

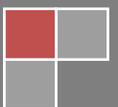
# Pirámide de seguridad

Reporte final de prácticas profesionales.

En este documento se mencionan las actividades que se realizaron en la empresa Ford Motor Company Planta de estampado y ensamble de Hermosillo.



Elder Calderon Lagarda  
Universidad de Sonora  
Ingeniería en sistemas de información



## Índice

Introducción.....	3
Descripción del área de la institución en la que desarrolló la práctica.....	3
Justificación del proyecto realizado.....	4
Objetivos del proyecto.....	4
Problemas planteados para resolverlos.....	4
Alcances y limitaciones en la solución de los problemas.....	4
Fundamento teórico de las herramientas y conocimientos aplicados.....	5
Procedimientos empleados y actividades desarrolladas.....	12
Resultados obtenidos.....	30
Conclusiones y recomendaciones.....	30
Retroalimentación.....	31
Referencias bibliográficas y virtuales.....	32
Anexos.....	33

## **Introducción**

El reporte técnico trata sobre las prácticas profesionales realizadas en la empresa Ford Motor Company.

Las actividades realizadas fueron el desarrollo de la base de datos, diseño y programación de la página web “pirámide de seguridad”, soporte en SharePoint a los empleados de Ford y análisis de las bases de datos que afectaban a la página web.

## **Descripción del área de la institución en la que desarrolló la práctica**

Ford Motor Company es una de las compañías manufactureras más importantes del mundo, tanto en capital, como en personal y capacidad productiva.

Armado con grandes ideas y un pequeño capital, un hombre llamado Henry Ford se decidió a poner el automóvil al alcance de todos.

Se ubica en el km 4.5 carretera a la Colorada, el complejo fue inaugurado en Noviembre de 1986, la planta está situada sobre un terreno de 112.9 hectáreas dentro del Parque Industrial, se producen los modelos Ford Fusion y Lincoln MKZ, así como sus versiones híbridas.

Al iniciar sus operaciones, Ford Hermosillo contaba con alrededor de 1200 empleados que trabajaban en un turno de producción, fabricando en promedio 270 unidades diarias.

A través de la expansión que sufrió la planta, a partir del verano 2005 se tiene una capacidad máxima anual de producción de 305,000 unidades.

La planta produce para todo el continente americano, las operaciones fundamentales van desde el estampado y construcción de carrocerías, pintura de los vehículos, ensamblado y pre entrega.

Las áreas de soporte con las que cuenta Planeación y Logística de Materiales, Contraloría, Recursos Humanos, Lanzamiento, Planeación de Manufactura e IT.

Su misión es “ser líder mundial en productos y servicios automotrices, y financieros. Nuestra Misión es mejorar continuamente nuestros productos y servicios a fin de satisfacer las necesidades de nuestros clientes, lo que nos permite prosperar como negocio y proporcionar utilidades razonables a nuestros accionistas quienes son propietarios de nuestro negocio”.

Su visión es “una buena compañía ofrece excelentes productos y servicios, una gran empresa además, se preocupa por hacer nuestro mundo un mejor lugar donde vivir”.

### **Justificación del proyecto realizado**

Se necesitaba una pagina web donde los empleados pudieran levantar reportes acerca de posibles riesgos de accidentes en la empresa, para llevar un control de los accidentes incapacitantes, accidentes no incapacitantes, daños materiales, incidentes sin perdidas y comportamiento inseguro de los empleados.

Con esta página se les dio a los empleados una herramienta fácil y rápida para levantar reportes, para en un futuro implementar nuevas acciones para evitar dichos accidentes.

### **Objetivos del proyecto**

Presentar las páginas web de la Pirámide de Seguridad con una imagen moderna y atractiva para el usuario, en la cual se puedan llevar las estadísticas de accidentes y para un mejor control de ellas.

### **Problemas planteados para resolverlos**

Mediante las páginas web se puede llevar un control de mejor calidad acerca de los accidentes dentro de la empresa, por esto mismo es necesario desarrollar varias paginas en las cuales el usuario pueda escribir los reportes y también otra para poder ver de una manera rápida las estadísticas de los accidentes.

## **Alcances y limitaciones en la solución de los problemas**

Para la realización de las paginas web y bases de datos, se utilizó Microsoft Visual Studio 2010 y SQL Server 2008 R2.

El departamento no contaba con un programa para modificar o diseñar imágenes, por lo que se tuvo que descargar de forma gratuita PhotoshopCS5 y NotePad++.

Otra limitación fue que al instalar utilizó Microsoft Visual Studio 2010 y SQL Server 2008 R2 no se contaba con los privilegios de administrador, por tal motivo se recurrió a personas con cuentas full Control, por este motivo se perdió mucho tiempo.

## **Fundamento teórico de las herramientas y conocimientos aplicados**

### **Pagina web y sitio web**

Una página web es el nombre de un documento o información electrónica adaptada para la *World Wide Web* y que puede ser accedida mediante un navegador . Esta información se encuentra generalmente en formato HTML o XHTML, y puede proporcionar navegación a otras páginas web mediante enlaces de hipertexto. Las páginas web frecuentemente incluyen otros recursos como hojas de estilo en cascada, guiones (*scripts*) e imágenes digitales, entre otros.

Las páginas web pueden estar almacenadas en un equipo local o un servidor web remoto. El servidor web puede restringir el acceso únicamente para redes privadas, p. ej., en una intranet corporativa, o puede publicar las páginas en la World Wide Web. El acceso a las páginas web es realizado mediante su transferencia desde servidores utilizando el protocolo de transferencia de hipertexto (HTTP).

Una página web está compuesta principalmente por información (sólo texto y/o módulos multimedia) así como por hiperenlaces; además puede contener o asociar Hoja de estilo, datos de estilo para especificar cómo debe visualizarse, y también Aplicación informática aplicaciones embebidas para así hacerla interactiva.

Las páginas web son escritas en un lenguaje de marcado que provee la capacidad de manejar e insertar hiperenlaces, generalmente HTML.

El contenido de la página puede ser predeterminado página web estática o generado al momento de visualizarla o solicitarla a un servidor web página web dinámica. Las páginas dinámicas que se generan al momento de la visualización, se especifican a través de algún lenguaje interpretado, generalmente JavaScript, y la aplicación encargada de visualizar el contenido es la que realmente debe generarlo. Las páginas dinámicas que se generan, al ser solicitadas, son creadas por una aplicación en el servidor web que alberga las mismas.

Respecto a la estructura de las páginas web, algunos organismos, en especial el W3C, suelen establecer directivas con la intención de normalizar el diseño, y para así facilitar y simplificar la visualización e interpretación del contenido.

Una página web es en esencia una tarjeta de presentación digital, ya sea para empresas, organizaciones, o personas, así como una tarjeta de presentación de ideas y de informaciones. Así mismo, la nueva tendencia orienta a que las páginas web no sean sólo atractivas para los internautas, sino también optimizadas (preparadas) para los buscadores a través del código fuente. Forzar esta doble función puede, sin embargo, crear conflictos respecto de la calidad del contenido. Si hablamos de posicionamiento web, una página web es la base para optimizar todo un sitio web el cual es un conjunto de páginas web.

Un sitio web es un gran espacio documental organizado que la mayoría de las veces está típicamente dedicado a algún tema particular o propósito específico. Cualquier sitio web puede contener hiperenlaces a cualquier otro sitio web, de manera que la distinción entre sitios individuales, percibido por el usuario, puede ser a veces borrosa.

No debemos confundir sitio web con página web; esta última es sólo un archivo HTML, una unidad HTML, que forma parte de algún sitio web. Al ingresar una dirección web, como por ejemplo [www.wikipedia.org](http://www.wikipedia.org), siempre se está haciendo referencia a un sitio web, el que tiene una página HTML inicial, que es generalmente la primera que se visualiza.

La búsqueda en Internet se realiza asociando el DNS ingresado con la dirección IP del servidor que contiene el sitio web en el cual está la página HTML buscada.

Los sitios web están escritos en código HTML (*Hyper Text Markup Language*), o dinámicamente convertidos a éste, y se acceden aplicando un software conveniente llamado navegador web, también conocido como un cliente HTTP. Los sitios web pueden ser visualizados o accedidos desde un amplio abanico de dispositivos con conexión a Internet, como computadoras personales, portátiles, PDAs, y teléfonos móviles.

Un sitio web está alojado en una computadora conocida como servidor web, también llamada servidor HTTP, y estos términos también pueden referirse al software que se ejecuta en esta computadora y que recupera y entrega las páginas de un sitio web en respuesta a peticiones del usuario. Apache es el programa más comúnmente usado como servidor web (según las estadísticas de Netcraft) y el Internet Information Services (IIS) de Microsoft también se usa con mucha frecuencia.

Un sitio web estático es uno que tiene contenido que no se espera que cambie frecuentemente y se mantiene manualmente por alguna persona o personas que usan algún tipo de programa editor.

Hay dos amplias categorías de programas editores usados para este propósito que son

- Editores de texto como Notepad, donde el HTML se manipula directamente en el programa editor o
- Editores WYSIWYG como por ejemplo Microsoft FrontPage y Adobe Dreamweaver, donde el sitio se edita usando una interfaz GUI y el HTML subyacente se genera automáticamente con el programa editor.

