

## **ECATI - DEISI**

### **Licenciatura em Informática de Gestão**

# **Aplicação móvel para Aconselhamento Nutricional a Gestantes**

Projecto Final de Licenciatura

Elaborado por: Ana Maria das Mercês Almeida da Silva

Aluna nº: a21105967      Turma: 3D1

Orientador: Professor Pedro Malta

Lisboa, Julho de 2015

Solicitado pelo curso de Ciências da Nutrição da ULHT

## **Agradecimentos**

Ao professor Pedro Malta, pela sua orientação, disponibilidade, motivação e apoio que me proporcionou durante a realização deste trabalho.

Aos responsáveis do Curso de Ciências da Nutrição: Dra Maria Costa e Professor Nelson Tavares, pelo tema proporcionado.

Aos meus familiares e amigos, em especial meu cunhado prof. Dr. Lourenço Joaquim da Costa do Rosário, por todo o apoio e incentivo.

E por último e em especial ao meu marido e às minhas filhas por tudo.

## RESUMO

Muitas empresas, buscam soluções de tecnologia móvel para melhorar os seus negócios, pois permite-lhes aumentar a eficiência, a qualidade de serviços e a produtividade.

A tecnologia móvel revolucionou o quotidiano das pessoas fazendo parte activa da vida da maioria, influenciando e modificando suas rotinas, o modo de vida e formas de tomar decisões, desde questões domésticas, académicas ou profissionais, entre outras.

Hoje em dia a maioria das pessoas não vive sem celular, estando deste modo conectados e disponíveis a qualquer hora e em qualquer lugar.

Assim os dispositivos móveis são a forma mais rápida e eficaz para circular informação e abranger um maior número de potenciais clientes.

Este trabalho apresenta o protótipo de uma solução de aplicação móvel para aconselhamento nutricional a gestantes, em resposta a solicitação da faculdade de ciências da nutrição desta universidade, com base nas informações já existentes no *site* da mesma.

## ABSTRACT

*Many companies looking for a mobile technology solutions to improve their business, as it allows them to increase the efficiency, quality of service and productivity.*

*Mobile technology has revolutionized the daily lives of people doing active part of most of life, influencing and changing their routines, the way of life and ways of making decisions, from domestic issues, academic or professional, among others.*

*Today most people cannot live without cellphone, this being connected mode and available anytime and anywhere.*

*So mobile devices are the fastest and most effective way to circulate information and cover a larger number of potential customers.*

*This paper presents the prototype of a mobile application solution for nutrition counseling to pregnant women, in response to request from the college of nutrition sciences at this university, based on existing information on the website of the same.*

## ÍNDICE

RESUMO.....	3
ABSTRACT.....	4
ÍNDICE.....	5
ÍNDICE DE FIGURAS.....	6
<b>1. INTRODUÇÃO.....</b>	<b>7</b>
1.1. OBJECTIVOS.....	8
1.2. PÚBLICO ALVO .....	8
1.3. DEFINIÇÕES, ACRÓNIMOS E ABREVIATURAS .....	8
1.4. ESTRUTURA DO DOCUMENTO .....	9
<b>2. DESCRIÇÃO DO PRODUTO .....</b>	<b>10</b>
2.1. CONTEXTO .....	10
<b>3. DESENVOLVIMENTO DO PROTÓTIPO.....</b>	<b>11</b>
3.1. DESCRIÇÃO DOS CASOS DE USO E ACTIVIDADES.....	11
3.2. DESCRIÇÃO DAS CLASSES.....	12
3.2.1. DIAGRAMA DE CLASSES.....	13
3.2.2. DIAGRAMA DE SEQUÊNCIA .....	13
3.3. IMPLEMENTAÇÃO E MODELO DE DADOS .....	14
3.3.1. DIAGRAMA UML DE IMPLEMENTAÇÃO.....	14
3.3.2. MODELO DE DADOS .....	15
3.4. ESCOLHA DO SISTEMA .....	16
3.5. TECNOLOGIAS DE DESENVOLVIMENTO .....	16
3.5.1. ANDROID STUDIO .....	16
3.6. TÉCNICAS DE PROGRAMAÇÃO USADOS .....	17
3.6.1. ACTION BAR, TABS AND FRAGMENTS .....	17
3.6.2. ACTIVITIES E INTENTS.....	19
3.6.3. ANDROID MANIFEST .....	23
3.6.4. CALCULADORA DE PESO .....	24
3.6.5. SQLITE .....	28
<b>4. CONCLUSÕES.....</b>	<b>30</b>
4.1. MELHORIAS PARA O FUTURO .....	30

5. BIBLIOGRAFIA .....	31
-----------------------	----

## ÍNDICE DE FIGURAS

Figura 1: Diagrama Caso de uso geral da aplicação .....	11
Figura 2: Diagrama de Actividade.....	12
Figura 3: Diagrama de Classes .....	13
Figura 4: Diagrama de Sequência_Login.....	13
Figura 5: Diagrama de Sequência_calculadora .....	14
Figura 6: Diagrama de Implementação .....	15
Figura 7: Modelo de Dados.....	15
Figura 8: Action Bar e Tabs .....	17
Figura 9: Codigo criação tabs e fragmentos.....	18
Figura 10: Código chamada da tab .....	18
Figura 11: Exemplo de uma activity.....	19
Figura 12: Ciclo de vida de uma activity .....	20
Figura 13: Interacção Intents com Activities.....	21
Figura 14: Uso de Intents em FAQ'S fragemnt Tab.....	21
Figura 15: ficheiro xml da pergunta 2.....	22
Figura 16: Ficheiro java da pergunta 2 .....	22
Figura 17: Exemplo do resultado do código gerado em FAQ'S .....	23
Figura 18: Ficheiro Android Manifest .....	24
Figura 19: Codigo do painel calculadora peso .....	25
Figura 20: Painel Calculadora de Peso .....	25
Figura 21: Codigo do calculo do peso ideal .....	27
Figura 22: Resultados Caluladora.....	27
Figura 23: Grafico exemplo .....	28
Figura 24: Tabela do SQLite.....	29
Figura 25: Codigo SQLite.....	29

## 1. INTRODUÇÃO

A procura de maior eficiência em processos e negócios, faz com que muitas empresas e instituições adotem a tecnologia móvel como meio de fazer chegar informação ao maior número possível de utilizadores:- o seu público-alvo.

A tecnologia móvel de informações e comunicação inclui infra-estruturas tecnológicas para a conectividade tais como *wi-fi*, dados móveis, *Bluetooth* e equipamentos móveis como *smartphones*, telemóveis, *Notebooks* e *tabletes*, cada um dos quais com sistemas operativos e um leque variado de aplicações, específicas a cada modelo, e com diferentes e variadas funcionalidades.

As empresas e instituições que adaptam a tecnologia móvel nos seus processos de negócios, permitem que um maior número de utilizadores aceda à informação a qualquer hora e em qualquer lugar.

É com este objectivo que a direcção da faculdade de Ciências da Nutrição, após ter criado um *site* com conteúdos para aconselhamento nutricional a gestantes, propôs-se a obter para o mesmo fim, uma aplicação móvel.

O aconselhamento nutricional a gestantes é de extrema importância, pois permite que as mulheres grávidas tenham a informação necessária para uma gravidez saudável.

Para uma gravidez saudável, é fundamental adoptar uma alimentação equilibrada, evitando problemas sérios, como o diabetes gestacional.

