

CONFIGURACION DE SERVIDORES

Universidad de Cuenca

Kevin Mateo Alvarado Suarez

01/04/2024

Programación Web

MARCO TEÓRICO

Servidor Web

Son programas que sirven para transmitir los archivos que forman parte de una web. Los servidores web están atentos a conexiones mediante el protocolo HTTP o HTTPS. Los clientes, habitualmente navegadores web, solicitan un recurso identificado una URL y los servidores web se encargan de devolverlo como respuesta.

Los archivos que forman parte de una web muchas veces se entregan tal cual, como ocurre generalmente con las imágenes o el código HTML o CSS estático, pero a veces el servidor web también tiene la tarea de solicitar la ejecución de ciertos códigos, en lenguajes como PHP, para construir el código HTML que debe de ser enviado al cliente.



Los servidores web más populares son los siguientes:

- Apache, con diferencia el más usado en la web.
- Nginx, que es más ligero y suele ser un poco más rápido.
- IIS, Internet Information Services, es el servidor web más tradicional para sistemas Windows.

Servidor de Aplicaciones

Se denomina servidor de aplicaciones a un servidor en una red de computadoras que ejecuta ciertas aplicaciones. Usualmente se trata de un dispositivo de software que proporciona servicios de aplicación a las computadoras cliente.

Un servidor de aplicaciones generalmente gestiona la mayor parte (o la totalidad) de las funciones de lógica de negociación y de acceso a los datos de las aplicaciones.

Los principales beneficios de la aplicación de la tecnología de servidores de aplicación son la centralización y la disminución de la complejidad en el desarrollo de aplicaciones.

Los proveedores y modelos más conocidos y populares son los siguientes:

- Apache Geronimo: apto para Java EE
- Apache Tomcat: para Servlets y JavaServer Pages (JSP)
- ColdFusion: el servidor de aplicaciones más antiguo
- GlassFish: solución de código abierto para Java EE
- Jetty: contenedor de servlets y JSP
- Resin: para Java y PHP5
- WEBrick: solución basada en Ruby
- Zope: un servidor de aplicaciones basado en Python

Hosting

Un hosting es un servicio de alojamiento web que permite publicar un sitio web o aplicación en Internet. Cuando se contrata un servicio de hosting, básicamente se alquila un espacio en un servidor físico donde se puede almacenar todos los archivos y datos necesarios para que un sitio web funcione correctamente.

Esto funciona de la siguiente manera:

Un servidor es una computadora o máquina física que funciona ininterrumpidamente para que un sitio web esté disponible todo el tiempo para cualquier persona que quiera verlo.

El proveedor de hosting web es el responsable de mantener el servidor en funcionamiento, protegerlo de ataques maliciosos y transferir el contenido (como texto, imágenes, etc) desde el servidor a los navegadores de los visitantes.

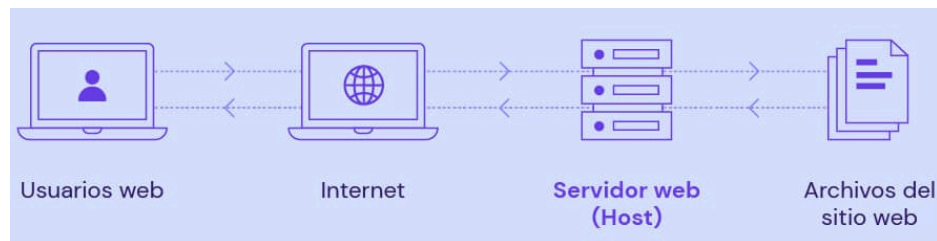
Cuando se decide crear una página web nueva, se tiene que encontrar una empresa de hosting que proporcione espacio en un servidor. El proveedor de hosting almacena todos los archivos, medios y bases de datos en el servidor.

Cada vez que alguien escribe el nombre de dominio en la barra de direcciones de su navegador, el servidor transfiere todos los archivos necesarios para atender la solicitud.

