Universidad de Antioquia



Parcial 1

Jhoan Sneyder Portillo Anaya

Informática II

UdeA – Facultad de Ingeniería

Tabla de contenido

[A. Análisis del Problema y Consideraciones para la Alternativa de Solución Propuesta. 3](#_Toc146398132)

[Descripción del Problema 3](#_Toc146398133)

[Consideraciones Iniciales 3](#_Toc146398134)

[Solución propuesta 3](#_Toc146398135)

[B. Esquema donde describa las tareas que usted definió en el desarrollo de los algoritmos. 4](#_Toc146398136)

[Tarea 1: Análisis del problema 4](#_Toc146398137)

[Tarea 2: Diseño de algoritmos 4](#_Toc146398138)

[Tarea 3: Implementación y prueba de los algoritmos 5](#_Toc146398139)

[Tarea 4: Documentación 5](#_Toc146398140)

[C. Problemas de desarrollo que afrontó 5](#_Toc146398141)

[Problemas de desarrollo con el integrado 74HC595 5](#_Toc146398142)

[D. Evolución de la solución y consideraciones a tener en cuenta en la implementación. 6](#_Toc146398143)

[E. Algoritmos implementados 7](#_Toc146398144)

[Función verificación 7](#_Toc146398145)

[Función patrones 7](#_Toc146398146)

[Función imagen 8](#_Toc146398147)

# Análisis del Problema y Consideraciones para la Alternativa de Solución Propuesta.

## Descripción del Problema

El problema consiste en conectar y controlar un sistema compuesto por 64 LEDs de manera eficiente utilizando una placa Arduino. Estos LEDs deben organizarse en una matriz 8x8 y ser controlados individualmente para poder encenderlos y apagarlos de acuerdo con un patrón específico o una secuencia definida.

## Consideraciones Iniciales

**Limitaciones de Pines de Salida:** Una placa Arduino Uno tiene un número limitado de pines de salida digital (normalmente 14 pines). Conectar directamente 64 LEDs a estos pines sería ineficiente y poco práctico.

**Multiplexación:** Para superar la limitación de pines, se debe utilizar una técnica de multiplexación o expansión de puertos. En este caso, se ha optado por utilizar el integrado 74HC595, que es un registro de desplazamiento de salida serial en paralelo.

**Organización de los LEDs:** Los 64 LEDs deben organizarse en una matriz de 8 filas por 8 columnas. Esto implica tener un control independiente sobre cada fila y columna para poder seleccionar un LED específico en la matriz.

**Limitación de Corriente:** Para evitar dañar los LEDs, es necesario limitar la corriente que fluye a través de cada uno de ellos. Esto se logra mediante la conexión de resistencias en serie con cada LED.

## Solución propuesta

La alternativa de solución propuesta es utilizar dos integrados 74HC595 en cascada para controlar los 64 LEDs organizados en una matriz 8x8. Esta solución se basa en las siguientes consideraciones:

**Expansión de Pines:** Los 74HC595 permiten controlar múltiples LEDs utilizando solo unos pocos pines de salida digital de la placa Arduino. Al conectar dos de estos integrados en cascada, se pueden controlar 64 LEDs utilizando solo 3 pines de salida digital.

**Multiplexación de Filas y Columnas:** Para organizar los LEDs en una matriz 8x8, se utilizará una técnica de multiplexación. Los 74HC595 controlarán las columnas de LEDs, mientras que las filas se seleccionarán individualmente mediante salidas digitales adicionales.

**Limitación de Corriente:** Se colocarán resistencias en serie con cada LED para limitar la corriente y evitar daños.

**Programación de Arduino:** Se desarrollará un programa en Arduino que envíe los datos adecuados a los registros de desplazamiento de los 74HC595 para encender y apagar los LEDs de acuerdo con un patrón definido por el usuario.

# Esquema donde describa las tareas que usted definió en el desarrollo de los algoritmos.

## Tarea 1: Análisis del problema

En esta tarea se identifica el problema a resolver y sus características. En este problema, las características del problema son las siguientes:

* Hay que conectar 64 LEDs a una placa Arduino.
* Los LEDs deben estar conectados en una estructura lineal.
* Los LEDs deben estar controlados por el integrado 74HC595.

## Tarea 2: Diseño de algoritmos

En esta tarea se diseña los algoritmos que resolverán el problema. En el caso de este problema, los algoritmos necesarios son los siguientes:

* Diseñar algoritmos para controlar los LEDs de forma individual o en grupos.
* Diseñar algoritmos para crear efectos con los LEDs.

