

# PROGRAMACIÓN WEB (TRABAJO 1 - CONFIGURACIÓN DE SERVIDORES)

Alex Fabián Guamán Carpio  
Universidad de Cuenca  
*alex.guaman@ucuenca.edu.ec*

**Abstract**—Esta actividad tiene como objetivo entender la configuración de servidores web locales, así como también la configuración de servidores de aplicaciones. Además, se persigue entender la diferencia entre un servidor web local y un servidor que permita despliegue en la nube.

## I. MARCO TEÓRICO

### A. Servidor Web

Un servidor web es un software diseñado para almacenar, procesar y entregar páginas web a los clientes que las solicitan a través de internet. Funciona como una plataforma que permite a los usuarios acceder a contenido web, como páginas HTML, imágenes, videos, etc., alojado en un servidor remoto. Los servidores web utilizan protocolos de comunicación estándar, como HTTP o HTTPS, para enviar y recibir datos entre el servidor y el cliente. Además del almacenamiento de archivos y consultas de navegación web, puede ofrecer servicios adicionales como correo electrónico, manejo de bases de datos, seguridad web, servidor proxy y de sistemas de nombres de dominio (DNS).

Algunos ejemplos de servidores web populares son:

- **Apache HTTP Server:** Es uno de los servidores web más utilizados en el mundo. Es de código abierto y es conocido por su estabilidad y flexibilidad. Apache es compatible con una amplia variedad de sistemas operativos, incluyendo Linux, Windows y macOS.
- **Nginx:** Es un servidor web ligero, de alto rendimiento y de código abierto. Se utiliza comúnmente como servidor proxy inverso y para servir contenido estático. Nginx es conocido por su capacidad para manejar grandes cantidades de tráfico web de manera eficiente.
- **Microsoft Internet Information Services (IIS):** Es el servidor web desarrollado por Microsoft para sus sistemas operativos Windows. IIS es ampliamente utilizado en entornos empresariales y es compatible con tecnologías de Microsoft como ASP.NET y aplicaciones web desarrolladas con tecnologías de Microsoft.
- **Lighttpd (Lighty):** Es un servidor web ligero y de alto rendimiento diseñado para manejar grandes cargas de tráfico web. Es conocido por su bajo consumo de recursos y su capacidad para servir contenido estático de manera eficiente.

### B. Servidor de Aplicaciones

Un servidor de aplicaciones es el encargado de proporcionar servicios y funcionalidades a las aplicaciones, lo que permite que estas puedan comunicarse con otras aplicaciones, bases de datos, servidores web y realizar tareas específicas para el correcto funcionamiento de los sistemas.

Un servidor de aplicaciones es como un anfitrión que sirve y gestiona las aplicaciones web para que los usuarios puedan interactuar con ellas. Estos servidores están diseñados para manejar múltiples solicitudes y proporcionar los recursos necesarios para que las aplicaciones funcionen correctamente.

Estos servidores actúan como intermediarios entre el usuario y la aplicación, encargándose de ejecutar la lógica de negocio, procesar datos y proveer respuestas adecuadas.

Los servidores de aplicaciones se basan en la tecnología de Internet Information Services (IIS) o en otras soluciones de servidor web, que les permiten recibir y procesar solicitudes HTTP/HTTPS de los usuarios y proporcionar respuestas en forma de páginas web o servicios.

Algunos ejemplos populares son:

- **Apache Geronimo:** apto para Java EE
- **Apache Tomcat:** para Servlets, JavaServer Pages (JSP) y Java WebSocket
- **ColdFusion:** el servidor de aplicaciones más antiguo
- **GlassFish:** solución de código abierto para Java EE
- **Jetty:** contenedor de servlets y JSP
- **Microsoft Azure App Service:** un servicio en la nube de Microsoft que permite a los desarrolladores hospedar y escalar aplicaciones web y móviles creadas con diferentes tecnologías, incluyendo .NET
- **Resin:** para Java y PHP5
- **WEBrick:** solución basada en Ruby
- **Zope:** un servidor de aplicaciones basado en Python

### C. Hosting

El hosting, o alojamiento web, es un servicio donde podrás alojar todos los contenidos (textos, imágenes, bases de datos, emails) en servidores web para almacenar y publicar contenido en internet, para que tu web y correo sea accesible por cualquier usuario en cualquier momento.