Se puede alojar uno mismo el sitio web, pero requiere amplios conocimientos técnicos. El autoalojamiento implica instalar y configurar un servidor web desde cero, incluyendo el equipo, la infraestructura, el hardware y el software. Además de encargarse del mantenimiento.

Un proveedor de servicios de alojamiento web garantiza que la página web funcione de forma óptima y con mejores protocolos de seguridad. Además, simplifica los numerosos y complejos aspectos del hosting de una página web, desde la instalación del software

hasta el soporte técnico.



Cloud Computing

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. En el modelo de nube, no hay necesidad de instalar aplicaciones localmente en computadoras.

La computación en la nube ofrece a los individuos y a las empresas la capacidad de un pool de recursos de computación con buen mantenimiento, seguro, de fácil acceso y bajo demanda.

La computación en la nube utiliza una capa de red para conectar los dispositivos de punto periférico de los usuarios, como computadoras, smartphones y accesorios portátiles, a recursos centralizados en el data center.

Antes de la computación en la nube, la ejecución confiable de software por las empresas que ofrecían servicios solo era posible si ellas podían también pagar por el mantenimiento de la infraestructura de los servidores necesarios.

Además, el software tradicional exigía, por lo general, un equipo completo de profesionales de TI, interno o externo, para lidiar con el inevitable conjunto de errores, desafíos de servicios y actualizaciones. El concepto de computación en la nube está libre de todos esos problemas y requisitos anticuados.

Despliegue continuo

El despliegue continuo es una estrategia de desarrollo de software en la que los cambios de código de una aplicación se publican automáticamente en el entorno de producción. Esta automatización se basa en una serie de pruebas predefinidas. Una vez que las nuevas actualizaciones pasan esas pruebas, el sistema envía las actualizaciones directamente a los usuarios del software.

El despliegue continuo ofrece varias ventajas para las empresas que quieren escalar su portfolio de aplicaciones y TI. En primer lugar, agiliza el tiempo de comercialización al eliminar el desfase entre la programación y el valor al cliente, que suele ser de días, semanas o incluso meses.

Para lograrlo, se deben automatizar las pruebas de regresión, de modo que se eliminen los altos costes de las pruebas de regresión manual. Los sistemas que las organizaciones

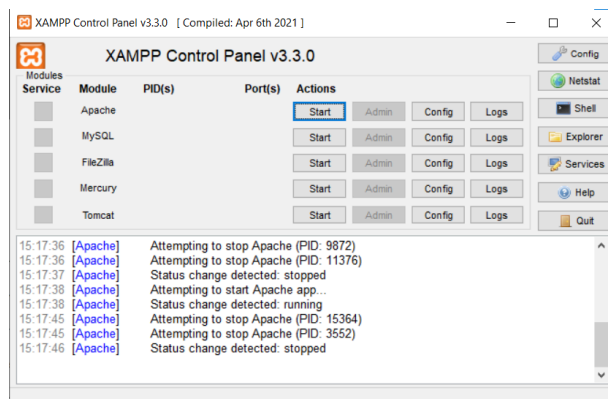
han puesto en marcha para gestionar grandes paquetes de cambios de producción, incluidas las reuniones de planificación y aprobación de publicaciones, también se pueden eliminar en la mayoría de los cambios.

EJERCICIOS

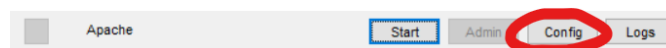
Apache XAMPP

Cambiar los puertos de Apache XAMPP

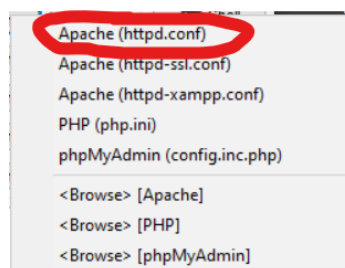
1. Abrir el panel de control de XAMPP.



