línea horizontal

**Informa2 S.A.S.**

16 de septiembre de 2021

# **Presentación**

Un nuevo trabajo ha sido adjudicado a la empresa Informa2, en esta ocasión ustedes han sido contratados para diseñar un sistema el cual permita presentar en una pantalla con leds RGB la nacionalidad de los competidores que han llegado al podio de triunfadores para los juegos olímpicos de París 2024. Esta iniciativa surge como una forma de actualizar la forma tradicional de presentar las nacionalidades de los ganadores como lo mostrado en la Figuras 1 y 2, en donde se presenta el país de origen de los competidores con textiles. La idea es realizar una implementación tecnológica como la vista en las carreras de Fórmula 1 (Figura 3), en donde se usan pantallas gigantes para la ceremonia de premiación con el propósito anteriormente mencionado, pero también para presentar publicidad e información relevante del evento.

# **Objetivos**

1. Desarrollar la capacidad de solución de problemas en los estudiantes enfrentándolos a problemáticas de la vida cotidiana.
2. Identificar si el estudiante adquiere las destrezas y conocimientos fundamentales de la programación con C++, en donde resaltamos la programación orientada a objetos.
3. Evaluar la capacidad del estudiante para trabajar con Arduino e integrar la programación en C++, además de usar de manera adecuada las funciones de la plataforma explicadas en clase que permiten controlar el puerto serial y los puertos digitales.



**Figura 1.** Presentación de las nacionalidades del podio de triunfadores usando banderas con materiales convencionales.



**Figura 2.** Podio de triunfadores prueba BMX Tokio 2020 en donde la corredora Colombiana Mariana Pajón ganó su tercera medalla olímpica.



Figura 3. Podio de triunfadores del gran premio de Azerbaiyán 2021.

# **Desafío**

En esta ocasión su tarea consiste en recibir una imagen en formato jpg y realizar el procesamiento de la información contenida en esta, de tal forma que se haga un ajuste de sus dimensiones, para que pueda ser presentada en una matriz de LEDs RGB.

|  |  |
| --- | --- |
|  |  |

Figura 4. Ejemplos de representación.

En la Figura 4 se observan ejemplos donde se representan las banderas de Japón y Colombia en una matriz de 8x8 LEDs.

Para esto le proporcionamos programas de ejemplo que muestran cómo tener acceso a la información de una imagen (lectura y escritura). Analice muy bien el programa *ImageRead* e *ImageWrite* desarrollado en Qt para que pueda entender como es el proceso de cargar una imagen desde su directorio de trabajo y cómo generarla.

De igual forma es importante que estudie las características principales del espacio de color RGB, puede apoyarse del famoso editor de imágenes *Paint,* cargando en él una imagen de prueba y pasando el cursor sobre esta para que identifique la composición en el espacio de color mencionado de cada pixel. Para complementar, puede cargar, usando los programas compartidos (de Qt), dicha imagen y comparar lo analizado en *Paint* con los resultados que presenta su programa en la consola.

Usted deberá desarrollar un algoritmo para que una vez se cargue la imagen, se exporte la información en las dimensiones de la matriz de LEDs disponible, como se muestra en los ejemplos de la Figura 4. La única restricción para la matriz de LEDs es que debe ser al menos de 8x8, pero usted es libre de poner cuantos LEDs considere por arriba de ese tamaño. Para representar la matriz de LEDs, deberá montar el circuito correspondiente usando TINKERCAD. Para la simulación, puede utilizar LEDs RGB, NeoPixel o tiras de NeoPixel. Además deberá implementar el controlador de la matriz de LEDs usando la plataforma Arduino. Como base en el manejo de los dispositivos mencionados, se les dará una instrucción con demostración de funcionamiento en la plataforma Arduino.

Tenga en cuenta que el usuario indicará la ubicación del archivo de imagen, nada más. Las dimensiones de la imagen serán arbitrarias, no se podrá indicar dimensiones específicas, el programa debe estar en capacidad de manipular la imagen sin importar sus dimensiones. La salida de este programa será un archivo txt que contendrá el segmento de código que debe ser agregado en el controlador de la matriz de LEDs de la plataforma Arduino, de tal forma que al agregar dicho código y simular, en la matriz de LEDs, se vea representada la imagen indicada por el usuario.

