|  |  |
| --- | --- |
| **Uso de las API’s dentro de un sistema** | |
| **Ventajas** | **Desventajas** |
| * Permite expandir el horizonte y alcance de nuestra aplicación por medio de las herramientas de terceros. * Ahorra tiempo y recursos para no tener que crear tecnologías ya implementadas por otras compañías. * Elimina el papeleo innecesario debido a que nos proporciona información de forma más rápida y sencilla. * Existen API’s públicos que podemos utilizar sin la necesidad de recibir autorización de quien lo creó. * Aumenta la seguridad y protección de nuestro sistema en casos específicos dependiendo de la API que se esté utilizando. * En la mayoría de los casos de usa un estándar de lenguaje. | * En ciertos casos requerimos la verificación de la empresa que se desea obtener la API (por ejemplo, Twitter Developer) lo cual nos puede llevar tiempo. * En algunos casos es más difícil utilizar los comandos de esta debido a la mala documentación de quien lo creó. * En ciertas ocasiones la curva de adaptación requiere tiempo para poder implementar las líneas de comando para el uso de estas. * Para ciertas ocasiones no es posible usar con libertad todas las API’s debido a ciertas prohibiciones (por ejemplo, los cobros por internet sin la previa autorización de los organismos oficiales). |

Interfaz: capa de abstracción permite la comunicación entre dos sistemas de una forma sencilla sin la necesidad de conocer lo que está pasando detrás a detalle.

Application Programming Interface (API): es una interfaz para que se comuniquen dos distintas aplicaciones o programas de software, además de compartir datos entre ellos.

Arquitectura de Software: Forma en la que está estructurado nuestro sistema y las funciones que cumple.

Servicio web: Sistema que permite la comunicación entre dos equipos que se encuentren en la misma red (con HTTP) y es la base de comunicación de API’s remotas.

REST: Arquitectura que administra los permisos para definir que aplicaciones pueden modificar y consumirse desde otras.

XML: Formato tradicional para enviar datos

JSON: Formato más usado para transferir datos, incluso se llega a considerar estándar.

Token: Es una parte de información que verifica si nuestra autenticación de las API’s privadas siguen siendo válidas.

URI: Permite solicitar la información de forma específica.

Códigos de estado: 200(exitoso), 300(redirecciones), 400(solicitud inválida), 500(errores en el servidor).

Métodos HTTP: get (solicitar información), post (enviar nueva información), put (cambiar información), delete (borrar información.