

Master Profesionalizante en Ingeniería Informática

Cloud Computing: Servicios y Aplicaciones

T4. Desarrollo de servicios en SaaS

Arquitectura tradicional

3 capas:

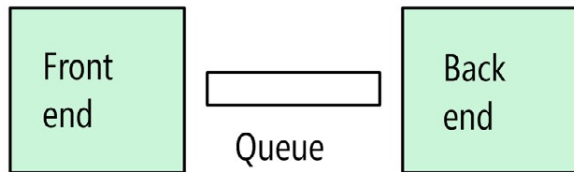
Presentación

Intermedio

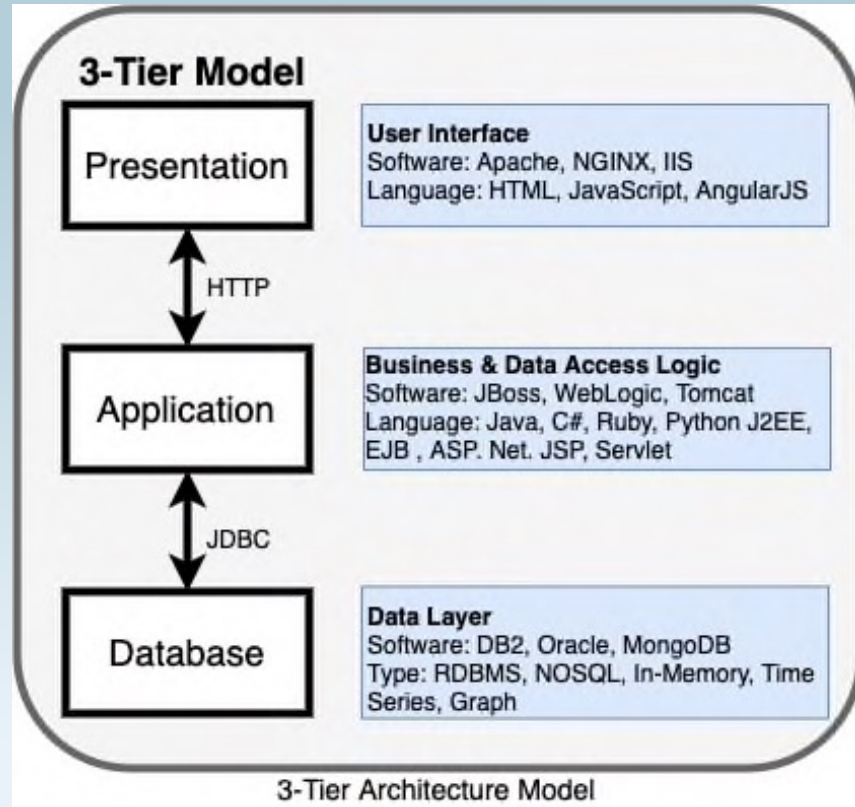
Datos

Desarrollo estático, asumiendo
servidores y red totalmente fiables
Sin capacidad de escalado

