Infocorp

Evaluación

Autor: Leonardo Lazbal
llazbal@gmail.com
15/01/2016

Introducción

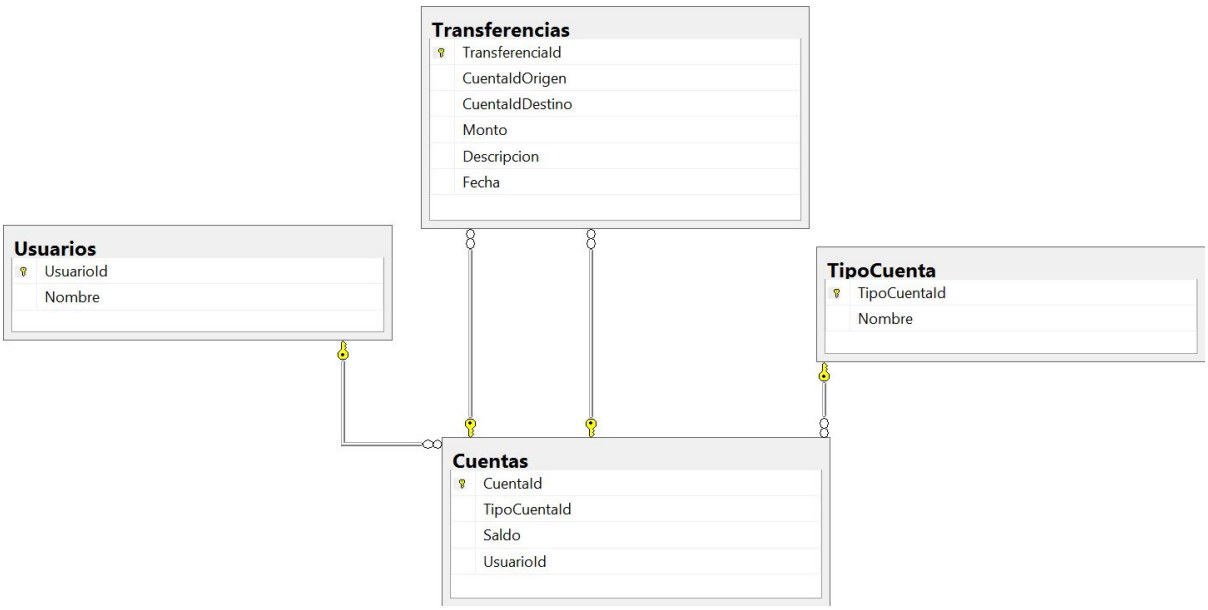
Como lo indican los requerimientos de la evaluación se generó una solución de Visual Studio 2013 con una aplicación web .NET MVC 4 con AngularJS.

Funcionalidades

- Listado de cuentas
- Muestreo de datos de la cuenta seleccionada en el listado
- Listado de transferencias
- Generación de transferencias
    - Verifica la selección de la cuenta origen en el combo
    - Verifica el ingreso de la cuenta destino
    - Verifica que la cuenta origen y destino sean distintas
    - Verifica el ingreso del campo monto
    - Verifica que la cuenta origen tenga saldo suficiente
    - Actualiza la cuenta origen y destino

