

# Workshop: Arquitecturas de Microservicios



UCSE-Rafaela



## Segundo Encuentro

- Frameworks de desarrollo para distintos lenguajes
- Contexto como medida de manejo performante de memoria
- Microservicios y el cloud
- Estrategias de despliegue de microservicios
- Patrones internos de desarrollo, manejo de paralelismo.

Hands on... GoLang. Estructura MVC, Gin-Gonic.



## Framewor~~k~~s de desarrollo para distintos lenguajes.

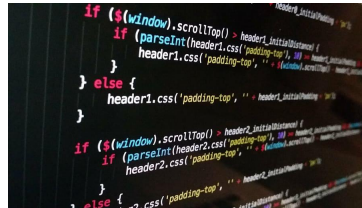


Herramientas  
genéricas de  
desarrollo

Libs



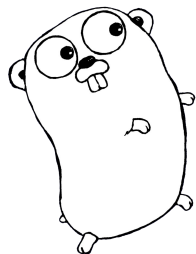
+



Herramientas  
reutilizables  
con propósito



## Frameworks de propósito específico de contenido MVC.



Golang

GinGonic  
Gorilla



Laravel  
Symfony  
CodeIgniter  
Project  
Yii (Framework)  
CakePHP  
Slim  
Phalcon  
FuelPHP



Grails  
Spring MVC



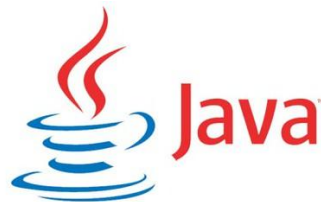
## Frameworks de propósito específico de contenido MVC.



Python



Django  
Flask  
FastApi



Spring Boot  
Micronaut  
Jooby  
Spark  
Struts  
Play  
Vaadin

Framework



Propósito



## Contexto como medida de manejo performante de memoria

Singleton



Instancia de un objeto  
autoreferenciada

Heredable, Extendible, Multi  
instancia, no vive en el  
PermSpace, testeable,  
inyectable

Contexto



Ej: connection pools, rest clients. (recursos limitados  
reutilizables)



# Contexto como medida de manejo performante de memoria

- Cuidado con la modificación de las propiedades si las hubiera.
- Es ideal que sean objetos puros de comportamiento.
- Explotan la capacidad de los lenguajes orientados a objetos, pero no es recomendable en otro tipo de lenguajes.
- Levantan en bootstrap, tienden a demorar el inicio de la app
- Hay frameworks que los traen por defecto, hay que saber manipularlos (Grails, Spring)

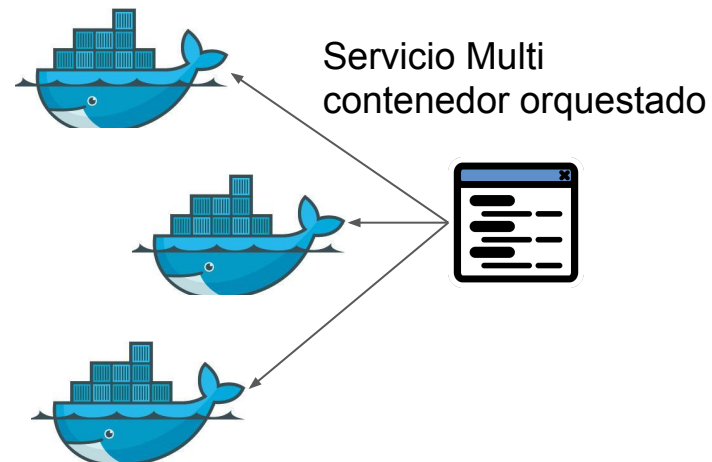
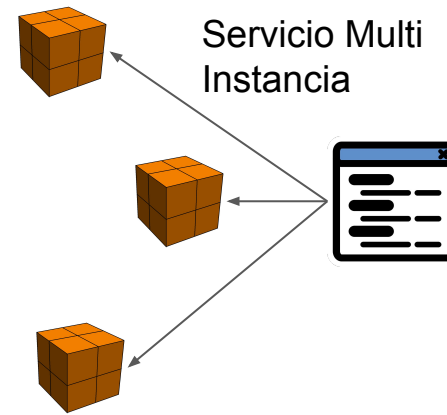


