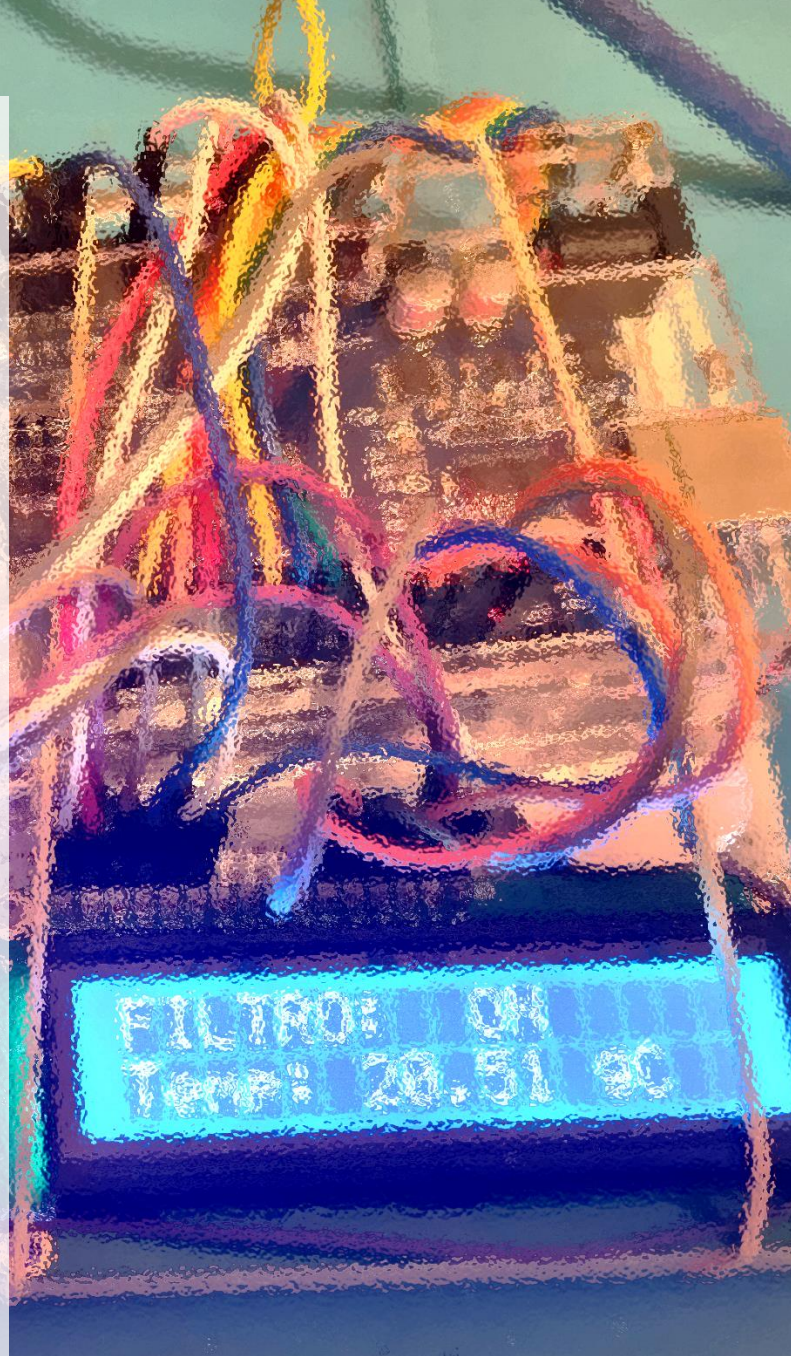


DISEÑO DE SISTEMAS ELECTRÓNICOS

CONTROL DE PISCINA:

Temperatura y Filtrado con Arduino UNO



Autor:

WAGNER, WALTER S.

Docentes:

ING. LISANDRO MARTÍN

ING. DANIEL CREPALDO

-2023-



ÍNDICE:

1. INTRODUCCIÓN:	2
2. OBJETIVOS Y DESAFÍOS:	3
3. TECNOLOGÍAS Y DESARROLLOS EXISTENTES:	4
Filtrado Automatizado:	4
Regulación de Temperatura:	6
Control Integrado:	7
CONCLUSIONES DEL ANÁLISIS DE MERCADO:	8
4. ESTUDIO ECONÓMICO:	9
1. Producción de 1 Dispositivo:	9
2. Producción de 10 Dispositivos:	10
3. Producción de 100 Dispositivos:	10
5. MODELADO:	12
6. IMPLEMENTACIÓN:	14
HARDWARE:	14
SOFTWARE:	18
7. CONCLUSIÓN:	24
8. BIBLIOGRAFÍA:	25
9. ANEXO I: MEF Modo:	26
10. ANEXO II: Manual de usuario	27

1. INTRODUCCIÓN:

El presente informe detalla un análisis de las viabilidades técnicas, económicas y comerciales relacionadas con la creación e implementación de un sistema de control para piscinas, que abarque tanto la automatización del proceso de filtrado como la regulación de la temperatura del agua. Además, se presenta un prototipo funcional desarrollado utilizando la plataforma Arduino UNO, destacando los desafíos enfrentados durante su realización, los logros obtenidos y las áreas de mejora identificadas tanto en términos de hardware como de software.

Este trabajo práctico fue ejecutado de manera individual en el contexto de la asignatura "Diseño de Sistemas Electrónicos" durante el presente año. El proceso fue supervisado y evaluado por los docentes a cargo de la cátedra. El objetivo central consistía en la creación de una aplicación con enfoque didáctico que, al mismo tiempo, pudiese ser escalable, proporcionando así las herramientas necesarias para abordar problemáticas propias de proyectos comerciales. De esta manera, se buscó establecer los cimientos utilizados en entornos profesionales, incorporando buenas prácticas y estándares aplicados en el diseño y desarrollo de sistemas electrónicos.

2. OBJETIVOS Y DESAFÍOS:

El presente proyecto se propone alcanzar una serie de objetivos específicos con el fin de abordar de manera integral el diseño, desarrollo y evaluación del sistema de control de piscina. Estos objetivos han sido establecidos con la finalidad de garantizar la creación de un prototipo funcional y escalable, así como de sentar las bases para potenciales aplicaciones comerciales futuras.

Como objetivo principal se destaca diseñar e implementar un sistema automatizado de control de piscina que integre funciones de filtrado automático y regulación precisa de la temperatura del agua, con el propósito de mejorar la eficiencia operativa y la comodidad del usuario. Se pretende alcanzar un prototipo funcional que, si bien pueda tener cuestiones a mejorar u optimizar, sirva de base como para evaluar la posibilidad de inserción en el mercado.

Dentro de los objetivos de segundo orden se creen conveniente mencionar:

- Investigar y analizar técnicamente las tecnologías disponibles para el filtrado automatizado de piscinas y la regulación de la temperatura del agua, identificando las soluciones más adecuadas para la aplicación propuesta.
- Indagar sobre los costos de las tecnologías antes mencionadas, en el mercado local a diferentes escalas de compra.
- Idear al menos tres posibles soluciones para pequeña, mediana y gran escala de producción, para poder así estudiar las posibilidades de penetración del mercado. Teniendo en cuenta y comparándolo con la relación costo/calidad de la competencia.
- Evaluar el rendimiento del prototipo mediante pruebas exhaustivas en un entorno controlado, identificando posibles fallos, deficiencias o áreas de mejora.
- Realizar un análisis de costos para determinar la viabilidad económica de la solución propuesta, considerando tanto los gastos iniciales de desarrollo como los posibles ahorros a lo largo del tiempo.
- Explorar oportunidades comerciales y aplicaciones potenciales para la tecnología desarrollada, considerando mercados que podrían beneficiarse de un sistema de control de piscina avanzado y eficiente.
- Proponer recomendaciones para optimizar tanto el diseño del hardware como el desarrollo del software, con el objetivo de mejorar la funcionalidad, la confiabilidad y la eficiencia del sistema.

3. TECNOLOGÍAS Y DESARROLLOS EXISTENTES:

En esta sección, se llevará a cabo un análisis de las tecnologías existentes relacionadas con el control automatizado de piscinas y la regulación de la temperatura del agua. Este estudio de mercado tiene como objetivo identificar las soluciones disponibles y evaluar sus características, ventajas y limitaciones en comparación con el enfoque propuesto en este proyecto. Se deja constancia que la cotización del dólar a la fecha es de \$640 y \$760 en el mercado informal.

