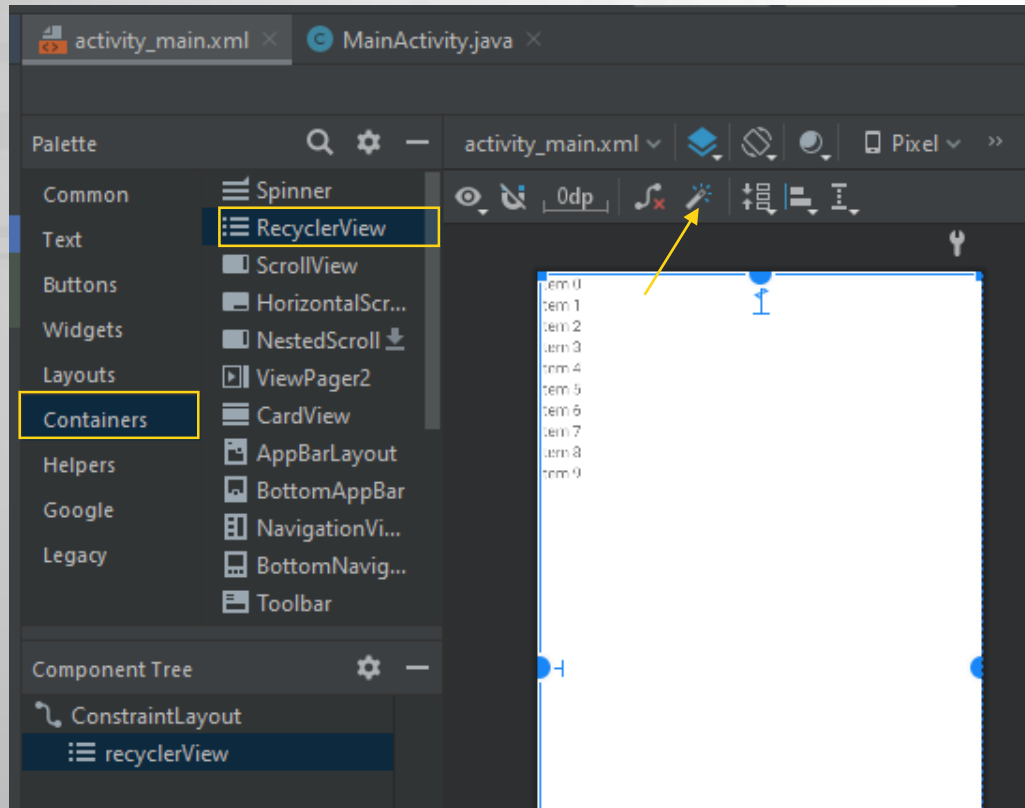




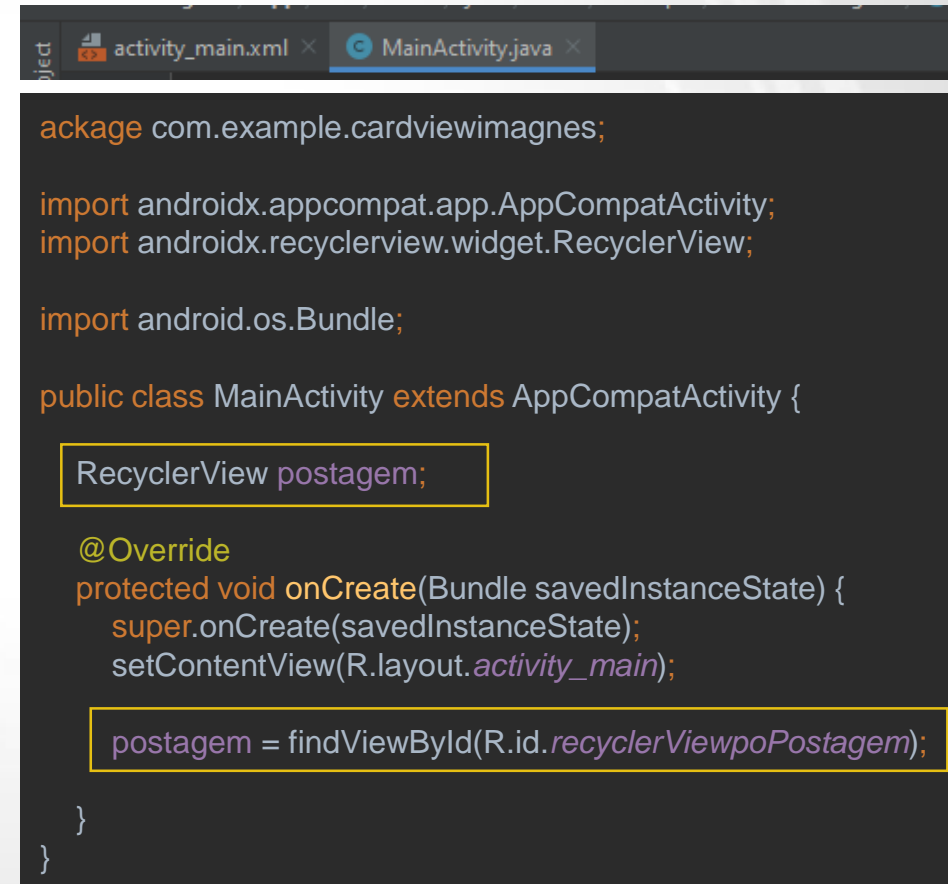
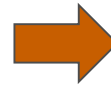
CardView

