

---

# AGENDA 9

---

**UTILIZANDO  
O ANDROID  
STUDIO PARA  
DESENVOLVIMENTO  
MOBILE**

GEEaD - Grupo de Estudos de Educação a Distância

Centro de Educação Tecnológica Paula Souza

GOVERNO DO ESTADO DE SÃO PAULO

EIXO TECNOLÓGICO DE INFORMAÇÃO E COMUNICAÇÃO

CURSO TÉCNICO EM DESENVOLVIMENTO DE SISTEMAS

PROGRAMAÇÃO MOBILE I

**Expediente**

Autor:

*GUILHERME HENRIQUE GIROLI Revisão*

Técnica:

*Eliana Cristina Nogueira Barion*

Revisão Gramatical:

*Juçara Maria Montenegro Simonsen Santos*

Editoração e Diagramação: *Flávio Biazim*

São Paulo – SP, 2020



## MERGULHANDO NO TEMA...

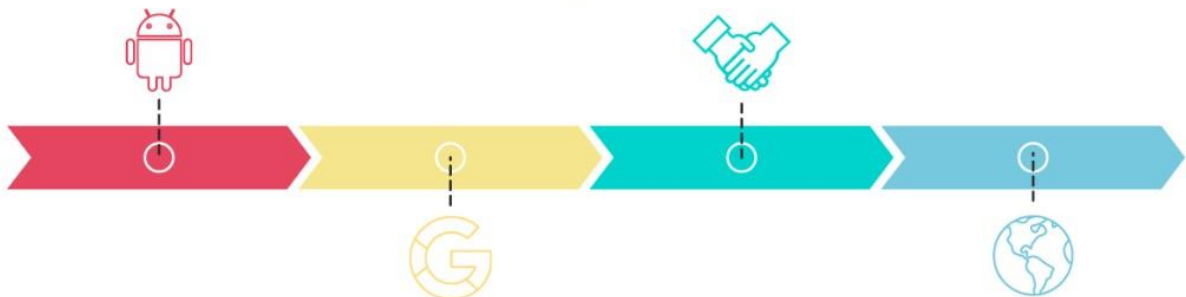
Como todos já sabem, um dispositivo computacional, seja ele qual for, é composto pela sua parte física, denominada **Hardware** e sua parte lógica, chamada de **Software**.

Não é diferente com os dispositivos Mobile, como celulares ou Smartphones, Tablets, etc. Não adiantaria tanta potência e design, sem um sistema operacional para fazer o dispositivo ganhar vida!

Com a evolução dos equipamentos, a indústria necessitava de um sistema genérico, que funcionasse em diversos *Hardwares* diferentes, e que estivesse em constante manutenção e evolução, assim como os dispositivos físicos.

Em setembro de **2003**, para satisfazer as necessidades da indústria, nascia o sistema operacional Android, uma derivação do sistema operacional Linux. A empresa responsável pela criação e desenvolvimento do sistema, diferente do que você imagina, era a empresa Android Inc.

A Google pretendia entrar no mercado de dispositivos móveis e, em **2007**, começou a oferecer um sistema com atualização constante para as principais empresas de Hardware do mercado e, logo, começou a fechar acordos comerciais, embarcando seu sistema nesses dispositivos.



Em 17 de agosto de **2005**, a Google, poderosa empresa de tecnologia e atual detentora do desenvolvimento do projeto Android, comprou a Android Inc, juntamente com o projeto do sistema operacional, contratando toda a equipe de desenvolvimento do sistema da empresa recém adquirida.

Outras empresas do setor de telefonia foram convidadas a conhecer e a fazer parte da aceitação ao novo sistema, como também, empresas de desenvolvimento de sistemas. Assim, iniciava-se a ampla disseminação do Android em todo mundo.



O sistema operacional Android, tem seu código aberto pela Google, e muitas empresas fabricantes de aparelhos, já trabalham alterando esse código, customizando e otimizando para se adequar aos seus produtos. Alguns dispositivos hoje, são lançados contendo uma versão do sistema que possui seu código aberto e parte sendo proprietário da empresa do aparelho, em virtude das customizações e otimizações.

Com sua grande utilização, o sistema operacional Android abriu as portas de um novo mercado, o mercado de desenvolvimento Mobile. O Android, possui uma loja de aplicativos, chamada de “Play Store”, que são comercializados aos seus usuários que buscam ferramentas para o lazer, trabalho, estudo, etc. E com a popularização da loja virtual da empresa Google, muitos desenvolvedores, tiveram a oportunidade de publicar seus trabalhos e ferramentas para todo o mundo, onde muitos saíram do anonimato e se tornando milionários.

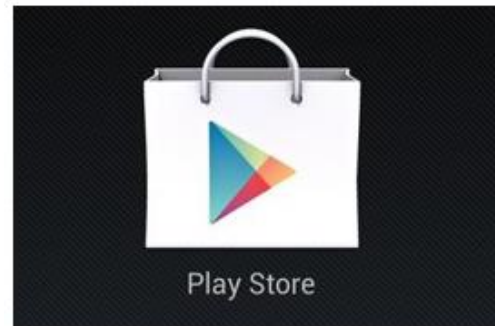


Figura 3 – Aplicativo de compras da loja da empresa Google, chamada de Play Store.

Uma outra grande forma de trabalho para o desenvolvedor, é criar projetos para fins particulares, ou seja, desenvolver aplicativos para empresas como bancos, empresas de vendas, etc.

Para atender cada vez mais a necessidade do mercado de desenvolvimento, foi criada a ferramenta Android Studio, para auxiliar e facilitar o desenvolvimento de aplicativos, seja ele para comercialização na loja da empresa Google, ou para atender empresas específicas.



No tópico a seguir, apresentamos o Android Studio e assim já estaremos aptos para instalar e executar o nosso primeiro projeto desenvolvido para o sistema operacional Android!



Com o surgimento e aceitação do sistema operacional Android, o mercado necessitava de ferramentas para atender o desenvolvimento de aplicativos, gerados a partir da linguagem Java.

Por algum tempo, os desenvolvedores utilizavam a ferramenta Eclipse. Para conseguir operar essa ferramenta, era necessário muitos ajustes e configurações. Muitos desenvolvedores abandonaram os estudos nessa área, pelo simples fato de não conseguir, configurar o ambiente e produzir o seu primeiro projeto. Tornando uma experiência frustrante, seu primeiro contato com esse universo da programação Mobile.