Hay plugins disponibles para navegadores, que se usan para mostrar contenido activo como Flash, Silverlight, Shockwave o applets, escritos en Java. El HTML dinámico también proporciona interactividad para los usuarios, y el elemento de actualización en tiempo real entre páginas web (i.e, las páginas no tienen que cargarse o recargarse para efectuar cualquier cambio), principalmente usando el DOM y JavaScript, el soporte de los cuales está integrado en la mayoría de navegadores web modernos.

## Tipos de sitios web

Existen muchas variedades de sitios web, cada uno especializándose en un tipo particular de contenido o uso, y ellos pueden ser arbitrariamente clasificados de muchas maneras. Unas pocas clasificaciones pueden incluir:

- Sitio archivo: usado para preservar contenido electrónico valioso amenazado con extinción. Dos ejemplos son: Internet Archive, el cual desde 1996 ha preservado billones de antiguas (y nuevas) páginas web; y Google Groups, que a principios de 2005 archivaba más de 845.000.000 mensajes expuestos en los grupos de noticias/discusión de Usenet, tras su adquisición de Deja News.
- Sitio weblog (o blog o bitácora digital): sitio usado para registrar lecturas online o para exponer contenidos en línea con la fecha del día de ingreso; también puede incluir foros de discusión. Ejemplos: Blogger, LiveJournal, WordPress.
- Sitio de empresa: usado para promocionar una empresa o servicio.
- Sitio de comercio electrónico: para comprar bienes, como Amazon.com.
- Sitio de comunidad virtual: un sitio o portal social donde las personas con intereses similares se comunican unos con otros, normalmente por chat o foros o simples mensajes. Por ejemplo: MySpace, Facebook, Hi5, Orkut, Habbo, Multiply, Quepasa.
- Sitio de Base de datos: un sitio donde el uso principal es la búsqueda y muestra de un contenido específico de la base de datos, como por ejemplo Internet Movie Database.
- Sitio de desarrollo: un sitio con el propósito de proporcionar información y recursos relacionados con el desarrollo de software, diseño web, etc.
- Sitio directorio: un sitio que contiene contenidos variados que están divididos en categorías y subcategorías, como el directorio de Yahoo!, el directorio de Google, y el Open Directory Project.
- Sitio de descargas: estrictamente usado para descargar contenido electrónico, como software, juegos o fondos de escritorio: Download, Tucows, Softonic, Baulsoft.
- Sitio de juego: un sitio que es propiamente un juego o un «patio de recreo» donde mucha gente viene a jugar, como MSN Games, Minijuegos.com, Pogo.com y los MMORPGs VidaJurásica, Planetarion y Kings of Chaos.
- Sitio de información: contiene contenido que pretende informar a los visitantes, pero no necesariamente de propósitos comerciales; tales como: Free Internet Lexicon y Encyclopedia. La mayoría de los gobiernos e instituciones educacionales y sin ánimo de lucro tienen un sitio de información.
- Sitio de noticias: Similar a un sitio de información, pero dedicada a mostrar noticias y comentarios de la actualidad.

## **Control GridView**

El control GridView del ASP.Net permite visualizar datos en una tabla en pantalla, editar, modificar y borrar registros del mismo.

El GridView es un control extremadamente flexible para mostrar tablas multicolumna.

Cada registro de una consulta del select configurado en un SqlDataSource genera una fila. Cada campo en el registro representa una columna.

El GridView es el control más poderoso que provee el ASP.Net. Este control trae funcionalidades ya implementadas para paginación, ordenamiento y edición de sus datos.

## **Hojas de estilo**

El concepto de hojas de estilo apareció por primera vez en 1996 cuando W3C publicó una recomendación nueva intitulada "Hojas de estilo en cascada" o CSS, su sigla en inglés.

El principio de las hojas de estilo consiste en la utilización de un solo documento para almacenar las características de presentación de las páginas asociadas a grupos de elementos.

Esto implica nombrar un conjunto de definiciones y características de presentación de las páginas, y activar esos nombres para aplicarlos a una parte del texto. Por ejemplo, se pueden configurar los títulos de una sección para que aparezcan en fuente Arial, en color verde y en cursiva.

Las hojas de estilo se desarrollaron para compensar los defectos de HTML con respecto a la presentación y al diseño de las páginas. HTML tiene varias etiquetas para modificar la presentación y definir los estilos del texto, pero cada elemento tiene su propio estilo, independientemente de los elementos que lo rodean.

Al utilizar hojas de estilo, cuando se necesite cambiar la apariencia de un sitio que tiene cientos de páginas Web todo lo que hay que hacer es editar las definiciones de la hoja de estilo en un solo lugar para cambiar la apariencia del sitio completo. Se denominan "hojas de estilo en cascada" porque se pueden definir múltiples hojas y los estilos pueden aplicarse a todas las páginas (con un sistema predefinido para resolver conflictos).

Las hojas de estilo pueden utilizarse para:

- lograr una apariencia uniforme de todo el sitio al activar una sola definición de estilo en cada página.
- cambiar un aspecto en todo el sitio Web con tan sólo editar unas pocas líneas.
- hacer que los códigos HTML sean más fáciles de leer ya que los estilos se definen por separado.
- permitir que las páginas se carguen más rápido ya que hay menos cantidad de HTML en cada página.

### Existen tres tipos de estilos

1. **Una hoja de estilo externa**, es una hoja de estilo que está almacenada en un archivo diferente al archivo donde se almacena el código HTML de la página Web. Esta es la manera de programar más potente, porque separa completamente las reglas de formateo para la página HTML de la estructura básica de la página:
2. **Una hoja de estilo interna**, es una hoja de estilo que está incrustada dentro de un documento HTML. (Va a la derecha dentro del elemento <head>.) De esta manera se obtiene el beneficio de separar la información del estilo del código HTML propiamente dicho. Se puede optar por copiar la hoja de estilo incrustada de una página a otra (esta posibilidad es difícil de ejecutar si se desea para guardar las copias sincronizadas). En general, la única vez que se usa una hoja de estilo interna, es cuando se quiere proporcionar alguna característica a una página Web en un simple fichero, por ejemplo, si se está enviando algo a la página Web.
3. **Un estilo en línea (inline)** es un método para insertar el lenguaje de estilo de página directamente dentro de una etiqueta HTML. Esta manera de proceder no es totalmente adecuada. El incrustar la descripción del formateo dentro del documento de la página Web, a nivel de código, se convierte en una manera larga, tediosa y poco elegante de resolver el problema de la programación de la página. Este modo de trabajo se podría usar de manera ocasional si se pretende aplicar un formateo con prisa, al vuelo.

## **Limitaciones al usar hojas de estilo**

Algunas limitaciones que se encuentran en el uso del CSS hasta la versión CSS2.1, vigente, pueden ser:

- Los selectores no pueden usarse en orden ascendente según la jerarquía del DOM (hacia padres u otros ancestros) como se hace mediante XPath. La razón que se ha usado para justificar esta carencia por parte de la W3C, es para proteger el rendimiento del navegador, que de otra manera, podría verse comprometido. XSLT soporta en la actualidad un mayor número de sistemas operativos. Así mismo, también es mejor para trabajar con la mayoría de buscadores de Internet.
- Dificultad para el alineamiento vertical; así como el centrado horizontal se hace de manera evidente en CSS2.1, el centrado vertical requiere de diferentes reglas en combinaciones no evidentes, o no estándares.
- Ausencia de expresiones de cálculo numérico para especificar valores.
- Las pseudo-clases dinámicas (como: hover) no se pueden controlar o deshabilitar desde el navegador, lo que las hace susceptibles de abuso por parte de los diseñadores en banners, o ventana emergentes.

## **Ventajas al usar hojas de estilo**

Las ventajas de utilizar CSS (u otro lenguaje de estilo) son:

- Control centralizado de la presentación de un sitio web completo con lo que se agiliza de forma considerable la actualización del mismo.
- Los navegadores permiten a los usuarios especificar su propia hoja de estilo local, que será aplicada a un sitio web, con lo que aumenta considerablemente la accesibilidad. Por ejemplo, personas con deficiencias visuales pueden configurar su propia hoja de estilo para aumentar el tamaño del texto o remarcar más los enlaces.
- Una página puede disponer de diferentes hojas de estilo según el dispositivo que la muestre o, incluso, a elección del usuario. Por ejemplo, para ser impresa, mostrada en un dispositivo móvil o ser "leída" por un sintetizador de voz.
- El documento HTML en sí mismo es más claro de entender y se consigue reducir considerablemente su tamaño (siempre y cuando no se utilice estilo en línea).

## **JQuery**

JQuery es una biblioteca de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a paginas web.

JQuery es software libre y de código abierto, posee un doble licenciamiento bajo la licencia MIT y la licencia publica general de GNU, permitiendo sus usos en proyectos libres y privativos.

## **Lenguaje asp.net**

ASP.NET es un framework para aplicaciones web desarrollado y comercializado por Microsoft. Es usado por programadores para construir sitios web dinámicos, aplicaciones web y servicios web XML. Apareció en Enero de 2002 con la versión 1.0 del .NET Framework, y es la tecnología sucesora de la tecnología Active Server Pages, está construido sobre Common Lenguaje Runtime, permitiendo a los programadores escribir código ASP.NET usando cualquier lenguaje admitido por el .NET Framework.

