

INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
MARANHÃO

Prof. MSc. Flávio Barros

*flavioifma@gmail.com*

**[www.flaviobarros.com.br](http://www.flaviobarros.com.br)**

# Programação para Dispositivos Móveis

## Aula - Layout (parte II)

# Roteiro

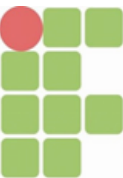
## ListView



# ListView

## Fundamentos

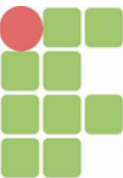
- É um grupo de exibições que exibe uma lista de itens roláveis;
- Os itens da lista são inseridos automaticamente na lista usando um `Adapter` que obtém conteúdo de uma origem como uma matriz ou consulta de banco de dados; e
- Converte cada resultado de item em uma exibição, que é colocada na lista.



# ListView

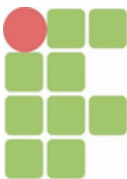
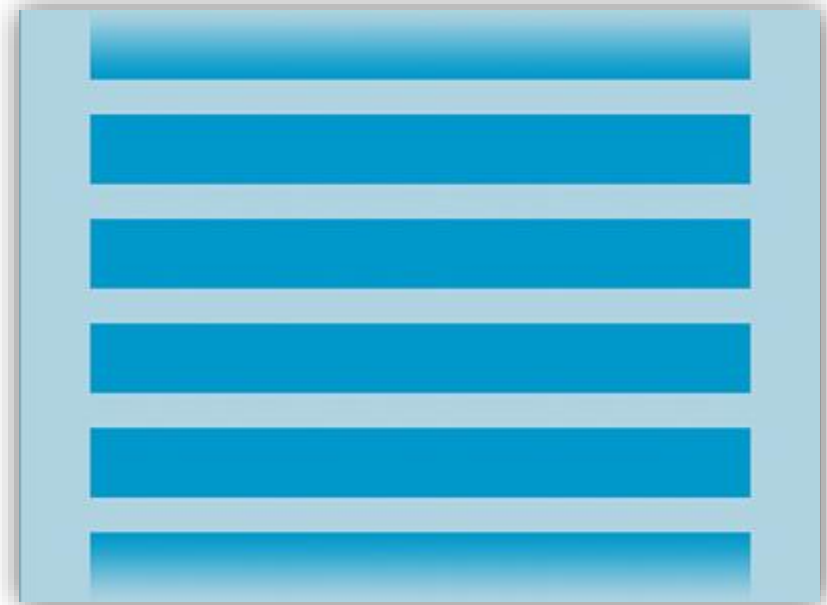
## Como usar um carregador

- O uso de um `CursorLoader` é a forma padrão de consultar um `Cursor` como uma tarefa assíncrona para evitar que a consulta bloqueie o encadeamento principal do aplicativo;
- Quando o `CursorLoader` recebe o resultado `Cursor`, os `LoaderCallbacks` recebem um retorno de chamada para `onLoadFinished()`, que é onde o `Adapter` é atualizado com o novo `Cursor`.



# ListView

```
<ListView  
    android:id="@+id/lst_lista"  
    android:layout_width="wrap_content"  
    android:layout_height="345dp"  
    android:layout_centerHorizontal="true"  
    android:layout_marginTop="10dp" />
```



# Roteiro



Prj\_Android09

# Prj\_Android09

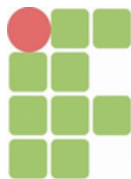
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="20dp"
    android:paddingRight="20dp"
    android:orientation="vertical" >
    <TextView
        android:id="@+id/txt_titulo"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Listagem"
        android:textSize="20dp"
        android:gravity="center_horizontal"
        android:layout_marginTop="20dp" />
    <ListView
        android:id="@+id/lst_lista"
        android:layout_width="wrap_content"
        android:layout_height="345dp"
        android:layout_below="@+id/txt_titulo"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="10dp" />
```

