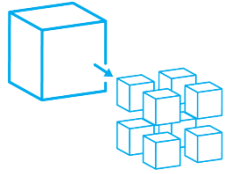


Microservicios – Arquitectura y Desarrollo

Por: Carlos Carreño

ccarrenovi@gmail.com

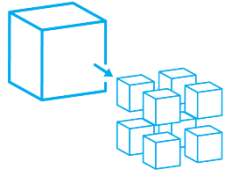
Noviembre, 2020



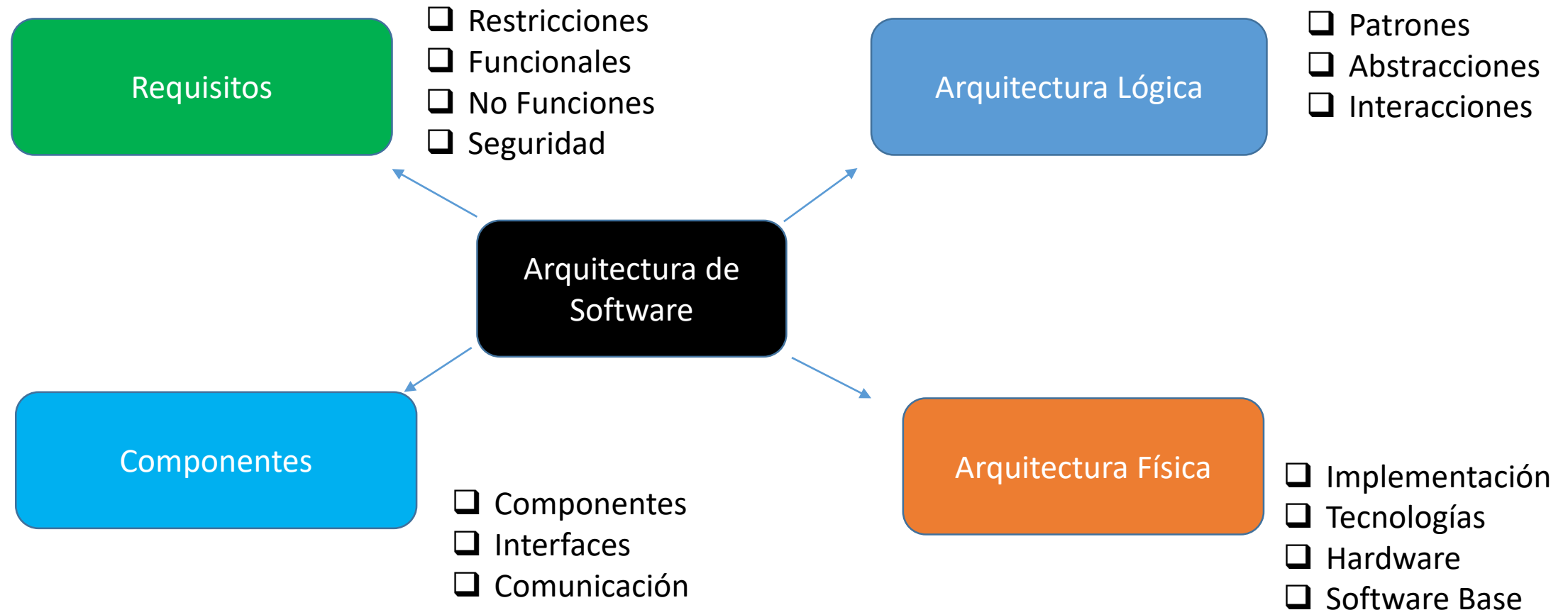
Modulo 1 Introducción a los Microservicios

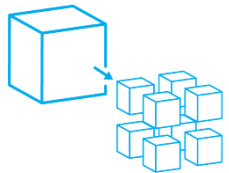
- Revisión de estilos de Arquitectura
- Arquitectura basada en Microservicios
- Características de los Microservicios
- Beneficios y limitaciones
- LAB1. Ejemplo descomponiendo una aplicación Monolítica en Microservicios

Revisión de estilos de Arquitectura: Introducción



- Que es la arquitectura de software?

