

**Licenciatura em Engenharia Informática**

Escola Superior de Tecnologia e Gestão

*Instituto Politécnico de Viana do Castelo*

Unidade curricular de

# Projeto I

Relatório de projeto

2023/2024

*Gestão de Produção de Whisky*

Tiago Baptista Nº28998

Pedro Poças Nº28976

## Índice

<b>1. Introdução.....</b>	<b>4</b>
1.1 Apresentação do tema .....	4
1.2 Objetivos do projeto.....	4
<b>2. Apresentação do negócio .....</b>	<b>5</b>
2.1 Âmbito e enquadramento do negócio .....	5
2.2 Modelação dos Processos de negócio.....	5
2.2.1 Processo 1 - Gestão de encomenda da matéria-prima.....	6
.....	6
2.2.2 Processo 2 - Gestão produção de Whisky .....	7
2.2.3 Processo 3 - Receção da encomenda e verificação de qualidade .....	9
2.2.4 Processo 4 - Gestão financeira.....	9
<b>3. Levantamento de Requisitos .....</b>	<b>10</b>
3.1 Tipos de utilizador .....	10
3.2 Requisitos Funcionais .....	10
3.3 Requisitos não funcionais.....	11
<b>4. Design e Modelação .....</b>	<b>12</b>
4.1 Modelo de casos de uso .....	12
4.1.1 Descrição casos de uso .....	12
4.2 Modelo de classes do Domínio.....	13
4.2.1 Descrição Modelo Classes .....	14
4.3 Diagramas de transição de estados .....	15
4.4 Especificação de Casos de Uso.....	16
4.4.1 Especificação usando Template de Casos de Uso .....	16
4.4.1.1 Caso de Uso -> Receber Encomenda .....	16
4.4.1.2 Caso de Uso -> Consultar Stock.....	17
4.4.1.3 Caso de Uso -> Realizar encomenda .....	18
4.4.1.4 Caso de Uso-> Estado empresa .....	19
4.4.2 Especificação usando Diagramas de atividades .....	20
4.4.2.1 Caso de Uso -> Receber encomenda .....	20
4.4.2.2 Caso de Uso -> Consultar Stock.....	20
4.4.2.3 Caso de Uso -> Estado empresa .....	21
4.4.2.4 Caso de Uso -> Realizar encomenda .....	21
4.4.3 Especificação usando Diagramas de sequência.....	22
4.4.3.1 Caso de Uso -> Gestor de produção efetua encomenda .....	22
<b>5 Desenho e Implementação da BD .....</b>	<b>23</b>
5.3 Modelação de dados .....	23
5.4 Modelo Físico da Base de dados .....	25
5.5 Tecnologias usadas e Criação das Base de Dados .....	26
5.6 Queries à Base de Dados .....	36
<b>6 Conclusão e Trabalho Futuro.....</b>	<b>41</b>

## Índice de Figuras

<b>Figura 1 - Modelo BPMN da encomenda de matéria-prima</b>	<b>6</b>
<b>Figura 2 - Modelo BPMN produção</b>	<b>7</b>
Figura 3 - Subprocesso "Maltagem"	7
Figura 4 - Subprocesso "Moagem"	7
Figura 5 - Subprocesso "Misturação"	8
Figura 6 - Subprocesso "Fermentação"	8
Figura 7 - Subprocesso "Destilação"	8
Figura 8 - Subprocesso "Maturação"	8
Figura 9 - Modelo BPMN receção e validação	9
Figura 10 - Modelo BPMN estado financeiro	9
Figura 11 - Modelação de casos de uso	12
Figura 12 - Modelo de Classes de Domínio	14
Figura 13 - Diagrama de Transição de Estados	15
Figura 14 - Diagrama de Atividade "Receber Encomenda"	20
Figura 15 - Diagrama de Atividade "Consultar Stock"	20
Figura 16 - Diagrama de Atividade "Estado Encomenda"	21
Figura 17 - Diagrama de Atividade "Realizar Encomenda"	21
Figura 18 - Diagrama de Sequência "Encomenda"	22
Figura 19 - Modelo de Entidade e Relacionamento Inicial	23
Figura 20 - Modelo Entidade e Relacionamento Final	24
Figura 21 - Modelo de Base de Dados pgAdmin 4	25
Figura 22 - Tabela "fatura"	26
Figura 23 - Tabela "tipofuncionario"	26
Figura 24 - Tabela "tipoproduto"	27
Figura 25 - Tabela "fornecedor"	27
Figura 26 - Tabela "estadoencomenda"	28
Figura 27 - Tabela "faseproducao"	28
Figura 28 - Tabela "prodfase"	29
Figura 29 - Tabela "funcionario"	30
Figura 30 - Tabela "loteproduzido"	31
Figura 31 - Tabela "encomenda"	32
Figura 32 - Tabela "loterecebido"	33
Figura 33 - Tabela "producao"	34
Figura 34 - Tabela "loteinput"	35
Figura 35 - Tabela "tipofuncionario"	38
Figura 36 - Tabela "fornecedor"	38
Figura 37 - Tabela "producao"	38
Figura 38 - Tabela "encomenda" em ordem crescente de envio	39
Figura 39 - Soma de valores na tabela "preco"	39
Figura 40 - Nomes dos Rececionistas	39
Figura 41 - View "view_salariofuncionario"	40

## 1. Introdução

### 1.1 Apresentação do tema

Na Unidade Curricular de Projeto I, foi-nos proposto o tema da "Gestão da Produção de Whisky". Esta tarefa, embora complexa, envolve a criação de uma das bebidas mais adoradas e emblemáticas do mundo. Neste relatório, permitirá investigar e compreender os principais aspetos envolvidos na gestão da produção de whisky, desde a seleção cuidadosa dos ingredientes até a entrega do produto final.

### 1.2 Objetivos do projeto

Este projeto visa aprimorar os modelos de negócio de uma fábrica de whiskey, criando uma base de dados capaz de suportar as operações e facilite a tomada de decisões estratégicas.

## 2. Apresentação do negócio

### 2.1 Âmbito e enquadramento do negócio

Este negócio visa a criação da bebida Whisky com junção de tradição e inovação nas suas diversas etapas de produção.

A partir dos grãos que são selecionados criteriosamente até aos diversos processos da sua produção esta obra segue um protocolo rigorosamente para que chegue ao final esperado ao final do envelhecimento.

Com isto, ficamos cientes e empolgados com o potencial que poderá ter o negócio na indústria de whisky.

### 2.2 Modelação dos Processos de negócio

#### **Aquisição de Ingredientes (Encomenda):**

Compra de grãos de cevada.

#### **Processos na produção do whisky:**

##### Maltagem:

- Preparação dos grãos para a produção de malte.

##### Moagem:

- Trituração dos grãos para criar uma mistura de grãos triturados, conhecida como grist.

##### Misturação:

- Mistura do grist com água quente para criar um mash;
- Extração dos açúcares do grist.

##### Fermentação:

- Adição de levedura ao mash para iniciar a fermentação;
- Conversão dos açúcares em álcool.

Destilação:

- Destilação do líquido fermentado para separar os componentes alcoólicos;
- Pode incluir uma ou várias destilações, dependendo do tipo de whiskey.

Maturação:

- Envelhecimento do destilado em barris de carvalho por um período de anos;
- Desenvolvimento do sabor e da cor do whiskey durante esse processo.

Engarrafamento e Rotulagem:

- Engarrafamento e rotulagem do produto final.

Controle de Qualidade:

- Testes regulares de qualidade para garantir a consistência do produto.

Armazenamento de Barris:

- Gestão do stock de barris de envelhecimento.

Gestão Financeira:

- Controle de custos e orçamento;
- Análise de rentabilidade e projeções financeiras.

## 2.2.1 Processo 1 - Gestão de encomenda da matéria-prima

O Gestor de encomendas vai comunicar com os fornecedores para poder efetuar a encomenda da matéria-prima necessária para a produção dos Whisky.

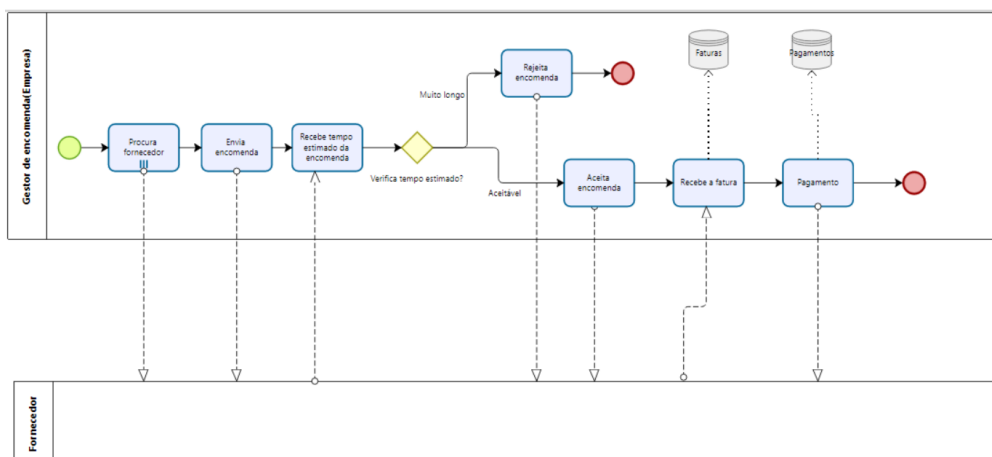


Figura 1 - Modelo BPMN da encomenda de matéria-prima

## 2.2.2 Processo 2 - Gestão produção de Whisky

A empresa vai efetuar a produção de Whisky, se não tiver stock para começar a produção vai comunicar ao gestor de encomendas para comunicar ao fornecedor, se já tiver stock segue para os processos e subprocessos para a produção de Whisky.

