

Preparando um ambiente de desenvolvimento para a plataforma android

Setting up a development environment for android platform

Felipe Pires de Oliveira¹, Bruno Morais Lemos².

Resumo

Este artigo descreve as etapas para a criação de um ambiente de desenvolvimento para a plataforma Android. O artigo foi dividido em quatro partes: Primeiro uma introdução sobre o assunto. Na segunda seção são mostrados os passos necessários para a configuração do ambiente de desenvolvimento. Na terceira seção, é demonstrado como criar uma aplicação para dispositivo Android que calculo Índice de Massa Corporal (IMC). A quarta seção apresenta as considerações finais. A solução abordada tem como base a IDE Eclipse e o SDK do Android, bem como o Kit de Desenvolvimento Java da Oracle (JDK).

Palavras-chave: Android. SDK. Eclipse. IDE. Java. Dispositivos Móveis.

Como citar esse artigo. Oliveira FP, Lemos BM. Preparando um ambiente de desenvolvimento para a plataforma android. Revista Teccen. 2015 Jul./Dez.; 08 (2): 17-28.

Abstract

This article describes the steps for setting up a development environment for the Android platform. It is divided into four parts: First an introduction about de subject. The second section shows the steps required to configure the development environment. The third section shows how to create an application for Android device that calculates Body Mass Index (BMI). The fourth section presents the final remarks. The solution discussed has based on the Eclipse IDE and the Android SDK and Oracle's Java Development Kit (JDK).

Keywords: Android. SDK. Eclipse. IDE. Java. Mobilie Devices.

Introdução

Com o constante avanço da Tecnologia da Informação (TI), é crescente o uso das novas mídias. A *Internet* cresce de forma exponencial e atualmente estima-se que possui cerca de 2 bilhões de usuários. Com o uso dessas novas tecnologias é cada vez maior o número de dispositivos que podem ser usados para acessar a *Internet*, sendo utilizados para as mais diversas finalidades, como: entretenimento, educação, fins comerciais e comunicação.

O uso de dados está crescendo rapidamente, de acordo com relatório da Oracle intitulado Opportunity Calling: The Future of Mobile Communications – Take Two. De 2010 até os dias atuais, 47% dos 68% usuários globais de smartphone passaram a usar um volume maior de dados, percentual superior ao uso de telefonia móvel para texto ou voz (ComputerWorld, 2011).

Este artigo tem como principal objetivo

apresentar a tecnologia utilizada e uma metodologia de desenvolvimento para a criação de uma aplicação (App) para a plataforma Android. Para atingir tal objetivo serão utilizadas como ferramentas: a IDE Eclipse, a linguagem de programação Java, o sistema operacional Linux com a distribuição Ubuntu e a plataforma para dispositivos móveis Android.

O ambiente de desenvolvimento

O sistema operacional usado para a instalação de todos os componentes foi o Linux em sua distribuição Ubuntu 12.04 – 32 bits, versão beta 2. O Eclipse foi escolhido, pois é uma das mais utilizadas IDEs (Integrated Development Environment) disponíveis atualmente. Esta seção irá demonstrar como obter, instalar e configurar os programas necessários para o desenvolver uma aplicação para a plataforma Android.

1. Universidade Severino Sombra, Sistemas de Informação, Vassouras-RJ, Brasil.

2. Universidade Severino Sombra, Sistemas de Informação, Engenharia de Computação, Matemática, Vassouras-RJ, Brasil.

O Java Development Kit (JDK)

Um dos primeiros passos para a montagem de nosso ambiente de desenvolvimento é a instalação do Java Development Kit (JDK). A obtenção do JDK é vital para a continuidade dos passos propostos neste artigo por conter o compilador Java, o depurador Java e outras ferramentas para programação, além do Java Runtime Environment (JRE) através do qual aplicações Java podem ser executadas. Para isso, visite o site <http://www.oracle.com/technetwork/java/javase/downloads/index.html> e clique no botão para *download* da mais nova versão do JDK disponível. Neste artigo, utilizou-se o Java Platform (JDK) 7u3.

Na página seguinte, marque a opção em *Accept License Agreement* e, em seguida clique no *link* para *download* da versão 32 bits para o *Linux* ([jdk-7u3-linux-i586.tar.gz](http://www.oracle.com/technetwork/java/javase/downloads/jdk-7u3-linux-i586.tar.gz)). Logo após realizar o *download*, mova ou copie o arquivo baixado para seu diretório de usuário.

