

https://www.google.com/search?q=Plataformas+de+entornos+cloud&rlz=1C1UUXU_esCO956CO956&oq=Plataformas+de+entornos+cloud&aqs=chrome..69i57j33i10i22i29i30.6267j1j7&sourceid=chrome&ie=UTF-8

¿Qué son las plataformas cloud?

De una manera simple, la computación en la nube (**cloud** computing) es una tecnología que permite acceso remoto a softwares, almacenamiento de archivos y procesamiento de datos por medio de Internet, siendo así, una alternativa a la ejecución en una computadora personal o servidor local.

Cloud computing? Llamado también computación en la nube.

¿Cuáles son sus plataformas de cloud computing?

Los 12 mejores proveedores de Cloud Computing en 2020

Microsoft. Bajo el **nombre de** Microsoft Azure encontramos la **plataforma de cloud computing** de Microsoft. ...

Amazon. ...

IBM. ...

Salesforce.com. ...

SAP. ...

CenturyLink. ...

Virtustream. ...

Rackspace.

¿Cuáles son los 3 tipos de almacenamiento en la nube?

Existen tres **tipos de nube de almacenamiento**: pública, privada e híbrida.

¿Cuáles son los servidores de nube pública?

A continuación conoceremos tres de los principales servicios de nube pública que existen en el mercado.

- Microsoft Azure. ...
- Amazon Web Services. ...
- Google Cloud.

¿Cuáles son las unidades de almacenamiento en la nube?

Los planes gratuitos de los principales servicios

	Google One	Amazon Drive
2016	15 GB	5 GB
2017	15 GB	5 GB
2018	15 GB	5 GB
2019	15 GB	5 GB

¿Cuáles son las nubes de almacenamiento más comunes?

5 plataformas de almacenamiento en la nube gratuito

DropBox. ...
 Claro drive. ...
 Google Drive. ...
 Microsoft OneDrive. ...
 Amazon Drive.

¿Qué son las nubes públicas y privadas?



Una **nube privada** es un servicio en la **nube** que no está compartido con ninguna otra organización. ... En cambio, una **nube pública** es un servicio en la **nube** que comparte servicios informáticos entre diferentes clientes, aunque los datos y las aplicaciones de cada cliente permanecen ocultos para los otros clientes de la **nube**.

¿Qué nube da más espacio gratis 2021?

Los 10 mejores servicios gratis de almacenamiento en la nube

Más de 150 GB de **espacio gratis** de almacenamiento en la **nube**.

Mega: almacenamiento en la **nube** de 50 GB **gratis**. ...
 Google Drive: almacenamiento en la **nube** de 15 GB **gratis**. ...
 OneDrive: almacenamiento en la **nube** de 5 GB **gratis**.

MEGA es uno de los servicios de almacenamiento Cloud más populares, debido a que ofrecen con su cuenta gratuita una capacidad de almacenamiento de 50GB, sin embargo, tenemos un límite de descarga y subida por dirección IP de tan solo 5GB de datos (con una cuenta registrada y versión gratuita).

¿Cuál es la función de la nube?

La **nube** es el nombre que le damos al servicio de almacenamiento de datos a servidores localizados en la red. Esta modalidad permite subir, abrir, modificar o usar programas y archivos a través de una conexión sin la necesidad de que se encuentren en el almacenamiento del dispositivo que usas.

¿Cómo se utiliza la nube?

Cómo usar Google Drive

Cómo comenzar a usar **Google** Drive. Tienes 15 GB de espacio sin cargo en tu unidad de Drive. ...

Paso 1: Ve a drive.**google**.com. En tu computadora, ve a drive.**google**.com. ...

Paso 2: Sube o crea archivos. ...

Paso 3: Comparte y organiza tus archivos.

<https://www.claro.com.co/institucional/almacenamiento-en-la-nube-gratuito/>

30 Marzo 2021

¿Dónde guardar documentos personales y empresariales?

