

**Primeira Fase Do Trabalho De Sistemas De Informação – T43D – G05**

|  |  |
| --- | --- |
|  | Nuno Bartolomeu |
|  | João Viegas  Miguel Moreira |

|  |  |
| --- | --- |
| Professor: | Nuno Leite |
|  |

Relatório final realizado no âmbito de Sistemas de Informação,  
do curso de licenciatura em Engenharia Informática e de Computadores  
Semestre de Verão 2022/2023

Maio de 2023

**Instituto Superior de Engenharia de Lisboa**  
Licenciatura em Engenharia Informática e de Computadores

**Primeira Fase Do Trabalho De Sistemas De Informação – T43D – G05**

|  |  |
| --- | --- |
| 47233 | Nuno António Oliveira Bartolomeu |
| 47000 | João Rei Da Porra Toda Viegas |
| 47000 | Miguel The Boss Moreira |

|  |  |
| --- | --- |
| Professor: | Dr. Nuno Miguel da Costa de Sousa Leite |
|  |  |

Relatório final realizado no âmbito de Sistemas de Informação,  
do curso de licenciatura em Engenharia Informática e de Computadores  
Semestre de Verão 2022/2023

Maio de 2023

# Resumo

A primeira fase do trabalho de sistemas de informação visa perceber as capacidades dos alunos para desenvolver modelos de dados adequados, implementar tal modelo para uma base de dados dinâmica e utilizar corretamente as ferramentas do SQL, sabendo justificar a necessidade das mesmas.

**Palavras-chave:** lista de palavras-chave, ordenadas alfabeticamente, separadas por ;.

**Índice**

[Resumo v](#_Toc417484088)

[Lista de Figuras xiii](#_Toc417484091)

[Lista de Tabelas xv](#_Toc417484092)

[1. Introdução 1](#_Toc417484093)

[1.1 Nome da secção deste capítulo 1](#_Toc417484094)

[1.2 A segunda secção deste capítulo 1](#_Toc417484095)

[1.2.1 A primeira sub-secção desta secção 1](#_Toc417484096)

[1.2.2 A segunda sub-secção desta secção 1](#_Toc417484097)

[1.3 Organização do documento 2](#_Toc417484098)

[2. Formulação do Problema 3](#_Toc417484099)

[2.1 Nome da secção deste capítulo 3](#_Toc417484100)

[2.2 Análise do problema - enumeração 3](#_Toc417484101)

[2.3 Outro problema - tabela 4](#_Toc417484102)

[2.4 Expressões matemáticas 4](#_Toc417484103)

[2.5 Figuras de grande dimensão 4](#_Toc417484104)

[3. Solução Proposta - Grandes Ideias 7](#_Toc417484105)

[3.1 Nome da primeira secção deste capítulo 7](#_Toc417484106)

[3.2 A segunda secção deste capítulo 8](#_Toc417484107)

[3.2.1 A primeira sub-secção desta secção 8](#_Toc417484108)

[3.2.2 A segunda sub-secção desta secção 8](#_Toc417484109)

[3.3 Descrição detalhada da solução 8](#_Toc417484110)

[4. Avaliação Experimental 11](#_Toc417484111)

[4.1 Nome da primeira secção deste capítulo 11](#_Toc417484112)

[4.2 A segunda secção deste capítulo 11](#_Toc417484113)

[4.2.1 A primeira sub-secção desta secção 11](#_Toc417484114)

[4.2.2 A segunda sub-secção desta secção 11](#_Toc417484115)

[4.3 Análise de resultados 11](#_Toc417484116)

[5. Conclusões 13](#_Toc417484117)

[Referências 14](#_Toc417484118)

[A.1 Diagramas da Aplicação 15](#_Toc417484119)

[A.2 Modelos de dados 17](#_Toc417484120)

# 

# Lista de Figuras

[Figura 1 - Legenda da figura com o logotipo do ISEL. 1](#_Toc416101905)

[Figura 2 - Legenda da figura com o logotipo do ISEL – versão 2. 5](#_Toc416101906)

[Figura 3 – Diagrama de casos de utilização. 15](#_Toc416101907)

# Lista de Tabelas

[Tabela 1 -Um exemplo de legenda de tabela. Prazos de entrega de Projecto e Seminário, 4](#_Toc416101908)

