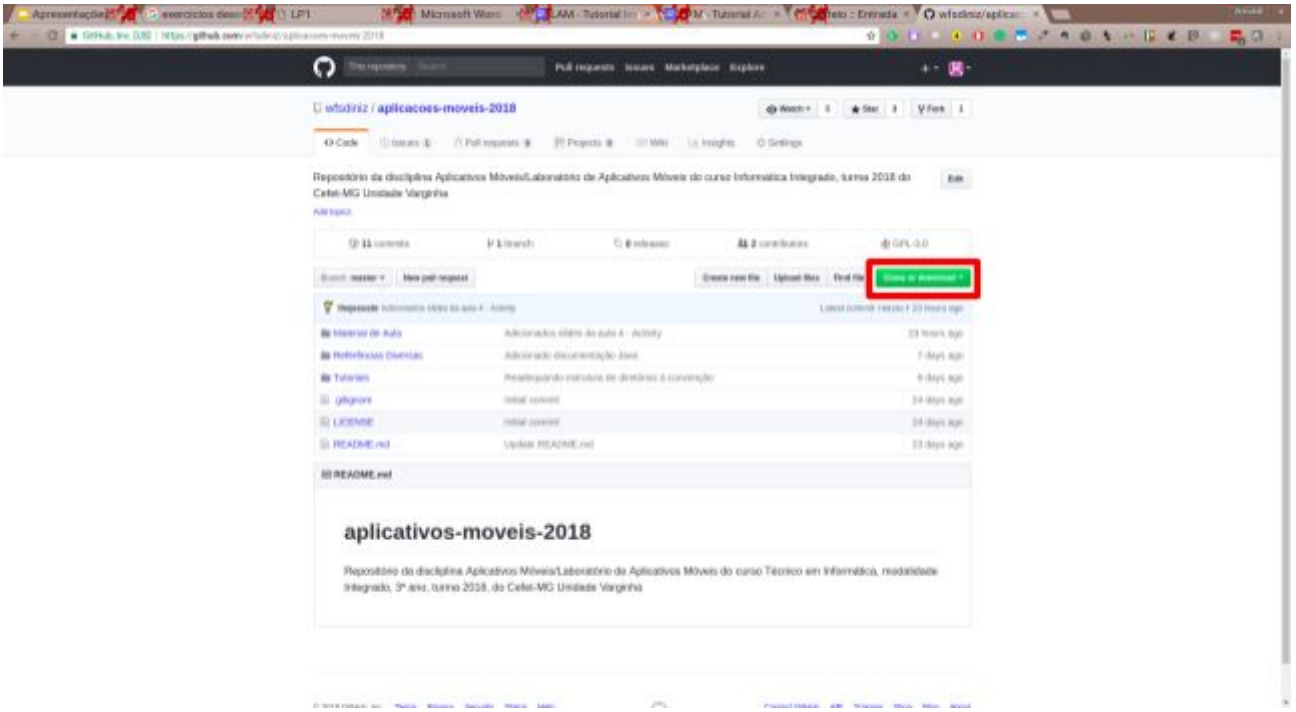
CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
CURSO: Técnico em Informática Integrado - 3º Ano
DISCIPLINA: Aplicações Móveis / Laboratório de Aplicações Móveis

