

Sistema de Gestão de Centros Comerciais

Projeto de Bases de Dados – Grupo 303

Definição do Modelo Conceptual

7 de Março de 2021

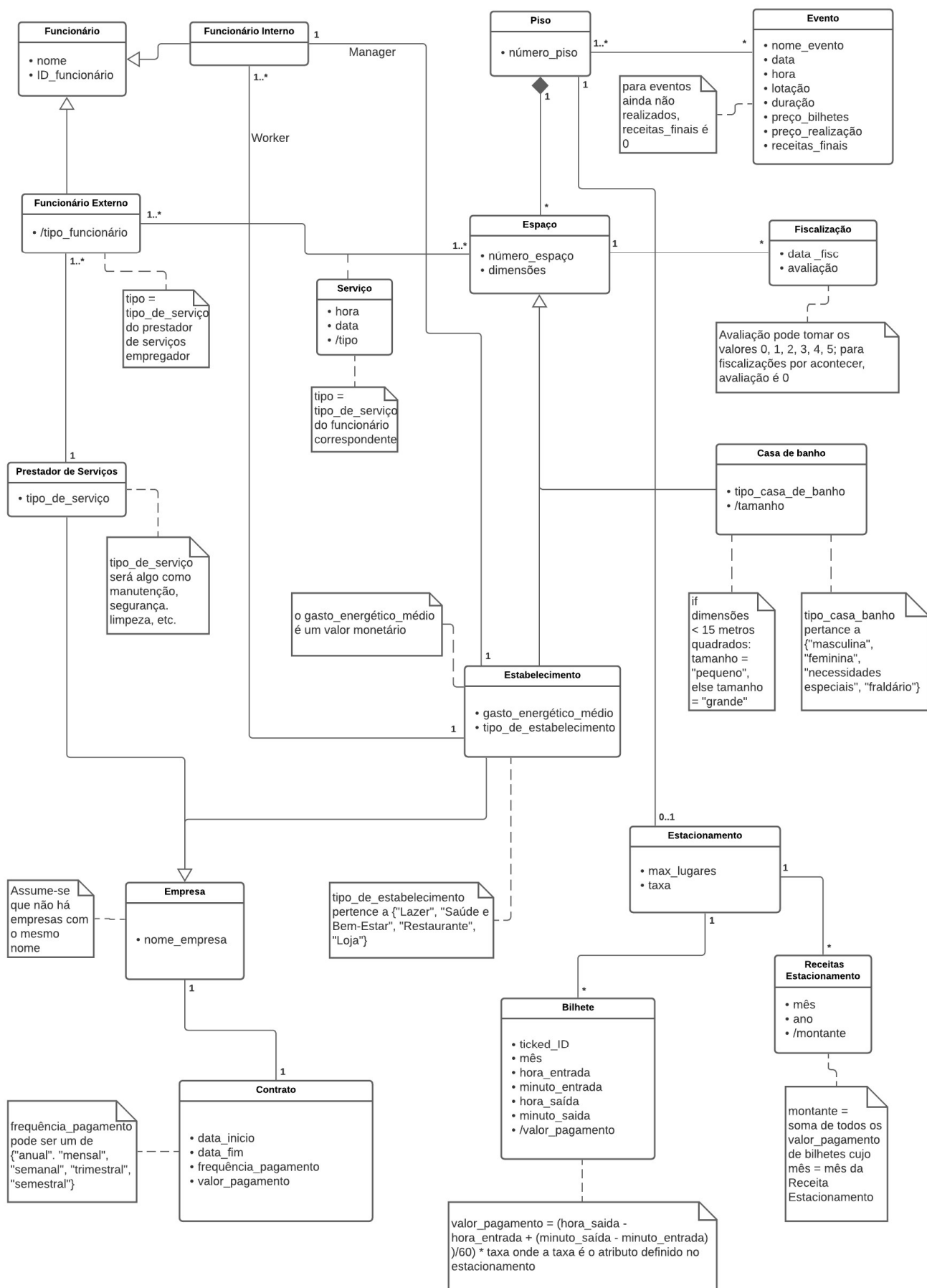
Trabalho realizado por:

