

Apellido y nombre: _____ Legajo: _____

PRIMER EXAMEN PARCIAL**8/07/2025**

Resolvé el siguiente ejercicio utilizando el lenguaje **Java** y el paradigma orientado a objetos. Lee el enunciado al menos dos veces antes de intentar confeccionar la solución.

ENUNCIADO

Una empresa de logística y eventos nos pide un software para manejar su flota de drones, los cuales pueden servir para misiones de entrega de paquetes o de filmación de eventos.

Todos los drones poseen ID, nivel de batería actual (entre 1 y 100), consumo de batería por cada minuto de vuelo y consumo de batería por cada kilómetro recorrido.

Un Dron Repartidor tiene una capacidad máxima de carga (en kg) y se especializa en trasladar un paquete, del cual conoceremos su peso y volumen en cm³.

Un Dron Filmador se utiliza en servicios de seguridad y cobertura de eventos, por eso, lleva acoplada una cámara para registrar imágenes, de la cual se sabe cuánto consumo de batería por minuto de grabación le resta a la batería del dron que la lleva acoplada.

Cada dron puede aceptar o rechazar una misión según sus características y estado actual: los drones que reparten paquetes deberán asegurar que pueden cargar con el paquete y realizar el trayecto completo, ida y vuelta, con suficiente batería. Como a veces deben esperar sobrevolando en el lugar, siempre hay que considerar un margen extra de 10 minutos de autonomía para esa eventualidad. En el caso de los drones que filman, se deberá verificar que tengan batería suficiente para todo el recorrido, ida y vuelta, y además para que la cámara pueda filmar el tiempo requerido.

Tanto los drones como las cámaras deben saber encenderse y apagarse. Al encenderse un dron, debe mostrar su nivel de batería actual. Si se apaga un dron, su cámara también debe hacerlo. Implementar esto con mensajes simples por consola.

Otro desarrollador ya inició el proyecto (UML de clases y Java). **Tu tarea es continuarlo.**

A partir del modelo enunciado, se pide:

- 1) Continuar el diagrama de clases provisto con todas las entidades y relaciones encontradas, en formato **.uxf** (de UMLetino)
- 2) Implementar los métodos **elegirDron()** de la clase **Empresa**, que en todos los casos deben **retornar** (no imprimir por consola) el dron con mayor nivel de batería (suponiendo único) que pueda aceptar la misión encomendada. No se admiten bloques de código repetidos alevosamente.
- 3) Implementar el método **apagarDrones()** de la clase **Empresa**, que apague a todos los drones.
- 4) Desarrollar un test en el método **main** del proyecto que, creando las instancias e invocando los métodos que correspondan, permita comprobar los resultados de todas las operaciones de la clase **Empresa**.

FORMATO DE ENTREGA

Archivo **.zip** exportado desde NetBeans y nombrado de la siguiente manera (reemplazando lo que está en verde):

INSPT-ProgII-p1-2025-Drones-TUAPELLIDO-TUNOMBRE

El **.zip** debe contener el proyecto Java, el cual debe tener la misma nomenclatura y el diagrama de clases del modelo en formato **.uxf**.

No cumplir con alguna indicación del formato de entrega disminuye la nota del examen.

CRITERIOS DE EVALUACIÓN

Para considerar aprobado el examen, el mismo debe demostrar la correcta aplicación de los siguientes conceptos de la programación orientada a objetos y el lenguaje Java:

- Definición de clases y asignación adecuada de sus responsabilidades.
- Encapsulamiento, ocultando detalles de implementación y utilizando métodos getters y setters sólo cuando corresponda.
- Modularización reutilizable y mantenible, usando funciones con correcta parametrización, aplicando alta cohesión y bajo acoplamiento.
- Correcta implementación de los constructores.
- Validación de los datos que ingresan al sistema.
- Aplicación de herencia y polimorfismo, incluyendo interfaces.
- Aplicación de los principios S.O.L.I.D
- Algoritmos precisos, finitos y eficientes.
- Buenas prácticas de programación:
 - Uso de constantes en lugar de números mágicos.
 - Nombres de variables, métodos y clases descriptivos, que sigan la convención del lenguaje.
 - Uso adecuado de los ciclos. No alterar la naturaleza de los mismos utilizando, por ejemplo, instrucciones de control como **break** y **continue**.
 - Código ordenado e indentado correctamente.