



Webinar SDK devices Android

Boris Adrián Torrado Bonilla
Solutions Architect

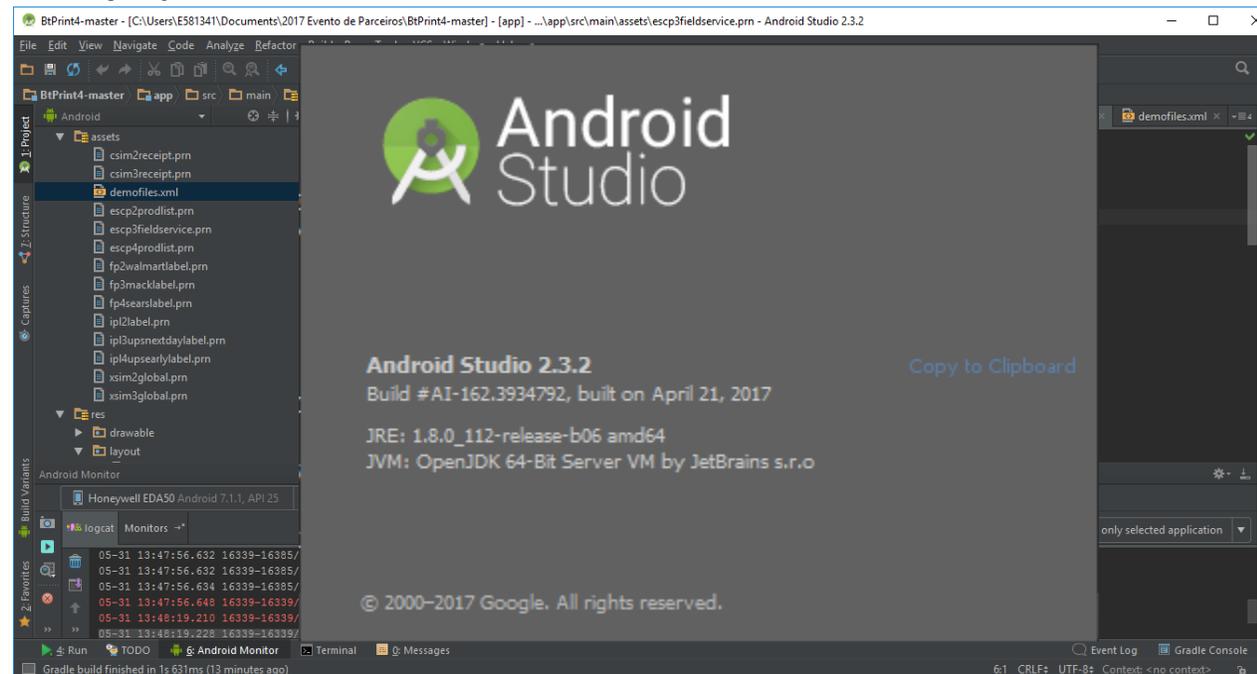
Honeywell

Conteúdo



Ambiente de Desenvolvimento

- Descarregar o [Android Studio](#) (versión 2.3.2, deste [link](#). É um instalador .exe de 1.9GB).
- Instalar os SDKs correspondentes aos API Level 19,21,22, 23, 24 e 25. Esta parte aparece em um dos passos da instalação (Next, Next, Next,... Finish).
- Instalar, caso não esteja presente o JDK 1.8.X

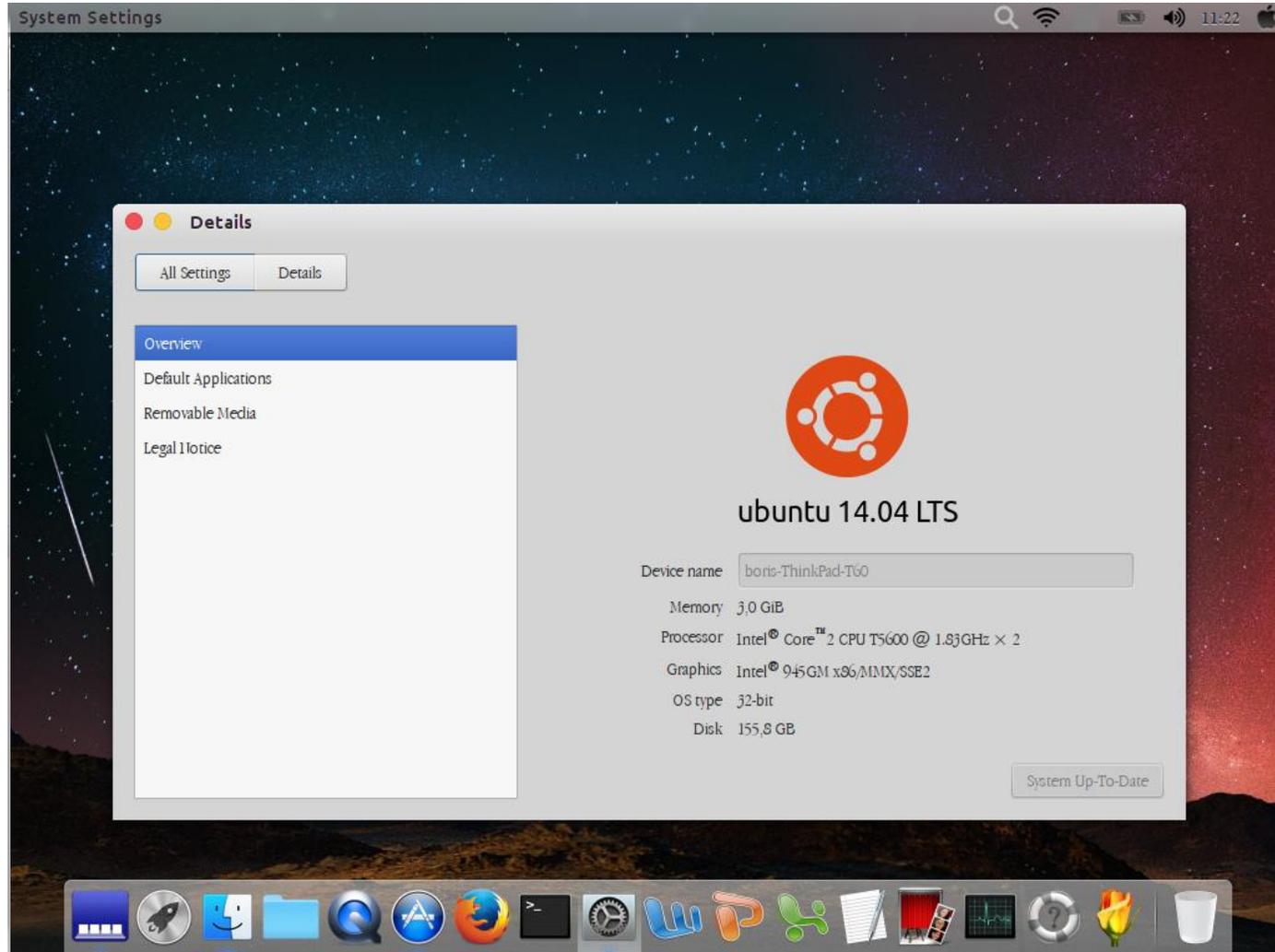


Compatibilidade entre Plataformas

Nosso SDK é compatível tanto com Windows, quanto Mac Os e Linux

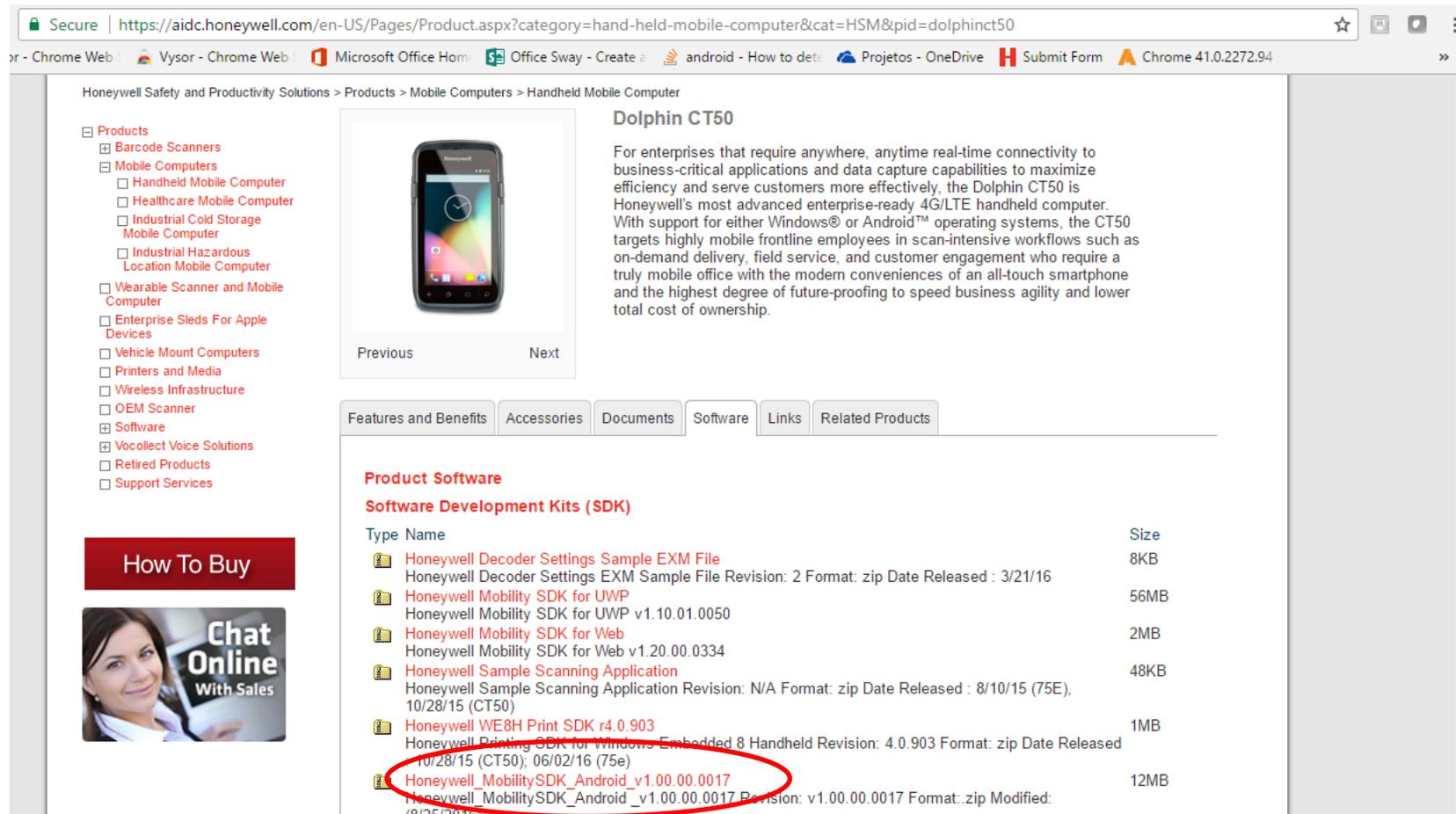
The screenshot displays the Android Studio environment. The top window shows the 'Android Device Monitor' for a device named 'honeywell-ct50-1514640019'. The CPU load section features a pie chart with a legend listing processes such as 'system_server (user)', 'system_server (kernel)', 'adb (user)', 'adb (kernel)', 'com.google.process.location (user)', 'com.google.process.location (kernel)', 'sensors.qcom (user)', 'sensors.qcom (kernel)', 'mediaserver (user)', 'mediaserver (kernel)', 'app_process (user)', and 'app_process (kernel)'. The LogCat window shows system logs with columns for 'Le Time', 'PID', 'TID', 'Application', 'Tag', and 'Text'. The bottom right shows a mobile application interface with buttons for 'Client Control', 'Enable Light & Aim', 'Enable Aim', 'Enable Light', and 'Kill Process KeyRemap'.

Compatibilidade entre Plataformas



Descarregando nosso SDK

O exemplo BarcodeExample se encontra dentro do SDK Android para o Dolphin CT50 ([Honeywell_MobilitySDK_Android_v1.00.00.0017](https://aidc.honeywell.com/en-US/Pages/Product.aspx?category=hand-held-mobile-computer&cat=HSM&pid=dolphinct50)) disponível no nosso [site](#), na Aba Software, baixo a seção Software Development Kits (SDK).



Honeywell Safety and Productivity Solutions > Products > Mobile Computers > Handheld Mobile Computer

Dolphin CT50

For enterprises that require anywhere, anytime real-time connectivity to business-critical applications and data capture capabilities to maximize efficiency and serve customers more effectively, the Dolphin CT50 is Honeywell's most advanced enterprise-ready 4G/LTE handheld computer. With support for either Windows® or Android™ operating systems, the CT50 targets highly mobile frontline employees in scan-intensive workflows such as on-demand delivery, field service, and customer engagement who require a truly mobile office with the modern conveniences of an all-touch smartphone and the highest degree of future-proofing to speed business agility and lower total cost of ownership.

Previous Next

Features and Benefits | Accessories | Documents | **Software** | Links | Related Products

Product Software

Software Development Kits (SDK)

Type	Name	Size
File	Honeywell Decoder Settings Sample EXM File Honeywell Decoder Settings EXM Sample File Revision: 2 Format: zip Date Released : 3/21/16	8KB
File	Honeywell Mobility SDK for UWP Honeywell Mobility SDK for UWP v1.10.01.0050	56MB
File	Honeywell Mobility SDK for Web Honeywell Mobility SDK for Web v1.20.00.0334	2MB
File	Honeywell Sample Scanning Application Honeywell Sample Scanning Application Revision: N/A Format: zip Date Released : 8/10/15 (75E), 10/28/15 (CT50)	48KB
File	Honeywell WE8H Print SDK r4.0.903 Honeywell Printing SDK for Windows Embedded 8 Handheld Revision: 4.0.903 Format: zip Date Released 10/28/15 (CT50); 06/02/16 (75e)	1MB
File	Honeywell_MobilitySDK_Android_v1.00.00.0017 Honeywell_MobilitySDK_Android_v1.00.00.0017 Revision: v1.00.00.0017 Format: .zip Modified: (8/25/2016)	12MB

How To Buy

Chat Online With Sales

Descarregando nosso SDK

author-hsmftp.honeywell.com

Apps Vysor - Chrome Web Vysor - Chrome Web Microsoft Office Home Office Sway - Create android - How to dete Projetos - OneDrive Submit Form Chrome 41.0.2272.94

• [Contact Us](#)

Computer Devices

- Handheld
 - CK3R CK3
 - CN51
 - Android
 - Arcl
 - Cur
 - Android
 - Arcl
 - Cur
 - WEH6.!
 - CN70 CN7
 - Dolphin 60
 - Dolphin 61
 - Dolphin 65
 - Dolphin 70e

Honeywell Software Download Manager

Download

Start Pause Remove Settings Help

File	Size	Status	Progress
00003102_Intermec-sdk-opt2.zip (Completed)	15.35 MB	15.35 MB	100%
Honeywell Mobility SDK Android v1.00.00.0011.zip...	11.64 MB	11.64 MB	100%
Honeywell_MobilitySDK_Android_v1.00.00.0017.zi...	11.58 MB	2.55 MB	21.99%

Download

Download

Rate: 366.43 kbps Version : 1.6

printing progress via events

Descarregando nosso SDK

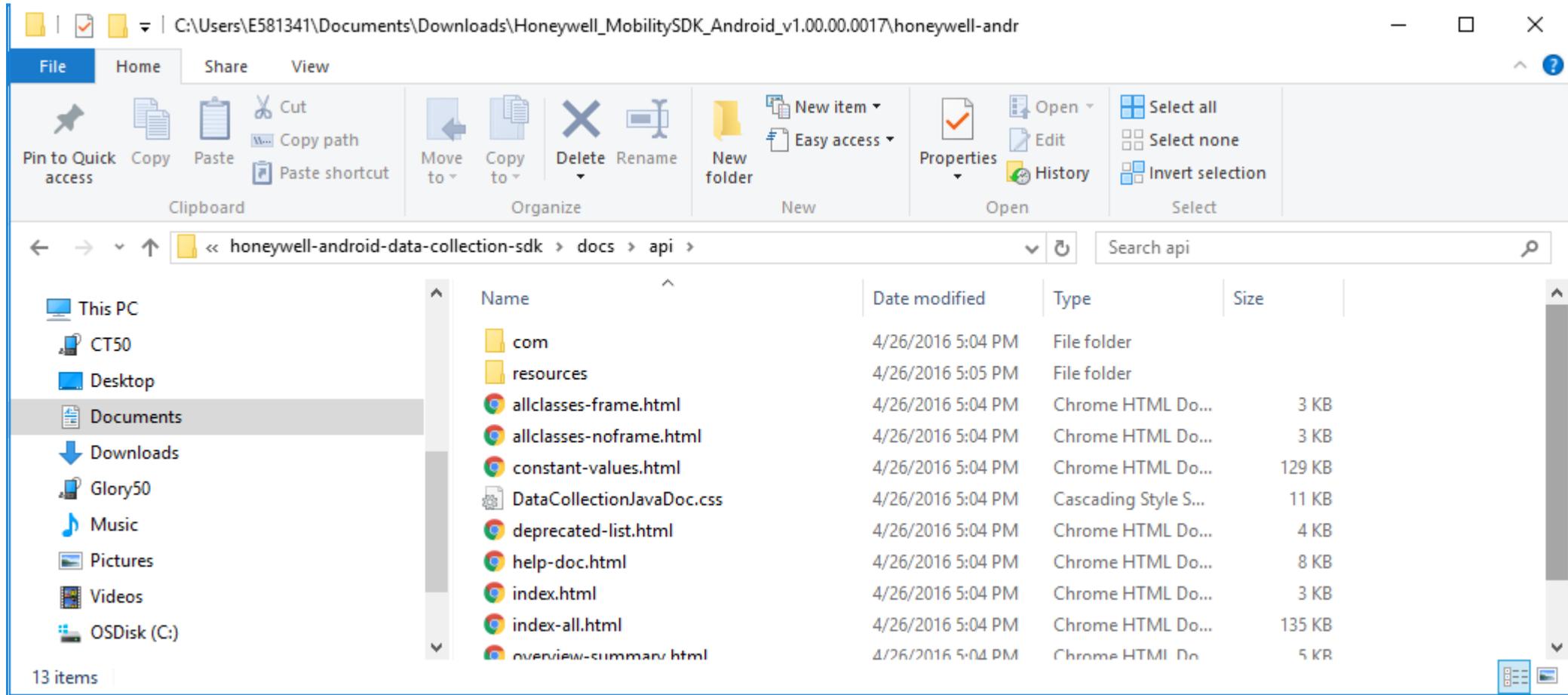
Em conclusão:

Para versão Android 4.4.4 Kit Kat (CT50, 75E, EDA50/K)
Honeywell Mobility SDK Android v1.00.00.0011.zip

Para versão Android 6.0 Marshmallow (CT50, 75E, CN51)
Honeywell_MobilitySDK_Android_v1.00.00.0017.zip

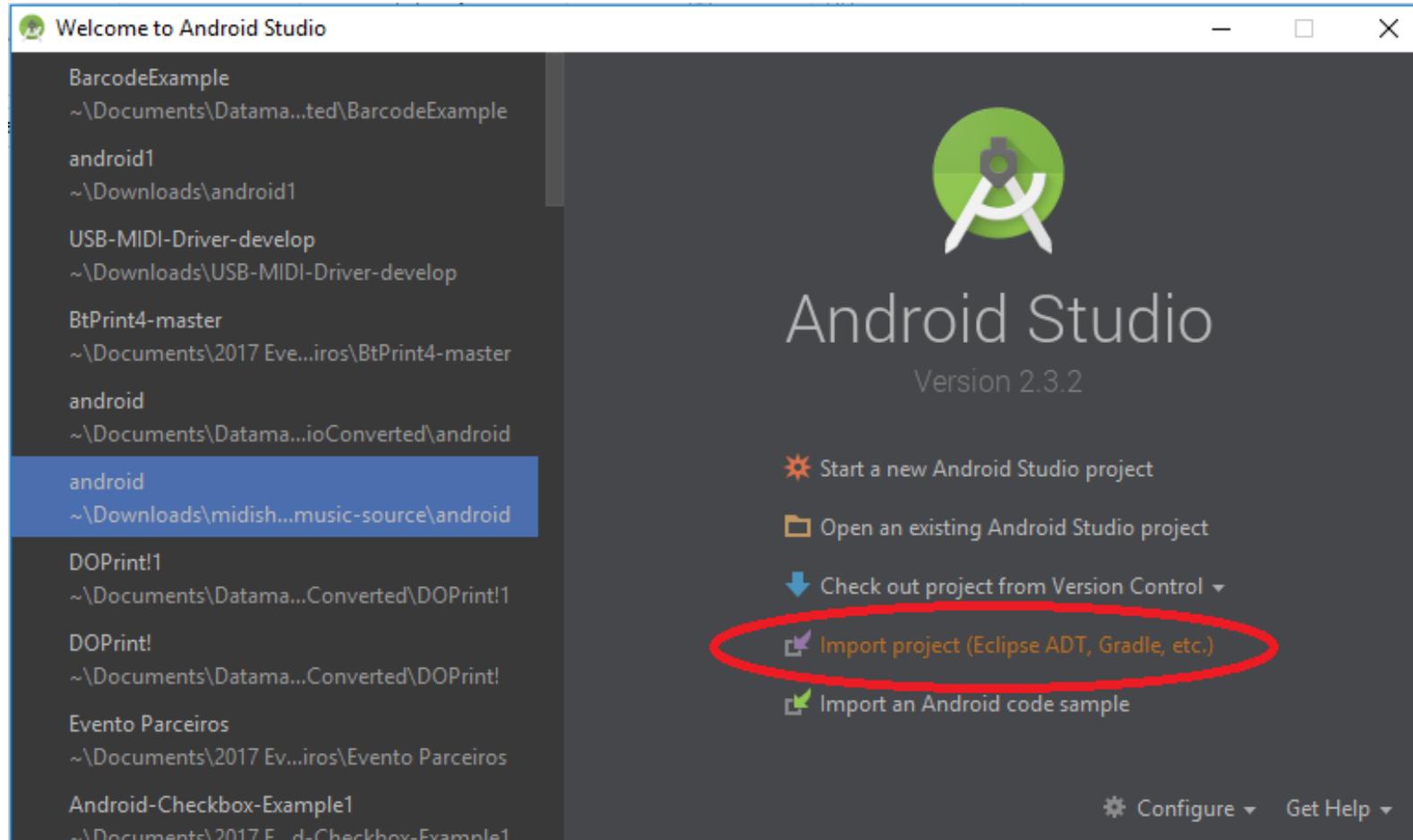
Para versão Android 4.2.2 Jelly Bean (CN51)
Android Platform SDK – CN51 ver. 1.00

Descarregando o nosso SDK



Importar um exemplo no Ambiente Android Studio

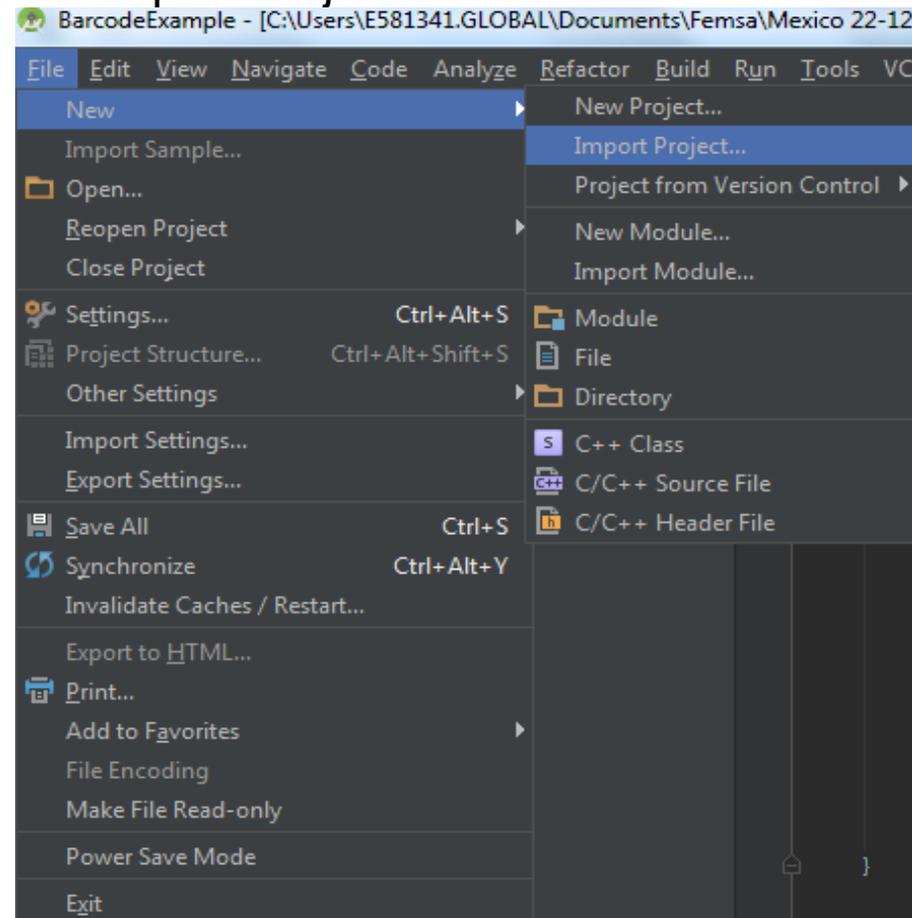
Selecionar a opção Import Project (Eclipse ADT, Gradle, etc.)



