Liceo Compu-Market

Nombre: Gerson Boche Peralta

Grado: 5to Bachillerato en computación con orientación en científica

Sección: “A”

Curso: Programación

Tema:

Examen del Bloque

Profesor: Erick Albert Gonzales Cruz

Fecha: 07 de agosto del 2019, Villa Nueva, Guatemala.

**API Interfaz de programación de aplicaciones**

La interfaz de programación de aplicaciones, conocida también por la sigla API, en [inglés](https://es.wikipedia.org/wiki/Idioma_ingl%C3%A9s), application programming interface, es un conjunto de [subrutinas](https://es.wikipedia.org/wiki/Subrutina), funciones y procedimientos (o [métodos](https://es.wikipedia.org/wiki/M%C3%A9todo_(inform%C3%A1tica)), en la [programación orientada a objetos](https://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos)) que ofrece cierta [biblioteca](https://es.wikipedia.org/wiki/Biblioteca_(programaci%C3%B3n)) para ser utilizado por otro [software](https://es.wikipedia.org/wiki/Software) como una [capa de abstracción](https://es.wikipedia.org/wiki/Capa_de_abstracci%C3%B3n).

Son usadas generalmente en las [bibliotecas](https://es.wikipedia.org/wiki/Biblioteca_(inform%C3%A1tica)) de programación.



Imagen gracias a: <https://www.abc.es/tecnologia/consultorio/20150216/abci--201502132105.html>

Quizás muchos usuarios hayan leído en algunos artículos la abreviatura «API». Por ejemplo, que Oracle demandó a Google por el supuesto uso de su API de Java en la creación de Android. (Oracle perdió el caso). La abreviatura viene del término en inglés Application Programming Interfaces (Interfaces de [programación](http://laguiatv.abc.es/programacion/) de aplicaciones) y según explica el científico en computación [Benjy Weinberger en su blog](http://benjyw.com/post/50031659939/10-tech-concepts-everyone-should-know), una API es una especificación formal sobre cómo un módulo de un software se comunica o interactúa con otro.

En otras palabras, las API son un conjunto de comandos, funciones y protocolos informáticos que permiten a los desarrolladores crear programas específicos para ciertos sistemas operativos. Las API simplifican en gran medida el trabajo de un creador de programas, ya que no tiene que «escribir» códigos desde cero. Estas permiten al informático usar funciones predefinidas para interactuar con el sistema operativo o con otro programa.

Desde el show HowStuffWorks señalan que le usuario nunca ve las API en pleno proceso de trabajo, pero sí detalles de sus acciones. La API es una interfaz o rostro que sólo da la cara al software. Es decir, el el usuario no ve eso. Con las API, las aplicaciones como Facebook, Twitter, y Candy Crush se pueden comunicar entre ellas sin que el usuario tenga que intervenir o incluso, percatarse.

Por ejemplo, cuando el usuario compra entradas a través de la página web de una sala de [cine](http://hoycinema.abc.es/) e introduce la información de su tarjeta de crédito, la web usa una API para enviar dicha información de forma remota a otro programa que verifica si los datos bancarios son correctos. Una vez que se confirma el pago, la aplicación remota envía la información al sitio web del [cine](http://hoycinema.abc.es/) y le da un «OK», por lo que esta página emite los tickets.

En todo ese proceso, el usuario solo ve una cara del proceso, la página del cine, pero tras bambalinas hay muchas aplicaciones que se están comunicando gracias a las API.

Los usuarios pueden «ver» muchas API en todos lados aunque no se percaten de ello. Por ejemplo, este artículo hay una columna lateral que tiene varios iconos de redes sociales, como Facebook, Twitter y Google+. Estos son enlaces que se encargan de «llamar» a las APIs asociadas a esos servicios para que el usuario pueda «tuitear» o compartir la información en Facebook sin tener que salir de la página web.

