



Programação para Android

Aula 01: Visão geral do android, instalação e configuração do ambiente de desenvolvimento, estrutura básica de uma aplicação para Android

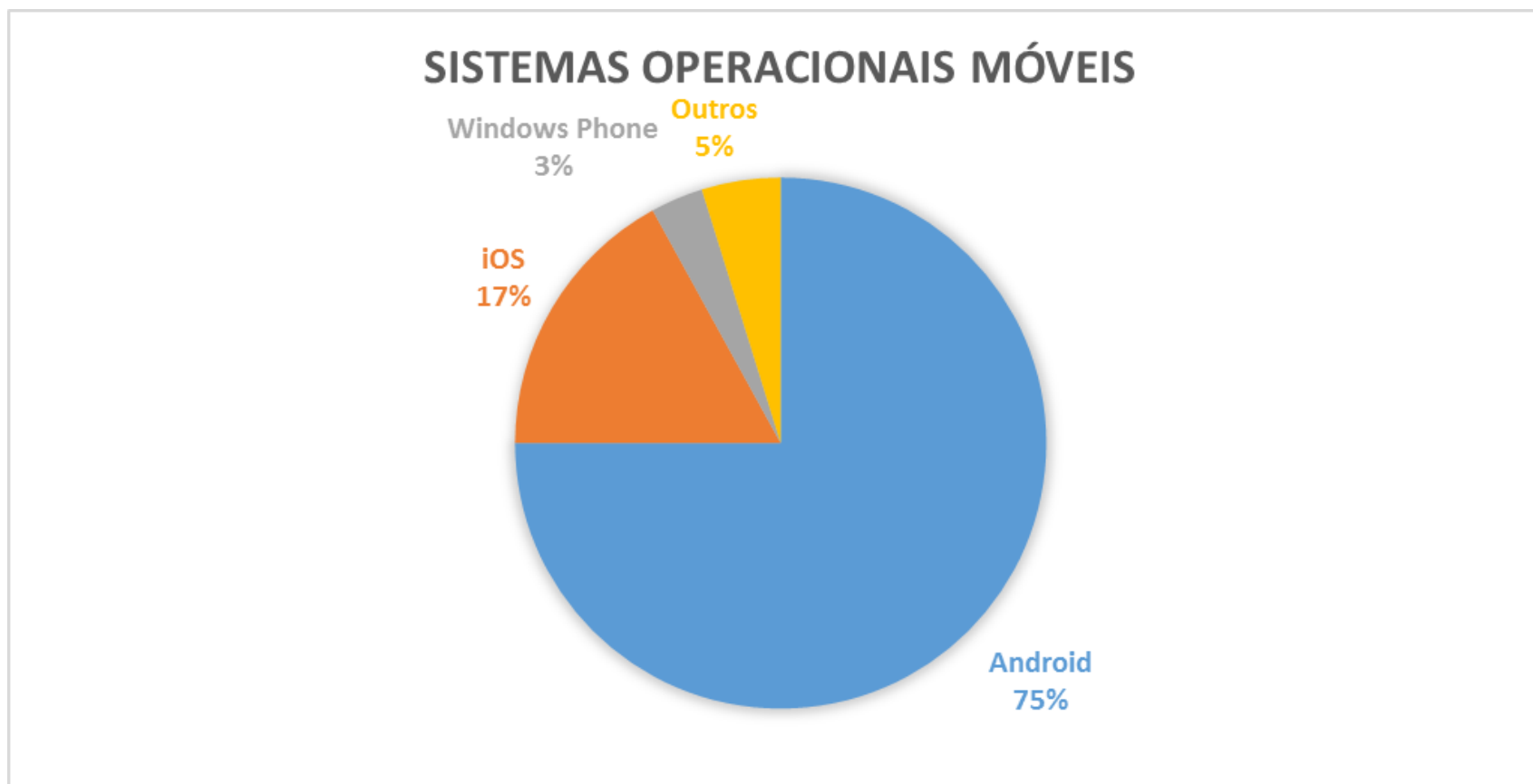
Objetivos

- ▶ Configurar o ambiente de trabalho e o dispositivo de teste (físico e virtual)
- ▶ Criar projetos
- ▶ Conhecer a estrutura de um projeto de aplicação Android
- ▶ Criar, compilar e executar uma aplicação Android



Parte 01: Visão geral do Android

Porque aprender programação para Android?



Fonte: Olhar Digital, maio de 2013

Visão geral do Android

- ▶ A primeira geração de telefones Android foi lançada em outubro de 2008.
- ▶ O sistema operacional Android foi desenvolvido pela Android, Inc., adquirida pelo Google em julho de 2005.
- ▶ Em novembro de 2007, foi formada a Open Handset Alliance, inicialmente formado por 34 empresas, para desenvolver Android, impulsionando a inovação na tecnologia móvel, melhorando a experiência do usuário e reduzindo os custos.
- ▶ Android é usado em Smartphones, dispositivos e-reader e tablets



Visão geral do Android

- ▶ Código-fonte aberto e gratuito
- ▶ Os aplicativos Android são desenvolvidos com Java
- ▶ Programação da interface gráfica é baseada em eventos: toques na tela e pressionamento de tecla
- ▶ Os dispositivos Android vêm com vários aplicativos incorporados: telefone, contatos, correio, navegadores, etc.
- ▶ Cada nova versão Android recebe um nome de sobremesa, em inglês.

Visão geral do Android

► Nome das versões



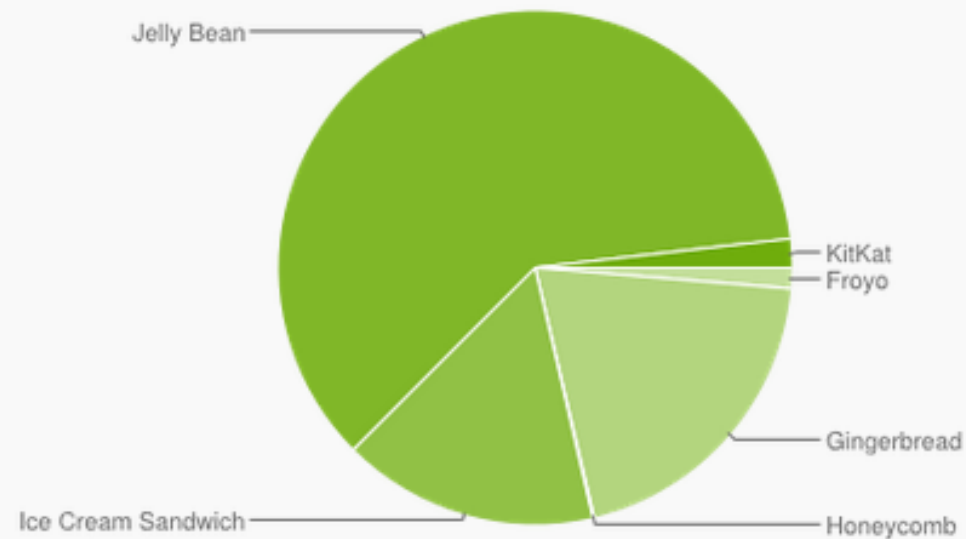
Visão geral do Android

- ▶ Versões 3.x: apenas para Tablets
- ▶ Versão única para tablets e smartphones a partir da versão 4.x
- ▶ Versões mais atuais:
 - ▶ 4.0: Ice Cream Sandwich (Outubro de 2011)
 - ▶ 4.1 - 4.2 - 4.3: Jelly Bean (Junho de 2012)
 - ▶ 4.4: KitKat (Versão atual lançada em Outubro de 2013)

Visão geral do Android

- Número de dispositivo x versões que utilizam o sistema Android

Version	Codename	API	Distribution
2.2	Froyo	8	1.3%
2.3.3 - 2.3.7	Gingerbread	10	20.0%
3.2	Honeycomb	13	0.1%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	16.1%
4.1.x	Jelly Bean	16	35.5%
4.2.x		17	16.3%
4.3		18	8.9%
4.4	KitKat	19	1.8%

