

ACTIVIDAD PRÁCTICA 7 (AP7)

Título

Ejercicios en C# (Programación estructurada)

Objetivos

- Realizar aplicaciones de consola básicas en C# utilizando programación estructurada.
- Dominar las diferentes formas de conversión de tipos utilizadas en lenguaje C#.
- Dominar el uso de estructuras de decisión, estructuras de repetición y programación modular utilizando el lenguaje C#.
- Utilizar métodos matemáticos (Math) y diversos métodos y propiedades de las clases String y Array.
- Saber gestionar y solventar de forma adecuada errores de compilación en el desarrollo de programas.

Temporalización

Previsto: Parte de diversas sesiones de clase presenciales + trabajo en casa durante el siguiente período (del 10 de diciembre al 07 enero).

Proceso de desarrollo

Tras haber trabajado durante la 1ª evaluación la programación estructurada y modular con el lenguaje JavaScript, el objetivo de esta actividad práctica (AP7) es adaptarse a la sintaxis y características del lenguaje de programación C# que utilizaremos durante la 2ª y 3ª evaluación.

En esta actividad práctica se propone la realización de ejercicios que ya han sido trabajados en las actividades prácticas del bloque temático 1 (AP1 a AP4), pero en esta ocasión utilizando el lenguaje de programación C#.

Deberás **realizar al menos 15 ejercicios** de todos los propuestos en la siguiente lista. Los **marcados con fondo gris son obligatorios**, y del resto podrás seleccionar aquellos que creas más interesantes.

Deberás comprimir todas tus soluciones en un fichero **AP7.zip o AP7.rar** y subirlo como entrega de la actividad práctica. En los comentarios de entrega deberás indicar, a parte de los ejercicios establecidos como obligatorios, cuales has realizado hasta completar los 15 necesarios, y los que has incluido de forma adicional.

EJERCICIO 01. Programa que tras leer un número entre 1 y 10, muestre su equivalente en número romano.

EJERCICIO 02. Programa que lea dos frases por teclado y compruebe si son iguales.

EJERCICIO 03. Programa que permita calcular el número de CDs necesarios para hacer una copia de seguridad de la información almacenada en un HD cuya capacidad se conoce (deberás pedirla por teclado). Considera que el disco duro está lleno de información y su capacidad expresada en GigaBytes. Un CD virgen tiene 700 Megabytes de capacidad y un Gigabyte es igual a 1.024 megabytes.

EJERCICIO 04. Programa que calcule el mayor de tres números proporcionados por teclado. Previamente a ello debe comprobar que los tres números suministrados son distintos. En caso de que no lo sea debe avisar que los números suministrados no son válidos.

EJERCICIO 05. Programa que, tras pedir la medida de los tres lados de un triángulo identifique qué tipo es (escaleno, equilátero o isósceles). Averigua cual es la característica de cada uno de ellos que te puede ayudar a identificarlos y pensar la lógica de programación a emplear.

EJERCICIO 06. Programa que solicite una vocal e indique el número / orden de esta, teniendo en cuenta que la “A” sería la primera y la “U” sería la quinta.

EJERCICIO 07. Programa que permita mostrar el sueldo promedio de 5 empleados. El programa deberá leer el sueldo de cada empleado en euros y calcular su promedio. Realiza una modificación posterior de tal forma que también informe del sueldo máximo introducido.

EJERCICIO 08. Elabora un programa que solicite dos números y muestre su producto por el método de las sumas sucesivas:

Por ejemplo: $5 \times 4 = 20 \rightarrow 5 + 5 + 5 + 5 = 20$ (sumamos 4 veces el número 5)

EJERCICIO 09. Programa que muestre los números múltiplos de 3 que existen entre el 1 y el 100.

EJERCICIO 10. Programa que muestre los términos de la siguiente serie (1, 3, 7, 15, 31, ...) que sean menores de 1000.

EJERCICIO 11. Elabora un programa que pida números por teclado y finalice cuando introduzcamos el número 0. Debe informarnos cuantos números se han introducido el total y de ellos cuantos son pares y cuantos impares.

EJERCICIO 12. Programa que permita ingresar caracteres y nos indique al final cuántos se han introducido entre el ‘0’ y ‘9’. El programa finalizará cuando se le introduzca el carácter ‘.’ (Punto).

EJERCICIO 13. Programa que permita obtener una combinación de lotería primitiva (6 números entre 1 y 49). Para generar los números de forma aleatoria puedes utilizar la clase *Random* del namespace System.

EJERCICIO 14. Programa que permita mostrar la suma de los números impares comprendidos entre dos valores numéricos positivos introducidos por teclado.

