

Realización de trazas.

Las trazas nos van a permitir conocer si existen errores en un programa, lo que hace un programa, que valores tomará las variables en cada interacción, evitar bucles infinitos, etc.

A continuación se expone un ejemplo de traza.

¿Qué realiza el siguiente trozo de código?.

Para poder comprobar que realiza el siguiente trozo de código, basta con realizar un traza donde se contemple los valores de las variables y las salidas.

	n	c	j	es	SALIDA
1 { int n=3,c=0;	3	0	0	false	
2 bool es = false;					
3 while(n<=20){					
4 c=n/2;					
5 es=true;					
6 j=2;					
7 while(j<=c)					
8 if(n % j == 0)					
9 es=false;					
10 j=j+1;					
11 }					
12 if(es)					
13 Console.WriteLine("El número {0} es primo.", n);					
14 n=n+1;					
}					

	n	c	j	es	SALIDA
1 { int n=3,c=0,j	3	0	0	false	
2 bool es = false;					
3 while(n<=20){					
4 c=n/2; <—		1	2	true	3
5 es=true; <—					
6 j=2; <—	4				
7 while(j<=c)					
8 if(n % j == 0)					
9 es=false;					
10 j=j+1;					
11 }					
12 if(es)					
13 Console.WriteLine("El número {0} es primo.", n);					
14 n=n+1; <—					
}					

```

static void Main(string[] ar
{
    1      int n=3,c=0,j=
    2      bool es = fals

    3      while(n<= 5){
    5          c=n/2;
    6          es=true;
    7          j=2;
    8          while(j<=c){
    9              if(n %
    10                 es=fal
    11                 j=j+1;
    12             }
    13             if(es)
    14                 Console.Wr

    15         n=n+1;
    16     }
}

```

n	c	j	es	SALIDA
3	0	0	false	
	1	2	true	
				3
4	2	2	true	

```

static void Main(string[] a
{
    1      int n=3,c=0,j=
    2      bool es = fal

    3      while(n<= 5){
    5          c=n/2;
    6          es=true;
    7          j=2;
    8          while(j<=c)
    9              if(n %
    10                 es=fa
    11                 j=j+1;
    12             }
    13             if(es)
    14                 Console.Wr

    15         n=n+1;
    16     }
}

```

n	c	j	es	SALIDA
3	0	0	false	
	1	2	true	
				3
4	2	2	true	
		3	false	
5				

```

static void Main(strin
{
    1      int n=3,
    2      bool es

    3      while(n<
    5          c=n/2
    6          es=true
    7          j=2;
    8          while(
    9              i
    10             j=
    11             }
    12             if(es)
    13                 Cons

    14         n=n+1;
    15     }
}

```

n	c	j	es	SALIDA
3	0	0	false	
	1	2	true	
				3
4	2	2	true	
		3	false	
5	2	2	true	

```

static void Main(string[] args)
{
    1      int n=3,c=0,j=0;
    2      bool es = false;

    3      while(n<=25){
    4          c=n/2;
    5          es=true;
    6          j=2;
    7          while(j<=c){
    8              if(n % j == 0
    9                  es=false;
   10              j=j+1;
   11          }

   12          if(es)
   13              Console.Write(n)
   14          n=n+1;
    }
}

```

n	c	j	es	SALIDA
3	0	0	false	
	1	2	true	
				3
4	2	2	true	
		3	false	
5	2	2	true	
		3		5
6				

TRAZA FINAL

Viendo la traza anterior, podemos interpretar que el código imprime por pantalla los números primos dentro del intervalo [3..5]

Ejercicio 1:

¿Qué mostraría el siguiente programa en pantalla?. **Muestra la traza para cuando i sea igual a 10.**

```
namespace Pruebas_software
{
    class Program
    {
        static void Main(string[] args)
        {
            int j=0;
            int i=10;

            while(i!=0){
                j=i;
                while(j!=0){
                    Console.Write(j)
                    j=j-1;
                }
                i=i-1;
            }
        }
    }
}
```

j	i	SALIDA
0	10	
10		10
9		9
8		8
7		7
6		6
5		5
4		4
3		3
2		2
1		1
0	9	
9		

TRAZA FINAL

Ejercicio 2.

Dado el siguiente fragmento de programa:

```
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int x = 1;

            if((a>3) && (b>7)){
                x=x-1;
                if ((c>4) && (d>8) && (e<5)) x=x-1;
                else if((f>3) && (g>2))
                    x=x*2;
                else x=x+4;
            }
            else x=x*4;
        }
    }
}
```

Responde a las siguientes preguntas:

1. ¿Qué valor tendrá x si a=3, b=6, c=5, d=10, e=4, f=1, g=5?
El valor de x seria 4.
2. ¿Qué valor tendrá x si a=7, b=7, c=5, d=15, e=1, f= 15, g=1?
El valor de x seria 4.
3. ¿Qué valor tendrá x si a=8, b=9, c=1, d=15, e=18, f= 6, g=4?
El valor de x seria 0.
4. ¿Qué valor tendrá x si a=8, b=8, c=1, d=15, e=18, f= 6, g=4?

El valor de x seria 0.

