

Tarea 3: Simulador Gráfico en C++ de Lámpara Domótica

Lea detenidamente la tarea. **Si algo no lo entiende, consulte. Si es preciso, se incorporarán aclaraciones en otro color.**

1.- Objetivos

- Ejercitar la configuración de un ambiente de trabajo para desarrollar aplicaciones en lenguaje C++, se pide trabajar con un IDE.
- Reconocer clases y relaciones entre ellas en lenguaje C++.
- Diseñar y programar clases en C++.
- Ejercitar la preparación y entrega de resultados de software (creación de makefiles, readme, documentación, manejo de repositorio GIT).
- Desarrollar aplicaciones gráficas usando Qt.
- Uso de metodología de desarrollo software "iterativa" e "incremental".

1.- Descripción General

Esta tarea busca programar en C++ algunas funcionalidades de las tareas 1 y 2. La lógica de la simulación es esencialmente la misma de las tareas previas. Por completitud, se detalla nuevamente el modelo y la programación de un tipo de lámparas.

Modelo de una lámpara controlada

La lámpara de esta tarea permite ser encendida y su luz cambia de intensidad un control remoto, Figura 2. La luz se compone de tres colores básicos rojo, verde y azul, los cuales el control de intensidad único los ajusta al mismo tiempo para generar distintas intensidades de luz blanca. Cada color básico es representado por un valor que varía de 0 a 255. Cuando una lámpara es instalada ("nace"), parte apagada y con intensidades blanca máxima (255 para cada color).

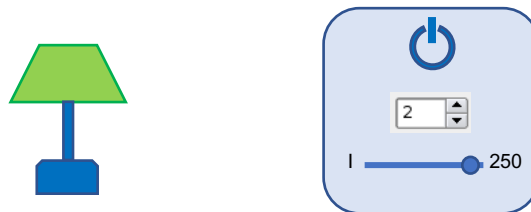


Figura 1: Modelo de una lámpara domótica y su control remoto

Cada lámpara tiene un canal asociado. Un control de lámpara controlará todas las lámparas de ese canal. Cada control permite cambiar su canal y así con un control manejar múltiples lámparas (use un QSpinBox). En Figura 1, se aprecia cómo bajo el botón de encendido/apagado se muestra el canal actual el cual se puede aumentar o reducir con los controles laterales.

Al encender una lámpara (botón "power" o tecla espacio)¹, ésta ilumina con la intensidad del valor correspondiente al control de intensidad. Se apaga con mismo botón ("power" o tecla espacio).

Para cambiar el nivel de intensidad, se usa el control mostrado (QSlider). No existen intensidades inferiores a 0 o superiores a 255.

"La Nube"

La nube no corresponde a un objeto físico sino a una aplicación corriendo en algún servidor remoto. Cada aparato controlable domóticamente se registra en la nube y puede ser controlado por el control de ese canal. Así, para cambiar un dispositivo domótico, un control envía mensajes a la nube y ésta los envía a cada dispositivo domótico.

¹ Cuando los controles tienen botón único de encendido y apagado, éste funciona conmutando el estado actual.

Interfaz Gráfica

En esta tarea su programa debe generar una interfaz, la cual usted puede crear con el modo "Design de Qt Creator, similar a lo mostrado en Figura 2. No se pide dar flexibilidad al número de elementos que la componen.

