



Curso
**Básico de
Programación
en Go**

Osmandi Gómez
@osmandi

Bienvenida

- Ing. Químico.
- Data Engineer en Platzi.
- Gopher #data-science
- Higor (Dataframe for Golang)

 @osmandi



¿Qué aprenderás?

- Variables y constantes
- Funciones y estructuras básicas
- Goroutines
- Go modules
- Librerías para Backend
- Tips

Introducción



Curso de Introducción a la Terminal
y Línea de Comandos



Curso Profesional de Git y GitHub

¿Qué es y por qué
usar Go?





¿Qué es Go?

- Compilado y estáticamente tipado.
- Creado en Google por Robert Griesemer, Rob Pike y Ken Thompson.
- Anunciado en Noviembre 2009.
- Primera versión en Marzo 2012.
- Go/Golang.
- Programadores: Gophers.

Go Gopher

- Renee French
- WFMU radio station
en New Jersey
- [Blog.golang.org
/gopher](http://Blog.golang.org/gopher)
- gopherize.me



Go ícono (2018)



GO



GO



GO



GO



GO



GO

¿Por qué Go?

- Gran velocidad de compilación
- Alto rendimiento para tareas pesadas
- Soporte nativo para concurrencia
- Top 5 más amados y mejores pagados en encuesta Stackoverflow 2020 (\$ 74 k)
- Obliga a implementar buenas prácticas
- Comunidad muy receptiva

¿Quiénes usan Go?

- MercadoLibre: 70.000 con 20 MB RAM
- Twitch: Usuarios concurrentemente
- Twitter: Procesar analítica de la App con Kafka
- Uber: Posición conductores y pasajeros
- Docker y Kubernetes: Para despliegue de Apps



Instalar Go en Linux

Nuestra primera línea en Go

“Hola mundo” en Go

```
package main
```

```
import "fmt"
```

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



Variables, funciones y documentación

Operadores aritméticos



Valores primitivos

Números enteros

- int = Depende del OS (32 o 64 bits)
- int8 = 8 bits = -128 a 127
- int16 = 16 bits = -2^{15} a $2^{15}-1$
- int32 = 32 bits = -2^{31} a $2^{31}-1$
- int64 = 64 bits = -2^{63} a $2^{63}-1$

Números enteros positivos

- uint = Depende del OS (32 o 64 bits)
- uint8 = 8 bits = 0 a 2^8-1
- uint16 = 16 bits = 0 a $2^{16}-1$
- uint32 = 32 bits = 0 a $2^{32}-1$
- uint64 = 64 bits = 0 a $2^{64}-1$

Números decimales

- float32 = 32 bits = +/- $1.18e^{-38}$ a +/- $3.4e^{38}$
- float64 = 64 bits = +/- $2.23e^{-308}$ a +/- $1.8e^{308}$



Textos y booleanos

- `string = ""`
- `bool = true o false`

Números complejos

- Complex64 = Real e imaginario float32
- Complex128 = Real e imaginario float64
- Ejemplo: $c := 10 + 8i$

Paquete “fmt”

Algo más que imprimir en consola



Uso de funciones



go doc

La forma de ver documentación

El poder del ciclo for en Golang



El condicional if

Múltiples condiciones anidadas con Switch

El uso de los keywords defer, break y continue



Arrays y Slices

Recorrido de Slices con Range



Solución al Reto

Hacer un palíndromo en Go



Llave valor con Maps



Structs

La forma de hacer clases en Go

Modificadores de acceso

en funciones y Structs



Structs y Punteros

Stringers

Personalizar el output de Structs



Interfaces y listas de interfaces

¿Qué es la
conurrencia?

