



Curso Práctico de Arquitectura Backend

Jorge Villalobos Gutiérrez
twitter: @jorge_vgut
youtube.com/latincoder



A quién va dirigido este curso

Personas interesadas en:

- Desarrollo y arquitectura de software en backend.
- Entender cómo funciona un sistema distribuido en backend.
- Aprendizaje a través de la práctica.

¿Qué es el Backend?



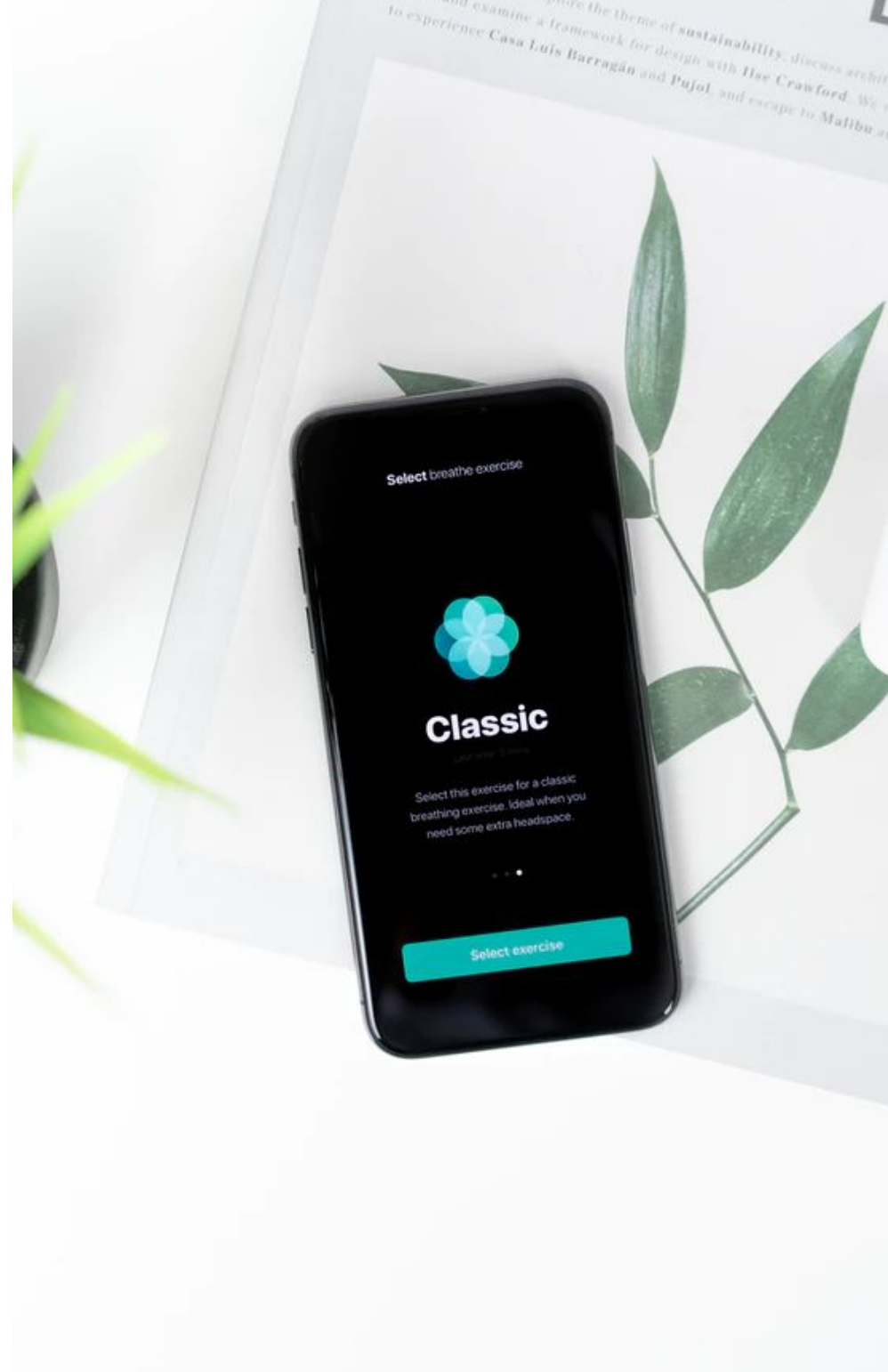


¿Qué es el Backend?

