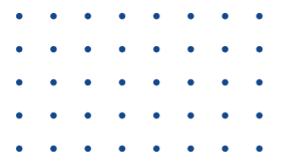
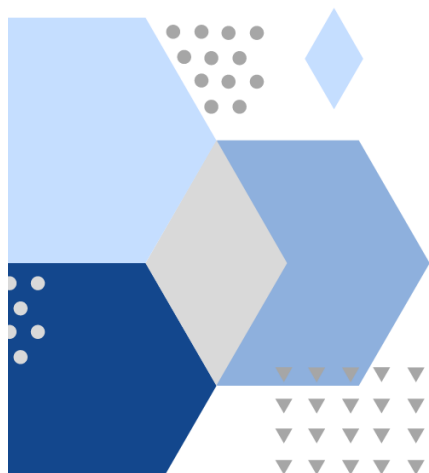


INSTITUTO CEI  
CARRERA ANALISTA PROGRAMADOR

# INFORME DE PRACTICO

PROFESOR: GONZALO ARIEL DUARTE  
ALUMNO: FEDERICO GOLDARACENA



## Resumen.

En el conjunto de ejercicios que se presentan a continuación, se abordan seis problemas distintos, todos ellos resueltos utilizando el lenguaje C#. El enfoque principal de esta serie de ejercicios es proporcionar una oportunidad para aprender y familiarizarse con la sintaxis de C#, además de familiarizarse con el entorno de desarrollo integrado (IDE) de Microsoft, conocido como Visual Studio Community. Cada uno de los ejercicios plantea desafíos específicos relacionados con la programación en C#, lo que brinda al estudiante la oportunidad de aplicar y consolidar su comprensión de los conceptos fundamentales de C#. Además, al trabajar en estos ejercicios utilizando Visual Studio Community, se adquiere experiencia práctica en el uso de esta herramienta de desarrollo ampliamente utilizada en la industria. En resumen, estos ejercicios prácticos en C# representan una valiosa oportunidad para aprender la sintaxis del lenguaje y para familiarizarse con una herramienta esencial en el desarrollo de software, Microsoft Visual Studio Community.

## Contenido

Resumen.	1
1. Introducción	3
1.1. Lenguaje C#	3
1.2. Imprimir una fecha	4
1.3. Imprimir cuadrado de asteriscos	4
1.4. Reloj digital	4
1.5. Calcular calificaciones de grupo alumnos	5
1.6 Calcular cuadrado	5
1.7 Cálculo si es año bisiesto	5
2. Conclusiones	6
3. Referencias	7

# 1.Introducción

## 1.1. Lenguaje C#

C# (pronunciado "C sharp") es un lenguaje de programación de propósito general desarrollado por Microsoft. Fue creado a principios de la década de 2000 y se ha convertido en uno de los lenguajes de programación más populares y ampliamente utilizados en el mundo del desarrollo de software. C# Es conocido por su sintaxis limpia y estructurada, lo que lo hace relativamente fácil de aprender y leer. Este lenguaje se utiliza principalmente para desarrollar aplicaciones de escritorio, aplicaciones web y aplicaciones móviles en el entorno de desarrollo de Microsoft, incluyendo el uso de la plataforma .NET. Una de las características distintivas de C# Es su capacidad de programación orientada a objetos, que permite a los desarrolladores crear software modular y escalable. También ofrece soporte para la programación asíncrona, lo que facilita la creación de aplicaciones que pueden manejar múltiples tareas de manera eficiente.

Microsoft pone al alcance, de toda la comunidad planetaria de programadores, sus plataformas de desarrollo, como Visual Studio Code, de licencia gratuita (Freeware: no pagas por utilizarla), multiplataforma (para Windows, Linux y Mac OS), bajo el entorno .NET Core; y también Visual Studio (de pago), en sus versiones de 2017/2019, para Windows y Mac OS, bajo el entorno de .NET Framework. La diferencia entre ambos entornos es el destinatario final de las aplicaciones, siendo el último solo para Windows, mientras que el primero para las 3 plataformas mencionadas.

Para trabajar con C# es recomendable utilizar Microsoft Visual Studio, ya que ha sido desarrollado especialmente para ello y soporta la carga y trabajo con su mismo lenguaje.

Por ello, nos descargamos Microsoft Visual Studio y ejecutamos el programa. Puede ocurrir que te pida la instalación de algunas librerías esenciales para un correcto funcionamiento del programa, que nos la podemos descargar gratuitamente desde Microsoft.

Una vez ejecutado el programa, puedes crear un documento de tipo C# y ya puedes escribir en este código.[1]

### 1.2POO

## 1.2. Imprimir una fecha

- 1- Leer tres números que denoten una fecha (día, mes, año). Comprobar que es una fecha válida. Si no es válido escribir un mensaje de error. Si es válido escribir la fecha cambiando el número del mes por su nombre. Ej. Si se introduce 1 2 2006, se deberá imprimir "1 de febrero de 2006".