Del mismo modo que un hotel aloja a sus huéspedes en habitaciones, un mismo servidor puede alojar varias webs

con su correo y bases de datos dentro de la misma máquina, a cada uno de esos espacios los llamamos hosting.

El hosting proporciona la infraestructura indispensable, como servidores, almacenamiento, ancho de banda y otras características técnicas, para que el sitio web sea accesible y funcional.

La mayoría de los proveedores de hosting ofrecen diferentes tipos de hosting web tanto alojamiento web gratis como de pago esto para poder satisfacer las necesidades de diferentes clientes, ya sea que desees ser dueño de un gran negocio en línea o crear un blog personal simple.

Los tipos de hosting que hay son:

- **Hosting Compartido:** este tipo de alojamiento es el más empleado. Dentro de un mismo servidor, compartiendo recursos de CPU y RAM, se alojan múltiples hostings. Este servicio está pensado para proyectos ligeros tanto personales como profesionales que no requieren de un uso demasiado intensivo de los recursos del servidor.

Proveedores de hosting compartido:

- Bluehost
- HostGator
- SiteGround

- **Hosting VPS:** en esta modalidad de alojamiento dispones de un servidor virtual con unos recursos de CPU y RAM garantizados para tu alojamiento. Al tener una serie de recursos dedicados para tu web evitas problemas de falta de recursos derivados de los hostings compartidos. Este tipo de alojamiento es el paso intermedio entre una máquina dedicada y el hosting compartido.

Proveedores de hosting VPS:

- DigitalOcean
- Linode
- Vultr

- **Hosting Dedicado:** en este caso dispones de un servidor físico enteramente para tu proyecto. Esto significa que tienes toda la potencia de CPU y RAM disponible del servidor en exclusiva para tu web. Se recomienda este tipo de servicios para web con un alto consumo de recursos.

Proveedores de hosting dedicado:

- Liquid Web
- OVH
- Rackspace

- **Hosting Cloud:** es un servicio de alojamiento donde los recursos de tu máquina se adaptan al consumo de la web. De esta forma, cuando la web tenga pocas visitas, tendrá una cantidad de recursos asignados pequeña y cuando tengas picos de visitas, la potencia de la máquina aumentará para poder seguir ofreciendo un servicio de forma óptima. Los servicios de Cloud están pensados para ofrecer una cantidad de recursos superior a la que puede ofrecer un servidor dedicado y están pensados para las webs más exigentes o los eventos más importantes.

Proveedores de hosting cloud:

- Amazon Web Services (AWS)
- Microsoft Azure
- Google Cloud Platform (GCP)

#### D. Cloud Computing

Cloud computing es la disponibilidad bajo demanda de recursos de computación como servicios a través de Internet. Esta tecnología evita que las empresas tengan que encargarse de aprovisionar, configurar o gestionar los recursos y permite que paguen únicamente por los que usen. El cloud computing proporciona más flexibilidad, pues permite el acceso a los datos a cualquier hora y desde cualquier lugar, algo crucial para equipos deslocalizados o empresas con diferentes demandas. Se puede acceder a este sistema a través de cualquier dispositivo conectado a internet, lo que supone un gran punto a favor para facilitar el trabajo de compañías de diferentes tamaños y esencial en un mundo globalizado.

#### Tipos de Cloud Computing

**Nube privada:** Se trata de un entorno diseñado para una sola organización con su propia nube de servidores y software para la utilización sin un punto de acceso público. No se comparten recursos con otros usuarios fuera de la red y es la más común en un entorno empresarial.