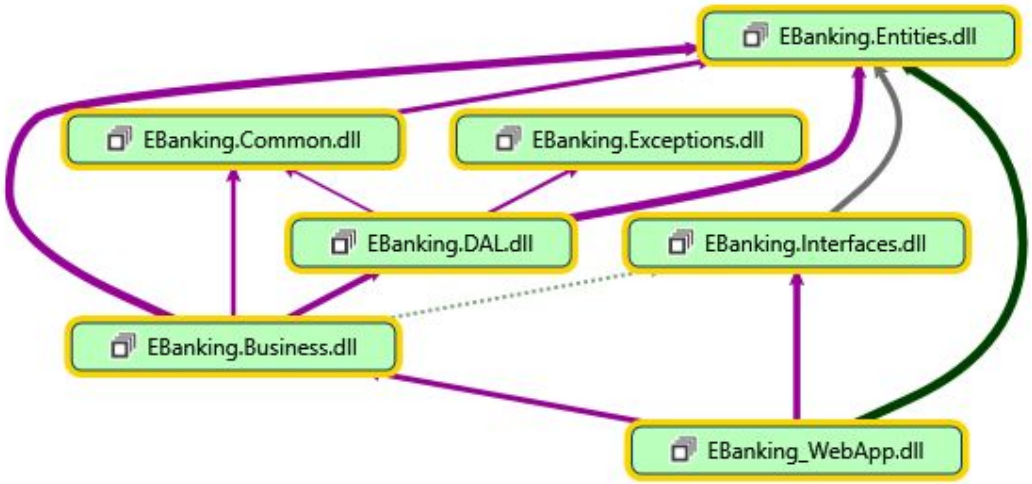
Diseño

Base de datos

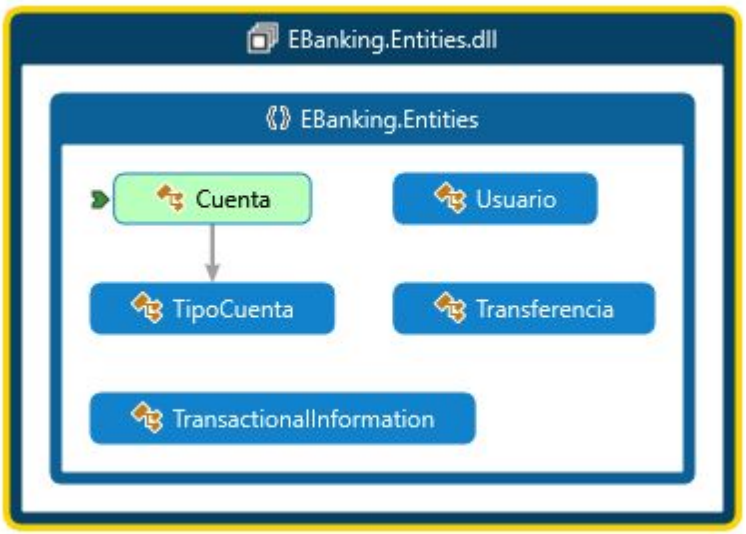


Proyecto Web (Solución de Visual Studio)