**Características**

Una API representa la capacidad de comunicación entre componentes de software. Se trata del conjunto de llamadas a ciertas bibliotecas que ofrecen acceso a ciertos servicios desde los procesos y representa un método para conseguir [abstracción](https://es.wikipedia.org/wiki/Abstracci%C3%B3n_(programaci%C3%B3n_orientada_a_objetos)) en la [programación](https://es.wikipedia.org/wiki/Programaci%C3%B3n), generalmente (aunque no necesariamente) entre los niveles o capas inferiores y los superiores del software. Uno de los principales propósitos de una API consiste en proporcionar un conjunto de [funciones](https://es.wikipedia.org/wiki/Subrutina) de uso general, por ejemplo, para dibujar [ventanas](https://es.wikipedia.org/wiki/Ventana) o [iconos](https://es.wikipedia.org/wiki/Icono_(inform%C3%A1tica)) en la [pantalla](https://es.wikipedia.org/wiki/Pantalla_de_ordenador). De esta forma, los programadores se benefician de las ventajas del API haciendo uso de su funcionalidad, evitándose el trabajo de programar todo desde el principio. Las API asimismo son abstractas: el [software](https://es.wikipedia.org/wiki/Software) que proporciona una cierta API generalmente es llamado la implementación de esa API.

Por ejemplo, se puede ver la tarea de escribir "[Hola Mundo](https://es.wikipedia.org/wiki/Programa_Hola_Mundo)" sobre la pantalla en diferentes niveles de [abstracción](https://es.wikipedia.org/wiki/Abstracci%C3%B3n_(programaci%C3%B3n_orientada_a_objetos)):

1.Haciendo todo el trabajo desde el principio:

1.Traza, sobre [papel](https://es.wikipedia.org/wiki/Papel) milimetrado, la forma de las letras (y espacio) "H, o, l, a, M, u, n, d, o".

2.Crea una [matriz](https://es.wikipedia.org/wiki/Matriz_(programaci%C3%B3n)) de [cuadrados](https://es.wikipedia.org/wiki/Cuadrado) negros y blancos que se asemeje a la sucesión de letras.

3.Mediante instrucciones en [ensamblador](https://es.wikipedia.org/wiki/Ensamblador), escribe la información de la matriz en la [memoria intermedia](https://es.wikipedia.org/wiki/Memoria_intermedia)(búfer) de pantalla.

4.Mediante la instrucción adecuada, haz que la [tarjeta gráfica](https://es.wikipedia.org/wiki/Tarjeta_gr%C3%A1fica) realice el [volcado](https://es.wikipedia.org/wiki/Volcado) de esa información sobre la pantalla.

2.Por medio de un [sistema operativo](https://es.wikipedia.org/wiki/Sistema_operativo) para hacer parte del trabajo:

1.Carga una [fuente](https://es.wikipedia.org/wiki/Tipo_de_letra) tipográfica proporcionada por el sistema operativo.

2.Haz que el sistema operativo borre la pantalla.

3.Haz que el sistema operativo dibuje el texto "Hola Mundo" usando la fuente cargada.

3.Usando una [aplicación](https://es.wikipedia.org/wiki/Aplicaci%C3%B3n_(inform%C3%A1tica)) (que a su vez usa el sistema operativo) para realizar la mayor parte del trabajo:

1.Escribe un documento [HTML](https://es.wikipedia.org/wiki/HTML) con las palabras "Hola Mundo" para que un [navegador 2.web](https://es.wikipedia.org/wiki/Navegador_web) como [Firefox](https://es.wikipedia.org/wiki/Firefox), [Chrome](https://es.wikipedia.org/wiki/Google_Chrome), [Opera](https://es.wikipedia.org/wiki/Opera_(navegador)), [Safari](https://es.wikipedia.org/wiki/Safari_(navegador)), [Midori](https://es.wikipedia.org/wiki/Midori_(navegador)), [Iceweasel](https://es.wikipedia.org/wiki/Iceweasel), [Web](https://es.wikipedia.org/wiki/Epiphany) o [Internet Explorer](https://es.wikipedia.org/wiki/Internet_Explorer) pueda representarlo en el [monitor](https://es.wikipedia.org/wiki/Pantalla_de_ordenador).