Marcelo Henriques Couto – up201906086@fe.up.pt

Carolina Cintra Fernandes Figueira – up201906845@fe.up.pt

Índice

Esquema UML.....	3
Contextualização.....	4
Introdução.....	4
Organização, Pisos, Espaços e Fiscalizações.....	4
Eventos.....	4
Espaços – Estabelecimentos e Casas de Banho.....	5
Estacionamento.....	5
Empresas – Prestadores de Serviços e Estabelecimentos.....	5
Funcionários.....	6
Esquema UML Atualizado.....	7
Notas.....	8
Esquema Relacional.....	9
Relações.....	9
Análise de Dependências Funcionais e Formas Normais.....	10
Restrições e Notas.....	14
Detalhes de Implementação e Notas adicionais.....	20
Sources.....	21



Contextualização

Introdução

Pretende-se armazenar a informação relativa ao funcionamento de um Centro Comercial. Para isso é necessário registar os estabelecimentos presente no mesmo, bem como outros componentes da sua infraestrutura. É importante registar os contratos do centro com empresas externas e seus funcionários cuja atividade se relaciona com o mesmo, dados sobre Eventos ocorridos ou por ocorrer, fiscalizações a determinados espaços, etc. Para isso, fabricámos um sistema cujo UML está na página anterior e cuja contextualização mais detalhada será feita a seguir.

Organização Pisos, Espaços e Fiscalizações

O Centro Comercial está organizando por pisos, que são caracterizados por um número de identificação. Estes pisos são compostos por espaços (zero ou mais), que podem especializar-se em estabelecimentos ou casas de banho. Sobre os espaços é importante saber o respetivo número e dimensões, assim como manter informação sobre a fiscalização que é feita periodicamente por órgãos públicos, nomeadamente a sua data e avaliação. Os pisos estão também possivelmente associados a um parque de estacionamento e a eventos.

Eventos

Nos pisos é possível que ocorram eventos e sobre estes deve ser armazenado o nome, data, hora, lotação, duração, preço dos bilhetes, preço de realização e receitas finais. Os eventos guardados na base de dados podem ainda não ter acontecido, tendo o atributo `receitas_finais` o valor 0.

Espaços – Estabelecimentos e Casas de Banho

Um espaço no centro comercial pode ser também um estabelecimento ou uma casa de banho. Sobre os estabelecimentos é necessário saber o seu tipo, podendo este se enquadrar como “Lazer”, “Saúde e Bem-Estar”, “Restaurante” ou “Loja”. A estes estabelecimentos estão associados um ou mais funcionários, para além de um funcionário especial, o manager. As casas de banho, têm um tipo, que pode variar entre “masculina”, “feminina”, “necessidades especiais” e “fraldário”, e um tamanho cuja classificação é de “grande” para dimensões superiores a 15 metros quadrados ou “pequeno” para dimensões inferiores.

