

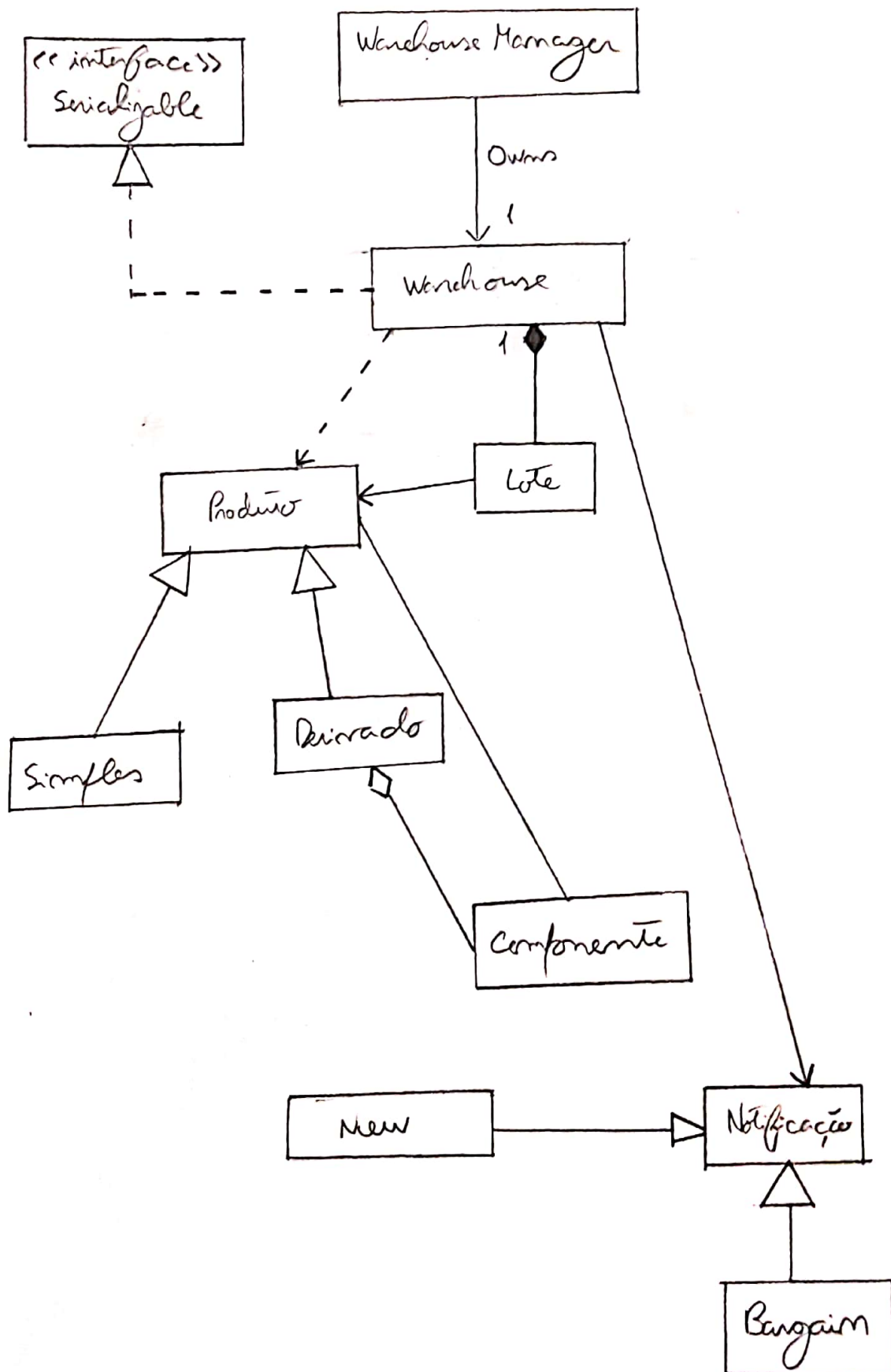
Declaro por minha honra que este diagrama  
foi realizado apenas pelos elementos que  
constituem o grupo do projeto.

