

# Aula XIII de Desenvolvimento de Aplicativo I

Configurando o celular para depuração

Detalhes importantes da Interface do Android Studio

Layouts

Componentes de Layout

Começando um projeto

---

## Fontes:

<https://developer.android.com/training/basics/firstapp?hl=pt-br>

<https://developer.android.com/training/basics/firstapp/building-ui?hl=pt-br>

<https://www.devmedia.com.br/tutorial-de-android-studio/34003>

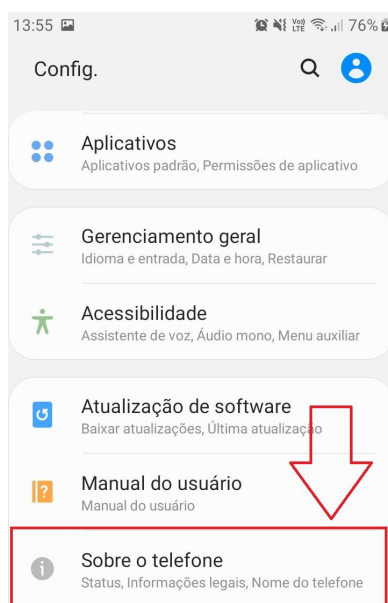
---

## Configurando o celular para depuração

Se você tem um celular Android e pode ser com uma versão mais antiga, podemos fazer a configuração para depuração pelo próprio dispositivo, desta maneira iremos literalmente poupar hardware e tempo para podermos ver a compilação e depuração diretamente pelo celular.

### Como fazer?

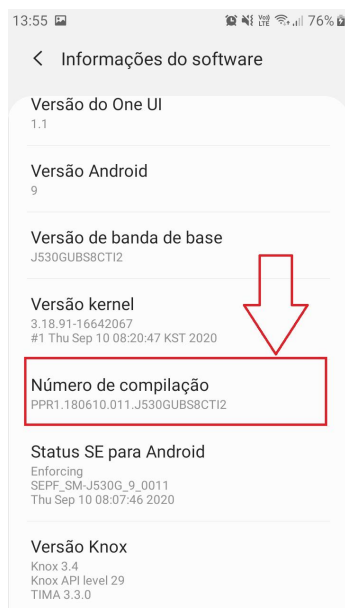
1. Pegue seu celular Android e acesse suas configurações;
2. Com as configurações abertas, acesse o menu **Sobre o telefone**;



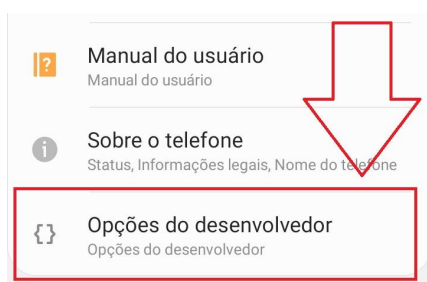
3. Após, você vai clicar sobre **Informações sobre o Software**;



4. Depois em **Número de compilação**, aqui você irá pressionar umas 10 vezes até receber a mensagem que o modo de desenvolvedor do telefone foi ativado.



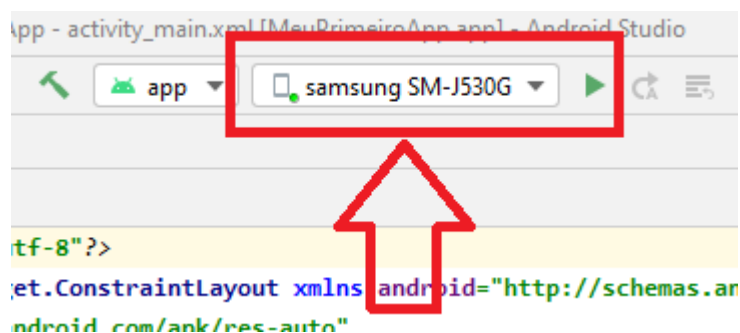
5. Retorne para a tela de **Configurações**, dê uma atualizada nesta tela e verifique se está aparecendo no final da tela;



6. Acesse **Opções do desenvolvedor** e habilite o recurso **Depuração USB**;

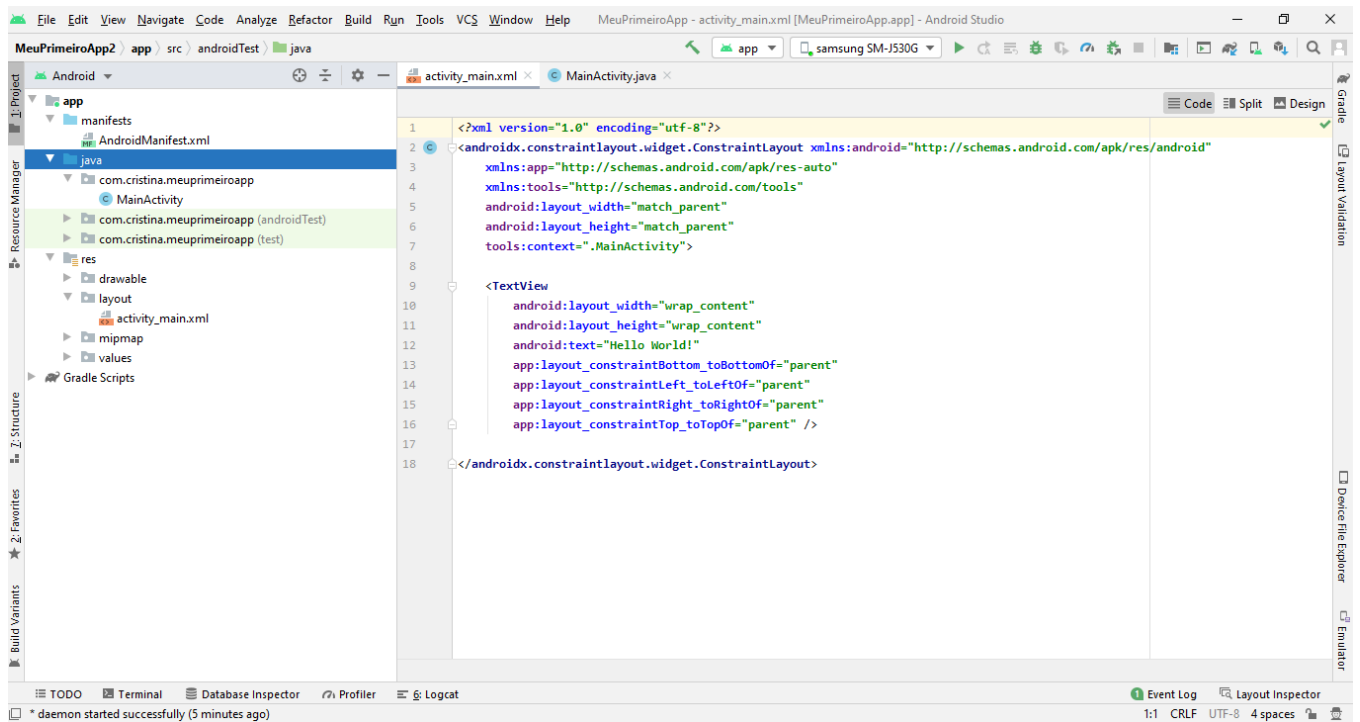


7. Pronto, seu celular está apto a receber todas as informações do Android Studio, **basta só conectá-lo via USB no computador**. Agora vamos abrir o Android Studio e verificar se o seu celular irá aparecer na lista de dispositivos para compilação.
8. Pronto, apareceu! Seu Android Studio está pronto para exibir as informações e desenvolvimento que faremos no celular. **Obviamente que a primeira vez que mandarmos a compilação ele vai demorar, mas isso é normal em função do Gradle que precisa "buildar", construir o gerenciamento da compilação.**