Estacionamento

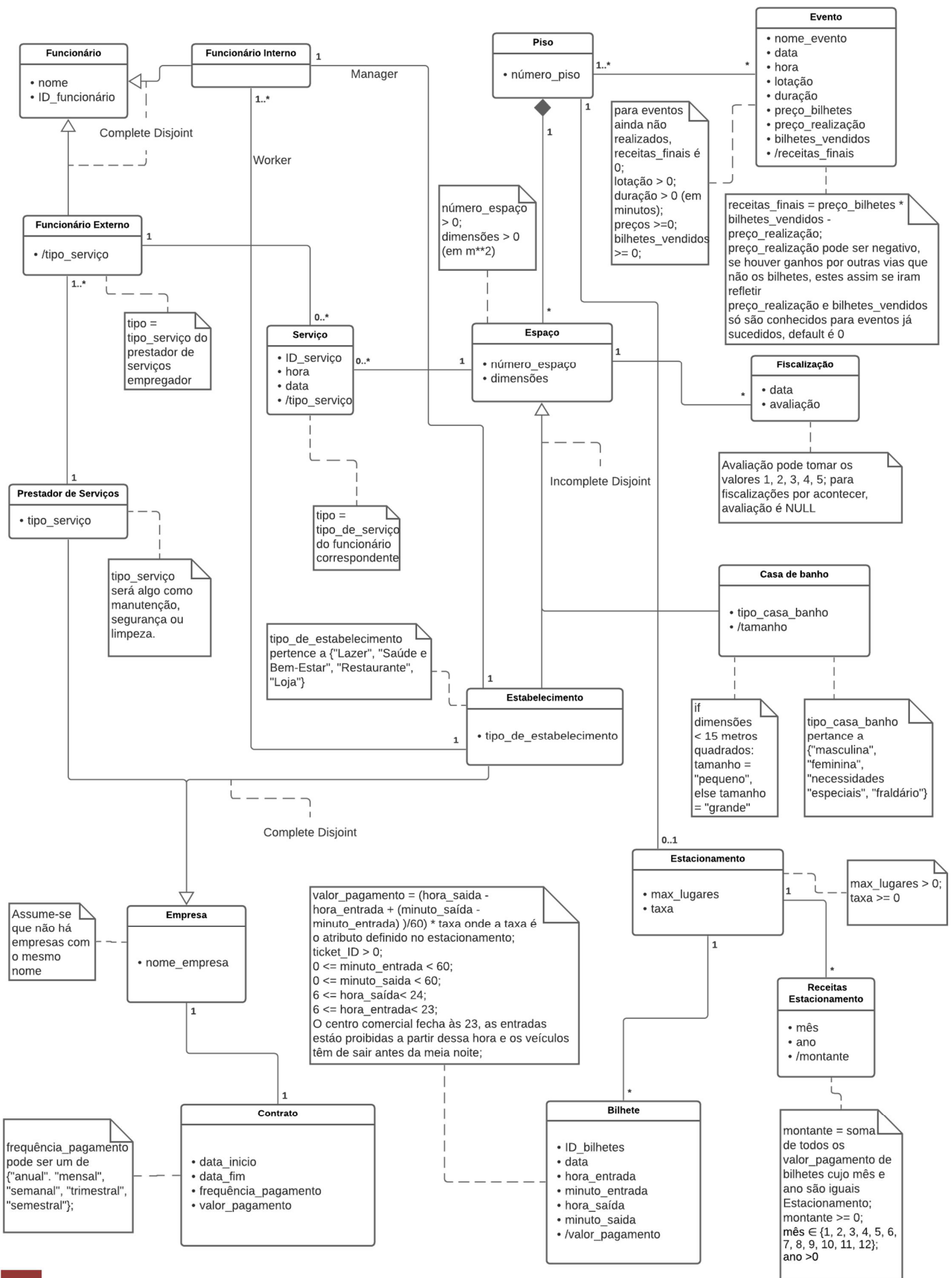
O estacionamento tem um número máximo de lugares e uma taxa, para além de estar associado às suas receitas mensais, que possuem o registo do seu respetivo mês, ano e montante. Associado ao estacionamento deve estar o registo dos bilhetes de estacionamento. Este deve salvaguardar o ID e o horário de chegada e saída do cliente, para o cálculo do valor de pagamento.

Empresas - Prestadores de Serviços e Estabelecimentos

Empresas que tem alguma ligação com o Centro Comercial devem ter um nome registado e podem enquadrar-se como prestadoras de serviços ou um estabelecimento do centro (um Estabelecimento tanto representa um Espaço como uma Empresa). Sobre a primeira deve-se armazenar o tipo de serviço que oferece. A esta estão associados funcionários externos. A descrição da segunda já foi feita anteriormente. A todas as empresas está associado um contrato com o centro. Este contrato, por sua vez, é caracterizado por datas de início e de fim, valor do pagamento e a frequência do mesmo (mensal, anual, etc.).

Funcionários

Os funcionários são caracterizados pelo seu ID e nome. Estes subdividem-se em funcionários *externos* e *internos*. Os internos estão ligados a estabelecimentos da forma especificada anteriormente: através destas ligações é possível diferenciar o gerente dos demais trabalhadores. As empresas prestadoras de serviços, por sua vez, têm funcionários externos. Estes podem ser também caracterizados pelo tipo de serviço que prestam aos espaços, sendo estes, por sua vez, caracterizados pela hora, data e tipo (“manutenção”, “segurança”, “limpeza”, etc.).



NOTAS NOVO UML

- Foram feitas as alterações sugeridas pelo professor nas anotações do primeiro relatório;
 - Mudança da classe de associação 'Serviço' para classe normal
 - Acréscimo de restrições sobre o valor obrigatório de alguns atributos e outras
 - Correção de multiplicidades
- Foi retirado o atributo 'consumo_energético' da relação 'Estabelecimento';
- Foi adicionado o atributo 'bilhetes_vendidos' à relação Evento;
- Foram acrescentadas e complementadas as notas e restrições referentes às classes e seus atributos;
- Foi adicionado a data e removido o mês em Bilhete.

