



**Universidad  
de Jaén**

Escuela Politécnica Superior  
de Linares

## Trabajo Fin de Grado

# **ASISTENTE DIGITAL PARA EL FOMENTO DE LA LECTURA DE LIBROS MEDIANTE GAMIFICACIÓN**

**Alumno:** Ainhoa Miró Mata

**Tutor:** Prof. D. Luis Ramón López López  
**Depto.:** Ingeniería de Telecomunicación

**Junio, 2024**



**Universidad  
de Jaén**

Escuela Politécnica Superior  
de Linares

# **ASISTENTE DIGITAL PARA EL FOMENTO DE LA LECTURA DE LIBROS MEDIANTE GAMIFICACIÓN**

**Alumno:** Ainhoa Miró Mata

**Tutor:** Prof. D. Luis Ramón López López

**Dpto.:** Ingeniería de Telecomunicación

**Junio, 2024**

Firma del alumno:

Firma del tutor:

## AGRADECIMIENTOS

En primer lugar, me gustaría dar las gracias a mi familia y a mi pareja Miguel, porque han sido mi pilar fundamental todos estos años. Muchas gracias papá y mamá por haberme dado la vida y por todo el cariño, amor y consejos que me habéis ofrecido siempre, os lo debo todo. Muchas gracias Álvaro y Mario, porque a pesar de ser unos terremotos a veces, sois el regalo más bonito de mi vida desde hace ya 19 años, os cuidaré y protegeré siempre. Muchas gracias Miguel por caminar siempre a mi lado y por ofrecerme tu mano siempre que lo he necesitado, te quiero mucho.

Por supuesto, también muchas gracias al resto de mi familia. Titos, primos, abuelas, gracias por los buenos momentos en familia que siempre hemos pasado y por el apoyo incondicional.

Abuelo Pepe y abuelo Manuel también os doy las gracias a vosotros, aunque ya no podéis verme, sé que estaríais muy orgullosos de mí, soy quien soy en parte por el cariño y el legado que me habéis dejado. Abuelo Manuel, a ti quiero dedicarte toda mi carrera, seguro que desde el cielo miras con ojos orgullosos lo que he conseguido.

También quiero dar las gracias a todos los amigos que conocí aquel primer curso, lleno de incertidumbre, con los que he caminado a lo largo de estos cuatro años. Sin vosotros nada hubiera sido igual, me habéis alegrado la vida y entre todos en equipo estamos alcanzando nuestras metas. También dar infinitas gracias a los amigos que me acompañaron durante mi estancia Erasmus, os habéis convertido en una segunda familia.

Y, por último, pero no por ello menos importante, muchas gracias a todo el personal de la Escuela Politécnica Superior de Linares. Gracias a mi tutor Luis Ramón, por ayudarme, enseñarme y animarme siempre con una sonrisa en todo momento, a pesar de las mil veces que aparecía en tutoría. Muchas gracias también a Pedro Aguilar, siempre que lo he necesitado me has ayudado de la mejor manera, eres esencial para todos, aportas una gran profesionalidad y bondad a esta escuela. Y gracias al resto de profesores que me han guiado en las enseñanzas del grado y de la vida misma.



# ÍNDICE

|                                                              |           |
|--------------------------------------------------------------|-----------|
| <b>1. RESUMEN</b>                                            | <b>15</b> |
| <b>2. INTRODUCCIÓN</b>                                       | <b>16</b> |
| 2.1. Motivación                                              | 16        |
| 2.2. Estructura del TFG                                      | 17        |
| <b>3. OBJETIVOS</b>                                          | <b>20</b> |
| <b>4. ESTADO DEL ARTE</b>                                    | <b>21</b> |
| 4.1. ESP32                                                   | 23        |
| 4.1.1. Espressif Systems                                     | 23        |
| 4.1.2. Inicio: ESP8266 y ESP32                               | 24        |
| 4.1.2.1. Comparativa ESP32 y ESP8266                         | 25        |
| 4.1.2.2. Dispositivos ESP32                                  | 26        |
| 4.1.2.3. Características y especificaciones ESP32            | 27        |
| <b>5. DISPOSITIVO Y COMPONENTES EMPLEADOS</b>                | <b>34</b> |
| 5.1. Comparación ESP32, ESP32-S2, ESP32-S3, ESP32-C3         | 34        |
| 5.2. Estudio previo elección placa desarrollo ESP32-S3       | 38        |
| 5.3. Desarrollo kit elegido y componentes                    | 42        |
| <b>6. PROGRAMACIÓN Y SOFTWARE</b>                            | <b>50</b> |
| 6.1. Programación del dispositivo                            | 50        |
| 6.1.1. Arduino                                               | 50        |
| 6.1.2. MicroPython                                           | 50        |
| 6.1.3. ESP-IDF                                               | 51        |
| 6.1.3.1. Capa de compatibilidad con Arduino                  | 53        |
| 6.1.4. Decisión sobre el lenguaje a utilizar                 | 53        |
| 6.2. Entorno de desarrollo                                   | 53        |
| 6.2.1. Características comunes                               | 53        |
| 6.2.2. Comparativa y elección de entorno                     | 54        |
| 6.3. PlatformIO                                              | 56        |
| 6.3.1. Introducción                                          | 56        |
| 6.3.2. Configuración                                         | 57        |
| <b>7. RECONOCIMIENTO Y DIFERENCIACIÓN DE LIBROS</b>          | <b>60</b> |
| 7.1. Planteamiento inicial y alternativa                     | 60        |
| 7.1.1. Problema de enfoque                                   | 60        |
| 7.1.2. Solución al problema de enfoque: código bidimensional | 61        |
| 7.1.2.1. Códigos bidimensionales                             | 61        |
| 7.1.2.2. Códigos QR                                          | 63        |
| 7.2. Conocer cómo se diferencian los libros                  | 64        |