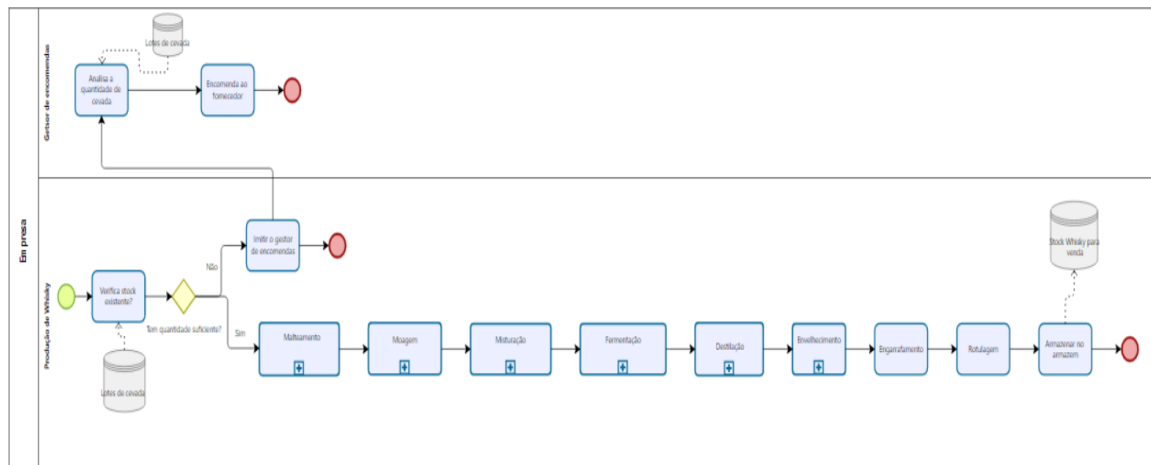


Figura 2 - Modelo BPMN produção

### Subprocessos:

#### Maltagem:

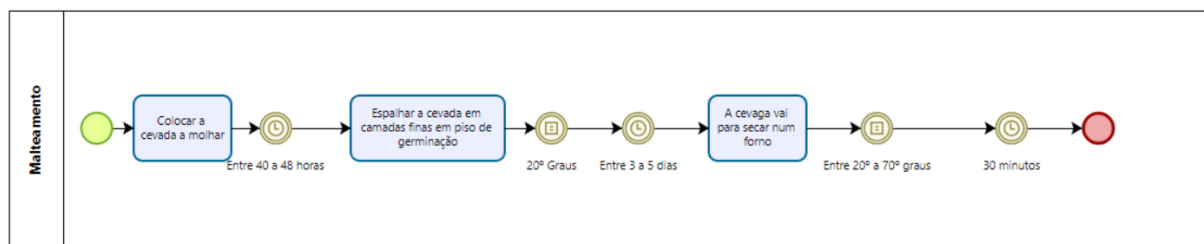


Figura 3 - Subprocesso "Maltagem"

#### Moagem:

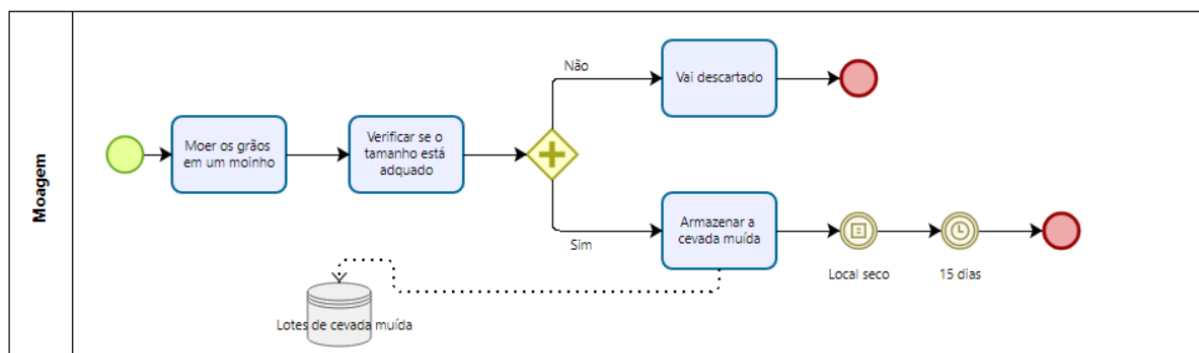


Figura 4 - Subprocesso "Moagem"

Misturação:

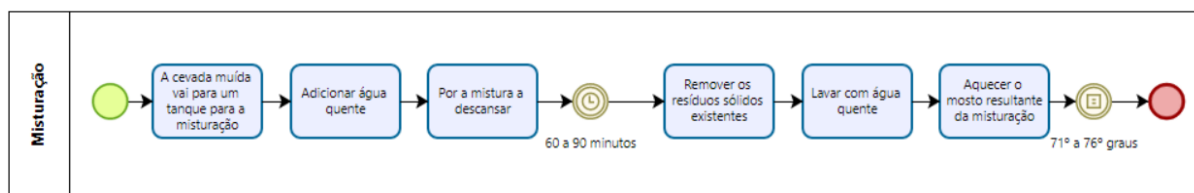


Figura 5 - Subprocesso "Misturação"

Fermentação:

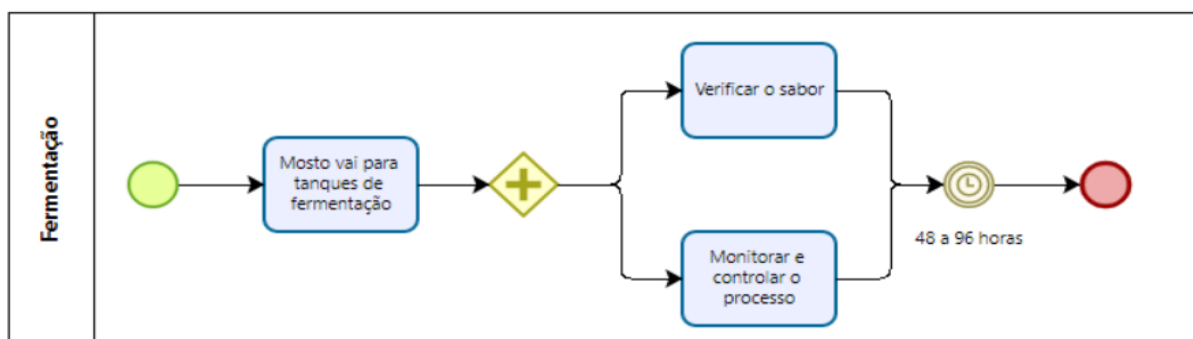


Figura 6 - Subprocesso "Fermentação"

Destilação:

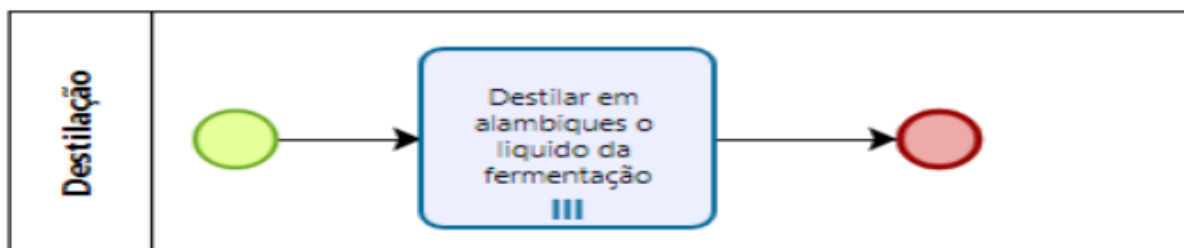


Figura 7 - Subprocesso "Destilação"

Maturação:

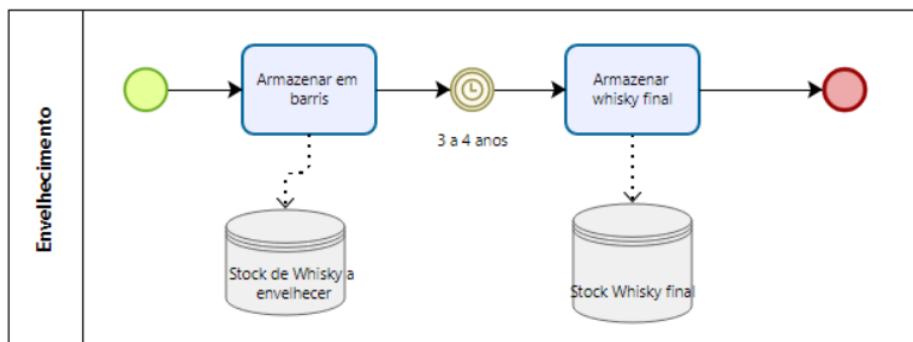


Figura 8 - Subprocesso "Maturação"



### 2.2.3 Processo 3 - Receção da encomenda e verificação de qualidade

A receção vai receber a encomenda efetuada pelo gestor de encomendas e também vai verificar se está tudo em conformidade com o fornecedor e com a matéria-prima.

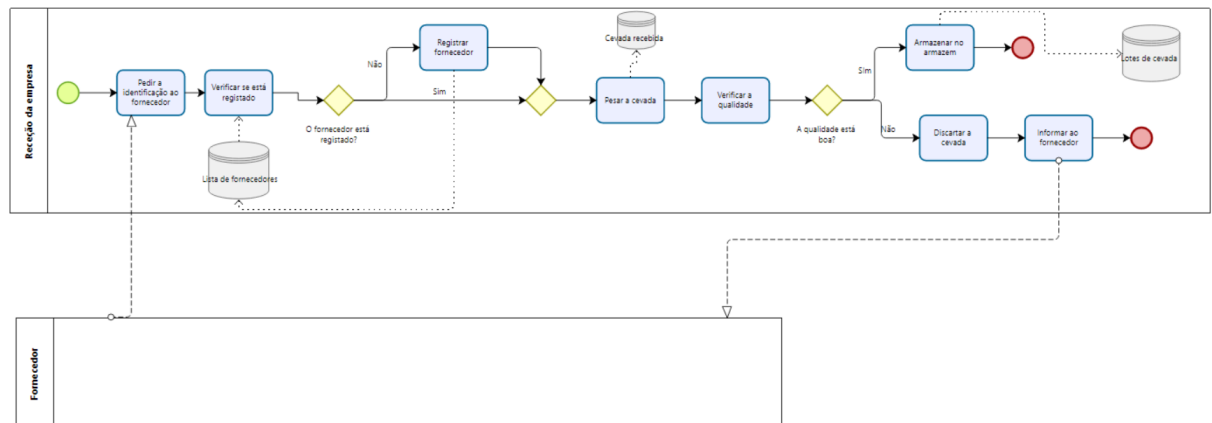


Figura 9 - Modelo BPMN receção e validação

### 2.2.4 Processo 4 - Gestão financeira

O gestor da empresa vai analisar as bases de dados e efetua o relatório para verificar o estado financeiramente da empresa.

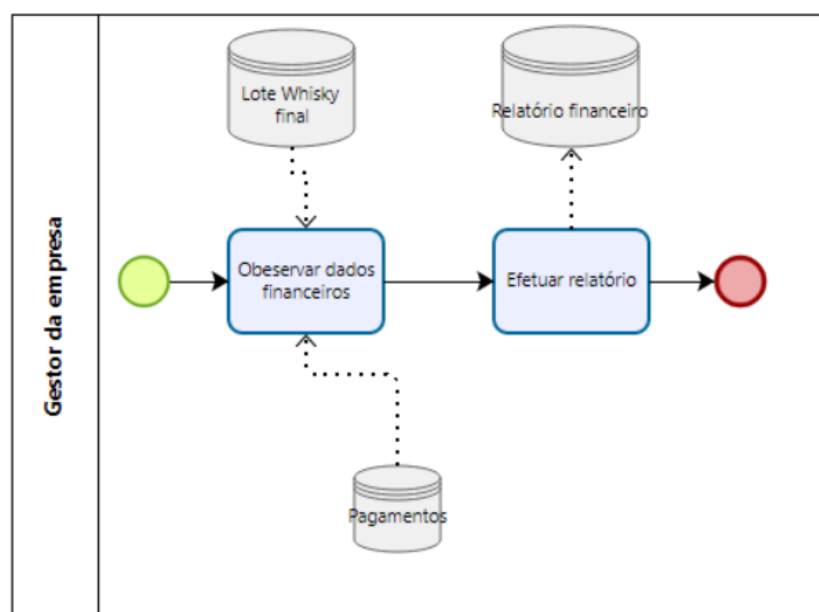


Figura 10 - Modelo BPMN estado financeiro

### 3. Levantamento de Requisitos

#### 3.1 Tipos de utilizador

Nesta aplicação os utilizadores presentes são: o Gestor de encomendas, funcionários da produção, rececionista e Gestor financeiro.