Filtrado Automatizado:

En el ámbito del filtrado automatizado de piscinas, se utilizan desde simples temporizadores que accionan la bomba de agua por una determinada cantidad de horas por día, hasta sistemas avanzados que utilizan sensores de calidad del agua para determinar el momento óptimo para activar el proceso de filtrado. Estos sistemas integran bombas y válvulas controladas por microcontroladores que ajustan automáticamente los tiempos y la intensidad del filtrado. Además, existen soluciones basadas en la nube que permiten monitorear y controlar el sistema de filtrado de forma remota a través de aplicaciones móviles.

Realizando un análisis del mercado, se puede observar que son 3 líneas de productos las que destacan en este segmento:

- a) Temporizadores convencionales: estos dispositivos se caracterizan por su económico costo de \$4.231 por una unidad y \$3.000 comprando a niveles mayorista (6.6 U\$D y 4.7 U\$D respectivamente a cotización “solidario”). Dentro de las características se destaca su robustez y calidad. En contrapartida se puede decir que no es un dispositivo que permita tener versatilidad de configuraciones. Se muestra un ejemplo de este producto en la Figura 3.1 a continuación:

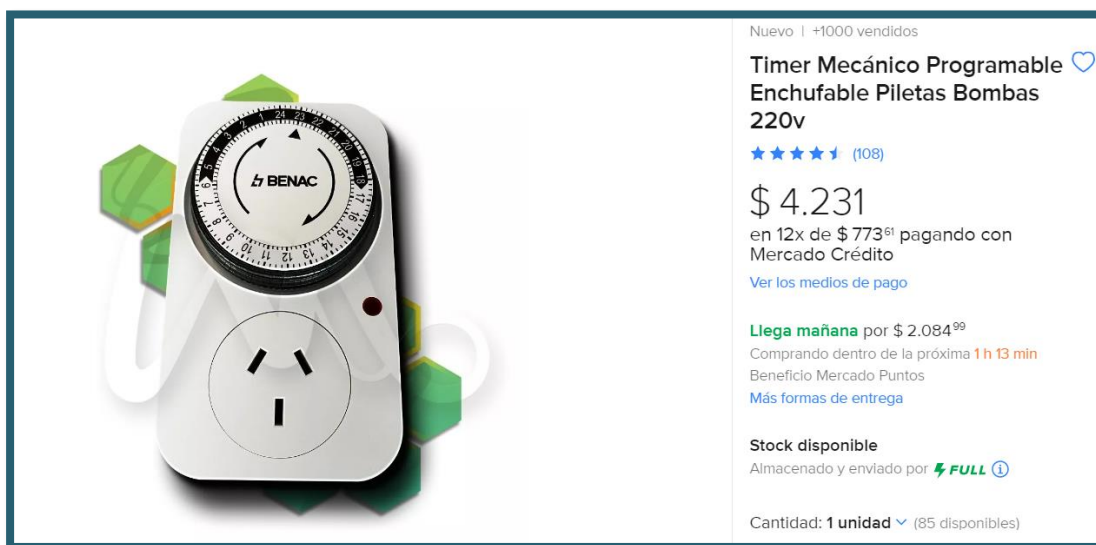


Figura 3.1: Temporizador convencional.

CONTROL DE PISCINA: Temperatura y Filtrado con Arduino UNO

- b) Temporizadores digitales: estos dispositivos se caracterizan por tener un costo intermedio que ronda entre \$25.000 y \$18.000 según la marca y los niveles de compra (entre 28 y 39 dólares “cotización solidario”). Ofrecen mayor versatilidad en cuanto a posibilidades de programación, pero suelen tener dificultades para la configuración. Se muestra un ejemplo de este producto en la *Figura 3.2* a continuación:

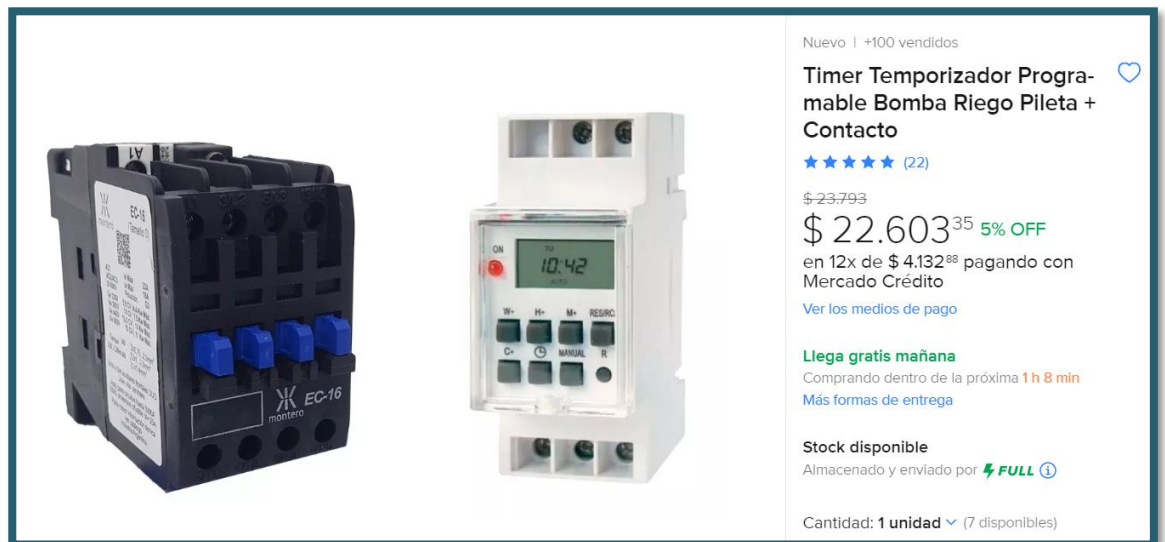


Figura 3.2: Temporizador digital.

- c) Temporizadores con más funcionalidades: estos dispositivos se caracterizan por brindar más soluciones al usuario, por ejemplo, controlar además la luminaria de la piscina y/o incorporar desarrollos de IoT basados en la nube. Suelen ser más amigables con el usuario, pero su costo puede ascender hasta los \$150.000, que equivalen a 234 dólares a la cotización “solidario”. Se muestra un ejemplo de este producto en la *Figura 3.3* a continuación:

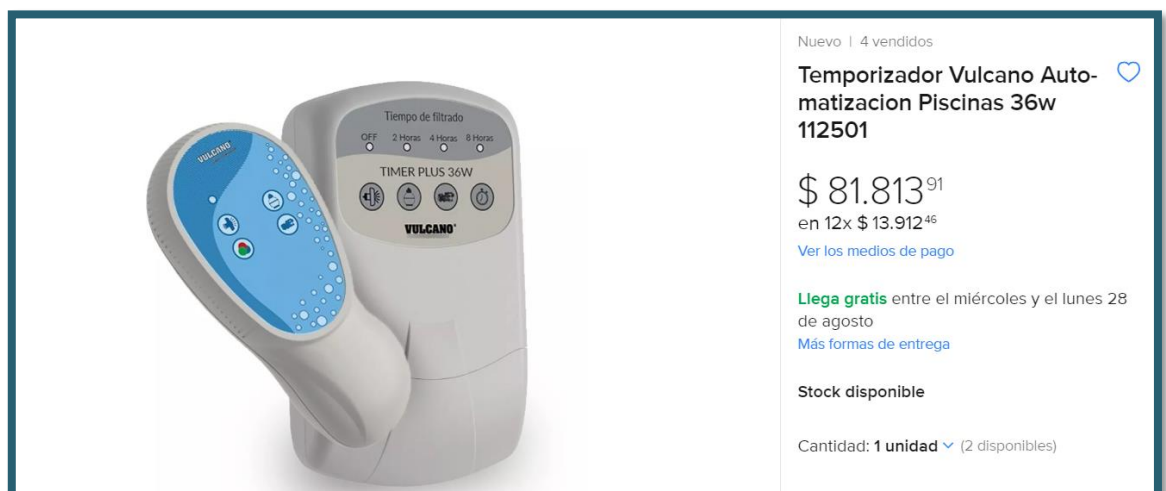


Figura 3.3: Temporizadores con más funcionalidades.