Ejemplo: Se ingresa 3 números enteros para día, mes y año (1, 2, 2006). Al ingresar los datos, se muestra por pantalla los números y mostrando el nombre del mes de la fecha junto con el día y año

## 1.3. Imprimir cuadrado de asteriscos

- 2- Algoritmo que lea un número entero (lado) y a partir de él cree un cuadrado de asteriscos con ese tamaño. Los asteriscos sólo se verán en el borde del cuadrado, no en el interior. Ejemplo, para lado = 4 escribiría:

Ejemplo: Se escribe el número entero, en este caso el 4, Lee el número y luego imprime la cantidad de asteriscos ingresados por el teclado y muestra en forma de cuadrado.

## 1.4. Reloj digital

- 3- Realiza un reloj digital que nunca pare. También debes hacer que espere un segundo real para darle más realismo.

Se inicia en cero las horas, los minutos y segundos (00:00:00). A partir de ahí, empieza el contador en bucle

## 1.5. Calcular calificaciones de grupo alumnos

4- Calcular las calificaciones de un grupo de alumnos. La nota final de cada alumno se calcula según el siguiente criterio: la parte práctica vale el 10%; la parte de problemas vale el 50% y la parte teórica el 40%. El algoritmo leerá el nombre del alumno, las tres notas, escribirá el resultado y volverá a pedir los datos del siguiente alumno hasta que el nombre sea una cadena vacía. Las notas deben estar entre 0 y 10, si no lo están, no imprimirá las notas, mostrará un mensaje de error y volverá a pedir otro alumno.

Pide ingresar datos del alumno como su nombre, la nota práctica, problemas y teórica. Se imprimen los datos del alumno y se mostrará el resultado, queda en bucle hasta que se presione la tecla Enter para salir el programa o ingresará más datos de los alumnos. En caso que la nota ingresada no sea entre 0 y 10, mostrará el mensaje: *“Nota inválida. Ingrese una nota entre 0 y 10”*.

## 1.6 Calcular cuadrado

- 5- Calcular el cuadrado de los N primeros números. Mostrar por pantalla

El programa pide que ingrese un número entero, (ejemplo: 5). Se imprime los primeros números ingresados por teclado y muestra el resultado del cuadrado:  $2^N$

## 1.7 Cálculo si es año bisiesto

6- Algoritmo que, dado un año, nos diga si es bisiesto o no. Un año es bisiesto bajo las siguientes condiciones:

Un año divisible por 4 es bisiesto y no debe ser divisible entre 100.

Si un año es divisible entre 100 y además es divisible entre 400, también resulta bisiesto.

Este algoritmo se encarga de evaluar si un año en particular es bisiesto o no, siguiendo dos condiciones específicas: El año debe ser divisible por 4, sin ser divisible por 100. Si un año cumple con esta primera condición, se considera bisiesto. Por ejemplo, el año 2020 es bisiesto porque es divisible por 4 pero no por 100. En el caso de que un año sea divisible por 100, como el año 1900, todavía puede ser bisiesto si también es divisible por 400. En el caso de 1900, no es bisiesto porque no cumple con esta segunda condición, ya que no es divisible por 400. Sin embargo, el año 2000 es bisiesto porque satisface ambas condiciones: es divisible por 100 y por 400. En resumen, este algoritmo se utiliza para determinar si un año dado tiene un día adicional, el 29 de febrero, en su calendario, basándose en estas dos condiciones.

## 2. Conclusiones

Los ejercicios prácticos presentados en este informe representan una valiosa oportunidad para que los estudiantes y desarrolladores adquieran experiencia práctica en el lenguaje de programación C#. Cada uno de estos ejercicios aborda desafíos específicos relacionados con la programación en C#, lo que les permite aplicar y consolidar su comprensión de los conceptos fundamentales de este lenguaje. Además de la práctica de la sintaxis de C#, estos ejercicios también brindan a los participantes la oportunidad de familiarizarse con el entorno de desarrollo integrado (IDE) de Microsoft, conocido como Visual Studio Community. Esta herramienta es ampliamente utilizada en la industria del desarrollo de software y, por lo tanto, la experiencia adquirida aquí puede resultar beneficiosa en su futura carrera. La diversidad de problemas abordados en estos ejercicios, que van desde la manipulación de fechas hasta el cálculo de calificaciones ponderadas, permite a los estudiantes y desarrolladores desarrollar habilidades de resolución de problemas en diversos contextos de programación. En última instancia, estos ejercicios representan un recurso valioso para aquellos que buscan mejorar sus habilidades en programación en C# y construir una base sólida para el desarrollo de software de alta calidad.

## 3. Referencias

[1] <https://bsw.es/que-es-c/>