



IPG Politécnico
| da | Guarda
Polytechnic
of Guarda

RELATÓRIO DE PROJETO

Licenciatura em Engenharia Informática

Ivo Miguel Silveira Neto

novembro | 2015

Instituto Politécnico da Guarda

Engenharia Informática

Projeto de Informática



Aplicação Android

**Sistema de orientação interior
com recurso a códigos QR**

Autor: Ivo Neto

Orientador: Prof. Carlos Carreto

Novembro 2015

Resumo

O presente documento relata o processo de desenvolvimento de uma aplicação para dispositivos móveis que fornece um sistema de orientação interior, visando a fácil movimentação dentro de estabelecimentos públicos. Começou-se por introduzir as tecnologias, *software* e equipamentos, de forma a disponibilizar as bases teóricas e introduzir as ferramentas que possibilitaram o processo implementação em concreto. São apresentadas as funcionalidades da aplicação, a temática e finalmente a descrição técnica das partes relevantes da implementação.

Palavras Chave: *Orientação, Localização, Deslocamento, Mapa Interior, Trajeto, Android, Código QR, Instituição pública.*

Conteúdo

1		6
1	Introdução	6
1.1	Enquadramento e Motivação	6
1.2	Descrição do Problema	7
1.3	Solução Proposta	7
1.4	Objetivo	8
1.5	Contribuição do projeto	9
1.6	Estrutura do Relatório	9
2		10
2	Tecnologias Usadas no Projeto	10
2.1	Tecnologias	10
2.2	Software utilizado	12
2.3	Equipamentos físicos	13
3		14
3	Análise de Requisitos	14
4	Aplicação Móvel Desenvolvida	16
4	Temática	17
5	Funcionalidades	18
6	Implementação	18
6.1	Manifesto	19
6.2	Fornecedor de Conteúdo	19
6.3	Base de Dados	20
6.4	Tarefas Assíncronas	23
6.5	Atividades	23
6.6	Métodos	25
6.7	Ações	27
6.8	Utilização de Códigos QR	28
6.9	Mapas Interiores	30

5	Testes e Resultados	32
6	Conclusão	38

Lista de Figuras

4.1	17
4.2	19
4.3	20
4.4	21
4.5	22
4.6	24
4.7	26
4.8	27
4.9	29
4.10	30
4.11	31
5.1	32
5.2	33
5.3	34
5.4	35
5.5	36
5.6	37

Siglas

- ◆ **QR Code** - Quick Response Code
- ◆ **XML** - eXtensible Markup Language
- ◆ **SQL** - Structured Query Language
- ◆ **IDE** - Integrated Development Enviromnet
- ◆ **API** - Application Programming Interface
- ◆ **HTTP** - Hipertext Transfer Protocol
- ◆ **HTTPS** - Hipertext Transfer Protocol Secure
- ◆ **TCP** - Transmission Control Protocol
- ◆ **DBMS** - Database Management System
- ◆ **DB** - DataBase
- ◆ **UI** - User Interface
- ◆ **GPS** - Global Positioning System
- ◆ **URI** - Uniform Resource Identifier
- ◆ **URL** - Uniform Resource Locater
- ◆ **URN** - Uniform Resource Name
- ◆ **UTF-8** - 8-bit Unicode Transformation Format

Capítulo 1

1 Introdução

A recente demanda pela tecnologia remete para a eventual possibilidade de melhorar a qualidade de vida das pessoas, resolvendo de forma prática problemas simples do quotidiano, através da utilização de dispositivos móveis extremamente comuns na atualidade, nomeadamente *smartphones e tablets*.

No âmbito da disciplina de Projeto de Informática foi proposto escolher entre a realização de um projeto ou de um estágio curricular, desta forma optou-se por desenvolver o presente projeto.

O trabalho produzido consiste numa aplicação móvel que fornece uma seleção de mapas que virtualizam um conjunto de percursos. Esta constitui um sistema de orientação interior (*indoor*).

1.1 Enquadramento e Motivação

O projeto enquadra-se na área das aplicações móveis, mais precisamente no sistema *Android* que apresenta atualmente milhões de utilizadores, o qual oferece inúmeras possibilidades criativas.

Pretende-se assim aprofundar a valiosa experiência nesta área, com uma vasta versatilidade e elevado interesse pessoal e social.

A principal motivação para o desenvolvimento deste projeto foi criar uma solução viável que permita auxiliar e facilitar a vida das pessoas em ambientes variados, tais como: dirigir-se a uma nova universidade ou a um hospital desconhecido.

Com a tendência para o aparecimento de edifícios cada vez mais volumosos e proporcionalmente mais complexos, surge a necessidade de produzir uma ferramenta útil que permita uma fácil acessibilidade das pessoas às várias secções dos estabelecimentos.

Sendo plausível aplicar este conceito em diversas realidades, torna-se mais pertinente focar o problema em instituições públicas de media e grande dimensão.

1.2 Descrição do Problema

Atribuindo importância aos fatores orientação e mobilidade, pretende-se solucionar um problema que o cidadão comum enfrenta em diversas situações do quotidiano.

O problema central deste projeto insere-se em um cenário real, que surge cada vez que se entra pela primeira vez em qualquer edifício público. A questão imediata nesta situação, normalmente seria qual o percurso a realizar de forma a deslocar-se do sítio onde se encontra inicialmente até ao local onde realmente se deseja ir.

Através do poder computacional dos dispositivos móveis e com o auxílio de códigos *QR* foi desenvolvida uma aplicação *Android* capaz de reconhecer o ponto inicial e apresentar ao utilizador uma forma de alcançar o destino pretendido. A solução técnica para este problema concede desta forma a localização da posição do utilizador em qualquer ponto num lugar que era à partida desconhecido.

1.3 Solução Proposta

A solução proposta procura ser bastante universal, podendo ser aplicada em diferentes contextos.

O Sistema desenvolvido apoia-se numa aplicação *Android* que é encarregada de obter e organizar toda a informação necessária ao longo da execução da mesma.

Inicialmente é fornecido (através do código lido) um conjunto de opções que dão acesso às rotas possíveis a partir de um determinado ponto de origem, onde a pessoa se encontra.