Las páginas de ASP.NET, conocidas oficialmente como web forms (formularios web), son el principal medio de construcción para el desarrollo de aplicaciones web. Los formularios web están contenidos en archivos con una extensión aspx, en jerga de programación, estos archivos contienen etiquetas html o xhtml, y también etiquetas definiendo controles web que se procesan del lado del servidor y controles de usuario donde los desarrolladores colocan todo el código estático y dinámico requerido por la página web. Adicionalmente, el código dinámico que se ejecuta en el servidor puede ser colocado en una página dentro de un bloque `<%-- código dinamico--%>` que es muy similar a otras tecnologías de desarrollo como php, jsp y asp, pero esta práctica es, generalmente, desaconsejada excepto para propósitos de enlace de datos pues requiere más llamadas cuando se genera la página.

## Procedimientos empleados y actividades desarrolladas

La realización de la página web, se debe a que se necesitaba un control de los accidentes dentro de la planta, que los empleados de Ford puedan levantar un reporte fácilmente, con solo entrar a la página web y escribir los detalles del accidente. Con esto se toman medidas contra estos casos para evitar futuros accidentes.

Esta es la página principal:

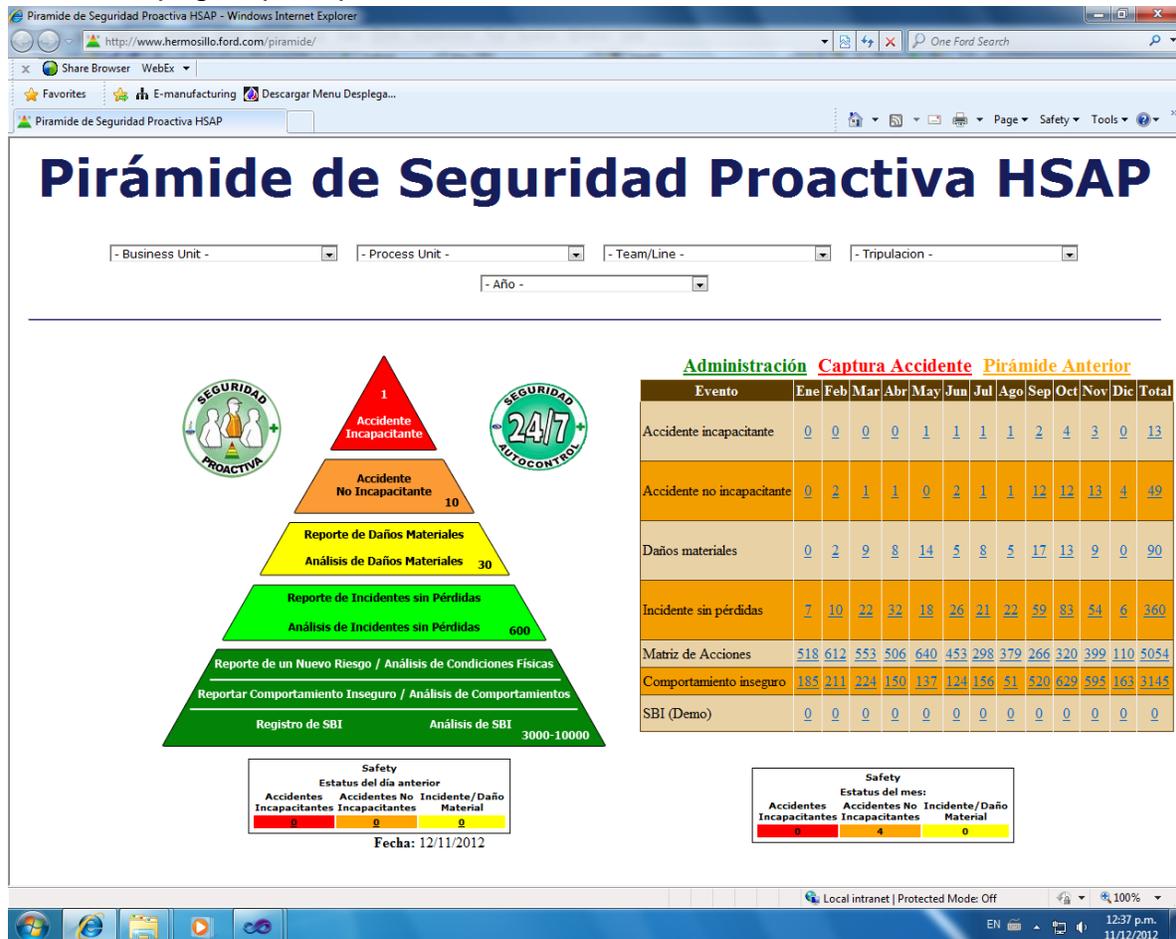


Fig.1 Página principal del módulo reportes

La página web se divide en varias partes:

Primero en una carpeta nueva, a la que llame administración, puse los webforms, en el cual llevan el código .aspx.

Utilice la herramienta ToolKitScriptManager para utilizar las extensiones de Ajax.

Dividí la página en paneles, tiene herramientas muy eficaces para actualizar la página después de una acción.

Para mostrar los datos que los encargados de cada área agregaban, utilice un GridView, una potente herramienta de Asp.net para datos.

Los únicos que podían subir información a la página eran los supervisores o encargados de cada área, como por ejemplo área de pinturas, área de carrocería, etc. Ellos tenían un acceso a la página con su usuario y contraseña. Pero todos los empleados podían informar a sus supervisores de personas que no cumplían con las reglas o un posible accidente, por ejemplo no llevar lentes de seguridad en áreas peligrosas.

Cada usuario tenía que poner cierta información, por ejemplo de que área era, cuál era el problema de la situación, si el empleado llevaba su ropa de seguridad, si usaba el celular en áreas restringidas, etc.

Yo hacía recorridos por la planta, para señalar posibles accidentes y ver en qué situaciones los empleados ignoraban las reglas.

Se dividió en campos, después de hacer un análisis de los problemas que ocurrían en la planta de ensamblado.

Al entrar a la planta se puso en la pared la imagen del gridview con los accidentes del mes, la razón de esto, era de que todos los empleados de Ford y proveedores, vieran el número de accidentes del mes y del año, para que tomaran conciencia de que se tenía que actuar con mucho cuidado y concentración dentro de la planta.

También dentro del comedor, en las televisiones se pasaba esta tabla, con información de cómo evitar estos posibles accidentes, la idea de esto era reducir estos casos.

```

</head>
<body>
  <form id="form1" runat="server">
    <div>
      <asp:ToolkitScriptManager ID="ToolkitScriptManager1" runat="server">
      </asp:ToolkitScriptManager>
      <asp:Panel ID="pnlPrincipal" runat="server" Height="500px">
        <asp:Panel ID="pnl1" runat="server" class="pnl1" Height="250px">
          <asp:GridView ID="gvPrincipal" runat="server"
AutoGenerateColumns="False"
          onrowdeleted="gvAlum_RowDeleted"
onrowdeleting="gvAlum_RowDeleting"
          onrowcancelingedit="gvAlum_RowCancelingEdit"
onrowediting="gvAlum_RowEditing"
          onrowupdated="gvAlum_RowUpdated"
onrowupdating="gvAlum_RowUpdating"
          BackColor="LightGoldenrodYellow" BorderColor="Tan"
BorderWidth="1px"
          CellPadding="2" ForeColor="Black" GridLines="None">
          <AlternatingRowStyle BackColor="PaleGoldenrod" />
          <Columns>
            <asp:CommandField CancelText="Cancelar" EditText="Editar"
ShowEditButton="True"
              UpdateText="Actualizar" />
            <asp:TemplateField HeaderText="ID" Visible="False">
              <ItemTemplate>
                <asp:Label ID="lblItemID" runat="server" Text='<## Bind ("id")
%>'></asp:Label>
              </ItemTemplate>
              <EditItemTemplate>
                <asp:Label ID="lblEditItemID" runat="server" Text='<##
Bind("id") %>'></asp:Label>
              </EditItemTemplate>
            </asp:TemplateField>
            <asp:TemplateField HeaderText="Accidente incapacitante">
              <ItemTemplate>
                <asp:Label ID="lblItemAccidIncapacitante" runat="server"
Text='<## Bind ("accidIncapacitante") %>'></asp:Label>
              </ItemTemplate>
              <EditItemTemplate>
                <asp:Label ID="lblEditItemAccidIncapacitante" runat="server"
Text='<## Bind("accidIncapacitante") %>'></asp:Label>
              </EditItemTemplate>
            </asp:TemplateField>
            <asp:TemplateField HeaderText="Accidente no incapacitante">
              <ItemTemplate>