Si has perdido tu teléfono móvil o se dañó tu computador de un momento a otro y no tienes copia de fotos, videos y documentos, te darás cuenta de la importancia de haber utilizado un espacio virtual para guardarlos de forma segura y accesible. En este sentido, hoy existen diferentes sistemas de almacenamiento en la nube gratuitos que brindan la posibilidad para que los colombianos puedan almacenar todos estos archivos de manera segura, y acceder a ellos desde sus computadores y dispositivos móviles, sin ocupar espacio de estos últimos o sin restricciones por la marca.

Conoce cuáles están disponibles, sus ventajas y beneficios.

1. DropBox

Cuenta con un plan que se llama Basic, con 2GB de espacio, que permite el almacenamiento, sincronización, acceso de fotos, videos, presentaciones y otros documentos en distintos dispositivos, carpetas y vínculos compartidos.

2. Claro drive

Incluye 25GB de almacenamiento sin costo para clientes de [planes móviles postpago](#) y [servicios fijos de Claro](#), para que puedan subir a la nube fotos, música, videos o presentaciones del colegio, universidad o trabajo.

Quienes quieran acceder solamente deberán ingresar a la página www.clarodrive.com desde el móvil Claro o el computador o descargar la aplicación **Claro drive**. De acuerdo con Rodrigo de Gusmao, director ejecutivo Unidad Mercado Masivo de Claro Colombia, *“por ejemplo, con este servicio gratuito de 25GB se tiene una capacidad, aproximadamente, para 5.689 fotos, más de 5.000 canciones, 350 mil documentos, o hasta 20 películas”*.

3. Google Drive

Con la cuenta de Google, los primeros 15GB de espacio de almacenamiento son gratuitos. Se puede acceder a los **archivos de Drive** desde cualquier Smartphone, tableta o computador.

4. Microsoft OneDrive

OneDrive Básico de 5 GB. Entre sus principales opciones se encuentra que permite edita y anotar en Office Docs y PDF, incluso en el teléfono móvil, funciona en PC y Mac.

5. Amazon Drive

Las personas con una **cuenta de Amazon** reciben 5 GB de almacenamiento gratuito en Amazon Drive compartido con Amazon Photos, el cual permite compartir archivos así como copias de seguridad las fotos, organizarlas y compartirlas desde el teléfono, el computador y otros dispositivos.

Principales beneficios de estos sistemas de almacenamiento

- 1. Facilidad de acceso:** únicamente necesitas de una conexión a Internet para acceder desde cualquier dispositivo y lugar a sus archivos.
- 2. Fáciles de usar:** están diseñados para que cualquier persona pueda utilizarlos, incluso para que de forma automática realicen copias de seguridad.
- 3. No toca preocuparse por nada:** las empresas que ofrecen estos servicios en la nube realizan el mantenimiento, las actualizaciones de software e invierten en infraestructura, de forma transparente para el usuario.
- 4. Los archivos corren menor riesgo de desaparecer:** prácticamente la totalidad de servicios en la nube disponen de réplicas. Así, en el hipotético caso que se estropeará o destruyera el centro de procesamientos de datos, no pasaría nada.
- 5. Permiten un ahorro económico:** todos los costos asociados a estas plataformas (servidores, mantenimiento o consumo eléctrico) se dividen entre los usuarios de servicios 'premium'. Por lo general, las opciones gratuitas solo entregan algunos cuantos GB de capacidad, por lo que si una persona o una empresa requiere mayor espacio, paga un adicional por obtenerlo.

Modelos de despliegue

1. IaaS (Infraestructura como servicio)
2. PaaS (Plataforma como servicio)

- 3. SaaS (Software como servicio)
 - 3.1 Office 365
 - 3.2 XXXXXXXX
- 4. FaaS

Tipos de informática en la nube

La informática en la nube ofrece a los desarrolladores y departamentos de TI la capacidad de concentrarse en lo que más importa y evitar arduas tareas como el aprovisionamiento, el mantenimiento y la planificación de capacidad. A medida que ha incrementado la popularidad de la informática en la nube, se han desarrollado varios modelos y estrategias de implementación para satisfacer las necesidades de los distintos usuarios. Cada tipo de servicio en la nube y método de implementación le aporta distintos niveles de control, flexibilidad y administración. Entender la diferencia entre la Infraestructura como servicio, la Plataforma como servicio y el Software como servicio, además de las estrategias de implementación disponibles, puede ayudarle a determinar el conjunto de servicios que más se adapta a sus necesidades.