**Nube pública:** Es un entorno de nube creado a partir de recursos ajenos al usuario final y donde diversas empresas pueden usar de manera simultánea, pero separadamente. Se alquila a un proveedor de hosting cloud y donde los recursos pueden ser compartidos. Ejemplos de nube pública son Microsoft Azure, Amazon Web Services (AWS) y Google Cloud Platform (GCP).

**Nube híbrida:** Está compuesta por dos o más infraestructuras de nubes distintas que se mantienen como entidades únicas, pero que pueden estar unidas por tecnología estandarizada o propia. Tiene cierto nivel de portabilidad, organización y gestión de las cargas de trabajo común y cambia de pública a privada según necesidad.

**Multiclouds:** También conocida como nube comunitaria, se trata de un sistema compuesto por más de una nube (pública o privada) que se puede conectar en red. Es muy común entre diferentes empresas u organizaciones que reúnen en un mismo pool sus recursos en la nube para resolver un problema común.

Los ejemplos de cloud computing son numerosos y se clasifican según el tipo de utilidades que ofrece la nube, como por ejemplo:

- **Backup como servicio de cloud computing:** consiste en alojar copias de seguridad de la información en la nube. De esta forma, si hay cualquier contratiempo, es posible recuperar esos datos con facilidad.
- **Procesamiento y almacenamiento de datos:** Las empresas generan cada día datos y datos que, tras procesar, almacenar y analizar, ayudan a descubrir nuevas tendencias de negocio.

- **Gestión de bases de datos:** permite el acceso más rápido a la información y desde múltiples puntos.
- **CRM:** Los CRM (customer relationship management o gestión de relaciones con el cliente) son softwares que administran el vínculo que la empresa tiene con el cliente. Se aloja gran cantidad de información sobre la empresa que, al analizarla, resulta ideal para las acciones de los departamentos de Ventas y Marketing.
- **Escritorios virtuales:** A través de un escritorio virtual, un usuario accede a un sistema operativo alojado en la nube. De esta forma es posible, por ejemplo, entrar en el ordenador de la oficina desde casa.

#### E. Despliegue continuo

El despliegue continuo permite cumplir con el concepto de producción continua que persigue la metodología DevOps. Todo el proceso de despliegue sigue una serie de pasos que deben ejecutarse en orden y de forma correcta. Si alguno de estos pasos no se concluye de forma satisfactoria, el despliegue no se llevará a cabo. Por eso es fundamental que el diseño de la automatización de pruebas se realice de forma correcta, pues al no producirse ninguna entrada o acción manual, dependerá en gran medida de cómo sea ese diseño. El despliegue continuo libera de carga a los equipos de operaciones de procesos manuales, que son una de la principal causa de retrasos en la distribución de aplicaciones.

El uso de estas prácticas de mejora en los procesos de desarrollo de software aporta una serie de beneficios como la entrega o liberación inmediata de código, una simplificación del trabajo colaborativo o de equipo, permite una detección temprana de errores, disminuye costes de desarrollo (ahorra tiempo y esfuerzo) y favorece la comunicación entre todos los implicados en el proceso de desarrollo.

Algunos ejemplos de empresas que utilizan el despliegue continuo son:

**Amazon:** Utiliza el despliegue continuo en muchos de sus servicios en la nube, como Amazon Web Services (AWS), para lanzar actualizaciones de forma rápida y eficiente.

**Netflix:** Utiliza el despliegue continuo para implementar cambios de forma rápida y segura en su plataforma de transmisión de video. Esto les permite ofrecer nuevas características y mejoras a sus clientes de manera constante.

**Spotify:** Utiliza el despliegue continuo para implementar cambios en su plataforma de manera frecuente y sin problemas. Esto les permite experimentar con nuevas características y recopilar retroalimentación de los usuarios de manera rápida.

