Portafolio - Trabajo #1

**Tema**:

-Arquitectura de software basada en microservicios

**Programa**: Programador(a) de Aplicaciones Informáticas

**Módulo**: Programación Orientada a Objetos

**Código**: CSTI0208

**Personas participantes:**

|  |
| --- |
| **Alejandro Quintero – Alejandra Villalta** |

|  |  |
| --- | --- |
| Fecha de asignación del trabajo: | Martes 7 de diciembre 2021 |
| Fecha límite de entrega: | Jueves 9 de diciembre 2021 |
| Medio de entrega: | Microsoft TEAMS |

**Introducción**

La arquitectura de software es un concepto fácil de entender y que la mayoría de los ingenieros aprecian intuitivamente, sobre todo los que tienen un poco de experiencia, pero resulta difícil definirlo con precisión. En concreto, es difícil dibujar una línea precisa entre el diseño y la arquitectura, la arquitectura es un aspecto de diseño que se concentra en algunas características específicas.

**Desarrollo**

**¿Qué es una arquitectura de microservicios?**

-La arquitectura de microservicios es un método de desarrollo de aplicaciones software que funciona como un conjunto de pequeños servicios que se ejecutan de manera independiente y autónoma, proporcionando una funcionalidad de negocio completa. En ella, cada microservicio es un código que puede estar en un lenguaje de programación diferente, y que desempeña una función específica. Los microservicios se comunican entre sí a través de APIs, y cuentan con sistemas de almacenamiento propios, lo que evita la sobrecarga y caída de la aplicación.

-Los microservicios han creado infraestructuras IT más adaptables y flexibles. Porque si se quiere modificar solamente un servicio, no es necesario alterar el resto de la infraestructura. Cada uno de los servicios se puede desplegar y modificar sin que ello afecte a otros servicios o aspectos funcionales de la aplicación.

-Veamos qué ventajas y desventajas tiene la aplicación de una arquitectura de microservicios frente a otros tipos de arquitectura. Hay que tener en cuenta estos puntos a la hora de identificar qué tipo de arquitectura software será la mejor para un determinado proyecto u organización.

## ****Ventajas****

* **Modularidad:** al tratarse de servicios autónomos, se pueden desarrollar y desplegar de forma independiente. Además un error en un servicio no debería afectar la capacidad de otros servicios para seguir trabajando según lo previsto.
* **Escalabilidad:** como es una aplicación modular, se puede escalar horizontalmente cada parte según sea necesario, aumentando el escalado de los módulos que tengan un procesamiento más intensivo.
* **Versatilidad:** se pueden usar diferentes tecnologías y lenguajes de programación. Lo que permite adaptar cada funcionalidad a la tecnología más adecuada y rentable.
* **Rapidez de actuación:** el reducido tamaño de los microservicios permite un desarrollo menos costoso, así como el uso de “contenedores de software” permite que el despliegue de la aplicación se pueda llevar a cabo rápidamente.
* **Mantenimiento simple y barato:** al poder hacerse mejoras de un solo módulo y no tener que intervenir en toda la estructura, el mantenimiento es más sencillo y barato que en otras arquitecturas.
* **Agilidad:** se pueden utilizar funcionalidades típicas (autenticación, trazabilidad, etc.) que ya han sido desarrolladas por terceros, no hace falta que el desarrollador las cree de nuevo.

## ****Desventajas****

* **Alto consumo de memoria:** al tener cada microservicio sus propios recursos y bases de datos, consumen más memoria y CPU.
* **Inversión de tiempo inicial:** al crear la arquitectura, se necesita más tiempo para poder fragmentar los distintos microservicios e implementar la comunicación entre ellos.
* **Complejidad en la gestión:** si contamos con un gran número de microservicios, será más complicado controlar la gestión e integración de los mismos. Es necesario disponer de una centralización de trazas y herramientas avanzadas de procesamiento de información que permitan tener una visión general de todos los microservicios y orquesten el sistema.
* **Perfil de desarrollador:**los microservicios requieren desarrolladores experimentados con un nivel muy alto de experiencia y un control exhaustivo de las versiones. Además de conocimiento sobre solución de problemas como latencia en la red o balanceo de cargas.
* **No uniformidad:** aunque disponer de un equipo tecnológico diferente para cada uno de los servicios tiene sus ventajas, si no se gestiona correctamente, conducirá a un diseño y arquitectura de aplicación poco uniforme.
* **Dificultad en la realización de pruebas:** debido a que los componentes de la aplicación están distribuidos, las pruebas y test globales son más complicados de realizar.
* **Coste de implantación alto:** una arquitectura de microservicios puede suponer un alto coste de implantación debido a costes de infraestructura y pruebas distribuidas.

**Hay varios puntos que tenemos que tener en cuenta**:

* No todos los servicios son microservicios. Al comenzar a trabajar en microservicios debes asegurarte que estén adecuadamente desacoplados. Si están fuertemente acoplados no será posible escalarlos de manera individual.
* No empieces con muchos servicios al mismo tiempo. Empieza poco a poco, con las funcionalidades clave y hazlo evolucionar.
* Conforme se vaya evolucionando el ecosistema de aplicaciones, comprende dónde se encuentra el proyecto en la actualidad y qué objetivos se están intentando lograr. Diseña aplicaciones para que los equipos pequeños puedan tener un control total del servicio o que la aplicación sea totalmente independiente de todo lo demás.