```

```

        <asp:Label ID="lblItemAccidNoIncapacitante" runat="server"
Text='<%=# Bind("accidNoIncapacitante") %>'></asp:Label>
    </ItemTemplate>
    <EditItemTemplate>
        <asp:Label ID="lblEditItemAccidNoIncapacitante" runat="server"
Text='<%=# Bind("accidNoIncapacitante") %>'></asp:Label>
    </EditItemTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="daños Materiales">
    <ItemTemplate>
        <asp:Label ID="lblItemdañosMateriales" runat="server"
Text='<%=# Bind("dañosMateriales") %>'></asp:Label>
    </ItemTemplate>
    <EditItemTemplate>
        <asp:Label ID="lblEditItemdañosMateriales" runat="server"
Text='<%=# Bind("dañosMateriales") %>'></asp:Label>
    </EditItemTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="Incidente sin perdidas">
    <ItemTemplate>
        <asp:Label ID="lblItemIncidSinPerdidas" runat="server"
Text='<%=# Bind("IncidSinPerdidas") %>'></asp:Label>
    </ItemTemplate>
    <EditItemTemplate>
        <asp:Label ID="lblEditItemIncidSinPerdidas" runat="server"
Text='<%=# Bind("IncidSinPerdidas") %>'></asp:Label>
    </EditItemTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="Matriz de acciones">
    <ItemTemplate>
        <asp:Label ID="lblItemMatrizAcciones" runat="server" Text='<%=#
Bind("MatrizAcciones") %>'></asp:Label>
    </ItemTemplate>
    <EditItemTemplate>
        <asp:Label ID="lblEditItemMatrizAcciones" runat="server"
Text='<%=# Bind("MatrizAcciones") %>'></asp:Label>
    </EditItemTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="Comportamiento inseguro">
    <ItemTemplate>
        <asp:Label ID="lblItemCompInseguro" runat="server" Text='<%=#
Bind("CompInseguro") %>'></asp:Label>
    </ItemTemplate>
    <EditItemTemplate>
        <asp:Label ID="lblEditItemCompInseguro" runat="server"
Text='<%=# Bind("CompInseguro") %>'></asp:Label>
    </EditItemTemplate>

```

```

        </asp:TemplateField>
        <asp:TemplateField HeaderText="SBI (Demo)">
        <ItemTemplate>
            <asp:Label ID="lblItemSbi" runat="server" Text='<%=# Bind("Sbi")
%>'></asp:Label>
        </ItemTemplate>
        <EditItemTemplate>
            <asp:Label ID="lblEditItemSbi" runat="server" Text='<%=#
Bind("Sbi") %>'></asp:Label>
        </EditItemTemplate>
        </asp:TemplateField>
        <asp:CommandField DeleteText="Eliminar"
ShowDeleteButton="True" />
    </Columns>
    <FooterStyle BackColor="Tan" />
    <HeaderStyle BackColor="Tan" Font-Bold="True" />
    <PagerStyle BackColor="PaleGoldenrod" ForeColor="DarkSlateBlue"
HorizontalAlign="Center" />
    <SelectedRowStyle BackColor="DarkSlateBlue"
ForeColor="GhostWhite" />
    <SortedAscendingCellStyle BackColor="#FAFAE7" />
    <SortedAscendingHeaderStyle BackColor="#DAC09E" />
    <SortedDescendingCellStyle BackColor="#E1DB9C" />
    <SortedDescendingHeaderStyle BackColor="#C2A47B" />
</asp:GridView>

```

Los datos del GridView venían de una base de datos, las cuales los mismos empleados llenaban.

Se había puesto un campo de edición, esto solo para el administrador, se podía cambiar y borrar un reporte, el cual estaba visible, pero después se quitó.

El mismo GridView tiene una propiedad para cambiar el color de las celdas, por default viene blanco, por ejemplo se puede cambiar a multicolor.

Para cambiar el color se va a la pestaña, de formato automático y después se elige el color que se desee.

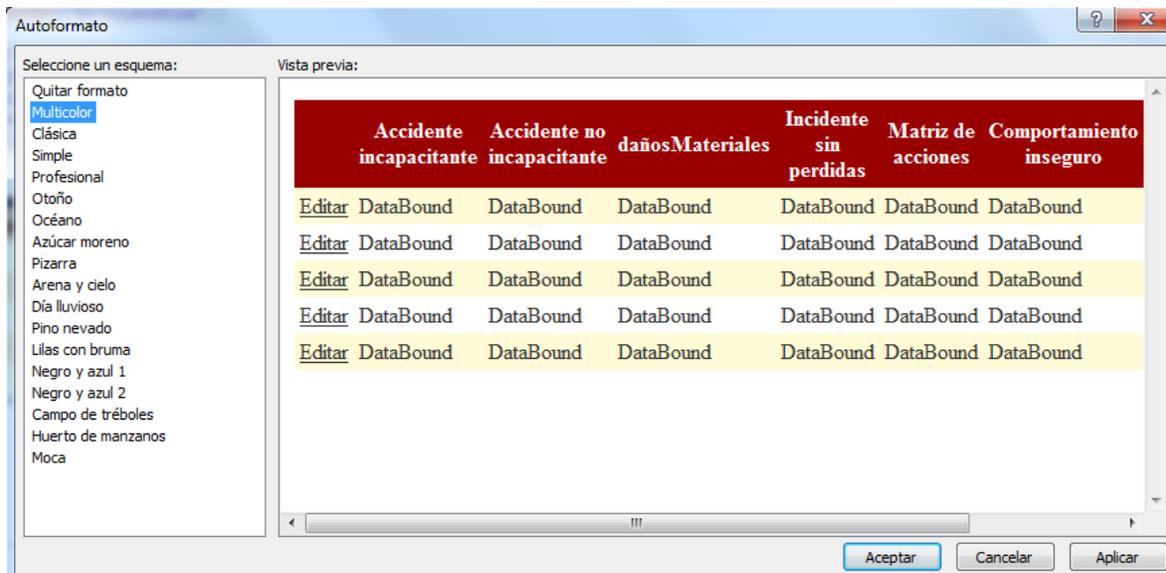


Fig. 2 Cambiar color a un GridView

En el code-behind se llama igual que en el webforms por default.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
```

```
public partial class administrador_Alumnos : System.Web.UI.Page
{
    AccidIncapacitante oAccidIncapacitante;
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!Page.IsPostBack)
        {
            LlenarGV();
        }
    }
}
```

```
public void LlenarGV()
{
    oAccidIncapacitante = new AccidIncapacitante ();
    gvAlum.DataSource = oAccidIncapacitante.Get();
    gvAlum.DataBind();
}
```

Lo primero es llamar a la clase poniendo su nombre y después instanciarla.

Al principio del code-behind utilice un if para realizar el postback de la página, esto me sirve para saber cuándo se inicia por primera vez en el sitio web.

Dentro del mismoPostBack se puso el método el cual llamaba al GridView, para que al iniciar la página mostrara los datos de la tabla.

Para poder realizar la ejecución del GridView instancie la clase, para después mandar llamar la base de datos con el método Get, después se utilizó un DataBind para la información.

Se formó una clase con el mismo nombre del webforms, para evitar confusiones, es muy importante tener un orden al nombrar clases, métodos y variables, ya que esto te puede llevar a confusión y errores, también mucha pérdida de tiempo. Yo utilizaba la sintaxis del camello, la cual si son dos nombres, en el segundo se aplica la primer letra mayúscula, ejemplo mQuery.

Para llamar a la base de datos utilizando la clase BDman y después mostrar el contenido del GridView.

Para poder utilizar las bases de datos, en el comienzo del código agregamos using System.Data y using System.Data.SqlClient.

Para mostrar el contenido en el GridView utilice un DataTable el cual llame Get.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Data;
using System.Data.SqlClient;

/// <summary>
/// Summary description for ComportInseguro
/// </summary>
public class ComportInseguro
{
    BDman oBDman;
    public ComportInseguro ()
    {
        //
        // TODO: Add constructor logic here
        //
    }
}
```

```

public DataTable Get()
{
    try
    {
        oBDman = new BDman();
        string mQuery;
        DataTable mTabla = new DataTable();
        mQuery = "SELECT id, businnesUnit, processUnit, teamLine, tripulacion,
descripcion, comportamiento, reglasProced, equipoProtecPersonal,
procesoCritico, Retroalimentacion FROM cat ComportInseguro ";
        mTabla = oBDman.Consultar(mQuery);
        return mTabla;
    }
    catch(Exception Errores)
    {
        return null;
    }
}

```

Utilice un Select para que me muestre la información de un campo de la base de datos.