Esquema Relacional

Relações

- Receita_Estacionamento (mês, ano, número_estacionamento -> Estacionamento, montante)
- Bilhete (ID_bilhete, data, hora_entrada, minuto_entrada, hora_saída, minuto_saída, valor_pagamento, número_estacionamento -> Estacionamento)
- Estacionamento (número_estacionamento -> Piso, max_lugares, taxa)
- Funcionário_Interno (ID_funcionário, nome, (número_espço, número_piso, nome_empresa) -> Estabelecimento)
- Funcionário_Externo (ID_funcionário, nome, tipo_serviço, nome_empresa -> Prestador_de_serviço)
- Prestador_de_Serviço (nome_empresa -> Empresa, tipo_serviço)
- Serviço (ID_serviço, hora, data, tipo_serviço, ID_funcionário -> Funcionário Externo, (número_espço, número_piso) -> Espaço)
- Empresa (nome_empresa)
- Contrato (nome_empresa -> Empresa, data_início, data_fim, frequência_pagamento, valor_pagamento)
- Estabelecimento (nome_empresa -> Empresa, (número_espço, número_piso) -> Espaço, tipo_de_estabelecimento, ID_gerente -> Funcionário_Interno)
- Fiscalização (ID_fiscalização, data, avaliação, (número_espço, número_piso) -> Espaço)
- Casa_de_Banho ((número_espço, número_piso) -> Espaço, tipo_casa_banho, tamanho)
- Espaço (número_espço, dimensões, número_piso -> Piso)
- Evento_Piso((nome_evento, data_evento) -> Evento, número_piso -> Piso)
- Piso (número_piso)
- Evento (nome_evento, data, hora, lotação, duração, preço_bilhetes, preço_realização, bilhetes_vendidos, receitas_finais)

Análise de Dependências Funcionais e Formas Normais

Legenda: ✓ -> não viola | ✗ -> viola

- Fiscalização (ID_fiscalização, data, avaliação, (número_espaco, número_piso)->Espaco)

FDs:

- ID_fiscalização -> data, avaliação, número_espaco, número_piso
- data, número_espaco, número_piso -> ID_fiscalização, avaliação

Formas Normais: BCNF ✓ 3NF ✓ ->Justificação 2

- Bilhete (ID_bilhete, data, hora_entrada, minuto_entrada, hora_saída, minuto_saída, valor_pagamento, número_estacionamento -> Estacionamento)

FDs:

- ID_bilhete -> data, hora_entrada, minuto_entrada, hora_saída, minuto_saída, número_estacionamento

Formas Normais: BCNF ✓ 3NF ✓ ->Justificação 2

- Receitas_Estacionamento (mês, ano, número_estacionamento -> Estacionamento, montante)

FDs:

- mês, ano, número_estacionamento -> montante

Formas Normais: BCNF ✓ 3NF ✓ ->Justificação 2

- Estacionamento (número_estacionamento -> Piso, max_lugares, taxa)

FDs:

- número_estacionamento -> max_lugares, taxa

Formas Normais: BCNF ✓ 3NF ✓ ->Justificação 2

- Funcionário_Interno (ID_funcionário, nome, (número_espço, número_piso, nome_empresa) -> Estabelecimento)

FDs:

- ID_funcionário -> nome, número_espço, número_piso, nome_empresa

Formas Normais: BCNF ✓ 3NF ✓ ->Justificação 2

- Funcionário_Externo (ID_funcionário, nome, tipo_serviço, nome_empresa -> Prestador_de_serviço)

FDs:

- ID_funcionário -> nome, tipo_de_serviço, nome_empresa

Formas Normais: BCNF ✓ 3NF ✓ ->Justificação 2

- Prestador_de_Serviço (nome_empresa -> Empresa, tipo_serviço)

FDs:

- nome_empresa -> tipo_serviço

Formas Normais: BCNF ✓ 3NF ✓ ->Justificação 2

- Serviço (ID_serviço, hora, data, tipo_serviço, ID_funcionário -> Funcionário Externo, (número_espço, número_piso) -> Espaço)

FDs:

- ID_serviço -> hora, data, tipo, ID_funcionário, número_espço, número_piso
- ID_funcionário, data, hora -> tipo, ID_serviço, número_espço, número_piso

Formas Normais: BCNF ✓ 3NF ✓ ->Justificação 2

- Empresa (nome_empresa)

Formas Normais: BCNF ✓ 3NF ✓ ->Justificação 1

- Contrato (nome_empresa -> Empresa, data_inicio, data_fim, frequência_pagamento, valor_pagamento)

FDs:

- nome_empresa -> data_inicio, data_fim, frequência_pagamento, valor_pagamento

Formas Normais: BCNF ✓ 3NF ✓ ->Justificação 2

- Estabelecimento (nome_empresa -> Empresa, (número_espaço, número_piso)->Espaço, tipo_de_estabelecimento, ID_gerente -> Funcionário_Interno)

