

Elaboro	Documento	Versión	Descripción	Fecha
Luis Sánchez Martínez	Introducción al diseño de APIs	1.0	Creación de documento	26/02/2021
Luis Sánchez Martínez	Introducción al diseño de APIs	1.1	Se ajusta información para desarrolladores	10/01/2022
Luis Sánchez Martínez	Introducción al diseño de APIs	1.2	Se actualiza documento	12/03/2024

Introducción al diseño de APIs

El alto costo de integración de soluciones en el desarrollo del código implica no solo temas de dinero sino de tiempo y energía. Recapitulando podríamos decir que pasar de las EAI (Integración empresarial de aplicaciones) a las APIs en los últimos años han seguido el siguiente camino.

Finales de los 90's: servicios y modelos de EAI

- Sin protocolos abiertos proliferaron las plataformas propietarias y las interfaces.
- La mayoría de las plataformas eran centros de integración con el uso de un “bus” para unir aplicaciones.
- La EAI abordó los desafíos transaccionales y de integración temprana mediante el uso de middleware orientado a mensajes.

Mediados de la década de 2000: SOA y mejores prácticas

- SOA admite protocolos abiertos como SOAP y WSDL para facilitar la integración y las plataformas EAI comenzaron a adoptarlos rápidamente.
- Los registros estandarizados como UDDI automatizan el descubrimiento dinámico de endpoints de las aplicaciones para consumers y publishers.
- Los desafíos económicos retrasan la adopción de protocolos abiertos críticos para la seguridad, transaccionalidad y estabilidad de SOA.
- Las plataformas EAI comienzan a soportar gran parte de los estándares soportados por SOA.
- Los proveedores más grandes como Oracle (Fusión) y SAP (PI) admiten a los proveedores de EAI más pequeños para que puedan soportar la gestión de procesos comerciales.

Principios de la década del 2010: plataformas de API, SDK y API management

- REST y JSON se convierten en el modelo de programación para aplicaciones móviles y el estándar para consumir datos de back end.
- Los nuevos protocolos como OAuth admiten funciones de seguridad para aplicaciones nativas y basadas en HTTP.
- Comparada con la arquitectura SOA esta plataforma demuestra ser liviana y ágil mientras mantiene un conjunto de funciones similares a los estándares SOAP/XML.
- Las tres generaciones de estas tecnologías coexisten hoy en una arquitectura integral.

2015 hasta la actualidad: arquitectura de microservicios, resiliencia, API y seguridad

- El mundo se está moviendo a la siguiente generación de HTTP, forjando estrategias para crear pequeños contenedores y multiplataforma que alojan microservicios independientes.
- Las tecnologías como Node y Spring funcionan bien en este modelo y Docker brinda aislamiento para diferentes servicios.
- Otros proyectos como Hystrix para arquitecturas resilientes y JWT para la gestión de seguridad, están completando otras piezas del rompecabezas.
- HTTP/2 impulsará la economía de APIs al proporcionar cargas útiles de llamadas múltiples en una sola conexión, un movimiento importante para las tecnologías REST.

Referencias

<https://www.photon.com/what-we-do/api-and-microservices/progression-of-APIs>

<https://www.akana.com/blog/brief-history-integration-technologies-eai-api-gateways>

<https://hackernoon.com/es/los-conceptos-basicos-del-diseno-de-una-arquitectura-api>