

Universidad Nacional de Chimborazo
Facultad de Ingeniería
Ingeniería en Tecnologías de la Información



Interoperabilidad de Plataformas

Nombre
Pablo Orozco C.

Docente
Ing. Milton López

Informe Sistemas Distribuidos

Objetivo

Desarrollar un sistema distribuido que permita ejecutar una calculadora con operaciones básicas utilizando NetBeans basado en RPC y dividiendo el sistema en módulos para que se ejecuten en nodos diferentes gestionando la conexión del cliente con el servidor.

Introducción

En el mundo actual, la necesidad de sistemas distribuidos eficientes y escalables ha llevado al auge de tecnologías que permiten la comunicación entre diferentes nodos de manera transparente. Uno de los enfoques más utilizados es el uso de RPC (Remote Procedure Call), una técnica que permite a un programa ejecutar procedimientos en un espacio de direcciones diferente, como si fueran locales.

Este proyecto tiene como objetivo principal desarrollar un sistema distribuido utilizando NetBeans como entorno de desarrollo, aprovechando las capacidades de RPC para facilitar la comunicación entre componentes distribuidos. En este contexto, NetBeans actuará como una herramienta integral que simplificará el diseño, desarrollo y depuración de nuestro sistema distribuido.

Exploraremos cómo NetBeans, con su conjunto de características robustas, puede optimizar el proceso de desarrollo para sistemas distribuidos basados en RPC. A lo largo de este proyecto, nos sumergiremos en la implementación de procedimientos remotos, gestionando la comunicación eficiente entre nodos y abordando los desafíos inherentes a la construcción de sistemas distribuidos escalables y tolerantes a fallos.

Al finalizar este proyecto, no solo se ganará un profundo conocimiento sobre el desarrollo de sistemas distribuidos, sino que también habrá aprovechado al máximo las herramientas que NetBeans proporciona para facilitar este proceso.

Marco Teórico:

Máquina Virtual Windows 8.1

“Básicamente una máquina virtual (MV) o virtual machine (VM) es una implementación de software de una máquina física ya sea un servidor o un pc. Estas máquinas virtuales utilizan recursos físicos como cpu, memoria y disco rígido del equipo físico donde corren los cuales le permiten a la misma correr un sistema operativo. Al mencionar que una máquina virtual es una implementación de software de una máquina física, estamos diciendo que lo que conforma a esta máquina virtual es un conjunto de archivos los cuales, en su mayoría, se pueden manipular como si se tratara de cualquier otro archivo presente en algún sistema operativo”.

Características:

- Aislamiento: Las máquinas virtuales proporcionan un entorno aislado para ejecutar sistemas operativos y aplicaciones, lo que significa que una VM no afecta directamente a otras VMs en el mismo host.
- Portabilidad: Puedes migrar fácilmente una máquina virtual de un host a otro, lo que facilita la implementación y la gestión de recursos.
- Snapshots: Las VMs permiten crear instantáneas o snapshots, que son imágenes del estado actual de la máquina virtual. Esto es útil para realizar copias de seguridad o para volver a un estado anterior.
- Encapsulamiento: Toda la configuración y datos de una máquina virtual están encapsulados en archivos, lo que facilita la copia, el traslado y la replicación.
- Eficiencia de recursos: Puedes ejecutar múltiples máquinas virtuales en un solo host físico, optimizando el uso de recursos y reduciendo costos.
- Facilidad de replicación: Las VMs pueden replicarse para crear entornos de desarrollo, prueba y producción similares, lo que ayuda a evitar problemas de compatibilidad.
- Escalabilidad: Puedes ajustar fácilmente los recursos asignados a una máquina virtual, como la cantidad de memoria RAM o el número de núcleos de CPU, para adaptarse a las necesidades de carga de trabajo.
- Desarrollo y pruebas: Las máquinas virtuales son ideales para entornos de desarrollo y pruebas, ya que permiten probar software en diferentes configuraciones sin afectar el entorno de producción.

Simplifica alguna de las tareas que, sobre todo en proyectos grandes, son tediosas. Asiste (parcialmente) en la escritura de código, pero no nos libera de aprender el lenguaje de programación. Ayuda en la navegación de las clases predefinidas en la plataforma (miles), Aunque puede ser costoso su aprendizaje, los beneficios superan las dificultades.

Netbeans no trabaja a nivel de archivo sino a nivel de proyecto. Un proyecto incluye todos los recursos necesarios para construir un programa:

- Archivos con el código
- Bibliotecas externas (p.e. ACM Task Force)
- Imágenes, sonidos, etc.

Físicamente un proyecto Netbeans no es más que un directorio con una organización especial.

“Netbeans es, ante todo, un entorno de desarrollo integrado para programadores de código abierto bien diseñado. Es potente, útil, extensible, abierto y gratuito”.

Java

Los padres de Java son James Gosling (emacs) y Bill Joy (Sun). Java desciende de un lenguaje llamado Oak cuyo propósito era la creación de software para la televisión interactiva. Las características de Oak eran:

- Pequeño.
- Robusto.
- Independiente de la máquina.
- Orientado a objetos.

El proyecto de televisión interactiva fracasó y el interés de los creadores de Oak sedirigió a Internet bajo el lema «La red es la computadora». Los criterios de diseño de Java fueron:

- Independiente de la máquina.
- Seguro para trabajar en red.
- Potente para sustituir código nativo.

“La principal característica de Java es la de ser un lenguaje compilado e interpretado. Todo programa en Java ha de compilarse y el código que se genera bytecodes es interpretado por una máquina virtual. De este modo se consigue la independencia de la máquina, el código compilado se ejecuta en máquinas virtuales que si son dependientes de la plataforma. Java es un lenguaje orientado a objetos de propósito general. Aunque Java comenzará a ser

conocido como un lenguaje de programación de applets que se ejecutan en el entorno de un navegador web, se puede utilizar para construir cualquier tipo de proyecto.

“Java implementa los protocolos de red estándares, lo que permite desarrollar aplicaciones cliente/servidor en arquitectura distribuida, con el fin de invocar tratamientos y/o recuperar datos de máquinas remotas.”.

Virtual Box

“Es posible que a alguno de nuestros lectores muchos de los conceptos expuestos en esta serie les sobrepasen, principalmente porque sus necesidades de virtualización sean poco ambiciosas. Es posible que busquen algo que se aleje de la consola, de compiladores del kernel, configuraciones de red extrañas, etc. Bien, es el momento de VirtualBox.” [9][10]

Un software de virtualización de código abierto desarrollado por Oracle. Características específicas de VirtualBox:

- Plataformas compatibles: Puede ejecutar VirtualBox en una variedad de sistemas operativos, como Windows, macOS, Linux y más.
- Soporte para múltiples sistemas operativos: VirtualBox es capaz de ejecutar una amplia gama de sistemas operativos, incluyendo Windows, Linux, macOS, Solaris y otros.
- Interfaz gráfica de usuario (GUI): Ofrece una interfaz fácil de usar que facilita la creación, configuración y gestión de máquinas virtuales.
- Extensiones: Puedes mejorar la funcionalidad de VirtualBox mediante la instalación de extensiones, que añaden soporte USB 2.0 y 3.0, VirtualBox RDP, entre otras características.
- Snapshots: Permite crear instantáneas para capturar el estado actual de una máquina virtual y revertir a ese estado en caso de problemas.
- Compatibilidad con hardware virtual: VirtualBox es compatible con la virtualización anidada y otras características de hardware virtual, lo que lo hace útil para entornos de desarrollo y pruebas avanzados.
- Opciones de red: Proporciona diversas opciones de configuración de red, incluyendo modos de red interna, red solo-anfitrión y red en modo puente.
- Portabilidad: Puedes exportar e importar máquinas virtuales fácilmente entre diferentes instancias de VirtualBox.
- Comunidad activa: VirtualBox cuenta con una comunidad activa de usuarios y desarrolladores, lo que significa que hay una gran cantidad de recursos y soporte disponible en línea.

Arquitectura Cliente Servidor

“Trata sobre arquitectura de modelo Cliente - Servidor y su utilidad en la internet. Se explica la teoría asociada que hay detrás de este modelo.” [9][10]

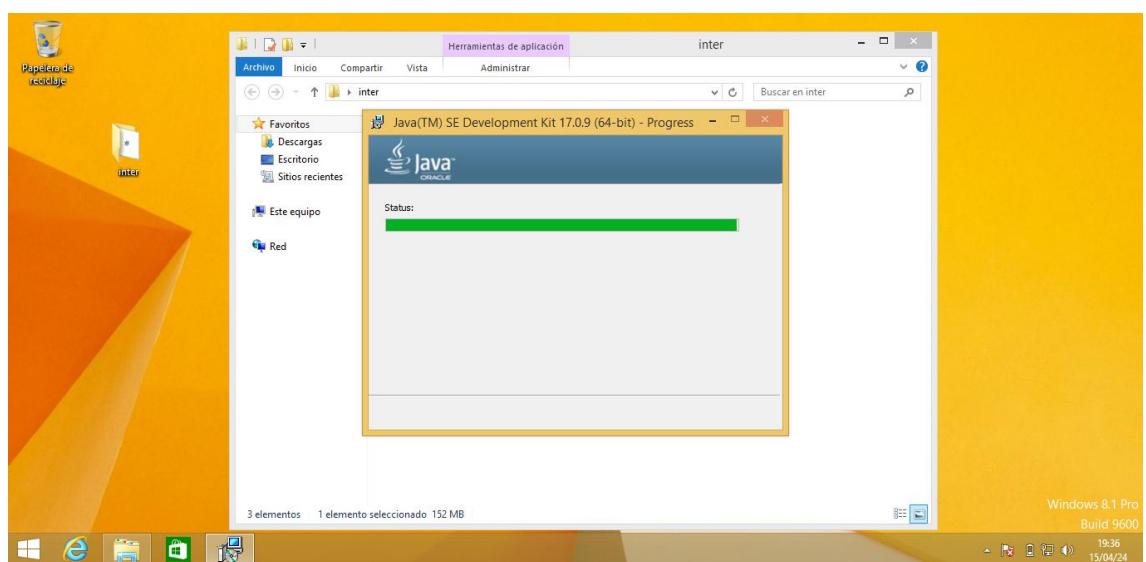
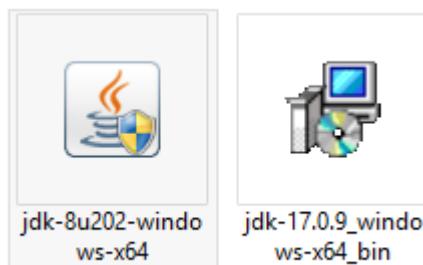
El Cliente y el Servidor.

- **Cliente:** Programa ejecutable que participa activamente en el establecimiento de las conexiones. Envía una petición al servidor y se queda esperando por una respuesta. Su tiempo de vida es finito una vez que son servidas sus solicitudes, termina el trabajo.
- **Servidor:** Es un programa que ofrece un servicio que se puede obtener en una red. Acepta la petición desde la red, realiza el servicio y devuelve el resultado al solicitante. Al ser posible implantarlo como aplicaciones de programas, puede ejecutarse en cualquier sistema donde exista TCP/IP y junto con otros programas de aplicación. El servidor comienza su ejecución antes de comenzar la interacción con el cliente.

Desarrollo

Como pasos previos para la ejecución de Netbeans procedemos a instalar los complementos siguientes:

Oracle y Java SE Developer Kit

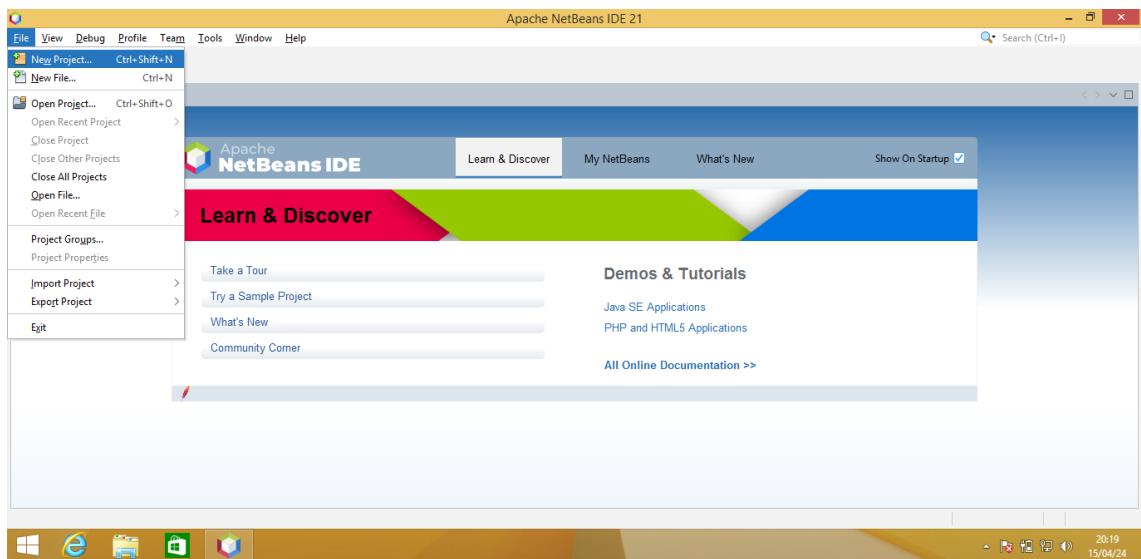


Una vez listos los complementos, instalamos Netbeans y lo ejecutamos

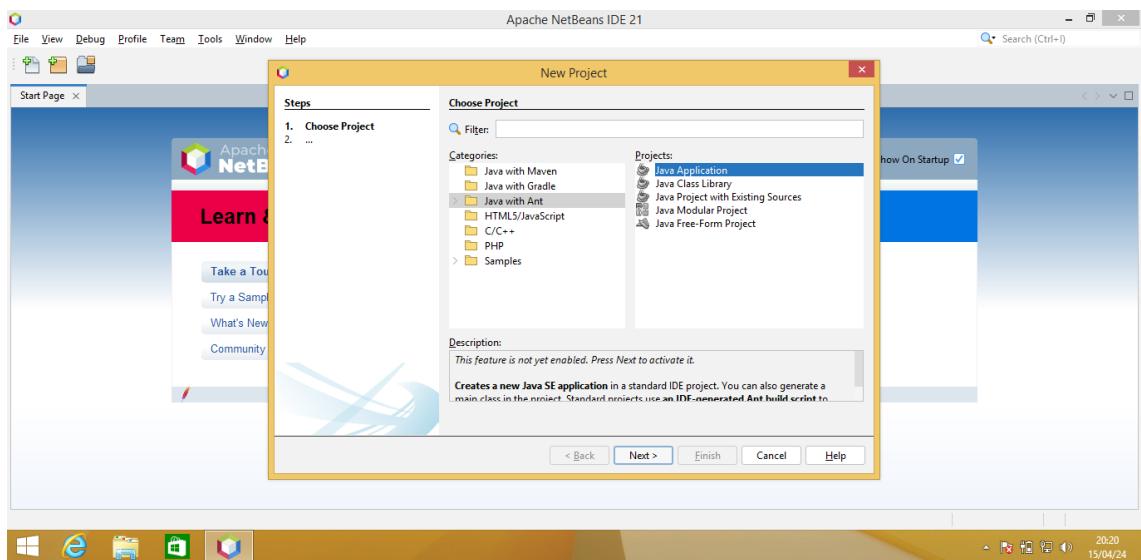


Para crear el programa vamos a ejecutar los siguientes pasos:

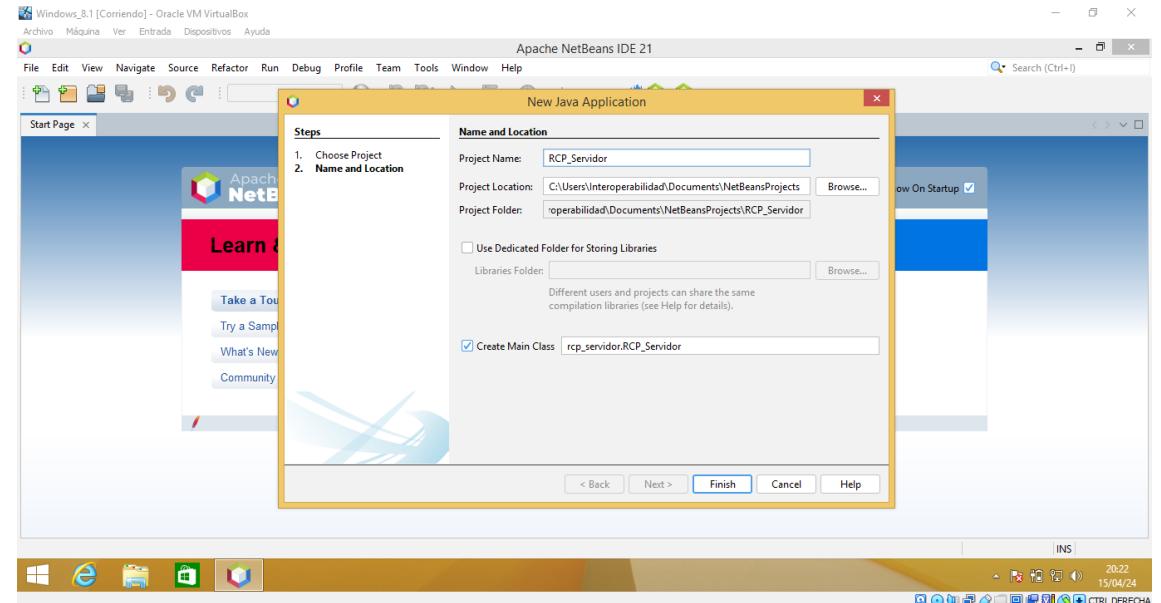
1. Crear un proyecto nuevo



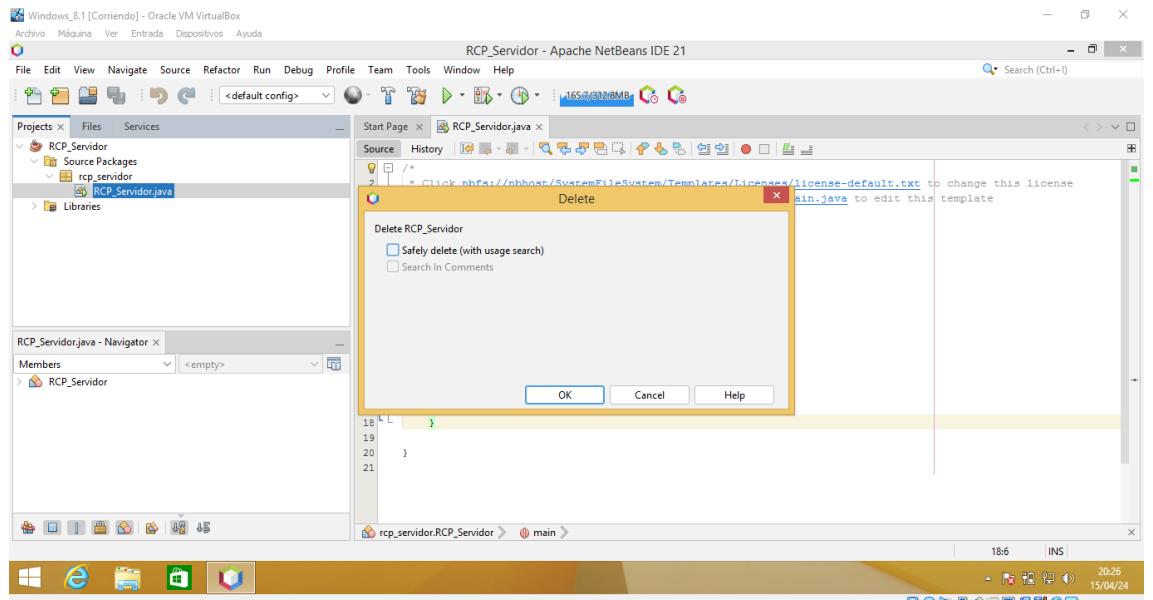
2. Se selecciona una aplicación en Java



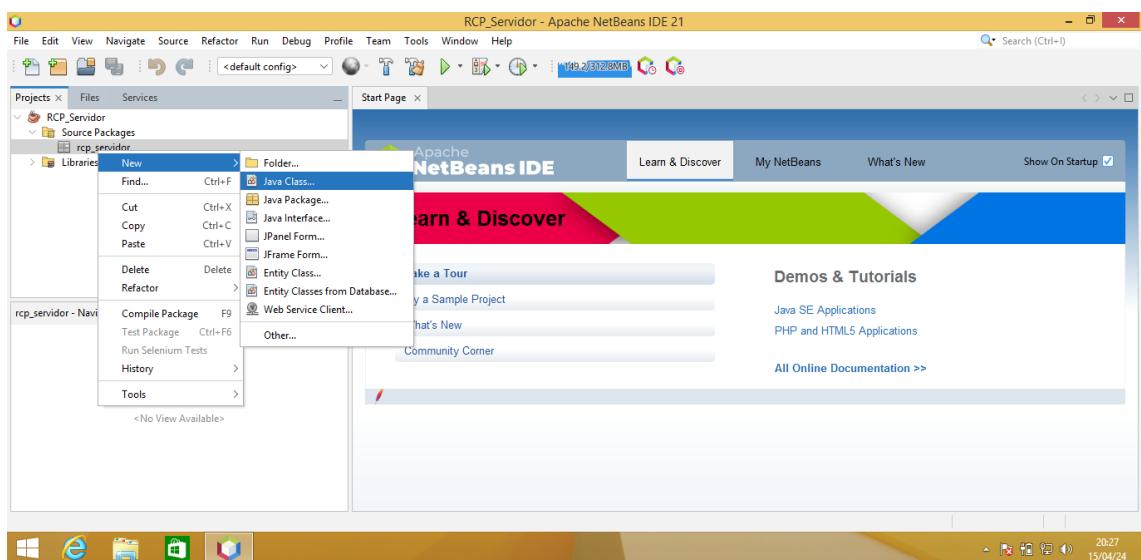
3. Designar el nombre “RCP_Servidor”



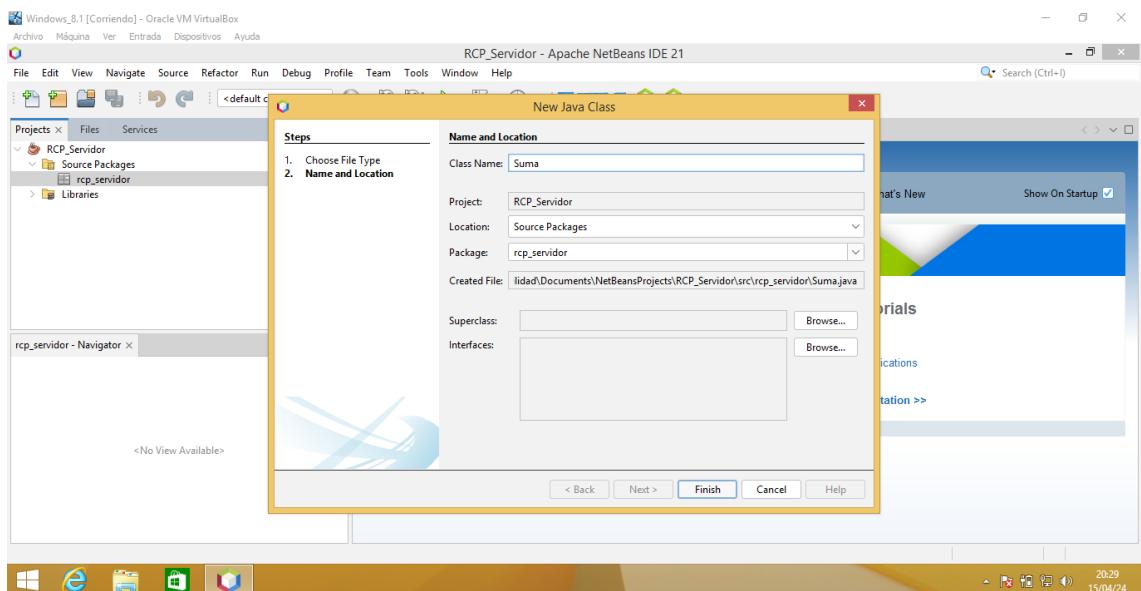
4. La clase que se crea tiene el método main asociado la cual por el momento se la eliminará sin seleccionar ninguna opción que aparece



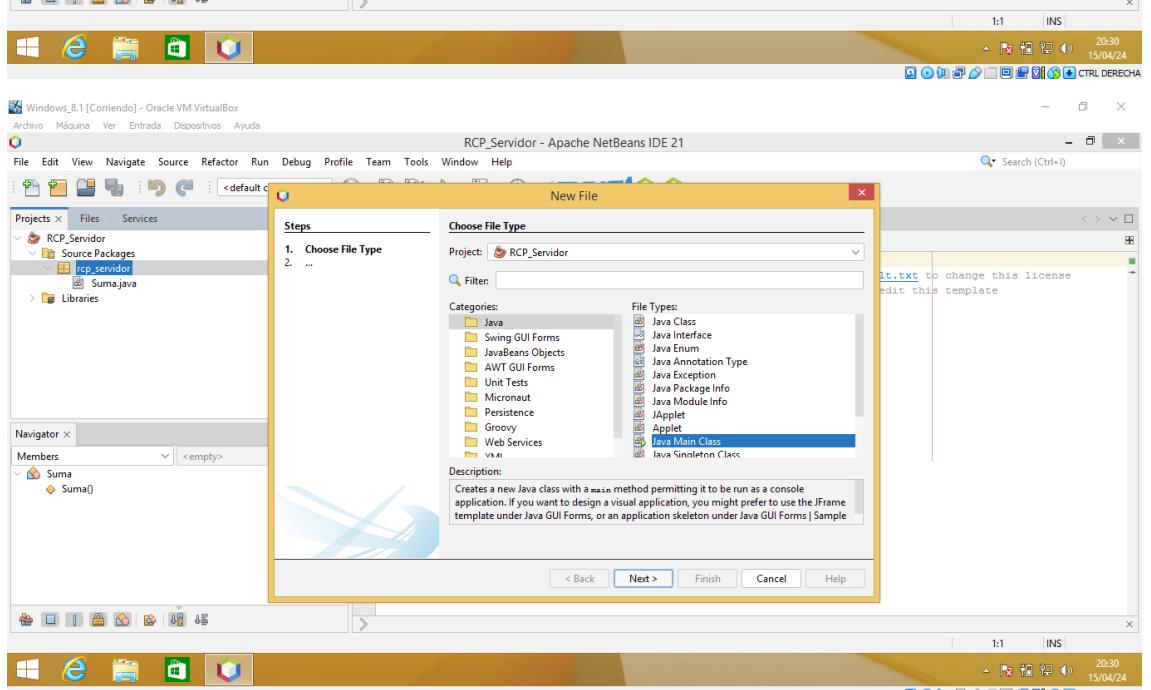
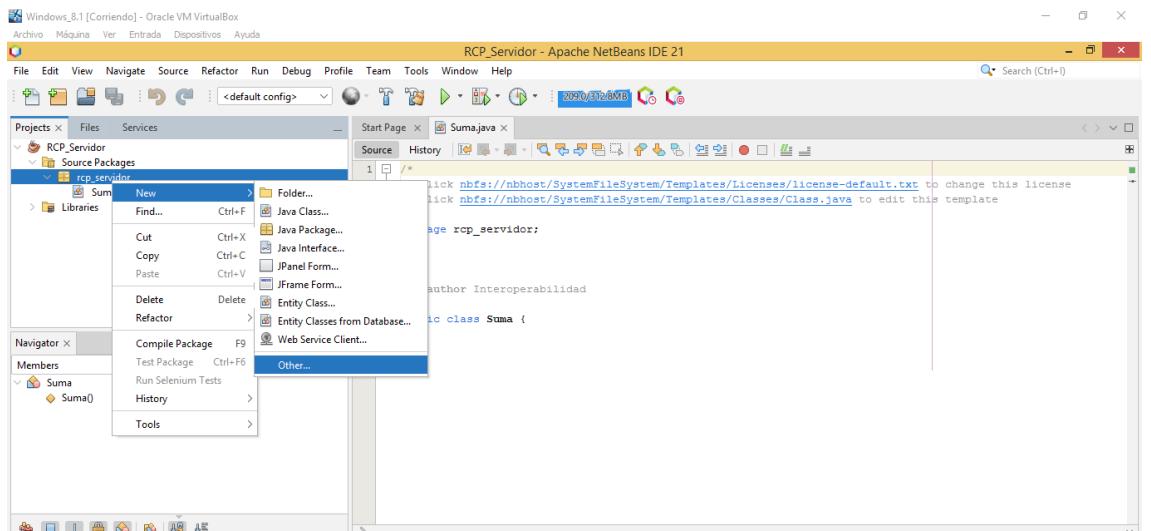
5. Crear una Java Class para cada una de las 4 operaciones básicas que se solicitan (Suma, resta, división y multiplicación)



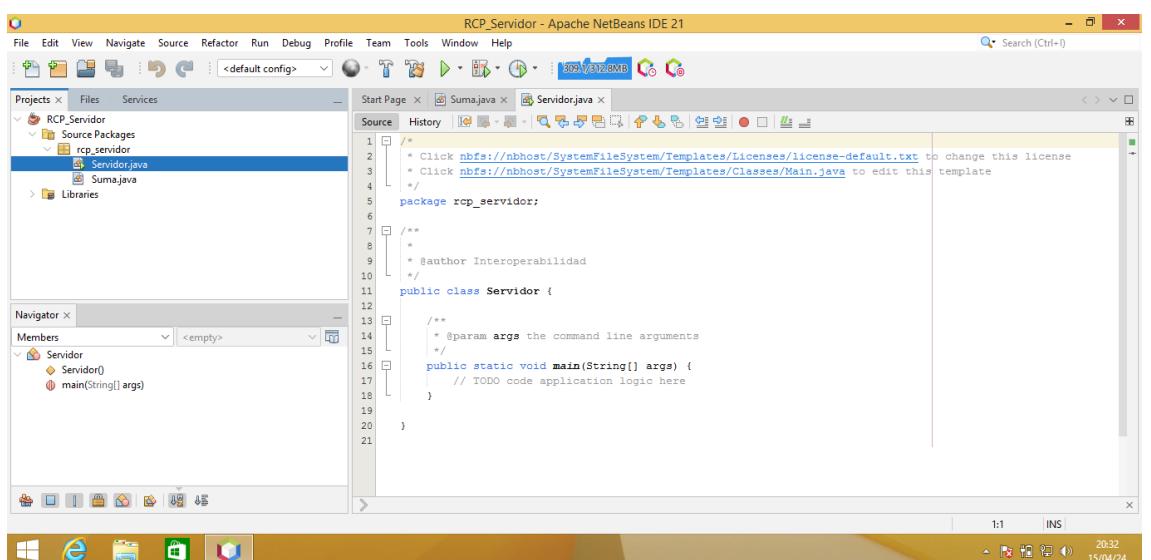
6. Se asigna el nombre a la clase dependiendo de la operación que se desea realizar. Se comienza con SUMA.



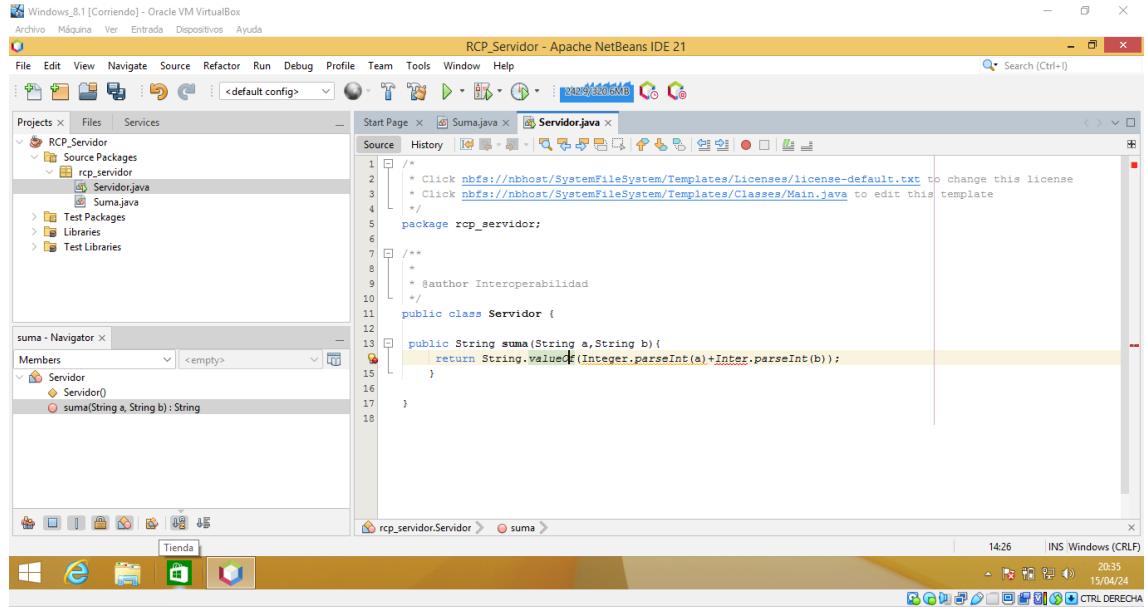
7. crear una Java main class y llamarle Servidor



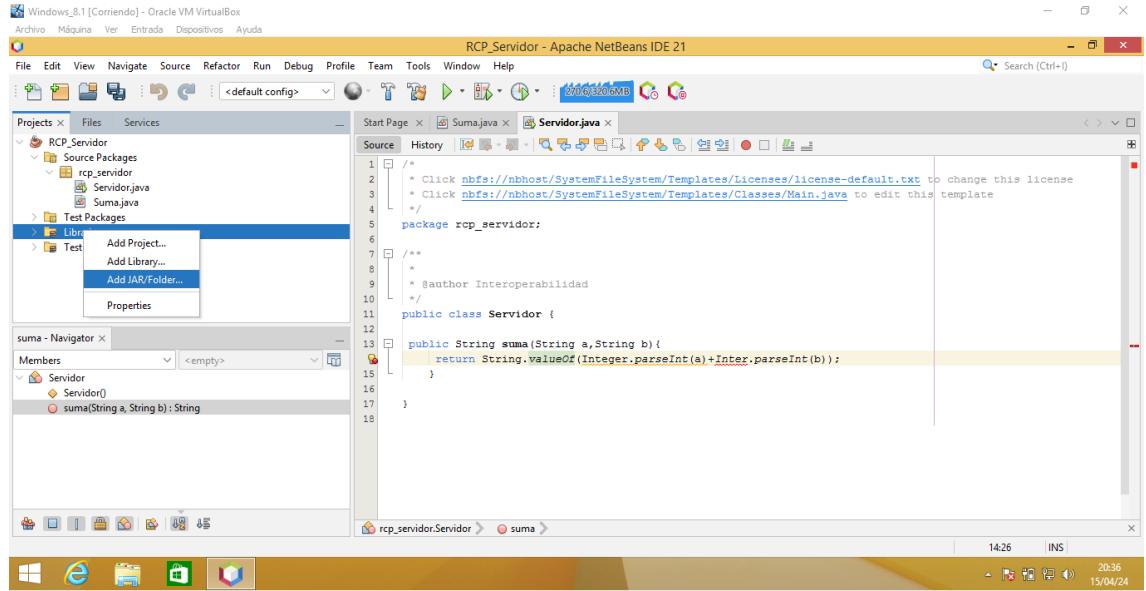
8. Le colocara el nombre “Servidor”, este no se replica



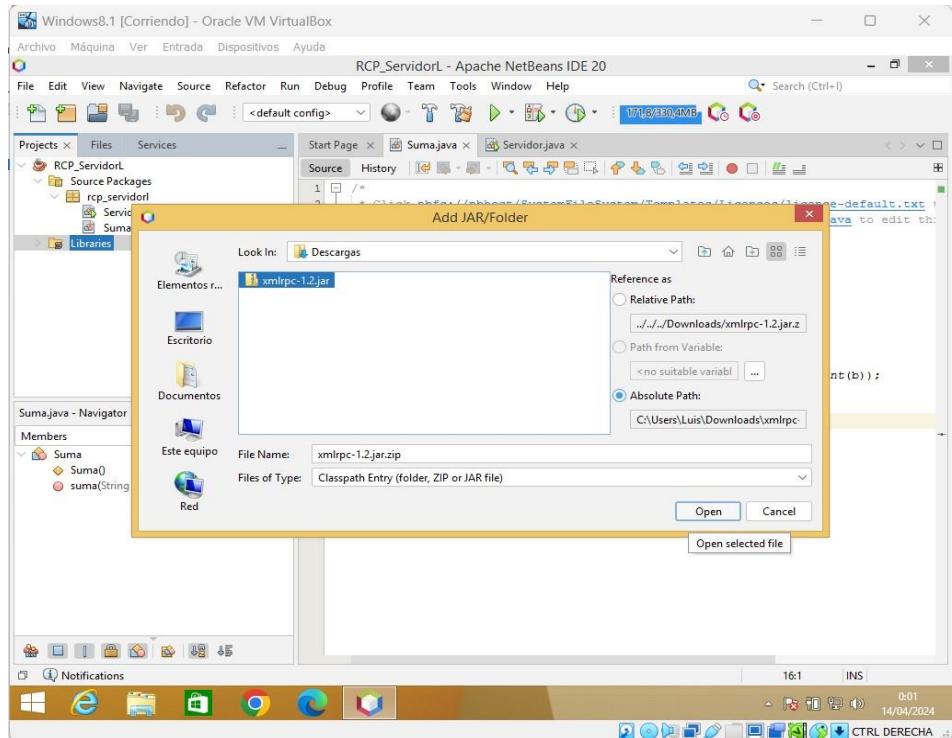
9. Programamos los códigos correspondientes para cada operación.



