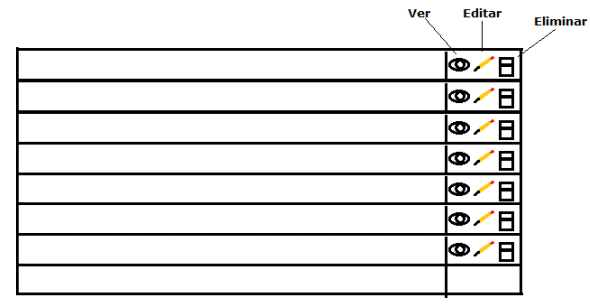
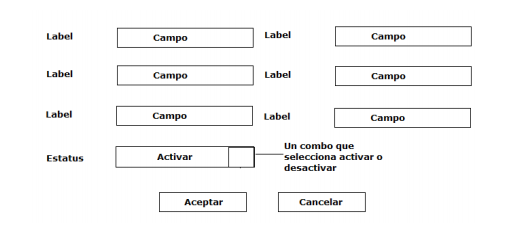
Estándares

1. Los témplate montados no deberán ser modificados por ningún grupo que no sea el integrador si tiene alguna queja o sugerencia por favor háganlo saber para cambiarlo.
2. La app se dividió en dos back office y front office por lo que no se puede referenciar algo hecho en uno con el otro se deberá copiar lo que se hizo.
3. En la carpeta de temp están los témplate cualquier página que no use el témplate no será integrada.
4. Las paginas serán guardadas por módulos en la carpeta Interfaz/web/nombremodulo.
5. Los css se guardaran en la carpeta css en común/resources.
6. Las imágenes se guardaran en la carpeta img en común/resources..
7. Los js se guardaran en la carpeta js en común/resources..
8. Al igual que las paginas en las carpetas mencionadas cada grupo deberá crear su carpeta y poner sus archivos css/nombremodulo , js/nombremodulo , img/nombremodulo.
9. Estándares en cuanto al código de las páginas:
   * 1. Todas las etiquetas deben estar en minúsculas.
     2. Primero se referenciara la clase de la etiqueta y luego el id propio cuando sea necesario. <etiqueta class=”” id=””></etiqueta>.
     3. Todas las etiquetas deben ser cerradas según corresponda <etiqueta></etiqueta> o en su defecto <etiqueta/>.
     4. En los js traten de ponen todas las acciones que usen en funciones separadas para que sea más ordenado (cuando sea posible).
     5. Se exige respetar la indentación para que el código sea más legible.
     6. No se aceptaran estilos que no estén en un css (no se puede poner en la etiqueta) a menos que sea totalmente necesario.
     7. No se aceptaran links de css, librerías y js en secciones que no estén creadas con ese propósito.
     8. Los componentes a utilizar serán los impuestos por el grupo integrador por lo que cuando creen un componente nuevo por ejemplo de botón <button class=”botón\_1”> deben llamar a la clase correspondiente.
     9. Estos estándares aplican para los componentes de asp que se usen también.
     10. En caso de necesitar un componente que no tenga estándar notificar al grupo integrador para crearlo de ser necesario o que ustedes lo creen y lo monten para todos (un componente que se reutilice).
10. Estándares en cuanto al diseño de las páginas:
    * 1. Todos los módulos deberán tener un título que se pone en la zona destinada para este.
      2. En caso de tener subtítulos estos deberán ser <h2></h2> y estarán centrados en el container.
      3. Todos deben usar las tablas impuestas.
      4. En caso de necesitar otro submenú en algún modulo notifíquelo para que sea agregado.
      5. En los gestionar del back\_office se podrán observar la lista de lo que se necesite y en caso de ser necesario se deberá agregar el modificar y eliminar como se puso en el ejemplo.
      6. Los botones solo se deberán cambiar de tamaño cuando sea necesario (que el texto lo amerite o en algún caso especial como los botones de menú).
      7. Nota: Lo que se menciona a continuación fueron estándares que se establecieron con la mayoría de los grupos del salón, las figuras que se muestran son para que tengan una idea de lo que tienen que realizar. Cualquier duda consultar al grupo integrador
      8. En la ventana de Consultar: Se mostrará un grid centrado que contendrá los datos a ser consultados, los campos que se muestren lo definirá cada grupo según su módulo (recomendamos que sean los datos más importantes), en cada registro que se muestre habrán 3 botones que ejecutarán las acciones de ver (Ojo), editar (Lápiz) y eliminar (Papelera), la paginación será de 8 registros máximos.
      9. Un ejemplo para que se guíen
11. El botón de editar llevará a otra ventana donde se mostrarán los campos a modificar, entre ellos se encontrará la opción de activar o desactivar dicho registro (esto si aplica en el módulo), esta ventana seguirá la siguiente estructura