EJERCICIO 15. Programa que pida por teclado una contraseña. Mientras la contraseña suministrada sea distinta a la palabra “ALDEBARAN” deberá seguir pidiéndola. Si se introduce la contraseña correcta el programa deberá informar de ello. Deberás controlar que sólo hay 5 intentos para introducir la contraseña. Tras agotarlos el programa debe finalizar informando que no se ha suministrado la contraseña correcta.

EJERCICIO 16. Programa que lea temperaturas obtenidas en observatorios meteorológicos. Al finalizar las lecturas, debe informar de cuál ha sido la temperatura máxima y mínima correspondiente. El programa finalizará cuando introduzcamos como temperatura el siguiente valor ‘99’.

EJERCICIO 17. Programa que permita adivinar al usuario un determinado número entero y positivo entre 1 y 100. El programa deberá generar el número de forma aleatoria (recuerda utilizar la clase *Random* del namespace System). Tras la generación, el programa deberá ir pidiendo números e informando al usuario si dicho número es mayor o menor que el que hay que adivinar. Una vez adivinado, informa al usuario del número de intentos que ha realizado para acertar.

EJERCICIO 18. Programa que permita calcular la gratificación de cada empleado de una empresa con motivo de la celebración del 20 aniversario de la organización. Se deberá de leer el nombre y número de años que el empleado lleva en la empresa (antigüedad). La gratificación se calculará en base a la siguiente tabla:

Años trabajados en la empresa	Gratificación en euros
Más de 3 años y menos de 10 años	100 euros por cada año de antigüedad
Entre 10 y 15 años (ambos incluidos)	115 euros por cada año de antigüedad
Más de 15 años	130 euros por cada año de antigüedad

El programa deberá finalizar cuando se responda “N” a la siguiente pregunta ¿Deseas calcular la gratificación de otro empleado (S/N)?

EJERCICIO 19. Programa que permita calcular e imprimir los números primos que hay entre el 2 y el 1000. Un número entero mayor que 1 se denomina número primo si sólo tiene como divisores positivos (factores) a sí mismo y a la unidad.

EJERCICIO 20. Realiza un programa que, tras introducir el precio inicial de un producto sin IVA, calcule su precio final teniendo en cuenta que:

- Se debe aplicar un tipo de IVA que introducido por teclado. El IVA permitido estará entre 16% y 21% incluidos. Mientras el tipo de IVA no esté en ese rango, deberá seguir pidiéndolos.

El programa debe informar del precio del producto sin IVA y con IVA, y finalizará cuando se introduzca un producto con precio inicial igual cero (0).

EJERCICIO 21. Programa que pida un número y muestre si el número es múltiplo de 10. Deberás utilizar una función esMultiplo, que recibirá el número correspondiente y devuelva true si es múltiplo de 10 y false si no lo es.

EJERCICIO 22. Programa que pida la edad de 2 hermanos e informe de la diferencia de edad existente entre ambos. Debes utilizar una función diferenciaEdad, que reciba las dos edades y retorne en valor positivo la diferencia entre ambos.

EJERCICIO 23. Programa que pida el número de partidos ganados, perdidos y empatados por un equipo en un torneo de fútbol. Debe calcular y mostrar el total de los puntos obtenidos, teniendo en cuenta que por cada partido ganador obtendrá 3 puntos, empatado sería 1 punto y perdido 0 puntos. Debes utilizar una función calculaPuntos que recibe como parámetros el número de partidos ganados, perdidos y empatados y retorna la cantidad de puntos totales obtenidos.

EJERCICIO 24. Programa que solicite la entrada un carácter por teclado hasta que se introduzca un punto '.'. En dicho momento deberá parar la ejecución. Mientras no se introduzca el punto, para cada letra introducida deberá informar si es un dígito (0 a 9), un carácter alfabético (a..z / A..Z). Utiliza una función tipoCaracter que reciba el carácter y devuelva si es dígito, alfabético u otro.

EJERCICIO 25. Programa que muestre la suma de los números pares comprendidos entre dos valores numéricos positivos introducidos por teclado. Debes utilizar una función calculaSuma que reciba los dos valores que acotan el rango y devuelva el resultado de la suma esperado.

EJERCICIO 26. Programa que pida un número positivo por teclado e informe si es un número primo o no. Un número entero mayor que 1 se denomina número primo si sólo tiene como divisores positivos (factores) a sí mismo y a la unidad. Para resolver el ejercicio deberás utilizar una función primo que recibirá el número a analizar y devolverá si es primo –true- o no lo es –false-.

