

# Programación Web: Trabajo #1

Max A. Samaniego  
Facultad de ingeniería  
Universidad de Cuenca  
Cuenca, Ecuador  
max.samaniego@ucuenca.edu.ec

**Abstract**—En esta práctica se trata la configuración básica de servidores locales, puertos, directorios raíz de despliegue, estructura de carpetas, etc. Además de describir algunos conceptos importantes para el desarrollo web como lo son hosting, cloud computing, entre otros.

## I. MARCO TEÓRICO

### A. Servidor web

Un servidor web (también conocido como servidor HTTP en algunas instancias) es un programa informático que procesa una aplicación server-side y puede realizar conexiones unidireccionales o bidireccionales, ya sean estas síncronas o asíncronas, con un cliente. El código recibido por el servidor es renderizado por un navegador web. Para la transferencia de datos se suele usar un protocolo, usualmente, HTTP.[Gil] Algunos ejemplos de servidores web son:

- Apache HTTP
- Internet Information Services
- NGINX
- LiteSpeed

### B. Servidor de aplicaciones

Se conoce como servidor de aplicaciones a un servidor que maneja la mayoría o la totalidad de funciones lógicas, de negociación y de acceso de datos para servir un servicio de aplicación a un cliente. Estos servidores a menudo se benefician de la centralización, que disminuye la complejidad de desarrollo.[ION] Algunos ejemplos de servidores de aplicaciones son:

- WebLogic
- JOnAS
- Wildfly
- Geronimo

### C. Hosting

El hosting es un servicio de alojamiento web que permite desplegar una aplicación en Internet. Esto implica que no se trabaja directamente con el servidor, el cual es manejado por el proveedor del servicio; únicamente se sube la aplicación y el servicio se encarga de ejecutarla y servirla en algún dominio. La forma de subir la aplicación varía de host a host, pero una forma común es mediante contenedores de Docker.[Hos]

### D. Cloud Computing

Se le llama Cloud Computing (o computación en la nube) al uso de una red de servidores remotos para almacenar y administrar datos, bases de datos, redes y software. Básicamente se trata de la disponibilidad bajo demanda de recursos de computación como servicios mediante internet.[Goo] Existen tres modelos de cloud computing:

- **Infraestructura como servicio:** Ofrece almacenamiento y servicios de computación.
- **Plataforma como servicio:** Provee un entorno de desarrollo y despliegue para crear aplicaciones en la nube.
- **Software como servicio:** Para ofrecer aplicaciones como servicios.

### E. Despliegue continuo

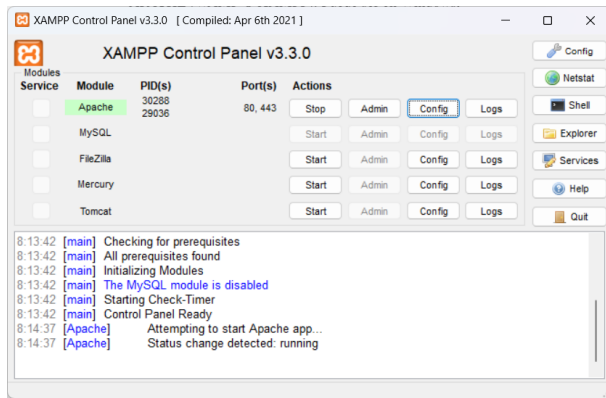
El despliegue continuo se refiere a la forma en que los equipos de desarrollo entregan funcionalidades nuevas para un software de forma frecuente mediante la automatización del proceso de despliegue. Esta automatización generalmente involucra establecer pruebas que son aplicadas automáticamente tras subir el código nuevo (por ejemplo, en GitHub). Si el código pasa todas las pruebas, entonces está listo para el despliegue que se debe realizar de forma automática. En el caso de una aplicación en la nube, este proceso puede tratarse de preparar la app para la plataforma (mediante contenedores) y/o enviar la aplicación. Algunas plataformas incluso facilitan esta automatización permitiendo obviar algunos pasos, por ejemplo, construyendo la aplicación directamente desde un repositorio.

## II. CONFIGURACIÓN DE SERVIDORES LOCALES

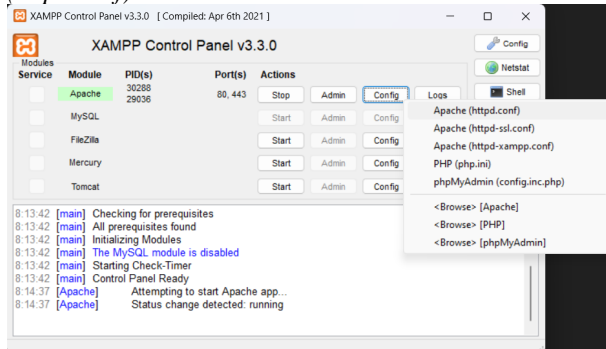
### A. Apache (XAMPP)

1) *Configuración de puerto:* Para configurar el puerto en que trabaja el servidor, se puede seguir los siguientes pasos:

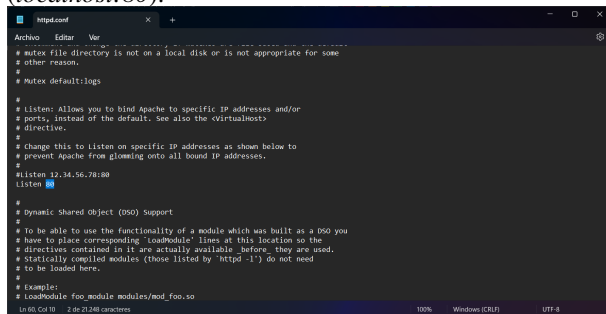
- 1) En el panel de control de XAMPP, hacer clic en el botón *Configuración* (o *Config*) de Apache.



