1 averiguar que microservicios ofrece facebook, twitter, google,microsoft entre otros (es mas, hacerme unos ejemplos conn almenos otras 3 aplicaciones)

Facebook

### Los servicios que ofrece Facebook

* **Lista de amigos:** En ella, el usuario puede agregar a cualquier persona que conozca y esté registrada, siempre que acepte su invitación. En Facebook se pueden localizar amigos con quienes se perdió el contacto o agregar otros nuevos con quienes intercambiar fotos o mensajes. Para ello, el servidor de Facebook posee herramientas de búsqueda y de sugerencia de amigos.
* **Chat:** Servicio de mensajería instantánea en dispositivos móviles y computadores a través de Facebook Messenger.
* **Grupos y páginas:** Es una de las utilidades de mayor desarrollo reciente. Se trata de reunir personas con intereses comunes. En los grupos se pueden añadir fotos, vídeos, mensajes, etc. Las páginas, se crean con fines específicos y a diferencia de los grupos no contienen foros de discusión, ya que están encaminadas hacia marcas o personajes específicos y no hacia ningún tipo de convocatoria.

Además, los grupos también tienen su normativa, entre la cual se incluye la prohibición de grupos con temáticas discriminatorias o que inciten al odio y falten al respeto y la honra de las personas. Si bien esto no se cumple en muchas ocasiones, existe la opción de denunciar y reportar los grupos que vayan contra esta regla, por lo cual Facebook incluye un enlace en cada grupo el cual se dirige hacia un cuadro de reclamos y quejas.

* **Muro:** el muro (*wall* en inglés) es un espacio en cada perfil de usuario que permite que los amigos escriban mensajes para que el usuario los vea. Sólo es visible para usuarios registrados. Permite ingresar imágenes y poner cualquier tipo de logotipos en tu publicación. Una mejora llamada supermuro permite incrustar animaciones flash, etc. En noviembre de 2011, Facebook comenzó a implementar un sustituto del muro, el cual llevará por nombre **Biografía**.
* **Fotos:** Según Facebook, hay:
  + 5 mil millones de fotos de usuario.
  + 160 terabytes de almacenaje.
* **Regalos:** los regalos o *gifts* son pequeños íconos con un mensaje. Los regalos dados a un usuario aparecen en la pared con el mensaje del donante, a menos que el donante decida dar el regalo en privado, en cuyo caso el nombre y el mensaje del donante no se exhibe a otros usuarios.
* **Botón me gusta:** Esta función aparece en la parte inferior de cada publicación hecha por el usuario o sus contactos (actualizaciones de estado, contenido compartido, etc), se caracteriza por un pequeño ícono en forma de una mano con el dedo pulgar hacia arriba. Permite valorar si el contenido es del agrado del usuario actual en la red social, del mismo modo se notifica a la persona que expuso ese tema originalmente si es del agrado del alguien más (alguno de sus contactos).

Una opción "anónima" está también disponible, por la cual cualquier persona con el acceso del perfil puede ver el regalo, pero solamente el destinatario verá el mensaje.

Algunos regalos son gratuitos y el resto cuestan un dólar (es necesario un número de tarjeta de crédito o cuenta Paypal).

* **App Center:** contendrá de las mejores apps disponibles para la red social. Mostrará los hábitos de cada persona, las aplicaciones que estén más relacionadas con su actividad diaria. Se podrá ingresar a la tienda desde internet como dispositos móviles. Cada aplicación tendrá una página con descripción, que incluirá imágenes y opiniones de usuarios.
* **Aplicaciones:** Son pequeñas aplicaciones con las que puedes averiguar tu galleta de la suerte, quien es tu mejor amigo, descubrir cosas de tu personalidad.
* **Juegos:** la mayoría de aplicaciones encontradas en Facebook se relacionan con juegos de rol, juegos parecidos al Trivial Pursuit (p. ej. geografía), o pruebas de habilidades (digitación, memoria). Entre los más célebres se encuentran los juegos de Playfish,como Pet society, los juegos de Zynga Games como FarmVille y CityVille además los juegos de Digital Chocolate como Tower Bloxx.

Twitter

### ****. Para hacer encuestas.****

¿Quiere enterarse de qué opinan sus amigos sobre un tema en particular y seguir los resultados? Con [POLL Pigeon](http://www.pollpigeon.com/) es posible crear encuestas de infinidad de asuntos en pocos pasos. El sitio enlaza la pregunta a su cuenta de Twitter o Facebook y le enseña los resultados periódicamente.

### ****2. Envíe tweets cuando esté ausente.****

Los adictos a Twitter seguramente no querrán que, al no poder conectarse a la Red, sus fanáticos o ‘followers’ dejen de recibir mensajes. Como solución, es posible utilizar [FutureTweets](http://www.futuretweets.com/), que permite programar tweets para que se vayan publicando en su cuenta de Twitter según la fecha y hora que usted disponga. Esta alternativa también la ofrecen algunos clientes de Twitter.

### ****3. Estadísticas de su perfil.****

Con estas herramientas podrá saber cuántos mensajes en promedio envía por día, en qué días y horas lo hace con más frecuencia, al igual que analizar la información de sus seguidores. No debe registrarse, tan solo escribir su nombre de usuario.[TwitterCounter](http://www.twittercounter.com/), [TwitterGrader](http://twitter.grader.com/), [TweetStats](http://www.tweetstats.com/). Una explicación más detallada se encuentra en [Herramientas para perder la inocencia en Twitter](http://www.enter.co/otros/herramientas-para-perder-la-inocencia-en-twitter/).

### ****4. Búsqueda de información.****

Twitter cuenta con una [herramienta de búsqueda propia](https://www.enter.co/search.twitter.com), desde la cual es posible hallar información relevante sobre la que se esté discutiendo en la red social. Esto también es posible hacerlo desde Google, al digitar ‘**site:twitter.com**‘ seguido de los términos que desee. Para encontrar personas en Twitter según los parámetros que usted defina, puede usar [Just Tweet It](http://www.justtweetit.com/).

### ****5. Compartir fotografías.****

[TwitPic](http://www.twitpic.com/) es un servicio sencillo que permite compartir imágenes en Twitter desde un teléfono móvil o a través del sitio web. Para usarlo no se necesita crear una cuenta de usuario, pues se utiliza la de Twitter. Allí mismo se van almacenando las fotos, que se pueden consultar en cualquier momento.

### ****6. Reduzca las direcciones web.****

Los 140 caracteres de cada mensaje en Twitter impiden escribir direcciones largas, por lo que se utilizan servicios que reducen esto a pocas letras. Uno de ellos es [Bit.ly](http://www.bit.ly/), que cuenta con una apariencia sencilla acorde con su propósito. En la casilla que se muestra en la parte superior se ingresa la dirección original del sitio web y al dar clic en ‘Shorten’ se muestra una versión acortada. Otra opción es [Goo.gl](http://www.goo.gl/). Los clientes oficiales de Twitter ya no necesitan de estos servicios, pues automáticamente acortan las direcciones con el servicio propio de Twitter, [t.co](http://t.co/).

### ****7. Video en línea.****

Si está en un lugar interesante y quiere transmitirles a sus seguidores las imágenes en vivo de lo que está sucediendo, puede usar [Ustream](http://www.ustream.tv/), que está disponible para diversos dispositivos móviles. Una vez comienza a transmitir el video es posible enviar un tweet con la dirección en la Red en la cual lo pueden ver. [Qik](http://www.qik.com/) también funciona para este fin.

### ****8. Preguntas y respuestas.****

Si no teme responder a preguntas de cualquier tipo que puedan plantearle sus seguidores, es bueno que pruebe [Formspring](http://www.formspring.me/), con el que estos pueden preguntarle de todo, incluso de forma anónima. La clave es contestar rápidamente y hacerlo de una manera que atraiga a más personas a preguntar. También es posible vincular la cuenta de Formspring con Facebook. Aunque el servicio estuvo de moda a finales de 2009, todavía tiene sus fanáticos y sigue siendo un espacio atractivo para conocer y dejarse conocer mejor de otros twitteros.

### ****9. Administre más aspectos su cuenta.****

Entre las funciones interesantes de [SocialToo](http://www.socialtoo.com/) se encuentran las siguientes: es posible configurar de manera automática para que usted siga a cada persona que lo sigue; deja de seguir a los que lo dejan de seguir a usted; crea una lista negra de los usuarios que usted considera como spam; y bloquea las comunicaciones que llegan a su correo cuando le envían un mensaje directo. También permite crear encuestas y publicarlas tanto en Twitter como en Facebook.

### ****10. Limpie su Twitter.****

[The Twit Cleaner](http://www.thetwitcleaner.com/) es un servicio útil para analizar a sus seguidores y a los que usted sigue para saber quiénes son. Así puede encontrar aquellos que poco participan, saber cuáles son ‘bots’ o usuarios fantasmas que se crean para enviar spam, entre otros. El sitio crea un reporte detallado con aquellos usuarios que considera sospechosos y permite dejarlos de seguir. Otra alternativa es [Tweepi](http://www.tweepi.com/).

### ****11. Comparta archivos.****

Con [FileSocial](http://www.filesocial.com/) usted podrá compartir cualquier tipo de archivo de hasta 50 MB con sus seguidores. Para acceder es posible utilizar la misma cuenta de Twitter y seguir los pasos para cargar la información y que, de inmediato, se muestre en tweet con el enlace. Allí cualquier persona puede descargar el contenido.

### ****12. Extensión para su navegador.****

Si quiere ver toda la información que se maneja en Twitter desde este mismo sitio en su navegador, puede instalar extensiones que lo mantengan conectado a Twitter permanentemente, Para Firefox, Chrome y Safari, instale la extensión [Power Twitter](http://addons.mozilla.org/en-US/firefox/addon/9591). Así, podrá ver videos de YouTube, mapas de Google, fotografías y más contenidos multimedia desde su propia cuenta, sin necesidad de abrir pestañas adicionales. Para Safari existe también una extensión oficial de Twitter, y para todos –incluido Opera– hay más extensiones con funciones puntuales, como compartir una página web en Twitter con un clic.

