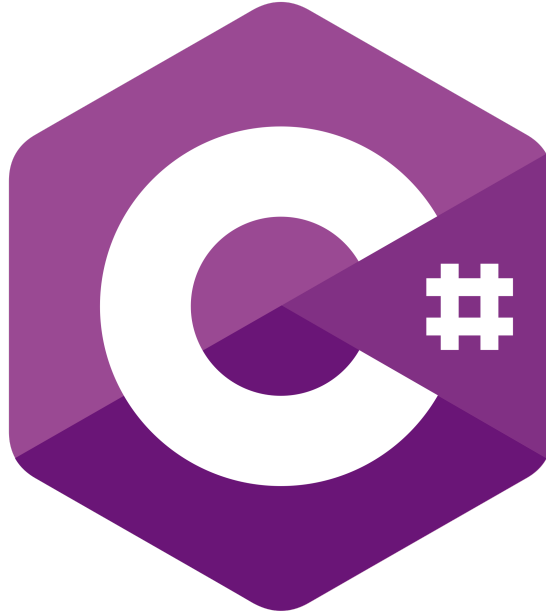


Workbook de C# 2



Esta hoja de ejercicios tiene como objetivo asentar los conocimientos adquiridos durante la primera y segunda semanas del **Módulo 0: Fundamentos de programación**

Esta hoja de ejercicios abarca

- Sintaxis básica de C#
- Algoritmos sencillos
- Objetos
- Algoritmos no tan sencillos.

Cómo usar este workbook

- Para cada ejercicio crear un nuevo [dotnetfiddle](#)¹.
- Cada ejercicio proporciona un problema. La expectativa es que describamos pretendemos solucionarlo con nuestras palabras en un comentario al comienzo del dotnetfiddle.
- Tratar de resolver el ejercicio cumpliendo las condiciones establecidas.

¹ Podemos crear el proyecto en nuestro ordenador con Visual Studio, pero es más sencillo en caso de querer enseñárselo a los compañeros utilizar .

- Puedes ver este documento en [Google Docs](#), e incluso hacer una copia para escribir tus resultados.

Índice

[Cómo usar este workbook](#)

[Índice](#)

[Recordatorio de Sintaxis](#)

[Programa sencillo](#)

[Crear distintos tipos de variables](#)

[Recorrer elementos de un string con un bucle for](#)

[Crear un array de enteros y acceder a sus elementos](#)

[Bucle While](#)

[Problemas Básicos](#)

[Problema 1](#)

[Descripción del problema con tus palabras](#)

[Solución en C#](#)

[Problema 2](#)

[Descripción del problema con tus palabras](#)

[Solución en C#](#)

[Problema 3](#)

[Descripción del problema con tus palabras](#)

[Solución en C#](#)

[Problema 4](#)

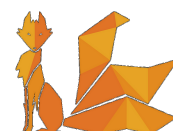
[Descripción del problema con tus palabras](#)

[Solución en C#](#)

[Problema 5](#)

[Descripción del problema con tus palabras](#)

[Solución en C#](#)



[Descripción del problema con tus palabras](#)

[Solución en C#](#)

[Problema 7](#)

[Descripción del problema con tus palabras](#)

[Solución en C#](#)

[Problema 8](#)

[Descripción del problema con tus palabras](#)

[Solución en C#](#)

[Problema 9](#)

[Descripción del problema con tus palabras](#)

[Solución en C#](#)

[Problema 10](#)

[Descripción del problema con tus palabras](#)

[Solución en C#](#)

[Problema 11](#)

[Descripción del problema con tus palabras](#)

[Solución en C#](#)

[Problemas con Listas y Strings](#)

[Problema 12](#)

[Descripción del problema con tus palabras](#)

[Solución en C#](#)

[Problema 13](#)

[Descripción del problema con tus palabras](#)

[Solución en C#](#)

[Problema 14](#)

[Descripción del problema con tus palabras](#)

[Solución en C#](#)

[Problema 15](#)

[Descripción del problema con tus palabras](#)

[Solución en C#](#)

[Problema 16](#)

[Descripción del problema con tus palabras](#)

[Solución en C#](#)

[Problema 17](#)

[Descripción del problema con tus palabras](#)

[Solución en C#](#)

[Problema 18](#)

[Descripción del problema con tus palabras](#)

[Solución en C#](#)

[Problema 19](#)

[Descripción del problema con tus palabras](#)

[Solución en C#](#)

[Problema 20](#)

[Descripción del problema con tus palabras](#)

[Solución en C#](#)

[Problema 21](#)

[Descripción del problema con tus palabras](#)

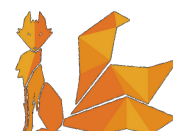
[Solución en C#](#)

[Problema 22](#)

[Descripción del problema con tus palabras](#)

[Solución en C#](#)

[Problema 23](#)



[Descripción del problema con tus palabras](#)

[Solución en C#](#)

[Problema 24](#)

[Descripción del problema con tus palabras](#)

[Solución en C#](#)

[Problema 25](#)

[Descripción del problema con tus palabras](#)

[Solución en C#](#)

[Problema 26](#)

[Descripción del problema con tus palabras](#)

[Solución en C#](#)

[Problema 27](#)

[Descripción del problema con tus palabras](#)

[Solución en C#](#)

[Problema 28](#)

[Descripción del problema con tus palabras](#)

[Solución en C#](#)

[Problema 29](#)

[Descripción del problema con tus palabras](#)

[Solución en C#](#)

[Autoevaluación](#)

[Notas finales](#)

Recordatorio de Sintaxis

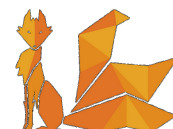
En esta sección se encuentran breves ejemplos de los patrones de código que utilizamos de forma más habitual.