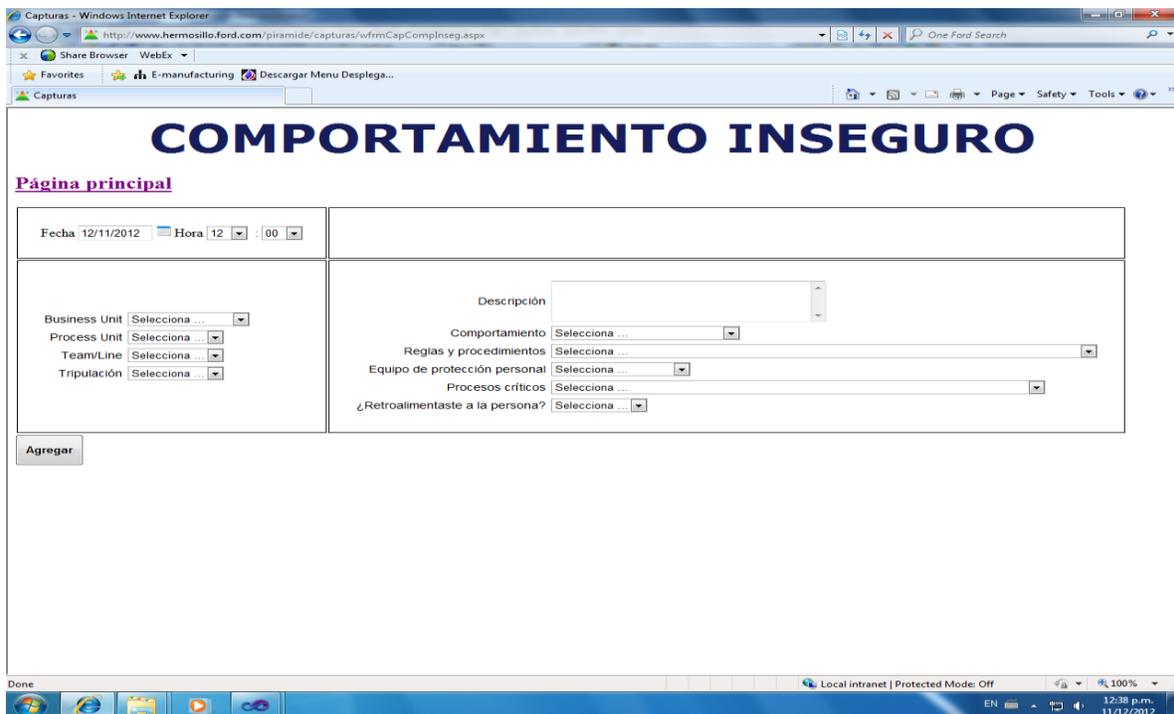


Fig.3 Pagina Comportamiento inseguro

Para poner el calendario se utilizó la herramienta de asp.net "calendar".  
Se utilizó un textbox.

```

<td class="tdFecha">
    &nbsp;<asp:TextBox ID="txtFechaRep" runat="server" Width="100px"
        ontextchanged="txtFechaRep_TextChanged" Font-Bold="False"
        ForeColor="Black"></asp:TextBox>
        &nbsp;<asp:Button class="btnFechaRep" ID="btnFechaRep"
runat="server"
        onclick="btnFechaRep_Click" Width="25px" Height="22px" />
    </td>
    <td class="tdCalendario">
        &nbsp;<asp:Calendar ID="CalendarRep" runat="server" BackColor="White"
            BorderColor="Black" EnableTheming="True"
            Font-Names="Times New Roman" Font-Size="6pt" ForeColor="Black"
Height="90px"
            NextPrevFormat="FullMonth" Width="120px" Visible="False"
            onselectionchanged="CalendarRep_SelectionChanged"
DayNameFormat="Shortest"
            TitleFormat="Month">
                <DayHeaderStyle Font-Bold="False" Font-Size="7pt"
ForeColor="#333333"
                    Height="10pt" BackColor="#84C2F5" />
                <DayStyle Width="14%" />
                <NextPrevStyle Font-Size="8pt" ForeColor="White" />
                <OtherMonthDayStyle ForeColor="#999999" />
                <SelectedDayStyle BackColor="#84C2F5" ForeColor="White" />
                <SelectorStyle BackColor="#CCCCCC" Font-Bold="True" Font-
Names="Verdana"
                    Font-Size="8pt" ForeColor="#333333" Width="1%" />
                <TitleStyle BackColor="#068CFA" Font-Bold="True"
                    Font-Size="13pt" ForeColor="White" Height="14pt" />
                <TodayDayStyle BackColor="#84C2F5" />
            </asp:Calendar>
    </td>

```

Dentro de la herramienta calendar, se puede modificar color, estilos, fuente, y tamaños de letras.

El empleado para llenar el formulario se le aparecían varias opciones, en otros casos tenía que llenarlo manualmente, se utilizó la herramienta DropDownList y Texbox.

Se utilizó un método Boolean para agregar la información del formulario a la base de datos sql server 2008 R2.

```
public Boolean Add(string businnesUnit, string processUnit, string teamLine, string
tripulacion, string descripcion, string comportamiento, string reglasProced, string
equipoProtecPersonal, string procesoCritico, string Retroalimentacion )
{
    try
    {
        oBDman = new BDman();
        string mQuery;
        mQuery = "INSERT INTO ComportInseguro (businnesUnit, processUnit,
teamLine, tripulacion, descripcion, comportamiento, reglasProced,
equipoProtecPersonal, procesoCritico, Retroalimentacion) VALUES ("+
businnesUnit +",""+ processUnit +",""+ teamLine +",""+ tripulacion +",""+
descripcion +",""+ comportamiento +",""+ reglasProced +",""+
equipoProtecPersonal +",""+ procesoCritico +",""+ Retroalimentacion +")";
        if(oBDman.ABC(mQuery))
        {
            return true;
        }
        else
        {
            return false;
        }
    }
    catch(Exception Errores)
    {
        return false;
    }
}
```

Con la sentencia Insert into, se agregó la información de cada campo a la base de datos. Se utilizó las sentencias de try and catch, para detectar posibles fallos. Esta sentencia es muy útil en asp.net, ya que en caso de fallas, es más fácil deducir donde está el error, la causa del problema.

En un GridView nos mostraba toda la información que el usuario había escrito.

**REPORTE: EVENTOS**

[REGRESAR](#)

Imprimir	Evento	Fecha Evento	Business Unit	Process Unit	Team / Line	Tripulacion	Pcon	Asesor Proceso	Estacion Columna	Descripcion	FTOV	DART	LTC	OAR	SII	FIRE	HPI	Comportamiento	Otro Comportamiento
<a href="#">Enviar</a>	Accidente incapacitante	11/16/2012 7:40:00 AM	MP&L	PU2	Sop/Logistica Externa	Logistica Externa	M2AA	Mirla Olivares	Exterior junto a Patio 22	Quando se encontraba descargando un contenedor de racks, un empleado estaba manejando un montacargas, se omitió el asegurar el contenedor utilizando el gancho de seguridad. Se adelanto a la operacion no dejando que el chofer asegurar el contenedor. Al momento de intentar ingresar al contenedor en el montacargas, se movio esta creando un espacio entre la rampa y el contenedor. Causando que el montacargas cayera de forma vertical.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Dejar inoperables los dispositivos de Seguridad	
	Accidente incapacitante	11/8/2012 1:15:00 PM	Body	PU0	WG C Dimensional	Tripulacion A	B0AA	Jesus Astrain	CMM B0	El operador de CMM sufrió una herida cortante en la mano derecha entre los dedos	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Falla en el Uso Apropiado del Equipo de Protección Personal (EPP)	

Fig.4 GridView información del formulario.

Nos mostraba todos los campos, la información de cada reporte generado, la posibilidad de editar el GridView, mediante los eventos update, delete, edit, cancel.

También se le inserto la paginación, ya que en un solo recuadro no alcanzaban todos los reportes.

Se creó un método para actualizar y otro para eliminar la información, los dos son public boolean.

```
public Boolean Update(int id, string a, string b, string c, int activo)
{
    try
    {
        oBDman = new BDman();
        string mQuery;
```

```

        mQuery = "UPDATE ComportInseguro SET a="+ a +", b = "+ b +", c = "+
c +" WHERE id = "+ id +";
        if(oBDman.ABC(mQuery))
        {
            return true;
        }
        else
        {
            return false;
        }
    }
    catch(Exception Errores)
    {
        return false;
    }
}

```

```

public Boolean Delete(int id)
{
    try
    {
        oBDman = new BDman();
        string mQuery;
        mQuery = "DELETE FROM ComportInseguro WHERE id="+ id +";
        if(oBDman.ABC(mQuery))
        {
            return true;
        }
        else
        {
            return false;
        }
    }
    catch(Exception Errores)
    {
        return false;
    }
}

```

Y después en el evento row\_update y row\_delete, se agregan en el code-behind. Esto se utiliza para poder editar un GridView.

Para poder actualizar un campo o eliminarlo, según sea el caso, esto se va directamente reflejado en la base de datos.