Una posible configuración puede ser tener una matriz de LEDs de 50x100 y que el usuario ingrese una imagen de 640x480, en tal caso el programa deberá submuestrear. También podría pasar que la imagen de entrada sea de 10x10, en este caso el programa deberá sobremuestrear.

**Requisitos mínimos**

A continuación se describen los requisitos que se deben cumplir. El incumplimiento de cualquiera de ellos implica que su nota sea cero.

1. La solución debe ser implementada en lenguaje C++.
2. El programa debe ser una aplicación desarrollada en las plataformas Qt y TINKERCAD.
3. El submuestreo o sobremuestreo de las imágenes debe ser realizado con código escrito por ustedes sin el apoyo de ninguna librería de procesamiento de imágenes.
4. Se debe crear un repositorio público en el cual se van a poder cargar todos los archivos relacionados a la solución planteada por usted (informe, código fuente, enlace al proyecto de TINKERCAD y otros anexos) de forma ordenada y por carpetas para cada requisito. El proyecto de TINKERCAD debe ser público y el enlace que se adjunte al repositorio debe ser el que se muestra al dar click en el ícono del corazón en el panel de proyectos.
5. Una vez cumplida la fecha de entrega no se podrá hacer modificación alguna al proyecto de TINKERCAD.
6. Una vez se cumpla el plazo de entrega, no se deberá hacer modificación alguna al repositorio, en caso contrario su nota es cero.
7. Se deben hacer commits de forma regular (al menos dos al día) de tal forma que se evidencie la evolución de la propuesta de solución y su implementación.
8. Se debe adjuntar un enlace de youtube a un video que debe incluir lo siguiente:

* Presentación de la solución planteada. Análisis realizado y explicación de la arquitectura del sistema (4 minutos máximo).
* Demostración de funcionamiento del sistema. Explicar cómo funciona en tiempo de simulación (4 minutos máximo).
* Explicación del código fuente. Tenga en cuenta que debe realizar una justificación de las tareas de su algoritmo, por ejemplo, describa de manera general las clases implementadas (atributos y métodos), bloques de código relevantes (bucles de inicialización, bloques condicionales, etc.), funciones. (7 minutos máximo).
* La duración total del video no debe exceder 15 minutos.
* Asegúrese que el video tiene buen sonido y que se puede visualizar con resolución suficiente para apreciar bien los componentes presentados.

1. El plazo de entrega es el día 21 de septiembre a las 20:00 para el informe de análisis y diseño.
2. El plazo de entrega es el día 28 de septiembre a las 20:00 para el informe de implementación.
3. A la tarea Parcial 2 - Informa2 del Classroom del curso, debe adjuntar dos enlaces: uno al repositorio y otro al video, nada más.

# **Especificaciones**

1. **[15%]** Genere un informe en Overleaf donde se detalle el análisis y diseño de la solución planteada para el desafío, explique entre otras cosas:
   1. Análisis del problema y consideraciones para la alternativa de solución propuesta. Incluya las relaciones entre las clases implementadas para el desarrollo del algoritmo.
   2. Esquema donde describa las tareas que usted definió en el desarrollo del algoritmo.
   3. Algoritmo diseñado, no código. Tenga en cuenta que el algoritmo no depende del lenguaje de programación.
   4. Consideraciones a tener en cuenta en la implementación.
2. **[15%]** Genere un informe en Overleaf donde se detalle la implementación de la solución planteada para el desafío, explique entre otras cosas:
   1. Las clases implementadas.
   2. Esquema donde describa la estructura final de las clases implementadas.
   3. Módulos de código implementado donde se evidencie la interacción entre las diferentes clases.
   4. Estructura del circuito montado.
   5. Problemas presentados durante la implementación.
3. **[10%]** Escriba un manual de uso corto que le permita a cualquier usuario poder usar su sistema. Tenga en cuenta que se leerá paso a paso el manual de acuerdo a sus instrucciones para la ejecución.
4. **[35%]** Implementación y funcionamiento adecuado (completamente funcional) del programa realizado en Qt. Tenga en cuenta que debe generar un archivo de salida con extensión .txt el cual al ser abierto con un editor de texto, permitirá copiar todo su contenido y pegarlo en el simulador Tinkercad para la representación de la imagen.
5. **[25%]** Implementación y funcionamiento adecuado (completamente funcional) del circuito en Tinkercad de la imagen asignada en la sustentación.