



Go - Tarea 2

Go, también conocido como Golang, es un lenguaje de programación de código abierto desarrollado por Google. Es conocido por su simplicidad, eficiencia y facilidad para la concurrencia. Go es ideal para construir aplicaciones de alto rendimiento y escalables, especialmente en entornos de servidor y sistemas distribuidos. Fue diseñado para ser concurrente, compilado y tipado estáticamente.

1. Enumere y explique las estructuras de control de flujo que ofrece.

Las estructuras de control que ofrece Go son:

- 1) **if**: Evalúa una condición y ejecuta un bloque de código si la condición es verdadera.

```
if condition {  
    ...  
}
```

- 2) **else if** y **else**: Permiten manejar múltiples condiciones. El bloque **else** se ejecuta si ninguna de las condiciones anteriores es verdadera.

```
if condition1 {  
    ...  
} else if condition2 {  
    ...  
} else {  
    ...  
}
```

- 3) **switch**: Selecciona entre múltiples bloques de código a ejecutar basándose en el valor de una expresión.

```
switch expression {  
case value1:  
    ...  
case value2:  
    ...  
default:  
    ...  
}
```

- 4) **for**: La única estructura de bucle en Go. Puede usarse de varias formas:

- Bucle tradicional con inicialización, condición y post-incremento:

```
for i := 0; i < 10; i++ {  
    ...  
}
```

- Bucle tipo "while":

```
for condition {  
    ...  
}
```

- Bucle infinito:

```
for {  
    ...  
}
```

- 5) **range**: Se utiliza con la estructura for para iterar sobre arrays, slices, mapas, cadenas y canales.

```
for index, value := range collection {  
    ...  
}
```

- 6) **goto**: Transfiere el control a una etiqueta definida en el código. Su uso es generalmente desaconsejado debido a que puede

```
label:  
    ...  
goto label
```

- 7) **defer**: Pospone la ejecución de una función hasta que la función que la contiene haya terminado. Las llamadas diferidas se ejecutan en el orden inverso en que se registraron.

```
func example() {  
    defer fmt.Println("Hola")  
    fmt.Println("Chao")  
}
```

En este ejemplo, "Chao" se imprimirá antes que "Hola".

2. Diga en qué orden evalúan expresiones y funciones.

- Se evalúan las expresiones de izquierda a derecha dentro de una expresión más grande y las funciones también se evalúan en este orden.
- Se utiliza evaluación aplicativa, lo que significa que los argumentos de una función se evalúan antes de que la función sea llamada.
- No tiene evaluación perezosa. Los argumentos de las funciones se evalúan inmediatamente cuando se llama a la función.
- Cabe destacar que los operadores lógicos `&&` (and) y `||` (or) realizan una evaluación de cortocircuito. Esto significa que si el resultado de la expresión ya está determinado después de evaluar el primer operando, el segundo operando no se evalúa.