



UNIVERSIDAD NACIONAL DE CHIMBORAZO
FACULTAD DE INGENIERÍA
CARRERA DE INGENIERÍA EN TECNOLOGÍAS DE LA
INFORMACIÓN

Unach
UNIVERSIDAD NACIONAL DE CHIMBORAZO
en movimiento

UNIVERSIDAD NACIONAL DE CHIMBORAZO
FACULTAD DE INGENIERÍA



INGENIERÍA EN TECNOLOGÍAS DE LA
INFORMACIÓN

NOMBRE:

Elvis Santiago Pilco Paucar

FECHA: 2024/04/16

INTEROPERABILIDAD DE PLATAFORMAS

SISTEMAS DISTRIBUIDOS

PERIODO ACADÉMICO 2024-1S



1. OBJETIVO GENERAL, ESPECÍFICOS, RECURSOS

1.1. OBJETIVO GENERAL

Aplicar los conocimientos teóricos relacionados con la arquitectura y el diseño de sistemas distribuidos adquiridos en el desarrollo de la unidad, a través del desarrollo de una aplicación distribuida e interoperable usando tecnología RPC.

1.2. OBJETIVOS ESPECÍFICOS

- Demostrar la interoperabilidad de plataformas mediante la creación, desarrollo y ejecución de sistemas distribuidos construidos bajo diferentes tecnologías.
- Simular las operaciones asociadas con los sistemas distribuidos mediante el uso de herramientas informáticas que muestren el proceso interno y externo de este tipo de software.
- Comprender los conceptos fundamentales y prácticas relativas a la arquitectura orientada a servicios mediante el desarrollo de un aplicativo que involucre los conceptos más relevantes de un sistema distribuido.

1.3. RECURSOS

- Computador
- VirtualBox
- Navegador web

2. INTRODUCCIÓN.

En el escenario tecnológico actual, los sistemas distribuidos son indiscutiblemente un pilar importante para la informática moderna. Estos sistemas, definidos como una colección de computadoras independientes que aparecen frente a los usuarios como una única computadora [1], brindan una solución innovadora para enfrentar los desafíos de escalabilidad, disponibilidad y rendimiento en aplicaciones y servicios. Es así como la naturaleza distribuida de estos sistemas les permite ser altamente resilientes y capaces de soportar fallos individuales sin interrumpir el funcionamiento general. Asimismo, su arquitectura descentralizada permite agregar más nodos al sistema para aumentar su capacidad (fácilmente escalable horizontalmente) [2], al mismo tiempo que por la distribución de tareas, existe un mayor aprovechamiento de los recursos disponibles. Este tipo de sistemas se utilizan en una amplia variedad de contextos, desde aplicaciones web y móviles hasta sistemas de control de procesos y redes de sensores. Algunos ejemplos de su uso se dan en la gestión de bases de datos, el procesamiento de imágenes y vídeos, el análisis de datos, el procesamiento de transacciones financieras y muchas otras aplicaciones [3] que, por la cualidad de distribución de los cálculos y almacenamiento de datos a través de diferentes componentes, garantizan una mayor eficiencia y capacidad de respuesta.

En un futuro cada vez más interconectado, los sistemas distribuidos serán fundamentales en la infraestructura tecnológica. Su uso es una realidad y es esencial comprender los principios y desafíos de este tipo de sistemas para diseñar y desarrollar soluciones eficientes y robustas en un mundo digital en constante evolución. Es por todo lo anterior que resulta necesario la creación de software basado en este tipo de sistemas mediante el uso de distintas tecnologías y herramientas a



priori del desarrollo de soluciones innovadoras y eficientes para enfrentar los desafíos de hoy y mañana.

3. MARCO TEÓRICO.

3.1. SISTEMA DISTRIBUIDO

Un sistema distribuido es una colección de computadoras interconectadas a través de una red que trabajan juntas para lograr una tarea común [4]. El objetivo de estos sistemas es compartir recursos y coordinar la ejecución de tareas de manera eficiente, distribuyendo la carga de trabajo entre múltiples ordenadores. La implementación sistemas distribuidos presenta desafíos técnicos relacionados con la gestión de concurrencia, y la garantía de la consistencia en la distribución de datos en tiempo real. [4] Aún así, este tipo de sistemas pueden proporcionar ventajas significativas sobre otras tecnologías entre las que destacan la redundancia y tolerancia a fallos; con el objetivo de mejorar la eficiencia de las operaciones y reducir costos en diversas aplicaciones empresariales.