Em maio de 2013, a empresa Google fez o lançamento da sua ferramenta de desenvolvimento para a plataforma Android, o Android Studio. Uma ferramenta de fácil instalação e configuração, utilizando a linguagem de marcação XML (*Extensible Markup Language*) para a criação das telas do aplicativo, as chamadas *Activity's*. E para o desenvolvimento operacional do aplicativo é utilizado a linguagem de programação Java.

O Android Studio é a ferramenta oficial para desenvolvimento exclusivo para a plataforma Android, ou seja, **esta ferramenta não desenvolve para outros sistemas operacionais.**

Atualmente, no mercado contamos com outras ferramentas de desenvolvimento, algumas permitem a programação de um único projeto, e exportação deste projeto para trabalhar/executar nas principais plataformas de dispositivos móveis. Essas ferramentas de desenvolvimento são chamadas de **multiplataforma.**



*Muitas empresas e desenvolvedores, defendem a ideia de que para a criação de um bom aplicativo, é fundamental que ele tenha desempenho e baixa utilização dos recursos do dispositivo Mobile.*

Desta forma, levantam a bandeira de que o desenvolvimento ideal é aquele realizado na ferramenta exclusiva da plataforma, pois essa ferramenta não trabalha com conversões, ela é nativa e desenvolvida para atender as necessidades de um único sistema operacional. É válido lembrar que isso não é uma lei, é apenas a opinião de uma parcela de desenvolvedores.

Por outro lado, o mercado de ferramentas de desenvolvimento multiplataforma cresce, pois otimizam o tempo gasto na produção de aplicativos.

Devidamente apresentada nossa ferramenta de desenvolvimento Android Studio, chegou a hora de efetuar a instalação dela, que além de robusta e sensacional, conta com a licença de uso no nível Apache2.0, ou seja, ela é de uso gratuito! Vamos aproveitar essa oportunidade ao máximo?

**Instalação Android Studio - <https://www.youtube.com/watch?v=EQDZqjMOX78>**

## Activity

A **Activity** é um componente do aplicativo responsável por fornecer uma tela para o usuário, desta forma gerando uma interface com ele. O aplicativo pode conter uma ou mais *Activity's*, de acordo com a sua necessidade.

Nos aplicativos com mais de uma tela, ou seja, mais de uma *Activity*, uma delas é escolhida como principal. É esta *Activity* principal que será exibida inicialmente ao usuário, e assim, será a responsável por chamar as outras telas do projeto.

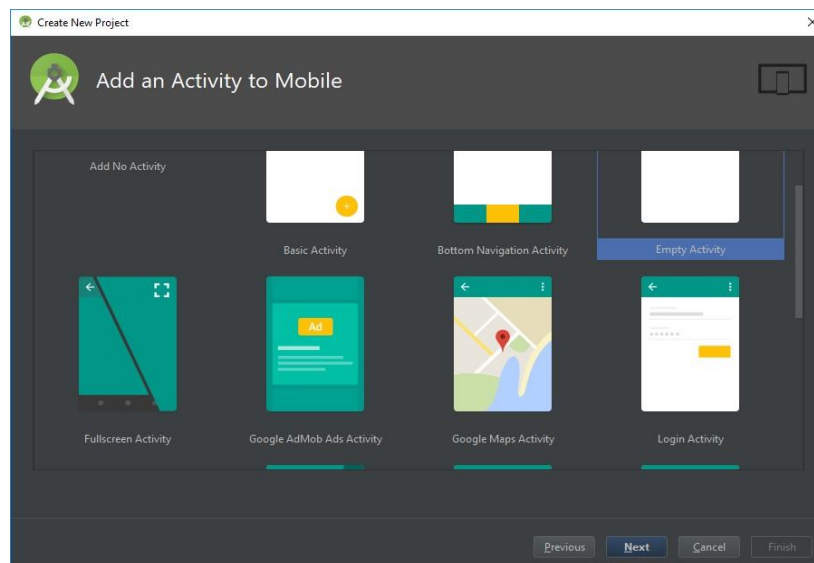


Figura 1 – Tela de sugestões de Activity.

Para o desenvolvimento da *Activity*, utilizamos o modo “Design” ou “XML”. No modo “Design” trabalhamos com uma paleta de componentes, onde arrastamos para a nossa tela, tornando o processo mais rápido, pois ele gera automaticamente o XML.

Já no modo “XML”, utilizamos a linguagem de marcação para codificar nossa interface, e por este processo o usuário necessita conhecer bem seus códigos e funções.

Na criação de um projeto no Android Studio, nossa IDE fornece alguns modelos de *Activity*'s, previamente desenvolvidas para ajudar os desenvolvedores Mobile no seu trabalho. A figura 1 mostra alguns exemplos de telas, para usos diversos, como menus, mapas, login, etc. Essas telas ficam disponíveis para utilização ou alteração do desenvolvedor, fazendo assim ele ganhar tempo no desenvolvimento de aplicativos.

Em nosso projeto inicial, vamos utilizar a “**Empty Activity**”, ou seja, é uma *Activity* vazia! Utilizada para a criação de projetos específicos, que necessita de uma tela neutra.



Marcelo, vamos criar nosso primeiro projeto no Android Studio?

Agora podemos desenvolver o primeiro projeto no Android Studio! É fundamental que neste momento você esteja na tela inicial do Android Studio (caso tenha algum projeto aberto, clique em “**File -> Close Project**”). Vamos iniciar clicando em “**Create New Project**” como demonstra a figura 2.



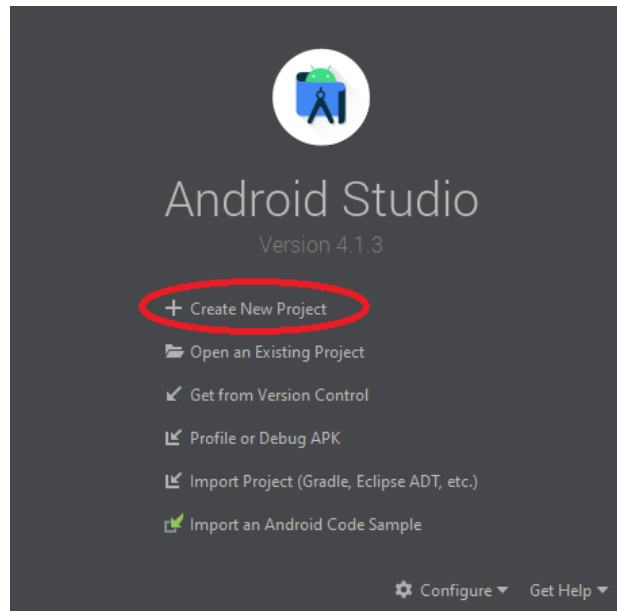


Figura 2 – Tela inicial do Android Studio.

