

# Base de Dados

*2018/2019*

Alunos:

- Gabriel Ambrósio, nº160221013
- Hugo Ferreira, nº160221039

Docente:

- Luis Damas

Turma(Laboratório):

- INF-03 (Terça 13h/15h)

# Índice

1. Sumário.....	3
2. Definição do domínio do problema.....	3
4. Especificação de requisitos funcionais.....	4
5. Modelo Entidade Relação	
5.1 – Especificação das entidades.....	5
5.2 – Especificação dos relacionamentos.....	5
5.3 – Especificação das relações e restrições.....	6
5.4 – Diagrama do modelo entidade-relação.....	7
6. Modelo Relacional	
6.1 – Especificação das tabelas.....	8
6.2 – Diagrama do modelo relacional.....	10
7. Demonstração e testes.....	11
8. Conclusões e Análises de Limitações.....	13

## Sumário

Este projeto advém da necessidade de criar uma plataforma que suporta a gestão de uma empresa de eventos variados que têm interesse em guardar e manipular essas informações sobre os utilizadores e que tipo de utilizadores são, os eventos/sessões, as salas onde essas sessões são apresentadas, a localização onde decorre o evento e quaisquer outros dados necessários que seja preciso persistir.

Posteriormente à criação da base de dados através dos creates e alters vai ser necessário preencher as tabelas com valores exclusivamente inventados para simular a consulta desses dados.

O objetivo principal deste projeto é, através do enunciado, fazer a análise, levantamento e especificação de requisitos e desenvolver um diagrama entidade-relação que posteriormente vai gerar um diagrama de um modelo relacional e fazer consultas baseados em vários aspetos do domínio do problema.

## Definição do domínio do problema

Para tentar cumprir o objetivo descrito no enunciado do projeto o primeiro passo foi identificamos as entidades necessárias para enquadrar os conhecimentos obtidos durante a unidade curricular com o domínio do sistema pretendido. Foram identificadas as seguintes entidades:

**Tipo de Utilizador** – tal como é descrito no enunciado, o utilizador ou é um ‘User’ do evento ou um ‘VIP’. Pode também ser um membro do ‘Staff’ encarregue de coordenar as sessões, ou um ‘Speaker’ de uma sessão. Podem ainda haver os utilizadores ‘Admin’ que estão encarregues de administrar o sistema e a base de dados;

**Utilizador** – qualquer utilizador terá dados que serão precisos persistir como o username, a password e outros detalhes;

**Avaliação** – todos os utilizadores, após saírem da sessão tem a oportunidade de fazer uma avaliação da sessão mas podem optar por não a fazer;

**Evento** - ;

**Localização** – esta vai servir para guardar o lugar onde é realizado o evento;

**Sessão** – um evento é composto por várias sessões e é necessário guardar informações sobre estas;

**Suporte de Sessão** – vai ser preciso guardar a informação relativa aos equipamentos que são ou não usados durante uma sessão;

**Equipamento** – informações sobre cada equipamento utilizado;

**Sala** – cada sessão vai ser realizada numa sala e é preciso persistir vários aspetos sobre o sítio onde a sessão decorre;

**Pack** – em cada sessão vai haver um pack de catering para que os utilizadores possam aproveitar de uma boa sessão;

## Especificação de requisitos funcionais

ID	Descrição
RF01	Permitir guardar informações sobre os utilizadores
RF02	Permitir guardar informações sobre as sessões, as salas e o plano de suporte às sessões
RF03	Permite fazer avaliações às sessões
RF04	Os utilizadores devem registar-se num evento através de uma inscrição
RF05	Permitir que cada sessão tenha um pack de serviço de catering

## Modelo Entidade-Relação

### Tabela de Entidades

Entidade	Atributos
<i>Tipo de Utilizador:</i>	designação (PK)
<i>Utilizador:</i>	username (PK) password email data de nascimento
<i>Evento:</i>	id (PK) designação hora de início hora de fim
<i>Sala:</i>	número (PK) descrição capacidade
<i>Localização:</i>	id (PK) morada
<i>Sessão:</i>	id (PK) designação hora de início hora de fim duração
<i>Equipamento:</i>	código (PK) designação preço quantidade
<i>Suporte Sessão:</i>	id (PK)
<i>Pack:</i>	designação (PK) descrição peso quantidade
<i>Avaliação:</i>	código (PK) classificação comentário data

### Relacionamentos

O utilizador tem um tipo de utilizador.

Um tipo de utilizador pode pertencer a vários utilizadores.