Se utilizó la herramienta textbox para poder actualizar un campo, si no se desea hacer cambios solo basta con poner un Label, así el usuario no podrá mover nada.

```
protected void gv_RowDeleted(object sender, GridViewDeletedEventArgs e)
{
    gv.EditIndex = -1;
    LlenarGV();
}
protected void gv_RowDeleting(object sender, GridViewDeleteEventArgs e)
{
    OAccidIncapacitante = new AccidIncapacitante ();
    Label mID = (Label)gv.Rows[e.RowIndex].Cells[1].FindControl("lblItemID");
    if (O AccidIncapacitante .Delete(int.Parse(mID.Text)))
    {
        Avisos(this, "eliminado");
    }
    else
    {
        Avisos(this, "No eliminado");
    }
}
protected void gv_RowUpdated(object sender, GridViewUpdatedEventArgs e)
{
    gv.EditIndex = -1;
    LlenarGV();
}
protected void gv_RowUpdating(object sender, GridViewUpdateEventArgs e)
{
    OAccidIncapacitante = new AccidIncapacitante ();
    string mA, mAp, mApm;
    TextBox mTxt;
    int mActivo;

    Label mID = (Label)gvAlum.Rows[e.RowIndex].Cells[1].FindControl("lblEditItemID");
    mTxt = (TextBox)gvAlum.Rows[e.RowIndex].FindControl("txtEditItemAccidIncapcitante");
    mA = !string.IsNullOrEmpty(mTxt.Text.Trim()) ? mTxt.Text : "";
```

```

mTxt = (TextBox)gvAlum.Rows[e.RowIndex].FindControl("txtEditItem");
mAp = !string.IsNullOrEmpty(mTxt.Text.Trim()) ? mTxt.Text : "";
mTpm = (TextBox)gvAlum.Rows[e.RowIndex].FindControl("txtEditItem");
mApm = !string.IsNullOrEmpty(mTpm.Text.Trim()) ? mTpm.Text : "";
CheckBox mckbActivo =
(CheckBox)gv.Rows[e.RowIndex].FindControl("ckbEditItemActivo");
mActivo = mckbActivo.Checked ? 1 : 0;
OAccidIncapacitante = new AccidIncapacitante ();

if (string.IsNullOrEmpty(mA))
{
    Avisos(this, "Introduce a");
}
if (string.IsNullOrEmpty(mAp))
{
    Avisos(this, "Introduce el b");
}
if (string.IsNullOrEmpty(mApm))
{
    Avisos(this, "Introduce c");
}
else
{
    if (OAccidIncapacitante .Update(int.Parse(mID.Text), mA, mAp, mApm,
mActivo))
    {
        Avisos(this, " Actualizado");
        LlenarGV();
    }
    else
    {
        Avisos(this, " No actualizado");
    }
}
}
protected void gv_RowCancelingEdit(object sender,
GridViewCancelEventArgs e)
{
    gv.EditIndex = -1;
    LlenarGV();
}
protected void gv_RowEditing(object sender, GridViewEditEventArgs e)
{
}
}

```

Utilice diferentes validaciones para diferentes eventos, por ejemplo para un campo que solo aceptaba números, utilice una validación ayudándome con JavaScript.

Cuando se utiliza una función JavaScript se debe poner en el webforms y después mandarla llamar en el code-behind.

```
<script type="text/javascript">
    function SoloNumeros(e) {
        var tecla = document.all ? tecla = e.keyCode : tecla = e.which;
        return ((tecla > 47 && tecla < 58) || tecla == 46);
    }
</script>
```

```
txtA.Attributes.Add("OnKeyPress", "return SoloNumeros(event)");
```

Se elaboró un método para identificar si un campo ya existía, si estaba repetido.

```
public bool ExistenDatos(string Query)
{
    CerrarConexion();
    cn = new SqlConnection(cnStr.ToString());
    cmd = new SqlCommand();
    cmd.Connection = cn;
    cmd.CommandText = Query;

    try
    {
        cn.Open();
        dr = cmd.ExecuteReader();
        if(dr.Read())
        {
            return true;
        }
        else
        {
            return false;
        }
    }
    catch(SqlException sqlError)
    {
        _Errores = sqlError.Message;
        return false;
    }
}
```

Para verificar que un campo no estuviera vacío, se escribió esta validación, esto en los DropDownList.

```
public string ValidarCampos()
{
    string mMsj = "";
    mMsj += ddl.SelectedIndex == 0 ? "Selecciona. \n\r" : "";
    mMsj += ddlMateriaCalif.SelectedIndex == 0 ? "Selecciona.\n\r" : "";
    mMsj += string.IsNullOrEmpty(txtCalif.Text) ? "Introduce.\n\r" : "";
    return mMsj;
}
```

Y un método para mandar un aviso al usuario.

```
public void Avisos(Control control, string Mensaje)
{
    ScriptManager.RegisterStartupScript(control, this.GetType(), "", "alert('" + Mensaje + "');", true);
}
```

Para realizar la conexión a la base de datos, se realizó una clase para poder hacer esto posible.

```
public class BDman
{
    SqlConnection cn;
    SqlCommand cmd;
    SqlDataAdapter da;
    SqlDataReader dr;
    SqlConnectionStringBuilder cnStr;
    DataTable dt;
    string _Errores;

    public BDman()
    {
        cnStr = new SqlConnectionStringBuilder();
        cnStr.DataSource = "WGC1HMDQ3QVR1";
        cnStr.InitialCatalog = "abc";
        cnStr.UserID = "sa";
        cnStr.Password = "sa2012";
        cnStr.IntegratedSecurity = false;
        cnStr.PersistSecurityInfo = true;
        cnStr.Pooling = true;
        cn = new SqlConnection(cnStr.ToString());
        cmd = new SqlCommand();
    }
}
```

También colabore utilizando el SharePoint, modificando los permisos del departamento de sistemas, esto se realizó debido a que los altos mandos de sistemas ya no iban a tener la responsabilidad de modificar un departamento, se le cambio ese trabajo a otras personas, por lo tanto le quite los permisos de súper usuario.

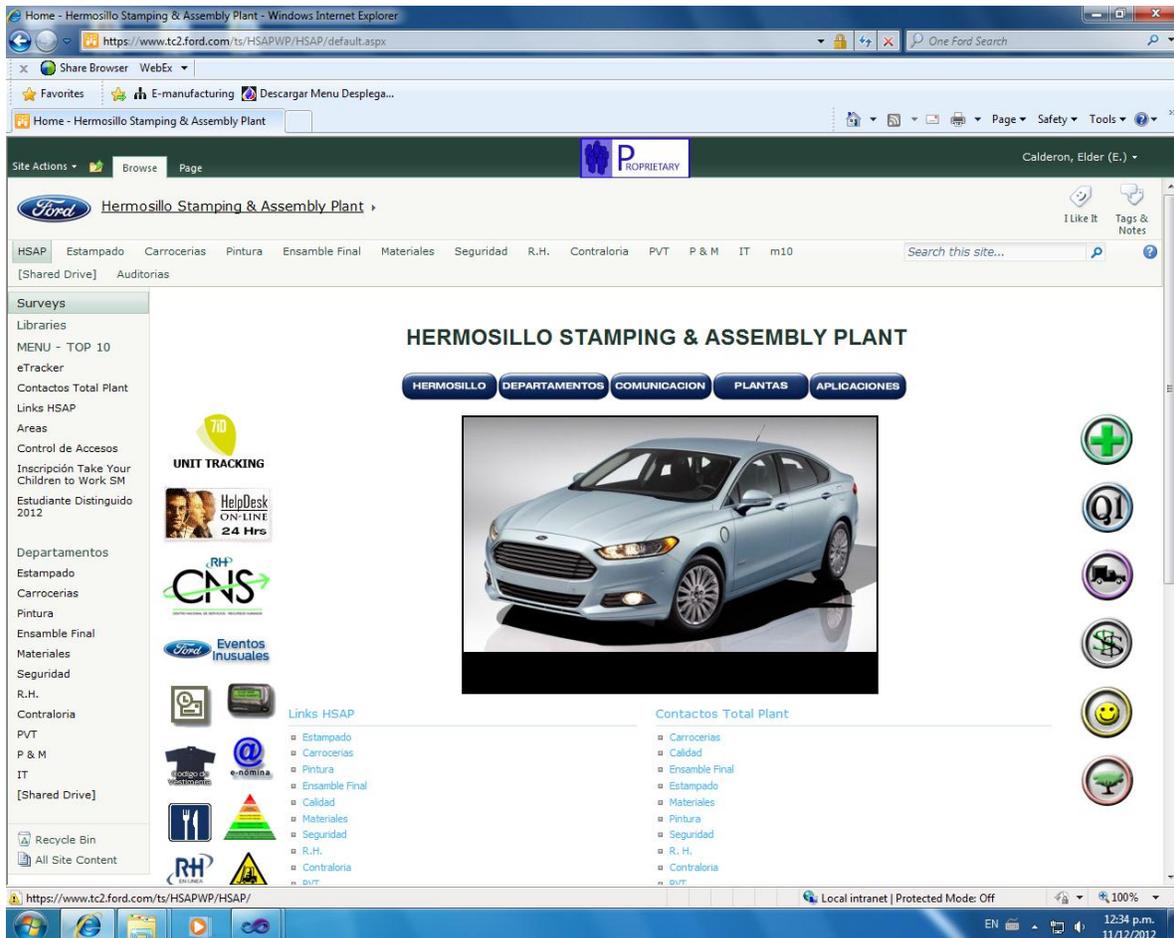


Fig.5 Pagina SharePoint

Para poder cambiar un permiso se tenía que ir a la pestaña site Actions, después en site Permissions.

Te aparecía una ventana con las diferentes opciones, por ejemplo modificar permisos, solo lectura, de en vez de eliminar, editar y lectura.

