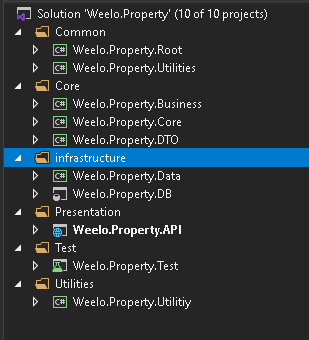
1. Descripción del proyecto

El back end se hizo basado en una arquitectura limpia donde cada una de las capas tiene su responsabilidad:

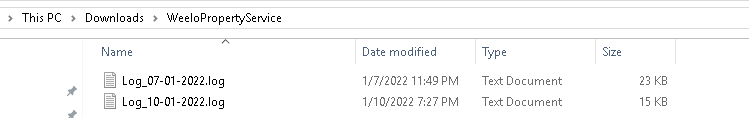


* 1. Root

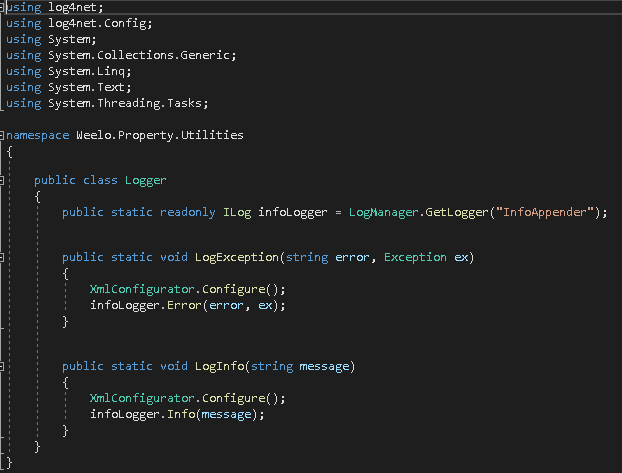
Este proyecto tiene la configuración para manejar la inserción de dependencias de toda la solución. Para realizar esta inserción se utilizó el package Microsoft.Extensions.DependencyInjection

* 1. Weelo.Property.Utilities

Proyecto que manejaría funcionalidades transversales que podrían utilizarse en todas las capas de la solución. Por ejemplo, en esta existe una clase para el manejo del logger por medio deLOg4net







* 1. Business

Proyecto que tiene toda la lógica de negocio este compuesto por:

* Services 🡪 funcionalidades de negocio, ejemplo consultar propiedades
* Mappers 🡪 se encuentran las trasformaciones que se hacen para cumplir con la necesidad de negocio. Ejemplo convertir un objeto de tipo base de datos a uno de negocio
  1. Core

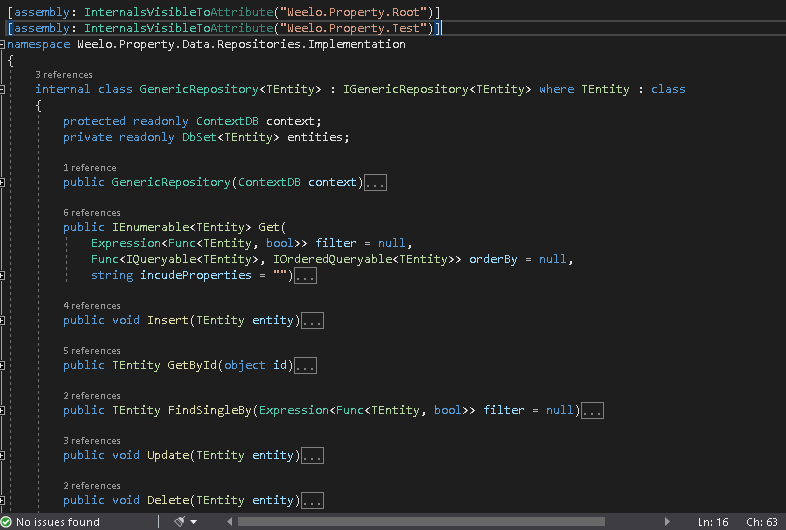
Proyecto donde se encuentran las clases de los objetos de base de datos.

* 1. DTO

Proyecto donde se encuentra las clases de los objetos de negocio.