#### 3.2 Requisitos Funcionais

ID	Prioridade	Requisitos Funcionais
RF1	1	Como gestor de encomendas, quero encontrar um fornecedor, para encomendar a matéria-prima.
RF2	1	Como gestor de encomendas, quero armazenar as faturas e os respetivos pagamentos das encomendas, para registá-los nas bases de dados.
RF3	1	Como funcionário da produção, quero verificar se tem stock de cevada, para poder proceder a produção.
RF4	1	Como funcionário da produção, quero armazenar a cevada miúda, para a registar na respetiva base de dados.
RF5	1	Como funcionário da produção, quero armazenar o stock de whisky a envelhecer, para registar na respetiva base de dados.
RF6	2	Como funcionário da produção, quero descartar coisas defeituosas, para garantir a qualidade.
RF7	1	Como funcionário da produção, quero armazenar o Whisky produzido, para registar na respetiva base de dados.
RF8	1	Como rececionista, quero aceder á base de dados dos fornecedores, para se necessário registá-lo.
RF9	1	Como rececionista, quero pesar a cevada encomendada, para registar na respetiva base de dados.
RF10	1	Como rececionista, quero avaliar a qualidade da cevada, para (Passar para não funcional)
RF11	1	Como gestor financeiro, quero aceder ás bases de dados dos pagamentos e os lotes de whisky final, para realizar o relatório financeiro da empresa.

### 3.3 Requisitos não funcionais

ID	Prioridade	Requisitos não funcionais
RNF1	2	O sistema deve garantir armazenar diversos dados nas bases de dados existentes.
RNF2	1	As consultas de base de dados devem ser executadas em um tempo breve.
RNF3	1	O sistema deve controlar o acesso de pessoal não autorizado às informações das bases de dados.
RNF4	1	O sistema deve ser intuitivo e fácil de usar para os trabalhadores da empresa.
RNF5	1	O sistema deve comunicar com a base de dados postgres.
RNF6	1	O sistema deve gerar automaticamente relatórios financeiros se o gestor financeiro pretender.

## 4. Design e Modelação

### 4.1 Modelo de casos de uso



**Figura 11 - Modelação de casos de uso**

#### 4.1.1 Descrição casos de uso

**Receber encomenda:** O rececionista na receção da encomenda poderá vir a registar o fornecedor se necessário, nisto avalia a encomenda. Caso for aceite será registado a matéria-prima da encomenda.

**Realizar encomenda:** O gestor de encomenda inicia o processo para encomendar matéria-prima, para isso é necessário encontrar um fornecedor que esteja em conformidade para tal efetuar.

**Registrar faturas/pagamentos:** O gestor de encomendas faz o registo das faturas/pagamentos efetuados.

**Estado empresa:** O gestor financeiro verifica os lotes de whisky final disponíveis e verifica os pagamentos efetuados para a realização do relatório.

**Consultar Stock:** O funcionário de produção realiza as seguintes operações: consultar stock, registar lote de cevada moída, registar lote de whisky a envelhecer e registar lote de whisky final.

#### 4.2 Modelo de classes do Domínio

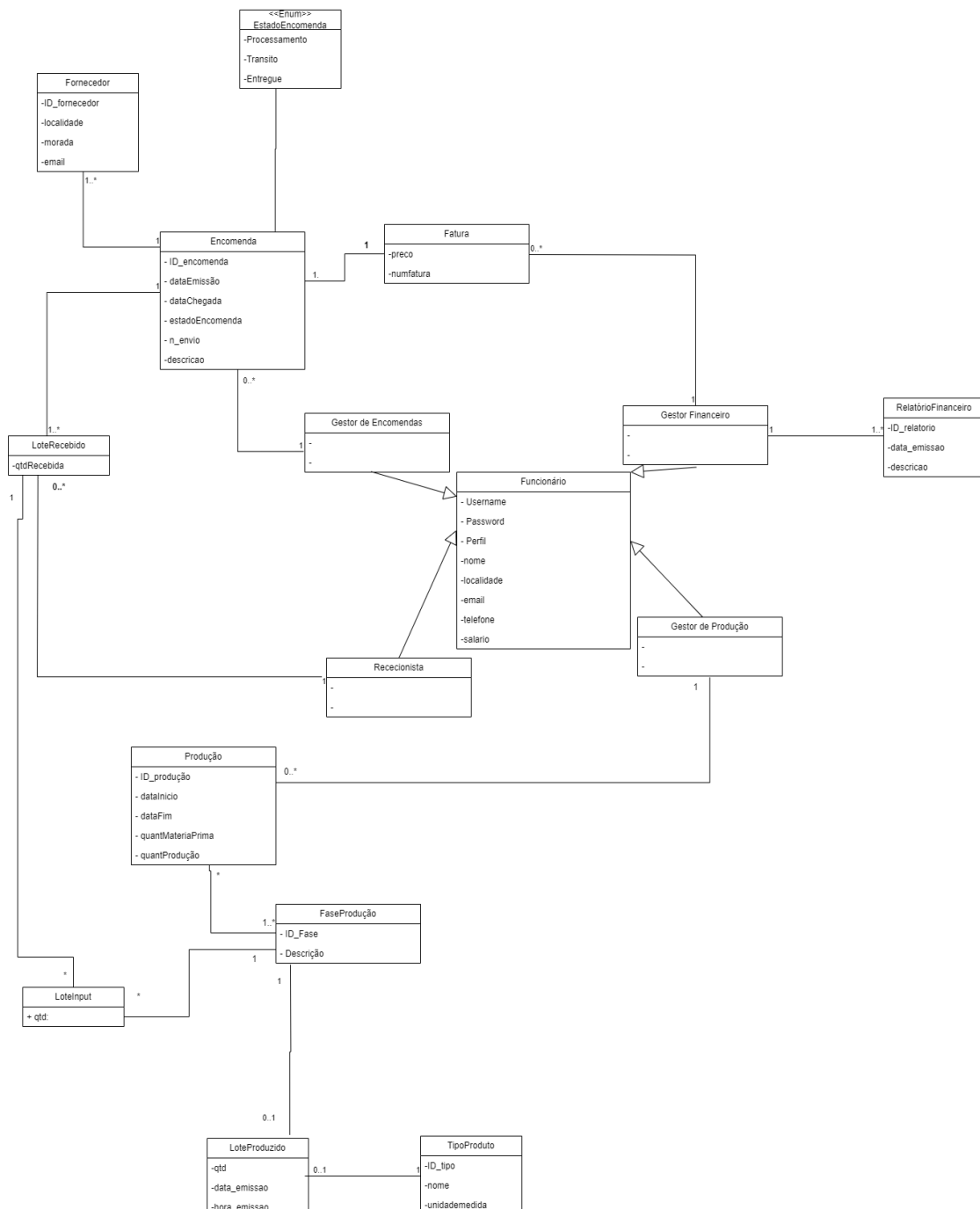


Figura 12 - Modelo de Classes de Domínio

#### 4.2.1 Descrição Modelo Classes

- Um funcionário é composto por vários funcionários (gestor de encomendas, gestor financeiro, rececionista, gestor produção).
- Um gestor financeiro pode realizar um ou vários relatórios financeiros.
- Um relatório financeiro é relativo a um e só um gestor financeiro.
- Um gestor de produção está responsável por nenhuma ou várias produções.
- Uma produção é relativa a um gestor de produção.
- Uma produção tem associada uma ou várias fases de produção.
- Uma fase de produção é relativa a diversas produções.
- Uma fase de produção pode ter nenhum ou um lote produzido.
- Um lote produzido é relativo a uma fase de produção.
- Um lote produzido tem associado um tipo do produto.
- Um tipo de produto poderá ter nenhum ou um lote produzido.
- Um rececionista pode não receber nenhum ou pode receber vários lotes de receção.
- Um lote recebido é relativo a um rececionista.
- Um lote recebido tem vários lotes de input.
- Um lote de input é relativo a um lote recebido.
- Um lote de input é relativo a uma fase de produção.
- Uma fase de produção tem vários lotes de input.
- Um gestor de encomendas poderá ter nenhuma ou várias encomendas.
- Uma encomenda é relativa a um gestor de encomenda.
- Uma encomenda poderá ter um ou vários lotes recebidos.
- Um lote recebido é relativo a uma encomenda.
- Uma encomenda é relativa a um ou vários fornecedores.
- Um fornecedor é relativo a uma encomenda.
- Uma encomenda tem uma enumeração de diferentes estados de uma encomenda.
- Uma encomenda tem associado uma fatura.
- Uma fatura poderá ter uma ou várias encomendas.
- Um gestor financeiro poderá ter nenhuma ou várias faturas.
- Uma fatura é relativa a um gestor financeiro.

