

# Trabajo 1. Configuración de Servidores

Emily Romero

*emily.romero@ucuenca.edu.ec*

**Docente** Ing. Priscila Cedillo

## PROGRAMACIÓN WEB



UNIVERSIDAD DE CUENCA

Facultad de Ingeniería

Computación

1 de abril de 2024

# 1. Marco Teórico

## Servidores Web

Un servidor web se define como una infraestructura orientada al servicio de los clientes, encargada de responder a las solicitudes HTTP (Hypertext Transfer Protocol), tales como los navegadores web, mediante la entrega de recursos estáticos. Estos recursos incluyen archivos como HTML, CSS, imágenes y archivos JavaScript. Su funcionalidad principal radica en la gestión de solicitudes HTTP y la provisión de recursos estáticos, sin ejecutar código en el lado del servidor. Tradicionalmente, se emplea para la distribución de páginas web estáticas y contenido multimedia. Algunos ejemplos de servidores web populares incluyen Apache HTTP Server, Microsoft Internet Information Services (IIS) y Nginx, reconocidos por su fiabilidad y robustez en la provisión de servicios web.

## Servidores de Aplicación

Los servidores de aplicación representan un entorno de ejecución de software diseñado para procesar solicitudes generadas por aplicaciones web dinámicas. Estos servidores se distinguen por su capacidad para ejecutar código del lado del servidor, utilizando lenguajes de programación como Java, Python, PHP, entre otros. Además, proporcionan soporte para frameworks y tecnologías que posibilitan el desarrollo de aplicaciones web complejas y dinámicas. Entre sus características principales se incluye la habilidad para interactuar con bases de datos, gestionar sesiones de usuario, implementar lógica de negocio, entre otras tareas asociadas al procesamiento dinámico de solicitudes web. Ejemplos notables de servidores de aplicación incluyen Apache Tomcat, utilizado principalmente para aplicaciones Java, Node.js, orientado hacia aplicaciones JavaScript, y Django, empleado para aplicaciones Python. Estas plataformas son reconocidas por su versatilidad y robustez en la ejecución eficiente de aplicaciones web dinámicas.

## Hosting

El hosting es un servicio de alojamiento de sitios web que ofrece a los usuarios la capacidad de almacenar archivos y datos de manera remota a través de servidores conectados permanentemente a internet. Estos servidores permiten el acceso en línea al contenido almacenado, incluyendo páginas web, archivos y bases de datos. Los proveedores de hosting web desempeñan un rol crucial al proveer la tecnología y los recursos necesarios para el funcionamiento efectivo y seguro de los sitios web. Esto incluye el mantenimiento operativo de los servidores, la implementación de medidas de seguridad y la garantía de una correcta transferencia de datos a los navegadores de los usuarios.

Existen diferentes tipos de hosting disponibles, como:

- El hosting compartido, donde varias cuentas comparten los recursos de un mismo servidor, siendo una opción económica pero potencialmente limitada en rendimiento y recursos.
- Los servidores virtuales privados (VPS), que ofrecen instancias virtualizadas de un servidor físico, proporcionando mayor control y flexibilidad que el hosting compartido.
- El hosting dedicado, que otorga al usuario acceso exclusivo a un servidor físico completo, brindando un alto nivel de control y rendimiento, aunque a un costo mayor, siendo apropiado para sitios web con altos niveles de tráfico y necesidades específicas.
- Por último, el Cloud Hosting que distribuye los recursos del servidor a través de una red de servidores interconectados en la nube, ofreciendo escalabilidad y redundancia, ya que los recursos pueden ajustarse según demanda.

Estas opciones de hosting varían en términos de recursos disponibles, como espacio en disco, ancho de banda y capacidad de procesamiento, ofreciendo así soluciones adaptadas a las necesidades específicas de cada proyecto web.

## Computación en la Nube

En relación con el cloud hosting, se tiene el término cloud computing que abarca una variedad de servicios y modelos de entrega a través de la nube. Además del alojamiento de sitios web y aplicaciones

(que es una forma de cloud hosting), el cloud computing también incluye servicios como el almacenamiento en la nube, la computación en la nube, la plataforma como: infraestructura como servicio(IaaS), plataforma como servicio (PaaS) y el software como servicio (SaaS).

- **Infraestructura como Servicio (IaaS):** Los proveedores ofrecen recursos básicos de cómputo, como servidores virtuales, almacenamiento y redes, mientras los usuarios mantienen el control sobre el sistema operativo, middleware, aplicaciones y datos. Por ejemplo, Amazon Web Services (AWS) proporciona instancias de servidores virtuales (EC2) y almacenamiento (S3) que los usuarios pueden configurar y gestionar según sus necesidades.
- **Plataforma como Servicio (PaaS):** Los proveedores suministran una plataforma completa de desarrollo y ejecución que permite a los desarrolladores crear, desplegar y gestionar aplicaciones sin preocuparse por la infraestructura subyacente. Por ejemplo, Google Cloud Platform (GCP) ofrece servicios como App Engine, que permite a los desarrolladores crear y alojar aplicaciones web sin preocuparse por la infraestructura de servidores.
- **Software como Servicio (SaaS):** Los usuarios acceden a aplicaciones y software alojados en la nube a través de internet, sin necesidad de instalar ni mantener el software localmente. Ejemplos incluyen servicios como Google Workspace, que proporciona herramientas de productividad como Gmail, Google Drive y Google Docs, accesibles a través de un navegador web.

## Despliegue Continuo

El despliegue continuo es una estrategia en el desarrollo de software que implica la automatización del proceso de entrega de código desde el repositorio de control de versiones hasta el entorno de producción de manera frecuente y consistente. 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 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.

## 2. Práctica

### 2.1. Configurar 3 tipos de servidores

- **APACHE(WAMP O XAMPP):** Puede ser en Windows, Linux o macOS, no importa la distribución.

1. Elegimos instalar XAMPP en Windows



Figura 1: Instalación de XAMPP en Windows.

- Después ejecutamos el instalador. Lo primero sale es una advertencia, esta no debe ser motivo para alarmarse con respecto a que se pudiera estar haciendo algo mal con la instalación, simplemente presionamos **OK** y le permitimos continuar con la instalación normalmente.

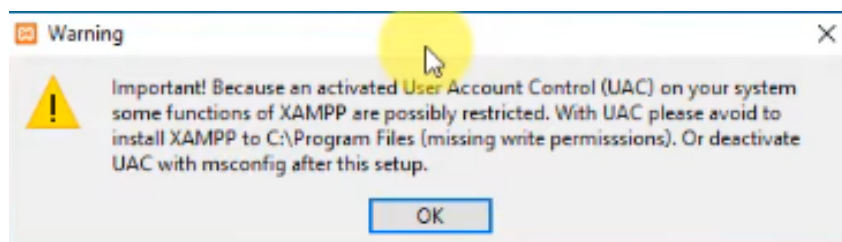


Figura 2: Advertencia al instalar.

- Por default la ventana de **Setup** va a tener todos los componentes señalados para la instalación. Sin embargo para la práctica solo es necesario instalar: **Apache** y **Tomcat**.

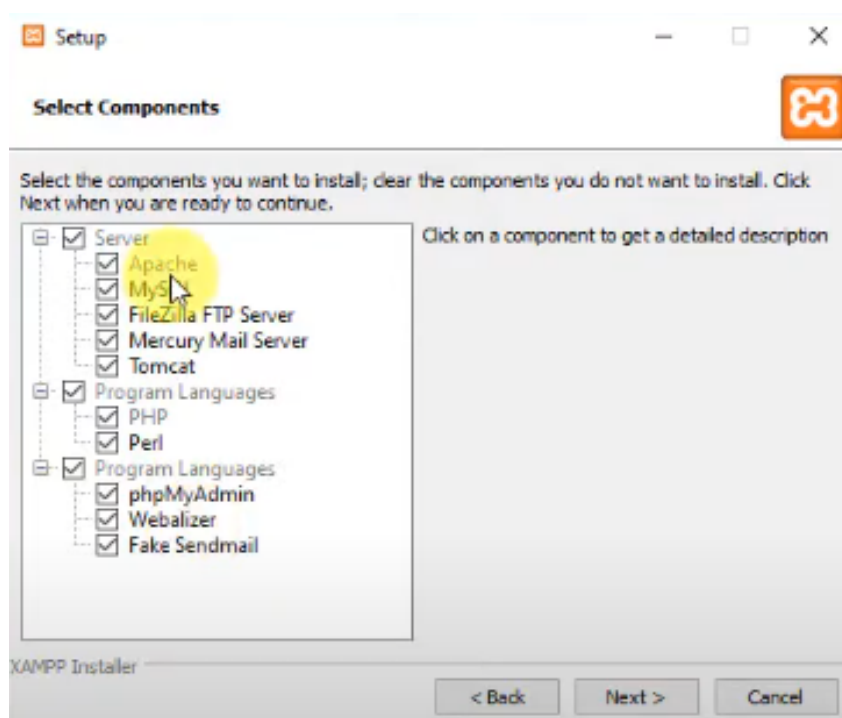


Figura 3: Lista de componentes de instalación.

- Después seleccionamos la carpeta de instalación del XAMPP, en este caso se recomienda la instalación de XAMPP en el disco C, ya que puede ayudar a evitar problemas de permisos, pues el disco C a menudo tiene menos restricciones de acceso.

