

**Universidade do Minho**

Licenciatura em Ciências da Computação

**Unidade Curricular de**

**Bases de Dados**

Ano Letivo de 2023/2024

**Calendário de Eventos na cidade de Braga**

**Diogo Coelho da Silva (a100092), Pedro Miguel Ramôa Oliveira (a97686)**

Janeiro, 2024

**BD**

|  |  |
| --- | --- |
| Data de Receção |  |
| Responsável |  |
| Avaliação |  |
| Observações |  |

**Calendário de Eventos na Cidade de Braga**

**Diogo Coelho da Silva (a100092), Pedro Miguel Ramôa Oliveira (a97686)**

Janeiro, 2024

<<Mês, Ano>>

# Resumo

Este relatório foi realizado no âmbito da unidade curricular de Bases de Dados, que tinha como objetivo principal criar um sistema de gestão de base de dados com o maior foco a incidir na análise, no desenho, modelação, arquitetura e implementação de um sistema deste tipo.

O projeto que nos vamos basear é um projeto fictício que foi criado e imaginado derivando de muitas situações em que muitas empresas ficaram depois da pandemia que assolou o país.

O projeto em questão surgiu como resposta ao declínio das atividades festivas e culturais em Braga, provocada pela pandemia global de 2019, bem como pelo encerramento de empresas voltadas para a gestão de eventos na cidade. Com o propósito de reavivar o espírito festivo de Braga e apoiar a sua economia local, um grupo de 3 jovens liderado por Martim Santos, entregou-nos a tarefa e o desafio de criar um Sistema de Bases de Dados para a Gestão de Eventos em Braga.

Os objetivos primordiais deste SBDG consistem em promover eventos, festividades e atrações turísticas na cidade, bem como em facilitar o acesso dos turistas a informação detalhadas sobre esses eventos. A administração da cidade também é beneficiada, pois este sistema auxilia na gestão eficaz de eventos e coleta dados essenciais sobre a participação de turistas e a população de eventos e atrações.

Numa primeira fase deste projeto, foram estabelecidos os requisitos que a nossa base de dados deveria comportar. Após a recolha dos requisitos e uma análise cuidada aos mesmos, foi altura de começar a realizar um modelo conceptual, em que foram estabelecidas entidades, derivadas dos requisitos anteriormente recolhidos, e os relacionamentos existentes entre si. Foi feita no fim destes dois passos, a validação do nosso modelo conceptual, e assim foi obtido o aval para avançar para a próxima etapa da criação do SGBD.

Após uma primeira avaliação do trabalho prático, foi retomado o mesmo com a conversão do modelo conceptual, anteriormente realizado, num modelo lógico. Neste passo foram revistos todos os relacionamentos e a criação das respetivas tabelas. De seguida, o mesmo modelo lógico foi traduzido para o sistema de gestão de bases de dados que foi escolhido, no nosso caso, o MySQL. Nesta fase foram feitas interrogações à nossa base de dados para verificar que se verificavam os resultados pretendidos e foi feito também um plano de segurança e recuperação de dados.

**Área de Aplicação:** Desenho, arquitetura, implementação e manipulação de um Sistema de Base de Dados.

**Palavras-Chave:** Bases de Dados, Bases de Dados Relacionais, Recolha de Requisitos, Análise de Requisitos, Modelo Conceptual, Entidades, Relacionamentos, Atributos, MySQL, Segurança e Recuperação de dados, Sistema de Gestão de base de dados, Modelo Lógico, Implementação Física, Interrogações.

**Índice**

[Resumo i](#_Toc149919009)

[Índice ii](#_Toc149919010)

[Índice de Figuras iii](#_Toc149919011)

[Índice de Tabelas v](#_Toc149919012)

[1. Introdução 1](#_Toc149919013)

[1.1. Contextualização 1](#_Toc149919014)

[1.2. Fundamentação 1](#_Toc149919015)

[1.3. Objetivos 2](#_Toc149919016)

[1.4. Viabilidade 3](#_Toc149919017)

[1.5. Recursos 3](#_Toc149919018)

[1.6. Equipa de Trabalho 4](#_Toc149919019)

[1.7. Plano de Execução 5](#_Toc149919020)

[1.8. Revisão e Aprovação 6](#_Toc149919021)

[2. Definição de Requisitos 7](#_Toc149919022)

[2.1. Método de levantamento e de análise de requisitos adotado 7](#_Toc149919023)

[2.2. Análise e Organização 10](#_Toc149919024)

[2.3. Análise e validação geral dos requisitos 13](#_Toc149919025)

[3. Modelação Concetual 14](#_Toc149919026)

[3.1. Apresentação da abordagem de modelação realizada 14](#_Toc149919027)

[3.2. Identificação e caracterização das entidades 14](#_Toc149919028)

[3.3. Identificação e caracterização dos relacionamentos 16](#_Toc149919029)

[3.4. Identificação e caracterização da associação dos atributos com as entidades e relacionamentos 19](#_Toc149919030)

[3.5. Apresentação e explicação do diagrama ER produzido 22](#_Toc149919031)

[4. Conclusões e Trabalho Futuro 54](#_Toc149919032)

[Referências 55](#_Toc149919033)

[Lista de Siglas e Acrónimos 56](#_Toc149919034)

**Índice de Figuras**

[Figura 1- Diagrama Gantt **Erro! Marcador não definido.**](#_Toc149918465)

[Figura 2 - Representação conceptual feita no software “BrModelo” da entidade “Evento” **Erro! Marcador não definido.**](#_Toc149918466)

[Figura 3 - Representação conceptual feita no software “BrModelo” da entidade “Bilhete” **Erro! Marcador não definido.**](#_Toc149918467)

[Figura 4 - Relacionamento Evento-Atividade **Erro! Marcador não definido.**](#_Toc149918468)

[Figura 5 - Relacionamentos Evento-Bilhete e Atividade-Bilhete **Erro! Marcador não definido.**](#_Toc149918469)

[Figura 6 - Relacionamento Bilhete-Utilizador **Erro! Marcador não definido.**](#_Toc149918470)

[Figura 7 - Relacionamento Atividade-Artista **Erro! Marcador não definido.**](#_Toc149918471)

[Figura 8 - Representação final do modelo conceptual do SGBD 22](#_Toc149918472)

# Índice de Tabelas

[Tabela 1 - Requisitos Gerais do SGBD **Erro! Marcador não definido.**](#_Toc149918473)

[Tabela 2 - Requisitos Descrição do SGBD. **Erro! Marcador não definido.**](#_Toc149918474)

[Tabela 3 - Requisitos Manipulação do SGBD **Erro! Marcador não definido.**](#_Toc149918475)

[Tabela 4 - Requisitos Controlo do SGBD **Erro! Marcador não definido.**](#_Toc149918476)

[Tabela 5 - Identificação e caracterização das entidades **Erro! Marcador não definido.**](#_Toc149918477)

[Tabela 6 - Identificação e caracterização dos relacionamentos **Erro! Marcador não definido.**](#_Toc149918478)

[Tabela 7 - Identificação e caracterização dos atributos com as entidades e relacionamentos **Erro! Marcador não definido.**](#_Toc149918479)

1. Introdução
   1. Contextualização

A cidade em questão é um importante destino turístico localizado numa região de beleza natural exuberante. Com uma população de cerca de aproximadamente 193 mil habitantes, esta cidade atrai visitantes de todo o mundo devido à sua rica cultura, património histórico e eventos variados. A sua importância turística reflete-se na contribuição significativa para a economia local, através da indústria do turismo, que abrange setores como a hotelaria, alimentação, lazer e comércio.

A cidade de Braga há muito é reconhecida pela sua juventude e pelo empenho constante em promover uma série de eventos culturais e festivos ao longo do ano. Contudo, esse ímpeto festivo foi interrompido em 2019, quando o mundo foi assolado por uma pandemia. Durante quase dois anos, a cidade viu-se privada de eventos, o que culminou num desinteresse por parte das entidades organizadoras em retomar tais atividades. Além disso, diversas empresas incumbidas da gestão de eventos e festividades na cidade acabaram por encerrar as suas operações, em virtude das dificuldades enfrentadas durante esse período desafiador.

Diante deste quadro, um trio de jovens decidiu que esta maré não podia continuar, e assim, uniram-se nos esforços de reavivar o espírito festivo de Braga. Martim Santos, nascido em 15 de março de 1996, natural da própria cidade, foi o principal impulsionador da iniciativa. Motivado pelo falecimento do seu avô, Albertino Faria, que figurava entre os pioneiros na organização destas festividades na cidade, comprometeu-se a reerguer o fervor festivo da localidade. Para tal desígnio, convidou os seus amigos de longa data, Francisco Ferreira, de 28 anos, e Ana Rodrigues, de 27 anos, a se juntarem a ele nesta aventura. Nasceu assim a empresa “Bracara Eventos”.

* 1. Fundamentação

A justificativa para implementação deste sistema de base de dados é baseada numa série de desafios enfrentados nos últimos anos, que afetaram a capacidade de organizar eventos e festas na cidade de forma eficaz. Primeiramente, a interrupção desses eventos levou ao declínio gradual do hábito de organizá-los, tornando a retoma da normalidade uma tarefa mais complicada. Além disso, ao longo do tempo, grupos e empresas envolvidos na organização destes eventos enfrentaram dificuldades para acompanhar os avanços tecnológicos que poderiam proporcionar maior eficiência e facilidade na gestão dos mesmos.

O aumento da população na cidade, juntamente com o crescente número de turistas que a visitam, aumentou a complexidade de organizar e divulgar os eventos de maneira satisfatória. Consequentemente, a tarefa de coordenar e comunicar eficazmente essas atividades tornou-se praticamente impossível.

Para abordar essas situações e recuperar o tempo perdido, o grupo decidiu realizar a implementação de um sistema de gestão bases de dados. Este sistema permitirá que os mesmos armazenem antecipadamente as informações necessárias, possibilitando um maior controle sobre cada evento festivo e turístico na cidade.

* 1. Objetivos

O grupo, comandado pela liderança, experiência e conhecimento adquiridos pelo Martim ao longo da sua vida com o seu falecido avô, estabeleceu um conjunto de objetivos que pretendem alcançar com o SGBD que estão prestes a criar, dos quais se destacam:

1. Reestruturar e aprimorar o modelo de gestão desses eventos, visando melhorar substancialmente a sua capacidade de organização e gerenciamento.
2. Aprimorar a eficácia e a otimização da gestão de cada evento.
3. Aumentar a eficácia na divulgação e promoção desses eventos na cidade.
4. Facilitar a pesquisa de eventos com base em diversos critérios, tais como data, tipo de evento, localização, etc…
5. Aumentar receitas a partir de bilhetes vendidos.
6. Registar os eventos de forma detalhada
7. Fornecer informações relevantes sobre cada evento, como programação, preço, disponibilidade de ingressos, etc…
8. Oferecer uma interface *user friendly* para os utilizadores acessem o calendário de eventos e interagirem com as informações disponíveis.

Estes objetivos visam aprimorar a gestão e a visibilidade dos eventos da cidade, proporcionando um serviço mais eficiente e acessível tanto para os residentes, como para os turistas e para todos os intervenientes na realização destes eventos.

* 1. Viabilidade

A viabilidade deste projeto é prometedora, considerando o potencial económico que a cidade oferece. O grupo está confiante de que ao implementar um sistema mais eficiente e moderno para a gestão de eventos na cidade de Braga, poderá alcançar os seguintes benefícios:

1. Recuperar aproximadamente 50% das perdas decorrentes do período de pandemia e confinamento. Este objetivo não apenas cobrirá os custos associados à criação deste Sistema de Gestão de Bases de Dados, mas também gerará lucro adicional.
2. Manter um acompanhamento constante do número de bilhetes vendidos para cada evento pago, bem como o lucro associado a cada um deles.
3. Monitorar o fluxo de pessoas em cada evento, o que contribuirá para uma gestão mais eficiente e segura.
4. Identificar e compreender as tendências e preferências dos residentes da cidade e dos turistas em relação aos eventos, permitindo a adaptação das atividades à demanda e ao gosto do público.

A viabilidade deste projeto é prometedora, considerando o potencial económico que a cidade oferece.

* 1. Recursos

A equipa de desenvolvimento deste SGBD delineou e apresentou, juntamente com o grupo, um conjunto de recursos que pensaram serem necessários para a implementação da mesma. Estes recursos foram divididos em dois sub-recursos: recursos materiais e recursos humanos.

**Recursos Humanos:**

* O grupo Bracara Eventos, constituído por Martim Santos, Francisco Ferreira e Ana Rodrigues
* Um Engenheiro de Bases de Dados
* Um Arquiteto de Bases de Dados
* Um especialista em segurança de redes
* Um Administrador de Bases de Dados
* 1 Equipa de 2 analistas
* Desenvolvedor de *Software*
* Artistas
* Staff
* Cidadãos
* Câmara Municipal de Braga

**Recursos Materiais:**

* *Hardware*:
  + 1 Servidor
  + 6 Postos de Venda de Bilhetes
  + 2 Posto de Informação
* *Software*:
* SGBD
* Aplicação para vendas de bilhetes e informação sobre os eventos
* BrModelo
* *SQL Workbench*
  1. Equipa de Trabalho

A equipa de trabalho vai ser divida em três categorias: Pessoal Interno, Pessoal Externo e Outros. O pessoal interno a este projeto vão ser os intervenientes responsáveis pela idealização e criação do mesmo. O pessoal externo vão ser todas as pessoas que vão ser contratadas para a realização deste projeto. Dito isto podemos distribuir a equipa da seguinte forma:

**Pessoal Interno:**

* Martim Santos
* Francisco Ferreira
* Ana Rodrigues

**Pessoal Externo:**

* Arquiteto de Bases de Dados
* Engenheiro de Bases de Dados
* Administrador de Bases de Dados
* Desenvolvedor *Software*
* Equipa de *Marketing*
* Analistas:
  + Diogo Coelho da Silva
  + Pedro Miguel Ramôa Oliveira

**Outros:**

* Cidadãos e turistas selecionados
* Camara Municipal de Braga

Podemos, agora, explicitar um pouco sobre o que cada categoria da equipa de trabalho vai trabalhar sobre.

O pessoal interno vai ser responsável pela gestão e organização dos eventos, como por exemplo contratar artistas, fazer o atendimento a novos clientes, validação de serviços, entre outros tópicos relacionados com esta área.

O pessoal externo vai ser responsável por implementar os pedidos do pessoal interno. O Arquiteto de Bases de Dados vai analisar os requisitos recolhidos anteriormente e esboçar um modelo conceptual do SBGB, que mais tarde irá ser utilizado pelo Engenheiro de Bases de Dados para converter este esboço conceptual para um modelo físico. O Administrador de Bases de Dados vai realizar a manutenção regular necessária para um projeto desta dimensão. O desenvolvedor de software vai criar uma aplicação, para que os cidadãos e turistas possam muito resumidamente consultar informações sobre os eventos e também realizar um conjunto de operações como por exemplo comprar bilhetes para eventos selecionados.

A adicionar, vai também ser contratada uma equipa de marketing de forma a divulgar e impulsionar o lançamento deste novo sistema.

