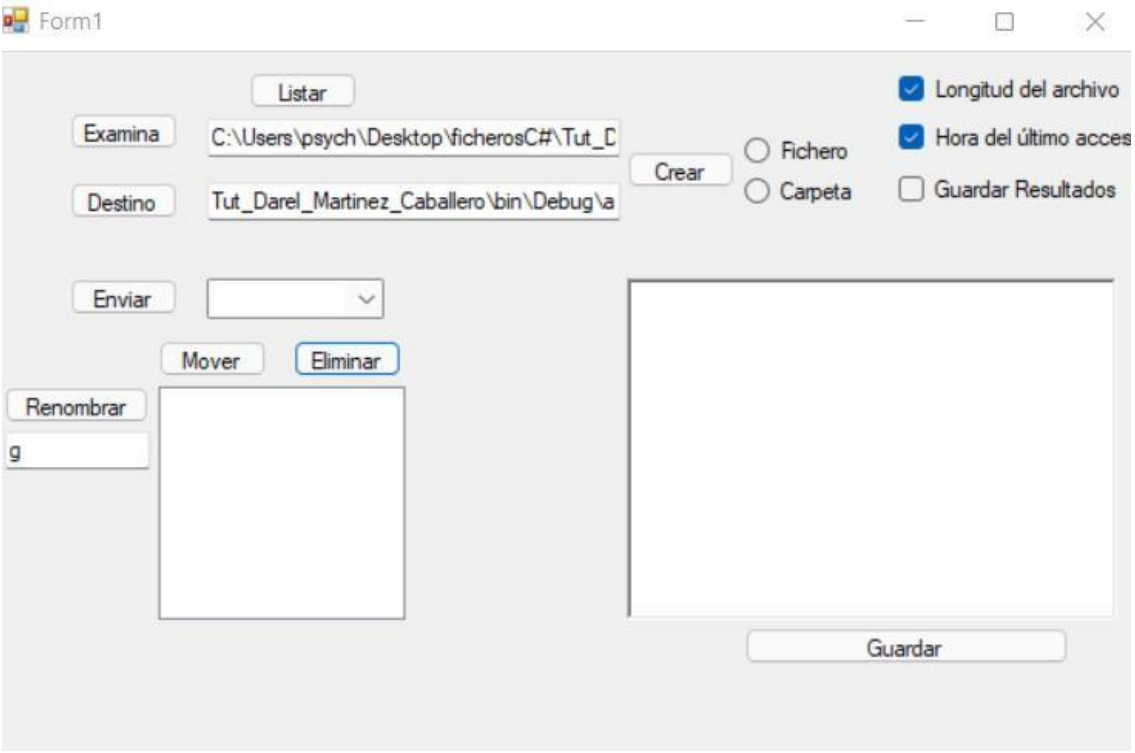
Proyecto C#

Trabajar con ficheros y carpetas



Darel Martínez Caballero

Listar ficheros y carpetas

Para ello se necesitará hacer uso de la ruta que quieres visualiza, podemos hacer uso de:

```
        Directory.GetCurrentDirectory();
```

Para obtener la ruta actual.

Además se necesita un objeto tipo lista, en este caso se hace uso de un listBox.

Podemos crear un método al que se le pase la ruta. Como en este caso.

A si podemos llamar desde donde queramos llamar a la función de listar la lista de elementos (Carpetas y ficheros). A si de manera dinámica podemos actualizar la lista de elementos.

Antes de nada nos aseguramos con :

```
            this.lbArchivos.Items.Clear();
            this.lbArchivos.Refresh();
```

De limpiar el contenido del listBox, si no lo hacemos a si al ser llamada la función se sigue rellenando la lista con nuevos datos, y quizás esto no es lo que queremos.

Métemos todo dentro de un blocke try-catch y capturamos la Exception que se produce al hacer click en las zonas donde no hay elementos, o si a la hora de listar ocurriese un error.

Como en este caso lo que nos llega al método es una ruta completa para listar,la limpiamos con Trim, por si tuviese espacios, y además nos aseguramos de que no esté vacia.