Uma avaliação é feita por um utilizador.

Um utilizador pode fazer várias avaliações.

Um evento regista vários utilizadores através da inscrição.

Um utilizador pode inscrever-se em vários eventos.

Um evento é constituído por várias sessões.

Uma sessão é feita num único evento.

Um evento é realizado numa só localização.

Uma localização pode albergar vários eventos.

Uma sessão realiza-se em apenas uma sala.

Uma sala pode ser utilizada para várias sessões.

Uma sessão é auxiliada por um suporte de sessão.

Um suporte de sessão suporta uma sessão.

Um equipamento faz parte de vários suportes de sessão.

Um suporte de sessão é composto por vários equipamentos.

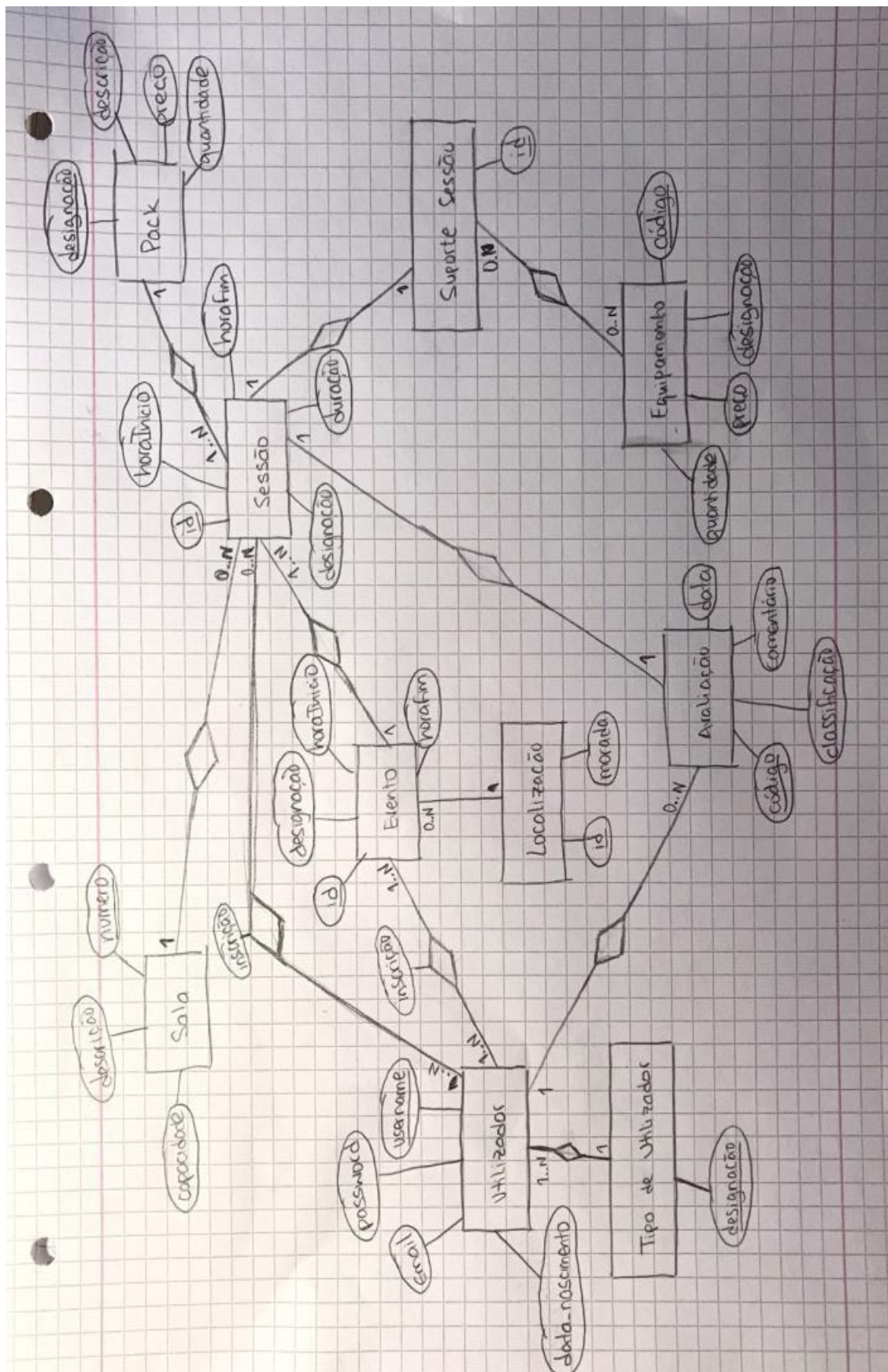
Uma sessão tem um pack de catering.

