

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

Índice

<i>Descrição sucinta do contexto.....</i>	<i>3</i>
<i>Diagrama UML</i>	<i>4</i>
<i>Esquema Relacional.....</i>	<i>5</i>
<i>Análise dependências funcionais e formas normais</i>	<i>6</i>
<i>Restrições e respetiva implementação.....</i>	<i>12</i>
<i>Criação e Povoamento da Base de Dados.....</i>	<i>15</i>

Descrição sucinta do contexto

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 evento 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 um review (0 a 5 estrelas) 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.

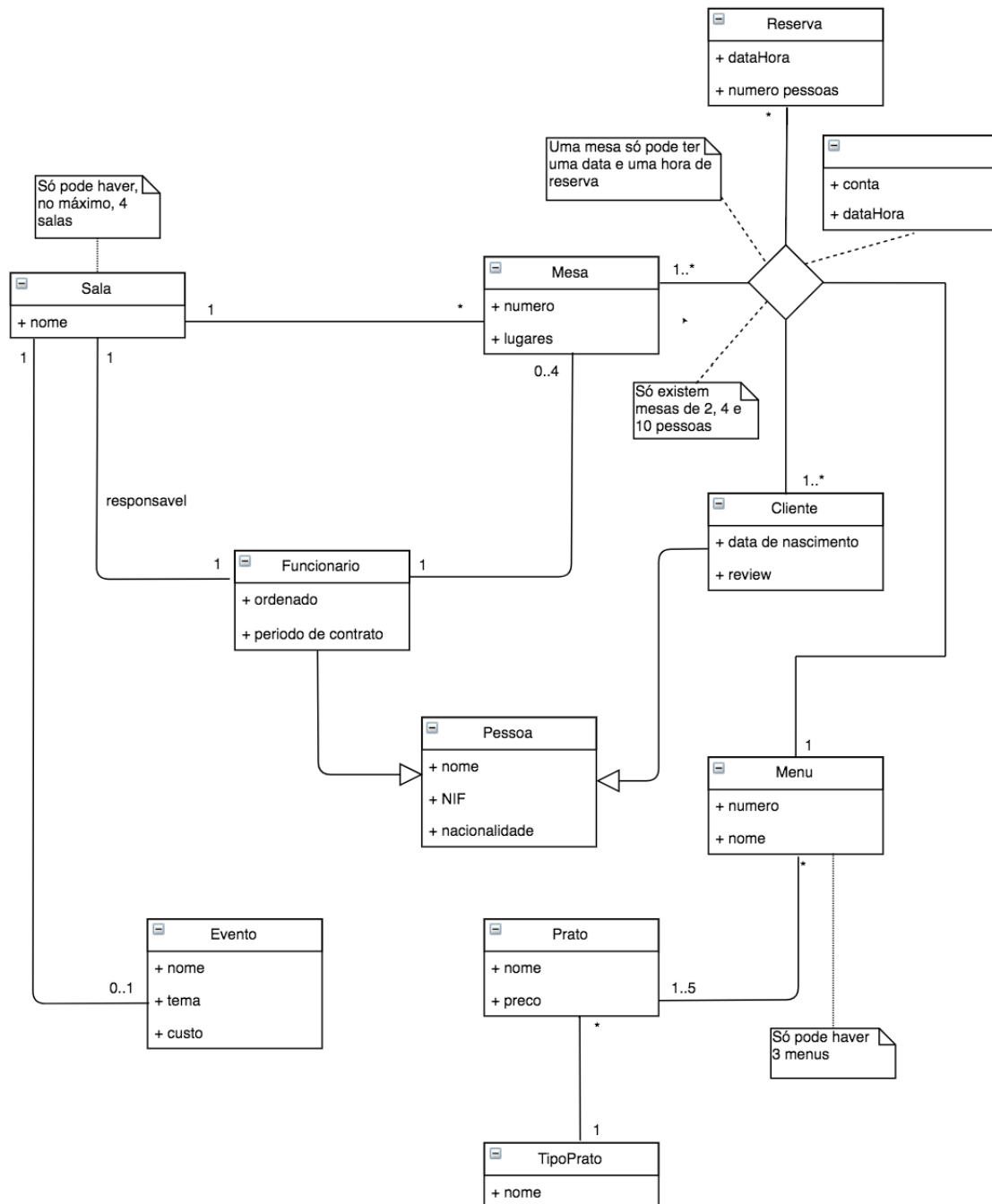
Os funcionários por sua vez são identificados por um ID único. Apenas um funcionário é responsável por cada **Sala** e cada um está encarregado, no máximo, de 4 mesas. No restaurante só podem existir até 4 salas.

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.

Diagrama UML



Esquema Relacional

O texto abaixo representa o nosso modelo relacional, sendo que os atributos a sublinhado são as chaves primárias de cada relação.

