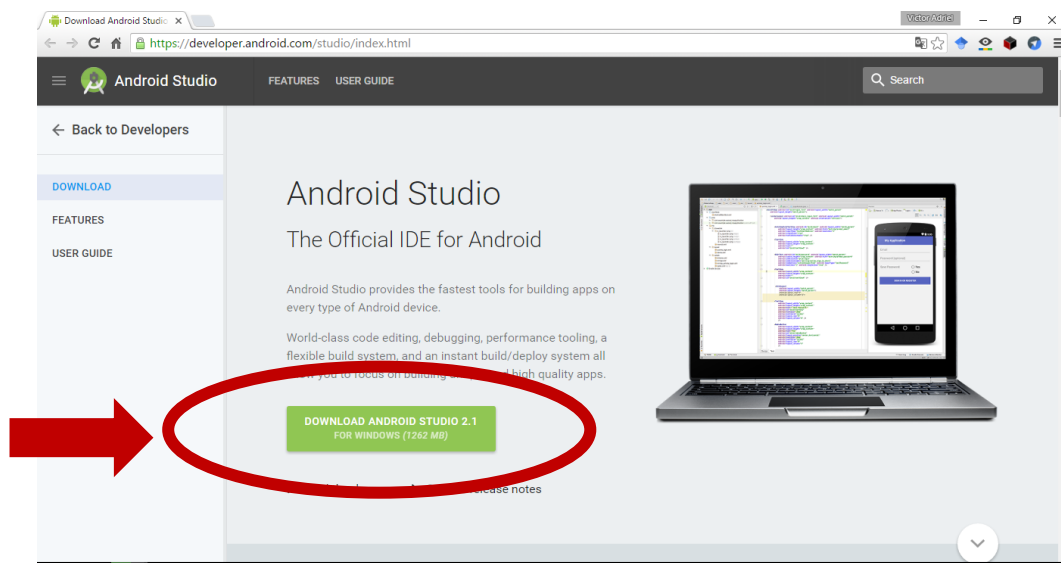


Introdução ao Desenvolvimento de Aplicativos Android

Preparando o ambiente de desenvolvimento

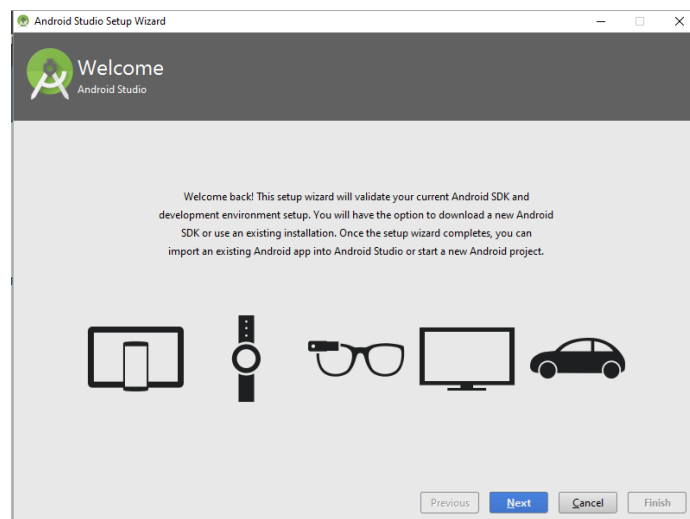
Entre as ferramentas que você precisa instalar estão:

- **JDK** - como o Android faz uso da tecnologia Java, é necessário ter o *Java Development Kit* instalado
- **Android Studio** - é possível adquirir o Android Studio e a SDK com um só *download*. Basta visitar a página <https://developer.android.com/studio/index.html> e clicar no botão indicado:



Caso você ainda não tenha o JDK instalado em seu computador, basta ir à página <http://www.oracle.com/technetwork/java/javase/downloads/index.html>, realizar o download de acordo com o sistema operacional utilizado e realizar a instalação.

Após o *download*, execute o arquivo e siga o assistente de configuração para instalar o Studio Android e quaisquer ferramentas do SDK necessárias.

