### Arquitectura de software basada en microservicios

**Área:** Ingenieria de Software

**Tipo:** Curso de Ampliación

**Unidades:** 5 Unidades Crédito

**Horas académicas:** 70

**Código:** --------

**Fundamentación:**

El diseño de software a gran escala para la empresa privada y plataformas de internet, requiere de técnicas para manejar la complejidad del diseño, implementación, mantenimiento, escalabilidad y operación de soluciones de misión critica. El termino **Microservicio** se uso inicialmente en un taller de arquitectos de software alrededor de Mayo de 2011 para describir una seria de estilos y patrones que ellos habían estado explorando [[[1]](#footnote-1)]. Desde entonces este estilo de arquitectura ha sido adoptado ampliamente por las compañías pioneras en tecnología tales como Netflix, Amazon y muchos otros, convirtiéndose en una practica común a la hora de diseñar e implementar soluciones a gran escala.

Los Microservicios cuando son implementados apropiadamente y por las razones y en el contexto correctos, ofrecen la posibilidad de agilizar y simplificar el desarrollo de aplicaciones complejas, permitiendo dividirlas en módulos que usualmente son mas fáciles de entender, diseñar, implementar, mantener y escalar acorde a la naturaleza de cada modulo. Al mismo tiempo, los Microservicios introducen sus complicaciones inherentes, y requieren de una cultura de disciplina y automatización que debe ser tomada en cuenta al momento de adoptar este estilo.

El conocimiento de cómo seleccionar, diseñar, implementar y operar una solución basada en Microservicios es de suma relevancia en campo laboral actual, y equipa al profesional con una herramienta muy bien cotizada. Al mismo tiempo, permite evitar los errores comunes al utilizar este estilo de arquitectura. Por otra parte, aun cuando se decida no utilizar el mismo, su entendimiento sirve como un punto de referencia para otros estilos y permite tomar mejores decisiones de diseño en proyectos de gran escala.

**Objetivos:**

Al finalizar el curso el participante debe ser capaz de:

* Entender que es la arquitectura de software y principales estilos
* Entender arquitecturas basadas en Microservicios y sus implicaciones
* Entender los principios de diseño de arquitecturas basadas en Microservicios
* Diseñar e implementar Microservicios de acuerdo a las practicas recomendadas de la industria
* Entender como operar Microservicios en producción

**Perfil del estudiante:**

El curso está dirigido a profesionales con conocimientos básicos/intermedios en programación y diseño de software orientado a objetos así como estudiantes de postgrado en las áreas de ciencias de la computación.

**Contenido Temático:**

* **Tema 1: Introducción a la Arquitectura de Software**. En qué consiste la Arquitectura de Software. Propiedades deseadas en una Arquitectura de Software. Documentando y comunicando Arquitectura de Software. Rol del Arquitecto de Software.
* **Tema 2: Estilos de Arquitectura de Software**. Monolítico. Monolítico modulares. Multi-capas. Micro-núcleo. Client/Server. Arquitectura Orientada a Servicios (SOA). Microservicios. Basada en eventos.
* **Tema 3: Principios generales de diseño**. Asignación de Responsabilidades. Cohesion y Acoplamiento. Modularidad y Abstracción. Ambito de la información. Interfases y APIs. Segregación por funcionalidad. Generalización y especialización.
* **Tema 4: Arquitecturas basadas en Microservicios.** Beneficios y retos con los Microservicios. Cuándo y cuándo no usar Microservicios. Granularidad de Microservicios. Almacenamiento de datos en Microservicios. Diseño de APIs en Microservicios. Integración de Microservicios. Plantilla de servicios.
* **Tema 5: Diseñando un Microservicio - Caso de estudio**. Capas y patrones comunes (MVC, Repositorio, Service Delegate, Adapters). Diseñando los APIs y controladores. Diseñando eventos. Diseñando capa de datos. Diseñando la capa de reglas del negocio.
* **Tema 6: Integración de Microservicios**. Punto a punto. Callback & Polling. PubSub, Eventos. Coreografía. Orquestación. Sagas.
* **Tema 7: Escalabilidad, Robustez y seguridad de Microservicios**. Ejes de particionamiento de servicios. Servicios con estados y sin estados. Uso de memorias cache. Fallas parciales. Falacias de sistemas distribuidos. Patrones: reintentos, Circuit breaker, Blukheads, exponential back off. Autenticación y autorización.
* **Tema 8: Operando Microservicios**. . Configuración. Pruebas. Instalación (Deployment). CI/CD. Monitoreo y Alertas. Trazas distribuidas (Distributed tracing).

**Evaluación**

* 3 quices 10% c/u
* 3 talleres 10% c/u
* 1 proyecto 40%

**Observaciones finales**:

* El curso tiene una duración de 70hrs.
* La modalidad es **en** **linea**.
* Se utilizaran herramientas de comunicación como Google Meet y el Laboratorio del Postgrado en Ciencias de la Computación.

|  |  |  |
| --- | --- | --- |
| **Docente** | **Coordinador** | **Personal Administrativo** |
| MSc. Fermin Ordaz | MSc. Fermin Ordaz | Rosiris Morales |

**Bibliografia**

* Mark Richards & Neal Ford. Fundamentals of Software Architecture, O’Reilly 2020
* Chris Richardson. Microservices Patterns. Manning 2019.
* Robert C. Martin. Clean Architecture. Prentice Hall 2018.
* George Fairbanks. Just Enough Software Architecture 2010.
* Martin Abbott & Michael Fisher, Scalability Rules, Addison-Wesley 2017.
* Michael Nygard, Release it! Design and Deploy Production-Ready Software 2007.
* Richard Rodger, The Tao of Microservices Manning 2018.
* Craig Larman, Applying UML and Patterns Pearson 2016.
* Eric Evans, Domain-Driven Design, Addison-Wesley 2004.
1. https://en.wikipedia.org/wiki/Microservices [↑](#footnote-ref-1)