### ****13. Encuentre lo más popular.****

Para encontrar qué temas se están discutiendo más en Twitter, [TweetMeme](http://www.tweetmeme.com/) recopila los enlaces más compartidos en esta red social y los muestra en un listado que permite ver cuáles son los más populares. Cuenta con categorías como comedia, entretenimiento, juegos, estilo de vida, ciencia, deportes y tecnología. Por otro lado, servicios como [Trendsmap](http://trendsmap.com/), [Trendistic](http://www.trendistic.com/) o [Twirus](http://www.twirus.com/) sirven para encontrar tendencias (‘trending topics’ o twitteros populares en un momento dado).

### ****14. Personalice su página.****

Si considera muy básicas o poco entretenidas las opciones de cambio de apariencia que ofrece Twitter, es posible acceder a servicios como [Themeleon](http://www.colourlovers.com/themeleon/twitter), que permite, utilizando la misma cuenta de Twitter, cambiar colores, diseño y estilos. Cuenta con más de 650 mil fondos e innumerables colores para crear lo que más se ajuste a su gusto.

### 15. Use ‘browsers sociales’.

Si Twitter –u otras redes sociales– es tan importante para usted que debería estar tan integrado a toda su experiencia de navegación web, y las extensiones sugeridas en el punto 12 no son suficientes, tal vez deba probar [Rockmelt](http://www.rockmelt.com/) o [Flock](http://www.flock.com/), los navegadores centrados en las redes sociales. El primero está basado en Webkit, la misma tecnología de Google Chrome y Safari, y el segundo, que se basaba en el motor Gecko, de Firefox, ahora también usa Webkit. El cambio a alguno de ellos no debería ser traumático para un usuario de los otros navegadores.

Google

* [2 Google Chrome, el navegador de Google](https://www.internet-didactica.es/servicios-e-historia-google/#Google_Chrome_el_navegador_de_Google)
* [3 Google+: La red que lo conecta todo](https://www.internet-didactica.es/servicios-e-historia-google/#Google_La_red_que_lo_conecta_todo)
* [4 YouTube, la plataforma de vídeos](https://www.internet-didactica.es/servicios-e-historia-google/#YouTube_la_plataforma_de_videos)
* [5 Gmail, una opción genial para gestionar nuestro correo](https://www.internet-didactica.es/servicios-e-historia-google/#Gmail_una_opcion_genial_para_gestionar_nuestro_correo)
* [6 Android, el sistema operativo móvil](https://www.internet-didactica.es/servicios-e-historia-google/#Android_el_sistema_operativo_movil)
* [7 Google Drive](https://www.internet-didactica.es/servicios-e-historia-google/#Google_Drive)
* [8 Google Docs](https://www.internet-didactica.es/servicios-e-historia-google/#Google_Docs)
* [9 Google Search Console y Analytics](https://www.internet-didactica.es/servicios-e-historia-google/#Google_Search_Console_y_Analytics)
* [10 Google Adwords](https://www.internet-didactica.es/servicios-e-historia-google/#Google_Adwords)
* [11 Google My Business](https://www.internet-didactica.es/servicios-e-historia-google/#Google_My_Business)

Microsoft

**Motivos para usar los microservicios**

**Compilar servicios de forma independiente**

Las aplicaciones basadas en microservicios están compiladas como una colección de servicios altamente desacoplados que controlan una sola acción. Los equipos pueden compilar, comprobar, implementar y supervisar cada servicio de manera independiente.

**Escalar servicios de forma autónoma**

Los servicios independientes se pueden escalar según las respectivas exigencias, sin que ello afecte al rendimiento general, en lugar de escalar o reducir verticalmente la aplicación completa.

**Usar el mejor enfoque**

Los equipos de desarrollo obtienen flexibilidad para usar el enfoque de implementación, el lenguaje, la plataforma y el modelo de programación más recomendados para cada servicio.

**Aislar puntos de error**

Al aislar áreas problemáticas potenciales para servicios individuales, las arquitecturas de microservicios mejoran la seguridad y la confiabilidad. Los servicios se pueden reemplazar o retirar sin que ello afecte a la estructura general.

**Entregar valor más rápido**

Los equipos pueden implementar rápidamente pequeños módulos independientes. Varios equipos pueden trabajar en distintos servicios al mismo tiempo y poner nuevas características en producción más rápido.

## Implementación de un microservicio CRUD sencillo con ASP.NET Core

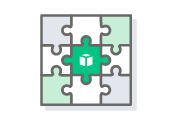
Amazon

# **La plataforma más completa para los microservicios**

AWS tiene bloques de creación integrados que admiten cualquier arquitectura de aplicación, independientemente de la escala, la carga o la complejidad.

## Computación

Procesamiento de la potencia de los microservicios



### Contenedores

Amazon Elastic Container Service

Servicio de administración de contenedores de alto rendimiento y escalabilidad compatible con los contenedores de Docker, que le permite ejecutar fácilmente aplicaciones distribuidas en un clúster administrado de instancias de Amazon EC2.  
[Más información »](https://aws.amazon.com/ecs/)

DevOps-Solution_logobreak_coursera

Con Amazon ECS, Coursera puede implementar cambios en el software en cuestión de minutos en lugar de demorar horas, en un entorno con los recursos aislados.  
[Más información »](https://aws.amazon.com/solutions/case-studies/coursera-ecs/)



### Sin servidor

AWS Lambda

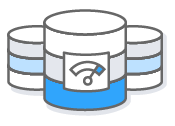
AWS Lambda le permite ejecutar código sin aprovisionar ni administrar servidores. Solo tiene que cargar el código y Lambda administrará todo lo necesario para ejecutar y escalar el código con alta disponibilidad.  
[Más información »](https://aws.amazon.com/lambda/)

DevOps-Solution_logobreak_localytics

Localytics usó AWS Lambda para crear microservicios que permitieron a sus equipos de desarrollo crear análisis personalizados sin soporte centralizado. [Más información »](https://aws.amazon.com/solutions/case-studies/localytics/)

## Almacenamiento y bases de datos

Almacenamiento de datos escalable, duradero y seguro.



### Almacenamiento en caché

Amazon ElastiCache

Amazon ElastiCache mejora el rendimiento del servicio, lo que le permite recuperar información de cachés en memoria rápidos y administrados, en lugar de depender totalmente de bases de datos más lentas basadas en disco. [Más información »](https://aws.amazon.com/elasticache/)

### Almacenamiento de objetos

Amazon S3

Amazon S3 ofrece a los desarrolladores y equipos de TI un almacenamiento de objetos altamente confiable, seguro y escalable para todos sus datos, grandes o pequeños. [Más información »](https://aws.amazon.com/s3/)

### Bases de datos NoSQL

Amazon DynamoDB

Un servicio de base de datos NoSQL rápido, flexible y completamente administrado para todas las aplicaciones que requieren latencias de milisegundos de un solo dígito, constantes y a cualquier escala. [Más información »](https://aws.amazon.com/dynamodb/)

### Bases de datos relacionales

Amazon RDS

Configure, maneje y escale fácilmente una [base de datos relacional](https://aws.amazon.com/relational-database/) en la nube. Seleccione entre seis motores de bases de datos conocidos, incluso Oracle, Microsoft SQL Server, PostgreSQL, MySQL y MariaDB. [Más información »](https://aws.amazon.com/rds/)

Amazon Aurora

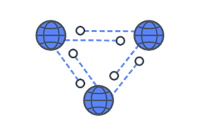
Es un motor de [base de datos relacional](https://aws.amazon.com/relational-database/) que combina la velocidad y la fiabilidad de las bases de datos comerciales de gama alta con la sencillez y la rentabilidad de las bases de datos de código abierto. Cuenta con un rendimiento hasta 5 veces mayor que el que se consigue con la ejecución de MySQL estándar en el mismo hardware.[Más información »](https://aws.amazon.com/rds/aurora/)

DevOps-Solution_logobreak_remind

Recuerde los tiempos de respuesta de las aplicaciones reducido en un 200 % mediante la creación de un PaaS para microservicios en Amazon ECS.   
[Más información »](https://aws.amazon.com/solutions/case-studies/remind/)

## Redes

Servicios de red con alto rendimiento y latencia inferior a milisegundos.



### Detección de servicios

AWS Cloud Map

AWS Cloud Map detecta servicios para todos sus recursos en la nube. Con Cloud Map, puede definir nombres personalizados para los recursos de su aplicación, y el servicio mantiene la ubicación actualizada de estos recursos que cambian dinámicamente.

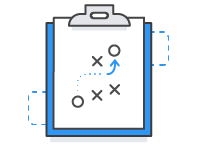
[Más información »](https://aws.amazon.com/cloud-map/)

### Red de servicios

AWS App Mesh

AWS App Mesh facilita la tarea de controlar y monitorizar microservicios que se ejecutan en AWS. App Mesh estandariza la comunicación de sus microservicios, brindándole visibilidad integral y ayudando a garantizar una alta disponibilidad para sus aplicaciones.

[Más información »](https://aws.amazon.com/app-mesh/)



### Elastic Load Balancing

Balanceador de carga de aplicaciones

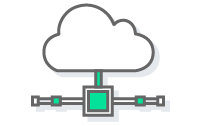
El balanceador de carga de aplicaciones equilibra la carga del tráfico HTTP y HTTPS en la capa de la aplicación (nivel 7) proporcionando un direccionamiento de solicitudes avanzado dirigido a la entrega de arquitecturas de aplicaciones modernas, incluso microservicios y contenedores.

[Más información »](https://aws.amazon.com/elasticloadbalancing/details/)

Balanceador de carga de red

El balanceador de carga de red ofrece un equilibrio de carga de alto rendimiento que opera en la capa de conexión de red (nivel 4) y le permite direccionar conexiones a microservicios basados en datos de protocolos de IP. El balanceador de carga de red puede manejar millones de solicitudes por segundo y mantener latencias muy bajas.

[Más información »](https://aws.amazon.com/elasticloadbalancing/details/)



### Proxy de la API

Amazon API Gateway

Amazon API Gateway ofrece una plataforma integral para la [administración de API](https://aws.amazon.com/api-gateway/api-management/). Con Amazon API Gateway, se pueden procesar cientos de miles de llamadas a la API simultáneas y se administran el tráfico, el control de acceso y la autorización, la monitorización y la administración de versión de API.

