



U N I V E R S I D A D E
LUSÓFONA

Kuanto Tempo Falta (KTF)

Trabalho Final de curso

Nome do Aluno: João Filipe Matias Ferreira do Nascimento de Oliveira:

Nome do Orientador: Alexandre Banha

Trabalho Final de Curso | LEI | 2015/2016

Direitos de cópia

Kuanto Tempo Falta (KTF), Copyright de João Filipe Matias Ferreira do Nascimento de Oliveira, ULHT.

A Escola de Comunicação, Arquitetura, Artes e Tecnologias da Informação (ECATI) e a Universidade Lusófona de Humanidades e Tecnologias (ULHT) têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objetivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Contents

Índices de Quadros, Figuras, Tabelas, Equações, source code exemplo.....	3
Resumo.....	4
Abstract (em Inglês).....	5
Introdução.....	6
Enquadramento teórico (com revisão bibliográfica).....	7
Método.....	9
Resultados.....	10
Conclusões e trabalho futuro.....	11
Bibliografia.....	12
Anexos (todo o source code do projecto deve ser colocado aqui).....	13
Source Code.....	14
A1. Anexo de Manual técnico da aplicação (se existir).....	34
A2. Anexo de Manual de utilizador da aplicação (se existir).....	35

Índices de Quadros, Figuras, Tabelas, Equações, source code exemplo....

Figuras

Fig01. "SMS ao minuto"	06
Fig02. "E-Mail ao minuto"	06
Fig03. Explicação do Serviço "SMS ao minuto"	07
Fig04. Explicação do Serviço "E-mail ao minuto"	07

Tabelas

Tab.01. Percentagem de Dispositivos Android / API	06
Tab.02 Sequencia de modificação do Excel original	09

Source Code

<Layout>

activity_main_menu.xml	14
activity_by_bus.xml	15
activity_by_stop.xml	15
activity_by_other.xml	16
activity_rede.xml	16

<Class>

MainMenu.java	18
ByBus.java	19
ByStop.java	21
ByOther.java	23
Rede.java	27
DbExternalHelper.java	30

Imagens Manual do Utilizador

img.01 Main Menu	32
img.02 By Bus	32
img.03 By Stop	32
img.04 By Other	32
img.05 Rede	32

Resumo

Neste trabalho procurou-se elaborar se uma aplicação para Android, que providenciasse ao utilizador um "interface" aos serviços providenciado pela Carris transportes de Lisboa, sendo estes [] "SMS ao minuto" e "E-mail ao minuto", Estes serviços são muito interessantes e bastante inovadores, mas tanto quanto consigo perceber muito pouco usados, eu penso que seja pelo "interface" com o serviço não ser intuitivo o suficiente, visto que por defeito não são muito utilizáveis, tentou-se que esta aplicação facilitasse o acesso a estes serviços, Android porque é o OS móvel mais [2] disseminado logo foi a escolha óbvia, um dos componentes mais importantes e que essencialmente possibilitou a criação desta aplicação foi um ficheiro Excel [no CD] (bastante extenso e algo mal organizado mas essencial) a dimensão deste ficheiro levou a necessidade de utilização de uma Base de Dados SQLite que resulta deste ficheiro (com algumas modificações para aumentar a performance da aplicação), depois de algumas tribulações acho que o resultado final apesar de não ser perfeito (ver Conclusões e trabalho futuro) é extremamente funcional e algo que usarei de hoje em diante.

Abstract (em Inglês)

In this project there was an attempt at creating an application for Android, that provided to the user an "interface" for the services provided by Carris transportes de Lisboa, these services being [] "SMS ao minuto" e "E-mail ao minuto", These services are very interesting and quite innovative, but as far as I can gather they are not used that much, I think that is due to the fact that the services "interface" is not intuitive and marginally usable, So this application attempts to ease the access to these services, Android was chosen because of the [2] dissemination of this OS therefore the obvious choice, one of the most important components that essentially allowed the creation of this application was a Excel File [no CD] (quite extensive and somewhat poorly organized but essential) the size of this file lead to the need to use a SQLite Data Base witch is in essence this file (with some modifications to enhance the performance of the application), After some tribulation I think the end result although not perfect (see the Conclusão e trabalho futuro) is extremely functional and something I will use from this day forward.

Introdução

Eu sou um utilizador inveterado dos transportes públicos e como tal viajo na carris diariamente, e já a algum tempo que uso os serviços "SMS ao minuto" e "E-mail ao minuto", acho que estes serviços são brilhantes mas muito pouco utilizáveis, assim sendo



Fig.01-"SMS ao Minuto"