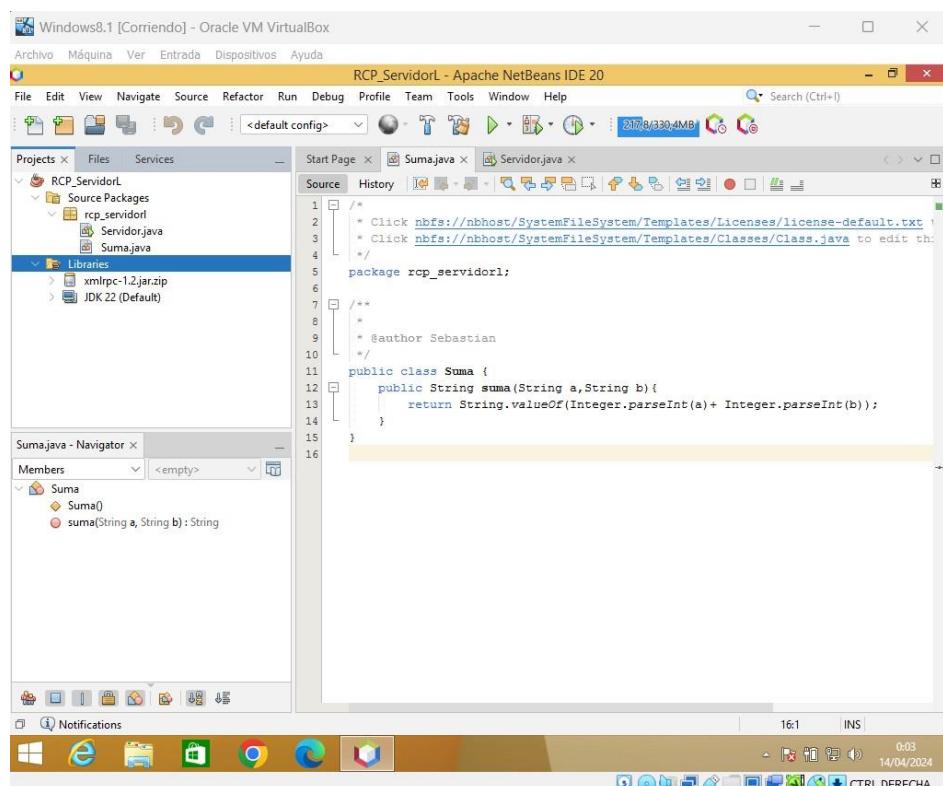
10. Añadir en archivo jar al proyecto (Jar Folder)



11. Buscar la carpeta que descargaron



12. Podemos observar que el archivo se encuentra de manera correcta



- Escribir el siguiente código dentro del RCP del Servidor

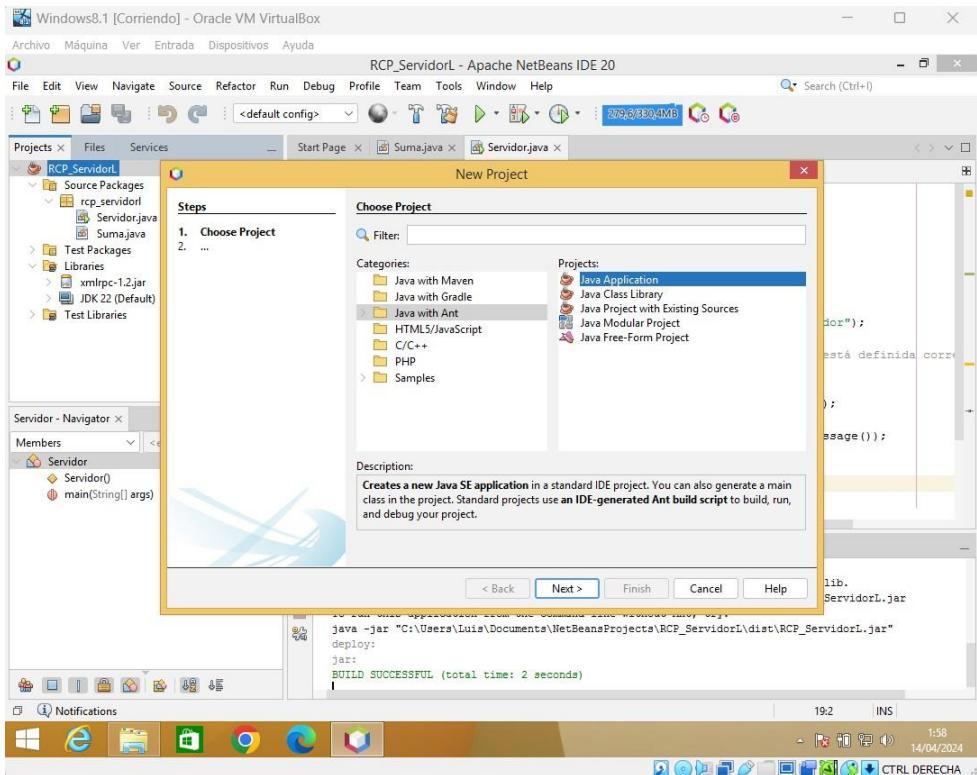
```

Windows8.1 [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
RCP_ServidorL - Apache NetBeans IDE 20
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config> 235.80.330.4MB Search (Ctrl+I)
Projects x Files Services
RCP_ServidorL
  Source Packages
    rcp_servidorL
      Servidor.java
      Sumajava
  Test Packages
  Libraries
    xmlrpc-1.2.jar
    JDK 22 (Default)
  Test Libraries
Start Page x Sumajava x Servidor.java x
Source History |<empty> 235.80.330.4MB
1 package rcp_servidorL;
2
3 import javax.swing.JOptionPane;
4 import org.apache.xmlrpc.WebServer;
5
6 public class Servidor {
7     public static void main(String[] args) {
8         try {
9             JOptionPane.showMessageDialog(null, "Iniciando el servidor");
10            WebServer server = new WebServer(8080);
11            Suma suma = new Suma(); // Asumiendo que la clase Suma está definida correctamente
12            server.addHandler("miServidorRpc", suma);
13            server.start();
14            JOptionPane.showMessageDialog(null, "Servidor en linea");
15        } catch (Exception e) {
16            JOptionPane.showMessageDialog(null, "Error: " + e.getMessage());
17        }
18    }
19
20
Output - RCP_ServidorL (cleanjar) x
Copying 1 file to C:\Users\Luis\Documents\NetBeansProjects\RCP_ServidorL\build
Copy libraries to C:\Users\Luis\Documents\NetBeansProjects\RCP_ServidorL\dist\lib.
Building jar: C:\Users\Luis\Documents\NetBeansProjects\RCP_ServidorL\dist\RCP_ServidorL.jar
To run this application from the command line without Ant, try:
java -jar "C:\Users\Luis\Documents\NetBeansProjects\RCP_ServidorL\dist\RCP_ServidorL.jar"
deploy:
jar:
BUILD SUCCESSFUL (total time: 2 seconds)
19:2 153 14/04/2024 CTRL DERECHA

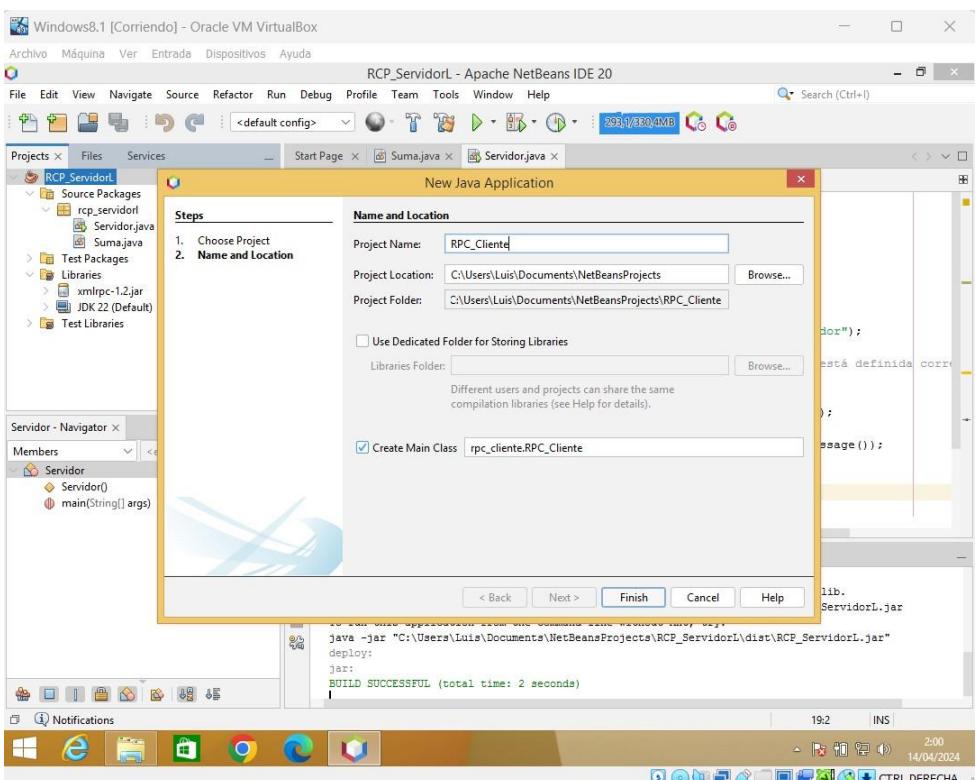
```

Ahora se trabaja sobre el proyecto cliente

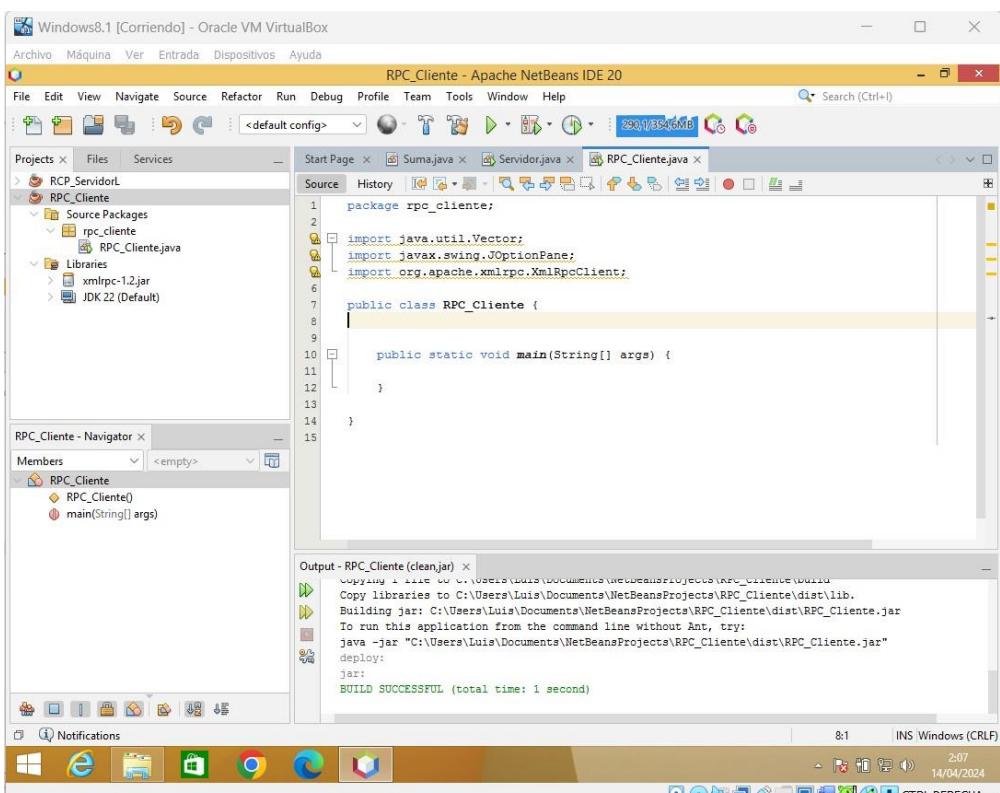
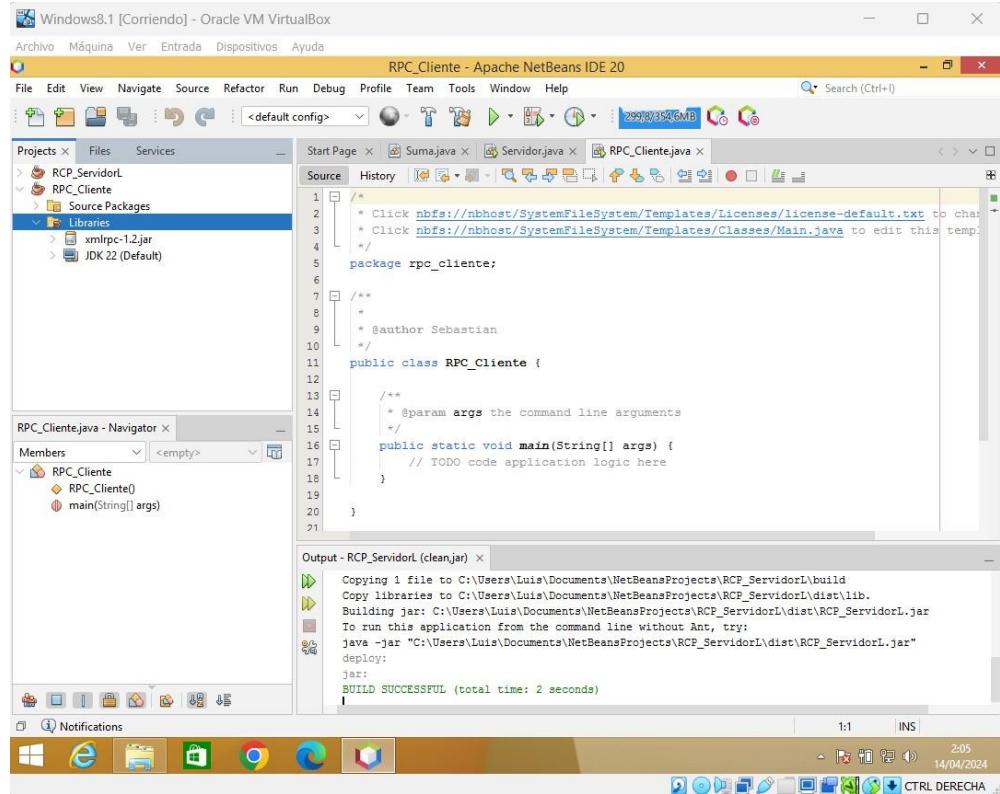
- Crear un nuevo Proyecto



- Crear el RPC_Cliente



- De la misma manera agregamos el archivo que anteriormente descargaron
- En la aplicación creada de igual forma añadir el paquete jar xmlrpc-1.2.jar



- En la clase importar las librerías que se muestran a continuación

- Escribimos el siguiente código dentro del main como se muestra a continuación

```

Windows8.1 (Comiendo) - Oracle VM VirtualBox
Archivo  M ásica  Ver  Entrada  Dispositivos  Ayuda
File Edit View Navigator Source Refactor Run Debug Profile Team Tools Window Help
RPC_Client - Apache NetBeans IDE 20
Projects  F iles Services
RPC_Servicio
RPC_Client
Source Packages
rpcliente
rpccliente
rpcclientejava
Test Packages
Libraries
JAR (2) (Default)
Test Libraries
Source History Start Page Sumar.java Sumar.java RPC_Client.java
Source rpcliente
1 package rpcliente;
2
3 import java.util.Vector;
4 import java.util.List;
5 import org.apache.xmlrpc.XmlRpcClient;
6
7 public class RPC_Cliente {
8
9     public static void main(String[] args) {
10
11         String x = "1" , y = "2";
12         System.out.println("Corregimos el signo de doble");
13
14         XmlRpcClient client = new XmlRpcClient("http://192.168.1.15:7000");
15         Vector<String> parametros = new Vector<String>();
16         OptionPanel opciones = new OptionPanel();
17
18         try {
19             String suma = client.execute("sumar", parametros);
20             JOptionPane.showMessageDialog(null, "suma de dos numeros"+suma);
21             JOptionPane.showMessageDialog(null, "El cliente se ha conectado");
22
23             System.out.println("El resultado es "+suma);
24
25             y = JOptionPane.showInputDialog(null, "Primer numero", JOptionPane.QUESTION_MESSAGE);
26             x = JOptionPane.showInputDialog(null, "Segundo numero", JOptionPane.QUESTION_MESSAGE);
27             parametros.addElement(x);
28             parametros.addElement(y);
29             resultados = client.execute("sumar", parametros);
30             int suma = Integer.parseInt(resultados.toString());
31             JOptionPane.showMessageDialog(null, "La suma es "+ suma);
32             JOptionPane.showMessageDialog(null, "El cliente se ha desconectado");
33
34             System.exit(0);
35         } catch (NumberFormatException e) {
36             JOptionPane.showMessageDialog(null, "Error: El valor ingresado no es un numero valido");
37         }
38     }
39 }
40
41 } catch (NumberFormatException e) {
42     JOptionPane.showMessageDialog(null, "Error: El valor ingresado no es un numero valido");
43 } catch (Exception e) {
44     JOptionPane.showMessageDialog(null, "Error: "+ e.getMessage());
45 }
46
47 }
48
49

```

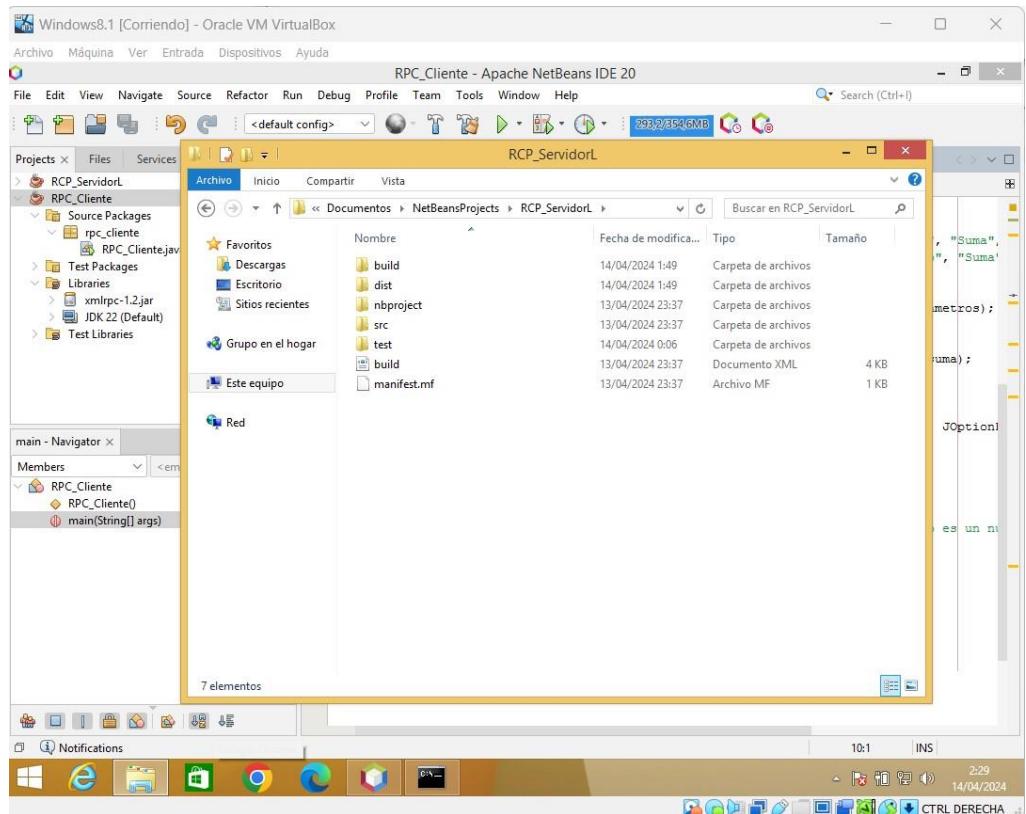
```

Windows8.1 (Comiendo) - Oracle VM VirtualBox
Archivo  M ásica  Ver  Entrada  Dispositivos  Ayuda
File Edit View Navigator Source Refactor Run Debug Profile Team Tools Window Help
RPC_Servicio
RPC_Client
Source Packages
rpccliente
rpccliente
rpcclientejava
Test Packages
Libraries
JAR (2) (Default)
Test Libraries
Source History Start Page rpcclientejava RPC_Server.java
Source rpccliente
1 package rpccliente;
2
3 import org.apache.xmlrpc.XmlRpcServer;
4 import org.apache.xmlrpc.XmlRpcServer.XmlRpcServer;
5 import org.apache.xmlrpc.XmlRpcServer.XmlRpcServer;
6
7 public class RPC_Server {
8
9     public static void main(String[] args) {
10
11         switch (menu) {
12             case "1":
13                 OptionPanel.showInputDialog(null, "Primer numero", JOptionPane.QUESTION_MESSAGE);
14                 y = OptionPanel.showInputDialog(null, "Segundo numero", JOptionPane.QUESTION_MESSAGE);
15                 parametros.addElement(x);
16                 parametros.addElement(y);
17                 resultados = client.execute("sumar", parametros);
18                 // Corregir el resultado
19                 int suma = Integer.parseInt(resultados.toString());
20                 JOptionPane.showMessageDialog(null, "La suma es "+ suma);
21                 JOptionPane.showMessageDialog(null, "El cliente se ha desconectado");
22                 parametros.clear();
23             break;
24
25             case "2":
26                 JOptionPane.showMessageDialog(null, "Saliendo", JOptionPane.PASSIVE_MESSAGE);
27                 System.exit(0);
28             break;
29
30             default:
31                 JOptionPane.showMessageDialog(null, "Opcion invalida");
32             break;
33         }
34     }
35
36     } catch (NumberFormatException e) {
37         JOptionPane.showMessageDialog(null, "Error: El valor ingresado no es un numero valido");
38     } catch (Exception e) {
39         JOptionPane.showMessageDialog(null, "Error: "+ e.getMessage());
40     }
41
42 }
43
44
45
46
47
48
49

```

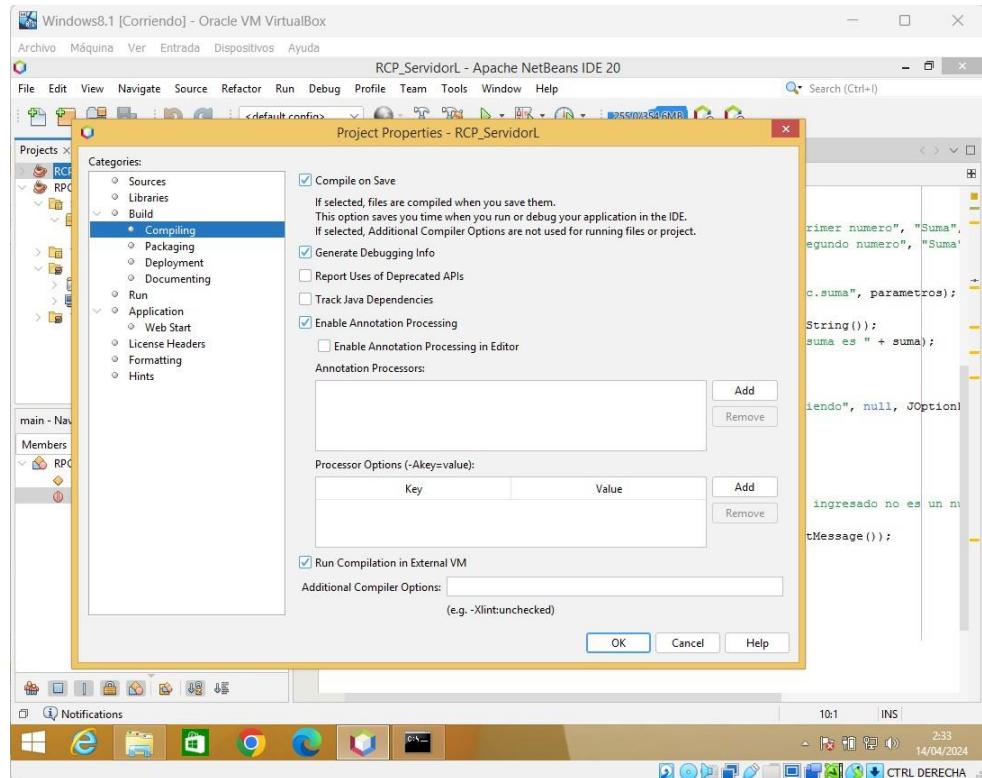
- Para que pueda ser distribuible al proyecto buscamos en la carpeta donde se crearon.

Vamos a configurar el servidor.

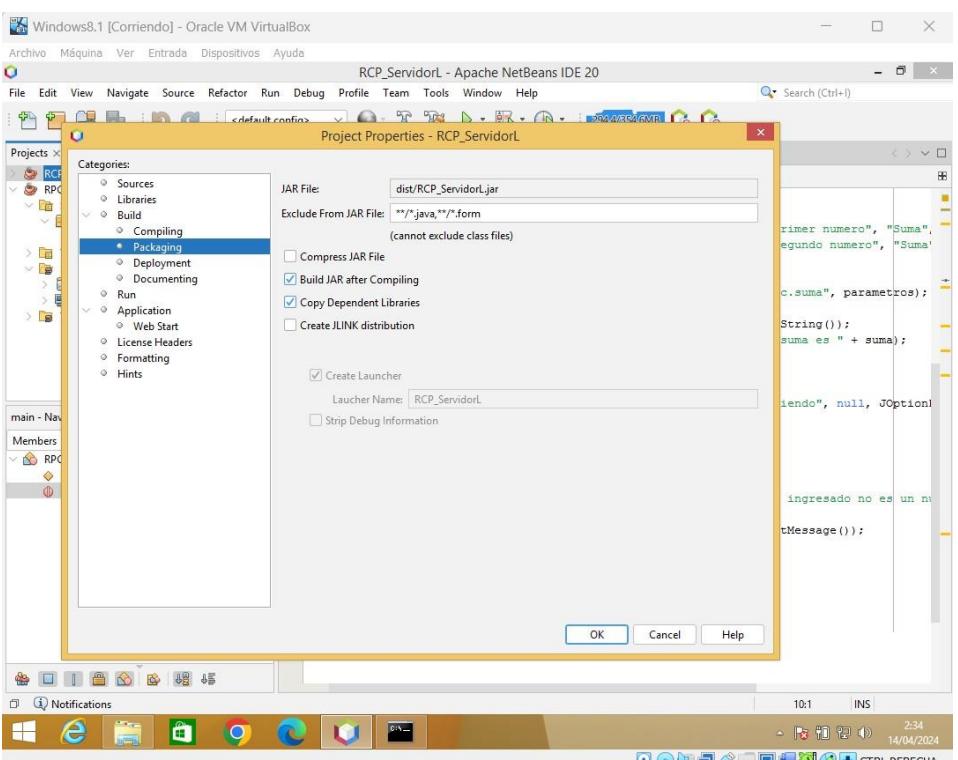


- Para distribuir el proyecto, necesitas la carpeta "dist" donde se genera el archivo JAR. Si no la encuentras, ve a las propiedades del proyecto y activa "Compilar en la carpeta dist". (A partir del paso 25 hasta el 28 son opciones)

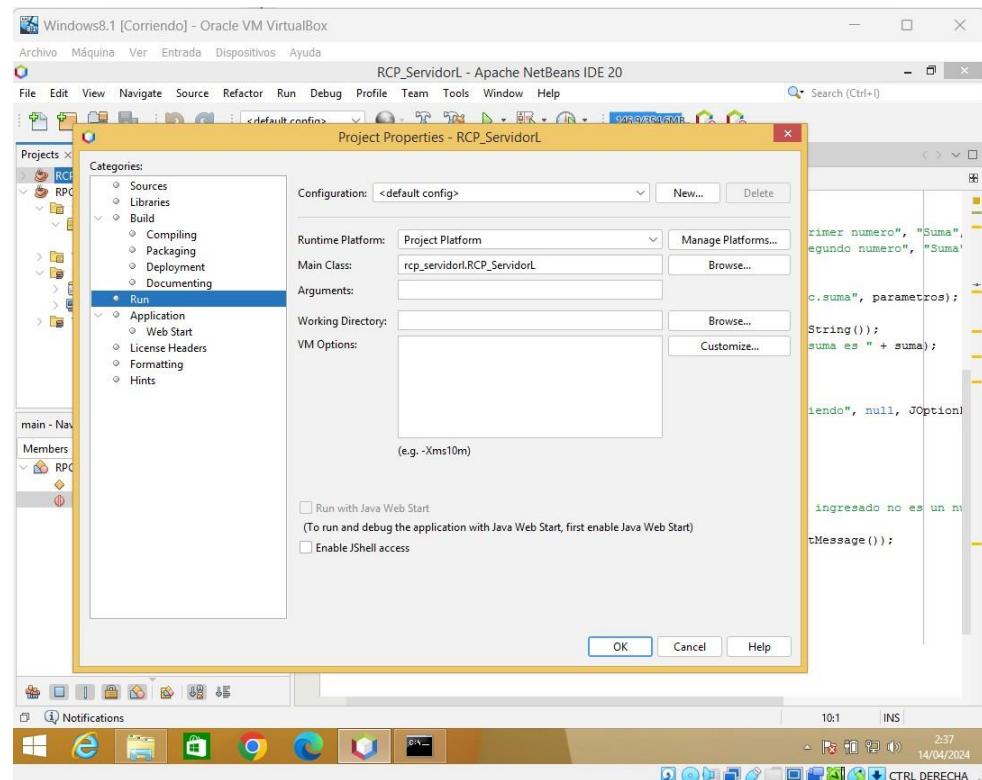
Señalamos todo en Compiling



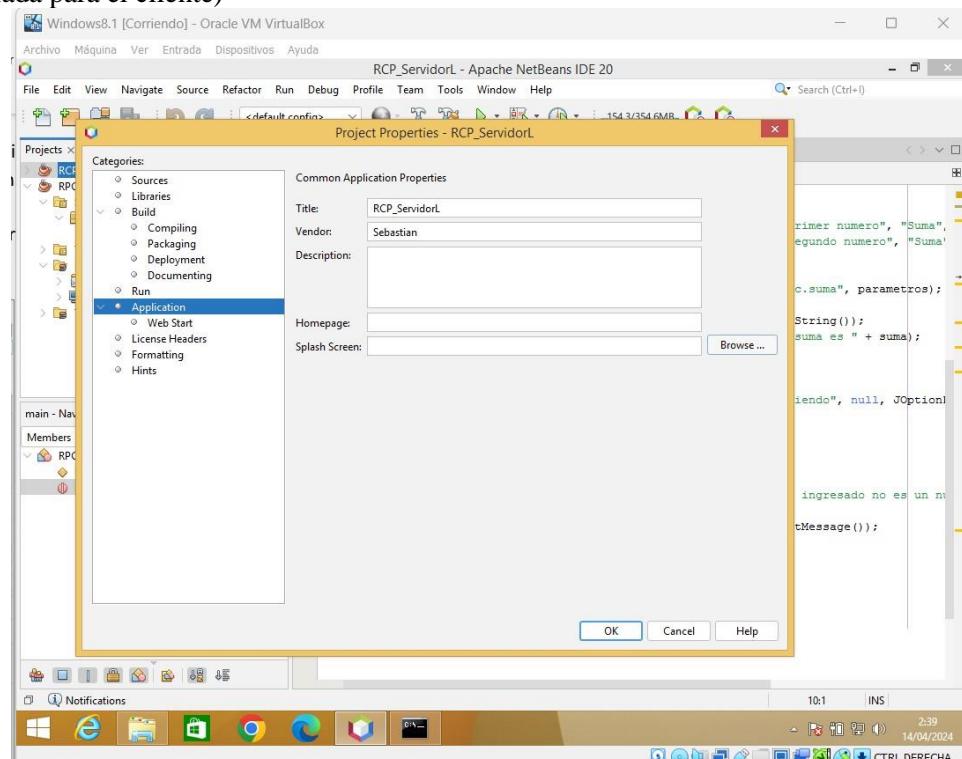
Señalamos todo en Packing



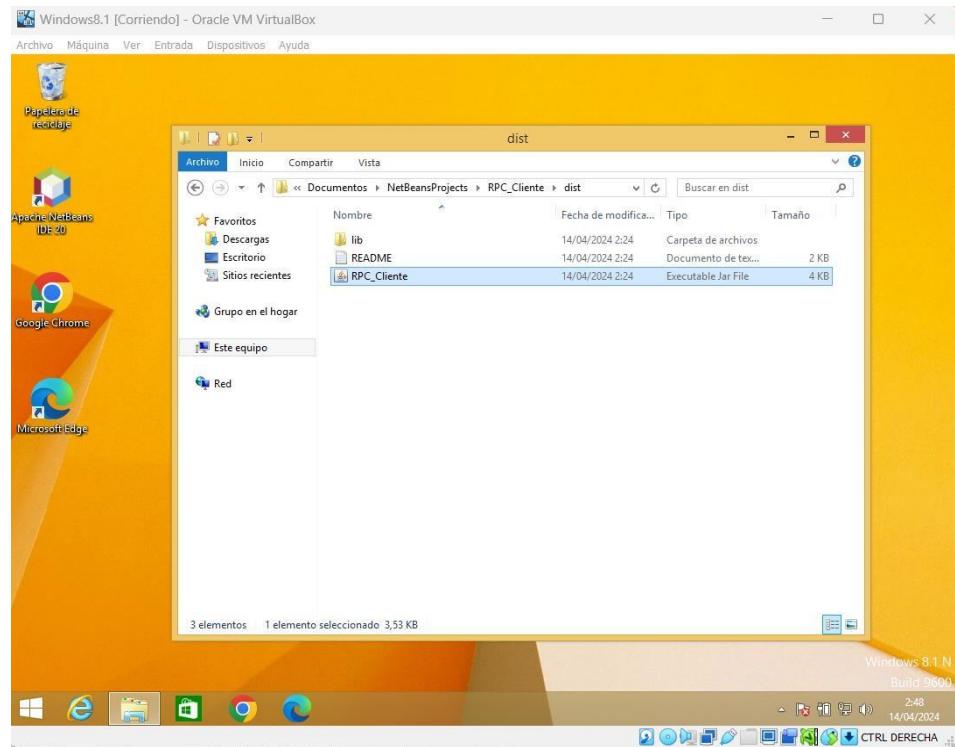
- Para que se genere el run debe estar señalada la clase



- En aplicación se puede dar el nombre de la aplicación a distribuir y una imagen asociada a través de splash screen (Buscar una imagen en Google y asociarla al programa)
- Repetir la configuración para el archivo de distribución de cliente (Buscar una imagen apropiada para el cliente)



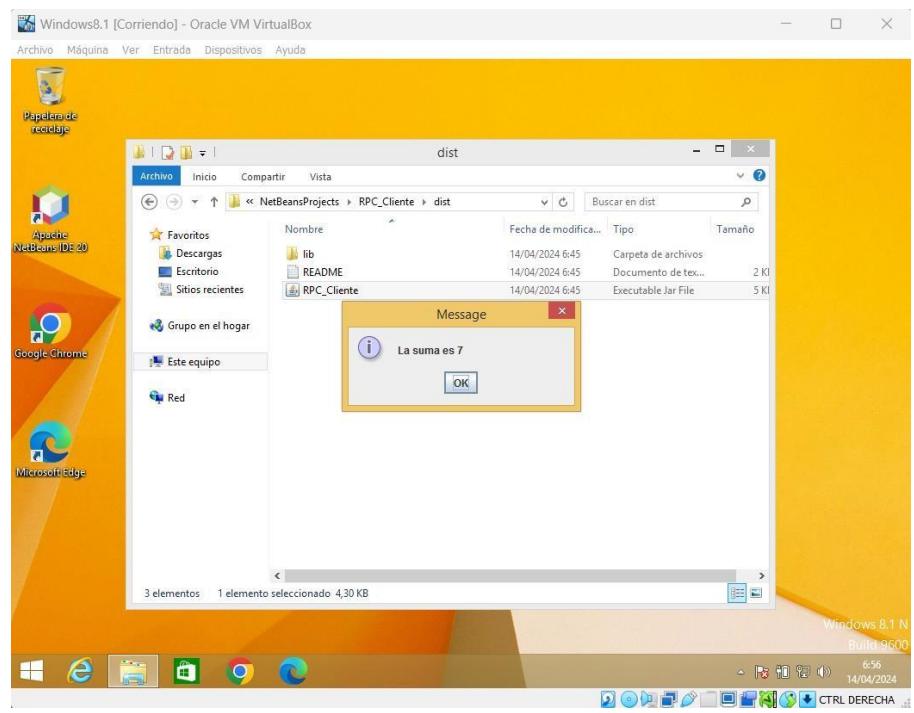
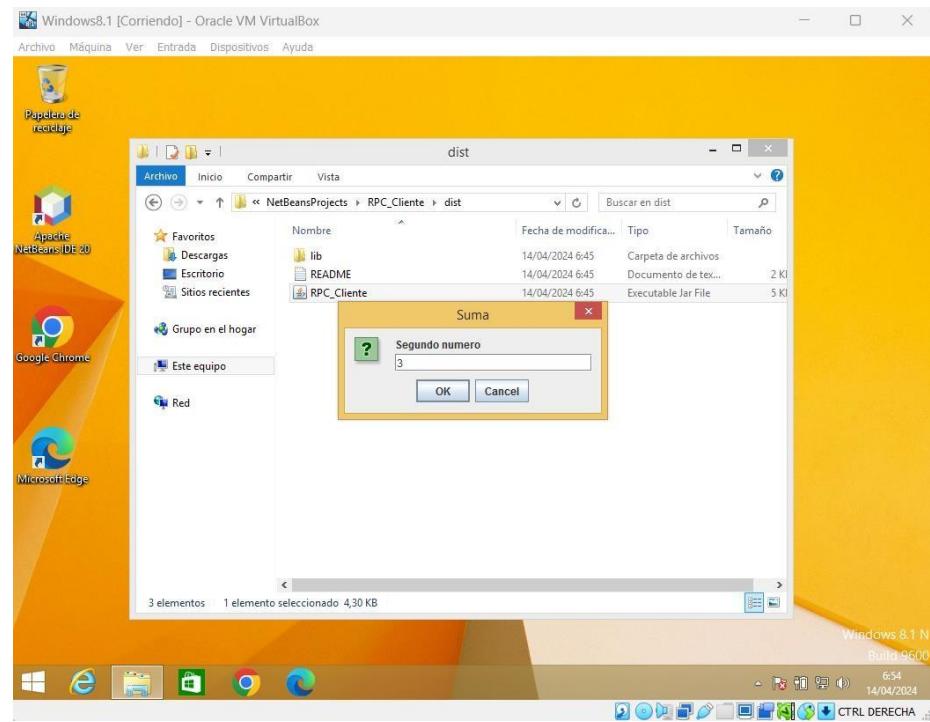
- Se genera el archivo jar distribuible. Si se da un doble click se inicia el servidor.



