



**INSTITUTO SUPERIOR  
DE TECNOLOGIAS AVANÇADAS**

**Licenciatura em informática**

**Projeto Global**

**Desenvolvimento de aplicação móvel para jogar o totobola**

Realizado por Fábio José Pinheiro Domingues

Aluno N° 2095

Coordenador: Dr. Pedro Brandão

ISTEC, Lisboa, 2016



**INSTITUTO SUPERIOR  
DE TECNOLOGIAS AVANÇADAS**

**Licenciatura em informática**

**Projeto Global**

**Desenvolvimento de aplicação móvel para jogar o totobola**

Realizado por Fábio José Pinheiro Domingues

Aluno Nº 2095

Coordenador: Dr. Pedro Brandão

ISTEC, Lisboa, 2016

**O autor é o único responsável pelas ideias expressas neste relatório**

## Resumo

O objetivo principal deste trabalho é o estudo e desenvolvimento de uma aplicação móvel para jogar o totobola.

O totobola é um jogo de apostas desportivas baseado em prognósticos para resultados de jogos de futebol. São 13 jogos para os quais se procura acertar num dos resultados possíveis: vitória, empate ou derrota da equipa visitada e um 14.º jogo, denominado "Super 14", no qual se prognosticam os golos marcados por cada uma das duas equipas: zero "0", um "1" ou mais golos "M".

As aplicações moveis para gerir as apostas nos jogos sociais do estado, onde se inclui o totobola, assentam as suas funcionalidades, essencialmente, na geração de chaves aleatórias e na capacidade dos dispositivos moveis acederem á internet para consulta de resultados, chave sorteada e ocorrência de prémios. A aplicação oficial, a distribuída pelo departamento de jogos da Santa Casa da Misericórdia de Lisboa (SCML), é a única que permite registar as apostas (jogar), se o apostador estiver registado e tiver conta com saldo suficiente para cobrir o custo das apostas. O totobola não consta, ainda, nestas aplicações incluindo a oficial que, por enquanto, só permite jogar no euromilhões, totoloto e joker.

A aplicação desenvolvida no âmbito deste trabalho, apresenta, entre outras, a funcionalidade de fazer desdobramentos de apostas para o totobola em conformidade com o sistema reduzido escolhido do conjunto de sistemas que a aplicação disponibiliza, e, como inovações! A transferência direta das apostas para outra aplicação, instalada num outro dispositivo eletrónico, fazendo uso da tecnologia de comunicação Bluetooth, e a possibilidade de codificar as apostas utilizando o código matricial *QR code*, para serem lidas por um leitor de códigos ou por uma aplicação leitora de códigos. A existência destas funcionalidades, Bluetooth e capacidade de leitura de *QR codes*, nos terminais de apostas dispostos nos mediadores dos jogos da SCML, permitiria ao apostador usar esta aplicação no seu *smartphone*, em vez do boletim em papel, para jogar, e manter o anonimato, evitando o registo, e dispensando a abertura e manutenção de conta, requisitos que a aplicação oficial exige.

## **Abstract**

The main objective of this work is the development of a mobile application to play the Portuguese football pools.

The Portuguese football pools is a game of sports betting based on forecasts for the results of football matches. There are 13 games for which demand hit one of the possible outcomes: win, draw or defeat the home team and a 14 game called "Super 14", in which the bettors predict the goals scored by each of the two teams: zero "0", "1" or more goals "M".

The features of mobile applications to manage the bets of Portuguese social games, are based, essentially, on the generation of random keys and on the ability of mobile devices to access the Internet to query results, key drawn and occurrence of premiums. The official application, distributed by the gaming department of Santa Casa da Misericórdia de Lisboa (SCML), is the one that lets you bets (play), if the bettor is registered and has sufficient cash balance to cover the cost of Bets. The football pools are not included in these app's, even in the official application that, for now, only allows us to play the Euromillions, lottery and joker.

The developed application have, among others, the functionality to do bets wheeling for football pools in accordance with the reduced system selected from systems which are provided by the application, and as innovations! The direct transfer of bets for another application installed on another electronic device, using the Bluetooth communication technology, and the possibility of encoding bets using the matrix code QR code to be read by a code reader or by a code reader application. The existence of these features in terminals bets would allow the bettors to use this application on their phone, instead of scoresheet, to play, and maintain anonymity, avoiding the registration and the account maintenance, that the official application requires.

## Índice

1. Introdução .....	1
2. Estado da arte .....	2
2.1. O Totobola .....	3
2.1.1. Metodologia do jogo .....	3
2.2. Tecnologia móvel – <i>Smartphones</i> .....	4
2.3. Tecnologias para transferência de dados entre dispositivos moveis .....	5
2.3.1. Identificação Automática e Captura de Dados (IACD) .....	5
2.3.2. Tecnologia de infravermelhos .....	6
2.3.3. Bluetooth .....	6
2.3.4. NFC ( <i>Near Field Communication</i> ) .....	8
2.4. Plataformas de desenvolvimento para dispositivos moveis .....	9
2.4.1. IOS da Apple .....	10
2.4.2. Android da Google .....	11
2.4.3. Windows Phone .....	11
2.4.4. Comparação entre sistemas operativos para dispositivos moveis ---	12
2.5. Aplicações moveis para gestão de apostas em jogos de futebol .....	13
3. Desenvolvimento da aplicação .....	14
3.1. Descrição da aplicação .....	15
3.2. Ambiente de desenvolvimento .....	21
3.3. Estrutura e Layout .....	21
4. Considerações sobre as apostas online e a aplicação .....	30
5. Conclusões .....	30
Bibliografia .....	32

## Índice de figuras

Figura 1 - Boletim do totobola-----	3
Figura 2 – Padrão de deteção do QR Code-----	5
Figura 3 – Pagamento por NFC -----	9
Figura 4 – Início da aplicação -----	16
Figura 5 – carregamento da tabela dos jogos -----	16
Figura 6 – Marcação de prognósticos simples -----	17
Figura 7 – Escolha do sistema de desdobramento -----	17
Figura 8 – Selecionar os jogos onde aplicar as triplas e as duplas -----	18
Figura 9 – Desdobramento feito -----	18
Figura 10 – comandos da aplicação -----	19
Figura 11 – Comandos da tela de registo das apostas-----	20
Figura 12 – IDE Android Studio -----	21
Figura 13 – Estrutura da aplicação -----	22
Figura 14 – ficheiro manifest.xml-----	23
Figura 15 - Exemplo de código xml de uma tablerow com uma edittext e inserida numa tablelayout-----	23

## Índice de gráficos

Gráfico 1 - Sistemas operativos para dispositivos moveis – quotas de mercado --	10
---	----

## Índice de tabelas

Tabela1- Atributos de qualidade para avaliação dos sistemas operativos -----	12
Tabela 2 – Resultado das avaliações -----	13
Tabela 3 – Legenda das avaliações -----	13
Tabela 4 – Existência das funcionalidades nas outras aplicações -----	14

## Lista de abreviaturas, siglas e acrónimos

<b>Apis</b>	<i>Application Programming Interface</i>
<b>ART</b>	<i>Android Runtime</i>
<b>EDR</b>	<i>Enhanced Data Rate</i>
<b>FH-CDMA</b>	<i>Frequency Hopping - Code-Division Multiple Access</i>
<b>GPS</b>	<i>Global Positioning System</i>
<b>HS</b>	<i>High Speed</i>
<b>IACD</b>	<i>Identificação Automática e Captura de Dados</i>
<b>IBAN</b>	<i>International Bank Account Number</i>
<b>IrDA</b>	<i>Infrared Data Association</i>
<b>IDE</b>	<i>Integrated Development Environment</i>
<b>ISM</b>	<i>Industrial, Scientific, Medical</i>
<b>JDK</b>	<i>Java Development Kit</i>
<b>NFC</b>	<i>Near Field Communication</i>
<b>QR Code</b>	<i>Quick response Code</i>
<b>RSSI</b>	<i>Received Signal Strength Indication</i>
<b>SCML</b>	<i>Santa Casa da Misericórdia de Lisboa</i>
<b>SDK</b>	<i>Software development kit</i>
<b>TDD</b>	<i>Time Division Duplexing</i>
<b>UUID</b>	<i>Universally unique identifier</i>
<b>VM</b>	<i>Virtual Machine</i>
<b>WI-FI</b>	<i>A local area network that uses high frequency radio signals to transmit and receive data over distances of a few hundred feet; uses Ethernet protocol</i>
<b>WLAN</b>	<i>Wireless local area network</i>
<b>XML</b>	<i>Extensible Markup Language</i>
<b>XNA</b>	<i>Framework que serve para o desenvolvimento de jogos para dispositivos com o Sistema operativo Windows.</i>



## 1. Introdução

A pesquisa, no âmbito deste projeto, revelou que as aplicações para dispositivos moveis que tratam os denominados jogos sociais do estado (Euromilhões, totoloto, lotaria, etc.) assentam as suas funcionalidades essencialmente na capacidade dos dispositivos moveis acedermem á internet para se tornarem usáveis. Sendo a aplicação oficial, a distribuída pelo departamento de jogos da Santa Casa da Misericórdia de Lisboa (SCML), a única que permite registar as apostas (jogar), mas, em contrapartida, obriga o apostador a registar-se e a ter conta com saldo monetário suficiente para cobrir o custo das apostas. De salientar que as aplicações encontradas, mesmo a oficial, não permitem, ainda, apostar no totobola.

Os modernos telemóveis, a par da capacidade de se ligarem a redes de dados e internet, dispõem também de tecnologias que os permitem ligar-se entre si, e com outros dispositivos eletrónicos, para comunicar e transferir dados tais como: *infrared, bluettoth, NFC etc.*

Usar estas tecnologias numa aplicação móvel para jogar os jogos da SCML, pressupondo também a sua existência nos terminais de apostas, permitiria ao apostador usar o seu *smartphone*, em vez do boletim em papel, para transferir as apostas para os terminais de jogos existentes junto aos mediadores dos jogos da SCML, mantendo o anonimato e dispensando a abertura e manutenção de conta que o método de jogar *online* requiere.

Apresentada a ideia, para desenvolver uma aplicação móvel para jogar o totobola, de forma informal a um grupo de apostadores, foi muito bem aceite esta forma de apostar que evita o registo e abertura de conta, mas a aplicação teria de satisfazer outros requisitos para se tornar efetivamente interessante: gerar chaves aleatórias e fazer desdobramentos de apostas mediante um sistema reduzido escolhido.

