

**Universidade do Minho**

Mestrado Integrado em Engenharia Informática

Licenciatura em Ciências da Computação

**Unidade Curricular de**

**Bases de Dados**

Ano Lectivo de 2016/2017

**IberoTrem – reserva de viagens online**

**João da Cunha Coelho (a74859), José Miguel Ribeiro da Silva (a74601), Luís Miguel Moreira Fernandes (a74748), Pedro João Novais da Cunha (a73958)**

Novembro, 2016

<<Mês, Ano>>

**BD**

|  |  |
| --- | --- |
| Data de Recepção |  |
| Responsável |  |
| Avaliação |  |
| Observações |  |

**IberoTrem – reserva de viagens online**

**João da Cunha Coelho (a74859), José Miguel Ribeiro da Silva (a74601), Luís Miguel Moreira Fernandes (a74748), Pedro João Novais da Cunha (a73958)**

Novembro, 2016

<</opcional Dedicatória>>

# Resumo

<<O resumo tem como objectivo descrever de forma sucinta o trabalho realizado. Deverá conter uma pequena introdução, seguida por uma breve descrição do trabalho realizado e terminando com uma indicação sumária do seu estado final. Não deverá exceder as 400 palavras.>>

**Área de Aplicação:** <<Identificação da Área de trabalho. Por exemplo: Desenho e arquitectura de Sistemas de Bases de Dados.>>

**Palavras-Chave:** <<Conjunto de palavras-chave que permitirão referenciar domínios de conhecimento, tecnologias, estratégias, etc., directa ou indirectamente referidos no relatório. Por exemplo: Bases de Dados Relacionais, Gestão de Índices, JAVA, Protocolos de Comunicação.>>

# Índice

[1. Introdução 1](#_Toc467333436)

[1.1. Contextualização 1](#_Toc467333437)

[1.2. Apresentação do Caso de Estudo 1](#_Toc467333438)

[1.3. Motivação e Objectivos 2](#_Toc467333439)

[1.4. Estrutura do Relatório 2](#_Toc467333440)

[2. Sugestões para Escrita do Relatório 3](#_Toc467333441)

[2.1. Sugestões Gerais 3](#_Toc467333442)

[2.2. Termos Estrangeiros 3](#_Toc467333443)

[2.3. Tabelas e Figuras 3](#_Toc467333444)

[2.4. Siglas e Acrónimos 6](#_Toc467333445)

[2.5. Referências Bibliográficas 6](#_Toc467333446)

[2.6. Tipo de Ficheiro 6](#_Toc467333447)

[3. Conclusões e Trabalho Futuro 7](#_Toc467333448)

**Anexos**

[I. Anexo 1 9](#_Toc535644882)

# Índice de Figuras

[Figura 1 - Ilustração de inserção de uma figura e legenda. 3](#_Toc535433491)

# Índice de Tabelas

Tabela 1 – Identificação das entidades

Tabela 2 – Identificação dos relacionamentos

Tabela 3 – Identificação e associação dos atributos com as entidades

1. Introdução

<<Este primeiro capítulo deverá ter obrigatoriamente as subsecções abaixo apresentadas.>>

* 1. Contextualização

A IberoTrem é uma empresa de transporte ferroviário sediada em Portugal, que estabelece a ligação por linhas férreas entre várias cidades do Norte de Portugal e de Espanha (Galiza). Foi criada em 2001 com o intuito de se tornar a primeira companhia *low cost* a operar na região, proporcionando bilhetes para as suas viagens a um custo semelhante ao dos bilhetes com desconto por compra antecipada praticado pelas grandes companhias. Desde início o projeto foi visto como viável, dado que é relativamente frequente uma pessoa deparar-se com a necessidade de comprar um bilhete para uma viagem que não estava prevista com uma semana (ou mais) de antecedência. Por outro lado, também não é raro um indivíduo atrasar a marcação de uma viagem com receio que, até à data, algo lhe altere os planos. Porém, pode acontecer que, quando confirme a intenção de viajar, já seja tarde para usufruir dos descontos de compra antecipada. A IberoTrem oferece assim uma alternativa em conta a quem esteja em situações semelhantes (e não só).

* 1. Apresentação do Caso de Estudo

Inicialmente, os serviços de transporte da empresa eram prestados por apenas três linhas: Linha do Minho, Linha de Vigo e Linha de Chaves. Posteriormente foi levado a cabo um processo de alargamento da oferta, que acrescentou dois troços aos serviços da companhia (Linha de Miranda e Linha da Galiza), pelo que a direção decidiu estender o serviço de reservas de viagens à web.