➤ Resultados

Una vez desarrollado y configurado el programa se ejecutará el mismo y se desplegará una ventana emergente con la calculadora diseñada, brindándonos las opciones de la operación correspondiente (Suma, Resta,Multiplicación, División).





Conclusión

NetBeans facilita la creación, prueba y mantenimiento de aplicaciones que se extienden por diversos nodos, permitiendo que trabajen en conjunto de manera armoniosa.

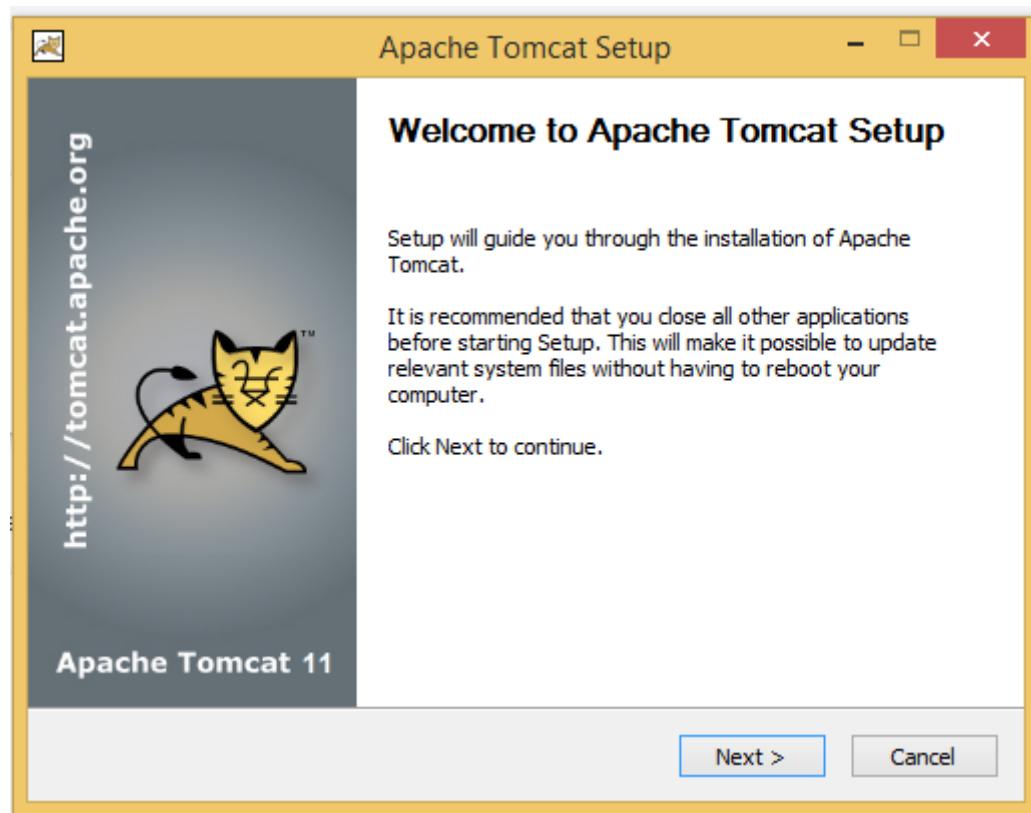
NetBeans no solo es una herramienta, sino un facilitador que transforma la programación distribuida en una sinfonía coherente de código. Así, alejarse de NetBeans, te embarcas en un viaje donde la complejidad se convierte en oportunidad y la coordinación entre componentes distribuidos se convierte en música para el éxito de tu aplicación.

Referencias:

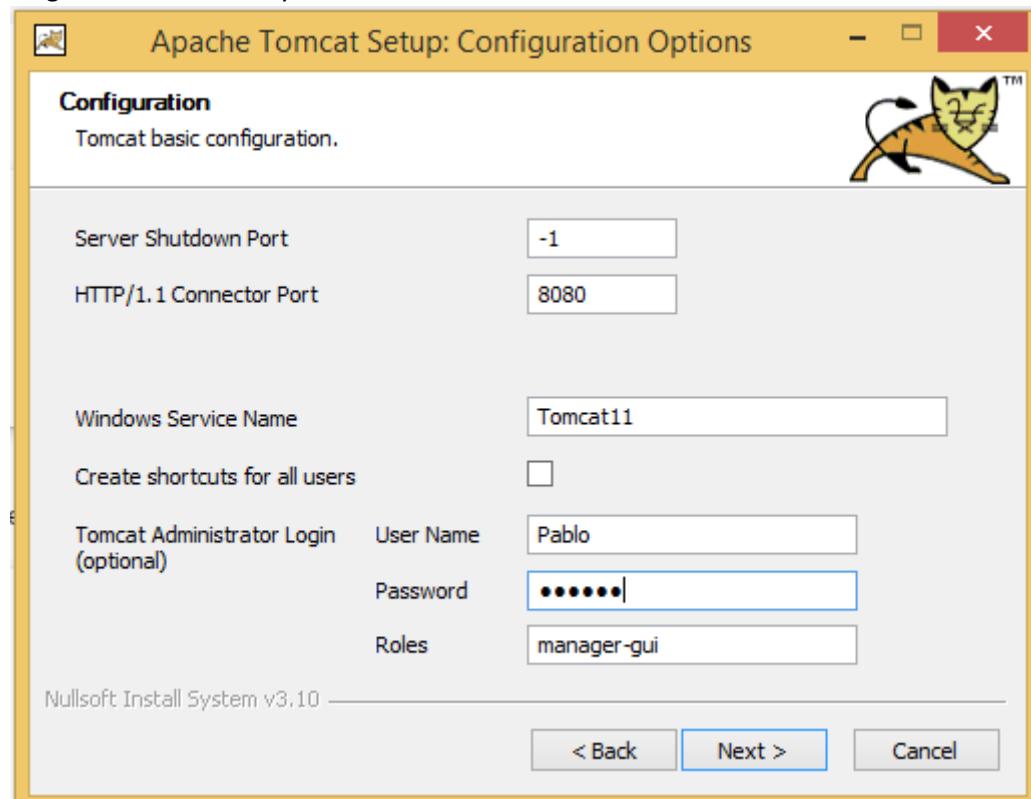
- P. Keegan, L. Champenois, G. Crawley, C. Hunt y C. Webster, "Netbeans™ ide field guide: developing desktop, web, enterprise, and mobile applications," Prentice Hall Press, 2005.
- N. Solop, "Qué es una máquina virtual," Recuperado de [https://www.wetcom.com/blog/blog-1/post/que-es-una-maquina-virtual-parte-1-204](https://www.wetcom.com/blog/blog-1/post/que-es-una-maquina-virtual-parte-1- 204), s.f.
- O. Lizama, G. Kindley, J. J. Morales y A. Gonzales, "Redes de computadores Arquitectura Cliente-Servidor," Universidad Tecnica Federico Santa Maria, 2016.
- L. Castedo, L. Dávila, R. González, M. Hernando Gutiérrez, S. López, P. Quesada et al., "Arquitectura cliente-servidor para un laboratorio remoto," 1970.
- D. Santo Orcero, "VirtualBox: virtualización para el escritorio," Todo Linux: la revista mensual para entusiastas de GNU/Linux, no. 106, pp. 35-39, 2009.
- P. J. V. Angulo, D. G. G. Herrera, S. E. M. Clerque y J. C. E. Álvarez, "Virtualbox como estrategia de enseñanza aprendizaje en la asignatura de soporte técnico," CIENCIAMATRIA, vol. 6, no. 3, pp. 498-519, 2020.
- M. Padilla, "VirtualBox: un sistema operativo dentro de otro," Todo Linux: la revista mensual para entusiastas de GNU/LINUX, no. 94, pp. 29-33, 2008.

Informa Java

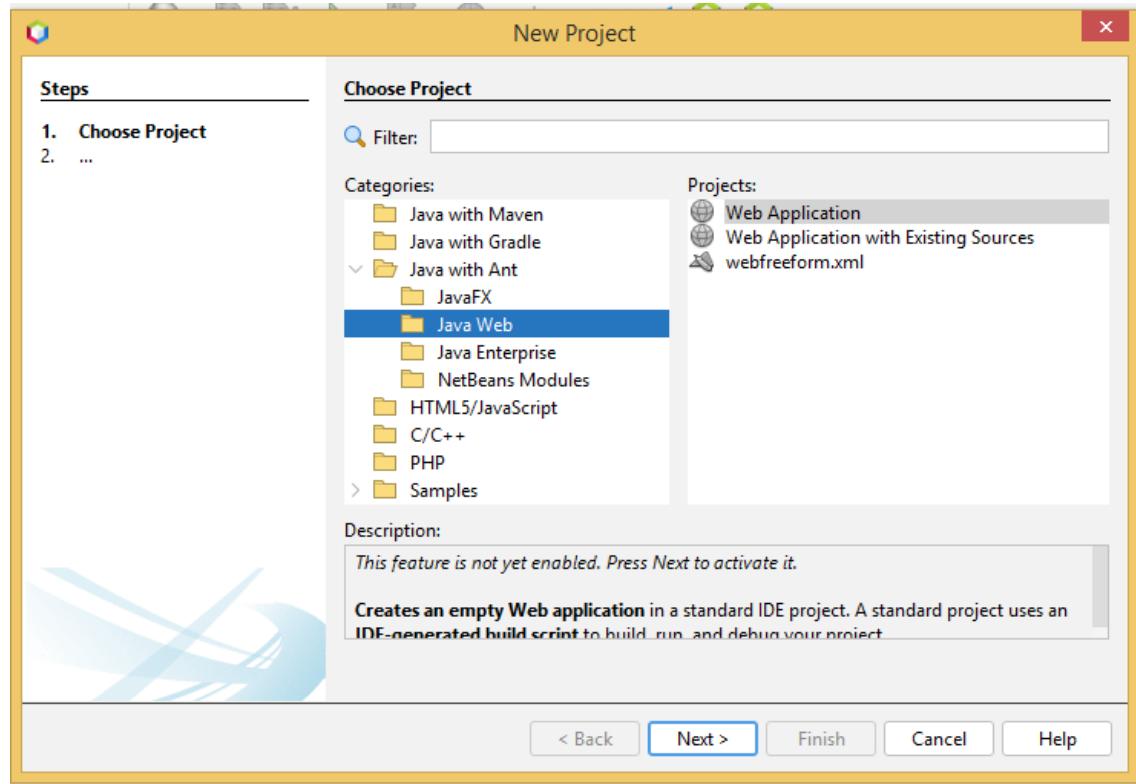
1.



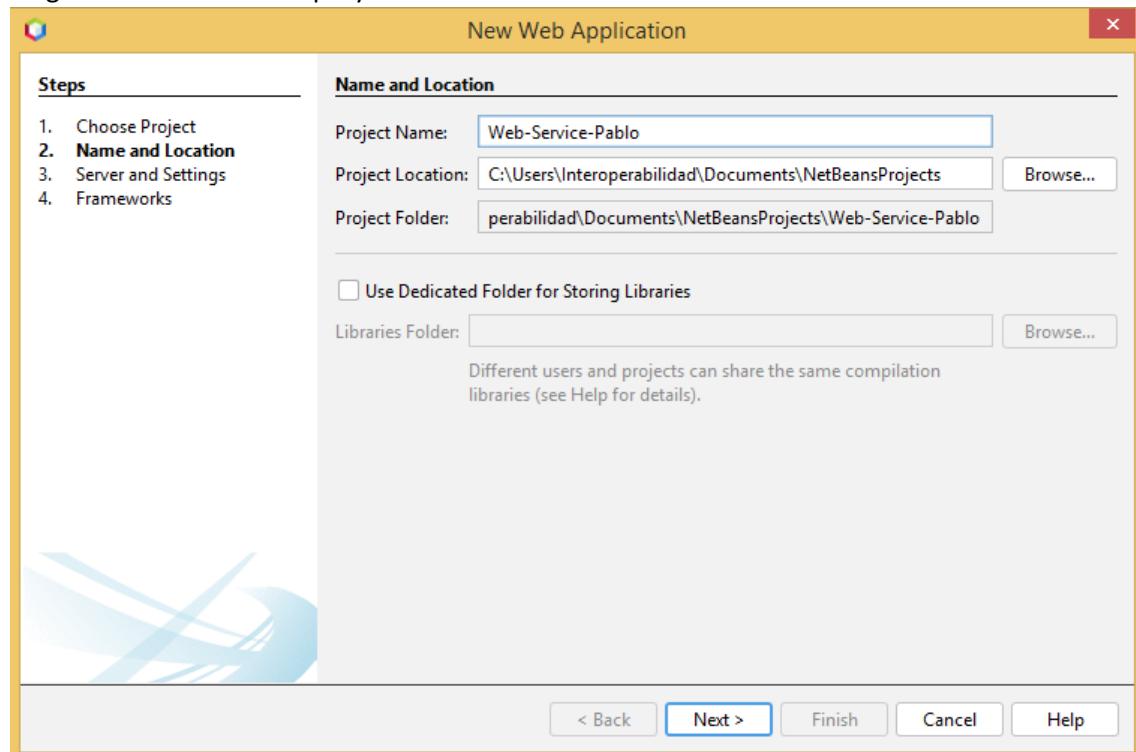
1. Asignamos un usuario y una contraseña



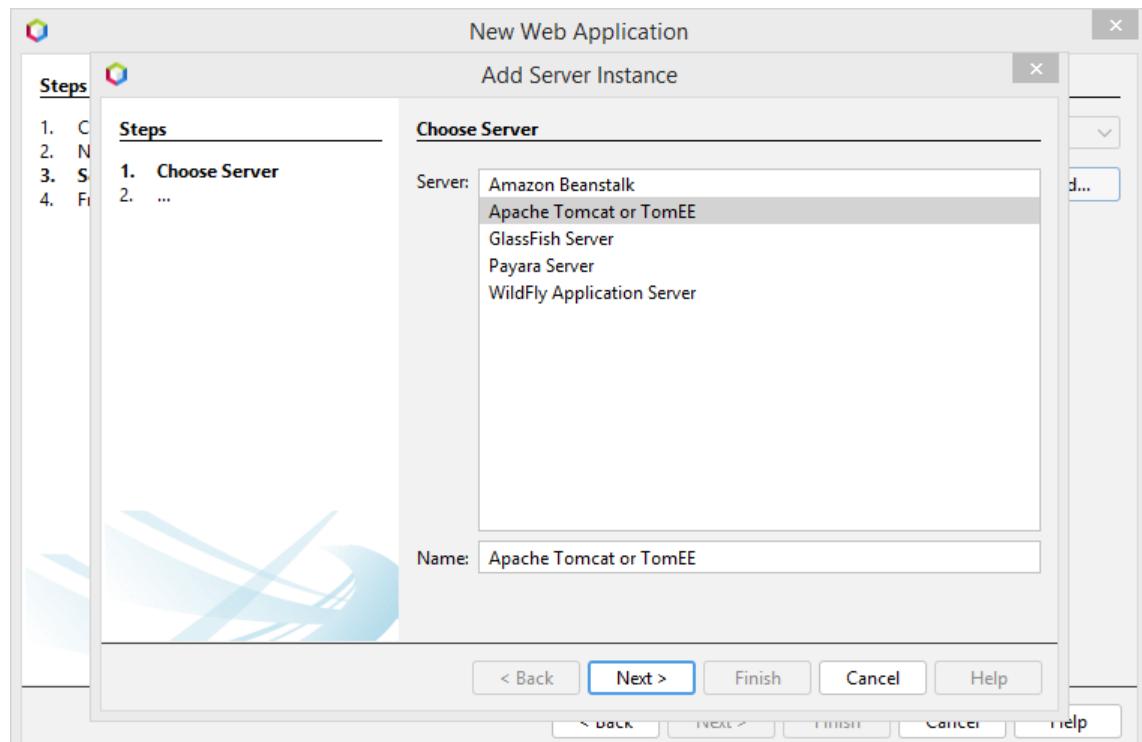
2. Creamos un nuevo proyecto



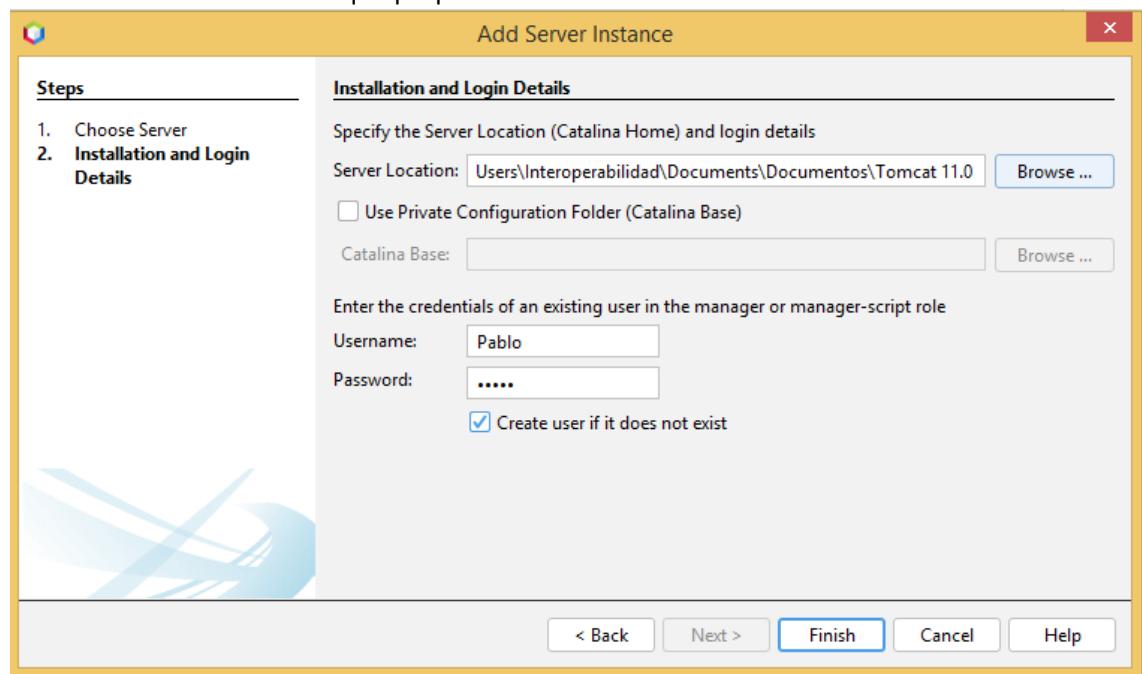
3. Asignamos un nombre al proyecto



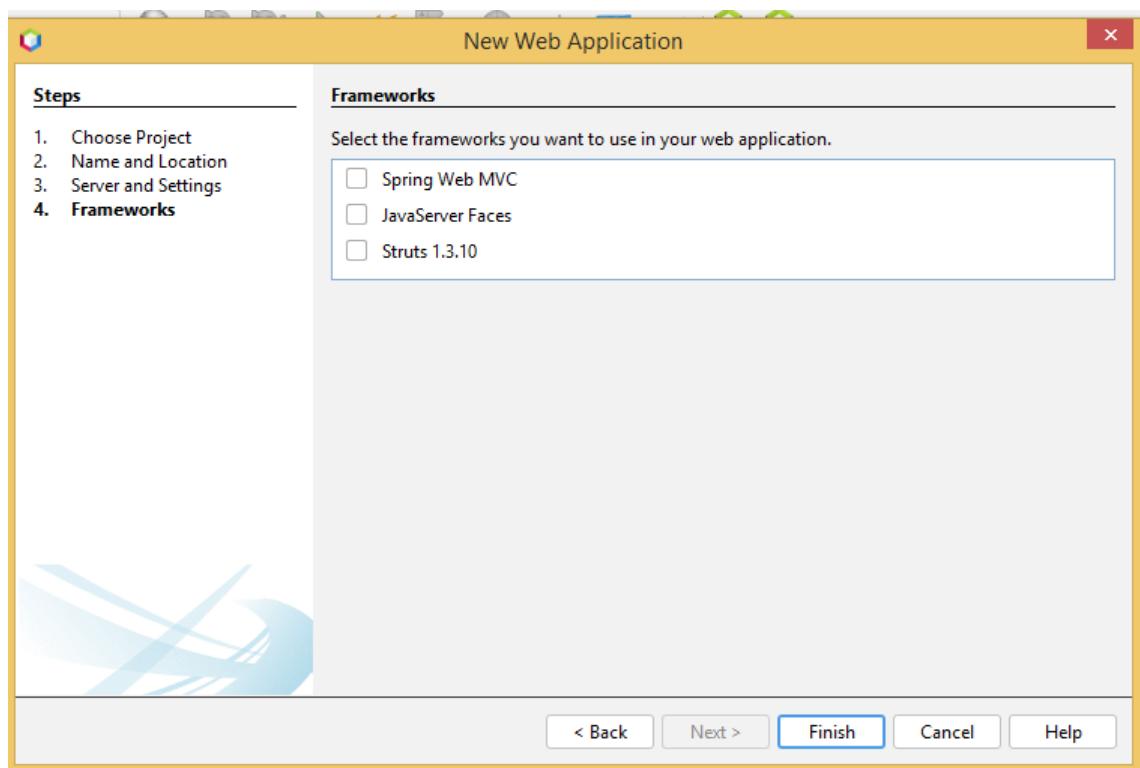
4. Seleccionamos TomCat en nueva instancia



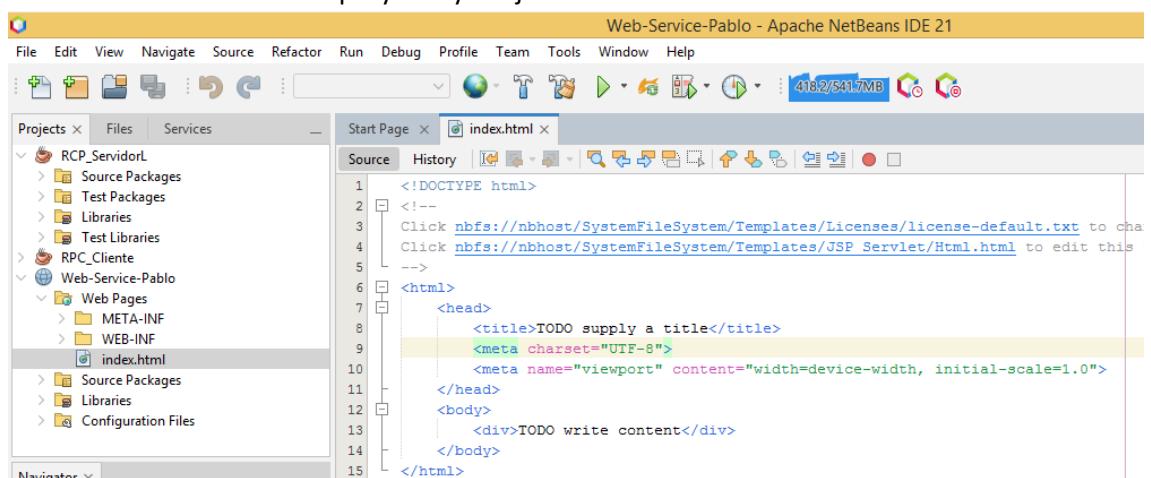
5. Colocamos las credenciales que proporcionamos en la instalación



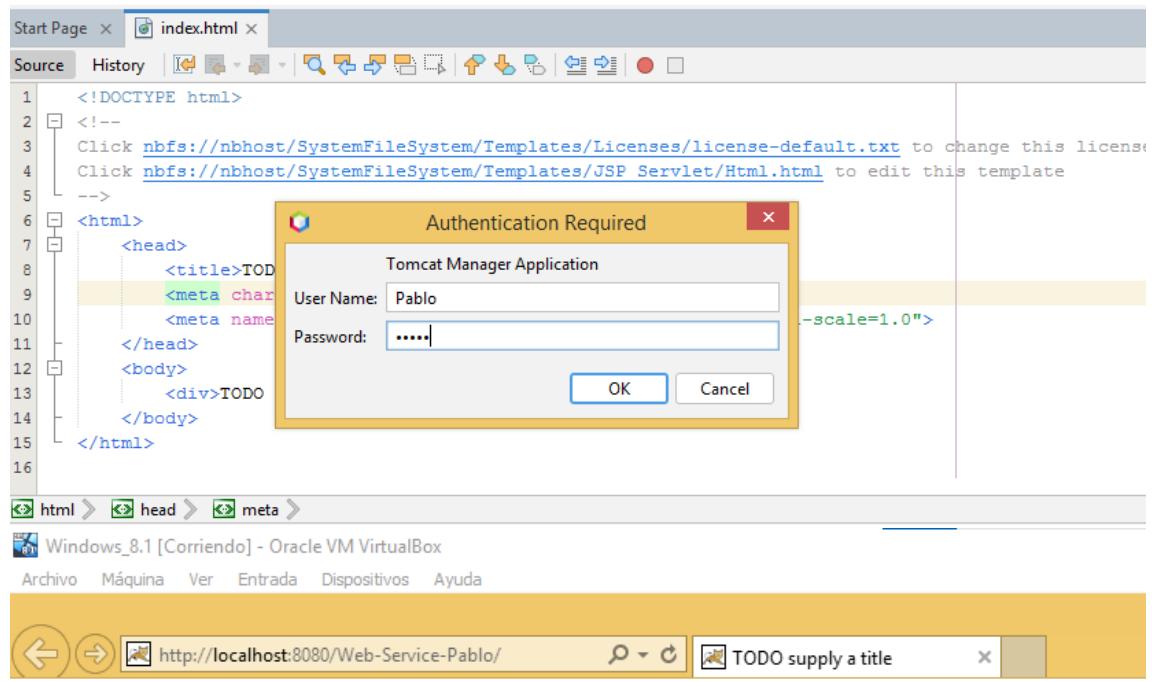
6. Finalizamos la configuración



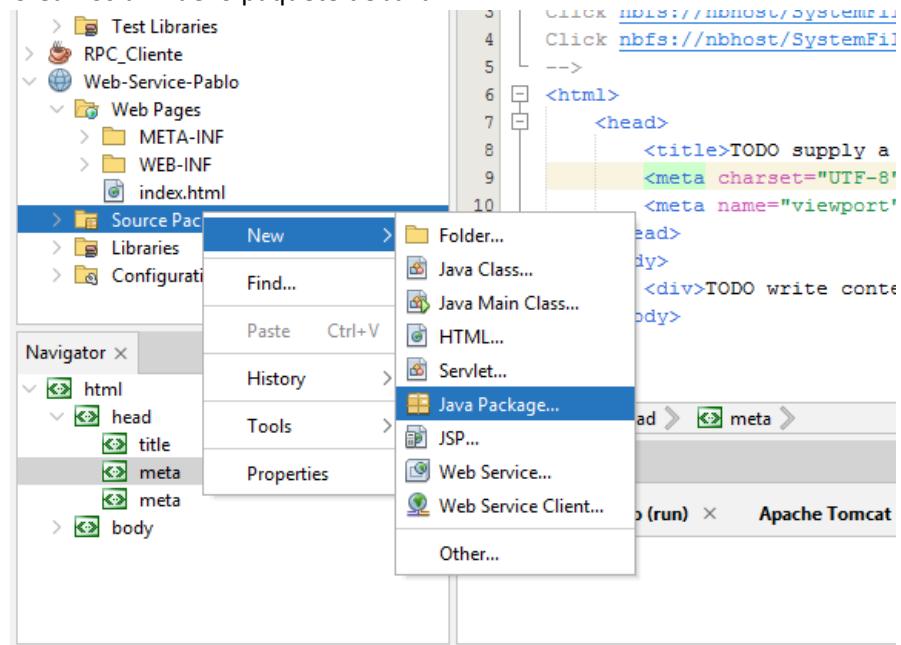
7. Abrimos el index del nuevo proyecto y lo ejecutamos

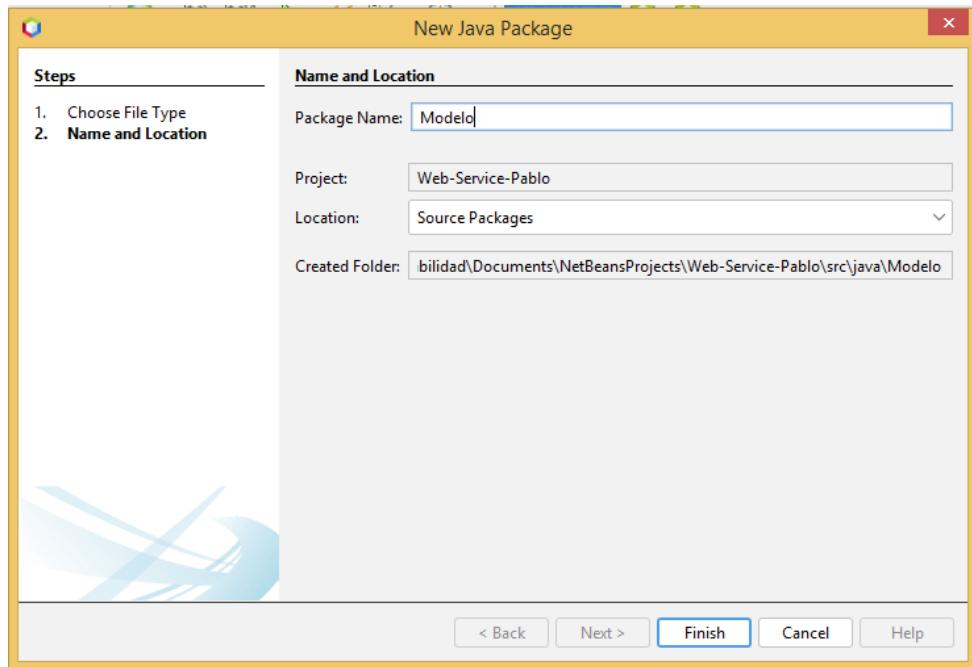


8. Nos autenticamos

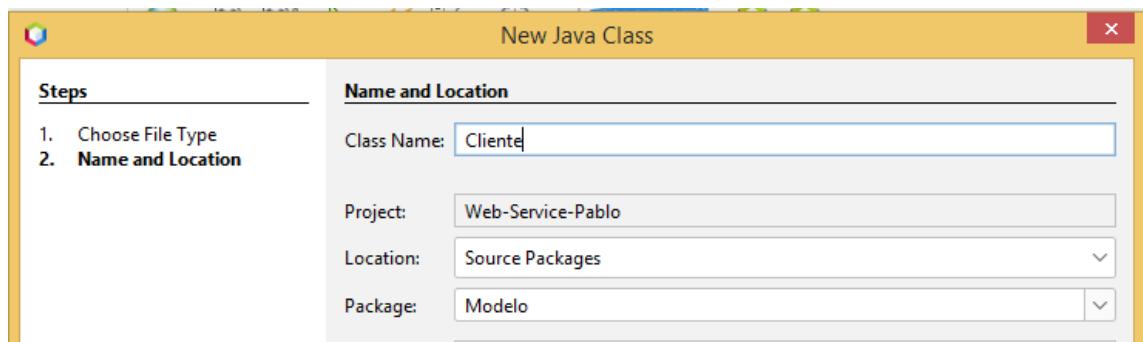
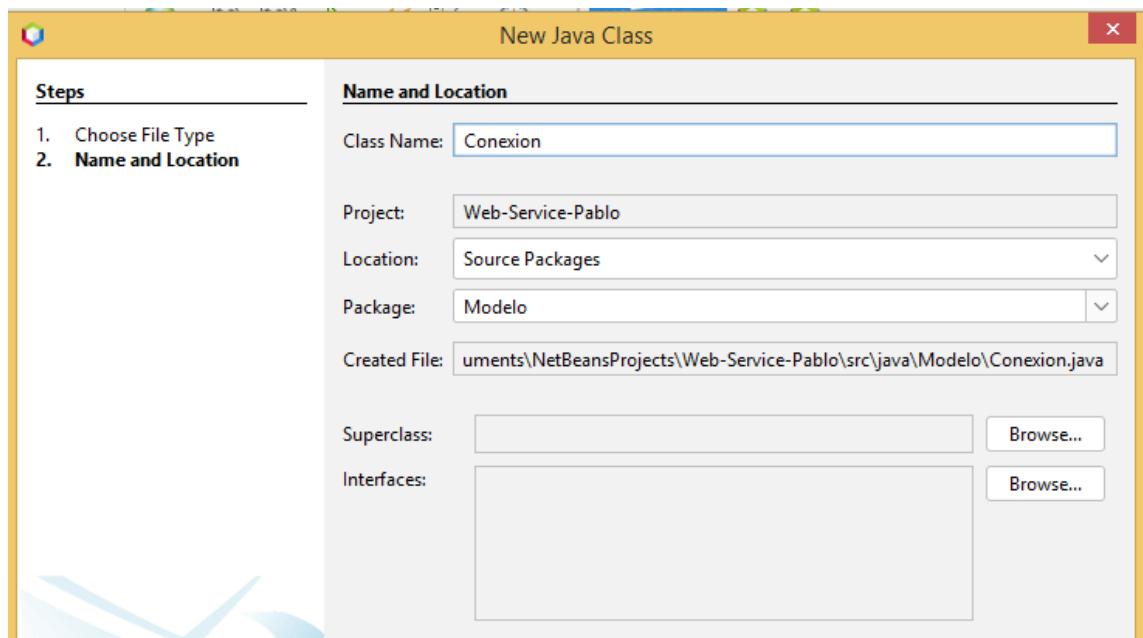


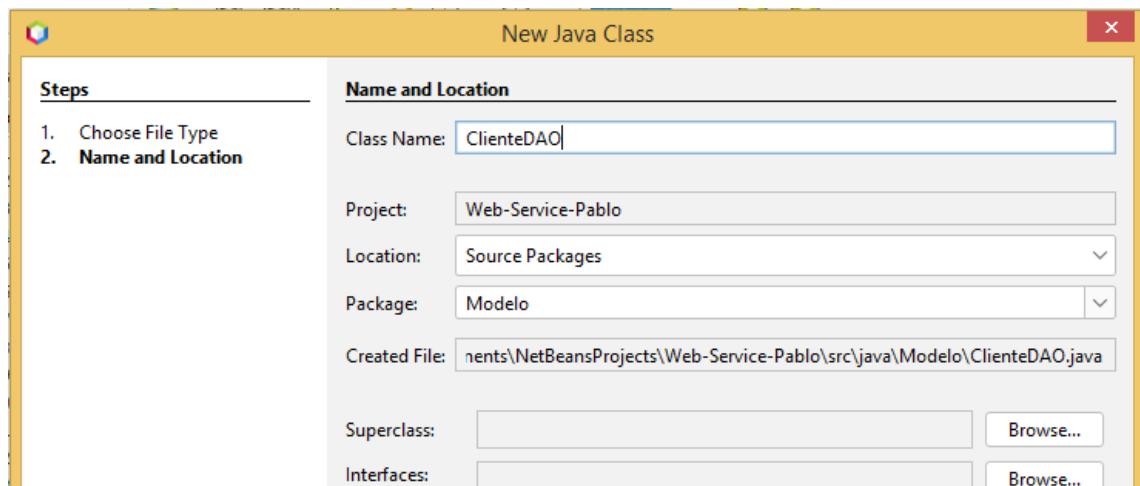
9. Creamos un nuevo paquete de Java



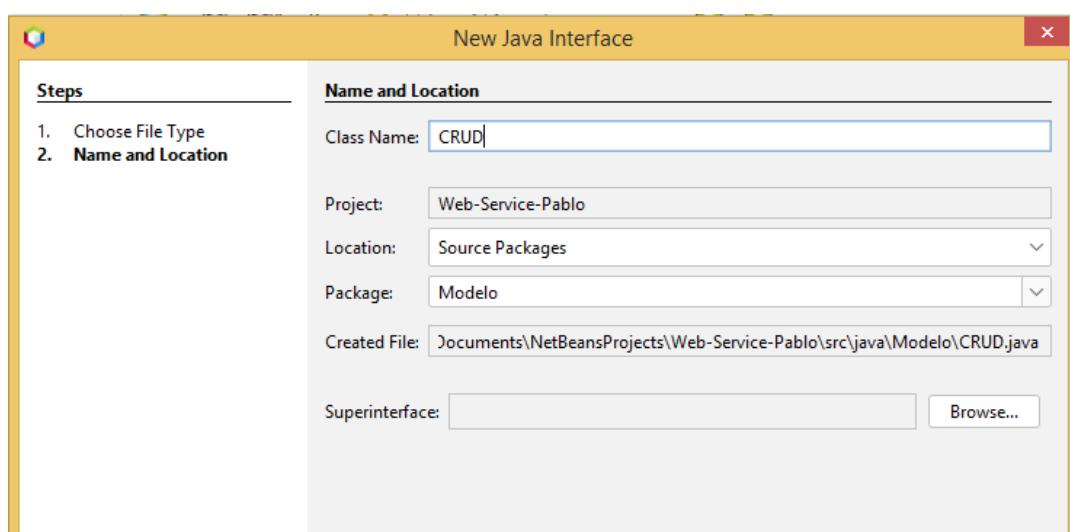
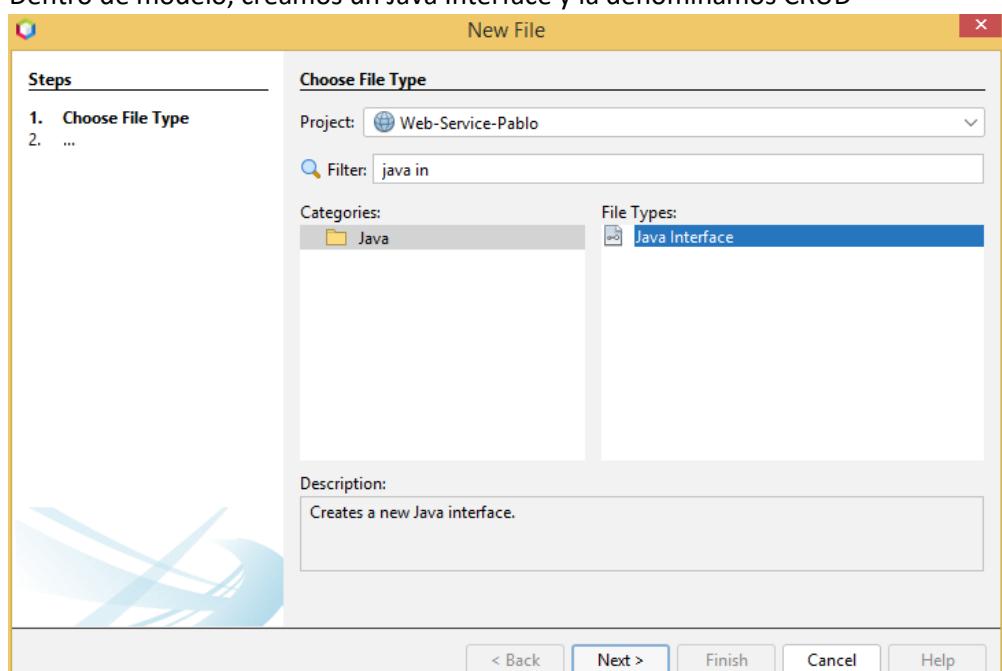


10. Creamos tres Java Class

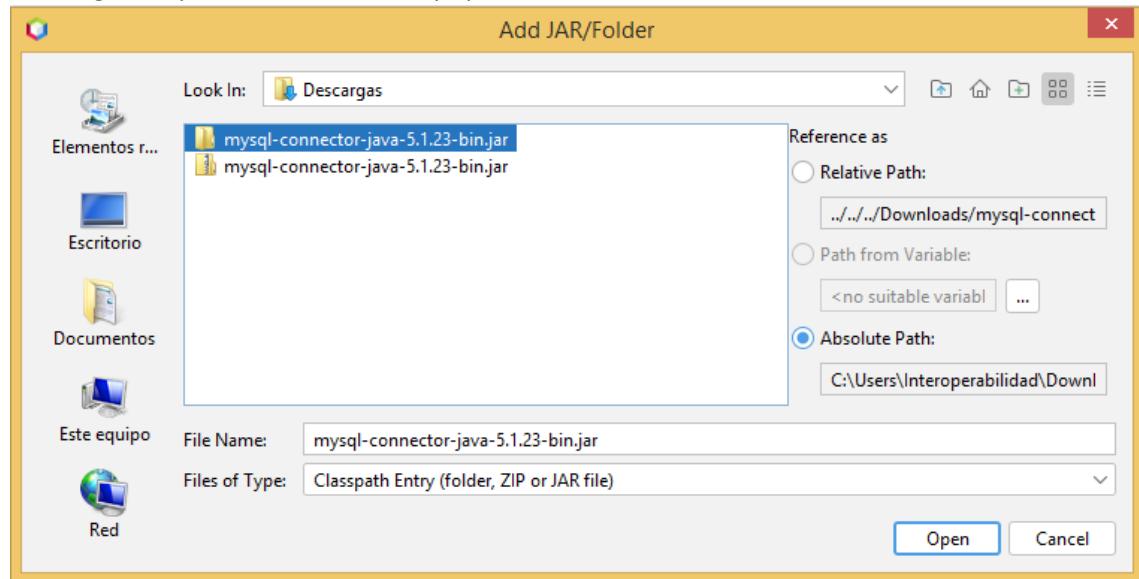




11. Dentro de modelo, creamos un Java Interface y la denominamos CRUD



12. Descargamos y leemos la librería mysql connector



13. Creamos una base de datos “Cliente” en mysql con los siguientes campos dentro de la tabla “clientes”.

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
1	id	int(11)			No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar Más
2	nombre	varchar(45)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
3	direccion	varchar(45)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar Más
4	estado	int(11)			No	Ninguna			Cambiar Eliminar Más

14. D

15. D

16. d

Informe APIREST

1. Tema

Aplicación para consumir un APIREST Comercial

2. Objetivos

Diseñar una interfaz de usuario intuitiva que permita a los usuarios finales realizar consultas a un APIREST comercial.

