



GO

BOOTCAMP

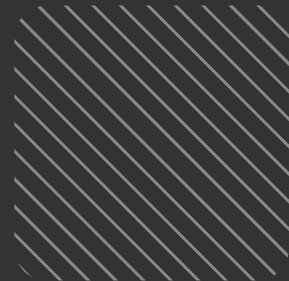


Clase en vivo

//Go Bases

IT BOARDING

BOOTCAMP





Objetivos de esta clase

- ◆ Obtener un primer acercamiento a programación con Golang
- ◆ Conocer los primeros pasos para diseñar y ejecutar un Hola Mundo.
- ◆ Repasar el uso de packages en Golang.
- ◆ Repasar el uso de modules con Golang

Índice



01 [Repaso](#)

02 [Primer programa en GO](#)

03 [Live Coding: Variables](#)

04 [Práctica](#)

IT BOARDING

BOOTCAMP



1

Repaso

IT BOARDING

BOOTCAMP



Package

Recordemos: cada archivo de Go pertenece a un package. Entonces, para declarar unos de estos archivos usamos la siguiente sintaxis:

```
{ } package packagename
```

- La carpeta que contiene los distintos archivos del package y el package se deben llamar igual a este último.
- La declaración del package anterior debe ser la primera línea de código en su archivo fuente de Go.
- Todas las funciones, tipos y variables definidas en su archivo fuente de Go pasan a formar parte del paquete declarado.
- Los nombres de los packages deberían ser de una sola palabra y todos en minúscula.



¿Cómo inicializar un module?

Antes de iniciar un módulo Go, tenemos que asegurarnos de crear un nuevo repositorio en Github u otros controles de versión (por ejemplo, Gitlab). Luego, podremos clonar el repositorio. Para iniciar un módulo Go usamos este comando:

Dominio

Nombre del module

```
{ } go mod init github.com/userGithub/go-simple-module
```

El dominio y el nombre del módulo deben coincidir con el nombre del repositorio que ya se creó.

Después de que se haya iniciado el módulo se crea el archivo `.go.mod`, este contiene las dependencias que se utilizan en la aplicación.





2

Primer programa en Go

IT BOARDING

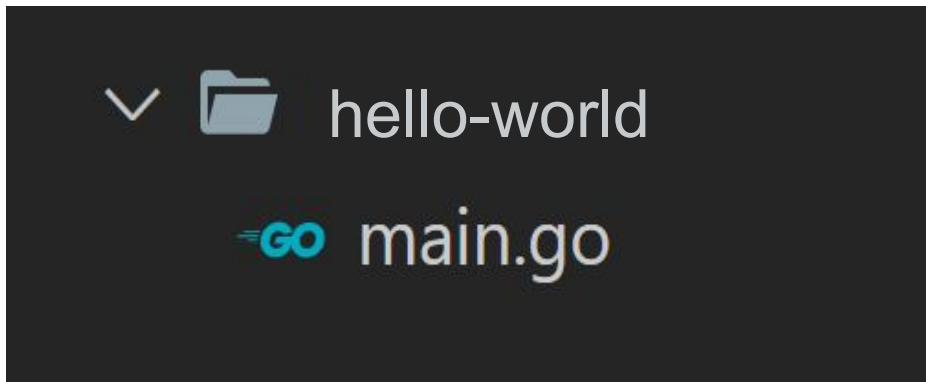
BOOTCAMP



Mi primer Hola Mundo!

Antes de comenzar con nuestro Hola Mundo, vamos a **crear una carpeta** donde estará nuestro archivo de ejecución.

Vamos a llamar a esa carpeta **hello-world** y dentro de ella creamos un archivo **main.go**.



Mi primer Hola Mundo!

Todas nuestras aplicaciones en Go corren desde un package principal, dentro de nuestro archivo en la primera línea debemos declarar nuestro package main.

```
{ } package main
```

Go nos proporciona varios packages standard que podremos utilizar, como fmt, una de las funcionalidades que nos permite imprimir mensajes por consola.

```
{ } import "fmt"
```



Mi primer Hola Mundo!

Debemos definir una función `main()` donde correrá nuestro programa, todo lo que esté dentro de nuestra función es lo que se ejecutará. En este caso utilizamos una de las funciones que nos proporciona el package que importamos para imprimir un mensaje por consola y lo hacemos de la siguiente manera:

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



Mi primer Hola Mundo!

El código debería verse de la siguiente manera:

```
{  
package main  
  
import "fmt"  
  
func main() {  
    fmt.Println("Hello world!")  
}
```



Mi primer Hola Mundo!

Para correr y validar que todo lo que hayamos hecho funcione bien, desde nuestra terminal ejecutamos el comando **go run** (nos permite correr nuestros comandos) dentro de la carpeta hola-mundo especificando el archivo.

```
{ } $ go run main.go
```



Actividad



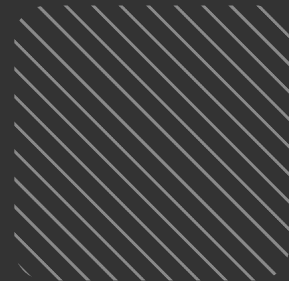


3

Live Coding: Variables

IT BOARDING

BOOTCAMP



A codear...





4

Práctica

IT BOARDING

BOOTCAMP



Actividad





Conclusiones

Aprendimos cómo organizar nuestro código mediante Packages y Modules.

También realizamos prácticas para asentar los conocimientos sobre cómo crear variables y sus distintos tipos.





Gracias.

IT BOARDING

BOOTCAMP

