



INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

Área Departamental de Engenharia de Electrónica e Telecomunicações e de Computadores

Licenciatura de Engenharia Informática e de Computadores

Sistemas de Informação 1

Semestre de Verão 2020-2021

Reservas online

Trabalho prático
(Fase 1 – 3)

Matilde Pato

Planeamento

As datas importantes a recordar são:

- Lançamento do enunciado: **22 de Março 2021**
- Entrega intermédia (Fase 1): **16 de Abril de 2021** (3 semanas)
- Entrega intermédia (Fase 2): **28 de Maio de 2021** (4 semanas)
- Entrega intermédia (Fase 3): **18 de Junho de 2021** (4 semanas)

Cada entrega intermédia deve apresentar o relatório e código (se houver) referentes **exclusivamente** a essa fase. O relatório deve seguir um dos *templates* fornecidos, obrigatoriamente, sob pena de penalização. Este deve ser conciso e apresentar a justificação de todas as decisões tomadas (ver Critérios de Avaliação). A capa do relatório deve indicar a composição do grupo, a unidade curricular e a fase do trabalho que relata. Caso tenha adendas e/ou correcções a fazer a modelos já entregues, deve indicá-las de forma explícita no relatório seguinte.

O pdf (e, o zip) gerado deve seguir o nome da seguinte forma: 'TPSI1-2021SI-GrupoNNFaseN.ext' (N representa um dígito, e 'ext' a extensão do ficheiro), exemplo: TPSI1-2021SI-Grupo01Fase1.pdf ou TPSI1-2021SI-Grupo14Fase1.pdf.

01 de Junho de 2021, Matilde Pato

Objectivos de aprendizagem

No final da **primeira fase do trabalho**, os alunos devem ser capazes de:

- Identificar correctamente as relações relevantes para os requisitos pretendidos;
- Identificar correctamente os atributos chave (primária) para cada uma das relações;
- Identificar correctamente os atributos descritivos de cada uma das relações;
- Determinar correctamente as chaves candidatas;
- Determinar correctamente as chaves estrangeiras;
- Identificar as restrições de integridade de domínio;
- Identificar as restrições de integridade de coluna;
- Identificar as restrições de integridade de utilizador (regras de negócio);
- Desenvolver modelos relacionais normalizados até à 3NF baseado nas chaves primárias;
- Garantir que não existem perdas de dependências funcionais.

Prêambulo

Viajar, em lazer ou em trabalho, faz parte da rotina da generalidade da população. Segundo a Pordata, no ano de 2019, 24.462,8 de portugueses realizaram viagens turísticas dentro e fora do país, por automóvel, comboio, avião ou outro meio de transporte. A revista “Volta ao Mundo” em 27 de julho de 2020, publicou uma reportagem cujo título era “Estudos revelam que viajar faz bem à saúde física e mental”.

Enunciado do trabalho (Documento de requisitos do sistema)

O Travel4Me é um sistema de gestão para reservas de viagens realizadas de comboio e/ou de autocarro. Os recursos incluem uma fácil gestão de reservas e de integração com as várias empresas/operadores de transporte. Os passageiros poderão realizar uma ou várias reservas no sistema, de viagens desde que existam lugares sentados disponíveis. O passageiro é identificado pelo número de identificação (cartão de cidadão ou passaporte), e caracterizado pelo nome, o e-mail para onde o bilhete é endereçado, a data-de-nascimento, e o género ('M', 'F', 'N/R'). Todos os campos, com excepção do género, são obrigatórios e servem para emissão do bilhete.

Para a reserva é armazenado um identificador (inteiro positivo) gerado pelo sistema. Associados à reserva temos o número de passageiros, o local de origem e de destino, o preço do total da viagem em euros, o meio de transporte (comboio ou autocarro), a data de reserva em "DD-MM-AAAA", e a hora da reserva (cujo formato é hh:mm). O preço da viagem depende não apenas do local de origem e de destino, e também da idade do passageiro e de equipamento que este queira transportar. Os valores possíveis são: (1) adulto (30 a 64 anos), (2) jovem (13 a 29 anos), (3) sénior (+65 anos), (4) criança (4 a 12 anos), (5) militar, (6) animal (com caixa), (7) prancha desporto, e (8) bicicleta. O local de origem não pode ser igual ao de destino, como se compreende. A reserva "fará" uma pesquisa pelas múltiplas viagens que, para aquela data e local, possam existir ou não (ver Figura 1, como exemplo). Assim, como a disponibilidade em termos de lugares sentados. A reserva fica concluída sendo "lançado" um *ticket id*, quando o passageiro selecciona a opção de acordo com a sua escolha e realiza o pagamento.

