



Universidade do Minho
Licenciatura em Ciências da Computação

Unidade Curricular de Bases de Dados

Ano Letivo de 2023/2024

Gestão de Eventos na Cidadela das Flores

**Miguel Martins A97441 &
Paulo Freitas A100053 &
Wellyson Vieira A100078**

BD

Janeiro, 2023

Data de Receção	
Responsável	
Avaliação	
Observações	

Gestão de Eventos em Cidadela das Flores

Miguel Martins A97441 &

Paulo Freitas A100053 &

Wellyson Vieira A100078

Janeiro, 2023

Resumo

O seguinte relatório foi realizado no âmbito de um projeto da unidade curricular de Bases de Dados, sendo este um ótimo meio de ajudar na aprendizagem do conteúdo lecionado na mesma, principalmente na medida de criar e desenvolver uma base de dados.

Face a um problema de gestão de eventos, decidimos abordá-lo com estes sistemas, de modo a existir uma melhor organização na calendarização da cidade. Após consulta com os habitantes, fomos capazes de delinear uma estratégia. Na primeira fase abordamos a definição do sistema, onde definimos os objetivos da base de dados, os recursos necessários para executar o projeto e a equipa que trabalhará no mesmo, de seguida abordamos a definição de requisitos, que consistiu na organização, análise e validação dos mesmos e por fim o modelo conceptual, na qual identificamos e caracterizamos as entidades, relacionamentos e dos atributos que foram associados com os mesmos, permitindo-nos assim formar e explicar um diagrama ER. Na segunda parte, debruçamo-nos na parte lógica e física do projeto, no qual se construiu e validou o esquema lógico, com ajuda também de interrogações do utilizador; posteriormente prosseguiu-se para a verificação das regras de normalização. De seguida, tratamos da implementação da parte física, começando por traduzir o nosso esquema lógico e implementando-se *queries*, *procedures*, vistas e indexações. Finalmente, foi calculado o espaço da nossa base de dados e também uma taxa de crescimento, além de ser projetado um plano de segurança e de recuperação de dados. Com isto tudo implementado, foi possível dar como terminado o nosso trabalho.

Área de Aplicação: Desenho e criação de Sistemas de Bases de Dados.

Palavras-Chave: Bases de Dados, Definição do Sistema, Levantamento de Requisitos, Modelo Conceitual, Diagrama ER., Modelo Lógico, Modelo Físico

Índice

Resumo	1
Índice	2
Índice de Tabelas	4
Índice de Figuras	5
1. Introdução	5
1.1. Contextualização	5
1.2. Objetivos	6
1.3. Análise da Viabilidade do Processo	6
1.4. Recursos e Equipa de Trabalho	6
1.5. Plano de Execução do Projeto	7
2. Levantamento e Análise de Requisitos	8
2.1. Método de Levantamento de Requisitos	8
2.2. Organização dos Requisitos	8
2.2.1 Requisitos de Descrição	9
2.2.2. Requisitos de Exploração	9
2.2.3. Requisitos de Controlo	9
2.3. Análise e Validação Geral dos Requisitos	10
3. Modelação Conceptual	11
3.1. Apresentação da abordagem de modelação realizada	11
3.2. Identificação e Caraterização das Entidades	11
3.3. Caraterização dos Relacionamentos	12
3.4. Identificação e Caraterização da Associação dos Atributos com as Entidades e os Relacionamentos	14
3.5. Apresentação e Explicação do Diagrama ER	15
4. Modelação Lógico	16
4.1. Construção, Validação e Apresentação do Modelo de Dados Lógico	16
4.2. Normalização de Dados	18
4.3. Validação do modelo com interrogações do utilizador	19
5. Implementação Física	20
5.1. Tradução do Modelo Lógico para o Modelo Físico	20

5.2. Tradução das interrogações do utilizador para SQL	21
5.3. Definição e caracterização das vistas de utilização em SQL	23
5.4. Cálculo do espaço da base de dados	26
5.5. Indexação do Sistema de Dados	28
5.6. Procedimentos implementados	28
5.7. Plano de segurança e recuperação de dados	31
6. Conclusões e Trabalho Futuro	32
7. Referências	33
8. Lista de Siglas e Acrónimos	34
9. Anexos	35
9.1. Script da criação da Base de dados	35
9.2. Script do Povoamento da Base de Dados	38
9.3. Script dos Procedimentos da Base de Dados	40

Índice de Tabelas

Tabela 1 - Diagrama de Gantt Parte 1	7
Tabela 2 - Diagrama de Gantt Parte 2	7
Tabela 3 - Requisitos de descrição	9
Tabela 4 - Requisitos de exploração	9
Tabela 5 - Requisitos de controlo	9
Tabela 6 - Caracterização dos Atributos da entidade Cliente	14
Tabela 7 - Caraterização dos Atributos da entidade Evento	14
Tabela 8 - Caraterização dos Atributos da entidade Agente	14
Tabela 9 - Caraterização dos Atributos da entidade Morada	15
Tabela 10 - Informações dos Clientes	21
Tabela 11 - Informações de um evento que correu numa dada data	21
Tabela 12 - Informações dos clientes que foram a um determinado evento e têm mais de 24 anos	22
Tabela 13 - Informações dos Agentes que participaram de um evento e receberam mais de 300 euros	22
Tabela 14 - Resultado da View Clientes_PBA	23
Tabela 15 - Resultado da View Agentes	24
Tabela 16 - Resultado da View Eventos	25
Tabela 17 - Resultado da View Participantes_FIF	25
Tabela 18 - Tamanho máximo da tabela Agente	26
Tabela 19 - Tamanho máximo da tabela AgenteEvento	26
Tabela 20 - Tamanho máximo da tabela Cliente	26
Tabela 21 - Tamanho máximo da tabela ClienteEvento	27
Tabela 22 - Tamanho máximo da tabela Evento	27
Tabela 23 - Tamanho máximo da tabela Morada	27
Tabela 24 - Resultado Procedimento GetMoradaMaisCara	28
Tabela 25 - Resultado do Procedimento GetRendaBilheteria	29
Tabela 26 - Resultado do Procedimento GetClientesPrecoBilhete	29
Tabela 27 - Resultado do Procedimento GetIdadeMedia	30

Índice de Figuras

Figura 1 - Relacionamento Cliente-Evento	12
Figura 2 - Relacionamento Agente de evento-Evento	12
Figura 3 - Relacionamento Cliente-Morada	13
Figura 4 - Relacionamento entre Morada-Evento	13
Figura 5 - Esquema Conceptual	15
Figura 6 - Esquema Lógico	18
Figura 7 - Código SQL para informações dos Clientes	21
Figura 8 - Código SQL para informações de um evento que ocorreu numa dada data	21
Figura 9 - Código SQL para informações dos clientes que foram a um dado evento e têm mais e 24 anos	22
Figura 10 - Código SQL para informações dos Agentes que participaram de um evento e receberam mais de 300 euros	22
Figura 11 – Código SQL da View Clientes_PBA	23
Figura 12 – Código SQL da View Agentes	24
Figura 13 - Código SQL da View Eventos	24
Figura 14 - Código SQL da View Participantes_FIF	25
Figura 15 - Índices index_morada e index_evento	28
Figura 16 - Código SQL do Procedimento GetMoradaMaisCara	28
Figura 17 - Código SQL do Procedimento GetRendaBilheteria	29
Figura 18 - Código SQL do Procedimento GetClientesPrecoBilhete	29
Figura 19 - Código SQL do Procedimento GetIdadeMedia	30

1. Introdução

Numa sociedade onde a informação é fundamental e a manipulação dos dados desempenha um papel cada vez mais crucial, a organização e a gestão eficaz de eventos são essenciais para o desenvolvimento de uma cidade, como é o exemplo de Cidadela das Flores. Os eventos culturais, desportivos e sociais desempenham um papel fundamental na identidade da comunidade, atraindo visitantes e turistas, o que melhora a economia local e enriquece a cidade. Portanto, a criação de uma base de dados aplicada a um calendário de eventos de uma cidade torna-se uma ferramenta vital para administrar e organizar os eventos.

Este trabalho visa abordar a importância de estabelecer uma estrutura de base de dados SQL eficaz e adequada para a gestão de eventos de Cidadela das Flores. Iremos explorar e identificar os diversos aspetos e requisitos que a comunidade, os organizadores e possíveis administradores reivindicam, para assim estabelecer as diferentes entidades e relações. Serão implementados diferentes tipos de modelos, de modo a possibilitar uma melhor perceção do problema e da sua resolução.

1.1. Contextualização

A Dra. Maria Santos é a mais recente vereadora da cultura da “Cidadela das Flores”. É uma pequena cidade no Alto-Minho que dispõe de uma paisagem incrível, com uma comunidade vibrante e uma rica cultura.