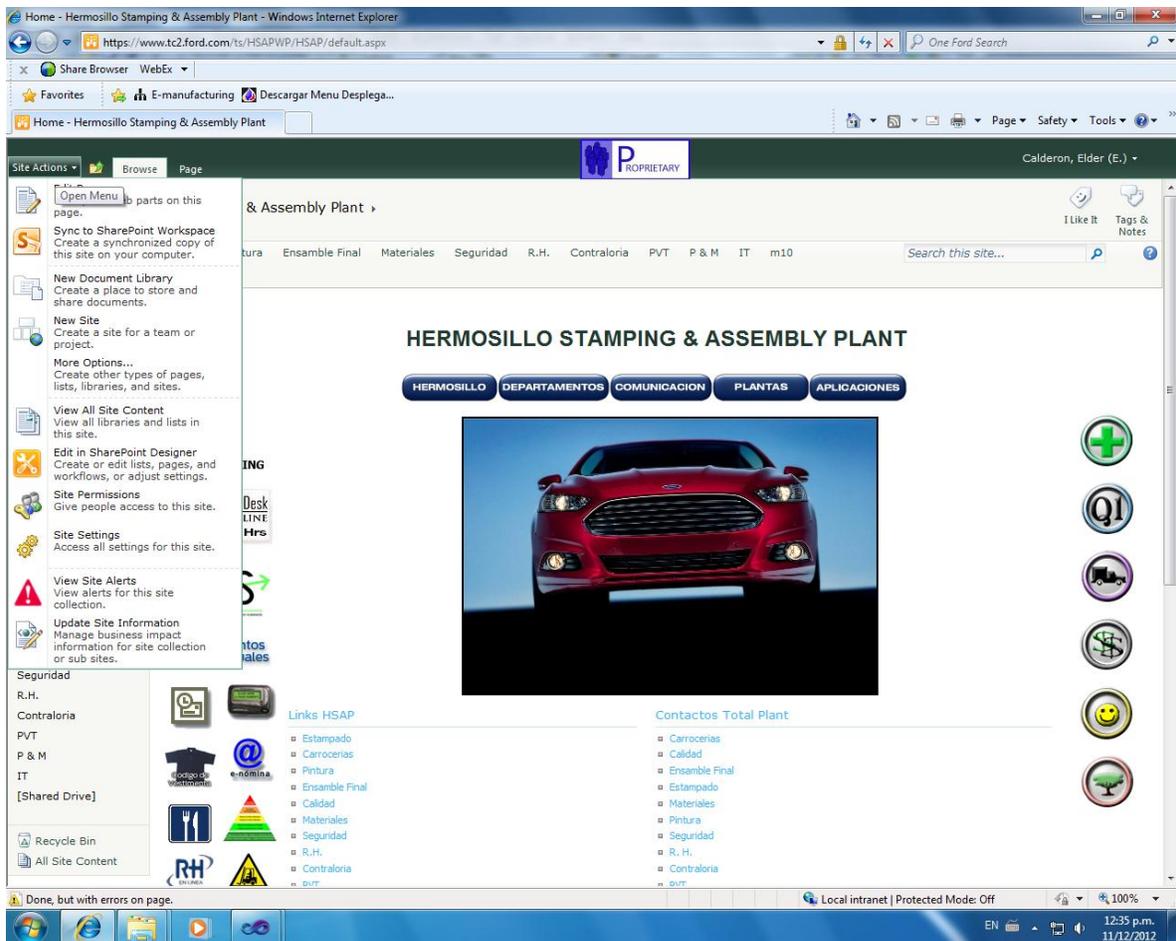


Fig.6 Pagina SharePoint permisos.

## Resultados obtenidos

La página de la pirámide de seguridad se puede acceder desde la dirección [www.hermosillo.ford.com](http://www.hermosillo.ford.com)

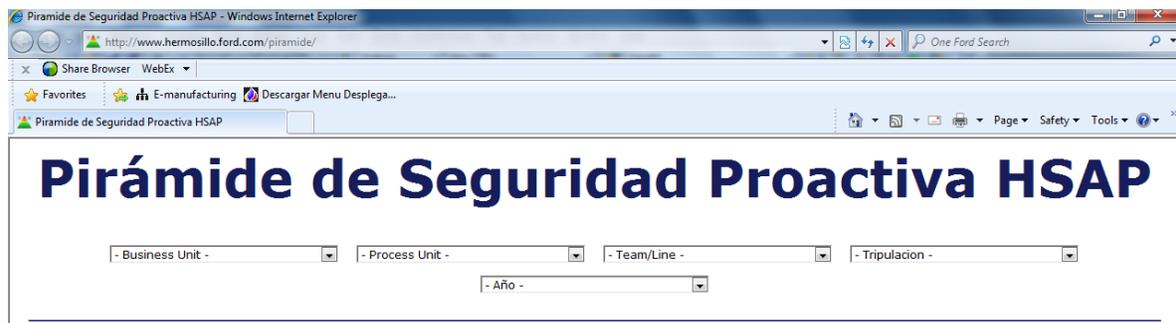


Fig.7 Acceso a página web pirámide de seguridad.

Con este sitio web los empleados pueden de manera fácil y rápida escribir las fallas que ven dentro de la empresa, todo esto con el afán de reducir accidentes y estar mejor preparados para evitarlos.

Se puede llevar un mejor control de los accidentes dentro de la planta.

## Conclusiones y recomendaciones

En mi beneficio aprendí lo que es trabajar con otro lenguaje de programación, el cual tiene muchas características diferentes al que tradicionalmente utilizamos como es php.

Se concluye con el trabajo de realización de pirámide de seguridad, la cual se llevó a cabo con el fin de un mejor control de accidentes.

Hubo una gran respuesta de los empleados, ya que diariamente se subía la información de posibles accidentes.

## **Retroalimentación**

La materia de programación visual, me ayudó mucho con este lenguaje de programación, este lenguaje es muy dinámico y tiene una potente herramienta de programación que es el Visual Studio.

Una desventaja es que tanto este software y Microsoft Sql Server 2008 R2 no son gratuitos, se ocupa de una licencia para poder utilizarlos.

Estuve practicando este lenguaje durante 6 meses, los cuales me llevaron a un gran entendimiento de c#.

Aprendí como utilizar un GridView, sus funciones; dar de alta información de un formulario a una base de datos, crear la conexión manualmente sin utilizar algún dataset.

Lo aprendido en bases de datos me ayudo bastante, ya que al realizar la base de datos, catálogos, tablas, relaciones, me fue de gran ayuda lo aprendido en clase.

Este lenguaje es moderno, pero muy utilizado por las grandes empresas, mi oportunidad de entrar en este ámbito, me abre las puertas para un trabajo con este tipo de programación.

Mi recomendación es tener una materia solo para este lenguaje, asp.net, utilizando c# y visual basic. Y darle más prioridad a utilizar sql server.

No solo adquirí habilidad para manejar este lenguaje de programación, al estar en contacto con demás personas, aprendí lo que es trabajar en equipo, tener ciertas responsabilidades, dar mi punto de vista sobre una situación, no quedarme callado y expresar lo que pienso.

## **Referencias bibliográficas y virtuales**

<http://es.wikipedia.org/wiki/JQuery>

[http://es.wikipedia.org/wiki/Sitio\\_web](http://es.wikipedia.org/wiki/Sitio_web)

[http://es.wikipedia.org/wiki/P%C3%A1gina\\_web](http://es.wikipedia.org/wiki/P%C3%A1gina_web)

<http://www.aspnetya.com.ar/detalleconcepto.php?codigo=67>

[http://www.industrial.uson.mx/files/guia\\_redaccion.pdf](http://www.industrial.uson.mx/files/guia_redaccion.pdf)

