

Seu primeiro aplicativo para Android

Programame seu andróide

A plataforma Android, do Google, vai equipar com tecnologia de ponta diversos aparelhos celulares. Veja como é fácil começar a programar aplicativos nela.

por **Alessandro de Oliveira Faria**

Jean Scheijen - www.sxc.hu

O Android é uma plataforma de código aberto para dispositivos portáteis criado pelo Google em parceria com a Open Handset Alliance (OHA). Trata-se de uma aliança para o telefone celular aberto, composta por mais de 30 empresas do mercado de TI que apóiam soluções de código-fonte aberto, como Samsung, Intel, Motorola, Qualcomm e Telefónica. A OHA pretende repetir a estratégia da IBM com a fabricação do PC, composto por partes facilmente adquiridas no mercado.

Essa plataforma funciona como um sistema operacional como os já existentes *Symbian* e *Windows® Mobile*, com a diferença de ser baseado em Código Aberto. Com isso, qualquer desenvolvedor pode criar aplicativos para a plataforma Android. O kit de desenvolvimen-

to do Android provê ferramentas e chamadas via APIs para o desenvolvimento de aplicativos baseados na linguagem Java.

Em outras palavras, trata-se de um pacote com programas para celulares, já com um sistema operacional, *middleware*, aplica-



Figura 1 Camadas da plataforma Android.

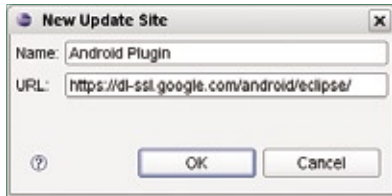


Figura 2 Especificação do site para o plugin

tivos e interface de usuário. Tais recursos permitirão que o Google e outras empresas ofereçam recursos mais ricos aos usuários que se encontram distantes de um computador de mesa.

Os principais recursos dessa plataforma móvel são a máquina virtual otimizada, o navegador integrado, a biblioteca 2D e 3D, o banco *SQLite* e o plugin para o *Eclipse*, chamado *ADT*.

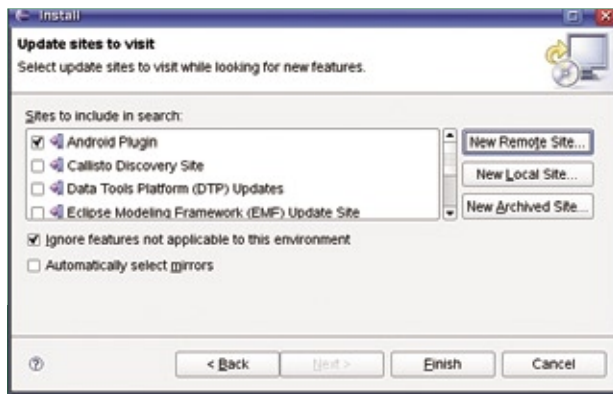


Figura 3 Seleção do novo site.

Arquitetura

A arquitetura do Android é dividida em cinco camadas, organizadas como mostra a **figura 1**: kernel Linux, bibliotecas, ambiente de execução, *framework* e aplicativo.

A camada do **kernel Linux** é composta pelo kernel 2.6 e se responsabiliza pelos serviços, segurança, gerenciamento de memória e processos, rede e drivers. Essa camada também é responsável pela abstração do hardware do dispositivo.

A segunda camada, a de **bibliotecas**, carrega consigo um conjunto de bibliotecas C/C++ utilizadas pelo sistema. Estão incluídas nesse conjunto a biblioteca C padrão (*Libc*) e também aquelas das áreas de

multimídia, visualização de camadas 2D e 3D, funções para navegadores web, funções para gráficos, funções de aceleração de hardware, renderização 3D, fontes bitmap e vetorizadas e funções de acesso ao banco

SQLite. Todos esses recursos estão disponíveis no framework para o desenvolvimento de aplicativos.

A pequena camada do **ambiente de execução** (*Android Runtime*, na **figura 1**) é uma instância da máquina virtual *Dalvik* criada para cada aplicação executada no Android. A *Dalvik* é uma máquina virtual com melhor desempenho, maior integração com a nova geração de hardware e projetada para executar várias máquinas virtuais paralelamente. Além disso, é otimizada para consumo mínimo de memória, bateria e CPU.

Sobre essas camadas, localiza-se a camada do **framework** (*Application Framework*, na **figura 1**). Nela, encontramos todas as APIs e os recursos utilizados pelos aplicativos, com classes visuais como botões e *views*, provedor de conteúdo (troca de recursos entre aplicativos) e gerenciadores de recursos, de notificação e de pacotes.