3.2. RPC

Un sistema RPC (Remote Procedure Call) es un modelo que permite a una aplicación solicitar a otra aplicación o proceso la ejecución de una función en otra máquina a través de la red. En este modelo, la máquina solicitante envía una solicitud al servidor, que procesa la solicitud y devuelve el resultado a través de la red [5]. De esta manera, las aplicaciones pueden interactuar y trabajar juntas de manera eficiente y coordinada, sin importar su ubicación física. El uso de sistemas RPC puede ayudar a simplificar el diseño y la implementación de sistemas distribuidos dado que permite una comunicación segura y eficiente entre diferentes aplicaciones. Los sistemas RPC se utilizan comúnmente en entornos empresariales para permitir la interacción entre diferentes sistemas y aplicaciones, y existen diferentes implementaciones y protocolos entre los que están RPC, XML-RPC, JSON-RPC, etc [5].

3.3. WEB SERVICE

Un servicio web es una tecnología de comunicación que permite a diferentes aplicaciones y sistemas intercambiar datos a través de una red, utilizando un conjunto de estándares y protocolos comunes. El objetivo principal de los servicios web es proporcionar una plataforma independiente del lenguaje de programación y la plataforma [9], lo que permite que diferentes sistemas se comuniquen de manera efectiva, se integren fácilmente y se reutilicen. Los servicios web se utilizan en una amplia variedad de aplicaciones, desde la integración de sistemas empresariales hasta el intercambio de información entre diferentes plataformas web y móviles, mediante dos estándares tecnológicos principales: SOAP y REST, que permiten estructurar la información intercambiada mediante protocolos como HTTP.

3.4. SERVICIOS WEB EXTERNO Y SU CONSUMO

A servicio web externo se refiere a aquellos servicios web que se encuentran fuera de una empresa u organización, es decir, que son proporcionados por terceros. Estos servicios permiten a las empresas acceder a una amplia variedad de recursos y funcionalidades, lo que puede resultar en una mayor eficiencia y competitividad en el mercado. Los servicios web externos utilizan estándares comunes como SOAP y REST y se comunican a través de protocolos de comunicación como HTTP. [10]



Para consumir un servicio web externo desde una aplicación, es primordial conocer la ubicación del servicio web, la estructura de sus mensajes y los métodos disponibles. Con esta información es posible utilizar librerías en diferentes lenguajes de programación que proporcionan una interfaz para la invocación de estos servicios. [11]

3.5. TECNOLOGIAS DE LOS SISTEMAS DISTRIBUIDOS

3.5.1. JAVA

Java es un lenguaje de programación que fue desarrollado por James Gosling y su equipo de Sun Microsystems en la década de 1990 [12]. Está disponible de forma gratuita bajo la licencia GPL y es compatible con una amplia gama de sistemas operativos. Es utilizado para el desarrollo de aplicaciones móviles, de escritorio y para la web, además de que es un lenguaje orientado a objetos, multiplataforma y de alto nivel que se ejecuta dentro de una máquina virtual, lo que permite que sus aplicaciones sean independientes del sistema operativo y la plataforma de hardware subyacente [12].

3.5.2. NETBEANS

NetBeans es un entorno de desarrollo integrado (IDE) de código abierto utilizado principalmente para desarrollar aplicaciones en Java, pero también es compatible con otros lenguajes de programación como C++, PHP y HTML5. Este IDE ofrece una amplia gama de herramientas y características para el desarrollo de aplicaciones, así como un amplio soporte de componentes y bibliotecas que pueden integrarse fácilmente en los proyectos [17].