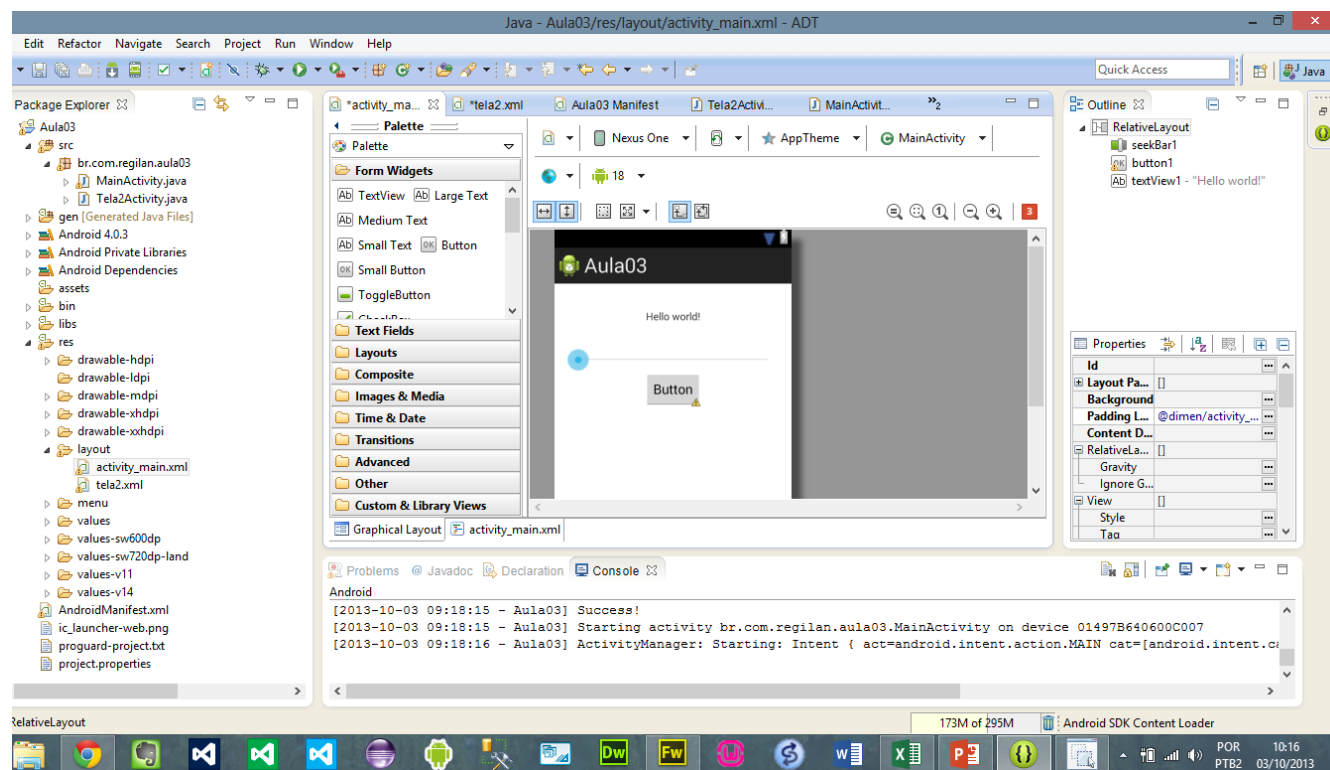


O SDK do Android

- ▶ SDK - Software Development Kit: fornece as ferramentas necessárias para construir aplicativos Android
- ▶ Esta disponível gratuitamente no site Android Developers:
<http://developer.android.com/sdk/index.html>
- ▶ Existe uma versão Preview de uma nova ferramenta: Android Studio 0.4.x for Windows
- ▶ O SDK é composto pelo JAVA SE, IDE Eclipse, SDK do Android e o Plugin ADT para eclipse

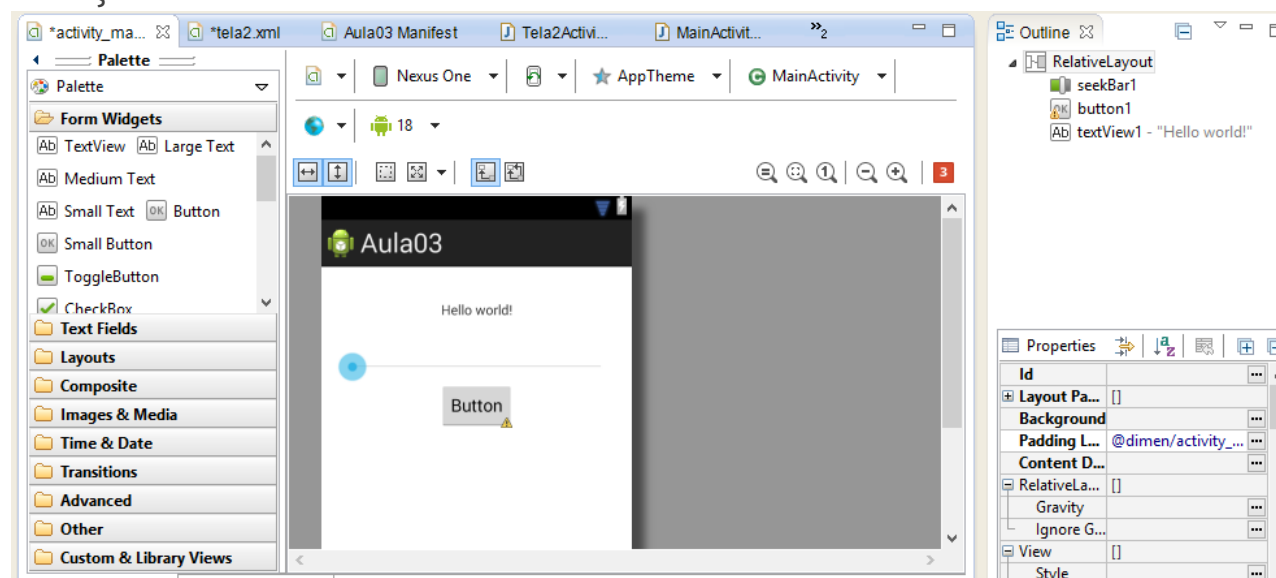
O SDK do Android

- **IDE Eclipse:** ambiente de desenvolvimento integrado recomendado para desenvolvimento Android, que possui editor de código, depurador, preenchimento automático, sintaxe colorida e outros recursos



O SDK do Android

- ▶ Plugin ADT (Android Development Tools) para Eclipse - uma extensão para o IDE Eclipse - permite criar, executar e depurar aplicativos Android, exportá-los para distribuição.
- ▶ O ADT contém uma ferramenta de projeto visual de interface gráfica do usuário, que podem ser arrastados e soltos no lugar para forma interfaces sem nenhuma codificação.



O emulador do Android

- ▶ O emulador do Android, incluído no SDK do Android, permite executar aplicativos Android em um ambiente simulado dentro do Windows, Mac OS X ou Linux.
- ▶ O emulador exibe uma janela de interface de usuário, porém antes de executar o aplicativo no emulador, você precisa criar um AVD (Android Virtual Device), o qual define as características do dispositivo, incluindo hardware, tamanho da tela, armazenamento, etc.



Google Play

- ▶ Google Play é a loja online mantida pela Google para distribuição de aplicações, jogos, filmes, música e livros. Anteriormente a loja chamava-se Android Market.
- ▶ As aplicações do Google Play estão disponíveis de graça ou a um custo, e podem ser baixados diretamente para um dispositivo Android.
- ▶ Para distribuir seus produtos por meio do Google Play há uma taxa de registro de US\$ 25 cobrados por uma conta de Console do desenvolvedor do Google Play.
- ▶ Em caso de aplicativos pagos, o Google fica com 30% do preço de venda e repassa 70% ao desenvolvedor



Instalação e configuração do ambiente de desenvolvimento

Instalação e configuração do ambiente de desenvolvimento

- ▶ Download do SDK no site Android Developers:
<http://developer.android.com/sdk/index.html>
- ▶ Com um único download obtemos:
 - ▶ Eclipse + ADT plugin
 - ▶ Android SDK Tools
 - ▶ Android Platform-tools
 - ▶ The latest Android platform
 - ▶ The latest Android system image for the emulator

Get the Android SDK

The Android SDK provides you the API libraries and developer tools necessary to build, test, and debug apps for Android.

If you're a new Android developer, we recommend you download the ADT Bundle to quickly start developing apps. It includes the essential Android SDK components and a version of the Eclipse IDE with built-in **ADT (Android Developer Tools)** to streamline your Android app development.

With a single download, the ADT Bundle includes everything you need to begin developing apps:

- Eclipse + ADT plugin
- Android SDK Tools
- Android Platform-tools
- The latest Android platform



Download the SDK
ADT Bundle for Windows

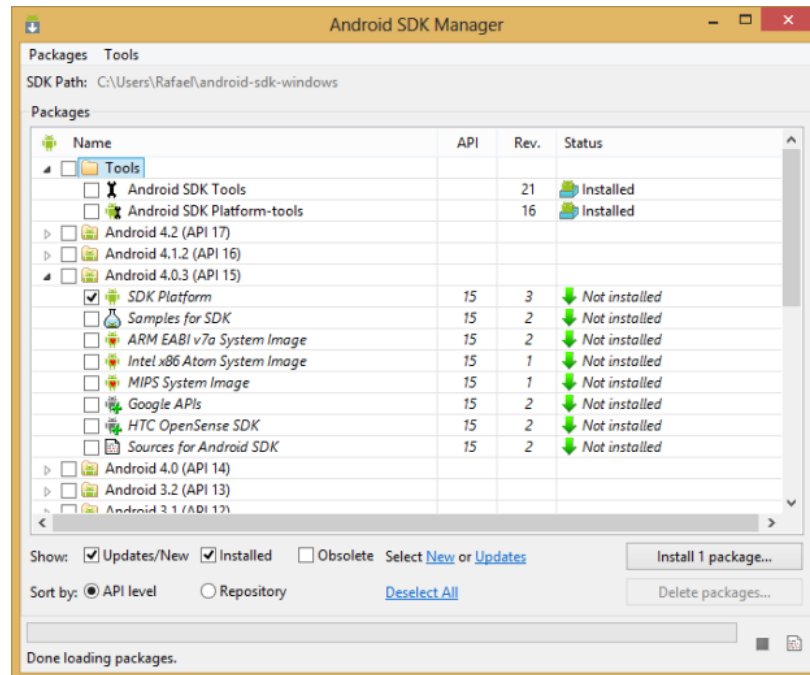
Instalação e configuração do ambiente de desenvolvimento

