**TALLER – PILAS EN C#**

**BRAYAN SNEYDER GARCIA CAMACHO**

**1. Invertir una cadena**

**Descripción:**

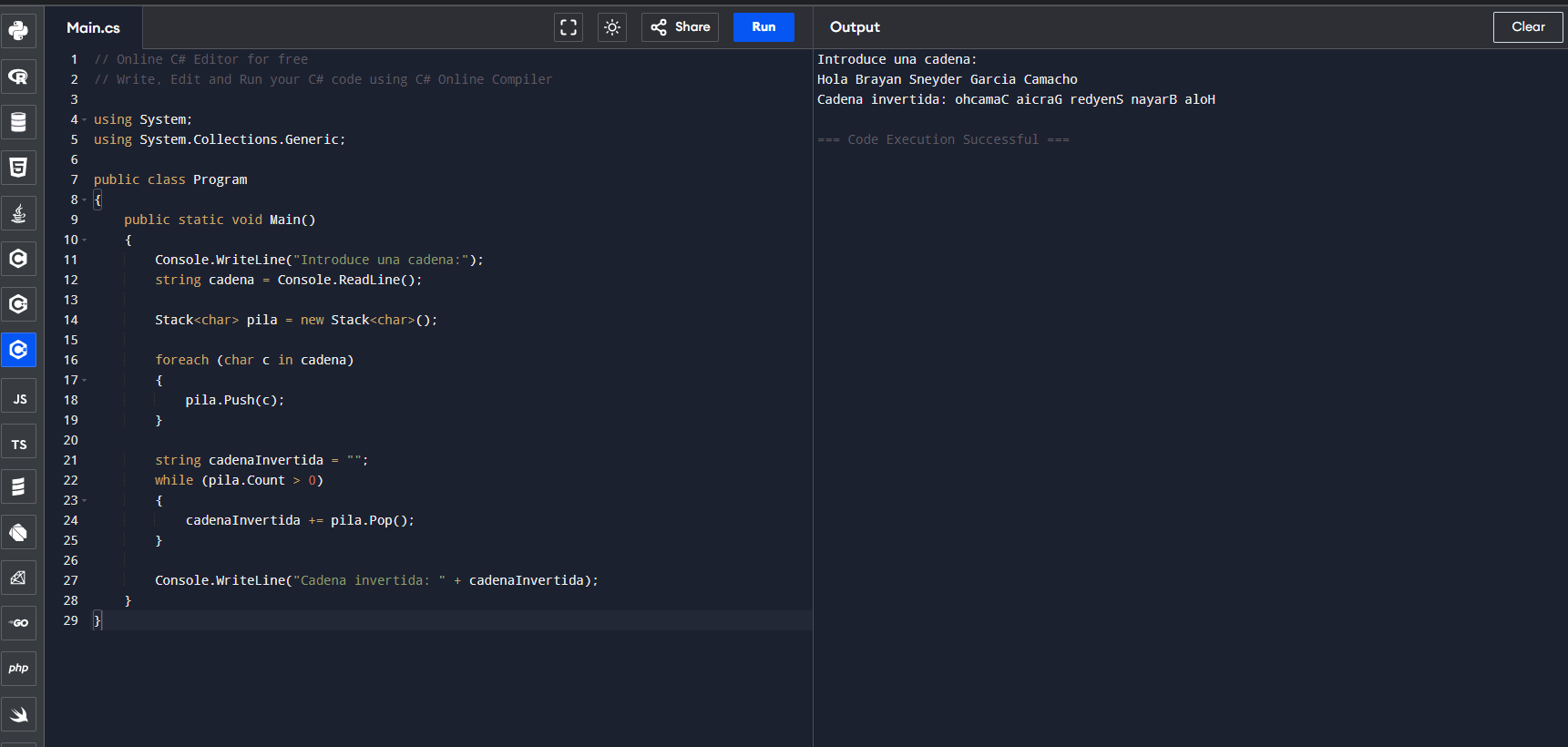
Pide al usuario una cadena y utiliza una pila para invertir el orden de los caracteres.

**Objetivo:**

Practicar el uso básico de Push() y Pop().

**Pista:**

Recorre la cadena, mete cada carácter en la pila y luego sácalos para formar la cadena invertida.



**2. Verificar paréntesis balanceados**

**Descripción:**

Dada una expresión con paréntesis ()[]{}, determina si están correctamente balanceados.

