Dispositivos Móveis

Aula 02 - Ambiente de desenvolvimento

Apresentação

Olá, pessoal! O objetivo de nossa segunda aula é demonstrar o download, instalação e configuração do principal ambiente de desenvolvimento utilizado para desenvolver aplicações Android: o Android Studio, lançado em versão estável em dezembro de 2014 e considerado como IDE padrão desde então. Com ele é possível desenvolver aplicações no Linux, no MacOS e no Windows. Nesta aula e ao longo do curso utilizamos o Windows como exemplo.

Estudaremos também, nesta aula, a utilização do emulador Android. Ele permite executar no próprio computador sua aplicação, através de um dispositivo móvel simulado. Veremos as capacidades desse dispositivo e como ele pode se assimilar bastante a um dispositivo real. Por fim, estudaremos a estrutura de um projeto Android e como criar um novo.

Objetivos

Ao final desta aula, você deverá ser capaz de:

- Configurar o ambiente de desenvolvimento de aplicações Android.
- Entender o funcionamento do emulador Android.
- Testar no emulador as aplicações desenvolvidas.
- Entender a estrutura de um projeto Android e como criá-lo.

Instalando o ambiente de desenvolvimento

Como estudamos na aula passada, aplicações Android são desenvolvidas em Java. Como qualquer aplicação Java, aplicações Android podem ser desenvolvidas utilizando qualquer IDE (Integrated Development Environment, ou, em português, ambiente de desenvolvimento integrado) que suporte Java. Apesar disso, a Open Handset Alliance e o Google disponibilizam dois ambientes de desenvolvimento pré-configurados: o Eclipse ADT e o Android Studio, baseados no IntelliJ IDEA.

Tanto o Eclipse ADT quanto o Android Studio têm características específicas, como mostrado na tabela a seguir. Apesar do Eclipse ser um dos ambientes para desenvolvimento Java mais populares, desde o lançamento oficial do Android Studio passou-se a recomendar a utilização dele como IDE oficial de desenvolvimento. Isso se deu principalmente devido ao fato de toda a interface do Android Studio ter sido desenvolvida voltada para o desenvolvimento de aplicações Android. Isso facilita a realização de diversas tarefas, como veremos ao longo do curso. Por esse motivo, nesta aula e durante o resto do curso, utilizaremos o Android Studio como ferramenta de desenvolvimento e, sempre que necessário, faremos referências à sua utilização e funcionamento. O Android Studio é uma IDE multiplataforma, porém durante as nossas aulas utilizaremos o Windows como sistema operacional.

Feature	**Android Studio**	**ADT**
Sistema de construção	Gradle	Ant
Maven-based build dependencies	Sim	Não
Construção de variantes e geração de múltiplos APK (ótimo para o Android Wear)	Sim	Não
Autocompletação avançada e refatoração de código Android	Sim	Não

Feature	Android Studio	ADT
Editor gráfico de layout	Sim	Sim
Assinatura de APK e gerenciamento de keystore	Sim	Sim
Suporte NDK	Sim	Sim

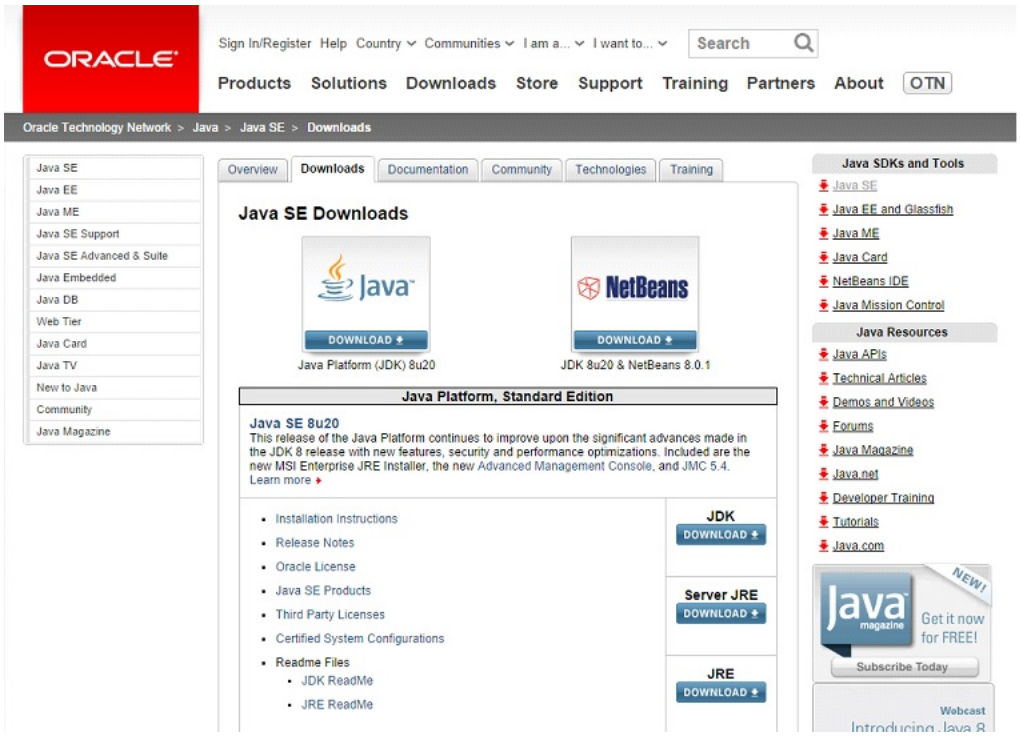
Tabela 1 - Comparação entre o Android Studio e o Eclipse com ADT

Se você desejar utilizar algum outro ambiente de desenvolvimento ou outro sistema operacional, poderá obter mais informações sobre as versões disponíveis, instalação e uso em http://developer.android.com/sdk