```
<Button
```

```
    android:id="@+id/btn_sair"
    android:layout_width="100dp"
    android:layout_height="wrap_content"
    android:layout_gravity="right"
    android:layout_marginTop="40dp"
    android:text="Sair"
    android:onClick="Sair"/>
```

```
</LinearLayout>
```

activity\_main.xml



# Prj\_Android09

```

public class MainActivity extends AppCompatActivity {

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

        ListView listView = (ListView) findViewById(R.id.lst_lista);
        String[] values = new String[] { "1. Arroz", "2. Feijão", "3. Macarrão", "4. Farinha",
            "5. Biscoito", "6. Manteiga", "7. Sal", "8. Milho Verde", "9. Azeitona",
            "10. Queijo", "11. Leite", "12. Refrigerante"};

        ArrayAdapter<String> adapter = new ArrayAdapter<String>( context: this,
            android.R.layout.simple_list_item_1, android.R.id.text1, values);

        listView.setAdapter(adapter);
    }

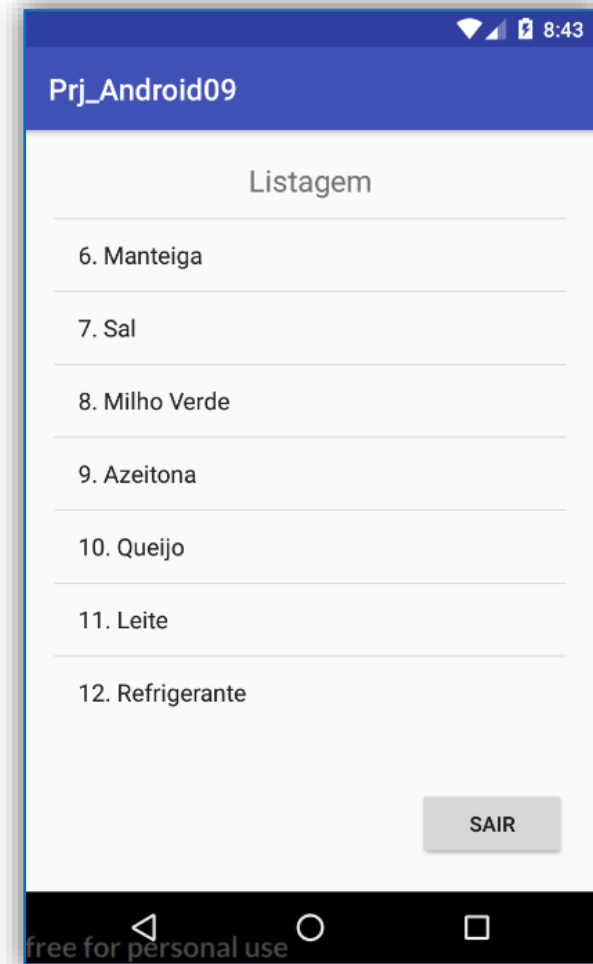
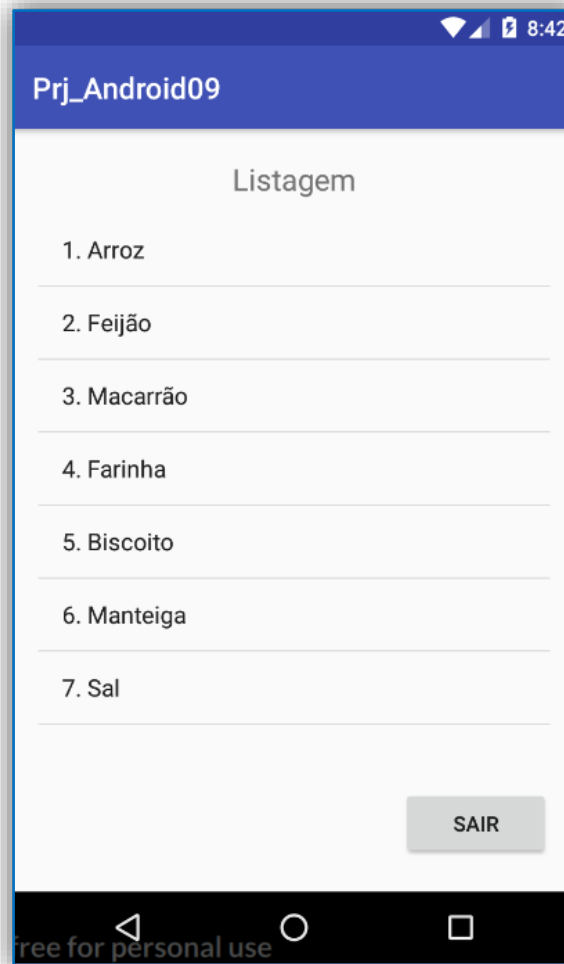