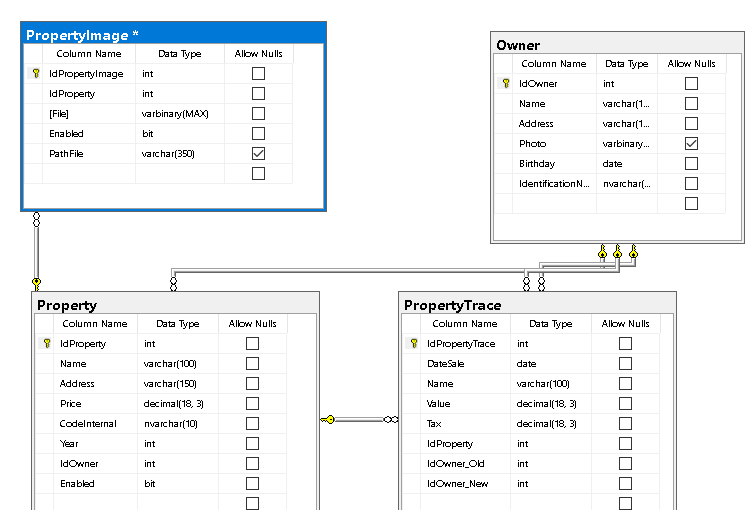
* 1. Data

Proyecto que contiene los métodos para poder interaccionar con la base de datos. Se hizo una clase genérica con el objetivo de que se pueda interaccionar con cualquier entidad de la base de datos y cualquier operación, lo cual evita la generación de código de los repositorios de cada oibjeto de la base de datos. Por medio de esta clase se pueden interactuar con Property,Owner, PropertyImage y PropertyTracking