Como instalar o JDK

Com o botão direito do mouse, clique sobre o arquivo [jdk.7u3-linux-i586.tar.gz](http://www.oracle.com/technetwork/java/javase/downloads/jdk-7u3-linux-i586.tar.gz) e, no menu, selecione a opção *Extrair aqui*. Por fim, tem-se um resultado semelhante ao encontrado na Figura 1.



Figura 1. O diretório do JDK recém extraído.

Abra um terminal e entre como superusuário através do comando `su` e execute os seguintes comandos

como no Quadro 1:

Quadro. Movendo o diretório de instalação do JDK para o diretório `/usr/`.

```
#cd Programas/
#mv jdk1.7.0_03 /usr/
```

O comando acima - `mv` - irá mover o diretório de instalação do JDK para o diretório `/usr/` tornando seu acesso por outros programas mais fácil.

O Eclipse

O Eclipse é uma *Integrated Development Environment* (IDE) que pode ser utilizada no desenvolvimento de softwares em qualquer linguagem, não apenas Java.

O projeto foi iniciado como substituto proprietário para o Visual Age for Java da IBM, mas teve seu código aberto em 2001. O Eclipse é agora controlado por uma organização independente e sem fins lucrativos chamada Eclipse Foundation (Burnette, 2005).

Para obter a IDE Eclipse visite o site <http://www.eclipse.org/> (Figura 2) e clique no link *Download Eclipse*.



Figura 2. Site oficial da IDE Eclipse.

Baixe a versão do IDE Eclipse mais apropriada para seu sistema (32 ou 64 bits) clicando no link *Eclipse IDE for Java EE Developers* (Figura 3).



Figura 3. Fazendo o download da mais nova versão do Eclipse IDE for Java Developers.

Após fazer o download da última versão do Eclipse para seu computador, crie um sub-diretório chamado Programas dentro do diretório de seu usuário e mova o arquivo para este local. O mesmo será descompactado posteriormente.

Como instalar o Eclipse IDE

Para efetuar a instalação do Eclipse, basta extrair

o arquivo eclipse-jee-indigo-SR2-linux-gtk.tar.gz dentro do diretório de Programas. Porém, antes de executá-lo, edite o arquivo /etc/profile através do comando vim executado como superusuário (Quadro 2). Este procedimento é necessário para que a IDE Eclipse possa acessar a máquina virtual Java.

Quadro. Editando o arquivo profile.

```
#vim /etc/profile
```

Entre no modo de edição pressionando a tecla 'I' e adicione as linhas abaixo logo após os comentários

da parte superior:

Quadro. Alterações realizadas no arquivo profile.

```
# Configurando o caminho do JDK
JAVA_HOME=/usr/jdk1.7.0_03; export JAVA_HOME
PATH=$PATH:$JAVA_HOME/bin
export PATH
# Fim do Configurando o caminho do JDK
```

Após editar o arquivo, pressione a tecla ESC para abandonar o modo de edição e insira o comando :wq! para salvar as alterações e sair do programa Vim.

Para que o sistema possa compreender corretamente a alteração que acaba de ser realizada, encerre a sessão e entre novamente. Uma reinicialização do sistema surtirá o mesmo efeito.

Executando o Eclipse

Ao executar o Eclipse pela primeira vez, faz-se necessário especificar um local padrão para armazenamento de nossos projetos. Para que a mensagem não volte a ser exibida, marque a caixa de verificação *Use this as default and do not ask again* (Figura 4).

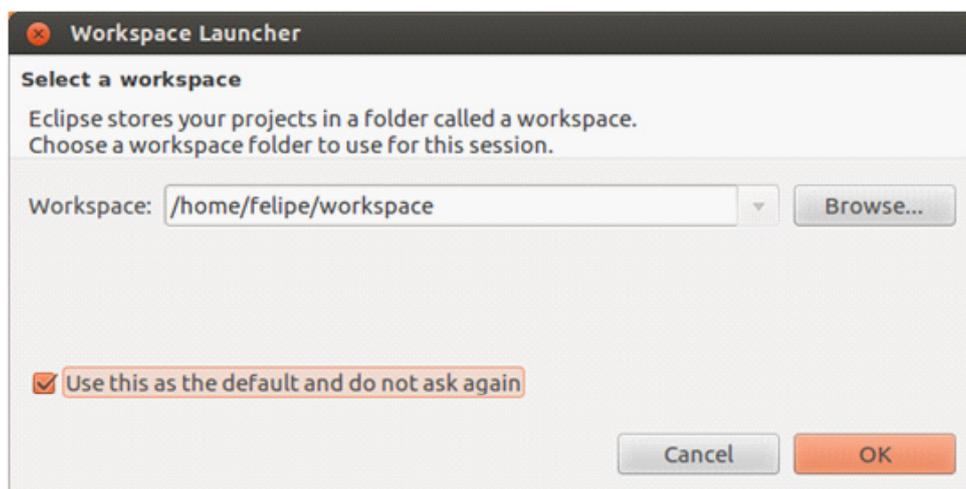


Figura 4. Parâmetros básicos para o ambiente de desenvolvimento Android.