A equipa de analista vai ser responsável pela recolha e organização de requerimentos.

A última categoria, “Outros”, vai incluir cidadãos e turistas selecionados para a realização de inquéritos de opinião e também a CMB, de forma a conseguir a aceder a documentos de informação arquivados, para a recolha informação que possa ser transformada em requisitos.

* 1. Plano de Execução

Uma imagem com texto, número, file, Tipo de letra

Descrição gerada automaticamenteO plano de execução do trabalho seguirá um diagrama de GANTT detalhado, abrangendo as fases de definição do sistema, definição de requisitos e modelação concetual. O lançamento do sistema está previsto para o dia 1/1/2024.

Após ter previsto as datas necessárias para a implementação dos passos, equipa sentiu a necessidade de adicionar uma semana para a realização do modelo conceptual.

* 1. Revisão e Aprovação

Após o término dos trabalhos da primeira fase deste projeto, no dia 28/9/2023, foi convocada uma reunião com a empresa Bracara Eventos, sendo o propósito da mesma uma análise da definição do sistema, bem como a fundamentação, os objetivos e a viabilidade o mesmo. Compareceu também nesta reunião o Engenheiro de Bases de Dados, o Arquiteto de Bases de Dados, a equipa de analistas, o Administrador de Bases de Dados, todos estes contratados para a realização deste projeto. Em conjunto todos os pormenores foram alinhavados e validaram tudo o que tinha sido definido anteriormente, assim como também foi aprovado o plano de trabalhos, descrito no diagrama de GANTT produzido. O grupo decidiu, então avançar com o processo de desenvolvimento e implementação do SGBD.

1. Definição de Requisitos
   1. Método de levantamento e de análise de requisitos adotado

Os requisitos foram coletados por meio de um processo abrangente. Esta coleta de requisitos resultou de uma combinação de entrevistas com *stackholders-*chave, ou por outras palavras, pessoas e organizações que podem ser afetadas por este projeto, pesquisas de campo e análises de documentos relacionados com eventos passados e relacionados com o mercado turísticos da cidade, estes últimos facultados pela CMB.

As entrevistas foram realizadas junto de organizadores de eventos locais, autoridades municipais e representantes da indústria do turismo nesta região, para tentar captar e perceber as suas necessidades e expectativas em relação ao SGBD.

As pesquisas de campo serviram para ter uma melhor compreensão das dinâmicas que os eventos têm e das preferências dos cidadãos e dos visitantes. Os documentos serviram para ter um *insight view* adicional.

Com o avanço dos trabalhos em relação ao último ponto, todos os requisitos que foram levantados, foram anotados num documento de recolha (documento de requisitos). Neste documento de requisitos, para cada requisito recolhido, os analistas registaram: a área da aplicação, quem fez o levantamento, quem forneceu o requisito, o texto de requisito, a data e hora do seu levantamento.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Número | **Data e Hora** | **Descrição** | **Área** | **Fonte** | **Analista** |
| 1 | 2/10/2023  10:54 | Todos os eventos que se vão realizar na cidade devem ser registados, com a atribuição de um número único. | Evento | Martim Santos | Diogo Silva |
| 2.3 | 3/11/2023  17:34 | Cada evento vai ter um ID do evento, ou seja, um número único para esse evento, um nome, uma pequena descrição, deverá designar se é pago ou não, se sim deverá incluir um preço, uma data e hora, uma localização onde o evento se irá realizar, deverá listar todas as atividades que possam ocorrer no evento, os artistas que no evento participem, e estará implícito qual foi o custo de realizar o mesmo. | Evento | Martim Santos | Diogo Silva |
| 3.2 | 2/11/2023  11:04 | Um evento poderá ser pago ou não. | Evento | Martim Santos | Pedro Oliveira |
| 4.1 | 3/10/2023  13:58 | A localização de um evento deverá ter a rua onde se realizará o evento, uma descrição do local e o código postal. | Evento | Martim Santos | Pedro Oliveira |
| 5 | 2/10/2023  11:32 | Um evento tem sempre uma atividade pelo menos, senão não é considerado um evento. Pode ter mais do que uma atividade. | Evento | Francisco Ferreira | Diogo Silva |
| 6 | 4/10/2023  15:23 | Uma atividade só pode ser atividade de um evento | Atividade | Francisco Ferreira | Pedro Oliveira |
| 7 | 4/10/2023  15:34 | Cada atividade vai ter um ID da atividade, um nome, uma pequena descrição, um preço, uma data e hora (horário), uma localização, e um artista, poderá ser paga ou não, e tem de estar implícito o custo de realizar uma atividade. | Atividade | Francisco Ferreira | Diogo Silva |
| 8.2 | 15/11/2023  14:26 | É possível ter um registo imediato, anexado aos eventos e às atividades do valor total de ganhos relativos ao evento ou atividade, da quantidade de bilhetes vendidos, bem como a quantidade de bilhetes ainda disponível para o evento ou atividade em questão. | Atividade | Francisco  Ferreira | Diogo Silva |
| 9 | 2/10/2023  11:39 | Cada artista poderá ser individual ou não (ex.: banda, grupo…) | Artista | Ana Rodrigues | Pedro Oliveira |
| 10 | 2/10/2023  11:45 | A localização de uma atividade vai ter a rua onde se realizará a atividade, uma descrição do local e o código postal. | Atividade | Martim Santos | Diogo Silva |
| 11 | 2/10/2023  11:47 | Um artista (vamos considerar um artista como sendo um elemento individual ou um grupo), tem um número único de artista para o identificar, um nome, uma descrição e é possível aceder aos dados do seu agente, e ainda tem o custo associado a contratar esse artista. | Artista | Ana Rodrigues | Diogo Silva |
| 12 | 2/10/2023  11:43 | Um agente tem um ID de agente único, um nome, um email e o seu número de telefone. | Agente | Francisco Ferreira | Pedro Miguel |
| 13 | 2/10/2023  14:15 | É possível registar a afluência de cada evento na cidade quando o evento é pago. | Gestão | Martim Santos | Diogo Silva |
| 14 | 7/10/2023  18:42 | Quando o evento não é pago, vai ser impossível controlar a afluência de pessoas, pelo menos no início deste projeto, uma vez que não temos capacidade de ter os meios necessários para tal. | Gestão | Martim Santos | Diogo Silva |
| 15 | 2/10/2023  14:18 | É possível ver o total de bilhetes vendidos para cada evento pago. | Venda | Martim Santos | Pedro Oliveira |
| 16.1 | 2/10/2023  14:20 | Um evento pode ser grátis, mas ter atividades pagas. | Venda | Martim Santos | Diogo Silva |
| 17 | 2/10/2023  14:21 | Quando se paga o bilhete de um evento, todas as atividades que nele existem estão consideradas pagas. | Venda | Martim Santos | Diogo Silva |
| 18 | 2/10/2023  14:25 | É possível ver o total de bilhetes vendidos para cada atividade paga. | Venda | Ana Rodrigues | Pedro Oliveira |
| 19 | 5/10/2023  11:06 | Vai ser possível pesquisar cada evento por data, se é pago ou não, se for pago pelo preço de bilhete. | Evento | Martim Santos | Diogo Silva |
| 20.3 | 5/10/2023  11:14 | Vai ser possível verificar o total de receitas de cada evento, com o total de bilhetes vendidos. | Venda | Martim Santos | Diogo Silva |
| 21 | 5/10/2023  11:17 | Cada evento pago vai ter um número máximo de bilhetes para serem vendidos, ou seja, a capacidade de cada atividade | Venda | Ana Rodrigues | Pedro Oliveira |
| 22 | 5/10/2023  14:53 | O sistema deverá estar operacional durante 24 horas. | SGBD | Francisco Ferreira | Pedro Oliveira |
| 23.5 | 5/10/2023  14:57 | O sistema poderá ser acedido pelo grupo Bracara Eventos de forma parcial, e de forma total pelo gestor do sistema de bases de dados. O sistema poderá ser acedido pelos utilizadores para apenas verem detalhes dos eventos e das atividades. O grupo apenas tem acesso a operações de consulta nas tabelas da base de dados. | SGBD | Francisco Ferreira | Diogo Silva |
| 24 | 5/10/2023  16:30 | Ao final de cada dia o sistema deverá realizar um relatório com os seguintes dados: o número total de bilhetes vendidos para cada evento, o valor faturado nesse dia com as vendas, o número de bilhetes comprados. | Gestão | Martim Santos | Diogo Silva |
| 25 | 5/10/2023  16:34 | A cada momento é possível ver a listagem de todos eventos na cidade ordenados pela data. | Gestão | Martim Santos | Pedro Oliveira |
| 26 | 5/10/2023  16:40 | A cada momento é possível filtrar quais são os eventos pagos ou não | Gestão | Martim Santos | Pedro Oliveira |
| 27 | 5/10/2023  17:04 | Os bilhetes vendidos deverão ter uma estrutura *default* para todo o tipo de eventos que sejam pagos. | Bilhete | Ana Rodrigues | Pedro Oliveira |
| 28 | 5/10/2023  17:08 | Todos os bilhetes vão ter um registo próprio com a data da venda do bilhete, para melhor controlo sobre a venda dos mesmos. | Bilhete | Ana Rodrigues | Pedro Oliveira |
| 29 | 5/10/2023  17:10 | Os bilhetes têm um número de bilhete, o nome do respetivo evento, o nome da respetiva atividade e um preço. | Bilhete | Ana Rodrigues | Pedro Oliveira |
| 30 | 13/10/2023  10:53 | Cada evento tem sempre um ou mais elemento de staff para ajudar na organização e manutenção do mesmo. | Staff | Ana Rodrigues | Pedro Oliveira |
| 31 | 13/10/2023  10:55 | Cada elemento do staff vai ter um identificador do staff, um nome, a função que cada elemento tem e a lista de números de telefone do elemento. | Staff | Ana Rodrigues | Pedro Oliveira |
| 32 | 13/10/2023  12:55 | Cada elemento do staff pode ajudar em mais do que um evento. | Staff | Martim Santos | Diogo Silva |
| 33 | 13/10/2023  12:57 | Um evento tem sempre pelo menos um elemento do staff. | Staff | Martim Santos | Diogo Silva |
| 34 | 20/11/2023  11:11 | O administrador tem de fazer backups diárias, em mais do que uma localização de modo a não perder dados. | SGBD | Ana Rodrigues | Diogo Silva |
| 35 | 20/11/2023  11:15 | É possível contar quantos eventos se realizaram num ano civil. | SGBD | Ana Rodrigues | Diogo Silva |
| 36 | 20/11/2023  16:11 | É possível obter o número de bilhetes vendidos para um evento e para as atividades em questão. | SGBD | Ana Rodrigues | Diogo Silva |
| 37 | 20/11/2023  18:00 | É possível comprar bilhetes para um evento, não sendo obrigatório, pois o evento pode ser gratuito. | Bilhete | Ana Rodrigues | Pedro Oliveira |
| 38 | 20/11/2023  18:11 | É possível comprar bilhetes para uma atividade, não sendo obrigatório, pois a atividade pode ser gratuita. | Bilhete | Ana Rodrigues | Diogo Silva |
| 39 | 20/11/2023  18:12 | Uma atividade pode ou não ter um artista associado, por exemplo se a atividade for uma prova de comida. Por outro lado, um artista pode participar em 1 ou mais atividades. | Atividade | Ana Rodrigues | Pedro Oliveira |
| 40 | 21/11/2023  19:23 | A cada momento é possível aceder a uma listagem dos eventos que se realizam num mês. | Gestão | Ana Rodrigues | Pedro Oliveira |
| 41 | 21/11/2023  19:32 | A cada momento é possível aceder a uma listagem dos 5 eventos que mais faturaram. | Gestão | Ana Rodrigues | Pedro Oliveira |

* 1. Análise e Organização

Quando os analistas verificaram que a maioria dos processos operacionais em torno da organização de eventos já tinham sido estudados e os seus requisitos levantados, começaram a fazer a análise e a organização dos requisitos obtidos. Nesta segunda fase, os analistas iriam verificar a ocorrência de erros, inconsistências, redundâncias, entre outros problemas que podiam ter aparecido anteriormente. Estes requisitos recolhidos anteriormente foram analisados um a um, e, tendo em conta as vistas de utilização definidas, começaram a organizá-los de acordo com as três vertentes do trabalho de dados do futuro sistema, nomeadamente:

* Descrição: Para acolher os requisitos que referiam a criação de objetos na base de dados (tabelas, atributos, domínios, restrições, etc., …);
* Manipulação: Para incluir tudo aquilo que referisse o povoamento ou exploração de dados, quer fosse através de simples interrogações à base de dados (*queries)*, ou fosse através de procedimentos, funções, utilização na aplicação, etc…
* Controlo: Para saber como é que iriam gerenciar a base de dados e a sua utilização

