



## Para saber mais: Conhecimentos valiosos

Trabalhar com hardware é uma tarefa um tanto delicada. Entenda hardware como: câmera, GPS, acelerômetro, giroscópio, sensor biométrico... Os celulares podem ou não ter algum desses elementos, assim como comportamentos completamente diferentes, dependendo do tipo de dispositivo.

No nosso caso, ao utilizarmos a câmera, poderíamos ter feito isso de duas formas distintas:

1. A primeira é codificar do zero a parte de comunicação com a câmera. A desvantagem de fazer isso é que precisamos criar testes de erros como: e se não tivermos permissão para executar a câmera? Ou se o usuário abriu e fechou a câmera sem tirar a foto mas o retorno esperado era uma foto e veio nulo?
2. Para ganhar tempo e não reinventar a roda optamos pela segunda opção, que consiste no uso da extensão [image\\_picker](https://pub.dev/packages/image_picker) ([https://pub.dev/packages/image\\_picker](https://pub.dev/packages/image_picker)), que traz a integração da câmera de uma maneira bastante simples. Basta criarmos uma instância do objeto `ImagePicker` e solicitar um `getImage` à ela, especificando que a fonte de obtenção da imagem será a câmera, e não a galeria.

Caso tenha curiosidade sobre como implementar da maneira “na mão”, codificando a comunicação com a câmera, sem utilizar extensões extras, sugiro a leitura deste [artigo sobre como tirar uma foto utilizando a câmera do dispositivo, disponibilizado no \*cookbook\* do Flutter](https://flutter.dev/docs/cookbook/plugins/picture-using-camera) (<https://flutter.dev/docs/cookbook/plugins/picture-using-camera>).