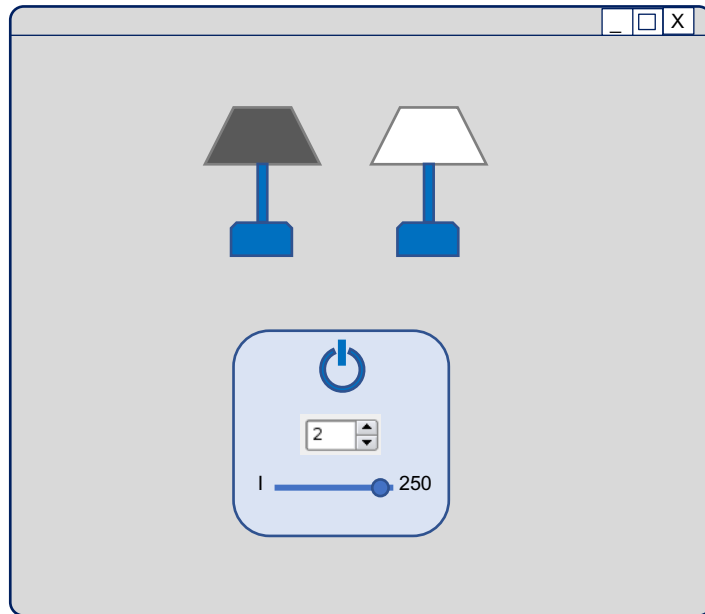


Figura 2: Apariencia pedida para la interfaz gráfica de usuario

En su solución trate que el programa emule lo que ocurre en la realidad.

2.- Desarrollo en Etapas

Para llegar al resultado final de esta tarea usted debe aplicar una metodología de tipo "Iterativa e Incremental" para desarrollo de software. Usted y su equipo irán desarrollando etapas donde los requerimientos del sistema final son abordados gradualmente. En cada etapa usted y su equipo obtendrá una solución que funciona para un subconjunto de los requerimientos finales o bien es un avance hacia ellos. En GitLab organice el código para cada etapa del desarrollo. **Su equipo deberá preparar una solución para cada una de las etapas** aun cuando la última integre las primeras. **El readme y archivo de documentación deben ser preparados solo para la última etapa.** Esto tiene por finalidad, educar en la metodología iterativa e incremental.

2.1.- Primera Etapa: Una lámpara con encendido On/Off

En esta etapa se pide crear una interfaz gráfica conteniendo sólo una lámpara y un control remoto que contendrá sólo el botón On/Off.

El archivo stage1.cpp contiene el método main.

Haga uso de Qt Creator y el modo Design para crear los elementos gráficos.

2.2.- Segunda Etapa: Una lámpara con encendido On/Off y control de Intensidad

Agregue un QSlider para cambiar la intensidad de lámpara de la etapa previa.

El archivo stage2.cpp contiene el método main.

Haga uso de Qt Creator y el modo Design para completar los elementos gráficos.

2.3.- Tercera Etapa: Dos lámparas y su control con funcionalidad completa

Para esta etapa se pide completar las opciones del control de lámparas y ubicar dos lámparas en canales 2 y 3. Debe agregar un QSpinBox para cambiar el canal del control. El control remoto debe permitir cambiar la intensidad de cada lámpara

El archivo stage3.cpp contiene el método main.

Haga uso de Qt Creator y el modo Design para completar los elementos gráficos.

2.4.- Extra crédito: (+10 puntos) Incorpore un menú en la parte superior para cambiar el control remoto a uno con 3 QSlider de manera de poder cambiar el color de cada lámpara.

3.- Elementos a considerar en su documentación

Entregue todo lo indicado en "[Normas de Entrega de Tareas](#)", excepto makefile.

Para su entrega deje en GitLab los archivos de su proyecto Qt. Se espera que usted omita archivos no esenciales. Para eso haga uso de archivo de configuración ".gitignore".

En su archivo de documentación (pdf o html) incorpore el diagrama de clases de la aplicación (etapa 4).

4.- Sobre la entrega de la tarea

El formato de la entrega seguirá la siguiente norma:

- La sección de entrega de la tarea en AULA será exclusivamente para agregar el link al repositorio Git donde esté alojada la solución de la tarea.
- La entrega mencionada anteriormente sigue estando regida por los plazos establecido en la tarea.
- No olvidar que la tarea debe ser subida al repositorio Git respetando estos mismos plazos.
- Cada grupo debe incorporar a su ayudante con el rol "reporter" a su proyecto Git
- La no entrega a tiempo, implica un descuento diario.