



**FEUP** FACULDADE DE ENGENHARIA  
UNIVERSIDADE DO PORTO

## *Farmácia*

Bases de Dados  
MIEIC

15/04/2018

Francisco Friande - [up201508213@fe.up.pt](mailto:up201508213@fe.up.pt)  
Francisco Ferreira- [up201605660@fe.up.pt](mailto:up201605660@fe.up.pt)  
João Pedro Fidalgo - [up201605237@fe.up.pt](mailto:up201605237@fe.up.pt)

# ***Descrição do Tema***

Decidimos implementar a base de dados de uma farmácia. O nosso ponto de partida foi o ato da venda de um medicamento (cliente, farmacêutico e medicamento) e posteriormente acrescentámos uma terceira entidade, o fornecedor e fomos adaptando o UML de forma a que não ficasse muito complexo.

Começando pela superclasse Pessoa, que dá origem a três subclasses ( Cliente, Farmacêutico e Fornecedor). Esta classe tem como atributos nome, número de telefone, localidade e idade.

Um único Cliente, que tem como atributo o número de cliente pode associar-se a várias vendas / transações, tendo por isso esta relação um multiplicidade de um para muitos. Uma Transação, que possui uma data e uma hora, está também relacionada com um Farmacêutico, onde uma transação possui apenas um farmacêutico, mas por outro lado, um farmacêutico pode estar associado a várias transações.

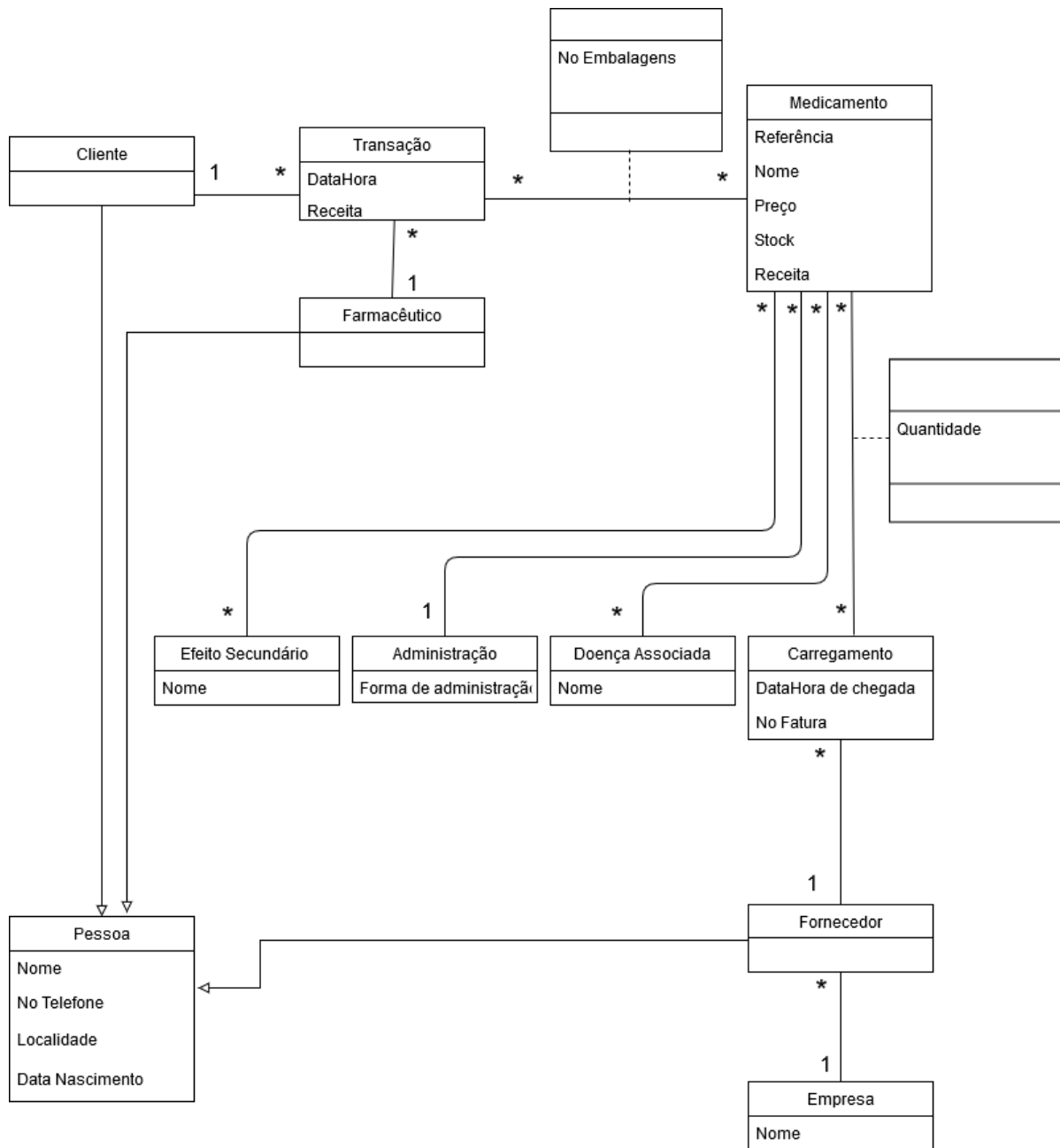
No centro de toda esta base de dados, está a classe Medicamento, cujos atributos são: referência, nome, preço e quantidade em stock. Medicamento e transação estão ligados por uma relação de muitos para muitos, pelo facto de que um mesmo medicamento pode ter várias transações mas o mesmo acontece vice-versa. Existe ainda uma característica que estas duas classes possuem em comum que é o número de embalagens de um certo medicamento que são vendidas numa certa transação.