2. Hacer clic en el botón "Config" para Apache.



3. Seleccionar "httpd.conf" para abrir el archivo de configuración de Apache.

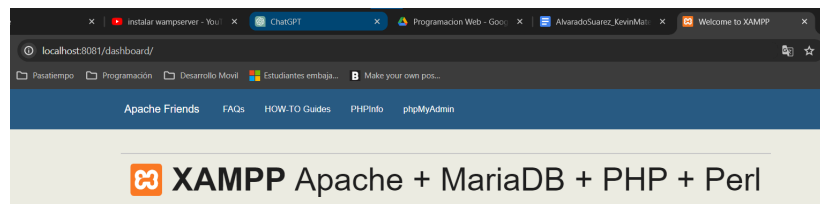


4. Buscar la línea que dice Listen 80 y cámbiala a Listen 8081.

```
# Change this to Listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses.
#
#Listen 12.34.56.78:80
Listen 8081

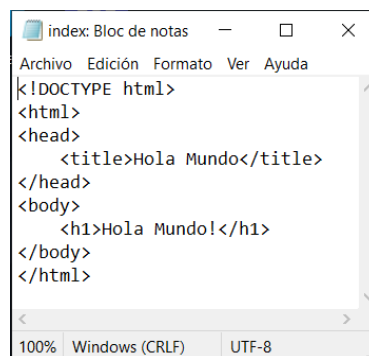
#
# Dynamic Shared Object (DSO) Support
```

5. Guarda los cambios y cierra el archivo.
6. Reinicia Apache desde el panel de control de XAMPP.
7. Para comprobar que los cambios fueron realizados, se debe abrir un navegador web, en la barra de direcciones, escribe `http://localhost:8081`. Si se ve la página de inicio de XAMPP o cualquier otra página que hayas configurado, significa que Apache está escuchando en el puerto 8081.



Desplegar una página web sencilla de “Hola Mundo”

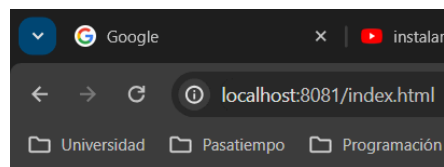
1. Crea un archivo HTML llamado `index.html` con el contenido de una página “Hola Mundo”.



2. Colocar este archivo en el directorio de publicación de Apache. Por lo general, este directorio es `htdocs` dentro del directorio de instalación de XAMPP, que en este caso se encuentra en la ruta `C:\xampp\htdocs`

Este equipo > Windows-SSD (C:) > xampp > htdocs			
Nombre	Fecha de modificación	Tipo	Tamaño
dashboard	31/3/2024 15:12	Carpeta de archivos	
img	31/3/2024 15:12	Carpeta de archivos	
webalizer	31/3/2024 15:11	Carpeta de archivos	
xampp	31/3/2024 15:12	Carpeta de archivos	
applications	15/6/2022 11:07	Chrome HTML Do...	4 KB
bitnami	15/6/2022 11:07	Documento de hoj...	1 KB
favicon	16/7/2015 10:32	Icono	31 KB
index	31/3/2024 15:38	Chrome HTML Do...	1 KB
index.php	16/7/2015 10:32	Archivo PHP	1 KB

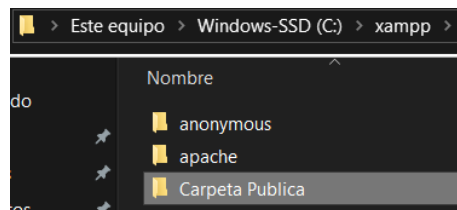
3. Para comprobar que funcione, abrir un navegador web, ingresar la URL correspondiente `http://localhost:8081/index.html` y se debería ver la página "Hola Mundo" que se creó.



Hola Mundo!

Cambiar carpeta de publicación

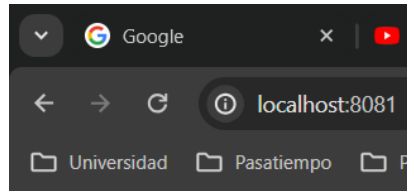
1. Crear una carpeta llamada "Carpeta Publica" en el directorio de instalación de XAMPP.



