

## Refatorando funções dos componentes

Continuando com a refatoração do Adapter, o nosso próximo passo é melhorar o código das funções que foram criadas na primeira parte da refatoração.

### Retornando variável a partir da estrutura de if

Sendo assim, vamos refatorar a função `adicionaValor()`, veja que atualmente sabemos que todo o código dentro dela tem o objetivo de adicionar o valor, porém, além de só adicionar, também é configurada uma cor.

Entretanto para entender que a cor está sendo modificada, precisamos interpretar uma estrutura de `if`. Em outras palavras, modifique o código para que a estrutura de `if` fique responsável em apenas devolver uma cor, isto é, uma variável e faça a configuração da cor fora do escopo do `if` recebendo a cor que o `if` devolver.

Nesta estada pode considerar o uso do **if expression** que permite o `if` devolver uma variável de acordo com o escopo verdadeiro ou `false (else)`

### Extraíndo função da estrutura de if

Em seguida, extraia o código do `if` para a função `corPor()` recebendo o enum `Tipo` como parâmetro para utilizar como referência durante a comparação do `if`.

### Análise de código

Dentro da função `corPor()` caso tenha utilizado o **if expression** para devolver o valor, considere o uso do *early return*, pois, dessa forma, temos o mesmo resultado com um código mais simples e compreensível por qualquer desenvolvedor.

### Realizando o mesmo comportamento em outro componente

Após o ajuste feito na função do componente `transacao_valor`, faça o mesmo processo na função `adicionaIcône()` que possui um código bem similar ao que tínhamos na função `adicionaValor()`, a única diferença é que nessa função será extraída a função `iconePor()` para retornar um ícone baseando-se no tipo. Por fim, execute a App e veja se tudo está funcionando como antes.