

MODELO DE SEGUNDO EXAMEN PARCIAL

😊 Leé por lo menos dos veces el enunciado antes de resolver.

Enunciado

Te encomiendan el diseño de un sistema de control de calidad para una fábrica de productos, que exclusivamente pueden ser electrodomésticos, vehículos o juegos de mesa.

Todos los productos que produce son *evaluables*, por lo que podrán ser consultados por su estado de funcionamiento (BUENO, REGULAR, MALO). Para realizar dicha evaluación, el sector de producción selecciona y envía al sector de control de calidad uno o más productos de cada tipo donde se deberán evaluar primero los últimos seleccionados.

Los productos tienen un nombre y una descripción. Aquellos que son electrodomésticos además poseerán un indicador de uso. Cuando este nivel de uso esté entre 70 y 100, su funcionamiento será BUENO, si es menor a 70 pero mayor que 45 será REGULAR, y, si no, será MALO.

Cuando se trate de un juego de mesa poseerá la cantidad de jugadores y cartas disponibles, su calificación será BUENO si la cantidad de cartas es múltiplo de la cantidad de jugadores, sino será MALO (sin posibilidad de ser REGULAR).

Por último, si el producto se trata de un vehículo, será BUENO cuando su cantidad de kms sea menor a 10kms y será MALO si tiene más de 50km, siendo REGULAR en el resto de los casos.

Del enunciado descripto, se pide:

- A) El diagrama de clases UML que lo modelice con sus relaciones, atributos y métodos.
- B) El método **promedioKmsHastaMantenimiento** el cual debe retornar la cantidad promedio de kilómetros entre todos los vehículos que estén esperando para ser evaluados.
- C) El método **obtenerResultadosDeEvaluacion** el cual debe devolver a los productos que estaban para ser evaluados (dejando vacía esa colección) pero discriminados por estado de funcionamiento en diferentes colecciones, del mismo tipo que la original.

Criterios

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

- Correcta definición de clases y asignación adecuada de sus responsabilidades.
- Encapsulamiento, ocultamiento de información y uso de getters y setters sólo cuando corresponda.
- Modularización reutilizable y mantenible con uso de métodos con correcta parametrización.
- Correcta aplicación de miembros de instancia y de clase.
- Correcta aplicación de herencia y polimorfismo.
- Correcta aplicación conceptual de las relaciones entre clases.
- Correcto uso y manejo de todos los TADs vistos en la materia.