Um pack de catering pode ser fornecido para várias sessões.

## Tabela de Relações e Restrições

Entidade	Relação	Entidade
<i>Tipo de Utilizador</i>	<b>1 – 1..N</b>	<i>Utilizador</i>
<i>Utilizador</i>	<b>1..N – 1..N</b>	<i>Evento</i>
<i>Evento</i>	<b>0..N – 1</b>	<i>Localização</i>
<i>Sala</i>	<b>1 – 0..N</b>	<i>Sessão</i>
<i>Sessão</i>	<b>1..N – 1</b>	<i>Evento</i>
<i>Pack</i>	<b>1 – 1..N</b>	<i>Sessão</i>
<i>Sessão</i>	<b>1 – 1</b>	<i>Avaliação</i>
<i>Avaliação</i>	<b>0..N – 1</b>	<i>Utilizador</i>
<i>Equipamento</i>	<b>0..N – 0..N</b>	<i>Suporte Sessão</i>
<i>Suporte Sessão</i>	<b>1 – 1</b>	<i>Sessão</i>
<i>Sessão</i>		<i>Utilizador</i>

## Diagrama Entidade-Relação



# Modelo Relacional

## Especificação das tabelas

Tipo de Utilizador			
Atributo	Tipos de Dados	Not Null	Chave(s)
designação	varchar(15)	S	PK

Utilizador			
Atributo	Tipos de Dados	Not Null	Chave(s)
username	varchar(200)	S	PK
password	varchar(20)	S	
email	varchar(30)	S	
data de nascimento	date	S	
tp_designação	varchar(15)	S	FK

Evento			
Atributo	Tipos de Dados	Not Null	Chave(s)
id	int	S	PK
designação	varchar(200)	S	
hora de início	timestamp	S	
hora de fim	timestamp	S	

Sala			
Atributo	Tipos de Dados	Not Null	Chave(s)
número	int	S	PK
descrição	varchar(250)	S	
capacidade	int	S	

Localização			
Atributo	Tipos de Dados	Not Null	Chave(s)
id	int	S	PK
morada	varchar(100)		



Sessão			
Atributo	Tipos de Dados	Not Null	Chave(s)
id	int	S	PK
designação	varchar(200)	S	
hora de início	timestamp	S	
hora de fim	timestamp	S	
duração	int	S	
ev_id	int	S	

Equipamento			
Atributo	Tipos de Dados	Not Null	Chave(s)
código	int	S	PK
designação	varchar(100)	S	
preço	int	S	
quantidade	int	S	

Suporte Sessão			
Atributo	Tipos de Dados	Not Null	Chave(s)
id	int	S	PK
número de equipamentos	int	--	

Pack			
Atributo	Tipos de Dados	Not Null	Chave(s)
designação	varchar(10)	S	PK
descrição	varchar(250)	S	
preço	int	S	
quantidade	int	S	

Avaliação			
Atributo	Tipos de Dados	Not Null	Chave(s)
código	int	S	PK
classificação	int	S	
comentário	varchar(250)	S	
data	datetime	S	

The diagram illustrates a database schema for a cinema system. It includes the following tables and their attributes:

- tipoutilizador**: tpUtil\_designacao VARCHAR(15)
- utilizador**: ut\_username VARCHAR(200), ut\_password VARCHAR(20), ut\_email VARCHAR(30), ut\_data\_nascimento DATE, ut\_tpUtil\_designacao VARCHAR(15)
- utilizador\_avaliacao**: utAVL\_ut\_username VARCHAR(200), utAVL\_avl\_codigo INT(11)
- sessao\_avaliacao**: sesAVL\_ses\_id INT(11), sesAVL\_avl\_codigo INT(11)
- sessao\_pack**: sesPc\_ses\_id INT(11), sesPc\_pack\_designacao VARCHAR(200)
- pack**: pk\_designacao VARCHAR(10), pk\_descricao VARCHAR(250), pk\_preco INT(11), pk\_quantidade INT(11)
- sessao**: ses\_id INT(11), ses\_designacao VARCHAR(200), ses\_hora\_inicio TIMEST AMP, ses\_hora\_fim TIMEST AMP, ses\_duracao INT(11), ses\_ev\_id INT(11)
- sala**: sala\_numero INT(11), sala\_descricao VARCHAR(250), sala\_capacidade INT(11)
- evento**: ev\_id INT(11), ev\_designacao VARCHAR(200), ev\_hora\_inicio TIMEST AMP, ev\_hora\_fim TIMEST AMP
- evento\_localizacao**: evLcl\_ev\_id INT(11), evLcl\_lcl\_id INT(11)
- localizacao**: lcl\_id INT(11), lcl\_morada VARCHAR(100)
- equipamento\_sessao**: equiSes\_equi\_codigo INT(11), equiSes\_sp\_id INT(11)
- equipamento**: equi\_codigo INT(11), equi\_designacao VARCHAR(100), equi\_preco INT(11), equi\_quantidade INT(11)
- suporte\_sessao**: sp\_id INT(11), sp\_numeroEquipamentos INT(11)
- suportesessao\_sessao**: spSesSes\_sesSup\_id INT(11), spSesSes\_ses\_id INT(11), spSesSes\_designacao VARCHAR(200), spSesSes\_hora\_inicio TIMEST AMP, spSesSes\_hora\_fim TIMEST AMP, spSesSes\_duracao INT(11)