Ejercicio 3.

Dado el siguiente código:

```
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int i, j, k = 1;

            for(i=0;i<4;i++)
                for(j=0;j<i;j++)
                {
                    if(j==2)
                        break;
                    k+=i+j;
                }
        }
    }
}
```

¿Cuál será el valor de "k"? **Muestra la traza final.**

i	j	k	SALIDA
0	0	1	
1	0	2	
2	0	4	
	1	7	
	2		
3	0	10	
	1	14	
	2		14

El valor de k es 14

Ejercicio 4

```
namespace ConsoleApplication1
{
    class Program
    {
        int funcion(int a, int b)
        {
            Int
            resultado;

            if(a<b)
                resultado=a;
            else
                resultado=funcion(a-b,b);

            return resultado;
        }

        static void Main(string[] args)
        {
            función(5,2);
        }
    }
}
```

¿Cuál será el resultado de invocar a la función "**funcion**"?. **Muestra la traza final.**

Al llamar la función, la a hace una resta de a-b, y devuelve resultado que seria 3 como a y 3 como b, así sucesivamente hasta que el resultado es un único numero

a	b	resultado	SALIDA
5	2	3, 2	
3	2		
1	2	1	1

Ejercicio 5

Encuentra los errores (pueden ser errores de compilación o bucles infinitos) en el siguiente trozo de código, indicando una descripción del error y la solución a dicho error.

NOTA: Utiliza trazas para poder encontrar los errores.

```
static void Main(string[] args)
```

```
{
int i, j; → Las
variables no están
declaradas con lo
cuál ese seria uno de
los errores.
declararemos las
variables.
```

```
i = 8;
```

```
j = 2;
```

```
    do{
        do
        {
            Console.WriteLine(j + "*" + i "=" + j*i);
            i=i+1;    j=j+1;    } while(i!=10)
        }
    while(j!=5)
}
```

i	j	SALIDA
8	2	2 * 8 = 16
9	3	3 * 9 = 27
10	4	
	4	

EN ESE PUNTO HABRIA OTRO ERROR YA QUE HAY UN BUCLE POR HACER PERO SERIA INFINITO YA QUE LA J NO SE INCREMENTA PARA QUE FINALICE EL BUCLE. SOLUCION: PONER J=J+1 FUERA DEL BUCLE DE CONSOLE.WRITE TAL COMO VOY A ESCRIBIR A CONTINUACION

```
i = 8; j = 2;
```

```
    do{
        do
        {
            Console.WriteLine(j + "*" + i "=" + j*i);
            i=i+1;
        }
        while(i!=10)
        j=j+1;
    }
    while(j!=5)
```


Ejercicio 6

Dado el siguiente código:

```
static void Main(string[] args)
{
    int i=0;

    while(i<4)
    {
        switch(i) {
            case 0: case 1: case 2:
                i++; break;
            case 3: i--;
                Console.write(i);
            case 4: i=i+2;
                Console.write(i);
            default:
                Console.write(i);
        }
    }
}
```

¿Cuál sería el resultado de ejecutar el programa anterior? En caso de que se produjera un error de compilación o un bucle sin fin, indica el motivo.

El motivo de el error es que seria un bucle infinito por el hecho de que cuando i es 3, se resta (i--) y vuelve a ser dos de nuevo, por lo tanto nunca termina. Aunque el bucle terminaría una vez se incremente la i de 3 a 4.

NOTA: Utiliza trazas para poder encontrar los errores.

i	SALIDA
0	
1	
2	
3	
2	2
3	
2	2

Ejercicio 7:

Dado el siguiente código:

```
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int j = 2;
            int s = 0;
            int n;

            n = Int32.Parse(Console.ReadLine());
            while(j<=n/2)
            {
                if (n / j == 0)
                {
                    s = s +1;
                    j = j +1;
                }
            }
            if (s == 0)
                Console.Write(n + "es primo");
            else
                Console.Write(n + "no es primo");
        }
    }
}
```

Imaginemos que n es 9, sabemos que no es primo

j	s	n	SALIDA
2	0	9	
			Primo(erroneo)

1. ¿Qué errores presenta?

En este caso presentaría que 9 es primo porque no haría el bucle. Hay un error el cual si no es mayor $n/2$ que j, no se haría el bucle.

Además la j se incrementa dentro del if, la cual debería incrementarse fuera.

Otro error es que en la condición del primer if, la división de n / j debe dar 0 en el resultado, cuando debería ser el resto. Para ello sustituimos el signo de división con el %.

2. Modificalo para que funcione correctamente.

Hola

```
int j = 2;
int s = 0;
int n;
n = int.Parse(txtNum.Text);

while (j <= n / 2)
{
    if (n % j == 0)
    {
        s++;
    }
    j++;
}

if (s == 0)
    Console.Write(n + "es primo");
else
    Console.Write(n + "no es primo");
```

3. Muestra una traza, para comprobar su funcionamiento.

j	s	n	SALIDA
2	0	9	
3	1		
4	1		
			9 No es primo

Probaremos con un numero mas primo como el 11

j	s	n	SALIDA
2	0	11	
3	0		
4	0		
5	0		
			11 es primo