Estes novos documentos têm uma estrutura semelhante ao documento de requisitos inicial, porem os requisitos têm uma numeração própria e o seu texto já foi revisto e analisado pelos analistas do sistema.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Número | Data e Hora | Descrição | Área | Fonte | Analista |
| RD1 | 02/10/2023  10:54 | Todos os eventos que se vão realizar na cidade devem ser registados, com a atribuição de um número único. | Evento | Martim Santos | Diogo Silva |
| RD2.1 | 03/10/2023  13:56 | Cada evento vai ter um ID do evento, ou seja, um número único para esse evento, um nome, uma pequena descrição, deverá designar se é pago ou não, se sim deverá incluir um preço, uma data e hora, uma localização onde o evento se irá realizar, deverá listar todas as atividades que possam ocorrer no evento, os artistas que no evento participem, e estará implícito qual foi o custo de realizar o mesmo. | Evento | Martim Santos | Diogo Silva |
| RD3 | 02/10/2023  10:59 | A localização de um evento deverá ter a rua onde se realizará o evento, uma descrição do local e o código postal. | Evento | Francisco Ferreira | Diogo Silva |
| RD4 | 02/10/2023  11:04 | Um evento tem sempre uma atividade pelo menos, senão não é considerado um evento. Pode ter mais do que uma atividade. | Evento | Martim Santos | Pedro Oliveira |
| RD5 | 02/10/2023  11:32 | Uma atividade só pode ser atividade de um evento | Atividade | Martim Santos | Pedro Oliveira |
| RD5.1 | 03/10/2023  13:58 | Cada atividade vai ter um ID da atividade, um nome, uma pequena descrição, um preço, uma data e hora (horário), uma localização, e um artista, poderá ser paga ou não, e tem de estar implícito o custo de realizar uma atividade. | Atividade | Martim Santos | Pedro Oliveira |
| RD6 | 02/10/2023  11:32 | É possível ter um registo imediato, anexado aos eventos e às atividades do valor total de ganhos relativos ao evento ou atividade, da quantidade de bilhetes vendidos, bem como a quantidade de bilhetes ainda disponível para o evento ou atividade em questão. | Gestão | Francisco Ferreira | Diogo Silva |
| RD7.2 | 04/10/2023  15:34 | Cada artista poderá ser individual ou não (ex.: banda, grupo…). | Artista | Francisco Ferreira | Diogo Silva |
| RD8 | 02/10/2023  11:39 | A localização de uma atividade vai ter a rua onde se realizará a atividade, uma descrição do local e o código postal. | Artista | Ana Rodrigues | Pedro Oliveira |
| RD9 | 02/10/2023  11:45 | Um artista (vamos considerar um artista como sendo um elemento individual ou um grupo), tem um número único de artista para o identificar, um nome, uma descrição e é possível aceder aos dados do seu agente, e ainda tem o custo associado a contratar esse artista. | Artista | Martim Santos | Diogo Silva |
| RD10 | 02/10/2023  11:47 | Um agente tem um ID de agente único, um nome, um email e o seu número de telefone. | Agente | Ana Rodrigues | Diogo Silva |
| RD11.1 | 02/10/2023  11:43 | Um evento pode ser grátis, mas ter atividades pagas. | Evento | Francisco Ferreira | Pedro Oliveira |
| RD12 | 02/10/2023  14:20 | Quando se paga o bilhete de um evento, todas as atividades que nele existem estão consideradas pagas. | Bilhete | Martim Santos | Diogo Silva |
| RD13 | 02/10/2023  14:21 | Cada evento pago vai ter um número máximo de bilhetes para serem vendidos, ou seja, a capacidade de cada atividade | Venda | Martim Santos | Diogo Silva |
| RD14 | 02/10/2023  14:30 | Os bilhetes vendidos deverão ter uma estrutura *default* para todo o tipo de eventos que sejam pagos. | Venda | Francisco Ferreira | Pedro Oliveira |
| RD15 | 02/10/2023  14:50 | Todos os bilhetes vão ter um registo próprio com a data da venda do bilhete, para melhor controlo sobre a venda dos mesmos. | Bilhete | Martim Santos | Diogo Silva |
| RD16 | 02/10/2023  14:54 | Os bilhetes têm um número de bilhete, o nome do respetivo evento, o nome da respetiva atividade e um preço. | Bilhete | Martim Santos | Diogo Silva |
| RD17 | 02/10/2023  14:55 | Cada evento tem sempre um ou mais elemento de staff para ajudar na organização e manutenção do mesmo. | Staff | Francisco Ferreira | Diogo Silva |
| RD18.1 | 02/10/2023  15:24 | Cada elemento do staff vai ter um identificador do staff, um nome, a função que cada elemento tem e a lista de números de telefone do elemento. | Staff | Francisco Ferreira | Diogo Silva |
| RD19 | 13/10/2023  12:55 | Cada elemento do staff pode ajudar em mais do que um evento. | Staff | Martim Santos | Diogo Silva |
| RD20 | 13/10/2023  12:57 | Um evento tem sempre pelo menos um elemento do staff. | Staff | Martim Santos | Diogo Silva |
| RD21 | 16/10/2023  13:00 | É possível comprar bilhetes para um evento, não sendo obrigatório, pois o evento pode ser gratuito. | Venda | Francisco Ferreira | Pedro Oliveira |
| RD21 | 16/10/2023  13:00 | É possível comprar bilhetes para uma atividade, não sendo obrigatório, pois a atividade pode ser gratuita. | Venda | Francisco Ferreira | Pedro Oliveira |
| RD22 | 17/10/2023  12:56 | Uma atividade pode ou não ter um artista associado, por exemplo se a atividade for uma prova de comida. Por outro lado, um artista pode participar em 1 ou mais atividades. | Atividade | Francisco Ferreira | Pedro Oliveira |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Número | Data e Hora | Descrição | Área | Fonte | Analista |
| RM1 | 02/10/2023  14:15 | É possível registar a afluência de cada evento na cidade quando o evento é pago. | Gestão | Martim Santos | Diogo Silva |
| RM2 | 07/10/2023  18:42 | Quando o evento não é pago, vai ser impossível controlar a afluência de pessoas, pelo menos no início deste projeto, uma vez que não temos capacidade de ter os meios necessários para tal. | Gestão | Martim Santos | Diogo Silva |
| RM3 | 02/10/2023  14:20 | É possível ver o total de bilhetes vendidos para cada evento pago. | Venda | Martim Santos | Pedro Oliveira |
| RM4 | 02/10/2023  14:25 | É possível ver o total de bilhetes vendidos para cada atividade paga. | Venda | Ana Rodrigues | Pedro Oliveira |
| RM5 | 02/10/2023  14:28 | É possível ver o total de bilhetes vendidos para cada atividade paga. | Venda | Francisco Ferreira | Pedro Oliveira |
| RM6 | 03/10/2023  10:53 | Vai ser possível pesquisar cada evento por data, se é pago ou não, se for pago pelo preço de bilhete. | Gestão | Martim Santos | Pedro Oliveira |
| RM7 | 03/10/2023  10:55 | Vai ser possível verificar o total de receitas de cada evento, com o total de bilhetes vendidos. | Gestão | Ana Rodrigues | Pedro Oliveira |
| RM8 | 03/10/2023  10:56 | Ao final de cada dia o sistema deverá realizar um relatório com os seguintes dados: o número total de bilhetes vendidos para cada evento, o valor faturado nesse dia com as vendas, o número de bilhetes comprados. | Gestão | Ana Rodrigues | Pedro Oliveira |
| RM9 | 03/10/2023  10:59 | A cada momento é possível ver a listagem de todos eventos na cidade ordenados pela data. | Gestão | Martim Santos | Pedro Oliveira |
| RM10 | 05/10/2023  11:06 | A cada momento é possível filtrar quais são os eventos pagos ou não | Evento | Martim Santos | Diogo Silva |
| RM11 | 05/10/2023  11:14 | É possível contar quantos eventos se realizaram num ano civil. | Gestão | Martim Santos | Diogo Silva |
| RM12 | 05/10/2023  16:30 | É possível obter o número de bilhetes vendidos para um evento e para as atividades em questão. | Gestão | Martim Santos | Diogo Silva |
| RM13 | 21/11/2023  19:23 | A cada momento é possível aceder a uma listagem dos eventos que se realizam num mês. | Gestão | Ana Rodrigues | Pedro Oliveira |
| RM14 | 21/11/2023  19:32 | A cada momento é possível aceder a uma listagem dos 5 eventos que mais faturaram. | Gestão | Ana Rodrigues | Pedro Oliveira |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Número | Data e Hora | Descrição | Área | Fonte | Analista |
| RC1 | 05/10/2023  14:53 | O sistema deverá estar operacional durante 24 horas. | SGBD | Francisco Ferreira | Pedro Oliveira |
| RC2 | 05/10/2023  14:57 | O sistema poderá ser acedido pelo grupo Bracara Eventos de forma, forma parcial, e de forma total pelo gestor do sistema de bases de dados. O sistema poderá ser acedido pelos utilizadores para apenas verem detalhes dos eventos e das atividades. O grupo apenas tem acesso a operações de consulta nas tabelas da base de dados. | SGBD | Francisco Ferreira | Diogo Silva |
| RC3 | 05/10/2023  16:53 | O administrador tem de fazer backups diárias, em mais do que uma localização de modo a não perder dados. | SGBD | Francisco Ferreira | Diogo Silva |

* 1. Análise e validação geral dos requisitos

Após todos os requisitos terem sido revistos pela equipa de analistas, realizou-se uma reunião com os restantes intervenientes do projeto para fazer a validação dos mesmos.

Quando se discutia sobre os requisitos em questão, especialmente os requisitos de controlo, surgiu uma grande dúvida no ar. Como todos presentes na reunião sabiam e entendem o quão valiosos são os dados de uma pessoa nos dias de hoje, surgiu a necessidade de criar um requisito unicamente utilizado para garantir aos utilizadores que os seus dados estão sobre o controlo dos mesmos, ou seja, os mesmos poderiam eliminá-los e aceder aos mesmos em qualquer momento.

Outro dos requisitos que decidiram que era crucial implementar foi a realização de backups regulares, pois, num sistema desta dimensão, é impensável e quase proibido perder qualquer dado que seja.

Por último, foram discutidos detalhes relativos á implementação de um sistema de subscrição, que permite aos utilizadores terem regalias adicionais.

No final, e após se limarem alguns detalhes e novos requisitos propostos durante a revisão, deu-se a aprovação dos mesmos, e o aval para começar a próxima etapa do projeto.

1. Modelação Concetual
   1. Apresentação da abordagem de modelação realizada

O processo da modelação conceptual é o primeiro passo, ou melhor, o primeiro esquema base de uma base de dados. O modelo conceptual que foi apresentado, é um esquema abstrato e de alto nível, representando as várias entidades envolvidas e descritas na definição dos requisitos, os relacionamentos entre as mesmas e os diversos atributos que caracterizam as entidades e os relacionamentos.

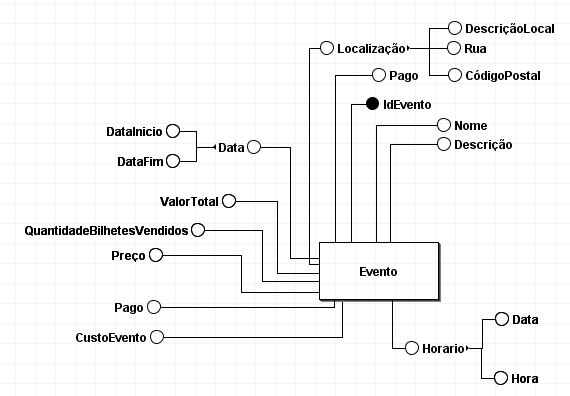
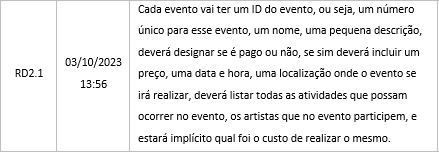
Para este processo decorrer de forma natural e sem qualquer tipo de surpresa pelo caminho, á importante assegurar que todos os requisitos que foram previamente levantados, conseguem definir e atingir todos os objetivos propostos para este projeto. Com base neles identificamos as principais entidades principais da futura base de dados de projeto, bem como os seus relacionamentos e atributos.

A criação do nosso modelo conceptual foi feita com recurso ao software “BrModelo”, utilizando para a implementação das entidades, relacionamentos e atributos, a notação de Peter Chen. O software vai nos permitir criar um Diagrama ER, que é um tipo de fluxograma que ilustra como entidades, por exemplo, pessoas, objetos ou conceitos, que se relacionam entre si dentro de um sistema. Portanto, o primeiro passo para criarmos o nosso diagrama, é identificar as entidades, e após isso determinar como é que as entidades estão relacionadas.

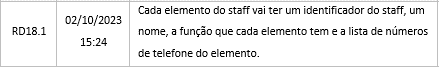
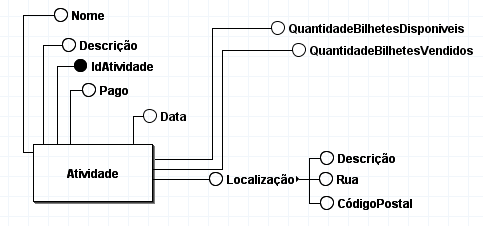
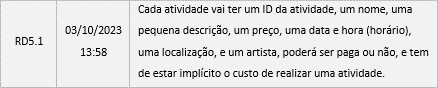
* 1. Identificação e caracterização das entidades

As entidades presentes no nosso modelo conceptual derivaram todas de requisitos de descrição recolhidos anteriormente. Foram identificadas seis entidades, a partir dos mesmos. Para as seguintes entidades, vamos corresponder um requisito que lhe deu origem.

**Entidade Evento:**

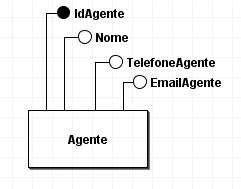
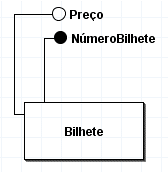


**Entidade Atividade:**



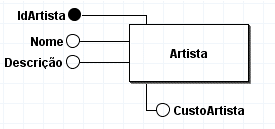
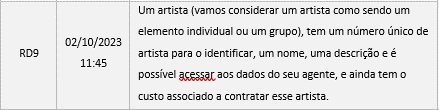
**Entidade Staff:**

**Entidade Bilhete:**



**Entidade Artista:**

**Entidade Agente:**



De seguida, vamos criar uma tabela para abrigar cada uma das entidades e descrever de uma forma sucinta o que cada uma vai representar.

|  |  |  |  |
| --- | --- | --- | --- |
| Designação | Descrição | Sinónimos | Ocorrência |
| Evento | Representa um evento que irá se realizar na cidade, acolhendo o registo de informações importantes para a realização do mesmo, bem como algumas informações financeiras também sobre o mesmo. | Acontecimento | Cada evento vai ter um número próprio, único e sequencial, que é atribuído quando o evento é registado na base de dados. |
| Atividade | Representa uma atividade dentro de um evento, acolhendo o registo de informações importantes para a realização do mesmo. | --- | Cada atividade vai ter um número próprio, único e sequencial, que é atribuído quando a atividade é registada na base de dados. |
| Staff | Informações sobre as pessoas que vão trabalhar num evento. | Voluntário | Cada indivíduo é identificado com um nº único e sequencial, que é obrigatório para se registar como staff. |
| Bilhete | Entidade com informações básicas e essenciais que irão também ser apresentadas nos próprios bilhetes. | Ingresso | Todos os bilhetes vão ter um número de bilhete único. |
| Artista | Representa a informação sobre um artista que vai atuar numa atividade de um evento. Quando utilizamos a palavra artista estamos a considerar que este pode ser individual ou pode ser uma banda. | Banda | Cada artista vai estar identificado na base de dados com um número identificador único. |
| Agente | Representa a informação sobre um agente que trabalha para uma banda. Contém informação básica para definir um agente. | *Manager* | Cada agente vai estar identificado na base de dados com um número identificador único. |

* 1. Identificação e caracterização dos relacionamentos

Temos um total de cinco relacionamentos entre entidades, derivando os mesmo de requisitos anteriormente definidos, que permitem interligar as entidades entre si.

Relacionamento Evento-Staff:

Uma imagem com texto, captura de ecrã, Tipo de letra, file

Descrição gerada automaticamente

A partir destes dois requisitos de descrição podemos claramente definir um relacionamento entre as entidades “Evento” e “Staff”. Os requisitos acusam ainda, que este relacionamento terá cardinalidade 1: N, devido a ser um relacionamento 1 para muitos. Podemos ler este relacionamento da seguinte maneira: Um Evento tem 1 ou mais elementos do *Staff*, e um elemento do staff pode ajudar num ou mais eventos.

Uma imagem com diagrama, file, Retângulo, Esquema

Descrição gerada automaticamente

Relacionamento Evento-Atividade:

Uma imagem com texto, captura de ecrã, Tipo de letra, file

Descrição gerada automaticamente

A partir dos requisitos, podemos definir um relacionamento entre as entidades “Evento” e “Atividade”, com cardinalidade 1 para N. Podemos traduzir o relacionamento da seguinte forma. Um evento tem pelo menos uma atividade podendo ter mais, mas uma atividade só é atividade desse evento.