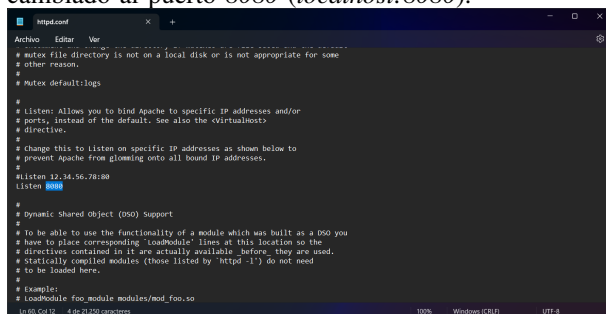
- 2) En el menú que se despliega, hacer clic en *Apache (httpd.conf)*.



- 3) Elegir esta opción abrirá el archivo de configuración en un bloc de notas. Dentro del mismo, habrá que ubicar la sección con la configuración *Listen* que por defecto tendrá el valor de 80, es decir, el puerto 80 (*localhost:80*).



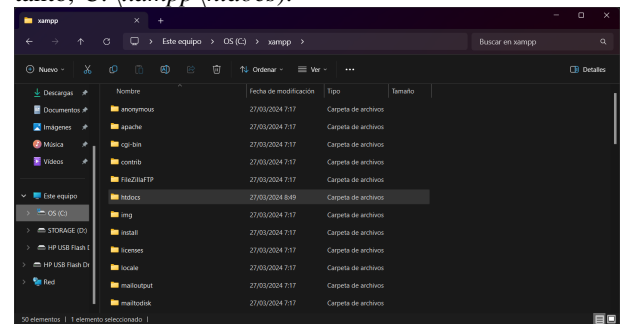
- 4) Cambiar el valor al puerto deseado, en este caso, se ha cambiado al puerto 8080 (*localhost:8080*).



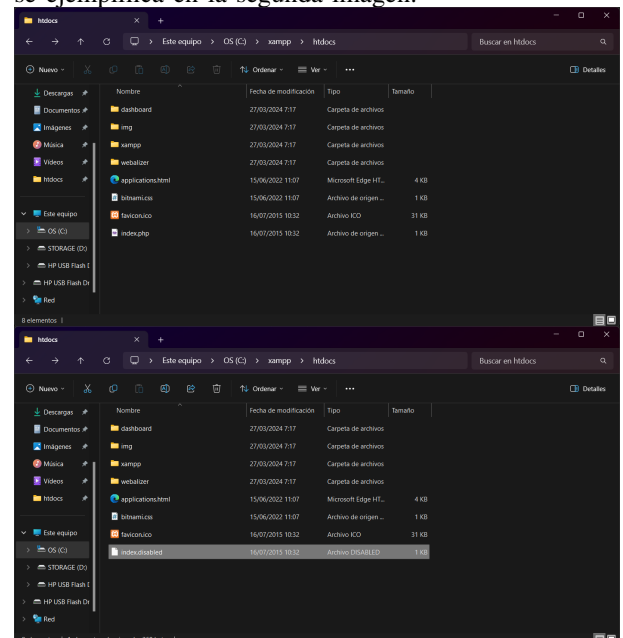
**Nota:** En caso de que el servidor esté corriendo durante este proceso, será necesario reiniciarlo para que se apliquen los cambios.

- 2) *Despliegue de un página web ¡Hola mundo!:* Para desplegar una página sencilla de *¡Hola mundo!* (en este caso, será únicamente una página estática en HTML) en el servidor de Apache, hace falta seguir los siguientes pasos:

- 1) Localizar la carpeta designada como *DocumentRoot*, por defecto, será la carpeta *htdocs* dentro del directorio de instalación de XAMPP (por defecto *C:\xampp*, por tanto, *C:\xampp\htdocs*).



- 2) Por defecto, en esta carpeta habrán varios archivos que muestran algo de información del servidor al visitar el dominio, y al entrar, redireccionan automáticamente al directorio *dashboard* (en este caso, *localhost:8080/dashboard*), para deshabilitar este comportamiento se puede eliminar/mover los archivos, o simplemente deshabilitar el archivo *index.php* como se ejemplifica en la segunda imagen.



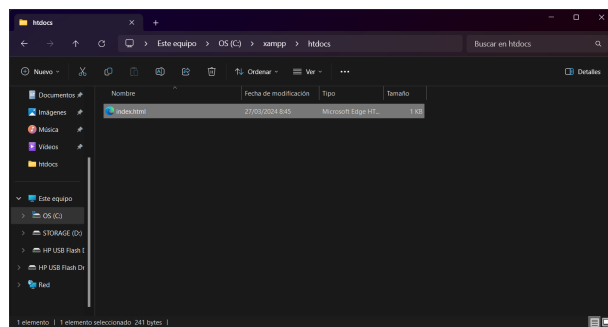
**Nota:** En este ejemplo se ha deshabilitado el archivo *index.php* cambiando la extensión a *.disabled*, pero es posible moverlo, eliminarlo, o cambiarle el nombre.

