

# **SISTEMAS ELECTRÓNICOS DIGITALES**

**Jose Antonio Molina Montoro – 74736108-T**

**josebet@correo.ugr.es**

## **Memoria del proyecto basado en microcontrolador**

### **1.- Descripción del global del proyecto**

El proyecto consiste en un sistema electrónico basado en PIC que controlará diversos parámetros del funcionamiento de una piscina cubierta. En concreto, controlará los niveles analógicos medidos de cloro y turbidez del agua, para que en consecuencia actúen los dosificadores de cloro o floculante para devolverlos a los niveles establecidos. Además, se controlará continuamente la temperatura del agua, de modo que siempre que sea necesario calentar el agua para alcanzar la temperatura deseada se accionará el calentador del agua. La otra funcionalidad será la de accionar el motor de funcionamiento de la depuradora, que tendrá una hora establecida de encendido y otra de apagado del motor.

El usuario interaccionará con el sistema mediante un display LCD que le mostrará diversos datos medidos, así como un menú accionado por pulsadores que mostrará las diversas opciones por el display LCD. Con este menú, se podrán configurar diversos parámetros, como los niveles deseados de temperatura, cloro y turbidez en el agua, así como la hora de encendido y apagado del motor de la depuradora. Estos parámetros al ser modificados se guardarán en memoria EEPROM, de modo que si se apaga el sistema, los datos podrán volver a ser leídos. Además, para elegir los valores de estos parámetros, se usará un teclado matricial, con el que podrán introducirse los valores más cómodamente conforme se muestra el valor a modificar por la pantalla del LCD.

Al respecto de la memoria EEPROM, los datos no se guardarán únicamente en la memoria EEPROM interna incluida en el PIC, sino que también se replicarán en una memoria I2C externa. Como esta memoria solo será leída en casos de arranque del sistema, solo se leerá de una de ellas al arrancar el sistema, y por defecto esta será la interna. Emplearemos un flag de inicio en la secuencia de datos guardados, de modo que siempre que modifiquemos datos en la memoria, incluiremos la secuencia en hexadecimal "ABCD" de modo que lo primero que hará al arrancar el sistema será comprobar si esta secuencia está en la parte de memoria donde guardamos la información relativa a estos parámetros. En caso de no estar (por ejemplo, que hubiera que reemplazar el PIC), comprobará en la memoria externa. En caso de que tampoco se hayan guardado datos en la memoria externa, cargará unos valores predefinidos.

### **2.- Descripción hardware**

En primer lugar, el sistema completo estará gestionado por el microcontrolador PIC 18F4520, de gama mejorada. Emplearemos este PIC debido a su inclusión de diversos puertos de entrada y salida, especialmente los relacionados con las interfaces que emplearemos, que

serán fundamentalmente I2C, 1-wire y RS-232. Además, este PIC también incluye diversas interrupciones externas que emplearemos para despertar al PIC.

Para llevar a cabo el control de la hora de activación y apagado de la depuradora cada día, emplearemos un dispositivo de reloj-calendario, en este caso un dispositivo I2C, el PCF8583. Este dispositivo estará conectado de tal manera que genere una interrupción en el PIC, que activará el flag que lleve a cabo la rutina de control de la depuradora.