Fig.02-"EMail ao minuto"

de instruções conseguissem utiliza-la, fosse o menos possível dependente do online pois nem todos os utilizadores possuem um pacote de telemóvel com dados / sequer sabem o que isso é, e apontei á [5] API 15 excluindo somente cerca de 2,8% dos dispositivos disponíveis assim aumentando o target desta aplicação, em suma o Objectivo deste trabalho é a criação de uma aplicação que seja o mais simples e linear para que o publico alvo os utilizadores da carris a usem independentemente do seu conhecimento tecnologico

que me encontrava sem tópico de TFC e já que isto é algo que me pareceu interessantes e útil resolvi tentar criar um "interface" para estes serviços, desde logo quis que esta interface fosse o mais simples e linear possível para que os utilizadores com

um mínimo

Version	Codename	API	Distribution
2.2	Froyo	8	0.1%
2.3.3 - 2.3.7	Gingerbread	10	2.7%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	2.5%
4.1.x	Jelly Bean	16	8.8%
4.2.x		17	11.7%
4.3		18	3.4%
4.4	KitKat	19	35.5%
5.0	Lollipop	21	17.0%
5.1		22	17.1%
6.0	Marshmallow	23	1.2%

Data collected during a 7-day period ending on February 1, 2016.

Any versions with less than 0.1% distribution are not shown.

[5] Tab.01 - percentagem de dispositivos android / API

Enquadramento teórico (com revisão bibliográfica)

Então comecei a estudar o problema como criar este "interface":

1. [7] Como Funcionam estes Serviços então?

SMS ao minuto



O sistema de mensagens SMS ao minuto tem como principal objetivo proporcionar informação, via telemóvel, sobre os horários reais de passagem dos veículos nas paragens. A pedido do Cliente, a CARRIS proporciona a consulta de informação fiável e de última hora sobre a passagem dos veículos, em cada paragem.

Como posso utilizar o serviço?

Para consultar tempos de espera (em minutos), relativamente a todos os veículos que passam numa determinada paragem:

- Deverá digitar C (espaço) Código da Paragem e enviar para o 3599 (custo de mensagem escrita normal);

Para consultar tempos de espera dos próximos três veículos de uma carreira, numa determinada paragem:

- Deverá digitar C (espaço) Código da Paragem (espaço) Nº da Carreira e enviar para o 3599 (custo de mensagem escrita normal)

Fig.03- Explicação do sistema "SMS ao minuto"

E-mail ao minuto



O serviço e-mail ao minuto tem como principal objetivo proporcionar informação, via internet, sobre os horários reais de passagem dos veículos nas paragens. A pedido do Cliente, a CARRIS proporciona a consulta de informação fiável e de última hora sobre a passagem dos veículos, em cada paragem.

Como posso utilizar o serviço?

Utilizando o motor de pesquisa Carris:

- Aceda ao motor de pesquisa existente no topo da página e selecione a opção "Códigos de Paragem";
- Selecione a localidade onde se situa a paragem de que deseja obter informação;
- Verifique das opções que o sistema apresenta qual a paragem de que pretende obter informação e selecione a opção "mensagem";
- Preencha com o seu e-mail e dentro de breves momentos receberá no seu e-mail uma mensagem com os tempos de espera da paragem que solicitou informação.

Utilizando e-mail próprio:

- Consulte a lista de códigos e selecione o código da paragem que lhe interessa;
- Preencha o assunto de um e-mail novo com a configuração "C (espaço) código de paragem";
- Envie para sms@carris.pt e dentro de breves momentos deverá receber uma mensagem no seu e-mail indicando a informação pretendida.

Fig.04 - Explicação do sistema "Email ao minuto"

Assim sendo o output final desta aplicação deve ser um mail ou um

SMS preenchidos com o código da paragem e o transporte desejado.

2.iria ser **em Android** pela difusão do mesmo [2] , sendo que o Android dá ao developer essencialmente 5 métodos de gravação de dados [3] , mas só 3 eram minimamente utilizáveis neste caso, pela extensão dos dados 1.Ficheiro 2.Net 3.Base de Dados SQLite, pela complexidade da informação uma consulta ao 1. era muito demorada, devido ao facto de eu querer que esta aplicação fosse totalmente independente (ao máximo) elimina-se o 2.

3.**Base de Dados SQLite** foi a escolha de sistema de armazenamento de dados, 1º problema a minha fonte de dados é um Excel [no CD] , passar de para SQLite necessitou de alguma mestria e o uso de um programa externo

4.**SQLite Browser** (foi essencial usar este programa pois manejar os dados só pelo código android sendo estes tão extensos seria impossível) Agora era preciso que o Android aceitasse esta Base de Dados

5.**Como fazer display** da informação, inicialmente queria usar uma [4] Expandable List View graficamente seria muito agradável, mas levava a que tivesse de carregar quase todos os dados da base de dados previamente ou quando um item era carregado levando a um tempo de espera desagradavelmente longo, assim não podendo usar Expandable List View, decidi usar 2 [6] List Views distribuidas por 2 activitys (para poder fazer "Back")

6.**Como Listar a informação** Por Autocarro? , Por Paragem? Por Autocarro e muito mais "Limpo" já que existem menos Autocarros que paragens (e o ficheiro inicialmente encontrava-se já orientado nesse sentido) mas como existe uma funcionalidade algo escondida no serviço da Carris, se mandarmos o código da paragem sem nr do autocarro ele devolve os 3 primeiros a passar e isto é uma funcionalidade muito útil e se escolhermos o autocarro "3 primeiros" teríamos uma lista de paragens de qualquer maneira... e achei estranho pedir ao utilizador que escolhesse um não autocarro, já para não falar que o utilizador pode não saber que autocarro quer... assim preferi deixar o utilizador escolher uma outra

