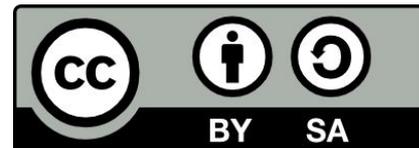


# From 0 to Cloud. Despliegue de aplicación Cloud Friendly en Azure.

# Atribución Compartirlgual 4.0 Internacional (CC BY-SA 4.0)



Usted es libre para:

**Compartir** – copiar y redistribuir el material en cualquier medio o formato.

**Adaptar** – remezclar, transformar y crear a partir del material para cualquier propósito, incluso comercialmente.

El licenciante no puede revocar estas libertades en tanto usted siga los términos de la licencia.

Bajo los siguientes términos:



**Atribución Paradigma Digital** – Usted debe darle crédito a esta obra de manera adecuada, proporcionando un enlace a la licencia, e indicando si se han realizado cambios. Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que usted o su uso tienen el apoyo del licenciante.



**Compartirlgual** – Si remezcla, transforma o crea a partir del material, debe distribuir su contribución bajo la misma licencia del original.

**No hay restricciones adicionales** – Usted no puede aplicar términos legales ni medidas tecnológicas que restrinjan legalmente a otros hacer cualquier uso permitido por la licencia.

<http://creativecommons.org/licenses/by-sa/4.0/deed.es>



¡Hola!



Rubén Villar  
Cloud Architect

[rvillar@paradigmadigital.com](mailto:rvillar@paradigmadigital.com)

<https://www.linkedin.com/in/rubenvillar/>



Javier Serrano  
Architecture Technical Lead

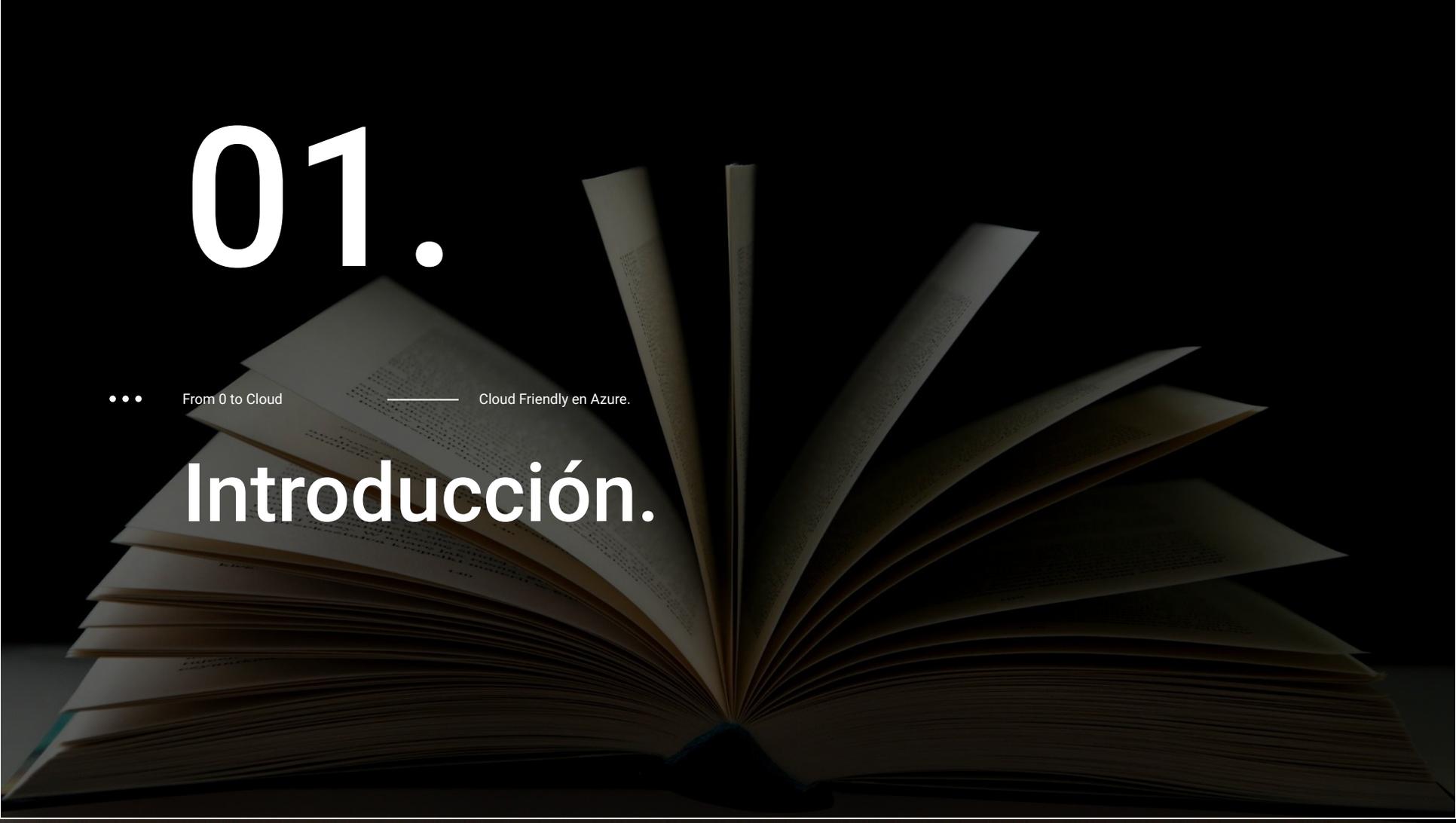
[jserrano@paradigmadigital.com](mailto:jserrano@paradigmadigital.com)

