

# PROGRAMACIÓN WEB

## TRABAJO 1

### CONFIGURACIÓN DE SERVIDORES

Meneses Andrés  
 Universidad de Cuenca  
 Ingeniería en Ciencias de la Computación  
 andres.meneses@ucuenca.edu.ec

**Resumen**— El documento aborda la configuración y gestión de servidores web y de aplicaciones, explorando específicamente los servidores locales Apache y Apache Tomcat, así como las opciones de servidores en la nube como Heroku, Microsoft Azure, Google Cloud Platform y Amazon Web Services (AWS). Se examinan aspectos clave como la modificación de puertos, el despliegue de páginas web, la configuración de firewall y las características ofrecidas por cada servicio en la nube. Se reflexiona sobre las ventajas y desafíos de cada opción, subrayando la importancia de evaluar las necesidades del proyecto y considerar la escalabilidad y adaptabilidad a largo plazo. El estudio proporciona una guía práctica para la implementación efectiva de servidores web y de aplicaciones, tanto locales como en la nube.

#### I. INTRODUCCIÓN

En el presente trabajo, se abordará la configuración y gestión de servidores web y de aplicaciones, fundamental para el desarrollo y despliegue efectivo de aplicaciones y sitios web. Se podrá enfocar especial en configurar tres tipos diferentes de servidores: Apache, Apache Tomcat y un servidor en la nube. Se comenzará definiendo y explicando conceptos clave como servidor web, servidor de aplicaciones, hosting, cloud computing y despliegue continuo, para establecer una base sólida de comprensión antes de entrar en los detalles de la configuración y los ejercicios prácticos. Posterior a ello se procederá a realizar configuraciones de los servidores locales, incluyendo la modificación de puertos, el despliegue de una página web sencilla, la creación de una carpeta pública para la publicación de contenido, la estructura recomendada de un árbol de archivos para la publicación y la configuración del firewall para restringir el acceso a los puertos específicos. Para finalizar con la configuración de un servidor en la nube, investigando opciones de prueba gratuitas o con limitaciones de transacciones, revisando costos, restricciones tecnológicas y características ofrecidas, para luego reflexionar sobre las ventajas y desafíos que presenta esta opción en comparación con los servidores locales.

#### II. MARCO TEÓRICO

##### A. Servidor Web - Ejemplos

Los servidores web son un ordenador de gran potencia que tienen como principal función almacenar, en web hosting, todos los archivos propios de una página web (imágenes, textos, videos, etc.) y transmitirlos a los usuarios a través de los navegadores mediante el protocolo HTTP. Un servidor web es un software diseñado para procesar solicitudes HTTP de clientes, como navegadores web, y enviarles recursos, como páginas web, imágenes, archivos, etc. Básicamente, es el encargado de alojar y distribuir el contenido de un sitio web a los usuarios que lo soliciten a través de Internet.

Ejemplos de servidores web:

- Apache HTTP Server: Es uno de los servidores web más antiguos y populares, de código abierto y muy utilizado en todo tipo de sistemas operativos.



Figura 1: Logo Apache HTTP SERVER

- Nginx: Es un servidor web y proxy inverso de alto rendimiento, conocido por su eficiencia y escalabilidad.



Figura 2: Logo Nginx

- Microsoft Internet Information Services (IIS): Es el servidor web desarrollado por Microsoft para sus sistemas operativos Windows, comúnmente utilizado en entornos empresariales.



Figura 3: Logo IIS

- LiteSpeed Web Server: Es un servidor web de alto rendimiento conocido por su velocidad y eficiencia



Figura 4: Logo LiteSpeed Web Server

- Google Web Server (GWS): Es el servidor web desarrollado internamente por Google para alojar sus servicios y sitios web. No disponible públicamente.



Figura 5: Web Server

Dentro de las características explícitas se encuentran la encapsulación de las fuentes de datos, interfaces con otros servicios de mayor valor, comunicación asíncrona, conservación de la persistencia, servicios de directorio, normas para la conexión de bases de datos; mientras que de las características implícitas incluyen la mejor escalabilidad sin modificaciones adicionales de las aplicaciones individuales, monitorización, funciones de registro, opciones de calibración, gestión del tiempo de ejecución, gestión del ciclo de vida del software mediante parcheado, actualización, distribución e instalación.

Ejemplos de servidores de aplicación:

- Apache Tomcat: Es uno de los servidores de aplicaciones de código abierto más utilizados. Está diseñado para ejecutar aplicaciones web Java Servlet y JavaServer Pages (JSP).



Apache Tomcat

Figura 6: Logo Apache TomCat

- WildFly: Servidor de aplicaciones de código abierto desarrollado por Red Hat. Soporta una amplia gama de tecnologías Java EE, como Servlets, JSP, CDI y muchas más.



Figura 7: Logo WildFly

### B. Servidor de Aplicaciones - Ejemplos

Es un componente fundamental en la arquitectura de aplicaciones web; son plataformas de software diseñadas específicamente para ejecutar y gestionar aplicaciones web complejas y dinámicas que proporcionan software a un cliente en una red con mayor seguridad y compatibilidad. El enfoque principal de un servidor web es permitir la interacción entre usuarios y la lógica de la aplicación del lado del servidor.

Un servidor de aplicaciones suele alojar distintos programas de aplicación y los pone a disposición de los clientes. Para ello, utiliza la lógica empresarial del lado del servidor para generar contenido dinámico y mostrarlo al cliente. Algunos ejemplos típicos de software que se encuentran en un servidor de aplicaciones son los programas ofimáticos, la gestión de direcciones, los calendarios corporativos y el acceso a bases de datos. Los procesos de carácter confidencial, como las transacciones o las autenticaciones, también pueden realizarse a través de un servidor de aplicaciones.