Arquitectura en 3 capas



Web-queue-worker

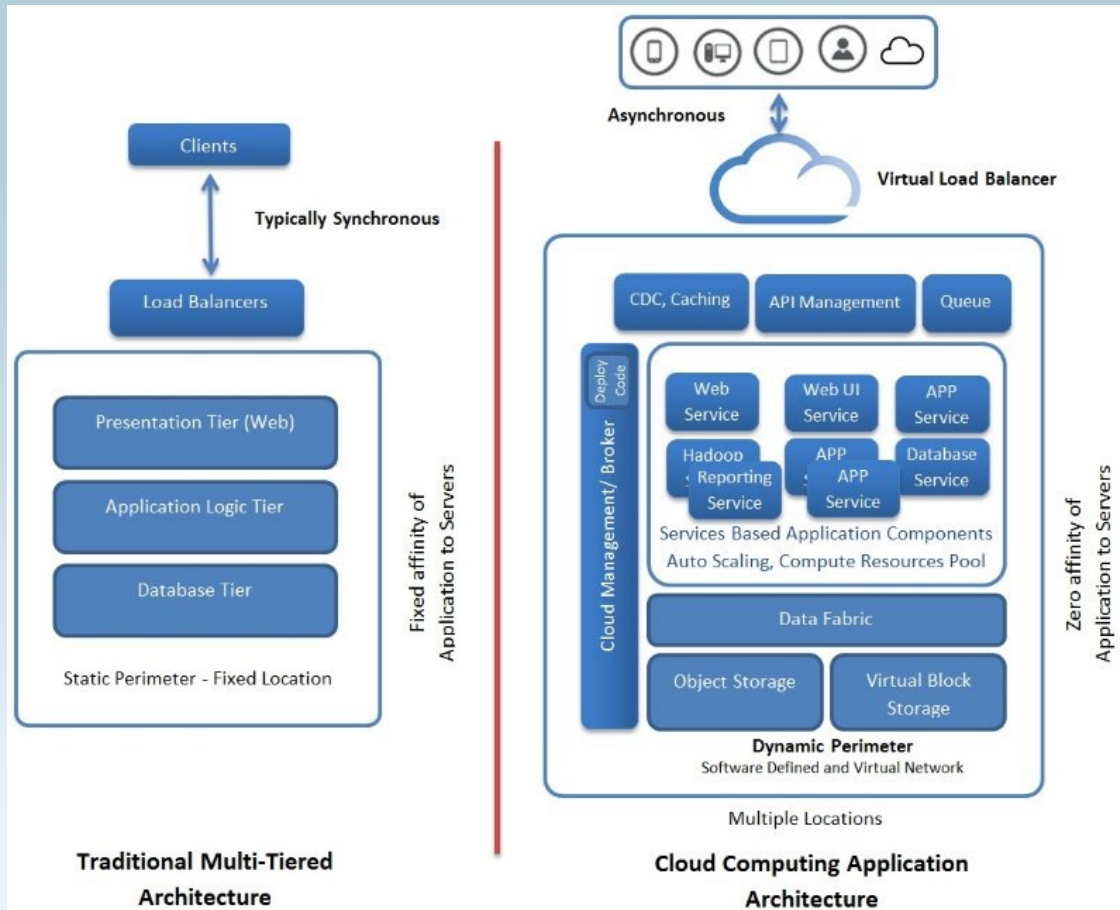


