**DOCUMENTO TÉCNICO 3.0**

DESARROLLO O CODIFICACIÓN

DEL SOFTWARE

CONTROL DE SERVICIO NO CONFORME

(MODULOTEC)

|  |  |  |
| --- | --- | --- |
| *Creado por* | : | Oscar Ibáñez |
| *Fecha de Creación* | : | 14-12-2015 |
| *Revisado por* | : |  |
| *Fecha de Revisión* | : |  |
| *Validado por* | : |  |
| *Fecha de Validación* | : |  |

|  |  |  |
| --- | --- | --- |
| *PM* | : | Oscar Ibáñez |
| *DBA* | : | Miguel Contreras |
| *PR* | : | Omar Carmona |

**REGISTRO DE CONTROL DE CAMBIOS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *N°* | *Fecha* | *Versión* | *Descripción* | *Autor* | *Revisado por* |
| 1 | 14-12-2015 | 1.0 | Creación del documento técnico | Oscar Ibáñez |  |

**INDICE DE CONTENIDOS**

|  |  |  |  |
| --- | --- | --- | --- |
| 1. | Introducción……………………………………………………………………………………………………………………………………………………….. | |  |
| 2. | Objetivo General………………………………………………………………………………………………………………………………………………… | |  |
| 3 | Objetivos Específicos………………………………………………………………………………………………………………………………………….. | |  |
| 4. | Alcance………………………………………………………………………………………………………………………………………………………………. | |  |
| 5. | Capa de Presentación..……………………………………………………………………………………………………………….………………………. | |  |
|  | 5.1 | Descripción de las vistas, ventanas y/o formularios….……………………………………..…………………………………….. |  |
|  | 5.2 | Código fuente de las vistas, ventanas y/o formularios más relevantes o representativos……..……………….. |  |
|  | 5.3 | Biblioteca de Controles de Formulario……………………………………………………………………………………………………. |  |
| 6. | Capa de Lógica de Negocios………………………………………………………………………………………………………………................... | |  |
|  | 6.1 | Descripción de Código Fuente de Eventos en Controles de Formulario……….………………………….................. |  |
|  | 6.2 | Descripción de Clases, Atributos y Métodos………………………....……………………………………………….................. |  |
|  | 6.3 | Descripción de Librerías utilizadas………………………………………………………………………………………………………….. |  |
| 7. | Capa de Acceso a Datos……………………………………………….…………………………………………………………………………………….. | |  |
|  | 7.1 | Descripción de Cadena de Conexión a la Base de Datos……..………………………………………………………………….. |  |
|  | 7.2 | Definición de los Script SQL LDD……………………………………………………………………………………………………………… |  |
|  | 7.3 | Definición de los Script SQL LMD.…………………………………………………………………………………………………………… |  |
| 8. | Conclusiones………………………………………………………………………………………………………………………………………………………. | |  |
| 9. | Bibliografía……………………………………………………………………………………………………………….………………………………………… | |  |
| 10. | Glosario técnico…………………………………………………………………………………………………………………………………………………. | |  |
| 11. | Anexos……………………………………………………………………………………………………………………………………………………………….. | |  |

**INDICE DE TABLAS Y FIGURAS**

|  |  |  |
| --- | --- | --- |
| **TABLAS** | | |
| Tabla 1 | Descripción del contenido de la tabla o título....…………………………………………………………………………………………… |  |
| Tabla 2 | Descripción del contenido de la tabla o título....…………………………………………………………………………………………… |  |
| Tabla 3 | Descripción del contenido de la tabla o título....…………………………………………………………………………………………… |  |
| Tabla 4 | Descripción del contenido de la tabla o título....…………………………………………………………………………………………… |  |

|  |  |  |
| --- | --- | --- |
| **FIGURAS** | | |
| Figura 1 | Descripción del contenido de la figura o título....…………………………………………………………………………………………. |  |
| Figura 2 | Descripción del contenido de la figura o título....…………………………………………………………………………………………. |  |
| Figura 3 | Descripción del contenido de la figura o título....…………………………………………………………………………………………. |  |
| Figura 4 | Descripción del contenido de la figura o título....…………………………………………………………………………………………. |  |

## INTRODUCCION

Es necesario para toda organización técnica de capacitación poseer y mantener un sistema de gestión de la calidad, esto para asegurar que sus procesos y administración de recursos sean de calidad.

En Chile aún existen organizaciones de capacitación que manejan su información de forma manual o tan solo utilizando herramientas de ofimática, esto si bien es funcional, en algunos casos la falta de organización y conocimiento de los recursos que se poseen al interior de la empresa representan una amenaza para una administración efectiva.

Uno de los módulos que más se requiere llevar un registro y control dentro de una OTEC es el Control de Servicio No Conforme, por la necesidad de llevar la documentación de las incidencias que ocurran dentro de la institución, a los encargados especializados de cada área para su análisis, evaluación de cumplimiento de las incidencias y lo más importante darle un seguimiento a las no conformidades de una forma rápida y efectiva.

## OBJETIVO GENERAL

El objetivo de este documento es proveer una guía del proceso de codificación del sistema de Control de Servicio No conforme, y mostrar de una manera técnica pero sencilla los componentes que incluirán el sistema, su proceso de elaboración, sus librerías, vistas y procesos de negocio del proyecto.

## OBJETIVOS ESPECÍFICOS

* Mostrar componentes de las diferentes capas y clases del sistema.
* Describir los procesos y eventos principales del sistema.
* Definir claramente los controles utilizados y validaciones.

## ALCANCES

En este informe solo se presentará la documentación referente al proceso de codificación del módulo, y en algunos casos (código fuente del sistema) solo se presentarán los componentes más representativos del sistema, dada la extensión del código que haría este informe difícil de comprender y/o interpretar.

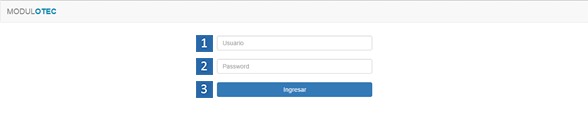
## CAPA DE PRESENTACIÓN

Se refiere a la presentación del programa frente al usuario, esta presentación debe cumplir su propósito con el usuario final, una presentación fácil de usar y amigable.

DESCRIPTOR DE LAS VISTAS DEL SISTEMA MODULOTEC

**VISTA DE ACCESO AL SISTEMA**

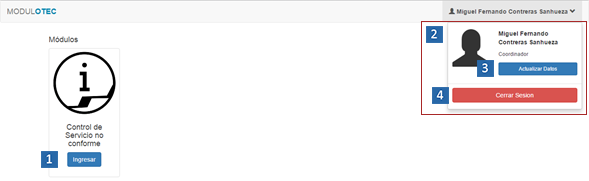
**Index.aspx**



|  |  |
| --- | --- |
| **Número** | **Descripción** |
| **1** | Campo de texto de Ingreso de Rut de Usuario. |
| **2** | Campo de texto de Ingreso de Password. |
| **3** | LinkButton Ingresar al sistema.  Valida campos requeridos y compara información de la base de datos. |

