

Desarrollo de un Reloj Digital con una Raspberry Pi Pico

Creación de Prototipos de IoT con Raspberry Pi

Índice

1. Antecedentes.....	2
2. Objetivo.....	3
3. Hardware empleado.....	3
4. I2C.....	5
5. Montaje Físico PASO A PASO.....	6
6. Montaje físico con FRITZING.....	8
7. Código Arduino.....	8
8. Librerías Empleadas.....	13
9. Principales Desafíos y Problemas Encarados en el Proyecto del Reloj Digital.....	14
10. Conclusión del Proyecto de Reloj Digital con Raspberry Pi.....	15
11. Agradecimientos Especiales.....	15

1. Antecedentes

Dentro del marco de nuestro curso de *Creación de Prototipos de IoT con Raspberry*, hemos emprendido el desarrollo de un reloj digital mediante la plataforma Raspberry Pi. La elección de este proyecto se fundamenta en la oportunidad de adquirir competencias técnicas sólidas en la integración de componentes tecnológicos de relevancia en la esfera de *IoT*, en particular, la comunicación *I2C* y el *Real-Time Clock (RTC)*.

Nuestro enfoque se basa en la comprensión y aplicación práctica de la funcionalidad de los relojes digitales y su implementación a través de una *Raspberry Pi Pico*. Este proyecto nos proporciona una plataforma idónea para explorar en detalle la arquitectura y operatividad de los relojes digitales, al tiempo que nos permite incorporar habilidades técnicas valiosas en la gestión de la comunicación *I2C* y el aprovechamiento del *RTC*. En resumen, la ejecución de este proyecto representa un hito significativo en nuestro proceso educativo, enriqueciendo nuestra comprensión y competencia en un ámbito tecnológico en constante progresión.

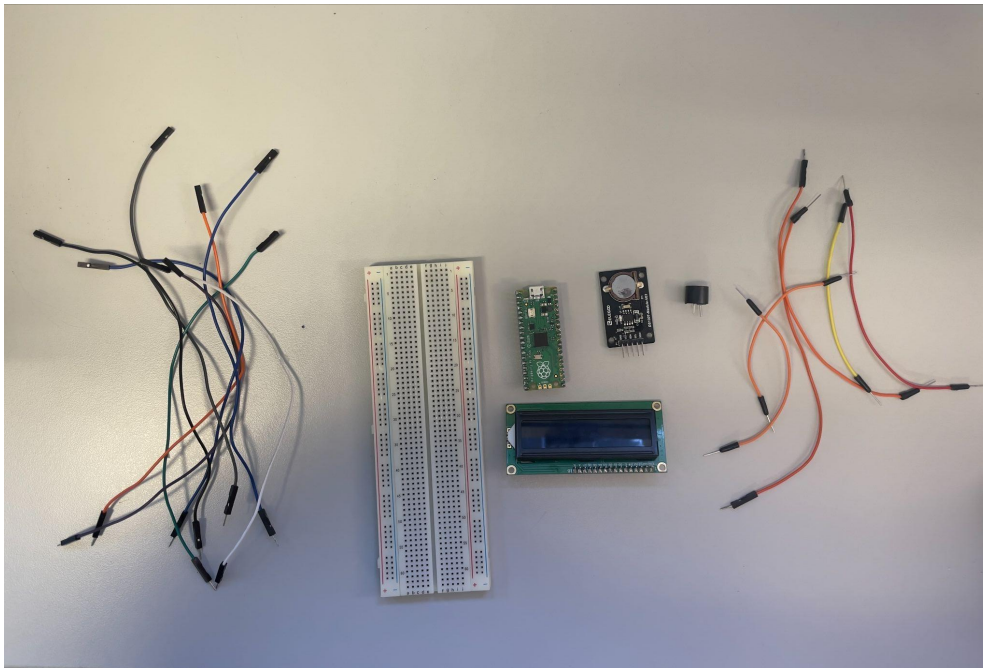
2. Objetivo

El objetivo fundamental de este proyecto es diseñar y crear un reloj digital plenamente funcional aprovechando la plataforma Raspberry Pi. Este proyecto se centra en la integración y comprensión de componentes tecnológicos esenciales, incluida la comunicación *I2C* y el *Real-Time Clock (RTC)*.

