

MANUAL TÉCNICO

PORTAL PARA EL CENTRO DE CAPACITACIÓN CONTINUA DE LA FICA

INSTALACIÓN

Es necesario instalar Linux Red Hat 8.0 como servidor, agregando los siguientes paquetes mínimos al momento de la instalación:

Desktops:

- Sistema X Window
- Entorno GNOME

Applications:

- Editores (Emacs, vi)
- Internet gráfica

Servers

- Herramientas de configuración del servidor
- Servidor Web
- Servidor de nombres DNS (BIND)
- Servidor de bases de datos
- Servidor FTP
- Servidor de red

Development:

- Herramientas de desarrollo (automake, gcc, perl, pitón y depuradores)
- Desarrollo de software X (aplicaciones para X window)
- Desarrollo de software de GNOME (desarrollar GTK+)

System

- Herramientas de administración (cuentas de usuario, gestión)
- Herramientas del sistema

- Soporte para impresión

Es muy importante instalar la parte de *Development*, por cuanto vamos a instalar Mono desde su código fuente, y es necesario compilar las librerías utilizando por ejemplo los compiladores de C++ (*gcc*). Si instalamos Mono utilizando los archivos *.rpm* al momento de instalar las aplicaciones nos indicará las dependencias necesarias.

Una vez que se tiene instalado la versión Red Hat 8.0 de Linux, e instalado la base de datos MySQL, debemos tener los instaladores (código fuente) de Mono que son los siguientes:

mono-0.28.tar.gz

mcs-0.28.tar.gz

xsp-0.5.tar.gz

Primeramente abrimos una ventana terminal, donde ejecutaremos todos los comandos.

Empezamos con *mono-0.28.tar.gz* (librerías de Mono).

- Copiar los instaladores a la ubicación en el disco duro:

/home/pablo/mono/

- Descomprimir el archivo con el comando:

tar xvfz mono-0.28.tar.gz

Se crea el directorio *mono-0.28*

- Ingresar al directorio

/home/pablo/mono/mono-0.28

- Teclar:

./configure

- Teclar:

make

- Teclar:

make install

Luego seguimos con *mcs-0.28.tar.gz* (compilador de C#, se compila a si

mismo)

- Copiar los instaladores en la ubicación en el disco duro:

/home/pablo/mono/

- Descomprimir el archivo con el comando:

tar xvfz mcs-0.28.tar.gz

, se crea el directorio *mcs-0.28*

- Ingresar al directorio:

/home/pablo/mono/mcs-0.28

- Teclar:

make

Por último *xsp-0.5.tar.gz* (pequeño servidor Web, necesario para ejecutar páginas Web asp.net)

- Copiar los instaladores en la ubicación en el disco duro:

/home/pablo/mono/

- Descomprimir el archivo con el comando:

tar xvfz xsp-0.5.tar.gz

, se crea el directorio *xsp-0.5*

- Ingresar al directorio:

/home/pablo/mono/xsp-0.5

- Teclar:

make

- Teclar:

make install

Nos dirigimos a la ubicación de instalación de *xsp-0.5* (ejemplo: */home/pablo/mono/xsp-0.5*):

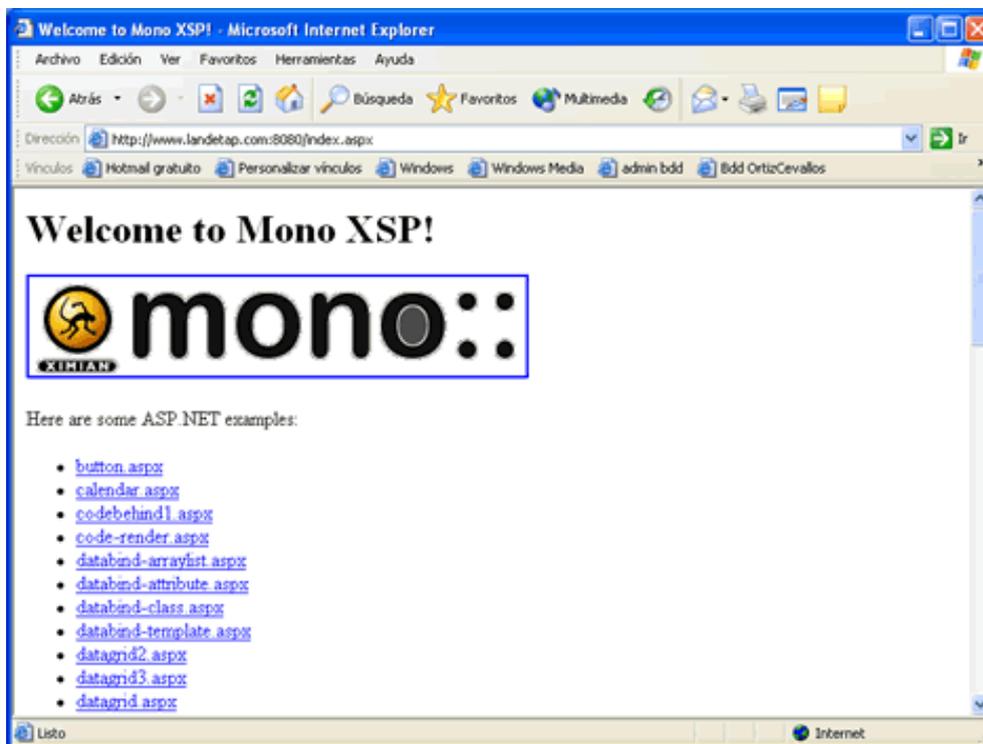
- Ingresar al directorio *server*

- Ingresar al directorio *test*

- Teclar: *mono xsp.exe*

- En el explorador poner: *http://127.0.0.1:8080/*

- Debemos obtener la página de bienvenida de Mono y vínculos a algunos ejemplos de páginas Web ASP.NET



Para parar el servidor mono, en la ventana terminal donde está corriendo presionamos *control + C*.

Si se desea trabajar con una dirección IP ficticia con el fin de crear una aplicación ASP.NET que funcione en red, es necesario configurar el servidor Linux con la dirección IP deseada e incluso si se desea, se puede configurar un dominio deseado.

Para saber como configurar el servidor Linux por favor revise un manual de administración de Linux

CONFIGURACIÓN DE ARCHIVOS INICIALES

Una vez que se tiene instalado *mono-0.28* y el servidor Web *xsp-0.5* que viene con la distribución de Mono, y que hemos configurado el servidor Linux con una dirección IP específica, debemos configurar el servidor XSP.

Para esto, existe un archivo llamado ***xsp.exe.config***, el cual se sitúa en:

/home/pablo/mono/xsp-0.5/server/test/

Es un archivo *xml*, en el cual especificamos la dirección IP del servidor donde se alojan las páginas Web, el puerto al cual escucha, etc. Debemos modificar su código para que se adapte a la configuración del servidor:

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <appSettings>
    <add key="MonoServerPort" value="8080"/>
    <add key="MonoServerAddress" value="172.20.3.24"/>
    <add key="MonoServerRootDir" value="/home/pablo/mono/xsp-
0.5/server/test"/>
    <!--      MonoServerVirtualDir
      MonoServerRootDir
      MonoServerAddress -->
    <!-- MonoServerDefaultIndexFiles is set in web.config -->
  </appSettings>
</configuration>
```

El archivo de configuración inicial del sitio Web, se llama ***web.config***, y también se sitúa en:

/home/pablo/mono/xsp-0.5/server/test/

En este archivo, se debe buscar una sección muy importante para la aplicación. Esta sección es la configuración del proveedor nativo de base de datos. El código es el siguiente:

```
<appSettings>
    <add key="MonoServerDefaultIndexFiles"
        value="index.aspx, Default.aspx, default.aspx, index.html,
index.htm" />
    <add key="DBProviderAssembly"
        value="ByteFX.Data"/>
    <add key="DBConnectionType"
        value="ByteFX.Data.MySQLClient.MySQLConnection"/>
    <add key="DBConnectionString"
        value=""/>
</appSettings>
```

Esta porción de código indica las extensiones para la página inicial, en nuestro caso se especifica que la página inicial puede ser *.aspx* o *html* o *htm*.

Luego existe la etiqueta *add key="DBProviderAssembly"*. Tiene como valor a *ByteFX.Data*, que es el nombre del proveedor para la base de datos MySQL en Mono, que es la que utilizamos en la presente aplicación.

Luego se tiene el tipo de conexión que se utiliza. El valor para este parámetro es *ByteFX.Data.MySQLClient.MySQLConnection*

Por último tenemos el parámetro *DBConnectionString* que es la cadena de conexión que se utilizará para conectarse a la base de datos. Es posible llenar este campo, sin embargo, se ha dejado vacío, pero luego se explicará la cadena de conexión que se debe ingresar en ciertos archivos propios de la aplicación. Esto se hizo, debido a que puede existir otra aplicación que se suba al servidor y que utilice otro tipo de base de datos, en ese caso igualmente se debe configurar ciertos archivos propios de la aplicación.

COPIA DE ARCHIVOS RELACIONADOS CON LA APLICACIÓN AL SERVIDOR

La aplicación del Portal para el Centro de Capacitación Continua de la FICA esta dividida las siguientes partes:

- 1. Archivos realizados en lenguaje ASP.NET y C#**
- 2. Archivos realizados en lenguaje Php**
- 3. Archivos de la Base de Datos**

1. Archivos en lenguaje ASP.NET y C#

Los archivos de la aplicación que están hechos en C# y ASP.NET se encuentran en la ruta del CD:

//Aplicativo/Sitio Web/ASP NET

En esta carpeta encontraremos:

1. carpeta *Templates* : contiene un archivo (*estilos.css*) que es un archivo de hojas de estilo cascada
2. carpeta *images* : contiene todas las imágenes que están insertadas en las páginas ASP.NET
3. archivos de ASP.NET: son de extensión *aspx* y pueden contener código C# así como también código html y ASP.NET
4. archivos con código C#, los cuales tienen la extensión *inc* y contienen funciones en código C# necesarios para la aplicación.
5. Un archivo de configuración *web.config*, el cual es un archivo con código *xml* que especifica el modo de autenticación, etc.

Su código es el siguiente:

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <system.web>
    <authentication mode="Forms">
      <forms name=".MONOAUTH" loginUrl="index.aspx"/>
    </authentication>
    <authorization>
      <deny users="" />
    </authorization>
  </system.web>
  <location path="administracion.aspx">
    <system.web>
      <authorization>
        <deny users="" />
      </authorization>
    </system.web>
  </location>
</configuration>
```

En este archivo se indica el modo de autenticación. Nuestro sitio utiliza la autenticación basada en formularios. Declara un formulario que indica la página en la cual se tiene la autenticación (*index.aspx*). Luego se especifica la página que es acceso restringido para usuarios normales, en nuestro caso es *administración.aspx*.

