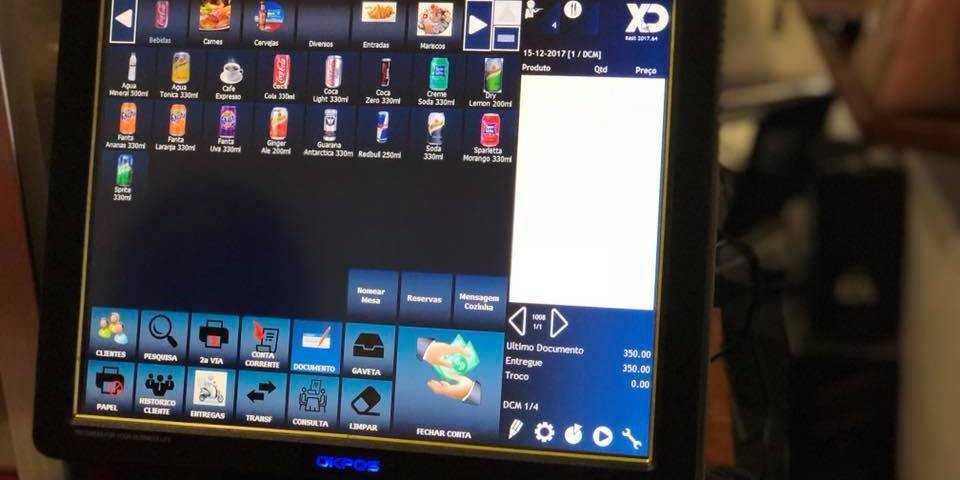


Bases de Dados

MIEIC

**Base de dados para gestão de um restaurante**



Turma 2

Grupo 203

Alunos:

Amadeu Prazeres Pereira – up201605646 Nuno Tiago Tavares Lopes – up201605337 Tomás Nuno Fernandes Novo – up201604503

# Descrição sucinta do contexto

***José Devezas***

*2018-03-13 09:59:00*

--------------------------------------------

Agora evento, certo?

***José Devezas***

*2018-03-13 09:59:54*

--------------------------------------------

Opinião é uma review textual ou um rating (p.e., em cinco estrelas)?

Neste projeto optámos pela elaboração de uma base de dados que facilite a gestão de um restaurante, objetivando uma confortável administração dos recursos humanos e espaços disponíveis do respetivo local de restauração.

As informações manipuladas incidem principalmente sobre os principais componentes necessários ao funcionamento eficaz do restaurante, nomeadamente dados relativamente aos funcionários e à quantidade de mesas disponíveis em cada sala e seu correspondente número de lugares. Também são geridas informações acerca do género de entretenimento de cada sala, das ementas existentes e, como não podia faltar, da organização dos clientes no restaurante.