O medicamento relaciona-se depois com as classes: efeito secundário, administração (forma como se administra o medicamento), doença associada e ainda carregamento.

Esta última classe (carregamento), juntamente com a classe Medicamento, que fazem parte de uma relação de muitos para muitos, têm o atributo quantidade em comum.

Finalmente, um carregamento está associado a um Fornecedor, sendo que um fornecedor pode ter vários carregamentos e um carregamento tem apenas um fornecedor.

# Diagrama UML



# ***Esquema Relacional e Dependências Funcionais***

Pessoa(IdPessoa, Nome, NoTelefone, Localidade, Data Nascimento)

1. { IdPessoa } -> { Nome, NoTelefone, Localidade, Data Nascimento }

Fornecedor(IdFornecedor->Pessoa, IdEmpresa->Empresa)

1. { IdFornecedor } -> { IdEmpresa }

Cliente(IdCliente->Pessoa)

Farmacêutico(IdFarmacêutico->Pessoa)

Transação(IdTransação, DataHora, Receita, IdCliente->Cliente, IdFarmacêutico->Farmacêutico)

1. { IdTransacao } -> { DataHora, Receita, IdCliente, IdFarmaceutico }

Venda(IdTransação->Transação, IdMedicamento->Medicamento, NoEmbalagens)

1. { IdTransação } -> { IdMedicamento, NoEmbalagens }

Medicamento(IdMedicamento, Referência, Nome, Preço, Stock, Receita, IdAdministração->Administração)

1. { IdMedicamento } -> { Referência, Nome, Preço, Stock, Receita, IdAdministração }
2. { Referência } -> { Nome, Preço, Stock, Receita, IdAdministração, IdMedicamento }

DoençaAssociada(IdDoença, Nome)

1. { IdDoenca } -> { Nome }

Administração(IdAdministração, FormaAdministração)

1. { IdAministracao } -> { FormaAdministracao }

EfeitoSecundário(IdEfeitoSecundário, Nome)

1. { IdEfeitoSecundario } -> { Nome }

Carga(IdMedicamento->Medicamento, IdCarregamento->Carregamento, Quantidade)

1. { IdMedicamento, IdCarregamento } -> { Quantidade }

Carregamento(IdCarregamento, DataHora, NoFatura, IdFornecedor->Fornecedor)

1. { IdCarregamento } -> { DataHora, NoFatura, IdFornecedor }

Empresa(IdEmpresa, Nome)

1. { IdEmpresa } -> { Nome }

## ***Formas Normais***

Para verificar eventuais violações da 3ª Forma Normal (e ao verificarmos isto verificamos também as violações da Forma Normal de Boyce-Codd, visto se tratar de uma forma ligeiramente mais restrita da anterior) basta-nos verificar se está na primeira e segunda Formas Normais e analisar o cumprimento ou não cumprimento da regra da não-transitividade.