[Más información »](https://aws.amazon.com/api-gateway/)

### DNS

Amazon Route 53

Amazon Route 53 es un servicio web de sistema de nombres de dominio (DNS) en la nube, altamente disponible y escalable que conecta de manera efectiva las solicitudes a la infraestructura que se ejecuta en AWS. Puede utilizarse para las comprobaciones de estado de IP y la detección de servicios para microservicios.

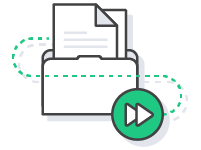
[Más información »](https://aws.amazon.com/route53/)



Airtime ofrece su experiencia en redes sociales a clientes de una manera más rápida, ágil y sin tiempo de retardo, después de haber rediseñado su aplicación como microservicios que se ejecutan en AWS. [Más información »](https://aws.amazon.com/solutions/case-studies/airtime/)

## Mensajería

Publique y coordine las comunicaciones entre los procesos.



### Suscripción y publicación de mensajes

Amazon Simple Notification Service (Amazon SNS)

Amazon SNS es un servicio de mensajería de publicación/suscripción completamente administrado que facilita la desasociación y el ajuste de escala de los microservicios, sistemas distribuidos y las aplicaciones sin servidor.  
[Más información »](https://aws.amazon.com/sns/)

### Colas de mensajes

Amazon Simple Queue Service (Amazon SQS)

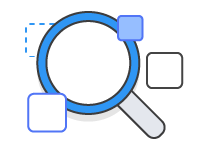
Amazon SQS es un servicio de colas de mensajes completamente administrado que facilita la desasociación y el escalado de microservicios, sistemas distribuidos y aplicaciones sin servidor.  
[Más información »](https://aws.amazon.com/sqs/)



Lyft usa AWS para moverse más rápido como empresa y administrar su crecimiento exponencial, aprovechando los productos de AWS para admitir más de 100 microservicios que mejoran cada elemento de la experiencia de sus clientes. [Más información »](https://aws.amazon.com/solutions/case-studies/lyft/)

## Registro y supervisión

Monitorice el rendimiento de los servicios y la utilización de recursos. Haga un seguimiento en arquitecturas complejas para resolver problemas y optimizarlas.



### Monitorización de API

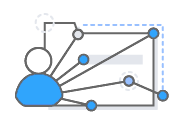
AWS CloudTrail

Con CloudTrail, puede registrar, monitorizar de manera continua y retener la actividad de la cuenta relacionada con acciones en toda su infraestructura. El historial de eventos de CloudTrail simplifica el análisis de seguridad, el seguimiento de cambios de recursos y la resolución de problemas. [Más información »](https://aws.amazon.com/cloudtrail/)

### Monitorización de aplicaciones y recursos

Amazon CloudWatch

Puede utilizar Amazon CloudWatch para recopilar y realizar el seguimiento de métricas, recopilar y monitorizar archivos de registro, establecer alarmas y reaccionar automáticamente a los cambios en los servicios en funcionamiento y en sus recursos de AWS.  
[Más información »](https://aws.amazon.com/cloudwatch/)



### Seguimiento distribuido

AWS X-Ray

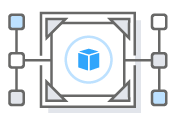
Obtenga una vista integral de las solicitudes a medida que avanzan por su aplicación y vea un mapa de los componentes subyacentes de la aplicación. Dado que un grupo de microservicios funciona en conjunto para manejar una solicitud, con AWS X-Ray puede tener una vista centralizada de los registros, lo que le permite monitorizar y resolver problemas de interacciones complejas. [Más información »](https://aws.amazon.com/xray/)



Al utilizar microservicios alojados en Amazon ECS, Shippable ha podido centrarse en ofrecer características a sus clientes y ha acelerado los tiempos de implementación de las características de una vez por semana a varias veces por día. [Más información »](https://aws.amazon.com/solutions/case-studies/shippable/)

## DevOps

Administre el ciclo de vida del código desde la confirmación hasta la ejecución.



### Repositorios de imágenes del contenedor

Amazon Elastic Container Registry (Amazon ECR)

Un registro de contenedores de Docker completamente administrado que se puede usar para almacenar, administrar e implementar imágenes de contenedores de Docker. Amazon ECR se integra con Amazon Elastic Container Service (Amazon ECS), lo que permite simplificar el desarrollo para el flujo de trabajo de producción para contenedores. [Más información »](https://aws.amazon.com/ecr/)



### Entrega continua

Herramientas para desarrolladores de AWS

Las herramientas para desarrolladores de AWS son un conjunto de servicios que permiten a los desarrolladores y profesionales de operaciones de TI que trabajan en DevOps entregar software de forma rápida y segura. En conjunto, estos servicios lo ayudan a almacenar de forma segura y a controlar la versión del código fuente de su aplicación. También le permiten compilar, probar e implementar automáticamente su aplicación en un entorno local o de AWS. [Más información »](https://aws.amazon.com/products/developer-tools/)



Gilt pasó de un centro de datos local a AWS para aprovechar la velocidad y la eficiencia de una infraestructura de microservicios basados en la nube.  
[Más información »](https://aws.amazon.com/solutions/case-studies/gilt/)

2 que es rest, soap, restful con ejemplos reales.  
  
que es rest,  
  
La **transferencia de estado representacional** (en [inglés](https://es.wikipedia.org/wiki/Idioma_ingl%C3%A9s) *representational state transfer*) o **REST** es un estilo de arquitectura *software* para sistemas [hipermedia](https://es.wikipedia.org/wiki/Hipermedia) distribuidos como la [World Wide Web](https://es.wikipedia.org/wiki/World_Wide_Web). El término se originó en el año [2000](https://es.wikipedia.org/wiki/2000), en una tesis doctoral sobre la web escrita por [Roy Fielding](https://es.wikipedia.org/wiki/Roy_Fielding), uno de los principales autores de la especificación del protocolo [HTTP](https://es.wikipedia.org/wiki/HTTP) y ha pasado a ser ampliamente utilizado por la comunidad de desarrollo.  
Si bien el término *REST* se refería originalmente a un conjunto de principios de arquitectura —descritos más abajo—, en la actualidad se usa en el sentido más amplio para describir cualquier interfaz entre sistemas que utilice directamente [HTTP](https://es.wikipedia.org/wiki/HTTP) para obtener datos o indicar la ejecución de operaciones sobre los datos, en cualquier formato ([XML](https://es.wikipedia.org/wiki/XML), [JSON](https://es.wikipedia.org/wiki/JSON), etc) sin las abstracciones adicionales de los protocolos basados en patrones de intercambio de mensajes, como por ejemplo [SOAP](https://es.wikipedia.org/wiki/SOAP). Es posible diseñar sistemas de servicios web de acuerdo con el estilo arquitectural REST de Fielding y también es posible diseñar interfaces [XMLHTTP](https://es.wikipedia.org/wiki/XMLHttpRequest) de acuerdo con el estilo de [llamada a procedimiento remoto](https://es.wikipedia.org/wiki/Remote_Procedure_Call) (RPC), pero sin usar SOAP. Estos dos usos diferentes del término *REST* causan cierta confusión en las discusiones técnicas, aunque [RPC](https://es.wikipedia.org/wiki/Remote_Procedure_Call) no es un ejemplo de REST.

REST afirma que la web ha disfrutado de escalabilidad como resultado de una serie de diseños fundamentales clave:

* Un **protocolo cliente/servidor**[**sin estado**](https://es.wikipedia.org/wiki/Protocolo_sin_estado): cada mensaje HTTP contiene toda la información necesaria para comprender la petición. Como resultado, ni el cliente ni el servidor necesitan recordar ningún estado de las comunicaciones entre mensajes. Sin embargo, en la práctica, muchas aplicaciones basadas en HTTP utilizan cookies y otros mecanismos para mantener el estado de la sesión (algunas de estas prácticas, como la reescritura de URLs, no son permitidas por REST)
* Un conjunto de **operaciones bien definidas** que se aplican a todos los *recursos* de información: HTTP en sí define un conjunto pequeño de operaciones, las más importantes son **POST**, **GET**, **PUT** y **DELETE**. Con frecuencia estas operaciones se equiparan a las operaciones [CRUD](https://es.wikipedia.org/wiki/CRUD) en bases de datos (CLAB en castellano: crear,leer,actualizar,borrar) que se requieren para la persistencia de datos, aunque POST no encaja exactamente en este esquema.
* Una **sintaxis universal** para identificar los recursos. En un sistema REST, cada recurso es direccionable únicamente a través de su [URI](https://es.wikipedia.org/wiki/Uniform_Resource_Identifier).
* El **uso de hipermedios**, tanto para la información de la aplicación como para las transiciones de estado de la aplicación: la representación de este estado en un sistema REST son **típicamente**[**HTML**](https://es.wikipedia.org/wiki/HTML)**o**[**XML**](https://es.wikipedia.org/wiki/XML). Como resultado de esto, es posible navegar de un recurso REST a muchos otros, simplemente siguiendo enlaces sin requerir el uso de registros u otra infraestructura adicional.

