

# Repaso C#.NET

Sistema de Aprendizaje Automático



**C#:** <https://learn.microsoft.com/es-es/dotnet/csharp/>

**Documentación de C#:**

<https://learn.microsoft.com/es-es/dotnet/csharp/tour-of-csharp/>

**Paseo por el lenguaje C#:**

<https://learn.microsoft.com/es-es/dotnet/csharp/tour-of-csharp/overview>

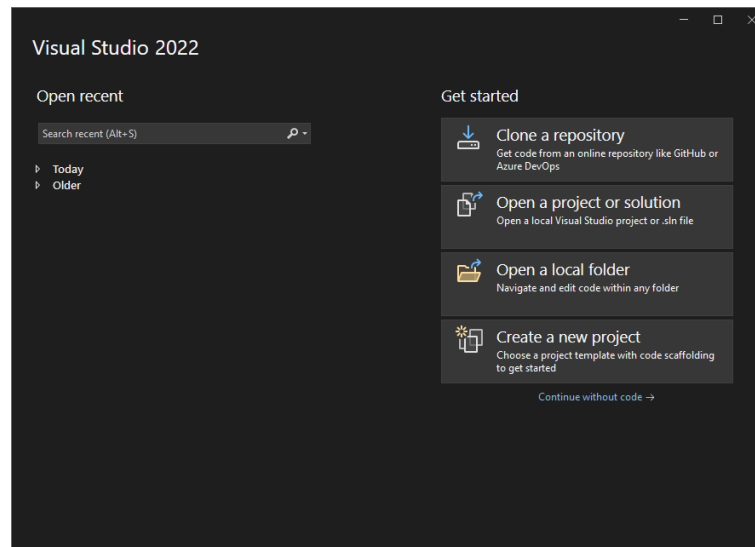
## Índice

---

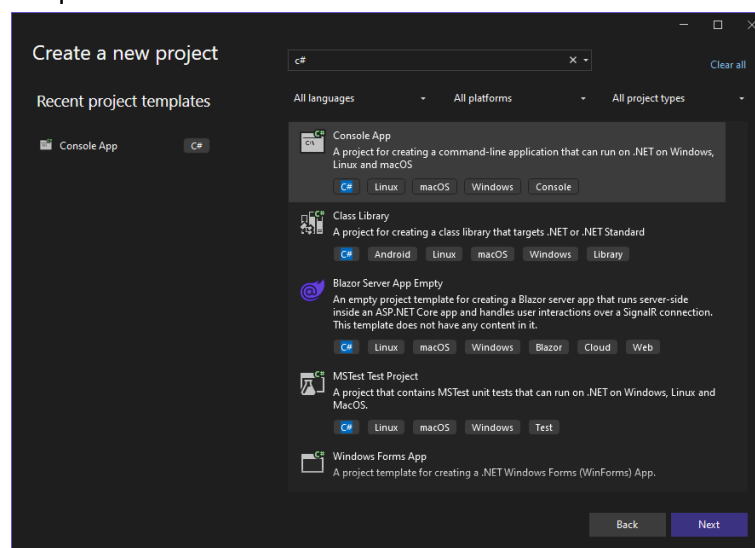
<b>Creación de Proyecto en Visual Studio</b>	<b>1</b>
Actividad	3
<b>Ejecución del Proyecto en Visual Studio</b>	<b>4</b>
Actividad	4
<b>Declaración de variables</b>	<b>5</b>
Actividad	5
<b>Mostrar/Pedir información por pantalla</b>	<b>6</b>
Actividad	6
<b>Condicionales</b>	<b>7</b>
Actividad	7
<b>Bucles</b>	<b>8</b>
Actividad	8
<b>Arrays</b>	<b>10</b>
Actividad	10
<b>Clases y Structs</b>	<b>11</b>
Actividad	12
<b>Leer de ficheros</b>	<b>13</b>
Actividad	15

