

Informe de Trabajo Práctico

N°1 - Parte A: Introducción a C++

Programación Orientada a Objetos
Ingeniería en Mecatrónica

Alumno: Juan Manuel BORQUEZ PEREZ
Legajo: 13567



UNCUYO
UNIVERSIDAD
NACIONAL DE CUYO



**FACULTAD
DE INGENIERÍA**

► **1983/2023**
40 AÑOS DE DEMOCRACIA

1 Consignas

- a) Prepare un directorio denominado `apellido_legajo_A` (reemplace con sus datos) que contenga 2 subdirectorios denominados `version_anterior` y `version_nueva`.
- b) Seleccione 1 ejercicio de los desarrollados en la asignatura Informática usando lenguaje C bajo paradigma estructurado y cópielo en el subdirectorio `version_anterior`. El ejercicio elegido debe satisfacer el siguiente requerimiento:
- Presentar operaciones de entrada/salida por consola.
 - Realizar al menos 2 tareas específicas.
 - Tomar decisiones lógicas en base a expresiones y/o variables.
 - Definir al menos 1 estructura de datos mediante `enum` o `struct`.
 - Usar, al menos, una colección de datos (arreglo, lista, cola, árbol, etc.).
- c) En el directorio `version_nueva`, implemente la misma solución pero adaptando la implementación a las características generales propias de C++. Esta adaptación implica que:
- Cada tarea específica debe implementarse en una función que maneje argumentos adecuados.
 - Las funciones que se definan deben ser consistentes en su diseño y reforzar la cohesión general.
 - Se elimine toda macro y/o variable global (excepto las macros de guarda).
 - Las variables usadas para almacenar datos lógicos deben declararse booleanas.
 - Todos los datos numéricos que se visualicen en pantalla deben hacerlo usando 2 dígitos de precisión y, si corresponde, debidamente encolumnados (utilice como ayuda el código del programa `ajustar_numeros.cpp`).
- d) Verifique que, en las primeras líneas de cada archivo de la nueva versión (`.h`, `.hpp`, `.cpp`), se incorpore información similar a la siguiente:

```
/**
 * @package agregar_si_corresponde
 * Incluya aqui el objetivo del programa, por ejemplo:
 * Aplicacion y Host RPC para control de robots
 * del tipo brazo articulado con 3 grados de libertad.
 *
 * @version 1.0
 * @date 2023.08.11
 * @author Cesar Omar Aranda
 * @contact cesar.aranda@ingenieria.uncuyo.edu.ar
 * @see https://youtu.be/gu-gkutFFo
 * @see https://youtu.be/qFtGcYrb1a4
 * @see https://youtu.be/qFtGcYrb1a4
 * @see https://www.thingiverse.com/thing:3674358
```

```
* @see https://github.com/CesarAranda/HostCLI\_RobotArm
*/
```

- e) Documente, brevemente cada función implementada. Suponiendo que la función sea:

```
double sumar(std::vector<double> & const valores) {
    .....
}
```

Agregue inmediatamente antes de la instrucción anterior, un texto similar a:

```
/**
 * Suma los números almacenados en un vector.
 *
 * @param valores Es el contenedor de los valores a ser sumados.
 * @returns la suma de los 'valores', o 0.0 si 'valores' está vacío.
 */
```

- f) Elabore un informe que documente básicamente la implementación realizada. Salvo indicaciones adicionales que puedan darse en clase, el mismo debe contener2:
- Carátula: donde consten los datos de identificación personales y de contexto.
 - Introducción/enunciado.
 - Esquema general de la solución: representación gráfica con el diseño del modelo lógico.
 - Interfaces de usuario: captura de pantallas de entrada/salida resultantes para una ejecución completa.
 - Uso: descripción de las consideraciones a tener en cuenta por un usuario que desea hacer uso del aplicativo resultante y/o reutilizar su código.
 - Conclusiones:
 - * Comentario sobre los cambios realizados, las dificultades encontradas, las particularidades de la implementación y/o conceptos que requieran atención especial.
 - * Comentario indicando de qué manera puede ser extendido cada uno de los temas del trabajo (si corresponde). **N/A**
 - * Resumen de observaciones/valoración personal a trabajo realizado. **N/A**
 - Referencias bibliográficas y URL (a sitios web, documentos, etc.) consultados.
 - Anexos.
- g) En la carpeta raíz (**apellido.legajo.A**) coloque el documento (en formato PDF preferentemente) con el informe asociado.
- h) Suba un archivo comprimido de la carpeta raíz creada (incluyendo todo su contenido) en la actividad denominada Entrega A, dentro del aula virtual.

1.1 Consigna del ejercicio del Trabajo Práctico de Informática elegido

Se escogió el ejercicio 5 de la Parte 1 - "Estructuras de Datos y Cadenas" del Trabajo Práctico N°3 de Informática, cuya consigna es la siguiente:

5. Modifique el ejercicio 12 del práctico anterior (el de las fechas) utilizando un **struct** que represente una fecha completa. Realice al menos una función que reciba por parámetro una fecha. Defina la estructura **fecha** como un tipo definido por el usuario (**typedef**).