Dado que la definición de REST es muy amplia, es posible afirmar que existe un enorme número de aplicaciones REST en la red (prácticamente cualquier cosa accesible mediante una petición HTTP GET). De forma más restrictiva, en contraposición a los servicios web y el RPC, REST se puede encontrar en diferentes áreas de la web:

* La [blogosfera](https://es.wikipedia.org/wiki/Blogosfera) -el universo de los [blogs](https://es.wikipedia.org/wiki/Blog)- está, en su mayor parte, basado en REST, dado que implica descargar ficheros XML (en formato [RSS](https://es.wikipedia.org/wiki/RSS) o [Atom](https://es.wikipedia.org/wiki/Atom_(formato_de_redifusi%C3%B3n))) que contienen listas de enlaces a otros recursos.
* [Amazon.com](https://es.wikipedia.org/wiki/Amazon.com) ofrece su [interfaz para desarrolladores](https://www.amazon.com/gp/aws/landing.html) tanto en formato REST como en formato [SOAP](https://es.wikipedia.org/wiki/SOAP) (siendo la versión REST la que recibe mayor tráfico).
* [eBay](https://es.wikipedia.org/wiki/EBay) ofreció hasta 2008 una interfaz REST [para desarrolladores](http://developer.ebay.com/rest/).
* El Proyecto ["Seniors Canada On-line"](https://web.archive.org/web/20060428061214/http:/www.seniors.gc.ca/index.jsp) del Gobierno de [Canadá](https://es.wikipedia.org/wiki/Canad%C3%A1) ofrece una interfaz REST [descrito aquí](http://www.megginson.com/blogs/quoderat/archives/2005/03/09/public-rest-application-seniors-canada-online/).
* [Bloglines](https://es.wikipedia.org/wiki/Bloglines) ofrece una API basada en REST [para desarrolladores](https://web.archive.org/web/20051228212538/http:/www.bloglines.com/services/).
* [Yahoo!](https://es.wikipedia.org/wiki/Yahoo!) ofrece una API en REST [para desarrolladores](http://developer.yahoo.com/).
* El mecanismo de enrutamiento de [Ruby on Rails](https://es.wikipedia.org/wiki/Ruby_on_Rails) soporta aplicaciones REST utilizando el [patrón de diseño](https://es.wikipedia.org/wiki/Patr%C3%B3n_de_dise%C3%B1o) [MVC](https://es.wikipedia.org/wiki/Modelo_Vista_Controlador).
* [Microsoft](https://es.wikipedia.org/wiki/Microsoft) tiene su implementación en ADO.NET Data Services Framework (anteriormente conocido como “Astoria”) [[1]](http://msdn.microsoft.com/en-us/library/cc668792.aspx).
* El mismo mecanismo en [Catalyst](http://www.catalystframework.org/) también soporta aplicaciones REST mediante [MVC](https://es.wikipedia.org/wiki/Modelo_Vista_Controlador).
* El publicador de objetos de [Zope](https://es.wikipedia.org/wiki/Zope).
* Implementación REST para Java: [RestLet](http://www.restlet.org/).
* Facebook ofrece una API basada en REST.
* Twitter ofrece una API basada en REST.
* MEGA ofrece una API basada en REST.

que es soap

**SOAP** (originalmente las siglas de *Simple Object Access Protocol*) es un [protocolo](https://es.wikipedia.org/wiki/Protocolo_(inform%C3%A1tica)) [estándar](https://es.wikipedia.org/wiki/Norma_(tecnolog%C3%ADa)) que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos [XML](https://es.wikipedia.org/wiki/XML). Este protocolo deriva de un protocolo creado por [Dave Winer](https://es.wikipedia.org/w/index.php?title=Dave_Winer&action=edit&redlink=1) en 1998, llamado [XML-RPC](https://es.wikipedia.org/wiki/XML-RPC). SOAP fue creado por [Microsoft](https://es.wikipedia.org/wiki/Microsoft), [IBM](https://es.wikipedia.org/wiki/IBM) y otros. Está actualmente bajo el auspicio de la [W3C](https://es.wikipedia.org/wiki/World_Wide_Web_Consortium). Es uno de los protocolos utilizados en los [servicios Web](https://es.wikipedia.org/wiki/Servicios_Web).

SOAP es un **paradigma de mensajería de una dirección sin estado**, que puede ser utilizado para formar protocolos más completos y complejos según las necesidades de las aplicaciones que lo implementan. Puede formar y construir la capa base de una "[pila de protocolos](https://es.wikipedia.org/wiki/Pila_de_protocolos) de web service", ofreciendo un framework de mensajería básica en el cual los web services se pueden construir. Este protocolo está basado en XML y se conforma de tres partes:

* **Sobre (envelope)**: el cual define qué hay en el mensaje y cómo procesarlo.
* **Conjunto de reglas de codificación** para expresar instancias de tipos de datos.
* **La Convención** para representar llamadas a procedimientos y respuestas.

El protocolo SOAP tiene tres características principales:

* **Extensibilidad** (seguridad y WS-routing son extensiones aplicadas en el desarrollo).
* **Neutralidad** (bajo protocolo de transporte [TCP](https://es.wikipedia.org/wiki/Transmission_Control_Protocol) puede ser utilizado sobre cualquier protocolo de aplicación como [HTTP](https://es.wikipedia.org/wiki/HTTP), [SMTP](https://es.wikipedia.org/wiki/SMTP) o [JMS](https://es.wikipedia.org/wiki/JMS)).
* **Independencia** (permite cualquier modelo de programación).

Como ejemplo de cómo el modelo SOAP pueda ser utilizado, consideraremos un mensaje SOAP que podría ser enviado a un web service para realizar la búsqueda de algún precio en una base de datos, indicando para ello los parámetros necesitados en la consulta. El servicio podría retornar un documento en formato XML con el resultado, un ejemplo, precios, localización o características. Teniendo los datos de respuesta en un formato estandarizado procesable (en inglés "parsable"), éste puede ser integrado directamente en un sitio Web o aplicación externa.

La arquitectura SOAP está formada por varias capas de especificación: [MEP](https://es.wikipedia.org/w/index.php?title=Message_Exchange_Patterns&action=edit&redlink=1) (*Message Exchange Patterns*) para el formato del mensaje, enlaces subyacentes del protocolo de transporte, el modelo de procesamiento de mensajes, y la capa de extensibilidad del protocolo. SOAP es el sucesor de [XML-RPC](https://es.wikipedia.org/wiki/XML-RPC), a pesar de que toma el transporte y la neutralidad de la interacción, así como el *envelope* / *header* / *body*, de otros modelos (probablemente de [WDDX](https://es.wikipedia.org/wiki/WDDX)).

**SOAP** originalmente significaba "Simple Object Access Protocol", pero esta sigla se abandonó con la versión 1.2 de la norma. La versión 1.2 se convirtió en una recomendación del [W3C](https://es.wikipedia.org/wiki/W3C) el 24 de junio de 2003. El acrónimo se confunde a veces con [SOA](https://es.wikipedia.org/wiki/Arquitectura_orientada_a_servicios), siglas de arquitectura orientada a servicios, pero las siglas no están relacionados.

Estructura del mensaje[[editar](https://es.wikipedia.org/w/index.php?title=Simple_Object_Access_Protocol&action=edit&section=3" \o "Editar sección: Estructura del mensaje)]

Un mensaje SOAP es un documento XML ordinario con una estructura definida en la especificación del protocolo. Dicha estructura la conforman las siguientes partes:

* **Envelope (obligatoria)**: raíz que de la estructura, es la parte que identifica al mensaje SOAP como tal.
* **Header**: esta parte es un mecanismo de extensión ya que permite enviar información relativa a cómo debe ser procesado el mensaje. Es una herramienta para que los mensajes puedan ser enviados de la forma más conveniente para las aplicaciones. El elemento "Header" se compone a su vez de **"Header Blocks"** que delimitan las unidades de información necesarias para el header.
* **Body (obligatoria)**: contiene la información relativa a la llamada y la respuesta.
* **Fault**: bloque que contiene información relativa a errores que se hayan producido durante el procesado del mensaje y el envío desde el "SOAP Sender" hasta el "Ultimate SOAP Receiver".

En los próximos apartados de este documento se podrá apreciar esta estructura con ejemplos concretos.

## SOAP sobre correo electrónico[[editar](https://es.wikipedia.org/w/index.php?title=Simple_Object_Access_Protocol&action=edit&section=5" \o "Editar sección: SOAP sobre correo electrónico)]

Los desarrolladores de aplicaciones hoy en día, pueden utilizar la infraestructura de correo electrónico de Internet para transmitir mensajes SOAP ya sea como mensajes de correo electrónico de texto o como adjuntos. Los ejemplos que se muestran a continuación muestran un modo de transmitir mensajes SOAP, y deben ser tomados como el modo estándar de hacerlo. Las especificaciones SOAP Versión 1.2 no especifican tal vínculo. Sin embargo, existe una Nota W3C no-normativa [SOAP Email Binding] que describe un vínculo de SOAP con el correo electrónico. Su propósito principal es comenzar a demostrar la aplicación de la Infraestructura general de Vínculos con el Protocolo SOAP.

Ventajas y desventajas[[editar](https://es.wikipedia.org/w/index.php?title=Simple_Object_Access_Protocol&action=edit&section=7" \o "Editar sección: Ventajas y desventajas)]

**Ventajas**[[editar](https://es.wikipedia.org/w/index.php?title=Simple_Object_Access_Protocol&action=edit&section=8" \o "Editar sección: Ventajas)]

* Debido al uso de XML permite invocar procedimientos remotos de muchos lenguajes, por lo tanto, presenta una gran interoperabilidad.
* Al utilizar una comunicación vía [HTTP](https://es.wikipedia.org/wiki/HTTP) es fácilmente escalable, además de ser casi siempre permitido por los cortafuegos.
* Puede ser implementado utilizando cualquier lenguaje y ejecutado en cualquier plataforma.
* Es posible utilizarlo mediante usuario anónimo y mediante autentificación.
* Es posible transmitirlo mediante cualquier protocolo de transporte capaz de transmitir texto, típicamente [HTTP](https://es.wikipedia.org/wiki/HTTP) o [SMTP](https://es.wikipedia.org/wiki/SMTP).