## Creación de Proyecto en Visual Studio

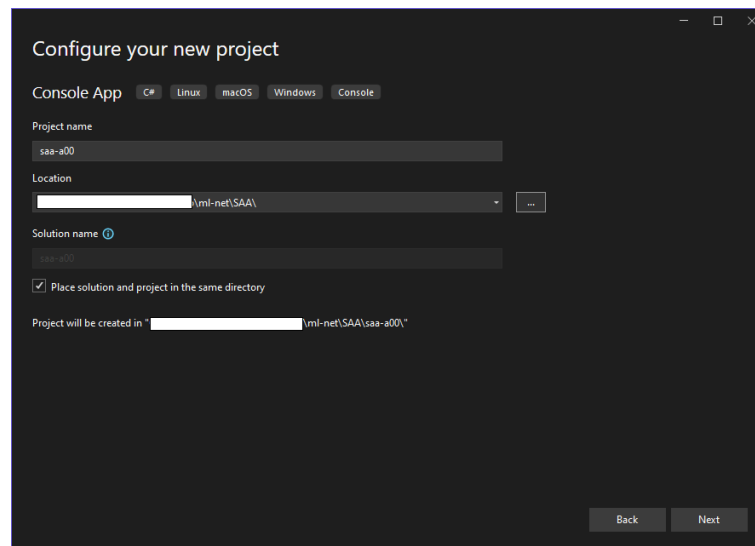
Una vez iniciada la aplicación, hacemos clic en “**Create a new project**”.



Vamos a usar una plantilla, en este caso vamos a utilizar “**Console App**” con el lenguaje C#. Para asegurarnos de que es C#, en el buscador escribimos C# y veremos en azul que efectivamente es la que necesitamos. Lo seleccionamos.



Una vez hayamos hecho clic en “**Next**”, establecemos un nombre para el proyecto. En este caso será “saa-a00” (sistemas de aprendizaje automático - actividad 00). Indicamos la ruta donde se va a crear (os recomiendo que creéis una ruta para las actividad de SAA). Y debemos marcar el check “**Place solution and project in the same directory**”.

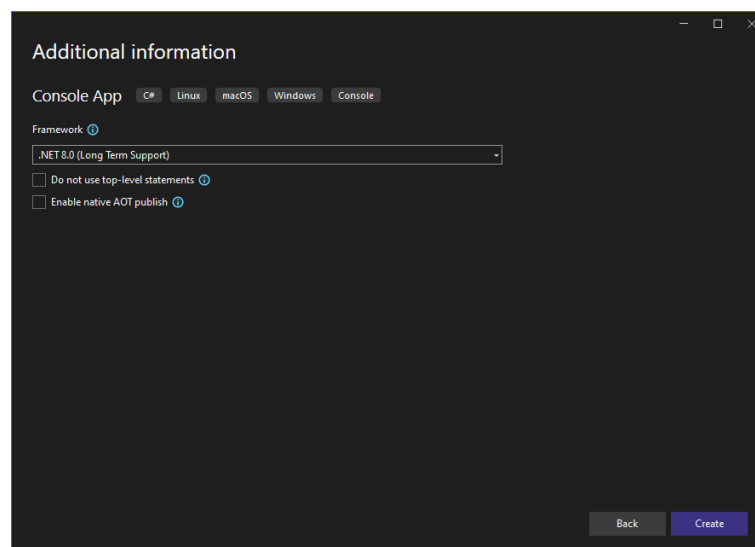


Una vez hecho clic en “**Next**”, debemos seleccionar la versión del framework. En este caso vamos a usar .NET 8 (LTS).

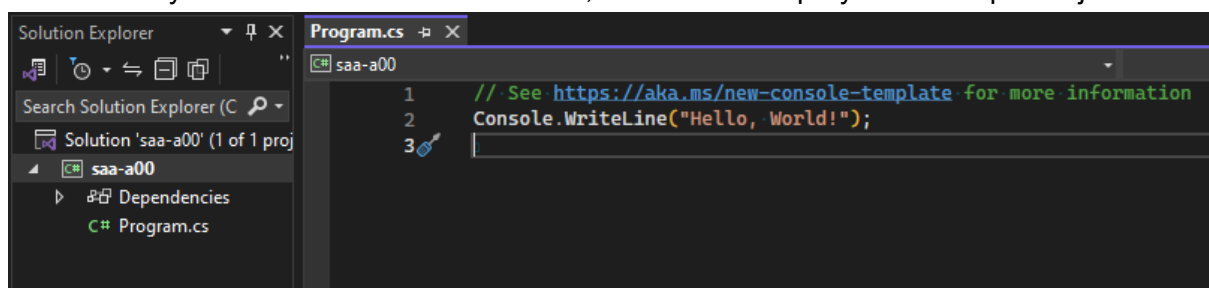


La versión **.NET 9** está a punto de salir.

