

Configuración de Servidores

Juliana Nicole Abril Cabrera
nicole.abrilc@ucuenca.edu.ec

ABSTRACT

This activity aims to understand the configuration of local web servers, as well as the configuration of application servers. In addition, the aim is to understand the difference between a local web server and a server that can be deployed in the cloud.

RESUMEN

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

Palabras clave: Apache, Programación Web, Tomcat, servidor local, servidor en la nube, Cloud Computing.

1. INTRODUCCIÓN

En la era digital actual, la infraestructura tecnológica es fundamental para el funcionamiento eficiente de cualquier organización, ya sea una empresa, una institución educativa o una entidad gubernamental. Dos elementos clave en esta infraestructura son los servidores locales y el cloud computing, cada uno con sus propias ventajas y desafíos.

Los servidores locales, instalados físicamente en las instalaciones de la empresa u organización, han sido la columna vertebral de la infraestructura informática durante décadas. Ofrecen control directo sobre los recursos de hardware y permiten personalizar la configuración según las necesidades específicas de la organización. Sin embargo, la gestión y el mantenimiento de servidores locales pueden ser costosos en términos de tiempo, dinero y recursos humanos. Es por ello que este trabajo, se encarga de enseñar el uso y la configuración tanto de servidores locales como servidores en la nube.

2. MARCO TEÓRICO

• Servidor Web

Un servidor web es un componente de software que entrega datos estáticos como imágenes, archivos y texto en respuesta a las solicitudes de los clientes. Es decir, aloja el código y los datos de un sitio web.

El servidor web contiene un identificador que funciona como una dirección única que permite ingresar al servidor web desde el navegador, es decir el url. El navegador y servidor web se comunican de la siguiente manera:

1. El navegador usa la url para encontrar la dirección IP del servidor, luego envía una solicitud HTTP de información.
2. El servidor web se comunica con un servidor de base de datos para encontrar los datos relevantes. De la misma forma, devuelve contenido estático, como páginas html, imágenes, videos o archivos, en una respuesta HTTP al navegador. A continuación, el navegador le mostrará la información.

Un sitio web que aloja contenido estático, como blogs, imágenes de encabezado o artículos, puede ejecutarse en un servidor web. Sin embargo, hoy en día la mayoría de los sitios web y aplicaciones web son mucho más interactivos y requieren un servidor de aplicaciones. [1]

Ejemplos:

- Servidor HTTP Apache
- Servidor Microsoft IIS
- Servidor Nginx [2]

• Servidor de Aplicaciones

Un servidor de aplicaciones agrega lógica empresarial para calcular la respuesta del servidor web. Es decir, amplía las capacidades de un servidor web, pues admite la generación de contenido dinámico, la lógica de la aplicación y la integración con varios recursos. Proporciona un entorno de tiempo de ejecución en el que se puede ejecutar el código de la aplicación e interactuar con otros componentes de software, como los sistemas de mensajería y las bases de datos.

Cuando intenta acceder al contenido interactivo de un sitio web, el proceso funciona de la siguiente manera:

1. El navegador usa la URL para encontrar la dirección IP del servidor y envía una solicitud HTTP de información.
2. El servidor web transfiere la solicitud al servidor de aplicaciones. El servidor de aplicaciones aplica la lógica empresarial y se comunica con otros servidores y sistemas de terceros para cumplir con la solicitud.
3. El servidor de aplicaciones representa una nueva página HTML y la devuelve como respuesta al servidor web y este le devuelve una respuesta al navegador.
4. El navegador muestra la información.

Para usar el ejemplo de un sitio web de comercio electrónico, cuando agrega artículos a su carrito o retira artículos, interactúa con el servidor de aplicaciones. [1]

Ejemplos:

- Apache Geronimo
- Apache Tomcat
- ColdFusion
- GlassFish [3]

• Hosting

Un hosting es un servicio de alojamiento web que permite publicar un sitio web o aplicación en Internet. Cuando se contrata un servicio de hosting, básicamente se alquila un espacio en

un servidor físico donde puede almacenarse todos los archivos y datos necesarios para que un sitio web funcione correctamente. Cuando se decide crear una página web nueva, se tiene que encontrar una empresa de hosting que proporcione espacio en un servidor. El proveedor de hosting almacena todos los archivos, medios y bases de datos en el servidor.

Cada vez que alguien escribe el nombre de dominio en la barra de direcciones de su navegador, el servidor transfiere todos los archivos necesarios para atender la solicitud. Un proveedor de servicios de alojamiento web garantiza que la página web funcione de forma óptima y con mejores protocolos de seguridad. Además, simplifica los numerosos y complejos aspectos del hosting de una página web, desde la instalación del software hasta el soporte técnico. [4]

• Cloud Computing

Cloud computing es la disponibilidad bajo demanda de recursos de computación como servicios a través de Internet. Esta tecnología evita que las empresas tengan que encargarse de aprovisionar, configurar o gestionar los recursos y permite que paguen únicamente por los que usen.

Hay tres modelos diferentes de despliegue de cloud computing: nube pública, nube privada y nube híbrida.

- Las nubes públicas las ejecutan proveedores de servicios en la nube externos. Ofrecen recursos de computación, de almacenamiento y de red por Internet, lo que permite a las empresas acceder a recursos bajo demanda compartidos en función de sus requisitos únicos y de sus objetivos empresariales.
- Las nubes privadas se crean, gestionan y pertenecen a una sola organización, y se alojan de forma privada en sus propios centros de datos, lo que comúnmente se denomina "on-premise". Proporcionan más control, seguridad y gestión de datos, al mismo tiempo que permiten que los usuarios internos se beneficien de un conjunto compartido de recursos de computación, almacenamiento y redes.
- Las nubes híbridas combinan modelos de nube pública y privada con los que las empresas pueden aprovechar los servicios de nube pública y mantener las funciones de cumplimiento y seguridad habituales. en arquitecturas de nube privada. [5]

• Despliegue continuo

El 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. 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.

Aunque “despliegue continuo” y “entrega continua” pueden parecer lo mismo, en realidad son dos métodos distintos de publicación frecuente.

La entrega continua es una práctica de desarrollo de software en la que el software se crea de forma que se puede publicar en producción en cualquier momento dado. Una vez que el código pasa todas las pruebas, en la entrega continua se requiere intervención humana para aprobar los despliegues en producción. El despliegue continuo amplía un poco más la automatización y elimina la necesidad de intervención manual. [6]

3. PRÁCTICA

a. Configurar 3 tipos de servidores

Configurar:

- APACHE (WAMP O XAMPP): Puede ser en Windows, Linux o macOS, no importa la distribución.
- APACHE TOMCAT: Buscar un script del lado del servidor de hola mundo y desplegarlo.
- UN SERVIDOR EN CLOUD COMPUTING: Puede ser en Azure, Heroku, Google Engine, Amazon Web Services, o el que usted elija.

4. EJERCICIOS A REALIZAR

Se debe realizar un paso a paso, tipo tutorial con capturas de pantalla mostrando lo siguiente

a. Servidores locales

i. APACHE XAMPP

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

Para lograr cambiar el puerto en Apache Xampp, se debe entrar en el archivo de configuración “httpd.conf”, ubicado en la carpeta xampp/apache/conf o desde el panel de control de Xampp.

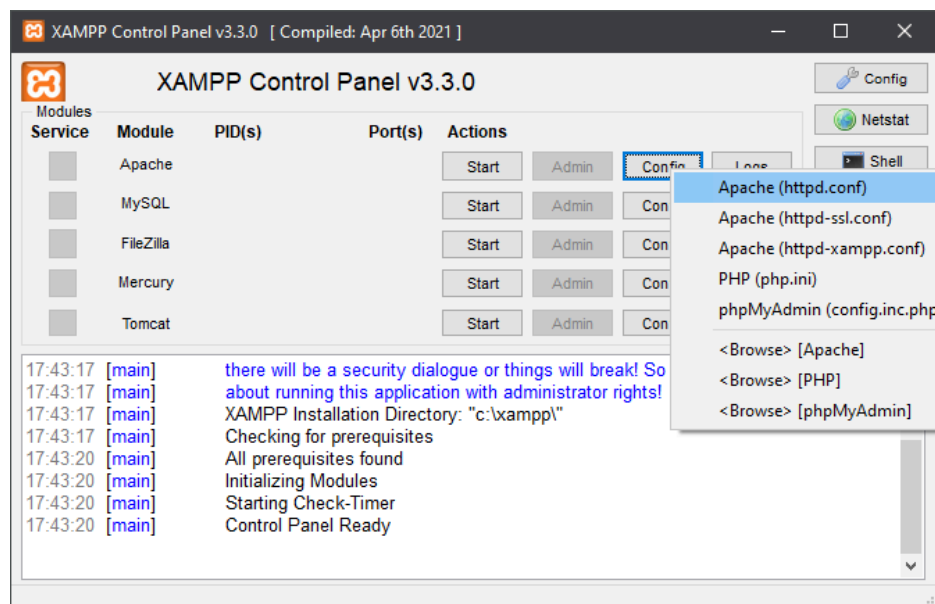


Fig 1. Panel de Control de Xampp.