**VISTA PRINCIPAL DE LOS MODULOS**

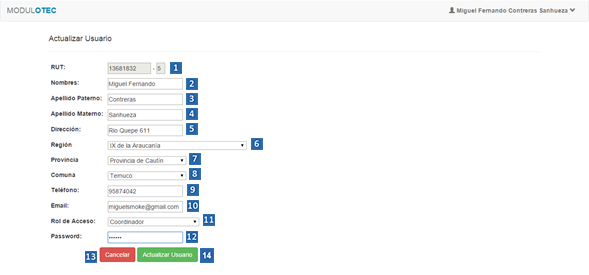
**MainModulos.aspx**



|  |  |
| --- | --- |
| **Número** | **Descripción** |
| **1** | Botón de Ingreso al Módulo de Control de Servicio no Conforme. |
| **2** | Panel de perfil de Usuario. |
| **3** | Botón de ingreso de actualización de los datos de Usuario, permite actualizar los datos del usuario. |
| **4** | Botón de Cerrar Sesión de perfil de Usuario, termina la sesión activa y envía a la vista de Acceso del Sistema “Index.aspx”. |

**VISTA DE AGREGAR Y ACTUALIZAR USUARIOS**

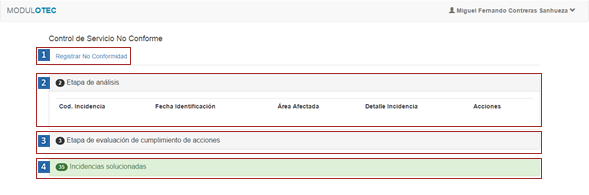
**AgregarUsuario.aspx**



|  |  |
| --- | --- |
| **Número** | **Descripción** |
| **1** | Campo de texto de Ingreso de Rut de Usuario y campo de digito verificador. |
| **2** | Campo de texto de Ingreso de Nombres del Usuario. |
| **3** | Campo de texto de Ingreso de Apellido Paterno de Usuario. |
| **4** | Campo de texto de Ingreso de Apellido Materno de Usuario. |
| **5** | Campo de texto de Ingreso de Dirección de Usuario. |
| **6** | DropDownList de despliege de la lista de Regiones, al cambiar el valor de la lista despliega el DropDownList de Provincias. |
| **7** | DropDownList de Provincias, desencadenado por el DropDownList Regiones, al cambiar el valor de la lista despliega el DropDownList de Comunas |
| **8** | DropDownList de Comunas, desencadenado por el DropDownList Provincias, muestra una lista de regiones. |
| **9** | Campo de texto de Ingreso de número de Telefono de Usuario. |
| **10** | Campo de texto de Ingreso de Email de Usuario, se valida un formato de correo valido xx@xx.xx |
| **11** | DropDownList de Roles de Acceso de Usuario, carga la lista de roles disponibles, para el usuario ya registrado, este aparece deshabilitado, no así al registrar un nuevo usuario. |
| **12** | Campo de texto de Ingreso de Password de Usuario. |
| **13** | Botón Cancelar, desencadena un mensaje de validación, al confirmar que envía a la vista principal “MainModulos.aspx”. |
| **14** | Botón Agregar /Actualizar Usuario, guarda los datos en la base de datos.  Valida los campos requeridos. |

**VISTA PRINCIPAL DEL MODULO DE CONTROL DE SERVICIO NO CONFORME**

**GestionIncidencias.aspx**



|  |  |
| --- | --- |
| **Número** | **Descripción** |
| **1** | LinkButton de Ingreso a la vista Registrar No Conformidad. |
| **2** | Panel desplegable de Incidencias en Etapa de Análisis. |
| **3** | Panel desplegable de Incidencias en Etapa de Evaluación de Cumplimiento de Acciones. |
| **4** | Panel desplegable de Incidencias Solucionadas. |

**VISTA DE INGRESO DE NO CONFORMIDADES**

**IngresarNoConformidad.aspx**



|  |  |
| --- | --- |
| **Número** | **Descripción** |
| **1** | DropDownList de selección de Tipo de Incidencia. |
| **2** | DropDownList de selección de Modo de Detección de la incidencia. |
| **3** | Campo de texto de ingreso de la Fecha de Identificación de la No Conformidad. |
| **4** | Campo de texto de ingreso del Area Afectada. |
| **5** | Campo de texto de ingreso del Proceso Afectado. |
| **6** | Campo de texto de ingreso de Descripción de la No Conformidad. |
| **7** | Botón Cancelar, tras validar que se desea salir, envía a la vista principal del Modulo de Servicio No Conforme “GestionIncidencias.aspx” |
| **8** | Botón Ingresar, registra todos los datos de la No Conformidad y los guarda en la base de datos.  Valida los campos requeridos. |

**VISTA DE ANALISIS DE NO CONFORMIDADES**

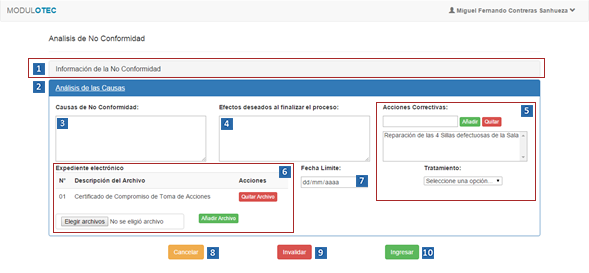
**AnalisarNoConformidades.aspx**



|  |  |
| --- | --- |
| **Número** | **Descripción** |
| **1** | Panel desplegable de Información de la No Conformidad. |
| **2** | Campo de texto desabilitado. Muestra el detalle de la No Conformidad. |
| **3** | Panel Desplegable de Análisis de las Causas. |
| **4** | Botón Cancelar que tras validar la salida, envía a la vista principal del Modulo de Control de Servicio No Conforme “GestionIncidencias.aspx”. |
| **5** | Botón Invalidar, invalida la No Conformidad y cambia el estado de la Incidencia a “Rechazada”. |
| **6** | Botón Ingresar, si no se han llenado los datosde Análisis de las Causas no tiene ninguna acción. |

**VISTA DE ANALISIS DE NO CONFORMIDADES**

**AnalisarNoConformidades.aspx**



|  |  |
| --- | --- |
| **Número** | **Descripción** |
| **1** | Panel desplegable de Información de la No Conformidad (Vista Anterior). |
| **2** | Panel Desplegable de Análisis de las Causas. |
| **3** | Campo de texto de Ingreso de Causas de No Conformidad. |
| **4** | Campo de texto de Ingreso de los Efectos Deseados al Finalizar el Proceso. |
| **5** | Panel de Añadir o Quitar Acciones Correctivas. |
| **6** | Panel de Añadir o Quitar Expediente Electronico. |
| **7** | Campo de texto de Fecha Limite. |
| **8** | Botón Cancelar que envía a la vista principal del Modulo de Control de Servicio No Conforme “GestionIncidencias.aspx”. |
| **9** | Botón Invalidar que invalida la No Conformidad y cambia el estado de la Incidencia a “Rechazada”. |
| **10** | Botón Ingresar que registra El Análisis de las Causas y guarda los datos en la base de datos, cambia el estado de la incidencia. Finaliza el proceso de análisis.  Valida los campos requeridos. |

