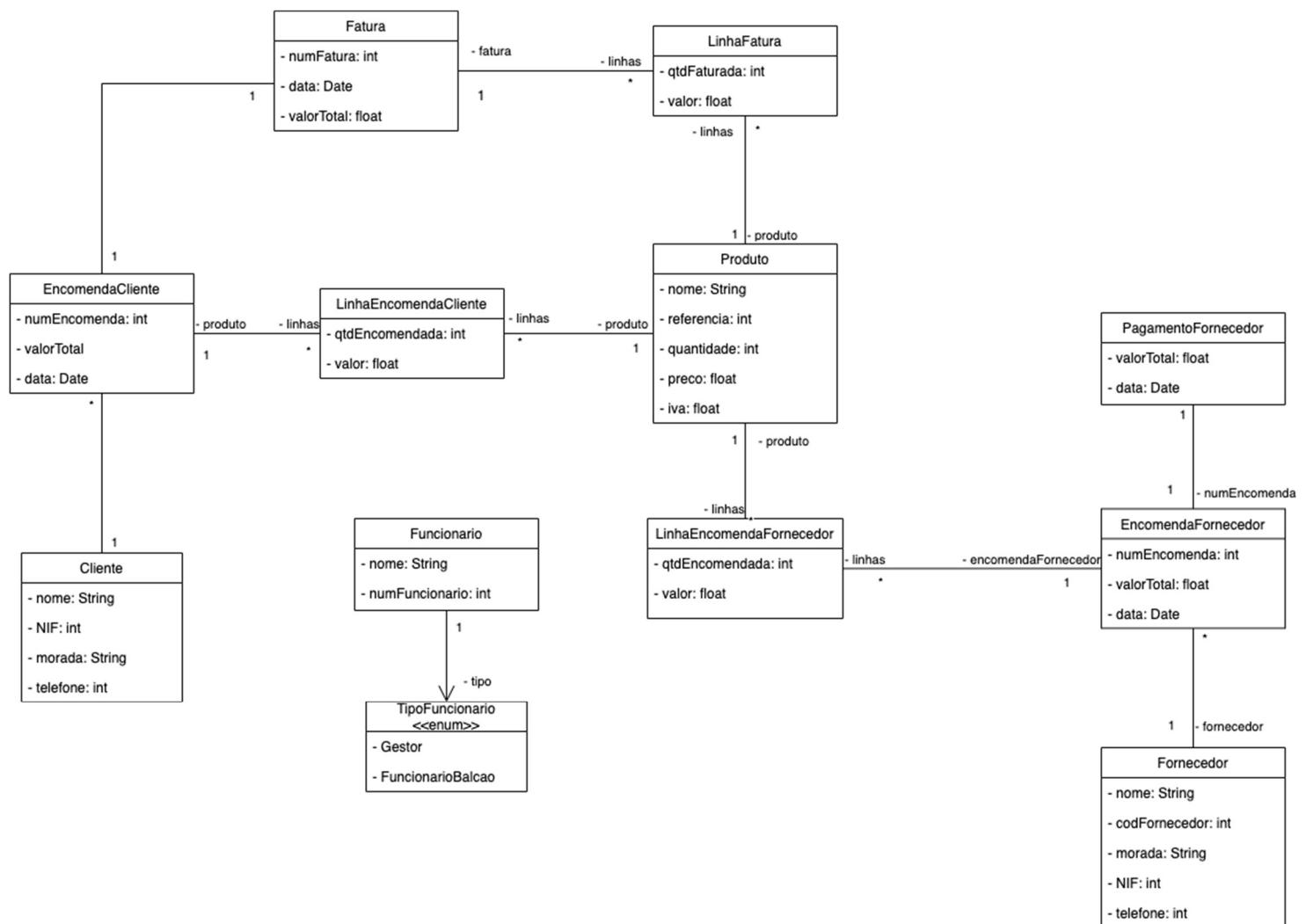
Id:	Caso de Uso: Listar clientes
Actor Principal:	Gestor, Funcionário
Pré-Condições:	O utilizador deve estar autenticado no sistema
Pós-Condições:	
Cenário Principal	<ol style="list-style-type: none"> <li>1. O sistema apresenta uma lista de clientes com ficha criada no sistema.</li> </ol>
Cenários Alternativos	<ol style="list-style-type: none"> <li>1. O utilizador visualiza a lista de clientes.</li> <li>2. O cliente não está registado no sistema.</li> <li>3. O utilizador inicia <b>Criar ficha cliente</b>.</li> </ol>
Excepções	

### Casos de uso referentes ao cliente

Id:	Caso de Uso: Visualizar produtos
Actor Principal:	Cliente
Pré-Condições:	
Pós-Condições:	
Cenário Principal	<ol style="list-style-type: none"> <li>1. O cliente visualiza os produtos disponíveis da loja.</li> <li>2. O cliente consulta os detalhes de certo produto.</li> </ol>
Cenários Alternativos	<ol style="list-style-type: none"> <li>1. O cliente pretende comprar o produto.</li> <li>2. O cliente adiciona ao carrinho.</li>   <li>1. O cliente pretende comprar o produto.</li> <li>2. O produto está sem stock.</li> <li>3. O cliente pretende aguardar restock.</li> </ol>
Excepções	

Id:	Caso de Uso: Consultar carrinho
Actor Principal:	Cliente
Pré-Condições:	
Pós-Condições:	
Cenário Principal	<ol style="list-style-type: none"> <li>1. O cliente adicionou produtos ao carrinho.</li> <li>2. O cliente consulta o seu carrinho.</li> </ol>
Cenários Alternativos	<ol style="list-style-type: none"> <li>1. O cliente consulta o seu carrinho.</li> <li>2. O cliente pretende confirmar encomenda.</li> <li>3. É iniciado <b>Registar pagamento</b>.</li>   <ol style="list-style-type: none"> <li>1. O cliente consulta o seu carrinho.</li> <li>2. O cliente decide limpar o seu carrinho.</li> </ol> </ol>
Excepções	

### 4.3 Modelo de classes



**FIGURA 4 - DIAGRAMA DE CLASSES**

#### 4.3.1 Descrição do modelo de classes

- Um cliente pode ter muitas encomendas em seu nome e muitas encomendas podem pertencer a um cliente.
- Uma encomenda tem uma fatura, onde tem o valor a pagar da encomenda efetuada pelo cliente e a data de quando a fatura foi emitida.
- Uma encomenda tem uma linhaEncomenda, onde podemos ver a quantidade de encomendas efetuadas por um cliente.
- Uma fatura pode ter várias linhas de fatura, ou seja, uma fatura contém uma listagem de todos os produtos que o cliente encomendou.
- Um produto tem várias linhasFatura, isto é, podemos listar e adicionar vários produtos a um carrinho de compras.
- Um produto tem uma linhaEncomenda que liga a encFornecedor, aqui é onde escolhemos a quantidade de um certo produto a encomendar a um fornecedor.
- Uma encFornecedor tem o número da encomenda, data e valor total da encomenda realizada pelo gestor.
- Uma encFornecedor tem também um pagamento associado, pagamento esse que o gestor depois tem de efetuar ao fornecedor cuja encomenda efetuou.
- Uma encFornecedor tem associada uma seleção de fornecedores no qual o gestor pode escolher para efetuar a sua encomenda de material.
- Um funcionário na nossa loja de vestuário, pode ser de 2 tipos, funcionário de balcão ou gestor.

## 4.4 Diagrama Entidades Relacionamento/ Modelo relacional

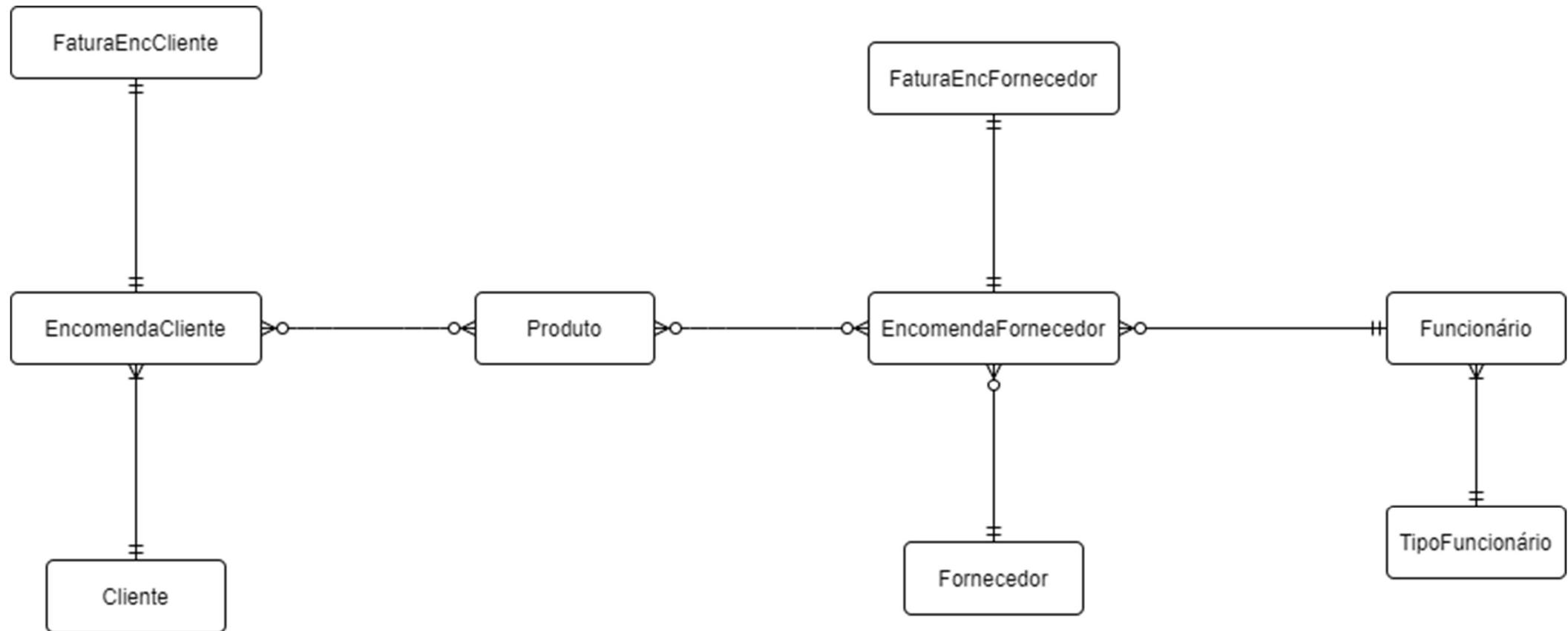
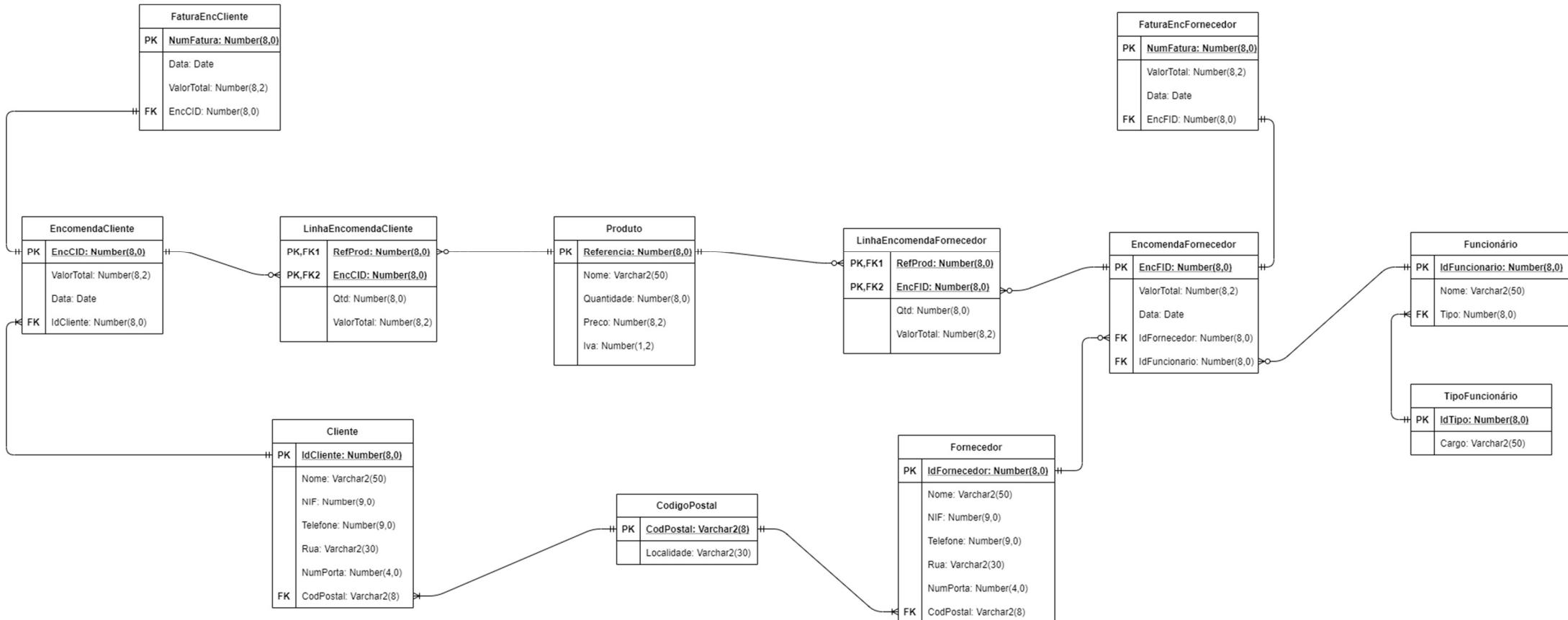