|                                                                           |            |
|---------------------------------------------------------------------------|------------|
| 7.2.1. Tipos de códigos de barras.....                                    | 64         |
| 7.2.2. Cómo funciona el código de barras.....                             | 67         |
| 7.2.3. ISBN (EAN-13).....                                                 | 68         |
| <b>8. DISEÑO E IMPLEMENTACIÓN.....</b>                                    | <b>71</b>  |
| <b>8.1. Arquitectura general del software .....</b>                       | <b>71</b>  |
| 8.1.1. Librerías .....                                                    | 72         |
| 8.1.2. Código fuente .....                                                | 73         |
| <b>8.2. Interfaz de usuario .....</b>                                     | <b>74</b>  |
| 8.2.1. LVGL.....                                                          | 74         |
| 8.2.2. Funcionamiento general de la interfaz gráfica.....                 | 75         |
| 8.2.2.1. Introducción a la GUI.....                                       | 75         |
| 8.2.2.2. Gamificación.....                                                | 76         |
| 8.2.2.3. Realimentación al usuario .....                                  | 80         |
| 8.2.3. Pestaña 1 de la interfaz (Inicio) .....                            | 80         |
| 8.2.4. Pestaña 2 de la interfaz (Biblioteca).....                         | 84         |
| 8.2.5. Pestaña 3 de la interfaz (Reconocimiento del libro) .....          | 86         |
| 8.2.6. Pestaña 4 de la interfaz (Estadísticas) .....                      | 92         |
| <b>8.3. Base de datos .....</b>                                           | <b>96</b>  |
| 8.3.1. Introducción a Firebase.....                                       | 96         |
| 8.3.2. Formato de almacenamiento de los datos .....                       | 96         |
| 8.3.3. Creación de Realtime Database de Firebase .....                    | 98         |
| 8.3.4. Información a almacenar .....                                      | 100        |
| 8.3.4.1. General .....                                                    | 100        |
| 8.3.4.2. Para la gamificación .....                                       | 100        |
| 8.3.4.3. Marca temporal.....                                              | 101        |
| 8.3.5. Configuración a nivel de código .....                              | 101        |
| <b>8.4. Lectura y decodificación de QR .....</b>                          | <b>104</b> |
| 8.4.1. Creación del código .....                                          | 104        |
| 8.4.2. Instalación de la librería de decodificación y funcionamiento..... | 104        |
| <b>9. CONCLUSIONES Y RESULTADOS.....</b>                                  | <b>107</b> |
| <b>9.1. Resultados obtenidos .....</b>                                    | <b>107</b> |
| 9.1.1. Diseño del dispositivo .....                                       | 107        |
| 9.1.2. Reconocimiento del ejemplar de libro .....                         | 107        |
| 9.1.3. Métricas de fomento de la lectura (gamificación e interfaz) .....  | 107        |
| 9.1.4. Almacenamiento de la información .....                             | 108        |
| 9.1.5. Gamificación y realimentación.....                                 | 109        |