Modelos de informática en la nube

Existen tres modelos principales de informática en la nube. Cada modelo representa una parte distinta de la pila de informática en la nube.



Infraestructura como servicio (IaaS)

La infraestructura como servicio, que a veces se abrevia a IaaS, contiene los bloques de creación fundamentales para la TI en la nube. Por lo general, permite acceder a las características de conexión en red, a los equipos (virtuales o en software dedicado) y al espacio de almacenamiento de datos. La infraestructura como servicio le ofrece el mayor nivel de flexibilidad y control de la administración en torno a sus recursos de TI y guarda el mayor parecido con los recursos de TI

existentes con los que muchos departamentos de TI y desarrolladores están familiarizados.



Plataforma como servicio (PaaS)

Las plataformas como servicio eliminan la necesidad de las compañías de administrar la infraestructura subyacente (normalmente hardware y sistemas operativos) y le permiten centrarse en la implementación y la administración de sus aplicaciones. Esto contribuye a mejorar su eficacia, pues no tiene que preocuparse del aprovisionamiento de recursos, la planificación de la capacidad, el mantenimiento de software, los parches ni ninguna de las demás arduas tareas que conlleva la ejecución de su aplicación.



Software como servicio (SaaS)

El software como servicio le proporciona un producto completo que el proveedor del servicio ejecuta y administra. En la mayoría de los casos, quienes hablan de software como servicio en realidad se refieren a aplicaciones de usuario final. Con una oferta de SaaS, no tiene que pensar en cómo se mantiene el servicio ni en cómo se administra la infraestructura subyacente. Solo debe preocuparse por cómo utilizar ese sistema de software concreto. Un ejemplo común de una aplicación SaaS es un programa de correo electrónico basado en la web que le permite enviar y recibir mensajes sin tener que administrar la incorporación de características ni mantener los servidores y los sistemas operativos en los que se ejecuta el programa de correo electrónico.

Modelos de implementación de informática en la nube



Nube

Una aplicación basada en la nube se encuentra implementada totalmente en la nube, de modo que todas las partes de la aplicación se ejecutan en esta. Las aplicaciones en la nube se han creado directamente en la nube o se han transferido de la infraestructura existente para aprovechar los beneficios de la informática en la nube. Las aplicaciones basadas en la nube se pueden construir en partes de infraestructura de bajo nivel o pueden utilizar servicios de nivel superior que proporcionan abstracción de los requisitos de administración, arquitectura y escalado de la infraestructura principal.



Solución híbrida

Una implementación híbrida es una manera de conectar la infraestructura y las aplicaciones entre los recursos basados en la nube y los recursos existentes situados fuera de la nube. El método más común de implementación híbrida consiste en conectar la nube y la infraestructura existente en las instalaciones para ampliar e incrementar la infraestructura de la organización en la nube al mismo tiempo que se conectan estos recursos en la nube con el sistema interno. Para obtener más información sobre cómo AWS lo puede ayudar a establecer una implementación híbrida, visite nuestra página acerca de la nube híbrida.



En las instalaciones

La implementación local de recursos mediante herramientas de administración de recursos y virtualización se denomina a veces "nube privada". La implementación local no aporta muchos de los beneficios de la informática en la nube, pero a veces se utiliza por su capacidad de ofrecer recursos dedicados. En la mayoría de los casos, este modelo de implementación es idéntico al de la infraestructura de TI antigua, mientras que utiliza tecnologías de virtualización y administración de aplicaciones para intentar incrementar el uso de los recursos.