## Tarea 3: Implementación y prueba de los algoritmos

En esta tarea se debe implementar los algoritmos en un lenguaje de programación. En este caso el lenguaje de programación es en C++ implementado en el Arduino. Así mismo también se debe de probar los algoritmos conectando los LEDs a una placa Arduino y cargando el código en la placa.

## Tarea 4: Documentación

* Documentar el código de manera adecuada, incluyendo comentarios descriptivos.
* Preparar una documentación de usuario que explique cómo usar y personalizar los patrones.
* Realizar un vídeo en el que se explique de manera detallada la solución planteada, el funcionamiento del sistema y la explicación del código fuente.

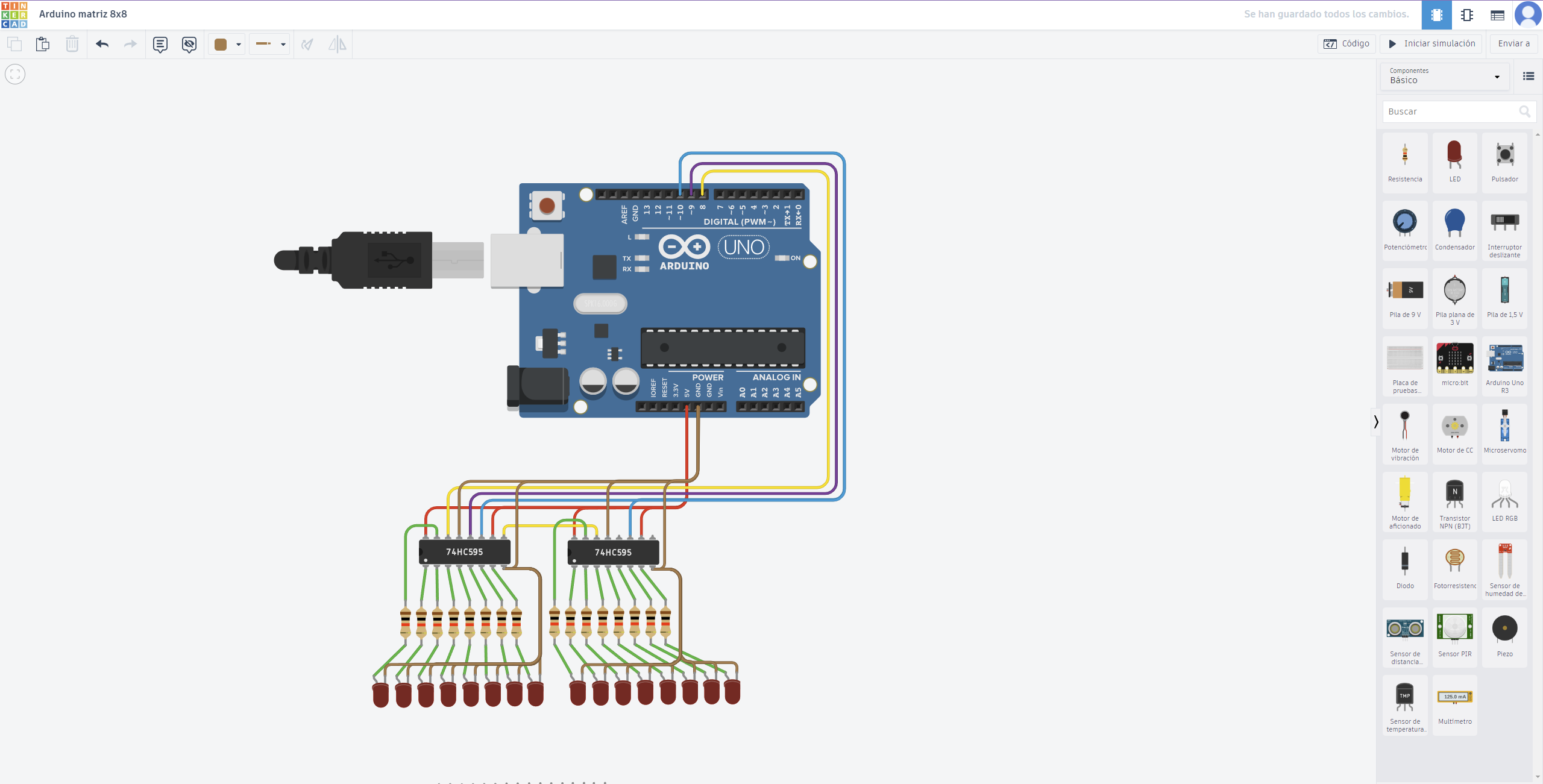
# Problemas de desarrollo que afrontó

## Problemas de desarrollo con el integrado 74HC595

Durante la fase inicial de desarrollo del proyecto, me enfrenté a un desafío significativo relacionado con la implementación del integrado 74HC595. Carecía de experiencia previa con este componente, lo que resultó en una curva de aprendizaje inicial.

