



ANDROID

TipCalc App



Prof. Alessandro Brawerman

TipCalc App

- O objetivo desta app é desenvolver uma calculadora de gorjetas.
- Para tal será projetada uma interface usando o `TableLayout` e os componentes `TextView`, `EditText` e `SeekBar`.
- Ao final a interface será igual a da figura abaixo.

TipCalc App - TableLayout

- Forma uma tabela de n linhas e m colunas.
- Linhas e colunas iniciam no índice 0.
- Cada célula pode estar vazia ou conter um elemento, o qual pode ser uma layout com outros componentes.
- Um componentes pode abranger várias colunas ou linhas.
- Por padrão, os componentes são adicionados da esquerda para direita em uma linha.

<http://developer.android.com/reference/android/widget/TableLayout.html>

<http://developer.android.com/reference/android/widget/TableRow.html>

TipCalc App - TableLayout

Rows and columns in a TableLayout

	column 0	column 1	column 2	column 3
row 0	Bill total	0.00		
row 1		10%	15%	20%
row 2	Tip	0.00	0.00	0.00
row 3	Total	0.00	0.00	0.00
row 4	Custom			18%
row 5	Tip	0.00	Total	0.00

TipCalc App

- Crie um projeto Android App e chame-o de TipCalc.
- Abra o xml da view e troque o layout para TableLayout.

```
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
    android:layout_height="match_parent" android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    android:paddingBottom="16dp" tools:context=".MainActivity">

</TableLayout>
```

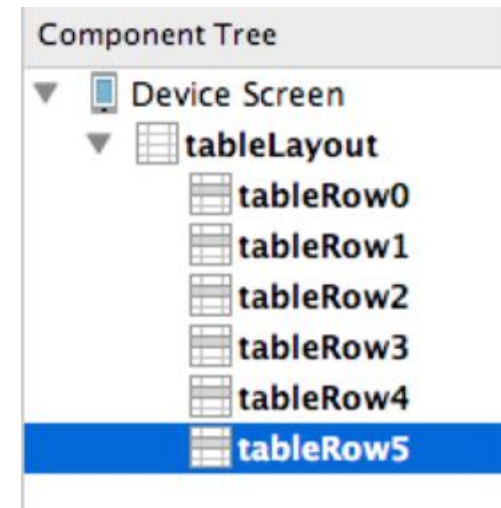
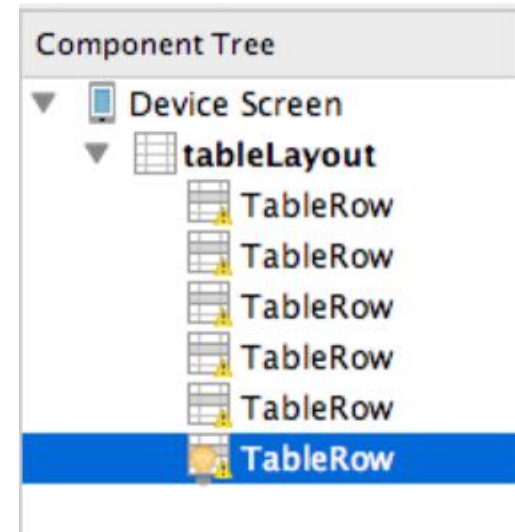
TipCalc App – Propriedades do Layout

- Mude o background do layout para o valor #DADADA.
 - Clique na propriedade background do layout e escolha a aba Color
 - Entre com o valor #DADADA (FFDADADA)
 - A tela fica com um tom de cinza
- Mude as propriedades abaixo do TableLayout:
 - Id → @+id/tableLayout
 - Shrink Columns → 1, 2, 3

TipCalc App – TableRow

- O componente TableRow forma as linhas da tabela.
- Adicione 6 TableRows ao TableLayout.
 - Na janela Outline, arraste os tableRows para o tableLayout

