

Configuración de Servidores

Camila Verónica Granda Salamea

camila.granda@ucuenca.edu.ec

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.

1. MARCO TEÓRICO

Servidor Web

Un servidor web es una plataforma computacional muy potente que resguarda datos para ser consultados por diversos usuarios. Cuenta con un software que entrega la información solicitada por visitantes o miembros de un grupo de trabajo, a través de un dispositivo conectado a la red.

Aunque existen muchos tipos de servidores web, hay algunas características que todos comparten.

- Necesitan soporte físico, donde los servidores web deben tener un hardware, que es el equipo que resguarda la información que será transmitida al ser solicitada por los usuarios.
- Requieren un sistema operativo, para que la información del hardware pueda ser extraída y direccionada a su destino final, es esencial tener un software diseñado para llevar a cabo estas tareas.
- Cuentan con un Protocolo de Transferencia de Hipertexto, también conocido como HTTP. Son instrucciones que permiten la transmisión de información. Los HTTP establecen la comunicación entre navegadores o plataformas de búsqueda con las bases de datos de los servidores web.
- Sirven para la búsqueda de información, los servidores tienen como objetivo responder de manera oportuna a las solicitudes de los usuarios de un sistema.
- Alta capacidad, un servidor web debe ser potente y contar con una gran capacidad de almacenamiento, ya que aloja diferentes sitios, cuentas e interacciones.

Ejemplo de Servidores Web.

1. *Apache HTTP.*

Es un servidor web de código abierto que tiene la ventaja de estar optimizado para sistemas UNIX, Microsoft y MacOS, por lo que es muy versátil para los

desarrolladores informáticos. Es el servidor HTTP más usado en todo el mundo y actualmente aloja el mayor número de sitios web.

2. *Internet Information Services.*

El servidor IIS es el segundo más utilizado a nivel mundial y está diseñado para el sistema operativo de Microsoft, por lo que se adapta más fácilmente a las herramientas de Windows y su paquetería. Este servicio tiene la ventaja de permitir que un equipo de cómputo almacene la información de un sitio web.

3. *Sun Java System Web Server.*

Es un servidor Web creado por Oracle, de alto rendimiento, escalable, versátil y muy seguro. Está enfocado en aplicaciones hechas en Java, posee archivos de configuración consolidados, proxy inverso integrado y muchas otras características.
[1]

Servidor de Aplicaciones

Los servidores de aplicaciones son componentes fundamentales en el mundo de la tecnología y el desarrollo web. Son los encargados de proporcionar servicios y funcionalidades a las aplicaciones, lo que permite que estas puedan comunicarse con otras aplicaciones, bases de datos, servidores web y realizar tareas específicas para el correcto funcionamiento de los sistemas. Cuando una aplicación necesita acceder a una base de datos, procesar información o realizar tareas complejas, se requiere un servidor de aplicaciones para gestionar estas acciones. Estos servidores actúan como intermediarios entre el usuario y la aplicación, encargándose de ejecutar la lógica de negocio, procesar datos y proveer respuestas adecuadas. Los servidores de aplicaciones se basan en la tecnología de Internet Information Services (IIS) o en otras soluciones de servidor web, que les permiten recibir y procesar solicitudes HTTP/HTTPS de los usuarios y proporcionar respuestas en forma de páginas web o servicios.

Ejemplo de Servidores de Aplicaciones.

1. *Apache Tomcat.*

Un servidor de aplicaciones de código abierto muy utilizado para desplegar aplicaciones web basadas en Java.

2. *Resin.*

Resin es un servidor de aplicaciones Java de código abierto rápido y confiable para aplicaciones web que abarca cientos de servidores Originado como un motor Servlet y JSP, Resin es compatible con Web Profile, CDI y EJB Resin incluye un administrador

de transacciones JTA de alto rendimiento, un proveedor JMS, clústeres, agrupación de conexiones y una consola de administración Resin es utilizado por compañías como CNET, Motorola y Salesforcecom.

3. *GlassFish*.

Es un servidor de aplicaciones de código abierto y plataforma Java EE (Enterprise Edition) desarrollado por Oracle Corporation. Como servidor de aplicaciones Java, GlassFish proporciona un entorno de tiempo de ejecución para ejecutar aplicaciones Java empresariales y servicios web. [2]

Hosting

Un hosting o alojamiento web es un término genérico que se utiliza para denominar a un tipo de servidor que aloja sitios web y/o datos relacionados, aplicaciones y servicios. Dicho de un modo más sencillo, es lo que permite que tu página web o blog sea visto por cualquier usuario con conexión a Internet. Es un servicio que permite a los usuarios publicar una página web en internet.

Tipos de Hosting

- *Hosting gratuito*: de baja calidad, pero alternativa viable para los sitios sin fines comerciales ni profesionales.
- *Hosting compartido*: se trata de un espacio reservado, dentro de un servidor compartido entre varios usuarios, es decir, que puede alojar varias webs a la vez.
- *Hosting VPS*: cada sitio web se almacena en un servidor potente dividido en compartimentos virtuales. Es decir, el cliente dispone de sus propios recursos RAM, CPU, etc.
- *Hosting Dedicado*: un servidor completo para el usuario. Es costoso pero permite máxima flexibilidad y velocidad.

¿Cómo alojar un hosting?

1. Elegir la empresa de hosting con la que trabajar. Dependiendo de las necesidades que se tenga se puede escoger entre las diferentes empresas del mercado.
2. Seleccionar el plan de hosting. Una vez se haya decidido la empresa con la que trabajar es importante seleccionar que tipo de alojamiento vamos a utilizar: compartido, VPS, dedicado.
3. Cambiar las DNS. Es necesario cambiar las DNS del dominio para que la web funcione.
4. Subir la web. A través del servidor con cPanel o FileZilla se publica la web en el hosting. [3]

Cloud Computing

La computación en la nube (cloud computing) es una tecnología que permite acceso remoto a softwares, almacenamiento de archivos y procesamiento de datos por medio de Internet, siendo así, una alternativa a la ejecución en una computadora personal o servidor local. En el modelo de nube, no hay necesidad de instalar aplicaciones localmente en computadoras. La computación en la nube ofrece a los individuos y a las empresas la capacidad de un pool de recursos de computación con buen mantenimiento, seguro, de fácil acceso y bajo demanda.

La computación en la nube utiliza una capa de red para conectar los dispositivos de punto periférico de los usuarios, como computadoras, smartphones y accesorios portátiles, a recursos centralizados en el data center. Antes de la computación en la nube, la ejecución confiable de software por las empresas que ofrecían servicios solo era posible si ellas podían también pagar por el mantenimiento de la infraestructura de los servidores necesarios. Además, el software tradicional exigía, por lo general, un equipo completo de profesionales de TI, interno o externo, para lidiar con el inevitable conjunto de errores, desafíos de servicios y actualizaciones. El concepto de computación en la nube está libre de todos esos problemas y requisitos anticuados.

Modelo de Servicios en la Nube.

1. *SaaS*

El modelo SaaS – Software como Servicio de computación en la nube – se centra en facilitar el acceso a la aplicación de software para el usuario por medio de una interfaz de navegador o de programa. Con este modelo, la red subyacente, el sistema operacional y los recursos funcionan tras bastidores.

2. *PaaS*

El modelo PaaS – Plataforma como Servicio – puede aprovechar los beneficios de la computación en la nube mientras mantiene la libertad de desarrollar aplicaciones personalizadas del software. Los usuarios pueden acceder a PaaS de la misma manera que se hace con el SaaS. El proveedor es responsable por el mantenimiento del sistema operacional, de la red, de los servidores y de la seguridad.

