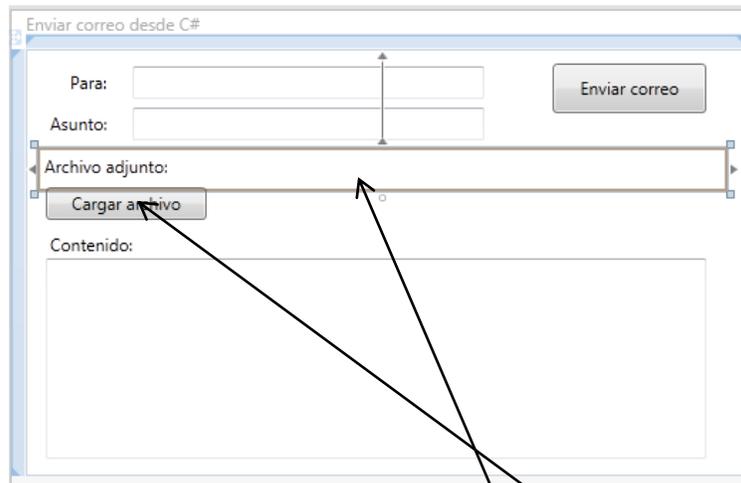


Enviar correo electrónico con archivo adjunto desde C# (Hotmail/outlook)

Este es continuación del ejemplo anterior... en el ejemplo anterior solo se podía enviar un correo sencillo, pero que pasa si necesitamos enviar un archivo adjunto?... para eso modificaremos el que ya teníamos y en este ejemplo veremos como.

Al formulario que ya habíamos trabajado agregaremos un control "Label" y le daremos el formato como se muestra en la imagen, debajo del el colocaremos un botón, es cuestión de acomodar los controles para que entren en el formulario que ya tenemos y se pueda trabajar sin problemas.



```
<TextBox Height="23" HorizontalAlignment="Left" Margin="75,12,0,0" Name="txTo" VerticalAlignment="Top" Width="247" />
<Label Content="Para:" Height="28" HorizontalAlignment="Left" Margin="26,10,0,0" Name="label1" VerticalAlignment="Top" />
<TextBox Height="23" HorizontalAlignment="Left" Margin="75,41,0,0" Name="txSubject" VerticalAlignment="Top" Width="247" />
<Label Content="Asunto:" Height="28" HorizontalAlignment="Left" Margin="12,39,0,0" Name="label2" VerticalAlignment="Top" />
<TextBox Height="141" HorizontalAlignment="Left" Margin="14,147,0,0" Name="txBody" VerticalAlignment="Top" Width="464" />
<Label Content="Contenido:" Height="28" HorizontalAlignment="Left" Margin="12,124,0,0" Name="label3" VerticalAlignment="Top" />
<Label Content="Label" Height="96" HorizontalAlignment="Left" Margin="102,14,0,0" Name="label4" VerticalAlignment="Top" />
<Button Content="Enviar correo" Height="35" HorizontalAlignment="Left" Margin="370,10,0,0" Name="btnSend" VerticalAlignment="Top" />
<Label Content="Archivo adjunto:" Height="31" Margin="8,69,12,0" Name="lbAdjunto" VerticalAlignment="Top" />
<Button Content="Cargar archivo" Height="23" HorizontalAlignment="Left" Margin="14,97,0,0" Name="btnAdjuntar" VerticalAlignment="Top" />
</Grid>
```

Primero necesitamos agregar una clase para utilizar el openFileDialog, ya que WPF no nos proporciona de manera directa un control que haga esta tarea.

```
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using Microsoft.Win32;
```

Necesitamos tambien un atributo al que llamaremos resultFileAdjunto el cual nos indicará si se ha adjuntado un archivo y asi saber a cual sendMail invocar.

```
namespace MailWPF
{
    /// <summary>
    /// Lógica de interacción para MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        Nullable<bool> resultFileAdjunto;
        string pathFileAdjunto;

        public MainWindow()
        {
            InitializeComponent();
        }
    }
}
```

Asi mismo un atributo mas donde guardaremos la ruta del archivo cargado, esto con el fin de poder mandarlo como parámetro al invocar el método sendMail.