A plataforma Android

O Android é uma plataforma compreensível e de código aberto projetada para dispositivos móveis. A plataforma defendida pela Google é de propriedade da Open Handset Alliance. A meta da aliança é acelerar a inovação no meio móvel e oferecer ao mercado consumidor uma mais rica, mais barata, e melhor experiência móvel (Gargenta, 2011).

O Android é a primeira plataforma móvel aberta que separa o *software* do *hardware*. Isso permite que um número muito maior de dispositivos seja executado a mesma aplicação e cria um ecossistema mais rico para desenvolvedores e consumidores.

Obtendo o Android Software Development Kit (SDK)

O Android Software Development Kit é tudo

Quadro. Executando o arquivo android.

```
$ cd Programas/android-sdk-linux/tools/
$ ./android
```

Na janela exibida, pode-se selecionar e baixar diferentes versões das APIs para o sistema Android. Não é necessário realizar o *download* de todas as APIs.

o que você precisa para desenvolver aplicações para Android. O SDK vem com um conjunto de ferramentas assim como uma plataforma para executar e visualizar todo o trabalho (Gargenta, 2011).

Para obter o SDK do Android, visite o site <http://developer.android.com/sdk/index.html> e baixe a última versão do SDK para o Linux. No momento que este artigo foi escrito, essa versão era *android-sdk_r17-linux.tgz*.

Como instalar o SDK do Android

Extraia o conteúdo do arquivo *android-sdk_r17-linux.tgz* e, dentro do diretório extraído, navegue ao subdiretório *tools* e execute o arquivo *android* (Quadro 3).

Particularmente este artigo optou por marcar as opções Android SDK Tools, Android 2.2 (API 8) e Extras (Figura 5).

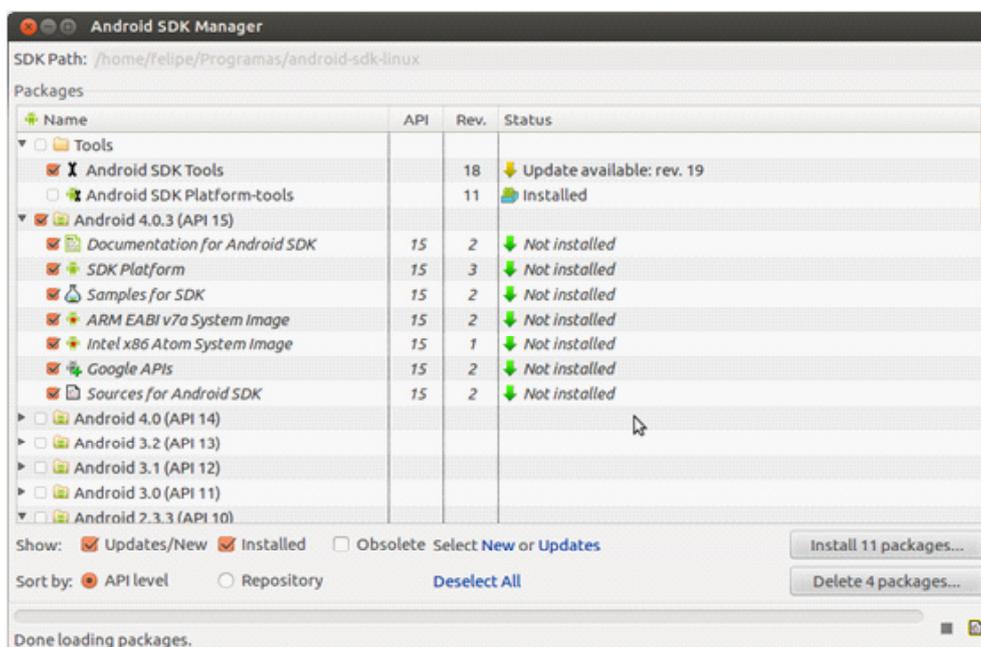


Figura 5. Selecionando as APIs para instalação.

Como instalar o Android Development Tools (ADT) Plugin

Para finalizar, pressione o botão *Install packages...*, marque a opção *Accept all* e clique em ok. Após o download terminar, feche a janela.

