

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ**  
**FACULTAD DE CIENCIAS E INGENIERÍA**

**PROGRAMACIÓN 2**  
**7ma 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 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, DE HACERLO NO SE LE CALIFICARÁN LAS CLASES INVOLUCRADAS.**

**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\_Laboratorio07\_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 los capítulos 6 y 7 del tema: "Programación orientada a objetos" y "Herencia".

Se solicita que desarrolle un proyecto denominado **"HERENCIA"** dentro de la carpeta correspondiente, **DE NO COLOCAR ESTE REQUERIMIENTO SE LE DESCONTARÁN 3 PUNTOS DE LA NOTA FINAL.**

El Ministerio de Transportes ha implementado un sistema de drones autónomos para observar y registrar infracciones de tránsito desde el aire. Existen distintos tipos de drones, cada uno especializado en observar cierto tipo de infracción, como:

- Drones de velocidad
- Drones de semáforo
- Drones de estacionamiento indebido

Cada uno tiene distintas formas de registrar datos, pero heredan una estructura base común de vigilancia aérea.

El ministerio cuenta con muchos archivos de especificaciones de Drones y Eventos por cada Centro de Monitoreo. Estos archivos se encuentran en la carpeta Datos, la estructura de estos archivos es la siguiente.

**Archivo: Zonas.txt**

```
Lima
Arequipa
Cusco
Trujillo
...
```

**Contenido:** Cada línea contiene el nombre de la Zona de los centros de monitoreo

**Archivos: Drones <ZONA>.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

**Archivos: Eventos <ZONA>.csv**

```
DV050,PLT-100,15/06/2025,65
DS253,PLT-101,15/06/2025,0
DE190,PLT-102,15/06/2025,1
...
```

**Contenido:** ID del Drone, Placa del vehículo que ocasionó el evento, fecha del evento, Velocidad/LuzRoja/ZonaCarga

Se solicita diseñar un proyecto en C++ que aplique los principios de herencia, con el objetivo de modelar adecuadamente la información de los drones y registrar las infracciones ocurridas durante los eventos previamente descritos.

### CREACIÓN DE LAS CLASES (8 puntos)

Para registrar las infracciones cometidas: Defina la clase denominada **"Infraccion"** que deberá contener los siguientes atributos:

1. un atributo denominado **fecha\_evento** (int),
2. un atributo denominado **tipo\_infraccion** (char\* ),
3. un atributo denominado **monto** (double),
4. un atributo denominado **fecha\_pago\_limite** (int),
5. un atributo denominado **placa** (char\* ).

Para esta clase deberá definir obligatoriamente el constructor por defecto, el constructor copia, el destructor, todos los métodos selectores y el operador de asignación.

Para representar drones: Defina la clase denominada **"Drone"** que deberá contener los siguientes métodos y atributos necesarios para su identificación:

1. un atributo denominado **id** (char\* ),
2. un atributo denominado **ubicacion** (char\* ),
3. un atributo denominado **capacidad** (int ),
4. un atributo denominado **cantidad\_infracciones** (int ),
5. un atributo denominado **infracciones** (arreglo estático de **Infraccion** , asuma 20 como número máximo de infracciones ),

---

Para registrar drones de tipo velocidad: Defina la clase denominada **"DroneVelocidad"**, que hereda de la clase **Drone**, y deberá añadir el siguiente atributo:

1. un atributo denominado **velocidad\_maxima\_permitida** (int).

---

Para registrar drones de tipo semáforo: Defina la clase denominada **"DroneSemaforo"**, que hereda de la clase **Drone**, y deberá añadir el siguiente atributo:

1. un atributo denominado **luz\_roja** (int - se interpreta como bool, 1 si tiene cámara roja).

---

Para registrar drones de tipo estacionamiento: Defina la clase denominada **"DroneEstacionamiento"**, que hereda de la clase **Drone**, y deberá añadir el siguiente atributo:

1. un atributo denominado **zona\_carga** (int - se interpreta como bool, 1 si se enfoca en una zona de carga).

---

Para manejar el conjunto de drones: Defina la clase denominada **"Flota"** que deberá contener los siguientes atributos:

1. un arreglo estático **drones\_velocidad** de tipo **DroneVelocidad** con 20 elementos,
2. un arreglo estático **drones\_semaforo** de tipo **DroneSemaforo** con 20 elementos,
3. un arreglo estático **drones\_estacionamiento** de tipo **DroneEstacionamiento** con 20 elementos,
4. un atributo **n\_drones\_velocidad** (int),
5. un atributo **n\_drones\_semaforo** (int),
6. un atributo **n\_drones\_estacionamiento** (int).

