

≡ 10

Stub, Dummy e Spy

Observe o código abaixo.

```
@patch("loja.logging.exception")
@patch("loja.enviar_email", side_effect=smtplib.SMTPConnectError(Mock(), Mock()))
def test_enviar_dados_do_pedido_registro_que_nao_foi_possivel_enviar_email(enviar_email):
    enviar_dados_do_pedido("usuario@email.com", "123456709")
    exception.assert_called_once_with("Não foi possível enviar mensagem para usuario@email.com")
```

O que podemos afirmar sobre ele?

Selezione 3 alternativas

A Podemos identificar três tipos de dublês neste caso de teste: `dummy` que são `Mock()` como parâmetros de `SMTPConnectError`, `exception` que é do tipo `Stub`, e `enviar_email` que é do tipo `Spy`.

B A função `enviar_dados_do_pedido` é a unidade sob teste e, segundo este caso de teste, espera-se que nela sejam chamadas as funções `logging.exception` e `enviar_email`.

C Os valores `'usuario@email.com'` e `123456709` poderiam ser representados por `Mock()` porque eles não participam da verificação de caso de teste.

D Se no lugar de `side_effect=smtplib.SMTPConnectError(Mock(), Mock())` estivesse `return_value=None`, o teste falharia.