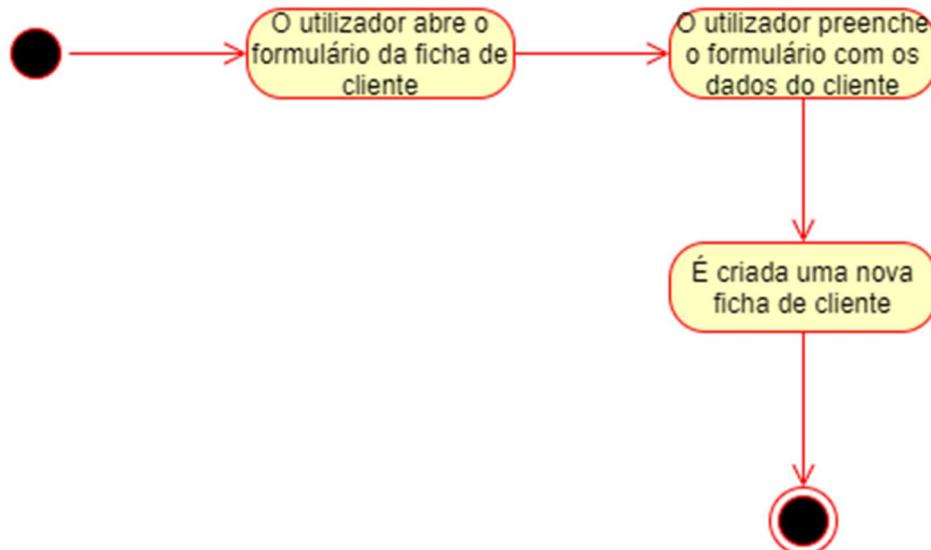
FIGURA 5 - DIAGRAMA ENTIDADES RELACIONAMENTO



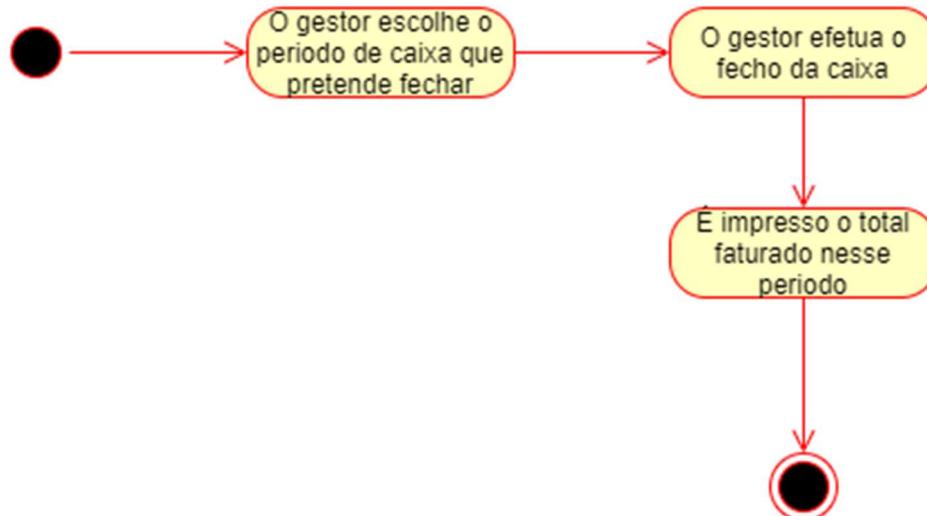
**FIGURA 6 - MODELO RELACIONAL**

#### 4.5 Diagramas de atividades

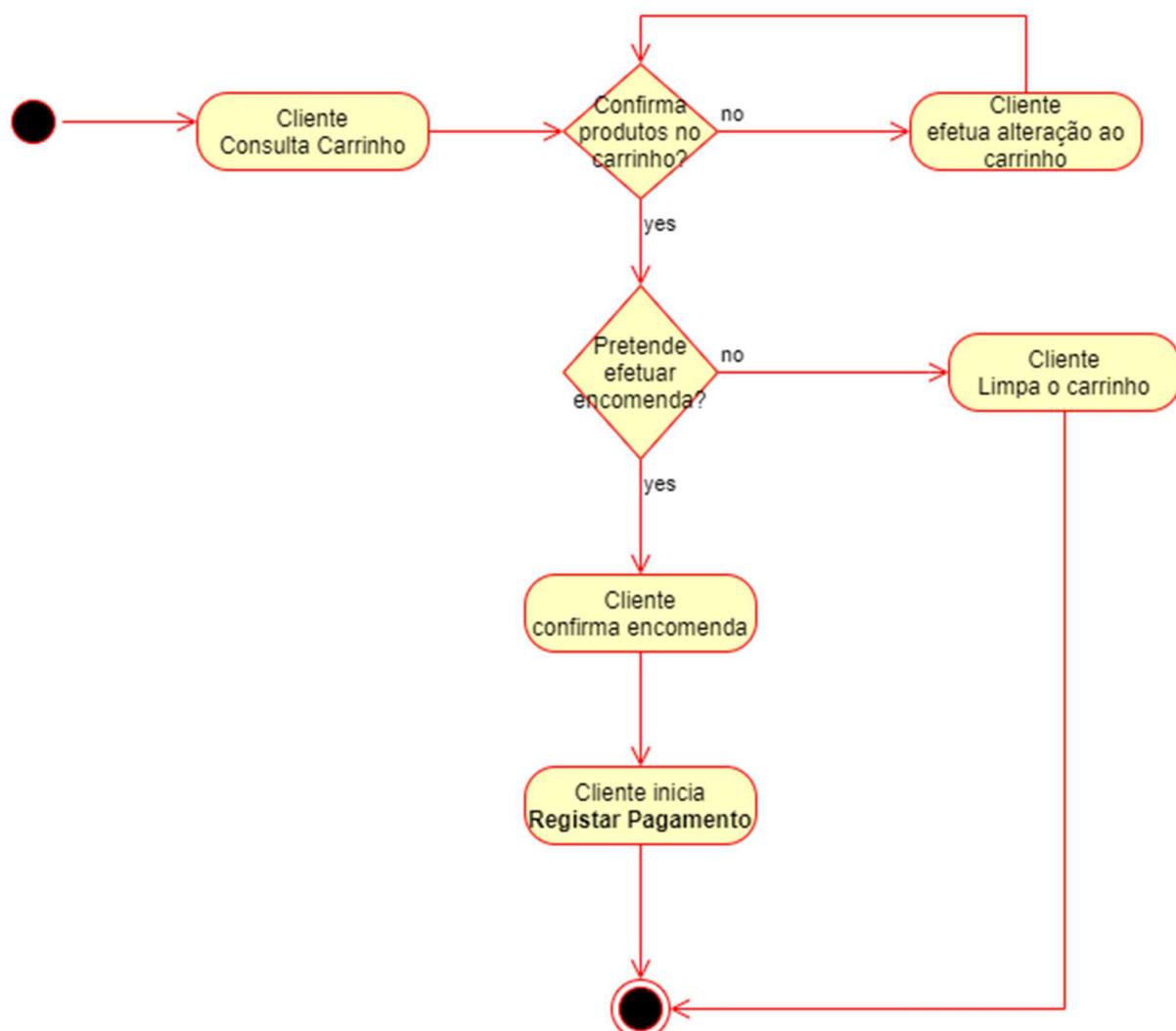
Aqui serão apresentados os diagramas de atividades referentes aos casos de uso do projeto.