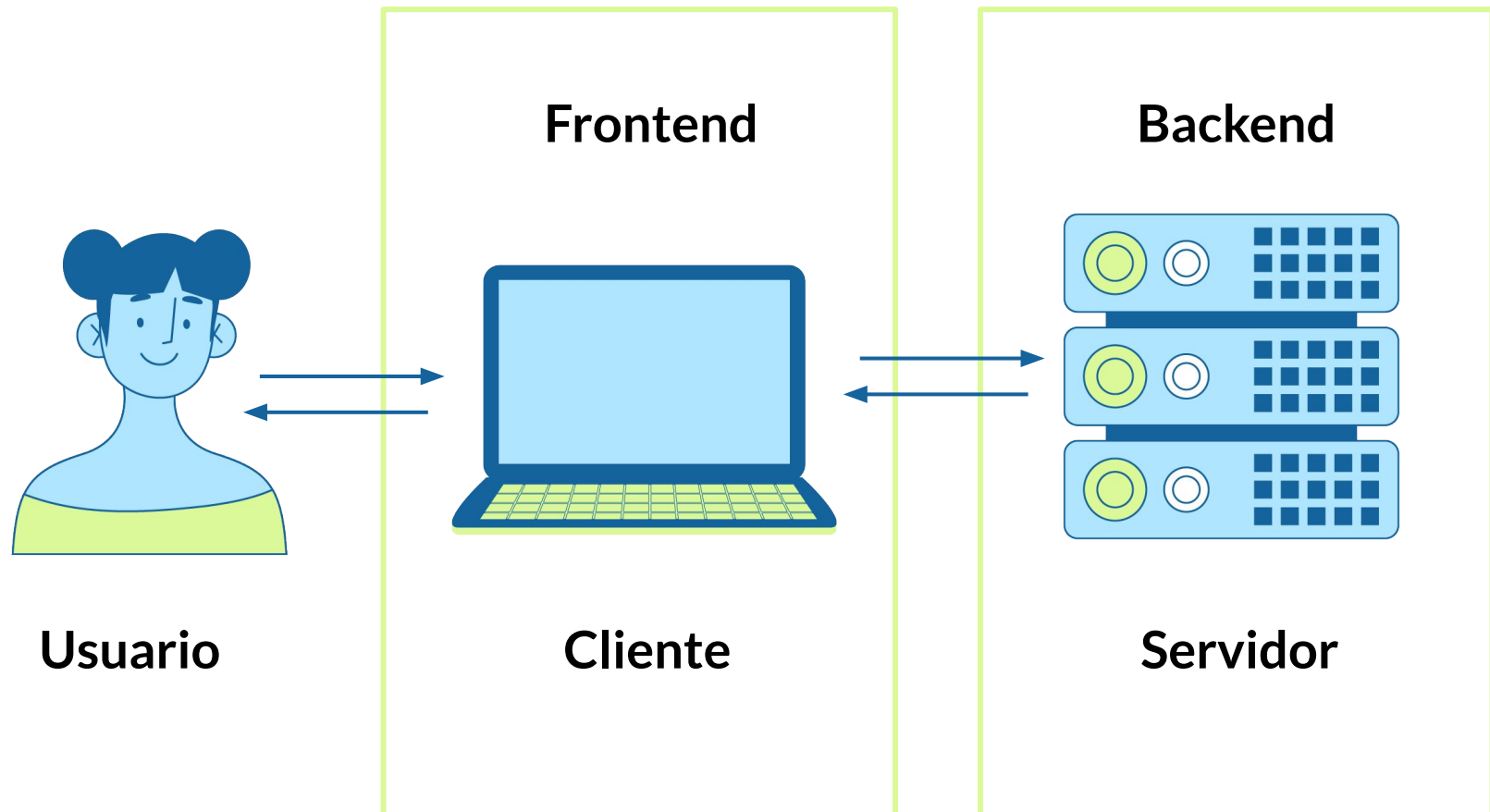
Tomando como base un modelo cliente- servidor. Se suele usar el término “backend” para referirse al software ejecutado en el servidor.