- Acerte os IDs das TableRows para ir de 0 a 5

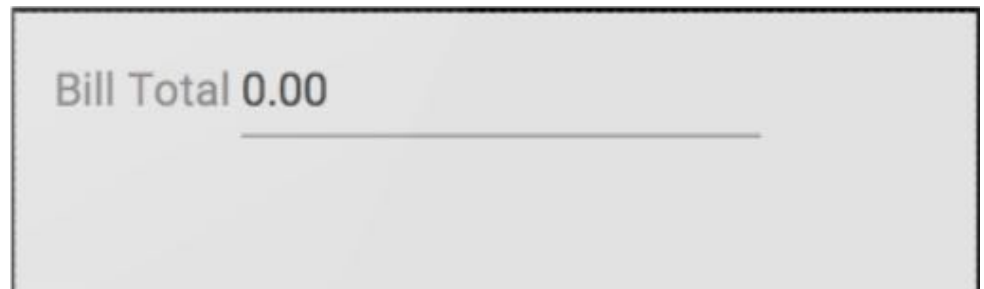
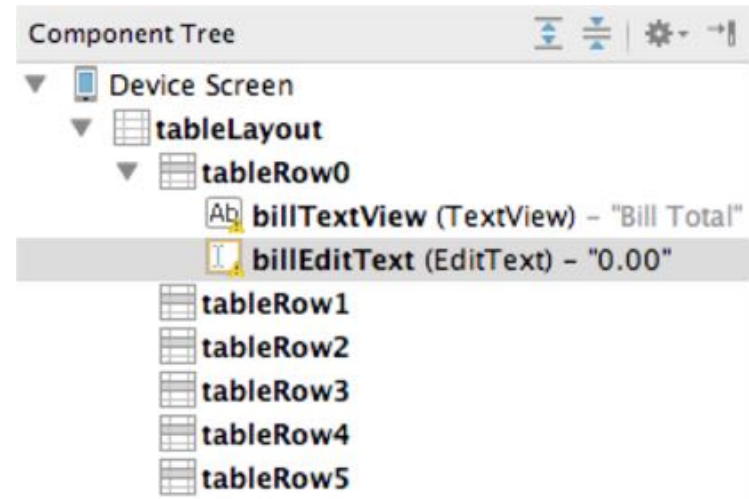


TipCalc App – TableRow

- Selecione cada TableRow e mude a propriedade Layout width para match_parent.
- Pode ser feito selecionando todos ao mesmo tempo.
- Com todos selecionados procure pela propriedade Width e selecione match_parent.
- Assim as linhas terão a largura total do layout.

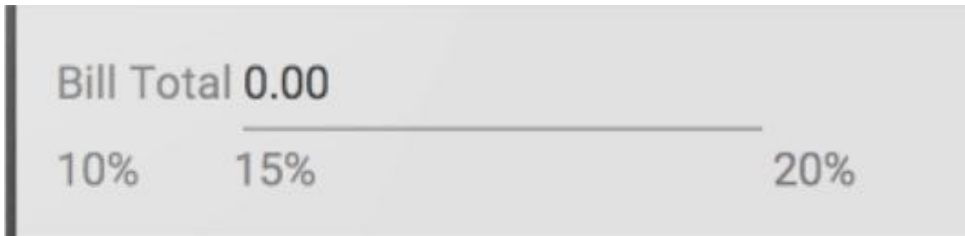
TableRow 0

- Adicione um Medium TextView e um TextField Number (Decimal).
- TextView
 - ID = billTextView
 - Text = Bill Total
- TextField
 - ID = billEditText
 - Text = 0.00



TableRow 1

- Adicione 3 componentes Medium TextView.
- Configure os IDs e Text para:
 - tenTextView e 10%
 - fifteenTextView e 15%
 - twentyTextView e 20%



Bill Total 0.00		
<hr/>		
10%	15%	20%

TableRow 2

- Adicione 1 Medium TextView e 3 Number TextFields.
- TextView:
 - ID = tipTextView
 - Text = Tip
- TextFields:
 - IDs = tip10EditTextField, tip15EditTextField, tip20EditTextField
 - Text = 0.00, 0.00, 0.00 (para cada TextField)

Bill Total 0.00			
10%	15%	20%	
Tip	0.00	0.00	0.00

TableRow 3

- Adicione 1 Medium TextView e 3 Number TextFields.
- TextView:
 - ID = totalTextView
 - Text = Total
- TextFields:
 - IDs = total10EditTextField, total15EditTextField, total20EditTextField
 - Input Type = number
 - Text = 0.00, 0.00, 0.00 (para cada TextField)

Bill Total 0.00			
10%	15%	20%	
Tip	0.00	0.00	0.00
Total	0.00	0.00	0.00

TableRow 4

- Adicione 1 Medium TextView, 1 SeekBar e 1 Medium TextView.
- TextView:
 - ID = customTextView
 - Text = Custom
- SeekBar:
 - Max = 30
 - Progress = 18
 - ID = customSeekBar
- TextView:
 - ID = customTipTextView
 - Text = 18%



Bill Total 0.00			
10%	15%	20%	
Tip	0.00	0.00	0.00
Total	0.00	0.00	0.00
Custom			18%

