



UNIVERSIDAD ISRAEL

CIENCIAS DE LA INGENIERÍA

CARRERA DE SISTEMAS DE INFORMACIÓN

PLATAFORMAS DE DESARROLLO 2

SEMESTRE 2020 B

INFORME DE LABORATORIO S3

TEMA: SOAP

ESTUDIANTE(S): MARCO ANTONIO AYALA LITUMA

CURSO: Octavo

PARALELO: "A"

PROFESOR: Mg. Luis Fernando Aguas Bucheli

QUITO, 2021

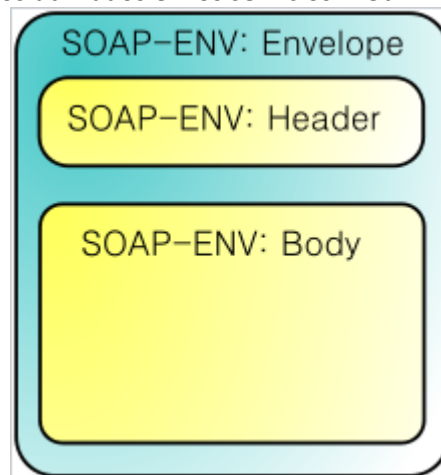
1. TEMA: SOAP

2. OBJETIVOS:

- Describir que son los servicios web.
- Adquirir los conceptos básicos relacionados con SOAP
- Reconocer las características de SOAP
- Conocer como se publica un servicio SOAP

3. INTRODUCCION:

SOAP (originalmente las siglas de Simple Object Access Protocol) es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML. Este protocolo deriva de un protocolo creado por Dave Winer en 1998, llamado XML-RPC. SOAP fue creado por Microsoft, IBM y otros. Está actualmente bajo el auspicio de la W3C. Es uno de los protocolos utilizados en los servicios Web.



SOAP es un paradigma de mensajería de una dirección sin estado, que puede ser utilizado para formar protocolos más completos y complejos según las necesidades de las aplicaciones que lo implementan. Puede formar y construir la capa base de una "pila de protocolos de web service", ofreciendo un framework de mensajería básica en el cual los web services se pueden construir. Este protocolo está basado en XML y se conforma de tres partes:

Sobre (envelope): el cual define qué hay en el mensaje y cómo procesarlo.
Conjunto de reglas de codificación para expresar instancias de tipos de datos.
La Convención para representar llamadas a procedimientos y respuestas.
El protocolo SOAP tiene tres características principales:

Extensibilidad (seguridad y WS-routing son extensiones aplicadas en el desarrollo).

Neutralidad (bajo protocolo de transporte TCP puede ser utilizado sobre cualquier protocolo de aplicación como HTTP, SMTP o JMS).

Independencia (permite cualquier modelo de programación).

Como ejemplo de cómo el modelo SOAP pueda ser utilizado, consideraremos un mensaje SOAP que podría ser enviado a un web service para realizar la búsqueda de algún precio en una base de datos, indicando para ello los parámetros necesitados en la consulta. El servicio podría

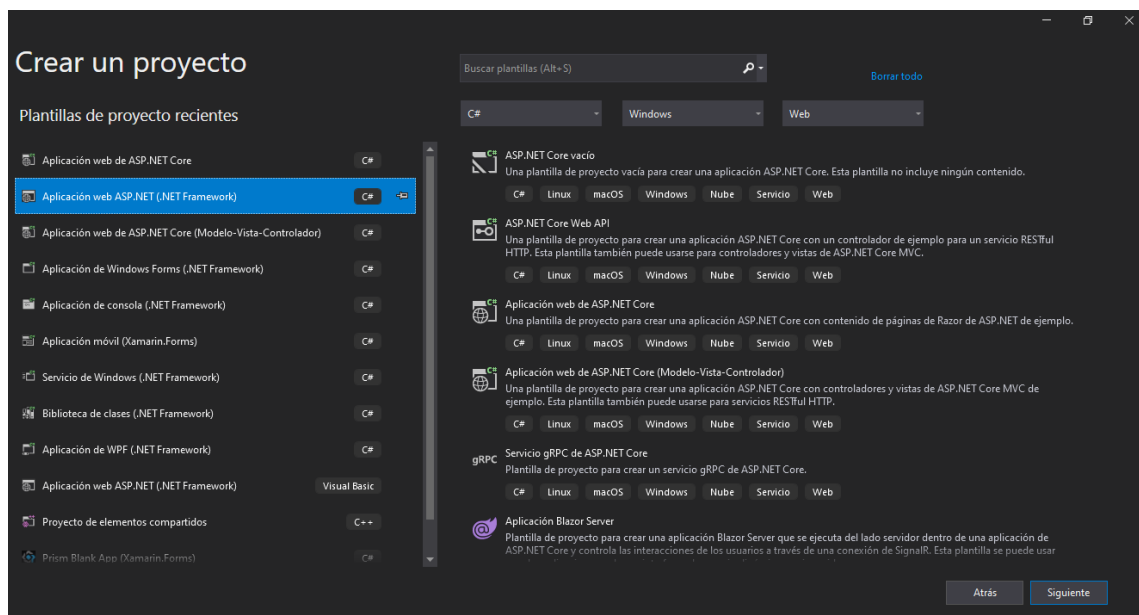


retornar un documento en formato XML con el resultado, un ejemplo, precios, localización o características. Teniendo los datos de respuesta en un formato estandarizado procesable (en inglés "parsable"), éste puede ser integrado directamente en un sitio Web o aplicación externa.

