

## Diseño y Programación Orientados a Objetos 1er. Semestre 2024

### Certamen II ELO-329 Diseño y Programación Orientada a Objetos (1s2024)

En este certamen usted no podrá hacer preguntas. Si algo no está claro, indíquelo en su respuesta, haga una suposición razonable y resuelva conforme a ella.

Segunda parte, **con apuntes**.

#### Tercera pregunta (35 pts.)

Desarrolle una simulación de un robot de limpieza virtual mediante una interfaz gráfica utilizando C++ junto con las bibliotecas Qt. El robot tendrá varios estados y su indicador cambiará con el paso del tiempo y la interacción del usuario. A continuación puede ver un ejemplo de la interfaz.



Figura 1: Ejemplo de visualización de la interfaz gráfica.

El robot también puede estar en diferentes estados. Estos estados están definidos como un dato tipo **enum** en los códigos de ayuda. A continuación se presentan la lista de estados posibles, así como sus condiciones específicas:

Estados del Robot(**enum RobotState**)

- **INACTIVO**: Estado por omisión del robot. El robot se encuentra **INACTIVO** cuando no está limpiando o su nivel de suciedad ha llegado al tope de 100 puntos.
- **ACTIVO**: El robot está limpiando y el nivel de suciedad es menor a 100 puntos.

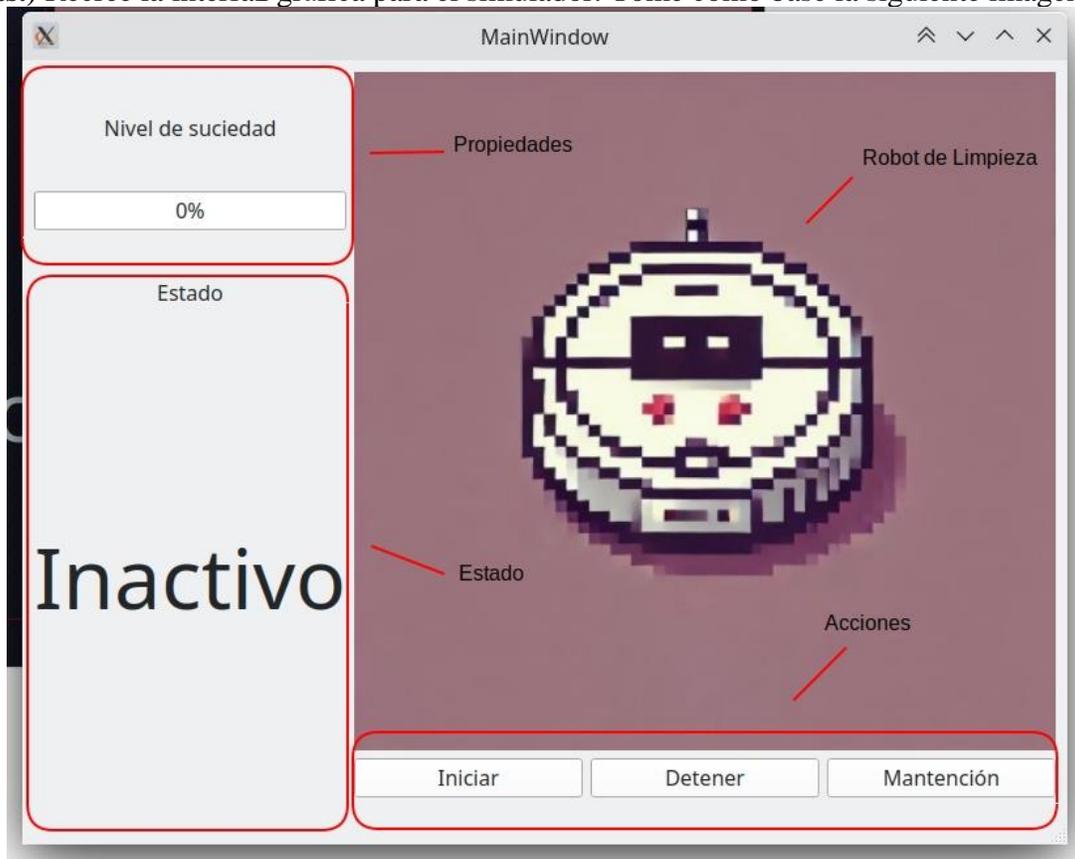
Además, considere el efecto de los siguientes botones:

- **Iniciar**, el cual cambia el estado del robot a **ACTIVO**. En este estado el nivel de suciedad aumenta en 20 puntos por cada ciclo de limpieza.
- **Detener**, el cual cambia el estado del robot a **INACTIVO**. El nivel de suciedad no cambia mientras el robot se mantenga en este estado.
- **Mantención**, el cual establece el nivel de suciedad recogida a 0 puntos (equivalente a limpiar el robot). Presionar este botón no cambia el estado en que se encuentra el robot.
- Considere el siguiente código con los estados del robot para su desarrollo:

### RobotState.h

#### Pregunta 3.1

(10 pts.) Recree la interfaz gráfica para el simulador. Tome como base la siguiente imagen.



Para la imagen del robot, utilice aquella disponible en el siguiente [enlace](#).

#### Pregunta 3.2

(10 pts.) Cree el código correspondiente a la clase **Robot**, incluyendo los siguientes atributos:

- **int nivel\_suciedad** (0 a 100), el cual representa el nivel de suciedad recogida por el robot. El robot debe iniciar con **nivel\_suciedad = 0**.
- **RobotState estado**, dato **enum** que representa los diferentes estados del robot. El robot debe iniciar con estado **INACTIVO**.

Para implementar la funcionalidad del robot, incluya los siguientes métodos/slots según requiera:

- **Constructor**: Inicializa los atributos del robot nivel de suciedad y estado inicial según lo indicado anteriormente.
- **limpiar()**: Este método debe cambiar el estado del robot a **ACTIVO** en caso de estar **INACTIVO**. Mientras el nivel de suciedad sea menor que 100, aumenta el nivel de suciedad en 20 puntos por cada llamada.
- **detener()**: Este método debe cambiar el estado del robot a **INACTIVO** en caso de estar **ACTIVO**. El robot debe detener su funcionamiento y no debe alterar su nivel de suciedad.
- **mantencion()**: Este método debe establecer el nivel de suciedad recogida a 0 puntos. Use este método para simular el mantenimiento del robot (limpieza) vaciando su contenedor de suciedad.

### Pregunta 3.3

(10 pts.) Implemente un mecanismo que simule el paso del tiempo utilizando la clase [QTimer](#) vista en clases. Configure el elemento **QTimer** para que el atributo **nivel\_suciedad** aumente en 20 puntos cada 1 segundo mientras el estado del robot es **ACTIVO**.

### Pregunta 3.4

(5 pts.) Conectar los diferentes elementos de la interfaz con sus respectivos elementos de la clase Robot.

Esto incluye el funcionamiento de los botones **Iniciar**, **Detener** y **Mantención** sobre los atributos del robot.

Asegúrese de que el indicador de nivel de suciedad, así como el estado del robot, se actualicen automáticamente en la interfaz gráfica. Puede agregar todos los métodos adicionales que considere necesarios para su desarrollo.

Una solución para esta pregunta está [aquí](#).

### Asignación de puntajes:

- 10 puntos: la interfaz gráfica muestra una vista similar a la imagen dada.
- 10 puntos: La clase Robot incluye atributos pedidos (nivel de suciedad y estado) e incluye los métodos pedidos (Constructor, limpiar, detener, mantención).
- 10 puntos: El programa implementa el paso del tiempo usando clase QTimer aumentando suciedad en 20 puntos/s mientras el robot está ACTIVO. El nivel de suciedad se refleja en la interfaz.
- 5 puntos: El programa conecta los botones de la interfaz con elementos de la clase Robot para reaccionar ante los botones Iniciar, Detener y Mantención. El estado se actualiza automáticamente.