Java Development Kit

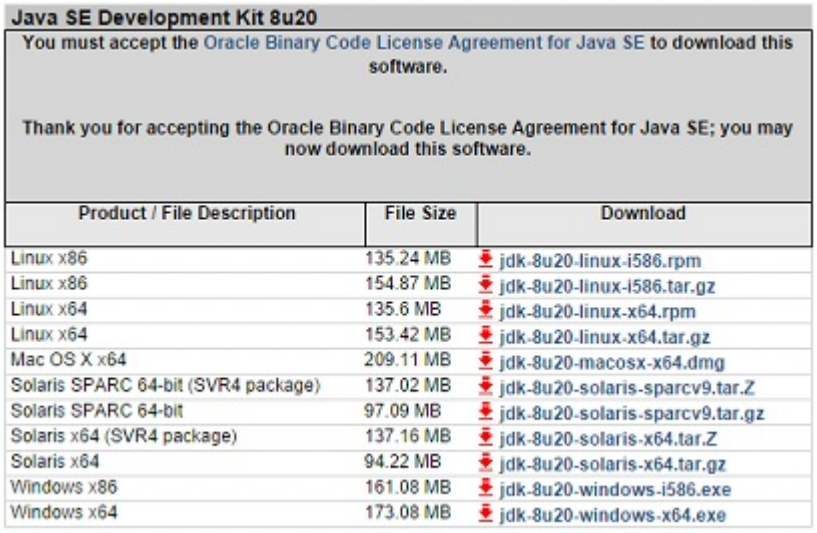
O primeiro componente que precisamos instalar na máquina para o desenvolvimento de aplicações em Java é o Java Development Kit, ou JDK. Esse kit está disponível para download gratuitamente no site da Oracle http://www.oracle.com/technetwork/java/javase/downloads/index.html, como mostra a **Figura 1**. Após aceitar a licença, escolha a versão do seu sistema operacional e faça o download do JDK (**Figura 2**). Quando terminar o download, execute o arquivo e siga os passos para concluir a instalação. Com o JDK instalado, temos a ferramenta Java que precisamos para prosseguir com os outros componentes

Figura 01 - Download do Java SDK.



Fonte: http://www.oracle.com/technetwork/java/javase/downloads/index.html.

Figura 02 - Download do Java SDK.



Java SE Development Kit 8u20

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.

Product / File Description	File Size	Download
Linux x86	135.24 MB	jdk-8u20-linux-i586.rpm
Linux x86	154.87 MB	jdk-8u20-linux-i586.tar.gz
Linux x64	135.6 MB	jdk-8u20-linux-x64.rpm
Linux x64	153.42 MB	jdk-8u20-linux-x64.tar.gz
Mac OS X x64	209.11 MB	jdk-8u20-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	137.02 MB	jdk-8u20-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	97.09 MB	jdk-8u20-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	137.16 MB	jdk-8u20-solaris-x64.tar.Z
Solaris x64	94.22 MB	jdk-8u20-solaris-x64.tar.gz
Windows x86	161.08 MB	jdk-8u20-windows-i586.exe
Windows x64	173.08 MB	jdk-8u20-windows-x64.exe

Fonte: http://www.oracle.com/technetwork/java/javase/downloads/index.html.

É importante ressaltar que algumas máquinas já possuem o JDK instalado, já que esse é um componente do desenvolvimento em Java e não do ambiente Android em si. Caso você já desenvolva em Java, não é necessário reinstalar esse componente, podendo seguir para o próximo passo. É importante notar, no entanto, que o Android Studio requer o JDK versão 7 ou superior para funcionar.

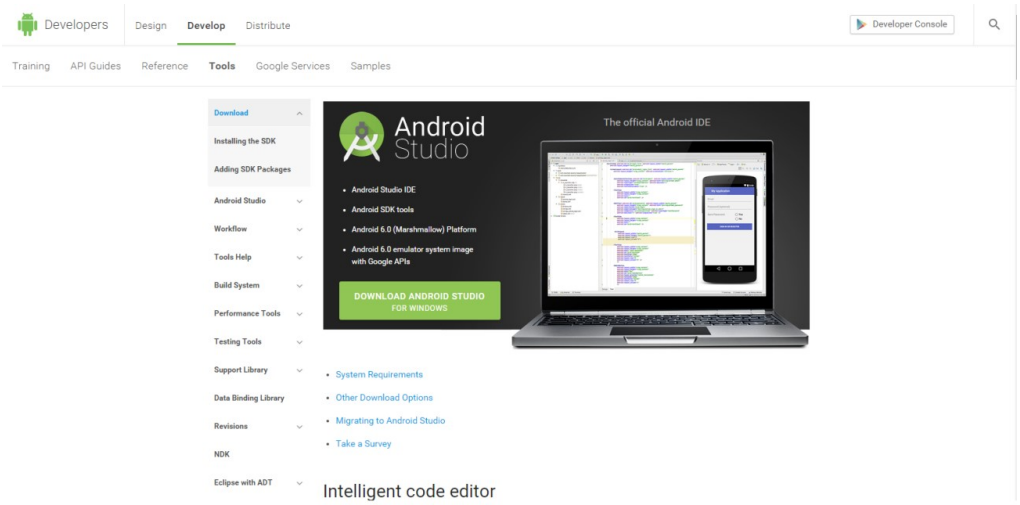
Se você está aproveitando uma instalação já existente, lembre-se de verificar se a versão é adequada!

Android Studio

As vantagens que o Android Studio trouxe ao desenvolvimento de aplicações Android já começam na instalação. Anteriormente, era necessário instalar e configurar o Eclipse, o Android SDK e o Android ADT separadamente para que fosse possível desenvolver. Agora, ao acessar o site do Android para o download do Android Studio já baixamos um pacote completo, o qual depende apenas da instalação prévia do JDK que vimos anteriormente.

Para obter o Android Studio, devemos acessar o site oficial da plataforma e fazer o download do executável para a instalação, clicando no botão “Download Android Studio”, visto na **Figura 3**.

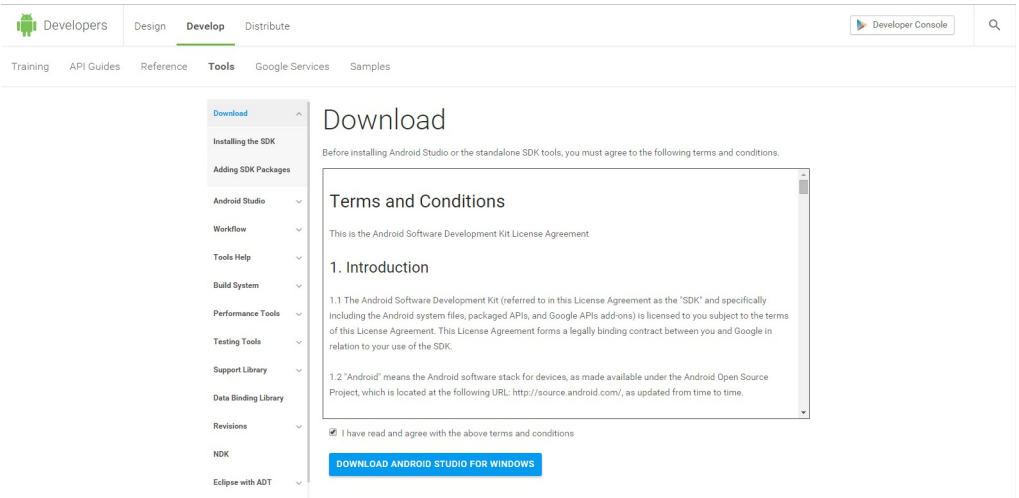
Figura 03 - Página de download do Android Studio



Fonte: http://developer.android.com/sdk/index.html.

