



**Universidade do Minho**  
Licenciatura em Ciências da Computação

## **Unidade Curricular de Bases de Dados**

Ano Letivo de 2023/2024

### **Gestão e Divulgação do Calendário de Eventos de uma Cidade**

**Bruno Neiva(a95311), Rúben Silva(a94633),  
José Ferreira(a96798), Rui Torres(a84588)**

# **BD**

Data de Receção	
Responsável	
Avaliação	
Observações	

## Gestão e Divulgação do Calendário de Eventos de uma Cidade

**Bruno Neiva(a95311), Rúben Silva(a94633),  
José Ferreira(a96798), Rui Torres(a84588)**

Novembro, 2023

# Resumo

No presente trabalho, abordamos a análise, planejamento, modelação, arquitetura e implementação de um Sistema de Base de Dados (SBD) relacional, com foco na gestão organização do calendário de eventos de uma cidade. A pesquisa começa com uma contextualização do projeto, ressaltando a sua relevância no contexto da gestão de eventos urbanos. Fundamentamos o nosso estudo em conceitos essenciais de bases de dados relacionais.

Os nossos objetivos abrangem a definição, motivação e viabilidade do SBD, com ênfase nas vantagens operacionais que o sistema proporcionará. Detalhamos os recursos necessários, a equipa de trabalho envolvida e um plano de execução do projeto, estabelecendo marcos e cronogramas.

Na etapa seguinte, definimos os requisitos do SBD. Utilizamos um método de levantamento e análise de requisitos, organizando-os em categorias de descrição, exploração e controle. Validamos rigorosamente esses requisitos para garantir sua clareza e detalhamento.

A fase subsequente é dedicada à modelação conceitual, na qual criamos o esquema conceitual da base de dados. Apresentamos a abordagem de modelação utilizada, identificamos e caracterizamos as entidades, relacionamentos e atributos envolvidos no sistema.

No estado atual do projeto, todas as etapas até a modelação conceitual foram concluídas com êxito.

**Área de Aplicação:** Desenvolvimento e implementação de Sistemas de Bases de Dados

**Palavras-Chave:** Bases de Dados, Bases de Dados Relacionais, Bases de Dados Não Relacionais, Análise de Requisitos, Modelo Lógico, Modelo Conceptual, Entidades, Atributos, Normalização, MySQL Workbench, SQL.

# Índice

Resumo	iii
Índice	iv
Índice de Figuras	vi
Índice de Tabelas	viii
1. Introdução	1
1.1. Contextualização	1
1.2. Fundamentação	2
1.3. Motivação e Objetivos	2
1.4. Viabilidade	3
1.5. Recursos a Utilizar	3
1.6. Equipa de Trabalho	4
1.7. Plano de execução	4
2. Definições de Requisitos	5
2.1. Método de levantamento e de análise de requisitos adotado	5
2.2. Organização dos requisitos levantados	6
2.2.1 Requisitos de descrição	6
2.2.2 Requisitos de exploração	8
2.2.3 Requisitos de controlo	9
2.3. Análise e validação geral dos requisitos	9
3. Modelação Concetual	10
3.1. Apresentação da abordagem de modelação realizada	10
3.2. Identificação e caracterização das entidades	10
3.3. Identificação e caracterização dos relacionamentos	12
3.3.1 Relação Evento <-> Bilhete	12
3.3.2 Relação Evento <-> Cliente	13
3.3.3 Relação Evento <-> Equipa	13
3.3.4 Relação Evento <-> Funcionário	14
3.3.5 Relação Evento <-> Headliner	14
3.3.6 Relação Evento <-> Localização	15
3.3.7 Relação Bilhete <-> Cliente	15
3.3.8 Relação Equipa <-> Funcionário	16
3.3.9 Relação Funcionário <-> Bilhete	16

3.4. Identificação e caracterização das associações dos atributos com as entidades e relacionamentos	17
3.4.1 Entidade – Evento	17
3.4.2 Entidade – Equipa	17
3.4.3 Entidade – Funcionário	18
3.4.4 Entidade – Bilhete	18
3.4.5 Entidade – Cliente	19
3.4.6 Entidade – Localização	19
3.4.7 Entidade – Headliner	20
3.4.8 Relacionamento – Evento Cliente	20
3.5. Apresentação e explicação do diagrama ER produzido	21
4. Modelação Lógica	22
4.1. Construção e validação do modelo de dados lógico	22
4.2. Normalização de Dados	24
4.3. Apresentação e explicação do modelo lógico produzido	25
4.4. Validação do modelo com interrogações do utilizador	26
5. Implementação Física	29
5.1. Tradução do esquema lógico para o sistema de gestão de bases de dados escolhido	29
5.2. Tradução das interrogações do utilizador para SQL	31
5.3. Definição e caracterização das vistas de utilização em SQL	33
5.4. Definição e caracterização de alguns utilizadores em SQL	37
5.5. Cálculo do espaço da base de dados (inicial e taxa de crescimento anual)	39
5.6. Indexação do Sistema de Dados	42
5.7. Procedimentos Implementados	43
5.8. Plano de segurança e recuperação de dados	46
5.9. Dashboard da Base de Dados (MS Power BI)	47
6. Conclusões e Trabalho Futuro	48
Referências	49

## **Anexos**

### **I. Anexo 1**

**Error!**

**Bookmark not defined.**

# Índice de Figuras

Figura 1 (Diagrama de GANTT)	4
Figura 2 (Rel. Evento_Bilhete)	12
Figura 3 (Rel. Evento_Cliente)	13
Figura 4 (Rel. Evento_Equipa)	13
Figura 5 (Rel. Evento_Funcionário)	14
Figura 6 (Rel. Evento_Headliner)	14
Figura 7 (Rel. Evento_Localização)	15
Figura 8 (Rel. Bilhete_Cliente)	15
Figura 9 (Rel. Equipa_Funcionário)	16
Figura 10 (Rel. Funcionário_Bilhete)	16
Figura 11 (Modelo Conceptual)	21
Figura 12 - Exemplo de Multivalorado	22
Figura 13 - Exemplo de Relação N:N	23
Figura 14 - Exemplo de Relação 1:N	23
Figura 15 - Modelo Lógico	25
Figura 16 - RE08 ListaClientes	26
Figura 17 - RE11 ListaEventosSucessoVendas	27
Figura 18 - RE16 ListaEventosClassificação	28
Figura 19 – Tabela Entidade SQL	29
Figura 20 - Tabela Evento avaliado Cliente SQL	30
Figura 21 - Tabela patrocinadores SQL	30
Figura 22 - Query RE04	31
Figura 23 - Output Query RE04	31
Figura 24 - Query RE06	32
Figura 25 - Output Query RE06	32
Figura 26 - Query RE14	33
Figura 27 – Output Query RE14	33
Figura 28 - Vista Cliente	34
Figura 29 - Output da Vista Cliente	34
Figura 30 - Vista Funcionário	35
Figura 31 - Output da Vista Funcionário	35

Figura 32 - Vista Bilhete	36
Figura 33 - Output da Vista Bilhete	36
Figura 34 - Criar Utilizadores SQL	37
Figura 35 – Garantir Permissões Utilizadores SQL	38
Figura 36 - Retirar Permissões Utilizadores SQL	38
Figura 37 - Projeção do Custo da Base de Dados da Eventure	41
Figura 38 - Query Sucesso Eventos	42
Figura 39 - Resultado Índices	42
Figura 40 - Comandos Criar Index	42
Figura 41 - Procedimentos dos Requisitos de Exploração	43
Figura 42 - Procedimentos de Adição na Base de Dados	44
Figura 43 - Exemplo de Procedimento de Adição na Base de dados	44
Figura 44 - Procedimentos de Remoção na Base de Dados	45
Figura 45 - Exemplo de Procedimento de Remoção na Base de dados	45
Figura 46 - Demonstração do mysqldump	46
Figura 47 - Dashboard Finanças	47

## Índice de Tabelas

Tabela 1 - Requisitos de Descrição CEO	6
Tabela 2 - Requisitos de Descrição Relações Públicas	6
Tabela 3 - Requisitos de Descrição Finanças	7
Tabela 4 - Requisitos de Descrição Marketing	7
Tabela 5 - Requisitos de Exploração CEO	8
Tabela 6 - Requisitos de Exploração Relações Públicas	8
Tabela 7 - Requisitos de Exploração Finanças	8
Tabela 8 - Requisitos de Exploração Marketing	8
Tabela 9 - Requisitos de Controlo	9
Tabela 11 (Entidade - Evento)	17
Tabela 12 (Entidade - Equipa)	17
Tabela 13 (Entidade - Funcionário)	18
Tabela 14 (Entidade - Bilhete)	18
Tabela 15 (Entidade - Cliente)	19
Tabela 16 (Entidade – Localização)	19
Tabela 17 (Entidade - Headliner)	20
Tabela 18 (Relacionamento – Evento_Cliente)	20
Tabela 36 - Custo de Memória SQL	39
Tabela 37 - Custa Tabelas Eventure	40



# 1. Introdução

## 1.1. Contextualização

A **Eventure** é uma empresa, fundada em 2018 por **Vítor Fonseca**, dedicada à Gestão e Divulgação de Eventos e fica situada na freguesia de Algés, em Lisboa. Vítor Fonseca tinha como sonho a criação e organização de um festival de música que juntaria os clássicos dos anos 80 e 90 com as modas dos 2000 e com isto deu-se início ao “**Gira o Disco e Toca o Mesmo**”.

Imediatamente um sucesso, o festival e a empresa foram crescendo, mas, em 2022, a **Câmara Municipal de Lisboa** com intenções de dinamizar culturalmente a cidade decidiu fazer uma parceria com a empresa que assim ficaria encarregue de alguns dos eventos realizados por toda a cidade. Com o acréscimo de responsabilidade, foi decidido que deveria ser implementada uma **base de dados** para ajudar a gerir o rápido crescimento da empresa.

A **Eventure**, com o aumento de responsabilidade optou também por criar 3 departamentos, divididos por: Finanças, Marketing e Relações-Públicas. No total a empresa emprega 11 funcionários principais que estão divididos pelos 3 departamentos. Mas os que mais ganham destaque são os diretores: Daniel Tardio, no departamento de **Finanças**, Vera Lama, no departamento de **Marketing** e Evaldo Perfeito, no departamento de **Relações Públicas**. E por alguns funcionários terceirizados.

Vítor Fonseca gostava também de ter uma proximidade maior com os seus clientes pois o objetivo da criação da empresa era a satisfação de quem visitava o seu festival e tem como objetivo que a base de dados a ser implementada o ajudasse nisso.

A empresa especializou-se e destacou-se na organização de festivais de música, mas não é o único género de eventos que organiza e agora com a honra de organizar os eventos fornecidos pela Câmara Municipal de Lisboa pretende expandir o tipo dos seus eventos.

## 1.2. Fundamentação

Devido ao crescimento acelerado da empresa, a **Eventure**, optou pela criação de uma base de dados em consequência do número elevado de eventos e da maior dificuldade na gestão dos tais. Com esta base de dados, pretendeu-se conceber um sistema de **profiling** de clientes que possibilita o máximo de informação sobre os visitantes dos eventos organizados e assim adaptar estratégias de marketing específicas aos públicos-alvo que se pretende atingir. Idealizou-se também um sistema de **feedback** dos eventos para opiniões dos visitantes serem analisadas e avaliadas e, desta forma, prestar o melhor serviço possível aos clientes da empresa. Com a **informatização** da Eventure a segurança também será aumentada devido ao uso de bilhetes eletrónicos, possibilitando assim um melhor controlo de entradas e saídas dos recintos onde ocorrem os eventos.

A **base de dados** deverá alocar os eventos já realizados e eventos futuros, em cada evento ficará armazenado a data, a localização, o número de visitantes, os dados dos visitantes e, se houver, o seu feedback do evento. Deverá também guardar todas as informações da gestão financeira da empresa como as receitas de bilheteria, de patrocínios e as despesas dos artistas.

## 1.3. Motivação e Objetivos

Como foi referido anteriormente, a principal motivação para o desenvolvimento desta base de dados foi facilitar a gestão do número elevado de eventos que a empresa tinha a realizar, com isso, o armazenamento de dados ajuda a compreender como melhorar tanto na organização como na qualidade em geral dos eventos. E acreditamos que as mecanizações de alguns procedimentos na gestão vão facilitar todos os processos da empresa.

A base de dados teve também na sua criação outros objetivos definidos, dos quais:

- Gestão e **registos da informação** dos eventos realizados e por realizar (data do evento, localização, número de visitantes...)
- Gestão mais eficiente das **finanças** da empresa
- Controlo das **entradas e saídas dos recintos** a partir dos bilhetes eletrónicos
- Criação de um sistema de **profiling de clientes**
- Criação de um sistema **feedback** dos visitantes
- Melhorar a **qualidade dos serviços**

## 1.4. Viabilidade

Com esta forma mais eficiente e moderna de gerir e registar os eventos realizados, a Eventure calcula que será capaz de:

- **Escalar** de forma estável e organizada a gestão e armazenamento de informações de cada um dos eventos a realizar.
- Controlar a **segurança** dos recintos e também dos próprios compradores evitando revendas de bilhetes.
- **Conhecer melhor os seus clientes** a partir do sistema de **profilling** e, a partir disso, adotar medidas para aumentar a satisfação do cliente e/ou estratégias de marketing para atrair mais compradores.
- **Reduzir custos** e agilizar a gestão financeira da empresa.
- **Melhorar a qualidade e acessibilidade** dos seus eventos a partir de um sistema de feedback do comprador onde a perspetiva do visitante é avaliada e assim atender de melhor forma as necessidades dele.
- **Armazenar informações** sobre os seus eventos realizados para auxiliar na gestão dos mesmos.

## 1.5. Recursos a Utilizar

Nos **recursos materiais** a utilizar, foi adquirido 1 servidor e 1 sistema de gestão de bases de dados e o seu software mais a sede da **Eventure** onde se vai utilizar maioritariamente a base de dados.

Quanto aos **recursos humanos**, os funcionários da empresa que podem aceder à base de dados e os especialistas contratados serão necessários para o funcionamento correto da base de dados, neles o arquiteto e os engenheiros da base de dados.

## 1.6. Equipa de Trabalho

A equipa de trabalho reunida para o funcionamento e manutenção desta base de dados é composta pelo arquiteto e engenheiros de base de dados, no total 4, que servirão para o levantamento de requisitos e para a modelação e implementação do sistema.

Sobre o pessoal interno, os diretores de cada departamento da empresa e Vítor Fonseca, CEO da **Eventure**, terão como funções a elaboração de objetivos, a validação de serviços e a execução das estratégias de marketing.

Os clientes também terão o seu papel na resposta aos inquéritos de opinião e na validação de serviços.