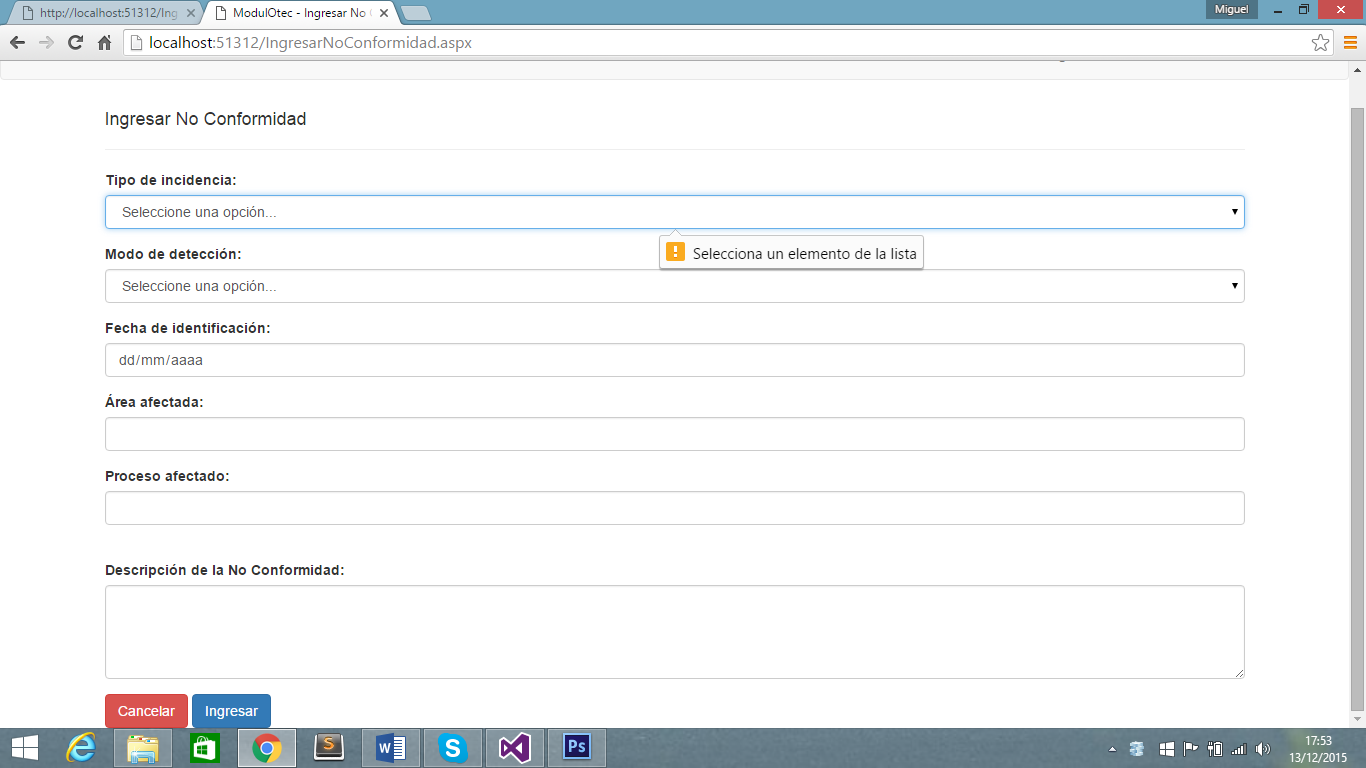
**VISTA PRINCIPAL DEL SEGUIMIENTO DE LAS INCIDENCIAS**

**RevisionCumplimiento.aspx**

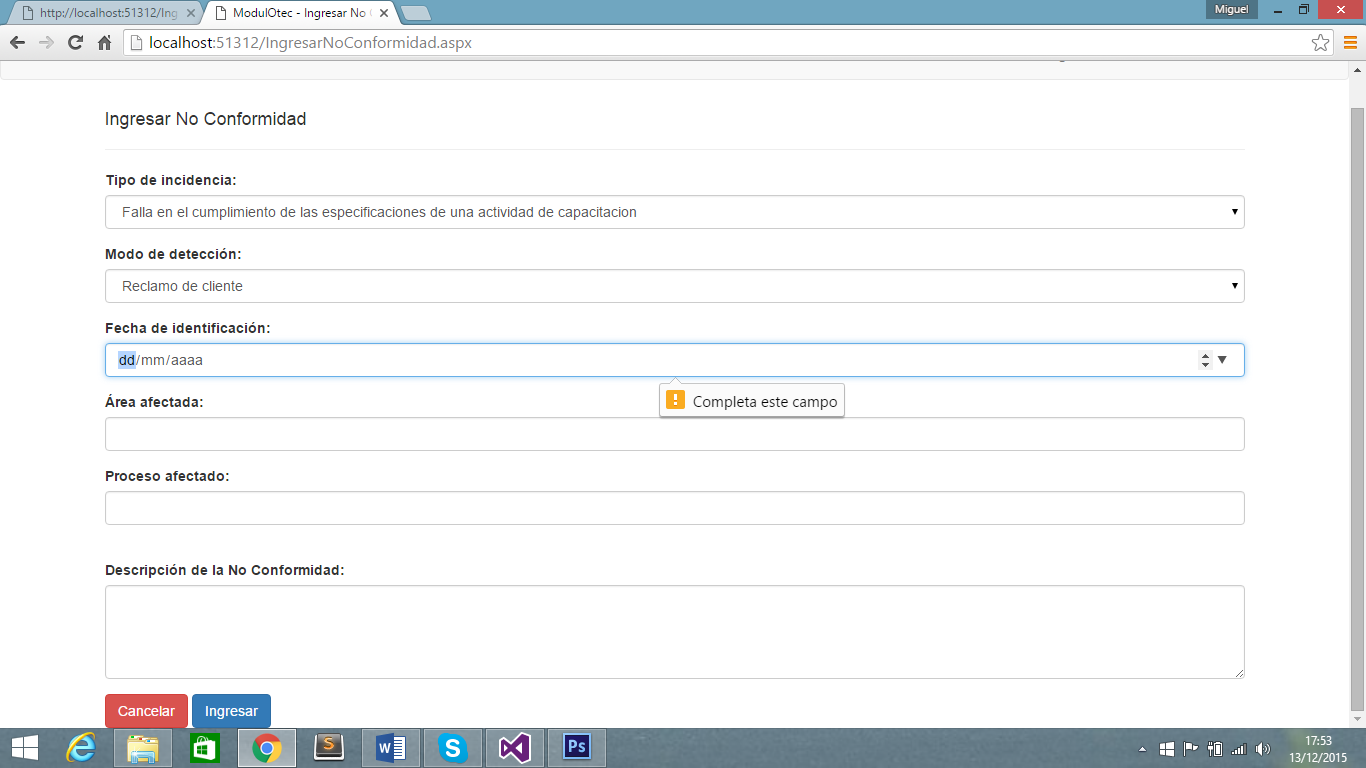


|  |  |
| --- | --- |
| **Número** | **Descripción** |
| **1** | Panel de Información de la No Conformidad |
| **2** | Panel de Información de las Acciones Correctivas (análisis de causas), ingresadas en la vista anterior. |
| **3** | Panel de Revisión de Cumplimiento |
| **4** | CheckBoxList de las Acciones Correctivas,ingresadas anteriormente. |
| **5** | Campo de texto para describir Observaciones de las acciones correctivas realizadas. |
| **6** | Botón Cancelar que envía a la vista principal del Modulo de Control de Servicio No Conforme “GestionIncidencias.aspx”. |
| **7** | Botón Re Análisis que cambia el estado de la incidencia y la envia a la etapa de Análisis de la No Conformidad. |
| **8** | Botón Finalizar que finaliza el proceso de la incidencia y cambia su estado a Finalizada. |