Prof. Hércules Santos

2023.1



Adicione primeiramente um RecyclerView



No MainActivity, crie o objeto e instancie dentro do método construtor



Definir o Layout

@Override

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

```
    postagem = findViewById(R.id.recyclerViewpoPostagem);
```

```
    RecyclerView.LayoutManager layoutManager = new LinearLayoutManager(this);
```

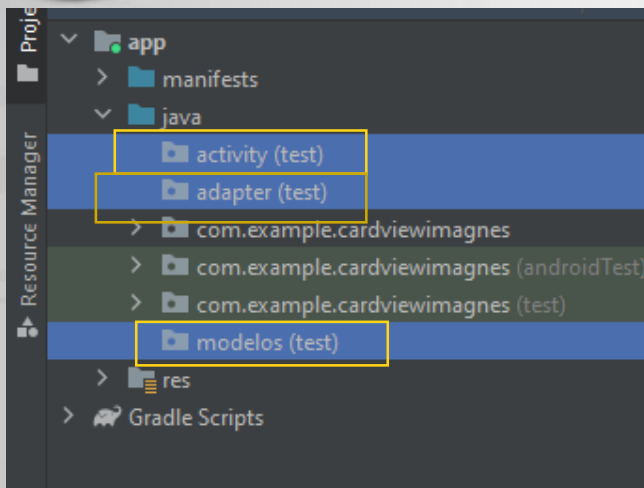
```
    postagem.setLayoutManager(layoutManager);
```

```
}
```

```
}
```



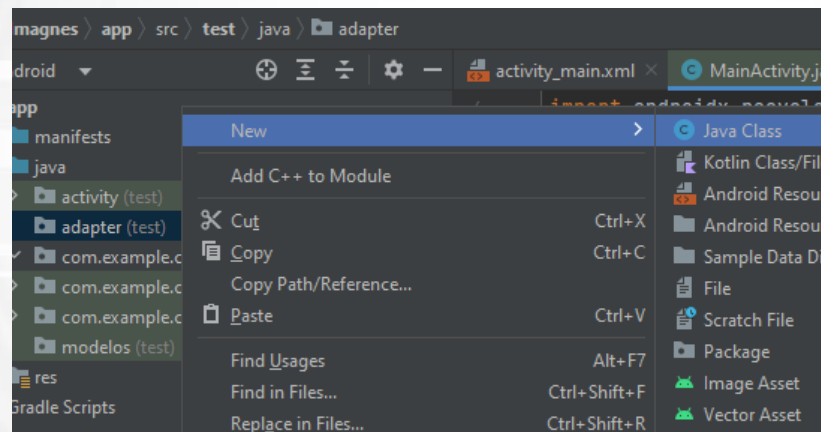
Criando nosso adaptador de postagens



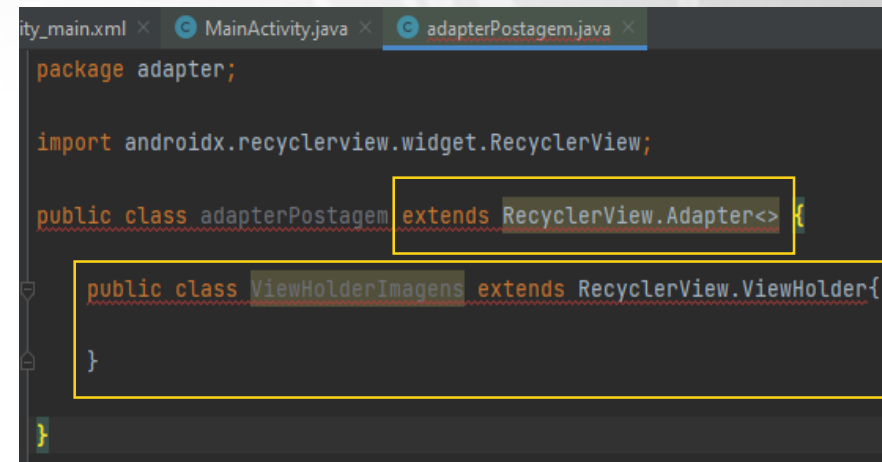
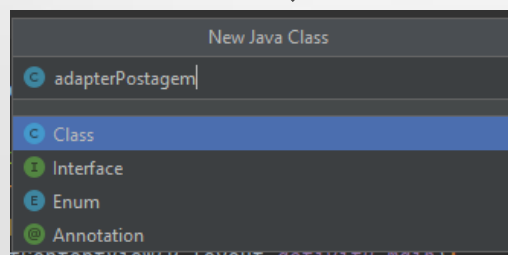
Primeiramente crie os seguintes packages:

- activity
- adapter
- modelos

Mova o MainActivity para dentro do package activity.



Com o botão direito, no package adapter, crie uma JavaClass que será onde iremos implementar nosso adaptador.



Adicione o extends RecyclerView.Adapter e em seguida crie outra classe dentro desta chamada ViwHolderImagens.



Classe ViewHolderImagens

Vamos criar os atributos que farão parte da nossa postagem

```
activity_main.xml x MainActivity.java x adapterPostagem.java x  
  
public class ViewHolderImagens extends RecyclerView.ViewHolder{  
    private TextView nome;  
    private TextView postagem;  
    private ImageView imagens;  
}
```

Esse são os atributos que aparecerão na nossa postagem

Feito isso, agora crie o método construtor para a classe ViewHolderImagens

```
activity_main.xml x adapterPostagem.java x MainActivity.java x  
1 package adapter;  
2  
3 import android.widget.ImageView;  
4 import android.widget.TextView;  
5  
6 import androidx.recyclerview.widget.RecyclerView;  
7  
8 public class adapterPostagem extends RecyclerView.Adapter<ViewHolderImagens> {  
9  
10 public class ViewHolderImagens extends RecyclerView.ViewHolder {  
11  
12 private TextView postagem;  
13 private ImageView imagens;  
14  
15 }  
16 }
```



```
activity_main.xml x adapterPostagem.java x MainActivity.java x  
package adapter;  
  
import android.view.View;  
import android.widget.ImageView;  
import android.widget.TextView;  
  
import androidx.annotation.NonNull;  
import androidx.recyclerview.widget.RecyclerView;  
  
public class adapterPostagem extends RecyclerView.Adapter<ViewHolderImagens> {  
  
    public class ViewHolderImagens extends RecyclerView.ViewHolder {  
  
        private TextView nome;  
        private TextView postagem;  
        private ImageView imagens;  
  
        public ViewHolderImagens(@NonNull View itemView) {  
            super(itemView);  
        }  
    }  
}
```

Click na lâmpada e escolha Create constructor



Acessando a classe ViewHolderImagens

activity_main.xml × AdapterPostagem.java × MainActivity.java >

```
import androidx.recyclerview.widget.RecyclerView;

public class adapterPostagem extends RecyclerView.Adapter<> {

    public class ViewHolderImagens extends RecyclerView.ViewHolder {

        private TextView nome;
        private TextView postagem;
        private ImageView imagens;

        public ViewHolderImagens(@NonNull View itemView) {
            super(itemView);
        }
    }
}
```



Agora click na lâmpada, para implementar os métodos

```
import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

public class AdapterPostagem extends RecyclerView.Adapter<AdapterPostagem.ViewHolder> {

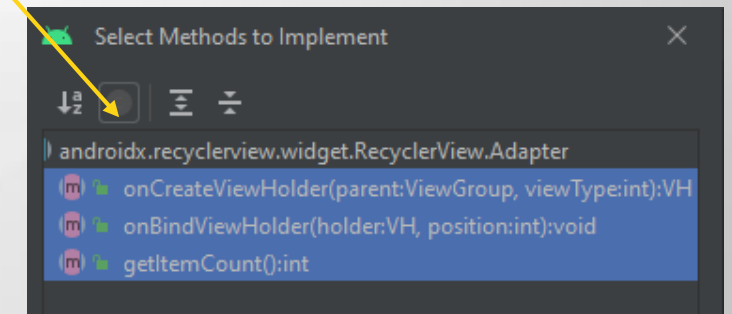
    private TextView nome;
    private TextView postagem;
    private ImageView imagens;
}
```

Uma vez criada a classe ViewHolderImagens, agora podemos acessá-la.

