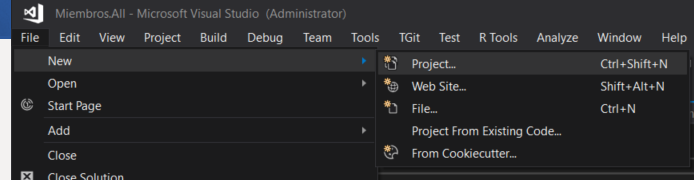
# **TUTORIAL REST**

# INTRODUCCIÓN

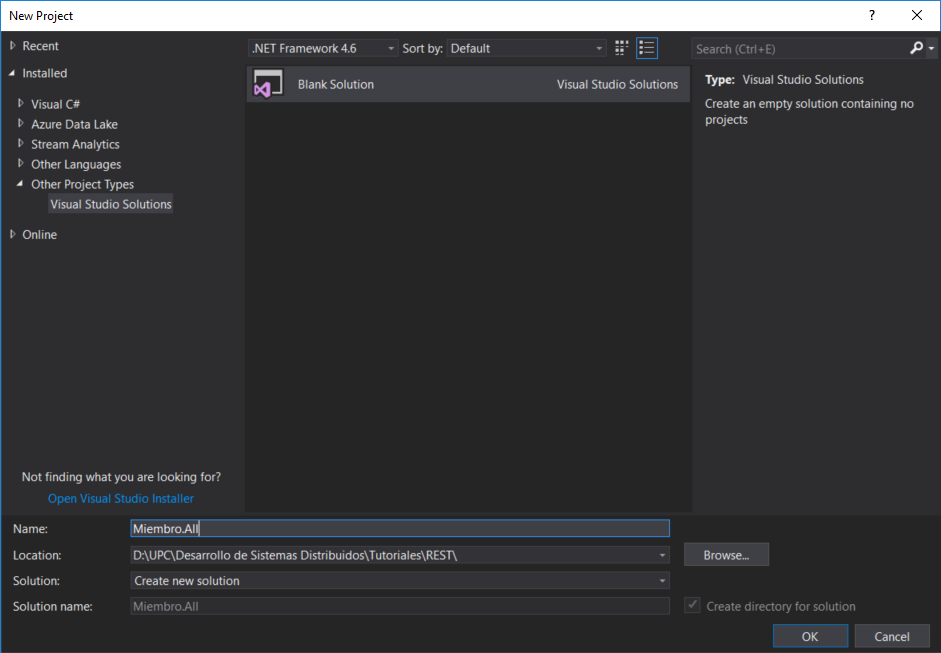
A continuación, se demostrará la creación, implementación, consumo y pruebas de integración de un servicio REST.

# PASOS

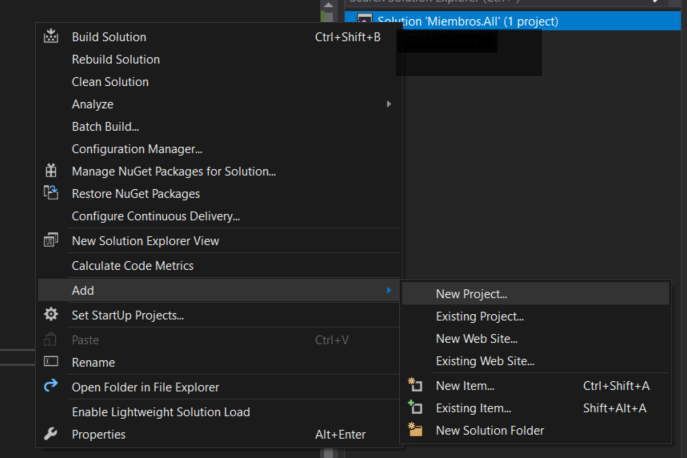
1. Creación de la solución. Esta solución contrendrá todos los proyectos de esta demo tales como WCFMiembros y WCFMiembrosTests.
   1. Ir a File, New y Project.



* 1. Se mostrará una ventada en donde se deberá seleccionar el componente a crear. En este caso ir a sección izquierda, en Installed, opción Visual Studio Solution y seleccionarlo.
  2. En el salo derecho seleccionar Blank Solution.
  3. Para finalizar, se le pone un nombre y una ruta.