<https://www.linkedin.com/in/javiserrano/>

# Qué vamos a ver.

1. Introducción.
2. Aterrizando en Azure
3. Re-Cap
4. What's next?

# 01.



From 0 to Cloud

Cloud Friendly en Azure.

## Introducción.

# Road to cloud.

## Cloud Ready

- Sin acceso a disco local.
- Autocontenida.
- Gestión de red y puertos delegada.
- Uso de servicios gestionados.

## Cloud Friendly

- 12 Factor App.
- Escalado horizontal.
- Alta disponibilidad delegada en plataforma.

## Cloud Resilient

- Diseño tolerante a fallo y resiliente.
- Implementación agnóstica.
- Monitorización y métricas.
- Testeo de fallos proactivo.

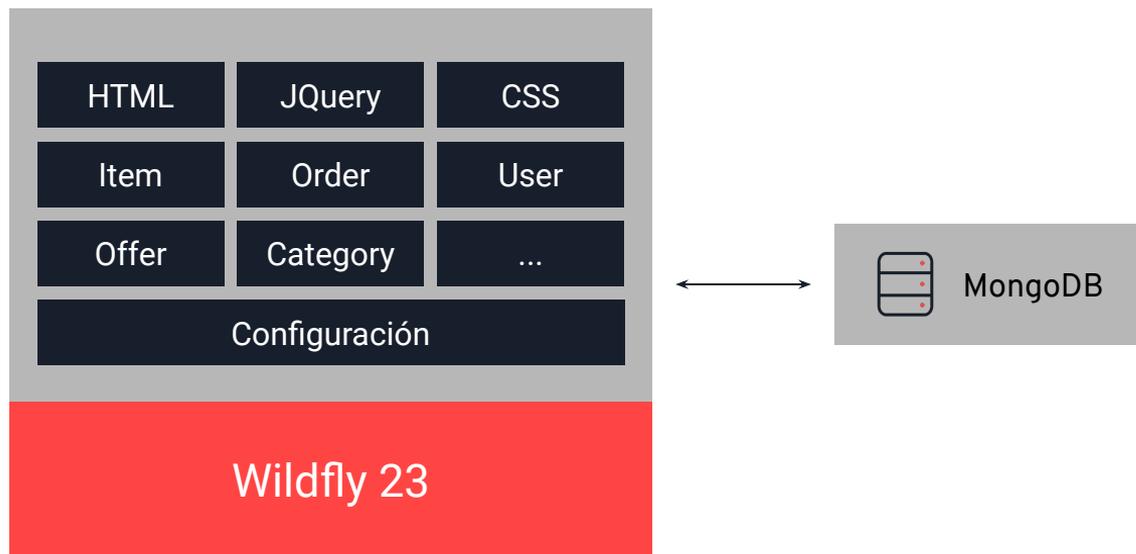
## Cloud Native

- API First.
- Arquitectura basada en Microservicios.