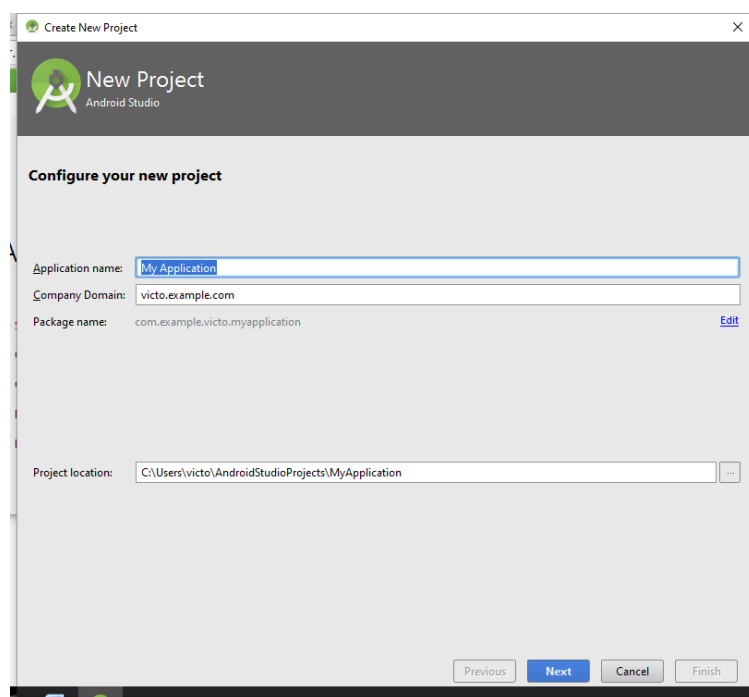


OBS: Em alguns sistemas Windows, o instalador pode não encontrar onde o JDK está instalado. Se você encontrar esse problema, você precisa definir uma variável de ambiente que indica o local correto.

Você deverá clicar em **Start > Computer > System Properties > Advanced System Properties**. Em seguida, abra a aba **Advanced > Environment Variables** e adicione uma nova variável de sistema **JAVA_HOME** que aponta para a pasta do JDK, por exemplo **C:\Program Files\Java\jdk1.8.0_77**.

Novo projeto

Se este é seu primeiro projeto, clique em *New Project* na tela inicial. Se você já tiver um projeto aberto, clique em *File -> New Project -> Android Application Project*. A seguinte tela aparecerá:



Na janela que se abrirá, o primeiro campo indica o nome da aplicação. Este é o nome que irá aparecer no *launcher* do Android, ou seja, aquele que os usuários verão. Preencha com “My First App”. No segundo campo, adicione “example.com”, onde diz language de programação selecionar “Java” e, para os outros campos, utilize os valores padrão e clique em Avançar.

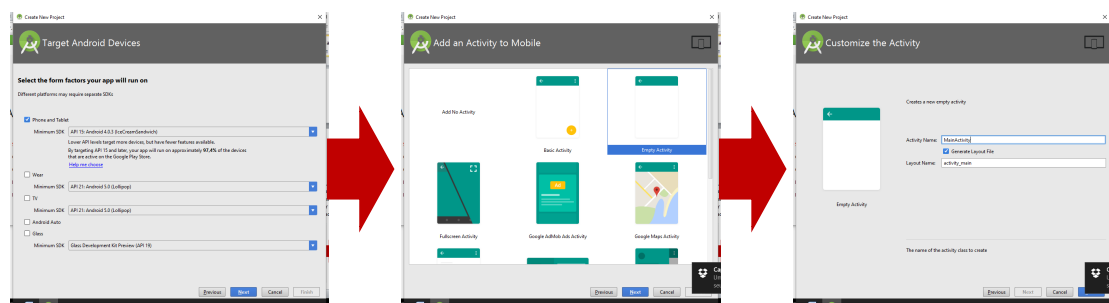
Em seguida você poderá selecionar o *target* do projeto. O campo Target SDK indica o target para o qual a aplicação será compilada.