FDs:

- número_espaço, número_piso, nome_empresa -> tipo_de_estabelecimento, ID_gerente

Formas Normais: BCNF e 3NF ->Justificação 2

- Casa_de_Banho ((número_espaço, número_piso) -> Espaço, tipo_casa_banho, tamanho)

FDs:

- número_espaço, número_piso-> tipo_casa_de_banho, tamanho

Formas Normais: BCNF ✓ 3NF ✓ ->Justificação 2

- Espaço (número_espaço, dimensões, número_piso -> Piso)

FDs:

- número_espaço, número_piso->dimensões

Formas Normais: BCNF ✓ 3NF ✓ ->Justificação 2

- Evento_Piso((nome_evento, data_evento) -> Evento, número_piso -> Piso)

Formas Normais: BCNF ✓ 3NF ✓ ->Justificação 1

- Piso (número_piso)

Formas Normais: BCNF ✓ 3NF ✓ -> Justificação 1

- Evento (nome_evento, data, hora, lotação, duração, preço_bilhetes, preço_realização, bilhetes_vendidos, receitas_finais, numero_piso)

FDs:

- nome_evento, data -> hora, lotação, duração, preço_bilhetes, preço_realização, bilhetes_vendidos, numero_piso

Formas Normais: BCNF ✓ 3NF ✓ -> Justificação 2

Observa-se portanto que todas as relações do modelo relacional seguem a Forma Normal de Boyce-Codd (BCNF) e, consequentemente, a 3ª Forma Normal (3NF). As relações que respeitam as duas formas normais referidas dividem-se em duas categorias: as que são justificáveis pela 'justificação 1' e as que o são pela 'justificação 2'.

- Justificação 1: A relação não tem dependências funcionais não triviais, o que é condição suficiente para que siga tanto a BCNF como a 3NF;
- Justificação 2: Para que uma relação esteja na BCNF, é condição necessária e suficiente que, para todas as FDs $A \rightarrow B$ não triviais, A seja uma super-chave ou *superkey*. Um atributo ou conjunto de atributos é super-chave quando o seu fecho (A^+) inclui todos os atributos da relação em questão. Esta é também condição suficiente para que a relação esteja em 3NF, justificando-se assim a causalidade referida anteriormente;

Outra condição suficiente para que uma relação estivesse na Terceira Forma Normal seria a de os atributos à direita da FD serem atributos primos, ou seja, membro de uma chave. No entanto esta condição não é útil em nenhuma situação nesta base de dados.

Nota: dependências funcionais que se baseassem em atributos derivados não foram consideradas, visto que estes apenas poderão ser implementados na próxima entrega.

Restrições e Notas

De forma a garantir a segurança e bom funcionamento da base de dados, são necessárias algumas restrições. As restrições 'PRIMARY KEY' refletem que um atributo ou conjunto de atributos é chave primária da relação, o que significa que não poderão haver dois tuplos diferentes com o este conjunto de atributos igual. 'FOREIGN KEY' surge junto a atributos que, logicamente, são foreign keys. As restrições 'NOT NULL' indicam que um dado atributo ou conjunto de atributos terá obrigatoriamente um valor definido. 'DEFAULT' será utilizado para indicar o valor que um atributo terá caso nenhum seja indicado. 'UNIQUE' refere-se a um dado atributo cujo valor tenha de ser obrigatoriamente diferente para tuplos diferentes. Por último, 'CHECK' será utilizado para garantir uma dada condição específica que não possa ser assegurada pelas outras restrições.

Receita_Estacionamento

- PRIMARY KEY (mês, ano, número_estacionamento)
- número_estacionamento FOREIGN KEY Estacionamento
- montante NOT NULL
- montante CHECK (montante ≥ 0)
- mês CHECK (mês $\in \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$)
- ano CHECK (ano > 0)

Notas: montante é suposto ser calculado através da soma dos valor_pagamento dos Bilhetes com o mesmo mês e ano(atributo dependente). Esta funcionalidade só será implementada na 3ª entrega com recurso a triggers, views ou outro. Consequentemente o atributo não estará presente no sql.