|                                                                                    |            |
|------------------------------------------------------------------------------------|------------|
| <b>9.2. Limitaciones.....</b>                                                      | <b>109</b> |
| 9.2.1. Lectura de QR y no de código de barras (ISBN) .....                         | 109        |
| 9.2.2. Mostrar imagen de cámara en pantalla y decodificar simultáneamente .....    | 109        |
| 9.2.3. Actualización de pantalla al cambiar datos desde la base de datos .....     | 110        |
| 9.2.4. Gestión de recursos en un sistema embebido.....                             | 110        |
| 9.2.5. Integración y sincronización de componentes de hardware .....               | 110        |
| 9.2.6. Creación de la interfaz gráfica de usuario (GUI) .....                      | 110        |
| 9.2.7. Conectividad WiFi y comunicación con la base de datos en la nube .....      | 110        |
| <b>10. LÍNEAS FUTURAS .....</b>                                                    | <b>112</b> |
| 10.1. Mejora en la captura y procesamiento de imágenes.....                        | 112        |
| 10.2. Nuevas funcionalidades para el usuario .....                                 | 113        |
| 10.3. Reconocimiento avanzado de libros.....                                       | 113        |
| 10.4. Interacción por voz con el usuario.....                                      | 114        |
| <b>11. BIBLIOGRAFÍA.....</b>                                                       | <b>115</b> |
| <b>12. ANEXOS.....</b>                                                             | <b>119</b> |
| 12.1. Manual de usuario.....                                                       | 119        |
| 12.1.1. Introducción .....                                                         | 119        |
| 12.1.2. Interfaz de Usuario.....                                                   | 119        |
| 12.1.2.1. Pestaña 1: Inicio y Gamificación.....                                    | 120        |
| 12.1.2.2. Pestaña 2: Biblioteca.....                                               | 121        |
| 12.1.2.3. Pestaña 3: Escaneo de QR y modificación de página actual del libro ..... | 121        |
| 12.1.2.4. Pestaña 4: Progreso del lector .....                                     | 123        |

## ÍNDICE DE FIGURAS

|                                                                                         |    |
|-----------------------------------------------------------------------------------------|----|
| Figura 2.1. Rendimiento en lectura en el año 2022 (1).....                              | 16 |
| Figura 2.2. Comparativa del rendimiento en lectura entre los años 2012 y 2022 (1) ..... | 16 |
| Figura 2.3. Motivos no lectura españoles en 2023 (2) .....                              | 17 |
|                                                                                         |    |
| Figura 4.1. Dispositivos fabricados hasta el momento .....                              | 24 |
| Figura 4.2. SoC ESP8266EX (13).....                                                     | 24 |
| Figura 4.3. Módulo ESP8266 (ESP8266MOD) (14) .....                                      | 25 |
| Figura 4.4. Placa de desarrollo que integra el módulo ESP8266MOD (15).....              | 25 |
| Figura 4.5. Bloques conectividad inalámbrica (16) .....                                 | 27 |
| Figura 4.6. Bloques procesador y coprocesador (16).....                                 | 28 |
| Figura 4.7. Bloques memorias (16) .....                                                 | 29 |
| Figura 4.8. Bloques periféricos (16) .....                                              | 31 |
| Figura 4.9. Bloques seguridad (16).....                                                 | 32 |
|                                                                                         |    |
| Figura 5.1. Diagrama de bloque funcional ESP32. (16) .....                              | 35 |
| Figura 5.2. Diagrama de bloque funcional ESP32-S2. (21).....                            | 35 |
| Figura 5.3. Diagrama de bloque funcional ESP32-S3. (22).....                            | 36 |
| Figura 5.4. Diagrama de bloque funcional ESP32-C3. (19).....                            | 36 |
| Figura 5.5. Diagrama de bloque funcional ESP32-C6. (20).....                            | 37 |
| Figura 5.6. Anverso placa del kit .....                                                 | 42 |
| Figura 5.7. Reverso placa del kit.....                                                  | 42 |
| Figura 5.8. Pines de la placa de desarrollo de Freenove (24) .....                      | 44 |
| Figura 5.9. Módulo ESP32-S3-WROOM-1 (23) .....                                          | 44 |
| Figura 5.10. Interfaces de ESP32-S3-WROOM-1 señaladas (24).....                         | 45 |
| Figura 5.11. Anverso placa desarrollo ESP32-S3 WROOM-1.....                             | 45 |
| Figura 5.12. Reverso placa desarrollo ESP32-S3 WROOM-1.....                             | 45 |
| Figura 5.13. Anverso pantalla.....                                                      | 46 |
| Figura 5.14. Reverso pantalla.....                                                      | 46 |
| Figura 5.15. Anverso cámara.....                                                        | 47 |
| Figura 5.16. Reverso cámara.....                                                        | 47 |
| Figura 5.17. Cable FPC cámara y placa extensora cámara .....                            | 47 |
| Figura 5.18. Anverso convertidor y amplificador de audio .....                          | 48 |
| Figura 5.19. Reverso convertidor y amplificador de audio .....                          | 48 |
| Figura 5.20. Altavoces .....                                                            | 48 |
| Figura 5.21. Anverso pulsómetro .....                                                   | 48 |
| Figura 5.22. Reverso pulsómetro .....                                                   | 48 |
| Figura 5.23. Cable de batería de 9 V .....                                              | 48 |
| Figura 5.24. Tarjeta SD y lector de tarjeta .....                                       | 49 |
| Figura 5.25. Tornillos y separadores de latón .....                                     | 49 |
|                                                                                         |    |
| Figura 6.1. Resultado del programa que se realizará.....                                | 54 |
| Figura 6.2. Programa “Hola Mundo” en Arduino IDE .....                                  | 55 |
| Figura 6.3. Programa “Hola Mundo” en Visual Studio Code.....                            | 56 |
| Figura 6.4. Programa “Hola Mundo” en CLion IDE.....                                     | 56 |