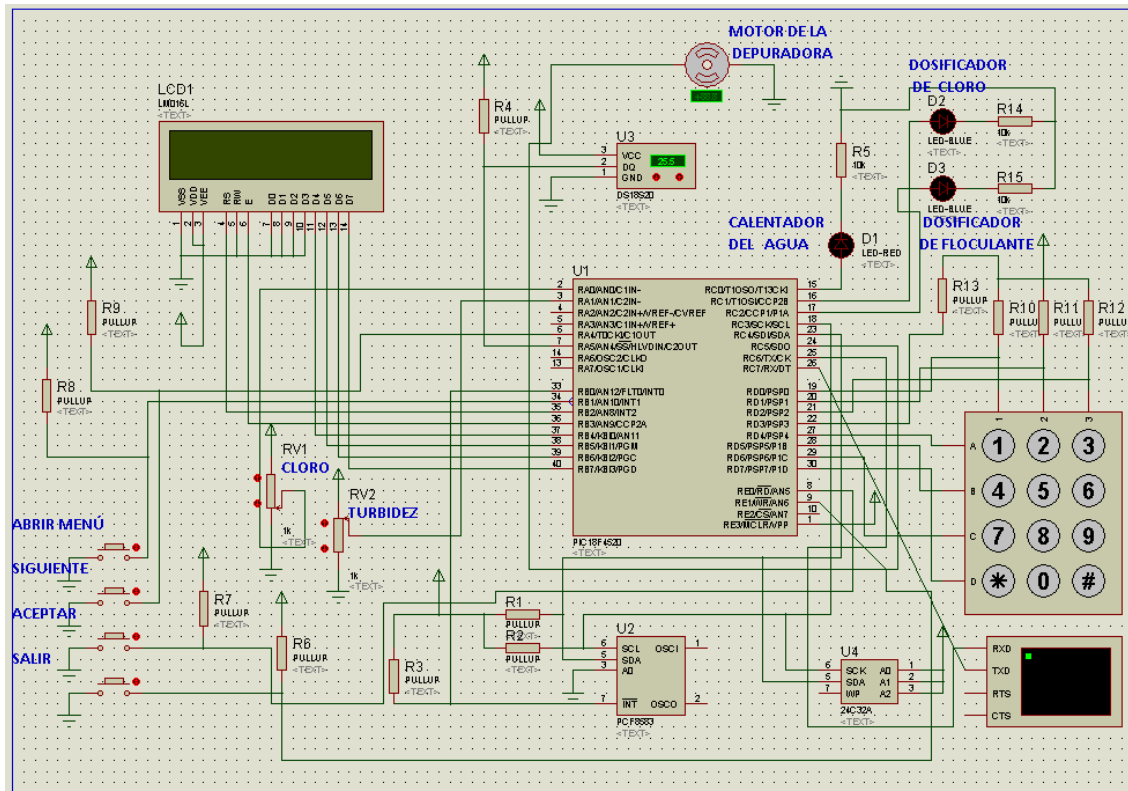
El otro dispositivo I2C que usaremos será una memoria EEPROM externa 24C32A, con 4089 palabras de 8 bytes (en total 32712 bytes). Esta memoria es de mayor capacidad que la interna del PIC, pero de todas formas, únicamente se empleará para guardar los parámetros de control de la piscina, que son únicamente 16 bytes entre todos los parámetros. En caso de que quisiéramos guardar un registro de los valores de temperatura medidos, por ejemplo, esta memoria sería la recomendada, dada su mayor capacidad de almacenamiento. Para que su funcionamiento sea correcto, su dirección I2C no deberá coincidir con la del PCF8583, así que se usarán las patillas de direccionamiento para distinguirlo de la dirección de este otro dispositivo.

Como sensor de temperatura, emplearemos un dispositivo 1-wire, el DS18S20, que es el que se empleó en prácticas anteriores.

Para la interfaz de usuario, emplearemos 4 pulsadores, un display LCD que conectamos como se especificó en prácticas anteriores, un teclado matricial de 4x3 y una terminal virtual para simular la conexión RS-232 (que únicamente mostrará unos valores por pantalla y no permitirá la interacción del usuario con el sistema).

El resto del hardware corresponderá a resistencias de pullup y lectores de valores analógicos (potenciómetros), aparte de especificar las salidas hacia los dosificadores, el calentador del agua y la depuradora, que se simularán con LEDs, y la depuradora con un motor de corriente continua.