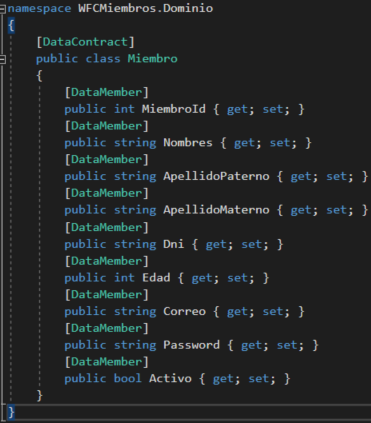


1. Creación del proyecto WCF. Por fines de simplicidad, este proyecto contendrá los objetos de negocio, objetos de acceso a datos, los contratos y su implementación.
   1. Ir a la solución creada y agregar un servicio WCF al cual llamaremos WCFMiembros
   2. Agregar el componente WCF.

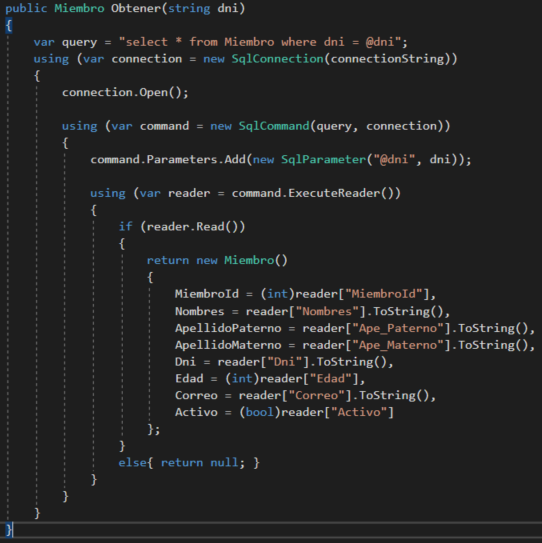


* 1. Eliminar el servicio creado por defecto.

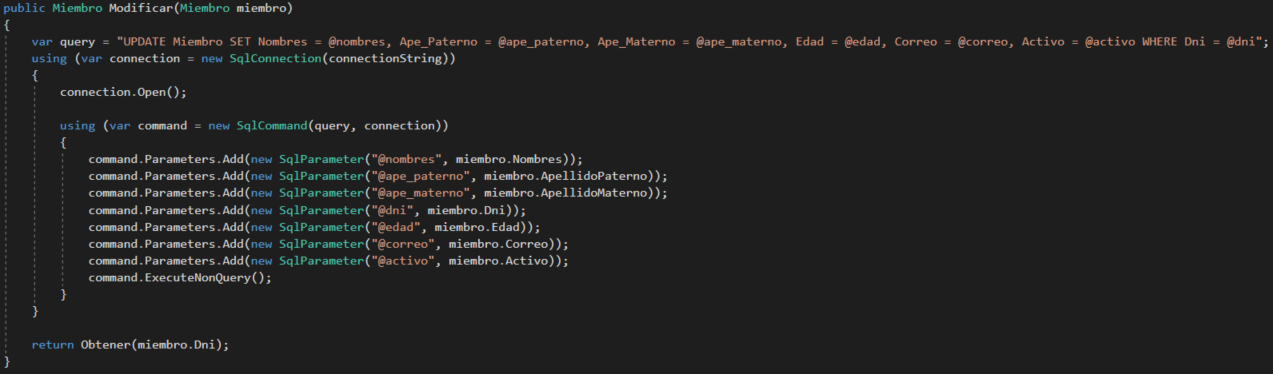
1. Creación del objeto Miembro
   1. Crear la carpeta Dominio dentro del proyecto WCF y en ella crear una clase llamada Miembro.
   2. Agregar los atributos de [DataContract] y [DataMember].