O propósito deste trabalho é desenvolver a aplicação idealizada para jogar o totobola tendo a preocupação de gerar a máxima satisfação no futuro utilizador.

Este documento está estruturado em 4 capítulos, mais um de conclusões. O 2º aborda as tecnologias de comunicação e transferência de dados entre dispositivos moveis, os principais sistemas operativos, outras questões relacionadas com a realização do projeto e que ajudam a compreende-lo e o que existe semelhante ao trabalho realizado. No capítulo 3 são

abordados alguns aspetos metodológicos e o desenvolvimento da aplicação descrevendo as funcionalidades da mesma. No 4º são levantadas algumas questões que, provavelmente, tornam pouco atrativa, para o explorador do jogo, esta ideia de aproveitar as tecnologias de comunicação e transferência de dados direta entre dispositivos. Por último as conclusões.

## 2. Estado da Arte

O totobola joga-se utilizando os boletins editados pelo departamento de jogos da Santa Casa da Misericórdia de Lisboa (SCML), distribuídos nos postos dos agentes oficiais, ou apostando *online* no portal dos Jogos da Santa Casa<sup>1</sup>. Se o requisito para a primeira opção é o uso de boletim em papel adquirido junto do mediador de apostas, a segunda implica a perda de privacidade do apostador ao ter que se registar e abrir conta expondo os seus dados pessoais, entre eles o *International Bank Account Number (IBAN)* (SCML© 2012a).

Os dados pessoais têm de ser certificados enviando cópia do documento de identificação para o portal do departamento de jogos da SCML.

Pretende-se criar uma alternativa às formas de jogar no totobola que não obrigue ao registo de dados do apostador no portal de jogos da SCML e que dispense a utilização de boletins de papel, desenvolvendo para o efeito uma aplicação móvel, para *smartphones*, que para além de responder às necessidades dos apostadores mais exigentes (gerar chaves aleatórias, aplicação de desdobramentos para 2º prémio, verificar a ocorrência de prémios, etc.) utilize uma das tecnologias de comunicação e transferência de dados entre dispositivos eletrónicos pressupondo a existência destas tecnologias num terminal de registo de apostas desportivas mais evoluído.

---

<sup>1</sup> A SCML lançou recentemente uma “App oficial dos jogos Santa Casa” para dispositivos moveis com sistemas operativos Ios e Android, mas não contempla o totobola e implica na mesma registo no portal e abertura de conta.

## 2.1. O Totobola

O totobola faz parte do grupo de jogos sociais explorados pela SCML com o aval do estado. O primeiro concurso teve lugar a 24 de Setembro de 1961.

É um jogo de apostas desportivas baseado em prognósticos para resultados de jogos de futebol. São 13 jogos para os quais se tenta acertar no resultado na forma de vitória, empate ou derrota da equipa visitada, considera-se equipa visitada a indicada em primeiro lugar, e um 14.º jogo, denominado "Super 14", no qual se prognosticam os golos marcados por cada uma das duas equipas: zero "0", um "1" ou mais golos "M" (SCML© 2012b).

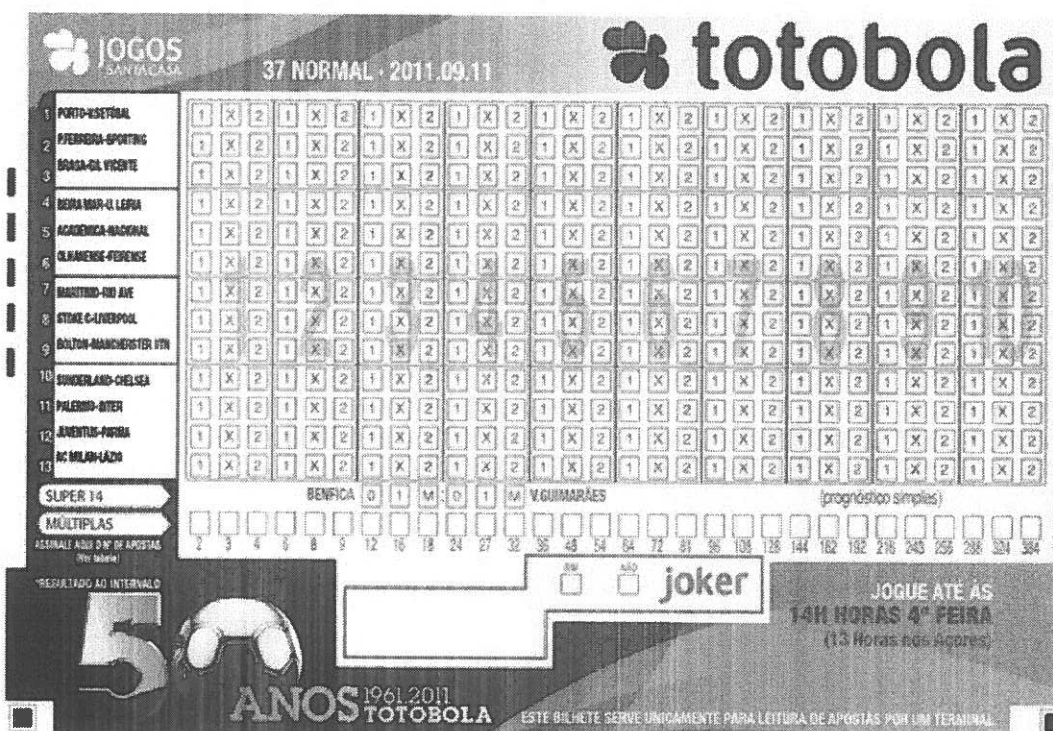


Figura 1 - Boletim do totobola

### 2.1.1. Metodologia do jogo

Para apostar no previsível desfecho dos jogos de futebol, compostos maioritariamente por desafios dos campeonatos nacionais, utilizam-se grupos de 3 colunas, que cruzam as linhas dos jogos, e marca-se uma delas com X. Significando a presença do símbolo na coluna 1 apostar na vitória (1) da equipa visitada, na coluna 2 no empate (X) e na 3ª e ultima coluna do grupo na derrota (2). Isto nas apostas simples que são obrigatoriamente agrupadas aos

pares. Pode-se também apostar na forma de múltiplas, respeitando as combinações admitidas (duplas e triplas) que constam na tabela do boletim, atribuindo aos jogos mais que um prognóstico e inscrevendo-os obrigatoriamente nas colunas do primeiro grupo, exemplo: marcar as colunas 1, 2 e 3 significa apostar uma tripla (vitória, empate e derrota), marcar as colunas 1 e 3 significa apostar uma dupla (vitória e derrota).

## 2.2. Tecnologia móvel – *Smartphones*

Tecnologia móvel é como o nome indica: uma tecnologia portátil. É um termo com o qual referenciamos um dispositivo capaz de executar uma ampla variedade de tarefas e que facilmente podemos levar connosco. É a tecnologia que transforma o simples telefone móvel para fazer chamadas e escrever mensagens de texto num equipamento altamente evoluído, o designado *smartphone* com inúmeras funcionalidades graças a um processador e a um sistema operativo que praticamente o tornam num mini computador.

O conceito de *smartphone* não é imutável! O que hoje o caracteriza já não o será amanhã face à evolução tecnológica que o tempo comanda. Contudo um *smartphone* será sempre um equipamento recente e topo de gama em termos de preço e funcionalidades.

Para (Firtman, 2013), um *smartphone* possui um sistema operativo multitarefa, um *browser* com as mesmas capacidades e funcionalidades dos *browsers* usados pelos computadores, ligação sem fios (*WLAN*, também conhecida por *WIFI*), conectividade 3G/4G, uma aplicação para reproduzir música e as seguintes características/funcionalidades:

- *GPS* (Sistema de posicionamento global) ou *A-GPS* (*GPS* assistido)
- Bússola digital
- Câmara de vídeo
- Saída de vídeo
- Comunicação Bluetooth
- Ecrã táctil
- Aceleração de vídeo com capacidade 3D
- Acelerómetro
- Giroscópio
- Magnetómetro

## 2.3. Tecnologias de comunicação e transferência de dados entre dispositivos moveis

### 2.3.1. Identificação Automática e Captura de Dados (IACD)

É uma tecnologia para obter informação e a transmitir de forma automática e direta, sem envolvimento humano, a sistemas computadorizados. Incluem-se nas IACD os código de barras bidimensionais *quick response (QR code)*.

#### QR Code

O código QR é um código de alta velocidade do tipo matricial, armazena informação em ambas as direções vertical e horizontal. O código é lido através do padrão de deteção de posição organizado em três cantos e a partir de qualquer direção, conforme mostra a figura 2. (Law, C. & So, S. 2010)



Figura 2 – Padrão de deteção do QR Code<sup>2</sup>

Um código QR tem mais capacidade do que os outros códigos bidimensionais, pode armazenar até 7.089 caracteres numéricos ou 4.296 caracteres caso sejam alfanuméricos. Ao contrário do código de barras tradicional, o QR não requer um leitor especial, a informação pode ser lida por um leitor de códigos de barras ou por uma aplicação leitora de códigos (*qr code reader app*) instalada num *smartphone*, ou num outro dispositivo móvel, usando a câmara fotográfica do mesmo (Vermaat, M, *et al.*, 2016).

<sup>2</sup> Fonte: Law, C. & So, S., 2010.

### **2.3.2. Tecnologia de infravermelhos**

*Infrared* - comunicação sem fios, padronizada pela *Infrared Data Association (IrDA)*, que utiliza impulsos de luz invisíveis ao olho humano. O dispositivo emissor e o recetor têm de estar muito próximos e as portas de infravermelhos de ambos frente a frente sem obstáculos entre si. Embora não precisando de meio físico para transmissão de dados é uma tecnologia em desuso nos telemóveis, é mais utilizada em controlos remotos, por ter baixas velocidades de transmissão, pouco alcance e necessitar que os dispositivos estejam dentro do campo de visão um do outro.

### **2.3.3. Bluetooth**

É uma tecnologia de comunicação sem fios, com baixo consumo de energia, que permite que dispositivos próximos se interliguem por ondas rádio de pequeno alcance e relativamente ao qual a podemos dividir em três classes:

Classe 1: alcance até 100 metros, com uma potência máxima de 100 mW (mili-watt);

Classe 2: alcance até 10 metros, com uma potência máxima de 2,5 mW;

Classe 3: alcance até 1 metro, com uma potência máxima de 1 mW.

Numa comunicação entre dispositivos de classes diferentes impera o limite daquele que possui o alcance menor (INFOWESTER, 2014).