É disponibilizado um código *QR* (*Quick Response*) para cada conjunto de trajetos possíveis. Este contém uma ligação a toda a informação necessária para alcançar o destino pretendido.

Para não sobrecarregar o dispositivo, a informação fica armazenada na *cloud* (*internet*), tornando assim a aplicação mais leve, versátil e escalável.

Para este efeito, são descarregados inicialmente os dados acerca das opções disponíveis (possíveis destinos) e as ligações (*links*) aos ficheiros que representam os mapas em concreto. Somente após o utilizador selecionar uma destas opções, são então descarregados os dados que representam o trajeto desde a origem (local onde se encontra o QR) até ao destino escolhido.

1.4 Objetivo

O projeto tem como objetivo criar e desenvolver um sistema de orientação *indoor* para estabelecimentos públicos, garantindo deste modo, uma forma rápida e fácil de determinar o percurso até ao destino pretendido.

A funcionalidade mais importante é permitir ao utilizador final orientar-se e localizar-se imediatamente dentro de um determinado edifício. O sistema produzido é composto por uma aplicação móvel e um conjunto de códigos de resposta rápida (*QR*).

Pretende-se fornecer a possibilidade de determinar o percurso desde um ponto inicial (onde é afixado o código), até uma determinada zona ou local. Posteriormente à leitura de um código é apresentada uma lista com várias opções possíveis que o utilizador seleciona para desta forma poder visualizará o trajeto pretendido. Se eventualmente for lido novo código são acrescentadas novas opções á lista de possíveis destinos.

Para que exista uma interação constante com sistema, durante a visualização do trajeto, o utilizador é forçado a pressionar o botão avançar para que lhe seja mostrado mais do percurso no mapa. Se durante essa movimentação o utilizador desejar voltar ao local de partida, o botão recuar apresenta a animação inversa.

1.5 Contribuição do projeto

Não existindo atualmente soluções semelhantes desenvolvidas, pretende-se trazer uma contribuição positiva e inovadora para as instituições públicas onde esta ideia irá ser aplicada.

Aplicou-se um novo conceito, visto que os sistemas de orientação existentes apenas funcionam no exterior, como é o caso do *GPS (Global Positioning System)*.

Com o sistema criado advém uma contribuição positiva para sistemas de orientação, alargando a sua influência para o interior de edifícios e não só no exterior.

1.6 Estrutura do Relatório

O Relatório está estruturado da seguinte forma:

Capítulo 1 : Introdução - O projeto é introduzido através da motivação e enquadramento; o problema em que se insere; o principal objetivo e a solução que foi proposta e desenvolvida.

Capítulo 2 : Tecnologias, Software e Equipamentos - Apresenta uma serie de tecnologias, *software* e equipamentos utilizados durante o processo criativo.

Capítulo 3 : Análise de Requisitos - Indica e analisa os requisitos que foram necessários.

Capítulo 4 : Aplicação Móvel Desenvolvida - Relata a aplicação em concreto, a sua temática, funcionalidades e é também descrito o processo de implementação em vários pontos.

Capítulo 5 : Testes e Resultados - Expõe alguns testes e resultados mais relevantes.

Capítulo 6 : Conclusão - São declaradas todas as conclusões tiradas.

Capítulo 2

2 Tecnologias Usadas no Projeto

2.1 Tecnologias

Plataforma Android

Plataforma *Android* é constituída por um sistema operativo baseado no núcleo ou *kernel Linux*. Foi inicialmente desenvolvido por uma corporação com o mesmo nome que a caracteriza e posteriormente adquirida pela *Google* no ano 2005.

Na atualidade é mais amplamente utilizada em dispositivos móveis, continuando em crescimento exponencial. A sua influência esta a ser alargada a outros setores, tais como: em relógios digitais, consolas, câmaras e até mesmo em algumas televisões (as *SmartTV's*).

Linguagem Java

Java baseia-se numa programação orientada a objetos, criada pela empresa *Sun Microsystems* no início dos anos noventa e foi vendida poucos anos depois para a *Oracle*.

Esta linguagem de programação constitui o grosso da lógica por detrás de qualquer aplicação *Android*.

Algumas das suas características mais importantes são:

- ◆ A sua enorme portabilidade;

- ◆ Recursos de rede;
- ◆ A sua segurança;
- ◆ O vasto conjunto de bibliotecas que disponibiliza;
- ◆ Uma vertente multi-tarefa indispensável na programação *Android*;
- ◆ Sintaxe semelhante á linguagem C++
- ◆ Memória a desalojada automaticamente.

Linguagem XML

Linguagem *XML* (*Extensible Markup Language*) é uma linguagem de marcação que define um conjunto de regras para a codificação de documentos, é amplamente utilizada na Internet.

Utiliza um tipo de codificação *Unicode* de comprimento variável. No projeto em causa foi escolhido o *UTF-8* (*8-bit Unicode Transformation Format*). Pode representar qualquer carater universal padrão, permitindo manipular de forma consistente o texto de qualquer sistema de escrita existente.

Apesar ser focada em documentos, é universalmente utilizado na representação de informação estruturada, tal como no presente projeto.

URI

Denominado por Identificador Uniforme de Recursos, integra uma cadeia compacta de caracteres e é utilizado para identificar e denominar um recurso existente na *internet*.

O principal propósito desta identificação é permitir a interação com representações do recurso em causa, através da rede. O *URI* é composto por *URL* (*Uniform Resource Locater*) e *URN* (*Uniform Resource Name*).

SQLite

SQLite é um sistema de gestão de base de dados relacional, contido numa biblioteca em linguagem C. Ao contrario das Base de dado *DB*, mais usual, esta, não é cliente-servidor (*client-server*). Em vez disso, é incorporada diretamente no programa e implementa a base de dados *SQL* (*Structured Query Language*) sem a necessidade da execução de um *DBMS* (*Database Management System*) separadamente.

Código QR

Código *QR* ou código de resposta rápida é a designação atribuída a um código de barras bidimensional, ou seja, baseia-se numa matriz binária (zeros e uns).

Trata-se de uma evolução do tradicional código de barras (vetor simples). O mesmo foi originalmente desenhado para a industria automóvel no Japão.