O Android Development Tools (ADT) é um plugin para a IDE Eclipse desenvolvido para oferecer aos desenvolvedores um ambiente integrado e poderoso

para o desenvolvimento de aplicações Android.

Segundo o site oficial Android Developers, o ADT plugin amplia as capacidades da IDE Eclipse permitindo a rápida configuração de projetos Android, criação de interfaces de usuário, adiciona pacotes baseados no API do Framework do Android, depura aplicações utilizando as ferramentas do SDK do Android, e ainda exporta arquivos no formato .apk para futura distribuição de

suas aplicações.

Utilizando a IDE Eclipse, abra o menu Help e clique em *Install New Software*. Na janela que abrirá, clique no botão *Add* localizado no canto superior esquerdo. A seguir, nomeie o Plugin como *ADT Plugin* e, no campo Location insira a seguinte url: <https://dl-ssl.google.com/android/eclipse/>. Para finalizar clique em *Ok* (Figura 6).

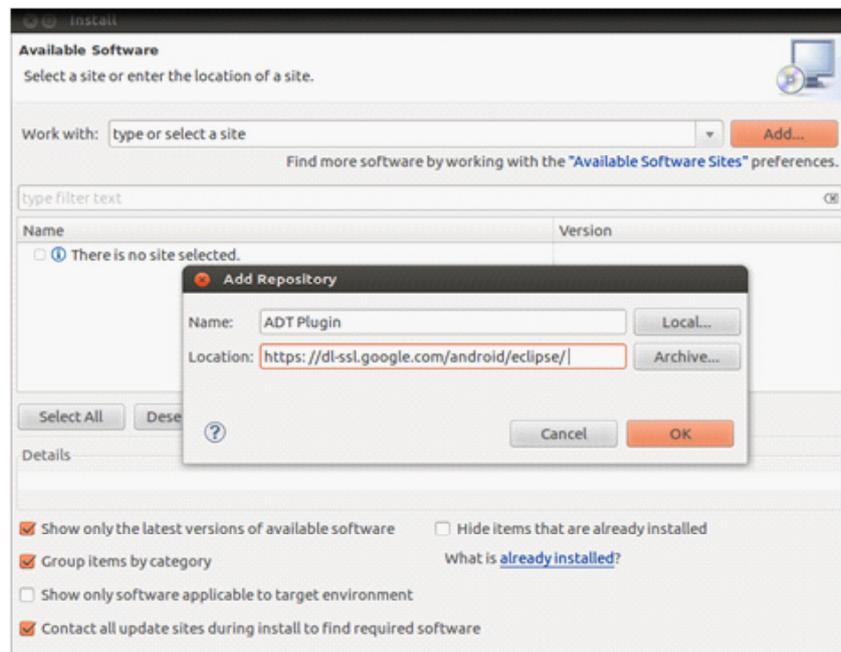


Figura 6. Adicionando o ADT plugin.

Espere até que o Eclipse reúna as informações necessárias para a instalação do plugin e, então marque a caixa de verificação *Developer Tools*. Clique em *Next* em todas as janelas que surgirem e a seguir, aceite os termos para então finalizar. Após finalizar a instalação, é necessário reiniciar a IDE do Eclipse.

Ao reiniciar o Eclipse, a janela *Welcome to Android Development* será exibida.

Recomenda-se desmarcar a opção *install the latest available version of Android APIs (support all the latest features)*, pois pode-se instalar a API mais nova posteriormente, se necessário.

Marque também a opção *Use existing SDKs* e informe o caminho – diretório - onde o SDK do Android está instalado (Figura 7).

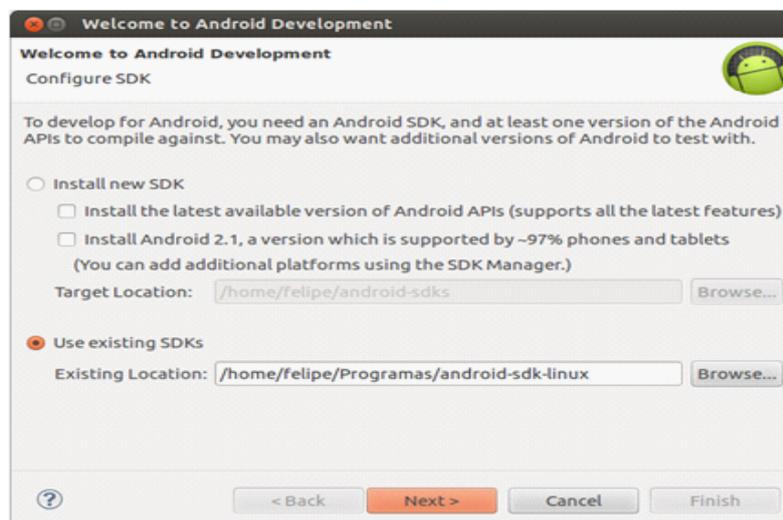


Figura 7. Indicando o diretório de instalação do SDK do Android.