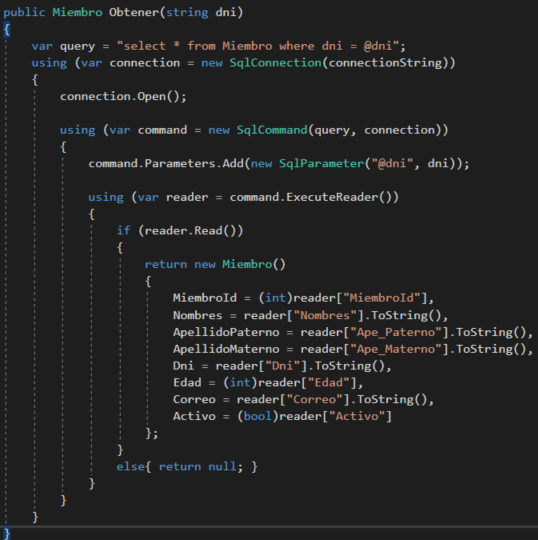
1. Creación del objeto MiembroDAO. Esta clase será la responsable de la conexión a datos y tendrá los métodos de Crear, Modificar, Obtener, ObtenerPorDniYContrasena, Eliminar y Listar.
   1. Crear miembro.



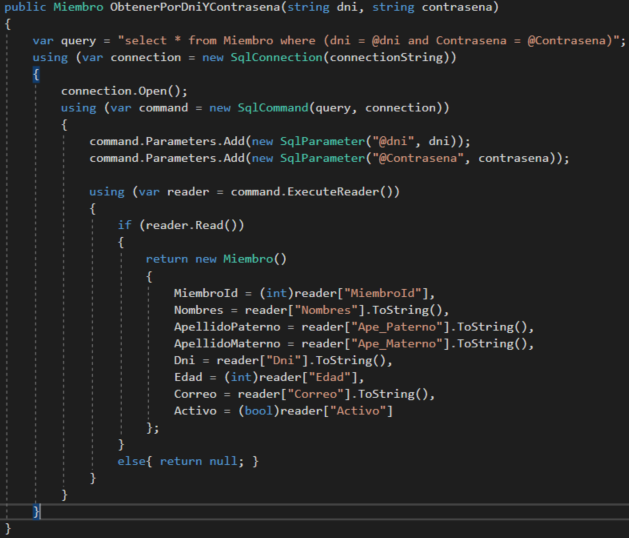
* 1. Modificar miembro.



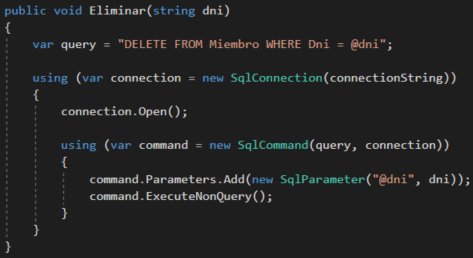
* 1. Obtener miembro.



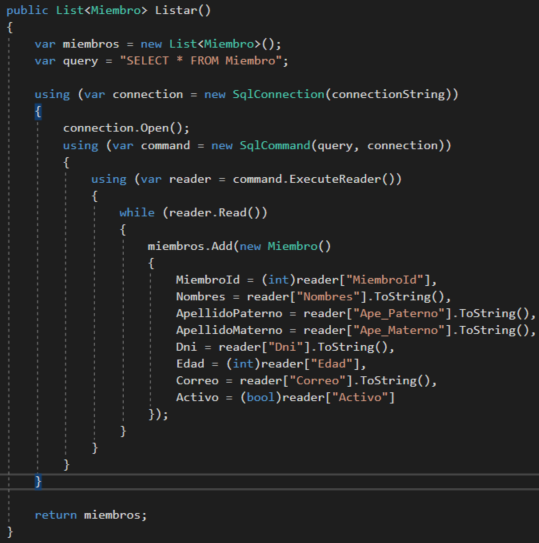
* 1. ObtenerPorDniYContrasena.



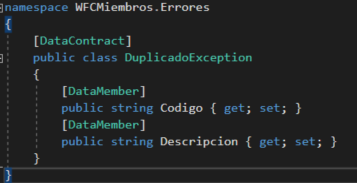
* 1. Eliminar miembro.



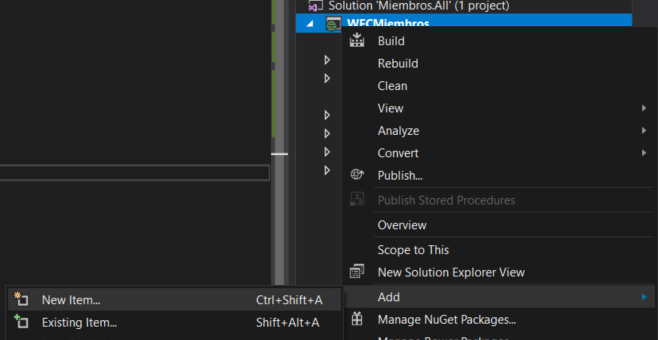
* 1. Listar miembros.