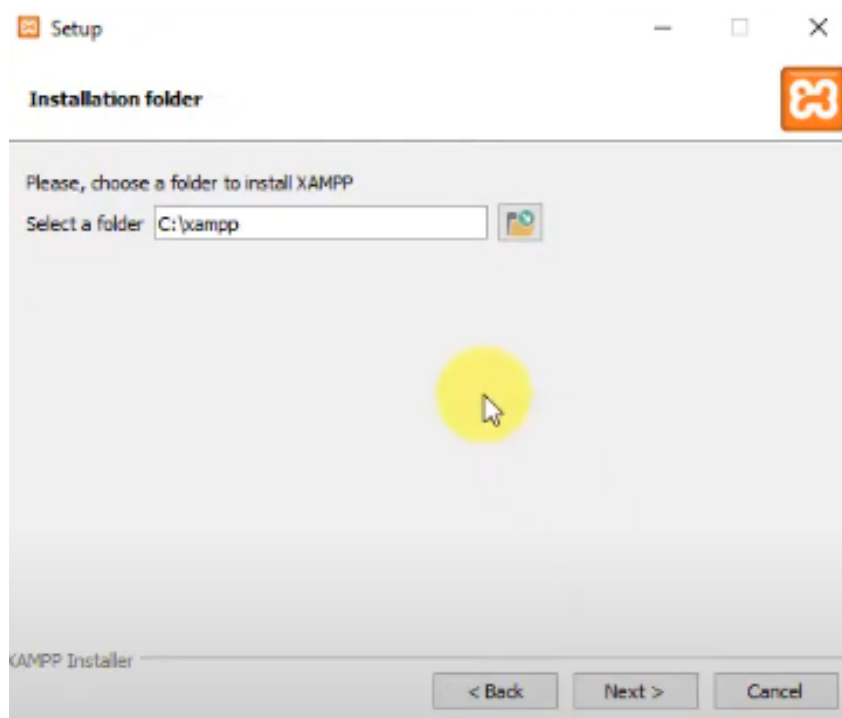
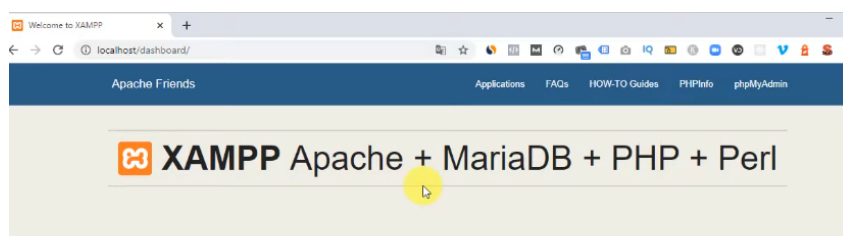


Figura 4: Ruta de instalación.

5. Por último seleccionamos el idioma por default *English* y desplegamos el servidor con la ruta `localhost/dashboard`.



### Welcome to XAMPP for Windows 8.0.3

You have successfully installed XAMPP on this system! Now you can start using Apache, MariaDB, PHP and other components. You can find more info in the FAQs section or check the HOW-TO Guides for getting started with PHP applications.

Figura 5: Despliegue de la página web de XAMPP

#### ■ **APACHE TOMCAT: Buscar un script del lado del servidor de hola mundo y desplegarlo.**

Como al instalar y configurar el XAMPP se lo hizo para el servidor de Apache y Tomcat, en este apartado veremos observaciones que se tuvo al momento de instalar y también más información sobre el servidor.

- Cuando instalamos el servidor Tomcat a través de XAMPP, este utiliza el puerto 8080 como puerto predeterminado para escuchar las solicitudes entrantes. Por lo que hay que tener cuidado que ese puerto no se esté usando porque sino causaría un conflicto.
- Los archivos que se despliegan en este servidor son los .WAR (Web Application Archive). Este es un archivo ZIP que contiene la estructura de directorios y archivos de la aplicación web, incluidos los archivos JSP, servlets, clases Java, archivos HTML, archivos CSS, JavaScript, bibliotecas de clases (JAR), etc.

- Por lo tanto cuando se desplegó la página web de Tomcat se obtuvo:

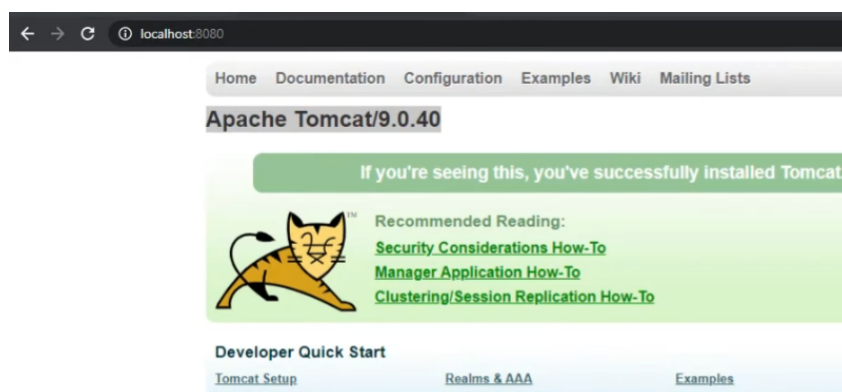


Figura 6: Aplicación web de Tomcat predeterminada en el servidor.

- Y en la carpeta de publicación se tenía archivos que no son .WAR pero que vienen de ellos ya que estos se expanden automáticamente en directorios en el sistema de archivos de Tomcat, una vez que se inicia el servidor y luego simplemente se puede eliminar los archivos de tipo .WAR .

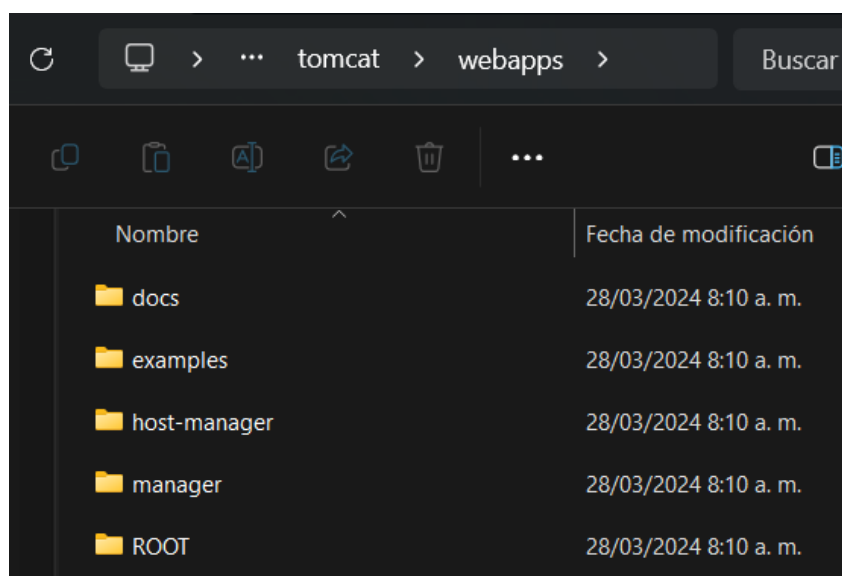


Figura 7: Archivos default en Tomcat.

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

Elegimos Amazon Web Services (AWS) para configurar el servidor por lo que se realizó los siguientes pasos:

1. Primero nos registramos en <https://aws.amazon.com/es/> para adquirir el nivel gratuito con los productos y servicios de AWS
2. Una vez que finalicemos la instalación iniciamos sesión en la consola.

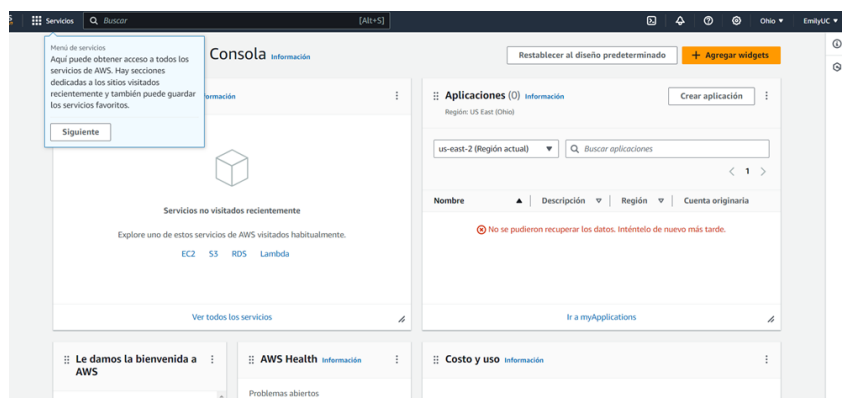


Figura 8: Página de la consola de AWS.

3. Estando en la parte de consola, buscamos el servicio **Amazon Simple Storage Service (S3)** que es comúnmente utilizado para publicar páginas web estáticas ya que está diseñado específicamente para almacenar objetos, lo que incluye archivos HTML, CSS, JavaScript, imágenes y otros recursos estáticos que componen una página web. Hay más cualidades que tiene este servidor pero se hablará de ello en la sección de **Servidor Cloud**. Una vez ahí se buscó el botón para crear un bucket.

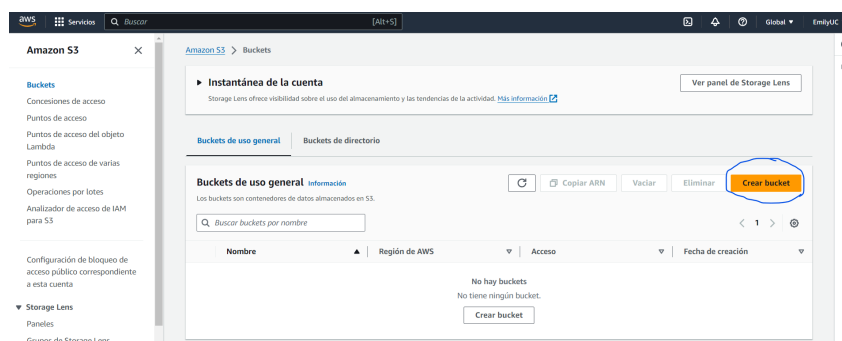


Figura 9: Servicio S3

Explicitamente un bucket es un contenedor de almacenamiento de objetos altamente escalable, seguro y flexible que se utiliza para almacenar y gestionar datos digitales en la nube de AWS.