#### **Versões do Bluetooth**

O Bluetooth é uma tecnologia em permanente evolução e está presente praticamente em todos os telemóveis modernos. Novas versões irão surgir com o passar do tempo, mas até ao momento são estas as disponíveis:

#### **Bluetooth 1.0**

A versão 1.0 e a 1.0B são as primeiras versões oficiais do Bluetooth com velocidades de transmissão de 721 Kb/s. Como a tecnologia era recente os fabricantes enfrentaram problemas de implementação e de interoperabilidade entre dispositivos.

### **Bluetooth 1.1**

Surgiu em fevereiro de 2001, adotou o padrão de comunicações IEEE 802.15 e incluiu o suporte *Received Signal Strength Indication (RSSI)*, sistema que mede a potência do sinal recebido.

### **Bluetooth 1.2**

Foi lançada em novembro de 2003. A versão 1.2 do Bluetooth tem como melhorias a proteção contra interferências, a velocidade de transmissão que passa a poder atingir 1Mbps e a qualidade sonora nas ligações áudio.

### **Bluetooth 2.0**

O seu lançamento oficial foi em novembro de 2004 e apresentou como principais novidades a diminuição do consumo de energia, melhorias no desempenho e aumento na velocidade de transmissão para até 3Mb/s graças ao padrão *Enhanced Data Rate (EDR)*.

### **Bluetooth 2.1**

Foi lançada em agosto de 2007, Apresentou como melhorias a segurança e informação no processo de ligação e a gestão do consumo de energia.

### **Bluetooth 3.0**

Versão lançada em abril de 2009, tem como novidade mais expressiva a velocidade de transferência de dados que pode chegar aos 24 Mb/s entre dispositivos compatíveis com as instruções *High Speed (HS)* e graças á utilização do padrão de comunicações 802.11.

### **Bluetooth 4.0**

Surgiu em dezembro de 2009 e trouxe vantagens ao nível da poupança de energia, quando o dispositivo tem o Bluetooth ativo mas não o está a utilizar o consumo de energia é baixíssimo. Esta característica tem impacto nos dispositivos moveis pois permite uma maior autonomia da bateria.

Se dois dispositivos tiverem versões diferentes de Bluetooth a velocidade da transmissão de dados será limitada à taxa suportada pela versão mais baixa.

### **Funcionamento do bluetooth**

Os dispositivos Bluetooth comunicam por ondas radio cuja frequência é abrangida pela faixa *Industrial, Scientific, Medical (ISM)*, que opera a uma frequência de 2,45 GHz com variações compreendidas entre 2,4 GHz e 2,5 GHz. Trata-se de uma faixa aberta e disponível para usar em qualquer sistema de comunicação e, como tal, as transmissões nesta frequência ficam sujeitas a uma maior interferência no sinal. Para se proteger desta interferência a tecnologia bluetooth utiliza o esquema de comunicações *Frequency Hopping - Code-Division Multiple Access (FH-CDMA)* dividindo a frequência em vários canais. O dispositivo que estabelece a ligação muda rapidamente de canal para evitar a interferência, se necessário. No Bluetooth podem-se utilizar até 79 frequências dentro da faixa *ISM*, espaçadas de 1 MHz. Uma das características da tecnologia Bluetooth é a capacidade de transmissão de dados bidirecional (modo *full-duplex*), utilizando uma técnica de múltiplo acesso por divisão de tempo, *Time Division Duplexing (TDD)* (INFOWESTER, 2014).

### **Redes Bluetooth**

Ao estabelecermos ligação entre dois ou mais dispositivos Bluetooth formamos uma rede denominada *piconet*. Numa *piconet* há um dispositivo que assume o papel de *master* (mestre) disponibilizando a ligação e regulando a transmissão de dados. Os outros dispositivos na rede são denominados *slaves* (escravos). Cada *piconet* pode suportar até 8 dispositivos (um *master* e 7 *slaves*). Podemos interligar *piconets* dentro do alcance dos dispositivos Bluetooth e formar uma rede maior denominada *scatternet*. Um dispositivo *slave* pode pertencer a mais do que uma *piconet* ao mesmo tempo, no entanto, um *master* só pode ocupar essa posição em uma única *piconet* dentro da *scatternet* (INFOWESTER, 2014).

#### **2.3.4. NFC (Near Field Communication)**

É uma nova tecnologia de comunicação sem fio (Wireless). Que liga dois dispositivos automaticamente por ondas rádio (se o *NFC* dos dispositivos estiverem ligados) mediante uma aproximação entre eles inferiores a 10cm. A curta distância é propositada para evidenciar a intenção de estabelecer a comunicação. Os mais recentes *smartphones* vêm equipados com este recurso. Para partilhar informação através desta tecnologia tem de se ativar a opção *NFC* nas definições do *smartphone* antes de seleccionar o conteúdo a transferir (por exemplo



uma imagem), depois aproximar as traseiras dos telemóveis envolvidos na transferência de informação e tocar no ecrã do *smartphone* emissor quando a miniatura do conteúdo a partilhar aparecer.

Há mais possibilidades de utilização desta tecnologia como por exemplo: Efetuar pagamentos automáticos em terminais para pagamentos via *NFC* utilizando o *smartphone* como exemplifica a figura 3, fazer a leitura de informações em pequenas etiquetas (*NFC tags*) incorporadas em produtos, em sinais, em caixas, ou em outros lugares como locais públicos, Museus, pontos de verificação etc. Uma *NFC tag* pode armazenar até 132 caracteres. (Firtman, 2013)

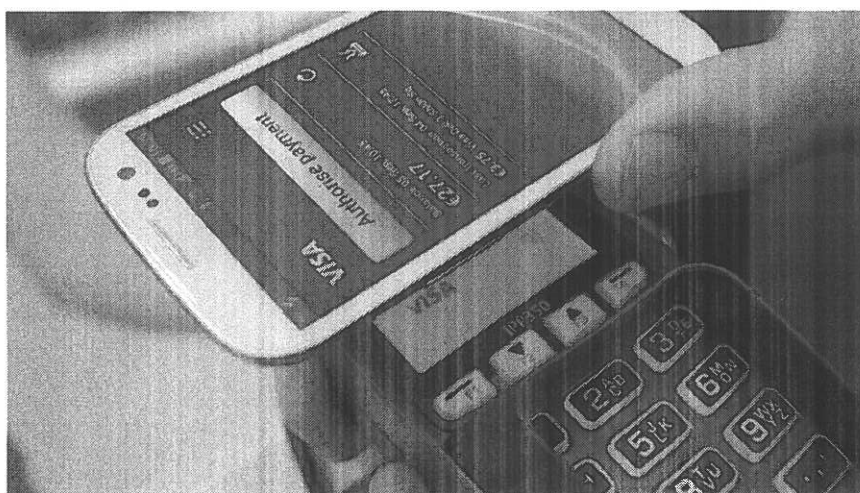


Figura 3 – Pagamento por NFC<sup>3</sup>

A tecnologia NFC trabalha na frequência de 13.56 MHz e a comunicação é *Half-Duplex* com velocidade de transmissão até 424 Kbps.

#### 2.4. Plataformas de desenvolvimento para dispositivos moveis

Tendo por base a atual quota de mercado, são 3 os principais sistemas operativos para dispositivos moveis: Android da Google, iOS da Apple e Windows Phone da Microsoft.

---

<sup>3</sup> Fonte: PPLWARE, 2014.

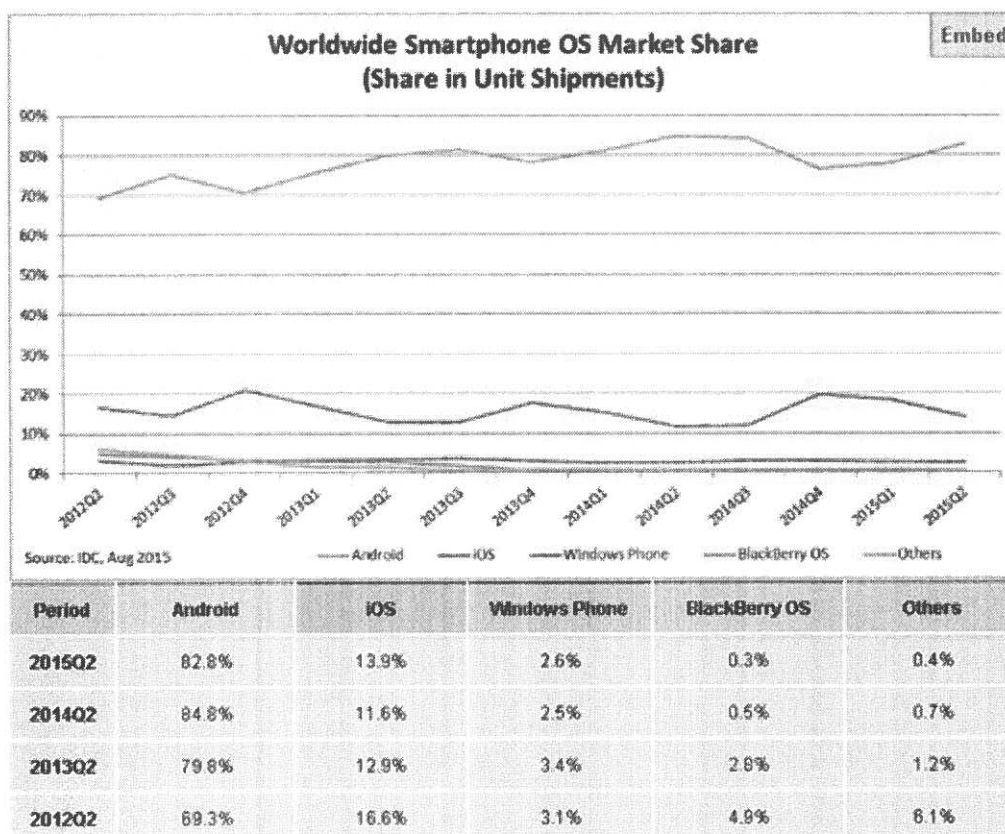


Gráfico 1 - Sistemas operativos para dispositivos moveis – quotas de mercado<sup>4</sup>

### 2.4.1. IOS da Apple

IOS é o sistema operativo do *iPhone* da Apple e é baseado nos Mac OS X. Só existem duas formas oficiais para o desenvolvimento de aplicações para o *iPhone*: utilizando tecnologias web móveis ou utilizando a plataforma nativa Cocoa Touch, construída em Objective-C. Existem contudo outras plataformas não-oficiais que também podem compilar aplicações nativas IOS, tais como a Adobe AIR, Corona ou a Appcelerator Titanium. Os dispositivos iOS já vêm com acesso à App Store, a loja da Apple para venda e distribuição de aplicações para o *iPhone*, *iPod touch* e *iPad*. (Firtman, 2013)

O desenvolvimento de aplicações para o *iPhone* é feito em computadores Macintosh com sistemas operativos Mac OS X *Snow Leopard* e processadores da Intel. A ferramenta de desenvolvimento é a *SDK (Software Development Kit)*, disponível no *site* da Apple. O *Kit* inclui o Xcode IDE (ambiente de desenvolvimento integrado), um simulador de *iPhone* e outras

<sup>4</sup> Fonte IDC, 2015.

ferramentas adicionais. Os programas são escritos em linguagem Objective-C. (Allen et al., 2010)

#### **2.4.2. Android da Google**

Android é um sistema operativo pertencente à Google baseado em Linux e projetado principalmente para dispositivos móveis. A estrutura de desenvolvimento de aplicações inclui várias bibliotecas escritas em java e em C/C++ que facilitam a inclusão de capacidades/funcionalidades dos dispositivos nas aplicações a criar pelo programador, tais como serviços de localização e utilização da câmara fotográfica e de vídeo.