**Desventajas**[[editar](https://es.wikipedia.org/w/index.php?title=Simple_Object_Access_Protocol&action=edit&section=9" \o "Editar sección: Desventajas)]

* Debido al uso de XML para el paso de mensajes, SOAP es considerablemente más lento que otros middleware como [CORBA](https://es.wikipedia.org/wiki/CORBA) ya que los datos binarios se codifican como texto. Para contrarrestar este punto débil en el caso de XML con código binario incrustado se desarrolló un método optimizado de transmisión de mensajes.
* Depende del [WSDL](https://es.wikipedia.org/wiki/WSDL) (Web Services Description Language).
* Al contrario que Java, PHP o Python ciertos lenguajes no ofrecen un apoyo adecuado para su uso ya sea a nivel de integración o de soporte [IDE](https://es.wikipedia.org/wiki/Entorno_de_desarrollo_integrado).
* **JAVA**: dentro de su librería estándar se encuentran implementaciones concretas a las que se da **soporte oficial**. También podemos encontrar librerías de terceros que, tal y como se ha comentado, ayudan al desarrollador simplificando las interfaces e implementando los casos de uso más habituales. Cabe destacar que los **IDEs más utilizados ofrecen soporte para la creación de servicios web SOAP** que, entre otras cosas, **generan automáticamente el fichero WSDL** y permiten diseñar de forma visual el API y las llamadas que contendrá. En cuanto el servidor a utilizar, se pueden considerar las opciones típicas en Java: Tomcat, Glassfish, etc. Aun así, la elección del servidor puede suponer algunas ventajas, por ejemplo, Glassfish genera una sencilla interfaz web para probar las diferentes llamadas del servicio. Además, la mayoría de herramientas permiten la [generación del cliente del servicio automáticamente a partir de su fichero WSDL](https://netbeans.org/kb/docs/websvc/jax-ws.html).
* **PHP**: **ofrece soporte y unas librerías de apoyo habilitando la**[**extensión SOAP**](http://www.php.net/manual/es/book.soap.php) en el servidor. Se ha desarrollado un [gran número de librerías de terceros](https://packagist.org/search/?q=SOAP), que combinadas con el uso de frameworks MVC, simplifican las interfaces e implementan los escenarios de uso más habituales. También son habituales las implementaciones de clientes para servicios web públicos concretos.
* **Python**: no ofrece un soporte en sus librerías estándar, sin embargo, existe un gran número de paquetes de terceros que permiten la implementación de servicios web SOAP y sus clientes. En el ámbito del desarrollo de servicios web en Python, predomina la utilización del [**Framework Django**](https://www.djangoproject.com/) que se puede combinar con cualquiera de las implementaciones de SOAP.
* **.NET**: dentro del Framework se ofrecen herramientas similares a las de Java para el diseño visual del servicio y la creación automática de WSDL . También da soporte para la creación de los clientes a partir del fichero de definición del servicio. En el caso de .NET, el IDE destacado es Visual Studio. En cuanto a librerías encontramos que el ecosistema .NET ofrece múltiples opciones en varios lenguajes, aunque la apuesta actual de Microsoft para el desarrollo web es su [Framework .NET MVC](https://es.wikipedia.org/wiki/ASP.NET_MVC_Framework). Se debe tener en cuenta, que Microsoft creó el formato [**Windows Communication Foundation**](http://msdn.microsoft.com/es-es/vstudio/aa663324.aspx) que es un modelo para la creación de sistemas orientados a servicios, similar y complementario al WSDL.

Casos de uso[[editar](https://es.wikipedia.org/w/index.php?title=Simple_Object_Access_Protocol&action=edit&section=10" \o "Editar sección: Casos de uso)]

La forma más habitual de utilizar el protocolo SOAP es mediante el patrón **petición-respuesta** con remitente SOAP y destinatario final SOAP, el cual es utilizado cuando los mensajes SOAP están predefinidos y únicamente se desea enviar una petición y consultar su valor de retorno.

No obstante, muchas veces este patrón no es suficiente, y es necesario establecer un intercambio múltiple de mensajes entre los nodos. La W3C define dos tipos de intercambios de mensajes SOAP para formar una conversación:

* **Intercambio de mensajes Conversacionales**: permite redefinir la información de la petición. Estos intercambios pueden acabar comportándose como un patrón de mensajes de ida y vuelta.
* **Llamadas a Procedimientos Remotos**: permite encapsular la funcionalidad de procedimientos remotos utilizando las ventajas de XML de extensibilidad y funcionalidad, por este motivo se ha definido en la especificación una representación uniforme para realizar invocaciones y respuestas RPC mediante mensajes SOAP.

En ocasiones, es necesario el uso de intermediarios en las comunicaciones SOAP, la especificación SOAP 1.2 define dos tipos:

* **Intermediario redirector**: se trata de un nodo SOAP, el cual redirige el mensaje SOAP a otro nodo SOAP según lo establecido en un bloque de encabezado que ha recibido el nodo destino o según el patrón de mensajes en uso.
* **Intermediario activo**: realiza un procesamiento adicional del mensaje SOAP antes de redirigirlo, sin utilizar criterios descritos en el encabezado del mensaje o del patrón de mensajes en uso.

Que es Restful

Aunque muchas veces se usan los términos Rest y Restful como sinónimos, no lo son. Rest (Representational State Transfer), es un modelo de arquitectura web basado en el protocolo HTTP para mejorar las comunicaciones cliente-servidor, mientras que Restful Web Service o Restful API son programas basados en REST. Su principal diferencia es que la API no necesita ejecutar una red, sino que es posible hacerlo desde un mismo ordenador.

**Rest sigue una serie de principios:**

1. Todo lo que se mueve a través de las comunicaciones web son recursos, es decir, los datos se representan con el formato específico que tienen y no como un archivo físico.
2. Cada uno de estos recursos ha de tener un identificador único, el cual se representa a través de su URL, puesto que hay una infinidad de recursos en la web y han de ser accesibles e identificables.
3. Este protocolo de transmisión de datos debe utilizar los verbos estándares de HTTP, definidos en el protocolo nativo, donde cada uno de estos verbos significa una acción diferente. Hay 8 acciones principales ya definidas: GET, POST, PUT, DELETE, HEAD, OPTIONS, TRACE, CONNECT.
4. Cada recurso puede tener múltiples representaciones, independientemente de cómo esté almacenado. Un ejemplo sería un recurso en formato XML y poder solicitarlo en JSON.
5. Comunicaciones cliente-servidor, se trata de comunicaciones que se denominan sin estado (STATELESS), lo que significa que cada petición al servidor es tratada de manera totalmente independiente.

Por lo tanto, **Rest** es un **concepto de arquitectura** (en cambio **Restful**, son los **Webs Services)** que siguen los siguientes principios:

**REST** (Representational State Transfer) es una arquitectura que se ejecuta sobre HTTP.

**RESTful** hace referencia a un servicio web que implementa la arquitectura REST.

En un ejemplo bastante básico (de crear un aplicación RESTful), tenemos un proyecto donde implementamos un CRUD para casas en PHP y con la architectura MVC, el resultado de las respuestas serán devueltas en JSON, más no es obligatorio manejar sólo ese tipo Content-type, se puede devolver HTML, text, etc...

RESTful se suele utilizar para referirse a los servicios web que ejecutan la arquitectura REST.

3 que es bioinformática

la aplicación de tecnologías computacionales y la estadística a la gestión y análisis de [datos](https://es.wikipedia.org/wiki/Dato) [biológicos](https://es.wikipedia.org/wiki/Biolog%C3%ADa).[1](https://es.wikipedia.org/wiki/Bioinform%C3%A1tica#cite_note-EBIdef-1)​ Los términos **bioinformática**, [**biología computacional**](https://es.wikipedia.org/wiki/Biolog%C3%ADa_computacional), **informática biológica** y, en ocasiones, **biocomputación**, son utilizados en muchas situaciones como [sinónimos](https://es.wikipedia.org/wiki/Sin%C3%B3nimo),[2](https://es.wikipedia.org/wiki/Bioinform%C3%A1tica#cite_note-2)​[3](https://es.wikipedia.org/wiki/Bioinform%C3%A1tica#cite_note-3)​ y hacen referencia a campos de estudios interdisciplinares muy vinculados que requieren el uso o el desarrollo de diferentes técnicas estudiadas universitariamente en la [Ingeniería Informática](https://es.wikipedia.org/wiki/Ingenieria_Informatica) como ciencia aplicada de la disciplina [informática](https://es.wikipedia.org/wiki/Inform%C3%A1tica).[4](https://es.wikipedia.org/wiki/Bioinform%C3%A1tica#cite_note-Bajic-4)​ Entre estas pueden destacarse las siguientes: [matemática aplicada](https://es.wikipedia.org/wiki/Matem%C3%A1tica_aplicada),[5](https://es.wikipedia.org/wiki/Bioinform%C3%A1tica#cite_note-Lander-5)​ [estadística](https://es.wikipedia.org/wiki/Estad%C3%ADstica),[6](https://es.wikipedia.org/wiki/Bioinform%C3%A1tica#cite_note-Woon-6)​ [ciencias de la computación](https://es.wikipedia.org/wiki/Ciencias_de_la_computaci%C3%B3n),[7](https://es.wikipedia.org/wiki/Bioinform%C3%A1tica#cite_note-7)​ [inteligencia artificial](https://es.wikipedia.org/wiki/Inteligencia_artificial),[8](https://es.wikipedia.org/wiki/Bioinform%C3%A1tica#cite_note-Frasconi-8)​ [química](https://es.wikipedia.org/wiki/Qu%C3%ADmica)[9](https://es.wikipedia.org/wiki/Bioinform%C3%A1tica#cite_note-Murray-Rust-9)​ y [bioquímica](https://es.wikipedia.org/wiki/Bioqu%C3%ADmica)[10](https://es.wikipedia.org/wiki/Bioinform%C3%A1tica#cite_note-Ibba-10)​ con las que el Ingeniero Informático soluciona problemas al [analizar datos](https://es.wikipedia.org/wiki/An%C3%A1lisis_de_datos), o [simular](https://es.wikipedia.org/wiki/Simulaci%C3%B3n) [sistemas](https://es.wikipedia.org/wiki/Sistema) o mecanismos, todos ellos de índole [biológica](https://es.wikipedia.org/wiki/Biolog%C3%ADa), y usualmente (pero no de forma exclusiva) en el nivel molecular.[11](https://es.wikipedia.org/wiki/Bioinform%C3%A1tica#cite_note-Altman-11)​ El núcleo principal de estas técnicas se encuentra en la utilización de recursos computacionales para solucionar o investigar problemas sobre escalas de tal magnitud que sobrepasan el discernimiento humano. La investigación en biología computacional se solapa a menudo con la [biología de sistemas](https://es.wikipedia.org/wiki/Biolog%C3%ADa_de_sistemas).[12](https://es.wikipedia.org/wiki/Bioinform%C3%A1tica#cite_note-Werner-12)​  
  