¿Qué es el Backend?

Una aplicación podría ser únicamente la “entrada” al Backend



¿Qué es el Backend?





¿Qué es el Backend?

El Backend de un sistema puede expandirse mucho más allá de una arquitectura cliente-servidor.

Sistemas distribuidos





De monolítico a distribuido

→ Sistema Monolítico

Es un sistema conformado por una sola “pieza”.


→ Sistema Distribuido

Compuesto de múltiples componentes que se comunican entre sí usando redes.

Ejemplo sistema monolítico

Un único servidor que hospeda uno o más sitios web con LAMP stack.

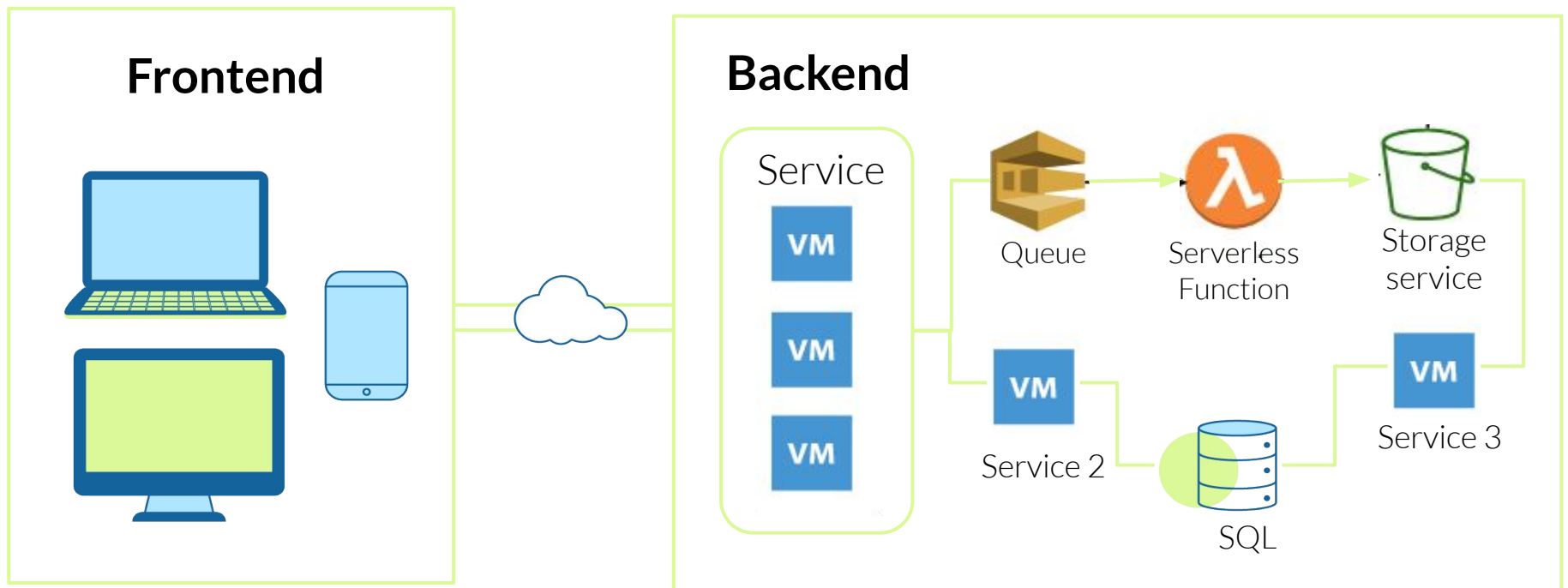




Significado de “Distribuido” en la práctica

Aunque sabemos que es un sistema conformado por muchos componentes conectados por medio de redes, puede manifestarse en diferentes escalas e implementar múltiples patrones de arquitectura.