Um grupo de engenheiros informáticos foi contratado, sendo-lhes pedido que implementassem no site da empresa, até então meramente informativo, uma secção própria para a reserva das viagens. Tendo por base os requisitos especificados pela direção da companhia, foi estipulado que o site deveria permitir que os clientes, caso o desejassem, se inscrevessem (nome, número do cartão de cidadão, data de nascimento e contactos - telefone e email), dispensando assim o fornecimento das informações pessoais na próxima compra online de bilhetes. À reserva ficará associado o preço, o lugar reservado no comboio e a data prevista para a viagem, a qual naturalmente possui um preço e uma hora de partida e de chegada. Por sua vez, cada comboio, além do número identificativo, possui ainda o registo dos seus lugares. Dada a campanha que a empresa leva a cabo desde a sua origem, no sentido de promover o uso do comboio pelos mais jovens, o preço da reserva poderá ser diferente do preço da viagem, uma vez que é feito um desconto de 25% aos utilizadores com menos de 25 anos.

Por fim, a empresa especificou à equipa de informáticos que o envio do comprovativo da viagem reservada seria realizado por email, recorrendo ao endereço eletrónico disponibilizado pelo cliente que efetuou a reserva.

* 1. Motivação e Objectivos

Numa empresa de transportes ferroviários, torna-se difícil controlar os lugares disponíveis sem ajuda de um sistema informático. Com uma base de dados melhora-se a eficiência de trabalho e a organização da informação. A informação fica mais fácil de gerir, e a manipulação dos dados é sempre efetuada de forma segura. Com a possibilidade de reserva online, torna-se essencial a existência da informação em formato digital, e uma base de dados é a melhor opção.

Os principais objetivos na elaboração deste projeto são:

* Construir uma base de dados, de forma eficiente, que contenha a informação referente às reservas de lugares. Cada uma correspondente a um cliente e uma viagem.
* Comparar com situações realistas de forma a perceber que problemas podem aparecer e as consequências que esses mesmos problemas geram.
* Assimilar todos os conteúdos de forma a saber como construir uma base de dados relacional. Estando esta funcional e devidamente organizada.
  1. Estrutura do Relatório

Após a apresentação do caso de estudo e da análise dos respetivos requisitos, segue-se a construção do modelo conceptual. Estando este devidamente validado e adaptado ao problema que se coloca, passar-se-á ao modelo lógico, construído com base no anterior. Por fim, a *forward engineering* permitir-nos-á obter o modelo físico a partir do lógico, pelo que passaremos a ter uma base de dados capaz de responder a *queries*. Na próxima secção serão apresentadas as etapas de construção destes modelos, seguindo-se uma secção final com conclusões sobre o trabalho realizado, nomeadamente os seus pontos fortes e fracos.

1. Sugestões para Escrita do Relatório
   1. Sugestões Gerais

<<O presente documento deverá servir de base para a escrita do relatório do trabalho realizado.>>

<<O tipo de letra a utilizar deverá ser Arial.. Porém recomenda-se em situações de escrita de excertos de programas a utilização do tipo de letra Courier New.>>

<< Alguns estilos documento: Heading1, Heading2, Heading3, Normal e Footnote Text; foram especialmente modificados para os relatórios da presente disciplina.>>

<<Os formatos e estilos de letra não devem estar constantemente a ser modificados ao longo do relatório. Tal situação dará origem a um relatório com um formato e apresentação muito heterogénea e com um aspecto pouco consistente.>>

* 1. Termos Estrangeiros

<<Os termos estrangeiros utilizados deverão ser apresentados num formato diferente do resto do texto, por exemplo: *Data Warehouse* (em itálico) ou "Data Warehouses" (entre aspas), devendo ser evitados sempre que se conheça uma tradução correcta para português. Para validação desses termos existem vários dicionários no mercado que poderão ser úteis.>>

* 1. Tabelas e Figuras

<<Caso seja necessário introduzir figuras ou tabelas no corpo do documento, estas devem seguir os formatos que se apresentam de seguida. Qualquer figura ou tabela deverá ter uma legenda associada, devendo esta estar correctamente apresentada no índice respectivo no início do relatório.>>

* 1. Siglas e Acrónimos

<<A utilização de siglas ou acrónimos deverão, tal como os termos estrangeiros, ser feita com base no seguinte formato: *Bases de Dados* (BD). Todas as siglas e acrónimos deverão ser apresentadas numa secção própria, no início (a seguir aos índices) ou no final (a seguir ao capítulo das conclusões e trabalho futuro) do relatório.