|                                                                                           |    |
|-------------------------------------------------------------------------------------------|----|
| Figura 6.5. Funciones de PlatformIO .....                                                 | 57 |
| Figura 7.1. Imagen tomada del ISBN con la cámara del ESP32-S3 .....                       | 60 |
| Figura 7.2. Diferencia entre almacenaje de datos entre QR y código de barras (34) .....   | 61 |
| Figura 7.3. Ejemplo código Data Matrix (36) .....                                         | 62 |
| Figura 7.4. Ejemplo código PDF 417 (36) .....                                             | 62 |
| Figura 7.5. Ejemplo código Aztec (36) .....                                               | 62 |
| Figura 7.6. Ejemplo código QR (36) .....                                                  | 63 |
| Figura 7.7. Composición QR versión 4, módulos 33x33 (34) .....                            | 63 |
| Figura 7.8. Ejemplo código EAN-13 (36) .....                                              | 65 |
| Figura 7.9. Ejemplo código UPC-A (36) .....                                               | 65 |
| Figura 7.10. Ejemplo código ITF-14 (36) .....                                             | 66 |
| Figura 7.11. Ejemplo Code 39 (36) .....                                                   | 66 |
| Figura 7.12. Ejemplo Codabar (36) .....                                                   | 66 |
| Figura 7.13. Ejemplo Code 128 (36) .....                                                  | 67 |
| Figura 7.14. Estructura general código de barras (39) .....                               | 68 |
| Figura 7.15. Estructura ISBN de 13 dígitos (41) .....                                     | 69 |
| Figura 7.16. Código de barras EAN-13 decodificado .....                                   | 70 |
| Figura 8.1. Esquema de la arquitectura general del sistema .....                          | 71 |
| Figura 8.2. Conjunto de librerías necesarias para el proyecto .....                       | 72 |
| Figura 8.3. Archivos de código fuente .....                                               | 73 |
| Figura 8.4. Principales técnicas para aplicar la gamificación (43) .....                  | 77 |
| Figura 8.5. Máximo nivel que se puede alcanzar (a partir de 1500 páginas) .....           | 78 |
| Figura 8.6. Felicitación motivadora tras terminar un libro .....                          | 79 |
| Figura 8.7. Pantalla principal de la pestaña 1 .....                                      | 81 |
| Figura 8.8. Pantalla secundaria explicativa de la pestaña 1 .....                         | 82 |
| Figura 8.9. Pantalla secundaria explicativa de la pestaña 1 .....                         | 82 |
| Figura 8.10. Pantalla secundaria explicativa de la pestaña 1 .....                        | 82 |
| Figura 8.11. Esquema de funcionamiento de la pestaña 1 .....                              | 83 |
| Figura 8.12. Pantalla principal de la pestaña 2 .....                                     | 84 |
| Figura 8.13. Pantalla secundaria de la pestaña 2 .....                                    | 85 |
| Figura 8.14. Pantalla secundaria de la pestaña 2 (título largo con scroll circular) ..... | 85 |
| Figura 8.15. Esquema de funcionamiento de la pestaña 2 .....                              | 86 |
| Figura 8.16. Pantalla principal de la pestaña 3 .....                                     | 87 |
| Figura 8.17. Pantalla secundaria de la pestaña 3 (escaneo de QR) .....                    | 88 |
| Figura 8.18. Tercera pantalla de la pestaña 3 .....                                       | 89 |
| Figura 8.19. Tercera pantalla de la pestaña 3 .....                                       | 89 |
| Figura 8.20. Tercera pantalla de la pestaña 3 .....                                       | 89 |
| Figura 8.21. Tercera pantalla de la pestaña 3 .....                                       | 90 |
| Figura 8.22. Esquema de funcionamiento de la pestaña 3 .....                              | 91 |
| Figura 8.23. Pantalla principal de la pestaña 4 .....                                     | 92 |
| Figura 8.24. Pantalla principal de la pestaña 4 .....                                     | 93 |
| Figura 8.25. Pantalla principal de la pestaña 4 .....                                     | 93 |
| Figura 8.26. Pantalla principal de la pestaña 4 .....                                     | 94 |
| Figura 8.27. Pantalla secundaria de la pestaña 4 .....                                    | 94 |
| Figura 8.28. Pantalla secundaria de la pestaña 4 .....                                    | 95 |