Como se puede ver, la primera opción requiere más pasos, cada uno de los cuales es mucho más complicado que los pasos de las opciones siguientes. Además, no resulta nada práctico usar el primer planteamiento para representar una gran cantidad de [información](https://es.wikipedia.org/wiki/Informaci%C3%B3n), como un artículo enciclopédico sobre la pantalla, mientras que el segundo enfoque simplifica la tarea eliminando un paso y haciendo el resto más sencillos y la tercera forma simplemente requiere escribir "Hola Mundo". Sin embargo, las API de alto nivel generalmente pierden flexibilidad; por ejemplo, resulta mucho más difícil en un [navegador web](https://es.wikipedia.org/wiki/Navegador_web) hacer girar texto alrededor de un punto con un contorno parpadeante que programarlo a bajo nivel. Al elegir usar una API se debe llegar a un cierto equilibrio entre su potencia, simplicidad y pérdida de flexibilidad.

**Ejemplos de interfaces de programación**

1. [CORBA](https://es.wikipedia.org/wiki/CORBA) (Common Object Request Broker Architecture)
2. [Drupal](https://es.wikipedia.org/wiki/Drupal) API
3. [Glibc](https://es.wikipedia.org/wiki/Glibc)
4. Microsoft [Win32](https://es.wikipedia.org/wiki/Win32) API
5. Microsoft [WMI](https://es.wikipedia.org/wiki/WMI)
6. ASPI, interfaz de software para asistir a dispositivos de bus [SCSI](https://es.wikipedia.org/wiki/SCSI)
7. The [Carbon](https://es.wikipedia.org/wiki/Carbon_(API)" \o "Carbon (API)) APIs for the [Apple Macintosh](https://es.wikipedia.org/wiki/Apple_Macintosh) [Mac OS X](https://es.wikipedia.org/wiki/Mac_OS_X)
8. [Symfony](https://es.wikipedia.org/wiki/Symfony) para [PHP](https://es.wikipedia.org/wiki/PHP)
9. [DirectX](https://es.wikipedia.org/wiki/DirectX)
10. [OpenGL](https://es.wikipedia.org/wiki/OpenGL)

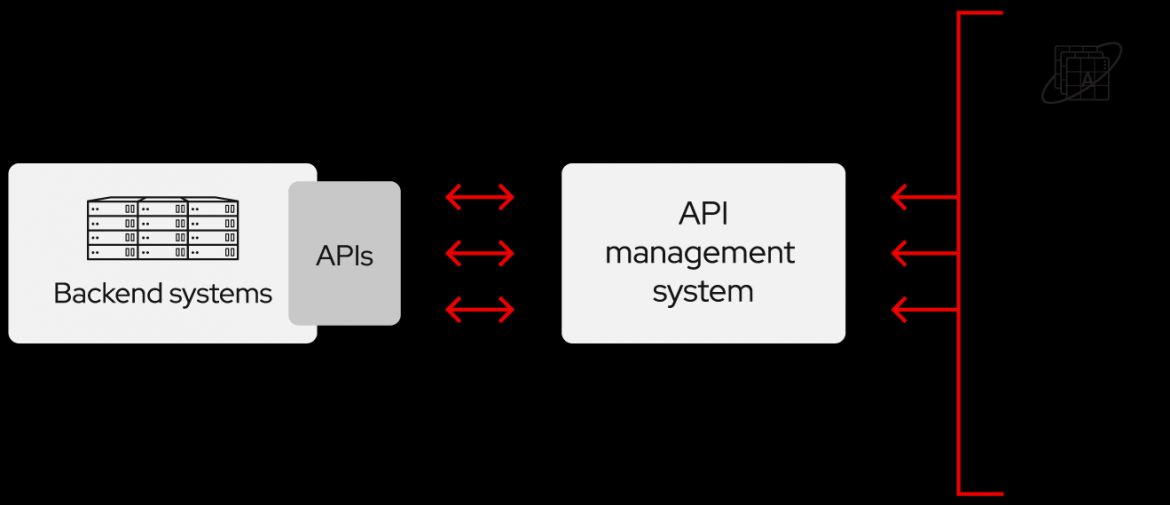
Una API es un conjunto de definiciones y protocolos que se utiliza para desarrollar e [integrar](https://www.redhat.com/es/topics/integration) el software de las aplicaciones. API significa interfaz de programación de aplicaciones.

