

Trabajo 1 - Configuración de Servidores

Jonathan Javier Ortega Molina
jonathan.ortega@ucuenca.edu.ec
Programación Web

CIENCIAS DE LA COMPUTACIÓN
FACULTAD DE INGENIERÍA



**UNIVERSIDAD
DE CUENCA**

Abril 1, 2024

1 Marco Teórico

1.1 Servidor Web

Un servidor web o servidor HTTP, es un programa informático que procesa una aplicación del lado del servidor, realizando conexiones bidireccionales o unidireccionales y síncronas o asíncronas con el cliente y generando o cediendo una respuesta en cualquier lenguaje o aplicación del lado del cliente. Estas conexiones son procesadas por un navegador web y se usa el protocolo HTTP para la comunicación. [1]

1.1.1 Ejemplos:

- **Servidor HTTP Apache:** Software de código abierto y gratuito, destaca por ser multiplataforma.
- **Servidor Microsoft IIS:** Servidor web desarrollado por Microsoft que se ejecuta gracias a la tecnología ISS (Internet Information Services). Es compatible con paginas programadas en ASP o .NET.
- **Servidor Nginx:** Software de Código abierto y multiplataforma. Conocido por bun funcionamiento al gestionar alto número de visitas simultaneas. Es el mas usado del mundo. [2]

1.2 Servidor de Aplicaciones

Los servidores de aplicaciones son programas de servidor en una red distribuida que proporciona el entorno de ejecución para un programa de aplicación. Más específicamente, el servidor de aplicaciones es el componente de tiempo de ejecución principal en todas las configuraciones y donde una aplicación se ejecuta. El servidor de aplicaciones colabora con el servidor web para ofrecer una respuesta dinámica y personalizada a una solicitud de cliente. [3]

1.2.1 Ejemplos:

- Apache Geronimo: para Java EE
- Apache Tomcat: usado para Servlets y JavaServer Pages (JSP)
- WebLogic: usado para aplicaciones Java EE de Oracle Middleware. [4]

1.3 Hosting

Es un servicio online que hace que se pueda acceder a tu sitio web en Internet.Un espacio en un servidor que almacena todos los archivos y datos de tu sitio web.

1.4 Cloud Computing

Es la disponibilidad bajo demanda de recursos de computación como servicios a través de Internet. Evita que las empresas tengan que encargarse de aprovisionar, configurar o gestionar los recursos y permite que paguen únicamente por los que usen. [5]

1.4.1 Ejemplos:

- AWS
- Microsoft Azure
- Google Cloud Platform
- Oracle Cloud [6]

1.5 Despliegue Continuo

Es una estrategia de desarrollo de software en la que los cambios de código de una aplicación se publican automáticamente en el entorno de producción. Se realizan una serie de pruebas predefinidas y una vez que las actualizaciones pasan estas pruebas, el sistema las envía directamente a los usuarios del software. [7]

2 Práctica: Configuración de Servidores

2.1 Apache



Figure 1: Welcome

Para instalar y configurar un servidor Apache HTTP sobre una distribución de Linux se necesitó simplemente de descargar y ejecutar con permisos de administrador un archivo de

instalación ".run". Una vez terminado este proceso se debe acceder al directorio donde fue instalado el servidor y ejecutar el archivo **manager-linux-x64.run** el cual abrirá una ventana la cual se muestra en la Fig. 1

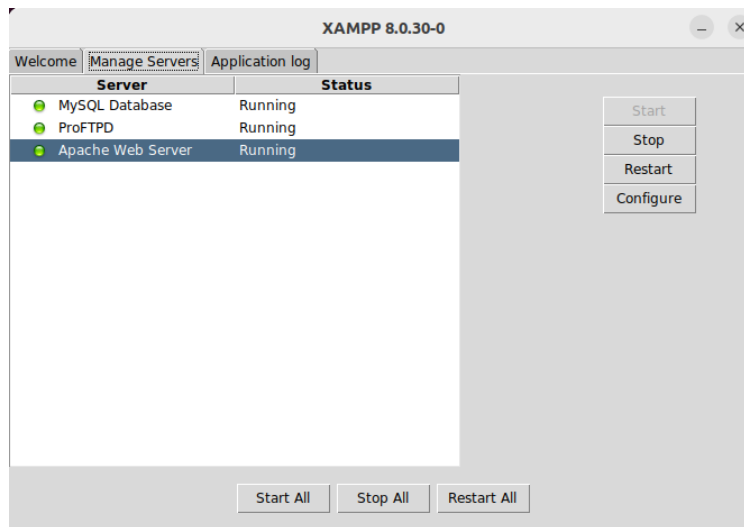


Figure 2: Manager

Para poder levantar o bajar el servidor de Apache, se debe dirigir hacia **Manage Servers** y hacer click sobre el botón **Start All** y **Stop All** respectivamente como se muestra en la Fig. 2

Verificamos que los servicios del servidor están funcionando ingresando las siguientes URL en un navegador de manera local.

`http://localhost/dashboard`

Esto nos dará como resultado algo similar a la fig. 3



Welcome to XAMPP for Linux 8.0.30

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.

XAMPP is meant only for development purposes. It has certain configuration settings that make it easy to develop locally but that are insecure if you want to have your installation accessible to others.

Start the XAMPP Control Panel to check the server status.

Community

XAMPP has been around for more than 10 years – there is a huge community behind it. You can get involved by joining our [Forums](#), liking us on [Facebook](#), or following our exploits on [Twitter](#).

