

## Fluxo de exceções

Sabemos que é possível capturar uma exceção e relançar ela na pilha de chamadas através do comando `raise`. Qual o comportamento esperado do código abaixo?

```
def dividir(divisor):  
    try:  
        aux = 10 / divisor  
        return aux  
    except (Exception):  
        print("Exceção capturada e relançada!")  
        raise  
        return -1  
  
dividir(2)  
dividir(0)
```

Selecione 2 alternativas

- A** O `except (Exception)` captura todas as exceções, então se executarmos uma divisão por 0 no bloco `try`, a mensagem `Exceção capturada e relançada!` será exibida.
- B** O código `return -1` nunca será executado, pois lançar uma exceção com `raise` muda o fluxo de execução. Desta forma, pode ser removido do código.
- C** É necessário incluir um `return` antes da instrução `raise`, de modo que todos os caminhos deste código tenham retorno.
- D** O código `return aux` nunca será executado, pois a exceção muda o fluxo de execução. Desta forma, pode ser removido do código.