

Programación Web: Configuración de Servidores

Christian Stalin Ordóñez Crespo

Ingeniería en Ciencias de la Computación, Universidad de Cuenca

Cuenca, Ecuador

christian.ordonez2310@ucuenca.edu.ec

Abstract— *This document shows some basic concepts in the world of web projects, concepts that must be clear to start working in the web world. A step-by-step tutorial is described to configure and deploy a simple web page on different servers in order to know how to differentiate and choose the best option according to our needs and the type of project we are carrying out. Finally, some conclusions about the development of the entire tutorial are mentioned.*

Resumen— *Este documento muestra algunos conceptos básicos en el mundo de proyectos web, conceptos que se deben tener claros para comenzar a trabajar en el mundo web. Se describe un tutorial paso a paso para configurar y desplegar una página web sencilla en diferentes servidores con el objetivo de saber diferenciar y elegir la mejor opción según nuestras necesidades y el tipo de proyecto que estemos realizando. Finalmente, se mencionan algunas conclusiones sobre el desarrollo de todo el tutorial.*

Palabras clave: Servidor, Web, Cloud, Hosting, Puerto.

I. INTRODUCCIÓN

En la actualidad, el despliegue de aplicaciones web y la gestión de servidores son aspectos fundamentales para el desarrollo y la operación exitosa de proyectos en línea. Los servidores web, servidores de aplicaciones y la computación en la nube ofrecen diversas soluciones para alojar y ejecutar aplicaciones web, cada una con sus propias características y ventajas.

El objetivo de este trabajo es proporcionar una vista de algunos conceptos básicos en relación al desarrollo web, conceptos como servidores web, hosting o cloud computing y principalmente mostrar un tutorial paso a paso para la configuración y despliegue de una página web sencilla en diferentes tipos de servidores como un servidor web XAMPP o un servidor de aplicaciones web Apache Tomcat. Al finalizar este tutorial, se espera que el lector tenga un vista general para configurar y desplegar sus propias aplicaciones web en diferentes entornos de servidor, dependiendo sus necesidades.

II. MARCO TEÓRICO

A. SERVIDOR WEB

Es un programa de software diseñado para servir páginas web, contenido estático o dinámico, a los clientes que lo soliciten a través de internet o de una red local. Funciona como intermediario entre los navegadores web de los usuarios y el servidor donde está alojado el sitio web. Cuando un usuario ingresa la dirección de un sitio web en un navegador, el servidor web recibe esta solicitud, busca la página web correspondiente y la envía de vuelta al navegador del usuario.

Ejemplos de servidores web:

- Apache HTTP Server: Es uno de los más utilizados en el mundo, de código abierto y es altamente configurable y compatible con los sistemas operativos, como Linux, Windows, y macOS.
- Nginx: Es un servidor web con buena eficiencia y escalabilidad, diseñado para manejar grandes volúmenes de tráfico web y es conocido por su rendimiento de alta velocidad y su bajo uso de recursos.
- Caddy: Es un servidor web moderno y de código abierto, destaca por su facilidad de uso y configuración. Incluye características avanzadas como soporte para HTTPS automático y renovación de certificados SSL, lo que lo hace ideal para proyectos de desarrollo web y sitios pequeños.[1]

B. SERVIDOR DE APLICACIONES

Un servidor de aplicaciones es un software diseñado para alojar aplicaciones y permitir que se ejecuten de una forma remota a través de una red. Los servidores de aplicaciones proporcionan un entorno de ejecución de aplicaciones que puede incluir servicios como seguridad, transacciones y gestión de recursos. Su función principal es gestionar la comunicación entre el cliente y el servidor, así como ofrecer servicios y herramientas que permitan el desarrollo, despliegue y administración de

aplicaciones de manera eficiente. Algunas de sus características incluyen:

- Gestión de solicitudes HTTP: son capaces de recibir y procesar solicitudes HTTP enviadas por los clientes y enviar respuestas adecuadas.
- Contenedor de aplicaciones: proporciona un entorno de ejecución para las aplicaciones web, así como permite la gestión de recursos como memoria, hilos de ejecución y conexiones de red.
- Soporte para múltiples lenguajes y tecnologías: estos suelen ser compatibles con una variedad de lenguajes de programación y tecnologías web, como Java, PHP, Python, Ruby, entre otros.
- Escalabilidad: generalmente los servidores de aplicaciones están diseñados para ser escalables horizontalmente, lo que significa que pueden manejar un mayor volumen de tráfico distribuyendo la carga entre múltiples instancias del servidor.
- Seguridad: suelen incluir características de seguridad integradas, como la autenticación de usuarios, autorización de acceso a recursos y cifrado de datos, para proteger las aplicaciones y los datos del usuario.