En algunos casos hay segmentos de pseudocódigo, usar con discreción.

Programa sencillo

```
using System;

namespace nuestro_namespace
{
    class Program
    {
        static void Main(string[] args)
        {
            //Aquí va nuestro código.
        }
    }
}
```



Crear distintos tipos de variables

```
int edad = 45;           //Número entero
string nombre = "Félix"; //String, texto
double saldo = 24.13;    //Número con decimales
```

Recorrer elementos de un string con un bucle for

```
string frase = "Esto es una frase";
for (int i = 0; i < frase.Length; i++)
{
    Console.WriteLine(frase[i]);
}
```

Crear un array de enteros y acceder a sus elementos

```
int[] listaNumeros = new int[10];
listaNumeros[4] = 109;
Console.WriteLine(listaNumeros[4]);
```

Bucle While

```
while(condicion) { //Si sentencia se resuelve en true, el bucle se repite
    sentencia;
    sentencia;
    sentencia;
}
```


Problemas Básicos

Problema 1

Crear un programa que escriba 'Hola Mundo' en la pantalla.

Descripción del problema con tus palabras

Solución en C#

Problema 2

Crear un programa que le pida al usuario su nombre y lo salude por su nombre.

Descripción del problema con tus palabras

Solución en C#

Problema 3

Modificar el programa anterior para que solo los usuarios llamados Alicia y Bernardo sean saludados por su nombre.

Descripción del problema con tus palabras

Solución en C#

Problema 4

Escribir un programa que pida al usuario un número n y escriba la suma de los números entre 1 y n.

Descripción del problema con tus palabras

Solución en C#

Problema 5

Modificar el programa anterior para que solo los múltiplos de tres y cinco sean considerados para la suma, por ejemplo 3, 5, 6, 9, 10, 12, 15 para $n=17$

Descripción del problema con tus palabras

Solución en C#

Problema 6

Escribir un programa que pida al usuario un número n y le de la posibilidad de calcular la suma o el producto de todos los números entre 1 y n .

Descripción del problema con tus palabras

Solución en C#

Problema 7

Escribir un programa que escriba las tablas de multiplicar para los números hasta el 12.

Descripción del problema con tus palabras

Solución en C#

Problema 8

Escribir un programa que escriba TODOS los números primos (o hasta que no quepan en un tipo *decimal*)

Descripción del problema con tus palabras

Solución en C#

Problema 9

Escribir un juego de adivinanzas, donde el usuario ha de adivinar un número secreto. Tras cada intento el programa le dice al usuario si el número era demasiado grande o demasiado pequeño. Al final, tras acertar, se muestra el número de intentos que fueron necesarios. Si se introduce el mismo número varias veces seguidas solo cuenta como un intento.

Descripción del problema con tus palabras

Solución en C#

Problema 10

Escribir un programa que escriba en pantalla los próximos 20 años bisiestos.

Descripción del problema con tus palabras

Solución en C#

Problema 11

Escribir un programa que escriba la suma de una serie alternada donde cada elemento de la serie es una expresión de la forma: $((-1)^{k+1})/(2*k-1)$ para cada valor de k entre 1 y un millón. multiplicado por 4. O en notación matemática:

$$4 \cdot \sum_{k=1}^{10^6} \frac{(-1)^{k+1}}{2k-1} = 4 \cdot (1 - 1/3 + 1/5 - 1/7 + 1/9 - 1/11...)$$

Descripción del problema con tus palabras

Solución en C#

Problemas con Listas y Strings

Problema 12

Escribir una función que retorne el elemento más grande de una lista.

Descripción del problema con tus palabras

Solución en C#

Problema 13

Escribir una función que invierta el orden de una lista, preferiblemente sin crear una nueva.

Descripción del problema con tus palabras

Solución en C#

Problema 14

Escribir una función que compruebe si un elemento se encuentra en una lista.

Descripción del problema con tus palabras

Solución en C#

Problema 15

Escribir una función que retorne una nueva lista solo con los elementos en posiciones impares de una lista.

Descripción del problema con tus palabras

Solución en C#

Problema 16

Escribir una función que reciba una lista de números y retorne otra lista con el total acumulado para cada posición (la suma del elemento de dicha posición y todos los anteriores)

Descripción del problema con tus palabras

Solución en C#

Problema 17

Escribir una función que compruebe si un string es un palíndromo.

Descripción del problema con tus palabras

Solución en C#

Problema 18

Escribir tres funciones para obtener la suma de los números en una lista, usando un bucle for, usando un bucle while y usando recursividad.