Estacionamento

- número_estacionamento PRIMARY KEY
- número_estacionamento FOREIGN KEY Piso
- taxa NOT NULL
- max_lugares NOT NULL
- max_lugares CHECK (max_lugares > 0)
- taxa CHECK (taxa ≥ 0)

Notas: taxa pode ter o valor 0, significa que o estacionamento é de graça

Bilhete

- ID_bilhete PRIMARY KEY
- número_estacionamento FOREIGN KEY
- número_estacionamento NOT NULL
- data NOT NULL
- hora_entrada NOT NULL
- minuto_entrada NOT NULL
- valor_pagamento NOT NULL
- hora_entrada CHECK (6 <= hora_entrada <= 22)
- hora_saída CHECK (6 <= hora_saída <= 23)
- (minuto_entrada, minuto_saída) CHECK (0 <= minuto_entrada, minuto_saída <= 59)
- data DEFAULT (data atual)

Notas:

- valor_pagamento é calculado através com recurso à hora e minuto de entrada e saída e à taxa do parque de estacionamento (atributo dependente). Esta funcionalidade só será implementada na 3ª entrega com recurso a triggers, views ou outro. Consequentemente o atributo não estará presente no sql
- assume-se que o centro comercial encerra às onze da noite, não podendo haver entradas a partir desta hora e havendo obrigatoriedade dos carros de sair antes da meia-noite

Funcionário Interno

- ID_funcionário PRIMARY KEY
- (número_espço, número_piso, nome_empresa) FOREIGN KEY Estabelecimento
- nome NOT NULL
- número_espço CHECK (número_espço > 0)

Notas: número_espço, número_piso e nome_empresa poderão ser NULL no caso do manager (atributo dependente). Esta funcionalidade só será implementada na 3ª entrega com recurso a triggers. Consequentemente o atributo não estará presente no sql.

Funcionário_Externo

- ID_funcionário PRIMARY KEY
- nome_empresa FOREIGN KEY Prestador_de_Serviços
- nome_empresa NOT NULL
- nome NOT NULL
- tipo_serviço NOT NULL
- tipo_serviço CHECK (tipo_serviço ∈ {'Segurança', 'Limpeza', 'Manutenção'})

Notas: O atributo tipo_serviço deverá ser o mesmo que o tipo_serviço do Prestador_de_Serviços identificado por nome_empresa (atributo dependente). Esta funcionalidade só será implementada na 3ª entrega com recurso a triggers, views ou outro. Consequentemente o atributo não estará presente no sql.

Empresa

- nome_empresa PRIMARY KEY

Prestador_de_Serviços

- nome_empresa PRIMARY KEY
- nome_empresa FOREIGN KEY Empresa
- tipo_serviço NOT NULL
- tipo_serviço CHECK (tipo_serviço ∈ {'Segurança', 'Limpeza', 'Manutenção'})

Contrato

- nome_empresa PRIMARY KEY
- nome_empresa FOREIGN KEY Empresa
- data_início NOT NULL
- data_fim NOT NULL
- valor_pagamento NOT NULL
- frequência_pagamento NOT NULL
- frequência_pagamento DEFAULT 'mensal'
- frequência_pagamento CHECK (frequência_pagamento ∈ {'anual', 'mensal', 'semanal', 'trimestral', 'semestral'})

Notas: valor_pagamento negativo refere-se a um contrato onde quem efetua o pagamento é o centro comercial.

Estabelecimento

- PRIMARY KEY (nome_empresa, número_espço, número_piso)
- nome_empresa FOREIGN KEY Empresa
- (número_espço, número_piso) FOREIGN KEY Espaço
- ID_gerente FOREIGN KEY Funcionário Interno
- ID_gerente NOT NULL
- tipo_estabelecimento NOT NULL
- tipo_estabelecimento CHECK (tipo_estabelecimento ∈ {'Lazer', 'Saúde e Bem-Estar', 'Restaurante', 'Loja'})
- número_espço CHECK (número_espço > 0)

Notas: o gerente requer o uso de triggers. Consequentemente o atributo não estará presente no sql e será implementado na 3ª entrega.

Espço

- PRIMARY KEY (número_espço, número_piso)
- número_piso FOREIGN KEY Piso
- dimensões NOT NULL
- dimensões CHECK (dimensões > 0)
- número_espço CHECK (número_espço > 0)

Notas: dimensões estão em m**2

Casa_de_Banho

- PRIMARY KEY (número_espço, número_piso)
- (número_espço, número_piso) FOREIGN KEY Espaço
- tipo_casa_banho NOT NULL
- tamanho NOT NULL
- tipo_casa_banho CHECK (tipo_casa_banho ∈ {'Masculina', 'Feminina', 'Necessidades especiais', 'Fraldário'})
- tamanho CHECK (tamanho ∈ {'grande', 'pequeno'})
- número_espço CHECK (número_espço > 0)

Notas: tamanho depende das dimensões do Espaço associado (atributo derivado). Esta funcionalidade só será implementada na 3ª entrega com recurso a triggers, views ou outro. Consequentemente o atributo não estará presente no sql.