Em seguida, deve-se ler e, caso concorde, aceitar os termos e condições para habilitar o botão de download, como indicado na **Figura 4**.

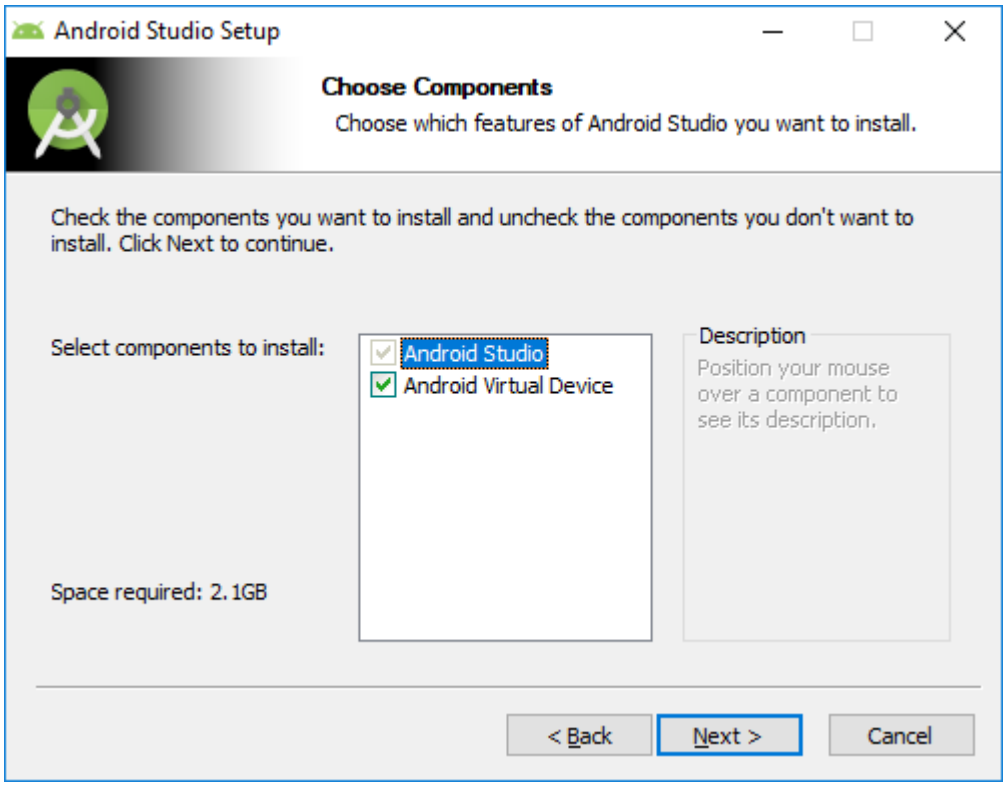
Figura 04 - Termos e Condições de uso do Android Studio.



Fonte: http://developer.android.com/sdk/index.html.

Uma vez concluído o download do arquivo, execute-o e siga as instruções de instalação para instalar o Android Studio. Para isso, basta marcar as opções que aparecerem quando solicitado, como exibido na **Figura 5**. Há um passo a passo disponível no site https://developer.android.com/studio/install.html! Atente apenas para o caminho do Android Studio que utilizaremos!

Figura 05 - Instalação do Android Studio, incluindo o Android SDK e o AVD.



Nesse curso utilizamos o caminho “C:\android\Android Studio” para a instalação do Android Studio. Você pode escolher qualquer pasta no seu computador para isso, mas recomendamos replicar essa escolha para facilitar o desenvolvimento do curso!

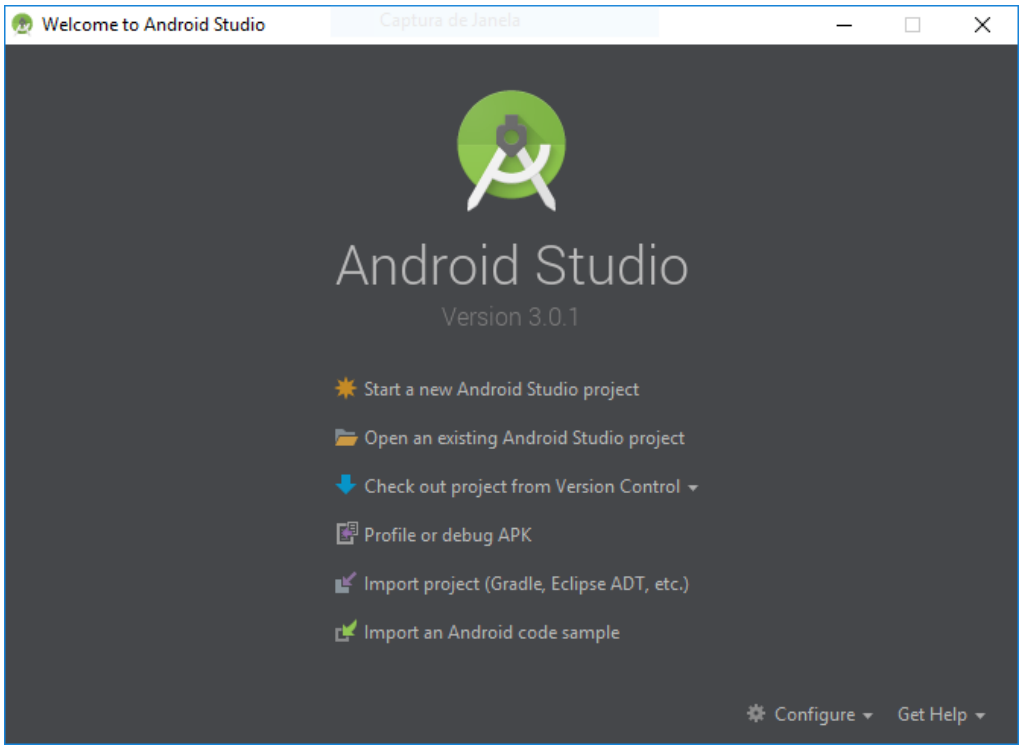
Um problema que pode surgir durante a instalação é a não detecção do JDK por parte do instalador. Se isso acontecer, certifique-se que a instalação do JDK foi feita adequadamente e verifique as variáveis de ambiente do sistema para ver se a variável JAVA_HOME foi criada adequadamente.