Las API permiten que sus productos y servicios se comuniquen con otros, sin necesidad de saber cómo están implementados. Esto simplifica el desarrollo de las aplicaciones y permite ahorrar tiempo y dinero. Las API le otorgan flexibilidad; simplifican el diseño, la administración y el uso de las aplicaciones, y proporcionan oportunidades de innovación, lo cual es ideal al momento de diseñar herramientas y productos nuevos (o de gestionar los actuales).

A veces, las API se consideran como contratos, con documentación que representa un acuerdo entre las partes: si una de las partes envía una solicitud remota con cierta estructura en particular, esa misma estructura determinará cómo responderá el software de la otra parte.

Debido a que simplifican la forma en que los desarrolladores integran los elementos de las aplicaciones nuevas en una arquitectura actual, las API permiten la colaboración entre el equipo comercial y el de TI. Las necesidades comerciales suelen cambiar rápidamente en respuesta a los mercados digitales en constante cambio, donde la competencia puede modificar un sector entero con una aplicación nueva. Para seguir siendo competitivos, es importante admitir la implementación y el desarrollo rápidos de servicios innovadores. El desarrollo de [aplicaciones nativas de la nube](https://www.redhat.com/es/topics/cloud-native-apps) es una forma identificable de aumentar la velocidad de desarrollo y se basa en la conexión de una arquitectura de aplicaciones de [micro servicios](https://www.redhat.com/es/topics/microservices) a través de las API.

Las API son un medio simplificado para conectar su propia infraestructura a través del desarrollo de aplicaciones nativas de la nube, pero también le permiten compartir sus datos con clientes y otros usuarios externos. Las API públicas representan un valor comercial único porque simplifican y amplían la forma en que se conecta con sus partners y, además, pueden rentabilizar sus datos (un ejemplo conocido es la API de Google Maps).



**Imagen gracias a:** <https://www.redhat.com/es/topics/api/what-are-application-programming-interfaces>

Por ejemplo, imagínese una empresa distribuidora de libros. *Podría* ofrecer a los clientes una aplicación que les permita a los empleados de la librería verificar la disponibilidad de los libros con el distribuidor. El desarrollo de esta aplicación podría ser costoso, estar limitado por la plataforma y requerir mucho tiempo de desarrollo y mantenimiento continuo.

Otra opción es que la distribuidora de libros proporcionara una API para verificar la disponibilidad en inventario. Existen varios beneficios de este enfoque:

* Permite que los clientes accedan a los datos con una API que les ayude a añadir información sobre su inventario en un solo lugar.
* La distribuidora de libros podría realizar cambios en sus sistemas internos sin afectar a los clientes, siempre y cuando el comportamiento de la API fuera el mismo.
* Con una API disponible de forma pública, los desarrolladores que trabajan para la distribuidora de libros, los vendedores o los terceros podrían desarrollar una aplicación para ayudar a los clientes a encontrar los libros que necesiten. Esto podría dar como resultado mayores ventas u otras oportunidades comerciales.

En resumen, las API le permiten [habilitar el acceso a sus recursos y, al mismo tiempo, mantener la seguridad y el control](https://www.redhat.com/es/topics/security). Cómo habilitar el acceso y a quiénes depende de usted. [La seguridad de las API tiene que ver con que se gestionen bien](https://www.redhat.com/es/topics/security/api-security). Para conectarse a las API y crear aplicaciones que utilicen los datos o las funciones que estas ofrecen, se puede utilizar una plataforma de integración distribuida que conecte todos los elementos, incluidos los sistemas heredados y el Internet de las cosas (IoT). Existen tres enfoques respecto a las políticas de las versiones de las API.

#### **Privado**

Las API solo se pueden usar internamente, así que las empresas tienen un mayor control sobre ellas. Esto le da a las empresas un mayor control sobre sus API.

**De partners**

Las API se comparten con partners empresariales específicos, lo cual puede ofrecer flujos de ingresos adicionales, sin comprometer la calidad. Esto puede proporcionar flujos de ingreso adicionales, sin comprometer la calidad.

