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

Tema: Listas e *Adapters*

Objetivos:

- Criação/Utilização de *layouts* que apresentem listas
- Criação de *Adapters*
- Criação de *ListView* e *RecyclerView* personalizados

## PARTE 1 – *ArrayList* e *Adapters*

### 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>SpinnerList</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)

Adicione o(s) método(s) necessários:

- Criação de um ficheiro, array de dados, na pasta res > values para armazenar a informação relativa a um conjunto de países;
- Implementar uma actividade com um layout semelhante ao da Figura 1
- Após seleccionar um item no spinner, devemos desbloquear o botão e apresentar o país seleccionado numa textview numa segunda actividade

No final a aplicação deverá apresentar uma interface semelhante à Figura 1.

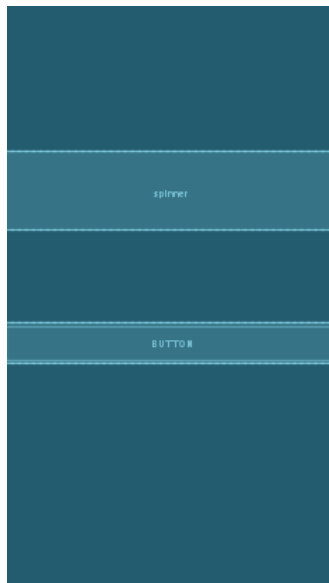



Figura 1 - Layout da MainActivity

## Exercício 2

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

<i>Atributo</i>	<i>Valor</i>
<i>Application Name</i>	<i>HugeArrayList</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)

Adicione o(s) método(s) necessários de modo a implementar as seguintes funcionalidades:

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

- *Activity* que utilize um *ArrayList* e com um *ArrayAdapter* personalizado
- Classe de dados para representar/armazenar a informação de cada item
- Cada elemento da lista deve conter uma imagem

Adicione mais de 500 elementos no *ArrayList* que foi declarado. Recorrendo ao *Android Profiler* verifique o consumo de memória da aplicação. Consegue verificar se existe degradação de *performance* da aplicação em comparação com a aplicação do exercício anterior?

## PARTE 2 – *RecyclerView*

### 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>RecyclerView</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)

Adicione o(s) método(s) necessários para adicionar as seguintes funcionalidades:

- *Activity* constituída por (ver Figura 2)
  - *FrameLayout*
  - *RecyclerView*
- Classe que represente o modelo de dados *Contacto* constituído por
  - Nome
  - Status online
  - Outra informação relevante
- *Adapter* que utilize uma lista de, pelo menos, 200 contactos. Este *Adapter* deve ser utilizado na *RecyclerView*
- *Separe os elementos da RecyclerView* com uma linha cinzenta
- Os contactos deverão ser distinguidos através do seu estado
  - *Online*: botão de enviar mensagem disponível e cor verde no status
  - *Offline*: botão não disponível e cor vermelha no status

- Quando clicado, o botão enviar mensagem deve abrir nova *Activity* com detalhes do contacto (Figura 3)