#### 4.3 Diagramas de transição de estados

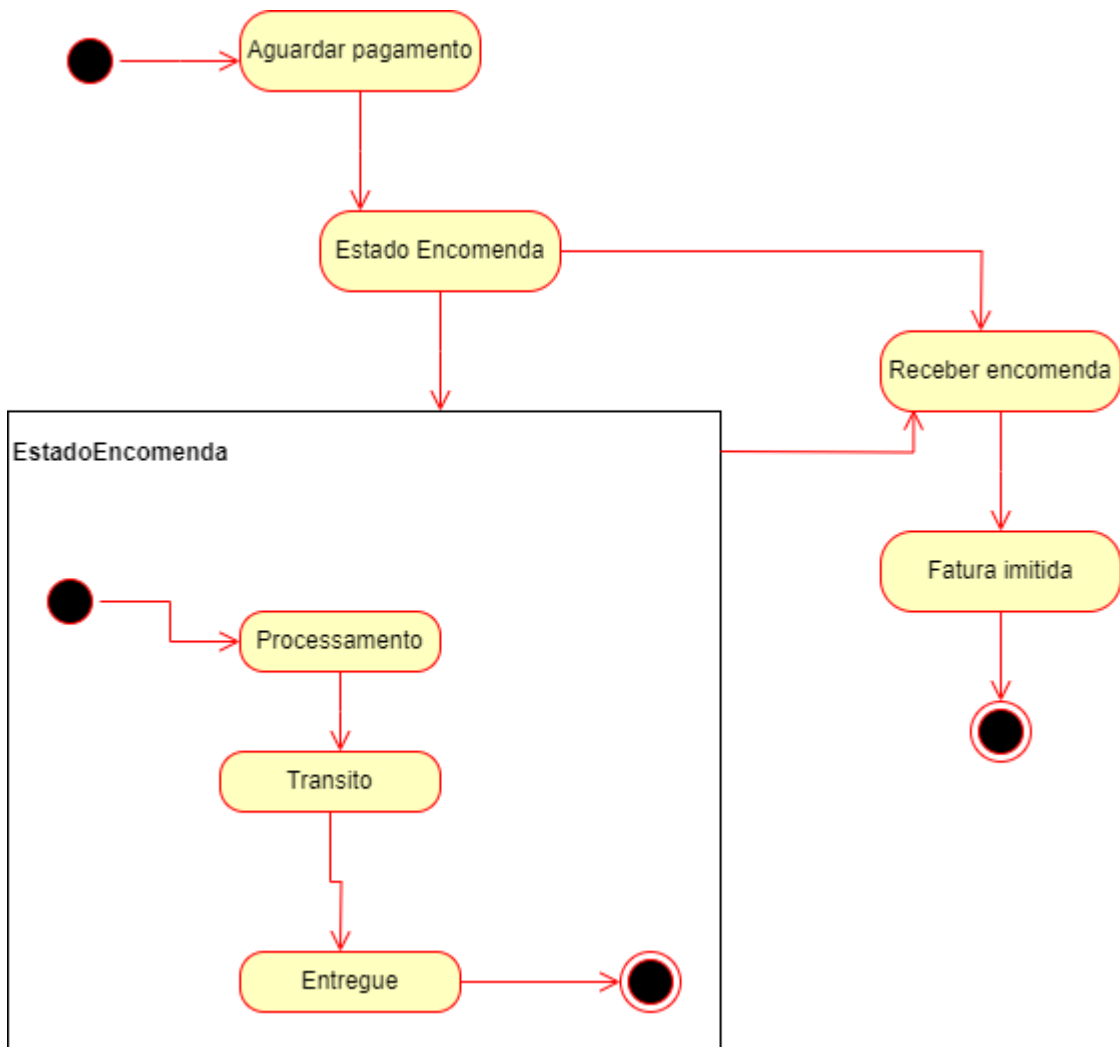


Figura 13 - Diagrama de Transição de Estados

## 4.4 Especificação de Casos de Uso

### 4.4.1 Especificação usando Template de Casos de Uso

#### 4.4.1.1 Caso de Uso -> Receber Encomenda

<b>ID:</b>	<i>Caso de Uso: Receber encomenda</i>
<b>Actor Principal:</b>	<i>Rececionista</i>
<b>Pré-Condições:</b>	<i>O rececionista deve estar registado no sistema</i>
<b>Pós-Condições:</b>	
<b>Cenário Principal:</b>	<ol style="list-style-type: none"> <li>1. <i>O Rececionista recebe encomenda.</i></li> <li>2. <i>O Rececionista verifica se o fornecedor está registado.</i></li> <li>3. <i>O Rececionista avalia a encomenda.</i></li> <li>4. <i>O Rececionista aceita encomenda.</i></li> <li>5. <i>O rececionista regista peso de matéria-prima.</i></li> </ol>
<b>Cenários Alternativos:</b>	<p><i>2.O fornecedor não está registado no sistema</i></p> <p><i>2.1 O rececionista regista fornecedor no sistema.</i></p> <p><i>4. O rececionista recusa encomenda</i></p> <p><i>4.1 O rececionista informa ao fornecedor o motivo.</i></p>
<b>Exceções</b>	



#### 4.4.1.2 Caso de Uso -> Consultar Stock

ID:	Caso de Uso: Consultar Stock
Actor Principal:	Funcionário Produção
Pré-Condições:	O funcionário deve estar registado no sistema
Pós-Condições:	
Cenário Principal:	1. O funcionário consulta stock em sistema.
Cenário Alternativos:	
Exceções	

#### 4.4.1.3 Caso de Uso->Registar lote cevada moída

ID:	Caso de Uso : Registar lote cevada moída
Actor Principal:	Funcionário Produção
Pré-Condições:	O funcionário da produção deve estar registado no sistema
Pós-Condições:	Criar um novo registo de cevada moída
Cenário Principal:	1. O funcionário de produção regista lote de cevada moída no sistema.
Cenário Alternativo:	
Exceção:	

#### 4.4.1.4 Caso de Uso -> Registar lote whisky a envelhecer

ID:	Caso de Uso : Registar lote whisky envelhecer
Actor Principal:	Funcionário Produção
Pré-condições:	O funcionário da produção deve estar registado no sistema
Pós-Condições:	Criar um novo registo de whisky a envelhecer
Cenário Principal:	1.O funcionário de produção regista lote de whisky a envelhecer no sistema.
Cenário Alternativo:	

Exceção:	
----------	--

#### 4.4.1.5 Caso de Uso -> Registrar lote whisky final

ID:	Caso de Uso: Registrar lote whisky final
Actor Principal:	Funcionário Produção
Pré-condições:	O funcionário da produção deve estar registado no sistema
Pós-Condições:	Criar um novo registo de lote de whisky final
Cenário-principal:	1. O funcionário da produção regista lote de whisky final no sistema.
Cenário-Alternativo:	
Exceção:	

#### 4.4.1.6 Caso de Uso -> Realizar encomenda

<i>ID:</i>	<i>Caso de Uso: Realizar encomenda</i>
<i>Actor Principal:</i>	<i>Gestor de encomendas</i>
<i>Pré-Condições:</i>	<i>O gestor de encomendas deve estar registado no sistema</i>
<i>Pós-Condições:</i>	<i>Criar uma nova encomenda</i>
<i>Cenário Principal:</i>	1.O gestor de encomenda procura fornecedor. 2. O gestor de encomenda seleciona a matéria-prima. 3.Verifica disponibilidade do fornecedor. 4.O gestor encomenda matéria-prima.
<i>Cenários Alternativos:</i>	4.O gestor não está satisfeito com encomenda 4.1 O gestor cancela encomenda 4.O gestor aceita encomenda 4.2 O gestor paga encomenda
<i>Exceções</i>	

#### 4.4.1.7 Caso de Uso -> Registar faturas/pagamentos

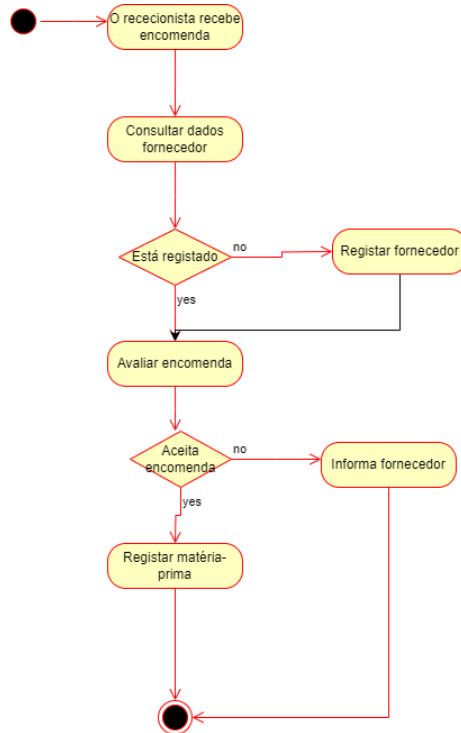
ID:	Caso de Uso: Registar faturas/pagamentos
Atores Principais:	Gestor de Encomenda, Gestor Financeiro
Pré-Condições:	Ambos os gestores devem estar registados no sistema
Pós-Condições:	Criar novo registo de fatura/pagamentos
Cenário Principal:	1.O gestor de encomenda e financeiro efetuam registos de faturas/pagamentos no sistema
Cenário Alternativo:	
Exceção	

#### 4.4.1.8 Caso de Uso-> Estado empresa

ID:	Caso de Uso: Estado empresa
Actor Principal:	Gestor Financeiro
Pré-Condições:	O gestor deve estar registado no sistema
Pós-Condições:	
Cenário Principal:	1.O gestor consulta lotes de whisky final. 2.O gestor verifica pagamentos. 3.O gestor efetua relatório financeiro.
Cenários Alternativos:	
Exceções	

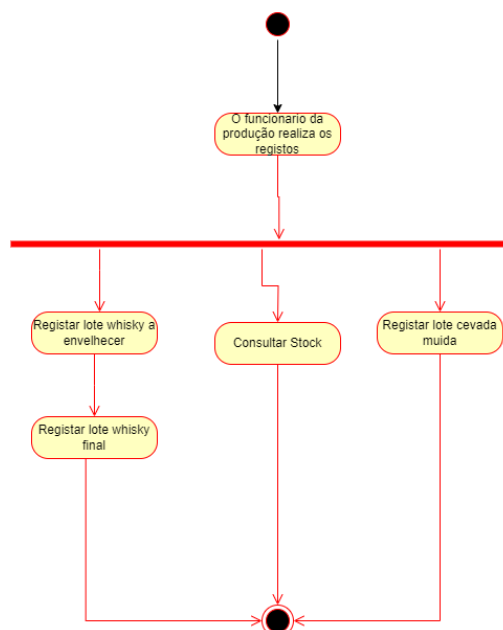
#### 4.4.2 Especificação usando Diagramas de atividades

##### 4.4.2.1 Caso de Uso -> Receber encomenda



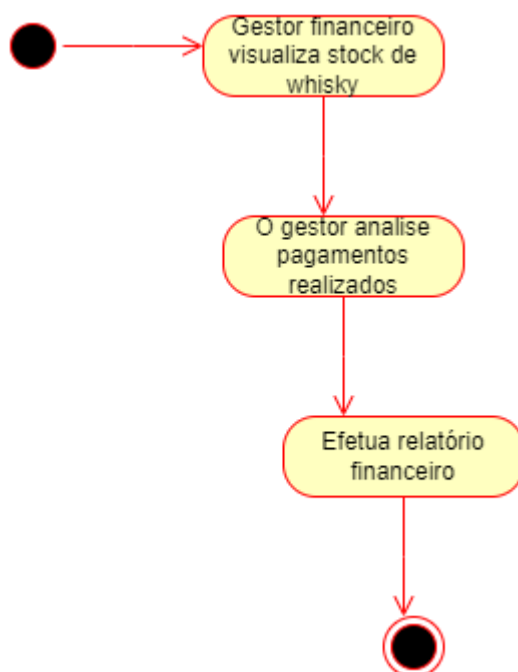
**Figura 14 - Diagrama de Atividade  
"Receber Encomenda"**

##### 4.4.2.2 Caso de Uso -> Consultar Stock



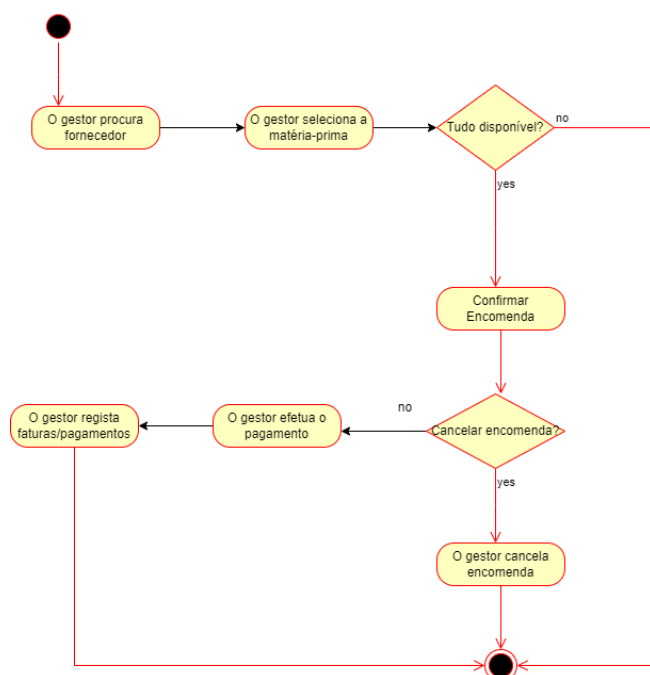
**Figura 15 - Diagrama de Atividade  
"Consultar Stock"**