2. Abrir el archivo de configuración de Apache (httpd.conf) como se hizo anteriormente.
3. Busca la línea que dice DocumentRoot y cámbiala para que apunte al directorio que se acaba de crear.

```
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
#
DocumentRoot "C:/xampp/Carpeta Publica"
<Directory "C:/xampp/htdocs">
```

4. Para comprobar que funcione, abrir un navegador web, ingresar la URL correspondiente `http://localhost:8081` y se debería ver la página "Hola Mundo" que se creó.



Hola Mundo 2!

Árbol de archivos para publicación

1. Para cada una de las páginas o subpáginas necesarias se tendría una estructura general como se muestra a continuación:

css	31/3/2024 16:32	Carpeta de archivos	
imagenes	31/3/2024 16:32	Carpeta de archivos	
scripts	31/3/2024 16:32	Carpeta de archivos	
index	31/3/2024 15:57	Chrome HTML Do...	1 KB

2. Para publicar en cada una de ellas sería tan solo necesario acceder a el archivo a través del navegador web utilizando la ruta adecuada, por ejemplo, `http://localhost/Pagina%Inicio/Subpagina/index.html`.

Configurar el firewall

1. Abrir el Panel de Control.
2. Seleccionar "Firewall de Windows Defender"
3. Seleccionar "Configuración avanzada"
4. Luego en "Reglas de entrada" y agregar una nueva regla para permitir el tráfico en el puerto deseado.
5. En el asistente, seleccionar el tipo de regla que desees crear. En este caso, seleccionar "Puerto" y hacer clic en "Siguiente".

¿Qué tipo de regla desea crear?

☐ **Programa**
Regla que controla las conexiones de un programa.

☒ **Puerto**
Regla que controla las conexiones de un puerto TCP o UDP.

☐ **Predefinida:**
@FirewallAPI.dll-80200
Regla que controla las conexiones de una experiencia con Windows.

☐ **Personalizada**
Regla personalizada.

6. Seleccionar "Puertos locales específicos" y luego ingresar el número de puerto que desees permitir. En este caso el puerto 8081 y seleccionar TCP. Hacer clic en "Siguiente".

¿Se aplica esta regla a TCP o UDP?

☒ **TCP**
☐ **UDP**

¿Se aplica esta regla a todos los puertos locales o a unos puertos locales específicos?

☐ **Todos los puertos locales**
☒ **Puertos locales específicos:**
Ejemplo: 80, 443, 5000-5010

7. Seleccionar "Permitir la conexión" y hacer clic en "Siguiente".

¿Qué medida debe tomarse si una conexión coincide con las condiciones especificadas?

☒ **Permitir la conexión**
Esto incluye las conexiones protegidas mediante IPsec y las que no lo están.

☐ **Permitir la conexión si es segura**
Esto incluye solamente las conexiones autenticadas mediante IPsec. Éstas se protegerán mediante la configuración de reglas y propiedades de IPsec del nodo Regla de seguridad de conexión.

☐ **Bloquear la conexión**

8. Seleccionar los perfiles de red a los que se desea aplicar la regla (pública, privada o de dominio). Por lo general, es seguro seleccionar todas las opciones para garantizar que la regla se aplique en todas las redes. Hacer clic en "Siguiente".

¿Cuándo se aplica esta regla?

☒ **Dominio**
Se aplica cuando un equipo está conectado a su dominio corporativo.

☒ **Privado**
Se aplica cuando un equipo está conectado a una ubicación de red privada, como una red doméstica o del lugar de trabajo.

☒ **Público**
Se aplica cuando un equipo está conectado a una ubicación de redes públicas.

9. Asignar un nombre y, opcionalmente, una descripción para la regla. Luego, hacer clic en "Finalizar". Y listo la regla estará aplicada.

