



PREPÁRATE  
PARA SER EL  
**MEJOR**



+ **ENTREMIENTO  
EXPERIENCIA**



**BIENVENIDOS.**





# Microservicios EDA, CQRS, SAGA y KAFKA

## Introducción y Arquitectura Base

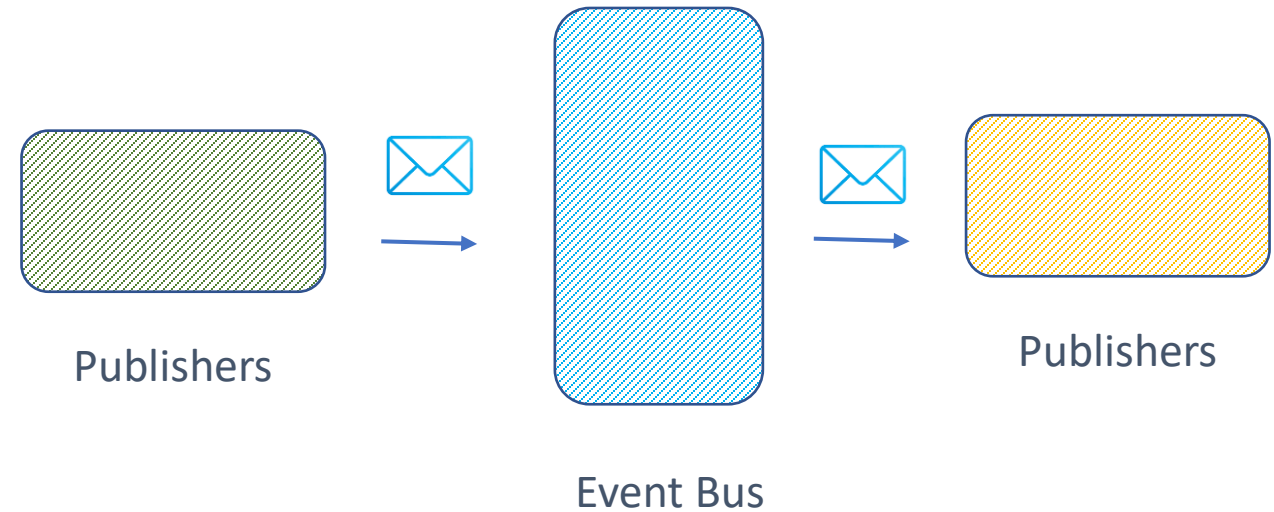
**Ing. Aristedes Novoa**  
Instructor Java  
[anovoa@galaxy.edu.pe](mailto:anovoa@galaxy.edu.pe)



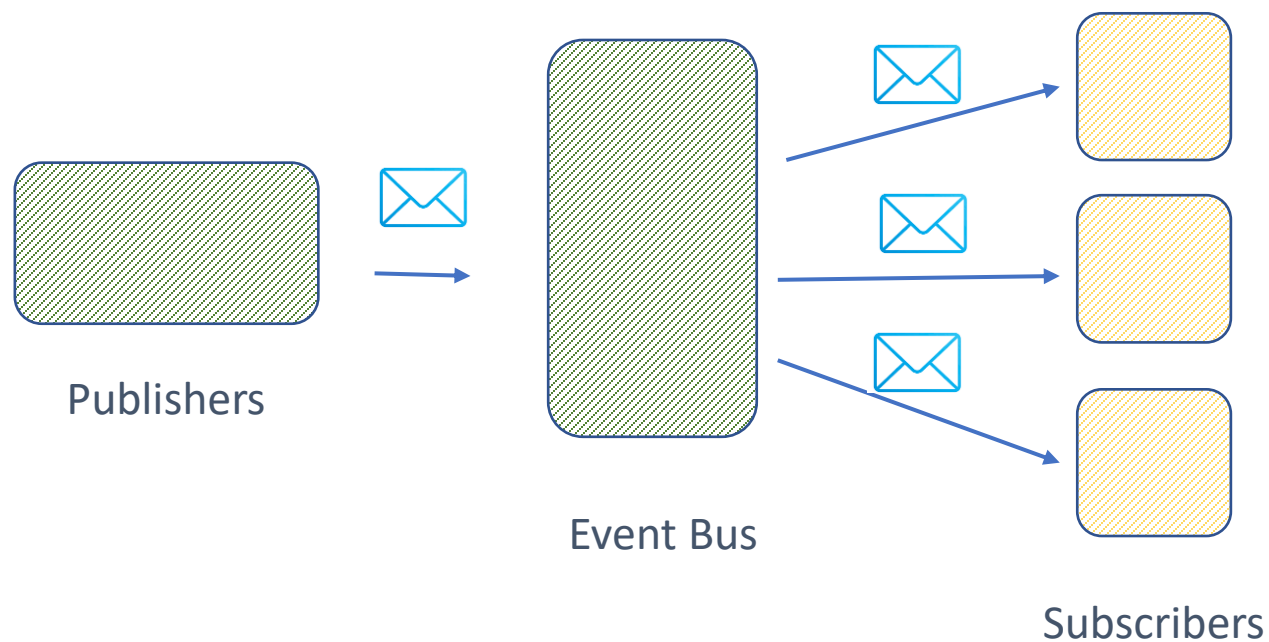
# Arquitectura basada en eventos (EDA)



La arquitectura basada en eventos (EDA) es un patrón de arquitectura moderna diseñado a partir de aplicaciones, servicios o microservicios desacoplados que publican, consumen o enrutan eventos.