* 1. Referências Bibliográficas

<<A forma de apresentação das referências bibliográficas deverão estar de acordo com as regras definidas pela IEEE. Consultar www.ieee.org>>

* 1. Tipo de Ficheiro

<<O relatório poderá ser enviado para o regente da disciplina por correio electrónico num dos seguintes formatos: html, word ou pdf>>

* 1. Construção do modelo conceptual

1. **Identificação das entidades**

Analisando os requisitos da base de dados levantados na apresentação do caso de estudo, chegou-se à seguinte tabela de entidades:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome da Entidade** | **Descrição** | **Sinónimo** | **Ação/Ocorrência** |
| Cliente | Termo geral que descreve todas as pessoas que utilizam os serviços de viagens prestados pela companhia IberoTrem. | Utilizador | Cada cliente pode efetuar várias reservas (limitadas à marcação de apenas um lugar). |
| Reserva | Termo geral que descreve o serviço de marcação de um lugar numa viagem prestada pela companhia IberoTrem. | Marcação | Cada reserva está associada a apenas uma viagem, podendo vários utilizadores fazer reservas para a mesma viagem. |
| Viagem | Termo geral que descreve a deslocação entre uma origem e um destino, durante um dado intervalo de tempo. | Trajeto, Percurso, Deslocação | Cada viagem é realizada por um único comboio, entre um destino e uma origem, tendo associada a si várias reservas. |
| Comboio | Termo geral que descreve o meio de transporte utilizado pela companhia IberoTrem para prestar os seus serviços. | - | Cada comboio efetua várias viagens. |
| Estação | Termo geral que descreve o lugar de embarque e desembarque dos clientes da companhia IberoTrem. | Paragem | Cada estação é simultaneamente origem e destino de várias viagens. |

Tabela 1 – Identificação das entidades

1. **Identificação dos relacionamentos**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entidade** | **Multiplicidade** | **Relação** | **Multiplicidade** | **Entidade** |
| Cliente | 1…1  (Dúvida) | Efetua | 0…N | Reserva |
| Reserva | 0…N | Relativa a | 1…1 | Viagem |
| Viagem | 0…N | Efetuada por | 1…1 | Comboio |
| Viagem | 0…N | Tem origem na | 1…1 | Estação |
| Viagem | 0…N | Tem destino na | 1…1 | Estação |

Com base nas entidades definidas, procedemos à análise das relações que entre elas se estabelecem de modo a que o sistema de base de dados possa responder às necessidades da empresa. Daqui resultou a seguinte tabela:

Tabela 2 – Identificação dos relacionamentos

1. **Identificação e associação de atributos com entidades ou relacionamentos e determinação do domínio dos atributos**

Nesta fase, identificaremos os atributos que constarão na base de dados, justificando a sua escolha. Estes poderão estar associados a entidades ou relacionamentos, mas no nosso modelo conceptual todos são referentes a entidades.

No que ao *Cliente* diz respeito, a informação que consideramos importante preservar foi: o número do cartão de cidadão, para termos um elemento único identificativo da pessoa; o nome, para aproximar a empresa do cliente com um tratamento mais pessoal; a data de nascimento, para avaliar se está sujeito a descontos em função da idade ou não; o contacto, nomeadamente o endereço de email, usado para enviar a confirmação da reserva e, caso haja registo no site, para fazer o login, e o telefone, útil para trocar informações com o cliente, constituindo uma alternativa ao email.

Já para a *Reserva,* os atributos escolhidos foram o lugar reservado no comboio da viagem, que é o elemento essencial na reserva de qualquer viagem e como tal não poderia deixar de fazer parte, a data da viagem reservada, importante para complementar a especificação da viagem e para controlar o período até ao qual é possível cancelar a reserva, o preço, derivado do preço da viagem dado que pode estar sujeito a desconto, e o id\_reserva, que torna cada reserva única.

Quanto à *Viagem*, são quatro os atributos considerados importantes: o id\_viagem, que atribui um valor único a uma viagem, o preço, em função do itinerário da viagem, a hora de partida e a hora de chegada, que juntas especificam, dentro do trajeto previsto, qual é a viagem referida.

A *Estação* foi caracterizada com recurso ao id\_estação, para garantir a unicidade de cada estação, à cidade onde se localiza e ao nome da estação, que varia dada a possibilidade de existirem várias estações por cidade.

Por último, a entidade *Comboio* caracterizou-se com recurso a uma identidade (id\_comboio), que garante que um comboio é único, e aos números dos lugares do comboio especificado para a viagem, que é importante para saber quais os lugares que ainda podem ser reservados.

Após estarem definidos os atributos relevantes para cada entidade, procedeu-se à análise do domínio – conjunto de valores que um atributo pode assumir - de cada um. Começando pelos atributos associados ao Cliente, o CC assume o valor de uma string de 9 caracteres predominantemente numéricos – os 8 primeiros – com o último caracter alfabético, dado que os documentos de identificação espanhóis são constituídos por 8 números e uma letra final, contrastando com os 8 números do CC português; a data de nascimento é do tipo Date e pode variar entre 01/01/1900 e o dia atual, seguindo sempre o formato dd/mm/aa; o nome será uma string com no máximo 64 caracteres; o telefone será representado por um inteiro com 9 dígitos, uma vez que quer em Portugal quer em Espanha os contactos telefónicos seguem esta regra; o endereço de email será uma string com no máximo 64 caracteres (emailexemplo@exemplo.com).