## 1.7. Plano de execução

### Implementação do Sistema de Base de Dados

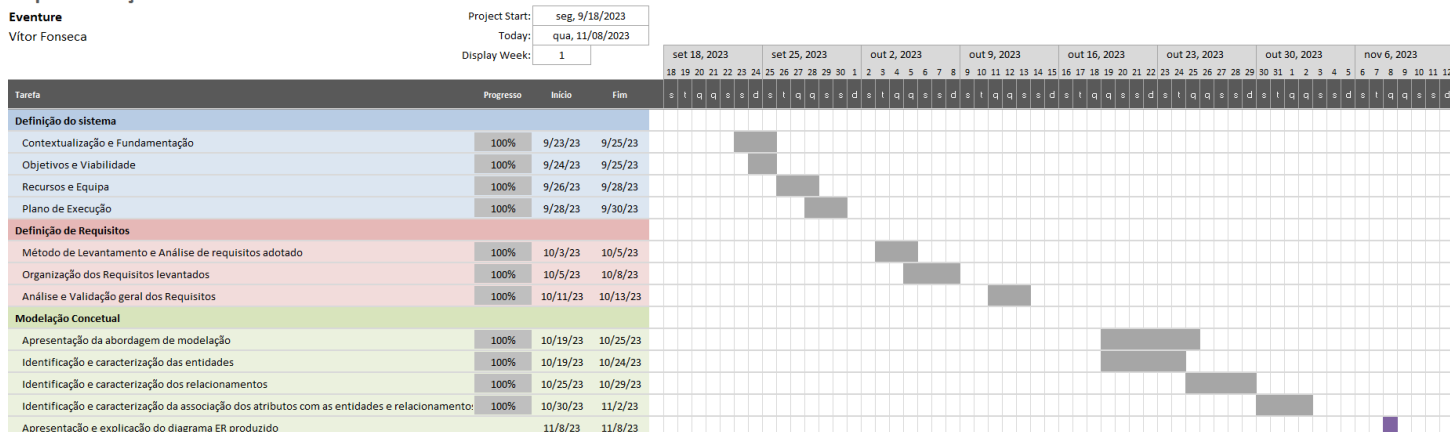


Figura 1 (Diagrama de GANTT)

Para uma execução do trabalho com mais eficiência, recorreremos ao diagrama de Gantt e dividimos as tarefas ao longo do tempo, proporcionando assim uma melhor gestão de tempo e de recursos humanos por parte da equipa de desenvolvimento da base de dados. O diagrama acima descreve o nosso percurso até à data atual.

## **2. Definições de Requisitos**

### **2.1. Método de levantamento e de análise de requisitos adotado**

Quanto ao levantamento de requisitos, decidiu-se fazer reuniões com cada departamento individualmente. Foram convocadas reuniões com Vítor Fonseca e cada departamento individualmente para discutir as necessidades de cada departamento em questão à base de dados.

Alguns dos requisitos também foram pedidos pelos responsáveis de cada departamento.

No departamento do Daniel Tardio (Finanças), ele preferiu escolher tudo o que seria necessário para a base de dados por motivos empíricos.

A Equipa de desenvolvimento também opinou sobre algumas características que a base de dados deveria ter visto estes terem bastante experiência na conceptualização deste tipo de sistemas.

As análises foram realizadas logo de seguida para agilizar os processos todos.

No tópico a seguir, consta o resultado dessas mesmas reuniões.

## 2.2. Organização dos requisitos levantados

### 2.2.1 Requisitos de descrição

CEO – Vítor Fonseca			
Número	Descrição	Objetivo	Fonte
Rd01	Todo o evento tem de estar registado tendo um nome e as datas de funcionamento	Facilitar a gestão geral da empresa	Vítor Fonseca
Rd02	Toda a Equipa tem de estar registada contendo o seu chefe e a função da mesma	Uma gestão mais eficiente dos recursos humanos disponíveis	Vítor Fonseca
Rd03	Cada funcionário tem de ter um registo contendo todas as informações referentes à sua identificação, desde morada, cargo, nome, NIF e data de nascimento e de admissão	Controlo detalhado dos recursos humanos para uma eficiência acrescida na hora de montar as equipas	Vítor Fonseca
Rd04	Cada bilhete tem de estar inserido no sistema junto com as suas descrições (Datas, lugar se necessário)	Controlar as entradas e saídas dos recintos e evitar a sua revenda	Vítor Fonseca
Rd05	Cada entidade na base de dados tem de conter um campo para observações	Eliminar futuros problemas técnicos com a manipulação da base de dados	Sugestão do Engenheiro Rúben
Rd06	Registos das localizações e das respetivas informações registos climatéricos (temperatura, vento e chuva)	Evitar mudanças de data dos eventos	Vítor Fonseca

Tabela 1 - Requisitos de Descrição CEO

Equipa de Relações Públicas			
Número	Descrição	Objetivo	Fonte
Rd07	A lotação máxima por evento	Dividir as vendas dos bilhetes pelas equipas	Reunião com a equipa
Rd08	Preço de venda de cada bilhete	Facilitar a venda de todos tipo de bilhetes	Evaldo Perfeito
Rd09	Cada cliente tem de ter um registo contendo nome, data de nascimento, morada e principalmente, os contactos (e-mail e telemóvel)	Aumentar a capacidade de persuasão, se possível, na hora de vender o bilhete a esse cliente	Reunião com a equipa
Rd10	Cada evento pode ter uma lista de patrocinadores	Aumentar o lucro do evento	Evaldo Perfeito

Tabela 2 - Requisitos de Descrição Relações Públicas

Equipa de Finanças			
Número	Descrição	Objetivo	Fonte
Rd11	Finanças gerais do evento (orçamento e custo)	Controlo da tesouraria	Daniel Tardio
Rd12	A necessidade de registar o cachê do headliner	Controlo da tesouraria	Daniel Tardio
Rd13	Cada funcionário tem de ter o seu salário registado	Projeções de despesas	Daniel Tardio

Tabela 3 - Requisitos de Descrição Finanças

Equipa de Marketing			
Número	Descrição	Objetivo	Fonte
Rd14	Todos os headliners registados têm de conter o seu chefe e os participantes	Facilitar a criação de estratégias personalizadas a cada evento	Vera Lama
Rd15	População de uma certa localização	Controlo de targeting do marketing	Vera Lama
Rd16	Será necessário um sistema de feedback contendo a opinião do cliente em relação a um certo evento, sendo esta opinião composta por uma classificação quantitativa, uma breve descrição de pontos positivos e negativos e a data da mesma	Melhoramento de estratégias de marketing anteriores e permitir um funcionamento melhor de futuros eventos	Vera Lama
Rd17	Cada evento tem de ter uma categoria (musical, culinária...)	Facilitar a criação de estratégias personalizadas a cada evento	Reunião com a equipa

Tabela 4 - Requisitos de Descrição Marketing

## 2.2.2 Requisitos de exploração

CEO – Vítor Fonseca			
Número	Descrição	Objetivo	Fonte
Re01	Quantidade de bilhetes	Estatística da empresa e análise de dados	Vítor Fonseca
Re02	Será necessário obter a quantidade de funcionários de uma equipa	Controlo dos recursos humanos de uma forma mais eficiente	Vítor Fonseca
Re03	É necessária uma Listagem do histórico dos eventos de uma localização	Evitar mudanças de data dos eventos	Vítor Fonseca
Re04	É importante uma lista de eventos nos próximos 30 dias	Um planeamento mais eficiente	Vítor Fonseca
Re05	Listagem das localizações pelos dados climáticos	Um planeamento mais eficiente	Vítor Fonseca
Re06	É relevante existir uma lista de membros por equipa	Um controlo detalhado dos recursos humanos	Sugestão do Arquiteto Bruno

Tabela 5 - Requisitos de Exploração CEO

Equipa de Relações Públicas			
Número	Descrição	Objetivo	Fonte
Re07	É necessário a disponibilização da lista de Patrocinadores	Saber quem patrocinou o evento	Reunião com a equipa
Re08	Listagem com os clientes (quantidade de aquisições e custo das mesmas)	Controlo de qualidade dos clientes	Evaldo Perfeito
Re09	Conseguir obter o histórico de um cliente	Controlo de qualidade dos clientes	Reunião com a equipa

Tabela 6 - Requisitos de Exploração Relações Públicas

Equipa de Finanças			
Número	Descrição	Objetivo	Fonte
Re10	Tem de ser permitido à equipa ter acesso às Finanças Gerais da empresa	Controlo da tesouraria	Daniel Tardio
Re11	É necessária uma ordenação dos eventos pelo seu sucesso a nível de vendas	Controlo da tesouraria	Daniel Tardio
Re12	É necessária uma ordenação dos eventos pela quantidade de bilhetes vendidas	Controlo da tesouraria	Daniel Tardio

Tabela 7 - Requisitos de Exploração Finanças

Equipa de Marketing			
Número	Descrição	Objetivo	Fonte
Re13	A base de dados tem de fornecer todas as vezes que o headliner tem participação com a empresa em si	Facilitar a criação de estratégias personalizadas a cada evento	Vera Lama
Re14	Tem de ser possível listar as opiniões de um cliente	Estudar potenciais pontos fulcrais dos eventos	Reunião com a equipa
Re15	Pontos Positivos/Negativos de um evento	Fazer um estudo de causa/consequência	Reunião com a equipa
Re16	Listagem dos eventos pelas avaliações quantitativas	Estudar potenciais pontos fulcrais dos eventos	Reunião com a equipa

Tabela 8 - Requisitos de Exploração Marketing



## 2.2.3 Requisitos de controlo

CEO – Vítor Fonseca			
Número	Descrição	Objetivo	Fonte
Rc01	O departamento de Marketing não tem acesso aos dados das equipas e funcionários, nem financeiros	Limitar o acesso indevido de informações	Vítor Fonseca
Rc02	O departamento de Relações Públicas tem só acesso aos dados dos Clientes, Headliners e patrocinadores	Limitar o acesso indevido de informações	Vítor Fonseca
Rc03	Os clientes podem alterar os próprios dados	Permitir os clientes usarem a base de dados	Vítor Fonseca
Rc04	Só a equipa de Finanças e o CEO podem tudo da base de dados	Para evitar manipulações indevidas	Sugestão do Arquiteto Bruno
Rc05	Todos os departamentos podem adicionar dados à base de dados desde que tenham acesso	Para evitar manipulações indevidas	Vítor Fonseca

Tabela 9 - Requisitos de Controlo

## 2.3. Análise e validação geral dos requisitos

O processo de análise dos requisitos foi segmentado em três etapas distintas: requisitos de descrição, requisitos de exploração, requisitos de controlo. Cada uma dessas etapas foi examinada em reuniões de grupo, onde os membros contribuíram com suas perspectivas e sugestões.

A primeira fase, que abordou os requisitos de descrição, o foco foi identificar os principais tópicos necessários para o funcionamento da **Eventure**. As descrições que poderiam dar algum benefício à empresa e aos seus respetivos departamentos.

Na segunda fase, dedicada aos requisitos de exploração, foram mapeadas as informações críticas que deveriam ser obtidas do sistema para ajudarem no objetivo final da **Eventure**.

Finalmente, na fase de requisitos de controlo, estabelecemos as restrições da base de dados.

## 3. Modelação Concetual

### 3.1. Apresentação da abordagem de modelação realizada

O Diagrama de Entidade-Relação (ER) é uma ferramenta utilizada para modelar e projetar bases de dados. Permite representar graficamente as entidades, relações e atributos envolvidos num sistema de gestão de dados.

Esta ferramenta ajuda a visualizar de forma clara e objetiva a estrutura da base de dados e as relações entre as entidades, facilitando a compreensão do sistema e ajudando a identificar possíveis problemas ou erros antes da implementação. Além disso, permite uma comunicação eficaz entre os programadores e os utilizadores do sistema, garantindo que todos estejam alinhados com o objetivo e a estrutura da base de dados.

### 3.2. Identificação e caracterização das entidades

- **Evento**

A entidade *Evento* é responsável pela identificação de tudo o que engloba a parte descritiva do evento, contendo os atributos financeiros e os atributos relativos à descrição direta do evento, como datas, patrocinadores, etc. como requerido pela empresa.

- **Equipa**

A entidade *Equipa* tem como objetivo ter na base de dados o registo das equipas associadas a funções logísticas para um melhor controlo da parte administrativa.

- **Funcionário**

A entidade *Funcionário* é uma parte fundamental da base de dados pois esta permite ter o registo aprofundado de todos os recursos humanos disponíveis pela empresa e com isto, é previsto uma facilidade aumentada da gestão de mão de obra humana pelas necessidades dos eventos.

- **Bilhete**

A entidade *Bilhete* é responsável pela identificação de cada bilhete, a decisão de tornar o bilhete uma entidade foi fundamental para facilitar a caracterização de cada bilhete, tornando único e mais facilmente controlável. Este contém os atributos associados às datas permitidas, preço e lugar caso seja necessário.

- **Cliente**

A entidade *Cliente* satisfaz a necessidade do registo de cada cliente, e essencialmente, a capacidade de armazenarmos na base de dados todas as opiniões que servirão para estudos futuros a favor da melhoria da empresa e dos eventos a si associados (estas estão associadas à relação entre o cliente e o evento, para facilitar a associação entre os dois e garantir que o cliente só tem opinião caso participe no evento).

- **Localização**

A entidade *Localização* tem como objetivo ter dados para controlo de eventos futuros, como dados climatéricos (evitar marcar um concerto ao ar livre quando, na média, chove todos os dias) e dados como a quantidade de população.

- **Headliner**

A entidade *Headliner* acolhe todas as cabeças de cartaz e as respetivas equipas/cachês.

### 3.3. Identificação e caracterização dos relacionamentos

#### 3.3.1 Relação Evento <-> Bilhete

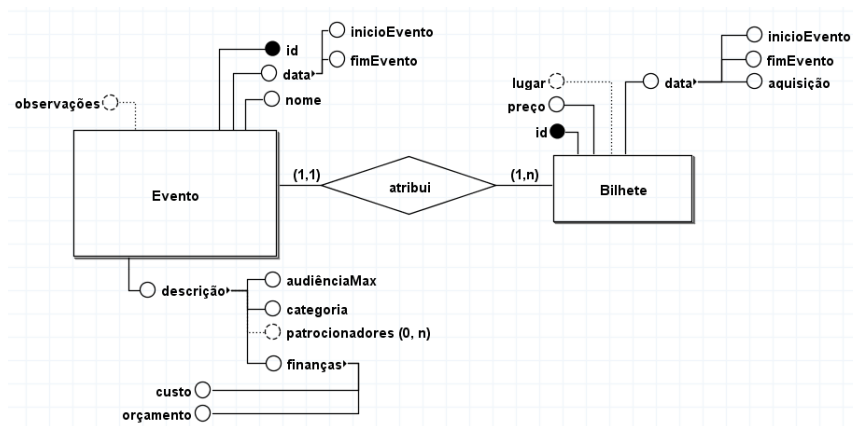


Figura 2 (Rel. Evento\_Bilhete)

Cada Evento atribui um ou um conjunto de bilhetes e cada bilhete está associado a um e só um evento.