Además tiene una sección de denegación de usuarios, que en nuestro caso es ninguno, lo que significa que cualquiera puede autenticarse e ingresar como un usuario del sistema.

Todo el contenido de la ruta del CD: *//Aplicativo/Sitio Web/ASP NET* se debe copiar en el servidor de una de las siguientes formas:

- Si se desea que la dirección URL sea:

`http://172.20.3.24:8080`

, entonces se deben copiar los archivos en:

`/home/pablo/mono/xsp-0.5/server/test/`

en este caso se debe tomar en cuenta que existen dos archivos llamados *web.config* y que al situar todo el sitio Web en la ruta especificada, se tendrá que copiar el contenido del más pequeño al más grande fijándose en las etiquetas que lleva.

- Si se desea que la dirección URL sea
`http://172.20.3.24:8080/capacitacion`
, entonces se deben copiar los archivos en:
`/home/pablo/mono/xsp-0.5/server/test/capacitacion/`

2. Archivos en lenguaje Php

Como reporte técnico, se debe comenzar diciendo que se tuvo que incluir código Php en la aplicación, debido a que la versión de Mono 0,28 no permitía todavía el envío de datos por medio de formularios. Por tanto, el lenguaje Php se utilizó en la presente aplicación única y exclusivamente para permitir el envío de datos (documentos, imágenes y animaciones) al servidor por medio de páginas web.

Los archivos de la aplicación que están hechos en Php se encuentran en la ruta del CD:

//Aplicativo/Sitio Web/PHP

En esta carpeta encontraremos:

1. carpeta *animaciones*: contiene los archivos ejecutables de flash (formato *swf*) que corresponden a 2 archivos que se tienen: el uno en la página inicial y el otro cuando un usuario inicia la sesión .
Los archivos que se suben al servidor desde la página de Parámetros Generales – Cambiar animación página principal y Parámetros Generales – Cambiar animación página bienvenida (consultar el manual de usuario administrador) se sitúan en esta carpeta.
Puede haber varios, pero solo dos estarían siendo utilizados.
2. carpeta *documentos*: contiene todos los documentos que el administrador o los instructores suben al servidor cuando se crea un nuevo curso.
Puede haber varios y sus extensiones pueden ser: *pdf, xls, doc, zip* y *ppt*

3. carpeta *images*: contiene los banners o imágenes que sirven de vínculos a sitios de interés y que se sitúan en el marco izquierdo del portal.

Puede haber muchos y sus extensiones pueden ser *gif* o *jpg*

4. archivos con código PHP de extensión *php* y *phtml*, los cuales son utilizados en la aplicación para subir documentos, imágenes y animaciones al servidor desde las páginas de administración del portal.

Todo el contenido de la ruta del CD: *//Aplicativo/Sitio Web/PHP* se debe copiar en la ruta del servidor:

/var/www/html/upload

La carpeta *upload* debe ser creada previamente.

3. Archivos de la Base de Datos

Los archivos de la base de datos se encuentran en la ruta del CD:

//Aplicativo/Base de Datos

En esta carpeta encontraremos la base de datos física que utiliza la aplicación. Consiste en una carpeta de nombre *ccc* que contiene tres archivos por cada tabla que se ha creado en la base de datos.

Toda la carpeta *ccc* debe ser copiada a la ruta del servidor:

/var/lib/mysql/

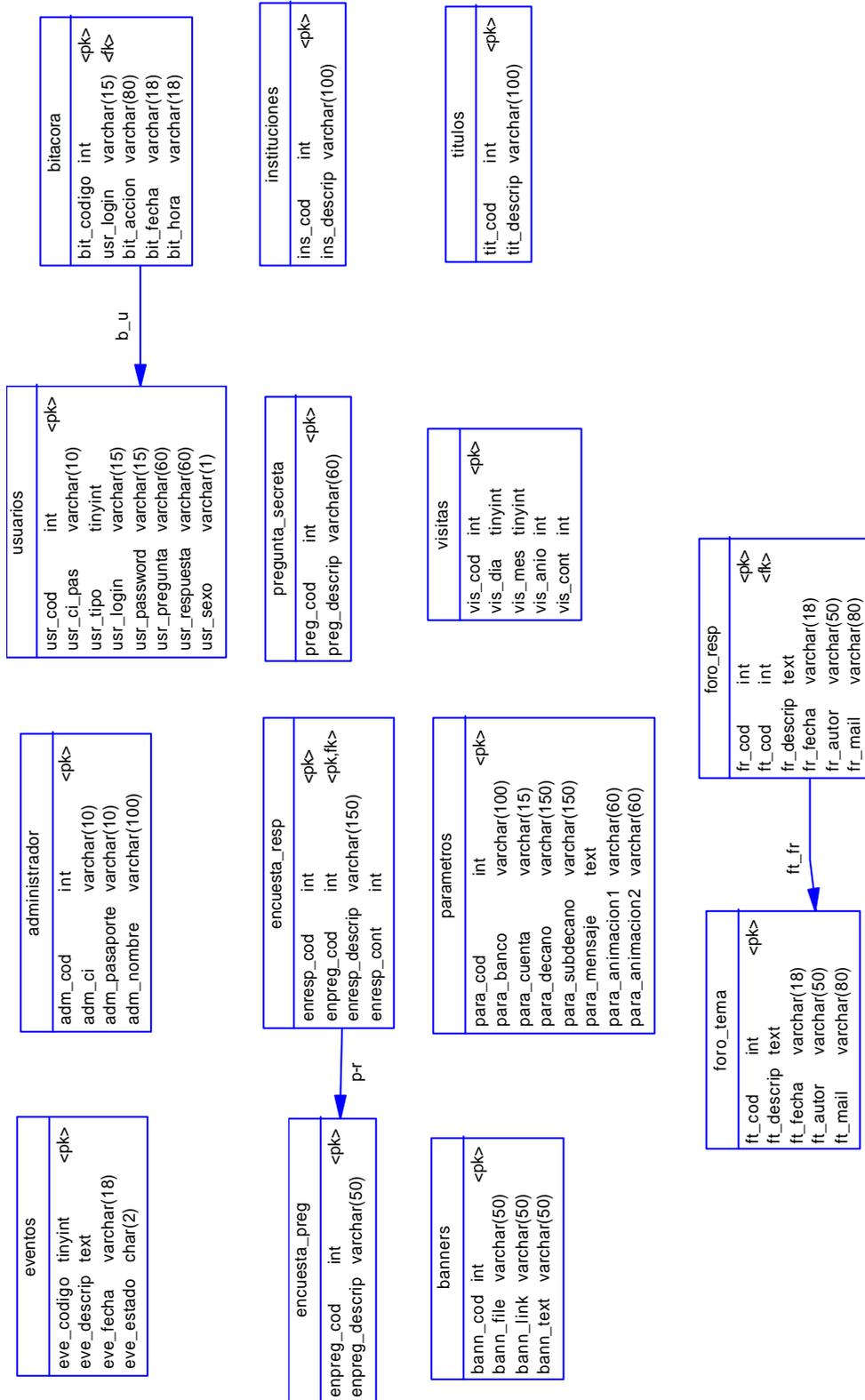
Para que la base de datos funcione bien, es necesario cambiar de dueño y los permisos de los archivos que contiene la carpeta *ccc* así como de la carpeta en general.

La carpeta `ccc` debe pertenecer al dueño `mysql` y se deberá asignar los siguientes permisos:

- lectura por el dueño
- escritura por el dueño
- ejecución por el dueño

Ingresamos a la carpeta `ccc` y verificamos que todos los archivos que están dentro de la carpeta `ccc` pertenezcan al dueño `mysql` y se deberán asignar los siguientes permisos:

- lectura por el dueño
- escritura por el dueño
- lectura por el grupo
- escritura por el grupo



ESTADO INICIAL DE LA BASE DE DATOS

La base de datos que se encuentra en el CD, contiene la información básica para iniciar su funcionamiento, de esta forma, sólo las siguientes tablas contienen datos iniciales:

Tabla	Datos
administrador	Contiene 1 registro de un administrador
banners	Contiene 2 registros: el primero la información del vínculo a sitio Web de la FICA y el segundo la información del vínculo a Google
parametros	Contiene 1 registro con los nombres del Decano y Subdecano actuales, el número de cuenta corriente actual del Banco del Pacífico y la información de las 2 animaciones necesarias para visualizar la página principal
pregunta_secreta	Contiene 4 registros de cuatro preguntas secretas para que los usuarios empiecen a registrarse
usuarios	Contiene 1 registro, información del primer administrador. <i>Usuario: administrador</i> <i>Contraseña: admin</i>

MODIFICACIÓN DE LA CADENA DE CONEXIÓN A LA BASE DE DATOS

En la instalación se debió instalar MySql y se debe crear el siguiente usuario de MySql con privilegios de administrador de MySql:

Usuario: *mono*

Contraseña: *mono*

Los siguientes archivos propios de la aplicación necesitan ser modificados para configurar la cadena de conexión que ya se encuentra ingresada:

funcion_tesis.inc

funcion_tesis2.inc

funcion_tesis4.inc

La cadena de conexión que se incluye en estos archivos es la siguiente:

```
cncString = "Server=172.20.3.24;User  
ID=mono;Password=mono;Database=ccc";
```

Esta línea significa, que como nombre del host se tiene a *172.20.3.24*, se conecta a la base de datos *ccc* y que se tiene como nombre de usuario a *mono* y como contraseña a *mono*.