Los principales esfuerzos de investigación en estos campos incluyen el [alineamiento de secuencias](https://es.wikipedia.org/wiki/Alineamiento_de_secuencias), la [predicción de genes](https://es.wikipedia.org/wiki/Predicci%C3%B3n_de_genes), [montaje](https://es.wikipedia.org/wiki/Montaje_de_secuencias) del [genoma](https://es.wikipedia.org/wiki/Genoma), [alineamiento estructural de proteínas](https://es.wikipedia.org/wiki/Alineamiento_estructural), [predicción de estructura de proteínas](https://es.wikipedia.org/wiki/Predicci%C3%B3n_de_estructura_de_prote%C3%ADnas), predicción de la [expresión génica](https://es.wikipedia.org/wiki/Expresi%C3%B3n_g%C3%A9nica), [interacciones proteína-proteína](https://es.wikipedia.org/wiki/Interacciones_prote%C3%ADna-prote%C3%ADna), y modelado de la [evolución](https://es.wikipedia.org/wiki/Evoluci%C3%B3n_biol%C3%B3gica).[13](https://es.wikipedia.org/wiki/Bioinform%C3%A1tica#cite_note-Kanehisa-13)​

Una constante en proyectos de bioinformática y biología computacional es el uso de herramientas matemáticas para extraer [información](https://es.wikipedia.org/wiki/Informaci%C3%B3n) útil de [datos](https://es.wikipedia.org/wiki/Dato) producidos por técnicas biológicas de alta productividad, como la [secuenciación del genoma](https://es.wikipedia.org/wiki/Secuenciaci%C3%B3n_de_ADN). En particular, el montaje o ensamblado de secuencias genómicas de alta calidad desde fragmentos obtenidos tras la [secuenciación del ADN a gran escala](https://es.wikipedia.org/wiki/Secuenciaci%C3%B3n_de_ADN#Estrategias_de_secuenciaci%C3%B3n_a_gran_escala) es un área de alto interés.[13](https://es.wikipedia.org/wiki/Bioinform%C3%A1tica#cite_note-Kanehisa-13)​[14](https://es.wikipedia.org/wiki/Bioinform%C3%A1tica#cite_note-Primer-14)​ Otros objetivos incluyen el estudio de la [regulación genética](https://es.wikipedia.org/wiki/Regulaci%C3%B3n_gen%C3%A9tica) para interpretar [perfiles de expresión génica](https://es.wikipedia.org/wiki/Perfil_de_expresi%C3%B3n_g%C3%A9nica) utilizando datos de [chips de ADN](https://es.wikipedia.org/wiki/Chip_de_ADN) o [espectrometría de masas](https://es.wikipedia.org/wiki/Espectrometr%C3%ADa_de_masas).[15](https://es.wikipedia.org/wiki/Bioinform%C3%A1tica#cite_note-Genevestigator-15)​

Principales áreas de investigación[[editar](https://es.wikipedia.org/w/index.php?title=Bioinform%C3%A1tica&action=edit&section=7" \o "Editar sección: Principales áreas de investigación)]

**Análisis de secuencias**[[editar](https://es.wikipedia.org/w/index.php?title=Bioinform%C3%A1tica&action=edit&section=8" \o "Editar sección: Análisis de secuencias)]

***Artículo principal:***[**Base de datos biológica**](https://es.wikipedia.org/wiki/Base_de_datos_biol%C3%B3gica)

***Artículo principal:***[**Alineamiento de secuencias**](https://es.wikipedia.org/wiki/Alineamiento_de_secuencias)La secuenciación conocida como "*shotgun*" (o *por perdigonada*: fue usada, por ejemplo, por el Instituto de Investigación Genómica -*The Institute for Genomic Research*, TIGR, hoy *J. Craig Venter Institute*- para secuenciar el primer genoma de bacteria, el [Haemophilus influenzae](https://es.wikipedia.org/wiki/Haemophilus_influenzae))[62](https://es.wikipedia.org/wiki/Bioinform%C3%A1tica#cite_note-Fleischmann-62)​ no da una lista secuencial de nucleótidos, pero en cambio nos ofrece las secuencias de miles de pequeños fragmentos de ADN (cada uno de aproximadamente 600 a 800 nucleótidos de largo). [Las terminaciones de estos fragmentos se superponen](https://es.wikipedia.org/wiki/Montaje_de_secuencias) y, cuando son alineados de la manera correcta, constituyen el genoma completo del organismo en cuestión.[97](https://es.wikipedia.org/wiki/Bioinform%C3%A1tica#cite_note-Pop_2004-97)​

**Anotación de genomas**[[editar](https://es.wikipedia.org/w/index.php?title=Bioinform%C3%A1tica&action=edit&section=9" \o "Editar sección: Anotación de genomas)]

***Artículo principal:***[**Predicción de genes**](https://es.wikipedia.org/wiki/Predicci%C3%B3n_de_genes)

El primer sistema software de anotación de genomas fue diseñado en 1995 por Owen White, quien fue miembro del equipo que secuenció y analizó el primer genoma en ser descodificado de un organismo independiente, la bacteria [Haemophilus influenzae](https://es.wikipedia.org/wiki/Haemophilus_influenzae). White construyó un software para localizar los genes (lugares en la secuencia de DNA que codifican una proteína), el [ARN de transferencia](https://es.wikipedia.org/wiki/ARN_de_transferencia), y otras características, así como para realizar las primeras atribuciones de función a esos genes.

### Biología evolutiva computacional[[editar](https://es.wikipedia.org/w/index.php?title=Bioinform%C3%A1tica&action=edit&section=10" \o "Editar sección: Biología evolutiva computacional)]

La [Biología evolutiva](https://es.wikipedia.org/wiki/Biolog%C3%ADa_evolutiva) es el estudio del origen ancestral de las [especies](https://es.wikipedia.org/wiki/Especies), así como de su cambio a través del tiempo.[101](https://es.wikipedia.org/wiki/Bioinform%C3%A1tica#cite_note-Futuyma-101)​ La informática ha apoyado a los biólogos evolutivos en diferentes campos clave. Ha permitido a los investigadores:

* Seguir la [evolución](https://es.wikipedia.org/wiki/Evoluci%C3%B3n_biol%C3%B3gica) de un alto número de organismos midiendo cambios en su [ADN](https://es.wikipedia.org/wiki/ADN), en lugar de hacerlo exclusivamente mediante su [taxonomía](https://es.wikipedia.org/wiki/Taxonom%C3%ADa) física u observaciones [fisiológicas](https://es.wikipedia.org/wiki/Fisiolog%C3%ADa).[39](https://es.wikipedia.org/wiki/Bioinform%C3%A1tica#cite_note-Doolittle_1981-39)​
* Más recientemente, comparar genomas completos, lo que permite el estudio de eventos evolutivos más complejos, tales como la [duplicación de genes](https://es.wikipedia.org/w/index.php?title=Duplicaci%C3%B3n_de_genes&action=edit&redlink=1), la [transferencia horizontal de genes](https://es.wikipedia.org/wiki/Transferencia_horizontal_de_genes), o la predicción de factores significativos en la [especiación](https://es.wikipedia.org/wiki/Especiaci%C3%B3n) bacteriana.[102](https://es.wikipedia.org/wiki/Bioinform%C3%A1tica#cite_note-Comparative_Genomics-102)​
* Construir modelos computacionales complejos de poblaciones para predecir el resultado del sistema a través del tiempo.[103](https://es.wikipedia.org/wiki/Bioinform%C3%A1tica#cite_note-Levin_1997-103)​
* Seguir y compartir información sobre un amplio y creciente número de especies y organismos.

**Servicios web en bioinformática**[[editar](https://es.wikipedia.org/w/index.php?title=Bioinform%C3%A1tica&action=edit&section=23" \o "Editar sección: Servicios web en bioinformática)]

Se han desarrollado [interfaces](https://es.wikipedia.org/wiki/Interfaz_de_usuario) basadas en [SOAP](https://es.wikipedia.org/wiki/SOAP) y en [REST](https://es.wikipedia.org/wiki/REST) (*Representational State Transfer*, transferencia de estado representacional) para una amplia variedad de aplicaciones bioinformáticas, permitiendo que una aplicación, corriendo en un ordenador de cualquier parte del mundo, pueda usar algoritmos, datos y recursos de computación alojados en [servidores](https://es.wikipedia.org/wiki/Servidor) en cualesquiera otras partes del planeta. Las principales ventajas radican en que el usuario final se despreocupa de actualizaciones y modificaciones en el software o en las bases de datos.[143](https://es.wikipedia.org/wiki/Bioinform%C3%A1tica#cite_note-EBIwebserv-143)​ Los servicios bioinformáticos básicos, de acuerdo a la clasificación implícita del [EBI](https://es.wikipedia.org/wiki/Instituto_Europeo_de_Bioinform%C3%A1tica), solían clasificarse en:[144](https://es.wikipedia.org/wiki/Bioinform%C3%A1tica#cite_note-EBIbasicWS-144)​

* Servicios de obtención de información en línea (consultas a [bases de datos](https://es.wikipedia.org/wiki/Base_de_datos_biol%C3%B3gica), por ejemplo).
* Herramientas de análisis (por ejemplo, servicios que den acceso a [EMBOSS](https://es.wikipedia.org/wiki/EMBOSS)).
* Búsquedas de [similitudes entre secuencias](https://es.wikipedia.org/wiki/Alineamiento_de_secuencias) (servicios de acceso a [FASTA](https://es.wikipedia.org/wiki/FASTA) o [BLAST](https://es.wikipedia.org/wiki/BLAST), por ejemplo).
* [Alineamientos múltiples de secuencias](https://es.wikipedia.org/wiki/Alineamiento_m%C3%BAltiple_de_secuencias) (acceso a [ClustalW](https://es.wikipedia.org/wiki/ClustalW) o [T-Coffee](https://es.wikipedia.org/wiki/T-Coffee)).
* Análisis estructural (acceso a servicios de [alineamiento estructural](https://es.wikipedia.org/wiki/Alineamiento_estructural) de proteínas, por ejemplo).
* Servicios de acceso a literatura especializada y [ontologías](https://es.wikipedia.org/wiki/Ontolog%C3%ADa_(inform%C3%A1tica)).