En el botón que acabamos de agregar debemos cargar el archivo que queremos adjuntar, para ello en dicho botón agregamos el siguiente código:

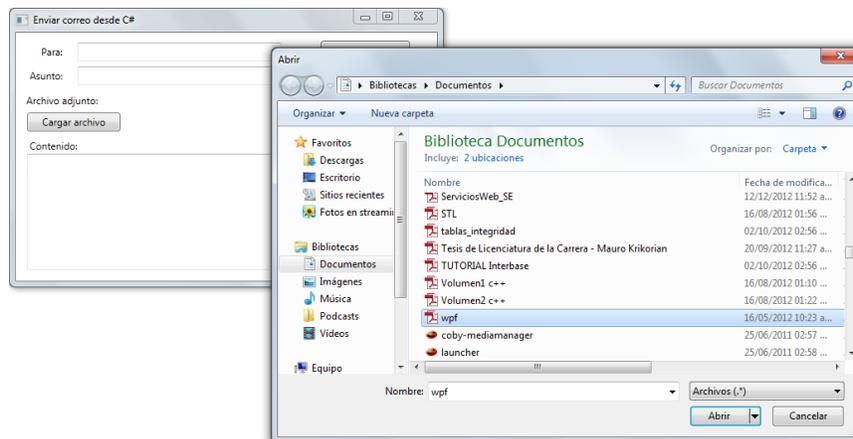
```
private void btnAdjuntar_Click(object sender, RoutedEventArgs e)
{
    OpenFileDialog dlg = new OpenFileDialog();

    // establecemos los filtros en este caso no buscamos un tipo en específico, así que ponemos "*"
    dlg.DefaultExt = ".*";
    dlg.Filter = "Archivos (*.*)|*.*";

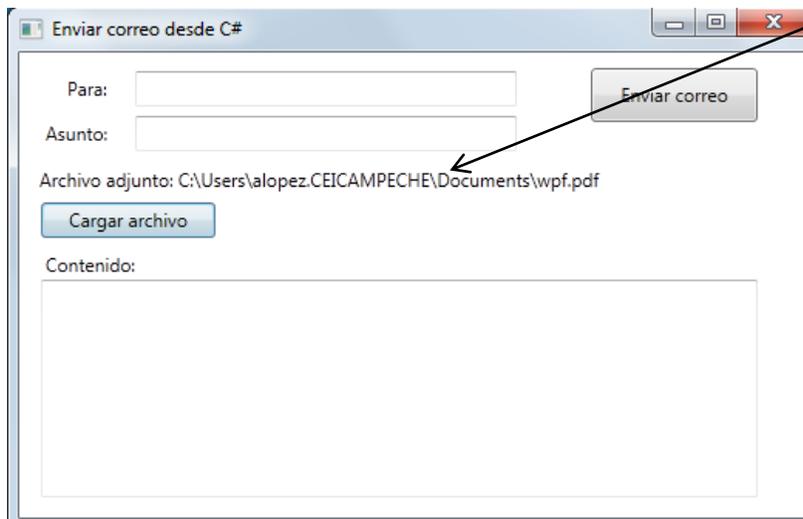
    // Mostramos el dialogo y guardamos el resultado del mismo
    resultFileAdjunto = dlg.ShowDialog();

    // Si se seleccionó un archivo, entonces mostramos el path en el label correspondiente
    if (resultFileAdjunto == true)
    {
        // Open document
        pathFileAdjunto = dlg.FileName;
        lbAdjunto.Content += " " + pathFileAdjunto;
    }
}
```

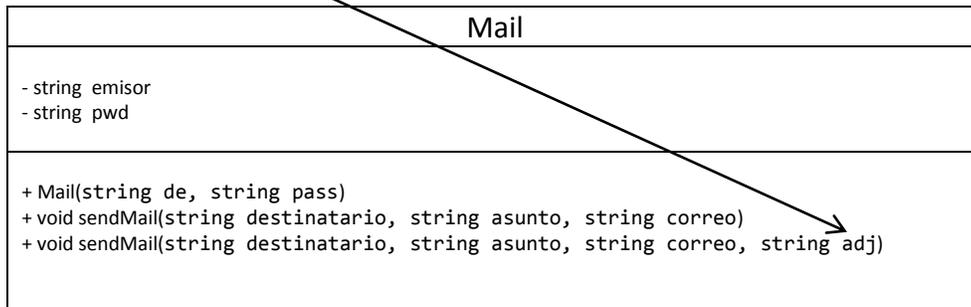
Con esto podremos abrir un cuadro de dialogo para abrir archivos, como se muestra en la figura.