Luego se busca cambiar la línea Listen 80 por Listen 8081 para modificar el puerto por el que estará escuchando Xampp, por último se guardan los cambios y se reinicia Xampp.

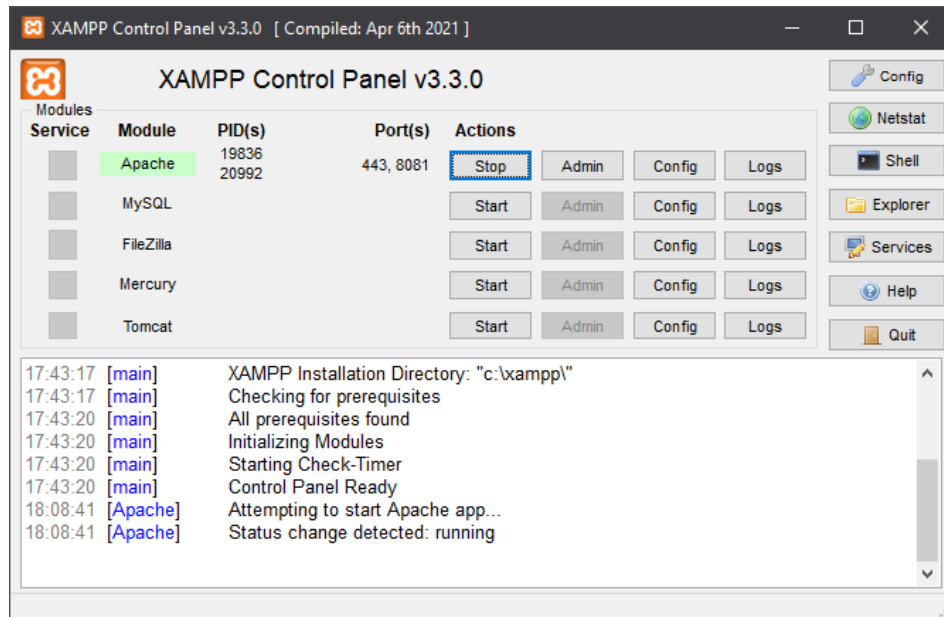


Fig 4. Panel de Control de Xampp.

Se ingresa al url <http://localhost:8081/index.html> para verificar que esté funcionando el servidor y nuestra aplicación se está desplegando.

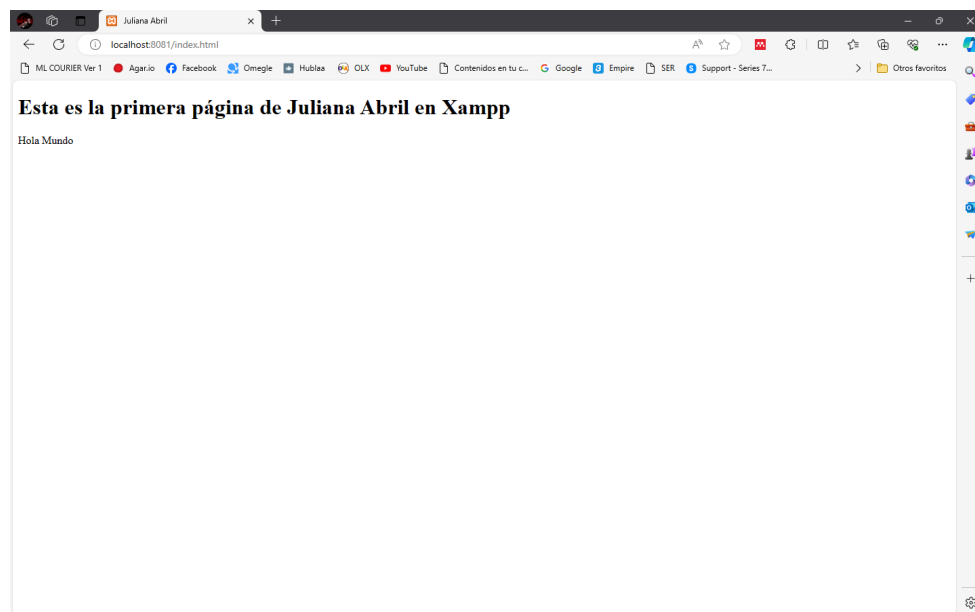
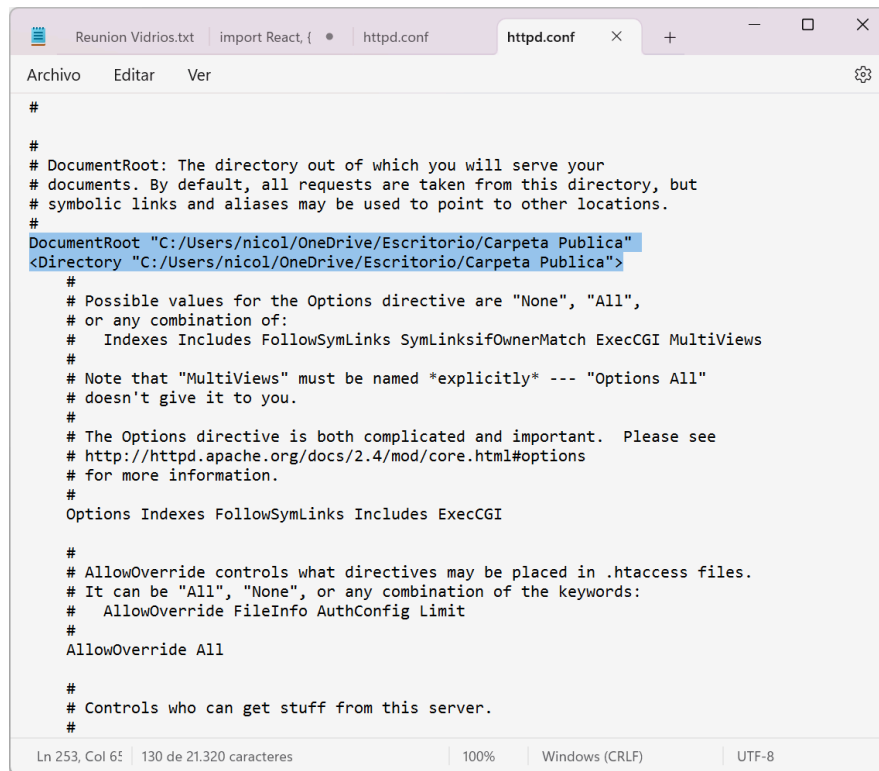


Fig 5. Página web de prueba.

3. Crear una carpeta que se llame "Carpeta Pública" y hacer que sea la carpeta de publicación (investigar).

Para crear una carpeta de publicación en Xampp, se debe crear una carpeta llamada "Carpeta Pública". Se abre el archivo de configuración httpd.conf y se busca la sección donde se define

el directorio raíz o directorio de publicación. Se debe cambiar por la dirección de la nueva carpeta de publicación.

A screenshot of a text editor window showing the httpd.conf file. The window has tabs for 'Reunion Vidrios.txt', 'import React, {', and 'httpd.conf'. The editor shows the configuration for the DocumentRoot, which is set to 'C:/Users/nicol/OneDrive/Escritorio/Carpeta Publica'. The file contains various comments and directives for the Apache server configuration.

```
#
#
# 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:/Users/nicol/OneDrive/Escritorio/Carpeta Publica"
<Directory "C:/Users/nicol/OneDrive/Escritorio/Carpeta Publica">
    #
    # Possible values for the Options directive are "None", "All",
    # or any combination of:
    #   Indexes Includes FollowSymLinks SymLinksifOwnerMatch ExecCGI MultiViews
    #
    # Note that "MultiViews" must be named *explicitly* --- "Options All"
    # doesn't give it to you.
    #
    # The Options directive is both complicated and important. Please see
    # http://httpd.apache.org/docs/2.4/mod/core.html#options
    # for more information.
    #
    Options Indexes FollowSymLinks Includes ExecCGI
</Directory>
#
# AllowOverride controls what directives may be placed in .htaccess files.
# It can be "All", "None", or any combination of the keywords:
#   AllowOverride FileInfo AuthConfig Limit
#
AllowOverride All
#
# Controls who can get stuff from this server.
#
```

Fig 6. Archivo de Configuración de Apache Servers.

