

# PROJETO DE BASE DE DADOS

**Gerência de Hotel**



**Grupo 2MIEIC5\_3 :**

Angela Cruz (up201806781)

Marina Dias (up201806787)

Raquel Sepúlveda (up201806664)

## INTRODUÇÃO

Para o nosso projeto da unidade curricular de “Base de Dados” tivemos a oportunidade realizar uma de base de dados sobre um tema à nossa escolha. Decidimos então realizar uma base de dados sobre a gerência de um hotel em que este terá serviço de quartos, de bar e de spa.

Na nossa base de dados será guardada a informação de cada cliente, a época do ano da sua reserva e o tipo de quarto que reservou. Será ainda guardar a informação de todos os funcionários do hotel tal como os serviços realizados pelos mesmos.

## CONTEXTUALIZAÇÃO

O Hotel pretende guardar a informação dos seus clientes. Para cada **cliente** pretende-se armazenar o seu nome, o seu número de identificação fiscal, o seu email, a respetiva data de nascimento e as suas reservas. Um cliente pode reservar vários quartos.

Os **quartos** são diferenciados pelo seu número e caracterizam-se pelo tipo. Os tipos de quarto existentes no hotel são: quarto solteiro, quarto duplo solteiro, quarto casal e quarto familiar.

Cada **reserva** tem um número, data de início, data de fim e o preço total a pagar pela estadia. O preço varia consoante a **época do ano**, o tipo de quarto e a **taxa de ocupação** do hotel. Cada reserva tem que ter obrigatoriamente um cliente associado. Se um cliente pedir que o seu registo seja eliminado, as reservas associadas a este cliente deixam de existir.

O hotel possui vários **funcionários** com cargos diferentes que operam tanto no spa como no bar e também realizam serviços de quarto.

O hotel presta três tipos de **serviços** diferentes que só podem ser usufruídos por clientes que tenham uma reserva: o **serviço de quartos**, **tratamentos de spa** e serviço de **bebidas**. Os preços de cada um destes serviços variam consoante o que o cliente pedir.

O Bar do hotel oferece um serviço de acumulação de pontos por bebida. Consoante o preço de cada bebida acumula-se pontos diferentes num **cartão de pontos** único por estadia de cada cliente. Esses pontos serão somados e multiplicados por dez. A totalidade de pontos do cartão permite obter um desconto no valor total da fatura de serviços requisitados pelo cliente que este pagará no check-out do hotel. Cada ponto no cartão equivale a 1 cêntimo de desconto na fatura.

Na **fatura** é mencionado a data de emissão, os serviços requisitados e o valor total a pagar, que é calculado através da soma de todos os serviços requisitados menos o valor descontado através dos cartões de pontos. Cada fatura está ligada a uma reserva.