Clique em *Next* e, se preferir, marque a opção de enviar estatísticas de uso para o Google. A seguir, finalize.

Como adicionar um novo Dispositivo Virtual Android

Através do menu *Window*, clique na opção *AVD*

Manager. Na janela que aparecerá, clique em *New...*

A seguir, preencha as informações da janela de acordo com suas necessidades. É importante salientar que pode-se adicionar ou remover quantos dispositivos virtuais Android for preciso (Figura 8).

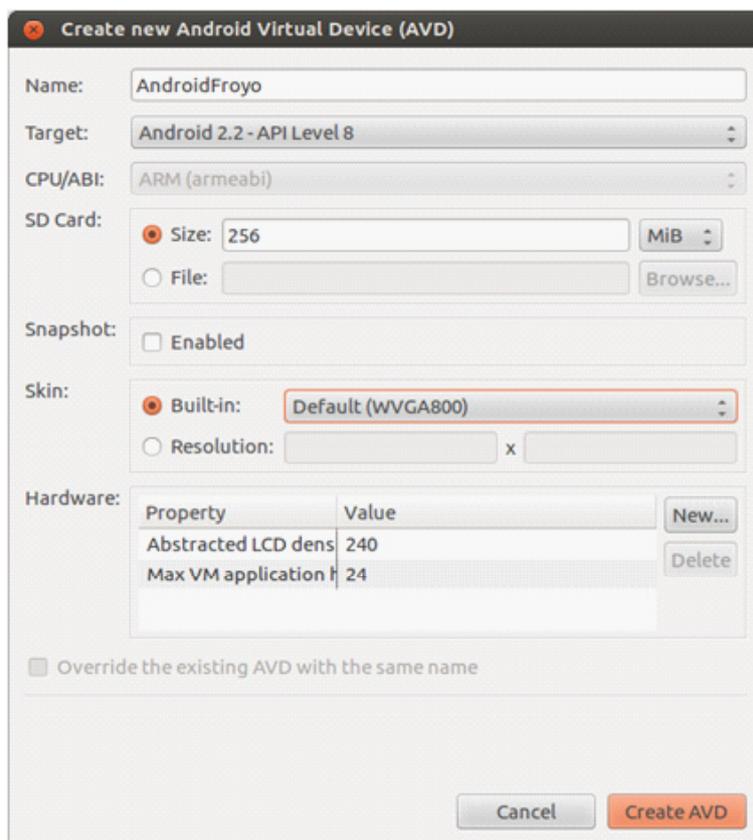


Figura 8. Configurando seu Dispositivo Virtual Android (AVD).

É importante destacar que deve-se Selecionar como *Target* uma *API level* como por exemplo *Android 2.2 – API Level 8* para assegurar maior compatibilidade com a maior número de modelos de *smartphones* e *tablets* do mercado. A seguir clique em *Create AVD* para finalizar.

Uma aplicação (App) que calcula o Índice de Massa Corporal (IMC)

Esta seção tratará de descrever através de etapas finitas e bem definidas, quais procedimentos deverão ser adotados para a criação de uma aplicação para

dispositivo móvel baseado no sistema operacional Android, que calcule o IMC de uma pessoa. É importante destacar que a abordagem do cálculo do IMC não é o foco deste trabalho, servindo muito mais como um veículo para demonstrar a metodologia de criação de aplicativos para dispositivos móveis que usam o sistema Android.

Criando o projeto

Através no menu *File*, aponte para *New* e clique em *Android Project*. Na janela *New Android Project* vamos configurar o nome do projeto como *AppIMC*. Clique em *Next* (Figura 9).