As aplicações para android são escritas em java, recompilados para *bytecodes Dalvik* e executadas numa máquina virtual *Dalvik* que é similar á maquina virtual do java (Java VM), mas otimizada para dispositivos com pouca memoria e capacidade de processamento. (Reynolds, M., 2014). A partir da versão 4.4 do android, a maquina virtual passou a ser a ART (*Android Runtime*), criada de raiz pela Google com melhorias na depuração de código, compilação e deteção de código inútil (sem uso).

Para desenvolver aplicações para o android é necessário o *Java SE Development Kit (JDK)* e o *SDK* do android com um dos ambientes de desenvolvimento integrado (IDE): *android studio*, *eclipse* ou *Netbeans*.

#### **2.4.3. Windows Phone**

O *Windows Phone 7* é o sistema operativo da Microsoft para dispositivos moveis que foi lançado em fevereiro de 2010 como sucessor do *Windows mobile*. Este fora introduzido no início deste milênio e a partir de 2009 deixou de ser atualizado.

As aplicações para o windows Phone são desenvolvidas utilizando as linguagens C# ou Visual Basic .NET juntamente com um dos componentes que fazem parte do ambiente de desenvolvimento: o Silverlight ou XNA. O Silverlight é sobretudo usado para desenvolver aplicações comerciais e jogos simples em duas dimensoões (2D). Faz uso do *Extensible Application Markup Language (XAML)* para criar *user interfaces (UIs)*. O XNA é usado principalmente para criação de jogos em 3D. (Lee & Chuvyrov, 2010)

No início deste ano de 2016 a Microsoft lançou a sua mais recente versão do seu sistema operativo pra dispositivos moveis: o Windows 10 mobile.

Para desenvolver aplicações para o Windows 10 mobile é conveniente obter o Visual Studio 2015 que contem o Windows 10 *SDK* e outras ferramentas de desenvolvimento ou, se for usado outro ambiente de desenvolvimento, obter de forma independente o Windows *SDK* para o Windows 10, como explicado no *site* da Microsoft.

#### 2.4.4. Comparação entre sistemas operativos pra dispositivos moveis

Com base nos atributos específicos da matriz seguinte de (Costa, N.P.O *et al.*, 2012), foi elaborada a Tabela 2, Tabela comparativa entre os três sistemas operativos para dispositivos moveis com a maior quota de mercado. (Costa, N.P.O *et al.*, 2012)

Tabela1- Atributos de qualidade para avaliação dos sistemas operativos

<b>Atributo Avaliado</b>	<b>Descrição</b>
<b>Facilidade de Uso</b>	Fácil percepção das funcionalidades e resposta prática em relação às necessidades dos usuarios.
<b>Simplicidade Interface</b>	Realização das tarefas de forma simples e prática. Aspectos amigáveis de interface gráfica.
<b>Disposição Automática</b>	Opção de organização automática dos objetos/icones dentro da área de trabalho.
<b>Redimensionamento Automático</b>	Ajuste automático da área de trabalho, em relação a posição do aparelho e do tamanho do aplicativo.
<b>Linguagens Suportadas</b>	Suportar diferentes tipos de linguagens de desenvolvimento em relação aos aplicativos.
<b>Open Source</b>	A licença não deve restringir de nenhuma maneira a venda ou distribuição do programa gratuitamente.
<b>Disponível para Múltiplos Fabricantes</b>	Permite a sua utilização por diferentes fabricantes de mercado, garantindo maior interoperabilidade.
<b>Multitarefa</b>	Permite repartir a utilização do processador entre várias tarefas aparentemente simultaneamente.
<b>Interface Multitouch</b>	Capazes de detectar e processar vários pontos de contato na sua superfície
<b>Navegador Web</b>	Capazes de lidar e suportar núcleos diferentes de navegadores.
<b>Dados multimídias</b>	Permite o armazenamento, edição e exibição de arquivos multimídias (fotos, vídeos, texto, etc).
<b>Conectabilidade (Wi-Fi, Bluetooth, etc)</b>	Provê maneiras de conectar e trocar informações entre dispositivos
<b>Upgrades</b>	Forma como é feita as atualizações do sistema, via sincronização, pacotes, online, etc.
<b>Aplicações Disponíveis</b>	Número de aplicações disponíveis para utilização, vendas e downloads.
<b>Permite App's não oficiais</b>	Permite a utilização de aplicativos de terceiros, não oficiais em relação aos fabricantes do sistema operacional.
<b>Suporte Flash</b>	Suporte a software de gráfico vetorial, imagens bitmap e vídeos.

Tabela 2 – Resultado das avaliações

<b>Avaliação Comparativa dos Sistemas Operacionais Móveis</b>			
<b>Atributo Avaliado</b>	<b>iOS 4.x</b>	<b>Windows Phone 7</b>	<b>Android 2.x</b>
Facilidade de Uso	5	4	5
Simplicidade	4	4	4
Interface	4	4	5
Disposição Automática	4	5	4
Redimensionamento Automático	5	4	5
Linguagens Suportadas	4	4	4
<i>Open Source</i>	3	3	5
Disponível para Múltiplos Fabricantes	3	5	5
Multitarefa	5	4	5
<i>Interface Multitouch</i>	5	5	5
Navegador Web	4	5	4
Dados multimídias	5	5	5
Conectabilidade (Wi-Fi, Bluetooth, etc)	3	5	5
<i>Upgrades</i>	4	4	5
Aplicações Disponíveis	5	4	4
Permite App's não oficiais	3	4	5
Suporte Flash	3	4	3

Tabela 3 – Legenda

<b>1</b>	<b>Oculto</b>
<b>2</b>	<b>Ruim</b>
<b>3</b>	<b>Regular</b>
<b>4</b>	<b>Bom</b>
<b>5</b>	<b>Muito Bom</b>

## 2.5. Aplicações moveis para gestão de apostas desportivas baseadas em jogos de futebol

Resultante da pesquisa efetuada na internet nomeadamente no *Google play store* confirma-se a existência de aplicações para gerir apostas desportivas baseadas em jogos de futebol, com maior relevância para os jogos da liga Espanhola. O que se encontrou foram aplicações desenvolvidas para satisfazer comunidades de apostadores ao nível de País.

Tendo por referência os objetivos que se pretendem cumprir pela aplicação móvel a desenvolver, das aplicações observadas a *La Quiniela en vivo – Oficial*, a *QuiniPhone*, a *quinidroid* e a *Toto foot* são as que mais se assemelham ao trabalho proposto. A tabela 4 compara esses objetivos com as funcionalidades das aplicações referidas.

Tabela 4 – Existência das funcionalidades nas outras aplicações

Funcionalidades	La Quiniela en vivo – Oficial	QuiniPhone	Toto foot	Quinidroid
Descarregar tabela de jogos	Não. ( <i>O Acesso aos jogos é feito online</i> )	Não. ( <i>O Acesso aos jogos é feito online</i> )	Não. ( <i>O Acesso aos jogos é feito online</i> )	Não. ( <i>O Acesso aos jogos é feito online</i> )
Gerar prognósticos aleatoriamente	Não	Não	Não	Sim
Aplicar desdobramentos de apostas	Sim	Sim	Sim	Sim
Guardar as chaves	Sim	Sim	Sim	Sim
Verificação de acertos automaticamente mediante introdução da chave ganhadora	Não.	Não.	Não.	Não
Utilização de infravermelhos, Bluetooth, NFC ou Qr codes para transferir as chaves para outro dispositivo	Não.	Não.	Não.	Não

Não se encontrou nenhuma aplicação deste género para os apostadores do totobola Português. O que reforça a motivação para o desenvolvimento desta aplicação.

### 3. Desenvolvimento da aplicação

Idealizada a aplicação, a primeira etapa do trabalho foi a pesquisa e análise de aplicações semelhantes. A segunda fase passou pela escolha da plataforma de desenvolvimento. Avaliados os sistemas operativos para dispositivos móveis, a escolha acabou por recair na plataforma da Google, Android versão 4, tendo em conta os seguintes fatores:

- 1) O Android é o Sistema operativo com maior quota de mercado, o que garante maior distribuição da aplicação.
- 2) Permite aplicações não oficiais.
- 3) O autor possui alguns conhecimentos de java e um *smartphone* android.

A fase de conceção do projeto foi acompanhada de pesquisa bibliográfica ao nível técnico: procura em *sites*<sup>5</sup>, tutoriais e livros especializados.

### 3.1. Descrição da aplicação

A aplicação desenvolvida foi batizada com a designação de Totobola1X2. As suas funcionalidades envolvem:

- Fazer prognósticos:

Manualmente

Aleatoriamente (simples e condicionados)

- Fazer desdobramentos de apostas usando sistemas reduzidos (para 2ºprémio)
- Guardar as apostas
- Verificação de acertos mediante introdução da chave ganhadora do totobola
- Preencher manualmente a tabela dos jogos referente ao concurso ou automaticamente fazendo o descarregamento do ficheiro jogos da internet.
- Enviar as apostas para outro dispositivo, via Bluetooth ou utilizando QR codes.

Para que duas aplicações instaladas em 2 dispositivos diferentes possam usar a funcionalidade Bluetooth para comunicarem diretamente (aplicação a aplicação) têm de partilhar um código de 128 bits denominado *UUid (Universally Unique Identifier)*<sup>6</sup> para identificar o serviço.