Los servidores de aplicaciones ofrecen diversas funciones para simplificar y mejorar los procesos, dentro de estos es importante diferenciar entre funciones explícitas e implícitas.

- IBM WebSphere Application Server: Es un servidor de aplicaciones Java EE desarrollado por IBM. Es conocido por su robustez, escalabilidad y capacidades de administración avanzadas.



Figura 8: Logo WebSphere Application Server

- Oracle WebLogic Server: Es otro servidor de aplicaciones Java EE de nivel empresarial desarrollado por Oracle.



Figura 9: Logo Oracle WebLogic Server

- GlassFish: Es un servidor de aplicaciones de código abierto y una implementación de referencia de Java EE.



**GlassFish**

Figura 9: Logo GlassFish

### B. Hosting

Un hosting es un servicio de alojamiento web que te permite publicar un sitio web o aplicación en Internet. Al contratar un servicio de hosting, básicamente alquilas un espacio en un servidor físico donde puedes almacenar todos los archivos y datos necesarios para que tu sitio web funcione correctamente. Para explicar cómo funciona el hosting o anclaje web, primero se debe tener claro que un servidor es una computadora o máquina física que funciona ininterrumpidamente para que tu sitio web esté disponible todo el tiempo para cualquier persona que quiera verlo. Ahora, un proveedor de hosting web es el responsable de mantener el servidor en funcionamiento, protegerlo de ataques maliciosos y transferir tu contenido (como texto, imágenes, etc.) desde el servidor a los navegadores de tus visitantes. En conclusión, siendo así, el hosting, el permite que una página web sea vista por cualquier usuario con conexión a Internet.

Existen distintos tipos de hosting: Hosting Compartido, Hosting VPS, Cloud Hosting, Hosting WordPress, Servidor dedicado.

El hosting es esencial para que un sitio web esté disponible en línea. Al elegir un proveedor de hosting, considera factores como el panel de control, el tiempo de disponibilidad, las especificaciones técnicas y la seguridad



Figura 10: Imagen de Hosting, idea gráfica de su significado.

### C. Cloud Computing

Es la disponibilidad bajo demanda de recursos informáticos como servicios a través de Internet, ofreciendo soluciones eficientes, seguras, flexibles y rentables para diversas necesidades empresariales. Este término, hace referencia a la computación en la nube, que permite el acceso remoto a software, almacenamiento de archivos y procesamiento de datos a través de Internet. Cabe mencionar que la nube proporciona a individuos y empresas un conjunto de recursos informáticos bien mantenidos, seguros, de fácil acceso y disponibles según la demanda.

Su funcionamiento se da a parte de utilizar una capa de red para conectar dispositivos periféricos a recursos centralizados en centros de datos, estos recursos incluyen servidores, almacenamiento y servicios de aplicaciones.

En sí, la computación en la nube es una tecnología que ofrece servicios a través de Internet, permitiendo a las organizaciones acceder a recursos informáticos de manera eficiente y segura; sin la necesidad de instalar aplicaciones localmente en computadoras, sino todo siendo ejecutado en servidores remotos; un modelo de nube que esencialmente democratiza el acceso a recursos a nivel internacional.



Figura 11: Imagen de Cloud Computing, idea gráfica de su significado.

### D. Despliegue Continuo

El despliegue continuo es una práctica en el desarrollo de software que se enfoca en automatizar y optimizar el proceso de entrega de aplicaciones, permitiendo que los cambios en el código se implementen de manera rápida y frecuente en un entorno de producción. Este enfoque busca reducir el tiempo y los riesgos asociados con el despliegue manual de aplicaciones, al permitir que los desarrolladores entreguen nuevas características o correcciones de errores de forma continua y sin interrupciones. A través del despliegue continuo, las empresas pueden mejorar la velocidad de lanzamiento de sus productos, obtener retroalimentación más rápida de los usuarios y mantener un alto nivel de calidad en sus aplicaciones. Este proceso se apoya en prácticas como la integración continua, la automatización de pruebas y la implementación automatizada, que garantizan la estabilidad y confiabilidad del software desplegado.

El despliegue continuo es una práctica en el desarrollo de software donde los cambios de código se prueban y publican automáticamente en el entorno de producción, basándose en una serie de pruebas predefinidas. Una vez que las

actualizaciones pasan estas pruebas, el sistema las envía directamente a los usuarios del software, lo que agiliza el tiempo de comercialización y elimina la necesidad de procesos manuales. Se diferencia de la entrega continua en que amplía la automatización y elimina la intervención humana en la aprobación de despliegues en producción, garantizando que el código nuevo se considere aprobado si pasa las pruebas y se inicie el despliegue en producción. La integración continua es clave para asegurar un despliegue continuo sin fisuras, ya que permite a todos los desarrolladores comunicar eficazmente los cambios en el proyecto.

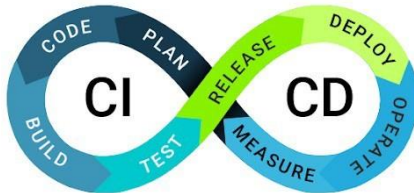


Figura 12: Imagen de Despliegue Continuo, idea gráfica de su significado.

### III. PRÁCTICA

#### A. Configurar 3 tipos de servidores

- **APACHE (WAMP O XAMPP):** Puede ser en Windows, Linux o macOS, no importa la distribución.
  1. **Instalación de Apache:** Se instalará XAMPP desde el sitio oficial y se descargará la versión adecuada para Windows.

#### Download

XAMPP is an easy to install Apache distribution containing MariaDB, PHP, and Perl. Just download and start the installer. It's that easy.

