



**Escola Superior
de Tecnologia
e Gestão**

Politécnico de Coimbra

Relatório de Projecto

Programação Aplicada | Programação IV

Avaliação [Periódica 2]

Autor:

<Daniel Santos | N°2019133865>

Data: <21/05/2022>

Resumo

Este trabalho consiste na explicação/demonstração de como o projeto periódico 2 de PA foi desenvolvido.

Palavras-chave

Base de Dados, Programação, Sockets.

Índice

Resumo.....	v
Lista de Figuras	xi
Lista de Tabelas.....	xiii
Lista de Acrónimos.....	xv
1. Introdução.....	17
2. Objectivos e Metodologias	19
2.1. Ferramentas e Tecnologias	19
2.2. Planeamento	19
3. Trabalho Desenvolvido	21
3.1. Requisitos Não Implementados	21
3.2. Classes e Packages.....	21
3.3. Algoritmos e Estrutura de Dados	21
3.4. Armazenamento de Dados.....	22
3.5. Procedimentos de Teste.....	23
4. Manual de Utilizador	24
4.1. Regras antes de executar o programa	24
4.2. Login	24
4.3. Menu	25
5. Conclusões	27
5.1. Forças	27
5.2. Limitações	27
5.3. Trabalho Futuro.....	27
6. Referências.....	29
7. Anexos	31

Lista de Figuras

Figura 1 - Base de Dados	22
Figura 2 - Programa login	Erro! Marcador não definido.
Figura 3 - Menu Gestor	Erro! Marcador não definido.
Figura 4 - Menu Especialista	Erro! Marcador não definido.
Figura 5 – Properties.....	24
Figura 6 – Login.....	Erro! Marcador não definido.
Figura 7 - Criar conta	Erro! Marcador não definido.
Figura 8 - Sair	Erro! Marcador não definido.
Figura 9 - MenuGestor.....	25

Lista de Tabelas

Não foi encontrada nenhuma entrada do índice de ilustrações.

Lista de Acrónimos

ER	Modelo Entidade-Relacionamento
PA	Programação Aplicada
BD	Base de Dados

1. Introdução

Este trabalho foi realizado no âmbito da disciplina de PA por Daniel Santos a pedido de trabalho do professor Marco Veloso.

Neste trabalho aprendi a comunicação com sockets de forma consistente para um jogo de quatro em linha em multiplayer.

O relatório será apresentado através da descrição do que fui fazendo e com o devido acompanhamento de screenshots.

2. Objectivos e Metodologias

2.1. Ferramentas e Tecnologias

Como tecnologia usei a linguagem Java e usei Query's de SQL.

Para desenvolver o programa usei ferramentas como o IntelliJ Community (java), powerdesigner(criação da BD), heidiSQL(manutenção/Query's da BD).

2.2. Planeamento

O trabalho foi desenvolvido de forma consistente no período desta última semana (4 horas por dia), sendo que, o maior tempo dedicado ao trabalho foi nas sockets de maneira a evitar erros ou inconsistências na sua comunicação.

3. Trabalho Desenvolvido

3.1. Requisitos Não Implementados

Não foi desenvolvido as notificações do sistema [R30],[R32]-[R35].

3.2. Classes e Packages

Existe apenas um package que se chama projeto2.

Não existe heranças.

Existem classes que gerem objetos de outras classes: GereJogador, GereJogo, GereLog, GereProperties.

Existem classes que processam outras classes: Cliente, Servidor, TratamentoMensagens, PropertiesDados.

Por fim, existe a classe Main que é onde o código pode ser executado.

3.3. Algoritmos e Estrutura de Dados

Sobre os algoritmos e as estruturas de dados implementados é sempre complicado de falar porque às vezes há dezenas de formas de resolver um problema não havendo uma solução ideal, eu tentei sempre criar algoritmos eficazes e reutilizáveis de acordo com a solução que para mim era a ideal.

Para a realização destes algoritmos foram utilizadas várias estruturas de dados, todas elas que para mim facilitaram muito:

Uso de listas para retornar algo de um algoritmo: Em muitas situações ao longo do meu código recorri a listas de java para retornar tanto inteiros como strings como outro tipo de objetos, sendo esta uma forma para mim bastante eficaz quando pretendemos passar o objeto por um todo e não apenas partes do mesmo.

3.4. Armazenamento de Dados

Base de dados usada:

Jogador			
<u>idJogador</u>	<pi>	Integer	<M>
nome		Variable characters (30)	
nJogos		Integer	
nVitorias		Integer	
tempoTotal		Integer	
id	<pi>		

Simulacao			
<u>idJogo</u>	<pi>	Integer	<M>
posicao		Variable characters (500)	
id	<pi>		

Figura 1 - Base de Dados

3.5. Procedimentos de Teste

Para testar o meu código, principalmente teste à comunicação com sockets, fui implementando um tipo de mensagem de cada vez e testando até que cheguei a um resultado final e consistente.