Por tanto, dependiendo de la configuración que se haya realizado en el servidor, podríamos tener que modificar en la cadena de conexión: la dirección IP del servidor, el nombre de usuario de MySql y/o la contraseña de MySql.

Eso es todo lo que se tiene que modificar; ahora se explicará un poco como es el funcionamiento de la conexión a la base de datos:

En los archivos (*funcion_tesis.inc*, *funcion_tesis2.inc* y *funcion_tesis4.inc*) se realiza la conexión con la base de datos de la siguiente forma:

Empieza con la verificación de los parámetros ingresados en el archivo de configuración *web.config*, esto lo hace con el siguiente código en C#:

```
void GetConnectionData (out string providerAssembly, out string
cncTypeName, out string cncString)
{
    providerAssembly = null;
    cncTypeName = null;
    cncString = null;
    NameValueCollection config = ConfigurationSettings.AppSettings as
NameValueCollection;
    if (config != null) {
        foreach (string s in config.Keys) {
            if (0 == String.Compare ("DBProviderAssembly", s, true)) {
                providerAssembly = config [s];
            } else if (0 == String.Compare ("DBConnectionType", s, true)) {
                cncTypeName = config [s];
            } else if (0 == String.Compare ("DBConnectionString", s, true)) {
                cncString = config [s];
            }
        }
    }
    if (cncString == null || cncString == "")
        cncString = "Server=172.20.3.24;User
ID=mono;Password=mono;Database=ccc";
}
```

Como podemos apreciar, al final de la función compara la cadena de conexión en *web.config* y como no existe le asignamos la cadena de conexión que utilizaremos.

La siguiente función en C# registra a pantalla los errores que pueden producirse en caso de que falle la conexión con la base de datos:

```
void ShowError (Exception exc)
{
    noDBLine.InnerHtml += "<p><b>El Error fue:</b>\n<pre> " + exc +
"</pre><p>";
    theForm.Visible = false;
    noDBLine.Visible = true;
}
```

Por último es necesario cargar los ensamblados del proveedor de base de datos *ByteFX.Data*, para lo cual realiza lo siguiente (código en C#):

```
IDbConnection cnc;
void Page_Init (object sender, EventArgs e)
{
    string connectionTypeName;
    string providerAssemblyName;
    string cncString;

    GetConnectionData (out providerAssemblyName, out
connectionTypeName, out cncString);
    if (cncType == null) {
        Assembly dbAssembly = Assembly.Load
(providerAssemblyName);
        cncType = dbAssembly.GetType (connectionTypeName, true);
        if (!typeof (IDbConnection).IsAssignableFrom (cncType))
            throw new ApplicationException ("The type " + cncType +
            " does not implement IDbConnection.\n" +
            "Check 'DbConnectionType' in server.exe.config.");
    }

    cnc = (IDbConnection) Activator.CreateInstance (cncType);
    cnc.ConnectionString = cncString;
    try {
        cnc.Open ();
    } catch (Exception exc) {
        ShowError (exc);
        cnc = null;
    }
}
```

Con esto se tiene descrito la parte correspondiente a la conexión entre Mono y MySQL.

MODIFICACIÓN DE OTROS ARCHIVOS

Es necesario cambiar los direccionamientos a los archivos que fueron copiados en la ruta */var/www/html/upload*

Simplemente es necesario cambiar todo lo que se encuentre como *www.landetap.com* por la dirección IP que será asignada en el servidor (ejemplo: *172.20.3.24*). Esto se debe hacer en los siguientes archivos de la ruta */home/pablo/mono/xsp-0.5/server/test/capacitacion*

Archivo	Coincidencias a cambiar
<i>ad_parametros.aspx</i>	3
<i>admin_curso2.aspx</i>	6
<i>banners.aspx</i>	1
<i>funcion_tesis.inc</i>	3
<i>inicial.aspx</i>	4

Se debe cambiar lo que se encuentre como *http://www.landetap.com:8080/tesis2/* por la dirección URL de nuestra aplicación (será *http://172.20.3.24:8080* o *http://172.20.3.24/capacitacion/*) de acuerdo a como hayamos creado las carpetas al momento de copiar los archivos (ver sección COPIA DE ARCHIVOS RELACIONADOS CON LA APLICACIÓN AL SERVIDOR) en los siguientes archivos de la ruta */var/www/html/upload/*

Archivo	Coincidencias a cambiar
<i>animas.php</i>	1
<i>docus.php</i>	1
<i>index.php</i>	1

En la misma ruta, en el archivo *funcion.phtml*, en la línea:

```
mysql_connect("ns.landetap.com","mono","mono")
```

, debe ser cambiado el nombre del servidor (poner dirección IP *172.20.3.24*) y también se podría cambiar el usuario y contraseña de MySQL si se han especificado otros.

PROBAR EL FUNCIONAMIENTO

Si se han seguido los pasos explicados anteriormente de una manera correcta, entonces ya se podrá empezar a utilizar todas las funcionalidades del Portal para el Centro de Capacitación Continua de la FICA.

Abrimos una ventana terminal y ejecutamos el comando:

```
mono /home/pablo/mono/xsp-0.5/server/test/xsp.exe
```

, con lo cual, en la misma pantalla terminal debemos obtener:

```
Listening on port:8080
```

```
Listening on ardes: 172.20.3.24
```

```
Root directory: /home/pablo/mono/xsp-0.5/server/test
```

```
Virtual directory: /
```

En cualquier computador que este conectado a la misma red que el servidor, abrimos cualquier browser (ejemplo: Internet Explorer) y en la barra de dirección escribimos:

```
http://172.20.3.24:8080/capacitacion
```

, y obtendremos:



Recuerde que la base de datos esta con los datos básicos para iniciar su funcionamiento. El administrador debe empezar a ingresar los datos para su uso óptimo.

EXPLICACIÓN DEL ACCESO A LOS DATOS (CONSULTAS, INGRESOS, ACTUALIZACIONES)

Los archivos *funcion_tesis.inc*, *funcion_tesis2.inc* y *funcion_tesis4.inc*, a más de contener las funciones para conectarse con la base de datos, también contienen las funciones utilizadas en la aplicación para acceder a la base de datos.

Consultas

El archivo *funcion_tesis.inc* contiene la mayoría de funciones de consulta de la aplicación.

Para crear una consulta a la base de datos, básicamente es necesario lo siguiente (en C#):

Crear el comando ADO.NET que realizará la consulta. Este comando se asocia al namespace *IDbCommand*.

```
IDbCommand selectCommand = cnc.CreateCommand();
```

Luego creamos un lector (*reader*) para la base de datos.

```
IDataReader reader;
```

Especificamos la sentencia de consulta en código sql:

```
string selectCmd = "SELECT cur_codigo FROM cursos";
```

Asociamos el comando de lectura que creamos anteriormente con la sentencia de consulta sql:

```
selectCommand.CommandText = selectCmd;
```

Ejecutamos la lectura de los datos:

```
reader = selectCommand.ExecuteReader ();
```

Por ultimo, en un ciclo repetitivo tendremos los resultados:

```
while (reader.Read ()) {  
.....  
}
```

Para acceder a un determinado campo de la consulta utilizamos:

```
reader.GetValue (0).ToString ()
```

, donde 0 representa el número del campo requerido.

De esta manera podremos realizar operaciones con los resultados o simplemente mostrarlos en controles ASP.NET

A continuación tenemos un ejemplo de esto con la función *MostrarCursos()* realizada en lenguaje C#, donde se realiza una consulta a la base de datos y se saca los resultados en controles ASP.NET (*label, hyperlink, table*)

```
void MostrarCursos (string seminario)  
{  
    IDbCommand selectCommand = cnc.CreateCommand();  
    IDataReader reader;  
    string selectCmd = "SELECT cur_codigo, cur_nombre, cur_descrip FROM  
cursos where cur_estado='0' and cur_seminario=" + seminario;  
    selectCommand.CommandText = selectCmd;  
    reader = selectCommand.ExecuteReader ();  
    Label n = new Label(); n.Text = "NOMBRE";  
    Label d = new Label(); d.Text = "DESCRIPCION";  
    TableRow r = new TableRow ();  
    TableCell c = new TableCell (); c.Controls.Add(n); r.Cells.Add(c);  
    TableCell p = new TableCell (); p.Controls.Add(d); r.Cells.Add(p);  
    myTable.Rows.Add (r);  
    while (reader.Read ()) {  
        TableRow row = new TableRow ();  
        for (int i = 1; i < reader.FieldCount; i++) {  
            TableCell cell = new TableCell ();  
            if (i==1){ //si es cur_nombre le pone un link  
                HyperLink link = new HyperLink();  
                link.Text = reader.GetValue (i).ToString ();  
                link.NavigateUrl = "cursos2.aspx?curso=" + reader.GetValue  
(0).ToString () + "&sem=" + seminario;  
                cell.Controls.Add (link);  
            }  
            else  
                cell.Controls.Add (new LiteralControl (reader.GetValue  
(i).ToString ());  
                row.Cells.Add (cell);  
            }  
            myTable.Rows.Add (row);  
        }  
    }  
}
```

Hay algunos casos en que se necesitó de consultas anidadas a la base de datos, teniéndose el inconveniente de que era necesario cerrar una antes de abrir otra y se perdían los datos. Por esta razón se realizó una asignación a vectores de los resultados obtenidos en una consulta, de manera que no se tenga que utilizar las consultas anidadas.

