

## Examen 2

### Pregunta 1

**a) Dé una breve descripción del lenguaje escogido.**

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

El lenguaje escogido fue Go, es un lenguaje creado por Google en el año 2007, con el propósito de ser eficiente como c++ pero simple de usar como los lenguajes de alto nivel como Python. Go es un lenguaje de programación compilado, además, es de tipado estático, entre sus características principales podemos decir que su sintaxis es simple, su compilación es rápida, posee ejecución eficiente y tiene excelente soporte para programación concurrente mediante "goroutines" y "channels".

Entre las estructuras de control de flujo que ofrece Go, podemos destacar:

**If / else if / else:**

Esta estructura se utiliza para la ejecución condicional, es decir, evalúa una condición de tipo booleana y ejecuta un bloque de código si la condición es verdadera.

Una característica a destacar es que no requiere paréntesis alrededor de la condición  
ejemplo: `if x > 0 { ... }`

También puede incluir una sentencia de inicialización al opcional antes de la condición, como lo puede ser `if err := foo(); err != nil { ... }`

**Switch:**

Switch es otra opción para evaluar las condiciones, esta estructura es bastante útil, sobre todo si tenemos muchas condiciones, ya que suele ser más legible la sintaxis. La forma de usarlo es mediante la palabra reservada "case", la cual es utilizada para evaluar una condición, podemos crear un "case" por cada una de ellas, en el caso que no se cumpla ninguna, se ejecutará el caso "default" el cual es el caso por defecto.

Switch tiene una característica llama type switch, la cual sirve para determinar el tipo dinámico de una variable de tipo interface {}. Se parece a un switch normal, pero en lugar de evaluar el valor de una variable, evalúa su tipo de dato.

**For:**

Go únicamente tiene una construcción de bucle y es for, tiene una estructura tradicional separadas por punto y coma, que se divide en "declaración", "condición" e "incremento".

```
for declaracion; condicion; incremento {  
    ...  
}
```

Ahora bien, estas características son opcionales y es por esto que aunque únicamente exista “for” como construcción de bucle, nos va a permitir simular otro tipo de formatos de ciclos que conocemos en otros lenguajes de programación, como por ejemplo el while y el do while.

El while se conforma únicamente por la “condición”, de esta forma se seguirá ejecutando mientras la condición sea verdadera.

```
for condición{  
    ...  
}
```

En el caso de do while sabemos que primero se debe ejecutar el código que está dentro del ciclo al menos una vez y luego se verifica la condición al final de este, por lo que para simular un do while en go se debe hacer uso de la palabra reservada break para poder salir del ciclo en caso de que cumpla dicha condición.

```
for {  
    ...  
    if condición{  
        break  
    }  
}
```

## **II) Diga en qué orden evalúan expresiones y funciones.**

### **A. ¿Tiene evaluación normal o aplicativa? ¿Tiene evaluación perezosa?**

Go tiene evaluación aplicativa o también conocida como evaluación estricta, esto significa que cuando se llama a una función, todos los argumentos se evalúan completamente antes de que la ejecución entre al cuerpo de la función.

Go no posee evaluación perezosa (lazy evaluation) de forma nativa, por lo que el valor de un argumento se calcula en el momento de la llamada, no cuando se usa por primera vez dentro de la función. Aunque se puede simular este comportamiento usando funciones anónimas o closures.

**B. La evaluación de argumentos/operandos se hace de izquierda a derecha, de derecha a izquierda o en un orden arbitrario.**

En Go la evaluación de los argumentos / operandos se hace estrictamente de izquierda a derecha, tanto para las llamadas a funciones como para las operaciones.

## **Bibliografía**

- Beecrowd, Lenguaje de Programación Go:

<https://beecrowd.com/es/blog-posts/lenguaje-de-programacion-go/#:~:text=Go%20es%20un%20lenguaje%20de,Quieres%20saber%20m%C3%A1s%20sobre%20Go?&text=El%20lenguaje%20de%20programaci%C3%B3n%20Go,utilizar%20en%20el%20desarrollo%20web>

- Go, Documentación:

[https://go.dev/doc/effective\\_go#:~:text=gramo\(\)%20%7D-,Estructuras%20de%20control,deben%20estar%20delimitados%20por%20laves](https://go.dev/doc/effective_go#:~:text=gramo()%20%7D-,Estructuras%20de%20control,deben%20estar%20delimitados%20por%20laves)

- Erick Suarez, Curso Go 02: Control De Flujo

[https://ericksuarez.dev/posts/control-de-flujo/#:~:text=Para%20representar%20este%20tipo%20de%20toma%20de,acciones%20%7D%20\\*%20for%20condici%C3%B3n%7B%20acciones%20%7D](https://ericksuarez.dev/posts/control-de-flujo/#:~:text=Para%20representar%20este%20tipo%20de%20toma%20de,acciones%20%7D%20*%20for%20condici%C3%B3n%7B%20acciones%20%7D)

Las siguientes preguntas se encuentran en el repositorio:

[https://github.com/CristyGomez/CI3641-CristinaGomez/tree/main/Examen2\\_CristinaGomez](https://github.com/CristyGomez/CI3641-CristinaGomez/tree/main/Examen2_CristinaGomez)

En cada pregunta añadí un README con las instrucciones para correr cada programa.