**Relacionamento:** Evento *atribui* Bilhete;

**Cardinalidade:** Evento (1,n) - Bilhete (1,1);

**Atributos:** Este relacionamento não possui atributos.

### 3.3.2 Relação Evento <-> Cliente

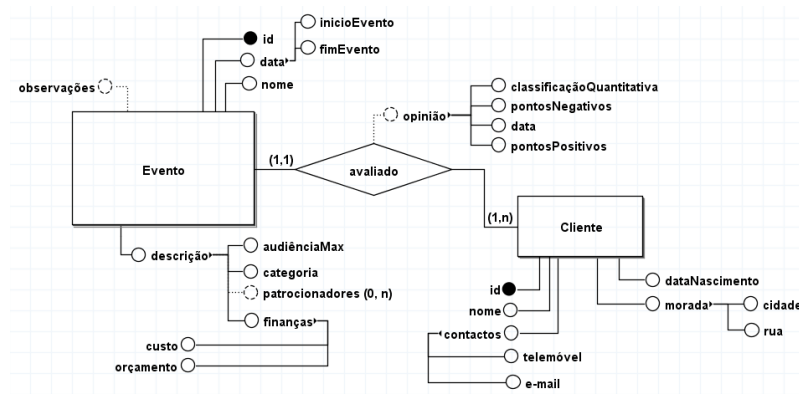


Figura 3 (Rel. Evento\_Cliente)

Cada Evento pode ser ou não, avaliado por um cliente e cada cliente pode ou não, avaliar o evento.

**Relacionamento:** Evento *avaliado* Bilhete;

**Cardinalidade:** Evento (0,n) - Bilhete (0,n);

**Atributos:** Pode existir ou não opinião, caso exista, esta contém uma classificação quantitativas, uma breve descrição dos pontos positivos e dos negativos, e estará associado a data desta avaliação.

### 3.3.3 Relação Evento <-> Equipa

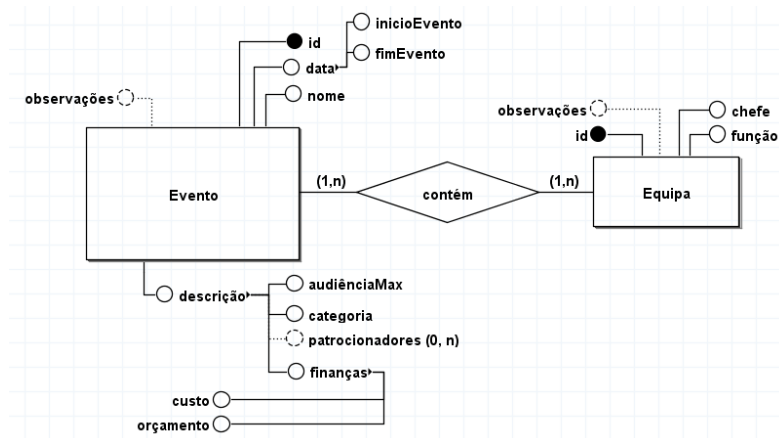


Figura 4 (Rel. Evento\_Equipa)

Cada Evento contém um ou um conjunto de equipas e cada equipa pode pertencer a um ou a múltiplos eventos.

**Relacionamento:** Evento *contém* Equipa;

**Cardinalidade:** Evento (1,n) - Equipa (1,n);

**Atributos:** Este relacionamento não possui atributos.

### 3.3.4 Relação Evento <-> Funcionário

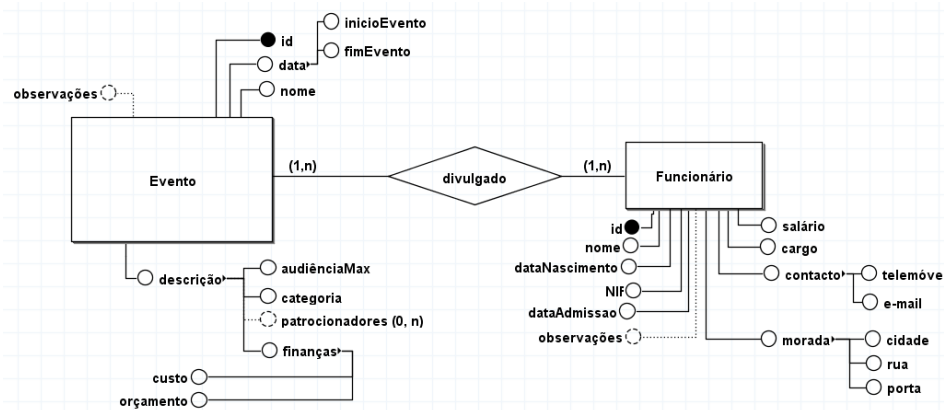


Figura 5 (Rel. Evento\_Funcionário)

Cada Evento é divulgado por um ou um conjunto de funcionários e cada funcionário pode divulgar um ou múltiplos eventos.

**Relacionamento:** Evento *divulgado* Funcionário;

**Cardinalidade:** Evento (1,n) - Funcionário (1,n);

**Atributos:** Este relacionamento não possui atributos.

### 3.3.5 Relação Evento <-> Headliner

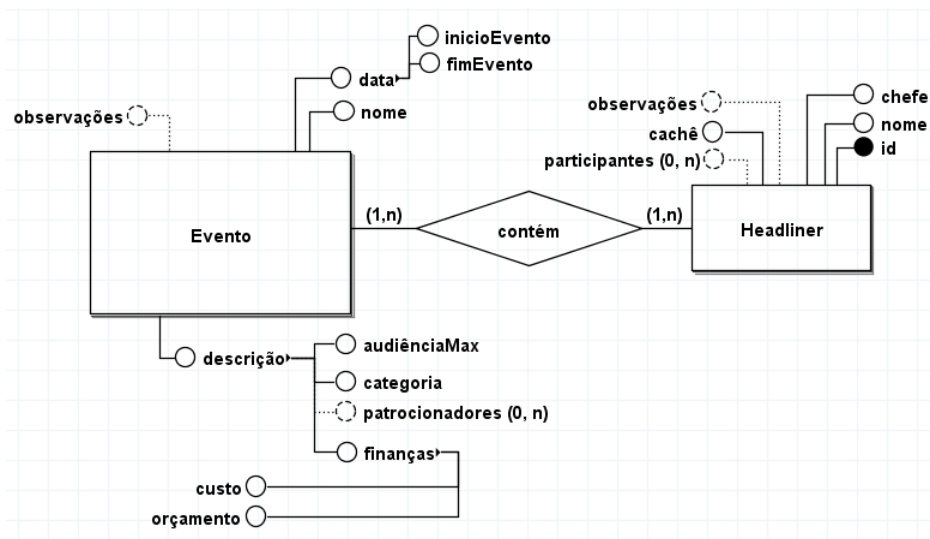


Figura 6 (Rel. Evento\_Headliner)

Cada Evento contém um ou um conjunto de Headliners e cada Headliner pode pertencer a um ou múltiplos eventos.

**Relacionamento:** Evento *contém* Headliner;

**Cardinalidade:** Evento (1,n) – Headliner (1,n);

**Atributos:** Este relacionamento não possui atributos.

### 3.3.6 Relação Evento <-> Localização

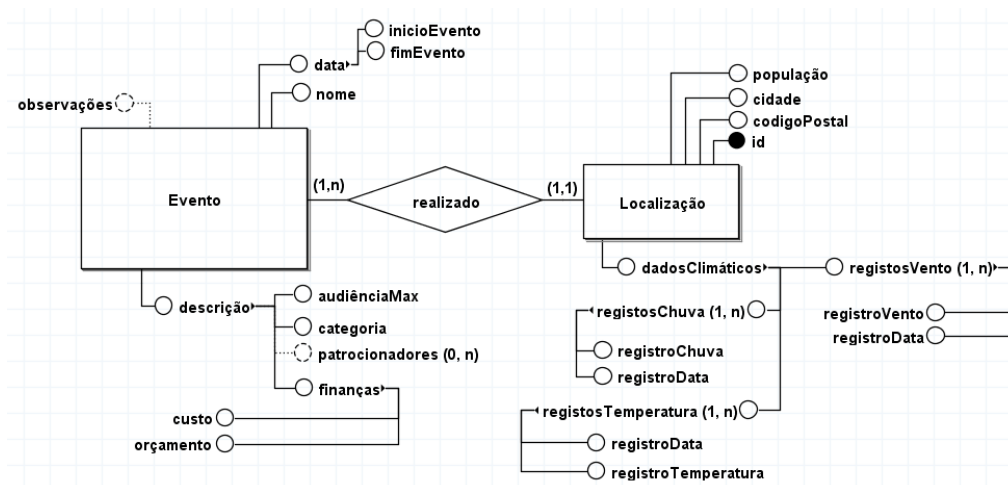


Figura 7 (Rel. Evento\_Localização)

Cada Evento contém uma e só uma localização e cada localização pode permitir decorrer vários eventos em simultâneo.

**Relacionamento:** Evento *realizado* Localização;

**Cardinalidade:** Evento (1,1) - Localização (1,n);

**Atributos:** Este relacionamento não possui atributos.

### 3.3.7 Relação Bilhete <-> Cliente

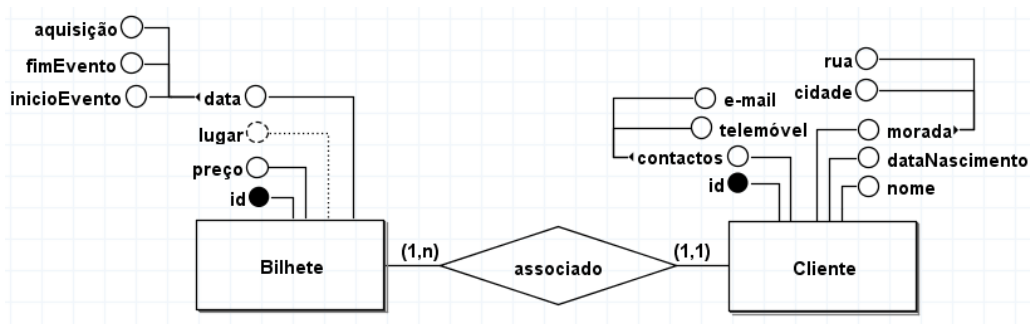


Figura 8 (Rel. Bilhete\_Cliente)

Cada Bilhete está associado a um e só um cliente e cada cliente pode ter comprado um ou mais bilhetes.

**Relacionamento:** Bilhete *associado* Cliente;

**Cardinalidade:** Bilhete (1,1) - Cliente (1,n);

**Atributos:** Este relacionamento não possui atributos.

### 3.3.8 Relação Equipa <-> Funcionário

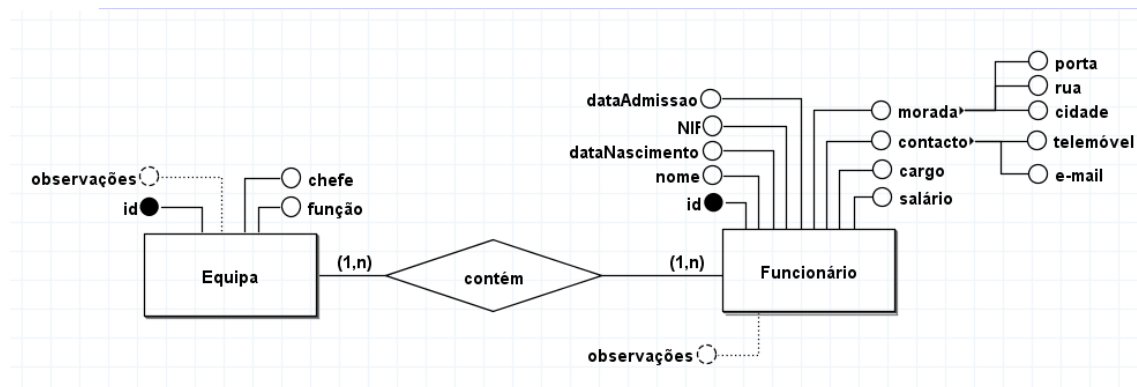


Figura 9 (Rel. Equipa\_Funcionário)

Cada Equipa contém um ou um conjunto de funcionários e cada funcionário pode trabalhar em uma ou mais equipas.

**Relacionamento:** Equipa *contém* Funcionário;

**Cardinalidade:** Equipa (1,n) - Funcionário (1,n);

**Atributos:** Este relacionamento não possui atributos.

### 3.3.9 Relação Funcionário <-> Bilhete

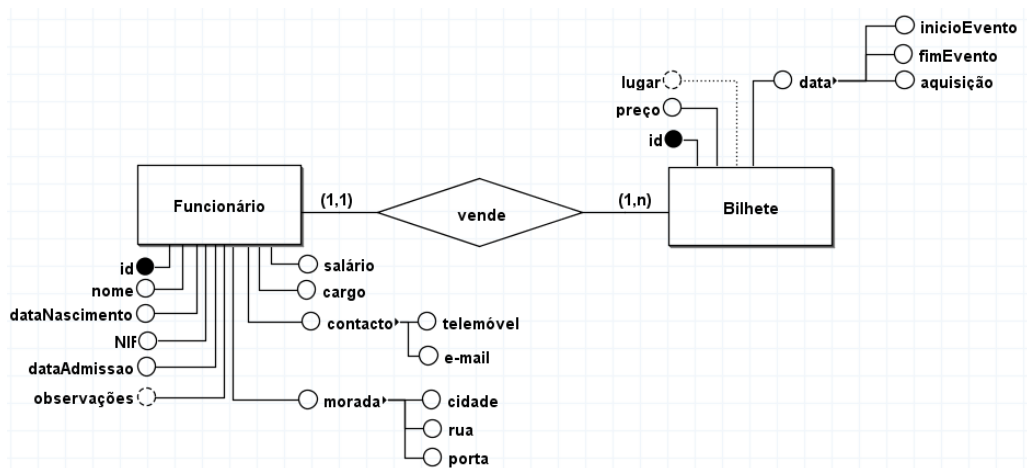


Figura 10 (Rel. Funcionário\_Bilhete)

Cada Funcionário pode vender um ou um conjunto de bilhetes e cada bilhete pode ter sido vendido por um e só um funcionário.