Para, isso digite dentro do < > digite:

<AdapterImagens.ViewHolderImagens>

```
public class AdapterPostagem extends RecyclerView.Adapter<AdapterPostagem.ViewHolderImagens> {
```





Configurando o Adapter

```
activity_main.xml x MainActivity.java x AdapterPostagem.java x

import android.os.Bundle;

import com.example.cardviewimagines.R;

import adapter.AdapterPostagem;

2 usages
public class MainActivity extends AppCompatActivity {

    3 usages
    RecyclerView postagem;

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

        postagem = findViewById(R.id.recyclerViewpoPostagem);

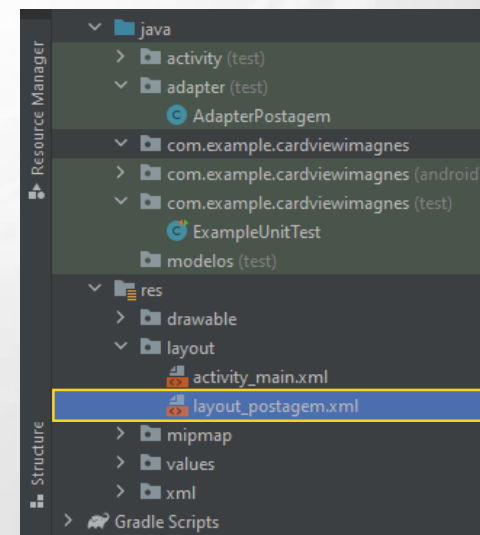
        RecyclerView.LayoutManager layoutImagens = new LinearLayoutManager(context: this);
        postagem.setLayoutManager(layoutImagens);

        AdapterPostagem adaptador = new AdapterPostagem();
        postagem.setAdapter(adaptador);
    }
}
```

Agora que já criamos o Adaptador: **AdapterImagens**, já podemos configurá-lo dentro do construtor do ActivityMain

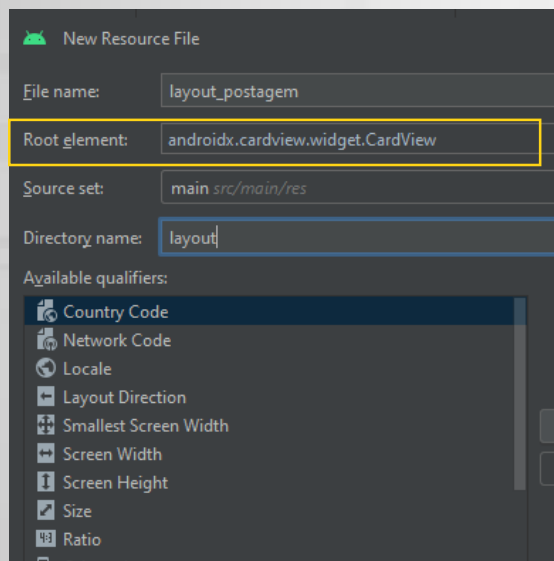
Para implementar cada método da classe AdapterPostagem, é preciso criar um Layout, que será a interface.