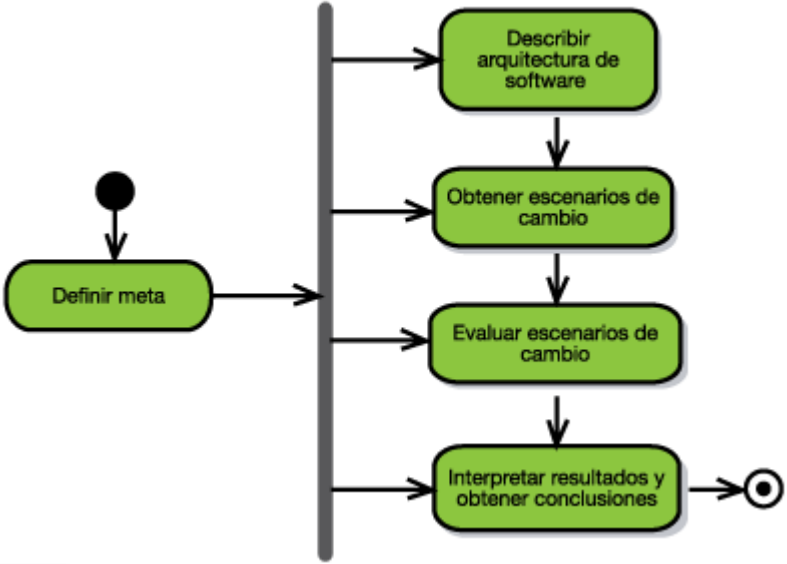


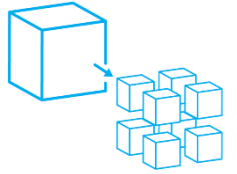


Revisión de estilos de Arquitectura

- Un *estilo de arquitectura* es una familia de arquitecturas que comparten determinadas características

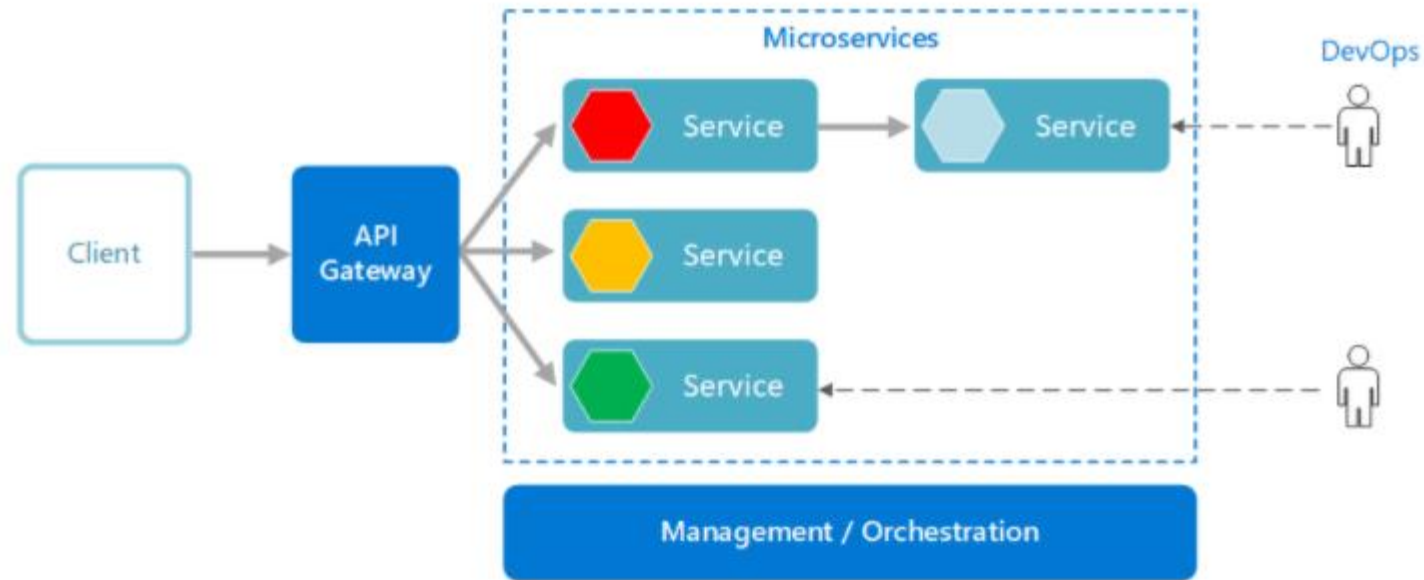
Estilo de arquitectura	Administración de dependencias	Tipo de dominio
N niveles	Niveles horizontales divididos por subred	Dominio empresarial tradicional. La frecuencia de las actualizaciones es baja.
Web-Cola-Trabajo	Trabajos de front-end y back-end, desacoplados mediante mensajería asíncrona.	Dominio relativamente sencillo con algunas tareas de uso intensivo de recursos.
Microservicios	Servicios descompuestos verticalmente (funcionalmente) que se llaman mutuamente mediante API.	Dominio complicado. Actualizaciones frecuentes.
Arquitectura basada en eventos.	Productores y consumidores. Vista independiente por cada subsistema.	IoT y sistemas en tiempo real
Macrodatos	Divide un conjunto de datos grande en fragmentos pequeños. Procesamiento en paralelo en los conjuntos de datos locales.	Análisis de datos por lotes y en tiempo real. Análisis predictivo mediante Machine Learning.
Big Compute	Asignación de datos a miles de núcleos.	Dominios con procesos intensivos como simulaciones.