#### 4.4.2.3 Caso de Uso -> Estado empresa



**Figura 16 - Diagrama de Atividade "Estado Encomenda"**

#### 4.4.2.4 Caso de Uso -> Realizar encomenda



**Figura 17 - Diagrama de Atividade "Realizar Encomenda"**

#### 4.4.3 Especificação usando Diagramas de sequência

##### 4.4.3.1 Caso de Uso -> Gestor de produção efetua encomenda

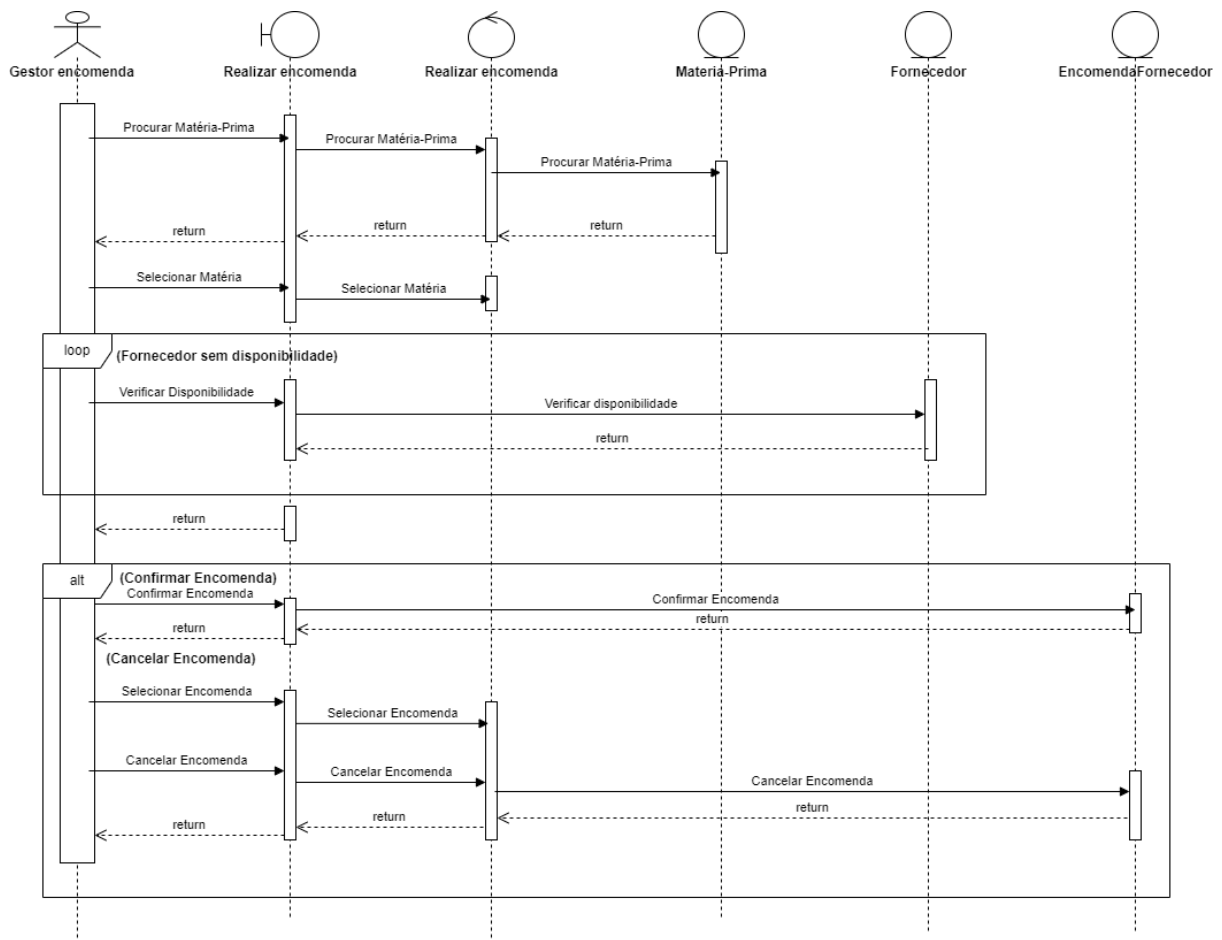
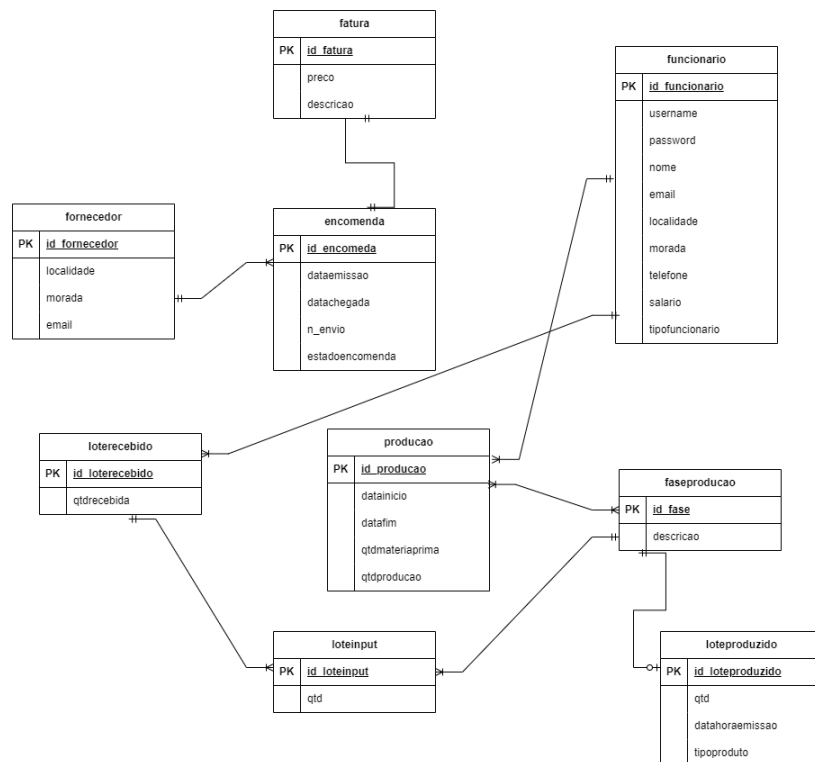


Figura 18 - Diagrama de Sequência "Encomenda"

## 5 Desenho e Implementação da BD

### 5.3 Modelação de dados



**Figura 19 - Modelo de Entidade e Relacionamento Inicial**

#### Esquema de Tabelas

funcionario (**id\_funcionario**, username, password, nome, email, localidade, morada, telefone, salario, tipofuncionario)

fornecedor (**id\_fornecedor**, localidade, morada, email)

encomenda (**id\_encomenda**, dataemissao, datachegada, n\_envio, estadoencomenda, id\_fornecedor, id\_funcionario)

fatura (**id\_fatura**, preco, descricao)

producao (**id\_producao**, datainicio, datafim, qtdmateria prima, qtdproducao, id\_funcionario)

lote recebido (**id\_lote recebido**, qtdrecebida, id\_funcionario, id\_encomenda)

fase producao (**id\_fase**, descricao)

lote input (**id\_lote input**, qtd, id\_fase, id\_lote recebido)

loteproduzido (**id\_loteproduzido**, qtd, datahoraemissao, tipoproduto)

### O Esquema de Tabelas na 3FN

- **1FN** (Eliminar a repetição de grupos de informação):

tipofuncionario (**id\_tipofuncionario**, tipo)

estadoencomenda (**id\_estadoencomenda**, tipoestado)

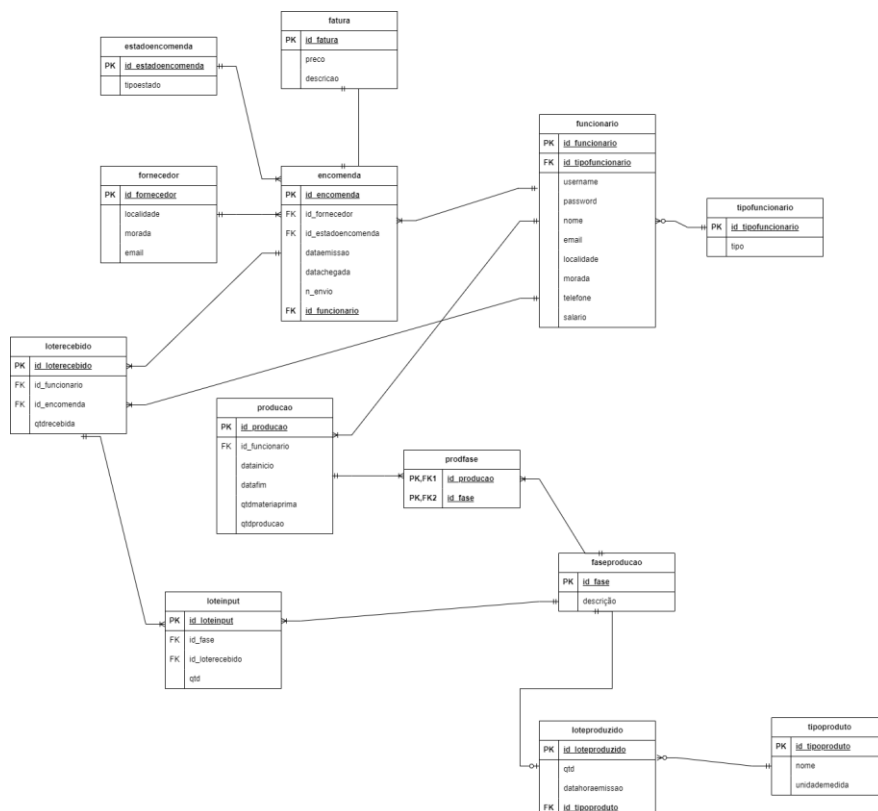
tipoproduto (**id\_tipoproduto**, nome, unidademedida)

- **2FN** (Evitar a redundância de dados):

prodfase (**id\_producao**, **id\_fase**)

- **3FN** (Eliminar dependências transitivas):

Não existe dependências!



