



INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

Departamento de Engenharia de Eletrónica e Telecomunicações e de  
Computadores

Licenciatura de Engenharia Informática e de Computadores

Introdução a Sistemas de Informação

Semestre de Verão 2021-2022

Gestão de casas de apostas

Trabalho prático

( Fase 2 )

Ana Rita Beire e Matilde Pós-de-Mina Pato

Abril, 2022

## Planeamento

As datas importantes a recordar são:

- Lançamento do enunciado: **22 de março 2022**
- Entrega intermédia (Fase 1): **8 de abril de 2022**
- Entrega intermédia (Fase 2): **6 de maio de 2022**
- Entrega intermédia (Fase 3): **17 de junho de 2022**

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 correções 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: 'GrupoNNFaseN.ext' (N representa um dígito, e 'ext' a extensão do ficheiro), exemplo: Grupo01Fase1.pdf ou Grupo14Fase1.pdf.

28 de abril de 2022, Ana Rita Beire e Matilde Pós-de-Mina Pato

## Objetivos de aprendizagem

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

- Utilizar corretamente a álgebra relacional, com os seus vários operadores, para expressar interrogações sobre um modelo relacional:
  - o operadores relacionais unários: select, project e rename;
  - o operadores relacionais binários: division;
  - o operadores sobre conjuntos: difference, intersect e union;
  - o operadores de junção: interna e externa;
  - o operador de agregação.
- Utilizar corretamente 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 transacional;
- Utilizar corretamente os operadores da teoria dos conjuntos em PostgreSQL;
- Utilizar corretamente as cláusulas INNER JOIN e OUTER JOIN;
- Utilizar corretamente sub-interrogações correlacionadas;
- Utilizar corretamente funções de agregação;
- Utilizar corretamente a cláusula HAVING;
- Utilizar corretamente a cláusula ORDER BY;
- Utilizar corretamente o termo DISTINCT;
- Utilizar corretamente 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 PostgreSQL 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 transacional (a atomicidade).

## Resultados pretendidos

Tendo em conta os objetivos 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 PostgreSQL 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 às seguintes alíneas:
  - a) Pretende-se o registo de todos os jogadores (casa de apostas, email, nome e nickname) que se encontram ativos;
  - b) Liste todos os administradores (email, nome e perfil) agrupados por perfil;
  - c) Apresente a lista de casas de apostas (id, nome e NIPC) juntamente com o número total de jogadores registados;
  - d) Apresente o número de documentos submetidos por jogador;
  - e) Pretende-se saber a lista de todas as apostas que o "Manuel Fernandes" fez.
3. Conceba, as interrogações que produzam os resultados a seguir indicados, utilizando apenas uma instrução PostgreSQL. Guarde num script autónomo de nome "queries.sql". **Para cada instrução deve ser também apresentada a descrição do raciocínio seguido.**
  - a) Implemente em PostgreSQL as interrogações pedidas na alínea 2;
  - b) Apresente o nome de todos os administradores por perfil, deve apresentar a lista dos nomes por ordem alfabética (em primeiro lugar) ~~e a data de registo;~~

- c) Identifique todos os jogadores que fizeram pelo menos um levantamento. O resultado deve apresentar o nome, o email e o nickname;
  - d) Liste todas as apostas pendentes de resolução. O resultado deve identificar a aposta (casa de apostas, e-mail e transação), o tipo e a descrição;
  - e) Crie uma vista que permita obter as apostas realizadas no ano passado (deve fazer uso de uma função de datetime e não fixar o valor 2022), incluindo o estado da resolução;
  - f) Crie uma vista que permita obter o saldo dos jogadores, sabendo que:  
 $\text{saldo} = \text{depósitos} - \text{levantamentos} - \text{apostas} + \text{resoluções}$ .
4. Apresente o(s) comando(s) que permitem inserir na BD a seguinte informação: Aposta simples no evento “FCP vs SLB - Empate” com a odd de 1.2. O valor dessa aposta é de 10€. Afetem essa aposta a um jogador já registado na BD.

Todas as simplificações e otimizações 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: 06 de maio de 2022 até às 23:59.**

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

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

Instituto Superior de Engenharia de Lisboa  
Departamento de Engenharia de Eletrónica e Telecomunicações e de  
Computadores

LEIC

## **Modelo Relacional a usar no trabalho prático (fase 2)**

**Introdução a Sistemas de Informação**

Semestre de Verão 2021/2022

Versão 1.0

## Modelo Relacional

Todos os atributos são obrigatórios nas relações, exceto quando indicado o contrário. As relações são apresentadas por ordem alfabética. **Nota:** Poderão não estar representadas todas as restrições de integridade, uma vez que esta componente também é avaliada. Mais, as restrições que não conseguirmos implementar no modelo físico serão asseguradas, posteriormente, na API que irão desenvolver na 3ª parte.