“

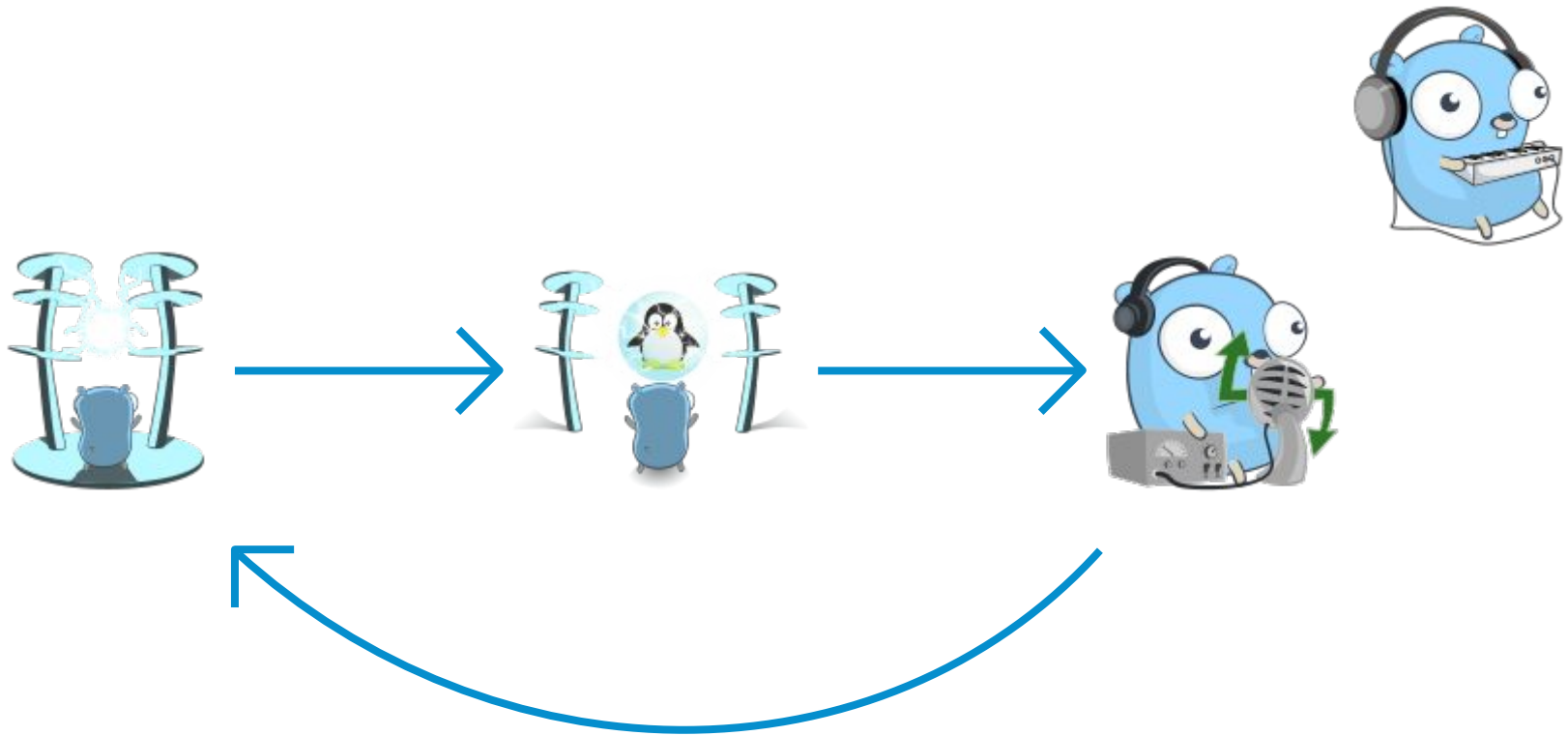
**Concurrency is dealing
multiple things at a single time
while parallelism is doing
multiple things at single time.**

”