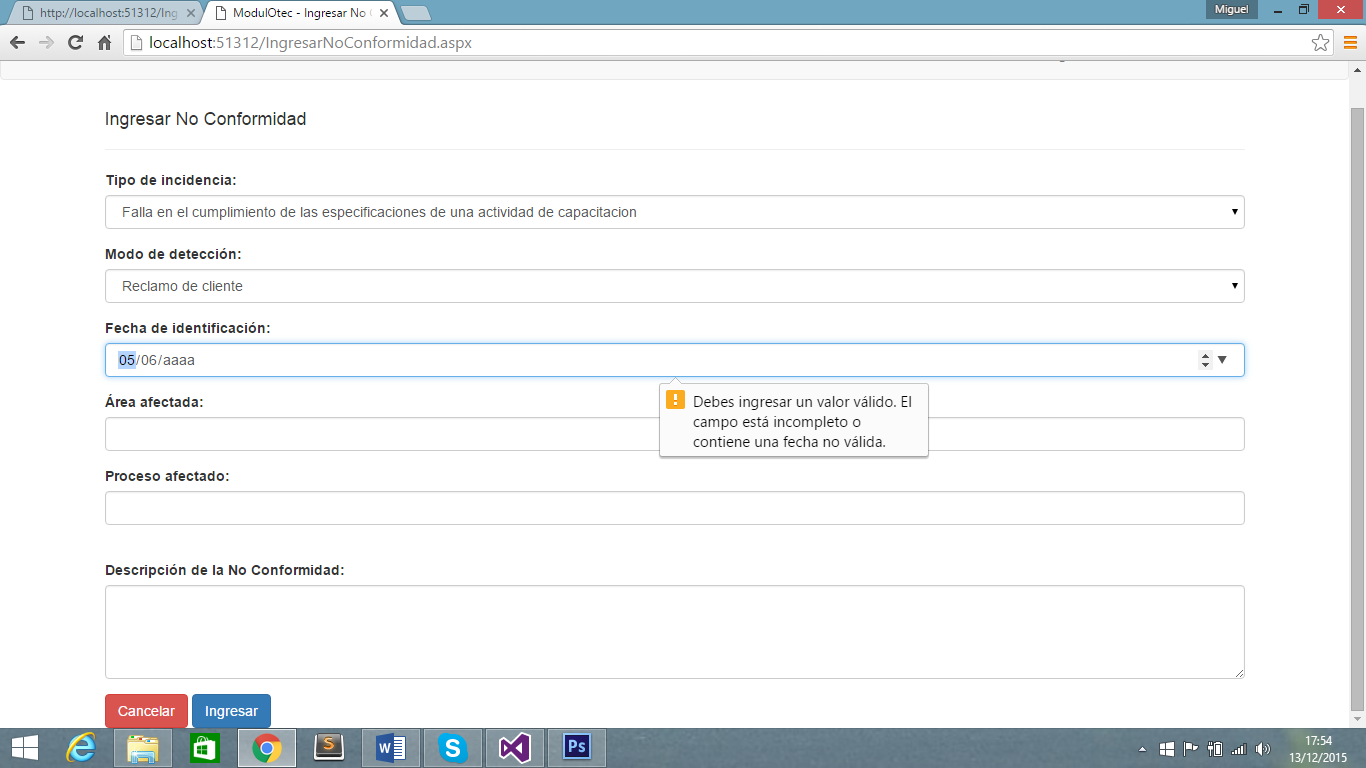
MENSAJES DE VALIDACION



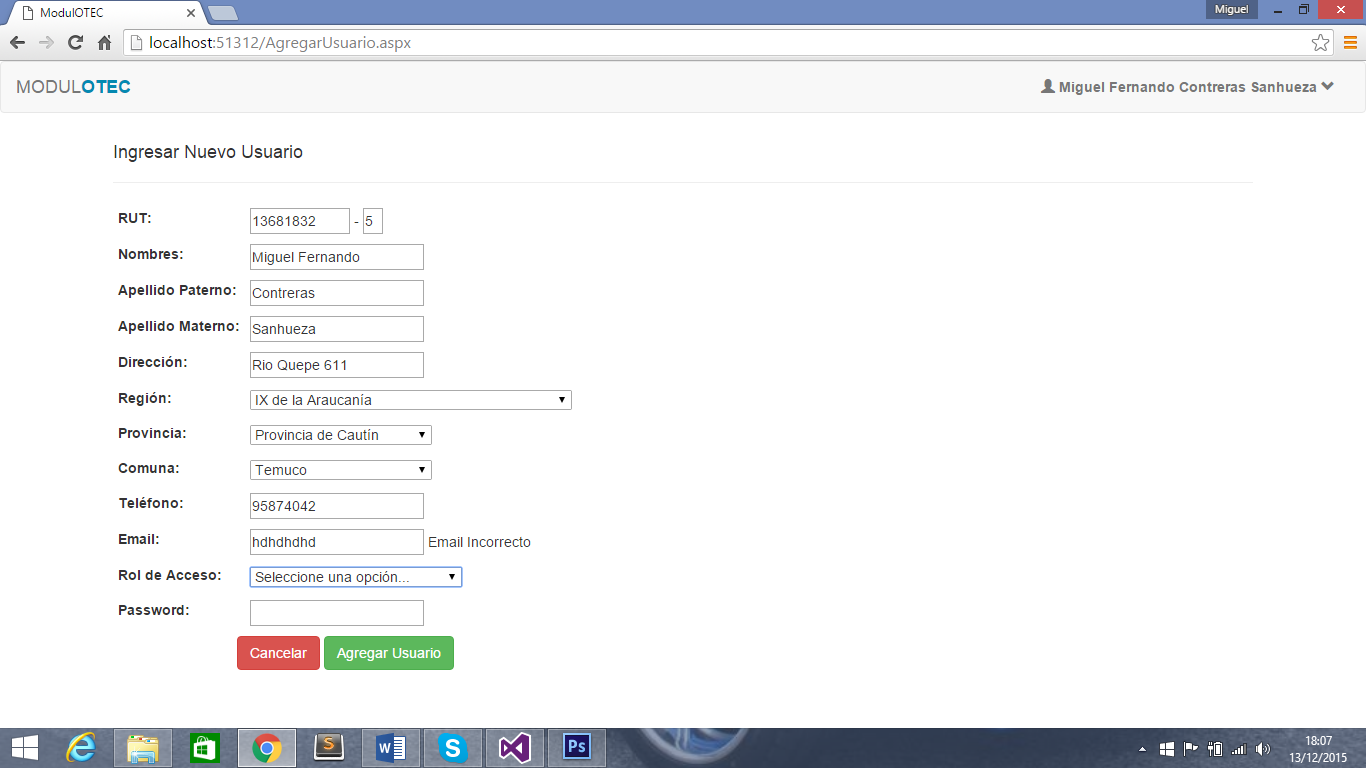
Mensaje de validación de selección de algún elemento de un DropDownList.