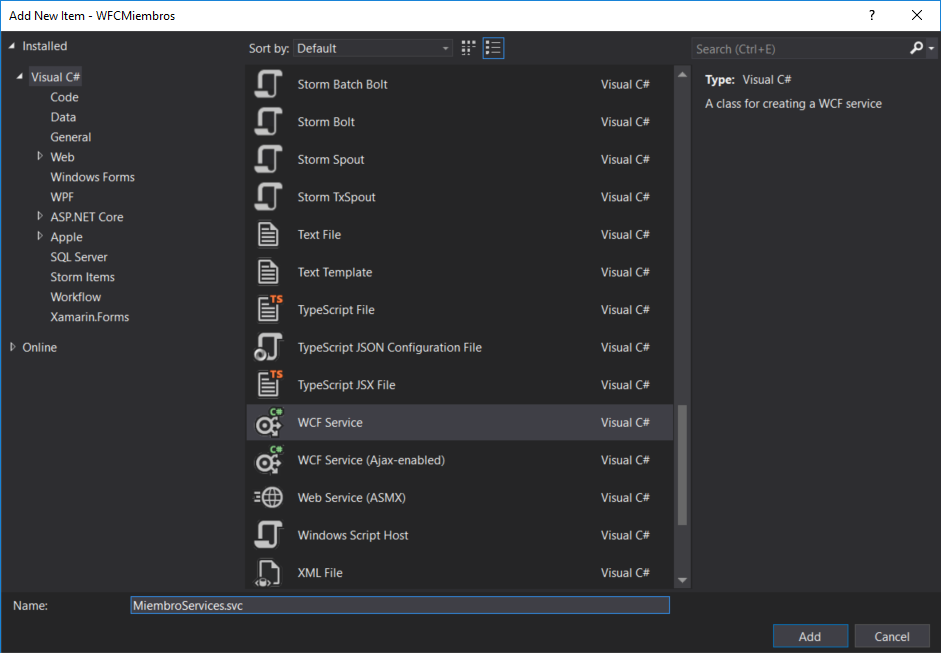


1. Creación de la clase de errores. Esta clase estará dentro de la carpeta errores y se llamará DuplicadoException. Puede haber tantas excepciones como desees, todo depende de la logica y errores que puedan hacer en tu flujo d negocio.
   1. Crear la clase llamada DuplicadoException.

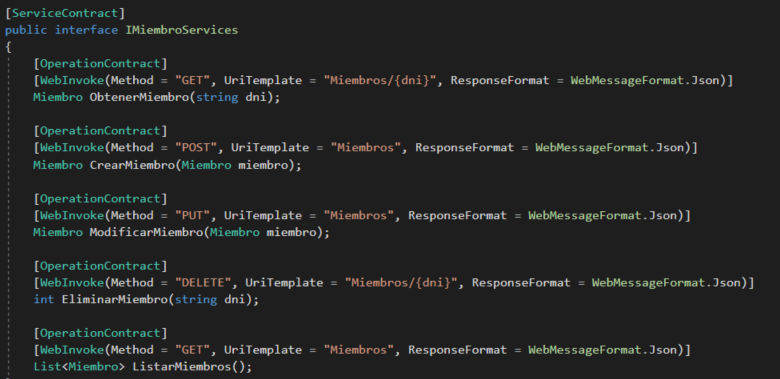


1. Creación del servicio MiembroServices. Este servicio será el que contendrá las operaciones necesarias del api. Además, también tendrán detalles de los Verbos HTTP, Url del recurso y el tipo de response de la misma.
   1. Ir al proyecto WCFMiembro y agregar el servicio llamado MiembroServices.