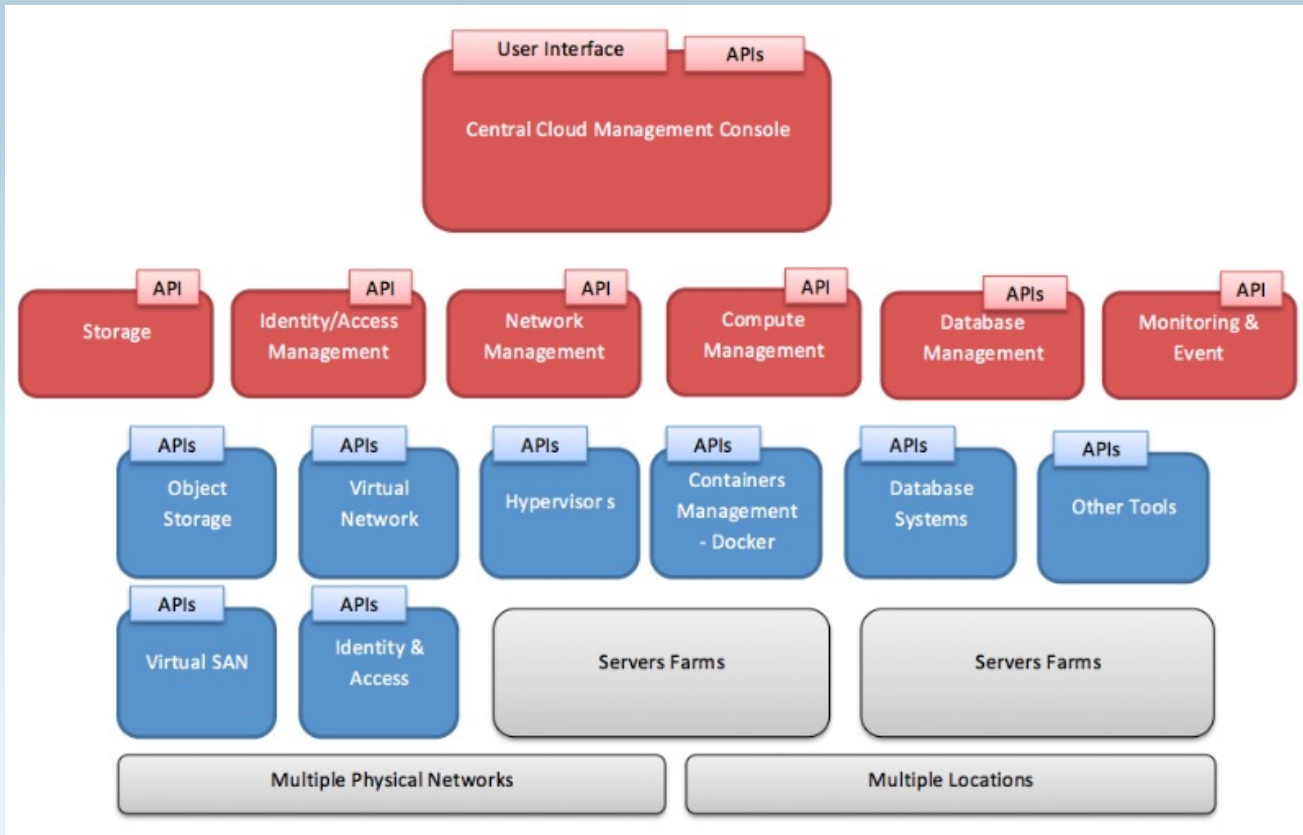
**La arquitectura tradicional no es
suficientemente flexible para
incorporar:**