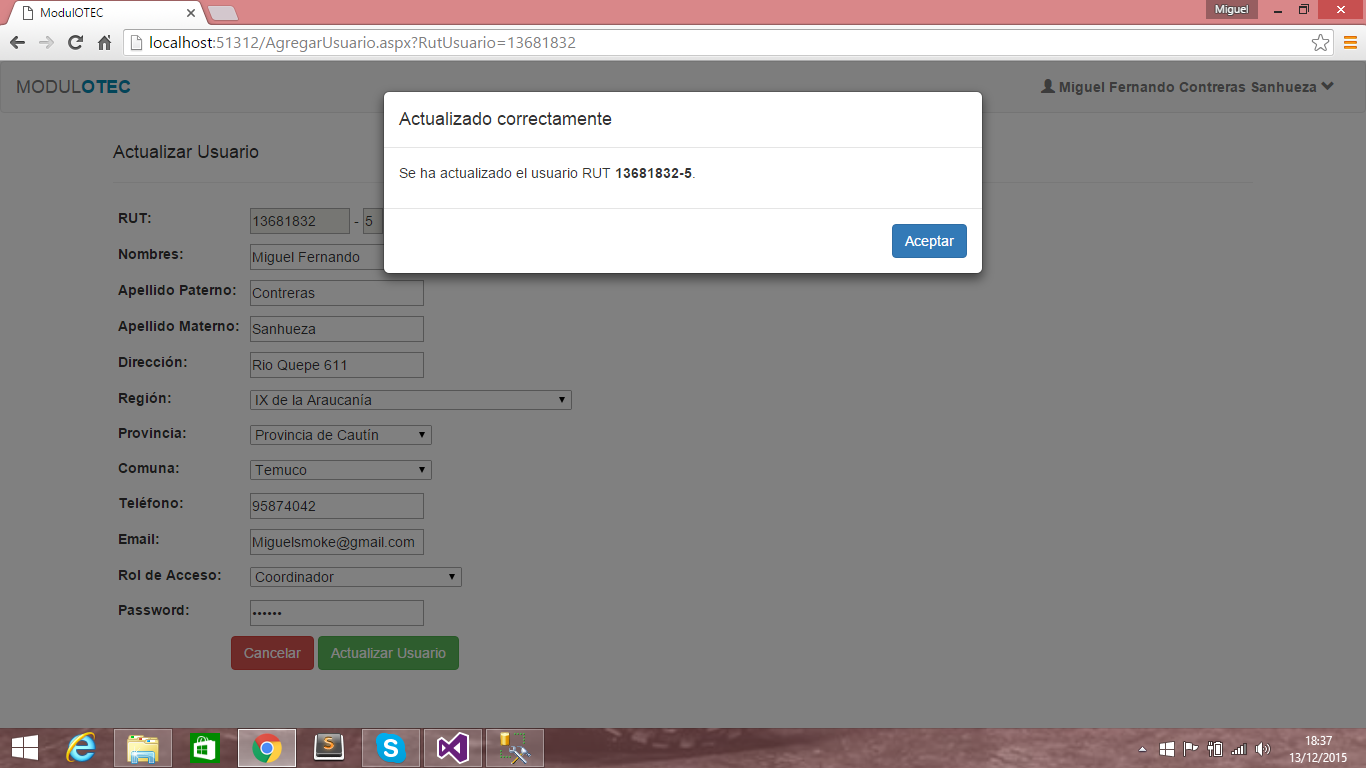
Mensaje de validación de campos de texto vacío.



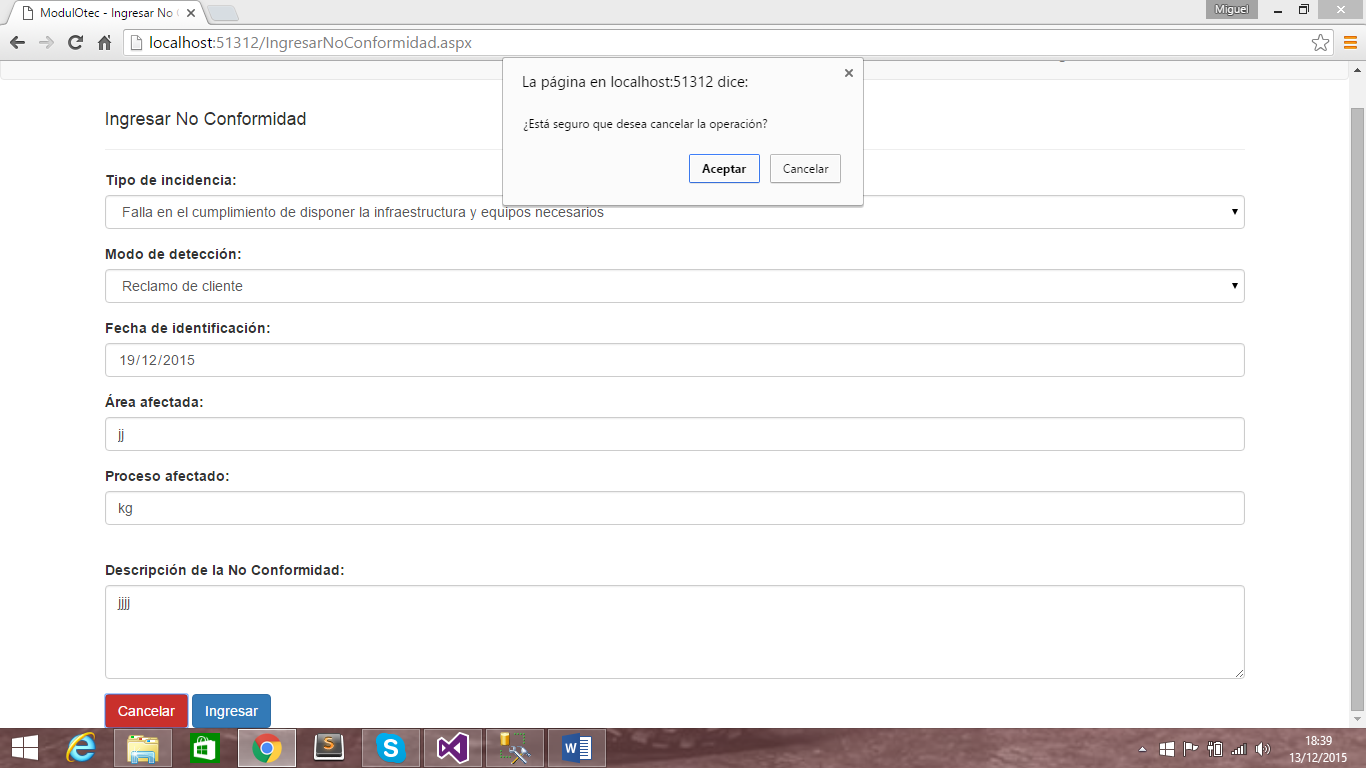
Mensaje de validación de que el valor ingresado no es válido, solo se permiten valores numéricos o la fecha no es válida.



Mensaje de validación que el Email es incorrecto.



Mensaje de que los datos se han actualizado correctamente.



Mensaje de consulta si se desea cancelar una operación.

**ESTRUCTURA DE LA APLICACIÓN**

ModulOTEC utiliza la siguiente distribución de directorios en etapa de desarrollo.

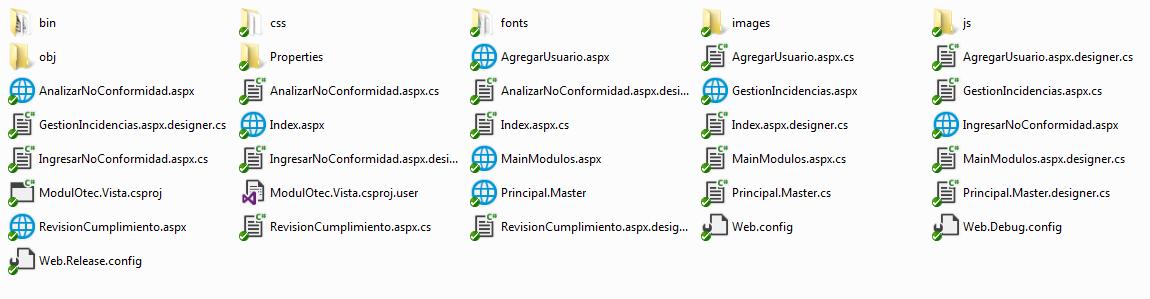


Ilustración 1 - Vista del Directorio de la Aplicación

**Bin:** ASP.NET reconoce la carpeta Bin de un sitio web como una carpeta especial para determinados tipos de contenido. Una carpeta Bin está diseñada para contener ensamblados compilados (archivos .dll) para controles y componentes ASP.NET personalizados, en este caso se encuentran los archivos de EntityFramework y los dll de los apartados de vistas, entidades, datos y negocio del sistema.