Figure 3: Dashboard

2.2 Apache Tomcat

En el caso de Apache Tomcat se configuró e instaló el servidor de manera manual, creando y configurando los directorios y sus respectivos permisos de usuario. Una vez realizado todo el procedimiento necesario se accede al servidor con la URL y se debe tener algo como la fig.4

`http://localhost:8080`

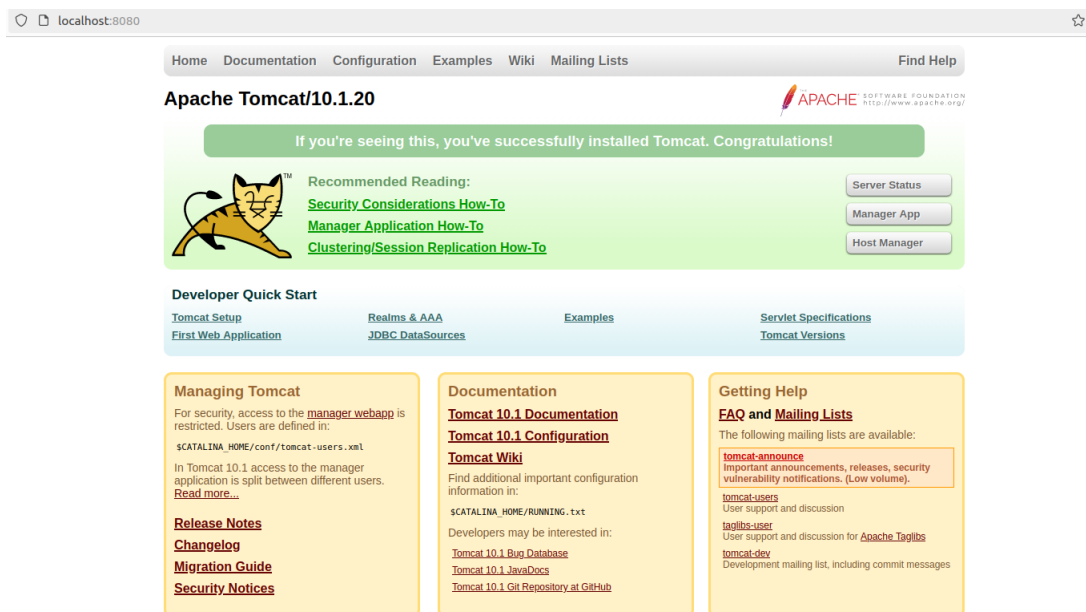


Figure 4: Servidor Tomcat

3 Tareas a Realizar

3.1 Cambiar los puertos para que funcionen en un puerto desde el 8080 hasta el 8089

3.1.1 XAMPP

En XAMPP tenemos dos formas de cambiar el puerto por defecto:

- Acceder al archivo `/XAMPP_HOME/etc/httpd.conf` y se cambiar el valor de "Listen 80" por el nuevo valor con el que queremos que funcione como en la fig. 5
- O, accediendo a la ventana de configuración mostrada en la fig.2, luego accedemos a "configure" y cambiamos el puerto por defecto al puerto que queramos como se muestra en la fig.6

Como paso final, reiniciaremos el servidor y comprobaremos los cambios accediendo al mismo por el nuevo puerto, como se muestra en la fig.7

```
#  
#Listen 12.34.56.78:80  
Listen 8080  
█
```

Figure 5: Puerto XAMPP

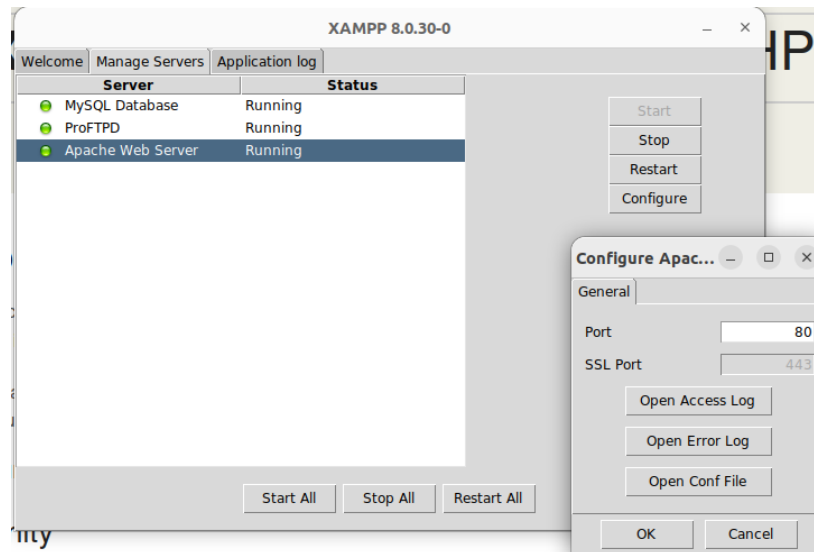


Figure 6: Puerto XAMPP por ventana



Figure 7: XAMPP por puerto 8085

3.1.2 Tomcat

Para el caso de TOMCAT debemos acceder al archivo `/CATALINA_HOME/conf/server.xml` y cambiar en el atributo "port" de la etiqueta "Connector" por el valor nuevo con el que queremos que funcione, como se muestra en la fig.8

De igual manera reiniciamos el servidor y comprobamos accediendo al mismo mediante el nuevo puerto configurado. Fig.9

```
-->  
<Connector port="8087" protocol="HTTP/1.1"  
    connectionTimeout="20000"  
    redirectPort="8443"  
    maxParameterCount="1000"  
/>
```