Importar um exemplo no Ambiente Android Studio

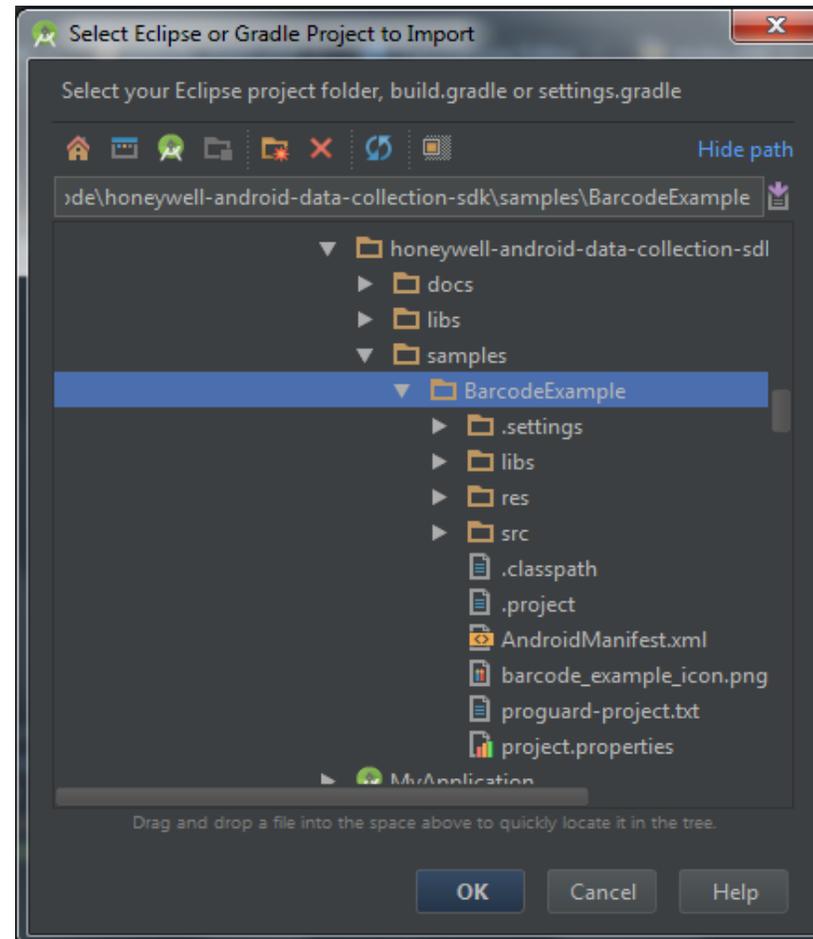
Ou se já esta aberto no ambiente com outro projeto...Para abrir corretamente este exemplo, devemos importá-lo da seguinte forma:

No Menú de opções File>New>Import Project...



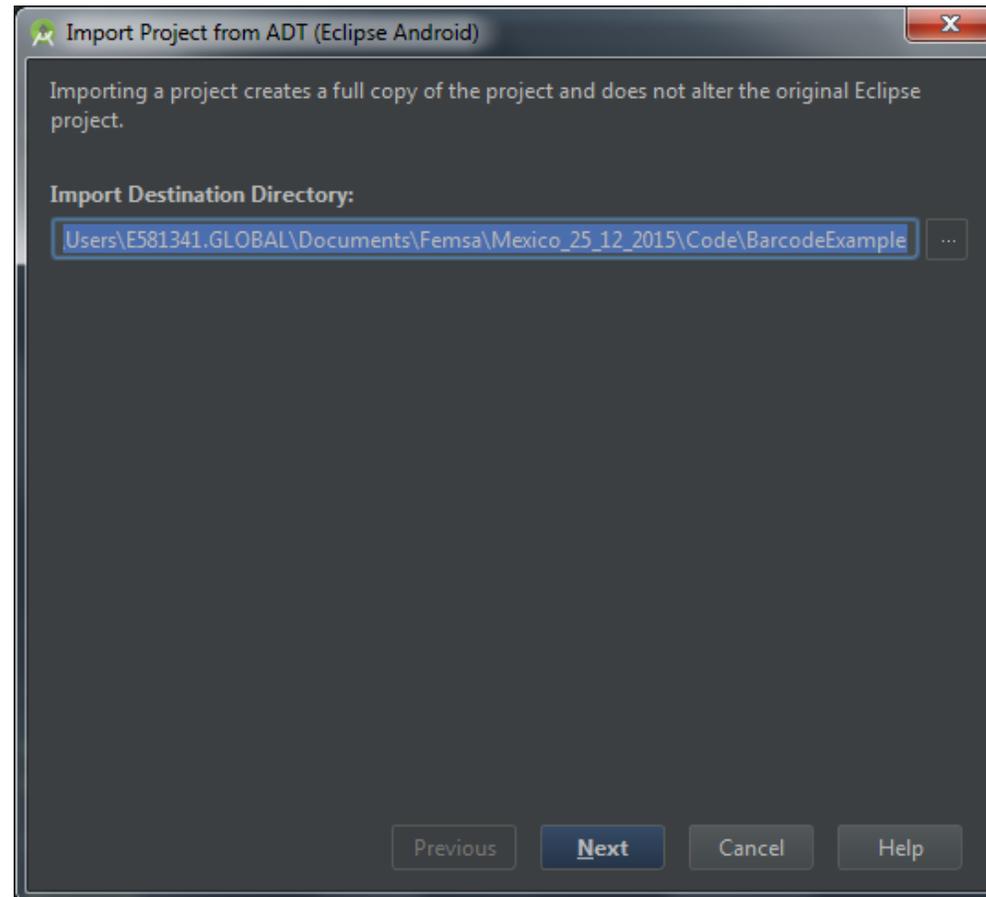
Importar um exemplo no Ambiente Android Studio

Devemos procurar na rota do exemplo



Importar un ejemplo al Ambiente Android Studio

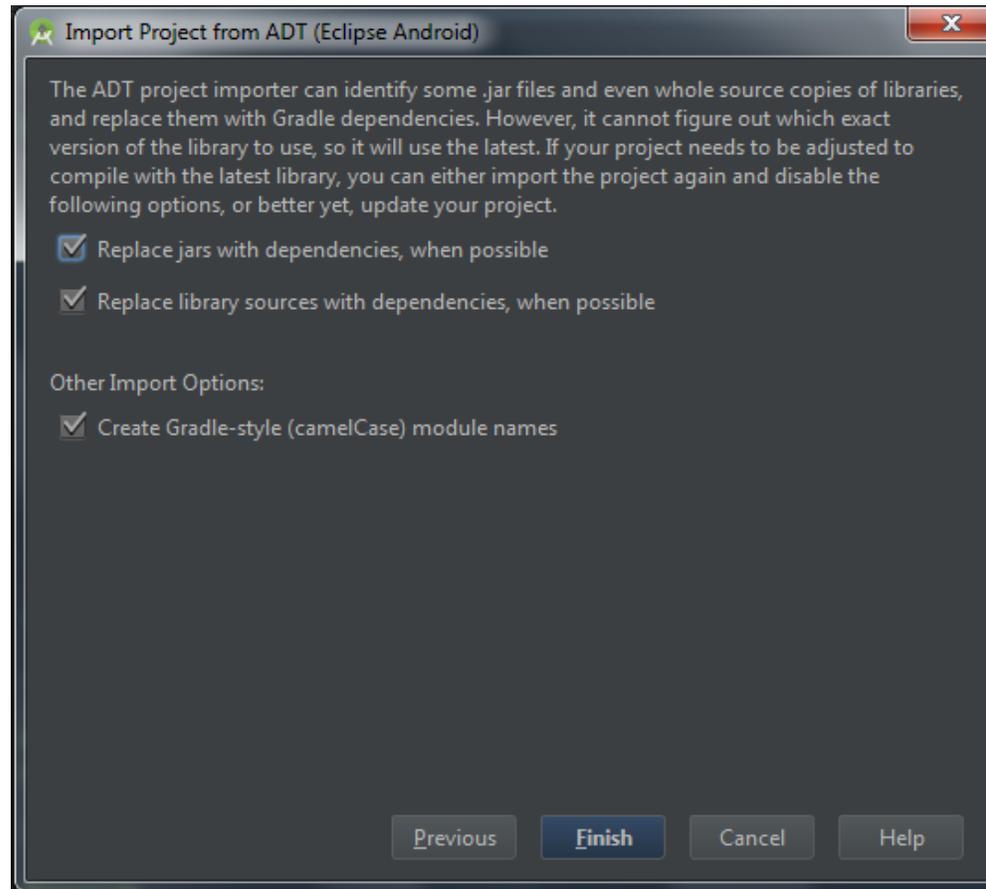
E escolher o diretório final onde será gerado o novo projeto importado:



Importar um exemplo no Ambiente Android Studio

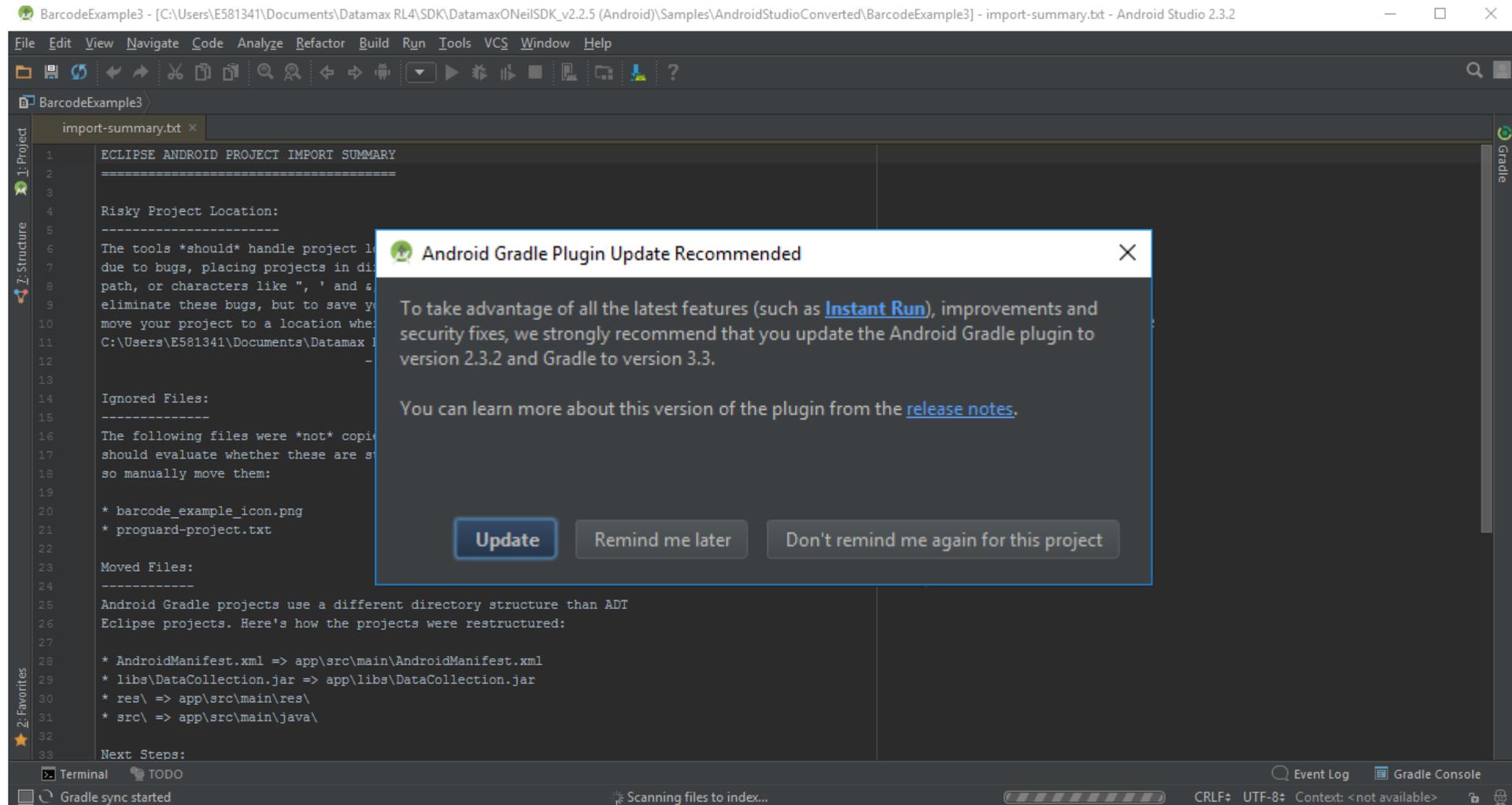
Ao pressionar o botão “Next”, mostrará uma janela com as opções para substituir todas as dependências com a nova estrutura em [Gradle](#), as quais devemos deixar selecionadas.

Pressionamos o botão “Finish”:



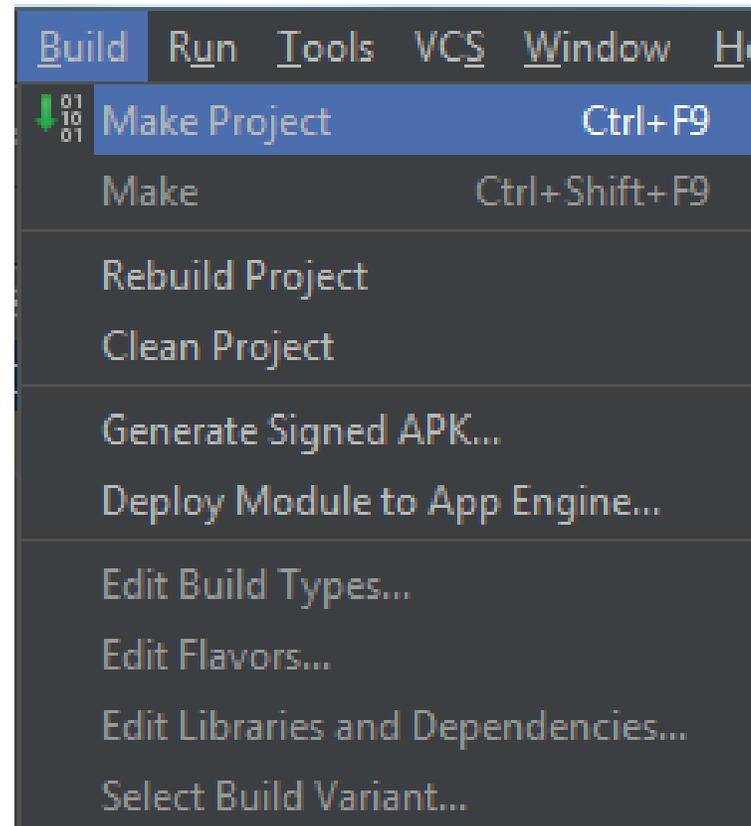
Importar um exemplo no Ambiente Android Studio

O IDE demorará um pouco gerando os arquivos a serem utilizados no ambiente



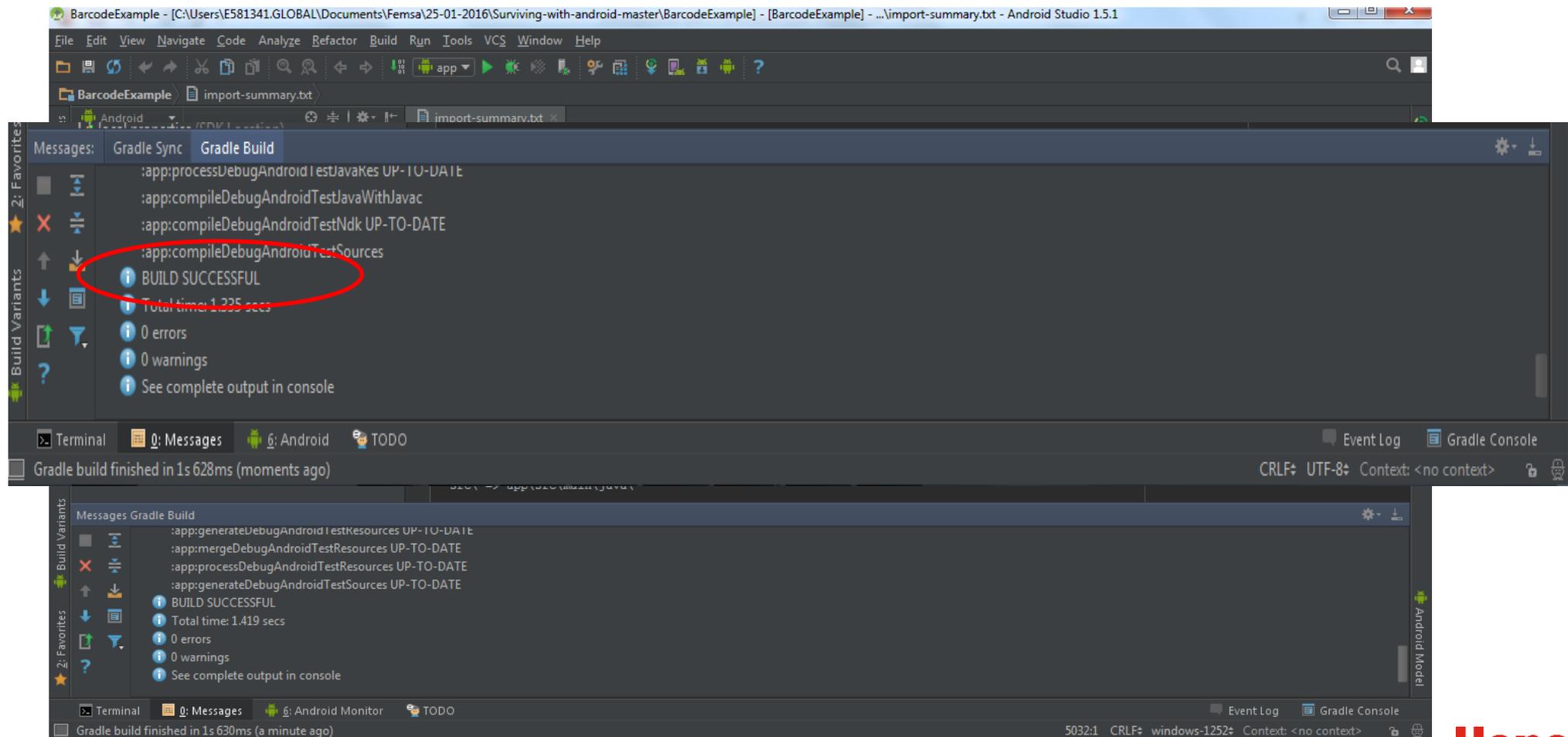
Compilar o exemplo disponível no nosso SDK

No Menú de Opções selecionamos Build, Make Project para gerar o executável que será transferido ao nosso device



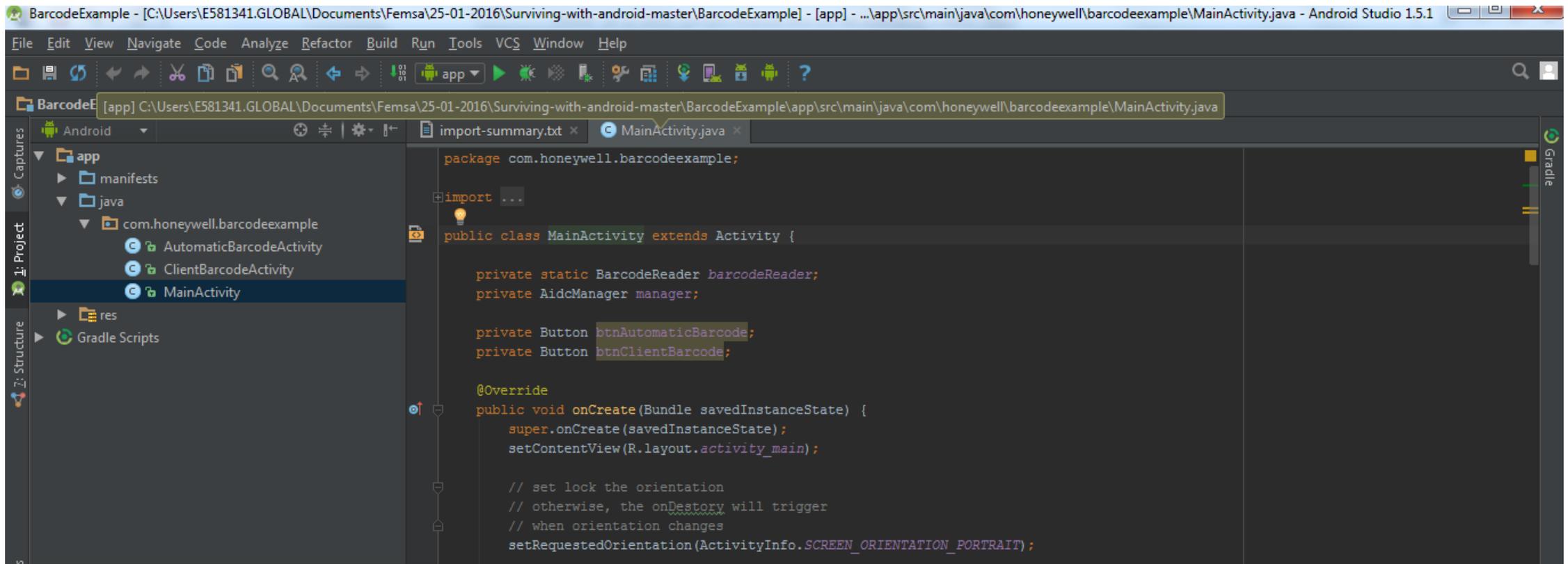
Compilar o exemplo disponível no nosso SDK