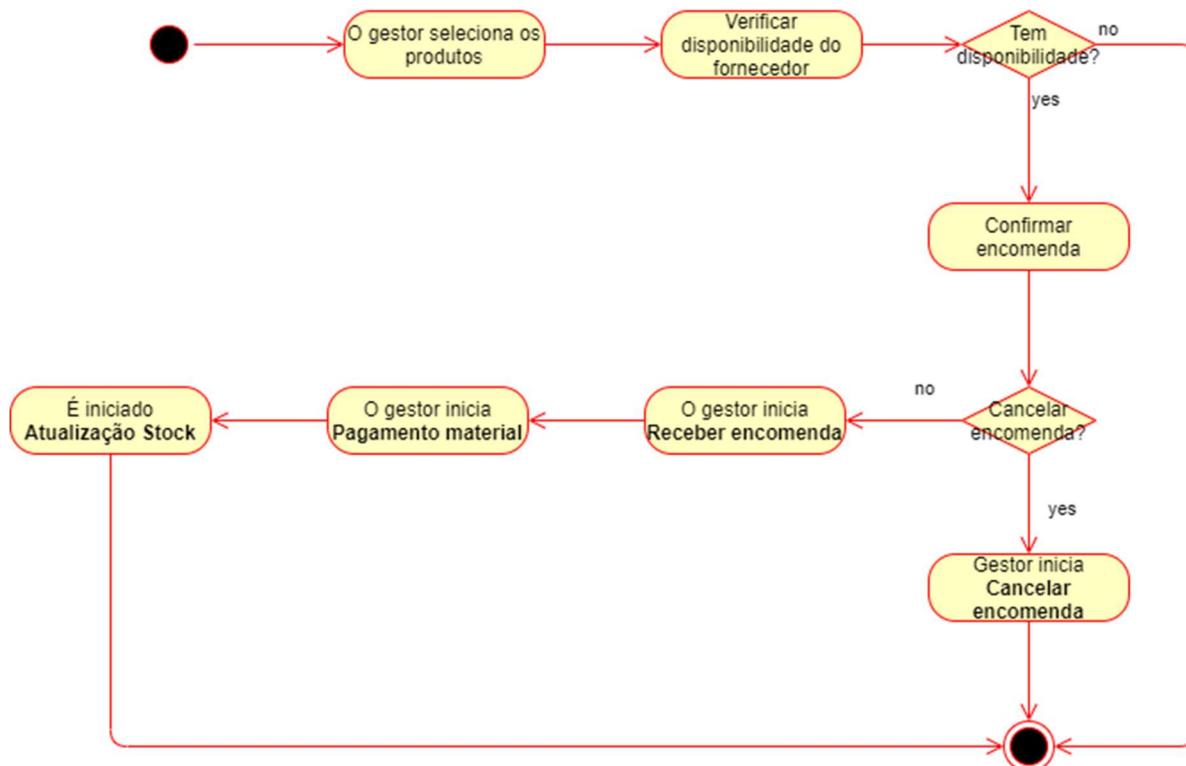
**FIGURA 7 – DIAGRAMA DE ATIVIDADES: CRIAR FICHA DE CLIENTE**



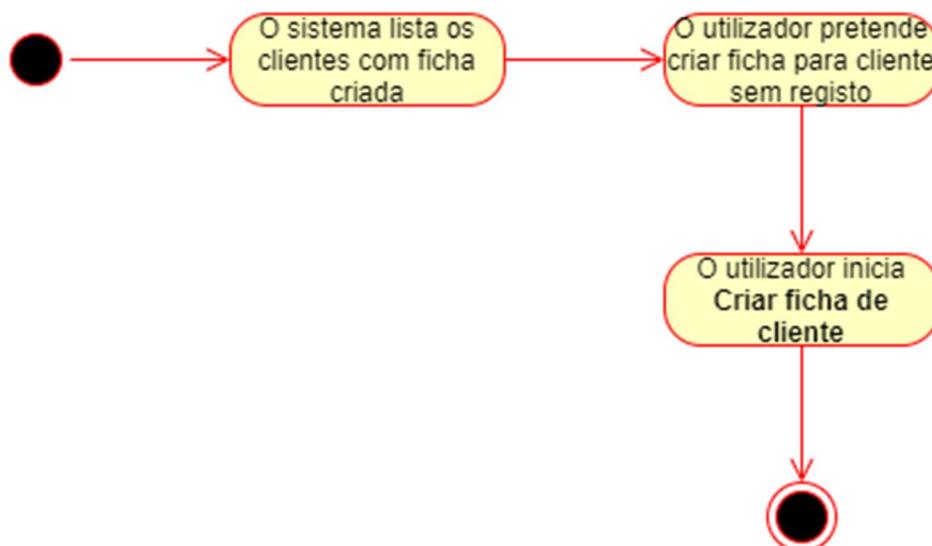
**FIGURA 8 - DIAGRAMA DE ATIVIDADES: FECHO DE CAIXA DE CERTO DIA/MES**



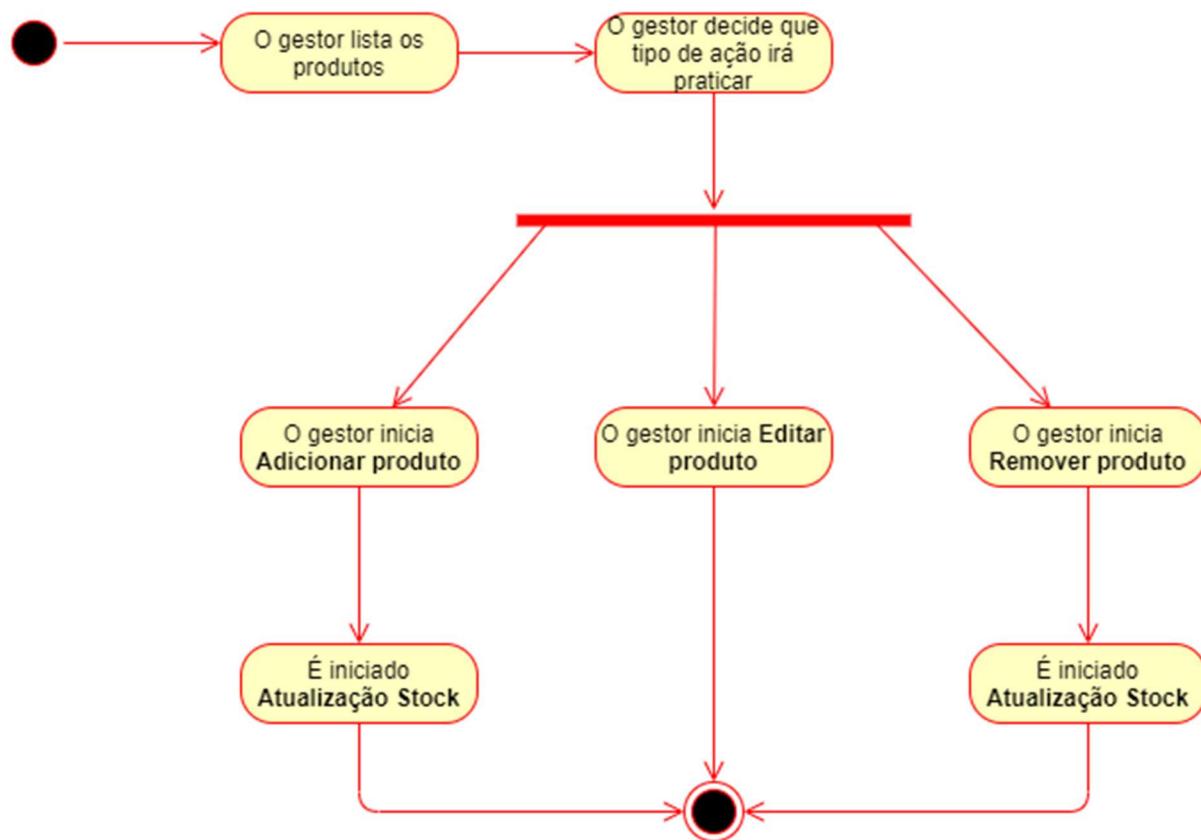
**FIGURA 9 - DIAGRAMA DE ATIVIDADES: CONSULTAR CARRINHO**



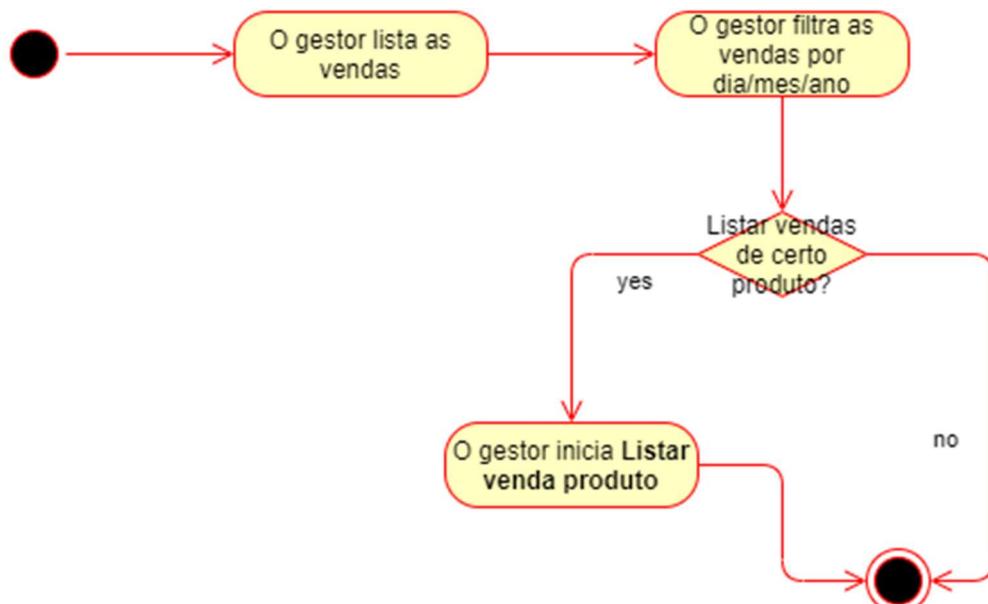
**FIGURA 10 - DIAGRAMA DE ATIVIDADES: EFETUAR ENCOMENDA MATERIAL**



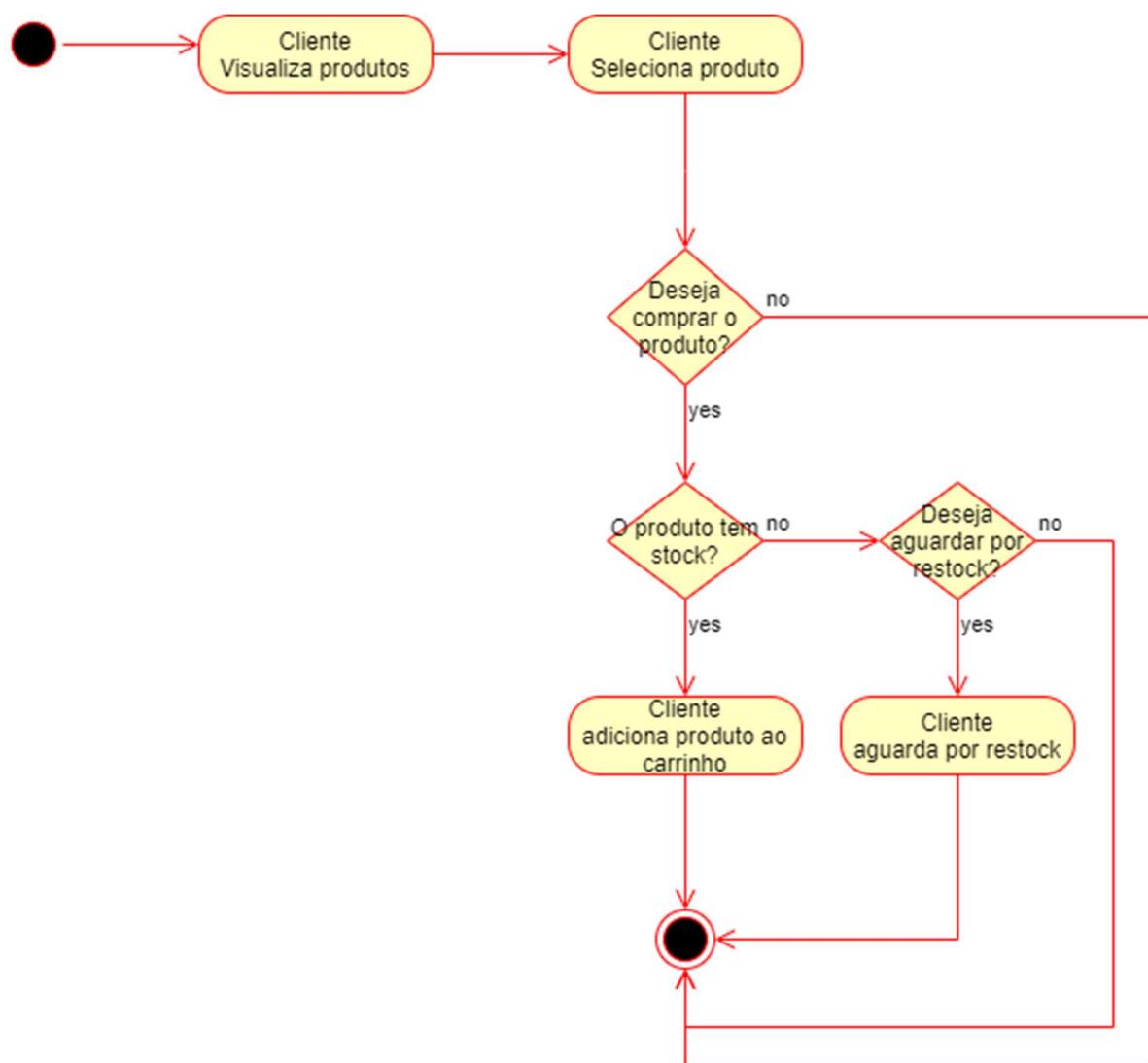