3. *IaaS*

El modelo IaaS – Infraestructura como Servicio – va un paso adelante en la abstracción, proporcionándoles a las organizaciones la capacidad de aprovechar recursos brutos del servidor mientras el resto de la administración de la plataforma y del software es de responsabilidad de la empresa. Eso permite mayor capacidad sin necesidad de preocuparse por requisitos de hardware. [4]

Despliegue Continuo

La implementación continua (CD) es un proceso de publicación de software que utiliza pruebas automatizadas para verificar si los cambios en un código base son correctos y estables para la implementación inmediata y autónoma en un entorno de producción. El ciclo de publicación del software ha evolucionado con el tiempo. El proceso heredado, consistente en mover código de una máquina a otra y en verificar si funciona como se esperaba, era un proceso propenso a errores y a consumir muchos recursos. Ahora hay herramientas que pueden automatizar este proceso de implementación, con lo que las organizaciones de ingeniería se pueden centrar en las necesidades de negocio principales y librarse de la carga de trabajo relacionada con las infraestructuras.

Diferencias entre implementación Continua y Entrega Continua

La diferencia principal entre entrega continua e implementación continua radica en el alcance y el enfoque de los procesos involucrados. La entrega continua se centra en la automatización de las etapas de desarrollo, prueba y despliegue del software, lo que permite liberaciones frecuentes y confiables de código listo para producción. Por otro lado, la implementación continua se enfoca específicamente en la fase de despliegue, garantizando que cualquier cambio validado en el código pueda implementarse automáticamente en el entorno de producción sin intervención manual. En resumen, mientras que la entrega continua abarca todo el ciclo de vida del desarrollo de software, desde la escritura del código hasta el despliegue, la implementación continua se concentra únicamente en la fase de implementación automatizada del código validado. [5]

2. EJERCICIOS

Servidores locales

1. Cambiar los puertos para que funcionen en un puerto desde el 8080 hasta el 8089
 - Apache

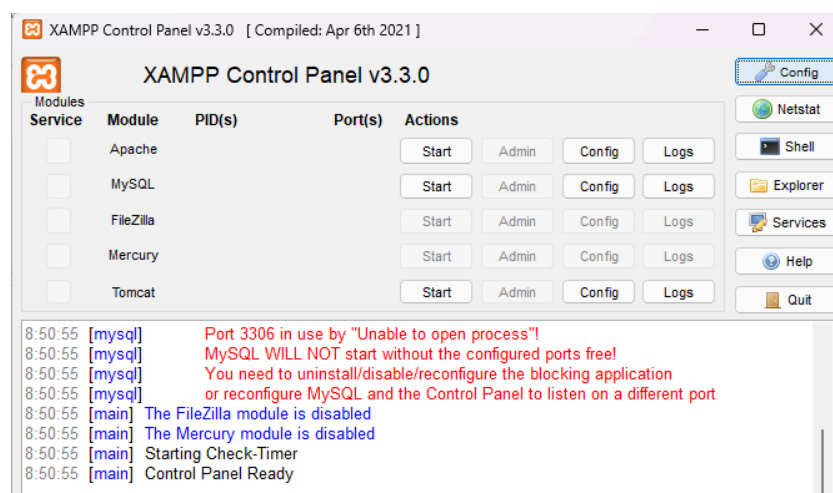


Fig 1. Panel de Control XAMPP

Para la configuración de Apache, nos ubicamos en el botón de Config.

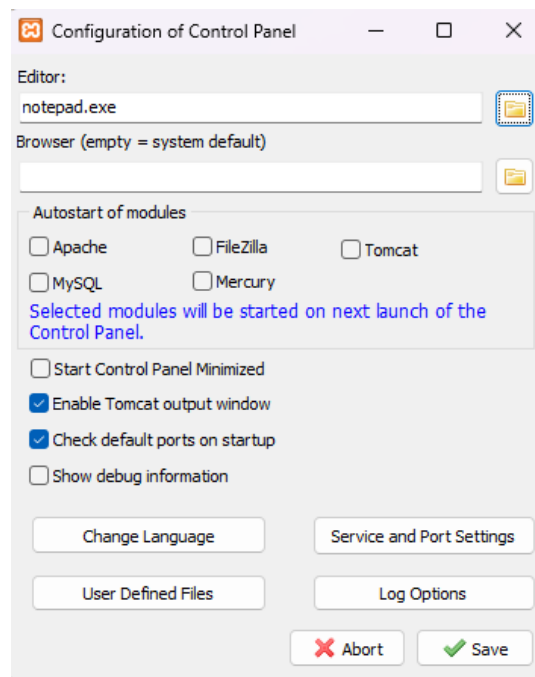


Fig 2. Configuración del Panel de Control XAMPP

Nos vamos a la opción de Service and Port Settings.

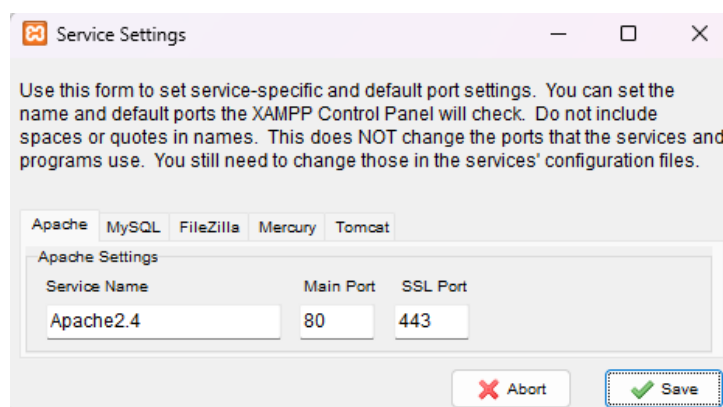


Fig 3. Configuración de Servicios de Apache XAMPP

Cambiamos del puerto 80 al puerto 8081.

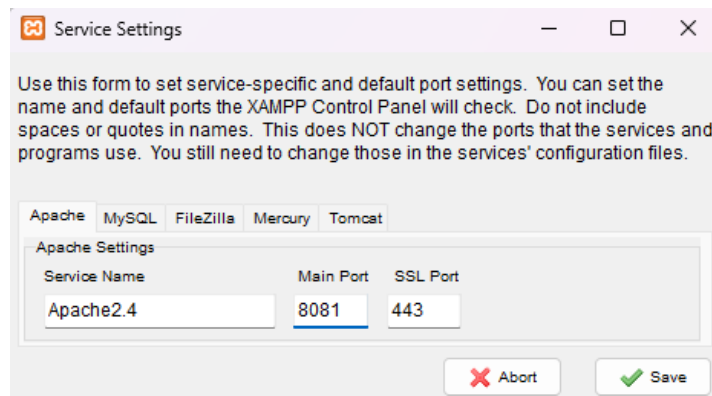


Fig 4. Configuración de Servicios de Apache XAMPP

Verificamos que en el apartado de config de Apache se haya cambiado el puerto en el archivo de httpd.conf:

```

Archivo  Editar  Ver

#
# 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 8081

#
# Dynamic Shared Object (DSO) Support
#
# To be able to use the functionality of a module which was built as a DSO you
# have to place corresponding 'LoadModule' lines at this location so the
# directives contained in it are actually available before they are used.

```

Fig 5. Archivo de configuración de Servicios de Apache XAMPP

Le ponemos start y después Admin para verificar que no exista ningún error y se realice la conexión.