# Publisher-Subscriber Patterns



SQS-AWS



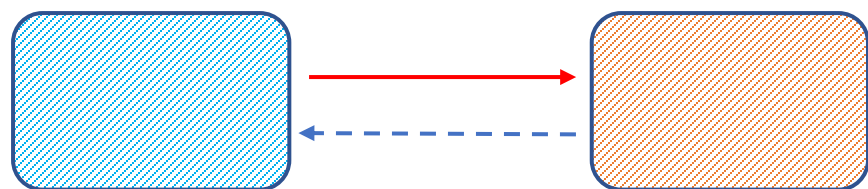
Service Bus-Azure



Pub/Sub-Azure



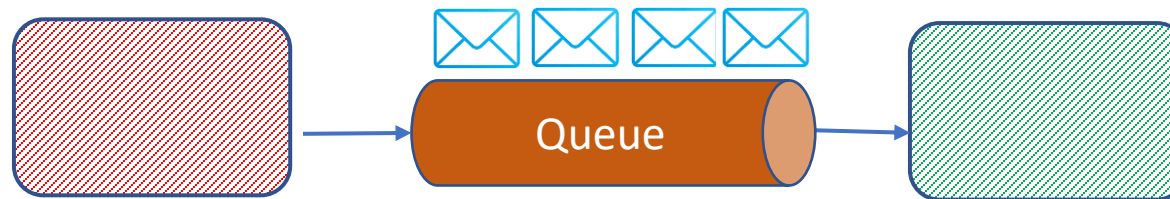
# Communication Síncrona vs Asíncrona



Sender

Receiver

Request/Response



Sender

Receiver

Fire & Forget

<https://www.reactivemanifesto.org/es>



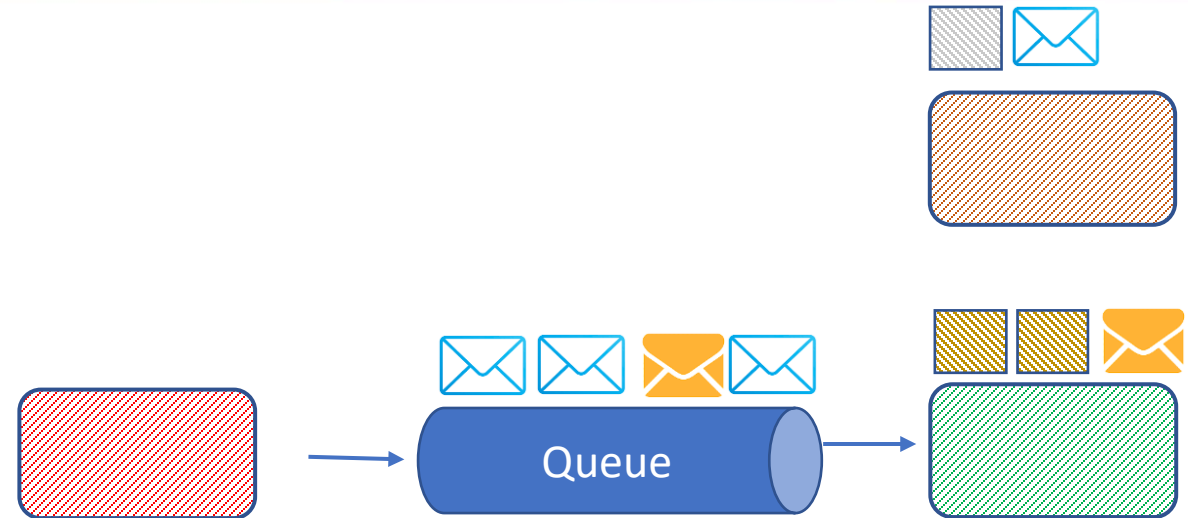
# Comunicación Síncrona vs Asíncrona



Sender

Receiver

Request/Response

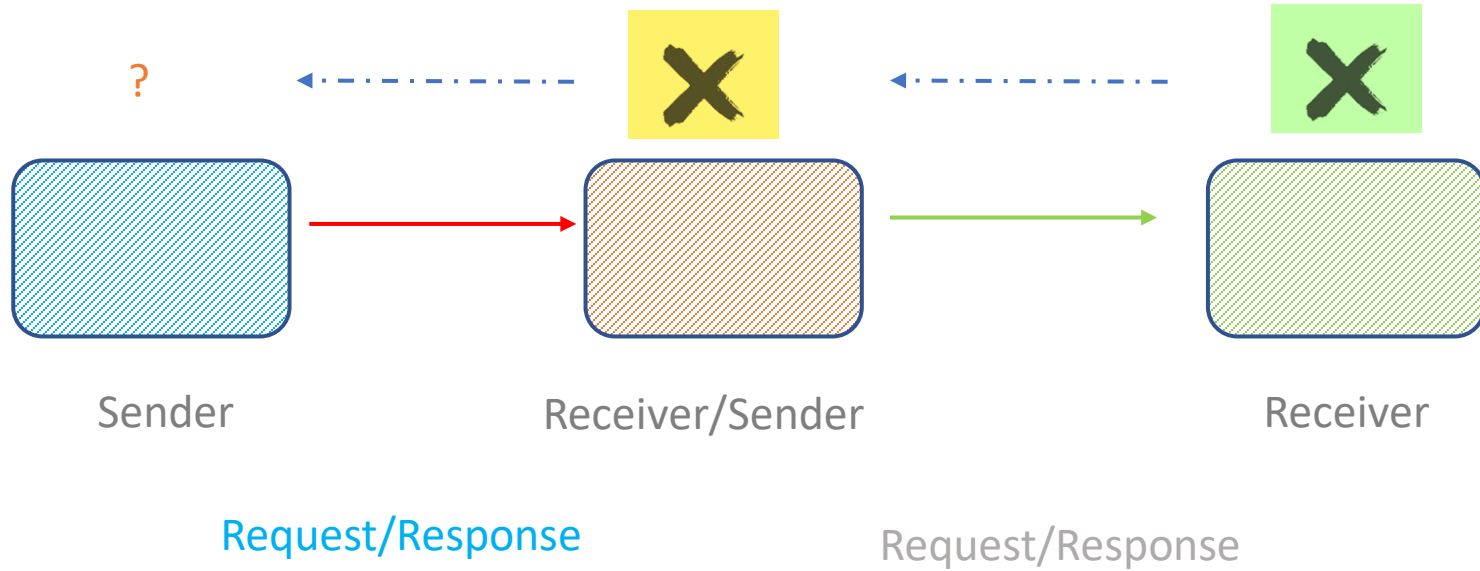


Sender

Receiver

Fire & Forget

# Communication Síncrona vs Asíncrona - Cadena



## Request/Response

1. Tiempo de respuesta - No óptimo
2. Capacidad Limitada (storage, memoria)
3. Tolerancia a fallos es limitada

## Fire/Forget

1. Tiempo de respuesta - optimizado
2. Capacidad extendida (storage, memoria)
3. Tolerancia a fallos gestionada, reintentos, reencolados, uso de protones, alta disponibilidad



## MS - NEGOCIO

---

1. [ms-negocio-gestion-productos](#) 8081 (ok)
2. [ms-negocio-gestion-clientes](#) 8082 (ok)
3. [ms-negocio-gestion-vendedor](#) 8083 (?)
4. [ms-negocio-gestion-pedidos](#) 8084 (ok)
5. [ms-negocio-gestion-clientes\\_backup](#) 8085 (??)