Para os atributos da *Reserva* usamos um inteiro com valor máximo de 99 (2 dígitos), considerando-o suficiente para suportar o número de reservas da empresa, para o id\_empresa. O preço será um float de valor positivo, o lugar um inteiro até 99 (2 dígitos) e a data será do tipo *Date,* com o limite inferior sendo o próprio dia e assumindo o formato dd/mm/aa.

Quanto à *Viagem*, o atributo id\_viagem será um inteiro até 99 (2 dígitos), o preço um float positivo e as horas de chegada e partida uma string com formato hh:mm.

Para a *Estação*, teremos um id\_estação inteiro, até 99 (2 dígitos), uma string a representar o nome da cidade onde se localiza e outra string representando o nome da estação, ambas com tamanho máximo de 32 carateres.

Por fim, o id\_comboio é representado por um inteiro até 9 (1 dígito) e os lugares, atributo multi-valor, são representados por inteiros até 99 (2 dígitos), dada a reduzida capacidade dos comboios da empresa.

A junção das escolhas enunciadas anteriormente conduziu à seguinte tabela:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Entidade** | **Atributos** | **Descrição** | **Domínio** | **Nulls** | **Multi-valorado** | **Derivado** | **Valor default** |
| Cliente | CC  Data de Nasc.  Nome  Contacto  Telefone  Email | Número do CC, que identifica o cliente;  Data de Nascimento do cliente;  Nome do cliente;  Contacto telefónico do cliente;  Endereço de email do cliente; | String até 9 caracteres;  Data (1900-atualidade) com formato dd/mm/aa;  String até 64 caracteres;  Inteiro com 9 dígitos;  String até 64 caracteres; | Não  Não  Não  Sim  Não | Não  Não  Não  Não  Não | Não  Não  Não  Não  Não | Não  Não  Não  Não  Não |
| Reserva | Id\_reserva  Preço  Lugar  Data | Identifica a reserva (valor único);  Preço da reserva do lugar numa viagem;  Lugar no comboio reservado;  Data da viagem reservada; | Inteiro até 99;  Float (positivo);  Inteiro até 99;  Data (atualidade-) com formato dd/mm/aa | Não  Não  Não  Não | Não  Não  Não  Não | Sim  Não  Não  Não | Não  Não  Não  Não |
| Viagem | Id\_viagem  Hora Partida  Hora Chegada  Preço | Identifica a viagem (valor único);  Hora a que o comboio da viagem parte da estação origem;  Hora a que o comboio da viagem chega à estação destino;  Preço da viagem; | Inteiro até 99;  String (hh:mm)  String (hh:mm)  Float (positivo); | Não  Não  Não  Não | Não  Não  Não  Não | Não  Não  Não  Não | Não  Não  Não  Não |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Estação | Nome  Cidade  Id\_estação | Nome da estação utilizada pelo comboio da viagem;  Nome da cidade onde se localiza a estação;  Identifica a estação (valor único); | String até 32 caracteres;  String até 32 caracteres;  Inteiro até 99; | Não  Não  Não | Não  Não  Não | Não  Não  Não | Não  Não  Não |
| Comboio | Lugares  Id\_comboio | Números dos lugares do comboio;  Identifica o comboio (valor único); | Inteiros até 99;  Inteiro até 9; | Sim  Não | Sim  Não | Não  Não | Não  Não |

Tabela 3 - Identificação e associação dos atributos com as entidades.

1. **Determinação das chaves candidatas, primárias e suplentes das entidades**

Na entidade *Cliente* podemos facilmente verificar que temos apenas 3 atributos candidatos - o CC, Telefone e Email – pois são os únicos a garantirem a unicidade de cada ocorrência dessa entidade. Tendo em conta ainda os critérios para a escolha da melhor chave candidata, que passará a ser a primária, notamos que tanto o atributo Telefone como o Email são atributos mais prováveis de ter os seus valores alterados do que o CC do Cliente, fazendo desta maneira com que este atributo seja a escolha lógica como chave primária. Telefone e Email serão então chaves suplentes.

Na entidade *Reserva*, mais uma vez, facilmente se verifica a não unicidade que os atributos Preço, Data e Lugar garantiriam. Poderíamos então pensar numa chave primária composta com Data e Lugar, mas devido ao facto de poderem ocorrer duas viagens no mesmo dia (Data igual) e em que o mesmo lugar fosse ocupado em ambas, excluímos esta hipótese, não nos restando alternativa senão a criação de um atributo Id\_reserva, que garantirá a unicidade de cada ocorrência. Este atributo será apenas um inteiro, logo terá sempre o menor valor possível.

