



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

Trabalho prático

(Fases 1 e 2)

Sistemas de Informação II

Semestre de Inverno 2016/2017

Versão 1.00

Docente: Nuno Datia

Planeamento

As datas importantes a recordar são:

- Lançamento do enunciado: **20 de Setembro de 2016**
- Entrega intermédia (Fase 1): **02 de Dezembro de 2016**
- Entrega intermédia (Fase 2): **13 de Janeiro de 2017**

Cada entrega intermédia tem de incluir o relatório e código (se houver) referentes a essa fase. O relatório deve ser conciso e apresentar a justificação de todas as decisões tomadas. Tem de indicar a composição do grupo, a unidade curricular e a fase do trabalho que relata. Caso tenha adendas e/ou correcções a fazer a entregas anteriores, deve enumerar cada uma de forma explícita no relatório. Entregas sem relatório não serão consideradas.

Cada fase do trabalho terá de ser mostrada ao docente a funcionar, em data, hora e local a combinar.

Objectivos de aprendizagem

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

- ☐ Desenvolver um modelo de dados adequado aos requisitos, normalizado até à 3NF;
- ☐ Conceber e implementar uma solução baseada em bases de dados dinâmicas, adequada aos requisitos;
- ☐ Utilizar correctamente controlo transaccional;
- ☐ Utilizar correctamente níveis de isolamento;
- ☐ Utilizar correctamente vistas, justificando o seu uso na solução;
- ☐ Utilizar correctamente procedimentos armazenados, justificando o seu uso na solução;
- ☐ Utilizar correctamente gatilhos, justificando o seu uso na solução;
- ☐ Utilizar correctamente funções, justificando o seu uso na solução;
- ☐ Escrever um relatório técnico sobre o trabalho desenvolvido.

Enunciado do trabalho (documento de requisitos do sistema)

Uma empresa denominada *Ænima*, que se dedica a aluguer de equipamentos de praia, pretende desenvolver um sistema informático para melhorar a gestão dos alugueres dos seus equipamentos. Nesse sistema, os equipamentos são caracterizados por um código (único), por uma descrição textual e por um tipo. Cada tipo tem um nome (único) e uma descrição. São exemplos de tipos de equipamento: canoas, gaivotas, motas de água e toldos.

Todos os equipamentos têm, pelo menos, um preço de aluguer. É necessário saber qual o período de validade do preço e o seu valor. Assume-se que podem existir vários preços associados a um equipamento, representando o valor a pagar para diferentes fracções temporais, e.g., hora, meia hora, dia, etc. Um aluguer de equipamento é caracterizado por uma data/hora de aluguer, um número de série, uma data de fim e um tipo a indicar a duração temporal do aluguer. Para cada aluguer, deve ficar registado o preço pago por cada equipamento, podendo esse valor ser diferente do preço de tabela por diversos factores, por exemplo, a existência de campanhas promocionais. Um aluguer é efectuado por um empregado, devendo ficar registado esse facto. Para o sistema, um empregado é caracterizado por um número (único) e por um nome. Também

tem ficar registado no sistema o cliente que efectuou o aluguer. Um cliente é caracterizado por um código único e, opcionalmente, um nome, um número fiscal e uma morada. Para facilitar os alugueres em cenários onde o cliente não quer dar os dados para emissão de factura, existe no sistema um registo de um cliente final com toda a informação pessoal nula. No sistema podem estar registadas promoções para os alugueres. As promoções podem ser de dois tipos: de tempo extra ou descontos. Na primeira, é atribuído um tempo extra para cada fracção de tempo identificado na promoção. Para a segunda, é efectuado um desconto no preço final, de acordo com a percentagem definida na promoção. Todas as promoções têm um identificador único, uma data de início, uma data de fim da promoção e tem uma descrição textual.



Garanta que o modelo de dados exposto é consistente com as entidades descritas, escondendo sempre detalhes de implementação do utilizador final. Note que, para efeitos de auditoria, não são permitidas alterações e remoções a alugueres já efectuados. Considera-se efectuado sempre que as datas de aluguer sejam mais antigas que a data actual.