Concluída a instalação do pacote, precisamos configurar o Android SDK, que não vem completo, pois existem diversos alvos de desenvolvimento diferentes (versões do Android diferentes, APIs diferentes, etc.), e trazer todos geraria um arquivo enorme e ocuparia bastante espaço na máquina. Vamos, então, conferir quais devemos instalar para esta disciplina e como fazer isso!

Configurando o ambiente de desenvolvimento

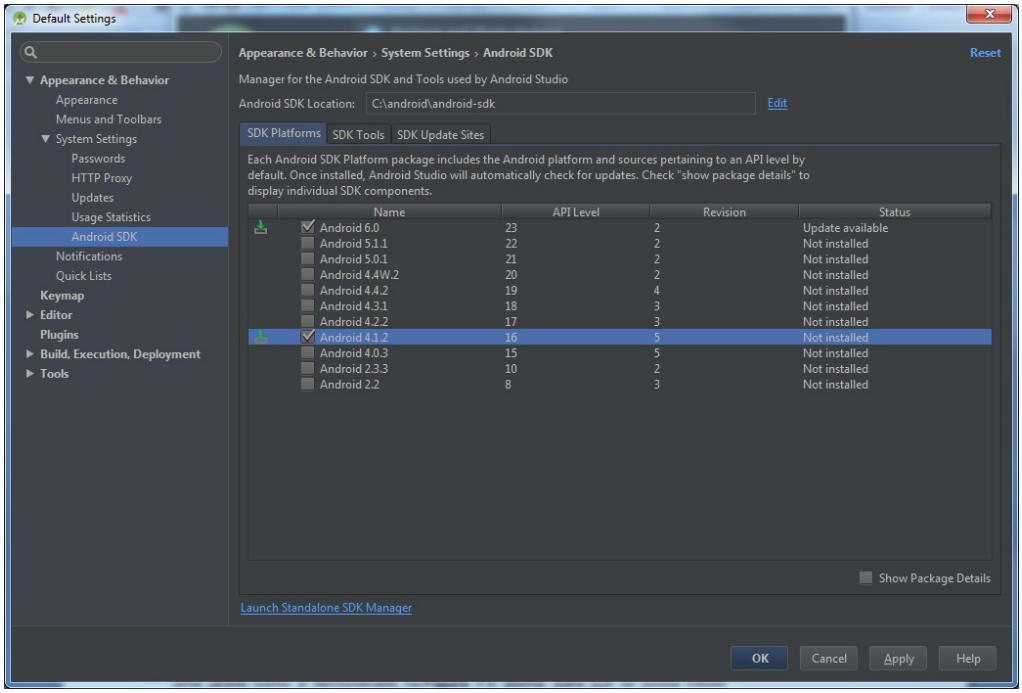
Com o Android Studio devidamente instalado, podemos acessar o Android SDK Manager através dele. Para isso, execute o Android Studio e clique no ícone de Configure -> SDK Manager na inicialização, como visto na **Figura 6**.

Figura 06 - Abrindo o SDK Manager através do Android Studio



Feito isso, podemos selecionar uma ou mais versões do Android com as quais gostaríamos de trabalhar. Ao longo deste curso, utilizaremos a API 16 para desenvolver as nossas aplicações. Sendo assim, instalemos essa versão!

Figura 07 - Janela para instalação dos componentes do Android SDK

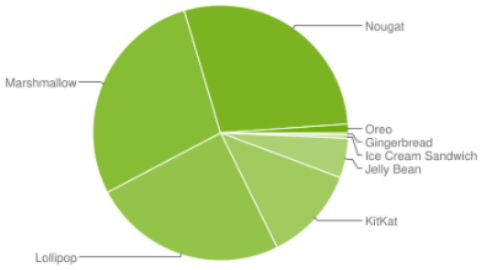


Dentro dessa janela, você deve escolher quais partes do componente serão instaladas no computador. O básico a ser instalado são os componentes SDK Tools, os quais normalmente já estão inclusos ao se instalar o SDK, incluindo o SDK Platform-Tools, que contém algumas das ferramentas que facilitam o desenvolvimento de aplicações Android. Além disso, pelo menos uma versão de um SDK Platform deve ser instalada. Por fim, é necessário instalar pelo menos uma versão do Android SDK Build-tools. Como dito anteriormente, nesta aula e durante o curso, abordaremos a versão 4.1 da plataforma (API 16), então essa versão do SDK Platforms deve ser inclusa.

Se você desejar desenvolver programas para versões mais antigas do Android, instale os respectivos componentes das versões. Atualmente o Android está na versão 8, mas ainda há muitos aparelhos que utilizam versões como a 2, como mostra o gráfico da **Figura 8**. Você pode consultar o uso das versões do Android no próprio site do Android Developer https://developer.android.com/about/dashboards/index.html.

Figura 08 - Gráfico das versões Android com porcentagem de uso

Version	Codename	API	Distribution
2.3.3 - 2.3.7	Gingerbread	10	0.3%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	0.4%
4.1.x	Jelly Bean	16	1.7%
4.2.x		17	2.6%
4.3		18	0.7%
4.4	KitKat	19	12.0%
5.0	Lollipop	21	5.4%
5.1		22	19.2%
6.0	Marshmallow	23	28.1%
7.0	Nougat	24	22.3%
7.1		25	6.2%
8.0	Oreo	26	0.8%
8.1		27	0.3%



Aplicativos desenvolvidos para versões mais antigas irão funcionar normalmente em versões mais novas do Android, contudo, aplicativos desenvolvidos com versões de SDK's mais recentes não terão sua compatibilidade garantida para aparelhos com versões anteriores, podendo não ser executados. Se você deseja criar um aplicativo

que possa ser usado por um grande número de usuários, abrangendo um grande número de diferentes aparelhos e versões, é importante decidir em qual versão o aplicativo será desenvolvido.

Se você deseja testar seu aplicativo desenvolvido em um aparelho real, instale também o “Google USB Driver”, na aba SDK Tools. Você poderá acessar novamente o Android SDK Manager para instalar novos pacotes, sempre que desejar. Outros pacotes terão usos específicos e serão abordados quando necessário.

Com o Android Studio e o SDK instalados adequadamente, podemos carregar o nosso primeiro projeto para, em seguida, conhecer como podemos utilizar um dispositivo virtual para testar nossas aplicações. O Android Studio possui alguns códigos que podem ser importados desde o início para conhecermos algumas funcionalidades básicas do Android sem a necessidade de ter muitos conhecimentos prévios. Perfeito para o momento em que estamos!

