PROYECTO FINAL DIVISAS TOMASZ DOMINIK DADAL

SERVICIOS UTILIZADOS:

* REPOSITORIOS: donde se ha creado un GenericRepository. De esta clase heredan los DivisaRepository y TransaccionRepository. Estos a su vez tienen los métodos del CRUD para poder ser inyectados en sus respectivos controladores. De la misma manera que se han creado los repositorios se podría crear un FakeRepository para poder hacer pruebas sobre él. Este únicamente necesitaría heredar del GenericRepository y hacer en el una inyección de datos diferentes pero de la misma clase que los obtenidos del WEBAPI divisas y transacciones. En el TransaccionRepository se han creado algunos métodos para realizar consultas LINQ sobre el repositorio.
* CONTROLLERS: Se han creado tres controllers, dos para operar sobre los repositorios y el tercero para hacer las respectivas queries de consulta sobre el TransaaccionRepository. Asimismo se ha creado un BaseController del que heredan los demás controladores para llevar a cabo un buen manejo de las excepciones.
* JSON CONVERTER SERVICE: este servicio nos permite obtener datos del web api y convertirlos a los modelos de divisa y transacción que utilizamos en el proyecto.
* LOG SERVICE: se ha utilizado para logear en un documento que se crea dentro del directorio del proyecto las excepciones que se puedan producir en los controladores.
* DATA ACCESS LAYER: con un CambioDivisasRepository el cual nos crea una base de datos local para con los datos obtenidos de la WebApi , del cual además obtienen los datos los repositorios.
* SPECIFICATIONS: para comprobar que los datos obtenidos de la WebApi son correctos, para que en caso de que no lo sean, no inyectarlos en el repositorio.

Se ha creado, asimismo, una carpeta llamada TransversalInfraestructure donde creamos nuestras excepciones personalizadas para controlar los fallos que puedan dar algunos servicios.

Por último, se han realizado test sobre los controladores, servicios y repositorios.