Os códigos *QR* incluem uma capacidade de armazenamento muito superior aos códigos de barras. A matriz que os constitui codifica os dados através de pequenos quadrados brancos e pretos que o programa interpreta como sendo o código binário. Os quadrados brancos representam os zeros e os pretos representam os uns, respetivamente.

Para além disto, contem também três quadrados maiores nos cantos, que a câmara utiliza para atribuir a orientação correta na imagem, sendo desta forma criada a sequência binária pretendida.

2.2 Software utilizado

Android Studio

Foi escolhido o recente *IDE* (*Integrated Development Enviromnent*) *Android Studio*, baseado no ambiente de desenvolvimento integrado *IntelliJ* da *JetBrains* e que foi lançado pela *Google* no final do ano 2014. Este é especializado no desenvolvimento de aplicações para a plataforma *Android*, munido de um conjunto de ferramentas e características que o tornam particularmente interessante.

DropBox

Utilizou-se a *Dropbox* como forma temporária de armazenar a informação na *cloud*, porque permite criar links diretos para os ficheiros que se pretende obter. A vantagem de utilizar esta ferramenta é a encriptação automática dos *links*.

LateX

LateX é um sistema para composição tipográfica de alta qualidade que é utilizado na publicação de documentos. Consiste numa biblioteca que é utilizada por um editor de texto específico, de forma a formatar documentos de carácter científico.

2.3 Equipamentos físicos

Foi necessário utilizar um dispositivo móvel com o sistema *Android*, onde a aplicação foi executada, testada diversas vezes e efetuado *debug* da mesma.

Foi igualmente imprescindível um computador portátil onde foi instalado o *IDE* que possibilitou a programação de todo o código que constitui a aplicação.

Capítulo 3

3 Análise de Requisitos

O acesso às plantas do edifício em questão foi um aspeto fulcral neste projeto, estas possibilitaram a criação dos mapas que representam os possíveis trajetos. Para que fosse possível desenvolver os mapas foram tiradas diversas fotografias às plantas de emergência espalhadas pelo ESTG, posteriormente foram selecionadas as mais relevantes e que abrangiam os lugares mais estratégicos.

Através de uma análise da escola foram selecionadas algumas entradas onde fica o respetivo código afixado e que representa a origem física de um grupo de percursos.

Foi necessária uma edição cuidadosa das plantas de forma a transformá-las em mapas mais simples e fáceis de perceber pelo utilizador, sem comprometer alterar a sua integridade.

Foram criados vários ficheiros *XML* que fornecem o conteúdo (*Feed*) necessário para a aplicação após a leitura do código *QR* respetivo. Nestes ficheiros existem vários itens cada um deles contém todos os elementos que constituem um trajeto.

Posteriormente a ser gerada uma estrutura de várias pastas contendo conjuntos de imagens foram proporcionados diversos diretórios em que a aplicação se baseia para proceder ao descarregamento dos ficheiros corretos que compõe os mapas.

Outros dos requisitos chave para a execução deste projeto foram uma planificação exaustiva, criar conhecimento aprofundado sobre as diversas tec-

nologias apresentadas e todo o conhecimento adquirido durante outras cadeiras do curso e Engenharia informática.

Capítulo 4

Aplicação Móvel Desenvolvida

A aplicação móvel consiste no produto final, fruto de uma planificação e escolha entre diversas tecnologias disponíveis.

Foi escolhido o sistema *Android*, porque é largamente mais utilizado no mercado, logo torna a aplicação mais abrangente.

A solução foi desenvolvida com o intuito de auxiliar os utilizadores em situações específicas, facilitando-lhes a vida ao oferecer uma seleção de opções que é obtida através da leitura de códigos *QR* gerados para o efeito. Cada uma destas opções dá acesso a um determinado mapa interior que representa o trajeto escolhido.

A arquitetura do sistema *Android* é demonstrado na figura seguinte:

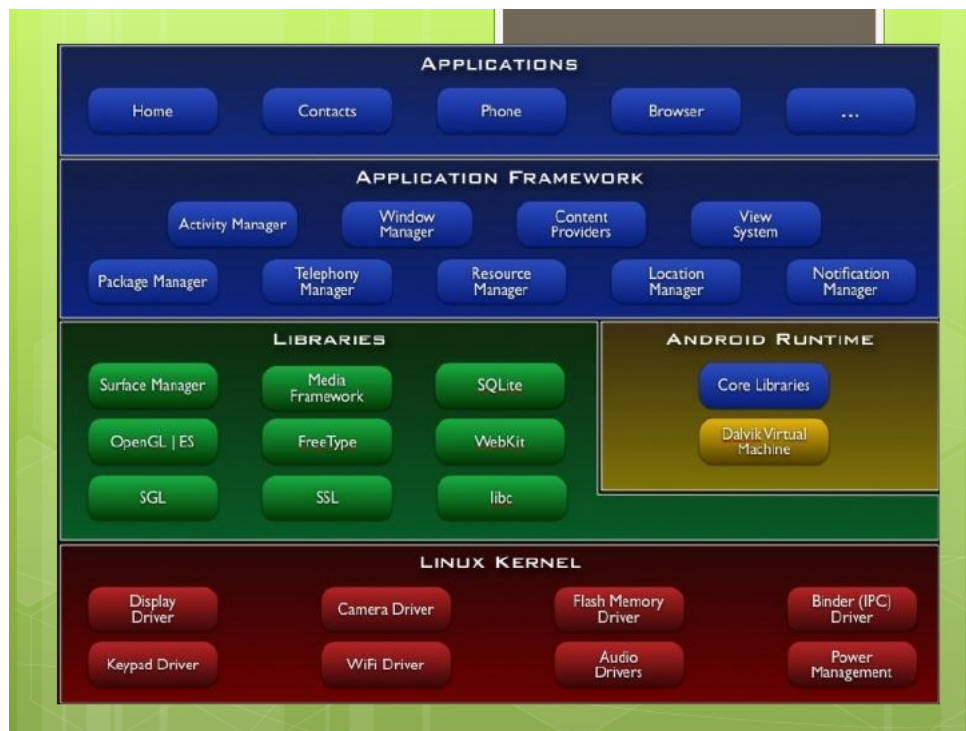


Figura 4.1:

A figura contém um esquema da arquitetura *Android*, com as suas bibliotecas, as *frameworks*, entre outros. Assinalado a vermelho, estão os elementos do núcleo do sistema *Linux*. Fornecem acesso aos módulos físicos do dispositivo como é possível visualizar.