*Alimentação equilibrada* ➡ *bebé e mãe saudáveis.*

A gravidez é um momento para fazer escolhas saudáveis relativas à alimentação, a fim de ganhar somente o peso necessário para a mãe e para o desenvolvimento do bebé, e não uma época para fazer dietas restritas e nem pensar em perder peso. Por outro lado, também não significa comer em excesso ou “comer por dois”.

Durante os nove meses, há necessidade de as mulheres grávidas seguirem os conselhos nutricionais para uma dieta balanceada em cada momento da sua gestação, avaliando e controlando os efeitos, por meio de vários mecanismos de controle, um dos quais iremos disponibilizar através da aplicação a desenvolver, objecto deste trabalho.

A aplicação móvel terá duas componentes: uma “tab” com perguntas frequentes das gestantes e respectivas respostas e a outra “tab” com uma calculadora que permite saber a evolução do peso em cada fase da gravidez.

A calculadora foi baseada no ICM (índice de massa corporal) antes da gravidez e na tabela para o ganho de peso durante a gravidez de um bebé, recomendada pelo *Programa Harvard Medical School Portugal – hmsportugal*.



### 1.1. OBJECTIVOS

Desenvolver uma aplicação móvel para aconselhamento nutricional a gestantes, tendo como base os conteúdos do site <http://nutricao.ulusoфона.pt/> possibilitando deste modo às mulheres grávidas um controlo mais próximo e eficaz para uma gestação saudável.

### 1.2. PÚBLICO ALVO

Mulheres grávidas, mulheres que pretendam engravidar, ou qualquer outra pessoa adulta que pretenda informação nutricional com vista a apoiar às gestantes a terem uma gestação feliz tanto para a mãe como para o bebé, evitando problemas ou doenças derivadas duma má alimentação, baixo peso ou excesso de peso durante a gravidez.

### 1.3. DEFINIÇÕES, ACRÓNIMOS E ABREVIATURAS

Termo / Sigla	Descrição
<b>ECATI</b>	Escola de Comunicação, Arte e Tecnologias de Informação
<b>DEISI</b>	Departamento de Engenharia Informática e Sistemas de Informação.
<b>LIG</b>	Licenciatura em Informática de Gestão
<b>ULHT</b>	Universidade Lusófona de Humanidades e Tecnologias
<b>Site ou website</b>	Também chamado em Portugal de sítio eletrónico/web/da internet, é um conjunto de páginas web, isto é, de hipertextos acessíveis geralmente pelo protocolo HTTP na internet.
<b>Diagrama de Use Cases</b>	Diagrama de casos de utilização: É uma das técnicas em UML para representar o levantamento de requisitos de um sistema. Representa a sequência de acções executadas pelo sistema, que geram resultado.
<b>Diagrama de Classes</b>	Linguagem UML para desenvolvimento orientado por Objectos. Descrição formal da estrutura de objectos num sistema. Descreve, para cada objecto, sua identidade, relacionamentos, atributos e operações.
<b>Diagrama de Actividades</b>	Descreve fluxos de trabalho numa organização e detalha operações de uma classe.
<b>IDE</b>	Integrated Development Environment
<b>SDK</b>	Software Development Kit
<b>BB ou db</b>	Base de dados

Tabela 1: Definições, Acrónimos e Abreviaturas

#### 1.4. ESTRUTURA DO DOCUMENTO

Para além deste capítulo onde consta a apresentação do trabalho e objectivos, para melhor compreensão da motivação para o desenvolvimento do aplicativo, este relatório encontra-se organizado da seguinte forma:

- Capítulo 2: descrição do produto, contexto e diagramas de forma a especificar os processos e definir os requisitos do sistema a implementar.
- Capítulo 3: desenvolvimento do protótipo. O sistema a implementar, a estrutura de dados que suporta todo o sistema e a tecnologia de desenvolvimento.
- Capítulo 4: conclusões, considerações finais e propostas para melhoria futura do protótipo.

## 2. DESCRIÇÃO DO PRODUTO

O produto é o protótipo de uma aplicação móvel para telemóveis, *smartphones* ou *tabletes*. O protótipo actual foi desenvolvido para telemóveis andróide e será posteriormente adaptado a outros sistemas operativos.

O aplicativo desenvolvido de forma simples e prática, visa fornecer informação nutricional a gestantes. Para tal, contém duas “tabs” das quais a primeira com o nome “FAQ’S” apresenta perguntas que normalmente são as principais preocupações das mulheres grávidas, em que ao seleccionar cada uma apresenta a resposta correspondente, e a segunda tab “Calculadora de peso” permite verificar se a gestante está ou não com o peso ideal em diferentes fases da gravidez.

Existe a possibilidade de o utilizador cadastrar-se no aplicativo criando o seu registo para uma análise posterior.

Em qualquer altura da utilização do aplicativo, é possível aceder ao *site* <http://nutricao.uluso.fona.pt/> através do botão *home*.

### 2.1. CONTEXTO

O Desenvolvimento da aplicação móvel "Nutrição Gestantes", teve como base a necessidade de fornecer às gestantes, informação nutricional útil para uma gestação saudável e feliz para a mãe e para o bebé, de forma eficaz podendo ser acedida em qualquer lugar e a qualquer hora.

Deste modo, a aplicação deve possibilitar *consultar os conselhos nutricionais, e calcular o peso durante os nove meses de gestação*, estabelecendo a devida comparação com peso considerado “ideal”, em cada fase da gravidez.

### 3. DESENVOLVIMENTO DO PROTÓTIPO

Para uma melhor percepção do que se pretende desenvolver, criamos diversos diagramas em *UML (Unified Modeling Language)*, tais como diagrama de casos de uso, diagrama de actividades, diagrama de classes, diagrama de sequência, diagrama de implementação e modelo de dados, que melhor descrevem as necessidades e objectivos da aplicação, e servem de suporte ao seu desenvolvimento.

#### 3.1. DESCRIÇÃO DOS CASOS DE USO E ACTIVIDADES

A aplicação é composta por duas TAB'S e correspondentes Fragmentos, conforme o diagrama de caso de uso geral do sistema exemplificado na figura 1:

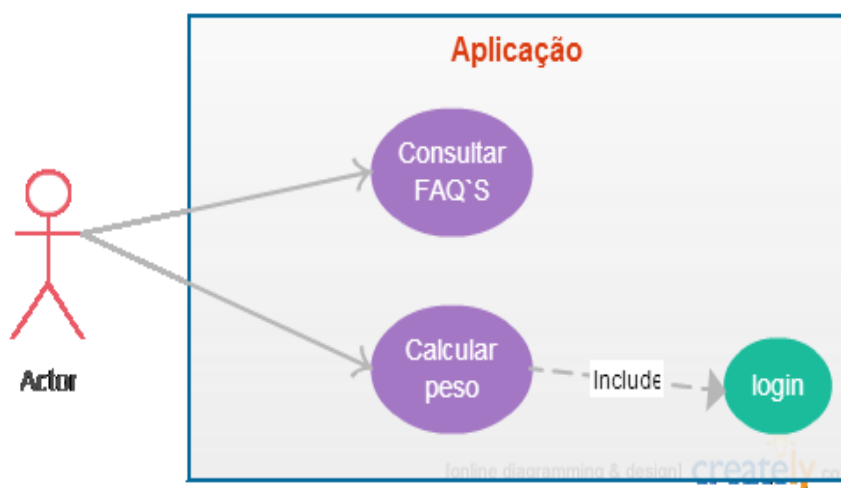


Figura 1: Diagrama Caso de uso geral da aplicação

O actor representa os utilizadores da aplicação, que podem ser mulheres grávidas, mulheres que pretendam engravidar ou qualquer pessoa que procure aconselhamento nutricional a gestantes.