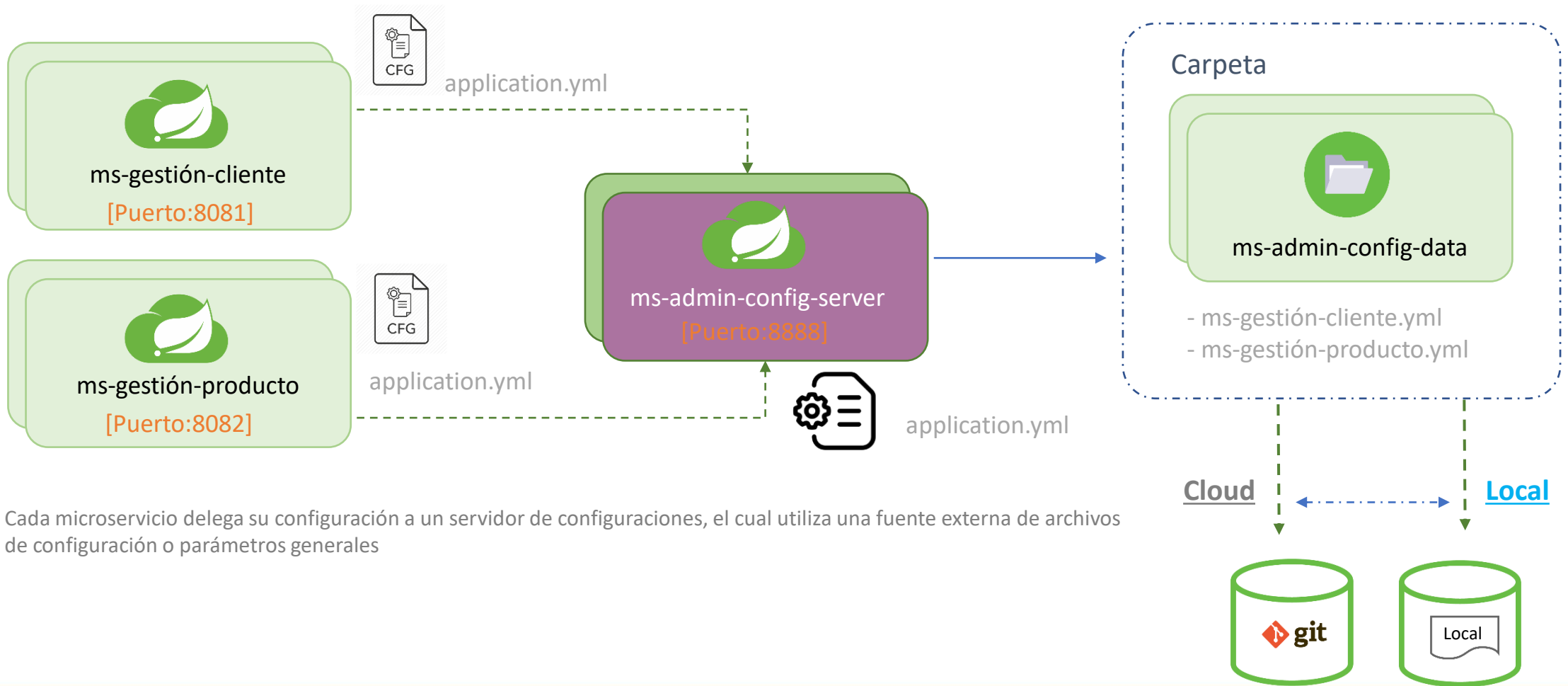
## MS – ADMINISTRACION(INFRAESTRUCTURA)

---

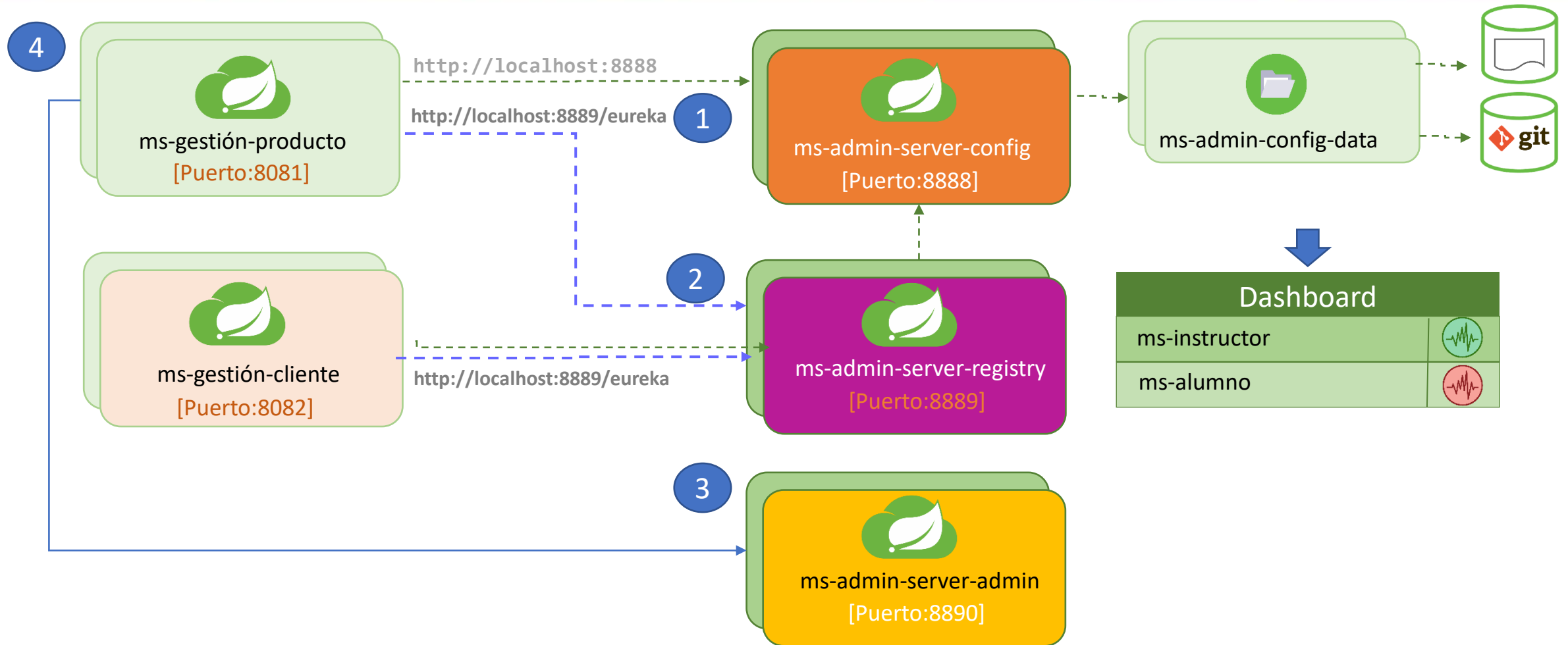
1. [ms-administracion-server-config](#) 8888 (ok)
2. [ms-administracion-server-registry](#) 8889 (ok)
3. [ms-administracion-server-admin](#) 8890 (ok)
4. [ms-administracion-api-gateway](#) 8887 (ok)

