

Implementando o Adapter inicial

Caso precise do projeto com as alterações realizadas na aula passada, você pode baixá-lo por meio [deste link](#) (<https://github.com/alura-cursos/android-alura-viagens-parte-1/archive/aula-4.zip>).

Agora que temos um layout próximo ao que o cliente solicitou, vamos para o próximo passo que é implementar o adapter da `ListView`.

Buscando a `ListView`

Para isso, dentro do `onCreate()` da Activity `ListaPacotesActivity`, logo depois da chamada do método `setContentView()`, busque a `ListView` por meio do método `findViewById()` enviando o id da mesma via parâmetro. Então, devolva para uma variável do tipo `ListView`.

Lembrando que a partir da versão 26, não é necessário utilizar o cast quando atribuímos um `findViewById()` para uma `View` mais específica, como é o caso do `ListView`.

Criando o Adapter

Com o objeto do tipo `ListView`, chame o método `setAdapter()` enviando a seguinte **intenção de instância** `new ListaPacotesAdapter()`, então, utilize o atalho **Alt + Enter** e execute a opção **create class 'ListaPacotesAdapter'**.

Nesse momento o AS vai pedir para indicar o pacote onde deseja criar a classe, deixe no pacote `br.com.alura.aluraviagens.ui.adapter`.

Nesta ação, provavelmente o AS vai fazer uma implementação da interface `ListAdapter`. Caso ocorrer esse detalhe, apague o trecho da implementação, pois utilizaremos uma outra abordagem.

Configurando o Adapter

Como já vimos, para criar um adapter faça a extensão da classe `BaseAdapter` e sobrescreva os seguintes métodos:

- `getCount()`
- `getItem()`
- `getItemId()`
- `getView()`

Entretanto, alguns desses métodos precisam de uma coleção e uma referência da classe `Context` para a implementação, no nosso caso, uma lista de pacotes.

Adicionando o modelo do pacote

Antes de declarar os atributos necessários, adicione o [modelo Pacote](#) (<https://raw.githubusercontent.com/alura-cursos/android-alura-viagens-parte-1/aula-5/app/src/main/java/br/com/alura/aluraviagens/model/Pacote.java>) dentro do pacote `br.com.alura.aluraviagens.model`

Recebendo os parâmetros necessários

Em seguida, no construtor do adapter, envie um parâmetro do tipo `List<Pacote>` e `Context` e associe com um atributo para cada um deles.

Implementando os métodos

Então, devolva o tamanho da lista no método `getCount()`, então, no método `getItem()` devolva um pacote por meio do método `get()` da lista de pacotes enviando a posição como parâmetro.

No `getItemId()` não há a necessidade de modificar a implementação, pois o nosso objeto alvo (`Pacote`) não possui um id, portanto, pode devolver `0` sem nenhum problema.

No método `getView()`, crie uma view a partir do layout `item_pacote` chamando o método `inflate()` da chamada estática `LayoutInflater.from()` enviando o atributo do tipo `Context` como parâmetro do `from()`.

Lembre-se que `inflate()` recebe os seguintes parâmetros `int resource`, `ViewGroup root`, `boolean attachToRoot`. Envie sucessivamente o `R.layout.item_pacote`, `parent`, `false`. Então, atribua essa chamada para uma variável do tipo `View` e a retorne no método `getView()`.

Neste exercício haverá problemas de compilação que serão resolvidos no exercício posterior, portanto, não é necessário executar a App.

