

FUNDAÇÃO INSTITUTO DE EDUCAÇÃO DE BARUERI

Unidade “Brasílio Flores de Azevedo”

PDMO (Programação para Dispositivos Móveis)

Aula01-1
if / else
try / catch

Prof Leomar Duarte

Estrutura Condicional

Permite que a execução de uma rotina ou bloco de instruções, seja executado após o resultado de uma avaliação condicional. Os comandos condicionais são if-else e switch-case.

Estrutura if

```
if (condição) {  
<Execução para condição verdadeira>  
}
```

Estrutura if-else

```
if (condição) {  
<Execução para condição verdadeira>  
} else {  
<Execução para condição falsa>  
}
```

Estrutura else if()

```
if (condição) {  
<Execução para condição verdadeira>  
}  
else if(condição){  
<Execução para condição verdadeira>  
}
```

Abra um projeto novo no Android Studio, clicar em New Project... e selecione o **Empty View Activity**

Coloque o **Name**: Testando_if_else. O Package name deixar o padrão ou utilizar o mesmo abaixo.

O local de **Save location**: Escolha o caminho que você definiu.

O importante é que o caminho em Save Location deve estar correto sendo: ...caminho/Testando_if_else.

A versão que utilizei foi a API 24. **Verifique a API mais apropriada**. Clicar em Finish

The screenshot shows the 'New Project' dialog in Android Studio. The title is 'New Project'. Below it, the selected option is 'Empty Views Activity'. A description says 'Creates a new empty activity'. The configuration fields are as follows:

- Name**: Testando_if_else
- Package name**: com.fieb.aula.testando_if_else
- Save location**: E:\pdm0-codigos\Testando_if_else
- Language**: Java
- Minimum SDK**: API 24 ("Nougat"; Android 7.0)

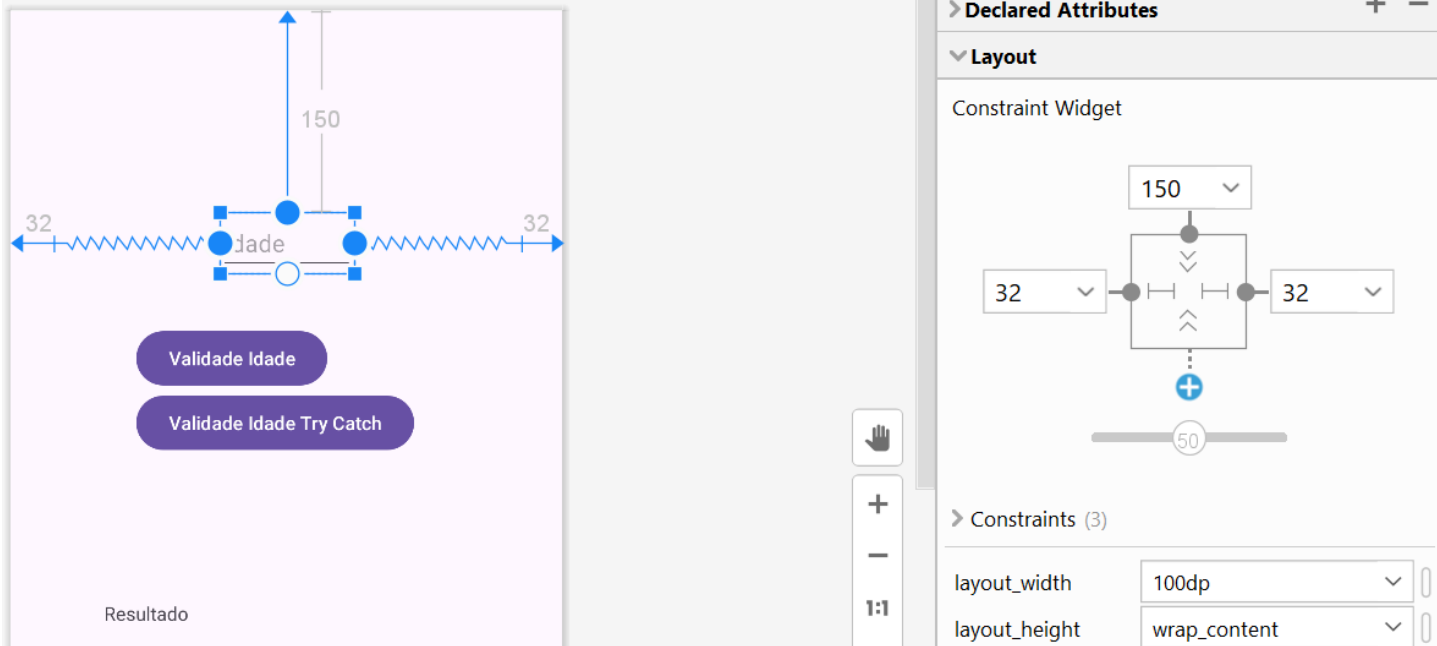
Below these fields, there is an information icon and text: 'Your app will run on approximately 96,3% of devices.' with a link 'Help me choose'.