Siguientes pasos

¿Listo para dar el siguiente paso? AWS puede ayudarle a obtener más información sobre AWS, a encontrar la solución que mejor se adapta a sus necesidades o a

contactar con un representante de ventas de AWS que lo asistirá a fin de encontrar la solución adecuada para usted.

Más información sobre AWS

Amazon Web Services ofrece la gama más amplia de servicios en la nube del mundo. Para obtener más información sobre los beneficios de implementar su próxima aplicación con AWS, visite nuestra página [¿Qué es AWS?](#).

Descubra las soluciones de AWS

Con soluciones y servicios para TI, operaciones de desarrollo y para desarrolladores, AWS dispone de una amplia plataforma que lo ayudará a hacer realidad su próximo proyecto. Para obtener más información sobre las soluciones y servicios disponibles en la nube de AWS, visite la página de soluciones de AWS.

Contáctese con un representante de ventas

¿Está listo para hablar con un empleado de AWS? Complete nuestro formulario de contacto y un representante de ventas de AWS capacitado lo llamará para hablar de sus necesidades y determinar cómo AWS puede ayudarlo a transferir su próximo proyecto a la nube.

¿Qué es el modelo en cascada?

El desarrollo en cascada (en inglés, waterfall model) es un **procedimiento lineal** que se caracteriza por dividir los procesos de desarrollo en sucesivas fases de proyecto. Al contrario que en los modelos iterativos, cada una de estas fases se ejecuta tan solo una vez. Los resultados de cada una de las fases sirven como hipótesis de partida para la siguiente. El waterfall model se utiliza, especialmente, en el desarrollo de software.

¿Cómo funciona el modelo en cascada?

El desarrollo del modelo se atribuye al teórico de la informática Winston W. Royce. Sin embargo, Royce no es el inventor de este modelo. Muy al contrario, en su ensayo de 1970 titulado [*Managing the Development of Large Software Systems*](#), el teórico presenta una **reflexión crítica acerca de los procedimientos lineales**. A modo de alternativa, Royce presenta un modelo iterativo incremental en el que cada una de las fases se basa en la anterior y verifica los resultados de esta.

Royce propone un modelo compuesto por siete fases que se ha de ejecutar en diversas vueltas (iteraciones):

1. **Requisitos de sistema**
2. **Requisitos de software**
3. **Análisis**
4. **Diseño**
5. **Implementación**
6. **Prueba**
7. **Servicio**

El procedimiento popularmente conocido como waterfall model se basa en las fases definidas por Royce, **pero solo prevé una iteración**.

En el ensayo publicado por Royce, el término no aparece en ningún momento.

En la práctica, se aplican **diversas versiones del modelo**. Los más habituales son los modelos que dividen los procesos de desarrollo en cinco fases. En ocasiones, las fases 1, 2 y 3 definidas por Royce se integran en una sola fase de proyecto a modo de análisis de los requisitos.

1. **Análisis:** planificación, análisis y especificación de los requisitos.
2. **Diseño:** diseño y especificación del sistema.
3. **Implementación:** programación y pruebas unitarias.
4. **Verificación:** integración de sistemas, pruebas de sistema y de integración.
5. **Mantenimiento:** entrega, mantenimiento y mejora.

La siguiente imagen explica por qué el procedimiento lineal se denomina metodología en cascada.



El modelo en cascada de cinco niveles, basado en las propuestas de Winston W. Royce, divide los procesos de desarrollo en las siguientes fases de proyecto: análisis, diseño, implementación, verificación y mantenimiento. El gráfico incluye una de las ampliaciones del modelo planteadas por Royce: la verificación de los resultados de cada una de las fases tomando en consideración las exigencias y especificaciones formuladas en el paso anterior.

En las ampliaciones de la metodología en cascada se añaden funciones iterativas al modelo básico como, por ejemplo, los saltos hacia atrás, que permiten comparar los resultados de cada una de las fases con las hipótesis obtenidas en la fase anterior, de modo que se puedan verificar.

