

WareHouse Manager

- filename: String
- warehouse: WareHouse

+ get Payments By Partner (id: String): Collection< Transaction >
+ get Products Batches Under Given Price (price: double): Collection< Batch >
+ advance Date (date: Date): void
+ display Date (): int
+ load (filename: String): void
+ save (): void
+ saveAs (filename: String): void
+ input File (textFile: String): void
+ get Global Balance (): double
+ register Partner (name: String, address: String, id: String): void
+ get Partners (): Collection< Partners >
+ get Partner (id: String): Partner
+ get Partner Acquisitions (id: String): Collection< Transaction >
+ get Partner Sales (id: String): Collection< Transaction >
+ toggle Product Notifications (id: String, idProduct: String)
+ get Products (): Collection< Product >
+ get Available Batches (): Collection< Batch >
+ get Batches By Partner (id: String): Collection< Batch >
+ get Batches By Product (id: String): Collection< Batch >
+ receive Payment (id: int): void
+ register Acquisition Transaction (id: String, idProduct: String, price: double, amount: int): void
+ register BreakDown Transaction (id: String, idProduct: String, amount: int): void
+ register Sale Transaction (id: String, limitDate: int, idProduct: String, amount: int): void
+ get Transaction (id: int): Transaction
+ create Batch (id: String, idProduct: String, amount: int, price: double): void
+ create Deemed Product (id: String, idProduct: String, numberCopy: int, aggregation: double, products: Collection< Product >): void
+ create Simple Product (id: String, idProduct: String, price: double, amount: int): void

Partner

- id: String
- name: String
- address: String
+ toggleNotification(id: String): void
+ getAcquisitions(): Collection< Transaction >
+ getSales(): Collection< Transaction >
+ getBreakDowns(): Collection< Transaction >
+ getFees(): Collection< Transaction >
+ getDiscounts(): Collection< Transaction >
+ getPoints(): int
+ chargePoints(amount: int): void
+ getSaleValue(): double
+ getAvgTransactionAmount(): double
+ getTotalAcquisitionValue(): double
+ getAvgSaleType(): PartnerType
+ addBreakdown(Transaction): void
+ addSale(Transaction): void
+ addAcquisition(Transaction): void

- id: String
- price: double
- stock: int

+ increaseStock(amount: int): void
+ decreaseStock(amount: int): void

Batch

- product: Product
- partner: Partner
- stock: int
- price: double
+ increaseStock(amount: int): void
+ decreaseStock(amount: int): void