7.**Qual dos métodos de envio usar** inicialmente era minha intenção o envio de Email para o "Email ao minuto" ser o método preferido (caso haja rede, claro) mas depois de algumas utilizações do serviço apercebi-me que o envio de Mail apesar de "gratis" é inconstante tanto leva 2 segundos como 1 minuto (as vezes um pouco mais) o que fez com que repensasse a intenção de ser o default pois o envio de SMS custa o valor normal de SMS do serviço de telefone do utilizador logo por exemplo Zons , Vodafone, Meos desconta simplesmente um SMS da sua quota mensal, então decidi que devia ser o utilizador a decidir.

Método

Aqui relatarei como resolvi os problemas que me foram aparecendo no decorrer da elaboração desta aplicação começando por dizer que a minha experiência previa com Android é limitada

1º problema experiência Android limitada não é exatamente resolvível mas acho que agora está menos evidente.

2º problema como converter Excel para SQLite este foi difícil e não muito evidente

1. Usando o Excel gravar como .csv a tabela

2. Usando o SQLite Browser crie uma nova SQLite DB

3. Import table as .csv escolha o ficheiro anterior

4. Modifique a estrutura da tabela acrescentando uma coluna `_id`

5. corra o script SQL → CREATE TABLE android_metadata (locale TEXT);
INSERT INTO android_metadata VALUES ('en_US');

6. agora temos uma DB SQLite funcional em Android a partir de um ficheiro Excel

3º problema o ficheiro Excel tem 4467 linhas de dados ineficientes.

N.º Carreira	Sentido	N.º Ordem	Código Paragem	Paragens - Nome	Localização	Destino	Suburb
--------------	---------	-----------	----------------	-----------------	-------------	---------	--------

destes os que se encontram a branco são algo inúteis visto que se assim são removidas

N.º Carreira	N.º Ordem	Código Paragem	Paragens - Nome	Localização	Destino
--------------	-----------	----------------	-----------------	-------------	---------

N.º Carreira e Destino são muito úteis mas pork não juntar-los assim minimizando as necessidades de acesso a informação

N.º Carreira + Destino	N.º Ordem	Código Paragem	Paragens - Nome	Localização
------------------------	-----------	----------------	-----------------	-------------

Sendo que muitas paragens tem nomes iguais mas localizações diferentes assim resolvendo algumas destas incompatibilidades

N.º Carreira + Destino	N.º Ordem	Código Paragem	Paragens - Nome + Localização
------------------------	-----------	----------------	-------------------------------

Assim tenho os dados da tabela simplificados mas paragens + localização ainda não é único e assim acrescento uns poquinhos sobre a forma de [X] para que não haja 2 paragens com o mesmo nome

N.º Carreira + Destino	N.º Ordem	Código Paragem	Paragens - Nome + Localização + [X]
------------------------	-----------	----------------	-------------------------------------

Mantive o N.º de Ordem apesar de não ser usado na minha aplicação como future proofing ver (trabalhos futuros)

_id	autocarro	ordem	Código	paragem
-----	-----------	-------	--------	---------

Mas mesmo assim usando esta estrutura na procura inicial por Autocarro ou por Paragem ele tinha que correr as 4467 linhas para devolver 1 de todas as ocorrências não era eficiente logo decidi criar 2 outras tabelas, |autos| que contem 1 exemplar de todos os autocarros e o obrigatório `_id` e |stops| que contem `_id` as Paragens e os seus códigos

Resultados

Em todas as execuções desta aplicação obtive resultados positivos mas também me deparei com resultados inesperados:

1. Os tempos de resposta do serviço de Email são muito erráticos, entre 5s e 1 min o que foi inesperado.

2. Quando pedimos horários de um autocarro que já acabou ou ainda não começou (ocore devido aos Autocarros de Madrugada 200 e assim) obtemos resultados não encontrados como resposta ao pedido, quer por SMS quer por EMail

Conclusões e trabalho futuro

A aplicação está funcional e faz o que é suposto (passarei a usá-la no dia a dia) e como tal foi bem sucedida, existem alguns pormenores que por falta de tempo e/ou arte não foram abordados que teriam melhorado este trabalho sendo estes trabalho futuro tais como:

(1) Geo Localização das Paragens - Implicava acrescentar a Base de dados um coluna com a localização (física) da paragem provavelmente tirada do Google Maps e a seleção inicial de Paragem era pré filtrada as X mais próximas do utilizador.

(2) Lista de Favoritos - Lista de "procuras" preferidas dos utilizadores.

(3) Filtragem da Lista de Paragens - e incrivelmente extensa e a possibilidade de filtra-la era muito benéfico.