Rob Pike

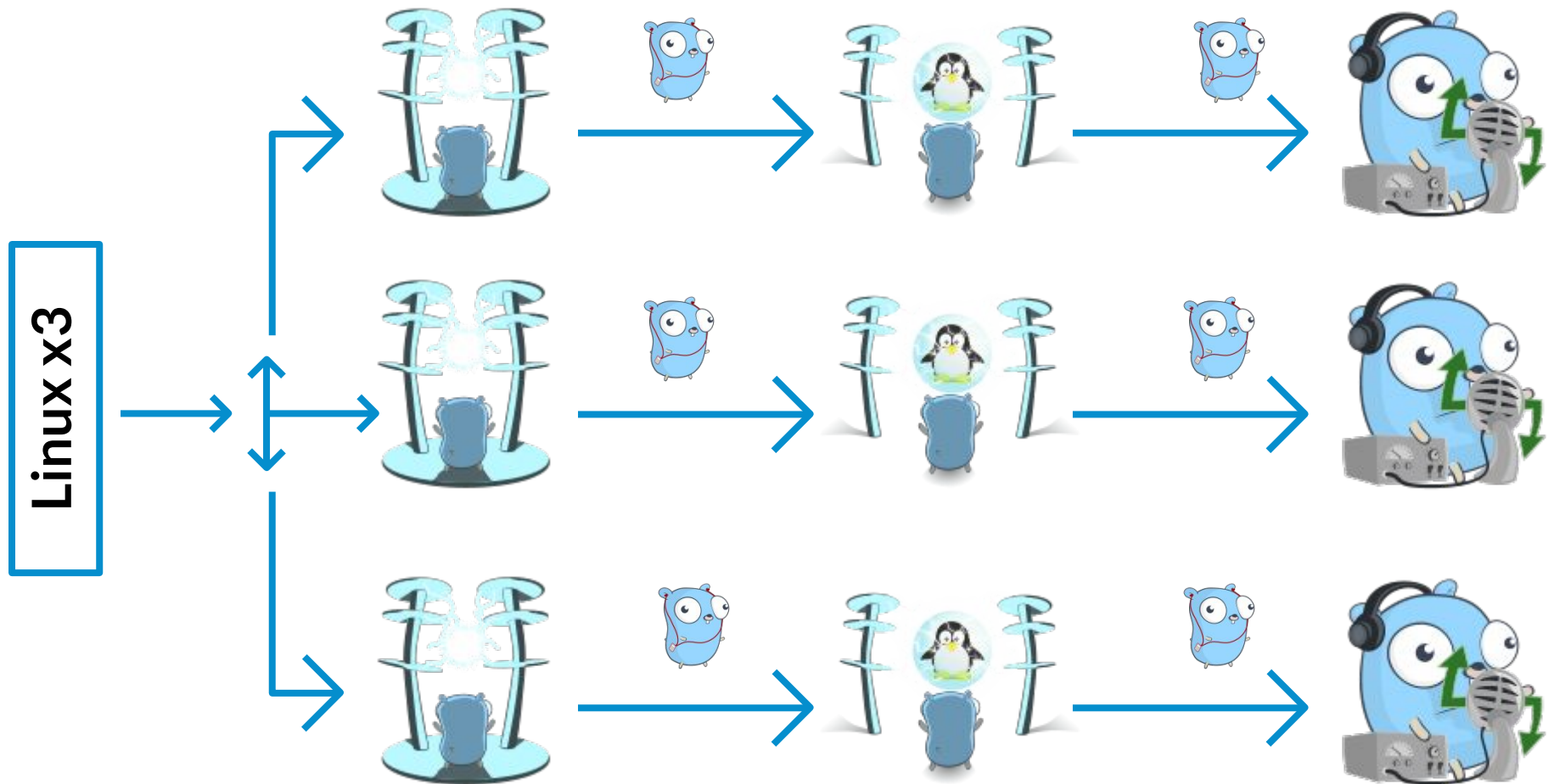
¿Qué es la concurrencia?