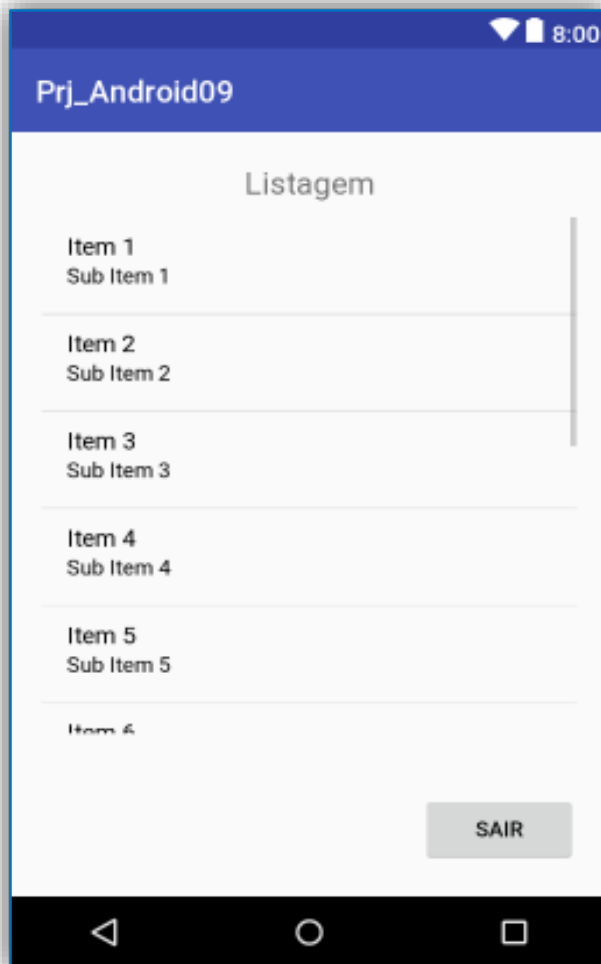
    protected void Sair(View view){
        finish();
    }
}

```

MainActivity.java



# Prj\_Android09



activity\_main.xml



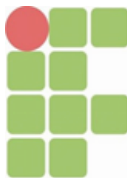
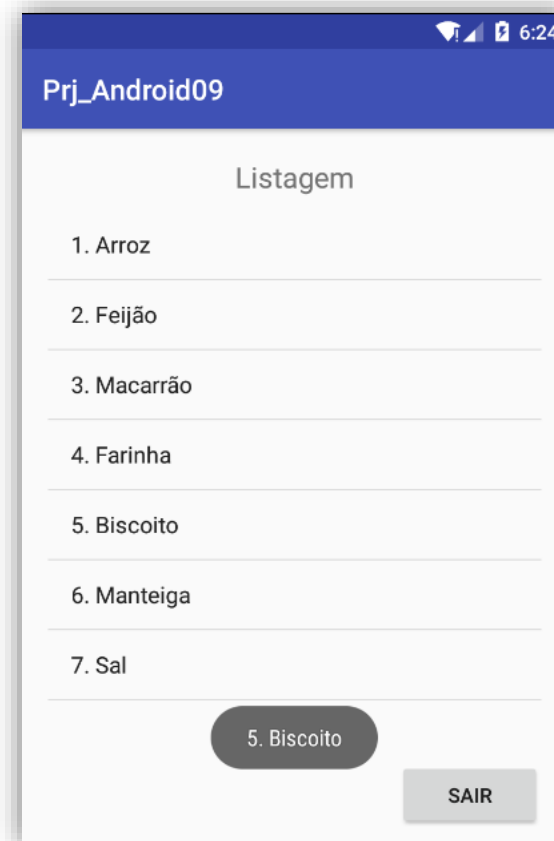
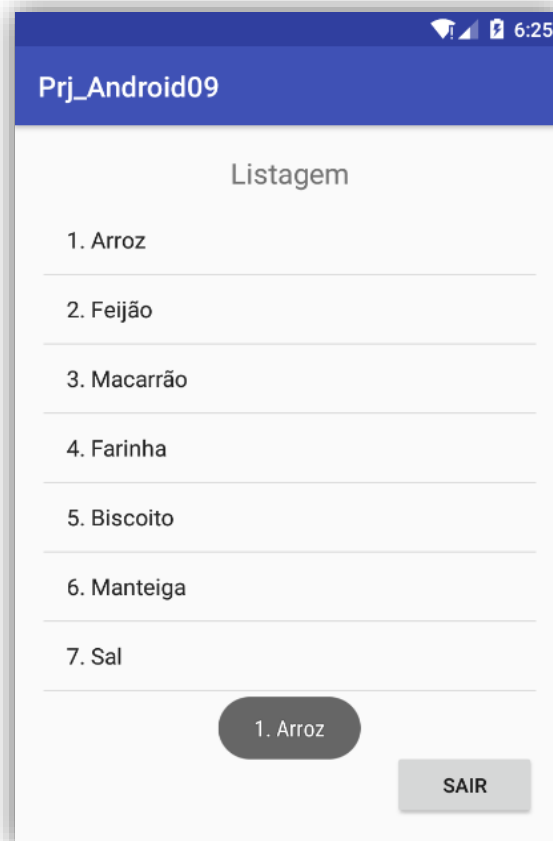
# Roteiro



Praticando...

# Prj\_Android\_Extra07

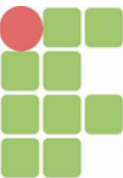
- Acrescentar linha de comando no **Prj\_Android09** que mostre o conteúdo através de `Toast.makeText()`, do respectivo item selecionado.



# Praticando

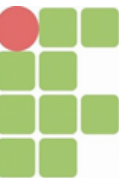
## Prj\_Android\_Extra08

- Criar um aplicativo que liste 3 nomes de tecnologias com imagem (como ícone), e mostre na parte de baixo da tela, uma descrição (com características) deste item selecionado.



# Roteiro

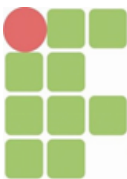
# GridLayout



# GridLayout

## Fundamentos

- Um layout que coloca seus filhos em uma grade retangular.
- A grade é composta por um conjunto de linhas infinitamente finas que separam a área de visualização em células;