Na tela inicial do Android Studio, clique na opção “Import an Android code sample”, como visto na **Figura 6**, à esquerda. Em seguida, selecione a opção Basic Notification. É possível buscar na lista diretamente pelo nome. Uma vez selecionado o projeto, o Android Studio se encarregará de baixar o código completo do exemplo e iniciar o projeto para edição. Com isso, podemos seguir adiante.

Vamos agora conhecer mais uma parte importante do desenvolvimento Android, que é a criação e utilização de um Android Virtual Device, ou AVD. Um AVD é um simulador que emula um dispositivo Android virtual. Nesse dispositivo emulado é possível testar as aplicações que estão sendo desenvolvidas mesmo sem possuir um dispositivo Android físico. Mais adiante, estudaremos as propriedades do emulador. Por enquanto, vamos focar em sua instalação e configuração, que é bem simples.

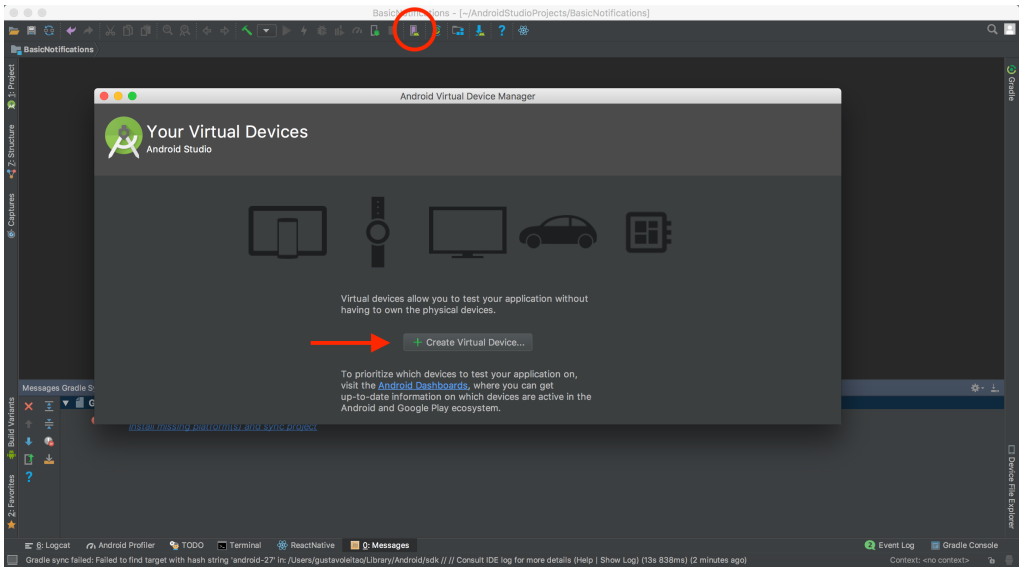
Android Virtual Devices

Os dispositivos virtuais são uma opção para os desenvolvedores de aplicações para Android que não possuem dispositivos físicos Android para testarem suas aplicações. Até mesmo para os que possuem esses dispositivos, os dispositivos virtuais proveem opções para simular chamadas, mensagens ou até mesmo

localizações de GPS, as quais seriam difíceis de simular utilizando um aparelho Android real. Por essas facilidades, os AVD são uma importante opção para o desenvolvedor Android.

Ao instalar uma SDK Platform, torna-se possível, dentro do menu de AVD, criar um dispositivo virtual para aquela versão da plataforma que foi instalada. Para criar novos dispositivos, basta clicar na opção de AVD, destacada na **Figura 9**. Em seguida, a tela para gerenciar os dispositivos virtuais vai aparecer, como vemos na mesma imagem.

Figura 09 - Janela para gerenciamento dos dispositivos virtuais



Nessa janela, é possível criar e gerenciar os dispositivos virtuais. Clicando no botão Create Virtual Device, abre-se a tela para configuração do novo AVD.

O Android Studio, por padrão, traz uma grande quantidade de configurações que representam dispositivos reais, adequadas para a simulação no próprio computador. Além disso, no menu da esquerda, pode-se escolher entre outros tipos de dispositivos além de smartphones, como tablets, relógios e até a Android TV. Essa variedade de dispositivos e tamanhos de tela é importante pois essa é uma das principais vantagens de se utilizar dispositivos virtuais: testar o aplicativo em resoluções e telas diversas sem o gasto de obter múltiplos aparelhos.

Também é possível criar novos dispositivos caso julgue necessário. Isso pode ser feito através do botão New Hardware Profile. Ao clicar nesse botão, você poderá escolher tamanho de tela, resolução, memória, opções de input e câmera, bem

como o tipo de dispositivo. Escolha um nome que descreva bem o novo dispositivo que você criar, caso o faça, para facilitar a identificação deste em meio aos outros no AVD Manager.

Seguindo com o nosso exemplo, utilizaremos um Nexus 5, com a API 16, como indicado anteriormente. Para isso, devemos clicar no botão Create Virtual Device e então selecionar, na parte de Phones, o Nexus 5. Em seguida, clique em Next.

Na tela seguinte serão exibidas opções de versões de plataforma para utilizar no seu dispositivo. Como queremos a API 16, devemos buscar por essa opção na coluna API Level. Caso a imagem ainda não esteja instalada, clique na opção “Download” que aparece ao lado do nome da imagem, aceite a licença e realize a instalação necessária.

Com o modelo selecionado e a versão do Android configurada, a última opção necessária é a escolha do nome do dispositivo. Escolha um nome que será facilmente reconhecido e lembrará você do motivo de ter criado aquele dispositivo. Não há necessidade de modificar, por enquanto, nenhuma das outras opções além do nome. Para finalizar, clique em Finish. Em seguida, o dispositivo aparecerá na lista do AVD Manager.

Tendo o dispositivo criado, basta clicar no botão play (verde) que está listado em Actions. Esse botão iniciará o AVD que você acabou de criar. A tela de inicialização com o nome Android demora um pouco, então tenha paciência! Após ter o sistema inicializado, temos um dispositivo Android pronto para ser utilizado em seu computador.

Com o ambiente instalado e o emulador configurado, podemos então começar a testar as nossas primeiras aplicações Android. Como nada sobre a programação Android foi apresentado ainda, utilizaremos o exemplo que inicializamos anteriormente para ver como o emulador funciona.