Sala (idSala, nome, responsavel->Funcionario)

Mesa (numero, lugares, idFuncionario->Funcionario, idSala->Sala)

Reserva (idReserva, dataHora, numeroPessoas)

Pessoa (idPessoa, nome, NIF, nacionalidade)

Cliente (idCliente->Pessoa, data de nascimento, review)

Funcionario (idFuncionario->Pessoa, ordenado, periodoContrato)

Menu (numero, nome)

Evento (idEvento, nome, tema, custo, idSala->Sala)

Prato (idPrato, nome, preco, idTipoPrato->TipoPrato)

TipoPrato (idTipoPrato, nome)

ClienteMesaReservaMenu (idReserva->Reserva, idMesa->Mesa, idCliente->Cliente, idMenu->Menu, conta, dataHora)

MenuPrato (idMenu->Menu, idPrato->Prato)

Análise dependências funcionais e formas normais

Nesta secção será feita a análise das dependências funcionais do nosso modelo, assim como a verificação da não violação à Forma Normal Boyce-Codd e à 3ª Forma Normal.

Sala

idSala -> nome, responsável

- Está na 1ª Forma Normal, pois todos os atributos têm exclusivamente valores atômicos.
- Está na 2ª Forma Normal, pois todos os atributos têm dependência funcional da chave primária.
- Está na 3ª Forma Normal, pois não existe dependência transitiva.
- Está na Forma Normal de Boyce-Codd, pois não existe redundância em dependências funcionais.

Mesa

numero -> lugares, idFuncionario, idSala

- Está na 1ª Forma Normal, pois todos os atributos têm exclusivamente valores atômicos.
- Está na 2ª Forma Normal, pois todos os atributos têm dependência funcional da chave primária.
- Está na 3ª Forma Normal, pois não existe dependência transitiva.
- Está na Forma Normal de Boyce-Codd, pois não existe redundância em dependências funcionais.

Reserva

idReserva -> dataHora, numeroPessoas

- Está na 1ª Forma Normal, pois todos os atributos têm exclusivamente valores atômicos.
- Está na 2ª Forma Normal, pois todos os atributos têm dependência funcional da chave primária.
- Está na 3ª Forma Normal, pois não existe dependência transitiva.
- Está na Forma Normal de Boyce-Codd, pois não existe redundância em dependências funcionais.

Cliente

idCliente -> data de nascimento, review

- Está na 1ª Forma Normal, pois todos os atributos têm exclusivamente valores atômicos.
- Está na 2ª Forma Normal, pois todos os atributos têm dependência funcional da chave primária.
- Está na 3ª Forma Normal, pois não existe dependência transitiva.
- Está na Forma Normal de Boyce-Codd, pois não existe redundância em dependências funcionais.

Funcionario

idFuncionario -> ordenado, periodoContrato

- Está na 1ª Forma Normal, pois todos os atributos têm exclusivamente valores atômicos.
- Está na 2ª Forma Normal, pois todos os atributos têm dependência funcional da chave primária.
- Está na 3ª Forma Normal, pois não existe dependência transitiva.
- Está na Forma Normal de Boyce-Codd, pois não existe redundância em dependências funcionais.

Pessoa

idPessoa -> nome, NIF, nacionalidade

NIF -> idPessoa, nome, nacionalidade

- Está na 1ª Forma Normal, pois todos os atributos têm exclusivamente valores atômicos.
- Está na 2ª Forma Normal, pois todos os atributos têm dependência funcional da chave primária.
- Está na 3ª Forma Normal, pois não existe dependência transitiva.
- Está na Forma Normal de Boyce-Codd, pois não existe redundância em dependências funcionais.

Evento

idEvento -> nome, tema, custo, idSala

- Está na 1ª Forma Normal, pois todos os atributos têm exclusivamente valores atômicos.
- Está na 2ª Forma Normal, pois todos os atributos têm dependência funcional da chave primária.
- Está na 3ª Forma Normal, pois não existe dependência transitiva.
- Está na Forma Normal de Boyce-Codd, pois não existe redundância em dependências funcionais.

Menu

numero -> nome

nome -> numero

- Está na 1ª Forma Normal, pois todos os atributos têm exclusivamente valores atômicos.
- Está na 2ª Forma Normal, pois todos os atributos têm dependência funcional da chave primária.
- Está na 3ª Forma Normal, pois não existe dependência transitiva.
- Está na Forma Normal de Boyce-Codd, pois não existe redundância em dependências funcionais.

Prato

idPrato -> nome, preco, idTipoPrato

nome -> idPrato, preco, idTipoPrato

- Está na 1ª Forma Normal, pois todos os atributos têm exclusivamente valores atômicos.
- Está na 2ª Forma Normal, pois todos os atributos têm dependência funcional da chave primária.
- Está na 3ª Forma Normal, pois não existe dependência transitiva.
- Está na Forma Normal de Boyce-Codd, pois não existe redundância em dependências funcionais.

TipoPrato

idTipoPrato -> nome

nome -> idTipoPrato

- Está na 1ª Forma Normal, pois todos os atributos têm exclusivamente valores atômicos.
- Está na 2ª Forma Normal, pois todos os atributos têm dependência funcional da chave primária.
- Está na 3ª Forma Normal, pois não existe dependência transitiva.
- Está na Forma Normal de Boyce-Codd, pois não existe redundância em dependências funcionais.