Figure 8: Puerto Tomcat

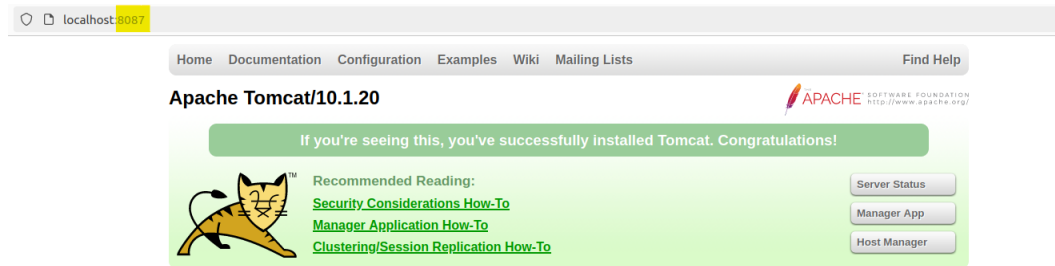


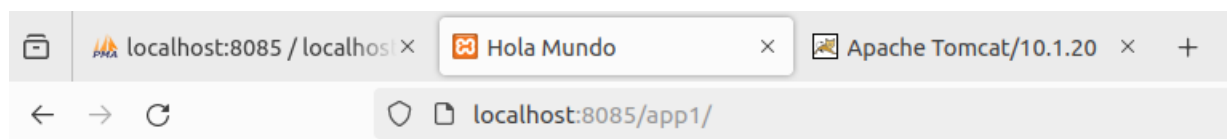
Figure 9: Tomcat por puerto 8087

3.2 Desplegar una página sencilla en cada servidor

3.2.1 XAMPP

1. Nos dirigimos hacia el directorio **htdocs** dentro del HOME de XAMPP. Aquí crearemos un nuevo directorio con el nombre de nuestra aplicación, en este caso **app1**
2. Copiaremos en este caso la página web a desplegar, es decir, su archivo **index.html**
3. Accedemos a nuestra página web con la URL y tendremos el siguiente resultado fig.10

http:localhost:8085/app1/



Página de prueba

Jonathan Ortega

"Hola Mundo "

Figure 10: Página Web en XAMPP

3.2.2 Tomcat

1. Nos dirigimos hacia el directorio **webapps** dentro del directorio CATALINA_HOME de tomcat. Aquí crearemos de igual manera un directorio con el nombre de nuestra aplicación, en este caso **app2**
2. Copiaremos dentro del directorio app2 la página a desplegar, es decir, el archivo **index.html**
3. Accedemos a nuestra página web con la URL y tendremos el siguiente resultado fig.11

http:localhost:8087/app2



Figure 11: Página Web en Tomcat

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

Una carpeta de publicación se refiere a un directorio en el servidor web donde se almacenan los archivos y recursos que se hacen accesibles al público a través de Internet. Estos archivos suelen incluir páginas HTML, imágenes, archivos CSS, archivos JavaScript y otros recursos multimedia que forman parte de un sitio web.

3.3.1 XAMPP

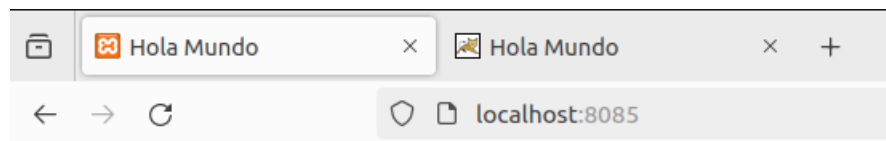
En XAMPP la carpeta de publicación por defecto es el directorio **htdocs** dentro del HOME de XAMPP, que es donde publicamos nuestra página web. Sin embargo, existe la posibilidad de cambiar el directorio de publicación y se debe realizar lo siguiente:

1. Nos dirigimos al archivo de configuración **/XAMPP_HOME/etc/httpd.conf** y cambiamos el atributo de DocumentRoot y Directory por el nuevo directorio raíz donde se alojara nuestro sitio web. Fig.12
2. Desplegamos sobre este directorio nuestro sitio web.

3. Proporcionamos los permisos necesarios al nuevo directorio.
4. Reiniciamos nuestro servidor y verificamos los cambios desde un navegador. Como observamos en la fig.13 ya no es necesario poner un **/app1** en la URL debido a que el nuevo directorio aloja ya nuestro sitio web completo y no solo una página web como lo hacía en el ejemplo anterior fig.10

```
#DocumentRoot "/opt/lampp/htdocs"  
DocumentRoot "/opt/lampp/htdocs/app1"  
<Directory "/opt/lampp/htdocs/app1">
```

Figure 12: Carpeta Pública XAMPP



Página de prueba

Jonathan Ortega

"Hola Mundo "

Figure 13: Sitio Web XAMPP

3.3.2 Tomcat

En Tomcat tenemos que la carpeta de publicación por defecto es el directorio **webapps** dentro del **CATALINA_HOME**. Para cambiar la debemos seguir los siguientes pasos:

1. Nos dirigimos al archivo **/CATALINA_HOME/conf/server.xml** y cambiamos el atributo "appBase" en el elemento "Host". Fig.14
2. Desplegamos nuestro sitio web sobre este nuevo directorio, en el caso de tomcat es necesario un directorio **ROOT** sobre el cual se deberá desplegar el sitio web ya que tomará a este como el directorio raíz.
3. Proporcionamos los permisos necesarios al nuevo directorio.
4. Reiniciamos nuestro servidor y verificamos los cambios desde un navegador. Al igual que XAMPP, no es necesario escribir un **/app2**. Fig.15

```
<Host name="localhost"  appBase="webapps/app2"  
      unpackWARs="true" autoDeploy="true">
```

