

**ILERNA**

Online

# **Videotutoría 6: Preguntas de la UF1**

Módulo 03A: Programación



# UF1: Preguntas

Referencias

```
static void Main(string[] args)
{
    int i = 0;
    int z = 3;

    while (i < 5) {
        i++;
    }

    Console.WriteLine("{0}", z);
    Console.ReadKey();
}
```

¿Qué salida producirá el siguiente código?

- a) 1
- ☒ b) 3
- c) 5
- d) El código contiene errores

# UF1: Preguntas

```
int [,] array = new int [4,2];
```

|     |    |
|-----|----|
| 1   | 2  |
| 10  | 21 |
| 100 | 9  |
| 7   | 8  |

¿De qué forma puedo acceder al número 100 en esta matriz?

- a) array [0,2]
- b) array [1,2]
- c) array [2,1]
- ☒ d) array [2,0]

# UF1: Preguntas

```
int [,] array = new int [4,2];
```

|     |    |
|-----|----|
| 1   | 2  |
| 10  | 21 |
| 100 | 9  |
| 7   | 8  |

¿De qué forma puedo acceder al número 100 en esta matriz?

- a) array [0,2]
- b) array [1,2]
- c) array [2,1]
- d) array [2,0]

# UF1: Preguntas

El operador que utilizamos para invertir el valor de un boolean es

- a. \~
- ☒ b. !
- c. !=
- d. ^

# UF1: Preguntas

¿Qué tipo de estructura no lleva a cabo ningún tipo de comprobación lógica?

- a) Las estructuras de selección.
- ☒ b) Las secuenciales.
- c) Las estructuras de iteración.
- d) Las secuencias repetitivas.

# UF1: Preguntas

¿Cuántas iteraciones realiza el siguiente bucle?

```
For (i=0;i<7;i++) {  
  Console.Write ("Imprimiendo desde dentro del bucle");  
}
```

a) 8

b) 7

c) 6

☒ d) Ninguna. La inicialización de la variable contadora es incorrecta

No hay inicialización de i. Si la tuviésemos, `int i = 0`, tendríamos 7 iteraciones.

# UF1: Preguntas

Cuando los pasos que permiten resolver un problema están escritos en algún lenguaje de programación, estamos hablando de...

- a) Programas
- b) Algoritmos
- c) Algoritmos y programas.
- d) Lenguajes de programación.



# UF1: Preguntas

Indica cuál es equivalente al operador condicional de C#:

Con switch le ponemos un valor, y seleccionamos donde queremos entrar. Es de selección, no de condición.

- a) Switch
- b) If, if-else, switch
- ☒ c) If o if-else
- d) Todas las respuestas son correctas

# UF1: Preguntas

¿Qué es la instrucción GOTO? Pon un ejemplo

¿Por qué no se debe utilizar? Razona tu respuesta

# UF1: Preguntas

¿Qué es la instrucción GOTO? Pon un ejemplo

¿Por qué no se debe utilizar? Razona tu respuesta

0 referencias

```
static void Main(string[] args)
{
    int i, j;

    for (i = 0; i <= 5; i++)
        for (j = 0; j <= 5; j++)
        {
            if ((i == 1) && (j == 0))
                goto salida;
            Console.WriteLine("i vale {0} y j vale {1}.", i, j);
        }

salida:
    Console.Write("Fin del programa");
    Console.ReadKey();
}
```

# UF1: Preguntas

La programación estructurada, propuesta por ***Edsger Dijkstra***, demuestra que todo programa puede escribirse utilizando únicamente las tres instrucciones de control. ¿Cuáles son?

# UF1: Preguntas

La programación estructurada, propuesta por **Edsger Dijkstra**, demuestra que todo programa puede escribirse utilizando únicamente las tres instrucciones de control. ¿Cuáles son? Pon un ejemplo de cada una de ellas

- Secuencia.
- Instrucción condicional.
- Iteración (bucle de instrucciones)

# UF1: Preguntas

Realiza un pequeño programa donde el usuario introduzca el número de columnas que desee para formar el siguiente triángulo:

```
1
12
123
1234
```

# UF1: Preguntas

Realiza un pequeño programa donde el usuario introduzca el número de columnas que desee para formar el siguiente triángulo:

1  
12  
123  
1234

```
0 referencias
static void Main(string[] args)
{
    int i, j, rows;

    Console.WriteLine("Introduce el número de columnas que desea : ");
    rows = Convert.ToInt32(Console.ReadLine());
    for (i = 1; i <= rows; i++)
    {
        for (j = 1; j <= i; j++)
            Console.Write("{0}", j);
        Console.WriteLine("\n");
    }
    Console.ReadKey();
}
```

Solo se incrementa cuando J no es igual o es inferior a i. Cada vez que se cumple la condición salimos del bucle, y cuando volvemos a entrar, volvemos a inicializarlo, en este caso a tener el valor de 1 en j.