- Uma grade com  $N$  colunas possui  $N + 1$  índices de grade que vão de 0 a  $N$ , inclusive;
- Independentemente de como o `GridLayout` é configurado, o índice de grade 0 é fixado à borda principal do contêiner e o índice de grade  $N$  é fixado em sua borda posterior (após o preenchimento ser levado em consideração).



# GridLayout

## Especificações de linha e coluna

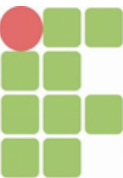
- Os filhos ocupam uma ou mais células contíguas, conforme definido pelos parâmetros de `layout rowSpec` e `columnSpec`;
- Cada especificação define o conjunto de linhas ou colunas que devem ser ocupadas; e como os filhos devem estar alinhadas dentro do grupo resultante de células;
- Embora as células normalmente não se sobreponham em um `GridLayout`, não impede que os filhos sejam definidas para ocupar a mesma célula ou grupo de células;
- Nesse caso, no entanto, não há garantia de que as crianças não se sobreponham após a conclusão da operação de layout.



# GridLayout

## Atribuição de Célula Padrão

- Se um filho não especificar os índices de linha e coluna da célula que deseja ocupar, o GridLayout atribuirá locais de célula automaticamente usando suas propriedades: `rowCount` e `columnCount`, etc.

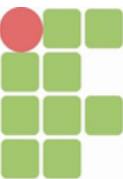




# GridLayout

## Espaço

- O espaço entre os filhos pode ser especificado usando instâncias da visualização de espaço dedicada ou definindo os parâmetros de layout `leftMargin`, `topMargin`, `rightMargin` e `bottomMargin`;
- Quando a propriedade `useDefaultMargins` é definida, as margens padrão em torno dos filhos são automaticamente alocadas com base no guia de estilo da interface do usuário predominante para a plataforma.



# GridLayout

<https://developer.android.com/reference/android/widget/GridLayout>

## Outros Atributos

- Também é possível estender uma linha de células para um número especificado de células; Isso também é aplicável a colunas;
- Existem diferentes atributos presentes em um layout de Grade e para os quais eles podem ser usados:
- **android: alignmentMode** - Quando definido como `alignMargins`, faz com que o alinhamento ocorra entre o limite externo de uma visão, conforme definido por suas margens;
- **android: columnCount** - O número máximo de colunas a serem criadas ao posicionar filhos automaticamente;

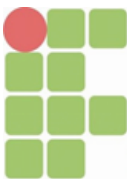


# GridLayout

<https://developer.android.com/reference/android/widget/GridLayout>

## Cont. Outros Atributos

- **android: columnOrderPreserved** - Quando definido como `true`, força os limites da coluna a aparecer na mesma ordem que os índices da coluna;
- **android: orientation** - A propriedade de orientação não é usada durante o layout;
- **android: rowCount** - O número máximo de linhas a serem criadas ao posicionar filhos automaticamente;
- **android: rowOrderPreserved** - Quando definido como `true`, força os limites de linha a aparecer na mesma ordem que os índices de linha;

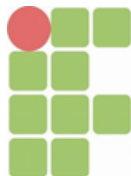
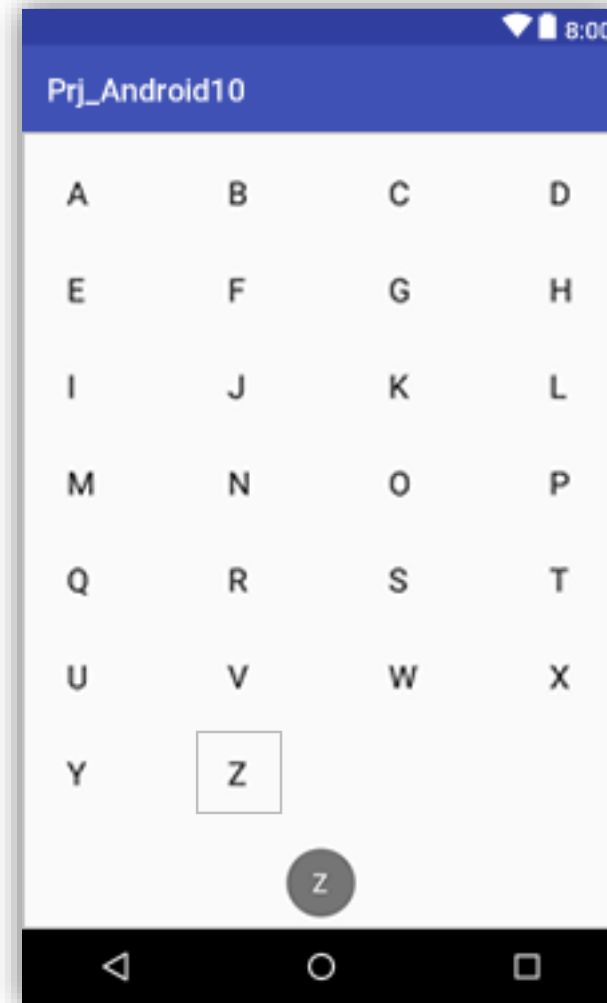
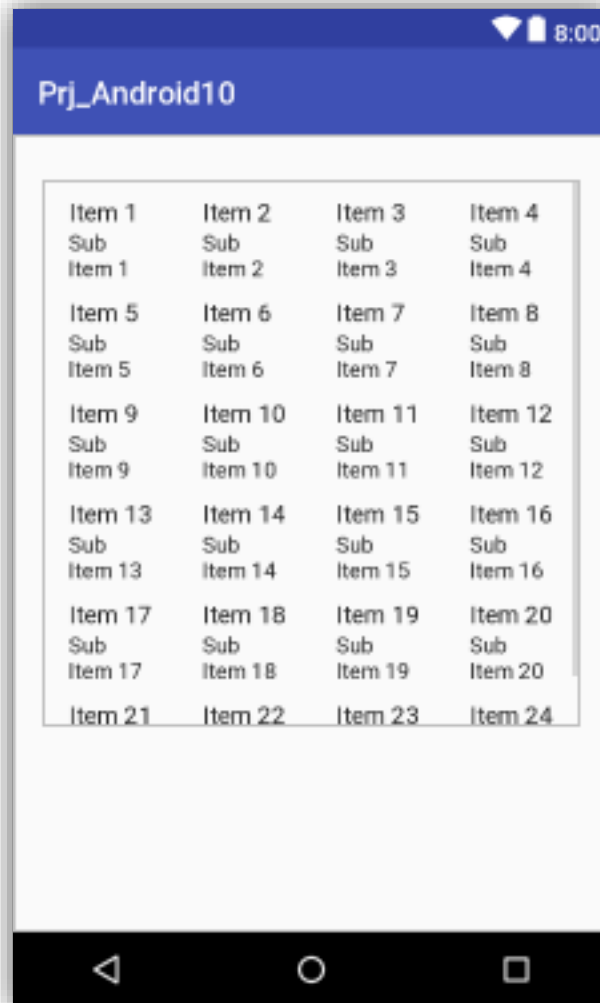


# Roteiro



## Prj\_Android10

# Prj\_Android10



# Prj\_Android10