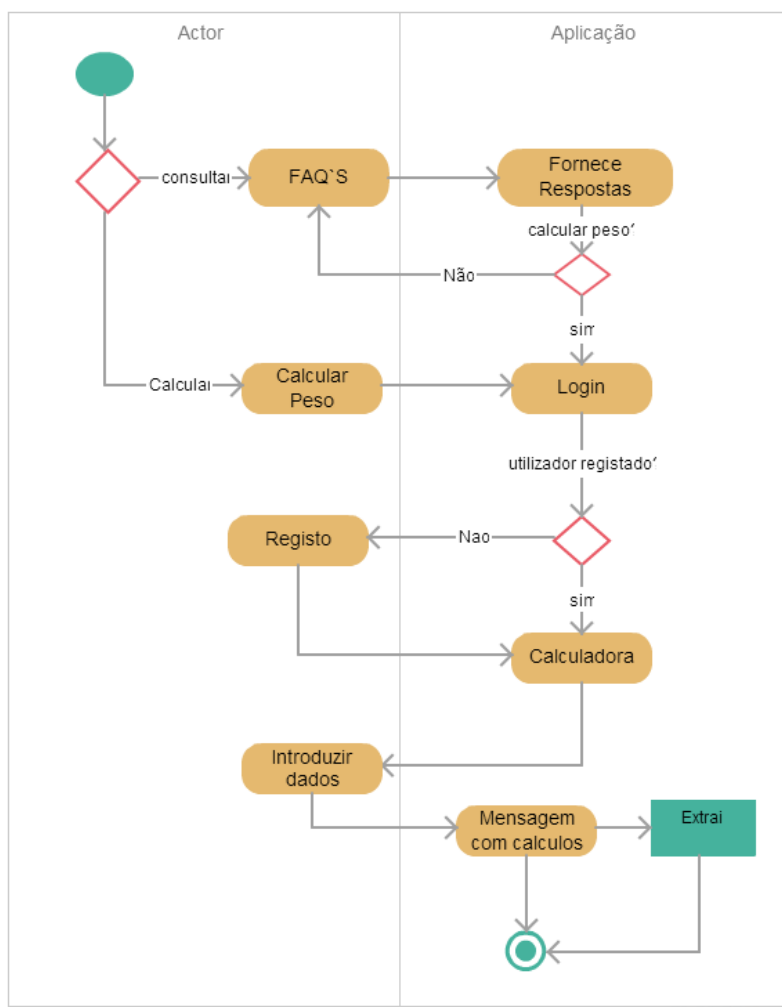
O “utilizador” terá duas acções principais a realizar sobre o sistema:

- Consultar perguntas e respostas frequentes sobre nutrição para gestantes;
- Calcular peso ideal de gestação.

Actor	Use Cases
Utilizador	Consultar Perguntas
	Calcular peso

Tabela 2: Descrição casos de uso

Estas acções estão igualmente representadas no diagrama de actividades a seguir:



**Figura 2: Diagrama de Actividade**

### 3.2. DESCRIÇÃO DAS CLASSES

O Aplicativo possui duas classes principais que interagem com a classe utilizador.

A classe “FAQ’S” é composta apenas por perguntas e respostas, e a classe “Calculadora de peso” apresenta mais atributos para o cálculo de peso.

Os atributos, métodos e associação estão representados no diagrama de classes, figura 3.

As classes serviram de base para os diagramas de sequência, figuras 4 e 5.

### 3.2.1. DIAGRAMA DE CLASSES

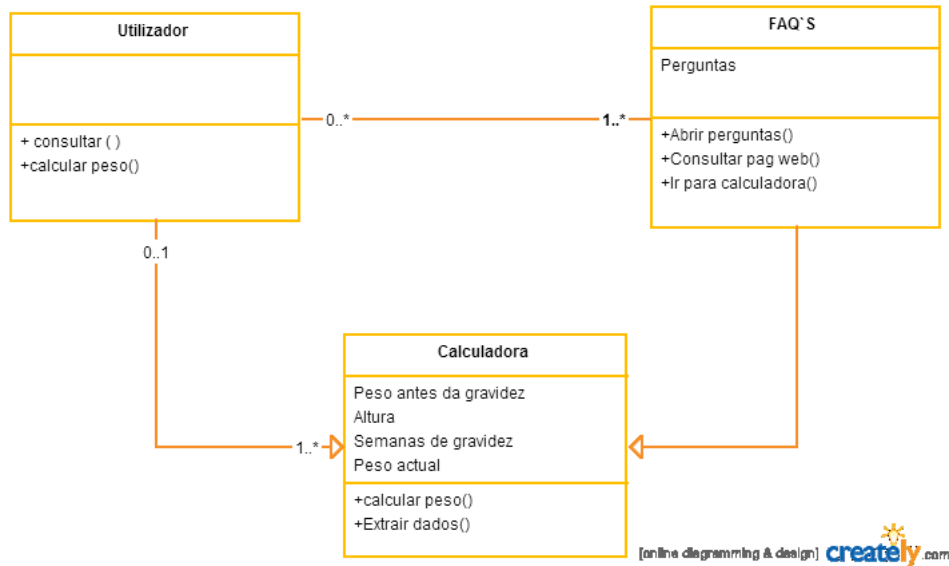


Figura 3: Diagrama de Classes

### 3.2.2. DIAGRAMA DE SEQUÊNCIA

Os diagramas de sequência representados a seguir mostram como os vários objectos interagem entre si, e de forma sequencial as mensagens trocadas em cada fase da interacção.