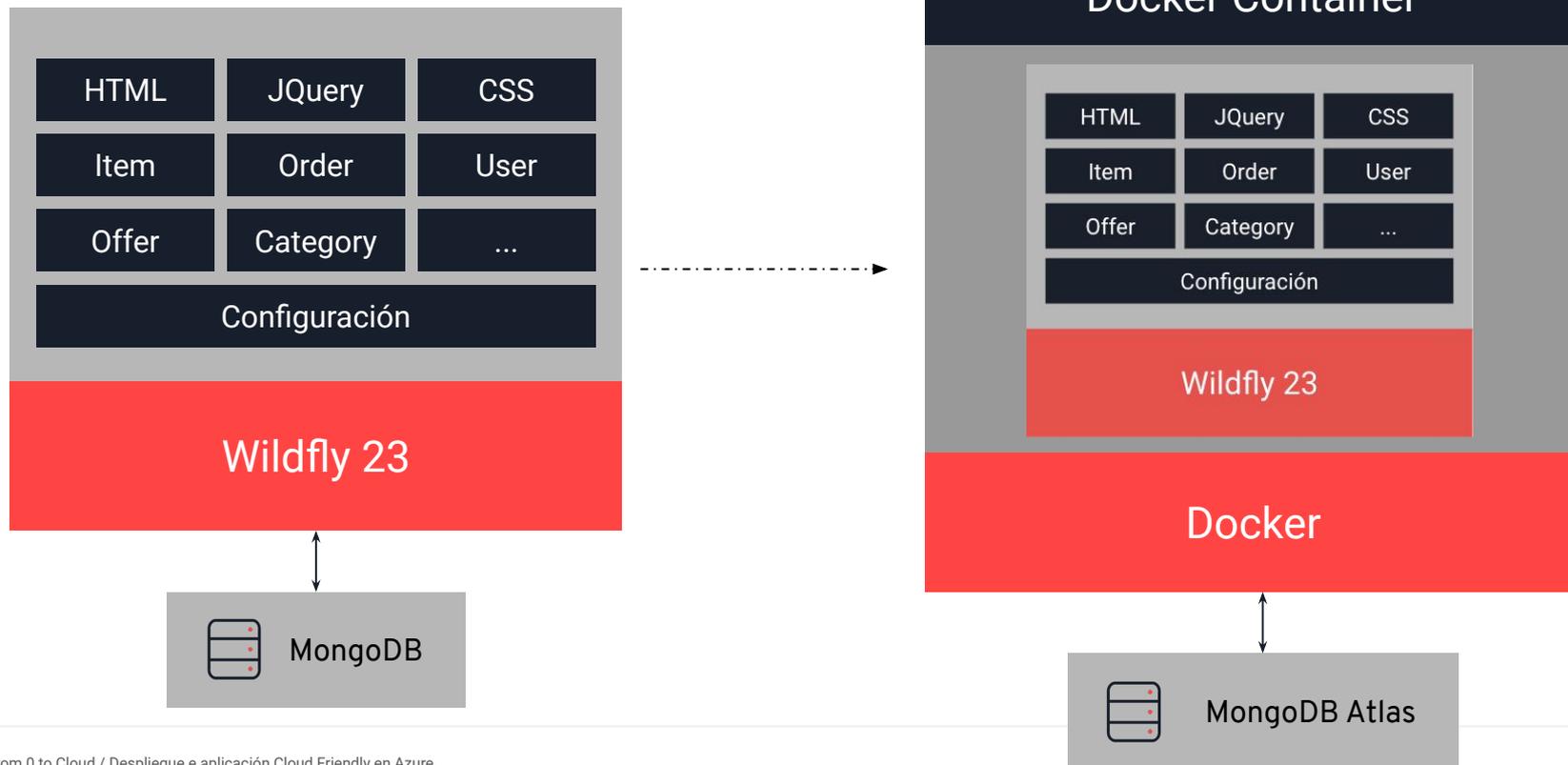
Nivel de Madurez



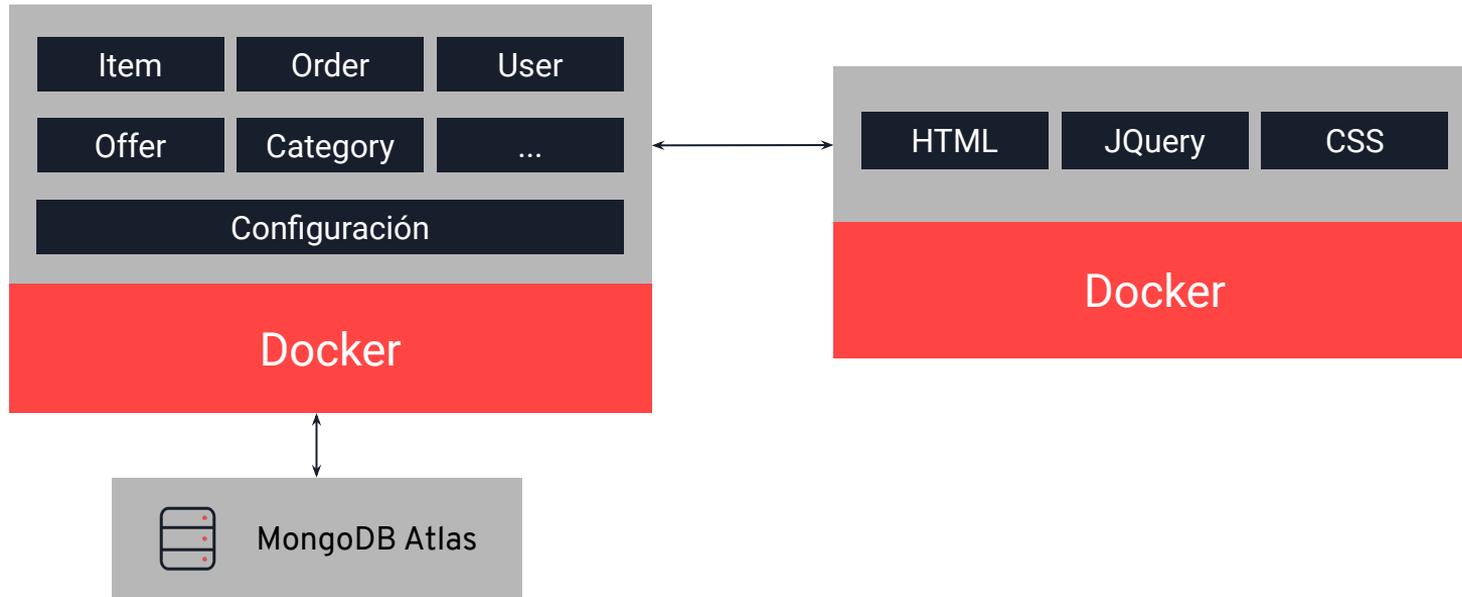
# Estado Inicial → Monolito



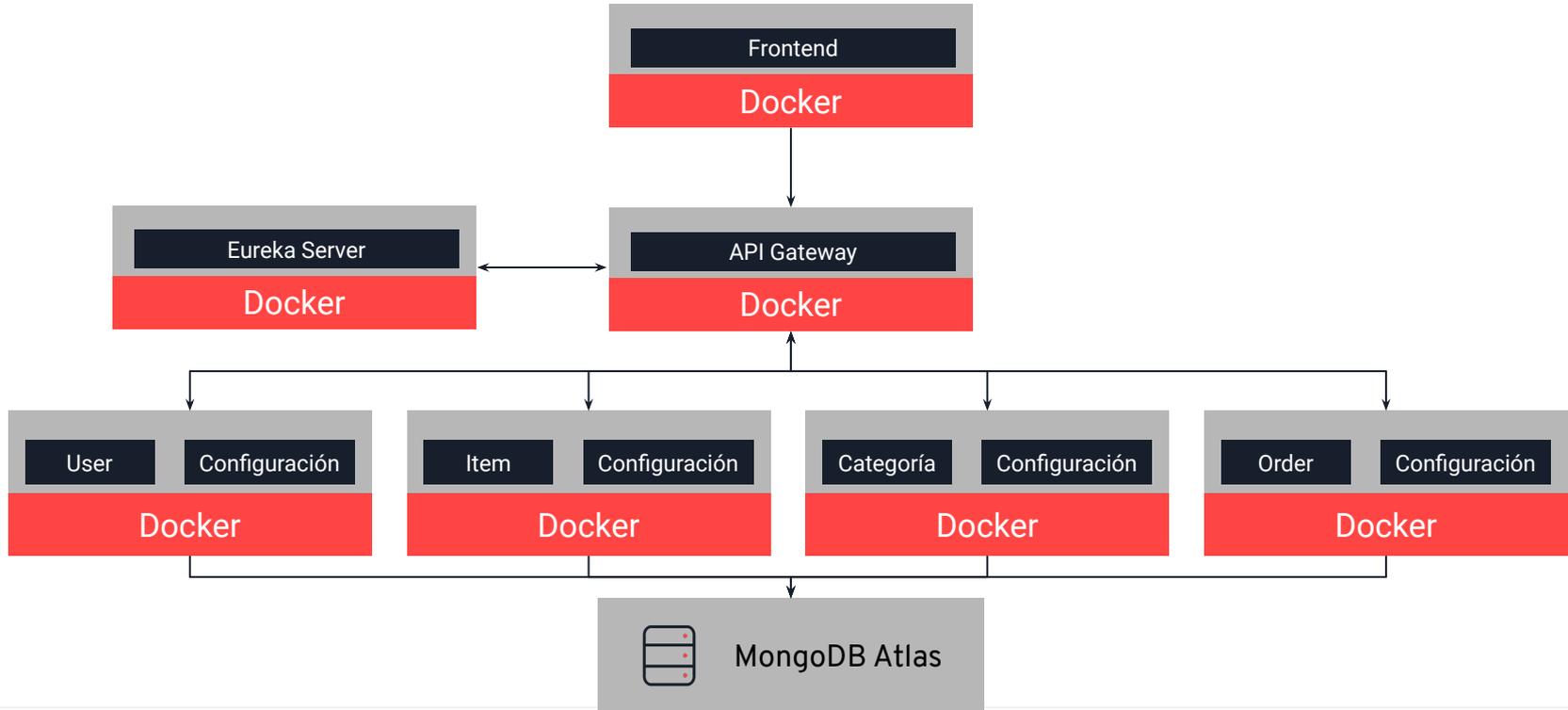
# Fase 0 → Lift & Shift



# Fase 1 → Rompimos el monolito



## Fase 2 → *μ-todo*



# Aviso a desarrolladores

El código que vais a ver a continuación no se acerca ni de lejos a un código bien desarrollado, por lo que puede herir sensibilidades, **tomadlo como ejemplo.**

**NO USAR EN PRODUCCIÓN**



# 02.



From 0 to Cloud

Cloud Friendly en Azure.

# Aterrizando en Azure.

# Estrategias de migración.

El despliegue de una **solución existente** en la nube suele llegar precedida de una **análisis** de qué estrategia es la más conveniente. Existen distintas aproximaciones comúnmente conocidas como las **cinco Rs**.