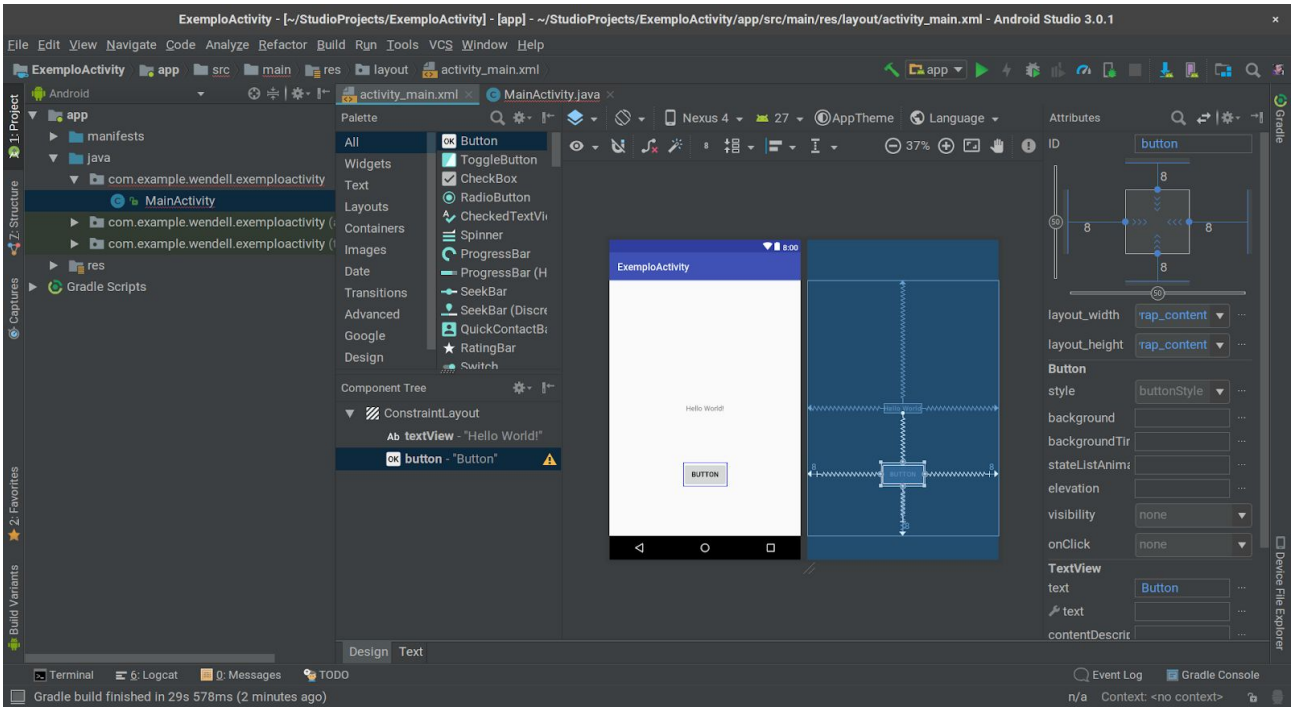
PROFESSOR: Wendell Fioravante da Silva Diniz

Tutorial Activity e eventos

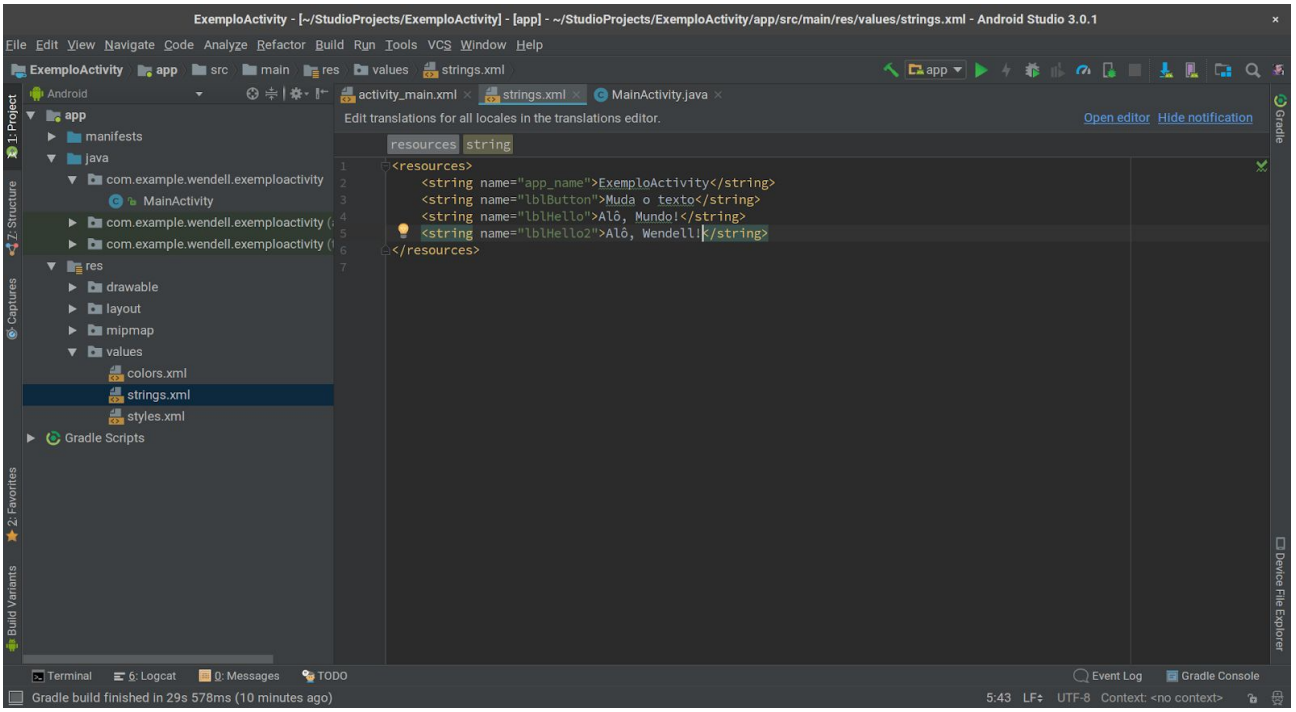
Na última aula, vimos as principais características de uma Activity e também como responder a eventos do usuário. Vamos primeiro praticar implementando o exemplo visto na sala de aula.
Comece clonando o seu repositório para podermos trabalhar nele. Visite-o no Github e copie o link do repositório, que pode ser encontrado na seção *Source*.