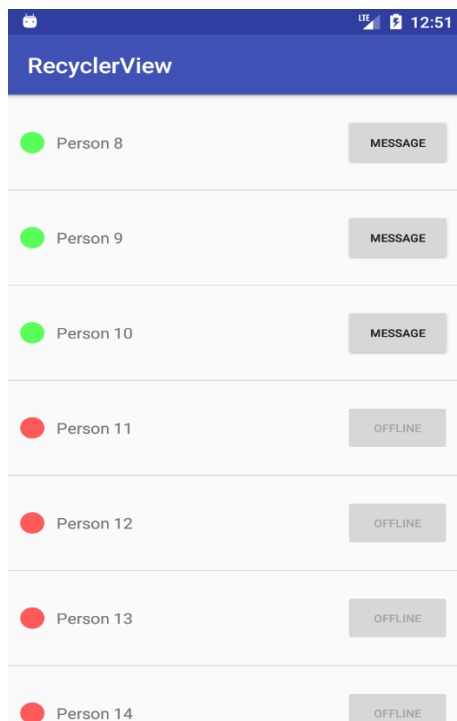


Figura 2 - Activity com RecyclerView

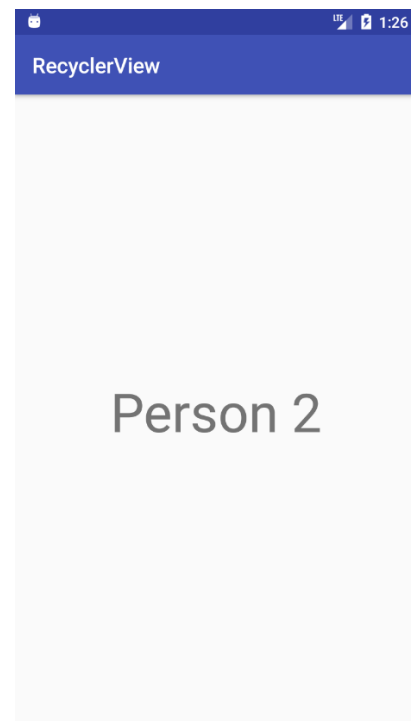


Figura 3 - Actividade com detalhes do Contacto

## Exercício 2

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

<i>Atributo</i>	<i>Valor</i>
<i>Application Name</i>	<i>RecyclerViewQuestions</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)

Implemente as seguintes funcionalidades:

- Activity constituída por um *ConstraintLayout* e uma *RecyclerView* (Figura 4)
- Na *RecyclerView* apresente as diferentes perguntas pelo título
- Cada pergunta (modelo de dados) deve ser constituída por
  - Título
  - Descrição
  - Resposta
  - Estado
- *Adapter* com lista de perguntas que deve utilizar na *RecyclerView*
- Elementos da *RecyclerView* separados por uma linha cinzenta
- Distinção das perguntas pelo seu estado
  - *Verde*: respondida com sucesso
  - *Vermelha*: respondida de forma errada
  - *Amarela*: por responder
- Botão jogar (presente em cada item) que abre uma nova *Activity* com os detalhes da pergunta e métodos que permitam ao utilizador inserir a resposta

- A inserção de uma resposta deverá atualizar o estado da *RecyclerView* de modo a alterar o status da questão respondida

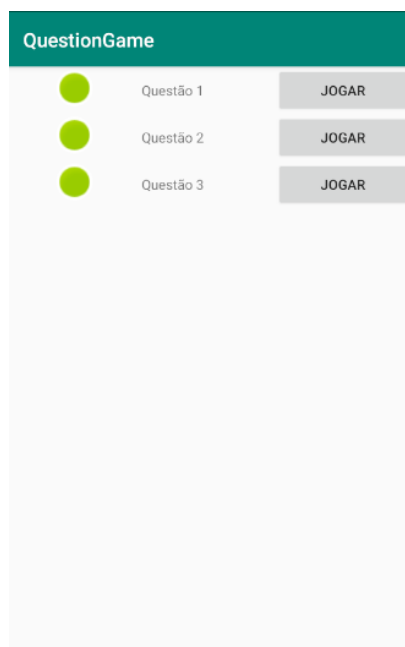


Figura 4 - Activity com RecyclerView

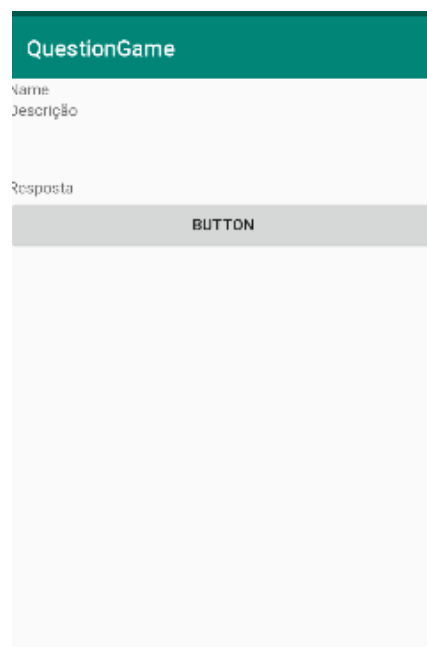



Figura 5 - Activity para inserção de respostas

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

## PARTE 3

### Questão 1

Qual das seguintes afirmações sobre *RecyclerView* é false? (selecione apenas uma)

- A *RecyclerView* é uma forma mais eficiente (considerando os recursos necessários) para apresentar listas com deslocação (*scroll*).
- É necessário fornecer um *Layout* apenas para um item da lista.
- Todos os elementos têm a mesma aparência.
- Não é necessário um *layout manager* para a *RecyclerView* lidar com a hierarquia e com o *rView* lidar com a hierarquia e com o *layout* de cada *View*.

### Questão 2

Que interface é necessário implementar de modo a criar eventos quando o utilizar clica na *RecyclerView*? Selecione uma.

- `View.OnClickListener`
- `RecyclerView.Adapter`
- `RecyclerView.ViewHolder`
- `View.onKeyListener`