**Rehost**



**Refactor**



**Revise**

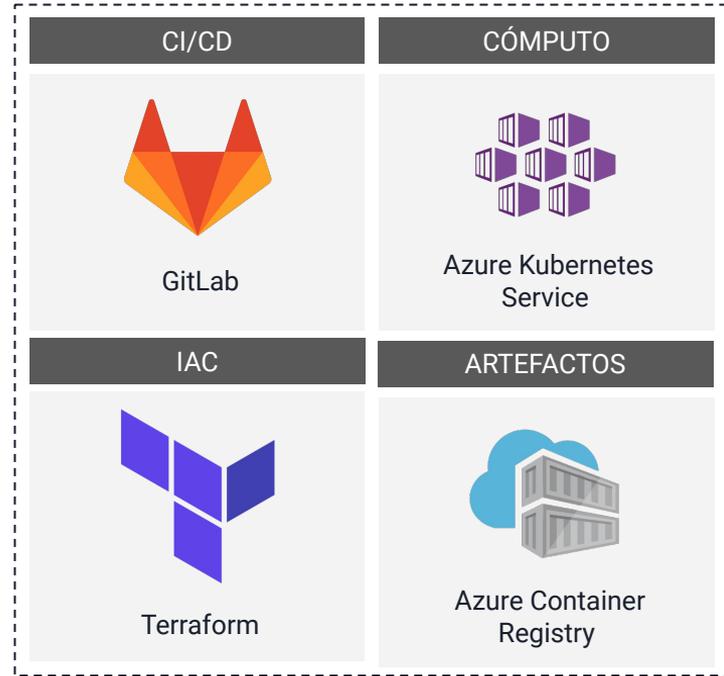
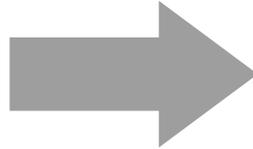
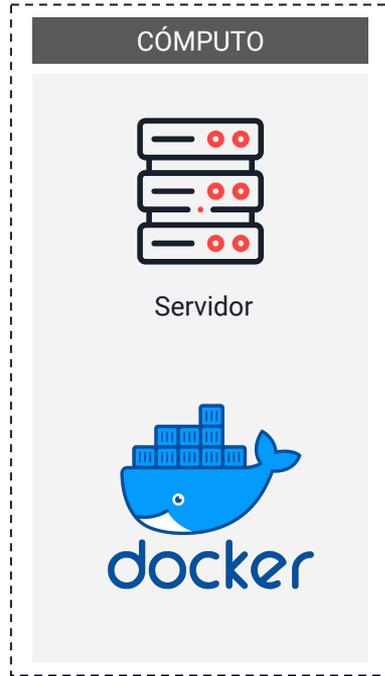