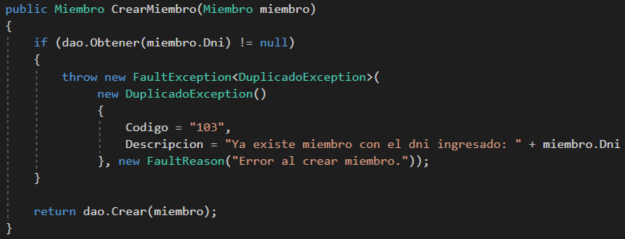




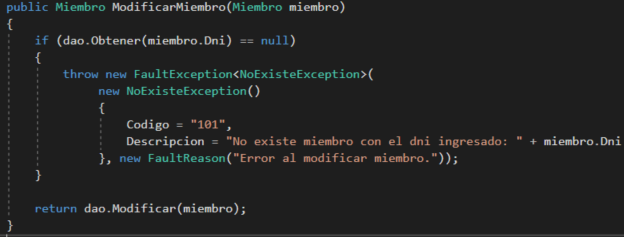
1. En IMiembroService agregar las operaciones necesarias que tendrá el servicio. Pueden ser las mismas que están en la clase de acceso a datos, todo depende de lo que se desea exponer al exterior. Hay que recordar también que tendrá WebInvoke para la configuración de acceso al recurso, url y tipo de respuesta, en este caso, JSon.
   1. Crear miembro trabajará con el verbo POST.
   2. Modificar miembro trabajará con el verbo PUT.
   3. Eliminar miembro trabajará con el verbo DELETE.
   4. Obtener miembro trabajará con el verbo GET.
   5. Listas miembros también trabajará con el verbo GET.



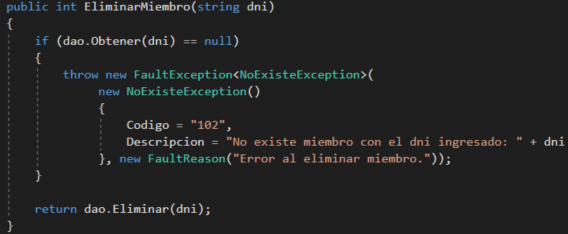
1. Implementación de IMiembroService. Acá de hará la implementación de todas las operaciones de la interface.
   1. Crear miembro contiene una validación relacionado el DNI. Si se intenta crear un miembro con un DNI existente, lanza una excepción de duplicidad.



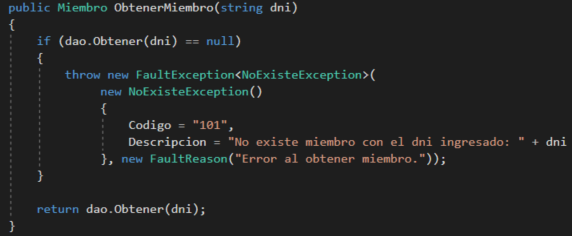
* 1. Modificar miembro contiene una validación relacionado el DNI. Si se intenta modificar un miembro con un DNI no existente, lanza una excepción de no existencia.



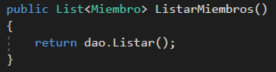
* 1. Eliminar miembro, al igual que modificar, si se intenta eliminar un miembro con un DNI que no existe, lanza una excepción de no existencia.



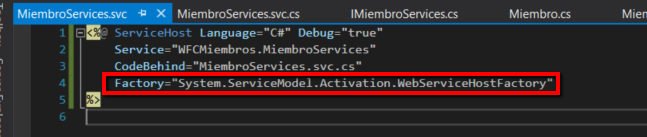
* 1. Obtener miembro, al igual que modificar y eliminar, si se intenta obtener un miembro con un DNI que no existe, lanza una excepción de no existencia.



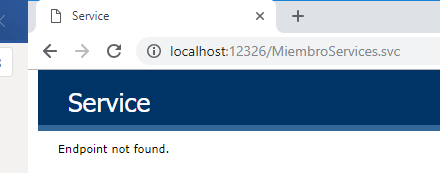
* 1. Listar miembros, no tiene una validación. Devuelve los miembros registrado.