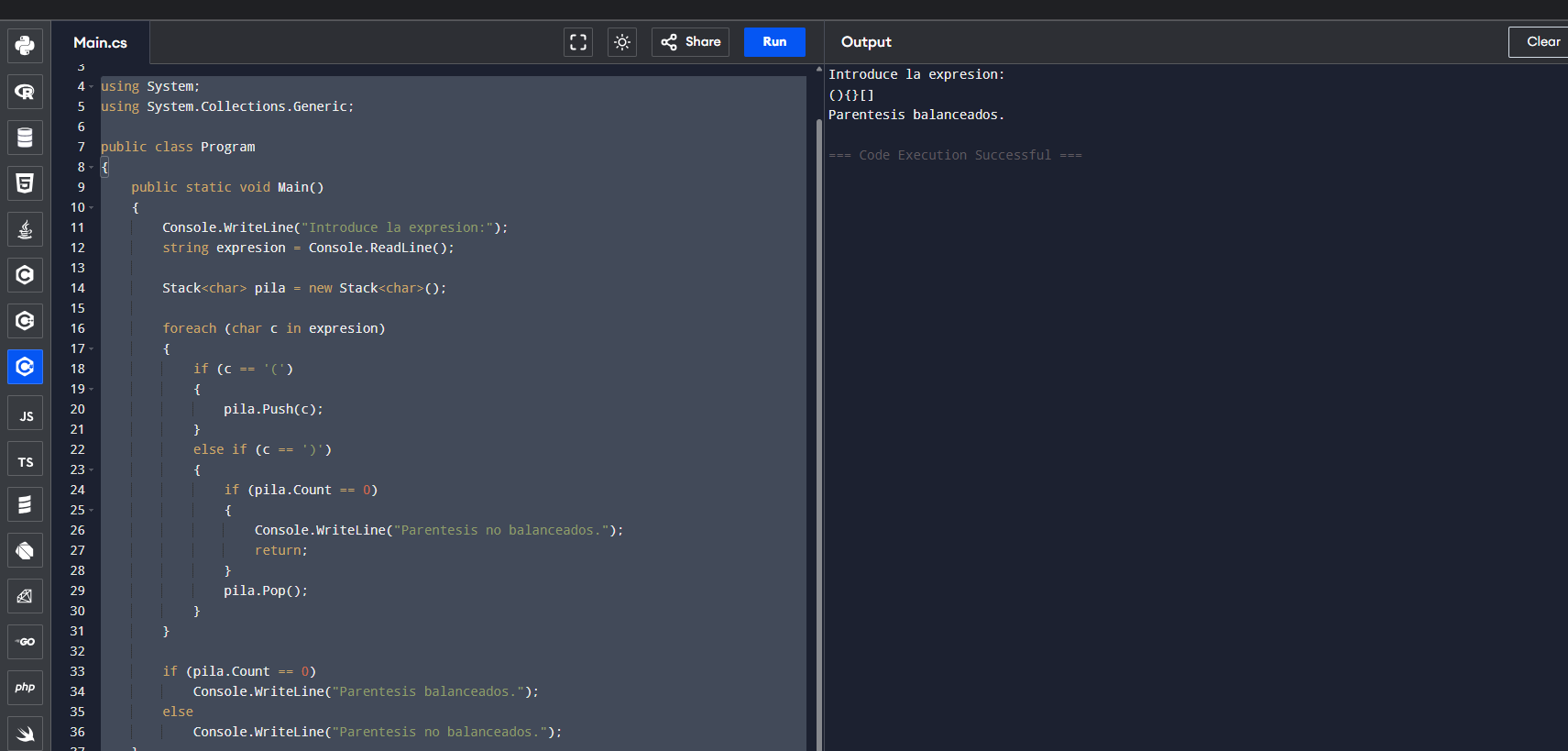
**Objetivo:**

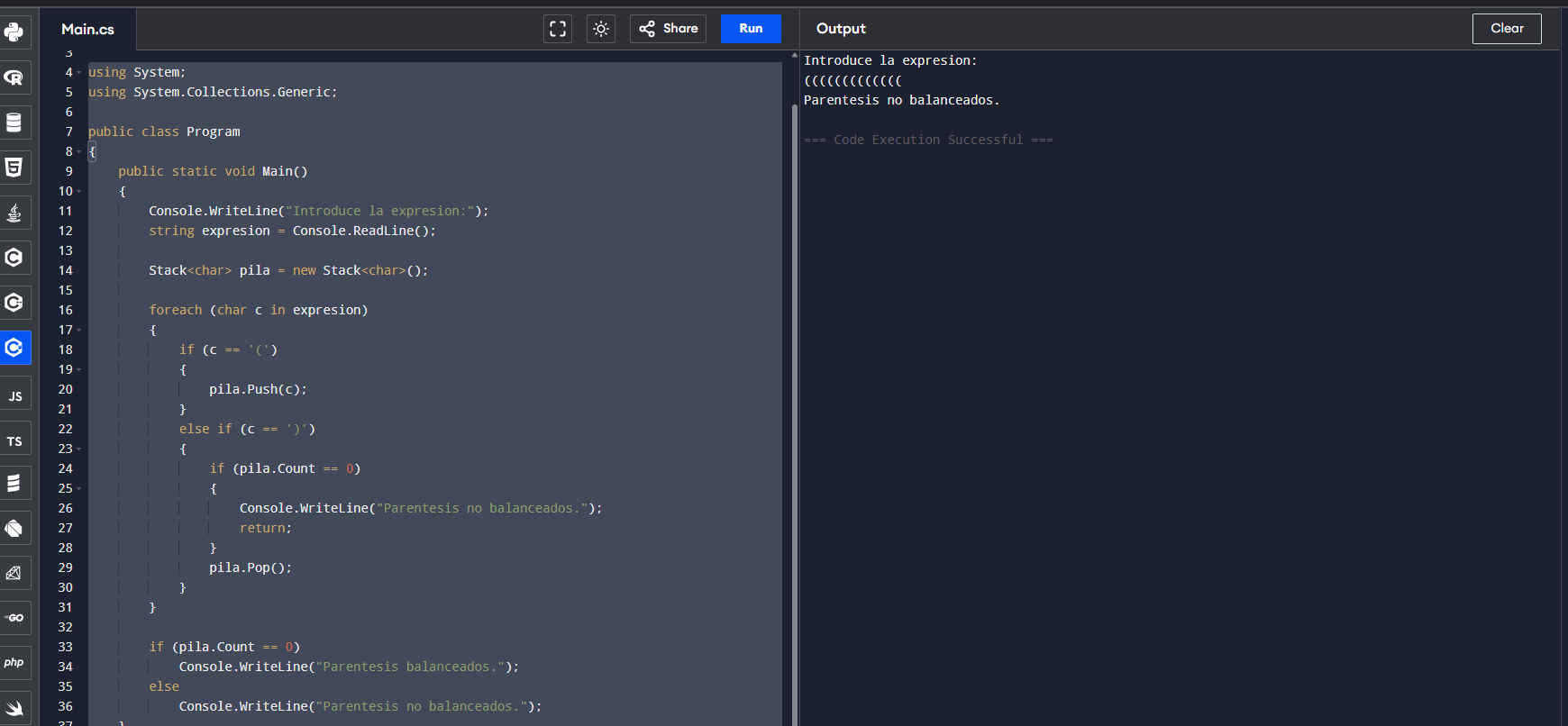
Aplicar la pila para analizar estructuras anidadas.