Librerías

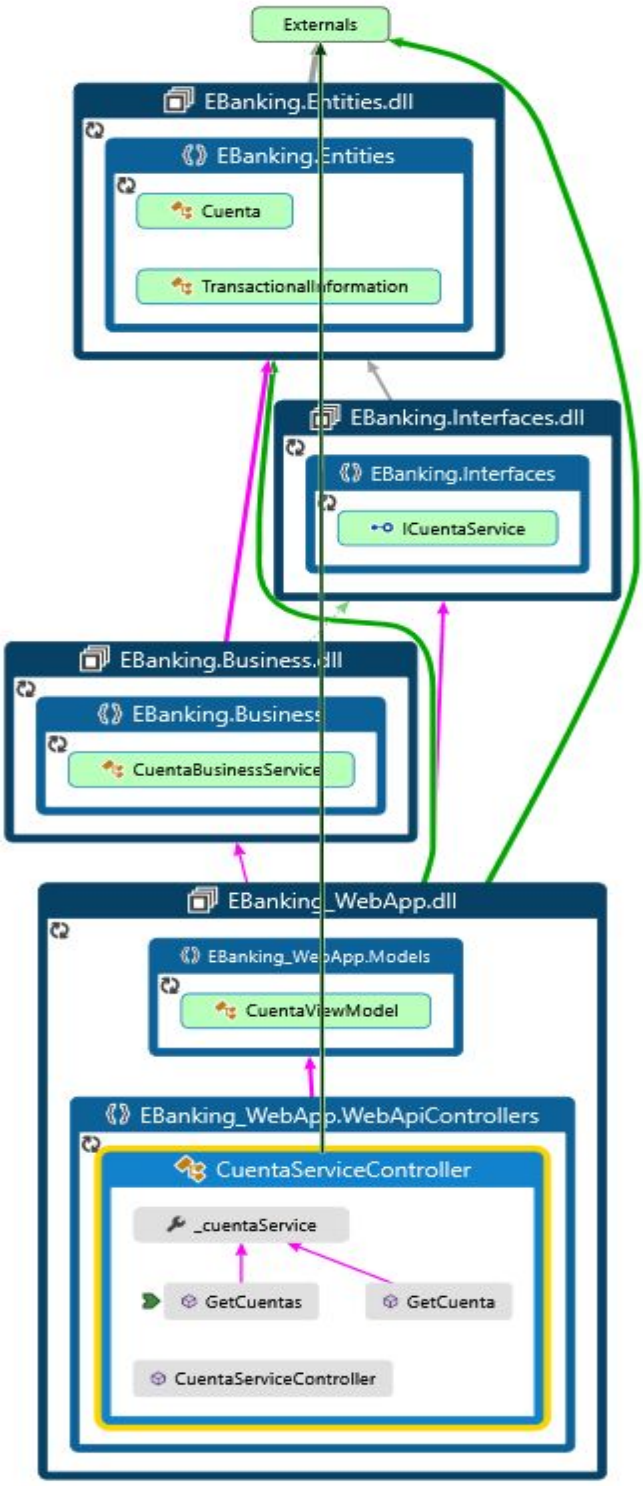


Entidades

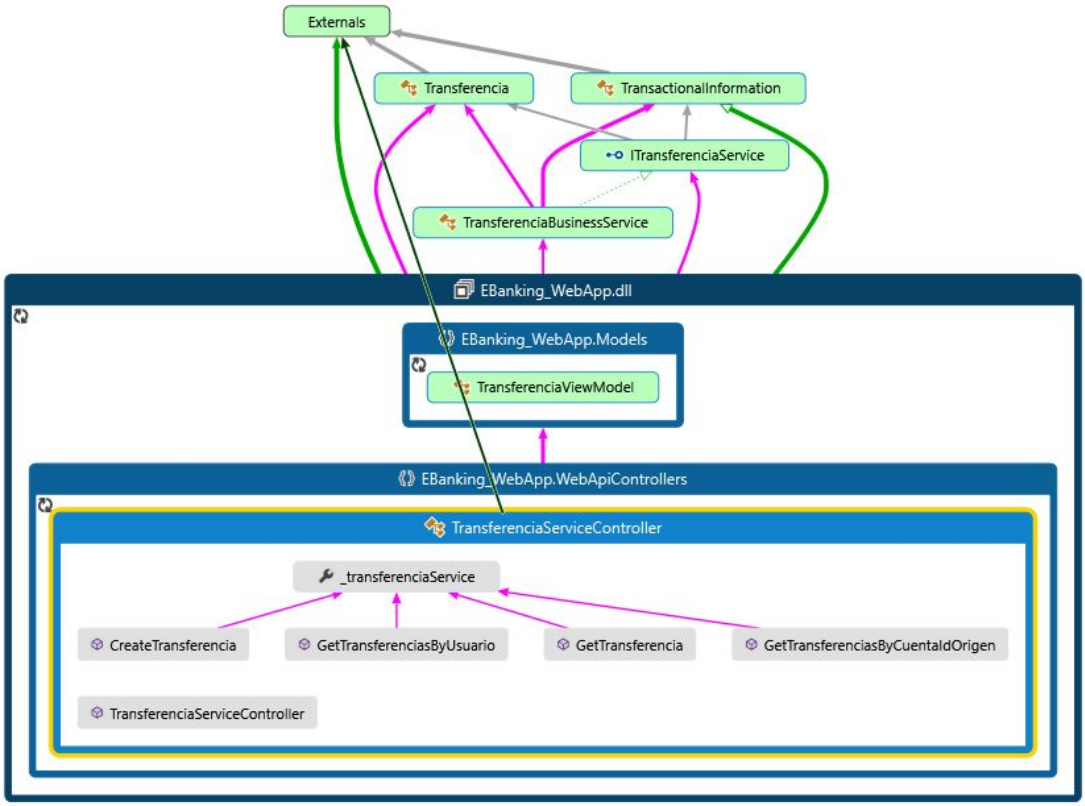


Flujo de datos

Cuentas



Transferencias



Mejoras

Manejo de usuarios

La aplicación no contempla el manejo de usuarios, por lo que se asume que el usuario logueado es el que tiene identificador = 1, este valor es generado por los scripts sql adjuntos a este documento.

Objeto Transferencia

Este objeto posee las propiedades int CuentaIdOrigen e int CuentaIdDestino, un mejor diseño sería usar objetos Cuenta de modo de poder acceder a información de las mismas en el muestreo de datos de transferencias. Para cargar esos objetos se debería aplicar el mismo modo que se hace con el objeto referenciado TipoCuenta en la clase Cuenta, lo cual consiste en

Customización de excepciones

El manejo de excepciones customizadas es conveniente al momento de unificar mensajes de error. Un ejemplo de su uso es la clase TransactionException ubicada en el proyecto EBanking.Exception. Esta clase hereda de System.Exception y contiene un enumerado con mensajes de error que son invocados mediante el método GetEnumDescription desde el constructor que recibe como parámetro ese enumerado.

Log de errores

Las librerías System.Diagnostics.TextWriterTraceListener son una buena opción para la generación de archivos de log que muestren información de los pasos que realiza la aplicación.

Archivo de configuración

Acceder a archivos con objetos serializadores (ej.: System.Xml.Serialization.XmlSerializer) son convenientes para la lectura de datos como ser strings de conección con datos encriptados que posteriormente la aplicación debería desencriptar. Librerías del espacio de nombres System.Security.Cryptography

aplican a esto.

Formateo de datos

Generar clases que apliquen formatos específicos a determinados tipos de datos como ser fechas y números unifican (simplifican) el muestreo de datos.

Tiempo de realización de proyecto

Proyecto en Visual Studio: 15
Base de datos: 3 horas