Deste modo, o nosso modelo proposto viola estas normas na classe Medicamento. Isto acontece porque através do ID do medicamento podemos chegar à sua referência e da sua referência existe um relação para todos os restantes atributos da classe. Assim sendo, existe uma conexão indireta entre o ID e o Nome, Preço, Stock, Receita e IdAdministração, sustentando-se uma relação de transitividade, havendo por isso uma violação da 3ª Forma Normal e consequentemente, da Forma Normal de Boyce-Codd.

Relativamente às restantes relações não existem violações pois para cada dependência, o seu lado esquerdo é uma super-key do esquema relacional, sendo por isso uma condição suficiente para o cumprimento destas formas normais .

# Restrições

A base de dados implementa várias restrições, de forma a assegurar uma boa manutenção desta, acrescentando ainda alguma segurança ao utilizador. Assim, existem restrições do tipo NOT NULL, UNIQUE, CHECK, do tipo chave e do tipo integridade referencial.

## NOT NULL :

Este tipo de restrição exclui valores nulos, que não fariam qualquer sentido no contexto de uma farmácia.

1. Administração - Forma de Administração
2. Carregamento - Data e Hora, No Fatura
3. Doença Associada - Nome
4. Efeito Secundário - Nome
5. Empresa - Nome
6. Medicamento - Referência, Nome, Preço, Receita
7. Transação - Data e Hora, Receita

## UNIQUE:

Os atributos associados a esta restrição têm de ter valores únicos, como o nome o indica. Um exemplo é que não podem existir duas pessoas diferentes na base de dados com o mesmo contacto telefónico, pois não faria sentido.

1. Carregamento - No Fatura
2. Medicamento - Referência
3. Pessoa - No Telefone

## CHECK:

Os atributos que abrangem esta restrição têm de ter os seus valores contidos nos intervalos por ela definidos. Um dos casos em baixo ilustrados é por exemplo a idade de uma pessoa, pois quer seja cliente, farmacêutica ou fornecedora, é “obrigada” a ter mais do que 18 anos para poder fazer parte da base de dados, algo que está associado ao contexto real de uma farmácia.

1. Medicamento - Preço (Preço > 0 AND Preço <= 1000)
2. Medicamento - Stock ( Stock < 1000)
3. Pessoa - Idade ( Idade > 18)

A nível de restrição de Primary Key, todas as suas classes possuem a sua, sendo de destacar as classes Venda e Carga pois apresentam chaves primárias compostas.

Restrições de entidade referencial aplicámos em classes intrinsecamente relacionadas com outras, usando chaves estrangeiras.

## Interrogações

1. Selecionar os medicamentos esgotados, para serem encomendados mais.
2. Selecionar o nome dos clientes que foram atendidos no ano de 2017.
3. Selecionar o número de Transações em que tanto o nome do cliente como o do farmacêutico era "Francisco".
4. Selecionar a média de preço de medicamentos sem receita cujo ID do fornecedor seja 9.
5. Selecionar o nome do cliente, o nome do medicamento que comprou e em que quantidade o fez, ordenado por ordem alfabética do seu nome.
6. Selecionar o nome e nr de telefone dos fornecedores que trouxeram medicamentos cuja administração é "via oral".
7. Selecionar a referência dos medicamentos que tenham sido trazidos pela empresa 'Infarmed' e cujo stock seja maior que 50.
8. Encontrar o medicamento com mais procura/menor número de vendas.
9. Selecionar o id das transações no mês de fevereiro de 2017 cujo número de embalagens vendidas tenha sido superior a 3 ou das transações no mês de abril de 2018 com menos que 2
10. Selecionar o nome e data de nascimento do Farmacêutico que vendeu mais embalagens de uma só vez em 2018.

## Gatilhos

1. Verifica se a data de nascimento é válida (maior que 18 anos).
- 2.1. Quando chega um carregamento de medicamentos o seu stock aumenta.
- 2.2. Quando vendemos medicamentos o seu stock diminui.
3. Erro caso o cliente não tenha receita para uma transação que assim o exija.