É indicado utilizar a API com uma distribuição alta. Recomendavel 4.0 Ice Cream Sandwich

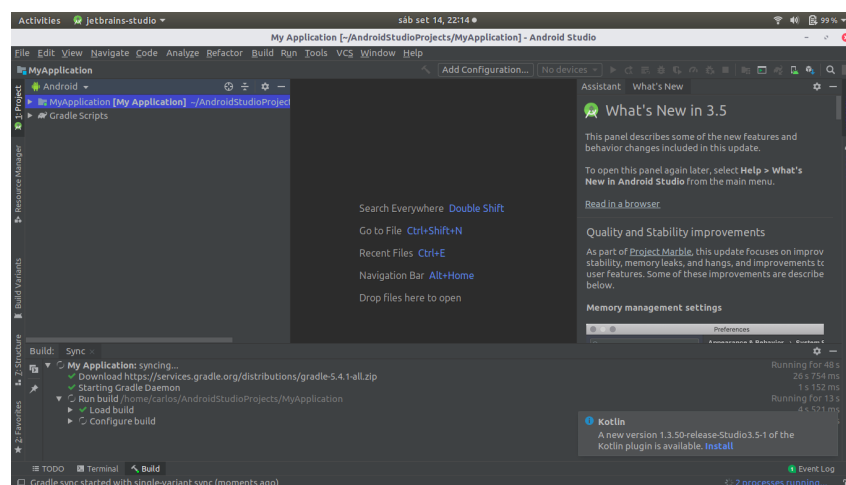
Android Platform/A		
ANDROID PLATFORM VERSION	API LEVEL	CUMULATIVE DISTRIBUTION
4.0 Ice Cream Sandwich	15	
4.1 Jelly Bean	16	99,6%
4.2 Jelly Bean	17	98,1%
4.3 Jelly Bean	18	95,9%
4.4 KitKat	19	95,3%
5.0 Lollipop	21	85,0%
5.1 Lollipop	22	80,2%
6.0 Marshmallow	23	62,6%
7.0 Nougat	24	37,1%
7.1 Nougat	25	14,2%
8.0 Oreo	26	6,0%
8.1 Oreo	27	1,1%

O campo *Minimum Required SDK* indica a menor API suportada pelo seu aplicativo. APIs abaixo desse número não serão capazes de rodar o aplicativo. Prossiga para a próxima tela.

Selecione “Empty Activity” e prossiga. Na última tela, deixe os campos como estão e finalize.

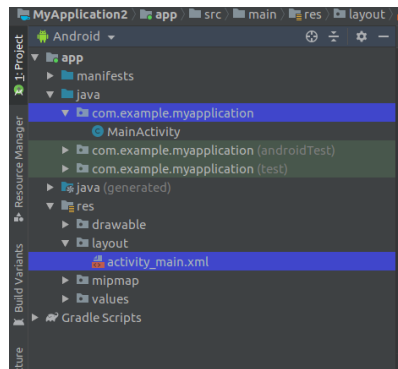


Ao concluir, nós teremos o nosso primeiro projeto Android. (Importante aguardar até que termine de finalizar a sincronização do projeto).