Ejemplos de un servidor de aplicaciones:

- Apache Tomcat: Es un servidor de aplicaciones web de código abierto diseñado para ejecutar aplicaciones web.
- Jetty: Se utiliza comúnmente en aplicaciones web embebidas y en sistemas distribuidos.
- Microsoft Internet Information Services (IIS): Es un servidor de aplicaciones desarrollado por Microsoft para el sistema operativo Windows. Es ampliamente utilizado en entornos corporativos que utilizan tecnologías Microsoft, como ASP.NET y aplicaciones web basadas en el marco .NET.[2]

C. HOSTING

El hosting se refiere al servicio que proporciona una empresa para almacenar y mantener un sitio web en sus servidores, haciéndolo accesible en internet. Esencialmente, el hosting permite que un sitio web esté disponible en línea para que los usuarios lo vean y accedan a su contenido en cualquier momento. Este servicio incluye la asignación de espacio en disco, ancho de banda y otras características técnicas necesarias para que un sitio web funcione correctamente. Los proveedores de hosting ofrecen una variedad de planes y servicios que se adaptan a las necesidades específicas de los sitios web, desde opciones básicas para sitios personales hasta soluciones más avanzadas para empresas y aplicaciones web de alto tráfico.[3]

D. CLOUD COMPUTING

La computación en la nube es acceso a demanda a través de internet, a recursos informáticos, servidores, almacenamiento de datos, herramientas de desarrollo, capacidades de creación de redes y más, aliados en un centro de datos remoto administrado por un procesador de servicios en la nube. Estos proveedores ponen a disposición estos recursos por una tarifa de suscripción mensual o los factura en función del uso.

La computación en la nube, en comparación con la computación tradicional, ayuda a:

- Reducir los costos de TI: permite reducir algunos o la mayoría de los costos y el esfuerzo de comprar, instalar, configurar y administrar su propia infraestructura local.
- Mejorar la agilidad y el tiempo de creación de valor: la empresa puede comenzar a utilizar aplicaciones empresariales en cuestión de minutos, en lugar de esperar semanas o meses para que TI responda a una solicitud, compre y configure hardware de soporte e instale el software.
- Escalar de manera más fácil y rentable: la nube proporciona elasticidad, en lugar de comprar capacidad excedente que permanece sin usar durante períodos largos de tiempo, puede aumentar o reducir la capacidad en respuesta a picos y caídas en el tráfico.[4]

E. 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. El despliegue continuo ofrece varias ventajas para las empresas que quieren escalar su portafolio de aplicaciones y TI. En primer lugar, agiliza el tiempo de comercialización al eliminar el desfase entre la programación y el valor al cliente, que de otra manera podría demorarse días, semanas o incluso meses.[5]

III. DESARROLLO: CONFIGURACIÓN DE SERVIDORES

A. APACHE XAMPP

Con el programa XAMPP instalado desde su página oficial, procedemos a configurar el puerto del servidor. El puerto por defecto que tiene este servidor es el puerto 80 y para este caso vamos a elegir cambiarlo al puerto 8085. Para lograr esto, abrimos la aplicación XAMPP y en la pantalla principal seleccionamos el botón llamado “config” como se muestra en la siguiente imagen:

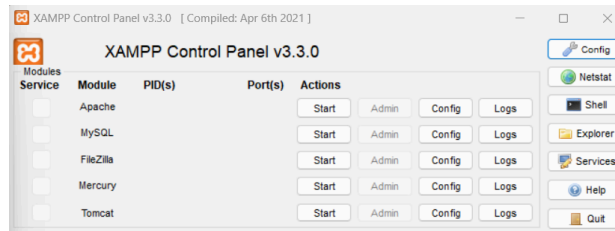


Figura 1. Pantalla principal de XAMPP.

Luego, en la pantalla desplegada, presionamos el botón llamado “Service and Port Settings”:

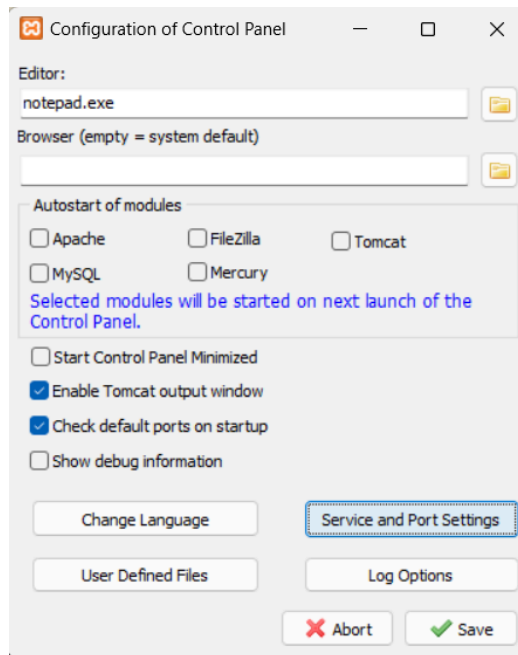


Figura 2. Panel de configuración de XAMPP.