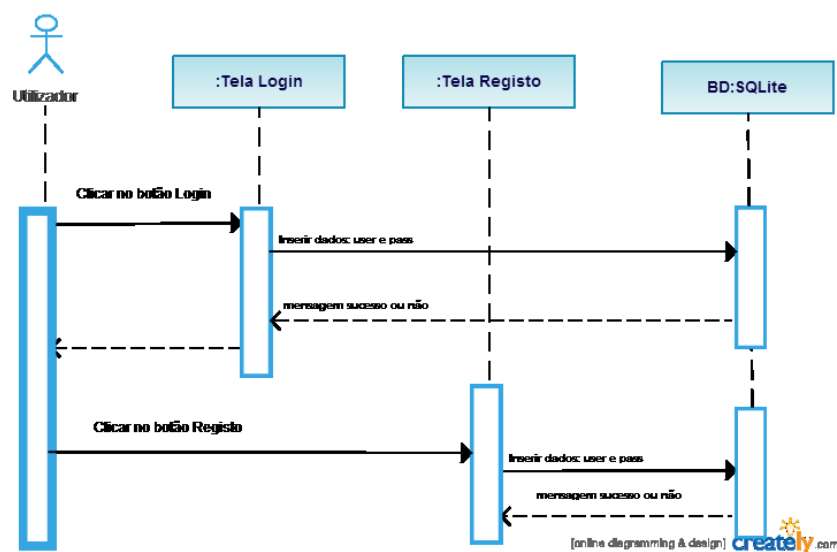


Figura 4: Diagrama de Sequência\_Login

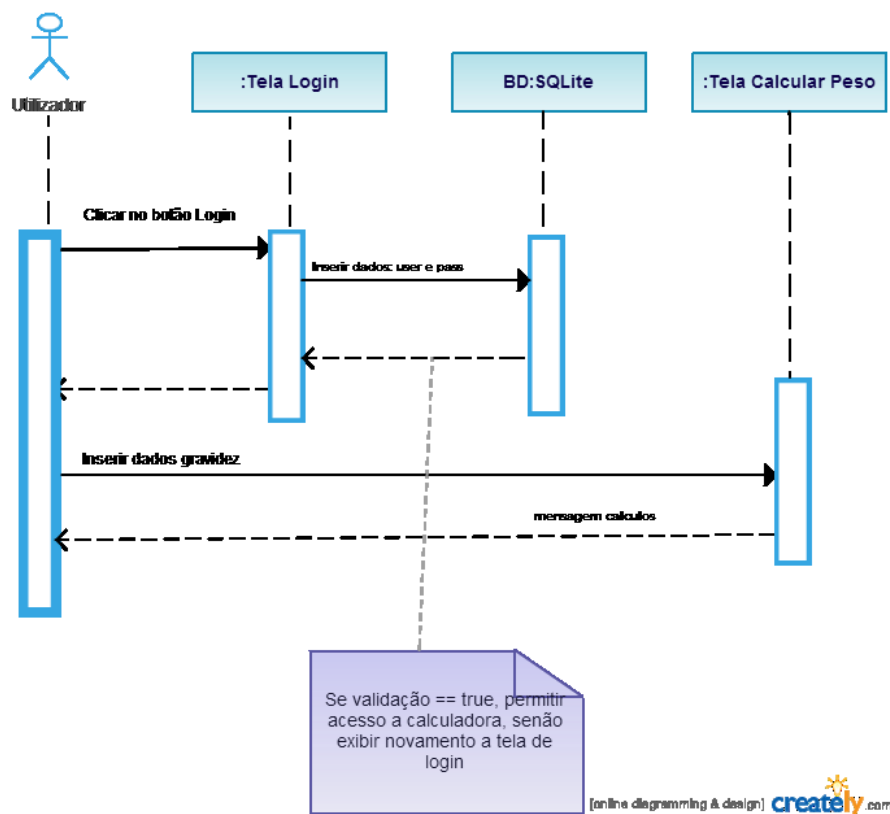


Figura 5: Diagrama de Sequência\_calculadora

### 3.3. IMPLEMENTAÇÃO E MODELO DE DADOS

A implementação do aplicativo no dispositivo móvel, é a seguir exemplificada, com base na modelação do diagrama UML de implementação e do modelo de dados. Apesar de se pretender que o aplicativo seja para qualquer dispositivo móvel, nesta etapa foi concebido apenas para *Android*.

#### 3.3.1. DIAGRAMA UML DE IMPLEMENTAÇÃO

A figura a seguir, mostra a arquitectura de instalação do aplicativo num dispositivo móvel, onde se pode ver, os componentes do software e sua interacção com os elementos de suporte ao processo de instalação.

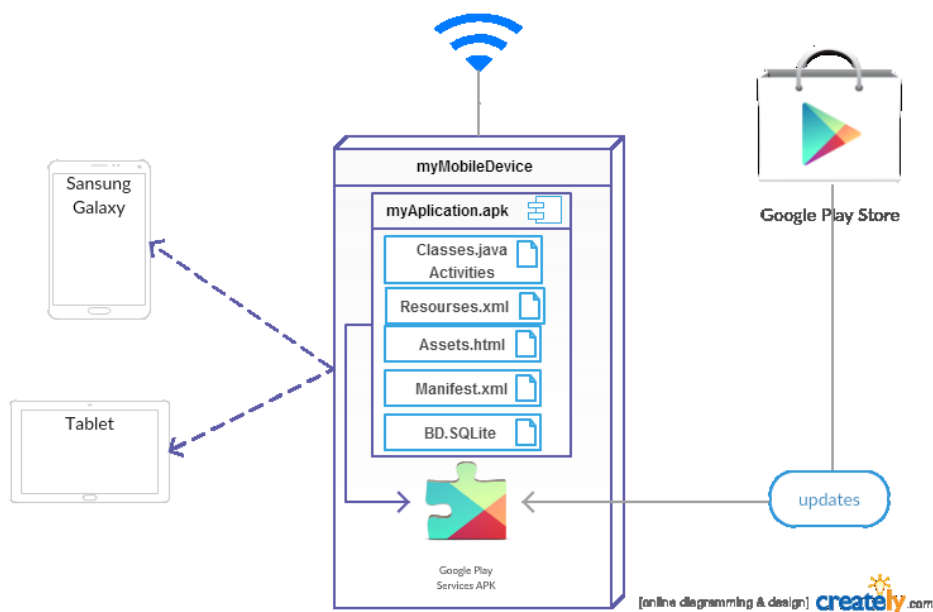


Figura 6: Diagrama de Implementação

### 3.3.2. MODELO DE DADOS

O modelo de dados a seguir dá-nos uma visão de como será construída a estrutura de dados que dará suporte ao processo do cálculo de peso, durante o período de gestação.

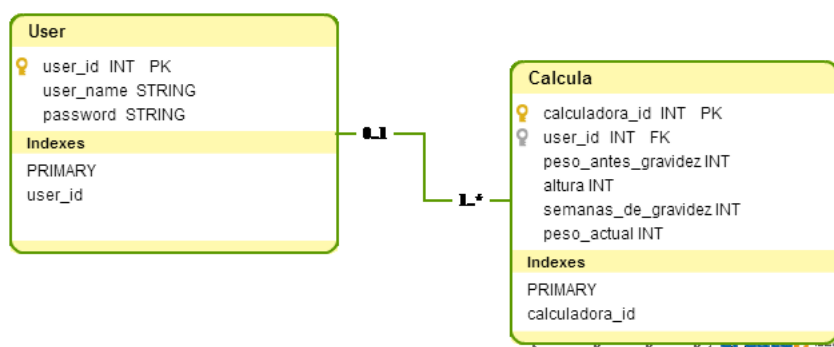


Figura 7: Modelo de Dados

A biblioteca *SQLite* foi a solução encontrada para ler e guardar dados da calculadora de peso.



### 3.4. ESCOLHA DO SISTEMA

Apesar de ter como objectivo o desenvolvimento numa aplicação móvel para aconselhamento a gestantes, para variados dispositivos móveis como *smartphones*, telemóveis e tablets, e para os respectivos sistemas operativos, a escolha para este protótipo foi o desenvolvimento numa aplicação com o sistema operativo *Android* e linguagem de programação *JAVA*, utilizando a plataforma *Android Studio*, pelas seguintes razões:

- Estar mais familiarizada com esta solução, visto ter semelhanças ao estudado na disciplina de Sistemas Móveis Empresariais;
- Ser uma plataforma aberta, facilmente adquirida e sem custos;
- Poder efectuar testes em cada fase de desenvolvimento, usando diversos emuladores tais como o *AVD (Android Virtual Device)* do *Android Studio*, *Genymotion* e no meu telemóvel que é igualmente *Android*.

### 3.5. TECNOLOGIAS DE DESENVOLVIMENTO

Breve apresentação da plataforma usada no projecto e suas componentes.

#### 3.5.1. ANDROID STUDIO

O *Android Studio* é um ambiente de desenvolvimento integrado (IDE) para o desenvolvimento na plataforma *Android*, criada pela Google e anunciada em 16 de maio de 2013.

O *Android Studio* estava em fase experimental com acesso antecipado a partir da versão 0,1 em Maio de 2013, de seguida, entrou em estágio beta a partir da versão 0.8, que foi lançado em junho de 2014. A primeira versão estável foi lançada em Dezembro de 2014, a partir da versão 1.0.