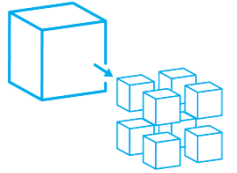


Arquitectura basada en Microservicios

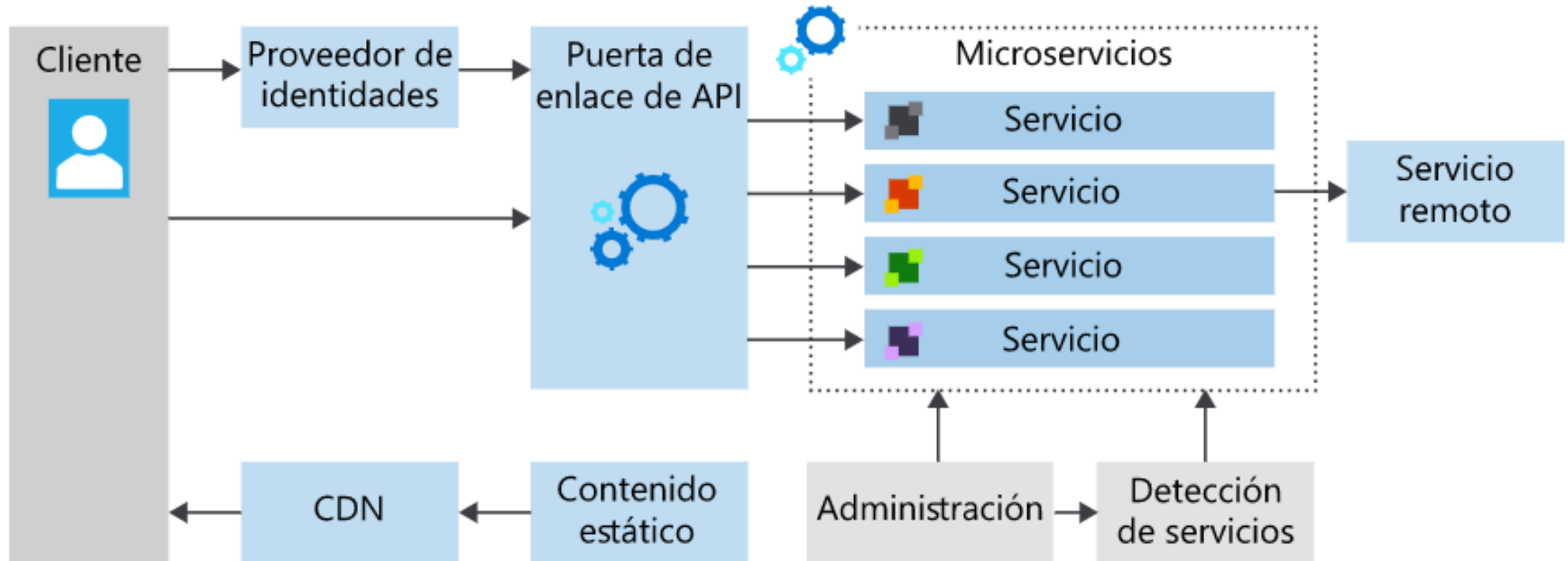
- Una arquitectura de microservicios consta de una **colección de servicios autónomos y pequeños**. Los servicios son independientes entre sí y cada uno debe implementar una funcionalidad de negocio individual.



