

---

## Licenciatura em Engenharia Informática e de Computadores

Área Departamental de Engenharia de Electrónica e Telecomunicações e de Computadores

2º Teste, 01 de Fevereiro de 2019

### Sistemas de Informação II

---

Duração: **2h30m**

Justifique devidamente cada resposta.

1. Considere uma base de dados para registar consultas feitas num hospital e receitas de fármacos feitas em cada consulta. Essa base de dados inclui as tabelas criadas pelo Código 1.

```
create table Farmaco(  
  NumF int primary key,  
  NomeFarmaco varchar(100),  
  Stock int not null  
)  
  
create table Medico(  
  CodMed int primary key,  
  Nome varchar(100) not null,  
  Genero char not null,  
  Especialidade varchar(100)  
)  
  
create table Consulta(  
  NumCons int primary key,  
  codMed int not null,  
  NomePaciente varchar(100) not null,  
  Data datetime not null,  
  constraint c1 foreign key(codMed)  
    references Medico  
)  
  
create table Receita(  
  NumCons int not null,  
  NumF int not null,  
  quantidade int not null,  
  primary key(numcons,numf)  
)
```

Código 1: Ilustração do modelo de dados usado nas questões

- (a) [2] Implemente, utilizando T-SQL, um *trigger* sobre a tabela *Receita* que garanta que uma receita apenas é emitida caso haja stock suficiente para todos os fármacos prescritos. Garanta a consistência dos stocks guardados na base de dados.
- (b) [4.5] Imagine que pretende eliminar todas as consultas referente a um médico quando este é eliminado.
- Implemente um *trigger* T-SQL sobre a tabela *Medico* de forma a obter a funcionalidade pretendida fazendo uso de cursores.
  - Implemente um *trigger* T-SQL sobre a tabela *Medico* de forma a obter a funcionalidade pretendida sem qualquer utilização de cursores.
  - Qual dos *triggers* terá melhor desempenho?
- (c) [1.5] Defina a função *multi-statement* T-SQL `top_MedicosPorFarmaco (NumF)` que apresenta os dez médicos (*Nome*) que mais prescreveram o fármaco indicado e o respectivo total da quantidade prescrita.
- (d) [1.5] Implemente uma vista que apresente todos os fármacos sem stock ainda não prescritos. Comente se a vista implementada é alterável.
- (e) [1] Indique as principais vantagens da utilização do padrão *Virtual Proxy*.
- (f) [2.5] Para uma camada de acesso a dados foram definidas as classe *Medico* e *Consulta*, como ilustrado no Código 2. Implemente uma solução em C# cujo carregamento das consultas dadas por um médico é realizado seguindo o padrão *Virtual Proxy*.

```

public class Medico{
    public Medico() {...}
    public int Nome { get; set; }
    public string Genero { get; set; }
    public string Especialidade { get; set; }
    public List<Consulta> consultasDadas { get; set; }
}

public class Consulta {
    public Consulta() {...}
    public int NumConsulta { get; set; }
    public int CodMed { get; set; }
    public string NomePaciente { get; set; }
    public string Data { get; set; }
}

```

Código 2: Definição das classes Medico e Consulta

2. Considere o processamento concorrente nos sistemas de gestão de bases de dados, responda às questões:
- (a) [1.5] As transações no escalonamento ilustrado na Tabela 1, são executadas com nível de isolamento REPEATABLE READ. Assumindo que as tabelas da base de dados se encontram vazias, indique que valores são escritos pelas instruções PRINT sabendo que n1 e n2 são variáveis numéricas. Justifique se as transações são serializáveis.

Passo	Transação T1	Transação T2
1	BEGIN TRAN;	
2		BEGIN TRAN;
3		SELECT COUNT(*) INTO n2 FROM Medico;
4		PRINT n2;
5		INSERT INTO Medico VALUES('1',...);
6	INSERT INTO Medico VALUES('2',...);	
7		SELECT COUNT(*) INTO n2 FROM Medico;
8		PRINT n2;
9		COMMIT;
10	SELECT COUNT(*) INTO n1 FROM Medico;	
11	PRINT n1;	
12	COMMIT;	

Tabela 1: Transacções T1 e T2

- (b) [2] Apresente um escalonamento com duas transações concorrentes em modo de isolamento REPEATABLE READ (recorrendo a código T-SQL) que exiba conflitos entre pelo menos duas instruções e que seja observado *phantom tuples*.
- (c) [1.5] Considere a execução das instruções ilustradas no Código 3. Assuma que as tabelas usadas existem na base de dados, mas não contêm registos. A transacção `inserirConsulta` termina com sucesso? Porquê? Caso considere que a transacção não termina com sucesso, adicione o código necessário à execução da transacção de forma a que esta possa terminar com sucesso, justificando.

Note que não é permitido alterar o esquema da base de dados nem alterar a ordem de execução das instruções.

```
begin tran inserirConsulta
insert into Consulta values (1, 1, 'John', '28-01-2019')
insert into Medico values ('1', 'Andre', 'M', 'Neurologia')
commit
```

Código 3: Transacção inserirConsulta

- (d) [2] No que consiste uma Saga? Quais as vantagens na sua utilização face ao conceito transaccional tradicional? Numa Saga, como é garantida a atomicidade? Discuta este tópico e justifique a sua resposta com exemplos.

01 de Fevereiro de 2019, Nuno Datia, Ricardo Silva