**Google:** Utiliza el despliegue continuo en muchos de sus productos y servicios, como Google Search, Gmail y Google Drive. Esto les permite implementar cambios de manera rápida y eficiente en una amplia gama de productos y servicios.

## II. DESARROLLO

### A. Servidores Locales

Para la práctica se usó XAMPP, primero lo instalamos (Fig. 1):

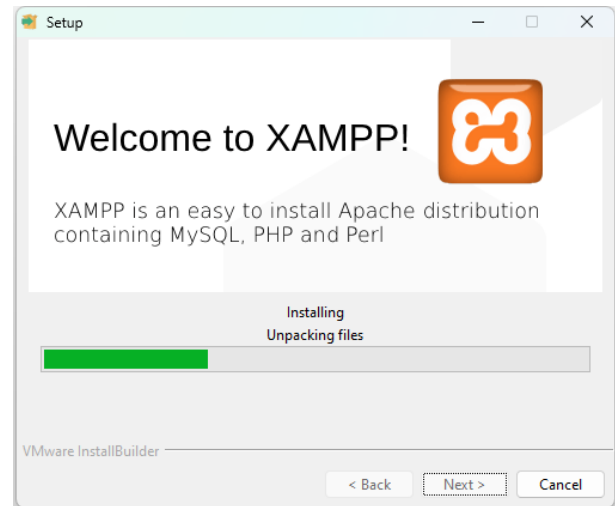


Fig. 1. Instalación de XAMPP

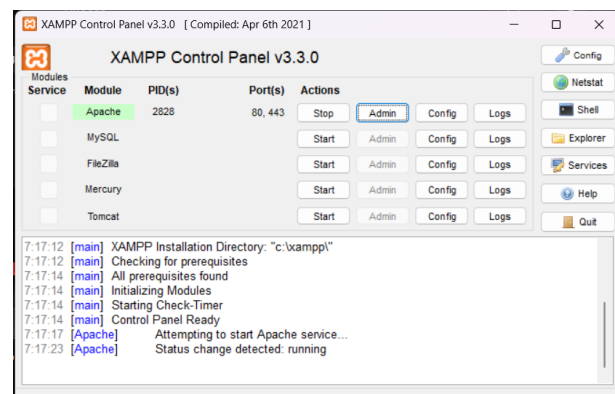


Fig. 2. Inicio de Servidor Apache

Ponemos a correr el servidor Apache y en la Fig. 2 se observa que se inicia el servidor en el puerto 80, procederemos a cambiar el puerto para lo cual abrimos los archivos de configuración correspondientes al servidor que estamos usando en este caso httpd.conf para Apache. Buscamos la sección donde se especifican los puertos y cambiamos el puerto predeterminado a 8083 (Fig. 3) e iniciamos nuevamente el servidor.

En la Fig. 4 entramos en el servidor y podemos verificar con la dirección <https://localhost:8083> que el servidor está iniciado correctamente.