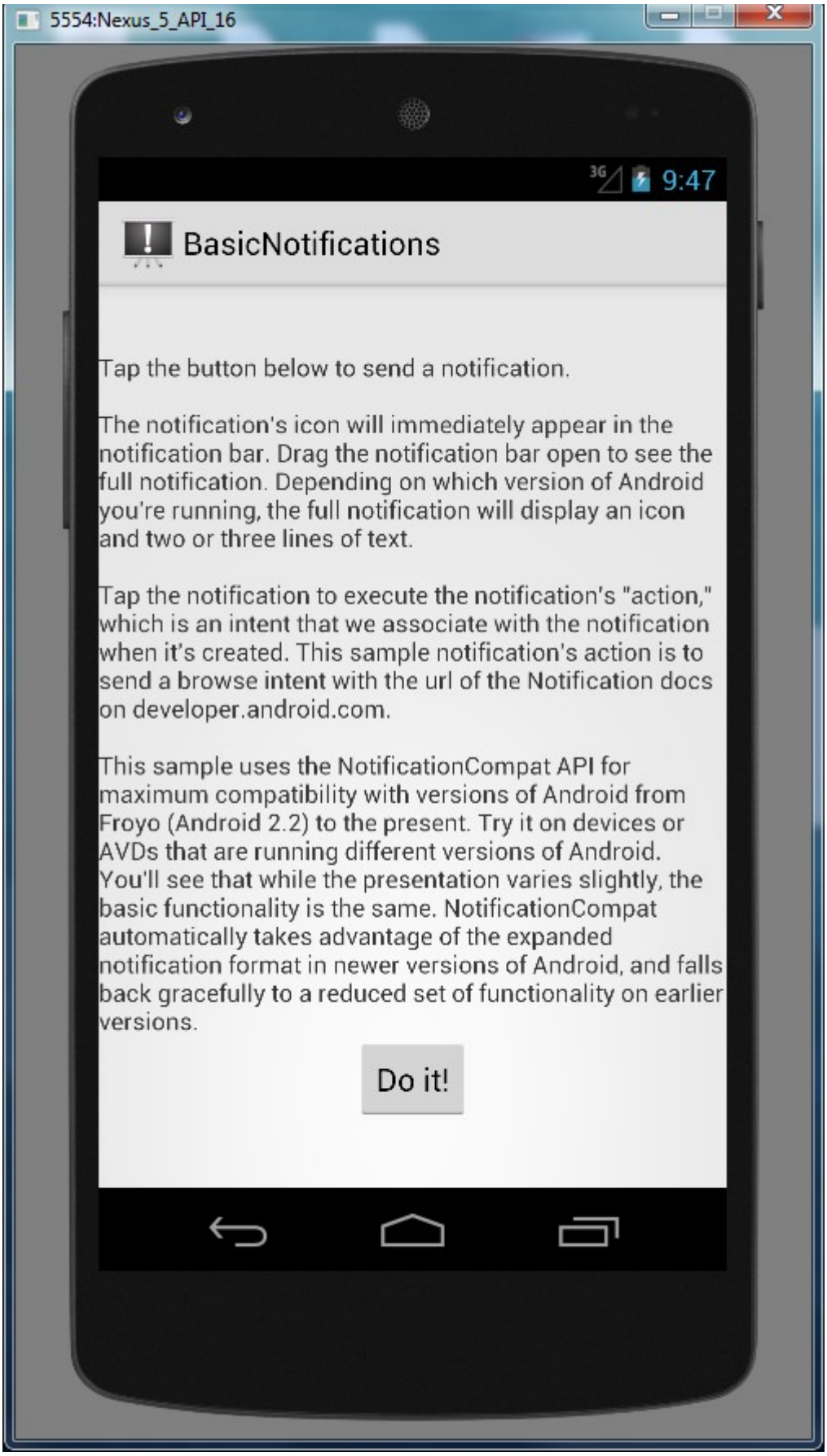
Emulando uma aplicação

Emular uma aplicação desenvolvida no Android Studio em Android Virtual Device é uma tarefa bem simples, facilitada pela própria IDE! Ele facilita esse processo, pois se encarrega de compilar e instalar a aplicação direto no emulador. Com a aplicação instalada, você já está pronto para testá-la. Caso você não tenha um emulador compatível com a aplicação aberto, porém, já tenha um criado no AVD um dispositivo que seja compatível com a aplicação, o Android Studio se encarrega de abrir esse dispositivo e instalar a aplicação, fazendo a maior parte do trabalho para você.

Como não temos ainda uma aplicação desenvolvida para ver como esse processo funciona, utilizaremos uma das aplicações que já são fornecidas junto ao SDK do Android, como vimos anteriormente.

Para instalar uma aplicação no emulador, basta clicar no botão de play (atalho shift+F10). Após clicar nessa opção, uma tela para escolher onde o projeto será instalado surgirá. É possível utilizar um dos AVDs já em execução ou lançar um novo emulador. O importante é que o alvo escolhido seja compatível com a versão da aplicação. Após selecionar o AVD a ser iniciado, é só aguardar o carregamento do mesmo e a instalação da aplicação. A **Figura 10** mostra como o emulador aparecerá em sua tela, após finalizar o carregamento.

Figura 10 - Tela do emulador, após ter sido carregado.



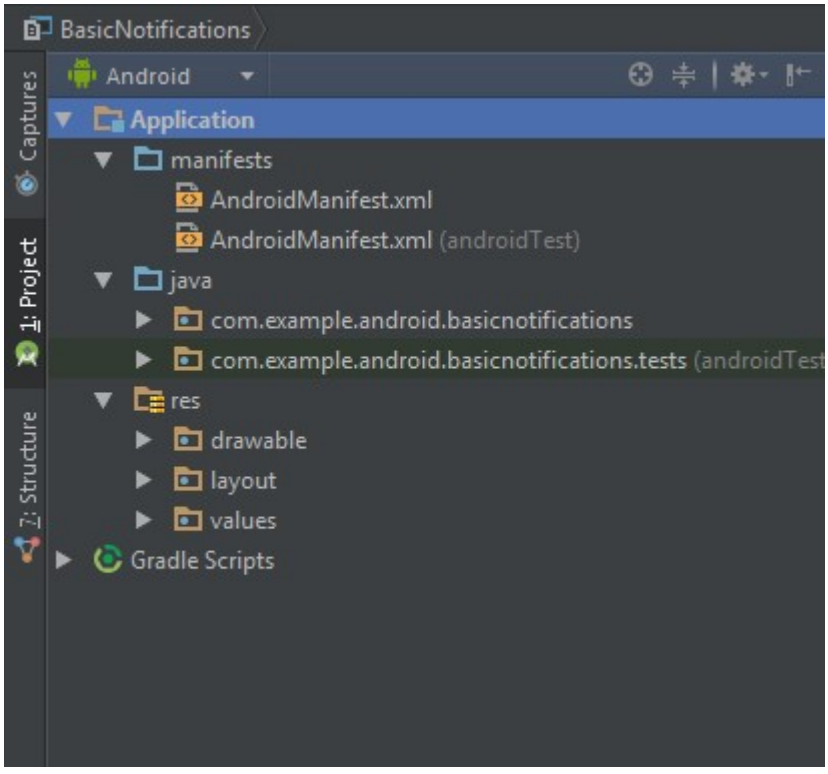
Após destravar a tela, a aplicação executada já estará em primeiro plano e pronta para ser testada. Você deve utilizar o mouse para interagir com o emulador, simulando toques na tela com cliques.