- ▶ O processo de instalação é bem simples, bastando seguir as instruções de instalação.
- ▶ Após o processo de download e instalação, teremos duas ferramentas importantes:
 - ▶ Eclipse: IDE de desenvolvimento
 - ▶ SDK Manager: Gerenciador do kit de desenvolvimento



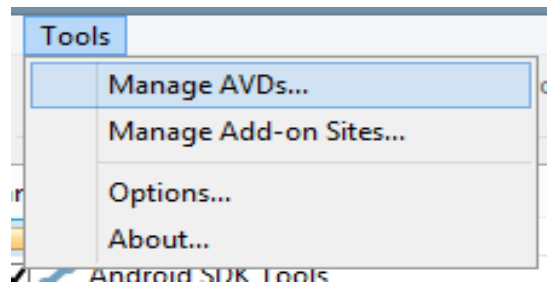
Download da versão do SDK

- ▶ Após a conclusão do download, descompacte o arquivo e execute o **SDK Manager** para baixar uma SDK para começarmos a programar.
- ▶ Ao executar pela primeira vez, o SDK Manager irá verificar os repositórios do Android em busca das últimas versões do SDK.



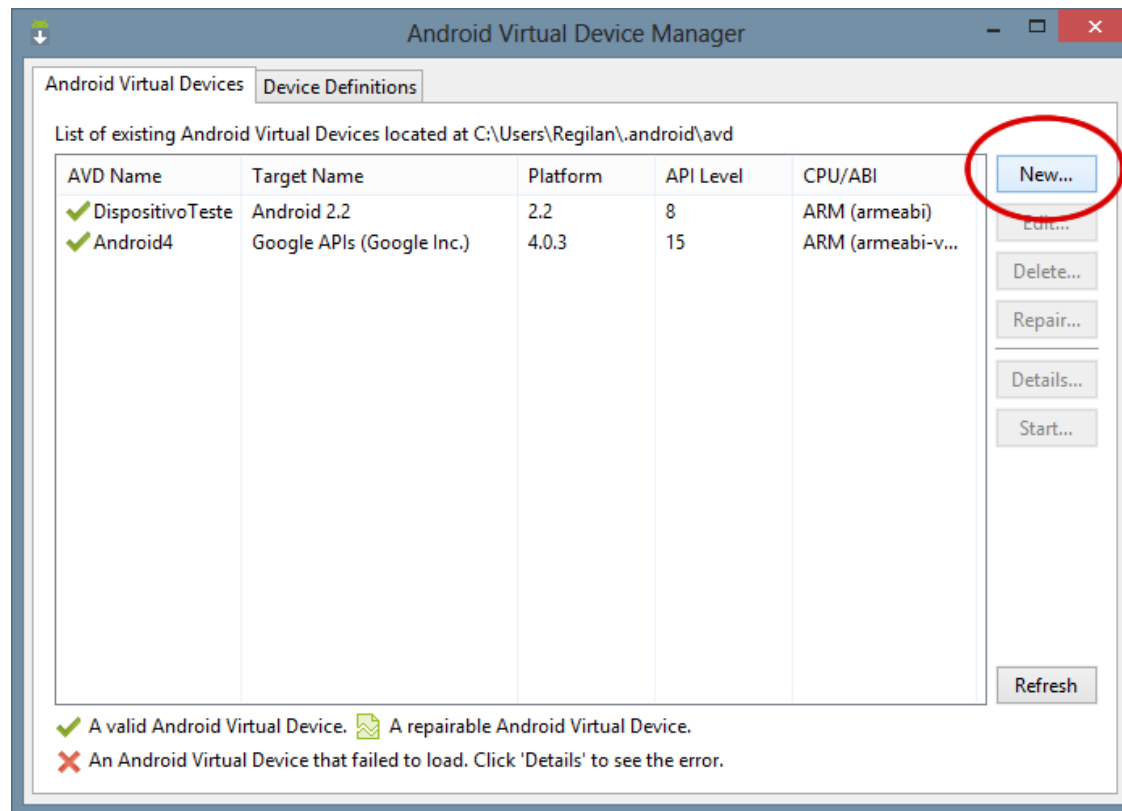
Download da versão do SDK

- ▶ Neste curso usaremos o SDK da versão **4.0.3**. Se quiser instalar outras versões, fique à vontade.
- ▶ Expanda a pasta **Android 4.0.3 (API 15)** e marque todas as opções , marque **Accept All** e então clique em **Install**. O download e configuração será feito automaticamente.
- ▶ Após instalação, iremos configurar um AVD - Android Virtual Device. Para isto clique no menu **Tools -> Manage AVD**



Configurando um AVD

- Na janela aberta, clique no botão NEW



Configurando um AVD

- ▶ Configure um nome para seu ADV (AVD Name)
- ▶ Defina um dispositivo (Device) e uma versão do Android (Target)
- ▶ Neste curso usaremos como versão do Android(Target), a opção Google Api - Api level 15
- ▶ Note a opção de criar um SDCard. Este recurso serve para você salvar informações no emulador. Como Configurações, arquivos, aplicativos instalado e etc. Você deve informar um valor em MB por exemplo. EX: 50 MB
- ▶ **OBS: API level é um valor inteiro que identifica uma versão do Android. Recomendamos escolher sempre a versão Google API, pois estas incluem recursos disponíveis da plataforma Google.**

Create new Android Virtual Device (AVD)

AVD Name:

Device:

Target:

CPU/ABI:

Keyboard: ☒ Hardware keyboard present

Skin: ☒ Display a skin with hardware controls

Front Camera:

Back Camera:

Memory Options: RAM: VM Heap:

Internal Storage: MiB

SD Card: ☒ Size: MiB ☐ File: Browse...

Emulation Options: ☐ Snapshot ☐ Use Host GPU

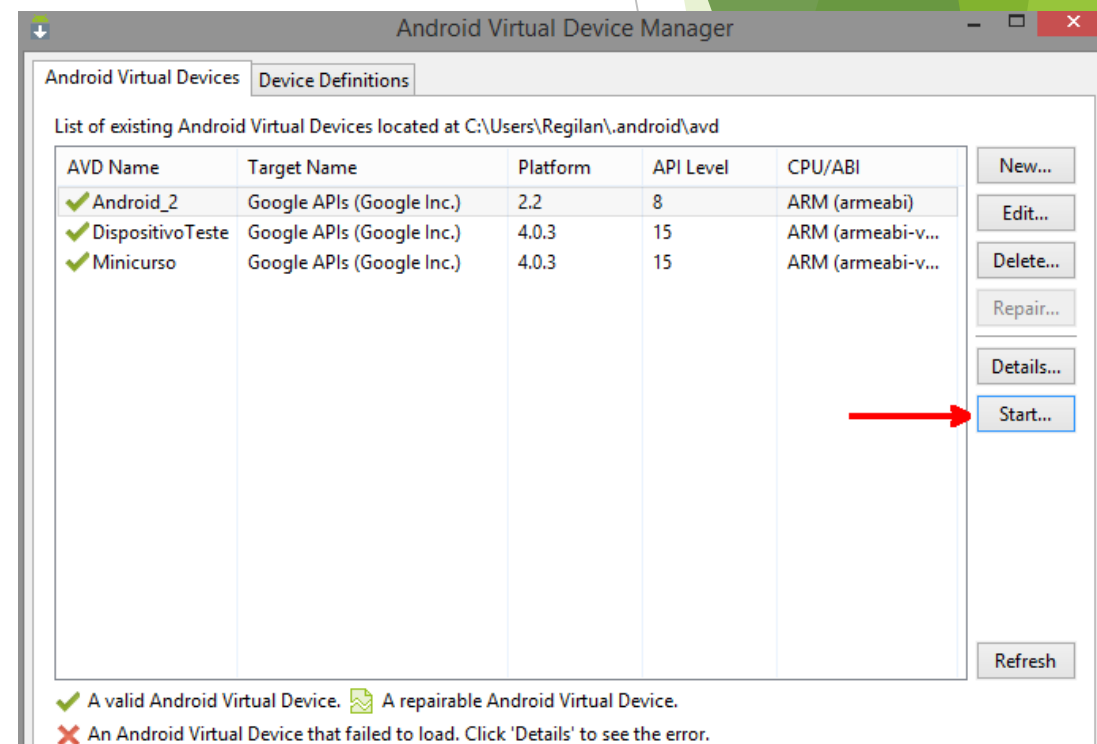
☐ Override the existing AVD with the same name