E na parte inferior poderemos verificar que o programa foi gerado sem erros, pressionando a aba/botão “Messages”:



Estrutura de um Projeto Android

Arquivos de código fonte extensão .java



```
BarcodeExample - [C:\Users\E581341.GLOBAL\Documents\Femsa\25-01-2016\Surviving-with-android-master\BarcodeExample] - [app] - ...\app\src\main\java\com\honeywell\barcodeexample\MainActivity.java - Android Studio 1.5.1
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
BarcodeE [app] C:\Users\E581341.GLOBAL\Documents\Femsa\25-01-2016\Surviving-with-android-master\BarcodeExample\app\src\main\java\com\honeywell\barcodeexample\MainActivity.java
Android
app
  manifests
  java
    com.honeywell.barcodeexample
      AutomaticBarcodeActivity
      ClientBarcodeActivity
      MainActivity
  res
  Gradle Scripts
import-summary.txt x MainActivity.java x
package com.honeywell.barcodeexample;

import ...

public class MainActivity extends Activity {

    private static BarcodeReader barcodeReader;
    private AidcManager manager;

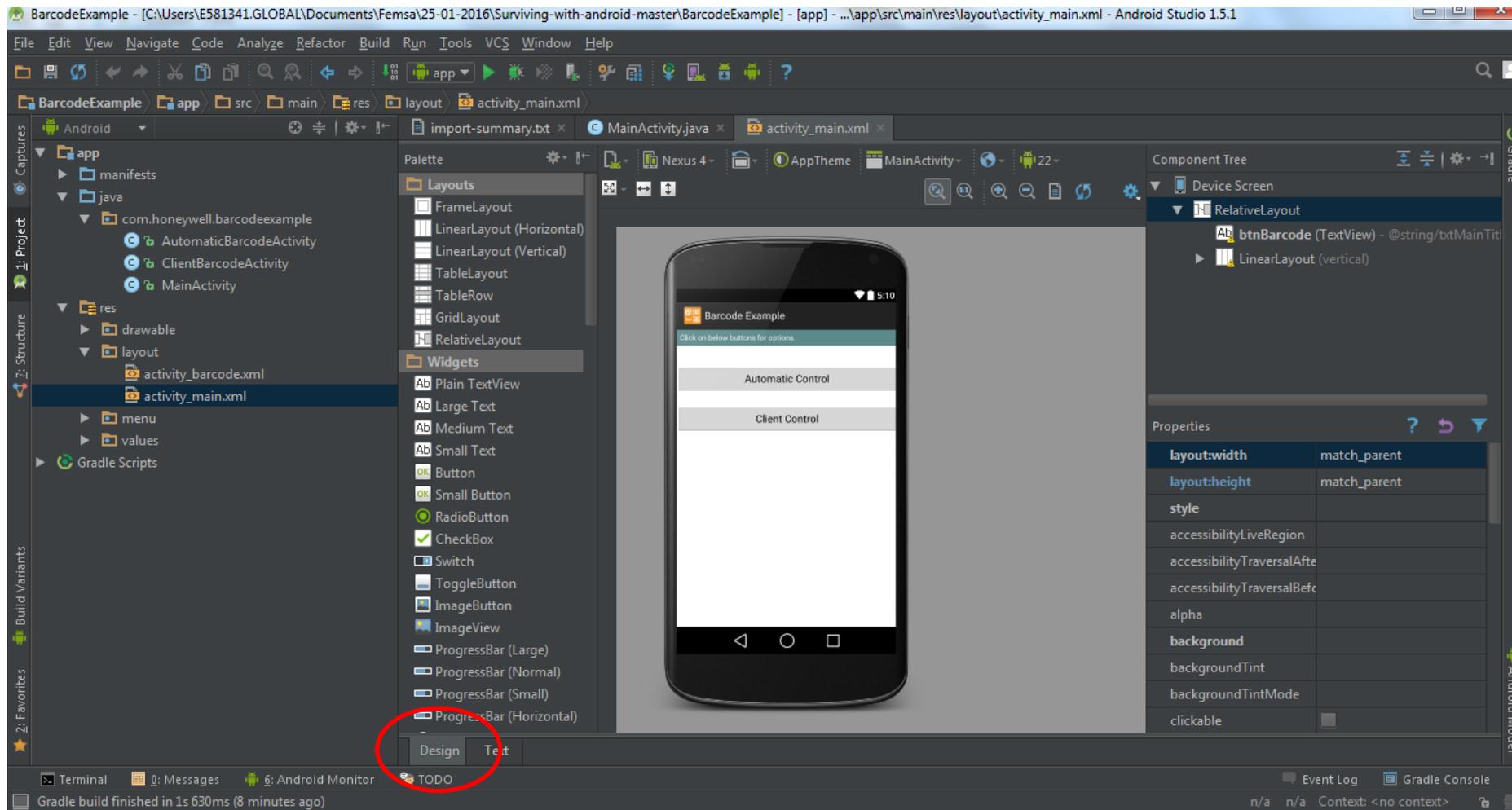
    private Button btnAutomaticBarcode;
    private Button btnClientBarcode;

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

        // set lock the orientation
        // otherwise, the onDestroy will trigger
        // when orientation changes
        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
    }
}
```

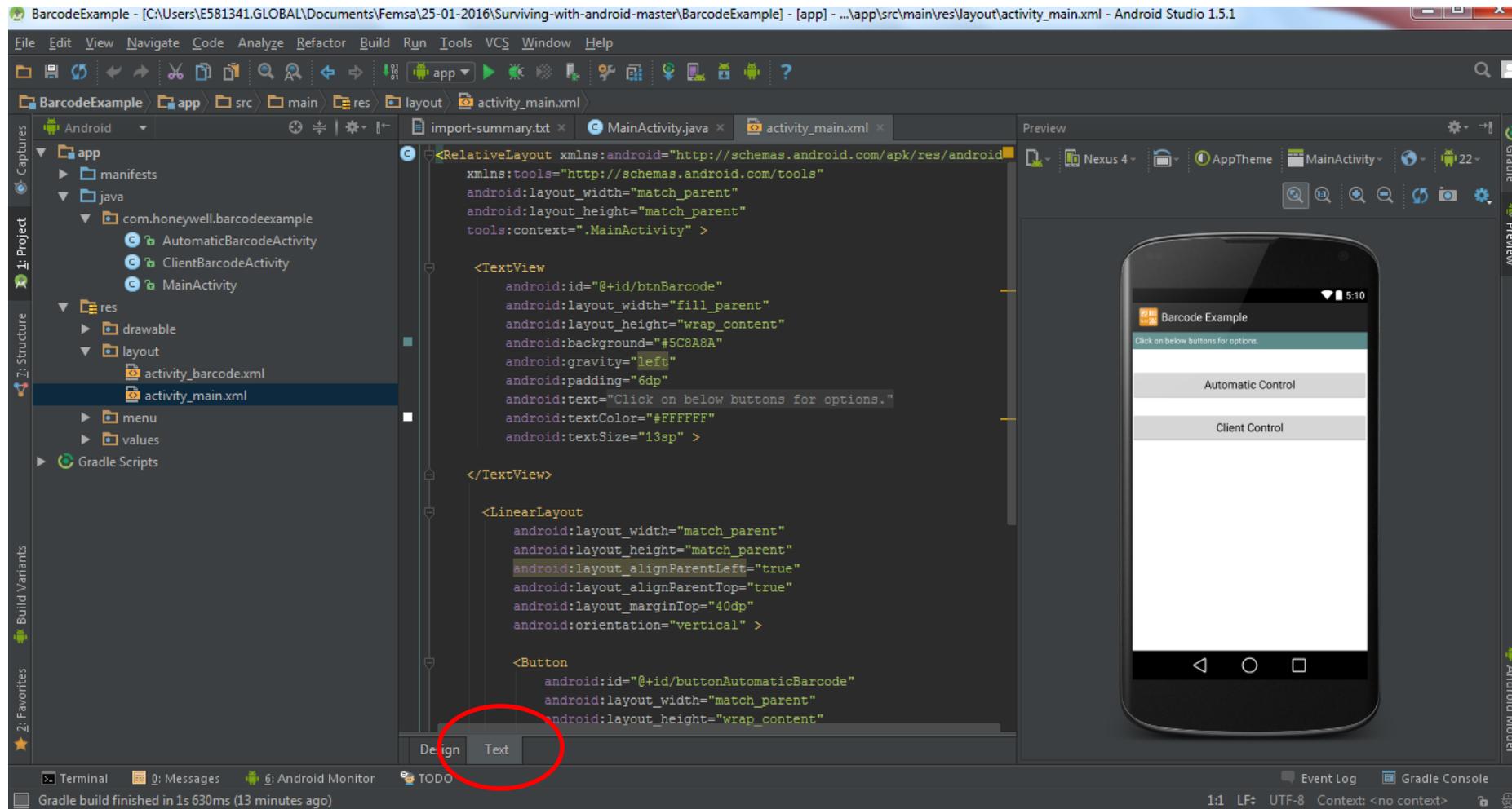
Estrutura de um Projeto Android

Layout/Design: Arquivos de extensão .xml que representam o Activity



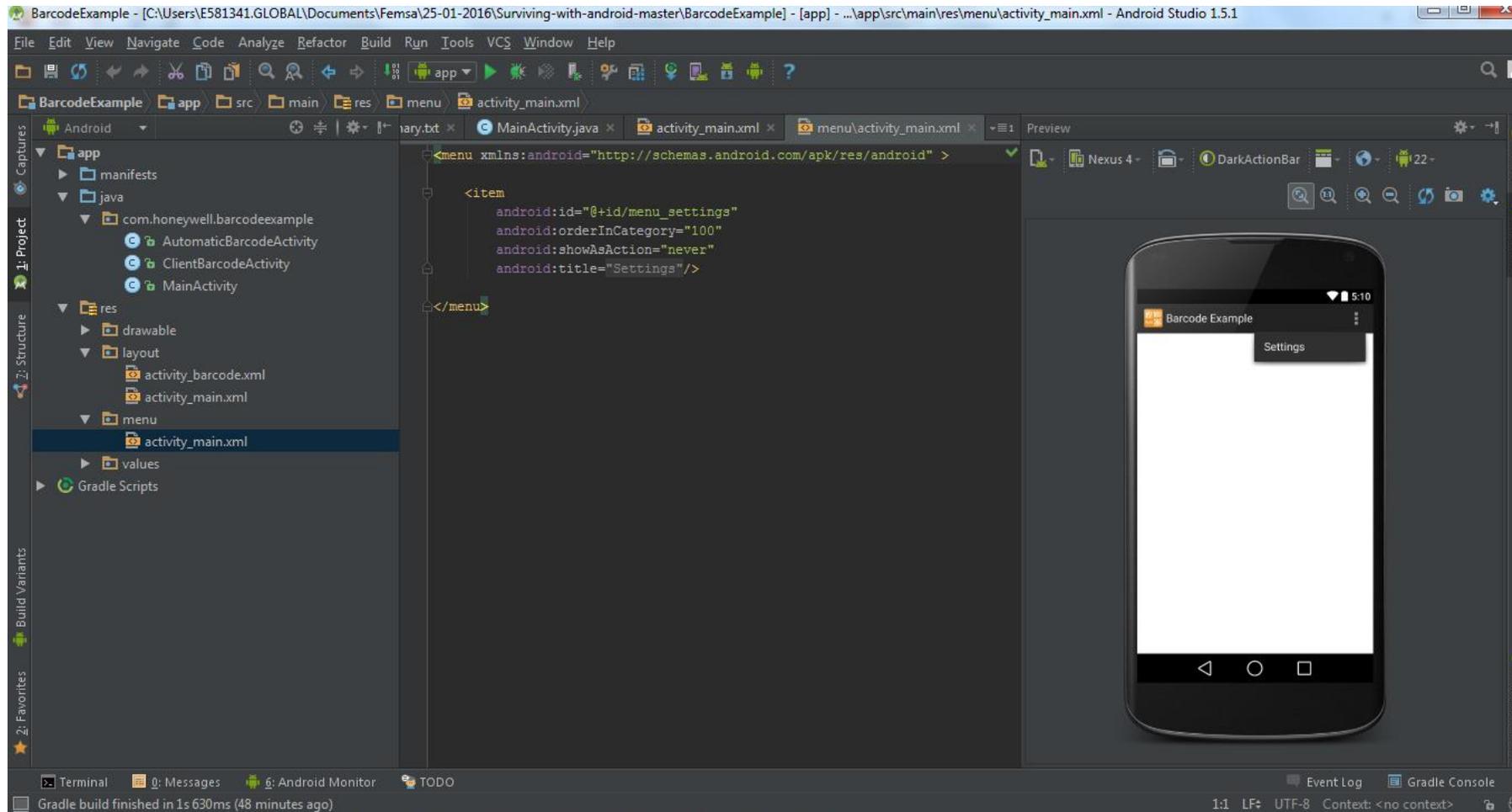
Estrutura de um Projeto Android

Layout/Text: Arquivos de extensão .xml que representam o Activity



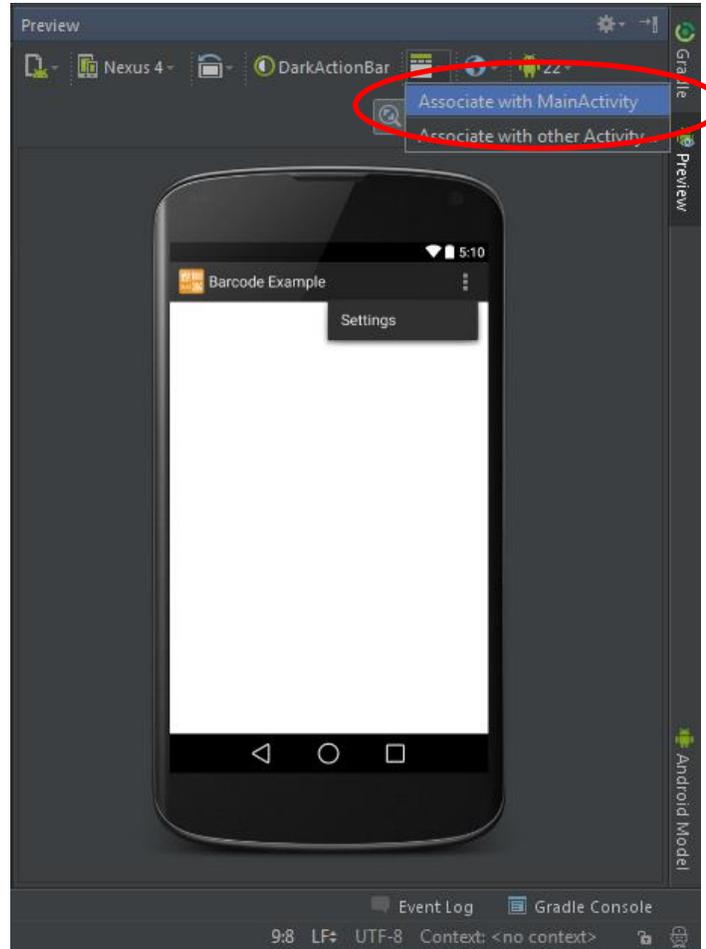
Estrutura de um Projeto Android

Menu: Arquivos de extensão .xml que representam a opção de Menu do Activity



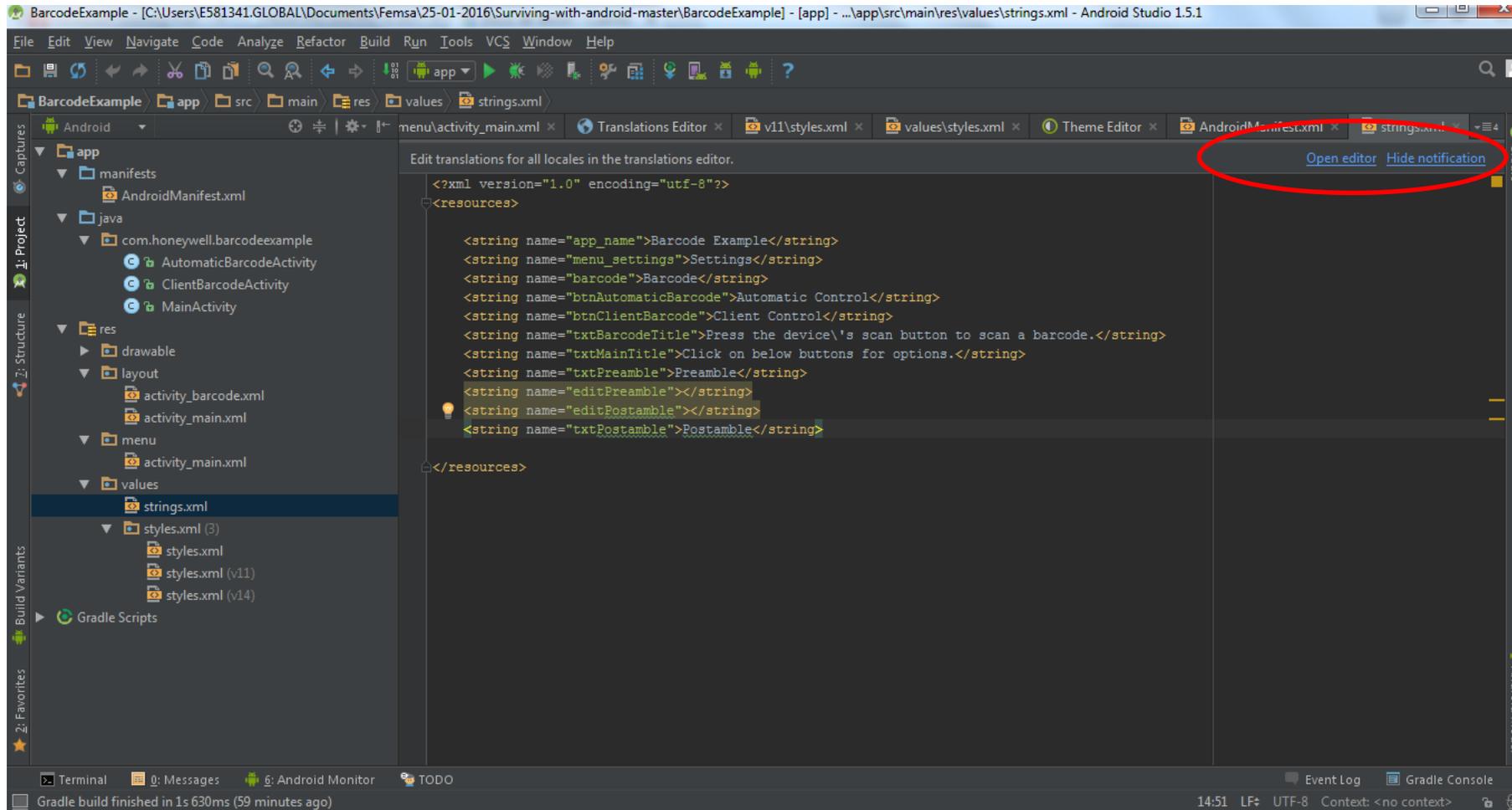
Estrutura de um Projeto Android

Menu: Arquivos de extensão .xml que representam a opção de Menu do Activity



Estrutura de um Projeto Android

Values: Arquivos de extensão .xml que representam os strings usados no Projeto



Estrutura de um Projeto Android

Values: Arquivos de extensão .xml que representam os strings usados no Projeto

The screenshot shows the Android Studio interface with the Translations Editor open. The editor displays a table of string resources for the project 'BarcodeExample'. The table has three columns: 'Key', 'Default Value', and 'Untranslatable'. The key 'btnClientBarcode' is selected, and its default value 'Client Control' is visible in the editor below the table.