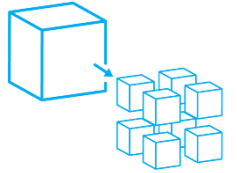
Arquitectura de microservicios: Servicios Adicionales



- Register, Api Management, Configuration

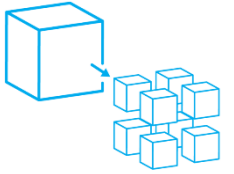


Características de los Microservicios



- **Los microservicios son pequeños e independientes**, y están acoplados de forma imprecisa. Un único equipo reducido de programadores puede escribir y mantener un servicio.
- **Cada servicio es un código base independiente**, que puede administrarse por un equipo de desarrollo pequeño.
- **Los servicios pueden implementarse de manera independiente**. Un equipo puede actualizar un servicio existente sin tener que volver a generar e implementar toda la aplicación.
- **Los servicios son los responsables de conservar sus propios datos o estado externo**. Esto difiere del modelo tradicional, donde una capa de datos independiente controla la persistencia de los datos.
- **Los servicios se comunican entre sí mediante API bien definidas**. Los detalles de la implementación interna de cada servicio se ocultan frente a otros servicios.
- **No es necesario que los servicios compartan la misma pila de tecnología**, las bibliotecas o los marcos de trabajo.

Beneficios y limitaciones

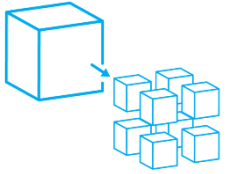


- **Beneficios**

- ☐ Agilidad
- ☐ Equipos pequeños
- ☐ Base de código pequeña
- ☐ Mezcla de Tecnologías
- ☐ Aislamiento de Errores
- ☐ Escalabilidad
- ☐ Aislamiento de los datos



Beneficios y limitaciones

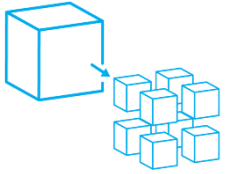


- **Limitaciones**

- ☐ Complejidad
- ☐ Desarrollo y pruebas
- ☐ Falta de gobernanza
- ☐ Congestionamiento y latencia
- ☐ Integridad de datos
- ☐ Administración
- ☐ Control de versiones
- ☐ Conjunto de habilidades