---

Para representar un centro regional de control: Defina la clase denominada **"CentralDeMonitoreo"** que deberá contener los siguientes atributos:

1. un atributo **flota** de tipo **Flota**,
2. un atributo **cantidad\_multas\_registradas** (int),
3. un atributo **total\_multas\_acumuladas** (double).

---

Para manejar todos los centros regionales del país: Defina la clase denominada **"MinisterioDeTransportes"** que deberá contener los siguientes atributos:

1. un arreglo estático **centros** de tipo **CentralDeMonitoreo** con 10 elementos,
2. un arreglo estático **zonas** de tipo **char\*** con 10 elementos (una cadena dinámica por cada zona),
3. un atributo **cantidad\_centros** (int).

**"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.**

### **OPERACIONES CON LAS CLASES (12 puntos)**

**Clase Infraccion** Debe implementar los siguientes métodos públicos:

- **void leer(ifstream &arch);** Lee los datos de una infracción desde un archivo de texto.
- **void imprimir(ofstream &arch) const;** Imprime los datos de la infracción en el formato especificado.

---

**Clase Flota** Debe implementar:

- **void cargar\_drone(ifstream &input, const char \*tipo);** Lee la información desde el archivo especificado y actualiza el conjunto de drones del tipo indicado.
- Métodos de búsqueda:
  - **int buscarDroneVelocidad(const char\* id);**
  - **int buscarDroneSemaforo(const char\* id);**
  - **int buscarDroneEstacionamiento(const char\* id);**

- `void actualizarDronesInfracciones(ifstream &arch);` Lee infracciones cometidas desde los archivos de eventos y actualiza los drones correspondientes.
- `imprimir(ofstream &output) const` Imprime la información de cada tipo de dron en la flota.

**Clase CentralDeMonitoreo** Debe implementar:

- `void leer(ifstream &arch, const char* tipo);` Carga la flota de drones según el tipo que corresponda. Deberá llamar al método descrito en flota.
- `void cargar_infracciones(ifstream &arch);` Actualiza la flota de drones con las infracciones en el archivo. Deberá llamar al método descrito en flota.
- `void imprimir(ofstream &arch) const;` Imprime un resumen del centro y su flota, incluyendo la cantidad de multas y el total acumulado.

---

**Clase MinisterioDeTransportes** Debe implementar:

- `void apertura_archivo_leer(ifstream& const char* nombre_archivo);` Abre un archivo de lectura con la ruta pasada por como parámetro
- `void apertura_archivo_imprimir(ofstream& const char* nombre_archivo);` Crea y abre un archivo de escritura con la ruta pasada como parámetro
- `int obtener_ruta(const char* carpeta_base, const char* tipo, const char* zona, char* ruta) const;`  
Método auxiliar para obtener la ruta de un archivo que se usará para leer dado la carpeta\_base("Datos") y el tipo("Drones" o "Eventos" así como también la Zona.
- `void cargar_drones(const char* ruta_base);` Carga el arreglo de CentralDeMonitoreo desde los archivos especificados en la ruta\_base (Drones\_<ZONA>.csv).
- `void procesar_todos_los_eventos(const char* ruta_base);` Carga las infracciones desde los archivos especificados en la ruta\_base (Eventos\_<ZONA>.csv).
- `void imprimir(const char* nombre_archivo);`  
Imprime un reporte general de la información guardada por cada Zona.

**Consideraciones:**

- La carpeta Datos que descargará junto con los archivos del laboratorio, deberá incluirla en el root del mismo.
- Está prohibido en este laboratorio, abrir la totalidad de archivos al mismo tiempo, debe manejar eficientemente la apertura y cerrado de archivos de lectura.
- Está prohibido el uso de la cláusula friend y de la palabra reservada virtual para este laboratorio.
- Deberá actualizar todos los valores totales y cantidades que se disponga a lo largo del proyecto.
- El valor a pagar por las infracciones dependerá de si es una infracción Leve(5%) o Grave(8%) y cada una se calculará dependiendo del valor constante de UIT 5350,
- Una infracción, dependiendo del tipo de infracción se etiquetará como "Leve" si:
  - DroneVelocidad: Si la velocidad registrada es menor a la velocidad máxima permitida en la zona por el dron.
  - DroneSemaforo: Si la luz roja observada es 0 y el dron observa luz\_roja 1 (luz\_roja\_evento < luz\_roja\_dron)
  - DroneEstacionamiento: Si la zona\_carga es 0 y el dron observa zona\_carga 1. (zona\_carga\_evento < zona\_carga\_dron)

Caso contrario, la infracción será "Grave".

- La fecha máxima de pago se establecerá a partir de la fecha del evento, con un plazo de hasta 30 días calendario posteriores para realizar el pago.

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, 06 de junio del 2025.