4. Al momento de crear el bucket se eligió la región **Norteamérica (Ohio) (us-east-2)** que ofrece una buena selección de servicios y es una opción viable para diversificar recursos entre varias regiones en la costa este de Estados Unidos. También la propiedad de objeto escogida fue la **ACL deshabilitadas**. Y lo último que se modificó fue la configuración de bloqueo de acceso público para este bucket **activandola** (*las cuatro opciones*). Con ello se creó el bucket con las demás especificaciones con su opción por default.

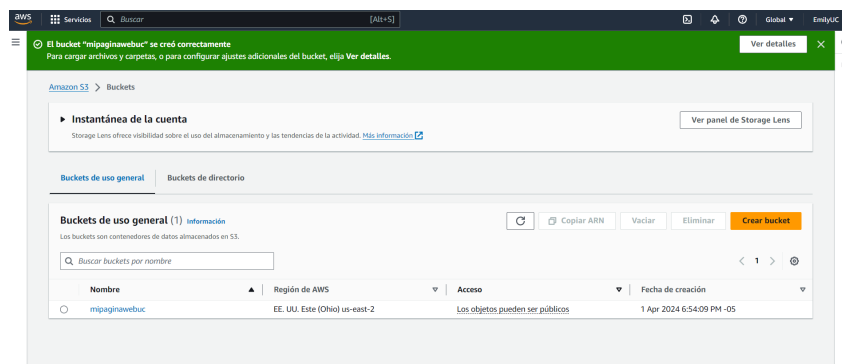


Figura 10: Creación del bucket.

Para comprobar que todo este correcto se va a despegar una página web estática.

5. Dentro del bucket nos se buscó la opción de cargar archivos en el servidor.

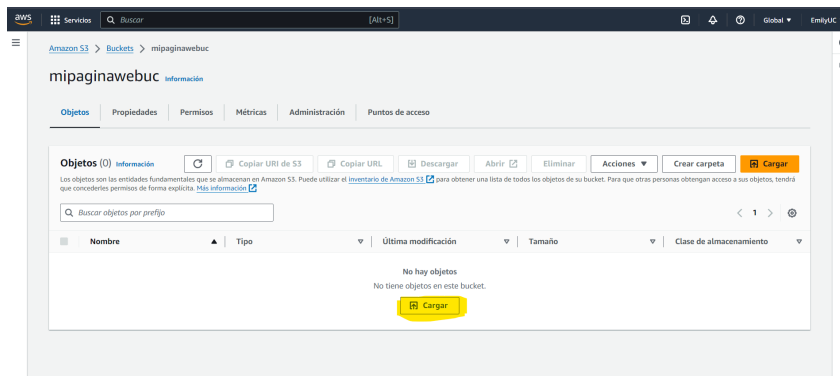


Figura 11: Página del bucket creado

6. Al dar click en Cargar, se tuvo la siguiente pantalla . En ella se eligió cargar una carpeta de un proyecto de una página estática que fue creada en vacaciones.

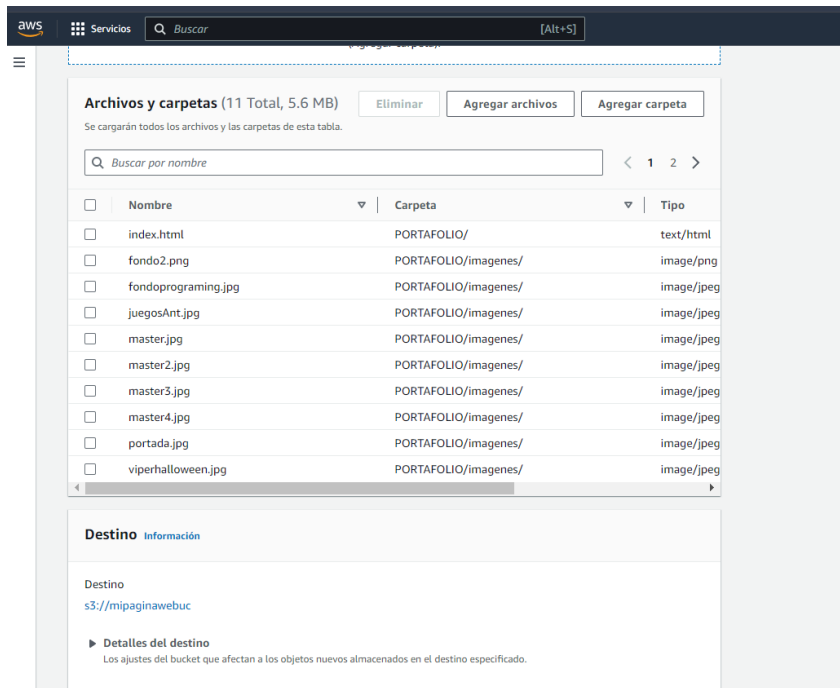


Figura 12: Página para cargar la información en el servidor.

7. Después procedemos a subir los archivos con el botón Cargar en la página para cargar información y con ello ya tendremos la carpeta con nuestra página. Luego se seleccionó el archivo index.html principal del proyecto y con ello se tuvo el URL de la página web estática en el servidor en la nube.



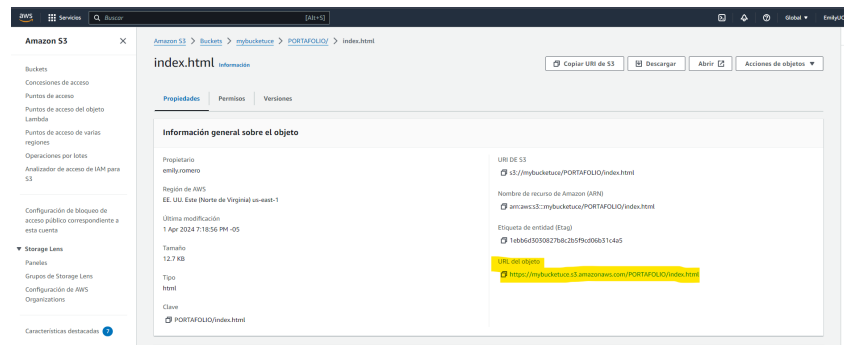


Figura 13: URL de la página web en el servidor en la nube.

### 3. Ejercicios a Realizar

#### 3.1. Servidores Locales

##### APACHE (XAMPP)

- Cambiar los puertos para que funcionen en un puerto desde el 8080 hasta el 8089 (elegir)
  1. Vamos a la configuración del panel de XAMPP y entramos en el apartado de servicios y puertos

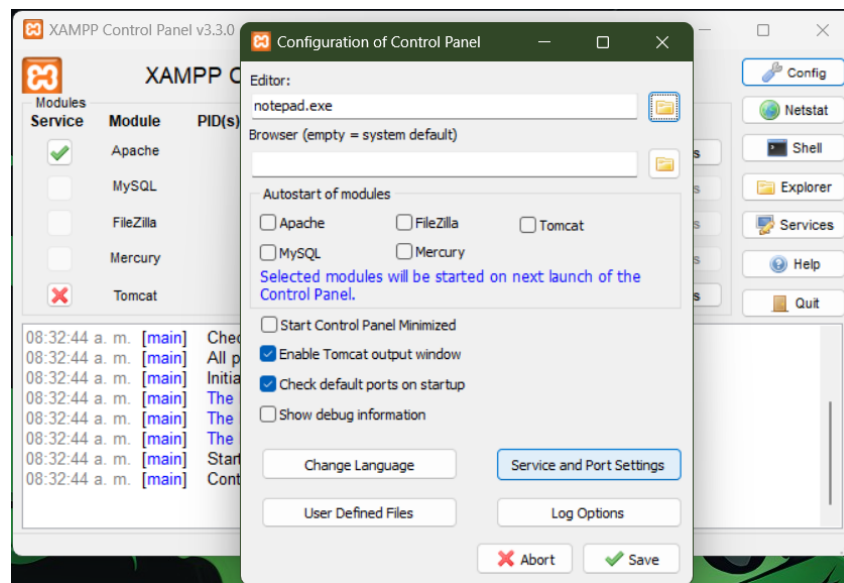


Figura 14: Configuración del panel de control del XAMPP.

2. Aquí cambiamos al puerto 8088 dentro de la configuración del XAMPP.

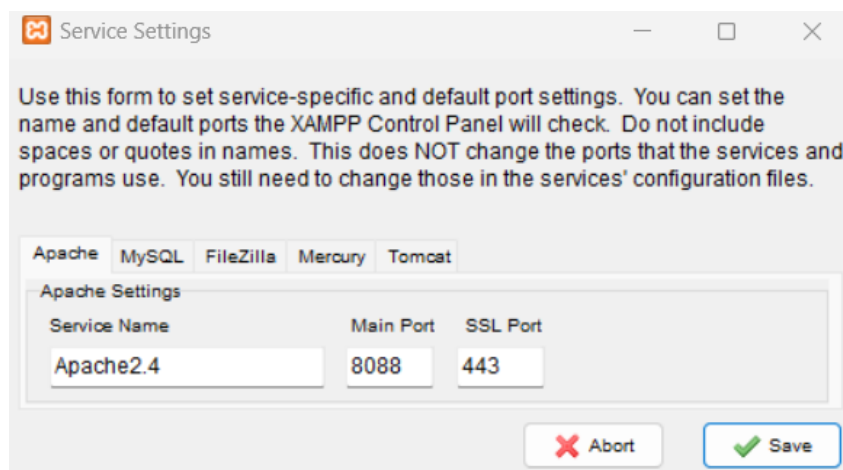


Figura 15: Cambio del puerto default en el XAMPP.

Sin embargo se conoce que los servicios van a seguir manteniendo los puertos iniciales por lo que para evitar conflictos es necesario cambiarlos también dentro de sus archivos.