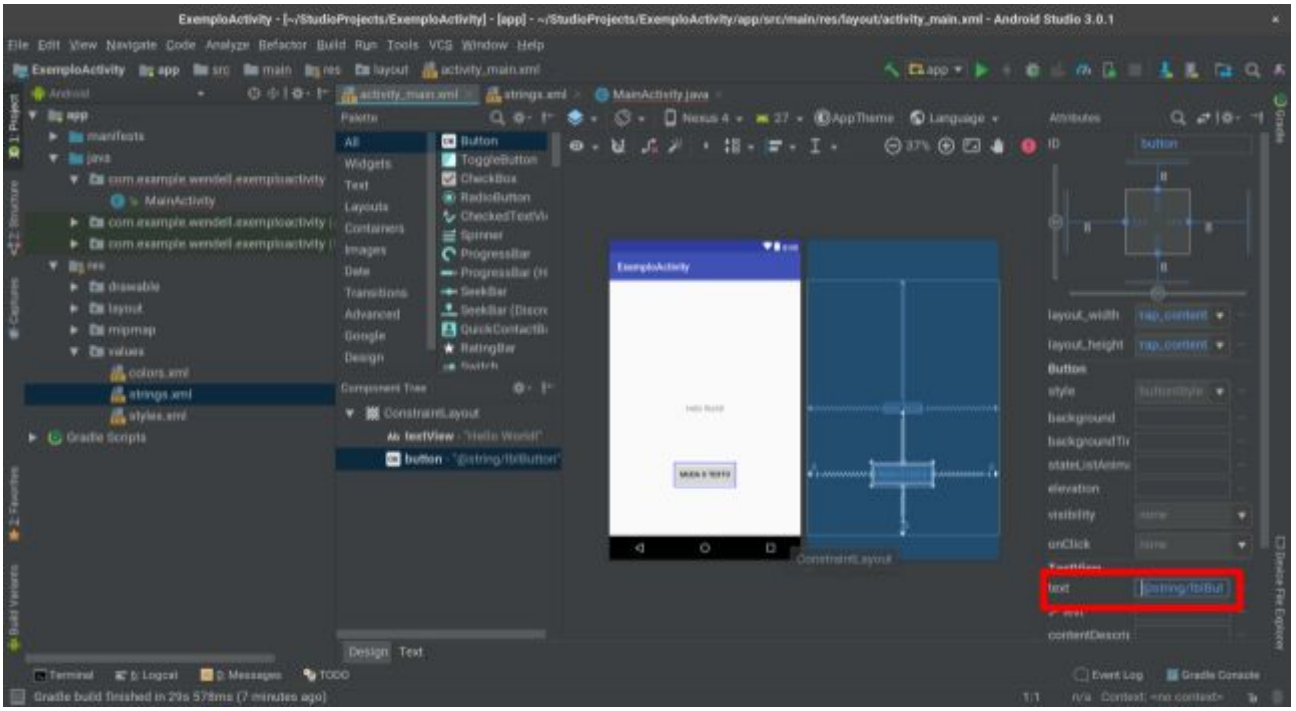
Depois, abra um terminal e insira o comando:
`git clone cole_seu_link_aqui`
Navegue para o diretório principal de seu repositório e crie uma nova pasta chamada **ExemploActivity.**
Agora, abra o Android Studio e inicie um novo projeto, chamando-o de *ExemploActivity*. No campo *Project Location*, mude para o diretório que você acabou de criar. Complete o wizard escolhendo o nível da API e criando uma *Empty Activity*.
Mude para a tela de design de interface e acrescente um botão.



Agora, abra o arquivo strings.xml, na pasta *res/values* que pode ser encontrada no navegador de projeto à esquerda do designer. Crie três novas entradas de string, uma com a chave *lblButton* e com o valor *Mudar o texto,* outra com a chave *lblHello* e valor *Alô, Mundo!* e outra com a chave *lblHello2* e o valor *Alô, <seu nome>!*



