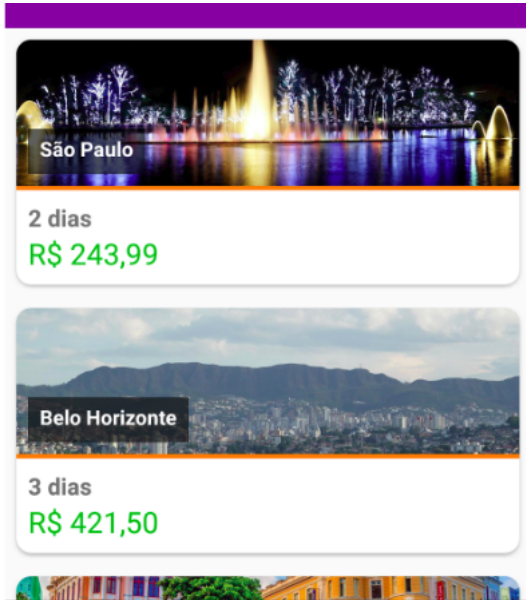


## Ajustando o CardView

Com a amostra adicionada percebemos que o layout de cada item que representa o pacote não bate de acordo com o que o nosso cliente solicitou, então o nosso foco será deixar o layout próximo à imagem abaixo:



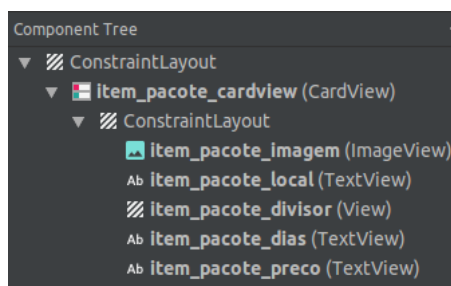
Veja que temos esse container com sombra e bordas arredondas. Para termos o mesmo aspecto visual precisamos adicionar uma view conhecida como `CardView`. Para uma inserção mais objetiva, considere o editor visual e vá na lista de componentes **AppCompat** > **CardView**, então, arraste e solte no meio do preview do layout.

Neste momento o AS vai adicionar o `CardView` via dependência do projeto, portanto, será necessário aguardar o processo finalizar.

### Ajustando o layout com o `CardView`

Com o `CardView` adicionado no projeto, o próximo passo é organizar o layout para que mantenha o mesmo aspecto visual considerando a necessidade do cliente.

Para uma abordagem mais intuitiva, considere o uso do **Component tree** e deixe o layout com a seguinte organização:



Perceba que nesta mudança foi adicionado mais um `ConstraintLayout`, ele é necessário pois todas as views foram para dentro do `CardView`, ou seja, as constraints que foram aplicadas não funcionarão como esperado.

É importante notar que nesta situação ocorreu um pequeno nível de hierarquia no qual fez com que as views fossem pra dentro do `CardView`. Isso significa que nem sempre teremos uma view totalmente flat com o `ConstraintLayout` para manter alguns aspectos visuais dependendo da view que utilizarmos.

## Ajustando o cardview

Neste exercício adicione as constraints para o `CardView` para que o layout dele fique próximo à imagem que está no começo do enunciado.

Não se preocupe se aparecer alguma divergência, a ideia deste exercício é realizar uma implementação objetiva para ajustarmos os detalhes dela mais pra frente :)

## Aplicando os atributos

De atributos, deixe a largura com o `match_constraint` e a altura com `wrap_content`. Em seguida, aplique a margem para todos os lados com o valor `8dp` e o raio com o valor `10dp`.

O raio pode ser aplicado por meio do atributo `app:cardCornerRadius` do `CardView`

Após ajustar todos os parâmetros propostos pelo exercício, veja se o layout que implementou fica próximo ao que o cliente solicitou.

Lembrando que não precisa ficar exatamente como esperado, pois iremos melhorar o layout posteriormente.