XAMPP for Windows 8.0.30, 8.1.25 & 8.2.12			
Version	Checksum	Size	
8.0.30 / PHP 8.0.30	What's Included? md5 sha1	Download (64 bit) 144 Mb	
8.1.25 / PHP 8.1.25	What's Included? md5 sha1	Download (64 bit) 148 Mb	
8.2.12 / PHP 8.2.12	What's Included? md5 sha1	Download (64 bit) 149 Mb	

Requirements More Downloads »

Windows XP or 2003 are not supported. You can download a compatible version of XAMPP for these platforms here.

Figura 13: Descargar Xampp para Windows

2. **Iniciar XAMPP Y Apache** haciendo click en el botón start.



Figura 14: Iniciar XAMPP y Apache

3. **Acceder al directorio de instalación de XAMPP.**

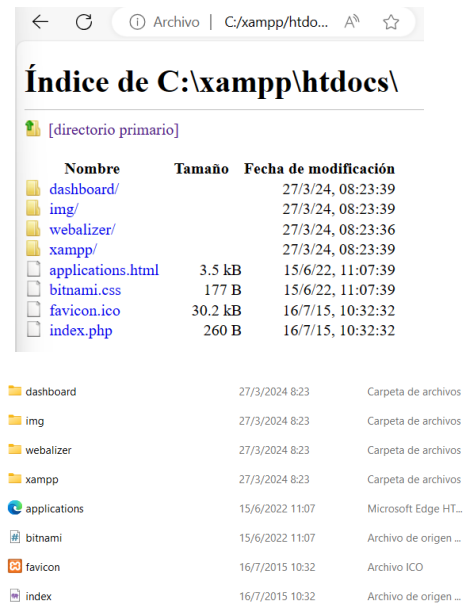


Figura 15: Imagen del directorio htdocs dentro de XAMPP.

4. **Acceder al sitio web.**



Figura 16: Acceso al sitio web.

- **APACHE TOMCAT:** Buscar un script del lado del servidor de hola mundo y desplegarlo.
  1. **Descargar e instalar TomCat**

apache-tomcat-10.1.20 27/3/2024 8:40

Figura 17: Carpeta Descargada TomCat

2. **Configurar el Apache Tomcat y desarrollar un script con el "Hola Mundo!".**





Figura 18: Hola Mundo en TomCat

- UN SERVIDOR EN CLOUD COMPUTING: Puede ser en Azure, Heroku, Google Engine, Amazon Web Services, o el que usted elija.

1. En la presente práctica se decidió buscar cada servidor expuesto como ejemplo, habiéndose creado cuentas en todos ellos; sin embargo, se notó la particularidad de que en todos hay que pagar.
2. Para conectar a la máquina virtual se puede conectar a ella mediante un cliente de acceso remoto para Windows y de allí configurarle al software y servicios necesarios para la aplicación o sitio web.

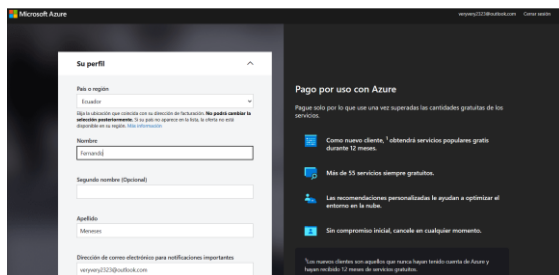


Figura 19: Azure, Servidor Cloud Computing

#### IV. EJERCICIOS A REALIZAR

Tutorial paso a paso con los ejercicios de la práctica.

##### A. Servidores Locales: Cambiar los puertos para que funcionen en un rango del 8080 al 8089

1. Abre la configuración de Apache en XAMPP.

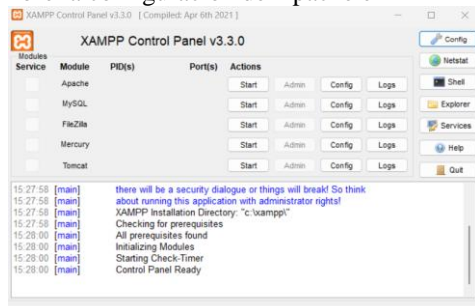


Figura 20: Abrir XAMPP

2. En el panel de control de XAMPP, haz clic en el botón "Config" junto al módulo de Apache.

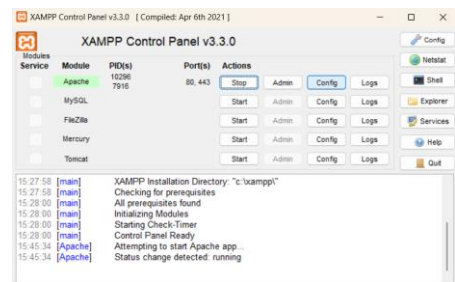


Figura 21: Presionar el botón "Config" en XAMPP

3. Selecciona "httpd.conf" para editar la configuración de Apache.

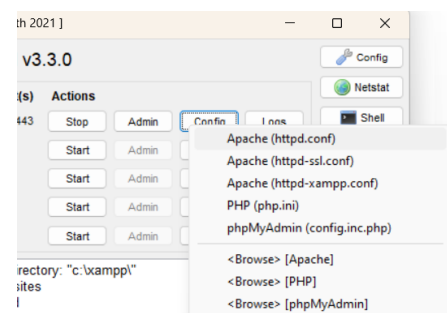


Figura 22: Selección de Apache httpd.conf

4. Busca la línea que comienza con **Listen** seguida por un número de puerto (generalmente 80). Cambia este número a un valor dentro del rango del 8080 al 8089. Por ejemplo, puedes cambiarlo a **Listen 8080**.

```

# Listen: Allows you to bind Apache to specific IP add
# ports, instead of the default. See also the <VirtualH
# directive.
#
# Change this to listen on specific IP addresses as sho
# prevent Apache from glomming onto all bound IP addre
#
#Listen 12.34.56.78:80
Listen 8080