A aplicação tem rolagem (*scroll*) da tela na vertical e horizontal, para evitar a redução dos componentes gráficos a um nível que a tornassem disfuncional, e utiliza dois ficheiros de apoio: *jogos.txt*, para guardar os jogos, e *apostas.txt*, para guardar as apostas. Quando se acede à aplicação são logo visualizadas as chaves contidas em *apostas.txt* se o ficheiro existir e se tiver conteúdo, como figura 4.

---

<sup>5</sup> Com especial relevância para <https://developer.android.com/guide/topics/connectivity/bluetooth.html>

<sup>6</sup> Existem vários sites na internet que geram UUid's. Ex: <https://www.uuidgenerator.net/>

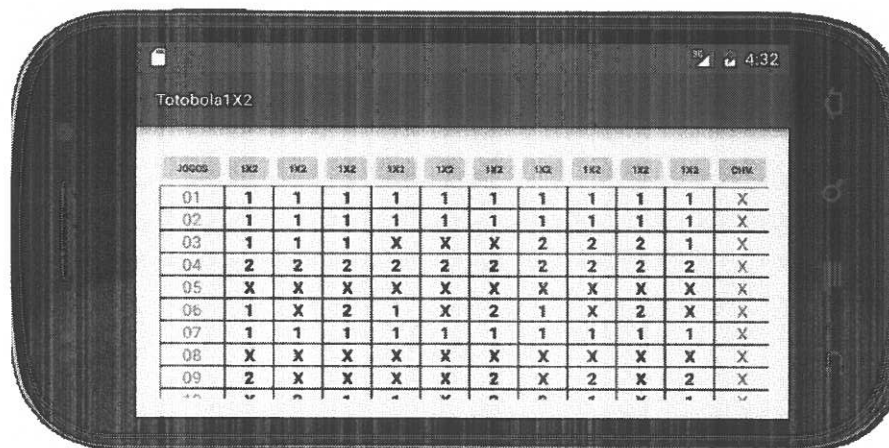


Figura 4 – Inicio

Selecionando jogos são mostradas as equipas que se confrontam nesses jogos.

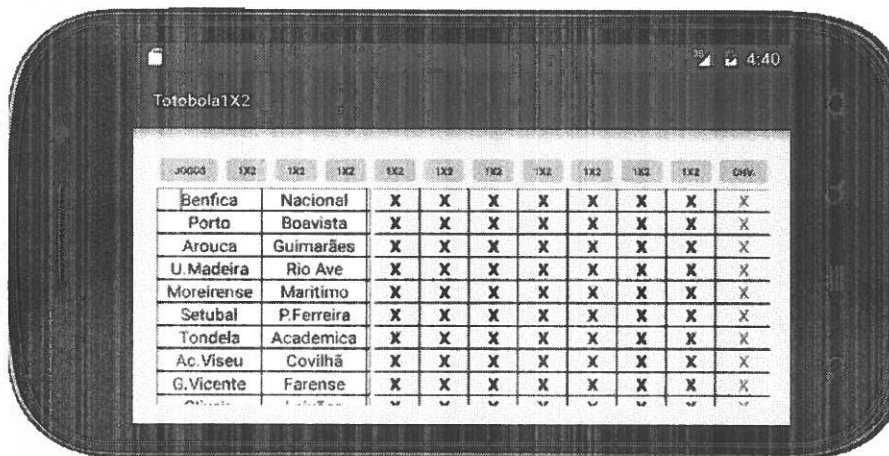


Figura 5 – carregamento da tabela dos jogos

Os jogos podem ser movidos da esquerda para a direita e vice-versa, a fim de facilitar o preenchimento das apostas. No decurso de escolha dos prognósticos, feita por toque nas células, os símbolos (1X2) são marcados a cor rosa, como figura 6. Tem de ser preenchida pelo menos uma aposta, 13 jogos marcados, para que a aplicação aceite criar o ficheiro apostas ou aceite adicionar prognósticos ao ficheiro.



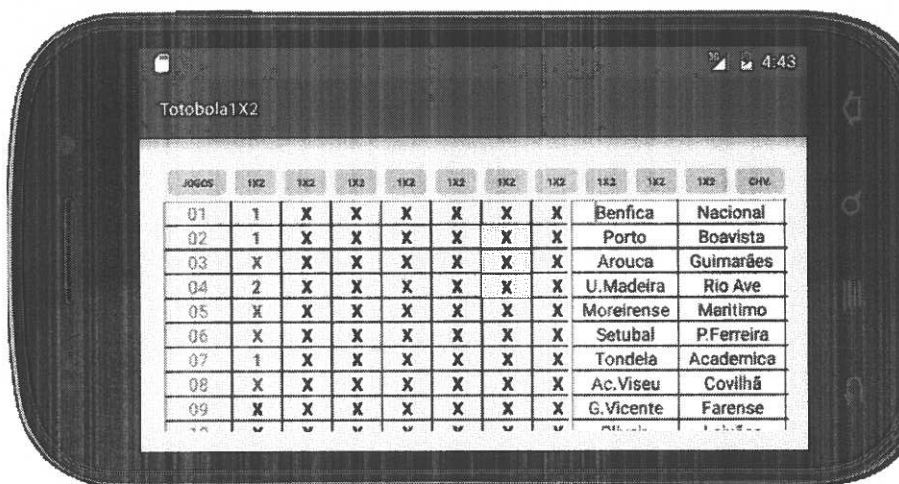


Figura 6 – Marcação de prognósticos simples

A aplicação permite acesso a um conjunto de sistemas reduzidos para efetuar desdobramentos. Estes sistemas estão alojados, assim como o ficheiro jogos.txt preenchido, em *site* do programador, sendo necessária ligação á internet para que a aplicação faça o descarregamento do sistema escolhido. O dispositivo mantém sempre, e só, o ultimo sistema de desdobramento descarregado.

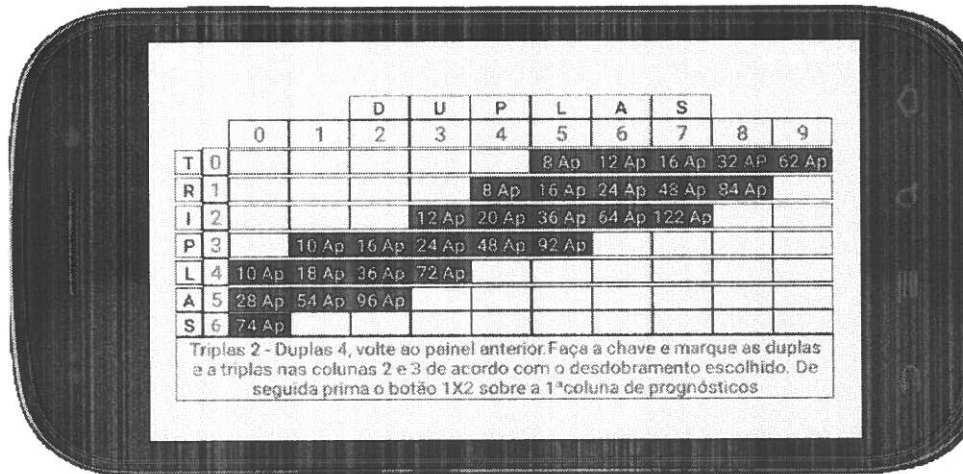


Figura 7 – Escolha do sistema de desdobramento

Depois de selecionado o sistema para desdobramento, como figura 7, volta-se ao painel inicial, escolhem-se os jogos onde aplicar as triplas e duplas, figura 8, e o desdobramento é feito, figura 9. As chaves resultantes são guardadas automaticamente no ficheiro apostas.txt. Quando as chaves são feitas de modo manual ou aleatório o utilizador tem de seleccionar a opção guardar.

Totobola1X2

JOGOS	1X2	1X2	1X2	1X2	1X2	1X2	1X2	1X2	1X2	1X2	1X2	OHV
01	1	X	X	X	X	X	X	X	X	X	X	X
02	1	X	X	X	X	X	X	X	X	X	X	X
03	X	2	X	X	X	X	X	X	X	X	X	X
04	1	X	2	X	X	X	X	X	X	X	X	X
05	1	X	X	X	X	X	X	X	X	X	X	X
06	1	X	2	X	X	X	X	X	X	X	X	X
07	1	X	X	X	X	X	X	X	X	X	X	X
08	X	2	X	X	X	X	X	X	X	X	X	X
09	1	2	X	X	X	X	X	X	X	X	X	X
10	X	X	X	X	X	X	X	X	X	X	X	X

Figura 8 – Selecionar os jogos onde aplicar as triplas e as duplas

Totobola1X2

JOGOS	1X2	1X2	1X2	1X2	1X2	1X2	1X2	1X2	1X2	1X2	1X2	OHV
01	1	1	1	1	1	1	1	1	1	1	1	X
02	1	1	1	1	1	1	1	1	1	1	1	X
03	2	X	2	2	X	X	X	X	X	X	2	X
04	1	1	1	1	1	1	X	X	X	X	X	X
05	1	X	X	1	X	1	X	1	1	X	X	X
06	1	1	X	X	2	2	1	1	X	X	X	X
07	1	1	1	1	1	1	1	1	1	1	1	X
08	2	X	2	X	X	2	2	X	2	X	X	X
09	1	1	2	2	1	2	2	2	1	1	1	X
10	X	X	X	X	X	X	X	X	X	X	X	X

Figura 9 – Desdobramento feito

As figuras 10 e 11 descrevem os comandos da aplicação.