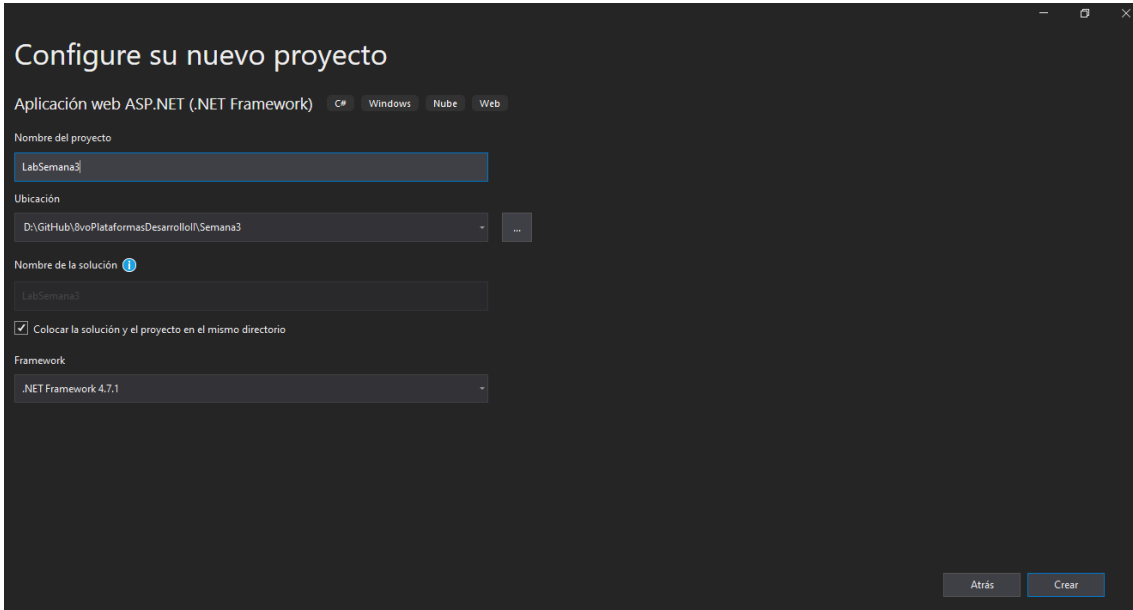
La arquitectura SOAP está formada por varias capas de especificación: MEP (Message Exchange Patterns) para el formato del mensaje, enlaces subyacentes del protocolo de transporte, el modelo de procesamiento de mensajes, y la capa de extensibilidad del protocolo. SOAP es el sucesor de XML-RPC, a pesar de que toma el transporte y la neutralidad de la interacción, así como el envelope / header / body, de otros modelos (probablemente de WDDX).

4. DESARROLLO:

Abrimos el visual studio y seleccionamos la plantilla del proyecto Aplicattion ASP.NET de tipo NetFramework



Ahora le damos un nombre al proyecto y la ubicación, también seleccionamos con framework.



Configure su nuevo proyecto

Aplicación web ASP.NET (.NET Framework) C# Windows Nube Web

Nombre del proyecto
LabSemana3

Ubicación
D:\GitHub\8voPlataformasDesarrollo\Semana3

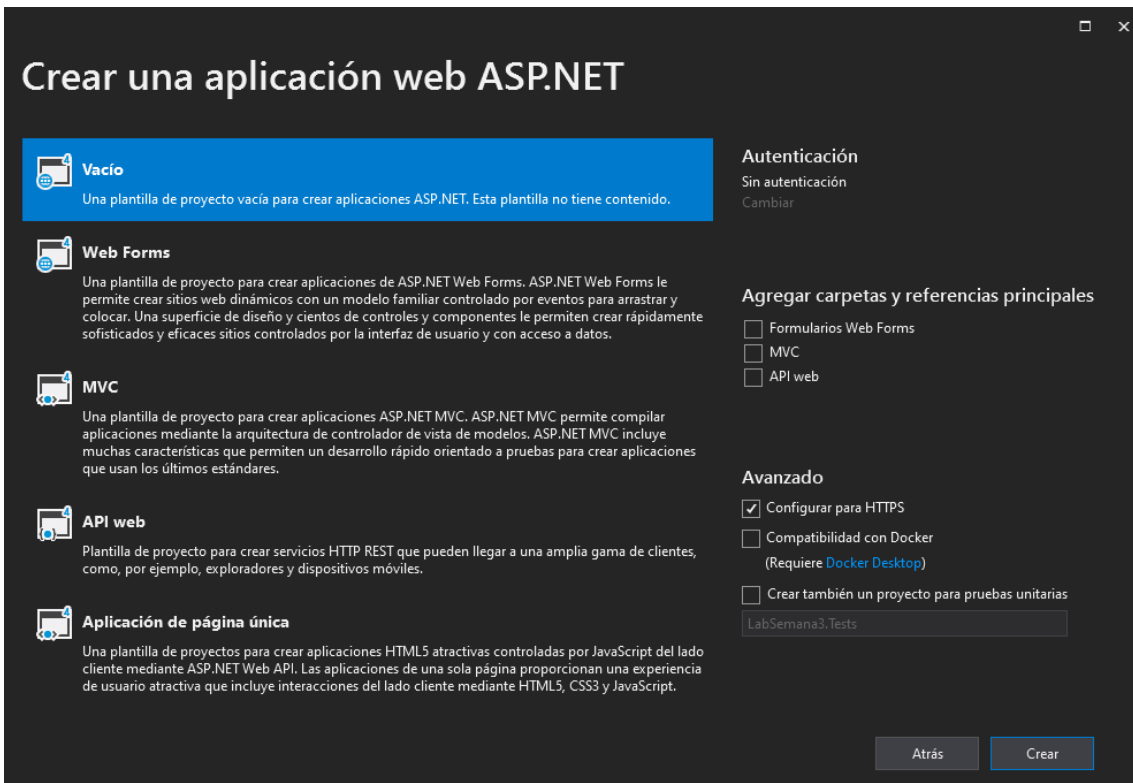
Nombre de la solución
LabSemana3

☒ Colocar la solución y el proyecto en el mismo directorio

Framework
.NET Framework 4.7.1

Atrás Crear

En esta pantalla seleccionamos el tipo de plantilla.