Por último, se guardan los cambios y se reinicia el servidor Apache desde el panel de control de XAMPP. Y se verifica copiando el archivo index.html en la carpeta pública.

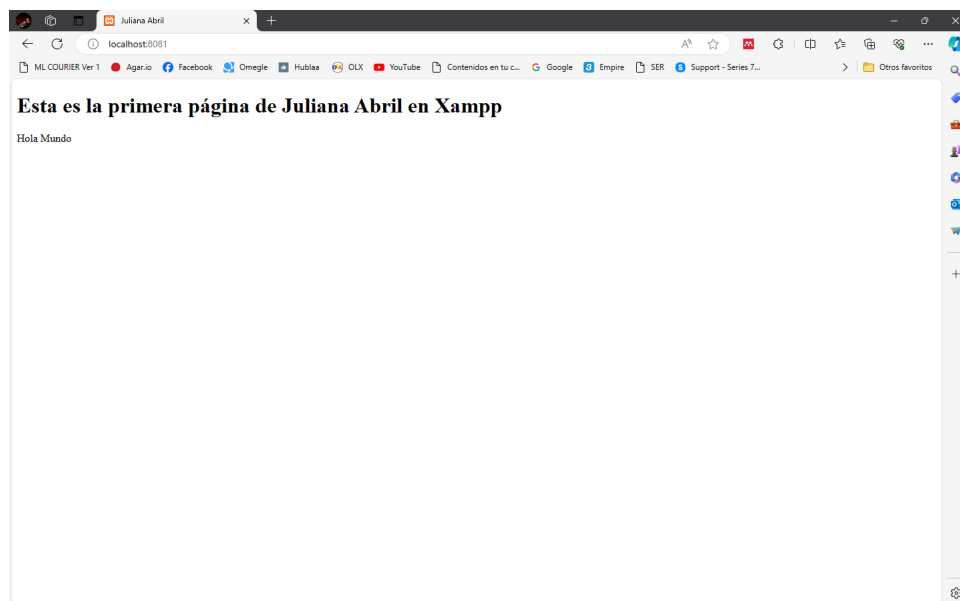


Fig 7. Página de prueba ejecutada.

4. 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 en cada una de ellas. Por ejemplo: En la universidad tuviera una carpeta en donde esté la página de inicio y una subcarpeta de cada facultad y dentro de ella carpetas de imágenes, css, etc. (Consultar en la bibliografía convenciones básicas de estructuras de carpetas).

Un ejemplo de estructura de archivos para publicación puede ser la siguiente, ya que contiene una carpeta dedicada a las imágenes, css, scripts, plantillas, etc. Y los archivos principales van fuera de las carpetas para que el servidor las pueda desplegar sin problema.

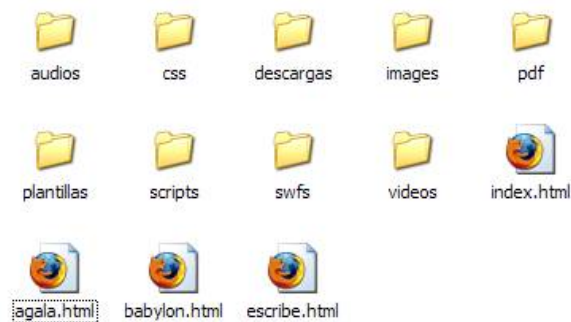


Fig 8. Estructura de archivos obtenida de [7].

Una estructura de carpetas bien planificada permite tener una visión clara de todos los archivos y recursos que conforman un sitio web. Esto facilita la colaboración entre desarrolladores y diseñadores, ya que cada uno sabe dónde encontrar y trabajar en sus respectivas partes del proyecto.

Como un ejemplo de un árbol de archivos, se ha descargado un proyecto sobre una página de comida desde el repositorio de Github en el siguiente enlace [GitHub - Aaron-Shrike/tiki-restobar: Sistema de administración y Página Web para Tiki Restobar con HTML, CSS, JS, PHP y MySQL. Proyecto del séptimo semestre de estudios.](https://github.com/Aaron-Shrike/tiki-restobar)

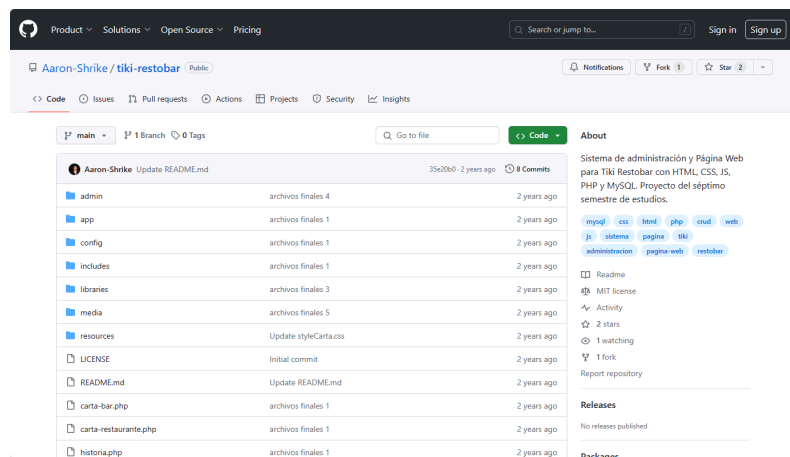


Fig 9. Proyecto en Github.

Desplegando el proyecto en la carpeta pública establecida se obtiene lo siguiente en el navegador:

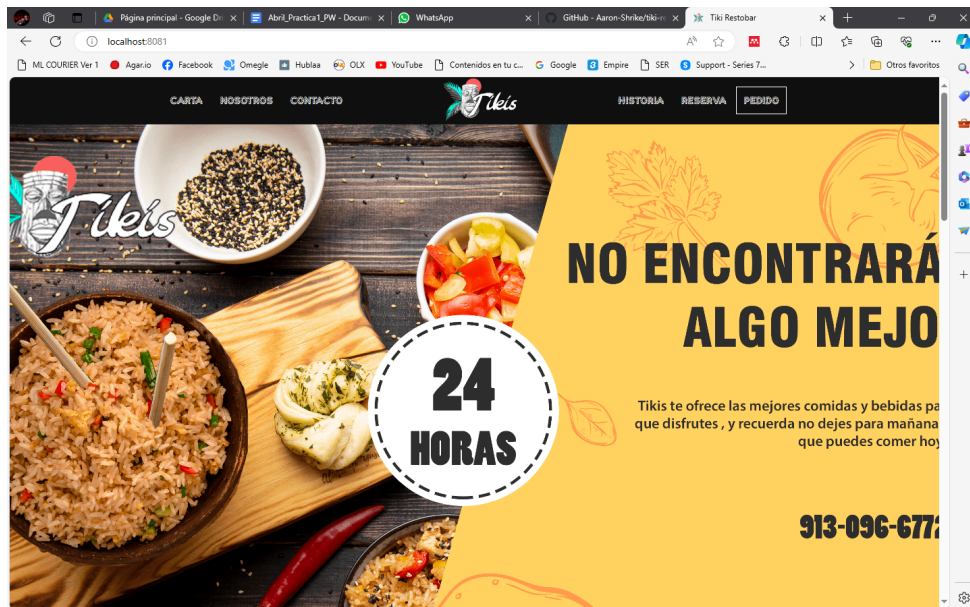


Fig 10. Proyecto montado en Apache Server.

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

Para configurar el firewall se debe ingresar a la seguridad de windows y a configuración avanzada eso se realiza abriendo la configuración del firewall del sistema operativo. Y se debe agregar una regla de entrada para permitir el tráfico en el puerto 8081.