## CLASSES E RESPECTIVOS ATRIBUTOS

### Pessoa

- nif
- nome
- email
- nr\_tel

### Cliente

- data\_nascimento

### Serviço

- nome
- preco
- data

### Tipo de Bebida

### Tratamento de Spa

### Funcionário

- cargo

### Reserva

- nr\_reserva
- data\_inicio
- data\_fim

### Época do ano

- data\_inicio
- data\_fim

### Serviço de Quartos

### Cartão de Pontos

- nr\_pontos

### Fatura

- data
- total

### Taxa de Ocupação

- percentagem

### Quarto

- nr\_quarto
- tipo\_quarto

## ESQUEMA RELACIONAL E ANÁLISE DE DEPENDÊNCIAS FUNCIONAIS E FORMAS NORMAIS

**Pessoa**(nif, nome, email, nr\_tel)

- {nif -> nome, email, nr\_tel}

**Cliente**(nif->Pessoa, data\_nascimento)

- {nif -> data\_nascimento}

**Funcionario**(nif->Pessoa, cargo)

- {nif -> cargo}

**Reserva**(nr\_reserva, nr\_quarto->Quarto, nif->Pessoa, data\_inicio, data\_fim)

- {nr\_reserva -> nr\_quarto, nif, data\_inicio, data\_fim}

**Quarto**(nr\_quarto, tipo\_quarto)

- {nr\_quarto -> tipo\_quarto}

**EpocaDoAno**(data\_inicio, data\_fim)

**TaxaDeOcupacao**(percentagem)

**Servico**(id\_servico, nome, preco, data)

- {id\_servico -> nome, preco, data}

**ServicoDeQuartos**(id\_servico->Servico, nome, preco, data)

- {id\_servico -> nome, preco, data}

**TratamentoDeSpa**(id\_servico->Servico, nome, preco, data)

- {id\_servico -> nome, preco, data}

**TipodeBebida**(id\_servico->Servico, nome, preco, data)

- {id\_servico -> nome, preco, data}

**CartaoDePontos**(id\_cartao, nif->Pessoa, nr\_pontos)

- {id\_cartao-> nif, nr\_pontos}

**Fatura**(id\_fatura, id\_cartao->Cartao, data, total)

- {id\_fatura->id\_cartao, data, total}

**PrecoDeQuarto**(nr\_quarto->Quarto, percentagem->TaxaDeOcupacao, data\_inicio->EpocaDoAno, data\_fim->EpocaDoAno, preco)

- {nr\_quarto, percentagem, data\_inicio, data\_fim-> preco}

**QuantidadeDeServiçosPrestados**(id\_fatura->Fatura, id\_servico->Servico, quantidade);

- {id\_fatura, id\_servico->quantidade}

**QuantidadeDeBebidas**(id\_cartao->Cartao, id\_servico->TipoDeBebida, quantidade)

- {id\_cartao, id\_servico->quantidade}

**DescontoAplicado**(id\_cartao->Cartao, id\_fatura->Fatura, desconto)

- {id\_cartao, id\_fatura->desconto}

Em todas as dependências funcionais da nossa base de dados, o lado esquerdo é sempre uma chave primária, logo nenhuma dependência viola a Forma Normal de Boyce-Codd e consequentemente também não violam a 3ª Forma Normal.

## RESTRIÇÕES

Neste projeto usamos restrições para assegurar a manutenção da base de dados.

Usou-se então restrições do tipo NOT NULL para referir a propriedades que são obrigatórias para a existência da classe, UNIQUE para distinguir atributos que não podem ser repetidos em certas classes daí serem denominada únicas e a restrição CHECK para restringir certos aspetos dos atributos. Usaram-se igualmente as restrições ON DELETE CASCADE e ON UPDATE CASCADE, para que atributos que fazem referência a outra classe sejam devidamente atualizados caso estes sejam removidos ou modificados na classe a que fazem referência.

### Pessoa

- O atributo nif é chave primária e tem que obrigatoriamente conter nove dígitos, daí a restrição CHECK(nif>=100000000 AND nif<=999999999).
- O atributo nome não pode ser nulo, logo é usada a restrição NOT NULL
- Os atributos email e nr\_tel são atributos que não podem ser repetido, pois é impossível que duas pessoas possuam o mesmo número de telemóvel ou o mesmo e-mail, daí usarmos a restrição UNIQUE.

### Cliente

- O atributo nif é chave primária com referência para Pessoa, quando apagado ou modificado na classe Pessoa é igualmente apagado e modificado na classe Cliente, daí serem usadas as restrições ON DELETE CASCADE e ON UPDATE CASCADE.
- O atributo data\_nascimento é um atributo necessário para a gerência do hotel para fins de serviços de maiores de idade como por exemplo o Bar do hotel, no caso de bebidas alcoólicas. A restrição usada é NOT NULL.

### Funcionario

- O atributo nif é chave primária com referência para Pessoa, quando apagado ou modificado na classe Pessoa é igualmente apagado e modificado na classe Funcionario, daí serem usadas as restrições ON DELETE CASCADE e ON UPDATE CASCADE.
- O cargo não pode ser nulo. A um funcionário tem de ser atribuído pelo menos um cargo, daí a restrição NOT NULL.

## Reserva

- O atributo nr\_reserva é chave primária.
- Os atributos nr\_quarto e nif fazem referência a Quarto e a Pessoa respetivamente, quando apagados ou modificado na classe Quarto e Pessoa são igualmente apagados e modificados na classe Reserva, daí serem usadas as restrições ON DELETE CASCADE e ON UPDATE CASCADE.
- O atributo data\_inicio não pode ser nulo, logo é usada a restrição NOT NULL.
- A data de fim da reserva não pode ser nula e necessita de ser obrigatoriamente uma data depois da data de início, daí as restrições NOT NULL e CHECK(data\_fim > data\_inicio).