✗ AVD Name cannot be empty

OK Cancel

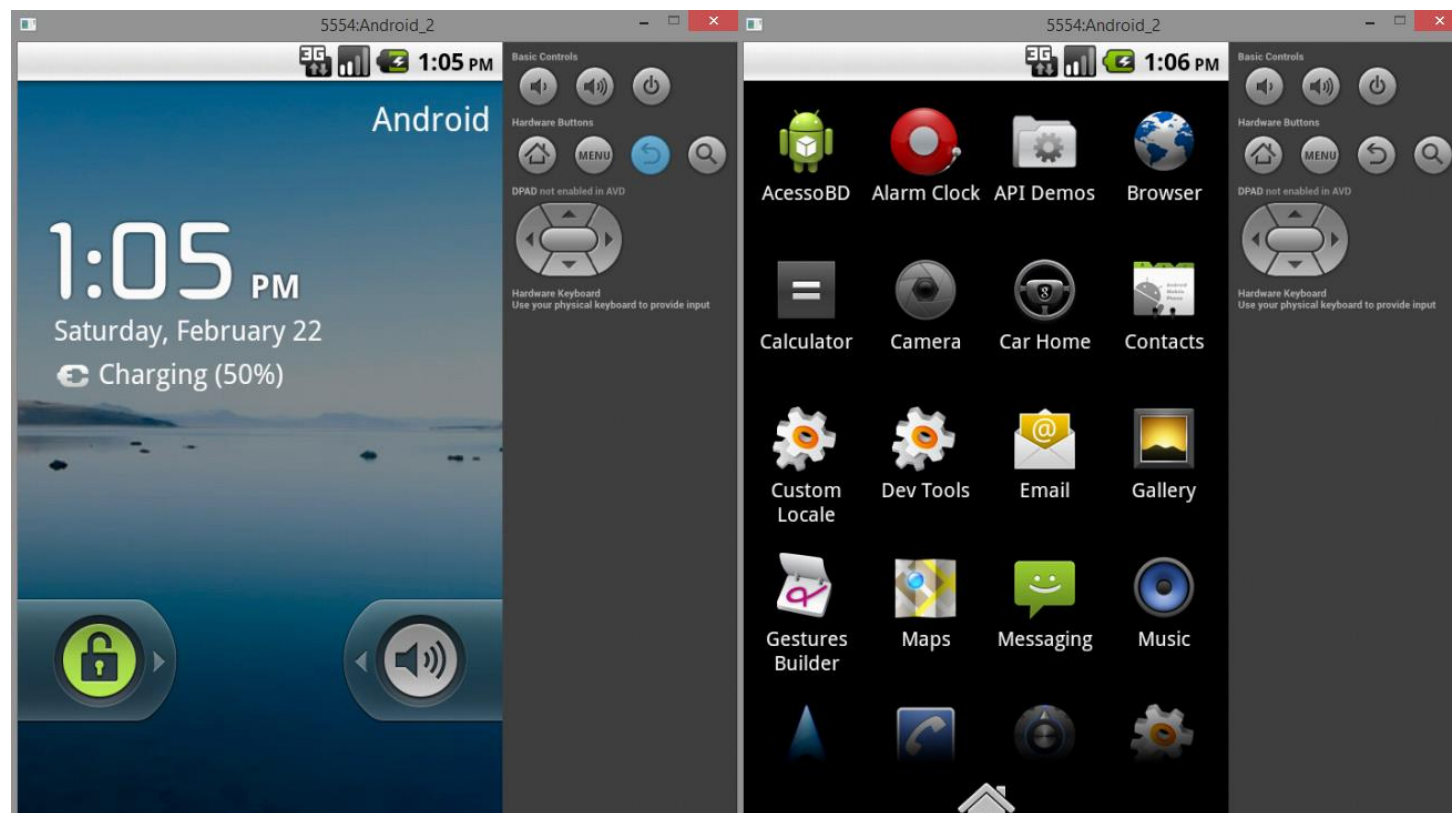
Executando um AVD

- ▶ Devido ao tempo consumido para iniciar um AVD, sempre que o mesmo for usado para testes recomendamos inicia-lo e somente fecha-lo após o termo da programação do aplicativo. Caso seja encerrado, uma nova inicialização será realizada.
- ▶ Para executar um AVD devemos escolher um dispositivo configurado e clicar no botão Start.
- ▶ Após este processo será carregado uma janela com um dispositivo ANDROID com as opções da plataforma escolhida. Lembre-se, como se trata de um dispositivo virtual alguns periféricos estão indisponíveis como GPS, câmera, etc.



Executando um AVD

- Dispositivo virtual em execução



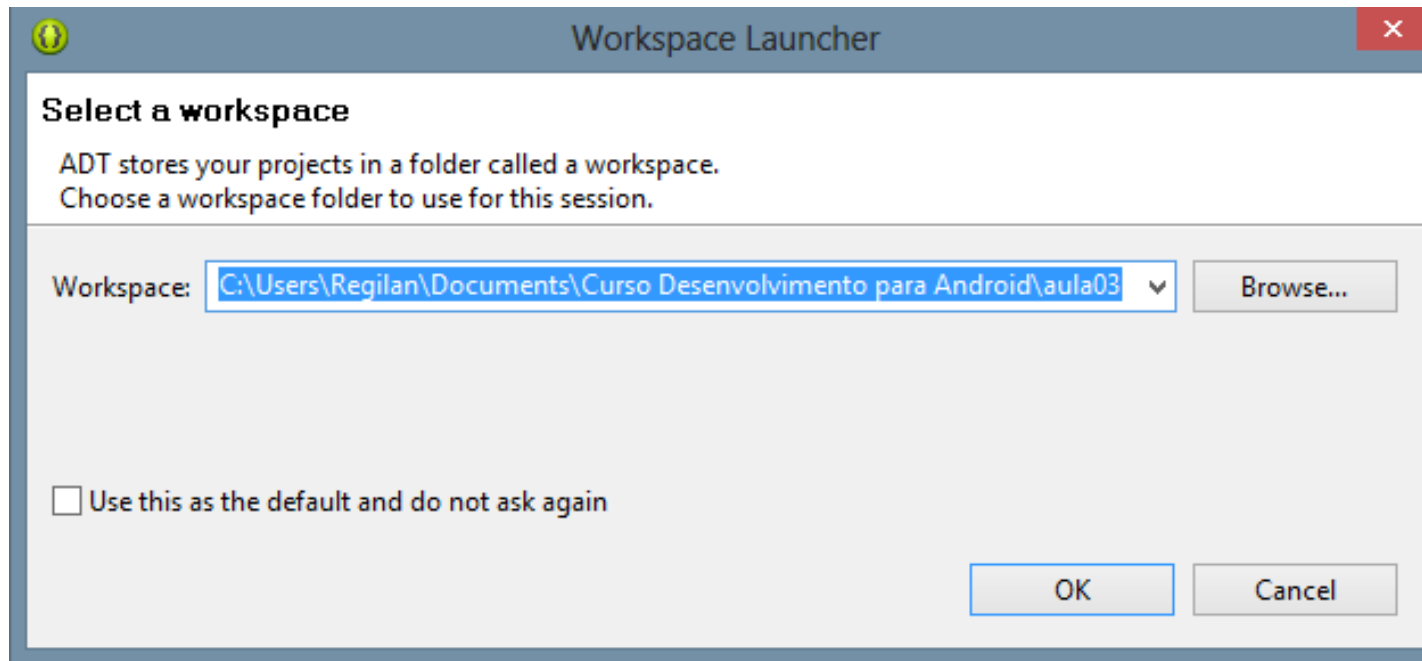
Inicializando o Eclipse

- ▶ Após a conclusão dos downloads e configuração do AVD, iniciaremos o Eclipse.



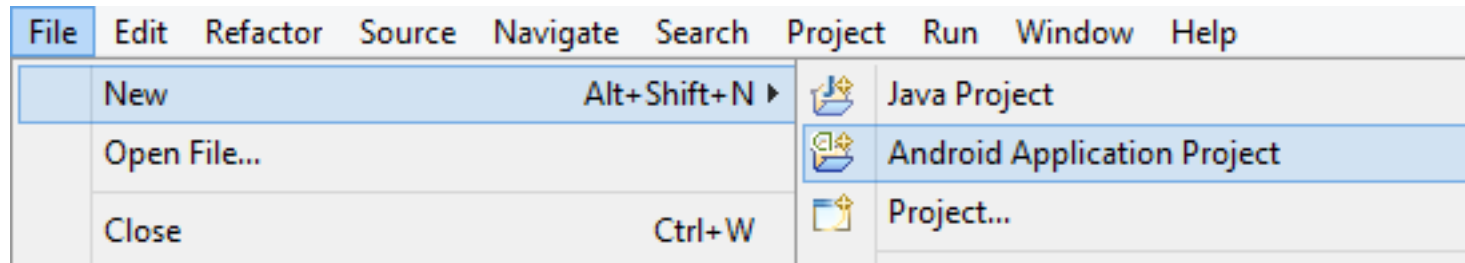
Inicializando o Eclipse

- ▶ Ao ser consultado sobre qual workspace (**workspace** é o local onde seus projetos serão salvos) utilizar, basta definir um local e utiliza-lo como padrão



