

Aplicações Móveis

Aula 9 - RecyclerView e CardView

Prof. Dr. Wendell Fioravante da Silva Diniz
3º Ano - Informática Integrado
Centro Federal de Educação Tecnológica de Minas Gerais
Unidade Varginha

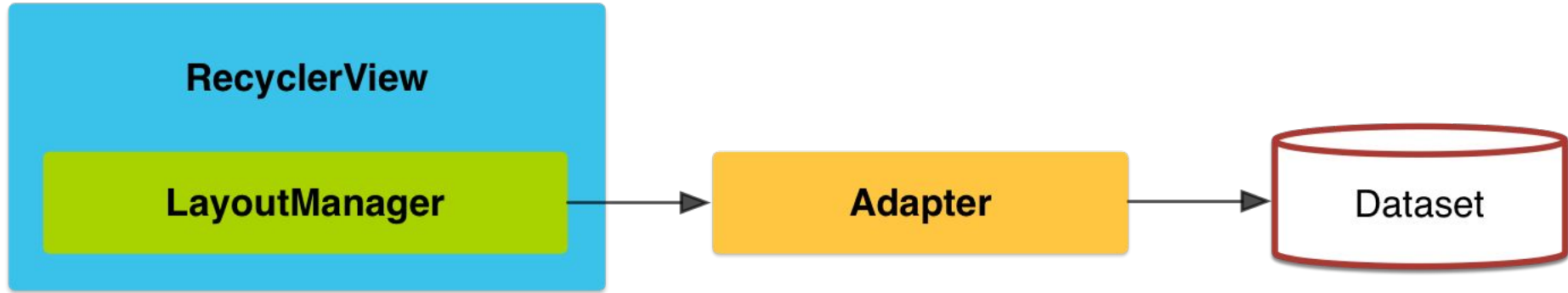
Recapitulando...

- A Toolbar substitui a ActionBar, introduzindo novas funcionalidades
- Serve para prover acesso a atividades da aplicação, mostrar ações frequentes e organizar a navegação
- Pode ser dividida em áreas diferentes da tela
- Agrupa as ações mais comuns em uma gaveta
- O provedor de compartilhamento facilita compartilhar itens

O RecyclerView

- Desde o início do Android, o ListView foi a view padrão para a exibição de listas
- Algumas desvantagens do ListView
 - Suporta apenas listas verticais
 - Não tem efeitos ou animações, estas deveriam ser implementadas pelo desenvolvedor
 - A otimização da rolagem também ficava por conta do desenvolvedor
- Quando o Material Design foi introduzido, um novo elemento de lista foi lançado, o RecyclerView
- Ele passa a ser a view recomendada para exibição de listas
- Adiciona funções como:
 - Animações para inserção e remoção de itens
 - Controle automático da reutilização (otimização de rolagem)
 - Permite alteração do layout para exibir listas, grades ou layouts customizados

Estrutura do RecyclerView



Como usar o RecyclerView?

- Incluir o componente no layout da atividade
- Informar uma subclasse de RecyclerView.LayoutManager
- Informar uma classe de animação de item

Como usar o RecyclerView?

Buscando a referência no layout

```
1 RecyclerView lista = findViewById(R.id.recyclerView);  
2 RecyclerView.LayoutManager myLayoutManager = new LinearLayoutManager(getActivity());  
3 lista.setLayoutManager(myLayoutManager);  
4 lista.setItemAnimator(new DefaultItemAnimator());
```

Como usar o RecyclerView?

Declarando um LayoutManager

```
1 RecyclerView lista = findViewById(R.id.recyclerView);  
2 RecyclerView.LayoutManager myLayoutManager = new LinearLayoutManager(getActivity());  
3 lista.setLayoutManager(myLayoutManager);  
4 lista.setItemAnimator(new DefaultItemAnimator());
```

Como usar o RecyclerView?

Passando o LayoutManager para o RecyclerView

```
1 RecyclerView lista = findViewById(R.id.recyclerView);  
2 RecyclerView.LayoutManager myLayoutManager = new LinearLayoutManager(getActivity());  
3 lista.setLayoutManager(myLayoutManager);  
4 lista.setItemAnimator(new DefaultItemAnimator());
```


Como usar o RecyclerView?