Nuestra meta principal consiste en desarrollar un reloj digital preciso que pueda mostrar la hora en una pantalla LCD y, adicionalmente, activar una alarma al producirse cada cambio de hora. A través de este proyecto, aspiramos a ampliar nuestros conocimientos y habilidades en la programación de Raspberry Pi, la interacción de componentes electrónicos y la gestión efectiva del tiempo en el contexto de *IoT*.

Además, buscamos mejorar nuestra capacidad para aplicar conceptos técnicos en una aplicación concreta, así como comprender la manera en que los dispositivos electrónicos se comunican y se integran para lograr un resultado funcional. Este proyecto representa una valiosa oportunidad para adquirir habilidades técnicas esenciales y profundizar nuestra comprensión en el campo de la tecnología IoT.

3. Hardware empleado



- Cables Dupont M-H (x8)
- BreadBoard (x1)
- Raspberry Pi Pico (x1)
- LCD QAPASS (x1)
- Real Time Clock "RTC" (x1)
- Buzzer (x1)
- Cables Dupont M-M (x6)

RTC

Un RTC (Real-Time Clock, en español "Reloj en Tiempo Real") es un dispositivo electrónico que se utiliza para llevar un registro preciso del tiempo, la fecha y, a menudo, la información de la hora actual en un sistema informático o

dispositivo. A diferencia de los relojes tradicionales, los RTCs son componentes electrónicos que utilizan una batería para mantener la hora en todo momento, incluso cuando el dispositivo se encuentra apagado o sin suministro de energía externa.

Estos dispositivos son esenciales en aplicaciones que requieren un seguimiento preciso del tiempo, como relojes digitales, sistemas de registro de datos, sistemas de alarma y dispositivos IoT (Internet de las Cosas), donde se necesita tiempo real para coordinar eventos o procesos.

LCD

Un LCD (Liquid Crystal Display, en español "Pantalla de Cristal Líquido") es una tecnología de pantalla que se utiliza ampliamente en dispositivos electrónicos, como televisores, monitores de computadora, teléfonos móviles y muchos otros dispositivos.

La tecnología LCD se basa en la capacidad de los cristales líquidos para modificar la cantidad de luz que pasa a través de ellos al aplicar una corriente eléctrica. Los cristales líquidos se encuentran entre dos capas de vidrio o plástico transparente y están dispuestos en una matriz de píxeles. Cada píxel se controla individualmente para permitir la representación de imágenes y texto en la pantalla.

4. I2C

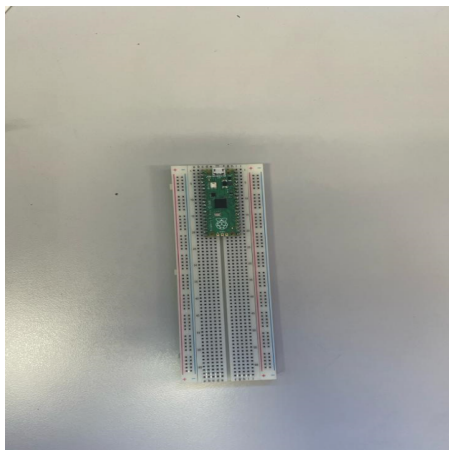
I2C, que significa Inter-Integrated Circuit, es un protocolo de comunicación serie que se utiliza para conectar y permitir la transferencia de datos entre diferentes componentes electrónicos en un sistema. Fue desarrollado originalmente por Philips (hoy conocida como NXP Semiconductors) y se ha convertido en un estándar ampliamente utilizado en la industria electrónica.

