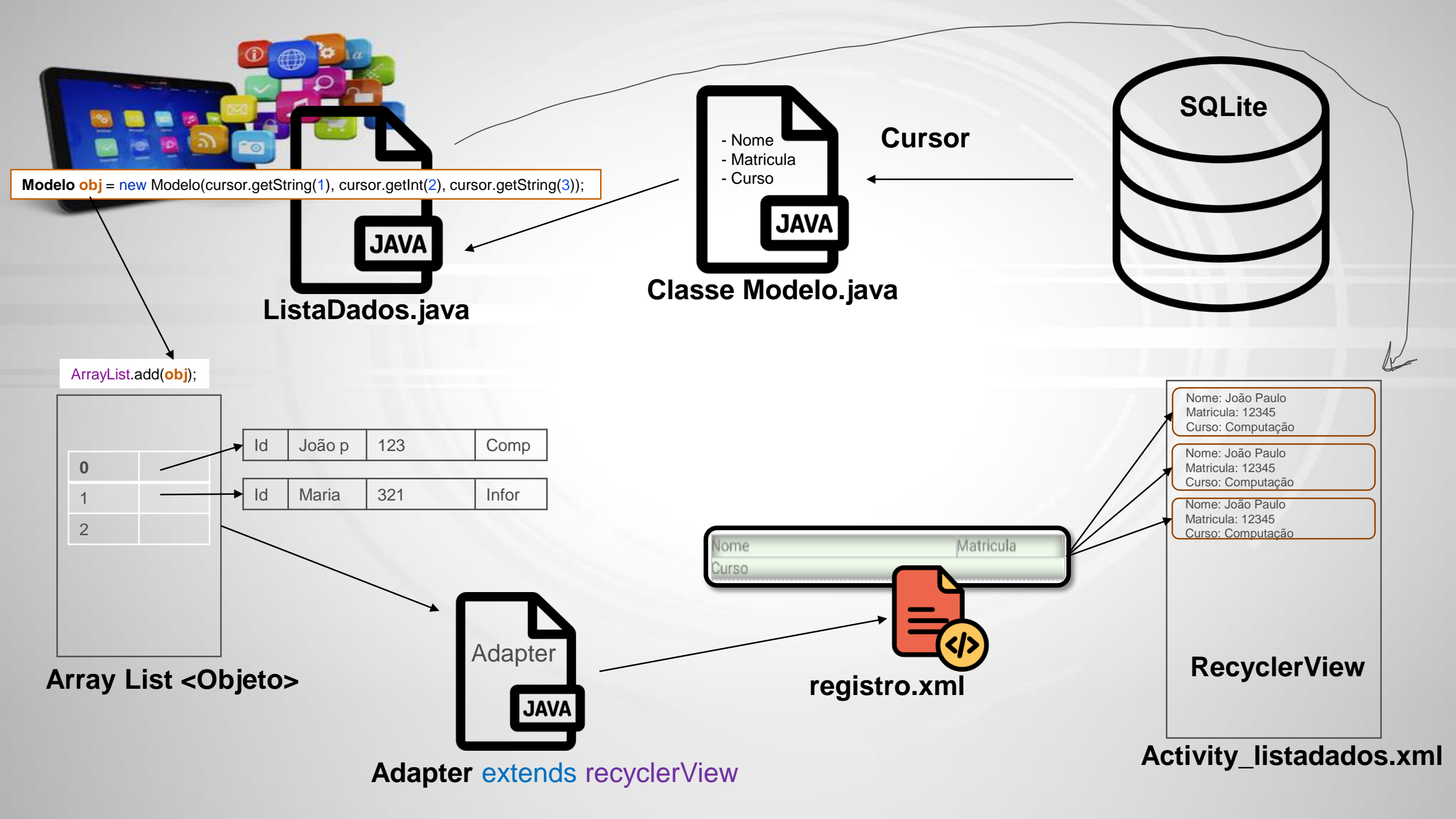
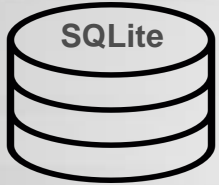