En el campo llamado “Main Port” cambiamos el valor del puerto al 8085 y presionamos “Save” en las dos pantallas desplegadas:

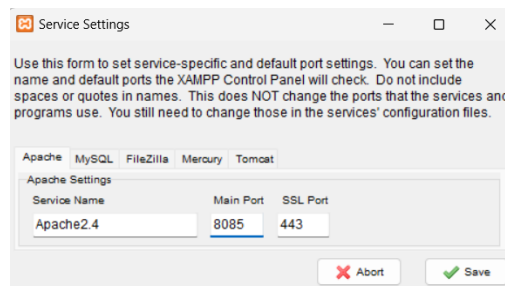


Figura 3. Cambio de puerto del servidor.

Además, para que el cambio se realice completamente, se debe cambiar el mismo valor del puerto 8085 en el archivo “httpd.conf” de apache desde la pantalla principal. En el botón “config” debemos seleccionar el archivo mencionado. Al abrir el archivo “httpd.conf”, presionamos la combinación de teclas “CTRL + F” para realizar una búsqueda en todo el

archivo. El texto que buscamos es el valor del puerto anterior que era “80” para reemplazarlo con el nuevo valor del puerto que es “8085”, el cambio se lo debe realizar en la etiqueta “Listen” y “ServerName”, guardar los cambios y cerrar el archivo:



Figura 4. Selección del archivo “httpd.conf”.

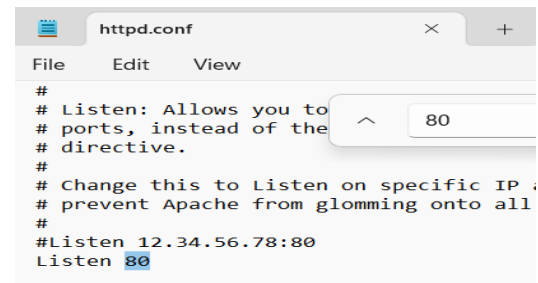


Figura 5. Cambiar el valor Listen a 8085.

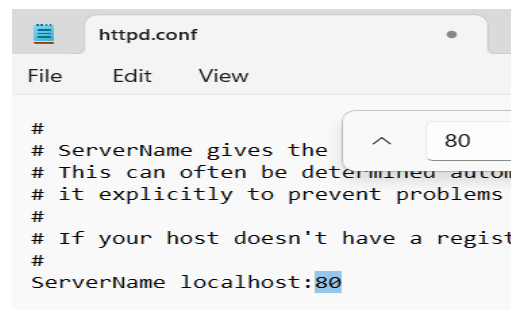


Figura 6. Cambiar el valor de ServerName a 8085.

Finalmente, cerramos la aplicación y la volvemos a abrir para que los cambios realizados tengan efecto, para probar que están funcionando los cambios iniciamos el servidor de Apache desde el botón “Start” y en el navegador escribimos la dirección “localhost:8085” y nos debería dirigir al dashboard del servidor de Apache:



Figura 7. Servidor iniciado en el puerto 8085.

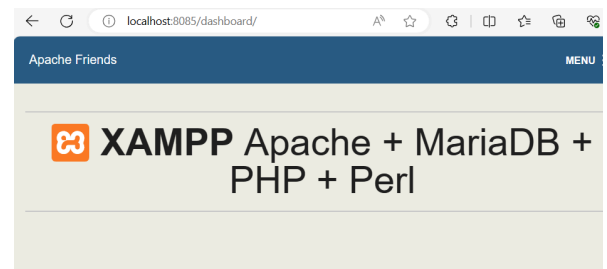


Figura 8. Dashboard del servidor.