El protocolo I2C se basa en una comunicación de bus de dos cables, uno para la transmisión de datos (SDA, Serial Data) y otro para la señal de reloj (SCL, Serial Clock). Permite que múltiples dispositivos se conecten al mismo bus I2C y se comuniquen entre sí de manera eficiente. Cada dispositivo en el bus I2C tiene una dirección única, lo que facilita la identificación de los dispositivos y la selección del destinatario de los datos transmitidos.

En el contexto de un reloj digital basado en una Raspberry Pi y otros dispositivos electrónicos, el protocolo I2C se utiliza para facilitar la comunicación y la interacción entre diferentes componentes del sistema. Aquí hay una explicación de cómo se usa el I2C en un reloj digital:

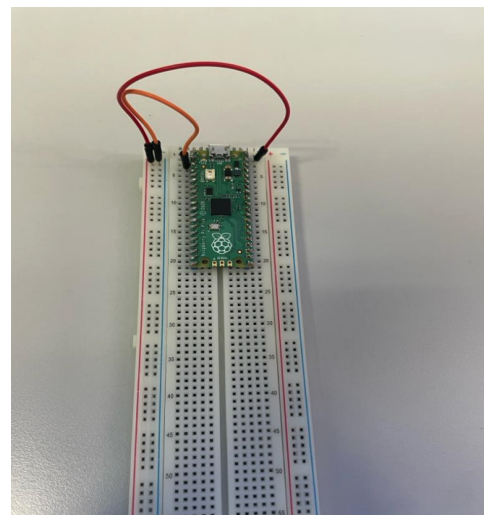
- Pantalla LCD: El reloj digital generalmente utiliza una pantalla LCD para mostrar la hora y, posiblemente, otras informaciones. Para controlar y enviar datos a la pantalla LCD, se utiliza una conexión I2C. El microcontrolador de la Raspberry Pi o un controlador dedicado envía comandos y datos a través del bus I2C al controlador de la pantalla LCD, que a su vez traduce esta información en lo que se muestra en la pantalla.
- Sensor de Tiempo Real (RTC): Aunque mencionaste que deseas excluir el RTC en tu descripción anterior, en muchos relojes digitales, especialmente aquellos que requieren una precisión en la hora incluso cuando la Raspberry Pi esté apagada, se emplea un RTC. El RTC se comunica con la Raspberry Pi a través del bus I2C y proporciona la hora actual incluso cuando la Raspberry Pi se apaga o reinicia. Esto garantiza que el reloj siempre muestre la hora correcta.

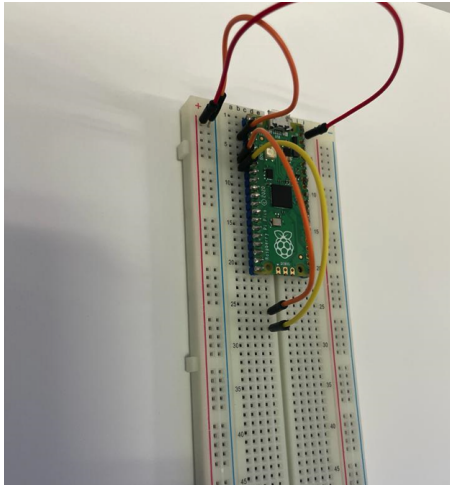
5. Montaje Físico PASO A PASO



1) CONECTAR LA RASPBERRY PI PICO AL BREADBOARD, PARA FACILITAR EL MONTAJE SITUAR LA ENTRADA DE DADOS AL BORDE DE LA PLACA.

