Proyecto PSEP

Ioritz Perugorria

Estructura general del proyecto:

El proyecto tiene 3 programas que se ejecutan independientemente: Un servidor asíncrono, un servidor de API Rest, y un número de clientes. Con el servidor asíncrono y la API ejecutándose, se puede ejecutar tantas instancias de clientes como se deseen, y el servidor manejara a cada usuario independientemente. Al iniciar un usuario, se mostrará un menú con opciones CRUD, y tras elegir una, enviará un mensaje al servidor. Este mensaje se enviará encriptado, y se desencripta al recibirse en el servidor. El servidor interpreta el contenido del mensaje, y realiza la operación necesaria. Tras terminar, enviará un mensaje de vuelta al cliente, encriptado, y se interpretará y mostrará al cliente.

En cuanto al código, la mayoría de la base parte de prácticas que hemos hecho en clase, con las modificaciones y añadidos necesarios para que se cumplan los requisitos del proyecto. La API se adecua al modelo de datos que se presenta en la segunda entrega, y entre el cliente y el servidor se envían datos encriptados.

Para ejecutar el programa: abrir en una terminal el fichero “APIProgram.cs” y ejecutar con “dotnet run”. Luego, abrir en otra terminal el fichero “server.cs” y ejecutarlo también con “dotnet run”. Por último, abrir tantas instancias del fichero “Client.cs” como se deseen y ejecutarlos con “dotnet run”

Cliente-Servidor:

Los protocolos de comunicación entre el cliente y el servidor son bastante similares a las prácticas anteriores, siendo la mayor diferencia la interpretación de los datos.

Cuando un cliente se conecta al servidor, se muestra un menú con varias opciones, todas opciones CRUD. Al seleccionar alguna, envía la información al servidor respecto a qué operación se desea realizar. En caso de necesitar datos del usuario, estos se pedirán al cliente y se enviarán al servidor junto a la operación a realizar. Por ejemplo, al seleccionar la opción de obtener por id, el programa le preguntará al usuario el id del registro a obtener (por ejemplo, 47), y enviará al servidor una cadena de texto con el tipo de operación e id (en este caso, “GET47”). El servidor recibe la información, comprueba que operación CRUD es, extrae la información y la interpreta.

Cuando el servidor termina de interpretar la información junto a la API, este enviará un mensaje de respuesta al cliente, con datos respecto a la operación que se ha realizado. Por ejemplo, si se ha eliminado un registro, devolverá “Se ha eliminado el usuario x”, y si se solicita un usuario por id, devolverá el json del registro.

API Rest:

En un programa aparte en la carpeta UserApi se levanta la API de datos. Contiene un modelo de datos (UserItem) y en base a este modelo hay varias operaciones CRUD. La estructura es la misma a una práctica anterior, cambiando el modelo y añadiendo las operaciones necesarias para el proyecto.

Para que el servidor se pueda comunicar con la API, se hace uso de un objeto HttpClient, el cual se le asigna la url de la API. Este objeto contiene métodos CRUD, por lo que tan solo es necesario interpretar la información del usuario de manera que se pueda realizar la petición exitosamente. Por ejemplo, cuando el servidor reciba el mensaje “GETALL”, realizará una consulta Get con la ruta del getAll de la API, y guardará la respuesta en un objeto HttpResponseMessage , que luego se transforma a texto y se envía al cliente.

Encriptación:

En cuanto a la encriptación de datos, se encriptan los datos que se envían entre el cliente y el servidor mediante el método AES. Está la clase AesEncryption, que contiene métodos que se encargan de encriptar y desencriptar información a arrays de bytes y strings respectivamente. Las clases del servidor y el cliente tienen cada uno su clase. Eso implicaría que tendrán la Key e IV diferentes, por lo cual cada vez que se establece conexión el cliente envía su clave al servidor para que la compartan. A partir de allí, cada vez que se va a enviar información entre los dos, se encripta y desencripta la información.

Nota: Por varios problemas, no he podido terminar la implementación de esta encriptación. Todo el código sigue en el proyecto, solo que esta comentado o si usar