Entidades (BE)

Las entidades representan a la fachada de una o más tablas del modelo de entidad relación o por otro lado pueden estar incluidas unas peticiones Request a un web service. Es decir con una entidad podemos representar una tabla Cliente o hasta el patrón Cabecera Detalle de una factura.

Dicha fachada especifica de forma fuertemente tipificada las propiedades que corresponden a los atributos de las tablas en cuestión o de los atributos combinados de distintas tablas que correspondan a un proceso de negocio representado por un Request/Result.

Las entidades son utilizadas por los componentes de negocio o por los servicios de negocio (transaccionales o de consulta). Por dicha razón se encuentra en color GRIS en contacto con estos tres módulos.

Entidades Simples

Las *entidades simples* contienen datos escalares de una o más tablas o pueden contener datos de una colección de entidades o hasta una entidad. Dichas entidades están compuestas por una serie de propiedades Get y Set que permiten el acceso a dichos datos.

Colecciones

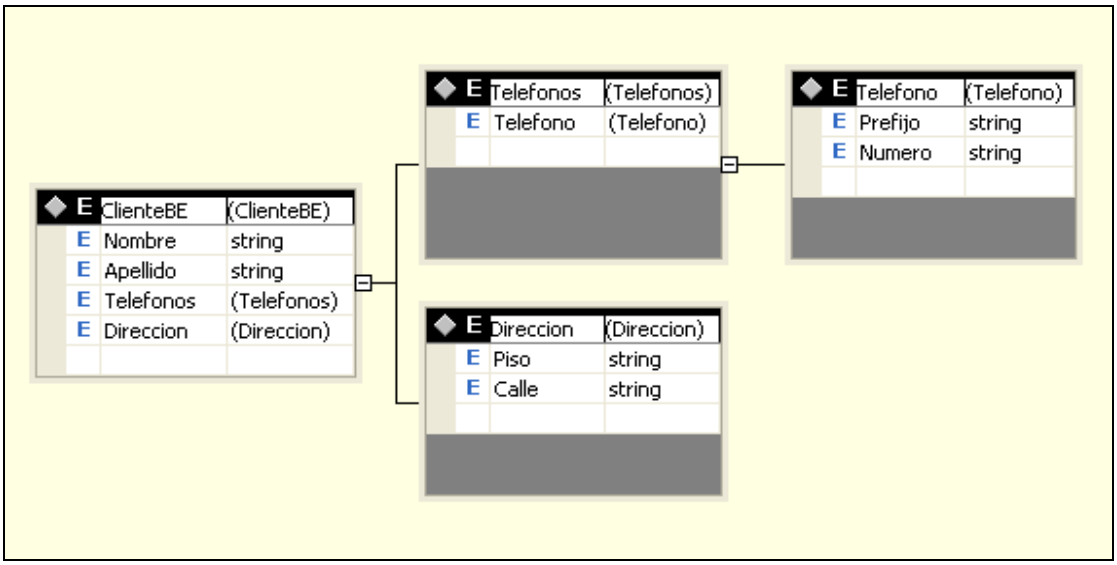
Las *entidades colecciones* son un conjunto de entidades simples. Las mismas están compuestas por una serie de métodos que me permitan realizar operaciones sobre la colección a efectos de permitir:

- Agregar una entidad de negocio simple
- Buscar una entidad de negocio
- Obtener un subconjunto de entidades de negocio
- Iterar por la colección
- Retornar xml o DataSet de la colección misma.

Ejemplo que detalla las distintas formas de relacionar colecciones con una entidad:
Podríamos tener una entidad Cliente que tenga una colección de de teléfonos como propiedad. Al mismo tiempo, si el negocio exige, la entidad de Cliente necesita almacenar la dirección, pero resulta que los clientes tienen siempre una sola dirección entonces el diseñador (un tanto limitado en este caso) decide establecer una propiedad a la clase Cliente de tipo complex type Direccion, y no utiliza una colección de direcciones. En este caso la clase cliente contiene como propiedad una entidad . (Direccion).