SQLite com RecyclerView: Formulário de cadastro





Class Banco.java

Criar banco de dados

Criar tabelas

Excluir e Atualizar tabelas

INSERT

SELECT

```
public class Banco extends SQLiteOpenHelper {  
  
    public Banco(@Nullable Context context) {  
        super(context, "dbalunos", null, 1);  
    }  
  
    @Override  
    public void onCreate(SQLiteDatabase sqLiteDatabase) { ... }  
  
    @Override  
    public void onUpgrade(SQLiteDatabase sqLiteDatabase, int i, int i1) { ... }  
  
    public String salvarAluno(String nome, Integer matricula, String curso) { ... }  
  
    public Cursor buscarAlunos() { ... }  
}
```

```
String sql = "CREATE TABLE IF NOT EXISTS tbaluno (" +  
    "id INTEGER PRIMARY KEY AUTOINCREMENT," +  
    "nome VARCHAR," +  
    "matricula INTEGER," +  
    "curso VARCHAR)";  
sqLiteDatabase.execSQL(sql);
```

```
String sql = "DROP TABLE IF EXISTS tbauno";  
sqLiteDatabase.execSQL(sql);  
onCreate(sqLiteDatabase);
```

```
SQLiteDatabase db = this.getWritableDatabase();  
  
ContentValues cv = new ContentValues();  
cv.put("nome", nome);  
cv.put("matricula", matricula);  
cv.put("curso", curso);  
float res = db.insert("tbaluno", null, cv);  
  
if(res == -1){  
    return "Flaha na inserção do registro";  
}else {  
    return "Registro inserido com sucesso!";  
}
```

```
SQLiteDatabase db = this.getWritableDatabase();  
  
String sql= "SELECT * FROM tbaluno";  
Cursor cursor = db.rawQuery(sql, null);  
return cursor;
```



Modelo.java

Essa classe é usada para fazer a relação entre a tabela do banco e o ArrayList, por meio dela os dados serão armazenados no ArrayList



Classe Modelo.java

Defina os parâmetros, e crie os Getter e Setter

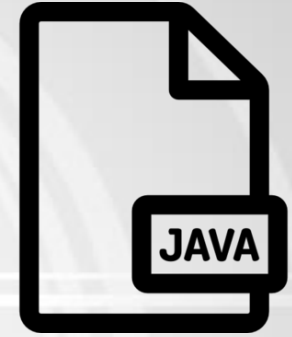
```
public class Modelo {  
    String nome;  
    Integer matricula;  
    String curso;  
  
    public Modelo(String nome, Integer matricula, String curso) {  
        this.nome = nome;  
        this.matricula = matricula;  
        this.curso = curso;  
    }  
  
    public String getNome() {  
        return nome;  
    }  
  
    public void setNome(String nome) {  
        this.nome = nome;  
    }  
  
    public Integer getMatricula() {  
        return matricula;  
    }  
  
    public void setMatricula(Integer matricula) {  
        this.matricula = matricula;  
    }  
  
    public String getCurso() {  
        return curso;  
    }  
  
    public void setCurso(String curso) {  
        this.curso = curso;  
    }  
}
```



Activity - ListarDados.java

Crie uma nova Activity – Empty Activity

New → Activity → Empty Activity



ListaDados.java

```
public class ListaDados extends AppCompatActivity {
```

```
    RecyclerView recyclerView;  
    ArrayList<Modelo> listaAlunos;
```

```
    @Override
```

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

Implementação ...

```
        recyclerView = findViewById(R.id.recyclerView);  
        recyclerView.setLayoutManager(new LinearLayoutManager(this));  
  
        Cursor cursor = new Banco(this).buscarAlunos( );  
        listaAlunos = new ArrayList<>( ); //Inicializar o ArrayList  
  
        while (cursor.moveToNext( )){  
  
            Modelo obj = new Modelo(cursor.getString(1), cursor.getInt(2), cursor.getString(3));  
            listaAlunos.add(obj);  
        }  
  
        MyAdapter adapter = new MyAdapter(listaAlunos);  
        recyclerView.setAdapter(adapter);
```




Activity - ListarDados.java



ListaDados.java

```
public class ListaDados extends AppCompatActivity {  
    RecyclerView recyclerView;  
    ArrayList<Modelo> listaAlunos;
```

Aqui é criado o ArrayList do Objeto Modelo

Adiciona o objeto ao ArrayList

Passa a lista de objeto para o Adapter

Carrega o Adapter no RecyclerView

```
Cursor cursor = new Banco(this).buscarAlunos( );
```

```
listaAlunos = new ArrayList<>( ); //Inicializar o ArrayList
```

```
while (cursor.moveToNext( )){
```

```
    Modelo obj = new Modelo(cursor.getString(1), cursor.getInt(2), cursor.getString(3));  
    listaAlunos.add(obj);
```

```
MyAdapter adapter = new MyAdapter(listaAlunos);
```

```
recyclerView.setAdapter(adapter);
```