Vamos fazer isso dentro do package layout

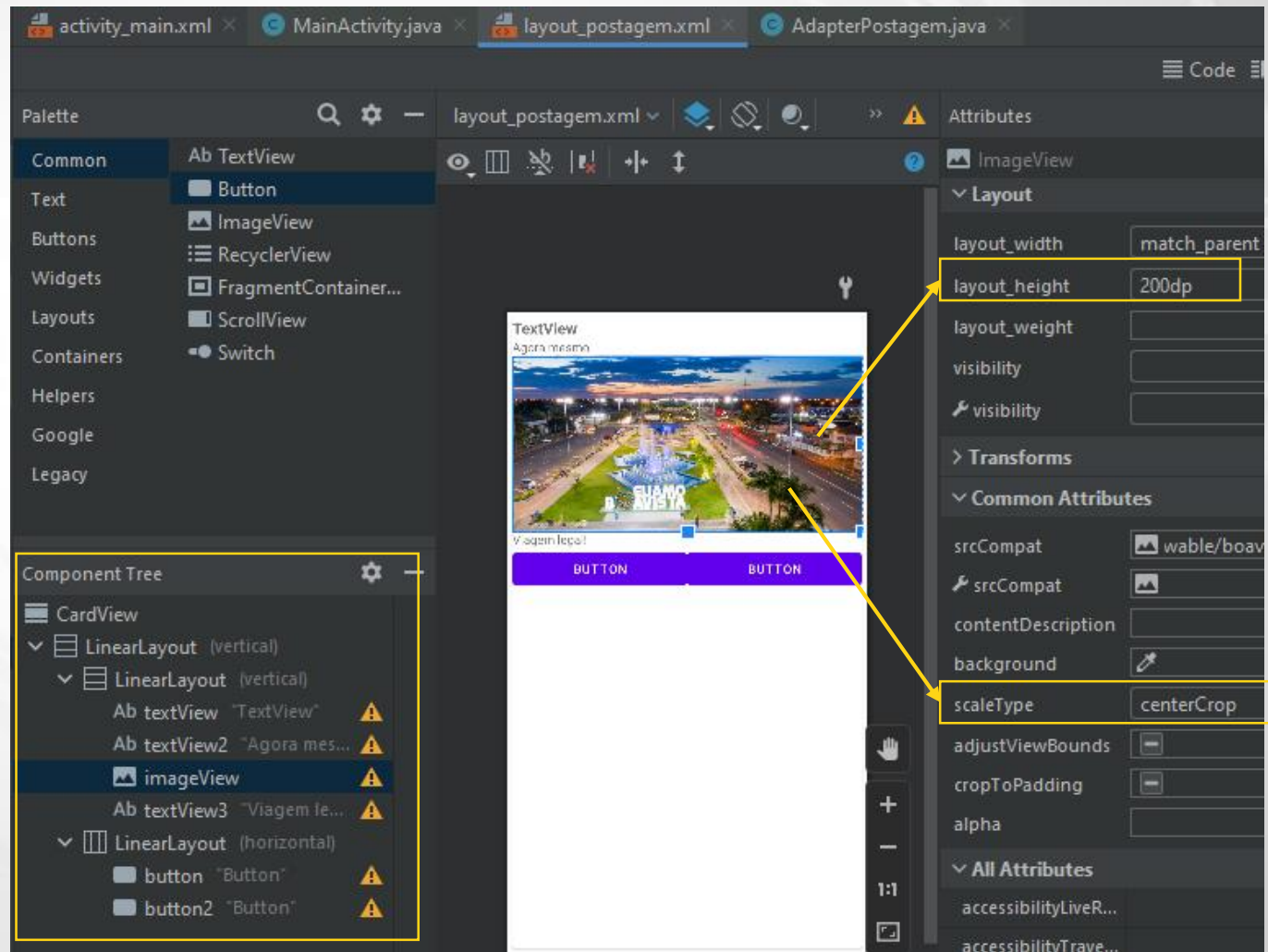




Layout



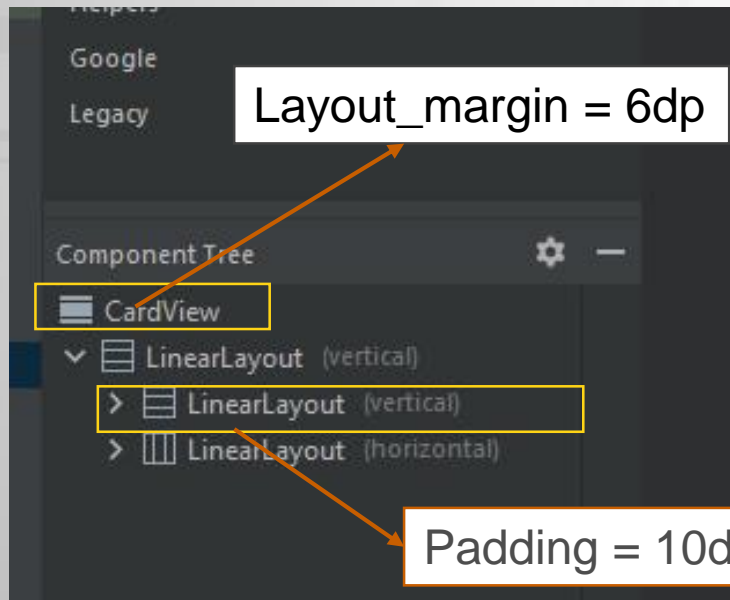
Ao criar um New Layout Resource File, escolha o Root element como CarView