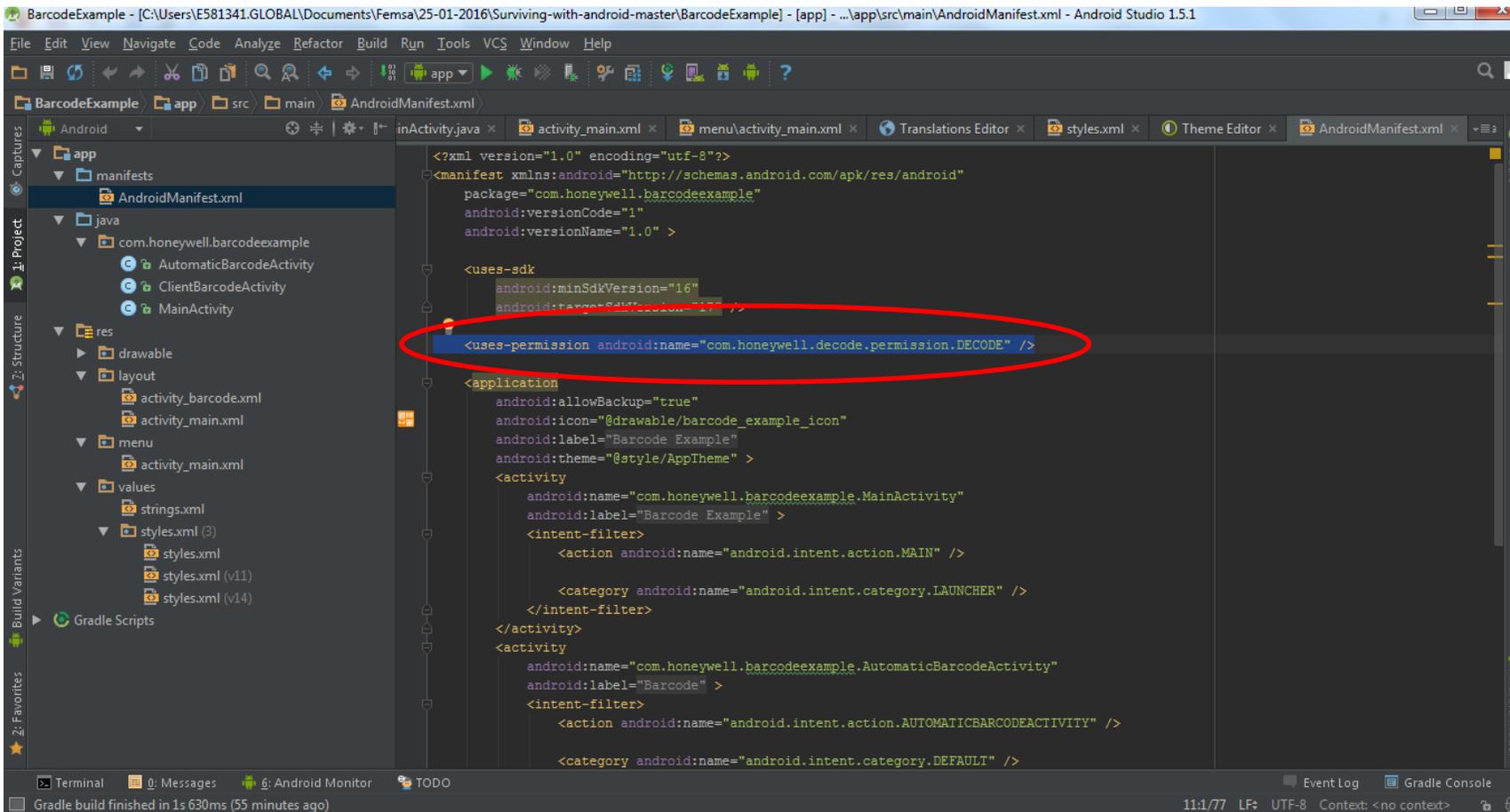
Key	Default Value	Untranslatable
app_name	Barcode Example	<input type="checkbox"/>
barcode	Barcode	<input type="checkbox"/>
btnAutomaticBarcod	Automatic Control	<input type="checkbox"/>
btnClientBarcode	Client Control	<input type="checkbox"/>
editPostamble		<input type="checkbox"/>
editPreamble		<input type="checkbox"/>
menu_settings	Settings	<input type="checkbox"/>
btBarcodeTitle	Press the device's scan button to scan a barcode.	<input type="checkbox"/>
btMainTitle	Click on below buttons for options.	<input type="checkbox"/>
btPostamble	Postamble	<input type="checkbox"/>
btPreamble	Preamble	<input type="checkbox"/>

Below the table, the editor shows the details for the selected key 'btnClientBarcode':

- Key: btnClientBarcode
- Default Value: Client Control
- Translation: (empty)

Estrutura de um Projeto Android

Manifest: Arquivos de extensão .xml que representam a estrutura, API Level e permissões do projeto



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

    <uses-sdk
        android:minSdkVersion="16"
        android:targetSdkVersion="17" />

    <uses-permission android:name="com.honeywell.decode.permission.DECODE" />

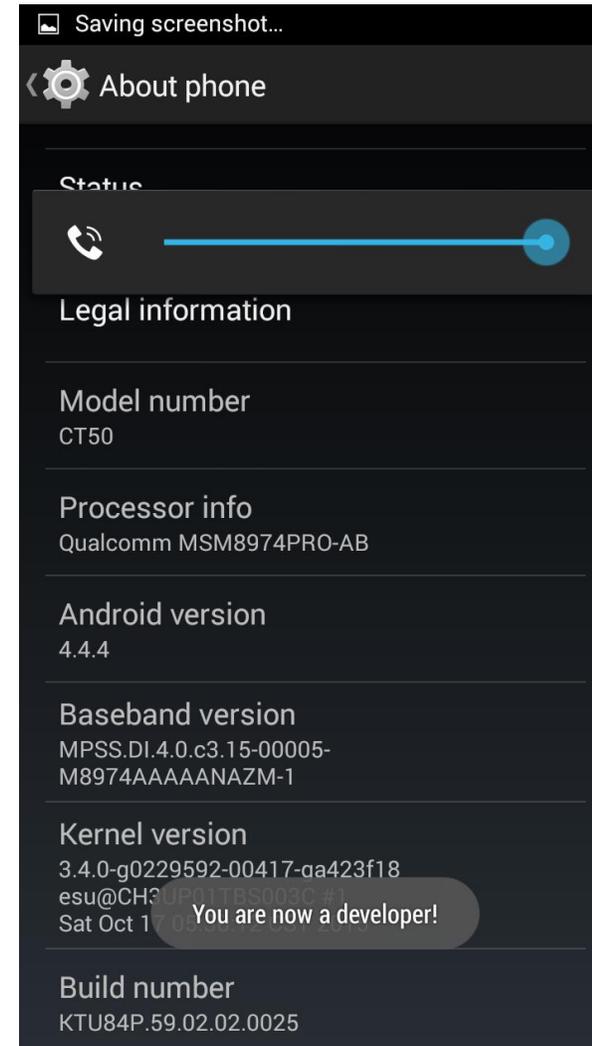
    <application
        android:allowBackup="true"
        android:icon="@drawable/barcode_example_icon"
        android:label="Barcode Example"
        android:theme="@style/AppTheme" >
        <activity
            android:name="com.honeywell.barcodeexample.MainActivity"
            android:label="Barcode Example" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name="com.honeywell.barcodeexampleAutomaticBarcodeActivity"
            android:label="Barcode" >
            <intent-filter>
                <action android:name="android.intent.action.AUTOMATICBARCODEACTIVITY" />

                <category android:name="android.intent.category.DEFAULT" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

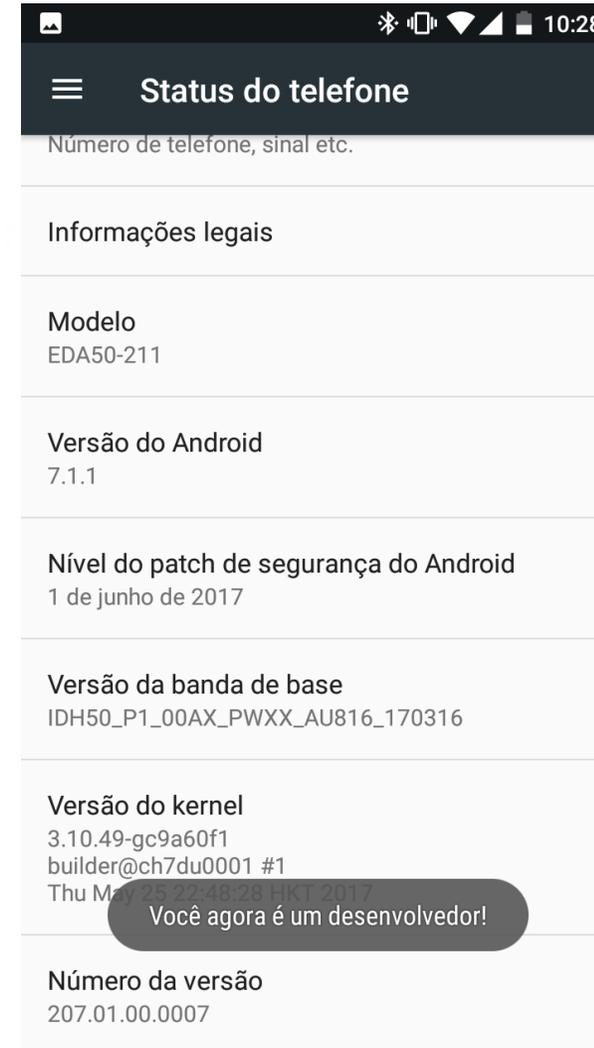
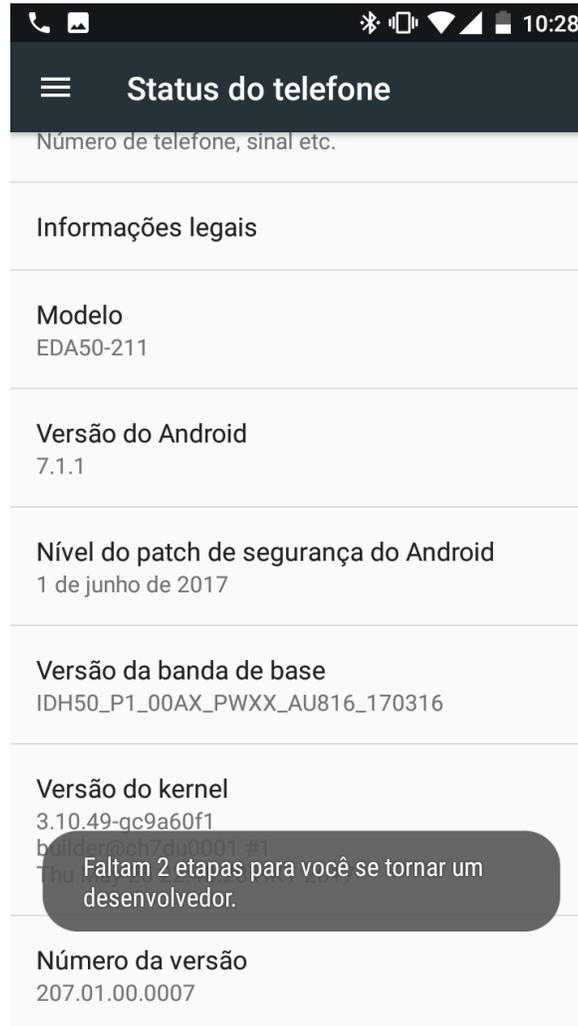
Configurando o Dolphin CT50 ou 75E para desenvolvimento

Settings>About Phone>Build number pressionar este item por 4 vezes até aparecer a mensagem "You are now a developer".



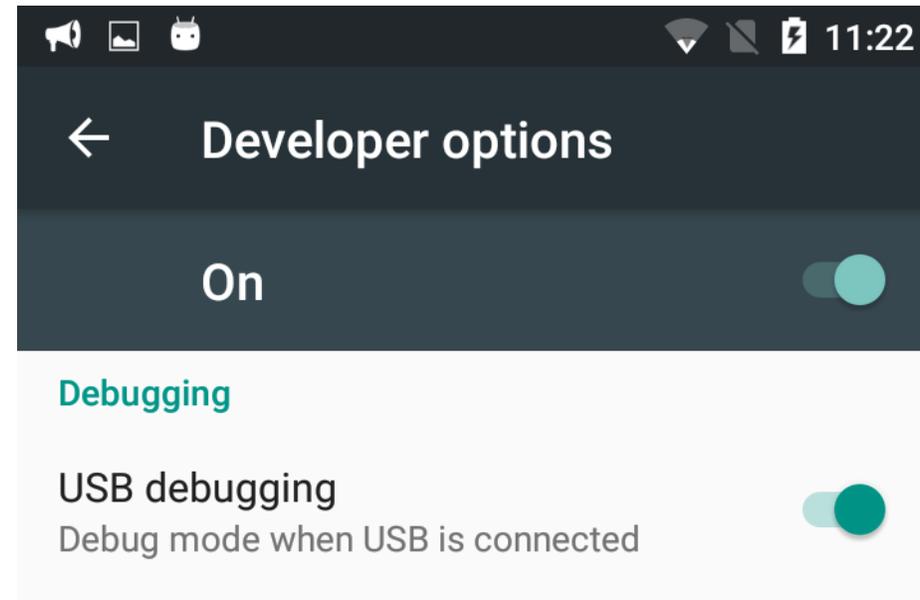
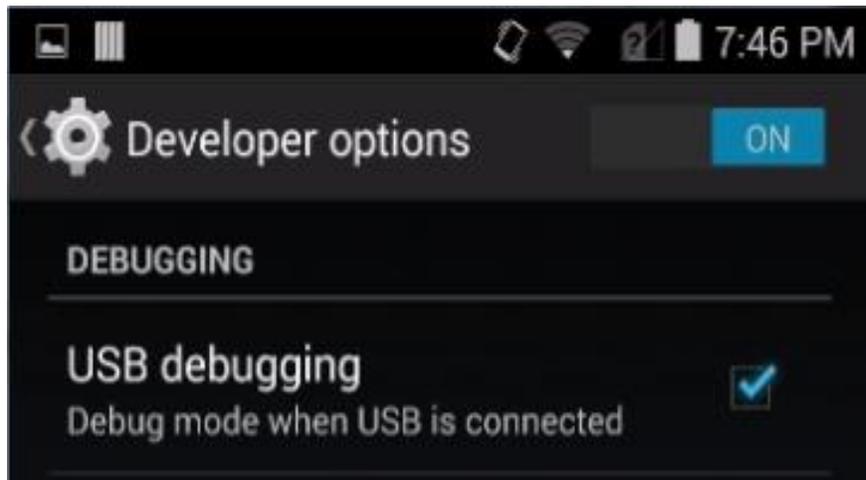
Configurando o ScanPal EDA 50 para desenvolvimento

Configurar>Sobre o dispositivo>Número da versão pressionar este item por 4 vezes até aparecer a mensagem "You are now a developer".



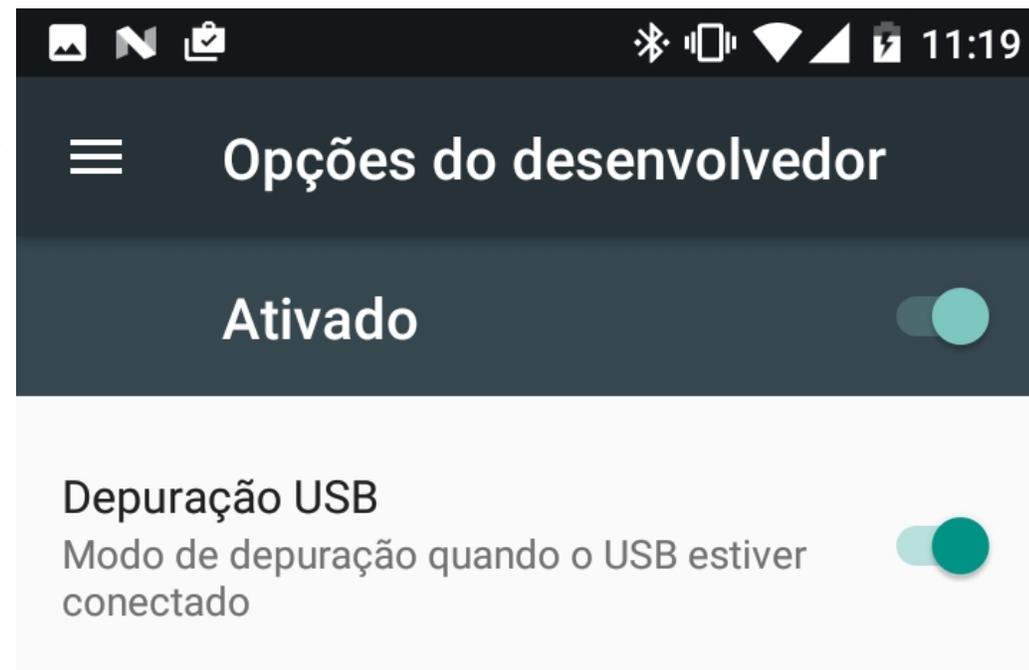
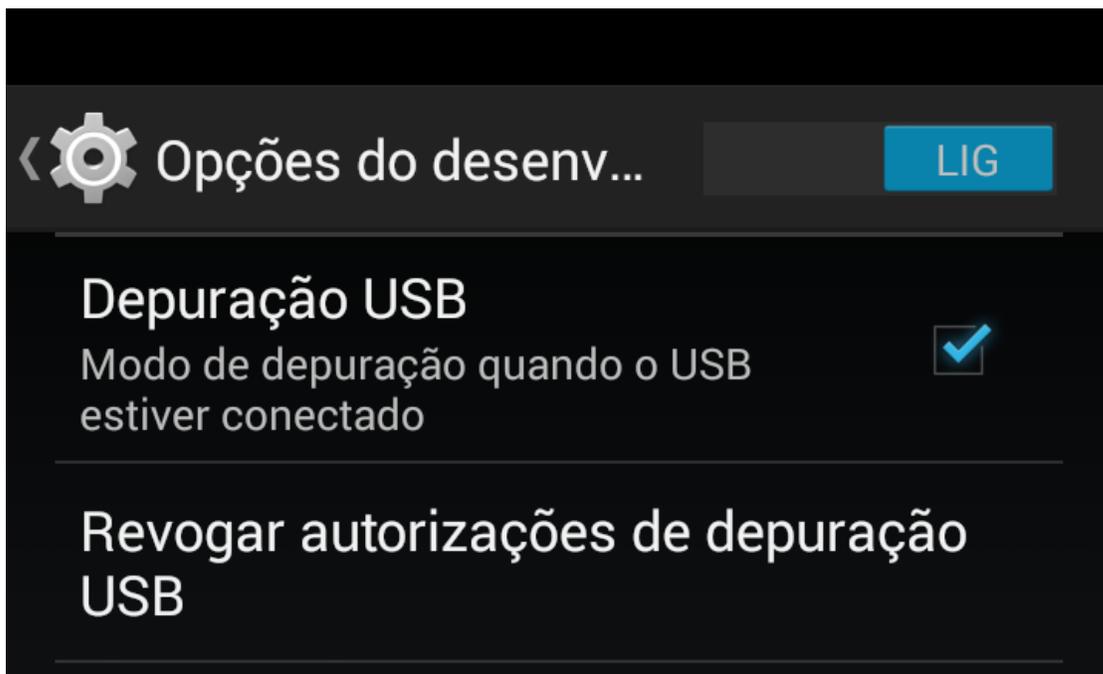
Configurando o Dolphin CT50 ou 75E para desenvolvimento

Settings>Developer Options, selecionar o check box que esta do lado do USB Debugging



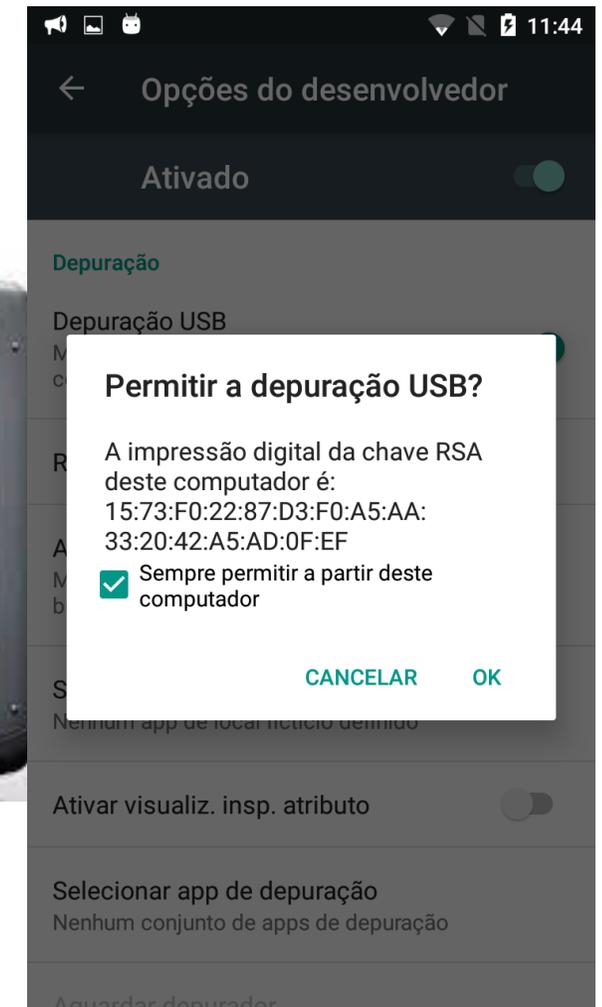
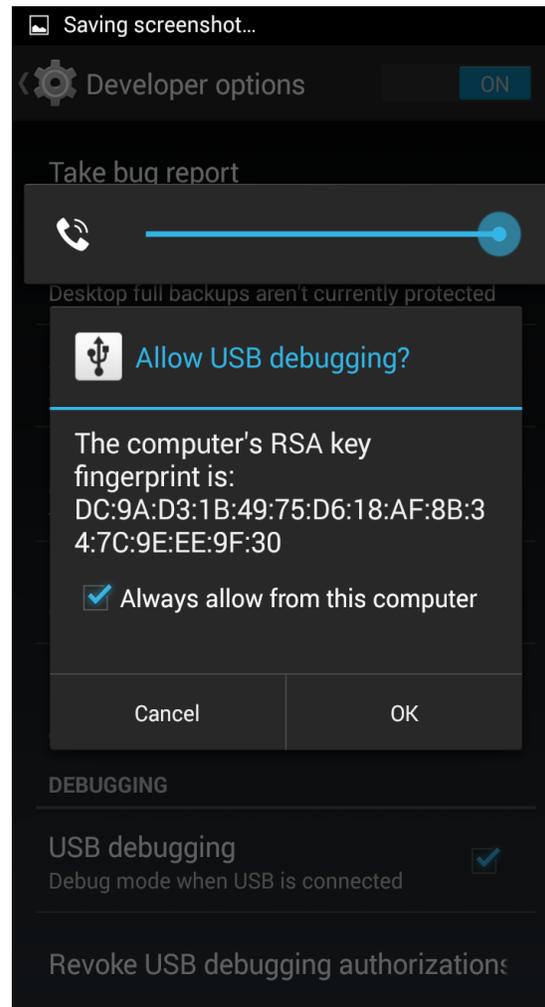
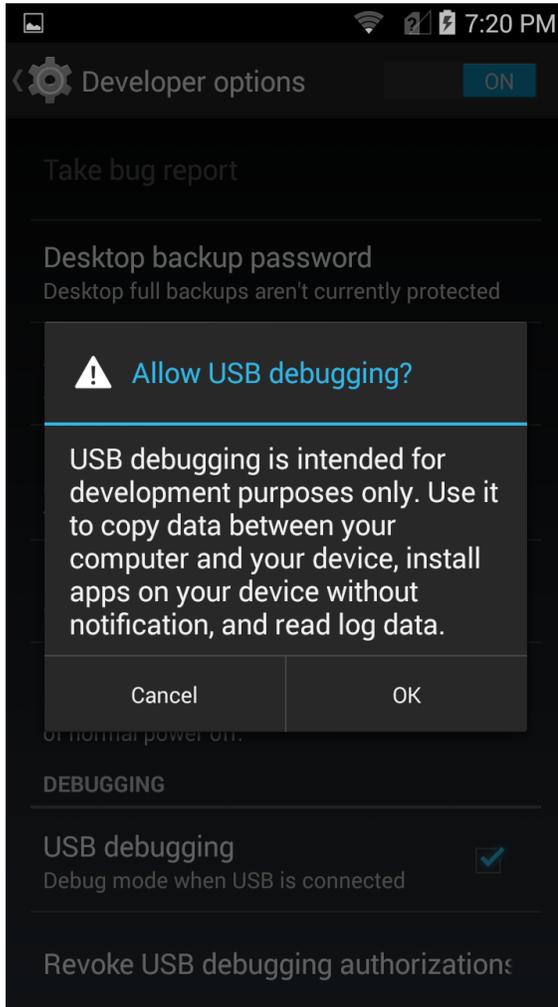
Configurando o EDA50 para desenvolvimento

Settings>Developer Options, selecionar o check box que esta do lado do USB Debugging