Figure 14: Carpeta Pública Tomcat

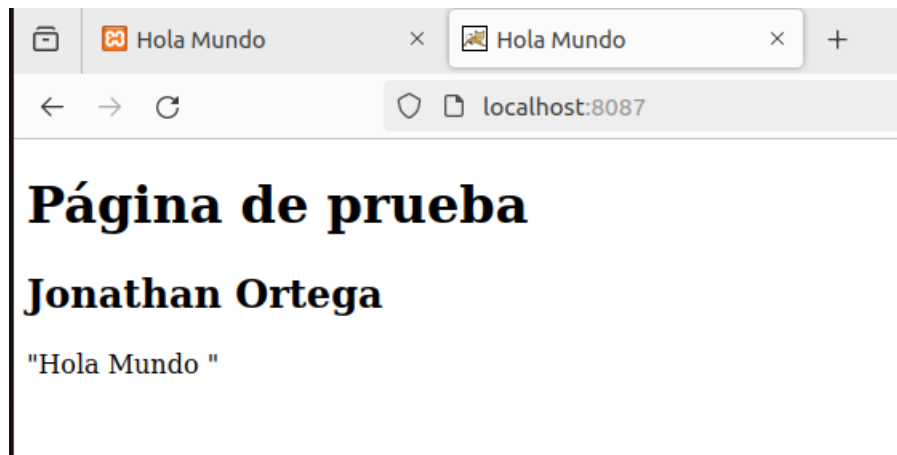


Figure 15: Sitio Web Tomcat

3.4 Estructura recomendada de un árbol de archivos

Una estructura de archivos es la forma de como estará organizado nuestro sitio web utilizando directorios y subdirectorios y dependerá de las necesidades de la aplicación que se esta desarrollando. Aunque no es necesario un árbol de directorios, debido a que podemos poner todos los recursos dentro de una sola carpeta y el sitio web tendrá el funcionamiento esperado. Sin embargo, se recomienda en gran medida crear esta estructura de archivos para organizar nuestro sitio web, poder llevar un control del contenido, mejorar la velocidad de accesos a los recursos, controlar versiones de despliegue, etc. Si bien no existe una plantilla específica para todos los sitios web, en la siguiente imagen se mostrará un ejemplo de como se recomienda crearlo. Fig.16

Dependiendo del tipo de página web tendremos diferentes estructuras. Algunos ejemplos a continuación:

- Departamentales.
- Estructuras por categorías.
- Basadas por fechas.
- Basadas en proyectos.
- Directorios organizados alfabéticamente.

