Relatório

Arquiteturas Móveis 2017/2018



Realizado por:

Miguel Almeida, nº21220106

Bruno Santos, nº21190312

Conteúdo

[Introdução 3](#_Toc503373505)

[Recursos utilizados do dispositivo 4](#_Toc503373506)

[Câmara 4](#_Toc503373507)

[SharedSettings 4](#_Toc503373508)

[Armazenamento de Dados 4](#_Toc503373509)

[Comunicação 4](#_Toc503373510)

[Suporte a diferentes línguas e ecrãs 4](#_Toc503373511)

[Organização 5](#_Toc503373512)

[MainActivity 6](#_Toc503373513)

[AboutActivity 6](#_Toc503373514)

[ProfileActivity 7](#_Toc503373515)

[CameraActivity 7](#_Toc503373516)

[GameActivity 8](#_Toc503373517)

[HistoryActivity 8](#_Toc503373518)

[History2Activity 9](#_Toc503373519)

[TypeGameActivity 10](#_Toc503373520)

[Win\_Activity 11](#_Toc503373521)

[MyService 11](#_Toc503373522)

[Organização de Recursos 12](#_Toc503373523)

[Conclusão 13](#_Toc503373524)

# Introdução

Neste relatório de execução pretende-se abordar a solução elaborada por nós para a execução de um jogo de xadrez proposto como trabalho prático na unidade curricular de Arquitecturas Moveis, referente a matéria de Android.

Desta forma, nas próximas paginas irão ser abordadas a varias decisões tomadas com vista a execução final do jogo. Irá ser também ser explicada toda a hierarquia e organização de código.

# Recursos utilizados do dispositivo

## Câmara

Neste trabalho é utilizado a câmara do dispositivo para captar uma imagem a adicionar ao perfil do utilizador. Para o efeito o utilizador, se assim o entender, poderá no seu perfil tirar uma foto utilizado a câmara da frente ou de trás do telefone.

## SharedSettings

A imagem falada no ponto anterior terá que ser guardada para futuro, assim como todo o perfil do utilizador. Desta forma a solução para guardar estas informações será um SharedSettings de onde é carregado um perfil quando é iniciado o jogo, na ausência de um perfil é de imediato pedido ao utilizador para criar um.

## Armazenamento de Dados

Vai ser usado o armazenamento de dados do dispositivo para ser guardado um ficheiro \*.dat que contem o histórico de jogos do utilizador. Este ficheiro é criado assim que um utilizador termina o seu primeiro jogo e continua a guardar os seguintes jogos que forem efectuados mantendo também o histórico de jogadas realizadas.

# Comunicação

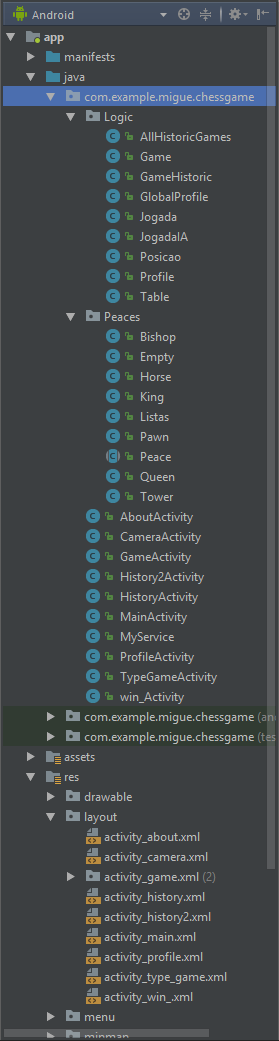
A comunicação implementada neste trabalho baseia-se em peer-to-peer utilizando sockets TCP e as suas funcionalidades. Esta foi a solução encontrada para poder jogar em rede, sendo um dos utilizadores o servidor/host inicializado uma escuta de um *serversocket* para o outro se conectar utilizando o IP fornecido na janela do host. Após a ligação feita com sucesso é inicializado o jogo, caso haja uma quebra de ligação e o socket seja fechado por uma das partes é dada a opção de escolha ao utilizador para continuar em modo de jogo a solo ou multijogador no mesmo dispositivo.