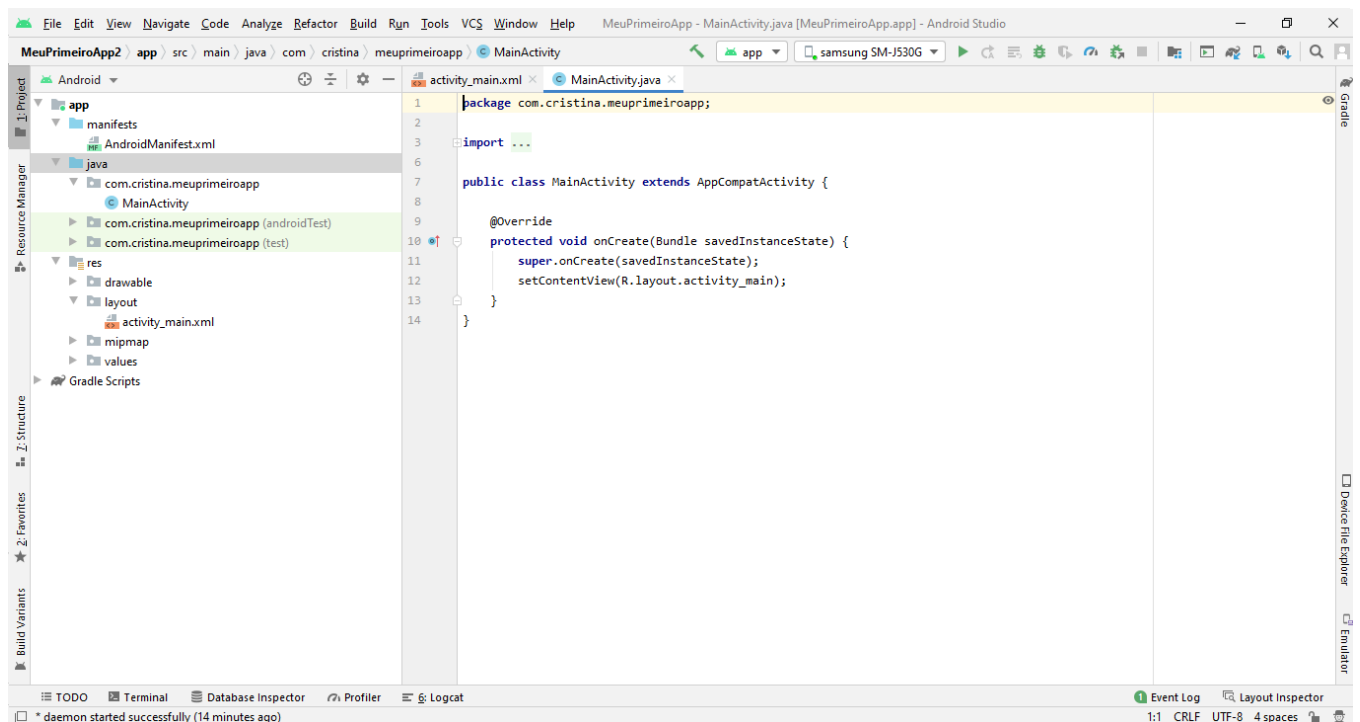


## Detalhes importantes da Interface do Android Studio

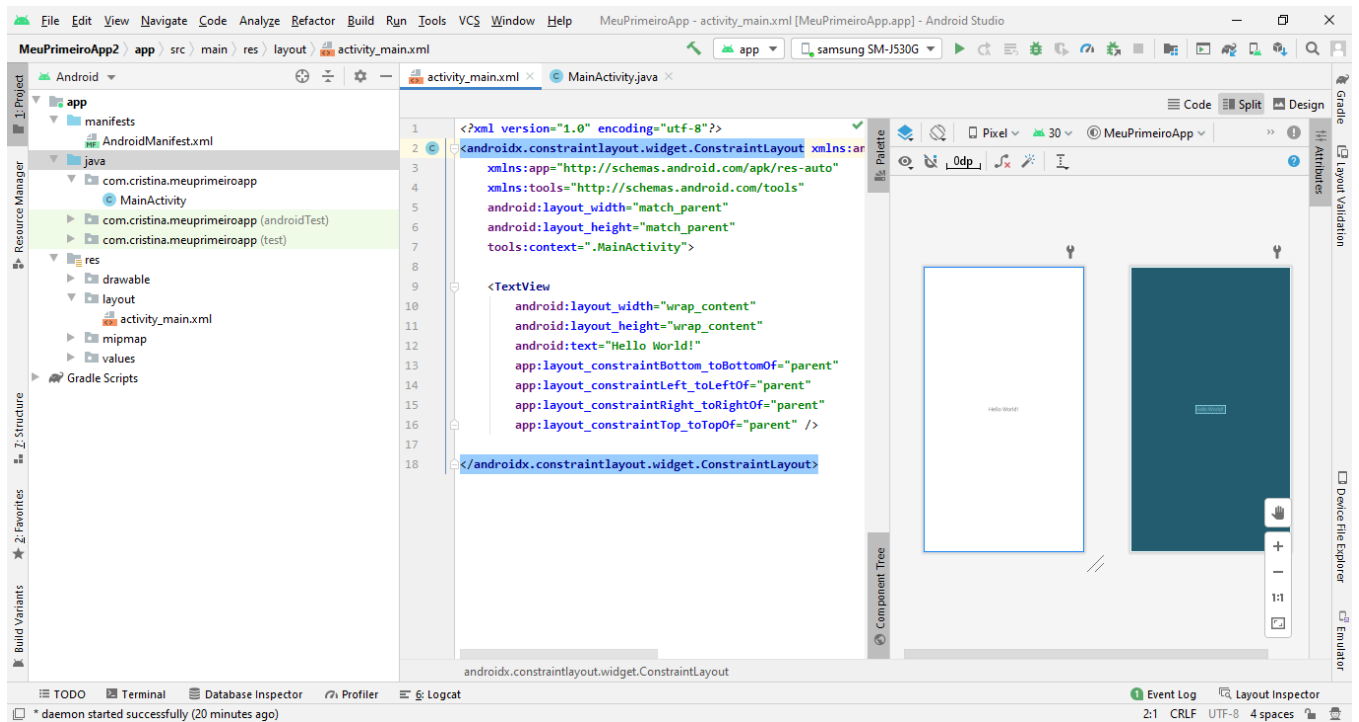
Quando abrimos um novo projeto e escolhemos a opção Empty Activity, um código XML é gerado automaticamente. Este arquivo xml conterá todas as informações dos componentes que a nossa aplicação terá, como por exemplo textos, imagens, vídeos e etc. O modo de visualização da imagem abaixo é **Code**:



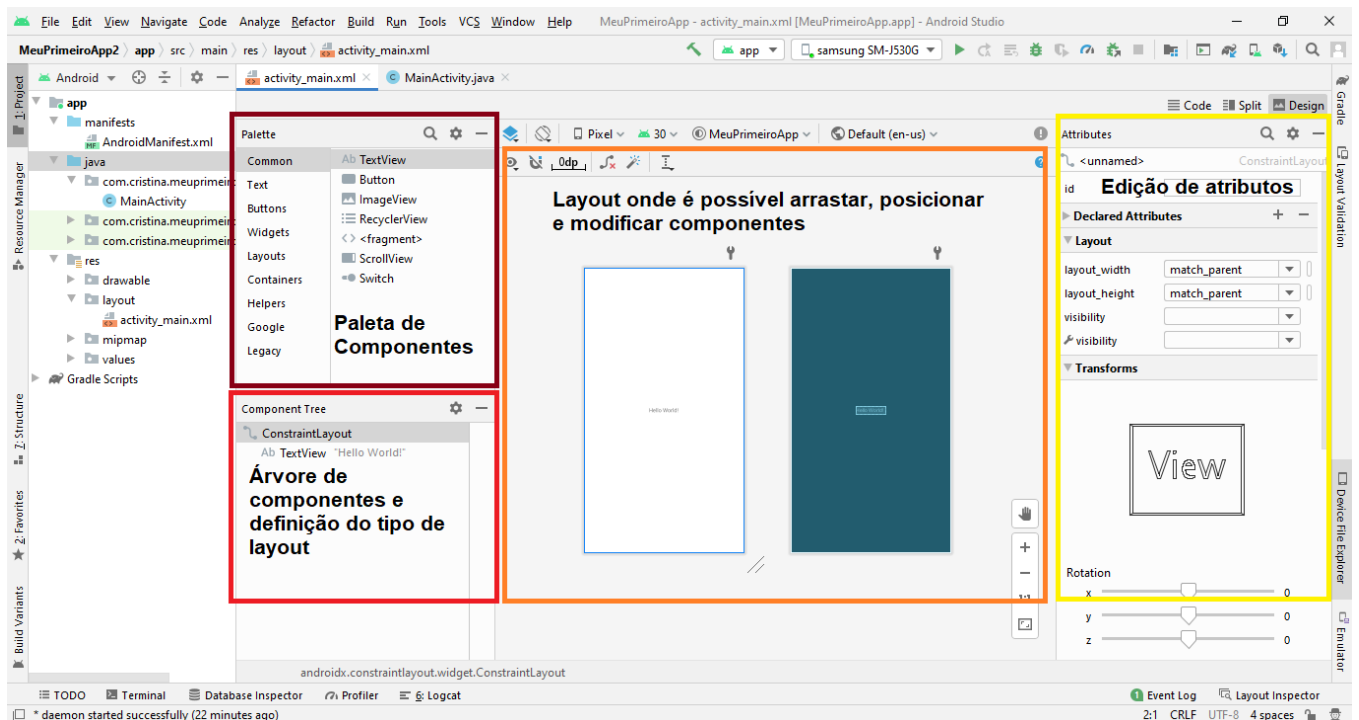
Já o arquivo em Java, será responsável pelo **“setamento e mapeamento”** dos componentes, pela **Classe R** (que logo veremos) e pela lógica e programação em Java propriamente dito, ou seja, a programação da nossa aplicação.



No modo **Split**, poderemos trabalhar diretamente olhando o nosso layout e a parte da programação.