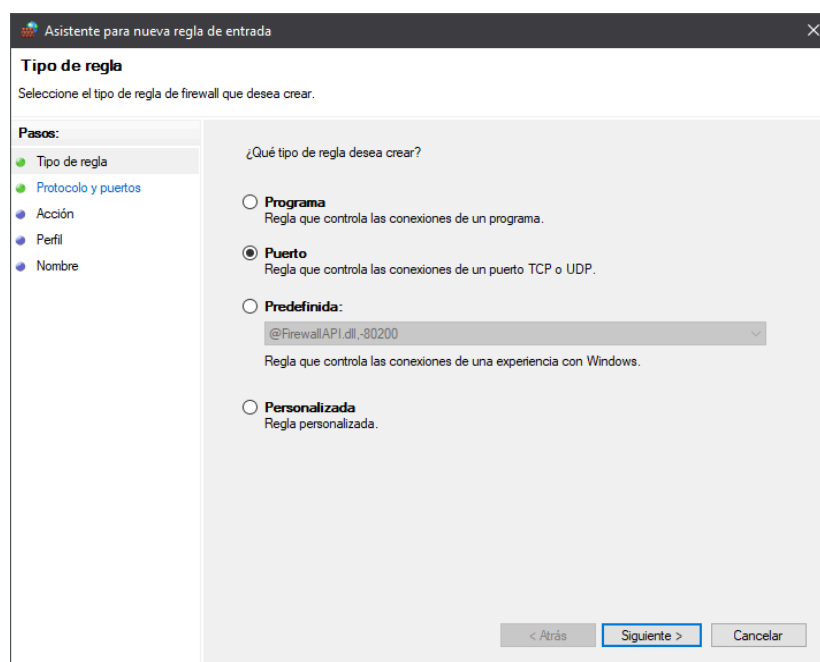


Fig 11. Configuración de Firewall.

Se configura el tipo de protocolo a usar, tipo de conexión, perfiles con acceso y por último el nombre de la regla de entrada.

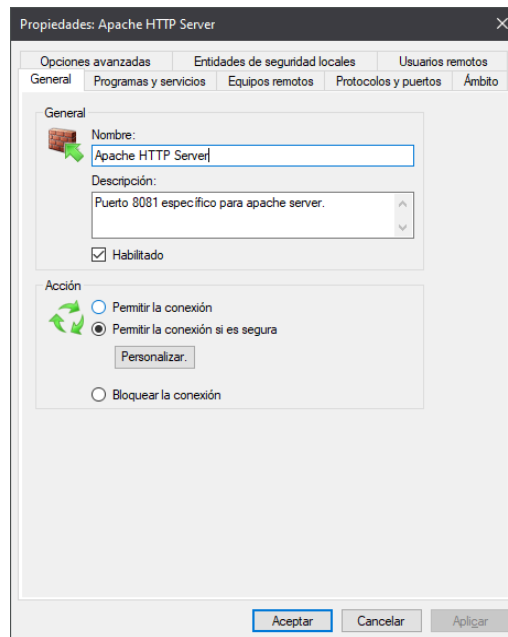


Fig 12. Agregar nueva regla de entrada.

ii. APACHE TOMCAT

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

Para cambiar el puerto de apache Tomcat, se debe dirigir al archivo de configuración server.xml en la sección de Service name Catalina y en la etiqueta Connector cambiar el puerto, en este caso por el puerto 8082. Por último se reinicia el servidor.

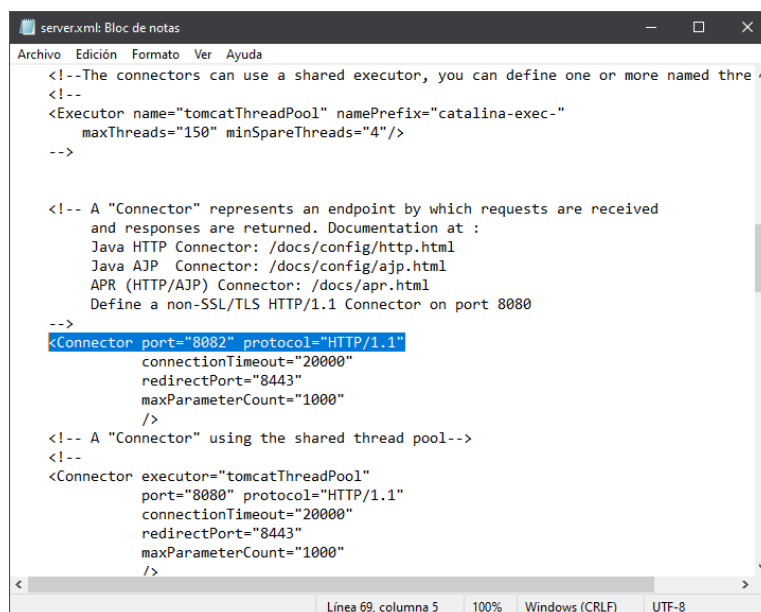


Fig 13. Archivo de Configuración de Apache Tomcat.

Se comprueba que el servidor esté corriendo en el puerto 8082.

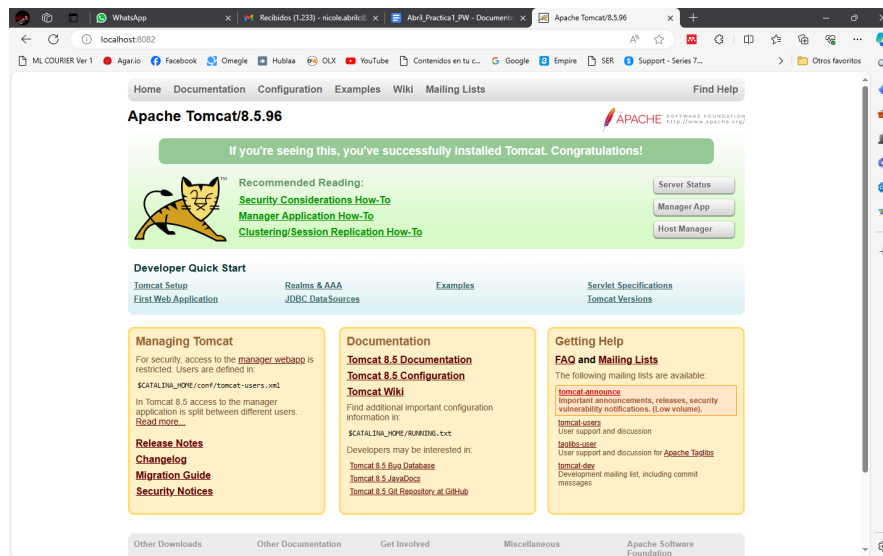


Fig 14. Ventana principal de Apache Tomcat.

2. Desplegar una página Web sencilla de hola mundo en cada uno.

Para desplegar una página web se coloca una página HTML en esta caso, index.html en el directorio webapps/ROOT de la instalación de Tomcat. Si no hay una carpeta llamada ROOT, se debe crearla.

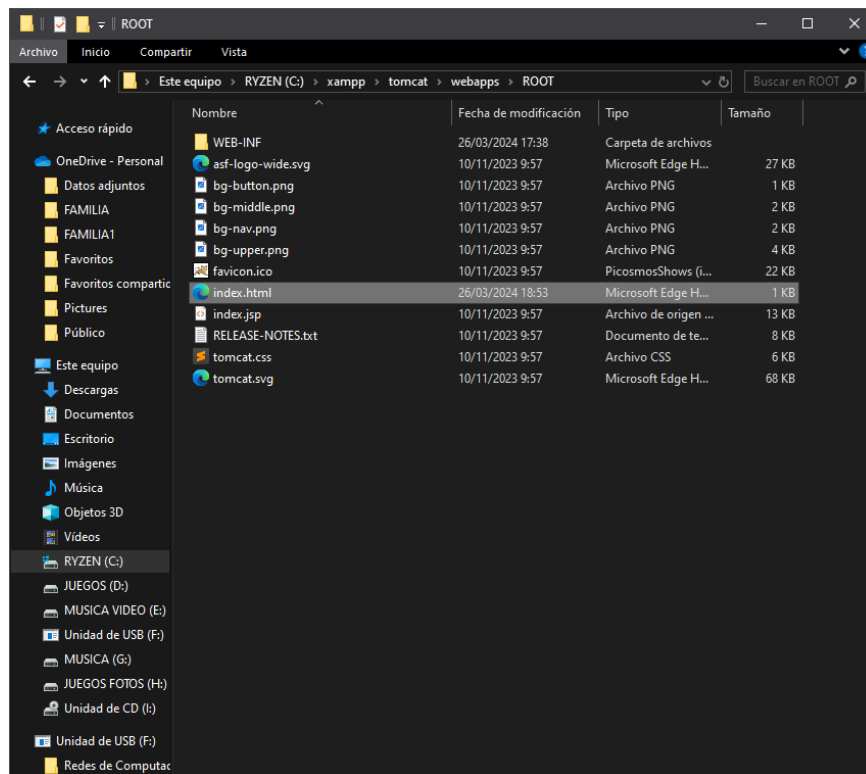


Fig 15. Directorio raíz de Apache Tomcat.