Elasticidad

Auto-servicio

**Variedad en los soportes de
computación y almacenamiento**

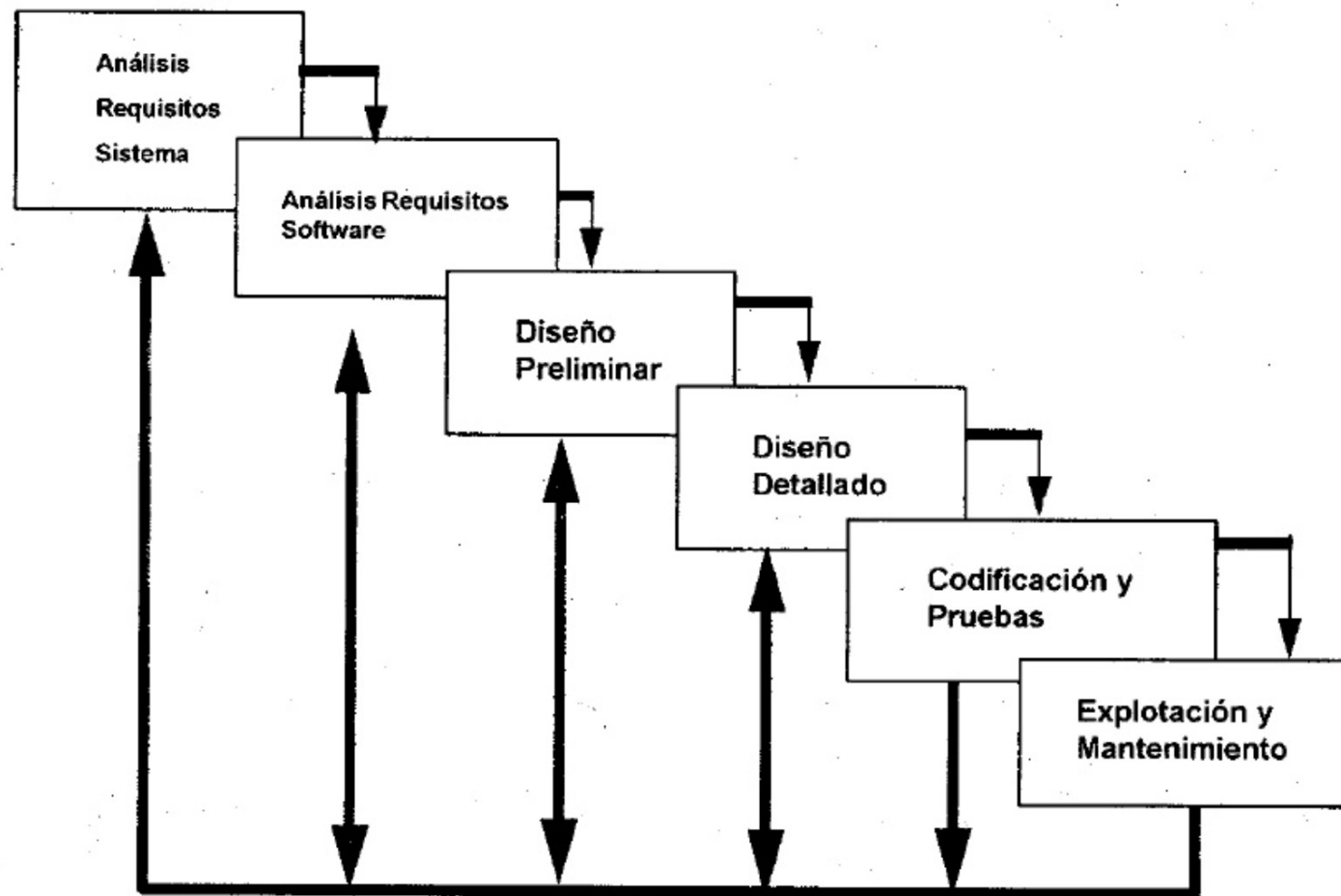




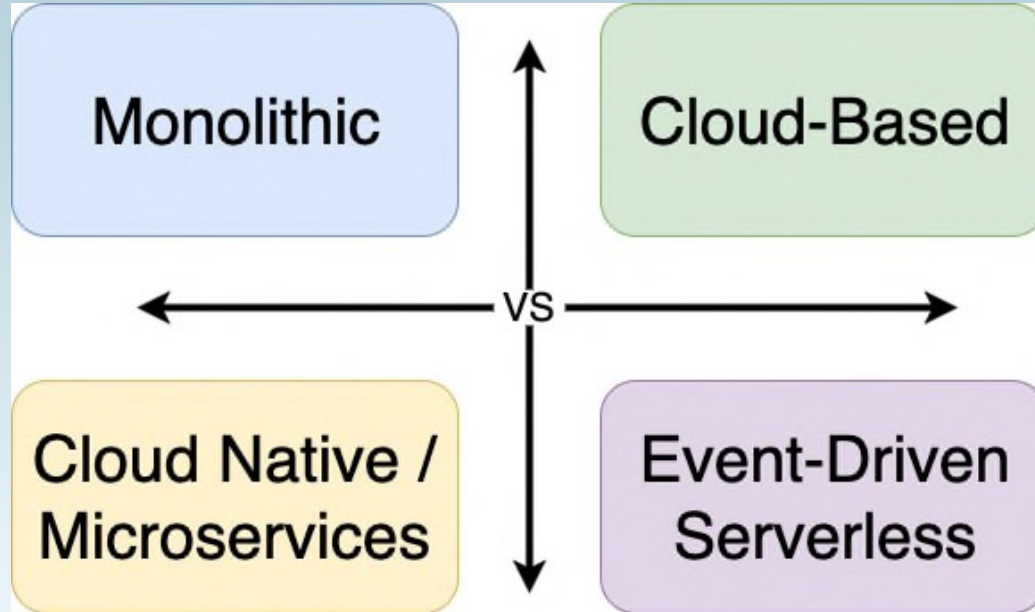
¿Problema con el modelo de 3 capas?

**Anticuoado: modelo pre-cloud
Las aplicaciones puede crecer
mucho y requerir muchos cambios**

MODELO EN CASCADA

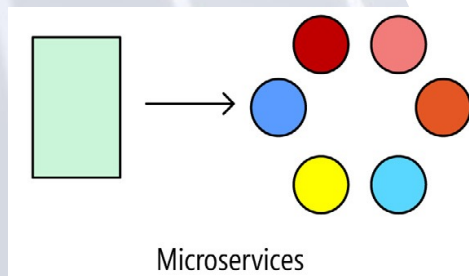


Arquitecturas de aplicaciones cloud



[Four Architecture Choices for Application Development in Cloud Age](#)

Microservicios



Microservicios (según IBM)

Aplicación: conjunto de servicios individuales (pequeños) totalmente desacoplados