Crear una aplicación web ASP.NET

Vacío
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
Sin autenticación
Cambiar

Agregar carpetas y referencias principales

- ☐ Formularios Web Forms
- ☒ MVC
- ☒ API web

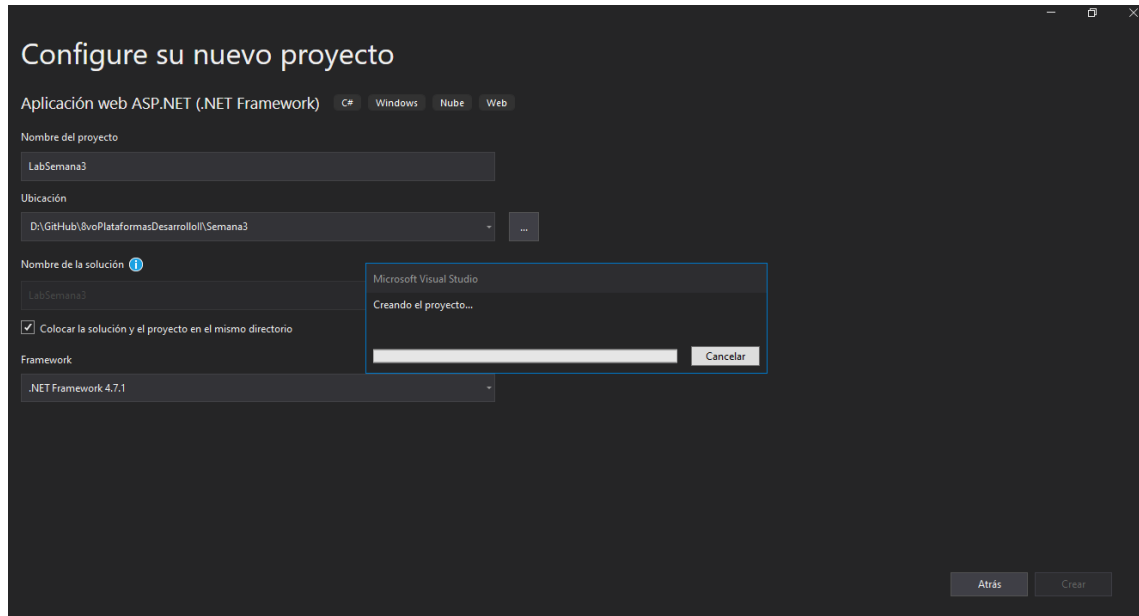
Avanzado

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

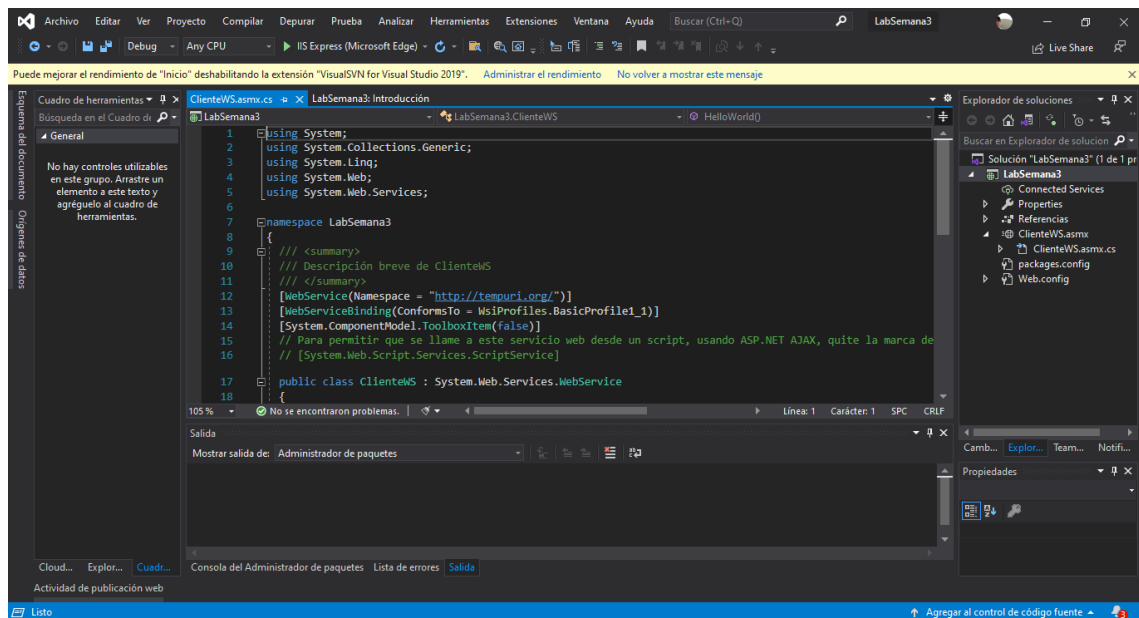
LabSemana3.Tests

Atrás Crear

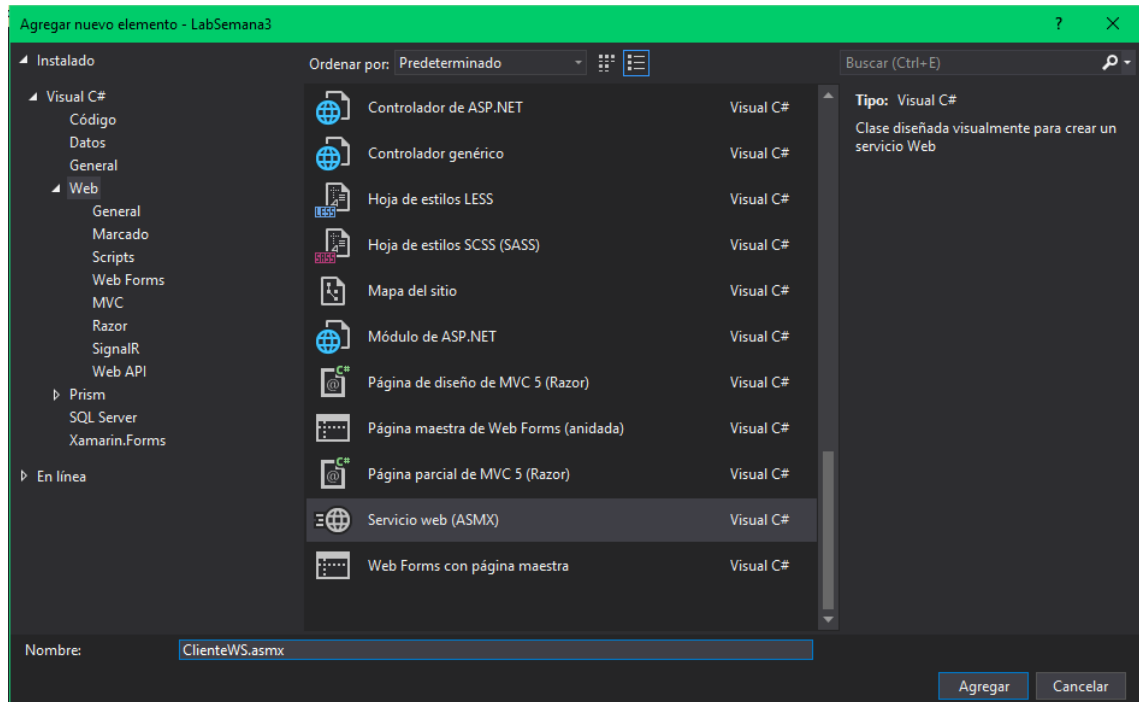
En esta pantalla vemos que se va cargando la plantilla con las características que hemos seleccionado.