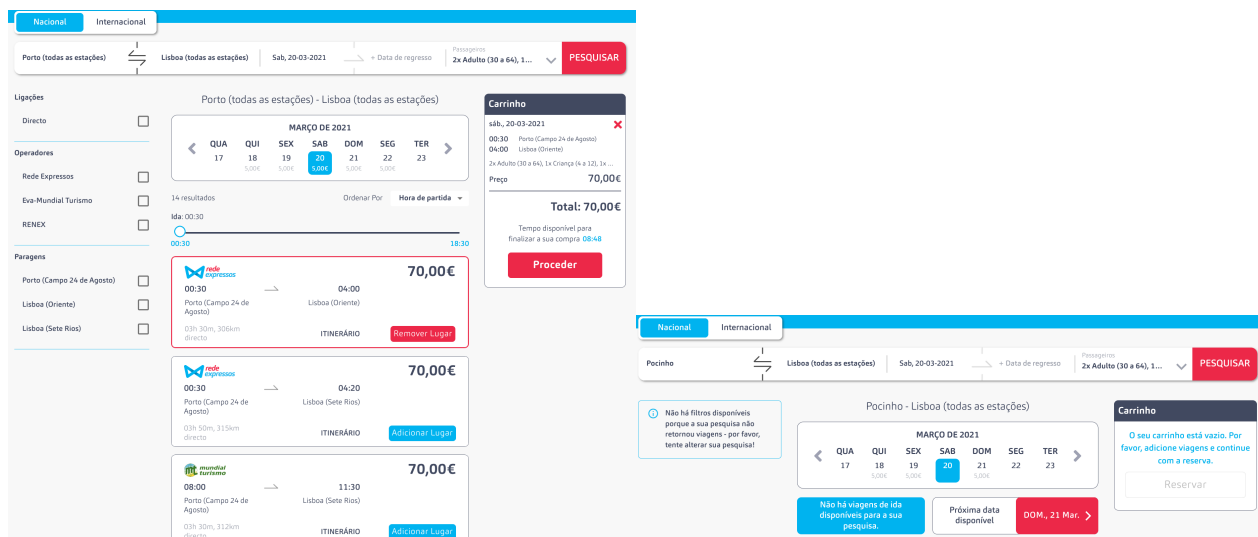


Figura 1: Imagem ilustrativa de duas tentativas de reserva na página da rede expressos

O pagamento pode ser feito por MB, MB Way, Pay Pal ou Cartão de crédito. Se, o pagamento for do tipo MB Way, o passageiro necessita de introduzir o seu número de telemóvel.

A viagem é identificada por um valor único, e caracterizada pelo a data (em “DD-MM-AAA”), a hora de partida e de chegada (cujo formato é hh:mm), e a distância em km. As viagens decorrem entre estações, sabendo que a estação de origem tem necessariamente de ser diferente da estação de chegada. As estações são identificadas pelo seu nome, tem um tipo (terminal ou paragem), e o número de plataformas. O seu nome, pode coincidir com o nome da localidade quando não existe mais do que uma, i.e. uma localidade pode ter mais do que uma estação. A localidade é identificada pelo código-postal (único), e caracterizada pelo nome da mesma. A viagem pode ser efectuada quer por autocarro, quer por comboio, mas nunca por ambos em simultâneo.

Um autocarro é caracterizado pela matrícula, a velocidade máxima (em km/h), a data de entrada ao serviço, o número de quilómetros, a data da próxima revisão e, um tipo. Cada tipo de autocarro tem uma marca, um modelo, e um número total de lugares sentados. O comboio é a composição de uma locomotiva e várias carruagens. O comboio tem um tipo (alfa-pendular - AP, inter-cidades - IC, inter-regionais - IR ou regionais - R), e o número de carruagens.