```

Figura 23: Cambiado Listen 8080

5. Guarda los cambios y cierra el editor. Finalmente reinicia Apache en XAMPP para que los cambios surtan efecto.

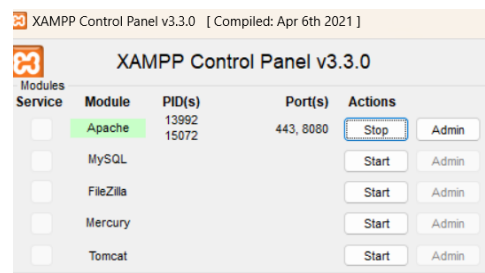


Figura 24: Se verifica que el puerto ha sido modificado en el valor de 8080

B. Servidores Locales: Desplegar una página web sencilla de "Hola Mundo" en cada servidor.

1. Crea un archivo HTML llamado **index.html** con el contenido de un “hola mundo” en html.

```
index.html X
C: > apache-tomcat-10.1.20 > webapps > HolaMundo > index.html > ...
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Hola Mundo</title>
5 </head>
6 <body>
7   <h1>Hola Mundo!</h1>
8 </body>
9 </html>
10
```

Figura 25: Se verifica que el puerto ha sido modificado en el valor de 8080

2. Coloca este archivo en el directorio **htdocs** de XAMPP para Apache y en el directorio de despliegue de Tomcat para Apache Tomcat.

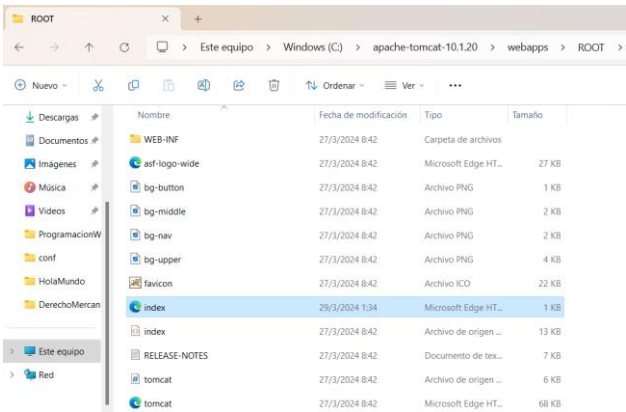


Figura 26: El index.html dentro de Apache TomCat dentro de ROOT.

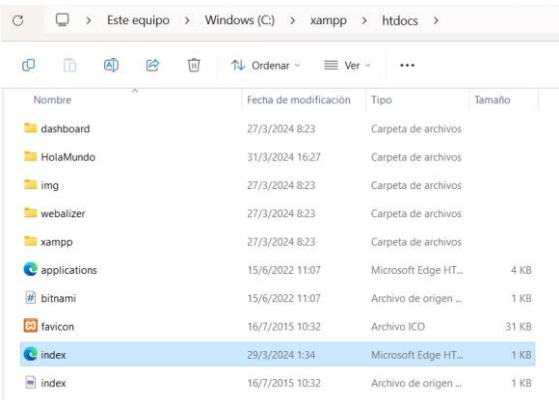


Figura 26: El index.html dentro de XAMPP, dentro de htdocs.

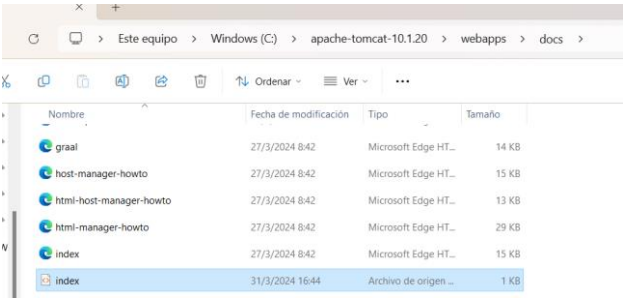


Figura 28: El index.jsp dentro de Apache TomCat dentro de docs

3. Abre un navegador web y accede a la página utilizando la URL correspondiente para cada servidor.



Figura 29: URL: <http://localhost:8080/index.html>

C. Servidores Locales: Crear una carpeta llamada “Carpeta Pública” y configurarla como la carpeta de publicación.

1. Crea una carpeta llamada "Carpeta Pública" en el directorio raíz de cada servidor local.

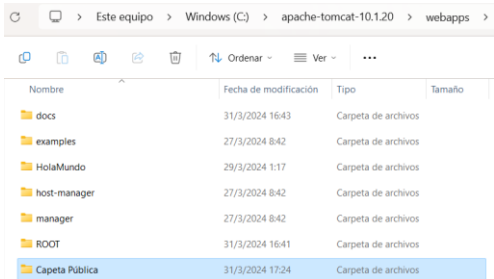


Figura 30: C:\Tomcat\apache-tomcat-<version>\webapps\

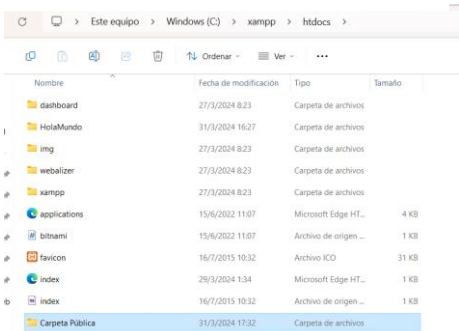


Figura 31: C:\xampp\htdocs\

$a + b = c.$

- Abre el archivo de configuración de cada servidor y busca la configuración del directorio raíz.