4. METODOLOGÍA DE DESARROLLO.

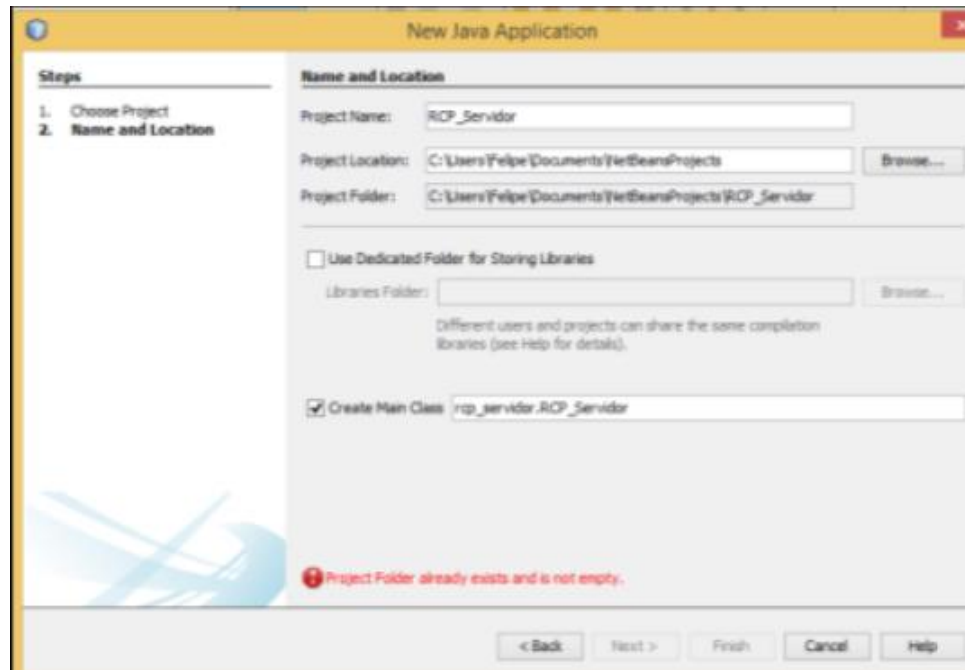
Para el desarrollo de la práctica acerca de la instalación paso a paso del framework, se sigue una metodología de desarrollo secuencial como se describe a continuación:

- **Análisis:** Con antecedente en la información recabada en la cátedra de la asignatura de Interoperabilidad de Plataformas, se hizo una revisión sobre la información relevante y las características de los sistemas basados en servicios web.
- **Diseño:** Con una comprensión del fundamento teórico se diseñó la aplicación, se dividió la creación en tres partes principales: descarga del framework, instalación y los comandos para la implementación.
- **Implementación:** Las fases de creación de la instalación se desarrollaron previamente a las horas correspondientes a la cátedra, para con el apoyo del docente y en función a las guías proporcionadas solventar cualquier inquietud.
- **Evaluación:** Cada una de las fases fue evaluada por el docente en las horas correspondientes de la asignatura para verificar comprensión del tema, aclarar dudas y demostrar los conocimientos.
- **Documentación:** Finalmente, se elaboró un informe que recabe la información relativa a la instalación de Laravel, en donde se detalla la información de todo el proceso de desarrollo.