Acima de todas as outras camadas está a de **aplicativos**, na qual se encontram todos os aplicativos (escritos em Java) do Android, como cliente de email, navegador web, contatos e outros. Isso significa que, para desenvolver programas para a plataforma Android, vamos criar os aplicativos em Java na máquina virtual *Dalvik*.

Exemplo 1: Código do projeto HelloAndroid

```
01 package com.android.hello;
02
03 import android.app.Activity;
04 import android.os.Bundle;
05 import android.widget.TextView;
06
07 public class HelloAndroid extends Activity {
08     /** Called when the activity is first created. */
09     @Override
10     public void onCreate(Bundle savedInstanceState) {
11         super.onCreate(savedInstanceState);
12         TextView tv = new TextView(this);
13         tv.setText("Ola Mundo - Linux Magazine!");
14         setContentView(tv);
15     }
16 }
```

Instalação e configuração

Para instalarmos o Android, primeiramente devemos efetuar o download do ambiente de desenvolvimento (*Android SDK*) em [1]. Antes de baixar o arquivo, é necessário clicar em "I agree to the terms of the SDK License" para aceitar os termos da licença do SDK. Logo em seguida, deve-se selecionar o pacote correspondente ao sistema

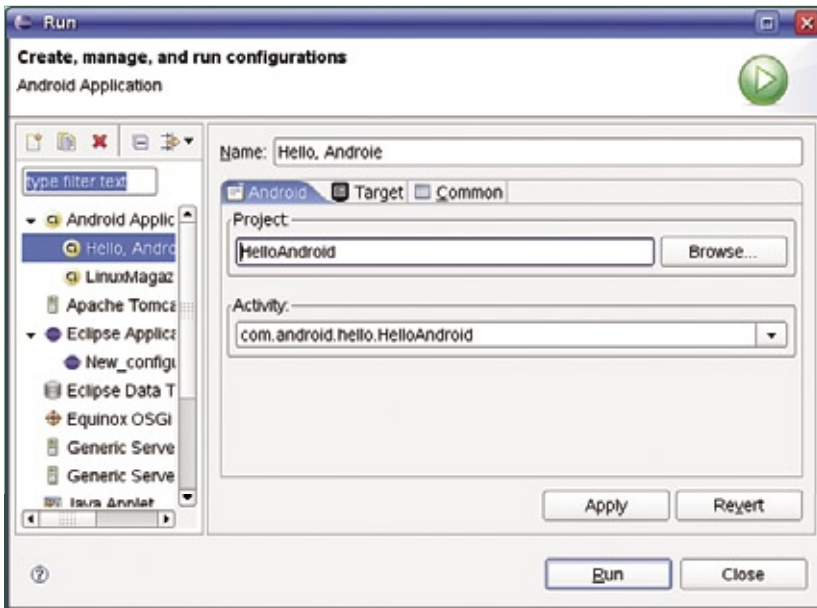


Figura 4 Preparação para execução do novo programa.

Olá, mundo

Vamos agora criar nosso primeiro aplicativo para o Android usando o Eclipse. Para isso, selecione a opção *File* no menu principal, depois a opção *New* e, logo após clique em *Project*. Em seguida, selecione o item *Android Project* e clique no botão *Next*.

Na janela de diálogo do projeto, digite os nomes do projeto, do pacote, da classe e da aplicação. Selecione as opções *Create new project in workspace* e *Use default location*, clicando no botão *Finish*, ao final. Se tudo estiver funcionando corretamente, um código semelhante ao do exemplo 1 será criado no projeto *HelloAndroid*, exceto pelas linhas 5, 12, 13 e 14, que devem ser acrescentadas ou editadas para que o resultado seja semelhante a ele.

Para executar o programa no emulador do Android, basta selecionar o item *Run | Run...* do menu principal. Na janela de diálogo (figura 4), selecione o projeto para execução e clique nos botões *Apply* e *Run*. Assim, o emulador



Figura 5 Resultado da execução do programa de teste.

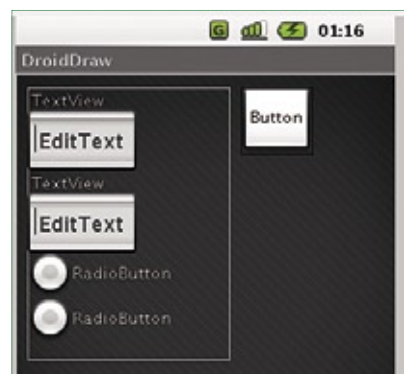


Figura 6 Layout do programa de conversão de câmbio.

operacional (Windows, Mac OS X ou Linux).

