

Para saber mais: Flux validator

Como estudado ao longo do curso, “reinventar a roda” não é muito interessante no mundo da programação. Validação é um problema comum a todos os programadores. Logo, antes de implementar na mão, do zero, todos os tipos de validação, é interessante buscarmos o que já existe pronto e disponível.

Por isso, consultamos o [pub.dev \(https://pub.dev/\)](https://pub.dev/), repositório de extensões para Flutter e encontramos o [flux validator \(https://pub.dev/packages/flux_validator_dart\)](https://pub.dev/packages/flux_validator_dart), que tem diversas validações para os padrões brasileiros, como:

- Placa de veículos no padrão Mercosul
- CEP
- CNH
- CNPJ
- CPF
- Renavam
- Data no formato dd/m/yyyy
- Entre outros...

Mas, você pode estar se perguntando qual a diferença do [flux validator \(https://pub.dev/packages/flux_validator_dart\)](https://pub.dev/packages/flux_validator_dart) para o [brasil fields \(https://pub.dev/packages/brasil_fields\)](https://pub.dev/packages/brasil_fields).

O brasil fields trata os campos para que estejam em formato brasileiro de maneira estética, por exemplo, podemos citar as máscaras de CPF e CEP. Mas, ter o formato 000.000.000-00 não significa que o CPF é real e válido, de acordo com u...

cálculo complexo que existe para a geração de CPFs. Daí vem o flux validator, que testa se a lógica do valor digitado é coerente com o padrão desejado. O CPF, CNPJ, Renavam e afins contém todo um cálculo por trás, que verifica se os números informados são válidos ou não. Por isso, apenas garantir que a estética do campo esteja correta e que apenas aceite números, não é 100% garantido.

É válido lembrar que, para declarar as extensões externas, fazemos isso no arquivo `pubspec.yaml`. Também é possível declarar extensões que não são do [pub.dev \(https://pub.dev/\)](https://pub.dev/), através do GitHub da seguinte maneira:

```
nome_da_extensão:  
git:  
  url: git://github.com/proprietario/repositorio.git  
  ref: master
```

[COPIAR CÓDIGO](#)

A `url` que precisamos informar é a url do repositório no GitHub. O atributo `ref` é qual branch do repositório desejamos utilizar.

Atenção: independente da abordagem utilizada, é necessário importar a extensão no arquivo `.dart` que você utilizar no código.

Nos exemplos trabalhados, não precisamos utilizar a importação via GitHub, mas fica a dica! Essa opção pode ser bastante útil futuramente para você. ;)