



Departamento de Engenharia de Electrónica e Telecomunicações e
de Computadores
Licenciatura em Engenharia Informática e de Computadores

Trabalho prático

(Fases 1,2 e 3)

Sistemas de Informação I

Semestre de Inverno 2015/2016

Versão 1.00

Docentes: Afonso Remédios, Lara Santos e Nuno Datia

Planeamento

As datas importantes a recordar são:

- Lançamento do enunciado: **21 de Setembro de 2015**
- Entrega intermédia (Fase 1): **02 de Novembro de 2015**
- Entrega intermédia (Fase 2): **21 de Dezembro de 2015**
- Entrega intermédia (Fase 3): **18 de Janeiro de 2016**

Cada entrega intermédia deve apresentar o relatório e código (se houver) referentes exclusivamente a essa fase. O relatório deve ser conciso e apresentar a justificação de todas as decisões tomadas. 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 entregados, deve indicá-las de forma explícita no relatório seguinte.

Objectivos de aprendizagem

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

- ☐ Identificar correctamente as entidades relevantes para os requisitos pretendidos;
- ☐ Identificar correctamente os atributos chave para cada uma das entidades;
- ☐ Identificar correctamente os atributos descritivos de cada uma das entidades;
- ☐ Identificar o domínio de cada atributo;
- ☐ Identificar correctamente as associações entre entidades, incluindo respectivas obrigadoriedades e cardinalidades;
- ☐ Desenvolver um modelo Entidade-Associação (EA) que cumpra os requisitos enunciados, capturando o maior número de restrições possível;
- ☐ Identificar os requisitos e restrições que não conseguem ser garantidos no modelo EA.

Enunciado do trabalho (Documento de requisitos do sistema)

A empresa **AcamPE** pretende criar um sistema de informação que permita gerir os seus parques de campismo.

Um parque de campismo é caracterizado por um nome (único), uma morada e uma localização em coordenadas WGS84. As coordenadas devem ser guardadas numa única cadeia de caracteres, especificando a latitude e longitude em formato decimal, e.g. "38.713889, -9.139444". Cada parque de campismo tem uma série de alojamentos, que podem ser espaços para tendas, espaços para rouletes ou *bungalows*. Um alojamento é caracterizado por identificador único no parque, uma descrição, um preço de diária e uma localização dentro do parque. O esquema de localização pode variar de parque para parque, devendo ser possível a escrita de caracteres alfanuméricos. Dentro de um parque não podem existir dois alojamentos com a mesma localização. Para os *bungalows* deve ser possível guardar a informação da tipologia, na forma 'Tx', onde x pode tomar um valor no intervalo [0..3]. OS espaços para tendas devem ter a indicação da área em M^2 . Cada alojamento pode ter um conjunto de extras disponíveis. Cada extra é caracterizado por um identificador único no sistema, uma descrição e um preço por dia. São exemplos de extras: (i) Electricidade, (ii) Estacionamento de um automóvel, (iii) Pessoa

Extra, (iv) Animal de companhia. Dependendo do parque, alguns tipos de alojamentos podem não ter todos os extras disponíveis.

O sistema deve ter a possibilidade de registar os hóspedes que ficaram alojados no parque de campismo. Para cada hóspede deve ser possível saber o número de identificação fiscal (NIF), um número do cartão de identificação, sendo aceites o nº do cartão de cidadão ou do passaporte, a morada e o nome. Devem ficar registadas as datas de entrada e saída de cada hóspede, designada de estadia, podendo o mesmo hóspede ficar alojado diversas vezes no mesmo alojamento, em períodos diferentes. Uma estadia tem um identificador único e pode ser resultado de uma reserva prévia. nesse caso, tem de ficar registado o valor do sinal que foi pago. Deve ser possível saber o número de extras que cada hóspede requereu para cada alojamento, no âmbito de uma estadia. Cada parque pode ter um conjunto de actividades para os hóspedes. Cada actividade é paga, devendo ficar registado o custo por pessoa no sistema. Além disso, cada actividade é caracterizada por uma data de realização, um nome, uma descrição e a lotação máxima. Não existem duas actividades com o mesmo nome na mesma data dentro de um parque. Deve ser possível obter do sistema as actividades que cada hóspede requereu durante a sua estadia, assim como o número de participantes nelas envolvidos.