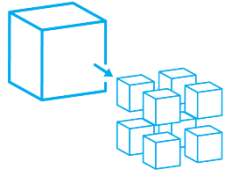


Aplicaciones Monolíticas

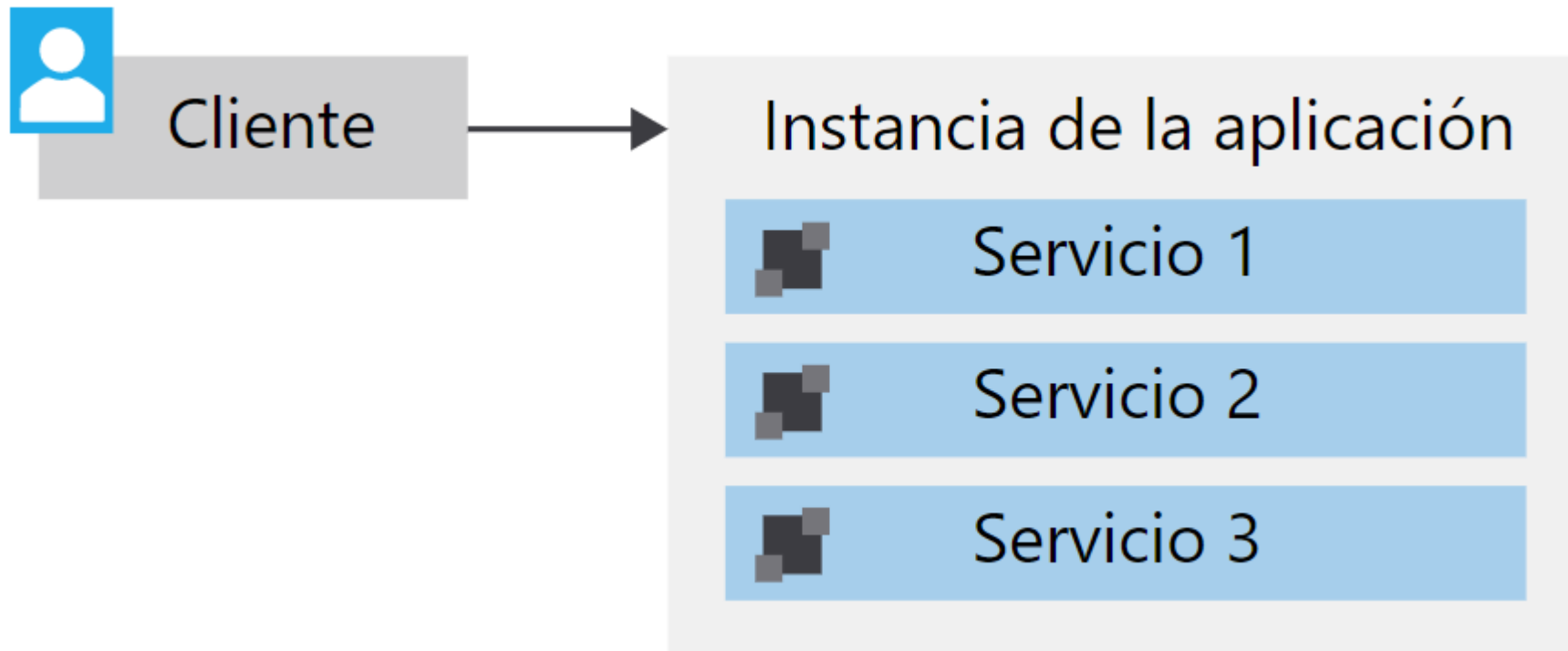


- Una arquitectura monolítica es aquella en la que todos los componentes de una aplicación se colocan en una sola unidad.
- La aplicación monolítica está restringida dentro de una única instancia en tiempo de ejecución de la aplicación.
- Las aplicaciones tradicionales suelen constar de una interfaz web, una capa de servicios y una capa de datos.
- En una arquitectura monolítica, estas capas se combinan en una instancia de la aplicación

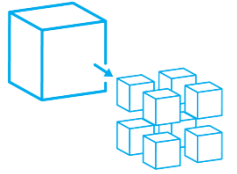
Arquitectura monolítica



- Las arquitecturas monolíticas suelen ser soluciones adecuadas para **aplicaciones pequeñas**, pero pueden resultar difíciles de manejar a medida que crece la aplicación.



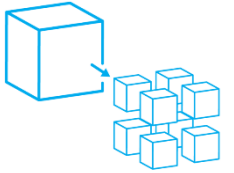
Limitaciones de las Aplicaciones Monolíticas



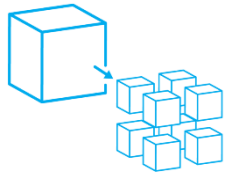
- Dificultad para escalar los servicios de forma independiente.
- Dificultad para desarrollar y administrar implementaciones a medida que aumenta el código base, lo que reduce las versiones y la implementación de características nuevas.
- Se vincula a una sola pila de tecnología, lo que limita la innovación en nuevas plataformas y SDK.
- Las actualizaciones del esquema de datos pueden ser cada vez más difíciles.



Laboratorio



- LAB 01-1 Descomponiendo una aplicación Monolítica en Microservicios
- LAB 01-2 Instalación de Herramientas
- LAB 02 Creando el Proyecto de Microservicios



Referencias

- https://cgrw01.cgr.go.cr/rup/RUP.es/SmallProjects/core.base_rup/guidances/concepts/software_architecture_4269A354.html
- https://es.wikipedia.org/wiki/Arquitectura_de_software
- <https://docs.microsoft.com/es-es/azure/architecture/guide/architecture-styles/>
- <https://docs.oracle.com/es/solutions/learn-architecture-microservice/design-microservices-based-application1.html#GUID-EAF2A1AD-2CA6-4034-B579-DE68632C7D38>
- <https://docs.microsoft.com/es-es/learn/modules/microservices-architecture/>