TableRow 5

- Adicione 1 Medium TextView, 1 Number TextField, 1 Medium TextView e outro Number TextField.
- TextView:
 - ID = tipCustomTextView
 - Text = Tip
- TextField:
 - ID = tipCustomEditTextField
 - Text = 0.00
- TextView:
 - Text = Total
- TextField:
 - ID = totalCustomEditTextField
 - Text = 0.00

Bill Total 0.00			
10%	15%	20%	
Tip	0.00	0.00	0.00
Total	0.00	0.00	0.00
Custom		18%	
Tip	0.00	Total	0.00

Acertando a cor dos TextViews

- Selecione todos os componentes TextView na janela Outline.
- Clique e mantenha o Ctrl pressionado.
- Selecione a propriedade Text Color e insira o valor de #000.

Acertando as Colunas das %

- Os campos 10, 15 e 20% estão nas colunas incorretas.
- Altere a propriedade layout:column para 1.
 - Veja como fica no xml
 - `android:layout_column="1"`
- As outras % automaticamente vão para as colunas subsequentes.

Bill Total 0.00			
	10%	15%	20%
Tip	0.00	0.00	0.00
Total	0.00	0.00	0.00
Custom			18%
Tip	0.00	Total	0.00

Centralizando Componentes e Acertando a Fonte

- Na TableRow1 selecione os 3 TextView e acerte a propriedade Gravity para Center (both).
- Em seguida, selecione todos os componentes TextField das tableRows 2, 3 e 5 e acerte Gravity = Center e Text size = 18sp.
- Acerte a fonte também do TextField da tableRow0.

Acerte o SeekBar

- Vamos fazer com que o SeekBar ocupe duas colunas.
- Selecione o componente e procure pela propriedade `layout:span`.
- Modifique-a para 2
- Veja o xml
 - `android:layout_span="2"`

The screenshot shows a bill calculator interface. At the top, it says "Bill Total 0.00". Below this is a table with three columns: "10%", "15%", and "20%". The rows are "Tip", "Total", and "Custom". The "Tip" row shows "0.00" for all three percentages. The "Total" row shows "0.00" for all three percentages. The "Custom" row shows a SeekBar with a teal slider and a blue dot, with the value "18%" displayed to the right. Below the "Custom" row, there is another row with "Tip" and "Total" labels, showing "0.00" for both.

	10%	15%	20%
Tip	0.00	0.00	0.00
Total	0.00	0.00	0.00
Custom			18%
Tip	0.00	Total	0.00

Acertando os TextViews Labels

- Selecione todos os TextView que estão a esquerda da View (Bill Total, Tip, etc) e acerte as propriedades abaixo:
 - Gravity = Center Vertical, Right
 - Padding Right = 5dp
 - Faça o mesmo para o campo Text View - Total na tableRow 5
- Selecione todas as tableRows e acerte o padding bottom para 10dp.

Acertando os TextViews Labels

- Após, selecione o componente customTipTextView na tableRow4 e acerte Gravity = Center Vertical e Padding Left = 5p.
- Acerte o layout height dos componentes TextView da tableRow4 para match_parent, deste forma eles ficarão centralizados verticalmente em relação à tableRow.

Finalizando o SeekBar

- Padding left e right como 8dp.
- Focusable = false (assim o billEditText não perde o foco ao movimentarmos o marcador do SeekBar)
 - Clique no focusable e clique novamente para desmarcar
- Selecione todos os TextFields, exceto o billEditText, e configure focusable = false e longClickable = false.

Finalizando o billEditText

- Acerte a propriedade span para ocupar o restante do tela.
 - `layout:span = 3.`

MainActivity.java

- Declaração de variáveis.

```
public class MainActivity extends ActionBarActivity {  
  
    private double currentBillTotal;  
    private int currentCustomPercent;  
    private EditText tip10EditText;  
    private EditText tip15EditText;  
    private EditText tip20EditText;  
    private EditText total10EditText;  
    private EditText total15EditText;  
    private EditText total20EditText;  
    private TextView customTipTextView;  
    private EditText billEditText;  
    private EditText tipCustomEditText;  
    private EditText totalCustomEditText;  
}
```