Para desplegar nuestra primera página web sencilla de “Hola Mundo” nos dirigimos a la carpeta “htdocs” que se encuentra en la ruta de instalación de XAMPP, que generalmente es “C:\xampp\htdocs”. Aquí encontraremos los archivos por defecto de XAMPP, para tener una mejor distribución colocamos todos esos archivos en una nueva carpeta que la llamaremos “xampp2”. Luego colocamos en la ruta de “htdocs” nuestra primera página web que creamos desde cualquier IDE:

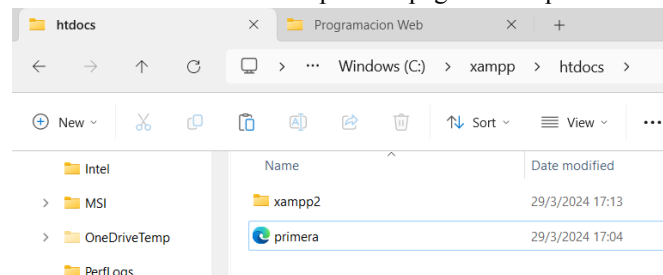


Figura 9. Copiar nuestra página web en el directorio.

Finalmente, desde el navegador ingresamos a la dirección de nuestro servidor local “localhost:8085” y le agregamos la ruta a nuestra primera página web que en este caso es “/primera.html” y el resultado debe ser que se despliega la página que creamos:



Figura 10. Primera página web en XAMPP.

Creamos una carpeta llamada “Carpeta_Publica” en algún directorio que creamos conveniente para colocar nuestros proyectos web, ésta será nuestra carpeta de publicación. Para configurar la carpeta como la carpeta de publicación, nos dirigimos al archivo “httpd.conf” de Apache desde el botón “Config” en la pantalla principal de la aplicación de XAMPP. Con la combinación de teclas “CTRL + F” buscamos la palabra “DocumentRoot” que contendrá la ruta por defecto de publicación que es la ruta hacia la carpeta “htdocs”. Entonces debemos cambiar la ruta hacia nuestra carpeta llamada “Carpeta_Publica” tanto en la etiqueta de “DocumentRoot” como en la etiqueta “<Directory>” que deberían ser las mismas rutas para que funcione nuestro servidor.

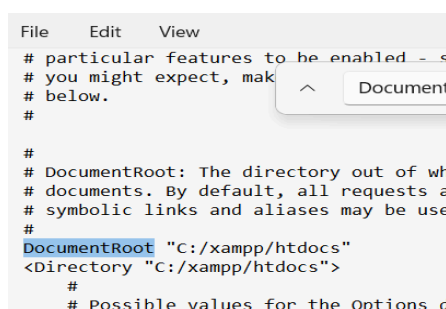


Figura 11. Ruta por defecto de publicación.

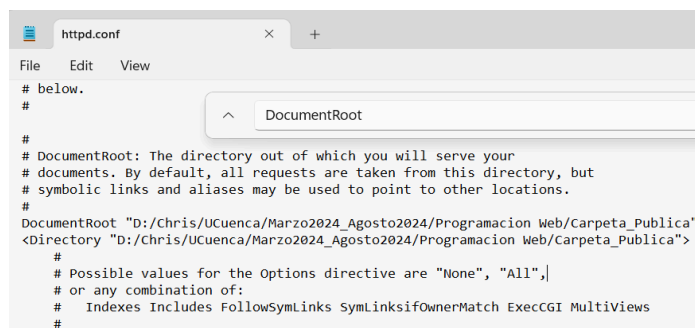


Figura 12. Cambio a la carpeta de publicación.

Guardamos los cambios, cerramos el archivo y reiniciamos nuestro servidor de Apache para que los cambios tengan efecto. Ahora nuestra dirección de publicación debería ser la carpeta que creamos, es decir, en esa carpeta debemos colocar nuestros proyectos web que queremos desplegar. Al iniciar el servidor, carga directamente nuestra primera página web.



Figura 13. Página en la nueva carpeta de publicación por defecto.

La estructura para publicar o construir un proyecto web puede variar según el tipo de proyecto que se esté desarrollando, ya sea por su naturaleza o por el lenguaje que se está utilizando. Lo que se debe tener en cuenta para estructurar de buena manera un proyecto de software es dividir todos los archivos, paquetes y código de tal manera que mientras más crezca el proyecto, sea fácil mantener, agregar o corregir las funcionalidades del proyecto. Se debe dividir y estructurar el proyecto para que cada archivo o parte de código tenga cohesividad, modularidad y escalabilidad. Por ejemplo para un proyecto web de una universidad se podría seguir la siguiente estructura de organización de archivos:

- proyecto_web_universidad
 - frontend
 - index.html
 - carpeta_estilos_css
 - styles.css
 - botones.css
 - campos_texto.css
 - carpeta_scripts_js
 - script.js
 - nuevo_estudiante.js
 - registrar_calificaciones.js
 - consulta_notas.js
 - eventos.js
 - carpeta_imagenes
 - logo_universidad.png
 - foto_perfil.png
 - escudo.png
 - carpeta_paginas_html
 - pagina_home.html
 - carpeta_paginas_facultades
 - pagina_facultad1.html
 - pagina_facultad2.html
 - pagina_login.html
 - pagina_perfil.html
 - pagina_horarios.html
 - carpeta_componentes_html
 - formulario_nuevo_estudiante.html
 - formulario_nueva_matricula.html
 - carpeta_recursos
 - backend
 - archivo_main
 - carpeta_dependencias
 - pom.xml
 - carpeta_recursos
 - imagen.png
 - archivo.xml
 - carpeta_modelo
 - entidades
 - modelo_solicitud_entidades
 - modelo_respuesta_entidades
 - carpeta_servicios
 - servicio_estudiantes
 - servicio_profesores
 - servicio_matriculas
 - carpeta_controladores
 - api_notas
 - api_estudiantes
 - api_profesores
 - api_foros
 - api_matriculas
 - assets
 - fonts
 - otros

Es decir, se podría dividir cada paquete hasta la modularidad que se requiera, incluso llegando a tener estilos para un grupo de botones, campos de texto, un archivo para cada facultad en cada carpeta, desde estilos, fuentes, recursos, servicios, entidades.

Para la configuración del firewall en el puerto específico que se quiera indicar, buscamos “firewall configuración avanzada” en el buscador de Windows, seleccionamos la sección de “Reglas de entrada” y creamos una nueva regla:

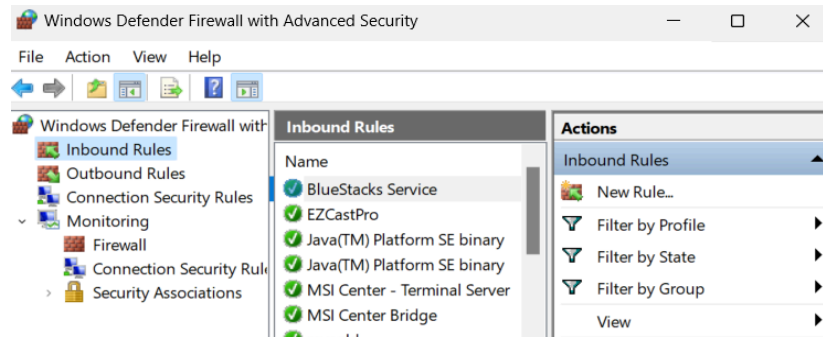


Figura 14. Seleccionar “New Rule” en la configuración avanzada del firewall de Windows.

Seleccionamos la opción para crear una regla para un puerto y colocamos el número de puerto que queremos habilitar. En este caso, como configuramos en XAMPP el puerto 8085, colocamos ese mismo puerto y permitimos la conexión.

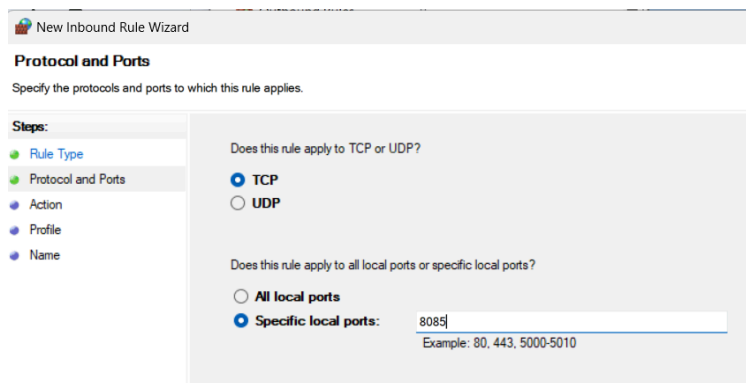


Figura 15. Configuración del puerto 8085 en el Firewall de Windows.

Colocamos el nombre “ApacheHTTP” a la regla que creamos y presionamos finalizar. Luego, presionamos doble clic sobre la columna llamada “Programa” en la regla creada y navegamos hacia la pestaña “Programas y servicios”, en donde en la parte superior seleccionamos “Este programa” y buscamos la ruta hacia un ejecutable llamado “httpd.exe” que se encuentra en la carpeta de instalación de XAMPP en la ruta “C:\xampp\apache\bin\httpd.exe”.

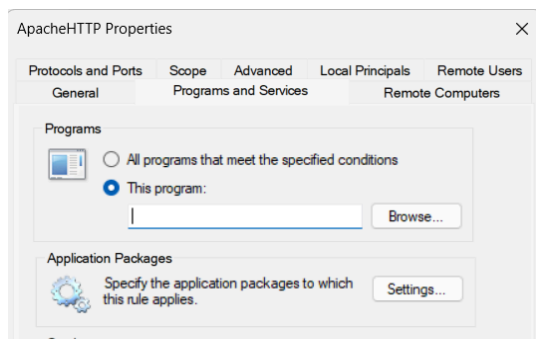


Figura 16. Programas y servicios.