✓ Conexión Puerto 8081	Todo	Sí
✓ adb.exe	Público	Sí
✓ adb.exe	Público	Sí

EJERCICIOS

Apache Tomcat

Cambiar los puertos de Apache Tomcat

1. Localizar el archivo server.xml. Normalmente se encuentra en la carpeta de configuración de Tomcat, en este caso C:\Program Files\Apache Software Foundation\Tomcat 10.1\conf\server.xml
2. Abrir el archivo server.xml con un editor de texto.
3. Buscar la sección donde está definido el conector HTTP. Buscar <Connector port="8080" ...> para el conector predeterminado.

```
-->
<Connector port="8080" protocol="HTTP/1.1"
    connectionTimeout="20000"
    redirectPort="8443"
    maxParameterCount="1000"
/>

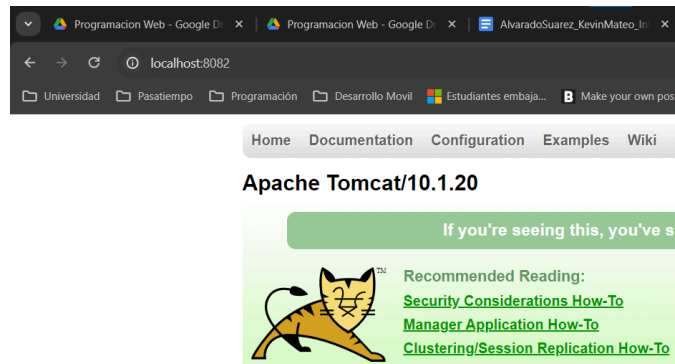
<!-- A "Connector" using the shared thread pool-->
<!--
<Connector executor="tomcatThreadPool"
    port="8080" protocol="HTTP/1.1"
    connectionTimeout="20000"
    redirectPort="8443"
    maxParameterCount="1000"
/>
```

4. Cambiar el valor del atributo port al puerto deseado. En este caso cambiarlo al puerto 8082, el conector se vería así: <Connector port="8082" ...>.

```
-->
<Connector port="8082" protocol="HTTP/1.1"
           connectionTimeout="20000"
           redirectPort="8443"
           maxParameterCount="1000"
           />

<!-- A "Connector" using the shared thread pool-->
<!--
<Connector executor="tomcatThreadPool"
           port="8082" protocol="HTTP/1.1"
           connectionTimeout="20000"
           redirectPort="8443"
           maxParameterCount="1000"
           />
```

5. Para comprobar que los cambios fueron realizados, se debe abrir un navegador web, en la barra de direcciones, escribe http://localhost:8082. Si se ve la página de inicio de Apache Tomcat o cualquier otra página que hayas configurado, significa que Apache está escuchando en el puerto 8082.

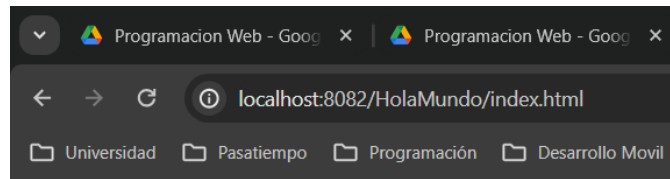


Desplegar una página web "Hola Mundo"

1. Crear un archivo HTML llamado index.html que contenga el siguiente contenido:

```
index: Bloc de notas
Archivo Edición Formato Ver Ayuda
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Página de Hola Mundo</title>
</head>
<body>
  <h1>Hola Mundo!</h1>
  <p>Esta es una página web sencilla desplegada en Apache Tomcat.</p>
</body>
</html>
```

2. Guardar este archivo en un directorio llamado HolaMundo en el directorio webapps de la instalación de Apache Tomcat. En este caso la ruta final es: C:\Program Files\Apache Software Foundation\Tomcat 10.1\webapps\HolaMundo.
3. Abrir el navegador web y navegar a la URL <http://localhost:8082/HolaMundo/index.html>. Se debería ver la página web "Hola Mundo" en el navegador.