Na entidade *Viagem* temos atributos como Hora Partida e Hora Chegada, bem como Preço, atributos estes que logo à partida nunca nos garantiriam unicidade, visto o preço ser igual para várias viagens e visto serem realizadas viagens com a mesma hora de partida e chegada em dias diferentes. Portanto, achou-se por bem a criação de um atributo Id\_viagem que, tal como Id\_reserva, será um inteiro que tomará sempre o menor valor possível de forma a assegurarmos a unicidade das ocorrências.

Na entidade *Estação* conseguimos perceber que os atributos Nome e Cidade sozinhos não nos garantem a unicidade necessária, porém se considerarmos uma chave primária composta por ambos os atributos, verifica-se a tal unicidade requerida. No entanto, e por uma questão de eficiência, decidimos não usar esta chave composta como chave primária e criar para isso um atributo Id\_estação, que, novamente, será um inteiro que tomará sempre o menor valor possível e que garanta a unicidade, sendo este um atributo que nos facilitará a comparação e nos garantirá um aumento na eficiência. Poderemos também considerar a chave primária composta (Nome, Cidade) como uma chave suplente.

Na entidade *Comboio* temos apenas o atributo Lugares que é logo descartado como candidato visto que vários comboios podem ter o mesmo nº de lugares, abrindo assim vaga à criação de um atributo Id\_comboio que, mais uma vez, será um inteiro que irá tomar sempre o menor valor possível que garanta unicidade.

1. **Procura de redundâncias no modelo**

Para se verificar que não existem quaisquer evidências de redundância é necessária uma averiguação que assenta em três passos:

1. **Examinar relações 1:1**. Dado que o nosso modelo não possui nenhuma relação 1:1 podemos concluir que, no que diz respeito a este passo, não existe qualquer suspeita de redundância.
2. **Remoção de relações redundantes**. Uma das situações que levanta mais suspeitas no que diz respeito à presença de relações redundantes são os modelos conceptuais onde existe mais do que um caminho entre duas entidades. No entanto, isto não implica necessariamente redundância, como podemos verificar pelo nosso modelo. Apesar de se estabelecer duas relações entre a entidade ‘Viagem’ e a entidade ‘Estação’, estamos perante duas associações diferentes e, neste caso particular, até mesmo opostas. Podemos concluir assim que, também no que diz respeito a este passo, não existe qualquer suspeita de redundância, já que todas as relações presentes no modelo lógico vêm proporcionar informação adicional e única às relações entre entidades.
3. **Considerar a dimensão do tempo**. Neste modelo conceptual, apenas a entidade *Estação* possui dois caminhos alternativos, porém estes representam relações diferentes tal como referimos anteriormente.
4. **Validação do modelo conceptual contra transações**

Neste passo deveremos assegurar que o modelo conceptual suporta os possíveis requerimentos da companhia IberoTrem. De seguida enumeraremos algumas das transações que na nossa opinião, caso efetuadas com sucesso, demostram a validade do modelo conceptual em questão.

* **Registar Cliente**.Para registar um cliente é necessário guardar as seguintes informações sobre o mesmo: Cartão de cidadão (ou documento de identificação equivalente, que permite distinguir o cliente de outros), nome, data de nascimento e contactos do mesmo. No modelo conceptual desenvolvido, a entidade Cliente tem como atributos: **CC, nome, data de nascimento** e **Contacto** composto por **email** e **telefone**.Pode concluir-se que a entidade Cliente tem todos os atributos necessários para a transação.
* **Registar Viagem.** Para efetuar o registo de uma viagem, é necessário guardar informações relativas à hora de partida e hora de chegada. Também é relevante que se guarde o preço da mesma e um identificador único que distinga a viagem em questão. Para além destas informações, é necessário saber quais as estações de origem e destino e qual o comboio que efetua a viagem. No modelo conceptual desenvolvido, a entidade Viagem os seguintes atributos: **id\_viagem, hora\_chegada, hora\_partida** e **preço.** Existem ainda os relacionamentos: **N viagens são efetuadas por 1 comboio, N viagens têm como destino 1 estação** e **N viagens têm como origem 1 estação.** Com isto pode concluir-se que a entidade Viagem tem todos os atributos necessários e está relacionada com as entidades relevantes para esta transação.
* **Registar Reserva.** Para registar reserva é necessário guardar as seguintes informações: identificador único para a distinguir, lugar que fica reservado, data da viagem e ainda o preço da reserva. Também é preciso saber a que viagem se refere e que cliente efetuou a reserva. No modelo conceptual desenvolvido a entidade Reserva tem como atributos o **lugar,** o **id\_reserva** e a **data**. Tem ainda, como atributo derivado, o **preço.** Existem os relacionamentos **1 Cliente efetua N reservas** e **N reservas são relativas a 1 Viagem.** Podemos concluir que a entidade Reserva possui os atributos necessários e está relacionada com as entidades relevantes para esta transação.