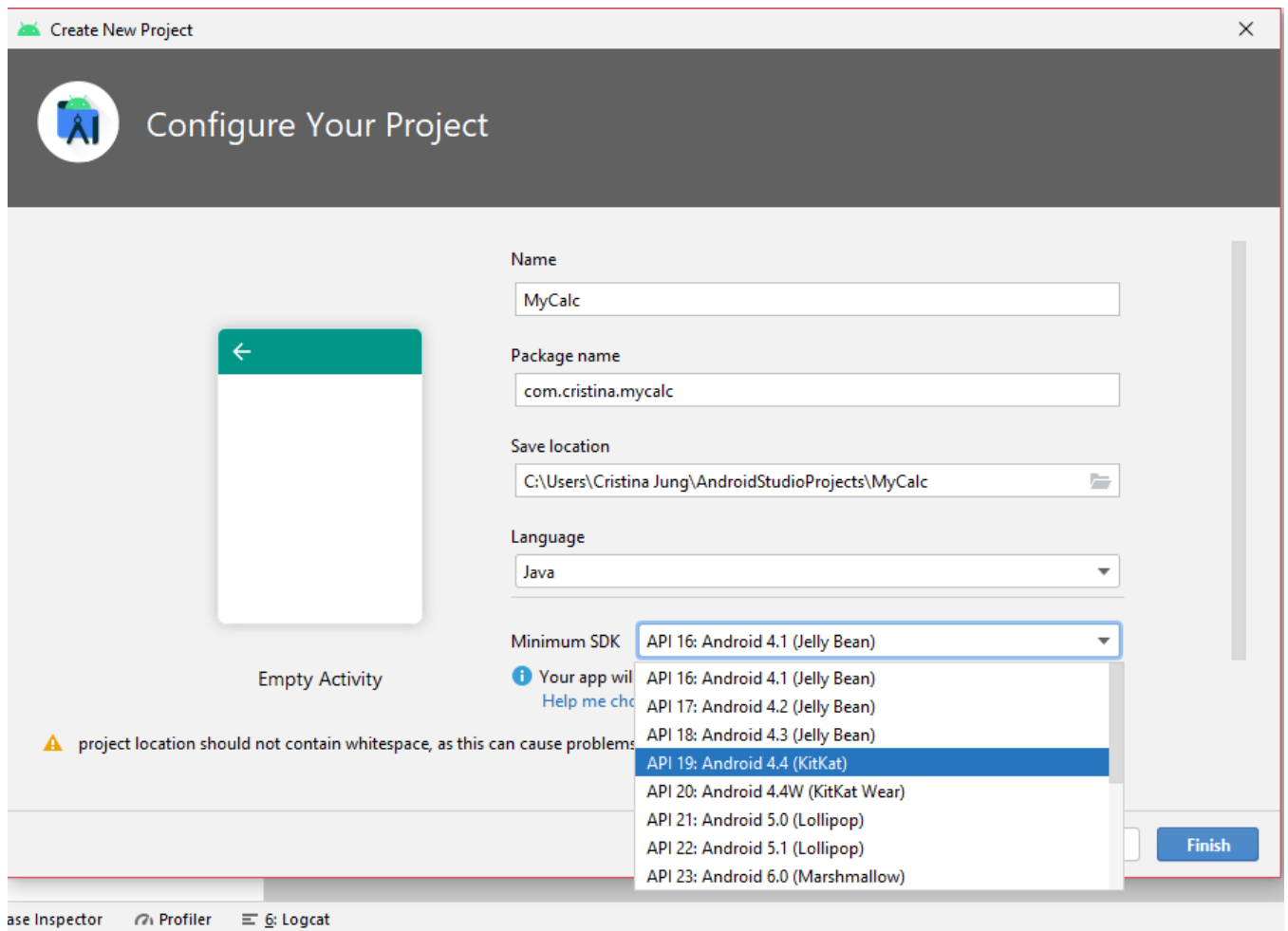
Já no **Design** é possível construir toda a estrutura visual, fazer a sua responsividade, editar e formatar componentes. Todos os componentes precisam ser nomeados, pois eles serão referenciados no código por meio do **atributo "ID"**.



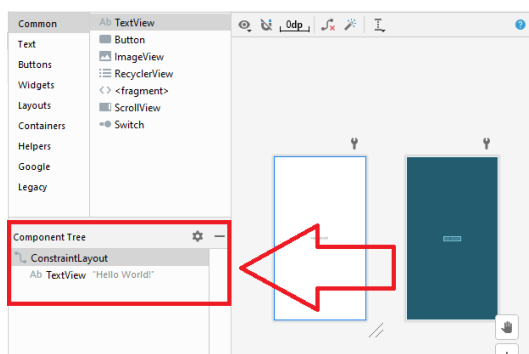
# Criando um Projeto Novo

## Layouts e Componentes de Layouts

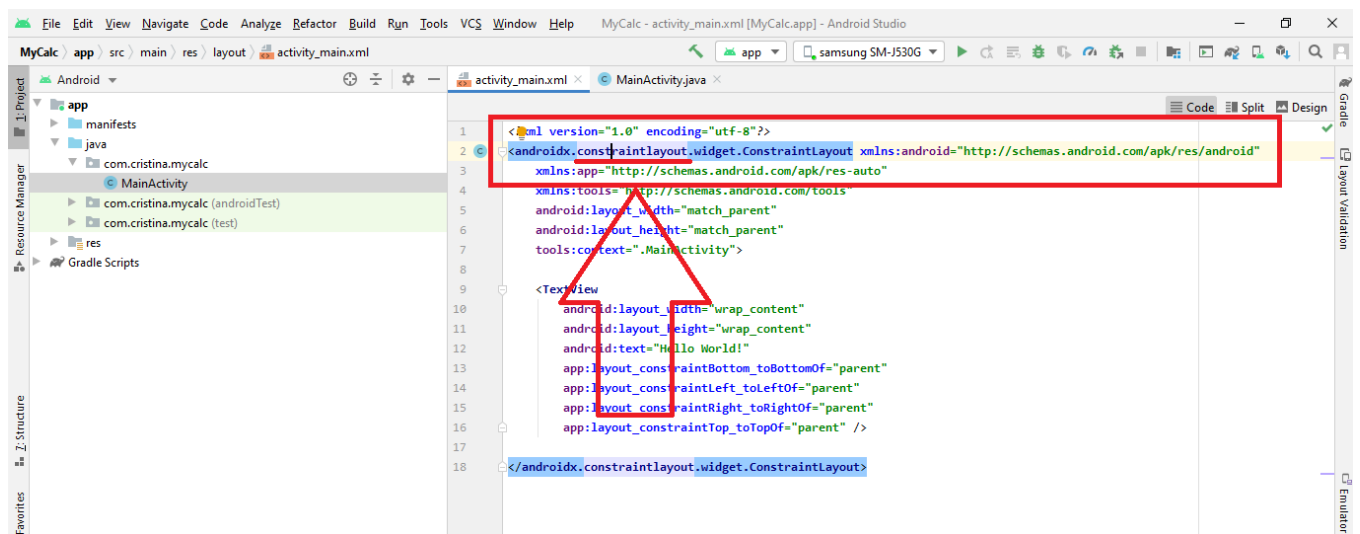
Para criar um novo projeto: **Menu File/New/Project**, em layout, vamos escolher a opção **Empty Activity**, pois iremos fazer o nosso projeto do zero. Bem importante a escolha dos nomes do projeto e de seu pacote, selecionar a linguagem Java e, na versão do Android, vamos escolher a opção **Android 4.4 - KitKat** ou o **7 também, o Nouget**.



Depois que escolhermos a definição do nosso projeto, iremos fazer uma calculadora e definir o tipo de layout que vamos usar. Ele será Linear, pois é mais fácil de adaptar do que o **ConstraintLayout**. Para que possamos trocar o tipo de layout de forma rápida e eficiente, iremos acessar o arquivo XML. Observe que, por padrão, o **AS** já vem com o **ConstraintLayout**, na paleta **Component Tree**.

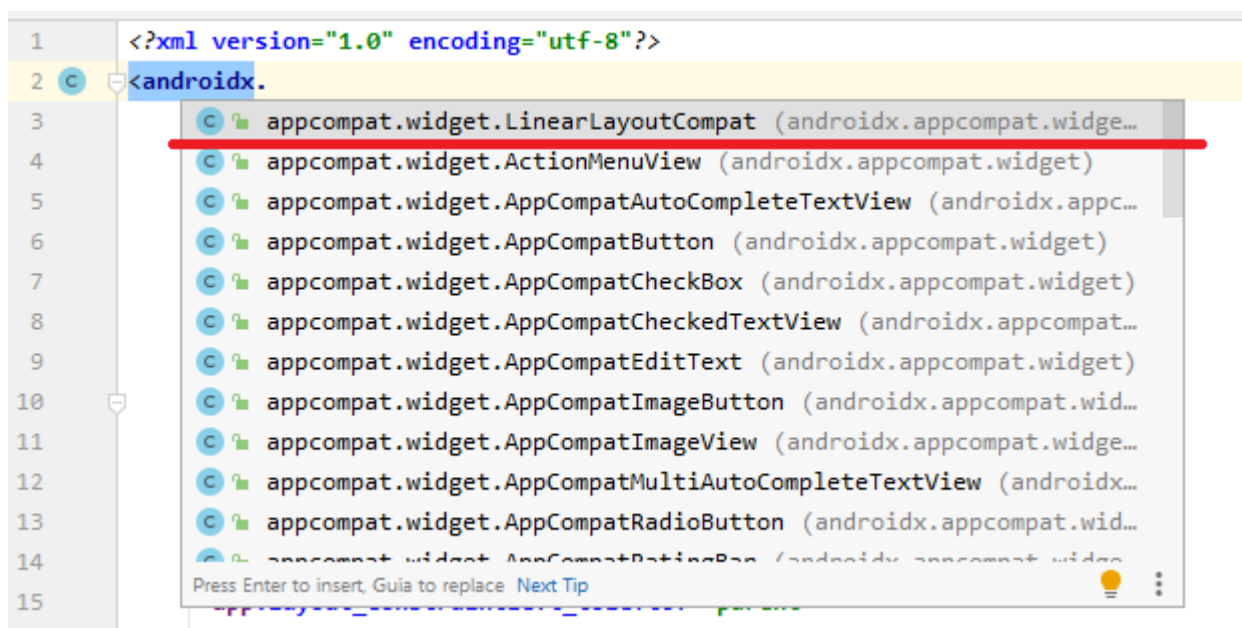


Vamos acessar o arquivo XML e:



Trocar o tipo de Layout! **Vale salientar que o arquivo activity\_main.xml se refere ao layout da nossa aplicação, observe que ele fica na pasta RES (Ressource) no lado esquerdo da tela do AS.**

Na primeira linha deste XML, encontramos a definição do layout Constraint, vamos apagar até **androidx** e digitamos linear. Vamos selecionar a opção da imagem abaixo!