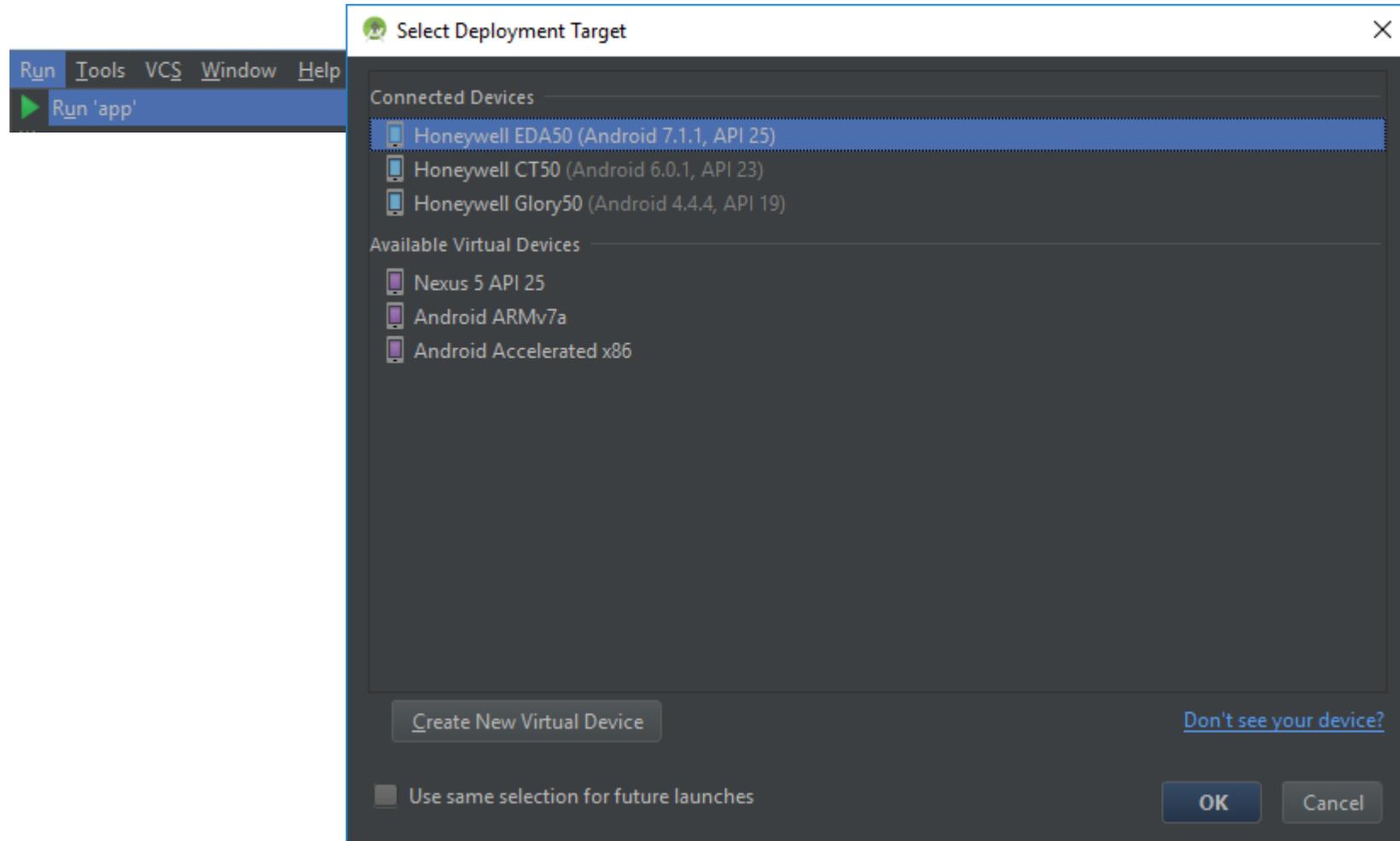
Configurando o Dolphin CT50 ou 75E para desenvolvimento

Allow USB Debugging? OK e selecionar “Always allow from this computer”



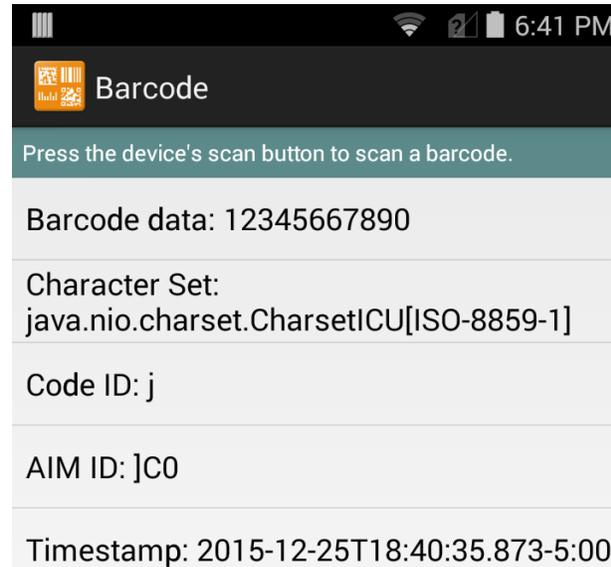
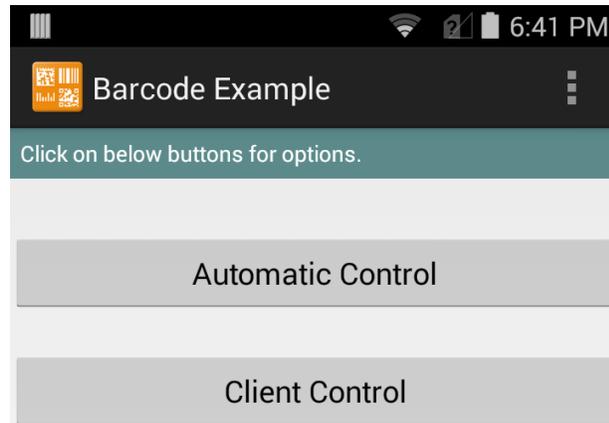
Executando o exemplo no nosso dispositivo Honeywell Android

Uma vez associado o dispositivo com o PC, poderemos executar e/ou depurar o aplicativo compilado no nosso dispositivo Dolphin CT50. Pressionando o botão Play ou no menú de opções Run>Run 'app':



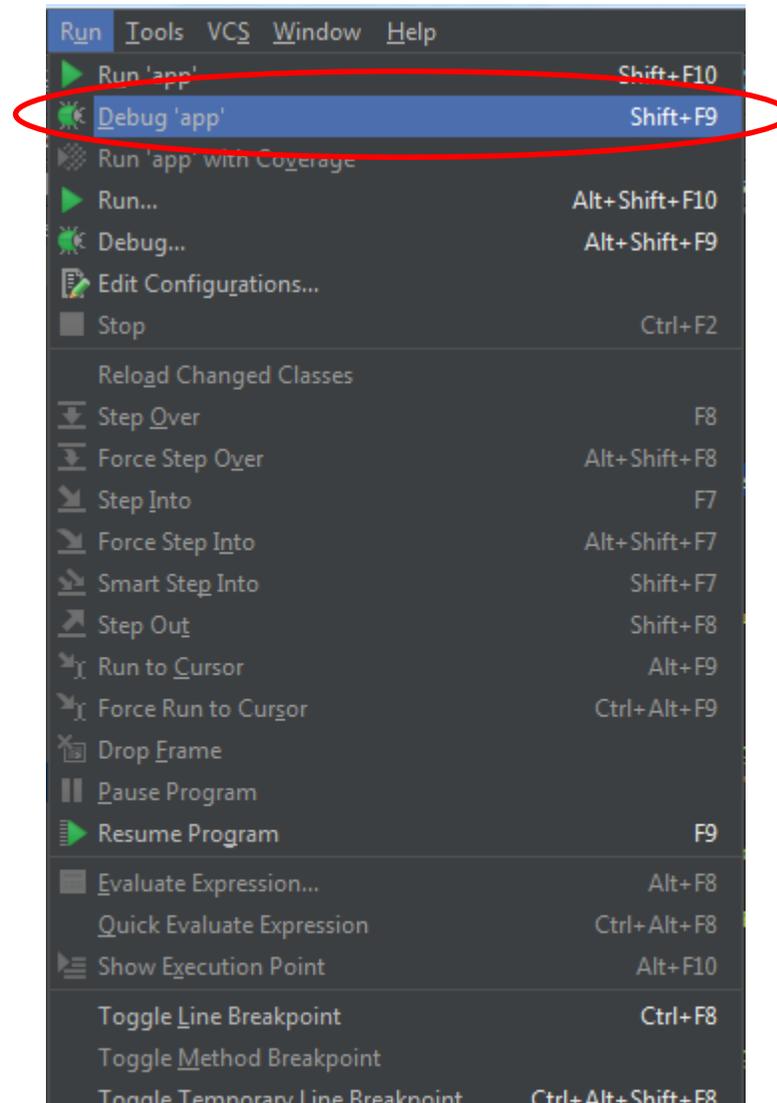
Executando o exemplo no nosso dispositivo Honeywell Android

E no dispositivo aparecerá a interface gráfica do nosso aplicativo BarcodeExample.



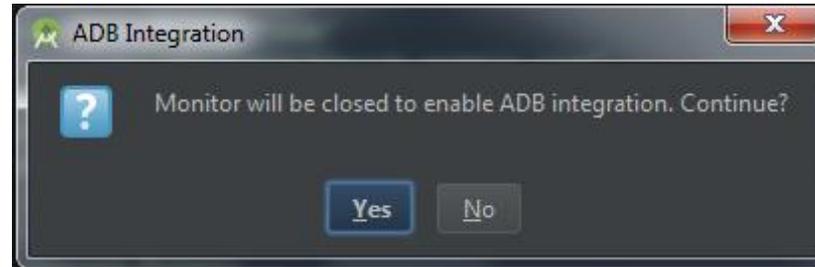
Depurando o exemplo no nosso dispositivo Honeywell Android

Para depurar, selecionamos a opção do Menú Run>Debug 'app'

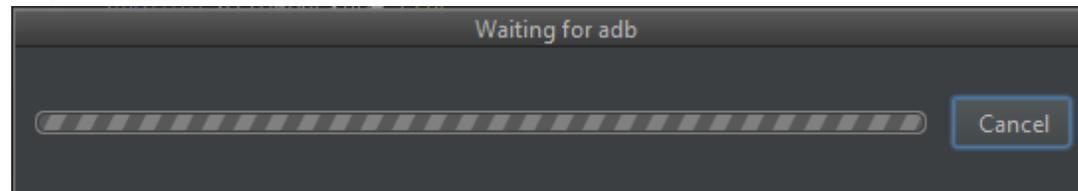


Depurando o exemplo no nosso dispositivo Honeywell Android

Aparecerá a caixa de Mensagens solicitando a confirmação para conectar-se ao dispositivo via ADB (Android Debug Bridge)

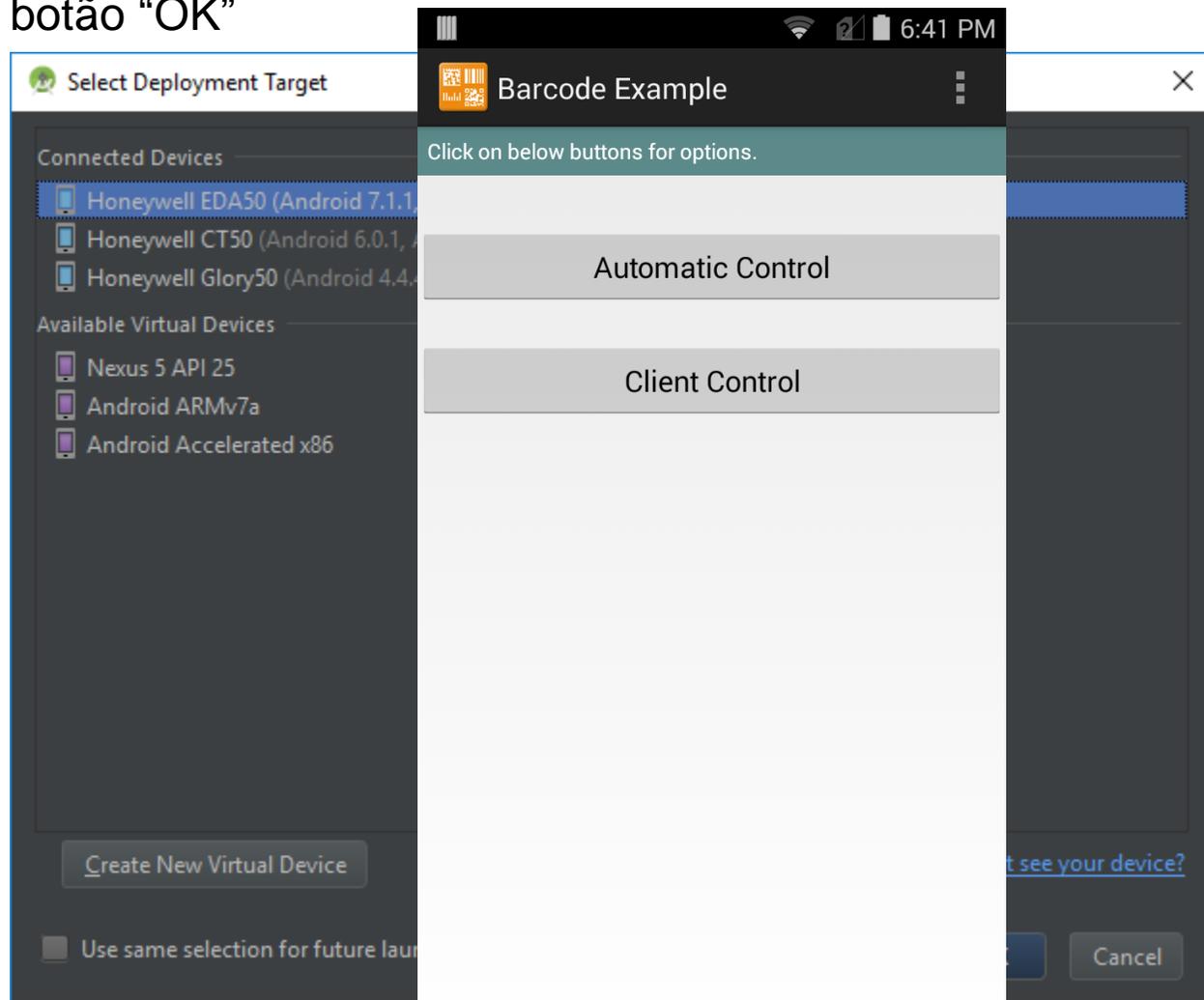


Pressionamos o botão "Yes", observaremos que aparecerá uma janela executando a tentativa de conexão

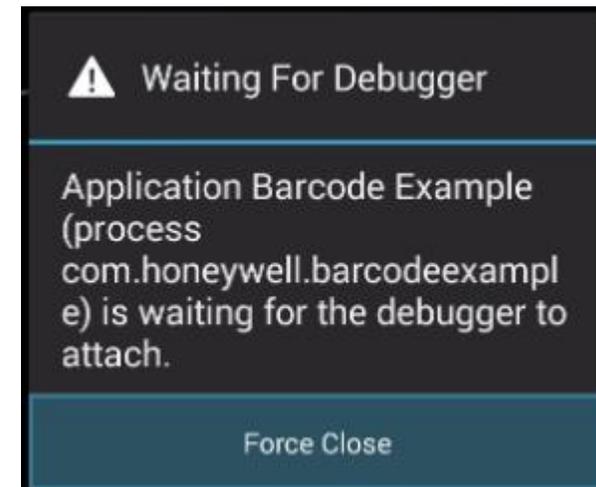


Depurando o exemplo no nosso dispositivo Honeywell Android

Aparecerá novamente a confirmação do nosso dispositivo Honeywell Android, confirmamos pressionando o botão “OK”

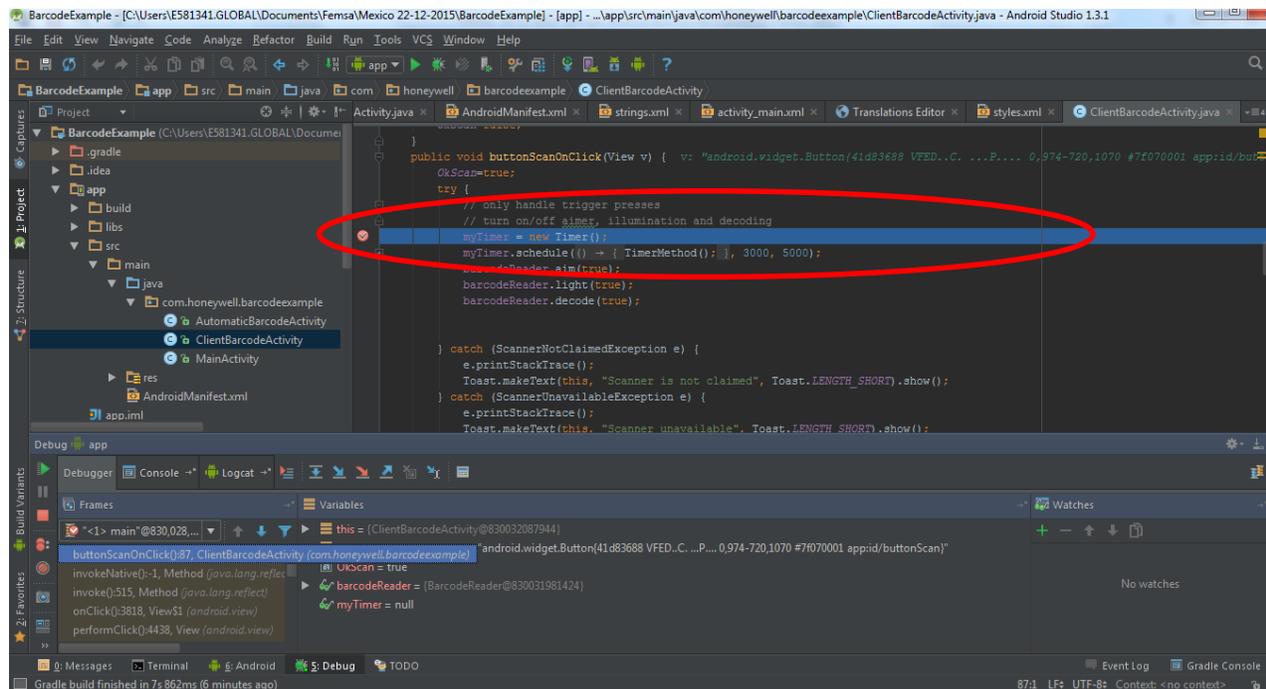


No dispositivo aparecerá uma caixa de Mensagens confirmando a execução da depuração

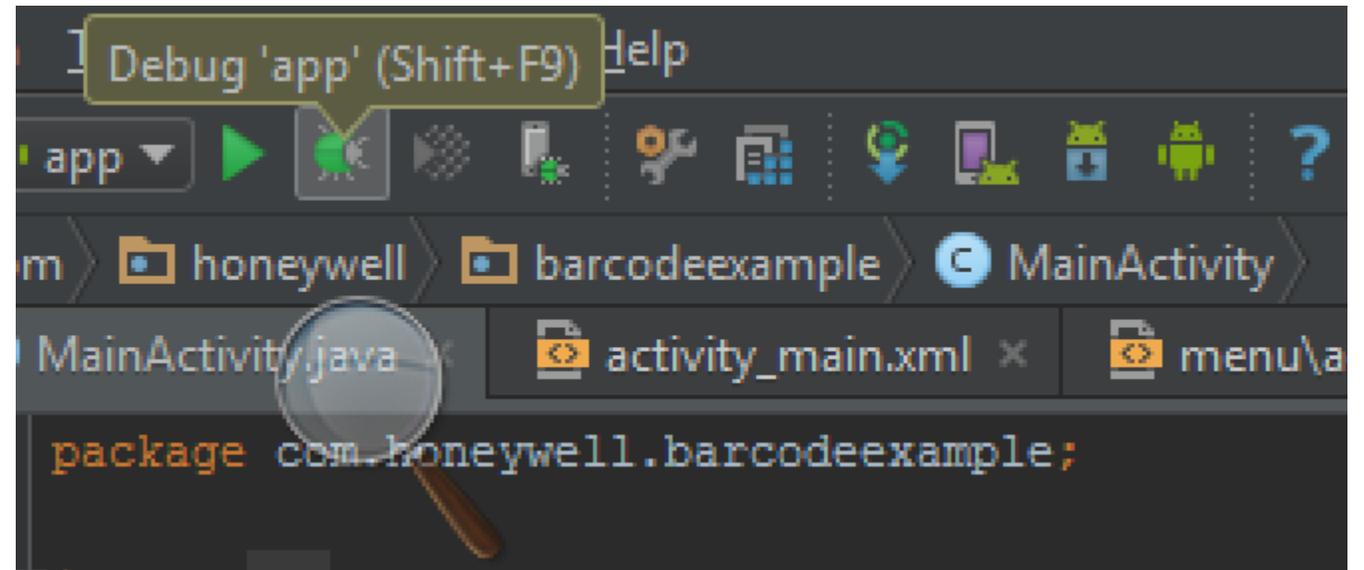
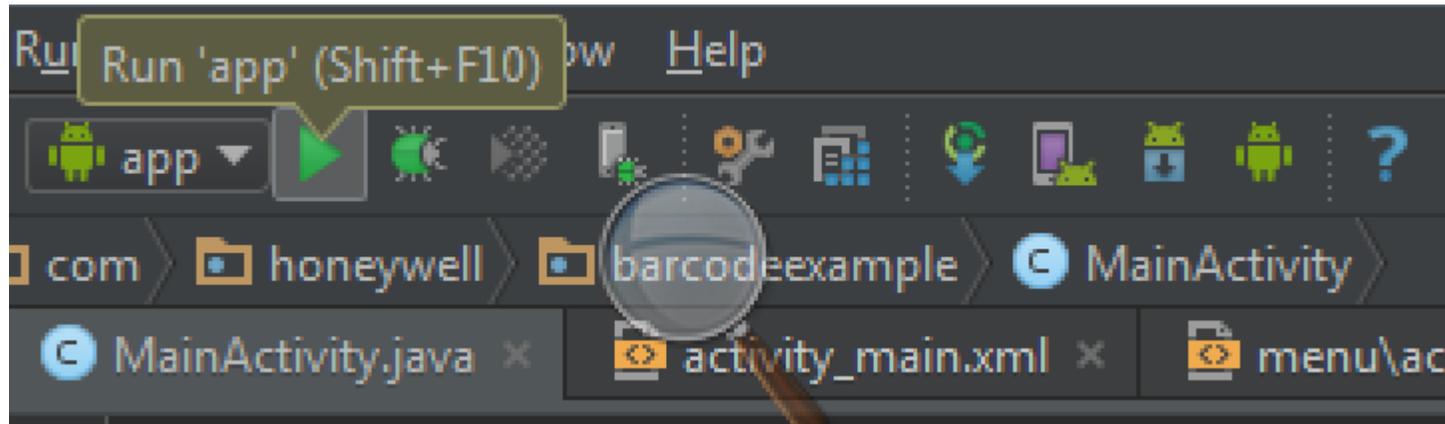