Es más recomendable mover/eliminar todos los archivos

para tener un directorio de trabajo limpio. Además, al dejar el resto de archivos, se seguirán sirviendo las páginas que se incluyen por defecto en el servidor en sus respectivos directorios (por ejemplo, aún se presentará la página de información al visitar `localhost:8080/dashboard` manualmente, simplemente estará deshabilitado el redireccionamiento automático).

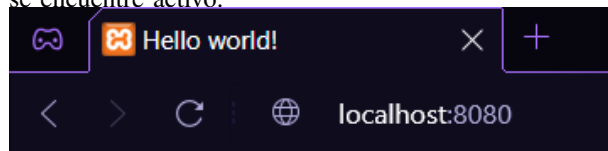
- 3) Una vez preparado el directorio, se crea un archivo HTML de ¡Hola mundo! con el nombre `index.html` y se coloca en él.

```
index.html > ...
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Hello world!</title>
7 </head>
8 <body>
9   <h1>¡Hola mundo!</h1>
10 </body>
11 </html>
```



El nombre del archivo es importante, ya que por defecto, se sirve el archivo `index`; de otra forma, habría que especificar el nombre del archivo en la URL (por ejemplo, `localhost:8080/otro_nombre.html`).

- 4) Hecho esto, se puede observar la página al visitar `localhost:8080` en el navegador siempre que el servidor se encuentre activo.



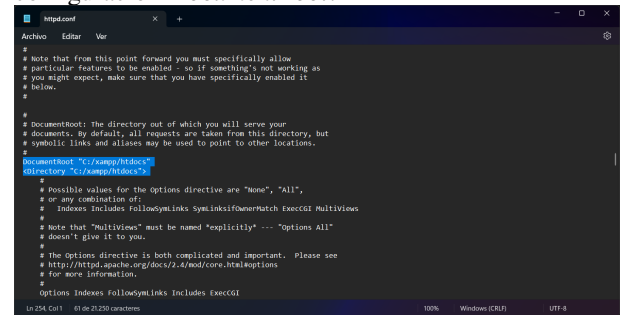
# ¡Hola mundo!

3) *Cambio de directorio de publicación:* Para cambiar el directorio de publicación del servidor (que por defecto era `C:\xampp\htdocs`), se siguen los siguientes pasos:

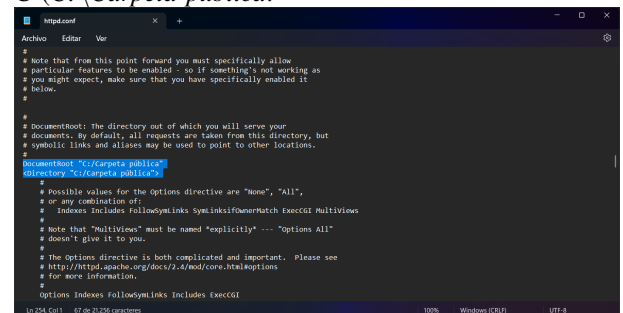
- 1) Abrir nuevamente el archivo de configuración. (Ver

pasos 1 y 2 de la subsección acerca de cómo cambiar el puerto.)

- 2) Esta vez, buscar en el archivo la sección con la configuración `DocumentRoot`.

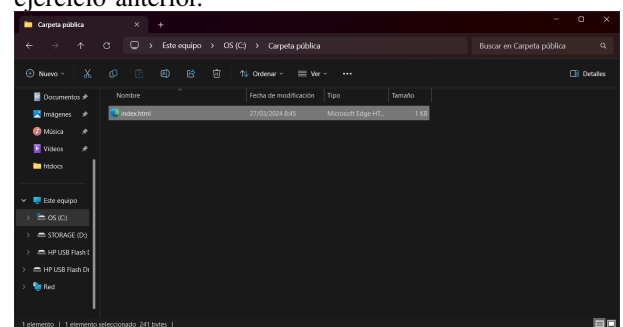


- 3) Cambiar este valor por la ruta absoluta del directorio que se desea usar para publicar la página. En este ejemplo, se usará la carpeta *Carpeta pública* ubicada en el disco C (`C:\Carpeta pública`).

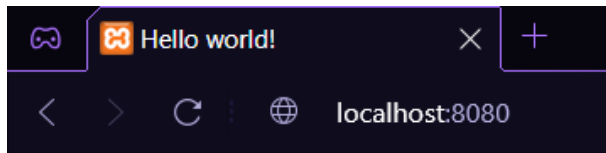


**Nota:** Si el servidor estaba activo durante el cambio, será necesario reiniciarlo.

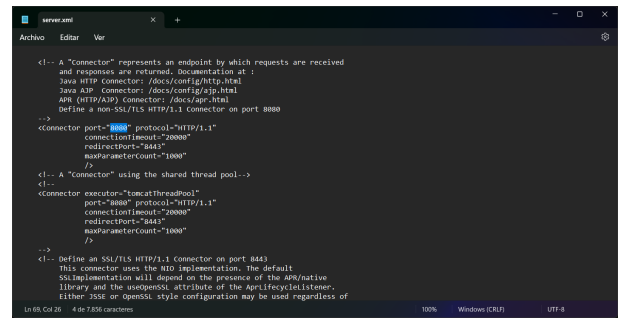
- 4) Colocar los archivos del sitio en este nuevo directorio, en este caso, el archivo `index.html` que se usó en el ejercicio anterior.