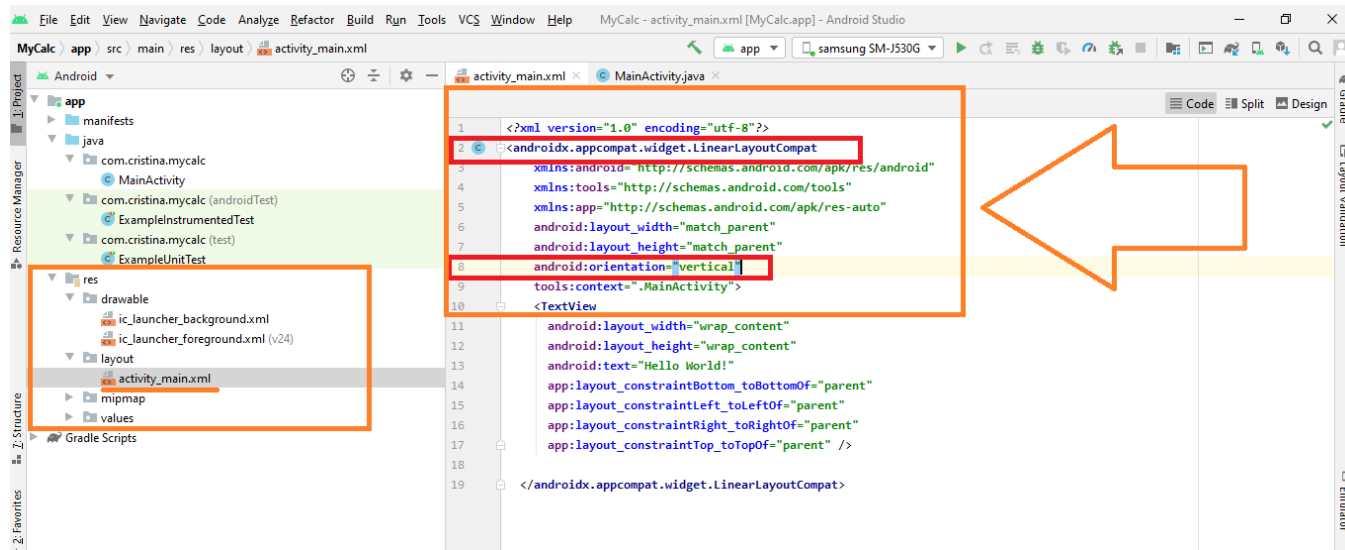
Observe como o **activity\_main.xml** ficou agora (imagem abaixo):

Toda vez que alteramos para o layout linear, precisamos fazer a configuração da orientação da nossa aplicação, neste caso, iremos optar pela vertical.

Ainda no arquivo activity\_main, no nó xml que se refere ao layout, iremos inserir:

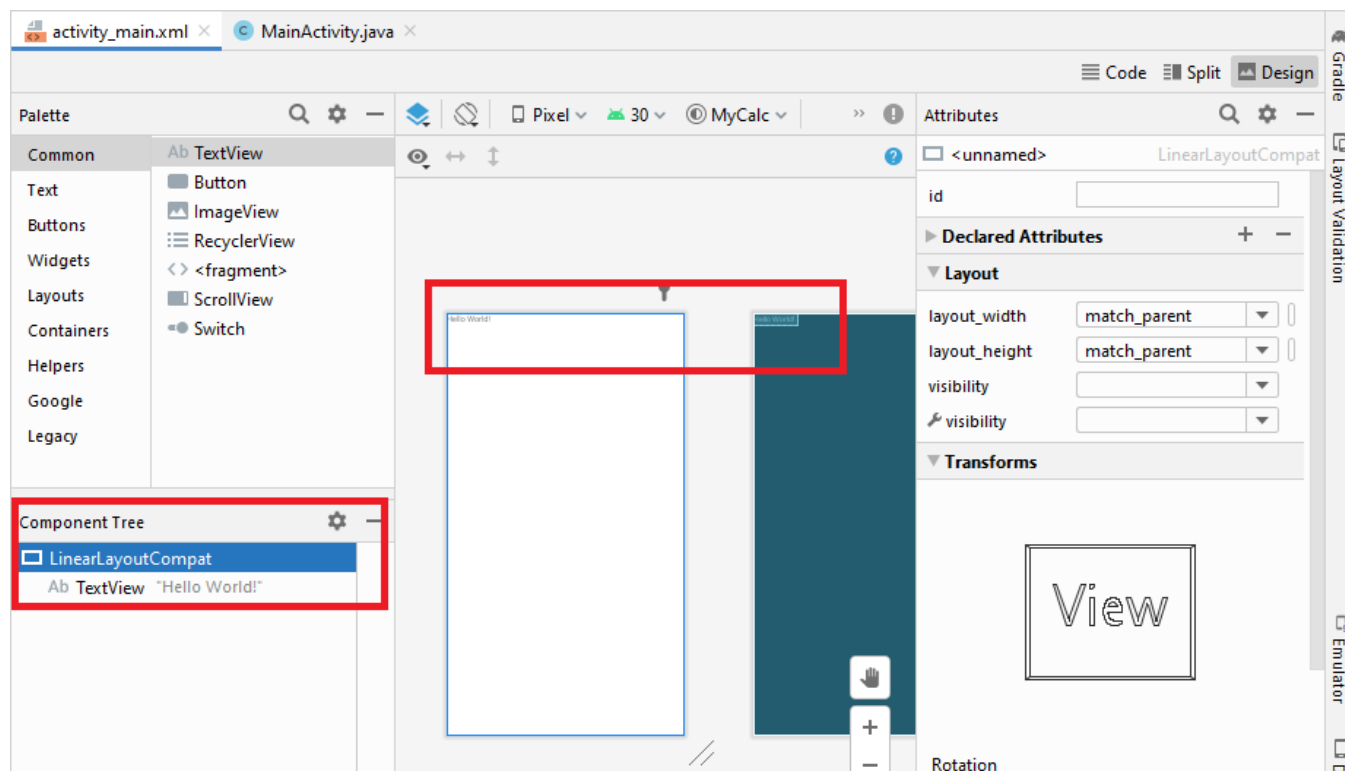
**`android:orientation="vertical"`**

Observe a imagem!



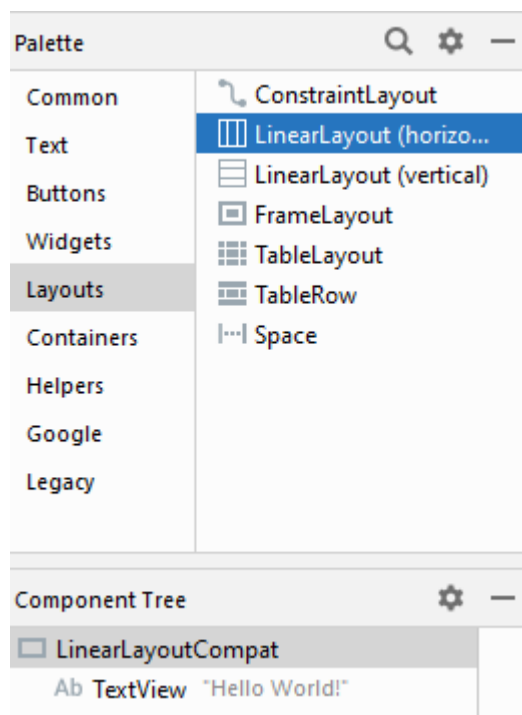
De volta ao modo de Design, vamos observar que o “Hello World” está no canto superior esquerdo e na paleta de Component Tree, o layout mudou! Agora temos Linear.

As questões do layout Linear são mais fáceis de manipular, pois o Constraint é responsivo mas opera em mapeamento de pontos para inserir os componentes. Trabalhar com o Linear é mais simplificado e podemos definir a distribuição (diagramação) em “peso”, ou seja, em quanto de espaço cada componente irá ocupar!