**Pista:**

Cada vez que se encuentre un símbolo de apertura, se apila; cuando se encuentre uno de cierre, se

desapila y se verifica el par correspondiente.





**3. Convertir número decimal a binario**

**Descripción:**

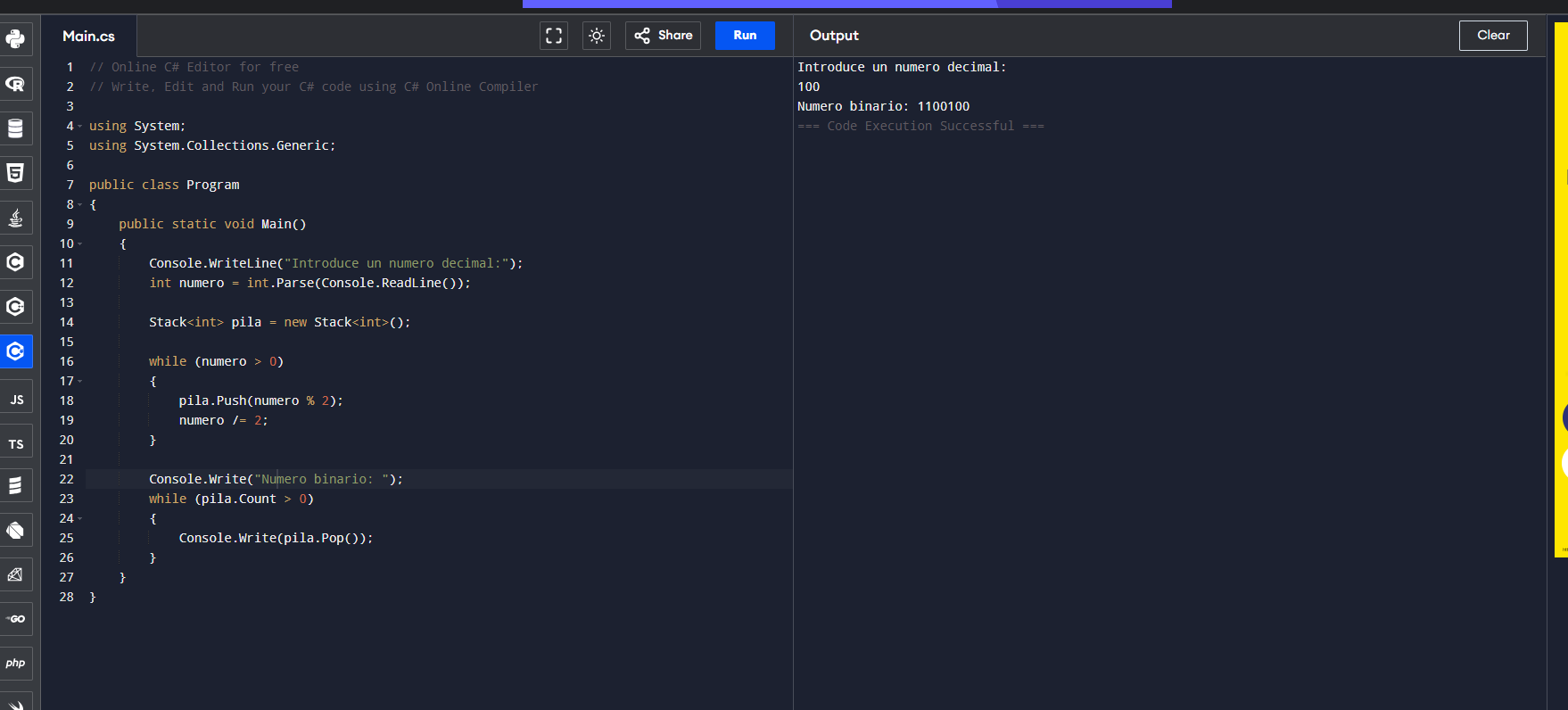
Convierte un número entero decimal a binario usando una pila.

**Objetivo:**

Usar pilas para transformar datos paso a paso.

**Pista:**

Divide el número entre 2, apila los restos, y luego sácalos para formar el número binario.



**4. Evaluar expresión postfija (notación polaca inversa)**

**Descripción:**

Evalúa expresiones como "5 3 + 8 2 - \*" usando una pila.