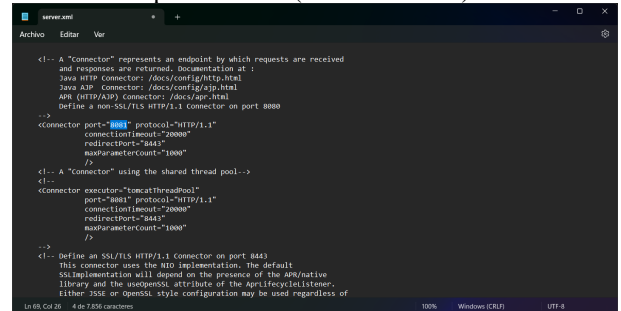
- 5) Al hacer esto, ya debería ser visible nuevamente la página al visitar `localhost:8080` en el navegador.



# ¡Hola mundo!



- 4) Cambiar el valor al puerto deseado, en este caso, se ha cambiado al puerto *8081 (localhost:8081)*.

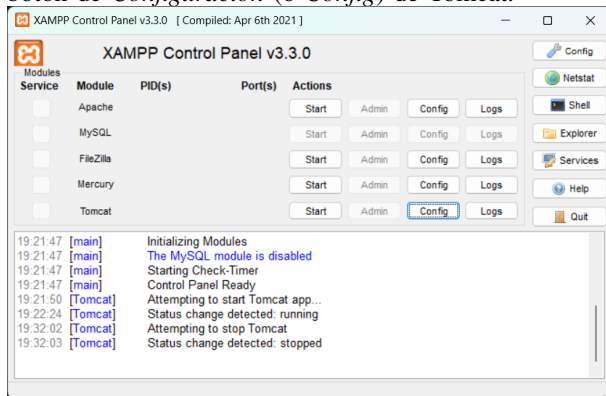


**Nota:** Al igual que con el servidor Apache, si el servidor estaba corriendo durante este proceso, será necesario reiniciar.

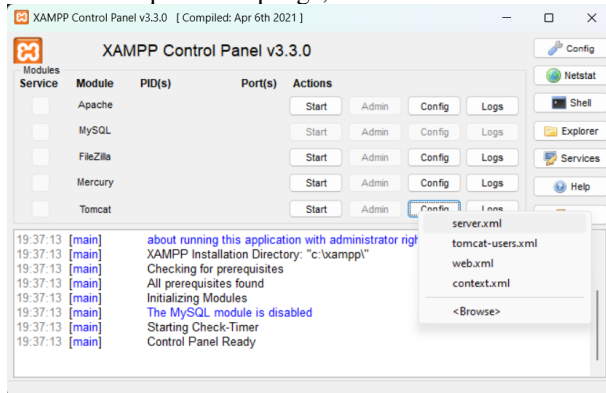
## B. Apache Tomcat

1) *Configuración de puerto:* Para cambiar el puerto en que funciona el servidor Tomcat, se siguen los pasos que se describen a continuación. Por defecto el servidor trabaja con el puerto 8080, así que se cambiará este a 8081.

- 1) En el panel de control de XAMPP, hacer clic en el botón de *Configuración* (o *Config*) de Tomcat.



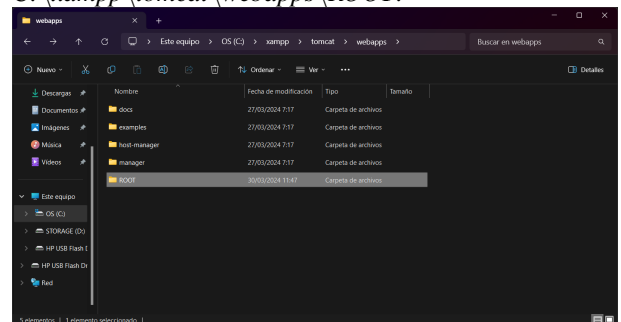
- 2) En el menú que se despliega, hacer clic en *server.xml*.



