

Apresentando o dialog

Agora que implementamos o listener, vamos para o próximo passo que é implementar o dialog. Inicialmente, remova o toast que criamos no exercício anterior.

Construindo o Dialog

Então, utilize o [AlertDialog](https://developer.android.com/guide/topics/ui/dialogs.html#AlertDialog) (<https://developer.android.com/guide/topics/ui/dialogs.html#AlertDialog>) para a implementação do Dialog. Porém, como vimos, para construí-lo, precisamos utilizar a classe `Builder` dele.

Portanto, crie uma instância da classe `AlertDialog.Builder` e envie uma referência de `Context` via construtor.

Adicionando título no Dialog

Com o `Builder` adicione um título no dialog chamando a função `setTitle()`. Repare que essa função espera um `CharSequence` ou um `Int`, a princípio, podemos enviar uma `String` com o seguinte valor "Adicionar receita".

Entretanto, vamos utilizar o resource que veio da App base pra isso, portanto, envie o resource `R.string.adiciona_receita`.

Adicionando a view no dialog

Após adicionar o título, vamos adicionar o layout do dialog. Para isso, chame a função `setView()` logo após a chamada da `setTitle()`.

Criando o layout do dialog

Entretanto, precisamos criar a view para enviar na função `setView()`, portanto, crie uma view a partir do resource `R.layout.form_transacao`.

Note que neste processo é necessário o envio do parâmetro `ViewGroup` que é justamente o **parent** da view. Temos acesso a esse parâmetro por meio da property `window.decorView` que trata-se de uma `View`, ou seja, basta apenas fazer o cast com a keyword `as`.

Lembre-se de não anexar a view no momento que estiver criando.

Após criar o objeto para representar a view, envie-o como parâmetro da `setView()`.

Apresentando o Dialog

Por fim, chame a função `show()` após a chamada da `setView()` para que o dialog apareça, então, execute a App e veja se o dialog aparece.