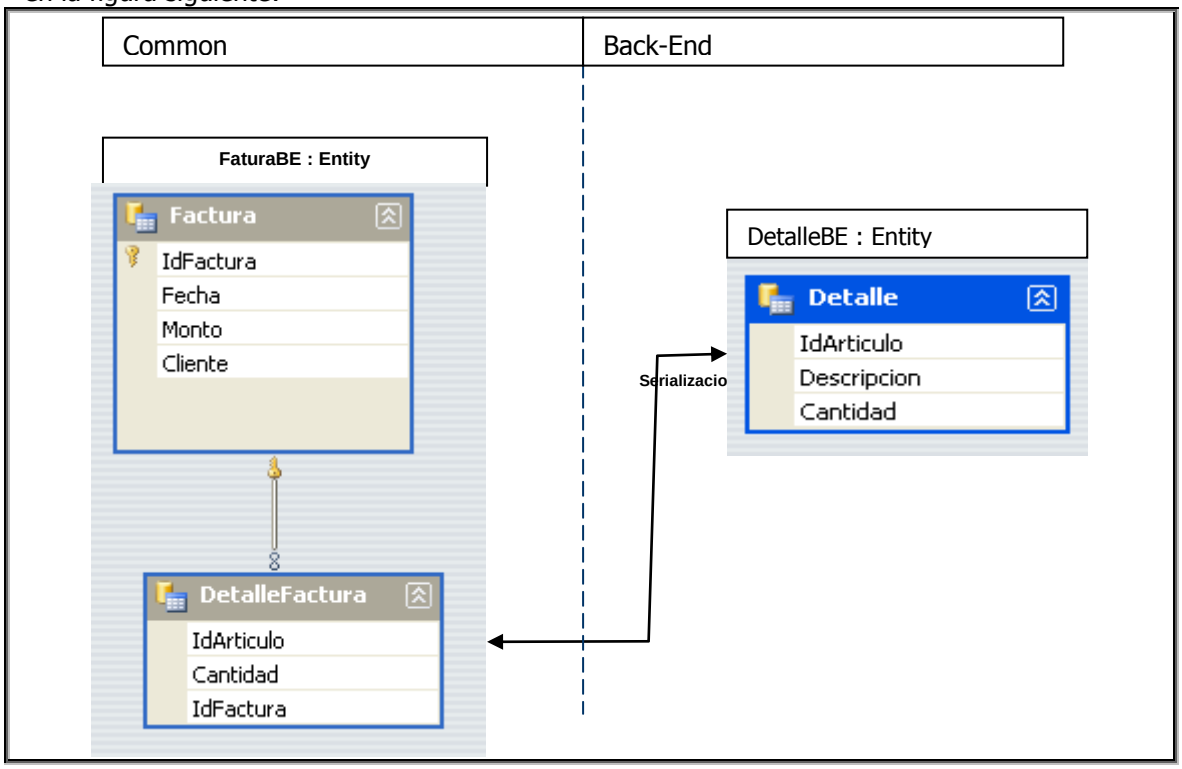
De esta forma podemos usar la clase cliente de la siguiente manera:

```
 ClienteEnt wCli = new  ClienteEnt();

       WCli.Nombre = "Marcelo";
       WCli.Nombre.Direccion.Calle = "Sarmiento";
       WCli.Nombre.Direccion.Piso = 2;
       WCli.Nombre.Telefonos[1].Numero = 1231244;

Donde:
     ClienteEnt : Entity
Dirección  : Entity
 Telefonos  : Entities
```



Serializacion

Otra de las características de las entidades es que son Serializables lo cual permite obtener un DataSet o un xml de la misma muy sencillamente. Además nos permite generar entrara para otra entidad más compleja que requiera el formato de la entidad en cuestión, como se muestra en la figura siguiente:



Entidades de relación (ER)

Las entidades de relación son colecciones utilizadas para representar relaciones de muchos a muchos, del modelo de datos. Si bien se comportan de la misma forma que las colecciones de entidades de negocio, se las define de diferente manera dado que no representan a una entidad de negocio en si misma. Es decir siempre serán utilizadas en relación con otras entidades.