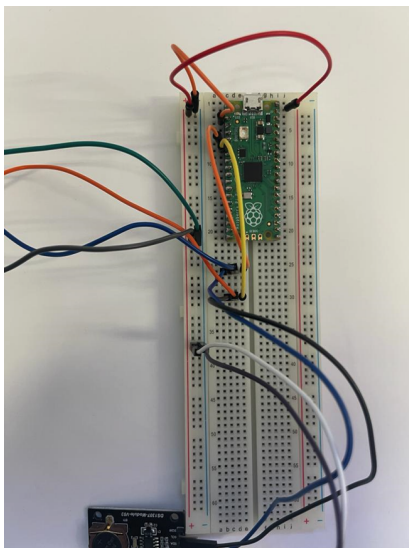
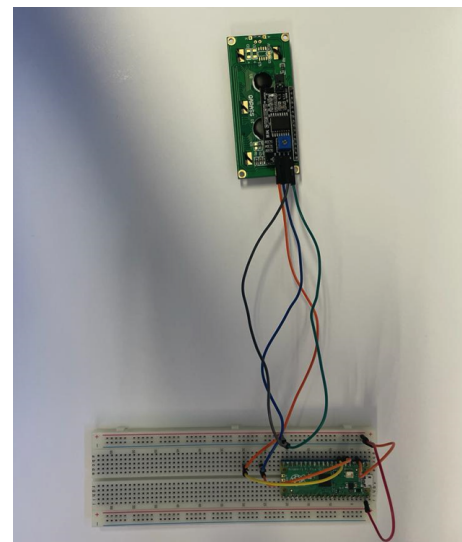
- 2) CONECTAR UN CABLE DUPONT M-M DESDE EL PIN 3V DE LA RASPBERRY PICO AL SIGNO (+) DE LA BOARD Y OTRO DESDE GROUND AL SIGNO (-) DE LA BOARD



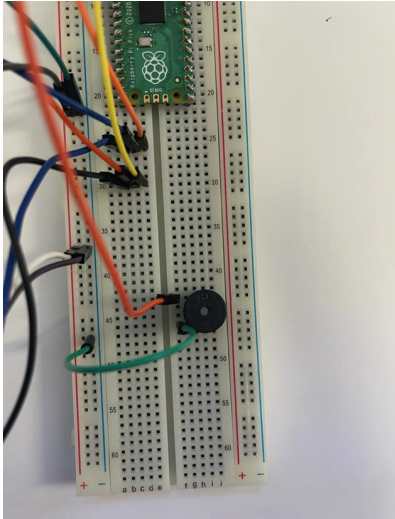


3) CONECTAR DOS CABLES M-M QUE SERVIRÁN COMO CABLES DE DATOS SDA Y SSL, LA RESPUESTA DE LA RASPBERRY NO ES ÓPTIMA PARA I2C SI ESTOS CABLES NO ESTÁN CONECTADOS AL PIN FÍSICO 6 Y 7 RESPECTIVAMENTE. POR LO TANTO CONECTAMOS EL SDA DEL PIN 6 A LA PLACA Y LO MISMO CON EL SSL

4) CONECTAR LOS 4 PINES DEL LCD CON CABLES M-H CON SUS RESPECTIVOS EN EL BOARD , SDA CON SDA, SSL CON SSL, VCC CON EL SIGNO (+) Y GND CON EL SIGNO (-)