**Relacionamento:** Funcionário *vende* Bilhete;

**Cardinalidade:** Funcionário (1,n) - Bilhete (1,1);

**Atributos:** Este relacionamento não possui atributos



### 3.4. Identificação e caracterização das associações dos atributos com as entidades e relacionamentos

#### 3.4.1 Entidade – Evento

Atributo	Descrição	Domínio/Tamanho	Nulo	Multi	Chave primária	Exemplo
id	Identificador do evento	INT	N	N	S	1
data	<b>Composto</b>		N	N	N	
-inicioEvento	Data do início do Evento	DATE	N	N	N	1992-12-01
-fimEvento	Data do fim do evento	DATE	N	N	N	1992-12-01
nome	Nome do evento	VARCHAR(255)	N	N	N	Ruben Goncalo Araujo Silva
observações	Algum tipo de dado atípico	VARCHAR(2047)	S	N	N	“Impossibilidade de rampas de acesso”
descrição	<b>Composto</b>		N	N	N	
-audienciaMax	Número. máximo de visitantes	INT	N	N	N	30000
-categoria	Tipo de evento	VARCHAR(255)	N	N	N	Festival Musical
-patrocinadores	Lista de patrocinadores	VARCHAR(255)	S	S	N	Ruben Silva, Pedro Gilberto....
-finanças	<b>Composto</b>		N	N	N	
--custo	Custo do evento	INT	N	N	N	13500
--orçamento	Orçamento do evento	INT	N	N	N	15000

Tabela 10 (Entidade - Evento)

#### 3.4.2 Entidade – Equipa

Atributo	Descrição	Domínio/Tamanho	Nulo	Multi	Chave primária	Exemplo
id	Identificador da Equipa	INT	N	N	S	1
observações	Algum tipo de dado atípico	VARCHAR(2047)	S	N	N	“Equipa precisa de um eletricista”
chefe	Responsável pela equipa	VARCHAR(255)	N	N	N	Pedro Alberto
função	Função da equipa	VARCHAR(255)	N	N	N	Equipa de Limpeza

Tabela 11 (Entidade - Equipa)

### 3.4.3 Entidade – Funcionário

Atributo	Descrição	Domínio/Tamanho	Nulo	Multi	Chave primária	Exemplo
Id	Identificador do Funcionário	INT	N	N	S	1
nome	Nome do Funcionário	VARCHAR(255)	N	N	N	Bruno Rolo Neiva
dataNascimento	Data de Nascimento	DATE	N	N	N	2002-01-01
NIF	Número de identificação fiscal	INT	N	N	N	87463526
dataAdmissão	Data admissão a funcionário	DATE	N	N	N	2022-10-06
observações	Algum tipo de dado atípico	VARCHAR(2047)	S	N	N	“Grávida”
salário	Salário do funcionário	INT	N	N	N	1500
cargo	Cargo na empresa	VARCHAR(255)	N	N	N	Diretor de Marketing
contacto	<b>Composto</b>		N	N	N	
-telemóvel	Número de telemóvel	INT	N	N	N	917755897
-e-mail	Endereço e-mail	VARCHAR(255)	N	N	N	curioso@gmail.com
morada	<b>Composto</b>		N	N	N	
-cidade	Nome da cidade	VARCHAR(255)	N	N	N	Lisboa
-rua	Nome da rua	VARCHAR(255)	N	N	N	Rua dos Dados
-porta	Número da porta	INT	N	N	N	30

Tabela 12 (Entidade - Funcionário)

### 3.4.4 Entidade – Bilhete

Atributo	Descrição	Domínio/Tamanho	Nulo	Multi	Chave primária	Exemplo
Id	Identificador do Bilhete	INT	N	N	S	1
preço	Preço do Bilhete	INT	N	N	N	20
lugar	Número do lugar do Bilhete	INT	S	N	N	25
data	<b>Composto</b>		N	N	N	
-inícioEvento	Data do início do evento	DATE	N	N	N	2022-10-06
-fimEvento	Data do fim do evento	DATE	N	N	N	2022-10-08
-aquisição	Data da compra do bilhete	DATE	N	N	N	2022-09-20

Tabela 13 (Entidade - Bilhete)

### 3.4.5 Entidade – Cliente

Atributo	Descrição	Domínio/Tamanho	Nulo	Multi	Chave primária	Exemplo
id	Identificador do Cliente	INT	N	N	S	1
nome	Nome do Cliente	VARCHAR(255)	N	N	N	Filipe Lopes da Silva
dataNascimento	Data de nascimento do cliente	DATE	N	N	N	1975-10-25
contactos	<b>Composto</b>		N	N	N	
-telemóvel	Número do telemóvel	INT	N	N	N	915778899
-e-mail	Endereço e-mail	VARCHAR(255)	N	N	N	antoniobd@gmail.com
morada	<b>Composto</b>		N	N	N	
-cidade	Nome da cidade	VARCHAR(255)	N	N	N	Porto
-rua	Nome da rua	VARCHAR(255)	N	N	N	Rua da Base

Tabela 14 (Entidade - Cliente)

### 3.4.6 Entidade – Localização

Atributo	Descrição	Domínio/Tamanho	Nulo	Multi	Chave primária	Exemplo
id	Identificador da localização	INT	N	N	S	1
codigoPostal	Código Postal	VARCHAR(255)	N	N	N	4760
cidade	Cidade	VARCHAR(255)	N	N	N	Vila Nova Famalicão
população	Número populacional	INT	S	S	N	1400000
dadosClimáticos	<b>Composto</b>		N	N	N	
-registosTemperatura	<b>Composto</b>		N	S	N	
--registroTemperatura	Temperatura Local	INT	N	N	N	17,18,14,22
--registroData	Data do registro da Temperatura	DATE	N	N	N	2023-12-01
-registosVento	<b>Composto</b>		N	S	N	
--registroVento	Velocidade do Vento	INT	N	N	N	7,0,3,12
--registroData	Data do registro do Vento	DATE	N	N	N	2023-12-02
-registosChuva	<b>Composto</b>		N	S	N	
--registroChuva	Percentagem de precipitação	INT	N	N	N	95,34,21,99
--registroData	Data do registro da Chuva	DATE	N	N	N	2023-12-03

Tabela 15 (Entidade – Localização)

### 3.4.7 Entidade – Headliner

Atributo	Descrição	Domínio/Tamanho	Nulo	Multi	Chave primária	Exemplo
Id	Identificador do Headliner	INT	N	N	S	1
Nome	Nome do Headliner	VARCHAR(255)	N	N	N	Jacques Berman Webster II
Chefe	Chefe do Headliner	VARCHAR(255)	N	N	N	Jacques Berman Webster II
Participantes	Grupo de participantes	VARCHAR(255)	S	S	N	Travis, Kid Scott, Jorge Mendes
Cachê	Custo do Headliner	INT	N	N	N	10000
Observações	Algum tipo de dado atípico	VARCHAR(2047)	S	N	N	“É preciso pizza antes do concerto”

Tabela 16 (Entidade - Headliner)

### 3.4.8 Relacionamento – Evento Cliente

Atributo	Descrição	Domínio/Tamanho	Nulo	Multi	Chave primária	Exemplo
opinião	Composto		S	N	N	
-classificaçãoQuantitativa	Nota de 0 a 10	INT	N	N	N	7
-pontosPositivos	Descri. pontos positivos	VARCHAR(2047)	N	N	N	“Gostei do Travis Scott e da bifana”
-pontosNegativos	Descri. pontos Negativos	VARCHAR(2047)	N	N	N	“Não gostei da fila para entrar”
-data	Data do feedback	DATE	N	N	N	2023-11-12

Tabela 17 (Relacionamento – Evento\_Cliente)

### 3.5. Apresentação e explicação do diagrama ER produzido

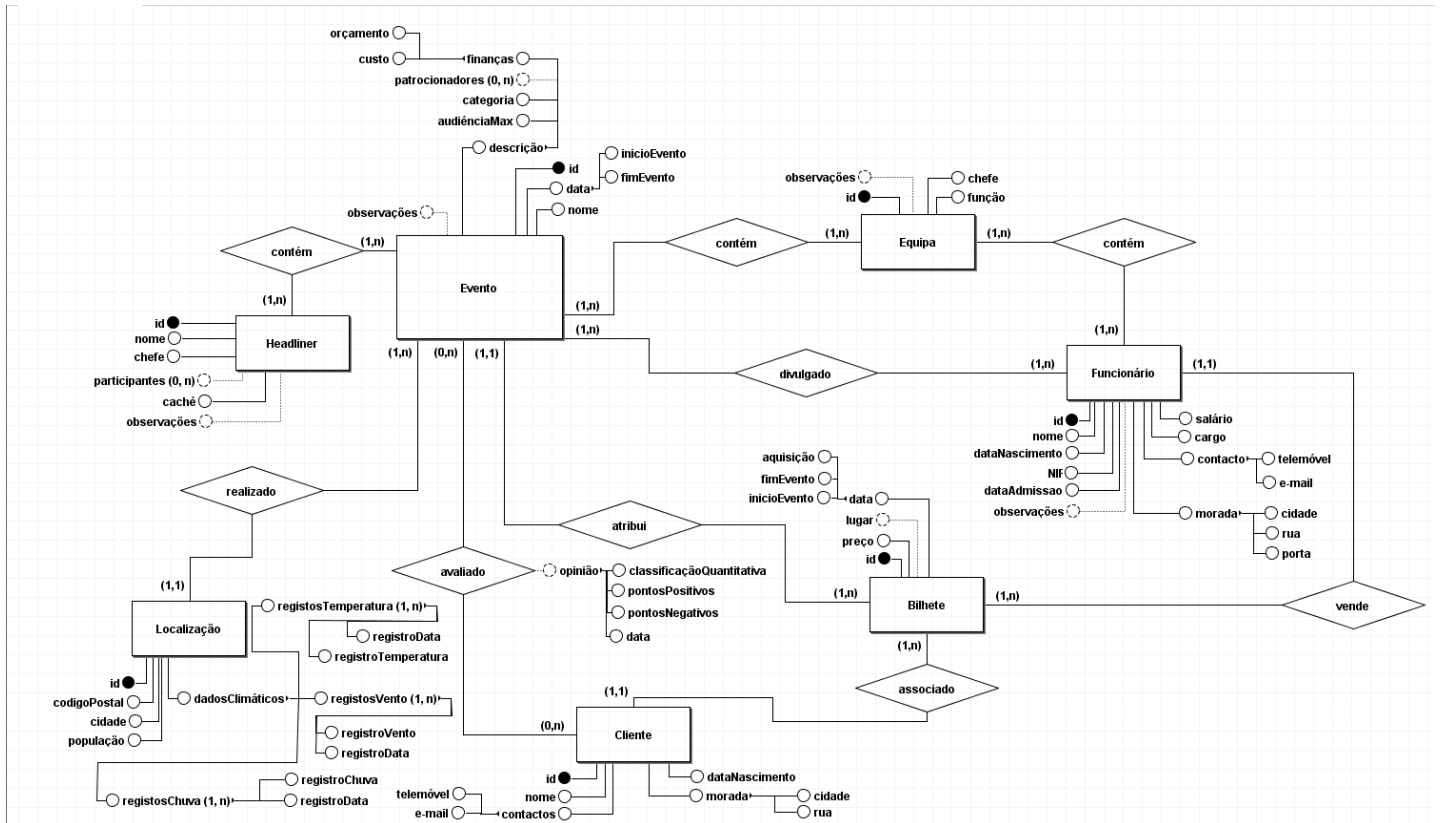


Figura 11 (Modelo Conceptual)

Após analisado profundamente e validado pela equipa de desenvolvimento, chegamos a este modelo conceptual da base de dados que vai de acordo com os requisitos descritivos pedidos pela equipa de desenvolvimento, cada ponto apresentado na figura 11, já foi abordado em tópicos anteriores.

## 4. Modelação Lógica

### 4.1. Construção e validação do modelo de dados lógico

Para a construção do nosso modelo lógico baseamo-nos no modelo conceptual e nas tabelas dos atributos definidos anteriormente e as seguintes noções sobre o modelo lógico:

- **Multivalorados**

Todos os multivalorados ganharam uma tabela nova, contendo como Primary Key um duplo da Foreign Key a que este pertence e o dado em si.

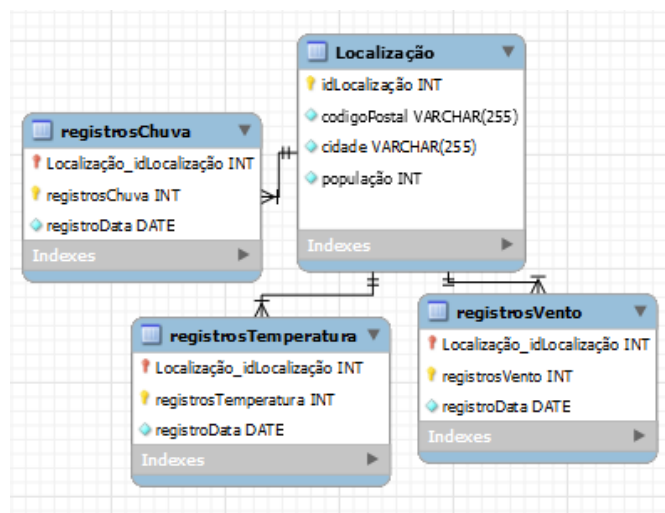


Figura 12 - Exemplo de Multivalorado

- **N:N**

Para transpormos este tipo de relação para o modelo lógico foi necessário criar uma tabela extra que fique no meio da relação entre as duas entidades, esta nova tabela permite 2 relações de 1:N. Com isto, é fundamental que esta relação tenha como Primary Key o duplo entre as 2 Foreign Keys.

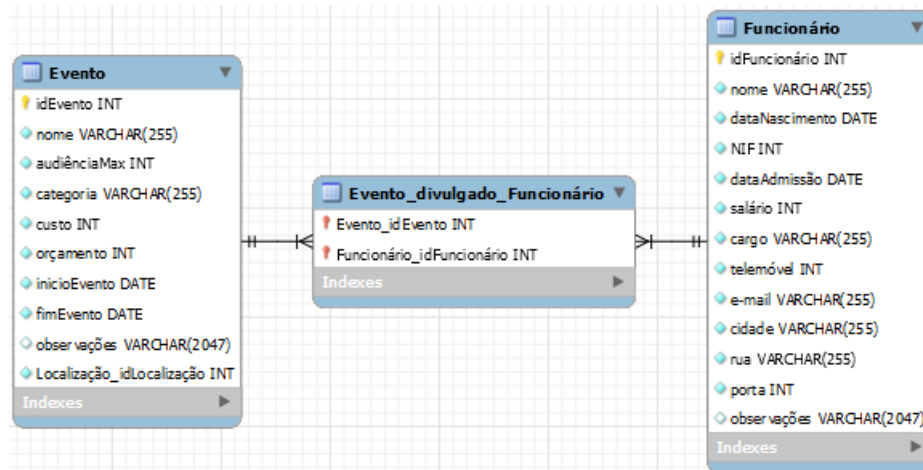


Figura 13 - Exemplo de Relação N:N

- **1:N**

Em relações de 1:N, simplesmente foi adicionado uma Foreign Key à tabela da entidade que necessita da informação. Como podemos ver na figura 14, um evento pode ser realizado em uma localização, e essa localização pode ter N eventos. Logo, a chave estrangeira da localização é entregue à tabela do evento

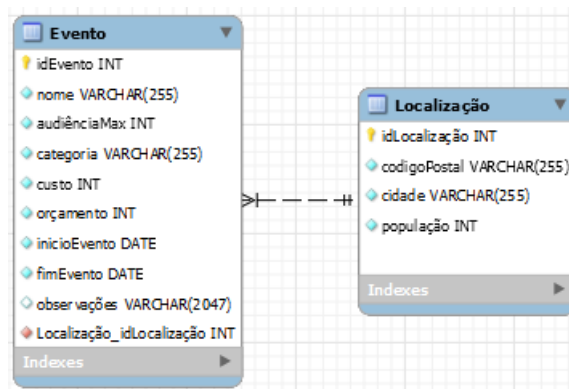


Figura 14 - Exemplo de Relação 1:N

## 4.2. Normalização de Dados

A normalização é um processo crucial para garantir que a base de dados esteja estruturada, otimizada e evite redundâncias nos dados.

É um processo progressivo, que depende de uma série de etapas, em que cada uma delas corresponde a uma forma normal específica – um critério concreto de validação - que vai transformando o esquema da base de dados sucessivamente em direção a um estado mais robusto e menos vulnerável. Para explicar sucintamente algumas das boas práticas de normalização que tentamos seguir na criação desta base de dados vamos nos focar nas 3 principais regras de normalização (1FN, 2FN, 3FN, 4FN).

