

ALUMNO:

EPS -Ingeniería Informática

Asignatura: G0460009 Programación II – Grupo A Turno1

Curso: 2022/2023  
Semestre: 2º

Examen: Parcial  
Convocatoria: Ordinaria

Fecha: 13-03-2023

### Parte Práctica (10 puntos; 70% nota final)

#### Tiempo: 1 hora

Los ficheros correspondientes a los ejercicios se deben entregar en la actividad Examen Parcial a través del campus antes de la finalización del tiempo establecido.

Cada fichero se llamará EjercicioX, donde la X será el número de dicho ejercicio. Los únicos formatos válidos serán **txt** o **c++** o **.h**, **sin comprimir**.

#### Criterios generales de evaluación

<b>Funciones/Métodos:</b> Si no se usa el paso por referencia constante cuando las variables de los parámetros de entrada no son de tipo simple.	40%
<b>Tipos de datos y variables:</b> <ul style="list-style-type: none"><li>• Uso de variables globales (fuera del ámbito de una función).</li><li>• Si no se usan los tipos contenedor vistos en clase (<code>std::array</code>; <code>std::vector</code>; <code>std::set</code>; <code>std::string</code>, etc.) para las variables que lo necesiten.</li></ul>	0%
El <b>programa no compila</b> o <b>no se asemeja</b> a lo pedido.	0%
Si no se cumplen los <b>criterios de entrega</b> indicados en la <b>actividad/examen</b> .	0%

#### Criterios particulares de evaluación

El elemento evaluable no compila o no se asemeja a lo que se pide	0%
El elemento evaluable no se aproxima suficientemente a lo pedido	40%
El elemento evaluable se aproxima suficientemente a lo pedido	60%
El elemento evaluable funciona correctamente y las estrategias y elementos de código elegidos son adecuados.	100%

#### IMPORTANTE:

- Todos los ejercicios del examen deberán ser resueltos de forma **algorítmica**, es decir, la **solución** propuesta tendrá que ser **general** y **no particular** para unos determinados datos/valores.



UNIVERSIDAD  
NEBRIJA

- Todos los ejercicios resueltos sin utilizar funciones cuando sea apropiado se valorarán con una nota máxima del 60% sobre la calificación prevista.
- Se recomienda una primera lectura del examen completo para planificar la realización del examen. Y una segunda lectura detallada antes de la realización de cada uno de los ejercicios propuestos.

### Ejercicio N°1 (5 puntos)

Escriba un programa en C++11, *Ejercicio1*, que permita probar el funcionamiento de la clase CFecha según las especificaciones que se describen a continuación. **(1 punto)**

Implementar la clase CFecha:

- Atributos: permite almacenar **dos valores enteros** (mes y año) y un **texto libre** con la descripción (**tipo string**) de dicha fecha.

Además, antes de guardar estos datos se tienen que validar tanto el mes como el año según lo siguiente: **(1 punto)**

- Si el mes no es un valor correcto, se lanza una excepción (**tipo int con valor 1**) para indicarlo.
- Si el año es superior a 2023 se lanza una excepción (**tipo int con valor 2**) para indicarlo.
- Métodos:
  - Constructor paramétrico con las excepciones especificadas. **(0,5 puntos)**
  - Constructor copia. **(0,5 puntos)**
  - El operador “==” sobrecargado para determinar si dos fechas son iguales o no. **(1 punto)**

Además, se debe implementar como función general el Operador “<<” sobrecargado para mostrar por pantalla el contenido de un objeto CFecha. **(1 punto)**

### Ejercicio N°2 (5 puntos)

Escriba un programa en C++11, *Ejercicio2*, que permita probar el funcionamiento de la clase templatizada CPrueba para varios tipos de datos, **al menos dos tipos de datos distintos**. **(1 punto)**

Implementar la **clase templatizada CPrueba**:

- Atributos: **(1 punto)**
  - un índice o posición: valor entero (no estará incluido en la plantilla/template) con el que se accede a una posición del vector.
  - un vector donde los datos se definen al declarar el objeto de la clase.



– Métodos:

- **Constructor paramétrico:** recibe como parámetro un número entero positivo para dimensionar el vector a través de **resize**. Además, el índice se inicializa con **valor -1**. (1 punto)
- **devolverElemento:** no tiene parámetros y devuelve el valor almacenado en la posición del vector indicada por el índice (según el objeto que la invoque). Este método accede con el valor del índice a la posición del vector para recuperar el dato que debe devolver. (1 punto)
- **incluirElemento:** tiene como parámetro un valor del tipo de dato almacenado en el vector (según el objeto que la invoque) y no devuelve nada. Este método incorpora el valor recibido como parámetro al vector en la posición siguiente a la indicada por el índice. Además, incrementa el valor del índice para que quede señalando el que ahora es el último elemento añadido al vector. (1 punto)