Luego al presionar aceptar se mostrará un mensaje de confirmación, si ocurre un error se le informará al usuario y se dirigirá de nuevo a la ventana de Consultar. Si presiona cancelar se devolverá a la ventana de consultar.

Nota: Como se visualiza en la imagen sólo habrá 2 columnas de label y de campo.

1. El botón de ver llevará a otra ventana donde se podrá observar todos los campos del registro que se seleccionó, dicha ventana seguirá esta estructura:



Al presionar el botón volver se devolverá a la ventana de consultar.

1. El botón de eliminar mostrará un mensaje de confirmación, luego informará si hubo éxito o no

En la ventana de Agregar: Se mostrará la siguiente estructura

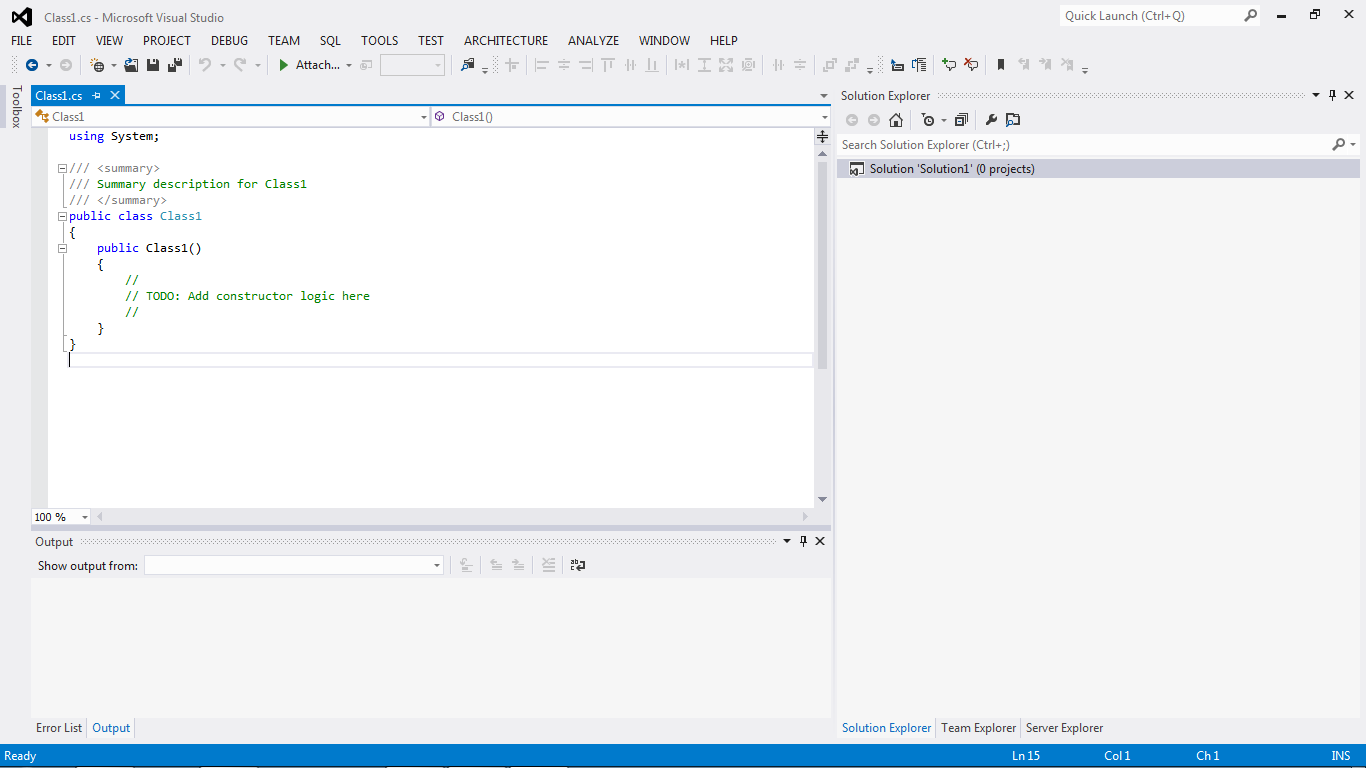


Al presionar aceptar aparecerá un mensaje de éxito como en el de la figura o uno de error con letras de color rojo, al realizar la acción se mantendrá en esta misma ventana.

Nota: Todo debe ir centrado.

Como se visualiza en la imagen sólo habrá 2 columnas de label y de campo.

La página principal la realizará el grupo con el módulo 1, es decir, La plantilla de login solo será usada por ellos, ya que se deben encargar de redireccionar al home del back-office y viceversa.