5. DESARROLLO DE LA PRACTICA.

1. Como primer paso, añadir un proyecto para nuestro servidor, en este caso lo llamamos “RCP_Servidor”



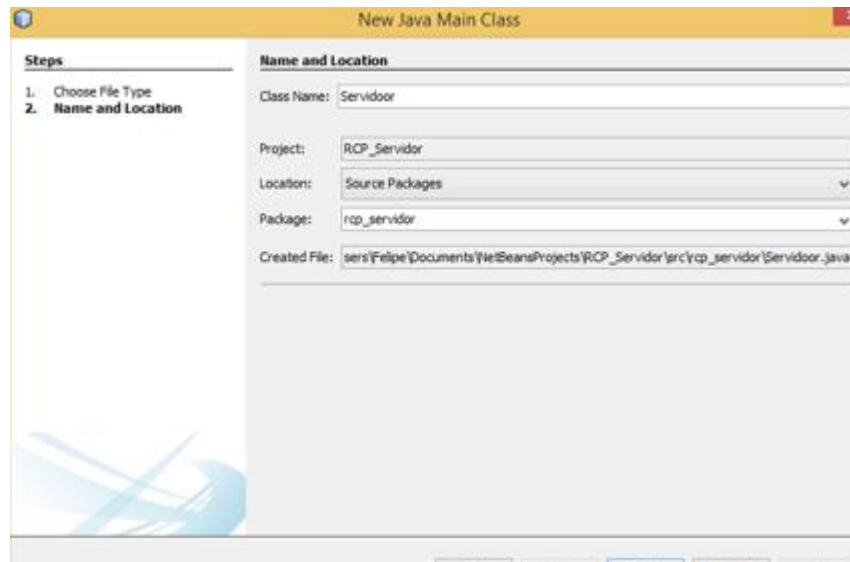
2. Crear una clase para la funcionalidad de nuestro sistema en este caso, “Operaciones” aquí estarán los métodos de suma, resta, multiplicar y división.

```
Source | History | ... |
1
2 package rop_servidor;
3
4
5
6 public class Operaciones {
7
8     public String suma(String a,String b){
9         return String.valueOf(Integer.parseInt(a)+ Integer.parseInt(b));
10    }
11
12     public String resta(String a,String b){
13         return String.valueOf(Integer.parseInt(a)- Integer.parseInt(b));
14    }
15
16     public String multiplicacion(String a,String b){
17         return String.valueOf(Integer.parseInt(a)* Integer.parseInt(b));
18    }
19
20     public String division(String a,String b){
21         return String.valueOf(Integer.parseInt(a)/ Integer.parseInt(b));
22    }
23 }
```

3. Añadir una Main Class llamada “Servidor” para llamar a los métodos y manejar las funciones de nuestra clase Operaciones.



UNIVERSIDAD NACIONAL DE CHIMBORAZO
FACULTAD DE INGENIERÍA
CARRERA DE INGENIERÍA EN TECNOLOGÍAS DE LA
INFORMACIÓN



- Primero se debe declarar las librerías necesarias y también el código necesario para la iniciar el servidor y llamar a los métodos de nuestra clase operaciones.

```
1 package rcp_servidor;
2
3
4 import javax.swing.JOptionPane;
5 import org.apache.xmlrpc.WebServer;
6
7 public class Servidor {
8
9
10 public static void main(String[] args) {
11
12     try {
13         JOptionPane.showMessageDialog(null, "Iniciando el servidor");
14         WebServer server = new WebServer(8080);
15         Suma suma = new Suma();
16         Suma resta = new Suma();
17         Suma multiplicacion = new Suma();
18         Suma division = new Suma();
19         server.addHandler("miServidorRpc", suma);
20         server.addHandler("miServidorRpc", resta);
21         server.addHandler("miServidorRpc", multiplicacion);
22         server.addHandler("miServidorRpc", division);
23         server.start();
24         JOptionPane.showMessageDialog(null, "Servidor en línea");
25     } catch (Exception e) {
26         JOptionPane.showMessageDialog(null, "Error"+e.getMessage());
27     }
28 }
29
30 }
```

- Añadir un nuevo proyecto para nuestro servidor, en este caso llamado “RCP_Cliente” donde crearemos nuestro formulario y conexión entre el cliente y servidor.