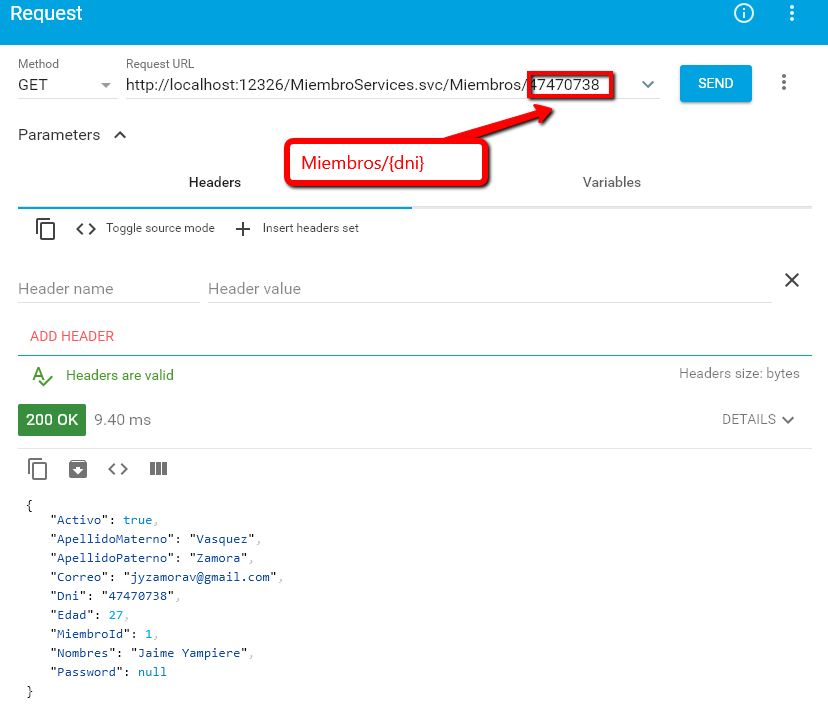
1. Configuración para visualizar el servicio en browser.
   1. Ir a miembros service.svc, click derecho y View MarkUp.
   2. Agregar Factory sección.