|                                                                                          |     |
|------------------------------------------------------------------------------------------|-----|
| Figura 8.29. Esquema de funcionamiento de la pestaña 4 .....                             | 95  |
| Figura 8.30. Esquema de comunicación con la base de datos (47).....                      | 96  |
| Figura 8.31. Ejemplo de JSON .....                                                       | 97  |
| Figura 8.32. JSON del proyecto completo .....                                            | 97  |
| Figura 8.33. Interfaz de Firebase donde se maneja el JSON .....                          | 98  |
| Figura 8.34. Cómo acceder a la consola de Firebase .....                                 | 98  |
| Figura 8.35. Creación de proyecto de Firebase.....                                       | 98  |
| Figura 8.36. Asignación de nombre descriptivo .....                                      | 99  |
| Figura 8.37. Reglas de seguridad para escritura y lectura de usuarios autenticados ..... | 99  |
| Figura 8.38. Instalación de la librería de Cliente Firebase para ESP32 .....             | 101 |
| Figura 8.39. Pasar de código de barras (ISBN) a QR que contiene el número del ISBN.....  | 104 |
| Figura 8.40. Librería para decodificar QR desde PlatformIO .....                         | 104 |
| Figura 8.41. Declaración del modelo de cámara empleado y sus pines .....                 | 105 |
|                                                                                          |     |
| Figura 12.1. Alimentación del dispositivo (24).....                                      | 119 |
| Figura 12.2. Pantalla principal de la pestaña 1 .....                                    | 120 |
| Figura 12.3. Pantalla secundaria explicativa de la pestaña 1.....                        | 120 |
| Figura 12.4. Pantalla secundaria explicativa de la pestaña 1.....                        | 120 |
| Figura 12.5. Pantalla secundaria explicativa de la pestaña 1.....                        | 120 |
| Figura 12.6. Pantalla principal de la pestaña 2 .....                                    | 121 |
| Figura 12.7. Pantalla secundaria de la pestaña 2.....                                    | 121 |
| Figura 12.8. Pantalla principal de la pestaña 3 .....                                    | 122 |
| Figura 12.9. Pantalla secundaria de la pestaña 3 (escaneo de QR) .....                   | 122 |
| Figura 12.10. Tercera pantalla de la pestaña 3 .....                                     | 123 |
| Figura 12.11. Tercera pantalla de la pestaña 3 .....                                     | 123 |
| Figura 12.12. Tercera pantalla de la pestaña 3 .....                                     | 123 |
| Figura 12.13. Tercera pantalla de la pestaña 3 .....                                     | 123 |
| Figura 12.14. Pantalla principal de la pestaña 4 .....                                   | 123 |
| Figura 12.15. Pantalla principal de la pestaña 4 .....                                   | 123 |
| Figura 12.16. Pantalla principal de la pestaña 4 .....                                   | 123 |
| Figura 12.17. Pantalla principal de la pestaña 4 .....                                   | 123 |
| Figura 12.18. Pantalla secundaria de la pestaña 4.....                                   | 124 |
| Figura 12.19. Pantalla secundaria de la pestaña 4.....                                   | 124 |