Uma imagem com diagrama, file, Retângulo, Esquema

Descrição gerada automaticamente

Da mesma forma podemos definir os restantes relacionamentos:

Evento-Bilhete e Atividade-Bilhete

Uma imagem com texto, captura de ecrã, Tipo de letra, branco

Descrição gerada automaticamente

Como existem eventos e atividades gratuitas, a obrigatoriedade de comprar bilhete não existe, então decidimos utilizar a cardinalidade 1: N, que se pode traduzir da seguinte forma para ambos os relacionamentos: Um evento pode ou não vender bilhetes. Cada bilhete é vendido mais do que uma vez para um evento.

Uma imagem com diagrama, file, Esquema, Retângulo

Descrição gerada automaticamente

Atividade-Artista:

Uma imagem com texto, Tipo de letra, captura de ecrã, file

Descrição gerada automaticamente

Este relacionamento vai ter cardinalidade 1: N – 0:1 e pode-se ler da seguinte forma: Uma atividade pode ou não ter um artista, sendo que se tiver é apenas 1. Um artista pode ser artista dessa atividade ou de outras.

Uma imagem com file, diagrama, captura de ecrã, Retângulo

Descrição gerada automaticamente

Artista – Agente

Uma imagem com texto, captura de ecrã, Tipo de letra, file

Descrição gerada automaticamente

Embora não esteja visível à primeira vista, podemos retirar um relacionamento entre “Artista” e “Agente”, a partir do momento que diz que é possível aceder aos dados do agente. A cardinalidade deste relacionamento é 1: N – 1:1 e pode ser traduzida da seguinte forma: Um artista tem um e um só agente. Um agente agência um ou mais artistas.

Uma imagem com diagrama, file, Gráfico, Tipo de letra

Descrição gerada automaticamente

Ficam agora resumidas os relacionamentos em formato tabela.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Entidade | Relacionamento | Cardinalidade | Entidade | Explicação |
| Evento | tem | 1:1 – 1: N | Atividade | Um evento pode ter uma ou várias atividades associadas a ele. Cada atividade pertence a um evento. |
| Evento | tem | 1: N- 1: N | Staff | Um evento tem um ou mais elementos do staff. Um elemento do staff faz parte desse e pode fazer parte de mais eventos. |
| Evento | vende | 1, N:0, N | Bilhete | Um evento pode ou não vender bilhetes. Cada bilhete é vendido mais do que uma vez para um evento. |
| Atividade | vende | 1, N:0, N | Bilhete | Uma atividade pode ou não vender bilhetes. Cada bilhete é vendido mais do que uma vez para uma atividade. |
| Atividade | possui | 1: N – 0:1 | Artista | Uma atividade pode ou não ter um artista. Se tiver um artista, será apenas um. Um artista pode estar associado a várias atividades em diferentes eventos. |
| Artista | tem | 1: N- 1:1 | Agente | Cada artista possui um único agente que pode representar vários artistas. |

* 1. Identificação e caracterização da associação dos atributos com as entidades e relacionamentos

Com base nos requisitos levantados, e para cada entidade e relacionamento definido, foram associados os seguintes atributos:

* Entidades:
  + **Evento** – ID, Pago, Nome, Descrição, Localização (Descrição, Rua, CódigoPostal), Data (Data\_Inicio, Data\_Fim), ValorTotal, QuantidadeBilhetesVendidos, Preço, CustoEvento
  + **Atividade** – Nome, Descrição, IdAtividade, Pago, Data, Localização (Descrição, Rua, CódigoPostal) e Lotação, Preço, CustoAtividade, QuantidadeBilhetesDisponveis, QuantidadeBilhetesVendidos
  + **Staff** – IdStaff, Nome, Função e Telefone
  + **Bilhete** –NúmeroBilhete e Preço
  + **Artista** –IdArtista, Nome, Descrição, CustoArtista
  + **Agente** – IdAgente, Nome, TelefoneAgente e eMailAgente

Em baixo está apresentada uma tabela que tem como intuito descrever de forma muito resumida o papel de cada atributo neste modelo conceptual da base de dados.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Entidade | Atributo | Descrição | Domínio e Tamanho | Nulo | Exemplo |
| Evento | IdEvento  Pago  Nome  Descrição  Localização (  Descrição  Rua  CódigoPostal)  Data  Data\_Inicio    Data\_Fim  ValorTotal  QuantidadeBilhetesVendidos  Preço  CustoEvento | Identificador do Evento  Valor booleano para dizer se o evento é pago ou não  Nome do Evento  Pequena descrição sobre o evento  Localização do Evento  Descrição do local do evento  Nome da Rua  Código Postal  Data que marca o início do evento  Data que marca o fim do evento  Valor Arrecadado  Quantos bilhetes foram vendidos  Preço derivado do bilhete  Quanto custa realizar o evento | INTEGER  BOOLEAN ou TINYINT(1)  VARCHAR(50)  TEXT  VARCHAR(100)  VARCHAR(75)  VARCHAR(20)  DATETIME  DATETIME  INTEGER  INTEGER  DECIMAL(8,2)  DECIMAL(8,2) | N  N  N  S  N  N  N  N  N  S  S  S  S | 1  True  Braga Authentica  “Festival de música onde irão estar presentes os maiores artistas do mundo, …”  Altice Fórum  R. Monsenhor Airosa  4705-002  17/12/2023  21/12/2023  15000  500  50.00  50000.00 |
| Atividade | IdAtividade  Nome  Descrição  Pago  Data  Localização    Descrição    Rua  CódigoPostal  Lotação  Preço  CustoAtividade  QuantidadeBilhetesDisponiveis  QuantidadeBilhetesVendidos | Identificador da atividade  Nome da Atividade  Pequena descrição sobre a atividade  Valor booleano para dizer se a atividade é paga ou não  Data que se irá realizar uma atividade  Localização onde decorrerá uma atividade  Descrição do local da atividade  Nome da Rua  Código Postal  Lotação máxima de espectadores numa atividade  Custo da atividade  Quanto custa realizar a atividade  Nrº Bilhetes Disponiveis  Nrº Bilhetes Vendidos | INTEGER  VARCHAR(50)  TEXT  BOOLEAN ou TINYINT(1)  DATETIME  VARCHAR(100)  VARCHAR(75)  VARCHAR(20)  INTEGER  DECIMAL(8,2)  DECIMAL(8,2)  INTEGER  INTEGER | N  N  S  N  N  N  N  N  N  S  S  S  S | 1  Concerto Placebo  “Placebo voltam aos palcos no festival “Braga Authentica…”  True  20/12/2023  Altice Fórum  R.Monsenhor Airosa  4705-002  50000  50.00  50000.00  1000  500 |
| Staff | IdStaff  Nome  Funçao  Telefone | Identificador de um elemento do Staff  Nome do elemento do Staff  Função que vai exercer no evento  Número Telefone | INTEGER  VARCHAR(75)  VARCHAR(50)  VARCHAR(15) | N  N  N  S | 1  Manuel Afonsino  Auxiliar de limpeza  +351 992 488 223 |
| Bilhete | NúmeroBilhete  Preço | Identificador de um bilhete  Preço do bilhete | INTEGER  INTEGER | N  N | 1  100 |
| Artista | IdArtista  Nome  Descrição  CustoArtista | Identificador do artista  Nome ou designação do artista (consideramos que um artista pode ser individual, ou então um grupo)  Pequena descrição adicional sobre o artista  Quanto custa o artista | INTEGER  VARCHAR (75)  TEXT  DECIMAL(8,2) | N  N  S  S | 1  Kurt Cobain, The Script  “Esta banda foi inaugurada no dia 24 de setembro de 1998…”  1000.00 |
| Agente | IdAgente  Nome  TelefoneAgente  eMailAgente | Identificador de um agente  Nome do agente  Telefone Agente  Email do agente | INTEGER  VARCHAR(75)  VARCHAR(15)  VARCHAR(50) | N  N  N  N | 1  Nicolino Andrade Vieira  +351 912 241 224  nicolino\_vieira23@outlook.pt |

Quanto a alguns tipos de atributos presentes nas tabelas, foram discutidas diferentes maneiras de representar o valor pretendido.

Por exemplo, nas entidades “Evento” e “Atividade”, e no respetivo atributo “Pago”, comum às duas entidades, foi escolhido o tipo *BOOLEAN*. Esta escolha prende-se no facto de querermos que este atributo apenas represente os valores *True* e *False*, de modo a exprimir se um evento é pago ou não, e se uma atividade é paga ou não. Na linguagem *SQL*, não existe um tipo especifico *BOOLEAN*, mas existe este tipo *BOOLEAN* como um sinónimo associado ao tipo *TINYINT*(1), que apenas apresenta dois valores: 1 para *True* e 0 para *False*. Para facilitar a leitura dos tipos dos atributos decidimos utilizar o tipo *BOOLEAN*, sabendo no entanto que o tipo certo para se utilizar nestas situações é o tipo *TINYINT*(1).

Decidimos também utilizar um atributo composto “Data” na entidade “Evento”, com os atributos “Data\_Inicio” e “Data\_Fim”, para marcar o dia em que começa um evento e o dia em que termina o mesmo evento.

Usamos nos atributos que guardam os números de telefone, um atributo multivalorado, de forma a permitir armazenar mais do que um número de telefone.

* 1. Apresentação e explicação do diagrama ER produzido

O nosso modelo conceptual finalizado, contém 6 entidades e 5 relacionamentos.

Das 6 entidades, a que tem maior grau (número de atributos que a constituem) ,16, é a entidade “Evento”, sendo a de menor grau, a entidade “Bilhete” com grau 2.

Dos 5 relacionamentos, todos são relacionamentos binários, ou sejam ligam apenas 2 entidades, à exceção do relacionamento “vende”, que liga 3 entidades, e assim é dito ternário.

Este modelo conceptual foi realizado com o objetivo de otimizar a simplificação na criação de tabelas e cumprir o maior número de requisitos possível.

Em baixo podemos observar como ficou definido o modelo conceptual final, depois de analisados todos os pormenores, todos as entidades, relacionamentos e atributos. O mesmo está disponível nos anexos, junto com o relatório.

Uma imagem com diagrama, Desenho técnico, Esquema, esquemático

Descrição gerada automaticamente

Figura 1 - Representação final do modelo conceptual do SGBD

Desta forma, e após ter uma melhor noção do modelo que queremos implementar na nossa base de dados de gestão de um calendário de eventos numa cidade turística, mais especificamente Braga, é possível avançar para a conceção de uma nova modelação, lógica, em que o objetivo final é chegar a uma implementação física.

1. Modelação Lógica
   1. Construção e validação do modelo de dados lógico

Nesta etapa do trabalho, são realizadas diversas operações para além da construção do modelo lógico. Nomeadamente estamos a falar de operações como a validação do nosso esquema através da normalização de dados, verificação das restrições de integridade e a revisão do esquema lógico final produzido com os seus utilizadores. Nesta última operação, vamos confirmar se a nossa base de dados vai ser capaz de responder às necessidades dos mesmos.

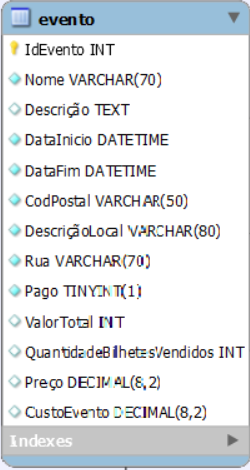
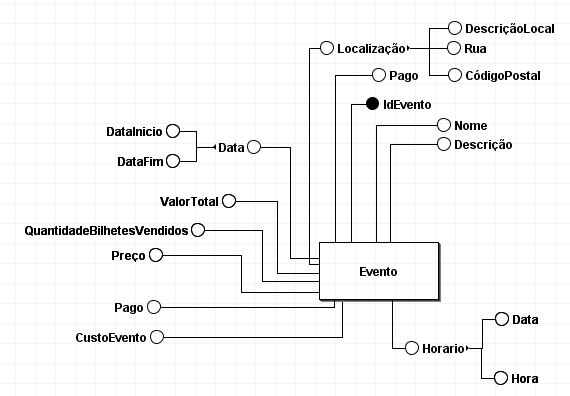
O modelo lógico infra apresentado resulta da conversão das entidades e relacionamentos do nosso modelo conceptual. Esta conversão é realizada a partir de uma linha definida de pontos que passamos a apresentar:

* Uma entidade no esquema conceptual corresponde a uma tabela no modelo lógico.
* Um atributo composto é representado na sua tabela base, apenas pelos seus atributos que o compõem.
* Um atributo multi-valor dá origem a uma nova tabela, com um relacionamento 1: N com a sua tabela de referência.
* Um relacionamento binário 1: N (um para muitos) é implementando com uma chave estrangeira do lado “N” com referência à tabela do lado do “1”.
* Um relacionamento binário N: M (muitos para muitos) dá origem a uma nova tabela, contendo na sua chave primária duas chaves estrangeiras, cada uma delas com referência a uma das tabelas envolvidas no relacionamento.
* Os identificadores de uma entidade passam a designar-se *Primary Key* ou *Foreign Key,* caso em que o identificador de uma entidade esteja presente noutra tabela.

Podemos passar então à realização da conversão. As tabelas que foram criadas foram as seguintes:

1. **Evento**
   1. ***Primary Key:*** IdEvento INT
   2. **Atributos:** IdEvento INT, Nome VARCHAR(70), Descrição TEXT, DataInicio DATETIME, DataFim DATETIME, CodPostal VARCHAR(50), DescriçãoLocal VARCHAR(50), Rua VARCHAR(70), Pago TINYINT(1), ValorTotal INT, QuantidadeBilhetesVendidos INT, Preço DECIMAL(8,2), CustoEvento DECIMAL(8,2)
   3. ***Foreing Key:*** Não tem

A tabela “Evento” foi uma derivação da entidade “Evento” no nosso modelo conceptual. Como o relacionamento é 1: N entre a entidade “Evento” e “Atividade”, não adicionamos nenhuma chave estrangeira do lado da tabela “Evento”. A chave primária foi obtida ao converter o identificador da entidade.

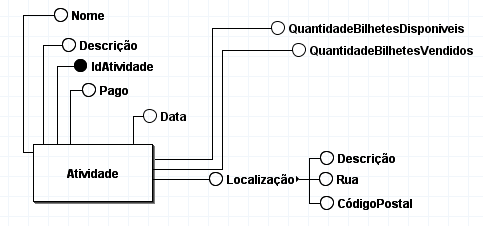
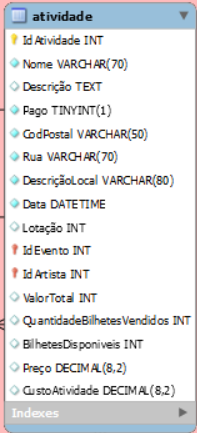


1. **Atividade**
   1. ***Primary Key:*** IdAtividade INT
   2. **Atributos:** IdAtividade INT, Nome VARCHAR (70), Descrição TEXT, Pago TINYINT(1), CodPostal VARCHAR(50), DescriçãoLocal VARCHAR(50), Rua VARCHAR(70),Data DATETIME, Lotação INT, IdEvento INT, IdArtista INT, ValorTotal INT, QuantidadeBilhetesVendidos INT, BilhetesDisponiveis INT, Preço DECIMAL (8,2), CustoAtividade DECIMAL(8,2)
   3. ***Foreing Key:*** IdEvento INT e IdArtista INT

A tabela “Atividade” foi uma derivação da entidade “Atividade” no nosso modelo conceptual. Como o relacionamento “tem” entre a entidade “Evento” e “Atividade” tem cardinalidade 1: N, então do lado da atividade incluímos a chave estrangeira “IdEvento”. Como o relacionamento “possui” entre “Atividade” e “Artista” tem cardinalidade (1,N)-(0,1), adicionamos uma chave estrangeira “IdArtista” na tabela “Atividade”. A chave primária foi obtida ao converter o identificador da entidade.