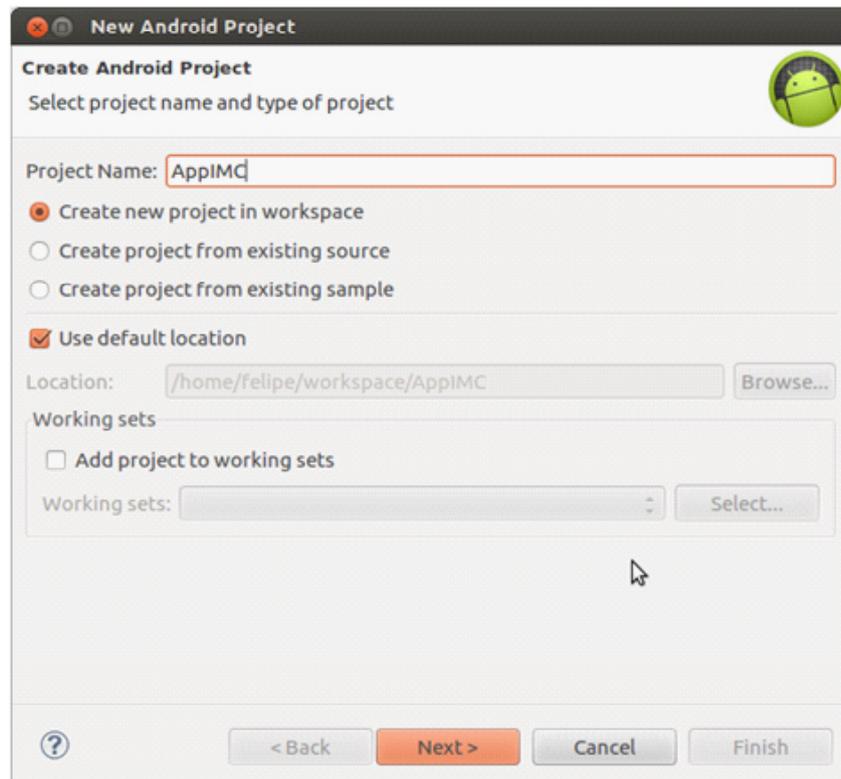


Figura 9. Criando um novo projeto Android.

Em seguida defina a versão do SDK mínima necessária para que o projeto funcione. Deixe a versão *Android 2.2 (API Level 8)* selecionada e clique em *Next*.

Configure os detalhes do projeto como na Figura 12 e clique em *Finish*.

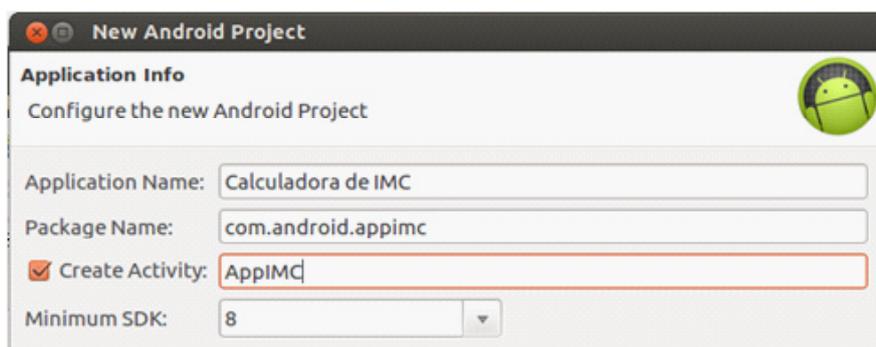


Figura 10. Definindo detalhes do projeto.

O arquivo main.xml

Criado o projeto, abra o arquivo *main.xml* que se encontra no diretório *res/layout/* (Figura 11).

O arquivo *main.xml* é responsável pela organização dos itens de interface de nossa aplicação Android.

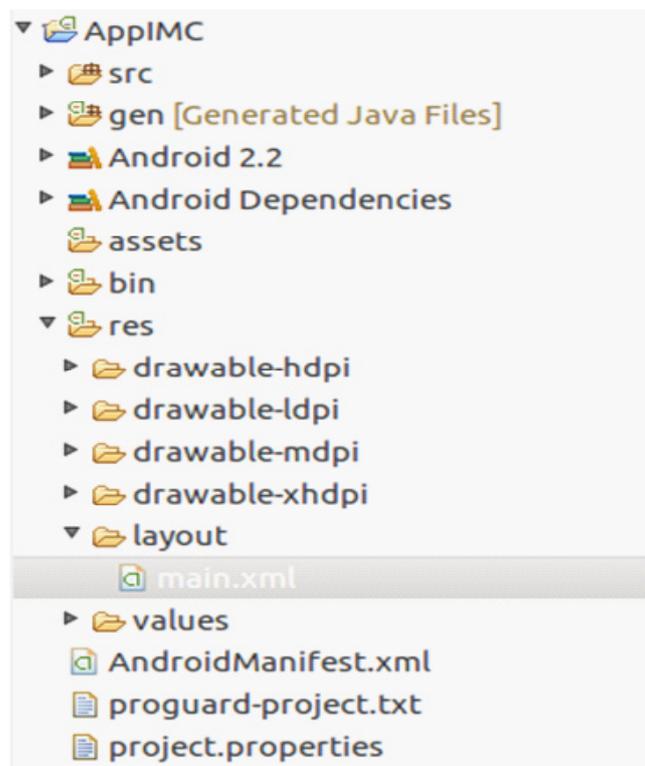


Figura 11. A estrutura de diretórios do projeto Android.