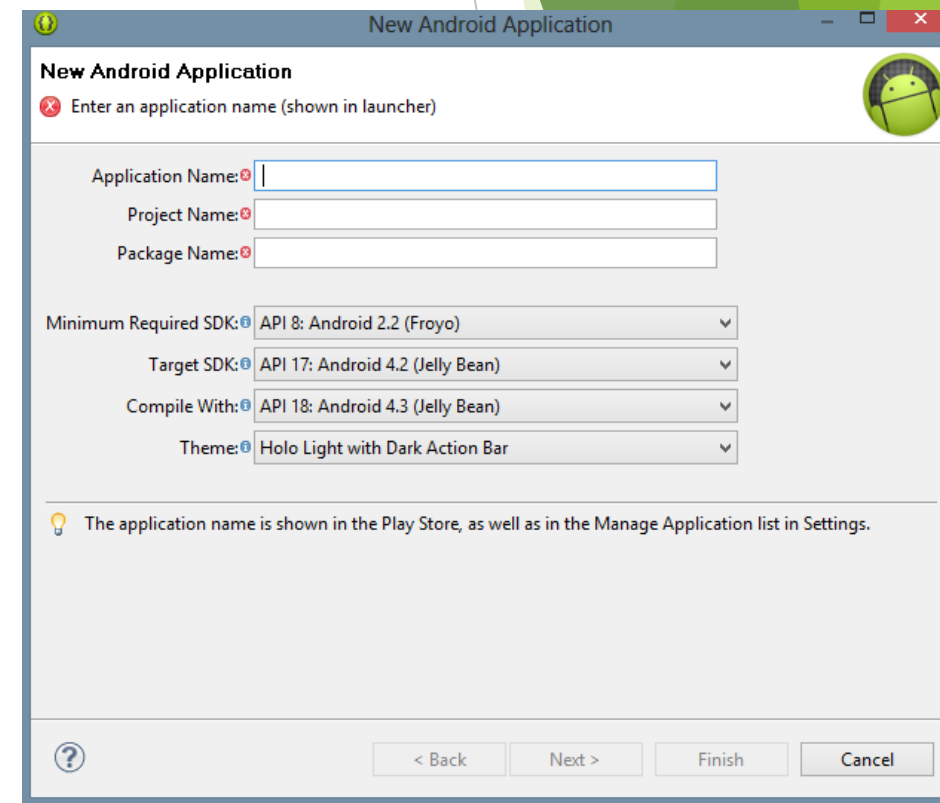
Criando um novo projeto para Android

- Para criar um novo projeto para Android, clique no menu:
FILE -> NEW -> ANDROID APPLICATION PROJET



Criando um novo projeto para Android

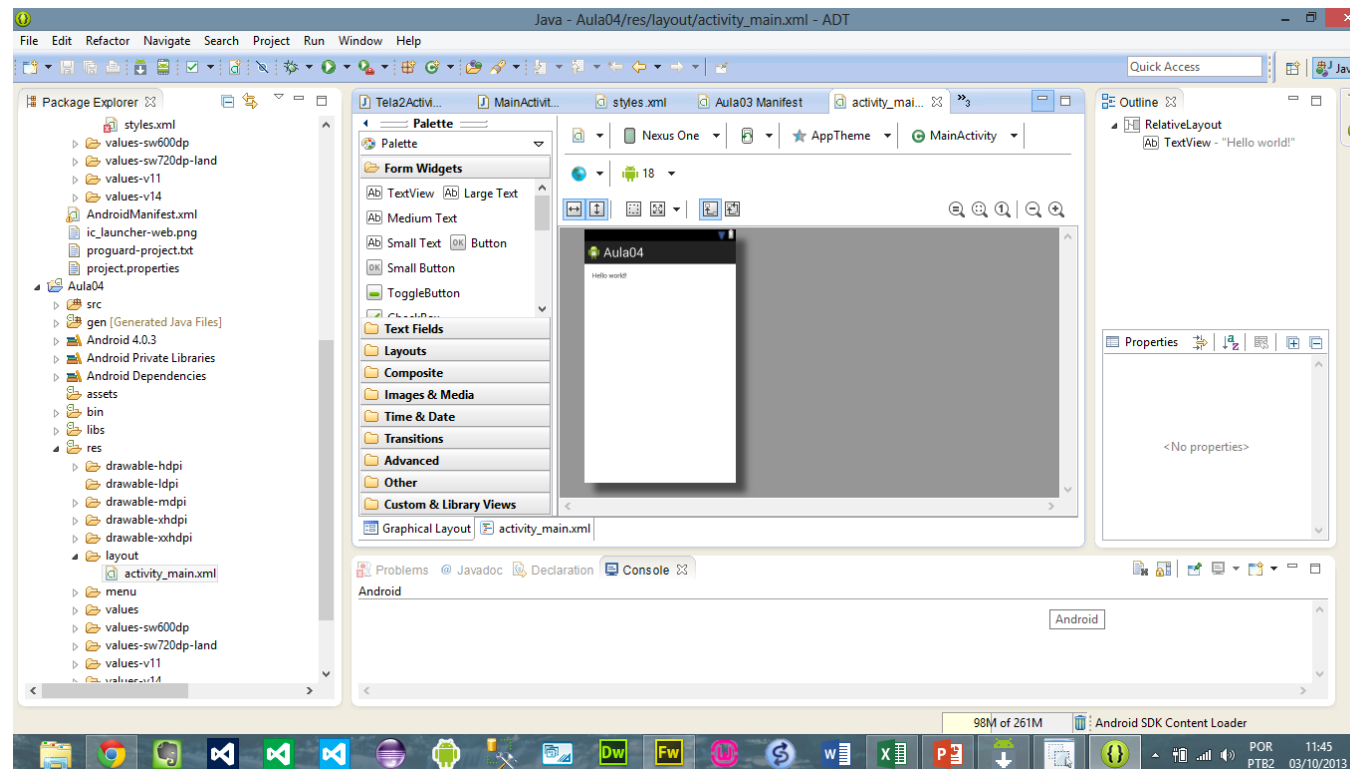
- ▶ Nesta janela, deve ser informado:
 - ▶ Nome da aplicação(Iniciando em maiusculo)
 - ▶ Nome do projeto
 - ▶ Nome do pacote(em geral no formato br.com.seudominio.nomeDaAplicacao)
- ▶ Em seguida configure as seguintes definições:
 - ▶ SDK Mínimo
 - ▶ Versão do android (Target SDK)
 - ▶ Versão da compilação (Compile With)
- ▶ Clique em FINISH para iniciar as definições de configuração da aplicação.
- ▶ **OBS:** Durante esta etapa será necessário criar uma Activity (TELA). Configure de acordo com o especificado.



Estrutura de um Projeto Android

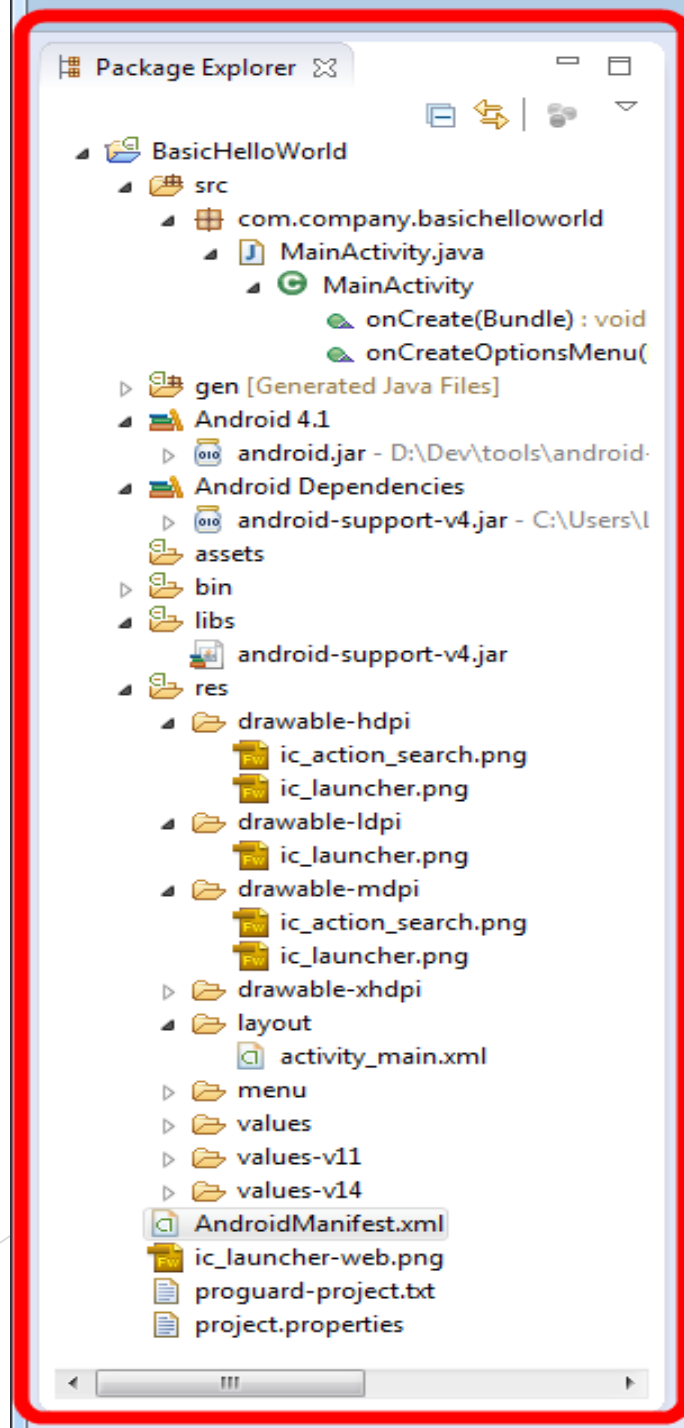
Estrutura de um projeto para Android

- ▶ Após a criação de um novo projeto Android, será carregado uma janela semelhante a abaixo:



Estrutura de um projeto para Android

- ▶ Quando criamos um novo projeto Android, temos uma estrutura de pastas para aproveitar o uso de recursos facilmente para a aplicação.
- ▶ Dentro de Eclipse na perspectiva Java, você deve ver seu projeto apresentado no painel Package Explorer no lado esquerdo da tela.
- ▶ Uma série de pastas são criadas automaticamente para dar funcionalidade a aplicação. Cada diretório tem um significado específico.
- ▶ Recomendamos não APAGAR as pastas e arquivos criados, em especial a pasta (gen), que é atualizado automaticamente a cada novo recurso utilizado no aplicativo.



Estrutura de um projeto para Android

► O diretório /src:

- Contém os arquivos de origem Java associados ao seu projeto. Por exemplo, a classe Activity chama a MainActivity.java que é armazenado neste diretório com o nome do pacote especificado no assistente de projeto Android. Esta classe MainActivity fornece todo o código da aplicação associados com o aplicativo criado.

► O diretório /gen:

- A pasta /gen contém os arquivos de origem Java e outros arquivos de código gerado pelo Eclipse, que estão associados com o seu projeto. Não edite estes arquivos diretamente. Por exemplo, o arquivo R.java é um arquivo gerado para vincular seus arquivos de recurso (como definido na estrutura do diretório /res) para uso em seus arquivos /src Java.
- Estes arquivos são recriados sempre que você adicionar recursos ao seu projeto ou recompilar seu projeto.

Estrutura de um projeto para Android

► O diretório /bin:

- A pasta /bin contém os arquivos de aplicativos resultantes de pacotes associados com o seu projeto uma vez que foi construído. Arquivos do pacote, ou apks, são o produto que você realmente instala em um dispositivo Android.

► O diretório /res:

- A pasta /res contém os arquivos de recursos associados ao seu projeto. Todos os gráficos, valores, layouts e outros arquivos de recursos são armazenados na hierarquia de arquivo de recurso no diretório /res.
- Diferentes tipos de recursos são armazenados em diretórios diferentes. Por exemplo, os gráficos são armazenados sob a tag diretório /drawable, enquanto valores e outras primitivas são armazenados sob a tag diretório /values. Recursos de interface do usuário são armazenados no diretório /layout. Tags especiais muitas vezes incluem alguma nomenclatura para organizar os recursos ainda por tipo de tela, versão do Android, e outros detalhes do dispositivo.

Estrutura de um projeto para Android

► **AndroidManifest.xml:**

- O arquivo AndroidManifest.xml é um arquivo de configuração muito importante que seu aplicativo é obrigatório.
- Este arquivo é usado para especificar tudo sobre a sua aplicação a partir do nome do aplicativo e um ícone para os arquivos de Atividade de classe utilizados e as permissões que seu aplicativo precisa para funcionar, entre muitos outros detalhes.

► **Proguard-PROJECT.TXT:**

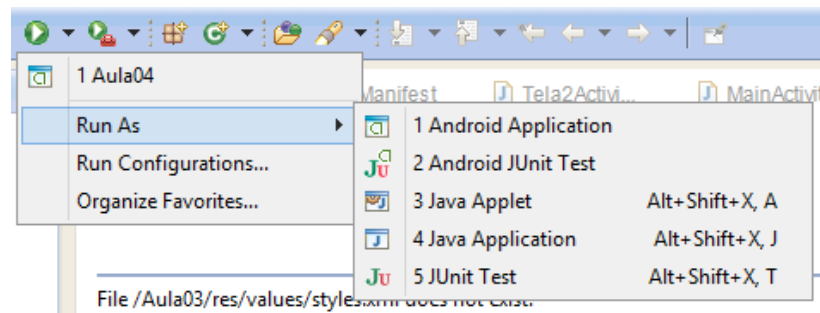
- O arquivo proguard-PROJECT.TXT é gerado pelo assistente de projeto Android. Este arquivo é usado para configurar as definições Proguard associados ao seu projeto. ProGuard é uma ferramenta que pode ser usada para ajudar a proteger o código de pirataria de software usando ofuscação e otimização de código.

► **project.properties:**

- O arquivo project.properties é gerado pelo assistente de projeto Android. Este arquivo é usado para configurar as definições do projeto Eclipse. Desenvolvedores raramente, ou nunca, precisam editar este arquivo diretamente. Em vez disso, clique direito sobre o projeto no Eclipse, escolha propriedades, e faça as alterações necessárias usando a interface do usuário do Eclipse.

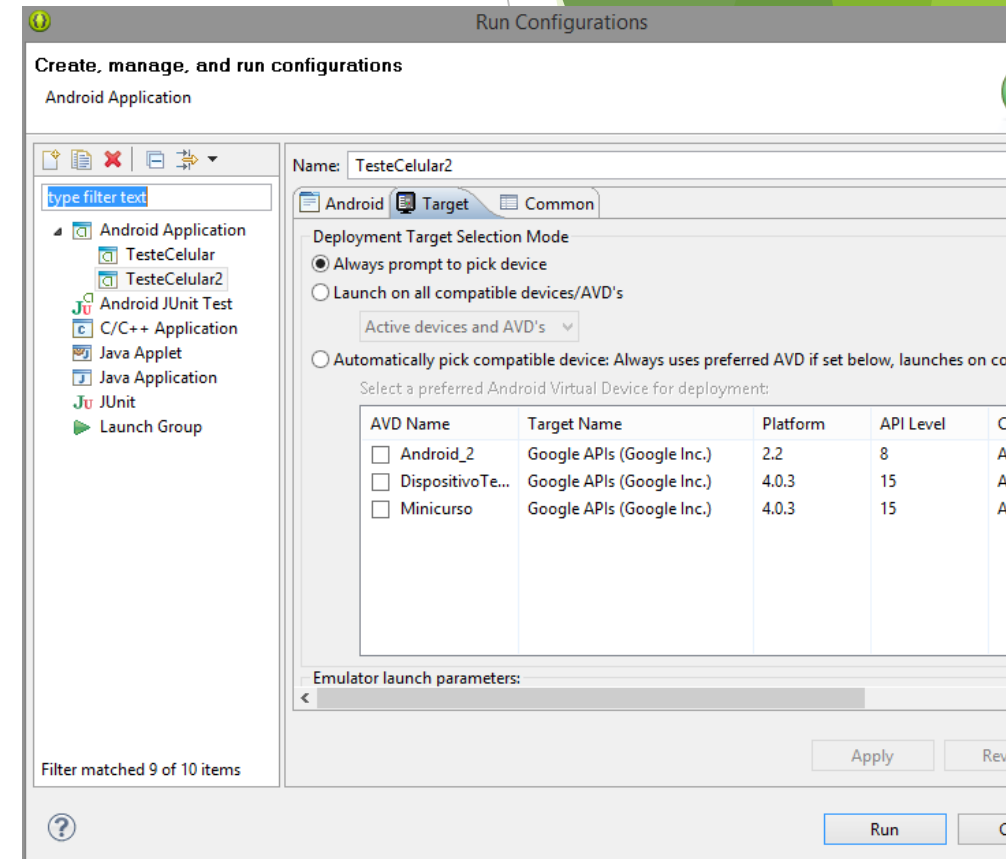
Executando o projeto

- ▶ Um projeto Android pode ser executado sob 2 formas:
 - ▶ Utilizando um dispositivo físico
 - ▶ Utilizando um AVD
- ▶ Em caso de dispositivo físico, o mesmo deve estar conectado em uma porta USB e ter seus drives configurados. Ao executar o aplicativo, o Eclipse reconhecerá o dispositivo e executará.
- ▶ Em caso de AVD, recomendamos antes de rodar a aplicação que o AVD seja iniciado através do AVD Manager, pois este processo é lento. **OBS: UMA VEZ INICIALIZADO, NÃO FECHÉ O AVD.**



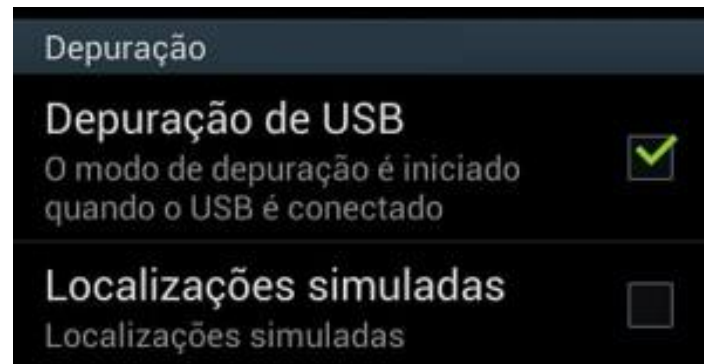
Executando o projeto - Dispositivo virtual