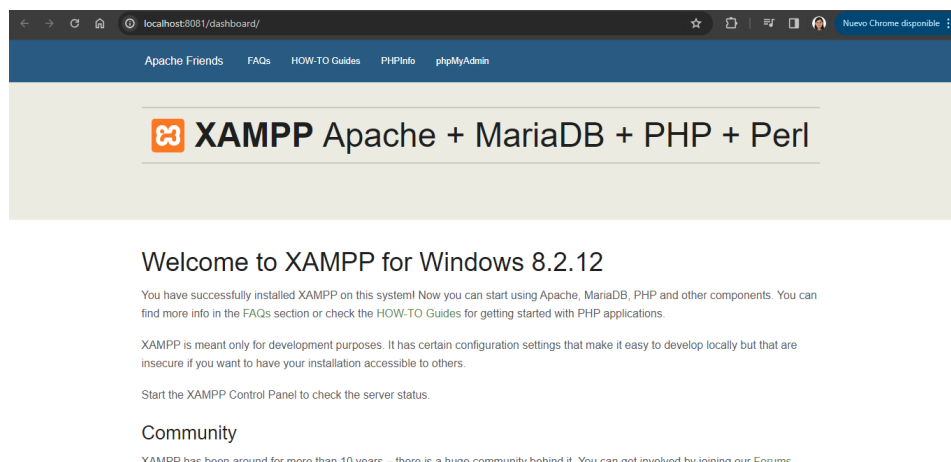
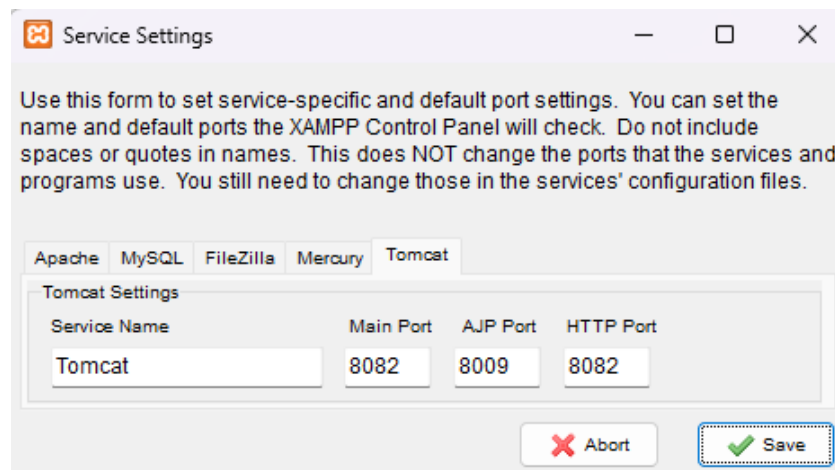


Fig 6. Página Web XAMPP

Ahora para configurar Tomcat, seguimos el mismo proceso que se hizo para Apache y nos vamos al apartado de Tomcat, para cambiar su puerto, y cambiamos al puerto 8082 y ponemos guardar.



Service Settings

Use this form to set service-specific and default port settings. You can set the name and default ports the XAMPP Control Panel will check. Do not include spaces or quotes in names. This does NOT change the ports that the services and programs use. You still need to change those in the services' configuration files.

Apache MySQL FileZilla Mercury Tomcat

Tomcat Settings

Service Name	Main Port	AJP Port	HTTP Port
Tomcat	8082	8009	8082

Abort Save

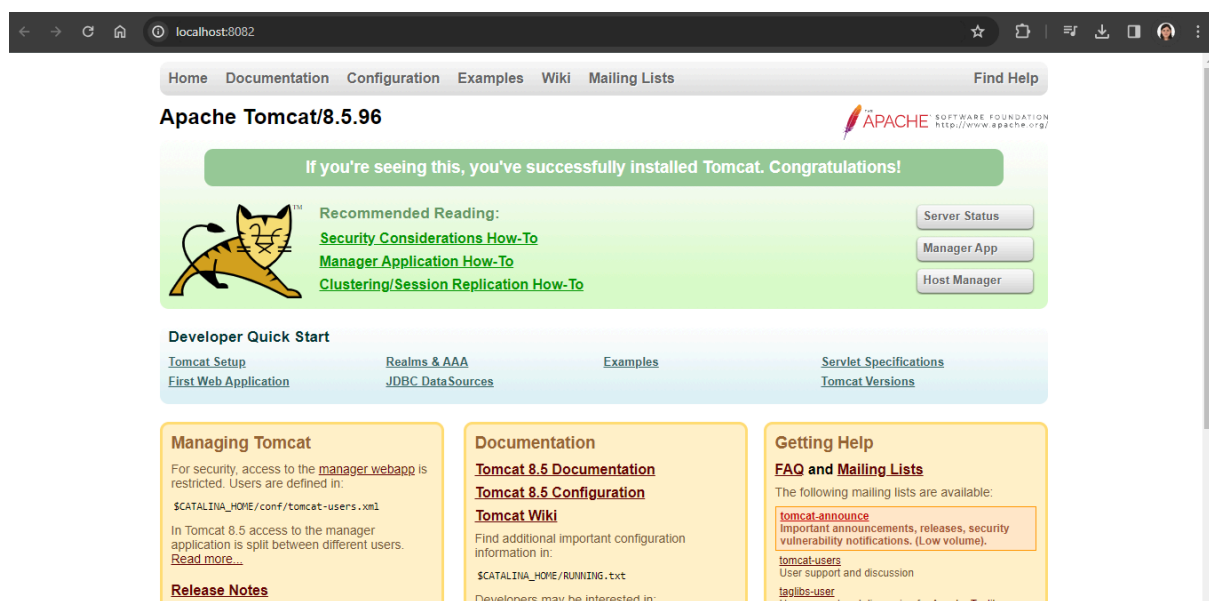
Fig 7. Configuración de Servicios de Tomcat XAMPP

Verificamos que se haya cambiado en server.xml:

```
-->  
<Connector port="8082" protocol="HTTP/1.1"  
    connectionTimeout="20000"  
    redirectPort="8443"  
    maxParameterCount="1000"  
    />  
<!-- A "Connector" using the shared thread pool-->
```

Fig 8. Archivo de configuración de Servicios de Tomcat XAMPP


Le ponemos start y después admin para observar que esté corriendo normal:



Home Documentation Configuration Examples Wiki Mailing Lists Find Help

Apache Tomcat/8.5.96

If you're seeing this, you've successfully installed Tomcat. Congratulations!

 Recommended Reading:
[Security Considerations How-To](#)
[Manager Application How-To](#)
[Clustering/Session Replication How-To](#)

Server Status
Manager App
Host Manager

Developer Quick Start
[Tomcat Setup](#) [Realms & AAA](#) [Examples](#) [Servlet Specifications](#)
[First Web Application](#) [JDBC DataSources](#) [Tomcat Versions](#)

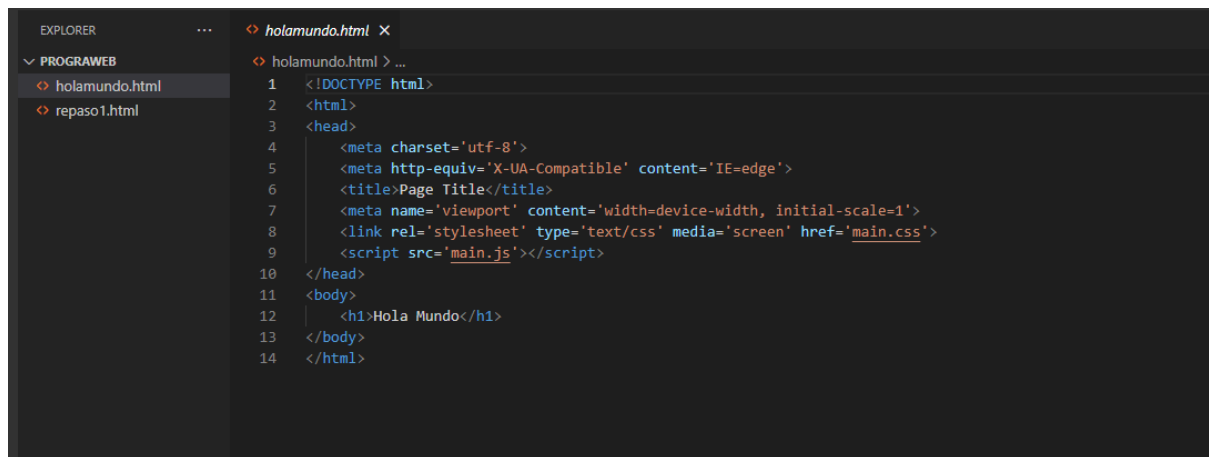
Managing Tomcat
For security, access to the `manager` webapp is restricted. Users are defined in:
`$CATALINA_HOME/conf/tomcat-users.xml`
In Tomcat 8.5 access to the manager application is split between different users.
[Read more...](#)
[Release Notes](#)