- id: String
- price: double
- stock: int

+ increaseStock(amount: int): void
+ decreaseStock(amount: int): void

Simple Product

- id: String
- price: double
- stock: int

- id: String
- price: double
- stock: int

- id: String
- price: double
- stock: int

- id: String
- price: double
- stock: int

- id: String
- price: double
- stock: int

- id: String
- price: double
- stock: int

- id: String
- price: double
- stock: int

- id: String
- price: double
- stock: int

- id: String
- price: double
- stock: int

- id: String
- price: double
- stock: int

- id: String
- price: double
- stock: int

- id: String
- price: double
- stock: int

- id: String
- price: double
- stock: int

- id: String
- price: double
- stock: int

- id: String
- price: double
- stock: int

- id: String
- price: double
- stock: int

- id: String
- price: double
- stock: int

- id: String
- price: double
- stock: int

- id: String
- price: double
- stock: int

- id: String
- price: double
- stock: int

- id: String
- price: double
- stock: int

- id: String
- price: double
- stock: int

- id: String
- price: double
- stock: int

- id: String
- price: double
- stock: int

- id: String
- price: double
- stock: int

- id: String
- price: double
- stock: int

- id: String
- price: double
- stock: int

- id: String
- price: double
- stock: int

- id: String
- price: double
- stock: int

- id: String
- price: double
- stock: int

- id: String
- price: double
- stock: int

- id: String
- price: double
- stock: int

- id: String
- price: double
- stock: int

- id: String
- price: double
- stock: int

- id: String
- price: double
- stock: int

- id: String
- price: double
- stock: int

- id: String
- price: double
- stock: int

- id: String
- price: double
- stock: int

- id: String
- price: double
- stock: int

- id: String
- price: double
- stock: int

- id: String
- price: double
- stock: int

- id: String
- price: double
- stock: int

- id: String
- price: double
- stock: int

- id: String
- price: double
- stock: int

- id: String
- price: double
- stock: int

- id: String
- price: double
- stock: int

- id: String

<tbl_r cells="1" ix="2"

Partner Type

- points: int

+ get Discount(deadline: int): double
+ get Incentive(deadline: int): double
+ ChangePoints(amount: int): void

+ ~~get Points(): int~~
+ ~~get Incentive(): double~~

Normal

- <<Pnl>> NORMAL : string

Selection

- <<Pnl>> SELECTION : string

Elite

- <<Pnl>> ELITE : string

Transaction

- id: int
- amount: int
- product: Product
- partner: Partner

BreakDown

- vbase: double
- totalPrice: double

- - date: int
- origin: Project

- ProjBatch: Collection<Batch>

Sale

- - unitSale: int
- - Payment Date: int
- - totalPrice: double
- - finalPrice: double
- + payOff: void

Acquisition

- price: double

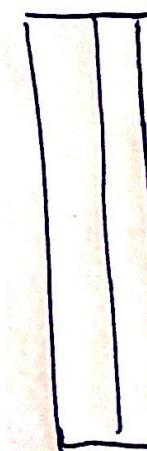
Refund

- <<Pnl>> REFUND : string

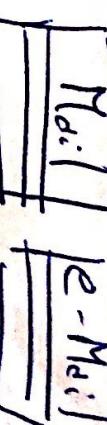
Notification

```
- - idProd: String  
- - description: String  
- - deliveryMethod: String  
+ send(): void
```

Delivery Method



Methods "equals", "toString" have been omitted since they are inherited from Object. The same for Constructors, getters, setters, have also been omitted.



Delivery Per Min, non que est diagno
po, realizab spose per elemto que
constitui o projeto.

concepto

Warehouse

```
- balance: double  
- globalBalance: double  
- date: int  
+ getPaymentsByPartner(id: String): Collection<Transaction>  
+ getProductPartnersUnderGivenPrice(price: double): Collection<Batch>  
+ addNewOrder(amount: int): void  
+ displayDate(): int  
+ importFile(textFile: String): void  
+ getGlobalBalance(): int  
+ registerPartner(name: String, address: String, id: String): void  
+ getPartners(): Collection<Partner>  
+ getPartnerAcquisitions(id: String): Collection<Transaction>  
+ getPartnerSales(id: String): Collection<Transaction>  
+ toggleProductNotifications(id: String, idProduct: String)  
+ getProducts(): Collection<Product>  
+ getAvailableBatches(): Collection<Batch>  
+ getBatchesByPartner(id: String): Collection<Batch>  
+ getBatchesByProduct(id: String): Collection<Batch>  
+ receivePayment(id: int): void  
+ reqForAcquisitionTransaction(id: String, idProduct: String, price: double, amount: int): void  
+ reqForBreakdownTransaction(id: String, idProduct: String, amount: int): void  
+ registerSaleTransaction(id: String, intDate: int, idProduct: String, amount: int): void  
+ getTransaction(id: int): Transaction  
+ createBatch(id: String, idProduct: String, amount: int, price: double)  
+ createDerivedProduct(id: String, idProduct: String, numberCopies: int, aggregation: double, products: Collection<Product>, collection<Integers>): void  
+ createSimpleProduct(id: String, idProduct: String, price: double, amount: int): void  
+ getPartner(id: String): Partner  
+ sendNotifications(idProduct: String, message: String): void
```