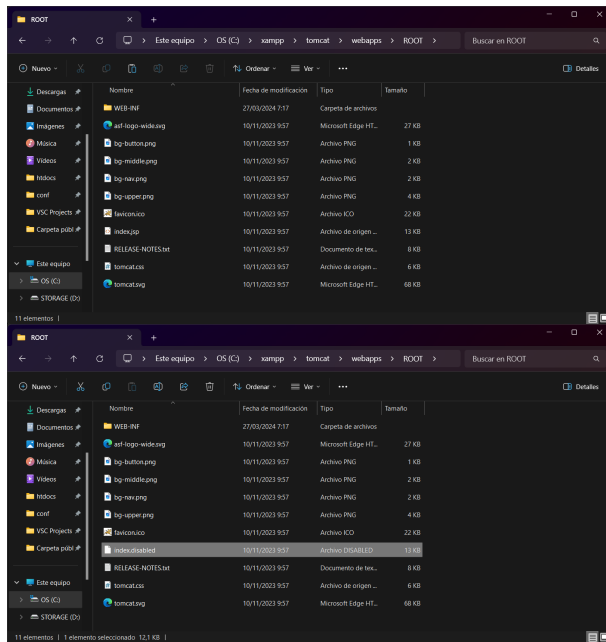
- 3) Al elegir esta opción, se abrirá un archivo XML de configuración en un bloc de notas. Dentro del mismo, hay que ubicar la sección sobre los *Connectors*.

- 2) *Desplegar una página web ¡Hola mundo!:* Desplegar una página *¡Hola mundo!* estática en Tomcat sigue un proceso muy similar al usado en Apache:

- 1) El primer paso es localizar la raíz del servidor. Por defecto esta se encuentra dentro de la carpeta de instalación de Tomcat, en la carpeta *webapps*. En el caso de la instalación con XAMPP, la ruta sería *C:\xampp\tomcat\webapps\ROOT*.

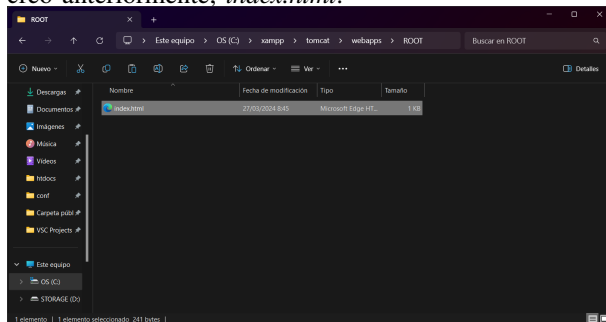


- 2) Dentro de esta carpeta, al igual que en Apache, se encontrarán varios archivos correspondientes a la página que Tomcat muestra por defecto al visitar el dominio, y, similarmente, se pueden eliminar/mover los archivos o deshabilitar el archivo *index.jsp* para mostrar una página distinta.

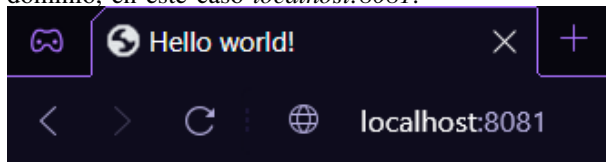


Una vez más, lo más recomendable es mover/eliminar todos los archivos para tener un directorio de trabajo limpio.

- 3) Una vez preparado el directorio de trabajo, basta con poner dentro de él el archivo de la página web que se creó anteriormente, *index.html*.



- 4) Hecho esto, la página ya debería ser visible al visitar el dominio, en este caso *localhost:8081*.



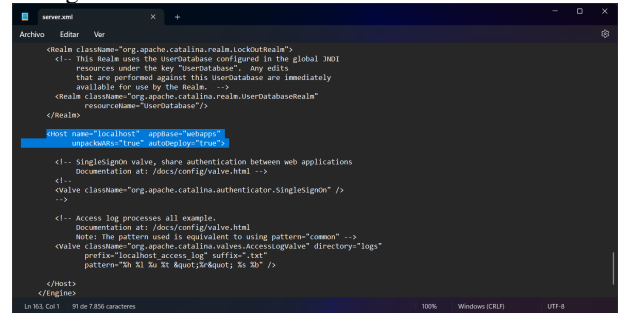
# ¡Hola mundo!

3) *Cambio de directorio de publicación:* Cambiar el directorio de publicación de Tomcat sigue un proceso similar a Apache, aunque un poco menos intuitivo:

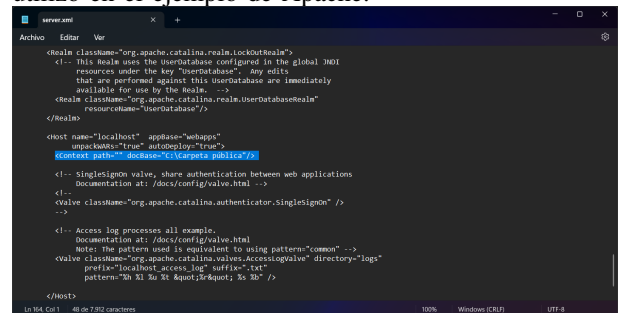
- 1) Primero, se vuelve a abrir el archivo de configuración.

(Ver pasos 1 y 2 de la subsección acerca de la configuración de puerto.)

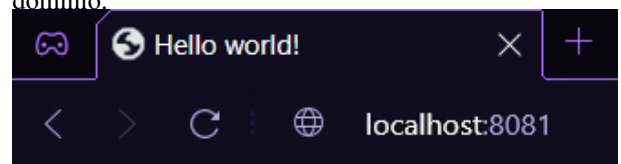
- 2) Esta vez se buscará en el archivo la sección con la configuración de *Host*.