O resultado final dos 6 passos abordados anteriormente está representado no esquema conceptual que a seguir se apresenta na figura seguinte:

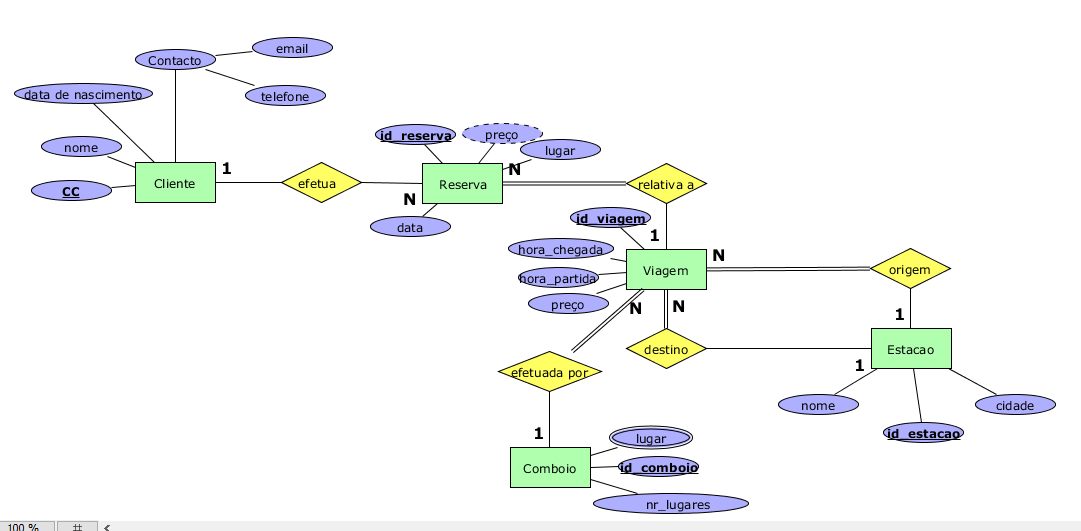


Figura 1 – Modelo conceptual

* 1. Construção do modelo lógico

Nesta secção esclareceremos como se construiu o modelo lógico, tendo por base a conversão do modelo conceptual previamente abordado.

1. **Derivar relacionamentos para o modelo lógico**
2. **Entidades fortes**

As entidades fortes que constituem o nosso modelo conceptual são: Cliente, Reserva, Viagem, Estação e Comboio. Para cada uma destas entidades criámos uma relação que inclui os seus respetivos atributos. A representação é feita seguindo o que a metodologia recomenda.

**Cliente** (CC, Nome, Data\_de\_Nascimento, Email, Telefone)

**Primary key** CC

**Reserva** (Id\_reserva, Preço, Lugar, Data)

**Primary key** Id\_reserva

**Viagem** (Id\_viagem, Hora\_chegada, Hora\_partida, Preco)

**Primary key** Id\_viagem

**Derived** preco (data\_nascimento) --- **saber se é assim !!!!!!!!!!!!**

**Estação** (Id\_estacao, Cidade, Nome)

**Primary key** Id\_estacao

**Comboio** (Id\_comboio, Lugar)

**Primary key** Id\_comboio

1. **Entidades fracas**

São entidades que não possuem existência própria (a sua existência depende da existência de outra entidade) ou que para serem identificadas dependem da identificação de outra entidade. No nosso modelo conceptual não existem entidades fracas.

1. **Relacionamentos 1:\***

No modelo concebido, todas as relações são do tipo 1:\* . De acordo com a metodologia, para a representação destes relacionamentos copia-se a chave primária da “entidade-pai” (correspondente a 1) para a relação que representa a “entidade-filho” (correspondente a \*), definindo-a como chave estrangeira. Assim obtemos:

**“Entidade-pai”**

**Cliente** (CC, Nome, Data\_de\_Nascimento, Email,Telefone)

**Primary key** CC

**“Entidade-filho”**

**Reserva** (Id\_reserva, Preço, Lugar, Data, CC)

**Primary key** Id\_reserva

**Foreign key** cc **references** Cliente(cc)

**“Entidade-pai”**

**Viagem (**Id\_viagem, Hora\_chegada, Hora\_partida, Preco)

**Primary key** Id\_viagem

**Derived** Preco (data\_nascimento) ???????

**“Entidade-filho”**

**Reserva** (Id\_reserva, Preço, Lugar, Data, Id\_viagem)

**Primary key** Id\_reserva

**Foreign key** Id\_viagem **references** Viagem(Id\_viagem)

**“Entidade-pai”**

**Comboio (**Id\_comboio, Nr\_lugar)

**Primary key** Id\_viagem

**“Entidade-filho”**

**Viagem** (Id\_viagem, Hora\_chegada, Hora\_partida, Preco,

Id\_comboio)