1. Validaciones de componentes:
   * 1. El label de mensajes se utilizara para las excepciones que puedan ocurrir en el sistema además de para informar al usuario del éxito en las operaciones por lo que debe estar debajo del título con el css .AvisoMensaje ver ErrorMensaje y ÉxitoMensaje que se debe agregar dinámicamente como segunda clase. este label debe estar en TODAS las páginas ya que en caso de ocurrir una excepción en alguna se debe pintar allí el texto explicando que paso.
     2. Los validator son los que se usaran para validar los campos y el valitation summary debe tener el css avisomensajeBot y como segunda clase ErrorMensaje. la tabla en donde este estará el validator debe estar centrada (intenten quede lo más centrado posible ver la página de agregar rol que fue modificada por nosotros como ejemplo) y debe estar encima del botón de aceptar.
     3. El botón de Buscar debe decir Buscar y debe estar en otra línea cada parámetro (campo) debe también estar en una línea distinta.
     4. Todas las tablas de los Buscar deben tener el estilo css que se montó y como máximo se mostraran 20 resultados por página. (ver página de usuarios internos para guiarse de cómo hacer la tabla).
     5. El calendario a usar con un textbox con la propiedad TextMode=”Date” (compatible con Chrome, opera, safary) y desplega el calendario solo (el css es el de textbox igual) En caso de ser solo hora usar el que tiene el módulo de servicios de playa.
     6. Los textfield deben estar validados para evitar HTML injection, SQL injection esto se hará con expresiones regulares y con queries parametrizados al final los links con ejemplos de cómo usarlos ustedes deben profundizar más.
     7. Deben validar la entrada del usuario en caso de que el campo sea solo número, número y texto, solo texto entre otros.
     8. Los botones y links deben desactivarse (enabled=false) luego de que el usuario haga clic en este para evitar que mande múltiples peticiones. El botón o link estará desactivado hasta que llegue el resultado de la bd o que se cambie de página.
     9. Los campos de fecha deben estar inactivos la fecha se manejara con el calendario que se creó (se montara hoy en la noche 24/11).
     10. La fecha se manejara con dd/mm/yyyy ejemplo 10/04/1992.
     11. El grupo 8 acordó subir un ejemplo de los componentes que se usan para validaciones de la manera que dijimos anteriormente.
2. Base de datos:
   * 1. TODO query debe estar en un procedimiento o función DENTRO de la base de datos no se aceptaran query’s afuera.
     2. El estándar para el nombre del procedimiento o función es Procedure\_nombre o Function\_nombre el nombre debe describir lo que hace por ejemplo Procedure\_insertarPersona usando minusculasMayúsculas para diferenciar cada palabra con estándar lowerCamelCase
     3. El estándar para las variables será \_lowerCamelCase
     4. Deben hacer sus inserts al menos 5 por tabla exponencial es decir si A es débil con respecto a B entonces B tendrá 5 registros y A tendrá 25 ,5 por cada registro de B.
     5. Para poder eliminar un registro, se debe eliminar todo lo que esté relacionado con este ya que no existe la condición On Delete Cascade.
     6. Los atts se deben guardar tal cual el usuario los ponga pero a la hora de hacer la consulta se debe usar UPPER en los dos lados para la comparación.
     7. La longitud máxima para los atributos de tipo varchar es de 100 caracteres. Esto NO quiere decir que todos a juro deben colocar como máxima longitud para escritura 100 caracteres, sino que en la BD estarán reservados como mucho 100.
     8. El proceso para el manejo de sus procedimientos y funciones es el siguiente:
        1. Implementan sus métodos a través del .MDF importado en su visual estudio (ver el tutorial en drive).
        2. Para subir sus métodos al repositorio, deben colocar la implementación de todos sus métodos en un único .sql en la carpeta denominada StoreProcedures del repositorio.
        3. El estándar para el nombre del .sql a subir será nombreDelModulo
3. Estándares de programación:
   * 1. En la capa de Interfaz solo se pueden tener los webforms y cs vinculados que ya fueron entregados.
     2. En la capa Data estarán los llamados a base de datos.