As locomotivas são identificadas pelo número de série, e caracterizadas pela marca, velocidade máxima (em km/h), data de entrada ao serviço, o número de quilómetros. O comboio do tipo AP não tem locomotiva. Contudo, pode ser caracterizado como a locomotiva com a diferença de que inclui o 2 classes distintas: “conforto” e “turística”, um número total de lugares sentados (299), e um número. As carruagens têm 2 classes distintas: primeira ou segunda, um número total de lugares sentados, e um número. No caso do AP, a numeração faz-se no sentido Norte → Sul, a carruagem 1 será a mais a Norte e a 6 (última carruagem). Nos restantes, as carruagens de 1ª classe começam por um “1” ou “8”, neste caso carruagem de 1ª classe com bar, (ex: 11, 12, ou 81) e estão ordenadas no sentido Norte → Sul. As carruagens de 2ª classe começam por “2” (ex: 21, 22, ...) e estão ordenadas no sentido Sul → Norte. Os lugares disponíveis são identificados por um número (corresponde ao número de lugar do autocarro/comboio) e têm um tipo. O tipo de lugar é identificado por um identificador único, e caracterizado por uma classe (1ª, 2ª classe ou superior) e o preço em euros. O preço varia com o passageiro: (1) adulto (30 a 64 anos), (2) jovem (13 a 29 anos), (3) sénior (+65 anos), (4) criança (4 a 12 anos), e (5) militar. Existem vários lugares do mesmo tipo.

Resultados pretendidos

Tendo em conta os objectivos de aprendizagem, deverão ser produzidos os seguintes resultados:

1. O modelo de dados lógico, indicando de forma clara, todas as chaves primárias, candidatas e estrangeiras, o domínio dos atributos. Deve incluir todas as restrições que não são garantidas pelo modelo lógico, nomeadamente as obrigatoriedades.
2. Para cada esquema de relação, devem também ser apresentadas as dependências funcionais associadas aos atributos do modelo lógico.
3. Para o modelo lógico obtido em 1. indique, justificando em que forma normal está cada relação. Para as que não se encontram na terceira forma normal decompõe-as baseando-se nas chaves primárias.

Data limite para entrega: 16 de Abril de 2021 até às 23:59.

A entrega deve incluir um relatório, enviados de forma electrónica através do Moodle. Os documentos **são entregues** em formato PDF (obrigatório).

Nota: Deve ser possível aferir cada um dos objectivos de aprendizagem no material que entregar.

Objectivos de aprendizagem

No final da **segunda fase do trabalho**, os alunos devem ser capazes de:

- Utilizar correctamente a álgebra relacional, com os seus vários operadores, para expressar interrogações sobre um modelo relacional:
 - operadores relacionais unários: `select`, `project` e `rename`;
 - operadores relacionais binários: `division`;
 - operadores sobre conjuntos: `difference`, `intersect` e `union`;
 - operadores de junção: interna e externa;
 - operador de agregação;
- Utilizar correctamente SQL/DDL para criar as tabelas num sistema de gestão de bases de dados (SGDB);
- Garantir as restrições de integridade identificadas enumerando aquelas que têm de ser garantidas pelas aplicações;
- Inserir dados em lote através da cláusula SQL `INSERT`, garantindo que as restrições de integridade são cumpridas;
- Garantir a atomicidade de instruções, utilizando processamento transaccional;
- Utilizar correctamente os operadores da teoria dos conjuntos em T-SQL;
- Utilizar correctamente as cláusulas `INNER JOIN` e `OUTER JOIN`;
- Utilizar correctamente sub-interrogações correlacionadas;
- Utilizar correctamente funções de agregação;
- Utilizar correctamente a cláusula `HAVING`;

- Utilizar correctamente a cláusula `ORDER BY`;
- Utilizar correctamente o termo `DISTINCT`;
- Utilizar correctamente os predicados `IN` e `EXISTS`.

Após a realização da 1ª fase do trabalho segue-se a implementação do modelo físico do sistema, i.e. deverá ser construído em T-SQL contemplando todas as restrições que consigam garantir na forma declarativa.