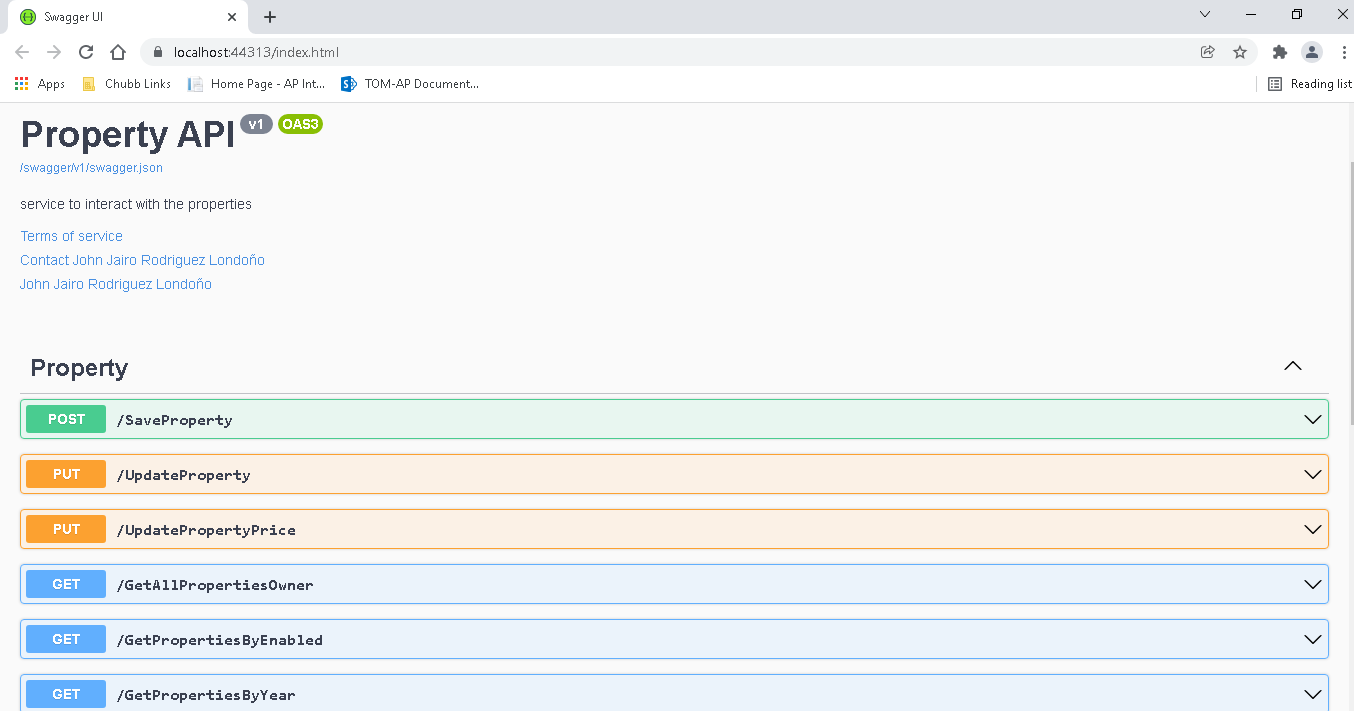
* 1. DB

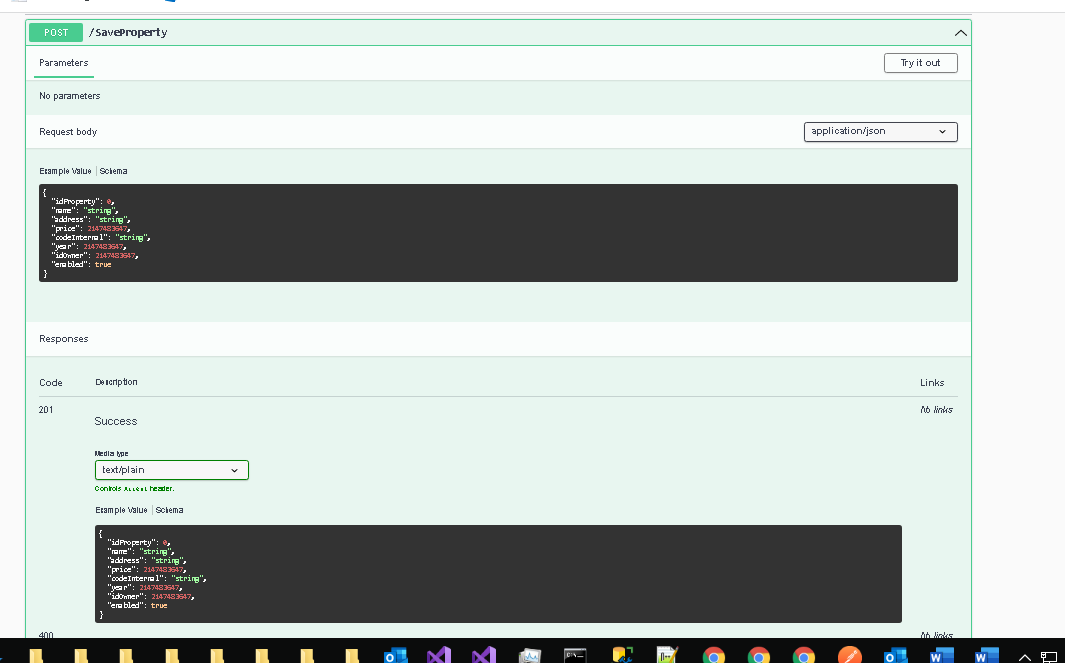
Proyecto que contiene todos los objetos y scripst de base de datos Configurado con SQL server 2014. No se pudo adicionar la data porque la tabl de owner e PropertyImage tiene valores de Varbinary(Max), se debe ejecutar el script que se adjunta por el SQL server Management



* 1. Weelo.Property.API

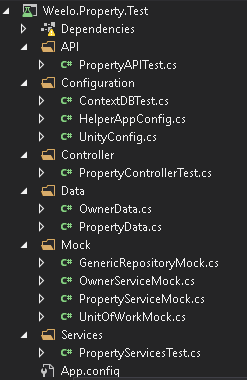
Proyecto que contiene las apis del proyecto Property, Manejo de Swagger para la documentación de las funcionalidades del servicio