Esta localidade foi elevada a cidade em 1980, e ao longo do tempo tem visto sua população crescer. Hoje, Cidadela das Flores possui cerca de 20 mil habitantes, sendo uma grande parcela desses habitantes jovens, que lá nascem ou mudam-se devido ao baixo preço do imobiliário. Como tal, Cidadela das Flores vê-se hoje, com uma grande vida noturna, assim como uma vasta diversidade de eventos locais, como festivais, feiras, apresentações culturais e atividades desportivas. Recentemente, por iniciativa local começou um festival anual “Funk in Flores” com o propósito de trazer vários artistas a atuarem no parque da cidade.

A vereadora tem notado recentemente algumas dificuldades na gestão e divulgação dos eventos que ocorrem em sua cidade, ela acredita que a sua cidade poderia beneficiar enormemente de um sistema de gestão e divulgação de eventos mais eficaz. Às vezes, ela percebe uma pouca afluência nos eventos. Muitas pessoas perdem eventos importantes porque não têm conhecimento dos mesmos com antecedência. Além disso, muitas vezes há sobreposição de eventos que o público gostaria de participar, mas é difícil planear as suas agendas sem um calendário universal e acessível. É por isso que a doutora Maria propôs a criação de uma base de dados que se concentre na gestão e divulgação do calendário de eventos de 'Cidadela das Flores'. Com uma plataforma centralizada, seria mais fácil para os organizadores de eventos compartilharem informações sobre datas, horários, locais e detalhes relevantes, assim como, informações sobre equipa técnica e o número de lugares ocupados. Isso também permitiria que os moradores, os visitantes, bem como os artistas, terem acesso a um calendário atualizado de eventos da cidade. Eles poderiam planear melhor o seu tempo, apoiar mais eventos locais, organizar de maneira mais eficiente e ajudar a fortalecer ainda mais

o senso de comunidade em 'Cidadela das Flores'. Seria uma vitória para todos: comerciantes, organizadores de eventos, residentes e, é claro, para a cidade, que poderia se destacar ainda mais como um destino cultural e turístico. Portanto, a Dra. Maria acredita que a implementação de uma base de dados para a gestão e divulgação do calendário de eventos é uma ideia que vale a pena explorar. Isso os ajudaria a aproveitar ao máximo tudo o que sua cidade tem a oferecer e a manter viva a rica tradição cultural e comunitária de 'Cidadela das Flores'.

1.2. Objetivos

Este projeto visa atingir os seguintes objetivos:

- **Melhorar a Gestão de Eventos:** Estabelecer um sistema de base de dados que otimize o agendamento e a organização de eventos, evitando conflitos de datas e redundâncias.
- **Aprimorar a Divulgação de Eventos:** Facilitar a divulgação de eventos para a comunidade, garantindo que todos os interessados tenham acesso às informações relevantes com antecedência.
- **Conhecer o Público-Alvo:** Coletar dados sobre o perfil dos participantes de eventos, permitindo a adaptação da oferta de eventos às preferências da comunidade.
- **Eficiência de Recursos:** Maximizar o uso de recursos, como locais e equipa técnica, garantindo que sejam alocados de maneira eficaz.

1.3. Análise da Viabilidade do Processo

Para garantir a viabilidade deste projeto, foram considerados os seguintes tópicos:

- **Recursos Disponíveis:** A cidade de 'Cidadela das Flores' conta com uma equipa de desenvolvimento e uma população ativa que pode contribuir para a criação e uso do sistema. Além disso, os recursos de hardware e software necessários estão disponíveis.
- **Benefícios para a Comunidade:** A implementação desse sistema traria benefícios significativos para a comunidade, como uma experiência mais organizada e rica em eventos culturais, o que justifica o investimento.
- **Proteção de Dados:** Garantir a segurança e a privacidade dos dados dos usuários será uma prioridade, garantindo a conformidade com regulamentos de proteção de dados.
- **Aceitação da Comunidade:** Envolver ativamente a comunidade na definição dos requisitos e na divulgação dos eventos é essencial para o sucesso do projeto.
- **Planeamento Financeiro:** O registo de custos totais, incluindo aluguel de locais, equipamentos e pagamento de artistas, permitirá um planeamento financeiro sólido e a manutenção da viabilidade a longo prazo.

1.4. Recursos e Equipa de Trabalho

Para a implementação desta base de dados a equipa de trabalho, constituída por 3 arquitetos de SQL e um gestor de eventos chamado Leandro Barbosa (que lidera a equipa) contratados pela Dra. Maria, com conhecimentos na área de gestão e criação de bases de dados, utilizou como recursos as ferramentas BrModelo, MySQL Server e MySQL Workbench. Para além disso, é necessário equipamentos e tecnologia para a infraestrutura do sistema (Hardware e Software), uma equipa técnica que engloba as áreas de limpeza e manutenção, assim como os artistas. Por fim, temos a população/turistas da cidade.

1.5. Plano de Execução do Projeto

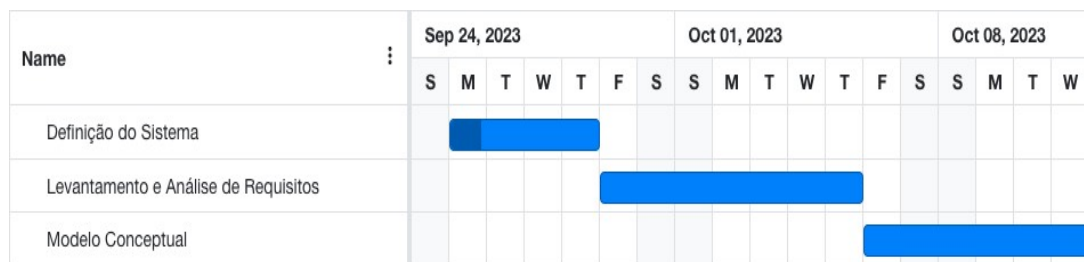


Tabela 1 - Diagrama de Gantt Parte 1

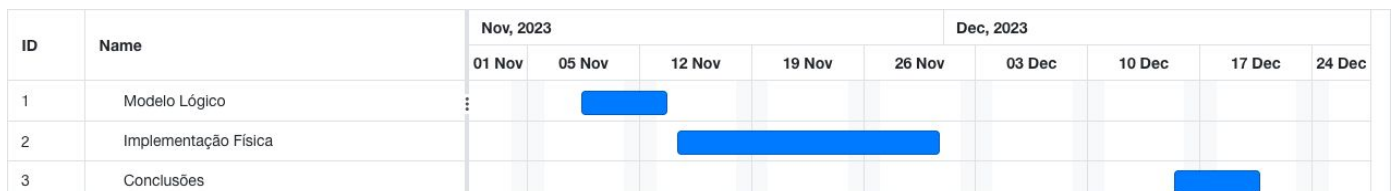


Tabela 2 - Diagrama de Gantt Parte 2

2. Levantamento e Análise de Requisitos

2.1. Método de Levantamento de Requisitos

Após várias queixas sobre a falta de conhecimento dos eventos por parte da população, os arquitetos, junto com o departamento da cultura, realizaram um questionário, tanto à população, quanto aos artistas e funcionários de manutenção dos espaços, de modo a identificar falhas e identificar quais os tipos de eventos prediletos. A distribuição dos questionários foi feita de forma manual ou via online usando a plataforma “*Google Forms*”. A distribuição dos questionários era feita ao final dos eventos que ocorreram, ou enviados diretamente à população da cidade via email. Posteriormente, esses questionários foram analisados pela equipa de arquitetos de SQL. Foram também realizadas reuniões entre a equipa de programadores e alguns organizadores de eventos da cidade, onde ocorreram análises de documentos e relatórios de eventos passados de forma a identificar os seus problemas, bem como a seu testemunho sobre esses mesmos eventos e os problemas pelo seu ponto de vista. Também foram acompanhados de perto pela equipa de programadores, os funcionários e equipas de manutenção dos espaços, artistas e organizadores de eventos, durante eventos que ocorreram no espaço de um mês.

2.2. Organização dos Requisitos

De modo a facilitar a organização dos requisitos, tendo em conta a estrutura imposta pela linguagem SQL, a equipa de programadores procedeu à organização dos requisitos levantados da seguinte forma: requisitos de descrição, requisitos de manipulação e requisitos de controlo.