Procuramos nunca ter valores repetidos nas tabelas e que cada célula contenha apenas valores atômicos (1FN), colocamos chaves primárias em todas as tabelas e as chaves estrangeiras estão a ser utilizadas para estabelecer os relacionamentos entre as tabelas respetivas. Removemos todas as potenciais dependências parciais (2FN) e eliminamos dependências funcionais transitivas (3FN), ou seja, que cada coluna dependa totalmente da chave primária. Também eliminamos redundâncias decorrentes de múltiplos valores associados a uma chave primária (4FN).

Desde o início da criação da base de dados que se cultivaram estas práticas para facilitar todo o processo de desenvolvimento da base dados e para ela poder escalar de forma normalizada.



### 4.3. Apresentação e explicação do modelo lógico produzido

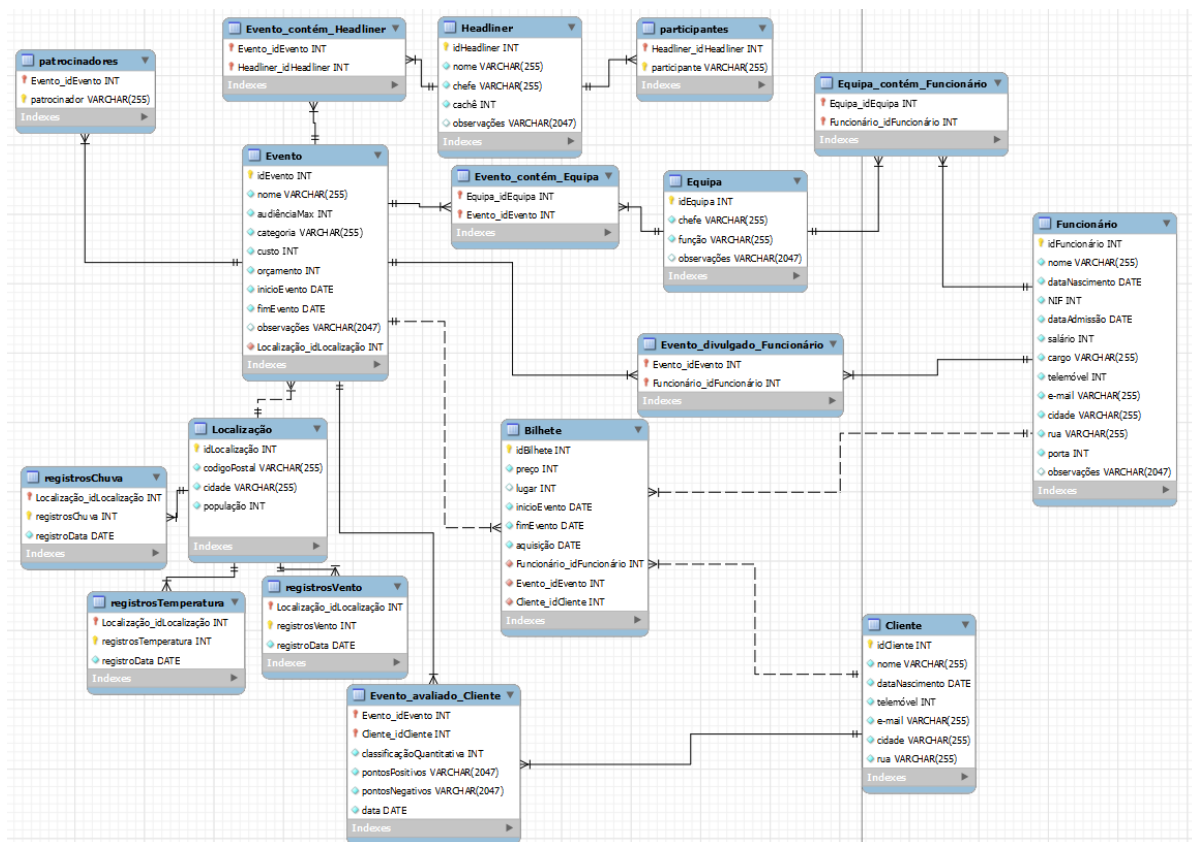


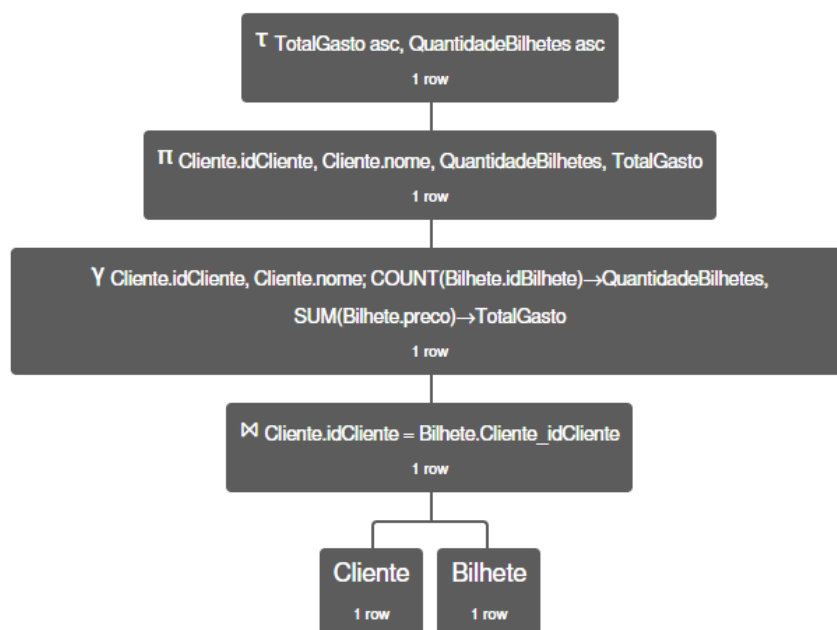
Figura 15 - Modelo Lógico

Após aplicarmos todas as técnicas e todas as descrições do modelo conceptual, obtivemos este modelo lógico da base de dados, cada tabela e relação apresentada na figura 15, já foi abordado ao longo dos tópicos anteriores e os valores estão detalhados nas tabelas do modelo conceptual associada aos atributos.

## 4.4. Validação do modelo com interrogações do utilizador

Para se validar o modelo lógico desenvolvido, foi usado a ferramenta ReLaX. Foi escolhido 3 dos Requisitos de Exploração para serem convertidos em Interrogações e descritos em Álgebra Relacional, de modo a confirmar a possibilidade de as executar no modelo referido. Com isto obtivemos as Árvore de demonstração abaixo descritas

### 1. Requisito de Exploração 8 Lista de Clientes



$\pi$  TotalGasto asc, QuantidadeBilhetes asc  $\Pi$  Cliente.idCliente, Cliente.nome, QuantidadeBilhetes, TotalGasto  $\gamma$   
 Cliente.idCliente, Cliente.nome; COUNT(Bilhete.idBilhete)→QuantidadeBilhetes,  
 SUM(Bilhete.preco)→TotalGasto ( Cliente  $\bowtie$  Cliente.idCliente = Bilhete.Cliente\_idCliente Bilhete )  
 Execution time: 0 ms

Figura 16 - RE08 ListaClientes

A interrogação da Figura 16 tem como objetivo listar todos os clientes e a quantidade correspondente de bilhetes comprados.

## 2. Requisito de Exploração 11 Lista de Eventos Pelo Sucesso de Vendas

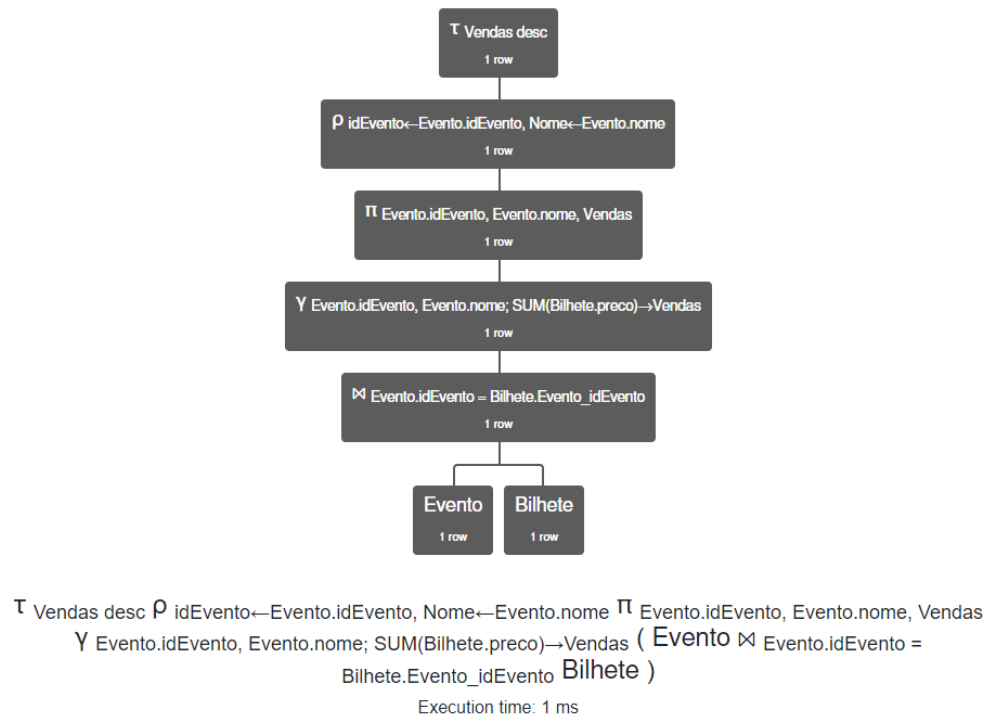


Figura 17 - RE11 ListaEventosSucessoVendas

A interrogação da Figura 17 tem como objetivo listar todos os pelo seu sucesso de vendas, isto é, quantos euros foram realizados.

### 3. Requisito de Exploração 16 Lista de Eventos por Classificação

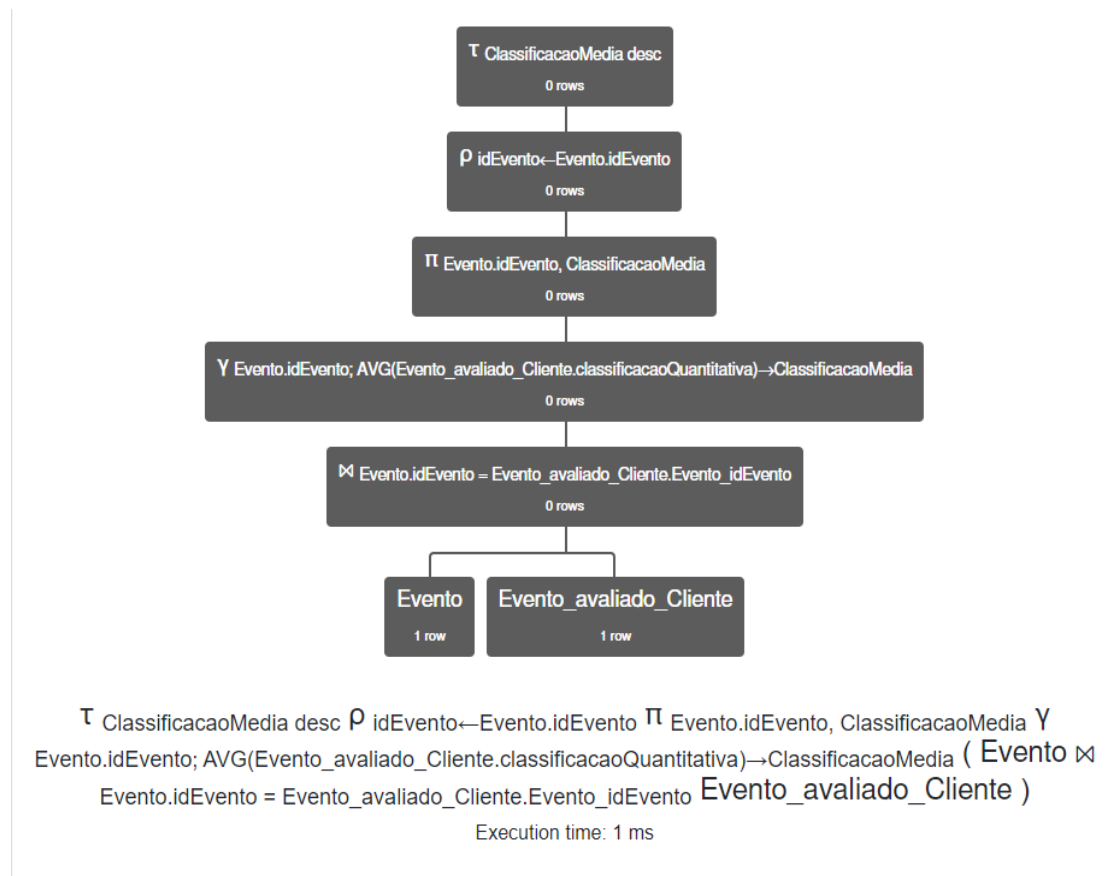


Figura 18 - RE16 ListaEventosClassificação

A

A interrogação da Figura 18 tem como objetivo listar todos os pela sua classificação, isto é, pela sua média quantitativa (0/10).

## 5. Implementação Física

### 5.1. Tradução do esquema lógico para o sistema de gestão de bases de dados escolhido

Neste trabalho, usamos o MySQL como sistema de gestão de bases de dados.

Traduzimos o modelo lógico para um sistema MySQL usando o MySQL Workbench para suportar e correr o nosso código. Para Gerar as tabelas necessárias executamos o código que está nos ficheiros do projeto.

Para explicar este tópico, iremos explicar uma tabela de cada tipo (entidade, relação N:N e multivalorado).

#### 1. Tabela da Entidade Evento

```
-- Tabela Evento
CREATE TABLE Evento (
  `idEvento` INT AUTO_INCREMENT,
  `nome` VARCHAR(255) NOT NULL,
  `audiênciaMax` INT NOT NULL,
  `categoria` VARCHAR(255) NOT NULL,
  `custo` INT NOT NULL,
  `orçamento` INT NOT NULL,
  `inicioEvento` DATE NOT NULL,
  `fimEvento` DATE NOT NULL,
  `observações` VARCHAR(2047),
  `Localização_idLocalização` INT NOT NULL,
  PRIMARY KEY(`idEvento`, `Localização_idLocalização`),
  FOREIGN KEY(`Localização_idLocalização`)
    REFERENCES `Localização`(`idLocalização`)
);
```

Figura 19 – Tabela Entidade SQL

Nesta Tabela é descrito o id sendo este um valor inteiro auto incrementado, é descrito também todos os outros atributos sendo estes todos “*Not Null*” excluindo as observações, campo reservado a uma breve frase referente a algum tópico atípico. A chave primária corresponde ao id do evento e ao id da localização (chave estrangeira). Em todas as tabelas, as chaves primárias é sempre n-tuplos contendo sempre o id da tabela e todas as chaves estrangeiras.