Descripción del problema con tus palabras

Solución en C#

Problema 19

Escribir una función que concatene dos listas `[a,b,c], [1,2,3] → [a,b,c,1,2,3]`

Descripción del problema con tus palabras

Solución en C#

Problema 20

Escribir una función que combine dos listas alternando los elementos, por ejemplo:

`[a,b,c]`, `[1,2,3]` → `[a,1,b,2,c,3]`

Descripción del problema con tus palabras

Solución en C#

Problema 21

Escribir una función que combine dos listas ordenadas en una nueva lista ordenada

`[1,4,6]`, `[2,3,5]` → `[1,2,3,4,5,6]`. Una forma de hacer esto más rápido es combinar las listas y luego utilizar la función sort.

Descripción del problema con tus palabras

Solución en C#

Problema 22

Escribir una función que rote una lista 'k' elementos. Por ejemplo la lista `[1,2,3,4,5,6]` rotada con k igual a 2 se convierte en `[3,4,5,6,1,2]` intenta solucionarlo sin copiar la lista. ¿Cuántas operaciones de intercambio o desplazamiento necesitas?

Descripción del problema con tus palabras

Solución en C#

Problema 23

Escribir una función que calcule la lista de los primeros 100 números de Fibonacci. Los dos primeros números de Fibonacci son 1 y 1. El número 'n + 1' ésimo de Fibonacci se puede calcular sumando el enésimo y el 'n - 1' ésimo números de Fibonacci. Por lo tanto los primeros son: 1, 1, $1+1 = 2$, $1+2 = 3$, $2+3=5$, $3+5=8$.

Descripción del problema con tus palabras

Solución en C#

Problema 24

Escribe una función que reciba un número y retorne una lista de todos sus dígitos. Por ejemplo **2342** debería retornar **[2, 3, 4, 2]**.

Descripción del problema con tus palabras

Solución en C#

Problema 25

Escribe funciones para sumar, restar y multiplicar dos números en su formato de lista de dígitos (y que retornen otra lista de dígitos). Si eres ambicioso puedes implementar la [multiplicación Karatsuba](#). Prueba diferentes [bases](#). ¿Cual es la base donde las operaciones son más rápidas?

Descripción del problema con tus palabras

Solución en C#

Problema 26

Escribe una función que reciba un número en formato lista de dígitos, una base inicial 'b1' una base objetivo 'b2' y convierta el número a la base 'b2' también en formato lista de dígitos. Por ejemplo `[2,1,0]` en base 3 se convierte en base 10 en `[2,1]`

Descripción del problema con tus palabras

Solución en C#

Problema 27

Implementa los siguientes algoritmos de ordenación: Selection sort, Insertion Sort, Merge Sort, Quick Sort, Stoooge Sort. Busca en Wikipedia las descripciones.

Descripción del problema con tus palabras

Solución en C#

Problema 28

Implementa una búsqueda binaria.

Descripción del problema con tus palabras

Solución en C#

Problema 29

Escribe una función que reciba una lista de strings y los imprima, uno por línea, en un marco rectangular. Por ejemplo la lista `["Hello", "World", "in", "a", "frame"]` se convierte en


```
*****  
* Hello *  
* World *  
* in    *  
* a     *  
* frame *  
*****
```

Descripción del problema con tus palabras

Solución en C#

Autoevaluación

Pregúntate estas cosas. Pueden ayudarte a descubrir si tienes alguna carencia en alguno de los conceptos que hemos tratado en el módulo hasta ahora.

- ☐ ¿Has tenido problemas con la sintaxis?
- ☐ ¿Confundes arrays, objetos, variables, métodos...?
- ☐ ¿Si te explican un problema paso a paso eres capaz de escribirlo?
- ☐ ¿Has tenido problemas para expresar un problema como un programa?
- ☐ ¿Has tenido problemas para determinar exactamente qué ha de recibir como parámetros o retornar una función?
- ☐ ¿Crees que serías capaz de volver a realizar los problemas del workbook (sin mirar las soluciones obtenidas) sin atascarte?

En caso de que muchas de las respuestas sean **sí**, intenta descubrir exactamente qué es lo que no entiendes o sabes hacer, prepara una lista y [házmela llegar](#) para que pueda tenerlo en cuenta para resolverlo en clases futuras.

Notas finales

Libro de trabajo elaborado por Marce Concepción con el objetivo de ayudar a los alumnos del **Curso Superior de Programación y Diseño Web** de la **CODIGN ACADEMY de ESM**, promoción de 2021, a asentar los conocimientos impartidos en la primera mitad del módulo de **Introducción al Desarrollo Front End**.

Si tienes interés en utilizar este workbook o sus contenidos con fines diferentes a los indicados **de forma comercial** por favor consúltame por [correo electrónico](mailto:info@nohaywebs.com)² o [Telegram](https://t.me/MarsC)³.

A todos los que se hayan tomado la molestia de leer hasta aquí:

Muchas gracias por la atención y espero que los problemas recogidos en el libro te hayan ayudado a mejorar tus habilidades.

² info@nohaywebs.com

³ <https://t.me/MarsC>