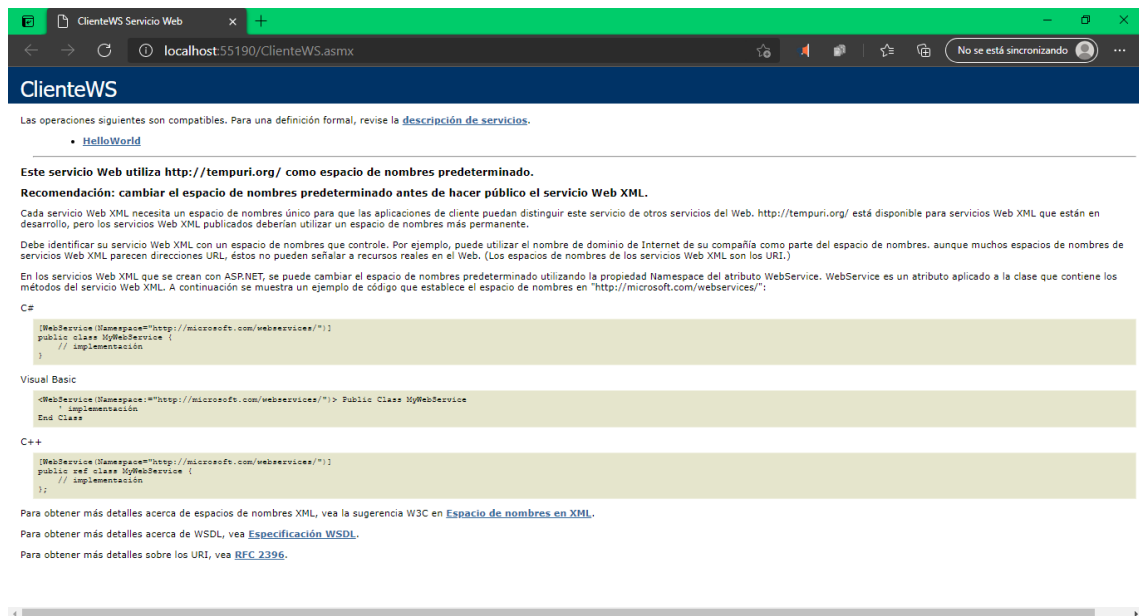
Ahora nuestro proyecto tendrá estos archivos.



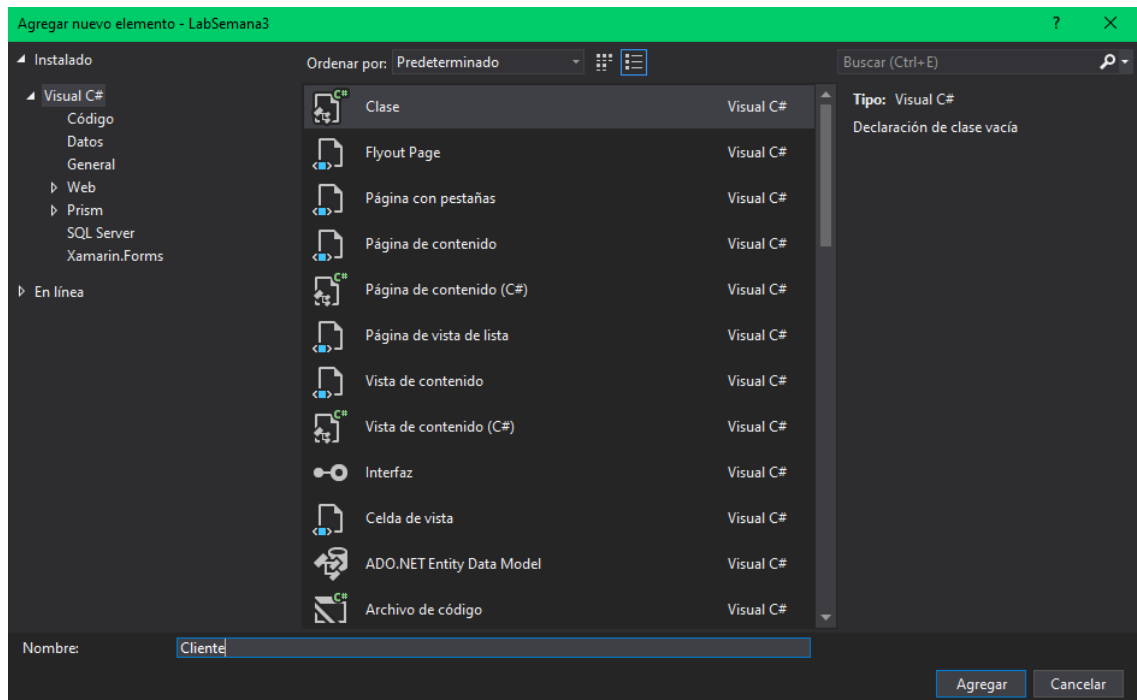
Ahora en el proyecto damos clic derecho y seleccionamos agregar, el elemento que utilizaremos es un servicio web de tipo ASMX .