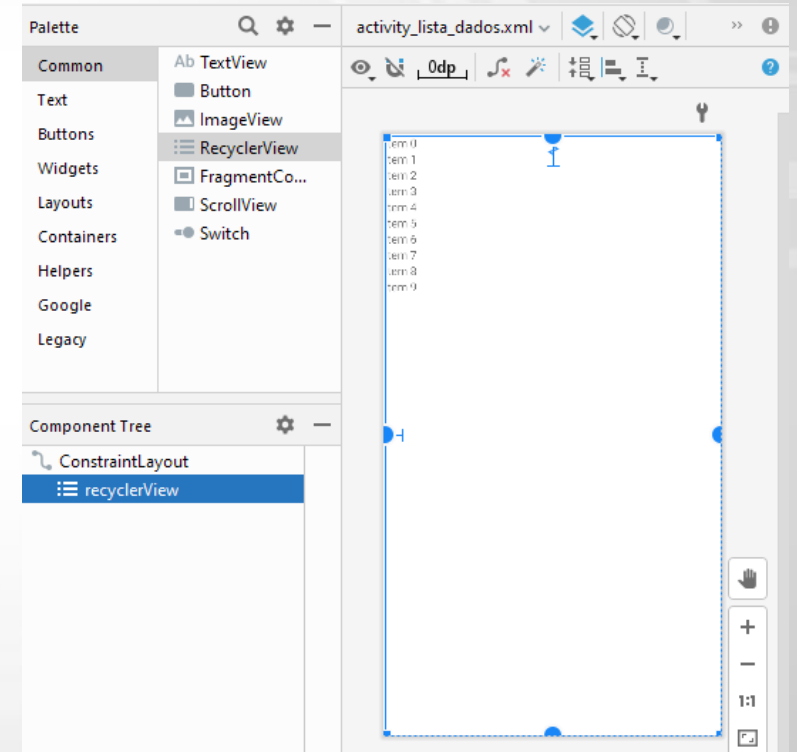


Activity_listadados.xml

Adicione o **RecyclerView** no Layout onde será carregado os registros inseridos da tabela aluno.



ListaDados.xml





Classe Adapter

A classe Adapter deve extender de `RecyclerView.Adapter`.
E o tipo deve ser `<MayAdapter.MayViewHolder>`

A classe `MayViewHolder` deve ser criada dentro da classe `Adapter`

Crie também o método construtor de receba como parâmetro o `ArrayList` de objetos

Em seguida implemente os métodos:

- `onCreateViewHolder`
- `onBindViewHolder`
- `getItemCount`



`MyAdapter.java`

```
public class MyAdapter extends RecyclerView.Adapter<MyAdapter.MayViewHolder>{

    ArrayList <Modelo> listaRegistros;

    public MyAdapter(ArrayList<Modelo> listaRegistros) {
        this.listaRegistros = listaRegistros;
    }

    class MayViewHolder extends RecyclerView.ViewHolder{

        TextView textNome, textMatricula, textCurso;

        public MayViewHolder(@NonNull View itemView) {
            super(itemView);
            textNome = itemView.findViewById(R.id.textViewNome);
            textMatricula = itemView.findViewById(R.id.textViewMatr);
            textCurso = itemView.findViewById(R.id.textViewCurso);
        }
    }
}
```




Classe Adapter



MyAdapter.java

onCreatViewHolder

```
public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  
    View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.registro, parent, false);  
    return new ViewHolder(view);  
}
```

Invoca o layout de registros



registro.xml

onBindViewHolder

```
public void onBindViewHolder(@NonNull ViewHolder holder, int position) {  
    holder.textNome.setText(listaRegistros.get(position).getNome());  
    holder.textMatricula.setText(String.valueOf(listaRegistros.get(position).getMatricula()));  
    holder.textCurso.setText(listaRegistros.get(position).getCurso());  
}
```

