



## Ajustando funções menores

Continuando com o processo de refatoração, o nosso objetivo é melhorar alguns passos que foram realizados nas funções principais, como é o caso da `configuraCampoData()` e a `configuraFormulario()`.

Sendo assim, vamos começar com a função `configuraCampoData()`.

### Ajustando a data inicial

Repare que atualmente estamos utilizando um valor fixo para definir a data inicial quando o usuário toca no campo de data.

Em outras palavras, a partir do objeto `hoje`, chame a função `get()` para pegar os valores para cada uma das variáveis que representam a data, segue as constantes para cada um dos campos:

- `Calendar.YEAR` : para o ano.
- `Calendar.MONTH` : para o mês.
- `Calendar.DAY_OF_MONTH` : para o dia.

Depois desse ajuste, podemos partir para a função `configuraFormulario()`.

### Extraindo funções conversoras

Dentro dela, repare que existem 2 blocos de código responsáveis em fazer a conversão tanto do campo valor como também da data.

Sendo assim, extraia a função `converteCampoValor()` enviando todo o código que contém o **try catch expression** responsável em tentar converter o valor em texto para `BigDecimal`.

### Extraindo extension function

No código que converte a data, vimos que não temos uma certa dependência com a classe `AdicionaTransacaoDialog`, pois não fazemos uso, por exemplo, do `try catch`. Logo, podemos criar a extension function para reutilizar esse mesmo comportamento em outros pontos do projeto se for necessário.

Sendo assim, crie a extension function `String.converteParaCalendar()` que vai devolver a API `Calendar` e coloque todo o código que faz a conversão da data em texto.

Claro, após essa implementação, envie a extension function para o arquivo `StringExtension.kt`.