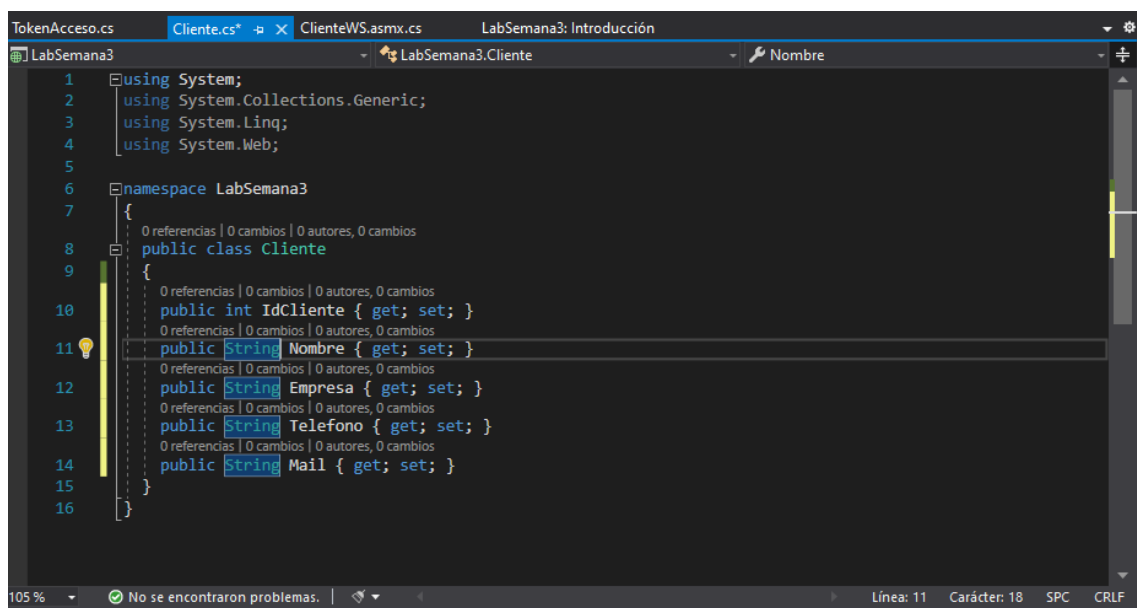
Para hacer una prueba damos compile el proyecto y se visualizará de esta manera.