Las fases del desarrollo en cascada

En este modelo, las diferentes fases de un proceso de desarrollo se suceden una detrás de otra como en una cascada. Cada una de las fases concluye con un **resultado provisional (hito)** como, por ejemplo, un catálogo de requisitos en

forma de pliego de condiciones, la especificación de una arquitectura de software o una aplicación a nivel alfa o beta.

Análisis

Todo proyecto de software comienza con una fase de análisis que incluye un estudio de viabilidad y una definición de los requisitos. En el **estudio de viabilidad** se evalúan los costes, la rentabilidad y la factibilidad del proyecto de software. El estudio de viabilidad da como resultado un pliego de condiciones (una descripción general de los requisitos), un plan y una estimación financiera del proyecto, así como una propuesta para el cliente, si fuera necesario.

A continuación, se realiza una **definición detallada de los requisitos**, incluyendo un análisis de la situación de salida y un concepto. Mientras que los análisis de salida se encargan de describir la problemática en sí, el concepto ha de definir qué funciones y características debe ofrecer el producto de software para cumplir con las correspondientes exigencias. La definición de los requisitos da como resultado un pliego de condiciones, una descripción detallada de cómo se han de cumplir los requisitos del proyecto, así como un plan para la prueba de aceptación, entre otros.

Por último, la primera fase del waterfall model incluye un **análisis de la definición de los requisitos** en el que los problemas complejos se dividen en pequeñas tareas secundarias y se elaboran las correspondientes estrategias de resolución.

Diseño

La fase de diseño sirve para formular una solución específica en base a las exigencias, tareas y estrategias definidas en la fase anterior. En esta fase, los desarrolladores de software se encargan de diseñar la **arquitectura de software**, así como un **plan de diseño detallado del mismo**, centrándose en componentes concretos, como interfaces, entornos de trabajo o bibliotecas. La fase de diseño da como resultado un borrador preliminar con el plan de diseño del software, así como planes de prueba para los diferentes componentes.

Implementación

La arquitectura de software concebida en la fase de diseño se ejecuta en la **fase de implementación**, en la que se incluye la **programación del software**, la **búsqueda de errores** y las **pruebas unitarias**. En la fase de implementación, el proyecto de software se traduce al correspondiente lenguaje de programación. Los diversos componentes se desarrollan por separado, se comprueban a través de las **pruebas unitarias** y se integran poco a poco en el producto final. La fase de implementación da como resultado un producto de software que se comprueba por primera vez como producto final en la siguiente fase (prueba alfa).

Prueba

La fase de prueba incluye la integración del software en el entorno seleccionado. Por norma general, los productos de software se envían en primer lugar a los

usuarios finales seleccionados en **versión beta** (pruebas beta). Las **pruebas de aceptación** desarrolladas en la fase de análisis permiten determinar si el software cumple con las exigencias definidas con anterioridad. Aquellos productos de software que superan con éxito las pruebas beta están listos para su lanzamiento.

Servicio

Una vez que la fase de prueba ha concluido con éxito, se autoriza la **aplicación productiva** del software. La última fase del modelo en cascada incluye la **entrega**, el **mantenimiento** y la **mejora del software**.

Pros y contras del modelo en cascada

Esta metodología permite **estructurar la organización de forma clara** en aquellos proyectos de desarrollo en los que las diversas fases de proyecto se diferencian claramente entre sí. Como cada una de las fases concluye con un hito, el proceso de desarrollo es muy fácil de comprender. El punto clave del modelo reside en la documentación de todos y cada uno de los pasos de proceso. Los conocimientos adquiridos se registran en pliegos de requisitos o borradores preliminares.

En teoría, el desarrollo en cascada pretende crear los requisitos previos para una ejecución rápida y rentable de los proyectos a través de una cuidada planificación previa. Sin embargo, **la utilización del modelo en la práctica es controvertida**. Por una parte, en el desarrollo de software las fases de proyecto no suelen estar claramente diferenciadas entre sí. Es precisamente en los proyectos de software más complejos donde los desarrolladores se suelen enfrentar al hecho de que los diversos componentes de una misma aplicación se encuentran en diferentes fases de desarrollo al mismo tiempo. Por otra parte, la secuencia lineal del waterfall model no suele coincidir con la realidad.

