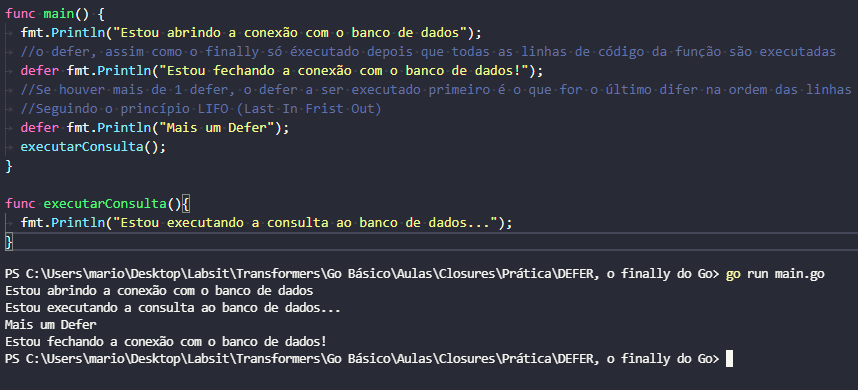
**DEFER, o “finally” do Go**

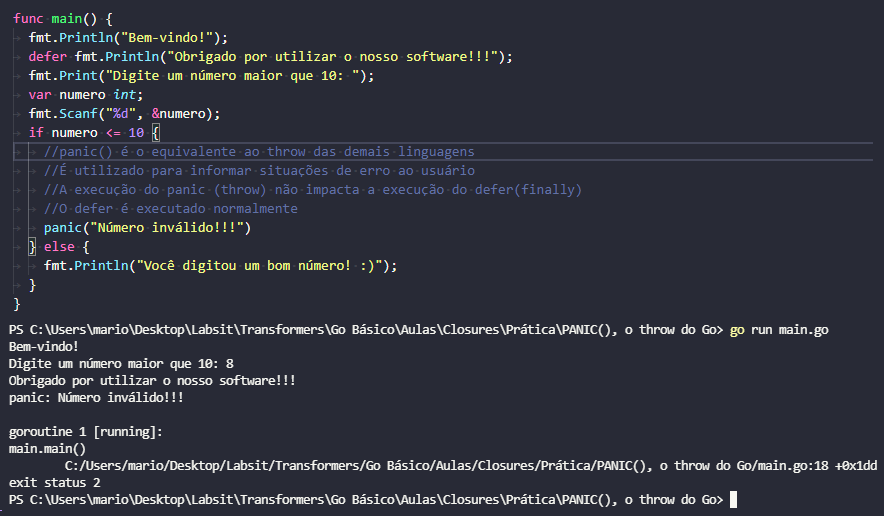
O defer, assim como o finally, só é executado após todas as linhas de código da função serem executadas.



Perceba, também, que caso haja mais de 1 defer na mesma função, o último defer será executado antes do anterior, seguindo o princípio LIFO (Last In First Out).

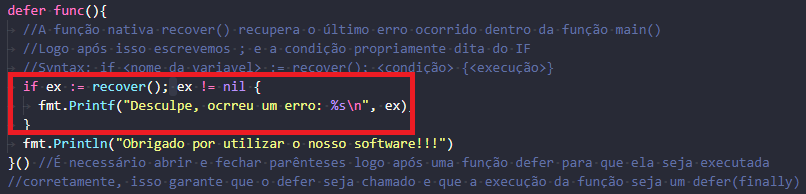
**PANIC, o “throw” do Go**

O panic() do Go é equivalente ao throw das demais linguagens. É utilizado para INFORMAR SITUAÇÕES DE ERRO ao usuário. A execução do panic (throw) não impacta na execução do defer (finally), portanto, os defers configurados serão executados normalmente es situações de erro.



**RECOVER(), o “catch” do Go**

A função nativa recover() recupera o último erro ocorrido dentro da função main(). Segue abaixo sua syntax:



Fazemos o encapsulamento da função recover() dentro de uma variável inicializada com inferência de tipo, logo após isso colocamos o ; e, em seguida, vem a condição propriamente dita da condicional IF. Também é necessário () logo após a função defer para que ela seja executada corretamente, isso garanto que o defer seja chamado e que a execução da função seja um defer (finally).