## Quarto

- O atributo nr\_quarto é chave primária.
- O atributo tipo\_quarto não pode ser nulo pois todos os quartos têm um tipo, daí a um cliente não pode ser atribuído um quarto sem este atributo. Este atributo também tem de ser de um tipo aceite pelo hotel (solteiro, solteiro duplo, casal, familiar). Com isto são usadas as restrições NOT NULL e CHECK.

## EpocaDoAno

- A chave primária é composta pelos atributos data\_inicio e data\_fim. Estes atributos não podem ser nulos e a data\_fim tem que ser maior do que a data\_inicio. As restrições usadas são NOT NULL e CHECK(data\_inicio < data\_fim).

## TaxaDeOcupacao

- A percentagem é a chave primária. Esta não pode ser maior que 100% ou menor que 0%, logo é usada a restrição CHECK(percentagem >= 0 AND percentagem <= 100).

## Servico

- O atributo id\_servico é a chave primária.
- O atributo preco não pode ser nulo e tem que ser superior ou igual a 0, por esta razão as restrições são NOT NULL e CHECK(preco >= 0).
- Os atributos nome e data não podem ser nulos, logo é usada a restrição NOT NULL.

## **ServicoDeQuarto TratamentoDeSpa e TipoDeBebidas**

- O atributo `id_servico` é chave primária e faz referência a `Servico`, quando apagado ou modificado na classe `Servico` é igualmente apagado e modificado nas classes `ServicoDeQuarto`, `TratamentoDeSpa` e `TipoDeBebida`, daí serem usadas as restrições `ON DELETE CASCADE` e `ON UPDATE CASCADE`.
- Os atributos `data`, `nome` e `preco` não podem ser nulos, logo é usada a restrição `NOT NULL`. O atributo `preco` também não pode ser inferior a zero, daí a restrição `CHECK(preco>=0)`.

## **Cartao**

- O atributo `id_cartao` é a chave primária.
- O atributo `nif` faz referência a `Pessoa`, quando apagado ou modificado na classe `Pessoa` é igualmente apagado e modificado na classe `Cartao`, daí serem usadas as restrições `ON DELETE CASCADE` e `ON UPDATE CASCADE`.
- O atributo `nr_pontos` não pode ser nulo, mesmo no caso de o cliente não ter pontos, o `nr_pontos` deve estar atribuído a zero. São então usadas as restrições `NOT NULL` e `CHECK(nr_pontos>=0)`.

## **Fatura**

- O atributo `id_fatura` é chave primária.
- O atributo `id_cartao` faz referência a `Cartao`, quando apagado ou modificado na classe `Cartao` é igualmente apagado e modificado na classe `Fatura`, daí serem usadas as restrições `ON DELETE CASCADE` e `ON UPDATE CASCADE`.
- O atributo `data` não pode ser nulo, logo é usada a restrição `NOT NULL`.
- O atributo `total` não pode ser nulo e tem que ser maior ou igual a zero, são então usadas as restrições `NOT NULL` e `CHECK(total>=0)`.

## **PrecoDeQuarto**

- A chave primária é composta pelos atributos `nr_quarto`, que faz referência a `Quarto`, `data_inicio` e `data_fim`, que fazem referência a `EpocaDoAno`, e `percentagem`, que faz referência a `TaxaDeOcupacao`. Quando apagados ou modificados nas classes a que fazem referência são igualmente apagados e modificados na classe `PrecoDeQuarto`, daí serem usadas as restrições `ON DELETE CASCADE` e `ON UPDATE CASCADE`.
- O atributo `preco` não pode ser nulo e tem que ser superior ou igual a zero, logo são usadas as restrições `NOT NULL` e `CHECK(preco>=0)`.