Documentation
[Tomcat 8.5 Documentation](#)
[Tomcat 8.5 Configuration](#)
[Tomcat Wiki](#)
Find additional important configuration information in:
`$CATALINA_HOME/RUNNING.txt`
Developers may be interested in:

Getting Help
FAQ and Mailing Lists
The following mailing lists are available:
[tomcat-announce](#)
Important announcements, releases, security vulnerability notifications. (Low volume).
[tomcat-users](#)
User support and discussion
[tomcat-dev](#)
Developer support and discussion for Apache Tomcat

Fig 9. Página Web Apache Tomcat

Para desplegar una página web en cada servidor local, primero creamos un hola mundo en html para después desplegarlo con el servidor web de Apache.



```
EXPLORER
...
holamundo.html X
holamundo.html > ...
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset='utf-8'>
5   <meta http-equiv='X-UA-Compatible' content='IE=edge'>
6   <title>Page Title</title>
7   <meta name='viewport' content='width=device-width, initial-scale=1'>
8   <link rel='stylesheet' type='text/css' media='screen' href='main.css'>
9   <script src='main.js'></script>
10 </head>
11 <body>
12   <h1>Hola Mundo</h1>
13 </body>
14 </html>
```

Fig 10. Archivo .html

Después verificamos que esté guardado en la siguiente carpeta:

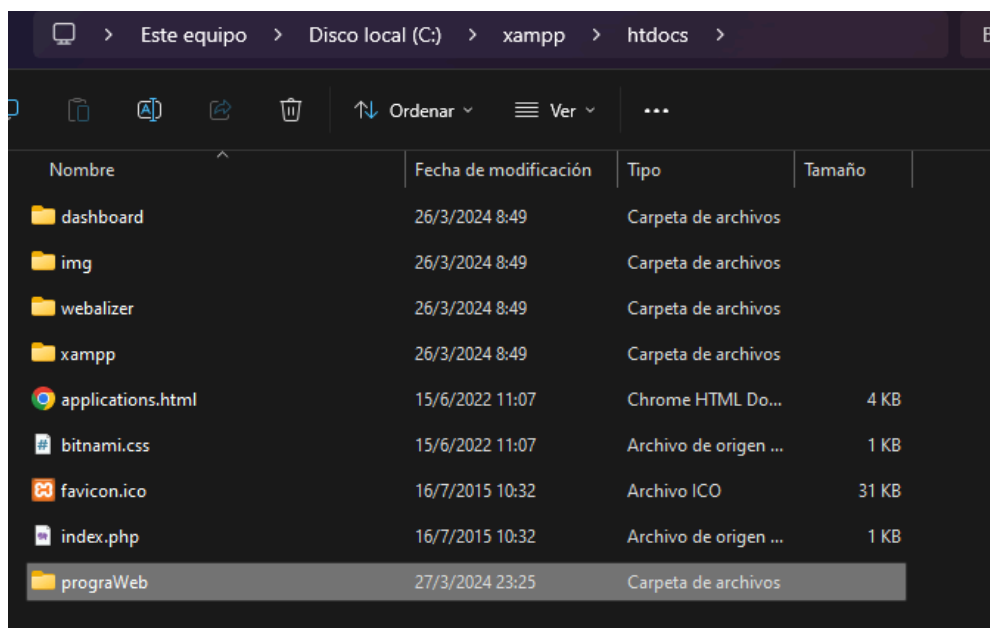


Fig 11. Carpeta htdocs de XAMPP

Esto es necesario debido a que con ello nos permite realizar el siguiente paso:



Fig 12. Prueba de despliegue de página Web Apache

Ahora para tomcat, nos movemos a otra carpeta, donde colocamos la misma carpeta:

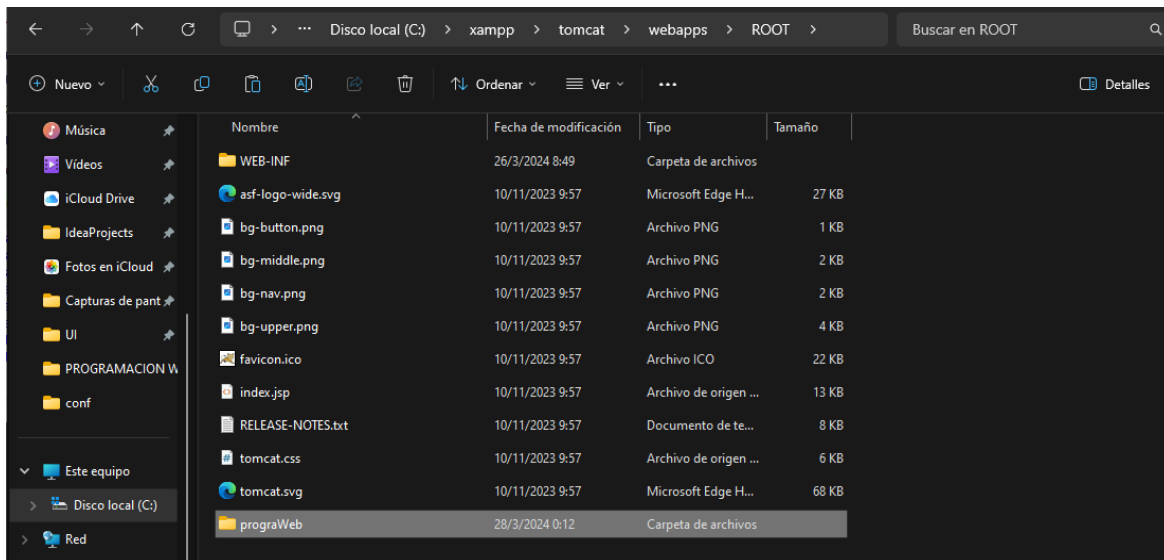


Fig 13. Carpeta ROOT de Tomcat

Luego de ello verificamos en el navegador:



Fig 14. Prueba de despliegue de página Web Tomcat

Se puede observar que se desplegaron correctamente con Apache y Tomcat.

A continuación, se crea una carpeta que se llama “Carpeta Pública” donde va ser nuestra carpeta de publicación.

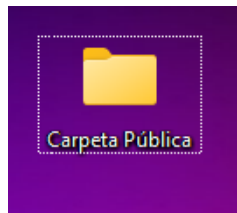


Fig 15. Creación Carpeta Pública

Luego de ello, cambiamos la dirección de donde se está tomando los .html para que nos tome desde la carpeta pública, para ello nos dirigimos al xampp y nos dirigimos al botón de config primero de Apache, donde se cambiará la ubicación que viene por defecto por donde se encuentra la carpeta que se creó.

```
#
DocumentRoot "C:\Users\Camila Granda\Desktop\Carpeta Pública"
<Directory "C:\Users\Camila Granda\Desktop\Carpeta Pública">
    #
    # Possible values for the Options directive are "None", "All",
```

Fig 16. Modificación de ubicación del archivo httpd.conf

Ahora podemos desplegar desde la nueva dirección que fue dada. Ahora se comprobará que se hace correctamente.