Por último se verifica que la página web esté desplegada en el servidor tomcat.

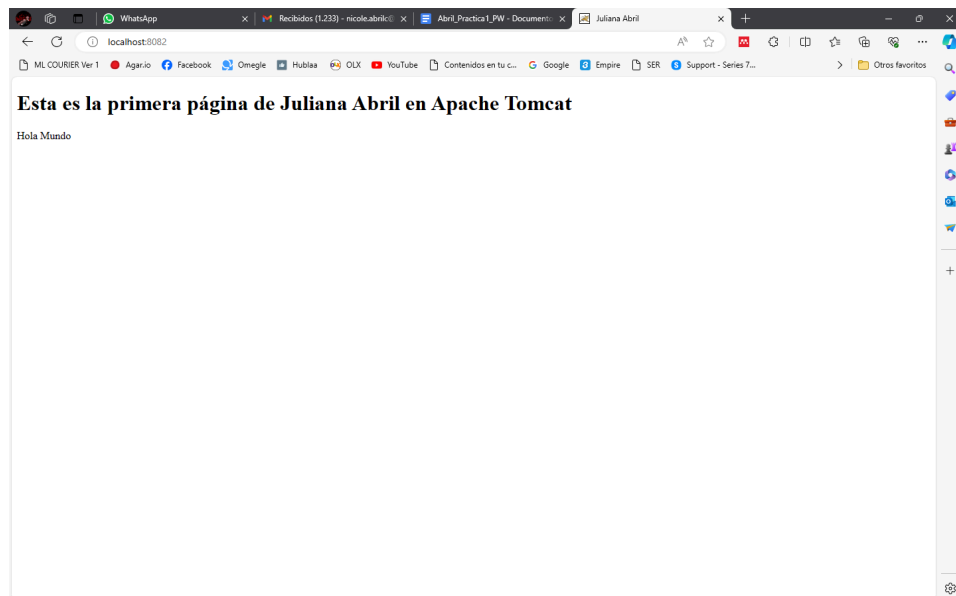


Fig 16. Página de prueba ejecutada.

3. Crear una carpeta que se llame "Carpeta Pública" y hacer que sea la carpeta de publicación (investigar).

Para crear una carpeta de publicación en Apache Tomcat, primero se debe crear una carpeta pública en cualquier directorio y luego se empieza con la configuración de la carpeta de publicación en el archivo conf/server.xml.

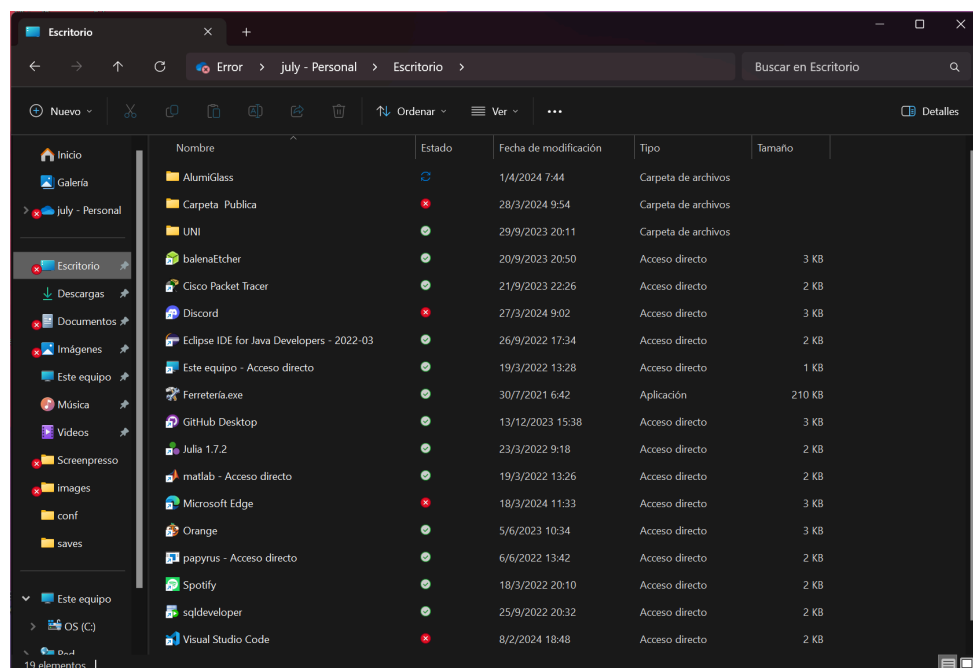
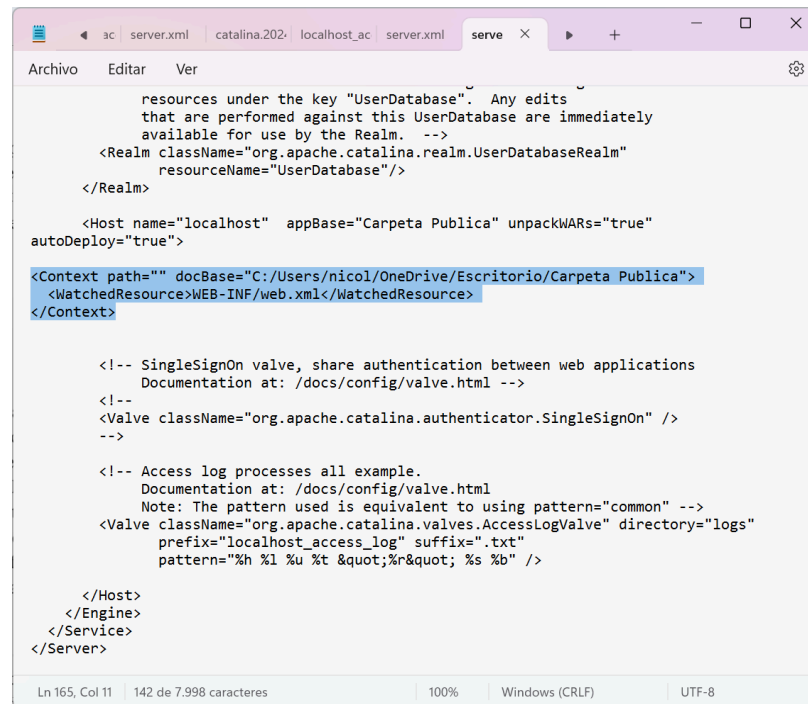


Fig 17. Directorio de la carpeta pública.

Ahora, para configurar la carpeta como la carpeta de publicación se abre el archivo de configuración conf/server.xml en el directorio de instalación de Apache Tomcat con un editor de texto. Se busca la etiqueta <Context> que se relaciona con la aplicación raíz. Y se coloca

la ruta de la carpeta pública creada. Si no existe la etiqueta se debe crearla dentro de host y encima de `<Valve>`.



```
resources under the key "UserDatabase". Any edits
that are performed against this UserDatabase are immediately
available for use by the Realm. -->
<Realm className="org.apache.catalina.realm.UserDatabaseRealm"
resourceName="UserDatabase"/>
</Realm>

<Host name="localhost" appBase="Carpeta Publica" unpackWARs="true"
autoDeploy="true">

  <Context path="" docBase="C:/Users/nicol/OneDrive/Escritorio/Carpeta Publica">
    <WatchedResource>WEB-INF/web.xml</WatchedResource>
  </Context>

  <!-- SingleSignOn valve, share authentication between web applications
  Documentation at: /docs/config/valve.html -->
  <!--
  <Valve className="org.apache.catalina.authenticator.SingleSignOn" />
  -->

  <!-- Access log processes all example.
  Documentation at: /docs/config/valve.html
  Note: The pattern used is equivalent to using pattern="common" -->
  <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
  prefix="localhost_access_log" suffix=".txt"
  pattern="%h %l %u %t &quot;%r&quot; %s %b" />

</Host>
</Engine>
</Service>
</Server>
```

Fig 18. Archivo de configuración de Apache Tomcat.

Se guarda el archivo index.html en la carpeta pública creada y se reinicia el servidor apache tomcat, luego se ejecuta en el puerto cambiado para verificar que sirva.