### ADMINISTRADOR

ADMINISTRADOR(id, email, nome, perfil, casa\_apostas)

| Atributo     | Tipo         | Restrições Integridade  |
|--------------|--------------|---|
| id           | int          | Foi adicionado este atributo como identificador de ADMINISTRADOR por questões de simplificação. Valor sequencial. |
| email        | varchar(60)  | O valor é do tipo "{C}@{C}", onde C representa caracter.<br>O e-mail é único, por casa de aposta.                 |
| nome         | varchar(150) |   |
| perfil       | varchar(15)  | Toma os valores "administrador", "supervisor" ou "operador".  |
| casa_apostas | int          | FK referência de CASA_APOSTAS.{id}.   |

### APOSTA

APOSTA(transacao, tipo, odd, descricao)

| Atributo  | Tipo         | Restrições Integridade                   |
|-----------|--------------|--|
| transacao | int          | FK referência de TRANSACAO.{numero}.     |
| tipo      | varchar(15)  | Toma os valores "simples" ou "múltipla". |
| odd       | real         | Valor superior a 1.                      |
| descricao | varchar(150) |  |

### BANCARIA

BANCARIA(transacao, operacao)

| Atributo  | Tipo        | Restrições Integridade                        |
|-----------|-------------|---|
| transacao | int         | FK referência de TRANSACAO.{numero}.          |
| operacao  | varchar(15) | Toma os valores "depósito" ou "levantamento". |

## CASA\_APOSTAS

CASA\_APOSTAS(id, nome, NIPC, aposta\_minima)

| Atributo      | Tipo         | Restrições Integridade                                     |
|---------------|--------------|--|
| id            | int          | Valor sequencial.  |
| nome          | varchar(150) |  |
| NIPC          | varchar(9)   | Corresponde ao Número de Identificação de Pessoa Coletiva. |
| aposta_minima | real         | Valor em €, com 2 casas decimais.                          |

## DOCUMENTO

DOCUMENTO(jogador, numero, descricao, estado, data\_registro)

| Atributo       | Tipo         | Restrições Integridade   |
|----------------|--------------|--|
| jogador        | int          | FK referência de JOGADOR.{id}.   |
| numero         | int          |  |
| descricao      | varchar(150) |  |
| estado         | varchar(15)  | Toma os valores “pendente”, “aceite” ou “recusado”.  |
| data_submissao | date         | Tem o formato “dd-mm-aaaa”. A data de submissão tem que ser superior à data de registro (da tabela JOGADOR.{data_registro}). |

## JOGADOR

JOGADOR(id, email, nome, nickname, estado, data\_nascimento, data\_registro, morada, codigo\_postal, localidade, casa\_apostas)

| Atributo        | Tipo         | Restrições Integridade  |
|-----------------|--------------|---|
| id              | int          | Foi adicionado este atributo como identificador de JOGADOR por questões de simplificação. Valor sequencial. |
| email           | varchar(60)  | O valor é do tipo “{C}@{C}”, onde C representa caracter.<br>O e-mail é único, por casa de aposta.           |
| nome            | varchar(150) |   |
| nickname        | varchar(20)  | O nickname é único, por casa de aposta.   |
| estado          | varchar(15)  | Toma os valores “ativo”, “suspense” ou “autoexcluído”.  |
| data_nascimento | date         | Tem o formato “dd-mm-aaaa”. A pessoa deve ter no mínimo 18 anos de idade à data atual.                      |
| data_registro   | date         | Tem o formato “dd-mm-aaaa”. A data_registro deve ser sempre superior a                                      |



|               |              |   |
|---------------|--------------|---|
|               |              | data_nascimento + 18 e menor que a data atual.                      |
| morada        | varchar(150) |   |
| codigo_postal | int          | Valor que contém 7 dígitos, que compõem o código postal, sem hífen. |
| localidade    | varchar(50)  |   |
| casa_apostas  | int          | FK referência de CASA_APOSTAS.{id}.                                 |

## RESOLUCAO

RESOLUCAO(id, valor, resultado, data\_resolucao, aposta)

| Atributo       | Tipo        | Restrições Integridade  |
|----------------|-------------|---|
| id             | int         | Foi adicionado este atributo como identificador de RESOLUCAO por questões de simplificação. Valor sequencial.                 |
| valor          | real        | Valor em €, com 2 casas decimais.   |
| resultado      | varchar(15) | Toma os valores “vitória”, “derrota”, “cashout” ou “reembolso”.   |
| data_resolucao | date        | Tem o formato “dd-mm-aaaa”. A data de resolução tem que ser superior à data da aposta (da tabela TRANSACAO.{data_transacao}). |
| aposta         | int         | FK referência de TRANSACAO.{numero}.  |

## TRANSACAO

TRANSACAO(numero, valor, data\_transacao, casa\_apostas, jogador)

| Atributo       | Tipo | Restrições Integridade   |
|----------------|------|--|
| numero         | int  | Valor sequencial.  |
| valor          | real | Valor em €, com 2 casas decimais.  |
| data_transacao | date | Tem o formato “dd-mm-aaaa”. A data de transacao tem que ser superior à data de registo (da tabela JOGADOR.{data_registo}). |
| casa_apostas   | int  | FK referência de CASA_APOSTAS.{id}.  |
| jogador        | int  | FK referência de JOGADOR.{id}.   |