

/* ESTA CLASE ACTUALIZA UNA APLICACIÓN (.EXE) DE UN SOLO ARCHIVO QUE NO SEA MUY PESADO, UTILIZA UN ARCHIVO LLAMADO "urlserver.txt", ESTE SE ENCARGA DE BRINDAR LA DIRECCION URL DEL SERVIDOR WEB DONDE SE ENCUENTRAN LAS ACTUALIZACIONES, TAMBIEN ESTA EL ARCHIVO DE TEXTO "version.txt" QUE SE UTILIZA PARA COMPARAR LAS VERSIONES ENTRE LA LOCAL Y LA WEB, PARA DETERMINAR SI SE VA ACTUALIZAR LA APLICACIÓN.

LA CLASE TIENE CUATRO METODOS:

1. comprobaractualizacion(): verifica la existencia de los archivos y compara las versiones
2. iniciaractualizacion(): descarga el ejecutable .exe y actualiza a la nueva versión
3. strAvance() : muestra mensajes por pantalla acerca del proceso de actualización
4. ejecutar() : ejecuta el archivo ejecutable .exe

PARA IMPLEMENTAR LA CLASE:

EN EL CONSTRUCTOR DE LA CLASE SE CONFIGURA LOS NOMBRES DE LOS ARCHIVOS, UD PUEDE CAMBIARLES EL NOMBRE "version.txt", "urlserver.txt", "MiAplicacion.exe"

EL ARCHIVO "urlserver.txt" LA CLASE LO CREA AUTOMATICAMENTE SI NO EXSITE, DESPUES ABRIRLO Y COLOCAR LA DIRECCION URL COMPLETA DEL SERVIDOR DONDE ESTAN LAS ACTUALIZACIONES.

EL METODO "strAvance()" CREA UN ARCHIVO DE TEXTO "strAvance_Log.txt" DONDE SE REGISTRAN LOS PASOS Y ERRORES DEL MANEJO DE EXEPCIONES DE LA CLASE. ESTO SERVIDRA PARA CONOCER LOS MOTIVOS POR EL CUAL LA CLASE NO ACTUALIA LA APLICACIÓN.

*/

```
using System;
using System.Collections;
using System.Configuration;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Web;
using System.IO;
using System.Net;
using System.Reflection;
using System.Diagnostics;
```

```
namespace InicioPN
```

```
{
```

```
    class Actualizacion
```

```
    {
```

```
        string nameFversion;
        string nameFserver;
        string nameFejecutable;
```

```
        string VersionWeb;
        string VersionLocal;
        string RutaLocal;
        public string UrlServer;
```

```
        public Actualizacion (){ /* Constructor de la clase */
```

```
            //configuracion inicial de los nombres de archivos a manipular
            this.nameFversion = "version.txt";
            this.nameFserver = "urlserver.txt";
            this.nameFejecutable = "ReportPN.exe";
```