3. Desarrollo del Informe

1.1. Introducción

En el mundo actual, la integración de sistemas de software juega un papel crucial en mejorar los procesos, la toma de decisiones y la eficiencia operativa. En este escenario, los APIREST (Interfaz de Programación de Aplicaciones Representacional del Estado Transferido) han surgido como una solución poderosa para facilitar la comunicación entre diversos sistemas y plataformas de manera efectiva y adaptable.

Estos APIREST permiten que aplicaciones y servicios se comuniquen a través de Internet utilizando el protocolo HTTP, lo que posibilita un intercambio de datos uniforme y estandarizado. Esta arquitectura basada en estándares abiertos ofrece múltiples ventajas, como la interoperabilidad, la escalabilidad y la modularidad, convirtiéndola en una opción atractiva para empresas de todos los sectores y tamaños.

En este contexto, el proyecto actual se centra en desarrollar e implementar una aplicación diseñada para utilizar un APIREST comercial. El propósito principal de esta aplicación es brindar a los usuarios una herramienta intuitiva y eficaz para acceder a datos y funciones proporcionados por un proveedor externo a través de su APIREST.

En resumen, este proyecto busca no solo desarrollar una aplicación funcional y segura para consumir un APIREST comercial, sino también contribuir al avance de la integración de sistemas en el ámbito empresarial, ofreciendo una solución que permita a las organizaciones aprovechar al máximo las oportunidades que ofrece la tecnología APIREST en la actualidad.

1.2. Marco teórico

Los APIREST (Interfaz de Programación de Aplicaciones Representacional del Estado Transferido) han surgido como un enfoque arquitectónico popular para el diseño de servicios web que se basa en los principios del estilo arquitectónico REST (Representational State Transfer) propuesto por Fielding en 2000 [1]. Este estilo arquitectónico se centra en la identificación de recursos como entidades clave, que se representan mediante URIs (Uniform Resource Identifiers) y se manipulan a través de métodos HTTP estándar, como GET, POST, PUT y DELETE. REST promueve la comunicación sin estado entre el cliente y el servidor, lo que aumenta la escalabilidad y la flexibilidad de los sistemas distribuidos [1].

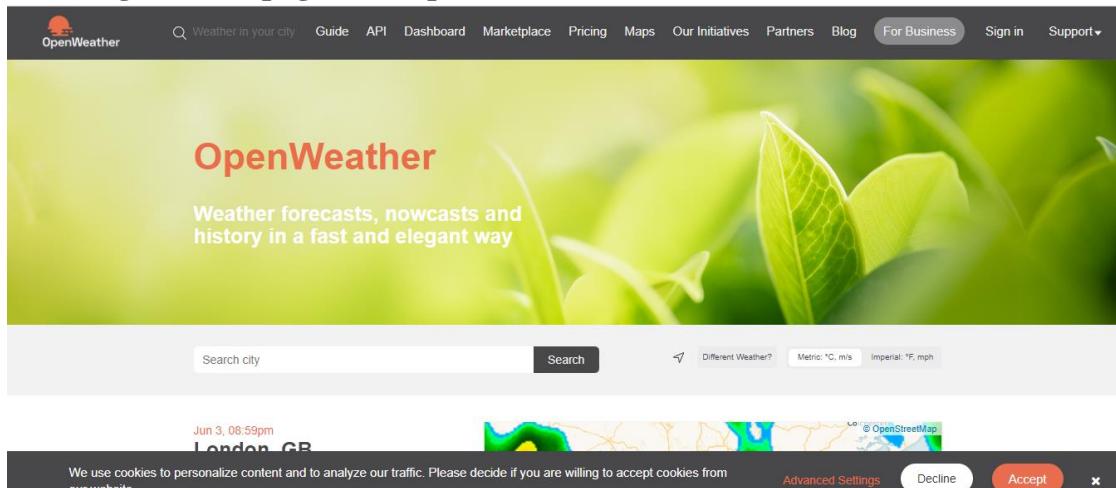
Autenticación con Tokens JWT

La autenticación basada en tokens JSON Web (JWT) es una técnica ampliamente utilizada en el desarrollo de aplicaciones web modernas. Un token JWT es una cadena de caracteres codificada que contiene información de usuario y está firmada digitalmente para garantizar su integridad. Los tokens JWT se utilizan para autenticar solicitudes HTTP y pueden contener datos como el ID de usuario, roles y otros atributos de autorización [2]. Esta forma de autenticación permite la implementación de sistemas sin estado (stateless), donde el servidor no necesita mantener información de sesión, lo que mejora la escalabilidad y la seguridad del sistema.

3.3. Descripción de las acciones realizadas

API de Clima

1. Nos dirigimos a la página de Open Weather



2. Nos registramos

Create New Account

We will use information you provided for management and administration purposes, and for keeping you informed by mail, telephone, email and SMS of other products and services from us and our partners. You can proactively manage your preferences or opt-out of communications with us at any time using Privacy Centre. You have the right to access your data held by us or to request your data to be deleted. For full details please see the OpenWeather [Privacy Policy](#).

I am 16 years old and over

I agree with [Privacy Policy](#), [Terms and conditions of](#)

3. Nos dirigimos al perfil y entramos a las llaves

The screenshot shows the OpenWeatherMap account interface. At the top, there's a navigation bar with links like 'Our Initiatives', 'Partners', 'Blog', 'For Business', 'Pablo' (with a dropdown menu), and 'Support'. Below the navigation is a green header bar with the text 'ail.'. Underneath is a main content area with three buttons: 'My orders', 'My profile', and 'Ask a question'. A dropdown menu from the 'Pablo' button lists options: 'My services', 'My API keys' (which is highlighted), 'My payments', 'My profile', and 'Logout'. Below this is a table titled 'API keys' with columns 'Key', 'Name', 'Status', and 'Actions'. One row is shown: 'dce0cbec4b05b90a79c11e8808eed3a2' (Default, Active, with edit and delete icons). To the right is a 'Create key' section with a 'Create key' button and a 'Generate' button. There's also a text input field for 'API key name'.

4. Generamos una llave y esperamos a que se active, para eso tendremos que revisar la bandeja de entrada del correo con el que nos registramos

Dear Customer!

Thank you for subscribing to Free [OpenWeatherMap](#)!

API key:

- Within the next couple of hours, your API key **dce0cbec4b05b90a79c11e8808eed3a2** will be activated and ready to use
- You can later create more API keys on your [account page](#)
- Please, always use your API key in each API call

Endpoint:

- Please, use the endpoint api.openweathermap.org for your API calls
- Example of API call:
api.openweathermap.org/data/2.5/weather?

5. Copiamos la llave y la reemplazamos en el siguiente link

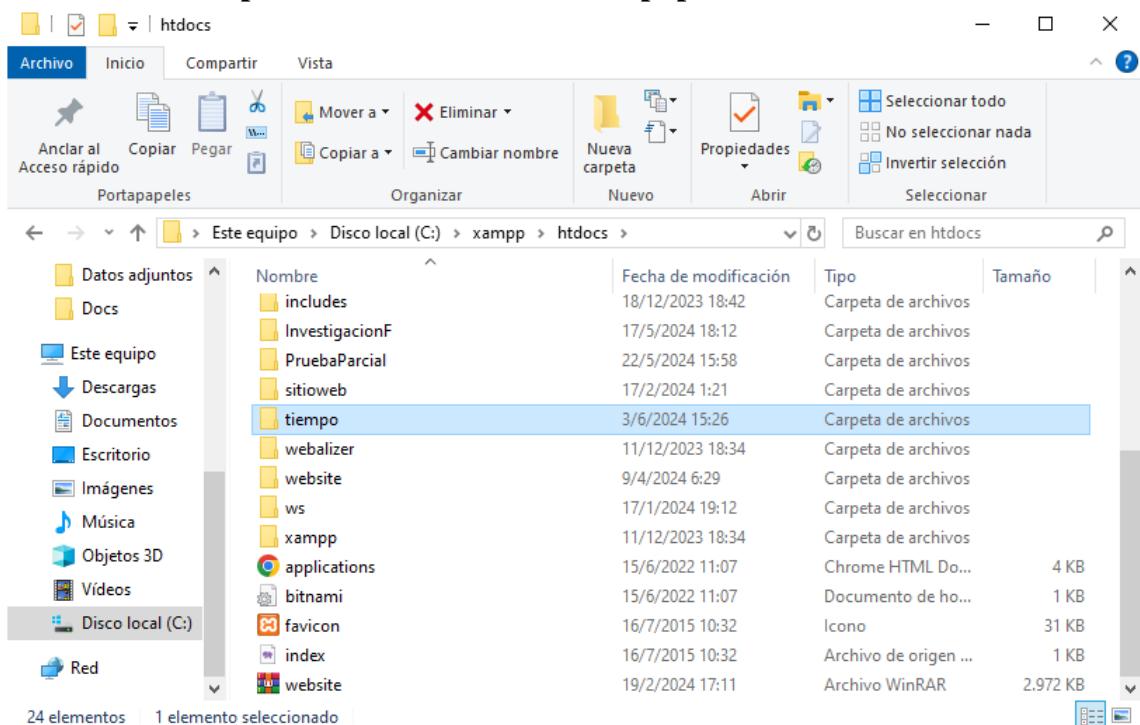
api.openweathermap.org/data/2.5/weather?q={city name}&appid={API key}

The screenshot shows a browser window with the address bar containing the URL: 'api.openweathermap.org/data/2.5/weather?q=Quito&appid=dce0cbec4b05b90a79c11e8808eed3a2'. Below the address bar are various browser tabs and icons. The main content area displays a JSON response from the API, including coordinates, weather conditions, and various atmospheric parameters for Quito, Ecuador.

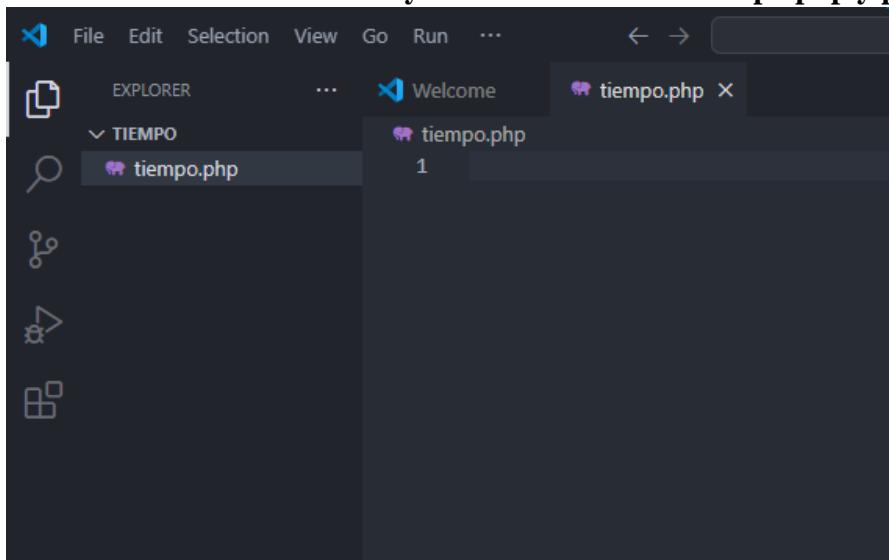
Se generará el link

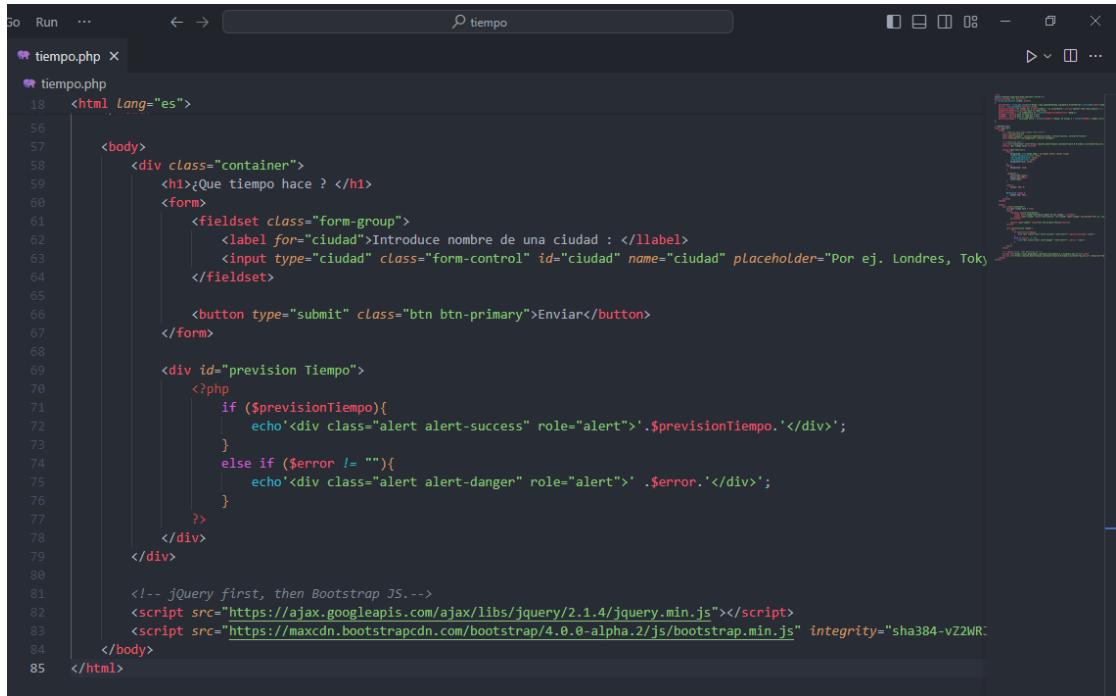
<https://api.openweathermap.org/data/2.5/weather?q=Quito&appid=dce0cbec4b05b90a79c11e8808eed3a2>

6. Creamos una carpeta en el servidor virtual de php



7. Abrimos Visual Studio Code y creamos el archivo tiempo.php y programamos

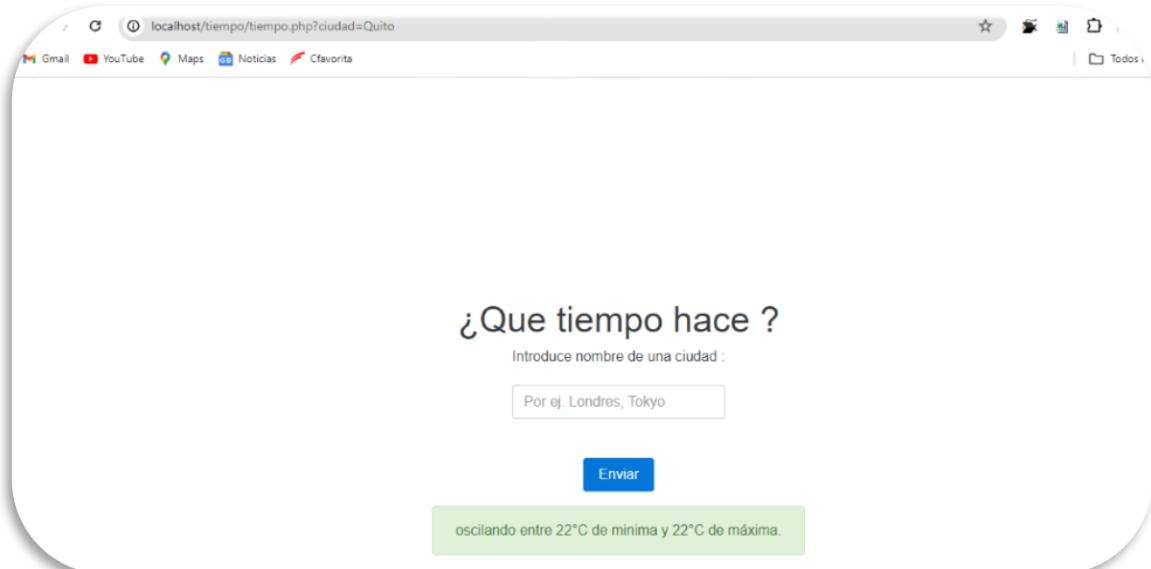




A screenshot of a code editor showing a file named 'tiempo.php'. The code is a simple web form for querying weather information. It includes an HTML header, a container div, a form with a fieldset for entering a city name, and a button to submit the form. Below the form, there's a PHP block that checks if a variable '\$previsionTiempo' is set and echoes an alert message. It also handles an error case where '\$error' is not empty. The code ends with a comment about loading jQuery and Bootstrap JS.

```
<?php
if ($previsionTiempo){
    echo'<div class="alert alert-success" role="alert">' . $previsionTiempo.'</div>';
}
else if ($error != ""){
    echo'<div class="alert alert-danger" role="alert">' . $error.'</div>';
}
?>
</div>
</div>
```

8. Abrimos Xampp y ejecutamos el localhost de tiempo



API de Geoloacalización

The screenshot shows a Google search results page for "Google Cloud Console". The top result is the official Google Cloud Platform website. To the right, there is a summary box for "Google Cloud" with details like "Aplicación", "Lenguajes de programación: Python, Go, Java, C++, Ruby", and "Lanzamiento: 7 de abril de 2008". Below the search results, there's a section titled "También se buscó" with links to Azure, AWS, Microsoft Azure, Amazon Web Services, BigQuery, and Gmail.

The screenshot shows the Google Cloud Console interface for a project named "My Project 6065". A modal dialog box is open, prompting the user to "Selecciona un proyecto". It lists one recent project: "My Project 6065" (ID: my-project-6065-1715963100043). There are tabs for "RECENTES", "DESTACADOS", and "TODOS". The "PROYECTO NUEVO" button is visible at the top right of the dialog. The background shows various Google Cloud services like Cloud Storage, BigQuery, and Kubernetes Engine.

Google Cloud My Project 6065 Buscar () recursos, documentos, productos y más Buscar

APIs y servicios Credenciales + CREAR CREDENCIALES BORRAR RESTABLECER CREDENCIALES BORRADAS

APIs y servicios habilitados Biblioteca Credenciales Pantalla de consentimiento ... Acuerdos de uso de páginas

Crea credenciales para acceder a tus API habilitadas. [Más información](#)

Recuerda configurar la pantalla de consentimiento de OAuth con información sobre tu app.

CONFIGURAR PANTALLA DE CONSENTIMIENTO

Claves de API

Nombre	Fecha de creación	Restricciones	Acciones
Clave de API 1	9 jun 2024	Ninguno	MOSTRAR CLAVE

ID de clientes OAuth 2.0

Nombre	Fecha de creación	Tipo	ID de cliente	Acciones
No hay clientes de OAuth para mostrar				

Cuentas de servicio

Correo electrónico	Nombre	Acciones
No hay cuentas de servicio para mostrar		

Google Cloud My Project 6065 Buscar () recursos, documentos, productos y más Buscar

APIs y servicios Credenciales + CREAR CREDENCIALES BORRAR RESTABLECER CREDENCIALES BORRADAS

APIs y servicios habilitados Biblioteca Credenciales Pantalla de consentimiento ... Acuerdos de uso de páginas

Clave de API

Crea credenciales para acceder a tus API habilitadas. Identifica tu proyecto con una clave de API simple para verificar la cuota y el acceso

Recuerda configurar la pantalla de consentimiento de OAuth para que tu app pueda acceder a sus datos

ID de cliente de OAuth

Solicita el consentimiento del usuario para que tu app pueda acceder a sus datos

Cuenta de servicio

Habilita la autenticación de servidor a servidor en el nivel de la app mediante cuentas robot

Claves de API

Nombre	Restricciones	Acciones
Ayúdame a elegir	Ninguno	MOSTRAR CLAVE

ID de clientes OAuth 2.0

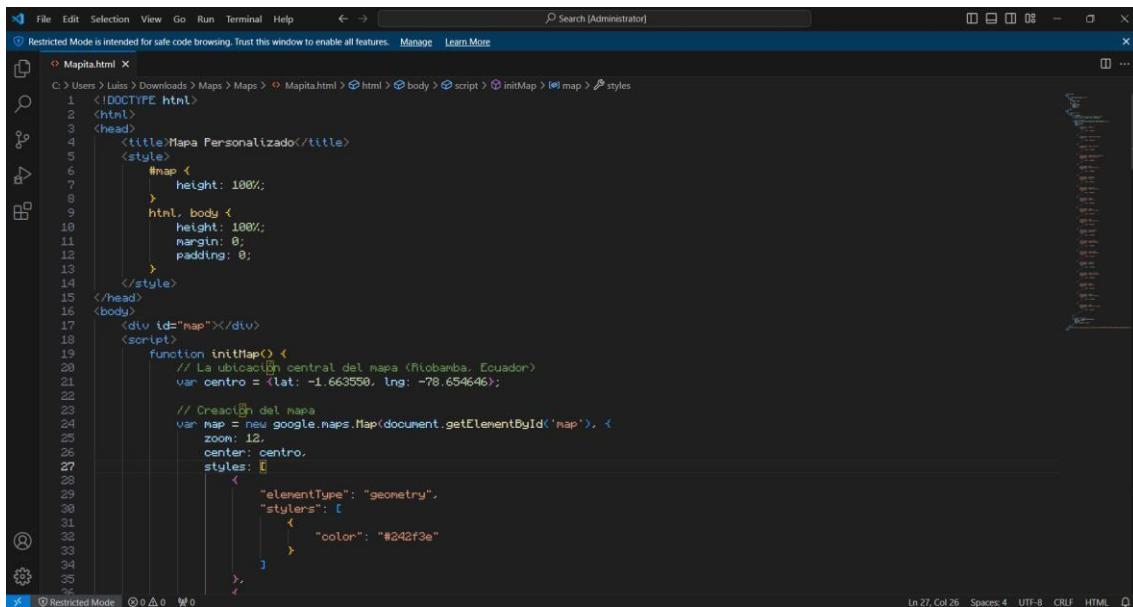
Nombre	Fecha de creación	Tipo	ID de cliente	Acciones
No hay clientes de OAuth para mostrar				

Cuentas de servicio

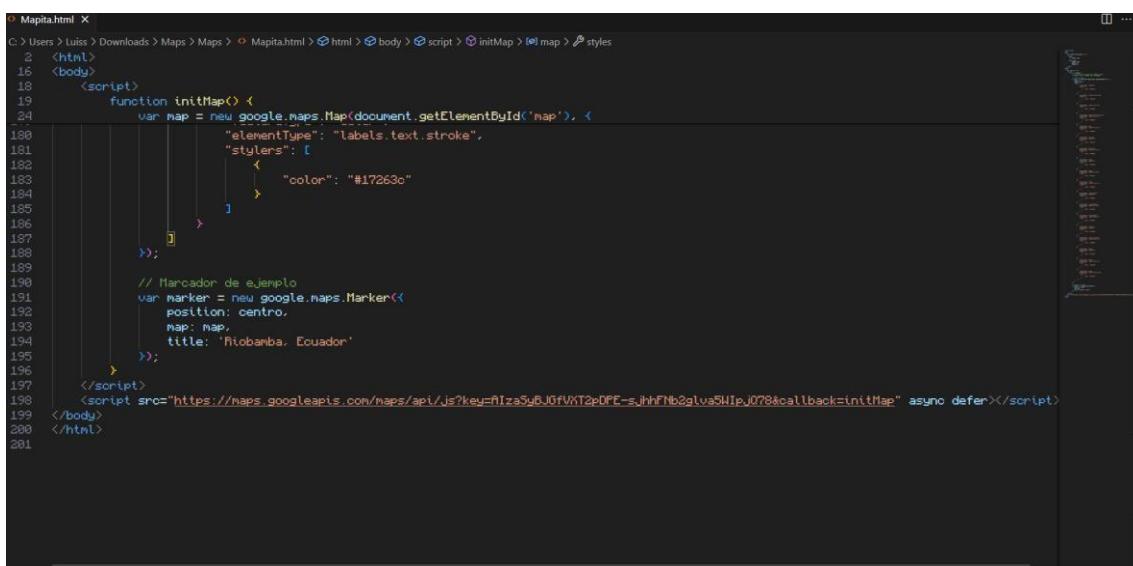
Correo electrónico	Nombre	Acciones
No hay cuentas de servicio para mostrar		

The screenshot shows the Google Cloud Platform API & Services Credentials page. A modal window titled "Se creó la clave de API" (The API key was created) is displayed, containing the key value "A1zaSybq_02qBPMcdfoIzm152nXQhg8UJWf1Ys". Below the modal, there is a note: "Para usar esta clave en tu aplicación, transfírela con el parámetro key=API_KEY." (To use this key in your application, transfer it with the key=API_KEY parameter.) A "Copiar en el portapapeles" (Copy to clipboard) button is visible next to the key value. The background shows sections for "Claves de API", "ID de clientes OAuth 2.0", and "Cuentas de servicio".

The screenshot shows a Windows File Explorer window titled "GEOLOCALIZACIÓN". The left sidebar shows standard locations like "Este equipo", "Descargas", "Documentos", etc. The main pane displays a folder structure under "Disco local (C:)". A file named "mapa" is selected, showing its details: "Nombre: mapa", "Fecha de modificación: 10/6/2024 14:22", "Tipo: Archivo de origen ...", and "Tamaño: 3 KB". The status bar at the bottom indicates "1 elemento" and "1 elemento seleccionado 2,62 KB".



```
<!DOCTYPE html>
<html>
<head>
    <title>Mapa Personalizado</title>
    <style>
        #map {
            height: 100%;
        }
        html, body {
            height: 100%;
            margin: 0;
            padding: 0;
        }
    </style>
</head>
<body>
    <div id="map"></div>
    <script>
        function initMap() {
            // La ubicación central del mapa (Quito, Ecuador)
            var centro = {lat: -1.663550, lng: -78.54646};
            // Creación del mapa
            var map = new google.maps.Map(document.getElementById('map'), {
                zoom: 12,
                center: centro,
                styles: [
                    {
                        "elementType": "geometry",
                        "stylers": [
                            {
                                "color": "#2f3e4e"
                            }
                        ]
                    }
                ]
            });
        }
    </script>
</body>
</html>
```



```
<!DOCTYPE html>
<html>
<head>
    <script>
        function initMap() {
            var map = new google.maps.Map(document.getElementById('map'), {
                styles: [
                    {
                        "elementType": "labels.text.stroke",
                        "stylers": [
                            {
                                "color": "#17263c"
                            }
                        ]
                    }
                ]
            });
            // Marcador de ejemplo
            var marker = new google.maps.Marker({
                position: centro,
                map: map,
                title: 'Quito, Ecuador'
            });
        }
    </script>
    <script src="https://maps.googleapis.com/maps/api/js?key=AIzaSyBJGfVXT2pDPE-sJhhFnB2alva5WIpj07&callback=initMap" async defer></script>
</head>
<body>
    <div id="map"></div>
</body>
</html>
```

localhost/GEOLOCALIZACIÓN/mapa.php

API DE GEOLOCALIZACIÓN

Tu posición actual es: [-1.6520607, -78.6429949](#)

[Mostrar tu posición en un mapa](#)

google.com/maps?q=-1.6520607,-78.6429949

3.4. Conclusiones

Durante el desarrollo del proyecto de la aplicación para consumir APIs de temperatura y geolocalización, se lograron cumplir con los objetivos establecidos.

Se logró una integración exitosa de las APIs de temperatura y geolocalización en la aplicación. Se implementaron funciones y métodos para realizar consultas a estas APIs, obtener datos relevantes y presentarlos de manera clara y comprensible para el usuario.

Se diseñó e implementó una interfaz de usuario intuitiva y fácil de usar que permite a los usuarios interactuar con la aplicación y acceder a los datos proporcionados por las APIs de temperatura y geolocalización. La interfaz proporciona opciones claras para realizar consultas específicas, así como una retroalimentación visual sobre los datos obtenidos.

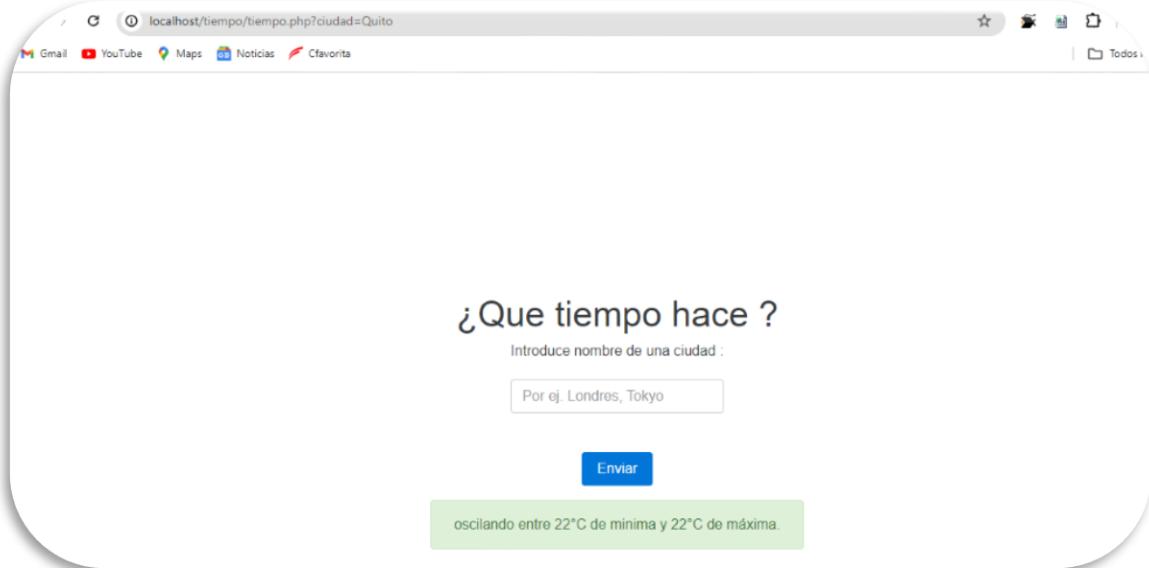
Se integró un sistema de autenticación basado en tokens JWT en la aplicación para garantizar la seguridad y la integridad de la autenticación del usuario. Los tokens JWT se utilizan para autenticar las solicitudes de los usuarios y acceder a las funcionalidades ofrecidas por las APIs de temperatura y geolocalización de manera segura.

3.5. Bibliografía

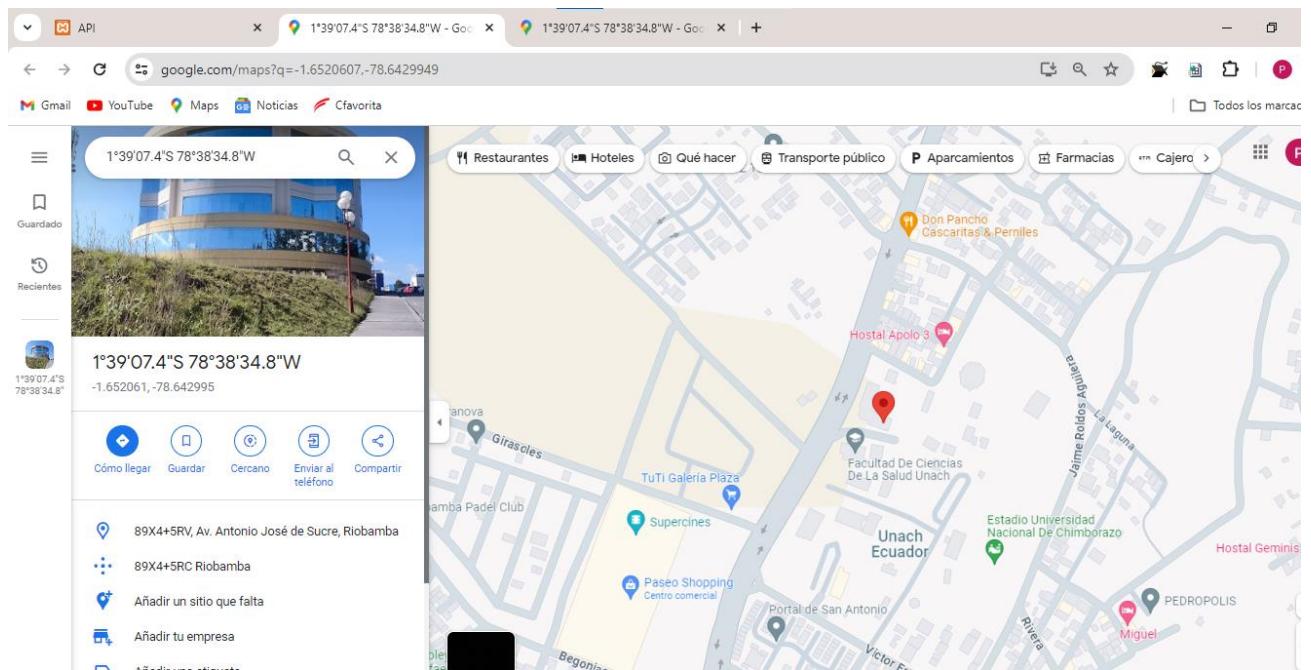
- [1] R. T. Fielding, "Architectural Styles and the Design of Network-based Software Architectures," Doctoral dissertation, University of California, Irvine, 2000.
- [2] Open Web Application Security Project (OWASP), "JSON Web Token (JWT) Cheat Sheet," Recuperado de https://cheatsheetseries.org/chheetsheets/JSON_Web_Token_Cheat_Sheet.html.

4. Anexos

API de clima



API de Geolocalización



Informe Backend y Frontend

1. Tema

Creación de un Frontend y Backend para consumo local.

2. Objetivo

Objetivo General:

Desarrollar un sistema de gestión de contenido web que permita a los usuarios crear, leer, actualizar y eliminar información de manera intuitiva y eficiente, utilizando un frontend atractivo y fácil de usar, junto con un backend robusto y escalable.

Objetivos Específicos:

1. Diseñar e implementar un frontend interactivo y receptivo que proporcione una experiencia de usuario fluida y atractiva, permitiendo a los usuarios acceder y manipular la información de manera intuitiva a través de una interfaz amigable.
2. Desarrollar un backend seguro y eficiente que maneje de manera confiable las solicitudes de los usuarios, garantizando la integridad y la confidencialidad de los datos, así como la escalabilidad del sistema para soportar un crecimiento futuro en el volumen de usuarios y datos.

3. Desarrollo

3.1. Introducción

La programación web es una disciplina que ha revolucionado la forma en que interactuamos con la información y los servicios en línea. Es un vasto campo que abarca diversas áreas, entre las cuales se destacan el frontend y el backend. Estos dos términos, a menudo mencionados en el contexto del desarrollo web, son fundamentales para comprender cómo funciona la web moderna y cómo se construyen las aplicaciones que utilizamos a diario.