CONTROL DE PISCINA: Temperatura y Filtrado con Arduino UNO**Regulación de Temperatura:**

En cuanto a la regulación de la temperatura del agua en piscinas, los calentadores y enfriadores automatizados son ampliamente utilizados. Algunos sistemas incorporan sensores de temperatura que permiten mantener el agua dentro de un rango predefinido, lo que garantiza una experiencia cómoda para los usuarios. Además, existen sistemas de intercambio de calor que aprovechan fuentes de energía renovable, como paneles solares, para calentar el agua de manera eficiente. Sobre estos últimos no se realiza análisis del mercado ya que en las especificaciones de la problemática a abordar se aclara que se debe automatizar el funcionamiento concreto de una bomba de calor.

Al examinar el mercado detenidamente, se evidencian tres categorías de productos que sobresalen en este nicho:

- a) Temporizadores convencionales: algunos sistemas de control de temperatura funcionan “sin realimentación”, es decir, no evalúan la temperatura con sensores. En su lugar, infiriendo experimentalmente, accionan una bomba de calor por una determinada cantidad de tiempo al día. Para esto podrían usarse los temporizadores mostrados anteriormente en la sección de filtrado.
- b) Controles de temperatura “Low-Cost”: con origen importado desde China y con escasas reseñas en cuanto a la calidad, algunos equipos brindan una solución al control de temperatura a un muy bajo precio. Estos equipos pueden costar cerca de \$7000 (alrededor de 11 U\$D “soludarios”). Se muestra un ejemplo de este producto en la *Figura 3.4* a continuación:



Figura 3.4: Control de temperatura “Low-Cost”.

CONTROL DE PISCINA: Temperatura y Filtrado con Arduino UNO

- a) Termostatos digitales para piscinas o calderas: estos equipos suelen brindar una amplia variedad de programas y posibles configuraciones, cuentan con una calidad garantizada y una implementación amigable con el usuario. Su costo asciende hasta \$35.000 o \$40.000 (unos 60 U\$D en promedio a la cotización “solidario”). Se muestra un ejemplo de este producto en la *Figura 3.5* a continuación:

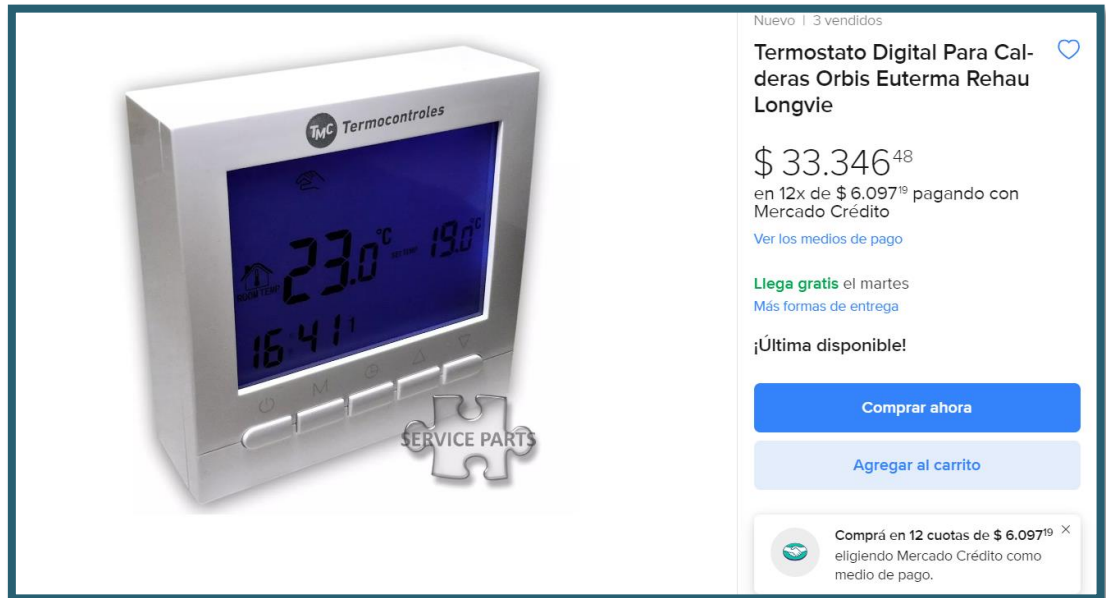


Figura 3.4: Termostatos digitales.

Control Integrado:

En algunos casos, se encuentran soluciones integradas que combinan el control automatizado tanto del filtrado como de la temperatura del agua. Estos sistemas permiten la programación y sincronización de ambas funciones, optimizando la operación y minimizando el consumo energético.

Realizando un estudio en el mercado local se pudo discriminar que, por limitaciones en las importaciones debido a la inestabilidad económica y las restricciones para ingresar al país, no hay variedad disponible en inventarios de estos dispositivos.

CONCLUSIONES DEL ANÁLISIS DE MERCADO:

En conclusión, el estudio de tecnologías existentes en el mercado revela un panorama diverso de soluciones para el control de piscinas y la regulación de la temperatura del agua. El enfoque de este proyecto busca aprovechar las ventajas de estas tecnologías existentes, superando sus limitaciones y brindando una alternativa más accesible y versátil.

Se ha determinado que existen muchos segmentos ya explotados en el mercado. Sin embargo, muchas de las soluciones disponibles presentan ciertas limitaciones. Algunos sistemas pueden ser costosos de implementar y mantener, lo que puede limitar su adopción en contextos residenciales o comerciales de menor escala. Además, la interfaz de usuario y la accesibilidad a veces pueden ser complicadas, dificultando la configuración y el monitoreo.

De todas maneras, se ha encontrado un nicho de mercado a explotar, que es un sistema que sea capaz de automatizar en simultáneo el control de temperatura y el control del filtrado de la piscina. Se pretenderá lograr que, haciendo uso de los recursos disponibles, se pueda alcanzar un prototipo aceptable capaz de cumplir esta tarea a un menor costo que la suma de los dos dispositivos por separado disponibles hoy en el mercado.

CONTROL DE PISCINA: Temperatura y Filtrado con Arduino UNO**4. ESTUDIO ECONÓMICO:**

En esta sección, se abordará un análisis detallado de los costos asociados con el diseño, desarrollo e implementación del sistema de control de piscina propuesto. Se evaluarán tanto los gastos iniciales como los costos operativos a lo largo del tiempo, con el objetivo de comprender la viabilidad económica de la solución planteada.

En el marco del análisis de costos, se llevarán a cabo tres estudios distintos, abordando diferentes enfoques para el desarrollo y producción del sistema. Se explorarán los costos asociados con la implementación basada en la plataforma Arduino UNO, considerando los siguientes niveles de producción: 1, 10 y 100 dispositivos.

En resumen, estos estudios de costos proporcionarán una comprensión integral de las implicaciones financieras asociadas con la implementación del sistema de control de piscina. A través de una comparación entre las alternativas de desarrollo a diferentes niveles de producción, se podrá determinar la estrategia más rentable y adecuada para llevar a cabo este proyecto.

DESARROLLO BASADO EN ARDUINO:

<u>COSTOS INDEPENDIENTE DEL NIVEL DE PRODUCCIÓN</u>				
Cantidad	Descripción	Proveedor	Costo	Total
20	Horas Programación	-	USD 8.0	USD 160.00
5	Horas Desarrollo Hardware	-	USD 8.0	USD 40.00
20	Horas Prueba y Optimización	-	USD 8.0	USD 160.00
TOTAL				USD 360.00

1. Producción de 1 Dispositivo:

En este escenario, se evaluarán los costos iniciales y operativos necesarios para producir un solo dispositivo utilizando la plataforma Arduino UNO. Dado que la producción es limitada, los costos fijos tendrán un mayor impacto en el costo total. Además, los costos de materiales se mantendrán en un nivel más alto debido a la adquisición de componentes en cantidades pequeñas.

<u>COSTOS DEPENDIENTE DEL NIVEL DE PRODUCCIÓN -> CASO 1</u>				
Cantidad	Descripción	Proveedor	Costo	Total
1	Arduino 1	MERCADOLIBRE	USD 8.2	USD 8.20
1	Teclado	MERCADOLIBRE	USD .9	USD 0.90
1	Display	MERCADOLIBRE	USD 4.0	USD 4.00
1	Modulo-Relés	MERCADOLIBRE	USD 3.5	USD 3.50
1	Varios	MERCADOLIBRE	USD 2.5	USD 2.50
TOTAL				USD 19.10

Costos totales estimados por producir 1 dispositivo: \$290.000

CONTROL DE PISCINA: Temperatura y Filtrado con Arduino UNO**2. Producción de 10 Dispositivos:**

Al aumentar la producción a 10 dispositivos, se espera una disminución en los costos de materiales gracias a la adquisición en volúmenes mayores. Los costos fijos podrán ser prorrateados entre más unidades, lo que reducirá su impacto en el costo individual de cada dispositivo. Esto puede resultar en una mejora en la viabilidad económica en comparación con la producción de un solo dispositivo.