Quando solicitamos a criação de um novo projeto, uma tela de assistente é aberta, para auxiliar nessa etapa.

Agora, chegou a hora de escolher a **Activity**, isto é, a primeira tela do projeto. O *Android Studio* permite desenvolvimento para os mais diferentes tipos de dispositivos mobile, desde relógios, passando por smartphones, até chegar em carros.

É nessa etapa que escolhemos quais dispositivos programar. O foco inicial são os smartphones, então vamos até a guia "**Phone and Tablet**".

Como combinamos anteriormente, vamos escolher a "**Empty Activity**", e aplicar "**Next**", como mostra a figura 3.

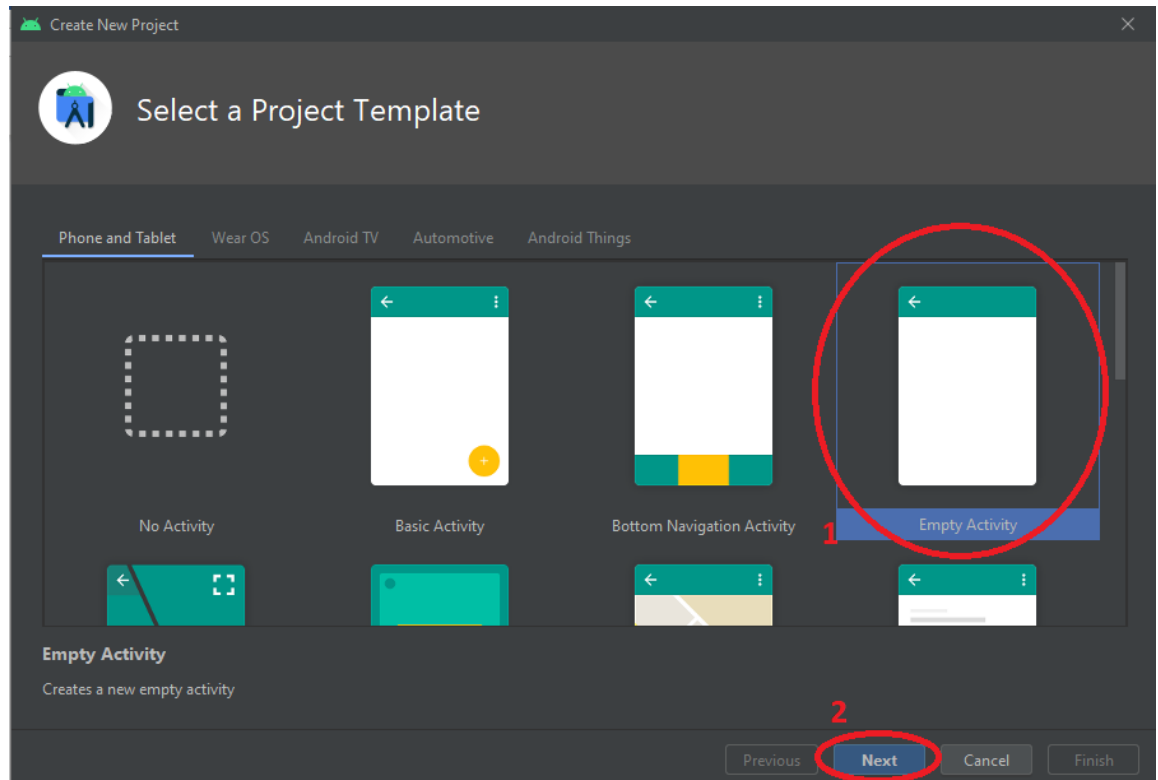


Figura 3 – Escolha da Activity do projeto.

São solicitados dados do projeto como o Nome da Aplicação e Nome do pacote. Vamos informar para o **"Name:"** do projeto **"Aula01"** e não alteraremos o **"Package name"**, ou nome do pacote do projeto.



*É importante lembrar que o Java e Android Studio utiliza "Case Sensitive", ou seja, faz diferenciação com letras maiúsculas e minúsculas, note na figura 17, como foi escrito o nome do projeto, isso faz total importância no decorrer da sua programação!*

Escolha uma pasta de fácil acesso no seu Hard Disk (HD) para salvar seu projeto, é importante criar uma pasta com um nome fácil e curto. Vamos utilizar uma pasta no **"C:"** com nome de **"androidDS"**.

Para desenvolvimento desse e dos demais projetos, vamos trabalhar com a linguagem de programação JAVA.

Indique qual é a API mínima que o dispositivo deve ter para funcionamento do aplicativo, vamos escolher a **"API 25: Android 7.1.1 (Nougat)"**. Lembre-se de instalar essa API no processo de instalação do Android Studio.

Veja como ficou os dados do primeiro projeto na figura 4 com a tela do assistente. Depois, é só clicar no botão **"Finish"**.

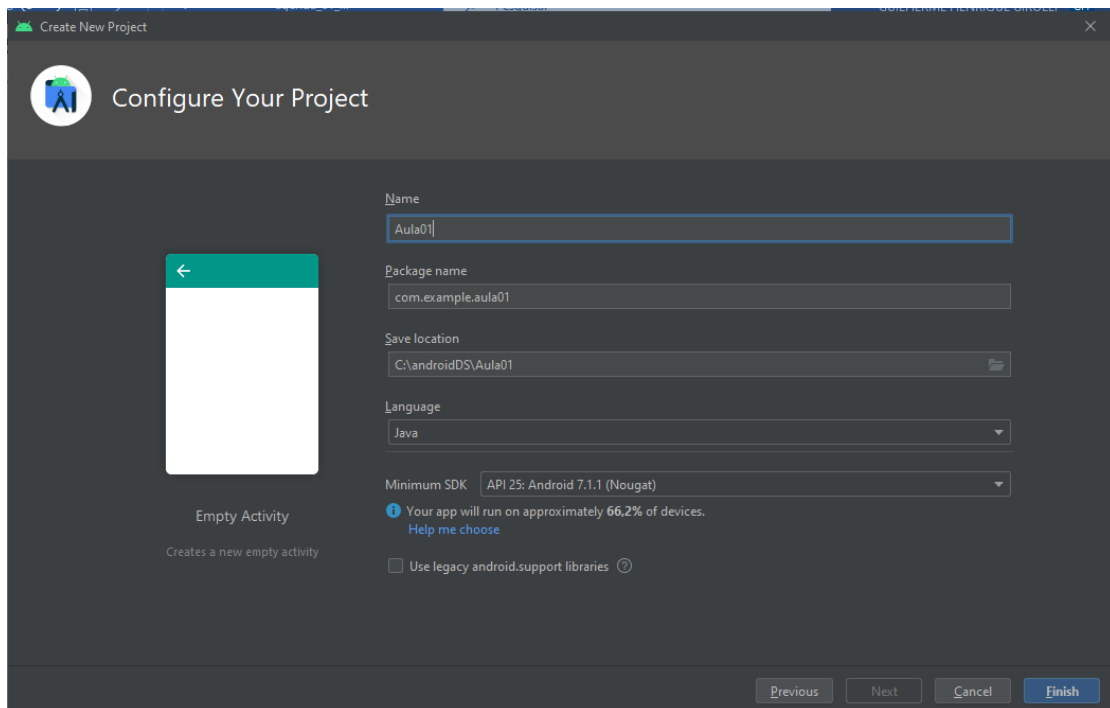


Figura 4 – Início da criação do projeto Aula01.