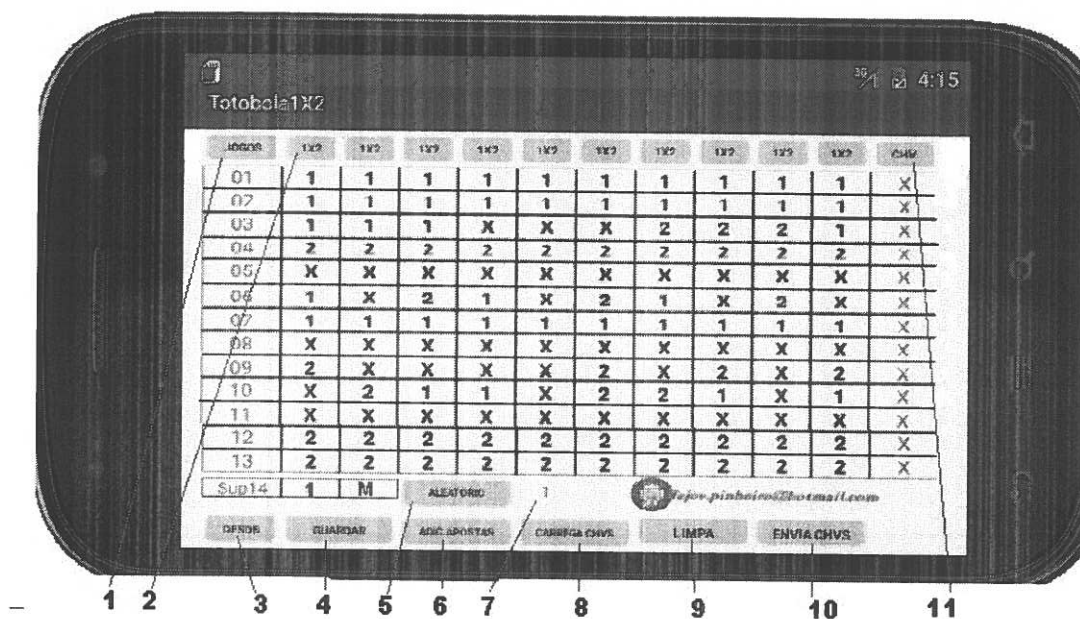


Figura 10 – comandos da aplicação

- 1 – Carrega os jogos.
- 2 – Inicia o desdobramento se houver triplas ou duplas ou triplas e duplas marcadas.
- 3 – Acesso ao painel de escolha dos desdobramentos.
- 4 – Guardar as chaves do utilizador em apostas.txt.
- 5 – Gerar chaves aleatórias em número indicado em 7.
- 6 – Adiciona chaves ao ficheiro apostas.txt.
- 7 – Indica a quantidade de chaves aleatórias para gerar.
- 8 – Visualiza as chaves do ficheiro.
- 9 – Preenche a grelha de visualização das chaves com X's.
- 10 – Passa as chaves para o painel de registo.
- 11 – Verifica a ocorrência de prémios e acertos nas chaves contidas na grelha de visualização mediante a chave introduzida na coluna CHV.

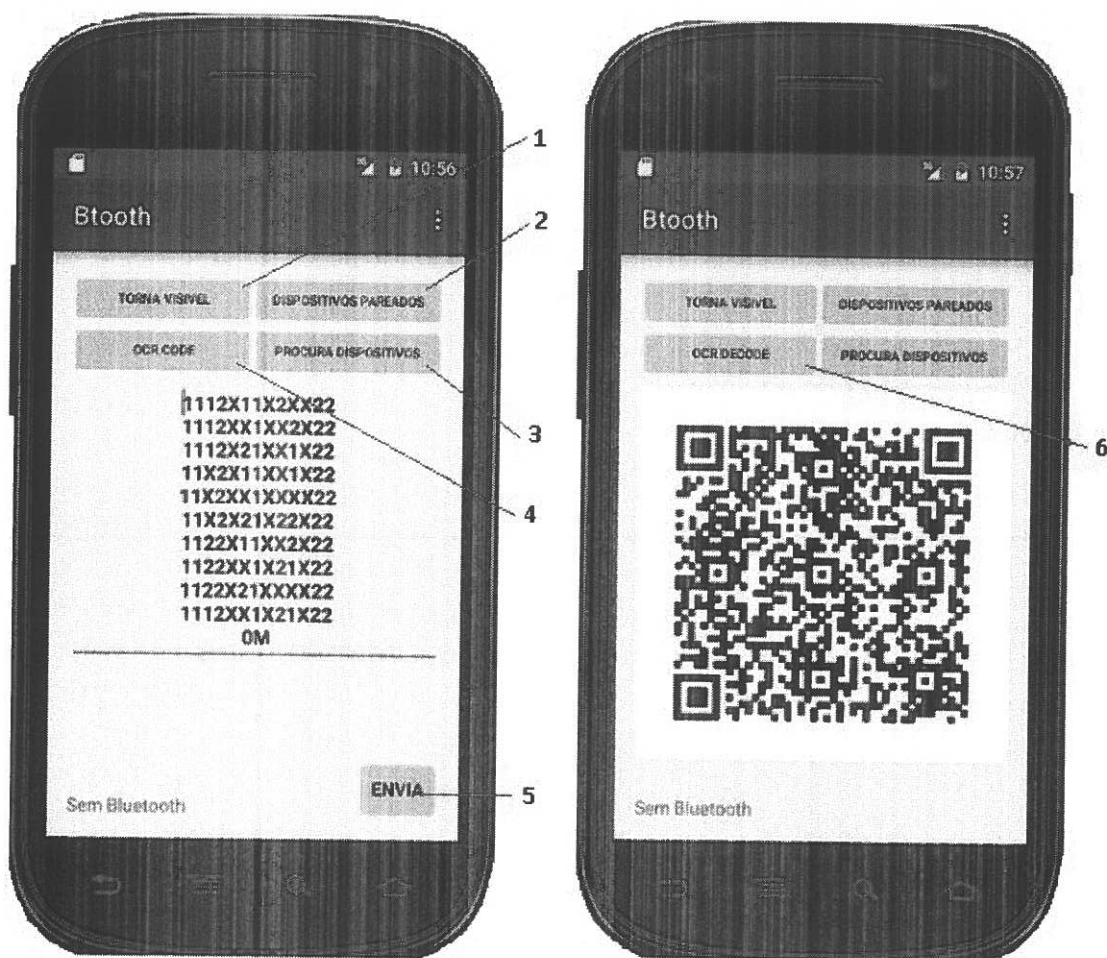


Figura 11 – Comandos da tela de registo das apostas

- 1 – Ativa a funcionalidade Bluetooth e torna visível o dispositivo para os outros dispositivos.
- 2 – Mostra a lista de dispositivos pareados.
- 3 – Procura novos dispositivos.
- 4 – Codifica as chaves em *QR code*.
- 5 – Envia as chaves via Bluetooth a outra aplicação num outro dispositivo.
- 6 – volta a mostrar as chaves no formato 1X2.

### 3.2. Ambiente de desenvolvimento

A aplicação desenvolvida no âmbito deste projeto destina-se a dispositivos móveis do tipo *smartphone* ou *tablet* com versão 4 ou superior do sistema operativo android.

A aplicação foi criada, num computador com o sistema operativo Windows 7 e com o *Java Development Kit 8 (JDK)* instalado, utilizando o ambiente de desenvolvimento padrão da Google, o *IDE Android Studio*, versão 1.5.1, com o *android SDK* que contém os recursos e *plug-ins* essenciais para programar neste ambiente. Como recurso externo foi utilizada a biblioteca (*Lib*) *zxing-1.3.jar*<sup>7</sup> para codificar as apostas em *QR code*.



Figura 12 – IDE Android Studio

Os testes à aplicação foram feitos no simulador do android studio e no equipamento do autor deste trabalho.

### 3.3. Estrutura e Layout

O *layout* é constituído por ficheiros xml, que definem a aparência, os conteúdos dos ecrãs e permitem a interação com o utilizador.

---

<sup>7</sup> Disponível em: <http://www.java2s.com/Code/Jar/z/Downloadzxing13jar.htm>

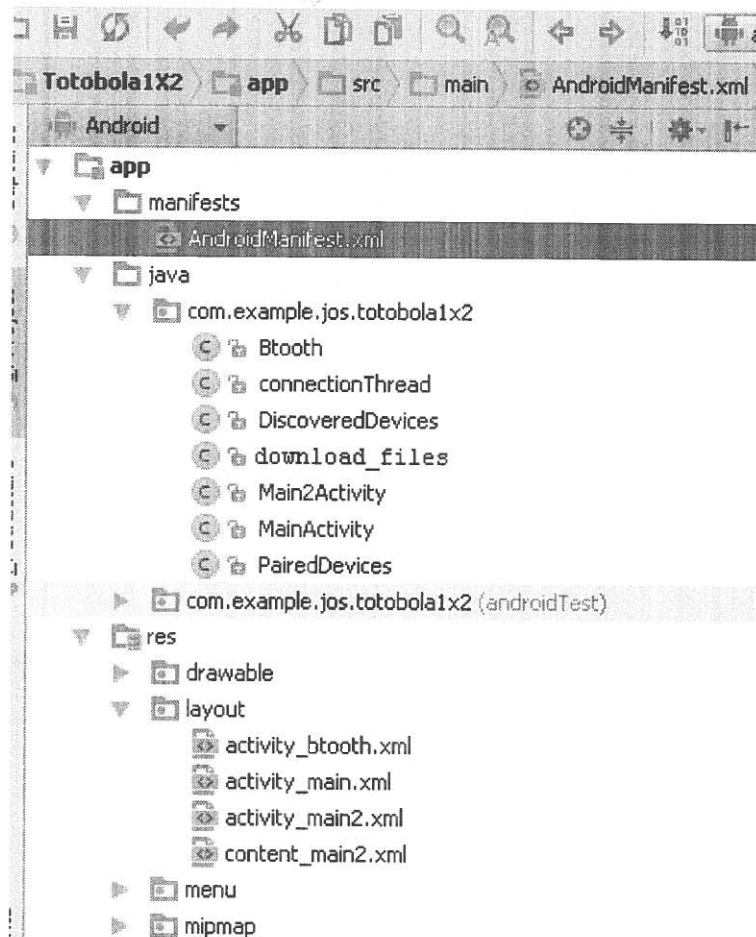


Figura 13 – Estrutura da aplicação

Importa salientar o ficheiro *manifest.xml*. Todas as aplicações android tem de ter um ficheiro *manifest.xml* que fornece a informação necessária ao sistema operativo para este poder executar a aplicação: nome, componentes (as *activities*), as permissões do aplicativo, versão mínima do android requerida para correr a aplicação etc.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.jos.totobola1x2">

    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"
/>
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.BLUETOOTH" />
    <uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/totobola"
        android:label="@string/app_name"
```

```
    android:supportsRtl="true"
    android:theme="@style/AppTheme">
    <activity
        android:name=".MainActivity"
        android:screenOrientation="landscape">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
</manifest>
```

Figura 14 – ficheiro manifest.xml

O código java distribuído pelas classes executa as diversas funcionalidades idealizadas para esta aplicação e a resposta ao acionamento dos objetos do *layout*. Os objetos com mais preponderância na visualização e desenvolvimento da aplicação são as *tablelayout*, as *tablerow* e as *Edittext*.

Uma *tablelayout* é uma *viewgroup* constituída por *tablerows*.

Cada *tablerow* representa uma linha na *tablelayout* podendo conter várias colunas, originando células que podem conter a variedade de objetos: *Imageviews*, *buttons*, *textviews*, *edittexts*, etc.