Uno de los principales obstáculos que enfrenté fue la falta de comprensión sobre cómo conectar y controlar eficazmente el 74HC595 en el circuito. Dado que la eficiencia en el uso de pines de salida era esencial para el proyecto, la incapacidad de aprovechar este integrado de manera adecuada planteó interrogantes fundamentales en cuanto a la viabilidad del diseño.

Tras una ardua investigación leyendo documentación, viendo tutoriales y siguiendo estos mismos, llegué al siguiente sistema:



Este sistema lo que busca es implementar el integrado 74HC595, utilicé un código para poder encender todos los LEDs, éste funcionaba sin errores, pero los LEDs no se encendían. Esto podía ser causado por dos factores. El código estaba mal implementado en el sistema, o el sistema estaba mal conectado.

# Evolución de la solución y consideraciones a tener en cuenta en la implementación.

Verificando la conexión del sistema me percaté que estaba mal, por lo tanto, siguiendo otras documentaciones y tutoriales pude realizar las conexiones de manera adecuada para poder encender todos los leds.

Imagen que contiene Diagrama

Descripción generada automáticamente

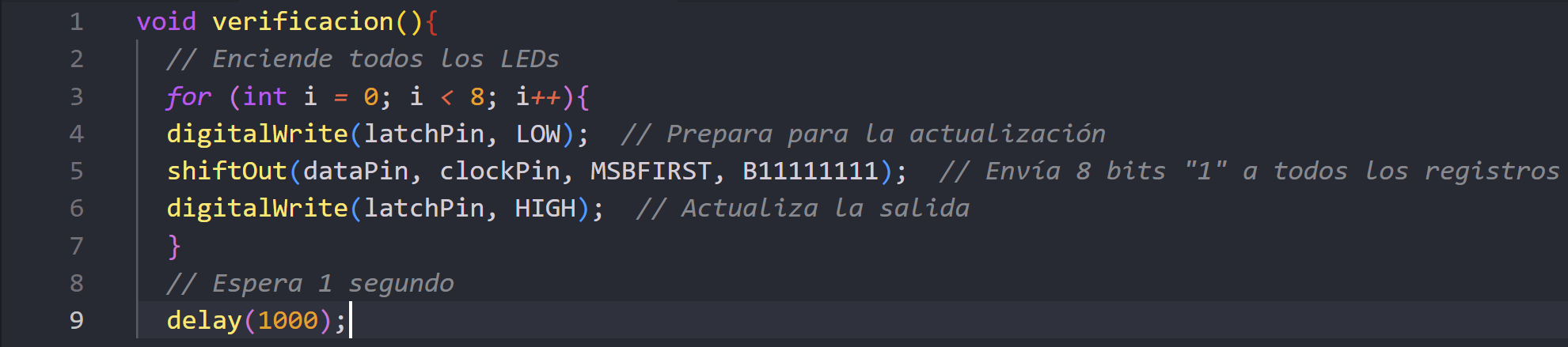
Este ya es el sistema completo con los 64 LEDs (matriz 8x8).

Se utilizaron 8 integrados 74HC595 conectados a 3 pines digitales del Arduino. También se utilizaron 8 resistencias las cuales la terminal 1 va conectado a los integrados y la terminal 2 va conectado a los 8 LEDs correspondientes; estos a su vez van conectados a las 8 salidas de cada integrado.

# Algoritmos implementados

A continuación, estas son las funciones que implementé en la solución del problema.

## Función verificación



Esta función comprueba que todos los LEDs de la matriz 8x8 están funcionando correctamente.

## Texto Descripción generada automáticamenteFunción patrones

Esta función muestra los patrones de forma secuencial con 3 segundos de delay.

## Función imagen



Esta función lo que hace es solicitarle al usuario que ingrese el patrón fila por fila de los LEDs a encender para mostrar el patrón deseado.