Após descompactar o arquivo baixado (formato ZIP), é recomendável acrescentar a pasta `tools/` criada no processo à variável `$PATH`, acrescentando ao arquivo `~/.bashrc` a linha:

```
export PATH=${PATH}:/pasta/de/
➔instalação/tools
```

Plugin ADT

O plugin Eclipse para desenvolvimento de aplicativos para o Android no IDE Eclipse se chama *Android Development Tools*, ou *ADT*. Na página do Android é mencionada a compatibilidade com as versões 3.2 e 3.3 do IDE. Para iniciar a instalação do ADT, inicie o Eclipse e selecione o item *Help | Software Updates | Find and Install...* no menu principal.

Na janela de diálogo *Install/Update*, selecione o item *Search for new features to install* e pressione o botão *Next*. Na janela de diálogo *Install*, clique no botão *New Remote Site...* e, na janela

que se abre (figura 2), digite o endereço <https://dl-ssl.google.com/android/eclipse/> para instalação do plugin e digite o nome que desejar (*Android Plugin*, na figura 2), confirmando, em seguida, com o botão *OK*.

Por último, basta selecionar o novo site (figura 3) e clicar em *Finish*, confirmando a instalação do plugin (que não é assinado – não precisa entrar em pânico) com *Install All*. Ao concluir a instalação, reinicie o Eclipse.

Após a reinicialização do Eclipse, atualize ou configure a localização do Android SDK no plugin ADT do Eclipse. Selecione o item “Windows” no menu principal e, ao abrir a janela de preferências, selecione a opção *Android* no painel da esquerda, informando a localização do Android SDK com o botão *Browse* no campo *SDK Location*. Para finalizar, pressione o botão *Apply* e, depois, *OK*.

Se tudo aconteceu como mostrado até esse ponto, o plugin do ADT já estará instalado e configurado corretamente no Eclipse.

será iniciado, nos permitindo ver esse primeiro programa em ação no Android (**figura 5**).

Sem Eclipse?

Também é possível escrever aplicativos para o Android sem usar o Eclipse, recorrendo apenas à linha de comando. Para isso, o Android SDK traz um script escrito em *Python* cha-

mado `activityCreator.py`. O comando adequado, nesse caso, seria:

```
activityCreator.py --out
➔HelloAndroid com.android.hello.
➔HelloAndroid
```

Em seguida, acesse a pasta `HelloAndroid/` (criada pelo script), onde se encontra-se o arquivo `build.xml`.

Partindo do princípio de que o *Apache Ant* esteja instalado corretamente, basta executar o comando `ant` para compilar o programa `HelloAndroid`.

Layout

No mundo dos aplicativos embarcados, a aparência tem uma importância ainda maior sobre a usabilidade. O *DroidDraw* é um editor de layout escrito em Java que possibilita a montagem de interfaces gráficas e sua gravação em um arquivo XML para posterior utilização no aplicativo. Essa ferramenta é muito útil para acelerar o desenvolvimento de aplicativos na plataforma Android.

O *DroidDraw* pode ser usado online[2] ou baixado a partir do site [3]. Para instalá-lo, basta descompactar o arquivo baixado, entrar no diretório criado por ele e executar o script `droiddraw.sh`.

Câmbio

Para explorar o *DroidDraw*, vamos criar um aplicativo para conversão de valores em reais para dólares.

Após iniciar o programa, o primeiro passo é selecionar a aba *Layouts* e, em seguida, o componente *LinearLayout*. Na aba *Properties*, altere a dimensão para 200 pixels de largura e 130 de altura, pressionando o botão *Apply* logo em seguida. Depois, na aba *Widgets*, arraste dois objetos *TextView*, dois *EditText*, dois *RadioButton* e um *Button* para a área da interface e crie um layout como na **figura 6**.

Novamente na aba *Properties*, altere o campo `id`

Exemplo 2: Aplicativo de câmbio

```
package com.android.lm;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.RadioButton;
import android.widget.TextView;

public class HelloLM extends Activity {
    TextView dolar;
    TextView real;
    RadioButton dtor;
    RadioButton rtod;
    Button convert;

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        dolar = (TextView)findViewById(R.id.dolar);
        real = (TextView)findViewById(R.id.real);
        dtor = (RadioButton)findViewById(R.id.dtor);
        dtor.setChecked(true);
        rtod = (RadioButton)findViewById(R.id.rtod);

        convert = (Button)findViewById(R.id.convert);
        convert.setOnClickListener(new Button.OnClickListener() {
            public void onClick(View v) {
                if (dtor.isChecked()) {
                    convertDolarToReal();
                }
                if (rtod.isChecked()) {
                    convertRealToDolar();
                }
            }
        });
    }

    protected void convertDolarToReal() {
        double val = Double.parseDouble(dolar.getText().toString());
        real.setText(Double.toString(val*1.8));
    }

    protected void convertRealToDolar() {
        double val = Double.parseDouble(real.getText().toString());
        dolar.setText(Double.toString(val/1.8));
    }
}
```

do primeiro objeto *EditText* para `@+id/dolar`, o do segundo *EditText* para `@+id/real`, o do primeiro *RadioButton* para `@+id/dor`, o do segundo *RadioButton* para `@+id/rtod` e o do *Button* para `@+id/convert`. Depois disso, basta clicar sobre o botão *Generate* para criar o arquivo do layout na janela *Output*.