Relationships are indicated by lines with crow's foot notation:

- tipoutilizador** to **utilizador**: 1:M relationship.
- utilizador** to **utilizador\_avaliacao**: 1:M relationship.
- utilizador\_avaliacao** to **sessao\_avaliacao**: 1:M relationship.
- sessao\_avaliacao** to **sessao**: 1:M relationship.
- sessao** to **sala**: 1:M relationship.
- sessao** to **evento**: 1:M relationship.
- sessao** to **pack**: 1:M relationship.
- evento** to **evento\_localizacao**: 1:M relationship.
- evento\_localizacao** to **localizacao**: 1:M relationship.
- evento** to **equipamento\_sessao**: 1:M relationship.
- equipamento\_sessao** to **equipamento**: 1:M relationship.
- equipamento\_sessao** to **suporte\_sessao**: 1:M relationship.
- suporte\_sessao** to **suportesessao\_sessao**: 1:M relationship.

## Documentação sucinta de todos os elementos desenvolvidos

### Demonstração e testes

#### QUERIES

/\*Q1 Lista de Locais nunca usados em eventos;\*/

```
SELECT lcl_id as 'Identificador Localização', lcl_morada as 'Morada'
FROM localizacao
WHERE lcl_id NOT IN (SELECT lcl_id
                     FROM localizacao JOIN evento JOIN evento_localizacao
                     ON evLcl_lcl_id = lcl_id AND evLcl_ev_id = ev_id);
```

/\*Q2 Lista de utilizadores VIP que participaram nalgum evento e respetivas sessões;\*/

```
SELECT DISTINCT ut_username as 'Nome do Utilizador', utSes_ses_id as
'Identificador Sessão'
FROM utilizador, tipoUtilizador, utilizador_evento, utilizador_sessao
WHERE ut_username = utEv_ut_username AND ut_username =
utSes_ut_username AND utEv_ut_username = utSes_ut_username AND
ut_tpUt_designacao = tpUt_designacao AND tpUt_designacao = 'VIP';
```

/\*Q3 Lista dos participantes que não estão inscritos em nenhum evento;\*/

```
SELECT ut_username as 'Nome do Utilizador'
FROM utilizador, utilizador_evento
WHERE utEv_ev_id NOT IN (SELECT ev_id
                        FROM evento, utilizador_evento
                        WHERE ev_id = utEv_ev_id);
```

## PROCEDURES

call sp\_numero\_utilizadores();

call sp\_item\_por\_sessao();

call sp\_MediaAvaliacoes();

call sp\_comentarios();

call sp\_sesoes(1);

## VIEWS

SELECT \* FROM vUtilizadores;

SELECT \* FROM vAvaliacoes;

SELECT \* FROM vHorariosEventos;

## Limitações e Conclusões

Dentro do separador das limitações do projeto são verificáveis algumas lacunas no que diz respeito ao modelo entidade-relação e conseqüentemente no modelo relacional. É bastante óbvio que montar um sistema de gestão de base de dados em cima de um modelo relacional não completamente certo e cem por cento funcional é complicado. As maiores dificuldades foram mesmo conseguir transpor todas as indicações sobre dados que era suposto persistir no sistema e passá-los para um modelo que cumpra totalmente isso.

Passando para as conclusões, tirando as limitações acima descritas pensamos, ainda assim, ter um projeto que cumpre bem muitas das funcionalidades pedidas sem pôr em causa a qualidade do sistema.