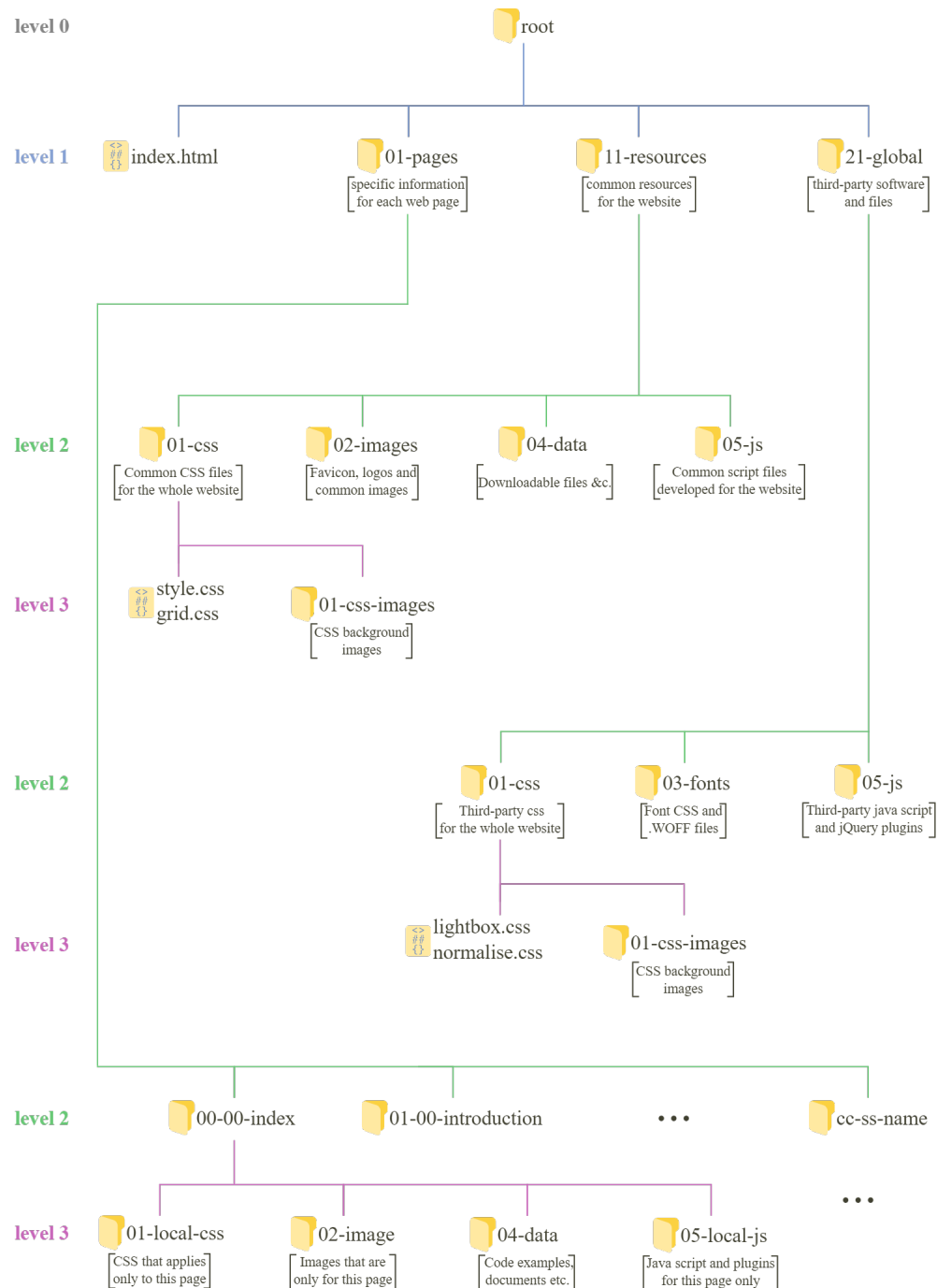


Figure 16: Árbol de archivos

3.5 Configurar puertos de firewall

Cuando tratamos de acceder a nuestro sitio web desde un servidor externo, nos encontraremos con el inconveniente de que no es accesible. Esto se debe a que se necesita configurar y abrir los puertos por donde nuestras solicitudes entrarán hacia nuestro sitio web. Fig17

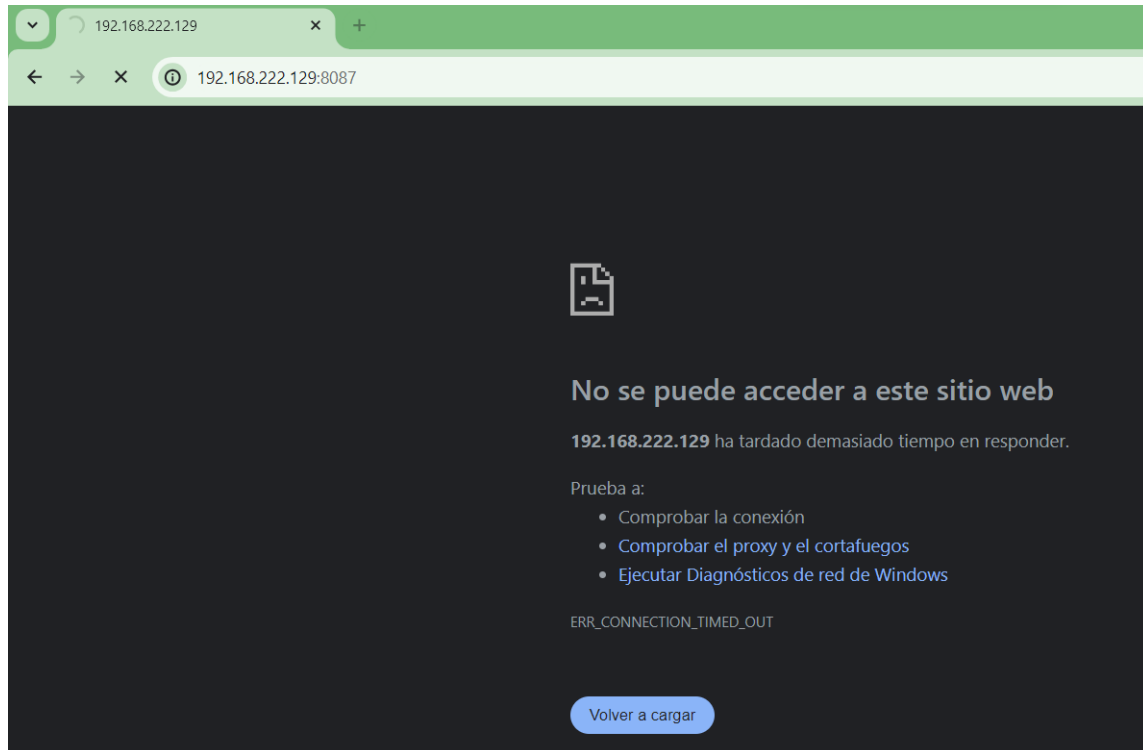


Figure 17: Acceso desde host externo

Para configurar los puertos en nuestro servidor se instaló **firewalld** usando el comando "apt install" y se realizó lo siguiente:

- Abrimos el puerto 8085 para XAMPP utilizando el comando

```
firewall-cmd --zone=public --add-port=8085/tcp --permanent
```

- Abrimos el puerto 8087 para Tomcat utilizando el comando

```
firewall-cmd --zone=public --add-port=8087/tcp --permanent
```

- Volvemos a cargar las configuraciones de firewalld utilizando el siguiente comando

```
firewall-cmd --reload
```

- Comprobamos que los cambios se realizaron volviendo a acceder desde un host externo.
Fig.19

```
root@user:/opt/tomcat/webapps/app2/ROOT# firewall-cmd --zone=public --add-port=8085/tcp --per
manent
success
root@user:/opt/tomcat/webapps/app2/ROOT# firewall-cmd --zone=public --add-port=8087/tcp --per
manent
success
root@user:/opt/tomcat/webapps/app2/ROOT# -firewall-cmd --reload
```

Figure 18: Activando puertos con firewalld

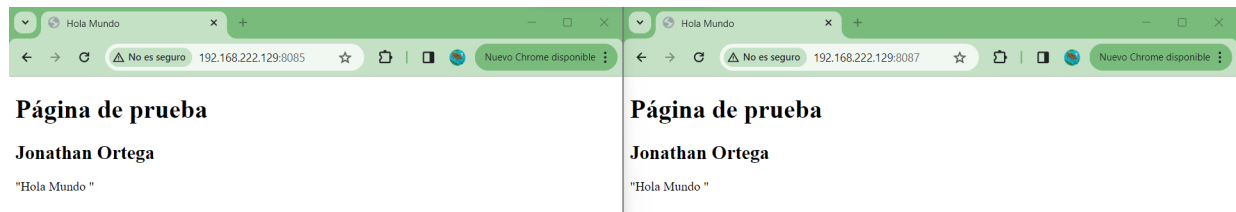


Figure 19: Acceso desde host externo con puertos abiertos

4 Servidor Cloud

AWS (Amazon Web Services) ofrece un sistema de pagos para los precios de la gran mayoría de los servicios de nube. Solo se paga por los servicios individuales que se necesitan durante el tiempo que se los utilice.