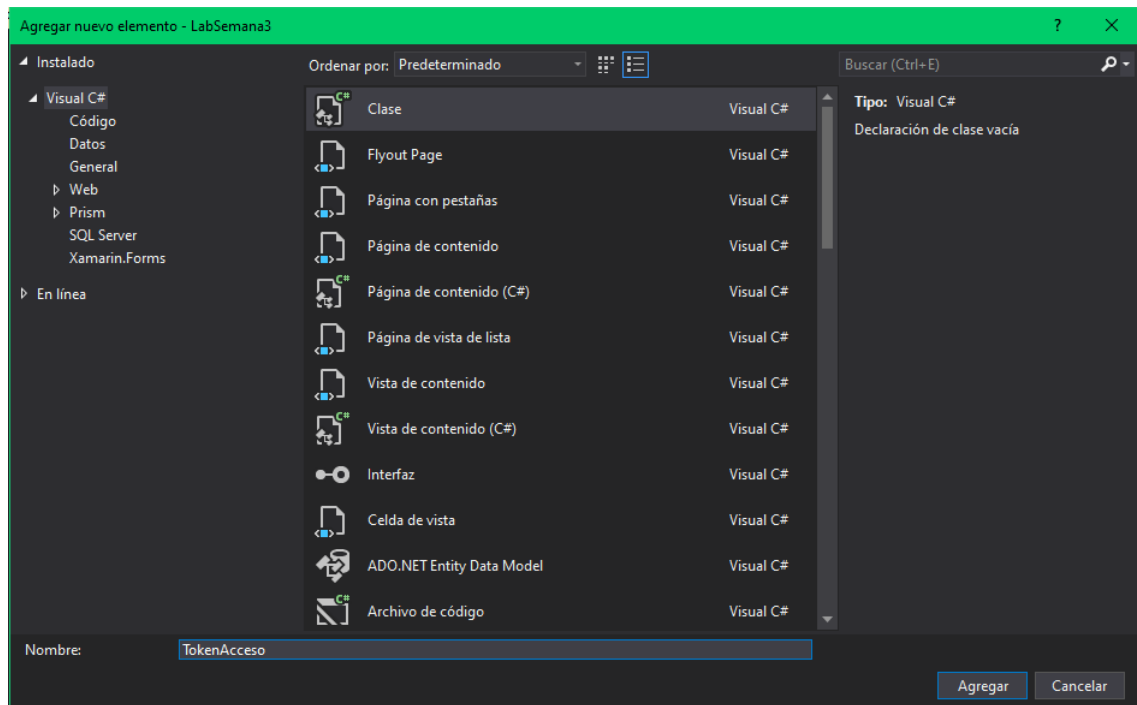
Ahora agregaremos esta clase Cliente



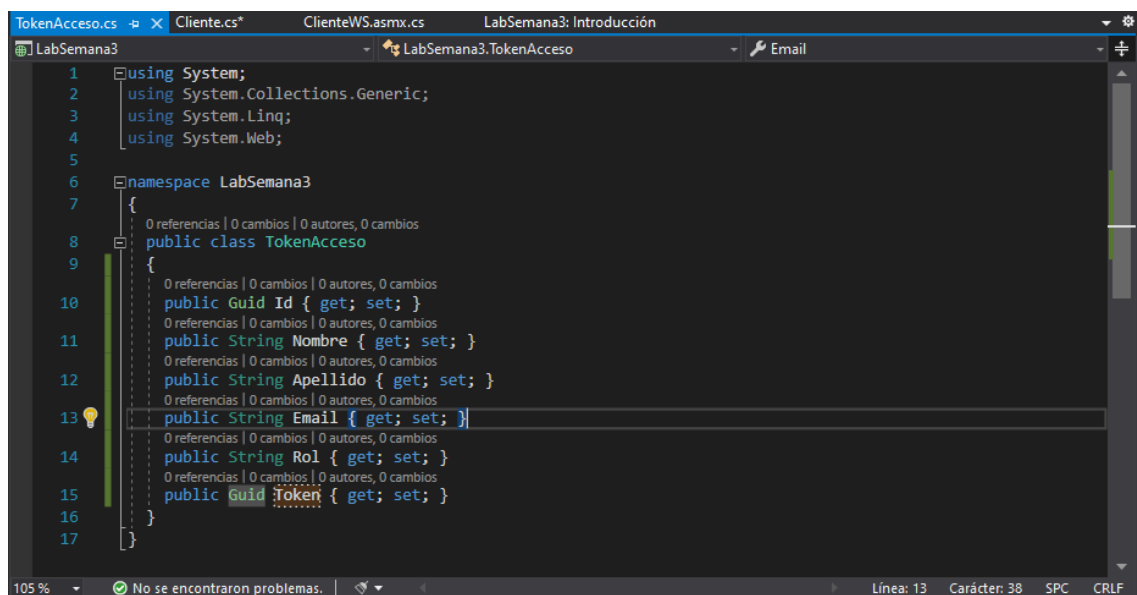
En la clase colocaremos estas variables



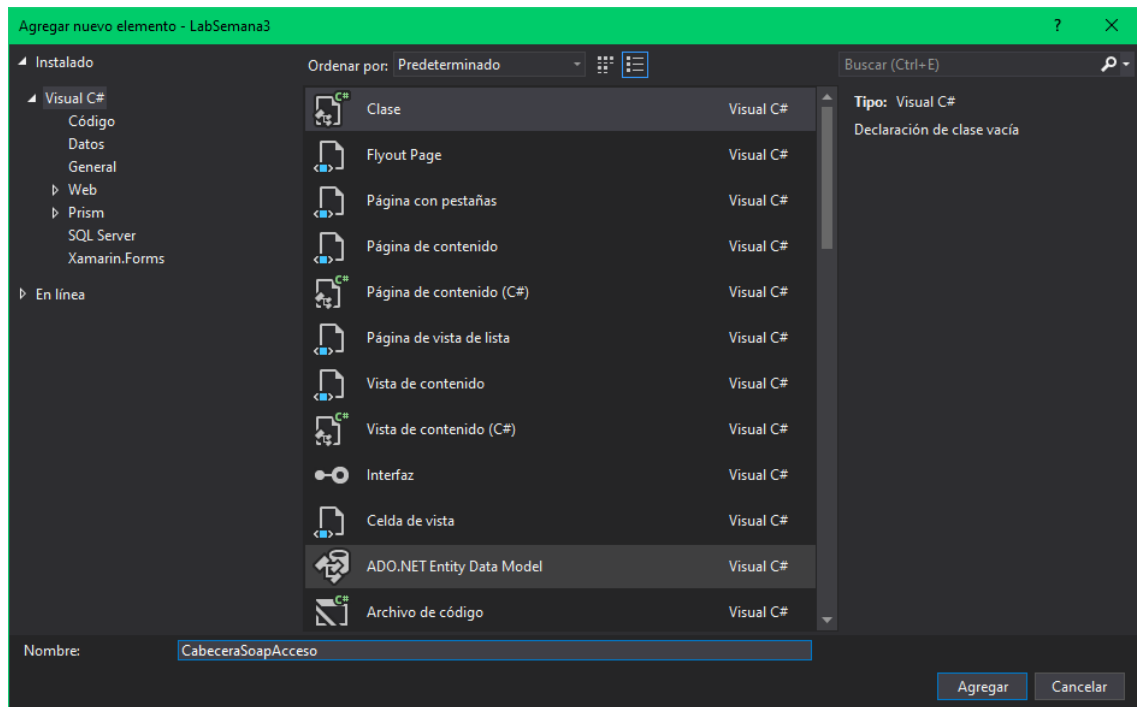
Ahora al mismo proyecto creamos esta clase TokenAcceso



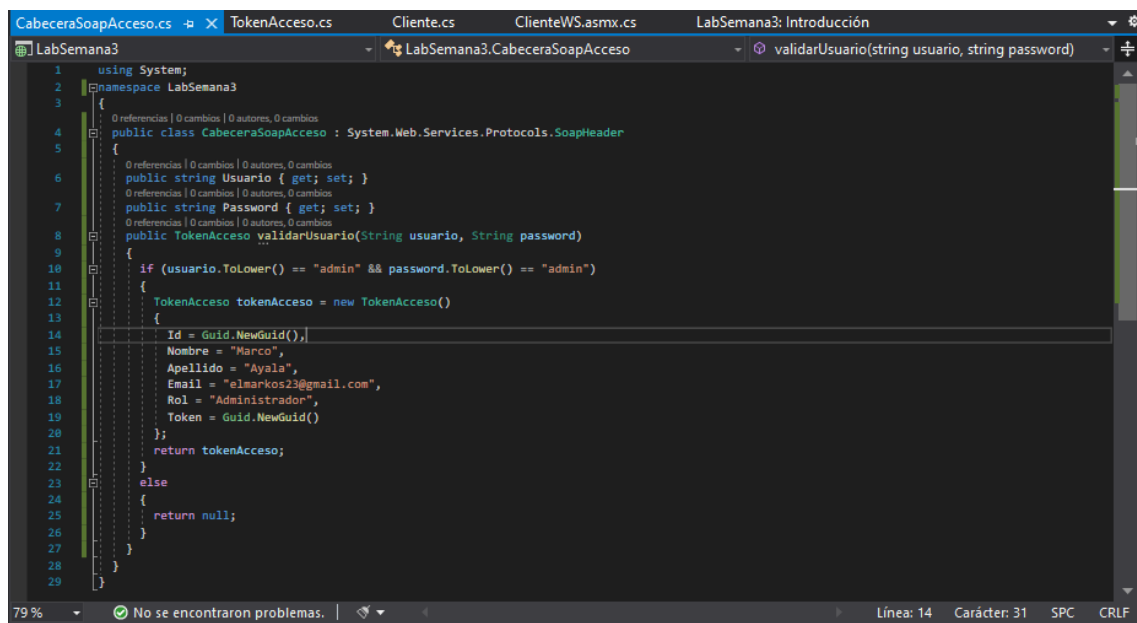
Ahora agregamos estas variables en la clase creada.