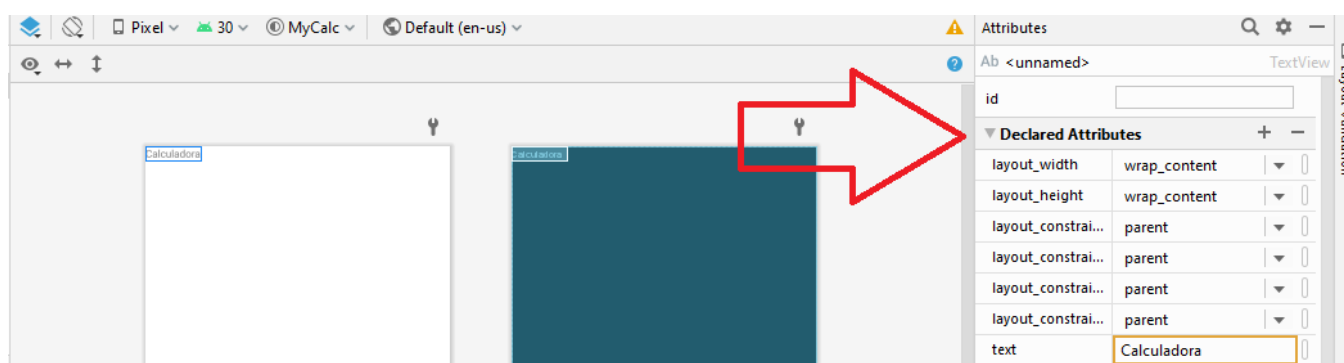




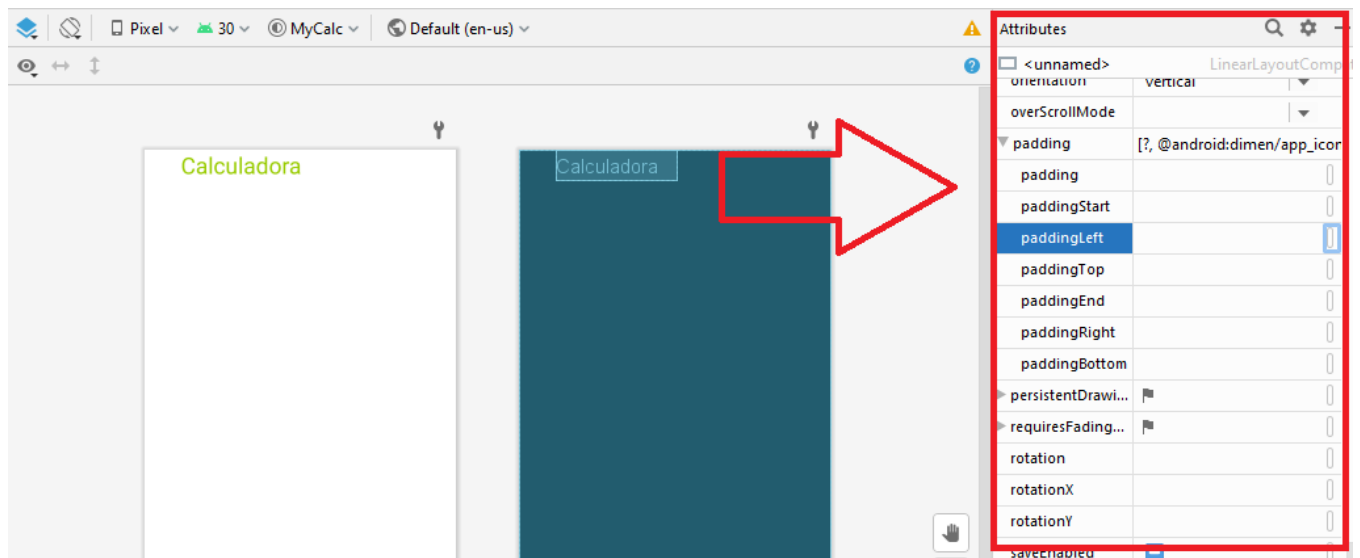
Na próxima etapa poderemos fazer as configurações do nosso layout, na Paleta, quando selecionamos a opção de layout, vários componentes aparecem para facilitar a nossa construção.



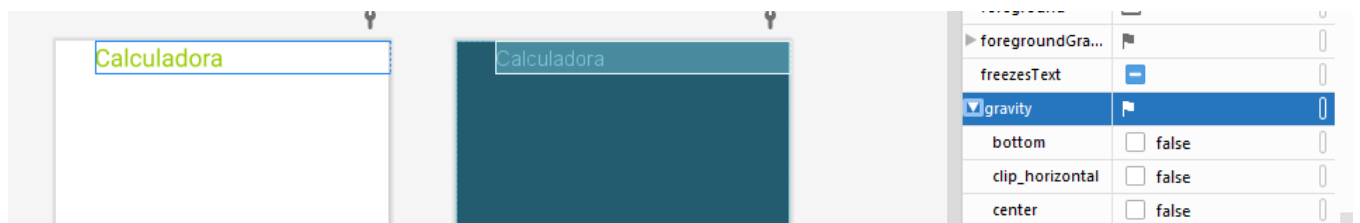
Agora iremos selecionar o texto **Hello World** e trocar por **Calculadora**, para isso, basta selecionar a caixa de texto, na paleta atributos, em **Declared Attributes**, em **text**, vamos fazer as alterações. Para fazer outras alterações, na mesma paleta ao lado direito, vamos em **All Attributes**!



Em **All Attributes** encontramos muitas opções para podermos elaborar o nosso Layout! Na imagem abaixo, o tamanho, a cor e o posicionamento do texto foram alterados!

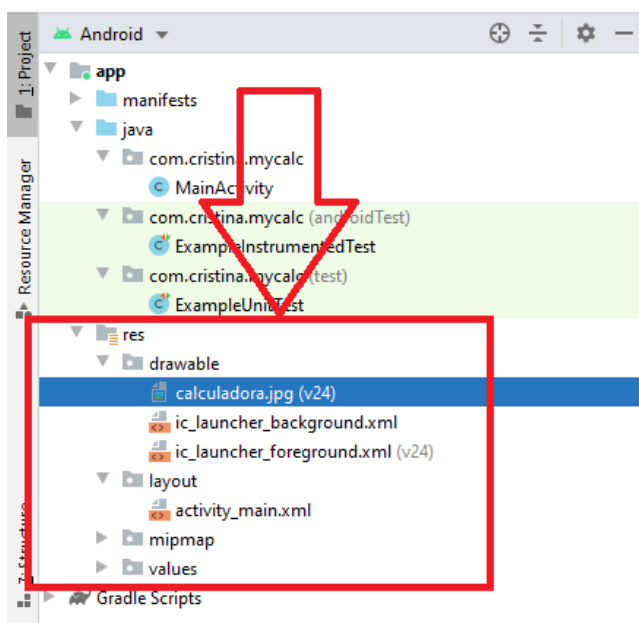


Ou utilizamos o gravity para fazer alinhamentos!

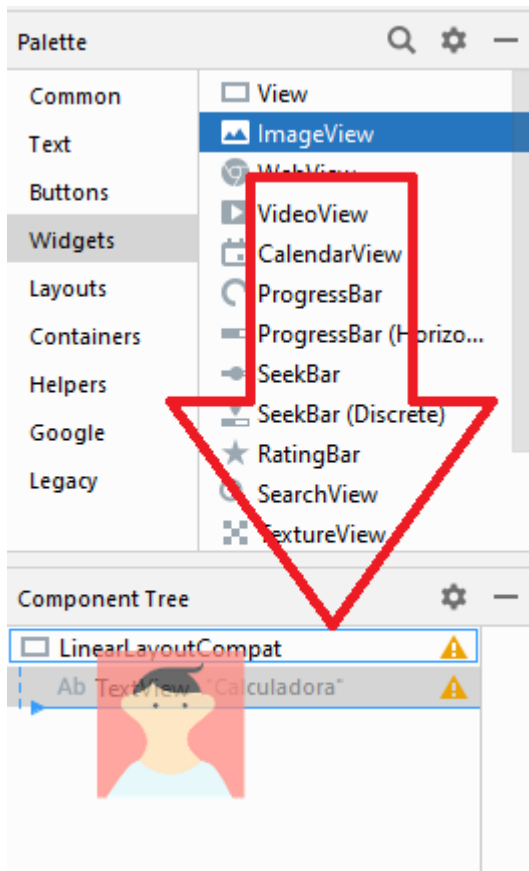


Próxima etapa do que iremos fazer, será inserir uma imagem! Como estamos fazendo de uma calculadora, iremos buscar na Internet a imagem da Calculadora. **Uma dica para vocês!** Quando salvamos a imagem da Web vamos ter o cuidado de não colocar números neste nome, existe um possível bug no AS que, quando inserimos uma imagem ele simplesmente fecha tudo.

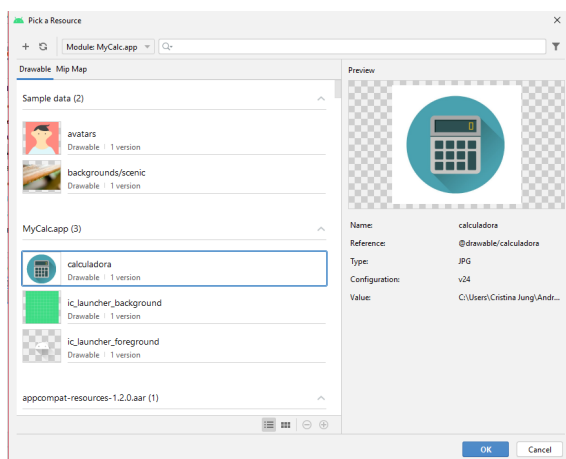
Podemos arrastar a imagem da calculadora para a pasta **RES em DRAWABLE!**



Na paleta de Componentes, vamos acessar **Widget/ImageView** e arrastar para o Layout - **Component Tree**.

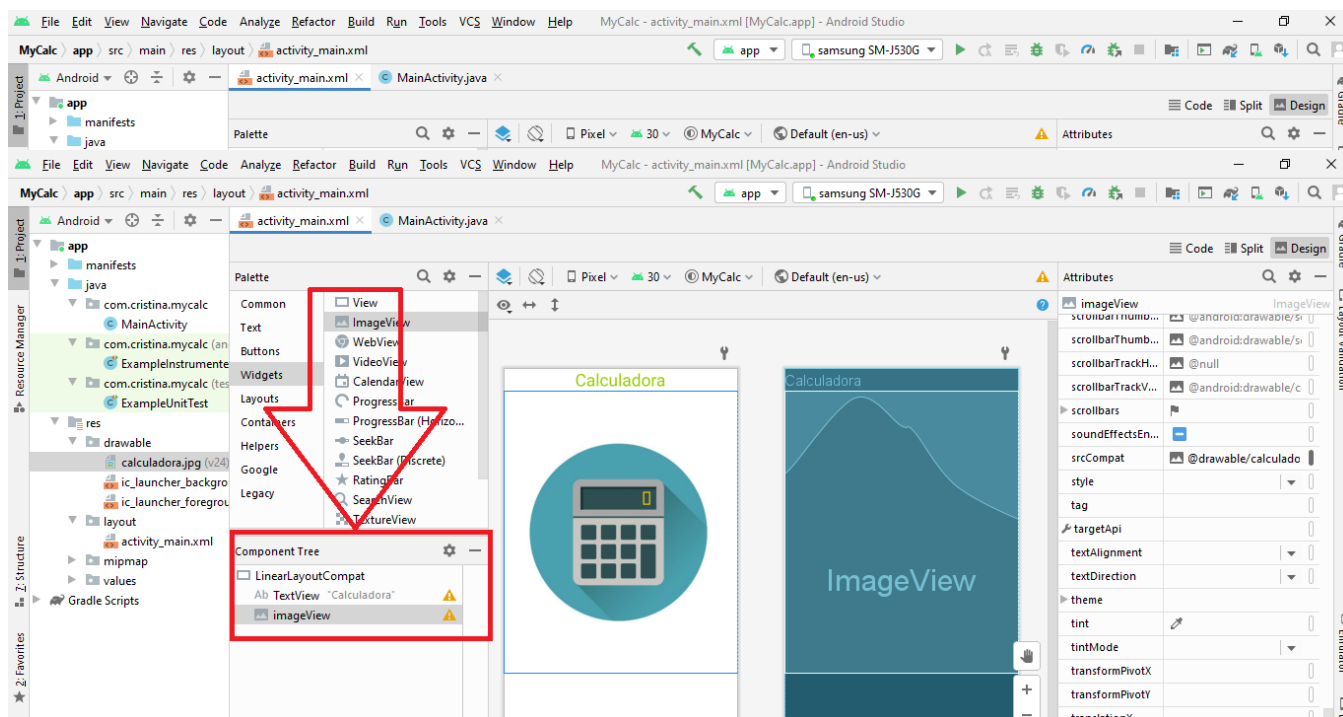


O AS irá solicitar qual a imagem que desejamos ....