4 Temática

Procura-se garantir acesso fácil a qualquer setor dentro de estabelecimentos públicos, sendo assim dispensável questionar direções e destinos.

Evita-se erros de trajeto e perdas desnecessárias de tempo, proporcionando uma alternativa rápida e prática para se deslocar e orientar no espaço em questão.

5 Funcionalidades

Para prover a constante interação entre utilizador e a aplicação e ao mesmo tempo torná-la mais dinâmica, foram desenvolvidas as seguintes funcionalidades:

- ⇒ Leitura de códigos QR
- ⇒ Escolha de coleções de trajetos
- ⇒ Visualização independente de cada trajeto
- ⇒ Avanço ou recuo no mapa, condicionado pelo utilizador

6 Implementação

A secção Implementação foi subdividida nas seguintes unidades:

- Manifesto
- Fornecedor de Conteúdo
- Base de Dados
- Tarefas Assíncronas
- Atividades
- Métodos
- Ações
- Utilização de Códigos QR
- Mapas Interiores

6.1 Manifesto

Este ficheiro, tal como muitos que integram o corpo da aplicação existe no formato *xml*, mas ao contrário de outros, a designação que lhe foi dada é universal. O Manifesto inclui instruções e indicações importantes para o funcionamento correto da aplicação.

Na figura seguinte são apresentadas as permissões que foram necessárias atribuir á aplicação desenvolvida:

```
-----  
"android.permission.INTERNET"  
"android.permission.WRITE_EXTERNAL_STORAGE"  
"android.permission.READ_EXTERNAL_STORAGE"  
"android.permission.ACCESS_WIFI_STATE"  
"android.permission.ACCESS_NETWORK_STATE"  
"android.permission.SET_DEBUG_APP"  
-----
```

Figura 4.2:

A figura demonstra todas as permissões que foi necessário definir no ficheiro manifesto para garantir acesso á *internet*, escrita e leitura no armazenamento interno do dispositivo, o estado de *wifi* e rede e *debud*.

6.2 Fornecedor de Conteúdo

Constitui um bloco primário do software desenvolvido, fornece o conteúdo (*Content Provider*) necessário para alcançar a finalidade desejada. Implementa as operações básicas *CRUD* (*Create, Read, Update, Delete*), permitindo assim a manipulação consistente da *DB* (*Database*) perante a execução da aplicação.

A informação é encapsulada através do método criado para esse efeito (método *insert*) e posteriormente fornecida através de um pedido (*Content*

Resolver) que é realizado no método *query*, o sistema inspeciona a autoridade do *URI* fornecido e passa o pedido registado.

De forma a caracterizar mais concretamente este processo é demonstrado na próxima figura o método *insert*.

```
-----  
public Uri insert(Uri uri, ContentValues values) {  
    if (URI_MATCHER.match(uri) != TRAJcode) return null;  
        //INSERT INTO TABLE  
        long linhaID = BD.insertWithOnConflict(TABELA_TRAJETOS,  
        null, values, SQLiteDatabase.CONFLICT_REPLACE);  
    if (linhaID > 0) {  
        Uri novoURI = ContentUris.withAppendedId(CONTENT_URI, linhaID);  
        this.getContext().getContentResolver().notifyChange(uri, null);  
        return novoURI;  
    } else {  
        Toast.makeText(getContext(), "Falhou a inserir a linha!!!",  
        Toast.LENGTH_LONG).show();  
        return null;  
    }  
}
```

Figura 4.3:

Código responsável inserir os dados de cada uma das colunas correspondentes na Tabela Trajetos criada anteriormente na base de dados interna ao sistema, nela são guardados os campos dos trajetos fornecidos pelo código lido. Logo de seguida são notificados todos os observadores registados de que a linha foi inserida ou atualizada.

6.3 Base de Dados

A aplicação desenvolvida utiliza a biblioteca *SQLite* para garantir a consistência e integridade dos dados guardados através da inserção (*insert*) e posterior leitura (*query*) á base de dados criada para o propósito.

Foi utilizado um fornecedor de conteúdo (*content provider*) personalizado, afim de criar uma *DB* inerente ao programa e desta forma não ser necessário a ligação a uma base de dados externa, o que neste caso o seria menos adequado.

No momento em que a aplicação é aberta a base de dados é criada no dispositivo e com ela a tabela dos trajetos, se esta ja existir é eliminada e criada novamente (a criação da mesma é demonstrada na figura 4.4). Apenas depois de ser lido um código *QR*, é realizado o *download* dos dados e posteriormente carregados na tabela através de uma tarefa *background*.

A criação da base de dados é demonstrada na seguinte figura:

```
-----  
static final String CRIAR_BD_TABELA = "CREATE TABLE " + TABELA_TRAJETOS +  
    " (id INTEGER PRIMARY KEY AUTOINCREMENT, " +  
    TITULO_DEST + " TEXT NOT NULL, " +  
    ORIGEM + " TEXT NOT NULL, " +  
    LINK + " TEXT NOT NULL, " +  
    DESCRI + " TEXT NOT NULL, " +  
    LINK_IMG1 + " TEXT NOT NULL, " +  
    LINK_IMG2 + " TEXT NOT NULL, " +  
    LINK_IMG3 + " TEXT NOT NULL, " +  
    LINK_IMG4 + " TEXT NOT NULL, " +  
    LINK_IMG5 + " TEXT NOT NULL, " +  
    LINK_IMG6 + " TEXT NOT NULL);";  
-----
```

Figura 4.4:

Código responsável por criar a base de dados do sistema, desta forma é criada a Tabela Trajetos e todos os campos que a constituem. Posteriormente são guardados nesta tabela os elementos que compõem os trajetos fornecidos pelo código lido.

Os campos da tabela criada na base de dados e as suas características e descrição são dispostos a seguir.