## Microservicios y el cloud

Estabilidad con redundancia, farming de servidores

Stateless

Carga distribuida (load balancer, auto scaling group) con infraestructura bajo demanda





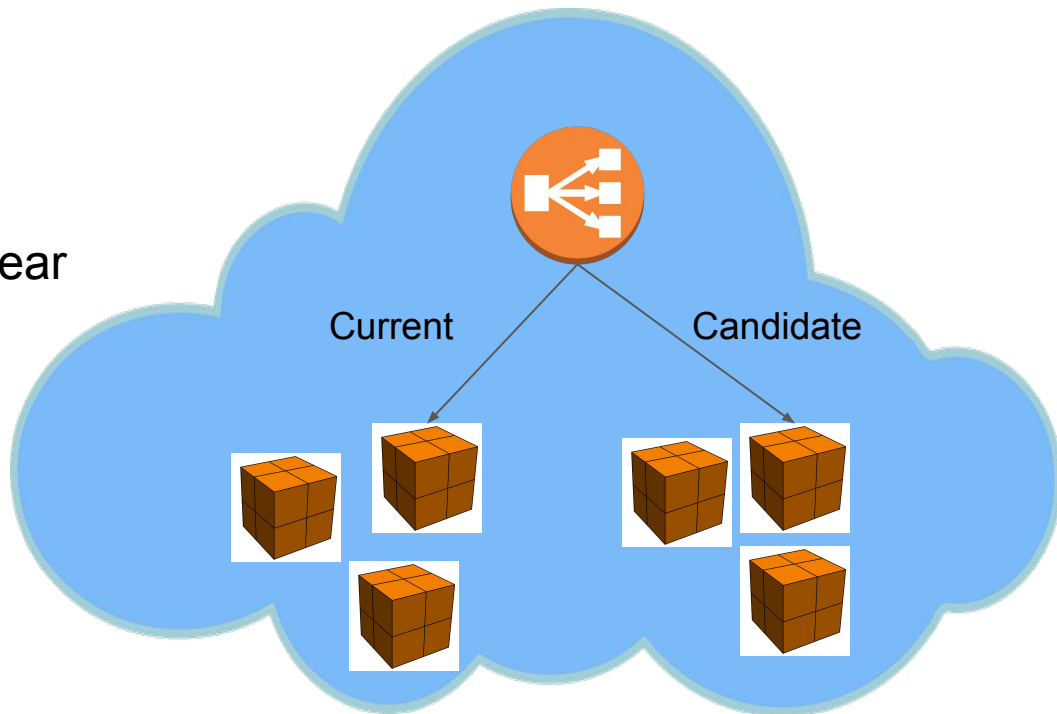


## Estrategias de despliegue de microservicios

- Blue green

### Deploy en caliente

ELB o ALB con capacidad de switchear el tráfico sin perder conexiones.