## ÍNDICE DE TABLAS

|                                                                                                                   |     |
|-------------------------------------------------------------------------------------------------------------------|-----|
| Tabla 4.1. Comparativa entre principales características de Raspberry Pi, ESP32 y Arduino UNO .....               | 22  |
| Tabla 4.2. Comparación entre ESP8266MOD y ESP32-WROOM-32 (3).....                                                 | 26  |
| Tabla 4.3. Chips, módulos y placas de desarrollo más comunes de ESP32.....                                        | 26  |
|                                                                                                                   |     |
| Tabla 5.1. Comparativa de las características técnicas distintos modelos de ESP32. (18; 19; 20; 21; 22; 16) ..... | 34  |
| Tabla 5.2. ESP32-S3 en Mouser.....                                                                                | 38  |
| Tabla 5.3. ESP32-S3 en DigiKey .....                                                                              | 39  |
| Tabla 5.4. ESP32-S3 en Aliexpress .....                                                                           | 39  |
| Tabla 5.5. ESP32-S3 en Amazon .....                                                                               | 40  |
| Tabla 5.6. ESP32-S3 en Elecrow .....                                                                              | 40  |
| Tabla 5.7. Características técnicas del módulo ESP32-S3-WROOM-1 (23) .....                                        | 43  |
| Tabla 5.8. Chip del kit del fabricante Freenove (22) .....                                                        | 43  |
| Tabla 5.9. Módulo del kit del fabricante Freenove (23).....                                                       | 43  |
| Tabla 5.10. Explicación interfaces señaladas (24).....                                                            | 45  |
|                                                                                                                   |     |
| Tabla 7.1. Codificación dígitos parte derecha e izquierda EAN-13 (42).....                                        | 70  |
| Tabla 7.2. Cálculo del primer número del código (42) .....                                                        | 70  |
|                                                                                                                   |     |
| Tabla 8.1. Breve resumen sobre el funcionamiento general de la interfaz .....                                     | 76  |
|                                                                                                                   |     |
| Tabla 10.1. Sensores de cámara compatibles con ESP32-S3 (54) .....                                                | 112 |