Proceso en un hilo



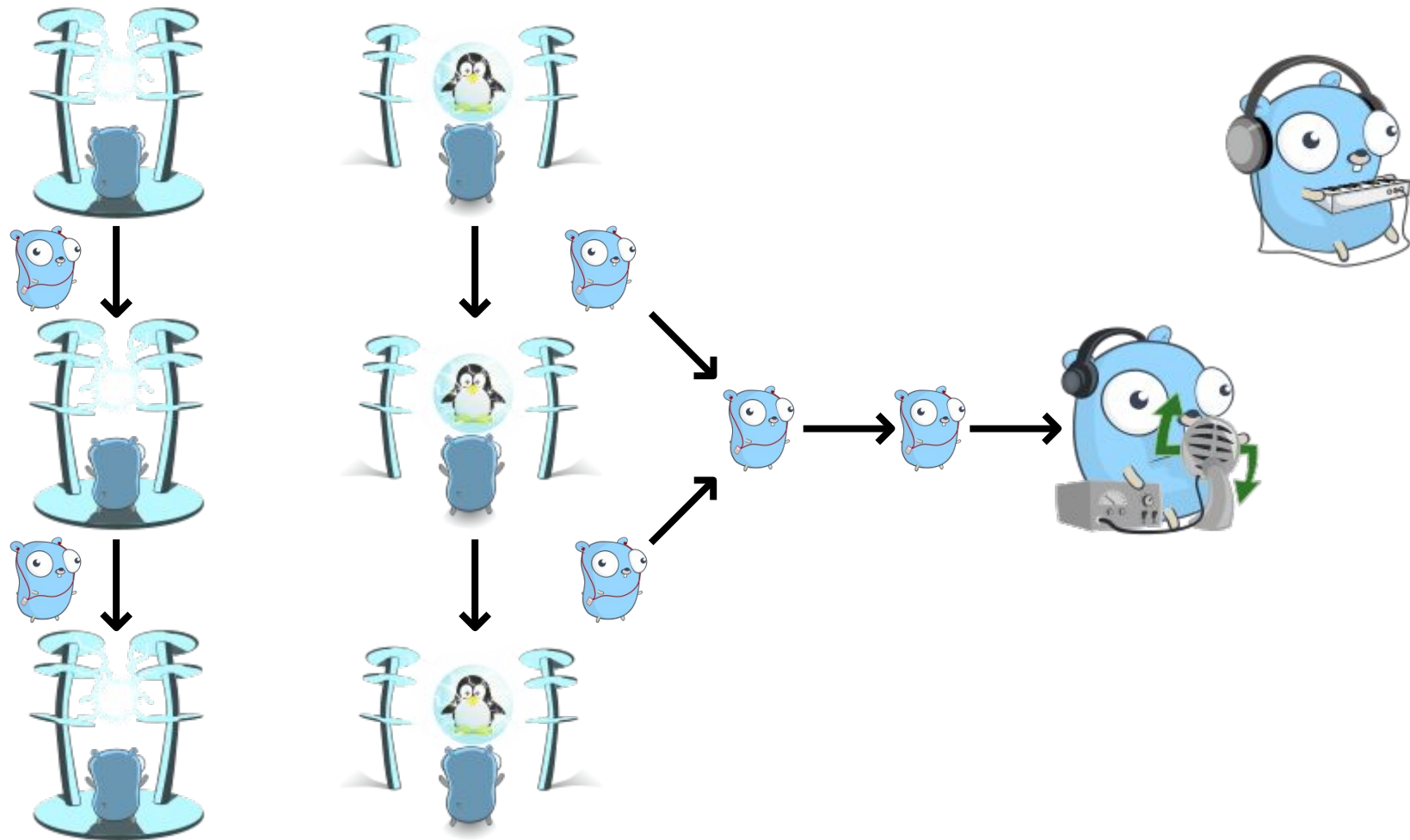
¿Qué es la concurrencia?

Paralelismo



¿Qué es la concurrencia?