```
<TableLayout
    android:layout_width="160dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="25dp"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:id="@+id/tableL2"
    android:orientation="vertical"
    android:padding="3dip"
    android:visibility="invisible"
    android:background="#ffffff">
    <TableRow
        android:layout_width="160dp"
        android:layout_height="match_parent"
        android:background="@drawable/bordas">
        <EditText
            android:layout_width="78dp"
            android:layout_height="16dp"
            android:background="@drawable/bordas"
            android:textAppearance="?android:attr/textAppearanceSmall"
            android:id="@+id/textV0"
            android:layout_column="0"
            android:textAlignment="center"
            android:textSize="13dp" />
    </TableRow>
```

Figura 15 - Exemplo de código xml de uma *tablerow* com uma *edittext* e inserida numa *tablelayout*

A aplicação inicia-se com a classe *MainActivity*, onde residem a maioria dos métodos, com o carregamento das apostas caso o ficheiro *apostas.txt* exista ou tenha informação, com o seguinte código:

```

....
String aposta = "";
int marcados;

String filePath = getFilesDir().toString() + "/Apostas.txt";
if (new File(filePath).isFile()) {
    //só carrega se o ficheiro existir, caso não exista não executa o resto do código

    TableLayout tbl1 = (TableLayout)findViewById(R.id.tableL1);
    // Crio uma referência para a TableLayout denominada tableL1
    try { // Permite-me capturar o erro caso este pedaço de código: entre o try e o catch corra mal
        FileInputStream fileapostas = openFileInput("Apostas.txt");
        for (int j = 1; j < 11; j++)//até á penultima textview de cada tablerow (1 a 10)
        {
            for (int i = 1; i < tbl1.getChildCount() - 2; i++) {
                TableRow tr = (TableRow) tbl1.getChildAt(i);
                //referencio as tablerows da 1º á 13º da tbl1 que por sua vez referencia a Tablelayout tableL1

                TextView Txch = (TextView) tr.getChildAt(j);// só uma textview de cada tablerow
                //é a tablerow que se incrementa para referenciar na vertical uma textview de cada tablerow
                (ex:a textview1 da tablerow 1,a textview1 da tablerow2 ..... textview1 da tablerow13. E só
                depois o j incrementa para aceder ás textview2 de toas as tablerows e por ai fora....
                if ((marcados = Fileapostas.read()) != -1)//se diferente de -1
                //ao mesmo tempo obtenho informação e testo. marcados é o numero na tabela ascii que
                representa o carácter.
                {
                    aposta = Character.toString((char) marcados);
                    //Converter em string o carácter ascii para inserir nas caixas de textos
                    Txch.setText(aposta);
                    Txch.setTextColor(Color.BLUE);//cor azul indica lidos
                    Txch.setTypeface(null, Typeface.BOLD);
                }
            }
        }
    } catch (Exception e)
    ...

```

Os prognósticos dos jogos podem ser escolhidos manualmente. Tocando nas *textviews*, da grelha das apostas, estas vão mudando as opções: 1, X ou 2, a cor do texto muda para magenta e o tipo de letra para negrito a fim de indicar que nos encontramos num processo de seleção das chaves.

código para programar as referidas funcionalidades:

```

String texto;
texto = textView.getText().toString();

```



```
        if (texto.equals("1")) {
            textView.setText("X");
        }
        if (texto.equals("X")) {
            textView.setText("2");
        }
        if (texto.equals("2")) {
            textView.setText("1");
        }
    }
    textView.setTextColor(Color.MAGENTA);
    textView.setTypeface(null, Typeface.BOLD);
}
```

Ou podem (os prognósticos) ser gerados aleatoriamente, com o seguinte código:

```
int i=0;

while (i < 13) {
    TableRow tr = (TableRow) tbl1.getChildAt(i + 1); // linhas
    TextView Txch = (TextView) tr.getChildAt(1); //1ªtext view de cada linha
    Random gerador = new Random();

    int numero = gerador.nextInt(2);

    if (numero = 0) {
        aleatoria = aleatoria.concat("1");
    } else {
        if (numero =1) {
            aleatoria = aleatoria.concat("2");
        }
        else {
            if (numero = 2 {
                aleatoria = aleatoria.concat("X");
            }
        }

        i++;
    }
}
```

O Desdobramento é feito pelo método `desdobra` que se orienta pelo tamanho da linha do ficheiro `desdobra`. O utilizador tem de respeitar o número de triplas e duplas nele contido.

Código do método `desdobra` para fazer o desdobramento das apostas:

```
public void desdobra() {
    byte[] dados;
    String desdobra = "";
    String aposta = "";
    int ntr; // as triplas no ficheiro desdobra vem primeiro. Sentido ntr principio →
    int tam; // as duplas no ficheiro desdobra vem em ultimo. Sentido tam ← fim
    String filePath = getFilesDir().toString() + "/desdobra.txt";
    if (new File(filePath).isFile()) {
        try {
```

```

    FileReader arq = new FileReader(filePath); // referencia o ficheiro do desdobramento
    BufferedReader lerArq = new BufferedReader(arq); // carregá-lo para um buffer
    FileOutputStream Fgrava = openFileOutput("Apostas.txt", MODE_PRIVATE); //
    cria o ficheiro apostas para gravar o desdobramento
    TableLayout tbl1 = (TableLayout) findViewById(R.id.tableLI);
    //verificar se a aposta (1ª coluna) está completa
    // Ler o ficheiro do desdobramento até ao fim e carregar linha a linha
    while (desdobra != null) {
        desdobra = lerArq.readLine(); // ler linha do ficheiro
        if (desdobra == null) { // a ultima linha pode dar null e a string desdobra passar a
            null depois do while
                arq.close();
                Fgrava.close();
                return;
            }
        tam = desdobra.length() - 1;
        ntr = 0;
        for (int i = 1; i < tbl1.getChildCount() - 2; i++) { //percorro as tableRow da 1 à 13
            da TableLayout
                TableRow tr = (TableRow) tbl1.getChildAt(i); // que são as TableRow
                TextView Txch = (TextView) tr.getChildAt(1); // interessam as primeiras 3
            textView de cada tableRow
                TextView Txch1 = (TextView) tr.getChildAt(2);
                TextView Txch2 = (TextView) tr.getChildAt(3);
                if (Txch.getCurrentTextColor() != Color.MAGENTA) { // se houver um
                    prognostico por marcar informa e sai.
                        String msg = ("Tem de marcar a aposta, as duplas e triplas\n de acordo
                    com o ficheiro que descarregou");
                        Toast.makeText(getApplicationContext(), msg,
                    Toast.LENGTH_SHORT).show();
                        //fecho dos ficheiros
                    arq.close();
                    Fgrava.close();
                    return; // sai
                } else {
                    if (Txch1.getCurrentTextColor() == Color.MAGENTA) { //esta aqui uma
                        dupla ou tripla se a Txch2 estiver marcada com cor magenta
                            if (Txch2.getCurrentTextColor() == Color.MAGENTA) { //tripla
                                aposta = aposta.concat(String.valueOf(desdobra.charAt(ntr)));
                                ntr++; //incrementa 1
                            } // fim tripla
                            else { //dupla
                                String temp = String.valueOf(desdobra.charAt(tam));
                                if (Txch.getText().toString().equals(temp) ||
                                Txch1.getText().toString().equals(temp)) {
                                    aposta = aposta.concat(temp);
                                    tam--;
                                } else { // as duplas no ficheiro do desdobramento estão representadas
                                    por 1X
                                        aposta = aposta.concat("2"); //se o carater na Txch e Tch1 for
                                        diferente do da posição correspondente da string desdobra! então temos uma dupla tipo 12 ou
                                        X2
                                            tam--; // decremenenta
                                        }
                                    }
                                }
                            }
                        } else {
                            aposta = aposta.concat(Txch.getText().toString());
                        }
                    }
                }
            }
        }
    }

```

```

    }
    } // fim do for
    dados = aposta.getBytes(); // aqui a string aposta está completa com 13 simbolos 1x2 e é
    convertida em bytes          Fgrava.write(dados); // para gravar no ficheiro

    aposta = ""; // limpa o conteúdo e volta a while
    } // fecho do while
    arq.close();
    Fgrava.close();
    } // fim do try
    catch (Exception e) {
        String msg = ("Erro: " + e.getMessage());
        Toast.makeText(getApplicationContext(), msg, Toast.LENGTH_LONG).show();
    }
} // fim do if isfile
}

```

O método `inicia()` serve para reiniciar a grelha das apostas (marcar tudo a X's):

```

public void inicia() {
    TableLayout tbl1 = (TableLayout) findViewById(R.id.tableL1); // identifico a TableLayout--
    >tableL1

    for (int i = 1; i < tbl1.getChildCount() - 2; i++) { // percorro todos os objetos da TableLayout
    na horizontal
        TableRow tr = (TableRow) tbl1.getChildAt(i); // que são as TableRow

        for (int j = 1; j < tr.getChildCount() - 1; j++) { // percorro todos os objetos da TableRow
            TextView Tx = (TextView) tr.getChildAt(j);
            Tx.setText("X");
            Tx.setTextColor(Color.BLACK);
        }
    }
}

```

A gravação dos prognósticos no ficheiro é feito pelo método `grava_chv()`:

```

public void grava_chv() {
    String aposta = "";
    String msg = "";
    byte[] dados;
    try {
        FileOutputStream Fgrava = openFileOutput("Apostas.txt", MODE_PRIVATE);
        // mode_private cria o ficheiro, em mode_public adiciona dados ao ficheiro
        TableLayout tbl1 = (TableLayout) findViewById(R.id.tableL1); // identifico a TableLayout--
        >tableL1
        for (int j = 1; j < 11; j++) { // até à penultima textView de cada TableRow (1 a 13)
            {
                for (int i = 1; i < tbl1.getChildCount() - 2; i++) { // percorro as TableRow da 1 á 13 da
                TableLayout
                    TableRow tr = (TableRow) tbl1.getChildAt(i); // que são as TableRow
                    TextView Txch = (TextView) tr.getChildAt(j);
                    if (Txch.getCurrentTextColor() != Color.MAGENTA) {
                        msg = String.valueOf(j - 1).concat("-Apostas guardadas");
                    }
                }
            }
        }
    }
}

```

```

        Toast.makeText(getApplicationContext(), msg, Toast.LENGTH_SHORT).show();
        //Mensagem no ecrã caso a app encontre aposta incompleta
        // se sair aqui tenho que fechar o ficheiro
        Fgrava.close();
        return;
    }
    aposta = aposta.concat(Txch.getText().toString()); // ai adicionando os simbolos 1 ou X
ou 2