Com base no *IntelliJ IDEA*, o *Android Studio* é projectado especificamente para o desenvolvimento de aplicações *Android*. Está disponível gratuitamente para *download* no Windows, Mac OS X e Linux, e substituindo a ferramenta *Eclipse Android Development (ADT)*, como IDE principal do Google, para desenvolvimento de aplicativos nativos Android.

As principais funcionalidades e ferramentas deste novo Android Studio são:

- Criação de aplicações para *smartphones*, *tablets*, *smartwatches*, TVs, *Android Auto* e *Google Glass*;
- Editor de código inteligente com “*code completion*”, “*refactoring*” e análise;

- Teste simulado em dispositivos com todos os tamanhos e resoluções disponíveis no mercado;
- Monitorização de recursos utilizados pelas aplicações criadas;
- Suporte na nuvem *Google Cloud Platform*.

Para instalar o *Android Studio* no Windows é necessário:

- Fazer o *download* do *Android Studio*;
- Fazer o *download* do *JAVA JDK 6* ou superior apropriado (versão 32bits ou 64bits) caso não esteja já pré-instalado no computador;
- Instalar o *JDK*;
- Configurar o *Java\_Home*(para Windows 7);
- Instalar o *Android Studio*.

### 3.6. TÉCNICAS DE PROGRAMAÇÃO USADOS

Breves considerações sobre algumas das funcionalidades do *Android* importantes no desenvolvimento do protótipo.

#### 3.6.1. ACTION BAR, TABS AND FRAGMENTS

A *Action Bar* (barra de acção) é uma janela de recursos que identifica a localização do utilizador, e fornece as acções e modos de navegação do utilizador tais como pesquisar, suportar métodos de listas em *drop-down*.

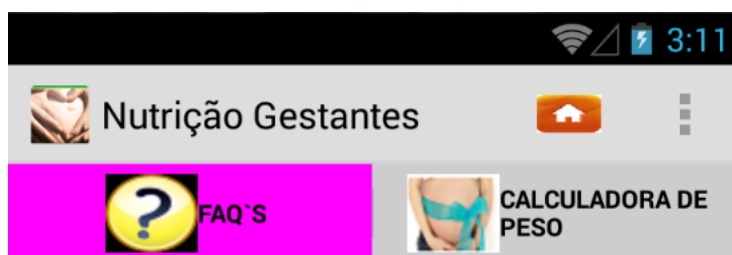


Figura 8: Action Bar e Tabs

Para o nosso protótipo, a *action bar* suporta, para além das *tabs* que compõem o aplicativo, um *icon* de acesso á página web de nutrição da ULHT.

As figuras a seguir, mostram o código para construção das *tabs* e fragmentos na barra de acção do aplicativo, e as condições em que estas se apresentam.

```
public class MainActivity extends Activity {

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {...}
    @Override
    public boolean onOptionsItemSelected(MenuItem item) {...}

    /**...*/
    private void LocationFound() {...}
    // Declaring our tabs and the corresponding fragments.
    ActionBar.Tab faqsTab, dadosTab;
    Fragment faqsFragmentTab = new FaqsFragmentTab();
    Fragment dadosFragmentTab = new DadosFragmentTab();

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

        // Asking for the default ActionBar element that our platform supports.
        ActionBar actionBar = getActionBar();

        // Screen handling while hiding ActionBar icon.
        actionBar.setDisplayHomeAsUpEnabled(true);

        // Screen handling while hiding ActionBar title.
        actionBar.setDisplayShowTitleEnabled(true);
    }
}
```

Figura 9: Código criação tabs e fragmentos

```
import ...

public class TabListener implements ActionBar.TabListener {

    private Fragment fragment;

    // The constructor.
    public TabListener(Fragment fragment) { this.fragment = fragment; }

    // When a tab is tapped, the FragmentTransaction replaces
    // Quando a tab é oculta, o Fragment substitui
    // O conteúdo do nosso layout principal com o fragmento especificado;
    // É por isso que declaramos um id para o layout principal.
    @Override
    public void onTabSelected(Tab tab, FragmentTransaction ft) {
        ft.replace(R.id.activity_main, fragment);
    }

    // Quando a tb não está seleccionada, precisamos ocultar da vista do utilizador.
    @Override
    public void onTabUnselected(Tab tab, FragmentTransaction ft) { ft.remove(fragment); }
}
```

Figura 10: Código chamada da tab

### 3.6.2. ACTIVITIES E INTENTS

Uma Activity é basicamente uma classe gerenciadora de UI (Interface com o usuário). Todo aplicativo android começa por uma Activity, isto é, quando uma aplicação android é executada, na verdade é a sua Activity principal que é lançada.

```
package pt.ulusofona.nutrigestante;

import ...

public class Pergunta1 extends Activity {

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.activity_main_actions, menu);

        return super.onCreateOptionsMenu(menu);
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Take appropriate action for each action item click
        switch (item.getItemId()) {

            case R.id.action_location_found:
                // location found
                LocationFound();
                return true;
            case R.id.action_check_updates:
                // check for updates action
                return true;
            default:
                return super.onOptionsItemSelected(item);
        }
    }
}
```

Figura 11: Exemplo de uma activity

As activities têm um ciclo de vida, conforme se pode ver no diagrama a seguir:

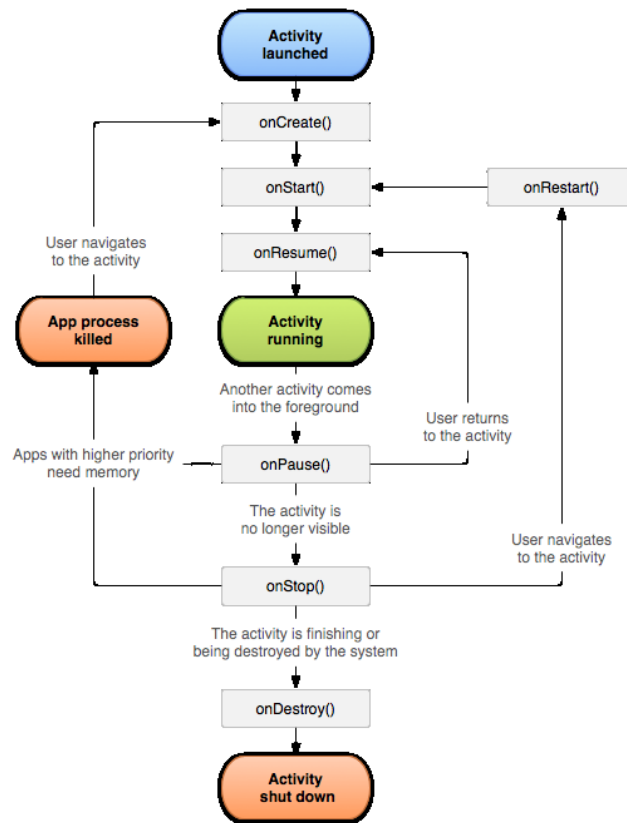


Figura 12: Ciclo de vida de uma activity

A figura acima apresenta os principais métodos de uma activity, que são:

1. **onCreate()** – Executado quando uma Activity é criada. Geralmente é o método responsável por carregar os layouts (XML) e outras operações de inicialização. Só é executado 1 vez durante o Ciclo de Vida da Activity.
2. **onStart()** – É chamado logo depois do **onCreate()** ou quando a Activity que estava em background volta a ter foco.
3. **onResume()** – Método chamado quando a Activity vai começar a interagir com o usuário (ou retoma foco). Sempre vem depois do **onStart**, caso a Activity tenha sido iniciada ou se ela estava parada (**onStop**), ou pode ocorrer direto (vindo depois do **onPause**) caso a Activity, que não estava em primeiro plano, mas ainda visível, volte para o topo.
4. **onPause()** – Ocorre logo antes da Activity perder o foco, ou seja, quando ir para o Background, mas não foi (ainda) destruída. Geralmente é usada para parar animações e recursos que estejam consumindo processamento e persistir dados não salvos, desta forma, se (por falta de recursos) a Activity for destruída, não perderemos essas informações. Tenha em mente que a Activity que está tomando o foco não será criada até que esse método retorne, sendo assim, o código aqui também deve ser leve.