<u>COSTOS DEPENDIENTE DEL NIVEL DE PRODUCCIÓN -> CASO 10</u>				
Cantidad	Descripción	Proveedor	Costo	Total
1	Arduino 1	MERCADOLIBRE	USD 6.6	USD 6.56
1	Teclado	MERCADOLIBRE	USD .7	USD 0.72
1	Display	MERCADOLIBRE	USD 3.2	USD 3.20
1	Modulo-Relés	MERCADOLIBRE	USD 2.8	USD 2.80
1	Varios	MERCADOLIBRE	USD 2.0	USD 2.00
1	Armado		USD 1.0	USD 1.00
TOTAL				USD 16.28

Costos estimados por equipo al producir 10 dispositivos: \$41.230

3. Producción de 100 Dispositivos:

En el escenario de producción de 100 dispositivos, los costos fijos se distribuirán de manera más efectiva, disminuyendo su influencia en el costo por unidad. Además, la economía de escala permitirá obtener descuentos en la compra de materiales, lo que podría resultar en una reducción sustancial en los costos de producción por dispositivo.

<u>COSTOS DEPENDIENTE DEL NIVEL DE PRODUCCIÓN -> CASO 100</u>				
Cantidad	Descripción	Proveedor	Costo	Total
1	Arduino 1	MERCADOLIBRE	USD 4.9	USD 4.92
1	Teclado	MERCADOLIBRE	USD .5	USD 0.54
1	Display	MERCADOLIBRE	USD 2.4	USD 2.40
1	Modulo-Relés	MERCADOLIBRE	USD 2.1	USD 2.10
1	Varios	MERCADOLIBRE	USD 1.5	USD 1.50
1	Armado		USD 1.0	USD 1.00
TOTAL				USD 12.46

Costos estimados por equipo al producir 100 dispositivos: \$12.160

CONTROL DE PISCINA: Temperatura y Filtrado con Arduino UNO

Al concluir el análisis de los costos asociados con la implementación del sistema de control de piscina, se destaca una perspectiva alentadora en relación con la viabilidad económica. La exploración detallada de tres enfoques distintos, basados en la plataforma Arduino UNO, revela que la rentabilidad se hace evidente a partir de un punto de inflexión clave: la producción de 100 productos.

Este hallazgo subraya la influencia de la economía de escala en la determinación de costos más eficientes. En particular, se observa que a medida que la producción aumenta, los costos fijos tienden a distribuirse de manera más equitativa entre las unidades, disminuyendo su impacto en el costo por dispositivo. Además, la adquisición de materiales en volúmenes más grandes permite acceder a precios más favorables, contribuyendo a una reducción significativa en los costos de producción.

5. MODELADO:

Para modelar el funcionamiento del dispositivo en cuestión, se emplearon técnicas de Máquinas de Estado Finito (MEF) jerárquicas, también conocidas como Statecharts UML. En esta aproximación de modelado, se busca una representación precisa del comportamiento del sistema, manteniendo la claridad y la simplicidad para facilitar su lectura y comprensión. Las complejidades que no contribuyen directamente al funcionamiento esencial del sistema se omiten deliberadamente, considerando que un análisis detallado podría realizarse a través de la revisión del código fuente, desarrollado en lenguaje C en este caso.

En un primer plano, se encuentra la MEF denominada "Modo", la cual se presenta en la *Figura 5.1* y se detalla con mayor resolución en el *Anexo I*.

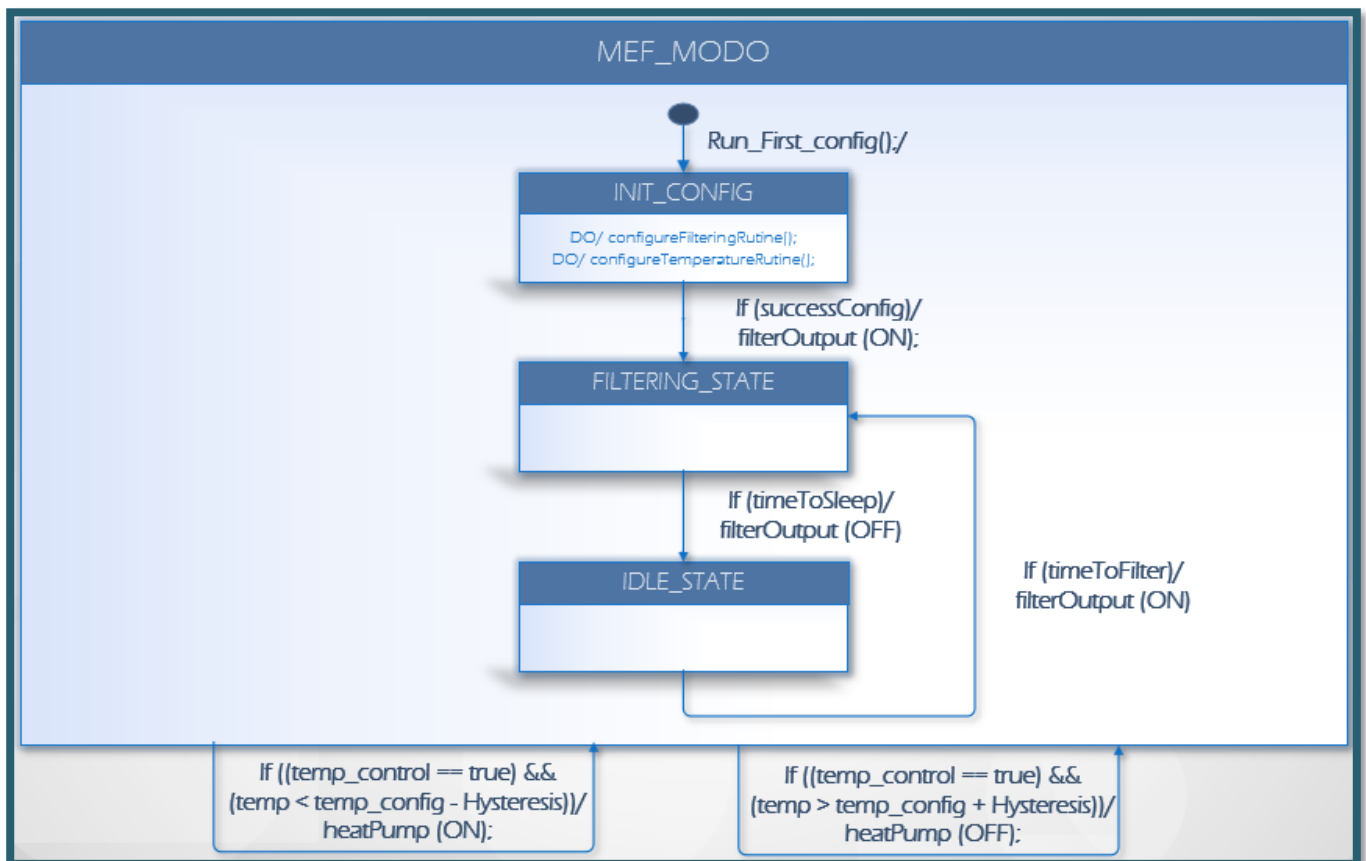


Figura 5.1: MEF MODO (Ver ANEXO I).

En su fase inicial, esta máquina de estados realiza una rutina de inicialización y luego transita al estado correspondiente al modo de operación en el que el dispositivo interactúa con el usuario a través de los periféricos, en concreto, el Teclado y el Display. En este estado, se configura la periodicidad del ciclo de filtrado y su duración. Asimismo, se establece si se activará el control de temperatura y, en caso afirmativo, se solicita al usuario definir la temperatura deseada para el agua.

CONTROL DE PISCINA: Temperatura y Filtrado con Arduino UNO

En el siguiente estado, el dispositivo controla la salida que activa la bomba del filtro, manteniéndola en estado alto hasta que el temporizador alcance el tiempo especificado para la rutina de filtrado. Una vez que se cumple este tiempo, se procede al siguiente estado, en el cual la bomba del filtro permanece apagada hasta que el temporizador indique que es momento de iniciar nuevamente el ciclo de filtrado. Se establece un ciclo repetitivo en el que estos estados se alternan, permitiendo la ejecución secuencial de las funciones.