# Suporte a diferentes línguas e ecrãs

É possível utilizar duas línguas na aplicação, português e inglês, com recurso aos *ficheiros.xlm* de língua que são construídos no IDE. A utilização das línguas depende da língua definida no dispositivo.

O suporte a rotação de ecrã existe apenas dentro do jogo, podendo o utilizador jogar em “landscape” ou “portrait”.

# Organização



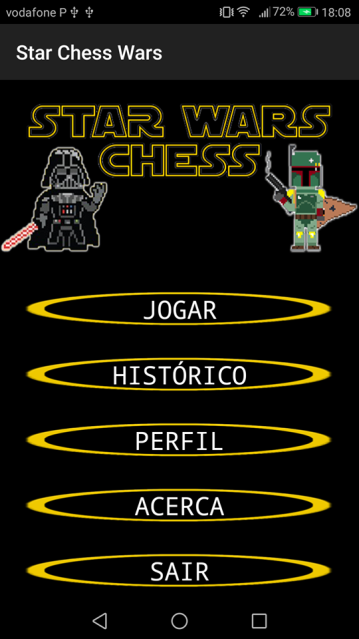
Na pasta primária, ou raiz, encontramos as actividades correspondestes as vistas da aplicação assim como o serviço que é utilizado para comunicações em multi-jogador.

Dentro da pasta Logic encontra-se toda a lógica de jogo, perfil e histórico de jogos. Na pasta Peaces estão as classes relativas a cada uma das peças possíveis de existirem no jogo assim como a classe listas onde são guardadas as peças ainda existentes num jogo para facilitar o acesso a elas.

## MainActivity

Nesta actividade foram programadas as acções de cada botão existente na vista correspondente com base nos *onClick* inicializando uma nova intenção para reencaminhar para as vistas desejadas. Existe também uma verificação da existência de um perfil de utilizador criado, e caso não exista é reencaminhado para a criação de um novo.

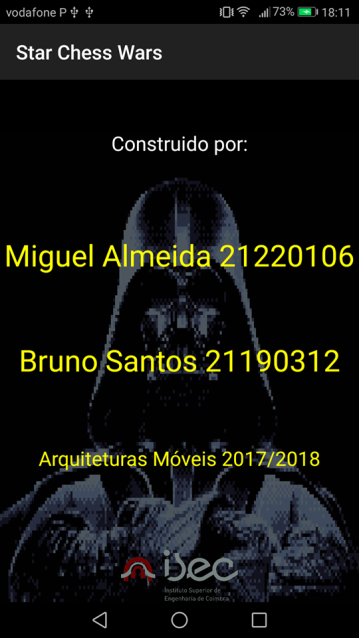
Estruturalmente a vista correspondente tem uma imagem no cabeçalho e de seguida cinco botões com funcionalidades declaradas programaticamente.



## AboutActivity

Nesta actividade não é declarado nada programaticamente a não ser a reprodução de uma musica proveniente de um ficheiro .mp3, sendo um layout completamente estático onde se encontra apenas a informação declarada no ficheiro xml.

A estrutura consiste numa imagem de fundo com campos de texto sobrepostos e o símbolo do ISEC no final do ecrã. A reprodução da música é inicializada assim que a actividade é iniciada e encerrada quando se sai.

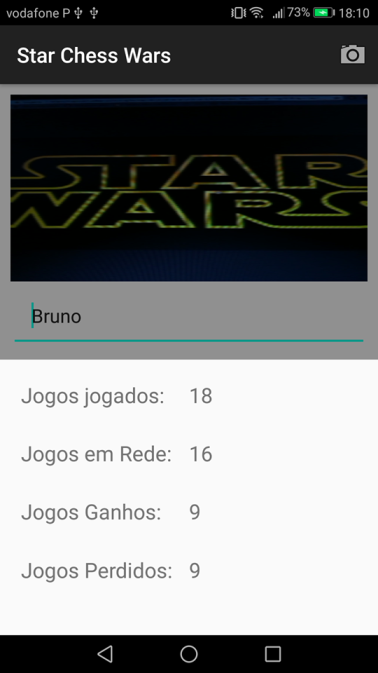


## ProfileActivity

Actividade onde é mostrado o perfil do utilizador, esta actividade tem a particularidade de ser arrancada automaticamente no inicio da aplicação se não houver um perfil definido nas SharedSettings.