    }
    dados = aposta.getBytes();
    Fgrava.write(dados);
    aposta = ""; // aqui a string aposta está completa com 13 simbolos 1x2 passa para o
ficheiro e volta a
    }
    Fgrava.close();
    aposta = "";
    msg = "10 Apostas guardadas";
    Toast.makeText(getApplicationContext(), msg, Toast.LENGTH_SHORT).show();

} catch (Exception e) {
    msg = ("Erro: " + e.getMessage());
    Toast.makeText(getApplicationContext(), msg, Toast.LENGTH_SHORT).show();
}
}
}

```

Para carregar os jogos e visualizá-los foi desenvolvido o método Le\_jogos():

```

public void Le_jogos() {
    String jogo = "";
    String msg = "";
    String filePath = getFilesDir().toString() + "/Jogos.txt";
    if (new File(filePath).isFile()) { //só carrega se o ficheiro se existir
        TableLayout tbl2 = (TableLayout) findViewById(R.id.tableL2); // identifico a
        TableLayout-->tableL2
        try {
            FileReader arq = new FileReader(filePath);
            BufferedReader lerArq = new BufferedReader(arq);
            for (int i = 0; i < tbl2.getChildCount() - 2; i++) {
                TableRow trl = (TableRow) tbl2.getChildAt(i);
                EditText Ett = (EditText) trl.getChildAt(0);
                EditText Ett1 = (EditText) trl.getChildAt(1);
                jogo = lerArq.readLine(); //exemplo: Sporting,Arouca
                String[] separaequip = jogo.split(","); //a virgula é o meu separador
                Ett.setText(separaequip[0]); //Na primeira edittext da tablerow fica Sporting
                Ett1.setText(separaequip[1]); //na segunda fica Arouca
            }
            arq.close();
            msg = "Jogos carregados";
            Toast.makeText(getApplicationContext(), msg, Toast.LENGTH_LONG).show();
        } catch (Exception e) {
            msg = ("Erro : " + e.getMessage());
            Toast.makeText(getApplicationContext(), msg, Toast.LENGTH_SHORT).show();
        }
    }
}
}
}

```

Desenvolvimento de uma aplicação móvel para jogar o totobola

Relativamente às funcionalidades de comunicação e transferência de dados com que se apetrechou a aplicação importa realçar os pedaços de código que se seguem.

Esta linha de código é para ativar o bluetooth, se o dispositivo possuir esta funcionalidade:

```
BluetoothAdapter meu_bt = BluetoothAdapter.getDefaultAdapter();
```

O trecho de código seguinte passa as apostas a outro dispositivo via Bluetooth:

```
BluetoothDevice device = btAdapter.getRemoteDevice(address); // aponto para o dispositivo
com que quero comunicar usando o seu macadress

//Para estabelecer a ligação é necessário o macadress e o UUID

btSocket = device.createRfcommSocketToServiceRecord(MY_UUID); //Cria um socket
Bluetooth.

//Sockets é um mecanismo de comunicação entre processos

btSocket.connect(); // Solicita ligação ao dispositivo

OutputStream = btSocket.getOutputStream(); // referencia para a saída de dados através do socket
Bluetooth

// depois de recolher as chaves numa string denominada apostas

byte[] msgBuffer = apostas.getBytes(); //converto a string apostas num array de bytes que
denomino de msgBuffer

OutputStream.write(msgBuffer); // transmito as chaves
```

Codificação das chaves em código de barras bidimensional (OCR):

```
...
ByteMatrix bitMatrix = writer.encode(txv.getText().toString(), BarcodeFormat.QR_CODE, 512,
512);
int width = 512;
int height = 512;
Bitmap bmp = Bitmap.createBitmap(width, height, Bitmap.Config.RGB_565); // Cada
pixel é armazenado em 2 bytes e somente os canais RGB são codificados : vermelho é
armazenado com 5 bits de precisão (32 valores possíveis) , o verde é armazenada com 6 bits de
precisão ( 64 valores possíveis) e azul é armazenado com 5 bits de precisão.

for (int x = 0; x < width; x++) {
    for (int y = 0; y < height; y++) {
        if (bitMatrix.get(x, y) == 0)
            bmp.setPixel(x, y, Color.BLACK);
        else
            bmp.setPixel(x, y, Color.WHITE);
    }
}
img_qr.setImageBitmap(bmp); //img_qr é a ImageView que mostra o OCR. . .
```

#### 4. Considerações sobre as apostas *online* e a Aplicação

Esta aplicação foi pensada e desenvolvida do lado do apostador, o próprio Autor é apostador, aproveitando as tecnologias e sem romper com a metodologia tradicional do jogo (registar as apostas no mediador, receber o talão e manter o anonimato) mas tornando-o mais fácil, mais simples, mais barato para o proprietário do jogo com a redução de boletins e mais disponível, sempre acessível ao apostador no *smartphone*, não precisando de adquirir boletins para jogar.

Para jogar *online*, acedendo pelo *browser* ao *site* das apostas ou através da aplicação dos proprietários do jogo, é necessário registo para identificar o apostador e abertura e manutenção de conta com saldo suficiente para pagamento das apostas que se pretendem realizar. Esta condição traduz-se numa vantagem em termos financeiros proporcional ao número de apostadores registados. Suponhamos que uma casa de apostas tem 300 mil apostadores registados, como as contas têm de ter sempre saldo positivo, basta que em média as contas tenham sempre no mínimo um valor remanescente de 3 euros, o que não é difícil! Para a casa ter um montante, permanente, de 900 mil euros ao seu dispor!

Será que, mediante o aumento da adesão ao método online, com as vantagens da redução de custos de emissão de boletins e de redução nas comissões aos mediadores dos jogos, a SCML evoluirá para este método exclusivamente?! Sujeitando os restantes apostadores e, já agora, comprometendo o negocio dos mediadores?

Face ao exposto, Creio que dificilmente os terminais de apostas irão evoluir no sentido da aplicação desenvolvida, não disponibilizando a funcionalidade de transferência das apostas diretamente do *smartphone* para eles. Em contrapartida o apelo e incentivo para que o apostador se registe devem aumentar.

#### 5. Conclusões

O desenvolvimento desta aplicação mostrou que, para lá do acesso à internet, existem outras potencialidades nos telemóveis modernos, ao nível da comunicação e transferência de dados, que podem ser explorados pelas aplicações moveis. A comunicação por infravermelhos, bluetooth, *NFC* e a codificação e leitura de códigos QR são tecnologias que os permitem ligar-se entre si e com outros dispositivos eletrónicos para comunicar e transferir dados.

A intenção deste projeto foi criar uma aplicação, para dispositivos moveis, que

eliminasse a utilização de papel em todos os passos do processo do jogo do totobola, incluindo o registo das apostas, ficando só o talão de registo emitido pelo terminal de jogos. Assim foram programados métodos na aplicação, para transferir dados, as apostas, com a utilização da funcionalidade Bluetooth e outros para codificarem as apostas em *QR Code* a fim de serem lidas num dispositivo com leitor próprio ou com recurso a uma aplicação leitora de códigos QR (único que pode ser lido por uma aplicação leitora de códigos). Para que o apostador não tenha que recorrer à utilização de papel quando pretende jogar com desdobramentos, a aplicação dispõe da funcionalidade de desdobrar apostas em conformidade com o sistema de desdobramento escolhido do grupo por ela disponibilizado. Possui, ainda, a funcionalidade de gerar chaves aleatórias simples ou condicionadas.

O outro objetivo da aplicação, pela elevada facilidade e disponibilidade que oferece, é ser opção ao método de jogar *online*, quer seja feito acedendo diretamente ao portal via *browser* ou através da *App* oficial, caso esta contemplasse o jogo do totobola.

jogar *online*, obriga a registo para identificar o apostador e abertura e manutenção de conta com saldo suficiente para o pagamento das apostas que se pretendem realizar. Uma vez que o saldo tem de ser sempre positivo, como se compreende, esta condição traz vantagem em termos financeiros, aos exploradores dos jogos, proporcional ao número de apostadores registados, como foi referido no capítulo 4, e será, talvez, um entrave á aceitação, por parte do proprietário do jogo, desta e de outras aplicações parecidas que possam vir a desenvolver-se.

## Bibliografia

- Allen, S., Graupera, V., Lundrigan, L. (2010). *Pro Smartphone Cross-Platform Development: iPhone, Blackberry, Windows Mobile and Android Development and Distribution*. (1 st ed.). New York: Apress.(pp 17-33)
- Costa, N.P.O., Filho, N.F.D.,Duarte, A.F.(2012). *Avaliação comparativa de sistemas operacionais para dispositivos móveis: foco em suas funcionalidades* (Relatório Final - 196, 9º CONTECSI ). São Paulo, TECSI – USP.(pp 3016-3023)
- Firtman, M. (2013). *Programming the Mobile Web, Second Edition*. Sebastopol, CA.: O'Reilly Media, Inc. (pp 5-12, 699-701)
- IDC (2015). *Smartphone OS Market Share, 2015 Q2*. Disponível em: <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>
- INFOWESTER (2014) *Tecnologia Bluetooth: o que é e como funciona?*. Disponível em: <http://www.infowester.com/bluetooth.php>
- Law, C. & So, S. (2010). *QR codes in education. Journal of Educational Technology Development and Exchange*, 3(1), 85-100. Hong Kong Institute of Education.(pp 85-87)
- Lee, H., & Chuvyrov, E. (2014). *Beginning Windows Phone APP Development* (Mango Release.). New York: Apress.(pp 4-7)
- PPLWARE (2014). *McDonalds vai adoptar pagamentos via NFC até 15 de Setembro*. Disponível em <http://pplware.sapo.pt/apple/mcdonalds-vai-adoptar-pagamentos-via-nfc-ate-15-de-setembro/>
- Reynolds, M. (2014). *Xamarin Mobile Application Development for Android*. Birmingham B3 2PB, UK: Packt Publishing Ltd.(pp 7-10)
- SCML© (2012a). *Condições de Utilização do Cartão de Jogador*. Disponível em: <https://www.jogossantacasa.pt/web/AjudaComoApostar/>
- SCML© (2012b). *Extrato do Regulamento do Totobola*. Disponível em: <https://www.jogossantacasa.pt/web/Ajuda/contextHelp?topic=76>
- Vermaat, M. E., Sebok, S. L., Freund, S. M., Frydenberg, M. F., Campbel, J. T. (2016). *Discovering Computers 2017, Enhanced Edition*. Boston: Cengage Learning.(pp 304-308)