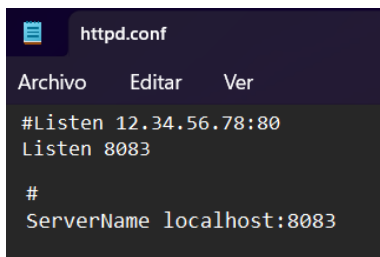


Fig. 3. Cambio de Puerto Servidor Apache

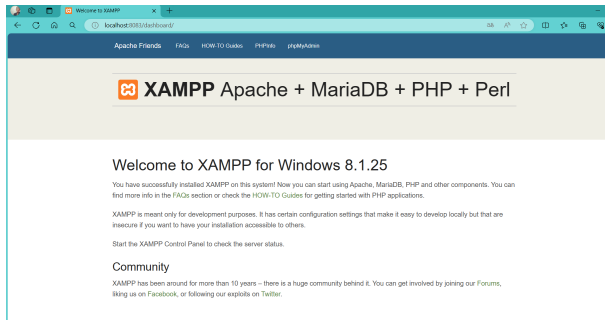


Fig. 4. Página inicial Apache

Ahora desplegaremos un archivo HTML simple que contiene el mensaje "Hola Mundo". Guardaremos el archivo en la carpeta de documentos raíz del servidor web en este caso htdocs para Apache. A su vez creamos una carpeta llamada "Carpeta Pública" en el directorio raíz del servidor web, podemos configurar el servidor en los directorios de acceso para especificar la ubicación de la carpeta pública y configurarla como la carpeta de publicación. Para comprobar si podemos acceder a la página web en el navegador y verificar que se ha desplegado correctamente Fig. 5.

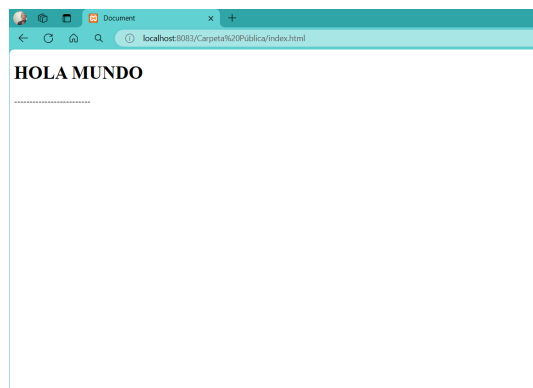


Fig. 5. Despliegue en Apache

Ahora usaremos el servidor Apache Tomcat el cuál podemos instalarlo directamente con su instalador desde su página web <https://tomcat.apache.org>, pero también podemos iniciarlo directamente con XAMPP.

Por defecto usa el puerto 8080 (Fig. 6) y lo cambiaremos igualmente en este caso al puerto 8086.

Para cambiar el puerto primero nos ubicamos en el directorio de configuración de Apache Tomcat. Por lo general, se

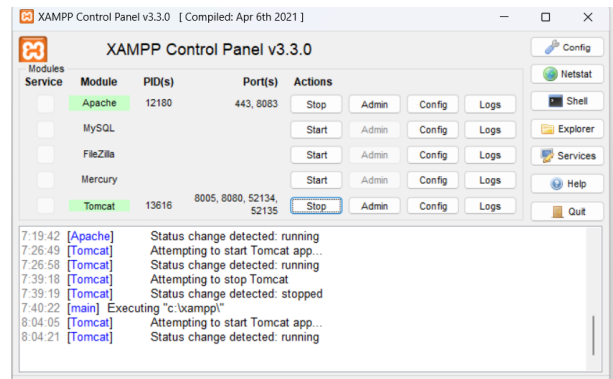


Fig. 6. Inicio de Servidor Apache Tomcat

encuentra dentro de la carpeta de instalación de Tomcat y se llama conf.

Dentro de la carpeta conf, abriremos un archivo llamado server.xml. Este archivo contiene la configuración del servidor Tomcat, incluyendo la configuración de los puertos. En el archivo buscamos la Configuración del Conector HTTP (Fig. 7).

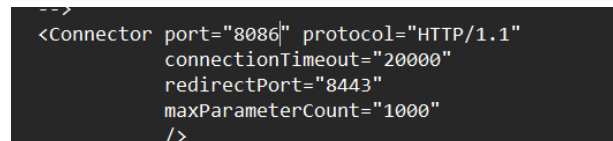


Fig. 7. Cambio de puerto Servidor Apache Tomcat

La línea <Connector> define el conector HTTP de Tomcat, y el atributo port especifica el puerto en el que Tomcat escuchará las solicitudes HTTP entrantes. Cambiamos el puerto a 8086, guardamos los cambios y reiniciamos el servidor para que los cambios surtan efecto.

En Tomcat, los scripts del lado del servidor suelen estar escritos en Java y se despliegan como archivos WAR (Web Application Archive). Usaremos un ejemplo de "Hola Mundo" en Java para usarlo. Para hacer el despliegue podemos hacerlo usando el administrador de Tomcat (Fig. 8).