EJ: ContratoComercialCondicionesComercialesBE

En esta entidad relación esta claro que relaciona dos entidades ContratoComercialBE y CondicionesComercialesBE

Por que tendríamos la necesidad de definir una ER ? Pues es necesario en algunos casos almacenar o mantener información de la relación de ambas tablas a las que se refieren las dos entidades anteriores. Podría ser que no solo se necesite los IDs de ambas tablas sino otra información que provenga del seteo de valores obtenidos por el sistema o configurados en una pantalla por el usuario.

Clases Base Entity/Entities

Todas las entidades heredan de una clase llamada Entity. Dicha clase contiene funcionalidad genérica para todos los tipos de entidades simples (un ejemplo de funcionalidad genérica puede ser GetXml() que retorna el xml de la entidad).
Todas las entidades poseen una entidad contenedora que hereda de Entities del framework de Action Line. Esta clase contenedora es la colección de la que hablamos arriba.
A continuación se visualiza un ejemplo de una entidad y su colección:

```
using System;
using System.Xml.Serialization;
using System.Collections.Generic;
using Fwk.Bases;
using helper = Fwk.HelperFunctions;
using System.ComponentModel;
namespace Pelsoft.SitemaContable.BackEnd.Facturacion BE
{
       // Entidad de tipo colección.
[XmlRoot("ClienteList"), SerializableAttribute]
public class ClienteList : :Entities<Cliente>
{
        //Métodos que sean útiles como:
        //buscar por Nombre,  por Id etc.-
}

// Clase base de las colecciones.
  [XmlInclude(typeof(Cliente)), Serializable]
    public class Cliente:Entity
    {

        #region [Atributes]

        private DateTime? mdt FechaNacimiento ;
        private string msz Nombre ;
        private string msz _Apellido;
        private int? msz _Edad;

        #endregion


        #region [Properties]

                public DateTime? FechaNacimiento
        {
            get { return dmt FechaNacimiento; }
            set { mdt_FechaNacimiento = value;  }
        }

        public string Nombre
        {
            get { return msz Nombre; }
            set { msz_Nombre = value; }
        }

        public string Apellido
        {
            get { return msz _Apellido; }
```

```
            set { msz _Apellido = value;  }
        }

        public int? Edad
        {
            get { return msz  Edad; }
            set { msz _Edad = value;  }
        }

         #endregion
  }
}
```

Conclusiones acerca de las entidades simples:

- Las entidades heredan de la clase Entity. Dicha clase encapsula funcionalidad genérica para todas sus derivadas.
- Las entidades de tipo simple solo contienen datos, todo comportamiento de negocio se encuentra en los componentes de negocio.
- Todas las entidades son serializables. Al igual que los request y response de los servicios de consulta o transaccionales.
- Todas las entidades pueden exponer sus datos a través de GetXml() o GetDataSet() .
- Solo contienen datos y comportamiento técnico, no contienen comportamientos funcionales de negocio.

Conclusiones acerca de las entidades tipo colecciones:

- Las colecciones exponen sus ítems como si fueran entidades simples tipificadas. Ejemplo, GetClienteById retorna una entidad simple de tipo Cliente.
- Son serializables.
- Debido al que retornan DataSet, las colecciones exponen métodos de búsqueda optimizados y ya desarrollados.
- Heredan de una clase base llamada Entities. Las colecciones se basan en listas bajo la siguiente definición:

  Entities<T> : List<T> where T : Entity

- Las clases bases Entities al heredar de List<T> poseen todas las ventajas de las listas genéricas.- http://msdn2.microsoft.com/en-us/library/s6hkc2c4.aspx
- Todas las colecciones deberían mantener un mismo patrón, es decir existe un conjunto de métodos comunes a todas. Ejemplo, el método GetClienteByName es un método común a todas las colecciones (solo cambia Cliente por alguna otra entidad).

```
// Entidad de tipo colección.
        public class ClienteList : Entities<Cliente>
        {
         public Cliente GetClienteByName
{
                        //Implementacion del metodo.-
}
        }
```