## ACRÓNIMOS

**ADC:** Conversión analógico a digital (*Analog to Digital Conversion*)

**AES:** Estándar de encriptación avanzado (*Advanced Encryption Standard*)

**AGC:** Control automático de ganancia (*Automatic Gain Control*)

**API:** Interfaz de programación de aplicaciones (*Application Programming Interface*)

**AIoT:** Inteligencia artificial de las cosas (*Artificial Intelligence of Things*)

**BLE:** Bluetooth de baja energía (*Bluetooth Low Energy*)

**CMOS:** Metal-óxido-semiconductor complementario (*Complementary Metal-Oxide-Semiconductor*)

**CPU:** Unidad central de procesamiento (*Central Processing Unit*)

**CRUD:** Crear, Leer, Actualizar y Borrar (*Create, Read, Update & Delete*)

**DAC:** Conversión digital a analógico (*Digital to Analog Conversion*)

**ECC:** Criptografía de curva elíptica (*Elliptic Curve Cryptography*)

**EAN:** Número de artículo europeo (*European Article Number*)

**ESO:** Educación Secundaria Obligatoria

**FIFO:** Primero que entra primero que sale (*First In First Out*)

**FPC:** Circuito impreso flexible (*Flexible Printed Circuit*)

**GPIO:** Entrada/salida de propósito general (*General Purpose Input/Output*)

**GUI:** Interfaz gráfica de usuario (*Graphical User Interface*)

**I<sup>2</sup>C:** Circuito inter-integrado (*Inter-Integrated Circuit*)

**I<sup>2</sup>S:** Sonido inter-circuitos integrados (*Inter-IC Sound*)

**IDE:** Entorno de desarrollo integrado (*Integrated Development Environment*)

**ISBN:** Estándar internacional de número de libro (*International Standard Book Number*)

**ITF:** Intercalado dos de cinco (*Interleaved Two of Five*)

**JSON:** Notación de objetos de JavaScript (*JavaScript Object Notation*)

**JPEG:** Grupo conjunto de expertos en fotografía (*Joint Photographic Experts Group*)

**LCD:** Pantalla de cristal líquido (*Liquid Crystal Display*)

**LED:** Diodo emisor de luz (*Light Emitting Diode*)

**LVGL:** Biblioteca gráfica ligera y versátil (*Light and Versatile Graphics Library*)

**MAC:** Control de acceso al medio (*Media Access Control*)

**MMC:** Tarjeta multimedia (*MultiMedia Card*)

**OCDE:** Organización para la Cooperación y el Desarrollo Económico

**PC:** Ordenador personal (*Personal Computer*)

**PDF:** Formato de documento portable (*Portable Document Format*)

**PNG:** Gráficos portables en red (*Portable Network Graphics*)

**PSRAM:** Memoria de Acceso Aleatorio Pseudoestática (*Pseudo Static Random Access Memory*)

**PWM:** Modulación por ancho de pulso (*Pulse Width Modulation*)

**QR:** Rápida respuesta (*Quick Response*)

**RAM:** Memoria de acceso aleatorio (*Random Access Memory*)

**RNG:** Generador de números aleatorios (*Random Number Generator*)

**RGB:** Rojo-verde-azul (*Red-Green-Blue*)

**RF:** Frecuencia radio (*Radio Frequency*)

**ROM:** Memoria de solo lectura (*Read Only Memory*)

**RTC:** Reloj en tiempo real (*Real Time Clock*)

**RSA:** Rivest-Shamir-Adleman

**SD:** Seguro digital (*Secure Digital*)

**SDIO:** Entrada/salida digital segura (*Secure Digital Input/Output*)

**SHA:** Algoritmo de Hash Seguro (*Secure Hash Algorithm*)

**SoC:** Sistema en un chip (*System on Chip*)

**SPI:** Interfaz periférica serial (*Serial Peripheral Interface*)

**SRAM:** Memoria de acceso aleatorio estática (*Static Random Access Memory*)

**TCP/IP:** Protocolo de control de transmisión/Protocolo de internet (*Transmission Control Protocol/Internet Protocol*)

**TFT:** Transistor de película fina (*Thin Film Transistor*)

**TFG:** Trabajo Fin de Grado

**ULP:** Consumo de energía ultra bajo (*Ultra Low Power*)

**UART:** Receptor/transmisor universal asíncrono (*Universal Asynchronous Receiver Transmitter*)

**UPC:** Código universal de producto (*Universal Product Code*)

**URL:** Localizador uniforme de recursos (*Uniform Resource Locator*)

**USB:** Bus universal en serie (*Universal Serial Bus*)

**UXGA:** Matriz gráfica ultra extendida (*Ultra Extended Graphics Array*)

## 1. RESUMEN

La lectura es una herramienta esencial para el desarrollo de las personas. Un buen hábito de lectura favorece a la mejora del lenguaje y del conocimiento.

Además, se ha comprobado que tener costumbre de leer aporta muchos otros beneficios, que pueden pasar desapercibidos, como, por ejemplo: aumenta la curiosidad y la imaginación, alimenta la inspiración, mejora la gramática y vocabulario, facilita la comunicación, permite relajación y entretenimiento, etc.

Por todos estos motivos y debido a la falta de interés generalizada en la lectura, especialmente en jóvenes, en este Trabajo Fin de Grado (TFG) se desarrollará un dispositivo que fomente el disfrute de los libros mediante el uso de la gamificación, desarrollando así una serie de herramientas con para conseguir una educación de calidad.

El dispositivo estará provisto de una pantalla y cámara que reconocerá un ejemplar de libro, facilitando el registro del avance del lector y permitiéndole obtener datos estadísticos informando sobre la evolución del ritmo de lectura y estableciendo comparativas.

De esta manera el usuario podrá registrar sus lecturas de una manera eficaz e interactiva, navegando entre los distintos menús de la interfaz gráfica en función de la actividad que desee realizar.

## 2. INTRODUCCIÓN

### 2.1. Motivación

El informe PISA (*Programme for International Student Assessment*) es un estudio internacional llevado a cabo que realiza la Organización para la Cooperación y el Desarrollo Económico (OCDE). Este estudio se realiza a los estudiantes de entre 15 y 16 años tras finalizar su Educación Secundaria Obligatoria (ESO).

Se les evalúa sobre competencias en matemáticas, lectura y ciencias. Concretamente, en este TFG se analizarán los datos relativos a la lectura.

La información arrojada por este estudio permite observar que desde el año 2012 aproximadamente, el rendimiento desciende de manera significativa, a pesar de que España se sitúa en la media en comparativa con los países participantes en este informe. (1)

Debido a estos datos se llega a la conclusión del deber de implantar nuevas alternativas para lograr el correcto desarrollo en este ámbito de los estudiantes.



Figura 2.1. Rendimiento en lectura en el año 2022 (1)

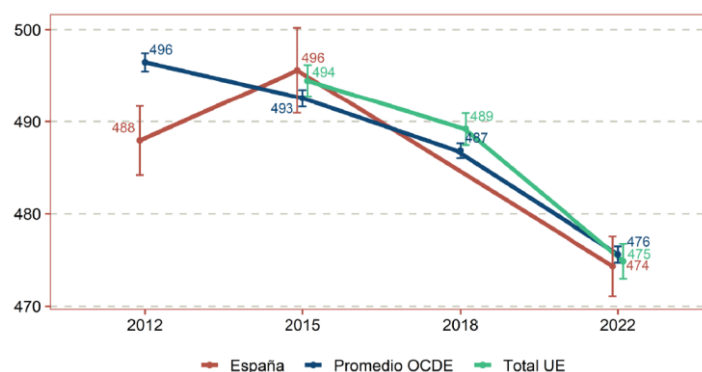


Figura 2.2. Comparativa del rendimiento en lectura entre los años 2012 y 2022 (1)

Sin embargo, no solo habría que analizar los datos sobre los estudiantes, sino también sobre el resto de población. El Ministerio de Cultura de España informa que tras la pandemia de COVID-19, se produjo un incremento de lectores, llegando aproximadamente al 68% de la población española en el año 2023. Sin embargo, hay una cifra que destaca, y es que el 35,9% de los españoles afirman no leer nunca o casi nunca, lo cual es un porcentaje ligeramente elevado (2).