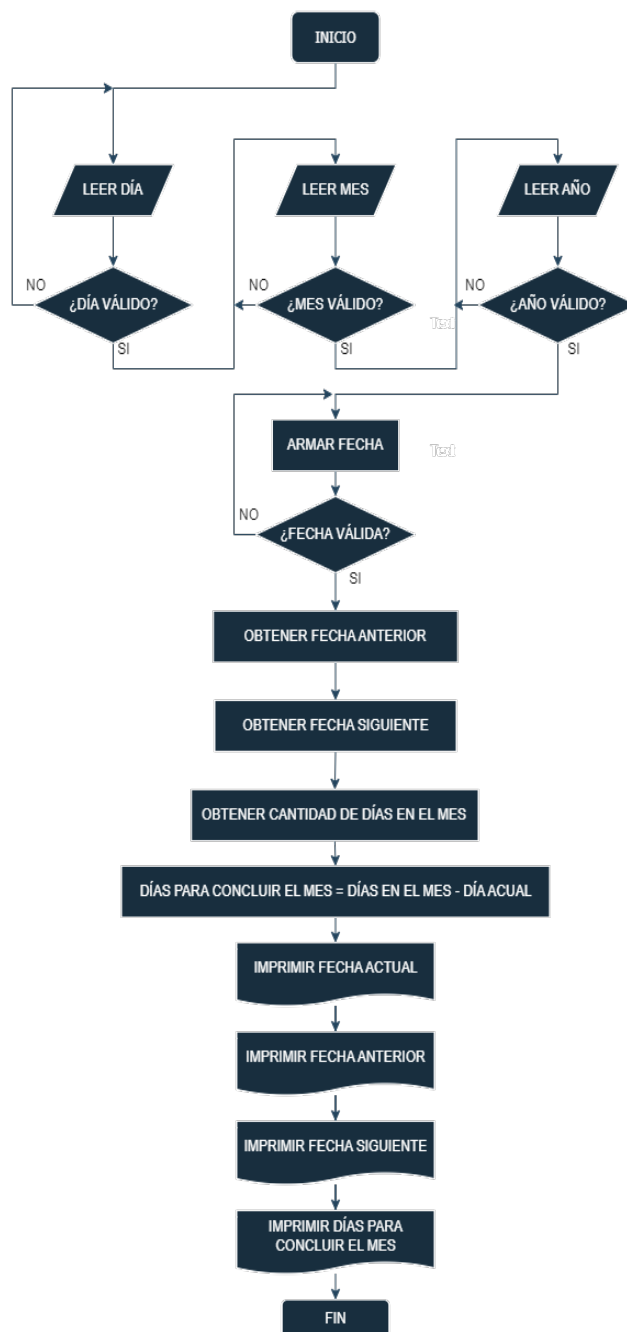
El ejercicio al que se hace referencia en el enunciado es al ejercicio 12 de la Parte 3 - "Subrutinas" del Trabajo Práctico N°2 de Informática, cuya consigna es la siguiente:

12. Escribir un programa que le pida al usuario una fecha del estilo DD/MM/AAAA y determine:
 - a. El día anterior y posterior.
 - b. El último día del mes y cuántos días faltan para el mismo.

Considerar la existencia de los años bisiestos.

Los archivos con las consignas de los mencionados trabajos prácticos se encuentran cargados en el directorio "version_anterior"

2 Esquema General de la Solución



3 Interfaces de Usuario

Secuencia de capturas para una ejecución completa



```
C:\Windows\System32\cmd.exe - Ejercicio5_TP3_A_CPP
C:\Users\Public\Documents\UNCUYO\4-Cuarto año\2-Octavo Semestre\2-Programación Orientada a Objetos\2-Trabajos Prácticos\TP1\BORQUEZ_13567_A\version_nueva>Ejercicio5_TP3_A_CPP
TP3-PARTE I-EJERCICIO 5

A continuación indicar una fecha con el formato DD/MM/AAAA
Si la fecha es incorrecta se solicitará que indique otra nuevamente
Se consideran solamente aquellas fechas desde el 1 de noviembre de 1582 inclusive
Presione ENTER:

C:\Windows\System32\cmd.exe - Ejercicio5_TP3_A_CPP
DD:

C:\Windows\System32\cmd.exe - Ejercicio5_TP3_A_CPP
DD:22

C:\Windows\System32\cmd.exe - Ejercicio5_TP3_A_CPP
MM:

C:\Windows\System32\cmd.exe - Ejercicio5_TP3_A_CPP
MM:8

C:\Windows\System32\cmd.exe - Ejercicio5_TP3_A_CPP
AAAA:

C:\Windows\System32\cmd.exe - Ejercicio5_TP3_A_CPP
AAAA:2023

C:\Windows\System32\cmd.exe
Fecha actual: 22/ 8/2023
Fecha anterior: 21/ 8/2023
Fecha siguiente: 23/ 8/2023
Días para que termine el mes: 9
C:\Users\Public\Documents\UNCUYO\4-Cuarto año\2-Octavo Semestre\2-Programación Orientada a Objetos\2-Trabajos Prácticos\TP1\BORQUEZ_13567_A\version_nueva>
```

4 Uso

La aplicación que se presenta permite la obtención de la fecha anterior, de la fecha siguiente y de la cantidad de días para la conclusión del mes de cualquier fecha del calendario gregoriano indicada por el usuario.