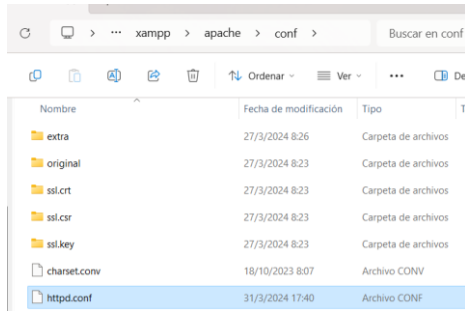


Figura 32: "C:\xampp\apache\conf\httpd.conf"

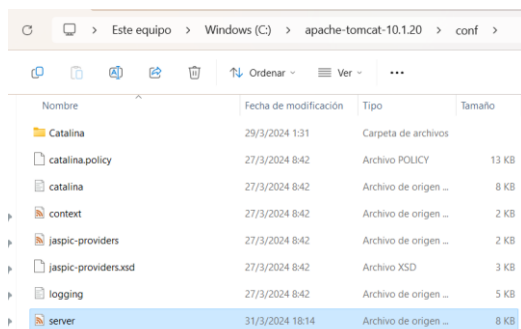


Figura 33: "C:\apache-tomcat-10.1.20\conf\server.xml"

- Cambia la configuración del directorio raíz para que apunte a la carpeta "Carpeta Pública" que acabas de crear.

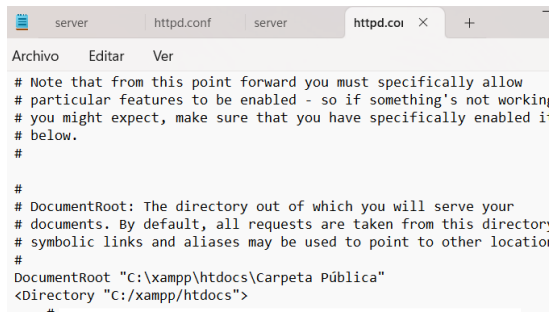


Figura 34: Configuración cambiada en XAMPP

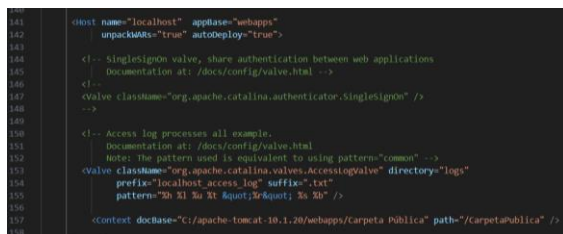


Figura 35: Configuración cambiada en TomCat

- Guarda los cambios y reinicia cada servidor para que los cambios surtan efecto.

*D. Servidores locales: Buscar la estructura recomendada de un árbol de archivos para publicación (En donde guarden imágenes, frontend, backend, etc.) Crear la estructura de carpetas y probar qué hacer para publicar en cada una de ellas. Por ejemplo: En la universidad tuviera una carpeta en donde esté la página de inicio y una sub carpeta de cada facultad y dentro de ella carpetas de imágenes, css, etc. (consultar en la bibliografía convenciones básicas de estructuras de carpetas).*

Una estructura de carpetas es bastante común y proporciona una organización clara y fácil de entender para los diferentes tipos de archivos web. Hay que tener en cuenta que se puede adaptar una estructura según las necesidades específicas de un proyecto y no simplemente un solo consenso respecto a qué estructura de archivos se debería usar.

Un ejemplo podría ser el siguiente:

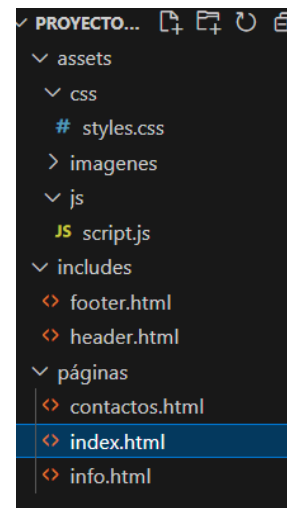


Figura 36: Ejemplo de estructura de carpetas

Para garantizar la coherencia y facilidad de mantenimiento en la estructura de carpetas de un proyecto web, es fundamental seguir algunas pautas clave. En primer lugar, los nombres de las carpetas deben ser descriptivos y reflejar claramente su contenido, lo que facilita la identificación rápida de los archivos necesarios. Por ejemplo, las carpetas como `css/`, `js/`, y `img/` deben contener estilos CSS, scripts JavaScript y archivos de imágenes, respectivamente. Además, la organización lógica dentro de cada carpeta es esencial para mantener la claridad y facilitar la navegación; por ejemplo, los estilos relacionados con la página de inicio podrían estar en una subcarpeta `inicio/` dentro de `css/`. La estructura de carpetas debe promover la facilidad de acceso y el mantenimiento sencillo del proyecto. Esto significa que los archivos comunes utilizados en todo el sitio, como encabezados y pies de página, deben estar ubicados en una carpeta centralizada, como `includes/`, para facilitar su modificación. Además, la

documentación clara, como un archivo `README.md`, puede proporcionar orientación sobre la estructura de carpetas y cómo navegar por el proyecto. Al seguir estas pautas, se garantiza que la estructura de carpetas sea coherente, fácil de mantener y eficiente para el desarrollo y la gestión del proyecto web.