Fiscalização

- ID_fiscalização PRIMARY KEY
- (número_espço, número_piso) FOREIGN KEY Espaço
- número_espço NOT NULL
- número_piso NOT NULL
- data NOT NULL
- avaliação CHECK (avaliação $\in \{1, 2, 3, 4, 5\}$)
- UNIQUE (data, número_espço, número_piso)

Serviço

- ID_serviço PRIMARY KEY
- (número_espço, número_piso) FOREIGN KEY Espaço
- ID_funcionário FOREIGN KEY Funcionário_Externo
- hora NOT NULL
- data NOT NULL
- tipo_serviço NOT NULL
- número_espço NOT NULL
- número_piso NOT NULL
- ID_funcionário NOT NULL
- tipo_serviço CHECK (tipo_serviço $\in \{\text{'Segurança'}, \text{'Limpeza'}, \text{'Manutenção'}\}$)
- UNIQUE (ID_funcionário, data, hora)

Notas: O atributo tipo_serviço deverá ser o mesmo que o tipo_serviço do Prestador_de_Serviços identificado por nome_empresa (atributo derivado). Esta funcionalidade só será implementada na 3ª entrega com recurso a triggers, views ou outro. Consequentemente o atributo não estará presente no sql.

Piso

- número_piso PRIMARY KEY

Evento_Piso

- PRIMARY KEY (nome_evento, data_evento, número_piso)
- (nome_evento, data_evento) FOREIGN KEY Evento
- número_piso FOREIGN KEY Piso

Notas: Um evento pode estar a ocorrer em vários pisos ao mesmo tempo.

Evento

- PRIMARY KEY (nome_evento, data)
- hora NOT NULL
- lotação NOT NULL
- duração NOT NULL
- preço_bilhetes NOT NULL
- preço_realização NOT NULL
- bilhetes_vendidos NOT NULL
- receitas_finais NOT NULL
- receitas_finais DEFAULT 0
- bilhetes_vendidos DEFAULT 0
- preço_realização DEFAULT 0
- hora CHECK (0 <= hora <= 22)
- lotação CHECK (lotação > 0)
- duração CHECK (duração > 0)
- preço_bilhetes CHECK (preço_bilhetes >= 0)
- bilhetes_vendidos CHECK (bilhetes_vendidos >= 0)
- receitas_finais CHECK (receitas_finais >= 0)

Notas:

- duração é em minutos
- assume-se que só pode existir um evento com um dado nome e data. As receita_finais são calculadas com recurso ao (número de) bilhetes_vendidos, preço_realização e preço_bilhetes (atributo_derivado). Assume-se que os únicos lucros obitos num evento são obtidos através dos blihetes. Qualquer outro bónus é refletido na diminuição do preço_realização, podendo este ser, portanto, negativo. Esta funcionalidade só será implementada na 3ª entrega com recurso a triggers, views ou outro. Consequentemente o atributo não estará presente no sql
- o atributo hora refere-se à hora de início
- os bilhetes_vendidos e o preço_realização só é conhecido para eventos já sucedidos.

Detalhes de Implementação e Notas Adicionais

Generalizações

No UML havia 3 generalizações: uma com respeito aos funcionários, outra às empresas e outra aos espaços. A dos funcionários foi implementada recorrendo ao 'Object Oriented Style'. Como se tratava de uma generalização 'complete disjoint', a relação Funcionário não foi implementada. Nos outros dois casos, foi utilizado o 'E/R Style'.

Integridade Referencial no SQL

No caso de um Evento ser eliminado, assume-se que Evento_Piso passa a ser irrelevante (ON DELETE CASCADE). Proíbe-se, no entanto, que seja atualizado o nome do evento associado (representaria mudar o evento em si) (ON UPDATE RESTRICT).

Para os elementos Estabelecimento, não permitimos que a empresa associada seja atualizada (mudar o nome) nem eliminada. No entanto, para os prestadores de serviços, caso a empresa seja eliminada, assim acontecerá com o mesmo, bem como aos seus funcionários (ON UPDATE RESTRICT ON DELETE CASCADE).