4.0.1 Categoría de Productos

- Análisis
- Integración de aplicaciones
- Blockchain
- Administración financiera en la nube
- Computación
- Contenedores
- Bases de datos
- Herramientas para desarrolladores
- Informática para usuarios finales
- Servicios de fronted web y móviles
- Internet de las cosas
- Machine Learning
- Administración y control
- Servicios multimedia

- Migración y transferencia
- Redes y entrega de contenido
- Tecnología cuántica
- Robótica
- Satélite
- Seguridad, identidad y cumplimiento
- Almacenamiento

4.1 Precios

En el caso de Servicios fronted web y móviles existe una prueba gratuita de 12 meses o hasta agotar la capacidad de gratuidad. Fig.20

Aloje una aplicación Gratuito por 12 meses	Aloje una aplicación Pago por uso
Implemente fácilmente su aplicación web o sitio web en la red de entrega de contenido (CDN) de AWS rápida, segura y confiable mediante Amplify Hosting.	¿Necesita más capacidad que el nivel gratuito? Amplify cambia a los precios de pago por uso una vez que supera los límites del nivel gratuito, sin interrupciones en sus aplicaciones.
CREACIÓN E IMPLEMENTACIÓN Sin costo hasta los 1000 minutos de creación por mes	CREACIÓN E IMPLEMENTACIÓN 0,01 USD por minuto
ALMACENAMIENTO DE DATOS Sin costo hasta los 5 GB almacenados en la CDN por mes	ALMACENAMIENTO DE DATOS 0,023 USD por GB por mes (este cargo se repite hasta que se elimine la aplicación)
TRANSFERENCIA SALIENTE DE DATOS Sin coste hasta los 15 GB por mes	TRANSFERENCIA SALIENTE DE DATOS 0,15 USD por GB servido
RECuento DE SOLICITUDES (SSR) Sin coste hasta 500 000 solicitudes por mes	RECuento DE SOLICITUDES (SSR) 0,30 USD por millón de solicitudes
DURACIÓN DE SOLICITUDES (SSR) Sin coste hasta los 100 GB por hora por mes	DURACIÓN DE SOLICITUDES (SSR) 0,20 USD por hora (GB por hora)
La oferta gratuita es por cuenta, independientemente del número de usuarios. Dura hasta que finaliza el periodo de 12 meses del nivel gratuito de AWS y luego se convierte en pago por uso.	Económico, sin tarifa plana por usuario. Incluye múltiples sitios por proyecto y certificados SSL públicos sin costo adicional.

Figure 20: Precios para Servicios Web

Para el caso de servicios de almacenamiento, que es una de las opciones donde podemos desplegar un sitio web estático, también tenemos una prueba gratuita hasta agotar su capacidad. Sin embargo, tenemos los siguientes precios. Fig.21

S3 Standard: almacenamiento de propósito general para cualquier clase de datos que se utiliza generalmente para datos a los que se accede con frecuencia	
Primeros 50 TB/mes	0,023 USD por GB
Siguientes 450 TB/mes	0,022 USD por GB
Más de 500 TB/mes	0,021 USD por GB
S3 Intelligent - Tiering *: almacenamiento con ahorros de costos automáticos para datos con patrones de acceso desconocidos o que cambian constantemente	
Monitoreo y automatización, todo el almacenamiento/mes (Objetos > 128 KB)	0,0025 USD por 1000 objetos
Capa de acceso frecuente, primeros 50 TB/mes	0,023 USD por GB
Capa de acceso frecuente, siguientes 450 TB/mes	0,022 USD por GB
Capa de acceso frecuente, más de 500 TB/mes	0,021 USD por GB
Nivel de acceso poco frecuente, todo el almacenamiento al mes	0,0125 USD por GB
Nivel de acceso instantáneo a archivos, todo el almacenamiento al mes	0,004 USD por GB
S3 Intelligent - Tiering*: niveles opcionales de acceso a archivos asíncronos	
Nivel de acceso a archivos, todo el almacenamiento al mes	0,0036 USD por GB
Nivel de acceso a archivo profundo, todo el almacenamiento al mes	0,00099 USD por GB
S3 Standard - Infrequent Access**: para almacenar datos de larga vida pero con poco acceso que requieren de acceso de milisegundos	
Todo el almacenamiento/mes	0,0125 USD por GB
S3 Express One Zone: almacenamiento de alto rendimiento para los datos a los que se accede con más frecuencia	
Todo el almacenamiento/mes	0,16 USD por GB

Figure 21: Precios para Almacenamiento

4.2 Restricciones

Existen varias restricciones al momento de contratar servicios en la nube. Estas pueden variar dependiendo del proveedor del servicio.