A continuación un ejemplo de asignación de los resultados de una consulta a vectores en código C#:

```
string selectCmd = "SELECT cur_codigo, inscrip_nota1, inscrip_nota2 FROM  
inscripciones WHERE alumno_codigo like " + dato + " AND inscrip_saldo like  
'0'";
```

```
selectCommand.CommandText = selectCmd;  
reader = selectCommand.ExecuteReader ();  
int m=0;  
string[] cod = new string[100];  
string[] n1 = new string[100];  
string[] n2 = new string[100];  
while (reader.Read ()) {  
    cod[m]= reader.GetValue (0).ToString ();  
    n1[m]= reader.GetValue (1).ToString ();  
    n2[m]= reader.GetValue (2).ToString ();  
    m++;  
}
```

Inserciones

En el archivo *funcion_tesis2.inc*, además de las funciones para la conexión a la base de datos, se tiene la mayoría de funciones para insertar datos en la base de datos.

Estas funciones que se utilizaron para ingresar datos son de tipo *void*, por tanto no retornan ningún valor.

Para realizar una inserción en la base de datos se requiere lo siguiente (códigos en C#):

Crear la cadena con una sentencia sql para insertar datos en campos

específicos de una tabla específica.

```
string ins3 = String.Format ("INSERT INTO areas(area_descrip) VALUES  
('{0}'),a);
```

Crear el comando ADO.NET para inserción en base de datos:

```
SqlCommand insertCommand = cnc.CreateCommand();
```

Asociar la sentencia sql para inserción con el comando de inserción creado:

```
insertCommand.CommandText = ins3;
```

Ejecutar la sentencia sql

```
k = insertCommand.ExecuteNonQuery ();
```

En algunos casos, si el ingreso se realiza correctamente, se redirecciona la página actual a una página (*mensajes.aspx*) que registra un mensaje de éxito en la transacción realizada:

```
Response.Redirect("mensajes.aspx?mensaje=Areaingresada&pagina=admin_  
areas.aspx");
```

A continuación un ejemplo completo de la función en lenguaje C# que ingresa áreas a la base de datos:

```
private void InsertarArea(string a)  
{  
    string ins3 = String.Format ("INSERT INTO areas(area_descrip) VALUES  
('{0}'),a);  
    SqlCommand insertCommand = cnc.CreateCommand();  
    insertCommand.CommandText = ins3;  
    int k;  
    try {  
        k = insertCommand.ExecuteNonQuery ();  
        Response.Redirect("mensajes.aspx?mensaje=EL AREA SE INGRESO  
CON EXITO&pagina=admin_areas.aspx");  
        statusLine.InnerHtml = String.Format ("{0} filas(s) insertadas", k);  
    } catch (Exception e) {  
        errorLine.InnerHtml = "<b>Error al insertar fila: " + e.Message + "</b>";  
    }  
}
```

Todas las demás funciones de ingreso de este archivo tienen el mismo formato, únicamente cambian los parámetros que reciben y las tablas a las que acceden.

Actualizaciones

En el archivo *funcion_tesis3.inc* existen funciones de actualización de la base de datos.

Para realizar una actualización (*update*) se necesita (códigos en C#):

Asignar a una variable tipo cadena la sentencia sql que realizará la actualización de un registro de una tabla de la base de datos.

```
string up = String.Format ("UPDATE eventos SET eve_estado = '{0}' WHERE  
eve_codigo = '{1}'", es, ev);
```

Debemos crear el comando para realizar la actualización:

```
IDbCommand updateCommand = cnc.CreateCommand();
```

Asociamos el comando para actualización que creamos con la variable que contiene la sentencia de actualización:

```
updateCommand.CommandText = up;
```

Ejecutamos la actualización:

```
i = updateCommand.ExecuteNonQuery ();
```

A continuación un ejemplo, tenemos la función *EliminarCurso()*, que actualiza el campo *cur_estado* de la tabla *cursos* colocando el valor a 1.

```
void EliminarCurso (string es, string cur)
{
    string up = String.Format ("UPDATE cursos SET cur_estado = '{0}' WHERE
cur_codigo = '{1}'", es, cur);
    IDbCommand updateCommand = cnc.CreateCommand();
    updateCommand.CommandText = up;
    int i;
    try {
        i = updateCommand.ExecuteNonQuery ();
        statusLine.InnerHtml = String.Format ("{0} fila(s) actualizada en Cursos", i);
    } catch (Exception e) {
        errorLine.InnerHtml = "<b>Error updating row: " + e.Message + "</b>";
    }
}
```

Las demás funciones de actualización tienen el mismo formato, lo que cambia son los valores que reciben y las tablas a las que acceden.

Eliminaciones

En el archivo *funcion_tesis3.inc* existen funciones de eliminación física de la base de datos.

Para realizar una eliminación (*delete*) se necesita (códigos en C#):

Asignar a una variable tipo cadena la sentencia sql que realizará la eliminación de un registro de una tabla de la base de datos.

```
string deleteCmd = String.Format ("DELETE FROM banners WHERE
bann_cod = {0}", c);
```

Debemos crear el comando para realizar la eliminación:

```
    IDbCommand deleteCommand = cnc.CreateCommand();
```

Asociamos el comando para eliminación que creamos con la variable que contiene la sentencia de eliminación:

```
    deleteCommand.CommandText = deleteCmd;
```

Ejecutamos la eliminación:

```
i = deleteCommand.ExecuteNonQuery ();
```

A continuación un ejemplo, tenemos la función *EliminarBanner()*, que elimina un registro de la base de datos.

```
void EliminarBanner (string c)  
{  
    string deleteCmd = String.Format ("DELETE FROM banners WHERE  
bann_cod = {0}", c);  
    IDbCommand deleteCommand = cnc.CreateCommand();  
    deleteCommand.CommandText = deleteCmd;  
    int i;  
    try {  
        i = deleteCommand.ExecuteNonQuery ();  
        statusLine.InnerHtml = String.Format ("{0} fila(s) eliminadas", i);  
    } catch (Exception e) {  
        errorLine.InnerHtml = "<b>Error borrando fila: " + e.Message + "</b>";  
    }  
}
```

ESTRUCTURA GENERAL DE LAS PÁGINAS WEB

En cuestiones de diseño, las páginas Web están diseñadas con tres marcos: el marco superior (*cabecera.aspx*) que contiene el menú principal, el logotipo de la UTN y el nombre del portal.

El marco de la izquierda (*banners.aspx*), contiene los diferentes banners y vínculos a otros sitios Web académicos o publicitarios, el vínculo a los foros y el vínculo a las estadísticas de acceso al portal.

En el marco principal se cargarán todas las páginas que el usuario solicite.

Todas las páginas *aspx* tienen la siguiente estructura general en su código fuente:

En la parte superior se ubican las llamadas a los *namespaces* que se importan para el correcto funcionamiento de las páginas. Además se especifica el lenguaje de programación que se está utilizando.

A continuación mostramos un ejemplo de esta sección:

```
<% @ Page Language="C#" %>  
<% @ import Namespace="System.Configuration" %>  
<% @ import Namespace="System.Data" %>  
<% @ import Namespace="System.Reflection" %>
```

Indica que se utiliza lenguaje C#, se importa el namespace *System.Configuration* para el funcionamiento general de las páginas ASP.NET. El namespace *System.Data* es el encargado de conectar ASP.NET con ADO.NET.

Luego de esto, encontramos el script de programación que se lo realiza en el lenguaje de programación C#.

Es decir tenemos:

```
<script runat="server">  
[Código C#]  
</script>
```

Por último esta la sección *html*, donde se sitúa todo el código *html* incluyendo los formularios y los controles ASP.NET.

Es decir aquí encontramos:

```
<html>
  <head>
    ....
  </head>
  <body>
    .....
  </body>
</html>
```

Página principal. Accesible sin autenticarse y como usuario autenticado

La página principal (*inicial.aspx*) tiene tres formularios ASP.NET, es decir con el siguiente formato:

```
<form id="theForm" runat="server">
.....
</form>
```

El primero (*id="form_noiniciado"*), muestra los cuadros de dialogo para ingresar el *login* y el *password* y el respectivo botón de Ingreso. Por tanto está activo cuando no se ha iniciado una sesión.

El segundo (*id="form_iniciado"*), se activa cuando un usuario a ingresado correctamente su *login* y su *password* y se ha autenticado como usuario del sistema. Muestra un botón para cerrar la sesión actual.

El tercero (*id="theForm"*), muestra los controles ASP.NET insertados en la página.

Dependiendo de si un usuario esta o no son su sesión iniciada se juega con la

propiedad *Visible* de los formularios, de la siguiente forma (código C#):

```
void Page_Load (object sender, EventArgs e)
{
    if(User.Identity.Name!=""){
        form_iniciado.Visible=true;
        form_noiniciado.Visible=false;
        bienvenida.Text = "Bienvenido " + User.Identity.Name;
        MostrarNotas ("1",User.Identity.Name);
    }else{
        form_iniciado.Visible=false;
        form_noiniciado.Visible=true;
    }
}
```

Este ejemplo indica que si el usuario esta autenticado, le mostrara sus notas, caso contrario simplemente se mostrará los cuadros de dialogo para autenticarse.

Cuando un usurario autenticado, desea cerrar la sesión, presiona el respectivo botón de cerrar sesión, lo cual llama a la siguiente función:

```
void Signout_Click (object sender, EventArgs e)
{
    FormsAuthentication.SignOut ();
    Response.Redirect ("index.aspx");
}
```

Lo que hace es cerrar la sesión actual y redireccionar la página actual a la página principal. Esta función se utiliza en todas las demás páginas para cerrar la sesión actual.