O arquivo será aberto no modo gráfico. Acesse o código fonte do arquivo através da aba *main.xml* presente na área inferior da janela (Figura 12).

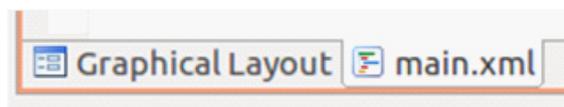


Figura 12. Alternando entre modo gráfico e código.

Em seguida visualize o código conforme visto na Figura 13:

```

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

    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello" />

</LinearLayout>

```

Figura 13. O código fonte do arquivo main.xml.

Altere o código do arquivo *main.xml* para que fique conforme abaixo:

Quadro. Alterando o código do arquivo *main.xml*.

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

    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/peso"/>

    <EditText
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:id="@+campo/peso"/>

    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/altura"/>

    <EditText
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:id="@+campo/altura"/>

    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/imc"/>

    <EditText
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:id="@+campo/imc"/>

    <Button
        android:id="@+botao/btcalcular"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/calcular"/>
</LinearLayout>
```

O arquivo *strings.xml*

O código apresentará alguns trechos grifados em vermelho como em `android:text="@string/peso"`,

indicando que a string `peso` ainda não foi declarada. Para realizar as declarações de variáveis necessárias, edite o arquivo *strings.xml*, presente no diretório `res/values/` deste projeto.

Quadro. Alterando o código do arquivo strings.xml.

```
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <string name="app_name">Calculadora de IMC</string>
    <string name="peso">Peso</string>
    <string name="altura">Altura</string>
    <string name="imc">IMC</string>
    <string name="calcular">Calcular</string>

</resources>
```

O arquivo AppIMC.java

Neste ponto, edite o arquivo *AppIMC.java* - encontrado no diretório

src/com.android.appimc/- que contém o código java de nossa aplicação. Altere o código do arquivo para que fique conforme abaixo:

Quadro. Alterando o código do arquivo AppIMC.java.

```
package com.android.appimc;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class AppIMC extends Activity {
    /** Called when the activity is first created. */

    EditText edpeso, edaltura, edimc;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        edpeso = (EditText) findViewById(R.campo.peso);
        edaltura = (EditText) findViewById(R.campo.altura);
        edimc = (EditText) findViewById(R.campo.imc);

        Button btMostrar = (Button) findViewById(R.botao.btcircular);
        btMostrar.setOnClickListener(new View.OnClickListener() {

            @Override
                public void onClick(View arg0) {
                    // TODO Auto-generated method stub
                    double peso = Double.parseDouble(edpeso.getText().toString());
                    double altura = Double.parseDouble(edaltura.getText().toString());
                    double imc = peso / (altura * altura);
                    String condicao;
```

```

edimc.setText(String.valueOf(imc));

        if(imc < 18.5)
            condicao = "Abaixo
do peso";
        else if(imc <= 24.9)
            condicao =
"Saudável";
        else if(imc <= 29.9)
            condicao = "Peso
em excesso";
        else if(imc <= 34.9)
            condicao =
"Obesidade Grau I";
        else if(imc <= 39.9)
            condicao =
"Obesidade Grau II";
        else
            condicao =
"Obesidade Grau III";