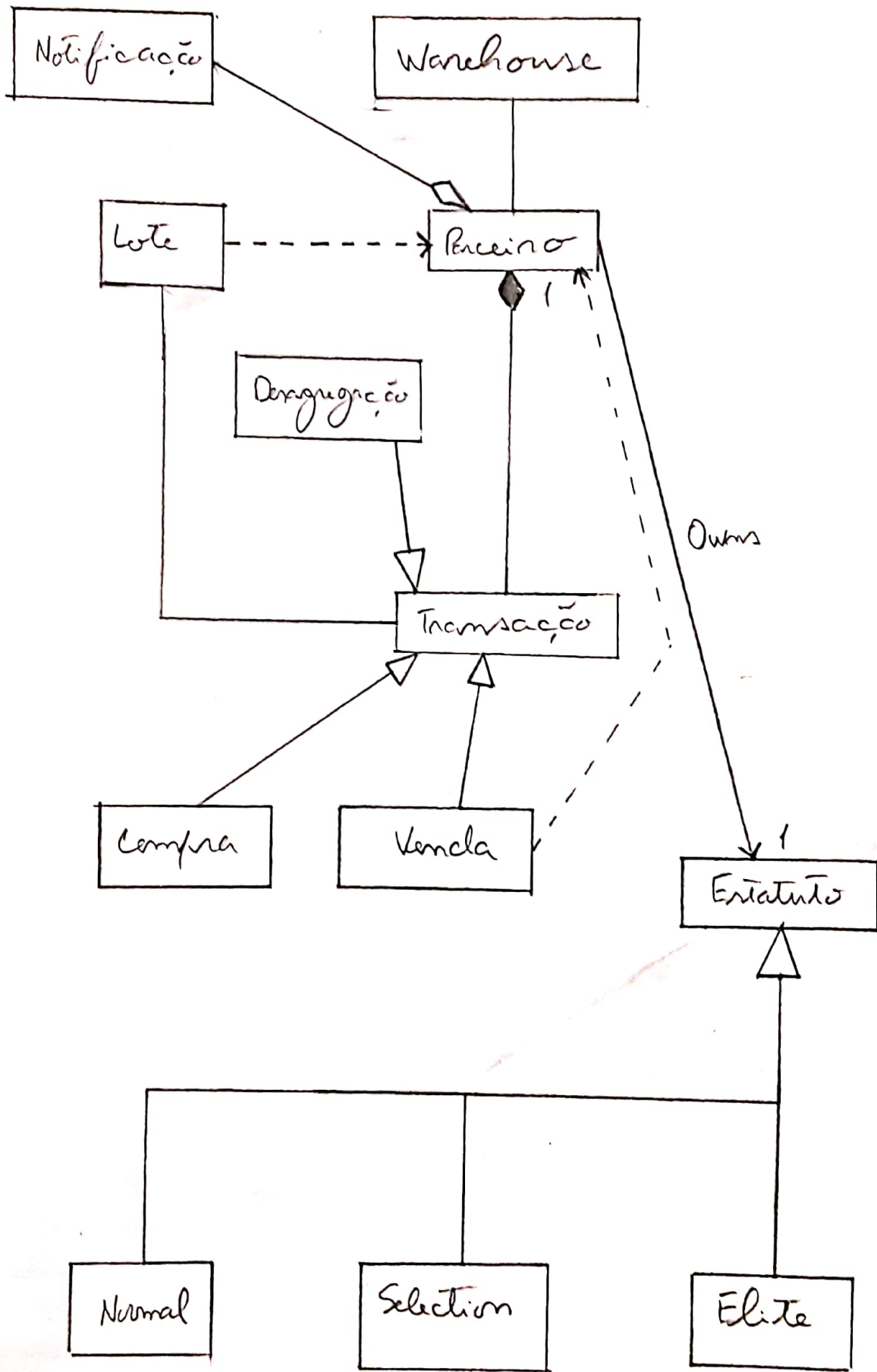
João Silva.

Declaro por minha honra que este diagrama foi  
realizado apenas pelos elementos que constituem  
o grupo do projeto

Luís Marques

\* Das classes aqui representadas, as únicas que não pertencem ao package `gec` são a interface `Serializable` e a classe `Exception`





gcl.exceptions

UnavailableFileException

ImportFileException

MissingFileAssociation

BadEntryException

Exception

```
graph LR; subgraph gcl_exceptions [gcl.exceptions]; UFE[UnavailableFileException]; IFE[ImportFileException]; MFA[MissingFileAssociation]; BE[BadEntryException]; end; UFE --> E[Exception]; IFE --> E; MFA --> E; BE --> E;
```

The diagram illustrates a class hierarchy for exceptions. A large container box labeled 'gcl.exceptions' at the top left contains four sub-boxes: 'UnavailableFileException', 'ImportFileException', 'MissingFileAssociation', and 'BadEntryException'. To the right of this container is a box labeled 'Exception'. Four arrows originate from the right side of each sub-box within the 'gcl.exceptions' container and point towards the 'Exception' box. Each arrow terminates in a small, hollow triangular arrowhead, indicating that each of the four sub-classes inherits from the 'Exception' base class.