# Problema

A empresa “GameOn” pretende desenvolver um sistema para gerir jogos, jogadores e as partidas que estes efetuam. O sistema deve registar os dados dos jogadores que são identificados por um id gerado pelo sistema, tendo também o email e o username como valores únicos e obrigatórios. O jogador toma um dos estados ‘Ativo’, ‘Inativo’ ou ‘Banido’ e pertence a uma determinada região. Para cada região apenas se deve registar o seu nome. Os jogos têm como identificador uma referência alfanumérica de dimensão 10, um nome obrigatório e único e um url para uma página com os detalhes do jogo. Interessa registar os jogadores que compraram determinado jogo, a data e o preço associados à compra. Cada vez que o jogo é jogado, é criada uma partida com um número sequencial e único para cada jogo, devendo ser guardadas as datas e horas de início e de fim da partida. A partida pode ser normal de apenas um jogador, ou multi-jogador. As partidas normais devem ter informação sobre o grau de dificuldade (valor de 1 a 5) e estar associadas ao jogador que as joga e à pontuação por ele obtida. As partidas multi-jogador devem estar associadas aos jogadores que as jogam sendo necessário guardar as pontuações obtidas por cada jogador em cada partida. Devem ainda conter informação sobre o estado em que se encontram, o qual pode tomar os valores ‘Por iniciar’, ‘A aguardar jogadores’, ‘Em curso’ e ‘Terminada’. Assume-se a existência de um sistema que atualiza a pontuação e estado durante a execução do jogo. Cada partida está associada a uma região e apenas jogadores dessa região a podem jogar. De modo a recompensar os jogadores, cada jogo pode ter um conjunto de crachás que são atribuídos aos jogadores quando um limite de pontos nesse jogo é atingido. Para isso interessa registar o nome do crachá que é único para cada jogo, o limite de pontos e o url para a imagem do crachá. Devem ficar registados na base de dados os crachás que são atribuídos a cada jogador. Deverão existir em tabelas próprias as estatísticas associadas aos jogadores e aos jogos. Interessa registar para cada jogador, o número de partidas que efetuou, o número de jogos diferentes que jogou e o total de pontos de todos os jogos e partidas efetuadas. Para cada jogo interessa registar o número de partidas, o número de jogadores e o total de pontos. Os jogadores podem adicionar outros jogadores como amigos. Portanto interessa registar essa relação de amizade. É também possível criar uma conversa entre vários jogadores com um identificador gerado pelo sistema e um nome. Associado à conversa existem mensagens com um número de ordem único e sequencial para cada conversa que serve de identificador, a data e hora da mensagem, o texto e qual o jogador que enviou a mensagem.

# Solução

A nossa solução é apresentada neste capítulo.

## 2.1 Divisão em tabelas

Após leitura do enunciado o grupo chegou a 14 tabelas essenciais para a resolução do problema:

* Jogadores;
* Regiões;
* Jogos;
* Compras;
* Partidas;
  + Normais;
  + Multi-jogador;
* Crachás;
* Estatísticas;
  + Jogador;
  + Jogo;
* Amigos;
* Conversas;
* Mensagens;

Após começarmos a tentar implementar a solução ficou aparente que precisamos de mudar algumas coisas no nosso modelo:

* Apagar a tabela estatísticas e atualizar as tabelas derivadas para: “Estatísticas Jogador” e “Estatísticas Jogo”. Apesar da partilha de atributos ambas as tabelas têm uma chave primária diferente, uma sendo derivada de jogador e uma de jogo.
* Criação da tabela “Pontuações” para guardar os pontos de cada partida.
* Transformação de algumas das entidades para fracas.

## 2.2 Diagrama ER

A picture containing diagram, line, plan, plot

Description automatically generatedUsando o que descrevemos no tópico anterior criámos o seguinte diagrama entidade-relação:

Figura 1 – Diagrama ER.

### 2.3 Modelo físico

O modelo físico segue o modelo Er adicionando as restrições de integridade e está presente na pasta de código sobre o nome “createTables.sql”.

### 2.4 Funcionalidades

As funcionalidades do projeto estão na pasta de código sobre o nome “exercises.sql” e os respetivos testes estão sobre o nome “tests.sql”.

### 2.5 Preparação para a segunda fase

Algumas funcionalidades especificas deixamos para a segunda fase, visto que faz mais sentido para o grupo fazê-las com as ferramentas que o Java proporciona, já que são mais adequadas para estas operações do que o Postgres.

Por exemplo:

* Adicionar uma verificação para garantir que um jogador que esteja banido não possa comprar jogos, ou jogar partidas.
* Garantir que os jogadores só podem jogar partidas que sejam feitas na mesma região que eles.
* Não deixar jogadores inativos criar conversas ou enviar mensagens.

# Conclusões

Neste trabalho recebemos um problema que pode ser facilmente associado a realidade para um programador. Aprendemos a ideologizar o código primeiro para podermos ter um caminho concreto a seguir e a verificar as funcionalidades após ser implementadas para garantir que fazem o que é suposto. Também tivemos em consideração que esta é só a primeira parte do trabalho e tomamos algumas decisões baseadas no que vamos ter de fazer para a segunda parte.

# Referências

|  |  |
| --- | --- |
| [1] | Wikipedia, “Big data --- Wikipedia, The Free Encyclopedia,” http://en.wikipedia.org/w/index.php?title=Big\_data&oldid=648786139, 2015. |
| [2] | X. Ding, X. Zhu e G. Wu, “Data mining with big data,” *IEEE Transactions on Knowledge and Data Engineering,* vol. 26, n.º 1, pp. 97-107, 2014. |
| [3] | J. Andrews, S. Buzzi, W. Choi, S. Hanly, A. Lozano, A. Soong e J. Zhang, “What Will 5G Be?,” *IEEE Journal on Selected Areas in Communications,* vol. 32, n.º 6, pp. 1065-1082, 2014. |
| [4] | L. Boytsov, “Indexing Methods for Approximate Dictionary Searching: Comparative Analysis,” *J. Exp. Algorithmics,* vol. 16, n.º may, p. 1.81, 2011. |
| [5] | T. Jurkiewicz e K. Mehlhorn, “On a Model of Virtual Address Translation,” *J. Exp. Algorithmics,* vol. 19, n.º jan, pp. 1-18, 2015. |
| [6] | J. Neumann, The Computer and the Brain, New Haven, CT, USA: Yale University Press, 1958. |
| [7] | B. Kernighan e P. Plauger, The Elements of Programming Style, New York, NY, USA: McGraw-Hill, Inc., 1982. |