

# **Técnicas Avanzadas**



# Temas a tratar:

5.1- Manipulación de Archivos y streams.

5.2- Acceso de Internet.

5.3- Envío de Email.

5.4- Colección de datos.

5.5- Colecciones Genéricas.

## 5.1.- Manipulación de archivos y streams

StreamReader está diseñado para la entrada de caracteres en una codificación determinada, mientras que la Stream clase está diseñada para la entrada y salida de bytes.

Utilice StreamReader para leer líneas de información desde un archivo de texto estándar.

```
namespace ConsoleApplication10
{
    class Program
    {
        static void Main(string[] args)
        {
            StreamWriter SW;
            SW = File.CreateText(@"C:\text.txt");

            SW.WriteLine("Hello C#1");
            SW.WriteLine("Hello C#2");
            SW.WriteLine("Hello C#3");
            SW.WriteLine("Hello C#4");
            SW.WriteLine("Hello C#5");

            SW.Close();

            StreamReader SR;
            SR = File.OpenText(@"C:\text.txt");
            Console.WriteLine(SR.ReadLine());
        }
    }
}
```

## 5.3.- Envío de Email

- **MailMessage:** Inicializa una instancia vacía de la clase *System.Net.Mail* el cual permite crear y enviar un correo por medio de SMTP.
- **SmtpClient:** Inicializa una nueva instancia de la clase *System.Net.Mail* que permite enviar un correo.

```
MailMessage mailMessage = new MailMessage();
mailMessage.From = new MailAddress("email@yourblog.com");
mailMessage.To.Add("youremail@gmail.com");
mailMessage.Subject = txtSubject.Text;

mailMessage.Body = "<b>Sender Name : </b>" + txtName.Text + "<br/>"
    + "<b>Sender Email : </b>" + txtEmail.Text + "<br/>"
    + "<b>Comments : </b>" + txtComments.Text;
mailMessage.IsBodyHtml = true;

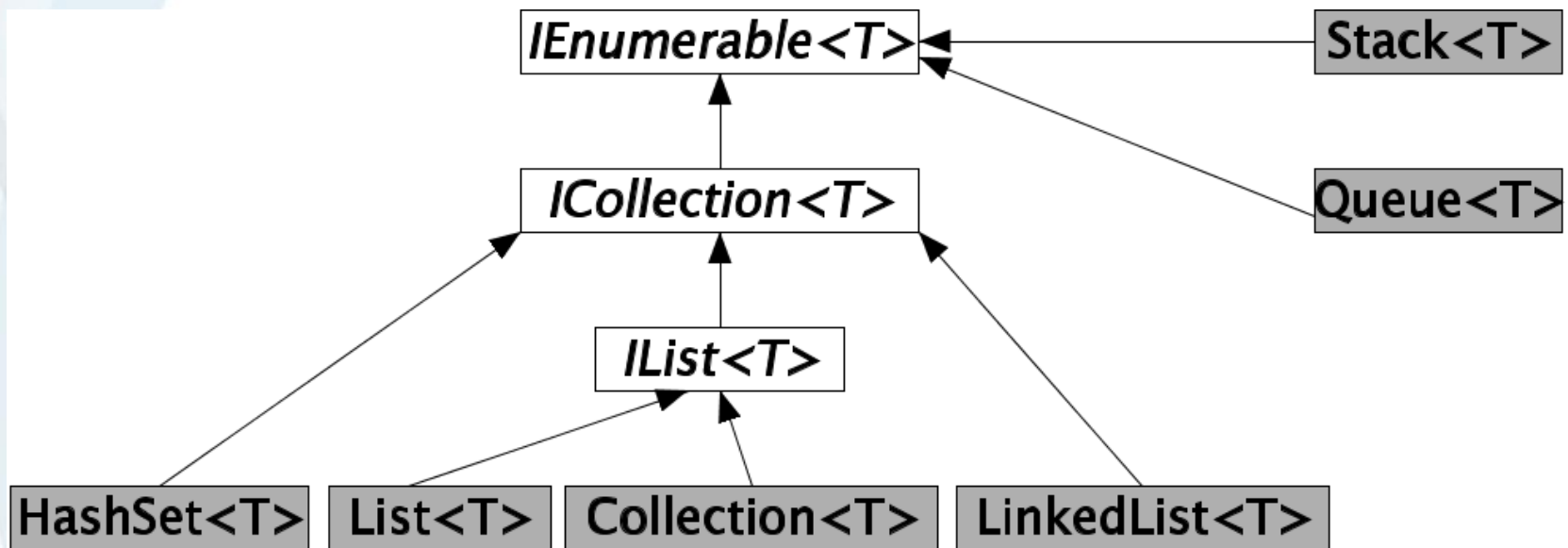
SmtpClient smtpClient = new SmtpClient("localhost", 25);
smtpClient.Send(mailMessage);

lblMessage.ForeColor = System.Drawing.Color.Blue;
lblMessage.Text = "Thank you, your email has been sent";
```

## 5.4.- Colecciones de datos

Las colecciones se basan en las interfaces `ICollection`, `IList`, `IDictionary` o en sus equivalentes genéricos.

La interfaz `IList` y la interfaz `IDictionary` se derivan ambas de la interfaz `ICollection`; por lo tanto, todas las colecciones se basan en la interfaz `ICollection` directa o indirectamente.



## 5.5.- Colecciones genericas

Las colecciones genéricas se agregaron en la versión 2.0 de .NET Framework y son colecciones con seguridad de tipos en tiempo de compilación. Debido a esto, las colecciones genéricas normalmente ofrecen un mejor rendimiento.

```
private void button1_Click(object sender, EventArgs e)
{
    List<string> students = new List<string>();

    students.Add("Jenny");
    students.Add("Peter");
    students.Add("Mary Jane");
}
```