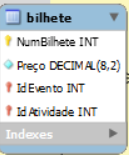
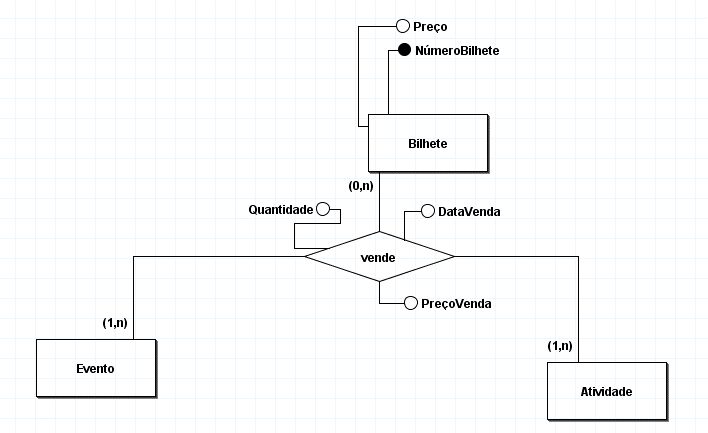
Chave estrangeira proveniente do relacionamento “Atividade” – “Artista”

Chave estrangeira proveniente do relacionamento “Evento” – “Atividade”

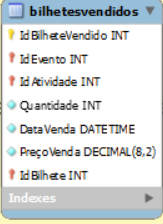
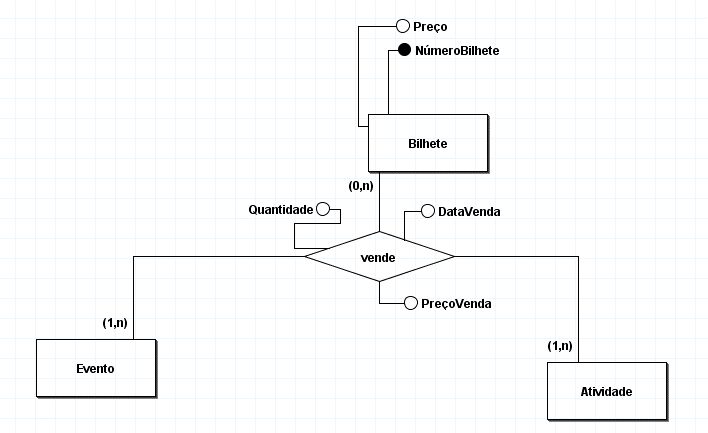


1. **Bilhete**
   1. ***Primary Key:*** NúmeroBilhete INT
   2. **Atributos:** NúmeroBilhete INT, Preço DECIMAL(8,2), IdEvento INT, IdArtista INT
   3. ***Foreing Key:*** IdEvento INT e IdArtista INT

A tabela “Bilhete” foi uma derivação da entidade “Bilhete” do nosso modelo conceptual. Como existe um relacionamento entre “Atividade” – “Bilhete” e “Evento” – “Bilhete”, ambos com cardinalidade 1: N, foram adicionadas duas chaves estrangeiras “IdEvento” e “IdAtividade”, ambas provenientes, respetivamente, das entidades “Evento” e “Atividade”.



1. **BilheteVendidos**
   1. ***Primary Key:*** IdBilheteVendidos INT
   2. **Atributos:** IdBilheteVendido INT, IdEvento INT, IdAtividade INT, Quantidade INT, DataVenda DATETIME, PreçoVenda DECIMAL(8,2), IdBilhete INT
   3. ***Foreing Key:*** IdEvento INT, IdArtista INT, IdBilhete INT



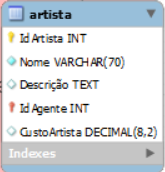
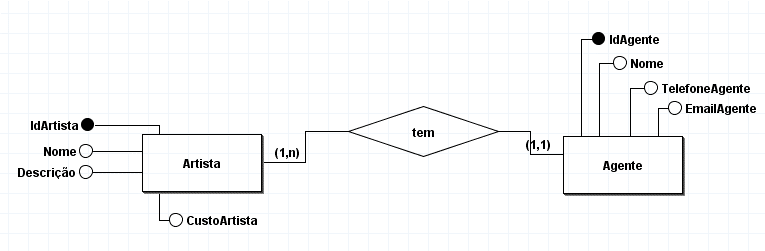
Esta tabela foi criada, posteriormente, com o objetivo de manter registo de todos os bilhetes vendidos, um por um, associados a um evento, atividade e a um tipo de bilhete criado para este evento ou atividade. Esta tabela resultou do relacionamento “vende”, entre “Evento”, “Atividade” e “Bilhete”. Sendo que as chaves estrangeiras presentes nesta tabela, relacionam-se respetiva com as chaves primárias das tabelas “Evento”, “Atividade” e “Bilhete”. Além das chaves estrangeiras que foram adicionadas, foram também adicionados à tabela os campos DataVenda, Quantidade e PreçoVenda, provenientes dos atributos presentes no relacionamento “Vende”.

1. **Artista**

***5. 1 Primary Key:*** IdArtista INT

* 1. **Atributos:** IdArtista INT, Nome VARCHAR (70), Descrição TEXT, IdAgente INT, CustoArtista DECIMAL(8,2)
  2. ***Foreing Key:*** IdAgente INT

Esta tabela foi uma derivação direta da entidade “Artista” e dos seus atributos. Como existe um relacionamento 1: N entre “Artista” e “Agente”, foi adicionada uma chave estrangeira IdAgente à tabela “Artista”.

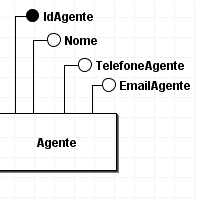
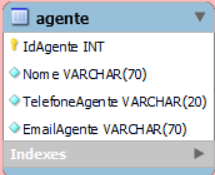


1. **Agente**

***6.1 Primary Key:*** IdAgente INT

* 1. **Atributos:** IdAgente INT, Nome VARCHAR (70), TelefoneAgente VARCHAR (20) , EmailAgente VARCHAR(70)
  2. ***Foreing Key:*** Não tem

Esta tabela foi uma derivação direta da entidade “Agente” e dos seus atributos.

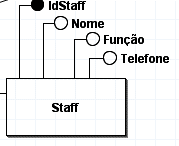
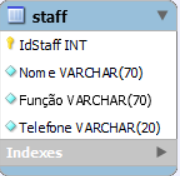


1. **Staff**

***7.1 Primary Key:*** IdStaff INT

* 1. **Atributos:** IdStaff INT, Nome VARCHAR (70), Telefone VARCHAR (20) , Função VARCHAR(70)
  2. ***Foreing Key:*** Não tem

Esta tabela é derivada diretamente a partir da entidade “Staff”

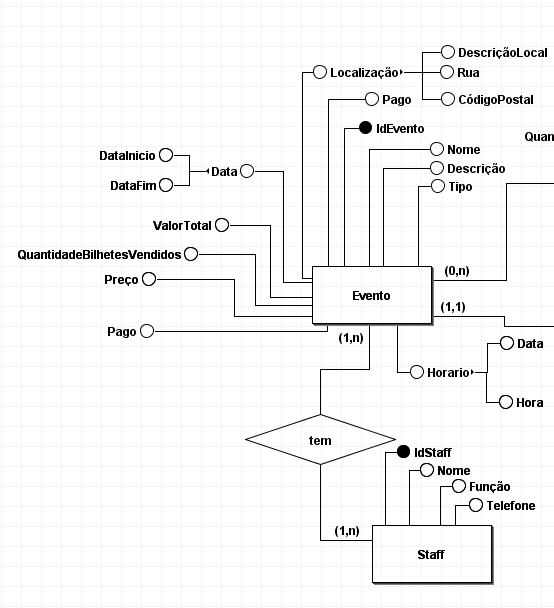
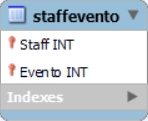


1. **StaffEvento**

***8.1 Primary Key:*** Staff INT , Evento INT

* 1. **Atributos:** Staff INT, Evento INT
  2. ***Foreing Key:*** *Staff INT, Evento INT*

Neste caso, e como existe um relacionamento N: N entre as entidades “Evento” e “Staff”, é criada esta tabela “StaffEvento” , que aporta ambas as chaves primárias da tabela “Evento” e “Staff”.



Assim, e após o término da nossa conversão podemos observar que obtivemos 8 tabelas, das quais 2 foram originadas a partir de relacionamentos.

* 1. Normalização de Dados

Nesta etapa do trabalho, e já com o modelo lógico praticamente finalizado, é altura de o validar através da normalização dos seus dados. A normalização de dados tem como objetivo evitar a redundância dos dados, permitindo assim otimizar da melhor forma a nossa base de dados e permite mais tarde, evitar erros que possam existir na inserção, remoção e alteração dos registos na base de dados.

Para efetuar uma validação preliminar através da normalização dos dados, é necessário verificar se o nosso modelo cumpre todas as formas normais até à terceira.

A Primeira Forma Normal (1ªFN) estipula que os atributos de uma tabela devem ser atómicos, isto é, as tabelas não podem conter registos duplicados. Ao analisar as tabelas geradas na conversão para o modelo lógico, verificamos que este critério é cumprido, confirmando que o nosso modelo satisfaz a 1ªFN.

A Segunda Forma Normal (2ªFN) requer que a 1ªFN seja satisfeita e que todos os atributos não-chave das tabelas do nosso modelo lógico dependam unicamente da chave primária dessa tabela. Ao reexaminar o modelo, constatamos que todos os atributos das tabelas dependem da chave primária das mesmas.

A Terceira Forma Normal (3ªFN) implica que, além da satisfação da 1ªFN e da 2ªFN, todos os atributos não-chave não dependem de outros atributos não-chave. Mais uma vez, após revisão das tabelas, observamos que este requisito é cumprido.

Portanto, tendo as três formas normais satisfeitas, podemos validar o nosso modelo lógico através da normalização dos dados.

* 1. Apresentação e explicação do modelo lógico produzido

Uma imagem com texto, captura de ecrã, diagrama, Paralelo

Descrição gerada automaticamenteO modelo lógico produzido, a partir da conversão do modelo conceptual foi o seguinte.

À medida o modelo lógico foi construído, foram feitas algumas alterações que não estavam previstas no modelo conceptual, como por exemplo a inserção dos atributos “Quantidade”, “DataVenda” e “PreçoVenda”, na tabela “bilhetesVendidos”, e posteriormente também foi revisto o modelo conceptual, para aportar estas alterações. Estas melhorias foram sempre feitas, respeitando os requisitos base do projeto, e visando melhorar a performance e otimização da base de dados, sem nunca comprometer a normalização dos mesmos. Toda esta conversão do modelo conceptual para o lógico foi explicada ao detalhe no ponto 4.1 deste relatório, e revista mais do que uma vez para respeitar todas as regras pré-definidas, e disponíveis também nesse mesmo ponto. Este modelo lógico permite uma gestão eficaz dos eventos, e também a apresentação de detalhes sobre os mesmos, assim como fácil gestão da venda dos bilhetes e organização do staff e dos artistas.

* 1. Validação do modelo com interrogações do utilizador

De modo a validar o nosso modelo lógico e entender se o mesmo consegue responder às necessidades dos utilizadores, e dos pré-requisitos base que foram escolhidos, selecionamentos algumas interrogações dos requisitos de exploração e, recorrendo à álgebra relacional, veremos se as mesmas podem ser respondidas de forma satisfatória.

1. **Contabilizar o número de eventos que foram realizados num ano:**

Uma imagem com texto, captura de ecrã, Tipo de letra, número

Descrição gerada automaticamente

A primeira operação a realizar é uma seleção (σ), que é usada para escolher todas as linhas da tabela “Evento”, que satisfazem a condição que diz que o ano é 2023.

Uma imagem com texto, captura de ecrã, Tipo de letra, cartão de visita

Descrição gerada automaticamente

De seguida foi utilizada uma operação de agregação de dados (γ), para ser possível realizar a contagem de todos os Eventos, cujo ano é de 2023, ou seja, dos valores que foram filtrados anteriormente. Por fim foi realizada uma operação de renomeação (→).

1. **Obter o número de bilhetes vendidos por evento, ordenados por quantidade de bilhetes vendidos**

Uma imagem com texto, captura de ecrã, Tipo de letra, file

Descrição gerada automaticamente

Não há muito que saber acerca do processo de formação desta fórmula. Foi agrupada a soma de bilhetes comprados para cada registo “IdEvento” presente na tabela “BilhetesVendidos”, e após isso foi renomeada a fórmula para “TotalBilhetesVendidos”.

1. **Calcular o total obtido por evento, ordenado pelo valor total em ordem crescente**

Uma imagem com texto, captura de ecrã, Tipo de letra, número

Descrição gerada automaticamente



Para responder a esta interrogação, basta projetar as colunas “IdEvento” e “valortotal” da tabela “Evento” e por fim ordenar por ordem decrescente a coluna “valortotal”.

1. **Listar todos os eventos grátis**

****

Uma imagem com texto, captura de ecrã, Tipo de letra, número

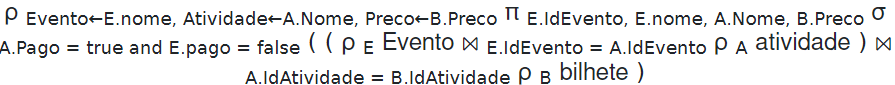
Descrição gerada automaticamente

São projetadas as colunas “IdEvento” e “nome”, da tabela “Evento”, onde o atributo “pago” é falso, ou seja, o evento é gratuito.

1. **Selecionar os eventos grátis que tenham atividades pagas**

Uma imagem com texto, captura de ecrã, Tipo de letra, design

Descrição gerada automaticamente



Esta expressão começa por realizar operações de mudança de nome, e após isso, une as tabelas “Evento”, designada por E na expressão, “Atividade” e “Bilhete”, a primeira junção com base nos atributos “IdEvento” de ambas as tabelas “Evento” e “Atividade”, a junção desta com base nos atributos “IdAtividade” das tabelas “Atividade” e “Bilhete”. Após a junção, a condição “σ A.Pago = *true* and E.Pago = *false*” filtra as linhas onde o evento é gratuito, mas tem atividades pagas. Em último lugar, são projetas as colunas desejadas.

1. Implementação Física

Esta parte do relatório é referente à implementação física. O sistema de gestão de base de dados que foi utilizado é o *MySQL*. Para desenvolver a base de dados, foi utilizado o *software* *MySQL* *Workbench*, e respetiva linguagem padrão, o *SQL* (*Structured Query Language*).