2.2.1 Requisitos de Descrição

ID	Data/Hora	Requisito	Área/Vista	Fonte	Analista
1	27/09/2023 16:18	Todos os eventos devem estar registados, juntamente com a data correspondente, local correspondente, artistas e clientes, utilizando um método de identificação exclusivo.	Administração	Análise de Documentos	Leandro Barbosa
2	27/09/2023 16:19	Nenhum evento deve estar agendado para o mesmo dia, evitando conflitos de datas.	Administração	Análise de Documentos	Leandro Barbosa
3	27/09/2023 16:20	Todos os usuários devem estar registados na base de dados incluindo informações de identificação, como e-mail, número de telefone, idade e nome. Assim como os agentes de evento devem estar registados na base de dados incluindo sua designação, o nome e informações de contacto como e-mail e número de telefone	Tecnologia	Análise de Documentos	Leandro Barbosa
4	27/09/2023 16:20	Cada evento terá uma designação, data, lotação e localização pré-definidos.	Administração	Análise de Documentos	Leandro Barbosa
5	27/09/2023 16:26	Os custos totais, incluindo aluguer de locais e pagamento de artistas, pagamento dos clientes, entre outros, serão registados. Assim como o estado deste pagamento	Vendas	Análise de Documentos	Leandro Barbosa
6	27/09/2023 17:25	Todos os agentes de evento devem estar registados para trabalhar no evento	Tecnologia	Análise de Documentos	Leandro Barbosa
7	11/10/2023 14:00	Cada cliente pode estar associado a no máximo uma morada e cada morada pode estar associada a nenhum ou vários clientes	Tecnologia	Análise de Documentos	Leandro Barbosa
8	11/10/2023 14:05	Cada cliente pode estar associado a vários eventos e cada evento pode estar associado a vários clientes	Tecnologia	Análise de Documentos	Leandro Barbosa
9	27/09/2023 16:21	Serão realizados questionários para entender as preferências do público em relação aos eventos.	Marketing	Análise de Documentos	Leandro Barbosa
10	11/10/2023 14:15	Um evento está associado a apenas uma morada (localização) e uma morada está associada a nenhum ou vários eventos	Tecnologia	Análise de Documentos	Leandro Barbosa
11	11/10/2023 14:20	Um agente de evento pode estar associado a vários eventos e vice-versa	Tecnologia	Análise de Documentos	Leandro Barbosa

Tabela 3 - Requisitos de descrição

2.2.2. Requisitos de Exploração

ID	Data/Hora	Requisito	Área/Vista	Fonte	Analista
1	11/10/2023 13:39	Deve ser possível listar todos os clientes que vão a um certo evento	Tecnologia	Análise de Documentos	Leandro Barbosa
2	11/10/2023 13:39	Deve ser possível enumerar todos os clientes que moram em uma localidade específica	Tecnologia	Análise de Documentos	Leandro Barbosa
3	11/10/2023 13:43	Deve ser possível enumerar todos os clientes que pagaram uma certa quantia de dinheiro	Tecnologia	Análise de Documentos	Leandro Barbosa
4	13/10/2023 15:40	Deve ser possível enumerar todos os agentes de evento em um certo evento	Tecnologia	Análise de Documentos	Leandro Barbosa
5	13/10/2023 15:43	Deve ser possível listar a morada com o preço de aluguer mais alto para um evento	Tecnologia	Análise de Documentos	Leandro Barbosa

Tabela 4 - Requisitos de exploração

2.2.3. Requisitos de Controlo

ID	Data/Hora	Requisito	Área/Vista	Fonte	Analista
1	27/09/2023 16:21	A base de dados deve ser protegida de forma a evitar acesso não autorizado.	Tecnologia	Análise de Documentos	Leandro Barbosa
2	27/09/2023 16:25	Leandro Barbosa, Dra. Maria Santos e os engenheiros terão acesso a todo a base de dados (administrador), podendo adicionar eventos que são pedidos pelos clientes	Tecnologia	Análise de Documentos	Leandro Barbosa
3	27/09/2023 16:40	O sistema terá de funcionar apenas entre as 8h até às 20h	Tecnologia	Análise de Documentos	Leandro Barbosa
4	11/10/2023 13:43	Na base de dados apenas é inserido utilizadores que tenham idade acima de 18 anos	Tecnologia	Análise de Documentos	Leandro Barbosa

Tabela 5 - Requisitos de controlo

2.3. Análise e Validação Geral dos Requisitos

Através dos requisitos mencionados anteriormente (requisitos de controlo, exploração e descrição), é possível ter uma base para o sistema que está a ser implementado. A partir dos requisitos de controlo, é feita a definição de medidas de proteção, como autenticação, autorização e criptografia, para garantir a confidencialidade, integridade e disponibilidade dos dados no sistema. Por meio dos requisitos de manipulação, usuários interagem com o sistema e exploram suas funcionalidades, abrangendo elementos como a interface do usuário, as funcionalidades disponíveis e a integração com outras ferramentas ou sistemas. Finalmente, com os requisitos de descrição, é possível entender a funcionalidade do sistema, ao definir um conjunto de termos específicos usados no sistema para garantir uma compreensão uniforme entre todos os envolvidos no projeto.

3. Modelação Conceptual

3.1. Apresentação da abordagem de modelação realizada

Tendo por base os requisitos levantados anteriormente e as viabilidades já apresentadas, procedemos à explicação da abordagem de modelação realizada. Para isso, vamos apresentar esquemas que representam o conceito do funcionamento da base de dados aqui referida. Sendo assim, serão apresentadas descrições sobre cada entidade e seus atributos, bem como a identificação e descrição de cada relacionamento entre essas entidades.

3.2. Identificação e Caracterização das Entidades

Após o levantamento de requisitos, de modo que consigamos cumprir os mesmos, consideramos como imprescindíveis, as seguintes entidades: Cliente, Evento, Agente de Evento, Morada.

- **Evento:** Referente ao conjunto dos eventos que vão ser ou já foram realizados na Cidadela das Flores. É identificado por um Id de evento e contém informações acerca da sua designação, a lotação do mesmo e a data na qual é ou foi realizado;
- **Cliente:** Referente aos clientes que estão registados para participar nos eventos. Esta entidade é identificada por um Id de cliente e possui informações sobre o nome do cliente, o seu telefone, e-mail e idade;
- **Agente de Evento:** Referente aos Agentes de Evento, como por exemplo artistas ou funcionários necessários para o funcionamento do mesmo. É identificado por um Id de Agente e contém informações acerca do seu nome, designação (função desempenhada pelo agente), o seu e-mail e telefone;
- **Morada:** Referente à morada dos clientes e dos eventos. Esta entidade é identificada por um Id de morada e possui informações sobre a rua, localidade, código postal e porta.

3.3. Caracterização dos Relacionamentos

Cliente-Evento

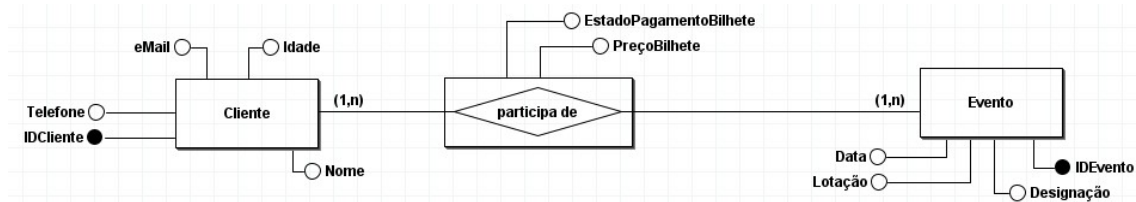


Figura 1 - Relacionamento Cliente-Evento

Relacionamento: Cliente participa de Evento

Descrição: Cada cliente pode participar de um, ou mais, eventos e vice-versa

Cardinalidade: Cliente (1, N) – Evento (1,N)

Atributos: Possui os atributos EstadoPagamentoBilhete e PreçoBilhete

Agente de evento-Evento

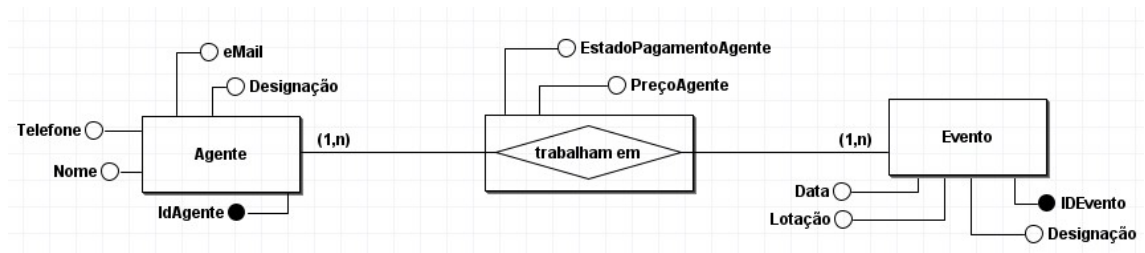


Figura 2 - Relacionamento Agente de evento-Evento

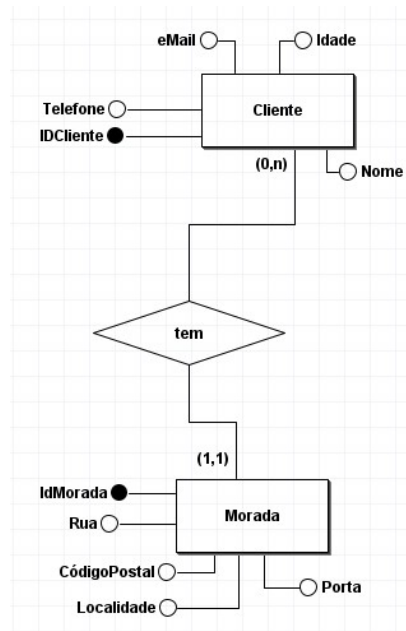
Relacionamento: Agente de evento trabalha em Evento