**Figura 20 - Modelo Entidade e Relacionamento Final**



## 5.4 Modelo Físico da Base de dados

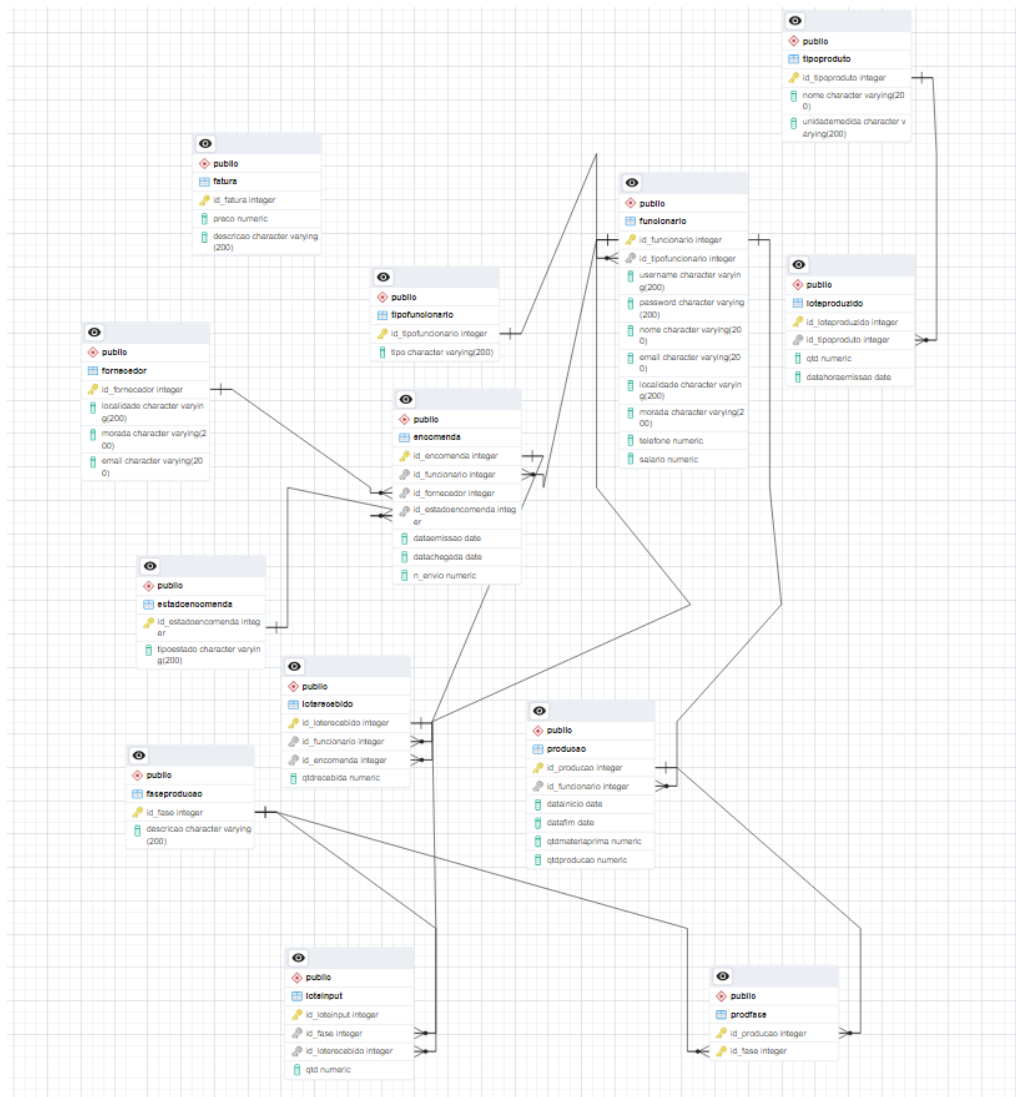


Figura 21 - Modelo de Base de Dados pgAdmin 4

## 5.5 Tecnologias usadas e Criação das Base de Dados

```
CREATE TABLE IF NOT EXISTS public.fatura  
(  
    id_fatura integer NOT NULL DEFAULT nextval('fatura_id_fatura_seq'::regclass),  
    preco numeric NOT NULL,  
    descricao character varying (200) COLLATE pg_catalog."default" NOT NULL,  
    CONSTRAINT fatura_pkey PRIMARY KEY (id_fatura)  
)
```

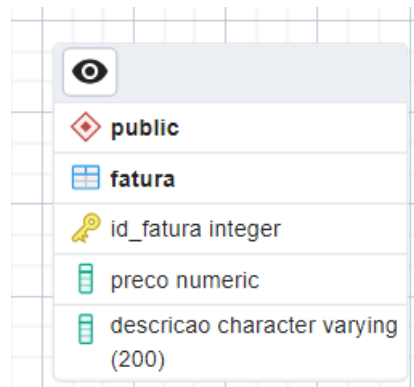


Figura 22 - Tabela "fatura"

```
CREATE TABLE IF NOT EXISTS public.tipofuncionario  
(  
    id_tipofuncionario integer NOT NULL DEFAULT  
    nextval('tipofuncionario_id_tipofuncionario_seq'::regclass),  
    tipo character varying (200) COLLATE pg_catalog."default" NOT NULL,  
    CONSTRAINT tipofuncionario_pkey PRIMARY KEY (id_tipofuncionario)  
)
```

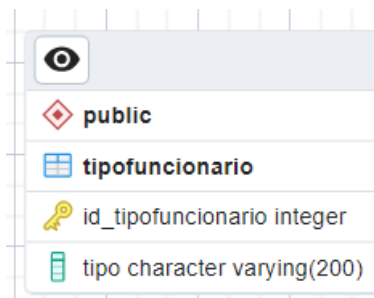
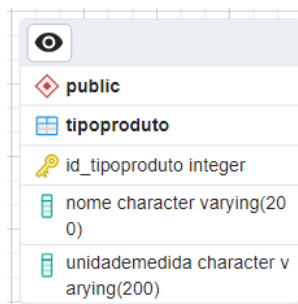


Figura 23 - Tabela  
"tipofuncionario"

```

CREATE TABLE IF NOT EXISTS public.tipoproduto
(
    id_tipoproduto integer NOT NULL DEFAULT
nextval('tipoproduto_id_tipoproduto_seq'::regclass),
    nome character varying (200) COLLATE pg_catalog."default" NOT NULL,
    unidademedida character varying (200) COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT tipoproduto_pkey PRIMARY KEY (id_tipoproduto)
)

```

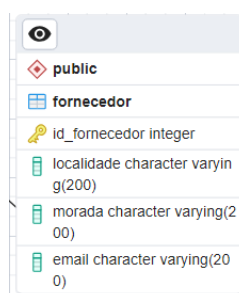


**Figura 24 - Tabela  
"tipoproduto"**

```

CREATE TABLE IF NOT EXISTS public.fornecedor
(
    id_fornecedor integer NOT NULL DEFAULT
nextval('fornecedor_id_fornecedor_seq'::regclass),
    localidade character varying(200) COLLATE pg_catalog."default" NOT NULL,
    morada character varying (200) COLLATE pg_catalog."default" NOT NULL,
    email character varying (200) COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT fornecedor_pkey PRIMARY KEY (id_fornecedor)
)

```



**Figura 25 - Tabela  
"fornecedor"**

```
CREATE TABLE IF NOT EXISTS public.estadoencomenda
(
    id_estadoencomenda integer NOT NULL DEFAULT
nextval('estadoencomenda_id_estadoencomenda_seq'::regclass),
    tipoestado character varying (200) COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT estadoencomenda_pkey PRIMARY KEY (id_estadoencomenda)
)
```



Figura 26 - Tabela  
"estadoencomenda"

```
CREATE TABLE IF NOT EXISTS public.faseproducao
(
    id_fase integer NOT NULL DEFAULT nextval('faseproducao_id_fase_seq'::regclass),
    descricao character varying (200) COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT faseproducao_pkey PRIMARY KEY (id_fase)
)
```

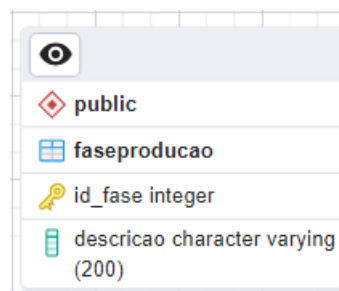


Figura 27 - Tabela  
"faseproducao"

```
CREATE TABLE IF NOT EXISTS public.prodfase  
(  
    id_producao integer NOT NULL DEFAULT  
nextval('prodfase_id_producao_seq'::regclass),  
    id_fase integer NOT NULL DEFAULT nextval('prodfase_id_fase_seq'::regclass),  
    CONSTRAINT prodfase_pkey PRIMARY KEY (id_producao, id_fase),  
    CONSTRAINT fk_id_fase FOREIGN KEY (id_fase)  
        REFERENCES public.faseproducao (id_fase) MATCH SIMPLE  
        ON UPDATE NO ACTION  
        ON DELETE NO ACTION  
        NOT VALID,  
    CONSTRAINT fk_id_producao FOREIGN KEY (id_producao)  
        REFERENCES public.producao (id_producao) MATCH SIMPLE  
        ON UPDATE NO ACTION  
        ON DELETE NO ACTION  
        NOT VALID  
)
```

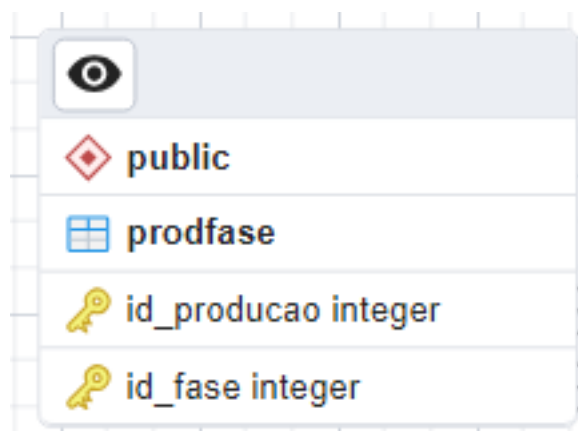


Figura 28 - Tabela "prodfase"

```

CREATE TABLE IF NOT EXISTS public.funcionario
(
    id_funcionario integer NOT NULL DEFAULT
nextval('funcionario_id_funcionario_seq'::regclass),

    id_tipofuncionario integer NOT NULL,

    username character varying (200) COLLATE pg_catalog."default" NOT NULL,

    password character varying (200) COLLATE pg_catalog."default" NOT NULL,

    nome character varying (200) COLLATE pg_catalog."default" NOT NULL,

    email character varying (200) COLLATE pg_catalog."default" NOT NULL,

    localidade character varying (200) COLLATE pg_catalog."default" NOT NULL,

    morada character varying (200) COLLATE pg_catalog."default" NOT NULL,

    telefone numeric NOT NULL,

    salario numeric NOT NULL,

    CONSTRAINT funcionario_pkey PRIMARY KEY (id_funcionario),

    CONSTRAINT fk_id_tipofuncionario FOREIGN KEY (id_tipofuncionario)

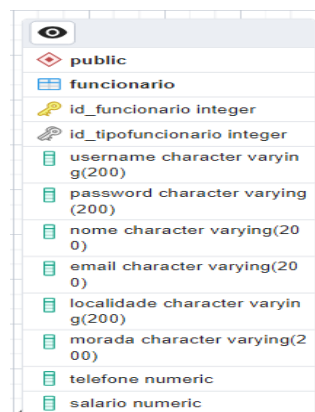
    REFERENCES public.tipofuncionario (id_tipofuncionario) MATCH SIMPLE

    ON UPDATE NO ACTION

    ON DELETE NO ACTION

    NOT VALID
)