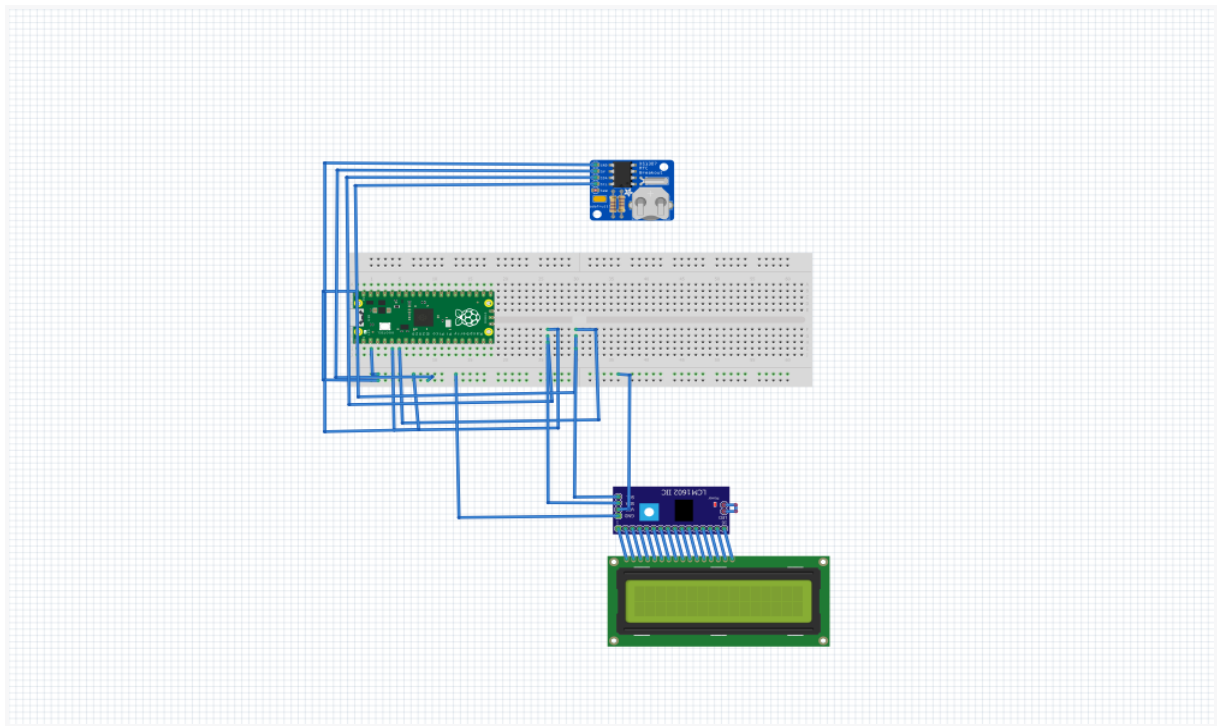


5) REALIZAR EL MISMO PROCEDIMIENTO QUE EL LCD CON EL RTC, MISMAS CONEXIONES!



6) CONECTAR EL BUZZER A LA PLACA, CON UN CABLE DUPONT M-M CONECTAR EL ÁNODO POSITIVO AL GPIO 28 DE LA RASPBERRY PICO Y EL ÁNODO NEGATIVO AL POLO (-) DE LA BREADBOARD

6. Montaje físico con FRITZING



Para facilitar el montaje y las conexiones, se ha realizado un montaje digital replicando todo el material y cableado necesario para la elaboración del reloj digital, el software empleado es FRITZING, que permite modelar y visualizar gráficamente el montaje.

7. Código Arduino

```
#include <LiquidCrystal_I2C.h>
```

```
#include <RTCLib.h>
```

```
#include <TaskScheduler.h>
```

```
Scheduler runner;
```

```
RTC_DS1307 rtc;
```

```
LiquidCrystal_I2C lcd(0x27, 16, 2);
```

```
int hora;
```

```
int minutos;
```

```
int segundos;
```

```
int buzzerpin = 28;
```

```
Task tasca1(0, TASK_FOREVER, []() {
```

```
DateTime now = rtc.now();

hora=now.hour();

minutos=now.minute();

segundos=now.second();

// Display the date & time on the LCD

lcd.setCursor(6, 1);

if (hora < 10) {

    lcd.print('0');

}

lcd.print(hora, DEC);

lcd.print(':');

if (minutos < 10) {

    lcd.print('0');

}

lcd.print(minutos, DEC);

lcd.print(':');

if (segundos < 10) {

    lcd.print('0');

}

lcd.print(segundos, DEC);

delay(1000);

});
```

```
Task tasca2(0, TASK_FOREVER, []() {  
    if (hora > 12) {  
        hora = hora % 12;  
    }  
    if (hora == 0){  
        hora = 12;  
    }  
    Serial.println(hora);  
    if ((minutos == 0)&&(segundos == 0)) {  
  
        for (int i = 0; i < hora; i++){  
            digitalWrite(buzzerpin, HIGH);  
            delay(500);  
            digitalWrite(buzzerpin, LOW);  
            delay(500);  
        }  
    }  
}  
);  
  
void setup() {  
    Serial.begin(115200);  
    pinMode(buzzerpin , OUTPUT );
```

```
// SETUP RTC MODULE
```

```
if (!rtc.begin()) {
```

```
    Serial.println("Couldn't find RTC");
```

```
    Serial.flush();
```

```
    while (1);
```

```
}
```

```
// automatically sets the RTC to the date & time on the PC this sketch was compiled
```

```
// rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
```

```
// manually sets the RTC with an explicit date & time, for example, to set
```

```
// January 21, 2021, at 3 am, you would call:
```

```
//rtc.adjust(DateTime(2021, 1, 21, 11, 59, 56));
```

```
lcd.init();    // Initialize the LCD
```

```
lcd.backlight(); // Turn on the backlight
```

```
lcd.setCursor(0, 0);
```

```
lcd.print("Time: ");
```

```
runner.init();
```

```
runner.addTask(tasca1);  
  
runner.addTask(tasca2);  
  
tasca1.enable();  
  
tasca2.enable();  
  
}
```

```
void loop(){  
  
    runner.execute();  
  
}
```