Esta MEF se considera un superestado en sí mismo. Si el usuario opta por activar el control de temperatura, el dispositivo encuesta periódicamente un sensor designado para esta función. Si la temperatura se encuentra por debajo de la temperatura deseada, ajustada con un margen de seguridad de cinco grados, se activa la salida que controla la bomba de calor. Esta salida permanece activa hasta que la temperatura alcance el valor deseado más cinco grados, aplicando nuevamente un ciclo de histéresis para evitar el encendido y apagado continuo de la bomba.

En resumen, la utilización de Máquinas de Estado Finito jerárquicas proporciona un enfoque estructurado y organizado para modelar el funcionamiento del dispositivo, abstrayendo los detalles no esenciales y priorizando la comprensión general del proceso. Este modelado eficiente permite una implementación más efectiva y facilita el seguimiento y la optimización del comportamiento del sistema en diferentes estados operativos.

6. IMPLEMENTACIÓN:

HARDWARE:

En esta sección, se detalla el conjunto de componentes y periféricos utilizados en el proyecto, así como la justificación de la elección de cada uno de ellos. Se priorizó la utilización de hardware disponible para maximizar la eficiencia económica, considerando que, aunque existan alternativas más avanzadas, en el contexto de un proyecto estudiantil se optó por minimizar los costos.

Arduino UNO:

El Arduino UNO fue seleccionado como el núcleo central del sistema debido a su versatilidad y amplia comunidad de soporte. A pesar de existir placas más avanzadas, el Arduino UNO satisface los requisitos del proyecto y permite una programación relativamente sencilla. Sus virtudes incluyen su facilidad de uso, abundante documentación y capacidad para interactuar con diversos sensores y actuadores. La principal limitación es su memoria limitada, que podría restringir futuras expansiones del proyecto. Se muestra este microcontrolador a continuación en la *Figura 6.1*:



Figura 6.1: Arduino UNO.

Display LCD 16x2:

El display LCD 16x2 proporciona una interfaz visual para interactuar con el sistema. Aunque pantallas más grandes y a color están disponibles, el LCD 16x2 es suficiente para mostrar la información esencial y mantener la simplicidad. Su ventaja radica en su bajo

CONTROL DE PISCINA: Temperatura y Filtrado con Arduino UNO

consumo de energía y en su compatibilidad con el Arduino UNO. La limitación clave es su capacidad de visualización, que podría requerir cierta compactación de la información. Se muestra este periférico a continuación en la *Figura 6.2*:

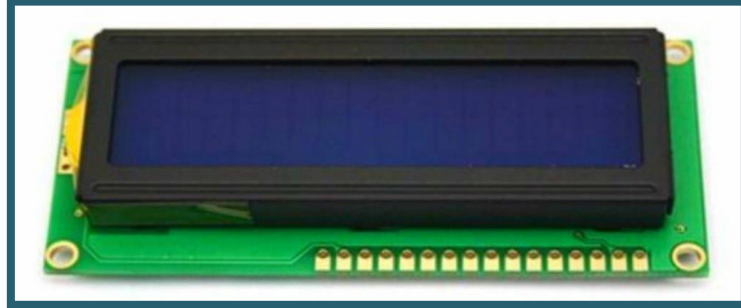


Figura 6.2: Display LCD 1602.

Teclado de Membrana 4x3:

El teclado de membrana 4x3 se eligió como una forma económica y eficiente de entrada de datos. Aunque existen teclados más sofisticados con mayor cantidad de teclas, el 4x3 satisface las necesidades básicas del proyecto. Su ventaja principal es su tamaño compacto y su sencilla conexión. Sin embargo, su limitación radica en la cantidad de entradas que “consume” del micro por no ser multiplexado, esto podría restringir futuras expansiones. Se muestra este periférico a continuación en la *Figura 6.3*:



Figura 6.3: Teclado 4x3.

Sensor de Temperatura LM35:

El sensor LM35 se seleccionó para medir la temperatura del agua de la piscina debido a su simplicidad y precisión. A pesar de alternativas más avanzadas, como los sensores digitales, el LM35 ofrece una lectura analógica directa que se adapta al enfoque del proyecto. Su virtud principal es su alta precisión y fácil integración con el Arduino

CONTROL DE PISCINA: Temperatura y Filtrado con Arduino UNO

UNO. Se aclara que en el prototipo se utiliza un sensor de encapsulado típico pero para la aplicación real debería utilizarse un encapsulado que le permita estar sumergido. Se muestra este sensor a continuación en la *Figura 6.4*:



Figura 6.4: Sensor LM35.

Leds para Emulación de Contactor:

Dos leds se emplearon para simular el encendido y apagado de los contactores de la bomba de calor y la bomba de filtrado. Aunque en una implementación comercial se usarían relés, los leds cumplen su función para demostraciones y pruebas. Su virtud principal es la simplicidad y la claridad visual de la indicación. La limitación es que no ofrecen la funcionalidad real de control de dispositivos eléctricos.

Potenciómetro para Brillo del Display:

El potenciómetro se utilizó para ajustar el brillo del display LCD. Aunque existen métodos más automatizados para el control del brillo, el potenciómetro ofrece una solución simple y funcional. Su ventaja radica en su facilidad de implementación y ajuste manual. La principal limitación es que podría no ser tan intuitivo como otros métodos de ajuste de brillo. En la implementación real se utilizaría una resistencia fija sin acceso a variaciones del usuario.

Protoboard y Cables de Conexión:

La protoboard y los cables de conexión fueron utilizados para crear el circuito electrónico del prototipo. Estos elementos son esenciales para la conexión segura y ordenada de los componentes. La virtud principal es su flexibilidad y reutilización para futuros proyectos. La limitación puede radicar en la complejidad de la conexión, que podría requerir una disposición cuidadosa para evitar confusiones.

CONTROL DE PISCINA: Temperatura y Filtrado con Arduino UNO

En resumen, la elección de hardware se orientó hacia soluciones funcionales y asequibles para el contexto estudiantil. Aunque existen alternativas más avanzadas, los componentes seleccionados permiten la demostración efectiva del sistema de control de piscina, cumpliendo con los objetivos del proyecto y estableciendo una base sólida para futuras expansiones y mejoras. Toda la implementación se muestra a continuación en la *Figura 6.5*:

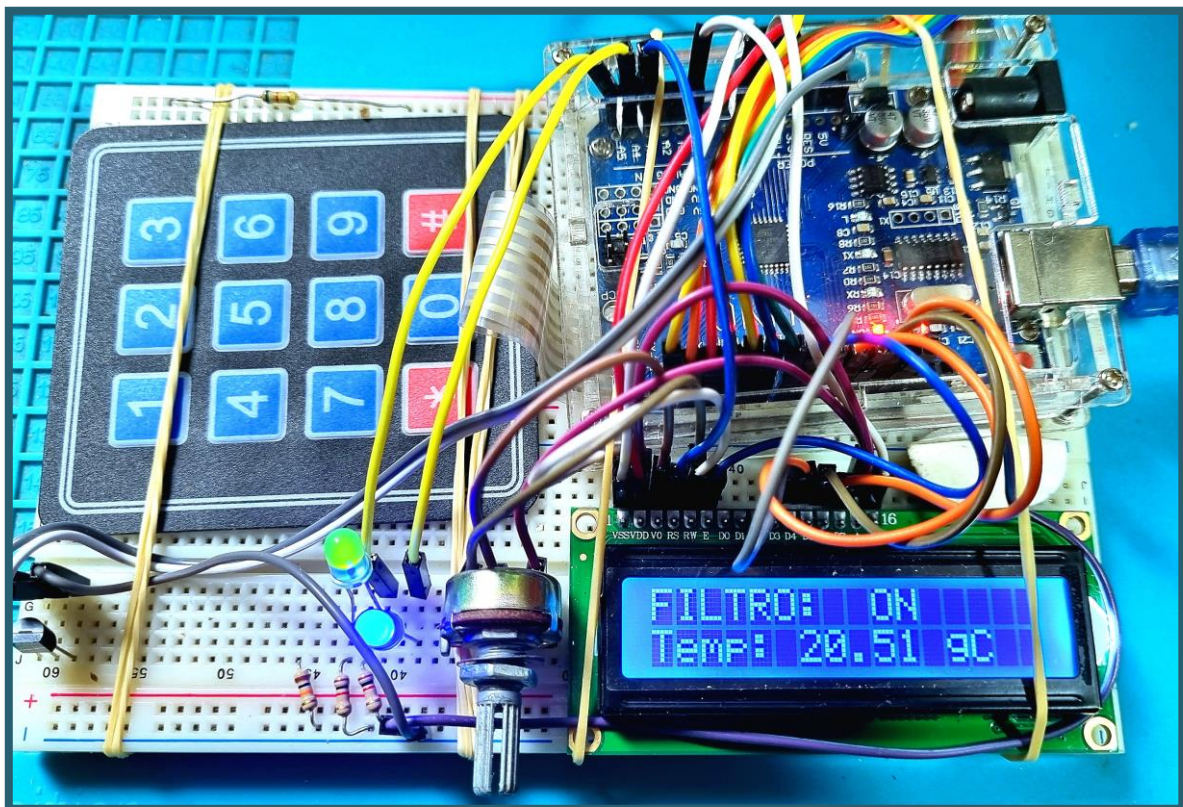


Figura 6.5: Prototipo del CONTROL DE PISCINA: Temperatura y Filtrado con Arduino UNO.