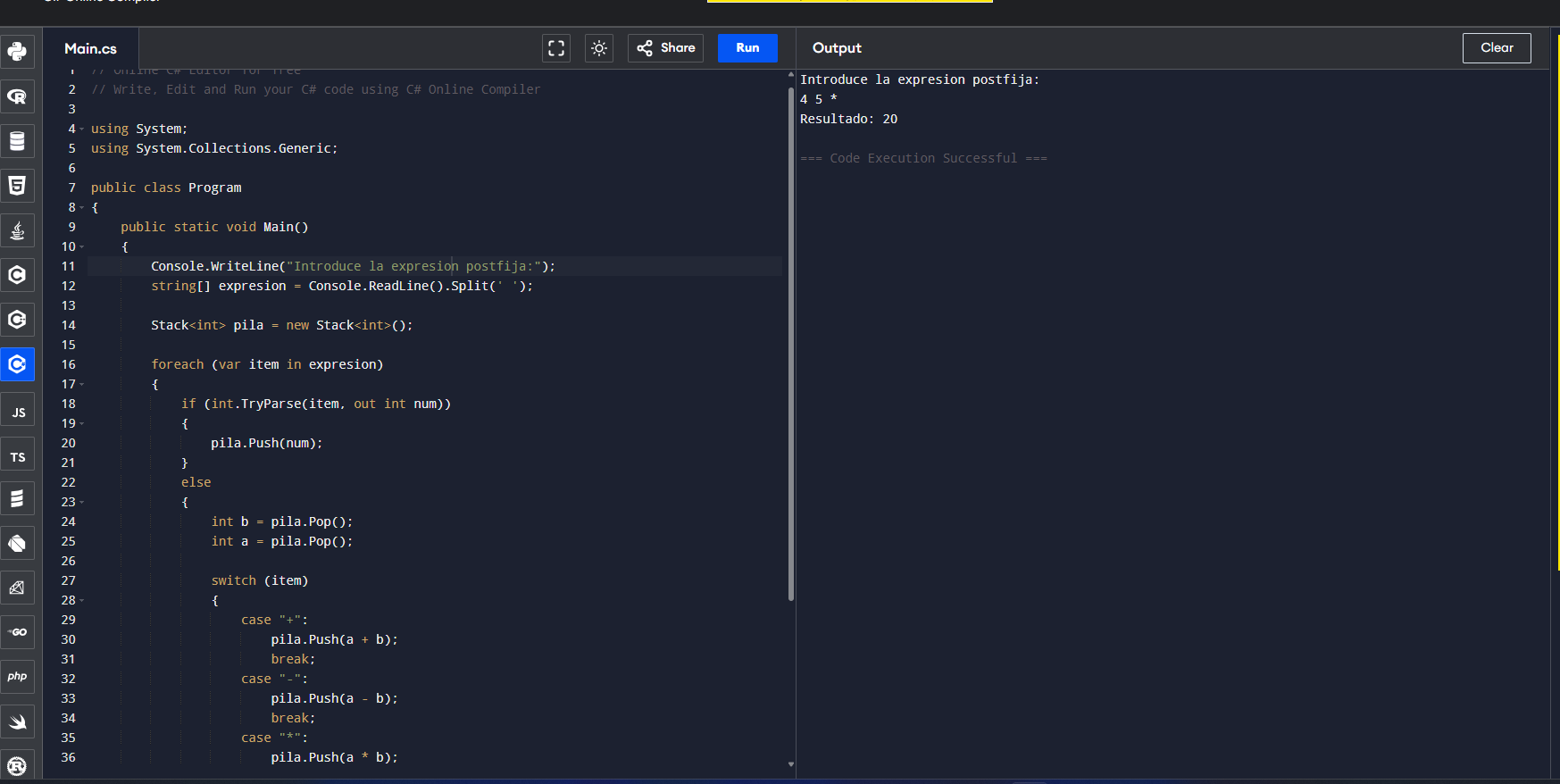
**Objetivo:**

Practicar manipulación avanzada de datos en la pila.

**Pista:**

Cuando se lea un número, apilarlo; cuando se lea un operador, desapilar dos operandos, aplicar la

operación y apilar el resultado.



**5. Deshacer (Undo) en un editor de texto simulado**

**Descripción:**

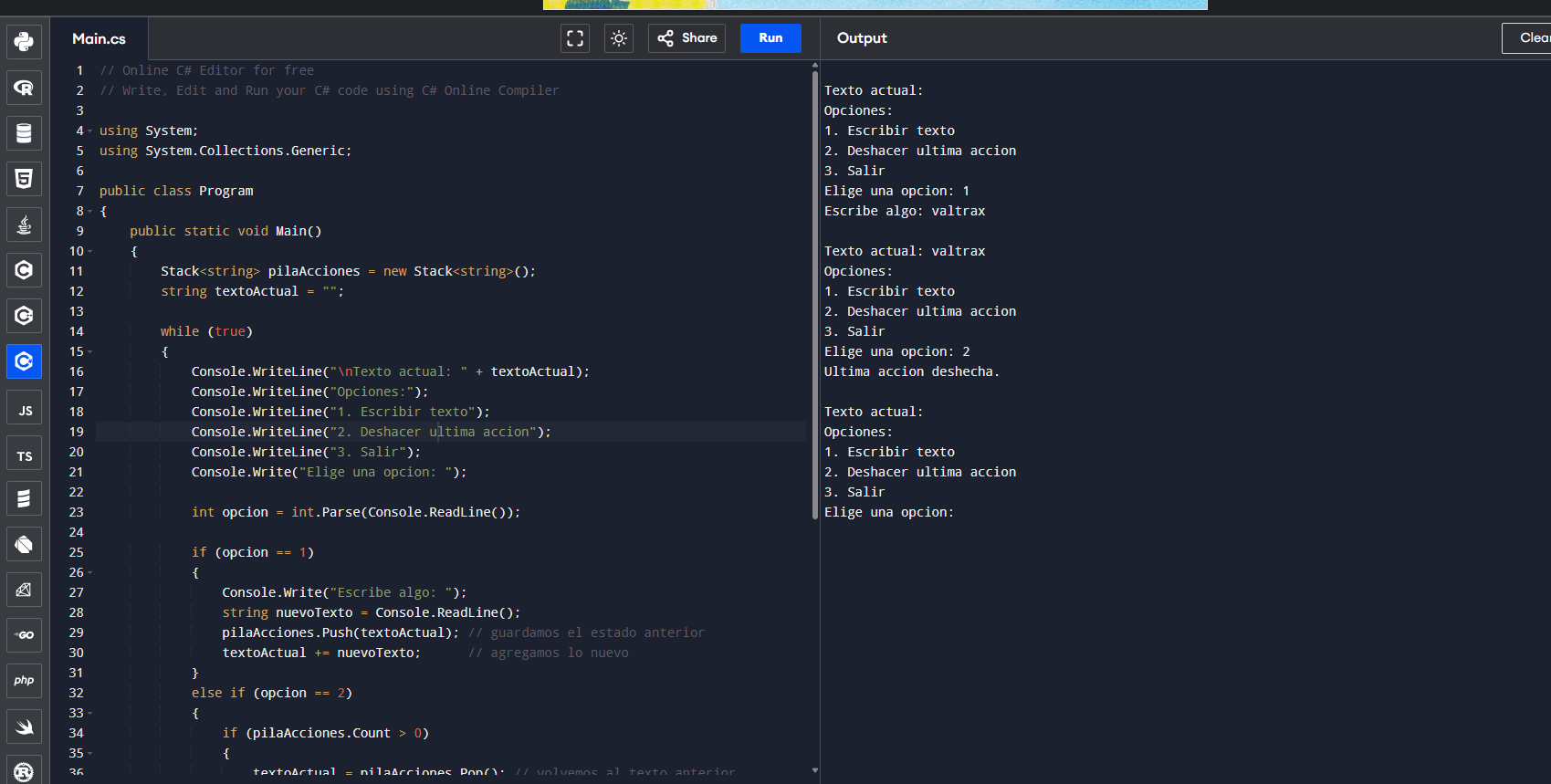
Simula un editor con las operaciones: escribir, deshacer.