ENTIDADES			
CAMPO	TIPO	CHAVE	DESCRIÇÃO
id	String	Primaria	ID do Trajeto
titulo_dest	String	---	Titulo do destino
origem	String	---	Origem do trajeto
link_dest	String	---	Link para o destino
descri	String	---	Descrição do trajeto
link_image1	String	---	Primeira frame do array
link_image2	String	---	Segunda frame do array
link_image3	String	---	Terceira frame do array
link_image4	String	---	Quarta frame do array
link_image5	String	---	Quinta frame do array
link_image6	String	---	Ultima frame do array

Figura 4.5:

Esta figura contem uma tabela que descreve e caracteriza a base lógica onde assenta a aplicação.

6.4 Tarefas Assíncronas

Sem as tarefas assíncronas (*AsynkTasks*) ou tarefas *background* não seria possível construir esta aplicação. Estas tarefas possibilitam o *download* da informação necessária, por detrás das Atividades, sem que o utilizador tenha consciência deste processo.

São utilizadas as seguintes tarefas *background*:

- ❑ Download Lista (*ListDownTask*)
- ❑ Download Trajeto (*TrajetoDownTask*)

6.5 Atividades

Atividades ou *Activitys* definem aquilo que o utilizador consegue visualizar e interagir. Representam as *thread's UI (User Interface)* da aplicação.

Esta aplicação usufrui das seguintes atividades:

- ⇒ Principal (*MainActivity*)
- ⇒ Lista de Trajetos (*ListaViewActiity*)
- ⇒ Visualização do Trajeto (*TrajetoActivity*)

De forma a tornar o funcionamento da aplicação mais explícito, a figura seguinte dispõe um diagrama que explica o processo de execução e a passagem da informação entre as varias atividades e também entre tarefas assíncronas e atividades.

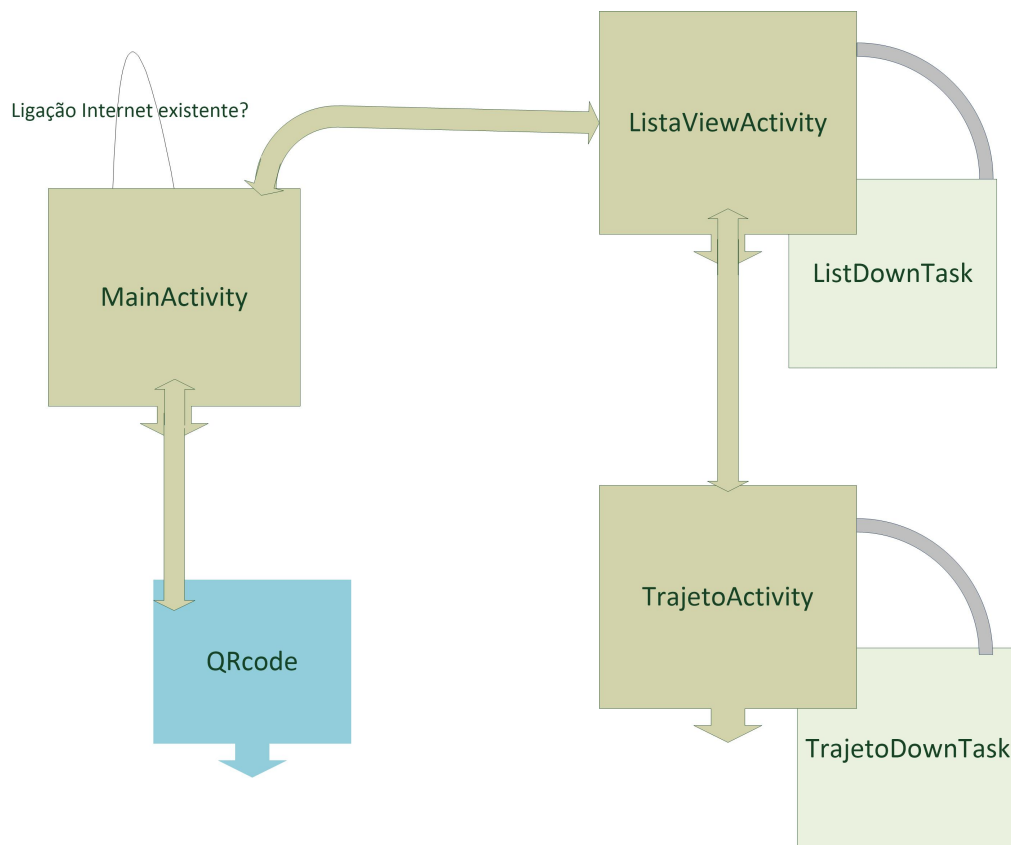


Figura 4.6:

A figura explicita o processo de funcionamento da aplicação de uma forma geral e também demonstra o *flow* de dados.

6.6 Métodos

Os métodos existem dentre das classes, estes desempenham papéis indispensáveis. A cada um deles é lhe atribuída uma diferente função.

Dependendo da tarefa para que são criados retornam um resultado que se pretende ou apenas executam o que se pretende sem que devolvam algum valor.

Na figura seguinte é apresentado um importante método que tem a tarefa de obter a lista de títulos para os trajetos armazenada previamente na tabela da *DB* e os seus *ID's*. Desta forma o cursor passa por todos os valores neste caso como se pretende obter todos os trajetos armazenados, são selecionados todos os elementos da tabela, sem a necessidade de criar outras restrições. Os elementos são introduzidos na lista com o auxílio de um adaptador, como é possível reparar nas ultimas linhas de código.

```
public void getTrajetosLista() {
    String[] projection = new String[]{AppContentProvider.id,
        AppContentProvider.TITULO_DEST};
    iCursor = null;
    listaMenu.clear();
    iCursor = resolver.query(Content_URI,
        projection, null, null, null);
    if (iCursor.moveToFirst()) {
        do {
            String id = iCursor.getString(
                iCursor.getColumnIndex(AppContentProvider.id));

            String titulo = iCursor.getString(
                iCursor.getColumnIndex(
                    AppContentProvider.TITULO_DEST));

            listaMenu.add(titulo);
        } while (iCursor.moveToNext());
    }
    iCursor.close();

    ListAdapter iAdap = new AdapterLista(this,
        R.id.tview_linha, listaMenu);
    iListView.setAdapter(iAdap);
}
```

Figura 4.7:

Código do método *getTrajetosLista* que tem a importante tarefa de obter o ID e o título de todos os trajetos existentes na tabela.