5. [onStop\(\)](#) – Análoga à onPause(), só é chamada quando a Activity fica completamente encoberta por outra Activity (não é mais visível).
6. [onRestart\(\)](#) – É chamado quando uma Activity que estava parada volta ao foco depois de estar em background, imediatamente antes da onStart().
7. [onDestroy\(\)](#) – Esse método ocorre logo antes da Activity ser destruída/finalizada.

As *Activities* podem ser iniciadas por *Intents*. Uma *Intent* (intenção) é uma descrição abstrata de uma operação a ser executada. Ela pode ser utilizada para iniciar uma Activity, “ativar” um broadcast, enviar uma mensagem para uma aplicação que roda em outro processo etc. Uma Intent faz parte da arquitetura do Android e em linguagem corrente representaria uma “Mensagem”, um pedido que é encaminhado ao sistema operacional. O sistema pegará a mensagem, verificará qual é a “Intenção da mensagem” e tomará uma decisão que pode ser desde abrir uma página na Web (um browser abrirá e a página será exibida), fazer uma chamada telefônica ou iniciar uma Activity.

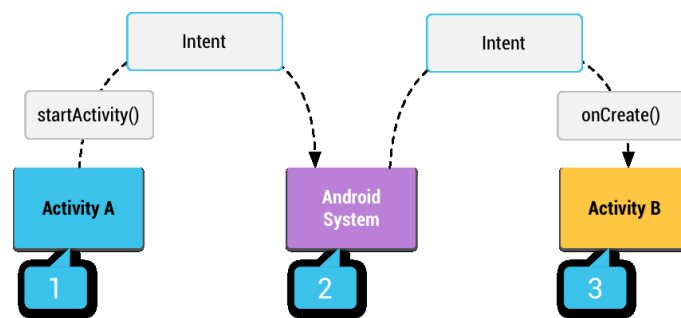


Figura 13: Interação Intents com Activities

Exemplo do uso de *Intents* no protótipo desenvolvido:

```
public void onItemClick(AdapterView<?> parent, View view, int position,
    long id) {
    switch( position)
    {
        case 0: Intent myIntent = new Intent(getActivity(), Pergunta1.class);
                startActivity(myIntent);
                break;

        case 1: Intent myIntent1 = new Intent(getActivity(), Pergunta2.class);
                startActivity(myIntent1);
                break;
        case 2: Intent myIntent2 = new Intent(getActivity(), Pergunta3.class);
                startActivity(myIntent2);
                break;
        case 3: Intent myIntent3 = new Intent(getActivity(), Pergunta4.class);
                startActivity(myIntent3);
                break;
        case 4: Intent myIntent4 = new Intent(getActivity(), Pergunta5.class);
                startActivity(myIntent4);
                break;
        case 5: Intent myIntent5 = new Intent(getActivity(), Pergunta6.class);
```

Figura 14: Uso de Intents em FAQ'S fragemnt Tab

Cada *Activity* está associada a um *layout* que podem ser definidos em *runtime* ou através de ficheiros *xml*. O protótipo usa ficheiros *xml* para o design da tela, e definição do tamanho e cor de botões e letras. Para a *TAB FAQ'S* foram usadas *webviews* para aceder aos ficheiros *html* onde constam as respostas das perguntas.

```
android:orientation="vertical"
tools:context="pergunta2">

<ScrollView
    android:id="@+id/scrollView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentRight="true"
    android:layout_alignParentTop="true"
    android:layout_marginRight="132dp"
    android:layout_marginTop="112dp" >

    </ScrollView>
    <WebView
        android:id="@+id/webView1"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Ir para Calculadora de Peso"
        android:id="@+id/button"
        android:layout_alignParentBottom="true"
        android:layout_alignParentRight="true"
```

Figura 15: ficheiro xml da pergunta 2

```
}
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.pergunta2);

    WebView webView = (WebView) this.findViewById(R.id.webView1);
    webView.loadUrl("file:///android_asset/pergunta2.html");
    webView.getSettings().setJavaScriptEnabled(true);
    webView.setWebViewClient( new WebViewClient() );

    Button button = (Button) findViewById(R.id.button);
    button.setOnClickListener((v) -> {

        Intent myIntent = new Intent(Pergunta2.this, CalculadoraActivity.class);
        startActivity(myIntent);

    });
}
```

Figura 16: Ficheiro java da pergunta 2





Figura 17: Exemplo do resultado do código gerado em FAQ'S

### 3.6.3. ANDROID MANIFEST

O AndroidManifest.xml é o arquivo principal do projeto, onde ficam todas as configurações. Ele obrigatoriamente deve ficar na pasta raiz do projeto, contendo todas as configurações necessárias para executar a aplicação, como o nome do pacote utilizado, o nome das classes de cada activity e várias outras configurações.

Dentro da tag <manifest> é necessário declarar o pacote principal do projeto, utilizando a tag <package>. É obrigatório que cada activity do projeto esteja declarada no arquivo AndroidManifest.xml, caso contrario não é possível utiliza-la. Para declarar a activity é utilizada a tag <activity>, que recebe o nome da classe, e é sempre relativa ao pacote principal.

O protótipo foi guardado no pacote "pt.lusofona.nutrigestante" e devidamente declarado conforme a seguir se indica:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android">
```



```

package="pt.ulusofona.nutrigestante"

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="pt.ulusofona.nutrigestante"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="11"
        android:targetSdkVersion="19" />

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission
        android:name="com.google.android.providers.gsf.permission.READ_GSERVICES" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />

    <uses-feature
        android:glEsVersion="0x00020000"
        android:required="true" />

    <uses-permission android:name="android.permission.READ_SMS" />
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
    <uses-permission android:name="android.permission.READ_PHONE_STATE" />
    <uses-permission
        android:name="android.permission.READ_EXTERNAL_STORAGE"
        android:maxSdkVersion="18" />

    <!--
    To retrieve OAuth 2.0 tokens or invalidate tokens to disconnect a user. This disconnect
    option is required to comply with the Google+ Sign-In developer policies
    -->
    <uses-permission android:name="android.permission.USE_CREDENTIALS" /> <!-- To retrieve
    the account name (email) as part of sign-in: -->
    <uses-permission android:name="android.permission.GET_ACCOUNTS" /> <!-- To auto-
    complete the email text field in the login form with the user's emails -->
    <uses-permission android:name="android.permission.READ_PROFILE" />
    <uses-permission android:name="android.permission.READ_CONTACTS" />
    <uses-permission android:name="android.permission.READ_CALL_LOG" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/MyTheme" >

        <!-- Splash screen -->
        <activity
            android:name=".Splash"
            android:label="@string/app_name"
            android:screenOrientation="portrait"
            android:theme="@android:style/Theme.Black.NoTitleBar" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