A continuación se adjunta el esquema empleado en el sistema para la simulación en PROTEUS:

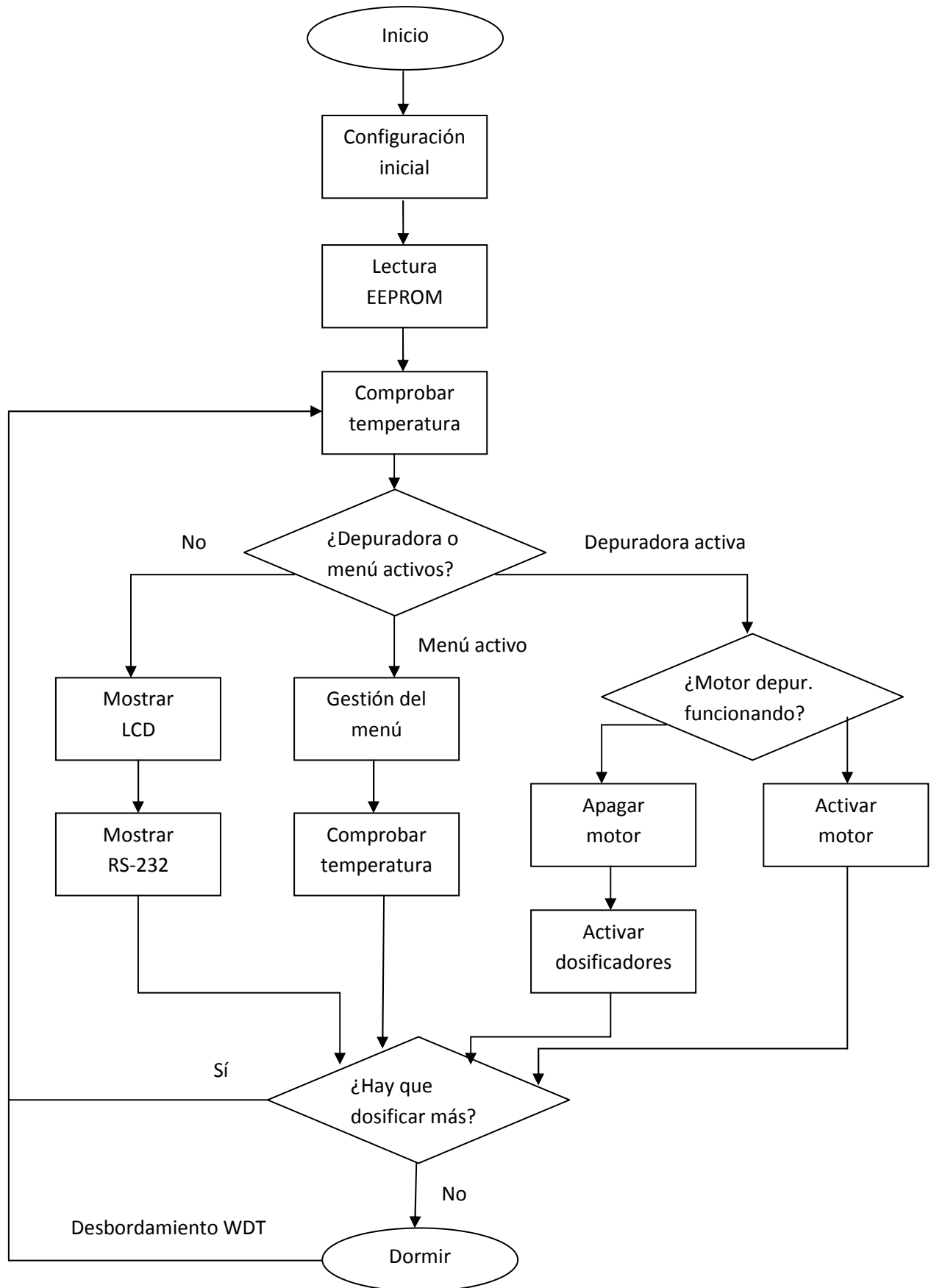


### 3.- Descripción software

El software del programa ha sido organizado en diversos archivos ".c". En primer lugar, el fichero principal del comportamiento del programa es "piscina.c", que será el que contiene el método main que se ejecutará en el programa. El primer fichero que cargará este fichero, será "datos\_funciones.c", que es un fichero auxiliar en el que definimos las variables y funciones que se emplearán, para así poder referenciar funciones y datos entre sí sin necesidad de haber definido el comportamiento de los métodos en líneas secuenciales de código en C, debido a la forma de funcionar de este lenguaje de programación.