### E. Servidores Locales: Configurar el firewall para que sirva exclusivamente en el puerto indicado:

1. Abre la configuración del firewall en tu sistema operativo.

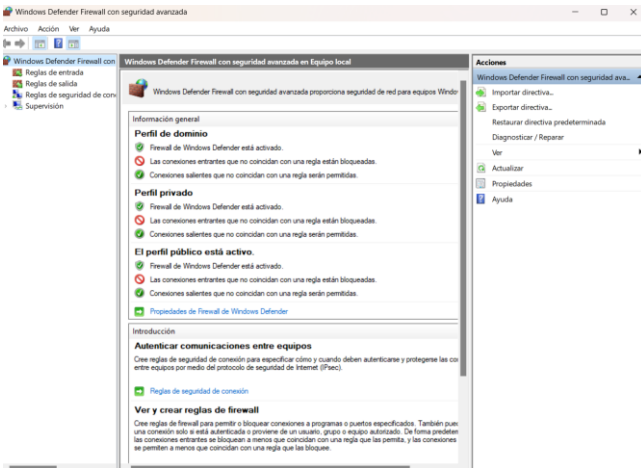


Figura 37: Se busca Windows defender firewall en el sistema operativo y se procede a abrir la configuración avanzada.

2. Agrega una regla de entrada que permita el tráfico en el puerto especificado (por ejemplo, el puerto 8080).

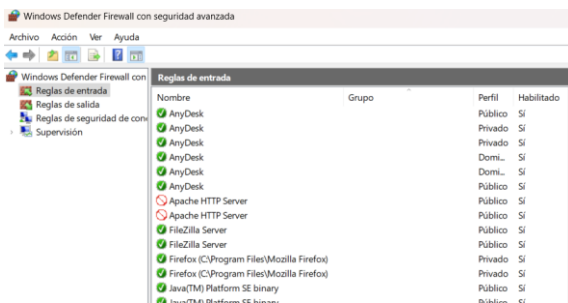


Figura 38: Se procede a localizar la Regla de entrada para crear una nueva regla.

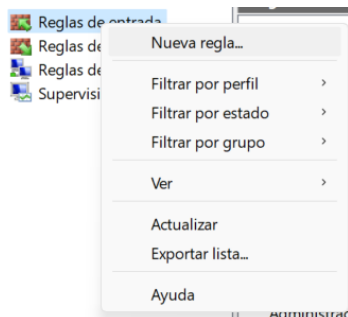


Figura 39: Click derecho en reglas de entrada y se hace click en Nueva regla

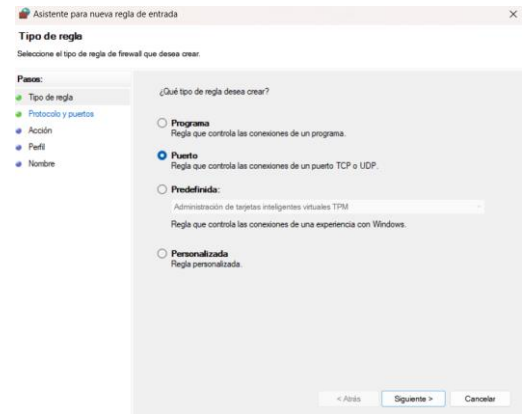


Figura 40: Se elige el tipo de puerto y se la da siguiente.

3. Asegúrate de que esta regla solo permita el tráfico en el puerto especificado y que esté configurada para el tipo de red adecuado (público, privado, etc.).

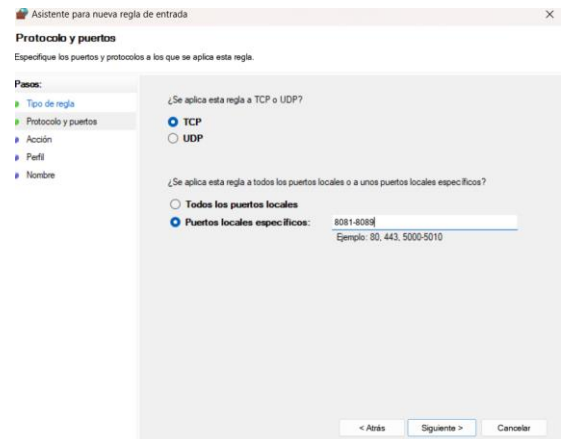


Figura 41: Se quiere dejar abierto solo el puerto 8080, por ende se coloca 8081-8089.

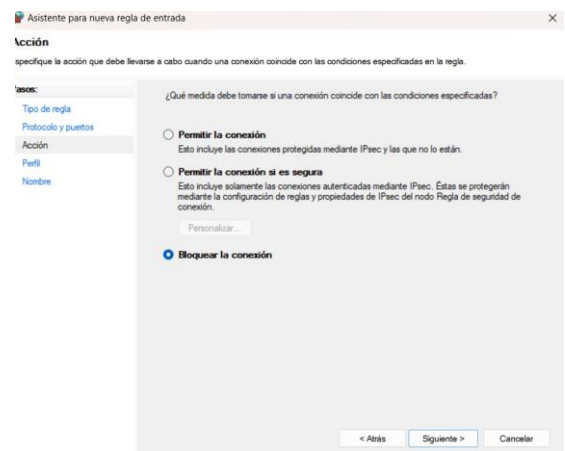


Figura 42: Para elegir la acción de la regla, para este ejercicio se elige bloquear conexión.



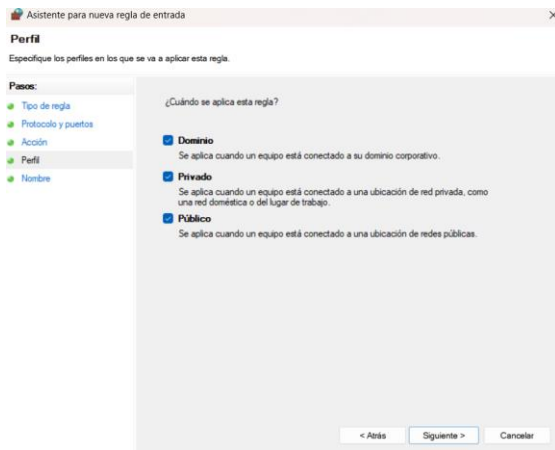


Figura 43: Para elegir la acción de la regla, para este ejercicio se elige bloquear conexión.