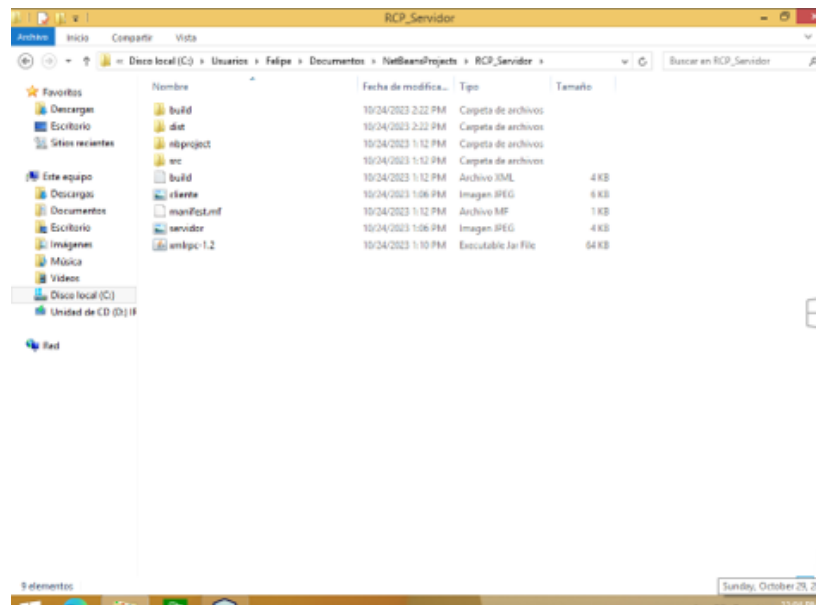


UNIVERSIDAD NACIONAL DE CHIMBORAZO
FACULTAD DE INGENIERÍA
CARRERA DE INGENIERÍA EN TECNOLOGÍAS DE LA
INFORMACIÓN



```
13  
14 RpcClient client = new XmlRpcClient("http://10.0.2.15:8080");  
15 tor<String> parametros = new Vector<String>();  
16 tionPane.showMessageDialog(null,"El cliente se ha conectado");  
17 le(true){  
18     String menu = JOptionPane.showInputDialog(null,"Operaciones con 2 numeros\n"  
19         + "1. Suma \n"  
20         + "2. Resta \n"  
21         + "3. Multiplicacion \n"  
22         + "4. Division \n"  
23         + "5. Salir \n","Cliente Servidor RPC", JOptionPane.DEFAULT_OPTION);  
24     switch(menu){  
25         case "1":  
26             x=JOptionPane.showInputDialog(null,"Primer numero","Suma",JOptionPane.QUESTION  
27             y=JOptionPane.showInputDialog(null,"Segundo numero","Suma",JOptionPane.QUESTIO  
28             parametros.addElement(x);  
29             parametros.addElement(y);  
30             resultado = client.execute("miServidorRpc.suma,parametros);  
31             JOptionPane.showMessageDialog(null,"La suma es "+resultado);  
32             parametros.clear();  
33             break;  
34         case "2":  
35             x=JOptionPane.showInputDialog(null,"Primer numero","Resta",JOptionPane.QUESTIO  
36             y=JOptionPane.showInputDialog(null,"Segundo numero","Resta",JOptionPane.QUESTIO  
37             parametros.addElement(x);  
38             parametros.addElement(y);  
39             resultado = client.execute("miServidorRpc.resta,parametros);  
40             JOptionPane.showMessageDialog(null,"La resta es "+resultado);  
41             parametros.clear();  
42             break;
```

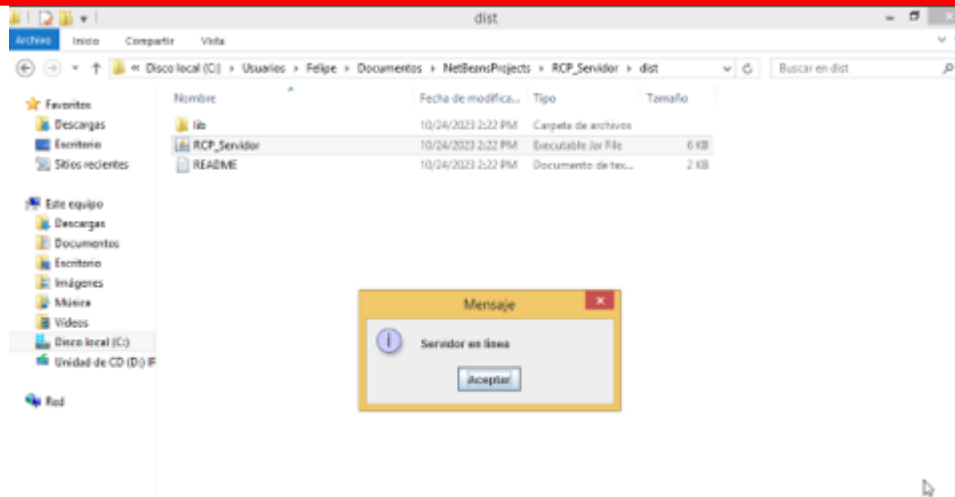
6. Editar las propiedades de estos, señalar todas las opciones de las categorías de Compiling, Packaging, Run y Application, donde se debe asociar una imagen a través de splash screen y al final generar nuestros proyectos.



