

# C# II

**Instructor:** Ing. Gabriel Roa  
**Email:** gabrielrb14a@gmail.com  
**Teléfono:** 0424-7592768  
**Fecha:** Enero del 2022

# AGENDA

- Depuración.
- Errores sintácticos vs Errores de lógica.
- Uso del bombillo.
- Modo de depuración en Visual Studio.
- Herramientas de depuración.
  - Breakpoints.
  - Navegación por el código.
  - Visor de objetos.
- Ejercicios.

# Depuración

- ¿Cómo detecta un programador los errores? 🤔
- *El 95% de los programadores es incapaz de escribir un código perfecto y sin errores al primer intento. El 5% restante miente.*
- El procedimiento de diagnosticar y solucionar los errores compone el proceso de depuración del código.



# Depuración

- La principal ventaja de un IDE es aprovechar las herramientas de depuración que nos ofrece.
- Visual Studio nos ofrece un depurador integrado, que nos brinda herramientas de desarrollo para inspeccionar de forma cercana el código escrito.



# Errores Sintácticos vs Errores de Lógica

- Un error sintáctico es aquel error que el compilador detecta y evita que el programa se ejecute correctamente.
- Suelen estar resaltados en Visual Studio, a través del subrayado.
- Subrayado rojo = error que no permite compilar.  
Subrayado verde = advertencia que debemos considerar.



# Errores Sintácticos vs Errores de Lógica

- Un error de lógica es aquel que no cohibe la compilación o ejecución del programa, pero que provoca comportamientos inesperados en el mismo.
- *No hay peor error que aquel que no da error.*



# Ejemplo

```
using System;

namespace EjemploDebug
{
    class Program
    {
        static void Main(string[] args)
        {
            List<string> lista = new List<string>() { "Hola", "Depuración" };
            Console.WriteLine("Hello World!");
            Console.ReadLine();
        }
    }
}
```

# Ejemplo

0 referencias

```
static void Main(string[] args)
{
    List<string> lista = new List<string>() { "Hola", "Depuración" };
    Console.WriteLine("Hello World!");
    Console.ReadLine();
}
```



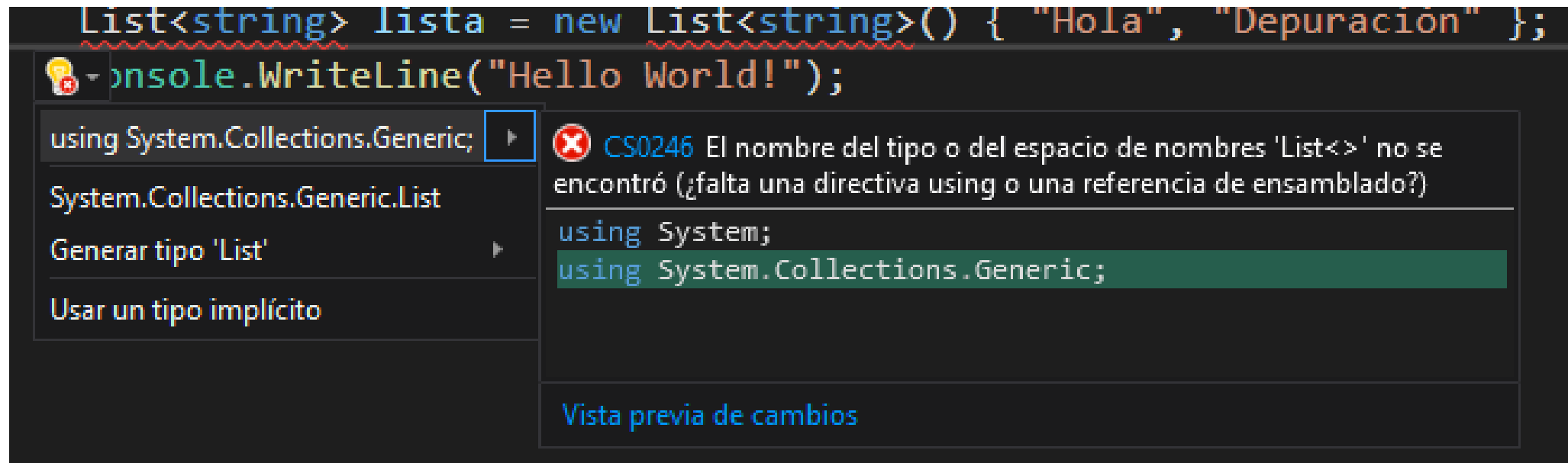
# Uso del bombillo

- Nuestro nuevo mejor amigo en Visual Studio.
- Ofrece soluciones básicas e inmediatas a los errores sintácticos que podemos encontrar en nuestro código.
- Debemos usarlo conscientemente, evaluando qué opciones nos ofrece y qué opciones estamos tomando. *No siempre nos brinda las soluciones a nuestros errores en específico.*



shutterstock.com · 569673730

# Uso del bombillo



# Modo de Depuración

- Cuando ejecutamos un programa en Visual Studio, lo estamos haciendo en el modo de depuración.
- Este modo nos permite analizar el código a través de las herramientas de depuración que nos ofrece.