**Primary key** Id\_viagem

**Foreign key** Id\_comboio **references** Comboio(Id\_comboio)

**“Entidade-pai”**

**Estação (**Id\_estacao, Cidade, Nome)

**Primary key** Id\_estacao

**“Entidade-filho”**

**Viagem** (Id\_viagem, Hora\_chegada, Hora\_partida, Preco,

Id\_estacao)

**Primary key** Id\_viagem

**Foreign key** Id\_estacao **references** Estação(Id\_estacao)

1. **Relacionamentos 1:1**

No modelo concebido não existem relacionamentos 1:1.

1. **Relacionamentos recursivos 1:1**

No modelo concebido não existem relacionamentos recursivos 1:1.

1. **Relacionamentos superclasse/subclasse**

No modelo concebido não existem relacionamentos superclasse/subclasse.

1. **Relacionamentos \*:\***

No modelo concebido não existem relacionamentos \*:\*.

1. **Relacionamentos complexos**

No modelo concebido não existem relacionamentos complexos.

1. **Atributos multi-valorados**

Para cada atributo multi-valorado pertencente a uma entidade deve-se criar uma relação para representar esse atributo e incluir a chave primária da entidade na nova relação, definindo-a como chave estrangeira. Assim obtemos:

**Comboio (**Id\_comboio, Lugar)

**Primary key** Id\_comboio

**Lugares** (Lugar, Id\_comboio)

**Primary key** Lugar

**Foreign key** Id\_comboio **references** Comboio(Id\_comboio)

Obtemos assim o seguinte quadro:

|  |
| --- |
| **Cliente** (CC, Nome, Data\_de\_Nascimento, email, telefone)  **Primary key** CC |
| **Estação (**Id\_estaçao, Cidade, Nome)  **Primary key** Id\_estacao |
| **Comboio (**Id\_comboio, Nr\_lugar)  **Primary key** Id\_comboio |
| **Reserva** (Id\_reserva, Lugar, Data, CC, Id\_viagem)  **Primary key** Id\_reserva  **Foreign key** CC **references** Cliente(CC)  **Foreign key** Id\_viagem **references** Viagem(Id\_viagem)  **Derived** preco (…..A verificar…) ???? |
| **Viagem** (Id\_viagem, Hora\_chegada, Hora\_partida, Preco,  Id\_estacao)  **Primary key** Id\_viagem  **Foreign key** Id\_estação **references** Estação(Id\_estação)  **Foreign key** Id\_comboio **references** Comboio(Id\_comboio) |
| **Viagem** (Lugar, Id\_comboio)  **Primary key** Lugar  **Foreign key** Id\_comboio **references** Comboio(Id\_comboio) |

**NORMALIZAÇÃO**

**…**

Definição de restrições de integridade

**Informação necessária**

No nosso dicionário, foi referido que apenas o telefone pode tomar o valor null. Assim, todos os atributos importantes têm sempre algum valor, por exemplo uma viagem tem sempre horas de chegada e partida.

**Restrições do domínio dos atributos**

Na identificação dos domínios dos atributos (Ponto 3 da construção do modelo conceptual), foi definido um domínio para cada um dos atributos.

**Multiplicidade**

A multiplicidade representa as restrições referentes aos relacionamentos, como por exemplo uma Viagem ter N Reservas. No ponto 2 da construção do modelo conceptual foram definidos os relacionamentos entre as entidades, e tudo o que é necessário para garantir integridade foi definido neste ponto.

**Integridade de Entidades**

As chaves primárias das entidades não podem tomar o valor null. No ponto 4 da construção do modelo conceptual, foi definido que isto não pode acontecer nunca.

**Integridade de Referências**

Para garantir integridade de referências, é preciso que todas as chaves estrangeiras not null (nas entidades “filhas”) tenham um valor que exista como chave primária numa entidade “pai”. Temos o exemplo de N Reservas relativas a 1 Viagem. A chave estrangeira existente num *tuple* de Reserva tem que ter um valor que corresponda à chave primária (id\_Viagem) de algum *tuple*’ existente de Viagem.

Em todos os relacionamentos referidos na construção do modelo conceptual, as entidades “filhas” têm que ter obrigatoriamente a chave estrangeira not null. Por exemplo, não faria sentido existir uma Reserva que não se referisse a nenhuma Viagem. Noutro exemplo podemos considerar o facto de uma Viagem ter que ser efetuada obrigatoriamente por um Comboio.