At the bottom, there is a field for 'Build configuration language' with a question mark icon, set to 'Kotlin DSL (build.gradle.kts) [Recommended]'.

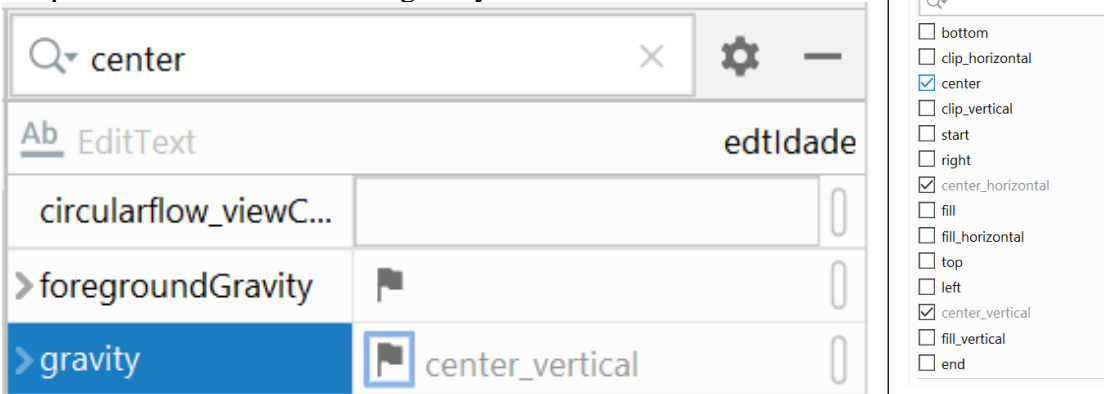
1. Após abrir, clique na aba: activity_main.xml, clique no objeto Hello World e apague.



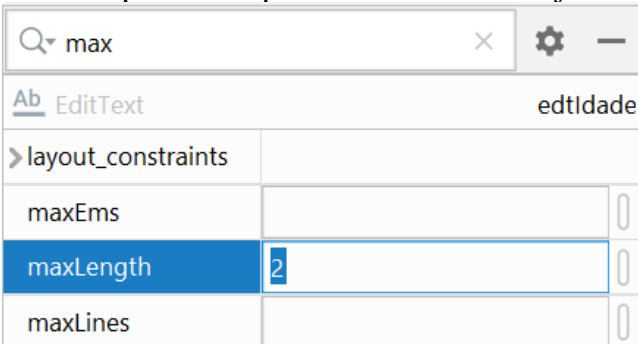
- 2.
3. Clique na idade inclua os valores indicados