- 3) Dentro de la etiqueta *host*, se coloca la siguiente línea: *<Context path="/" docBase="C:\Carpeta pública"/>* donde *docBase* es la ruta de la carpeta de publicación, en este caso, se está reutilizando la carpeta que se utilizó en el ejemplo de Apache.



- 4) Una vez hecho esto, se colocan los archivos en el nuevo directorio. Ya que se está reutilizando la carpeta del ejercicio de Apache, este paso ya está hecho. (Ver paso 4 de la subsección sobre cambio de directorio en Apache.)
- 5) Ahora la aplicación se sirve desde el nuevo directorio y ya debería ser visible en el navegador al visitar el dominio.



# ¡Hola mundo!

## C. Estructura de árbol de archivos

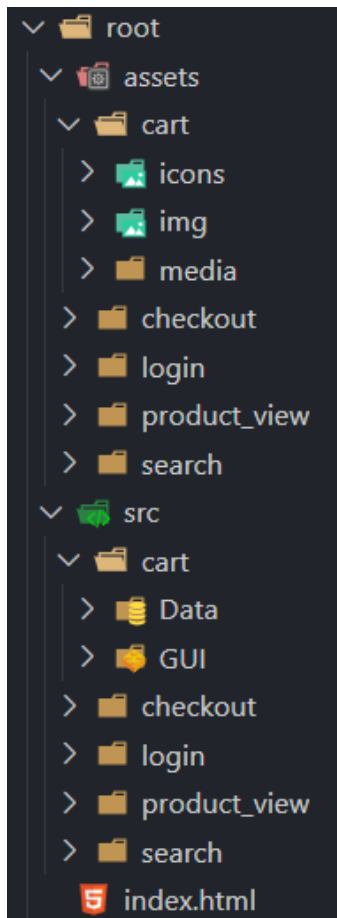
No existe un solo consenso respecto a qué estructura de archivos debería tener una aplicación web, puesto que hay muchas formas distintas de hacerlo, e incluso la estructura puede encontrarse sujeta a estándares de la empresa en algunas ocasiones. Dicho esto, hay algunas recomendaciones

generales, como lo es separar el código requerido para la interfaz gráfica del resto de la lógica del programa. Esto toma aún más importancia tratándose de desarrollo web, donde incluir el código que administra los datos en el archivo de código de la interfaz implica un riesgo de seguridad. Otra recomendación común es separar los recursos multimedia en una carpeta aparte, cuya estructura generalmente imitará la estructura de directorios del código; esta carpeta usualmente se puede encontrar en la raíz del proyecto y suele tener por nombre *assets*, *resources* o algo similar.

Un buen ejemplo de una estructura a seguir al trabajar en un proyecto web es separar el código por funcionalidad, donde cada paquete corresponde a una función del sitio, y, a su vez, cada paquete está separado también en la capa de interfaz y la capa de lógica. Así pues, si tomamos como ejemplo una tienda web, podríamos tener los siguientes paquetes:

- Login
- Vista de productos
- Búsqueda
- Carrito de compras
- Checkout/Pagos

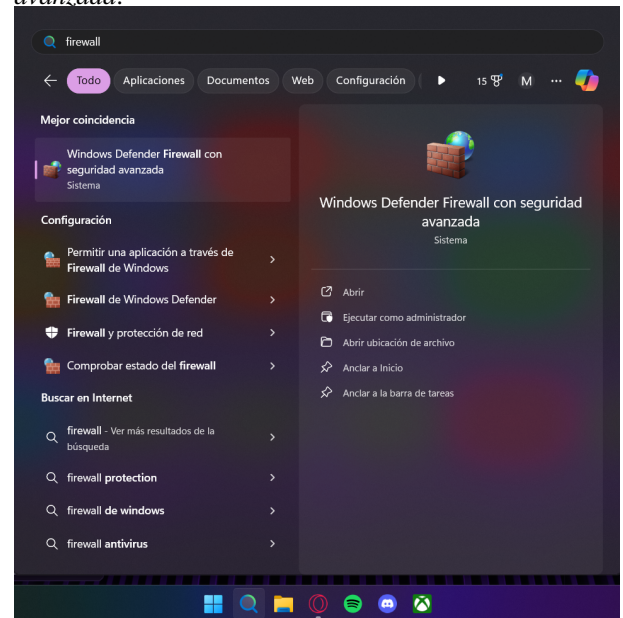
De esta forma, el árbol de directorios sería algo similar a esto:



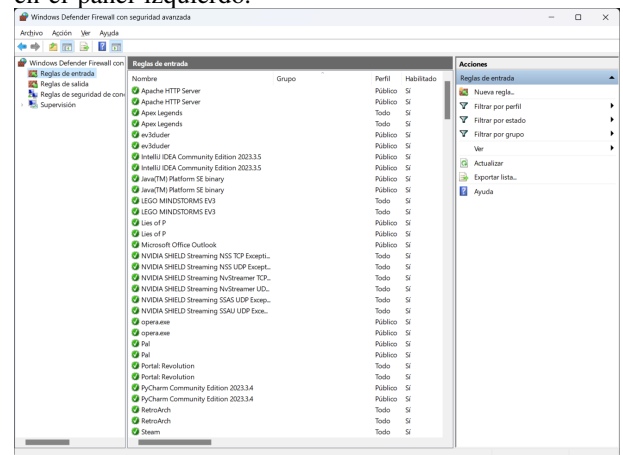
#### D. Configuración de puertos en el Firewall

Para permitir o denegar conexiones a un puerto de la computadora, hay que abrir o cerrar los puertos respectivamente. Esto se puede lograr a través de la configuración del Firewall mediante los siguientes pasos:

- 1) Buscar "firewall" en el menú inicio y hacer clic en la opción *Windows Defender Firewall con seguridad avanzada*.