Si bien el sistema presentado actualmente se encuentra en una etapa de prototipo, la visión a largo plazo apunta hacia su transformación en un producto finalizado y robusto. Esta aspiración implica la integración del sistema en un gabinete estándar y normalizado, que se asemeje a los gabinetes utilizados para albergar componentes eléctricos como disyuntores diferenciales y otros elementos de control. Esta dirección se alinea con las normativas de seguridad eléctrica, facilitando la instalación y el uso del sistema en entornos diversos. En esta configuración final, el sistema se encontraría acompañado por dos contactores distintos: uno destinado a la bomba de calor y otro para la bomba de filtrado.

SOFTWARE:

Mostrar la codificación de todo el proyecto en este informe sería un despropósito, ya que implicaría exhibir muchas de líneas de código extendiendo desmesuradamente la presentación. En su lugar, se creyó conveniente hacer énfasis en las cuestiones forma que se contemplaron al desarrollar el código. Si el lector deseara consultar cuestiones en lo particular (-se recomienda hacerlo-), se deja al pie el proyecto publicado en GitHub¹ para que pudiese tenerlo a disposición.

A lo largo del proceso de convertir el problema modelado por MEFs en una implementación en lenguaje C consistente, se trabajó en capas diferenciando aplicación de drivers, haciendo uso de archivos y carpetas diferentes se logró una organización intuitiva del proyecto, como se puede visualizar en la Figura 6.6. Esto permite reutilizar fácilmente código de cualquier capa, pero también simplifica en gran medida la depuración y optimización del proyecto.

Además, se tuvo en cuenta como buena práctica colocar un nombre significativo a cada archivo y a cada función o subrutina del mismo. En adición, el nombre de cada subalgoritmo comienza con el nombre del archivo que lo contiene.

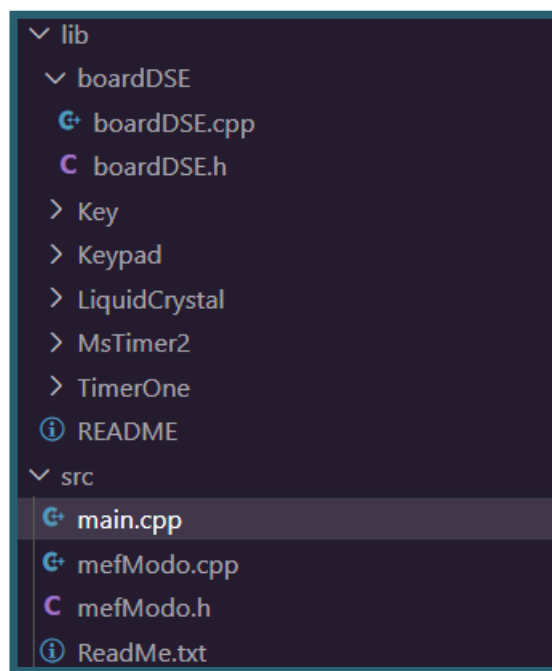


Figura 6.6: Organización de los archivos del proyecto.

En consecuencia, de haber segmentado y modularizado el código se obtuvo un archivo principal “main.c” corto y muy sencillo, como se muestra en la Figura 6.7.

¹ https://github.com/WalyWagner/ControlDePiscina_Temperatura-Filtrado_ArduinoUNO

```

G+ main.cpp > ...
> /***** CONTROL DE PISCINA - Temperatura y Filtrado *****/
/**
 * @file main.cpp
 * @brief Contains the main program to be uploaded to ARDUINO UNO.
 */

/*=====Inclusions=====*/
#include <Keypad.h>
#include <Key.h>
#include <mefModo.h>
#include <boardDSE.h>
#include <Arduino.h>
#include <LiquidCrystal.h>

/*=====Internal Data Definitions=====*/
char keys[KEYPAD_ROWS][KEYPAD_COLS] = {
    {'1','2','3'},
    {'4','5','6'},
    {'7','8','9'},
    {'*','0','#'}
};
byte rowPins[KEYPAD_ROWS] = {5, 4, 3, 2};
byte colPins[KEYPAD_COLS] = {8, 7, 6};

/*=====Object Instantiations=====*/
Keypad keypad = Keypad( makeKeymap(keys), rowPins , colPins , KEYPAD_ROWS, KEYPAD_COLS );
LiquidCrystal lcd(LCD_RS, LCD_EN, LCD_D4, LCD_D5, LCD_D6, LCD_D7);

/*=====Setup configurations=====*/
/**
 * @brief We need to initialize the hardware and the application
 */
void setup(){
    boardDSE_init();
    mefModo_init();
}

/*=====Application Layer=====*/
/**
 * @brief Here we are polling our MEF!
 */
void loop(){
    mefModo();
}

/*=====End of file=====*/
** * @} */
/*=====File: main.cpp=====*/

```

Figura 6.7: Archivo principal “main.c” del proyecto.

Al igual que el Hardware, el software aún se encuentra en etapa de desarrollo y prototipado. Dentro de las cuestiones a mejorar se destaca:

- ✓ La capa de Aplicación todavía incurre en “HardCoding”, manejando directamente a los periféricos. Sabemos que esto no es una buena práctica, ya que en caso de cambiar un periférico por otro, no es esperado tener que modificar la capa de aplicación.

CONTROL DE PISCINA: Temperatura y Filtrado con Arduino UNO

- ✓ La interfaz del display con el usuario no es del todo amigable, debería mostrar de manera más gráfica los mensajes, dándole la posibilidad de “equivocarse” al usuario y volver para atrás por ejemplo y mostrando avisos y alertas cuando es necesario.
- ✓ Los temporizadores deben aún pre-escalarse adecuadamente. Como el prototipo está en estadio de desarrollo, cuando el usuario ingresa, por ejemplo, que desea filtrar su piscina cada 8 horas con ciclos de 2 horas, el programa configura la salida en alto cada 8 segundos por un ciclo de 2 segundos, respectivamente.

Se mostrará a continuación, en la *Figura 6.8* el código que corresponde a la capa de aplicación, en donde se implementa en C++ la MEF propuesta en la etapa de modelado:

```

/***** CONTROL DE PISCINA - Temperatura y Filtrado *****/
/**
 * @file mefModo.cpp
 * @brief This is the application layer of the implementation.
 * Must be upgraded!
 */

/*=====Inclusions=====*/
#include <mefModo.h>

/*=====External Data Definitions=====*/
extern Keypad keypad;
extern LiquidCrystal lcd;

/*=====Internal Data Definitions=====*/
static bool filtering = false; // ->Flag to indicate the actual state
static bool waiting = false; // ->Flag to indicate the actual state
static bool wantToControlTemp; // ->Flag to indicate if the temperature control is ON
static int hoursToFilter, hoursToSleep, temperatureToHeat; // -> Timers & temperature

/*=====Internal Data Declarations=====*/
static estMefModo_enum est_MefModo;

/*=====internal function declaration=====*/
/**
 * @brief this function is attached to TIMER 1 interruption routine
 *
 * You to change from filtering mode to sleep mode when the work is done!
 */static void filteringTimer(void);

/**
 * @brief this function is attached to TIMER 2 interruption routine
 *
 * You want to measure temperature periodically

