



CICLO 4a

[FORMACIÓN POR CICLOS]

Desarrollo de **APLICACIONES WEB**

Glosario Microservicios



**UNIVERSIDAD
DE ANTIOQUIA**

Facultad de Ingeniería

Microservicios

- **Monolito.** O arquitectura monolítica, es la forma de programar altamente acoplada, que supone toda la aplicación como un único servicio.
- **REST.** Es una arquitectura estándar para construir y comunicarse con servicios web.
- **Node.JS.** Es un entorno en tipo de ejecución multiplataforma que permite crear aplicaciones con JavaScript.
- **Express.** Es una librería de HTTP para node.JS que permite configurar el servidor y generar un desarrollo más amigable.
- **JSON.** Es una notación de objetos usada en JavaScript que permite un formato ligero de intercambio de datos, que se encuentra construido en pares de tipo nombre/valor.

Microservicios

- **WebSocket.** Es un protocolo de red basado en TCP que establece cómo deben intercambiarse datos entre redes. Puesto que es un protocolo fiable y eficiente, es utilizado por prácticamente todos los clientes. El protocolo TCP establece conexiones entre dos puntos finales de comunicación, llamados sockets. De esta manera, el intercambio de datos puede producirse en las dos direcciones.
- **HTTP (hypertext transfer protocol).** El protocolo de transferencia de hipertexto es el protocolo de comunicación que permite las transferencias de información a través de archivos (XHML, HTML ...) en la World Wide Web.
- **API (application programming interface).** La interfaz de programación de aplicaciones es un conjunto de subrutinas, funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizada por otro *software* como una capa de abstracción.

Microservicios

- **Proxy.** Es un servidor —programa o dispositivo— que hace de intermediario en las peticiones de recursos que realiza un cliente (A) a otro servidor (C). Por ejemplo, si una hipotética máquina A solicita un recurso a C, lo hará mediante una petición a B, que a su vez trasladará la petición a C; de esta forma, C no sabrá que la petición procedió originalmente de A. Esta situación estratégica de punto intermedio le permite ofrecer diversas funcionalidades: control de acceso, registro del tráfico, restricción a determinados tipos de tráfico, mejora de rendimiento, anonimato de la comunicación, caché web, etc.
- **Circuit Breaker.** Es un patrón de diseño que ayuda a prevenir que una falla de red o de servicio se produzca en cascada a otros servicios.
- **EndPoint (punto final).** Es un dispositivo informático remoto que se comunica con una red a la que está conectado.