Ahora hacemos uso de DirectoryInfo y con sus métodos GetFiles() y GetDirectories()

Que nos devuelven un array de elementos. Ahora solo los recorremos y añadimos cada elemento (nombre de Fichero o Carpeta) con el método ítems.Add() del listBox. Nos debería quedar a sí el código.

```
private void actualizar_Lista(String directory)
        {
            this.lbArchivos.Items.Clear();
            this.lbArchivos.Refresh();
            try
            {
                if (directory.Trim() != String.Empty)
                {
                    DirectoryInfo di = new DirectoryInfo(directory);

                    foreach (var item in di.GetFiles())
                    {
                        this.lbArchivos.Items.Add(item.Name);
                    }

                    foreach (var direcotory in di.GetDirectories())
                    {
                        lbArchivos.Items.Add(direcotory.Name);
                    }
                }
            }
            catch (Exception ex) { }

        }
```

Crear ficheros o carpetas

Como hacemos uso de una interfaz necesitaremos de los botones tipo radio para que el usuario seleccione el tipo de elemento a crear.

Comenzamos con bloque try-catch capturando la excepción “DirectoryNotFoundException” en el caso de que la ruta no la encuentre o no sea capaz de crear el elemento.

Haciendo uso del método “Checked” podemos comprobar que radio button está marcado, y crear en función una carpeta o fichero.

Para crear un directorio

```
                        Directory.CreateDirectory(rutaFinal);
```

Para crear un fichero:

```
                        File.Create(rutaFinal);
```

Por último verificamos si se creado con el método exists y si no lo ha hecho podemos informar al usuario con MessageBox. Por último podemos llamar al método listar si queremos que se actualiza el textBox de elementos con el nuevo fichero o carpeta.

Por último nos quedaría a sí:

```
private void Crear(object sender, EventArgs e)
        {
            String rutaFinal = txtDirectory.Text + "\\" +
this.txtNombre.Text;
            try
            {
                if (rdBtnCarpeta.Checked)
                {
                    if (!Directory.Exists(rutaFinal))
                    {
                        Directory.CreateDirectory(rutaFinal);
                    }
                }
                else
                {
                    if (!File.Exists(rutaFinal))
                    {
                        File.Create(rutaFinal);
                    }
                }
            }
            catch (DirectoryNotFoundException ex) { }
            actualizar_Lista(this.txtDirectory.Text);
        }
```

Borrar archivos o carpetas

Podemos hacer uso de una función. Necesitaremos de la ruta completa incluido el nombre del fichero o carpeta.

Además con Directory.Exists() Nos aseguramos de dos cosas, si realmente el fichero se a borrado, y si realmente este fichero o carpeta existe para poder borrarlo.

Hacemos uso de:

```
System.IO.DirectoryInfo di = new DirectoryInfo(@rutaArchivo);
```

Para poder trabajar con ficheros y carpetas haciendo uso de los métodos GetFiles() y GetDirectories() obtenemos un array con los ficheros y carpetas. Al ser una carpeta y

queriendo borrar todo su contenido deberemos recorrer este array que nos proporcionan estos métodos y con el método Delete de la clase FileInfo y DirectoryInfo borramos el fichero/s y carpeta/s.

Nos quedaría de la siguiente forma:

```
private void borrar(object sender, EventArgs e)
        {
            String nombreArchivo =
this.lbArchivos.SelectedItem.ToString();
            String rutaArchivo = txtDirectory.Text + "\\" +nombreArchivo;

            if (Directory.Exists(rutaArchivo))
            {
                System.IO.DirectoryInfo di = new
DirectoryInfo(@rutaArchivo);

                foreach (FileInfo file in di.GetFiles())
                {
                    file.Delete();
                }
                foreach (DirectoryInfo dir in di.GetDirectories())
                {
                    dir.Delete(true);
                }

                di.Delete();
            }
            if (File.Exists(rutaArchivo))
            {
                File.Delete(rutaArchivo);
            }

        }
```

Renombrar archivos y carpetas

Para partir de un método. Lo primero que deberemos hacer es comprobar si el fichero o carpeta existe haciendo uso de :

```
            File.Exists(path)
```

```
           Directory.Exists(path)
```

Haciendo uso del método Move y utilizando la misma ruta del fichero podemos hacer un “swap” y renombrarlo, aunque por detrás realmente se está moviendo el fichero.