**Objetivo:**

Aplicar pilas en el manejo de estados.

**Pista:**

Cada acción de escritura se apila. Al hacer “deshacer”, se desapila la última acción.



**6. Comprobar si una palabra es palíndroma**

**Descripción:**

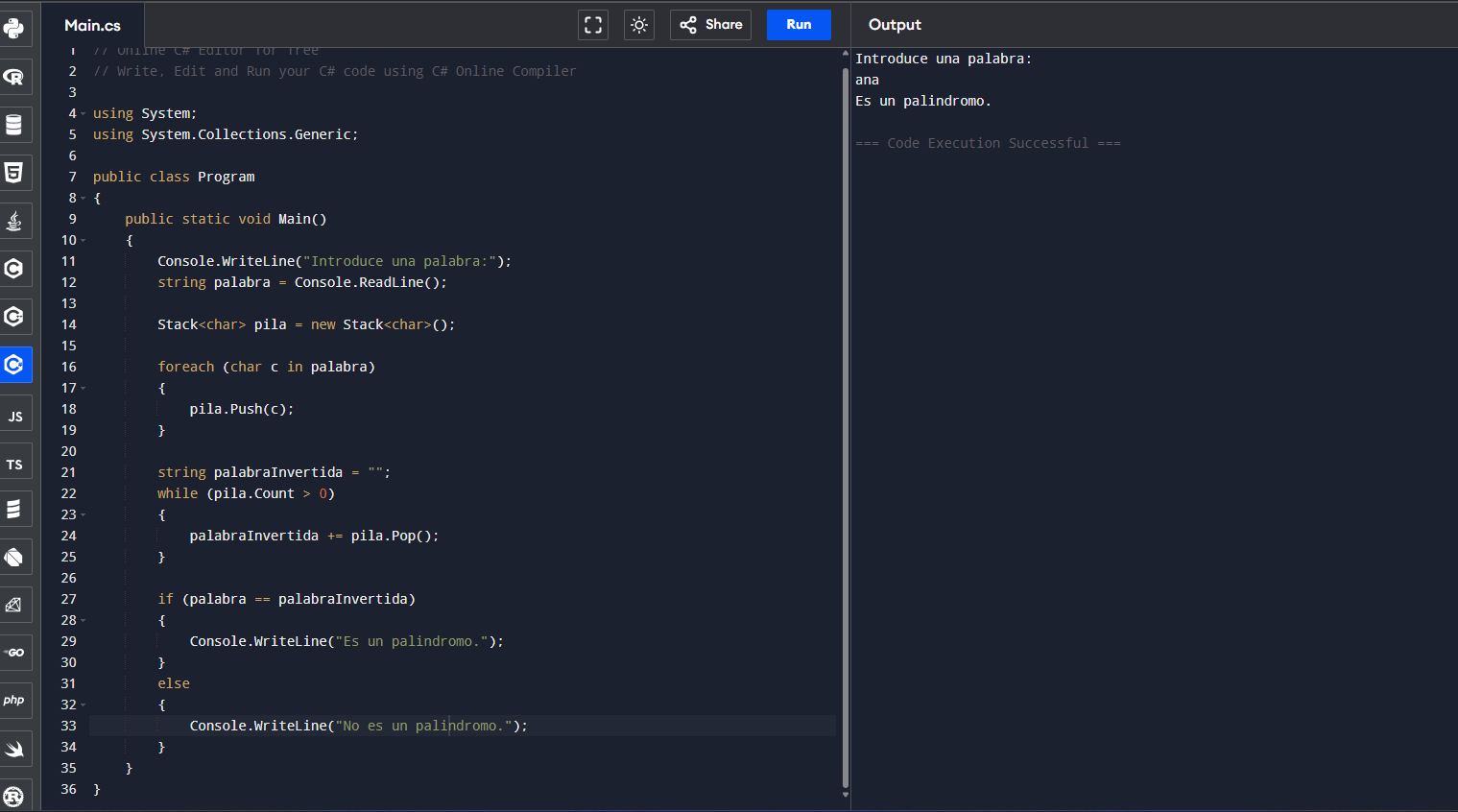
Verifica si una palabra se lee igual al derecho y al revés usando una pila.