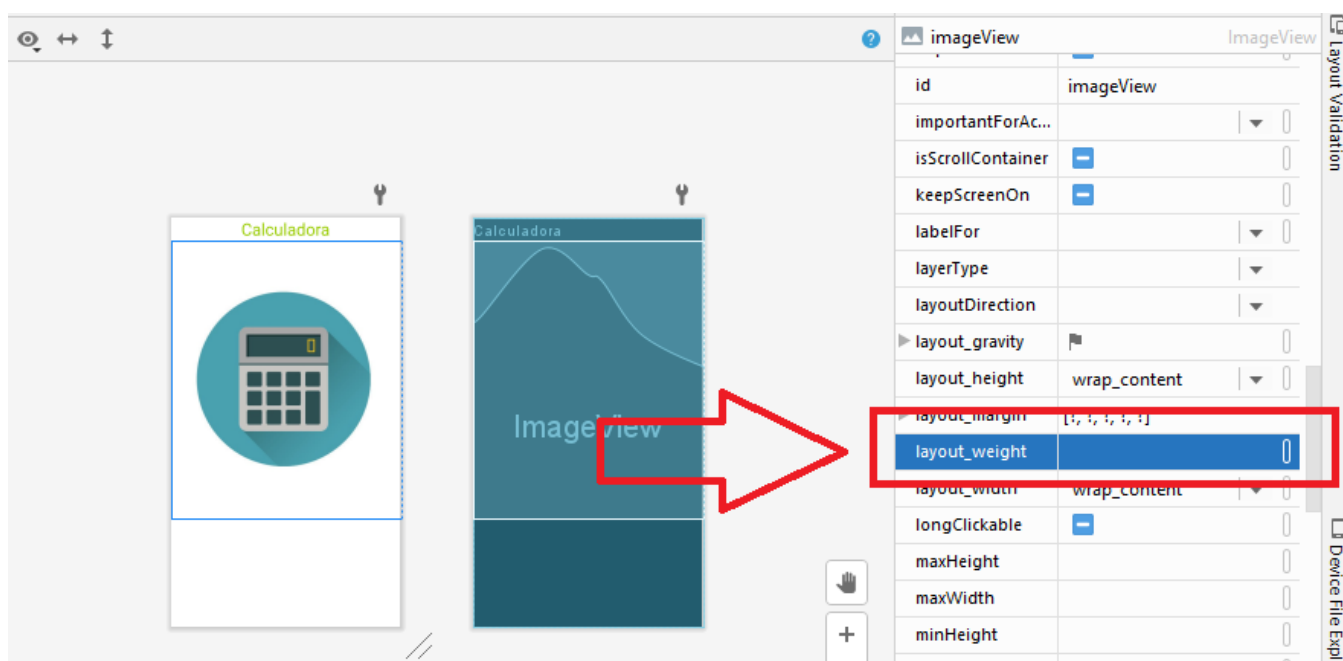
Escolhemos a imagem, no caso a **calculadora.jpg!** **Tenha sempre o cuidado de não usar caracteres maiúsculos, no máximo um “\_”**

O nosso Layout está começando a ficar dessa forma!



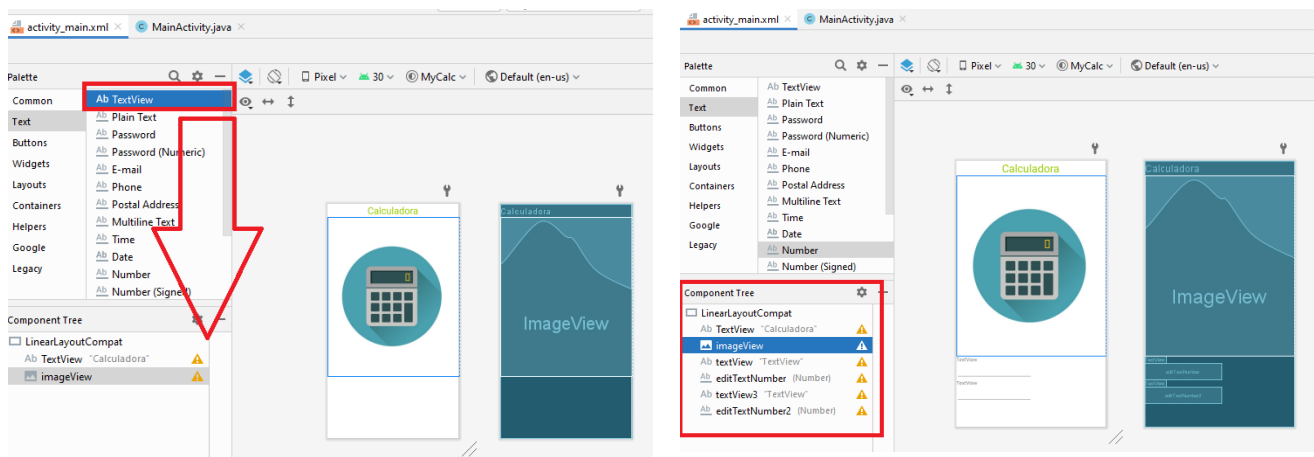
Agora iremos fazer as configurações de tamanho e proporção dos elementos e componentes que estamos inserindo no layout.

Vamos usar o atributo **Layout Weight**, **peso do layout**! É o mais adequado para o tipo de estrutura que estamos criando, pois ele trabalha com a proporção dos elementos e/ou componentes que estão sendo inseridos nele. Eu particularmente gosto muito de usá-lo pois é um recurso em que conseguimos inserir a responsividade dentro de um layout linear.

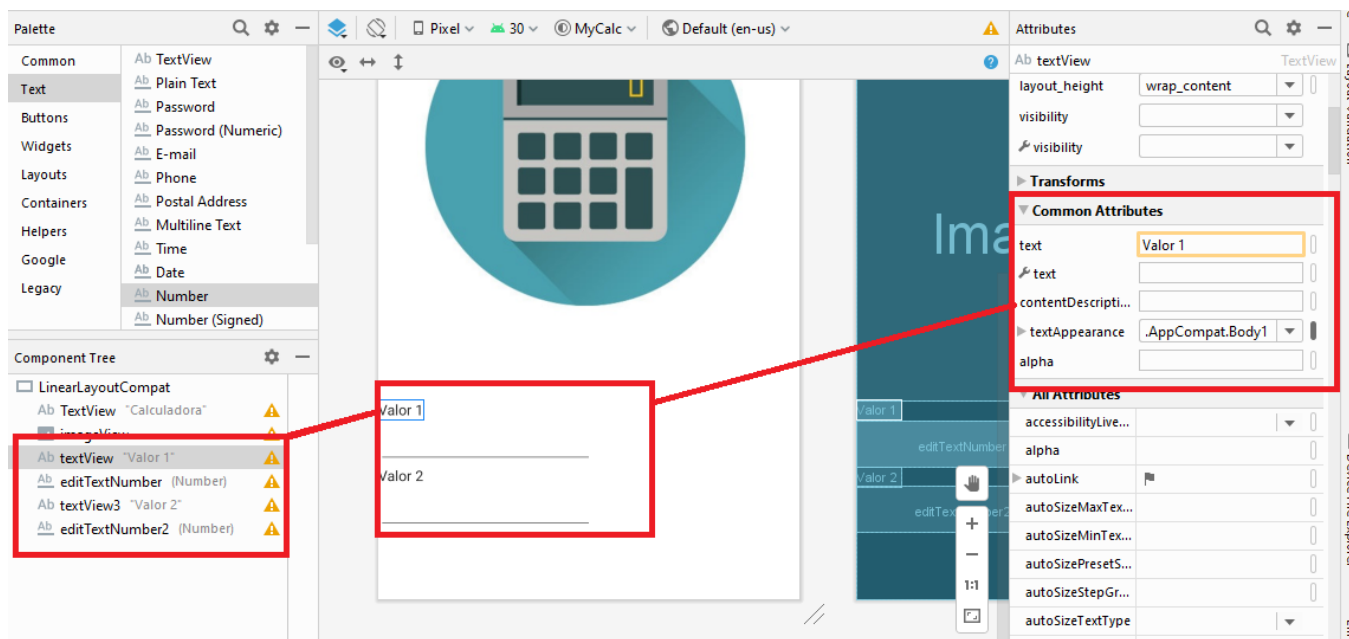


Após inserirmos todos os nossos elementos, voltaremos aqui para fazer as configurações do weight. Neste momento, precisamos inserir outros componentes para a nossa calculadora, pois até agora só configuramos o texto e inserimos uma imagem.