```

```

*/static void temperatureTimer(void);

/*=====external functions definition=====*/

/**
 * @brief Initializes the MEF
 */
void mefModo_init() {
    est_MefModo = EST_INIT_CFG;
}

/**
 * @brief Run the MEF
 */
void mefModo() {
    switch (est_MefModo) {
        case EST_INIT_CFG:

            /*FALTA MEJORAR QUE LA CAPA DE APLICACIÓN NO OPERE DIRECTAMENTE EL HARDWARE !!
            * ES DECIR, HABRÍA QUE HACER EN boardDSE.cpp SUBROUTINAS QUE OPEREN EL DISPLAY,
            * EL TECLADO, ETC Y QUE SEAN CONVOCADAS DESDE AQUÍ */

            lcd.setCursor(0, 0);    lcd.print("INGRESE LAS HS.");
            lcd.setCursor(0, 1);    lcd.print("ENTRE CICLOS DE");
            delay(2000);            lcd.clear();
            lcd.setCursor(0, 0);    lcd.print("FILTRADO EN 2 ");
            lcd.setCursor(0, 1);    lcd.print("DIGITOS (1-24)");
            hoursToSleep = keypad.waitForKey() - 48;    // Ingresa la decena y se castea a int
            hoursToSleep = hoursToSleep * 10 + keypad.waitForKey() - 48; // Ingresa la unidad y se castea a int
            acumulando la decena
            lcd.clear(); lcd.print(hoursToSleep); delay(1000); lcd.clear();

            lcd.setCursor(0, 0);    lcd.print("INGRESE LAS HS.");
            lcd.setCursor(0, 1);    lcd.print("DE DURACION DE");
            delay(2000);            lcd.clear();
            lcd.setCursor(0, 0);    lcd.print("CADA CICLO EN 2");
            lcd.setCursor(0, 1);    lcd.print("DIGITOS (1-24)");
            hoursToFilter = keypad.waitForKey() - 48;    // Ingresa la decena y se castea a int
            hoursToFilter = hoursToFilter * 10 + keypad.waitForKey() - 48; // Ingresa la unidad y se castea a int
            acumulando la decena
            lcd.clear(); lcd.print(hoursToFilter); delay(1000); lcd.clear();

            lcd.setCursor(0, 0);    lcd.print("DESEA ACTIVAR EL");
            lcd.setCursor(0, 1);    lcd.print("CONTROL DE TEMP?");
            delay(2000);            lcd.clear();
            lcd.setCursor(0, 0);    lcd.print("INGRESE UN '1'");
            lcd.setCursor(0, 1);    lcd.print("EN CASO DE SI !");

```

CONTROL DE PISCINA: Temperatura y Filtrado con Arduino UNO

```

    if ((keypad.waitForKey()-48) == 1){
        wantToControlTemp=true;
        lcd.setCursor(0, 0);    lcd.print("INGRESE LA TEMP");
        lcd.setCursor(0, 1);    lcd.print("DESEADA EN GRADOS");
        delay(2000);            lcd.clear();
        lcd.setCursor(0, 0);    lcd.print("HAGALO EN 2 ");
        lcd.setCursor(0, 1);    lcd.print("DÍGITOS ");
        //Serial.println("INGRESE A LA TEMPERATURA QUE DESEA MANTENER EL AGUA");
        //Serial.println("HÁGALO EN FORMATO DE 2 DÍGITOS (1-60)");
        temperatureToHeat = keypad.waitForKey() - 48;    // Ingresar la decena y se castea a int
        temperatureToHeat = temperatureToHeat * 10 + keypad.waitForKey() - 48;    // Ingresar la unidad y
se castea a int acumulando la decena
        //Serial.print("Grados:");
        //Serial.println(temperatureToHeat);
        MsTimer2::set(1000,temperatureTimer);
        MsTimer2::start();
        lcd.clear(); lcd.print(temperatureToHeat); delay(1000); lcd.clear();
    }

    Timer1.initialize(1000000 * hoursToFilter);    //Configura el TIMER en hoursToFilter Segundos
    Timer1.attachInterrupt(filteringTimer) ; //Configura la interrupción del Timer 1

    filtering = true;

    lcd.clear(); lcd.print("FILTRO:  ON");

    digitalWrite(FILTERING_PORT,HIGH);

    est_MefModo = EST_FILTERING;

    break;

case EST_FILTERING:
    if(waiting){
        est_MefModo = EST_WAITING;
        lcd.setCursor(0, 0);    lcd.print("FILTRO:  OFF");
        digitalWrite(A4,LOW);
    }

    break;

case EST_WAITING:
    if(!waiting && filtering){
        est_MefModo = EST_FILTERING;
        lcd.setCursor(0, 0);    lcd.print("FILTRO:  ON");
        digitalWrite(A4,HIGH);
    }
}

```

```

        break;
    }
}
/*=====[internal function definition]=====*/
void filteringTimer(void){
    if (filtering==true && waiting == false) // we have interrupted because filtering routine is done
    {
        filtering = false ;
        waiting = true;
        Timer1.stop();
        Timer1.initialize(1000000 * hoursToSleep);      //Configura el TIMER en hoursToFilter Segundos
        Timer1.attachInterrupt(filteringTimer) ; //Configura la interrupción del Timer 1
        return;
    }
    if (filtering==false && waiting == true) // we have interrupted because filtering routine is done
    {
        filtering = true ;
        waiting = false;
        Timer1.stop();
        Timer1.initialize(1000000 * hoursToFilter);      //Configura el TIMER en hoursToFilter Segundos
        Timer1.attachInterrupt(filteringTimer) ; //Configura la interrupción del Timer 1
        return;
    }
}
void temperatureTimer(void){ // CON HISTÉRESIS DE +- 5 GRADOS !!
    float tempC = analogRead(TEMPERATURE_SENSOR);
    tempC = (5.0 * tempC * 100.0)/1024.0;
    lcd.setCursor(0, 1);    lcd.print("Temp: ");
    lcd.setCursor(6, 1);    lcd.print(tempC);
    lcd.setCursor(12, 1);   lcd.print("gC");

    if (tempC < temperatureToHeat - 5){
        digitalWrite(HEATING_PORT,HIGH);
    }else{
        if (tempC > temperatureToHeat + 5){
            digitalWrite(HEATING_PORT, LOW);
        }
    }
}
}
/*=====[End of file]=====*/
/**
 * @}
 */
/*=====[File: mefModo.cpp]=====*/

```

Figura 6.8: mefMOD0.cpp

7. CONCLUSIÓN:

En el presente informe técnico, se ha abordado el diseño, desarrollo y evaluación de un prototipo de sistema de control automatizado para la gestión eficiente de la temperatura y el comando preciso del filtrado de piscinas. Este proyecto surge como resultado del trabajo llevado a cabo en el marco de la asignatura Diseño de Sistemas Electrónicos, con el objetivo de ofrecer una aplicación práctica que combine conocimientos teóricos con habilidades de implementación. A lo largo de este informe, se han detallado los aspectos esenciales de este proceso, desde la conceptualización hasta la viabilidad económica y las perspectivas de implementación futura.

La introducción del informe estableció las bases, resaltando las motivaciones detrás del proyecto y delineando los objetivos a alcanzar. La investigación y el análisis de tecnologías existentes en el mercado permitieron un entendimiento de las soluciones disponibles para el control de piscinas y la regulación de la temperatura del agua. Esta revisión proporcionó el contexto necesario para la selección y la justificación del enfoque propuesto.

La sección de objetivos delineó claramente las metas que se aspiraban alcanzar en este proyecto, desde el diseño y la implementación del sistema automatizado hasta la evaluación de su rendimiento y viabilidad. La subdivisión en objetivos principales y específicos garantizó un enfoque estructurado para la ejecución del proyecto.

El estudio de costos constituyó un componente esencial en la toma de decisiones informadas. La evaluación de los costos de desarrollo y operativos en diferentes niveles de producción permitió identificar el punto de inflexión donde la rentabilidad del proyecto se hace evidente. La comparación entre tres enfoques de desarrollo y las diversas escalas de producción proporcionaron un panorama completo de las implicaciones económicas.

En el análisis detallado del hardware utilizado, se presentaron los componentes clave seleccionados para la implementación del sistema de control de piscina. La elección de hardware se orientó hacia soluciones funcionales y económicas, considerando la disponibilidad de recursos y el contexto de un proyecto estudiantil. Se describieron los periféricos utilizados y se resaltaron sus virtudes y limitaciones, estableciendo una base técnica sólida. Paralelamente se hizo un detalle del software, en donde plantearon los paradigmas y estándares sobre los que se trabajaron. Se mencionaron las ventajas y desventajas del prototipo, marcando el rumbo de los nuevos objetivos que hay por cumplir.