**Público**

Todos tienen acceso a las API, así que otras empresas pueden desarrollar API que interactúen con las de usted y así convertirse en una fuente de innovaciones. Esto permite que terceros desarrollen aplicaciones que interactúan con su API, y puede ser un recurso para innovar.

Breve historia de las API

Las API surgieron los primeros días de la informática, mucho antes que la computadora personal. En esa época, una API normalmente se usaba como biblioteca para los sistemas operativos. Casi siempre estaban habilitadas localmente en los sistemas en los que operaban, aunque a veces pasaban mensajes entre las computadoras centrales. Después de casi 30 años, las API se expandieron más allá de los entornos locales. A principios del año 2000, ya eran una tecnología importante para la integración remota de datos.

Las API remotas

Las API remotas están diseñadas para interactuar en una red de comunicaciones. Por "remoto" nos referimos a que los recursos que administran las API están, de alguna manera, fuera de la computadora que solicita alguno de dichos recursos. Debido a que la red de comunicaciones más usada es Internet, la mayoría de las API están diseñadas de acuerdo con los estándares web. No todas las API remotas son API web, pero se puede suponer que las API web son remotas.

Las API web normalmente usan HTTP para solicitar mensajes y proporcionar una definición de la estructura de los mensajes de respuesta. Por lo general estos mensajes de respuesta toman la forma de un archivo XML o JSON, que son los formatos preferidos porque presentan los datos en una manera fácil de manejar para otras aplicaciones.

¿Qué se ha hecho para mejorar las API?

A medida que las API han evolucionado en las ahora generalizadas API web, se han realizado muchos esfuerzos para simplificar su diseño y facilitar su implementación.

Un poco de SOAP, mucho de REST

A medida que se han difundido las API, se desarrolló una especificación de protocolo para permitir la estandarización del intercambio de información; se llama Protocolo de Acceso a Objetos Simples, más conocido como SOAP. Las API diseñadas con SOAP usan XML para el formato de sus mensajes y reciben solicitudes a través de HTTP o SMTP. Con SOAP, es más fácil que las aplicaciones que funcionan en entornos distintos o están escritas en diferentes lenguajes compartan información.

Otra especificación es la Transferencia de Estado Representacional (REST). Las API web que funcionan con las limitaciones de arquitectura REST se llaman API de RESTful. La diferencia entre REST y SOAP es básica: SOAP es un protocolo, mientras que REST es un estilo de arquitectura. Esto significa que no hay ningún estándar oficial para las API web de RESTful. Tal como se define en la tesis de Roy Fielding, [“Architectural Styles and the Design of Network-based Software Architectures”](https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm), las API son RESTful siempre que cumplan con las 6 limitaciones principales de un sistema RESTful:

Las API y Red Hat

Red Hat le ofrece soluciones de API modulares, ligeras e integrales que utilizan el open source y los estándares abiertos, y que están disponibles tanto en entornos locales como en la nube. Son un factor importante en cómo puede [cambiar su infraestructura de integración actual para que sea más flexible y aporte valor con mayor rapidez](https://www.redhat.com/es/topics/integration/why-choose-red-hat-integration).

**Comentario Personal**

Un api para mi es una base de datos registrados de forma de un array permitiendo que las personas puedan partir de parte de sus datos el conocimiento de actuar y mover códigos para poder transmitir mensajes para que el api sea algo globalizado “Las APIs se convierten en un elemento esencial para exponer información y operativas para su consumo por terceros de una forma estándar y controlada. Desde mi punto de vista, las APIs son la puerta que dejamos abierta, a propósito, para presentar al mundo nuestros productos y servicios de cara a que cada individuo o empresa las utilice de la forma que crea que le aportará valor (AS-IS, combinando otras fuentes, creando aplicaciones agregadas, etc.). Con ello esperamos generar algún beneficio: cualitativos y/o cuantitativos”, "El objetivo es poner las aplicaciones en manos de la gente de forma rápida y eficiente, asistiendo al proceso de innovación. Y es que no hay nada que frene más la innovación que la fricción o los frenos en los procesos operativos. Con las APIs, cualquiera puede experimentar y ofrecer un nuevo servicio".