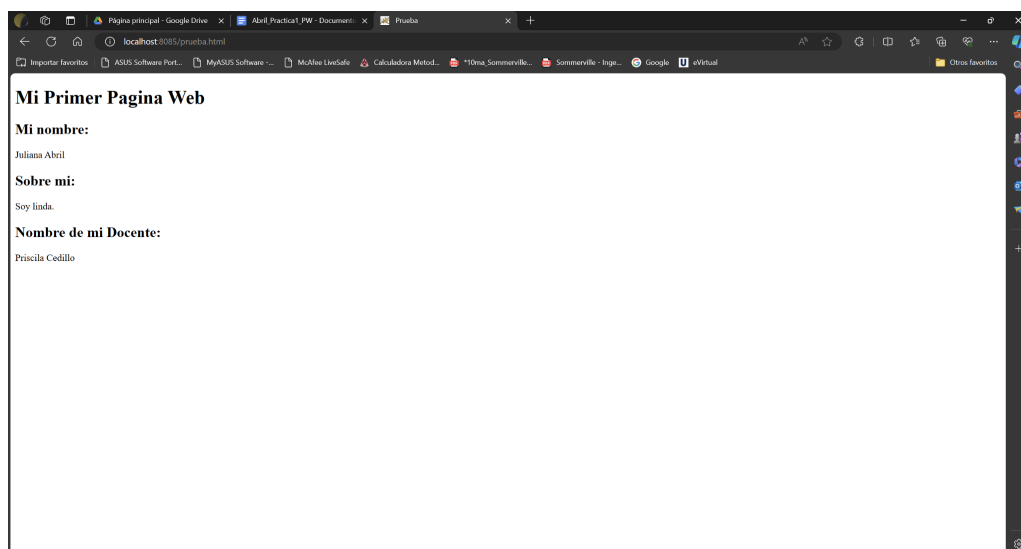


Fig 19. Página de prueba ejecutada.

4. 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 en cada una de ellas. Por ejemplo: En la universidad tuviera una carpeta en

donde esté la página de inicio y una subcarpeta de cada facultad y dentro de ella carpetas de imágenes, css, etc. (Consultar en la bibliografía convenciones básicas de estructuras de carpetas).

Un ejemplo de estructura de archivos para publicación puede ser la siguiente, ya que contiene una carpeta dedicada a las imágenes, css, scripts, plantillas, etc. Y los archivos principales van fuera de las carpetas para que el servidor las pueda desplegar sin problema y es recomendada por la documentación de Tomcat 7.

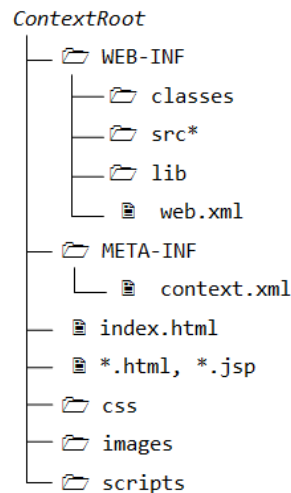


Fig 20. Estructura de archivos obtenida de [9].

Como ejemplo se ha descargado un proyecto de Github en el siguiente enlace [devopsbyhari/JenkinsTomcatExample: This is the Sample project for Jenkins Tomcat example \(github.com\)](https://github.com/devopsbyhari/JenkinsTomcatExample) para poder observar la estructura de carpetas para publicación.

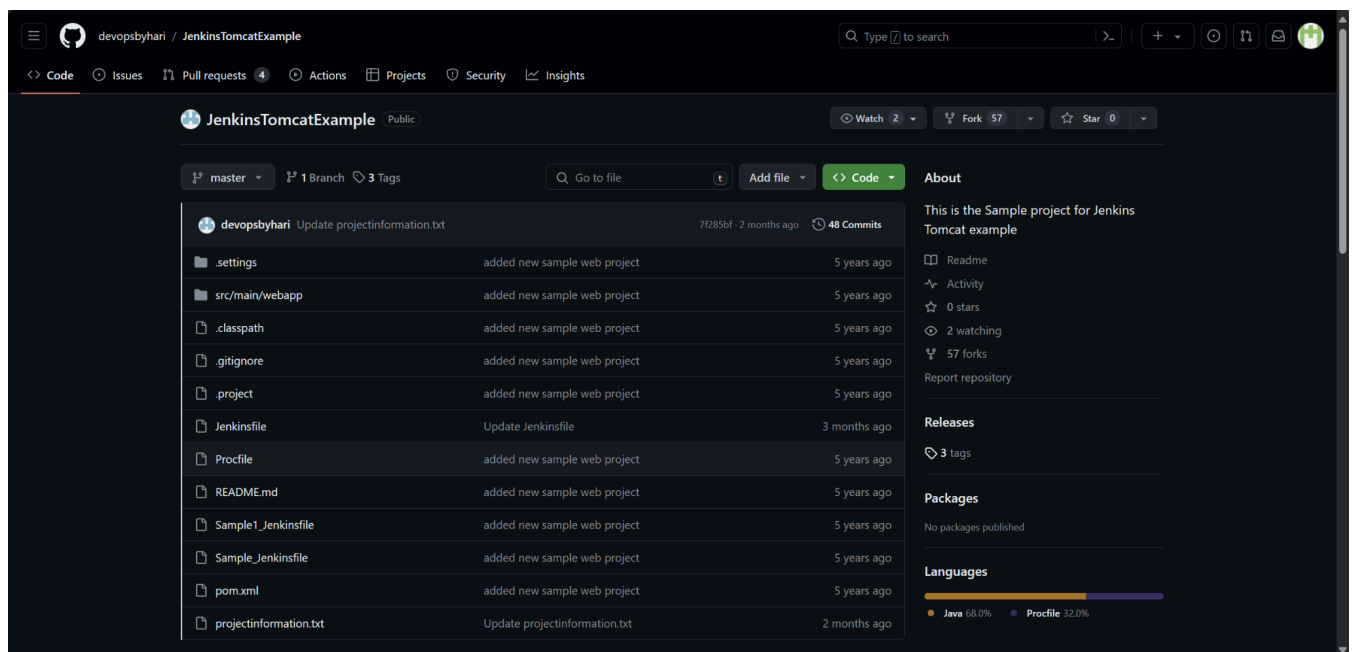


Fig 21. Proyecto de ejemplo.

Y desplegado el proyecto en Apache Tomcat se visualiza de la siguiente manera:

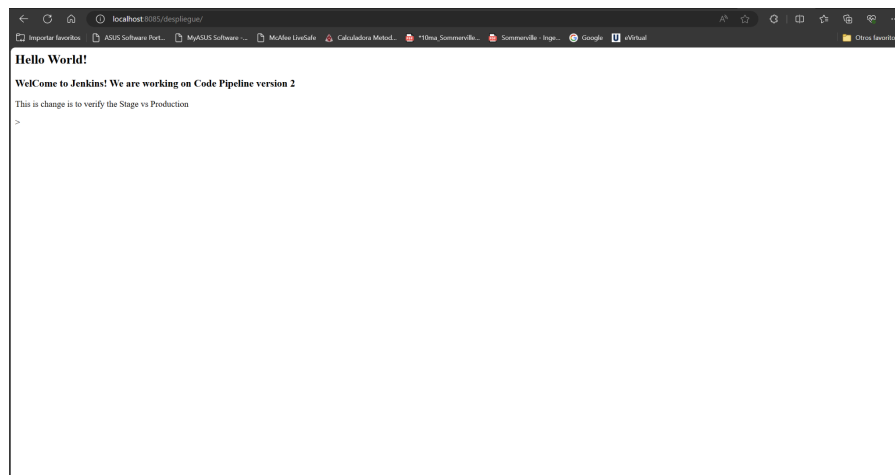


Fig 22. Proyecto desplegado en Apache Tomcat.

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

Para configurar el firewall se debe ingresar a la seguridad de Windows y a configuración avanzada eso se realiza abriendo la configuración del firewall del sistema operativo. Y se debe agregar una regla de entrada para permitir el tráfico en el puerto 8082.