Ejemplo del backend de un sistema distribuido



Elementos en los sistemas distribuidos



Elementos que podemos encontrar

- Load Balancers
- Máquinas Virtuales
- Bases de Datos (Relacionales y No Relacionales)
- Servicios de Colas (ej. SQS)
- Servicios de Storage (ej. S3)
- Servicios Publisher-Subscriber (ej. SNS)
- Pipelines
- etc.

“

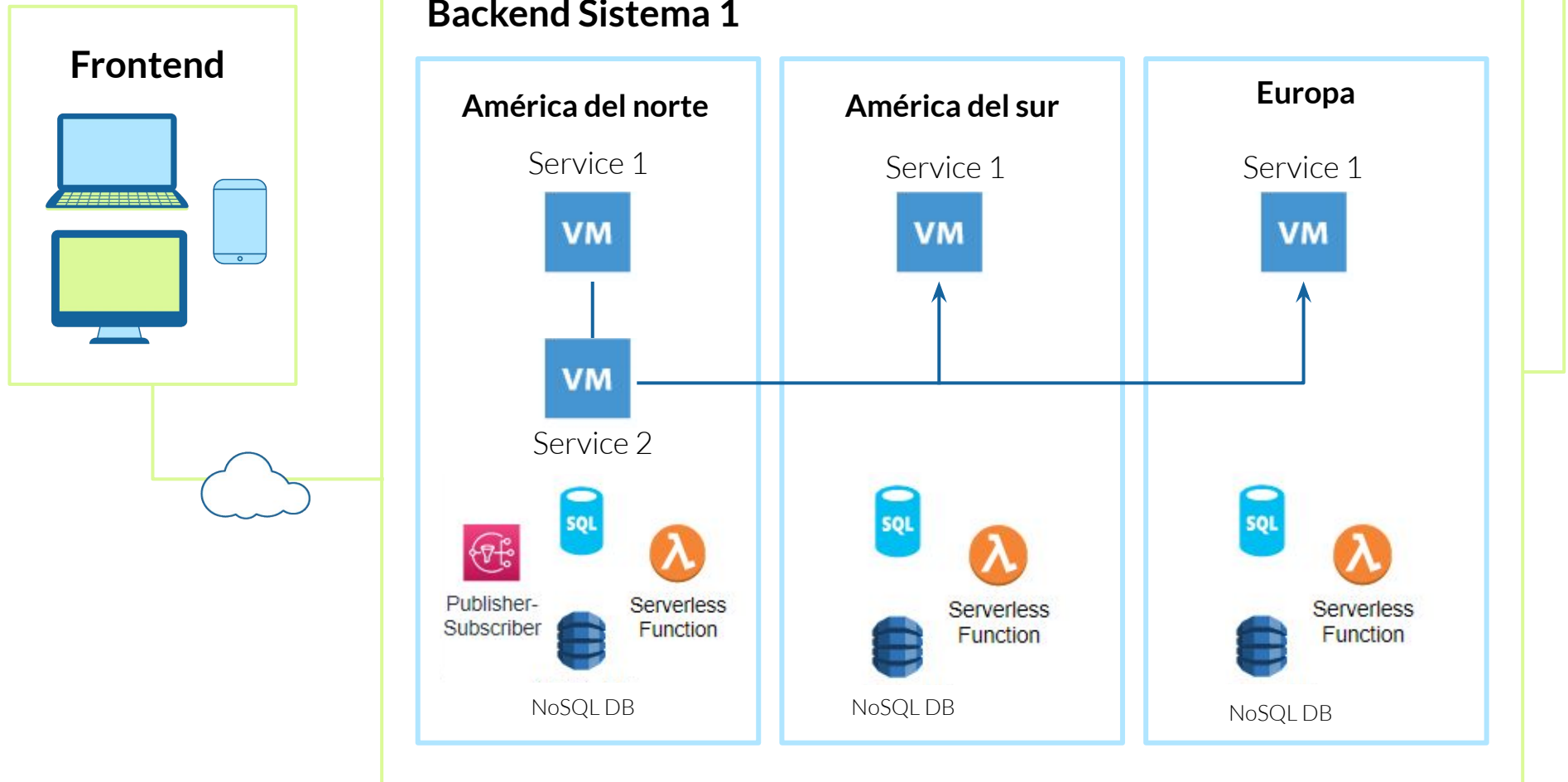
**¿Hasta dónde puede llegar o
crecer un sistema distribuido?**