     3. En la capa de Logica estar el Core de la app es decir estarán las clases que controlara la aplicación.
     4. En la capa de excepciones se pondrán las excepciones que cada uno necesite.
     5. En la capa de Dominio debe ir las clases de dominio de la app.
     6. En la carpeta lib pueden poner los dll que necesiten para luego hacerles la referencia.
     7. En cada capa debe hay una carpeta con el nombre de su módulo que es donde se guardaran sus clases.
     8. Los mensajes de confirmación YA NO SE HARAN CON JS se harán usando las configuraciones para los botones de asp.
     9. El calendario a usar es el proporcionado por el grupo integrador.
     10. Los mensajes de confirmación error y éxito deberán tener el formato que dio el grupo integrador.
     11. El estándar para las variables es \_nombre notación lowerCamelCase
     12. Todas las variables globales deben tener getter y setter
     13. Todas las clases deben tener su constructor
     14. El estándar de nombre de clases es primera letra mayúscula lo demás en minúsculas ejemplo Usuario.
     15. Las llaves se deben iniciar debajo del nombre ejemplo: 
4. Todos los metodos y funciones deben estar comentados como esta en el siguiente link:<http://msdn.microsoft.com/es-es/library/aa288481(v=vs.71).aspx>
5. Se debe respetar la indentacion como esta en la imagen (tabulador default 4 espacios).
6. Todas las variables internas de los metodos deben declararse al principio de este a no ser que sea un caso especial como por ejemplo un bucle en tal caso se declarara al principio de este solo una variable declarada por linea.
7. Un metodo debe tipicamente tener entre 1 a 25 lineas como maximo. Si un metodo tiene mas de 25 lineas de codigo , debes considerar refactorizarlo en metodos separados
8. Se deben usar los tipos especificos de C# en vez de los tipos definidos en el espacio de nombres System(que resalte en azul oscuro y no en azul claro) por ejemplo:
   * 1. int \_edad; (no Int)
     2. string \_nombre; (no String)
9. Reglas Generales:
   * 1. Todas las clases en las que se haga push deben tener sus pruebas unitarias respectivas a cada método.
     2. Todas las clases de su módulo deben estar en la carpeta dedicada a este en caso de ocurrir convergencia entre clases en varios módulos ustedes deben decidir quién la desarrollara y se usaran regiones para que ambos la puedan usar.
     3. No se debe hacer push de cosas no se han probado
10. Links de referencias que pueden ser de utilidad:
    * 1. ejemplo de query parametrizado en asp.net: <http://www.w3schools.com/sql/sql_injection.asp>
      2. Manual de expresiones regulares: <http://msdn.microsoft.com/en-us/library/hs600312(v=vs.110).aspx>
11. Mensajes de aviso:
    * 1. Creación:
12. Exitosa: ‘El ítem ha sido creado satisfactoriamente’.
13. No exitosa: ‘El ítem no ha podido ser creado debido a que X’.
14. X puede ser:
    * + - 1. Los datos ingresados son inválidos.
          2. Ya está registrado en el sistema. (también aplica para los registrados en modo desactivado)
          3. Existe un conflicto con la conexión a la base de datos.
          4. Faltan datos por ingresar.
          5. Alguna otra condición de fracaso.
      1. Eliminación:
         1. Exitosa: ‘El ítem ha sido eliminado satisfactoriamente’.
         2. No exitosa: ‘El ítem no ha podido ser eliminado debido a que X’.
         3. X puede ser:
            1. Está referenciado en uno o más registros de otra u otras tablas.
            2. Existe un conflicto con la conexión a la base de datos.
            3. Alguna otra condición de fracaso.
      2. Modificación:
         1. Exitosa: ‘El ítem ha sido modificado satisfactoriamente’.
         2. No exitosa: ‘El ítem no ha podido ser modificado debido a que X’.
         3. X puede ser:
            1. Los datos ingresados son inválidos.
            2. Ya está registrado el ítem en el sistema.
            3. Existe un conflicto con la conexión a la base de datos.
            4. Alguna otra condición de fracaso.
      3. Consulta
         + 1. Exitosa: muestra los datos.
           2. No exitosa: ‘No se han podido cargar los datos debido a que X’
           3. X puede ser:

Existe un conflicto con la conexión a la base de datos.

Alguna otra condición de fracaso.

* + 1. Inicio de sesión
       1. Exitoso: muestra la página después de login exitoso.
       2. No exitoso: ‘No ha podido iniciar sesión debido a que X’
       3. X puede ser:
          1. Los datos ingresados son incorrectos (correo y/o contraseña).
          2. Existe un conflicto con la conexión a la base de datos.
          3. Alguna otra condición de fracaso.
    2. Activación
       1. Exitosa: ‘El ítem ha sido activado satisfactoriamente’
       2. No exitosa: ‘El ítem no ha podido ser activado debido a que X’
       3. X puede ser:
          1. Existe un conflicto con la conexión a la base de datos.
          2. Alguna otra condición de fracaso.
    3. Desactivación

Exitosa: ‘El ítem ha sido desactivado satisfactoriamente’

No exitosa: ‘El ítem no ha podido ser desactivado debido a que X’

X puede ser:

Existe un conflicto con la conexión a la base de datos.

Alguna otra condición de fracaso. (puede ser que este ítem este siendo referenciado por uno o más registro de otra u otras tablas).

1. Mensajes de Confirmacion:
   * + - 1. Eliminar: ‘Esta acción eliminará el ítem permanentemente del sistema ¿desea continuar?’
         2. Modificar: ‘Esta acción modificará el ítem en el sistema ¿desea continuar?’
         3. Desactivar: ‘Esta acción desactivará el ítem en el sistema hasta que decida activarlo nuevamente ¿desea continuar?’
     1. Ejemplos:
        + 1. Mensaje de aviso de eliminación:
          2. El ítem no ha podido ser eliminado debido a que está referenciado en uno o más registros de otra u otras tablas.
          3. Mensaje de aviso de inicio de sesión no exitoso:
          4. No ha podido iniciar sesión debido a que los datos ingresados son incorrectos (correo y/o contraseña).
          5. NOTA: si alguno considera que hace falta algún otro mensaje, expóngalo en el grupo de WhatsApp. Se agradece la colaboración de cada grupo en este aspecto.