Destá forma é criada a lista com os destinos possíveis que representam as opções selecionáveis pelo utilizador. Depois disto é feito novo *query* para assim obter o destino selecionado.

6.7 Ações

Ações, mais conhecidas por *Intent's* no sistema *Android* possibilitam a passagem de informação entre diferentes atividades. Alterando desta forma o conteúdo da página que o utilizador consegue visualizar em diferentes pontos de execução.

Estas permitem criar uma estrutura dinâmica e estão implementadas dentro de botões, noutros locais do código e sempre que se pretende passar informação entre atividades.

Os dados são colocados num *bundle* (pacote), é também atribuída uma chave a esses dados que são recuperados na segunda atividade, através dessa mesma chave definida.

Na seguinte figura é demonstrada a passagem de um *array de bitmaps* de uma atividade para a outra.

```
-----  
Bundle bundle = new Bundle();  
String chaveL = "chaveLista";  
bundle.putStringArrayList(chaveL, arrayURLsImgs);  
String chaveT = "chaveTitulo";  
  
String chave = "chaveLista";  
Bundle bundle = this.getIntent().getExtras();  
arraylistURLs = bundle.getStringArrayList(chave);  
-----
```

Figura 4.8:

Na primeira metade do código apresentado o array de *bitmaps* é colocado num *bundle* com uma chave e é-lhe atribuída. A segunda metade demonstra como é recebido na outra atividade (*TrajetoActivity*) através da chave que-lhe foi atribuída.

6.8 Utilização de Códigos QR

Os códigos *QR* favorecem um fácil e rápido acesso á informação que se pretende obter e para além disto proporcionam uma origem física para um conjunto de trajetos possíveis.

No futuro enquadramento deste projeto num determinado estabelecimento publico, estes códigos serão colocados em locais específicos e favoráveis, como por exemplo nas entradas, nos acessos, entre outros.

Foi ainda ponderada a necessidade de programar integralmente a leitura do códigos, mas atendendo ao velho ditado "não reinventar a roda", foi utilizado um outro programa externo que é executado aquando a leitura do código. No caso de o dispositivo não conter este programa é mostrada uma caixa de dialogo com a questão se deseja ou não, instalar o *scanner* de *QR's*.

A figura que se segue demonstra a caixa de diálogo que é apresentada, se não for encontrado o scanner de códigos *QR* no dispositivo.

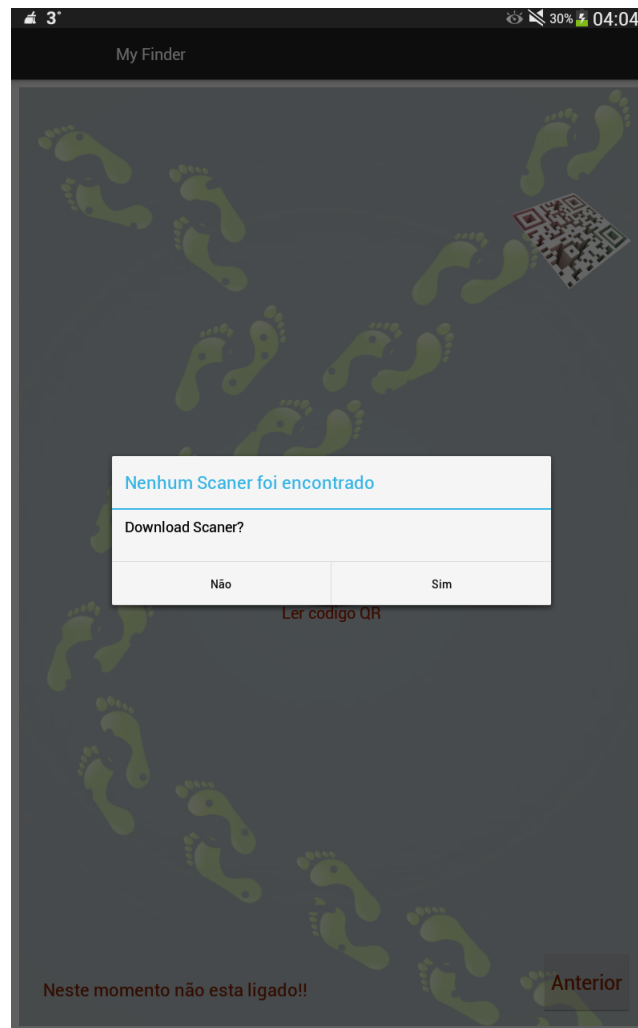


Figura 4.9:

A figura contém um *ScreenShot* com a caixa de diálogo para que se possa instalar o scanner de QR's.

Foram selecionadas algumas entradas estratégicas no edifício da ESTG, para as quais foram gerados vários códigos *QR* que fornecem várias opções com alguns dos destinos mais importantes. Cada vez que é lido um código é acrescentado um novo grupo à lista. Um dos códigos gerados é apresentado na figura seguinte.



Figura 4.10:

A figura disponibiliza um dos Códigos *QR* gerados que fornecem a ligação aos dados necessários à aplicação. Neste caso através da leitura do código apresentado é possível obter o conjunto de opções com os trajetos correspondentes à entrada oeste da Escola.

6.9 Mapas Interiores

Com o intuito de favorecer uma visualização dinâmica dos trajetos e paralelamente garantir uma maior interação entre o utilizador e a aplicação foram desenvolvidos uma série de mapas, cada um deles compostos por várias *frames* que compõe um vetor de imagens.

Os códigos *QR* foram colocados em algumas entradas estratégicas. de modo a incluir na demonstração algumas dos pontos mais importantes no

edifício.

Pretende-se que o utilizador avance no mapa (pressionando o botão avançar) conforme progride no meio físico, ou se voltar para onde veio pressionar o botão recuar. Desta forma torna-se possível melhorar a sua localização e orientação no mapa.

Através de várias plantas de emergência da *ESTG*, que foi utilizado como caso de estudo para o projeto, foram criados diversos mapas interiores dinâmicos.