**Rebuild**

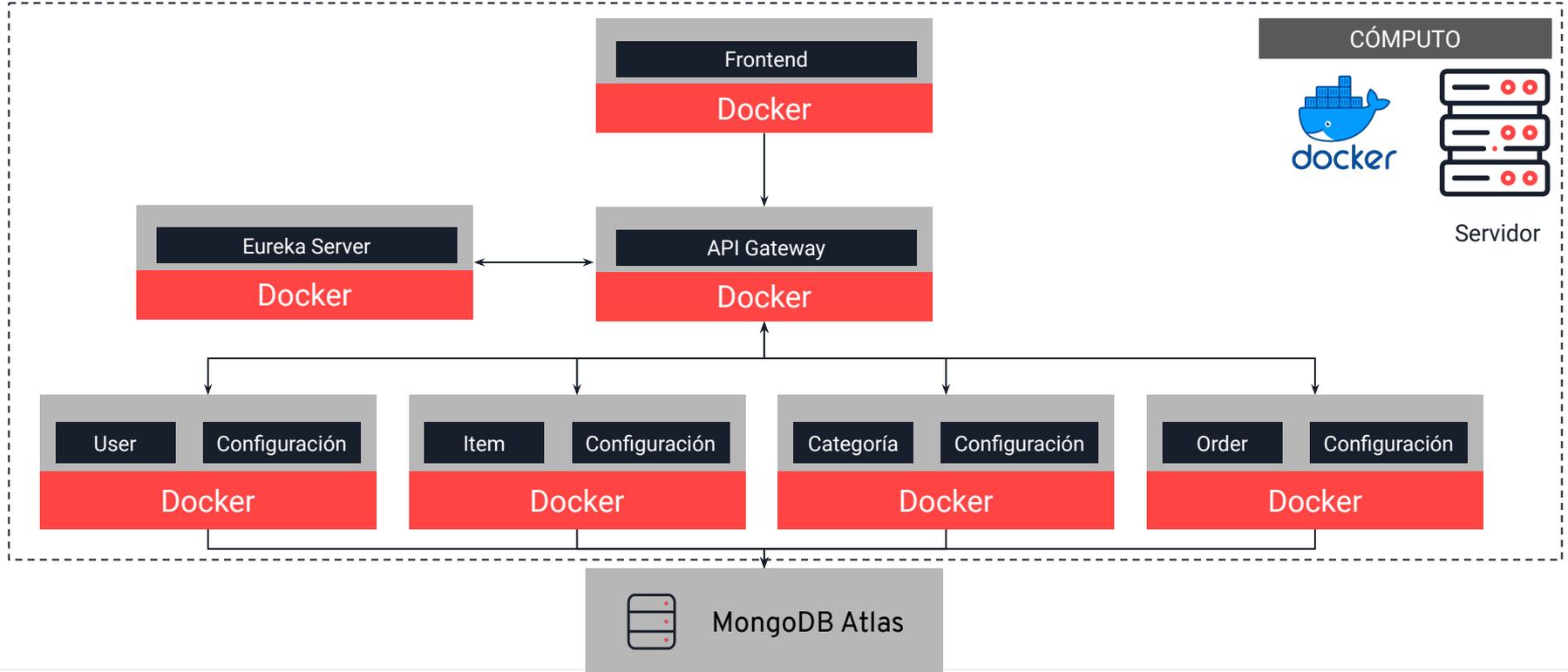


**Replace**

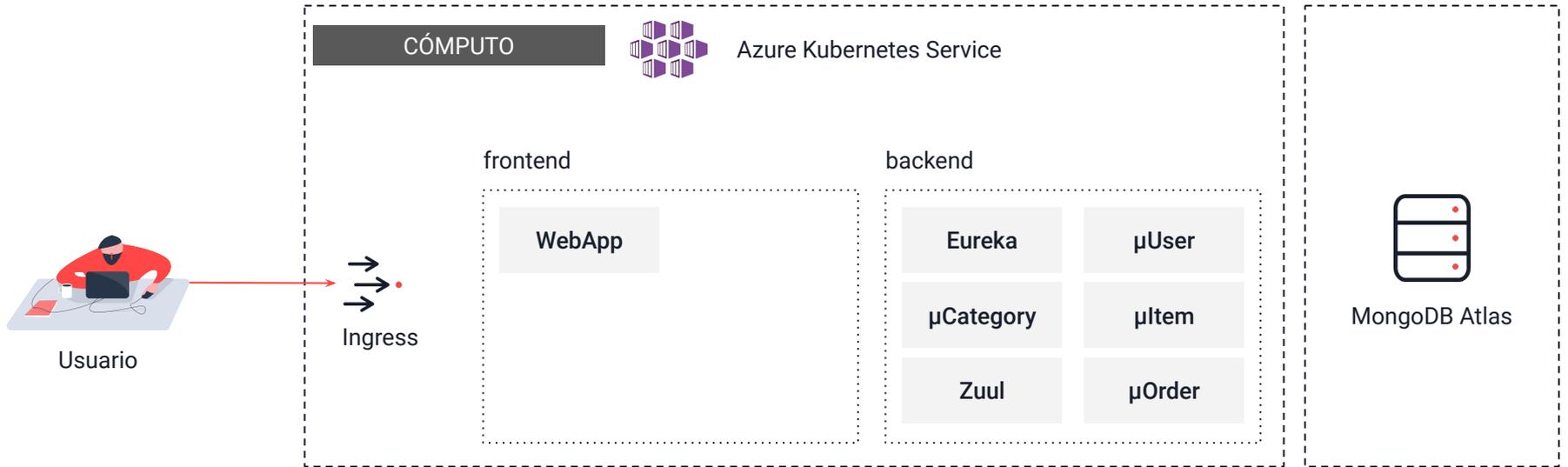
# Solución arquitectónica.



# Despliegue original.



# Despliegue objetivo.



# 03.

THOUGHT  
CATALOG



From 0 to Cloud



Cloud Friendly en Azure.

## Re-Cap.



# Qué hemos visto.

- 01 — Hemos adaptado el despliegue de un ecosistema de microservicios y frontal web de **Docker a Kubernetes**.
- 02 — En primer lugar, hemos declarado la infraestructura sobre **Azure** utilizando **Terraform** como herramienta IaC.
- 03 — En segundo lugar, hemos orquestado los distintos despliegues, tanto de infraestructura como de aplicación, sobre **GitLab Pipelines**.
- 04 — Por último, hemos definido los despliegues sobre **Kubernetes** para declarar los objetos necesarios.

# 04.



From 0 to Cloud



Cloud Friendly en Azure.

## What's Next? .

# Y a partir de ahora ¿qué?

- 01 — Despliegue en AWS usando mismo procedimiento que hoy y herramientas propias de AWS.
- 02 — Inclusión de servicios Cloud para API Gateway y almacenamiento de información más allá de el uso de Atlas.
- 03 — Incluiremos elementos Serverless que amplíen y mejoren la arquitectura.
- 04 — Creación de catálogos de componentes y catálogo de microservicios.



**¡Muchas  
Gracias!**