Acertando OnCreate – Referências UI

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

    tip10EditText = (EditText)findViewById(R.id.tip10EditTextField);
    tip15EditText = (EditText)findViewById(R.id.tip15EditTextField);
    tip20EditText = (EditText)findViewById(R.id.tip20EditTextField);
    total10EditText = (EditText)findViewById(R.id.total10EditTextField);
    total15EditText = (EditText)findViewById(R.id.total15EditTextField);
    total20EditText = (EditText)findViewById(R.id.total20EditTextField);
    customTipTextView = (TextView)findViewById(R.id.customTipTextView);
    tipCustomEditText = (EditText)findViewById(R.id.tipCustomEditTextField);
    totalCustomEditText = (EditText)findViewById(R.id.totalCustomEditTextField);
    billEditText = (EditText)findViewById(R.id.billEditText);
    SeekBar customSeekBar = (SeekBar)findViewById(R.id.customSeekBar);
}
```


Finalizando OnCreate

```
currentCustomPercent = customSeekBar.getProgress();  
  
//Lidar com evento de mudança do campo de entrada da conta  
billEditText.addTextChangedListener(billTextWatcher);  
  
//Eventos de mudança de valor do seekbar  
customSeekBar.setOnSeekBarChangeListener(customSeekBarListener);  
}
```

Atualizando Campos de %

//Atualiza os tip EditTexts de 10, 15 e 20%

```
private void updateStandard() {  
    double tenPercentTip = currentBillTotal * .10;  
    double tenPercentTotal = currentBillTotal + tenPercentTip;  
    double fifteenPercentTip = currentBillTotal * .15;  
    double fifteenPercentTotal = currentBillTotal + fifteenPercentTip;  
    double twentyPercentTip = currentBillTotal * .20;  
    double twentyPercentTotal = currentBillTotal + twentyPercentTip;  
  
    tip10EditText.setText(String.format("%.02f", tenPercentTip));  
    total10EditText.setText(String.format("%.02f", tenPercentTotal));  
    tip15EditText.setText(String.format("%.02f", fifteenPercentTip));  
    total15EditText.setText(String.format("%.02f", fifteenPercentTotal));  
    tip20EditText.setText(String.format("%.02f", twentyPercentTip));  
    total20EditText.setText(String.format("%.02f", twentyPercentTotal));  
}
```

Atualizando Campos de %

```
//Atualiza gorjeta personalizada  
private void updateCustom() {  
    customTipTextView.setText(currentCustomPercent + "%");  
  
    double customTipAmount = currentBillTotal * currentCustomPercent * .01;  
    double customTotalAmount = currentBillTotal + customTipAmount;  
  
    tipCustomEditText.setText(String.format("%.02f", customTipAmount));  
    totalCustomEditText.setText(String.format("%.02f", customTotalAmount));  
}
```

Mudança no SeekBar

- A interface `OnSeekBarChangeListener` “ouve” qualquer mudança de status no `SeekBar`.
- Quando há uma mudança o evento `onProgressChanged` é chamado.
- Devemos implementar então os nossos controllers que cuidarão da mudança do `SeekBar` internamente ao código.
- A ideia é capturar a mudança do `SeekBar` e propagar a mesma ao chamar o método `updateCustom`.

Mudança no SeekBar

```
//Evento de mudança de posição do SeekBar
private SeekBar.OnSeekBarChangeListener customSeekBarListener =
    new SeekBar.OnSeekBarChangeListener() {
        @Override
        public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {
            currentCustomPercent = progress;
            updateCustom();
        }