8. Librerías Empleadas

LiquidCrystal_I2C.h:

Uso en el Proyecto: La librería LiquidCrystal_I2C se utiliza en tu proyecto para facilitar la comunicación entre la Raspberry Pi y la pantalla LCD del reloj digital. Permite mostrar la hora y otros datos en la pantalla de cristal líquido de manera eficiente. Esta librería simplifica la gestión de la pantalla LCD y la visualización de información en ella a través de la comunicación I2C. Su uso posibilita una interfaz más amigable y una implementación más sencilla de la pantalla en el proyecto de reloj digital.

RTCLib.h:

Uso en el Proyecto: La librería RTCLib se utiliza en tu proyecto para gestionar el módulo de tiempo real (RTC) incorporado en el reloj digital. El RTC garantiza que el reloj siempre muestre la hora correcta, incluso cuando la Raspberry Pi se apaga o reinicia. La librería RTCLib permite establecer y leer la hora y la fecha del RTC, así como programar alarmas en momentos específicos, lo que es esencial para asegurar la precisión del reloj digital.

TaskScheduler.h:

Uso en el Proyecto: La librería TaskScheduler se emplea en tu proyecto para programar y gestionar tareas específicas del reloj digital. Permite la ejecución de actividades programadas, como la actualización de la hora en la pantalla LCD, la verificación de alarmas y otras tareas relacionadas con el funcionamiento del reloj. La librería TaskScheduler asegura que las tareas se ejecuten en el momento adecuado, lo que es esencial para mantener un funcionamiento preciso y confiable del reloj digital.

Estas librerías desempeñan un papel crucial en la implementación de las funciones esenciales del reloj digital, lo que garantiza la gestión eficiente de la pantalla LCD, la precisión en la medición del tiempo y la ejecución puntual de las tareas programadas.

9. Principales Desafíos y Problemas Encarados en el Proyecto del Reloj Digital

Durante el desarrollo de nuestro proyecto de reloj digital con Raspberry Pi, nos enfrentamos a varios desafíos significativos que requirieron soluciones técnicas y refinamiento del diseño. A continuación, se detallan los problemas más relevantes que encontramos:

- **Desconfiguración al Desconectar el Reloj:** Uno de los problemas iniciales que nos enfrentamos fue la desconfiguración del reloj cada vez que desconectamos la Raspberry Pi. Esto se debía a la pérdida de la hora y la fecha al apagar la Raspberry Pi, lo que afectaba la precisión del reloj. Para resolver este problema, incorporamos un módulo de Tiempo Real (RTC) que mantuvo un seguimiento continuo del tiempo, incluso cuando la Raspberry Pi no estaba en funcionamiento.
- **Hora Fija en el Reloj:** Otro desafío recurrente era la visualización de la misma hora en el reloj digital cada vez que lo iniciamos. Este problema se originaba en la falta de sincronización con un servidor de tiempo externo o una fuente precisa. Implementamos una sincronización periódica con una fuente de tiempo en línea, lo que garantizó que el reloj siempre mostrara la hora actual y precisa.
- **Número Permanente en la Pantalla:** Otra dificultad consistía en la visualización constante del mismo número en la pantalla del reloj digital. Esto se debía a un error en el código que no actualizaba adecuadamente los dígitos de la hora. Revisamos y corregimos el código para garantizar que los dígitos se actualizaran regularmente y mostrarán la hora correcta en tiempo real.