3. Por ello cambiamos también el puerto en la carpeta `config`,<sup>en</sup> el archivo `"httpd.conf"`

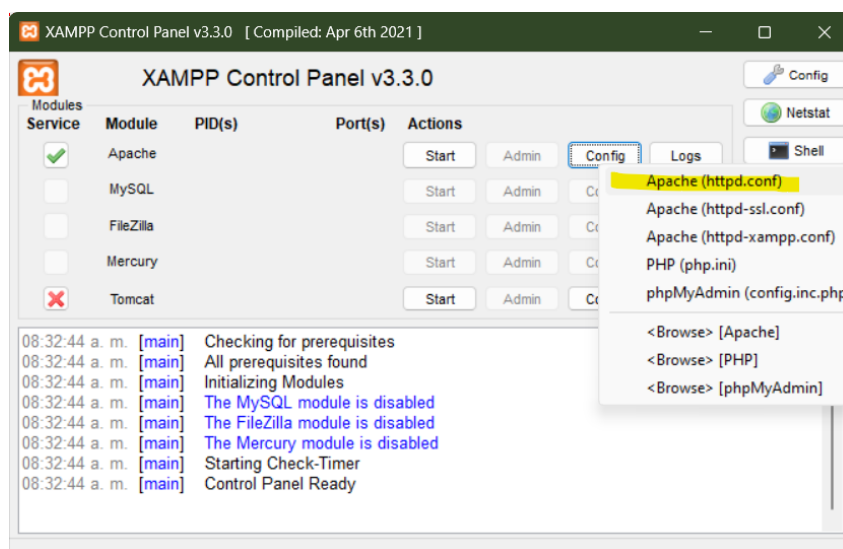


Figura 16: Encontramos el archivo `httpd.conf`.

4. Para realizar el cambio se buscó el número `"80"` dado que es el puerto default en el archivo. En este caso lo cambiamos en las dos partes que se encuentra.

```

# other reason.
#
# Mutex default:logs
#
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, instead of the default. See also the <VirtualHost>
# directive.
#
# 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 8088
#
# Dynamic Shared Object (DSO) Support
#
#
# ServerName gives the name and port that the server uses to identify itself.
# This can often be determined automatically, but we recommend you specify
# it explicitly to prevent problems during startup.
#
# If your host doesn't have a registered DNS name, enter its IP address here.
#
ServerName localhost:8088

```

Figura 17: Cambio del puerto dentro el archivo httpd.conf.

5. Por último verificamos que el cambio sea correcto iniciando el servidor.

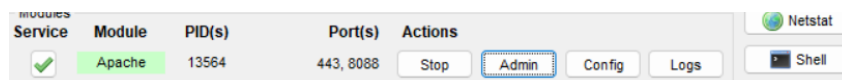


Figura 18: Servidor iniciado en el puerto 8088.

## ■ Desplegar una pagina Web sencilla de hola mundo

1. Primero vamos a la carpeta de xampp y ubicamos la carpeta **htdocs**.

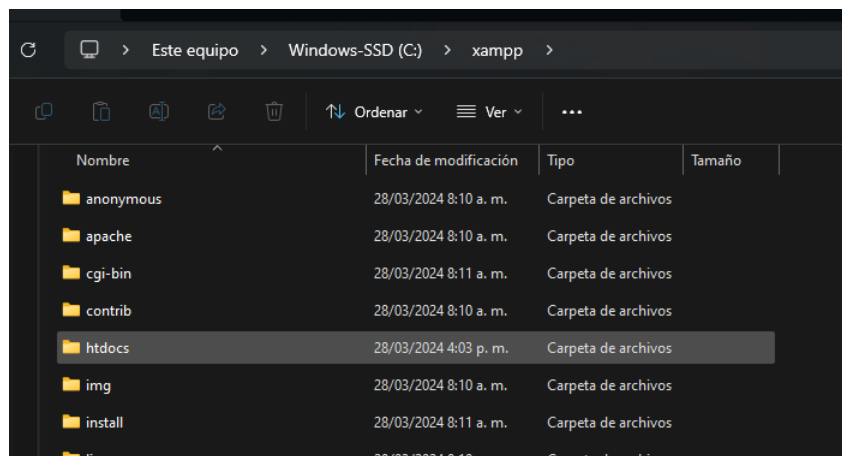


Figura 19: Ubicamos la carpeta de publicación del servidor web Apache.

2. Aquí se puso el archivo HTML que se desplegará en el servidor. La práctica recomendada suele ser organizar los archivos en subcarpetas según la estructura del sitio web para mantenerlo ordenado y fácil de gestionar. Sin embargo, para archivos HTML únicos o pequeños proyecto como en este caso, se puede colocar directamente en la carpeta de publicación.

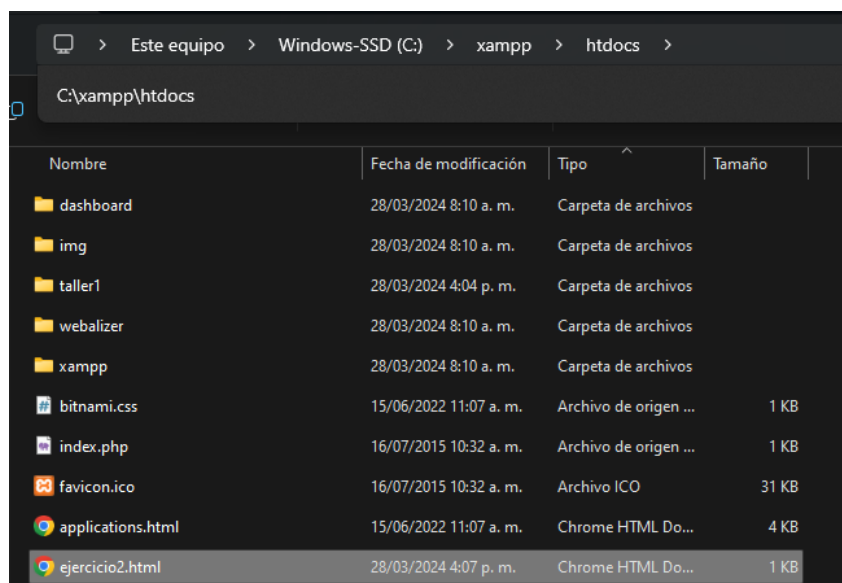


Figura 20: Guardamos el archivo HTML en la carpeta de publicación.

3. Finalmente servimos la página.

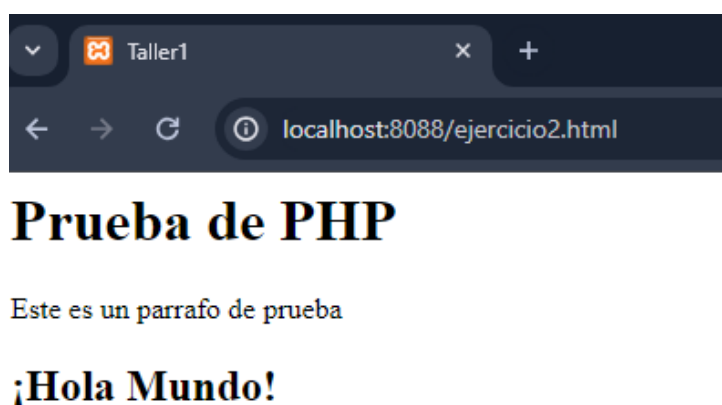


Figura 21: Servimos la página en el puerto 8088.

■ Crear una carpeta que se llame "Carpeta Pública" y hacer que sea la carpeta de publicación.

1. Se creó la carpeta esta nueva carpeta dentro de la carpeta de **xampp**. Para ejemplificar, se colocó un archivo HTML dentro de la misma.

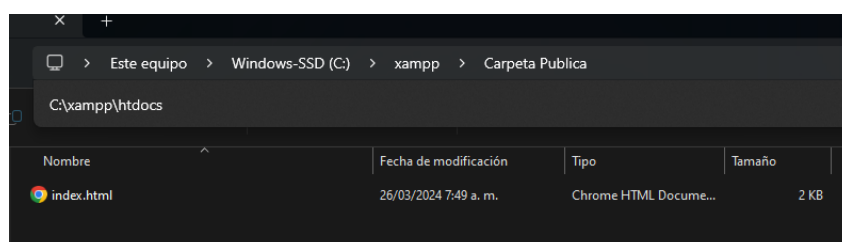


Figura 22: Creación de la nueva carpeta pública.

- En la carpeta **conf** se buscó editar el documento **httpd.conf**. (Para acceder a este archivo se hizo a través de xampp)

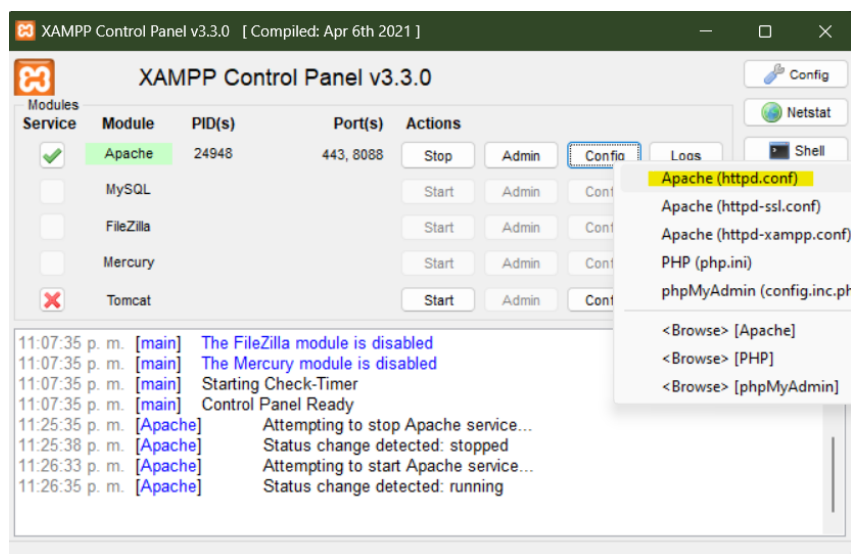


Figura 23: Ubicamos el archivo httpd.conf para el cambio de ruta.