7. A los archivos .jar de ambos proyectos se inician para usar nuestra aplicación de sistemas distribuidos RPC



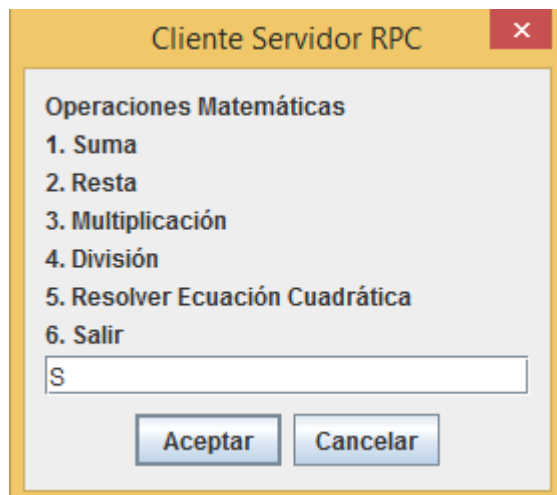
UNIVERSIDAD NACIONAL DE CHIMBORAZO
FACULTAD DE INGENIERÍA
CARRERA DE INGENIERÍA EN TECNOLOGÍAS DE LA
INFORMACIÓN



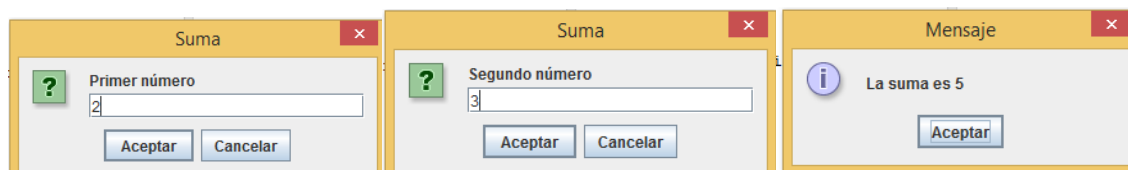
6. RESULTADOS.

6.1. PRUEBAS DE LOS MÉTODOS DEL SERVICIO WEB ASP .NET

A. MENU PRINCIPAL



B. SUMAR



C. RESTAR



Resta

?

Primer número

10

Aceptar Cancelar

Resta

?

Segundo número

2

Aceptar Cancelar

Mensaje

i

La resta es 8

Aceptar

D. MULTIPLICAR

Multiplicación

?

Primer número

8

Aceptar Cancelar

Multiplicación

?

Segundo número

7

Aceptar Cancelar

Mensaje

i

La multiplicación es 56

Aceptar

E. DIVIDIR

División

?

Dividendo

10

Aceptar Cancelar

División

?

Divisor

2

Aceptar Cancelar

Mensaje

i

La división es 5

Aceptar

F. ECUACIÓN CUADRÁTICA

Resolver Ecuación Cuadrática

?

Coefficiente A

1

Aceptar Cancelar

Resolver Ecuación Cuadrática

?

Coefficiente B

-5

Aceptar Cancelar

Resolver Ecuación Cuadrática

?

Coefficiente C

6

Aceptar Cancelar

Mensaje

i

Resultado de la ecuación cuadrática: Raíces reales: $x_1 = 3.0$, $x_2 = 2.0$

Aceptar



7. CONCLUSIONES.

- Los sistemas distribuidos desempeñan un papel fundamental en el entorno empresarial actual, ya que permiten una mayor eficiencia en la gestión de la información y los recursos, aumenta la capacidad de procesamiento, reduce los tiempos de respuesta, pero sobre todo poseen escalabilidad y disponibilidad, lo que los hace ideales para la realidad tecnológica presente y futura.
- Los web services son una tecnología potente y escalable que permiten el intercambio de datos entre sistemas de forma eficiente
- La utilización de XML en sistemas distribuidos es de gran importancia, ya que permite la integración entre distintas plataformas y sistemas, facilita la comunicación entre aplicaciones y garantiza su compatibilidad y adaptabilidad a diferentes entornos y necesidades.
- A pesar de los beneficios, la implementación de un servicio web puede ser complicada, requiriendo una planificación cuidadosa y una coordinación adecuada para garantizar que el proyecto sea completado a tiempo y con un alto estándar.