- **Problemas con la alarma:** El funcionamiento de la alarma también presentó desafíos. Inicialmente, la alarma solo emitía un "beep" cada hora en lugar de funcionar como un mecanismo de notificación efectivo. Investigamos y ajustamos el código de la alarma para garantizar que sonara en el momento deseado y con la duración y frecuencia adecuadas.
- **Fallo de la Alarma:** Adicionalmente, hubo una instancia en la que la alarma no sonaba en absoluto. Este problema se originó en un error de programación que impedía que la alarma se activara correctamente. Mediante pruebas y depuración, identificamos y corregimos el código relacionado con la alarma, asegurando que funcionara de manera confiable.

10. Conclusión del Proyecto de Reloj Digital con Raspberry Pi

El proyecto de desarrollo de un reloj digital con Raspberry Pi ha sido una valiosa experiencia que ha llevado a la creación de un dispositivo funcional y preciso. A lo largo de este proyecto, nos enfrentamos a diversos desafíos técnicos que nos exigieron investigar, depurar y afinar constantemente nuestro diseño. Al superar estos obstáculos, adquirimos conocimientos significativos y habilidades técnicas esenciales que enriquecieron nuestra comprensión en el ámbito de la tecnología y la programación.

La incorporación de un módulo de Tiempo Real (RTC) resolvió de manera efectiva el problema de la desconfiguración del reloj al desconectar la Raspberry Pi, asegurando que el reloj mantuviera la hora correcta incluso en circunstancias de falta de energía. Además, establecimos una sincronización con una fuente de tiempo en línea para garantizar que el reloj siempre marcara la hora actual y precisa.

La resolución de problemas relacionados con la visualización de la hora, la operación de la alarma y la corrección de errores en el código contribuyeron a un funcionamiento más confiable y eficiente del reloj digital. Estos desafíos también nos permitieron profundizar en la depuración y la mejora continua del código, competencias esenciales en la programación.

En última instancia, el proyecto culminó en la creación de un reloj digital totalmente funcional que satisface nuestros objetivos iniciales. Esta experiencia no solo amplió nuestra comprensión de la tecnología y la electrónica, sino que también subraya la importancia de la perseverancia y la resolución de problemas en el proceso de desarrollo de proyectos tecnológicos. El reloj digital no solo representa un logro técnico, sino también un testimonio de nuestro crecimiento y aprendizaje en este emocionante campo de la programación.

11. Agradecimientos Especiales

Queremos expresar nuestro más sincero agradecimiento a las siguientes personas que desempeñaron un papel fundamental en la realización de nuestro proyecto:

Profesor Joan Masdemont: Agradecemos al profesor Joan Masdemont por su invaluable orientación y apoyo en la elaboración del código, así como por proporcionar asistencia técnica esencial a lo largo de todo el proyecto. Su experiencia y conocimiento fueron fundamentales para superar desafíos técnicos y lograr el éxito en nuestro empeño.

Guillermo León: Extendemos nuestro agradecimiento a Guillermo León por su mirada crítica y su valiosa retroalimentación. Su enfoque analítico y perspicaz fue fundamental para identificar áreas de mejora y refinamiento en nuestro proyecto, lo que contribuyó significativamente a su éxito final.

Estamos profundamente agradecidos por el apoyo inquebrantable de estas personas y todos nuestros compañeros de clase, cuya contribución desinteresada y dedicación ayudó a dar forma a nuestro proyecto y a enriquecer nuestra experiencia de aprendizaje. Sus consejos y guía fueron invaluable en nuestro camino hacia el éxito.