Apenas referenciar as classes que possuem atributos ou métodos de referência relevante no UML

## Warehouse Manager

```
- filename: String = ""  
- warehouse: Warehouse  
- dataAtual: int = 0  
+ save(): void  
+ saveAs(filename: String): void  
+ load(filename: String): void  
+ importFile(textFile: String): void  
+ definirData(dataAtual: int): void  
+ adicionarData(dias: int): void  
+ todosPorceiros(): String  
+ todosProdutos(): String  
+ lotesPorceiro(): String  
+ lotesProduto(): String  
+ mostraPorceiro(): String  
+ visualizar(): String  
+ ligarNotificacao(): String  
+ desligarNotificacao(): String  
+ mostrarTransPorceiro(): String  
+ mostrarTransUendas(): String  
+ mostrarCompras(): String  
+ mostrarLoteBaixo(): String  
+ AprovarmentoMultiplicativo(): float
```

# Warehouse

- `serialVersionUID` : long
- `saldo` : int = 0
- `lotes` : Lote [0..\*]

- + `importFile (txtFile : String) : void`
- + `execDesagregacao (idPorceno : String) : Componente [1..*]`
- + `exCompna (compna : Compna) : void`
- + `execVenda (venda : Venda) : void`
- + `removeLote (-lotes : Lote) : String`
- + `sendNotification (notification : Notificacao) : void`
- + `todosPorceno (porceno : Lote) : String`
- + `todosProdutosDispo (idPorceno : String) : String`
- + `lotsPorceno (idPorceno : String) : String`
- + `lotsProdutos (idProdutos : String) : String`
- + `mostrePorceno (porceno : Porceno) : String`
- + `visualizar (porceno : Porceno) : String`
- + `displayNotificacao (notificacao : Notificacao) : String`
- + `deslizerNotificacao (notificacao : Notificacao) : String`
- + `mostreTodosPorceno (porceno : Porceno) : String`
- + `mostreTodosVendas ( ) : String`
- + `mostreCompna ( ) : String`
- + `mostreLoteAlcivar ( ) : String`
- + `AgrupamentoMultiplicativo (produtos : Produtos) : float`



Lote
<ul style="list-style-type: none"> <li>- fornecedor : Pizzino</li> <li>- m Produto : int</li> <li>- precoPorProduto : float</li> </ul>
<ul style="list-style-type: none"> <li>+ vender() : void</li> <li>+ comprar() : void</li> <li>+ fabricar() : void</li> <li>+ definirPreco() : void</li> </ul>

Componente
<ul style="list-style-type: none"> <li>- idComponente : String</li> <li>- m Componente : int</li> </ul>
+ verificarProduto() : boolean

Receita
- receita : Componente [1..*]

Produto
- id Produto : String

Simples



## Unavailable File Exception

- <<final>> serialVersionUID : long  
- filename : String

+ getFilename() : String

## Import File Extension

- <<final>> serialVersionUID : long

## Missing File Association Exception

- <<final>> serialVersionUID : long

## Bad Entry Exception

- <<final>> serialVersionUID : long  
- entrySpecification : String

+ getEntrySpecification() : String

Estatuto
- multa : float
- descensor : float

Normal
- <<final>> minimoPontos : int = 0

Selection
- <<final>> minimoPontos : int = 2000

Elite
- <<final>> minimoPontos : int = 25000

Venda
- parceiro : Parceiro
- mProdutos : int
- data Limite Pag : int
- price : float
+ preco Por Tempo (data Limite Pag : int, data Atual : int) : float
+ aplica Estatuto (parceiro : Parceiro) : float
+ pontos Venda (parceiro : Parceiro) : void
+ ref Atualizar() : boolean

## Parceiro

- id Parceiro : String
- marca : String
- estatuto : Estatuto
- regist Transac : Transacao [0..\*]
- notificacoes : Notificacao [0..\*]

- + Liga Notificacao (id Produto : String) : void
- + desLiga Notificacao (id Produto : String) : void
- + verifica Alteracao Nivel () : void

## Transacao

- idTrans : int = 0
  - data Pagamento : int
- + addTrans() : void

## Compra

- parceiro : Parceiro
- m Produto : int
- price : float

## Desagregacao

- parceiro : Parceiro
- produtor : Produtor

+ Taxa (produtor : Produtor, componentes : Componente [1..\*]) : float

## Notificacao

- idProdutor : String
- send Metodo : String

## New

- notification Descrip : String = "New"

+ get Stock (warehouse : Warehouse) : String

## Bargain

- notification Descrip : String = "Bargain"

+ cheap Stock (warehouse : Warehouse) : String