

≡ 06

## Requisição no Servidor

Rita criou um serviço que busca as informações das fotos da AluraPic no servidor. Agora, ela quer garantir a qualidade da sua entrega, criando o seguinte teste:

```
it("O serviço PhotoService deve retornar uma foto com um id", fakeAsync(() => {
  const fakePhoto = {
    id: 1,
    description: 'photo1',
  };

  service.getPhoto("1").subscribe(response => {
    expect(response.body).toEqual(fakePhoto);
  });

  const request = httpMock.expectOne(req => {
    return req.method === "GET";
  });

  request.flush(fakePhoto);

  tick();
}));
```

Com base no código podemos afirmar que:

Selezione 3 alternativas

- A** Rita utilizou o `fakeAsync()`, pois a operação de uma requisição `http` é assíncrona.
- B** Ao chamar o método `'request.flush()'`, o código do projeto front-end da Rita chamou o servidor back-end, requisitando a informação.
- C** A avaliação do teste é realizada dentro de um método `subscribe()`, pois uma requisição `http` retorna um *Observable* no Angular.
- D** A função `tick()` faz com que o teste espere a quantidade de segundos informada no parâmetro.
- E** O método `httpMock.expectOne()` retorna um objeto válido se as condições de chamada da requisição estiverem de acordo com as condições informadas nos parâmetros.