Páginas accesibles sin necesidad de autenticarse

A continuación se muestra el nombre de la página y su uso:

Nombre de la página	Descripción
acerca_de.aspx	Muestra los créditos del sistema. Es un enlace desde todas las páginas del portal.
admin_alumno.aspx	Muestra los campos necesarios para registrar los datos personales de un nuevo alumno así como ingresar un nombre de usuario y contraseña. Es un enlace desde <i>inicial.aspx</i> y también desde <i>inscripciones.aspx</i>
ayuda.aspx	La página principal de ayuda para usuarios con perfil de alumno, contiene un pequeño índice en base al menú principal para mostrar el resto de la ayuda. Es un enlace desde <i>cabecera.aspx</i>
ayuda_cur.aspx	La ayuda para el submenú “Cursos”. Es un enlace desde <i>ayuda.aspx</i>
ayuda_eve.aspx	La ayuda para el submenú “Eventos”. Es un enlace desde <i>ayuda.aspx</i>
ayuda_pre.aspx	La ayuda para el submenú “Preinscripciones”. Es un enlace desde <i>ayuda.aspx</i>
ayuda_prin.aspx	La ayuda para el submenú “Principal”. Es un enlace desde <i>ayuda.aspx</i>
ayuda_sem.aspx	La ayuda para el submenú “Seminarios”. Es un enlace desde <i>ayuda.aspx</i>
banners.aspx	El marco izquierdo que contiene los banners de enlaces directos, el vínculo a foros y el vínculo a las estadísticas. Es siempre visible.
cabecera.aspx	El marco superior que contiene el logotipo de la UTN y el nombre del portal. Es siempre visible.
contrasenia.aspx	La página que sirve al usuario a recordar su contraseña olvidada.
cursos.aspx	Muestra el reporte sin detalle de los cursos que están

	activos. Es un enlace desde <i>cabecera.aspx</i>
<i>cursos2.aspx</i>	Muestra el reporte detallado del curso seleccionado. Incluye datos principales del curso o seminario, requisitos, temas, subtemas, horarios e instructores. Recibe como parámetros el código del curso y una bandera indicando si es seminario o no dependiendo donde proviene. Es un enlace desde <i>cursos.aspx</i> y desde <i>seminarios.aspx</i>
<i>encuestas.aspx</i>	Muestra las encuestas que han sido ingresadas anteriormente. Es un enlace desde <i>inicial.aspx</i>
<i>estadisticas1.aspx</i>	Muestra de forma gráfica las estadísticas de acceso al portal por mes. También muestra el número de usuarios registrados y sus porcentajes. Es un enlace desde <i>banners.aspx</i>
<i>estadisticas2.aspx</i>	Muestra de forma gráfica las estadísticas de acceso al portal por días de un determinado mes que llega como parámetro. Es un enlace desde <i>estadisticas1.aspx</i>
<i>eventos.aspx</i>	Muestra el reporte de los eventos que se encuentran activos. Es un enlace desde <i>cabecera.aspx</i>
<i>foro_resp.aspx</i>	Muestra los campos necesarios para responder a un tema del foro. Recibe como parámetro el número del foro. Es un enlace desde <i>foros2.aspx</i>
<i>foros.aspx</i>	Muestra la lista de los temas ingresados en el foro por el administrador. Es un enlace desde <i>banners.aspx</i>
<i>foros2.aspx</i>	Muestra las respuestas de los visitantes de un tema del foro. Recibe como parámetro el número del foro. Es un enlace desde <i>foros.aspx</i>
<i>index.aspx</i>	Es la página central del portal. Contiene los tres marcos principales: <i>cabecera.aspx</i> , <i>banners.aspx</i> e <i>inicial.aspx</i> .
<i>inicial.aspx</i>	Es el marco principal del portal. Muestra la animación de bienvenida, el mensaje, el área de autenticación y el área de encuestas.

inscripciones.aspx	Muestra los campos necesarios para preinscribirse en un curso o seminario. Es un enlace desde <i>cabecera.aspx</i>
instructor.aspx	Muestra el reporte detallado del instructor seleccionado, incluye datos personales del instructor y su hoja de vida (cursos seguidos y experiencia). Recibe como parámetro el código del instructor. Es un enlace desde <i> cursos.aspx</i> y desde <i>seminarios.aspx</i>
mensajes.aspx	Es una página accesible desde cualquier ingreso que se realice en el sistema. Muestra un mensaje de éxito de cualquier ingreso. Recibe como parámetros el mensaje en si y la página a la cual debe retornar al presionar el botón de regreso.
resultados.aspx	Muestra de forma gráfica los resultados obtenidos en las votaciones de las encuestas de la página principal. Recibe como parámetro el número de la encuesta. Es un enlace desde <i>inicial.aspx</i> y desde <i>encuestas.aspx</i>
seminarios.aspx	Muestra el reporte sin detalle de los seminarios que están activos. Es un enlace desde <i>cabecera.aspx</i>
votar.aspx	Muestra los campos necesarios para realizar un voto de una encuesta ya pasada. Recibe como parámetro el número de la encuesta. Es un enlace desde <i>encuestas.aspx</i>

Estas páginas son accesibles para cualquier persona que visite el sitio sin necesidad de autenticarse como usuario del sistema.

Estas páginas cuentan con tres formularios:

El primero (*id="form_nosesion"*), muestra un enlace directo a la página principal, con el objetivo de que se inicie la sesión.

El segundo (*id="form_sesion"*), se lo utiliza cuando un usuario está autenticado como usuario del sistema. Muestra un vínculo para cerrar la

sesión actual, además se lo utiliza para mostrar el nombre del usuario actual cuando un usuario esta autenticado como usuario del sistema

El tercero (*id="theForm"*), se utiliza para mostrar el cuerpo principal de las páginas. En este formulario se sacarán los reportes o se mostraran cuadros de dialogo para ingresos, etc.

Para estos casos, también se juega con la propiedad *Visible* de los formularios, de la siguiente manera (código C#):

```
void Page_Load (object sender, EventArgs e)
{
    if (!IsPostBack){
        MostrarEventos ();
        if(User.Identity.Name!=""){
            form_nosesion.Visible=false; //indica iniciar sesion
            form_sesion.Visible=true; //es el que indica el nombre de usuario
            user.Text+= User.Identity.Name;
        }else{
            form_nosesion.Visible=true;
            form_sesion.Visible=false;
        }
    }
}
```

Páginas accesibles solo para usuarios con perfil de administrador

A continuación se muestra el nombre de la página y su uso:

Nombre de la página	Descripción
ad_alumnos.aspx	Muestra un enlace para registrar a los alumnos en el sistema. Es un enlace desde <i>administración.aspx</i>
ad_areas.aspx	Muestra opciones relacionadas con la administración de áreas (reportes e ingresos). Es un enlace desde <i>administración.aspx</i>

ad_cursos.aspx	Muestra un conjunto de opciones relacionadas con la administración de cursos. Es un enlace desde <i>administración.aspx</i>
ad_encuestas.aspx	Muestra un conjunto de opciones relacionadas con la administración de encuestas, foros y tests generales. Es un enlace desde <i>administración.aspx</i>
ad_eventos.aspx	Muestra un conjunto de opciones relacionadas con la administración de eventos. Es un enlace desde <i>administración.aspx</i>
ad_instructores.aspx	Muestra un conjunto de opciones relacionadas con la administración de instructores. Es un enlace desde <i>administración.aspx</i>
ad_parametros.aspx	Muestra un conjunto de opciones relacionadas con la administración de parámetros generales del sistema. Es un enlace desde <i>administración.aspx</i>
ad_posibles.aspx	Muestra un conjunto de opciones relacionadas con la administración de los futuros cursos que se planean organizar en la FICA. Es un enlace desde <i>administración.aspx</i>
ad_seminarios.aspx	Muestra un conjunto de opciones relacionadas con la administración de seminarios. Es un enlace desde <i>administración.aspx</i>
ad_usradmin.aspx	Muestra un conjunto de opciones relacionadas con la administración de usuarios de administración. Es un enlace desde <i>administración.aspx</i>
admin_areas.aspx	Muestra los campos necesarios para ingresar una nueva área. Es un enlace desde <i>ad_areas.aspx</i>
admin_curso.aspx	Muestra los campos necesarios para ingresar un nuevo curso. Incluye campos para escoger el área del curso y el instructor. Es un enlace desde <i>ad_cursos.aspx</i>
admin_curso2.aspx	Es un punto de referencia para ingresar requisitos, temas, subtemas y horarios de un curso ingresado

	previamente. Recibe como parámetros el código del curso, el código de instructor, la fecha de inicio y fin del curso y el código para ver si es seminario (1) o no (0). Es una secuencia desde <i>admin_curso.aspx</i>
<i>admin_curso3.aspx</i>	Muestra los campos necesarios para ingresar un tema, su horario y varios subtemas de un curso ingresado previamente. Recibe como parámetros el código del curso, el código de instructor, la fecha de inicio y fin del curso y el código para ver si es seminario (1) o no (0). Es un enlace desde <i>admin_curso2.aspx</i>
<i>admin_curso4.aspx</i>	Muestra los campos necesarios para ingresar un requisito de un curso ingresado previamente. Recibe como parámetros el código del curso, el código de instructor, la fecha de inicio y fin del curso y el código para ver si es seminario (1) o no (0). Es un enlace desde <i>admin_curso2.aspx</i>
<i>admin_encuestas.aspx</i>	Muestra los campos necesarios para crear una nueva encuesta de opción múltiple para la página inicial. Es un enlace desde <i>ad_encuestas.aspx</i>
<i>admin_eventos.aspx</i>	Muestra los campos necesarios para ingresar un nuevo evento. Es un enlace desde <i>ad_eventos.aspx</i>
<i>admin_foro.aspx</i>	Muestra los campos necesarios para ingresar un nuevo tema del foro. Es un enlace desde <i>ad_encuestas.aspx</i>
<i>admin_generales.aspx</i>	Es una página que sirve para el ingreso de nuevas preguntas secretas, instituciones y títulos universitarios dependiendo del valor del parámetro. Recibe como parámetro la tabla a cual debe acceder. Es un enlace desde <i>ad_parametros.aspx</i>
<i>admin_instruc.aspx</i>	Muestra los campos necesarios para ingresar un nuevo instructor. Es un enlace desde <i>ad_instructores.aspx</i>