## 2. Tabela do multivalorado patrocinadores

```
-- Tabela Evento avaliado Cliente (relação N:N)
CREATE TABLE `Evento_avaliado_Cliente` (
  `Evento_idEvento` INT NOT NULL,
  `Cliente_idCliente` INT NOT NULL,
  `classificaçãoQuantitativa` INT NOT NULL,
  `pontosPositivos` VARCHAR(2047) NOT NULL,
  `pontosNegativos` VARCHAR(2047) NOT NULL,
  `date` DATE NOT NULL,
  PRIMARY KEY(`Cliente_idCliente`,`Evento_idEvento`),
  FOREIGN KEY(`Cliente_idCliente`)
    REFERENCES `Cliente`(`idCliente`),
  FOREIGN KEY(`Evento_idEvento`)
    REFERENCES `Evento`(`idEvento`)
);
```

Figura 20 - Tabela Evento avaliado Cliente SQL

Neste tipo de tabela, sendo esta a correspondente aos patrocinadores de um evento, é sempre necessário a chave estrangeira da tabela “mãe” e o atributo correspondente. Estes dois formam sempre a chave primária. Esta estrutura é análoga a todo tipo de tabelas de multivalorados.

## 3. Tabela da Relação Evento avaliado Cliente

```
-- Tabela patrocinadores -> Evento
CREATE TABLE `patrocinadores` (
  `Evento_idEvento` INT NOT NULL,
  `patrocinador` VARCHAR(255) NOT NULL,
  PRIMARY KEY(`patrocinador`,`Evento_idEvento`),
  FOREIGN KEY(`Evento_idEvento`)
    REFERENCES `Evento`(`idEvento`)
);
```

Figura 21 - Tabela patrocinadores SQL

Nesta tabela, é descrita uma relação de N para N. Todas as tabelas com este tipo de objetivo, têm sempre 2 chaves estrangeiras, neste caso, o id do evento e o id do cliente, e estas duas chaves formam a chave primária.

Neste caso específico, esta relação possui atributos correspondentes à opinião do cliente em relação ao evento participado, como pedido pelo CEO da empresa.

## 5.2. Tradução das interrogações do utilizador para SQL

Para a tradução das interrogações do utilizador para SQL, criamos 16 queries (cada uma ligada ao seu requisito de exploração). Abaixo serão apresentadas 3 delas e os respetivos outputs. O resto das queries estão nos scripts do projeto.

### 1. Lista dos Eventos nos próximos 30 dias

```
-- RE04
-- É importante uma lista de eventos nos próximos 30 dias
DROP PROCEDURE IF EXISTS ProximosEventos;
DELIMITER $$
CREATE PROCEDURE ProximosEventos()
BEGIN
SELECT * From Evento Ev
    WHERE Ev.inicioEvento > NOW() AND Ev.inicioEvento < NOW() + INTERVAL 30 DAY;
END$$
DELIMITER ;
CALL ProximosEventos();
```

Figura 22 - Query RE04

Nesta query, é possível introduzir ver os próximos eventos numa janela de 30 dias. Para tal, chamamos a tabela do evento e filtramos o eventos que estão entre o dia atual e os próximos 30.

	IdEvento	nome	audiênciaMax	categoria	custo	orçamento	inicioEvento	fimEvento	observações	Localização_IdLocalização
▶	8	Rap in Rio	10000	Festival	30000	32000	2023-12-11	2024-03-13	A ser processados pelo Rock in Rio	6
	9	Maybe there something gud	10000	peddypaper	2000	2000	2023-12-12	2024-03-06	max	11

Figura 23 - Output Query RE04

## 2. Lista de membros por Equipa

```
-- RE06
-- É relevante existir uma lista de membros por equipa
DROP PROCEDURE IF EXISTS ListaEquipa_Funcionários;
DELIMITER $$
CREATE PROCEDURE ListaEquipa_Funcionários(
    IN id INT
)
BEGIN
SELECT E.idEquipa AS 'ID Equipa', E.Chefe AS 'Chefe', E.função AS 'Função da Equipa',
    F.idFuncionário AS 'ID Funcionário', F.nome AS 'Nome Funcionário', F.cargo AS 'Cargo', F.`e-mail` AS 'E-Mail',
    F.`telemóvel` AS 'Telemóvel', F.`dataNascimento` AS 'Data Nascimento', F.`dataAdmissão` AS 'Data Admissão',
    F.`cidade` AS 'Cidade', F.`rua` AS 'Rua', F.`porta` AS 'Porta', F.`observações` AS 'Obs'
FROM Equipa E
JOIN equipa_contém_funcionário EF ON EF.Equipa_idEquipa = E.idEquipa
AND E.idEquipa = id
JOIN Funcionário F ON EF.Funcionário_idFuncionário = F.idFuncionário
GROUP BY F.idFuncionário;
END$$
DELIMITER ;
CALL ListaEquipa_Funcionários(2);
```

Figura 24 - Query RE06

Nesta query, é possível dar o id da equipa por parâmetro e obter todos os dados de todos os funcionários correspondentes a essa equipa. Para isto foi necessário usar 3 tabelas e usar a reunião das mesmas. As tabelas usadas foram a da Equipa, a do Funcionário e a da Relação entre estas duas.

ID Equipa	Chefe	Função da Equipa	ID Funcionário	Nome Funcionário	Cargo	E-Mail	Telemóvel	Data Nascimento	Data Admissão	Cidade	Rua	Porta	Obs
2	Daniel Tardio	Finanças	6	Daniel Tardio	Diretor de Finanças	dir_financas@mail.com	933333333	1969-03-02	2018-04-01	Braça	Rua Do Minho	3	Registo criminal
2	Daniel Tardio	Finanças	7	Daniel Homano	Finanças	financas_homano@mail.com	933333331	2002-07-11	2019-06-03	Porto	Rua Da Portela	15	
2	Daniel Tardio	Finanças	8	Rafael Pedro Grão	Finanças	financas_grao@mail.com	933333332	2002-04-12	2020-11-04	Leiria	Rua Da Lei da Ria	147	Tem dislexia de
2	Daniel Tardio	Finanças	9	Saluh Mah Hed	Finanças	financas_hed@mail.com	933333333	1950-04-12	2022-07-14	Marrakech	Ru di Marrak	1231	Tem dificuldades

Figura 25 - Output Query RE06



### 3. Opiniões de um Cliente

```
-- RE14
-- Tem de ser possível listar a opinião dos clientes
DROP PROCEDURE IF EXISTS OpiniãoCliente;
DELIMITER $$
CREATE PROCEDURE OpiniãoCliente(
    IN id INT
)
BEGIN
SELECT C.idCliente AS 'ID Cliente', C.nome AS 'Nome Cliente',
    E.idEvento AS 'ID Evento', E.nome AS 'Nome Evento',
    EC.classificaçãoQuantitativa AS 'Classificação 0/10',
    EC.pontosPositivos AS 'Pontos Positivos', EC.pontosNegativos AS 'Pontos Negativos',
    EC.`date` AS 'Data da Avaliação'
FROM Cliente C
JOIN `evento_avaliado_cliente` EC ON C.idCliente = EC.Cliente_idCliente
AND C.idCliente = id
JOIN Evento E ON E.idEvento = EC.Evento_idEvento
GROUP BY C.idCliente, E.idEvento;
END$$
DELIMITER ;
CALL OpiniãoCliente(1);
```

Figura 26 - Query RE14

Esta query permite à equipa que manuseará a Base Dados, a funcionalidade de obter todas as opiniões (feedback) de um cliente.

Para isto, usamos a reunião entre 3 tabelas (Evento, cliente, Relação evento e cliente).

ID Cliente	Nome Cliente	ID Evento	Nome Evento	Classificação 0/10	Pontos Positivos	Pontos Negativos	Data da Avaliação
1	Pedro Koi Tado	2	Meo Sudo Leste	4	Boa música e bom espaço	Maus acessos	2023-08-27
1	Pedro Koi Tado	6	Alquem quer	3	Bem organizado e com boa música	Casas de banho longe do palco princ...	2023-08-27

Figura 27 – Output Query RE14

## 5.3. Definição e caracterização das vistas de utilização em SQL

Para facilitar a utilização da base de dados com objetivo e focando nos dados principais a representar-se, foi criado algumas vistas (*views*) em SQL. Em baixo está o código implementado.

## 1. Vista Cliente

```
DROP VIEW IF EXISTS vistaCliente;  
CREATE VIEW vistaCliente AS  
    SELECT C.nome AS 'Nome do Clinte', E.nome AS 'Evento', B.preço AS 'Preço do Bilhete',  
           B.inicioEvento AS 'Inicio', B.fimEvento AS 'Fim'  
    FROM Cliente C  
    JOIN Bilhete B ON C.idCliente = B.Cliente_idCliente  
    JOIN Evento E ON B.Evento_idEvento = E.idEvento  
    GROUP BY C.nome,E.nome;
```

Figura 28 - Vista Cliente

Esta vista permite visualizar os clientes, os seus bilhetes e os eventos a qual este último está associado. Com isto, os clientes vão poder saber, caso se esqueçam, de qual evento o bilhete pertence, e os respetivos dados essenciais.

	Nome do Clinte	Evento	Preço do Bilhete	Inicio	Fim
▶	Pedro Koi Tado	Pimenta Vermelha Picante	10	2023-03-04	2023-03-04
	Alberto Quintas	Pimenta Vermelha Picante	12	0202-03-04	2023-03-04
	Bruno Miguel	Pimenta Vermelha Picante	22	2023-03-04	2023-03-04
	Henrique Casal	Pimenta Vermelha Picante	20	2023-03-04	2023-03-04
	Antonio Ferramenta	Meo Sudo Leste	7	2023-08-25	2023-08-28
	Gonçalo Ferreira	Meo Sudo Leste	30	2023-08-25	2023-08-28
	Alberto Quintas	Meo Sudo Leste	15	2023-08-25	2023-08-28
	Filipe Pereira	Meo Sudo Leste	15	2023-08-25	2023-08-28
	Vitor Lima	Meo Sudo Leste	27	2023-08-25	2023-08-28
	Tania Lemos	Meo Sudo Leste	10	2023-08-25	2023-08-28
	Nelson Almeida	Outono Prima Sound	50	2023-07-20	2023-03-04
	Mariana Correia	Outono Prima Sound	10	2023-03-04	2023-03-07
	Sofia Mata	Outono Prima Sound	9	2023-07-20	2023-03-04
	Daniel Fontes	Outono Prima Sound	13	2023-03-04	2023-03-07
	Pedro Alves	Outono Prima Sound	15	2022-03-04	2022-03-06
	Daniel Fontes	Culinária Portuguesa	5	2024-03-12	2024-03-13

Figura 29 - Output da Vista Cliente

## 2. Vista Funcionário

```
DROP VIEW IF EXISTS vistaFuncionário;  
CREATE VIEW vistaFuncionário AS  
SELECT F.nome AS 'Nome do Funcionário', EF.Equipa_idEquipa AS 'ID da Equipa', E.chefe AS 'Chefe'  
FROM Funcionário F  
JOIN equipa_contém_funcionário EF ON EF.Funcionário_idFuncionário = F.idFuncionário  
JOIN Equipa E ON E.idEquipa = EF.Equipa_idEquipa  
GROUP BY F.nome, EF.Equipa_idEquipa;
```

Figura 30 - Vista Funcionário

Esta Vista permite ver todos os funcionários da *Eventure* e a qual equipa eles pertencem. Esta vista é essencial principalmente para a gestão de recursos humanos, visto que a *Eventure* pode estar a organizar eventos de dimensões bastante grandes.

	Nome do Funcionário	ID da Equipa	Chefe
►	Vera Lama	1	Vera Lama
	João da Maria	1	Vera Lama
	Carlos Ramboia	1	Vera Lama
	Maria Marcelina	1	Vera Lama
	Daniel Tardio	2	Daniel Tardio
	Daniel Homano	2	Daniel Tardio
	Rafael Pedro Grão	2	Daniel Tardio
	Saluh Mah Hed	2	Daniel Tardio
	Evaldo Perfeito	3	Evaldo Perfeito
	Vale do Imperfeito	3	Evaldo Perfeito
	Jony P. Heinstain	3	Evaldo Perfeito
	Roberto Desleal	4	Roberto Desleal

Figura 31 - Output da Vista Funcionário

### 3. Vista Bilhete

```
DROP VIEW IF EXISTS vistaBilhete;  
CREATE VIEW vistaBilhete AS  
SELECT B.idBilhete AS 'ID Bilhete', C.nome AS 'Cliente que tem o bilhete'  
FROM Bilhete B  
JOIN Cliente C ON B.Cliente_idCliente = C.idCliente  
GROUP BY B.idBilhete, C.nome  
ORDER BY B.idBilhete;
```

Figura 32 - Vista Bilhete

Esta vista permite ver o ID do bilhete e a quem este está associado, isto tem como objetivo ajudar a controlar a entrada e saída dos recintos nos eventos da *Eventure*. Com isto, a base de dados ganhou a capacidade de listar todos os bilhetes e a quem estes estão associados.

	ID Bilhete	Cliente que tem o bilhete
▶	1	Pedro Koi Tado
	2	Alberto Quintas
	3	Nelson Almeida
	4	Bruno Miguel
	5	Nelson Almeida
	6	Daniel Fontes
	7	Antonio Ferramenta
	8	Bruno Miguel
	9	Tania Lemos
	10	Mariana Correia
	11	Gonçalo Ferreira
	12	Francisco Pacheco
	13	Adelia Campos
	14	Telmo Graça
	15	Henrique Casal
	16	José Barbosa
	17	Maria Falecida

Figura 33 - Output da Vista Bilhete

## 5.4. Definição e caracterização de alguns utilizadores em SQL

Para a resolução deste problema, criamos alguns utilizadores e adicionamos as permissões necessárias seguindo os requisitos levantados pelo CEO da empresa. Nos tópicos a seguir é mostrado um excerto do script que permite criarmos utilizadores, garantir e remover permissões, o resto do script encontra-se nos ficheiros do projeto.

### 1. Definição de utilizadores

```
-- Criação do utilizador para o CEO
CREATE USER 'CEOEventure'@'localhost';
SET PASSWORD FOR 'CEOEventure'@'localhost' = 'VitorFonseca';
-- Criação do utilizador 'prog' para a equipa de PROGAMAÇÃO
CREATE USER 'prog'@'localhost';
SET PASSWORD FOR 'prog'@'localhost' = 'LCC2023';
-- Criação do utilizador para o Daniel Tardio
CREATE USER 'FinancasEventure'@'localhost';
SET PASSWORD FOR 'FinancasEventure'@'localhost' = 'DanielTardio';
-- Criação do utilizador para o Vera Lama
CREATE USER 'MarketingEventure'@'localhost';
SET PASSWORD FOR 'MarketingEventure'@'localhost' = 'VeraLama';
-- Criação do utilizador para o Evaldo Perfeito
CREATE USER 'RPEventure'@'localhost';
SET PASSWORD FOR 'RPEventure'@'localhost' = 'EvaldoPerfeito';
-- Criação de um utilizador 'funcionário'
CREATE USER 'funcionarioEventure'@'localhost';
SET PASSWORD FOR 'funcionarioEventure'@'localhost' = 'funcionarioEventure2023';
-- Criação de um utilizador 'funcionário'
CREATE USER 'clienteEventure'@'localhost';
SET PASSWORD FOR 'clienteEventure'@'localhost' = 'JonyBravoEventure2023';
```

Figura 34 - Criar Utilizadores SQL

A criação de utilizadores é bastante importante para assegurarmos a integridade da base de dados impossibilitando acessos ou manipulações indevidas. Para isso a equipa de desenvolvimento criou acesso ao CEO, aos chefes de cada departamento, aos funcionários gerais e criamos um cliente para testarmos a experiência de um cliente.

## 2. Dar todas permissões a certos utilizadores

```
-- -----  
-- Permissões  
-- -----  
-- RC04/RC05  
-- Permissão de acesso a todos os objectos da base de dados em 'localhost'.  
GRANT ALL ON Eventure.* TO 'CEOEventure'@'localhost';  
GRANT ALL ON Eventure.* TO 'prog'@'localhost';  
  
-- Adicionar todas as permissões, é mais fácil remover depois  
GRANT SELECT, INSERT, UPDATE ON Eventure.* TO 'FinancasEventure'@'localhost';  
GRANT SELECT, INSERT, UPDATE ON Eventure.* TO 'MarketingEventure'@'localhost';  
GRANT SELECT, INSERT, UPDATE ON Eventure.* TO 'RPEventure'@'localhost';
```

Figura 35 – Garantir Permissões Utilizadores SQL

Após a criação dos utilizadores, é imprescindível garantir e revogar acesso às funções da base de dados. Para isto, a equipa de desenvolvimento preferiu garantir acesso a tudo e retirar o que não é permitido pela requisição do CEO da *Eventure* (como podemos ver na figura 35 e 36).

## 3. Retirar certas permissões a utilizadores

```
-- RC01/RC02  
-- Remover Permissões  
REVOKE EXECUTE ON PROCEDURE ListaNFuncionários_Equipa FROM 'MarketingEventure'@'localhost';  
REVOKE EXECUTE ON PROCEDURE ListaEquipa_Funcionários FROM 'MarketingEventure'@'localhost';  
REVOKE EXECUTE ON PROCEDURE FinancasGerais FROM 'MarketingEventure'@'localhost';  
REVOKE EXECUTE ON PROCEDURE AddEvento FROM 'MarketingEventure'@'localhost';  
REVOKE EXECUTE ON PROCEDURE AddEquipa FROM 'MarketingEventure'@'localhost';  
REVOKE EXECUTE ON PROCEDURE AddFuncionário FROM 'MarketingEventure'@'localhost';  
REVOKE EXECUTE ON PROCEDURE RemEvento FROM 'MarketingEventure'@'localhost';  
REVOKE EXECUTE ON PROCEDURE RemEquipa FROM 'MarketingEventure'@'localhost';  
REVOKE EXECUTE ON PROCEDURE RemFuncionário FROM 'MarketingEventure'@'localhost';
```

Figura 36 - Retirar Permissões Utilizadores SQL

Com esta implementação, foi permitido ao CEO da *Eventure* e à sua equipa de finanças o controlo máximo da base de dados. Foi retirado bastantes funcionalidades aos funcionários e aos outros departamentos da empresa e por fim, foi dado a possibilidade aos clientes de alterar os próprios dados na base de dados, caso estes queiram.

## 5.5. Cálculo do espaço da base de dados (inicial e taxa de crescimento anual)

Recorrendo à documentação oficial do MySQL, obtivemos a tabela abaixo que contém todos os domínios usados ao longo deste projeto:

Custo Domínios	
Tipo	Custo
INT	4 Bytes
VARCHAR(M), 0<=M<=255	M + 1 Bytes
VARCHAR(M), M>=255	M + 2 Bytes
DATE	3 Bytes

Tabela 18 - Custo de Memória SQL

Tendo em conta os valores acima, agora é necessário fazer a conta de todas os atributos presentes na base de dados correspondentes a cada tabela para assim prosseguir mos para a sua contabilização.

Seguindo a seguinte fórmula:

$$custo = x * size(INT) + y * size(VARCHAR(255)) + z * size(VARCHAR(2047)) + k * size DATE)$$

$x =$  Quantidade de INTs;

$z =$  Quantidade de VARCHARs(2047);

$y =$  Quantidade de VARCHARs(255);

$k =$  Quantidade de DATEs;

(e seguindo os valores de tamanho da tabela 36 acima apresentada)

Domínios na base de dados	
Tabela	Custo
Evento	$(5 * 4) + (2 * 256) + (1 * 2049) + (2 * 3) = 20 + 512 + 2049 + 6 = 2587$ Bytes
Localização	$(2 * 4) + (2 * 256) + (0 * 2049) + (0 * 3) = 8 + 512 + 0 + 0 = 520$ Bytes
Headliner	$(2 * 4) + (2 * 256) + (1 * 2049) + (0 * 3) = 8 + 512 + 2049 + 0 = 2569$ Bytes
Bilhete	$(6 * 4) + (0 * 256) + (0 * 2049) + (3 * 3) = 24 + 0 + 0 + 9 = 33$ Bytes
Cliente	$(2 * 4) + (4 * 256) + (0 * 2049) + (1 * 3) = 8 + 1024 + 0 + 3 = 1035$ Bytes
Equipa	$(1 * 4) + (2 * 256) + (1 * 2049) + (0 * 3) = 4 + 512 + 2049 + 0 = 2565$ Bytes
Funcionário	$(5 * 4) + (5 * 256) + (1 * 2049) + (2 * 3) = 20 + 1280 + 2049 + 6 = 3355$ Bytes
patrocinadores	$(1 * 4) + (1 * 256) + (0 * 2049) + (0 * 3) = 4 + 256 + 0 + 0 = 260$ Bytes
participantes	$(1 * 4) + (1 * 256) + (0 * 2049) + (0 * 3) = 4 + 256 + 0 + 0 = 260$ Bytes
registrosChuva	$(2 * 4) + (0 * 256) + (0 * 2049) + (1 * 3) = 8 + 0 + 0 + 3 = 11$ Bytes
registrosTemperatura	$(2 * 4) + (0 * 256) + (0 * 2049) + (1 * 3) = 8 + 0 + 0 + 3 = 11$ Bytes
registrosVento	$(2 * 4) + (0 * 256) + (0 * 2049) + (1 * 3) = 8 + 0 + 0 + 3 = 11$ Bytes
Evento_contém_Headliner	$(2 * 4) + (0 * 256) + (0 * 2049) + (0 * 3) = 8 + 0 + 0 + 0 = 8$ Bytes
Evento_contém_Equipa	$(2 * 4) + (0 * 256) + (0 * 2049) + (0 * 3) = 8 + 0 + 0 + 0 = 8$ Bytes
Equipa_contém_Funcionário	$(2 * 4) + (0 * 256) + (0 * 2049) + (0 * 3) = 8 + 0 + 0 + 0 = 8$ Bytes
Evento_divulgado_Funcionário	$(2 * 4) + (0 * 256) + (0 * 2049) + (0 * 3) = 8 + 0 + 0 + 0 = 8$ Bytes
Evento_avaliado_Cliente	$(3 * 4) + (0 * 256) + (2 * 2049) + (1 * 3) = 12 + 0 + 4098 + 3 = 4113$ Bytes
<b>Soma</b>	<b>17 362 Bytes</b>

Tabela 19 - Custa Tabelas Eventure

Se todas as tabelas da base de dados da *Eventure* tiverem exatamente o mínimo de preenchimento, o custo da base de dados é de 17 362 Bytes.



### Crescimento ao longo de 10 anos

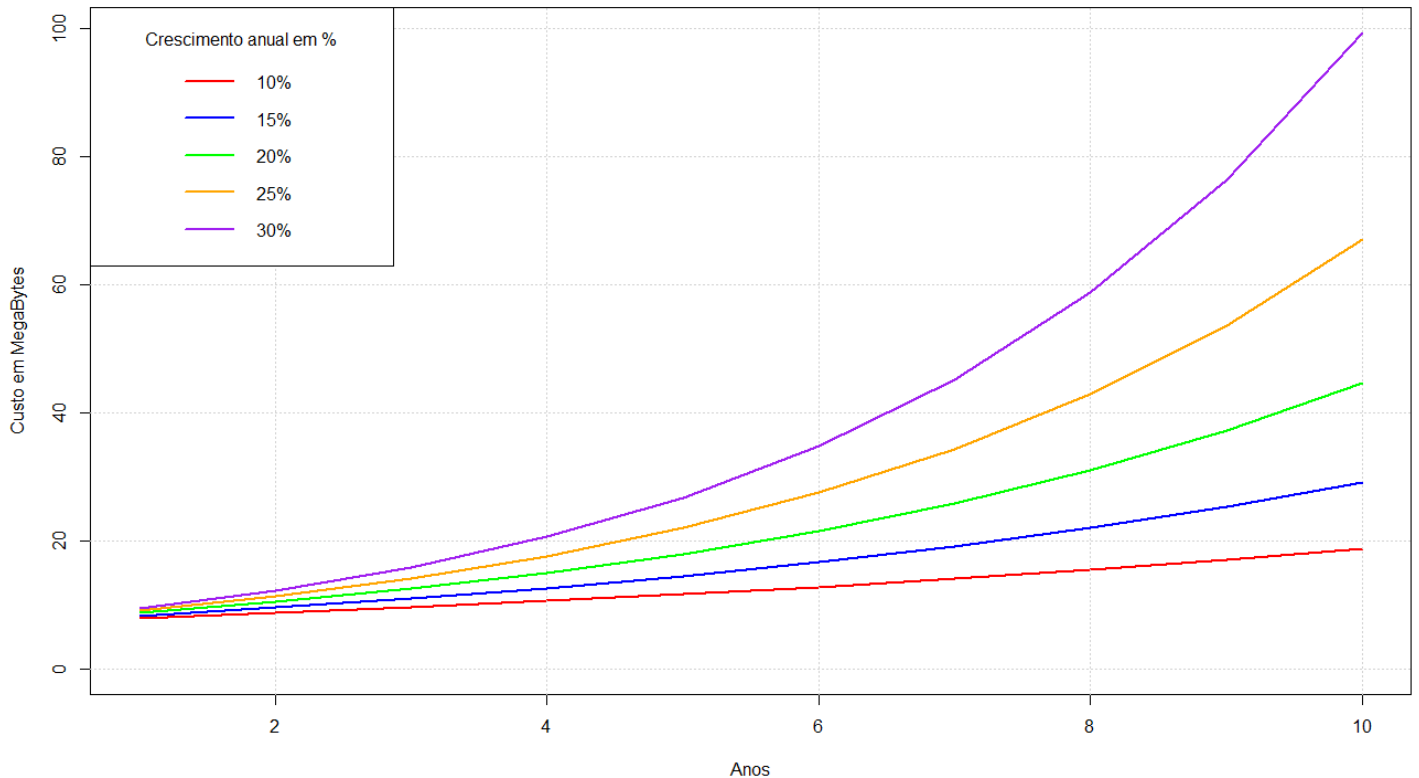


Figura 37 - Projeção do Custo da Base de Dados da Eventure

Uma projeção é muito difícil de se fazer devido às várias variáveis que todo o processo está dependente logo será necessário definir alguns pontos cruciais, partindo do princípio de uma média de novos 99 eventos no primeiro ano, uma média de 29 999 bilhetes vendidos e 2 999 novos clientes, 299 headliners novos e mantendo as localizações, 1 999 Relações entre entidades novas e cerca de 599 feedbacks, funcionários e equipas adição nem remoção na base de dados.

$$custo_{1^{a}ano} = 17\,362 + (99 * 2\,587) + (29\,999 * 33) + (2\,999 * 1\,035) + (299 * 2\,569) + (1\,999 * 8) + (499 * 4\,113)$$

$$\equiv custo_{1^{a}ano} = 17362 + 256\,113 + 989\,967 + 3\,103\,965 + 768\,131 + 15\,992 + 2\,052\,387$$

$$\equiv custo_{1^{a}ano} = 7\,203\,917 \text{ Bytes} \approx 7.2 \text{ MegaBytes}$$

Partindo do pressuposto que no primeiro ano são necessários 7 203 917 Bytes e imaginando um crescimento anual de 10%, 15%, 20%, 25% e 30%, conseguimos obter o seguinte gráfico.

## 5.6. Indexação do Sistema de Dados

Foi tentada a indexação do sistema criando índices nos id's principais das tabelas, mas em nenhuma query testada foi notada uma melhoria significativa de performance, mesmo testando com, por exemplo, 50.000 bilhetes todos aleatórios (máximo de amostra que o software permitiu) e mesmo usando o computador mais fraco da equipa.

```
-- Executa a consulta
SELECT E.idEvento AS 'ID Evento', E.nome AS 'Nome',
       SUM(B.preço) AS 'Vendas €'
FROM Evento E
JOIN Bilhete B ON E.idEvento = B.Evento_idEvento
GROUP BY E.idEvento
ORDER BY SUM(B.preço) DESC;
```

Figura 38 - Query Sucesso Eventos

Foi usada a query de cima para o teste e os resultados estão em baixo, a seta azul é referente à utilização da query sem utilizar índices e a seta verde corresponde à utilização dos índices na query.

6	0.89083950	SELECT E.idEvento AS 'ID Evento', E...	←
7	<del>0.00000000</del>	SHOW WARNINGS	
8	0.94817200	SELECT E.idEvento AS 'ID Evento', E...	←

Figura 39 - Resultado Índices

Para criarmos os índices foram usado os dois comandos abaixo.

```
-- Criar os índices
CREATE INDEX idx_Evento_idEvento ON Evento(idEvento);
CREATE INDEX idx_Bilhete_Evento_idEvento ON Bilhete(Evento_idEvento);
```

Figura 40 - Comandos Criar Index

Com isto, a equipa de desenvolvimento concluiu que não há necessidade, no estado atual da base de dados, de criar índices visto que a base de dados não ter qualquer problema de desempenho detetado, uma implementação futura que poderia ser interessante seria a base de dados continuar a evoluir e ficar cada mais complexa a cada update.

## 5.7. Procedimentos Implementados

Na base de dados da *Eventure*, foi implementado uma lista de procedimentos, sendo estes, todos necessários para resolver os requisitos de exploração levantados na fase inicial, e alguns extras com o objetivo de adicionar ou remover informação da base de dados.

Abaixo está algumas amostras de todos os procedimentos criados e a explicação de uma adição e remoção na base de dados. O resto da implementação encontra-se nos ficheiros do projeto.