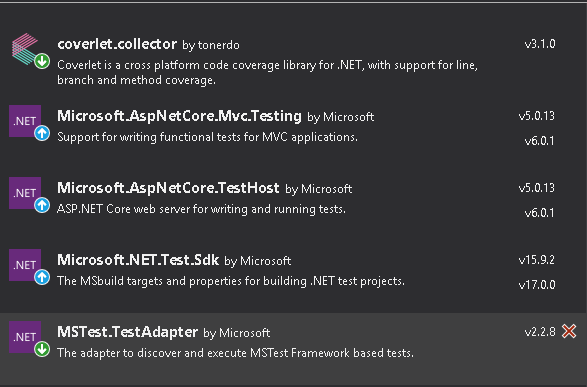
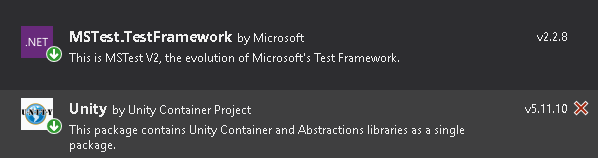


* 1. Test

Proyecto de pruebas donde se prueba cada una de las capas del proyecto, con sus correspondientes mock o componentes directos.



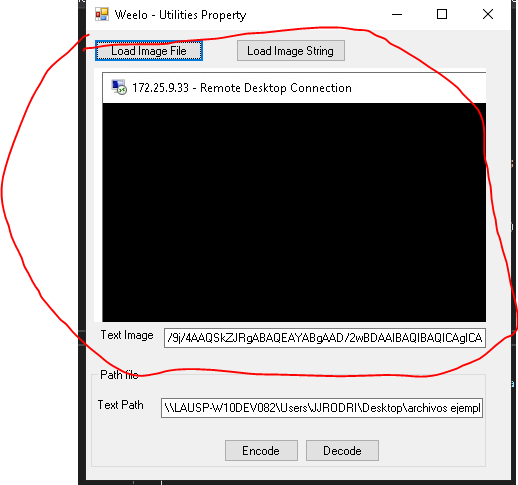
Componentes utilizados:

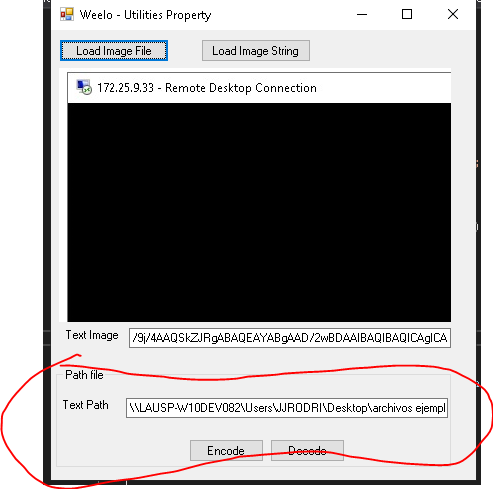
* 1. Weelo.Property.Utilitiy

Aplicación de Escritorio que sirve para:

Convierte una imagen en un String y un string en una imagen:



Codificar y decodificar la ruta del archivo donde se deja la imagen que se adiciono

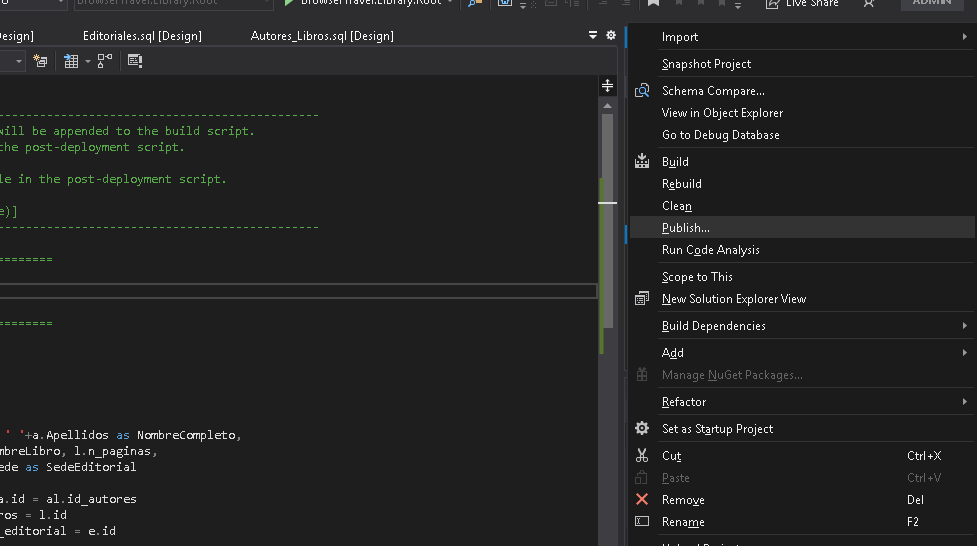


Nota estas funcionalidades se hicieron porque la imagen no va a quedar almacenada como image en base de datos sino como un varbinary o la ruta de la ubicación de un repositorio.