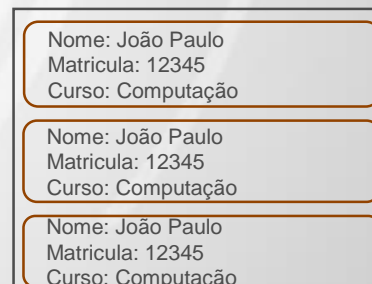
Carrega os dados no Layout



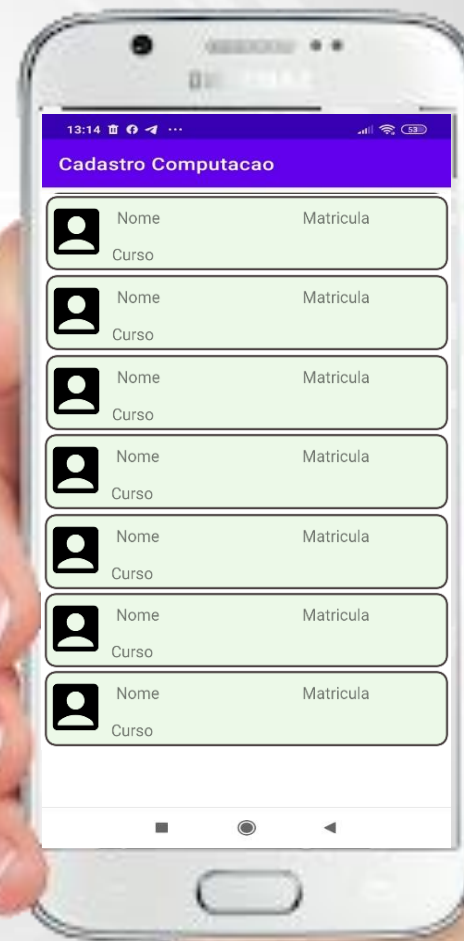
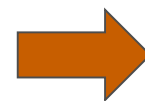
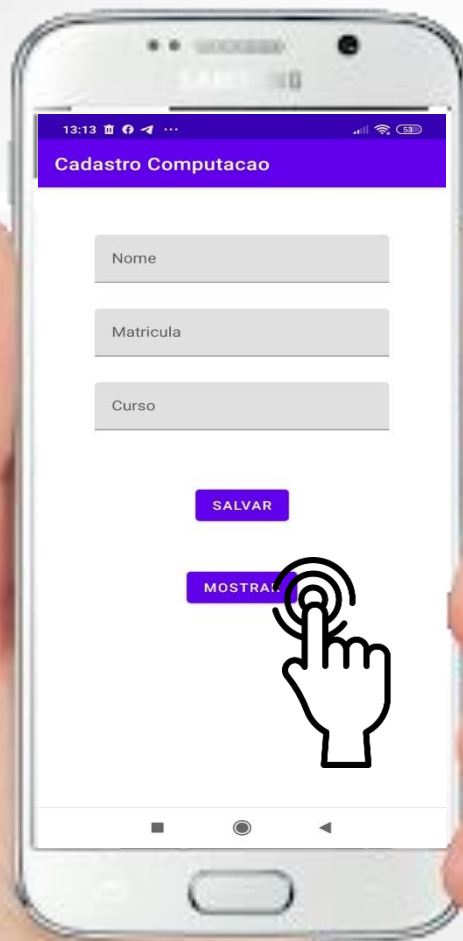
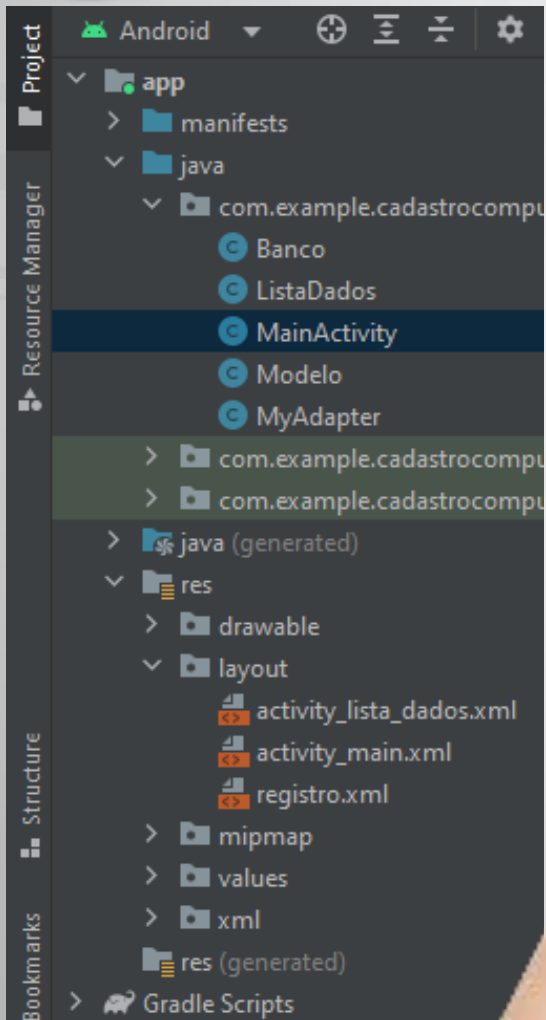
getItemCount

```
public int getItemCount() {  
    return listaRegistros.size();  
}
```