El frontend y el backend son dos partes esenciales de cualquier aplicación web. Juntos, forman la columna vertebral de la experiencia del usuario y permiten que las

aplicaciones funcionen de manera eficiente y sin problemas. Para entender mejor estos conceptos, es útil desglosarlos y examinarlos individualmente.

Comencemos con el frontend. El frontend, también conocido como el lado del cliente, se refiere a todo lo que un usuario ve y con lo que interactúa en una aplicación web. Esto incluye la interfaz de usuario, el diseño, la disposición de los elementos visuales y cualquier elemento con el que un usuario pueda interactuar directamente. En términos técnicos, el frontend está compuesto principalmente por lenguajes de marcado como HTML para la estructura, CSS para el diseño y la presentación, y JavaScript para la interactividad y la lógica del lado del cliente.

HTML, o HyperText Markup Language, proporciona la estructura básica de una página web, utilizando elementos como encabezados, párrafos, listas y más para organizar el contenido. CSS, o Cascading Style Sheets, se utiliza para dar estilo a estos elementos, controlando aspectos como el color, la tipografía, el tamaño y la disposición. Por último, JavaScript es un lenguaje de programación que permite agregar interactividad a una página web, como formularios dinámicos, animaciones, y la manipulación del contenido de la página en tiempo real.

Por otro lado, el backend, también conocido como el lado del servidor, es la parte de la aplicación web que los usuarios no ven directamente, pero que es fundamental para su funcionamiento. El backend se encarga de procesar las solicitudes del cliente, realizar operaciones en la base de datos, y devolver los resultados al cliente. En términos generales, el backend está compuesto por un servidor, una aplicación y una base de datos.

El servidor es el software que se ejecuta en un ordenador remoto y que escucha las solicitudes de los clientes, procesa esas solicitudes y devuelve las respuestas apropiadas. La aplicación es el código que se ejecuta en el servidor y que se encarga de manejar la lógica de negocio de la aplicación, como la autenticación de usuarios, el procesamiento de pagos y cualquier otra operación que requiera acceso a datos o recursos del servidor. La base de datos es un sistema de almacenamiento que permite a la aplicación almacenar y recuperar datos de manera eficiente.

Para que una aplicación web funcione correctamente, el frontend y el backend deben trabajar juntos de manera armoniosa. El frontend envía solicitudes al backend, el backend procesa esas solicitudes y devuelve los resultados al frontend, que los

muestra al usuario final. Esta comunicación continua entre el frontend y el backend es lo que permite que las aplicaciones web sean dinámicas, interactivas y capaces de realizar tareas complejas.

En resumen, el frontend y el backend son dos componentes fundamentales de cualquier aplicación web. Mientras que el frontend se encarga de la interfaz de usuario y la interactividad del lado del cliente, el backend maneja la lógica de negocio y el almacenamiento de datos del lado del servidor. Juntos, forman un sistema completo que permite crear aplicaciones web modernas y eficientes que mejoran la experiencia del usuario y satisfacen sus necesidades en línea.

3.2. Marco Teórico

Visual Studio Code (VS Code)

Visual Studio Code es un entorno de desarrollo integrado (IDE) de código abierto desarrollado por Microsoft. Es ampliamente utilizado por desarrolladores de software debido a su interfaz de usuario intuitiva, su velocidad y su amplia gama de extensiones que permiten personalizar y ampliar sus funcionalidades.

Características clave de Visual Studio Code:

- 1. Edición de código inteligente:** VS Code ofrece resaltado de sintaxis, completado automático, sugerencias inteligentes y refactorización de código para mejorar la productividad del desarrollador.
- 2. Depuración integrada:** Permite depurar aplicaciones directamente desde el editor, con puntos de interrupción, seguimiento de variables y consola integrada.
- 3. Control de versiones integrado:** Integra perfectamente con sistemas de control de versiones como Git, lo que facilita el seguimiento de cambios en el código y la colaboración en equipos.
- 4. Extensiones:** Ofrece una amplia variedad de extensiones que permiten a los desarrolladores personalizar su entorno de desarrollo y agregar soporte para diferentes lenguajes de programación, herramientas y tecnologías.

Backend

El backend, también conocido como lado del servidor, es la parte de una aplicación web que se ejecuta en el servidor y es responsable de procesar las solicitudes del cliente, interactuar con la base de datos y realizar cualquier procesamiento necesario para generar una respuesta que se envía de vuelta al cliente.

Tecnologías comunes en el desarrollo de backend:

- 1. Lenguajes de programación:** Tales como Python (con frameworks como Django y Flask), JavaScript (con Node.js), Java (con Spring Framework) y Ruby (con Ruby on Rails).
- 2. Bases de datos:** Como MySQL, PostgreSQL, MongoDB y SQLite.
- 3. Servidores web:** Como Apache, Nginx y Microsoft IIS.
- 4. Frameworks y bibliotecas:** Que proporcionan funcionalidades predefinidas y facilitan el desarrollo de aplicaciones web, como Express.js para Node.js, Django para Python y Spring Boot para Java.

Frontend

El frontend, también conocido como lado del cliente, es la parte de una aplicación web con la que interactúan los usuarios directamente. Se compone principalmente de HTML, CSS y JavaScript, que se utilizan para estructurar el contenido, aplicar estilos y añadir interactividad a la aplicación.

Componentes clave del frontend:

- 1. HTML (HyperText Markup Language):** Define la estructura y el contenido de una página web mediante etiquetas y elementos como encabezados, párrafos, imágenes y enlaces.
- 2. CSS (Cascading Style Sheets):** Controla la presentación y el diseño de una página web, permitiendo establecer estilos como colores, fuentes, márgenes y alineaciones.
- 3. JavaScript:** Es un lenguaje de programación que permite agregar interactividad y dinamismo a una página web, mediante la manipulación del DOM (Document

Object Model), la gestión de eventos del usuario y la comunicación con el backend.

Postman

Postman es una plataforma de colaboración para el desarrollo de APIs que permite a los desarrolladores crear, probar y depurar APIs de manera eficiente. Ofrece una interfaz intuitiva que permite enviar solicitudes HTTP, ver respuestas, configurar entornos de prueba y automatizar tareas relacionadas con las API.

Características de Postman:

- 1. Entorno de desarrollo completo:** Proporciona todas las herramientas necesarias para el desarrollo de APIs en un solo lugar, incluyendo la creación de solicitudes, la documentación de APIs y la automatización de pruebas.
- 2. Pruebas automatizadas:** Permite crear y ejecutar pruebas automatizadas para verificar la funcionalidad y el rendimiento de las APIs, lo que facilita la detección de errores y la mejora de la calidad del software.
- 3. Colaboración:** Facilita la colaboración entre equipos de desarrollo al permitir compartir colecciones de API, colaborar en entornos de trabajo compartidos y comentar sobre solicitudes y respuestas.
- 4. Integraciones:** Se integra con otros servicios y herramientas populares, como Git, Jenkins y Slack, para facilitar la integración continua y la comunicación en equipos de desarrollo.

SQL (Structured Query Language)

SQL es un lenguaje de programación utilizado para gestionar y manipular bases de datos relacionales. Permite realizar operaciones como crear, leer, actualizar y eliminar datos en una base de datos, así como realizar consultas para recuperar información específica.

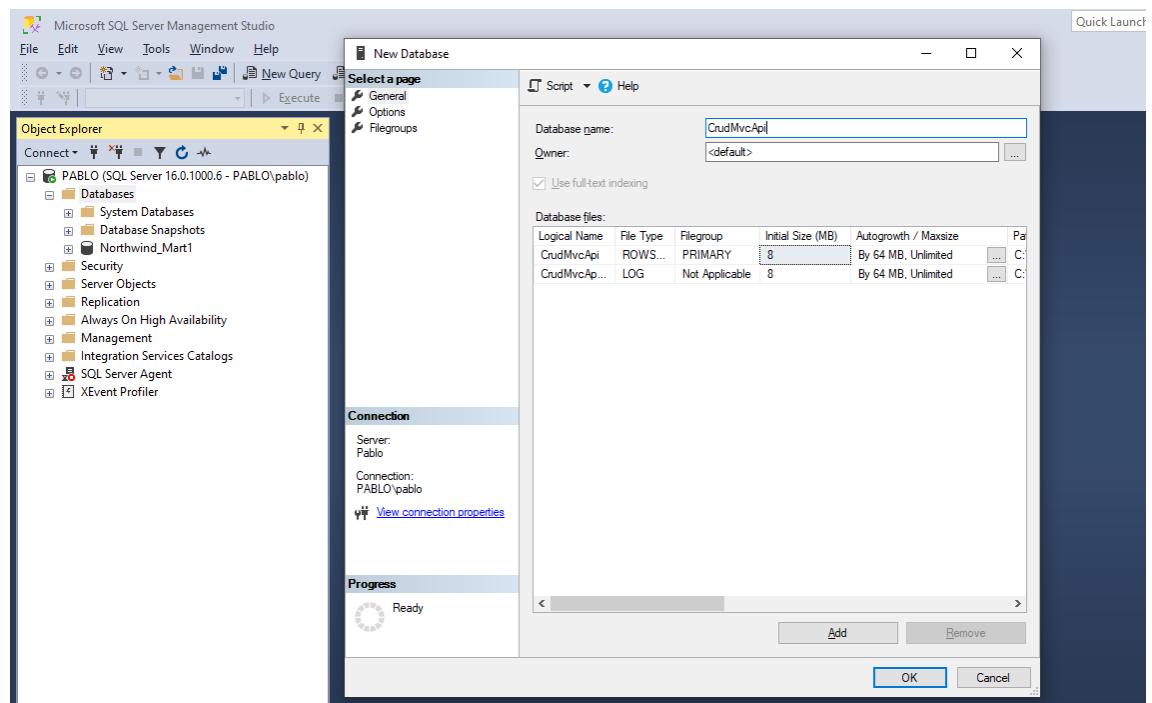
Principales operaciones SQL:

- 1. SELECT:** Se utiliza para recuperar datos de una o varias tablas de la base de datos.

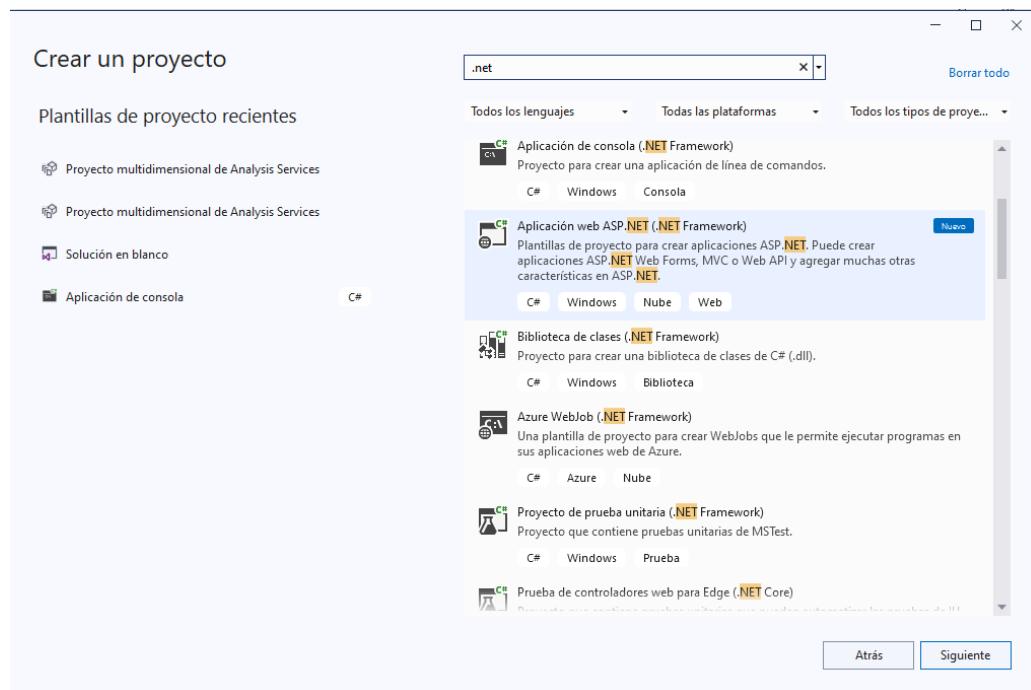
2. **INSERT:** Se utiliza para añadir nuevos registros a una tabla.
3. **UPDATE:** Se utiliza para modificar datos existentes en una tabla.
4. **DELETE:** Se utiliza para eliminar registros de una tabla.
5. **CREATE:** Se utiliza para crear nuevas tablas, índices o vistas en la base de datos.
6. **ALTER:** Se utiliza para modificar la estructura de una tabla existente, como añadir, modificar o eliminar columnas.
7. **DROP:** Se utiliza para eliminar objetos de la base de datos, como tablas, índices o vistas.

3.3.Resultados

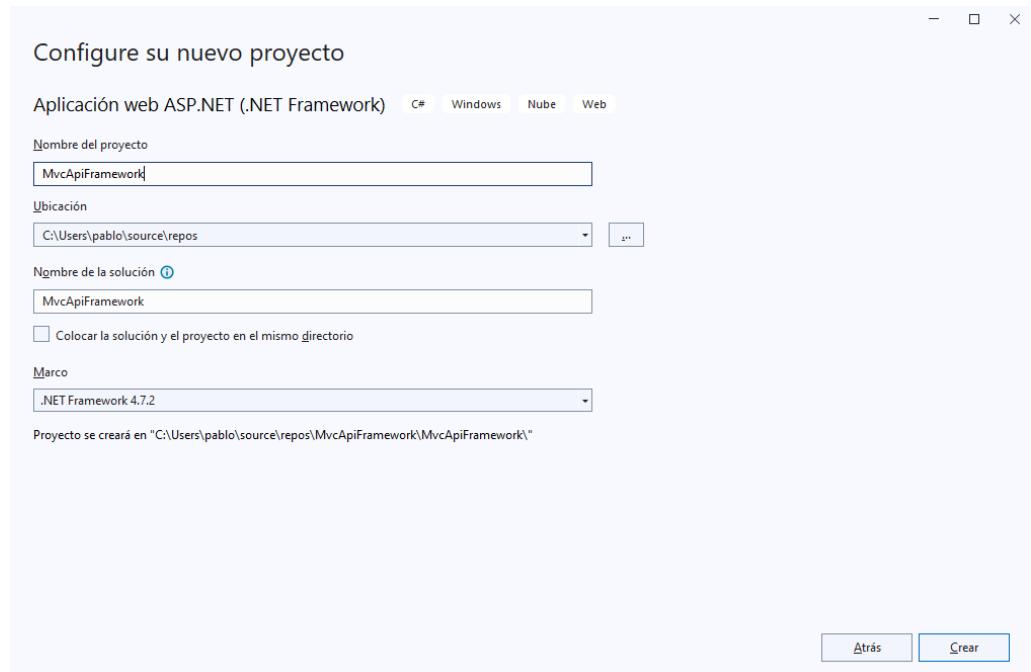
1.



2.



3.



4.

Crear una aplicación web ASP.NET

Vacio
Una plantilla de proyecto vacía para crear aplicaciones ASP.NET. Esta plantilla no tiene contenido.

Web Forms
Una plantilla de proyecto para crear aplicaciones de ASP.NET Web Forms. ASP.NET Web Forms le permite crear sitios web dinámicos con un modelo familiar controlado por eventos para arrastrar y colocar. Una superficie de diseño y cientos de controles y componentes le permiten crear rápidamente sofisticados y eficaces sitios controlados por la interfaz de usuario y con acceso a datos.

MVC
Una plantilla de proyecto para crear aplicaciones ASP.NET MVC. ASP.NET MVC permite compilar aplicaciones mediante la arquitectura de controlador de vista de modelos. ASP.NET MVC incluye muchas características que permiten un desarrollo rápido orientado a pruebas para crear aplicaciones que usan los últimos estándares.

API web
Plantilla de proyecto para crear servicios HTTP REST que pueden llegar a una amplia gama de clientes, como, por ejemplo, exploradores y dispositivos móviles.

Aplicación de página única
Una plantilla de proyectos para crear aplicaciones HTML5 atractivas controladas por JavaScript del lado cliente mediante ASP.NET Web API. Las aplicaciones de una sola página proporcionan una experiencia de usuario atractiva que incluye interacciones del lado cliente mediante HTML5, CSS3 y JavaScript.

Autenticación
Ninguno

Agregar carpetas y referencias principales

Formularios Web Forms
 MVC
 API web

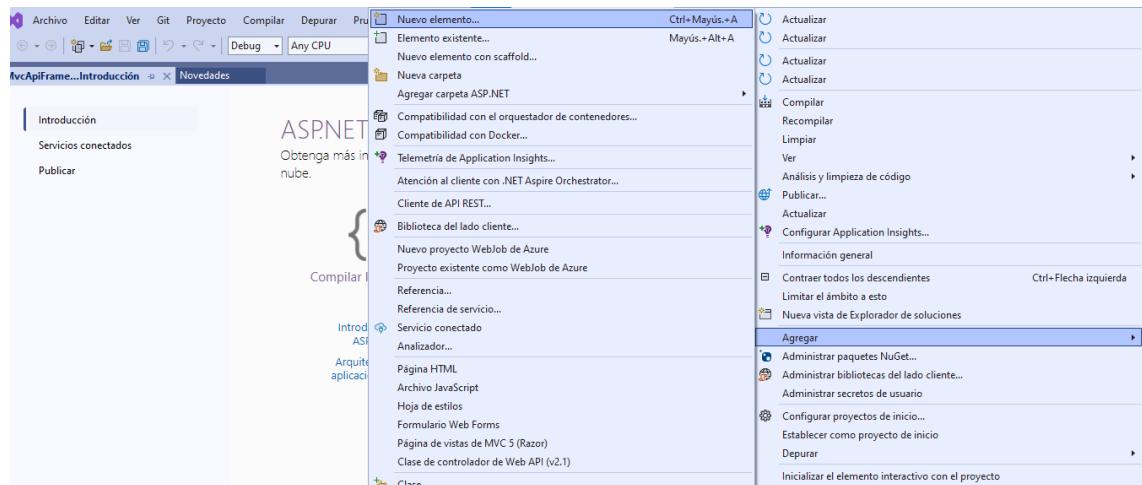
Avanzado

Configurar para HTTPS
 Compatibilidad con Docker
(Requiere Docker Desktop)
 Crear también un proyecto para pruebas unitarias

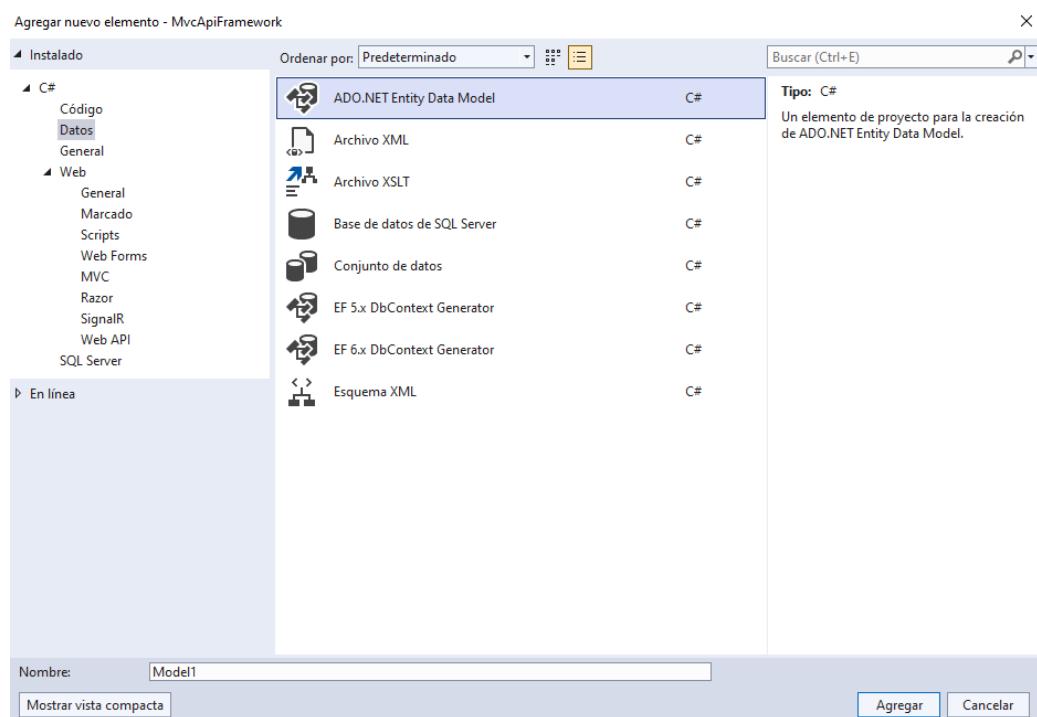
MvcApiFramework.Tests

Atrás Crear

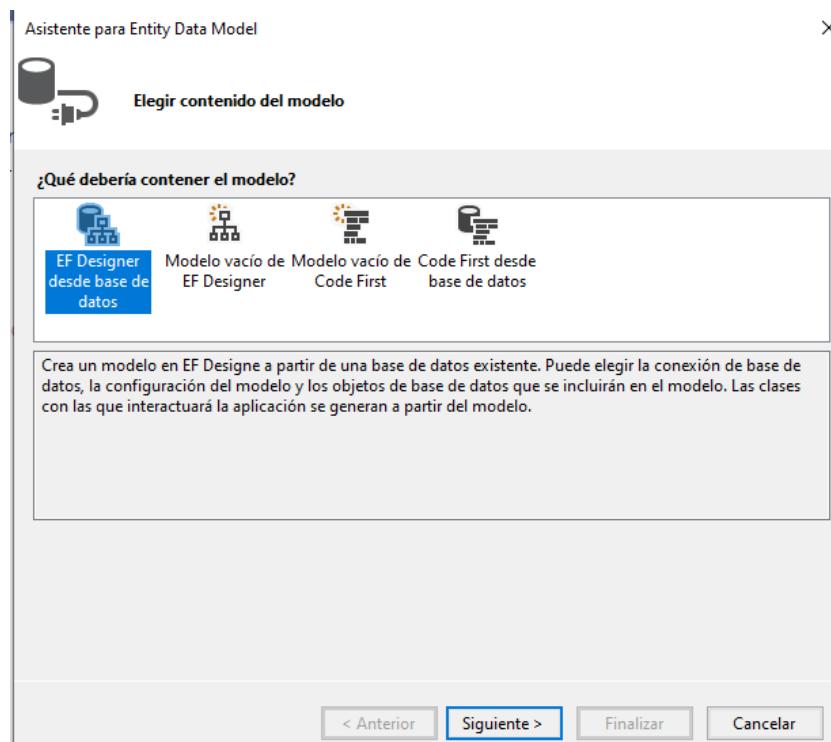
5.



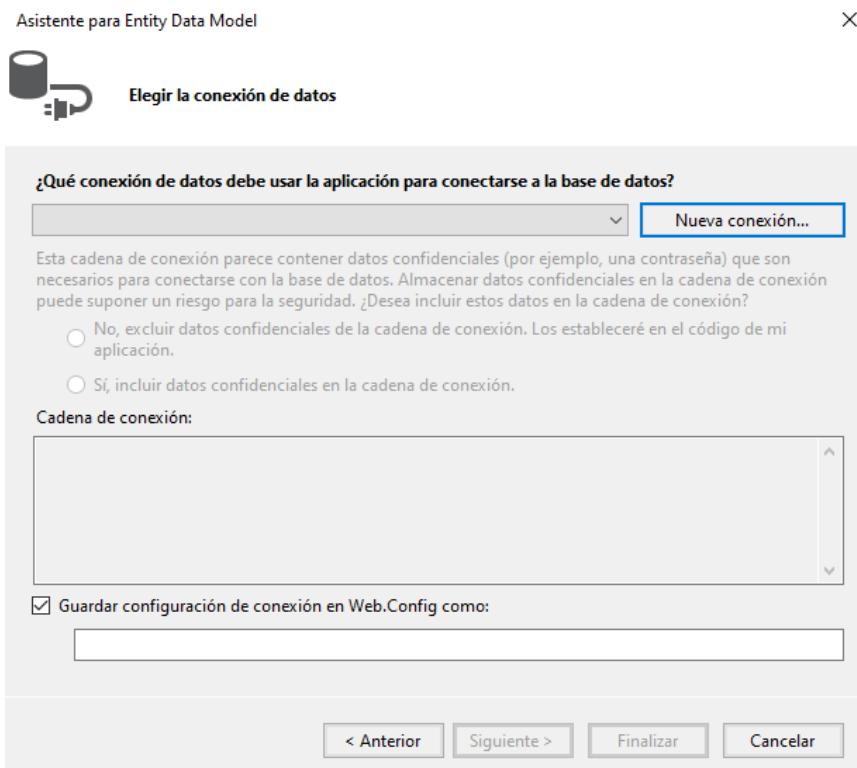
6.



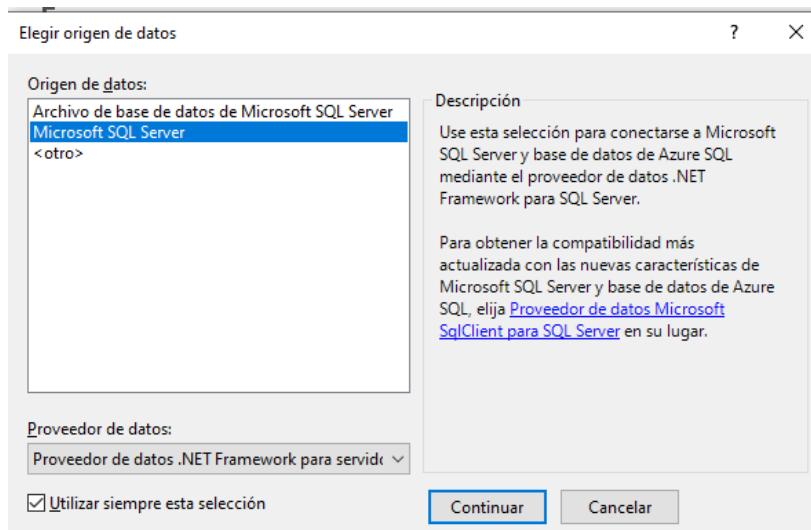
7.



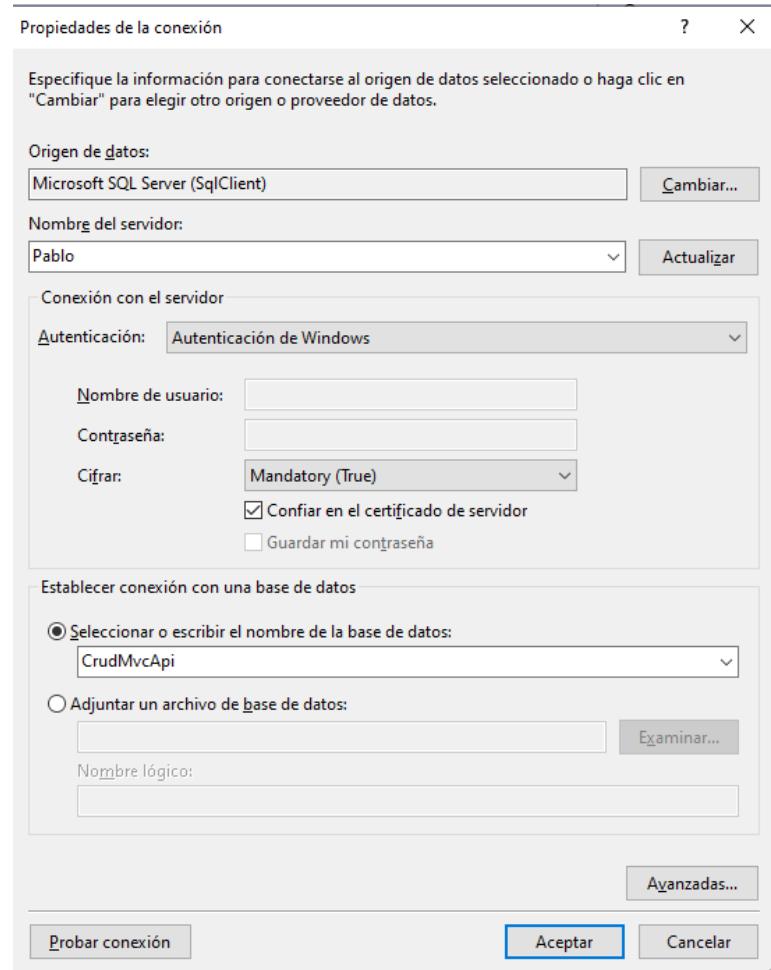
8.



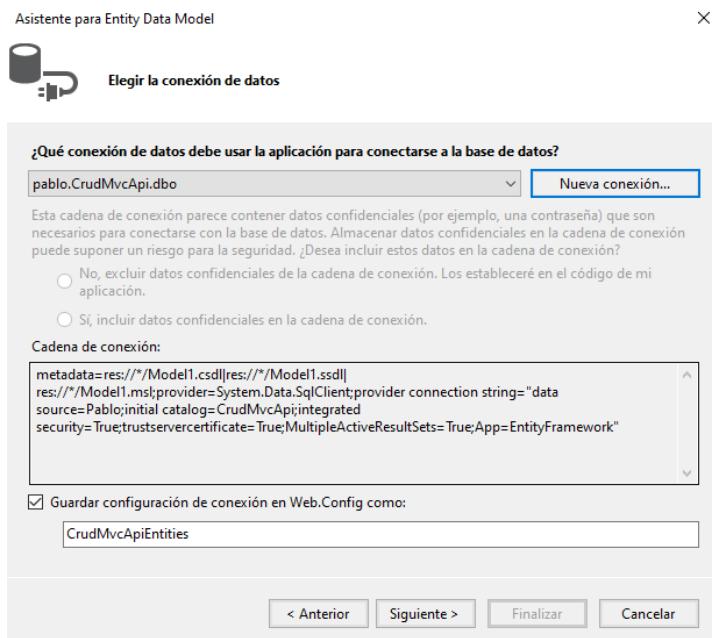
9.



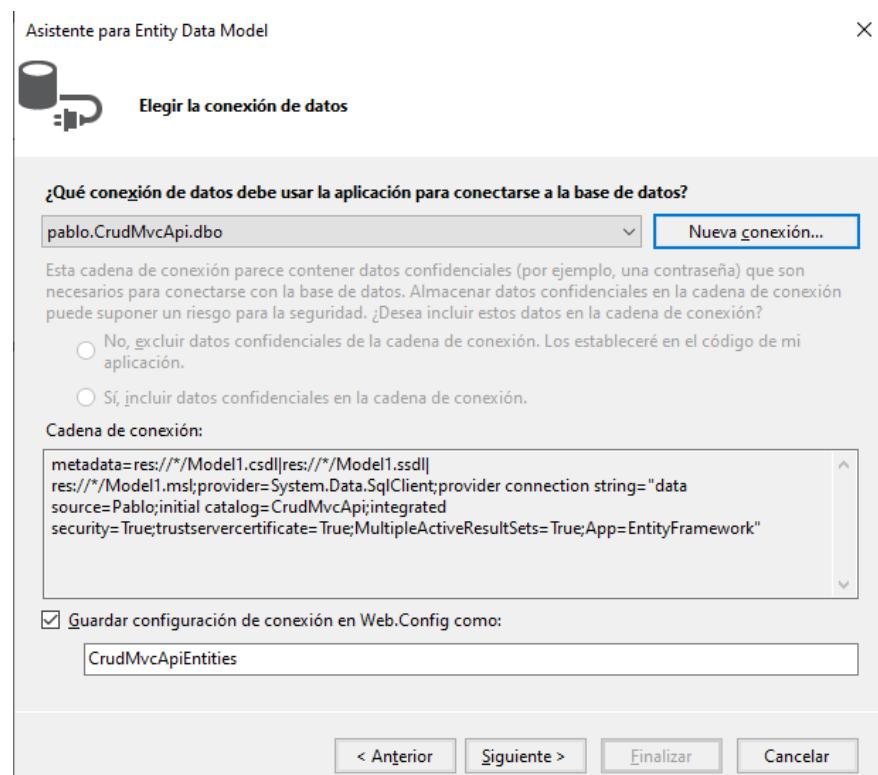
10.



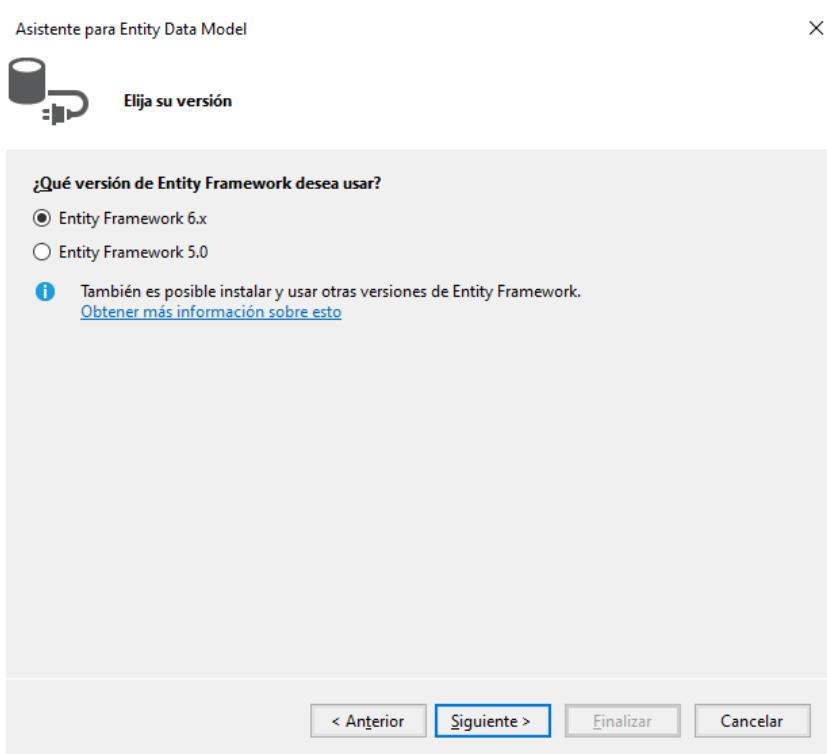
11.



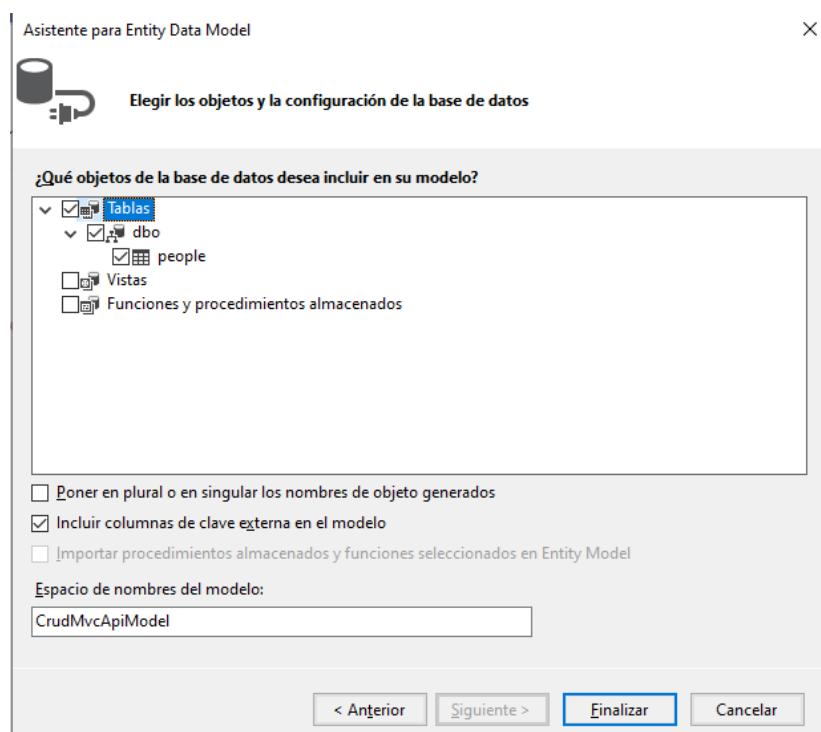
12.



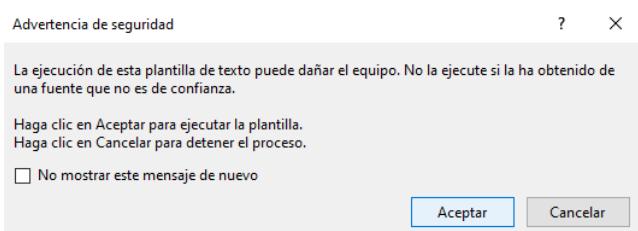
13.



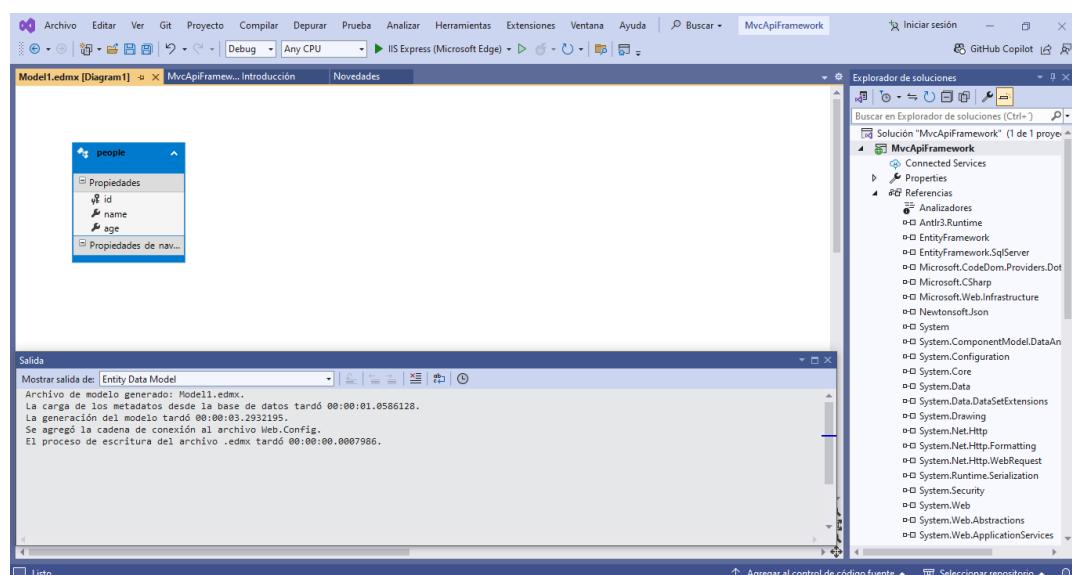
14.