        @Override
        public void onStartTrackingTouch(SeekBar seekBar) {
        }

        @Override
        public void onStopTrackingTouch(SeekBar seekBar) {
        }
    };
```

Mudança no SeekBar

- Java exige que cada método de uma interface que seja implementada deva ser sobrescrito.
- Portanto, os métodos `onStartTrackingTouch` e `onStopTrackingTouch` devem ser sobrescritos no código.
- Como estes métodos não são usados em nosso código, devemos apenas fornecer o protótipo dos mesmos.

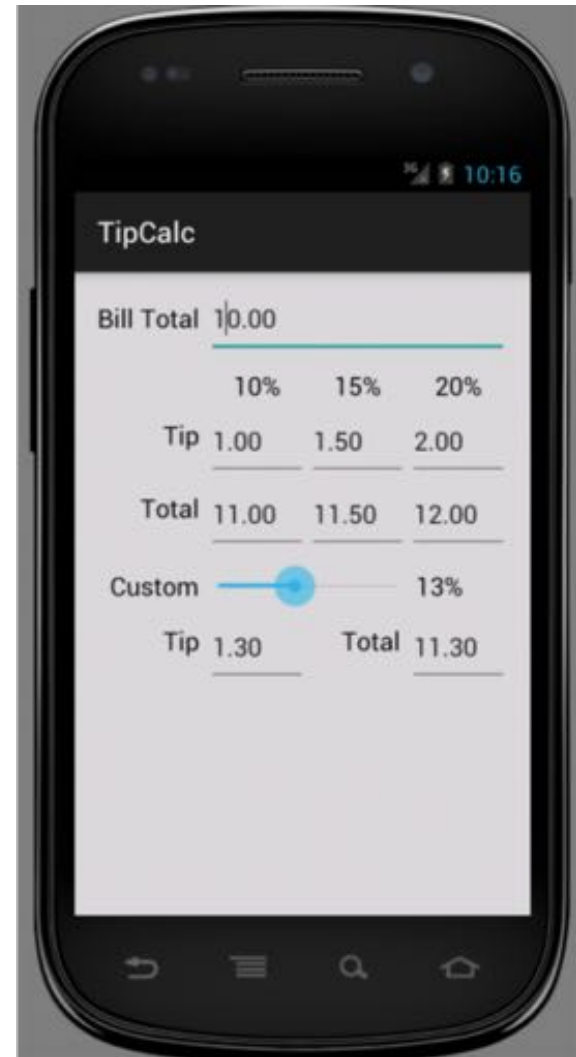
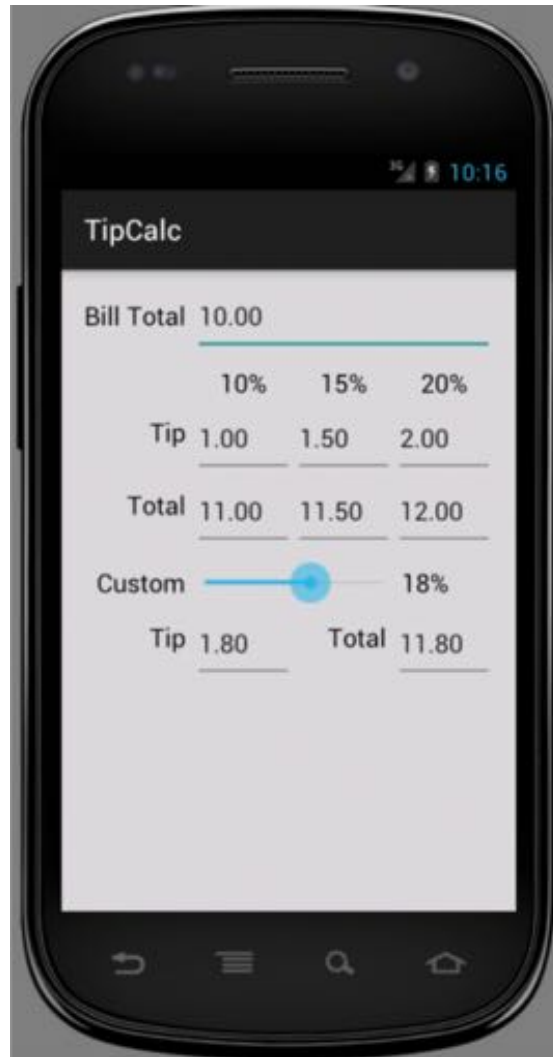
Mudança no Campo da Conta

- É necessário também perceber mudanças no campo conta do nosso app.
- Para tal, o evento `onTextChanged` é acionado e em nosso caso adicionamos um `TextWatcher` para “ouvir” a estes eventos.
- O evento recebe 4 parâmetros:
 - `CharSequence` – texto do campo
 - `Start` – posição inicial
 - `Count` – quantos caracteres
 - `Before` – texto anterior de comprimento `before`
- Os `count` caracteres, a partir de `start`, substituíram o texto anterior de tamanho `before`.

Mudança no Campo da Conta

```
private TextWatcher billTextWatcher = new TextWatcher() {  
    @Override  
    public void beforeTextChanged(CharSequence s, int start, int count, int after) {  
    }  
  
    @Override  
    public void onTextChanged(CharSequence s, int start, int before, int count) {  
        try {  
            currentBillTotal = Double.parseDouble(s.toString());  
        } catch (NumberFormatException e) {  
            currentBillTotal = 0.0;  
        }  
        updateCustom();  
        updateStandard();  
    }  
  
    @Override  
    public void afterTextChanged(Editable s) {  
    }  
};
```


Exemplo



Exercício

- Amplie o app TipCalc para que o usuário possa dividir a conta em até 10 pessoas.
 - Insira na view um seekbar e textview para informar o número de pessoas que irão dividir a conta.
 - Insira ainda um textview que informe o preço da conta total (conta + gorjeta) por pessoa.