

## Criando classes utilitárias

Por mais que tenhamos feito uma refatoração grande, ainda existem pontos do código que podem ser melhorados, como por exemplo, nos métodos do Adapter:

- `mostraImagem()` ;
- `mostraDias()` ;
- `mostraPreco()` .

Note que dentro desses métodos o nosso código ainda tem uma certa complexidade, pois é necessário interpretá-lo para entender o que realmente ele faz.

Faremos então uma extração desses comportamentos para melhorar a leitura do código, como por exemplo, todo o código que formata a moeda pode ser extraído. Sendo assim, extraia todos os blocos de código que possuem um comportamento específico dentro de cada um dos métodos.

Após separar essa responsabilidade dos métodos que só apresentam as informações nas views, o adapter acaba tendo mais responsabilidade do que ele deve ter. Isso significa que é necessária mais uma etapa de refatoração.

### Delegando a responsabilidade

Portanto, crie uma classe utilitária para cada um dos métodos extraídos que possuem comportamentos que não devem ser do adapter, como por exemplo, no código que devolve um `Drawable` é possível delegá-lo para uma classe chamada de `ResourceUtil` .

**Dica:** No momento que for delegar a responsabilidade para uma classe utilitária, tente evitar acoplamento desnecessário, por exemplo, para pegar um `Drawable` não é necessário um `Pacote` e sim uma `String` , portanto, faz todo o sentido o método receber a `String` como parâmetro.

### Detalhes sobre classes utilitárias

É importante ressaltar que classes como essas são comuns por não precisar de uma instância das mesmas, ou seja, ao criar uma classe utilitária, mantenha os membros de maneira estática.

Também considere as mesmas técnicas de refatoração que vimos anteriormente, como extração de variáveis/constante e métodos. Claro, se surgir mais técnicas para deixar o código mais simples, também aplique, como é o caso do [early return](http://blog.alura.com.br/quanto-mais-simples-melhor/) (<http://blog.alura.com.br/quanto-mais-simples-melhor/>).

Por fim, execute a App e veja se tudo funciona como antes.