admin_instruc2.aspx	Es un punto de referencia para el ingreso de la experiencia y cursos recibidos por un instructor ingresado previamente. Recibe como parámetro el código del instructor. Es una secuencia desde <i>admin_instruc.aspx</i> (perfil administrador) y un enlace desde <i>administracion2.aspx</i> (perfil instructor)
admin_instruc3.aspx	Muestra los campos necesarios para ingresar un nuevo un registro con información de la experiencia de un instructor. Recibe como parámetro el código del instructor. Es un enlace desde <i>admin_instruc2.aspx</i>
admin_instruc4.aspx	Muestra los campos necesarios para ingresar un nuevo un registro con información de cursos recibidos por un instructor. Recibe como parámetro el código del instructor. Es un enlace desde <i>admin_instruc2.aspx</i>
admin_instruc5.aspx	Sirve para actualizar los datos de un instructor y/o para mostrar la liquidación económica de cursos por instructor. Recibe como parámetro el la acción que se desea ejecutar (una de las dos especificadas). Es un enlace desde <i>ad_instructores.aspx</i>
admin_parametros.aspx	Muestra los campos para actualizar los parámetros generales del sistema: nombres del Decano FICA y Subdecano FICA, número de cuenta corriente, banco que genera la cuenta corriente y nombres de las animaciones de la página principal. Es un enlace desde <i>ad_parametros.aspx</i>
admin_posibles.aspx	Muestra los campos para ingresar un posible futuro curso a organizarse en la FICA. Es un enlace desde <i>ad_posibles.aspx</i>
admin_seminario.aspx	Muestra los campos necesarios para ingresar un nuevo seminario. Incluye un campo para seleccionar el área del seminario. Es un enlace

	desde <i>ad_seminarios.aspx</i>
<i>admin_seminario2.aspx</i>	Es un punto de referencia para ingresar requisitos, temas, subtemas, horarios e instructores de un seminario ingresado previamente. Recibe como parámetros el código del seminario, la fecha de inicio y fin del seminario y el código para ver si es seminario (1) o no (0). Es una secuencia desde <i>admin_seminario.aspx</i>
<i>admin_seminario3.aspx</i>	Muestra los campos necesarios para ingresar un requisito de un seminario ingresado previamente. Recibe como parámetros el código del curso, la fecha de inicio y fin del seminario y el código para ver si es seminario (1) o no (0). Es un enlace desde <i>admin_seminario2.aspx</i>
<i>admin_seminario4.aspx</i>	Muestra los campos necesarios para ingresar un tema, su instructor, su horario y varios subtemas de un seminario ingresado previamente. Recibe como parámetros el código del seminario, la fecha de inicio y fin del seminario y el código para ver si es seminario (1) o no (0). Es un enlace desde <i>admin_seminario2.aspx</i>
<i>admin_testadm.aspx</i>	Muestra los campos para ingresar un nuevo test general aplicado a cursos. Se ingresan solo las preguntas, ya que las respuestas son predeterminadas. Es un enlace desde <i>ad_encuestas.aspx</i>
<i>admin_testinst.aspx</i>	Muestra los campos para ingresar un test de conocimientos aplicado a un curso o seminario específico. Se ingresan solo las preguntas, ya que las respuestas son predeterminadas para que sean abiertas o de opinión. Es un enlace desde <i>admin_cursos2.aspx</i> y <i>admin_seminarios2.aspx</i>
<i>admin_usradmin.aspx</i>	Muestra los campos para ingresar un nuevo usuario

	<p>con perfil de administrador al sistema. Es un enlace desde <i>ad_usradmin.aspx</i></p>
administracion.aspx	<p>Es la página principal de la administración y contiene todos los enlaces a las demás páginas de administración. Es un enlace desde <i>cabecera.aspx</i></p>
alumnos.aspx	<p>Recibe 3 parámetros: <i>vista</i>, <i>sem</i> y <i>who</i>. Cuando se recibe <i>vista=1</i>, <i>sem=0</i> y <i>who=inscritos</i>, muestra un reporte de los alumnos inscritos en un curso seleccionado previamente. Cuando se recibe <i>vista=1</i>, <i>sem=0</i> y <i>who=todos</i>, muestra un reporte de los alumnos preinscritos en un curso seleccionado previamente. Cuando se recibe <i>vista=6</i> y <i>sem=0</i> muestra los campos necesarios para ingresar las notas de un curso que se seleccione previamente. Cuando se recibe <i>vista=2</i> y <i>sem=0</i> muestra los campos necesarios para inscribir a un alumno seleccionado previamente e inscrito en un curso seleccionado previamente. Cuando se recibe <i>vista=10</i> es para actualizar datos de un curso que se selecciona (esto lo puede hacer un instructor solamente). Cuando se recibe <i>vista=20</i> es para ver el reporte de tests de un curso seleccionado. Si en las anteriores opciones se cambia el valor de <i>sem</i> a 1, se tendrá lo mismo pero para seminarios.</p>
alumnos2.aspx	<p>Es un enlace que se encuentra en la página <i>alumnos.aspx</i>. Recibe dos parámetros: <i>al</i> y <i>p</i>. El parámetro <i>al</i> corresponde al código del alumno y el parámetro <i>p</i> corresponde a la función de la página. Cuando <i>p=1</i> muestra un reporte detallado de los datos personales del alumno cuyo código es <i>al</i>. Cuando <i>p=2</i> muestra un reporte detallado de los</p>

	datos personales del alumno cuyo código es <i>al</i> y además permite inscribir a ese alumno en un curso o seminario, es decir ingresar el pago respectivo.
alumnos3.aspx	Muestra un reporte de los alumnos inscritos en el curso o seminario cuyo código es el valor del parámetro <i>curso</i> . Incluye la fecha de inscripción, nota de evaluación y % de asistencia. Recibe los parámetros <i>curso</i> y <i>sem</i> desde <i>histo_cursos.aspx</i> .
eliminar_curso.aspx	Muestra todos los cursos activos, y da la posibilidad de eliminarlos uno por uno. Una vez que se elimina un curso, la página se redirecciona a <i>observaciones.aspx</i> y le envía como parámetro el código del curso.
observaciones.aspx	Recibe como parámetro la variable <i>curso</i> desde <i>eliminar_curso.aspx</i> . Permite ingresar una observación del curso eliminado y especificar si se realizó con éxito o no.
sugerencias.aspx	Muestra un reporte de las sugerencias escritas por los alumnos en un curso previamente seleccionado. Muestra la sugerencia y el alumno que la escribió.
histo_cursos.aspx	Recibe de <i>administracion.aspx</i> el parámetro <i>semi</i> . Da la posibilidad de escoger un rango de fechas y muestra el reporte de cursos o seminarios cuya fecha de inicialización esté en ese rango y que ya han sido eliminados anteriormente (campo <i>cur_estado=1</i> de la tabla <i>cursos</i>). Si <i>semi=0</i> muestra el reporte histórico de cursos. Si <i>semi=1</i> muestra el reporte histórico de seminarios.
areas.aspx	Muestra un reporte de las áreas ingresadas. Es un enlace desde <i>ad_areas.aspx</i>
bitacora.aspx	Muestra el reporte de las acciones realizadas (modificaciones en la base de datos) por los

	usuarios administradores e instructores en el sistema. Es un enlace desde <i>administración.aspx</i>
<i>depositos.aspx</i>	Muestra un reporte de los comprobantes de depósitos de un curso o seminario. Recibe los parámetros <i>curso</i> y <i>sem</i> desde <i>histo_cursos.aspx</i> .
<i>diplomas.aspx</i>	Muestra un reporte de notas de evaluación y aprobados o no. Incluye un vínculo para imprimir el diploma del alumno si es que aprueba el curso o seminario. Es un enlace desde <i>ad_cursos.aspx</i> y <i>ad_seminarios.aspx</i>
<i>diplomas2.aspx</i>	Muestra el reporte del diploma de un curso o seminario para un alumno específico. Este diploma se puede imprimir. Recibe como parámetros el código del alumno, el código del curso y el tipo de diploma (asistido - aprobado o asistido). Es un enlace desde <i>diplomas.aspx</i>
<i>eco_cursos.aspx</i>	Muestra un histórico financiero de cursos o seminarios. Indica el precio unitario y total recaudado en un curso o seminario. Recibe el parámetro <i>semi</i> desde <i>ad_cursos.aspx</i> y desde <i>ad_seminarios.aspx</i>
<i>eco_instruc.aspx</i>	Muestra el reporte de la liquidación económica de un instructor seleccionado anteriormente en todos los cursos que ha participado. Recibe como parámetro el código del instructor. Es una secuencia desde <i>admin_instruc5.aspx</i> (perfil administrador) y un enlace desde <i>administracion2.aspx</i> (perfil instructor)
<i>eliminar_banner.aspx</i>	Permite eliminar banners o accesos directos de la página <i>banners.aspx</i> . Es un enlace desde <i>ad_parametros.aspx</i>
<i>eliminar_curso.aspx</i>	Muestra todos los cursos activos, y da la posibilidad de eliminarlos uno por uno. Una vez que se elimina