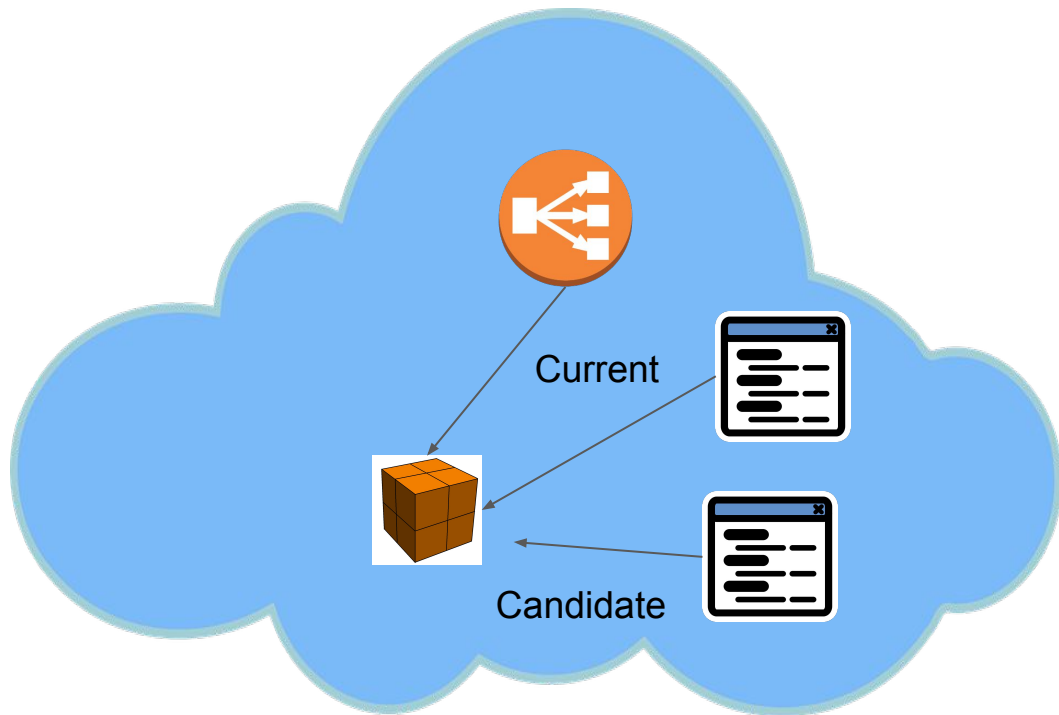




## Estrategias de despliegue de microservicios

- All in

Deploy con downtime



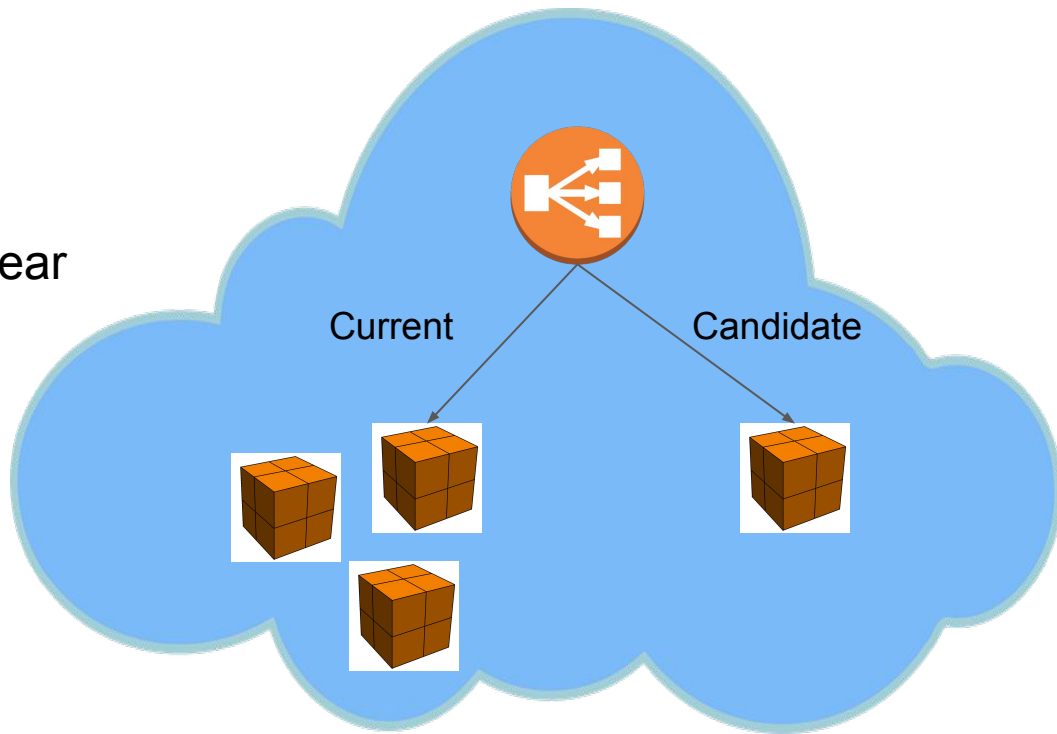


## Estrategias de despliegue de microservicios

- Canary

### Deploy en caliente

ELB o ALB con capacidad de switchear el tráfico sin perder conexiones.





