

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

PROGRAMACIÓN 2
8va práctica (tipo b)
Primer Semestre 2025

Indicaciones Generales:

Duración: 110 minutos.

NO SE PERMITE EL USO DE APUNTES DE CLASE, FOTOCOPIAS NI MATERIAL IMPRESO

- No se pueden emplear variables globales. NO PUEDE UTILIZAR LA CLASE string. Tampoco se podrán emplear las funciones de C que gestionen memoria como malloc, realloc, memset, strdup, strtok o similares, igualmente no se puede emplear cualquier función contenida en las bibliotecas stdio.h, cstdio o similares y que puedan estar también definidas en otras bibliotecas. **NO PODRÁ EMPLEAR PLANTILLAS EN ESTE LABORATORIO**
- **EL PROYECTO DEBERÁ SER DESARROLLADO BAJO EL PARADIGMA DE PROGRAMACIÓN ORIENTADO A OBJETOS POR LO QUE SALVO EN LA SOBRECARGA DE LOS OPERADORES >> Y <<, NO SE PODRÁN DEFINIR FUNCIONES (NI PLANTILLAS DE FUNCIONES) INDEPENDIENTES QUE NO ESTÉN LIGADAS COMO MÉTODOS A ALGUNA DE LAS CLASES PLANTEADAS.**
- Deberá mantener en todo momento el encapsulamiento de todos los atributos de las clases, esto implica también que un método público NO puede devolver un puntero a algún dato del objeto. También debe guardar los estándares en la definición y uso de todas las clases desarrolladas. No deberá repetir código en los diferentes métodos. No se considerará en la nota las clases que violen esto.
- Cada método NO debe sobrepasar las 20 líneas de código aproximadamente.
- En el archivo main.cpp deberá colocar un comentario en el que coloque claramente su nombre y código, **de no hacerlo se le descontará 0.5 puntos en la nota final.**
- El código comentado NO SE CALIFICARÁ. De igual manera NO SE CALIFICARÁ el código de un método si su llamado está comentado.
- Los programas que presenten errores de sintaxis o de concepto se calificarán en base al 40% de puntaje de la pregunta. Los que no muestren resultados o que estos no sean coherentes en base al 60%.
- Se tomará en cuenta en la calificación el uso de comentarios relevantes.
- **TAMPOCO SE PODRÁ EMPLEAR LA CLÁUSULA protected NI LA CLÁUSULA friend, SALVO EN LAS CLASES AUTOREFERENCIADAS PARA DECLARAR AMIGA A LA CLASE QUE CONTenga AL ÁRBOL, SE PROHÍBE QUE LOS NODOS DECLAREN friend A OTRAS CLASES QUE NO TENGAN QUE VER CON EL ÁRBOL COMO, POR EJEMPLO, LOS ATRIBUTOS DEL NODO, DE HACERLO SE NO SE LE CALIFICARÁN LA PREGUNTA.**

SE LES RECUERDA QUE, DE ACUERDO CON EL REGLAMENTO DISCIPLINARIO DE NUESTRA INSTITUCIÓN, CONSTITUYE UNA FALTA GRAVE COPIAR DEL TRABAJO REALIZADO POR OTRA PERSONA O COMETER PLAGIO.

NO SE HARÁN EXCEPCIONES ANTE CUALQUIER TRASGRESIÓN DE LAS INDICACIONES DADAS EN LA PRUEBA

- **Puntaje total: 20 puntos.**

INDICACIONES INICIALES

Cree un proyecto de C++ en NetBeans siguiendo estrictamente las indicaciones que a continuación se detallan:

- La unidad de trabajo será **t:** (Si lo coloca en otra unidad, no se calificará su laboratorio y se le asignará como nota cero)
- Cree allí una carpeta con el nombre **"CO_PA_PN_Laboratorio08_2025_1"** donde **CO** indica: Código del alumno, **PA** indica: Primer Apellido del alumno y **PN** primer nombre (de no colocar este requerimiento se le descontarán 3 puntos de la nota final). **Allí colocará el proyecto solicitado en la prueba.**

Cuestionario:

La finalidad principal de este laboratorio es la de reforzar los conceptos contenidos en el capítulo 9 del tema: "Clases autoreferenciadas".

PARTE01 (14 puntos): CREACIÓN DE LAS CLASES

Se solicita que desarrolle un proyecto "Lab08_Arbol_Drones_Parte1" dentro de la carpeta correspondiente, DE NO COLOCAR ESTE REQUERIMIENTO SE LE DESCONTARÁ 2 PUNTOS DE LA NOTA FINAL, en la cual se declaren las clases descritas con las relaciones necesarias, que permitan manipularlas empleando herencia:

- **Para manejar los drones:** La clase se denominará "Dron" y deberá contener lo siguiente: 1) un atributo denominado **id** (**char***), 2) un atributo denominado **ubicacion** definido por una cadena de caracteres dinámica (**char***), 3) un atributo denominado **capacidad** (**int**).
- **Para manejar los drones tipo semáforo:** La clase se denominará "DroneSemaforo" y deberá contener lo siguiente: 1) un atributo denominado **luz_roja** (**int** - se interpreta como **bool**, 1 si tiene cámara roja). Además, esta clase posee datos heredados de la clase **Dron**.
- **Para manejar los drones tipo estacionamiento:** La clase se denominará "DroneEstacionamiento" y deberá contener lo siguiente: 1) un atributo denominado **zona_carga** (**int** - se interpreta como **bool**, 1 si se enfoca en la zona de carga). Además, esta clase posee datos heredados de la clase **Dron**.
- **Para manejar los drones tipo velocidad:** La clase se denominará "DroneVelocidad" y deberá contener lo siguiente: 1) un atributo denominado **velocidad_maxima_permitida** (**double**). Además, esta clase posee datos heredados de la clase **Dron**.
- **Para manejar los Nodos:** La clase se denominará "Nodo" y deberá contener lo siguiente: 1) un atributo denominado **dron**, este es un puntero de la clase **Dron**, 2) un atributo denominado **izq**, este atributo es un puntero a la clase **Nodo** (autoreferenciado), 3) un atributo denominado **der**, este atributo es un puntero a la clase **Nodo** (autoreferenciado).
- **Para manejar el árbol:** La clase se denominará "Arbol" y deberá contener lo siguiente: 1) un atributo denominado **raiz**, este atributo es un puntero de clase **Nodo**. Esta estructura se trata de un ABB ordenado por el **id** del dron (ascendente, recuerde en un ABB los nodos del lado izquierdo siempre son menores a la raíz y los del lado derecho siempre son mayores).
- **Para manejar la central de drones:** La clase se denominará "Central" y deberá contener lo siguiente: 1) un atributo denominado **ADrones**, este atributo es un objeto de la clase **Arbol**, donde se guardarán todos los drones que existen en el archivo a cargar.

"DEBE EMPLEAR OBLIGATORIAMENTE LOS NOMBRES DE LAS CLASES, DE SUS ATRIBUTOS Y TIPOS DE DATOS"

NO PUEDE DEFINIR ATRIBUTOS ADICIONALES A LAS CLASES PERO, SI PODRÁ DEFINIR LOS MÉTODOS ADICIONALES QUE CREA CONVENIENTE.

Con las clases indicas debe realizar las siguientes operaciones:

- (5 puntos) En la clase **Central** debe implementar el método **carga**, que se encarga de la lectura del archivo "drones.csv" y cargar la información en el árbol ABB denominada **ADrones**. Para la lectura de los datos del dron debe utilizar el método polimórfico **leer**, de acuerdo con el tipo de dron que aparece en el archivo. Recuerde que el tipo no se almacena en ningún atributo. Los valores para los atributos deben leerse en la clase que los define.
- (5 puntos) En la clase **Central** implementar el método **muestra**, que se encargue de realizar la impresión de un archivo de prueba debidamente tabulado (**sin usar el carácter '\t'**), que muestre los datos del dron. Para este paso debe utilizar el método polimórfico **imprimir**. Los valores de los atributos deben imprimirse en la clase que los define.

Para esta pregunta, por lo menos debe desarrollar los siguientes métodos polimórficos:

- **leer:** para la lectura de los datos de cada uno de los drones de acuerdo con su tipo.
- **imprimir:** para la impresión de los datos de cada uno de los drones de acuerdo con su tipo.

Consideraciones:

Para el desarrollo de ambas preguntas debe considerar el siguiente código, con excepción del método **actualiza** que solo debe estar en la pregunta 2:

```
#include "Central.h"

int main(int argc, char** argv) {
    Central cdrones;

    cdrones.carga();
    cdrones.actualiza();
    cdrones.muestra();

    return 0;
}
```

**NO PUEDE
CAMBIAR
ESTE CÓDIGO**

En la misma carpeta, copie el proyecto de la pregunta anterior con el nombre de **"Lab08_Arbol_Drones_Parte2"**

PARTE 2 (6 puntos): Proceso final.

En la misma carpeta, copie el proyecto de la pregunta anterior con el nombre de **"Lab08_Arbol_Drones_Parte2"** y agréguele el llamado al método **actualiza**. El proyecto ejecutará las tareas descritas a continuación:

- Cargar el árbol **ADrones** de acuerdo con lo indicado a la pregunta anterior.
- Desarrollar un método denominado **actualiza** que pertenece a la clase **Central**, que se encargue de actualizar el atributo **luz_roja** con el valor 0 al 50% de los drones semáforos, de la misma forma actualizar el atributo **zona_carga** con el valor 1 al 50% de los drones estacionamiento. **Es obligatorio para esta operación utilizar polimorfismo con el fin de acceder a los drones de acuerdo con el tipo, en caso contrario la respuesta no tiene valor.**
- Finalmente desarrolle el método **muestra** para imprimir el siguiente reporte:

REPORTE DE DRONES			
Codigo	Ubicacion	Capacidad	Velocidad/Zona/Luz
DE830	Avenida C	15	0
DE885	Avenida I	15	0
DE960	Avenida F	15	0
DE999	Avenida I	15	0
DS071	Avenida B	10	0
..			
DS714	Avenida B	10	1
DS799	Avenida E	10	1
DS863	Avenida H	10	1
DV001	Avenida A	5	60
...			

Se recomienda revisar el archivo que servirá para la lectura de datos, el cual se describe a continuación:

Archivos: Drones.csv

Velocidad,DV050,Avenida Javier Prado,5,60
Semaforo,DS253,Avenida Universitaria,10,1
Estacionamiento,DE190,Avenida Miguel Grau,15,0
...

Contenido: Tipo de Drone, ID del Dron, Ubicación, Capacidad de Almacenamiento, Límite Velocidad/LuzRoja/ZonaCarga

Recuerde que si no usa polimorfismo la respuesta no será válida. Los métodos pueden ser recursivos o iterativos según lo desee. Los métodos no deben devolver o recibir punteros a objetos con excepción de aquellos que se implementen para el recorrido del árbol manejándose como privados.

Al finalizar la práctica, comprima la carpeta dada en las indicaciones iniciales empleando el programa Zip que viene por defecto en el Windows, no se aceptarán los trabajos compactados con otros programas como RAR, WinRAR, 7zip o similares.

Profesores del curso:
Rony Cueva
Andrés Melgar
Miguel Guanira

Erasmus Gómez
Erick Huiza

San Miguel, 20 de junio del 2025.