ClienteMesaReservaMenu

idReserva, idMesa, idCliente, idMenu -> conta, dataHora

- Está na 1ª Forma Normal, pois todos os atributos têm exclusivamente valores atômicos.
- Está na 2ª Forma Normal, pois todos os atributos têm dependência funcional da chave primária.
- Está na 3ª Forma Normal, pois não existe dependência transitiva.
- Está na Forma Normal de Boyce-Codd, pois não existe redundância em dependências funcionais.

MenuPrato

Não possui dependências funcionais.

- Está na 1ª Forma Normal, pois todos os atributos têm exclusivamente valores atômicos.
- Está na 2ª Forma Normal, pois todos os atributos têm dependência funcional da chave primária.
- Está na 3ª Forma Normal, pois não existe dependência transitiva.
- Está na Forma Normal de Boyce-Codd, pois não existe redundância em dependências funcionais.

Restrições e respetiva implementação

Os seguintes atributos (identificados com sublinhado) possuem a restrição PRIMARY KEY, na sua relação (identificada a **negrito**), dado que identificam um tuplo na tabela respetiva de forma única:

Reserva: idReserva (não pode haver duas reservas com o mesmo id)

Mesa: numero (não pode haver duas mesas com o mesmo número)

Cliente: idCliente (não pode haver dois clientes com o mesmo id)

Funcionario: idFuncionario (não pode haver dois funcionários com o mesmo id)

Sala: idSala (não pode haver duas salas com o mesmo id)

Pessoa: idPessoa (não pode haver duas pessoas com o mesmo id)

Evento: idEvento (não pode haver dois eventos com o mesmo id)

Prato: idPrato (não pode haver dois pratos com o mesmo id)

Menu: numero (não pode haver dois menus com o mesmo número)

TipoPrato: idTipoPrato (não pode haver dois tipos de prato com o mesmo id)

Os seguintes atributos (identificados com sublinhado) possuem a restrição FOREIGN KEY, na sua relação (identificada a **negrito**), dado que representam entidades pertencentes a outras tabelas da base de dados, tendo de existir na sua relação:

Sala: responsavel (chave estrangeira para Funcionario)

Mesa: idFuncionario (chave estrangeira para Funcionario), idSala (chave estrangeira para Sala)

Cliente: idCliente (chave estrangeira para Pessoa -> generalização)

Funcionario: idFuncionario (chave estrangeira para Pessoa -> generalização)

Evento: idSala (chave estrangeira para Sala)

Prato: idTipoPrato (chave estrangeira para TipoPrato)

ClienteMesaReservaMenu: idReserva (chave estrangeira para Reserva), idMesa (chave estrangeira para Mesa), idCliente (chave estrangeira para Cliente), idMenu (chave estrangeira para Menu)

MenuPrato: idMenu (chave estrangeira para Menu), idPrato (chave estrangeira para Prato)

Os seguintes atributos (identificados com sublinhado) possuem a restrição UNIQUE, na sua relação (identificada a **negrito**), dado que são atributos únicos em cada relação:

Menu: nome (não existem dois menus com o mesmo nome)

Pessoa: NIF (não existem duas pessoas com o mesmo NIF)

Prato: nome (não existem dois pratos com o mesmo nome)

TipoPrato: nome (não existem dois tipos de prato com o mesmo nome)

Os únicos atributos (identificados com sublinhado) que não possuem a restrição NOT NULL, na sua relação (identificada a **negrito**), são os seguintes (todos os outros possuem):

ClienteMesaReservaMenu: conta, dataHora (de momento ambos estes atributos podem ser NULL pois mais tarde vão ter de ser calculados a partir de outros atributos)

Cliente: dataNascimento (o cliente pode não querer dar a sua data de nascimento)

As seguintes restrições são implementadas na respetiva **relação** através de um CHECK para assegurar que qualquer combinação dos atributos envolvidos nas instancias tornem verdade as seguintes condições:

Cliente: A review tem de ser um valor compreendido entre 0 e 5

Funcionario: O ordenado e o período de contrato têm de ser superiores a 0

Mesa: As mesas só podem ter 2, 4 ou 10 lugares

Reserva: Uma reserva tem de ter um número de pessoas superior a 0

As seguintes restrições precisam de um gatilho para ser implementada, por essa razão só na 3ª entrega é que serão implementadas:

Sala: Só podem existir no máximo 4 salas

ClienteMesaReservaMenu: Uma mesa não pode ter várias reservas para um mesmo horário

Menu: Só pode haver no máximo 3 menus

MenuPrato: Cada menu tem de ter pelo menos 1 prato e só pode ter no máximo 5 pratos

Funcionario: Cada funcionário só pode ter no máximo 4 mesas

Criação e Povoamento da Base de Dados

As instruções SQL para a criação (com as restrições possíveis de implementar) e povoamento da base de dados encontram-se nos ficheiros `criar.sql` e `povoar.sql`, respetivamente.