Quando o código estava em “construção” fui sempre dando `print()` às mensagens recebidas para saber onde as processar.

No final acabei por testar todas as opções possíveis disponibilizadas pelo multiplayer e não detetei nenhum erro.

4. Manual de Utilizador

4.1. Regras antes de executar o programa

Para não ocorrerem erros o programa deve arrancar com as configurações de acesso necessárias no documento Properties.txt, tal como mostra a figura:

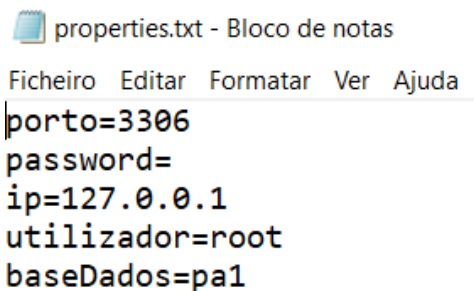


Figura 2 – Properties

Para navegar pelos menus use exclusivamente os números indicados em cada instrução, enquanto não inserir um número não sai do menu onde se encontra.

Quando não conseguir avançar a um determinado pedido (esse pedido repete-se), o mais provável é que o valor que esteja a colocar já existe ou não pode ser inserido, a maior parte das vezes há uma mensagem a informar o ocorrido.

Sempre que é pedido apenas clique ENTER para continuar, a inserção de outros caracteres pode causar erros e mau funcionamento do programa.

Introduzir valores nulos é possível em alguns casos mas pode causar erros no programa posteriormente, para tal deve sempre introduzir algo quando lhe é pedido.

4.2. Login

No login é bastante simples pois só exige um nome e carregar no Enter, ou, se deseja entrar como anónimo pode apenas carregar no botão ENTER:

```
Bem vindo ao jogo 4 em linha!
Introduza o nome do jogador (para entrar como anónimo apenas pressione ENTER): ola
Bem-vindo ola!
```

Figura 3 - Login

4.3. Menu

Existe apenas um Menu principal neste programa:

```
Bem vindo ao jogo 4 em linha!  
Introduza o nome do jogador (para entrar como anónimo apenas pressione ENTER): ola  
Bem-vindo ola!  
Modos de jogo:  
1.Singleplayer 2.Local Multiplayer(host) 3.Local Multiplayer(client)  
4.Remote Multiplayer(host) 5.Remote Multiplayer(client)  
Listagens:  
6.Listar jogadores 7.Listar jogos 8.Listar Log  
0.Sair  
Introduza a opção que deseja:
```

Figura 4 – Menu Principal

Os menus têm todos este tipo de organização, apenas tenha em atenção que certos menus podem ter submenus.

4.4. Multiplayer

As opções na comunicação multiplayer é bastante restringida e o programa dará feedback sempre que o jogador responder algo diferente do que é pedido.

Existe um caso particular que é a inicialização de server e clientes pois se um jogador introduzir a porta ou o ip mal (no caso do cliente) então o programa dará erro mas não afetará outras instâncias abertas na mesma máquina, ou em máquinas diferentes:

```
Introduza a opção que deseja: 2  
Indique o porto: 4000
```

Figura 5 - Inicialização Servidor

Se respeitar estas regras o programa deve correr a 100% e sem qualquer tipo de erros!

5. Conclusões

5.1. Forças

Como eu estou no meu terceiro ano de licenciatura penso que tenho mais prática e conhecimento em Java e não sinto tanta dificuldade no desenvolvimento do código.

5.2. Limitações

Devido a ter muitos trabalhos de outras disciplinas não pude dedicar o tempo que queria para o projeto sendo que não consegui “polir os cantos” ao programa.

Acabei por fazer sockets como aprendemos no segundo ano e tenho pena não ter feito como aprendemos no terceiro ano pois facilitaria o desenvolvimento do projeto e a sua organização na comunicação.

5.3. Trabalho Futuro

Apesar de ter cumprido 90% dos requisitos, pode haver falhas devido á falta de tempo para testar e rever se tudo está correto, portanto para além da implementação do que falta futuramente irei melhorar também: prevenção contra o que o utilizador vai escrever, métodos mais curtos e objetivos, menos repetição de código, algoritmos mais eficientes.

6. Referências

[1]

- [1] Optimus, «Resultset error», *Stack Overflow*, 20 de janeiro de 2013.
<https://stackoverflow.com/q/14424261> (acedido 21 de maio de 2022).

7. Anexos

- LEI_PA_Relatorio_PT.pdf
- /JavaDoc
- /Java
- DiagramaClasses.png
- Projeto2.jar