Com isso, temos a IDE configurada e pronta para uso, bem com o conhecimento necessário para testar as nossas aplicações dentro de um emulador. Vamos agora conhecer um pouco mais sobre os projetos Android dentro do Android Studio, conhecendo sua estrutura e também como é feita a criação de um novo projeto.

Estrutura dos projetos Android

Em um projeto Android, temos uma organização padrão de pastas para os arquivos de código, recursos e configurações do aplicativo. Utilizando o projeto de exemplo da sessão anterior, vamos analisar o que são cada uma dessas pastas e onde devemos colocar os arquivos criados. A aba que aparece ao lado esquerdo no Android Studio mostra a organização das pastas para o projeto BasicNotifications, como mostra a **Figura 11**.

Figura 11 - Organização dos arquivos do projeto Basic Notification.



- Pasta java: A pasta *java* é onde colocamos as classes Java da nossa aplicação, contendo nosso código fonte.

- Pasta res: A pasta res (resources) é onde ficam os arquivos de layout, imagens, XML de configuração, XML com strings internacionalizáveis, sons, etc. A pasta res, juntamente com a pasta src, são as pastas mais utilizadas da aplicação. Todos os arquivos que não são arquivos de código Java ficam na pasta res, e podemos separá-los de acordo com as configurações dos dispositivos onde a aplicação será executada. Sendo assim, as classes Java da pasta src fazem uso dos arquivos de recursos da pasta res, gerando uma aplicação mais robusta e que possa ser utilizada em diversos dispositivos com características diferentes. Dentro da pasta res podemos encontrar outras pastas, como drawable, layout, e values. Podemos ter mais ou menos pastas, com ou sem modificadores, mas geralmente essas três pastas estarão presentes.

- A pasta *res/drawable* é onde colocamos as imagens. Notem que há diferentes pastas *drawable* no nosso projeto exemplo. Essas diferentes pastas trazem imagens com diferentes resoluções, sendo usadas para aparelhos com tamanhos de telas diferentes, por exemplo, *drawable-hdpi* (alta resolução), *drawable-land* (formato de tela landscape), etc. Por enquanto, utilizaremos a pasta padrão *drawable* sem nenhum modificador.

- A pasta *res/layout* é onde ficam os arquivos XMLs que formam o layout das telas de nossa aplicação. Também podemos ter modificadores específicos para diferentes telas, mostrando layouts diferentes para cada tipo de tela. Um exemplo útil é utilizar um layout para uma tela de smartphone padrão e outro layout diferente para uma tela de tablet. Por enquanto, também utilizaremos somente a pasta *layout* padrão, sem nenhum modificador.

- A pasta *res/values* contém os arquivos de Strings ou outros valores que podem ser usados na aplicação. Muito útil quando queremos criar uma aplicação que suporte vários idiomas. Para isso, criamos diferentes arquivos com as Strings para diferentes idiomsa que queremos na aplicação.

Veremos mais detalhes sobre o uso de diferentes recursos para aparelhos com características diferentes adiante no curso, na aula sobre recursos

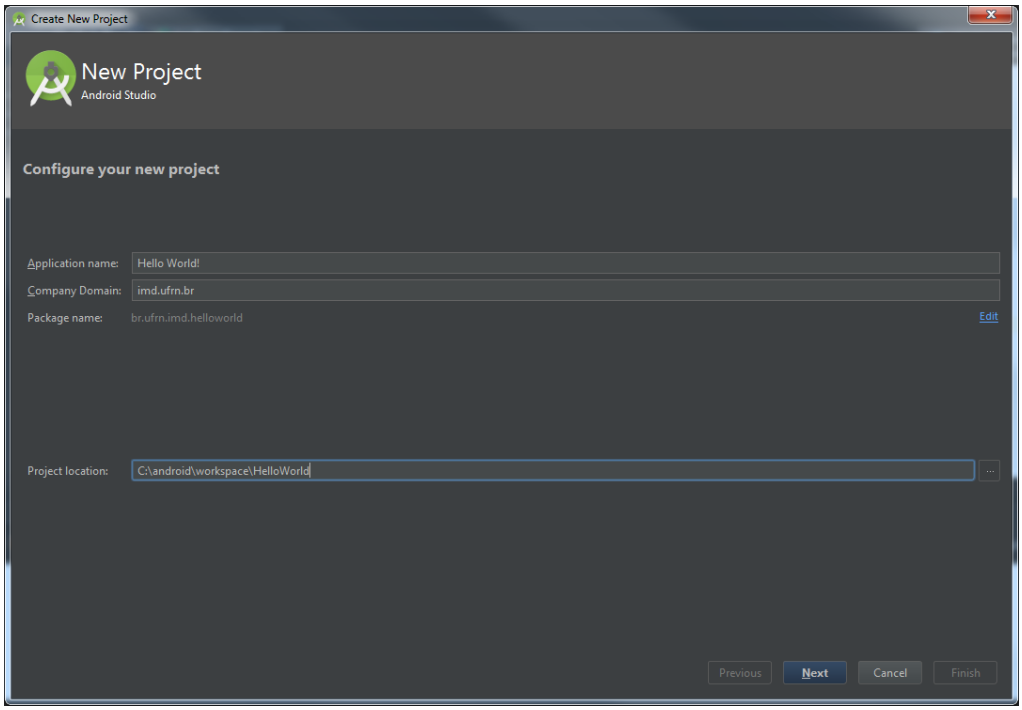
- Pasta manifests, contendo o arquivo AndroidManifest.xml: Este arquivo contém informações sobre nossa aplicação, como versão da aplicação, desenvolvedor, versão mínima do Android para que a aplicação seja executada, permissões necessárias para instalação, telas e serviços disponíveis, etc. São informações essenciais e que devem ser registradas pelo desenvolvedor sempre que necessário.