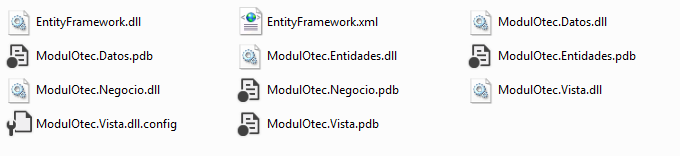


Ilustración 2 - Carpeta bin

**Css:** En esta carpeta se encuentran los archivos de hojas de estilos externos que utilizará el sistema, acá podemos hallar las hojas de estilo del framework css Bootstrap, y el archivo css con los estilos personalizados de esta web.

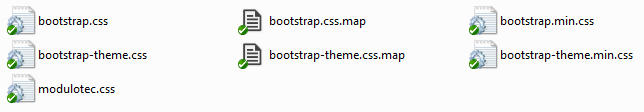


Ilustración 3 - Carpeta css

**Fonts:** Se almacenan las fuentes que utiliza actualmente sistema.

**Images:** Se almacenan las fuentes que utiliza actualmente sistema.

**Js:** En esta carpeta se incluyen las librerías JavaScript del proyecto, se pueden observar las librerías js de Bootstrap y Jquery.

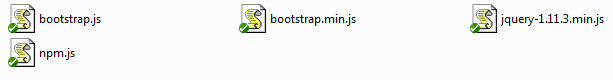


Ilustración 4 - Carpeta js

**Obj:** Se almacenan archivos generados por Visual Studio en el proceso de depuración del sistema.

**Properties:** Se guardan archivos que guardan la información general sobre un ensamblado.

## CAPA DE LÓGICA DE NEGOCIO

La capa de negocio se comunica con la capa de presentación, para recibir las solicitudes de información y presentar los resultados, y con la capa de datos para almacenar o recuperar datos de la base de datos.

Estructura de la capa

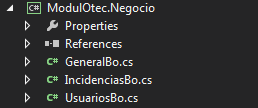


Ilustración 5 - Estructura Capa de Negocio

Se pueden observar tres clases dentro de la estructura de la capa, GeneralBo.cs, IncidenciasBo.cs y UsuariosBo.cs, además incorpora las propiedades de la capa (Properties) y las Referencias a diferentes componentes del proyecto.

**GeneralBo (Clase)**

Clase utilizada con la finalidad de cargar la información general de las vistas

**GeneralBo (using):** Bibliotecas de clases que incluye la Clase GeneralBo.

**System:** Contiene clases fundamentales y clases base que definen tipos de datos de referencia y de valor de uso frecuente, eventos y controladores de eventos, interfaces, atributos y excepciones de procesamiento.

**System.Collections.Generic:** Contiene interfaces y clases que definen colecciones genéricas, lo que permite que los usuarios creen colecciones fuertemente tipadas para proporcionar una mayor seguridad de tipos y un rendimiento mejor que los de las colecciones no genéricas fuertemente tipadas.

**System.Linq:** Proporciona clases e interfaces que admiten consultas que utilizan Language-integrated query (LINQ).

**Entidades:** Contiene clases para la utilización de la base de datos a través de objetos y entidades.

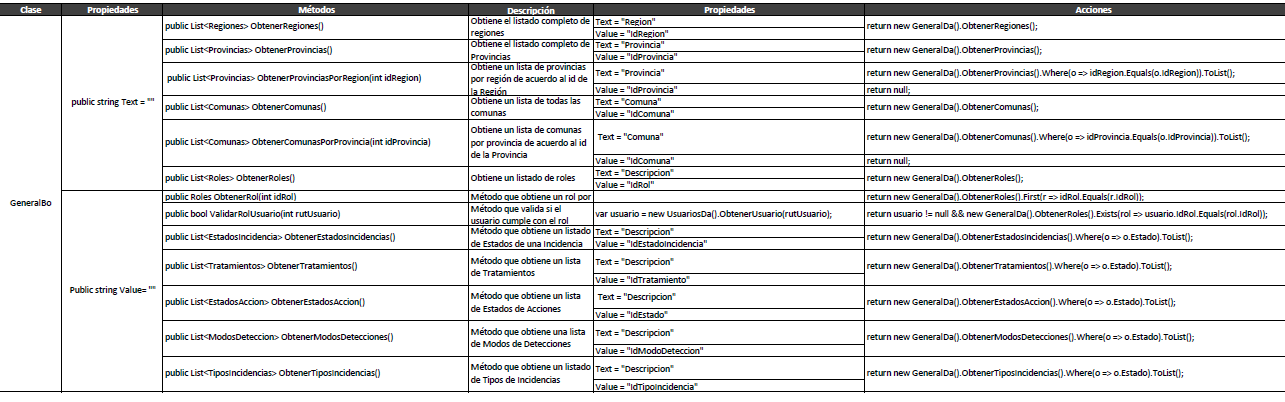
**Datos:** Contiene acceso a las clases y métodos de la capa de datos del sistema, para solicitar el almacenamiento o la recuperación de datos.

Ilustración - Métodos Clase GeneralBo

**IncidenciasBo (Clase)**

**IncidenciasBo (using):** Bibliotecas de clases que incluye la Clase **IncidenciasBo**.

**System.Collections.Generic:** Contiene interfaces y clases que definen colecciones genéricas, lo que permite que los usuarios creen colecciones fuertemente tipadas para proporcionar una mayor seguridad de tipos y un rendimiento mejor que los de las colecciones no genéricas fuertemente tipadas.

**Datos:** Contiene acceso a las clases y métodos de la capa de datos del sistema, para solicitar el almacenamiento o la recuperación de datos.

**Entidades:** Contiene clases para la utilización de la base de datos a través de objetos y entidades.

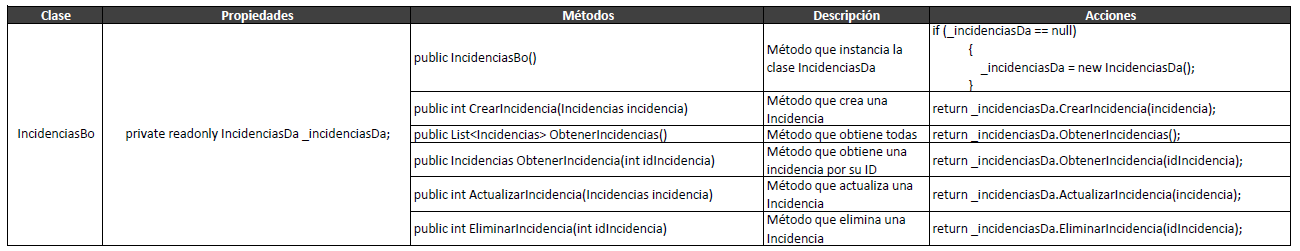


Ilustración - Métodos Clase IncidenciasBo

**UsuariosBo (Clase)**

**UsuariosBo (using):** Bibliotecas de clases que incluye la Clase **UsuariosBo.**

**System.Collections.Generic:** Contiene interfaces y clases que definen colecciones genéricas, lo que permite que los usuarios creen colecciones fuertemente tipadas para proporcionar una mayor seguridad de tipos y un rendimiento mejor que los de las colecciones no genéricas fuertemente tipadas.

**Datos:** Contiene acceso a las clases y métodos de la capa de datos del sistema, para solicitar el almacenamiento o la recuperación de datos.