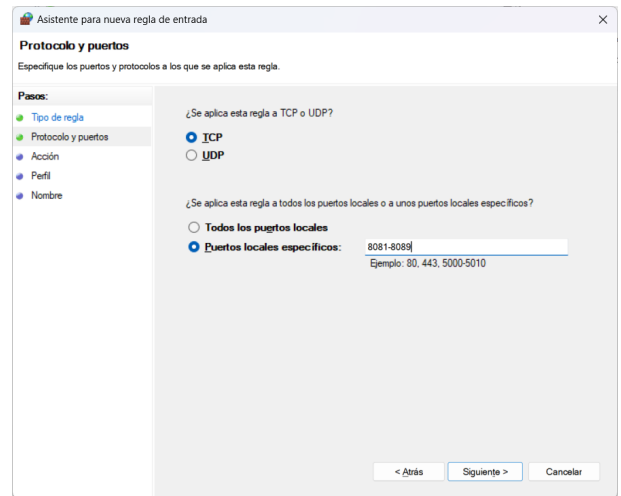
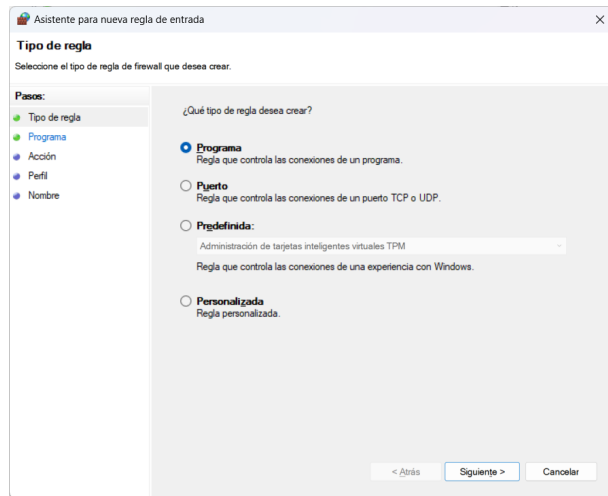


- 2) Esto abrirá la ventana de configuración del Firewall, donde se debe localizar el apartado *Reglas de entrada* en el panel izquierdo.

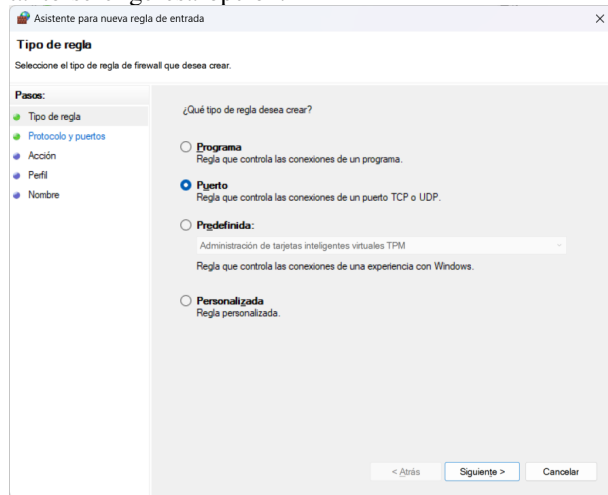


- 3) Luego, en el panel derecho, hacer clic en la opción *Nueva regla*, lo que abrirá otra ventana.

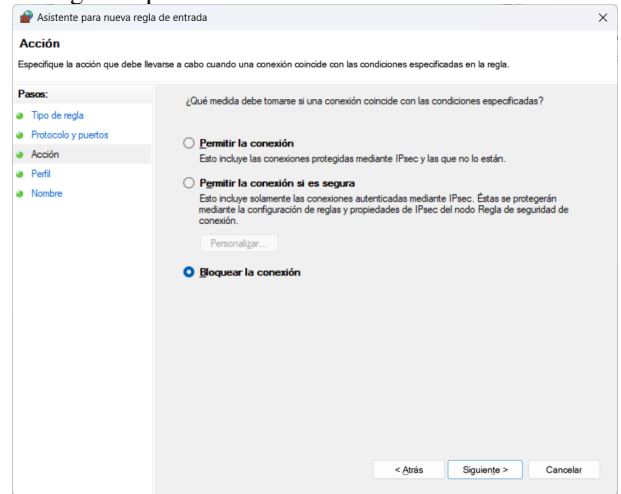




- 4) Aquí se podrá elegir el tipo de regla. En este caso se busca una regla para la conexión en los puertos, por tanto se elige esa opción.