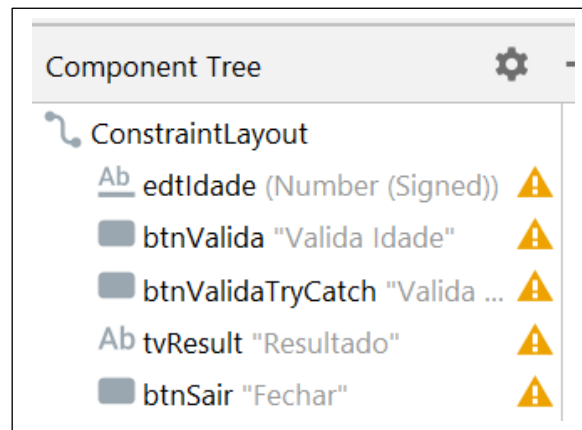
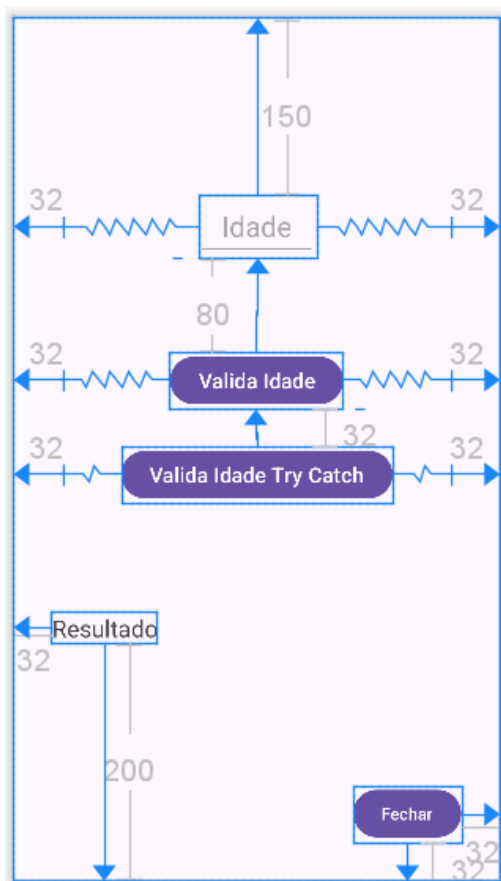
- 4.
5. Para centralizar a Idade faça pesquisa por: center no objeto edtIdade
6. Clique na bandeira do atributo gravity e selecione somente **center**



- 7.
8. Altere a quantidade para 2 caracteres no objeto Idade, procure por max e altere o **maxLength** para 2



- 9.
10. Altere o textSize dos objetos:
 - a. **Idade**: 24dp
 - b. Os **botões e resultado** para 18dp
11. Proposta da Tela: Caso a tela fique diferente por causa do tipo de celular, altere os espaços



12.

C MainActivity.java

13. Clicar no **MainActivity.java**, para obter o código Java

14. Crie as variáveis a partir da class MainActivity para ser global (todos acessem)

```
public class MainActivity extends AppCompatActivity {
```

1 usage

```
EditText edtIdade;
```

1 usage

```
Button btnValida, btnValidaTryCatch, btnSair;
```

1 usage

```
TextView tvResult;
```

15.

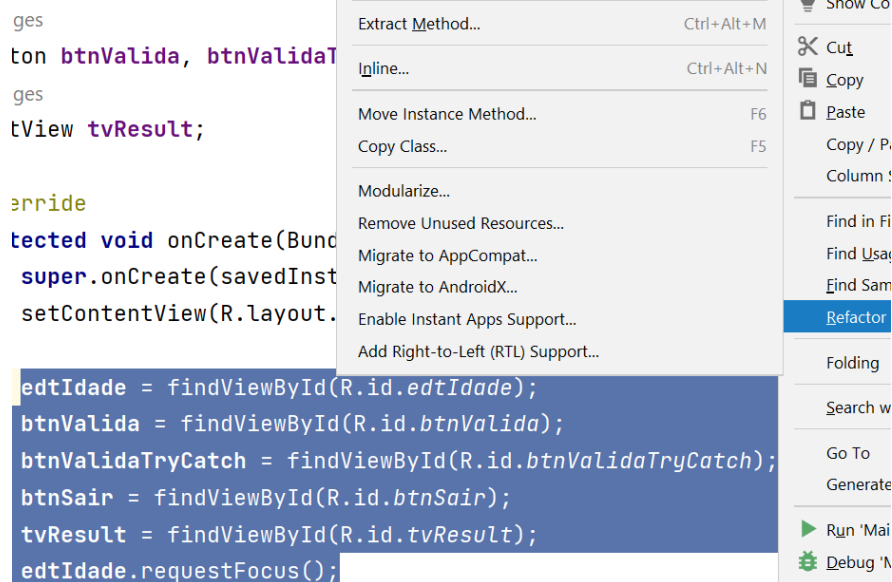
16. Atribua um “link” de acesso entre os objetos e a classe java.