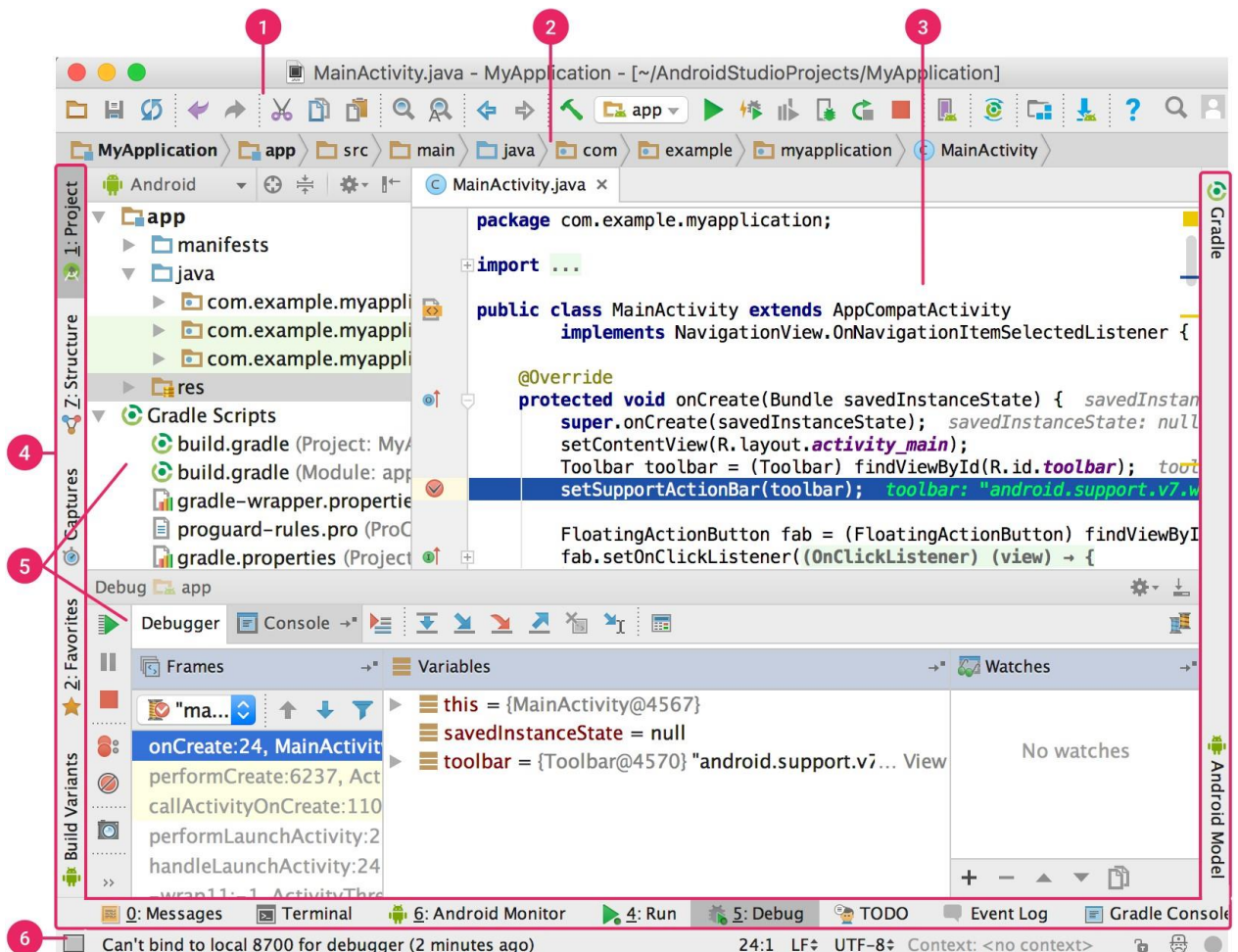


*Uma observação importante, vista na figura 4, é que ao escolher a API, o Android Studio nos informa uma estimativa de dispositivos compatíveis com o aplicativo que será desenvolvido, neste caso 66,2% de todos os dispositivos do mundo. Isso é importante, pois nos auxilia no processo de escolha da API mínima em projetos futuros, durante a sua promissora carreira de desenvolvedor Mobile.*

## Conhecendo a IDE (Integrated Development Environment) do Android Studio

Após todas as etapas anteriores, e criação do primeiro projeto, chegou a hora de conhecer alguns detalhes importantes sobre a ferramenta. Precisamos alinhar alguns termos e locais que utilizaremos com mais frequência na IDE (Integrated Development Environment) ou Ambiente Integrado de Desenvolvimento. Assim o processo de evolução se torna mais rápido.

A figura 5, retirada do site oficial do Android Studio, identifica todas as áreas da nossa IDE, com uma breve explicação sobre cada grupo de itens. Vamos acompanhar!



- 1 A **barra de ferramentas** permite executar diversas ações, incluindo executar aplicativos e inicializar ferramentas do Android.
- 2 A **barra de navegação** ajuda na navegação pelo projeto e na abertura de arquivos para edição. Ela oferece uma visualização mais compacta da estrutura visível na janela **Project**.
- 3 A **janela do editor** é o local em que você cria e modifica código. Dependendo do tipo de arquivo atual, o editor pode mudar. Por exemplo, ao visualizar um arquivo de layout, o editor abre o Editor de layout.
- 4 A **barra de janela de ferramentas** fica fora da janela do IDE e contém os botões que permitem expandir ou recolher a janela de cada ferramenta.
- 5 A **janela das ferramentas** dá acesso a tarefas específicas, como gerenciamento de projetos, busca, controle de versão e muitos outros. Você pode expandi-las e recolhê-las.
- 6 A **barra de status** mostra o status do projeto e do próprio IDE, além de advertências e mensagens.

