



**Go 101: Primeros Pasos**

**@oscar\_lopez**

# Presentación: Go

- Programación imperativa, estructurada y libre
- También: programación funcional básica
- Compilado, estática y fuertemente tipado
- Seguro: tipos y memoria
- Programación del sistema -> propósito general
- Desarrollado por Google, apareció en 2009, diseñado por Robert Griesemer, Rob Pike y Ken Thompson
- Influencias: C, Limbo, Modula, Newsqueak, Oberon, Pascal, Python
- Recolector de basura eficiente, libre de latencia
- Primitivas de concurrencia modernas
- Compilación muy rápida, interacción con C
- Soportado en Google App Engine

# Estructura Básica

- Hola, Mundo!
- Paquetes
- Imports
- Tipos de datos
- Funciones I
- Constantes
- Variables

# Instrucciones de Control

- If
- Switch
- For
- Range

# Estructuras de Datos

- Structs
- Apuntadores
- Arreglos
- Slices
- Mapas
- Listas doblemente encadenadas, heap
- Funciones II

# Estructura Avanzada

- Métodos
- Interfaces
- Manejo de errores

`defer()` , `panic()` , `recover()`

# Concurrencia

- Goroutines
- Channels
- Close, Range
- Select

# Otros Paquetes y Funciones

- Paquetes
- Servidor web
- Imágenes
- Funciones predefinidas:

`close()`, `len()`, `cap()`, `make()`,  
`new()`, `append()`, `copy()`, `delete()`,  
`complex()`, `real()`, `imag()`,  
`panic()`, `recover()`



# Bibliografía

- Documentación <http://golang.org/doc/>
- Instalación y ejecución de programas en Go  
<http://golang.org/cmd/go/>
- App Engine  
<https://developers.google.com/appengine/docs/go/gettingstarted/>
- Mejores prácticas  
[http://golang.org/doc/effective\\_go.html](http://golang.org/doc/effective_go.html)
- Especificación del lenguaje  
<http://golang.org/ref/spec>

# Bibliografía

- Tutorial programación funcional  
<http://golang.org/doc/codewalk/functions/>
- Tutorial aplicaciones web  
<http://golang.org/doc/articles/wiki/>
- Tutorial concurrencia  
<http://golang.org/doc/codewalk/sharemem/>
- Manejo de errores  
[http://golang.org/doc/articles/defer\\_panic\\_recover.html](http://golang.org/doc/articles/defer_panic_recover.html)
- Uso de *slices*  
[http://golang.org/doc/articles/slices\\_usage\\_and\\_internals.html](http://golang.org/doc/articles/slices_usage_and_internals.html)

# Reto

La posada “La Vecindad del Chavo” lleva más de diez años brindando servicio de alojamiento a un conjunto de personas. Con el fin de organizar sus finanzas, quiere implementar un Sistema de Información que le permita administrar los dineros aportados por sus inquilinos por concepto de Arriendo.

Para el cobro del arriendo se debe tener en cuenta.

El pago del arriendo es de \$450.000 mensuales, sin embargo tendrá un incremento del 10% si no es pagado en los 5 primeros días del mes.

El sistema deberá entonces:

1. Permitir el ingreso de los datos básicos de los arrendatarios en cada lugar: Nombres, documento, lugar de residencia.
2. Permitir el ingreso de los pagos mensuales de cada inquilino

Enunciado:

<https://docs.google.com/document/d/1c6EEDyryYFa4Uydv-T00VMhXYobTmiQplttxuhPtMtw/edit?pli=1>