* 1. Tradução do esquema lógico para o sistema de gestão de bases de dados escolhido

O esquema da base de dados da mercearia foi implementado em MySQL utilizando o *script* SQL “202324-UM-LCC-BD-G04-CriaçãoBD”, que foi desenvolvido para este efeito.

Tudo começa com uma simples instrução para criar a nossa base de dados.

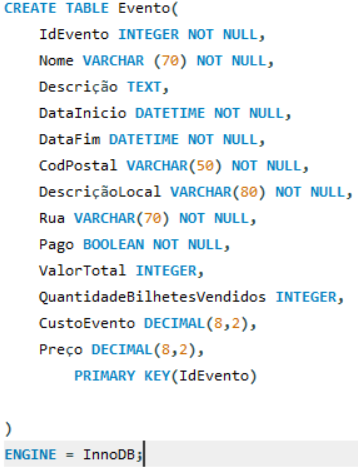
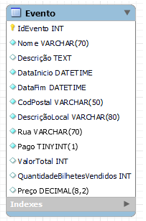
De seguida, foi utilizado o seguinte comando para explicitar ao *MySQL* *Server* qual é a base de dados em que queremos trabalhar.



Após este comando, podemos começar a implementar as nossas tabelas na base de dados “Eventos”.

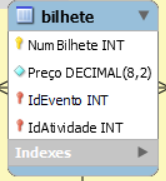
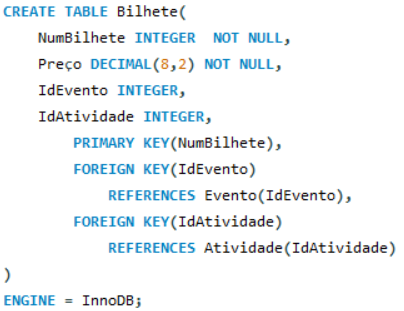
O comando para criar uma tabela na linguagem *SQL* é o seguinte:

Para o caso da tabela “Evento”, foi explicitado ao comando em cima referido, qual é o nome da tabela que pretendemos criar, assim como os nomes e tipos das suas colunas. Por fim, podemos ou não dizer qual é o tipo de motor de criação das tabelas que pretendemos utilizar. No nosso caso queremos utilizar o motor *InnoDB*. Este motor é o motor padrão do *MySQL*, e é frequentemente utilizado devido ao seu alto desempenho e também às suas funcionalidades, como por exemplo o suporte ao padrão de transações *ACID* (Atomicidade, Consistência, Isolamento e Durabilidade). Usamos também em certos atributos, a sintaxe *NOT NULL*, que não permite inserir dados na tabela, sem preencher estes atributos.



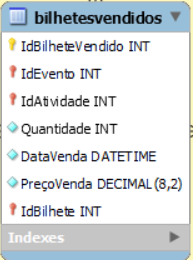
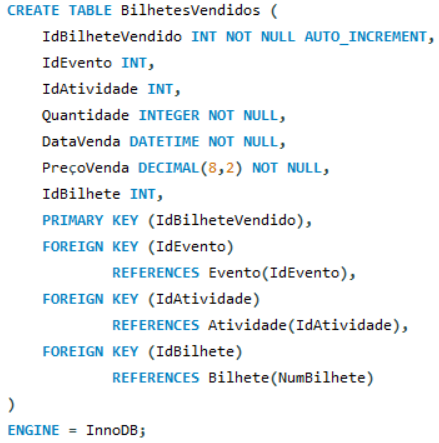
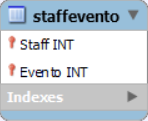
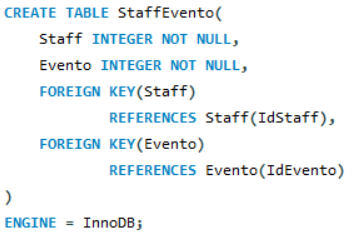
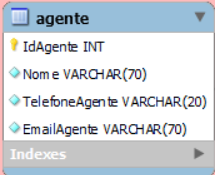
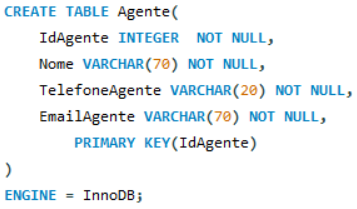
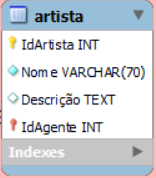
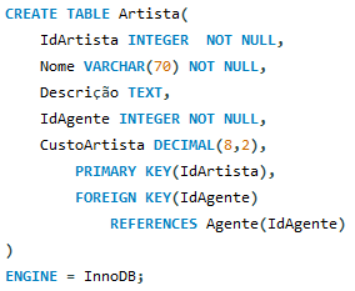
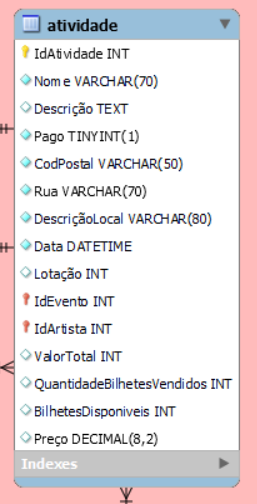
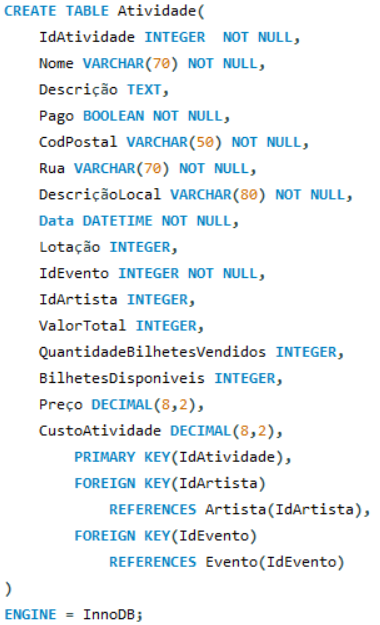
No *script* *SQL* para a criação da BD, vamos ter sintaxes específicas para identificar as chaves primárias e as chaves estrangeiras.

Vejamos a seguinte tabela para exemplificar este ponto:



Neste exemplo, a chave primária foi identificada com a sintaxe *PRIMARY KEY* e as chaves estrangeiras com a sintaxe *FOREIGN KEY REFERENCES*, esta última identificando a chave estrangeira, e respetivamente a tabela onde a chave primária está inserida.

Prosseguimos desta forma para todas as restantes tabelas.



* 1. Tradução das interrogações do utilizador para SQL

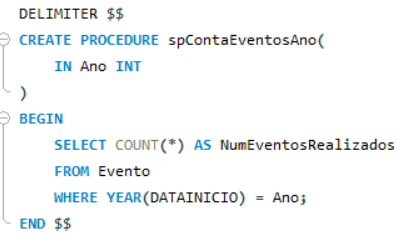
Anteriormente desenvolvemos algumas interrogações, recolhidas com base nos requisitos de manipulação, de modo a validar o nosso modelo lógico e se o mesmo seria capaz de responder a interrogações por parte dos utilizadores. Utilizaremos as mesmas interrogações, agora para comprovar a capacidade da base de dados em as satisfazer.

***Queries* de consulta:**

1. **Contabilizar o número de eventos que foram realizados num ano:**

Para que possamos realizar esta *querie*, de forma satisfatória e para todos os anos que forem pedidos, vamos recorrer a um procedimento (*stored procedure)*, de modo que, dado um determinado ano, ele possa fazer a *querie* pretendida para esse mesmo ano.

Para criar um procedimento, temos de utilizar uma sintaxe específica. Primeira é delimitado o procedimento com dois símbolos iguais à escolha. Por conveniência e para evitar possíveis conflitos, vamos convencionar que para este trabalho, os símbolos vão ser “$$”. Começamos por colocar o comando *“DELIMITER $$”* Após este comando, é assinalado que vai ser criado um procedimento, e todo o código inerente à *querie*, vai ficar contido dentro do mesmo. Como o procedimento recebe um ano como argumento de entrada, o mesmo é identificado com a sintaxe correta. O código completo do procedimento é o seguinte.

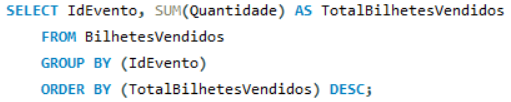


Resultado

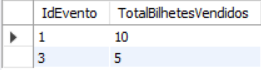


1. **Obter o número de bilhetes vendidos por evento, ordenados por quantidade de bilhetes vendidos**

Para responder a esta interrogação na linguagem *SQL*, foram seguidos os seguintes passos. Primeiramente foi selecionada a coluna “IdEvento” e a soma de todas as quantidades de cada evento, a partir da tabela “BilhetesVendidos”. De seguida, foram agrupadas todos as somas por evento, com recurso ao comando “*GROUP BY”*. Este comando permite que não haja linhas repetidas com informações redundantes do mesmo evento. Por último foi utilizado o comando “*ORDER BY”* para ordenar todos os registos do total de bilhetes vendidos por ordem decrescente.

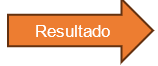


Resultado



1. **Calcular o total obtido por evento, ordenado pelo valor total em ordem crescente**

Da forma como está desenhada a nossa base de dados, esta interrogação torna-se bastante fácil de se realizar. Como existe um campo “ValorTotal” na nossa tabela “Eventos” que é atualizado, a partir de uma *stored procedure*, que é responsável por adicionar bilhetes comprados à nossa base de dados. Muito resumidamente, o procedimento incrementa a este campo o valor dos bilhetes multiplicado pela quantidade que foi comprada dos mesmos. Então, para realizar a *querie*, basta selecionar as colunas “IdEvento” para identificar qual é o evento, e a coluna “ValorTotal” para exibir o total obtido pelo respetivo evento. Por fim é utilizado um comando de ordenação, para ordenar os resultados por ordem decrescente.



1. **Selecionar os agentes que agenciam mais do que 2 artistas e quem agenciam**

Para realizar esta interrogação vamos recorrer ao uso de *sub-queries*, que muito resumidamente consiste em executar uma *querie*, e após a execução da mesma, utilizar o resultado para realizar a segunda *querie*. Primeiramente é executada a *querie* interior, que filtra dos agentes, aqueles que agenciam mais do que um artista. É realizada uma operação de junção de tabelas, com base nos atributos “IdAgente” da tabela “Agente” e “IdAgente” da tabela “Artista”. Feito isto temos acesso aos artistas que são agenciados pelo um agente específico. Após isso, é agrupado o resultado por agente, e realizada uma operação de filtragem, para selecionar apenas os agentes que têm 2 ou mais artistas agenciados. Na *querie* exterior, são projetados os nomes do agente e do artista respetivamente, e é usada novamente uma operação de junção para selecionar os artistas que são agenciados por um determinado agente.

Uma imagem com texto, captura de ecrã, Tipo de letra, número

Descrição gerada automaticamenteUma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente



1. **Fazer um relatório diário da receita gerada por cada evento. (por ano)**

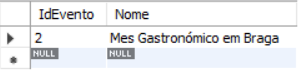
Para realizar esta interrogação vamos recorrer a um recurso do *MySQL* chamado *Event* *Scheduler*, que permite agendar comandos *SQL*, ou uma *querie* neste caso, para um horário em específico e em repetição. Como pretendemos obter um relatório diário da receita gerada, foi criado um *scheduler*, que todos os dias à meia-noite, adiciona numa tabela “RelatorioDiário”, que vai ser criada, o relatório diário pretendido com informações da data, a quantidade de bilhetes vendidos para cada evento e o valor faturado.

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

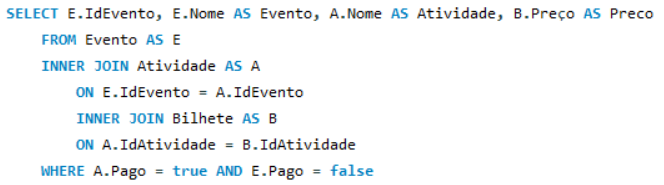
1. **Selecionar todos os eventos gratuitos**

Esta *querie* é muito simples, mas cumpre um dos requisitos de manipulação, que diz que é possível filtrar por eventos pagos e gratuitos. De modo análogo podemos realizar esta interrogação para o caso de pretendermos selecionar todos os eventos pagos.



1. **Selecionar os eventos grátis que tenham atividades pagas**

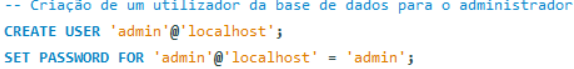
De certo modo, esta interrogação é semelhante à anterior, mas mais uma vez cumpre um dos requisitos de descrição e manipulação, que dizem respetivamente, que existem eventos gratuitos com atividades pagas, e que é possível visualizar quais são, e quais são as respetivas atividades. A *querie* começa com a seleção das colunas “IdEvento” e “Nome” da tabela “Evento”, da coluna “nome” da tabela “Atividade” e da coluna” Preço” da tabela “Bilhete”. De seguida, e de modo a conseguir recolher estes dados realiza duas operações de junção interligadas para aceder à tabela “Atividade” e de seguida à tabela “Bilhete”. Por fim escolhe os resultados onde para onde a coluna “Pago” nas tabelas “Evento” e “Atividade”, tem valores “*false*” e “*true*” respetivamente, ou seja, os valores onde os eventos são gratuitos e as atividades pagas.



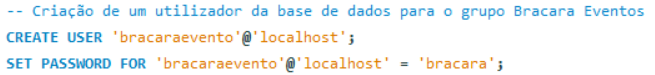
* 1. Definição e caracterização das vistas de utilização em SQL

Durante a definição de requisitos, nomeadamente na parte dos requisitos de controlo, foram definidas uma série de “regras” para a parte de controlo da nossa base de dados. Por outras palavras, foram definidas horas de operação da nossa base de dados, quem podia trabalhar na mesma, assim como as permissões que os utilizadores teriam com a mesma. Desta forma, e a partir dos seguintes requisitos, desenvolvemos uma série de instruções *SQL*, para implementar os mesmos requisitos.

Antes de desenvolver o script de permissões, foram criados os utilizadores para a base de dados.

Primeiro foi criado um utilizador *admin*, que é o responsável por administrar e fazer a manutenção da base de dados.

De seguida, foi criado um utilizador comum para o grupo Bracara Eventos.

Por último, foi criado um utilizador para representar um utilizador diário da nossa base de dados.



Com os utilizadores criados, prosseguiu-se com a definição dos utilizadores que cada utilizador tem na base de dados.

**Requisitos de controlo:**

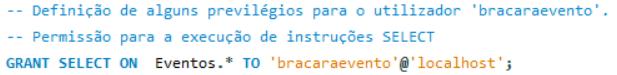
**RC2-** “O sistema poderá ser acedido pelo grupo Bracara Eventos de forma, forma parcial, e de forma total pelo gestor do sistema de bases de dados. O sistema poderá ser acedido pelos utilizadores para apenas verem detalhes dos eventos e das atividades. O grupo apenas tem acesso a operações de consulta nas tabelas da base de dados.”