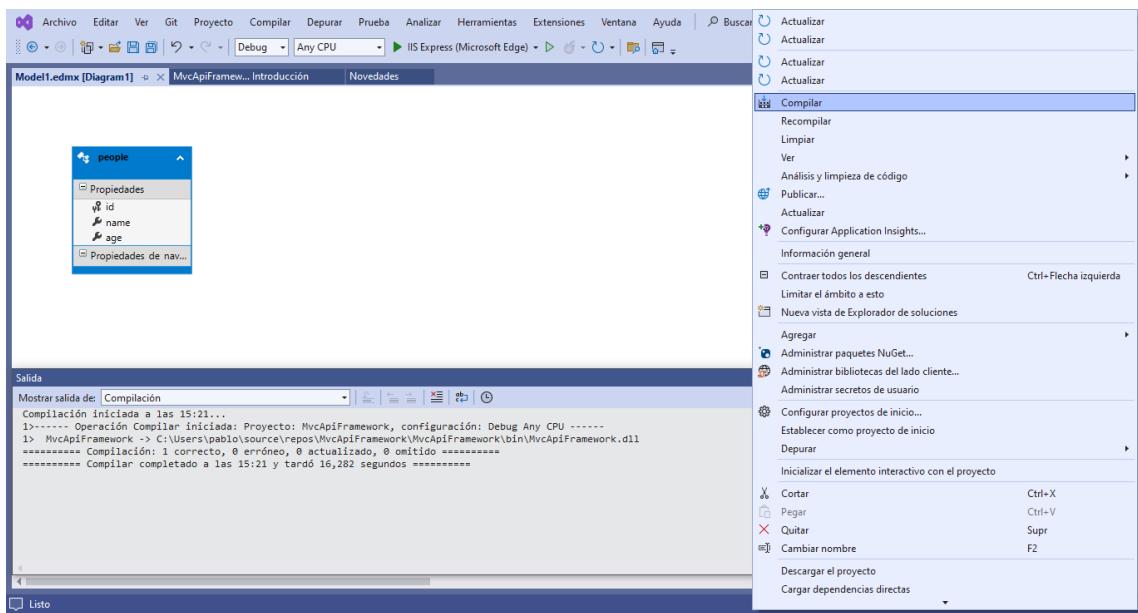
15.



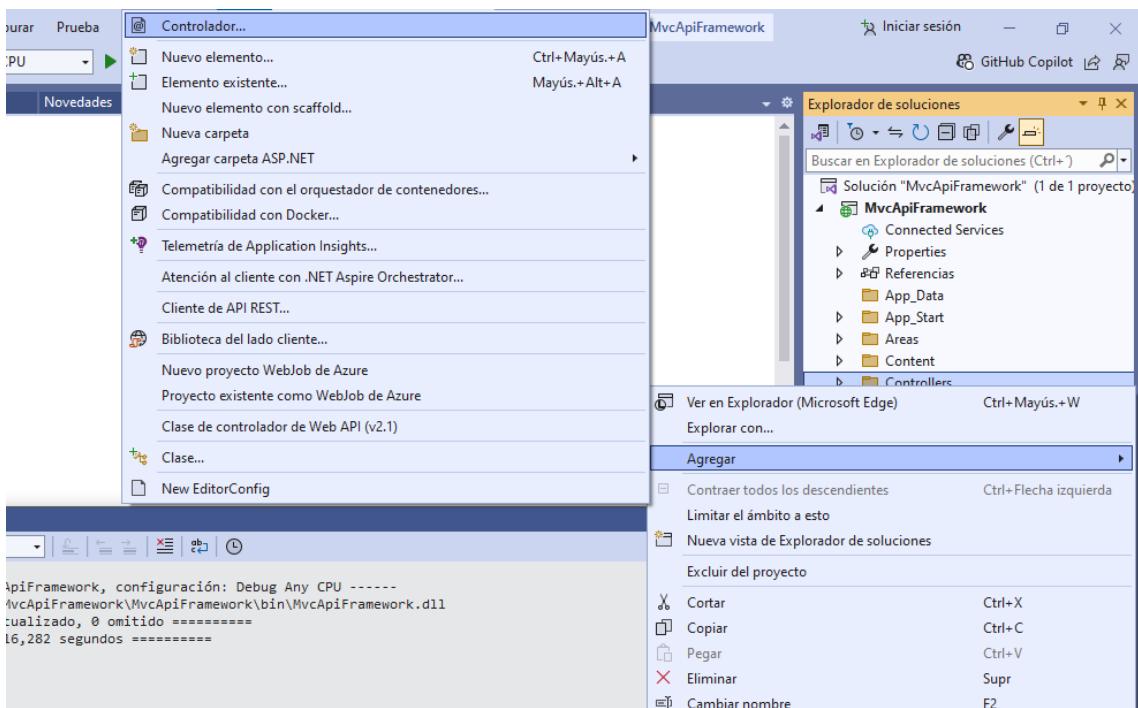
16.



17. Compilar

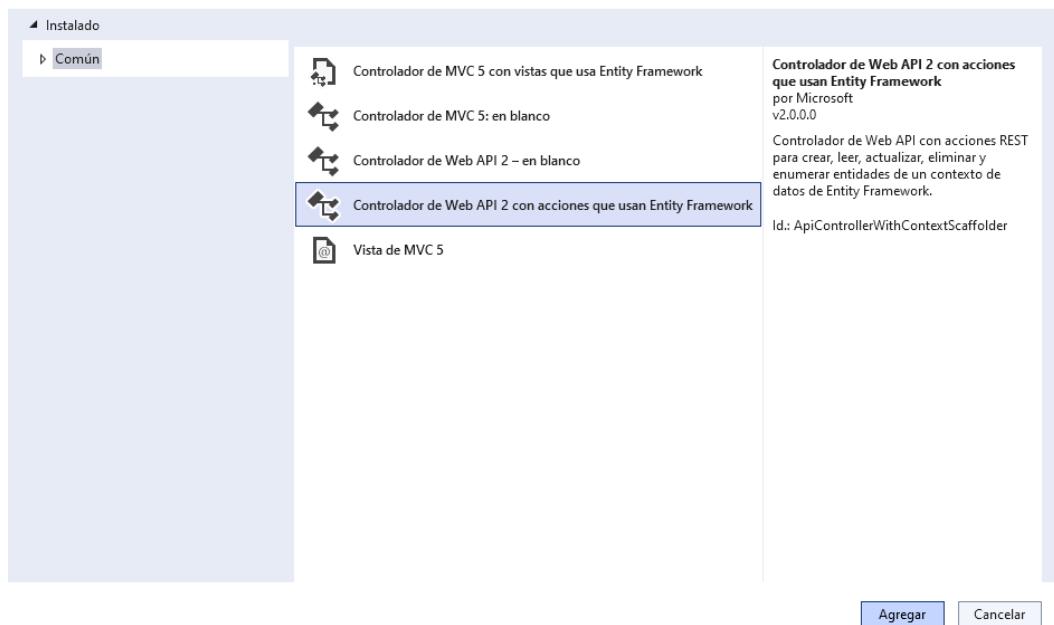


18.

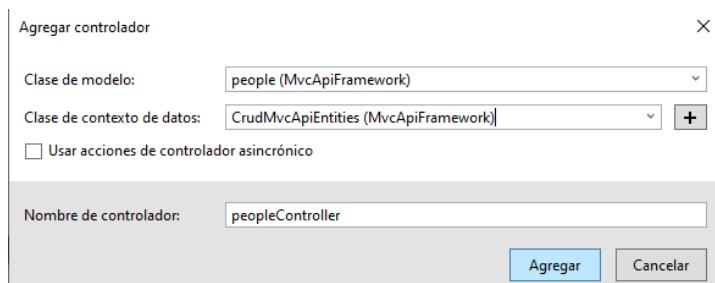


19.

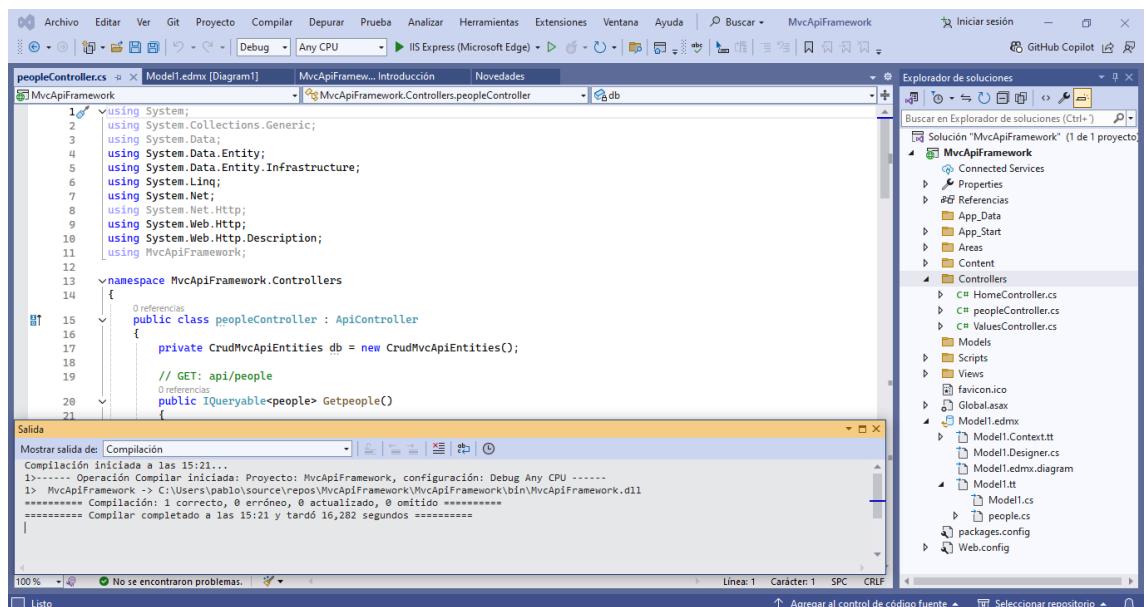
Agregar nuevo elemento con scaffolding



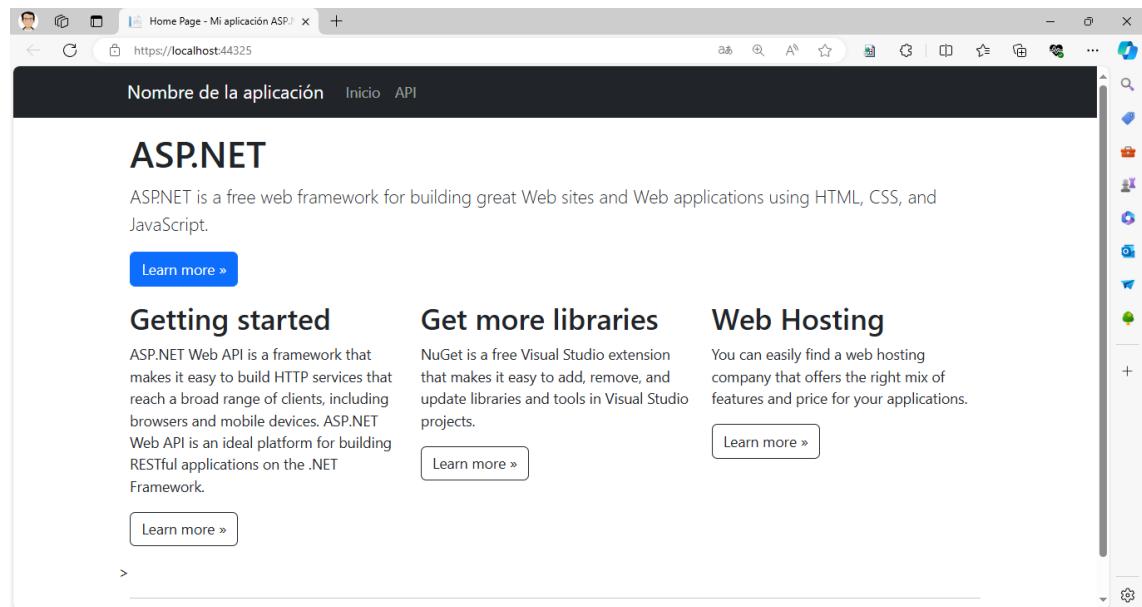
20.



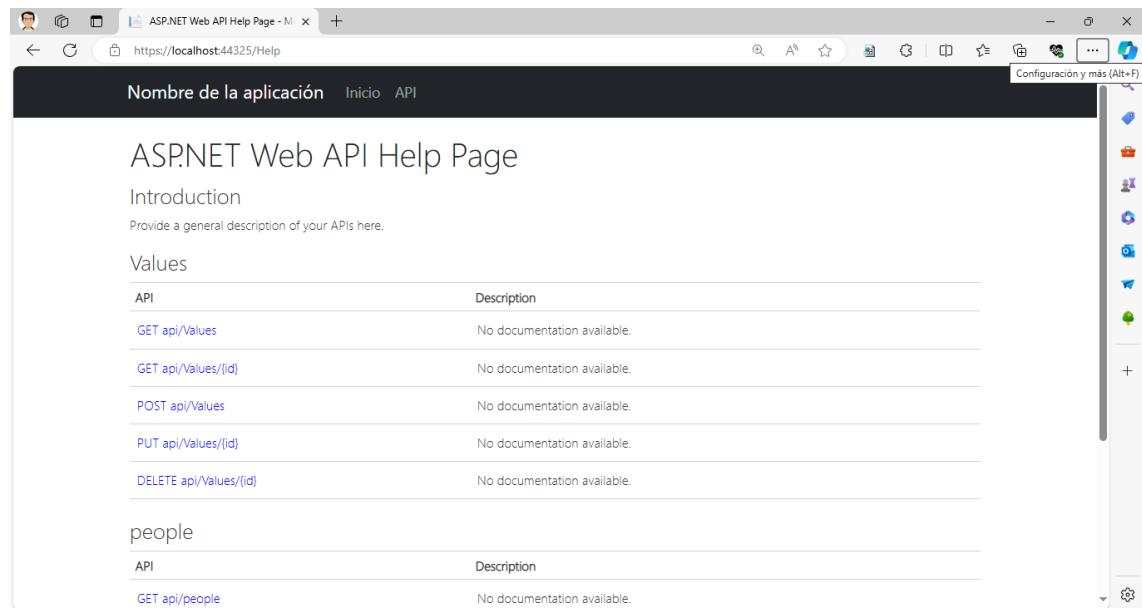
21.



22. Ejcutamos



23. Se observan los métodos



24.

The screenshot shows the API Network interface with a workspace named "Prueba1". A modal dialog is open under the "New" tab, titled "Blank collection". It includes a "Create from template" section with options like "REST API basics", "Integration testing basics", and "API documentation", along with a "View more templates →" link. The background shows the workspace's overview and some API endpoints.

RES



This ten
variable

25. En la URL ingresamos la de la API

The screenshot shows the API Network interface with a workspace named "Prueba1". A modal dialog is open under the "New" tab, titled "HTTP REST API basics: CRUD, test & variable / Post data". It shows a "POST" method selected with the URL "https://localhost:44325/api/people". The "Body" tab is active, showing a table for "Query Params" with two rows: "Key" and "Value".

26.

The screenshot shows a web browser window with the address bar set to "localhost:44325/api/people". The page content displays an XML document:

```
<ArrayOfpeople xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://schemas.datacontract.org/2004/07/MvcApiFramework"/>
```

27.

HTTP REST API basics: CRUD, test & variable / Post data

POST https://localhost:44325/api/people

Params Authorization Headers (8) Body Scripts Settings Cookies

Body (raw JSON)

```
1 {
2   "id": 1,
3   "name": "Juan Sebastian",
4   "age": 22
5 }
```

Body Cookies Headers (11) Test Results (1/1)

Status: 201 Created Time: 16 ms Size: 478 B Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 1,
3   "name": "Juan Sebastian",
4   "age": 22
5 }
```

28.

Pablo.CrudMvcApi - dbo.people

	id	name	age
	0	Add your name...	0
▶	1	Juan Sebastian	22
*	NULL	NULL	NULL

FRONT END

1.

The screenshot shows the Visual Studio Marketplace search results for 'asp.net'. The search bar at the top contains 'asp.net'. Below it, there are three dropdown menus: 'Todos los lenguajes', 'Todas las plataformas', and 'Todos los tipos de proye...'. A search result for 'gRPC Servicio gRPC de ASP.NET Core' is shown, followed by 'Aplicación web ASP.NET (.NET Framework)', which is highlighted with a gray background. This result includes a description: 'Plantillas de proyecto para crear aplicaciones ASP.NET. Puede crear aplicaciones ASP.NET Web Forms, MVC o Web API y agregar muchas otras características en ASP.NET.' Below this are other results: 'Aplicación web ASP.NET (.NET Framework)' and 'ASP.NET Core vacío'.

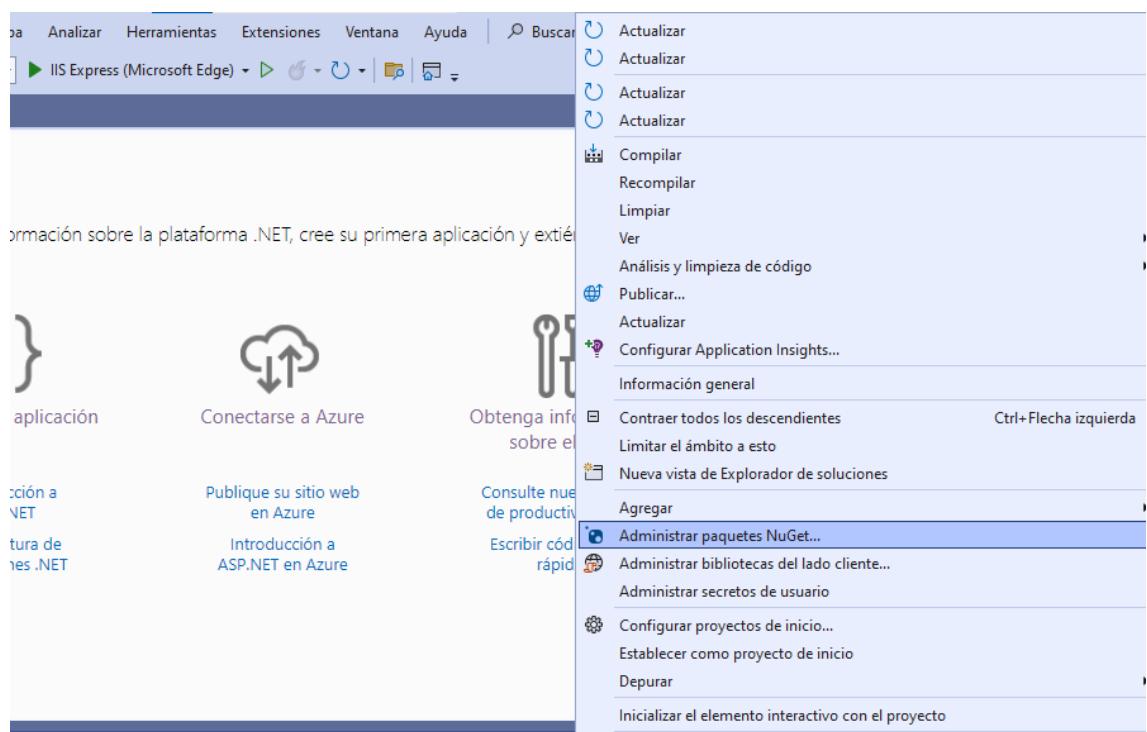
2.

The screenshot shows the 'Configure su nuevo proyecto' (Configure your new project) dialog for an 'Aplicación web ASP.NET (.NET Framework)' project. The title bar says 'Aplicación web ASP.NET (.NET Framework)' with tabs for C#, Windows, Nube, and Web. The 'Nombre del proyecto' field contains 'CleinteWebApi'. The 'Ubicación' field shows the path 'C:\Users\pablo\source\repos'. The 'Nombre de la solución' field also contains 'CleinteWebApi'. There is an unchecked checkbox 'Colocar la solución y el proyecto en el mismo directorio'. The 'Marco' field is set to '.NET Framework 4.7.2'. At the bottom, a note states 'Proyecto se creará en "C:\Users\pablo\source\repos\CleinteWebApi\CleinteWebApi"'.

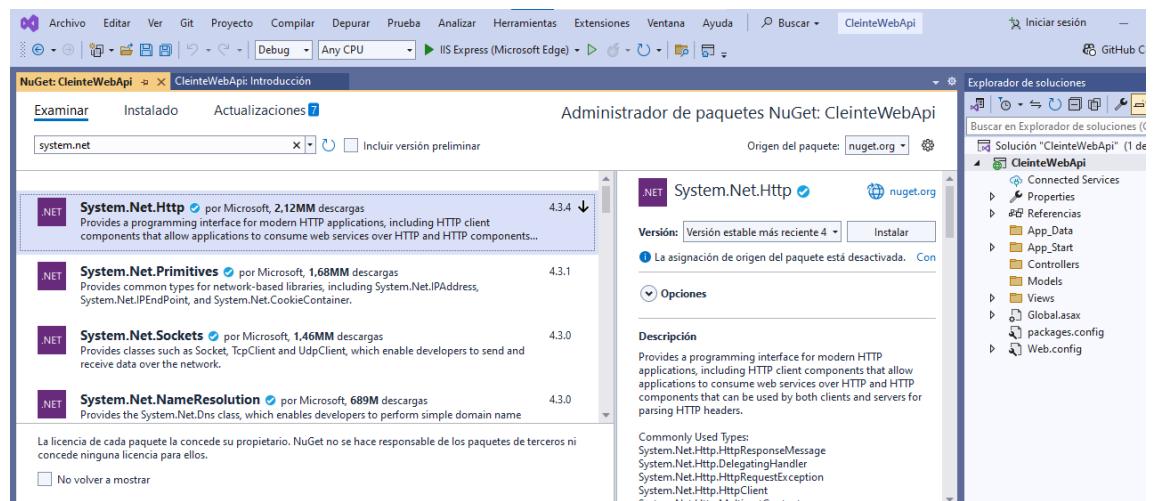
3.



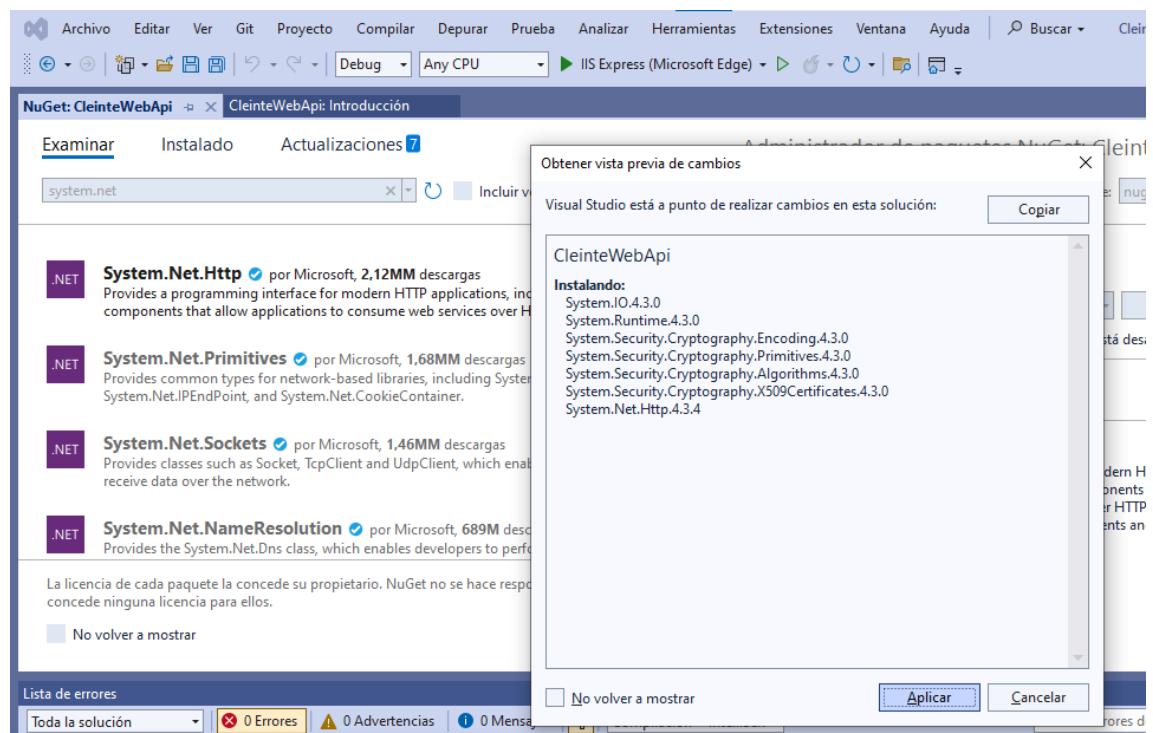
4.

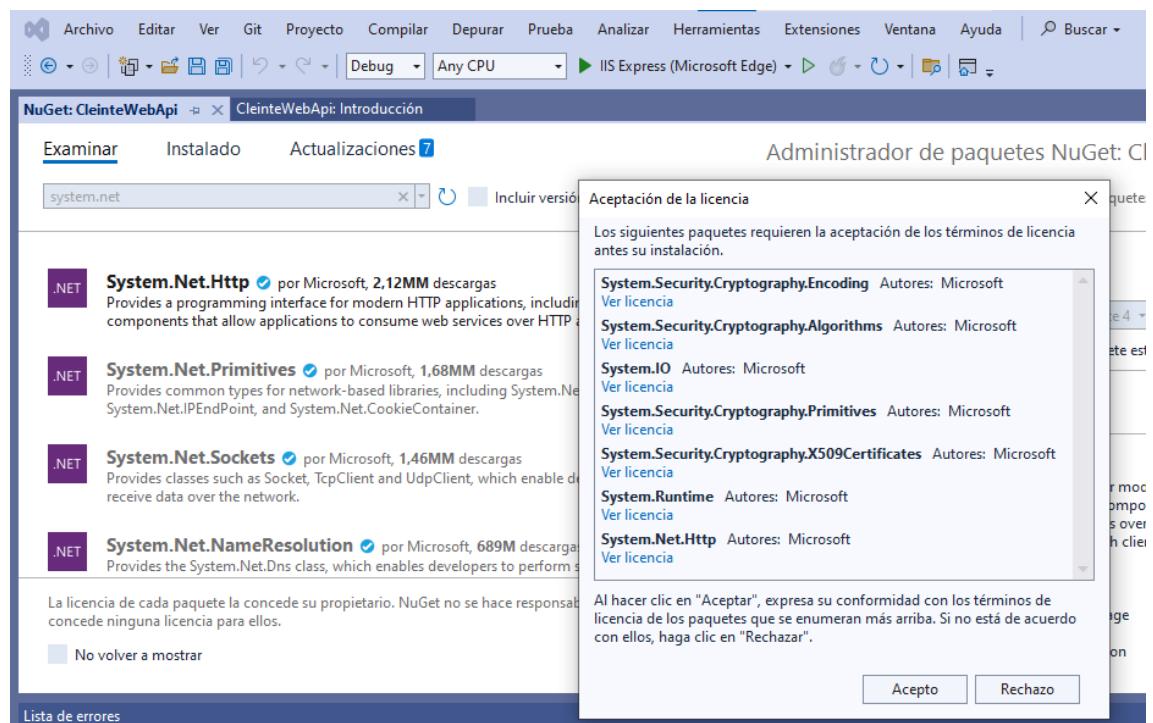


5.

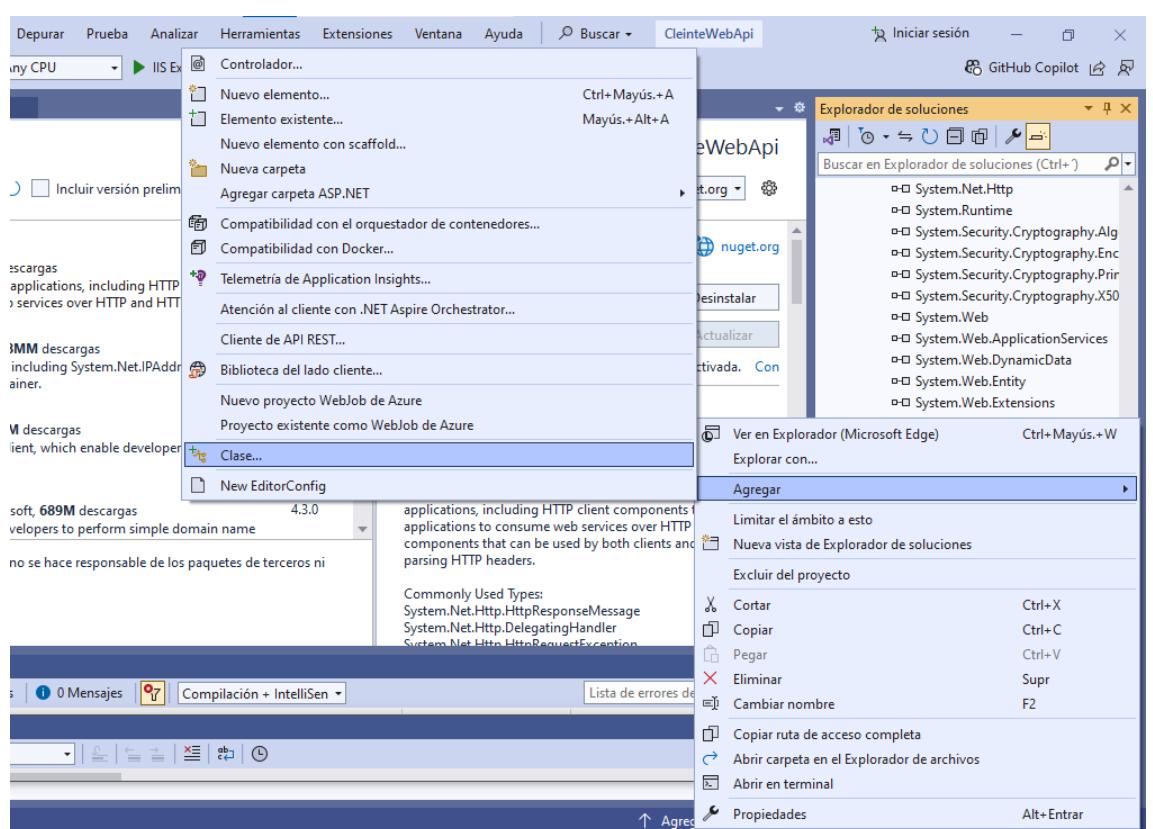


6.

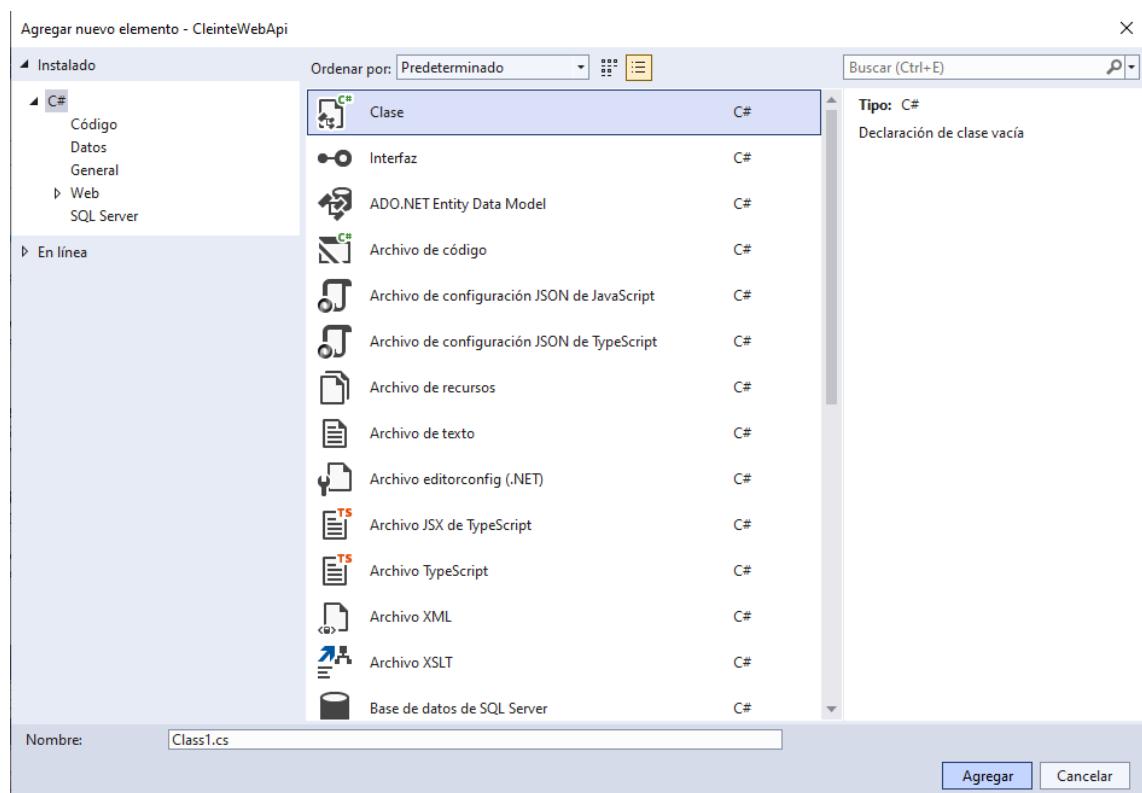




7.



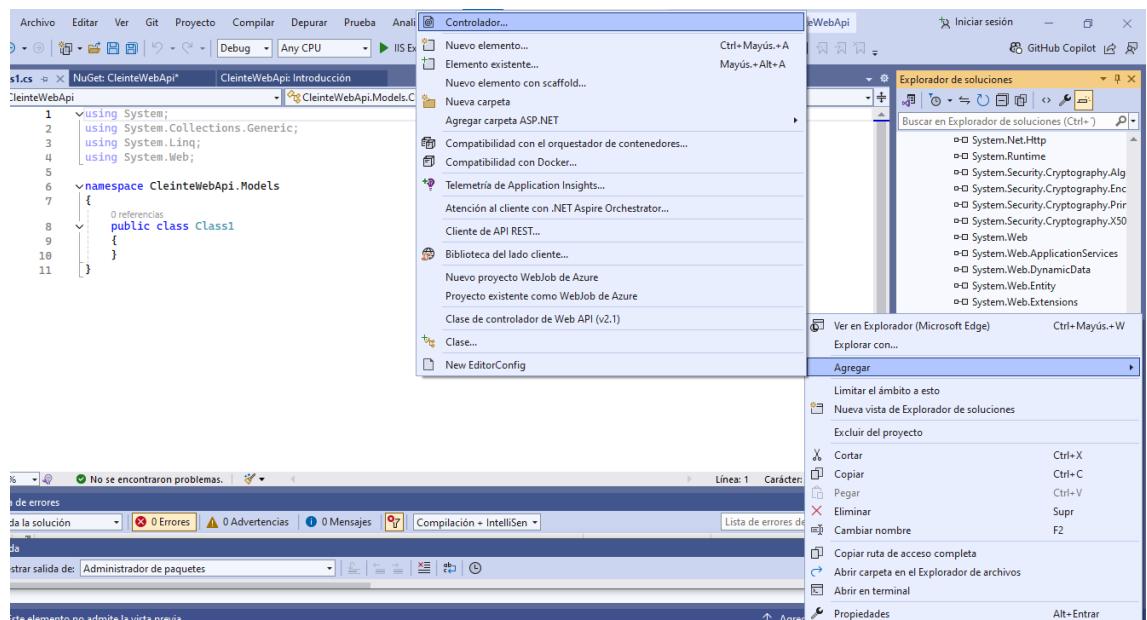
8.