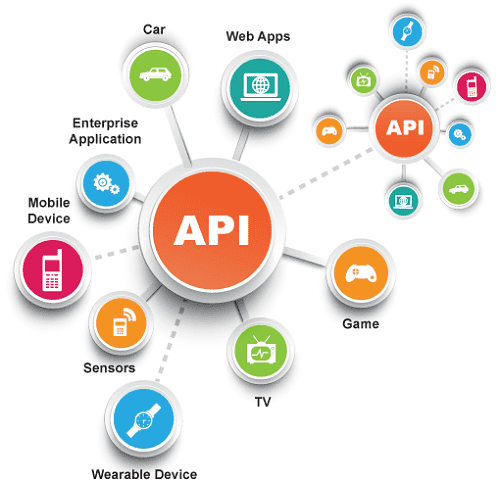


Imagen gracias a: <https://blog.loadimpact.com/load-testing-restful-apis-with-k6>

“En mi opinión, uno de los mayores beneficios de diseñar, desarrollar y trabajar con APIs en la web es que los desarrolladores se han visto obligados a prestar atención a lo que todo el mundo está haciendo y deseando. Yo sufro de esto. Hacer web APIs, proporcionar la mía propia y consumir APIs de terceros me obliga a prestar atención a proveedores y consumidores fuera de mi burbuja. Esto es bueno.” [Kin Lane,](https://www.linkedin.com/in/kinlane" \t "_blank)



Imagen gracias a: <https://bbvaopen4u.com/es/actualidad/7-opiniones-de-grandes-expertos-sobre-apis>

“Las APIs proporcionan muchísimas oportunidades para cualquier empresa y cualquier negocio. La API es una de las tecnologías más transversales que está haciendo posible la transformación digital de los negocios tradicionales”. [**José Manuel de la Chica**](https://www.linkedin.com/in/delachica)**,**

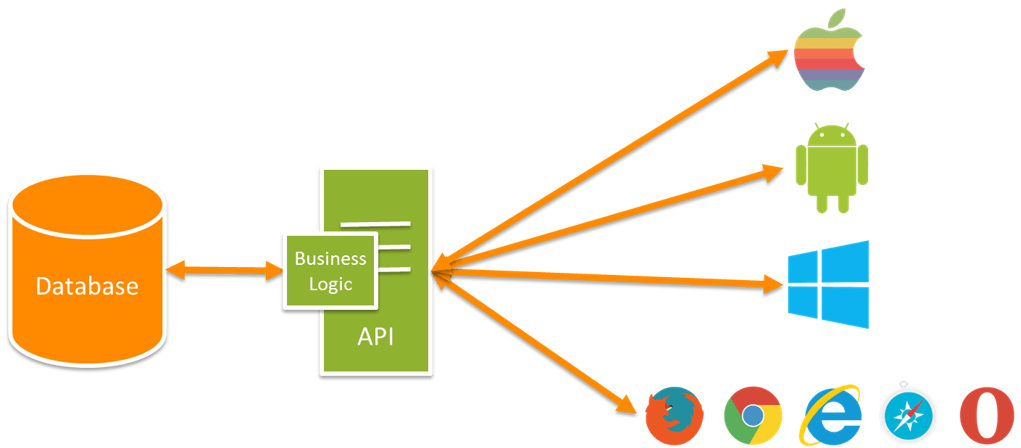


Imagen gracias a: <https://snipcart.com/blog/apis-integration-usage-benefits>

“Se habla mucho hoy en día de APIs y plataformas. Es un tema casi tan candente como blockchain. Una parte está siendo impulsada por la reglamentación, especialmente en Europa. [**La PSD2**](https://bbvaopen4u.com/es/actualidad/como-afecta-la-psd2-la-identidad-digital-de-los-clientes-de-banca)**está obligando a los bancos a abrir sus APIs**, aunque en parte no es más que la evolución del sector”. [**Shamir Karkal**](https://www.linkedin.com/in/shamirkarkal),

Gracias Por dedicarte a leer este trabajo Dios te bendiga.