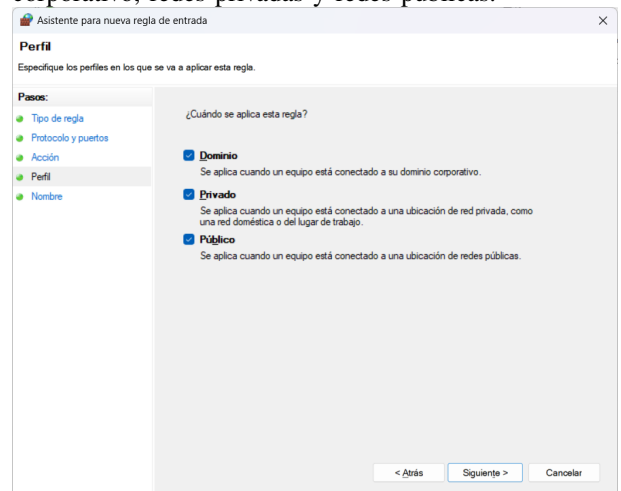


- 6) Después, habrá que elegir la acción de la regla. Esta podría ser permitir la conexión siempre, permitirla solo si es segura, o bloquear la conexión. Para este ejercicio, se elige bloquear la conexión.



- 5) Tras hacer clic en *Siguiente*, la ventana presentará las opciones de las conexiones y puertos con los que va a trabajar esta regla, que puede referirse a TCP o UDP, y trabajará con los puertos o el rango de puertos que se definan en el apartado. En este caso, se supondrá que solo se quiere dejar abierto el puerto 8080, y bloquear las conexiones de los puertos 8081 a 8089, por lo tanto, se especifica el rango de puertos 8081-8089.

- 7) Luego, se escoge el perfil de la regla, es decir, en qué redes está activa. Éstas pueden ser un dominio corporativo, redes privadas y redes públicas.



- 8) Finalmente, se asigna un nombre a la regla para poder reconocerla después en caso de requerir modificarla o eliminarla.

Después de realizar este proceso, la computadora rechazará las conexiones en los puertos 8081 al 8089. Si lo que se quisiera fuera permitir la conexión en un puerto o serie de puertos específicos, se seguiría exactamente el mismo proceso, simplemente eligiendo *Permitir la conexión* en lugar de *Bloquear* en el paso 6.

### E. Servidores Cloud

Existe una gran variedad de servidores cloud disponibles hoy en día que ofrecen una gran variedad de servicios para desplegar una aplicación, incluido el escalamiento automático de la misma cuando sea necesario para manejar la cantidad de tráfico que reciba.

Aunque todos estos servicios son pagados, el modelo que ofrecen es distinto. Mientras que algunos cobran una tarifa fija por una cantidad fija de recursos, otros cobran basados en el uso de la aplicación desplegada. *Google*, por ejemplo, da cierta cantidad de tiempo-CPU gratuito, y una vez sobrepasado, empezará a cobrar en función del mismo y otros recursos utilizados. En contraste, *Render*, tiene un límite de recursos que ofrece por un costo mensual fijo, como se muestra en la imagen[Ren]:

se muestra en la imagen[con].

From

\$0 USD

per month

✓ Web services with HTTP/2 and full TLS

✓ Private services

✓ Background workers

✓ Node, Python, Go, Rust, Ruby, and Elixir

✓ Custom Docker containers

✓ SSDs for \$0.25/GB per month

| Web Services  | Private Services | Background Workers |     |
|---------------|------------------|--------------------|-----|
| INSTANCE TYPE | PRICING          | RAM                | CPU |
| Free          | \$0/month        | 512 MB             | 0.1 |
| Starter       | \$7/month        | 512 MB             | 0.5 |
| Standard      | \$25/month       | 2 GB               | 1   |
| Pro           | \$65/month       | 4 GB               | 2   |
| Pro Plus      | \$175/month      | 8 GB               | 4   |
| Pro Max       | \$225/month      | 16 GB              | 4   |
| Pro Ultra     | \$450/month      | 32 GB              | 8   |

A menudo estos sitios cuentan también con otras limitaciones como por ejemplo: número de conexiones, almacenamiento de logs, almacenamiento de builds anteriores (para rollbacks), disponibilidad del servicio, ancho de banda, etc.

A pesar de que las limitaciones de las ofertas gratuitas de estos sitios frecuentemente sean bastante serias, hay que considerar todas las facilidades que ofrecen. Además tratándose de una aplicación sencilla, la categoría gratuita puede ser suficiente para probarla, y probar el servicio para decidir si es la mejor opción o se debería considerar otros proveedores.

### REFERENCES

- [Gil] Alexander S. Gillis. *¿Qué es Servidor web?* URL: <https://www.computerweekly.com/es/definicion/Servidor-web>.
- [Goo] Google. *¿Qué es Cloud Computing?* URL: <https://cloud.google.com/learn/what-is-cloud-computing?hl=es>.
- [Hos] Hostinger. *¿Qué es un hosting y cómo funciona? Guía para principiantes.* URL: <https://www.hostinger.es/tutoriales/que-es-un-hosting>.
- [ION] IONOS. *¿Qué es un servidor de aplicaciones?* URL: <https://www.ionos.es/digitalguide/servidores/know-how/servidor-de-aplicaciones/>.
- [Ren] Render. *Pricing — Render.* URL: <https://render.com/pricing>.