8. RECOMENDACIONES

- Es recomendable utilizar herramientas y tecnologías estándar para garantizar la compatibilidad con otros sistemas, al mismo tiempo que se aplican estándares como XML y SOAP para ayudar a asegurar la interoperabilidad con otros sistemas.
- Se recomienda validar los datos antes de utilizarlos y tener un manejo o control adecuado de los errores en el código; así pues, cuando se consume un web service externo, se debe asegurar la calidad de los datos recibidos, sin errores ni vacíos.
- Se recomienda el uso de herramientas para probar y depurar los web services, pues al ser una herramienta fácil de usar y altamente efectiva permite realizar pruebas tanto en servidores locales como en externos.

9. BIBLIOGRAFÍA

- [1] F. d. A. López, Sistemas Distribuidos, J. Rosas, Ed., Universidad Autónoma Metropolitana, 2015.
- [2] G. Coulouris, J. Dollimore, T. Kindberg y G. Blair, DISTRIBUTED SYSTEMS: Concepts and Design, Quinta ed., M. Hirsch, Ed., Pearson Education Inc, 2012.
- [3] S. Ghosh, Distributed Systems: An Algorithmic Approach, 2 ed., Iowa City: CRC Press, 2015.
- [4] A. Kshemkalyani y M. Singhal, Distributed Computing: Principles, Algorithms, and Systems, Cambridge: Cambridge University Press, 2011.
- [5] g. Authors, «gRPC,» 2023. [En línea]. Available: <https://grpc.io/about/>.
- [6] IBM, «IBM Topics,» 2023. [En línea]. Available: https://www.ibm.com/topics/soa?mhsrc=ibmsearch_a&mhq=soa.



-
- [7] T. Erl, Service-Oriented Architecture: Concepts, Technology, and Design, Pearson Education, Inc, 2005.
 - [8] W3C, «SERVICIOS,» 2007. [En línea]. Available: <https://www.w3.org/TR/soap12/>.
 - [9] W3C, «Working Group Note,» 2004. [En línea]. Available: <https://www.w3.org/TR/2004/NOTE-ws-arch-20040211/>.
 - [10] IBM, «Developer IBM,» 2023. [En línea]. Available: <https://developer.ibm.com/articles/what-is-a-web-service/>.
 - [11] Oracle, «Web Services Support,» 2023. [En línea]. Available: <https://docs.oracle.com/cd/E19798-01/821-1841/bnabs/index.html>.
 - [12] Oracle, «Java Documentation,» [En línea]. Available: <https://docs.oracle.com/javase/tutorial/>.
 - [13] Microsoft, «Microsoft Docs,» [En línea]. Available: <https://dotnet.microsoft.com/es-es/learn/dotnet/what-is-dotnet>.
 - [14] Mozilla, «Working with JSON,» 2023. [En línea]. Available: <https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/JSON>.
 - [15] Microsoft, «XML Documents and Data,» 2023. [En línea]. Available: <https://learn.microsoft.com/en-us/dotnet/standard/data/xml/>.
 - [16] W3C Schools, «JSON,» 2023. [En línea]. Available: https://www.w3schools.com/js/js_json_intro.asp.
 - [17] Apache, «Apache Netbeans,» 2023. [En línea]. Available: <https://netbeans.apache.org/>.
 - [18] Microsoft, «Visual Studio,» 2019. [En línea]. Available: <https://visualstudio.microsoft.com/vs/>.
 - [19] Postman INC, «Postman,» 2023. [En línea]. Available: <https://www.postman.com/>.