```



Column Name	Data Type	Constraints
id_funcionario	Integer	Primary Key
id_tipofuncionario	Integer	Foreign Key to public.tipofuncionario
username	character varying(200)	NOT NULL
password	character varying(200)	NOT NULL
nome	character varying(200)	NOT NULL
email	character varying(200)	NOT NULL
localidade	character varying(200)	NOT NULL
morada	character varying(200)	NOT NULL
telefone	numeric	NOT NULL
salario	numeric	NOT NULL

Figura 29 - Tabela  
"funcionario"

```
CREATE TABLE IF NOT EXISTS public.loteproduzido  
(  
    id_loteproduzido integer NOT NULL DEFAULT  
nextval('loteproduzido_id_loteproduzido_seq'::regclass),  
    id_tipoproduto integer NOT NULL,  
    qtd numeric NOT NULL,  
    datahoraemissao date NOT NULL,  
    CONSTRAINT loteproduzido_pkey PRIMARY KEY (id_loteproduzido),  
    CONSTRAINT fk_id_tipoproduto FOREIGN KEY (id_tipoproduto)  
    REFERENCES public.tipoproduto (id_tipoproduto) MATCH SIMPLE  
    ON UPDATE NO ACTION  
    ON DELETE NO ACTION  
    NOT VALID  
)
```

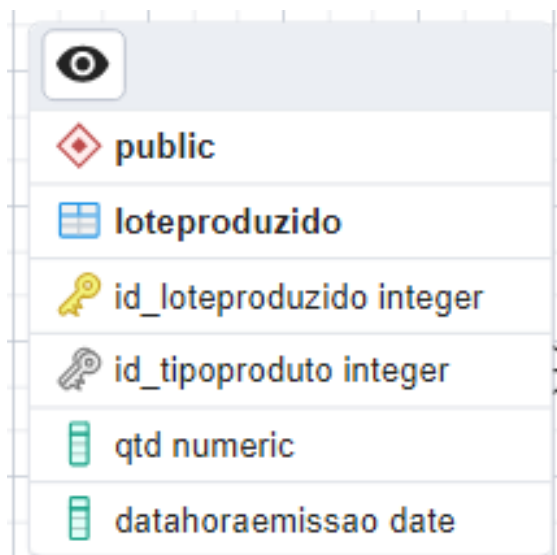


Figura 30 - Tabela "loteproduzido"

```
CREATE TABLE IF NOT EXISTS public.encomenda
```

```
(
```

```
    id_encomenda integer NOT NULL DEFAULT  
    nextval('encomenda_id_encomenda_seq'::regclass),
```

```
    id_funcionario integer NOT NULL,
```

```
    id_fornecedor integer NOT NULL,
```

```
    id_estadoencomenda integer NOT NULL,
```

```
    dataemissao date NOT NULL,
```

```
    datachegada date NOT NULL,
```

```
    n_envio numeric NOT NULL,
```

```
    CONSTRAINT encomenda_pkey PRIMARY KEY (id_encomenda),
```

```
    CONSTRAINT fk_id_estadoencomenda FOREIGN KEY (id_estadoencomenda)
```

```
        REFERENCES public.estadoencomenda (id_estadoencomenda) MATCH SIMPLE
```

```
        ON UPDATE NO ACTION
```

```
        ON DELETE NO ACTION
```

```
        NOT VALID,
```

```
    CONSTRAINT fk_id_fornecedor FOREIGN KEY (id_fornecedor)
```

```
        REFERENCES public.fornecedor (id_fornecedor) MATCH SIMPLE
```

```
        ON UPDATE NO ACTION
```

```
        ON DELETE NO ACTION
```

```
        NOT VALID,
```

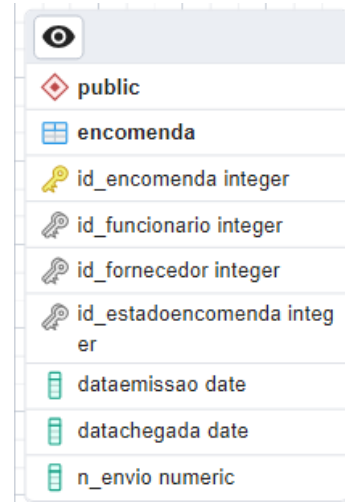
```
    CONSTRAINT fk_id_funcionario FOREIGN KEY (id_funcionario)
```

```
        REFERENCES public.funcionario (id_funcionario) MATCH SIMPLE
```

```
        ON UPDATE NO ACTION
```

```
        ON DELETE NO ACTION
```

```
        NOT VALID
```



Schema	Table	Column	Data Type	Constraints
public	encomenda	id_encomenda	integer	Primary Key
public	encomenda	id_funcionario	integer	
public	encomenda	id_fornecedor	integer	
public	encomenda	id_estadoencomenda	integer	
public	encomenda	dataemissao	date	
public	encomenda	datachegada	date	
public	encomenda	n_envio	numeric	

**Figura 31 - Tabela  
"encomenda"**



```
CREATE TABLE IF NOT EXISTS public.loterecebido  
(  
    id_loterecebido integer NOT NULL DEFAULT  
nextval('loterecebido_id_loterecebido_seq'::regclass),  
    id_funcionario integer NOT NULL,  
    id_encomenda integer NOT NULL,  
    qtdrecebida numeric NOT NULL,  
    CONSTRAINT loterecebido_pkey PRIMARY KEY (id_loterecebido),  
    CONSTRAINT fk_id_encomenda FOREIGN KEY (id_encomenda)  
        REFERENCES public.encomenda (id_encomenda) MATCH SIMPLE  
        ON UPDATE NO ACTION  
        ON DELETE NO ACTION  
        NOT VALID,  
    CONSTRAINT fk_id_funcionario FOREIGN KEY (id_funcionario)  
        REFERENCES public.funcionario (id_funcionario) MATCH SIMPLE  
        ON UPDATE NO ACTION  
        ON DELETE NO ACTION  
        NOT VALID  
)
```

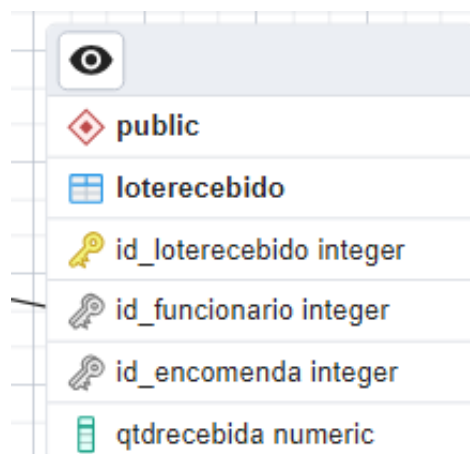


Figura 32 - Tabela "loterecebido"

```
CREATE TABLE IF NOT EXISTS public.producao
(
    id_producao integer NOT NULL DEFAULT
nextval('producao_id_producao_seq'::regclass),
    id_funcionario integer NOT NULL,
    datainicio date NOT NULL,
    datafim date NOT NULL,
    qtdmateriaprima numeric NOT NULL,
    qtdproducao numeric NOT NULL,
    CONSTRAINT producao_pkey PRIMARY KEY (id_producao),
    CONSTRAINT fk_id_funcionario FOREIGN KEY (id_funcionario)
REFERENCES public.funcionario (id_funcionario) MATCH SIMPLE
ON UPDATE NO ACTION
ON DELETE NO ACTION
NOT VALID
)
```

public
producao
id_producao integer
id_funcionario integer
datainicio date
datafim date
qtdmateriaprima numeric
qtdproducao numeric

Figura 33 - Tabela "producao"

```
CREATE TABLE IF NOT EXISTS public.loteinput
(
    id_loteinput integer NOT NULL DEFAULT
nextval('loteinput_id_loteinput_seq'::regclass),
    id_fase integer NOT NULL,
    id_loterecebido integer NOT NULL,
    qtd numeric NOT NULL,
    CONSTRAINT loteinput_pkey PRIMARY KEY (id_loteinput),
    CONSTRAINT fk_id_fase FOREIGN KEY (id_fase)
        REFERENCES public.faseproducao (id_fase) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID,
    CONSTRAINT fk_id_loterecebido FOREIGN KEY (id_loterecebido)
        REFERENCES public.loterecebido (id_loterecebido) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
)
```

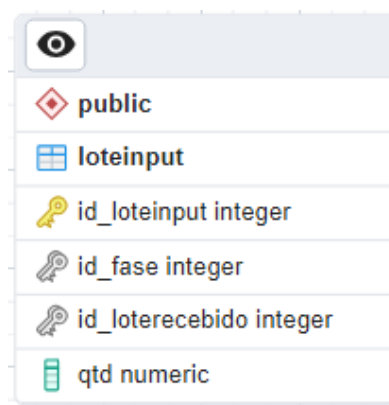


Figura 34 - Tabela "loteinput"

## 5.6 Queries à Base de Dados

### Inserts na Base de Dados

Exemplos de inserção de dados nas tabelas:

- **encomenda**

```
insert into encomenda (id_funcionario, id_fornecedor, id_estadoencomenda,  
dataemissao, datachegada, n_envio) values (2, 1, 1, '2023-09-28 07:00:47', '2023-10-27  
14:30:07', 3470);
```

```
insert into encomenda (id_funcionario, id_fornecedor, id_estadoencomenda,  
dataemissao, datachegada, n_envio) values (2, 10, 1, '2023-09-06 13:43:49', '2023-10-15  
21:42:36', 3892).
```

- **estadoencomenda**

```
insert into estadoencomenda (tipoestado) values ('Aguardar pagamento');
```

```
insert into estadoencomenda (tipoestado) values ('Processamento').
```

- **faseproducao**

```
insert into faseproducao (descricao) values ('Malteamento');
```

```
insert into faseproducao (descricao) values ('Moagem').
```

- **fatura**

```
insert into fatura (preco, descricao) values (1568, 'at nulla suspendisse potenti cras in  
purus');
```

```
insert into fatura (preco, descricao) values (1763, 'posuere felis sed lacus morbi sem').
```

- **fornecedor**

```
insert into fornecedor (localidade, morada, email) values ('Nantes', '92 Morrow Hill',  
'vbausmann0@macromedia.com');
```

```
insert into fornecedor (localidade, morada, email) values ('Arraial do Cabo', '73 Randy  
Point', 'nhurrion1@4shared.com').
```