En sentido estricto, el modelo en cascada no prevé la realización de ajustes a lo largo del proyecto. Sin embargo, un proyecto de software en el que todos los detalles del desarrollo se definieran al comienzo, solo podría concluir con éxito si desde el principio se invirtiera una gran cantidad de tiempo y dinero en análisis y diseño. A todo esto, se añade que los proyectos de software de más envergadura se suelen prolongar durante varios años y, de no adaptarse a los avances más actuales, obtendrían resultados **que ya estarían obsoletos en el momento de su aplicación**.

Resumen de las ventajas y desventajas del modelo en cascada

Ventajas	Inconvenientes
✓ Una estructura sencilla gracias a unas fases de proyecto claramente diferenciadas.	✗ Por norma general, los proyectos más complejos o de varios niveles no permiten su división en fases de proyecto claramente diferenciadas.
✓ Buena documentación del proceso de desarrollo a través de unos hitos bien definidos.	✗ Poco margen para realizar ajustes a lo largo del proyecto debido a un cambio en las exigencias.
✓ Los costes y la carga de trabajo se pueden estimar al comenzar el proyecto.	✗ El usuario final no se integra en el proceso de producción hasta que no termina la programación.
✓ Aquellos proyectos que se estructuran en base al modelo en cascada se pueden representar cronológicamente de forma sencilla.	✗ En ocasiones, los fallos solo se detectan una vez finalizado el proceso de desarrollo.

La metodología en cascada se suele emplear, especialmente, en aquellos proyectos cuyos requisitos y procesos se pueden describir de forma precisa durante la fase de planificación, en los que cabe suponer que las hipótesis no van a sufrir una gran variación durante el transcurso del proyecto. Los procedimientos estrictamente lineales se adaptan, así, especialmente bien a **proyectos de software pequeños, sencillos y claramente estructurados**. A esta misma conclusión llegó Royce en los años 1970. Por este motivo, la alternativa al procedimiento lineal que propuso, y que más tarde se conocería como waterfall model, incluía tres ampliaciones principales:

Verificación tras cada fase de proyecto

Según Royce, los resultados de cada una de las fases de proyecto se deben comparar y verificar inmediatamente con los documentos elaborados previamente. Es decir, inmediatamente después de desarrollar un módulo, por ejemplo, se debería garantizar que este cumple con las exigencias definidas con anterioridad sin esperar a que concluya el proceso de desarrollo.

Al menos, dos iteraciones

Según Royce, el modelo se debería ejecutar en, al menos, dos ocasiones: primero para **elaborar un prototipo** y, a continuación, para **desarrollar el producto de software en sí**.

Pruebas que incluyen al usuario final

La tercera ampliación del modelo en cascada propuesta por Royce en su ensayo es una medida que, a día de hoy, se ha convertido en un procedimiento estándar en el desarrollo de productos: la inclusión del usuario final en el proceso de producción. Royce propone incluir al usuario en tres momentos diferentes del proceso de

desarrollo de software: durante la planificación del software en la fase de análisis, entre el diseño del software y su implementación y en la fase de prueba que precede al lanzamiento del software.

Procedimientos alternativos al modelo en cascada

Debido a la secuencia estrictamente lineal de las fases sucesivas de proyecto, el modelo en cascada se adaptaría, en el mejor de los casos, a proyectos de software de poca envergadura. Por el contrario, los procesos complejos de varios niveles serían difícilmente representables con este modelo. Por este motivo, con el paso del tiempo han ido surgiendo enfoques alternativos.

Mientras que algunos modelos, como el modelo en espiral o el modelo en V, se consideran una **evolución del waterfall model clásico**, algunos métodos, como la programación extrema, el desarrollo ágil de software o el desarrollo iterativo, se centran en un enfoque completamente diferente y suelen permitir una adaptación a los cambios más recientes o a las exigencias más actuales.