EJERCICIO 27. Realiza un programa que, tras introducir precios iniciales de productos, calcule sus precios finales teniendo en cuenta que:

- Se debe aplicar un descuento que puede variar entre el 0% y el 20%. El descuento debe ser pedido por teclado.
- Se debe aplicar un tipo de IVA (16%) que es siempre el mismo.

El programa debe visualizar el precio del producto final (aplicado el descuento y el IVA) debiendo calcularse este dato mediante la utilización de una función. El programa finalizará cuando se introduzca un producto con precio inicial igual cero (0).

EJERCICIO 28. Desarrollar un programa en el que se pida al usuario un array de 5 números enteros e indique en pantalla la media aritmética de todos sus elementos.

EJERCICIO 29. Crea un array de 10 posiciones de números (con valores pedidos por teclado). Visualiza el índice y el valor al que corresponde. Haz dos funciones, una para rellenar los valores del array y otra para mostrar sus elementos.

EJERCICIO 30. Programa que pida por teclado 10 números y los almacene en un array. Una vez introducidos los números, debe informar posteriormente sobre cuantos números son PARES y cuantos IMPARES.

EJERCICIO 31. Desarrollar un programa en el que se pidan al usuario dos listas de números enteros e indique en pantalla el producto de los elementos que ocupan la misma posición. En la petición de cada lista, la introducción de elementos terminará cuando se suministre el valor cero (o). Si las listas proporcionadas no tienen la misma longitud, se ha de escribir en pantalla un mensaje indicativo y no se realizará el proceso.

EJERCICIO 32. Crea dos Arrays de números positivos entre 0 y 20 que sean generados de forma aleatoria. El tamaño de los Arrays debe ser de 10 elementos. Después, crea una función que reciba como parámetros los dos Arrays creados y devuelva un nuevo Array con la suma de la posición cero del array1 con el del array2 y así sucesivamente hasta su último elemento. Por último, muestra el contenido de cada array (array1, array2 y el resultante de la suma de los elementos de ambos).

EJERCICIO 33. Desarrollar un programa en el que se pidan al usuario un array de 5 números enteros. Tras visualizar todos los números introducidos, debe indicar si dicho vector es capicúa, es decir, la secuencia de sus elementos es igual vista de delante hacia atrás y de detrás hacia delante.

EJERCICIO 34. Crea una cadena de caracteres (string) que contenga de la 'A' a la 'Z' (solo las mayúsculas). Después, ve pidiendo posiciones por teclado y si la posición es correcta, la letra correspondiente a esa posición se añadirá a un nuevo string que se mostrará al final del programa. Se dejará de insertar posiciones cuando se introduzca un -1. Por ejemplo, si escribo los siguientes números:

```
0 // Añadirá la 'A'
5 // Añadirá la 'F'
25 // Añadirá la 'Z'
50 // No hará nada ya que la posición no pertenece al array.
-1 // Finalizará y visualizará "Cadena resultante: A,F,Z"
```

EJERCICIO 35. En España cada persona está identificada con un Numero de Identificación Fiscal (NIF) en el que figura un número y una letra, por ejemplo: 56999545W.

Crea un programa que calcule la letra de un NIF a partir del número de DNI que introduzca el usuario. Es decir, se debe pedir el DNI sin la letra por teclado y el programa nos devolverá el DNI completo (con la letra), es decir, el NIF.

Para calcular la letra a partir del DNI se sigue el siguiente algoritmo: se debe tomar el resto de dividir nuestro número de DNI entre 23. El resultado debe estar por tanto entre 0 y 22. La letra correspondiente será a la que apunte dicho índice en la siguiente tabla:

Crea una función obtenerLetra(numeroDNI) donde según el resultado de la anterior fórmula busque en un string/array de caracteres la posición que corresponda a la letra.

RESTO	0	1	2	3	4	5	6	7	8	9	10	11	12
LETRA	T	R	W	A	G	M	Y	F	P	D	X	B	N
RESTO	13	14	15	16	17	18	19	20	21	22			
LETRA	J	Z	S	Q	V	H	L	C	K	E			

Por ejemplo, si introducimos el DNI 20267079, el resto de dividirlo por 23 sería 8, luego la letra sería la P, que es la que ocupa esa posición en la tabla de caracteres.

Evaluación

Esta actividad práctica no es una actividad de evaluación. Su realización forma parte del 10% de evaluación correspondiente al seguimiento e interés de la asignatura por parte del alumnado. Además, supone un refuerzo / estudio de cara a la realización de las actividades de evaluación entregables y las pruebas teórico-prácticas.

Recursos

Disponibles en plataforma (Recursos didácticos del Bloque2):

- Transparencias del Bloque 2.
- Libro1. C#: Guía total del programador
- Documentación de ayuda .NET de Microsoft.