Figura 5 – Explicação sobre as áreas da IDE do Android Studio. Disponível em: <https://developer.android.com/studio/intro/?hl=pt-br>

A estrutura do projeto é exibida pela ferramenta “**Project**”. Nela encontramos os arquivos Java, arquivos de configuração do projeto, arquivos XML que geram a interface.

Para acesso aos arquivos Java da aplicação, devemos localizar o pacote criado, que fica em “**App** -> **java** -> **com.example.\*.aula01**”, conforme figura 6, e clicar duas vezes sobre o arquivo que deseja abrir.

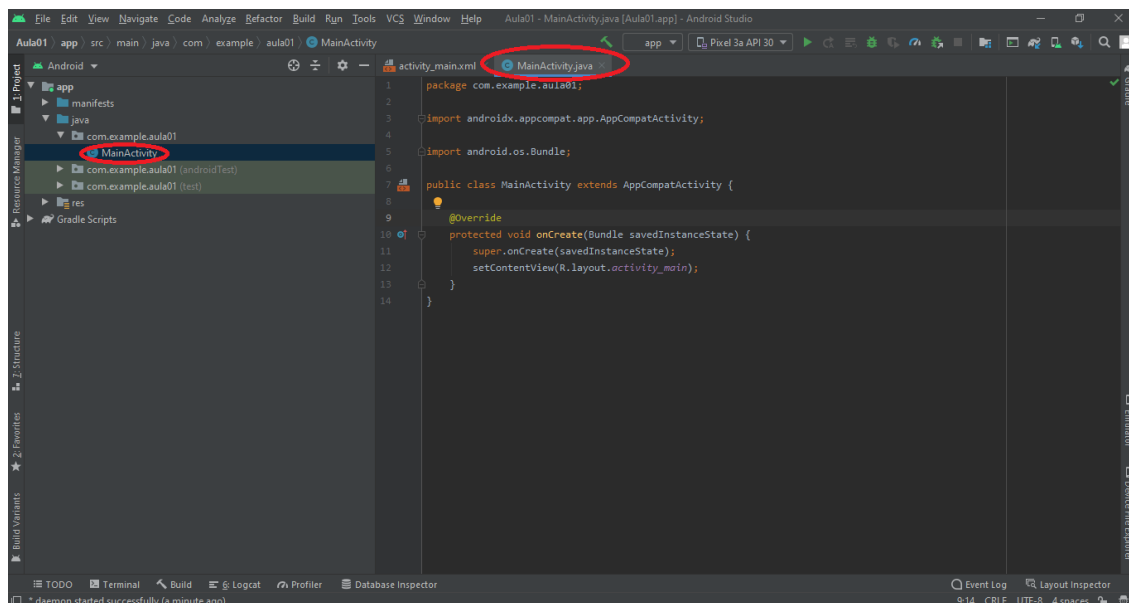


Figura 6 – Arquivos Java do projeto.

Para acesso aos arquivos de *Layout* das *Activity's* que geram as telas, basta procurar na ferramenta “**Project**” a pasta *layout*, que fica em “**App -> res -> layout**”. Verifique a figura 7, e dê duplo clique para abrir o arquivo.

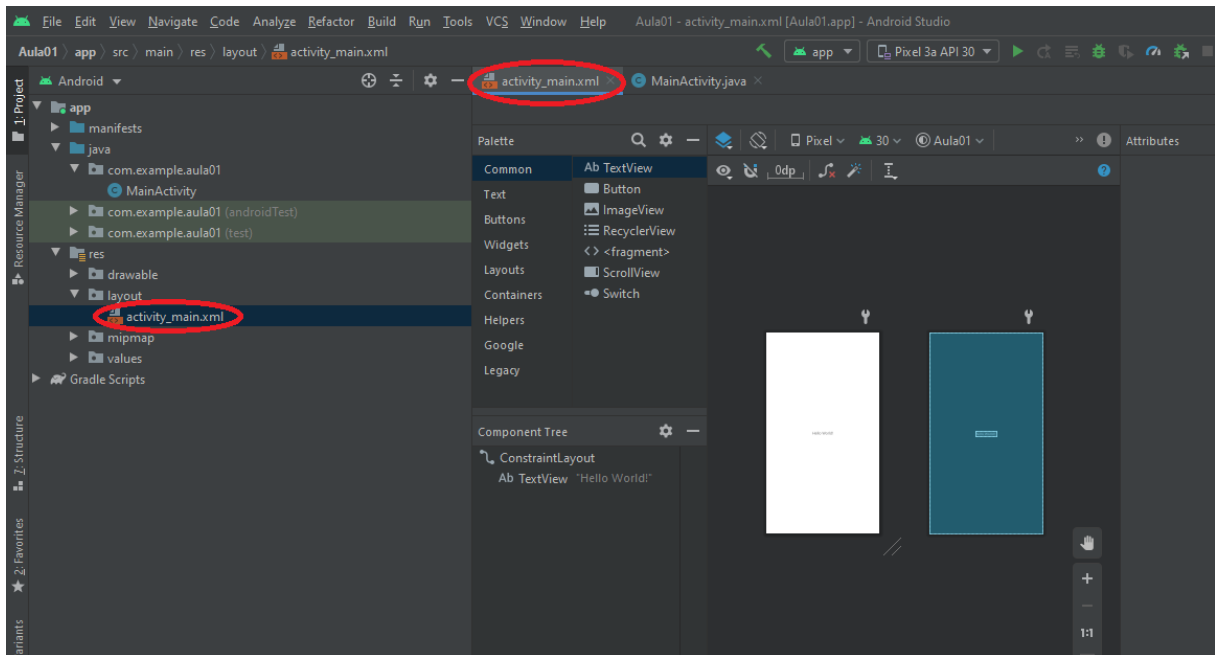


Figura 7 – Arquivos XML do projeto.

Na barra de ferramentas do Android Studio, é importante também conhecer alguns botões responsáveis pela execução da aplicação e outras ferramentas do Android Studio, verifique os figura 8.