	un curso, la página se redirecciona a <i>observaciones.aspx</i> y le envía como parámetro el código del curso. Es un enlace desde <i>ad_cursos.aspx</i>
<i>eliminar_encuesta.aspx</i>	Permite eliminar encuestas. Es un enlace desde <i>ad_encuestas.aspx</i>
<i>eliminar_evento.aspx</i>	Muestra todos los eventos activos, y da la posibilidad de eliminarlos uno por uno. Es un enlace desde <i>ad_eventos.aspx</i>
<i>eliminar_foro.aspx</i>	Permite eliminar temas del foro. Es un enlace desde <i>ad_encuestas.aspx</i>
<i>eliminar_foro2.aspx</i>	Permite eliminar temas del foro. Es un enlace desde <i>eliminar_foro.aspx</i>
<i>eliminar_posible.aspx</i>	Permite eliminar posibles futuros cursos. Es un enlace desde <i>ad_posibles.aspx</i>
<i>eliminar_seminario.aspx</i>	Muestra todos los seminarios activos, y da la posibilidad de eliminarlos uno por uno. Una vez que se elimina un seminario, la página se redirecciona a <i>observaciones.aspx</i> y le envía como parámetro el código del seminario.
<i>eliminar_usradmin.aspx</i>	Permite eliminar usuarios de administración registrados en el sistema. Es un enlace desde <i>ad_usradmin.aspx</i>
<i>encuesta2.aspx</i>	Muestra el reporte de las encuestas que se realizan a los estudiantes para saber que curso les gustaría seguir. Es un enlace desde <i>ad_posibles.aspx</i>
<i>histo_cursos.aspx</i>	Muestra un histórico de los cursos o seminarios que se ha nrealizado en la FICA. Es un enlace desde <i>ad_cursos.aspx</i> o desde <i>ad_seminarios.aspx</i>
<i>histo_eventos.aspx</i>	Muestra los eventos realizados anteriormente. Es un enlace desde <i>ad_eventos.aspx</i>
<i>instructor1.aspx</i>	Muestra un reporte de los instructores ingresados en el sistema. Es un enlace desde

	<i>ad_instructores.aspx</i>
observaciones.aspx	Permite ingresar las observaciones de un curso que se elimina. Es una secuencia desde <i>eliminar_curso.aspx</i>
observaciones2.aspx	Muestra el reporte de las observaciones ingresadas en un curso cuando se lo eliminó manualmente. Recibe como parámetros el código del curso y si el número que indica si es o no seminario. Es un enlace desde <i>histo_cursos.aspx</i>
posibles.aspx	Es un reporte de los posibles futuros cursos que se han ingresado en el sistema. Es un enlace desde <i>ad_posibles.aspx</i>
recibo.aspx	Es un reporte imprimible del recibo que se obtiene al inscribir a un alumno. Recibe como parámetros el código del alumno y del curso. Es una secuencia desde <i>alumnos.aspx</i>
sugerencias.aspx	Es el reporte de ests generales o tests de conocimientos que se aplica para cursos específicos. Recibe como parámetros el número que indica si es o no seminario, y el tipo de vista que se quiere (test de conocimientos o test general). Es un enlace desde <i>ad_encuestas.aspx</i> y desde <i>ad_cursos.aspx</i>
sugerencias2.aspx	Es un reporte individual de un alumno acerca de: test general o test de conocimientos. Recibe como parámetros el código del curso, código del alumno, el código del test, y la vista dependiendo si se quiere ver el test general (administrador) o el test de conocimientos (instructor). Es un secuencia desde <i>sugerencias.aspx</i>
sugerencias3.aspx	Muestra los resultados estadísticos del test general de un curso específico. Es un enlace desde <i>sugerencias.aspx</i>

Estas páginas cuentan con cuatro formularios:

El primero (*id="form_noiniciado"*), muestra un control tipo *label* que muestra un mensaje de error en el caso de que un usuario distinto al administrador ingrese a una de estas páginas.

El segundo (*id="form_nosesion"*), muestra un enlace directo a la página principal, con el objetivo de que se inicie la sesión.

El tercero (*id="form_sesion"*), se lo utiliza cuando un usuario esta autenticado como usuario del sistema. Muestra un botón para cerrar la sesión actual y el nombre de usuario autenticado.

El cuarto (*id="theForm"*), se utiliza para mostrar el cuerpo principal de las páginas en el caso de que el usuario que ingresa sea el administrador. En este formulario se sacarán los reportes o se mostraran cuadros de dialogo para ingresos, etc.

Para estos casos, también se juega con la propiedad *Visible* de los formularios, de la siguiente manera (código C#):

```
void Page_Load (object sender, EventArgs e)
{
    if (!IsPostBack){
        if(User.Identity.Name!=""){
            if(User.Identity.Name=="administrador"){
                theForm.Visible=true;
                form_iniciado2.Visible=true;
                form_noiniciado2.Visible=false;
                form_iniciado3.Visible=true;
                user.Text+= User.Identity.Name;
            }
            else{
                theForm.Visible=false;
                form_noiniciado.Visible=true;
                error.Text="Usted no es el administrador de este sitio y no
                podr&aacute; ver el contenido de esta secci&oacute;n. Por favor ingrese como
                administrador.";
                form_iniciado2.Visible=true;
                form_noiniciado2.Visible=false;
            }
        }
    }
}
```

```

        form_iniciado3.Visible=true; //es el que indica el nombre de usuario
        user.Text+= User.Identity.Name;
    }
}
else{
    theForm.Visible=false;
    form_noiniciado.Visible=true;
    error.Text="Usted no es el administrador de este sitio y no
    podr&aacute; ver el contenido de esta secci&oacute;n. Por favor ingrese como
    administrador.";
    form_iniciado2.Visible=false;
    form_noiniciado2.Visible=true;
    form_iniciado3.Visible=false;
}
}
}
}

```

Páginas accesibles solo para usuarios con perfil de alumnos

A continuación se muestra el nombre de la página y su uso:

admin_alumno2.aspx	Permite al usuario alumno actualizar sus datos personales e inclusive su contraseña. Recibe como parámetro el código del alumno. Es un enlace desde <i>inicial.aspx</i>
encuesta.aspx	Permite que el usuario alumno vote por un determinado posible futuro curso que se planea organizar en la FICA. Recibe como parámetro el código del alumno y realiza una verificación de manera que el alumno pueda votar una sola vez por un curso. Es un enlace desde <i>inicial.aspx</i>
test.aspx	Permite al usuario alumno ingresar tanto el test de conocimientos como el test general de un curso, siempre y cuando esté inscrito en el mismo. Incluye una revisión para que el alumno ingrese el test una sola vez. Recibe como parámetros el código del curso y el código del alumno. Es un enlace desde <i>inicial.aspx</i>

Páginas accesibles solo para usuarios con perfil de instructor

A continuación se muestra el nombre de la página y su uso:

admin_curso2.aspx	Es un punto de referencia para ingresar requisitos, temas, subtemas y horarios de un curso que se selccione previamente (solo de los que el instructor dicte). Recibe como parámetros el código del curso, el código de instructor, la fecha de inicio y fin del curso y el código para ver si es seminario (1) o no (0). Es una secuencia desde <i>admin_curso.aspx</i>
admin_curso3.aspx	Muestra los campos necesarios para ingresar un tema, su horario y varios subtemas de un curso seleccionado previamente. Recibe como parámetros el código del curso, el código de instructor, la fecha de inicio y fin del curso y el código para ver si es seminario (1) o no (0). Es un enlace desde <i>admin_curso2.aspx</i>
admin_curso4.aspx	Muestra los campos necesarios para ingresar un requisito de un curso seleccionado previamente. Recibe como parámetros el código del curso, el código de instructor, la fecha de inicio y fin del curso y el código para ver si es seminario (1) o no (0). Es un enlace desde <i>admin_curso2.aspx</i>
admin_instruc2.aspx	Muestra la información de experiencia y cursos recibidos por un instructor y permite el ingreso de nueva información relacionada a su currículo. Recibe como parámetro el código del instructor. Es una secuencia desde <i>admin_instruc.aspx</i> (perfil administrador) y un enlace desde <i>administracion2.aspx</i> (perfil instructor)
admin_instruc3.aspx	Muestra los campos necesarios para ingresar un nuevo un registro con información de la

	<p>experiencia de un instructor. Recibe como parámetro el código del instructor. Es un enlace desde <i>admin_instruc2.aspx</i></p>
<i>admin_instruc4.aspx</i>	<p>Muestra los campos necesarios para ingresar un nuevo un registro con información de cursos recibidos por un instructor. Recibe como parámetro el código del instructor. Es un enlace desde <i>admin_instruc2.aspx</i></p>
<i>administración2.aspx</i>	<p>Es la página principal de administración para un usuario con perfil de instructor. Es un enlace desde <i>cabecera.aspx</i></p>
<i>admin_testinst.aspx</i>	<p>Muestra los campos para ingresar un test de conocimientos aplicado a un curso o seminario específico. Se ingresan solo las preguntas, ya que las respuestas son predeterminadas para que sean abiertas o de opinión. Es un enlace desde <i>admin_cursos2.aspx</i> y <i>admin_seminarios2.aspx</i></p>
<i>alumnos.aspx</i>	<p>Recibe 3 parámetros: <i>vista</i>, <i>sem</i> y <i>who</i>.</p> <p>Cuando se recibe <i>vista=1</i>, <i>sem=0</i> y <i>who=inscritos</i>, muestra un reporte de los alumnos inscritos en un curso seleccionado previamente. (perfil administrador)</p> <p>Cuando se recibe <i>vista=1</i>, <i>sem=0</i> y <i>who=todos</i>, muestra un reporte de los alumnos preinscritos en un curso seleccionado previamente. (perfil administrador)</p> <p>Cuando se recibe <i>vista=6</i> y <i>sem=0</i> muestra los campos necesarios para ingresar las notas de un curso que se seleccione previamente. (perfil administrador e instructor)</p> <p>Cuando se recibe <i>vista=2</i> y <i>sem=0</i> muestra los campos necesarios para inscribir a un alumno seleccionado previamente e inscrito en un curso</p>