<https://learn.microsoft.com/es-es/dotnet/core/whats-new/dotnet-9/overview>



Una vez hayamos hecho clic en “**Create**”, tendremos el proyecto listo para ejecutarse.



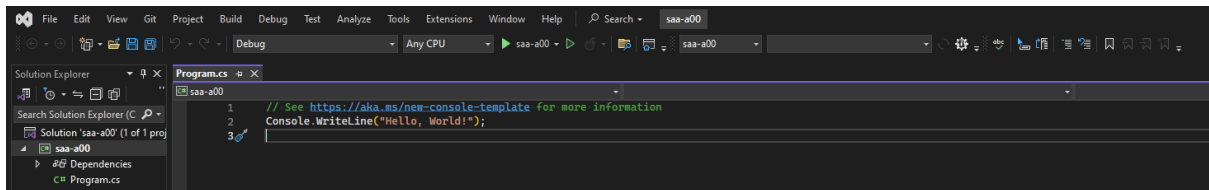
## Actividad

Crear un proyecto y mostrar el contenido de “**Program.cs**”:

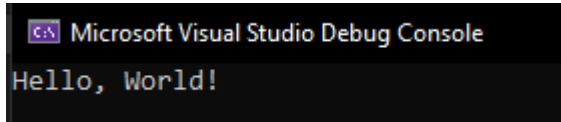
```
// See https://aka.ms/new-console-template for more information
Console.WriteLine("Hello, World!");
```

## Ejecución del Proyecto en Visual Studio

Una vez creado el proyecto, si hacemos clic en el botón verde (*triángulo*), también podemos hacer uso de la tecla **F5**, se ejecutará el programa.

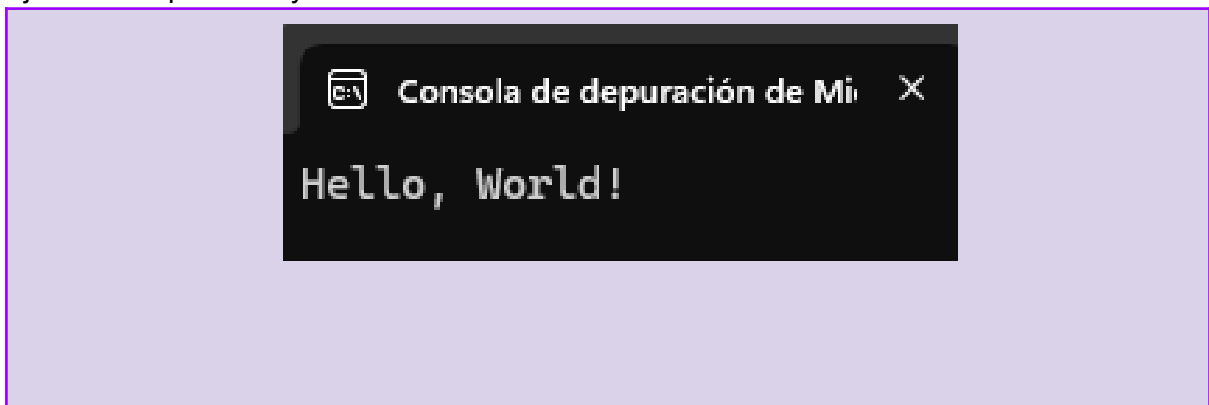


En mi caso se ejecuta en una terminal:



## Actividad

Ejecutar la aplicación y mostrar la consola entera:



## Declaración de variables

En el siguiente código podemos ver diferentes tipos de variables y operaciones:

```
Program.cs
saa-a00
1 // Ejercicios con variables.
2
3 string nombre = "Pedro";
4
5 Console.WriteLine("Hola, " + nombre);
6 Console.WriteLine($"Hola, {nombre}"); // Mejor esta forma.
7
8
9 float temperaturaActual = -273.15f;
10 char simboloTemperatura = 'K';
11
12 Console.WriteLine($"La temperatura es de {temperaturaActual} {simboloTemperatura} (Kelvin).");
13
14 int operando1 = 15;
15 int operando2 = 5;
16
17 Console.WriteLine($"{operando1} + {operando2} = {operando1 + operando2}.");
18 Console.WriteLine($"{operando1} - {operando2} = {operando1 - operando2}.");
19 Console.WriteLine($"{operando1} * {operando2} = {operando1 * operando2}.");
20 Console.WriteLine($"{operando1} / {operando2} = {operando1 / operando2}.");
21
```

## Actividad

1. Crear un programa que muestre la tabla de multiplicar del 5.
2. Concatenar vuestro nombre, apellidos y edad en la misma línea (cada valor es una variable).

```
for(var i = 1; i <= 10; i++)
var nombre = "Carlos";
var apellidos = "Robledo Serradilla";
var edad = 24;

Console.WriteLine(nombre + " " + apellidos + ", " + edad);

Console de depuración de Mi
Carlos Robledo Serradilla, 24
5*6 = 30
5*7 = 35
5*8 = 40
5*9 = 45
5*10 = 50
```

## Mostrar/Pedir información por pantalla

Podemos hacer uso de **Console.Write** o **Console.WriteLine**. El primero no hace nueva línea (el equivalente a la tecla enter) y el segundo crea una nueva línea.

Para leer datos de la consola, debemos hacer uso de **Console.ReadLine**. Este devuelve el valor en forma de **string**, concretamente **string?**, es decir, un string que puede ser **null**.

```
Console.Write("Escribe una marca: ");  
string? marca = Console.ReadLine();  
  
Console.WriteLine("La marca que ha escrito el usuario es: " + marca);
```

Si lo ejecutamos tenemos que:

```
Microsoft Visual Studio Debug Console  
Escribe una marca: nike  
La marca que ha escrito el usuario es: nike
```



C# tiene el concepto de **tipo nullable**, para ello se hace uso del operador **?**.

Esto permite escribir código más seguro, de tal modo que el programador en base a esta característica se verá forzado a hacer las comprobaciones evitando crasheos del programa por **NullReferenceException**.

<https://learn.microsoft.com/es-es/dotnet/csharp/language-reference/builtin-types/nullable-value-types>

## Actividad

3. Crear un programa que muestre la tabla de multiplicar de un número que pida por pantalla.
4. Concatenar vuestro nombre, apellidos y edad en la misma línea, cada valor es una variable que pide al usuario.

```
Console.Write("Escribe un número: ");
string? num_text = Console.ReadLine();
int num;
if(int.TryParse(num_text, out num)) {
    for (var i = 1; i <= 10; i++)
    {
        Console.WriteLine($"{num} * {i} = {num*i}");
    }
}
```

Consola de depuración de Mi × + ▾

Escribe un número: 8

```
8 * 1 = 8
8 * 2 = 16
8 * 3 = 24
8 * 4 = 32
8 * 5 = 40
8 * 6 = 48
8 * 7 = 56
8 * 8 = 64
8 * 9 = 72
8 * 10 = 80
```

```
Console.Write("Introduce tu nombre: ");
string? nombre = Console.ReadLine();
if(nombre != null)
{
    Console.Write("Introduce tus apellidos: ");
    string? apellidos = Console.ReadLine();
    if (apellidos != null)
    {
        Console.Write("Introduce tu edad: ");
        string? edad = Console.ReadLine();
        if (edad != null)
        {
            Console.WriteLine($"{nombre} {apellidos}, {edad}");
        }
    }
}
```

Consola de depuración de Mi × + ▾

```
Introduce tu nombre: Carlos
Introduce tus apellidos: Robledo Serradilla
Introduce tu edad: 24
Carlos Robledo Serradilla, 24
```



## Condicionales

```
float temperaturaActual = 8f;

if (temperaturaActual < 10)
{
    Console.WriteLine("Hace frío");
}
```

Si lo ejecutamos:

```
C:\> Microsoft Visual Studio Debug Console
Hace frío
```

## Actividad

Crear un programa que:

1. Pida al usuario un número
2. Pida al usuario un segundo número
3. Muestre si son iguales
4. Muestre si el primero es mayor que el segundo
5. Muestre si el segundo es menor o igual que el primero
6. Muestre si son distintos

```
Console.WriteLine("Escribe un número: ");
string? num_text = Console.ReadLine();
int num;
if(int.TryParse(num_text, out num)) {
    Console.WriteLine("Escribe un segundo número: ");
    string? num_text_dos = Console.ReadLine();
    int num_dos;
    if(int.TryParse(num_text_dos, out num_dos))
    {
        if(num_dos == num)
        {
            Console.WriteLine("Ambos números son iguales");
        }
        else
        {
            Console.WriteLine("Los números son distintos, ");
            if (num > num_dos)
            {
                Console.WriteLine($"{num} es mayor que {num_dos}");
            }
            else
            {
                Console.WriteLine($"{num} es menor que {num_dos}");
            }
        }
    }
}
```

Consola de depuración de Mi

Escribe un número: 1  
Escribe un segundo número: 7  
Los números son distintos, 1 es menor que 7

Escribe un número: 6  
Escribe un segundo número: 3  
Los números son distintos, 6 es mayor que 3

Escribe un número: 5  
Escribe un segundo número: 5  
Ambos números son iguales

## Bucles

---

```
for (int i = 0; i <= 10; i++)  
{  
    Console.WriteLine($"iteración {i}");  
}
```

Microsoft Visual Studio Debug Console

```
iteración 0  
iteración 1  
iteración 2  
iteración 3  
iteración 4  
iteración 5  
iteración 6  
iteración 7  
iteración 8  
iteración 9  
iteración 10
```

```
int i = 0;  
while (i <= 10)  
{  
    Console.WriteLine($"iteración {i}");  
    i++;  
}
```

Microsoft Visual Studio Debug Console

```
iteración 0  
iteración 1  
iteración 2  
iteración 3  
iteración 4  
iteración 5  
iteración 6  
iteración 7  
iteración 8  
iteración 9  
iteración 10
```

## Actividad

Crear un programa que pida por consola un número y calcule su tabla de multiplicar:

```
Console.Write("Escribe un número: ");
string? num_text = Console.ReadLine();
int num;
if(int.TryParse(num_text, out num)) {
    for (var i = 1; i <= 10; i++)
    {
        Console.WriteLine($"{num} * {i} = {num*i}");
    }
}
```

Consola de depuración de Mi X + v

Escribe un número: 8

```
8 * 1 = 8
8 * 2 = 16
8 * 3 = 24
8 * 4 = 32
8 * 5 = 40
8 * 6 = 48
8 * 7 = 56
8 * 8 = 64
8 * 9 = 72
8 * 10 = 80
```

## Arrays

---

Creación de array de enteros (int) y recorrer con for i.

```
int[] numeros = {1, 2, 3, 4, 5};  
  
for (int i = 0; i < numeros.Length; i++)  
{  
    Console.WriteLine("Elemento en el índice " + i + ": " + numeros[i]);  
}
```

```
Microsoft Visual Studio Debug Console  
Elemento en el índice 0: 1  
Elemento en el índice 1: 2  
Elemento en el índice 2: 3  
Elemento en el índice 3: 4  
Elemento en el índice 4: 5
```

Creación de array de cadenas de caracteres (string) y recorrer con for each.

```
string[] nombres = {"Juan", "Ana", "Carlos", "Lucía"};  
  
foreach (string nombre in nombres)  
{  
    Console.WriteLine("Nombre: " + nombre);  
}
```

```
Microsoft Visual Studio Debug Console  
Nombre: Juan  
Nombre: Ana  
Nombre: Carlos  
Nombre: Lucía
```

## Actividad

Crear un programa que recorre un array vacío de tamaño 5, de tipo string, y lo va rellenando, pidiendo por consola un nombre.

```
string[] nombres = new string[5];  
for (int i = 0; i < nombres.Length; i++)  
{  
    Console.WriteLine($"Introduce el nombre {i + 1}: ");  
    string? nombre = Console.ReadLine();  
    if(nombre != null)  
    {  
        nombres[i] = nombre;  
    }  
}
```