Desde 2009 los servicios bioinformaticos básicos son clasificados por el [EBI](https://es.wikipedia.org/wiki/European_Bioinformatics_Institute) en tres categorías[145](https://es.wikipedia.org/wiki/Bioinform%C3%A1tica#cite_note-145)​ :

* [similitudes entre secuencias](https://es.wikipedia.org/wiki/Alineamiento_de_secuencias) (SSS)
* [Alineamientos múltiples de secuencias](https://es.wikipedia.org/wiki/Alineamiento_m%C3%BAltiple_de_secuencias) (MSA)
* [análisis de secuencia bioinformático](https://es.wikipedia.org/w/index.php?title=An%C3%A1lisis_de_secuencia_bioinform%C3%A1tico&action=edit&redlink=1) (BSA)

### Software libre en bioinformática[[editar](https://es.wikipedia.org/w/index.php?title=Bioinform%C3%A1tica&action=edit&section=22" \o "Editar sección: Software libre en bioinformática)]

Muchas herramientas de [software libre](https://es.wikipedia.org/wiki/Software_libre) existen y continúan apareciendo desde los década de 1980s.[140](https://es.wikipedia.org/wiki/Bioinform%C3%A1tica#cite_note-obf-main-140)​ La necesidad de nuevos [algoritmos](https://es.wikipedia.org/wiki/Algoritmos) para el análisis the nuevos datos de origen biológico, en combinación con el potencial para experimentos innovadores [*in silico*](https://es.wikipedia.org/wiki/In_silico) y la disponibilidad de [repositorios](https://es.wikipedia.org/wiki/Repositorios) gratuitos para [software libre](https://es.wikipedia.org/wiki/Software_libre) han ayudado a crear oportunidades para que grupos de investigación realicen aportes a la bioinformatica y al [código libre](https://es.wikipedia.org/w/index.php?title=C%C3%B3digo_libre&action=edit&redlink=1) disponible, independientemente de sus fuentes de financiamiento. Las herramientas de código abierto a menudo actúan como incubadoras de ideas, o como [complemento](https://es.wikipedia.org/w/index.php?title=Complemento_(computaci%C3%B3n)&action=edit&redlink=1) en aplicaciones comerciales. Pueden también proveer estándares [*de facto*](https://es.wikipedia.org/wiki/De_facto) y modelos o estructuras que aportan al desafío de la integración en bioinformática.

4 que son variables de entorno, ventajas y usos con ejemplos

Una **variable de entorno** es una variable dinámica que puede afectar al comportamiento de los procesos en ejecución en un ordenador.

Son parte del entorno en el que se ejecuta un proceso. Por ejemplo, un proceso en ejecución puede consultar el valor de la variable de entorno TEMP para descubrir una ubicación adecuada para almacenar archivos temporales, o la variable HOME o USERPROFILE para encontrar la estructura de directorios propiedad del usuario que ejecuta el proceso.

Fueron introducidos en su forma moderna en 1979 con la versión 7 de Unix, por lo que están incluidos en todos los sabores y variantes del sistema operativo Unix a partir de ese momento, incluyendo Linux y macOS. Desde PC DOS 2.0 en 1982, todos los sistemas operativos de Microsoft, incluyendo Microsoft Windows y OS/2, también los han incluido como una característica, aunque con sintaxis, uso y nombres de variables estándar algo diferentes.

Es posible acceder al valor de una variable, ya sea tanto para consultarlo como para modificarlo, tanto dentro de [scripts](https://es.wikipedia.org/wiki/Script) como desde la [línea de comandos](https://es.wikipedia.org/wiki/L%C3%ADnea_de_comandos). La forma en que se accede al contenido de una variable varía para cada [sistema operativo](https://es.wikipedia.org/wiki/Sistema_operativo). En el caso de sistemas [Unix](https://es.wikipedia.org/wiki/Unix), depende del [intérprete de comandos](https://es.wikipedia.org/wiki/Int%C3%A9rprete_de_comandos) (o *shell*) que se use, pues, este es el encargado del manejo de las variables de entorno.

### Windows[[editar](https://es.wikipedia.org/w/index.php?title=Variable_de_entorno&action=edit&section=5" \o "Editar sección: Windows)]

**%COMSPEC%**

Esta variable contiene la ruta completa al procesador de comandos, cmd.exe

**%PATH%**

Esta variable contiene una lista separada por [punto y comas](https://es.wikipedia.org/wiki/Punto_y_coma) de directorios en los cuales el intérprete de comandos buscará los [archivos ejecutables](https://es.wikipedia.org/wiki/Archivo_ejecutable) que no se invocan con una ruta explícita. Nótese que es posible asignarle valores a PATH con el comando homónimo, sin necesidad de utilizar SET. Por ejemplo, para poder invocar el navegador firefox desde cualquier directorio:

#### Valores por defecto de las variables de entorno**[[editar](https://es.wikipedia.org/w/index.php?title=Variable_de_entorno&action=edit&section=6" \o "Editar sección: Valores por defecto de las variables de entorno)]**

|  |  |  |
| --- | --- | --- |
| **Variable** | **Windows XP** | **Windows Vista/7** |
| %ALLUSERSPROFILE% (%PROGRAMDATA%) | C:\Documents and Settings\All Users | C:\ProgramData |
| %APPDATA% | C:\Documents and Settings\{username}\Datos de Programa | C:\Users\{username}\AppData\Roaming |
| %COMPUTERNAME% | {nombredeordenador} | {nombredeordenador} |
| %COMMONPROGRAMFILES% | C:\Archivos de programa\Archivos Comunes | C:\Archivos de programa\Archivos Comunes |
| %COMMONPROGRAMFILES(x86)% | C:\Archivos de programa (x86)\Archivos Comunes | C:\Archivos de programa (x86)\Archivos Comunes |
| %COMSPEC% | C:\Windows\System32\cmd.exe | C:\Windows\System32\cmd.exe |
| %HOMEDRIVE% | C:\ | C:\ |
| %HOMEPATH% | C:\Documents and Settings\{username} | \Usuarios\{username} |
| %LOCALAPPDATA% | No disponible si es una declaración explícita. | C:\Usuarios\{username}\Application Data\Local |
| %LOGONSERVER% | \\{domain\_logon\_server} | \\{domain\_logon\_server} |
| %PATH% | C:\Windows\system32, C:\Windows, C:\Windows\System32\Wbem | C:\Windows\system32, C:\Windows, C:\Windows\System32\Wbem |
| %PATHEXT% | .COM, .EXE, .BAT, .CMD, .VBS, .VBE, .JS, .WSF, .WSH | .com, .exe, .bat, .cmd, .vbs, .vbe, .js, .jse, .wsf, .wsh, .msc |
| %PROGRAMFILES% | C:\Archivos de programa\ | C:\Archivos de programa\ |
| %PROGRAMFILES(X86)% | C:\Archivos de programa (x86) (solo en versiones 64-bit) | C:\Archivos de programa (x86) (solo en versiones 64-bit) |
| %PROMPT% | Código del formato de [Cmd.exe](https://es.wikipedia.org/wiki/Cmd.exe). El código es normalmente $P$G | Código del formato de [Cmd.exe](https://es.wikipedia.org/wiki/Cmd.exe). El código es normalmente $P$G |
| %SystemDrive% | C:\ | C:\ |
| %SystemRoot% | El directorio de [Windows](https://es.wikipedia.org/wiki/Windows), normalmente C:\Windows; anteriormente C:\WINNT | C:\Windows |
| %TEMP% y %TMP% | C:\Documents and Settings\{username}\Local Settings\Temp | C:\Users\{username}\AppData\Local\Temp |
| %USERDOMAIN% | {userdomain} | {userdomain} |
| %USERNAME% | {username} | {username} |
| %USERPROFILE% | %SystemDrive%\Documents and Settings\{username} | C:\Users\{username} |
| %WINDIR% | C:\Windows | C:\Windows |
| %PUBLIC% |  | C:\Users\Public |
| %PSModulePath% |  | %SystemRoot%\system32\WindowsPowe |

echo %PATH%

Observaciones: El comando SET, sin argumentos, muestra todas las variables de entorno junto con sus respectivos valores.

Para asignar un valor a una variable se usa:[1](https://es.wikipedia.org/wiki/Variable_de_entorno#cite_note-1)​

SET *VARIABLE*=*valor*

#### Pseudo-variables**[[editar](https://es.wikipedia.org/w/index.php?title=Variable_de_entorno&action=edit&section=7" \o "Editar sección: Pseudo-variables)]**

No almacenan un valor fijo, sino que se evalúan al momento de pedir su valor.

**%CD%**

Muestra la cadena del directorio actual.

**%DATE%**

Muestra la fecha actual usando el mismo formato que el comando DATE.

**%TIME%**

Muestra la hora actual usando el mismo formato que el comando TIME.

**%RANDOM%**

Muestra un número decimal aleatorio entre 0 y 32767.

**%ERRORLEVEL%**

Muestra el valor de NIVEL DE ERROR actual (normalmente, el código de retorno del último comando externo ejecutado).

**%CMDEXTVERSION%**

Muestra el número de versión de las extensiones del procesador de comandos.

**%CMDCMDLINE%**

Muestra la línea de comando original que invocó el procesador de comandos.

///////

Una variable de entorno es, para que todos podamos entenderlo, **una palabra, un texto fácilmente recordable que nos permitirá acceder a rutas más complejas de forma mucho más sencilla.**

Para que podamos hacernos una idea, mi equipo dispone de una variable de entorno llamada **Path** que hace referencia a la ruta **C:/ProgramData/Oracle/Java/javapath;%SystemRoot%/system32;%SystemRoot%** **;%SystemRoot%/System32/Wbem;%SYSTEMROOT%/System32/WindowsPowerShell/v1.0/** **;C:/Program Files (x86)/ATI Technologies/ATI.ACE/Core-Static** por lo que es fácilmente comprensible que es preferible trabajar con una única palabra que no con esa enorme ruta.

Además de simplificar el uso de rutas enormes, estas variables son muy útiles en sistemas operativo multi usuario, ya que **nos permitirán llegar a todos los usuarios de la misma manera, olvidándonos del nombre de usuario concreto de cada persona**, por ejemplo, la variable **%USERNAME%** recogerá cual es el nombre de usuario que está activo en este momento, por lo que a alguna aplicación que necesite ese dato, ya lo tiene ahí recogido y además **cambiará en función de qué usuario esté trabajando en este momento.**

Otro ejemplo es la variable **%TMP%, que mostrará la ruta completa de archivos temporales del usuario** en cuestión.

¿Donde se encuentran las variables de entorno?