Nota: Deverão preencher a base de dados com informação necessária que permita em seguida realizar interrogações que apresentem resultados pertinentes. Na etapa de preenchimento da base de dados, os alunos deverão ter particular atenção ao cumprimento das **restrições de integridade, utilizando de forma adequada o controlo transaccional (a atomicidade)**.

Resultados pretendidos

Tendo em conta os objectivos de aprendizagem, deverão ser produzidos os seguintes resultados:

1. Construção do modelo físico do sistema, contemplando todas as restrições de integridade passíveis de ser garantidas declarativamente, assim como a **atomicidade** nas operações.

O código T-SQL que permite:

- (a) Criar o modelo físico (1 *script* autónomo): `createTable.sql`;
- (b) Remover o modelo físico (1 *script* autónomo): `removeTable.sql`;
- (c) Preenchimento inicial da base de dados (1 *script* autónomo): `insertTable.sql`;
- (d) Apagar todos os dados existentes nas tabelas (1 *script* autónomo): `deleteTable.sql`.

Os dados introduzidos devem permitir validar todas as interrogações pedidas nesta fase do trabalho.

2. Considerando o esquema relacional obtido anteriormente e fornecido no final deste documento ("Adenda"), apresente as expressões em álgebra relacional (AR) que respondam:

- (a) Pretende-se o registo de todas as viagens de Lisboa para o Porto, com a indicação do dia e hora de partida.

- (b) Liste todos os passageiros (nome e idade) e o número de lugar deste.
 - (c) Apresente o modelo, a marca e número de lugares sentados dos meios de transporte em uso. Faça uso de um operador da teoria dos conjuntos.
 - (d) Obtenha o número de lugares vazios dos autocarros que partem de Lisboa. Faça uso de uma sub-interrogação não correlacionada.
 - (e) Indique o número total de passageiros por género, faça uso do operador de união (apenas).
 - (f) Indique a soma dos preços de bilhete para a categoria de adultos.
 - (g) Apresente a média de idades dos passageiros por reserva. O conjunto apresentado deve conter a média de idades, o número de reserva e a data desta. Deve recorrer à cláusula `GROUP BY`.
3. Conceba, na linguagem T-SQL, as interrogações que produzam os resultados a seguir indicados, utilizando apenas uma instrução T-SQL. Guarde num *script* autónomo de nome "queries.sql". **Para cada instrução deve ser também apresentada a descrição do raciocínio seguido.**
- (a) Implemente em T-SQL as interrogações pedidas na alínea 2.
 - (b) Apresente o nome de todos os passageiros que viajaram para Lisboa, por ordem alfabética (em primeiro lugar) e depois por idades, sendo que deve ir do menor para maior.
 - (c) Mostre todas as viagens de Lisboa para o Porto, que se realizaram no período da manhã (6h00 - 12h00).
 - (d) Mostre o tipo de pagamento pela média de idades dos passageiros que efectuaram a reserva.
 - (e) Apresente os passageiros que realizaram mais reservas, por ordem decrescente?
 - (f) Crie uma vista que permita obter informação sobre todas as reservas, que se realizaram no ano passado (deve fazer uso de uma função de `datetime` e não fixar o valor 2020).
 - (g) Crie uma vista que permita obter informação sobre todas as viagens que estejam a decorrer, com indicação da estação de partida e de regresso, a hora de partida e o tempo previsto (ou, a duração).
4. Apresente o(s) comando(s) que permitem inserir na BD a seguinte informação: Viagem de alfa-pendular, com partida de Lisboa (Parque das Nações) e destino Beja, todos os

dias da semana, pelas 12h00. Esta viagem, vem substituir aquela que se realizava pelas 10h30 num inter-regional.

Todas as simplificações e optimizações realizadas ao modelo devem ser indicadas e justificadas.

PS.1 Sugere-se que consulte o manual do SGDB para obter informação sobre as funções de manipulação de datas.

Data limite para entrega: 28 de Maio de 2021 até às 23:59.

A entrega deve incluir um documento com as respostas de AR e o código T-SQL, enviados de forma electrónica através do Moodle. O documento **é entregue** em formato PDF.