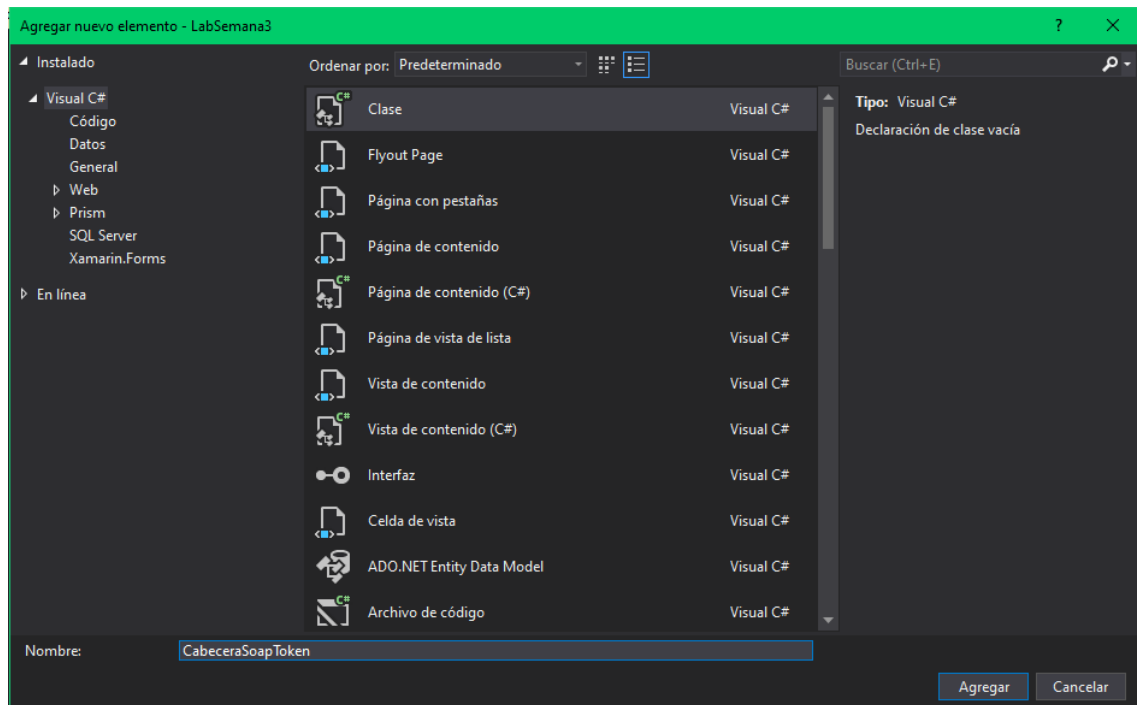
Ahora agregaremos esta clase CabeceraSoapAcceso



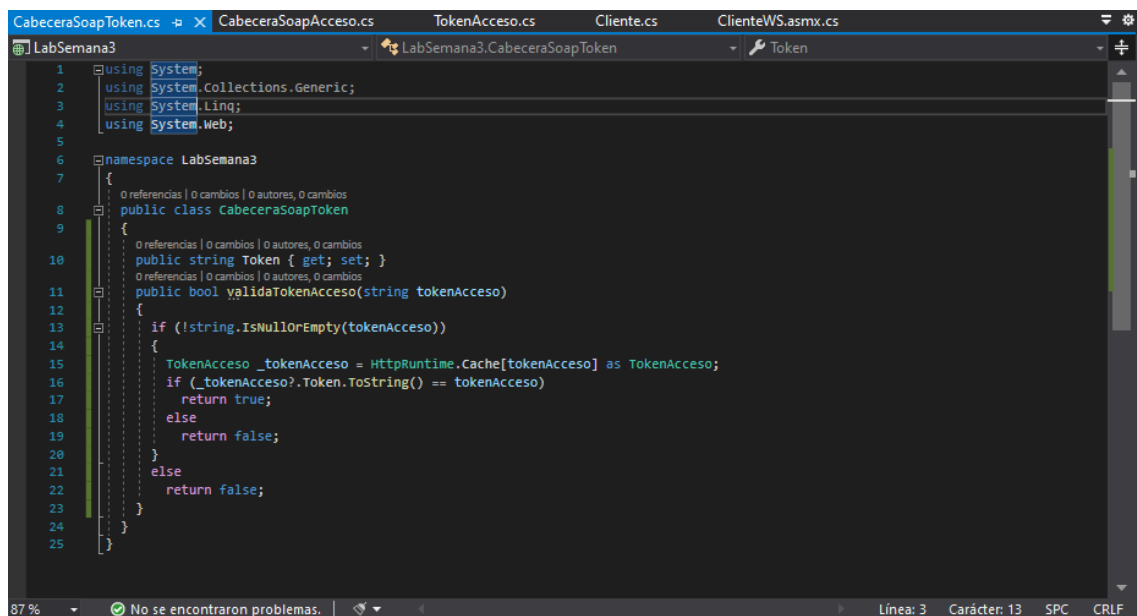
Agregamos el siguiente código en la clase creada.