        <!-- Main activity -->
        <activity
            android:name=".MainActivity"
            android:label="@string/app_name" >
        </activity>
        <activity
            android:name=".Pergunta1"
            android:label="@string/app_name" >
        </activity>
        <activity
            android:name=".Pergunta2"
            android:label="@string/app_name" >
        </activity>
        <activity
            android:name=".Pergunta3"
            android:label="@string/app_name" >
        </activity>
        <activity
            android:name=".Pergunta4"
            android:label="@string/app_name" >
        </activity>
        <activity
            android:name=".Pergunta5"
            android:label="@string/app_name" >
        </activity>
        <activity
            android:name=".Pergunta6"
            android:label="@string/app_name" >
        </activity>
        <activity
            android:name=".Pergunta7"
            android:label="@string/app_name" >
        </activity>
    </application>

```

Figura 18: Ficheiro Android Manifest

### 3.6.4. CALCULADORA DE PESO

A calculadora de peso é a principal componente deste protótipo pois é onde permite que as gestantes introduzam os seus dados, para daí extrair informação útil.

Para aceder á calculadora, é necessário que a gestante (o utilizador) se registre no aplicativo e faça login, de modos a que lhe seja atribuído um *id* para guardar e gerir os seus dados relativos a semana de gravidez e peso. Com base nestes dados será possível verificar se a gestante está ou não dentro do peso ideal.

```
public class DadosFragmentTab extends Fragment implements OnClickListener {  
  
    Button button2, button3;  
  
    @Override  
    public View onCreateView(LayoutInflater inflater, ViewGroup container,  
        Bundle savedInstanceState) {  
        View rootView = inflater.inflate(R.layout.dados_layout, container, false);  
  
        button2 = (Button) rootView.findViewById(R.id.button2);  
        button3 = (Button) rootView.findViewById(R.id.button3);  
        button2.setOnClickListener(this);  
        button3.setOnClickListener(this);  
  
        return rootView;  
    }  
    @Override  
    public void onClick(View v) {  
        // TODO Auto-generated method stub  
        switch (v.getId()) {  
            case R.id.button2:  
                Intent intent = new Intent(getActivity(), LoginActivity.class);  
                this.startActivity(intent);  
                break;  
            case R.id.button3:  
                Intent myintent1 = new Intent(getActivity(), RegisterActivity.class);  
                startActivity(myintent1);  
                break;  
        }  
    }  
}
```

Figura 19: Código do painel calculadora peso



Figura 20: Painel Calculadora de Peso

Não foi possível usar neste protótipo o mesmo código da calculadora constante na página web de nutrição já referida neste documento, pelo que a solução encontrada foi criar um código de raiz que obtivesse os mesmos resultados. Para tal, tive como referência o site de recomendações de peso antes e depois da gravidez do programa Havard Medical School Portugal <https://hmsportugal.wordpress.com/2011/08/17/o-seu-peso-antes-e-durante-a-gravidez-importancia-e-recomendacoes/>

Deste modo, para calcular o peso ideal é necessário antes calcular o peso saudável para a população adulta em geral, que varia de pessoa para pessoa em função, principalmente, da altura. O cálculo de peso saudável permite verificar se o peso da pessoa está adequado a sua altura, através do indicador de IMC (índice de massa corporal).

Para calcular o IMC usa-se a fórmula:

$$\text{IMC} = \text{peso (kg)} \div \text{altura}^2 \text{ (m)}$$

Classificação do IMC:

IMC menor que 18,5	↔	<b>Baixo peso</b>
IMC entre 18,5 e 24,9	↔	<b>Peso normal</b>
IMC maior ou igual a 25	↔	<b>Excesso de peso</b>
IMC maior ou igual a 30	↔	<b>Obesidade</b>

**Recomenda-se que as mulheres procurem engravidar com um peso normal, ou seja, com um IMC entre 18,5 e 24,9.**

Portanto, após o cálculo do IMC antes da gravidez, associa-se a indicadores de ganho de peso médio por semana de gravidez, para obter o resultado do peso ideal em cada etapa da gravidez. Os dados e os cálculos são relativos a gravidez de apenas um bebé.

```

if (!(valorAltura != 0) && (valorPanterior > 30 && valorPanterior < 180) && (valorSgravidez>=0 && valorSgravidez<=42)) {
    Toast.makeText(this, "introduza um valor valido", Toast.LENGTH_LONG).show();
}
else {
    float resultadoapeso = valorPactual - valorPanterior;
    float resultadoimc = valorPanterior / (valorAltura * valorAltura);

    TextView imc = (TextView) findViewById(R.id.imc);
    imc.setText("IMC =" + resultadoimc);

    TextView apeso = (TextView) findViewById(R.id.apeso);
    apeso.setText("Aumento de Peso =" + resultadoapeso + "kg");

    float limMin = (float) 0.4 * (valorSgravidez-1);
    float limMax = (float) 0.4 * (valorSgravidez);

    TextView pideal = (TextView) findViewById(R.id.pideal);
    pideal.setText("O Intervalor de peso saudável para esta etapa da gravidez está entre:" + (float) (limMin+valorPanterior) + "kg "