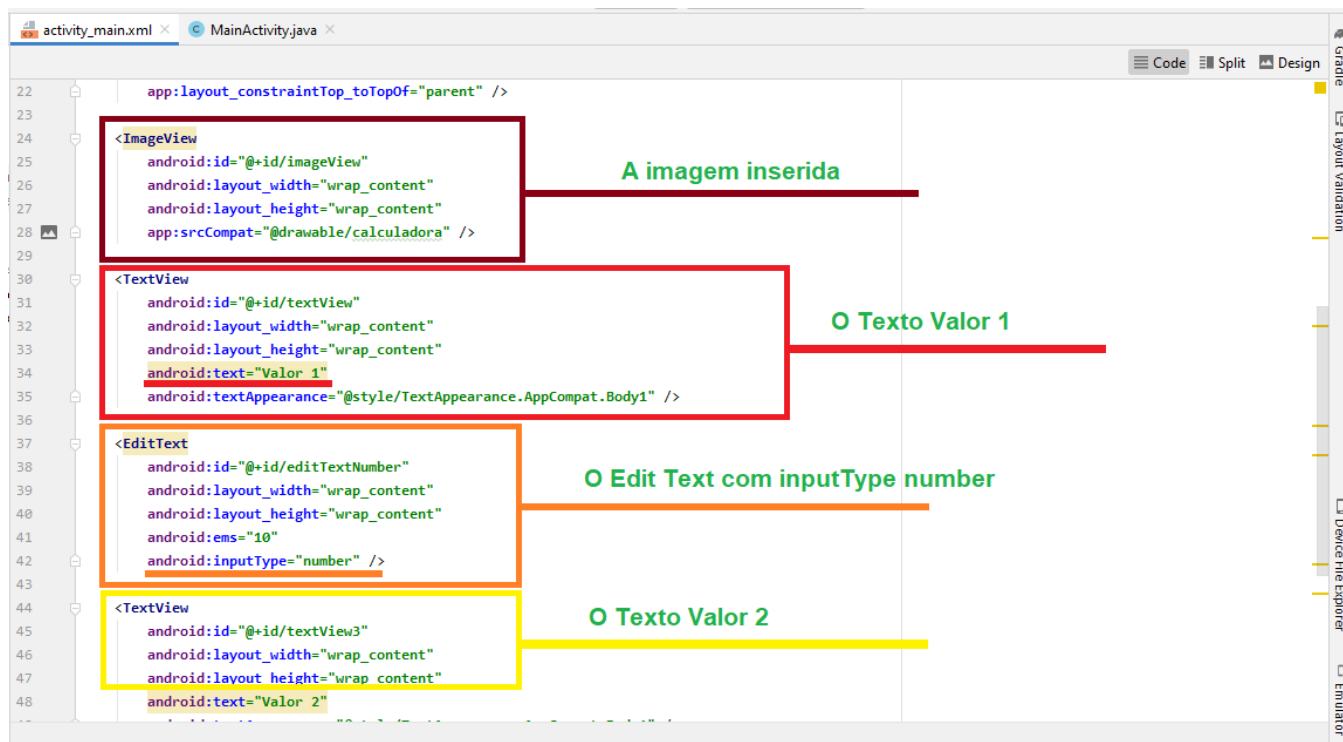
Como o nosso layout é vertical, tudo o que fomos inserindo irá ficar empilhado. Então, desta forma, vamos inserir os componentes que precisamos para que o usuário possa digitar os valores para posteriormente ele poder executar os cálculos, vamos arrastar o componente TextView e Number, a árvore de componentes ficará como nas imagens abaixo:



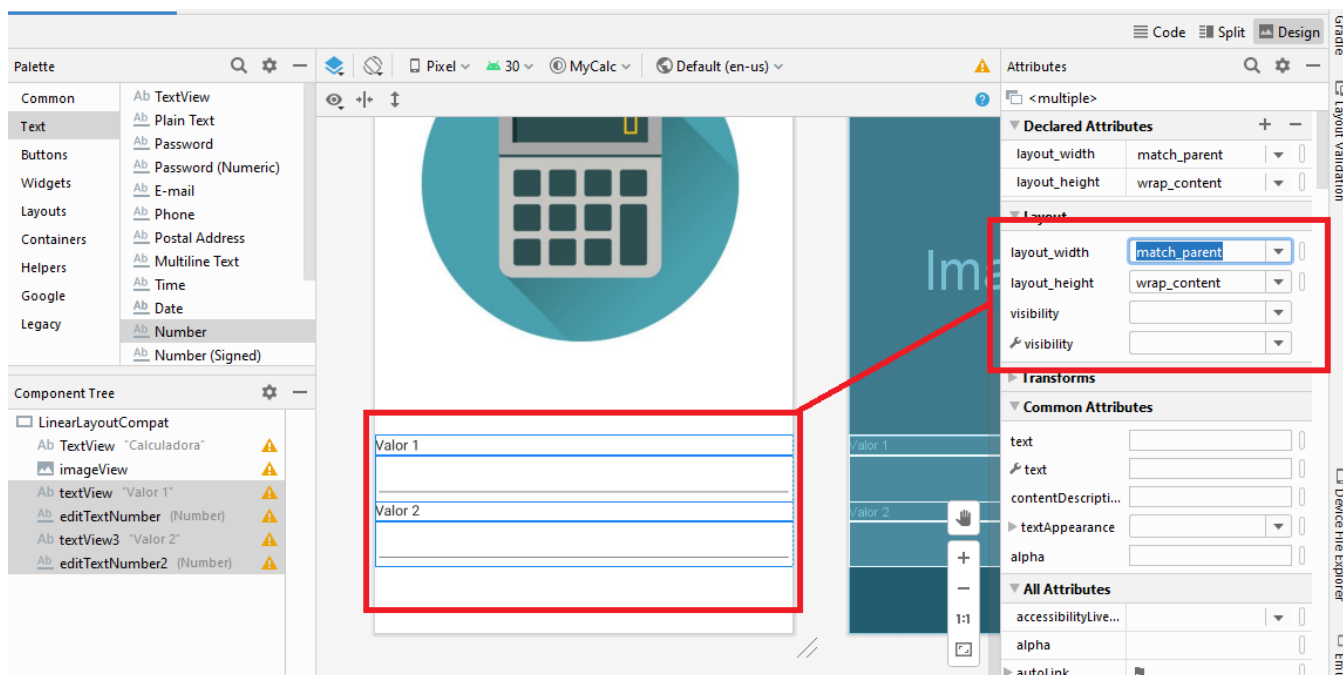
Após, podemos fazer as configurações dos textos. É muito importante salientar que aqui estamos usando a construção do layout com o assistente, também podemos fazer via código, o que, com o tempo fica mais fácil e rápido.



Vamos observar a disposição no **XML do nosso Layout**, clique sobre **Code** e observe seu código, ele estará como a imagem abaixo:



Vamos selecionar os componentes de texto e a imagem se assim desejarem, na paleta **Layout**, vamos selecionar a opção **match\_parent** em layout width. **Gravity** iremos configurar como **center\_horizontal**, desta forma nossos elementos sempre se ajustam ao tipo de dispositivo.

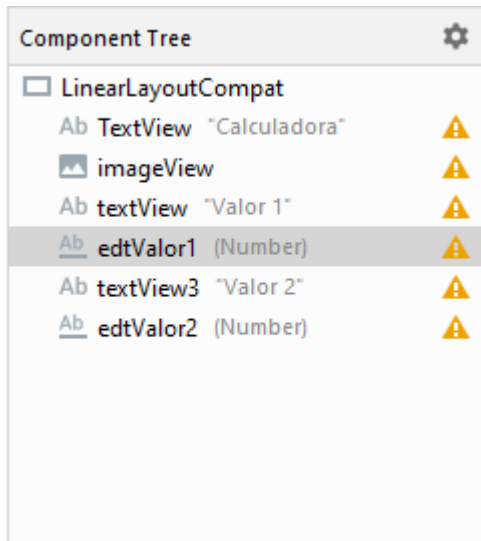


Após esta configuração, podemos fazer o afastamento do layout das bordas, para isto, basta selecionarmos o atributo **layout\_margin** e setar as configurações para **20dp layout\_marginRight** e **layout\_marginLeft**

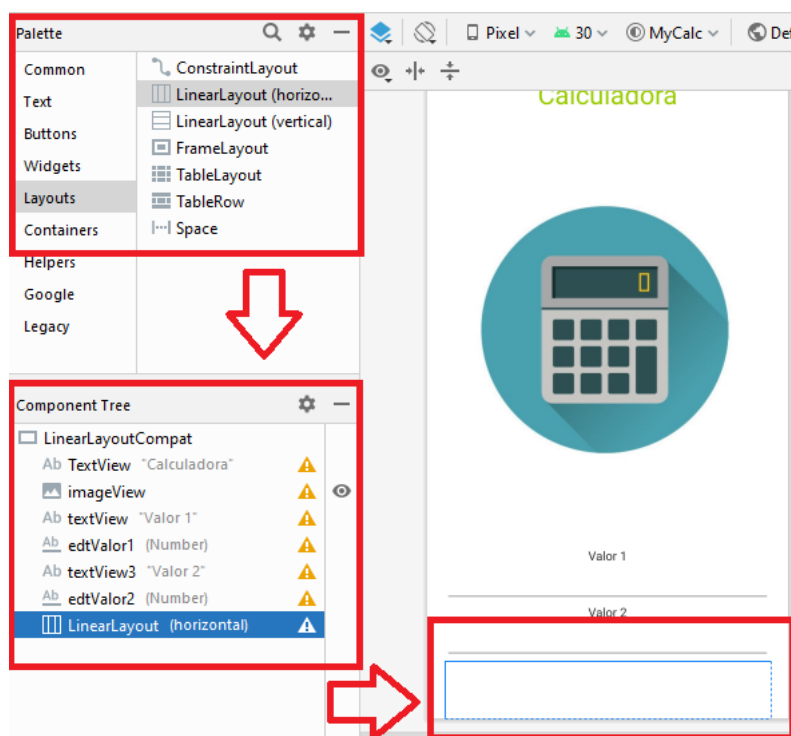
A próxima etapa, será fazer a configuração do ID das entradas de valores. A identificação que daremos é muito importante, pois ela será a referência do elemento quando iremos inserir a programação nele.