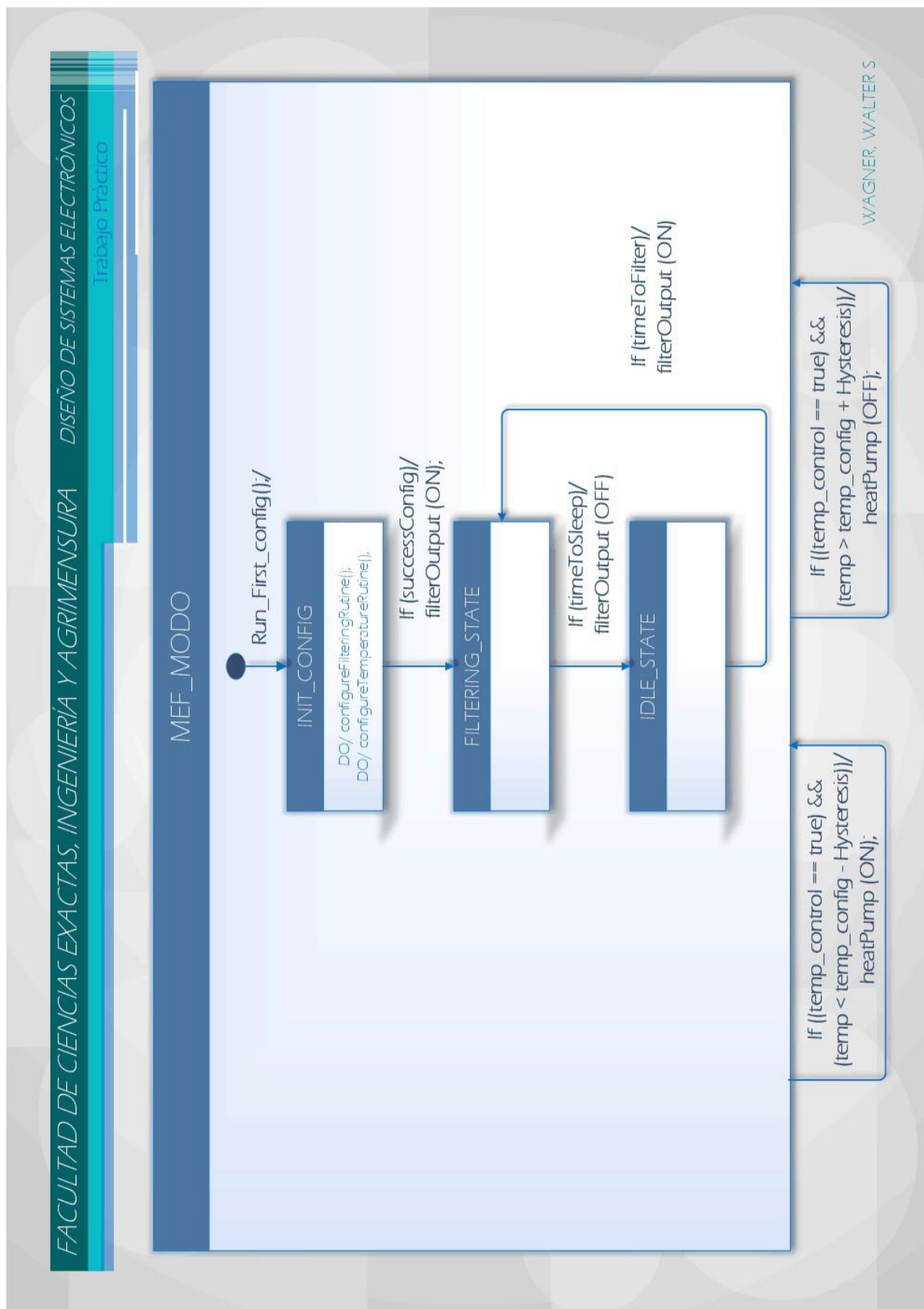
Se destacó la intención de transformar el prototipo en un producto finalizado y comercializable, siguiendo normativas y estándares de seguridad. La inclusión de elementos como un gabinete normalizado y contactores adicionales reforzaría tanto la funcionalidad como la seguridad del sistema en aplicaciones comerciales y residenciales.

En resumen, este informe técnico ha documentado exhaustivamente el proceso de diseño, desarrollo y evaluación de un sistema de control automatizado de piscinas. Desde la conceptualización hasta las perspectivas de implementación futura, cada etapa ha sido abordada con rigor y enfoque.

En lo consignado se solicitaba desarrollar un manual de usuarios, es por esto que en el *Anexo II* podrá el lector encontrar la guía de instrucciones. Se sobre entiende que esta guía debería sufrir pequeñas modificaciones cuando el desarrollo pase de la etapa de prototipo a producto.

8. **BIBLIOGRAFÍA:**

- ARDUINO Curso práctico de formación. Óscar Torrente Arteo. Alfaomega.
- Practical UML Statecharts in C/C++. Event-Driven Programming for Embedded Systems. Miro Samek. Elsevier. 2° Edición, 2009
- <https://www.arduino.cc/reference/en/libraries/ds18b20/>
- <https://github.com/PaulStoffregen/TimerOne>
- <https://www.arduino.cc/reference/en/libraries/mstimer2/>
- <https://www.arduino.cc/reference/en/libraries/liquidcrystal/>
- <https://playground.arduino.cc/Code/Keypad/>

9. ANEXO I: MEF Modo:

10. ANEXO II: Manual de usuario

CONTROL DE PISCINA: Temperatura y Filtrado

Manual de usuario:

¡Bienvenido al Dispositivo de Control Automatizado de Piscina! Este manual le guiará paso a paso para configurar y operar su sistema, diseñado para hacer el mantenimiento de su piscina más sencillo y eficiente. No se requieren conocimientos en electrónica para utilizarlo.

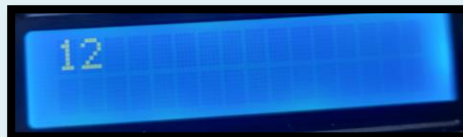
1. **Conexión Inicial:** Conecte el dispositivo a los contactores y la fuente de alimentación según el esquemático proporcionado en la caja del producto.

2. **Configuración de ciclos de filtrado 1:** Un mensaje aparecerá en pantalla pidiéndole que ingrese cada cuántas horas desea realizar un ciclo de filtrado. Ingrese su respuesta en horas en dos dígitos. Por ejemplo:

- a. 2 horas → 02
- b. 10 horas → 10



3. **Verificación 1:** Un nuevo mensaje aparecerá en pantalla informándole las horas ingresadas, por favor verifique que coincida con lo que usted deseaba ingresar. En el ejemplo se ingresó 12.



4. **Configuración de ciclos de filtrado 2:** El display mostrará un mensaje indicándole que ingrese de cuántas horas desea que sean los ciclos de filtrado. Ingrese su respuesta en horas en dos dígitos. Por ejemplo:

- a. 2 horas → 02
- b. 10 horas → 10



5. **Verificación 2:** Un nuevo mensaje aparecerá en pantalla informándole las horas ingresadas, por favor verifique que coincida con lo que usted tenía en mente. En el ejemplo se ingresó 02.



CONTROL DE PISCINA: Temperatura y Filtrado con Arduino UNO

6. **Control de temperatura:** Se le indicará un cartel solicitándole que ingrese un 1 si desea activar el control de temperatura. Ingrese cualquier otra tecla en caso de que su respuesta sea negativa.



7. **Configuración de la temperatura:** En caso de que su respuesta sea positiva se le pedirá que ingrese la temperatura a la cual desea mantener su piscina, por favor ingrésele en grados centígrados en DOS dígitos. Por ejemplo:
- a. 40 grados → 40
 - b. 35 grados → 35



8. **Verificación de la temperatura:** En caso de que haya activado el control de temperatura, una vez seteada la misma en el paso 7, se le mostrará el valor ingresado. Por favor verifique que éste coincida con el esperado. En el ejemplo se ingresó 45:



9. **Inicio del sistema:** ¡LSITO! El sistema comenzará a funcionar. Indicará en el display el estado de la bomba de filtrado y la temperatura del agua en grados centígrados.



IMPORTANTE:

- ✓ En caso de haber tenido problemas en la configuración reinicie el dispositivo y vuelva a empezar.
- ✓ Se recomienda la conexión de una batería como lo indica el esquemático en la caja para no tener que repetir toda la configuración cada vez que se suspenda el suministro eléctrico.