Com a (1) geo localização também podíamos:

(4) Dizer ao utilizador quando sair - quando a localização do utilizador estiver a chegar a localização do seu destino notificamos o utilizador que deve sair na próxima (temos sempre o congestionamento dos autocarros a contabilizar não podemos simplesmente dizer saia já daí ter mantido a ordem na BD).

(5) Ultimas Procuras - Lista de "procuras" com as ~5 ultimas procuras realizadas.

(6) Update da BD - atualmente para atualizar a Base de Dados é necessário apaga-la e correr novamente a aplicação, uma aplicação que seletivamente atualizava dados da base de dados era mais elegante.

Estas Funcionalidades estão hierarquizadas por o que julgo ser o seu valor acrescentado a aplicação

Assim sendo existem sempre melhorias que podem ser introduzidas a uma aplicação deste estilo

Gostaria de Agradecer ao Professor:

Alexandre Pereira

obrigado pelo apoio e direção e pelo facto de ter aceite um aluno como TFC que tecnicamente chegou tarde... e entregou mesmo em cima da hora em 2ª Fase, espero conseguir passar e justificar esse apoio e confiança.

Bibliografia

- [1] <http://carris.transporteslisboa.pt/pt/informacao-ao-passageiro/> consultado a 24/02/2016 as 13:14
- [2] <http://www.idc.com/prodserv/smartphone-os-market-share.jsp> consultado a 24/02/2016 as 13:00
- [3] <http://developer.android.com/guide/topics/data/data-storage.html> consultado a 24/02/2016 as 13:20
- [4] <http://developer.android.com/reference/android/widget/ExpandableListView.html> consultado a 24/02/2016 as 14:00
- [5] <http://developer.android.com/about/dashboards/index.html> consultado a 24/02/2016 as 15:00
- [6] <http://developer.android.com/guide/topics/ui/layout/listview.html> consultado a 24/02/2016 as 20:00
- [7] <http://carris.transporteslisboa.pt/pt/informacao-ao-passageiro/> consultado a 24/02/2016 as 16:00

Anexos (todo o source code do projecto deve ser colocado aqui)

No CD entregue junto com este relatorio encontra-se:

CD>

Este relatório em formato .pdf

<Aplicação KTF>

<Anexos>

CD>Aplicação KTF

pastas da aplicação Kquanto Tempo Falta (KTF_iota)

CD>Anexos>

<Base de Dados>

<Excel>

<Imagens Usadas no Projecto>

CD>Anexos>Base de Dados

ktf.sqlite3 (base de dados do projecto SQLite)

CD>Anexos>Excel

Excel Original.xls (Excel original antes de modificações)
<csv>

CD>Anexos>Excel>csv

Ficheiros .csv que foram usados para criar as tabelas da BD

CD>Anexos>Imagens Usadas no Projecto

imagens usadas na aplicação

Source Code

<Layouts.xml>

<activity_main_menu.xml>

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
tools:context=".MainMenu"
    android:background="@android:color/holo_blue_dark">

    <TextView android:text="@string/main_menu"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:editable="false"
        android:textSize="30dp"
        android:id="@+id/main_menu_text1"
        android:layout_alignParentEnd="true"
        android:textColor="#ff000000"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Autocarro"
        android:id="@+id/main_btn1"
        android:drawableLeft="@drawable/by_bus"
        android:drawableRight="@drawable/by_bus"
        android:layout_below="@+id/main_menu_text1"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_marginTop="89dp"
        android:textSize="30dp"
        android:elegantTextHeight="false"
        android:focusableInTouchMode="false" />

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Paragem"
        android:id="@+id/main_btn2"
```

```
        android:drawableLeft="@drawable/by_stop"
        android:drawableRight="@drawable/by_stop"
        android:textSize="30dp"
        android:layout_marginTop="106dp"
        android:layout_below="@+id/main_btn1"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true" />
```

```
</RelativeLayout>
```

<activity_by_bus.xml>

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <EditText
        android:layout_width="match_parent"
        android:background="@android:color/holo_blue_dark"
        android:layout_height="wrap_content"
        android:id="@+id/bybus_lb1"
        android:text="@string/by_bus"
        android:textIsSelectable="false"
        android:editable="false"
        android:linksClickable="false"
        android:longClickable="false" />

    <ListView
        android:id="@android:id/list"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent">
    </ListView>

</LinearLayout>
```

<activity_by_stop.xml>

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/bystop_lb1"
        android:text="@string/by_stop"
        android:editable="false"
```

```
        android:focusable="false"
        android:textAlignment="center"
        android:autoText="false"
        android:background="@android:color/holo_blue_dark" />
```

```
<ListView
    android:id="@android:id/list"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
</ListView>
```

```
</LinearLayout>
```

<activity_by_other.xml>

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/other_lbl1"
        android:textIsSelectable="false"
        android:editable="false"
        android:linksClickable="false"
        android:longClickable="false"
        android:nestedScrollingEnabled="false"
        android:background="@android:color/holo_blue_dark" />

    <ListView
        android:id="@android:id/list"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent">
    </ListView>

</LinearLayout>
```