Criando nosso primeiro projeto Android

Agora que conhecemos o ambiente de desenvolvimento, vamos aprender a criar um projeto Android novo e como definir suas propriedades iniciais. Inicialmente, vá no menu File > New > New Project. Em seguida, será exibida uma tela, como na **Figura 12**, onde você deve informar:

- Application Name: o nome do aplicativo;
- Company Domain: o domínio utilizado para os aplicativos de sua companhia;
- Package Name: o nome do pacote da aplicação. Geralmente utilizamos o site da aplicação ou do desenvolvedor para nomear o pacote. Por exemplo, para uma aplicação chamada MeuAplicativo e que possui o site “www.meuaplicativo.com.br”, podemos nomear o pacote como br.com.meuaplicativo. Note que a convenção é utilizar um endereço web que pertença ao desenvolvedor, de traz para frente;
- Project Location: o local onde o projeto será criado no disco local da máquina. Lá estarão os arquivos que forem sendo criados ao longo do projeto.

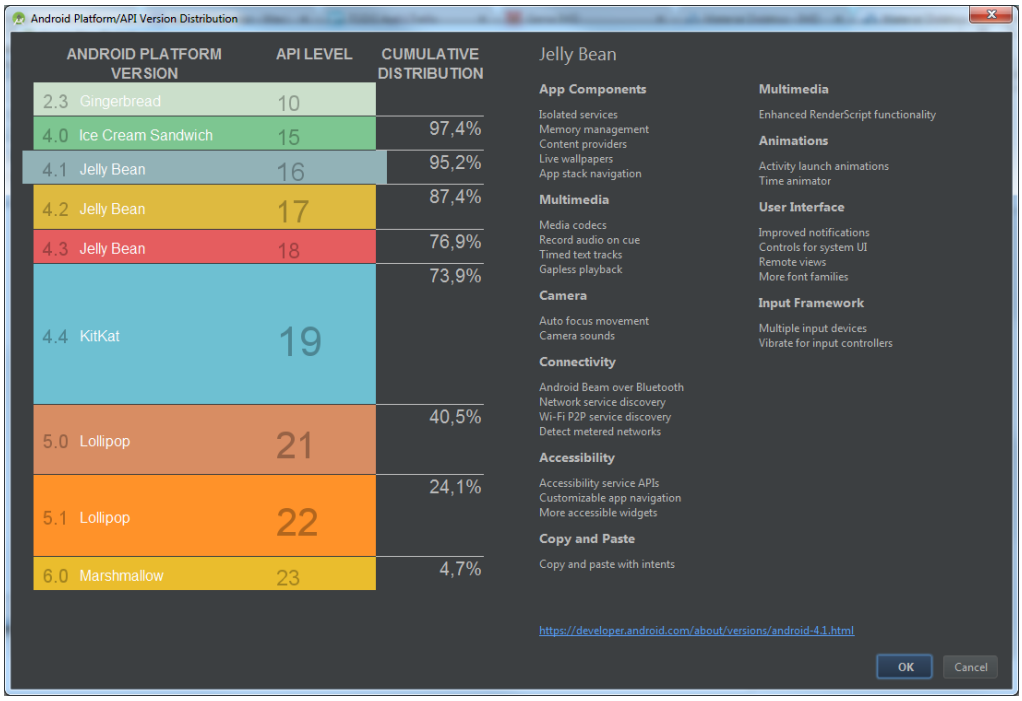
Figura 12 - Propriedades iniciais do projeto Android



Em seguida, ao clicar em next, um novo conjunto de opções será exibido, então você deverá escolher qual o alvo da aplicação. Os alvos possíveis são Telefone e Tablet, Relógios, TVs, Automóveis e Óculos. Ao escolher para qual tipo de dispositivo deseja-se desenvolver, deve-se indicar a versão Android mínima que poderá executar o aplicativo criado.

Perceba que nessa tela apresentada há um link escrito *Help Me Choose*. Ao clicar nesse link, você verá uma tela como a exibida na **Figura 13**, a seguir, que demonstra a porcentagem de usuários do Android que você conseguirá atingir ao desenvolver uma aplicação utilizando aquela versão, bem como as funcionalidades que foram adicionadas na versão. Essa tela é muito interessante para que vocês possam decidir a versão que irão atuar. Também podemos ver nela que, escolhendo a API 16, estaremos atingindo 95,2% dos usuários Android, que é uma quantidade bastante relevante! Por isso manteremos essa como a oficial de nosso curso.

Figura 13 - Distribuição de utilização entre versões do Android.



Fonte: Captura de tela do Android Studio

Ao clicarmos em next, seguiremos para o próximo passo, em que deveremos escolher o tipo de Activity que será a tela inicial da aplicação. Por enquanto, escolhemos Blank Activity e clicamos em next.

A última tela solicita informações relacionadas à criação da Activity que escolhemos como inicial. Insira um nome para esta Activity inicial e finalize a criação do projeto. Seu aplicativo será criado com uma tela inicial com um texto “Hello World” e em seguida você já poderá executar a aplicação no emulador normalmente.

Nas próximas aulas iremos aprender como criar nossos layouts personalizados e integrá-los com o resto da aplicação. Até lá!

Leitura Complementar

ANDROID DEVELOPERS. Managing AVDs with AVD Manager. Disponível em: http://developer.android.com/guide/developing/devices/managing-avds.html. Acesso em: 15 abr. 2015.

Resumo

Nesta aula, você estudou como instalar os componentes necessários para começar a criar aplicações Android, além de aprender a inicializar e utilizar o emulador Android para testar as aplicações desenvolvidas. Vimos também a estrutura de um projeto Android e como criar um novo projeto. Com essas habilidades, é possível começar a desenvolver suas próprias aplicações, ou pelo menos modificar os exemplos que o Android inclui em seu SDK.

Autoavaliação

1. Quais versões do Android são possíveis de executar no emulador?
2. Quais os pontos importantes para se lembrar ao escolher a versão do seu dispositivo virtual?

Referências

ANDROID Developers. 2015. Disponível em: http://www.developers.android.com. Acesso em 15 abr. 2015.

DIMARZIO, J. **Android**: a programmer's guide. São Paulo:McGraw-Hill, 2008. Disponível em: http://books.google.com.br/books?id=hoFI5pxjGesC. Acesso em: 16 abr. 2015.

HASEMAN, C. **Android essentials**. Berkeley, CA. USA: Apress, 2008.

LECHETA, Ricardo R. **Google android**: aprenda a criar aplicaçoes. 2. ed. São Paulo: Novatec, 2010.

_____. **Google android para tablets**. São Paulo: Novatec, 2012.

MEIER, R. **Professional Android 2 application development**. New York: John Wiley & Sons, 2010. Disponível em: http://books.google.com.br/books?id=ZthJlG4o-2wC. Acesso em: 16 abr. 2015.