# Patrones internos de desarrollo.

- Circuit Breaker
- Object Pools (restclient pool, connection pool)
- Rate Limiters
- Local Caches



## Manejo de paralelismo

- Threads
- Semaforos, estrategias de locking
- Canales
- Componentes thread safe





# Hands on GoLang.

Para la próxima clase descarguen e instalen MongoDB:

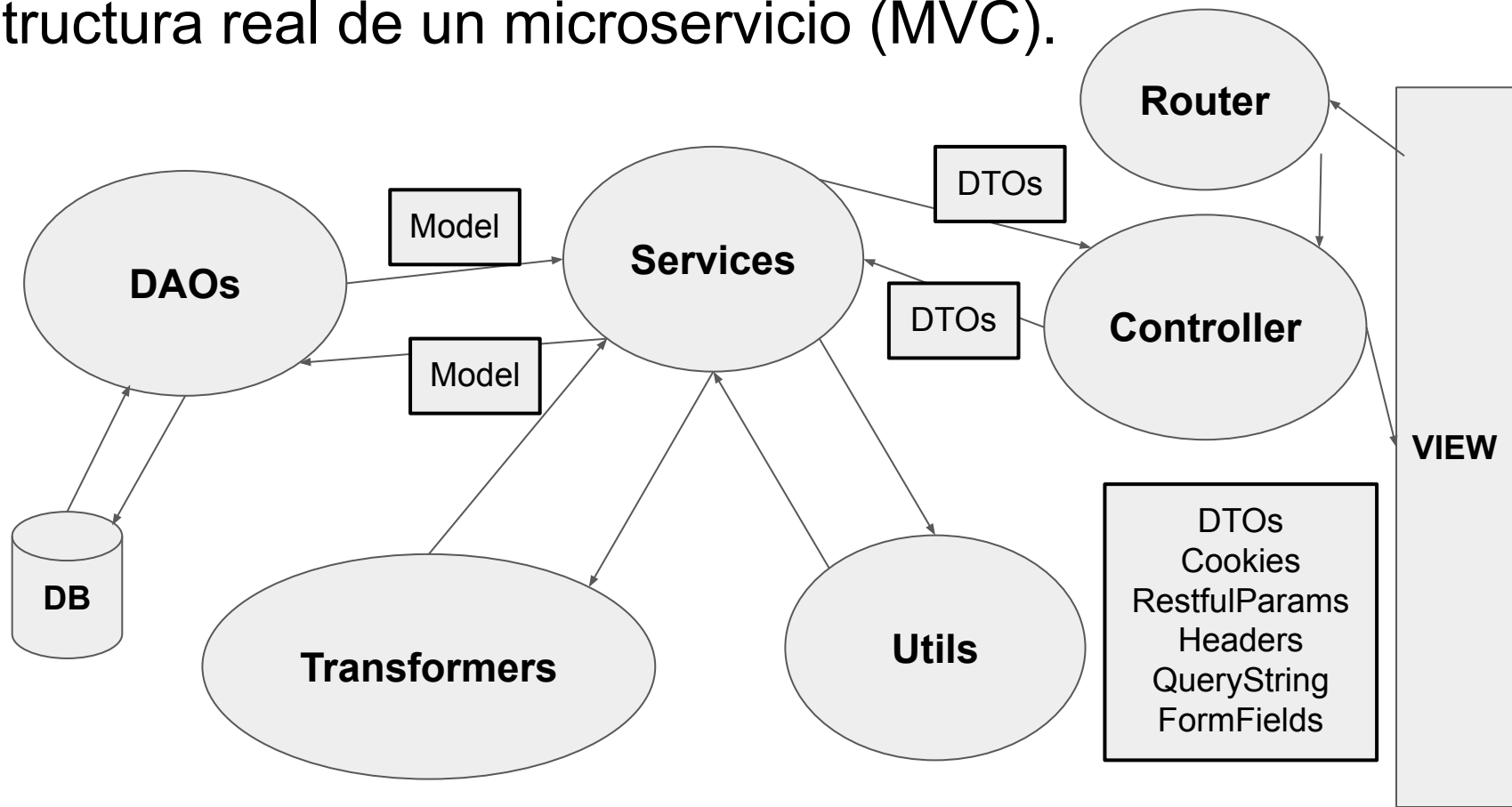
<https://www.mongodb.com/try/download/community>

Github:

[https://github.com/francoyadon/arq\\_microservicios](https://github.com/francoyadon/arq_microservicios)



## Estructura real de un microservicio (MVC).





Gracias

Que está mal? (que falta)