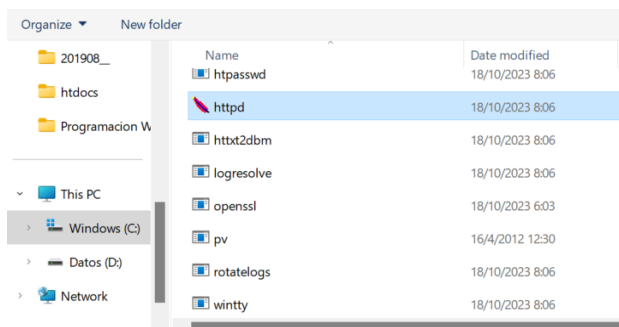


Figura 17. Selección del programa de XAMPP para uso del puerto habilitado.

Finalmente, reiniciamos o iniciamos el servidor de XAMPP el cual ahora estará disponible para escuchar solamente peticiones o solicitudes que lleguen desde el puerto 8085 especificado anteriormente.

B. APACHE TOMCAT

Con la aplicación de Apache Tomcat ya instalada desde su página oficial, procedemos a cambiar el puerto por defecto que es el 8080, en este caso vamos a colocar el puerto 8082 para este servidor. El cambio de puerto de Apache Tomcat se lo debe hacer desde un archivo llamado “server.xml” que está ubicado en la ruta de instalación de la aplicación, en la carpeta “conf”, para este caso la ruta del archivo es “C:\Program Files\Apache Software Foundation\Tomcat 10.1\conf\server.xml”. Lo abrimos con un editor de texto y procedemos a buscar el texto “8080” mediante la combinación de de teclas “CTRL + F” y realizamos el cambio al puerto que deseamos, al puerto 8082.

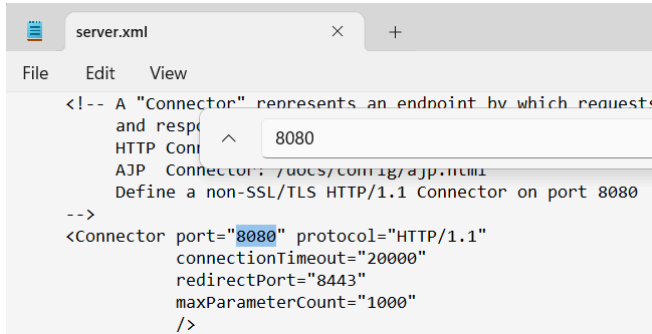


Figura 18. Configuración por defecto en server.xml.

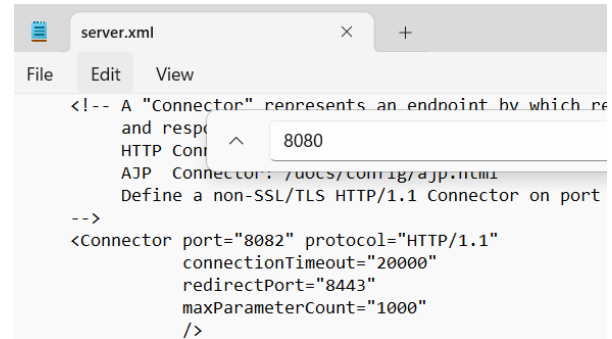


Figura 19. Cambio al puerto 8082.

Con los cambios realizados, el nuevo puerto del servidor debería ser el 8082, para comprobarlo, iniciamos el servidor y en el navegador nos dirigimos a la dirección “localhost:8082”, en la cual ingresamos al servidor de Apache Tomcat.

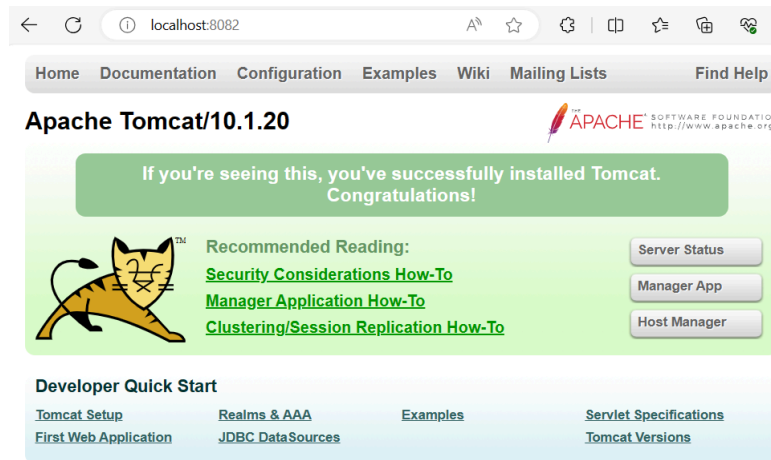


Figura 20. Página principal de Apache Tomcat en el puerto 8082.