Define a quantidade de itens a ser exibidos



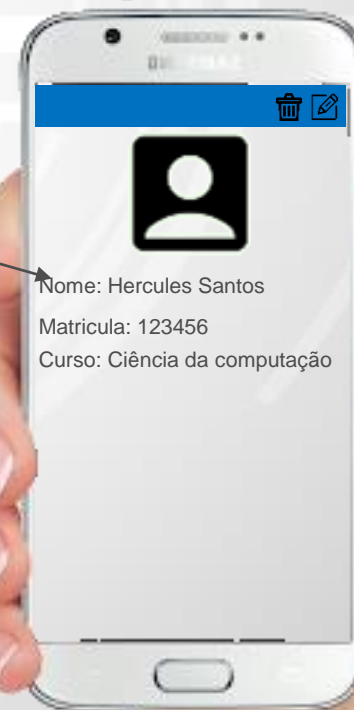
RecyclerView





Atividade – Dados pessoais

Ao clicar em um item, abrir uma nova activity com os dados pessoais





Criando Evento de Clicker no RecyclerView

1 – Crie uma Interface para lidar com eventos de clique nos itens do RecyclerView em New → JavaClass → Interface

```
public interface OnItemClickListener {  
  
    void onItemClick(Modelo item);  
  
}
```

2 – No MyViewHolder do seu Adapter, defina um método bind() para associar os métodos do modelo ao item do RecyclerView

```
public void bind(final Modelo item, final OnItemClickListener listener) {  
    textNome.setText(item.getNome());  
    textMatricula.setText(String.valueOf(item.getMatricula()));  
    textCurso.setText(item.getCurso());  
  
    itemView.setOnClickListener(new View.OnClickListener() {  
        @Override  
        public void onClick(View view) {  
            listener.onItemClick(item);  
        }  
    });  
}
```



Criando Evento de Clicker no RecyclerView

3 – No Adapter, declare o **listener** e acrescente como parâmetro no método construtor.

```
private OnItemClickListener listener;
```

```
public MyAdapter(ArrayList<Modelo> listaRegistros, OnItemClickListener listener) {  
    this.listaRegistros = listaRegistros;  
    this.listener = listener;  
}
```

4 – No método **onBindViewHolder()** do seu Adapter, chame o método **bind()** do ViewHolder e passe a instância da interface OnItemClickListener:

```
Modelo item = listaRegistros.get(position);  
holder.bind(item, listener );
```




Criando Evento de Clicker no RecyclerView

5 – No Activity (método **onCreate()**) da) que contem o RecyclerView, implemente a interface **OnItemClickListener** e sobrescreva o método **onItemClick()**.

```
public class ListaDados extends AppCompatActivity implements.OnItemClickListener {
```

```
@Override  
public void onItemClick(Modelo item) {  
    Intent intent = new Intent(this, DadosPessoais.class);  
    intent.putExtra("nome", item.getNome());  
    intent.putExtra("matricula", item.getMatricula());  
    intent.putExtra("matricula", item.getCurso());  
  
    startActivity(intent);  
}
```

inicie a nova Activity **DadosPessoais.class** e passe os dados relevantes como extras:

5.1 – Acrescente o parâmetro **listener: this**, ao setar o Adapter

```
MyAdapter adapter = new MyAdapter(listaAlunos, this);  
recyclerView.setAdapter(adapter);
```

Obs: DadosPessoais é a nova activity onde será exibido os dados individuais de cada item do recyclerView



Criando Evento de Clicker no RecyclerView

6 – Na Activity **DadosPessoais**, obtenha os extras passados e exiba os dados nos componentes de visualização apropriados:).