Estes mapas são compostos por um vetor (*array*) de imagens que é incrementado ou decrementado, através da ação utilizador de pressionar um dos botões, avançar ou recuar respetivamente.

O algoritmo responsável por realizar parte desta tarefa (avançar) segue-se na figura.

```
-----  
int tamanho = arrayListURLs.size();  
if (avancar < tamanho - 1) {  
    buttRecuaFrame.setEnabled(true);  
    avancar++;  
    Bitmap bitmap = (Bitmap) arrayListBitmaps.get(avancar);  
    imageViewTrajeto.setImageBitmap(bitmap);  
} else if (avancar == tamanho - 1) {  
    buttAvancaFrame.setEnabled(false);  
}  
-----
```

Figura 4.11:

Código/algoritmo responsável por criar os mapas dos trajetos de característica dinâmica.

Capítulo 5

Testes e Resultados

Durante todas as fases de implementação foram realizados *debug's* de forma a corrigir erros que foram surgindo e manter o código estável. Esta prática contribuiu acentuadamente para alcançar o resultado pretendido.

Abaixo é apresentado o exemplo de um dos *debug's* realizados.

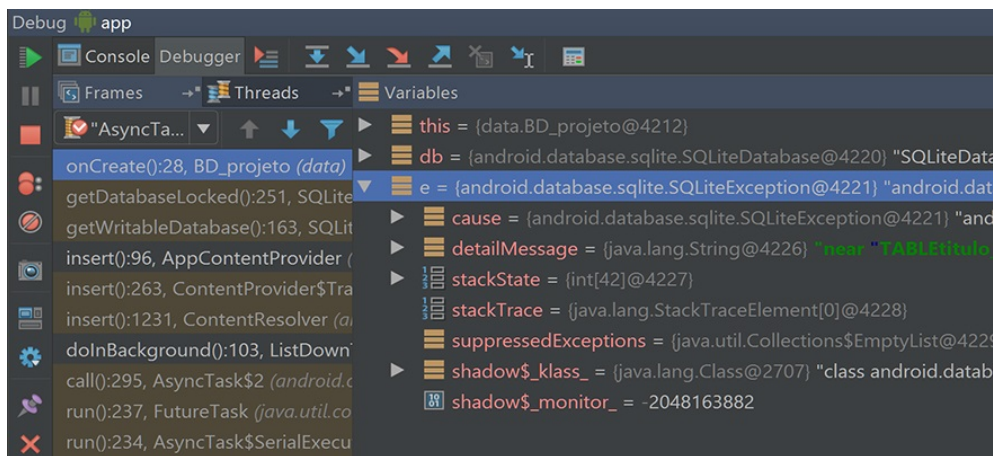


Figura 5.1:

A figura demonstra um exemplo entre os múltiplos *debug's* feitos durante a implementação. Neste caso o *breakpoint* demonstra uma exceção na base de dados (*SQLite*), mais propriamente um erro de sintaxe numa das colunas da tabela.

Outra das formas de testar o funcionamento da aplicação é através da monitorização, na seguinte figura é demonstrada a utilização desta ferramenta que vem integrada no *Android Studio*.

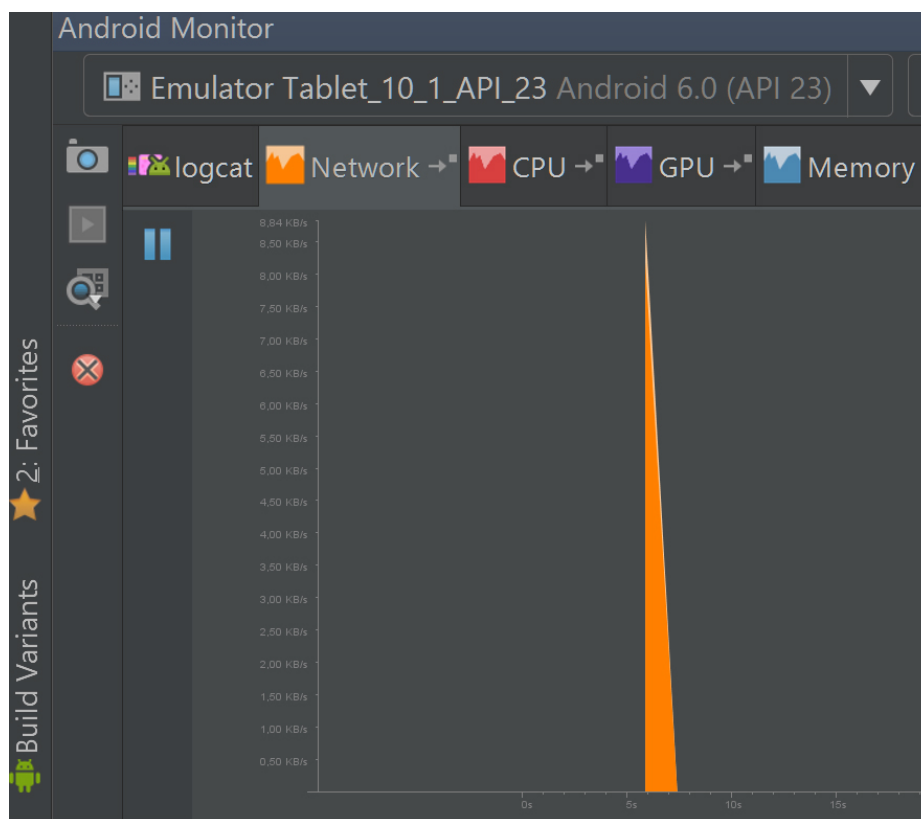


Figura 5.2:

A figura exemplifica o processo de monitorização da quantidade *KiloBytes* que esta a ser descarregada em função do tempo, através da rede.

Como resultado final foi desenvolvido um sistema que dispõe de um auxílio à orientação como se pretendia.

A figura que se segue demonstra a interface utilizador (*UI*) inicial, que corresponde à atividade principal (*MainActivity*), na qual o utilizador tem a opção de ler um novo código ou visualizar a lista anterior de opções.



Figura 5.3:

A figura apresenta um *ScreenShot* da interface utilizador inicial.