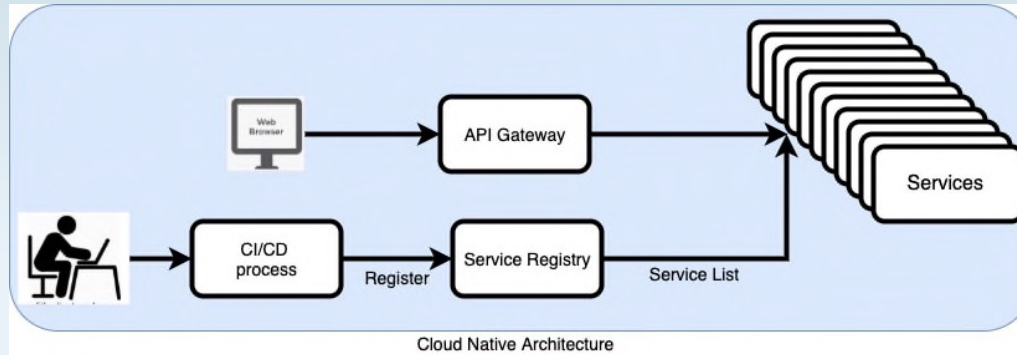
Desarrollo, validación y despliegue individual

Se comunican mediante API, flujos de eventos o gestores de mensajes

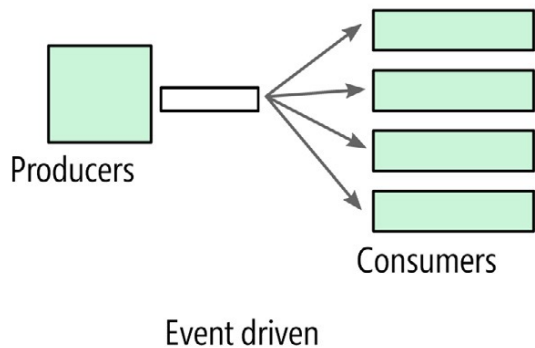
Gestión descentralizada

Arquitectura Cloud Native

DevOps
Continuous Integration
Microservicios
Contenedores



Serverless dirigida por eventos

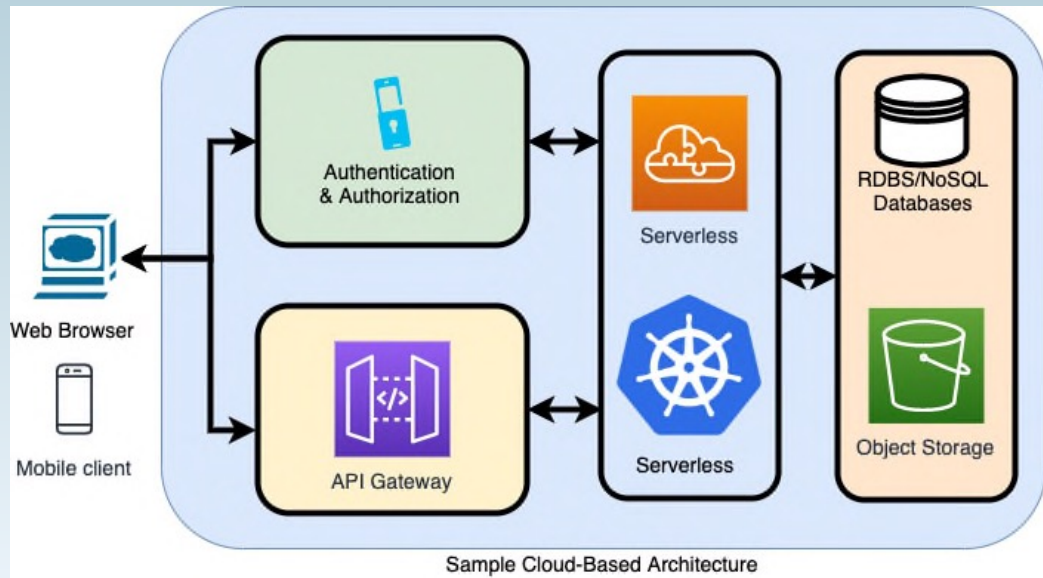


Sistemas desacoplados que responden a eventos el desarrollador simplemente sube su código, en forma de microservicio, y se despreocupa de todo lo demás.