FOR: necesitamos un índice que se declara dentro de for : for (int i = 0; i < 10; i++) //1ª condición establecer la variable, 2ª establecer condición de salida. La tercera opción (j++) es lo que hacemos cada vez que aún no se cumple la condición.

# UF1: Preguntas

¿Qué tipo de error, es un error lógico? Pon un ejemplo

No impide el funcionamiento pero no es lo buscado.

Da una salida errónea. Esperamos un 10 pero sale un 8.



# UF1: Preguntas

Dime alguna característica de un IDE

# UF1: Preguntas

## Dime alguna característica de un IDE

- Son multiplataforma.
- Actúan como soporte para diferentes lenguajes de programación.
- Reconocen sintaxis.
- Están integrados con sistemas de control de diferentes versiones.
- Tienen un depurador.
- Permiten importar y exportar proyectos.
- Manejan diferentes idiomas.
- Facilitan un manual de ayuda para el usuario.

# UF1: Preguntas

¿Qué saca por pantalla el siguiente código?

0 referencias

```
static void Main(string[] args)
{
    int i, j;

    for (i = 1; i <5; i++)
    {
        for (j = 1; j <5; j++)

            if (i==5)
                Console.Write("{0}", i, j);
    }
    Console.ReadKey();
}
```

Nunca va a escribir nada ya que nunca valdrá 5.

# UF1: Preguntas

¿Qué sentencia falta para que salga la salida indicada?

```
class BreakTest
{
    static void Main()
    {
        for (int i = 1; i <= 100; i++)
        {
            if (i == 5)
            {
                ;
            }
            Console.WriteLine(i);
        }

        // Keep the console open in debug mode.
        Console.WriteLine("Press any key to exit.");
        Console.ReadKey();
    }
}
/*
Output:
1
2
3
4
*/
```

- a) Continue;
- b) Break;**
- c) Return
- d) Goto

# UF1: Preguntas

¿Qué tipo de operador es %?

- a) Aritmético
- b) Lógico
- c) De comparación
- d) booleano

# UF1: Preguntas

¿Qué tipo de error es el que muestra la imagen? [Error de compilación](#)

0 referencias

```
static void Main(string[] args)
{
    int i, j;

    for (i = 1; i <5; i++)
    {
        for (j = 1; j <5; j++)
        {
        }
    }
    Console.ReadKey();
}
```

$x \% y$  es como  $\rightarrow x += y$  y sería lo mismo que  $\rightarrow x = x/y$

El término de expresión '}' no es válido  
Se esperaba ;

# UF1: Preguntas

En un bucle do-while:

- a) Entraremos 0 o más veces dependiendo de la condición del while
- ☒ b) Entraremos como mínimo una vez Con WHILE a secas si la condición no se cumple no entraríamos ni una sola vez
- c) Entraremos 0 o más veces dependiendo de la condición del do
- d) Ninguna respuesta es correcta

# UF1: Preguntas

¿Es correcto este código?

```
0 referencias
static void Main(string[] args)
{
    String cadena = "estudio en Ilernaonline";
    int i = cadena.Length;
    Console.WriteLine(i);
}
```

- a) No. No podemos asignar el método Length a una variable entera
- ☒ b) Sí. La salida será un número entero que indica el tamaño de la cadena
- c) No. Se produce un error en tiempo de ejecución
- d) Sí. La salida será un número entero que indica el tamaño de la cadena hasta el primer espacio en blanco



# UF1: Preguntas

Realiza un pequeño código donde se explique la sentencia switch

# UF1: Preguntas

Realiza un pequeño código donde se explique la sentencia switch

```
0 referencias
static void Main(string[] args)
{
    Console.WriteLine("Ingrese un valor entre 1 y 5:");
    int valor = int.Parse(Console.ReadLine());
    switch (valor)
    {
        case 1:
            Console.WriteLine("uno");
            break;
        case 2:
            Console.WriteLine("dos");
            break;
        case 3:
            Console.WriteLine("tres");
            break;
        case 4:
            Console.WriteLine("cuatro");
            break;
        case 5:
            Console.WriteLine("cinco");
            break;
        default:
            Console.WriteLine("Se ingreso un valor fuera de rango");
            break;
    }
    Console.ReadKey();
}
```

# UF1: Preguntas

¿Qué elemento puede no existir en una sentencia de selección múltiple?