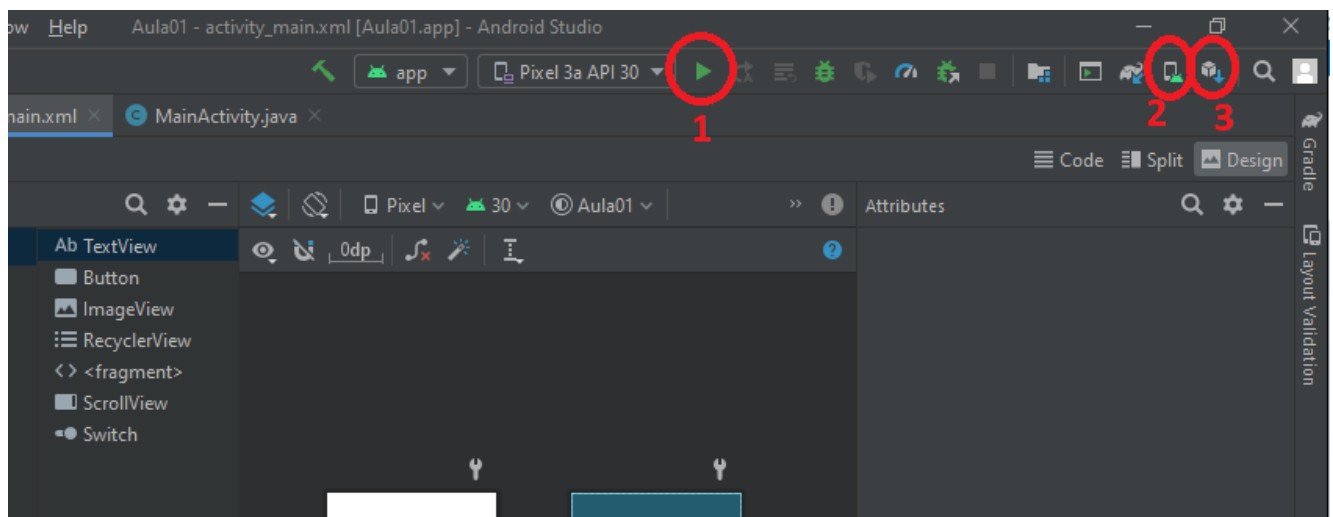


Figura 8 – Barra de ferramentas do Android Studio.

**Item 1 – Run ‘app’, ou seja, execução do aplicativo para teste.**

**Item 2 – AVD (Android Virtual Device) Manager.**

**Item 3 – Segunda forma para acessar o SDK Manager, visto anteriormente neste material.**

A seguir conheça a AVD (Android Virtual Device), necessária para executar os aplicativos durante a fase de desenvolvimento.

### AVD (Android Virtual Device)

Como verificamos, o Android Studio desenvolve aplicativos nativos para o sistema operacional Android. O ambiente de laboratório e estudos que Marcelo utiliza é o sistema operacional Windows 10, o que gerou certa dúvida. Como fazer para testar o aplicativo durante o seu desenvolvimento?

Assim, é importante conhecer a AVD (*Android Virtual Device*), ou Dispositivo Virtual Android que é uma ferramenta presente no Android Studio, para emular um dispositivo Android no sistema operacional Windows 10. Essa ferramenta nada mais é que uma máquina virtual que roda no sistema operacional Linux o Android.

Também é permitido a execução do aplicativo em dispositivos físicos conectados ao Android Studio, por meio do cabo USB e até pela rede Wi-Fi, porém esse dispositivo necessita que o Android tenha o modo de desenvolvedor habilitado. Para saber mais sobre esse processo, efetue uma pesquisa na documentação oficial do Android Studio disponível em <https://developer.android.com/studio/run/device?hl=pt-br>.

A AVD torna mais flexível o processo de testes do aplicativo porque podemos gerar um número infinito de equipamentos, e assim executar a aplicação em diversos tipos de dispositivos, o que às vezes não seria possível com os equipamentos físicos de alto custo, como *Smartphone* de última geração. Os desenvolvedores iniciantes e empresas pequenas, não contam com recursos financeiros altos para compra de equipamentos, ainda mais para testes, não é verdade?

## Executando o projeto

Com o projeto criado, podemos executá-lo na AVD desenvolvida no processo de instalação do Android Studio. A *Activity* criada possui um *TextView* com o texto **“Hello World”**. Vamos executar o projeto para efetuar um teste, clicando no botão **“Run app”**.

O processo de execução do aplicativo pode demorar alguns segundos, dependendo da configuração do seu computador. É importante você verificar na barra de status as etapas que estão sendo executadas pelo Android Studio. A figura 9 mostra a barra de status e uma barra de progressão em uma das etapas da execução do aplicativo.

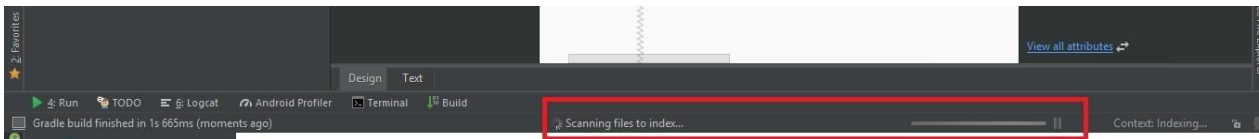


Figura 9 – Barra de Status do Android Studio.

Não é necessário configurar nada na AVD, após execução do aplicativo, ele aparece automaticamente na tela do celular virtual, como demonstra a figura 10.



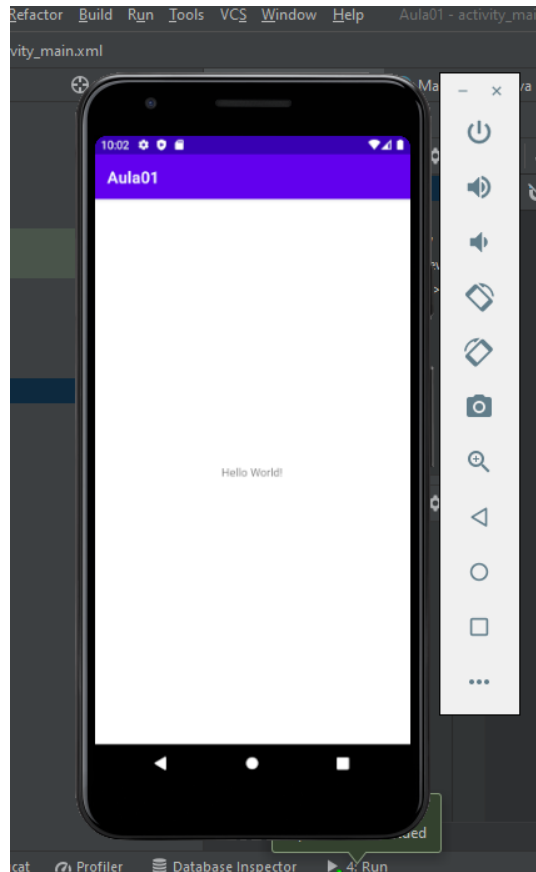


Figura 10 – AVD executando o aplicativo.

