**INFORME DE DESARROLLO DE PROYECTO**

**Parcial I – Informática II**

Elaborado por Sergio Andrés Berbesí Builes y Salomé Bermudez Macias

A continuación, se presenta el desarrollo del proyecto para la elaboración de un sistema que comprende el montaje de un circuito en plataforma Arduino y la codificación en lenguaje C++ para gestionar diferentes opciones relacionadas a un display de LED´s que un usuario final va a usar e interactuar.

**PRIMERA PARTE: Análisis del problema y consideraciones para la alternativa de la solución propuesta.**

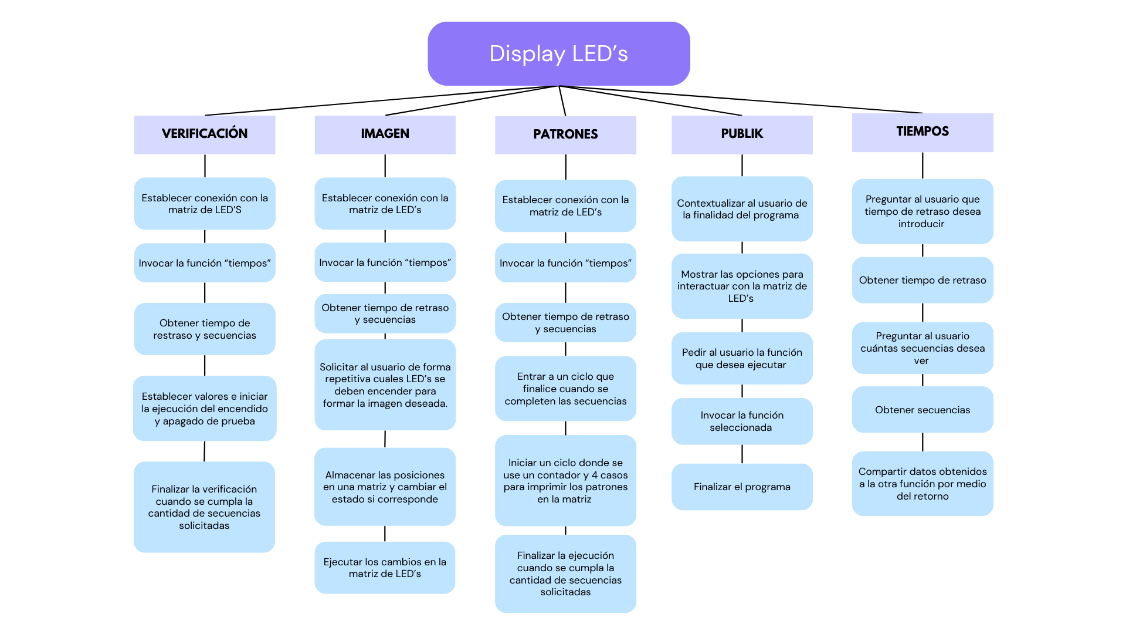
**Problema:**

Generar un programa donde el usuario pueda seleccionar entre diferentes opciones para interactuar con un display compuesto por 64 LED’s, ubicados en un cuadrado de 8 filas y 8 columnas. El programa debe ofrecer un menú con 3 opciones principales: verificación del funcionamiento del display, mostrar una imagen de prueba y, mostrar de forma alternada 4 patrones prediseñados.

**Consideraciones para la solución:**

1. Generar 3 funciones para interactuar con el display de LED’s.
2. Crear una función principal para que el usuario pueda seleccionar qué opción desea visualizar.
3. Adicionar una función que en cualquiera de los casos que se puedan ofrecer, le solicite al usuario el tiempo de retraso entre el encendido y apagado del display o la imagen que se esté mostrando, al igual que, la cantidad de secuencias a mostrar.

**SEGUNDA PARTE: Esquema de tareas para algoritmos.**

****

**TERCERA PARTE: Algoritmos implementados.**

A continuación, se presenta un modelo del programa diseñado por medio de algoritmos en pseudocódigo.

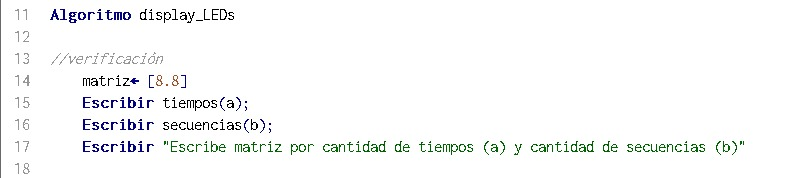
****

Imagen 1 – Función verificación.

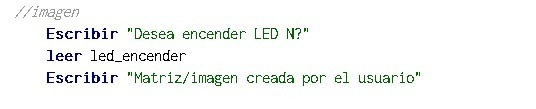


Imagen 2 – Función imagen.

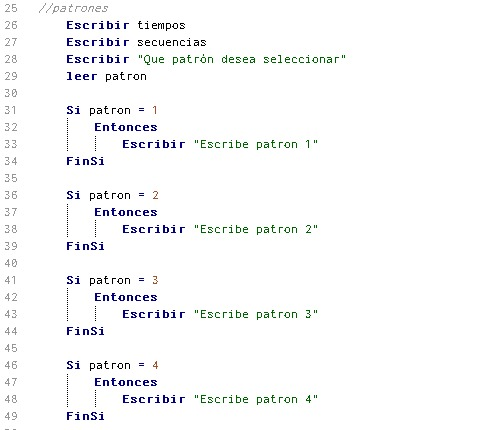


Imagen 3 – Función patrones.