17. Vamos gerar um método interno para modularizar o código (código limpo).

a. Para isso marcar as linhas para a extração e clicar com o botão direito → Refactor → Extract Method...

b. Ou após marcar as linhas usar o atalho: Ctrl + Alt + M

c. Figura a seguir:



18. `edtIdade.requestFocus();`

19. Digite o nome inicializarComponentes e enter:

```
extracted ();
```

```
inicializarComponentes();  
}
```

20. `private void extracted() {` → Ficar assim →

```
1 usage  
private void inicializarComponentes() {  
    edtIdade = findViewById(R.id.edtIdade);
```

21. Assim quando for “lido” o método: inicializarComponentes será executado tudo que está dentro dele.

22. Ativar o botão **btnValida**:

23. Comece digitando e assim que ver o código do autocompletar abaixo de o <enter>:

```
btnValida.setOnCli
```

24. `setOnClickListener(OnClickListener l)`

25. Depois complete digitando new OnClickListener... Assim que aparecer a imagem abaixo de o <enter>

26. Não Digite Tudo

```
inicializarComponentes();
```

```
View.OnClickListener{...} (android.view.  
Ctrl+Abaixo and Ctrl+Acima will move caret down and up in the editor. Next Tip
```

27. `btnValida.setOnClickListener(new OnClickListener{`

28. Segue o código completo:

```
btnValida.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        if (edtIdade.getText().toString().isEmpty()) {  
            edtIdade.setError("Insira a idade");  
            tvResult.setText("");  
            edtIdade.requestFocus();  
        } else {  
            int idade = Integer.parseInt(edtIdade.getText().toString());  
            if (idade < 18) {  
                tvResult.setText("Você tem " + idade +  
                                " anos \nEntão é menor de idade");  
                edtIdade.requestFocus();  
            } else {  
                tvResult.setText("Você tem " + idade +  
                                " anos \nEntão é maior de idade");  
                edtIdade.requestFocus();  
            }  
        }  
    }  
});
```

29. `});`

30. Ativar o botão **btnValidaTryCatch** → Use os mesmos procedimentos dos itens acima

```
btnValidaTryCatch.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        try {  
            int idade=Integer.parseInt(edtIdade.getText().toString());  
            if(idade<18){  
                tvResult.setText("Você tem "+idade  
                                + " anos\nEntão é menor de idade!");  
            }else if(idade>=18){  
                tvResult.setText("Você tem "+idade  
                                + " anos\nEntão é maior de idade!");  
            }  
        }  
        catch (Exception err){  
            edtIdade.setError("Informe a idade");  
            edtIdade.requestFocus();  
        }  
    }  
});
```

31. `});`

32. Botão Sair

```
btnSair.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        finish();  
    }  
});
```

33. });

Try/Catch

Estrutura de controle para exceções de erros.

Quando utilizamos o bloco de código dentro de um try{ }, está sendo informado ao programa: “Tente executar este código e caso ocorra algum erro, faça algo”.

O bloco catch{ } é onde você especifica o que fazer quando uma exceção é encontrada.

Segue a proposta da tela

The image displays four sequential screenshots of an Android application interface, illustrating the implementation of age validation using Try/Catch blocks. Each screen features a text input field labeled 'Idade', two purple buttons labeled 'Valida Idade' and 'Valida Idade Try Catch', a 'Resultado' label, and a 'Fechar' button at the bottom.

- Screen 1:** The 'Idade' field is empty.
- Screen 2:** The 'Idade' field contains the text 'Insira a idade' with a red error indicator (a small red circle with an exclamation mark) above it.
- Screen 3:** The 'Idade' field contains the value '16'. Below the buttons, the text 'Você tem 16 anos' and 'Então é menor de idade' is displayed.
- Screen 4:** The 'Idade' field contains the value '99'. Below the buttons, the text 'Você tem 99 anos' and 'Então é maior de idade' is displayed.

Exercício

Como exercício inclua:

- Botão para limpar os campos: **Idade** e **Resultado**
- Coloque o foco em idade