Fig. 8. Administrador Servidor Apache Tomcat

En la parte de "WAR file to deploy", en "Choose File" seleccionamos el archivo WAR, hacemos clic en el botón "Deploy" para iniciar el proceso de despliegue. Tomcat desplegará el archivo WAR y lo hará accesible como una aplicación web en el servidor. Verificamos el Despliegue utilizando la URL correspondiente (Fig. 9).

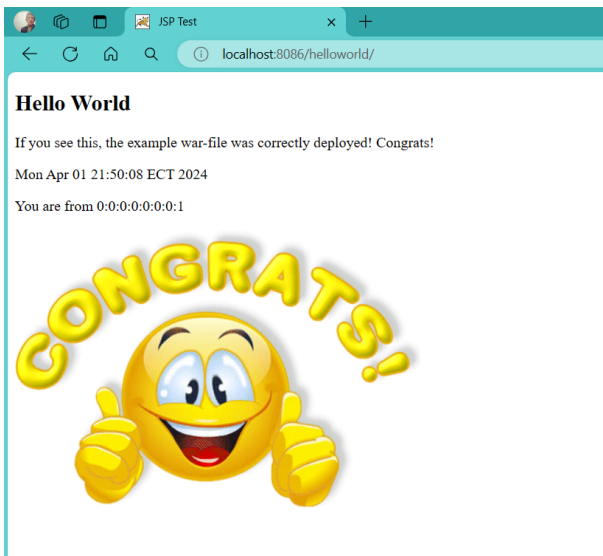


Fig. 9. Despliegue en Apache Tomcat

## Configuración del Firewall

Para evitar conflictos en los puertos podemos configurar el firewall en nuestro ejemplo como es en windows configuramos el firewall entrando en: Panel de Control -> Firewall de Windows -> Configuración Avanzada. Agregamos una regla de entrada que permita el tráfico en el puerto que elegimos para el servidor web en nuestro caso 8083 y 8086 creamos la regla para permitir el tráfico tanto en TCP como en UDP.

## Estructura de Carpetas

Para la estructura de carpetas y archivos cuando el número de estos es considerable, resulta muy útil ubicarlos en carpetas para facilitar su localización y edición. Con carácter general se proponen un modelo basado en la organización por tipos de archivos.

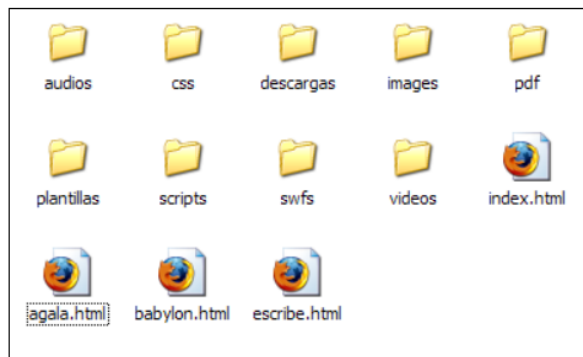


Fig. 10. Estructura de carpetas de un sitio web

Las páginas HTML se guardarán en el directorio o carpeta principal mientras que los elementos que utilizan (audios, vídeos, hojas de estilo, imágenes, etc) se situarán en las

subcarpetas correspondientes.

Esta organización está especialmente pensada para un sistema de páginas HTML que comparten recursos: imágenes, hoja de estilo, javascript, etc.