## Anexos

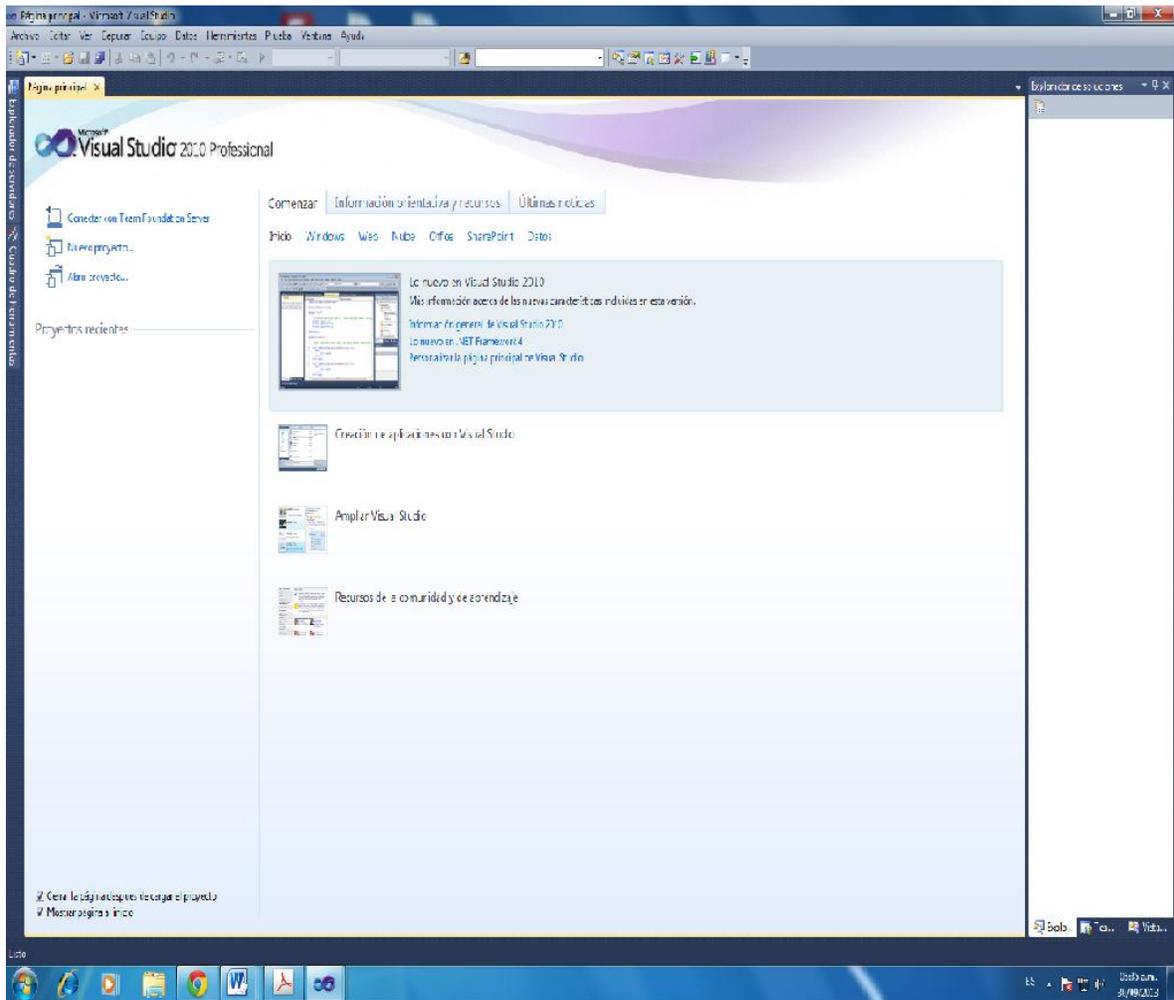
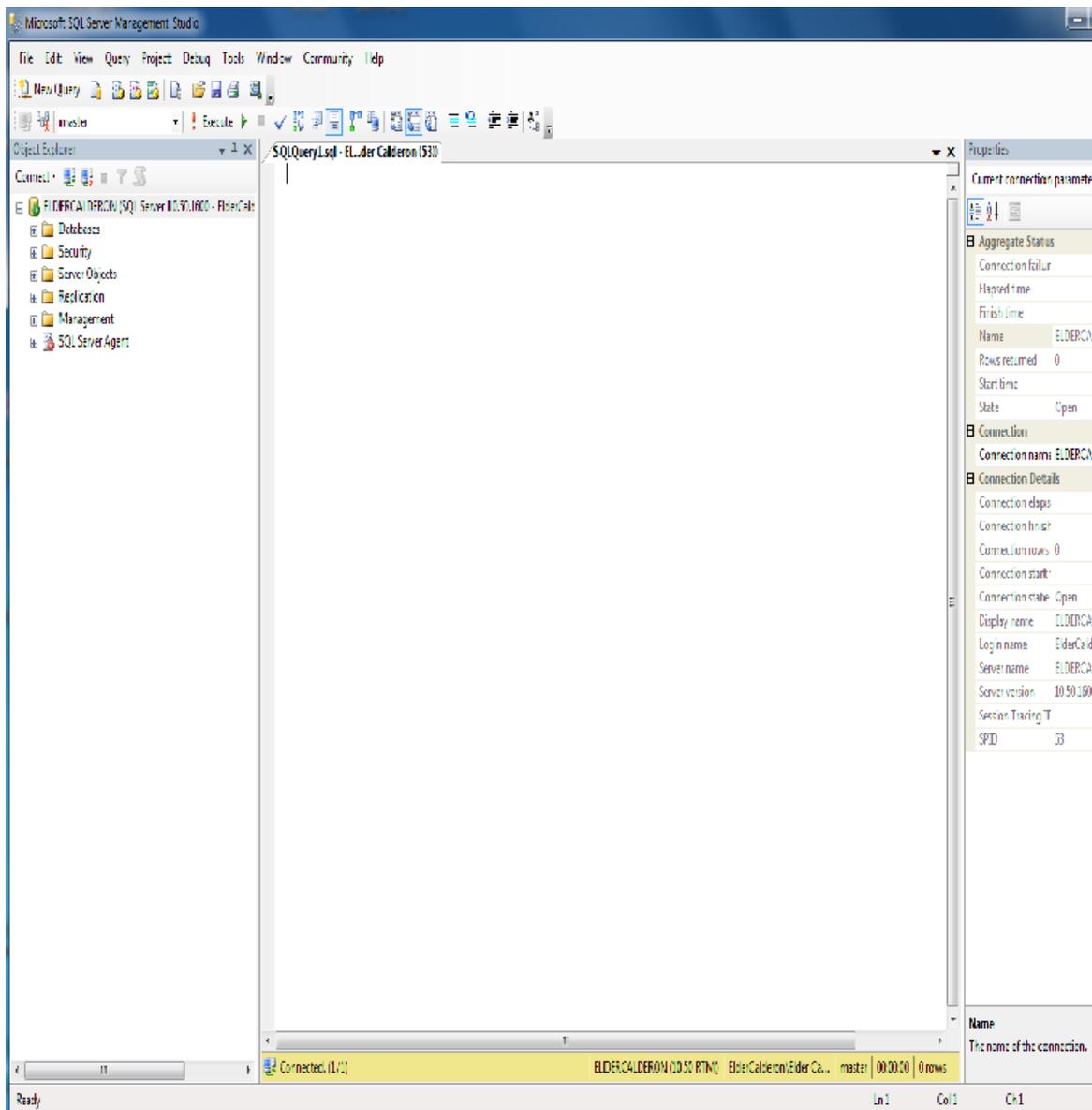


Fig.8 Herramienta de programación Visual studio.

Fig.9 Herramienta para realizar bases de datos.



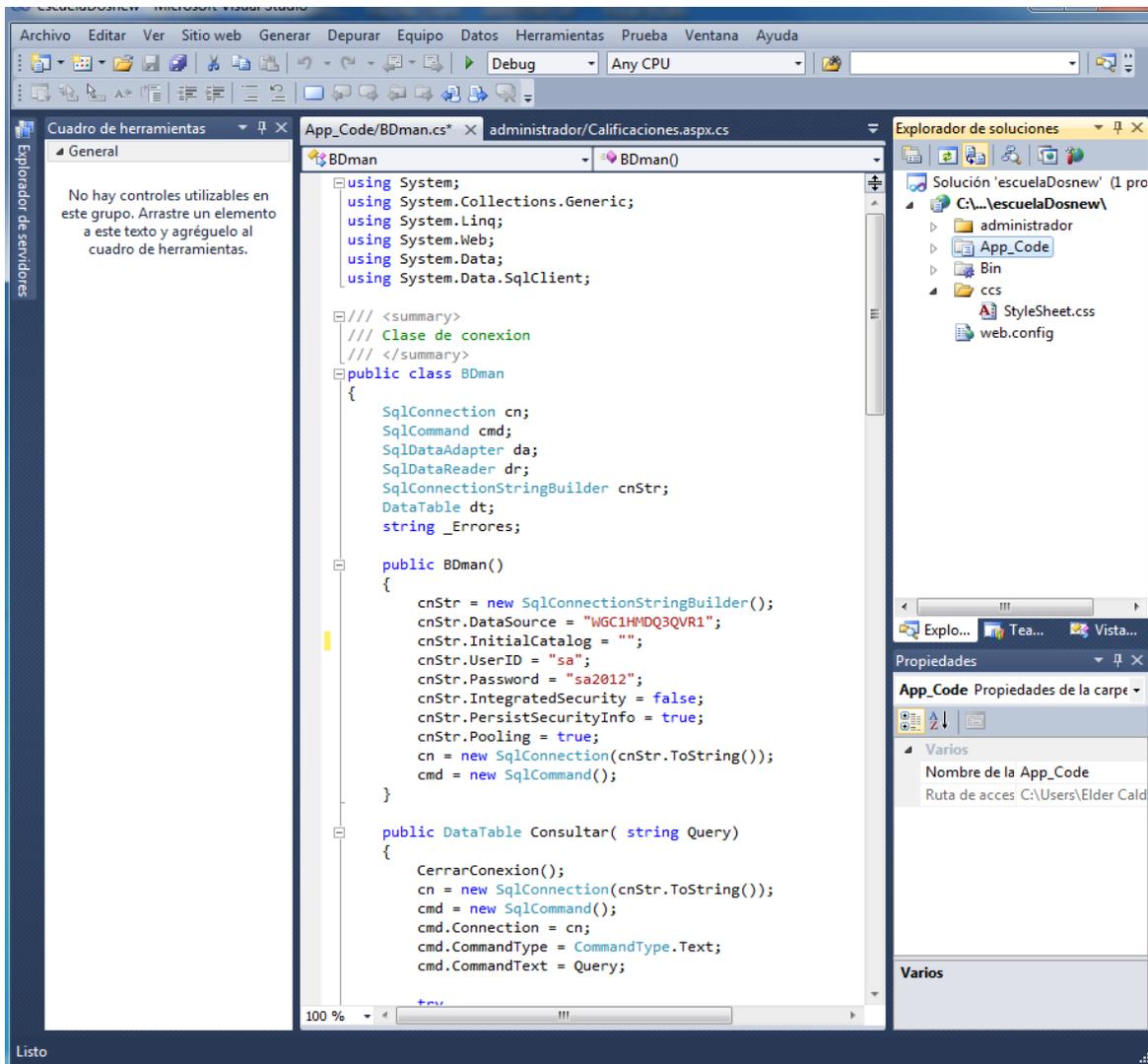


Fig.10 Conexión base de datos.