

07

## Configurando o campo de data

Por mais que tenhamos adicionado o layout no dialog, vimos que existem alguns detalhes que precisamos ajustar, como por exemplo, o campo de data e categoria.

Considerando esses detalhes, começaremos ajustando o campo de data.

### Adicionando o valor padrão na data

Como vimos na App base, neste campo temos como valor padrão a data atual. Portanto, vamos adicioná-la também.

Para isso, crie um objeto para representar a data atual chamando a função `getInstance()` da API `Calendar` de forma estática.

Então, com a view que foi criada para representar o layout do dialog (`viewCriada`), chame o componente `form_transacao_data` e atribua o valor da data atual chamando a função `setText()` e enviando a data via parâmetro.

Lembre-se de chamar a extension function `formataParaBrasileiro()`.

Após a inserção do valor padrão para representar a data do dia, vamos começar com a implementação do calendário.

### Adicionando o calendário no campo de data

Para isso, implemente o listener de clique do componente `form_transacao_data` por meio da função `setOnClickListener()`.

**Dica:** considere a implementação via expressão lambda para o listener de clique :)

Em seguida, dentro do escopo do listener, crie a instância da classe `DatePickerDialog`. Como vimos, essa classe possui um construtor que não é compatível com a nossa API mínima que é a **19**. Sendo assim, considere o seguinte construtor:

```
DatePickerDialog(Context,  
    DatePickerDialog.OnDateSetListener,  
    Int,  
    Int,  
    Int)
```

Como primeiro parâmetro, envie a referência `this` que é a própria Activity. Em seguida, precisamos implementar a interface responsável por receber a data que o usuário selecionou.

### Implementando a interface para pegar a data

Sendo assim, implemente a interface `DatePickerDialog.OnDateSetListener` utilizando a expressão lambda. Nessa implementação, a interface nos envia 4 parâmetros:

- `view` : a view do dialog

- ano : ano selecionado
- mes : mês selecionado
- dia : dia selecionado

Ou seja, crie um objeto do tipo `Calendar` por meio do `getInstance()` e atribua os três parâmetros por meio da função `set()` que espera respectivamente um **ano**, **mes** e **dia** via parâmetro.

Então, reatribua o valor do texto do componente `form_transacao_data` com a função `setText` enviando essa nova data que foi criada baseando-se na data que o usuário selecionou.

## Adicionando a data inicial do calendário

Então, envie como três últimos parâmetros do construtor da `DatePickerDialog` as variáveis `ano` , `mes` e `dia` respectivamente. Claro, declare e inicialize cada uma delas considerando a data de atual.

Lembrando que melhoraremos essa abordagem mais pra frente

Por fim, chame a função `show()` e teste a App, veja se aparece o valor padrão como a data atual e se quando toca no campo data o calendário aparece.