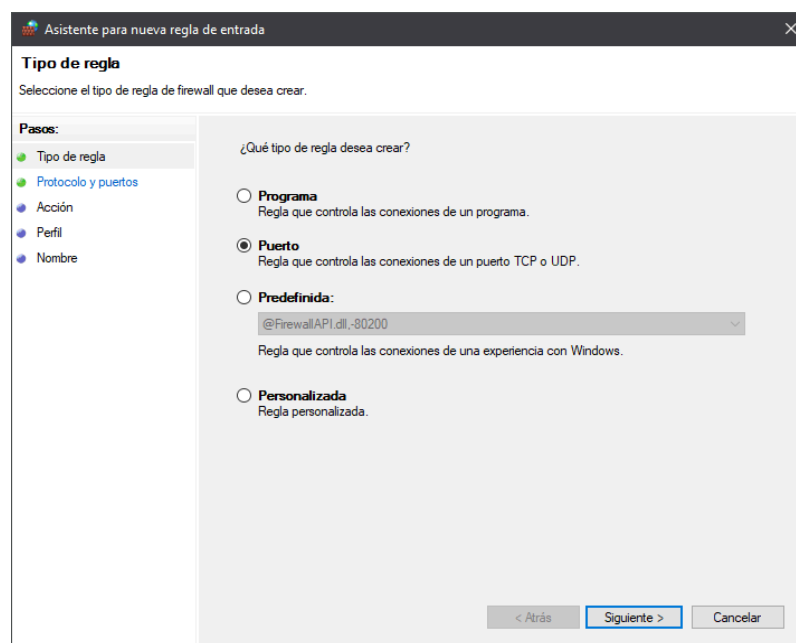


Fig 23. Ventana de configuración de Firewall.

Se configura el tipo de protocolo a usar, tipo de conexión, perfiles con acceso y por último el nombre de la regla de entrada.

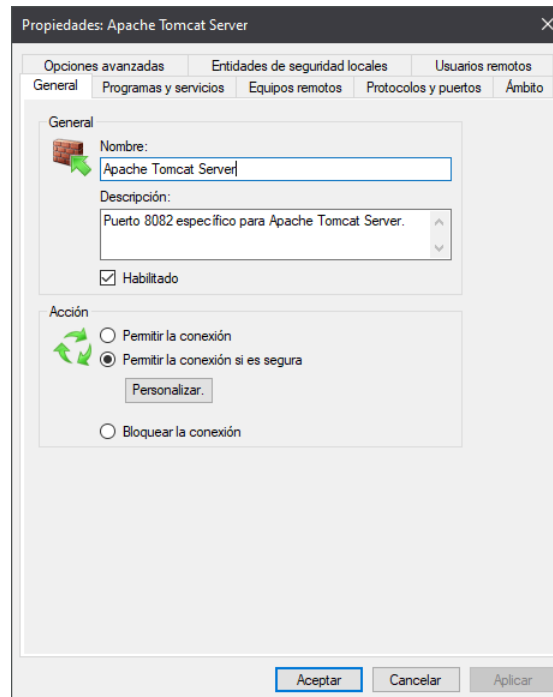


Fig 24. Crear nueva regla de entrada.

b. Servidor Cloud

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

Amazon Web Services (AWS) es la nube más adoptada y completa en el mundo, que ofrece más de 200 servicios integrales de centros de datos a nivel global. Millones de clientes, incluso las empresas emergentes que crecen más rápido, las compañías más grandes y los organismos gubernamentales líderes, están usando AWS para reducir los costos, aumentar su agilidad e innovar de forma más rápida.

AWS ofrece un sistema de pago por uso para los precios de la gran mayoría de los servicios de nube. Con AWS solo paga por los servicios individuales que necesita durante el tiempo que los utilice, sin contratos a largo plazo ni licencias complejas. Los precios de AWS son similares a las tarifas de los servicios de agua y electricidad. Solo paga por lo que consume y, una vez que cancela el servicio, no se aplican costos adicionales ni cuotas de cancelación. [8]

Registrarse en AWS

Seleccionar un plan de soporte

Elija un plan de soporte para su cuenta personal o empresarial. [Compare planes y ejemplos de precio](#). Puede cambiar su plan en cualquier momento desde la consola de administración de AWS.

<input checked="" type="radio"/> Soporte de nivel Basic: gratis <ul style="list-style-type: none">Recomendado para los usuarios nuevos que recién comienzan a utilizar AWSAcceso de autoservicio las 24 horas del día, los 7 días de la semana a los recursos de AWSSolo para problemas de facturación y cuentasAcceso a Personal Health Dashboard y Trusted Advisor 	<input type="radio"/> Soporte Developer: a partir de 29 USD al mes <ul style="list-style-type: none">Recomendado para desarrolladores que experimentan con AWSAcceso por correo electrónico a AWS Support durante el horario laboralTiempos de respuesta de 12 horas (horario laboral) 	<input type="radio"/> Soporte Business: a partir de 100 USD al mes <ul style="list-style-type: none">Recomendado para ejecutar cargas de trabajo de producción en AWSSoporte técnico las 24 horas, los 7 días de la semana por correo electrónico, teléfono y chatTiempos de respuesta de 1 horaConjunto completo de recomendaciones de prácticas de Trusted Advisor 
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------



¿Necesita soporte de nivel Enterprise?

A partir de los 15 000 USD por mes, tendrá tiempos de respuesta de 15 minutos y una experiencia de conserje con un director técnico de cuenta asignado. [Más información](#)

Fig 25. Ventana de registro y soporte de AWS.

Además se comunica al usuario que al momento de registrarse con el soporte gratis obtiene 12 meses de más de 100 servicios de AWS gratis, aunque hay que tener cuidado ya que al momento de tener un uso excesivo de los servicios provocará que estos sean cobrados con tarifas por medio de la tarjeta ingresada al momento del registro de la cuenta.

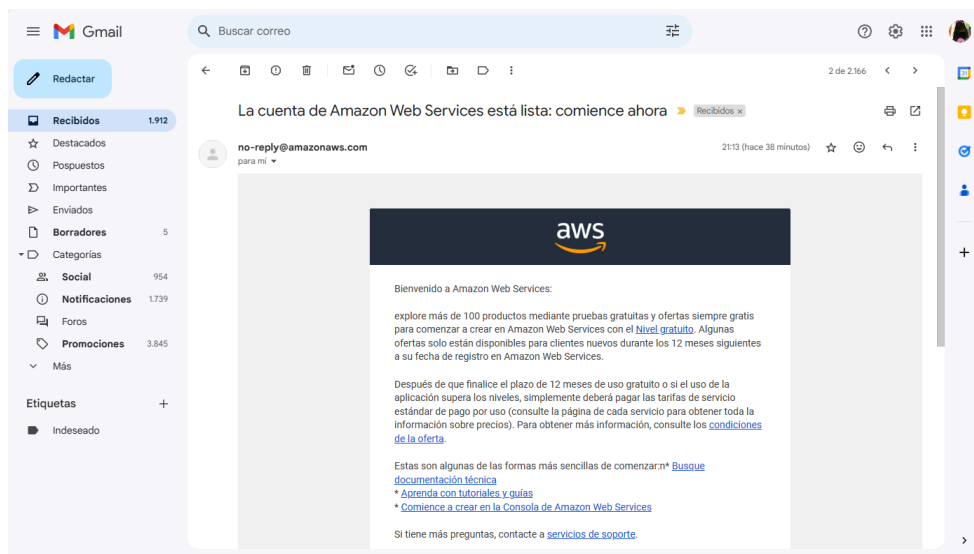


Fig 26. Correo de confirmación de registro de cuenta de AWS.

Los servicios más comunes de AWS son: Amazon EC2 para capacidad informática escalable, Amazon S3 para almacenamiento de objetos, Amazon RDS para bases de datos relacionales, Amazon DynamoDB para bases de datos NoSQL, AWS Lambda para computación sin servidor, Amazon VPC para redes virtuales, etc.