Criando e passando um animador padrão para o RecyclerView

```
1 RecyclerView lista = findViewById(R.id.recyclerView);  
2 RecyclerView.LayoutManager myLayoutManager = new LinearLayoutManager(getActivity());  
3 lista.setLayoutManager(myLayoutManager);  
4 lista.setItemAnimator(new DefaultItemAnimator());
```

Gerenciando os dados

- O RecyclerView é muito bom em trabalhar com um volume muito grande de dados de forma eficiente
- Ele usa o mesmo conceito dos Adapters (adaptadores de dados) que o ListView introduziu
- O adaptador pode ser customizado normalmente, mas deve agora ser subclasse de RecyclerView.Adapter

Gerenciando os dados

```
1 List<Pessoa> pessoas = new List<>();  
2 lista.setAdapter(new PessoaAdapter(this, pessoas, onClickPessoa()));
```

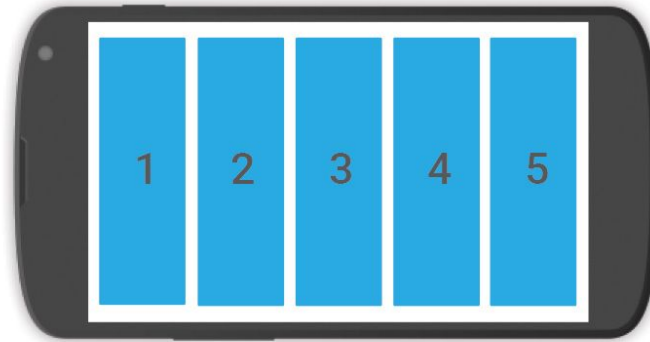
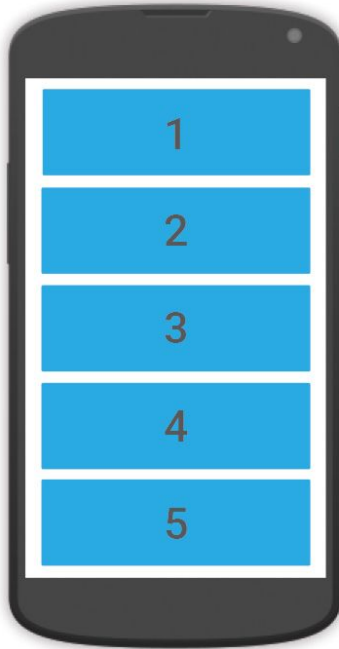
LayoutManager

- O RecyclerView é um componente especializado no reaproveitamento das views, ou seja recicla as views automaticamente
- Diferentemente do ListView, que acumula as funções de gerenciar a reciclagem e desenhar a interface, o RecyclerView não sabe desenhar
- Para isso, usamos um LayoutManager
- Podemos usar um dos tipos predefinidos ou criar nosso próprio item customizado:
 - LinearLayoutManager
 - GridLayoutManager
 - StaggerGridLayoutManager

LinearLayoutManager

- Organiza as views em uma sequência
- Pode ser definido vertical ou horizontal
- Para replicar o comportamento do ListView, use este gerenciador

LinearLayoutManager

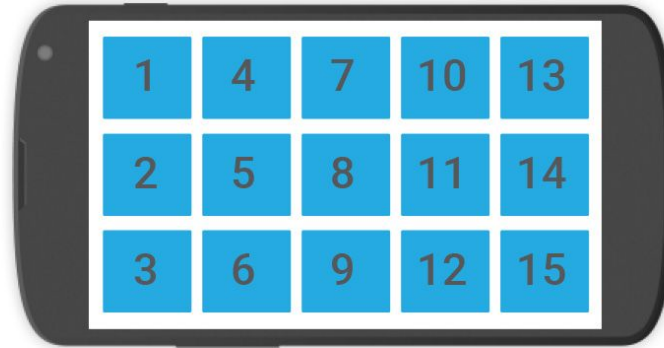
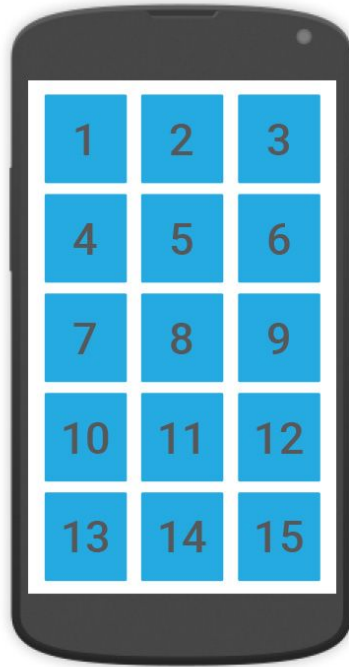


