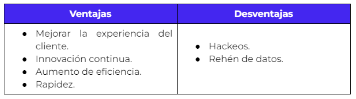
EJB STATEFUL Y EXPOSICION COMO SERVICIO

ECONOMIA DE LAS APIS

Se trata de un sistema que permite ofrecer productos y servicios así como una mejor experiencia del cliente a través de APIs. Su objetivo es facilitar la creación de soluciones que contribuyan a los objetivos de un negocio para mejorar el rendimiento de sus recursos humanos y técnicos.

En otras palabras, la economía de las APIs permite a las empresas transformarse rápidamente en una plataforma digital con diversas funcionalidades para aumentar sus resultados.

Y es que, gracias a las APIs, podemos obtener datos de terceros de forma rápida y accesible, esto permite que muchas empresas cambien sus modelos de negocio y agilicen sus procesos.



COMO IMPLEMENTAR UNA ECONOMIA DE APIS SOLIDA

1. Toma de conciencia del valor de las APIs para el negocio. Es importante tener en mente la magnitud y valor que ofrecen las APIs para un negocio. Debemos entender a una API como un instrumento para aportar valor de investigación, desarrollo e innovación en terceros.
2. Definición de los objetivos dentro del mercado. Antes de comenzar a transformar lo que ya se tiene es importante definir tres puntos esenciales: qué, por qué y para qué se quiere una API.
3. Arquitectura de servicios que garanticen la escabilidad. Si se crece rápido, el viralizar un negocio en el exterior con millones de clientes puede traer un problema para la infraestructura de IT. Es necesario controlar cada uno de los detalles.
4. APIs como un acelerador de compromiso. Las APIs son el pegamento de toda la comunidad de desarrolladores. Facilitar las interfaces de desarrollo de aplicaciones en una plataforma online sencilla, donde se centralice todo el uso por parte de partners, proveedores y desarrolladores, es un acelerador del compromiso con el negocio y la marca de una compañía.
5. Gestión de las APIs. Es necesario llevar un control para monitorizar su rendimiento para responder a la demanda y problemas derivados del uso de la API. Algunas herramientas que ayudan a realizar esto son: 3scale, ApiAxle, Oracle SOA, IBM API Management, etc. El utilizar estas herramientas simplifica todo lo necesario para tener el control como documentación, analítica, despliegue, entorno de pruebas, seguridad, monetización, disponibilidad, soporte, etc.

MICROSERVICIOS

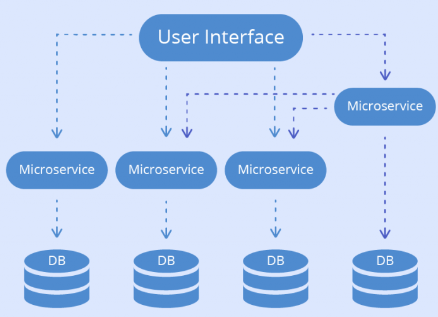
Un servicio (service) es un componente de una aplicación que puede realizar operaciones de larga ejecución en segundo plano y que no proporciona una interfaz de usuario. Un servicio puede ejecutarse en segundo plano, incluso cuando el usuario no está interactuando con la aplicación.

Existen 3 tipos de servicios:

1. Primer plano. Realizan una operación que el usuario puede notar y deben mostrar una notificación. Estos servicios continúan ejecutándose incluso si el usuario deja de interactuar con la aplicación.
2. Segundo plano. Realiza una operación que el usuario no nota directamente.
3. Enlace. Ofrece una interfaz cliente-servidor que permite a los componentes interactuar con el servicio, enviar solicitudes y recibir resultados. Este tipo de servicio se ejecuta solo mientras otro componente de aplicación está enlazado a él. Se pueden enlazar varios componentes con el servicio a la vez, pero el servicio se destruye cuando todos ellos se desenlazan.

QUE ES UN MICROSERVICIO

Según AWS, un microservicio es un tipo de arquitectura que sirve para desarrollar software donde el software está compuesto por pequeños servicios independientes que se comunican a través de API bien definidas. Con los microservicios, las aplicaciones se dividen en elementos más pequeños e independientes entre sí.



La arquitectura de microservicios hace que las aplicaciones sean más fáciles de escalar y más rápidas de desarrollar. Esto permite la innovación y acelerar el tiempo de comercialización de las nuevas características.

BENEFICIOS

* Agilidad. Con los microservicios los equipos actúan en un contexto pequeño y bien comprendido, y están facultados para trabajar de formas más independiente y más rápida. Esto acorta los tiempos del ciclo de desarrollo.
* Escalado flexible. Los microservicios permiten que cada servicio se escale de forma independiente para satisfacer la demanda de las características de la aplicación que respalda. Esto permite a los equipos adecuarse a las necesidades de la infraestructura.
* Implementación sencilla. Permiten la integración y la entrega continuas, lo que facilita probar nuevas ideas y revertirlas si algo no funciona. Este bajo costo de los errores permite experimentar trayendo como consecuencia la facilidad para actualizar el código y la aceleración del tiempo de comercialización de las nuevas características.
* Libertad tecnológica. Puedes elegir que herramienta es la mejor para resolver los problemas específicos ya que esta arquitectura de microservicios no sigue un enfoque de diseño único.
* Código reutilizable. La división del software en pequeños módulos y bien definidos permite a los equipos usar funciones para diferentes propósitos. Un servicio escrito para una función específica se puede usar como un componente básico para otra característica.
* Resistencia. La independencia entre servicios aumenta la resistencia de una aplicación a los errores. Si hay un error en todo el servicio, las aplicaciones lo manejan degradando la funcionalidad sin bloquear toda la aplicación.