**FIGURA 11 - DIAGRAMA DE ATIVIDADES: LISTAR CLIENTES**



**FIGURA 12 - DIAGRAMA DE ATIVIDADES: GESTÃO DE PRODUTOS NO SISTEMA**



**FIGURA 13 - DIAGRAMA DE ATIVIDADES: LISTAR VENDAS**



**FIGURA 14 - DIAGRAMA DE ATIVIDADES: VISUALIZAR PRODUTO**

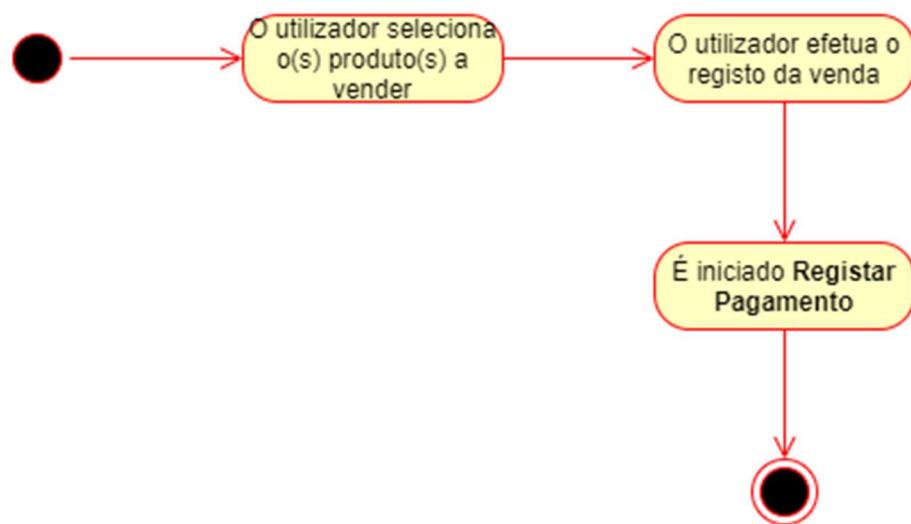
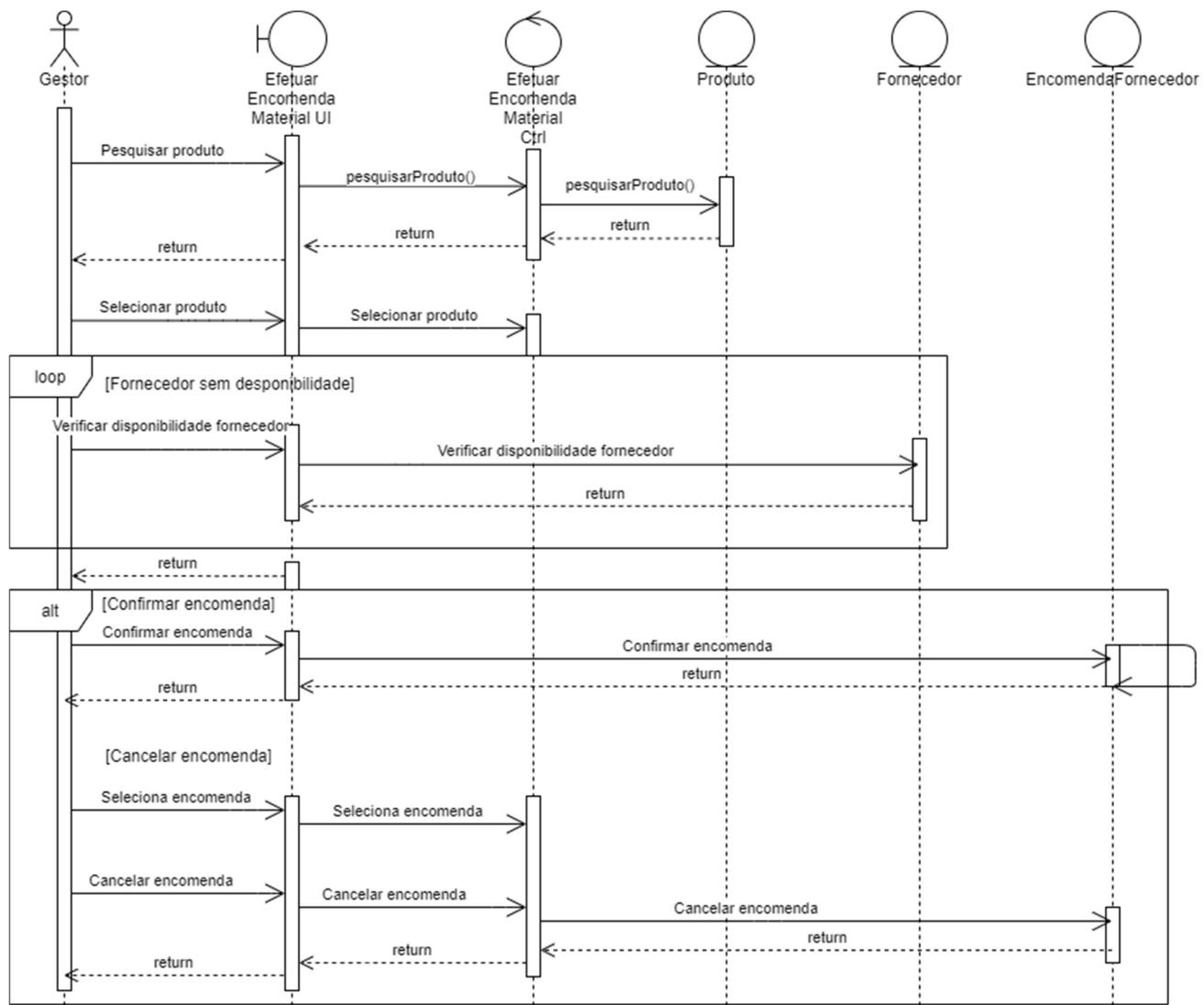


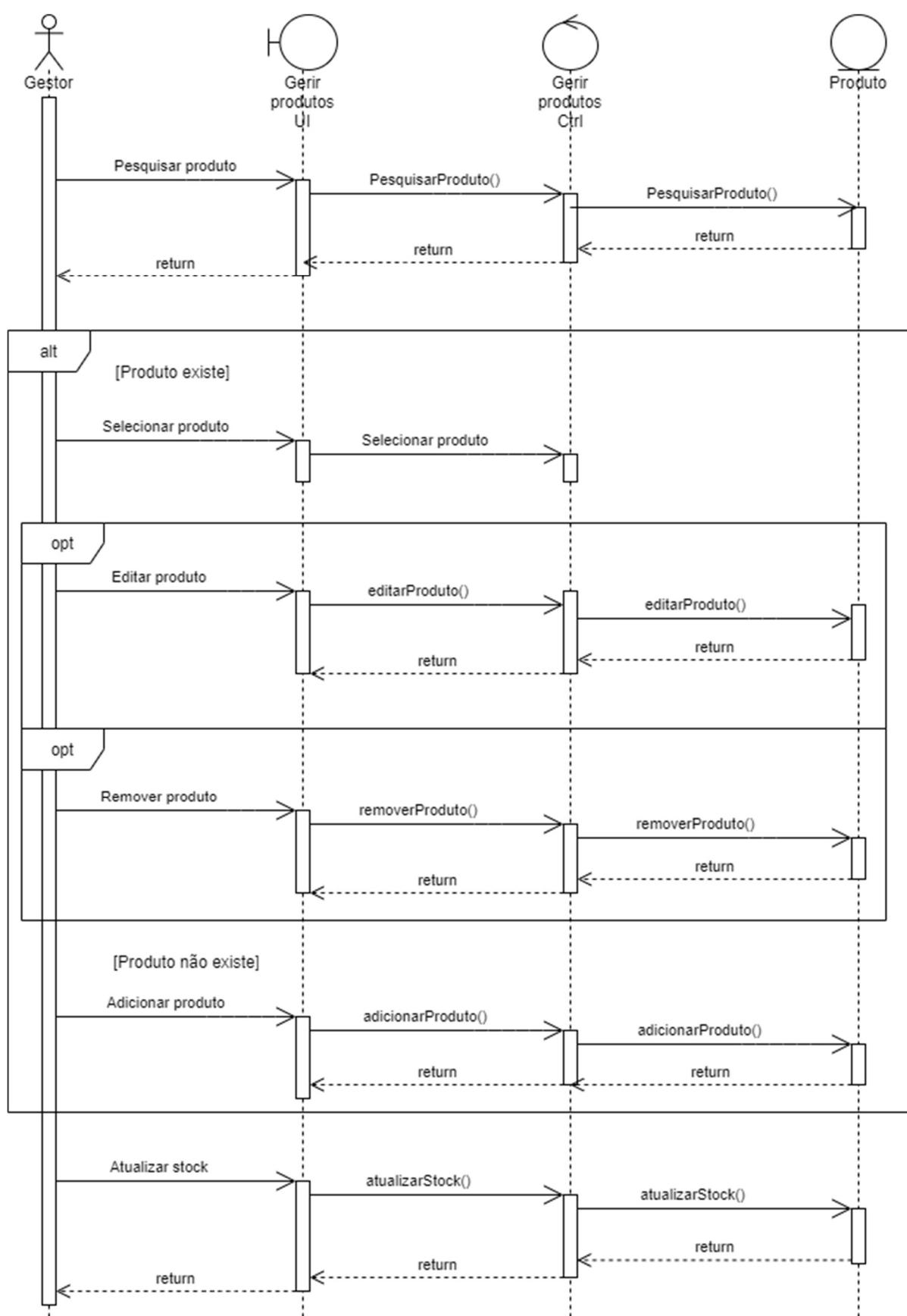
FIGURA 15 - DIAGRAMA DE ATIVIDADES: REGISTAR VENDA

#### 4.6 Diagramas de sequência

Aqui serão apresentados alguns diagramas de sequência referentes a certos casos de uso do projeto.



