Instalacion de ambiente de desarrollo

- Instalar Golang
- Instalar VS Code 2
- Probar en terminal que se ha instalado Go, usando comando:

```
go version
```

Tipos de datos

Enteros

```
uint8       unsigned  8-bit integers (0 to 255)
uint16      unsigned 16-bit integers (0 to 65535)
uint32      unsigned 32-bit integers (0 to 4294967295)
uint64      unsigned 64-bit integers (0 to 18446744073709551615)
int8        signed  8-bit integers (-128 to 127)
int16       signed 16-bit integers (-32768 to 32767)
int32       signed 32-bit integers (-2147483648 to 2147483647)
int64       signed 64-bit integers (-9223372036854775808 to 9223372036854775807)
```

Flotantes

```
float32         IEEE-754 32-bit floating-point numbers
float64         IEEE-754 64-bit floating-point numbers
complex64       complex numbers with float32 real and imaginary parts
complex128      complex numbers with float64 real and imaginary parts
```

Otros

```
bool      1 byte.
byte      alias for uint8
rune      alias for int32 (char)

uint      unsigned, either 32 or 64 bits
int       signed, either 32 or 64 bits
uintptr   unsigned integer large enough to store the uninterpreted bits of a pointer value
string
```

Go permite que las variables declaradas sean inicializadas de forma predeterminada.

- **Enteros (incluido rune)**: 0
- **flotantes**: 0.0
- **string**: cadena vacia
- **bool**: false

Declaracion de variables

```
var foo int // declaracion sin inicializacion
var foo int = 42 // declaracion con inicializacion
var foo, bar int = 42, 1302 // declaracion con inicializacion multiple
var foo = 42 // tipo omitido, se infiere del valor
foo := 42 // declaracion corta, solo en funciones
const constant = "This is a constant"

// iota puede ser usado para crear enumeraciones
const (
    _ = iota
    a
    b
    c = 1 << iota
    d
)
    fmt.Println(a, b) // 1 2 (0 es omitido)
    fmt.Println(c, d) // 8 16 (2^3, 2^4)
```