	<p>seleccionado previamente. (perfil administrador)</p> <p>Cuando se recibe <i>vista=10</i> es para actualizar datos de un curso que se selecciona (esto lo puede hacer un instructor solamente). (perfil administrador e instructor)</p> <p>Cuando se recibe <i>vista=20</i> es para ver el reporte de tests generales o de conocimientos de un curso seleccionado. (perfil administrador e instructor)</p> <p>Si en las anteriores opciones se cambia el valor de <i>sem</i> a 1, se tendrá lo mismo pero para seminarios.</p>
eco_instruc.aspx	<p>Muestra el reporte de la liquidación económica de un instructor seleccionado anteriormente en todos los cursos que ha participado. Recibe como parámetro el código del instructor. Es una secuencia desde <i>admin_instruc5.aspx</i> (perfil administrador) y un enlace desde <i>administracion2.aspx</i> (perfil instructor)</p>

FUNCIONES DE CÓDIGO ESPECIALES UTILIZADAS EN LA APLICACIÓN

Codificación de contraseñas

Con el objetivo de implementar un seguridad a los datos del sistema, se implementó un pequeña función de codificación de contraseñas, de manera que cuando un nuevo usuario se registre, su contraseña se guarde en la base de datos de una manera diferente a la ingresada.

De igual manera esta función sirve para decodificar las contraseñas, por ejemplo cuando el usuario ya registrado trata de ingresar al sistema con su login y password, el sistema accede a esta función para decodificar la información de la base de datos y producir el resultado deseado.

A continuación se presenta esta función, la cual se encuentra en los archivos *funcion_tesis.inc* y *funcion_tesis2.inc*:

```
string codifica(string p)
{
    int n=p.Length; //saca el tamaño de la cadena
    string p2="";
    char aux;
    for( int j=0 ; j<n ; j++){
        aux=p[j];
        if (p[j]=='a') aux='Z';      if (p[j]=='Z') aux='a';
        if (p[j]=='e') aux='Y';      if (p[j]=='Y') aux='e';
        if (p[j]=='i') aux='X';      if (p[j]=='X') aux='i';
        if (p[j]=='o') aux='W';      if (p[j]=='W') aux='o';
        if (p[j]=='u') aux='V';      if (p[j]=='V') aux='u';
        if (p[j]=='1') aux='A';      if (p[j]=='A') aux='1';
        if (p[j]=='2') aux='B';      if (p[j]=='B') aux='2';
        if (p[j]=='3') aux='C';      if (p[j]=='C') aux='3';
        if (p[j]=='4') aux='D';      if (p[j]=='D') aux='4';
        if (p[j]=='5') aux='E';      if (p[j]=='E') aux='5';
        if (p[j]=='6') aux='F';      if (p[j]=='F') aux='6';
        if (p[j]=='7') aux='G';      if (p[j]=='G') aux='7';
        if (p[j]=='8') aux='H';      if (p[j]=='H') aux='8';
        if (p[j]=='9') aux='I';      if (p[j]=='I') aux='9';
        if (p[j]=='0') aux='J';      if (p[j]=='J') aux='0';
        p2 = p2 + aux;
    }
    return p2;
}
```

Función para caracteres especiales

Uno de los grandes problemas en la realización del aplicativo fue el despliegue incorrecto de caracteres especiales.

Cuando se ingresaban datos con caracteres especiales, esta información se guardaba con caracteres extraños, y por supuesto, al momento de producir una consulta se visualizaban esos caracteres extraños que estaban ingresados.

Por esta razón, se desarrollo una función que permitió tomar todas las codificaciones extrañas que se encontraban, de manera que puedan ser reemplazadas por su respectivo código pero en lenguaje html.

De esta manera, cuando se ingresa una palabra con un carcter especial, el sistema lo detecto y reemplaza o los caracteres especiales encontrados por su equivalencia en lenguaje html.

Ejemplo:

Si en un ingreso cualquiera del sistema, escribo la palabra *López*, el sistema detecta una letra con tilde y la reemplaza con su equivalencia en html, ingresándose la palabra como *López*

Para sacar una consulta, el sistema detecta automáticamente que la cadena *ó* representa a la letra *ó* y saca a pantalla el resultado correcto que sería la palabra *López*

Para algunas letras no se encontró la debida correspondencia, por loe que no se realiza el cambio. Por tanto hay que evitar el ingreso de los siguientes caracteres:

Ç " \ ÿ i

A continuación se presenta la función que realiza este cambio, la cual se encuentra en el archivo *funcion_tesis2.inc*

```

string espaniol(string p)
{
    StringBuilder sb = new StringBuilder(p);
    sb.Replace("%C3%A1", "&acute;"); //letra: á
    sb.Replace("%C3%A9", "&eacute;"); //letra: é
    sb.Replace("%C3%AD", "&iacute;"); //letra: í
    sb.Replace("%C3%B3", "&oacute;"); //letra: ó
    sb.Replace("%C3%BA", "&uacute;"); //letra: ú
    sb.Replace("%C3%B1", "&ntilde;"); //letra: ñ
    sb.Replace("%C3%81", "&Aacute;"); //letra: Á
    sb.Replace("%C3%89", "&Eacute;"); //letra: É
    sb.Replace("%C3%8D", "&Iacute;"); //letra: Í
    sb.Replace("%C3%93", "&Oacute;"); //letra: Ó
    sb.Replace("%C3%9A", "&Uacute;"); //letra: Ú
    sb.Replace("%C3%91", "&Ntilde;"); //letra: Ñ
    sb.Replace("%0D%0A", "<br>"); //letra: enter (salto de línea)
// sb.Replace("%C3%87", "Ç"); //letra: Ç //ojo: saca otros caracteres
    sb.Replace("+", " "); //letra: espacio
    sb.Replace("%40", "@"); //letra: @
    sb.Replace("%21", "!"); //letra: !
// sb.Replace("%22", ""); //letra: " //ojo me da error si le pongo
    sb.Replace("%23", "#"); //letra: #
    sb.Replace("%24", "$"); //letra: $
    sb.Replace("%25", "%"); //letra: %
    sb.Replace("%26", "&"); //letra: &
    sb.Replace("%27", "'"); //letra: '
    sb.Replace("%28", "("); //letra: (
    sb.Replace("%29", ")"); //letra: )
    sb.Replace("%2A", "*"); //letra: *
    sb.Replace("%2B", "+"); //letra: +
    sb.Replace("%2C", ","); //letra: ,
    sb.Replace("%3B", ";"); //letra: ;
    sb.Replace("%2F", "/"); //letra: /
    sb.Replace("%3A", ":"); //letra: :
    sb.Replace("%3C", "<"); //letra: <
    sb.Replace("%3D", "="); //letra: =
    sb.Replace("%3E", ">"); //letra: >
    sb.Replace("%3F", "?"); //letra: ?
    sb.Replace("%C2%B7", "."); //letra: · punto alto ??? //ojo mal
    sb.Replace("%C2%BF", "¿"); //letra: ¿ //ojo: saca un caracter adicional
    sb.Replace("%C2%A1", "¡"); //letra: ¡ //ojo: saca un caracter adicional
    sb.Replace("%5B", "["); //letra: [
// sb.Replace("%5C", "\"); //letra: \ //ojo me da error si le pongo
    sb.Replace("%5D", "]"); //letra: ]
    sb.Replace("%7C", "|"); //letra: |
    sb.Replace("%7B", "{"); //letra: {
    sb.Replace("%7D", "}"); //letra: }
    sb.Replace("%7E", "~"); //letra: ~
    return sb.ToString();
}

```

Validación de fechas

Otro de los problemas que tiene que afrontar todo desarrollador de software es la validación de fechas.

Existen años bisiestos, algunos meses tienen más días que otros, etc.

Por tanto fue necesario crear una función que solucione estos inconvenientes.

A continuación se presenta esta función, la cual se encuentra en los archivos *funcion_tesis.inc* y *funcion_tesis2.inc*

```
string revisa_fecha(string d, string m, string a){
string result="1";
if(m=="04" || m=="06" || m=="09" || m=="11"){
if(int.Parse(d)>30)
result="0"; //0 es mal
else
result="1";
}

if(m=="02"){
if(int.Parse(a)%4==0)
if(int.Parse(d)>29)
result="0";
else
result="1"; //1 es bien
if(int.Parse(a)%4!=0)
if(int.Parse(d)>28)
result="0";
else
result="1"; //1 es bien
}
return result;
}
```

CREACIÓN DE RESPALDOS

Como se había dicho, la aplicación del Portal para el Centro de Capacitación Continua de la FICA esta dividida las siguientes partes:

- ? **Archivos realizados en lenguaje ASP.NET y C#**
- ? **Archivos realizados en lenguaje Php**
- ? **Archivos de la Base de Datos**

Hay que tomar en cuenta que lo más importante es tener respaldos constantes de la información contenida en la base de datos, ya que esa se puede estar generando constantemente por medio del sistema.

Si se realizan cambios de las páginas Web, también es necesario sacar sus debidos respaldos del código fuente, y por supuesto, documentar los cambios para evitar futuras complicaciones.

Por tanto, debemos sacar respaldos de las siguientes ubicaciones del servidor:

- Archivos en lenguaje ASP.NET y C#

Ubicación: */home/pablo/mono/xsp-0.5/server/test/capacitacion/*

- Archivos en lenguaje Php

Ubicación: */var/www/html/upload*

- Archivos de la Base de Datos

Ubicación: */var/lib/mysql/ccc/*