Para este efeito, foram dadas permissões de consulta e visualização de dados ao utilizador “bracaraevento”, para todas as tabelas da nossa base de dados.

Uma imagem com texto, Tipo de letra, captura de ecrã

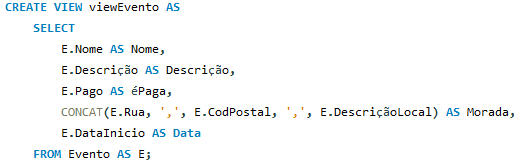
Descrição gerada automaticamente

**RC3- “**O administrador da base de dados tem permissão total para toda a base de dados.”

Para cumprir este requisito, foram concedidads todas as permissões possíveis para o administrador na base de dados “Eventos”.

**RC4- “**Os utilizadores da base de dados apenas podem visualizar dados essenciais dos eventos e das atividades, como por exemplo detalhes como as datas, nome, artistas que atuam, descrição, etc… Não têm acesso a informações como o valor total arrecadado pelo evento, nem informações como custo para se realizar o evento.”

Para cumprir este requisito foram criadas duas vistas de utilização de dados, para exibir detalhes dos eventos e das atividades. A vista “viewEvento” expõe apenas os detalhes necessários dos eventos, e a vista “viewAtividade” mostra apenas os detalhes necessários das atividades.



Com as duas vistas implementadas, foram dadas as permissões necessárias para os utilizadores da base de dados conseguirem visualizar estes dados.



* 1. Cálculo do espaço da base de dados

Tendo em conta os nossos conhecimentos prévios e pesquisas relativamente ao espaço que cada tipo de dado ocupa, foi calculado o espaço ocupado por cada atributo pertencente a cada tabela, de forma a estimar qual será a ocupação em espaço da nossa base de dados. Para ajudar no cálculo, foi assumido o tipo *TEXT* como sendo um tipo *VARCHAR (300).*

**Evento**

|  |  |  |
| --- | --- | --- |
| Atributos | Tipo | Tamanho (Bytes) |
| IdEvento | INTEGER | 4 |
| Nome | VARCHAR(50) | 52 (50+2) |
| Descrição | TEXT | 302 |
| CustoEvento | DECIMAL(8,2) | 5 (4+1) |
| Pago | BOOLEAN | 1 |
| DescriçãoLocal | VARCHAR(100) | 102 |
| Rua | VARCHAR (75) | 77 |
| CódigoPostal | VARCHAR (20) | 22 |
| DataInicio | DATETIME | 8 |
| DataFim | DATETIME | 8 |
| ValorTotal | INTEGER | 4 |
| QuantidadeBilhetesVendidos | INTEGER | 4 |
| Preço | DECIMAL(8,2) | 5 |
| TOTAL | **-** | **594** |

**Atividade**

|  |  |  |
| --- | --- | --- |
| Atributos | Tipo | Tamanho (Bytes) |
| IdAtividade | INTEGER | 4 |
| Nome | VARCHAR(50) | 52 |
| Descrição | TEXT | 302 |
| Pago | BOOLEAN | 1 |
| CodPostal | VARCHAR(50) | 52 |
| DescriçãoLocal | VARCHAR(80) | 82 |
| Rua | VARCHAR(70) | 72 |
| Data | DATETIME | 8 |
| Lotação | INTEGER | 4 |
| IdEvento | **INTEGER** | **4** |
| IdArtista | INTEGER | 4 |
| ValorTotal | INTEGER | 4 |
| QuantidadeBilhetesVendidos | INTEGER | 4 |
| BilhetesDisponiveis | INTEGER | 4 |
| Preço | DECIMAL(8,2) | 5 |
| CustoAtividade | DECIMAL(8,2) | 5 |
| TOTAL | **-** | **607** |

**Artista**

|  |  |  |
| --- | --- | --- |
| Atributos | Tipo | Tamanho (Bytes) |
| IdArtista | INTEGER | 4 |
| Nome | VARCHAR(70) | 72 |
| Descrição | TEXT | 302 |
| IdAgente | INTEGER | 4 |
| CustoArtista | DECIMAL(8,2) | 5 |
| TOTAL | **--------** | **387** |

**Agente**

|  |  |  |
| --- | --- | --- |
| Atributos | Tipo | Tamanho (Bytes) |
| IdAgente | INTEGER | 4 |
| Nome | VARCHAR(70) | 72 |
| TelefoneAgente | VARCHAR(20) | 22 |
| EmailAgente | VARCHAR(70) | 72 |
| TOTAL | **------** | **170** |

**Staff**

|  |  |  |
| --- | --- | --- |
| Atributos | Tipo | Tamanho (Bytes) |
| IdStaff | INTEGER | 4 |
| Nome | VARCHAR(70) | 72 |
| Função | VARCHAR(20) | 22 |
| Telefone | VARCHAR(70) | 72 |
| TOTAL | **------** | **170** |

**Bilhete**

|  |  |  |
| --- | --- | --- |
| Atributos | Tipo | Tamanho (Bytes) |
| NumBilhete | INTEGER | 4 |
| Preço | DECIMAL(8,2) | 5 |
| IdEvento | INTEGER | 4 |
| IdAtividade | INTEGER | 4 |
| TOTAL | **------** | **17** |

**StaffEvento**

|  |  |  |
| --- | --- | --- |
| Atributos | Tipo | Tamanho (Bytes) |
| Staff | INTEGER | 4 |
| Evento | INTEGER | 4 |
| TOTAL | ------ | 8 |

**BilhetesVendidos**

|  |  |  |
| --- | --- | --- |
| Atributos | Tipo | Tamanho (Bytes) |
| IdBilheteVendido | INTEGER | 4 |
| IdEvento | INTEGER | 4 |
| IdAtividade | INTEGER | 4 |
| Quantidade | INTEGER | 4 |
| DataVenda | DATETIME | 8 |
| PreçoVenda | DECIMAL(8,2) | 5 |
| IdBilhete | INTEGER | 4 |
| TOTAL | ------ | 33 |

Sem povoamento, o tamanho total ocupado pela nossa base de dados é 594+607+387+170+170+17+8+33 = 1986 *bytes* ≈ 1,9 kB

Com o povoamento inicial a nossa base de dados ocupa:

594\*8 + 607\*51 + 387\*61 + 170\*70 + 170\*7 + 17\*15 + 8\*7 + 33\*100 (este valor é alterado regularmente por isso vamos supor 100) = 75 847 *bytes* ≈ 74,1 kB. Estes valores são uma estimativa sobre o primeiro mês de utilização. Se contarmos com um acréscimo de 1000% em relação aos bilhetes vendidos vamos ter um número de entradas na tabela “BilhetesVendidos” a rondar a casa dos 100\*1000 = 100 000 bilhetes vendidos. Se contarmos com um acréscimo de 50 % nos valores de atividades e eventos, teríamos a cada mês um total de mais 4 eventos e 20 atividades. No fim de um ano, são 48 eventos e 240 atividades. Podemos contar com um acréscimo de 20% nos artistas e nos respetivos agentes, o que dará um total de 73 artistas e 84 agentes registados na base de dados. Ora com estes dados, e contando que teremos cerca de 100 elementos do staff registados para ajudar nos eventos, a nossa base de dados ao fim de um ano ocuparia: 56 \* 594 + 607 \* 291 + 387 \* 73 + 170 \* 84 + 170\*100 + 17\* 100 (100 tipos de bilhetes) + 8\*100 + 33 \* 100000 = 3 571 932 *bytes* ≈ 3 488,2 kB ≈ 3,4 MB

Precisamos de ter em conta que esta é meramente uma estimativa do tamanho de ocupação da base de dados, com base em cálculos bastante rudimentares.

* 1. Indexação do Sistema de Dados

A vantagem de definir os nossos próprios índices na base de dados, em adição aos já criados automaticamente, é que estes aceleram operações de consulta e seleção. Devemos apenas criar índices para tabelas que são consultadas com elevada frequência. Devemos também ter em atenção à quantidade de entradas nas tabelas, pois quantas mais entradas uma tabela tiver, mais eficiente será o índice. Podemos desde já então excluir índices para as tabelas “Staff” e “Agente”.

A tabela “BilhetesVendidos”, como são efetuadas muitas operações de consulta e seleção, e irá ter inevitavelmente muitas entradas, é uma boa candidata a ter índices. Portanto foi criado um índice para a data de venda dos bilhetes. Pode ser útil procurar os bilhetes vendidos numa dada data.



A certo ponto da nossa base de dados, a tabela “Eventos” também poderá ser afetada com o elevado número de entradas na mesma. Pode ser útil fazer pesquisas por nome e data de começo de um evento, e por isso foram criados estes dois índices.



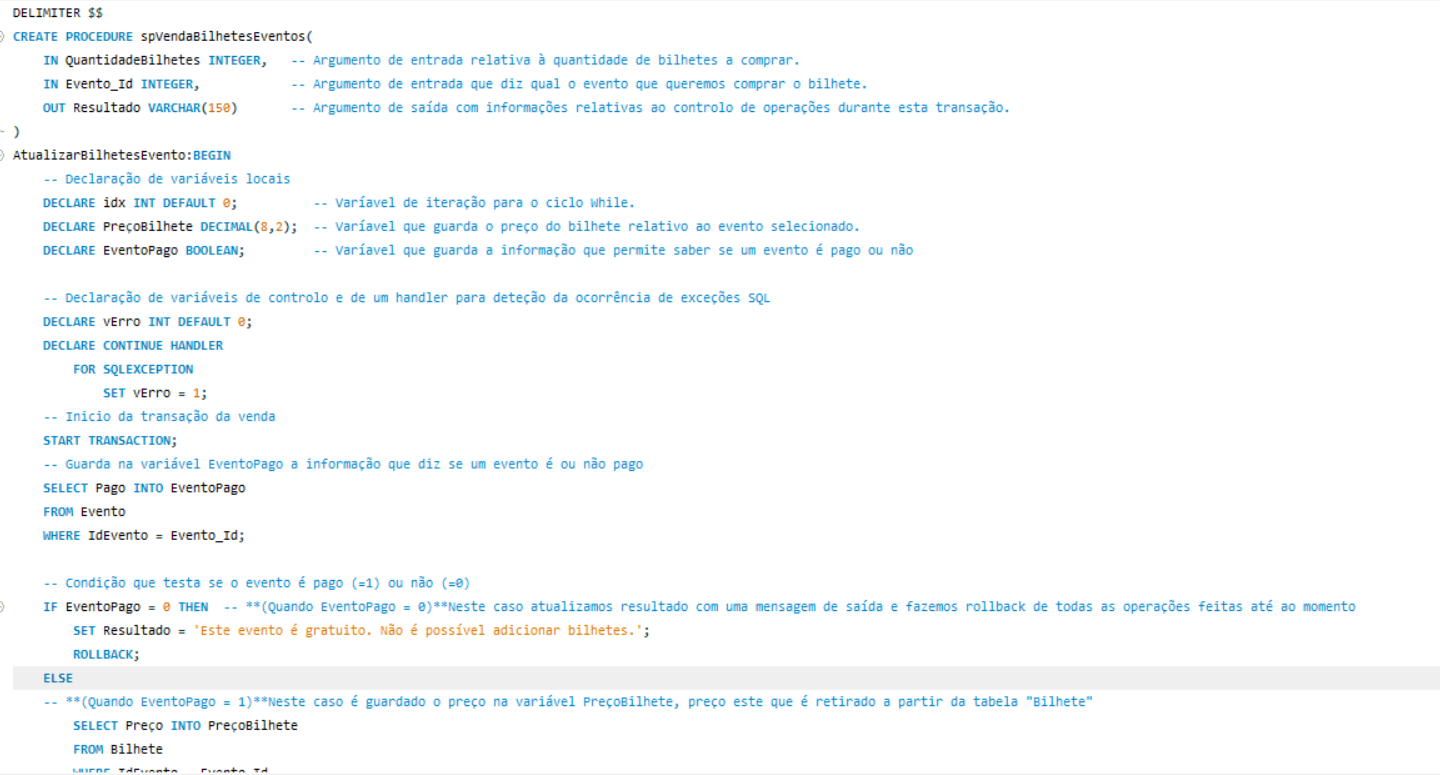
Do mesmo modo podemos, e a pensar no crescimento da nossa base de dados, foram criados dois índices iguais aos anteriores, mas desta vez para a tabela “Atividade”.



* 1. Procedimentos Implementados

De forma a conseguir simular a venda de bilhetes, quer para eventos, quer para atividades de eventos, decidimos implementar procedimentos, com operações de transações, para inserir quantidades de bilhetes para um certo identificador de evento ou de atividade à tabela “BilhetesVendidos”. À medida que desenvolvemos estes procedimentos, foram realizadas algumas alterações às tabelas “Eventos” e “Atividades”, que já foram mostradas anteriormente no trabalho, de modo a aportar informações relevantes ,como por exemplo o valor total que cada evento ou atividade arrecadou com a venda dos bilhetes, a quantidade de bilhetes disponíveis e até mesmo o preço dos bilhetes para as respetivas tabelas. De modo a explicar os procedimentos da melhor forma, vamos por etapas tentar esclarecer o raciocínio por detrás da lógica dos mesmos.

Os procedimentos são os seguintes:

**spVendaBilhetesEventos**

e **spVendaBilhetesAtividades**

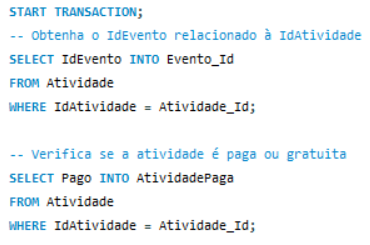
Estes são apenas excertos do código dos mesmos, uma vez que seria impraticável ter o código neste documento. O mesmo está disponível na pasta “Scripts SQL” com o nome 202324-UM-LCC-BD-G04-StoredProcedures.sql.

Começando pelo primeiro procedimento.