<activity_rede.xml>

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:background="@android:color/holo_blue_dark"
    android:layout_height="match_parent"
    android:weightSum="1">

    <CheckBox
```



```
android:layout_width="350dp"
android:layout_height="wrap_content"
android:text="@string/setworkstatus"
android:textColor="#ff000000"
android:id="@+id/checkbox"
android:clickable="false"
android:layout_weight="0.07"
android:textSize="30dp"
android:layout_alignParentRight="true"
android:layout_alignParentEnd="true"
android:layout_alignParentLeft="true"
android:layout_alignParentStart="true"
android:checked="false"
android:background="@color/button_material_light" />
```

<Button

```
android:layout_width="100dp"
android:layout_height="wrap_content"
android:text="Send E-Mail"
android:id="@+id/button"
android:layout_gravity="center_horizontal"
android:drawableTop="@drawable/email"
android:layout_weight="0.07"
android:layout_marginTop="89dp"
android:layout_below="@+id/checkbox"
android:layout_centerHorizontal="true" />
```

<Button

```
android:layout_width="100dp"
android:layout_height="wrap_content"
android:text="Send SMS"
android:id="@+id/button2"
android:layout_gravity="center_horizontal"
android:drawableTop="@drawable/sms"
android:layout_weight="0.07"
android:layout_marginTop="84dp"
android:layout_below="@+id/button"
android:layout_alignLeft="@+id/button"
android:layout_alignStart="@+id/button"
android:layout_alignRight="@+id/button"
android:layout_alignEnd="@+id/button" />
```

<Button

```
style="?android:attr/buttonStyleSmall"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:drawableEnd="@drawable/refresh"
android:id="@+id/button3"
android:layout_alignParentBottom="true"
android:layout_alignParentRight="true"
android:layout_alignParentEnd="true" />
```

</RelativeLayout>

<Class.java>

<MainMenu.java>

```
package com.john.ktf_iota;
```

```
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
```

```
public class MainMenu extends Activity {
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main_menu);
```

```
        Button bybus = (Button) findViewById(R.id.main_btn1);
        Button bystop = (Button) findViewById(R.id.main_btn2);
```

```
        bybus.setOnClickListener(new View.OnClickListener() {
```

```
            public void onClick(View arg0) {
```

```
                //Starting a new Intent
```

```
                Intent bybus = new Intent(getApplicationContext(),
ByBus.class);
```

```
                Log.e("n", "bybus");
```

```
                startActivity(bybus);
```

```
            }
```

```
        });
```

```
        bystop.setOnClickListener(new View.OnClickListener() {
```

```
            public void onClick(View arg0) {
```

```
                //Starting a new Intent
```

```
                Intent bystop = new
Intent(getApplicationContext(), ByStop.class);
```

```
                Log.e("n", "byStop");
```

```
                startActivity(bystop);
```

```

        }

        });

    }
}

```

<ByBus.java>

```

package com.john.ktf_iota;

import android.app.Activity;
import android.app.ListActivity;
import android.content.Intent;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

import java.util.ArrayList;

/**
 * Created by John on 21/02/2016.
 */

public class ByBus extends ListActivity {

    //Db a consultar
    private static final String DB_NAME = "ktf2.sqlite3";
    //nome da tabela
    private static final String TABLE_NAME = "autos";
    //Colunas Relevantes
    private static final String AUTOS_ID = "_id";
    private static final String AUTOS_NUMERO = "autoc";

    private SQLiteDatabase database;
    private ListView listView;
    private ArrayList<String> bus;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_by_bus);

        //invocação do helper
        DbExternalHelper dbOpenHelper = new DbExternalHelper(this,

```

```

DB_NAME);
    database = dbOpenHelper.openDataBase();
    // A base de dados ja foi aberta aqui
    getBuses();
    setUpList();

}

private void setUpList() {
    //Usamos um adaptador standard e um layout
    setListAdapter(new ArrayAdapter<String>(this,
        android.R.layout.simple_list_item_1, bus));
    listView = getListView();

    listView.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
        public void onItemClick(AdapterView<?> parent, View
view, int position, long id) {

            //Invoca a Lista de Paragens
            Intent ByOther = new
Intent(getApplicationContext(), ByOther.class);
            //Dados para a Nova Actividade
            ByOther.putExtra("a", ((TextView)
view).getText().toString());
            ByOther.putExtra("from", "bybus");

            startActivity(ByOther);

        }
    });
}

//Get dos Autocarros
private void getBuses() {
    bus = new ArrayList<String>();
    Cursor aCursor = database.query(TABLE_NAME, new String[] {
AUTOS_ID, AUTOS_NUMERO}, null, null, null, null, AUTOS_NUMERO,
null );
    aCursor.moveToFirst();
    if(!aCursor.isAfterLast()) {
        do {

            String name = aCursor.getString(1);
            bus.add(name);

        } while (aCursor.moveToNext());
    }
    aCursor.close();
}

```

<ByStop.java>

```
package com.john.ktf_iota;

import android.app.Activity;
import android.app.ListActivity;
import android.content.Intent;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.Adapter;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

import java.util.ArrayList;