Depurando o exemplo no nosso dispositivo Honeywell Android

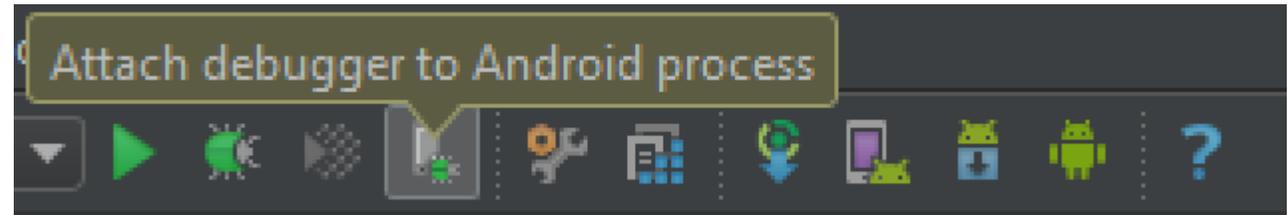
Podemos colocar um ponto de ruptura ou Break Point, pressionando o lado esquerdo da interface (se mostrará um ponto de cor vermelho claro) e uma vez pare podemos ir passo a passo pressionando a tecla de função F8 ou continuar, pressionando a tecla de função F9



Ferramentas do IDE



Ferramentas do IDE



BarcodeExample - [C:\Users\E581341.GLOBAL\Documents\Femsa\25-01-2016\Surviving-with-android-master\BarcodeExample] - [app] - ...\app\src\main\java\com\honeywell\barcodeexample\MainActivity.java - Android Studio 1.5.1

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

BarcodeExample app src main java com honeywell barcodeexample MainActivity

Android

- app
 - manifests
 - AndroidManifest.xml
 - java
 - com.honeywell.barcodeexample
 - AutomaticBarcodeActivity
 - ClientBarcodeActivity
 - MainActivity
 - res
 - drawable
 - layout

MainActivity.java

```
package com.honeywell.barcodeexample;

import ...

public class MainActivity extends Activity {

    private static BarcodeReader barcodeReader;
    private AidoManager manager;

    private Button btnAutomaticBarcode;
    private Button btnClientBarcode;

    @Override
```

Debug: Android Debugger (8600) Android Debugger (8601)

Debugger Console Dump #2 Dump #3

Build Variants

- <1> main@830031654488 (runnable)
- <7> FinalizerDaemon@830034634792 (waiting)
- <8> FinalizerWatchdogDaemon@830034635224 (waiting)
- <6> ReferenceQueueDaemon@830034634432 (waiting)
- <10> Binder_2@830034648920 (runnable)
- <9> Binder_1@830034648064 (runnable)
- <5> Crmpiler@830034634192 (waiting)
- <3> Signal Catcher@830034633712 (waiting)
- <2> GC@830034633488 (waiting)

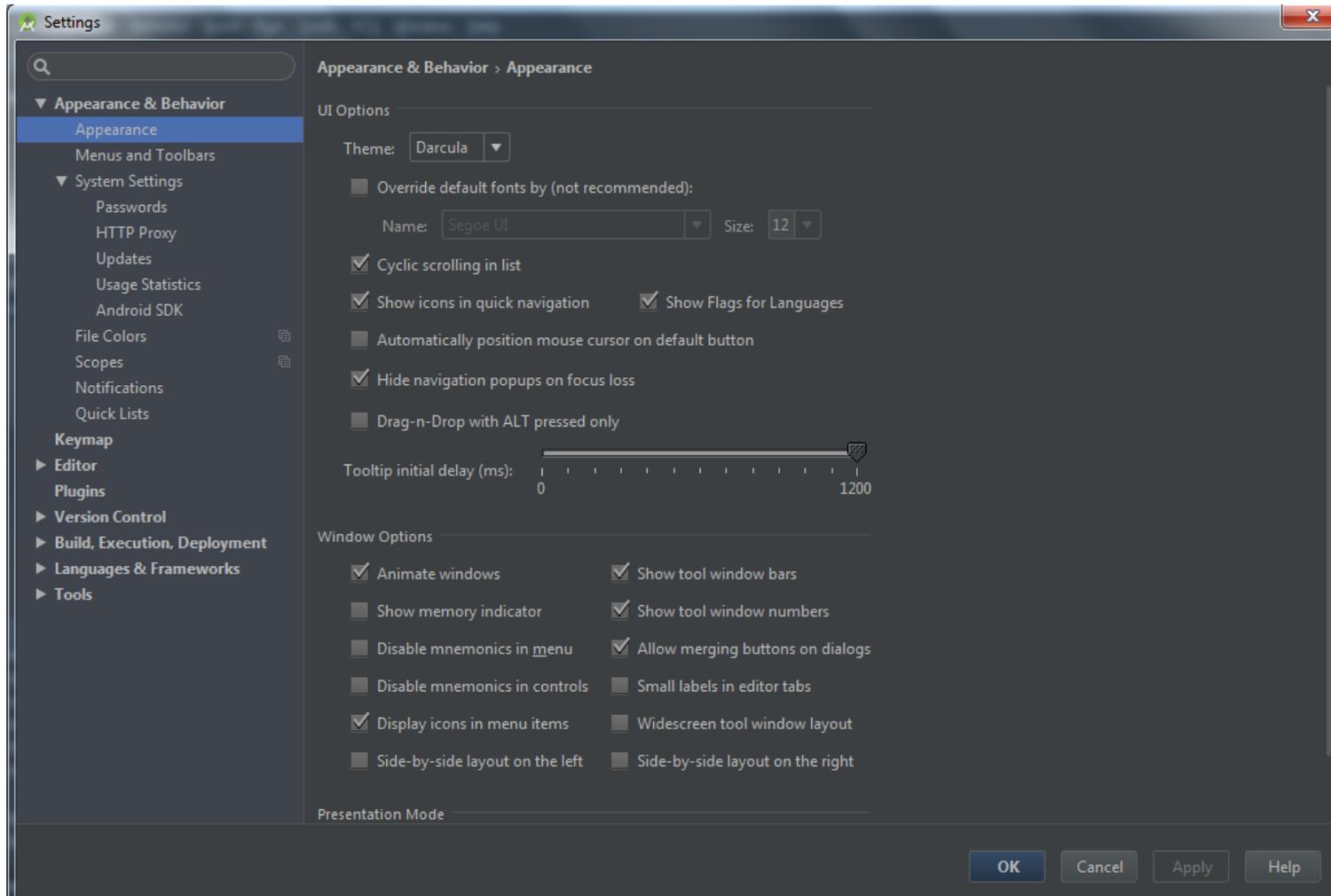
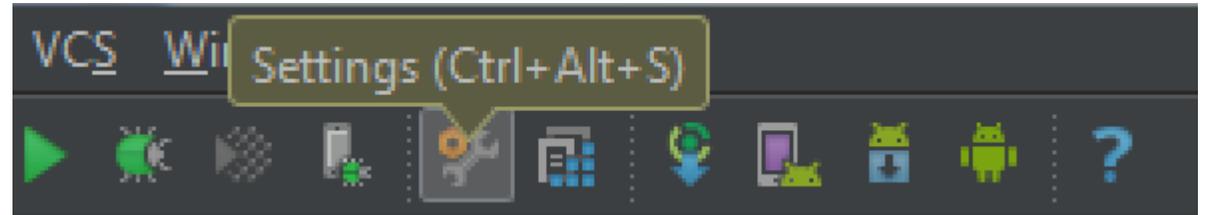
Get thread dump

```
<6> ReferenceQueueDaemon@830034634432" daemon prio=5 waiting
java.lang.Thread.State: WAITING
    at java.lang.Object.wait(Object.java:-1)
    at java.lang.Object.wait(Object.java:364)
    at java.lang.Daemons$ReferenceQueueDaemon.run(Daemons.java:130)
    at java.lang.Thread.run(Thread.java:841)
```

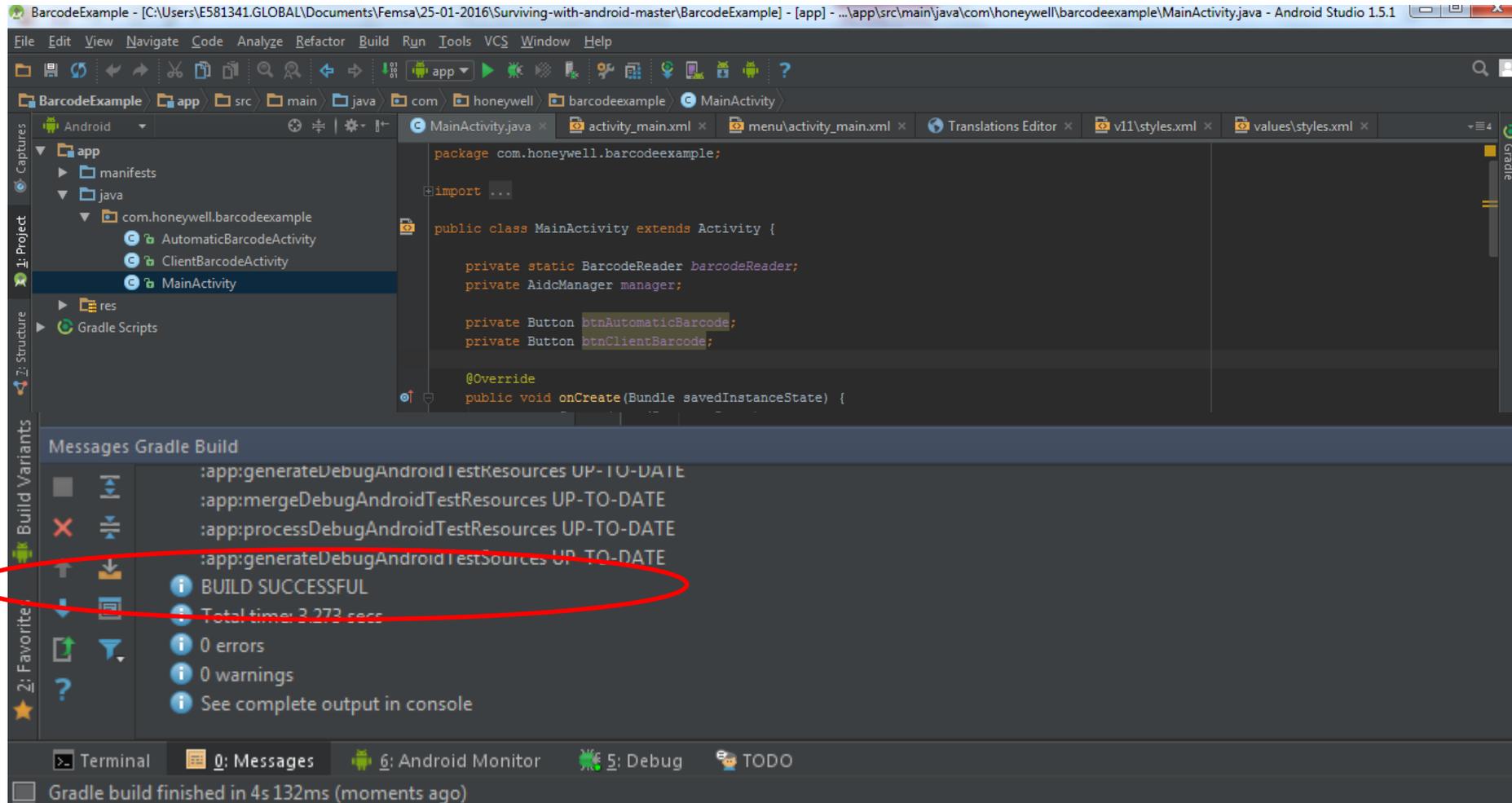
Terminal 0: Messages 6: Android Monitor 5: Debug TODO

Gradle build finished in 1s 630ms (today 7:11 AM)

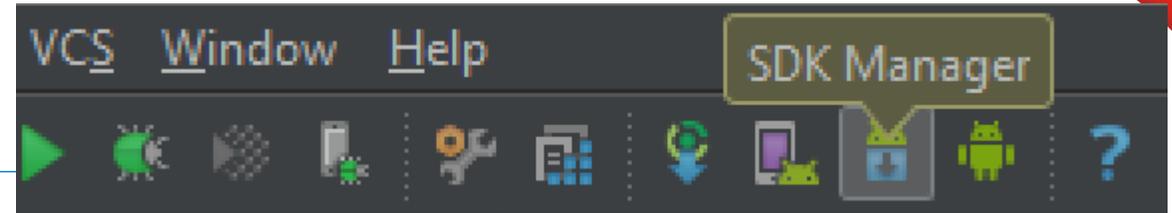
Ferramentas do IDE



Ferramentas do IDE



Ferramentas do IDE



Default Settings

Appearance & Behavior > System Settings > Android SDK

Manager for the Android SDK and Tools used by Android Studio

Android SDK Location: [Edit](#)

SDK Platforms | SDK Tools | SDK Update Sites

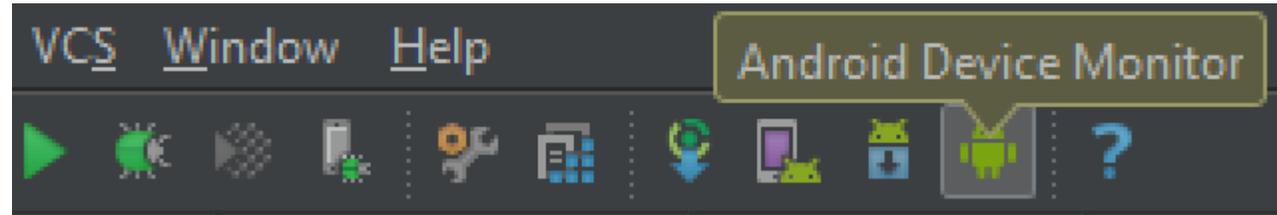
Each Android SDK Platform package includes the Android platform and sources pertaining to an API level by default. Once installed, Android Studio will automatically check for updates. Check "show package details" to display individual SDK components.

	Name	API Level	Revision	Status
<input checked="" type="checkbox"/>	Android 7.1.1 (Nougat)	25	3	Installed
<input checked="" type="checkbox"/>	Android 7.0 (Nougat)	24	2	Installed
<input checked="" type="checkbox"/>	Android 6.0 (Marshmallow)	23	3	Installed
<input checked="" type="checkbox"/>	Android 5.1 (Lollipop)	22	2	Installed
<input checked="" type="checkbox"/>	Android 5.0 (Lollipop)	21	2	Installed
<input checked="" type="checkbox"/>	Android 4.4W (KitKat Wear)	20	2	Installed
<input checked="" type="checkbox"/>	Android 4.4 (KitKat)	19	4	Installed
<input type="checkbox"/>	Android 4.3 (Jelly Bean)	18	3	Not installed
<input checked="" type="checkbox"/>	Android 4.2 (Jelly Bean)	17	3	Installed
<input type="checkbox"/>	Android 4.1 (Jelly Bean)	16	5	Partially installed
<input type="checkbox"/>	Android 4.0.3 (IceCreamSandwich)	15	5	Not installed
<input type="checkbox"/>	Android 4.0 (IceCreamSandwich)	14	4	Not installed
<input type="checkbox"/>	Android 3.2 (Honeycomb)	13	1	Not installed
<input type="checkbox"/>	Android 3.1 (Honeycomb)	12	3	Not installed
<input checked="" type="checkbox"/>	Android 3.0 (Honeycomb)	11	2	Installed
<input checked="" type="checkbox"/>	Android 2.3.3 (Gingerbread)	10	2	Installed
<input type="checkbox"/>	Android 2.3 (Gingerbread)	9	2	Not installed
<input checked="" type="checkbox"/>	Android 2.2 (Froyo)	8	3	Installed
<input type="checkbox"/>	Android 2.1 (Eclair)	7	3	Not installed

Show Package Details

OK Cancel Apply Help

Ferramentas do IDE

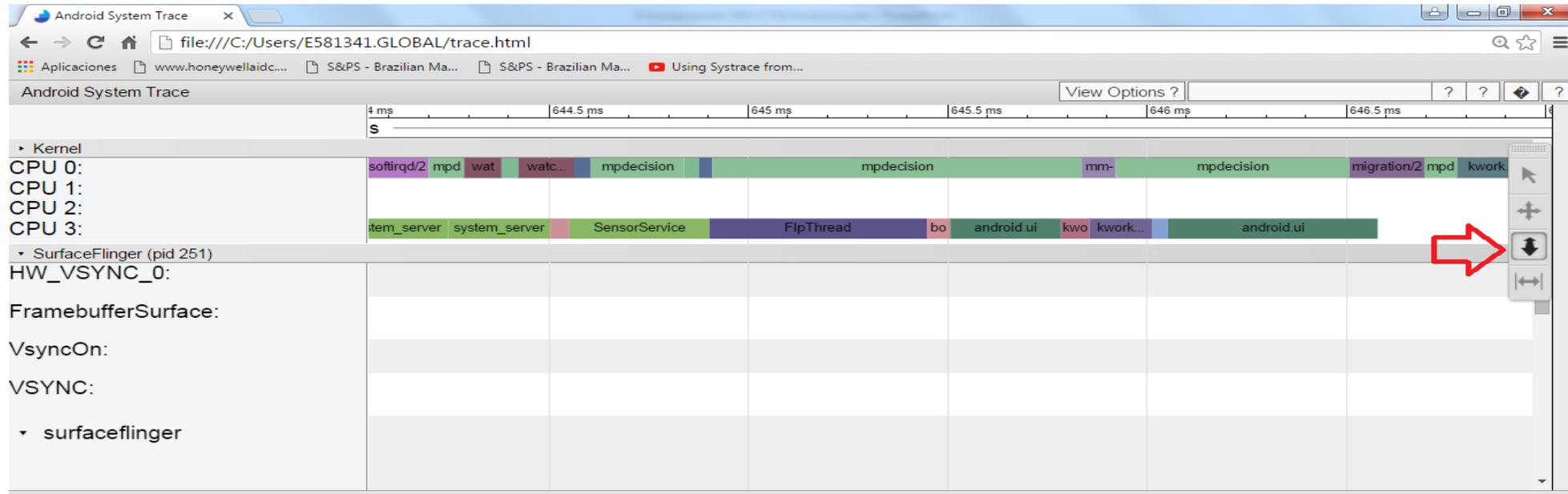
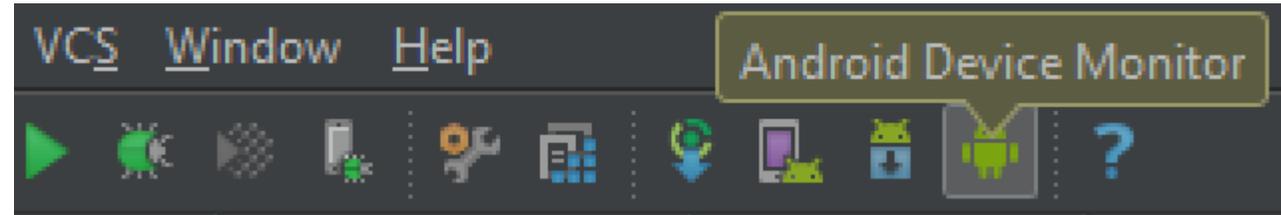


The screenshot displays the Android Studio IDE with the Android Device Monitor (ADM) window open. The ADM window is divided into several sections:

- Devices:** A table listing virtual devices. The 'honeywell-ct50-1514640019' device is selected, showing it is online and running Android 4.4.4. The application 'com.honeywell.barcodeexample' is running on it.
- UI Preview:** A visual representation of the application's user interface, showing a 'Barcode' screen with a 'Scan' button. A red dashed box highlights a 'ListView' component.
- View Hierarchy:** A tree view showing the UI structure. The selected 'ListView' is highlighted in blue.
- Node Detail:** A table showing properties for the selected 'ListView' node, such as 'resource-id', 'class', 'package', 'content-d...', 'checkable', and 'checked'.
- Process List:** A table showing the running processes of the application, including their IDs, TIDs, statuses, and names.
- LogCat:** A window at the bottom showing system logs, including DDMS messages: '[2016-01-22 09:56:07 - ddm-heap] *** Received REAL', '[2016-01-22 09:56:20 - ddm-heap] *** Received REAL', and '[2016-01-22 10:02:01 - ddm-heap] *** Received REAL'.

The status bar at the bottom right indicates '142M of 491M' of memory used.

