**Microservicios: teoría y aplicación**

Como lo expresa en el vídeo el termino de Micro-servicios se entiende como una nueva formar particular de diseñar software como un conjunto de servicios de implementación independiente. Con ello, reemplazar los sistemas monolíticos, estos están construidos como una sola unidad, un único ejecutable lógico y para realizar un cambio en una pequeña parte de la aplicación requiere que se reconstruya y despliegue todo el monolito. Con el tiempo, a menudo es difícil mantener una buena estructura modular, lo que hace más difícil mantener los cambios que solo deberían afectar a un módulo dentro de ese módulo. Por este motivo, surgen los microservicios, para darle una solución eficaz a los sistemas monolíticos, ya que permite la escalabilidad.

Podemos encontrar una serie de razones para utilizar microserviciso; por ejemplo, permite implementarlo de forma independiente, descomponer las funciones, a largo plazo generara beneficios porque permite la escalabilidad y permite una mejor interacción con el usuario. Asimismo, Pueden ser descompuesto en diferentes partes independientes, por ello cada uno de los servicios puede ser desplegado y modificado sin afectar a otros aspectos funcionales de la aplicación.

**Ventajas:**

* Escalabilidad
* Equipos de trabajo reducidos
* Funcionalidad modular, módulos independientes.
* Libertad del desarrollador de desarrollar y desplegar servicios de forma independiente
* Uso de contenedores permitiendo el despliegue y el desarrollo de la aplicación rápidamente

**Desventajas:**

* Alto consumo de memoria
* Se necesita una buena interfaz de trabajo
* Necesidad de tiempo para poder fragmentar distintos microservicios
* Complejidad de gestión de un gran número de servicios
* Pruebas o testeos complicados al despliegue distribuido

**Tecnologías:**

* Event Sourcing

Garantizar que cada cambio en el estado de una aplicación se capture en un objeto de evento, y que estos objetos de evento se almacenen en la secuencia que se aplicaron durante el mismo tiempo de vida que el estado de la aplicación.

* Domain Driven Design

trata con modelos grandes dividiéndolos en diferentes contextos delimitados y siendo explícitos sobre sus interrelaciones.

* API Gateway

Es un servicio de AWS que permite a los desarrolladores crear, publicar, mantener, supervisar y asegurar API a cualquier escala.