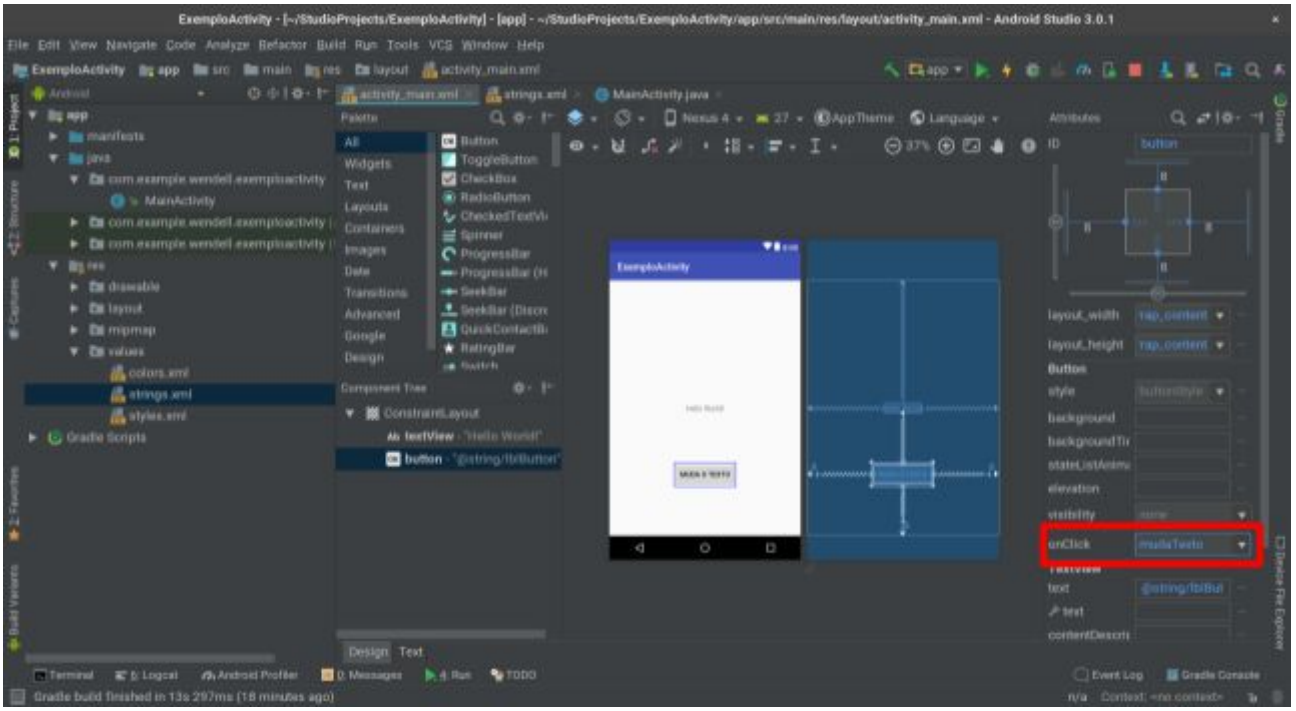
Agora, no designer, mude a propriedade *text* do botão usando o identificador da string que criamos, usando o seletor: `@string/lblButton`.



Mude também o valor do `textView` para a string *lblHello*.
Agora vamos acrescentar o código fonte com o método que vai substituir as strings do `textView`:

```
1  package com.example.wendell.exemploactivity;
2
3  import android.support.v7.app.AppCompatActivity;
4  import android.os.Bundle;
5  import android.view.View;
6  import android.widget.TextView;
7
8  public class MainActivity extends AppCompatActivity {
9
10     @Override
11     protected void onCreate(Bundle savedInstanceState) {
12         super.onCreate(savedInstanceState);
13         setContentView(R.layout.activity_main);
14     }
15
16     public void mudaTexto(View v) {
17         TextView etiqueta = (TextView) findViewById(R.id.textView);
18         etiqueta.setText(getResources().getString(R.string.lblHello2));
19     }
20 }
```

Veja que para acessar um recurso, neste caso uma string, usamos o método `getResources().getString(ind id)`. O id da string é encontrado na classe `R`. Volte à tela do designer e selecione o botão. Mude a propriedade `onClick` do botão para responder ao método `mudaTexto`.



Pronto! Agora você pode rodar o aplicativo no seu telefone ou no emulador. **Não se esqueça de fazer os commits e o push para atualizar seu repositório remoto!!!**

Lembra do jogo de adivinhar o número que fizemos como exercício de Java? Agora, implemente-o como um app! Crie um novo projeto no seu repositório local.
Dicas: Use `TextView` para mostrar mensagens e valores ao usuário. Use um `EditText` (na barra de widgets ele aparece como Plain Text) para receber valores do usuário e um botão para ativar o palpite. Não se esqueça de reinicializar o estado do jogo quando o usuário acertar ou errar.
Para pegar o valor de um `EditText`, use o seguinte método:

```
EditText userInput = findViewById(R.id.idDoEditText);
String teste = userInput.getText().toString();
```

Não se esqueça de usar os métodos adequados para converter `String` para tipos numéricos e vice-versa. E sempre é bom lembrar, faça os commits e pushes sempre antes de modificar código que está funcionando. Bom trabalho.