Através da leitura de dois dos códigos gerados para o caso de estudo, a aplicação cria a lista de opções que é apresentada seguidamente. Estas disponibilizam alguns dos locais mais importantes facilmente acessíveis através da zona onde o código foi lido.

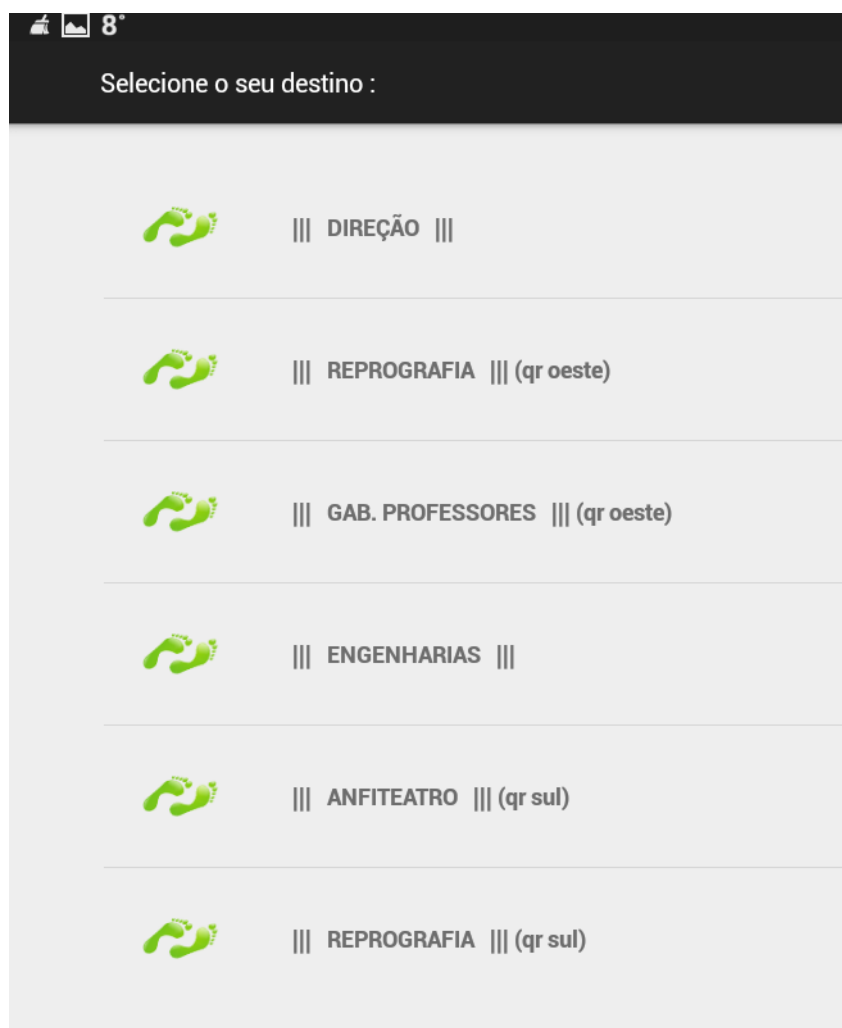


Figura 5.4:

A figura contém uma possível lista com um conjunto de opções para visualizar os percursos.

Quando o utilizador tem a ação de avançar, pressiona o botão respetivo e é acrescentada um novo par de pegadas e assim sucessivamente até alcançar o local pretendido. Se o destino for muito perto é acrescentado apenas um pé de cada vez.



Figura 5.5:

A figura representa metade da progressão no mapa e também no meio físico, do percurso até á reprografia.

Como se pode verificar observando a figura anterior e a seguinte, o mapa não apresentado instantaneamente, em vez disso. O espaçamento entre as pegadas varia de trajeto para trajeto de modo a criar uma perspectiva da distância a percorrer, logo o comprimento do percurso é diretamente proporcional ao espaçamento entre os pares de pegadas.



Figura 5.6:

A figura apresenta um dos mapas criados, neste caso com o percurso desde a entrada este até á reprografia.

Capítulo 6

Conclusão

Pode concluir-se que a realização deste projeto foi um processo extremamente gratificante e produtivo, contribuindo para adquirir e consolidar conhecimentos em diversas áreas tecnológicas.

Em síntese o trabalho realizado tornou possível alcançar os objetivos propostos inicialmente, como se pretendia foi produzido um sistema capaz de favorecer a movimentação e a orientação dentro de estabelecimentos, desconhecidos a partida.

Para tornar possível este projeto foi necessário recorrer a diversas ferramentas, tais como o *Android Studio*, um dispositivo móvel, um gerador de códigos *QR*, entre outras referidas anteriormente.

Em suma, através de uma minuciosa planificação e criando um maior conhecimento sobre o leque de ferramentas disponíveis, permitiu desta forma poder tomar decisões adequadas durante a fase de implementação. Foi assim possível criar e desenvolver uma solução prática e funcional para o problema proposto inicialmente.

Sugere-se que num projeto deste tipo se dê a devida importância a uma planificação atenta do que se pretende realmente fazer e como se irá proceder antes de passar á programação em concreto.

Pretende-se criar novas versões da aplicação, corrigindo algumas limitações e alargando a sua influência a outras instituições. Uma das limitações seria no caso de as plantas disponíveis serem demasiado grandes a forma como

os mapas foram criados teria de ser alterada e a apresentação dos mesmos repensada.

Como primeira versão foi realizado o caso de estudo através da sua implementação em determinadas zonas da Escola Superior de Gestão e Tecnologia do Instituto Politécnico da Guarda.

Referências

<http://developer.android.com/index.html>

<http://developer.android.com/guide/topics/providers/content-provider-creating.html>

<http://developer.android.com/guide/topics/ui/declaring-layout.html>

<http://developer.android.com/guide/topics/ui/layout/listview.html>

<https://developer.android.com/training/multiple-threads/communicate-ui.html>

<http://www.oracle.com/index.html>

<http://stackoverflow.com>

<http://www.xmlmath.net/texmaker>

<https://www.the-qr-code-generator.com>

<http://www.javaworld.com/category/mobile-java>

<http://tex.stackexchange.com>