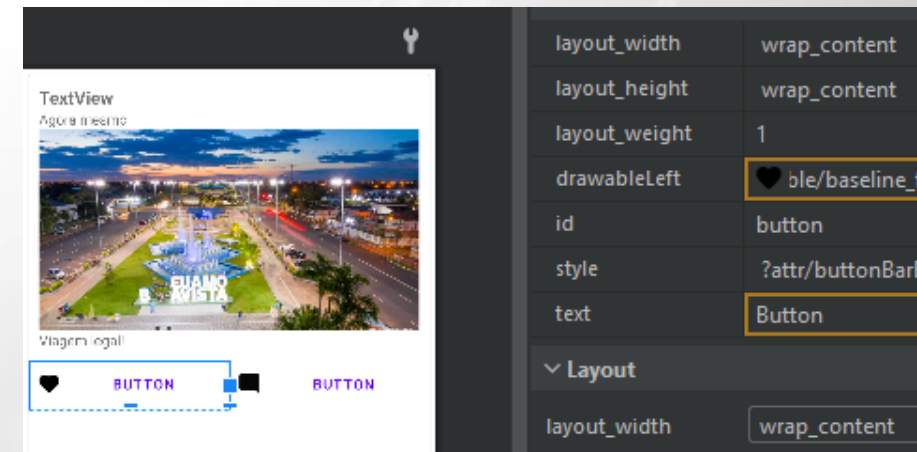
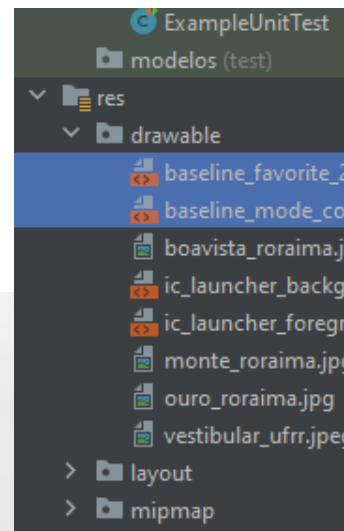
Crie uma estrutura dessa forma



Configurando



```
<Button
    android:id="@+id/button"
    style="?attr/buttonBarButtonStyle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:drawableLeft="@drawable/baseline_favorite_24"
    android:text="Button" />
```



Crie dentro de drawable, um New Vector Asset



Configurando a Visualização no Adapter

```
public class AdapterPostagem extends RecyclerView.Adapter<AdapterPostagem.ViewHolderImagens> {  
  
    @NonNull  
    @Override  
    public ViewHolderImagens onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  
        View itemlista = LayoutInflater.from(parent.getContext())  
            .inflate(R.layout.layout_postagem, parent, false);  
  
        return new ViewHolderImagens(itemlista);  
    }  
}
```

Crie o **itemlista** no método construtor da classe AdapterPostagem

Configure os atributos para receber os dataset



```
public class ViewHolderImagens extends RecyclerView.ViewHolder{  
  
    private TextView nome;  
    private TextView postagem;  
    private ImageView imagens;  
  
    public ViewHolderImagens(@NonNull View itemView) {  
        super(itemView);  
        nome = itemView.findViewById(R.id.textviewNome);  
        postagem = itemView.findViewById(R.id.textviewPostagem);  
        imagens = itemView.findViewById(R.id.imageViewPostagem);  
    }  
}
```



Configuran OnbindViewHolder

```
17     @NonNull
18     @Override
19     public ViewHolderImagens onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
20         View itemlista = LayoutInflater.from(parent.getContext())
21             .inflate(R.layout.layout_postagem, parent, attachToRoot: false);
22
23         return new ViewHolderImagens(itemlista);
24     }
25
26     @Override
27     public void onBindViewHolder(@NonNull ViewHolderImagens holder, int position) {
28         holder.postagem.setText("Computação");
29         holder.nome.setText("#tbt Viagem legal");
30         holder.imagens.setImageResource(R.drawable.boavista_roraima);
31     }
32
33     @Override
34     public int getItemCount() {
35         return 6;
36     }
```

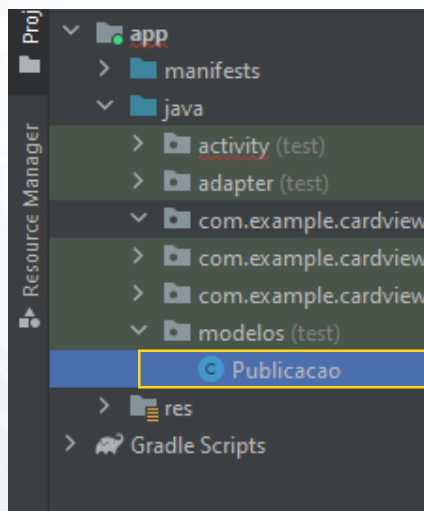




Visualização dinamica

Crie uma listagem para guardar esses itens

```
activity_main.xml x MainActivity.java x layout_p
15 public class MainActivity extends App
16
17 3 usages
18 private RecyclerView postagem;
19 private List<> postgens;
20
21 @Override
22 protected void onCreate(Bundle savedInstanceState) {
23     super.onCreate(savedInstanceState);
24     setContentView(R.layout.activity_main);
25 }
```



