**Solucion**

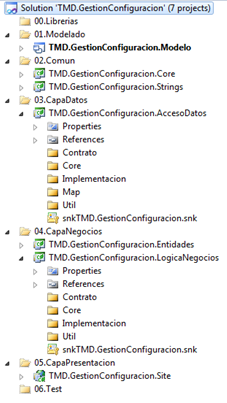
**Nombre de la Solución:** [compañía].[Abreviatura de cada módulo]

Ejemplo: TMD.GC

**Nombre de los proyectos:** [compañía].[Abreviatura de cada módulo].[funcionalidad]

Ejemplo: TMD.GC.LogicaNegocios

**[FALTA ACTUALIZAR IMAGEN]**



**Nomenclatura:**

**Proyectos**

**TMD.GC.AccesoDatos**

TMD.GC.AccesoDatos.Contrato

Prefijo: I

Ejemplo: IProyectoData

TMD.GC.AccesoDatos.Implementacion

Sufijo: Data

Ejemplo: ProyectoData

**TMD.GC.LogicaNegocios**

TMD.GC.LogicaNegocios.Contrato

Prefijo: I

Ejemplo: IProyectoLogica

TMD.GC.LogicaNegocios.Implementacion

Sufijo: Logica

Ejemplo: ProyectoLogica

**Codificacion**

**Pascal case (NombreDeElemento)**

* Espacios de Nombre: **namespace TMD.GC.LogicaNegocios.Implementacion**
* Clases: **public class ServicioPaypal**
* Métodos: **public PaypalResponseDTO CrearCuenta()**
* Propiedades: **public string CodigoTransaccion{ get; set; }**
* Interfaces: **public interface IServicioPaypal**
* Enumerados: **public enum TipoPago**

**Camel case (nombreDeElemento)**

* Variables locales: **IPaypalAccountLogic \_lnPaypalAccount;**
* Parámetros: **public ServicioPaypal(IPaypalAccountLogic lnPaypalAccount)**

**Comentarios**

Todas las clases, interfaces y enumeradores deben especificar la funcionalidad que cumplen

**/// <summary>**

**/// Realiza operaciones en Paypal**

**/// </summary>**

**public class ServicePaypal : IServicePaypal**

**{**

Todos los métodos deben especificar la funcionalidad que cumplen y se tiene que describir cada parametro que reciben.

**/// <summary>**

**/// Asocia una tarjeta de credito a un cliente en paypal**

**/// </summary>**

**/// <param name="paymentCardRequest">Objeto PaymentCardDTO</param>**

**/// <returns>PaypalResponseDTO</returns>**

**public PaypalResponseDTO AddPaymentCard(PaymentCardDTO paymentCardRequest)**

**{**