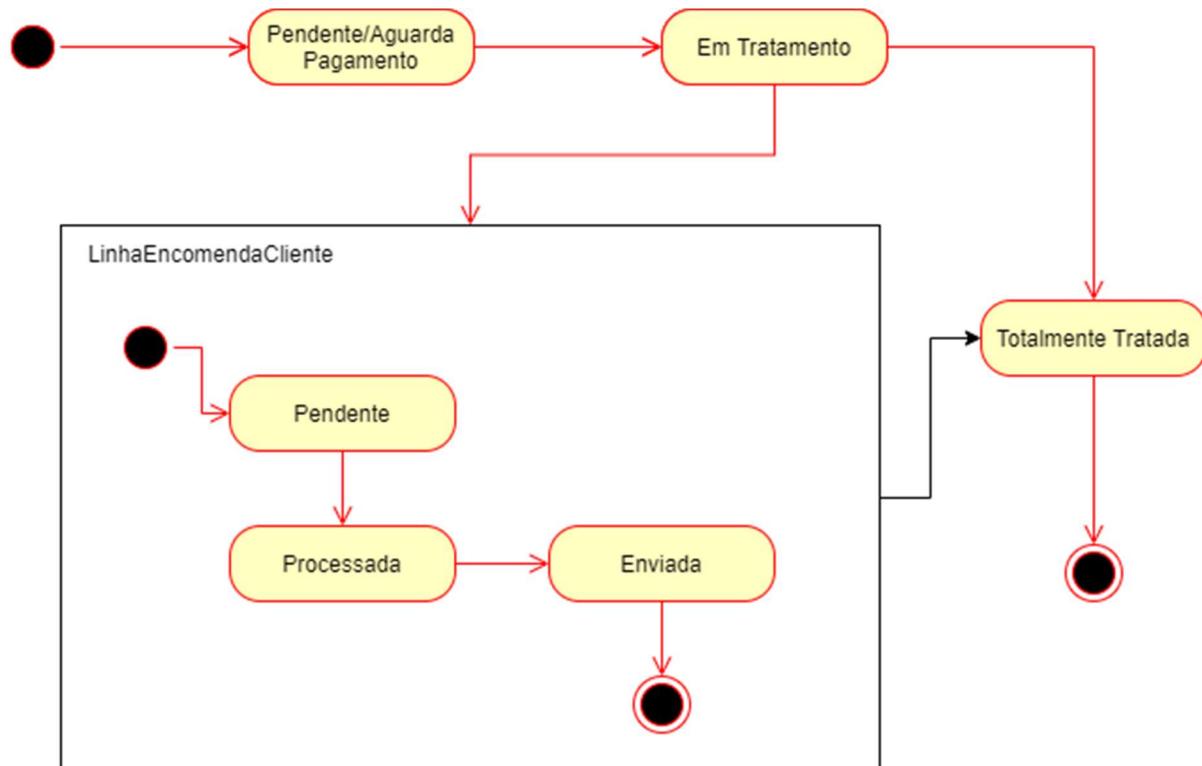
**FIGURA 16 - DIAGRAMA DE SEQUÊNCIA: EFETUAR ENCOMENDA MATERIAL**



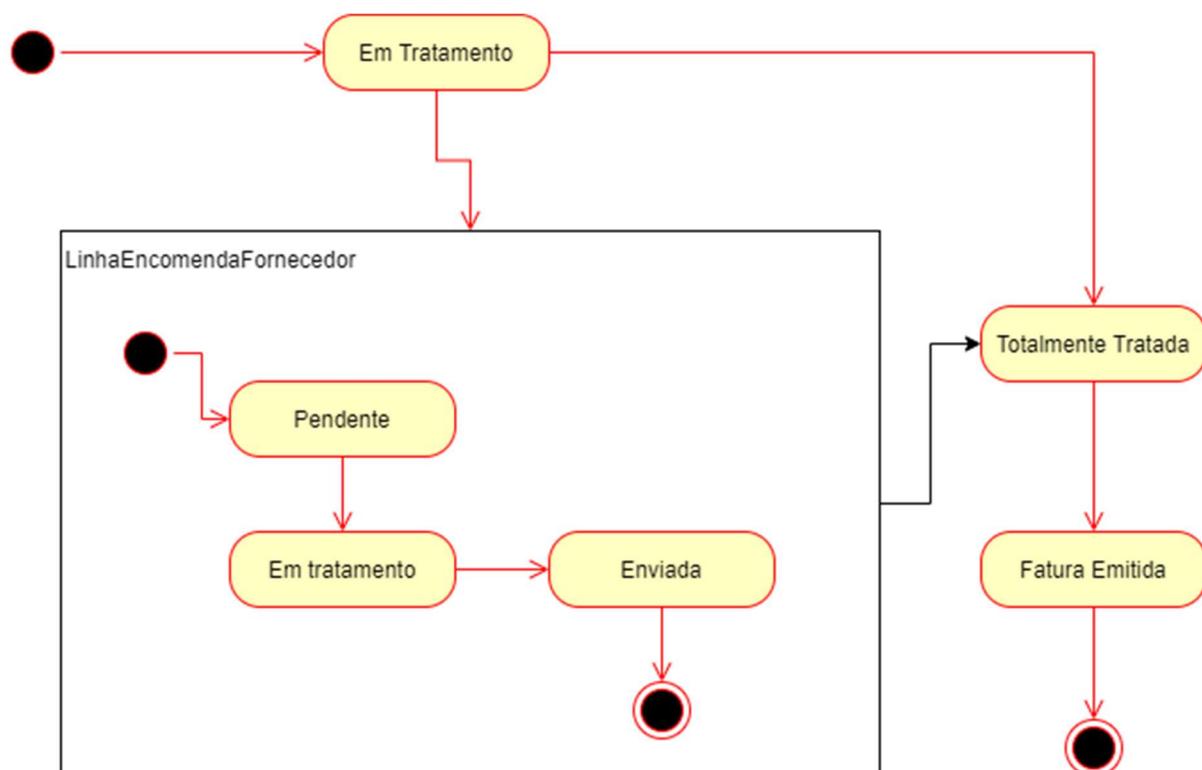
**FIGURA 17 - DIAGRAMA DE SEQUÊNCIA: GERIR PRODUTOS NO SISTEMA**

#### 4.7 Diagramas de transição de estados

Aqui serão apresentados os diagramas de transição de estados referentes ao projeto.



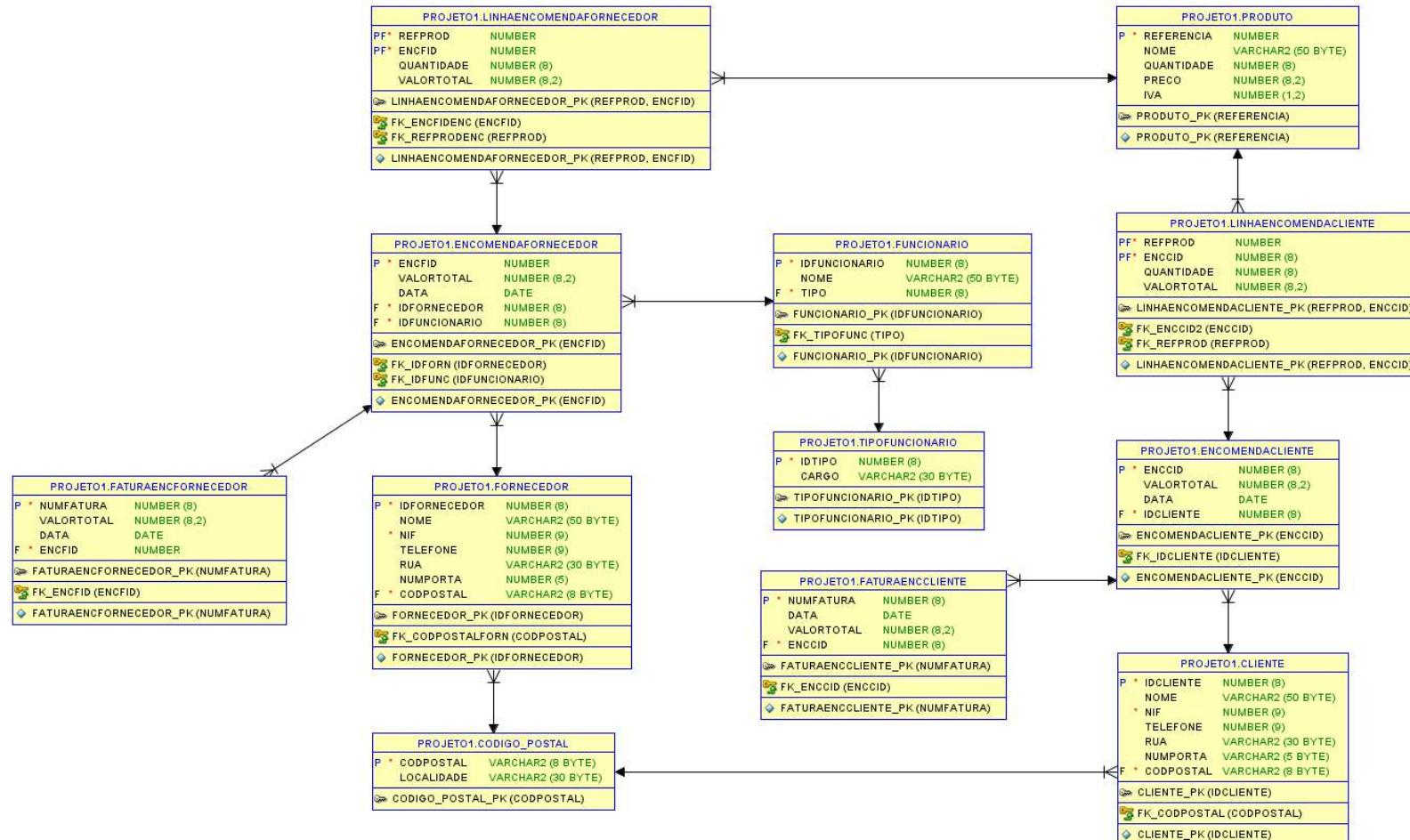
**FIGURA 18 - ENCOMENDA DE CLIENTE**



**FIGURA 19 - ENCOMENDA A FORNECEDOR**

## 5. Implementação da BD

Para a implementação da base de dados foi utilizado o software SQL Developer e o Oracle DataBase.



**FIGURA 20 - MODELO RELACIONAL GERADO PELO SQL DEVELOPER**

## Scripts de criação e triggers da base de dados

---

-- DDL for Table CLIENTE

---

```
CREATE TABLE "PROJETO1"."CLIENTE"
(
    "IDCLIENTE" NUMBER(8,0),
    "NOME" VARCHAR2(50 BYTE),
    "NIF" NUMBER(9,0),
    "TELEFONE" NUMBER(9,0),
    "RUA" VARCHAR2(30 BYTE),
    "NUMPORTA" VARCHAR2(5 BYTE),
    "CODPOSTAL" VARCHAR2(8 BYTE)
) SEGMENT CREATION DEFERRED
PCTFREE 10 PCTUSED 40 INITTRANS 1 MAXTRANS 255
NOCOMPRESS LOGGING
TABLESPACE "USERS";
```

---

-- DDL for Table CODIGO\_POSTAL

---

```
CREATE TABLE "PROJETO1"."CODIGO_POSTAL"
(
    "CODPOSTAL" VARCHAR2(8 BYTE),
    "LOCALIDADE" VARCHAR2(30 BYTE)
) SEGMENT CREATION DEFERRED
PCTFREE 10 PCTUSED 40 INITTRANS 1 MAXTRANS 255
```

*NOCOMPRESS LOGGING*

*TABLESPACE "USERS";*

---

-- *DDL for Table ENCOMENDA CLIENTE*

---

*CREATE TABLE "PROJETO1"."ENCOMENDA CLIENTE"*

(     *"ENCCID" NUMBER(8,0),*  
      *"VALORTOTAL" NUMBER(8,2),*  
      *"DATA" DATE,*  
      *"IDCLIENTE" NUMBER(8,0)*  
    *) SEGMENT CREATION DEFERRED*

*PCTFREE 10 PCTUSED 40 INITTRANS 1 MAXTRANS 255*

*NOCOMPRESS LOGGING*

*TABLESPACE "USERS";*

---

-- *DDL for Table ENCOMENDA FORNECEDOR*

---

*CREATE TABLE "PROJETO1"."ENCOMENDA FORNECEDOR"*

(     *"ENCFID" NUMBER,*  
      *"VALORTOTAL" NUMBER(8,2),*  
      *"DATA" DATE,*  
      *"IDFORNECEDOR" NUMBER(8,0),*  
      *"IDFUNCIONARIO" NUMBER(8,0)*  
    *) SEGMENT CREATION DEFERRED*

*PCTFREE 10 PCTUSED 40 INITTRANS 1 MAXTRANS 255*

*NOCOMPRESS LOGGING*

TABLESPACE "USERS" ;

---

-- DDL for Table FATURAENCLIENTE

---

CREATE TABLE "PROJETO1"."FATURAENCLIENTE"

( "NUMFATURA" NUMBER(8,0),  
"DATA" DATE,  
"VALORTOTAL" NUMBER(8,2),  
"ENCCID" NUMBER(8,0)  
) SEGMENT CREATION DEFERRED

PCTFREE 10 PCTUSED 40 INITTRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

TABLESPACE "USERS" ;

---

-- DDL for Table FATURAENCORNECEDOR

---

CREATE TABLE "PROJETO1"."FATURAENCORNECEDOR"

( "NUMFATURA" NUMBER(8,0),  
"VALORTOTAL" NUMBER(8,2),  
"DATA" DATE,  
"ENCFID" NUMBER  
) SEGMENT CREATION DEFERRED

PCTFREE 10 PCTUSED 40 INITTRANS 1 MAXTRANS 255

NOCOMPRESS LOGGING

TABLESPACE "USERS" ;

---

-- DDL for Table FORNECEDOR

---

```
CREATE TABLE "PROJETO1"."FORNECEDOR"  
(    "IDFORNECEDOR" NUMBER(8,0),  
        "NOME" VARCHAR2(50 BYTE),  
        "NIF" NUMBER(9,0),  
        "TELEFONE" NUMBER(9,0),  
        "RUA" VARCHAR2(30 BYTE),  
        "NUMPORTA" NUMBER(5,0),  
        "CODPOSTAL" VARCHAR2(8 BYTE)  
) SEGMENT CREATION DEFERRED  
PCTFREE 10 PCTUSED 40 INITTRANS 1 MAXTRANS 255  
NOCOMPRESS LOGGING  
TABLESPACE "USERS";
```