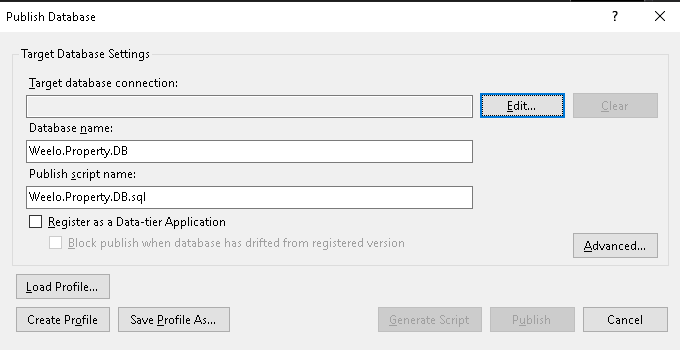
El servicio no tiene capacidad de almacenar archivos se deja a responsabilidad del Front end

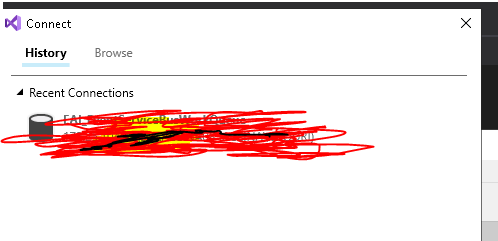
1. Instalación SCRIPTS base de datos

En el paquete hay un proyecto llamado Weelo.Property.DB carpeta infraestucture. Clic derecho y seleccionar la opción Publish

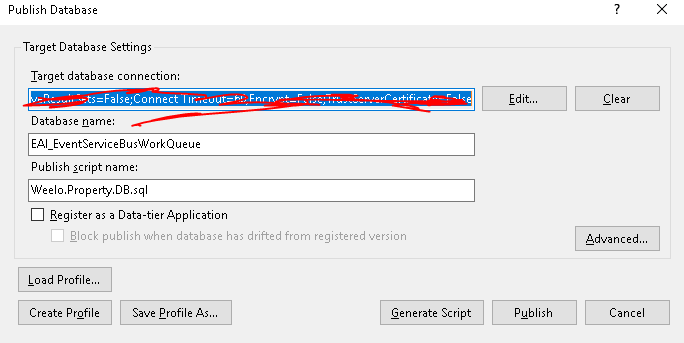


Se hace clic en el botón EDIT y se coloca la configuración de la base de datos.

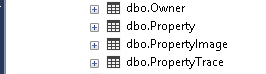




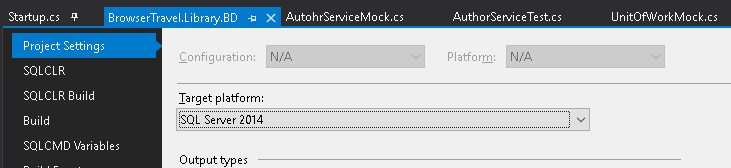
Luego de tener configurado la cadena de conexión y nombre de base de datos se hace clic en el botón Publish



Se verifica en la base de datos que hayan quedado las tablas y el SP



***Nota: hay que tener en cuenta que versión de SQL se tiene en la maquina el proyecto esta con SQL 2014***



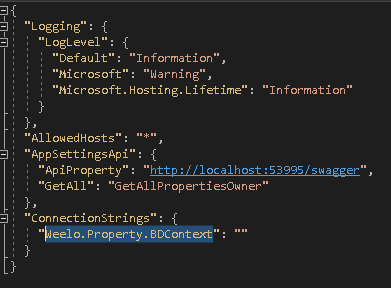
Ejecutar el archivo Script en la base de datos creada:



1. Ajustes Configs

Se debe configurar los config tanto del API como del proyecto de pruebas para la cadena de conexión a la base de datos





La configuración del archivo log es en el archivo app.Config