Ferramentas do IDE



Exemplos SDK Android: Entendendo o BarcodeExample

```
package com.honeywell.barcodeexample;

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

import com.honeywell.aidc.AidcManager;
import com.honeywell.aidc.AidcManager.CreatedCallback;
import com.honeywell.aidc.BarcodeReader;
import com.honeywell.aidc.ScannerNotClaimedException;
import com.honeywell.aidc.ScannerUnavailableException;

public class MainActivity extends Activity {

    private static BarcodeReader barcodeReader;
    private AidcManager manager;

    private Button btnAutomaticBarcode;
    private Button btnClientBarcode;

    private final static String TAG = "ScanResultReceiver";
    private static boolean Ok;

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

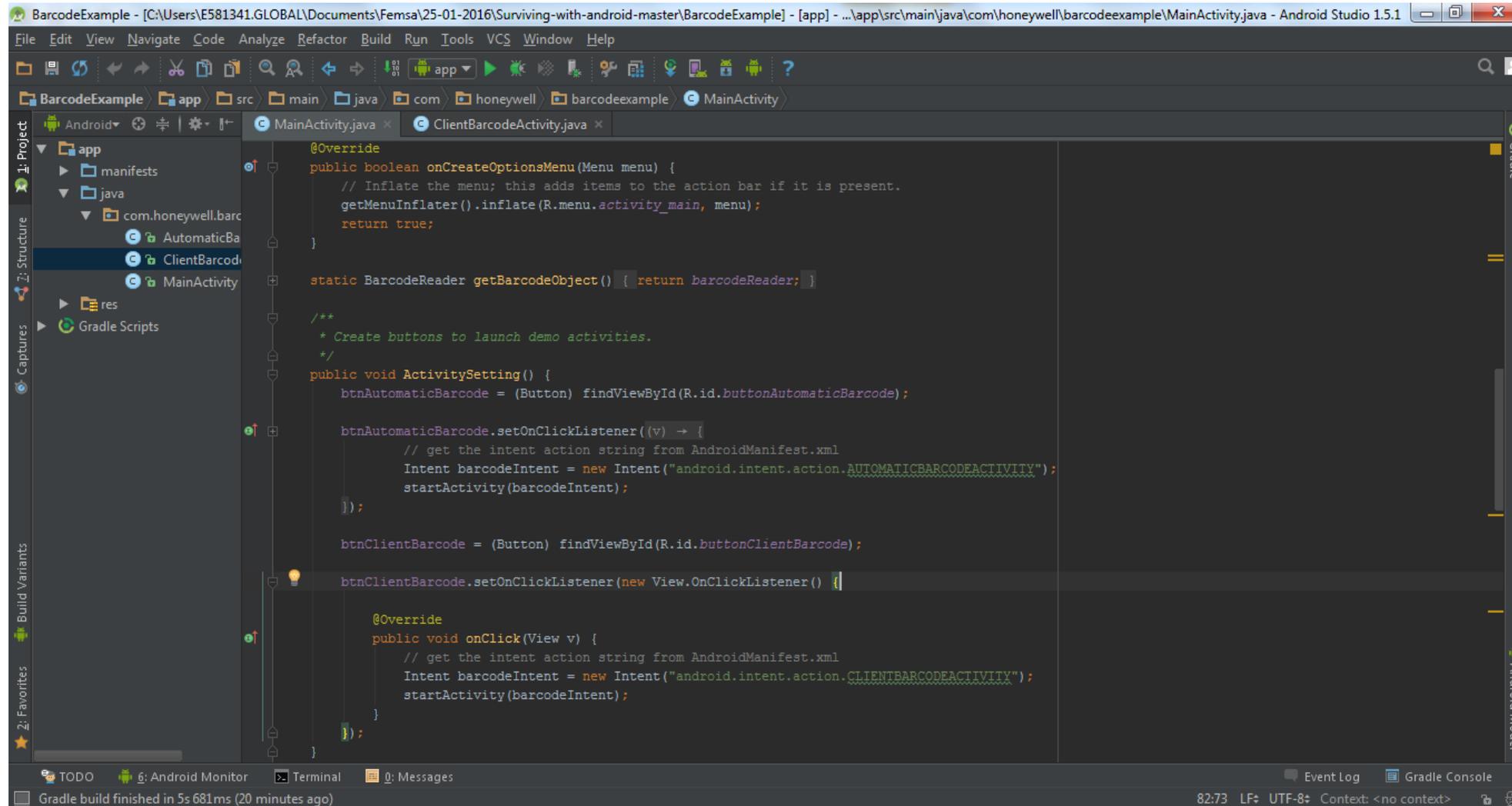
Imports Android

Imports Honeywell

Declaração Variáveis

Ponto de entrada do App

Exemplos SDK Android: Entendendo o BarcodeExample



```
BarcodeExample - [C:\Users\E581341.GLOBAL\Documents\Femsa\25-01-2016\Surviving-with-android-master\BarcodeExample] - [app] - ...\app\src\main\java\com\honeywell\barcodeexample\MainActivity.java - Android Studio 1.5.1
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
BarcodeExample app src main java com honeywell barcodeexample MainActivity
Project
  app
    manifests
    java
      com.honeywell.barcodeexample
        AutomaticBarcodeActivity.java
        ClientBarcodeActivity.java
        MainActivity.java
    res
    Gradle Scripts
Structure
Captures
Build Variants
Favorites
Android Monitor
Terminal
Messages
Event Log
Gradle Console
Gradle
Android Model

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.activity_main, menu);
    return true;
}

static BarcodeReader getBarcodeObject() { return barcodeReader; }

/**
 * Create buttons to launch demo activities.
 */
public void ActivitySetting() {
    btnAutomaticBarcode = (Button) findViewById(R.id.buttonAutomaticBarcode);

    btnAutomaticBarcode.setOnClickListener((v) -> {
        // get the intent action string from AndroidManifest.xml
        Intent barcodeIntent = new Intent("android.intent.action.AUTOMATICBARCODEACTIVITY");
        startActivity(barcodeIntent);
    });

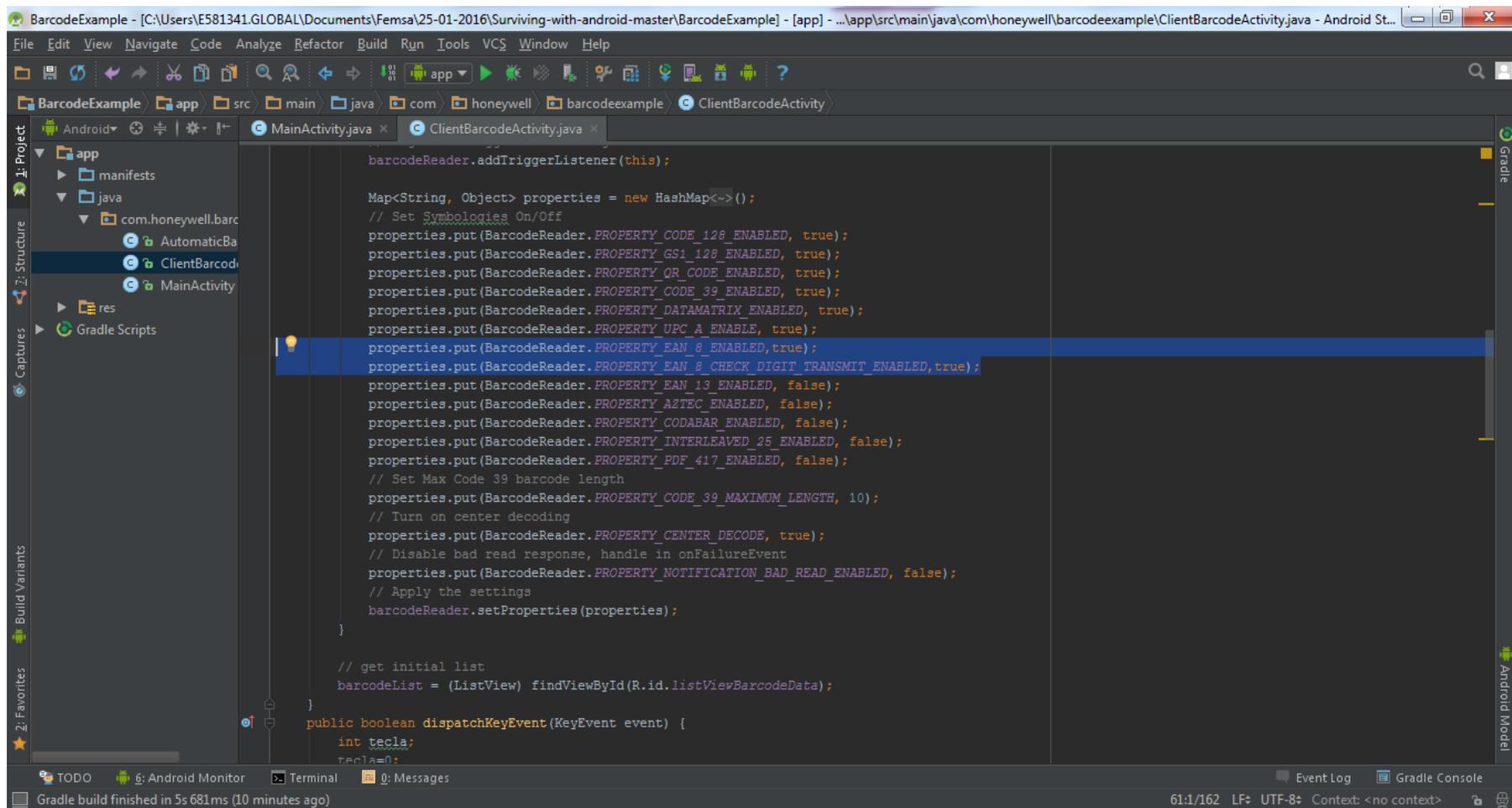
    btnClientBarcode = (Button) findViewById(R.id.buttonClientBarcode);

    btnClientBarcode.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View v) {
            // get the intent action string from AndroidManifest.xml
            Intent barcodeIntent = new Intent("android.intent.action.CLIENTBARCODEACTIVITY");
            startActivity(barcodeIntent);
        }
    });
}
}

Gradle build finished in 5s 681ms (20 minutes ago)
82:73 LF+ UTF-8+ Context: <no context>
```

Exemplos SDK Android: Entendendo o BarcodeExample



```
BarcodeExample - [C:\Users\E581341.GLOBAL\Documents\Femsa\25-01-2016\Surviving-with-android-master\BarcodeExample] - [app] - ...\app\src\main\java\com\honeywell\barcodeexample\ClientBarcodeActivity.java - Android St...  
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help  
BarcodeExample app src main java com honeywell barcodeexample ClientBarcodeActivity  
Project  
Android  
app  
manifests  
java  
com.honeywell.barcodeexample  
AutomaticBarcodeExample  
ClientBarcodeExample  
MainActivity  
res  
Gradle Scripts  
Captures  
Build Variants  
Favorites  
Gradle  
Android Model  
Event Log  
Gradle Console  
61:1/162 LF: UTF-8 Context: <no context>  
Gradle build finished in 5s 681ms (10 minutes ago)
```

```
barcodeReader.addTriggerListener(this);  
  
Map<String, Object> properties = new HashMap<>();  
// Set Symbologies On/Off  
properties.put(BarcodeReader.PROPERTY_CODE_128_ENABLED, true);  
properties.put(BarcodeReader.PROPERTY_GS1_128_ENABLED, true);  
properties.put(BarcodeReader.PROPERTY_QR_CODE_ENABLED, true);  
properties.put(BarcodeReader.PROPERTY_CODE_39_ENABLED, true);  
properties.put(BarcodeReader.PROPERTY_DATAMATRIX_ENABLED, true);  
properties.put(BarcodeReader.PROPERTY_UPC_A_ENABLED, true);  
properties.put(BarcodeReader.PROPERTY_EAN_8_ENABLED, true);  
properties.put(BarcodeReader.PROPERTY_EAN_8_CHECK_DIGIT_TRANSMIT_ENABLED, true);  
properties.put(BarcodeReader.PROPERTY_EAN_13_ENABLED, false);  
properties.put(BarcodeReader.PROPERTY_AZTEC_ENABLED, false);  
properties.put(BarcodeReader.PROPERTY_CODABAR_ENABLED, false);  
properties.put(BarcodeReader.PROPERTY_INTERLEAVED_25_ENABLED, false);  
properties.put(BarcodeReader.PROPERTY_PDF_417_ENABLED, false);  
// Set Max Code 39 barcode length  
properties.put(BarcodeReader.PROPERTY_CODE_39_MAXIMUM_LENGTH, 10);  
// Turn on center decoding  
properties.put(BarcodeReader.PROPERTY_CENTER_DECODE, true);  
// Disable bad read response, handle in onFailureEvent  
properties.put(BarcodeReader.PROPERTY_NOTIFICATION_BAD_READ_ENABLED, false);  
// Apply the settings  
barcodeReader.setProperties(properties);  
}  
  
// get initial list  
barcodeList = (ListView) findViewById(R.id.listViewBarcodeData);  
  
public boolean dispatchKeyEvent(KeyEvent event) {  
    int tecla;  
    tecla=0;
```

Exemplos SDK Android: Entendendo o BarcodeExample



(..\honeywell-android-data-collection-sdk\docs\api\com\honeywell\aidc\BarcodeReader.html)



```

aim
public void aim(boolean on)
    throws ScannerNotClaimedException,
           ScannerUnavailableException

Sets the aiming state of the scanner.

Parameters:
    on -
        • True: turn on scan aimer
        • False: turn off scan aimer

Throws:
    ScannerNotClaimedException
    ScannerUnavailableException

    .BarcodeListener,

light
public void light(boolean on)
    throws ScannerNotClaimedException,
           ScannerUnavailableException

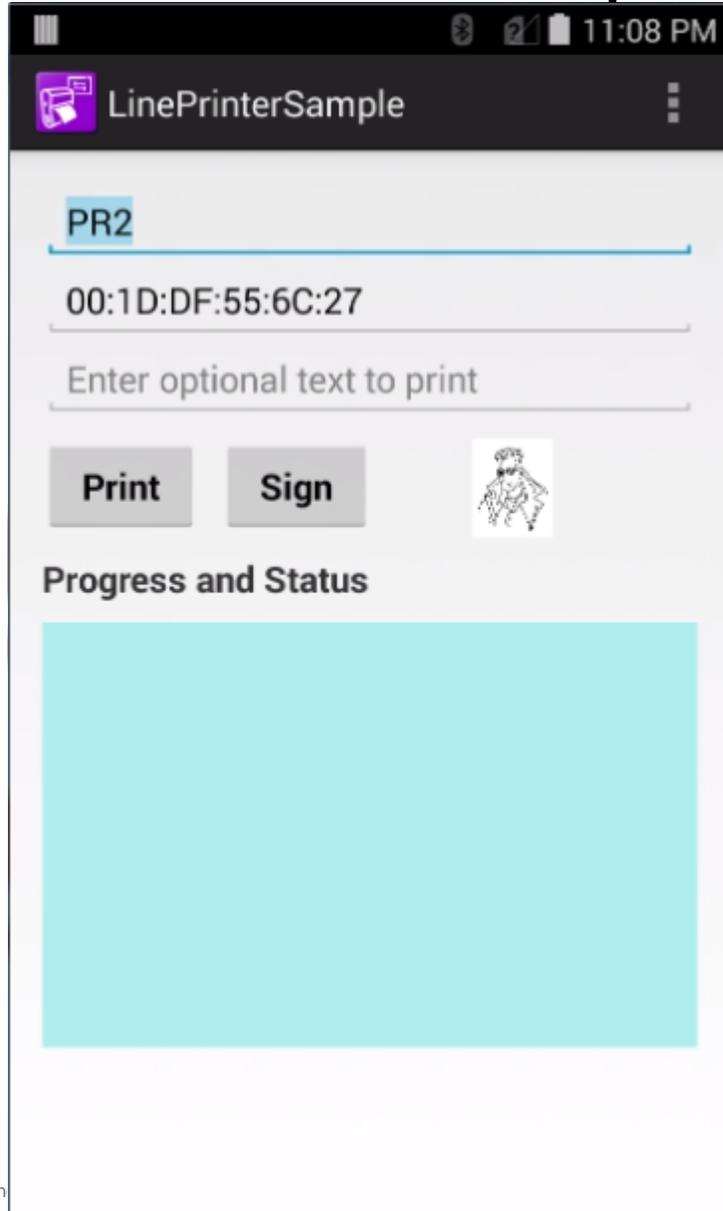
Sets the illumination state of the scanner.

Parameters:
    on -
        • True: turn on scan light
        • False: turn off scan light

Throws:
    ScannerNotClaimedException
    ScannerUnavailableException

    session
    // associated with the internal imager
  
```

Exemplos SDK Dolphin CT50/75E/CN51/EDA50 : LinePrinterSample PR2/PR3



Exemplos SDK Dolphin CT50/75E/CN51/EDA50 : Documentação LinePrinterSample PR2/PR3

(.. \honeywell-android-printing-sdk_r01\honeywell-android-printing-sdk\docs\api\com\honeywell\mobility\print\LinePrinter.html

Overview

Prev Class Next Class Frames No Frames All Classes

Summary: Nested | Field | Constr | Method Detail: Field | Constr | Method

com.honeywell.mobility.print

Class LinePrinter

java.lang.Object
com.honeywell.mobility.print.Printer
com.honeywell.mobility.print.LinePrinter

```
public class LinePrinter
extends Printer
```

This class represents a Honeywell mobile receipt printer and provides methods to communicate with the printer.

Overview of LinePrinter Features

- Communicate with the printer via Bluetooth.
- Sends data string or byte array to the printer for receipt printing or custom printer commands.
- Change text font styles. Font styles supported by the LinePrinter class include bold, compressed, double high, double wide, italic, strikethrough and underline. Font style commands are ignored without generating an error condition if sent to a printer that does not support the requested font styles.
- Print bitmap images and bar codes.
- Provide the progress of the receipt printing via the `PrintProgressEvent`. These events inform the application of when the printing is started and when the printing is completed.

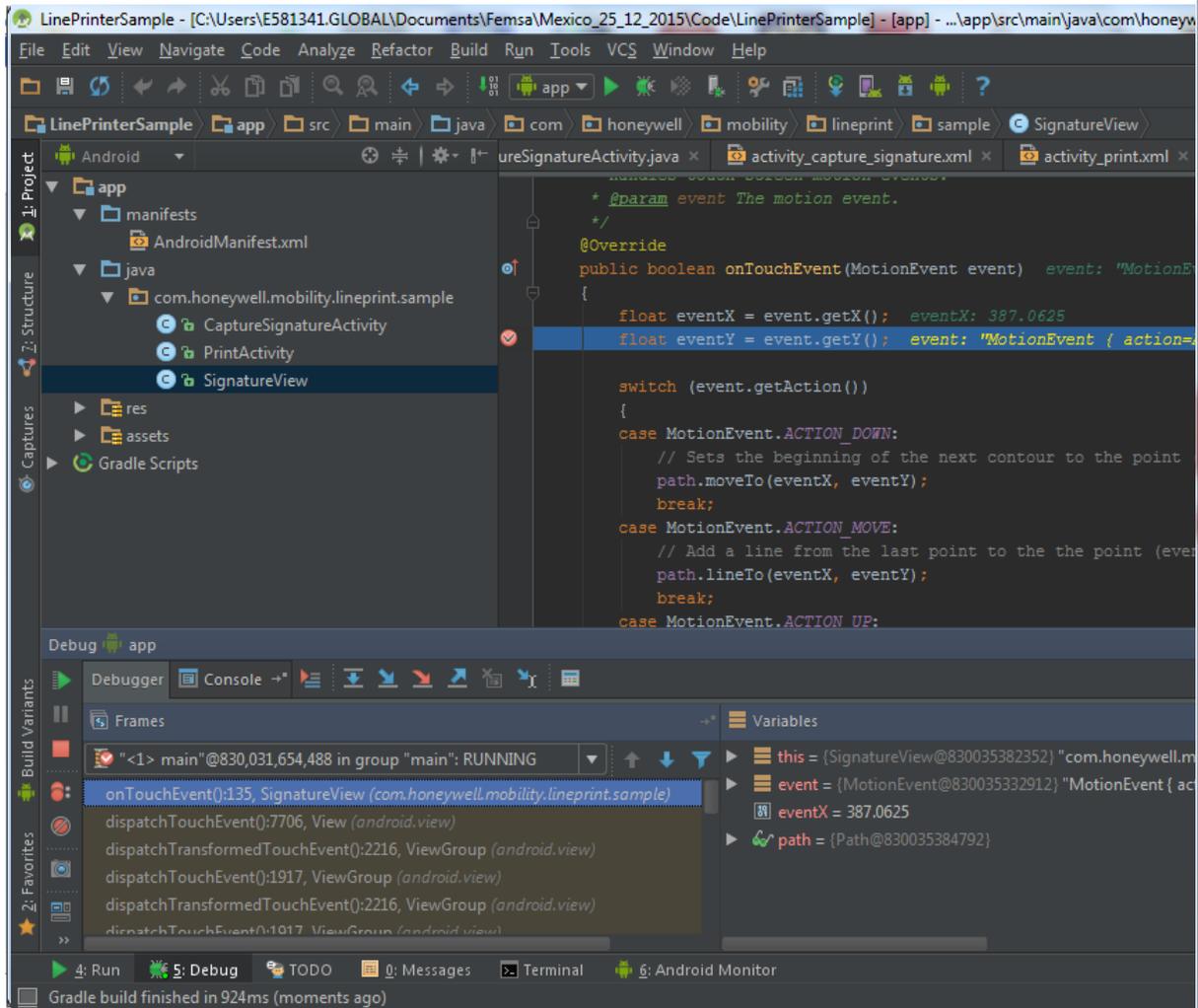
In release 1.30, the LinePrinter class became a subclass of the `Printer` class. Some public methods have been moved to the `Printer` class and inherited by the `LinePrinter` class. For backward compatibility, these inherited methods still throw `LinePrinterException` if they are called from a `LinePrinter` object. The `LinePrinterException` class is a subclass of `PrinterException` so you may catch the exceptions with either class.

Note: It is generally a good practice to create a separate thread for printing because the UI may not be responsive if the printing is done in the main UI thread. On the Android platform, the application must call the `LinePrinter` constructors and methods in a separate thread from the main UI thread; otherwise, a `LinePrinterException` will be thrown. The only exceptions to this requirement are the methods that add or remove an event listener. This requirement is to prevent the Android "Application Not Responding" dialog from being displayed because communicating with the printer via network connection may invoke unpredictable delays.

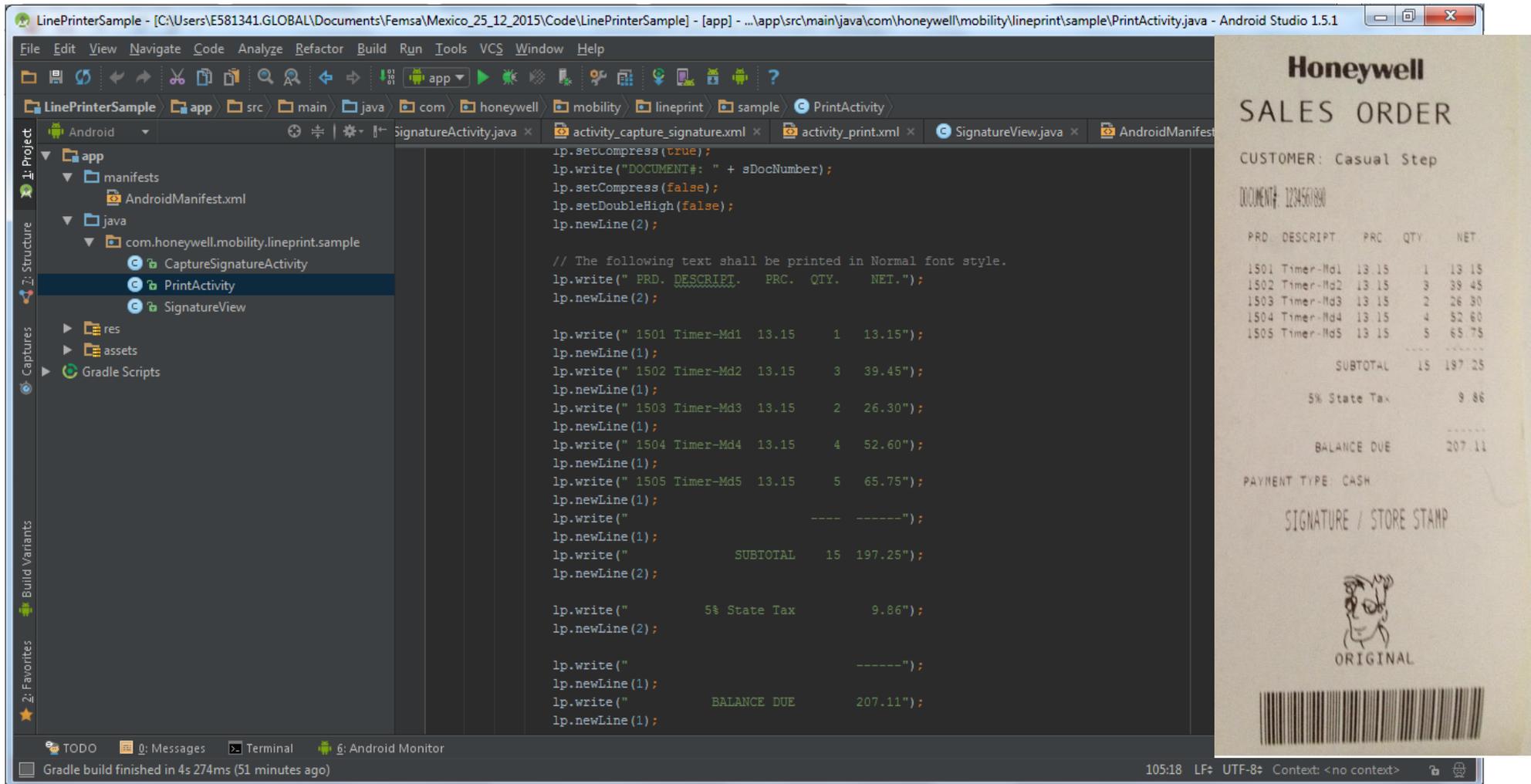
The following is a sample code snippet that demonstrates calling the `LinePrinter` API on the Android platform.