Concurrencia



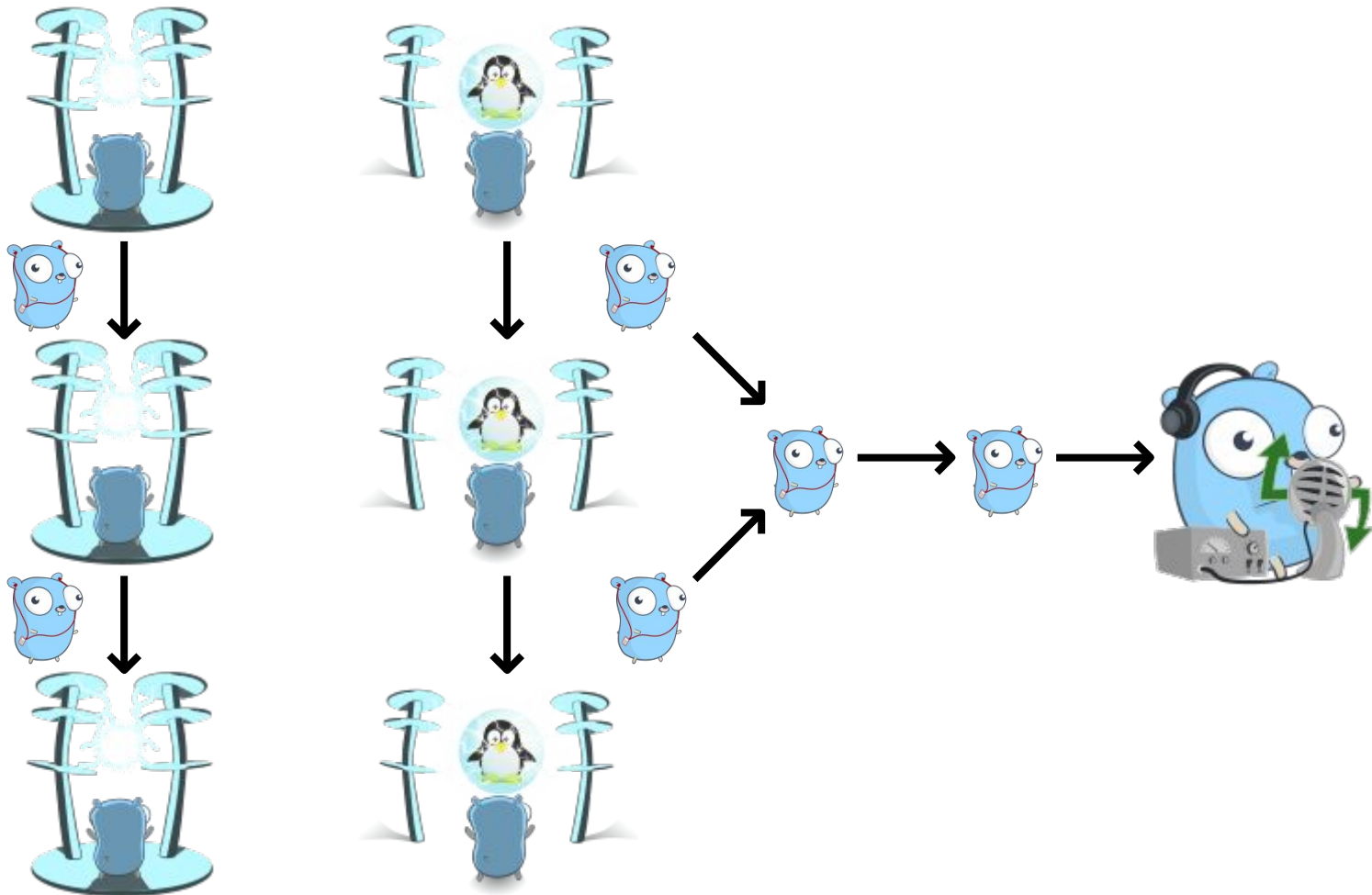
Primer contacto con las Goroutines



Channels

La forma de organizar las goroutines

Channels



Range, Close y Select en channels

go get

El manejador de paquetes



go modules

Ir más allá del GoPath



Modificando módulos con Go

Hugo

Generador de archivos estáticos

Librerías para BackEnd con Go

Librerías para BackEnd con Go

- Gin-Gonic: gin-gonic.com
- Beego: beego.me
- Echo: echo.labstack.com
- Revel: revel.github.io
- Bufalo: gobuffalo.io
- Gorilla: gorillatoolkit.org

Despedida del curso



¿Con qué seguir?

- Gophers Slack: gophers.slack.com
- Go tour: tour.golang.org
- Golang Weekly: golangweekly.com
- Play With Go: play-with-go.dev
- Go by example: gobyexample.com
- Go Time (en Spotify)



Data Science con Go



Data Science con Go

- Jupyter Kernel
Gophernotes
- Dataframes
qframe, Gota, dataframe-go
- Visualizaciones
gonum/plot, go-echarts
- Machine Learning
GoLearn, Gorgonia (Deep Learning)