 <p>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</p>	<p>Computação Móvel e Ubíqua</p> <p>1º Semestre</p> <p>Docente: fas</p> <p>Ficha Prática: Room</p>
---------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------

Tema: *Room Database*

Objetivos:

- Utilização de bases de dados em Android
- Operações CRUD

PARTE 1

Exercício 1

Utilizando o IDE Android Studio crie um novo projeto com as seguintes características:

<i>Atributo</i>	<i>Valor</i>
<i>Application Name</i>	<i>ToolbarAndPreferences</i>
Project Location	Leave the default value
Form factor	Phone and tablet only
Minimum SDK	API 23 Marshmallow
Type of Activity	Empty
Activity Name	MainActivity (default)
Layout Name	Activity_main (default)

Deve implementar um projeto que faça uso dos seguintes componentes:

- Uma *MainActivity* que possua uma *Toolbar* e *RecyclerView* (ver Figura 1)
- A *Toolbar* deve definir um título customizado ("Lista de Contactos") e um *action menu* com um *icon* de adicionar. A ação do menu deverá abrir uma *ContactInsertActivity*
- A *RecyclerView* possui uma lista de contactos que devem ser obtidos de uma base de dados *Room*.
- Cada item contacto da *RecyclerView* possui um botão de "Eliminar" que elimina o contacto da base de dados.

- A *ContactInsertActivity* possui uma *Toolbar*, uma *EditText* que permita inserir o nome do contacto, uma segunda *EditText* que permita inserir o número de telemóvel do contacto e um Botão de inserir contacto (ver Figura 2)
- A *Toolbar* deve possuir um título personalizado ("Inserir Contacto") e um botão de voltar ao ecrã anterior.
- O botão de inserir deverá inserir os detalhes do contacto para a base de dados *Room*

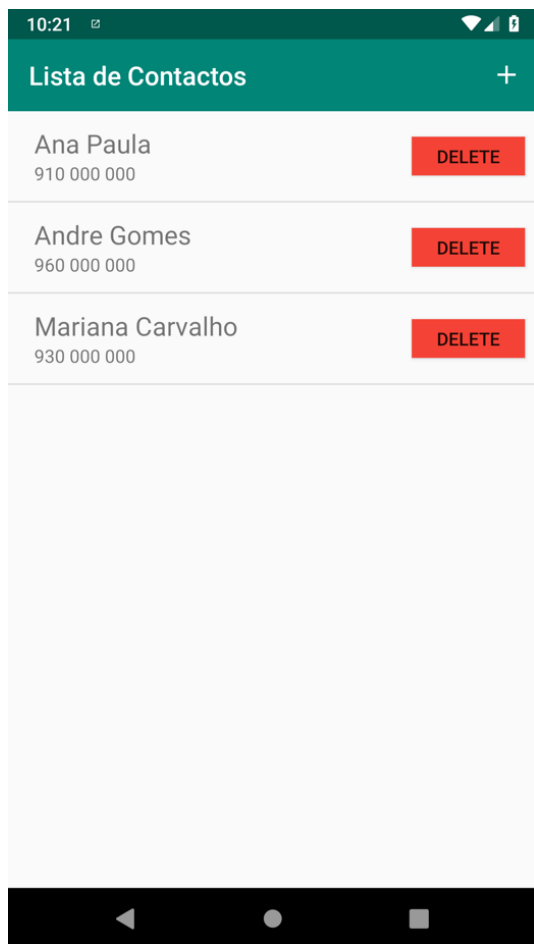


Figura 1 - MainActivity com RecyclerView

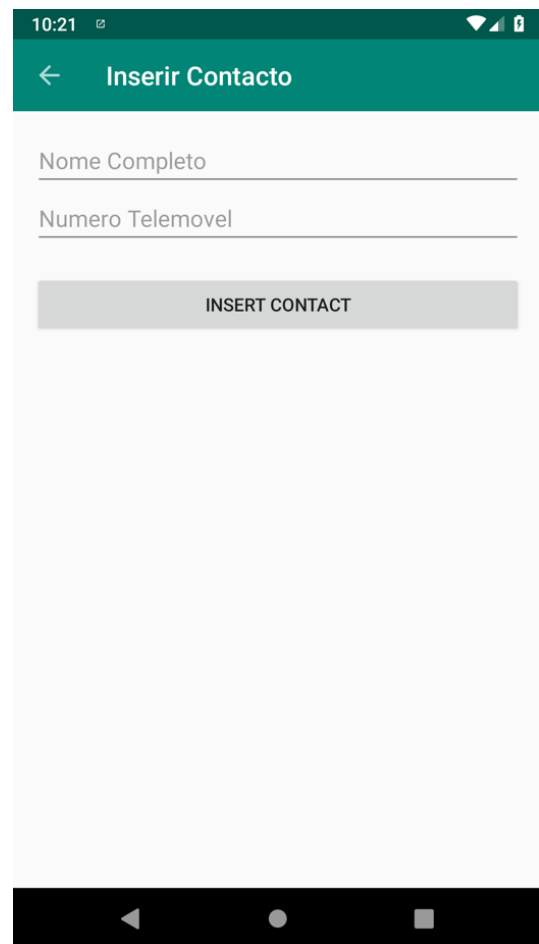




Figura 2 - ContactInsertActivity

Nota:

Para evitar fazer as operações sobre a base de dados na UI thread podemos usar o executor service para realizar operações em threads paralelas.

	ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO	Computação Móvel e Ubíqua 1º Semestre Docente: fas Ficha Prática: Room
-----------------------------------------------------------------------------------	-------------------------------------------------	----------------------------------------------------------------------------------------

```
public static final ExecutorService databaseWriteExecutor =  
    Executors.newFixedThreadPool(NUMBER_OF_THREADS);  
  
databaseWriteExecutor.execute(() -> {  
    //Código aqui será executado numa thread paralela  
});
```

 <p>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</p>	<p>Computação Móvel e Ubíqua</p> <p>1º Semestre</p> <p>Docente: fas</p> <p>Ficha Prática: Room</p>
---------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------

PARTE 2

Questão 1

Quais as vantagens de utilização de uma base de dados *Room*?

- Criação e gestão automática da uma base de dados SQLite em Android.
- Eliminação de código repetitivo.
- Ajuda na gestão de múltiplos *backends*.
- A utilização de um DAO fornece um mecanismo para mapear os métodos Java com *queries* à base de dados.

Questão 2

O que é um *DAO*?

- Acrónimo para *Data Access Object*.
- Uma biblioteca para gestão das *queries* à base de dados.
- Uma interface anotada que mapeia os métodos Java a *queries* SQLite.
- Uma classe cujos métodos irão ser sempre executados em segundo plano (*background*), não na *main thread*.
- Uma classe onde o compilador verifica a existência de erros em SQL, e que utiliza para gerar *queries* a partir das anotações.