

## Objetivos

Al finalizar el desarrollo de este ejercicio, el estudiante:

- Habrá utilizado algunos de los básicos de aplicaciones Web en ASP .Net y se habrá familiarizado con la forma como deben configurarse dichos controles, para su correcto funcionamiento y el manejo de eventos en el framework .Net
- Habrá evidenciado a través del consumo de un servicio Web, las ventajas de la utilización de la tecnología .Net para la integración de recursos o aplicaciones desarrollados en diferentes plataformas por diferentes equipos de desarrollo.
- Habrá aplicado dos de las posibles formas de arquitecturas de diseño utilizadas en el desarrollo de aplicaciones con tecnología .Net: El patrón MVC (modelo-vista-control) y el enlace directo a los datos desde la vista.

## Metodología

A continuación se presenta el enunciado del ejercicio y una serie de pasos para su desarrollo. El estudiante debe seguir cada una de las instrucciones que en algunos casos son muy específicas y en otros casos buscan que el estudiante solucione el problema planteado apoyándose en la documentación de .Net [http://msdn.microsoft.com/en-us/library/w0x726c2\(VS.100\).aspx](http://msdn.microsoft.com/en-us/library/w0x726c2(VS.100).aspx)

## El Ejercicio

Se trata de una sencilla aplicación para el registro de los juguetes que una distribuidora despacha a sus clientes. De la distribuidora (este sistema soporta únicamente la información de una distribuidora), se registra en el sistema su NIT y su nombre. De cada juguete se almacena en el sistema el nombre, la edad mínima y máxima recomendada para quienes jueguen con él, la marca y las unidades máximas disponibles en bodega.

Además del registro de la información del inventario de cada juguete, debe ser posible la consulta de sus datos.

El sistema debe permitir la consulta de los productos (juguetes) disponibles en bodega para una marca determinada y retirar de inventario un número de unidades indicado por pantalla (no pueden retirarse más unidades de las disponibles). La consulta debe mostrar por pantalla la identificación, el nombre, el tipo y las unidades disponibles, así como el histórico de entradas y salidas de inventario.

Requerimientos especiales:

- Debe validarse al ingresar un nuevo juguete que no haya sido registrado anteriormente. Se tendrá en cuenta la aplicación de las buenas prácticas de la POO (por ejemplo sobrescritura de métodos equals, etc).
- Las marcas de juguetes con las cuales trabaja la distribuidora, son suministrados en una base de datos. El estudiante deberá conectar la aplicación a la base de datos para desplegar a través de un control tipo combobox, marcas disponibles.
- Cuando se consulte por pantalla o se genere el archivo plano de un juguete de tipo *personaje*, debe realizarse la conversión de su tamaño de pulgadas a centímetros.

Para esto DEBE consumir el servicio Web disponible en la siguiente URL  
<http://www.websvcex.net/length.aspx>

- Cuando se hace un retiro de inventario, debe almacenarse el registro de salida capturando la fecha y hora actual del sistema.
- La información de las fechas deben mostrarse en formato mm/dd/year, por ejemplo 23/07/2007 y la hora en formato hh:ss am, por ejemplo 10:35 am.

NO SE REQUIERE QUE LA APLICACIÓN IMPLEMENTE PERSISTENCIA CON LA BASE DE DATOS, la interacción con está será únicamente para la consulta de las marcas con las que trabaja la distribuidora.

### Desarrollo

1. Cree un proyecto de tipo aplicación Web de acuerdo al diseño visual que usted prefiera. La aplicación debe involucrar mínimo dos páginas y debe ser posible el llamado a la funcionalidad de registro y consulta a través de un menú.
2. Debe desarrollar el ejercicio implementando el patrón MVC.
3. En el proyecto de la lógica debe crear las clases de la lógica con el código necesario para el funcionamiento correcto de la aplicación. Debe basarse en el diagrama de clases mostrado en la figura 2 para la implementación de la capa lógica, es necesario que complete el diagrama:

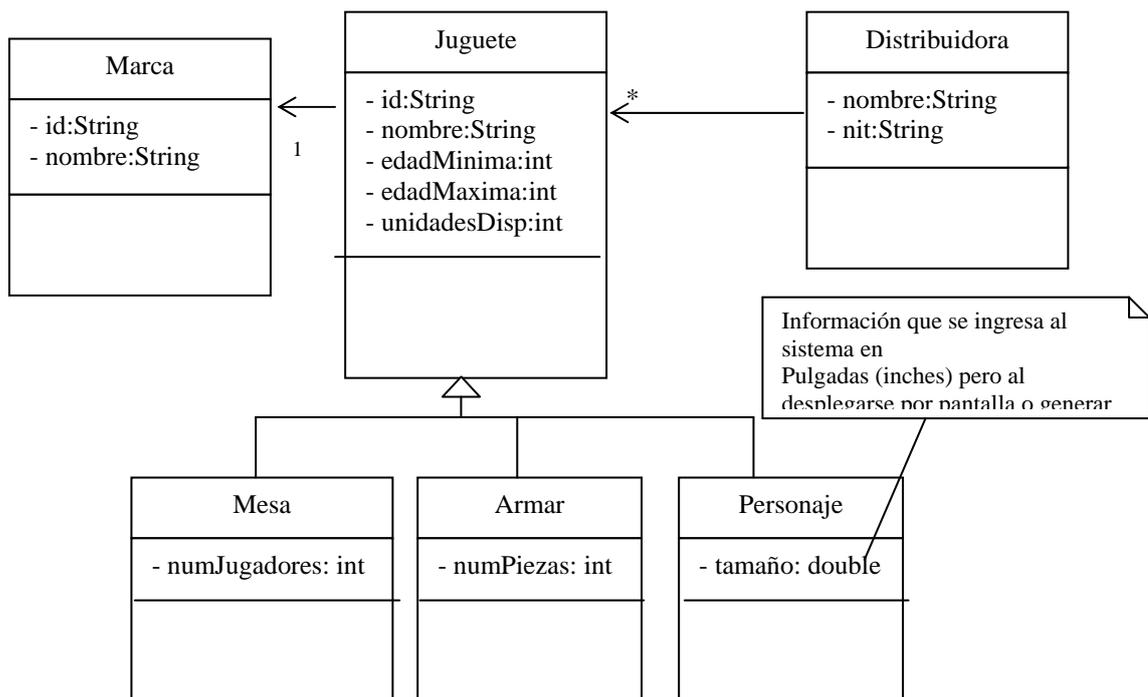


Figura 1 – Diagrama de clases

4. En el proyecto control, debe incluir lo necesario para lograr la comunicación entre las diferentes capas de la aplicación.
5. Para enlazar el combobox a la información de las marcas de juguetes en la base de datos.
  - a. Cree un archivo extensión XML en algún lugar de su proyecto con el siguiente cuerpo y adiciónelo al proyecto de la vista

```

<?xml version="1.0" encoding="utf-8"?>
<marcas>
  <marca>Matel</marca>
</marcas>
  
```

```
<marca>Fisher Price</marca>
<marca>Coquito</marca>
<marca>BabyToy</marca>
<marca>Hashbro</marca>
</marcas>
```

- b. Use los controles de Datos para adicionar el de XMLDataSource y configúrelo para leer ese archivo XML que usted creó.
  - c. Clic sobre la flecha para activar el menú contextual del control de tipo ComboBox.
6. Para referenciar el servicio Web que le permitirá hacer la conversión entre dos unidades de medida:

Un servicio Web es una aplicación que provee servicios a otras aplicaciones. Los servicios Web serán estudiados con mayor detalle en la sección de desarrollo avanzado con ASP.Net. Esta parte del ejercicio le permitirá consumir un servicio Web gratuito.

- a. En el explorador de soluciones, hacer clic derecho sobre el proyecto y seleccionar la opción “agregar referencia Web”.
- b. En el cuadro de diálogo de descubrimiento de servicios Web, ubique el url <http://www.websvcex.net/length.aspx> en el campo “Dirección URL” y haga clic en IR. En este momento, .Net está tratando de ubicar el servicio Web que usted ha especificado. Este es un servicio Web gratuito convierte valores de una unidad de medida a otra (puede mirar la descripción del servicio y el código XML para obtener información adicional). Si pulsa clic en el método proporcionado por el servicio, podrá probar su funcionamiento. Pulse clic en el botón agregar referencia.

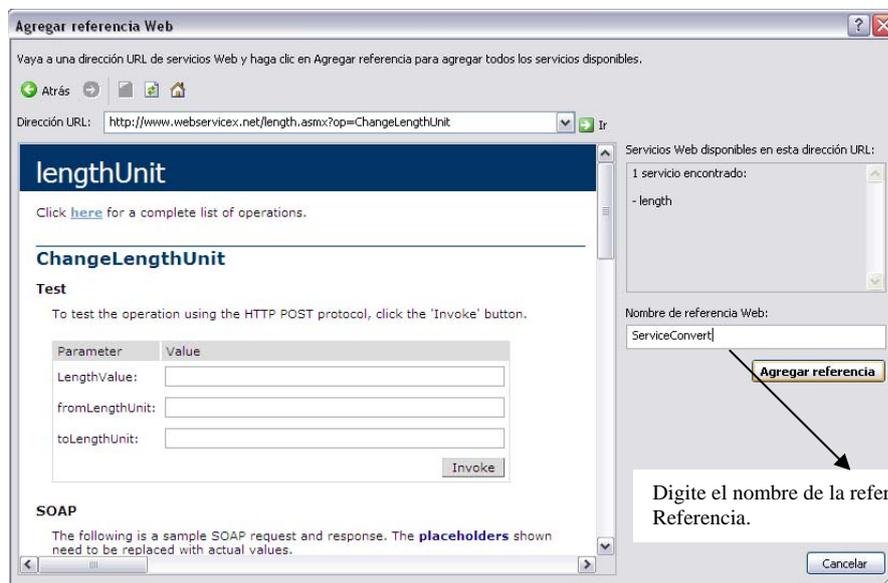


Figura 3 – Creación de la referencia a un servicio web desde una aplicación .Net

- c. La referencia Web ha quedado agregada, debe estar visible en el explorador de soluciones bajo el nodo Web references.

- d. La referencia Web le permitirá tener a su disposición una clase que representa al servicio Web que se va a consumir. Esta clase permite la creación de una instancia para a través de ella acceder al método del servicio Web que proporciona conversión entre las unidades de medida. A continuación se presenta un ejemplo del código que le permitirá consumir el método del servicio Web. ¿Cuál es la clase de la solución en la que debería ir el método que permite realizar la conversión de medidas a través del servicio Web?

```
ServiceConvert.lengthUnit servicio = new PruebaDelegados.ServiceConvert.lengthUnit();  
double num = servicio.ChangeLengthUnit(100, ServiceConvert.Lengths.Centimeters,  
ServiceConvert.Lengths.Meters);
```