# Requisitos previos

Pattern	Consideraciones	Cumple <sup>1</sup>		Comentarios de revisión
		Si	No	
Decompose by businss capability	Incluir como mínimo 2(dos) procesos de negocio como mínimo(ejemplo: depósitos, retiros, estado de cuenta, <u>pedidos</u> , pagos, delivery)			
Externalized configuration	Crear un servidor de configuración y publicar los archivos de configuración en un repositorio en cloud. <u>En Kubernetes puede utilizar ConfigMap, Secrets en su remplazo.</u>			
Service Registry	Crear un servidor registro, <u>si utiliza Docker Compose o Kubernetes puede remplazar esta funcionalidad con sus configuraciones respectivas.</u>			
API Gateway	Implementar un API Gateway o remplazar con un equivalente si su despliegue lo realiza a nivel de Cloud			
Circuit Breaker (Reliability)	Implementar resiliencia con nivel 3. Si el primer servicio falla, se pone en marcha el segundo y si éste falla se activa el tercero. Se recomienda utilizar Resillence4J, pero puede utilizar otras tecnologías alternativas			



# MS Server Registry & Admin

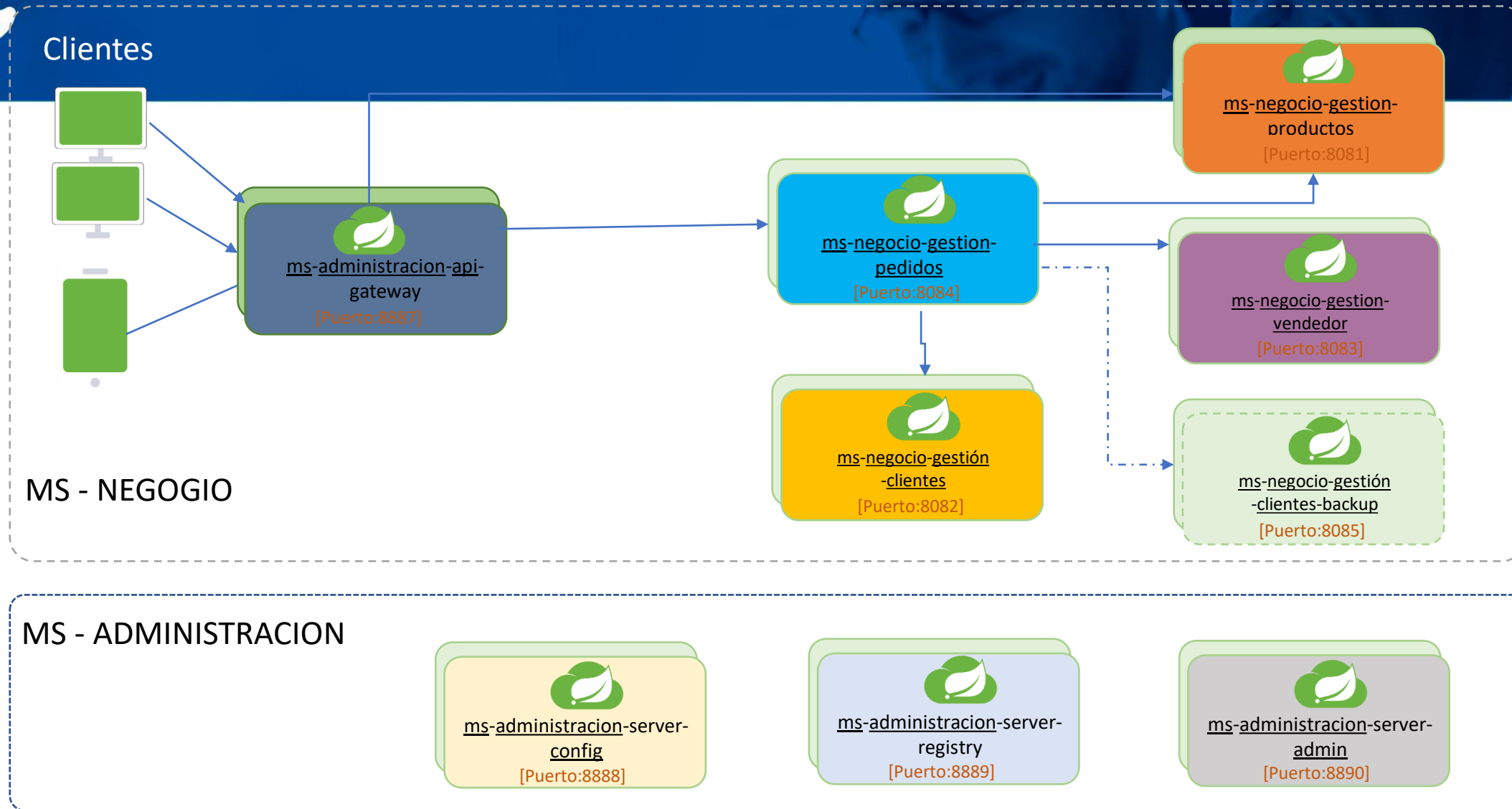


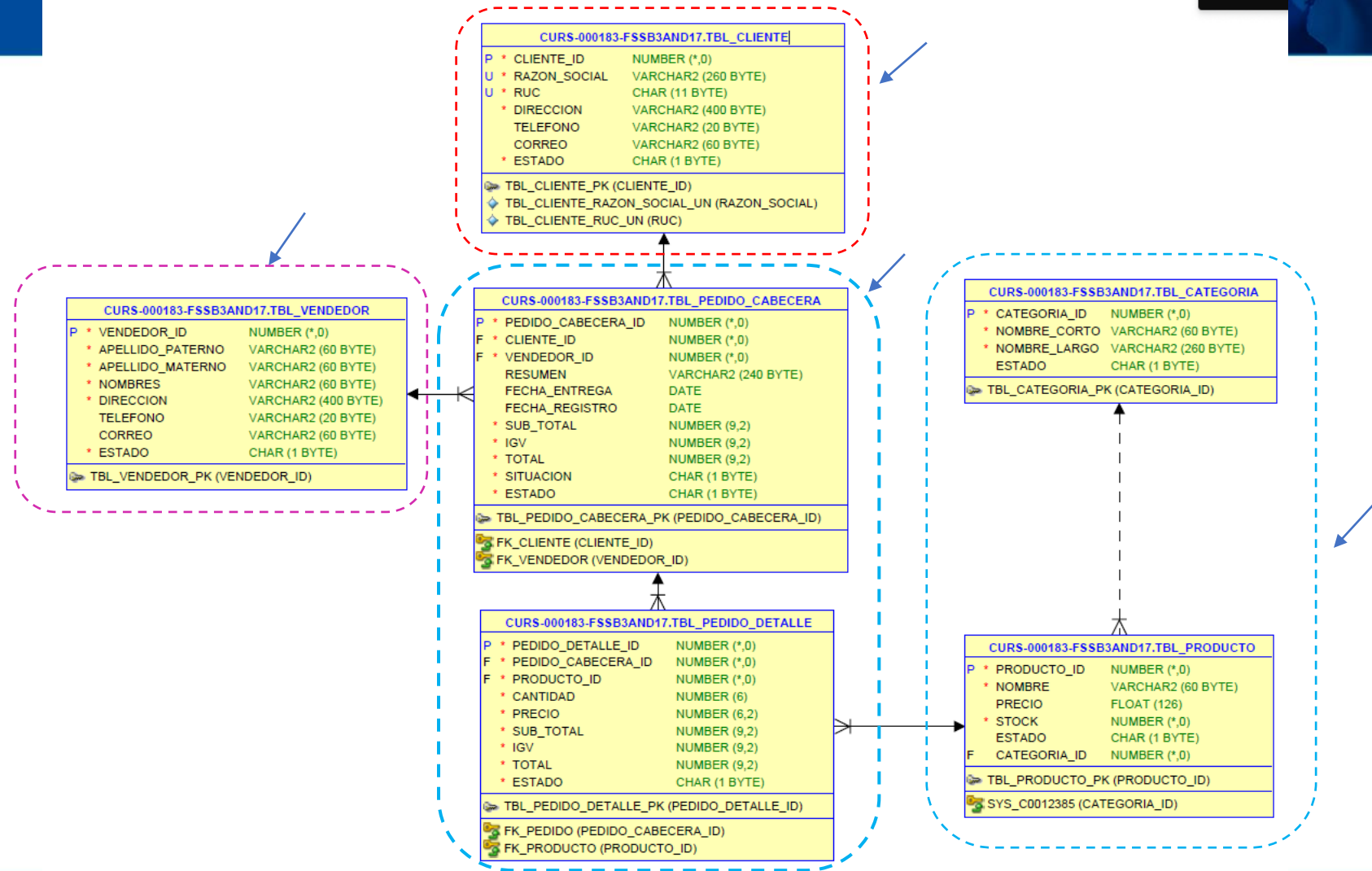


# Arquitectura de Microservicios base

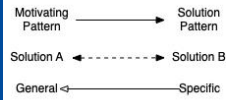


Avanzado

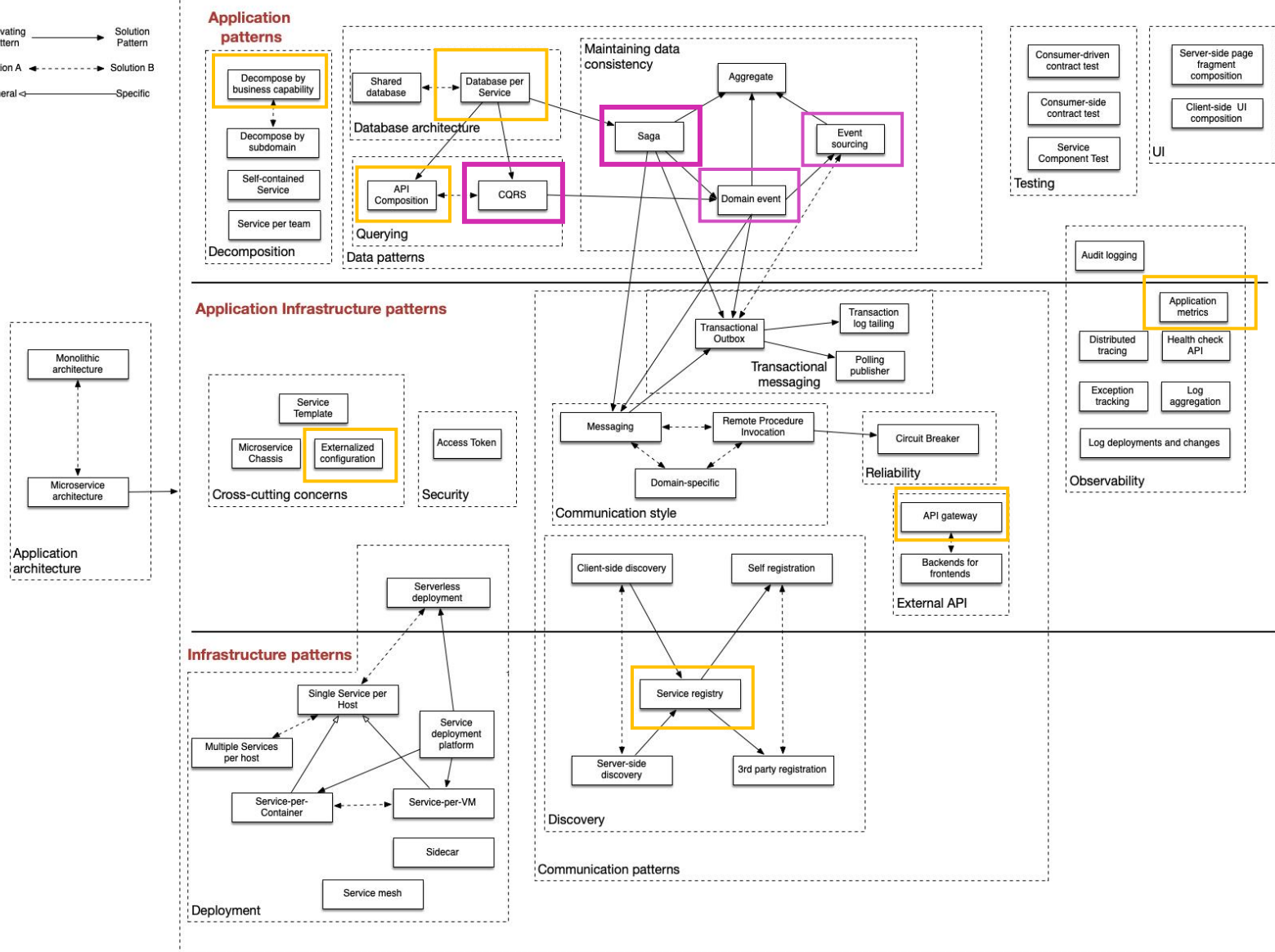








## The Microservice Architecture Pattern Language



Copyright © 2023. Chris Richardson Consulting, Inc. All rights reserved.

Learn-Build-Assess Microservices <http://adopt.microservices.io>

# LÍNEA DE ESPECIALIZACIÓN EN MICROSERVICIOS (PARTE I)

1

## Fundamentos y Arquitectura de Microservicios



Estamos aquí

Event Driven, Architecture RabbitMQ, Apache Kafka y Spring AMQP

Microservicios Reactivos con Spring WebFlux

Microservicios con Jakarta EE - Microprofile

**MICROSERVICIOS  
CON SPRING BOOT**

## Contenedores y Orquestación

Docker

Kubernetes

2

## Gestión, Comunicación y Seguridad de Microservicios

Seguridad de Microservicios

Gestión de Servicios con Service Mesh e Istio

Monitoreo y Observabilidad con Stack ELK, Grafan y Prometheus

Contenerización y Gobierno Docker y Kubernetes

## Base de Datos SQL

Oracle

MS SQL Server

3

## Gestión, Comunicación y Seguridad de Microservicios

→ Microservicios y QA

→ Microservicios y DevOps

Aplicaciones Web Angular, React y Vue

Aplicaciones Móviles Android, Ionic y React JS

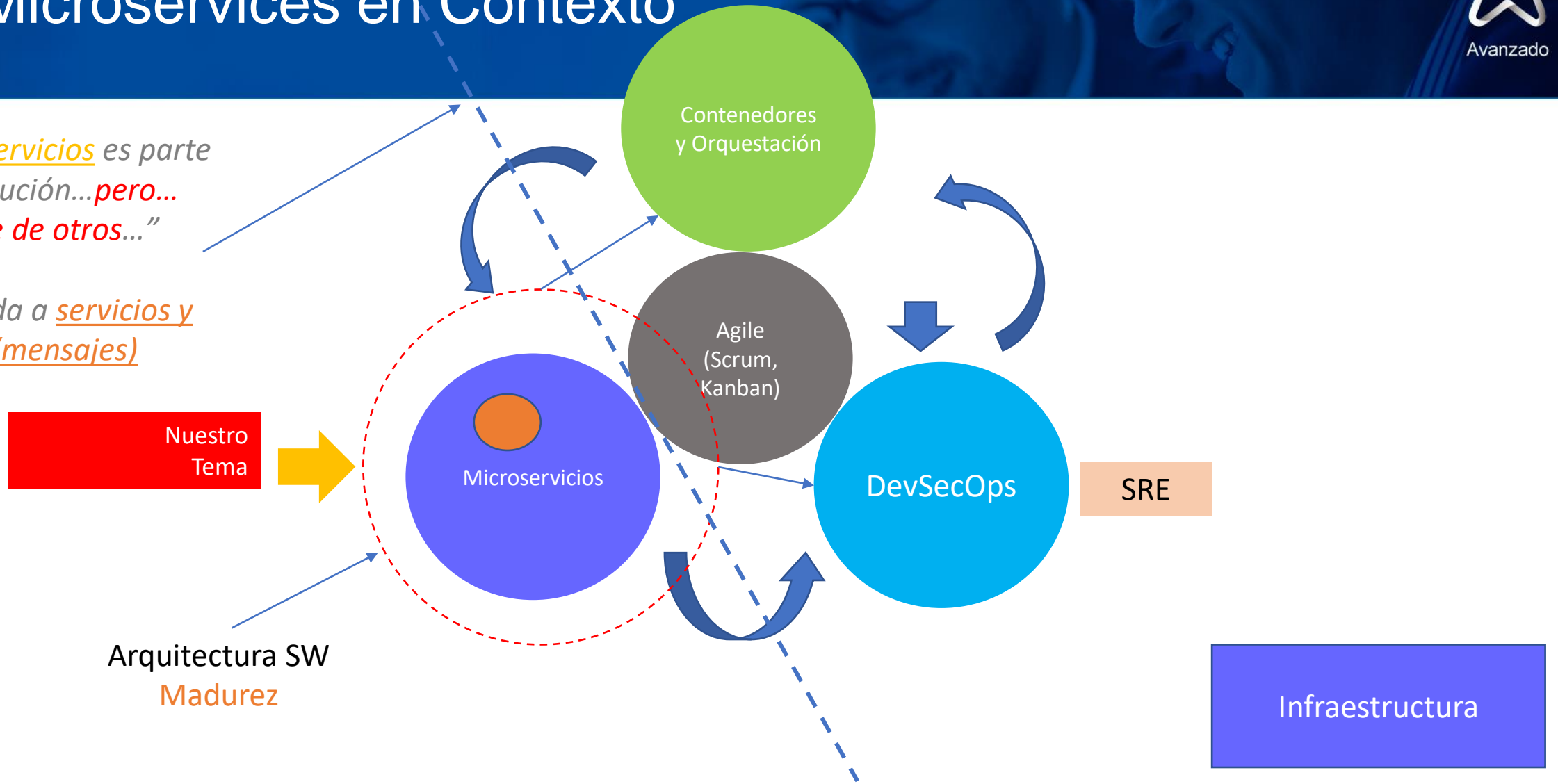
Microfronts

## Base de Datos NO SQL y Caché

Mongo DB

Cassandra

Redis





GALAXY  
TRAINING