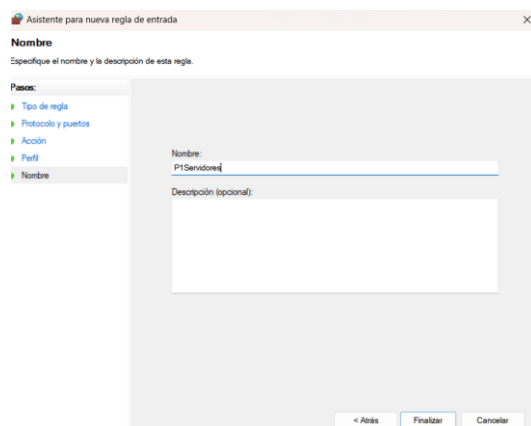


Figura 44: Se coloca el nombre de la nueva regla

- Finalmente los cambios han sido guardados en la configuración del firewall y solo queda vigilar su funcionamiento.

## F. Servidores Cloud

### • Heroku

Heroku es una plataforma en la nube que brinda un enfoque simplificado para lanzar y administrar proyectos de aplicaciones sin la necesidad de preocuparse por la gestión de servidores e infraestructura. Dirigido especialmente al aprendizaje y desarrollo de estudiantes y desarrolladores, Heroku ofrece un entorno de tiempo de ejecución totalmente gestionado junto con una amplia gama de herramientas y servicios integrados. Esto permite a los usuarios concentrarse en escribir un código de calidad y construir aplicaciones atractivas, mientras Heroku se encarga de la complejidad de la gestión de operaciones de desarrollo. La flexibilidad de Heroku permite a los desarrolladores construir aplicaciones utilizando el lenguaje o framework que mejor conocen o prefieren.

Además, ofrece una variedad de planes de bajo costo y servicios gratuitos para que los usuarios experimenten, aprendan y prototipen nuevas ideas. Con una amplia documentación, soporte robusto y una comunidad activa, Heroku ofrece recursos valiosos para resolver problemas rápidamente y facilitar el crecimiento y la escalabilidad de las aplicaciones. En conclusión, Heroku es una plataforma accesible y poderosa que ayuda a los desarrolladores a centrarse en mejorar sus habilidades y llevar sus proyectos de aplicaciones al siguiente nivel sin preocuparse por la infraestructura subyacente.

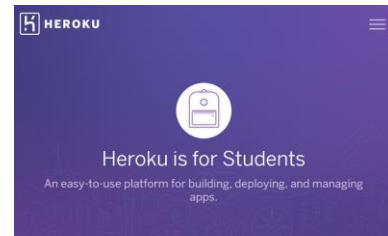


Figura 45: Entorno Heroku

### • Microsoft Azure

Microsoft Azure es una plataforma de computación en la nube que ofrece una amplia gama de servicios y funcionalidades para satisfacer las necesidades de diversas empresas. Entre sus ventajas destacan su escalabilidad y flexibilidad, permitiendo ajustar los recursos según la demanda del negocio y pagando solo por lo que se utiliza. Además, Azure ofrece un entorno integrado que se integra perfectamente con las herramientas de desarrollo de Microsoft, lo que facilita la implementación y gestión de aplicaciones. En términos de seguridad, Azure cuenta con robustas medidas de seguridad y cumplimiento normativo, lo que lo convierte en una opción confiable para empresas de todos los tamaños. Sin embargo, Azure también presenta desafíos, como su complejidad y posibles preocupaciones sobre costos. La amplia gama de servicios y opciones puede resultar abrumadora para los usuarios nuevos, lo que puede generar una curva de aprendizaje pronunciada. Además, aunque Azure ofrece un modelo de pago por uso, gestionar los costos puede ser complicado sin una planificación y monitorización cuidadosas. Habiendo presentado esta información, se concluye que Azure es una plataforma de computación en la nube sólida y completa, pero las empresas deben evaluar cuidadosamente sus necesidades específicas y considerar los desafíos potenciales antes de adoptarla completamente. Comparar Azure con otros proveedores de la nube puede ayudar a tomar una decisión informada y aprovechar al máximo el potencial de la computación en la nube para impulsar el éxito empresarial.

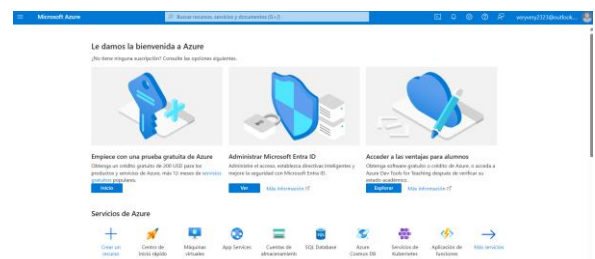


Figura 46: Entorno Microsoft Azure

### • Google Cloud Platform

Google Cloud Platform es una plataforma de computación en la nube que ofrece una amplia gama de servicios para empresas de todos los tamaños. Entre sus ventajas destacan su infraestructura global y escalable, que permite a las empresas expandirse sin preocuparse por la capacidad de sus servidores. Además, GCP ofrece una amplia variedad de servicios, desde computación y almacenamiento hasta inteligencia artificial y análisis de datos, lo que permite a las empresas desarrollar y desplegar aplicaciones de manera eficiente y escalable. Sin bien las ventajas se ven entusiasmadas, GCP también presenta desafíos, como su curva de aprendizaje y su interfaz a veces compleja para nuevos usuarios. Además, el soporte al cliente puede ser inconsistente en comparación con otros proveedores de la nube. En conclusión, Google Cloud Platform ofrece una sólida infraestructura en la nube con una amplia gama de servicios, pero las empresas deben considerar cuidadosamente sus necesidades específicas y evaluar la complejidad y el soporte antes de adoptarla completamente.