Ahora vamos a desplegar una aplicación web sencilla que contenga un mensaje de “Hola Mundo”. En este caso creamos una aplicación sencilla de JSP (Java Server Pages) en IntelliJ llamada “HolaMundo”. Desde IntelliJ generamos el archivo llamado “HolaMundo.war” que será la aplicación que se despliegue en el servidor de Apache Tomcat. Entonces, desde la página inicial de Apache Tomcat en el navegador, ingresamos a “Manager App” con nuestras credenciales colocadas al momento de la instalación del servidor. Navegamos a la sección “Deploy” y en el apartado “WAR file to deploy” debemos buscar el archivo .war que generamos de nuestra aplicación JSP.

← ↻ ⓘ localhost:8082/manager/html ⌵ ⚙️ 📄 ⌵ 🔍

| | | | | | |
|---------|----------------|-----------------------------------|------|---|--|
| /webapp | None specified | Archetype Created Web Application | true | 1 | Undeploy |
| | | | | | Expire sessions with idle ≥ 30 minutes |

Deploy

Deploy directory or WAR file located on server

Context Path:

Version (for parallel deployment):

XML Configuration file path:

WAR or Directory path:

Deploy

WAR file to deploy

Select WAR file to upload Choose File No file chosen

Deploy

Figura 21. Sección Deploy de Manager App.

Una vez seleccionado nuestro archivo .war creado, presionamos el botón “Deploy” para desplegar nuestra primera aplicación sencilla de “HolaMundo” en el servidor de Apache Tomcat. En la sección llamada “Applications” se mostrará la aplicación que acabamos de desplegar.

WAR file to deploy

Select WAR file to upload Choose File HolaMundo.war

Deploy

Figura 22. Aplicación seleccionada para desplegar.

| Applications | | | | | |
|--------------|----------------|-----------------------------------|---------|----------|---|
| Path | Version | Display Name | Running | Sessions | Commands |
| / | None specified | Welcome to Tomcat | true | 0 | Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes |
| /HolaMundo | None specified | Archetype Created Web Application | true | 0 | Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes |
| /docs | None specified | Tomcat Documentation | true | 0 | Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes |
| /manager | None specified | Tomcat Manager Application | true | 1 | Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes |
| /webapp | None specified | Archetype Created Web Application | true | 1 | Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes |

Figura 23. Lista de aplicaciones en nuestro servidor.

Finalmente, accedemos a la aplicación con un click en el nombre de nuestra aplicación y tenemos el resultado que la aplicación se ha desplegado correctamente en el servidor y en el puerto 8082 configurado.



Figura 24. Aplicación desplegada en el servidor.

Para cambiar la carpeta de publicación del servidor de Apache Tomcat, nos dirigimos nuevamente al archivo “server.xml” y buscamos el texto “appBase” que está dentro de la etiqueta <Host>. Por defecto la ruta de publicación es la carpeta “webapps”, entonces la vamos a cambiar por la ruta de la carpeta que creamos llamada “CarpetaPublicaTomcat” en la ubicación que creamos conveniente.

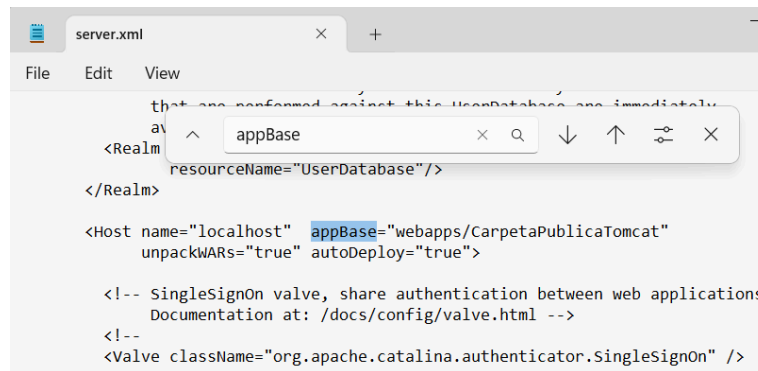


Figura 25. Cambio hacia la nueva ruta de la carpeta de publicación.

Dentro de la carpeta de publicación del servidor de Tomcat, siempre debe existir el proyecto o carpeta llamado “ROOT”, entonces colocamos nuestra aplicación de “Hola Mundo” con el nombre “ROOT” dentro de la carpeta pública que creamos y reiniciamos el servidor. Ahora en el navegador al entrar a la dirección “localhost:8082” veremos inmediatamente nuestra aplicación desplegada.