Pretende-se manter um histórico de preços para extras, actividades e alojamentos, sendo necessário saber as datas de alteração e fim de validade. De notar que as alterações de preço podem acontecer em qualquer momento, mas apenas uma vez por dia para cada entidade.

Resultados pretendidos

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

1. O modelo de dados conceptual, incluindo o diagrama entidade-associação, a descrição das entidades, os seus atributos e associações;
2. Descrição das regras de negócio aplicáveis, para além das que são implícitas no resultado do ponto 1;
3. Informação complementar fornecida pelo cliente e que seja importante para clarificar aspectos do domínio de aplicação;
4. Discussão de alternativas de modelação e as razões da escolha da solução apresentada no ponto 1;

Data limite para entrega: 02 de Novembro de 2015 até às 23:59.

A entrega deve incluir um relatório e o modelo EA, enviados de forma electrónica.

Nota: Sugere-se que o relatório seja organizado de acordo com os pontos anteriores. 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:

- ☐ Aplicar correctamente as regras de passagem de EA para relacional;
- ☐ Identificar as dependências funcionais existentes no sistema;
- ☐ Determinar as chaves primárias, alternativas e estrangeiras;
- ☐ Garantir que não existem perdas de dependências funcionais;
- ☐ Desenvolver modelos relacionais normalizados até à 3NF;
- ☐ Utilizar correctamente SQL/DDL para criar as tabelas num SGDB;
- ☐ Escolher correctamente os tipos de dados para cada atributo;
- ☐ Garantir as restrições de integridade identificadas, não esquecendo as que resultam da passagem do EA para o relacional, enumerando aquelas que têm de ser garantidas pelas aplicações;
- ☐ Utilizar correctamente a álgebra relacional, com os seus vários operadores, para expressar interrogações sobre um modelo relacional;
- ☐ 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 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.

Enunciado do trabalho (2ª fase)

Após a realização da 1ª fase do trabalho, os alunos têm agora a oportunidade de reflectir sobre o modelo que apresentaram. Após realizarem as correcções e alterações que considerem necessárias e/ou relevantes, devem prosseguir para a passagem do modelo conceptual para o modelo lógico. O modelo lógico a apresentar será o modelo relacional e deverá ser acompanhado de todas as restrições que não são garantidas pelo modelo, incluindo aquelas já apresentadas em conjunto com o modelo conceptual.

Após a obtenção do modelo lógico, os alunos deverão fazer o levantamento do conjunto de dependências funcionais que julguem adequadas para o domínio de aplicação e, com base nelas, verificar a normalização do modelo lógico, fazendo-lhe as alterações necessárias para garantir a sua normalização até à 3ª forma normal (3NF). Os alunos poderão ainda decidir fazer alterações, simplificações ou optimizações ao modelo, desde que devidamente indicadas e fundamentadas. Com base no modelo lógico obtido, deverá ser construído em SQL, o modelo físico do sistema, contemplando todas as restrições que consigam garantir na forma declarativa.

Nota: Deverão preencher a base de dados com informação 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.

Resultados pretendidos

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

1. O modelo de dados conceptual final, incluindo diagrama entidade-associação, descrição das entidades, seus atributos e associações, descrição final de todas as regras de negócio aplicáveis.
2. O modelo de dados lógico, obtido por transformação do modelo conceptual apresentado. Deve ser indicado, de forma clara, todas as chaves estrangeiras, as chaves candidatas e primárias, incluindo todas as restrições que não são garantidas pelo modelo lógico obtido e que derivem da passagem do modelo conceptual para o modelo lógico, nomeadamente as obrigatoriedades. Devem também ser apresentadas as dependências funcionais associadas aos atributos do modelo lógico.
3. A análise da normalização do modelo lógico produzido na alínea 2, tendo em conta o conjunto de dependências funcionais. Deverá ser garantida a normalização até à terceira forma normal. Todas as simplificações e optimizações que forem realizadas ao modelo devem ser igualmente indicadas e justificadas.
4. Construção do modelo físico do sistema, contemplando todas as restrições de integridade passíveis de ser garantidas declarativamente. Produza dois *scripts* SQL: um para a criação

e outro para a destruição do modelo.