Además, definimos otros ficheros para las funciones de lectura y escritura en EEPROM y para la interfaz del menú de usuario. Además, se incluyen los ficheros que describen las funciones de los demás dispositivos hardware empleados, haciendo algunas modificaciones en el fichero del PCF8583 y del teclado matricial, para adaptarlos a este sistema.

A continuación se introducirá el organigrama del comportamiento del sistema, para después describir el comportamiento de las principales funciones del proyecto:



En primer lugar, tras hacer la configuración inicial de los parámetros del sistema (entre los que se encuentra la hora actual), se procede a leer la memoria EEPROM, en primer lugar obteniendo los valores que debería haber en las posiciones donde se incluiría el flag de inicio de la secuencia de datos en EEPROM. De esta forma, según estén los flags correspondientes o no, se leerá la memoria interna, la externa I2C o se cargarán unos valores predeterminados.

A continuación ya se entraría en el bucle principal del programa, en la parte que se ejecutaría siempre. Lo primero que se haría sería hacer la comprobación de temperatura, que simplemente comprobaría si la temperatura alcanza cierto valor, y en caso de no alcanzarla, activaría la salida correspondiente al calentador del agua. El agua estará calentándose mientras que no se haya llegado a la temperatura deseada, y esta comprobación en condiciones de normalidad se haría siempre que se desbordara el perro guardián, configurado para que lo haga cada algo más de 2 minutos. También se llegará a este método cuando salten interrupciones externas (botón del menú o interrupción del reloj-calendario) o cuando se haya terminado de gestionar el menú de usuario.

En caso de que la depuradora no tenga que activarse o no se haya pulsado el botón que activa el menú, se procederá a mostrar datos por la pantalla del LCD. En este caso, primero mostrará los niveles de cloro y turbidez leídos, y un segundo después la temperatura actual y la hora a la que ha hecho esta comprobación. Este último dato será el que se quedará en la pantalla cuando el PIC se quede durmiendo. A continuación además, se mostrarán datos por la terminal conectada mediante RS-232, relativos a las variables configuradas con el menú y que determinarán el comportamiento del sistema. Para modificar estas variables no obstante habrá que recurrir al menú de usuario mediante pulsadores.

En caso de haber pulsado la tecla del menú, se activará un flag que hará que luego en el bucle principal del programa se pase a gestionar la interfaz de usuario, que está incluida en un fichero aparte. Esto consta de diversos métodos que pedirán interacción del usuario empleando 4 botones y mostrando todos los datos por la pantalla LCD. Se permite introducir valores numéricos empleando un teclado matricial. En cada bucle ejecutado en el menú, habrá una comprobación acerca de que haya que encender o apagar el menú de la depuradora y sobre si debe o no encender la salida de calentamiento del agua. Esto se hace así para que en caso de estar mucho tiempo en el menú, el programa siga funcionando correctamente.

Cuando salte la alarma correspondiente del PCF8583 significará que debe encenderse o bien apagarse el motor de la depuradora, según estuviera o no funcionando anteriormente. Esto una vez más se hará activando un flag en la interrupción externa y luego ejecutándolo en el bucle principal de ejecución del programa. Se hará la comprobación correspondiente, se activará o apagará el motor y se cargará en el PCF8583 la hora de inicio o de fin del ciclo de la depuradora, según cuál sea el próximo evento relacionado con ello que debería ocurrir. Además, cuando acabe el ciclo de funcionamiento de la depuradora (es decir, cuando vaya a apagarse el motor), se activarán los dosificadores de cloro y floculante.