Para poder hacer esto debemos con ficheros debemos hacer uso de:

```
        System.IO.Directory.Move(directorioInicial, nuevoDirectorioName);
```

Si queremos hacerlo con ficheros:

```
        System.IO.File.Move(dirCopy, newNamePath);
```

Por último podemos hacer uso de un MessageBox para informar al usuario si a sido renombrado o si la ruta no existe. Quedando el código de la siguiente manera:

```
private void renombrar(object sender, EventArgs e)
        {
            String directory = txtDirectory.Text;
            String dirCopy = directory+ "\\" + this.nombreArchivo;
            String newNamePath = directory+"\\"+txtNombre.Text;

            string path = @"C:\";


            if (File.Exists(path))

            {
                System.IO.File.Move(dirCopy, newNamePath);
            }

            else if (Directory.Exists(path))

            {
                System.IO.Directory.Move(dirCopy, newNamePath);
            }

            else

            {

                MessageBox.Show("No existe");

            }

            actualizar_Lista(directory);
        }
```

Mover ficheros y carpetas

Similar a renombrar un fichero o carpeta con la única diferencia que el nombre es el mismo y la ruta final distinta.

Comprobamos con Exists e informamos al usuario.

Para poder mover un fichero hacemos uso de System.IO.File.Move donde el primer método es la ruta actual con el nombre del fichero y el segundo parámetro es la ruta final con el nombre del fichero.

```
System.IO.File.Move(posicionInicial+"\\"+ this.nombreArchivo,
posicionFinal+"\\"+this.nombreArchivo);
```

En el caso de que queramos hacerlo con carpetas deberemos hacer uso de System.IO.Directory.Move.

```
System.IO.Directory.Move(posicionInicial+"\\"+ this.nombreArchivo,
posicionFinal+"\\"+this.nombreArchivo);
```

Nos quedaría de la siguiente forma el método:

```
private void mover(object sender, EventArgs e)
        {
            String posicionInicial = txtDirectory.Text;
            String posicionFinal = txtDestino.Text;

            if (File.Exists(posicionInicial + "\\" +
this.nombreArchivo)){

                System.IO.File.Move(posicionInicial+"\\"+
this.nombreArchivo, posicionFinal+"\\"+this.nombreArchivo);

            }else if (Directory.Exists(posicionInicial + "\\" +
this.nombreArchivo)){

                System.IO.Directory.Move(posicionInicial + "\\" +
this.nombreArchivo, posicionFinal + "\\" + this.nombreArchivo);

            }else{
                MessageBox.Show("No existe");
            }
            actualizar_Lista(posicionInicial);
        }
```

Leer un fichero

Podemos hacer uso de un método leer como se ha hecho hasta ahora. También se pregunta por si existe o no esa ruta para tenerlo controlado.

Se introduce todo dentro de un bloque try-catch.

Se crea el stream usando StreamReader y se le pasa la ruta por parámetro

```
StreamReader sr = new StreamReader(ruta);
```

Se procede a leer línea a línea haciendo uso del método readLine que nos provee StreamReader y usando un while hasta obtener null (final del documento), cada línea se guarda en una variable. Una vez termina se le pasa el contenido de esta variable al richTextBox igualándolo a esta variable. Por último se cierra el stream.

Quedando el método de la siguiente manera:

```
public void readFile()
        {
            try
            {
                if (File.Exists(this.txtRutaExaminar.Text + "\\" +
this.nombreArchivo))
                {
                    StreamReader sr = new
StreamReader(this.txtRutaExaminar.Text + "\\" + this.nombreArchivo);

                    String lineas; ;

                    String contenido = "";
                    while ((lineas = sr.ReadLine()) != null)
                    {
                        contenido += lineas;
                    }

                    this.richTextDocumentos.Text = contenido;
                    sr.Close();
                }
            }
            catch (FileNotFoundException ex)
            {
                MessageBox.Show("El archivo no se encuentra");
            }
        }
```

Escribir en un fichero (Guardar)

Para poder escribir o guardar un fichero se hace uso de la clase StreamWriter y se le pasa la ruta completa por parámetro del fichero a escribir.

```
 public void writeFile(String result)
        {
            String contenido = result;
```

```
            StreamWriter sw = new StreamWriter(this.txtRutaExaminar.Text
+ "\\" + this.nombreArchivo);

            sw.WriteLine(contenido);
            sw.Close();

            this.richTextDocumentos.Clear();
            this.lbElementos.Refresh();
            MessageBox.Show("Fichero guardado");
        }
```