Así mismo se cambia en server.xml para desplegar con Tomcat.

```
<!--<Host name="localhost" appBase="webapps" -->
<Host name="localhost" appBase="Carpeta Pública"
    unpackWARs="true" autoDeploy="true">
    <Context path="" docBase="/C:/Users/Camila Granda/Desktop/Carpeta Pública">
        <WatchedResource>WEB-INF/web.xml</WatchedResource>
    </Context>
```

Fig 17. Modificación de ubicación del archivo server.xml

A continuación buscamos una estructura recomendada de un árbol de archivos para publicación de nuestra página web .

Basic File Structure (Websites)

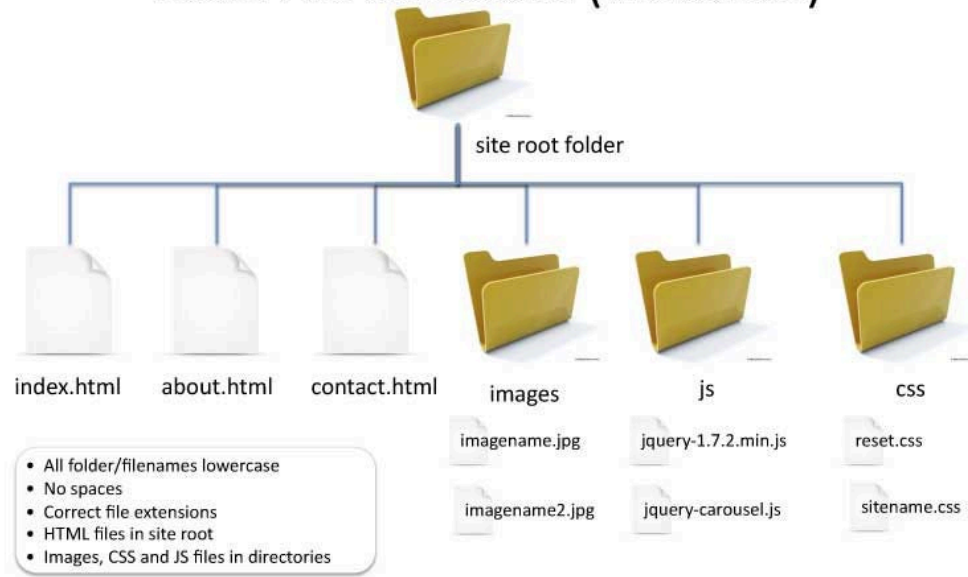


Fig 18. Estructura de Archivos para Páginas Web

Luego de ello, probamos con una aplicación web, en este caso descargamos un proyecto de github <https://github.com/C-Alvarez98/Proyecto-Electro> :

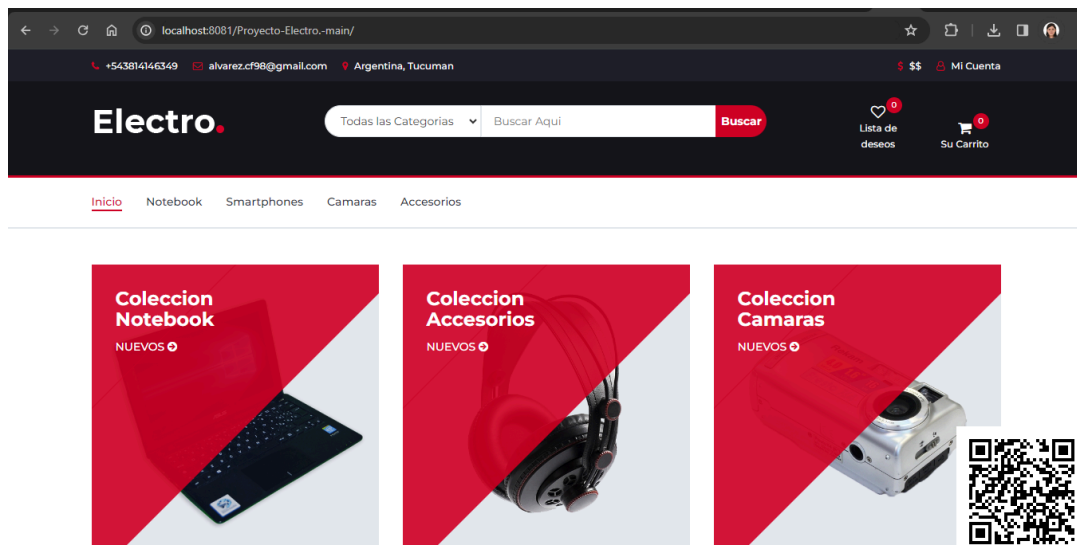


Fig 19. Proyecto de Github

Donde la estructura de este proyecto es con la estructura mencionada anteriormente.

css	28/3/2024 10:18	Carpeta de archivos	
fonts	28/3/2024 10:18	Carpeta de archivos	
img	28/3/2024 10:18	Carpeta de archivos	
js	28/3/2024 10:18	Carpeta de archivos	
checkout(Revisa).html	28/3/2024 10:18	Chrome HTML Do...	14 KB
index.html	28/3/2024 10:18	Chrome HTML Do...	36 KB
LICENSE	28/3/2024 10:18	Archivo	2 KB
product_AriculaGamer.html	28/3/2024 10:18	Chrome HTML Do...	18 KB
product_Ariculasony.html	28/3/2024 10:18	Chrome HTML Do...	18 KB
product_camara_nikon.html	28/3/2024 10:18	Chrome HTML Do...	18 KB
product_pc1_msi.html	28/3/2024 10:18	Chrome HTML Do...	18 KB
product_pc2_hp.html	28/3/2024 10:18	Chrome HTML Do...	13 KB
product_pc3_asus.html	28/3/2024 10:18	Chrome HTML Do...	13 KB
product_pc4_bangho.html	28/3/2024 10:18	Chrome HTML Do...	12 KB
product_s6-edge.html	28/3/2024 10:18	Chrome HTML Do...	13 KB
product_tablet-sony.html	28/3/2024 10:18	Chrome HTML Do...	12 KB
README.md	28/3/2024 10:18	Archivo de origen ...	1 KB

Fig 20. Creación de la estructura de carpetas en nuestro computador

Para configurar el firewall para que sirva exclusivamente en el puerto que se quiera indicar, nos dirigimos al panel de control a sistemas y seguridad:

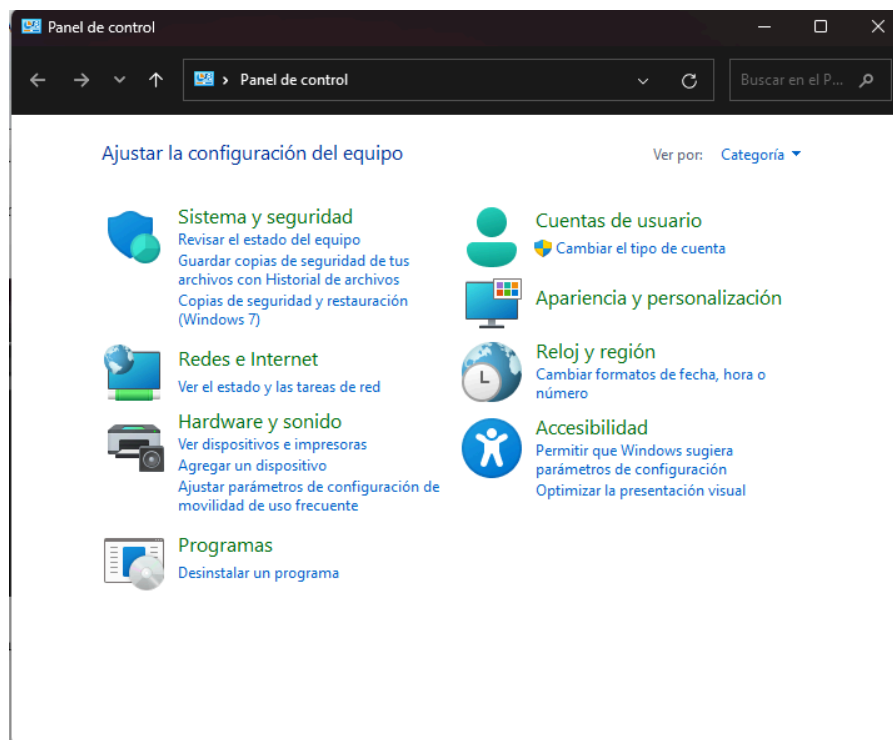


Fig 21. Panel de Control de Windows

Nos dirigimos al apartado de Firewall de Windows Defender:



Fig 22. Interfaz de Sistema y Seguridad de Windows

Nos dirigimos a configuración avanzada:



Fig 23. Interfaz de Firewall de Windows Defender

Luego de ello nos vamos a reglas de entrada:

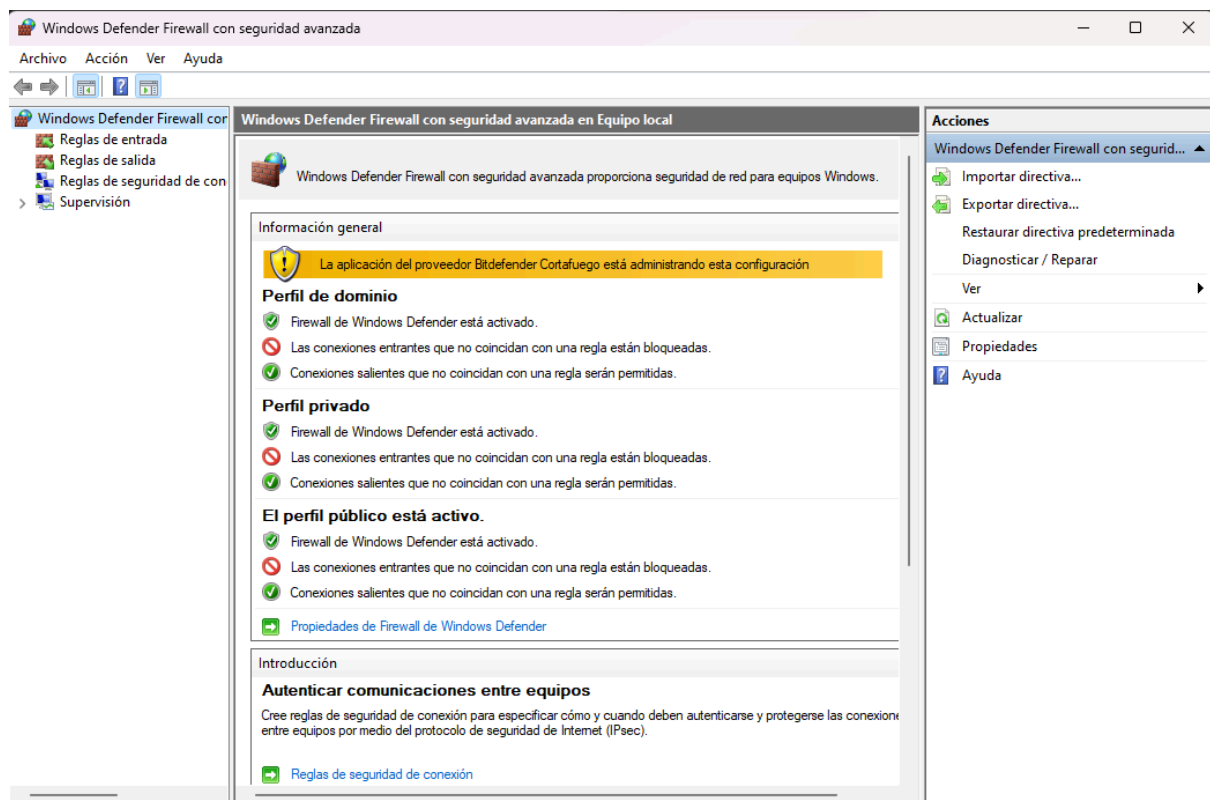


Fig 24. Interfaz de seguridad avanzada de Windows Defender Firewall

Colocamos nueva regla:

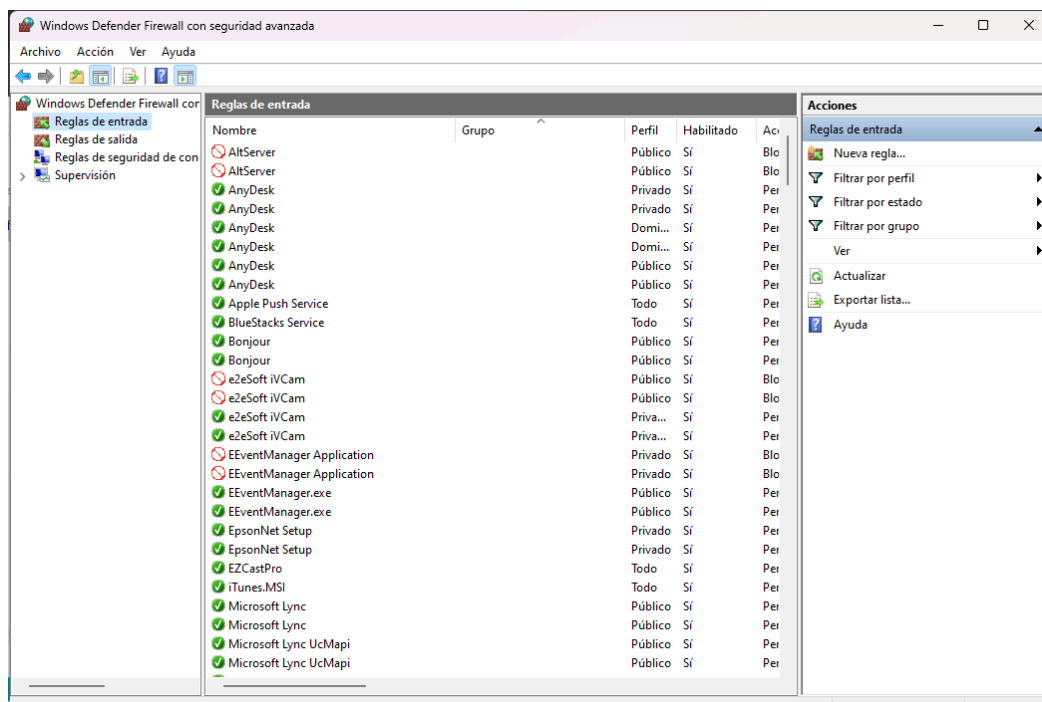


Fig 25. Creación de nuevas reglas de entrada

Elegimos la opción Puerto y siguiente:

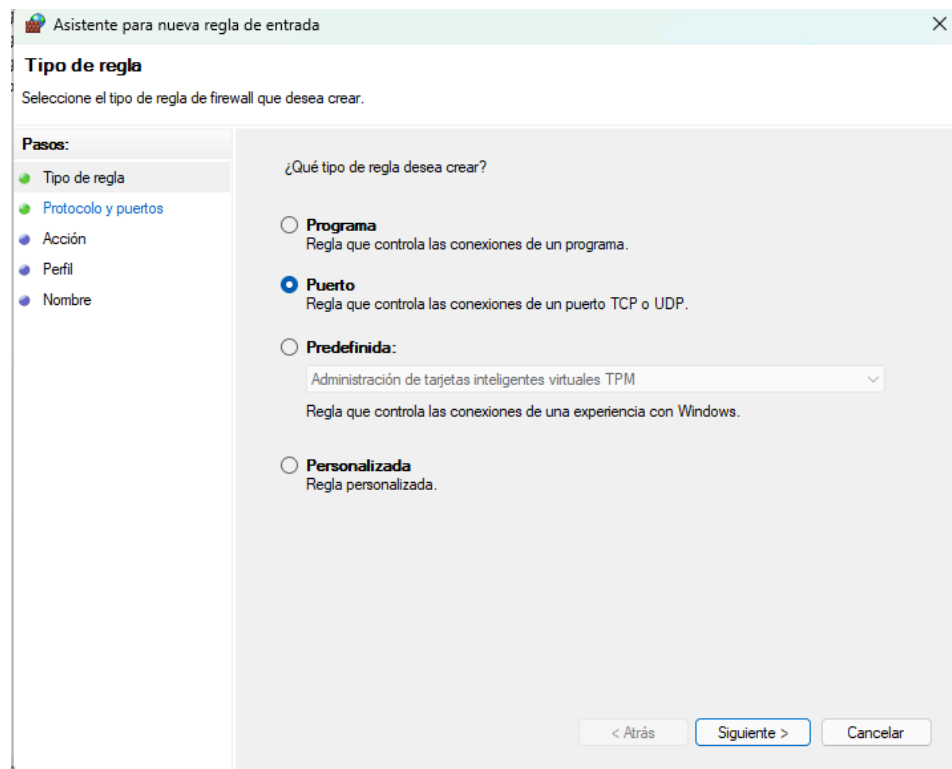


Fig 26. Definición del tipo de regla que se quiere crear

Definimos en qué protocolo se aplica la regla ya sea para TCP o UDP, primero definimos para el puerto que está utilizando Apache:

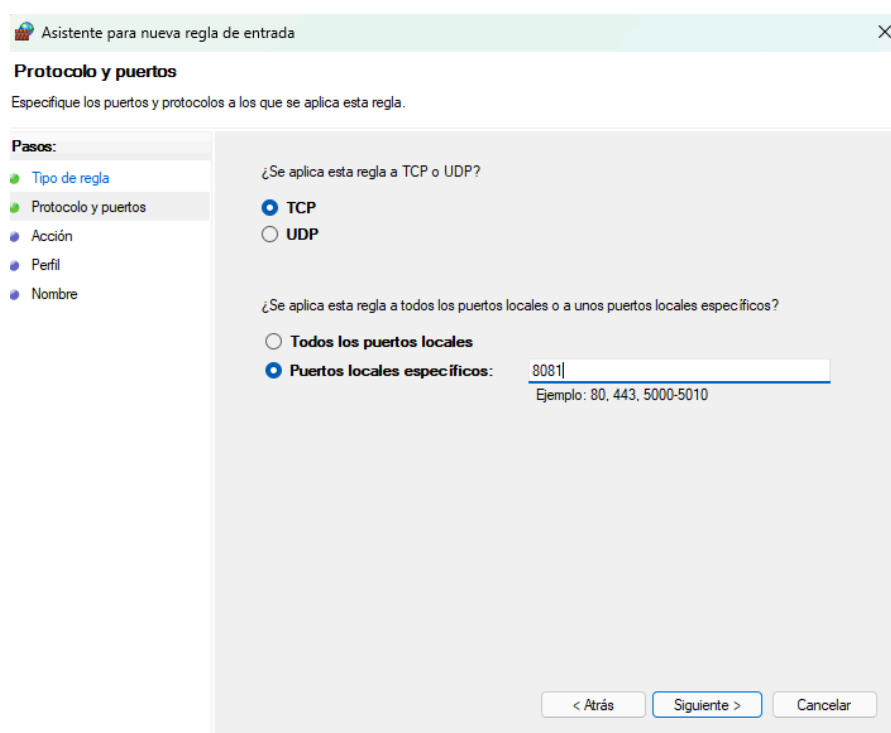


Fig 27. Especificación de puertos donde se quiere aplicar la regla

Luego de ello, permitimos la conexión y ponemos siguiente:

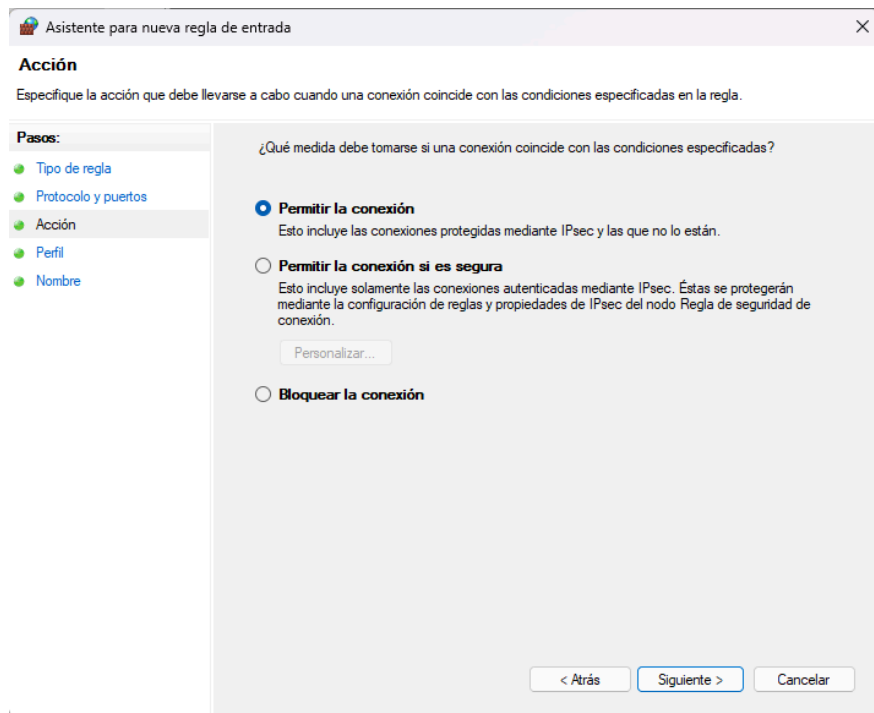


Fig 28. Permiso de conexión

Se pone siguiente:

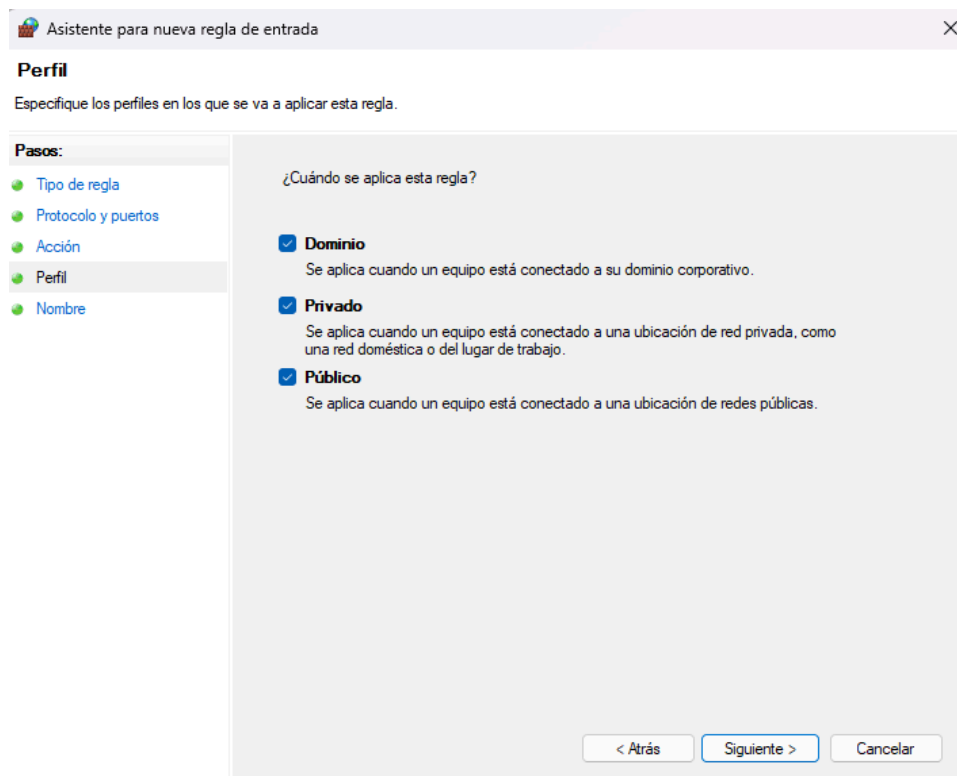


Fig 29. Indicaciones de donde se aplica la regla

Definimos un nombre y ponemos finalizar:

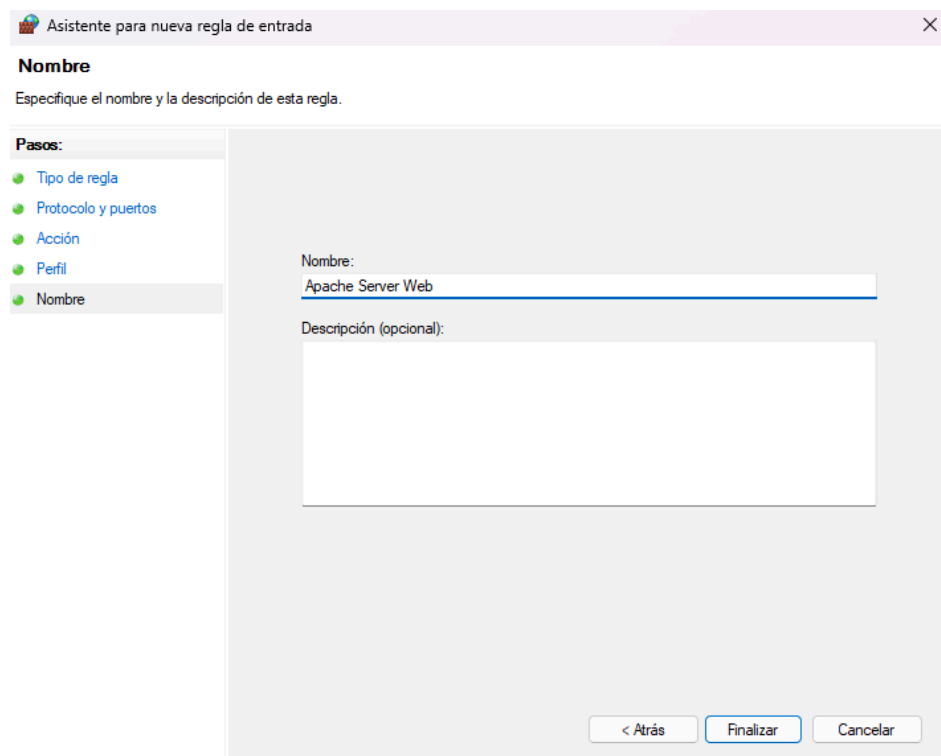
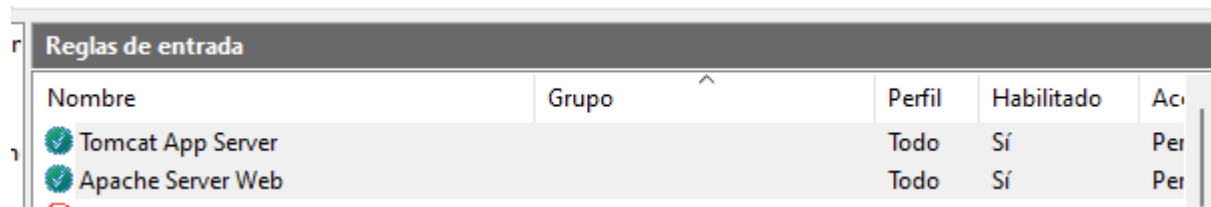


Fig 30. Definición de la nueva regla

Lo mismo se realiza para abrir un puerto al servidor de Tomcat, teniendo como resultado lo siguiente:



Nombre	Grupo	Perfil	Habilitado	Acción
Tomcat App Server		Todo	Sí	Per
Apache Server Web		Todo	Sí	Per

Fig 31. Configuración del firewall para que sirva exclusivamente en el puerto

Servidor Cloud

- Google Cloud

Google Cloud ofrece una variedad de opciones de prueba gratuita de 90 días para Spanner con 10 GB de almacenamiento y 30 días para Looker y niveles gratuitos para explorar sus servicios. Los clientes pueden acceder a almacenamiento sin costo, conectarse a diversas bases de datos y ejecutar cargas de trabajo en la nube con un crédito de \$300. Además tiene más de 20 productos gratuitos para todos los clientes, donde obtienen experiencia práctica gratuita con productos populares, incluidos Compute Engine y Cloud Storage, hasta alcanzar los límites mensuales. [6]

Google Cloud proporciona una amplia variedad de servicios y productos avanzados que pueden beneficiar a los usuarios y al avance tecnológico en general. A través de opciones de prueba gratuita y niveles gratuitos, se puede explorar y familiarizarse con las plataformas de Google Cloud sin costo inicial, lo que fomenta la innovación y el aprendizaje. Google Cloud ofrece acceso a tecnologías avanzadas lo que permite que exista un impulso en el desarrollo de proyectos en diversos campos.

REFERENCIAS BIBLIOGRÁFICAS

[1] M. Coppola. “Qué es un servidor web, para qué sirve, cómo funciona y ejemplos”. Blog de HubSpot | Marketing, Ventas, Servicio al Cliente y Sitio Web. Accedido el 25 de marzo de 2024. [En línea]. Disponible: <https://blog.hubspot.es/website/que-es-servidor-web>

[2] A. Casero. “¿Qué son los servidores de aplicaciones?” KeepCoding Bootcamps. Accedido el 25 de marzo de 2024. [En línea]. Disponible: <https://keepcoding.io/blog/que-son-los-servidores-de-aplicaciones/>

[3] “Qué es Hosting - Definición, significado y ejemplos”. Arimetrics. Accedido el 25 de marzo de 2024. [En línea]. Disponible: <https://www.arimetrics.com/glosario-digital/hosting>

[4] “¿Qué es Cloud Computing?” Salesforce. Accedido el 25 de marzo de 2024. [En línea]. Disponible: <https://www.salesforce.com/mx/cloud-computing/>

[5] S. Pittet. “Despliegue continuo | Atlassian”. Atlassian. Accedido el 25 de marzo de 2024. [En línea]. Disponible: <https://www.atlassian.com/es/continuous-delivery/software-testing/continuous-deployment>

[6] “Productos y servicios del nivel gratuito y de prueba gratuita”. Google Cloud. Accedido el 27 de marzo de 2024. [En línea]. Disponible: <https://cloud.google.com/free?hl=es-419>