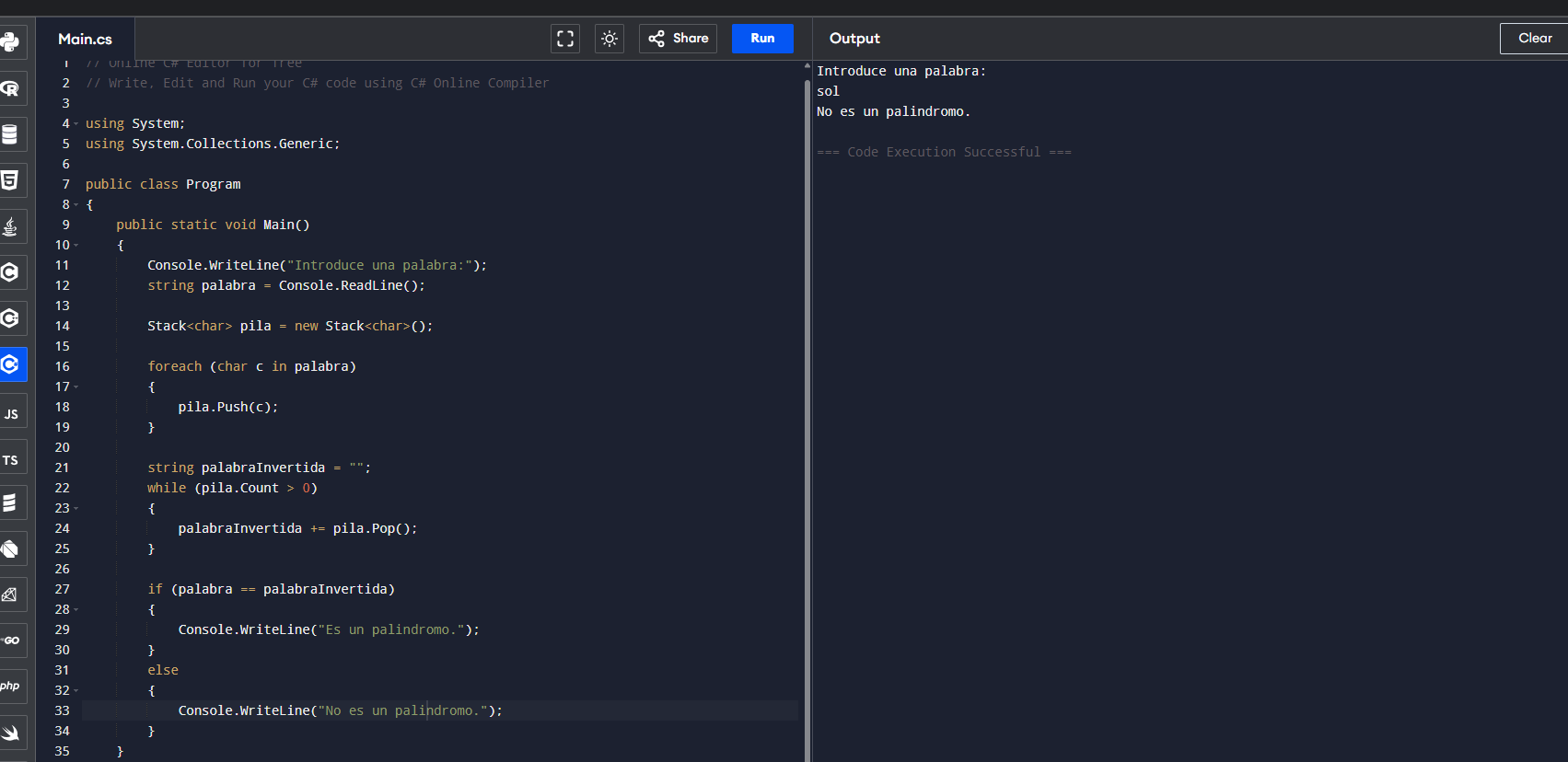
**Objetivo:**

Usar pilas para comparar secuencias.

**Pista:**

Apila los caracteres, luego desapílalos y compara con la cadena original.