**Función publik**

Texto

Descripción generada automáticamente

Esta función lo primero que hace es mostrar un menú con las opciones:

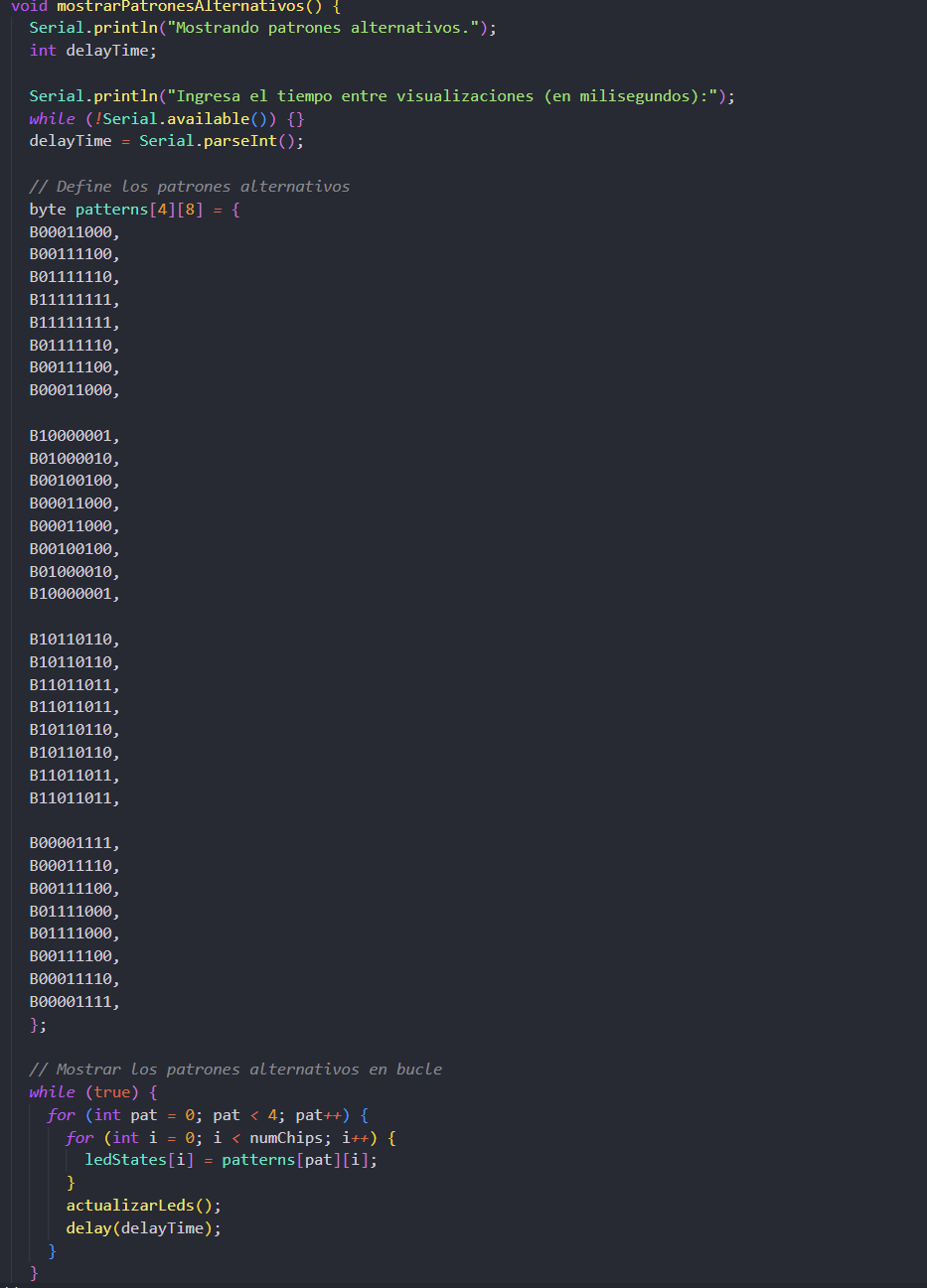
1. Verificar funcionamiento de LEDs.
2. Mostrar imagen de prueba.
3. Mostrar patrones alternativos.

Cada opción es una función diferente. La opción “a” es la función “verificarFuncionamientoLEDs()” y verifica que todos los LEDs funcionan correctamente.

Texto

Descripción generada automáticamente

La opción “b” es la función “mostarImagenPrueba()” y el usuario ingresa el patrón o imagen de prueba que desea ver.



La opción “c” es la función “mostrarPatronesAlternativos()” y muestra los 4 patrones con un delay que establece el usuario.