Se consideran como fechas válidas únicamente aquellas que corresponden al calendario gregoriano y por lo tanto son posteriores al 15 de Octubre de 1582 (15/08/1582). Sin embargo, por cuestiones de simplicidad del código, solamente se consideran como válidas una fecha igual o posterior al 1 de Noviembre de 1582 (01/09/1582).

En la ejecución del código se pide al usuario que ingrese el día, el mes y el año de la fecha. En cada entrada de dato se realiza una verificación cíclica. Si se ingresa un número de día erróneo (por ejemplo, un número menor a 1 o mayor a 31, o un valor no numérico), la aplicación solicitará repetidamente un nuevo valor para el día hasta que se ingrese un valor correcto. El mismo proceso se aplica para el mes y el año. Si después de verificar cada dato individual, la fecha en conjunto es errónea debido a que no pertenece al calendario gregoriano, la aplicación también solicitará repetidamente el ingreso de nuevos datos de fecha hasta que se ingrese una fecha válida en su conjunto.

Tenga en cuenta que la aplicación presenta un error: si en el prompt para un dato de la fecha se presiona la tecla "ENTER" antes de introducir un dato numérico (o un carácter alfabético), la aplicación se bloqueará.

5 Conclusiones

5.1 Cambios Realizados en el Código de la Aplicación

La mayoría de los cambios realizados en el código se encuentran indicados en el código fuente de la aplicación como comentarios de línea. Otros de los cambios realizados son los siguientes:

- Los datos de la fecha no se asignan directamente a la estructura de la fecha (la que tiene todos los campos numéricos) y en cambio se lleva a cabo una verificación previa de que los datos ingresados por el usuario sean numéricos. Para ello, se realiza el siguiente cambio.
- Incorporación de funciones para la verificación de los datos de una fecha dados en un string:

```
– bool valid_DD(char* DD);  
– bool valid_MM(char* MM);  
– bool valid_AAAA(char* AAAA);
```

- Incorporación de una función que concentra la muestra de los resultados de la ejecución:

```
void print_out(Date current, Date previous, Date next, short int days_MM);
```

- Se realizaron cambios en la presentación de algunas funciones y correcciones por fallas encontradas en la lógica del programa.

5.2 Dificultades Encontradas

Se utilizó la siguiente forma de tomar los datos de la fecha:

```
cin.get(auxiliar, 3);
```

De esta manera, se almacena en la variable "auxiliar" como máximo dos caracteres de los ingresados por el usuario (para el caso del mes y del día, mientras que son 4 para el caso del año). Si el usuario ingresa más de 2 caracteres antes de presionar "ENTER", el resto de los caracteres en el buffer de entrada son consumidos por la siguiente línea:

```
cin.ignore(numeric_limits<streamsize>::max(), '\n');
```

El problema se presenta cuando el usuario no ingresa ningún carácter antes de presionar "ENTER". En este caso, el programa se bloquea en el ciclo `do while`, ya que aparentemente queda en el buffer de entrada un carácter no imprimible. Esto se pudo determinar por pruebas imprimiendo por pantalla el código ASCII del carácter obtenido del buffer de entrada.

6 Referencias Bibliográficas

[1], [5], [4], [2], [6], [7], [3]

Referencias

- [1] cplusplus.com. *std::setw*. URL: <https://cplusplus.com/reference/iomanip/setw/>.
- [2] cplusplus.com. *std::stoi*. URL: <https://cplusplus.com/reference/string/stoi/>.
- [3] GeeksforGeeks. *cin.get() in C++ with examples*. URL: <https://www.geeksforgeeks.org/cin-get-in-c-with-examples/>.
- [4] GeeksforGeeks. *Clearing the input buffer in C/C++*. URL: <https://www.geeksforgeeks.org/clearing-the-input-buffer-in-c-c/>.
- [5] GeeksforGeeks. *setw() function in C++ with examples*. URL: <https://www.geeksforgeeks.org/setw-function-in-cpp-with-examples/>.
- [6] GeeksforGeeks. *std::stoi() function in C++*. URL: <https://www.geeksforgeeks.org/stdstoi-function-in-cpp/>.
- [7] Stack Overflow. *Why can the cin.get() function strip the n in a string?* URL: <https://stackoverflow.com/questions/28854498/why-can-the-cin-get-function-strip-the-n-in-a-string>.