Las principales carpetas que utiliza son:

**audios:** archivos de audio mp3.

**css:** hojas de estilo css.

**descargas:** archivos zip, exe, etc que se ofrecen para descarga.

**images:** imágenes jpg, gif o png.

**pdf:** documentos pdf.

**scripts:** archivos js con código javascript reutilizable.

**swfs:** archivos con animaciones flash (\*.swf).

**vídeos:** ficheros en distintos formatos de vídeo: \*.wmv, \*.flv, \*.mp4

## B. Servidor Cloud

El servidor que se revisará en esta práctica es Microsoft Azure la cuál es una plataforma compuesta por más de 200 productos y servicios, entre los cuales encuentras infraestructuras (IaaS: redes, máquinas virtuales y almacenamiento) y plataformas (PaaS: bases de datos SQL, CMS para desarrollo de web y backend). Puede hospedar aplicaciones existentes y simplificar el desarrollo de nuevas aplicaciones. Azure puede incluso mejorar las aplicaciones locales, Azure integra los servicios en la nube que necesita para desarrollar, probar, implementar y administrar aplicaciones y, todo ello, mientras aprovecha las ventajas de la informática en la nube.

Funciona con una gran selección de sistemas operativos, lenguajes de programación y herramientas. Todas las tecnologías necesarias que utilizan la mayoría de desarrolladores y profesionales de TI alrededor del mundo.

Con el hospedaje de las aplicaciones en Azure, puede empezar con tamaño pequeño y escalar fácilmente su aplicación a medida que aumente la demanda de los clientes. Microsoft Azure ofrece un período de prueba gratuito que permite a los usuarios probar varios servicios y funcionalidades de la plataforma durante un período limitado, con un crédito de \$200 dólares para gastar en servicios durante ese tiempo. Lo que se puede hacer en el período de prueba tiene ciertas limitaciones y restricciones que los usuarios deben considerar.

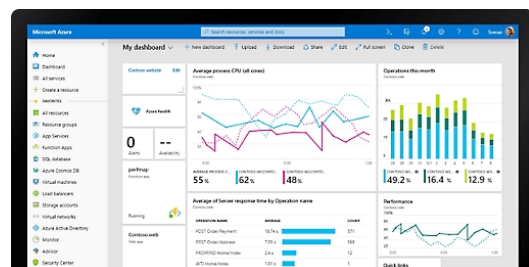


Fig. 11. Dashboard de Azure

En la documentación podemos encontrar la lista de todos los servicios que Azure ofrece en: <https://learn.microsoft.com/es-es/azure/?product=popular>

Se podría decir que el período de prueba gratuito de Microsoft Azure ofrece una excelente oportunidad para explorar y familiarizarse con los servicios en la nube de Azure sin incurrir en costos significativos. Sin embargo, se debe tener en cuenta las limitaciones y restricciones del período de prueba, así como comprender los costos potenciales una vez que finalice el período de prueba, el cuál se puede calcular a través de la misma página de Microsoft Azure.

#### REFERENCES

- [1] Equipo editorial de IONOS. “¿Qué son los servidores de aplicaciones?” IONOS Digital Guide. Accedido el 29 de marzo de 2024. [En línea]. Disponible: <https://www.ionos.es/digitalguide/servidores/know-how/servidor-de-aplicaciones/>
- [2] “Qué es un servidor web, para qué sirve, cómo funciona y ejemplos”. Blog de HubSpot | Marketing, Ventas, Servicio al Cliente y Sitio Web. Accedido el 29 de marzo de 2024. [En línea]. Disponible: <https://blog.hubspot.es/website/que-es-servidor-web>
- [3] “¿Qué es un hosting y cómo funciona?” Tutoriales Hostinger. Accedido el 29 de marzo de 2024. [En línea]. Disponible: <https://www.hostinger.es/tutoriales/que-es-un-hosting#:text=Un>
- [4] “Qué es el cloud computing? | Tipos, ejemplos, servicios...”. Plain Concepts. Accedido el 29 de marzo de 2024. [En línea]. Disponible: <https://www.plainconcepts.com/es/que-es-cloud-computing/>
- [5] “Integración continua, entrega continua y despliegue continuo”. Ilimit Pre. Accedido el 1 de abril de 2024. [En línea]. Disponible: <https://ilimit.com/blog/integracion-continua-entrega-continua-despliegue-continuo/>