- Dentro del archivo ubicamos la directiva **DocumentRoot** y se cambió la ruta de la carpeta de publicación. Luego de ello se reinició el servidor con el objetivo de que los cambios surjan efecto.

```
#
# 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/htdocs"
<Directory "C:/xampp/htdocs">
```

---

```
#
# 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/Carpeta Publica">
```

Figura 24: Antes y después del cambio de ruta.

- Finalmente se comprobó que la ruta ha sido cambiada exitosamente.



Figura 25: Página Web en la carpeta 'Carpeta Publica'.

- **Buscar la estructura recomendada de un árbol de archivos para publicación (En donde guarden imágenes, front- end, backend, etc.) Crear la estructura de carpetas y probar que hacer para publicar e cada una de ellas.**

1. Se comenzó creando las carpetas iniciales de la página web. En este caso se tomo el ejemplo del enunciado que se trataba de la universidad.

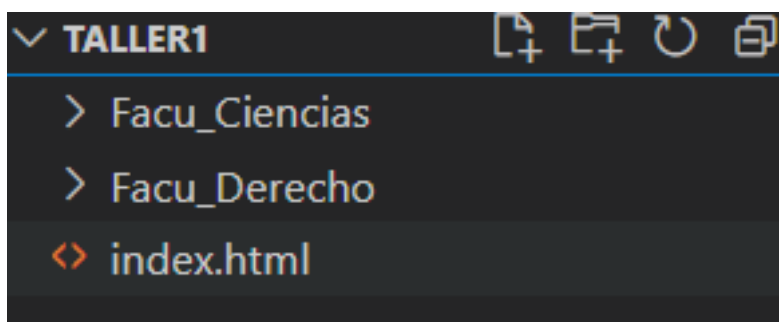


Figura 26: Creamos las carpetas iniciales

La carpeta *TALLER1* es la carpeta principal de publicación que contiene todos los archivos y carpetas del sitio web.

El archivo *index.html* dentro de la carpeta principal es la página de inicio del sitio.

2. Para cada facultad (*o cualquier otra sección que pueda existir*), se crea una subcarpeta correspondiente. Dentro de cada carpeta de facultad, se pueden tener archivos HTML adicionales, así como carpetas para imágenes (*images*), archivos CSS (*css*), archivos JavaScript (*js*), y cualquier otro recurso necesario para esa sección en particular.

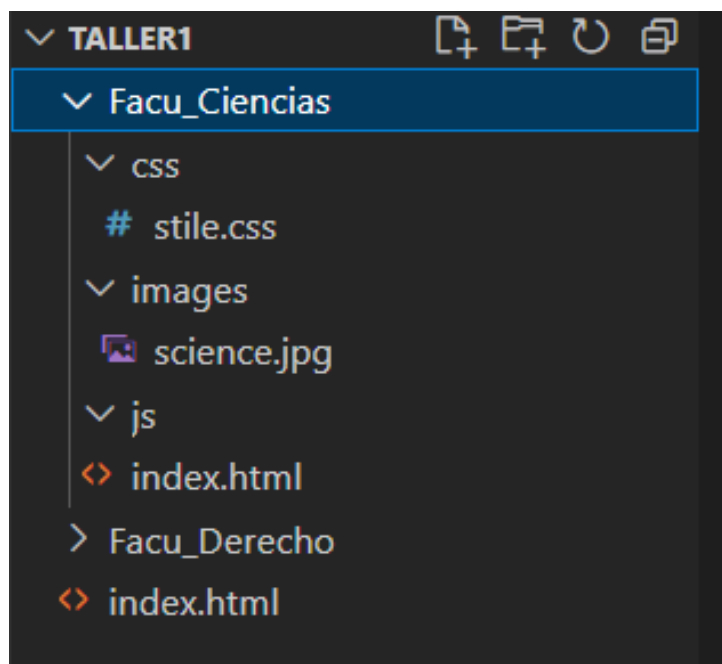


Figura 27: Ejemplo de estructura de árbol de archivos

Esta estructura permite una organización clara y jerárquica del contenido del sitio, facilitando la gestión y mantenimiento del mismo, además de mejorar la legibilidad y accesibilidad de los recursos. En el caso de la carpeta **js**, no se encuentra ningún archivo debido que ello aún no se ha visto.

3. En caso de existir secciones más grandes o complejas dentro de una facultad (*como recursos, estudiantes, académicos, admisiones, etc.*), se crea subcarpetas adicionales para organizar las páginas de manera más ordenada.

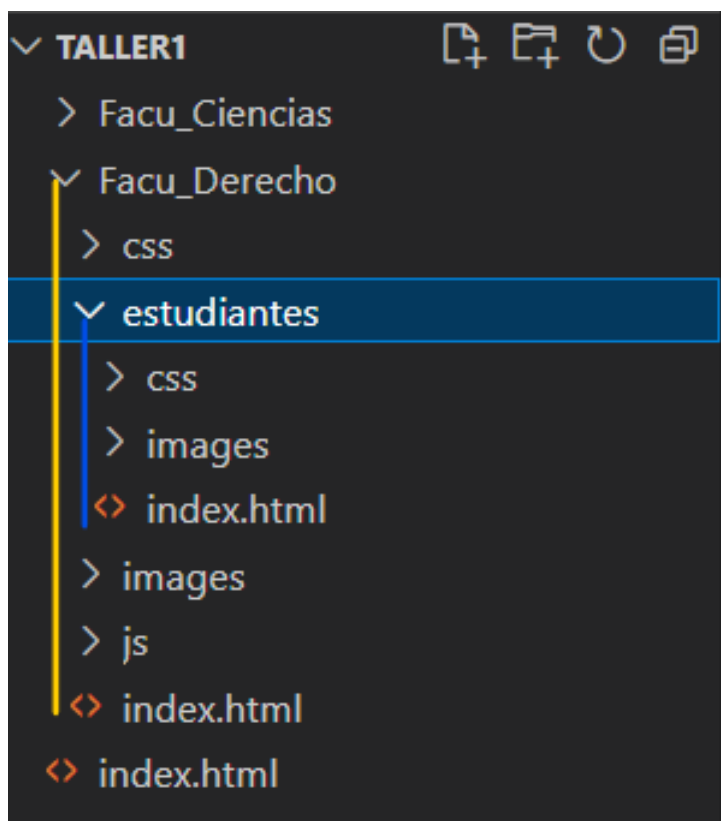


Figura 28: Ejemplo de estructura de árbol de archivos

Todo lo que se encuentra en la línea amarilla está en la carpeta *Facu Derecho* y todo lo que está en azul es lo que está en la carpeta *estudiantes*.

- **Configurar el firewall para que sirva exclusivamente en el puerto que se quiera indicar.**

En mi caso, mi computadora tiene el antivirus *Kaspersky* por lo cual para configurar el firewall tenemos que entrar en la configuración del mismo. Sin embargo si estuviera con *Windows Defender Firewall* necesitaría configurar manualmente la regla del firewall.

Esto implicaría abrir el Panel de control de Windows Defender Firewall, crear una nueva regla de entrada o salida, especificar el puerto y permitir el tráfico en ese puerto específico.

En cambio, Kaspersky detecta que Apache está escuchando en un nuevo puerto al cambiarlo y ajusta automáticamente sus reglas del firewall para permitir el tráfico en ese puerto. Esto se hace para proporcionar una protección continua y sin interrupciones mientras se mantienen los niveles de seguridad adecuados.

Sin embargo para cumplir con el enunciado podemos modificar las reglas y los permisos de la siguiente forma:

1. Abrimos Kaspersky y nos dirigimos al apartado *Más herramientas*, una vez ahí vamos a la parte de **Mi red** y seleccionamos **Monitor de red**.





Figura 29: Apartado de Red en Herramientas.

2. Nos dirigimos al apartado de puertos abiertos y buscamos el proceso de Apache HTTP Server en el puerto 8088.

Proceso	Id. del proc...	Puer...	Dirección IP local	Protocolo
AltServer	1980	49888	0.0.0.0	TCP
AltServer	1980	49889	0.0.0.0	TCP
AltServer	1980	49892	0.0.0.0	TCP
Apache HTTP Server	10396	443	0.0.0.0	TCP
Apache HTTP Server	10396	8088	0.0.0.0	TCP
Apache HTTP Server	10396	443	::	TCP
Apache HTTP Server	10396	8088	::	TCP

Figura 30: Proceso Apache HTTP Server en el puerto 8088.

3. En este caso negamos las reglas que estaban por default y agregamos la regla para el puerto 8088. Ya que no se menciona nada específico sobre la subred la colocamos como local.

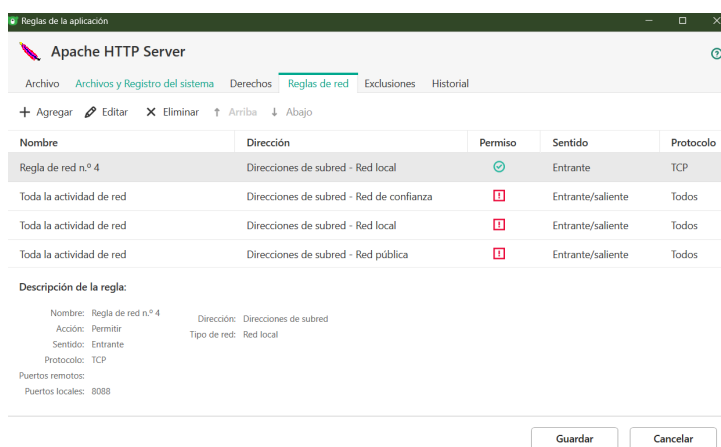


Figura 31: Agregamos la regla para el puerto 8088

## APACHE TOMCAT

- Cambiar los puertos para que funcionen en un puerto desde el 8080 hasta el 8089 (elegir)

1. Entramos en la carpeta config por medio del XAMPP en el archivo **server.xml**.

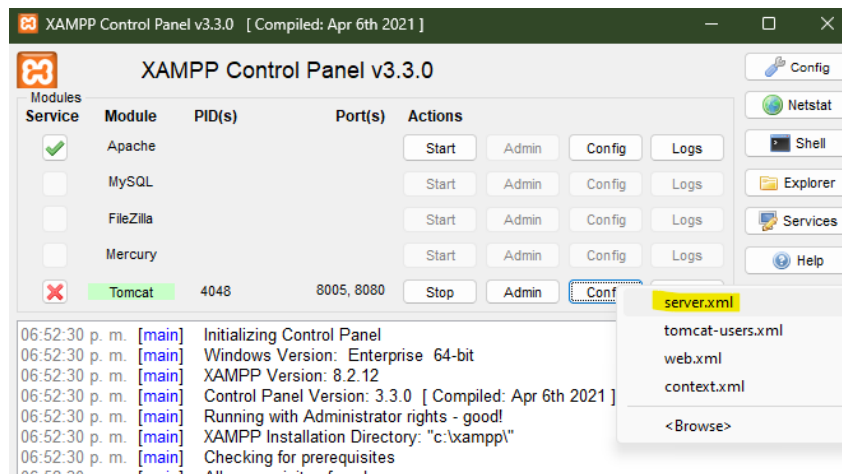


Figura 32: Ubicamos el archivo server.xml.