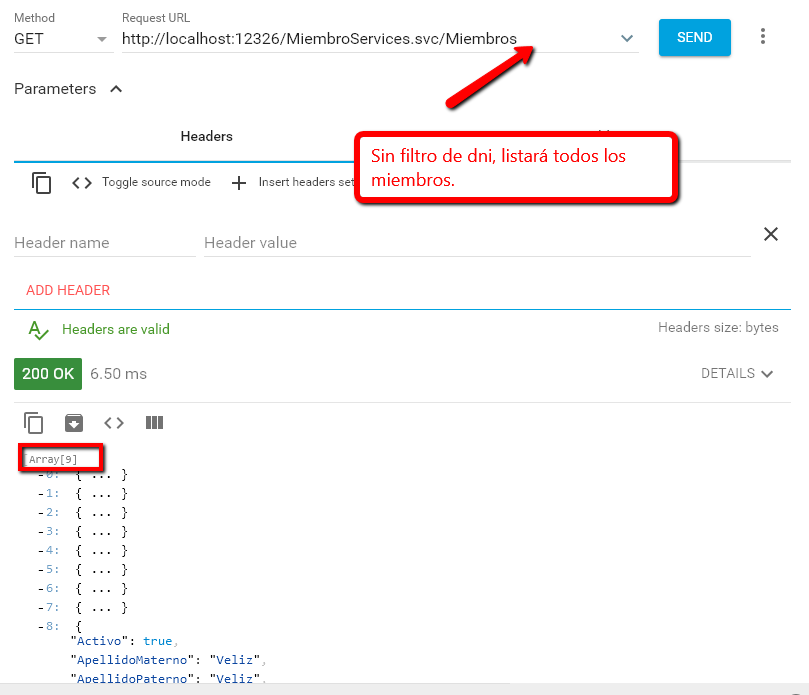
* 1. Abrir el servicio en el navegador



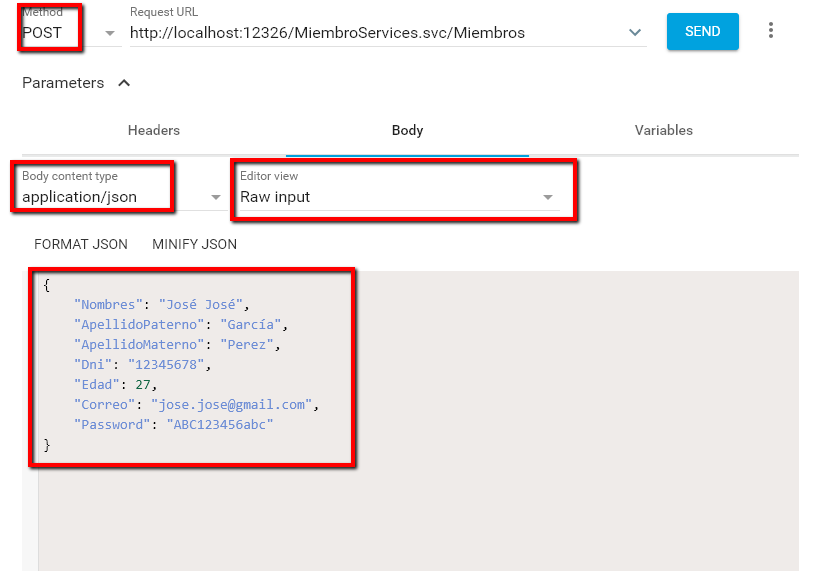
1. Agregar la extensión de Chrome Advance REST Client.
2. Realizaremos la prueba de los servicios con Rest client.
   1. Obtener miembro.

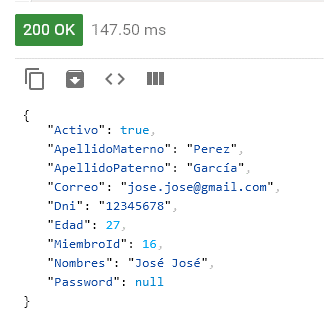


* 1. Listar miembro.