Se o espaço ou funcionário associado a um serviço for eliminado, achámos que este também o deveria ser (ON DELETE CASCADE). No entanto, não consideramos que seja permitido alterar o Funcionário Externo responsável ou o Espaço destino deste Serviço.

Por último, não é permitida a alteração de uma empresa associada a um Contrato, mas caso a empresa seja eliminada, este também o será (ON UPDATE RESTRICT ON DELETE CASCADE).

Queries

1. Resulta numa tabela que apresenta uma linha e duas colunas, apresentando a dimensão média dos estabelecimentos e a dimensão média das casas de banho cada uma.
2. Valor que cada empresa associada a um estabelecimento paga por mês por metro quadrado de área do espaço do seu estabelecimento.
3. Lista o nome dos funcionários que prestaram serviços em casas de banho do tipo fraldários, por ordem alfabética do seu nome, bem como os respetivos espaço e piso
4. Lista os prestadores de serviços e o número de serviços prestados, por ordem decrescente do número de serviços
5. Número de fiscalizações com avaliação superior a 3 por cada espaço do centro comercial (identificados por número de espaço e número de piso)
6. Número de funcionários nos estabelecimentos do tipo 'Loja'
7. Funcionários que trabalham na empresa de nome 'NewClean Limpeza' e nunca fizeram um serviço num estabelecimento do tipo 'Lazer'
8. Número de pessoas que estacionaram no piso -2 após o início do evento mais lucrativo
9. Nome da empresa de serviços cuja média de avaliação dos espaços onde ditos serviços foram feitos é a melhor de entre todos os prestadores de serviços, e respetiva média
10. Nome dos eventos que têm saldo positivo, ocorreram há menos de dois anos e aconteceram em todos os pisos ao mesmo tempo

Gatilhos

Gatilho 1 – Cálculo do Valor das Receitas de Estacionamento

- `montanteReceitasUpd`
- `montanteReceitasDel`

Dois *triggers* foram implementados para garantir o cálculo correto do valor da Receita de Estacionamento correspondente a cada mês e ano e a cada parque de estacionamento. Para isto, o primeiro *trigger* garante que, a cada atualização do valor de um bilhete do mesmo mês e ano que os da receita, o montante da receita sofrerá as alterações necessárias. Este *trigger* é também ativado na adição de um novo bilhete devido ao comportamento do **Gatilho 2**. Por sua vez, o segundo está encarregue de, na remoção de um bilhete da base de dados, o seu valor seja subtraído ao do montante da receita de estacionamento com o mesmo mês e ano.

Gatilho 2 – Cálculo do Valor de Pagamento do Bilhete

- `insertBilhete`
- `updateBilhete`

O cálculo do valor de pagamento de um bilhete é conseguido através do uso a estes dois *triggers*. O primeiro é executado na sequência da inserção de um novo bilhete e calcula o valor de pagamento de um bilhete consoante as horas e minutos de entrada e saída e a taxa do parque de estacionamento a que corresponde. O segundo realiza a mesma operação mas no seguimento de uma atualização num bilhete já existente.

Nota: No **Gatilho 1**, o `montanteReceitasUpd` é também executado na inserção de um bilhete porque qualquer uma das partes do **Gatilho 2** o desencadeia, visto ambas executarem uma atualização no valor de pagamento do bilhete. Isto foi desta forma porque era necessário que o *trigger* '`montanteReceitasUpd`' fosse executado apenas após algum destes.

Gatilho 3 – Manager de Estabelecimento

- managerAssign

Este *trigger* é responsável por controlar a inserção de estabelecimentos na base de dados no que toca aos seus managers. Este trigger está dividido em duas partes:

1. impede a adição de um estabelecimento se o seu 'ID_manager' referenciar um funcionário que já trabalha noutro estabelecimento, ou seja, cujo estabelecimento não esteja a NULL. No caso disto acontecer uma mensagem de erro é apresentada, e a ação é cancelada
2. está encarregue de, no caso de o manager ser válido, atualizar os dados sobre o estabelecimento em que trabalha, para que deixe de o ser para qualquer outro Estabelecimento

Sources

Fotografia de Fundo da Capa:

[https://commons.wikimedia.org/wiki/File:Shopping_Center_Dolce_Vita_Tejo_-_Amadora_\(Portugal\)_\(3517775477\).jpg](https://commons.wikimedia.org/wiki/File:Shopping_Center_Dolce_Vita_Tejo_-_Amadora_(Portugal)_(3517775477).jpg)