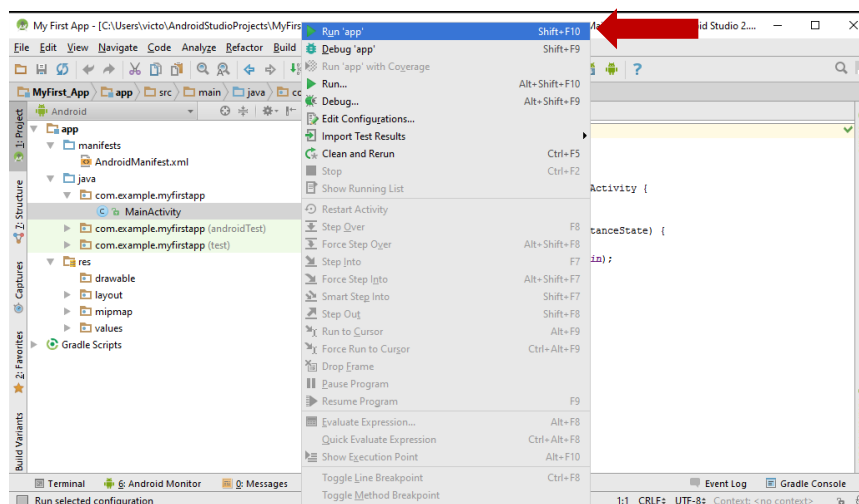


Executando nosso primeiro projeto

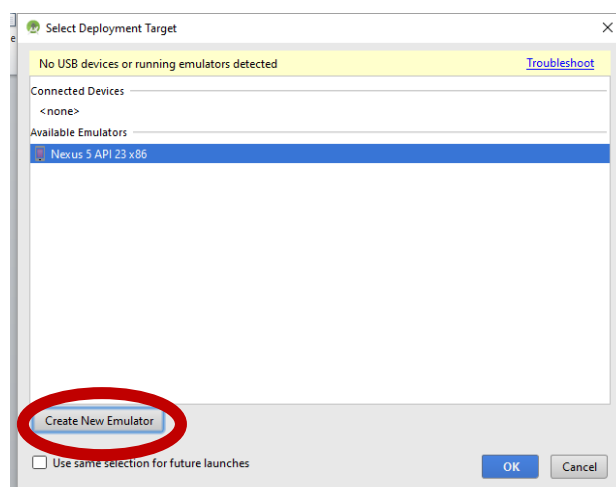
As fontes que devem aparecer abertas são “activity_main.xml” e “MainActivity.java”, sem não fosse o caso deve ficar na seguinte direção:



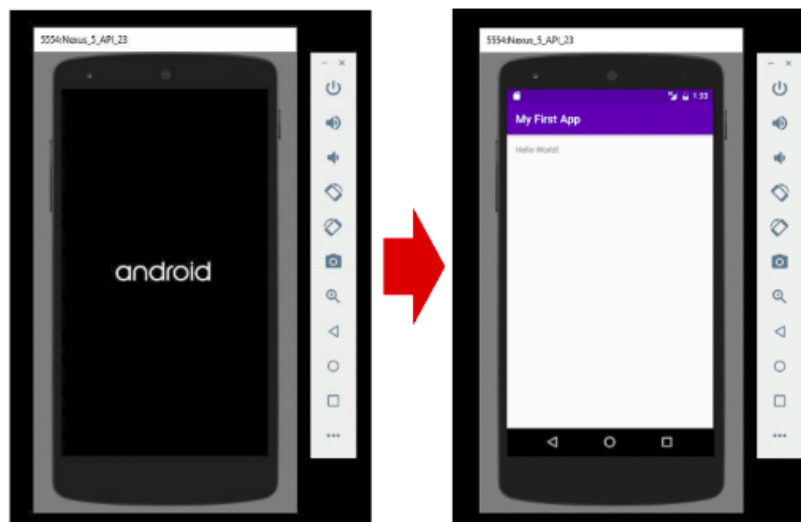
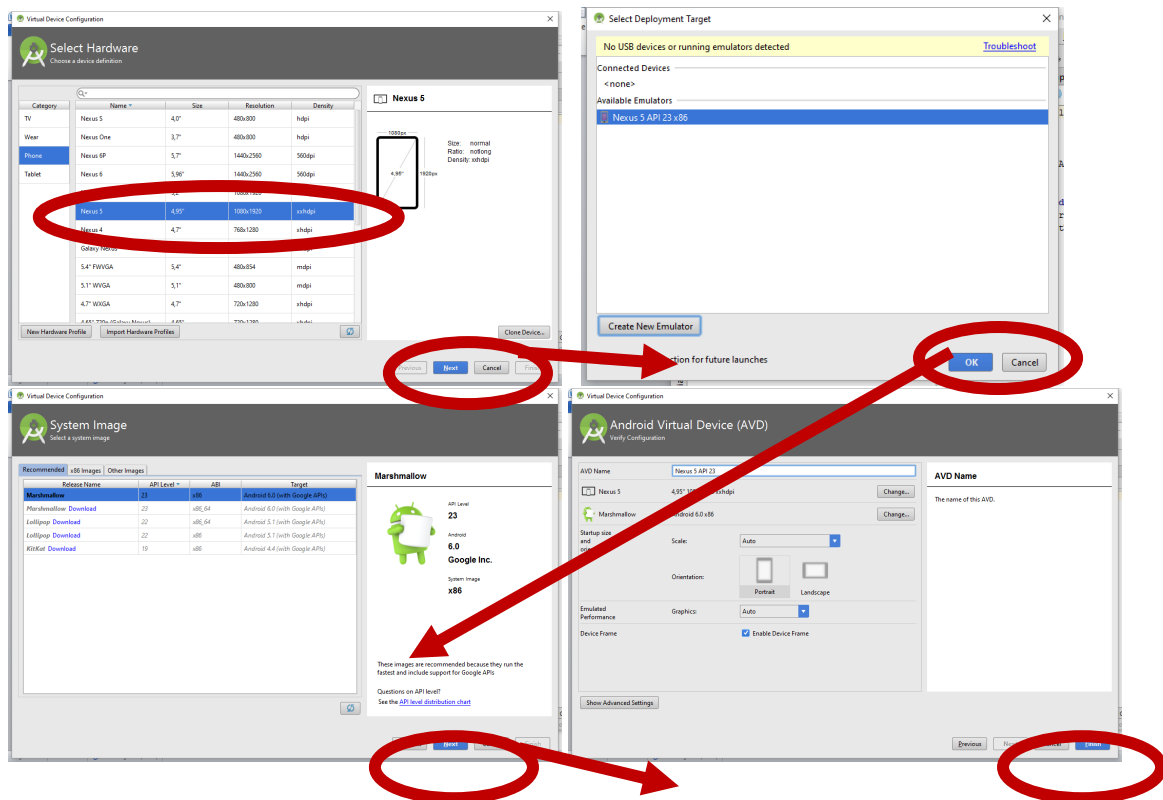
Agora, para roda a aplicação, basta clicar em *Run -> Run App* para listar os dispositivos e emuladores criados no gerenciador.