Descrição: Cada agente de evento trabalha em um, ou mais, eventos e vice-versa

Cardinalidade: Agente de evento (1, N) – Evento (1, N)

Atributos: Possui os atributos EstadoPagamentoAgente e PreçoAgente

Cliente-Morada



Relacionamento: Cliente tem morada

Descrição: Um cliente tem uma morada e uma morada tem nenhum ou vários clientes

Cardinalidade: Cliente (0, N) – Morada (1, 1)

Figura 3 - Relacionamento Cliente-Morada

Morada-Evento

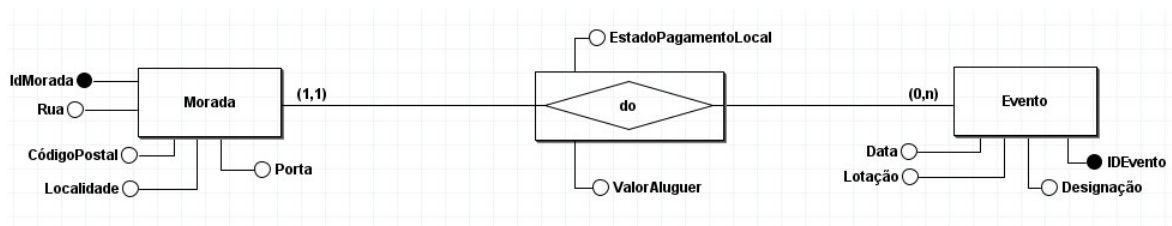


Figura 4 - Relacionamento entre Morada-Evento

Relacionamento: Morada do Evento

Descrição: Um Evento tem uma morada e uma morada pode ter nenhum ou vários eventos.

Cardinalidade: Morada (1, 1) – Evento (0, n)

Atributos: Possui os atributos EstadoPagamento e ValorAluguer.

3.4. Identificação e Caracterização da Associação dos Atributos com as Entidades e os Relacionamentos

- **Cliente**

Atributos	Tipo de Dados	Nulo	Composto	Multivalorado	Derivado	Chave Primária
IDCliente	INT	Não	Não	Não	Não	Sim
Nome	VARCHAR(50)	Não	Não	Não	Não	Não
Telefone	VARCHAR(20)	Não	Não	Não	Não	Não
eMail	VARCHAR(50)	Não	Não	Não	Não	Não
Idade	INT	Não	Não	Não	Não	Não

Tabela 6 - Caracterização dos Atributos da entidade Cliente

- **Evento**

Atributos	Tipo de Dados	Nulo	Composto	Multivalorado	Derivado	Chave Primária
IDEvento	INT	Não	Não	Não	Não	Sim
Designação	VARCHAR(50)	Não	Não	Não	Não	Não
Lotação	INT	Não	Não	Não	Não	Não
Data	DATETIME	Não	Não	Não	Não	Não

Tabela 7 - Caracterização dos Atributos da entidade Evento

- **Agente de evento**

Atributos	Tipo de Dados	Nulo	Composto	Multivalorado	Derivado	Chave Primária
IDAgente	INT	Não	Não	Não	Não	Sim
Nome	VARCHAR(50)	Não	Não	Não	Não	Não
Telefone	VARCHAR(20)	Não	Não	Não	Não	Não
eMail	VARCHAR(50)	Não	Não	Não	Não	Não
Designação	VARCHAR(50)	Não	Não	Não	Não	Não

Tabela 8 - Caracterização dos Atributos da entidade Agente

- Morada

Atributos	Tipo de Dados	Nulo	Composto	Multivalorado	Derivado	Chave Primária
IDMorada	INT	Não	Não	Não	Não	Sim
Rua	VARCHAR(50)	Não	Não	Não	Não	Não
Porta	VARCHAR(50)	Não	Não	Não	Não	Não
Localidade	VARCHAR(50)	Não	Não	Não	Não	Não
CódigoPostal	VARCHAR(10)	Não	Não	Não	Não	Não

Tabela 9 - Caraterização dos Atributos da entidade Morada

3.5. Apresentação e Explicação do Diagrama ER

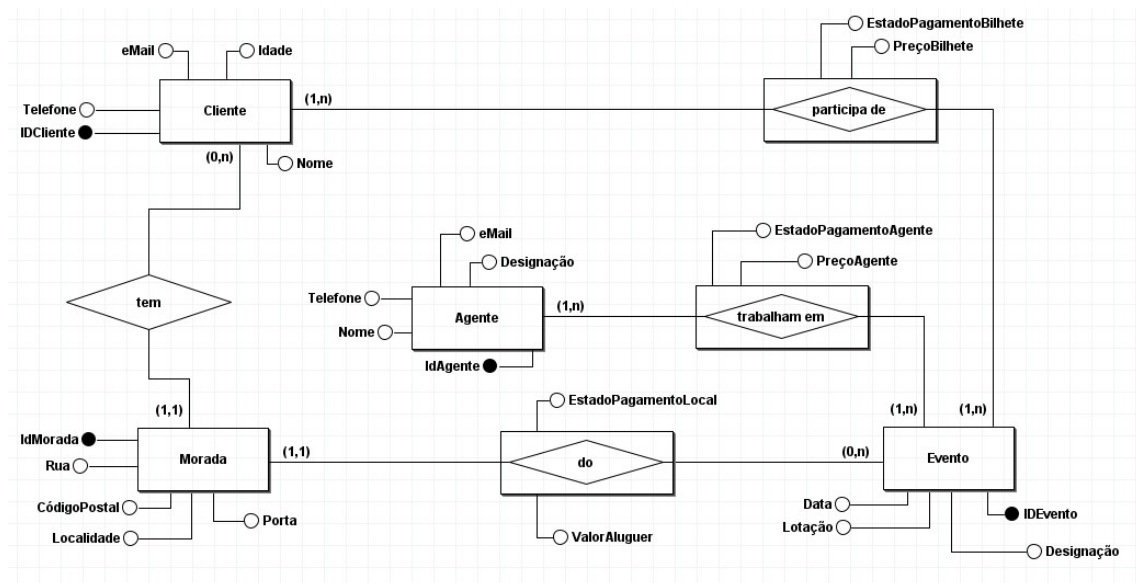


Figura 5 - Esquema Conceptual

Através do diagrama, podemos interpretar que um cliente pode participar de um ou mais eventos e vice-versa, um dado evento está associado a uma morada, uma morada pode ter eventos ou não, um agente de evento está associado a vários eventos e vice-versa, e um cliente está associado a uma morada, porem existem moradas sem clientes.

4. Modelação Lógico

4.1. Construção, Validação e Apresentação do Modelo de Dados Lógico

A elaboração do modelo lógico provem do já apresentado modelo conceptual. Nesta etapa, as entidades e relacionamentos são convertidos em tabelas. Para isso, são chaves primárias (Primary Key) os atributos que identificam uma entidade, e uma chave estrangeira (Foreign Key) os identificadores de outras entidades.

Entidades:

- **Morada:**
 - Primary Key: IDMorada (INT);
 - Atributos: Rua (VARCHAR(50));
Porta (VARCHAR(50));
CodigoPostal (VARCHAR(10));
Localidade (VARCHAR(50));

- **Cliente:**
 - Primary Key: IDCliente (INT);
 - Atributos: Nome (VARCHAR(75));
Telefone (VARCHAR(20));
Email (VARCHAR(50));
Idade (INT);
 - Foreign Key: IDMorada (INT);

- **Agente:**
 - Primary Key: IDAgente (INT);
 - Atributos: Nome (VARCHAR(50));
Email (VARCHAR(50));
Designação (VARCHAR(50));
Telefone (VARCHAR(20));
- **Evento:**
 - Primary Key: IDEvento (INT);
 - Atributos: Designação (VARCHAR(50));
Data (DATETIME);
Lotacao (INT);
ValorAluguer (INT);
EstadoPagamento (BINARY(1));
 - Foreign Key: idMorada (INT);

Relacionamentos:

- **AgenteEvento:**
 - Primary Key: IDAgente (INT);
IDEvento (INT);
 - Atributos: PrecoAgente (DECIMAL(8,2))
EstadoPagamento (BINARY(1));
 - Foreign Key: IDAgente (INT);
IDEvento (INT);
- **ClienteEvento:**
 - Primary Key: IDCliente (INT);
IDEvento (INT);
 - Atributos: PrecoPagamentoBilhete (DECIMAL(8,2))
EstadoPagamento (BINARY(1));
 - Foreign Key: IDCliente (INT);
IDEvento (INT);

Após a análise das entidades e relacionamento acima descritos e as respectivas cardinalidades entre os mesmos, chegamos ao seguinte esquema do Modelo Lógico:

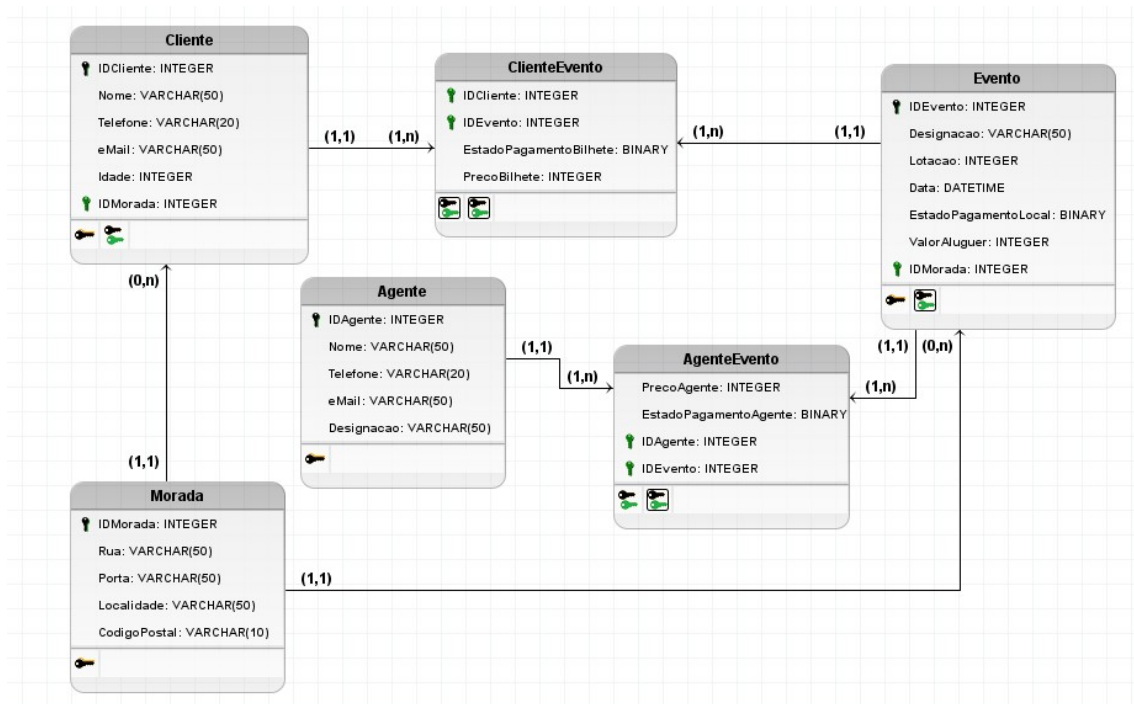


Figura 6 - Esquema Lógico

4.2. Normalização de Dados

Em relação à normalização de dados, as tabelas da Base de Dados encontram-se na Terceira Forma Normal (3FN). Os nossos atributos nunca são nulos, e são constituídos unicamente por um simples valor do seu domínio. Todos os atributos não-primos de cada tabela são totalmente dependentes da chave primária correspondente, além disso, estes são mutuamente independentes, não existindo assim dependências transitivas.

4.3. Validação do modelo com interrogações do utilizador

- O identificador, o nome e a idade de todos os clientes:

Π IDCLIENTE, NOME, IDADE (CLIENTE)

- O identificador e a designação de todos os eventos em uma determinada data:

Π IDEVENTO, DESIGNACAO (σ DATA='2022-06-15 15:00' (EVENTO))

- O nome, o identificador do evento e a idade de todos os clientes que vão a um certo evento e têm uma idade acima de 24 anos:

Π NOME, IDEVENTO, IDADE (σ IDEVENTO=2 \wedge IDADE>24 (CLIENTE \bowtie CLIENTE.IDCLIENTE=CLIENTEEVENTO.IDCLIENTE) CLIENTEEVENTO))

- O nome, a designação e a remuneração de todos os agentes de um certo evento que foram pagos com um valor acima de 300 euros:

Π NOME, DESIGNACAO, PRECOAGENTE (σ IDEVENTO = 1 \wedge PRECOAGENTE > 300 \wedge ESTADOPAGAMENTOAGENTE = 1 (AGENTE \bowtie AGENTE.IDAGENTE=AGENTEEVENTO.IDAGENTE AGENTEEVENTO))

5. Implementação Física

5.1. Tradução do Modelo Lógico para o Modelo Físico

Para realizar o modelo físico, recorreremos ao MySQL para a construção do código, na qual foi fortemente recomendado pelos docentes e onde possui um sistema SQL como interface. Nesse código são criados *schema's* que possuem as tabelas de cada entidade e relacionamento presente no esquema lógico desenvolvido anteriormente. Nessas tabelas são também enumerados os vários atributos e destacadas as chaves primárias, e chaves estrangeiras caso existam. Nesta etapa também é indicado o tipo das variáveis de cada atributo. Nos *schema's* são construídos, em primeiro lugar, tabelas de entidades que não possuem chaves estrangeiras (como o exemplo da tabela “Morada”), em segundo lugar, são descritas tabelas de entidades que possuem chaves estrangeiras (aqui se enquadra as tabelas “Cliente”, “Agente” e “Evento”, tabelas das quais possuem chaves estrangeiras da tabela “Morada”), e por fim são construídas as tabelas dos relacionamentos entre as entidades anteriormente definidas.

5.2. Tradução das interrogações do utilizador para SQL

- O identificador, o nome e a idade de todos os clientes:

```
SELECT IDCLIENTE, NOME, IDADE FROM CLIENTE;
```

Figura 7 - Código SQL para informações dos Clientes

IDCLIENTE	NOME	IDADE
1	Irenilde da Silva Costa	45
2	Richarlyson Vieira	19
3	Jasinto Leite	37
4	Jeremias Fonseca	52
5	Ana Maria das Dores	71
6	Fábio Martins	18
NULL	NULL	NULL

Tabela 10 - Informações dos Clientes

- O identificador e a designação de todos os eventos em uma determinada data:

```
SELECT IDEVENTO, DESIGNACAO FROM EVENTO  
WHERE DATA = '2022-06-15 15:00';
```

Figura 8 - Código SQL para informações de um evento que ocorreu numa dada data

IDEVENTO	DESIGNACAO
1	Funk in Flores
NULL	NULL

Tabela 11 - Informações de um evento que correu numa dada data

- O nome, o identificador do evento e a idade de todos os clientes que vão a um certo evento e têm uma idade acima de 24 anos:

```
SELECT NOME, IDEVENTO, IDADE
FROM CLIENTE AS C
INNER JOIN CLIENTEEVENTO AS CE
ON C.IDCLIENTE = CE.IDCLIENTE
WHERE IDEVENTO = 2 AND IDADE > 24;
```

Figura 9 - Código SQL para informações dos clientes que foram a um dado evento e têm mais e 24 anos

	NOME	IDEVENTO	IDADE	
	Jasinto Leite	2	37	

Tabela 12 - Informações dos clientes que foram a um determinado evento e têm mais de 24 anos

- O nome, a designação e a remuneração de todos os agentes de um certo evento que foram pagos com um valor acima de 300 euros:

```
SELECT NOME, DESIGNACAO, PRECOAGENTE
FROM AGENTE AS A
INNER JOIN AGENTEEVENTO AS AE
ON A.IDAGENTE = AE.IDAGENTE
WHERE IDEVENTO = 1 AND PRECOAGENTE > 300 AND ESTADOPAGAMENTOAGENTE = 1
```

Figura 10 - Código SQL para informações dos Agentes que participaram de um evento e receberam mais de 300 euros

	NOME	DESIGNACAO	PRECOAGENTE	
	Vincent Braun	DJ	1020.00	
	Robert Nielsen	Cantor	1300.00	

Tabela 13 - Informações dos Agentes que participaram de um evento e receberam mais de 300 euros

5.3. Definição e caracterização das vistas de utilização em SQL

- Vista de todos os clientes que são residentes no Porto, Braga ou Arouca, ordenados pela idade

```
CREATE VIEW Clientes_PBA
AS
    SELECT Nome, Idade
    FROM CLIENTE c
    INNER JOIN MORADA m ON c.IdMorada = m.IdMorada
    WHERE Localidade = 'Porto' OR Localidade = 'Braga' OR Localidade = 'Arouca'
    ORDER BY Idade ASC;

SELECT * FROM Clientes_PBA;
```

Figura 11 – Código SQL da View Clientes_PBA

	Nome	Idade
►	Fábio Martins	18
	Richarlyson Vieira	19
	Jasinto Leite	37

Tabela 14 - Resultado da View Clientes_PBA

- Vista de Todos os Agentes de Evento, bem como a sua designação, ordenado por id de cada agente

```
CREATE VIEW Agentes
AS
SELECT IdAgente AS "Nº de Agente",
       Nome,
       Designacao
FROM AGENTE
ORDER BY IdAgente ASC;

SELECT * FROM Agentes;
```