- ▶ Recomendamos que em caso de execução via AVD o dispositivo seja iniciado (**SDK Manager -> Tools -> Manager AVDs -> Start**) antes de rodar aplicação devido ao tempo de inicialização. Uma vez iniciado é recomendável fechar o dispositivo somente quando os testes forem encerrados.
- ▶ O Eclipse automaticamente instalará e executará o APP no dispositivo virtual.
- ▶ OBS: Através da opção RUN As -> RUN Configuration, podemos definir como o Eclipse executará o aplicativo, onde podemos definir 3 opções:
 - ▶ Sempre mostrar uma janela para escolher o dispositivo que receberá o APP
 - ▶ Definir se o APP será executado por um dispositivo físico ou virtual
 - ▶ Definir um dispositivo virtual padrão



Executando o projeto - Dispositivo físico

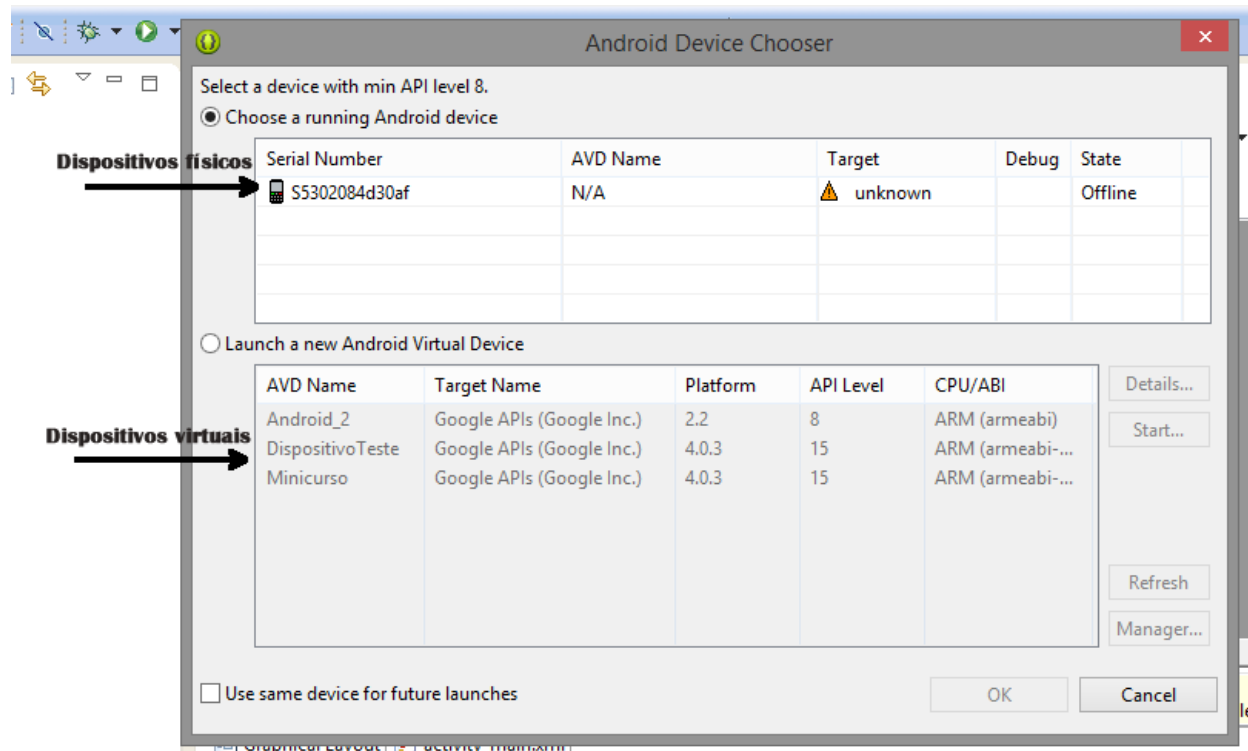
- ▶ **1º Passo:** Verificar na Configuração do aparelho se o modo Depuração USB está ativo, caso não esteja ative-o.



- ▶ **2º Passo:** Conectar o seu celular/Tablet no computador via USB.
- ▶ **3º Passo:** Rodar sua aplicação (Run) e verificar se o seu dispositivo foi detectado. Caso não tenha sido detectado, geralmente é necessário instalar os drivers do aparelho.

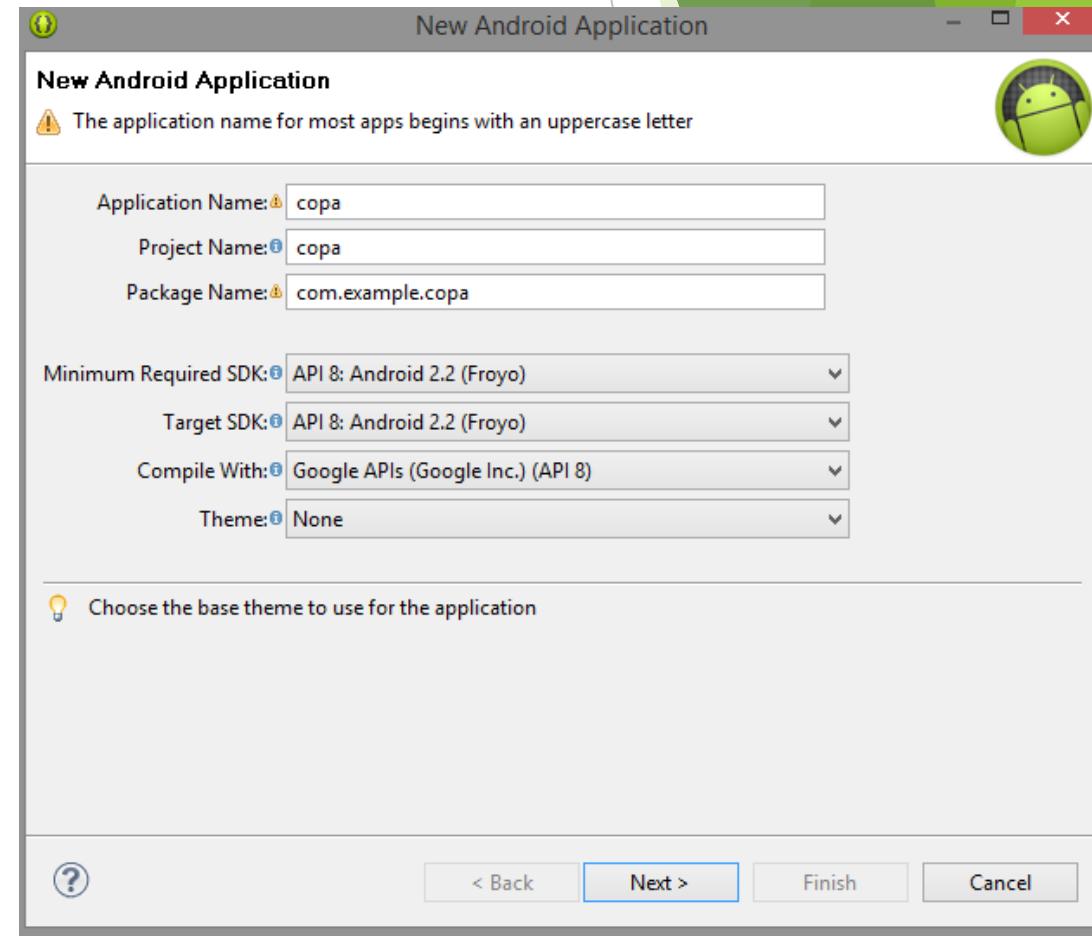
Executando o projeto - Dispositivo físico