Hola Mundo!

Esta es una página web sencilla desplegada en Apache Tomcat.

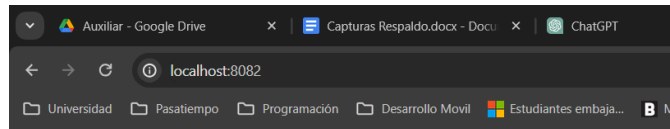
Crear y configurar una carpeta de publicación

1. Crear un directorio llamado "Carpeta Publica" en el directorio webapps de la instalación de Apache Tomcat. En este caso la ruta final es: C:\Program Files\Apache Software Foundation\Tomcat 10.1\webapps\Carpeta Publica.
2. Abrir el archivo server.xml ubicado en la carpeta conf dentro del directorio de instalación de Apache Tomcat.
3. Buscar la etiqueta <Host> en el archivo server.xml. Dentro de esta etiqueta, se encontrará una sección llamada <Context>. Agrega una nueva entrada <Context> que apunte a tu carpeta "Carpeta Pública".

```
<Host name="localhost" appBase="C:\Program Files\Apache Software Foundation\
unpackWARs="true" autoDeploy="true">

  <!-- Agrega la siguiente entrada Context -->
  <Context path="" docBase="C:\Program Files\Apache Software Foundation\Tomcat
  <!-- SingleSignOn valve, share authentication between web applications
  Documentation at: /docs/config/valve.html -->
```

4. Para comprobar que funcione, abrir un navegador web, ingresar la URL correspondiente <http://localhost:8082> y se debería ver la página "Hola Mundo" que se creó.



Hola Mundo!

Esta es una página web sencilla desplegada en Apache Tomcat cambiando la carpeta de publicación.

Árbol de archivos para publicación

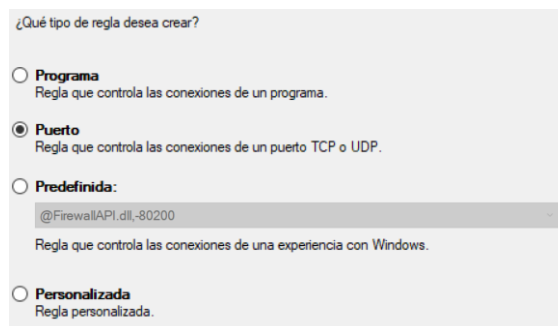
1. Para cada una de las páginas o subpáginas necesarias se tendría una estructura general como se muestra a continuación:

css	31/3/2024 16:32	Carpeta de archivos	
imagenes	31/3/2024 16:32	Carpeta de archivos	
scripts	31/3/2024 16:32	Carpeta de archivos	
index	31/3/2024 15:57	Chrome HTML Do...	1 KB

2. Para publicar en cada una de ellas sería tan solo necesario acceder a el archivo a través del navegador web utilizando la ruta adecuada, por ejemplo, <http://localhost/Pagina%Inicio/Subpagina/index.html>.

Configurar el firewall

1. Abrir el Panel de Control.
2. Seleccionar "Firewall de Windows Defender"
3. Seleccionar "Configuración avanzada"
4. Luego en "Reglas de entrada" y agregar una nueva regla para permitir el tráfico en el puerto deseado.
5. En el asistente, seleccionar el tipo de regla que deseas crear. En este caso, seleccionar "Puerto" y hacer clic en "Siguiente".



6. Seleccionar "Puertos locales específicos" y luego ingresar el número de puerto que deseas permitir. En este caso el puerto 8081 y seleccionar TCP. Hacer clic en "Siguiente".

¿Se aplica esta regla a TCP o UDP?

☒ TCP
☐ UDP

¿Se aplica esta regla a todos los puertos locales o a unos puertos locales específicos?

☐ Todos los puertos locales
☒ Puertos locales específicos:

Ejemplo: 80, 443, 5000-5010

7. Seleccionar "Permitir la conexión" y hacer clic en "Siguiente".

¿Qué medida debe tomarse si una conexión coincide con las condiciones especificadas?