```
Class1.cs NuGet: CleinteWebApi* CleinteWebApi: Introducción
CleinteWebApi CleinteWebApi.Models.Class1

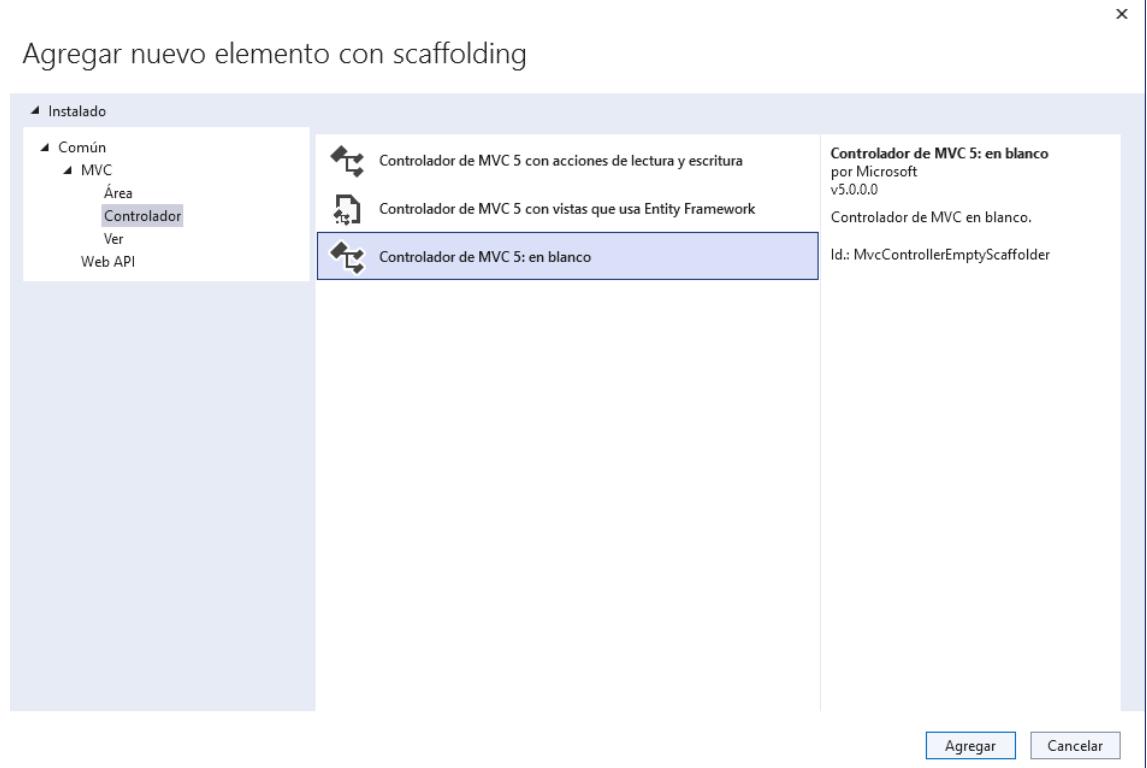
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Web;
5
6  namespace CleinteWebApi.Models
7  {
8      public class Class1
9      {
10  }
```

9.

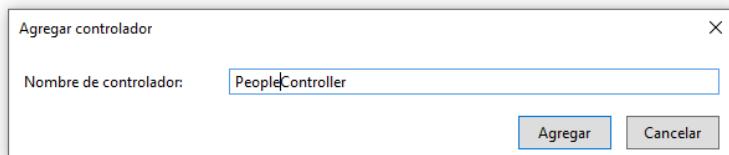


10.

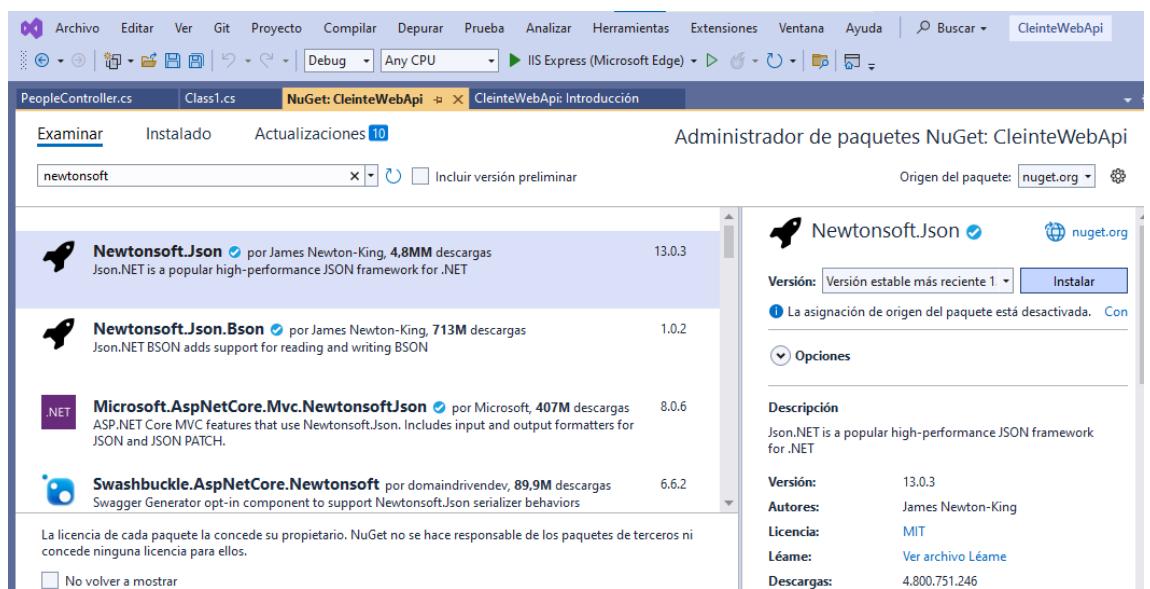
Agregar nuevo elemento con scaffolding



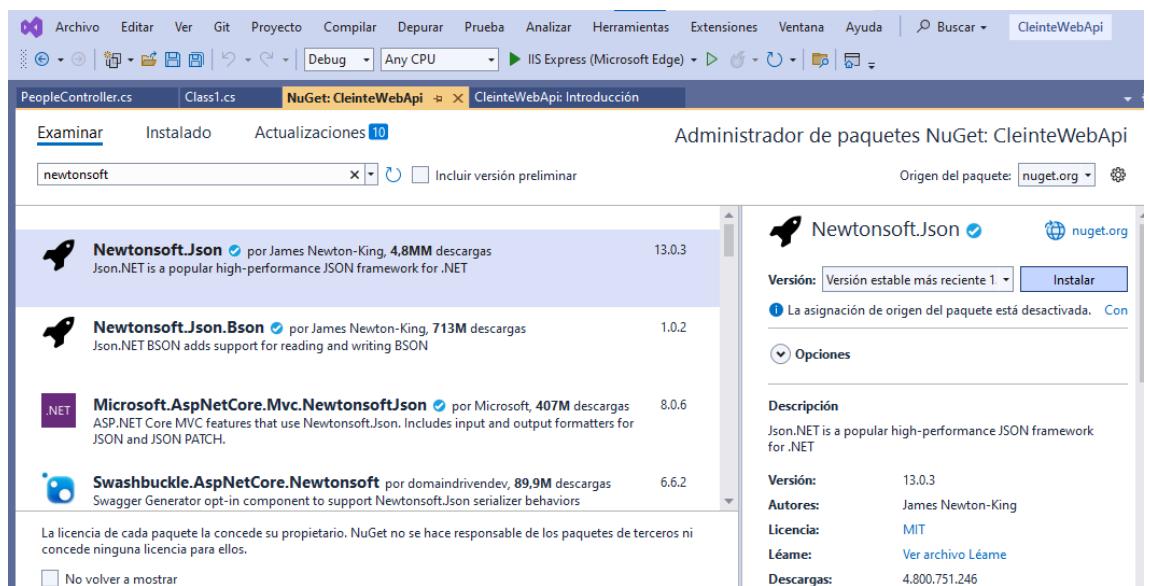
11.



12.



13.

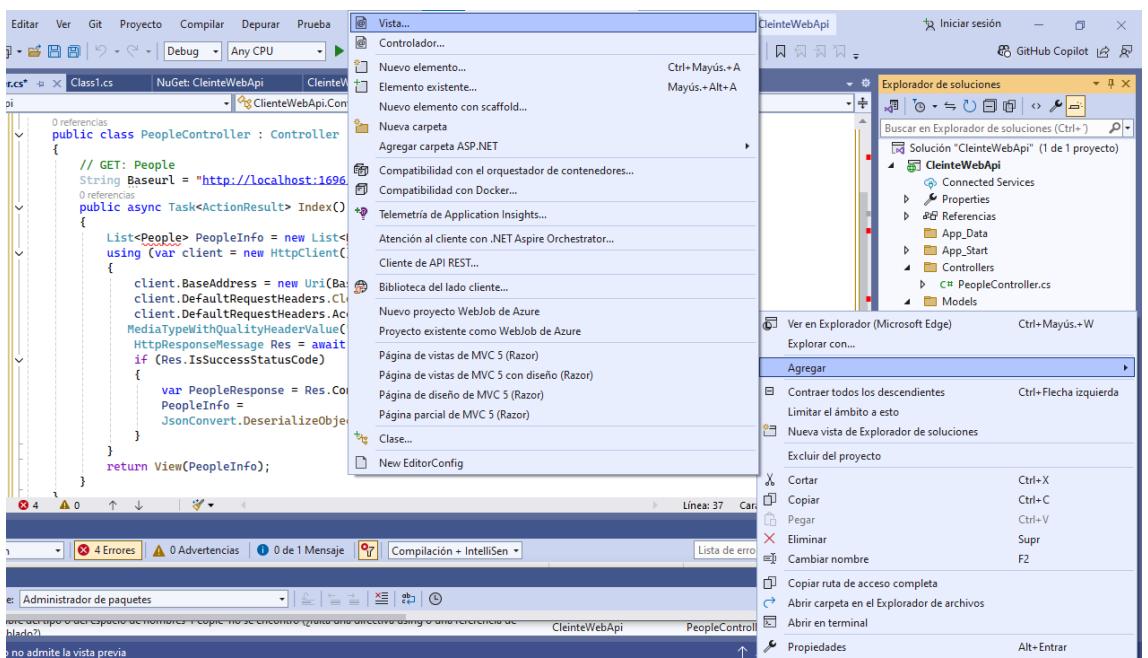


14.

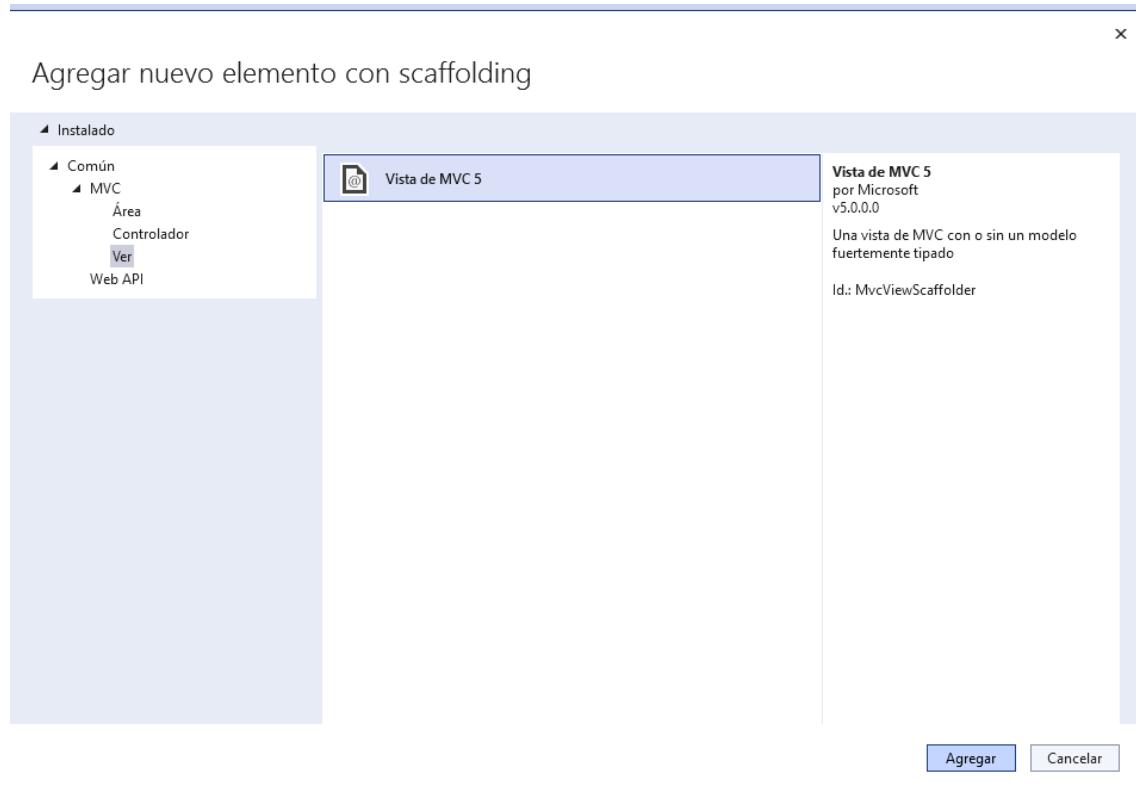


```
13  public class PeopleController : Controller
14  {
15      // GET: People
16      String Baseurl = "http://localhost:1696/";
17      public async Task<ActionResult> Index()
18      {
19          List<People> PeopleInfo = new List<People>();
20          using (var client = new HttpClient())
21          {
22              client.BaseAddress = new Uri(Baseurl);
23              client.DefaultRequestHeaders.Clear();
24              client.DefaultRequestHeaders.Accept.Add(new
25                  MediaTypeWithQualityHeaderValue("application/json"));
26              HttpResponseMessage Res = await client.GetAsync("api/people/");
27              if (Res.IsSuccessStatusCode)
28              {
29                  var PeopleResponse = Res.Content.ReadAsStringAsync().Result;
30                  PeopleInfo =
31                      JsonConvert.DeserializeObject<List<People>>(PeopleResponse);
32              }
33          }
34      }
35      return View(PeopleInfo);
36  }
```

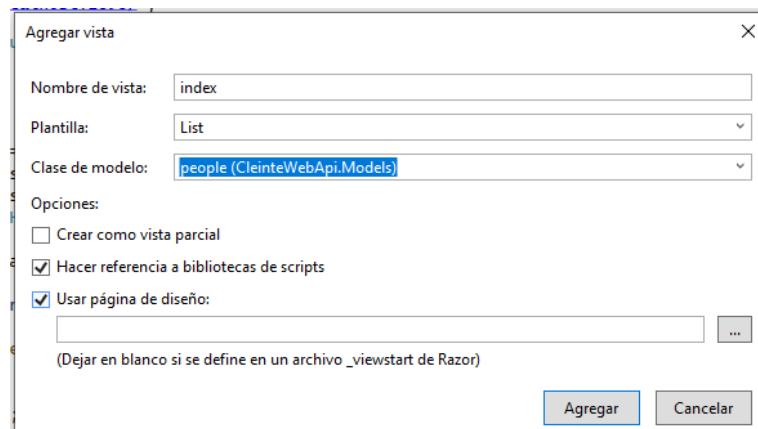
15.



16.



17.



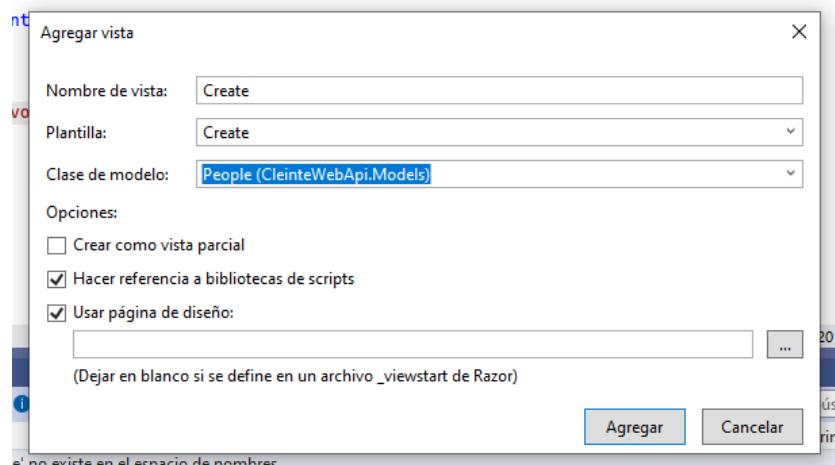
18. Reemplazar People por Home en el routeConfig

```

7
8     <namespace CleinteWebApi
9     {
10        <public class RouteConfig
11        {
12            <public static void RegisterRoutes(RouteCollection routes)
13            {
14                routes.IgnoreRoute("{resource}.axd/{*pathInfo}");
15
16                routes.MapRoute(
17                    name: "Default",
18                    url: "{controller}/{action}/{id}",
19                    defaults: new { controller = "people", action = "Index", id = UrlParameter.Optional }
20                );
21            }
22        }
23    }
24

```

19. Añadimos la opción crear



```

1     @model ClientWebApi.Models.People
2
3     @{
4         ViewBag.Title = "Create";
5     }
6
7     <h2>Create</h2>
8
9     <using (Html.BeginForm())>
10    {
11        @Html.AntiForgeryToken()
12
13        <div class="form-horizontal">
14            <h4>People</h4>
15            <hr />
16            @Html.ValidationSummary(true, "", new { @class = "text-danger" })
17
18            <div class="form-group">
19                @Html.LabelFor(model => model.Id, htmlAttributes: new { @class = "control-label col-md-2" })
20                <div class="col-md-10">
21                    @Html.EditorFor(model => model.Id, new { htmlAttributes = new { @class = "form-control" } })
22                    @Html.ValidationMessageFor(model => model.Id, "", new { @class = "text-danger" })
23                </div>
24            </div>
25
26            <div class="form-group">
27                @Html.LabelFor(model => model.Name, htmlAttributes: new { @class = "control-label col-md-2" })
28                <div class="col-md-10">
29

```

Consumir una API

Create

People

Id

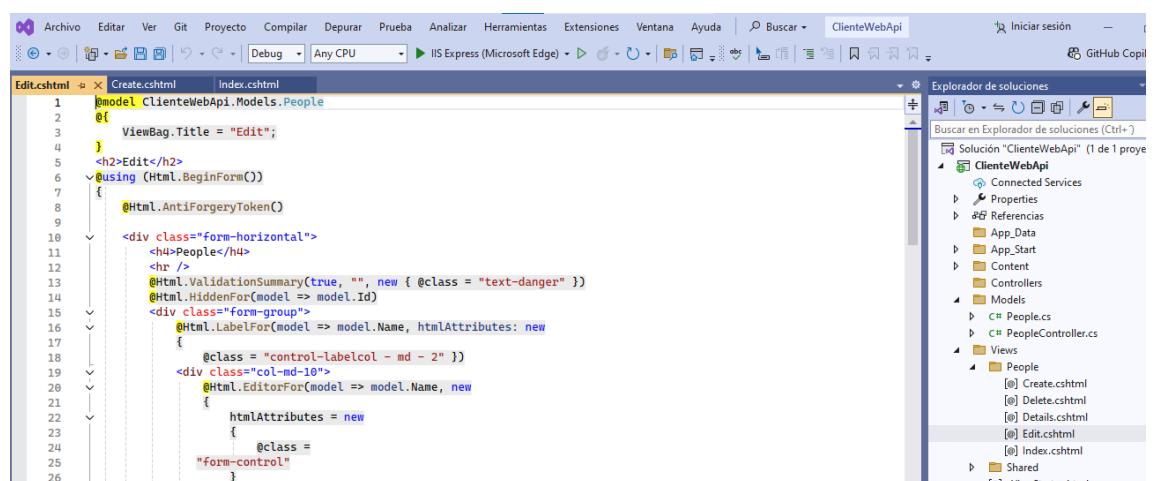
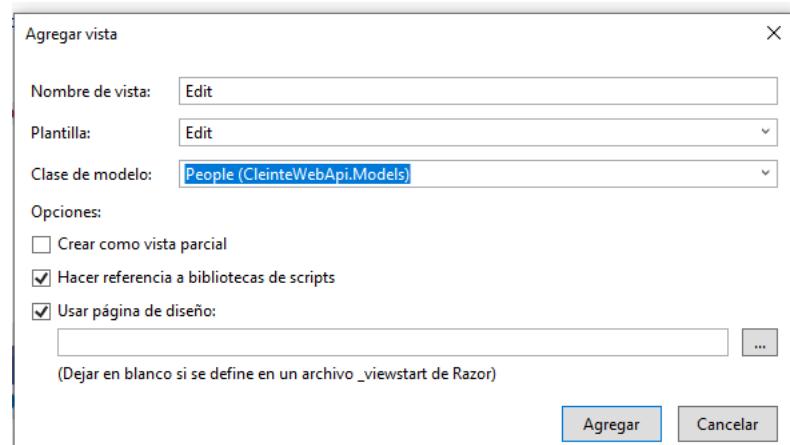
Name

Age

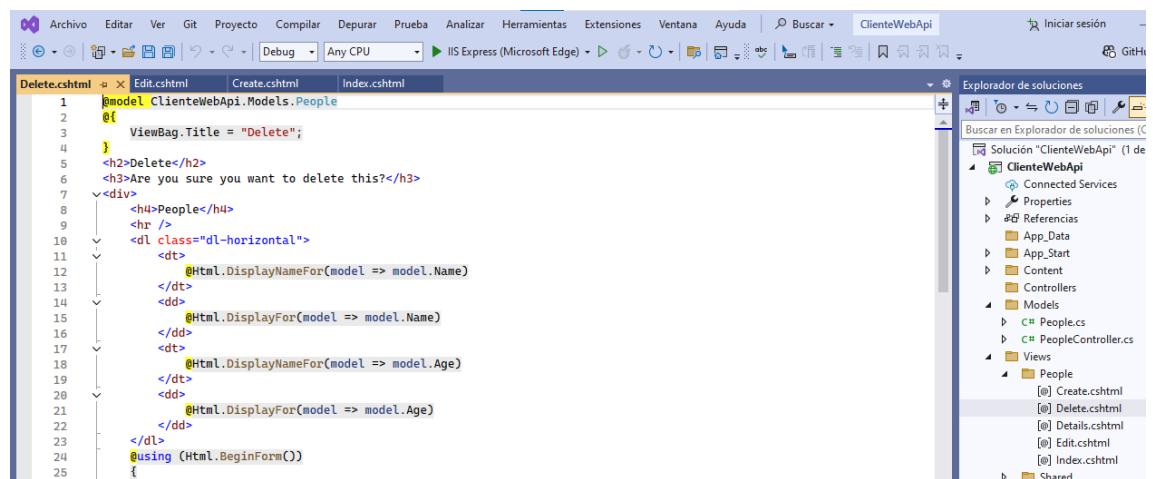
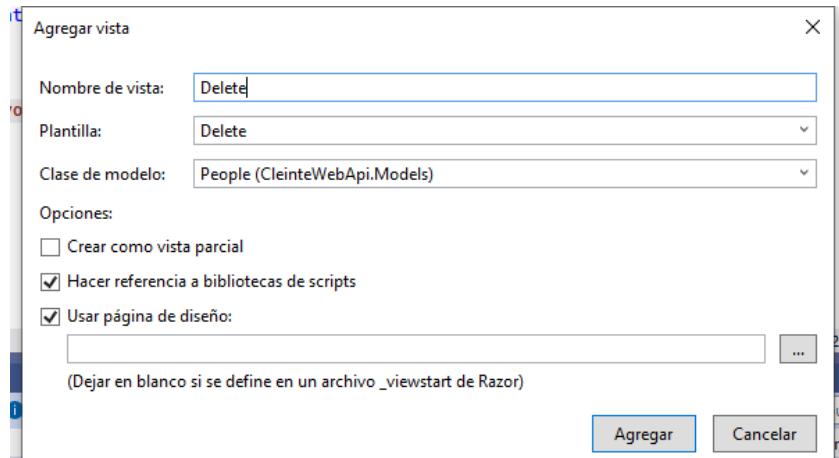
Create

[Back to List](#)

20. Añadimos la opción actualizar



21. Añadimos la opción eliminar



4. Conclusiones

En conclusión, el desarrollo de un sistema de gestión de contenido web que integre un frontend atractivo y fácil de usar con un backend robusto y escalable es fundamental para proporcionar una experiencia de usuario satisfactoria y garantizar el funcionamiento eficiente y seguro del sistema.

Al alcanzar el objetivo general de este proyecto, se logra crear una plataforma que permite a los usuarios interactuar de manera intuitiva con la información, facilitando la creación, lectura, actualización y eliminación de contenido. El frontend desarrollado proporciona una interfaz amigable que mejora la experiencia del usuario y fomenta la participación activa en el sistema.

Por otro lado, el backend implementado garantiza la seguridad y la confiabilidad de los datos, así como la escalabilidad del sistema para adaptarse a futuros cambios y crecimientos en la demanda. Esto se logra mediante la implementación de prácticas de desarrollo seguras, la gestión eficiente de las solicitudes de los usuarios y el diseño de una arquitectura flexible y escalable.

5. Bibliografía

- Microsoft. (2022). Visual Studio Code. Recuperado de <https://code.visualstudio.com/>
- W3Schools. (s. f.). Backend. Recuperado de https://www.w3schools.com/whatis/whatis_backend.asp
- Mozilla. (s. f.). Frontend web development. Recuperado de https://developer.mozilla.org/en-US/docs/Learn/Front-end_web_developer
- Postman. (s. f.). Postman. Recuperado de <https://www.postman.com/>
- Oracle. (s. f.). SQL (Structured Query Language). Recuperado de <https://www.oracle.com/database/what-is-sql.html>

6. Anexos



Datos

Ingrese un nuevo Elemento

Nombre	Edad	Acciones

© 2024 - My ASP.NET Application

A screenshot of a web browser window. The title bar at the top says "Consumir una API".

Create

People

Id 1	Name Cristian	Age 21
----------------	-------------------------	------------------

Create

[Back to List](#)

© 2024 - My ASP.NET Application

A screenshot of a web browser window. The title bar at the top says "Consumir una API".

Datos

Ingrese un nuevo Elemento

Nombre	Edad	Acciones
Cristian	21	Edit Details Delete

© 2024 - My ASP.NET Application

100 %

Results Messages

	Id	name	age
1	1	Cristan	21

Informe Azure

1. Tema

Api Management Service

2. Objetivos

Consumir una API

3. Introducción

Los servicios de gestión de APIs (API Management Services) juegan un papel fundamental en la arquitectura moderna de software, facilitando la exposición, protección y gestión de interfaces de programación de aplicaciones (APIs) de manera eficiente y segura. En el contexto actual de la integración de sistemas y la proliferación de aplicaciones distribuidas, los servicios de gestión de APIs se han convertido en una herramienta indispensable para las empresas que buscan optimizar la conectividad y la colaboración entre sistemas y aplicaciones.

En esencia, un servicio de gestión de APIs actúa como una capa intermedia entre los desarrolladores de aplicaciones y las APIs subyacentes que proporcionan funcionalidades específicas. Proporciona un conjunto de herramientas y funcionalidades que permiten a las organizaciones publicar, controlar, asegurar y analizar el acceso a sus APIs de una manera controlada y segura. Esto incluye la capacidad de definir políticas de acceso, autenticación, autorización, límites de uso, y monitoreo del rendimiento y la salud de las APIs.

Entre las características clave de los servicios de gestión de APIs se encuentran la capacidad de proporcionar puertas de enlace (gateways) que gestionan el tráfico entrante y saliente de las APIs, permitiendo la transformación de protocolos y formatos de datos según sea necesario. Además, ofrecen capacidades de seguridad robustas, como la autenticación mediante tokens y la encriptación de datos, para proteger las APIs contra accesos no autorizados y ataques cibernéticos.

Estos servicios no solo simplifican la gestión operativa de las APIs, sino que también promueven la reutilización y la gobernanza al proporcionar un catálogo centralizado de APIs disponibles, facilitando su descubrimiento y uso por parte de desarrolladores internos y externos. Esto fomenta la colaboración y la innovación al permitir a las empresas crear ecosistemas de aplicaciones más ágiles y conectados.

4. Marco Teórico

API Management Service

API Management, o gestión de API, se refiere al proceso de creación, publicación y gestión de conexiones API dentro de un entorno empresarial y multinube. Más que un simple lugar para que vivan estas conexiones API, la administración de API ofrece una

plataforma unificada y escalable que permite a las empresas compartir y socializar sus configuraciones de API mientras controla el acceso, recopila y analiza estadísticas de uso y aplica las políticas de seguridad asociadas.[1]

Exposición y Descubrimiento de APIs:

Portal Developer: Un componente fundamental es el portal del desarrollador, que proporciona una interfaz amigable donde los desarrolladores pueden descubrir APIs disponibles, acceder a documentación detallada, y probar las APIs antes de integrarlas en sus aplicaciones.

Catálogo de APIs: Permite la catalogación centralizada de todas las APIs disponibles en la organización, facilitando la búsqueda y el uso por parte de desarrolladores internos y externos.

Gestión de Versiones: Capacidad para gestionar múltiples versiones de una misma API de manera que los usuarios puedan migrar a nuevas versiones sin interrupciones.

Seguridad y Control de Acceso: Autenticación y Autorización: Implementación de mecanismos de autenticación robustos como OAuth, OpenID Connect, o integración con sistemas de directorio corporativos.

Gestión de Tokens: Emisión y validación de tokens de acceso para asegurar que solo aplicaciones y usuarios autorizados puedan consumir las APIs.

Políticas de Seguridad: Definición de políticas de seguridad para proteger las APIs contra ataques como denegación de servicio (DDoS), inyecciones SQL, y otros tipos de amenazas.

Gestión del Ciclo de Vida de las APIs:

Diseño y Documentación: Herramientas para diseñar APIs de manera consistente con estándares como RESTful o GraphQL, y documentarlas adecuadamente con descripciones de parámetros, ejemplos de uso, y casos de prueba.

Gestión de Versiones: Capacidad para gestionar múltiples versiones de una API de manera que los usuarios puedan migrar a nuevas versiones sin interrupciones.

Monitorización y Análisis: Herramientas para monitorizar el rendimiento de las APIs en tiempo real, recolectar métricas como tiempo de respuesta y número de solicitudes, y generar informes para optimización continua.

Modelos de Monetización: Ofrecer modelos de tarifas flexibles como pago por uso, suscripción mensual o anual, y planes freemium para permitir a las organizaciones

Monetización y Métricas de Uso:

monetizar el acceso a sus APIs.

Análisis de Uso: Generar informes detallados sobre el uso de las APIs por parte de desarrolladores y aplicaciones consumidoras, lo que permite a las organizaciones tomar decisiones informadas sobre la optimización de recursos y la expansión de servicios.[2]

Transformación y Orquestación de Datos:

Transformación de Protocolos y Formatos: Capacidad para transformar datos entre diferentes protocolos como JSON, XML, o SOAP, según las necesidades de las aplicaciones consumidoras.

Orquestación de Servicios: Integración y orquestación de múltiples servicios y APIs subyacentes para ejecutar flujos de trabajo complejos y optimizar la eficiencia operativa.

Integración con Ecosistemas y DevOps:

Integración Continua y Despliegue Continuo (CI/CD): Soporte para integración y despliegue automatizado de APIs en entornos DevOps, facilitando la entrega continua y la colaboración entre equipos de desarrollo y operaciones.

Compatibilidad y Flexibilidad: Adaptabilidad para integrarse con diferentes herramientas y plataformas de desarrollo como Kubernetes, Docker, y servicios de nube pública como AWS, Azure, o Google Cloud Platform.[3]

Componentes Clave

1. Gateway de API: Es el punto de entrada para todas las API disponibles en la plataforma de gestión. Actúa como un intermediario que gestiona las solicitudes de los clientes y las redirige a las API correspondientes.
2. Portal de Desarrolladores: Es una interfaz para que los desarrolladores descubran, comprendan y utilicen las APIs proporcionadas. Suele incluir documentación detallada, ejemplos de código y capacidades de prueba.
3. Seguridad y Control de Acceso: Permite la autenticación y autorización de las solicitudes de API para proteger los datos y garantizar el cumplimiento de las políticas de seguridad de la organización.
4. Monitoreo y Análisis: Proporciona visibilidad en tiempo real sobre el rendimiento de las APIs, la utilización de recursos y el comportamiento de los usuarios, permitiendo la optimización y la toma de decisiones basadas en datos.
5. Gestión del Ciclo de Vida de las APIs: Facilita la creación, versión, modificación y desactivación de APIs de manera controlada, asegurando la coherencia y la compatibilidad entre versiones. [4][5]

Beneficios

- Agilidad y Flexibilidad: Facilita la integración de sistemas y la creación de nuevas aplicaciones de forma rápida y eficiente.

- Seguridad Mejorada: Control centralizado sobre las políticas de seguridad y acceso.
- Análisis y Optimización: Mejora continua basada en datos sobre el uso y el rendimiento de las APIs.
- Experiencia del Desarrollador: Portal de desarrolladores intuitivo que fomenta la adopción y colaboración.
- Cumplimiento Normativo: Ayuda a cumplir con regulaciones como GDPR, PCI-DSS, entre otras.

Implementación

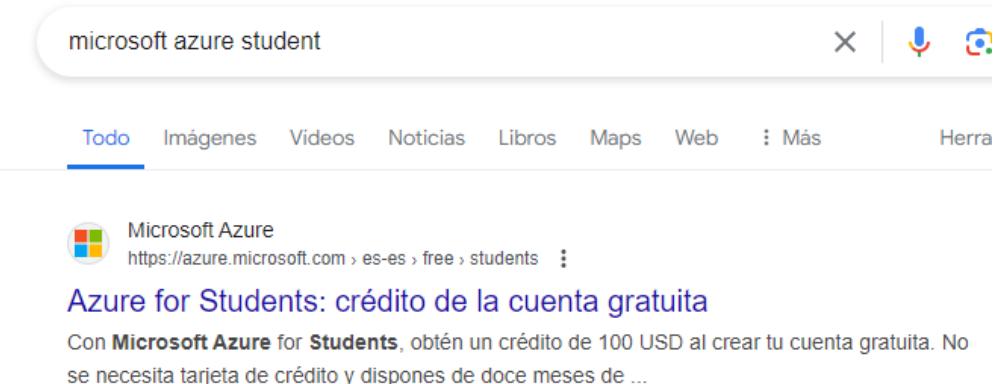
- On-premise vs Cloud: Puede ser implementado tanto en infraestructuras locales como en la nube, dependiendo de las necesidades y la estrategia de la organización.
- Integración con Herramientas Existentes: Debe integrarse con sistemas de gestión de identidades, sistemas de monitoreo y otras herramientas de administración de TI existentes.

Tendencias y Futuro

- Microservicios y Arquitecturas Basadas en APIs: Cada vez más organizaciones adoptan arquitecturas orientadas a APIs para facilitar la innovación y la escalabilidad.
- Automatización y DevOps: Integración con prácticas de desarrollo ágil y DevOps para mejorar la entrega continua y la automatización de procesos.[4]

5. Procedimiento realizado

1. Entramos a la página de azure



2.

Verificación académica

Para empezar, escriba su nombre según los registros de la escuela. Seleccione el país o región de la escuela y escriba el nombre de la escuela. Escriba su fecha de nacimiento según los registros de la escuela.

Nombre
Pablo Andres

Apellidos
Orozco Centeno

País o región
Ecuador

Si su país no aparece en la lista, la oferta no está disponible en su región. [Más información](#)

Nombre de la escuela
UNACH

El nombre de la escuela le ayudará a proporcionar a Microsoft información adicional para la verificación. Si está disponible, introduzcalo aquí.

Garantizar la seguridad de la cuenta

Microsoft Authenticator

Digitalización del código QR

Use la aplicación Microsoft Authenticator para escanear el código QR. Así, la aplicación Microsoft Authenticator y la cuenta quedarán emparejadas.

Después de escanear el código QR, elija "Siguiente".

¿No puede escanear la imagen?

Atrás Siguiente

3.

Microsoft Azure

Buscar recursos, servicios y documentos (G+/)

Inicio > Education | Información general

Empezar Información general

Información general

Recursos de aprendizaje

¿Necesita ayuda?

Detalles de la oferta para estudiantes

Créditos disponibles
100 US\$ de 100 US\$

Días hasta que expire el crédito
365
Expira el 09/07/2025

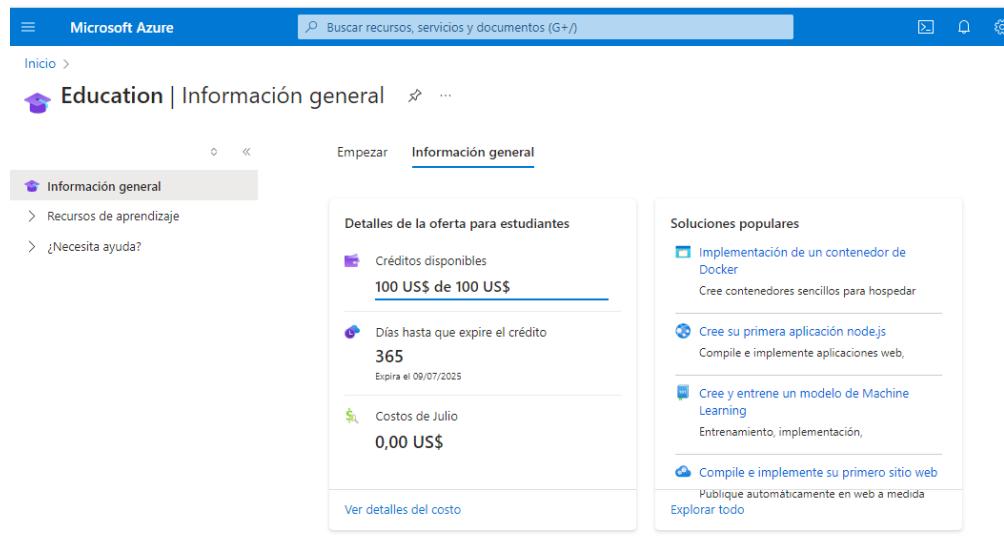
Costos de Julio
0,00 US\$

Ver detalles del costo

Soluciones populares

- Implementación de un contenedor de Docker
- Cree contenedores sencillos para hospedar
- Cree su primera aplicación node.js
- Compile e implemente aplicaciones web,
- Cree y entrene un modelo de Machine Learning
- Entrenamiento, implementación,
- Compile e implemente su primer sitio web
- Publique automáticamente en web a medida

Explorar todo



4.

api man

Todo Servicios (34) Marketplace (10) Más (4)

Servicios

Ver más

Servicios API Management



5.

Microsoft Azure

Buscar recursos, servicios y documentos (G+/)

Inicio >

Servicios API Management

Universidad Nacional de Chimborazo (unach.edu.ec)

+ Crear ⌂ Recover ⌂ Administrar vista ⌂ Actualizar ⌂ Exportar a CSV ⌂ Abrir consulta | Asignar etiquetas

Filtrar por cualquier campo Suscripción es igual a todo Grupo de recursos es igual a todo × Ubicación es igual a todo × Agregar filtro

⚠ Support for the single-tenant v1 (STv1) platform ends on 8/31/24. Migrate instances before that date to the new platform version (STv2) for continued support and access to new features. See this document for more information.

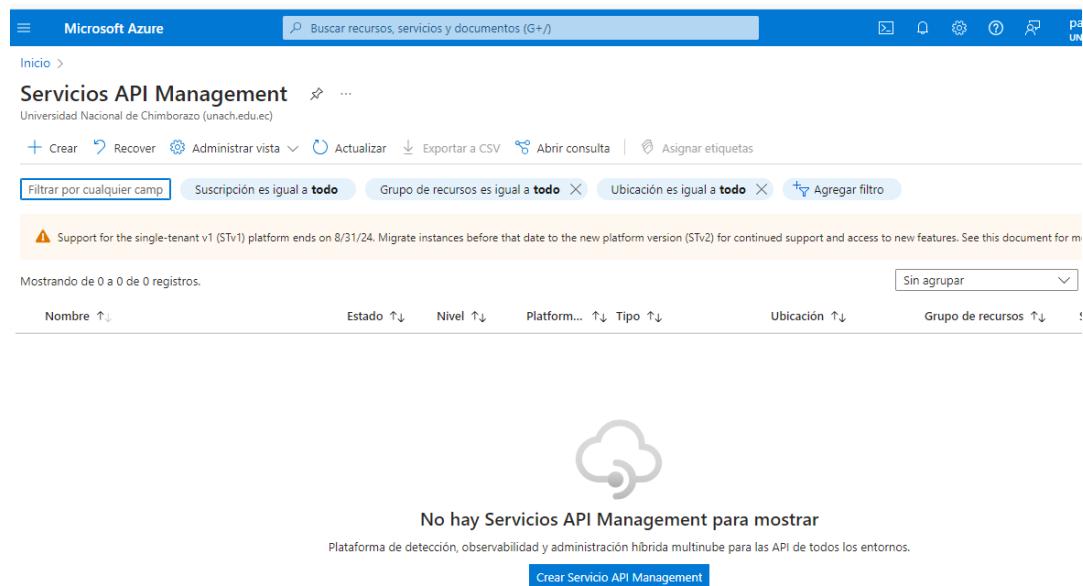
Mostrando de 0 a 0 de 0 registros.

Nombre ↑↓ Estado ↑↓ Nivel ↑↓ Platform... ↑↓ Tipo ↑↓ Ubicación ↑↓ Grupo de recursos ↑↓

No hay Servicios API Management para mostrar

Plataforma de detección, observabilidad y administración híbrida multinube para las API de todos los entornos.

Crear Servicio API Management



6.

Create API Management service

Servicio API Management

Seleccione la suscripción para administrar los recursos y costos implementados. Use grupos de recursos, como carpetas, para organizar y administrar todos los recursos.

Suscripción * ⓘ

Azure for Students

Grupo de recursos * ⓘ

(Nuevo) practicaWS

[Crear nuevo](#)

Detalles de la instancia

Región * ⓘ

(US) West US 2

Nombre del recurso *

apiazuremp24

Organization name * ⓘ

PABLO

Correo electrónico del administrador * ⓘ

pablo.orozco@unach.edu.ec

Plan de tarifa

Los planes de tarifa de API Management tienen distintas capacidades de cálculo por unidad y ofrecen conjuntos de características diferentes, por ejemplo, compatibilidad con redes virtuales, implementaciones en varias regiones o puertas de enlace autohospedadas. Para dar cabida a más solicitudes de API, considere agregar unidades de servicio de API Management en su lugar. [Más información ↗](#)

[Revisar y crear](#)

[< Anterior](#)

[Siguiente: Monitor + secure >](#)

7.

Create API Management service

Servicio API Management

Datos básicos

Monitor + secure

Red virtual

Identidad administrada

[Etiquetas](#)

Revisar e instalar

Las etiquetas son pares de nombre y valor que permiten clasificar los recursos y ver una facturación consolidada mediante la aplicación de la misma etiqueta a varios recursos y grupos de recursos. [Más información acerca de las etiquetas](#)

Tenga en cuenta que, si crea etiquetas y después cambia la configuración de recursos en otras pestañas, las etiquetas se actualizarán automáticamente.

Nombre ⓘ

Valor ⓘ

Recurso

 :

2 seleccionados



[Revisar y crear](#)

[< Anterior](#)

[Siguiente: Revisar e instalar >](#)

Create API Management service ...

Servicio API Management

Al hacer clic en "Crear", (a) acepto los términos legales y las declaraciones de privacidad relacionadas con las ofertas de Marketplace indicadas más arriba; (b) autorizo a Microsoft a facturar a mi método de pago actual las cuotas relacionadas con las ofertas con la misma frecuencia de facturación que mi suscripción a Azure; y (c) autorizo a Microsoft a compartir mis datos de contacto, de transacción y de uso con los proveedores de las ofertas para actividades de soporte técnico, facturación y transacciones. Microsoft no proporciona derechos sobre ofertas de terceros. Para obtener más información, consulte los [Términos de Azure Marketplace](#).

Datos básicos

Suscripción	Azure for Students
Grupo de recursos	practicaWS
Región	West US 2
Nombre del recurso	apiazuremp24
Organization name	PABLO
Correo electrónico del administrador	pabloa.orozco@unach.edu.ec
Plan de tarifa	Developer
Unidades	1
Zonas de disponibilidad	-

Supervisión

[Crear](#)

[< Anterior](#)

[Siguiente >](#)

[Descargar una plantilla para la automatización](#)

8.

The screenshot shows the Azure portal interface for managing an API Management service. The top navigation bar includes 'Dashboard', 'Create a resource', 'Search', 'My services', and 'Logout'. Below the navigation, there's a search bar and a breadcrumb trail: 'Implementación' > '668daeb93674328b69af0cd9 | Información general'. The main content area displays the 'Información general' tab under 'Implementación'. It shows the implementation status as 'En curso' (In progress). Key details listed include: Nombre de implementación: 668daeb93674328b69af..., Hora de inicio: 9/7/2024, 16:42:32; Suscripción: Azure for Students; Grupo de recursos: practicaWS; Id. de correlación: 6604e20c-2cdc-4f8b-8381-86939d130641. A table titled 'Detalles de implementación' shows one entry: Recurso: apiazuremp24, Tipo: Microsoft.ApiManagement..., Estado: Created. At the bottom, there are links for 'Enviar comentarios' and 'Cuéntenos su experiencia con la implementación'.

9.

The screenshot shows the Azure portal interface for a deployment. At the top, it displays the deployment ID: 668daeb93674328b69af0cd9 and the title "Información general". Below this, there's a search bar and several action buttons: Eliminar, Cancelar, Volver a implementar, Descargar, and Actualizar. A prominent green checkmark indicates that the implementation has completed successfully. The "Detalles de implementación" section provides basic information: Nombre de implementación: 668daeb93674328b69..., Suscripción: Azure for Students, Grupo de recursos: practicaWS, Hora de inicio: 9/7/2024, 16:42:32, and Id. de correlación: 6604e20c-2cdc-4f8b-8381-86939d130641. There are also sections for "Pasos siguientes" and a "Ir al recurso" button. Below the main content, there are links to "Enviar comentarios" and "Cuéntenos su experiencia con la implementación".

10.

The screenshot shows the Azure API Management service interface. The left sidebar lists various management options like Settings, APIs, Workspaces, API, Products, Subscriptions, etc. The main area is titled "Define a new API" and features three large cards: "HTTP" (Manually define an HTTP API), "WebSocket" (Streaming, full-duplex communication with a WebSocket server), and "GraphQL" (Access the full data from a single endpoint). Below these cards, there's a section titled "Create from definition" with three icons: a gear, an angle bracket, and a magnifying glass. The URL in the browser bar is portal.azure.com/#@unach.edu.ec/resource/subscriptions/32804379-a281-42f3-a7fc-dcdc79c8b1fe/resourceGroups/practicaWS/providers/Microsoft.ApiManagement/service/apiazuremp24

11.

The screenshot shows the Azure API Management service interface, specifically the "Design" tab for an API definition named "Echo API". The left sidebar is similar to the previous screenshot. The main area shows the "General" configuration section with fields for Display name (Echo API), Name (echo-api), Description, Web service URL (http://echoapi.cloudapp.net/api), URL scheme (HTTPS selected), and API URL suffix (echo). The "REVISION 1" header indicates the creation date: CREATED Jul 9, 2024, 4:51:26 PM. At the bottom, there are "Save" and "Discard" buttons.

12.

The screenshot shows the Azure API Management portal interface. On the left, there's a sidebar with 'Search APIs', 'Filter by tags', 'Group by tag', '+ Add API', and 'All APIs' sections. Under 'All APIs', 'Echo API' is selected. The main area is titled 'REVISION 1' (CREATED Jul 9, 2024, 4:51:26 PM). It has tabs for 'Design', 'Settings', 'Test', 'Revisions (1)', and 'Change log'. The 'Design' tab is active, showing an operation named 'create-resource' with a 'POST' method and URL '/resource'. The 'Description' field contains a note about transforming JSON to XML. The 'Tags' field has 'Booking' entered. Below the main form are tabs for 'Template', 'Query', 'Headers', 'Request', and 'Responses'. At the bottom are 'Save' and 'Discard' buttons.

13. Probamos en Postman con JSON

The screenshot shows the Postman application interface. The left sidebar lists 'REST API basics: CRUD, test & variable' with actions: 'GET Get data', 'POST Post data', 'PUT Update data', and 'DEL Delete data'. The main area shows a 'POST' request to 'http://echoapi.cloudapp.net/api'. The 'Body' tab is selected, showing a JSON payload:

```
1 {
2   "id": 1,
3   "plato": "papas",
4   "bebida": "soda"
5 }
```

Below the request, the 'Body' tab is highlighted. The status bar at the bottom indicates 'Status: 200 OK'.

14. Probamos en Postman con XML

The screenshot shows the Postman application interface, similar to the previous one but with XML selected. The left sidebar lists 'REST API basics: CRUD, test & variable' with actions: 'GET Get data', 'POST Post data', 'PUT Update data', and 'DEL Delete data'. The main area shows a 'POST' request to 'http://echoapi.cloudapp.net/api'. The 'Body' tab is selected, showing an XML payload:

```
1 <Document>
2   <id>1</id>
3   <plato>papas</plato>
4   <bebida>soda</bebida>
5 </Document>
```

Below the request, the 'Body' tab is highlighted. The status bar at the bottom indicates 'Status: 200 OK'.

15.

The screenshot shows the Microsoft Azure API Management interface. On the left, there's a sidebar with various navigation options like 'Información general', 'Registro de actividad', 'Control de acceso (IAM)', 'Etiquetas', 'Diagnosticar y solucionar problemas', 'Eventos', 'Settings', and 'APIs' (which is expanded to show 'Workspaces (vista previa)', 'API', and 'Productos'). The 'Suscripciones' option is selected. The main area is titled 'apiazuremp24 | Suscripciones'. A modal window titled 'Suscripción de acceso total' is open, containing a warning message: 'La suscripción de acceso total permite el acceso a cada API de esta instancia de API Management y solo la deberían usar los usuarios autorizados. No la use nunca para acceder a la API de rutina o insertar la clave de acceso total en las aplicaciones cliente.' with 'Aceptar' and 'Cancelar' buttons. At the top of the main page, there are filters for 'Estado' (Todos, Pendiente de aprobación) and 'Ámbito' (Todos).

Nombre para mostrar	Clave principal	Clave secundaria	Ámbito	Estado
.....	Producto: Starter	Activo
.....	Producto: Unlimited	Activo
Built-in all-access sub...	4eb89357a0ef4...	95a0373c3d214...	Servicio	Activo

16.

The screenshot shows the Postman application interface. At the top, it says 'HTTP REST API basics: CRUD, test & variable / Get data'. Below that, a 'POST' request is selected with the URL 'https://apiazuremp24.azure-api.net/echo/resource'. The 'Authorization' tab is active, showing 'Auth Type: API Key'. A note says: 'Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about [variables](#)'. Below this, there are fields for 'Key' (containing 'Ocp-Apim-Subscription-Key') and 'Value' (containing '4eb89357a0ef4d7da000d88dbcdc9c13'). At the bottom, the status bar shows 'Status: 200 OK Time: 386 ms Size: 581 B' and 'Save as example'.

17.

The screenshot shows the Postman interface. The URL is `https://apiazuremp24.azure-api.net/echo/resource`. The method is set to `POST`. The body contains the following JSON:

```
1 {
2   "id":1,
3   "plato":"papas",
4   "bebida":"soda"
5 }
```

The response status is `200 OK`, time `1581 ms`, size `8.51 KB`. The XML response is as follows:

```
1 <Document>
2   <id>1</id>
3   <plato>papas</plato>
4   <bebida>soda</bebida>
5 </Document>
```

18. Creamos una aplicación ASP Net core Web API Deploy to Azure App Service.

The screenshot shows the Azure portal's App Services blade. It displays a list of resources with the following columns: Nombre, Estado, Ubicación, Plan de tarifa, Plan de App Service, Suscripción, and Tipo de There are no resources listed under 'Mostrando de 0 a 0 de 0 registros.'

19.

Microsoft Azure Buscar recursos, servicios y documentos (G+/)

Inicio > App Services >

Crear aplicación web

carpetas para organizar y administrar todos los recursos.

Suscripción * ⓘ Azure for Students ▼

Grupo de recursos * ⓘ (Nuevo) practicaWS24 ▼
[Crear nuevo](#)

Detalles de instancia

Nombre * pronosticotiempoapi24 .azurewebsites.net ✓

Pruebe un nombre de host predeterminado único (versión preliminar). [Más información sobre esta actualización ↗](#)

Publicar * Código Contenedor Aplicación web estática

Pila del entorno en tiempo de ejecución * Selección de una pila del entorno en tiempo de ejecución ▼

Sistema operativo Linux Windows

Región * East US ▼

Microsoft Azure Buscar recursos, servicios y documentos (G+/)

Inicio > App Services >

Crear aplicación web

Pila del entorno en tiempo de ejecución * .NET 6 (LTS) ▼

Sistema operativo * Linux Windows

Región * East US ▼

ⓘ ¿No encuentra su plan de App Service? Pruebe otra región o seleccione su App Service Environment.

Planes de precios

El plan de tarifa de App Service determina la ubicación, las características, los costos y los recursos del proceso asociados a la aplicación. [Más información ↗](#)

Plan de Linux (East US) * ⓘ (Nuevo) ASP-practicaWS24-b2d0 ▼
[Crear nuevo](#)

Plan de precios Gratis F1 (Infraestructura compartida) ▼
[Explorar planes de precios](#)

Inicio > App Services >

Crear aplicación web

...

Datos básicos **Implementación** Redes Supervisión y protección Etiquetas Revisar y crear

Configuración de implementación continua

Configure la implementación continua para implementar código desde el repositorio de GitHub mediante Acciones de GitHub con facilidad. [Más información](#)

Implementación continua

Deshabilitar Habilitar

! No se puede configurar la implementación con Acciones de GitHub durante la creación de aplicaciones con las selecciones del sistema operativo y el plan de App Service. Si quiere mantener estas selecciones, puede configurar la implementación con Acciones de GitHub después de crear la aplicación web.

Configuración de GitHub

Configure Acciones de GitHub para insertar contenido en la aplicación siempre que se realicen cambios de código en el repositorio. Nota: La cuenta de GitHub debe tener acceso de escritura al repositorio seleccionado para poder agregar un archivo de flujo de trabajo que administre las implementaciones en la aplicación.

Cuenta de GitHub

[Autorizar](#)

[Revisar y crear](#)

[< Anterior](#)

[Siguiente: Redes >](#)

Crear aplicación web

...

Datos básicos Implementación **Redes** Supervisión y protección Etiquetas Revisar y crear

Las Web Apps se pueden aprovisionar con la dirección de entrada pública a Internet o aislada en una red virtual de Azure. Las Web Apps también pueden aprovisionarse con el tráfico saliente capaz de llegar a los puntos de conexión de una red virtual, regirse por grupos de seguridad de red o verse afectadas por rutas de red virtual. De forma predeterminada, la aplicación está abierta a Internet y no puede acceder a ninguna red virtual. Estos aspectos también se pueden cambiar una vez que se haya aprovisionado la aplicación. [Más información](#)

Habilitar el acceso público *

Activado Desactivado

! La inserción de red solo está disponible en los planes dedicados de App Service Básico, Estándar, Premium, Premium V2 y Premium V3.

Habilitar la inserción de red

Activado Desactivado

[Revisar y crear](#)

[< Anterior](#)

[Siguiente: Supervisión y protección >](#)

Microsoft Azure Buscar recursos, servicios y documentos (G+)

Inicio > App Services >

Crear aplicación web

Datos básicos Implementación Redes **Supervisión y protección** Etiquetas Revisar y crear

Las siguientes características son opcionales y se facturan por separado. Microsoft recomienda habilitarlos para garantizar las protecciones y funcionalidades más sólidas para supervisar y proteger las aplicaciones web.

Application Insights

Azure Monitor application insights is an Application Performance Management (APM) service for developers and DevOps professionals. Enable it below to automatically monitor your application. It will detect performance anomalies, and includes powerful analytics tools to help you diagnose issues and to understand what users actually do with your app. Your bill is based on amount of data used by Application Insights and your data retention settings. [Más información ↗](#)

[Precios de App Insights ↗](#)

Habilitar Application Insights * No Sí

Microsoft Defender for Cloud

Al agregar el plan de Defender para App Service a su suscripción de Azure, obtiene una solución de seguridad nativa de la nube que supervisa los registros, las solicitudes, la instancia de máquina virtual, etc., lo que le ayuda a detectar amenazas y ataques continuos en los recursos. [Más ventajas de Defender para App Service ↗](#)

[Defender for Cloud pricing ↗](#)

20.

Crear aplicación web

Datos básicos Implementación Redes Supervisión y protección Etiquetas **Revisar y crear**

Resumen



Aplicación web
de Microsoft

SKU Gratis

Precio estimado - Gratis

! La autenticación básica para esta aplicación está deshabilitada actualmente y puede afectar a las implementaciones.
[Haga clic aquí para más información.](#)

Detalles

Suscripción	32804379-a281-42f3-a7fc-dcdc79c8b1fe
Grupo de recursos	practicaWS24
Nombre	pronosticotiempoapi24
Publicar	Código
Pila del entorno en tiempo de ejecución	.NET 6 (LTS)

Crear

< Anterior

Siguiente >

[Descargar una plantilla para la automatización](#)

21.

Microsoft Azure | Microsoft.Web-WebApp-Portal-ac50da37-9a0b | Información general

Nombre de implementación: Microsoft.Web-WebApp-Portal...
Suscripción: Azure for Students
Grupo de recursos: practicaWS24

Hora de inicio: 12/7/2024, 15:06:16
Id. de correlación: ec484471-0a55-4b93-abd0-5f02e3

Se completó la implementación

Detalles de implementación

Administre las implementaciones de la aplicación. Recomendado
Proteja la aplicación mediante la autenticación. Recomendado

Pasos siguientes

Enviar comentarios

Cost Management
Obtenga una notificación para permanecer dentro del presupuesto y evitar cargos inesperados en su factura.
[Configurar alertas de costo >](#)

Microsoft Defender for Cloud
Proteja sus aplicaciones e infraestructura.
[Ir a Microsoft Defender for Cloud >](#)

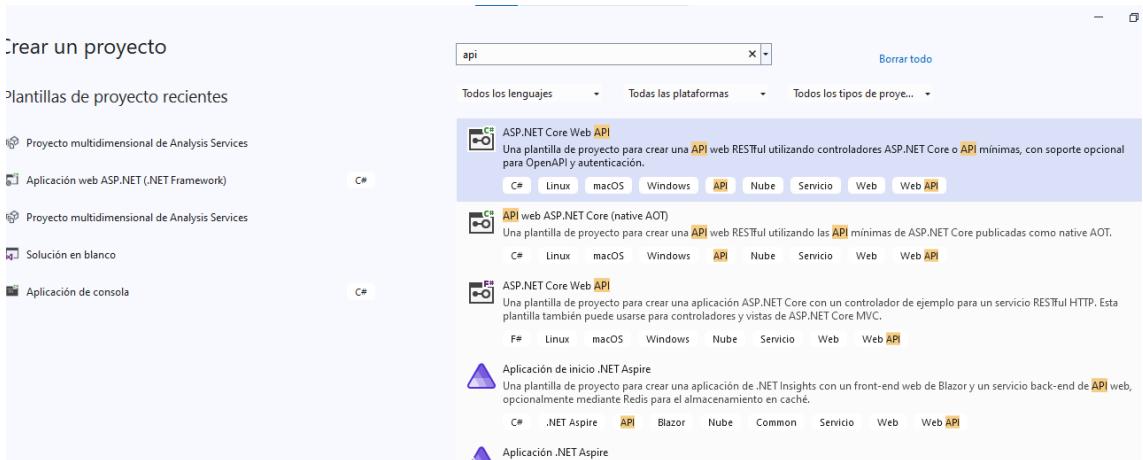
pronosticotiempoapi24 | Configuración

Actualizar Guardar Descartar Salir de los comentarios

Configuración de plataforma

Credenciales de publicación: Activado
Credenciales de publicación: Activado
Estado FTP: Solo FTPS
Versión HTTP: 1.1
Proxy HTTP 2.0: Desactivado

22. Abrimos Visual Studio



23.

Configure your new project

ASP.NET Core Web API | C# | Linux | macOS | Windows | API | Nube | Servicio | Web | Web API

Name of the project

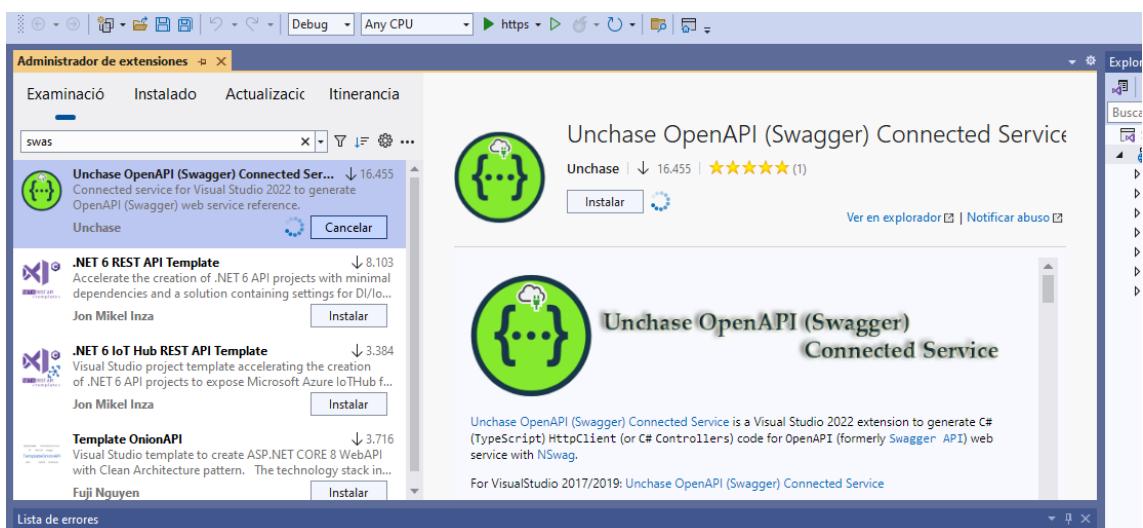
Location

Name of the solution

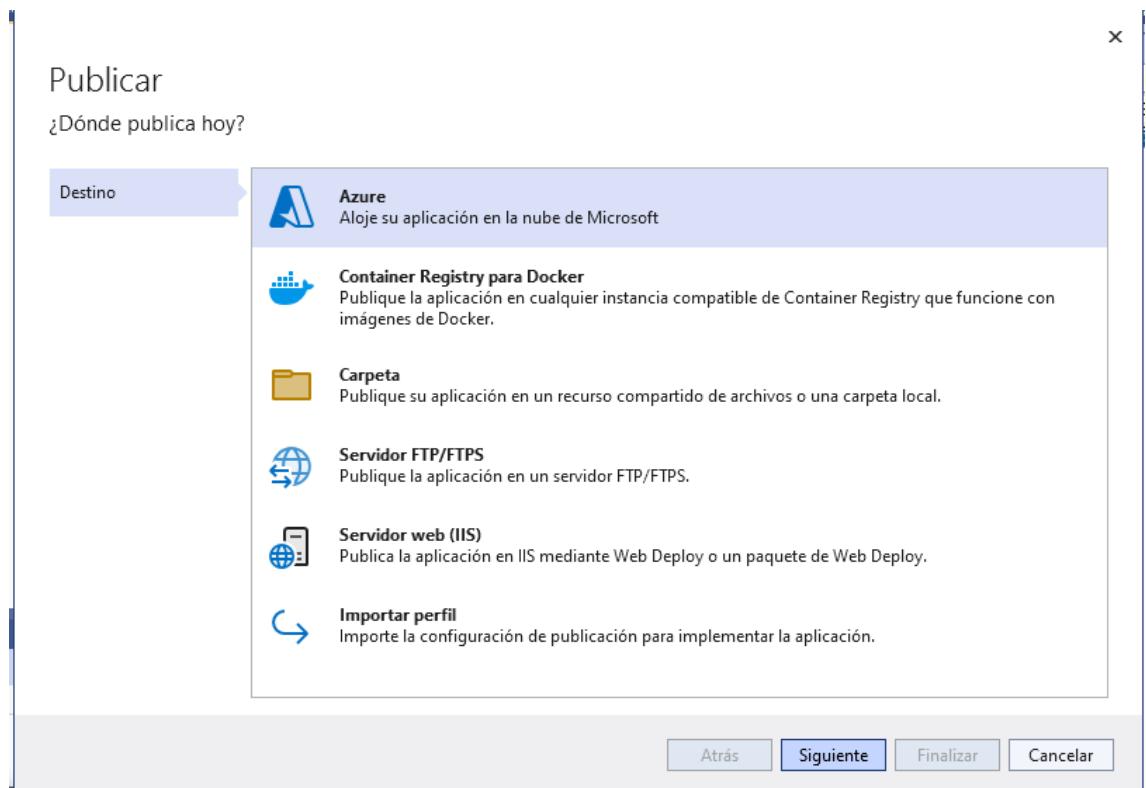
Place the solution and the project in the same directory

Project will be created in "C:\Users\pablo\source\repos\WeatherForecastapi\WeatherForecastapi"

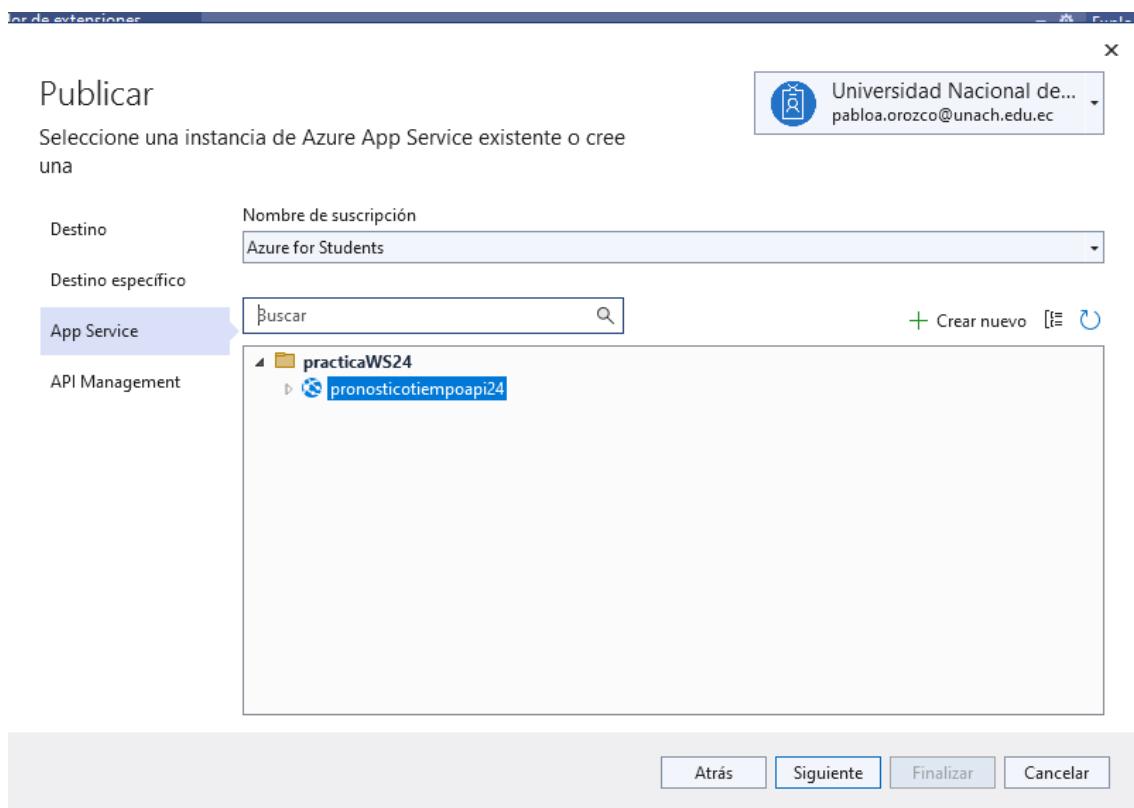
24.



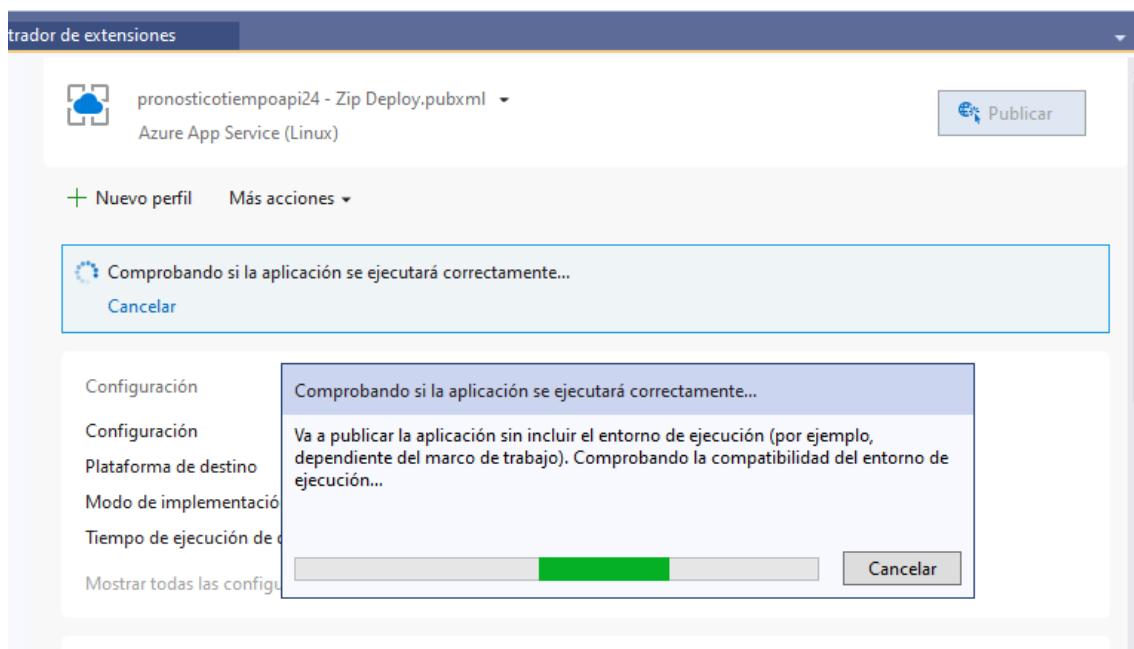
25.



26.



27.



28. S

The screenshot shows the Swagger UI interface for a WeatherForecast API. At the top, it says "WeatherForecast". Below that, under "GET /WeatherForecast", there is a "Parameters" section which says "No parameters". To the right of this is a "Try it out" button. Under "Responses", there is a table with one row. The row has three columns: "Code" (200), "Description" (Success), and "Links" (No links). Below the table, it says "Media type" and shows "text/plain" selected in a dropdown. There is also a note: "Controls Accept header." and "Example Value | Schema".

6. Referencias

[1] “¿Qué es API Management? | IBM”. IBM - United States. Accedido el 12 de julio de 2024. [En línea]. Disponible: <https://www.ibm.com/es-es/topics/api-management#:~:text=API%20Management,%20o%20gestión%20de,un%20entorno%20empresarial%20y%20multinube>.

[2] “Información general y conceptos clave de Azure API Management”. Microsoft Learn: Build skills that open doors in your career. Accedido el 12 de julio de 2024. [En línea]. Disponible: <https://learn.microsoft.com/es-es/azure/api-management/api-management-key-concepts>

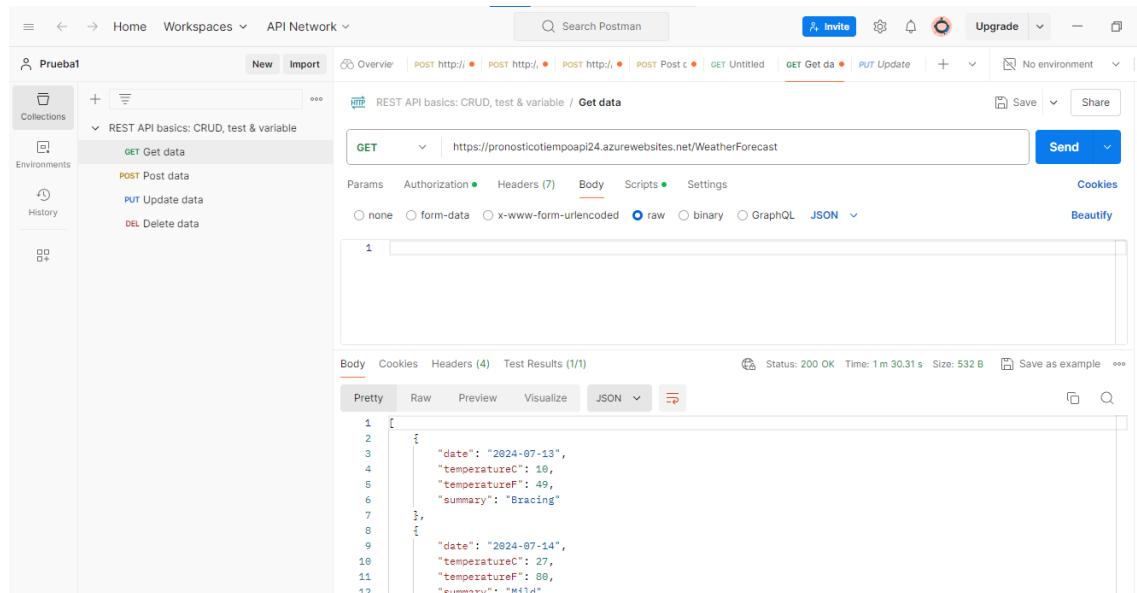
[3] “Azure API Management: Consideraciones y conceptos claves”. Chakray. Accedido el 12 de julio de 2024. [En línea]. Disponible: <https://www.chakray.com/es/azure-api-management-consideraciones-conceptos-claves/>

[4] “¿Qué es API management?” Finerio Connect. Accedido el 12 de julio de 2024. [En línea]. Disponible: <https://blog.finerioconnect.com/que-es-api-management/>

[5] “¿Qué es la gestión de API? | API management benefits”. Software AG. Accedido el 12 de julio de 2024. [En línea]. Disponible: https://www.softwareag.com/es_es/resources/api/guide/api-management.html

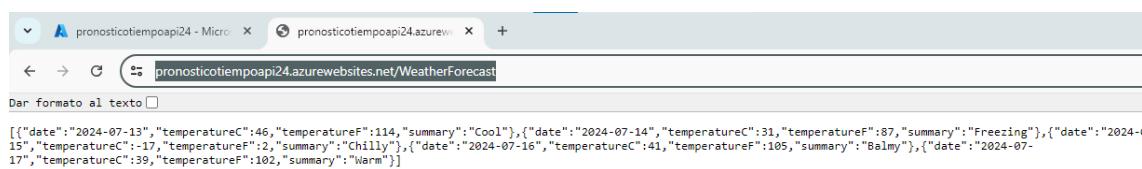
7. Anexos

Funcionamiento Postman



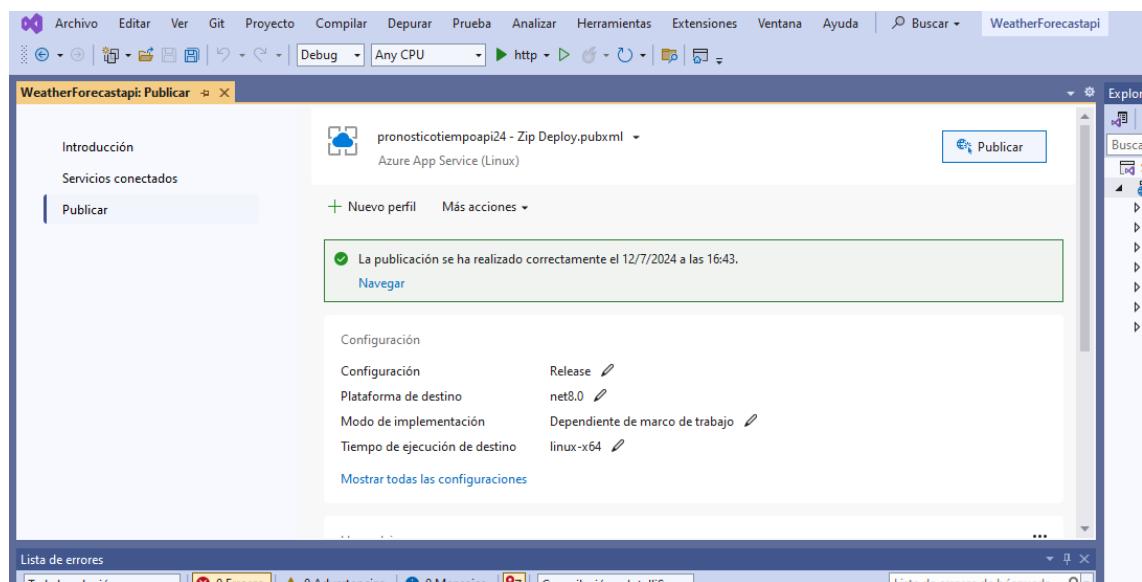
The screenshot shows the Postman application interface. In the left sidebar, there is a collection named "Prueba1" containing a single item: "REST API basics: CRUD, test & variable". This item has four methods listed: "GET Get data", "POST Post data", "PUT Update data", and "DEL Delete data". The "GET Get data" method is selected. The "Body" tab is active, showing the JSON response from the API call. The response body is:

```
[{"date": "2024-07-13", "temperatureC": 10, "temperatureF": 49, "summary": "Bracing"}, {"date": "2024-07-14", "temperatureC": 27, "temperatureF": 80, "summary": "Mild"}]
```



The screenshot shows a Microsoft Edge browser window with the URL "pronosticotiempoapi24.azurewebsites.net/WeatherForecast" in the address bar. The page content displays the JSON response from the API call, which is identical to the one shown in the Postman screenshot.

```
[{"date": "2024-07-13", "temperatureC": 10, "temperatureF": 49, "summary": "Bracing"}, {"date": "2024-07-14", "temperatureC": 27, "temperatureF": 80, "summary": "Mild"}]
```



The screenshot shows the Visual Studio IDE interface with the project "WeatherForecastapi" open. The "Publish" tab is selected in the left navigation pane. A deployment log message is displayed: "La publicación se ha realizado correctamente el 12/7/2024 a las 16:43." Below this, deployment configurations are listed:

Configuración	Release
Plataforma de destino	net8.0
Modo de implementación	Dependiente de marco de trabajo
Tiempo de ejecución de destino	linux-x64