---

-- DDL for Table FUNCIONARIO

---

```
CREATE TABLE "PROJETO1"."FUNCIONARIO"  
(    "IDFUNCIONARIO" NUMBER(8,0),  
        "NOME" VARCHAR2(50 BYTE),  
        "TIPO" NUMBER(8,0)  
) SEGMENT CREATION DEFERRED  
PCTFREE 10 PCTUSED 40 INITTRANS 1 MAXTRANS 255  
NOCOMPRESS LOGGING  
TABLESPACE "USERS";
```

---

-- DDL for Table LINHAENCOMENDA CLIENTE

---

```
CREATE TABLE "PROJETO1"."LINHAENCOMENDA CLIENTE"
```

```
(    "REFPROD" NUMBER,  
    "ENCCID" NUMBER(8,0),  
    "QUANTIDADE" NUMBER(8,0),  
    "VALORTOTAL" NUMBER(8,2)  
) SEGMENT CREATION DEFERRED  
PCTFREE 10 PCTUSED 40 INITTRANS 1 MAXTRANS 255  
NOCOMPRESS LOGGING  
TABLESPACE "USERS";
```

---

-- DDL for Table LINHAENCOMENDA FORNECEDOR

---

```
CREATE TABLE "PROJETO1"."LINHAENCOMENDA FORNECEDOR"
```

```
(    "REFPROD" NUMBER,  
    "ENCFID" NUMBER,  
    "QUANTIDADE" NUMBER(8,0),  
    "VALORTOTAL" NUMBER(8,2)  
) SEGMENT CREATION DEFERRED  
PCTFREE 10 PCTUSED 40 INITTRANS 1 MAXTRANS 255  
NOCOMPRESS LOGGING  
TABLESPACE "USERS";
```

---

-- DDL for Table PRODUTO

---

```

CREATE TABLE "PROJETO1"."PRODUTO"
(
    "REFERENCIA" NUMBER,
    "NOME" VARCHAR2(50 BYTE),
    "QUANTIDADE" NUMBER(8,0),
    "PRECO" NUMBER(8,2),
    "IVA" NUMBER(1,2)
) SEGMENT CREATION DEFERRED
PCTFREE 10 PCTUSED 40 INITTRANS 1 MAXTRANS 255
NOCOMPRESS LOGGING
TABLESPACE "USERS";

```

---

-- DDL for Table TIPOFUNCIONARIO

---

```

CREATE TABLE "PROJETO1"."TIPOFUNCIONARIO"
(
    "IDTIPO" NUMBER(8,0),
    "CARGO" VARCHAR2(30 BYTE)
) SEGMENT CREATION DEFERRED
PCTFREE 10 PCTUSED 40 INITTRANS 1 MAXTRANS 255
NOCOMPRESS LOGGING
TABLESPACE "USERS";

```

---

-- DDL for Index CLIENTE\_PK

---

```

CREATE UNIQUE INDEX "PROJETO1"."CLIENTE_PK" ON "PROJETO1"."CLIENTE" ("IDCLIENTE")
PCTFREE 10 INITTRANS 2 MAXTRANS 255

```

TABLESPACE "USERS" ;

-- DDL for Index UNIQUE\_NIFCLI

CREATE UNIQUE INDEX "PROJETO1"."UNIQUE\_NIFCLI" ON "PROJETO1"."CLIENTE" ("NIF")

PCTFREE 10 INITTRANS 2 MAXTRANS 255 COMPUTE STATISTICS

TABLESPACE "USERS" ;

-- DDL for Index SYS\_C008337

CREATE UNIQUE INDEX "PROJETO1"."SYS\_C008337" ON "PROJETO1"."CODIGO\_POSTAL"  
("CODPOSTAL")

PCTFREE 10 INITTRANS 2 MAXTRANS 255

TABLESPACE "USERS" ;

-- DDL for Index ENCOMENDA\_CLIENTE\_PK

CREATE UNIQUE INDEX "PROJETO1"."ENCOMENDA\_CLIENTE\_PK" ON  
"PROJETO1"."ENCOMENDA\_CLIENTE" ("ENCCID")

PCTFREE 10 INITTRANS 2 MAXTRANS 255

TABLESPACE "USERS" ;

-- DDL for Index ENCOMENDA\_FORNECEDOR\_PK

```
CREATE UNIQUE INDEX "PROJETO1"."ENCOMENDAFORNECEDOR_PK" ON
"PROJETO1"."ENCOMENDAFORNECEDOR" ("ENCFID")
```

```
PCTFREE 10 INITRANS 2 MAXTRANS 255
```

```
TABLESPACE "USERS";
```

---

```
-- DDL for Index FATURAENCCLIENTE_PK
```

---

```
CREATE UNIQUE INDEX "PROJETO1"."FATURAENCCLIENTE_PK" ON
"PROJETO1"."FATURAENCCLIENTE" ("NUMFATURA")
```

```
PCTFREE 10 INITRANS 2 MAXTRANS 255
```

```
TABLESPACE "USERS";
```

---

```
-- DDL for Index FATURAENCFORNECEDOR_PK
```

---

```
CREATE UNIQUE INDEX "PROJETO1"."FATURAENCFORNECEDOR_PK" ON
"PROJETO1"."FATURAENCFORNECEDOR" ("NUMFATURA")
```

```
PCTFREE 10 INITRANS 2 MAXTRANS 255
```

```
TABLESPACE "USERS";
```

---

```
-- DDL for Index FORNECEDOR_PK
```

---

```
CREATE UNIQUE INDEX "PROJETO1"."FORNECEDOR_PK" ON "PROJETO1"."FORNECEDOR"
("IDFORNECEDOR")
```

```
PCTFREE 10 INITRANS 2 MAXTRANS 255
```

```
TABLESPACE "USERS";
```

---

```
-- DDL for Index UNIQUE_NIFFORN
```

---

```
CREATE UNIQUE INDEX "PROJETO1"."UNIQUE_NIFFORN" ON "PROJETO1"."FORNECEDOR" ("NIF")
PCTFREE 10 INITTRANS 2 MAXTRANS 255 COMPUTE STATISTICS
TABLESPACE "USERS";
```

```
-- DDL for Index FUNCIONARIO_PK
```

```
CREATE UNIQUE INDEX "PROJETO1"."FUNCIONARIO_PK" ON "PROJETO1"."FUNCIONARIO"
("IDFUNCIONARIO")
PCTFREE 10 INITTRANS 2 MAXTRANS 255
TABLESPACE "USERS";
```

```
-- DDL for Index LINHAENCOMENDACLIENTE_PK
```

```
CREATE UNIQUE INDEX "PROJETO1"."LINHAENCOMENDACLIENTE_PK" ON
"PROJETO1"."LINHAENCOMENDACLIENTE" ("REFPROD", "ENCCID")
PCTFREE 10 INITTRANS 2 MAXTRANS 255
TABLESPACE "USERS";
```

```
-- DDL for Index LINHAENCOMENDAFORNECEDOR_PK
```

```
CREATE UNIQUE INDEX "PROJETO1"."LINHAENCOMENDAFORNECEDOR_PK" ON
"PROJETO1"."LINHAENCOMENDAFORNECEDOR" ("REFPROD", "ENCFID")
PCTFREE 10 INITTRANS 2 MAXTRANS 255
TABLESPACE "USERS";
```

-- DDL for Index *PRODUTO\_PK*

---

```
CREATE UNIQUE INDEX "PROJETO1"."PRODUTO_PK" ON "PROJETO1"."PRODUTO" ("REFERENCIA")
PCTFREE 10 INITTRANS 2 MAXTRANS 255
TABLESPACE "USERS";
```

---

-- DDL for Index *TIPOFUNCIONARIO\_PK*

---

```
CREATE UNIQUE INDEX "PROJETO1"."TIPOFUNCIONARIO_PK" ON "PROJETO1"."TIPOFUNCIONARIO"
("IDTIPO")
PCTFREE 10 INITTRANS 2 MAXTRANS 255
TABLESPACE "USERS";
```

---

-- DDL for Trigger *CLIENTE\_TRG*

---

```
CREATE OR REPLACE NONEDITIONABLE TRIGGER "PROJETO1"."CLIENTE_TRG"
BEFORE INSERT ON CLIENTE
FOR EACH ROW
BEGIN
<<COLUMN_SEQUENCES>>
BEGIN
IF INSERTING AND :NEW.IDCLIENTE IS NULL THEN
SELECT CLIENTE_SEQ.NEXTVAL INTO :NEW.IDCLIENTE FROM SYS.DUAL;
END IF;
```

```

END COLUMN_SEQUENCES;

END;
/

ALTER TRIGGER "PROJETO1"."CLIENTE_TRG" ENABLE;
-----  

-- DDL for Trigger ENCOMENDA CLIENTE TRG
-----
```

```

CREATE OR REPLACE NONEDITABLE TRIGGER "PROJETO1"."ENCOMENDA CLIENTE TRG"
BEFORE INSERT ON ENCOMENDA CLIENTE
FOR EACH ROW
BEGIN
<<COLUMN_SEQUENCES>>
BEGIN
IF INSERTING AND :NEW.ENCCID IS NULL THEN
SELECT ENCOMENDA CLIENTE SEQ.NEXTVAL INTO :NEW.ENCCID FROM SYS.DUAL;
END IF;
END COLUMN_SEQUENCES;
END;
/
```

```
ALTER TRIGGER "PROJETO1"."ENCOMENDA CLIENTE TRG" ENABLE;
```

```
-----  

-- DDL for Trigger ENCOMENDA FORNECEDOR TRG
-----
```

```

CREATE OR REPLACE NONEDITABLE TRIGGER "PROJETO1"."ENCOMENDA FORNECEDOR TRG"
BEFORE INSERT ON ENCOMENDA FORNECEDOR
FOR EACH ROW
```

BEGIN

<<COLUMN\_SEQUENCES>>

BEGIN

IF INSERTING AND :NEW.ENCFID IS NULL THEN

SELECT ENCOMENDAFORNECEDOR\_SEQ.NEXTVAL INTO :NEW.ENCFID FROM SYS.DUAL;

END IF;

END COLUMN\_SEQUENCES;

END;

/

ALTER TRIGGER "PROJETO1"."ENCOMENDAFORNECEDOR\_TRG" ENABLE;

---

-- DDL for Trigger FATURAENCLIENTE\_TRG

---

CREATE OR REPLACE NONEDITABLE TRIGGER "PROJETO1"."FATURAENCLIENTE\_TRG"

BEFORE INSERT ON FATURAENCLIENTE

FOR EACH ROW

BEGIN

<<COLUMN\_SEQUENCES>>

BEGIN

IF INSERTING AND :NEW.NUMFATURA IS NULL THEN

SELECT FATURAENCLIENTE\_SEQ1.NEXTVAL INTO :NEW.NUMFATURA FROM SYS.DUAL;

END IF;

END COLUMN\_SEQUENCES;

END;

/

ALTER TRIGGER "PROJETO1"."FATURAENCLIENTE\_TRG" ENABLE;

-- DDL for Trigger FATURAENFORNECEDOR\_TRG

---

```
CREATE OR REPLACE NONEDITABLE TRIGGER "PROJETO1"."FATURAENFORNECEDOR_TRG"  
BEFORE INSERT ON FATURAENFORNECEDOR  
FOR EACH ROW  
BEGIN  
<<COLUMN_SEQUENCES>>  
BEGIN  
IF INSERTING AND :NEW.NUMFATURA IS NULL THEN  
SELECT FATURAENFORNECEDOR_SEQ.NEXTVAL INTO :NEW.NUMFATURA FROM SYS.DUAL;  
END IF;  
END COLUMN_SEQUENCES;  
END;  
/  
ALTER TRIGGER "PROJETO1"."FATURAENFORNECEDOR_TRG" ENABLE;
```