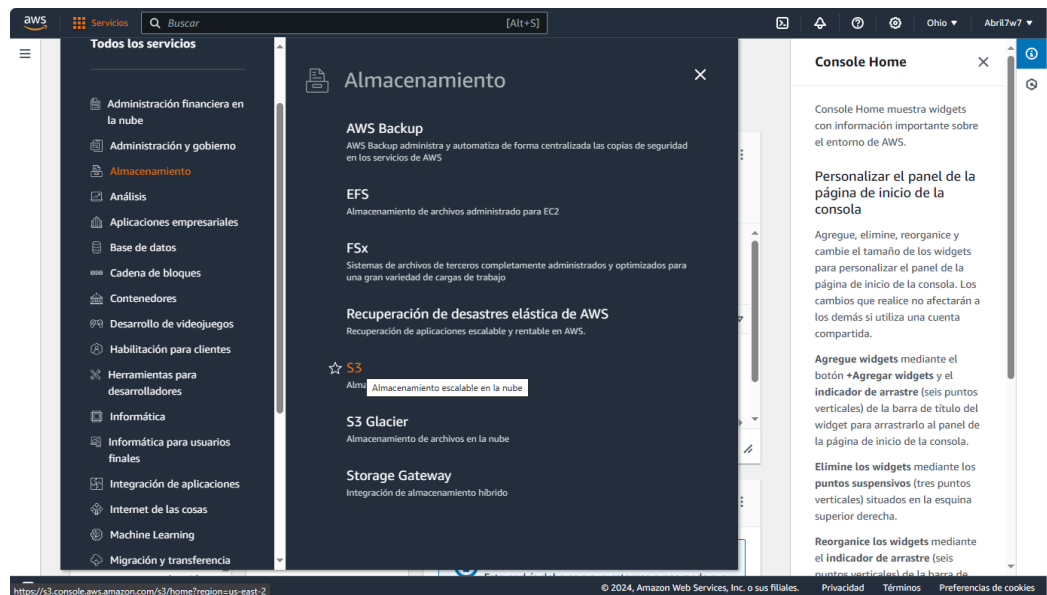


Fig 27. Ventana principal de AWS.

Se ha desplegado una aplicación en AWS como un ejemplo del uso del servicio de almacenamiento S3, para eso se ha creado un bucket con un nombre único, se han cargado los archivos de la aplicación y se lo ha colocado público. Por último se ha habilitado el alojamiento de sitios web estáticos.

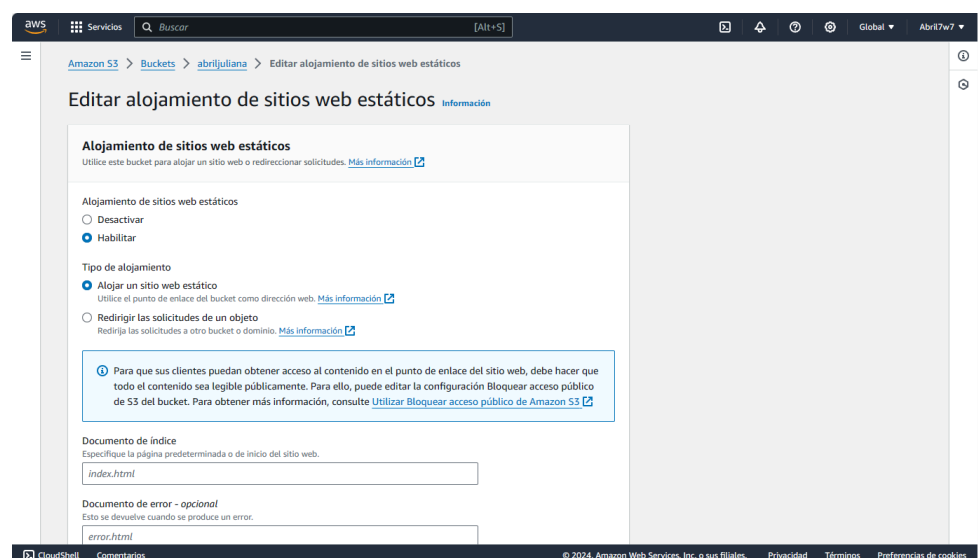


Fig 28. Editar alojamiento de sitios web.

Para observar la aplicación desplegada simplemente se copia la url `http://abriljuliana.s3-website.us-east-2.amazonaws.com/` entregada por AWS y se la copia en

algún navegador de confianza. Este servicio de amazon permite 20.000 solicitudes get y 2000 solicitudes put gratis durante 12 meses.

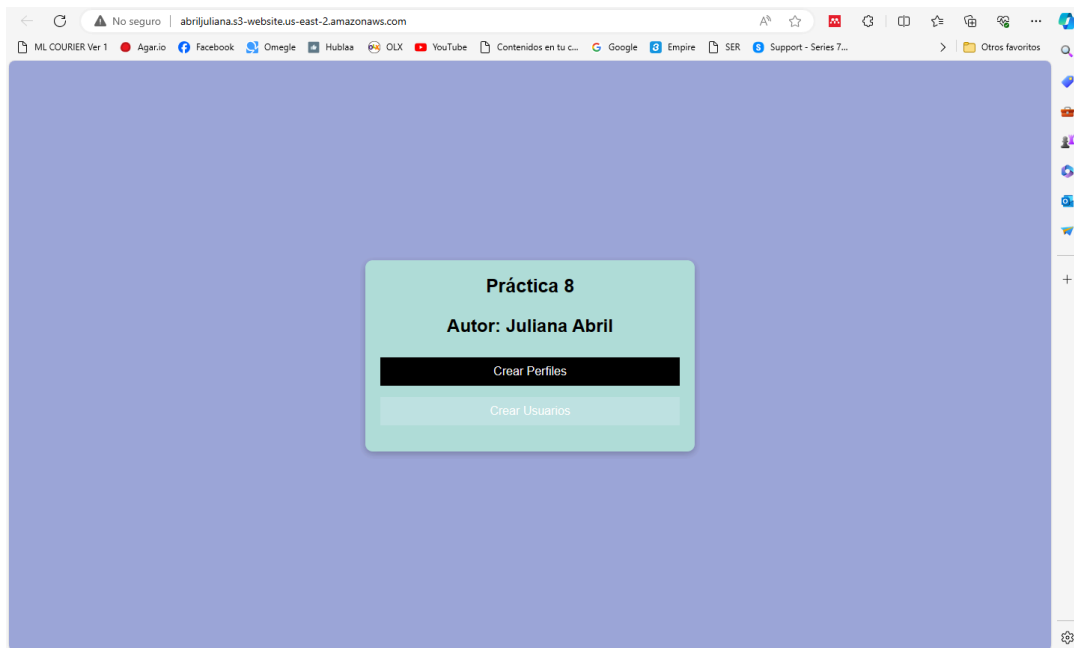


Fig 29. Página de prueba ejecutada.

5. BIBLIOGRAFÍA

- [1] “Servidores web en comparación con servidores de aplicaciones: diferencia entre servidores tecnológicos - AWS”. Amazon Web Services, Inc. Accedido el 25 de marzo de 2024. [En línea]. Disponible: <https://aws.amazon.com/es/compare/the-difference-between-web-server-and-application-server>
- [2] “¿Qué es un servidor web y para qué sirve? - Webempresa”. Webempresa. Accedido el 25 de marzo de 2024. [En línea]. Disponible: <https://www.webempresa.com/hosting/que-es-servidor-web.html>
- [3] Equipo editorial de IONOS. “¿Qué son los servidores de aplicaciones?” IONOS Digital Guide. Accedido el 25 de marzo de 2024. [En línea]. Disponible: <https://www.ionos.es/digitalguide/servidores/know-how/servidor-de-aplicaciones/>
- [4] “¿Qué es un hosting y cómo funciona?” Tutoriales Hostinger. Accedido el 25 de marzo de 2024. [En línea]. Disponible: <https://www.hostinger.es/tutoriales/que-es-un-hosting>
- [5] “¿Qué es cloud computing? | Google Cloud”. Google Cloud. Accedido el 25 de marzo de 2024. [En línea]. Disponible: <https://cloud.google.com/learn/what-is-cloud-computing?hl=es>
- [6] “¿Qué es el despliegue continuo? | IBM”. IBM in Deutschland, Österreich und der Schweiz. Accedido el 25 de marzo de 2024. [En línea]. Disponible:

<https://www.ibm.com/es-es/topics/continuous-deployment#:~:text=El%20despliegue%20cont%20inuo%20es%20una,una%20serie%20de%20pruebas%20predefinidas.>

[7] “La estructura de archivos y carpetas”. Universitat de València. Accedido el 1 de abril de 2024. [En línea]. Disponible: <https://www.uv.es/fragar/html/html1302.html>

[8] “AWS | Cloud Computing - Servicios de informática en la nube”. Amazon Web Services, Inc. Accedido el 28 de marzo de 2024. [En línea]. Disponible: <https://aws.amazon.com/es/>

[9] “Advanced Tutorial on Tomcat 7”. Nanyang Technological University. Accedido el 1 de abril de 2024. [En línea]. Disponible:
https://www3.ntu.edu.sg/home/ehchua/programming/howto/Tomcat_More.html