Ahora crearemos la siguiente clase CabeceraSoapToken



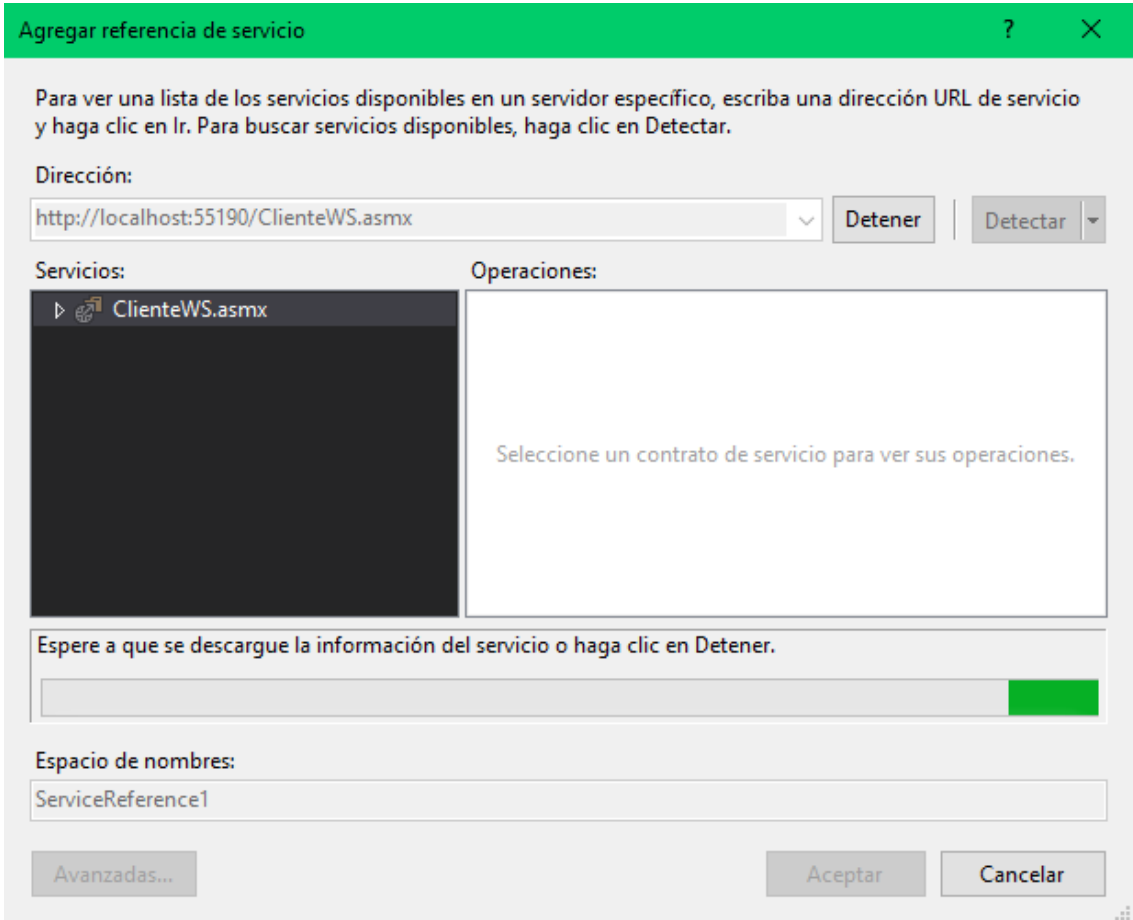
Agregamos el siguiente código fuente



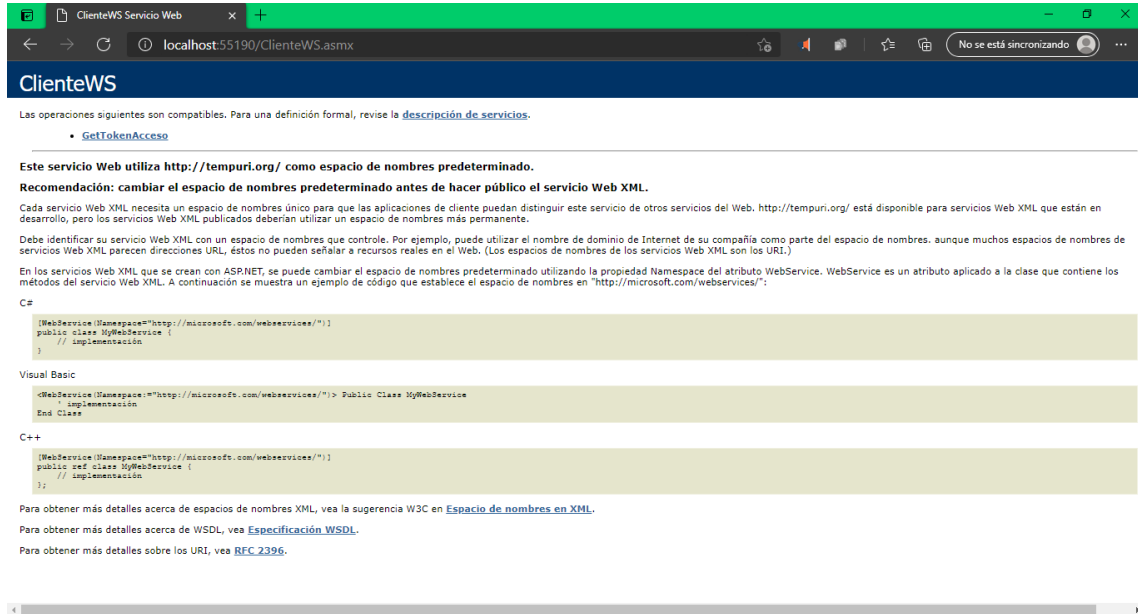
Ahora en nuestra clase del servicio web agregamos el siguiente código

```
[WebMethod]
[SoapHeader("cabeceraSoapAcceso")]
0 referencias | 0 cambios | 0 autores, 0 cambios
public TokenAcceso GetTokenAcceso()
{
    string usuario;
    string password;
    string token = string.Empty;
    TokenAcceso tokenAcceso;
    if (cabeceraSoapAcceso == null || string.IsNullOrEmpty(cabeceraSoapAcceso?.Usuario) || string.IsNullOrEmpty(cabeceraSoapAcceso?.Password))
    {
        throw new SoapException("Acceso no Autorizado", SoapException.ClientFaultCode, new Exception(@"ERROR"));
    }
    else
    {
        usuario = cabeceraSoapAcceso.Usuario;
        password = cabeceraSoapAcceso.Password;
    }
    tokenAcceso = cabeceraSoapAcceso.validarUsuario(usuario, password);
    if (token != null)
    {
        tokenAcceso.Token = Guid.NewGuid();
        HttpRuntime.Cache.Add(tokenAcceso.Token.ToString(), tokenAcceso, null, System.Web.Caching.Cache.NoAbsoluteExpiration, TimeSpan.FromMinutes(2), System.Web.Caching.CacheItemPriority.Default);
    }
    else
    {
        throw new SoapException("Error", SoapException.ClientFaultCode, new Exception(@"ERROR"));
    }
    if (!string.IsNullOrEmpty(tokenAcceso.Token.ToString()))
    {
        return tokenAcceso;
    }
    else
    {
        return null;
    }
}
```

Ahora hacemos clic derecho en el proyecto y seleccionamos agregar servicio referencia y hacemos clic en detectar.



Ahora nuestro servicio web ya compilado se visualizara con el siguiente método de llamado al acceso token.



5. CONCLUSIONES:

- Los servicios web son de gran ayuda para interactuar con plataformas externas
- La programación orientada a objetos nos facilita la codificación.
- Se puede trabajar con clases nativas de seguridad
- El servicio web trabaja bajo demanda.

6. RECOMENDACIONES:

- No sobre cargar servicios web con demanda información de retorna ya que esto puede ocasionar una perdida de paquetes cuando se utilice en ambientes con latencia de conectividad.
- Aprender a Utilizar también servicios REST ya que facilitan el manejo y mapeo de datos.
- Acostumbrarse a utilizar seguridades como token o autenticación para usuarios externos.
- Manejar con cautela la información de la cache del servidor web y navegación.

7. BIBLIOGRAFIA:

Soap Simple Object Access Protocol [Simple Object Access Protocol - Wikipedia, la enciclopedia libre](#)

Raya González, L. y Toharia Rabasco, J. P. (2015). Lenguajes de marcas y sistemas de gestión de la información. RA-MA Editorial. <https://elibro.net/es/ereader/uisrael/62483?page=1>