2. Buscamos el número 8080, que es el puerto default del servidor para cambiarlo por otro. En este caso se puso el puerto 8086.

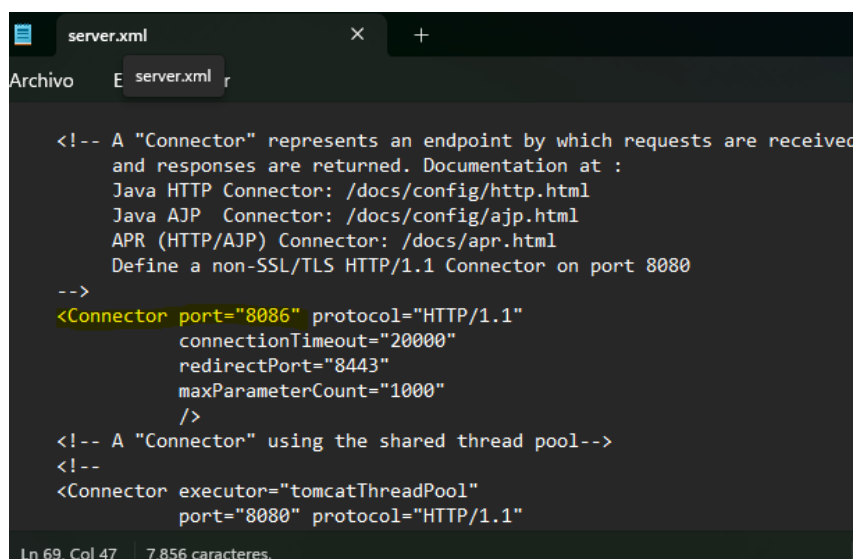


Figura 33: Cambiamos el puerto a 8086.

3. Igual que antes, también es necesario que cambiemos el puerto en la configuración del panel de control del XAMPP.

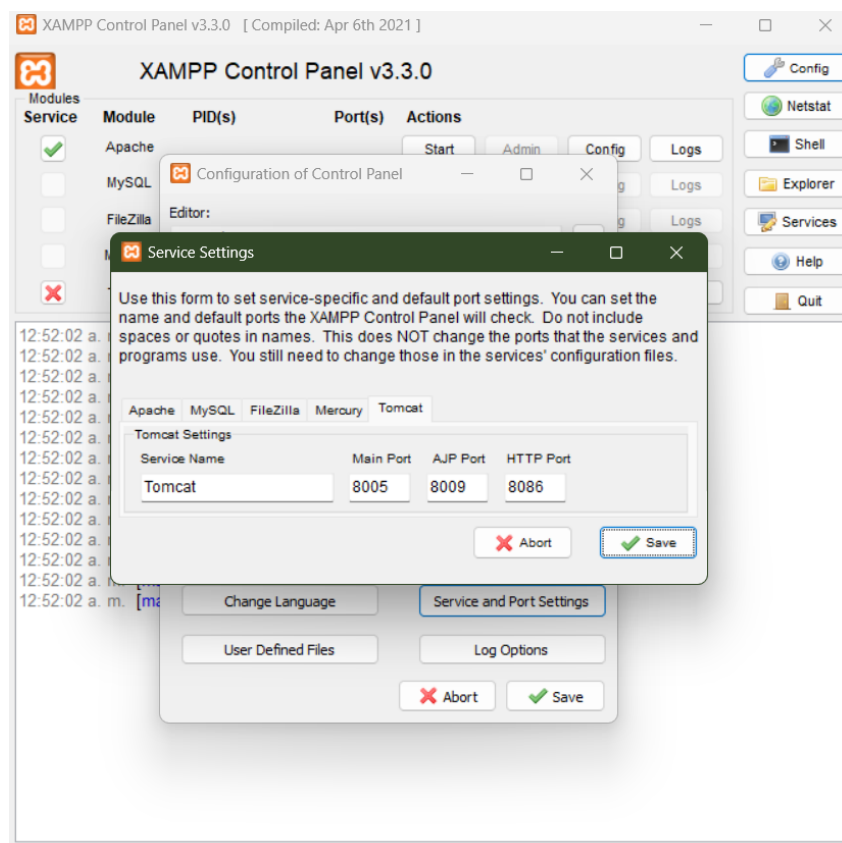


Figura 34: Cambiamos el puerto en la configuración del XAMPP.

4. Finalmente comprobamos la conexión.

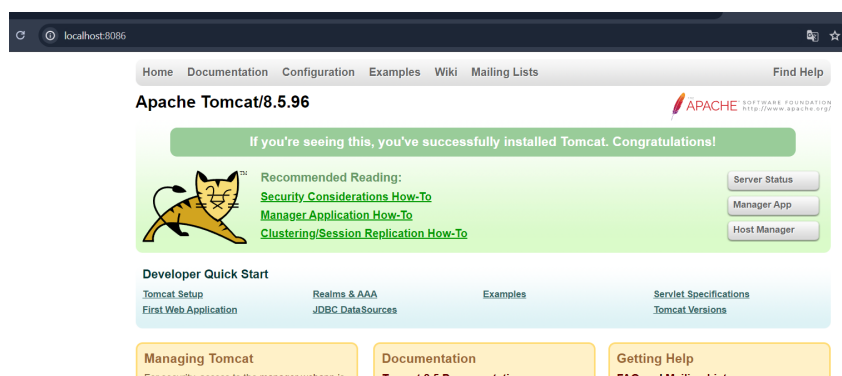


Figura 35: Página web en el puerto 8086.

## ■ Desplegar una pagina Web sencilla de hola mundo

1. Ubicamos la carpeta de publicación del servidor. Esta carpeta se llama **webapps**. Dentro de ella colocamos nuestro archivo **.WAR** para poder desplegarlo, en este caso utilizamos sample.war descargador de internet.

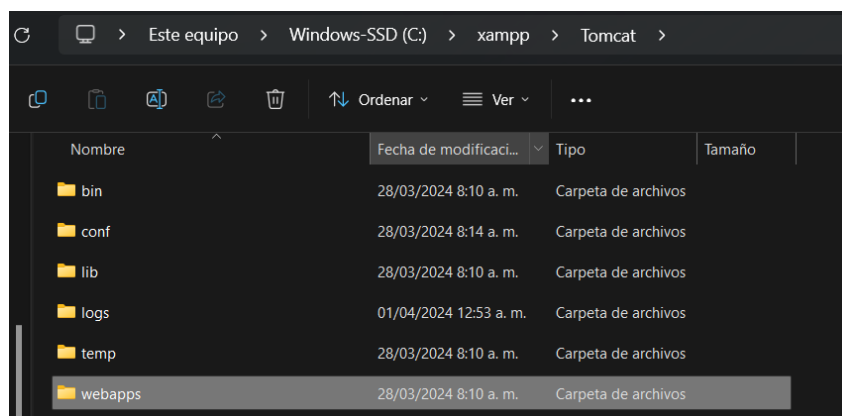


Figura 36: Ubicamos la carpeta de publicación.

- Después lo colocamos dentro de la carpeta webapps en el directorio de instalación de Tomcat y reiniciamos el servidor. De esta manera Tomcat detectará automáticamente el archivo WAR y lo desplegará como una nueva aplicación web.

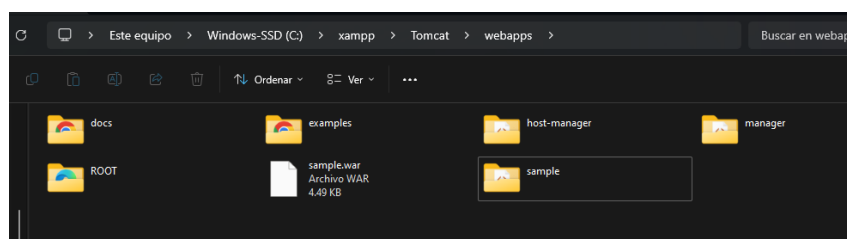


Figura 37: Nueva aplicación "sample".

- Comprobamos que no haya ningún problema despegando la aplicación obteniendo:

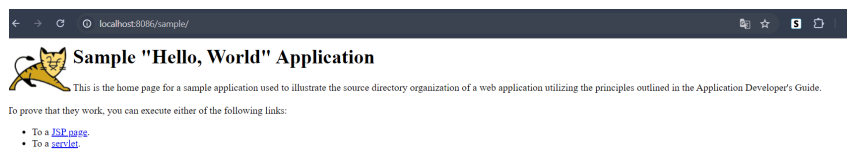


Figura 38: Aplicación desplegada.

- Crear una carpeta que se llame "Carpeta Pública" y hacer que sea la carpeta de publicación.
- Primero detenemos Tomcat antes de hacer cualquier cambio. Después por medio de XAMPP vamos al archivo **server.xml** y buscamos el elemento *Host*.