Para assegurar a integridade das referências temos que definir restrições para inserções, atualizações e remoções:

Visto que só existem relacionamentos 1:N, todas as situações seguintes são referentes a esse tipo de relacionamentos.

* **Inserções**
  + Inserir um *tuple* de alguma entidade que seja entidade filha de alguma outra, exemplo:

Inserir um *tuple* de Reserva. O valor em Id\_Viagem tem que corresponder a um *tuple* de Viagem existente, ou seja, tem que ser chave primária do mesmo. Visto que não é permitido Id\_Viagem tomar valor null, esta é a única situação possível.

* + Inserir um *tuple* de alguma entidade que seja apenas entidade pai (não há nenhum caso em que seja entidade filha), exemplo:

Inserir um *tuple* de Cliente. Esta inserção não causa qualquer problema na integridade referencial, pois apenas se torna um Cliente que ainda não efetuou reservas. Este tipo de inserções nunca causa problemas, porque como é um novo “pai”, não existem referencias ao mesmo.

* **Remoções**
  + Apagar *tuple* de alguma entidade que apenas seja “filha”, exemplo:

Remover um *tuple* de Reserva. Como Reserva não é entidade “pai” em nenhum relacionamento, esta situação não causa qualquer problema na integridade referencial. O que acontece é que algum *tuple* de Viagem fica com menos uma reserva e algum Cliente também perde uma Reserva. Depois desta remoção, nenhuma referencia se torna inválida.

* + Apagar *tuple* de alguma entidade que seja “pai” em algum relacionamento, exemplo:

Remover um *tuple* de Cliente. Como a entidade Cliente é a entidade “pai” no relacionamento **(Cliente)1:N(Reserva)**, a integridade referencial pode ser perdida. Isto porque, caso exista algum *tuple* da entidade “filha” deste relacionamento, num *tuple* da entidade “filha” há uma chave estrangeira referente a um *tuple* apagado. Neste caso é prevenida a remoção do elemento (NO ACTION), não é permitido remover um *tuple* que seja referenciado em algum outro *tuple*.

* **Atualizações**
  + Atualizar a chave estrangeira de um *tuple* de uma entidade que é “filha” em algum dos relacionamentos. Acontece a mesma situação que no primeiro caso referido em **Inserções**.
  + Atualizar chave primária de um *tuple* de uma entidade que é “pai” em algum dos relacionamentos. Exemplo:

Atualizar a chave primária de Cliente. Atualizar a chave primária de algum *tuple* de Cliente, pode implicar a perda da integridade referencial. Isto porque podem existir *tuples* de Reserva com chave estrangeira referente a este *tuple*. Para evitar esta perda vamos utilizar a estratégia CASCADE, ou seja, a atualização num *tuple* “pai” vai atualizar todas as referências a este.

Para garantir que acontece o que foi referido a cima em todos os relacionamentos, temos que inserir, em todas as referências, o seguinte: ON DELETE NO ACTION ON UPDATE CASCADE. Visto que o descrito a cima é válido para todos os relacionamentos.

**Restrições gerais**

Será necessário garantir que as chaves estrangeiras que se referem a estação destino e origem, em Viagem, nunca se refiram à mesma Estação. Isto porque não faria sentido existir uma viagem com origem e destino iguais.

1. Conclusões e Trabalho Futuro

<<Elaborar uma apreciação crítica sobre o trabalho realizado, apontando os seus pontos fortes e fracos. Adicionalmente, caso se aplique, enunciar eventuais tarefas a realizar futuramente ou novas opções para estender o trabalho realizado.>>

Referências

<<Apresentar a lista de referências bibliográficas referidas ao longo do relatório; recomenda-se a utilização do formato Harvard - <http://libweb.anglia.ac.uk/referencing/harvard.htm>>>

Connolly, T. and Begg, C., 1995. *Database Systems - A Practical Approach to Design, Implementation, and Management.*

Lista de Siglas e Acrónimos

<<Apresentar uma lista com todas as siglas e acrónimos utilizados durante a realização do trabalho. O formato base para esta lista deverá ser da forma como abaixo se apresenta.>>

**BD** Base de Dados

DW Data Warehouse

OLTP *On-Line Analytical Processing*

*... ...*

# Anexos

<<Os anexos deverão ser utilizados para a inclusão de informação adicional necessária para uma melhor compreensão do relatório o para complementar tópicos, secções ou assuntos abordados. Os anexos criados deverão ser numerados e possuir uma designação. Estes dados permitirão complementar o Índice geral do relatório relativamente à enumeração e apresentação dos diversos anexos.>>

1. Anexo 1