

## EXAMEN FINAL

08/07/2021

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

### Enunciado

Hace mucho tiempo, en una galaxia muy lejana... había muchos droides que formaban parte de la existencia cotidiana de todos.

Algo particular unía a todos esos droides: la capacidad que tenían de autorrepararse. Ante algún desperfecto, podían reemplazar la o las piezas que no estaban funcionando por otras (con el mismo nombre) para seguir operando.

Para esto, cada droide llevaba un registro detallado de las piezas que lo componían, manteniéndolas por separado entre piezas operativas y no operativas. Cada una de estas piezas tiene un nombre (**String**) que es el mismo en todos los droides (por ejemplo, "Batería de litio", "Sensor de proximidad", "Visor nocturno", etc).

Siempre que un droide encontraba a otro droide fuera de servicio (luego de batallas o simplemente por el uso) lo registraba para así, de ser necesario, usar las piezas sanas de éste para repararse. Al necesitar una pieza buscaba entre los droides rotos que tenía registrados, chequeando si encontraba entre ellos las piezas sanas que necesitaba. De encontrarlas, reemplazaba sus piezas no operativas por las operativas encontradas en los otros droides.

### Se pide:

- Confeccionar el diagrama UML que describe el escenario del enunciado incluyendo los atributos de cada clase y los métodos a desarrollar y aquellos que creas conveniente.
- Desarrollar el método **autoRepararse()** de la clase Droide, que no recibe parámetros. Debe intentar reemplazar sus piezas no operativas por las piezas operativas que pudiera encontrar en alguno de los otros droides. Cada vez que una pieza se reemplaza la pieza no operativa original se descarta. Este método devuelve alguno de estos resultados:
  - COMPLETAMENTE\_OPERATIVO: cuando todas las piezas del droide están operativas.
  - REPARACION\_PARCIAL: cuando quedan algunas piezas no operativas, pero algunas se pudieron reparar.
  - REPARACION\_IMPOSIBLE: cuando no se logre reparar ni siquiera una de las piezas no operativas que pudiera tener.

### Criterios

Para considerar aprobado el examen, el mismo debe resolver lo pedido y aplicar los siguientes conceptos de la programación orientada a objetos:

**EXAMEN FINAL**

**08/07/2021**

- Detección de clases, atributos, métodos y relaciones (asociativas y de uso).
- Modularización reutilizable y mantenible usando métodos con correcta parametrización.
- Asignación de responsabilidades a cada clase y correcto encapsulamiento.
- Manejo del concepto de instancia y de la interacción entre objetos.
- Manipulación de listas de objetos (ArrayList) y su uso en ciclos condicionales y for-each, eligiendo siempre el más adecuado en cada situación.
- Manejo de diagramas Nassi-Schneiderman y UML de clases.

|               |        |         |         |    |         |    |    |         |    |     |
|---------------|--------|---------|---------|----|---------|----|----|---------|----|-----|
| %<br>Correcto | 0 a 20 | 25 a 45 | 50 a 55 | 60 | 65 a 70 | 75 | 80 | 85 a 90 | 95 | 100 |
| Nota          | 1      | 2       | ¿4?     | 4  | 5       | 6  | 7  | 8       | 9  | 10  |