En este caso estamos cargando un PDF que se llama wpf, después de cargarlo se mostrará el path del archivo cargado.



Sin embargo hasta ahora solo lo hemos seleccionado, lo que necesitamos es adjuntarlo en el correo que se va a enviar, para ello sobrecargaremos el método sendMail de la clase Mail, agregando un parámetro mas que indicará precisamente el archivo a adjuntar.



El método que agregaremos es el siguiente:

```

/// <summary>
/// Se envía un correo electronico con archivo adjunto
/// </summary>
/// <param name="destinatario">Quien escribe el correo</param>
/// <param name="asunto">Asunto del correo</param>
/// <param name="correo">Cuerpo del correo (contenido)</param>
/// <param name="adjunto">Direccion del archivo adjunto</param>
public void sendMail(string destinatario, string asunto, string correo, string adj)
{
    MailMessage msg = new MailMessage();
    //Quien escribe al correo
    msg.From = new MailAddress(emisor);
    //A quien va dirigido
    msg.To.Add(new MailAddress(destinatario));
    //Asunto
    msg.Subject = asunto;
    //Contenido del correo
    msg.Body = correo;
    //Adjuntamos archivo
    msg.Attachments.Add(new Attachment(adj, System.Net.Mime.MediaTypeNames.Application.Pdf));

    //Servidor smtp de hotmail
    SmtplibClient clienteSmtplib = new SmtplibClient();
    clienteSmtplib.Host = "smtp.live.com";
    clienteSmtplib.Port = 25;
    clienteSmtplib.EnableSsl = false;
    clienteSmtplib.UseDefaultCredentials = false;

    //Se envia el correo
    clienteSmtplib.Credentials = new NetworkCredential(emisor, pwd);
    clienteSmtplib.EnableSsl = true;

    try
    {
        clienteSmtplib.Send(msg);
        MessageBox.Show("Correo enviado");
    }

    catch (Exception ex)
    {
        MessageBox.Show("Error al enviar el correo " + ex.Message);
    }
}

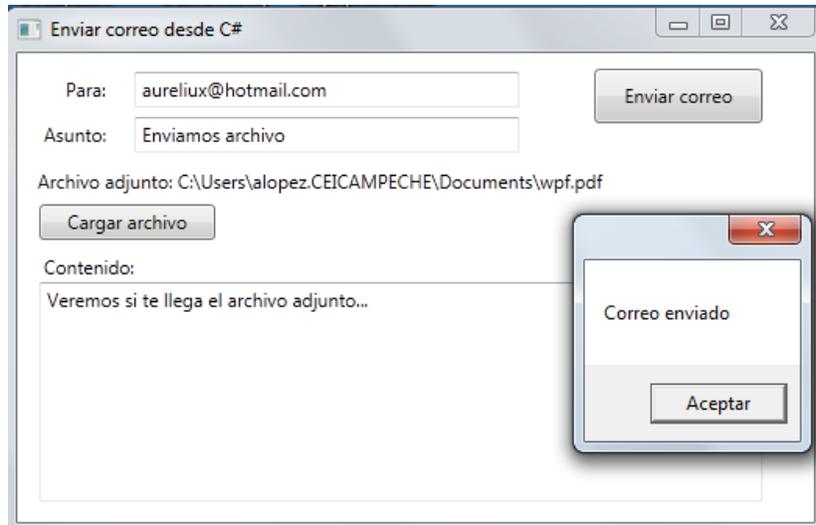
```

En el botón colocamos el siguiente código

```
private void btnSend_Click(object sender, RoutedEventArgs e)
{
    Mail m = new Mail("ceicampeche@hotmail.com", "██████████");
    if (resultFileAdjunto == true)
        m.sendMail(txTo.Text, txSubject.Text, txBody.Text);
    else
        m.sendMail(txTo.Text, txSubject.Text, txBody.Text, pathFileAdjunto);
}
```

Nota: En el código se ha borrado la contraseña, pero se entiende que el dato es la contraseña de la cuenta.

Ahora si ya podemos hacer una prueba adjuntando un archivo y mandándolo a una cuenta de correo.



Vemos que el correo ha llegado satisfactoriamente a la bandeja de entrada con el archivo adjunto.

Enviarnos archivo



1 dato adjunto (854.9 kB)

Outlook Vista activa



Descargar como zip

Veremos si te llega el archivo adjunto