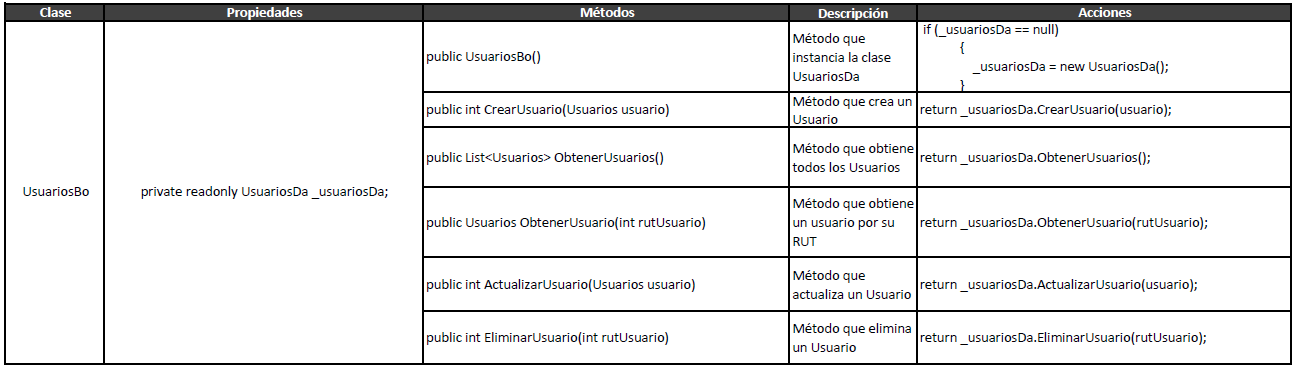
**Entidades:** Contiene clases para la utilización de la base de datos a través de objetos y entidades.

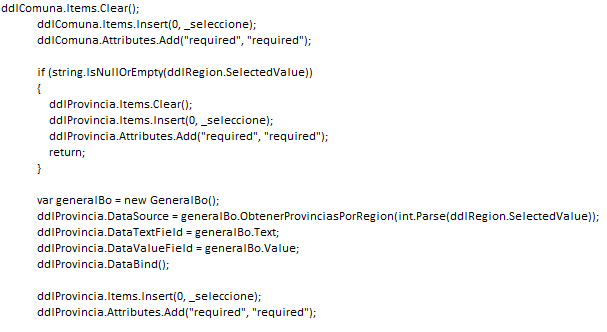
Ilustración - Métodos Clase UsuariosBo

**Item:** AgregarUsuario.aspx

**Control de formulario:** DropDownList "ddlRegion"

**Evento:** OnSelectedIndexChanged="RegionSelectedIndexChanged"

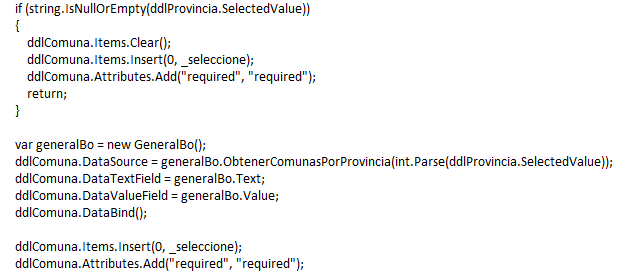
**Código Fuente:**



**Descripción:** Al cambiar la opción seleccionada deldropdownlist "ddlRegion", realiza un auto post back y carga las provincias de acuerdo al Identificador de la Región seleccionada.

**Item:** DropDownList "ddlProvincia"

**Control de formulario:** OnSelectedIndexChanged="ProvinciaSelectedIndexChanged"

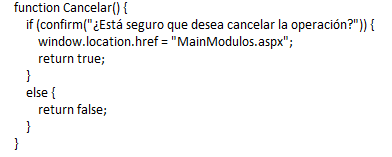
**Código Fuente:** 

**Descripción:** Al cambiar la opción seleccionada del dropdownlist "ddlProvincia", realiza un auto post back y carga las Comunas de acuerdo al Identificador de la Provincia seleccionada.

**Control de Formulario: Button "btnCancelar"**

**Evento:** **OnClientClick="return Cancelar();"**

**Código Fuente:**

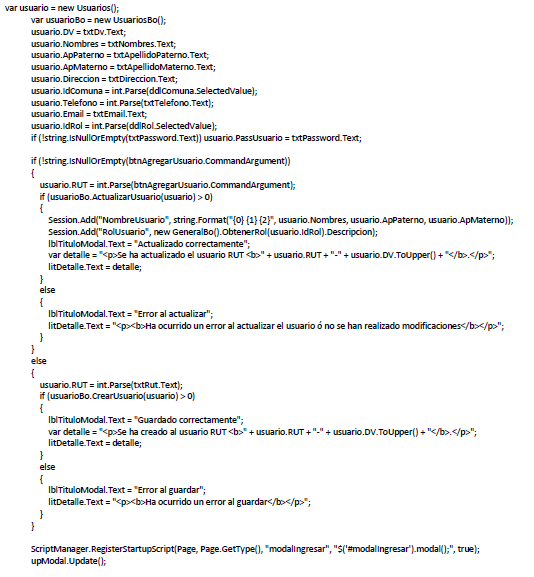


**Descripción:** Validación Javascript, al ser presionado el botón Cancelar, la validación es activada, el código genera una ventana de confirmación, preguntando al Usuario si está seguro que desea cancelar la operación.

**Control de Formulario:** Button "btnAgregarUsuario"

**Evento:** OnClick="AgregarUsuarioClick"

**Código Fuente:**



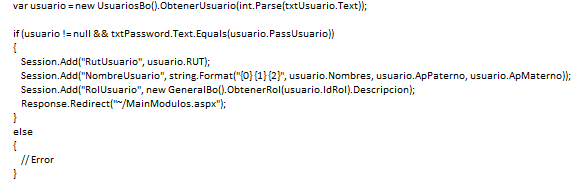
**Descripción:** Al darle click al botón Agregar Usuario, el sistema comprueba que los datos ingresados estén validados, asigna los campos de texto y seleccionables a las propiedades de la clase Usuarios(), en caso de que el proceso sea satisfactorio se genera un Modal al usuario que muestra en mensake "Se ha actualizado el usuario (RUT del usuario actualizado)", en caso de que la operación falle muestra el siguiente mensaje en el modal "Ha ocurrido un error al actualizar el usuario o no se han realizado modificaciones".

**Item:** Index.aspx

**Control de Formulario:** LinkButton "lnkIngresar"

**Evento:** OnClick="IngresarOnClick"

**Código Fuente:**



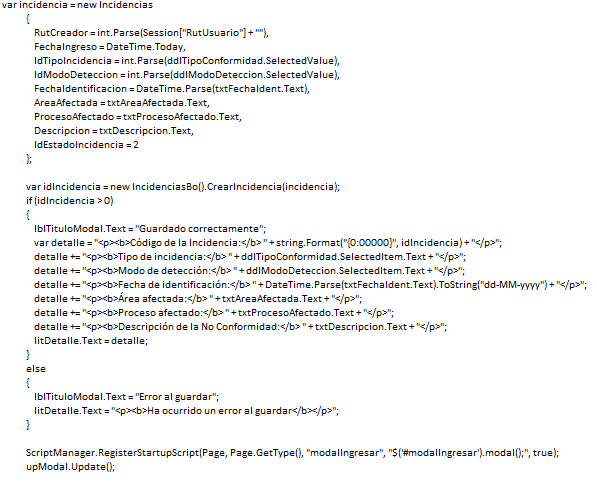
**Descripción:** Obtiene los datos ingresados en las cajas de texto del formulario y los compara a través de la capa de datos con la información del usuario, en caso de coicindir ambos campos (usuario y contraseña) redirecciona a la pagina MainModulos.aspx, en caso de error muestra un mensaje de error.