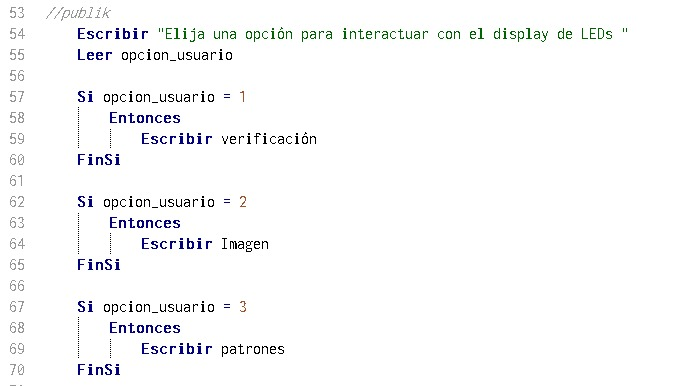


Imagen 4 – Función publik.

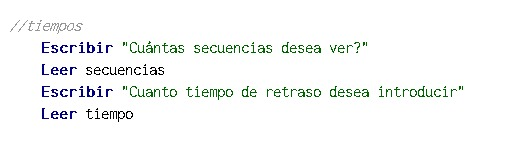


Imagen 5 – Función tiempos.

**CUARTA PARTE: Problemas presentados en el desarrollo.**

A lo largo de la planeación relacionada a la implementación de la propuesta para crear el programa solicitado, se evidencian puntos específicos donde se tuvo que realizar nuevas consideraciones y planteamientos para evitar inconvenientes, como:

1. El hecho de preguntar y solicitar al usuario repetidas veces la misma información sobre el tiempo de retraso y la cantidad de secuencias, implica que se reduce la eficiencia del programa, sin importar la opción que el usuario elija; además, implica que el programa va a tener partes de la codificación con algo grado de similitud, de forma innecesaria. Por lo tanto, se decide implementar una quinta función, llamada “tiempos”, la cual, solicita al usuario esta información, que se requiere en cualquiera de las opciones que hay disponibles para ejecutar.
2. Al realizar el uso de la función “verificación”, si esta se implementa por medio de un ciclo donde se cambia el estado de cada LED paso a paso, implica llegar a un resultado de forma menos objetiva y eficaz, a comparación del establecimiento de condiciones en un ciclo anidado. Adicionalmente, resulta más breve realizar los cambios de estados de los LED´s por medio del ciclo for, donde solamente se le van a asignar a las variables los datos solicitados al usuario en la función “tiempos” y sería posible darle uso a los punteros, de manera que, en el ciclo anidado se van a modificar los estados de cada LED (almacenados independientemente como variables) al ir pasando por las posiciones de memoria e ir sumando el valor de los bits que tiene el tipo de dato a usar para llegar a la posición siguiente (n+1).
3. En la función imagen se presenta el problema o cuestionamiento sobre cómo solicitar al usuario una imagen para probar el display de LED´s, por lo cual, surge la alternativa de solicitar al usuario repetidas veces si desea que el LED de la posición actual va a estar encendido o no, mientras se podría almacenar esa información en arreglos dinámicos, para luego poder acceder con mayor facilidad a los estados de los LED´s, sin importar si el usuario decide solo usar k filas y j columnas de la matriz disponible.
4. Respecto a la función “patrones”, surge el problema de plantear soluciones que no sean triviales, es decir, que no impliquen encender ciertos LED´s de forma arbitraria, sino que, se encuentran los algoritmos utilizados para generar las figuras requeridas.

**QUINTA PARTE: Evolución de la solución propuesta y consideraciones a tener en cuenta para la implementación.**

La solución inicial estuvo encaminada de la misma manera en que hasta ahora lo hace, sin importar que se ha tomado en cuenta los diferentes problemas que surgen ante las condiciones naturales de la correctitud del programa y las adicionadas para este proyecto. El desarrollo de la solución se estima que será satisfactorio, puesto que se están considerando diferentes factores relevantes, al igual que, la implementación representa el éxito del análisis realizado para que el programa cumpla con los requisitos mínimos a satisfacción, ofrezca un menú amigable y que cumpla las necesidades del usuario final sin presentar fallas o respuestas no correspondientes.

En general, la solución propuesta se trabaja por medio de las cinco funciones mencionadas anteriormente, de las cuales, se debe tener en consideración que, tres de ellas van a ser llamadas solo cuando en la función principal el usuario decida que opción es la que busca. Igualmente, la función restante, se debe invocar siempre que se use cualquiera de las funciones ofrecidas en el menú de opciones.

Hasta la fecha de elaboración de este informe, la evolución de la solución propuesta abarca el análisis del problema, implementación de las tareas en un algoritmo base (a modo de esqueleto del programa final), y el inicio de la codificación de las funciones básicas.

Finalmente, se toma como punto a considerar, la elaboración de un montaje físico que corresponda al mismo que se realiza de forma virtual para simular de forma más intuitiva los comportamientos del programa, y que el usuario pueda interactuar de manera más agradable ante el llamado de atención que genera la implementación física del sistema (Arduino y codificación en C++).