### **QuantidadeDeServiçosPrestados**

- A chave primária é composta pelos atributos id\_fatura e id\_servico que fazem referência a Fatura e Servico respectivamente, quando apagados ou modificados na classe Fatura e Servico são igualmente apagados e modificados na classe QuantidadeDeServiçosPrestados, daí serem usadas as restrições ON DELETE CASCADE e ON UPDATE CASCADE.
- O atributo quantidade não poder ser nulo e tem que ter um valor maior ou igual a zero, logo são usadas as restrições NOT NULL e CHECK(quantidade>=0).

### **QuantidadeDeBebidas**

- A chave primária é composta pelos atributos id\_cartao e id\_servico que fazem referência a Cartao e TipoDeBebida respectivamente, quando apagados ou modificados nas classes Cartao e TipoDeBebida são igualmente apagados e modificados na classe QuantidadeDeBebidas, daí serem usadas as restrições ON DELETE CASCADE e ON UPDATE CASCADE.
- O atributo quantidade não poder ser nulo e tem que ter um valor superior a zero, logo são usadas as restrições NOT NULL e CHECK(quantidade>=0).

### **DescontoAplicado**

- A chave primária é composta pelos atributos id\_cartao e id\_fatura que fazem referência a Cartao e Fatura respectivamente, quando apagados ou modificados na classe Cartao e Fatura são igualmente apagados e modificados na classe DescontoAplicado, daí serem usadas as restrições ON DELETE CASCADE e ON UPDATE CASCADE.
- O atributo desconto não poder ser nulo e tem que ter um valor superior a zero, logo são usadas as restrições NOT NULL e CHECK(desconto>=0).

## INTERROGAÇÕES

1. **Número de Funcionários do Hotel:** total de funcionários que trabalham no hotel.
2. **Reservas entre 20/03/2020 e 25/03/2020:** identificação das reservas que se encontram da data 20/03/2020 à data 25/03/2020.
3. **Nome dos clientes que já ficaram hospedados no quarto 301:** de todos os clientes foram escolhidos apenas os que já lhe foi atribuído o quarto número 301.
4. **Pessoa que frequenta o bar:** seleção de pessoas que frequentam o bar a partir do número de pontos, quem tem número de pontos do Cartão diferente de 0 já foi ao bar.
5. **Diferença média de pontos acumulados nos cartões de Clientes que ficaram hospedados em quartos de solteiro e Clientes que ficaram hospedados em quartos familiares.**
6. **Reservas a partir da data atual em ordem crescente da data:** de todas as reservas são selecionadas apenas as que começam a partir da data atual.
7. **Preço dos quartos de casal quando a taxa de ocupação do hotel é superior a 75% e época do ano de '20/03/2020' a '21/06/2020' :** de todos os preços de quartos são selecionados apenas os dos quartos de casal em que a taxa de ocupação seja superior a 75% e a época do ano seja de '20/03/2020' a '21/06/2020'.
8. **Nome da pessoa que teve a segunda fatura mais cara:** de todas as pessoas é somente selecionada aquela cujo o valor total da sua fatura é o segundo mais alto.
9. **Email das pessoas que passaram mais de 3 dias seguidos hospedados no hotel:** de todas as pessoas é unicamente selecionado o email daquela cujo a estadia seja superior a três dias.
10. **O valor total de cada fatura é o preço do quarto mais os serviços requisitados menos o desconto:** o total de cada fatura corresponde à soma de todos os quartos reservados pelo cliente associado a esta, mais a soma de todos os serviços requisitados por este, menos o desconto associado ao seu cartão.

## GATILHOS

1. **QuartoOcupado:** ocorre uma mensagem de erro quando é tentado juntar um tuplo à tabela Reserva cujo nr\_quarto já tenha sido reservado por outro cliente na mesma data.
2. **NifCheck:** ocorre uma mensagem de erro quando é tentado juntar um tuplo à tabela Pessoa cujo nif já tenha sido previamente inserido, ou seja, não é possível existirem duas ou mais pessoas com o mesmo nif.
3. **AdicionaPontos:** após a inserção de um tuplo na tabela QuantidadeDeBabidas, será atualizado o nr\_pontos, do cartão associado à inserção, na tabela Cartao, com o valor da soma de todos os serviços associados a esse cartão vezes as suas respetivas quantidades vezes dez.

Diagrama UML HOTEL