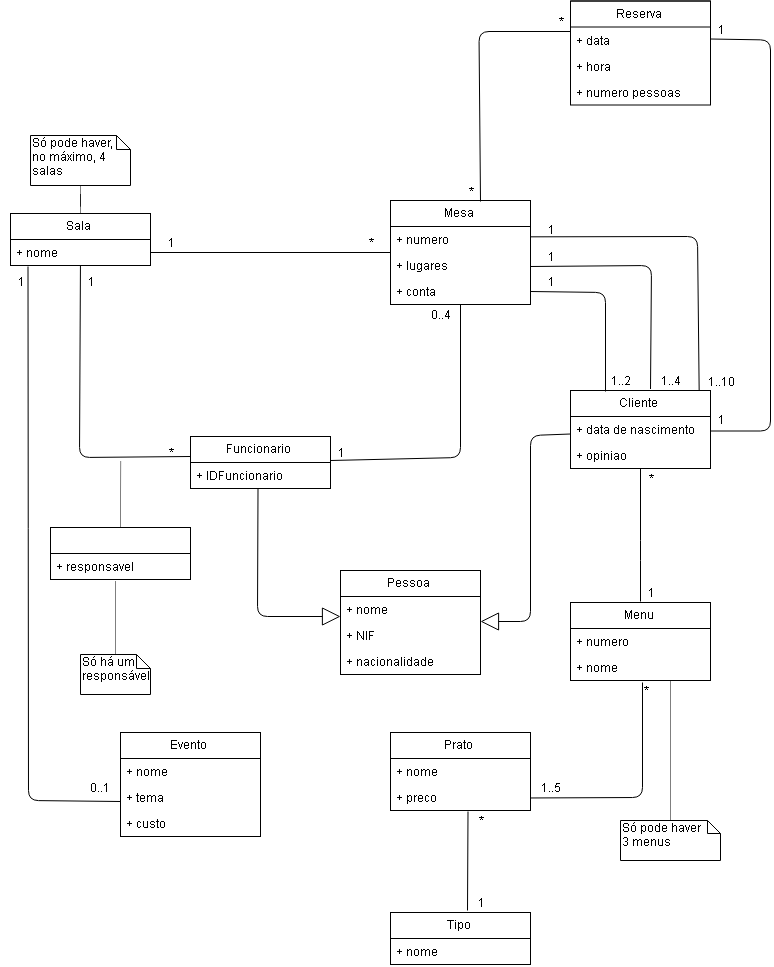
Na base de dados temos as classes **Cliente** e **Funcionario**, ambos herdam atributos da classe **Pessoa** (nome, NIF, nacionalidade). Os clientes têm como atributos a data de nascimento e a sua opinião em relação ao serviço desempenhado pelo restaurante. Estes podem se sentar numa **Mesa** de 2, 4 ou 10 pessoas e podem efetuar uma **Reserva** destas mesmas, indicando a hora, a data e o número de pessoas. A classe **Mesa** possui um número que a identifica, o número de lugares e a despesa total dos clientes que pertencem à mesa.

Os funcionários por sua vez são identificados por um ID único. Apenas um funcionário é responsável por cada **Sala** e cada um está encarregado, no máximo, de 4 mesas. No restaurante só podem existir até 4 salas.

Em cada sala pode existir um **Evento**, que é identificado pelo seu nome, pelo tipo (Concerto, Magia, etc.) e pelo custo da organização do evento.

À disposição dos clientes, o restaurante possui 3 **Menus**, sendo que cada um pode possuir entre 1 a 5 **Pratos**, sendo estes distinguidos entre si através do nome, **Tipo** e preço.

Desta forma, a solução proposta visa organizar eficientemente a realização das atividades de restauração.



***José Devezas***

*2018-03-13 10:02:23*

--------------------------------------------

Usem (?) para indicar booleanos. Ajuda aler o modelo conceptual.

***José Devezas***

*2018-03-13 10:07:02*

--------------------------------------------

Conta não é um atributo de mesa. Surge da interação entre cliente e mesa. O ideal será criar uma classe de associação por cada uma das três associações. Ou seja, quando atribuímos um cliente a uma mesa, este tem uma conta individual que pode ser somada para ter a conta geral da mesa.

***José Devezas***

*2018-03-13 10:09:43*

--------------------------------------------

Parece-me que não fará muito sentido este atributo no modelo conceptual. No limite podem deixar a classe vazia, mas não haverá outros atributos relevantes para um funcionário? Ordenado, período do contrato, etc.?

***José Devezas***

*2018-03-13 10:13:12*

--------------------------------------------

Pode ter zero funcionários? Seria melhor 1..\*

Mas...

Só havendo um responsável e visto quejá tem a associação de funcionários às salas via mesa, basta que utilizem até uma associação um-para-um. Nem precisam de 1..\*. Assim a restrição fica implícita.

***José Devezas***

*2018-03-13 10:14:47*

--------------------------------------------

Não precisam de separar data e hora. Existem sempre um tipo data/hora nas BDs. Em SQLite guarda-se um INTEGER com o timestamp.

***José Devezas***

*2018-03-13 10:16:08*

--------------------------------------------

Alteraria a multiplicidade nestas associações para 0..1 mesas. Isto porque um cliente pode ter feito uma reserva mas ainda não estar no restaurante.

***José Devezas***

*2018-03-13 10:17:07*

--------------------------------------------

Chamem-lhe TipoPrato. Se mais tarde precisarem de mais tipos para outras coisas, fica inconsistente.

# Diagrama UML

***José Devezas***

*2018-03-13 10:14:16*

--------------------------------------------

Colocaria uma restrição nesta associação que garantisse que a mesma mesa não é reservada para a mesma data/hora.