”

Backend Sistema 2



Backend Sistema 1



Cómo se conecta todo en el Backend



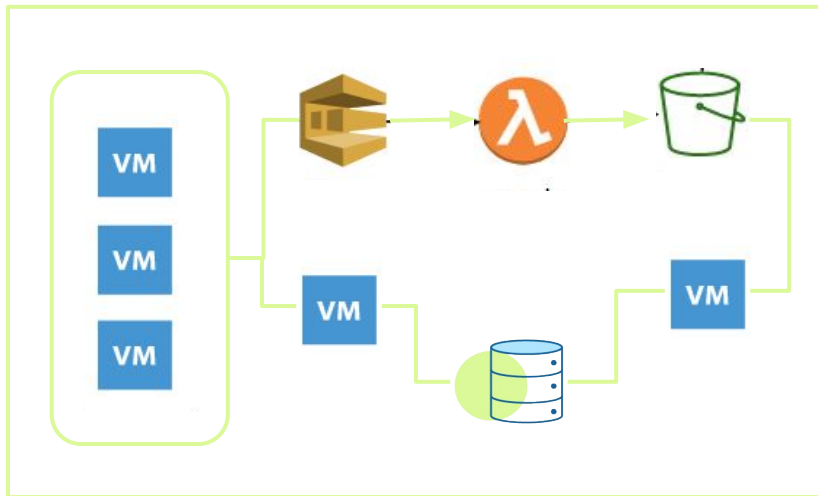
¿Cómo se conecta todo en el backend?

El backend de un sistema puede estar conformado por múltiples componentes para su funcionamiento.

Sin embargo, independientemente de la implementación, generalmente debe exponer una **API (Interfaz de Programación de Aplicaciones)**.

Comprendiendo la API

Cómo está implementado el sistema



Lo que hace el sistema

- Registrar usuario (POST)
- Obtener Usuario (GET)
- Actualizar Usuario (PUT/PATCH)
- Comprar Producto (POST)
- Reembolsar Producto (POST)

Recap y objetivo del curso





Recordemos

- Diferencias entre Frontend y Backend
- Sistemas Monolíticos
- Sistemas Distribuidos
- Elementos que encontramos en estos sistemas
- Qué es un API



Proyecto del curso

Dados ciertos requerimientos de negocio, diseñar e implementar un sistema sobre el que ejecutaremos una suite de tests confirmando el correcto funcionamiento por medio de su API.

Referencias

→ Artículos:

[Pattern: Monolithic Architecture](#)

(StackOverflow)[what-is-a-lamp-stack](#)

→ Wikipedia

[Distributed Computing](#)

Arquitectura y planeación

Definición de requerimientos de negocio



¿Qué es un
documento de diseño?



Elementos de un documento de diseño

Puede incluir pero no se limita a:

- Objetivo (problema a resolver)
- Alcance y expectativas
- Solución propuesta y alternativa
- Casos de Uso a soportar
- Componentes de la Arquitectura
- Límites (Escalabilidad, trade-offs)
- Costos

Elaboración de un documento de diseño



Revisión de documento de diseño



Elaboración de la arquitectura del sistema



Plan de integración continua



Diseño de bajo nivel, planes de prueba e integración continua



Diseño de bajo nivel, planes de prueba e integración continua



Definiendo el “Code Complete”



Recapitulación del módulo y consejos





Desarrollo e implementación

Práctica: Diseño de una “Entidad”



Práctica: Implementación de una “Entidad”



Práctica: Usando “TDD”



Pasos para completar el proyecto



Challenge: Servicio de Lectura



Challenge: Servicio de Autenticación



Challenge:
Escalabilidad
Throttling y
RetryPolicies

Pasos para completar el proyecto





Conclusiones