## Desenvolvendo o projeto Calculadora de soma entre dois valores

Vamos seguir mergulhando nos estudos do Android Studio e pausar a execução do aplicativo por meio do botão “*Stop app*” como apresentado na figura 11. Vale lembrar que não é necessário fechar a AVD, ela pode ficar em execução minimizada em seu computador, sem interferir no desenvolvimento do projeto.

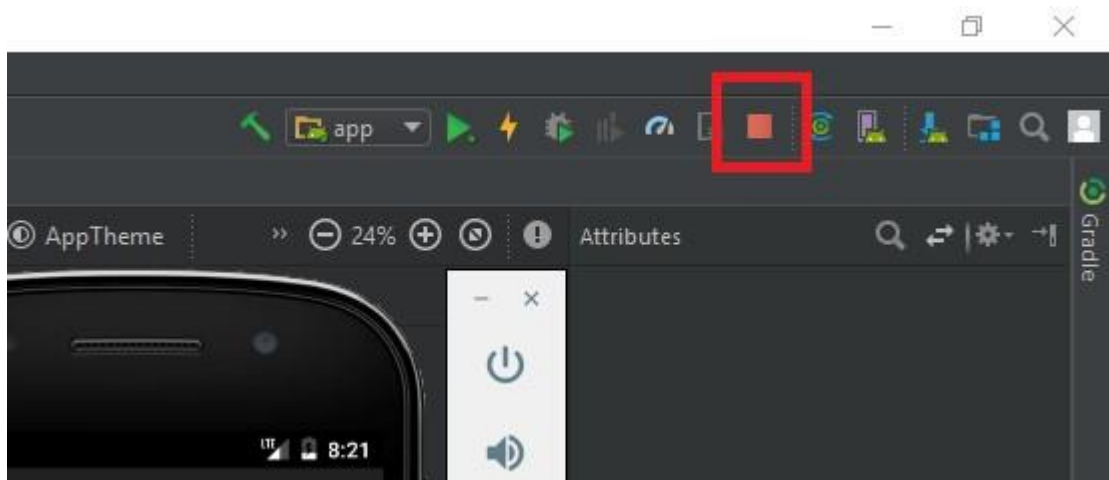


Figura 11 – Botão “Stop app”.

Para desenvolver o projeto calculadora, é necessário popular a *Activity* com alguns componentes. Vamos utilizar os seguintes componentes para esse projeto: *TextView*, *EditText* e *Button*. Cada componente com seus respectivos *ID* e *Text*.

O *ID* é o identificador do componente, ou seja, o nome dele. E este identificador é único no projeto, assim o Android Studio não permite a inserção de dois componentes com o mesmo *ID*. Ele é utilizado para “chamar” a utilização de um componente na programação Java do aplicativo.

A figura 12, mostra o *Layout* do aplicativo, da interface gráfica com o usuário. Para desenvolver este trabalho assista ao vídeo 1 a seguir que aborda a construção do *layout* da aplicação. Esta etapa utiliza o arquivo “*activity\_main.xml*” ou “*activity\_calculadora.xml*” como apresentado no vídeo a seguir.



Figura 12 - Layout do aplicativo.



Vídeo 1- Layout do aplicativo. Disponível em: <https://www.youtube.com/watch?v=Y4wh6K18Qro>

Após o desenvolvimento da *Activity*, é necessário efetuar a programação da classe Java, responsável pelas ações dos componentes presentes na interface gráfica. O arquivo “**MainActivity.java**” ou “**Calculadora.java**” contém a classe responsável pelas ações da interface e de seus componentes. A figura 13 apresenta a codificação inicial da classe.

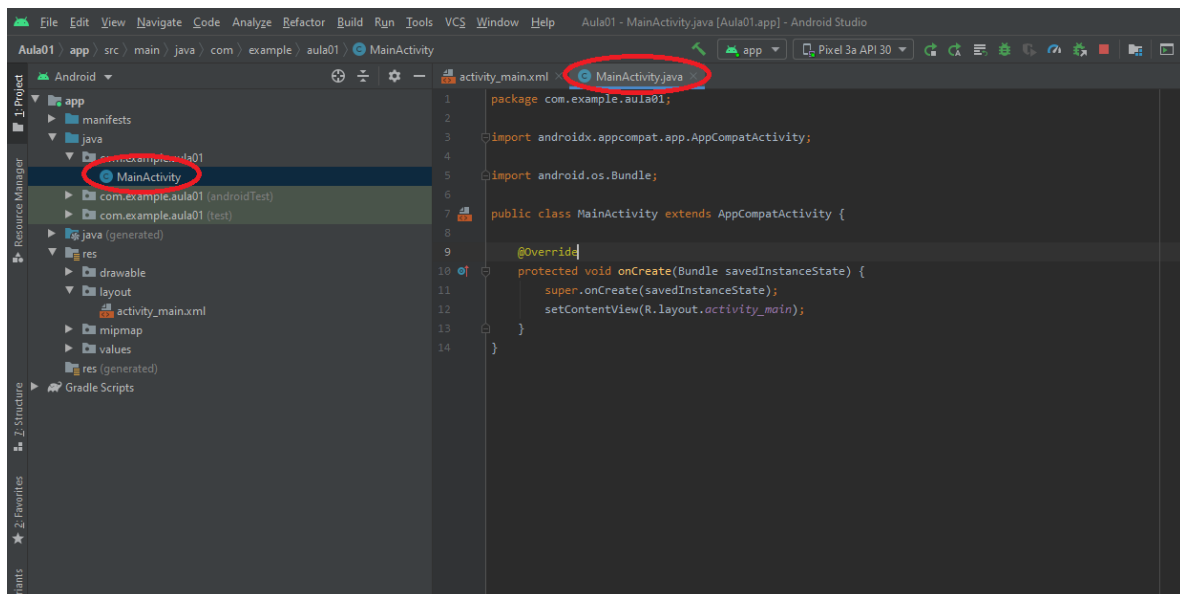


Figura 13 - Classe “MainActivity.java” ou “Calculadora.java”.