Figura 12 – Código SQL da View Agentes

	Nº de Agente	Nome	Designacao
▶	1	Vincent Braun	DJ
	2	Robert Nielsen	Cantor
	3	João Matheus	Técnico de Som
	4	Lewis Hamilton	Cantor
	5	Jonas Pistolas	Técnico de manutenção
	6	Marta Nielsen	DJ

Tabela 15 - Resultado da View Agentes

- Vista de todos os eventos, bem como a sua lotação e ordenados pela sua data

```
CREATE VIEW Eventos
AS
SELECT Designacao AS "Nome",
       Lotacao,
       Data
FROM EVENTO
ORDER BY Data DESC;

SELECT * FROM Eventos;
```

Figura 13 - Código SQL da View Eventos

	Nome	Lotacao	Data
►	Véspera de ano novo	5200	2022-12-31 18:00:00
	Halloween	2005	2022-10-31 18:00:00
	Festival de Verão	3900	2022-08-02 15:30:00
	Festival das Flores	2500	2022-07-24 14:00:00
	Funk in Flores	4700	2022-06-15 15:00:00
	Festival de Rancho	1630	2022-01-09 09:00:00

Tabela 16 - Resultado da View Eventos

- Vista de todos os clientes e agentes que participam no Funk in Flores, ordenados por nome

```
CREATE VIEW Participantes_FIF
AS
SELECT Nome, Email
FROM CLIENTE c
INNER JOIN CLIENTEEVENTO e ON c.IdCliente = e.IdCliente
WHERE e.IdEvento = '1'
UNION
SELECT Nome, Email
FROM AGENTE a
INNER JOIN AGENTEEVENTO e ON a.IdAgente = e.IdAgente
WHERE IdEvento = '1';

SELECT * FROM Participantes_FIF;
```

Figura 14 - Código SQL da View Participantes_FIF

	Nome	Email
►	Irenilde da Silva Costa	irenilde123@gmail.com
	Richarlyson Vieira	richas24@outlook.com
	Vincent Braun	vbrown@gmail.com
	Robert Nielsen	robert.nielsen@gmail.com

Tabela 17 - Resultado da View Participantes_FIF

5.4. Cálculo do espaço da base de dados

De forma a obter o espaço ocupado pela nossa Base de Dados, iremos calcular o espaço máximo que cada tabela poderá ocupar, isto devido ao uso de *Varchar's*, que variam consoante o número de caracteres utilizado. Cada Inteiro ocupa 4 bytes, um VARCHAR(N) ocupa N + 1 bytes, pois N < 255, se não seria N + 2 bytes, um Binário ocupa 1 byte, um DECIMAL(N,M) ocupa 8 Bytes e o DATETIME ocupa 5 bytes.

- **Agente**

Atributos	Tipo de Dados	Tamanho máximo
IDAgente	INT	4 bytes
Nome	VARCHAR(75)	76 bytes
Email	VARCHAR(50)	51 bytes
Designação	VARCHAR(50)	51 bytes
Telefone	VARCHAR(20)	21 bytes

Tabela 18 - Tamanho máximo da tabela Agente

- **Agenteevento**

Atributos	Tipo de Dados	Tamanho máximo
IDAgente	INT	4 bytes
IDEvento	INT	4 bytes
PreçoAgente	DECIMAL(8,2)	8 bytes
EstadoPagamentoAgente	BINARY(1)	1 byte

Tabela 19 - Tamanho máximo da tabela AgenteEvento

- **Cliente**

Atributos	Tipo de Dados	Tamanho máximo
IDCliente	INT	4 bytes
Nome	VARCHAR(75)	76 bytes
Email	VARCHAR(50)	51 bytes
Idade	INT	4 bytes
IDMorada	INT	4 bytes
Telefone	VARCHAR(20)	21 bytes

Tabela 20 - Tamanho máximo da tabela Cliente

- **Clienteevento**

Atributos	Tipo de Dados	Tamanho máximo
IDCliente	INT	4 bytes
IDEvento	VARCHAR(75)	76 bytes
PreçoBilhete	VARCHAR(50)	51 bytes
EstadoPagamentoBilhete	VARCHAR(50)	51 bytes

Tabela 21 - Tamanho máximo da tabela ClienteEvento

- **Evento**

Atributos	Tipo de Dados	Tamanho máximo
IDEvento	INT	4 bytes
Designação	VARCHAR(50)	51 bytes
Data	DATETIME	5 bytes
Lotação	INT	4 bytes
ValorAluguer	DECIMAL(8,2)	8 bytes
EstadoPagamentoLocal	BINARY(1)	1 byte
IDMorada	INT	4 bytes

Tabela 22 - Tamanho máximo da tabela Evento

- **Morada**

Atributos	Tipo de Dados	Tamanho máximo
IDMorada	INT	4 bytes
Rua	VARCHAR(50)	51 bytes
Porta	VARCHAR(50)	51 bytes
CódigoPostal	VARCHAR(50)	51 bytes
Localidade	VARCHAR(50)	51 bytes

Tabela 23 - Tamanho máximo da tabela Morada

Somando todas as tabelas, temos o resultado do tamanho (máximo) da base de dados inicial, que é 847 bytes. Fazendo uma estimativa de quantos registos serão adicionados a cada tabela por mês no primeiro ano de uso da base de dados, chegamos ao valor de 25 registos, ou seja, 300 registos nesse ano. Considerando que no início existiam 1500 registos, portanto, a nossa taxa de crescimento no primeiro ano será de 20%.

5.5. Indexação do Sistema de Dados

O uso de índices em um banco de dados é uma prática comum para melhorar o desempenho de consultas, mas a decisão de criar índices deve ser feita com cuidado, tendo em consideração o padrão de acesso aos dados e os requisitos específicos do seu sistema. Pelo uso excessivo e o número de registos nas tabelas, decidimos em criar os índices para

```
CREATE INDEX index_morada ON MORADA(IDMORADA);  
CREATE INDEX index_evento ON EVENTO(IDEVENTO);
```

Figura 15 - Índices index_morada e index_evento

Morada e Evento.

5.6. Procedimentos implementados

Implementamos diversos procedimentos, os que mais se destacam são:

- **Listar a morada com o preço de aluguer mais alto para um evento**

```
DELIMITER $$  
CREATE PROCEDURE GetMoradaMaisCara()  
) BEGIN  
    SELECT LOCALIDADE, RUA, PORTA, VALORALUGUER FROM MORADA AS M  
    INNER JOIN EVENTO AS E ON M.IDMORADA = E.IDMORADA  
    ORDER BY E.VALORALUGUER DESC  
    LIMIT 1;  
END  
$$  
  
CALL GetMoradaMaisCara();
```

Figura 16 - Código SQL do Procedimento GetMoradaMaisCara

LOCALIDADE	RUA	PORTA	VALORALUGUER
Cidadela das Flores	Avenida das Máquinas	2	7025.00

Tabela 24 - Resultado Procedimento GetMoradaMaisCara

- **Calcular a renda total de bilheteria de um dado evento**

```
DELIMITER $$
CREATE PROCEDURE GetRendaBilheteria(IN nome_evento VARCHAR(50))
BEGIN
    SELECT SUM(CE.PRECOBILHETE) AS "Lucro Total" FROM CLIENTEEVENTO AS CE
    INNER JOIN EVENTO AS E ON CE.IDEVENTO = E.IDEVENTO
    WHERE E.DESIGNACAO = nome_evento AND CE.ESTADOPAGAMENTO = '1';
END
$$
CALL GetRendaBilheteria('Funk in Flores');
```

Figura 17 - Código SQL do Procedimento GetRendaBilheteria

Lucro Total
420.00

Tabela 25 - Resultado do Procedimento GetRendaBilheteria

- **Enumerar todos os clientes que pagaram uma certa quantia ou menor**

```
DELIMITER $$
CREATE PROCEDURE GetClientesPrecoBilhete(IN quantia INT)
BEGIN
    SELECT NOME, PRECOBILHETE
    FROM CLIENTE AS C
    LEFT JOIN CLIENTEEVENTO AS E
    ON C.IDCLIENTE = E.IDCLIENTE
    WHERE PRECOBILHETE <= quantia
    ORDER BY PRECOBILHETE ASC;
END
$$
CALL GetClientesPrecoBilhete(200);
```

Figura 18 - Código SQL do Procedimento GetClientesPrecoBilhete

NOME	PRECOBILHETE
Richarlyson Vieira	120.00
Jeremias Fonseca	190.00
Jasinto Leite	200.00

Tabela 26 - Resultado do Procedimento GetClientesPrecoBilhete

- **Calcular a idade média dos clientes que vão a um dado certo evento**

```
DELIMITER $$
CREATE PROCEDURE GetIdadeMedia(IN nome_evento VARCHAR(50))
) BEGIN
    SELECT ROUND(AVG(C.IDADE)) AS "Idade Média" FROM CLIENTE AS C
    INNER JOIN CLIENTEEVENTO AS CE ON CE.IDCLIENTE = C.IDCLIENTE
    INNER JOIN EVENTO AS E ON E.IDEVENTO = CE.IDEVENTO
    WHERE E.DESIGNACAO = nome_evento;
END
$$
CALL GetIdadeMedia('Festival das Flores');
```

Figura 19 - Código SQL do Procedimento GetIdadeMedia

Idade Média
37

Tabela 27 - Resultado do Procedimento GetIdadeMedia

5.7. Plano de segurança e recuperação de dados

Este sistema está sujeito a uma perda de dados ou até mesmo à corrupção dos mesmos, proveniente de erros que possam ocorrer na base de dados.

De modo a minimizar os danos gerados por essas falhas, foi desenvolvido um plano de segurança. Neste plano, o backup incremental é realizado diariamente no fim do período laboral (20 horas) e segue o método “3-2-1”. Neste método, são criados três cópias do backup, onde uma delas permanece no servidor das instalações. As restantes duas cópias são transferidas para fora das instalações, das quais uma é enviada para a *Cloud*, disponibilizado por alguma empresa de *Cloud Storage Service*. A restante cópia é transferida para um dispositivo físico não conectado à rede.

Mensalmente, no primeiro dia útil de cada mês, um backup completo é realizado e guardado em um dispositivo físico não conectado à rede.

6. Conclusões e Trabalho Futuro

Com este projeto dado por terminado, podemos concluir que esta base de dados será capaz de ser uma grande ajuda perante aos problemas apresentados anteriormente, como por exemplo, uma melhor gestão da calendarização de eventos evitando assim conflitos ou redundâncias que podem vir a existir, além disso, asseguramos também uma melhor eficiência de recursos e com este sistema deixamos em aberto a possibilidade de expansão do mesmo, de forma a que possa acompanhar a evolução cultural e económica da Cidadela das Flores. Perante esta iminente expansão, é possível afirmar que poderiam ter sido implementadas mais funcionalidades, relativamente aos requisitos de controlo, e também o facto de que no cálculo do espaço ocupado pelo tipo de dados decimal, não foi feito da maneira mais exata, assim como o cálculo do espaço da base de dados inicial.

Em suma, para os problemas apresentados, esta base de dados está equipada para os resolver, ajudando assim no crescimento da cidade, contudo, como referido, o trabalho terá de ser continuo na mesma, de modo que seja possível continuar a contribuir para a prosperidade da cidade.

7. Referências

Belo, O., 2021. Bases de Dados Relacionais: Implementação com MySQL. FCA, Editora de Informática,

Connolly, T., Begg, C., 2014. Database Systems: A Practical Approach to Design, Implementation, and Management. Addison-Wesley, Global Edition,

MySQL. (2024). MySQL 8.0 Reference Manual: 11.7 Data Type Storage Requirements. Disponível em: <https://dev.mysql.com/doc/refman/8.0/en/storage-requirements.html> [Última consulta em 14 de janeiro de 2024]

8. Lista de Siglas e Acrónimos

BD	– Bases de dados
SQL	– Structured Query Language
ER	– Entity-Relationship
3FN	– Terceira Forma Normal
Dra.	– Doutora

9. Anexos

9.1. Script da criação da Base de dados

```
-- -----  
--  
-- CODIGO GERADOR DA BASES DE DADOS  
-- TABELAS E ATRIBUTOS  
--  
-- -----  
  
-- -----  
-- CRIACAO BASE DE DADOS  
-- -----  
CREATE SCHEMA IF NOT EXISTS GEVENTOS;  
  
-- -----  
-- SELECAO BASE DE DADOS  
-- -----  
USE GEVENTOS;  
  
-- -----  
-- CRIACAO TABELA - MORADA  
-- -----  
CREATE TABLE IF NOT EXISTS MORADA(  
  IDMORADA INT NOT NULL,  
  RUA VARCHAR(50) NOT NULL,  
  PORTA VARCHAR(50) NOT NULL,  
  CODIGOPOSTAL VARCHAR(50) NOT NULL,  
  LOCALIDADE VARCHAR(50) NOT NULL,  
  PRIMARY KEY(IDMORADA)  
);
```

```

-- -----
-- CRIACAO TABELA - CLIENTE
-- -----

CREATE TABLE IF NOT EXISTS CLIENTE(
IDCLIENTE INT NOT NULL,
NOME VARCHAR(75) NOT NULL,
EMAIL VARCHAR(50),
IDADE INT NOT NULL,
IDMORADA INT NOT NULL,
TELEFONE VARCHAR(20) NOT NULL,
PRIMARY KEY (IDCLIENTE),
FOREIGN KEY (IDMORADA) REFERENCES MORADA(IDMORADA)
);

-- -----
-- CRIACAO TABELA - AGENTE
-- -----

CREATE TABLE IF NOT EXISTS AGENTE(
IDAGENTE INT NOT NULL,
NOME VARCHAR(75) NOT NULL,
EMAIL VARCHAR(50) NOT NULL,
DESIGNACAO VARCHAR(50) NOT NULL,
TELEFONE VARCHAR(20) NOT NULL,
PRIMARY KEY (IDAGENTE)
);

-- -----
-- CRIACAO TABELA - EVENTO
-- -----

CREATE TABLE IF NOT EXISTS EVENTO(
IDEVENTO INT NOT NULL,
DESIGNACAO VARCHAR(50) NOT NULL,
DATA DATETIME NOT NULL,
LOTACAO INT NOT NULL,
VALORALUGUER DECIMAL(8,2) NOT NULL,
ESTADOPAGAMENTOLOCAL BINARY(1) NOT NULL,
IDMORADA INT NOT NULL,
PRIMARY KEY (IDEVENTO),
FOREIGN KEY (IDMORADA) REFERENCES MORADA(IDMORADA)
);

```

```

-----
-- CRIACAO TABELA - AGENTEEVENTO
-----

CREATE TABLE IF NOT EXISTS AGENTEEVENTO(
IDAGENTE INT NOT NULL,
IDEVENTO INT NOT NULL,
PRECOAGENTE DECIMAL(8,2) NOT NULL,
ESTADOPAGAMENTOAGENTE BINARY(1) NOT NULL,
PRIMARY KEY(IDAGENTE,IDEVENTO),
FOREIGN KEY(IDAGENTE) REFERENCES AGENTE(IDAGENTE),
FOREIGN KEY(IDEVENTO) REFERENCES EVENTO(IDEVENTO)
);

-----

-- CRIACAO TABELA - CLIENTEEVENTO
-----

CREATE TABLE IF NOT EXISTS CLIENTEEVENTO(
IDCLIENTE INT NOT NULL,
IDEVENTO INT NOT NULL,
PRECOBILHETE DECIMAL(8,2) NOT NULL,
ESTADOPAGAMENTOBILHETE BINARY(1) NOT NULL,
PRIMARY KEY(IDCLIENTE,IDEVENTO),
FOREIGN KEY(IDCLIENTE) REFERENCES CLIENTE(IDCLIENTE),
FOREIGN KEY(IDEVENTO) REFERENCES EVENTO(IDEVENTO)
);

-----

-- SELECAO - CONTEUDO DAS TABELAS
-----

SELECT * FROM MORADA;
SELECT * FROM CLIENTE;
SELECT * FROM AGENTE;
SELECT * FROM EVENTO;
SELECT * FROM AGENTEEVENTO;
SELECT * FROM CLIENTEEVENTO;

```


9.2. Script do Povoamento da Base de Dados

```
-- -----  
--  
-- CODIGO DO POVOAMENTO  
-- BASES DE DADOS  
--  
-----  
  
-- -----  
-- SELECIONA A BASE DE DADOS  
-- -----  
USE GEVENTOS;  
  
-- -----  
-- POVOAMENTO MORADA  
-----  
  
insert into MORADA values(1,'Rua das Lágrimas','34','4566-224','Cidadela das Flores');  
insert into MORADA values(2,'Rua das Acácias','25','4221-525','Porto');  
insert into MORADA values(3,'Avenida de São Romeo','42','4624-367','Braga');  
insert into MORADA values(4,'Rua das Andorinhas','12','3115-312','Cidadela das Flores');  
insert into MORADA values(5,'Avenida dos Pandas','16','4114-323','Cidadela das Flores');  
insert into MORADA values(6,'Rua dos sapos','156','4762-578','Cidadela das Flores');  
insert into MORADA values(7,'Rua Comendador Paulo Freitas','2','4127-254','Bragança');  
insert into MORADA values(8,'Rua da Maria das Dores','564','4234-691','Localidade da Maria  
das Dores');  
insert into MORADA values(9,'Rua do Pontinho','12','4982-393','Arouca');  
insert into MORADA values(10,'Rua da Dona Acácia','68','4652-271','Cidadela das Flores');  
insert into MORADA values(11,'Avenida das Máquinas','2','4989-898','Cidadela das Flores');  
insert into MORADA values(12,'Rua da Madeira','87','4234-567','Cidadela das Flores');  
  
-- -----  
-- POVOAMENTO CLIENTE  
-----  
  
insert into CLIENTE values(1,'Irenilde da Silva  
Costa','irenilde123@gmail.com',45,1,'912345092');  
insert into CLIENTE values(2,'Richarlyson Vieira','richas24@outlook.com',19,2,'914551234');  
insert into CLIENTE values(3,'Jasinto Leite','jasleite@hotmail.com',37,3,'943323011');  
insert into CLIENTE values(4,'Jeremias Fonseca','wells@sapo.pt',52,7,'961961961');  
insert into CLIENTE values(5,'Ana Maria das  
Dores','anamariadores08@gmail.com',71,8,'928008543');
```

```
insert into CLIENTE values(6,'Fábio Martins','masterpt2000@yahoo.com',18,9,'914487245');
```

```
-- -----  
-- POVOAMENTO EVENTO  
-- -----
```

```
insert into EVENTO values(1,'Funk in Flores','2022-06-15 15:00:00',4700,1500,1,4);  
insert into EVENTO values(2,'Festival das Flores','2022-07-24 14:00:00',2500,1025,1,5);  
insert into EVENTO values(3,'Halloween','2022-10-31 18:00:00',2005,1310,1,6);  
insert into EVENTO values(4,'Festival de Rancho','2022-01-09 09:00:00',1630,1500,0,10);  
insert into EVENTO values(5,'Véspera de ano novo','2022-12-31 18:00:00',5200,7025,1,11);  
insert into EVENTO values(6,'Festival de Verão','2022-08-02 15:30:00',3900,6500,1,12);
```

```
-- -----  
-- POVOAMENTO AGENTE  
-- -----
```

```
insert into AGENTE values(1,'Vincent Braun','vbraun@gmail.com','DJ','943323011');  
insert into AGENTE values(2,'Robert Nielsen','robert.nielsen@gmail.com','Cantor','956755263');  
insert into AGENTE values(3,'João Matheus','jmath@hotmail.com','Técnico de  
Som','967241145');  
insert into AGENTE values(4,'Lewis  
Hamilton','maxverstappen@hotmail.com','Cantor','968790000');  
insert into AGENTE values(5,'Jonas Pistolas','jonas10benfica@gmail.com','Técnico de  
manutenção','975431987');  
insert into AGENTE values(6,'Marta Nielsen','marta@yahoo.com','DJ','981234567');
```

```
-- -----  
-- POVOAMENOT - AGENTEEVENTO  
-- -----
```

```
insert into AGENTEEVENTO values(1,1,1020,1);  
insert into AGENTEEVENTO values(2,1,1300,1);  
insert into AGENTEEVENTO values(3,2,750,1);  
insert into AGENTEEVENTO values(4,3,980,1);  
insert into AGENTEEVENTO values(5,6,700,1);  
insert into AGENTEEVENTO values(6,5,1200,1);
```

```
-- -----  
-- POVOAMENTO TABELA - CLIENTEEVENTO  
-- -----
```

```
insert into CLIENTEEVENTO values(1,1,300,1);  
insert into CLIENTEEVENTO values(2,1,120,1);
```

```

insert into CLIENTEEVENTO values(3,2,200,1);
insert into CLIENTEEVENTO values(4,3,190,0);
insert into CLIENTEEVENTO values(5,5,302,1);
insert into CLIENTEEVENTO values(6,6,500,1);

```

```

SELECT * FROM AGENTE;
SELECT * FROM AGENTEEVENTO;
SELECT * FROM CLIENTE;
SELECT * FROM CLIENTEEVENTO;
SELECT * FROM EVENTO;
SELECT * FROM MORADA;

```

9.3. Script dos Procedimentos da Base de Dados

```

-- -----
--
-- PROCEDIMENTOS
--
-- -----

-- -----
-- SELECIONA GEVENTOS
-- -----

USE GEVENTOS

-- -----
-- Deve ser possível listar todos os clientes que vão a um certo evento
-- -----

DELIMITER $$
CREATE PROCEDURE GetInscritos(IN nome_evento VARCHAR(50))
BEGIN
    SELECT NOME,IDADE
        FROM CLIENTE AS C
            INNER JOIN CLIENTEEVENTO AS CE ON C.IDCLIENTE =
CE.IDCLIENTE
                INNER JOIN EVENTO as E ON CE.IDEVENTO =
E.IDEVENTO
        WHERE DESIGNACAO = nome_evento
        ORDER BY NOME ASC;

```

```

END
$$
CALL GetInscritos('funk in flores');

-- -----
-- Caso seja preciso remover o procedimento
-- -----

DROP PROCEDURE GetInscritos;

-- -----
-- Deve ser possível enumerar todos os clientes de uma localidade específica
-- -----

DELIMITER $$
CREATE PROCEDURE GetResidentes(IN nome_morada VARCHAR(50))
BEGIN
    SELECT NOME,IDADE
        FROM CLIENTE AS C
            LEFT JOIN MORADA AS M
                ON C.IDMORADA = M.IDMORADA
        WHERE LOCALIDADE = nome_morada
        ORDER BY NOME ASC;
END
$$
CALL GetResidentes('Cidadela das Flores')

-- -----
-- Caso seja preciso remover o procedimento
-- -----

DROP PROCEDURE GetResidentes;

-- -----
-- Deve ser possível enumerar os clientes que pagaram menor ou igual a uma certa quantia
-- -----

DELIMITER $$
CREATE PROCEDURE GetClientesPrecoBilhete(IN quantia INT)
BEGIN
    SELECT NOME,PRECIBILHETE
        FROM CLIENTE AS C
            LEFT JOIN CLIENTEEVENTO AS E
                ON C.IDCLIENTE = E.IDCLIENTE

```

```

        WHERE PRECOBILHETE <= quantia
        ORDER BY PRECOBILHETE ASC;
END
$$
CALL GetClientesPrecoBilhete(200);

-- -----
-- Caso seja preciso remover o procedimento
-- -----

DROP PROCEDURE GetClientesPrecoBilhete;

-- -----
-- Deve ser possível enumerar todos os agentes de evento de um evento
-- -----

DELIMITER $$
CREATE PROCEDURE GetAgentes(IN nome_evento VARCHAR(50))
BEGIN
    SELECT NOME,EMAIL FROM AGENTE AS A
        INNER JOIN AGENTEEVENTO AS AE ON A.IDAGENTE = AE.IDAGENTE
        INNER JOIN EVENTO AS E ON AE.IDEVENTO = E.IDEVENTO
        WHERE E.DESIGNACAO = nome_evento
        ORDER BY NOME ASC;
END
$$

CALL GetAgentes('Funk in Flores');

-- -----
-- Caso seja preciso remover o procedimento
-- -----

DROP PROCEDURE GetAgentes;

-- -----
-- Deve ser possível listar a morada com o preço de aluguer mais alto para um evento
-- -----

DELIMITER $$
CREATE PROCEDURE GetMoradaMaisCara()
BEGIN
    SELECT LOCALIDADE,RUA,PORTA,VALORALUGUER FROM MORADA AS M
        INNER JOIN EVENTO AS E ON M.IDMORADA = E.IDMORADA

```

```

ORDER BY E.VALORALUGUER DESC
LIMIT 1;
END
$$

CALL GetMoradaMaisCara();

-- -----
-- Caso seja preciso remover o procedimento
-- -----
DROP PROCEDURE GetMoradaMaisCara;

-- -----
-- Procedimento utilizado para saber qual o lucro total de um certo evento
-- -----
DELIMITER $$
CREATE PROCEDURE GetRendaBilheteria(IN nome_evento VARCHAR(50))
BEGIN
    SELECT SUM(CE.PRECOBILHETE) AS "Lucro Total" FROM CLIENTEEVENTO AS
CE
    INNER JOIN EVENTO AS E ON CE.IDEVENTO = E.IDEVENTO
    WHERE E.DESIGNACAO = nome_evento AND CE.ESTADOPAGAMENTOBILHETE = '1';
END
$$
CALL GetRendaBilheteria('Funk in Flores');

-- -----
-- Caso seja preciso remover o procedimento
-- -----
DROP PROCEDURE GetRendaBilheteria;

-- -----
-- Procedimento utilizado para saber qual a idade média dos clientes que vão a um certo
evento
-- -----
DELIMITER $$
CREATE PROCEDURE GetIdadeMedia(IN nome_evento VARCHAR(50))
BEGIN
    SELECT ROUND(AVG(C.IDADE)) AS "Idade Média" FROM CLIENTE AS C
    INNER JOIN CLIENTEEVENTO AS CE ON CE.IDCLIENTE = C.IDCLIENTE

```

```

        INNER JOIN EVENTO AS E ON E.IDEVENTO = CE.IDEVENTO
        WHERE E.DESIGNACAO = nome_evento;
END
$$
CALL GetIdadeMedia('Festival das Flores');

-- -----
-- Caso seja preciso remover o procedimento
-- -----

DROP PROCEDURE GetIdadeMedia;

-- -----
-- Indexes
-- -----

CREATE INDEX index_morada ON MORADA(IDMORADA);
CREATE INDEX index_evento ON EVENTO(IDEVENTO);

```