Para isso crie uma New Class dentro de modelos

Construtor

Gets e Sets

```
package modelos;

public class Publicacao {
    String nome;
    String postagem;
    int Imagens;

    public Publicacao(){

    }

    public String getNome() {
        return nome;
    }

    public void setNome(String nome) {
        this.nome = nome;
    }

    public String getPostagem() {
        return postagem;
    }

    public void setPostagem(String postagem) {
        this.postagem = postagem;
    }

    public int getImagens() {
        return Imagens;
    }

    public void setImagens(int imagens) {
        Imagens = imagens;
    }
}
```



Feito isso você já pode

```
activity_main.xml x MainActivity.java x layout_p
15 public class MainActivity extends App
16
17 3 usages
18 private RecyclerView postagem;
19 private List<> postgens;
20
21 @Override
22 protected void onCreate(Bundle savedInstanceState) {
23     super.onCreate(savedInstanceState);
24     setContentView(R.layout.activity_main);
25 }
```



```
activity_main.xml x MainActivity.java x Publicacao.java
2 usages
16 public class MainActivity extends AppCompatActivity {
17
18 3 usages
19 private RecyclerView postagem;
20 private List<Publicacao> postgens;
21
22 @Override
23 protected void onCreate(Bundle savedInstanceState) {
24     super.onCreate(savedInstanceState);
25     setContentView(R.layout.activity_main);
26 }
```

Crie dentro do MainAcitivit um método para gerar as postagens

```
public void prepararPostagem(){
    Publicacao p = new Publicacao("Computação", "tbt", R.drawable.ouro_roraima);
    this.post.add(p);
    p = new Publicacao("Estudar", "legal", R.drawable.vestibular_ufrn);
    this.post.add(p);
    p = new Publicacao("Mecânica", "aventura", R.drawable.ouro_roraima);
    this.post.add(p);
    p = new Publicacao("Minha cidade", "diversão", R.drawable.boavista_roraima);
    this.post.add(p);
}
```



Crie um objeto List Publicações

```
import java.util.List;

import adapter.AdapterPostagem;
import modelos.Publicacao;

2 usages
public class MainActivity extends AppCompatActivity {

    3 usages
    private RecyclerView postagem;

    5 usages
    private List <Publicacao> post;
```

Em AdapterPostagem acrescente o objeto List Publicacao

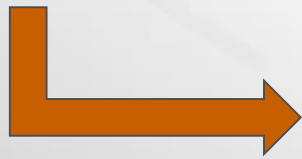
```
public class AdapterPostagem extends RecyclerView.Adapter<AdapterPostagem.ViewHolderImagens> {

    private List <Publicacao> post;
    public AdapterPostagem(List<Publicacao> p){
        this.post = p;
    }
}
```



O método `onBindViewHolder` será alterado para

```
}  
  
@Override  
public void onBindViewHolder(@NonNull ViewHolderImagens holder, int position) {  
    holder.postagem.setText("Computação");  
    holder.nome.setText("#tbt Viagem legal");  
    holder.imagens.setImageResource(R.drawable.boavista_roraima);  
}
```



Para isso

```
        return new ViewHolderImagens(itemlista);  
    }  
  
    @Override  
    public void onBindViewHolder(@NonNull ViewHolderImagens holder, int position) {  
        Publicacao pt = post.get(position);  
        holder.postagem.setText(pt.getPostagem());  
        holder.nome.setText(pt.getNome());  
        holder.imagens.setImageResource(pt.getImagens());  
    }  
  
    @Override
```




Crie um método
prepararPostagem
dentro de MainActivity

Crie também uma
lista, onde será
passado os atributos
da Postagem.

Crie o método
preparaPostagem
dentro do construtor
de ActivityMain

```
activity_main.xml × layout_postagem.xml × MainActivity.java × Postagem.java × AdaptadorPostagem.java ×
2 usages
20 AdaptadorPostagem adp;
3 usages
21 private List<Postagem> listpt = new ArrayList<>();
22
23 @Override
24 protected void onCreate(Bundle savedInstanceState) {
25     super.onCreate(savedInstanceState);
26     setContentView(R.layout.activity_main);
27
28     this.prepararPostagem();
29
30     p = findViewById(R.id.recyclerView);
31     adp = new AdaptadorPostagem();
32
33     RecyclerView.LayoutManager l = new LinearLayoutManager(context, this);
34     p.setLayoutManager(l);
35     p.setAdapter(adp);
36 }
1 usage
37 public void prepararPostagem(){
38     Postagem p = new Postagem( nomePostgaem: "Computação", postPostagem: "comp", R.drawable.imagembv);
39     this.listpt.add(p);
40     p = new Postagem( nomePostgaem: "Elétrica", postPostagem: "eltr", R.drawable.imagembv);
41     this.listpt.add(p);
42     p = new Postagem( nomePostgaem: "Matemática", postPostagem: "mat", R.drawable.imagembv);
43     this.listpt.add(p);
44 }
```