---

-- DDL for Trigger FORNECEDOR\_TRG

---

```
CREATE OR REPLACE NONEDITABLE TRIGGER "PROJETO1"."FORNECEDOR_TRG"  
BEFORE INSERT ON FORNECEDOR  
FOR EACH ROW  
BEGIN  
<<COLUMN_SEQUENCES>>  
BEGIN  
IF INSERTING AND :NEW.IDFORNECEDOR IS NULL THEN  
SELECT FORNECEDOR_SEQ.NEXTVAL INTO :NEW.IDFORNECEDOR FROM SYS.DUAL;
```

```
END IF;

END COLUMN_SEQUENCES;

END;

/

ALTER TRIGGER "PROJETO1"."FORNECEDOR_TRG" ENABLE;
```

```
-- DDL for Trigger FUNCIONARIO_TRG
```

```
CREATE OR REPLACE NONEDITABLE TRIGGER "PROJETO1"."FUNCIONARIO_TRG"
BEFORE INSERT ON FUNCIONARIO
FOR EACH ROW
BEGIN
<<COLUMN_SEQUENCES>>
BEGIN
IF INSERTING AND :NEW.IDFUNCIONARIO IS NULL THEN
SELECT FUNCIONARIO_SEQ.NEXTVAL INTO :NEW.IDFUNCIONARIO FROM SYS.DUAL;
END IF;
END COLUMN_SEQUENCES;
END;

/

ALTER TRIGGER "PROJETO1"."FUNCIONARIO_TRG" ENABLE;
```

```
-- DDL for Trigger TIPOFUNCIONARIO_TRG
```

```
CREATE OR REPLACE NONEDITABLE TRIGGER "PROJETO1"."TIPOFUNCIONARIO_TRG"
BEFORE INSERT ON TIPOFUNCIONARIO
```

FOR EACH ROW

BEGIN

<<COLUMN\_SEQUENCES>>

BEGIN

IF INSERTING AND :NEW.IDTIPO IS NULL THEN

SELECT TIPOFUNCIONARIO\_SEQ.NEXTVAL INTO :NEW.IDTIPO FROM SYS.DUAL;

END IF;

END COLUMN\_SEQUENCES;

END;

/

ALTER TRIGGER "PROJETO1"."TIPOFUNCIONARIO\_TRG" ENABLE;

---

-- Constraints for Table CLIENTE

---

ALTER TABLE "PROJETO1"."CLIENTE" MODIFY ("IDCLIENTE" NOT NULL ENABLE);

ALTER TABLE "PROJETO1"."CLIENTE" MODIFY ("NIF" NOT NULL ENABLE);

ALTER TABLE "PROJETO1"."CLIENTE" MODIFY ("CODPOSTAL" NOT NULL ENABLE);

ALTER TABLE "PROJETO1"."CLIENTE" ADD CONSTRAINT "CLIENTE\_PK" PRIMARY KEY ("IDCLIENTE")

USING INDEX PCTFREE 10 INITTRANS 2 MAXTRANS 255

TABLESPACE "USERS" ENABLE;

ALTER TABLE "PROJETO1"."CLIENTE" ADD CONSTRAINT "UNIQUE\_NIFCLI" UNIQUE ("NIF")

USING INDEX PCTFREE 10 INITTRANS 2 MAXTRANS 255 COMPUTE STATISTICS

TABLESPACE "USERS" ENABLE;

---

-- Constraints for Table CODIGO\_POSTAL

---

```
ALTER TABLE "PROJETO1"."CODIGO_POSTAL" ADD PRIMARY KEY ("CODPOSTAL")
```

```
USING INDEX PCTFREE 10 INITTRANS 2 MAXTRANS 255
```

```
TABLESPACE "USERS" ENABLE;
```

---

```
-- Constraints for Table ENCOMENDA CLIENTE
```

---

```
ALTER TABLE "PROJETO1"."ENCOMENDA CLIENTE" MODIFY ("ENCCID" NOT NULL ENABLE);
```

```
ALTER TABLE "PROJETO1"."ENCOMENDA CLIENTE" MODIFY ("IDCLIENTE" NOT NULL ENABLE);
```

```
ALTER TABLE "PROJETO1"."ENCOMENDA CLIENTE" ADD CONSTRAINT "ENCOMENDA CLIENTE_PK"  
PRIMARY KEY ("ENCCID")
```

```
USING INDEX PCTFREE 10 INITTRANS 2 MAXTRANS 255
```

```
TABLESPACE "USERS" ENABLE;
```

---

```
-- Constraints for Table ENCOMENDA FORNECEDOR
```

---

```
ALTER TABLE "PROJETO1"."ENCOMENDA FORNECEDOR" MODIFY ("ENCFID" NOT NULL ENABLE);
```

```
ALTER TABLE "PROJETO1"."ENCOMENDA FORNECEDOR" MODIFY ("IDFORNECEDOR" NOT NULL  
ENABLE);
```

```
ALTER TABLE "PROJETO1"."ENCOMENDA FORNECEDOR" MODIFY ("IDFUNCIONARIO" NOT NULL  
ENABLE);
```

```
ALTER TABLE "PROJETO1"."ENCOMENDA FORNECEDOR" ADD CONSTRAINT  
"ENCOMENDA FORNECEDOR_PK" PRIMARY KEY ("ENCFID")
```

```
USING INDEX PCTFREE 10 INITTRANS 2 MAXTRANS 255
```

```
TABLESPACE "USERS" ENABLE;
```

---

```
-- Constraints for Table FATURAEN CLIENTE
```

---

```
ALTER TABLE "PROJETO1"."FATURAENCLIENTE" MODIFY ("NUMFATURA" NOT NULL ENABLE);

ALTER TABLE "PROJETO1"."FATURAENCLIENTE" MODIFY ("ENCCID" NOT NULL ENABLE);

ALTER TABLE "PROJETO1"."FATURAENCLIENTE" ADD CONSTRAINT "FATURAENCLIENTE_PK"
PRIMARY KEY ("NUMFATURA")

USING INDEX PCTFREE 10 INITTRANS 2 MAXTRANS 255

TABLESPACE "USERS" ENABLE;
```

---

-- Constraints for Table FATURAENCORNECEDOR

---

```
ALTER TABLE "PROJETO1"."FATURAENCORNECEDOR" MODIFY ("NUMFATURA" NOT NULL
ENABLE);

ALTER TABLE "PROJETO1"."FATURAENCORNECEDOR" MODIFY ("ENCFID" NOT NULL ENABLE);

ALTER TABLE "PROJETO1"."FATURAENCORNECEDOR" ADD CONSTRAINT
"FATURAENCORNECEDOR_PK" PRIMARY KEY ("NUMFATURA")

USING INDEX PCTFREE 10 INITTRANS 2 MAXTRANS 255

TABLESPACE "USERS" ENABLE;
```

---

-- Constraints for Table FORNECEDOR

---

```
ALTER TABLE "PROJETO1"."FORNECEDOR" MODIFY ("IDFORNECEDOR" NOT NULL ENABLE);

ALTER TABLE "PROJETO1"."FORNECEDOR" MODIFY ("NIF" NOT NULL ENABLE);

ALTER TABLE "PROJETO1"."FORNECEDOR" MODIFY ("CODPOSTAL" NOT NULL ENABLE);

ALTER TABLE "PROJETO1"."FORNECEDOR" ADD CONSTRAINT "FORNECEDOR_PK" PRIMARY KEY
("IDFORNECEDOR")

USING INDEX PCTFREE 10 INITTRANS 2 MAXTRANS 255

TABLESPACE "USERS" ENABLE;

ALTER TABLE "PROJETO1"."FORNECEDOR" ADD CONSTRAINT "UNIQUE_NIFFORN" UNIQUE ("NIF")

USING INDEX PCTFREE 10 INITTRANS 2 MAXTRANS 255 COMPUTE STATISTICS
```

TABLESPACE "USERS" ENABLE;

---

-- Constraints for Table FUNCIONARIO

---

ALTER TABLE "PROJETO1"."FUNCIONARIO" MODIFY ("IDFUNCIONARIO" NOT NULL ENABLE);

ALTER TABLE "PROJETO1"."FUNCIONARIO" MODIFY ("TIPO" NOT NULL ENABLE);

ALTER TABLE "PROJETO1"."FUNCIONARIO" ADD CONSTRAINT "FUNCIONARIO\_PK" PRIMARY KEY ("IDFUNCIONARIO")

USING INDEX PCTFREE 10 INITTRANS 2 MAXTRANS 255

TABLESPACE "USERS" ENABLE;

---

-- Constraints for Table LINHAENCOMENDA CLIENTE

---

ALTER TABLE "PROJETO1"."LINHAENCOMENDA CLIENTE" MODIFY ("REFPROD" NOT NULL ENABLE);

ALTER TABLE "PROJETO1"."LINHAENCOMENDA CLIENTE" MODIFY ("ENCCID" NOT NULL ENABLE);

ALTER TABLE "PROJETO1"."LINHAENCOMENDA CLIENTE" ADD CONSTRAINT "LINHAENCOMENDA CLIENTE\_PK" PRIMARY KEY ("REFPROD", "ENCCID")

USING INDEX PCTFREE 10 INITTRANS 2 MAXTRANS 255

TABLESPACE "USERS" ENABLE;

---

-- Constraints for Table LINHAENCOMENDA FORNECEDOR

---

ALTER TABLE "PROJETO1"."LINHAENCOMENDA FORNECEDOR" MODIFY ("REFPROD" NOT NULL ENABLE);

ALTER TABLE "PROJETO1"."LINHAENCOMENDA FORNECEDOR" MODIFY ("ENCFID" NOT NULL ENABLE);

```
ALTER TABLE "PROJETO1"."LINHAENCOMENDAFORNECEDOR" ADD CONSTRAINT  
"LINHAENCOMENDAFORNECEDOR_PK" PRIMARY KEY ("REFPROD", "ENCFID")
```

```
USING INDEX PCTFREE 10 INITTRANS 2 MAXTRANS 255
```

```
TABLESPACE "USERS" ENABLE;
```

---

```
-- Constraints for Table PRODUTO
```

---

```
ALTER TABLE "PROJETO1"."PRODUTO" MODIFY ("REFERENCIA" NOT NULL ENABLE);
```

```
ALTER TABLE "PROJETO1"."PRODUTO" ADD CONSTRAINT "PRODUTO_PK" PRIMARY KEY  
("REFERENCIA")
```

```
USING INDEX PCTFREE 10 INITTRANS 2 MAXTRANS 255
```

```
TABLESPACE "USERS" ENABLE;
```

---

```
-- Constraints for Table TIPOFUNCIONARIO
```

---

```
ALTER TABLE "PROJETO1"."TIPOFUNCIONARIO" MODIFY ("IDTIPO" NOT NULL ENABLE);
```

```
ALTER TABLE "PROJETO1"."TIPOFUNCIONARIO" ADD CONSTRAINT "TIPOFUNCIONARIO_PK"  
PRIMARY KEY ("IDTIPO")
```

```
USING INDEX PCTFREE 10 INITTRANS 2 MAXTRANS 255
```

```
TABLESPACE "USERS" ENABLE;
```

---

```
-- Ref Constraints for Table CLIENTE
```

---

```
ALTER TABLE "PROJETO1"."CLIENTE" ADD CONSTRAINT "FK_CODPOSTAL" FOREIGN KEY  
("CODPOSTAL")
```

```
REFERENCES "PROJETO1"."CÓDIGO_POSTAL" ("CODPOSTAL") ENABLE;
```

-- Ref Constraints for Table ENCOMENDA CLIENTE

---

```
ALTER TABLE "PROJETO1"."ENCOMENDA CLIENTE" ADD CONSTRAINT "FK_IDCLIENTE" FOREIGN KEY  
("IDCLIENTE")
```

```
    REFERENCES "PROJETO1"."CLIENTE" ("IDCLIENTE") ENABLE;
```

---

-- Ref Constraints for Table ENCOMENDA FORNECEDOR

---

```
ALTER TABLE "PROJETO1"."ENCOMENDA FORNECEDOR" ADD CONSTRAINT "FK_IDFORN" FOREIGN  
KEY ("IDFORNECEDOR")
```

```
    REFERENCES "PROJETO1"."FORNECEDOR" ("IDFORNECEDOR") ENABLE;
```

```
ALTER TABLE "PROJETO1"."ENCOMENDA FORNECEDOR" ADD CONSTRAINT "FK_IDFUNC" FOREIGN  
KEY ("IDFUNCIONARIO")
```

```
    REFERENCES "PROJETO1"."FUNCIONARIO" ("IDFUNCIONARIO") ENABLE;
```