☒ **Permitir la conexión**
Esto incluye las conexiones protegidas mediante IPsec y las que no lo están.

☐ **Permitir la conexión si es segura**
Esto incluye solamente las conexiones autenticadas mediante IPsec. Éstas se protegerán mediante la configuración de reglas y propiedades de IPsec del nodo Regla de seguridad de conexión.

☐ **Bloquear la conexión**

8. Seleccionar los perfiles de red a los que se desea aplicar la regla (pública, privada o de dominio). Por lo general, es seguro seleccionar todas las opciones para garantizar que la regla se aplique en todas las redes. Hacer clic en "Siguiente".

¿Cuándo se aplica esta regla?

☒ **Dominio**
Se aplica cuando un equipo está conectado a su dominio corporativo.

☒ **Privado**
Se aplica cuando un equipo está conectado a una ubicación de red privada, como una red doméstica o del lugar de trabajo.

☒ **Público**
Se aplica cuando un equipo está conectado a una ubicación de redes públicas.

9. Asignar un nombre y, opcionalmente, una descripción para la regla. Luego, hacer clic en "Finalizar". Y listo la regla estará aplicada.

✓ Conexión Puerto 8082	Todo	Sí
✓ adb.exe	Público	Sí

INVESTIGACIÓN

Google Cloud Platform (GCP)

Servicios Ofrecidos

Google Cloud Platform ofrece una amplia gama de servicios en la nube, incluyendo cómputo, almacenamiento, bases de datos, análisis, inteligencia artificial, aprendizaje automático, IoT y más.

Algunos de los servicios destacados incluyen:

- Google Compute Engine para máquinas virtuales.
- Google Cloud Storage para almacenamiento de objetos.
- Google BigQuery para análisis de datos a gran escala.
- Google Cloud SQL para bases de datos relacionales.
- Google Cloud Functions para ejecutar código sin servidor.
- Google Kubernetes Engine para orquestación de contenedores.

Características Distintivas

Infraestructura global: GCP cuenta con una red global de centros de datos que permite a los usuarios implementar aplicaciones cerca de sus usuarios finales para reducir la latencia.

Enfoque en tecnologías de código abierto: GCP tiene un fuerte enfoque en tecnologías de código abierto, como Kubernetes, TensorFlow y Apache Beam.

Servicios de datos avanzados: Ofrece una amplia gama de servicios de datos, incluyendo BigQuery para análisis de datos a gran escala, Firestore para bases de datos NoSQL y Cloud Spanner para bases de datos distribuidas.

Restricciones Tecnológicas

GCP es compatible con una amplia variedad de tecnologías y marcos de desarrollo, incluyendo Python, Java, Node.js, Go, Ruby y más.

También es compatible con contenedores Docker y Kubernetes, lo que permite a los usuarios implementar y administrar aplicaciones en contenedores de manera eficiente.

Precios Aproximados

Las máquinas virtuales en Google Compute Engine comienzan desde alrededor de \$0.0104 por hora.

El almacenamiento en Google Cloud Storage comienza desde alrededor de \$0.020 por GB al mes.

Los precios de BigQuery para análisis de datos comienzan desde alrededor de \$5 por TB procesado.

Interpretación

Google Cloud Platform (GCP) destaca por su infraestructura global, escalabilidad y enfoque en tecnologías de código abierto como Kubernetes. Sus servicios avanzados de datos ofrecen herramientas poderosas para el análisis y la gestión de datos. GCP es compatible con una variedad de tecnologías y lenguajes de programación, lo que brinda flexibilidad a los desarrolladores. Sin embargo, es importante considerar los costos asociados y asegurarse de utilizar herramientas de gestión de costos para optimizar el gasto en la nube. En resumen, GCP ofrece una plataforma sólida y competitiva para aquellos que buscan soluciones en la nube, pero la elección del proveedor dependerá de las necesidades específicas de cada organización.