/**
 * Created by John on 21/02/2016.
 */

public class ByStop extends ListActivity {

    //Db a consultar
    private static final String DB_NAME = "ktf2.sqlite3";
    //nome da tabela
    private static final String TABLE_NAME = "stops";
    //Colunas Relevantes
    private static final String STOPS_ID = "_id";
    private static final String STOPS_PARAGEM = "paragem";
    private static final String STOPS_CODE = "cod";

    private SQLiteDatabase database;
    private ListView listView;
    private ArrayList<String> bus;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_by_stop);

        //invocação do helper
        DbExternalHelper dbOpenHelper = new DbExternalHelper(this,
DB_NAME);
        database = dbOpenHelper.openDataBase();
        // A base de dados ja foi aberta aqui

        getBuses();
        setUpList();
    }
}
```

```

    }

    private void setUpList() {
        //Usamos um adaptador standard e um layout

        setListAdapter(new ArrayAdapter<String>(this,
            android.R.layout.simple_list_item_1, bus));

        listView = getListView();

        //Mensagem mostarda ao click
        listView.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
            public void onItemClick(AdapterView<?> parent, View
view, int position, long id) {

                //Invoca a Lista de Paragens
                Intent ByOther = new
Intent(getApplicationContext(), ByOther.class);
                //Dados para a Nova Actividade
                ByOther.putExtra("a", ((TextView)
view).getText().toString());
                ByOther.putExtra("from", "bystop");

                startActivity(ByOther);

            }
        });
    }

    //Fetcha as Stops
    private void getBuses() {
        bus = new ArrayList<String>();
        Cursor aCursor = database.query(TABLE_NAME, new String[] {
STOPS_ID, STOPS_PARAGEM, STOPS_CODE}, null, null, null, null,
STOPS_ID, null );
        aCursor.moveToFirst();
        if(!aCursor.isAfterLast()) {
            do {

                String name = aCursor.getString(1);
                bus.add(name);

            } while (aCursor.moveToNext());
        }
        aCursor.close();
    }
}

```

<ByOther.java>

```
package com.john.ktf_iota;

import android.app.ListActivity;
import android.content.Intent;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.TextView;

import java.util.ArrayList;

/**
 * Created by John on 23/02/2016.
 */
public class ByOther extends ListActivity {

    //Nome da BD
    private static final String DB_NAME = "ktf2.sqlite3";
    // Nome da Tabela
    private static final String TABLE_NAME = "carreiras";
    //nome dos Elementos Relevantes
    private static final String CARREIRAS_ID = "_id";
    private static final String CARREIRAS_NUMERO = "autocarro";
    private static final String CARREIRAS_CODIGO = "codigo";
    private static final String CARREIRAS_PARAGEM = "paragem";

    private SQLiteDatabase database;
    private ListView listView;
    private ArrayList<String> display;
    private String bus;
    private String autocarro;
    private String stop;
    private String paragem;
    private String codigo;
    private String from;

    EditText txt1;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_by_other);

        Intent i = getIntent();
        // Receiving the Data
```

```

        from = i.getStringExtra("from");
        bus = i.getStringExtra("a");
        stop = i.getStringExtra("a");
/*
        Toast.makeText(getApplicationContext(),
                        bus,
                        Toast.LENGTH_SHORT).show();
*/
        //invocação do helper
        DbExternalHelper dbOpenHelper = new DbExternalHelper(this,
DB_NAME);
        database = dbOpenHelper.openDataBase();
        // A base de dados ja foi aberta aqui

        txt1 = ((EditText) findViewById(R.id.other_lb1));

        if(from.equals("bybus")) {

            txt1.setText(bus);

            display = getStops(bus);
            setUpList();

        }else if(from.equals("bystop")){

            txt1.setText(stop);

            display = getBuses(stop);
            setUpList();

        }
    }

    private void setUpList() {
        //Usamos um adaptador standard e um layout
        setListAdapter(new ArrayAdapter<String>(this,
            android.R.layout.simple_list_item_1, display));
        listView = getListView();

        //Mensagem mostarda ao click
        listView.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
            public void onItemClick(AdapterView<?> parent, View
view, int position,long id) {

                //Como fundi as colunas nrautocarro e paragem
                Destino tive que cortar a string a 3 chares
                //String tres = bus.substring(0,
                Math.min(bus.length(), 3));

```



```

        if(from.equals("bybus")) {

            paragem = ((TextView)
view).getText().toString();
            codigo = "C " + getCodigo(bus, paragem);

            //Como fundi as colunas nrautocarro e paragem
Destino tive que cortar a string a 3 chares
            autocarro = bus.substring(0,
Math.min(bus.length(), 3));

        }else if(from.equals("bystop")){

            paragem = stop;
            codigo = "C " + getCodigo(paragem);

            String a = ((TextView)
view).getText().toString();

            //Como fundi as colunas nrautocarro e paragem
Destino tive que cortar a string a 3 chares
            autocarro = a.substring(0,
Math.min(a.length(), 3));