**7. Revertir el orden de palabras en una oración**

**Descripción:**

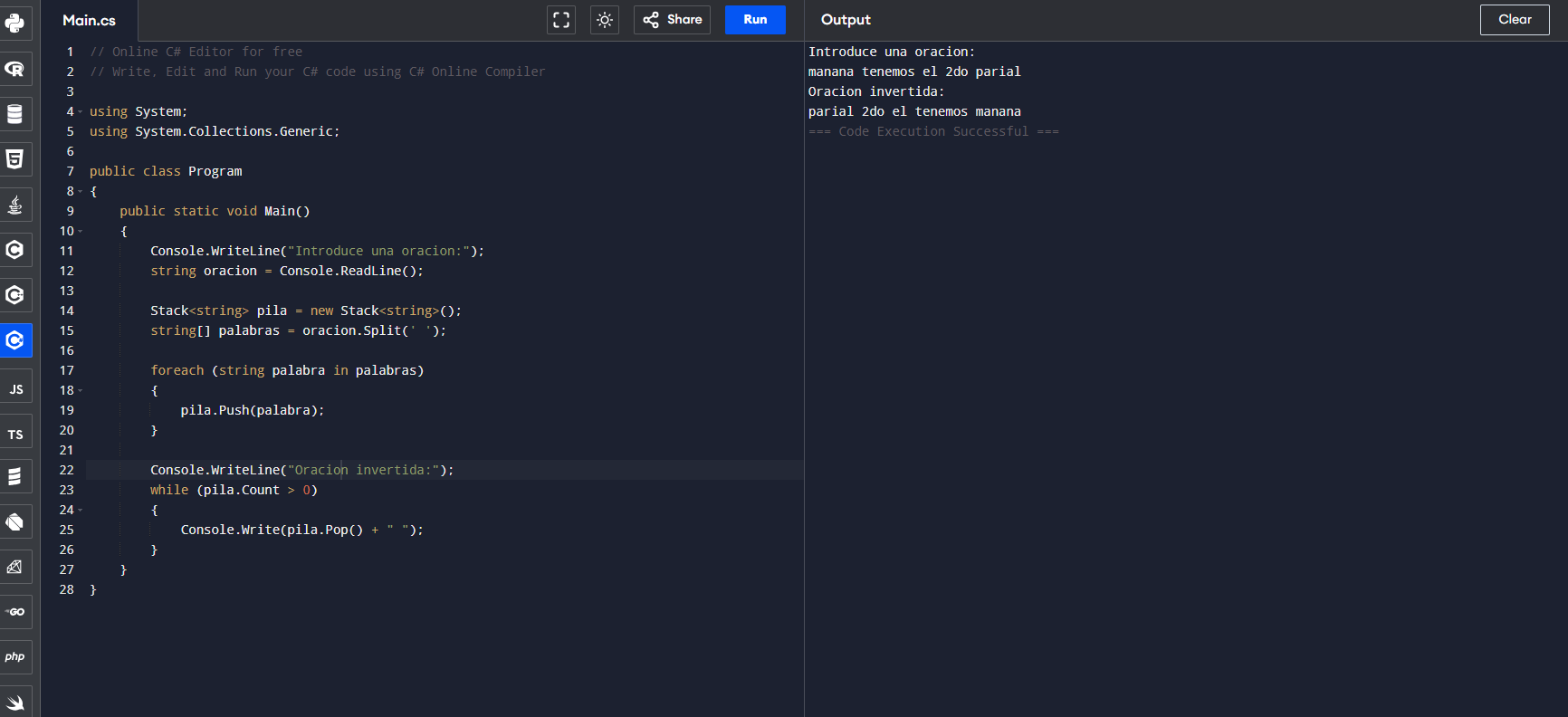
Invierte el orden de las palabras en una oración sin invertir las letras de cada palabra.

**Ejemplo:**

Entrada: "C# es genial" → Salida: "genial es C#"

**Pista:**

Divide la oración en palabras, apílalas y sácalas en orden inverso.



**8. Evaluar expresión infija simple**

**Descripción:**

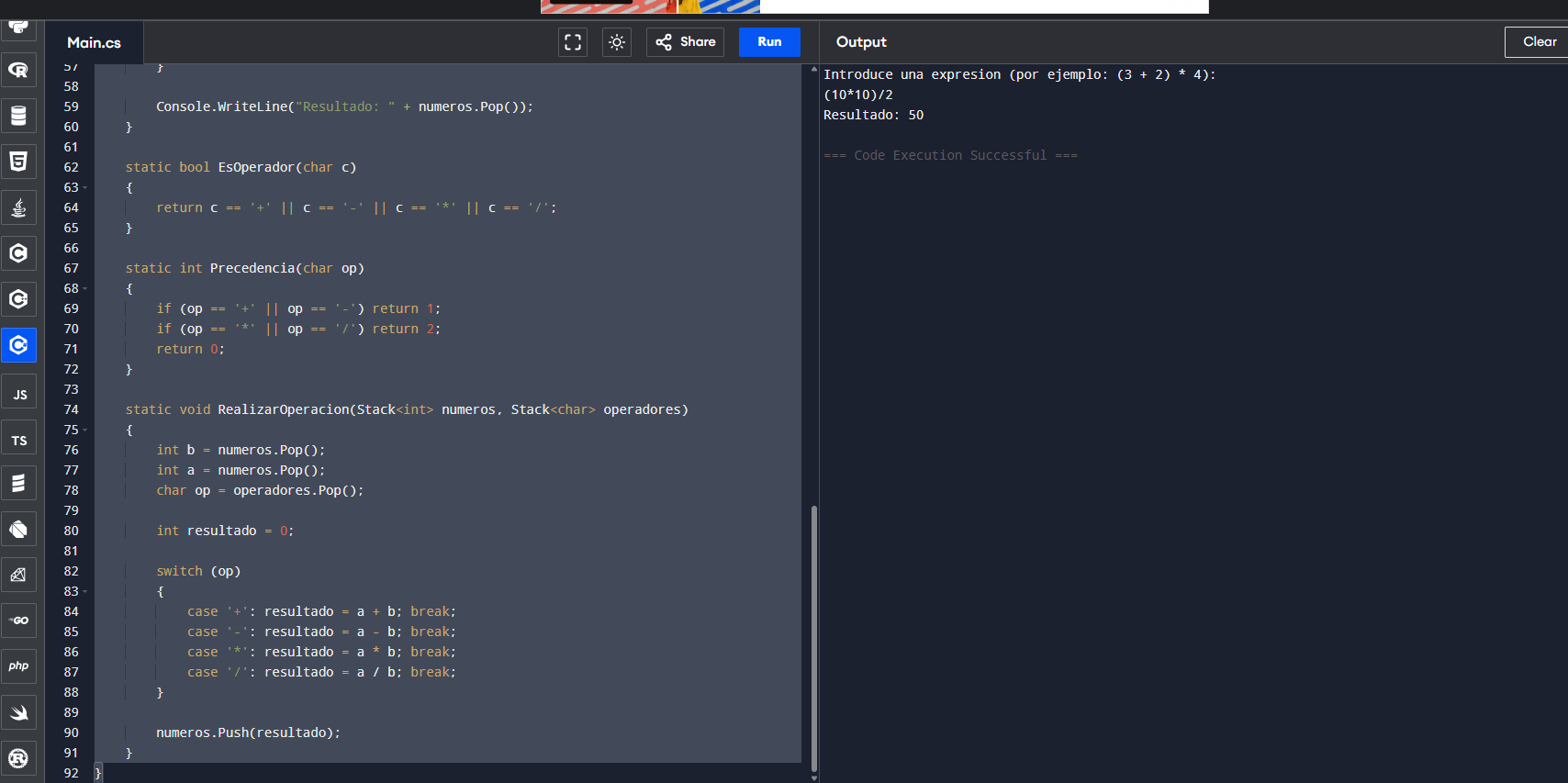
Evalúa una expresión aritmética infija con paréntesis, como "3 + (2 \* 4)".

