**UNIVERSIDAD TECNOLÓGICA DE TULA TEPEJI**

**DESARROLLO DE SOFTWARE MULTIPLATAFORMA**

**APLICACIONES PARA 14.0**

**Protocolo FTP:**

El protocolo de transferencia de archivos permite transferir archivos directamente de un dispositivo a otro. Puedes configurar un servidor FTP de manera local en tu pc o mediante un hoster que ya lo incluya, como lo es InfinityFree. Algunas aplicaciones de escritorio hacen esta función como lo es FileZilla.

Interfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamente

Como se puede observar, el hoster nos da nuestro usuario, contraseña, puerto y servidor para poder utilizar FTP.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamenteEn esta sección la variable “username” almacena el nombre de usuario para autenticarse con el servidor FTP de tu hoster. De igual forma la contraseña y el servidor son los que nos proporciona nuestro hoster.

En la sección anterior se crea un combinado de la URL del servidor FTP con el nombre del archivo que se ha seleccionado de manera local. Creamos una instancia de FtpWebRequest utilizando la ruta completa del archivo en el servidor FTP (ftpFilePath). Se establece el método de la solicitud como WebRequestMethods.Ftp.UploadFile, para indicar que se realizará la carga de un archivo. Proporcionamos las credenciales de nuestro hoster para autenticarnos con el servidor FTP, utilizamos using para abrir un flujo de entrada hacia el archivo local que ya seleccionamos y un flujo de salida hacia el servidor FTP. Y finalmente fileStream.CopyTo(ftpStream) copia el contenido del archivo local al servidor FTP subiéndolo al administrador de archivos de nuestro hoster.

**Código:**

using System;

using System.IO;

using System.Net;

using System.Windows.Forms;

using static System.Windows.Forms.DataFormats;

namespace InterfazFTP

{

public partial class Form1 : Form

{

private string ftpUsername = "if0\_35765689";

private string ftpPassword = "xgiQ120qJE0WR6";

private string ftpServer = "ftp://ftpupload.net/";

public Form1()

{

InitializeComponent();

}

private void btnBuscar\_Click(object sender, EventArgs e)

{

OpenFileDialog openFileDialog = new OpenFileDialog();

openFileDialog.Title = "Seleccionar archivo";

openFileDialog.Filter = "Todos los archivos|\*.\*";

// Mostrar el diálogo y verificar si el usuario seleccionó un archivo

if (openFileDialog.ShowDialog() == DialogResult.OK)

{

// Mostrar la ruta del archivo seleccionado en el TextBox

txtRuta.Text = openFileDialog.FileName;

btnSubir.Enabled = true;

}

}

private void btnSubir\_Click(object sender, EventArgs e)

{

try

{

// Combinar la ruta FTP con el nombre del archivo

string ftpFilePath = ftpServer + Path.GetFileName(txtRuta.Text);

// Crear la solicitud FTP

FtpWebRequest request = (FtpWebRequest)WebRequest.Create(ftpFilePath);

request.Method = WebRequestMethods.Ftp.UploadFile;

request.Credentials = new NetworkCredential(ftpUsername, ftpPassword);

// Leer el archivo local y cargarlo al servidor FTP

using (Stream fileStream = File.OpenRead(txtRuta.Text))

using (Stream ftpStream = request.GetRequestStream())

{

fileStream.CopyTo(ftpStream);

}

MessageBox.Show("Archivo subido exitosamente.", "Éxito", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception ex)

{

MessageBox.Show($"Error al subir el archivo: {ex.Message}", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void txtRuta\_TextChanged(object sender, EventArgs e)

{

}

private void btnVisualizar\_Click(object sender, EventArgs e)

{

Visualizar visualizar = new Visualizar(ftpUsername, ftpPassword, ftpServer);

visualizar.ShowDialog();

}

}

}

**Protocolo SFTP:**

El protocolo de transferencia segura de archivos permite transferir archivos cifrados entre tu PC local y el espacio web a través del Secure Shell (ssh). La diferencia de este protocolo y el FTP es que este tiene mayor seguridad al cifrar los datos.

**Protocolo HTTPS:**

El protocolo de transferencia de hipertexto seguro es la versión segura de HTTP. Este protocolo es utilizado para enviar datos entre un navegador web y un sitio web, de igual forma, está encriptado para aumentar la seguridad de las transferencias de datos para evitar que se difunda nuestra información por la web. Utiliza el puerto 443 mientras que el HTTP utiliza el puerto 80.

**Protocolo SMTP:**

El protocolo simple de transferencia de correo se utiliza para enviar y recibir mensajes de correo electrónico a través de internet. Es uno de los más utilizados, ya que permite que los mensajes se transfieran de manera fiable entre diferentes servidores de correo.

Texto

Descripción generada automáticamente

En este código de mi aplicación utilizo un NuGet llamado MailKit para gestionar el envío de correos electrónicos mediante el protocolo SMTP. Configuro mis credenciales con mi correo, el puerto y la contraseña de acuerdo con la documentación de Gmail para crear un mensaje con la información del remitente, destinatario, el asunto y el mensaje. Se establece una conexión con el servidor SMTP, se autentica el usuario, se envía el mensaje y se cierra la conexión. En caso de que haya algún inconveniente, se captura la excepción mostrando un mensaje.

**Protocolo IMAP:**

El protocolo de acceso a mensajes de internet permite el acceso desde cualquier computadora con conexión a internet a los mensajes que se encuentran almacenados en un servidor. Únicamente se ejecuta para permitir que ingreses tus correos y no realizar envíos.

Texto

Descripción generada automáticamente

En el código anterior también utilizo el NuGet de MailKit para implementar funcionalidades del protocolo IMAP (en este caso la recepción de correos). Utilizo un listbox para que permita filtrar y visualizar los correos recibidos. Al hacer click en el botón de cargar se inicia el proceso de conexión al servidor IMAP de Gmail, autenticación del usuario y carga los primeros 5 recibidos. La lista de correos se actualiza en tiempo real permitiendo buscar un correo específico.

**Protocolo SSH:**

El protocolo Secure Shell es un protocolo de administración remota que permite controlar y modificar servidores remotos a través de internet. Podemos conectarnos con servidores mediante internet como vía para las comunicaciones, también identifica al usuario de manera remota.

**Protocolo SET:**

El protocolo Secure Electronic Transaction se utiliza cuando se realiza una transacción segura, los datos del cliente son enviados al servidor del vendedor. Funciona porque se basa en el uso de una firma electrónica del comprador y una transacción que involucra al comprador, vendedor y a su respectivo banco, protege el numero de la tarjeta del titular.

**Protocolo SSL:**

El protocolo de Secure Sockets Layer, también denominados “certificados digitales”, se utilizan para establecer una conexión cifrada entre un navegador o el ordenador de un usuario y un servidor o un sitio web. Permite cifrar el tráfico de datos entre un navegador web protegiendo la conexión, en pocas palabras permite tener una conexión segura en tu sitio o cualquier sitio que puede ser fácilmente identificado por el icono de un candado en el sitio.