utiliza en el resto de la clase

```
            //para establecer las rutas de descargas
            this.RutaLocal = AppDomain.CurrentDomain.BaseDirectory + "\\\";
```

```
            if (File.Exists(this.nameFserver))
            {
```

```
                //Obtengo la Ruta del Servidor Web desde el archivo de Texto
                StreamReader FurlServer = new StreamReader(this.RutaLocal + this.nameFserver);
                this.UrlServer = FurlServer.ReadLine();
                FurlServer.Close();
```

```
}else{
```

```
//Si no existe el archivo, se crea, insertando un texto para informar que falta la Url del
```

```
Servidor
```

```
StreamWriter FurlServer = new StreamWriter(this.RutaLocal + this.nameFserver);
```

```
FurlServer.WriteLine("Agregue en una sola linea la direccion url del servidor web");
```

```
FurlServer.Close();
```

```
this.UrlServer = "N/D";
```

```
}
```

```
}
```

```
public void comprobaractualizacion()
```

```
{
```

```
    strAvance("Iniciando");
```

```
/* RutaLocal = AppDomain.CurrentDomain.BaseDirectory + "\\\"; */
```

```
//Verifico si existe el Archivo ejecutable
```

```
if (File.Exists(this.RutaLocal + this.nameFejecutable))
```

```
{
```

```
    strAvance("Archivo Ejecutable Encontrado");
```

```
//verifico si existe el archivo de texto de la version
```

```
if (File.Exists(this.RutaLocal + this.nameFversion))
```

```
{
```

```
    strAvance("Archivo Version Local Encontrado");
```

```
//abro el archivo de texto y obtengo el valor de la version
```

```
StreamReader FileVersionLocal = new StreamReader(this.RutaLocal + this.nameFversion);
```

```
VersionLocal = FileVersionLocal.ReadLine();
```

```
FileVersionLocal.Close();
```

```
try {
```

```
    //creo un objeto cliente web que me permite descargar el valor de la version web en la
```

```
ruta del server
```

```
System.Net.WebClient webClient = new System.Net.WebClient();
```

```
VersionWeb = webClient.DownloadString(this.UrlServer + this.nameFversion);
```

```
//comparo las versiones (Local vrs Web)
```

```
if (this.VersionLocal != this.VersionWeb)
```

```
{
```

```
    // si son diferentes las versiones inicio la descarga del archivo exe
```

```
strAvance("Actualizando Aplicacion");
```

```
iniciaractualizacion();
```

```
}
```

```
else
```

```
{
```

```
    // si son iguales Mando a ejecutar el archivo exe en un nuevo proceso
```

```
strAvance("No es necesario Actualizar, Iniciando Aplicacion");
```

```
Ejecutar();
```

```
}
```

```
}catch(Exception e ){
```

```
    //si hay error en la conexion, me olvido de la actualizacion y mando a jecutar el
```

```
archivo exe existente
```

```
strAvance(" A Ocurrido un Error al Momento de Comparar la Version de Actualizacion " +
```

```
e.Message.ToString());
```

```
Ejecutar();
```

```
}
```

```
}
```

```
else
```

```
{
```

```
    //si el archivo de texto "version.txt" no existe inicio la descarga del archivo exe
```

```
strAvance("Actualizando Aplicacion");
```

```
iniciaractualizacion();
```

```
}
```

```

    }
    else
    {
        //si el archivo ejecutable .exe no existe, inicio la descarga inmediatamente
        strAvance("Actualizando Aplicacion");
        iniciaractualizacion();
    }
}

//*****
*****

private void iniciaractualizacion(){

    try{

        //creo un objeto cliente web que permite descargar desde un servidor web
        WebClient webClient = new WebClient();

        //direccion url del archivo exe que se va a descargar
        string filePath = this.UrlServer + this.nameFejecutable;

        //descarga el archivo exe y le indica el directorio activo donde se va a guardar
        webClient.DownloadFile(filePath, this.RutaLocal + Path.GetFileName(filePath));

        //obtengo la ruta completa del archivo exe que se descargo
        string @Ruta = this.RutaLocal + Path.GetFileName(filePath);

        //descargo el archivo de texto de la version actual
        filePath = this.UrlServer + this.nameFversion;
        webClient.DownloadFile(filePath, this.RutaLocal + Path.GetFileName(filePath));

        //libera los recursos utiizados por este objeto
        webClient.Dispose();

        //mando a ejecutar el archivo exe en un nuevo proceso
        Ejecutar();

        strAvance("Actualizacion Realizada con exito");

    }catch(Exception e ){
        strAvance(" A ocurrido un error al Momento de actualizar la Aplicacion " +
e.Message.ToString());

        if (File.Exists(this.RutaLocal + this.nameFejecutable))
        {
            Ejecutar();
        }
    }
}

//*****
*****

public void strAvance(string Prmmsg)
{
    // muestra mensajes por pantalla acerca del proceso de comprobacion y actualizacion
    string cad = "";
    string msg = Prmmsg;
    int ancho = 79;

    int margen = 10;

    if (msg.Length <= ancho)

```

```

        {
            Console.WriteLine(cad.PadLeft(margen, ' ') + msg + cad.PadLeft((ancho - margen - msg.Length), '
''));
        }
        else
        {
            Console.WriteLine(cad.PadLeft(margen, ' ') + msg);
        }

//Guardar el Mensaje en el Log

string log_text = Prmmmsg + "\n";

if (File.Exists("strAvance_Log.txt"))
{
    StreamWriter F_log = File.CreateText(this.RutaLocal + "strAvance_Log.txt");
    F_log.Close();
    File.WriteAllText(this.RutaLocal + "strAvance_Log.txt", log_text);
}
else
{
    File.WriteAllText(this.RutaLocal + "strAvance_Log.txt", log_text);
}

}

//*****
*****

public void Ejecutar()
{
    //ejecuta el archivo .exe en un nuevo proceso
    Process Proceso = new Process();
    ProcessStartInfo sti = new ProcessStartInfo(this.RutaLocal + this.nameFejecutable);
    Proceso.StartInfo = sti;
    Proceso.Start();

}

}

}

```