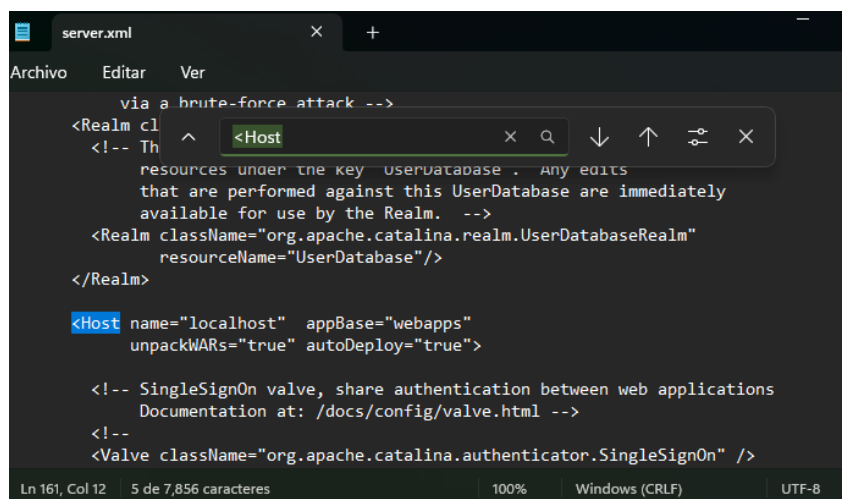


Figura 39: Ubicamos la sección de Host dentro del archivo server.xml.

2. El atributo `appBase=webapps` especifica la carpeta predeterminada donde Tomcat busca las aplicaciones web para desplegarlas. Por ello procedemos a cambiarlo por **Carpeta Publica** y guardamos.

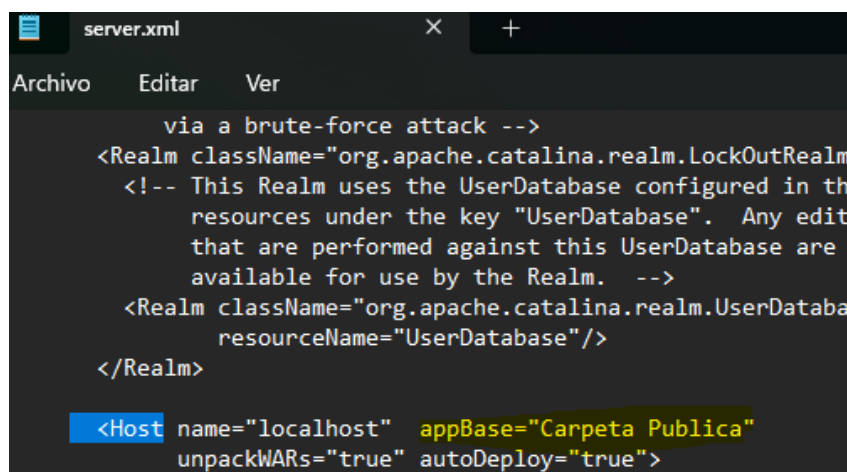


Figura 40: Cambio de la ruta de la carpeta de publicación.

3. Luego colocamos el archivo .war en la nueva carpeta de publicación e iniciamos el servidor de nuevo para poder desplegar la nueva aplicación web.

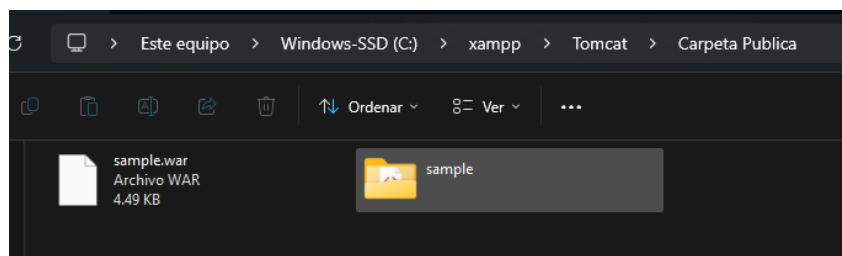


Figura 41: Nueva aplicación web desplegada.

Podemos saber que funcionó porque en el XAMPP al presionar en *Admin* la página inicial por default no es encontrada dado que la aplicación web no se encuentra en la carpeta.

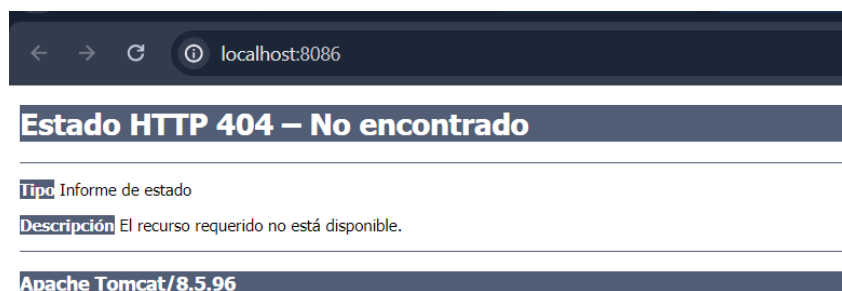


Figura 42: Aplicación web por default.

Sin embargo la aplicación *sample* se despliega sin problema.

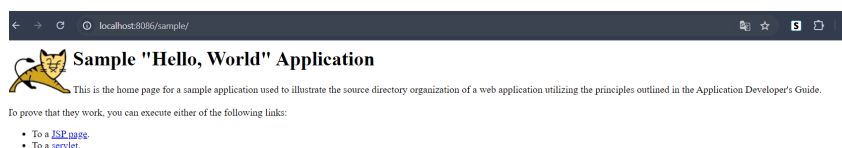


Figura 43: Aplicación desplegada en la nueva carpeta de publicación.

- **Buscar la estructura recomendada de un árbol de archivos para publicación (En donde guarden imágenes, front- end, backend, etc.) Crear la estructura de carpetas y probar que hacer para publicar e cada una de ellas.**
  1. Se buscó la estructura recomendada tomando en cuenta que son archivos .war, por lo que se tuvo el siguiente árbol de archivos.

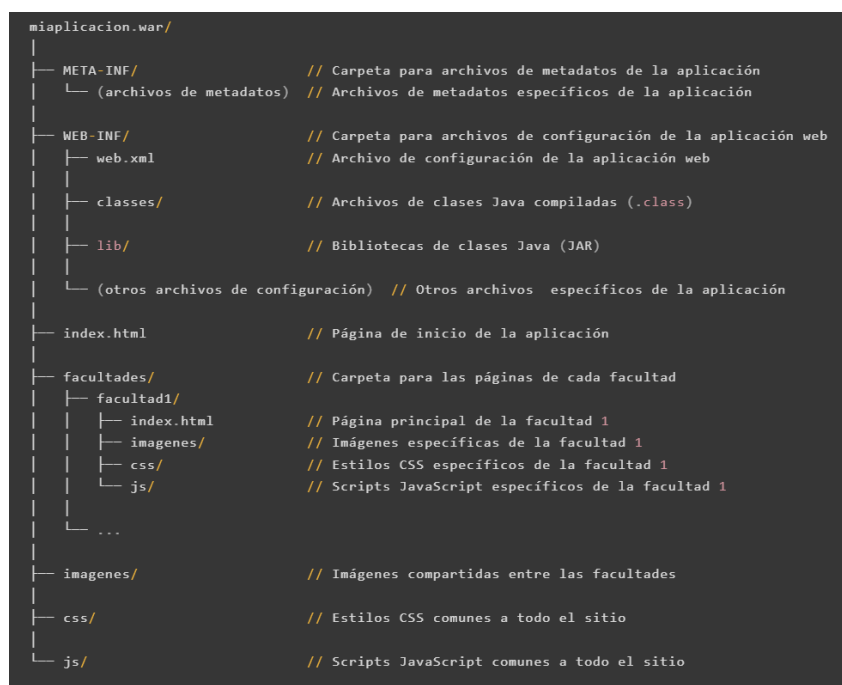


Figura 44: Árbol de archivos para publicación

2. Adaptamos la aplicación web *sample* con aquella arquitectura. La carpeta inicialmente tenía las subcarpetas: images, META-INF, WEB-INF y los archivos hello.sjp, index.thml. Por ello agremos la carpeta que usamos con el servidor de *Apache* que ya tiene la estructura mostrada

en la Figura anterior, con un archivo `index.html` como página principal y las subcarpetas de `imagenes`, `css` y `js`.

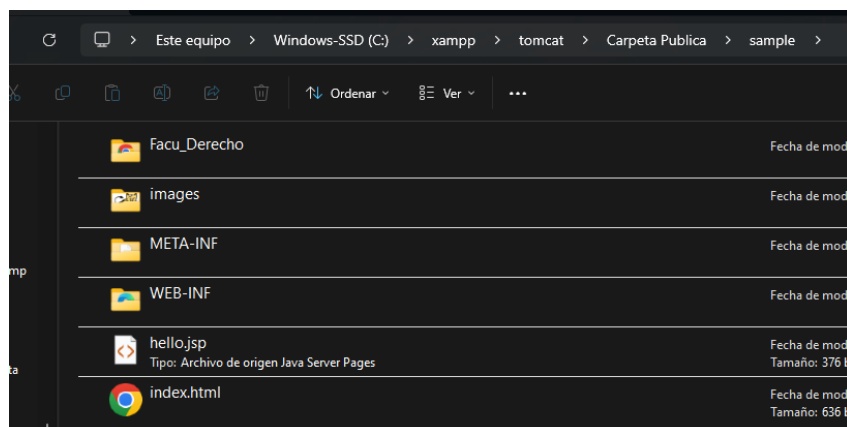


Figura 45: Aplicación web sample reestructurada.

3. De igual forma probamos que no haya ningún problema al desplegar la página:



Figura 46: Apartado de la Facultad de derecho en la aplicación de sample.

- **Configurar el firewall para que sirva exclusivamente en el puerto que se quiera indicar.**

1. Entramos en Kaspersky en Herramientas y nos dirigimos al apartado de **Monitor de red**. Ahí seleccionamos puertos abiertos y buscamos el puerto 8086.