Consola de depuración de Mi X + v

```
Introduce el nombre 1: Pablo  
Introduce el nombre 2: Andrea  
Introduce el nombre 3: Miguel  
Introduce el nombre 4: Rosa  
Introduce el nombre 5: Juan
```

m

## Clases y Structs

```

Temperatura temperaturaLunes = new Temperatura(20.0f, 'C');
Console.WriteLine("ValorTemperatura: " + temperaturaLunes.ValorTemperatura);
Console.WriteLine("TipoGrado: " + temperaturaLunes.TipoGrado);

3 references
public class Temperatura
{
    /// < Propiedades de la clase
    2 references
    public double ValorTemperatura { get; set; }
    2 references
    public char TipoGrado { get; set; }

    /// < Constructor para inicializar la clase
    1 reference
    public Temperatura(double valorTemperatura, char tipoGrado)
    {
        ValorTemperatura = valorTemperatura;
        TipoGrado = tipoGrado;
    }
}

```

```

c# Microsoft Visual Studio Debug Console
ValorTemperatura: 20
TipoGrado: C

```

### Structs

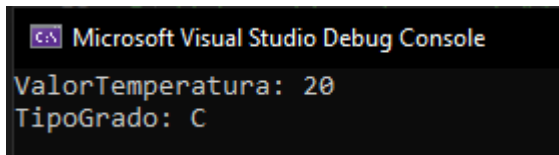
```

Temperatura temperaturaLunes = new Temperatura(20.0f, 'C');
Console.WriteLine("ValorTemperatura: " + temperaturaLunes.ValorTemperatura);
Console.WriteLine("TipoGrado: " + temperaturaLunes.TipoGrado);

3 references
public struct Temperatura
{
    /// < Propiedades del struct
    2 references
    public double ValorTemperatura { get; set; }
    2 references
    public char TipoGrado { get; set; }

    /// < Constructor del struct
    1 reference
    public Temperatura(double valorTemperatura, char tipoGrado)
    {
        ValorTemperatura = valorTemperatura;
        TipoGrado = tipoGrado;
    }
}

```



## Actividad

Crear un programa que tenga la estructura Mascota, que tiene un nombre y edad:

Crear 2 mascotas y mostrar su datos.

Crear un programa que tenga la clase Persona, que tiene un nombre, apellido1, apellido2 y edad y una Mascota:

Crear 2 personas y mostrar su datos.

Asignar a la propiedad mascota a las creadas al principio

```

1  Mascota perro = new Mascota("Elmo", 8);
2  Mascota gato = new Mascota("Gato", 4);
3  Console.WriteLine($"La primera mascota se llama {perro.nombre} y tiene {perro.edad}");
4  Console.WriteLine($"La segunda mascota se llama {gato.nombre} y tiene {gato.edad}");
5  Persona josejuan = new Persona("Jose Juan", "Correncueros", "Perez", 27, perro);
6  Persona paquita = new Persona("Paquita", "Hernandez", "Cortes", 52, gato);
7  Console.WriteLine($"La primera persona se llama {josejuan.nombre} {josejuan.apellido1} {josejuan.apellido2}, tiene {josejuan.edad} años y su mascota es {josejuan.mascota.nombre}");
8  Console.WriteLine($"La primera persona se llama {paquita.nombre} {paquita.apellido1} {paquita.apellido2}, tiene {paquita.edad} años y su mascota es {paquita.mascota.nombre}");
9
10 public struct Mascota
11 {
12     3 referencias
13     public string nombre { get; set; }
14     3 referencias
15     public int edad { get; set; }
16 }
17
18 2 referencias
19 public Mascota(string nombre, int edad)
20 {
21     this.nombre = nombre;
22     this.edad = edad;
23 }
24
25
26 5 referencias
27 public struct Persona
28 {
29     3 referencias
30     public string nombre { get; set; }
31     3 referencias
32     public string apellido1 { get; set; }
33     3 referencias
34     public string apellido2 { get; set; }
35     3 referencias
36     public int edad { get; set; }
37     3 referencias
38     public Mascota mascota { get; set; }
39 }
40
41 2 referencias
42 public Persona(string nombre, string apellido1, string apellido2, int edad, Mascota mascota)
43 {
44     this.nombre = nombre;
45     this.apellido1 = apellido1;
46     this.apellido2 = apellido2;
47     this.edad = edad;
48     this.mascota = mascota;
49 }
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