- **funcionario**

```
insert into funcionario (id_tipofuncionario, username, password, nome, email,  
localidade, morada, telefone, salario) values (2, 'adenison0', 'qW2"oU"qV', 'Adriana Denison',  
'adenison0@japanpost.jp', 'Krasnokamensk', '361 Banding Plaza', '6374456962', 2014);
```

```
insert into funcionario (id_tipofuncionario, username, password, nome, email,
localidade, morada, telefone, salario) values (1, 'ssprigg1', 'mW8|5> (E&', 'Siffre Sprigg',
'ssprigg1@marriott.com', 'Huai Mek', '26483 Utah Center', '8918631773', 2333).
```

- **loteinput**

```
insert into loteinput (id_fase, id_loterecebido, qtd) values (1, 1, 2093);
```

```
insert into loteinput (id_fase, id_loterecebido, qtd) values (1, 2, 2063).
```

- **loteproduzido**

```
insert into loteproduzido (id_tipoproduto, qtd, datahoraemissao) values (1, 49, '2024-
03-16 05:05:46');
```

```
insert into loteproduzido (id_tipoproduto, qtd, datahoraemissao) values (2, 50, '2024-
03-06 02:13:15').
```

- **loterecebido**

```
insert into loterecebido (id_funcionario, id_encomenda, qtdrecebida) values (4, 1,
901);
```

```
insert into loterecebido (id_funcionario, id_encomenda, qtdrecebida) values (4, 2,
626).
```

- **prodfase**

```
insert into prodfase (id_producao, id_fase) values (1, 4);
```

```
insert into prodfase (id_producao, id_fase) values (2, 1).
```

- **producao**

```
insert into producao (id_funcionario, datainicio, datafim, qtdmateriaprima,
qtdproducao) values (1, '2024-01-19 22:11:58', '2024-02-15 20:16:57', 389, 2765);
```

```
insert into producao (id_funcionario, datainicio, datafim, qtdmateriaprima,
qtdproducao) values (1, '2024-01-18 18:20:21', '2024-02-16 14:59:25', 171, 2543).
```

- **tipofuncionario**

```
insert into tipofuncionario (tipo) values ('Gestor de Producao');
```

```
insert into tipofuncionario (tipo) values ('Gestor de Encomendas').
```

- **tipoproduto**

```
insert into tipoproduto (nome, unidademedida) values ('cevada', 'kg');
```

```
insert into tipoproduto (nome, unidademedida) values ('cevada_muida', 'kg').
```

## Consultas base de dados

- **SELECT \* FROM tipofuncionario;**

Ao executar este comando, a base de dados fornece todas as linhas e colunas da tabela “tipofuncionario”.

id_tipofuncionario	tipo
1	Gestor de Producao
2	Gestor de Encomendas
3	Gestor financeiro
4	Rececionista

(4 rows)

Figura 35 - Tabela "tipofuncionario"

- **SELECT \* FROM fornecedor;**

Semelhante ao comando anterior, os dados fornecidos são provenientes da tabela “fornecedor”.

id_fornecedor	localidade	morada	email
1	Nantes	92 Morrow Hill	vbausmann0@macromedia.com
2	Arraial do Cabo	73 Randy Point	nhurrian1@4shared.com
3	El Matama	3 3rd Court	asleney2@house.gov
4	Kofu-shi	336 Florence Hill	jbinch3@github.io
5	Satipo	4634 Raven Place	gturri4@wordpress.org
6	Próchoma	72 Bobwhite Place	hvell5@hhs.gov
7	Jajarm	2 Hanover Point	eabsalom6@cyberchimps.com
8	Funchal	10 Hintze Park	rtantrum7@mac.com
9	Belmullet	1 Everett Court	rkemmett8@ted.com
10	Huangtan	734 Del Sol Terrace	cpiddlehinton9@reverbNation.com

Figura 36 - Tabela "fornecedor"

- **SELECT \* FROM producao;**

As linhas e colunas representadas na imagem seguinte fazem parte da tabela “producao”.

id_producao	id_funcionario	datainicio	datafim	qtdmateriaprima	qtdproducao
1	1	2024-01-19	2024-02-15	389	2765
2	1	2024-01-18	2024-02-16	171	2543
3	1	2024-01-04	2024-02-05	116	2990
4	1	2024-01-21	2024-02-19	866	3045
5	1	2024-01-13	2024-02-21	351	3141
6	1	2024-01-11	2024-02-16	628	3212
7	1	2024-01-08	2024-02-23	1897	2073
8	1	2024-01-30	2024-02-16	142	3597

Figura 37 - Tabela "producao"

- `SELECT * FROM encomenda ORDER BY n_envio ASC;`

Com a execução deste comando, todos os registos da tabela “**encomenda**” serão apresentados em ordem ascendente (asc) com base no valor da coluna “**n\_envio**”.

id_encomenda	id_funcionario	id_fornecedor	id_estadoencomenda	dataemissao	datachegada	n_envio
7	2	5	2	2023-09-12	2023-10-07	2444
4	2	8	1	2023-09-20	2023-10-15	2602
5	2	3	1	2023-09-17	2023-11-03	2623
1	2	5	3	2023-09-05	2023-10-06	2647
6	2	2	3	2023-09-10	2023-10-07	3046
2	2	6	2	2023-09-08	2023-10-22	3227
8	2	8	4	2023-09-28	2023-10-22	3596
10	2	8	4	2023-09-29	2023-10-25	3782
3	2	8	2	2023-09-25	2023-10-16	3948
9	2	3	3	2023-09-16	2023-11-03	3998

(10 rows)

Figura 38 - Tabela "encomenda" em ordem crescente de envio

- `SELECT sum(preco) FROM fatura;`

Através do comando mencionado, podemos obter o resultado que corresponde à soma de todos os valores presentes na coluna “**preco**” da tabela “**fatura**”.

```
sum
-----
99333
```

Figura 39 - Soma de valores na tabela "preco"

- `SELECT f.nome , t.tipo FROM funcionario f, tipofuncionario t WHERE f.id_tipofuncionario = t.id_tipofuncionario AND t.tipo = 'Rececionista';`

Começamos por atribuir os valores de “**f**” e “**t**”, respetivamente, às tabelas “**funcionario**” e “**tipofuncionario**” através de “**FROM funcionario f, tipofuncionario t**”.

Pretendemos selecionar as colunas “**nome**” da tabela “**funcionario**” (f.nome) e “**tipo**” da tabela “**tipofuncionario**” (tipofuncionario t).

Existe uma condição, onde o “**id\_tipofuncionario**” na tabela “**funcionario**” tem de ser o igual ao “**id\_tipofuncionario**” na tabela “**tipofuncionario**” e o “**tipo**” na tabela “**tipofuncionario**” é igual a ‘**Rececionista**’.

Portanto, obtemos os nomes dos funcionários que têm como tipo ‘**Rececionista**’.

nome	tipo
Agustin Birts	Rececionista
Forest Ruffy	Rececionista
Nichole Ales0	Rececionista
Sapphira Kiezler	Rececionista
Thia Farrington	Rececionista
Tasha Donan	Rececionista
Emily Marcam	Rececionista
Vinnie Eyres	Rececionista
Midge Buncom	Rececionista

Figura 40 - Nomes dos Rececionistas

- **CREATE VIEW view\_salariofuncionario AS SELECT nome,email,morada  
FROM funcionario WHERE salario <= 2000;**

Este comando cria uma visão (view) chamada “view\_salariofuncionario” Esta visão representa uma consulta que nos mostra os nomes, emails e moradas dos funcionários cujos salários sejam iguais ou inferiores a 2000.

Uma vez criada a visão, podemos consultar a mesma através do comando a seguir:

```
databaseprojeto1=# select * from view_salariofuncionario;
```

nome	email	morada
Bat McKellen	bmckellen2@eepurl.com	9 Shopko Terrace
Eloisa Hutchin	ehutchin5@example.com	47811 Paget Avenue
Edd Packington	epackingtona@squidoo.com	9346 Carey Place
Gonzalo McMearty	gmcmeartyb@thetimes.co.uk	54355 Lake View Court
Evangelia Goodwell	egoodwelld@nytimes.com	50968 Brickson Park Place
Modestine Montfort	mmontfortg@imageshack.us	65 Dottie Place
Rancell O'Geneay	rogeneayh@ebay.co.uk	88333 Elgar Circle
Prudy Kilius	pkiliusj@artisteer.com	57 Union Court
Danya Wadsworth	dwadsworthk@sun.com	2648 Spohn Drive
Forest Ruffy	fruffyl@altervista.org	8264 Fair Oaks Lane
Aindrea Yakunkin	ayakunkinn@washington.edu	01646 Reinke Court
Robbie Andrea	randreap@lulu.com	7059 Vahlen Parkway
Thia Farrington	tfarringtonr@de.vu	4 Farmco Lane
Tasha Donan	tdonanw@rambler.ru	21 Fulton Drive
Valry Gladdis	vgladdisy@sbwire.com	2891 Upham Street
Inglis Asaaf	iasaaf10@barnesandnoble.com	56537 Browning Avenue
Emily Marcam	emarcam15@google.es	538 Eggendart Park
Marje Lau	mlau19@unesco.org	23074 Bellgrove Lane
Vinnie Eyres	veyres1f@utexas.edu	4254 Valley Edge Center
Midge Buncom	mbuncom2k@biglobe.ne.jp	6 Towne Way
Zea Kerwin	zkerwin2m@cbslocal.com	60968 Service Lane
Syd Witchell	switchell2n@exblog.jp	2 Menomonie Junction

(22 rows)

**Figura 41 - View “view\_salariofuncionario”**



## 6 Conclusão e Trabalho Futuro

Este relatório, elaborado no âmbito da disciplina de Projeto I, ilustra o progresso e a aplicação dos conhecimentos adquiridos ao longo das aulas. Contamos com o apoio constante do professor para esclarecer dúvidas, direcionando-nos para a resolução eficaz dos problemas enfrentados. Os objetivos delineados foram alcançados, permitindo-nos aprofundar nosso entendimento sobre os aspetos fundamentais do projeto, que abrange tanto a disciplina Engenharia de Software I como também Base de Dados.

Especificamente no contexto da gestão da produção de uma fábrica de whiskey, a disciplina de Engenharia de Software I foi essencial para que este projeto nos proporcionasse uma compreensão profunda dos processos de fabrico, controle de qualidade e logística que para tal foram realizados todos os BPMN (Business Process Model and Notation) do negócio em si. Foi envolvido um levantamento detalhado de requisitos que nos fez entender as necessidades do sistema de informação a ser desenvolvido e, posteriormente, dedicamo-nos à modelagem de casos de uso, o que foi fundamental para ilustrar as diversas funcionalidades do sistema e como os usuários interagem com ele.

A fase de desenho e implementação da base de dados foi uma etapa crucial, onde aplicamos técnicas e aprendizagens provenientes da disciplina de Base de Dados, com o objetivo de suportar todos os processos operacionais da fábrica, desde a aquisição da matéria-prima até a distribuição do produto desejado.

Para concluir, este projeto não só solidificou o nosso conhecimento em Engenharia de Software e Bases de Dados, como também contribuiu para uma maior preparação para enfrentar desafios futuros. É importante destacar que este projeto também proporcionou a oportunidade de lidar com situações do mundo real, como a gestão da produção em uma fábrica de whiskey, que vai além do ambiente académico.