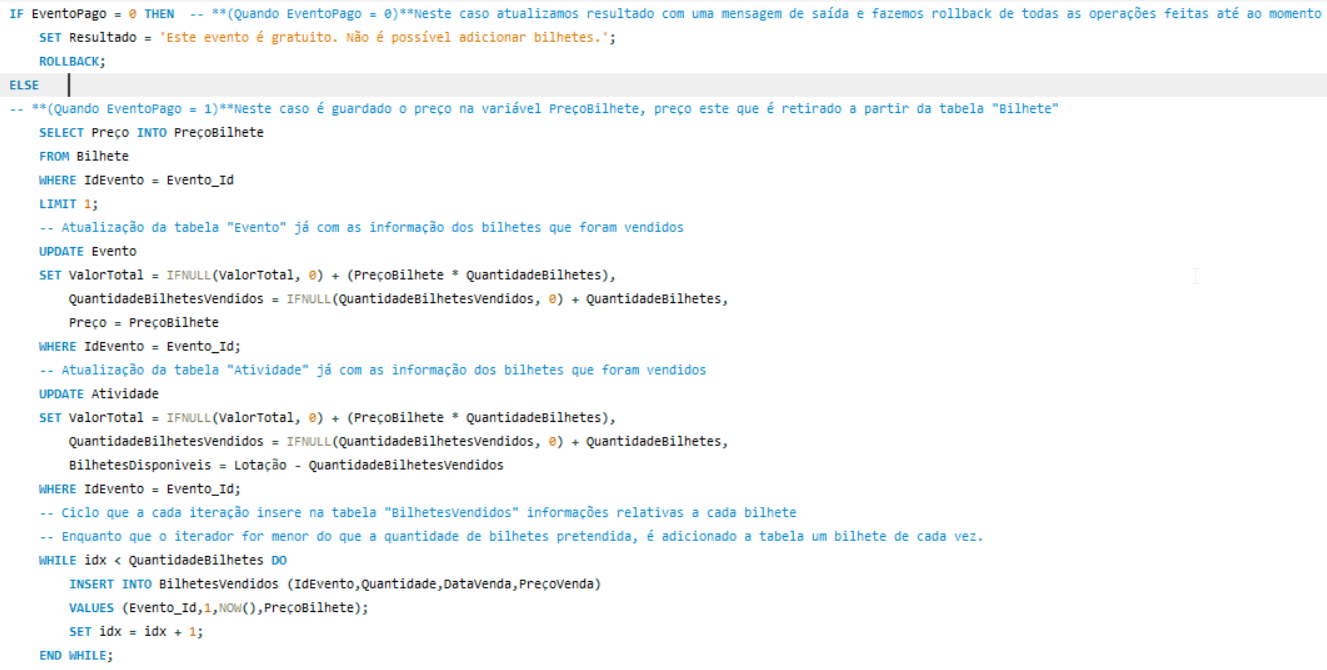
Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

É declarada a assinatura do procedimento, com a definição dos argumentos de entrada e saída do procedimento. A seguir a isso, foram declaradas uma série de variáveis que vão ser utilizadas ao longo do mesmo, assim como variáveis de controlo de erros.

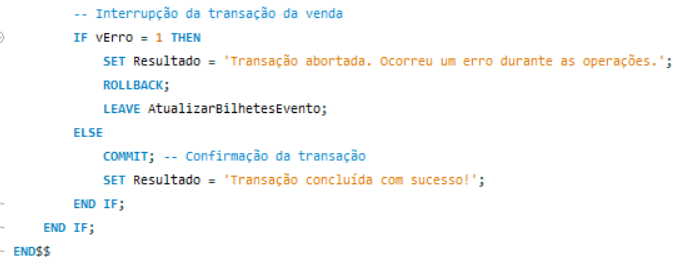


De seguida foi indicado que a transação começa neste ponto. Foi utilizada uma transação para assegurar a integridade dos dados que vão ser introduzidos. Na vertente de vendas é comumente utilizado este recurso, por estas utilizarem o padrão *ACID*, ou seja, garantem a atomicidade de dados, a consistência, o isolamento e durabilidade.

Após este passo, foram atribuídas às variáveis, anteriormente criadas, informações relativas ao identificador do evento e se o mesmo é pago ou não.

De seguida, é verificado se o evento não é pago. Se não o for, é apresentada uma mensagem de erro e fazemos *rollback* às operações até aí realizadas, ou seja, desfazemo-las. Se o evento for pago, prosseguimos com a venda dos bilhetes. É atualizada a tabela “Evento”, de modo que a quantidade de bilhetes vendidos aumente, e o preço na tabela seja atualizado para o preço do bilhete correspondente ao evento. É também atualizada a tabela “Atividade”, pois comprando bilhete para um evento, todas as atividades ficam pagas. Nesta tabela, é atualizado também o campo de bilhetes disponíveis, que decresce com a quantidade de bilhetes vendidos. A forma que arranjamos para inserir, um a um, os bilhetes na tabela “BilhetesVendidos”, foi criar um ciclo, que vai começar em 0 e vai um a um, até ao número de bilhetes pretendidos para venda. A cada iteração deste ciclo, são adicionadas as informações à tabela “BilhetesVendidos”.

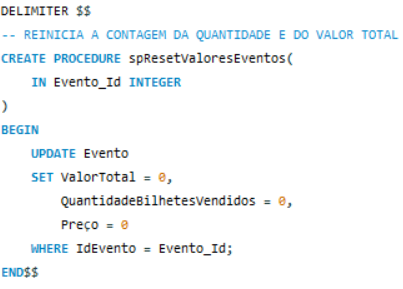
Esta pode não ser a melhor forma de fazer isto, pois se a quantidade de bilhetes de uma só vez for muito elevada, perdemos performance e eficácia na nossa base de dados.



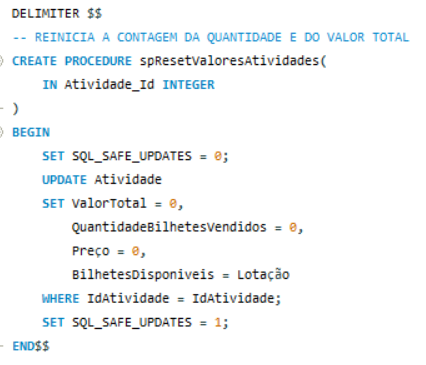
Se não existirem mais erros, é feita a confirmação da transação e mostrado o resultado com mensagem de sucesso.

A explicação do procedimento para a venda de bilhetes para as atividades é análoga à anterior.

Foram também desenvolvidos procedimentos para realizar a limpeza da base de dados, tendo estes que ser utilizados de forma responsável e segura. Como apenas o administrador tem acesso a estes procedimentos, à partida não existe qualquer problema. Os procedimentos são:

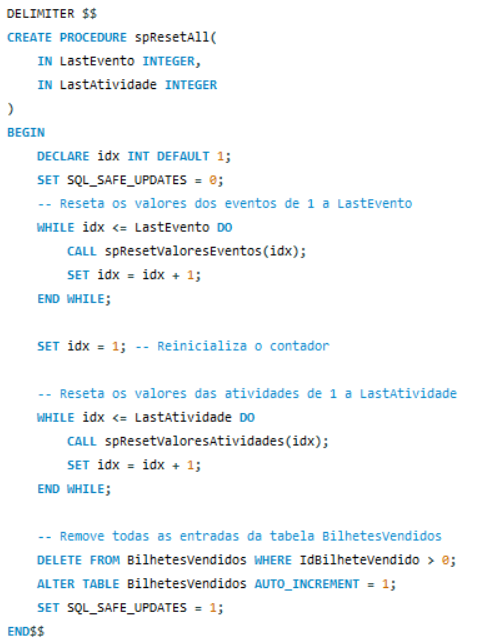


Que dado um identificador de evento, faz um *reset* (coloca os valores pretendidos a 0) aos valores da tabela do mesmo.



Que dado um identificador de atividade, faz um *reset* aos valores da tabela da mesma.

Estes dois últimos procedimentos, embora possam ser executados individualmente, foram desenvolvidos para serem chamados por um procedimento “spResetAll”, que como o nome indica faz uma limpeza geral. Faz um *reset* aos valores das tabelas “Evento” e “Atividade” e apaga todas as entradas da tabela “BilhetesVendidos”. Este procedimento recebe dois argumentos. O ID do evento até onde queremos que sejam reinicializados os valores e o mesmo para o ID da atividade. São utilizados dois ciclos, um para percorrer desde 0 até ao número do último ID do evento, e a cada iteração, executa uma vez o procedimento “spResetValoresEvento” para o ID em que a iteração está, e o mesmo se verifica para o ciclo responsável por chamar o procedimento “spResetValoresAtividades”. Por fim apaga da tabela “BilhetesVendidos” todas os registos.

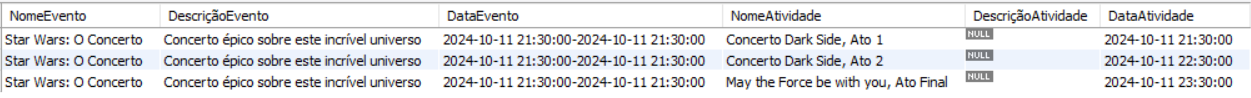
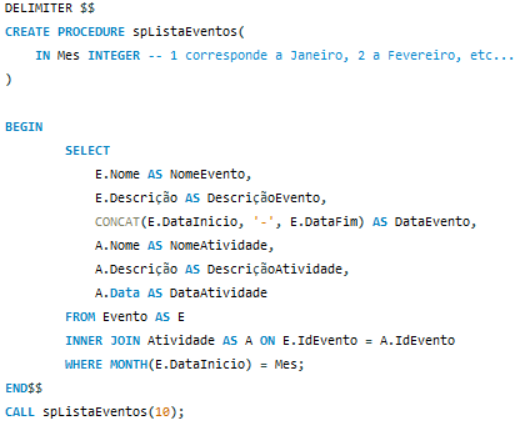
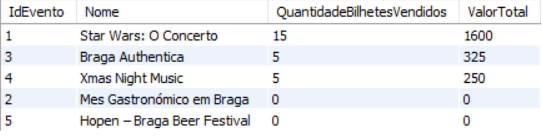
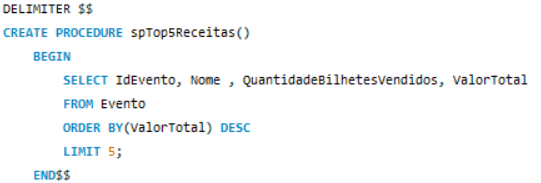


Posteriormente foram criados também dois procedimentos: spTop5Receitas e spListaEventos, que pretendem responder a requisitos de manipulação recolhidos anteriormente.

Uma imagem com texto, captura de ecrã, Tipo de letra, branco

Descrição gerada automaticamente

A partir destes requisitos resultaram os seguintes procedimentos:





Foram também criados dois *triggers* e uma função. Os *triggers* são utilizados para atualizar o custo de um evento, sempre que adicionamos uma atividade correspondente ao mesmo e para não permitir a inserção de datas incorretas na tabela evento.

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamenteA função desenvolvida recebe um recebe um ano, e devolve o total de gastos por ano, de todos os eventos juntos.

* 1. Plano de segurança e recuperação de dados

A implementação de um plano é um passo tão importante quanto a implementação bem-sucedida de uma base de dados. Por isso, decidimos fazer este plano para garantir a segurança e integridade dos dados armazenados.

O primeiro passo deste plano, é assegurar a segurança dos dados, e só depois ter um plano, caso o primeiro passo falhe, de recuperação de dados. O plano de segurança dos dados passa por estes pontos:

1. **Realização de backups regulares**: Serão realizados backups diários, em horários de menor tráfico de dados na nossa base de dados, em mais do que 3 localizações, para reduzir o risco da perda total dos dados em caso de uma ocorrência inesperada. As 3 localizações serão: um servidor privado para armazenar a base de dados e os seus backups, um computador na posse do administrador da base de dados e por último um backup numa plataforma de clouding.
2. **Contratação de um engenheiro de Segurança de Redes**: De forma a prevenir ataques e identificar possíveis ameças à nossa base de dados, será contratado um especialista nesta área que irá implementar medias proativas e reativas, mantendo este plano de segurança o mais atualizado possível.
3. **Controlo do acesso à base de dados**: Foram definidas regras muito específicas sobre quem pode ou não aceder à base de dados, e quais os privilégios que têm sobre a mesma.

No caso de uma quebra neste plano, foram implementados mais 2 pontos que permitem a recuperação dos dados que foram comprometidos:

1. **Procedimentos pré-definidos de recuperação dos dados:** No caso de falha do sistema ou perca dos dados, foram definidos uma série de passos a seguir para restaurar os dados a partir dos backups.
2. **Testes de recuperação frequentes:** Vão existir frequentes simulações de perda de dados, para otimizar o processo de restauro dos mesmos.
3. Conclusões e Trabalho Futuro

O presente projeto teve como objetivo a criação de uma base de dados para a gestão de um calendário de eventos. Uma vez realizadas todas as etapas para a produção de uma base de dados completamente operacional, é necessário analisar a prestação de cada etapa, sendo possível retirar conclusões.

Este projeto tem por base a criação de uma "história" para suportar toda a ideia e a necessidade da criação de uma base de dados. Uma vez realizada esta parte, prosseguiu-se para a listagem dos objetivos, análise da viabilidade do projeto e, ainda, a elaboração de um diagrama de GANTT, representando esta última um desafio para a realização deste projeto. A elaboração de um diagrama de GANTT, embora importante para o tipo de trabalhos com prazos, não foi bem-sucedida no decorrer da realização deste trabalho, talvez pela falta de compreensão do funcionamento dos mesmos. É, por isso, um ponto de destaque a melhorar para futuros trabalhos.

A segunda etapa a ser realizada foi a criação de um modelo conceptual, a partir dos requisitos recolhidos, que viria a servir de base para mais tarde. A criação de entidades e respetivos relacionamentos foi algo pensado e concretizado com sucesso. Considerando, agora, em retrospetiva, poderia ter sido introduzida uma maior complexidade à base de dados, que se poderia ter traduzido num maior número de ações a realizar com a mesma.

A partir do modelo conceptual, foi possível realizar a conversão para o modelo lógico, de acordo com regras pré-definidas, e que, portanto, decorreu também com bastante facilidade. A segunda dificuldade surgiu quando, ao validar o modelo, foi necessária a utilização do software “RelaX” para a criação de árvores de álgebras relacionais. Apesar das inúmeras tentativas para a criação destas árvores, o software apresentava erros incoerentes, provavelmente pela inexperiência dos elementos do grupo na utilização da plataforma. Por este motivo, surgiu a necessidade de reduzir o nível das *queries* à base de dados. Este é o segundo ponto a melhorar em trabalhos futuros.

Por último, foi realizado um *script* de criação da base de dados na linguagem *SQL*. Este último passo tratou de converter para algo funcional tudo o que tinha sido realizado até ao momento. Foi talvez a parte mais entusiasmante do trabalho. A realização de *queries*, os procedimentos e todos os métodos que foram implementados permitiu ver a base de dados em ação, tendo sido bastante motivador.

Sem dúvida que o projeto em questão abriu os horizontes a este paradigma da tecnologia, e foi importante para perceber as saídas que existem nesta área.

Relativamente a trabalhos futuros, o objetivo é não só melhorar os dois pontos supracitados, como também aprimorar a técnica nesta área. Para além disso, seria interessante, fora do contexto desta disciplina, proceder à criação de uma aplicação que confira uma interface visual à nossa base de dados.

Por fim, pode concluir-se que o sistema criado encontra-se apto para o objetivo proposto inicialmente: a gestão de um calendário de eventos.

Referências (ESTILO APA)

* Lucidchart O que é um diagrama entidade relacionamento? Lucidchart. Available at: <https://www.lucidchart.com/pages/pt/o-que-e-diagrama-entidade-relacionamento> (Accessed: 03 November 2023).
* Connolly, T., Begg, C., Database Systems: A Practical Approach to Design, Implementation, and Management, Addison-Wesley, Global Edition, 26 Sep 2014. ISBN-10: 1292061189, ISBN-13: 978-1292061184.

Lista de Siglas e Acrónimos

BD Base de Dados

SGBD Sistema Gestão Base de Dados

CMB Câmara Municipal de Braga

ER Entidade Relação

NIF Número de Identificação Fiscal