```

La primera mascota se llama Elmo y tiene 8
La segunda mascota se llama Gato y tiene 4
La primera persona se llama Jose Juan Correncueros Perez, tiene 27 años y su mascota es Elmo
La primera persona se llama Paquita Hernandez Cortes, tiene 52 años y su mascota es Gato

C:\Users\Administrador\source\repos\RepasoC#\bin\Debug\net8.0\RepasoC#.exe (proceso 13668) se cerró
Para cerrar automáticamente la consola cuando se detiene la depuración, habilite Herramientas -> Opción
Cerrar la consola automáticamente al detenerse la depuración.
Presione cualquier tecla para cerrar esta ventana. . .

```

## Leer de ficheros

---

Leer línea por línea:

```
using System;
using System.IO;

class Program
{
    static void Main()
    {
        // Especifica la ruta del archivo
        string rutaArchivo = "ruta/del/archivo.txt";

        try
        {
            // Usar StreamReader para leer el archivo línea por línea
            using (StreamReader lector = new StreamReader(rutaArchivo))
            {
                string linea;
                // Leer línea por línea hasta el final del archivo
                while ((linea = lector.ReadLine()) != null)
                {
                    Console.WriteLine(linea);
                }
            }
        }
        catch (Exception e)
        {
            // Capturar y mostrar cualquier excepción
            Console.WriteLine("Ocurrió un error al leer el archivo:");
            Console.WriteLine(e.Message);
        }
    }
}
```

Leer todo el fichero:

```
using System;
using System.IO;

class Program
{
    static void Main()
    {
        // Especifica la ruta del archivo
        string rutaArchivo = "ruta/del/archivo.txt";

        try
        {
            // Leer todo el contenido del archivo en una sola operación
            string contenido = File.ReadAllText(rutaArchivo);
            Console.WriteLine(contenido);
        }
    }
}
```



```

        catch (Exception e)
        {
            // Capturar y mostrar cualquier excepción
            Console.WriteLine("Ocurrió un error al leer el archivo:");
            Console.WriteLine(e.Message);
        }
    }
}

```

Leer todo el fichero y meterlo en un array:

```

using System;
using System.IO;

class Program
{
    static void Main()
    {
        // Especifica la ruta del archivo
        string rutaArchivo = "ruta/del/archivo.txt";

        try
        {
            // Leer todas las líneas del archivo y almacenarlas en un array
            string[] lineas = File.ReadAllLines(rutaArchivo);

            // Recorrer el array y mostrar cada línea
            foreach (string linea in lineas)
            {
                Console.WriteLine(linea);
            }
        }
        catch (Exception e)
        {
            // Capturar y mostrar cualquier excepción
            Console.WriteLine("Ocurrió un error al leer el archivo:");
            Console.WriteLine(e.Message);
        }
    }
}

```

### Importante:



**StreamReader** es útil cuando estás leyendo archivos grandes y quieres procesarlos línea por línea sin cargar todo el contenido en la memoria de una vez. **File.ReadAllText()** y **File.ReadAllLines()** son útiles para archivos pequeños o cuando quieres todo el contenido de una sola vez.

## Actividad

Crear un archivo txt que contenga, por cada línea, la temperatura de los días de la semana, cargarlo en un array, y mostrar el contenido.

```
string rutaArchivo = "C:\\Users\\Administrador\\source\\repos\\RepasoC#\\temperaturas.txt";

try
{
    string[] lineas = File.ReadAllLines(rutaArchivo);

    Console.WriteLine("Temperaturas de la semana:");
    for (int i = 0; i < lineas.Length; i++)
    {
        Console.WriteLine($"Día {i + 1}: {lineas[i]} °C");
    }
}
catch (Exception e)
{
    Console.WriteLine("Ocurrió un error al leer el archivo:");
    Console.WriteLine(e.Message);
}
```

Consola de depuración de Mi

```
Temperaturas de la semana:
Día 1: Lunes: 20 °C
Día 2: Martes 23 °C
Día 3: Miercoles: 22 °C
Día 4: Jueves: 18 °C
Día 5: Viernes: 17 °C
Día 6: Sabado: 18 °C
Día 7: Domingo: 20 °C
```