La dosificación de estos productos se hará mediante unas variables definidas en el programa en general, pero que solo tomarán un valor cuando el motor de la depuradora se apague. Entonces, se calcularán las diferencias de nivel de cloro y turbidez entre el deseado y el medido, y se guardarán en esta variable. Entonces, si esta diferencia es positiva, significará que

hay que dosificar, y entonces se hará una nueva comprobación cada 0.2 segundos, que será cuando se desborde el timer1. Cada vez que se desborde el timer1, se restará 0.1 al valor de diferencia analógica medido para estas variables. Así, el tiempo de dosificación tomará un valor entre 0 y 10 segundos, según esta diferencia sea mayor o menor. Una vez se detecte que no hay que seguir dosificando, se desactivarán las correspondientes salidas y el PIC podrá volver a dormir hasta que se desborde el WDT o se active una interrupción externa.

Todas las variables y funciones que se emplean para llevar a cabo estas funcionalidades, están definidas en el fichero "datos\_funciones.c".

#### **4.- Funcionalidad**

En primer lugar, el programa se cargará con la hora actual, que deberá ser introducida por el usuario empleando el teclado matricial. Una vez introducida esta hora, el programa comenzará a funcionar y la hora no podrá volver a ser modificada. El programa entonces mostrará los niveles medidos de cloro y turbidez, mostrará la temperatura del agua y la hora a la que hizo esta medición, mostrará los parámetros leídos de EEPROM por la terminal RS-232 y pasará a modo de bajo consumo.

Entonces, el usuario simplemente podrá interactuar con el programa activando el menú de usuario. Para ello, deberá pulsar la tecla correspondiente de abrir el menú. Entonces, puede alternar entre 4 opciones diferentes usando la tecla de "Siguiete" y seleccionando la opción que se desee modificar con "Aceptar". En la opción "Depuradora", habrá 2 posibles horas que modificar (la de inicio y fin del ciclo de funcionamiento de la depuradora), por lo que al entrar en este menú se puede alternar entre cualquiera de estas dos horas con la opción "Siguiete". Una vez se ha especificado alguna de estas opciones, se introducen tantos números como se pide empleando el teclado matricial, y a continuación se confirma con el botón "Aceptar".

Para salir de cualquier opción, se debe usar el botón "Salir", que cuando llega a cierto nivel, permite efectivamente salir del menú de usuario del sistema.

#### **5.- Problemas encontrados y soluciones aportadas**

En primer lugar, el problema que más tiempo consumió en la resolución del problema, fue el error del PCF8583 que hace que sin razón aparente se detenga la cuenta del tiempo en el reloj-calendario y de ese modo queden deshabilitadas las funcionalidades de la depuradora, que dependen de la alarma del PCF. Una solución primaria fue hacer que el PCF no cambiara de año en la simulación (en la hora que se introdujo inicialmente para las cuentas sí lo hacía), de modo que no se paraba con tanta frecuencia. Sin embargo, el PCF siguió deteniéndose sin motivo aparente, y un simple cambio de hora permitía que esto funcionara por más tiempo con normalidad. Esto resultó ser un error del modelo de Proteus, pero considero que debería avisarse previamente de este erróneo comportamiento para no perder demasiado tiempo en ello y que si fuera posible debería solventarse este error de funcionamiento buscando un modelo más correcto de Proteus o un dispositivo reloj-calendario con alarma que sí esté bien simulado en Proteus.

Además, inicialmente traté de implementar el control de la temperatura cada cierto tiempo (unos 15 minutos) empleando la funcionalidad de temporizador del PCF, pero no activaba los flags correspondientes y su funcionamiento no resultó ser el obtenido en los esquemas de hoja de características. Por tanto, se optó por emplear el watchdog como era más habitual, de esta manera comprobando la temperatura en cada desborde, que es aproximadamente cada algo más de 2 minutos al emplear el prescaler correspondiente en el WDT.