1. **Selecionamos o primeiro editText;**
2. **Na paleta, vamos em ID;**
3. **Digitamos: `edtValor1`.**

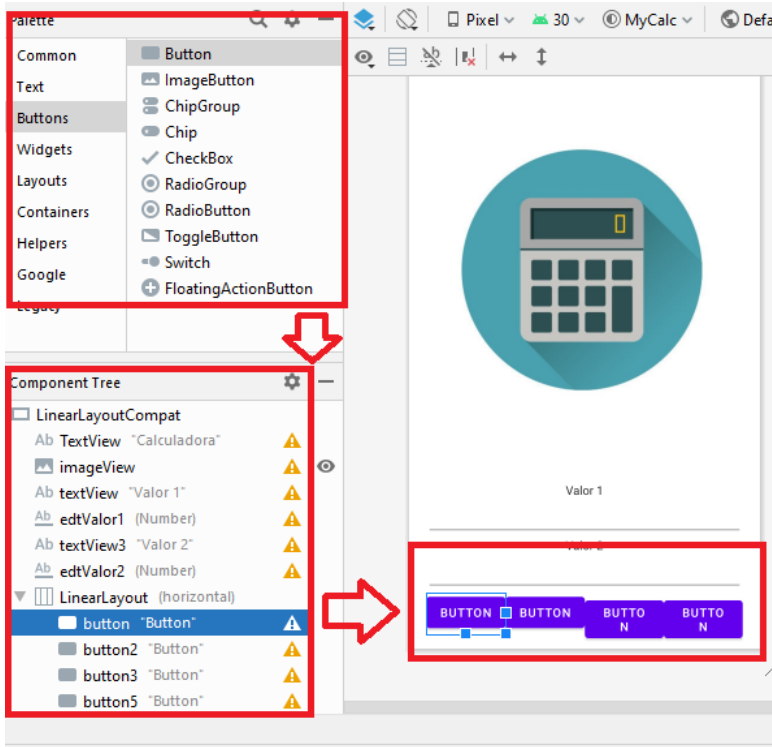
Repetimos a mesma operação para o `editText2` e vamos inserir o ID como `edtValor2`;



Vamos inserir os botões para cálculo e desejamos colocar um ao lado do outro. Como fazer se estamos em um layout linear? Este tipo de layout empilha os elementos ..... desta feita, iremos inserir elementos que nos permite colocar os botões um ao lado do outro.

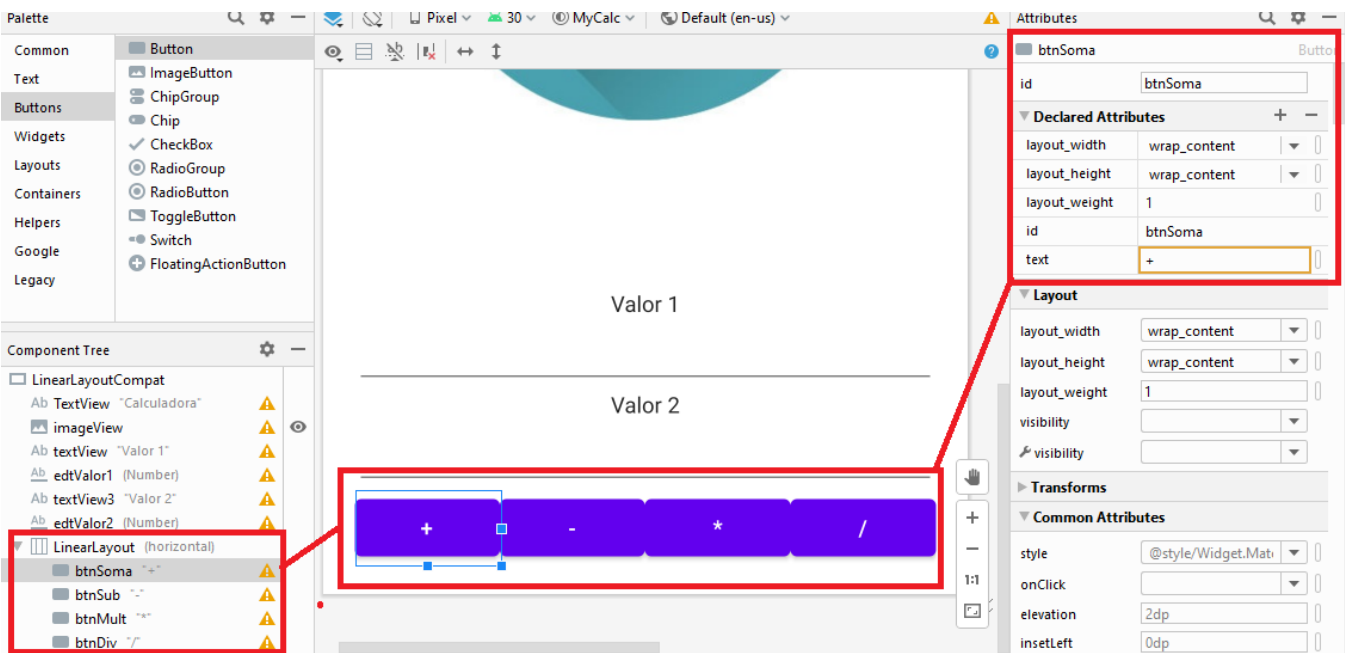


Agora basta inserirmos os botões para dentro dele .....



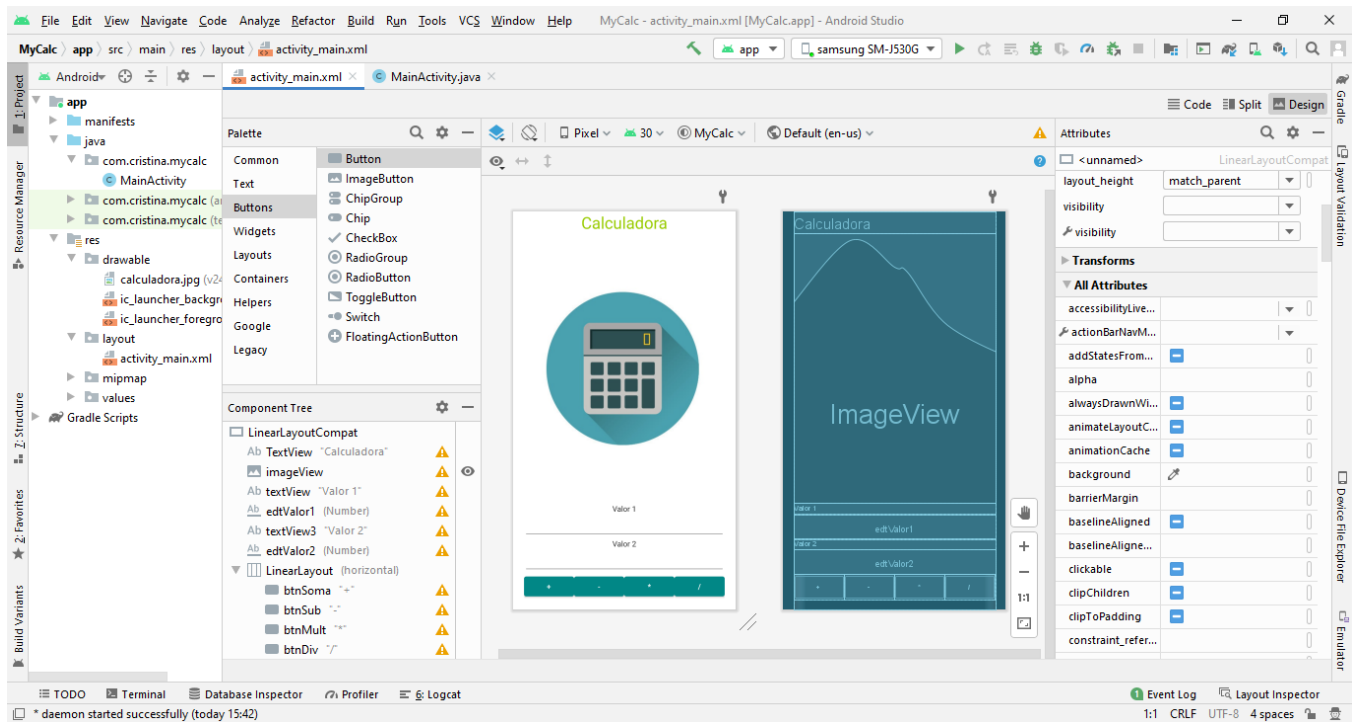
E vamos configurar o nome destes botões! Para facilitar a nossa vida neste momento, eu coloquei os nomes dos botões da seguinte forma:

Botão Soma	Text = "+"	ID = btnSoma
Botão Subtração	Text = "-"	ID = btnSub
Botão Multiplicação	Text = "*"	ID = btnMulti
Botão Divisão	Text = "/"	ID = btnDiv





## Layout até aqui da Calculadora



Se tentarmos executar a nossa aplicação ela ainda não irá funcionar pois ainda não temos a programação dela, mas podemos ver a construção do Gradle, lembrando que ele vai gerenciar a compilação, mas retorna falha.