```
import android.os.AsyncTask;
import android.app.Activity;
```

Exemplos SDK Dolphin CT50/75E/CN51/EDA50 : LinePrinterSample PR2/PR3



Exemplos SDK Dolphin CT50/75E/CN51/EDA50: LinePrinterSample PR2/PR3



The screenshot shows the Android Studio 1.5.1 interface. The main editor displays the Java code for `PrintActivity.java` in the `com.honeywell.mobility.lineprint.sample` package. The code is as follows:

```

lp.setCompress(true);
lp.write("DOCUMENT#: " + sDocNumber);
lp.setCompress(false);
lp.setDoubleHigh(false);
lp.newLine(2);

// The following text shall be printed in Normal font style.
lp.write(" PRD.  DESCRPT.  PRC.  QTY.  NET.");
lp.newLine(2);

lp.write(" 1501 Timer-Md1  13.15    1  13.15");
lp.newLine(1);
lp.write(" 1502 Timer-Md2  13.15    3  39.45");
lp.newLine(1);
lp.write(" 1503 Timer-Md3  13.15    2  26.30");
lp.newLine(1);
lp.write(" 1504 Timer-Md4  13.15    4  52.60");
lp.newLine(1);
lp.write(" 1505 Timer-Md5  13.15    5  65.75");
lp.newLine(1);
lp.write("          -----");
lp.newLine(1);
lp.write("          SUBTOTAL    15  197.25");
lp.newLine(2);

lp.write("          5% State Tax    9.86");
lp.newLine(2);

lp.write("          -----");
lp.newLine(1);
lp.write("          BALANCE DUE    207.11");
lp.newLine(1);

```

The document generated by the code is a sales order from Honeywell. The document content is as follows:

Honeywell
SALES ORDER
CUSTOMER: Casual Step
DOCUMENT#: 1234567890

PRD.	DESCRPT.	PRC.	QTY.	NET.
1501	Timer-Md1	13.15	1	13.15
1502	Timer-Md2	13.15	3	39.45
1503	Timer-Md3	13.15	2	26.30
1504	Timer-Md4	13.15	4	52.60
1505	Timer-Md5	13.15	5	65.75
SUBTOTAL				15 197.25
5% State Tax				9.86
BALANCE DUE				207.11

PAYMENT TYPE: CASH
SIGNATURE / STORE STAMP

ORIGINAL

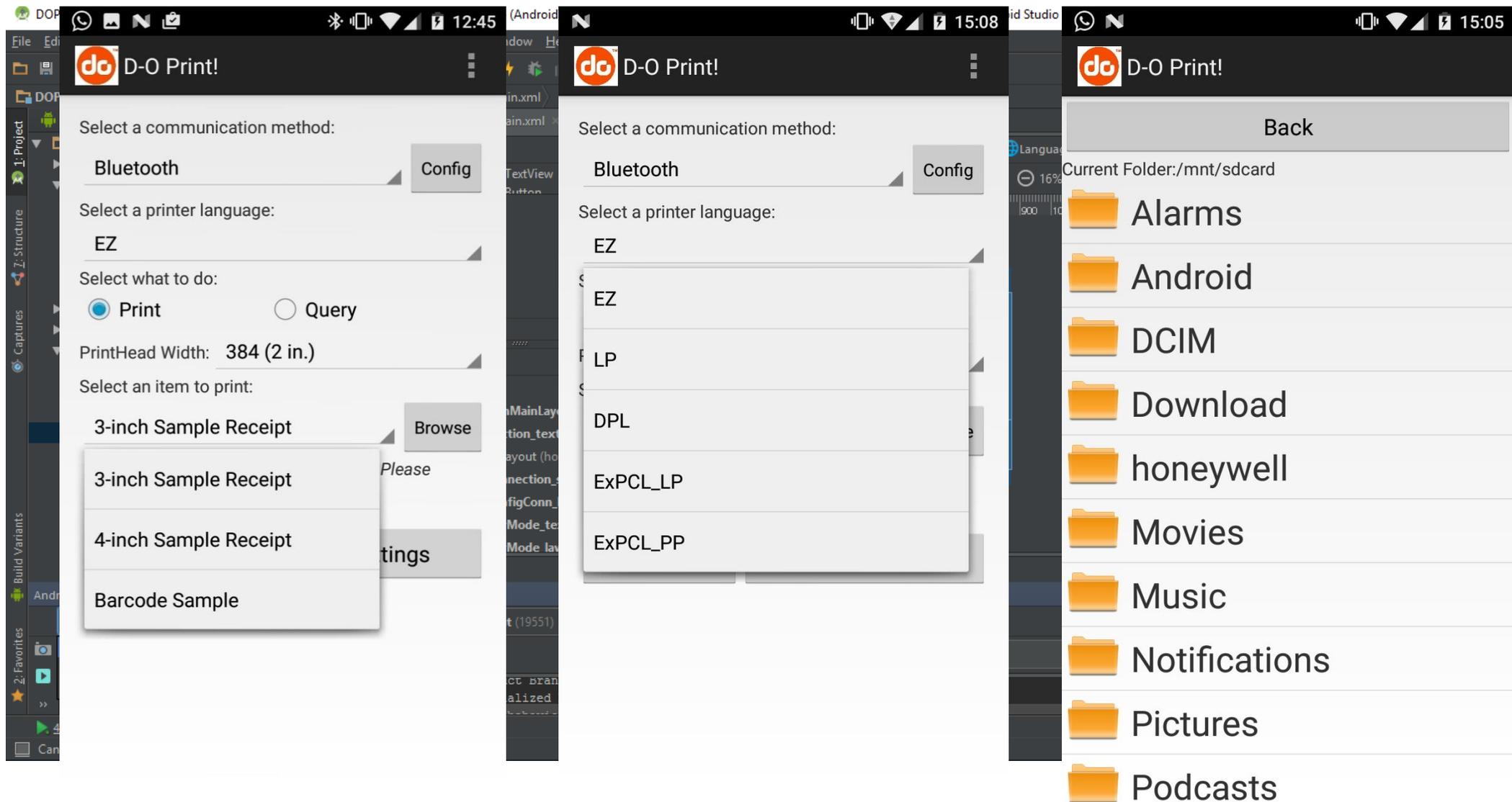

Exemplos Honeywell SDK Android: DoPrint! RL3/RL4

The screenshot shows the Honeywell Software Download Manager interface. The browser address bar displays 'author-hsmftp.honeywell.com'. The application window has a title bar 'Honeywell Software Download Manager' and a menu bar with 'Start', 'Pause', 'Remove', 'Settings', and 'Help'. Below the menu bar is a table of files for download:

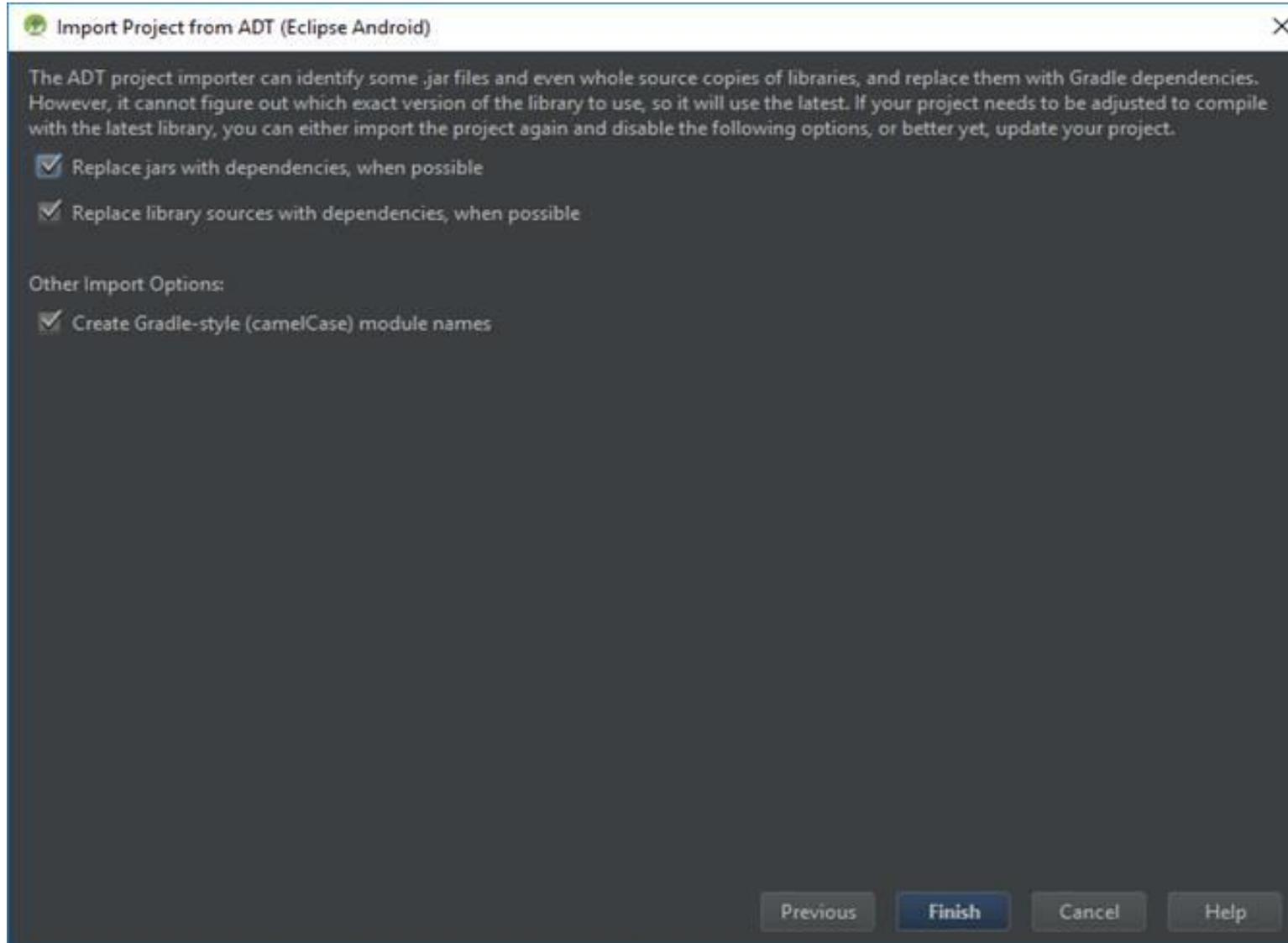
File	Size	Status	Progress	Download
DatamaxONeilSDK_v2.4.4 (Android).zip (Downloa...	52.45 MB	1.78 MB	3.39%	Download
DatamaxONeilSDK_v2.4.4 (Java).zip SDK for Java				Download

At the bottom of the interface, there is a status bar showing 'Rate: 70.25 kbps' and 'Version : 1.6'. The background shows a file explorer view of the website's directory structure, including folders like 'Printers', 'Desktop', 'Industrial', 'Mobile - Po', '6822', 'Apex2 A', 'LP3', 'MF2te I', 'OC2 OQ', 'PB21 P', 'PB42', 'PB50 P', 'PR2 PR', 'PrintPa', 'RL3 RL', 'Arck', 'Cur', 'Utility', 'RP2 RP3 RP4', and 'Printer Applications'.

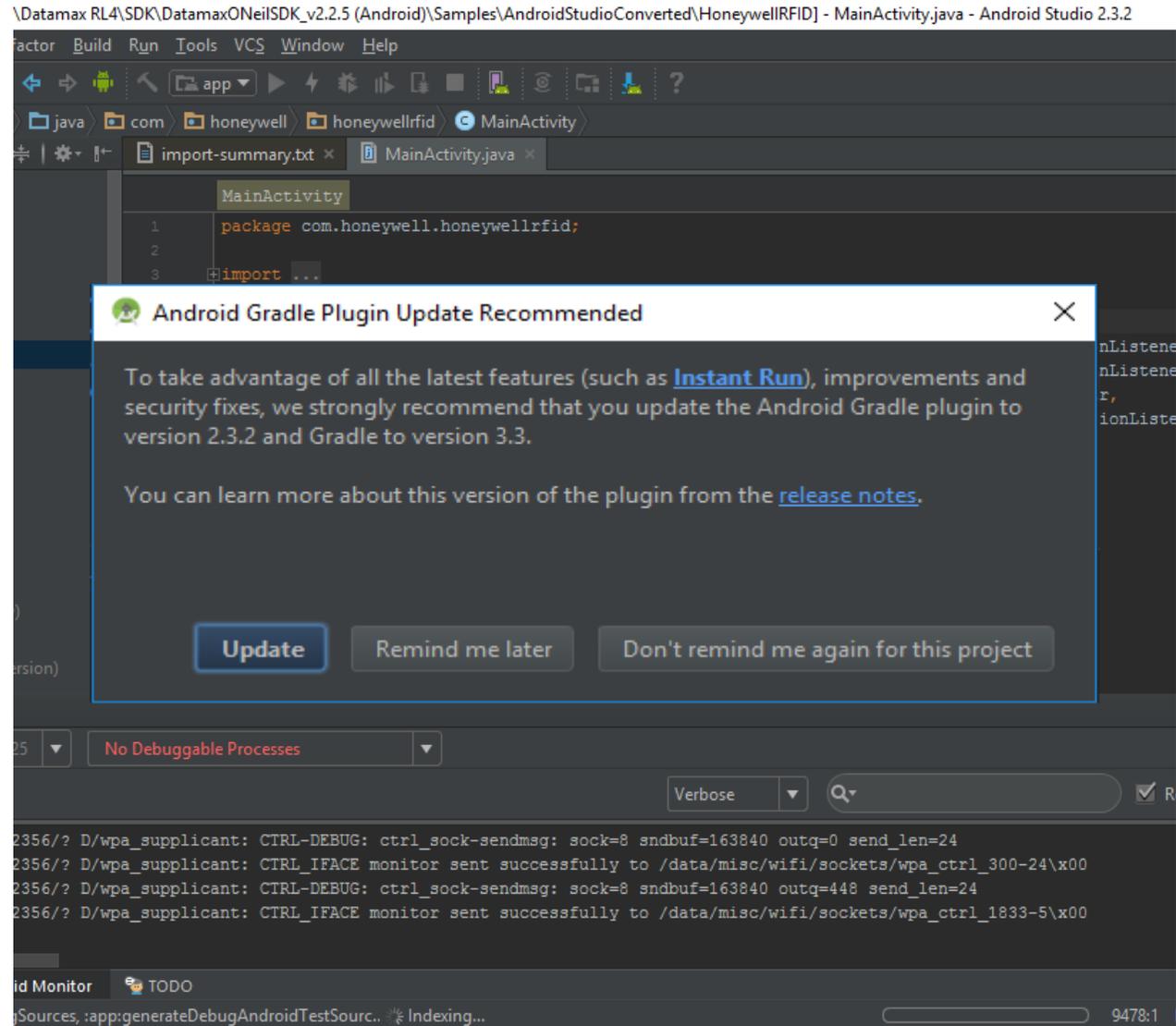
Exemplos Honeywell SDK Android: DoPrint! RL3/RL4



Exemplo Porta Serial RFID IP30



Exemplo Porta Serial RFID IP30



Exemplo Porta Serial RFID IP30



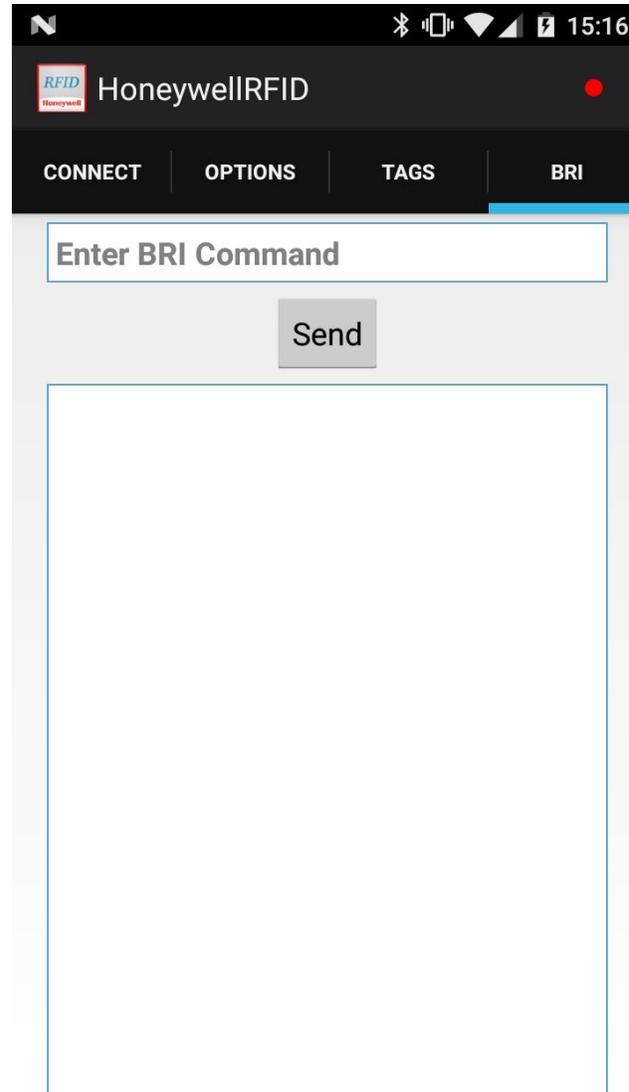
RFID HoneywellRFID

CONNECT OPTIONS TAGS BRI

6 dB RF Power=20 dB 30 dB

Unique tags=0

To locate a tag, select one from the list. To stop locating, [tap here](#).

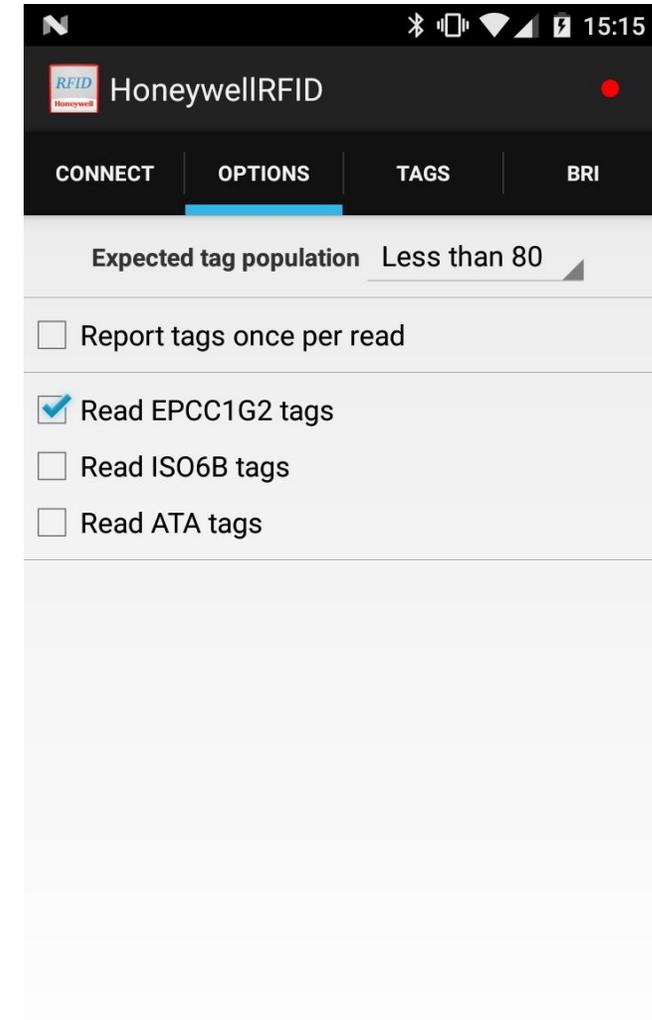


RFID HoneywellRFID

CONNECT OPTIONS TAGS BRI

Enter BRI Command

Send



RFID HoneywellRFID

CONNECT OPTIONS TAGS BRI

Expected tag population Less than 80

Report tags once per read

Read EPCC1G2 tags

Read ISO6B tags

Read ATA tags

Exemplo Generalista: BTPrint4

The screenshot shows a web browser displaying the GitHub repository page for 'hjgode / BtPrint4'. The browser's address bar shows the URL 'https://github.com/hjgode/BtPrint4'. The repository page includes a navigation bar with 'This repository', 'Search', 'Pull requests', 'Issues', 'Marketplace', and 'Gist'. Below the navigation bar, the repository name 'hjgode / BtPrint4' is displayed, along with statistics: 7 Watch, 15 Star, and 14 Fork. The 'Code' tab is selected, showing a list of files and folders. The repository has 27 commits, 2 branches, 0 releases, and 1 contributor. The commit history shows a recent commit by 'hjgode' adding a binary file 'apk/app-debug.apk' on Jan 23. Other files include '.idea', 'app', 'build/intermediates/dex-cache', 'doc', 'doc2', 'gradle/wrapper', '.gitignore', and 'BtPrint4.iml'.

my first android steps

27 commits 2 branches 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

File/Folder	Description	Time
hjgode	added binary apk/app-debug.apk	Latest commit af892fc on Jan 23
.idea	updated for current AS and Build tools 22.0.1	4 months ago
app	added binary apk/app-debug.apk	4 months ago
build/intermediates/dex-cache	updated for current AS and Build tools 22.0.1	4 months ago
doc	file explorer ready to use	3 years ago
doc2	added images for play store	2 years ago
gradle/wrapper	updated for current AS and Build tools 22.0.1	4 months ago
.gitignore	Initial commit	3 years ago
BtPrint4.iml	updated for current AS and Build tools 22.0.1	4 months ago

Perguntas?



Boris Adrián Torrado Bonilla
Solutions Architect

Honeywell | Safety & Productivity Solutions

Office: (+55 11) 3711-6794

Mobile: (+55 11) 99340-6608

boris.bonilla@honeywell.com

www.honeywellaidc.com | aidc.honeywell.com | www.datamax-oneil.com

Twitter: [@HoneywellAIDC](https://twitter.com/HoneywellAIDC)

Base de Conhecimento:

<http://hsm.force.com/publickb>

Portal do Suporte Técnico:

<https://hsm.secure.force.com/thetechsupportall/>

Telefone:

[\(11\) 4700-2155 Opção 4](tel:(11)4700-2155)

Email:

ACSHSMSuporteBrasil@Honeywell.com

