

System.IO.File.WriteAllText(ruta, contenido) -- guardar en un fichero que si no existe se crea

FontDialog fd = new FontDialog(); -- Formato de fuente

txtContenido.Text.IndexOf(busq, posFinal); -- nos da la posición del primer carácter de busq empezando desde posFinal

txtContenido.Select(inicio, busq.Length);// el primer parametro es la posicion del texto a seleccionar y el segundo la longitud del texto a seleccionar

txtContenido.ScrollToCaret();//Hace scroll hasta la posicion del texto seleccionado

txtNuevo.Focus(); -- sitúa el cursor en ese elemento

treeView1.ImageList = imageList;//Asignamos la lista de imágenes al control TreeView

treeView1.BeforeExpand += treeView1_BeforeExpand;//Asignamos el evento BeforeExpand al control TreeView

treeView1.AfterSelect += treeView1_AfterSelect;//Asignamos el evento AfterSelect al control TreeView

TreeNode nodoRaiz = new TreeNode("C:\\", 0, 1);//Creamos el nodo raíz con la etiqueta "C:" y la imagen correspondiente

nodoRaiz.Tag = new DirectoryInfo("C:\\");//Asignamos al nodo raíz el directorio raíz "C:"

nodoRaiz.Nodes.Add(new TreeNode());//Añadimos un nodo hijo vacío al nodo raíz

treeView1.Nodes.Add(nodoRaiz);//Añadimos el nodo raíz al control TreeView

TreeNode node = e.Node;//Nodo que se va a expandir

if (node.Nodes[0].Text == "" && node.Nodes.Count == 1)//Si el nodo hijo está vacío y no tiene más nodos hijos

node.Nodes.Clear();//Limpiamos los nodos hijos

DirectoryInfo directory = (DirectoryInfo)node.Tag;//Directorio asociado al nodo que se va a expandir

Try { foreach (DirectoryInfo d in directory.GetDirectories())//Recorremos los subdirectorios del directorio

TreeNode nuevoNodo = new TreeNode(d.Name, 0, 1);//Creamos un nuevo nodo con el nombre del subdirectorio y las imagenes correspondiente

nuevoNodo.Tag = d;//Asignamos al nodo el subdirectorio

nuevoNodo.Nodes.Add(new TreeNode());//Añadimos un nodo hijo vacío al nodo

node.Nodes.Add(nuevoNodo);//Añadimos el nodo al nodo que se va a expandir

listView1.SmallImageList = imageList;//Asignamos la lista de imágenes al control ListView

DirectoryInfo directory = (DirectoryInfo)e.Node.Tag;//Directorio asociado al nodo seleccionado

listView1.Items.Clear();//Limpiamos los elementos del control ListView

try { foreach (DirectoryInfo d in directory.GetDirectories())

//Recorremos los subdirectorios del directorio

ListViewItem item = new ListViewItem(d.Name, 0);//Creamos un nuevo elemento con el nombre del subdirectorio y la imagen correspondiente

item.SubItems.Add(d.LastWriteTime.ToString());//Añadimos la fecha de modificación del subdirectorio

item.SubItems.Add("");//Añadimos un subelemento vacío

listView1.Items.Add(item);//Añadimos el elemento al control ListView

foreach (FileInfo f in directory.GetFiles())//Recorremos los archivos del directorio

ListViewItem item = new ListViewItem(f.Name, 2);//Creamos un nuevo elemento con el nombre del archivo y la imagen correspondiente

item.SubItems.Add(f.LastWriteTime.ToString()); //Añadimos la fecha de modificación del archivo

item.SubItems.Add(f.Length.ToString()); //Añadimos el tamaño del archivo

listView1.Items.Add(item); //Añadimos el elemento al control ListView

formularios MDI se abren con show

nombreformulario.MdiParent = this; -- para decir que ese formulario es hijo del actual, se escribe esta línea antes de mostrar el formulario nuevo

LayoutMdi(MdiLayout.”modo”) – para seleccionar el tipo de disposicion de las ventanas MDI

ListViewItem item = listView1.SelectedItems[0]; -- seleccionar un elemento de una listview

listView1.SelectedItems.Count > 0 – comprobación de elementos seleccionados

Libro libro = (Libro)item.Tag; -- convierte un elemento a un objeto determinado y poder gestionarlo como ese objeto

Agregar un elemento y subelementos a una listview

ListViewItem item = new ListViewItem(emp.Nombre);

item.SubItems.Add(emp.Apellidos);

string.IsNullOrEmpty(txtTitulo.Text) – comprobar si un elemento esta vacio o en blanco

using (var context = new BibliotecaContext()){AQUI}

para utilizar conexión a base de datos, en AQUÍ utilizar context.objetoautilizar.metodo para gestionar interactuación con base de datos (no solo eso.. el codigo que sea necesario)

var libros = context.Libros.Include(l => l.Autor).Include(l => l.Categoria).ToList(); - para filtrar

context.SaveChanges() – para modificar cambios en base de datos

para modificar un registro en bd buscarlo en el programa crear un objeto modificar el objeto y ejecutar savechanges

context.Autores.FirstOrDefault(a => a.Nombre == autor.Nombre && a.Apellidos == autor.Apellidos && a.Nacionalidad == autor.Nacionalidad); - para buscar un elemento en concreto de la base de datos

en datagridview se utiliza Row en lugar de ítems para manejar elemento seleccionado

grid.Rows.Add(args) – añadir fila introducir elementos en args separados por comas

DataGridViewRow row = dgvBiblio.SelectedRows[0]; fila seleccionada

Row.Cells[n].value.toString() – valor de celda de la fila seleccionada