- Ancho de banda: La velocidad puede verse limitada por la infraestructura del proveedor.
- Capacidad de almacenamiento: Si bien los servicios en la nube ofrecen una gran capacidad de almacenamiento, esta no es infinita y por lo tanto puede haber limitaciones; sobre todo cuando exista un crecimiento repentino en el uso del recurso de almacenamiento.
- Latencia: Los servidores están geolocalizados y dependiendo del proveedor de los servicios que contratemos tendremos mas o menos latencia.
- Disponibilidad: Puede existir el caso de que el proveedor deje de estar disponible por tareas de mantenimiento, esto se soluciona fácilmente configurando una alta disponibilidad, pero incrementando los costos.
- Compatibilidad: Diferentes proveedores manejan diferentes tecnologías, dándonos una restricción de compatibilidad que se soluciona contratando diferentes servicios en la nube de diferentes proveedores.

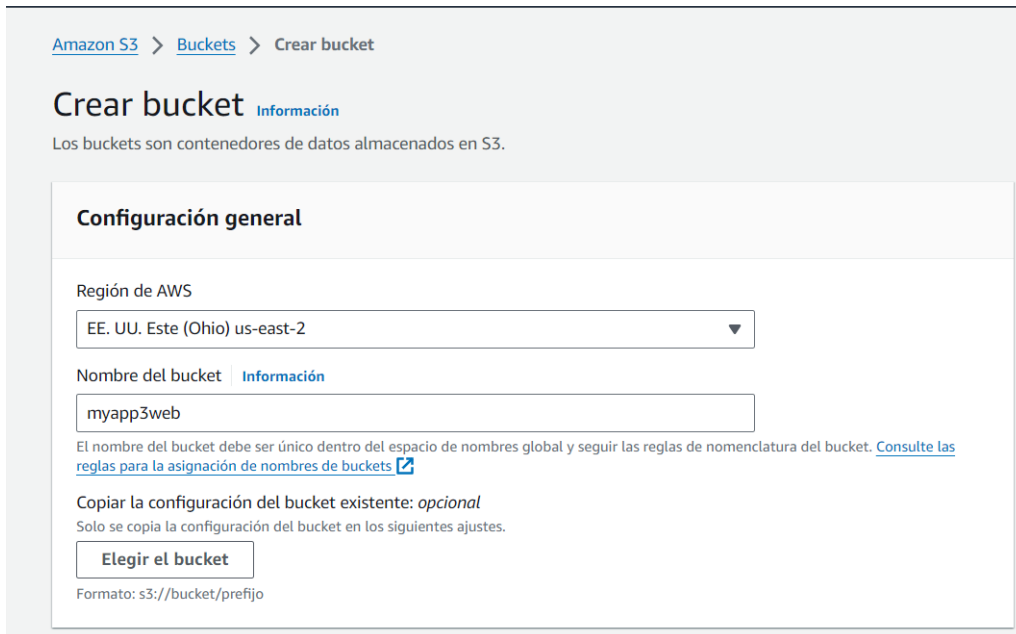
Un ejemplo de restricción de almacenamiento de AWS es el siguiente: "El tamaño de los objetos individuales de Amazon S3 puede variar entre un mínimo de 0 bytes y un máximo de 5 TB. El objeto más grande que se puede cargar en un solo PUT es de 5 GB."

4.3 Despliegue de página web en AWS

Crearemos un despliegue de una página web estática creando un bucket en el servicio Amazon S3.

1. Creamos un bucket público. Fig.22
2. Subimos la carpeta de nuestro sitio web, en este caso simplemente nuestro archivo index.html. Fig.23
3. Nos dirigimos a propiedades, alojamiento de sitios web. Y activamos esta propiedad. Fig.24
4. Generamos una política para que los objetos del bucket sean públicos. Fig.25
5. Finalmente accedemos a la página web con el siguiente URL, fig.26

`http://myapp3web.s3-website.us-east-2.amazonaws.com/`



Amazon S3 > Buckets > Crear bucket

Crear bucket Información

Los buckets son contenedores de datos almacenados en S3.

Configuración general

Región de AWS

EE. UU. Este (Ohio) us-east-2 ▼

Nombre del bucket Información

myapp3web

El nombre del bucket debe ser único dentro del espacio de nombres global y seguir las reglas de nomenclatura del bucket. [Consulte las reglas para la asignación de nombres de buckets](#)

Copiar la configuración del bucket existente: *opcional*

Solo se copia la configuración del bucket en los siguientes ajustes.