### 1. Lista de procedimentos dos Requisitos Exploração

```
-----  
-- Chamar as Proc. de Exploração  
-----  
  
CALL QuantidadeBilhetes('2023-01-01','2024-12-30');           -- recebe 2 datas  
CALL ListaNFuncionários_Equipa(2);                           -- recebe 1 id de uma Equipa  
CALL ListaEventos_Localização(7);                             -- recebe 1 id de uma Localização  
CALL ProximosEventos();  
CALL ListaDadosChuva_Localização('2023-01-01','2024-12-30', 3); -- recebe 2 datas e 1 id de uma Localização  
CALL ListaDadosTemperatura_Localização('2023-01-01','2024-12-30', 3); -- recebe 2 datas e 1 id de uma Localização  
CALL ListaDadosVento_Localização('2023-01-01','2024-12-30', 3); -- recebe 2 datas e 1 id de uma Localização  
CALL ListaEquipa_Funcionários(2);                             -- recebe 1 id de uma Equipa  
CALL ListaPatrocinadores(2);                                  -- recebe 1 id de um Evento  
CALL ListaClientes();  
CALL HistóricoCliente(2);                                     -- recebe 1 id de um Cliente  
CALL FinançasGerais();  
CALL ListaEventosSucessoVendas();  
CALL ListaEventosSucessoLucros();  
CALL ListaEventos_Headliner(1);                               -- recebe 1 id de um Headliner  
CALL OpiniãoCliente(1);                                       -- recebe 1 id de um Cliente  
CALL PosNegEventos(2);                                         -- recebe 1 id de um Evento  
CALL ListaEventosClassificação();
```

Figura 41 - Procedimentos dos Requisitos de Exploração

Todos os procedimentos acima referidos têm em comentário todos os inputs necessários para a sua execução, para amostra da realização destes procedimentos, temos alguns exemplos no capítulo 5.2.

## 2. Lista de procedimentos de Adição na Base de Dados

```
-- Chamar as funções de ADD

CALL AddEvento('Tá Bem tá', 10000, 'Uma Cena QQL', 13000, 12000, '2022-05-04', '2022-05-07', '5', NULL);
CALL AddCliente('Quim das Sirenes', '1982-01-02', 91133322, 'OsBomBeiros@mail.com', 'Braga', 'Rua da Estrada');
CALL AddEquipa('Jony P. Heinstein', 'Montar as Coisas', 'Precisam de uma carrinha');
CALL AddBilhete(12, NULL, '2023-08-15', '2024-08-16', '2023-02-01', 1, 1, 3);
CALL AddFuncionário('Saluh Saly', '1950-04-12', 26643242, '2022-07-14', 3549, 'Finanças', '9355333', 'finanças_Saly@mail.com', 'Marrakech', 'Ru di Marrak', 1231, NULL);
CALL AddHeadliner('Travis Scott', 'Jacques Webster II', 3000, 'Quer um AP');
CALL AddLocalização('3333', 'Lisboa dos Mouros', 666);
CALL AddParticipantes(2, 'Kurt Cobain bem High');
CALL AddPatrocinadores(3, 'Au Chane');
CALL AddRegistroChuvaLoc('7', '67', '2022-08-15');
CALL AddRegistroTempLoc('8', '67', '2022-08-15');
CALL AddRegistroVentoLoc('9', '67', '2022-08-15');
CALL AddRelacaoEquipa_Funcionário(5, 5);
CALL AddRelacaoEvento_Cliente(5, 5, 4, 'Boa música e bom espaço', 'Maus acessos', '2023-08-27');
CALL AddRelacaoEvento_Equipa(5, 6);
CALL AddRelacaoEvento_Headliner(5, 5);
CALL AddRelacaoEvento_Funcionário(6, 6);
```

Figura 42 - Procedimentos de Adição na Base de Dados

Todos estes procedimentos têm como fundamento o comando *insert* do SQL, e simplesmente recebem os parâmetros necessários para adicionar os dados à base de dados. Foi criado um procedimento de adicionar por cada tabela presente na base dados da *Eventure*.

Abaixo temos um exemplo da realização desta tarefa.

```
-- Adicionar Relação Entre Evento Cliente
DROP PROCEDURE IF EXISTS AddRelacaoEvento_Cliente;
DELIMITER $$
) CREATE PROCEDURE AddRelacaoEvento_Cliente(
    IN Evento_idEvento1 INT,
    IN Cliente_idCliente1 INT,
    IN classificaçãoQuantitativa1 INT,
    IN pontosPositivos1 VARCHAR(2047),
    IN pontosNegativos1 VARCHAR(2047),
    IN date1 DATE
)
) BEGIN
INSERT INTO `evento_avaliado_cliente`
(`Evento_idEvento`, `Cliente_idCliente`, `classificaçãoQuantitativa`, `pontosPositivos`, `pontosNegativos`, `date`)
VALUES
(Evento_idEvento1, Cliente_idCliente1, classificaçãoQuantitativa1, pontosPositivos1, pontosNegativos1, date1);
) END$$
DELIMITER ;
```

Figura 43 - Exemplo de Procedimento de Adição na Base de dados

Todos os procedimentos deste tópico seguem a mesma estrutura lógica, parâmetros de todos os atributos e posteriormente o *insert* na tabela.

### 3. Lista de procedimentos de Remoção na Base de Dados

```
CALL RemEvento(1,'7');
CALL RemCliente(1);
CALL RemEquipa(1);
CALL RemBilhete(1, 1, 1, 3);
CALL RemFuncionário(1);
CALL RemHeadliner(1);
CALL RemLocalização(1);
CALL RemParticipantes(3, 'Matt Helders');
CALL RemPatrocinadores(1, 'ErreEfeEme');
CALL RemRegistroChuvaLoc('1','67');
CALL RemRegistroTempLoc('1','23');
CALL RemRegistroVentoLoc('1','67');
CALL RemRelacaoEquipa_Funcionário(1,2);
CALL RemRelacaoEvento_Cliente(2,1);
CALL RemRelacaoEvento_Equipa(1,1);
CALL RemRelacaoEvento_Headliner(2,1);
CALL RemRelacaoEvento_Funcionário(2,1);
```

Figura 44 - Procedimentos de Remoção na Base de Dados

Todos os procedimentos descritos acima possuem também a mesma estrutura, mas neste caso, recebem a chave primária referente à tabela, e a partir daí é possível remover aquela porção de informação. Só é permitido remover informação corresponde a uma chave primária de cada vez. Em baixo temos um excerto a mostrar uma função.

```
-- Remover Bilhete
DROP PROCEDURE IF EXISTS RemBilhete;
DELIMITER $$
CREATE PROCEDURE RemBilhete(
    IN id INT,
    IN idFuncionário INT,
    IN idEvento INT,
    IN idCliente INT
)
BEGIN
DELETE FROM Bilhete B WHERE B.idBilhete = id AND B.Funcionário_idFuncionário = idFuncionário
AND B.Evento_idEvento = idEvento AND B.Cliente_idCliente = idCliente;
END$$
DELIMITER ;
```

Figura 45 - Exemplo de Procedimento de Remoção na Base de dados

Como citado acima, todos os procedimentos de remoção são iguais estruturalmente. Recorremos ao comando ‘delete’ para apagar da base de dados as informações que o utilizador desejar. Isto, comparando sempre os ids do parâmetro com os ids pretendidos.

## 5.8. Plano de segurança e recuperação de dados

“Mais vale prevenir do que remediar” é o ditado popular que mais importa numa base de dados. A segurança e integridade dos dados tem de ser garantida. Com isso, a equipa de desenvolvimento ensinou ao CEO da Eventure como fazer um backup.

Os comandos entregues ao Vítor Fonseca, partindo da informação que usa Windows foram:

1. Abrir CMD com permissões de administrador
2. `>> cd..` (até chegar ao topo do disco)
3. `>> cd Program Files`
4. `>> cd MySQL`
5. `>> cd MySQL Server 8.1` (8.1 é a versão atual do server)
6. `>> cd bin`
7. `>> mysqldump -u USER -p PASSWORD Eventure > backupEventure.sql`

Após esta execução sequenciada, é possível gerar o backup da base de dados de toda a Eventure. Após isto, é fundamental guardar o backup num local de memória desconectado de qualquer sistema informático online para máxima segurança, uma Pen-Drive ou um Disco Externo seriam o ideal para tal.

Foi aconselhado também fazer um backup todos os dias para evitar ao máximo perdas de dados.

Com esta pequena medida deixamos o Vítor Fonseca o mais confiante possível com o sistema implementado. Em baixo está a sequência executada no computador do desenvolvedor Rúben que serviu para demonstração ao CEO.



```
Administrator: Linha de comandos
Microsoft Windows [Version 10.0.22621.2715]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Windows\System32>cd ..
C:\Windows>cd..
C:\>cd Program Files
C:\Program Files>cd Mysql
C:\Program Files\Mysql>cd Mysql server 8.1
C:\Program Files\Mysql\MySQL Server 8.1>cd bin
C:\Program Files\Mysql\MySQL Server 8.1\bin>mysqldump -u root -p password eventure > C:\Users\ruben\Desktop\eventureBackup.sql
```

Recuar até ao topo da memória

Chegar à diretoria do MySQL

Figura 46 - Demonstração do mysqldump

O output do ‘*mysqldump*’ é um ficheiro que contém as instruções que permite o restauro da base de dados complete.

## 5.9. Dashboard da Base de Dados (MS Power BI)

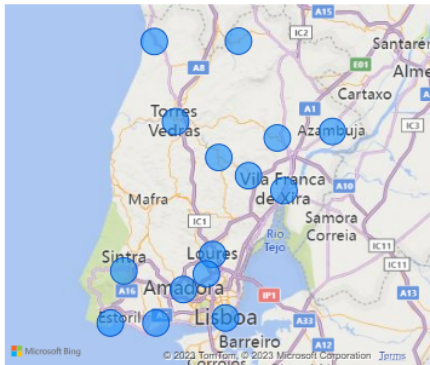
Ao fim da implementação toda, numa conversa com o Daniel Tardio, pediu à equipa de desenvolvimento se seria possível fazer uma Dashboard com alguns valores que estão na base de dados relativos às finanças da Eventure.

Como solução a esse pedido, decidimos usar o MS Power BI (ferramenta de estatística da Microsoft).

Foi requerido 6 tópicos pelo departamento de finanças:

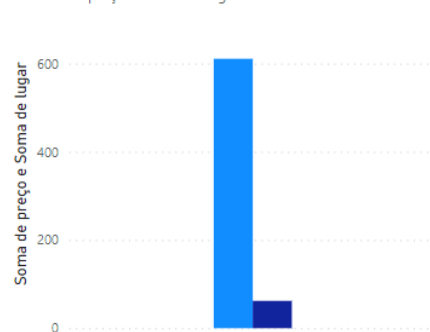
- Um mapa com a soma do Cachê da localização;
- Uma tabela com a soma dos bilhetes com e sem lugar;
- O orçamento dos eventos com um custo mais elevado;
- A audiência máxima de cada Evento;
- Um gráfico circular com o cachê de cada Headliner;
- Finanças gerais da empresa.

Soma de cachê por cidade

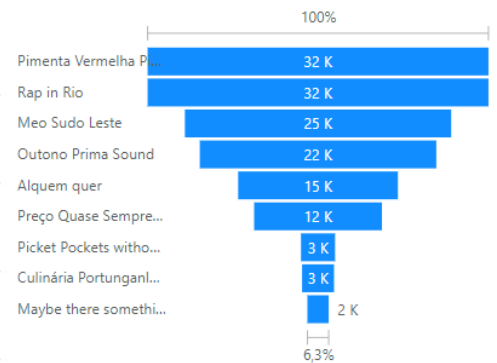


Soma de preço e Soma de lugar

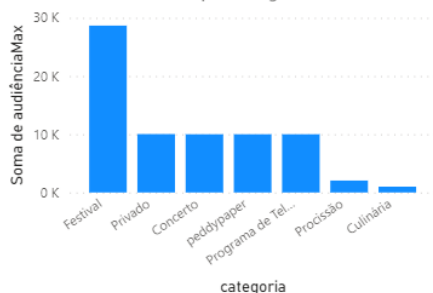
● Soma de preço ● Soma de lugar



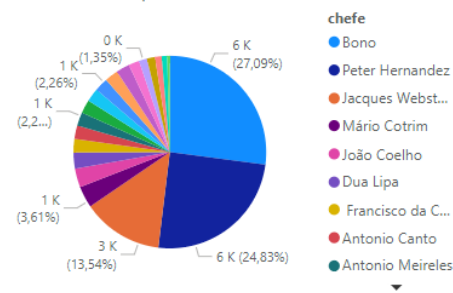
Soma de orçamento por nome



Soma de audiênciaMax por categoria



Soma de cachê por chefe



Soma de custo e Soma de orçamento

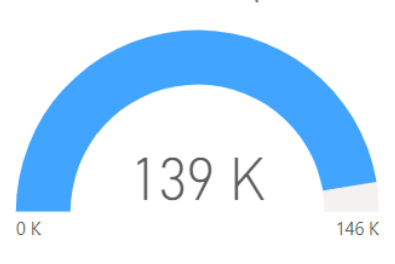


Figura 47 - Dashboard Finanças

## 6. Conclusões e Trabalho Futuro

Durante a realização deste trabalho tivemos um claro crescimento no conhecimento inerente às matérias lecionadas na cadeira de Bases de Dados. Acharmos que experiencia de fazer um trabalho a partir de requisitos foi fundamental para o nosso desenvolvimento a nível de projeção de um trabalho coeso e lógico foi bastante positiva. O facto deste trabalho ser baseado numa necessidade real numa empresa que trabalha com algum nível de abstração fez com que abordássemos o problema numa maneira diferente do habitual, a possibilidade da liberdade criativa foi algo que promoveu bastante o nosso esforço. Para finalizar, vamos enumerar alguns pontos que podem ser melhorados numa futura versão desta base de dados. O desenvolvimento de um modelo conceptual bastante melhor. Nós deparamo-nos muitas vezes a editar o modelo conceptual porque alguns atributos claramente não faziam muito sentido. O desenvolvimento de gatilhos é também um ponto que gostaríamos de abordar visto que estes são essenciais para o bom funcionamento da base de dados. Em relação desempenho coletivo, haverá sempre pontos a melhorar, mas num caso geral tivemos todos um papel fundamental. A nossa dificuldade na coletividade claramente foi a divergência criativa que levou muitas vezes ao *'roteiro'* a não sair como o esperado na altura e levar nos bastante vezes a refazer certas partes.

Também sentimos que podíamos ter perdido mais tempo a trabalhar num plano de segurança melhor. Uma dashboard mais completa e “esteticamente mais bonita” seria interessante de ser melhorado.

Um ponto fundamental que achamos que *'falhamos redondamente'* foi na atribuição de tamanho aos atributos 'VARCHAR', existe demasiado espaço desperdiçado que só reparamos na implementação física do trabalho e quando fizemos a projeção do custo à Eventure.

Concluindo este trabalho, achamos que fizemos um trabalho bom e tivemos todos uma prestação sólida. Mas o principal é trazer aos funcionários da Eventure uma facilidade de gestão daquela incrível empresa que põe um sorriso em cada lisboeta que presencia os seus eventos.



## Referências

- Material da unidade curricular
- [https://www.youtube.com/watch?v=Ofktsne-utM&list=PLHz\\_AreHm4dkBs-795Dsgvau\\_ekxg8g1r&ab\\_channel=CursoemVÍdeo](https://www.youtube.com/watch?v=Ofktsne-utM&list=PLHz_AreHm4dkBs-795Dsgvau_ekxg8g1r&ab_channel=CursoemVÍdeo)
- <https://dev.mysql.com/doc/refman/8.0/en/storage-requirements.html>Anexos
- <https://learn.microsoft.com/en-us/power-bi/>