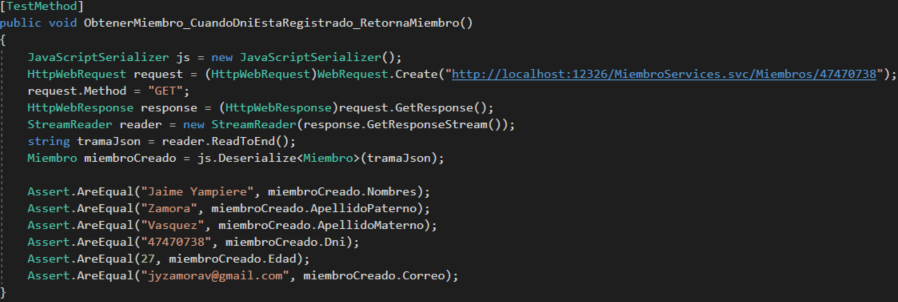


* 1. Crear miembro

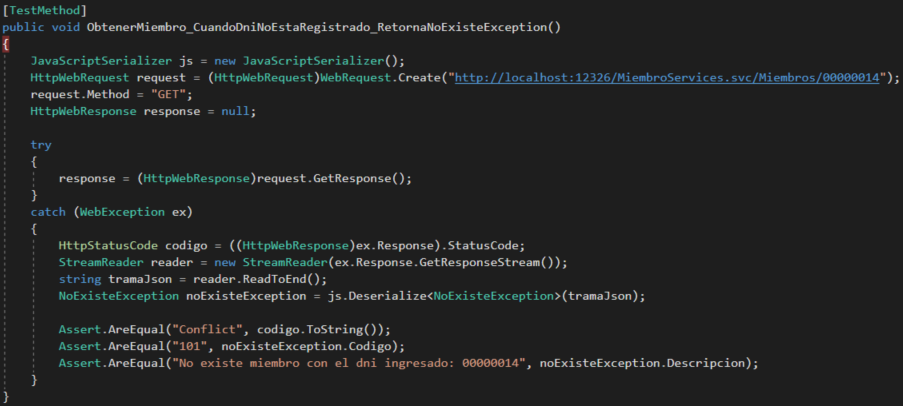




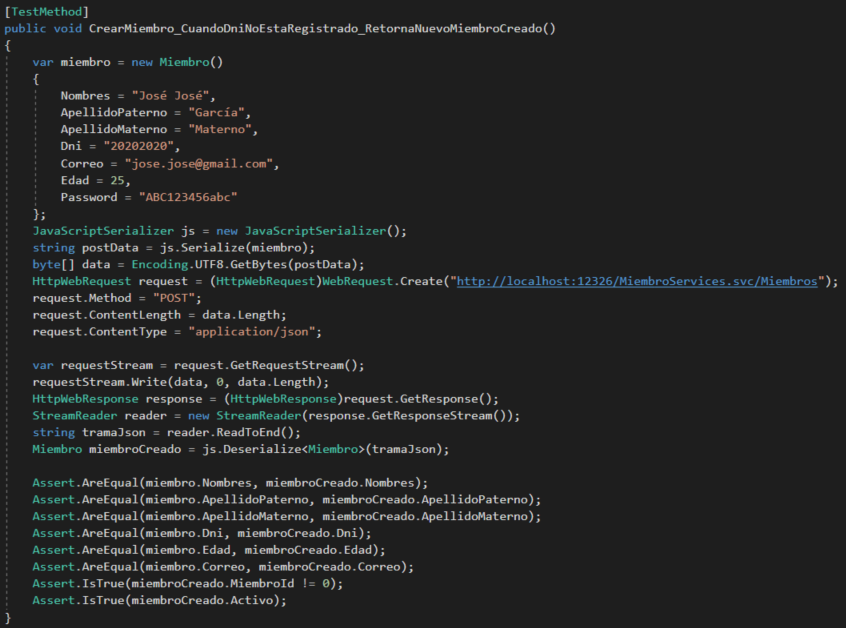
1. Pruebas de integración. A solicitud del docente se mostrará a implementación de un método que obtiene miembro. Dicho método tiene como parámetro el DNI y en caso no exista en la base de datos pues retornaría una excepción del tipo NoExisteException. Para la creación también, si se intenta crear un miembro con un DNI existente pues retornara una excepción del tipo DuplicadoException.
   1. Obtener miembro



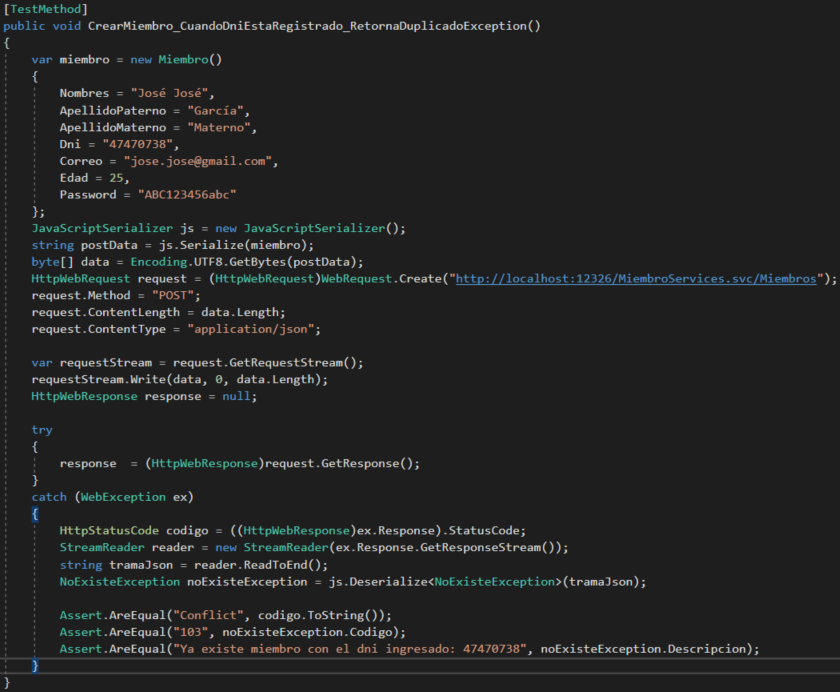
* 1. Obtener miembro con excepción



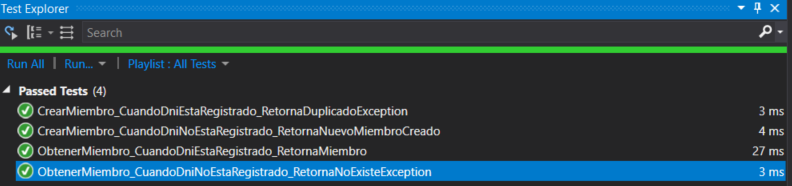
* 1. Crear miembro



* 1. Crear miembro con excepción



* 1. Resultado de la ejecución de las pruebas



**¡Eso es todo amigos!**