- Janela de escolha do dispositivo a qual será executado a aplicação Android



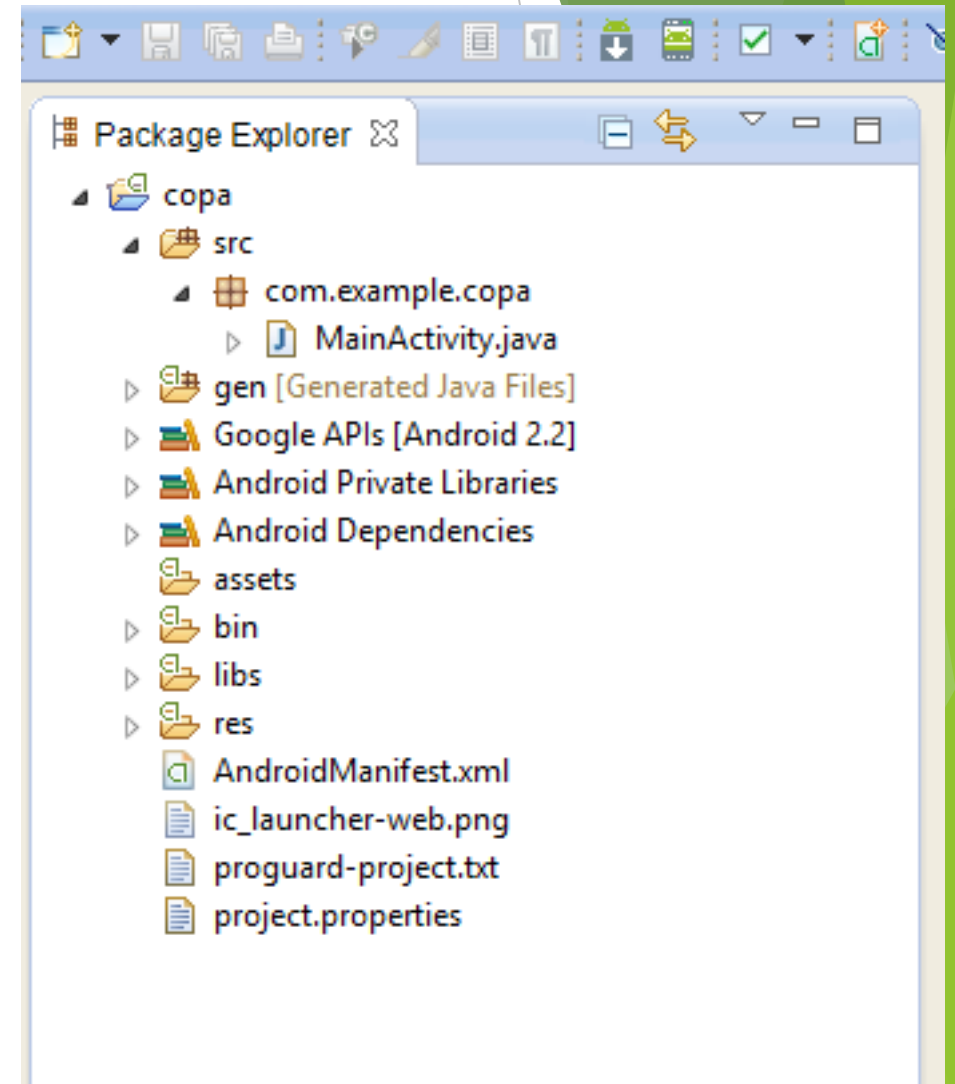
Exemplo 01 - Criando e executando a primeira aplicação

- ▶ Para criar um projeto no Eclipse para Android, devemos escolher menu FILE a opção NEW -> ANDROID APPLICATION PROJECT.
- ▶ Em seguida será carregada uma janela para definirmos o nome da aplicação, nome do projeto, SDK, etc. O preenchimento dos dados é bastante intuitivo, caso algum dado incorreto a ferramenta mostrará um alerta informando que será necessário corrigi-lo.



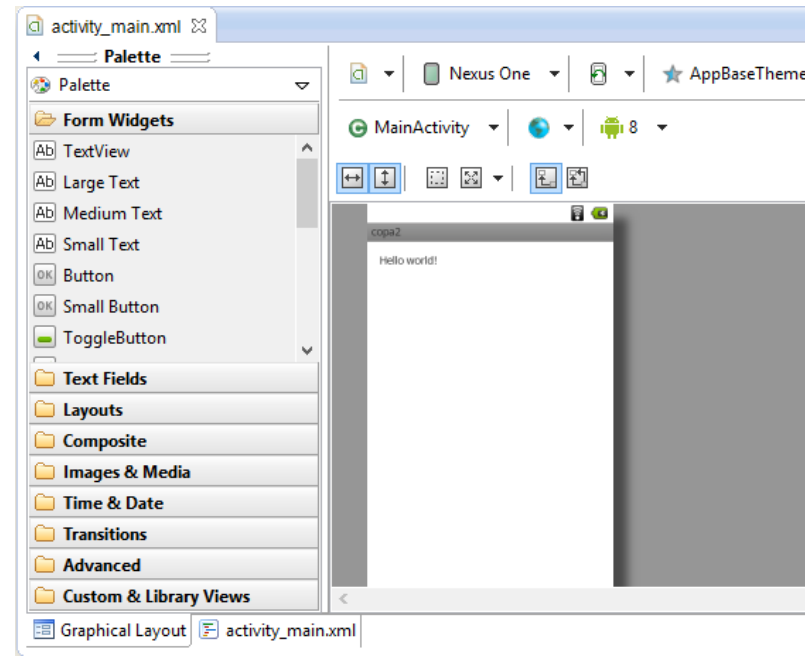
Exemplo 01 - Criando e executando a primeira aplicação

- ▶ Na janela seguinte clicamos em NEXT e em seguida devemos escolher um ícone para nossa aplicação. As janelas seguintes podemos realizar algumas configurações como nome da ACTIVITY(veremos este conceito nas próximas aulas). Neste momento clique em NEXT até o final deste processo.
- ▶ Ao final deste processo será criado uma árvore de diretório de nossa aplicação ANDROID (Package Explorer).
OBS: Nenhum dos diretórios ou arquivos devem ser apagados.
- ▶ Neste primeiro momento será apresentado o diretório **RES** -> **layout**. Este diretório contém o layout de nossa aplicação.

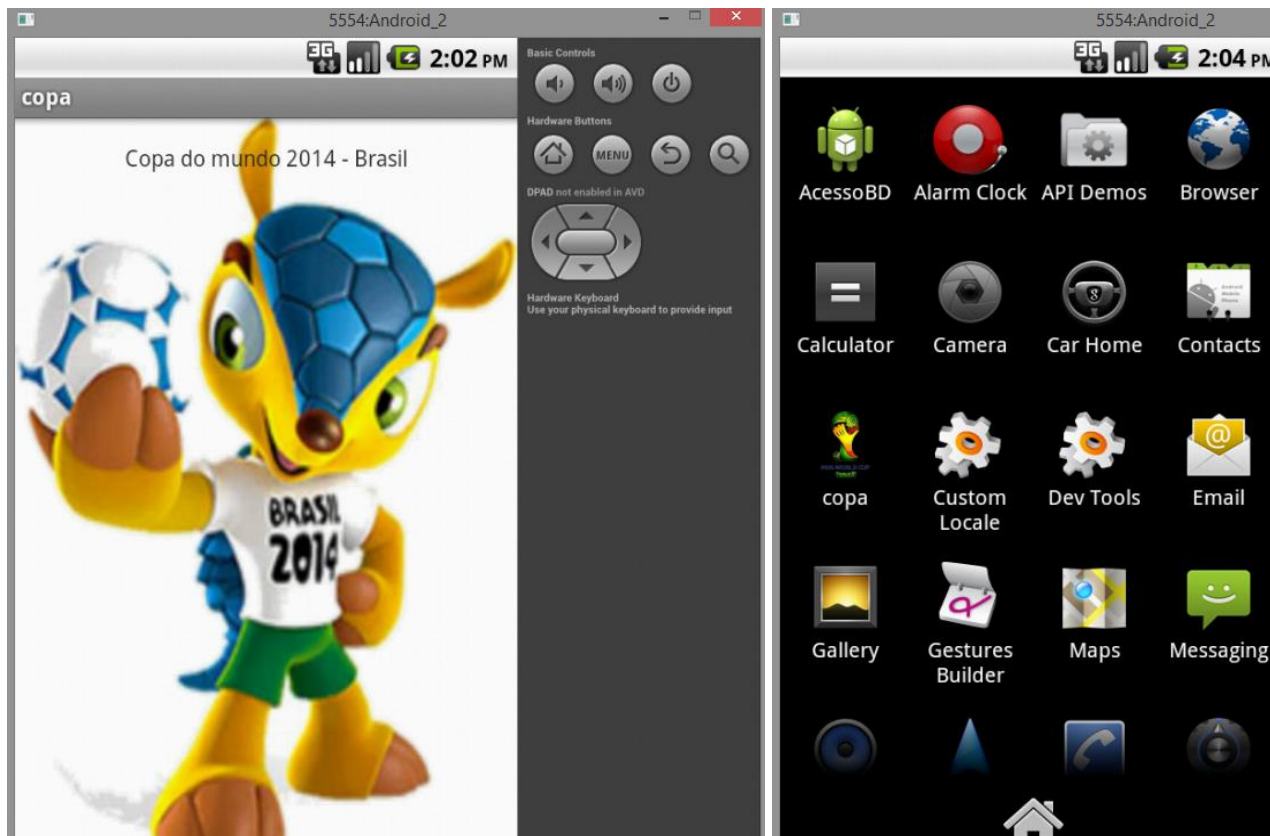


Exemplo 01 - Criando e executando a primeira aplicação

- ▶ Ao expandirmos o **diretório layout** -> **activity_main.xml** será apresentando o layout do aplicativo. Suas configurações podem ser realizados de modo Graphical ou via código XML. Para esta primeira aplicação definiremos um texto para TextView HelloWorld e alteraremos o background do layout.



Exemplo 01 - Criando e executando a primeira aplicação



Na próxima aula...

- ▶ Manipulando layouts: RelativeLayout, LinearLayout e TableLayout
- ▶ Widgets: TextView, ImageView, EditText, Button