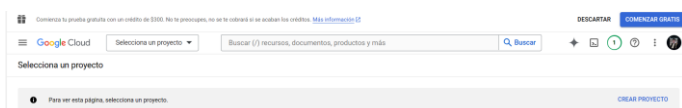


Figura 47: Entorno Google Cloud

### • Amazon Web Services

Amazon Web Services (AWS) es una plataforma líder en servicios en la nube que ofrece una amplia gama de soluciones para empresas de todos los tamaños y sectores. Su robusta infraestructura global, respaldada por certificaciones y auditorías reconocidas en la industria, proporciona seguridad y confianza a los usuarios. AWS ofrece una variedad de servicios escalables y flexibles que permiten a las empresas adaptarse rápidamente a las demandas del mercado y mejorar su agilidad operativa. Además, su modelo de precios basado en el pago por uso y su capacidad de adaptación hacen que sea una opción atractiva para empresas de cualquier presupuesto y necesidad. A pesar de sus numerosas ventajas, AWS presenta ciertos desafíos, como su complejidad para usuarios principiantes y la necesidad de una curva de aprendizaje considerable para aprovechar al máximo sus capacidades. Además, su enfoque genérico puede no ser la opción más adecuada para empresas que operan en entornos altamente regulados, como el sector farmacéutico y de la salud. Definitivamente AWS ofrece una plataforma sólida y versátil para la computación en la nube, pero las empresas deben evaluar cuidadosamente sus necesidades y considerar alternativas específicas si operan en sectores con requisitos regulatorios particulares.



Figura 48: Entorno Amazon Web Service

Los servicios en la nube ofrecidos por Heroku, Microsoft Azure, Google Cloud Platform (GCP) y Amazon Web Services (AWS) son herramientas poderosas para impulsar proyectos digitales, pero cada uno presenta sus propias características y limitaciones. Heroku se destaca por su facilidad de uso y enfoque en permitir a los desarrolladores concentrarse en la escritura de código de calidad, aunque su versión gratuita puede ser limitada en recursos para proyectos más grandes. Azure, GCP y AWS ofrecen períodos de prueba gratuitos con créditos iniciales, pero es crucial monitorear y gestionar los costos después de este período, ya que pueden aumentar rápidamente si no se toman precauciones. Además, estos proveedores ofrecen una amplia gama de servicios, desde almacenamiento hasta inteligencia artificial, lo que permite a los usuarios seleccionar la plataforma que mejor se adapte a sus necesidades específicas de tecnología y presupuesto.

La elección de un servidor en la nube con trial o transacciones limitadas es la importancia de evaluar cuidadosamente las necesidades del proyecto y considerar el equilibrio entre la funcionalidad ofrecida y los costos asociados. Si bien los períodos de prueba pueden ser una excelente manera de probar la plataforma y sus características, es esencial tener en cuenta cómo evolucionarán los requisitos del proyecto una vez que se supere esta fase inicial. Hay que tener en cuenta que comprender las restricciones de tecnología y los límites de recursos impuestos por cada proveedor para garantizar que la plataforma seleccionada pueda satisfacer las demandas del proyecto a largo plazo. Sin embargo, la elección del servidor en la nube debe basarse en una evaluación exhaustiva de las necesidades del proyecto, los recursos disponibles y la capacidad de escalar y adaptarse a medida que el proyecto crezca.

### REFERENCIAS

- [1] *Ventajas del cloud computing* | Google Cloud. (s. f.). Google Cloud. <https://cloud.google.com/learn/advantages-of-cloud-computing?hl=es>
- [2] Antony, B. (s. f.). *Amazon Web Services: Ventajas, desventajas y alternativas*. <https://www.ambit-bst.com/blog/amazon-web-services-ventajas-desventajas>
- [3] *Heroku is for Students* | Heroku. (s. f.). <https://www.heroku.com/students>
- [4] *azure advantages and disadvantages* - Bing. (s. f.). Bing. <https://www.bing.com/search?q=azure+advantages+and+disadvantages&qs=UT&pq=azure+advantages+a&sc=10-18&cvvid=09C8628F36384D54A2D868EAF656C63&FORM=QBRE&sp=1&ghc=1&lq=0>
- [5] . AyudanTec. (2022, 25 abril). *Instalación y configuración de Apache Tomcat* [Video]. YouTube. [https://www.youtube.com/watch?v=o9rQH\\_1aEYM](https://www.youtube.com/watch?v=o9rQH_1aEYM)
- [6] *Configuración avanzada de firewalls* | Citrix SD-WAN 11.4 unidad de red. (s. f.). <https://docs.netScaler.com/es-es/citrix-sd-wan/current-release/security/stateful-firewall-nat-support/advanced-firewall-settings.html>
- [7] De Souza, I. (2022, 24 junio). *¿Qué es un servidor web y para qué sirve en Internet?* Rock Content - ES. <https://rockcontent.com/es/blog/que-es-un-servidor/>
- [8] Equipo editorial de IONOS. (2023, 1 agosto). *¿Qué son los servidores de aplicaciones?* IONOS Digital Guide. <https://www.ionos.es/digitalguide/servidores/know-how/servidor-de-aplicaciones/>
- [9] B, G., & B, G. (2024, 25 enero). *¿Qué es un hosting y cómo funciona?* Tutoriales Hostinger. <https://www.hostinger.es/tutoriales/que-es-un-hosting>

- [10] *¿Qué es el despliegue continuo?* / IBM. (s. f.). <https://www.ibm.com/es-es/topics/continuous-deployment>
- [11] OpenAI. (Año). ChatGPT [Modelo de lenguaje]. <https://openai.com/chatgpt>