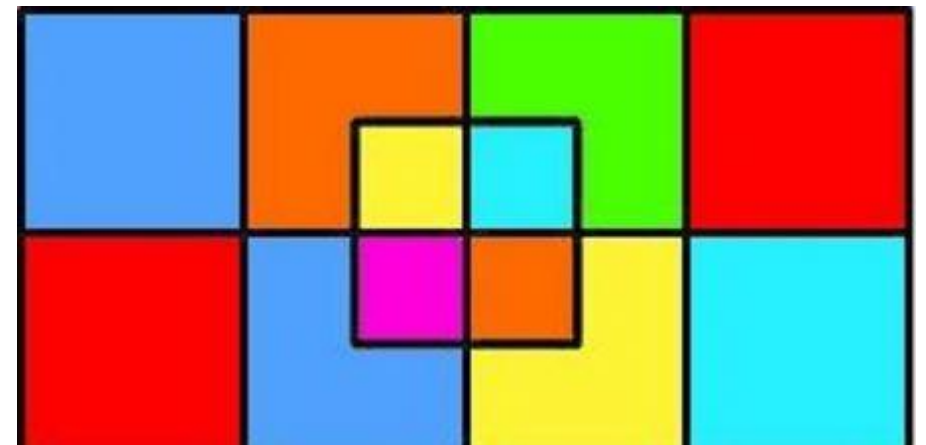
# Breakpoints

- Un breakpoint es un punto de interrupción; define una línea a partir de la cual el programa se va a interrumpir para mostrarnos el estado actual del mismo.
- Un breakpoint se define en una línea de código en particular al hacer click en la barra izquierda.



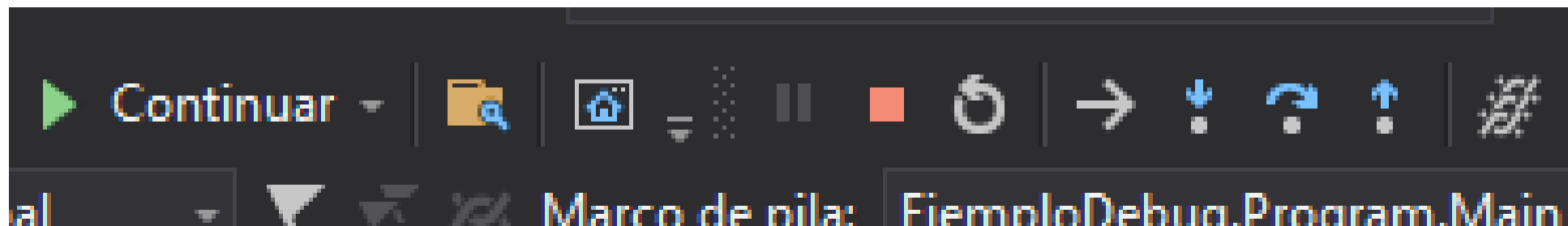
# Ejemplo

- Escriba el código de un programa de consola que lea el valor del lado de un conjunto de cuadrados, guarde en una lista el valor de sus áreas e imprima en pantalla todos estos valores junto al mayor y menor valor.



# Navegación por el código

- Cuando un breakpoint entra en ejecución, Visual Studio nos brinda una serie de botones con los que podemos navegar por el código del programa.



# Visor de Objetos

- Es una herramienta que nos brinda una vista pormenorizada de todos los objetos y variables actualmente instanciados, junto a su tipo y valor actual.
- Visual Studio nos ofrece tres visores de objetos distintos: Automático, Variables Locales e Inspección.



# Ejercicio

- Agrega gestión de excepciones a través de try – catch y realiza el proceso de depuración para que el ejercicio de Guardaparques funcione a la perfección.





# Referencias Bibliográficas

- Farrell, J. (2018). *Microsoft Visual C# 2017: An Introduction to Object-Oriented Programming*. Boston: Cengage Learning.
- Griffiths, I. (2019). *Programming C# 8.0*. Sebastopol: O'Reilly Media, Inc.
- Microsoft. (28 de 01 de 2021). *C# documentation*. Obtenido de Microsoft Docs: <https://docs.microsoft.com/en-us/dotnet/csharp/>