```
<RelativeLayout
```

```
    xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:padding="20dp" >
```

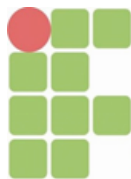
```
    <GridView
```

```
        android:id="@+id/gridview1"  
        android:layout_width="match_parent"  
        android:layout_height="350dp"  
        android:columnWidth="90dp"  
        android:gravity="center"  
        android:numColumns="4"  
        android:stretchMode="columnWidth"  
        android:layout_marginTop="10dp" >
```

```
    </GridView>
```

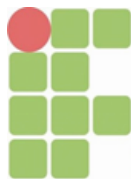
```
</RelativeLayout>
```

activity\_main.xml



# Prj\_Android10

```
public class MainActivity extends AppCompatActivity {  
  
    GridView gridView;  
  
    static final String[] numbers = new String[] {  
        "A", "B", "C", "D", "E",  
        "F", "G", "H", "I", "J",  
        "K", "L", "M", "N", "O",  
        "P", "Q", "R", "S", "T",  
        "U", "V", "W", "X", "Y", "Z"  
    };  
};
```



# Prj\_Android10

```

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    gridView = (GridView) findViewById(R.id.gridview1);

    ArrayAdapter<String> adapter = new ArrayAdapter<String>( context: this,
        android.R.layout.simple_list_item_1, numbers);

    gridView.setAdapter(adapter);
    gridView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View v, int position, long id) {
            Toast.makeText(getApplicationContext(),
                ((TextView) v).getText(), Toast.LENGTH_SHORT).show();
        }
    });
}
}

```

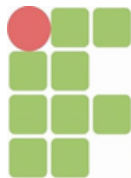
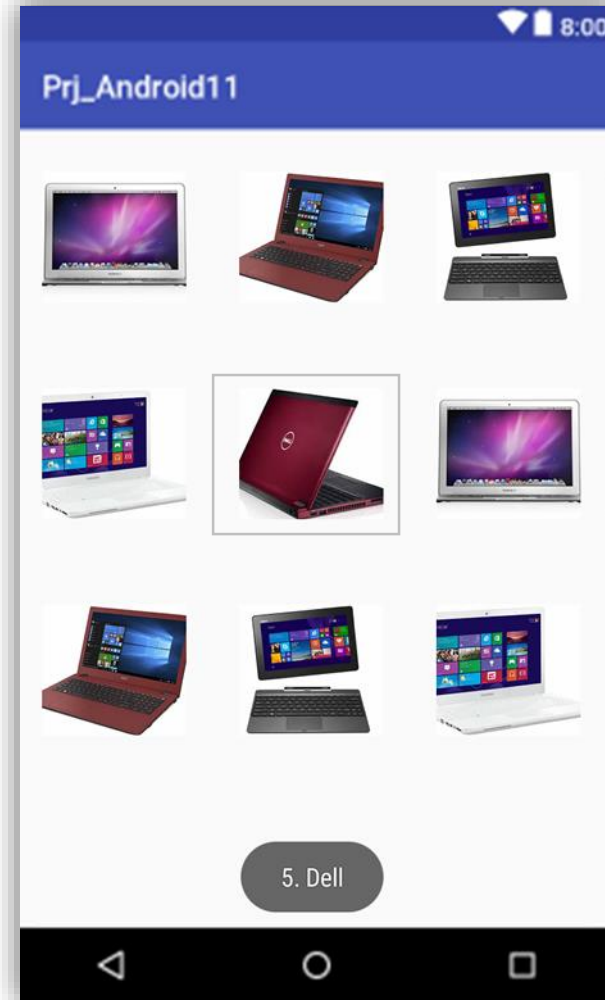
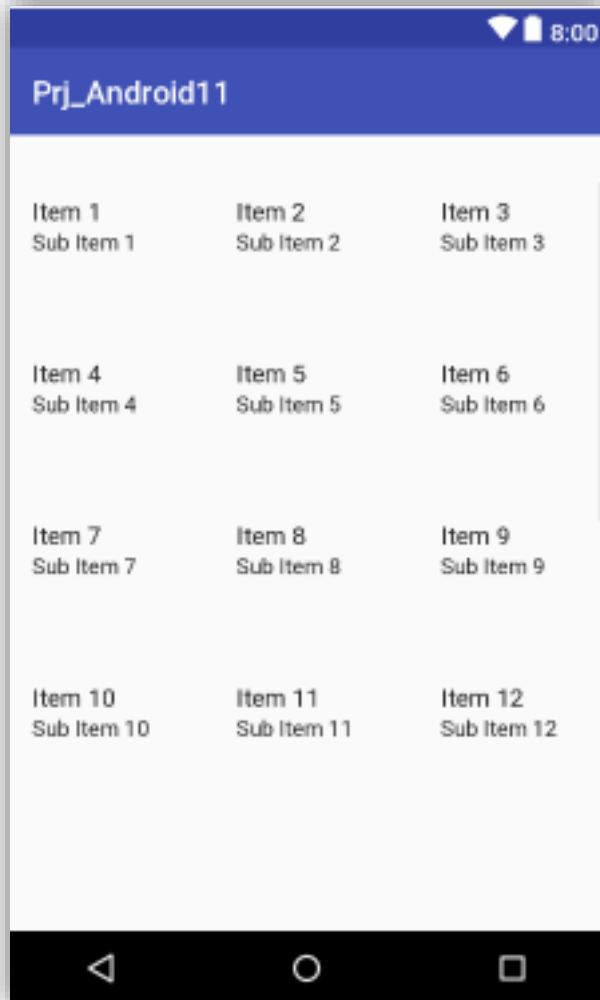


# Roteiro



Prj\_Android11

# Prj\_Android11



# Prj\_Android11