Com o layout gerado, crie uma nova aplicação (com o Eclipse ou pelo script) e substitua o conteúdo do arquivo `res/layout/main.xml` pela estrutura XML gerada pelo DroidDraw. Depois, na pasta `src`, localize o arquivo `nome-da-classe.java` e insira o código do **exemplo 2** no arquivo fonte. Para visualizar o aplicativo em funcionamento, basta executar o programa com a opção *Run*, como mencionado anteriormente.

Emulador

O Android SDK traz ainda um prático emulador. Com ele, o desenvolvedor tem à disposição um aparelho celular emulado, capaz de executar todos os aplicativos desenvolvidos para a plataforma. Isso é ótimo para realizar testes sem um dispositivo físico. Todas as funções de um celular estão disponíveis, exceto aquelas relacionadas a chamadas telefônicas.

Para executar o emulador do Android, basta executar o comando `emulator`, que se encontra na pasta `tools/` do SDK. Por padrão, ele traz quatro “temas”, chamados de *skins*. O que muda entre os skins é apenas a resolução da tela. Há as opções *QVGA* (320x240) e *HVGA* (480x320), nas orientações paisagem (sufixo *L*) e retrato (sufixo *P*). Para usar a resolução *HVGA* em formato paisagem, basta iniciar o emulador com o parâmetro `-skin HVGA-L`.

Exemplo 3: Shell do Android

```
$ adb shell
# ls -l
drw-rw-rw- root    root    2008-04-14 06:29 cache
drwxr-xr-x root    root    2008-04-14 06:29 d
-rwxr-xr-x root    root    1970-01-01 00:00 init
drwxr-xr-x root    root    1970-01-01 00:00 etc
drwxr-xr-x root    root    1970-01-01 00:00 var
drwxrwx-x-x system system 2007-11-11 20:59 data
drwxr-xr-x root    root    2008-02-29 01:19 system
drwxr-xr-x root    root    1970-01-01 00:00 sys
drwxrwxrwt root    root    2008-04-14 06:49 tmp
dr-xr-xr-x root    root    1970-01-01 00:00 proc
drwxr-xr-x root    root    1970-01-01 00:00 sbin
drwx-- root    root    1970-01-01 00:00 root
drwxr-xr-x root    root    2008-04-14 06:29 dev
```

Depuração

O ADB, ou *Android Debug Bridge*, é um gerenciador de depuração poderoso. Este artigo ficaria muito extenso caso fossem mencionadas todas as opções e recursos disponíveis na ferramenta; por isso, serão abordadas apenas suas funções principais.

Para listar os emuladores disponíveis, usa-se o comando `adb devices`. A instalação de aplicativos também é possível e usa-se o comando `adb install /caminho/do/aplicativo.apk`. A desinstalação é igualmente fácil, pois requer apenas o comando `adb shell rm data/app/aplicativo.apk`. Como esse último comando leva a crer, é possível acessar uma shell do Android com o comando `adb shell` (**exemplo 3**).

O Android tem, embutido, um banco de dados *SQLite*, que se pode acessar, a partir da shell, com o comando:

```
# sqlite3 /data/data/com.
➤ example.
google.rss.rssexample/databases/
➤ rssitems.db
Ao vivo
```

Quem desejar poupar esforços de instalação do Android SDK pode recorrer ao *VD_Android*[4], uma distribuição Linux em *Live*

CD que já traz o SDK instalado, assim como o IDE Eclipse e o aplicativo DroidDraw. A distribuição oferece ainda a opção de instalação no disco rígido ou em um *pendrive*. O *VD_Android* é baseado no *Debian Lenny* e utiliza como ambiente desktop padrão o *Xfce4*.

Conclusão

Para quem deseja ter contato com o Android, este artigo é um bom ponto de partida. A plataforma Android é poderosa e, como se vê, fácil de usar. Esperamos que ele ajude a iniciar uma longa caminhada de aprendizado. ■

Mais informações

[1] Download da SDK do Android: <http://code.google.com/intl/pt-BR/android/download.html>

[2] DroidDraw online: <http://www.droiddraw.org/>

[3] Download do DroidDraw: <http://droiddraw.googlecode.com/files/droiddraw-r1b8.tgz>

[4] *VD_Android*: <http://tinyurl.com/6z2c3v>