Se você não tiver nenhum dispositivo virtual criado ainda no gerenciador, clique em *Create New Emulator* na tela que aparecerá.



Selecione Nexus 5 e prossiga até finalizar. Depois de criado, basta clicar em “Ok” para rodar o dispositivo virtual. Atenção: o emulador pode demorar vários minutos para iniciar.



Pronto! Você pode ver na imagem acima à direita, o emulador rodando nosso primeiro projeto Android. Para testar sua aplicação usando o seu aparelho Android é necessário configurá-lo antes.

Conecte o dispositivo ao computador com um cabo USB. Se você está desenvolvendo no Windows, pode ser necessário instalar o driver USB adequado para o seu dispositivo. Para obter ajuda para a instalação dos drivers, consulte:

<https://developer.android.com/studio/run/oem-usb.html>

Ative a depuração USB no seu dispositivo, indo em Configurações > Opções do desenvolvedor. No Android 4.2 e versões mais recentes, as opções do desenvolvedor estão ocultas por padrão. Para torná-lo disponível, vá para Configurações > Sobre o telefone e toque em número de compilação sete vezes. Voltar para a tela anterior para encontrar Opções de Desenvolvedor ativa.

No Android Studio, basta clicar em *Run* -> *Run App*. Selecione seu dispositivo e clique em *Ok*. O Android Studio instalará o app em desenvolvimento no seu aparelho e o executará.

Para testar o ciclo da vida dos Activity com notificações

Coloque as seguintes linhas de código

```
package com.example.myapplication;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);

        Toast.makeText(this, "OnCreate", Toast.LENGTH_SHORT).show();

        // The activity is created

    }

    @Override

    protected void onStart() {

        super.onStart();

        Toast.makeText(this, "OnStart", Toast.LENGTH_SHORT).show();

        // The activity is about to become visible.

    }

    @Override

    protected void onResume() {

        super.onResume();

        Toast.makeText(this, "OnResume", Toast.LENGTH_SHORT).show();

        //The activity has become visible (now it "resumes").

    }

    @Override
```

```
protected void onPause() {  
  
    super.onPause();  
  
    Toast.makeText(this, "OnPause", Toast.LENGTH_SHORT).show();  
  
    // Focus on another activity (this activity is about to be "stopped").  
  
}  
  
@Override  
  
protected void onStop() {  
  
    super.onStop();  
  
    Toast.makeText(this, "OnStop", Toast.LENGTH_SHORT).show();  
  
    // The activity is no longer visible (now it is "stopped")  
  
}  
  
@Override  
  
protected void onDestroy() {  
  
    super.onDestroy();  
  
    Toast.makeText(this, "OnDestroy", Toast.LENGTH_SHORT).show();  
  
    // The activity is about to be destroyed.  
  
}  
}
```

Para continuar o desenvolvimento do projeto

Você pode agora continuar desenvolvendo seu projeto como quiser. A programação é feita em Java, logo você precisa ter conhecimento na linguagem para dar continuidade ao projeto. Para desenvolver a interface, basta arrastar componentes:

