08

Para saber mais - Explorando mais recursos do LiveData

Além de usar recursos de leitura e escrita do `LiveData` com o Data Binding, somos capazes de usar mais features do `LiveData`, como é o caso da classe `Transformations` (https://developer.android.com/reference/android/arch/lifecycle/Transformations) que possui algumas funções que reagem com outros `LiveData`s, como por exemplo, a `map()` (https://developer.android.com/reference/android/arch/lifecycle/Transformations.html#map%28android.arch.lifecycle.LiveData%) que permite criar um novo `LiveData` que vai reagir conforme as mudanças de um `LiveData` existente.

Considerando um exemplo mais concreto, podemos criar uma property do tipo `LiveData<Boolean>` que vai ser criada a partir de um `map()`:

```
class NotaData(
    //outras properties,
    val imagemUrl: MutableLiveData<String> = MutableLiveData<String>().also { it.value = nota.imagem
) {

    val temUrl: LiveData<Boolean> =
        Transformations.map(imagemUrl) { url ->
            url.isNotEmpty()
        }

    //outras funções

}
```

O objetivo dessa amostra é fazer com que a property `temUrl` receba o valor no `onChange()` com base no retorno da expressão lambda (`url.isNotEmpty()`). Essa expressão lambda é apenas executada, a cada momento que o `onChange()` de `imagemUrl` é executado, portanto, a `temUrl` só ser modificada, apenas quando `imagemUrl` for observado.

Dada a integração feita entre Data Binding e LiveData, somos capazes de usar essa solução, portanto, podemos, por exemplo, alterar a `ImageView` para que utilize essa nova abordagem:

```
<ImageView
    android:id="@+id/formulario_nota_imagem"
    <!-- demais properties -->
    android:visibility="@{nota.temUrl ? View.VISIBLE : View.GONE}"
     />
```

Nesse exemplo, trocamos até mesmo as posições de `VISIBLE` e `GONE`, dado que a validação é feita para casos que temos a imagem.

É importante ressaltar também que, dada a peculiaridade da property `imagemUrl` ser observada para o `Transformations.map()` funcionar, ao integrar com o Data Binding, o resultado será apresentado apenas quando indicar um `LifecycleOwner` válido.