Figura 26. Aplicación web en la nueva carpeta de publicación.

C. GOOGLE CLOUD

Google Cloud Platform (GCP) es una plataforma de computación en la nube que ofrece una amplia gama de servicios para ayudar a empresas y desarrolladores a construir, implementar y escalar aplicaciones y servicios en la infraestructura global de Google. Cuenta con una red global de centros de datos y una amplia gama de servicios integrados, GCP permite a las empresas innovar rápidamente y llevar sus aplicaciones al mercado con mayor rapidez. Además, Google Cloud ofrece un alto nivel de seguridad, cumplimiento normativo y rendimiento para satisfacer las necesidades de todo tipo de cargas de trabajo, desde aplicaciones web y móviles hasta soluciones empresariales complejas. Algunas de sus características son:

- **Costos:** Google Cloud ofrece un modelo de precios flexible que incluye opciones de pago por uso y tarifas preestablecidas. Los costos pueden variar según el servicio utilizado, la región y la cantidad de recursos consumidos. Además, GCP ofrece programas de créditos gratuitos para nuevos usuarios y descuentos por uso continuado, por ejemplo, al iniciar sesión como un nuevo usuario de la plataforma se obtienen 300 créditos gratis
- **Tecnología:** Google Cloud admite una amplia gama de tecnologías y herramientas, incluidas soluciones de infraestructura como Google Compute Engine para máquinas virtuales, Google Kubernetes Engine para orquestación de contenedores, Google Cloud Storage para almacenamiento de objetos, BigQuery para análisis de datos, y muchos más
- **Servicios ofrecidos:** Google Cloud ofrece una amplia variedad de servicios en categorías como computación, almacenamiento, bases de datos, redes, inteligencia artificial y aprendizaje automático, análisis de datos, Internet de las cosas, seguridad y más. Algunos de los servicios más populares incluyen Google Compute Engine, Google Kubernetes Engine, Google Cloud Storage, BigQuery, Cloud SQL, Cloud Firestore, Cloud Functions, entre otros

IV. CONCLUSIONES

En conclusión, se han mencionado los conceptos claves sobre aplicaciones web, lo que sirve para entender la diferencia entre un servidor web y un servidor de aplicaciones web, saber qué es el hosting y tener en cuenta las ventajas que ofrece el cloud computing en función de las necesidades que tengamos para nuestro proyecto web. Esta práctica sirve como una pequeña introducción al mundo web, en el aspecto de que se han configurado servicios de la manera más sencilla e intuitiva posible, tanto en la configuración de los servidores como en la construcción del proyecto web que se ha desplegado. Se debe tener siempre en cuenta las ventajas que podría suponer utilizar cloud computing para nuestro proyecto antes de lanzarnos a comprar o implementar nuestros propios servidores, ya que muchos de estos proveedores de cloud ponen a disponibilidad el pago por uso, que puede ser la mejor opción.

V. REFERENCIAS

- [1] “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 31 de marzo de 2024. [En línea]. Disponible: <https://blog.hubspot.es/website/que-es-servidor-web#:~:text=5%20caracteristicas%20de%20un%20servidor%20web%201%201.,de%20informaci3n%20...%205%205.%20Alta%20capacidad%20>
- [2] “Servidor de aplicaciones: todo lo que necesitas saber - Polaridad.es”. Polaridad.es. Accedido el 31 de marzo de 2024. [En línea]. Disponible: <https://polaridad.es/servidor-de-aplicaciones-todo-lo-que-necesitas-saber/#:~:text=Los%20servidores%20de%20aplicaciones%20web%20son%20aquellos%20que,aplicaciones%20web%20son%20Apache%20Tomcat,%20JBoss%20y%20GlassFish.>
- [3] “¿Qué es un hosting y cómo funciona?” Tutoriales Hostinger. Accedido el 31 de marzo de 2024. [En línea]. Disponible: <https://www.hostinger.mx/tutoriales/que-es-un-hosting>
- [4] “¿Qué es la computación en la nube? | IBM”. IBM in Deutschland, Österreich und der Schweiz. Accedido el 31 de marzo de 2024. [En línea]. Disponible: <https://www.ibm.com/mx-es/topics/cloud-computing>
- [5] “¿Qué es el despliegue continuo? | IBM”. IBM in Deutschland, Österreich und der Schweiz. Accedido el 31 de marzo de 2024. [En línea]. Disponible: <https://www.ibm.com/es-es/topics/continuous-deployment>