        }

        //Invoca a Rede
Intent Rede = new Intent(getApplicationContext(),
Rede.class);

        //Dados para a Nova Actividade
Rede.putExtra("bus", autocarro );
Rede.putExtra("stop", codigo );

        startActivity(Rede);

        //Toast.makeText(getApplicationContext(), co ,
Toast.LENGTH_SHORT).show();
        //SEND BY STUFF

    }

    });
}

```

```

//-----ByBus-----//

//Popula a lista com as Pragens do Autocarro X recebe ( Nr. do
Autocarro )
public ArrayList getStops ( String i ) {
    ArrayList a = new ArrayList<String>();
    Cursor stopCursor = database.query(TABLE_NAME, new
String[] { CARREIRAS_ID, CARREIRAS_NUMERO, CARREIRAS_CODIGO,
CARREIRAS_PARAGEM }, CARREIRAS_NUMERO + "=?", new String[]{i} ,
null , null, CARREIRAS_ID , null );
    stopCursor.moveToFirst();
    if(!stopCursor.isAfterLast()) {
        do {

            String name = stopCursor.getString(3);

            a.add(name);

        } while (stopCursor.moveToNext());
    }
    stopCursor.close();
    return a;
}

//Determina o codigo da Paragem recebe ( Nr. do Autocarro ,
Nome da Paragem )
public String getCodigo( String a , String b ) {

    Cursor bCursor = database.query(TABLE_NAME, new String[] {
CARREIRAS_ID, CARREIRAS_NUMERO, CARREIRAS_CODIGO,
CARREIRAS_PARAGEM}, CARREIRAS_NUMERO + "=? " + " AND " +
CARREIRAS_PARAGEM + "=?"
        ,new String[]{a,b} , null , null, CARREIRAS_ID ,
"1" );
    bCursor.moveToFirst();
    if(!bCursor.isAfterLast()) {
        do {

            codigo = bCursor.getString(2);

            //      cod.add(codigo);
        } while (bCursor.moveToNext());
    }
    bCursor.close();
    return codigo; }

//-----ByStop-----//

//Lista os Autocarros da Paragem X
public ArrayList getBuses ( String i ) {

```

```

        ArrayList a = new ArrayList<String>();
        Cursor stopCursor = database.query(TABLE_NAME, new
String[] { CARREIRAS_ID, CARREIRAS_CODIGO, CARREIRAS_NUMERO,
CARREIRAS_PARAGEM}, CARREIRAS_PARAGEM + "=?", new String[]{i} ,
null , null, CARREIRAS_ID , null );
        stopCursor.moveToFirst();
        if(!stopCursor.isAfterLast()) {
            do {

                String name = stopCursor.getString(2);

                a.add(name);

            } while (stopCursor.moveToNext());
            a.add("Os Primeiros 3");
        }
        stopCursor.close();
        return a;
    }

    //Atribui a "codigo" o Valor doCodigo da Paragem X
    public String getCodigo ( String i ) {
        Cursor dCursor = database.query(TABLE_NAME, new String[] {
CARREIRAS_ID, CARREIRAS_CODIGO, CARREIRAS_NUMERO,
CARREIRAS_PARAGEM}, CARREIRAS_PARAGEM + "=?", new String[]{i} ,
null , null, CARREIRAS_ID , "1" );
        dCursor.moveToFirst();
        if(!dCursor.isAfterLast()) {
            do {

                codigo = dCursor.getString(1);

            } while (dCursor.moveToNext());

        }
        dCursor.close();
        return codigo;
    }

}

```

<Rede.java>

```

package com.john.ktf_iota;

import android.app.Activity;
import android.content.Context;

```

```

import android.content.Intent;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.RadioButton;

/**
 * Created by John on 23/02/2016.
 */
public class Rede extends Activity {

    private String c = "";
    private String b = "";
    private String codigo = "";

    private RadioButton rbul;

    private CheckBox checkBox;

    private Button btn1;
    private Button btn2;
    private Button btn3;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_rede);

        btn1 = ((Button) findViewById(R.id.button2));
        btn2 = ((Button) findViewById(R.id.button));
        btn3 = ((Button) findViewById(R.id.button3));

        checkBox = ((CheckBox) findViewById(R.id.checkBox));

        checkBox.setChecked(false);
        checkBox.setText("A Rede Está Indesponível");

        Intent i = getIntent();
        // Receiving the Data
        b = i.getStringExtra("bus");
        c = i.getStringExtra("stop");

        btn2.setEnabled(false);

        CheckOnline();
    }

```

```

btn1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        SenderSMS(c, b);

    }
});

btn2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        SenderMail(c, b);

    }
});

btn3.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        CheckOnline();

    }
});

}

public void CheckOnline() {

    ConnectivityManager conectmnager = (ConnectivityManager)
getSystemService(Context.CONNECTIVITY_SERVICE);
    NetworkInfo ninfo = conectmnager.getActiveNetworkInfo();

    if (ninfo != null && ninfo.isConnected()) {

        checkBox.setChecked(true);
        checkBox.setText("A Rede Está Funcional");

        btn2.setEnabled(true);
        btn2.setText("Send EMail");