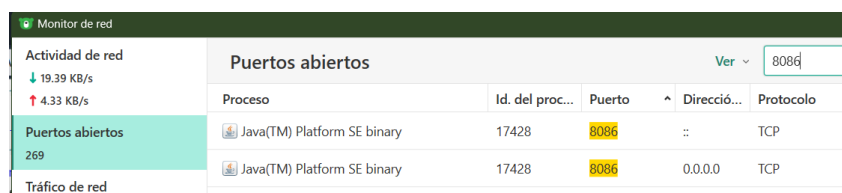
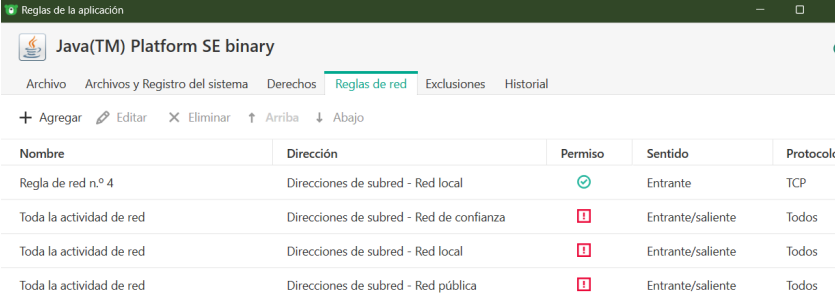


Figura 47: Búsqueda del puerto 8086 en Kaspersky

2. Configuramos una nueva regla que solo incluya al puerto 8086.



Nombre	Dirección	Permiso	Sentido	Protocolo
Regla de red n.º 4	Direcciones de subred - Red local	✓	Entrante	TCP
Toda la actividad de red	Direcciones de subred - Red de confianza	1	Entrante/saliente	Todos
Toda la actividad de red	Direcciones de subred - Red local	1	Entrante/saliente	Todos
Toda la actividad de red	Direcciones de subred - Red pública	1	Entrante/saliente	Todos

Figura 48: Regla solo para el puerto 8086 en Kaspersky

### 3.2. Servidor Cloud

- **Buscar un servidor con trial o transacciones limitadas, revisar los costos, restricciones de tecnología. Ver que ofrecen y hacer una reflexión sobre dicho contenido.**

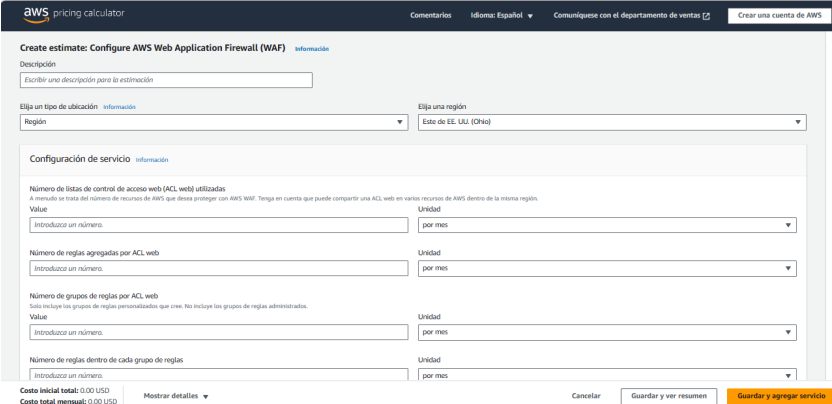
El servidor en la nube elegido es Amazon Web Services (AWS), que ofrece un sistema de pago por uso para la mayoría de sus servicios. Con AWS, solo pagas por los servicios que necesitas mientras los utilizas, sin contratos a largo plazo ni licencias complicadas. Sus precios son comparables a las tarifas de los servicios de agua y electricidad: solo pagas por lo que consumes, y una vez que cancelas el servicio, no hay costos adicionales ni cuotas de cancelación. Las categorías de productos principales son:

#### Explorar las categorías de productos principales



Figura 49: Categorías de productos principales en AWS

Y en total se tiene 321 productos de AWS. De estos 31 son gratuitos para siempre para todo usuario de AWS, 28 tienen doce meses de uso gratis y 55 cuentan con meses de prueba. Para obtener más información sobre su valor es necesario consultar la página de cada uno para obtener información completa sobre los precios. Para un producto que es pagado se tiene la *Calculadora de Precios de AWS*, que claro varía entre productos.



**aws pricing calculator**

Comentarios Idioma: Español Comuníquese con el departamento de ventas Crear una cuenta de AWS

**Create estimate: Configure AWS Web Application Firewall (WAF)**

Descripción  
Escribir una descripción para la estimación

Elige un tipo de ubicación Elige una región  
Región Este de EE. UU. (Ohio)

**Configuración de servicio**

Número de listas de control de acceso web (ACL web) utilizadas  
A medida que creas más listas de control de acceso web (ACL web) que deseas proteger con AWS WAF, ten en cuenta que puedes compartir una ACL web en varios recursos de AWS dentro de la misma región.

Valor Unidad  
Introducir un número por mes

Número de reglas agregadas por ACL web  
Introducir un número por mes

Número de grupos de reglas por ACL web  
Solo incluye los grupos de reglas personalizados que creas. No incluye los grupos de reglas administrados.

Valor Unidad  
Introducir un número por mes

Número de reglas dentro de cada grupo de reglas  
Introducir un número por mes

**Costo inicial total:** 0.00 USD  
**Costo total mensual:** 0.00 USD

Mostrar detalles Cancelar Guardar y ver resumen Guardar y agregar servicio

Figura 50: Calculadora del producto Web Application Firewall (WAF)



The screenshot shows the 'Create estimate: Configure Amazon API Gateway' page. It includes a description field, a region selector set to 'US East (Ohio)', and an 'API de HTTP' section. Under 'API de HTTP', there are fields for 'Unidades de solicitudes de la API HTTP' (set to 'millones') and 'Solicitudes' (set to '34'). There is also a 'Tamaño promedio de cada solicitud' section with a 'Valor' field set to '34' and a unit selector set to 'KB'. At the bottom, it shows 'Total Upfront cost: 0.00 USD' and 'Total Monthly cost: 0.00 USD'. Buttons for 'Cancelar', 'Guardar y ver resumen', and 'Guardar y agregar servicio' are visible.

Figura 51: Calculadora del producto API Gateway

En cuanto a restricciones de tecnología, limitaciones en ciertos servicios o características específicas, se establece que dependería de varios factores como la región, el tipo de cuenta y la configuración de seguridad. Esto debido a que:

- Algunos servicios pueden no estar disponibles en todas las regiones de AWS.
- Alguna tecnología podrían tener imitaciones en su compatibilidad o interoperabilidad con los servicios de AWS. Tecnologías como aplicaciones monolíticas heredadas o tecnologías específicas de un proveedor.
- Algunas tecnologías pueden tener restricciones de red o seguridad que pueden interferir con la conectividad y la interoperabilidad con los servicios de AWS.

A pesar de estas restricciones, no quita que AWS ofrece una plataforma en la nube que es completa y ampliamente adoptada en todo el mundo, con millones de clientes que confían en ella para soportar su infraestructura y aplicaciones. El hecho de que AWS proporcione servicios tecnológicos bajo demanda a través de internet, con precios de pago por uso, permite a las organizaciones reducir costos, ser más ágiles e innovar más rápido.

Como ya se detallo, la plataforma AWS ofrece una amplia gama de servicios y varias características dentro de estos, lo que la hace más rápida, fácil y rentable para mover aplicaciones existentes a la nube y construir nuevas soluciones. Los usuarios pueden utilizar las últimas tecnologías para experimentar e innovar más rápidamente, desde tecnologías para infraestructura, optimización informática, almacenamiento y bases de datos, hasta nuevas tecnologías como el aprendizaje automático, la inteligencia artificial, el análisis de datos y el Internet de las cosas (IoT).

Además, ofrece una amplia variedad de bases de datos diseñadas específicamente para diferentes tipos de aplicaciones, lo que permite a los usuarios elegir la herramienta adecuada para la tarea adecuada. AWS también está constantemente innovando para inventar tecnologías completamente nuevas que los clientes pueden utilizar para transformar sus negocios.

En términos de disponibilidad y durabilidad, AWS ofrece una red global de regiones y zonas de disponibilidad físicamente separadas y altamente redundantes, lo que facilita el diseño y la operación de aplicaciones escalables y tolerantes al fallo con la máxima disponibilidad.

La información sobre Amazon Web Services (AWS) resalta su posición como líder indiscutible en el mercado de la nube. La flexibilidad del sistema de pago por uso, junto con la disponibilidad de una amplia gama de servicios, demuestra el compromiso de AWS con la accesibilidad y la satisfacción del cliente. Sin embargo, es importante tener en cuenta las posibles restricciones en cuanto a la disponibilidad de servicios y la compatibilidad con ciertas tecnologías. A pesar de estas limitaciones, la capacidad de AWS para innovar constantemente y garantizar la disponibilidad y durabilidad de la infraestructura la convierte en una opción atractiva para empresas de todos los tamaños en todo el mundo. En este contexto, a mi parecer AWS sigue siendo una opción sólida y confiable para aquellas organizaciones que buscan migrar sus operaciones a la nube.

## 4. Bibliografía

- [1] J. Andreu, Gestión de servidores web (Servicios en red). Editex, 2011.
- [2] J. Niño, Servidores de aplicaciones web (Aplicaciones web). Editex, 2011
- [3] Rao, Minnesota, Computación en la nube. PHI Aprendizaje Pvt. Limitado, 2015.
- [4] Rosario Peiró, "*Hosting*", Economipedia.com, 12 de abril, 2019.
- [5] G. Gustavo, "*Hosting: ¿Qué es y cómo funciona?*", Hostinger Academia, 2024.
- [6] J. A. Pérez López, *Integración Continua y Despliegue Continuo de una aplicación*, 2023.