**ITEM:** IngresarNoConformidad.aspx

**Control de Formulario:** Button "btnIngresar"

**Evento:** OnClick="IngresarIncidencia"

**Código Fuente:**

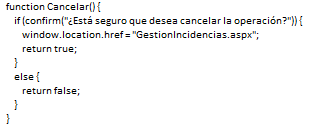


**Descripción:** Asigna los valores de los campos del formulario a las propiedades de las Incidencia, formateando los datos para que coincidan con los especificados en la base de datos, Ingresa la incidencia, en caso de que el proceso sea satisfactorio muestra una ventana con los detalles de la inserción, el caso de error, muestra el siguiente mensaje de error "Ha ocurrido un error al guardar".

**Control de Formulario:** Button "btnCancelar"

**Evento:** onclick="return Cancelar();"

**Código Fuente:**

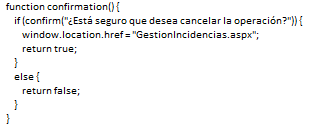


**Descripción:** Añade un archivo seleccionable desde el equipo, hacia el sistema, se asocia al número de análisis en caso de ser correctamente asociado, en caso de error muestra el siguiente mensaje "Ha ocurrido un error al guardar el documento".

**Control de Formulario:** Button "btnCancelar"

**Evento:** OnClientClick="return confirmation();"

**Código Fuente:**



**Descripción:** Validación Javascript, al ser presionado el botón Cancelar, la validación es activada, el código genera una ventana de confirmación, preguntando al Usuario si está seguro que desea cancelar la operación.

## CAPA DE ACCESO A DATOS

Es donde se escribe el código que habla con la base de datos. Es en esta capa donde se definen las consultas a la base de datos (el SQL). No realiza validaciones entre entidades, sólo las validaciones que van hacia la base de datos. La intención de esta capa es que el resto de la aplicación no se preocupe de los detalles (en cierta medida) de la estructura de la base de datos y trabaje a un nivel de objetos. Es la encargada de la entrada y salida de datos hacia y desde la base de datos. Ya que no debe de preocuparse de las reglas de negocio el código de la capa debería de ser fácil de seguir y de mantener. Las capas construidas encima de esta (es decir que usen esta capa) deben de dejar de pensar en tablas y registros para pensar en objetos y colecciones.

Estructura de la Capa

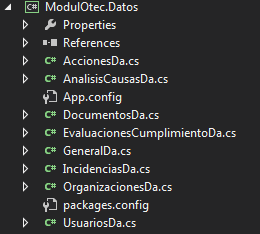
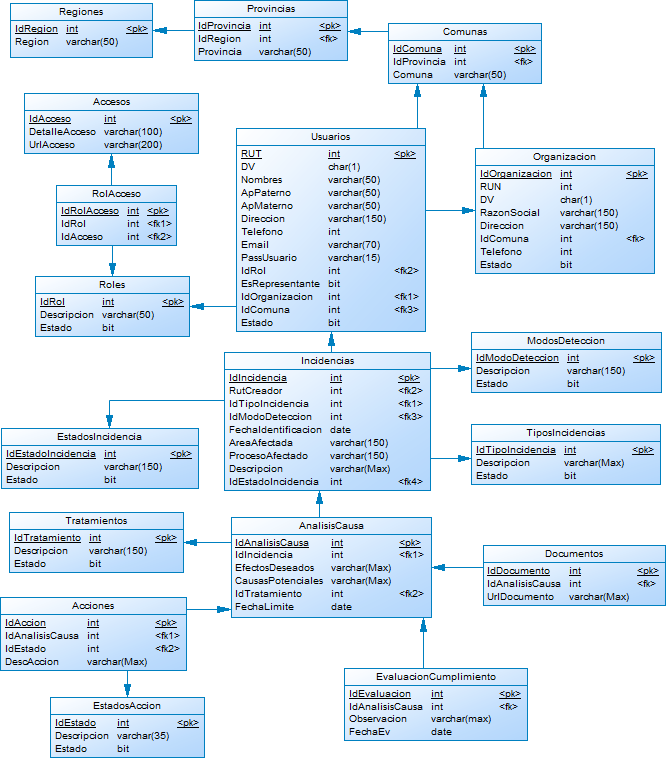


Ilustración 9 - Estructura capa de datos

**MODELO DE DATOS**

****

LENGUAJE DE MANIPULACIÓN Y DEFINICIÓN DE DATOS

Una vez creado el modelo de datos es necesaria la utilización de un lenguaje que nos permita la manipulación y el trabajo con la base de datos que almacenará la información.

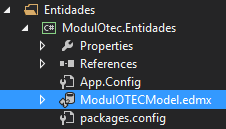
Este proyecto hace uso de SQL Server 2008 R2 para el almacenamiento de los datos, aplicando Lenguaje de Manipulación de Datos no procedural, justificado en la utilización del Framework Entity de Microsoft®.

**ENTITY FRAMEWORK**

“Entity Framework es la tecnología de acceso a datos de Microsoft recomendada para las nuevas aplicaciones.”

Se incorpora a este proyecto ModulOTEC Entity Framework dada su integración nativa con las tecnologías .NET, sus características permiten trabajar con datos en forma de objetos y propiedades específicos del dominio, como clientes y direcciones de cliente, sin tener que preocuparse por las tablas y columnas de la base de datos subyacente donde se almacenan estos datos. Con Entity Framework, se puede trabajar en un nivel mayor de abstracción al tratar con datos, y pueden crear y mantener aplicaciones orientadas a datos con menos código que en las aplicaciones tradicionales.

Se utiliza una importación al proyecto directamente desde VisualStudio el Modelo de Datos a utilizar a la capa de Entidades, evitando el trabajo con consultas sql desde la aplicación, esto gracias a Entity Framework.



CADENA DE CONEXIÓN SQL SERVER

<connectionStrings>

<add name="ModulOTECEntities" connectionString="metadata=res://\*/ModulOTECModel.csdl|res://\*/ModulOTECModel.ssdl|res://\*/ModulOTECModel.msl;provider=System.Data.SqlClient;provider connection string=&quot;**data source=localhost**;**initial catalog=ModulOTEC**;integrated security=True;MultipleActiveResultSets=True;App=EntityFramework&quot;" providerName="System.Data.EntityClient" />

</connectionStrings>

**Data source:** Referencia al servidor donde se aloja el sistema, en el caso específico de la etapa de desarrollo del sistema este se encuentra corriendo desde un servidor local, por ello se incorpora la cadena “localhost”.

**Initial catalog:** En este apartado se identifica la base de datos creada en el servidor con la que trabajará el sistema (ModulOTEC), debe ser consistente con el modelo de datos incorporado al momento del desarrollo.

**DEFINICIONES**

**LINQ:** Language-Integrated Query (LINQ) es un conjunto de características presentado en Visual Studio 2008 que agrega capacidades de consulta eficaces a la sintaxis de los lenguajes C# y Visual Basic.LINQ incorpora patrones fáciles y estándar para consultar y actualizar datos, y la tecnología se puede ampliar para proporcionar compatibilidad prácticamente con cualquier tipo de almacén de datos.Visual Studio incluye ensamblados de proveedor LINQ que habilitan el uso de LINQ con colecciones de .NET Framework, bases de datos de SQL Server, conjuntos de datos ADO.NET y documentos XML.

## CONCLUSION

De acuerdo al desarrollo o codificación del software, en el cual pasamos por las etapas de objetivos del proyecto, alcances, la capa de presentación, la capa de lógica de negocio y la capa de acceso de datos, podemos iniciar a crear las pruebas e implementación del sistema ModulOtec.