Elegir el bucket

Formato: s3://bucket/prefijo

Figure 22: Creacion de Bucket

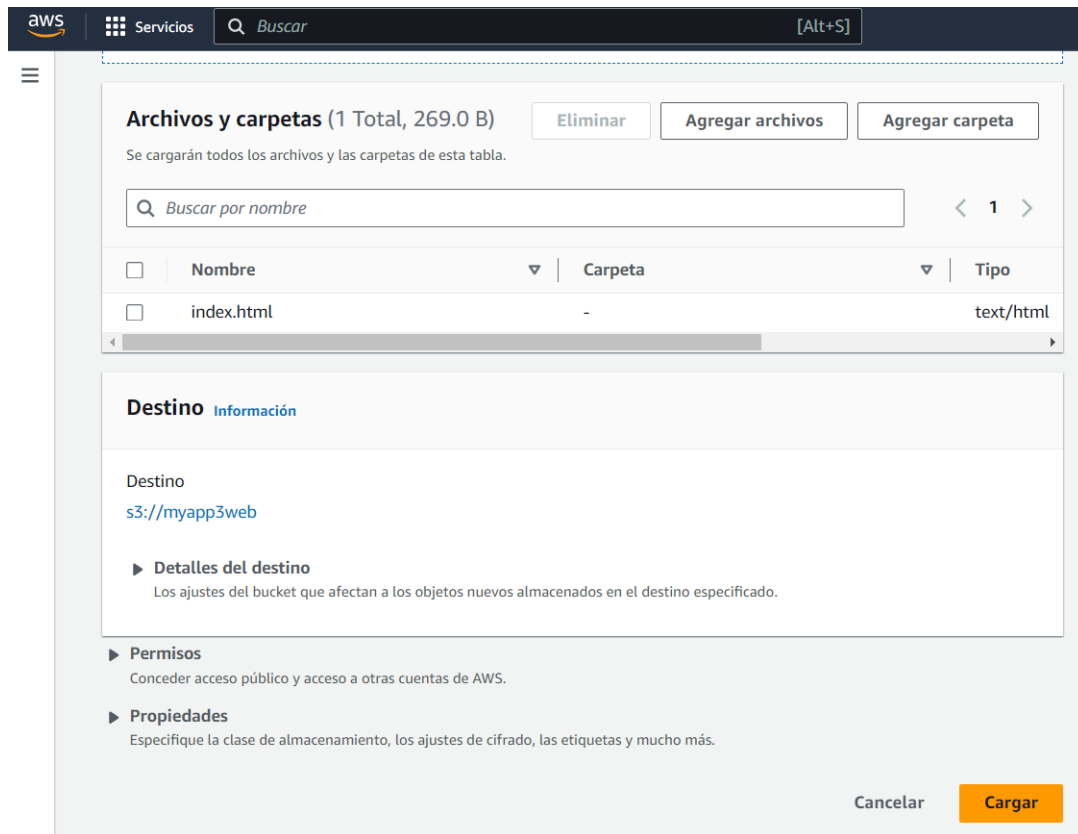


Figure 23: Cargar archivo index.html

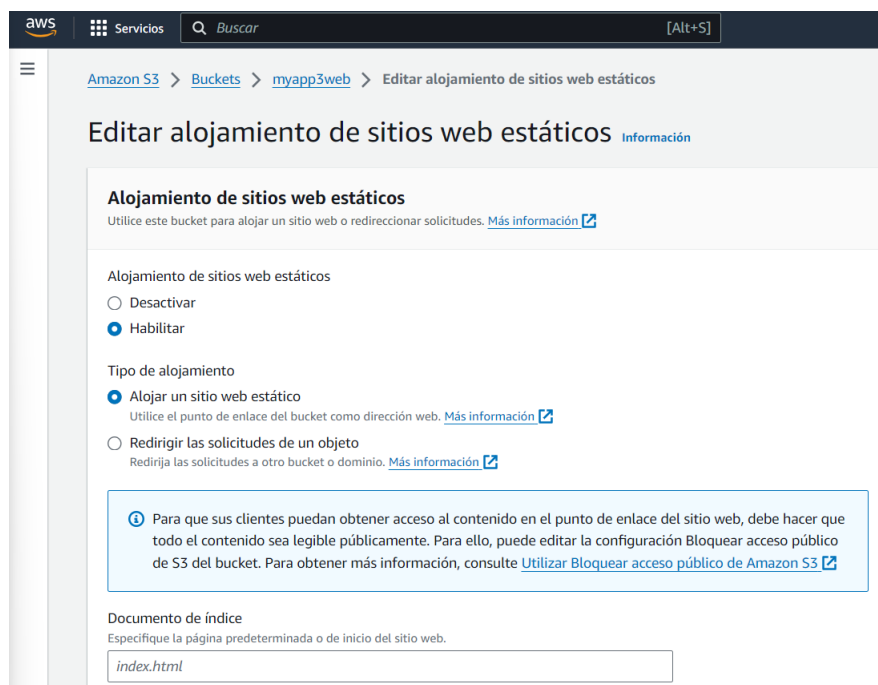


Figure 24: Configuración de alojamiento sitios web

**Figure 25:** Generar policy**Figure 26:** Página en AWS

4.4 Reflexión

Una de las mayores ventajas de los servicios en la nube es su capacidad para adaptarse a las necesidades cambiantes de las organizaciones. Con la infraestructura basada en la nube, las empresas pueden escalar rápidamente sus recursos según la demanda, lo que les permite manejar tráfico y cambios en las cargas de trabajo de manera más eficiente que con un servidor local. Sin embargo, es importante tener en cuenta que los servicios en la nube también tienen sus limitaciones. Por ejemplo, la seguridad y la privacidad de los datos siguen siendo preocupaciones importantes. Además, el costo total de propiedad de los servicios en la nube puede variar según el proveedor, el tipo de servicio y el uso real de los recursos.

References

- [1] Wikipedia. *Servidor web*. Mar. 2024. URL: https://es.wikipedia.org/wiki/Servidor_web.
- [2] Gerard Martínez. *¿Qué es un servidor web y para qué sirve?* - Webempresa. Aug. 2023. URL: https://www.webempresa.com/hosting/que-es-servidor-web.html#Tipos_de_servidores_Web.
- [3] IBM documentation. URL: <https://www.ibm.com/docs/es/i/7.4?topic=serving-application-servers>.
- [4] Equipo editorial de IONOS. *¿Qué son los servidores de aplicaciones?* Aug. 2023. URL: <https://www.ionos.es/digitalguide/servidores/know-how/servidor-de-aplicaciones/>.
- [5] *¿Qué es cloud computing?* | Google Cloud. URL: <https://cloud.google.com/learn/what-is-cloud-computing>.
- [6] Frankier Flores. *Cloud Computing: Tipos de nubes, servicios y proveedores*. Apr. 2023. URL: <https://openwebinars.net/blog/tipos-de-cloud-computing/>.
- [7] *¿Qué es el despliegue continuo?* | IBM. URL: <https://www.ibm.com/es-es/topics/continuous-deployment>.