    if (resultadoapeso<0) {
        Toast.makeText(this, "Está abaixo do peso. Consulte seu médico.", Toast.LENGTH_LONG).show();
    }else if ((resultadoapeso < limMin) && (resultadoimc < 18.5)) {
        Toast.makeText(this, "Está abaixo do peso. Consulte seu médico.", Toast.LENGTH_LONG).show();
    }else if ((resultadoapeso<=limMin)&& (resultadoimc>=18.5 && resultadoimc <= 24.5)) {

```

Figura 21: Código do calculo do peso ideal

O código acima permite que a gestante obtenha informação sobre o seu peso comparativamente ao considerado ideal com base nos cálculos de IMC e tabela de referência citada acima, e guarda os dados numa tabela da bd SQLite.

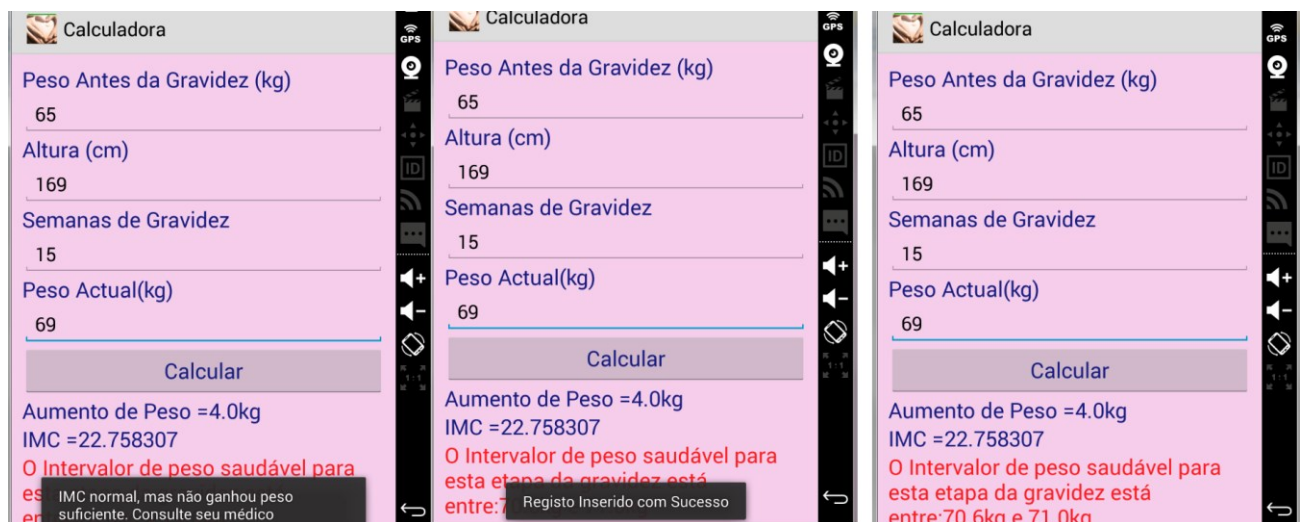


Figura 22: Resultados Caluladora

A inserção dos dados da gestante na base de dados local do aplicativo, tem como objectivo obter um output relativo a evolução do peso durante a gravidez, que poderá ser um gráfico ou tabela, tal como mostra o exemplo a seguir:

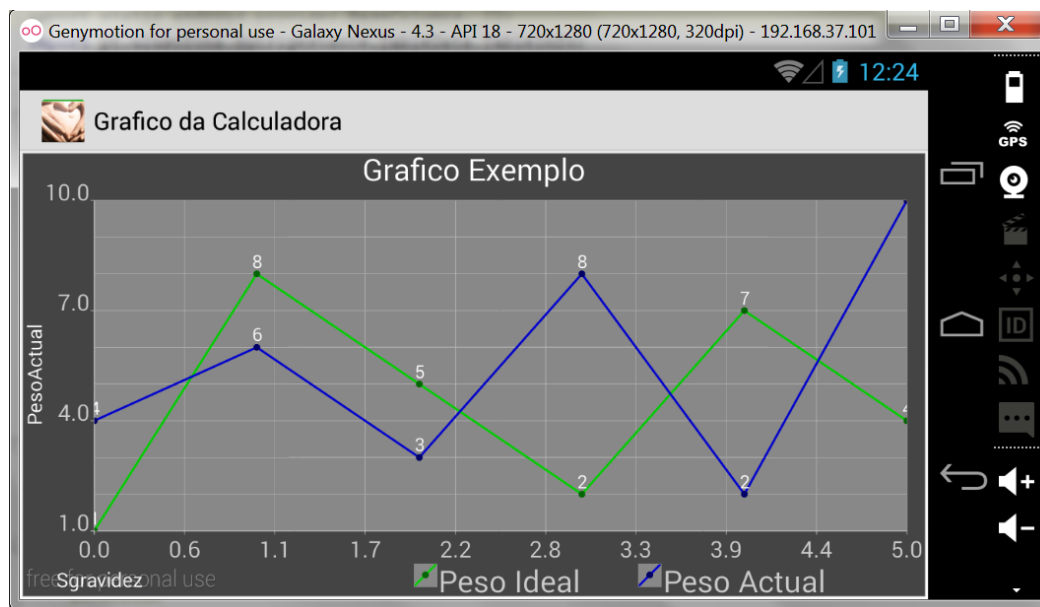


Figura 23: Gráfico exemplo

Os valores apresentados no gráfico não provêm da calculadora, serviram apenas para criar o gráfico exemplo.

### 3.6.5. SQLite

Além das várias inovações implementadas, o Android traz também suporte nativo para o SQLite.

O SQLite é uma biblioteca, desenvolvida em linguagem C, que implementa um amplo subconjunto do *standard* SQL 92, sendo a sua reputação proveniente da combinação do motor de base de dados com a interface dentro de uma única biblioteca. As aplicações que usam SQLite podem ter assim acesso a uma base de dados racional SQL, sem a necessidade de executarem processos RDBMS (*relational database management system*) em separado e sem grandes *overheads*. O SQLite foi desenvolvido em 2000 e é actualmente uma base de dados amplamente adoptada em dispositivos móveis, suportando até 2 TB de dados.

A escolha do SQLite para este projecto teve como base a necessidade de permitir ao utilizador o uso do aplicativo “Nutrição Gestantes”, mesmo não

havendo ligação á *internet*. Assim, o aplicativo tem uma base de dados local, onde armazena os dados necessários para a calculadora de peso.

```
import android.provider.BaseColumns;

/**
 * Created by AnaSilva on 01-06-2015.
 */
public class TableData {

    public TableData()
    {

    }

    public static abstract class TableInfo implements BaseColumns {

        public static final String USER_NAME = "user_name";
        public static final String USER_PASS = "user_pass";
        public static final String DATABASE_NAME = "user_info";
        public static final String TABLE_NAME = "reg_info";

    }

}
```

Figura 24: Tabela do SQLite

```
public class DatabaseOperations extends SQLiteOpenHelper {

    public static final int database_version = 1;
    public String CREATE_QUERY = "CREATE TABLE "+TableInfo.TABLE_NAME+" (" +TableInfo.USER_NAME+" TEXT,"+TableInfo.USER_PASS+" TEXT)";

    public DatabaseOperations(Context context) {
        super(context, TableInfo.DATABASE_NAME, null, database_version);
        Log.d("Database operations", "Database created");
    }

    @Override
    public void onCreate(SQLiteDatabase sdb) {
        sdb.execSQL(CREATE_QUERY);
        Log.d("Database operations", "Table created");
    }

    @Override
    public void onUpgrade(SQLiteDatabase sqLiteDatabase, int i, int i2) {

    }

    public void putInformation(DatabaseOperations dop, String name, String pass){

        SQLiteDatabase SQ = dop.getWritableDatabase();
        ContentValues cv = new ContentValues();
        cv.put(TableInfo.USER_NAME, name);
        cv.put(TableInfo.USER_PASS, pass);
        long k = SQ.insert(TableInfo.TABLE_NAME, null, cv);
        Log.d("Database operations", "One row inserted");
    }

}
```

Figura 25: Código SQLite

## 4. CONCLUSÕES

O objectivo principal deste trabalho era o desenvolvimento do protótipo de uma aplicação móvel que permitisse às gestantes não só obter informação sobre aconselhamento nutricional, como possibilitar o controlo de peso durante cada etapa de gravidez, para uma gestação feliz e saudável.

Conforme se pode verificar com base nos testes efectuados ao protótipo desenvolvido, os objectivos propostos foram alcançados.

Tratando-se de um protótipo, podem ser efectuadas melhorias, atendendo as tendências e necessidades do público-alvo.

### 4.1. MELHORIAS PARA O FUTURO

Podem ser efectuadas diversas melhorias, tanto a nível da interface gráfica e *design*, como relacionadas com a funcionalidade do aplicativo.

As mais importante e a realçar:

- Introdução do cálculo para gravidez de gémeos;
- Mostrar os dados gravados na BD em forma de gráfico ou tabela, de modos a acompanhar a evolução da gestante;
- Adaptar o código a outros sistemas operativos e dispositivos móveis;
- Criar a versão em Inglês.

## 5. BIBLIOGRAFIA

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

<https://hmsportugal.wordpress.com/2011/08/17/o-seu-peso-antes-e-durante-a-gravidez-importancia-e-recomendacoes/>

[https://en.wikipedia.org/wiki/Android\\_Studio](https://en.wikipedia.org/wiki/Android_Studio)

<http://www.cs.dartmouth.edu/~campbell/cs65/lecture08/lecture08.html>

<https://developer.android.com/training/building-userinfo.html>

<http://examples.javacodegeeks.com/android/core/app/fragment/android-fragments-example/>

<http://v4all123.blogspot.pt/2013/04/fragments-in-android.html>

<http://instinctcoder.com/android-studio-sqlite-database-example/>

**Queirós, R (2014). *Desenvolvimento de Aplicações Profissionais em Android*, FCA**