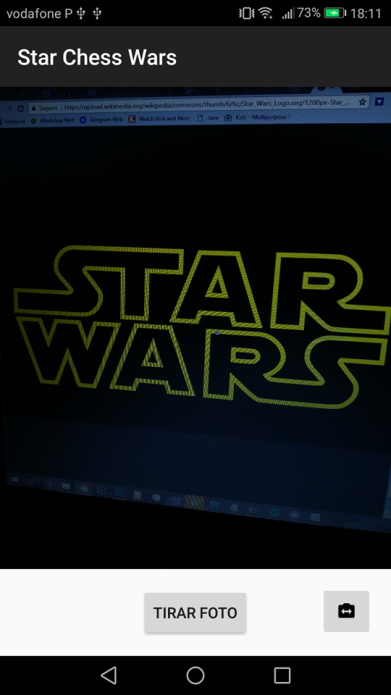
Contem um campo onde mostra a imagem associada, um campo para poder alterar o nome do perfil e um pequeno resumo das prestações do jogador com a apresentação da quantidade de jogos jogados, quantos em rede e o numero de jogos ganhos e perdidos.

Na *navigationbar* existe também o botão para a actividade a abordar no ponto seguinte.



## CameraActivity

Esta actividade surge quando se pretende associar uma imagem ao perfil de utilizador. Programaticamente são definidos cada um dos componentes a fim de captar a imagem proveniente da câmara do dispositivo, existindo um botão para tirar a foto e outro para mudar a câmara em uso. Estruturalmente é muito básico, tendo um campo para apresentação da imagem da camara e em baixo os dois botões.



## GameActivity

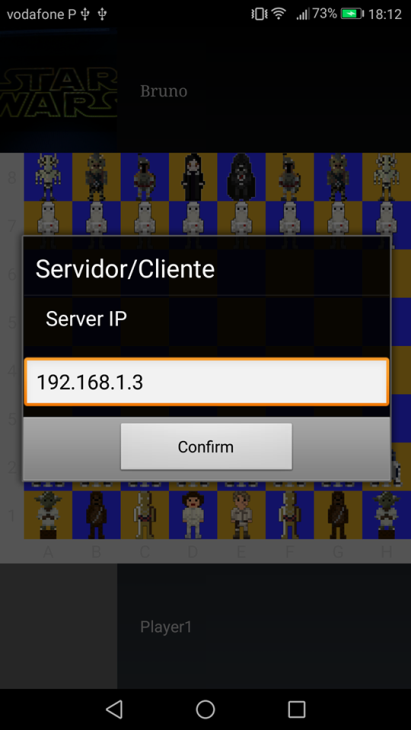
Esta é a maior actividade de toda a aplicação, contendo o tabuleiro, feito de botões, com todas as peças e os espaços destinados a apresentação da imagem e do nome do utilizador.

Programaticamente é feita toda a construção do jogo e chamado o serviço consoante o modo de jogo escolhido anteriormente. Pegando no modo de jogo são inicializadas todas as variáveis referentes a lógica assim como todas as acções de todos os botões que constituem o tabuleiro.

Sendo o modo de multi-jogador em rede é também mostrada a mensagem correspondente ao servidor/cliente antes de inicializar o jogo.

As funcionalidades de guardar o histórico do jogo no ficheiro assim como todos os métodos respeitantes a contagem do tempo caso o utilizador escolha esse modo são declaradas aqui.

Comunica com o serviço caso seja inicializado um jogo em modo de rede, utiliza um BroadcastReceiver para receber dados e envia intents para despoletar ações.



## HistoryActivity

Esta actividade apresenta uma lista com todo o histórico de jogos indicando a data e hora e o resultado. Não havendo registo de nenhum jogo a actividade termina de imediato e avisa o utilizador da inexistência de registos.

Toda a inserção de dados na lista é feita programaticamente com recurso a leitura do ficheiro e apresentação de cada um dos registos na lista, sendo este registo clicável por forma a abrir uma nova actividade com todas as jogadas do jogo clicado.

