



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 2015/2016

Versão 1.1

Docentes: Nuno Datia

Planeamento

As datas importantes a recordar são:

- Lançamento do enunciado: **21 de Setembro de 2015**
- Entrega intermédia (Fase 1): **07 de Dezembro de 2015**
- Entrega intermédia (Fase 2): **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 entregas, 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:

- ☐ Desenvolver um modelo de dados adequado aos requisitos, normalizado até à 3NF;
- ☐ Desenvolver uma solução baseada em bases de dados dinâmica, adequada aos requisitos;
- ☐ Utilizar correctamente vistas;
- ☐ Utilizar correctamente procedimentos armazenados;
- ☐ Utilizar correctamente gatilhos;
- ☐ Utilizar correctamente funções;
- ☐ Utilizar correctamente níveis de isolamento;
- ☐ Utilizar correctamente controlo transaccional.

Enunciado do trabalho (Documento de requisitos do sistema)

Foi colocado um desafio a um conjunto brilhante de alunos para criarem um sistema de *tickets* para ser usado em balcões de apoio ao cliente. Resolveu-se chamar a esse projecto “L-IckEtS”. O sistema poderá ser usado por diversos dispositivos, pelo que não é garantida a verificação de restrições por parte de quem usa a base de dados.

No âmbito desse sistema, um *ticket* é caracterizado por um código, tem um estado e uma descrição. Cada *ticket* é criado quando um utilizador envia um email para um endereço de apoio, ficando a descrição do *ticket* igual ao corpo do email. Inicialmente, o estado do *ticket* é de ‘Waiting’, podendo ter os estados ‘In progress’ ou ‘Closed’. Estes indicam que estão a ser feitas as diligências necessárias para resolver o problema e que o evento que originou o *ticket* foi considerado finalizado, respectivamente. Deve ficar registado no sistema a data de criação do *ticket*. Quando um utilizador criar um *ticket* fica registado no sistema, identificado pelo seu email. Eventualmente, pode ter um nome associado.

No sistema estão registados técnicos que efectuem as diligências para resolver o problema. Note-se, que essas diligências não são guardados no sistema e podem, em muitas situações ser acções no mundo real. Um técnico é caracterizado por um número (único), um email e um nome. Um *ticket* tem obrigatoriamente um técnico responsável, atribuído através de um

algoritmo de escalonamento, e.g. *Round-robin*. É o técnico responsável que analisa o *ticket* e lhe atribui um tipo, de entre os disponíveis no sistema e que lhe atribui uma prioridade em três possíveis — urgente, prioritário e normal. Quando um ticket passa a ter um tipo, automaticamente o seu estado passa para '*In progress*'. Um tipo de *ticket* tem um identificador único, um nome, e tem associado a si um conjunto de passos necessário à resolução desse tipo. Cada passo tem um número de ordem dentro de cada tipo e uma descrição. Por exemplo, o tipo "Login failed", tem associadas as seguintes passos: (i) Verificar se o nome de utilizador existe; (ii) Verificar se o processo de registo foi completado com sucesso; (iii) Fazer *reset* das credenciais associadas à conta. Note que não é necessário efectuar todos os passos associados a um tipo. Se a execução de um deles resolveu o problema, o *ticket* pode ser fechado.

Sempre que um *ticket* está '*In progress*' é possível tomar acções para resolver o problema nele relatado. Cada acção é única dentro de um *ticket* e tem associada a si o passo que visa resolver. Para cada acção é necessário registar a data de início, a data de fim (se existir) e nota associada à acção. Note-se que é necessário saber quem foi o técnico que executou a acção, podendo este ser diferente do técnico responsável pelo *ticket*. Nos requisitos actuais, um *ticket* só pode ser fechado pelo técnico responsável, devendo ficar registada a data de fecho. Qualquer alteração efectuada sobre um *ticket* tem de ficar registada no sistema. Desse registo deve constar: (i) O número da alteração, única para cada ticket; (ii) a data de alteração; (iii) o comando sql que foi usado na alteração; (iv) qual o **utilizador da base de dados** que efectuou a alteração. Note que, além disso, não pode ser removido do sistema as informações relativas a *tickets* e acções a eles associadas. Tenha também em atenção que: (i) um *ticket* não pode ser fechado se estiver no estado 'Waiting' (ii) um *ticket* não pode ser fechado não tiver nenhuma acção associada; (iii) um *ticket* pode mudar de prioridade, mas não pode ficar menos prioritário do que é.

Um utilizador pode pedir informações sobre o estado de um determinado *ticket*, ficando registado no sistema a data desse pedido. Note que só é permitido esse pedido ao utilizador que criou o *ticket*. Os técnicos (responsáveis ou envolvidos numa acção de resolução do *ticket*) podem responder aos pedidos de informação, ficando registado a data e o conteúdo da resposta. Note que deve ser possível determinar o tempo de resposta a cada pedido de informação.

É possível ordenar os *tickets* por prioridade, sendo a regra definida da seguinte forma:

1. Os *tickets* com prioridade urgente devem ser tratados primeiro do que os com prioridade prioritário
2. Os *tickets* com prioridade prioritário devem ser tratados antes daqueles com prioridade normal;
3. Entre os *tickets* com a mesma prioridade, tentam resolver-se primeiro os mais antigos;

Note que a cada 7 dias, o ticket sobe de prioridade, de forma automática.