Resultados pretendidos

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

1. O modelo de dados (conceptual e relacional), incluindo todas as restrições de integridade;
2. O código T-SQL que permite:
 - (a) Criar o modelo físico;
 - (b) Remover o modelo físico;
 - (c) Inserir, remover e actualizar informação de um cliente;
 - (d) Inserir, remover e actualizar informação de um equipamento;
 - (e) Inserir, remover e actualizar informação de promoções;
 - (f) Inserir um aluguer com inserção de um cliente com dados completos;
 - (g) Inserir um aluguer usando um cliente existente;
 - (h) Remover um aluguer;
 - (i) Efectuar alterações de preço;
 - (j) Listar todos os equipamentos livres, para um determinado tempo e tipo;
 - (k) Listar os equipamentos sem alugueres na última semana;
 - (l) Testar cada um dos requisitos anteriores.

Note que para os utilizadores, todas as instruções produzem os resultados esperados, sendo sempre garantidas as regras de negócio.

Data limite para entrega: 02 de Dezembro de 2016 até às 23:59.

A entrega tem de incluir um relatório (entregue em PDF) e o código T-SQL, enviados de forma electrónica.

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:

- ☐ Desenvolver uma aplicação em C#, que use diferentes implementações de acesso a dados;
- ☐ Utilizar correctamente processamento transaccional, através de mecanismos disponíveis na plataforma .NET, para garantir as propriedades ACID nas operações;
- ☐ Utilizar correctamente (ADO.NET) Entity Framework para acessos a dados;
- ☐ Utilizar correctamente ADO.NET em modo “conectado”;
- ☐ Garantir a correcta libertação de ligações e recursos, quando estes não estejam a ser utilizados;
- ☐ Garantir a correcta implementação das restrições de integridade e/ou lógica de negócio;
- ☐ Utilizar XML e XML Schema para produzir documentos válidos e bem formados a partir de dados guardados num SGBD relacional.

Enunciado do trabalho (2ª fase)

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

1. Crie uma única aplicação .NET que permita:
 - (a) Aceder às funcionalidades **(e)** a **(k)**, descritas na alínea 2 da fase 1 deste trabalho;
 - (b) Exportar a informação dos alugueres (individuais e em conjunto) entre duas datas, de acordo com o XML Schema em anexo;
 - (c) Usar uma implementação de acesso a dados desenvolvida usando Entity Framework (EF)
 - (d) Usar uma implementação de acesso a dados desenvolvida usando objectos “conectados” do ADO.NET.
2. Compare, usando métricas objectivas, a tecnologia EF e ADO.NET quanto à facilidade de programação e desempenho.

3. Indicar em que situações a EF e o ADO.NET apresentam vantagens/desvantagens em termos de garantia da consistência dos dados.

Data limite para entrega: 13 de Janeiro de 2016 até às 23:59.

A entrega deve incluir um relatório (em PDF), os projectos *Visual Studio*, o código C# e o código SQL, enviados de forma electrónica.

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

XML Schema a usar

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
3   xmlns:xs="http://www.w3.org/2001/XMLSchema">
4   <xs:element name="xml" type="xmlType"/>
5   <xs:complexType name="xmlType">
6     <xs:sequence>
7       <xs:element type="alugueresType" name="alugueres"/>
8     </xs:sequence>
9   </xs:complexType>
10  <xs:complexType name="aluguerType">
11    <xs:sequence>
12      <xs:element type="xs:positiveInteger" name="cliente"/>
13      <xs:element type="xs:positiveInteger" name="equipamento"/>
14    </xs:sequence>
15    <xs:attribute type="xs:positiveInteger" name="id"/>
16    <xs:attribute type="xs:string" name="tipo"/>
17  </xs:complexType>
18  <xs:complexType name="alugueresType">
19    <xs:sequence>
20      <xs:element type="aluguerType" name="aluguer"/>
21    </xs:sequence>
22    <xs:attribute type="xs:string" name="dataInicio"/>
23    <xs:attribute type="xs:string" name="dataFim"/>
24  </xs:complexType>
25 </xs:schema>
```