        Toast mostrar =
Toast.makeText(getApplicationContext(), condicao, Toast.LENGTH_LONG);
        mostrar.show();
    }
});
}
}

```

O arquivo R.java

O arquivo *R.java* presente no diretório *gen/com.android.appimc/* é o arquivo que liga os códigos java aos arquivos de resource de nosso projeto. Ele é um arquivo gerado e modificado automaticamente sempre que realizamos alguma alteração em nosso projeto. Recomenda-se não realizar alterações neste arquivo.

Executando o projeto

Para verificar como a aplicação funcionará em um dispositivo Android, não é condição necessária

dispor de um *smartphone* ou um *tablet* equipado com o sistema operacional da Google. Lembre-se que anteriormente, foi criado um AVD - Android virtual Device - que funcionará como um emulador de um dispositivo real. Para utilizar este recurso, clique na raiz de seu projeto e, através do menu *Run*, aponte para *Run* as e clique em *Android Application*. Espere até que o emulador Android carregue o sistema por completo e utilize sua primeira aplicação Android (Figura 14).

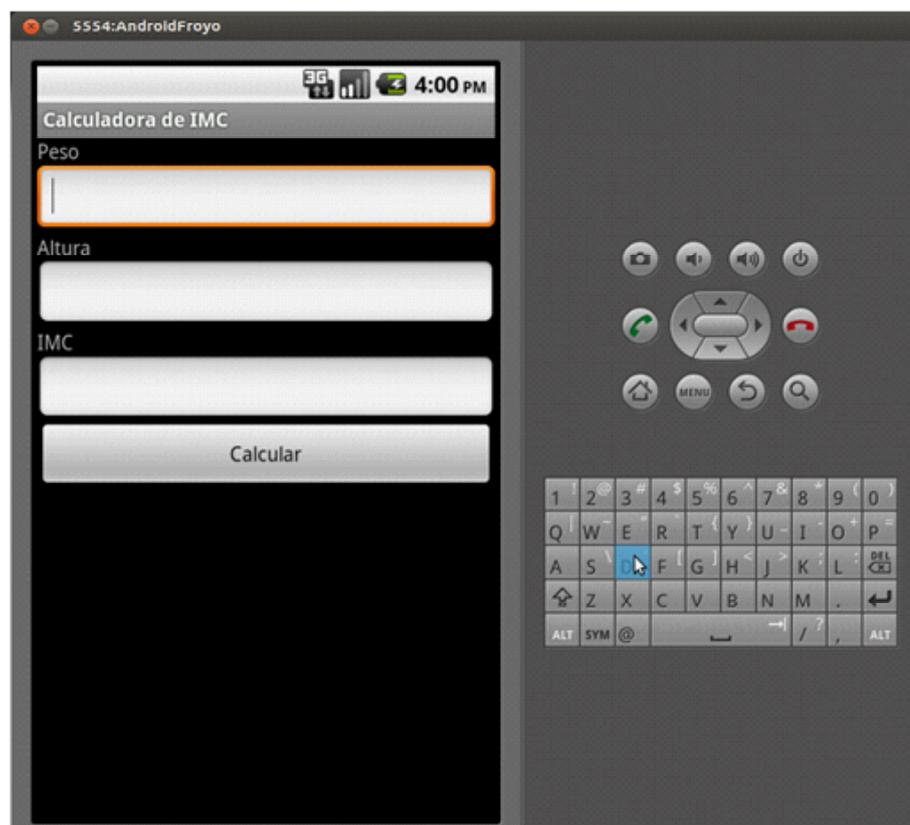


Figura 14. A aplicação rodando no emulador Android.

Considerações finais

O crescimento do mercado móvel vivido nos últimos anos tem possibilitado uso cada vez mais acentuado dos dispositivos móveis - como celulares, *smartphones* e *tablets* - e computadores. Desenvolvedores do mundo inteiro enxergaram este novo nicho de mercado e dia após dia, novas aplicações – referenciados comumente como *Apps* – são apresentadas adicionando novas funcionalidades que tiram proveito dos recursos de *hardware* e melhorias constantes nos sistemas operacionais móveis como o *Android* da Google e *iOS* da Apple. Espera-se que este trabalho tenha conseguido demonstrar, através de etapas supracitadas, como criar aplicações para dispositivos móveis baseados na plataforma *Android*.

A principal contribuição deste artigo é a abordagem e uma metodologia de desenvolvimento, baseado no sistema operacional Linux, de uma aplicação exemplo para dispositivos móveis que utilizam sistema Android.

Os autores pedem sinceras desculpas caso alguma das tecnologias ou linguagens citadas esteja levemente obsoletas, pois este artigo foi escrito no ano de 2012.

Referências

ComputerWorld. *Cresce uso de dados em dispositivos móveis, diz estudo*. <http://computerworld.uol.com.br/telecom/2011/11/29/cresce-uso-de-dados-em-dispositivos-moveis-diz-estudo/>. Acesso em 05/12/2011.

Deitel, H. M. & Deitel, P. J. (2003). *Java, como programar*. 4ª ed. Pearson

Prentice Hall, São Paulo.

Gargenta, M. (2011). *Learning Android: Building Applications for the Android Market*. 1.ed. Sebastopol, CA, O'Reilly Media.

Burnette, Ed. (2005). *Eclipse IDE Pocket Guide*. 1 ed. O'Reilly Media.

Lecheta, R. R. (2010). *Google Android: aprenda a criar aplicações para dispositivos móveis com o Android SDK*. 2. ed. rev. e ampl. São Paulo, SP: Novatec.

Android Developers. *ADT Plugin*. <http://developer.android.com/tools/sdk/eclipse-adt.html>. Acesso em 10/09/2012.