5. Criação de um *script* SQL para preencher o modelo físico. Os dados devem permitir validar todas as interrogações pedidas nesta fase do trabalho. Crie também um *script* SQL para apagar todos os dados existentes nas tabelas.
6. Considerando o esquema relacional obtido nesta fase, apresente as expressões em álgebra relacional que produzam:
 - (a) Para o hóspede com o cartão de cidadão com o número '123456789', o seu nome, a data das suas estadias, o tipo e a localização dos alojamento utilizados.
 - (b) O NIF e o nome do(s) hóspede(s) com mais alojamentos no parque de campismo "Prado verde".
 - (c) A lista de hóspedes que tiveram estadias em 2015 e que ainda não tem reservas ou estadias para 2016.
 - (d) A lista dos NIF dos hóspedes que usaram todos os extras de um alojamento onde tiveram hospedados.
 - (e) A listas das actividades de 2015, cujo número de participantes é igual à lotação máxima da actividade.
 - (f) A lista de alojamentos do parque "Lua Azul" que tiveram mais de 10 ocupações em 2015. Mostre a chave do alojamento e a sua localização.
7. Conceba, na linguagem SQL, as interrogações que produzam os resultados a seguir indicados, utilizando apenas uma instrução SQL. Para cada instrução deve ser também apresentada a descrição do raciocínio seguido.
 - (a) Apresente em SQL as interrogações pedidas na alínea 6.
 - (b) Apresente a lista dos parques de campismo (nome e coordenadas WGS84) incluindo, se existirem, as actividades neles desenvolvidas no 3º trimestre de 2015. Das actividades, apresente apenas o nome e a descrição. A lista deve vir ordenada por nome do parque (1º critério) e nome da actividade (2º critério).
 - (c) Apresente os alojamentos do parque "Lua Azul" que tiveram hóspedes todos os dias entre '01-07-2015' e '31-08-2015'. Sugerimos a utilização da função `datediff` para obter o número de dias entre duas datas.
 - (d) Desenvolva uma instrução que permita saber quanto o hóspede '100100100' tem a pagar pelo alojamento iniciado a '21-08-2015'. Tenha em atenção que o preço a pagar é sempre referente aos preços em vigor na data de início do alojamento, mesmo que tenham sido feitas actualizações nos preços.

- (e) Apresente os NIFs dos hóspedes que fizeram reservas em 2015 para mais de 7 noites de estadia e que não têm nenhum alojamento registado no sistema do tipo *bungalow*.
- (f) Crie uma vista que permita ao gestor da empresa ver os valores pagos pelos hóspedes para as estadias que já terminaram. Deve ser possível visualizar, para além do valor pago, quem foi o hóspede, em que parque ficou alojado e a data do alojamento.

Data limite para entrega: 21 de Dezembro de 2015 até às 23:59.

A entrega deve incluir um relatório e o código SQL, enviados de forma electrónica.

Nota: Sugere-se que o relatório seja organizado de acordo com os pontos anteriores. 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;
- ☐ Utilizar correctamente transacções 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 restrições de integridade aplicacionais.

Enunciado do trabalho (3ª fase)

Após a realização da 2ª fase do trabalho, os alunos têm agora a oportunidade de utilizar a API JDBC para, através de uma aplicação Java, acederem e manipularem os dados existentes no modelo físico criado na fase anterior.

Sempre que se justificar devem ser usados os mecanismos transaccionais 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

Uma aplicação Java (em modo linha de comandos ou GUI) que permita realizar as seguintes operações:

1. Opção para criar alojamentos em parques de campismo;
2. Opção para criar extras em alojamentos;
3. Opção para remover um alojamento.

4. Opção para fazer uma reserva de estadia, para hóspedes não registados no sistema;
5. Opção que permita saber quanto o hóspede '100100100' tem a pagar pelo alojamento iniciado a '21-08-2015 (implementação da alínea 7d da 2ª fase do trabalho);
6. Opção para actualizar o preço de um alojamento.

Data limite para entrega: 18 de Janeiro de 2015 até às 23:59.

A entrega deve incluir um relatório e o código, enviados de forma electrónica.

Nota: Não deve ser descuidada o acesso aos dados de forma correta, ou seja, não deve ser feito demasiado investimento na interface com o utilizador, sem antes garantir a correcta implementação de todas as funcionalidades.

Sugere-se que o relatório seja organizado de acordo com os pontos anteriores. Deve ser possível aferir cada um dos objectivos de aprendizagem no material que entregar.