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. O código T-SQL que permite:
 - (a) Criar o modelo físico;
 - (b) Inserir, remover e actualizar informação de um técnico;
 - (c) Inserir, remover e actualizar informação de um utilizador;
 - (d) Inserir, remover e actualizar informação de um *ticket*;
 - (e) Inserir um tipo de *ticket* e os passos a ele associados;
 - (f) Actualizar o estado de um *ticket*;
 - (g) Actualizar o prioridade de um *ticket*;
 - (h) Atribuir um técnico responsável a todos os *tickets* no estado '*Waiting*';
 - (i) Inserir uma acção associada a um *ticket*;
 - (j) Fechar um *ticket*;
 - (k) Inserir de forma automática o registo de alterações efectuadas sobre os *tickets*;
 - (l) Inserir uma resposta a um pedido de informação;
 - (m) Obter os n *tickets* mais prioritários.
 - (n) Listar os tickets. Deve ser possível ordená-los considerando o estado, a prioridade, a idade (em dias) e o tipo;
 - (o) Obter a informação de um ticket, usando o seu identificador. Devem ser incluídas as acções, os pedidos de informação e as resposta fornecidas de forma cronológica;
 - (p) Remover um *ticket*.

Data limite para entrega: 07 de Dezembro de 2015 até às 23:59.

A entrega deve incluir um relatório, o modelo EA e o código T-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 **segunda fase do trabalho**, os alunos devem ser capazes de:

- ☐ Utilizar correctamente transacções para garantir atomicidade nas operações, utilizando ADO.NET;
- ☐ Utilizar correctamente (ADO.NET) Entity Framework para acessos a dados;
- ☐ Utilizar correctamente ADO.NET em modo “conectado”;
- ☐ Garantir a libertação de recursos, quando estes não estejam a ser utilizados;
- ☐ Garantir as restrições de integridade aplicacionais;
- ☐ Utilizar XML e XML Schema para produzir documentos válidos e bem formados a partir de dados guardados num SGBD.

Enunciado do trabalho (2ª fase)

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

1. Crie uma aplicação .NET que, usando para acesso a dados o (ADO.NET) Entity Framework, permita disponibilizar as seguintes funcionalidades:
 - (a) Listar os tickets não fechados existentes no sistema, possibilitando o acesso ao detalhe destes (e.g. pedidos de informação).
 - (b) Atribuir um técnico responsável ao *ticket* mais recente.
 - (c) Inserir uma acção associada a um *ticket*, podendo ser possível alterar-lhe o estado;
 - (d) Remover um *ticket*;
 - (e) Exportar a informação de um *ticket*, obedecendo ao XML Schema em anexo;
2. Repita o exercício anterior, mas usando desta vez os objectos “conectados” do ADO.NET.
3. Compare as soluções usando as duas tecnologias, dos pontos de vista de facilidade de programação e garantia da consistência dos dados.

Data limite para entrega: 18 de Janeiro de 2016 até às 23:59.

A entrega deve incluir um relatório, os projectos *Visual Studio*, o código C# 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.

```

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="ticket" type="ticketType"/>
5      <xs:complexType name="ownerType">
6          <xs:simpleContent>
7              <xs:extension base="xs:string">
8                  <xs:attribute type="xs:string" name="ownerID" use="required"/>
9                  <xs:attribute type="xs:string" name="name"/>
10                 <xs:attribute type="xs:string" name="email" use="required"/>
11             </xs:extension>
12         </xs:simpleContent>
13     </xs:complexType>
14     <xs:complexType name="actionType">
15         <xs:simpleContent>
16             <xs:extension base="xs:string">
17                 <xs:attribute type="xs:string" name="orderNum" use="required"/>
18                 <xs:attribute type="xs:string" name="beginDate" use="required"/>
19                 <xs:attribute type="xs:string" name="endDate" use="optional"/>
20             </xs:extension>
21         </xs:simpleContent>
22     </xs:complexType>
23     <xs:complexType name="supervisorType">
24         <xs:simpleContent>
25             <xs:extension base="xs:string">
26                 <xs:attribute type="xs:string" name="technicianID" use="required"/>
27                 <xs:attribute type="xs:string" name="name" use="required"/>
28                 <xs:attribute type="xs:string" name="email" use="required"/>
29             </xs:extension>
30         </xs:simpleContent>
31     </xs:complexType>
32     <xs:complexType name="typeType">
33         <xs:simpleContent>
34             <xs:extension base="xs:string">
35                 <xs:attribute type="xs:string" name="typeID" use="required"/>
36                 <xs:attribute type="xs:string" name="name" use="required"/>
37             </xs:extension>
38         </xs:simpleContent>
39     </xs:complexType>
40     <xs:complexType name="actionsType">
41         <xs:sequence>
42             <xs:element type="actionType" name="action" maxOccurs="unbounded" minOccurs="0"/>
43         </xs:sequence>
44     </xs:complexType>
45     <xs:complexType name="ticketType">
46         <xs:sequence>
47             <xs:element type="ownerType" name="owner"/>
48             <xs:element type="supervisorType" name="supervisor"/>
49             <xs:element type="xs:string" name="description"/>
50             <xs:element type="typeType" name="type"/>
51             <xs:element type="actionsType" name="actions"/>
52         </xs:sequence>
53         <xs:attribute type="xs:string" name="type" use="required"/>
54         <xs:attribute type="xs:string" name="ticketID" use="required"/>
55         <xs:attribute type="xs:string" name="status" use="required"/>
56     </xs:complexType>
57 </xs:schema>

```