**Es muy fácil localizar las variables de entorno de Windows**, ya que para acceder a ellas, bastará con seguir los siguientes pasos:

1. **Pulsa con el botón derecho en Equipo o Mi PC.**
2. **Pulsa en Propiedades.**
3. **Pulsa en configuración avanzada del sistema.**
4. **Pulsa en Variables de entorno…**

Ahí, en el punto cuatro, verás **dos cuadros de texto**, en el primero se muestran las **variables que son únicas para el usuario** con el que estás actualmente trabajando y en segundo cuadro verás las **variables que son de sistema**, es decir, las que estarán vigentes independientemente al usuario con el que se esté trabajando.

5 que es docker y como se usa

Docker se define como un **proyecto de código abierto que proporciona una capa de abstracción y virtualización a nivel de sistema operativo, a través de la instalación de contenedores de software**.

Seguramente os habréis quedado como estabais. Tranquilos, es normal. Lo mejor forma de explicar qué es Docker y para qué es útil es **ilustrarlo con una situación que, seguramente, os habrá pasado unas cuantas veces**. A ver si os suena.

Imagina que quieres instalar en tu ordenador/servidor una nueva tecnología como, por ejemplo, una nueva aplicación Web self-hosted que acabas de descubrir y quieres probar, o un servicio como un broker MQQT que quieres instalar en tu servidor.

Para instalarla tienes que **instalar y configurar un montón de "otras cosas"**. Por ejemplo, instalar un servidor Web, NodeJs con 20 módulos, una base de datos MariaDb, crear usuarios y password para cada cosa, añadir variables de sistema, tocar tropecientos ficheros de configuración, y trasquilar una cabra virgen sobre un pentateuco durante luna creciente.

¿Os suena? La cosa es mucho peor aun cuando tienes que instalar múltiples tecnologías en el mismo ordenador. Una base de datos se pega con la otra, este servicio con aquel y el servidor web con el de más allá. **A medida que tenemos más cosas instaladas en la misma máquina todo empieza a complicarse cada vez más**.

¡Pero aún podría empeorar! Cuando quieres desinstalar algo tienes que jugar a no pasar la delgada línea roja de, o bien quitar de menos e ir llenando progresivamente nuestro ordenador de basura, o quitar de más y que te deje de funcionar cosas que no querías que dejarán de funcionar.

Muy divertido ¿verdad? Sobre todo si lo único que querías era probar una nueva tecnología para ver cómo es.

Esta es la situación que intenta resolver Docker. Docker trabaja con imágenes y contenedores. De forma resumida, **una imagen es un paquete que incluye "todo lo que necesita" una determinada tecnología para funcionar**.

Las imágenes se inicializan en **una o varias instancias que se denominan contenedores**. De esta forma solo tienes que crear el contenedor y a funcionar. Por supuesto, podemos incluso tener varios contenedores de una misma imagen ejecutándose simultáneamente.

Muchos desarrolladores **ofrecen sus aplicaciones como un contenedor**. Algunas de ellas están en <https://hub.docker.com/>, el repositorio de Docker. En otras ocasiones los desarrolladores alojan sus docker en su propia página web, o en github, entre otros.

Virtualizar con Docker nos ofrece una serie de ventajas respecto a hacerlo con máquinas virtuales convencionales:

* **Portabilidad**. Todos los contenedores son portables, por lo que podemos llevarlos fácilmente a cualquier otro equipo con Docker sin tener que volver a configurar nada.
* **Ligereza**. Al no virtualizar un sistema completo, sino solo lo necesario, el consumo de recursos es mínimo. El ahorro de recursos es de alrededor de un 80%.
* **Autosuficiencia**. Docker se encarga de todo, por lo que los contenedores tan solo deben tener lo necesario para que la aplicación funcione, por ejemplo, aquellas librerías, archivos y configuraciones necesarias para poder realizar su función.

Un sistema de contenedores Docker se compone principalmente de 5 elementos:

* **Demonio**: Es el proceso principal de la plataforma.
* **Cliente**: Es el binario que constituye la interfaz y que permite al usuario interactuar con el Demonio.
* **Imagen**: Plantilla utilizada para crear el contenedor para la aplicación que queremos ejecutar.
* **Registros**: Directorios donde se almacenan las imágenes, tanto de acceso público como privado.
* **Contenedores**: Carpetas donde se almacena todo lo necesario (librerías, dependencias, binarios, etc) para que la aplicación pueda ejecutarse de forma aislada.

Docker también cuenta con una serie de **repositorios**, similares a los de Linux, donde los usuarios publican sus propios contenedores de manera que los usuarios que los necesiten los puedan bajar rápidamente desde allí.

Spotify, eBay y Rackspace, entre otras muchas empresas, utilizan este sistema de contenedores en sus centros de datos. Además, esta plataforma cuenta con el apoyo de otras grandes compañías de Internet como Amazon y Google, lo que le permite seguir creciendo y mejorando constantemente.

Más adelante veremos cómo instalar y cómo funcionan estos contenedores de virtualización, sin embargo, si queréis descargar tanto la plataforma como el código fuente de la misma podéis hacerlo de forma totalmente gratuita [**desde su página web principal**](https://www.docker.com/).

## ien, la cosa promete. ¿Cómo empezar?

**Primero: instala docker y docker-compose.**

Puedes acceder a las instrucciones oficiales [aquí](https://docs.docker.com/engine/installation/) y [aquí](https://docs.docker.com/compose/install/).

**Segundo: añade un Dockerfile propio a tu aplicación**

Dockerfile contiene una lista de instrucciones que se tienen que ejecutar para preparar el entorno, para luego poder ejecutar tu aplicación. Por ejemplo, aquí copias tus archivos de proyecto, instalas todas las dependencias del sistema y la aplicación, haces los ajustes de configuración, etc... Por ejemplo, veamos un Dockerfile que se utiliza para un proyecto en Python:

FROM python:3.6

ENV PROJECT\_ROOT=/srv

WORKDIR $PROJECT\_ROOT

COPY requirements.txt .

RUN pip install -r requirements.txt

COPY . .

CMD ["python", "manage.py", "runserver"]

Por lo general no se necesita nada más sofisticado. Para encontrar instrucciones sobre cómo crear un Dockerfile para tu lenguaje de programación solo tienes que buscarlo en Google, hay muchísimos recursos sobre esto en particular.

Si quieres aprender sobre buenas prácticas, [este artículo](https://rock-it.pl/how-to-write-excellent-dockerfiles/) ya ha ayudado a muchas personas. También te puedes dirigir a la [página oficial](https://docs.docker.com/engine/userguide/eng-image/dockerfile_best-practices).

**Tercero: crea un archivo docker-compose.yml.**

Docker-compose es una herramienta genial que nos permite gestionar simultáneamente varias pilas de contenedores sin traumas. Este archivo especial le dice a docker-compose qué contenedores tienen que iniciarse.

Este es un ejemplo de docker-compose.yml que utilizo:

version: '2'

services:

redis:

image: redis

database:

image: postgres

app:

build: .

environment:

DEBUG: 1

ports:

- 8000:8000

volumes:

- .:/srv

command: python manage.py runserver

Este archivo indica que queremos que Redis y Postgres se creen usando imágenes oficiales, y que nuestra aplicación se cree desde Dockerfile en local, disponible desde la red de Docker externa a través del puerto 8000, con todos los archivos locales montados dentro del contenedor para facilitar el desarrollo. Se pueden encontrar todas las opciones [aquí](https://docs.docker.com/compose/compose-file/). En este ejemplo se usa la versión 2 porque la versión 3 soporta Docker Swarm que no se necesita en desarrollo.

Ahora, cuando introduzcamos docker-compose up, esto es lo que va a suceder:

1. Docker intentará descargar / crear todas las imágenes requeridas. Si usamos la opción image, se descargará de Internet, por defecto desde DockerHub. La opción build dará instrucciones a Docker de crear la imagen usando el Dockerfile ubicado en un directorio concreto, en nuestro caso . (dir actual). Si la imagen ya existe, no se volverá a comprobar automáticamente (para hacerlo, debemos añadir la bandera --build)
2. Cuando las imágenes están disponibles, docker-compose comprobará si la configuración ha cambiado desde la última ejecución. Si no, se inician contenedores usados con anterioridad, y en caso contrario, se crean nuevos contenedores. La configuración de cosas como puertos, volúmenes, variables, redes, etc... se toman desde docker-compose.yml.
3. Si no se indica lo contrario, se crea una red especial, abarcando todos los contenedores creados. Les permite referenciarse entre ellos usando nombres en vez de direcciones IP. Como muestra, en el ejemplo, la base de datos de la aplicación estará disponible en postgres:5432.
4. Todos los contenedores se inician a la vez, y vemos logs de todos ellos.

Y eso es todo. Ahora toda la pila arranca automáticamente. Evidentemente deberás hacer cosas como crear automáticamente una base de datos si no existe, o volver a cargar el código de la aplicación tras un cambio para hacer que la experiencia de desarrollo sea más agradable. Lo importante aquí es que solo tienes que preparar esto una vez y todo el equipo de programación se beneficiará de ello inmediatamente.

**Aviso**: a la hora de programar, probablemente debas usar siempre la opción volumes en tu archivo docker-compose.yml, montando los archivos locales en el contenedor. Sin esta opción, tendrás que rehacer el contenedor cada vez que haya algún cambio.

## Uso típico

Estos son algunos de los comandos más utilizados en el día a día.

* **Empezando a trabajar:** docker-compose up o docker-compose up --rebuild si sabes que algunas dependencias han cambiado.
* **Ejecutando comandos una sola vez**: depende de si necesitamos disponibles otros contenedores, como la base de datos, o no. Si es así, arranca la pila con un comando anterior, y en otro ventana de terminal ejecuta docker-compose exec $SERVICE\_NAME $COMMAND. Si no, solo usa docker-compose run --rm $SERVICE\_NAME $COMMAND (yo recomiendo el "flag" --rm para eliminarlo después del comando)
* **Eliminar todo lo creado por docker-compose**: docker-compose down. Eliminará todos los contenedores, imágenes y redes.
* **Eliminar un contenedor con todos los datos asociados**: docker-compose rm -v $SERVICE\_NAME (muy útil si quieres restablecer completamente una base de datos).

Tutorial ubunto

<https://www.digitalocean.com/community/tutorials/como-instalar-y-usar-docker-en-ubuntu-18-04-1-es>