También se tuvo un problema al emplear el direccionamiento que se usó en las prácticas para la memoria EEPROM I2C, que resultó ser la misma que la del PCF8583, así que hubo que redireccionar este dispositivo con sus patillas correspondientes para que no se modificaran ambos dispositivos al escribir en I2C por alguno de ellos.

Para que el control de temperatura y de la depuradora se realizara incluso cuando se está navegando por el menú, se tuvo que incluir estas comprobaciones en todo bucle que hubiera en cada opción del menú, ya que de lo contrario, se encontraba el problema de que este control no se activara hasta que parara de navegarse por el menú.

Además, inicialmente en las pruebas se introducía una hora mediante el código en el PCF. Para que automáticamente eligiera la hora del sistema, no le di el código correspondiente a introducir la hora al PCF, por lo que cogía la hora predefinida del sistema, pero de algún modo el modelo de Proteus funcionaba incorrectamente (no mostrando ciertas ventanas como la terminal o la hora del PCF), por lo que consideré más adecuado introducir una hora manualmente mediante el teclado matricial al iniciar el programa.

Por el modo de funcionar de C (leyendo todo el código secuencialmente), no podía emplear funciones que referenciaran a funciones o datos que se definían posteriormente, por lo que tenía que seguir un orden concreto. Sin embargo, esto a veces no era posible, así que había que optar por definir funciones vacías en el principio del código. Por eficiencia, decidí crear un fichero únicamente para esto, de modo que los demás ficheros podían referenciar a cualquier función o variable sin problemas.

El resto de problemas encontrados no consumió tanto tiempo, más que el típico cuando se afronta un problema de diseño de software.

## **6.- Presupuesto**

En el presupuesto, vamos a suponer que ya tenemos comprados tanto el motor de la depuradora, el calentador del agua, los dosificadores de cloro y floculante, así como los sensores que nos hacen estas mediciones. Únicamente se incluyen los componentes relativos al control del sistema, es decir, a lo que se ha especificado en el proyecto en concreto. Sin contar tampoco el dinero gastado en pagar a los operarios que diseñan y montan el sistema, el presupuesto aproximado sería el siguiente:

- Microcontrolador PIC 18F4520	8.50 €
- Reloj-calendario PCF8583	2.82 €
- Sensor temperatura DS1820	1.18 €
- Memoria EEPROM 24C32A	1.30 €
- Display LCD LM016L	12.66 €
- 4 pulsadores	1.60 €
- Teclado matricial	4.25 €
- 12 resistencias de pullup (10kΩ)	0.36 €
<b>TOTAL</b>	<b>32.67 €</b>

La mayor parte de estos precios ha sido comprobada en páginas de venta en Internet como ebay, por lo que podrían variar bastante del encontrado al comprar en tiendas especializadas al implementar el proyecto de forma real.

## 7.- Planificación

El proyecto podría descomponerse en distintas tareas diferenciadas, que serían más o menos en las mismas en las que me estructuré al trabajar en él:

- Elección de hardware y conexiones para implementar las funciones del sistema
- Control de las alarmas para funcionamiento de la depuradora
- Control de temperatura, cloro y turbidez del agua
- Interfaz de usuario mediante pulsadores
- Interacción con la memoria EEPROM
- Montaje del sistema en la piscina

Para el desarrollo del proyecto, probablemente estimaba gastar unas 40 horas aproximadamente, pero debido a diversos problemas encontrados (principalmente el que se detuviera el PCF sin motivo aparente) gasté demasiado tiempo sin hacer muchos avances. Podría estimar que al final el proyecto me llevó algo más de 50 horas de tiempo, aunque parte de este tiempo no fuera dedicado exclusivamente a trabajo en el código, pero sí a estar dedicando el tiempo al proyecto y no a otras cosas.