Criar um objeto do tipo Postagem para recuperar essas postagens

```
activity_main.xml x layout_postagem.xml x MainActivity.java x Postagem.java x AdaptadorPostagem.java x
public class AdaptadorPostagem extends RecyclerView.Adapter<AdaptadorPostagem.ViewHolder> {
    2 usages
    private List<Postagem> postg;
    1 usage
    public AdaptadorPostagem(List<Postagem> p) {
        this.postg = p;
    }

    @NonNull
    @Override
    public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        View itens = LayoutInflater.from(parent.getContext())
            .inflate(R.layout.layout_postagem, parent, attachToRoot: false);
        return new ViewHolder(itens);
    }

    @Override
    public void onBindViewHolder(@NonNull ViewHolder holder, int position) {
        Postagem publicacao = postg.get(position);
        holder.post.setText(publicacao.getPostPostagem());
        holder.nome.setText(publicacao.getNomePostgaem());
        holder.imag.setImageResource(publicacao.getImagemPostagem());
    }

    @Override
    public int getItemCount() { return 6; }
}
```

Crie um construtor em AdaptadorPostagem para passar essas postagens como parametro



No método getItemCount, altere
o return para postg.size();

```
activity_main.xml × layout_postagem.xml × MainActivity.java × Postagem.java × AdaptadorPostagem.java ×  
3 usages  
20 private List<Postagem> postg;  
1 usage  
21 public AdaptadorPostagem(List<Postagem> p) {  
22     this.postg = p;  
23 }  
24 @NonNull  
25 @Override  
26 public MayViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  
27     View itens = LayoutInflater.from(parent.getContext())  
28         .inflate(R.layout.layout_postagem, parent, attachToRoot: false);  
29     return new MayViewHolder(itens);  
30 }  
31 @Override  
32 public void onBindViewHolder(@NonNull MayViewHolder holder, int position) {  
33     Postagem publicacao = postg.get(position);  
34     holder.post.setText(publicacao.getPostPostagem());  
35     holder.nome.setText(publicacao.getNomePostgaem());  
36     holder.imag.setImageResource(publicacao.getImagemPostagem());  
37 }  
38 @Override  
39 public int getItemCount() {  
40     return postg.size();  
41 }  
42 }  
43 }  
44 }
```



Agora você irá passar o objeto lista para o AdaptadorPostagem

Agora sim você já pode executar seu App

```
activity_main.xml × layout_postagem.xml × MainActivity.java × Postagem.java × AdaptadorPostagem.java
2 usages
20 AdaptadorPostagem adp;
4 usages
21 private List<Postagem> listpt = new ArrayList<>();
22
23 @Override
24 protected void onCreate(Bundle savedInstanceState) {
25     super.onCreate(savedInstanceState);
26     setContentView(R.layout.activity_main);
27
28     this.prepararPostagem();
29
30     p = findViewById(R.id.recyclerView);
31     adp = new AdaptadorPostagem(listpt);
32
33     RecyclerView.LayoutManager l = new LinearLayoutManager(context, this);
34     p.setLayoutManager(l);
35     p.setAdapter(adp);
36 }
1 usage
37 public void prepararPostagem(){
38     Postagem p = new Postagem(nomePostgaem: "Computação", postPostagem: "comp", R.drawable.i
39     this.listpt.add(p);
40     p = new Postagem(nomePostgaem: "Elétrica", postPostagem: "eltr", R.drawable.i
41     this.listpt.add(p);
42     p = new Postagem(nomePostgaem: "Matemática", postPostagem: "mat", R.drawable.i
43     this.listpt.add(p);
```




Layout e orientação

Aqui as imagens aparecem uma ao lado da outra

```
LinearLayoutManager l = new LinearLayoutManager(this);  
l.setOrientation(RecyclerView.HORIZONTAL);
```

Aqui as imagens aparecem em colunas

```
RecyclerView.LayoutManager l = new GridLayoutManager(this, 2);  
p.setLayoutManager(l);
```

Substitua essa linha de código por uma dessas para ver o resultado.

```
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    this.prepararPostagem();  
  
    p = findViewById(R.id.recyclerView);  
    adp = new AdaptadorPostagem(listpt);  
  
    RecyclerView.LayoutManager l = new LinearLayoutManager(context, this);  
  
    p.setLayoutManager(l);  
    p.setAdapter(adp);  
}  
  
1 usage  
public void prepararPostagem(){  
    Postagem p = new Postagem( nomePostgaem: "Computação", postPostagem: "comp", R.drawable.imag  
    this.listpt.add(p);
```