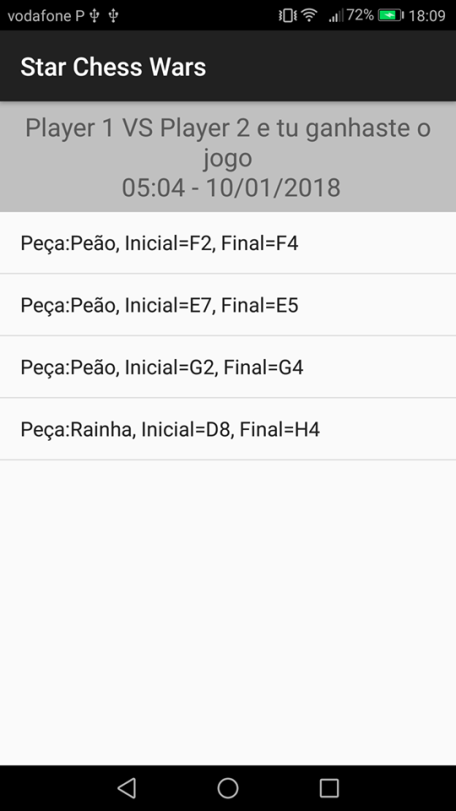
Existe também na *navigationbar* um botão para eliminar todos os registos existentes no ficheiro.



## History2Activity

Esta actividade recebe uma intenção por parte da actividade de listagem de jogos por forma a saber de que jogo se pretende mostrar todo o histórico de jogadas e posterior apresentação.

Contem um campo de texto em cima onde é mostrado o resultado do jogo e de seguida uma lista onde vão ser apresentadas cada uma das jogadas no jogo a consultar.



## TypeGameActivity

Nesta actividade existem os botões para os três diferentes modos de jogo, Solo, multi-jogador e multi-jogador em rede.

Optando por um dos modos multi-jogador vai ser mostrada uma mensagem para escolher cliente/servidor e se pretende definir um tempo para o jogo antes de iniciar a actividade de jogo.

## https://scontent.flis6-1.fna.fbcdn.net/v/t34.0-12/26855551_10210598681427968_1178059537_n.png?oh=710674a3677f769a891faa3ad599bb20&oe=5A583F5D

## Win\_Activity

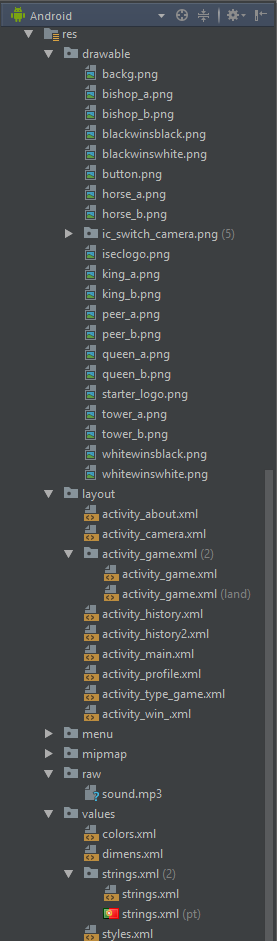
Actividade inicializada quando a condição de término de jogo é despoletada, consoante o resultado e tambem o jogador, caso esteja em multi-jogador em rede. Toda esta apresentação é feita programaticamente lendo as variáveis passadas na intent enviada.



## MyService

Este é o serviço que é inicializado quando um novo jogo em rede é começado, esta responsável por toda a comunicação de sockets, inclui as threads de comunicação, inicialização de sockets de cliente e servidor. Comunica única e exclusivamente com o GameActivity por meio de um Broadcast de *intent*.

# Organização de Recursos



Na pasta drawble estão todas as imagens relativamente as peças, logótipos e fundos presentes na aplicação. É possível encontrar também os ficheiros xml respeitantes a todas as vistas. Na pasta raw está a musica a ser reproduzida a AboutActivity. É possível ver também as duas línguas existentes na aplicação, a por defeito, inglesa e a portuguesa.

# Conclusão

Face ao proposto no enunciado houveram alguns pontos que não conseguimos executar tal como a passagem para Solo após uma falha de ligação ou a partilha de perfis num jogo em rede. Apesar de tudo consolidamos os nossos conhecimentos na programação Java para Android assim como na comunicação por sockets via threads.