---

-- Ref Constraints for Table FATURAEN CLIENTE

---

```
ALTER TABLE "PROJETO1"."FATURAEN CLIENTE" ADD CONSTRAINT "FK_ENCCID" FOREIGN KEY  
("ENCCID")
```

```
    REFERENCES "PROJETO1"."ENCOMENDA CLIENTE" ("ENCCID") ENABLE;
```

---

-- Ref Constraints for Table FATURAEN FORNECEDOR

---

```
ALTER TABLE "PROJETO1"."FATURAEN FORNECEDOR" ADD CONSTRAINT "FK_ENCFID" FOREIGN  
KEY ("ENCFID")
```

```
    REFERENCES "PROJETO1"."ENCOMENDA FORNECEDOR" ("ENCFID") ENABLE;
```

-- Ref Constraints for Table FORNECEDOR

---

```
ALTER TABLE "PROJETO1"."FORNECEDOR" ADD CONSTRAINT "FK_CODPOSTALFORN" FOREIGN KEY  
("CODPOSTAL")
```

```
    REFERENCES "PROJETO1"."CODIGO_POSTAL" ("CODPOSTAL") ENABLE;
```

---

-- Ref Constraints for Table FUNCIONARIO

---

```
ALTER TABLE "PROJETO1"."FUNCIONARIO" ADD CONSTRAINT "FK_TIPOFUNC" FOREIGN KEY  
("TIPO")
```

```
    REFERENCES "PROJETO1"."TIPOFUNCIONARIO" ("IDTIPO") ENABLE;
```

---

-- Ref Constraints for Table LINHAENCOMENDA CLIENTE

---

```
ALTER TABLE "PROJETO1"."LINHAENCOMENDACLIENTE" ADD CONSTRAINT "FK_REFPROD"  
FOREIGN KEY ("REFPROD")
```

```
    REFERENCES "PROJETO1"."PRODUTO" ("REFERENCIA") ENABLE;
```

```
ALTER TABLE "PROJETO1"."LINHAENCOMENDACLIENTE" ADD CONSTRAINT "FK_ENCCID2" FOREIGN  
KEY ("ENCCID")
```

```
    REFERENCES "PROJETO1"."ENCOMENDACLIENTE" ("ENCCID") ENABLE;
```

---

-- Ref Constraints for Table LINHAENCOMENDA FORNECEDOR

---

```
ALTER TABLE "PROJETO1"."LINHAENCOMENDA FORNECEDOR" ADD CONSTRAINT  
"FK_REFPRODENC" FOREIGN KEY ("REFPROD")
```

REFERENCES "PROJETO1"."PRODUTO" ("REFERENCIA") ENABLE;

ALTER TABLE "PROJETO1"."LINHAENCOMENDAFORNECEDOR" ADD CONSTRAINT "FK\_ENCFIDENC"  
FOREIGN KEY ("ENCFID")

REFERENCES "PROJETO1"."ENCOMENDAFORNECEDOR" ("ENCFID") ENABLE;

## Inserts na base de dados

Insert into PROJETO1.CODIGO\_POSTAL (CODPOSTAL, LOCALIDADE) values ('4900-531', 'Viana do Castelo');

Insert into PROJETO1.CLIENTE (IDCLIENTE, NOME, NIF, TELEFONE, RUA, NUMPORTA, CODPOSTAL)  
values ('1', 'João Gonçalves', '264114536', '915435442', 'Rua dos Santos', '563', '4490-445');

Insert into PROJETO1.CLIENTE (IDCLIENTE, NOME, NIF, TELEFONE, RUA, NUMPORTA, CODPOSTAL)  
values ('2', 'Filipe Oliveira', '255143884', '934453114', 'Avenida Central', '75', '4900-531');

Insert into PROJETO1.CLIENTE (IDCLIENTE, NOME, NIF, TELEFONE, RUA, NUMPORTA, CODPOSTAL)  
values ('3', 'Maria Filipa', '243553144', '916432431', 'Rua da escola', '134', '4490-445');

Insert into PROJETO1.ENCOMENDA CLIENTE (ENCCID, VALORTOTAL, DATA, IDCLIENTE) values  
(1, '130', to\_date('19-12-2021', 'DD-MM-YYYY'), 1);

Insert into PROJETO1.ENCOMENDA CLIENTE (ENCCID, VALORTOTAL, DATA, IDCLIENTE) values  
(2, '25', to\_date('19-12-2021', 'DD-MM-YYYY'), 1);

Insert into PROJETO1.ENCOMENDA CLIENTE (ENCCID, VALORTOTAL, DATA, IDCLIENTE) values  
(3, '250', to\_date('17-12-2021', 'DD-MM-YYYY'), 2);

Insert into PROJETO1.ENCOMENDA CLIENTE (ENCCID, VALORTOTAL, DATA, IDCLIENTE) values  
(4, '367', to\_date('18-12-2021', 'DD-MM-YYYY'), 3);

Insert into PROJETO1.TIPOFUNCIONARIO (IDTIPO, CARGO) values ('1', 'Balcao');

Insert into PROJETO1.TIPOFUNCIONARIO (IDTIPO, CARGO) values ('2', 'Gestor');

Insert into PROJETO1.FUNCIONARIO (IDFUNCIONARIO, NOME, TIPO) values ('1', 'Bruno', 1);

Insert into PROJETO1.FUNCIONARIO (IDFUNCIONARIO, NOME, TIPO) values ('2', 'Carlos', 2);

Insert into PROJETO1.ENCOMENDA FORNECEDOR  
(ENCFID, VALORTOTAL, DATA, IDFORNECEDOR, IDFUNCIONARIO) values ('1', '430', to\_date('19-12-  
2021', 'DD-MM-YYYY'), 1, 1);

Insert into PROJETO1.ENCOMENDA FORNECEDOR  
(ENCFID, VALORTOTAL, DATA, IDFORNECEDOR, IDFUNCIONARIO) values ('2', '780', to\_date('17-12-  
2021', 'DD-MM-YYYY'), 1, 2);

Insert into PROJETO1.ENCOMENDAFORNECEDOR  
 (ENCFID,VALORTOTAL,DATA,IDFORNECEDOR,IDEFUNCIONARIO) values ('3','1535',to\_date('17-12-2021','DD-MM-YYYY'),'2','2');

Insert into PROJETO1.ENCOMENDAFORNECEDOR  
 (ENCFID,VALORTOTAL,DATA,IDFORNECEDOR,IDEFUNCIONARIO) values ('4','1105',to\_date('18-12-2021','DD-MM-YYYY'),'2','1');

## Consultas na base de dados

select \* from cliente;

IDCLIENTE	NOME	NIF	TELEFONE	RUA	NUMPORTA	CODPOSTAL
1	1 João Gonçalves	264114536	915435442	Rua dos Santos	563	4490-445
2	2 Filipe Oliveira	255143884	934453114	Avenida Central	75	4900-531
3	3 Maria Filipa	243553144	916432431	Rua da escola	134	4490-445

select \* from cliente order by nome asc;

IDCLIENTE	NOME	NIF	TELEFONE	RUA	NUMPORTA	CODPOSTAL
1	2 Filipe Oliveira	255143884	934453114	Avenida Central	75	4900-531
2	1 João Gonçalves	264114536	915435442	Rua dos Santos	563	4490-445
3	3 Maria Filipa	243553144	916432431	Rua da escola	134	4490-445

select \* from fornecedor;

IDFORNECEDOR	NOME	NIF	TELEFONE	RUA	NUMPORTA	CODPOSTAL
1	Fashion Povoa LDA	300600300	252400200	Rua das flores	5	4490-445
2	Clothes Norte LDA	200500700	252700333	Rua praia norte	21	4900-531

select \* from funcionario;

IDFUNCIONARIO	NOME	TIPO
1	Bruno	1
2	Carlos	2

select \* from tipofuncionario;

IDTIPO	CARGO
1	Balcao
2	Gestor

select c.nome, ec.valortotal as "Valor Total", ec.data from cliente c, encomendacliente ec  
 where c.idcliente = ec.idcliente;

NOME	Nº ENC	Valor Total	DATA
João Gonçalves	1	130	19-12-2021
João Gonçalves	2	25	19-12-2021
Filipe Oliveira	3	250	17-12-2021
Maria Filipa	4	367	18-12-2021

```
select ef.encfid as nºEnc, ef.valortotal as "Valor Total €", f.nome, func.nome from
encomendafornecedor ef, funcionario func, fornecedor f
```

```
where f.idfornecedor = ef.idfornecedor and ef.idfuncionario = func.idfuncionario;
```

NºENC	Valor Total €	NOME	NOME
1	430 Fashion Povoa LDA	Bruno	
2	780 Fashion Povoa LDA	Carlos	
3	1535 Clothes Norte LDA	Carlos	
4	1105 Clothes Norte LDA	Bruno	

```
select c.nome, e.valortotal as "Total da Encomenda" from cliente c, encomendacliente e
where c.idcliente=e.idcliente order by c.nome
```

	NOME	Total da Encomenda
1	Filipe Oliveira	250
2	João Gonçalves	130
3	João Gonçalves	25
4	Maria Filipa	367

```
select ef.encfid, f.nome, func.nome, ef.data, ef.valortotal from encomendafornecedor ef, fornecedor
f, funcionario func
```

```
where ef.idfornecedor=f.idfornecedor and ef.idfuncionario=func.idfuncionario order by ef.valortotal
DESC
```

ENCFID	NOME	NOME_1	DATA	VALORTOTAL
1	3 Clothes Norte LDA	Carlos	17-12-2021	1535
2	4 Clothes Norte LDA	Bruno	18-12-2021	1105
3	2 Fashion Povoa LDA	Carlos	17-12-2021	780
4	1 Fashion Povoa LDA	Bruno	19-12-2021	430

## Trigger na base de dados

```

CREATE OR REPLACE TRIGGER ATUALIZASTOCK
AFTER INSERT ON FATURAENCLIENTE
BEGIN
    UPDATE PRODUTO
    SET QUANTIDADE = QUANTIDADE - 1;
    NULL;
END;

```

TRIGGER_NAME	TRIGGER_TYPE	TRIGGER_OWNER	TRIGGERING_EVENT	STATUS	TABLE_NAME
1 FATURAENCLIENTE_TRG BEFORE EACH ROW PROJETO1	INSERT	PROJETO1	ENABLED	FATURAENCLIENTE	
2 ATUALIZASTOCK	AFTER STATEMENT PROJETO1	PROJETO1	ENABLED	FATURAENCLIENTE	

```

| Refresh: 0 |
| TRIGGER ATUALIZASTOCK
| AFTER INSERT ON FATURAENCLIENTE
| BEGIN
|     UPDATE PRODUTO
|     SET QUANTIDADE = QUANTIDADE - 1;
|     NULL;
| END;
| 
```

## 6. Conclusão e Trabalho Futuro

Este presente relatório referente à disciplina de projeto I, demonstra durante o seu percurso do desenvolvimento, os conhecimentos que adquirimos durante as aulas e da ajuda do professor sempre que surgiam dúvidas, assim nos guiando para um caminho onde estaríamos corretos na resolução dos nossos problemas.

Em relação aos objetivos propostos neste relatório, todos foram completados e consequentemente, graças a isso, conseguimos aprender ainda mais sobre o que engloba este projeto, tanto de Engenharia de Software, como também de Base de dados.

Durante o todo o processo de desenvolvimento deste relatório, o que mais surgiu como dificuldade foi os diagramas de sequência, pois tivemos pouco treino na resolução deste tipo de diagramas e acabou por nos atrasar um bocado para o desenvolvimento deste projeto I.

Em suma, apesar de todas as dificuldades que tivemos durante este longo período, felizmente todas essas dificuldades, vistas como obstáculos, foram ultrapassadas, assim contribuindo positivamente para a nossa aprendizagem.