Declarar componentes

Inicializar componentes

Exibi os dados nos componentes

```
public class DadosPessoais extends AppCompatActivity {  
    TextView textNome, textMatricula, textCurso ;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_dados_pessoais);  
  
        textNome = findViewById(R.id.textViewNome);  
        textMatricula = findViewById(R.id.textViewMatricula);  
        textCurso = findViewById(R.id.textViewCurso);  
  
        Intent intent = getIntent();  
        if (intent != null) {  
            String nome = intent.getStringExtra("nome");  
            String matricula = intent.getStringExtra("matricula");  
            String curso = intent.getStringExtra("curso");  
  
            textNome.setText(nome);  
            textMatricula.setText(matricula);  
            textCurso.setText(curso);  
        }  
    }  
}
```



Atividade

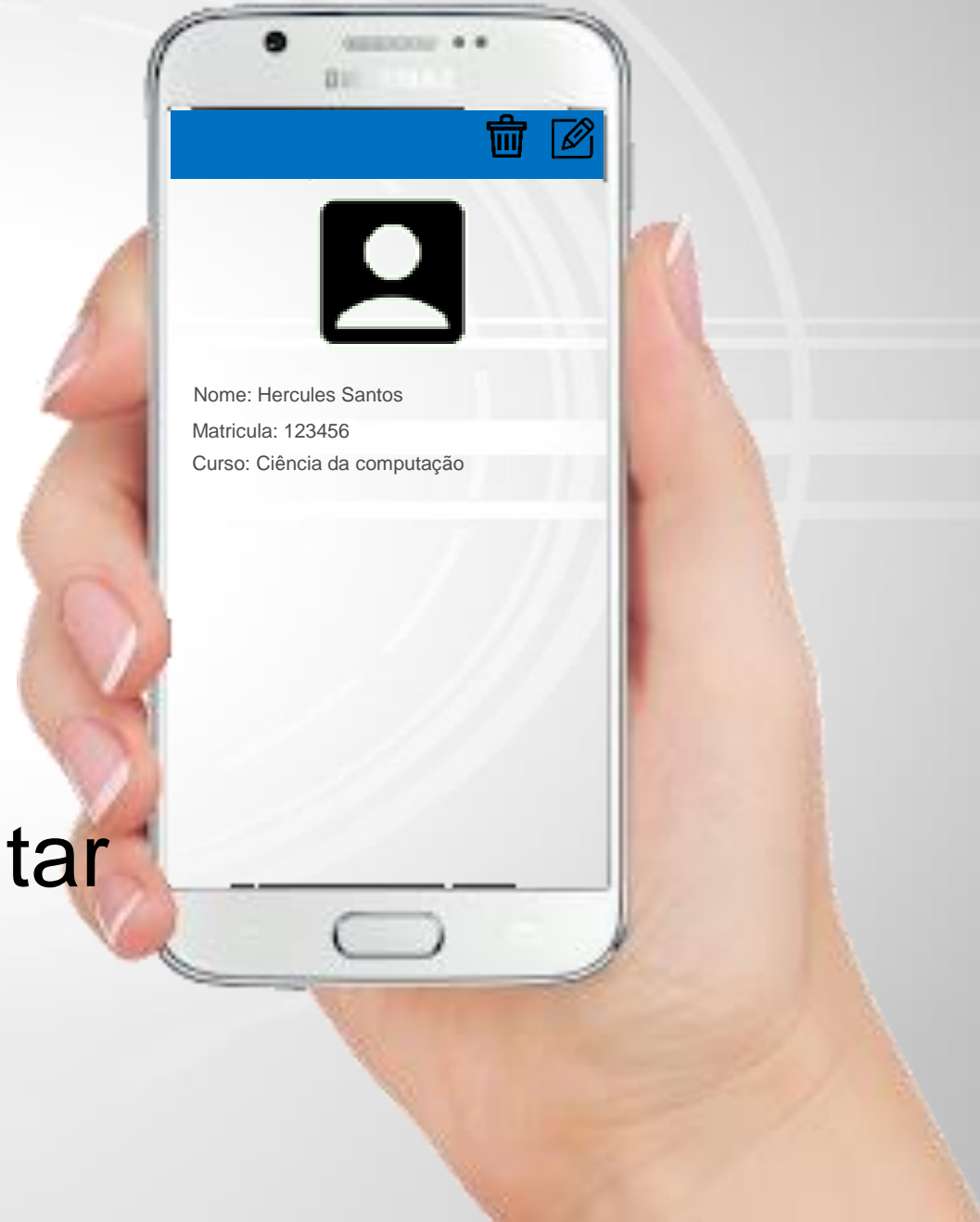


- Excluir



- Editar

Implemente a função excluir e editar



Entrega 26/06/2023
Pontuação: 10,0

