

## Formulário de Categorias

Na criação do formulário de inclusão de categorias de fotos, não foi criado nenhum caso de teste. Então, Cecília puxou essa tarefa para melhorar a qualidade do AluraPic e criou o seguinte código:

```
it("deve incluir uma categoria", () => {
  const navigateSpy = spyOn(router, "navigate");
  spyOn(categoryService, "newCategory").and.returnValue(of(null));
  component.categoryForm.get("id").setValue("1");
  component.categoryForm.get("description").setValue("Paisagem");
  component.insertNewCategory();
  expect(navigateSpy).toHaveBeenCalledWith(["category"]);
});

it("deve realizar o log caso ocorra algum erro", () => {
  spyOn(categoryService, "signup").and.returnValue(
    throwError("Erro de Servidor")
  );
  component.categoryForm.get("id").setValue("1");
  component.categoryForm.get("description").setValue("Paisagem");
  component.insertNewCategory();
  const spyLog = spyOn(console, "log");
  component.insertNewCategory();
  expect(spyLog).toHaveBeenCalledWith("Erro de Servidor");
});
```

Com base no código podemos afirmar que:

Selecione 3 alternativas

- A** Como vimos no caso de teste “deve realizar o log caso ocorra algum erro”, podemos criar spy até de classes padrão do javascript como a classe `console`.
- B** O método `get` do componente de formulário permite selecionar um campo através do id configurado no html da aplicação.
- C** Com método `setValue()`, Cecília consegue simular a digitação do campo pelo usuário.
- D** No teste "deve incluir uma categoria" ela precisou criar dois *spies*: um para simular a chamada do serviço `categoryService`, e outra para conseguir avaliar o direcionamento da rota depois da ação `insertNewCategory`.
- E** A função `throwError()` gera uma `Exception`, possibilitando assim a simulação de uma situação de erro.