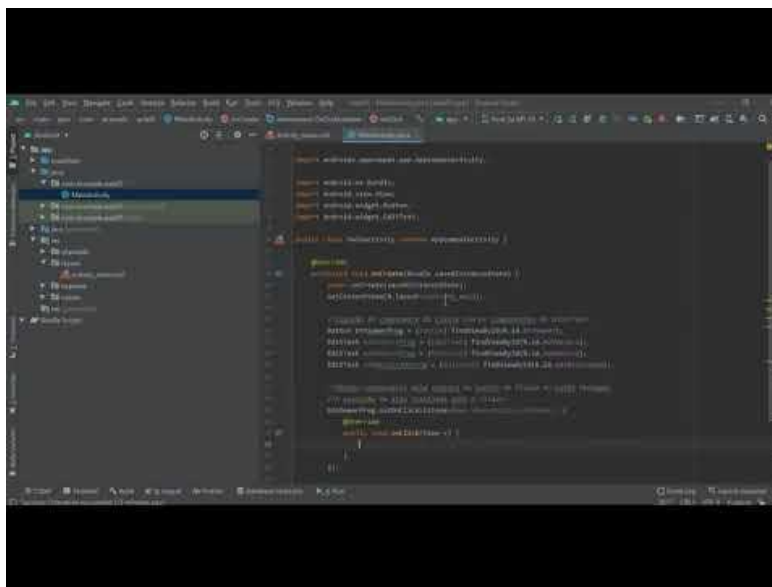
Para o funcionamento do aplicativo é necessário desenvolver uma espécie de ligação, entre o componente da interface gráfica e o componente criado na classe. Essa ligação é utilizada apenas nos componentes que demonstram interatividade com o usuário, ou seja, aqueles que têm uma comunicação direta com o usuário. Um exemplo é o **“EditText”** que recebe ou exibe um determinado valor para o usuário.

O método **onCreate** é responsável basicamente pela criação e configuração da *Activity* na execução do aplicativo. É nele que vamos implementar o método que captura o evento de clique no botão “Somar” da interface, e converte esse clique por meio do método **onClick** em ações.

Para efetuar e mostrar a soma entre o valor 1 e o valor 2, é necessário efetuar uma conversão. O **“EditText”** entrega os números na forma de caracteres, desta forma, é necessário converter para o tipo *Double*. Vamos utilizar também as variáveis para guardar os valores iniciais e o resultado, todas as três do tipo *Double*.

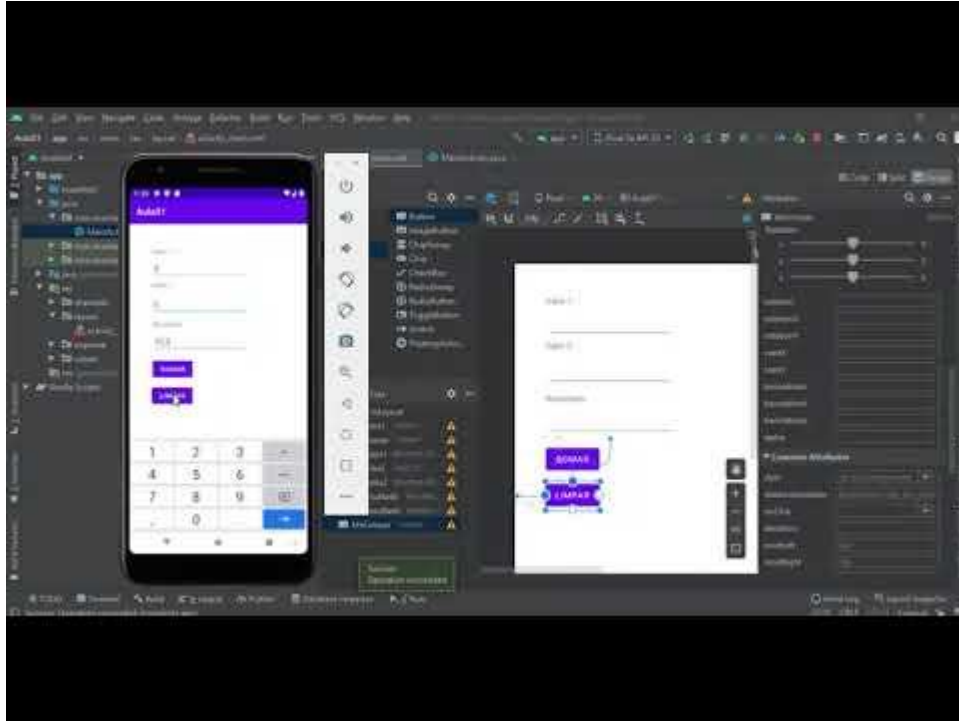
Após a leitura e armazenamento dos valores, efetuamos a soma, e na sequência, voltamos o resultado para a interface. Para apresentar o resultado na interface do usuário, é necessário efetuar uma conversão, já que o resultado se encontra no formato *Double*, e o **“EditText”** recebe apenas valores do tipo texto (*String*).

Para desenvolver estas etapas assista o vídeo 2 a seguir que aborda a construção da classe Java da aplicação. Esta etapa utiliza o arquivo **“MainActivity.java”** ou **“Calculadora.java”**.



Vídeo 2 – Códigos Java do aplicativo. Disponível em: <https://youtu.be/sSYgNW4LmQg>

Para treinar e fixar o conteúdo apresentado até aqui, vamos inserir mais um botão no projeto. Verifique, no vídeo 3, o processo para construção do botão Limpar.



Vídeo 3 – Códigos Java do aplicativo. Disponível em: <https://youtu.be/c5N4ROuB2ws>

Veja a seguir como ficou o arquivo **“MainActivity.java”** ou **“Calculadora.java”**.

```
package com.example.aula01;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

public class MainActivity extends AppCompatActivity {

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

```

//Ligação do componente da classe com os componentes da interface.
Button btnSomarProg = (Button) findViewById(R.id.btnSomar);
Button btnLimparProg = (Button) findViewById(R.id.btnLimpar);
EditText edtValor1Prog = (EditText) findViewById(R.id.edtValor1);
EditText edtValor2Prog = (EditText) findViewById(R.id.edtValor2);
EditText edtResultadoProg = (EditText) findViewById(R.id.edtResultado);

//Método responsável pela captura do evento de Clique no botão btnSomar
//e execução da ação realizada após o clique.
btnSomarProg.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        //Conversão e armazenamento dos caracteres obtidos na interface
        //gráfica.
        double num1 = Double.parseDouble(edtValor1Prog.getText().toString());
        double num2 = Double.parseDouble(edtValor2Prog.getText().toString());
        //Soma e armazenamento do resultado.
        double resultado = num1 + num2;
        //Retorno para a interface gráfica do resultado.
        edtResultadoProg.setText(String.valueOf(resultado));
    }
});

//Método responsável pela captura do evento de Clique no botão btnLimpar
//e execução da ação realizada após o clique.
btnLimparProg.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        edtResultadoProg.setText("");
        edtValor1Prog.setText("");
        edtValor2Prog.setText("");
        edtValor1Prog.requestFocus();
    }
});
}
}

```

Acompanhe a “Atividade Online” para colocar em prática todo o conteúdo da Agenda 1.