Tipos:

- Functions-as-a-Service
- Backend-as-a-Service
- Mobile-Backend-as-a-Service

Arquitectura basada en la nube





Aplicación cloud tradicional

Elementos

Balanceadores de carga

Servidores web

Servidores de aplicaciones

Bases de datos

Características:

Escalabilidad elástica

Redes definidas por software

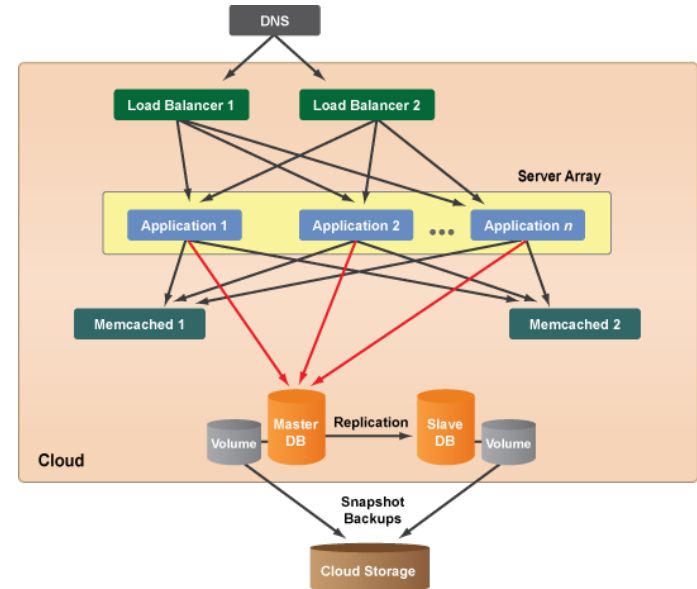
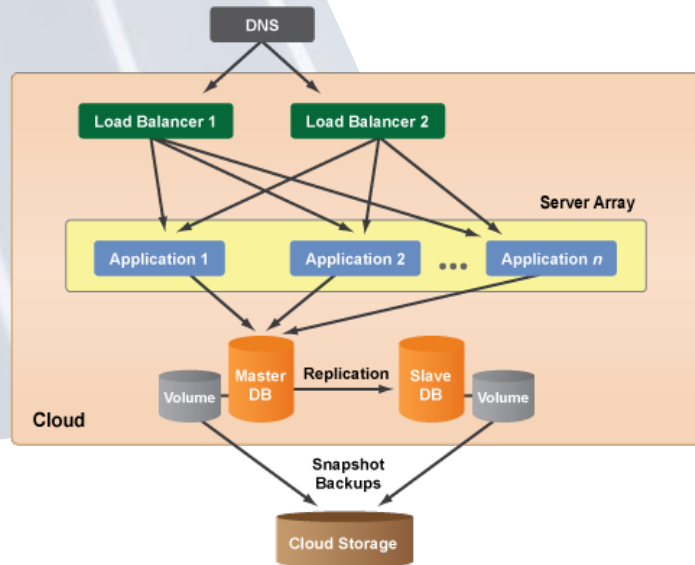
Auto-aprovisionamiento

Alta disponibilidad

Conjunto de recursos amplio

Architecture style	Dependency management	Domain type			
N-tier	Horizontal tiers divided by subnet.	Traditional business domain. Frequency of updates is low.	CQRS	Read/write segregation. Schema and scale are optimized separately.	Collaborative domain where lots of users access the same data.
			Event-driven architecture	Producer/consumer. Independent view per sub-system.	IoT and real-time systems.
Web-Queue-Worker	Front and backend jobs, decoupled by async messaging.	Relatively simple domain with some resource intensive tasks.	Cloud Application Architecture Guide		
			Big data	Divide a huge dataset into small chunks. Parallel processing on local datasets.	Batch and real-time data analysis. Predictive analysis using ML.
Microservices	Vertically (functionally) decomposed services that call each other through APIs.	Complicated domain. Frequent updates.	Big compute	Data allocation to thousands of cores.	Compute intensive domains such as simulation.

Cloud Computing System Architecture Diagrams



Capa de gestión de infraestructura

Automatización del despliegue y
gestión de infraestructura:
servidores, red,
almacenamiento, bases de datos,
seguridad, ...

Capa de gestión de infraestructura:
proporciona APIs para usuarios y
desarrolladores para gestión de la
infraestructura

Programación de infraestructura

Escribes código para:
la funcionalidad de tu aplicación
**crear y gestionar la
infraestructura de tu aplicación**

Principales herramientas

Ansible
Puppet
Chef