    }else {

        btn2.setText("Não Há Rede");
        btn2.setEnabled(false);

        checkBox.setChecked(false);
        checkBox.setText("A Rede Está Desligada");

    }
}

```

```

}

private void SenderMail(String c, String b ){

    String to = "sms@carriis.pt";
    String subject = c + " " + b;

    Intent email = new Intent(Intent.ACTION_SEND);
    email.putExtra(Intent.EXTRA_EMAIL, new String[]{ to});
    email.putExtra(Intent.EXTRA_SUBJECT, subject);

    //need this to prompts email client only
    email.setType("message/rfc822");

    startActivity(Intent.createChooser(email, "Choose an Email
client :"));

}

private void SenderSMS (String c,String b){

    String phoneNo = "3599";
    String message = c + " " + b;

    Intent smsIntent = new Intent(Intent.ACTION_VIEW);
    smsIntent.setType("vnd.android-dir/mms-sms");
    smsIntent.putExtra("address", phoneNo);
    smsIntent.putExtra("sms_body",message);
    startActivity(smsIntent);

}

private void Refresh (){

}

}

```

<DbExternalHelper>

```

package com.john.ktf_iota;

import java.io.FileOutputStream;
import java.io.IOException;

```

```

import java.io.InputStream;
import java.io.OutputStream;

import android.content.Context;
import android.database.SQLException;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Log;

/**
 * Created by John on 21/02/2016.
 */

public class DbExternalHelper extends SQLiteOpenHelper {

    //Caminho para o dispositivo com a DB
    public static String DB_PATH;

    //Nome do ficheiro da DB
    public static String DB_NAME;
    public SQLiteDatabase database;
    public final Context context;

    public SQLiteDatabase getDb() {
        return database;
    }

    public DbExternalHelper(Context context, String databaseName)
    {
        super(context, databaseName, null, 1);
        this.context = context;
        //Caminho para a DB da aplicação
        String packageName = context.getPackageName();
        DB_PATH = String.format("//data//data//%s//databases//",
packageName);
        DB_NAME = databaseName;
        openDataBase();
    }

    //Cria a DB se já não o Estiver
    public void createDataBase() {
        boolean dbExist = checkDataBase();
        if (!dbExist) {
            this.getReadableDatabase();
            try {
                copyDataBase();
            } catch (IOException e) {
                Log.e(this.getClass().toString(), "Copying
error");
                throw new Error("Error copying database!");
            }
        } else {
            Log.i(this.getClass().toString(), "Database already

```

```

exists");
    }
}

//Verifica a existencia da DB
private boolean checkDataBase() {
    SQLiteDatabase checkDb = null;
    try {
        String path = DB_PATH + DB_NAME;
        checkDb = SQLiteDatabase.openDatabase(path, null,
SQLiteDatabase.OPEN_READONLY);
    } catch (SQLException e) {
        Log.e(this.getClass().toString(), "Error while
checking db");
    }

    // e Preciso fechar a a DB ou o android rebenta
    if (checkDb != null) {
        checkDb.close();
    }
    return checkDb != null;
}

//Para Copiar a DB externa para a Interna
private void copyDataBase() throws IOException {
    //Para Ler a DB
    //Não esquecer de por a DB Externa na pasta assets
    InputStream externalDbStream =
context.getAssets().open(DB_NAME);

    //Caminho para a DB virgem
    String outFileName = DB_PATH + DB_NAME;

    //Escreve na DB virem todos os bytes
    OutputStream localDbStream = new
FileOutputStream(outFileName);

    //Copia Byte a byte
    byte[] buffer = new byte[1024];
    int bytesRead;
    while ((bytesRead = externalDbStream.read(buffer)) > 0) {
        localDbStream.write(buffer, 0, bytesRead);
    }
    //Fechar ou da caca
    localDbStream.close();
    externalDbStream.close();
}

public SQLiteDatabase openDataBase() throws SQLException {
    String path = DB_PATH + DB_NAME;
    if (database == null) {
        createDataBase();
        database = SQLiteDatabase.openDatabase(path, null,

```



```
        SQLiteDatabase.OPEN_READWRITE);  
    }  
    return database;  
}
```

```
@Override  
public synchronized void close() {  
    if (database != null) {  
        database.close();  
    }  
    super.close();  
}  
@Override  
public void onCreate(SQLiteDatabase db) {}  
@Override  
public void onUpgrade(SQLiteDatabase db, int oldVersion, int  
newVersion) {}  
}
```

A1. Anexo de Manual técnico da aplicação (se existir)

Atualização á base de dados - Para fazer isto presumivelemnete partindo novamente de um ficheiro excel temos que:
da aplicação temos que desinstalar a aplicação do dispositivo para que este apague a Base de Dados SQLite e a qua

A2. Anexo de Manual de utilizador da aplicação (se existir)

Kuanto Tempo Falta (KTF) Manual do Utilizador