**Objetivo:**

Usar dos pilas: una para operandos y otra para operadores.

**Pista:**

Implementa el algoritmo del “Shunting Yard” (Dijkstra).



**9. Historial de navegación**

**Descripción:**

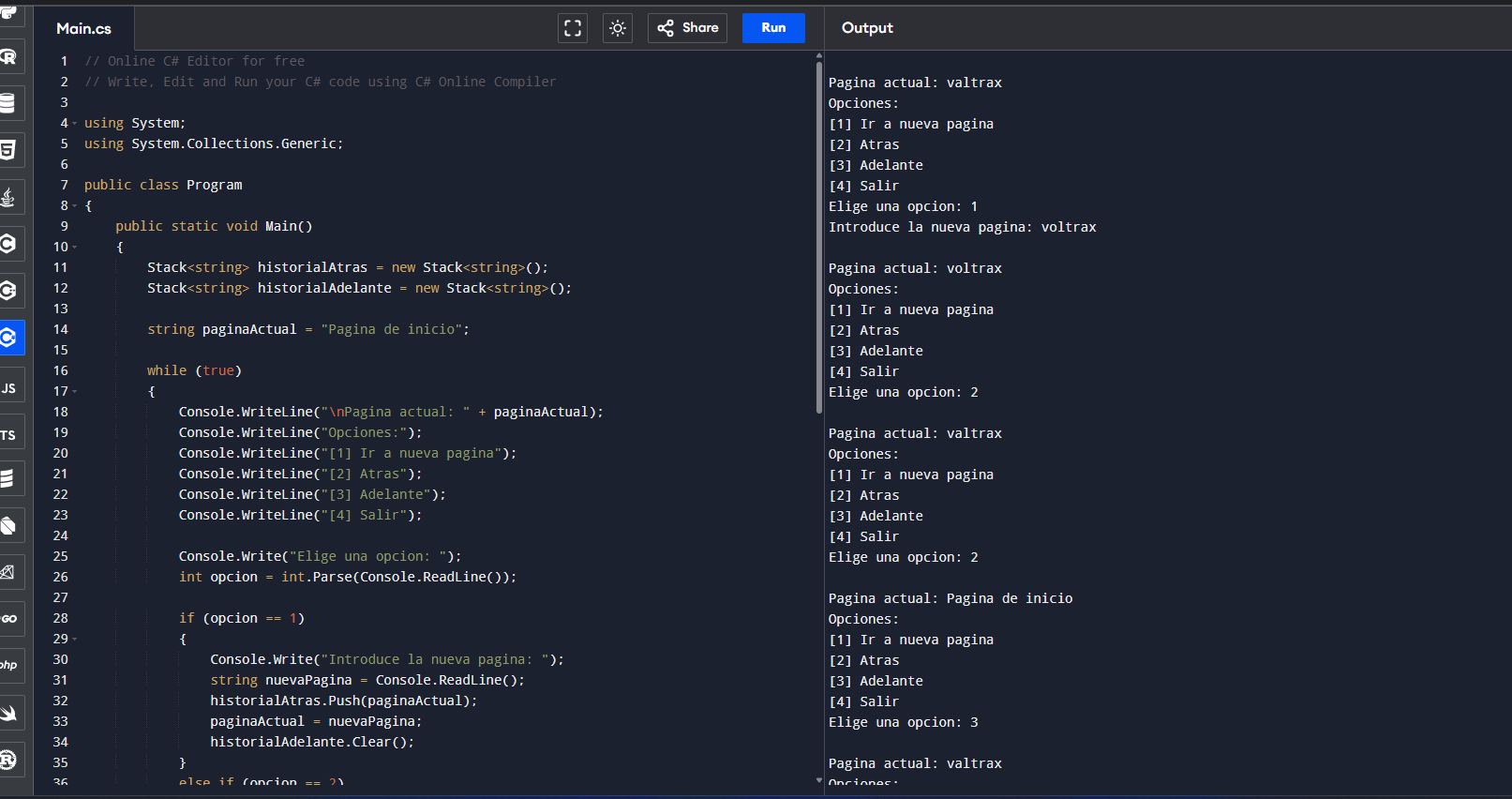
Simula un navegador con “atrás” y “adelante”.

**Objetivo:**

Usar dos pilas: una para el historial anterior y otra para las páginas a las que se puede volver.

**Pista:**

Al visitar una nueva página, apílala en “atrás” y limpia la pila “adelante”.



**10. Convertir expresión infija a postfija**

**Descripción:**

Convierte expresiones infijas como "A + B \* C" a postfijas "A B C \* +".

**Objetivo:**

Practicar el control de prioridad de operadores con pilas.

**Pista:**

Usa una pila para los operadores y una lista de salida para los operandos.