```
<RelativeLayout
```

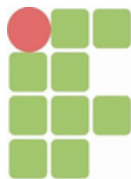
```
xmlns:android="http://schemas.android.com/apk/res/android"  
android:layout_width="match_parent"  
android:layout_height="match_parent" >
```

```
<GridView
```

```
    android:id="@+id/gridview"  
    android:layout_width="match_parent"  
    android:layout_height="400dp"  
    android:layout_marginTop="30dp"  
    android:columnWidth="90dp"  
    android:numColumns="3"  
    android:verticalSpacing="45dp"  
    android:horizontalSpacing="10dp"  
    android:stretchMode="columnWidth"  
    android:gravity="center" />
```

```
</RelativeLayout>
```

activity\_main.xml



# Prj\_Android11

```

public class MainActivity extends AppCompatActivity {

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

        GridView gridView = (GridView) findViewById(R.id.gridview);
        gridView.setAdapter(new ImageAdapter( C this));

        final String[] produtos = new String[] {"1. MacBook", "2. Acer", "3. Azus", "4. Samsung",
            "5. Dell", "6. MacBook", "7. Acer", "8. Azus", "9. Samsung",
            "10. MacBook", "11. Acer", "12. Azus"};

        gridView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> adapterView, View view, int position, long l) {
                String nome_item = produtos[position];
                Toast.makeText( context MainActivity.this, nome_item, Toast.LENGTH_SHORT).show();
            }
        });
    }
}

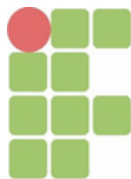
```

MainActivity.java

# Prj\_Android11

```
public class ImageAdapter extends BaseAdapter {  
    private Context mContext;  
  
    public ImageAdapter(Context c) {  
        mContext = c;  
    }  
  
    public int getCount() {  
        return mThumbIds.length;  
    }  
  
    public Object getItem(int position) {  
        return null;  
    }  
  
    public long getItemId(int position) {  
        return 0;  
    }  
}
```

MainActivity.java



# Prj\_Android11

```

@Override
public View getView(int position, View convertView, ViewGroup parent){
    ImageView imageView = new ImageView(mContext);
    imageView.setLayoutParams(new GridView.LayoutParams( w: 180, h: 180));
    imageView.setScaleType(ImageView.ScaleType.CENTER_CROP);
    imageView.setPadding( left: 8, top: 8, right: 8, bottom: 8);
    imageView.setImageResource(mThumbIds[position]);
    return imageView;
}

private Integer[] mThumbIds = {
    R.drawable.img01, R.drawable.img02,
    R.drawable.img03, R.drawable.img04,
    R.drawable.img05, R.drawable.img01,
    R.drawable.img02, R.drawable.img03,
    R.drawable.img04, R.drawable.img01,
    R.drawable.img02, R.drawable.img03
};
}

```

MainActivity.java



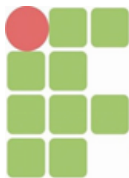
# Roteiro



Praticando...

# Prj\_Android\_Extra09

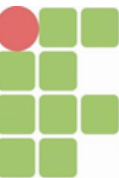
- Crie aplicativo que tenha no mínimo 6 imagens da bandeira de países. Ao pressionar na imagem, mostra-se um `Toast` informando: *"Tocando o hino do Brasil"*, deve-se tocar o seu respectivo hino. Também deverá conter um botão SAIR.





# Roteiro

## Referências



# Referências

- DEITEL, P.; DEITEL, H.; DEITEL, A. **Android 6 - Para Programadores - Uma Abordagem Baseada em Aplicativos**. 2.ed. Bookman, 2015.
- Developer Android “**ListView**”. Disponível em:  
<<https://developer.android.com/guide/topics/ui/layout/listview.html>>.
- Developer Android “**GridLayout**”. Disponível em:  
<<https://developer.android.com/reference/android/widget/GridLayout.html>>.