Linear Layout Manager

GridLayoutManager

- Organiza as views em uma grade
- A visualização é uniforme

GridLayoutManager

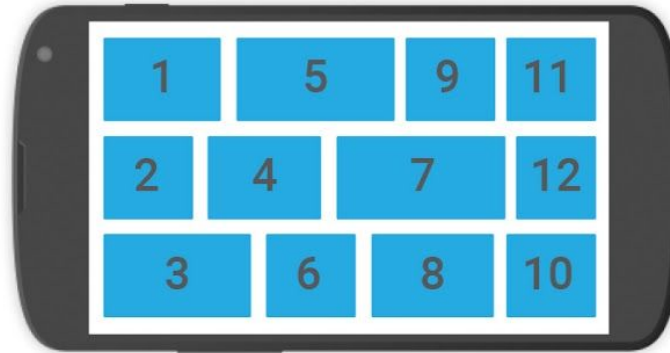
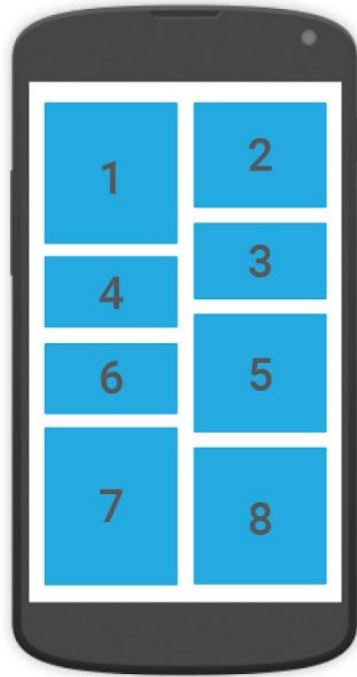


Grid Layout Manager

StaggeredGridLayoutManager

- Organiza as views em um grid que suporta orientação tanto vertical quanto horizontal
- Os itens não precisam ter o mesmo tamanho

StaggeredGridLayoutManager



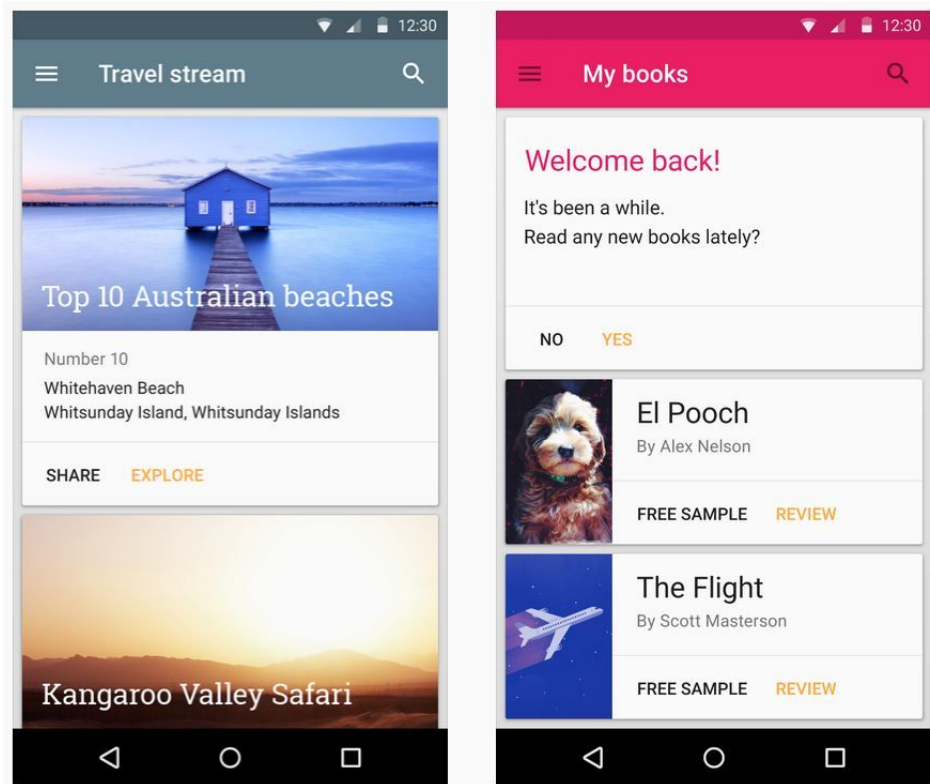
Staggered Grid
Layout Manager

Animações

- Outra melhoria do RecyclerView é o suporte a animações
- Uma animação é exibida quando um item da lista é removido ou adicionado
- As animações devem ser filhas de RecyclerView.ItemAnimator
- A classe DefaultItemAnimator oferece animações padrão

O CardView

- O CardView é outro widget que foi introduzido com o Material Design
- Foi criado para oferecer uma forma de visualização consistente em todas as plataformas
- Para utilizá-lo plenamente, deve-se seguir as recomendações do Material Design
- O CardView é filho de FrameLayout, ou seja, é um container onde pode-se adicionar outras views



Leitura complementar recomendada

- Capítulo 11.8 do livro **Google Android** de Ricardo Lecheta - sobre RecyclerView
- Capítulo 11.7 - sobre CardView

