

## Instalación:

Ir al sitio web [go.dev](https://go.dev) y descargar el instalador de Golang

Abrir el archivo MSI que has descargado y seguir las instrucciones para instalar Go.

Se puede ejecutar el comando `$go version` para comprobar que Go está instalado en el sistema.

## Información general:

Golang es un lenguaje compilado, de sintaxis simple y moderna. Creado en 2009 por Google.

Usa un sistema de tipos estático y fuerte. Gestiona automáticamente la memoria (gracias a su garbage collector) y soporta una concurrencia eficiente con goroutines y canales.

Compatibilidad multiplataforma, permitiendo compilar a distintos sistemas operativos.

## Hola mundo en Go:

```
package main

import "fmt"

func main() {
    fmt.Println("Hello, World!")
}
```

Acciones que están sucediendo en este fragmento de código:

Declarar de un paquete main (un paquete es una forma de agrupar funciones, y está formado por todos los archivos de un mismo directorio).

Importar el paquete fmt, que contiene funciones para manipular texto, incluyendo la impresión a la consola. Este paquete es uno de los paquetes de la biblioteca estándar de Go.

Implementar una función main para imprimir un mensaje en la consola. Una función main se ejecutará por defecto cuando se ejecuta el paquete main.

## Tipos de variables:

Entre los tipos de variables más comunes y usados de Golang encontramos:

**bool:** Se declara con la palabra clave `bool` y solo puede tomar los valores `true` o `false`.

**string:** Una cadena es en realidad una envoltura de una secuencia de bytes.

**int(8– 64):** Se utilizan para almacenar un número entero sin decimales.

**uint(8– 64):** Es un tipo de dato que se usa para almacenar números enteros sin signo.

**uintptr:** Es un tipo entero sin signo que representa una dirección de memoria.

**float(32, 64):** Almacenar números positivos y negativos con un punto decimal.

**complex(64, 128):** Toma 2 literales de punto flotante, para la parte real y para la parte imaginaria.

```
var fruit string = "apple"
```

Los tipos de datos siempre van después del nombre de referencia de la variable.

## Ciclos:

En Go, no existen los ciclos while o do while, tenemos un for básico con tres componentes separados por punto y coma:

```
counter := 0
for counter < 10 {
    counter ++
}
```

La declaración init: Se ejecuta antes de la primera iteración.

La expresión de condición: Se evalúa antes de cada iteración.

La declaración posterior: Se ejecuta al final de cada iteración.

La declaración init a menudo será una breve declaración de variable, y las variables declaradas allí solo serán visibles en el alcance de la declaración del for.

```
HolaMundo := "Hola mundo"
for index, letra := range HolaMundo {
    fmt.Printf("%d %c \n", index, letra)
}
```

Tenemos también range, que permite recorrer una estructura iterable y acceder al respectivo índice y elemento. Ideal para recorrer arrays, slices, maps, channels, cualquier estructura que pueda recorrerse.

### Estructuras de control:

En go tenemos palabras reservadas como if, else, else if o switch, para alterar el comportamiento de nuestros paquetes, y ejecutar diversas acciones en nuestros programas, que funcionan igual que en otros lenguajes. (El operador := sirve para declarar y asignar una variable en una misma línea)

```
edad := 18
if edad < 18 {
    fmt.Println("Menor de edad")
} else if edad > 18 {
    fmt.Println("Mayor de edad")
} else {
    fmt.Println("Tiene 18 exactamente")
}
```

### Herramientas del ecosistema:

- go build: Compila el código fuente de Go y genera un ejecutable binario que puede ejecutarse directamente en tu sistema operativo.
- go fmt: Aplica un formato estándar al código Go para mantener un estilo consistente en el proyecto. Esto incluye la indentación, el espaciado y otras reglas.
- go test: Ejecuta las pruebas unitarias definidas en los archivos de prueba (\*\_test.go) de tu proyecto. Muestra los resultados de las pruebas, incluyendo errores, tiempos y métricas.
- go doc: Muestra documentación sobre funciones, tipos, paquetes o cualquier parte del código directamente en la línea de comandos.
- go mod: Administra las dependencias de tu proyecto usando un archivo llamado go.mod. Permite declarar las bibliotecas externas que necesita tu proyecto y asegura que siempre uses las versiones correctas.