- a) La expresión.
- b) Los case.
- ☒ c) La cláusula default.
- d) El break de dos o más cases

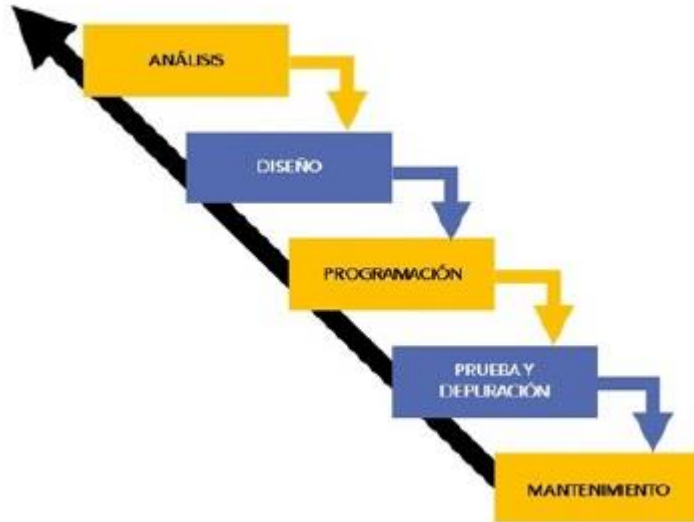
# UF1: Preguntas

¿Cuál es la primera fase del ciclo de vida de un programa informático?

Análisis

# UF1: Preguntas

¿Cuál es la primera fase del ciclo de vida de un programa informático?



# UF1: Preguntas

¿Cuál es la salida del siguiente código? [Error de compilación](#)

```
static void Main(string[] args)
{
    int x = 10; No es booleano para comparar
    int y = 9;

    if (x>y && x)
    {
        Console.WriteLine(x);
    }

    Console.ReadKey();
}
```

# UF1: Preguntas

Según el siguiente código responde a las siguientes cuestiones:

```
0 referencias
static void Main(string[] args)
{
    string cadena;
    int desplazamiento;
    Console.WriteLine("Introduzca cadena");
    cadena = Console.ReadLine();
    Console.WriteLine("-Desplazamiento");
    desplazamiento = int.Parse(Console.ReadLine());

    if (desplazamiento > 26) {
        desplazamiento %= 26;

        for (int i = 0; i < cadena  i++) {
            Console.Write("{0}", (  (cadena[i] + desplazamiento));
        }
    }

    Console.ReadKey();
}
```

Según el código indicado, ¿qué instrucción falta en el primer hueco?

- a) charAt()
- b) indexOf()
- c) subString()
- ☒ d) length()

# UF1: Preguntas

Según el siguiente código responde a las siguientes cuestiones:

```
0 referencias
static void Main(string[] args)
{
    string cadena;
    int desplazamiento;
    Console.WriteLine("Introduzca cadena");
    cadena = Console.ReadLine();
    Console.WriteLine("-Desplazamiento");
    desplazamiento = int.Parse(Console.ReadLine());

    if (desplazamiento > 26) {
        desplazamiento %= 26;

        for (int i = 0; i < cadena[ ] i++) {
            Console.Write("{0}", ( ) (cadena[i] + desplazamiento));
        }
    }

    Console.ReadKey();
}
```

Según el código indicado, ¿qué instrucción falta en el primer hueco?

- a) (char)
- b) cadena[i]
- c) (double)
- d) (int)



# UF1: Preguntas

Según el siguiente código responde a las siguientes cuestiones:

```
O referencias
static void Main(string[] args)
{
    string cadena;
    int desplazamiento;
    Console.WriteLine("Introduzca cadena");
    cadena = Console.ReadLine();
    Console.WriteLine("-Desplazamiento");
    desplazamiento = int.Parse(Console.ReadLine());

    if (desplazamiento > 26) {
        desplazamiento %= 26;

        for (int i = 0; i < cadena.Length; i++) {
            Console.Write("{0}", (cadena[i] + desplazamiento));
        }
    }

    Console.ReadKey();
}
```

La línea 'desplazamiento %=26;', ¿es correcta?

a) No existe este tipo de instrucciones en C#.

b) Sí. Estamos realizando un % de una variable

☒ c) Sí. Estamos realizando una división y guardando el resto de esta

d) Sí. Estamos realizando una división y guardando el cociente de esta

# UF1: Preguntas

Según el siguiente código responde a las siguientes cuestiones:

```
0 referencias
static void Main(string[] args)
{
    string cadena;
    int desplazamiento;
    Console.WriteLine("Introduzca cadena");
    cadena = Console.ReadLine();
    Console.WriteLine("-Desplazamiento");
    desplazamiento = int.Parse(Console.ReadLine());

    if (desplazamiento > 26) {
        desplazamiento %= 26;

        for (int i = 0; i < cadena.Length; i++) {
            Console.Write("{0}", (cadena[i] + desplazamiento));
        }
    }

    Console.ReadKey();
}
```

¿Qué realiza el código?

Desplaza cada carácter el número de veces indicado