Nota: Deve ser possível aferir cada um dos objectivos de aprendizagem no material que entregar.

Objectivos de aprendizagem

No final da **terceira fase do trabalho**, os alunos devem ser capazes de:

- Estabelecer uma ligação ao SGBD pretendido, correctamente parametrizada, utilizando JDBC;
- Utilizar correctamente comandos parametrizados para executar operações em JDBC (*prepared statement*);
- Utilizar correctamente transações para garantir atomicidade nas operações, utilizando JDBC;
- Gerir correctamente o tempo de vida das ligações JDBC;
- Garantir a libertação de recursos, quando estes não estejam a ser utilizados;
- Utilizar correctamente o tipo `ResultSet`;
- Implementar **todas as restrições de integridade** aplicacionais que não foram possíveis em T-SQL.

Enunciado do trabalho (3ª fase)

Considerando o texto com os requisitos do sistema, apresentados na primeira fase do trabalho, e o modelo de dados implementado na segunda fase, deve garantir, no programa desenvolvido em Java que:

1. não são vendidos mais bilhetes que os permitidos no meio de transporte escolhido;
2. o tempo de chegada é igual ao estipulado (definido no campo “horachegada”) a partir da distância e da velocidade máxima do meio de transporte;
3. todas as reservas realizadas e cujo modo de pagamento foi por “MBWay” estão registadas na tabela PAGMBWAY;

4. o atributo “ncarruagens” de COMBOIO é 6 quando o seu tipo é “AP”.

Os alunos terão agora a oportunidade de utilizar uma API JDBC para, através de uma aplicação Java, acederem e manipularem os dados existentes no modelo físico criado na fase anterior. As alterações/actualizações/consultas irão ser feitas a nível da aplicação e não directamente na BD.

Sempre que se justificar devem ser usados os mecanismos transacionais necessários para garantir a atomicidade das operações. Também devem utilizar mecanismos que evitem ataques de “*SQL injection*”, bem como que evitem problemas de formatações, e.g. campos de datas.

Resultados pretendidos

Tendo em conta os objectivos de aprendizagem, deverão ser produzidos os seguintes resultados:

1. Uma aplicação Java (em modo linha de comandos) que permita realizar as seguintes operações **(não deve fazer qualquer alteração à BD já construída)**:
 - (a) Opção para efectuar uma nova reserva;
 - (b) Opção para substituir uma nova viagem, como o exemplo da alínea 4 (da Fase 2);
 - (c) Opção para colocar um autocarro fora de serviço, i.e. este deixa de efectuar viagens. Para tal, em Java deve criar uma nova tabela onde ficam registados todos os autocarros que não estão em circulação;
 - (d) Opção para calcular o número total de quilómetros de um autocarro. O utilizador deve apenas indicar a matrícula deste;
 - (e) Opções que implementem as alíneas 2.d, 2.f, 2.g, 3.c e 3.d (da Fase 2), garantindo que todos os parâmetros variáveis são alteráveis na interface com o utilizador e.g. em 2.d “autocarros que partem de Lisboa” pode ser substituído por outra localidade; em 2.f a categoria (“adulto”) pode ser variável.

Todas as instruções devem vir indicadas (e, explicadas) no relatório que dá suporte a este trabalho. Se considerar necessário, pode a título ilustrativo, mostrar tabela(s) de resultados.

Data limite para entrega: 18 de Junho de 2021 até às 23:59.

A entrega deve incluir um relatório e o código, enviados de forma electrónica através do Moodle. O relatório **é entregue** em formato PDF.

Notas:

1. Deve garantir a correcta implementação de todas as funcionalidades, incluindo o acesso a dados;
2. Deve criar em Java um modelo de dados que mapeie as relações utilizadas para objectos em memória;
3. A lógica de interface com o utilizador deve estar em classes separadas da lógica de acesso a dados;
4. Deve ser possível aferir cada um dos objectivos de aprendizagem no material que entregar.

Todo o código entregue tem de ser executado independente do ambiente de desenvolvimento, em linha de comandos. Os alunos têm de fornecer as instruções de execução, assumindo como único pré-requisito a existência da máquina virtual Java 1.8.