Otra encuesta a estas personas arroja información relevante sobre las causas de que no lean libros, y es importante analizarlo para desarrollar la herramienta adecuada de cara al incentivo de la lectura.

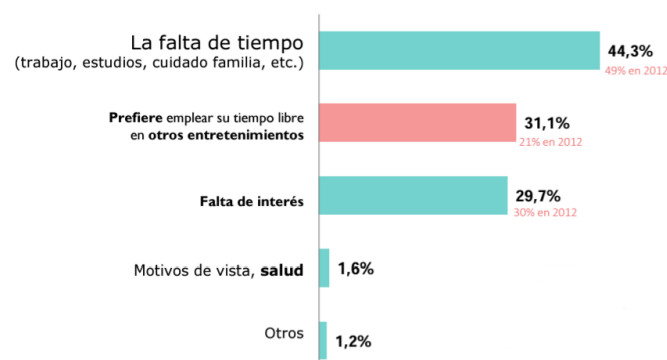


Figura 2.3. Motivos no lectura españoles en 2023 (2)

En la Figura 3 se puede observar que más de la mitad de la población no lee por falta de interés y por preferencias a pasar el tiempo de otra manera. Por ello el objetivo de este TFG radica en desarrollar un dispositivo, dados los avances tecnológicos, que sea motivador e interesante para el incentivo de estos habitantes.

## 2.2. Estructura del TFG

A continuación, se detalla la estructura de la presente memoria de TFG y un breve resumen de cada uno de los 12 capítulos que la componen.

- **Capítulo 1:** Resumen. Se aborda la lectura como clave para el crecimiento personal, destacando su rol en el enriquecimiento del lenguaje y conocimiento. Se propone un dispositivo con gamificación para incentivar la lectura entre la población, ofreciendo seguimiento y análisis del progreso lector.
- **Capítulo 2:** Introducción. Se examina la disminución en competencias lectoras de estudiantes españoles según el informe PISA, y la falta de interés generalizada por la lectura. Se plantea la creación de un dispositivo tecnológico para motivar esta carencia.

- **Capítulo 3:** Objetivos. Se plantea diseñar una herramienta que fomente la lectura y mejore la educación. Incluye el desarrollo de un dispositivo con cámara y pantalla táctil, métricas de lectura, almacenamiento en la nube y gamificación.
- **Capítulo 4:** Estado del arte. Se examinan aplicaciones móviles para el seguimiento de lecturas, resaltando a Goodreads por su funcionalidad. Se proponen varias herramientas para crear una solución similar que resulte innovadora, decantándose finalmente por el hardware ESP32 por su balance entre eficiencia y coste, y se describen sus características principales.
- **Capítulo 5:** Dispositivo y componentes empleados. Se realiza una comparación entre los modelos más eficaces de ESP32, para elegir un kit de desarrollo apropiado y explicar sus componentes.
- **Capítulo 6:** Software. Se analizan distintos entornos de desarrollo para programar ESP32 en lenguaje Arduino, destacando Arduino IDE, CLion IDE y Visual Studio Code por su compatibilidad y facilidades de desarrollo. Se elige CLion por su interfaz intuitiva y gestión eficiente de librerías con PlatformIO.
- **Capítulo 7:** Reconocimiento y diferenciación de libros. Se explica la manera de reconocer los libros, a través de ISBN o QR y en qué consisten estos tipos de códigos.
- **Capítulo 8:** Diseño e implementación. Se detalla extensamente sobre el funcionamiento y desarrollo de la aplicación para el lector con ESP32-S3, así como los métodos de realimentación del usuario, gamificación, conexión con la base de datos para almacenar sus lecturas y la decodificación del código del libro mostrado a la cámara del dispositivo.
- **Capítulo 9:** Conclusiones y resultados. Se resumen los objetivos cumplidos y cómo se han logrado, además de explicar también las limitaciones que han surgido a lo largo del proyecto.
- **Capítulo 10:** Líneas futuras. Se detallan posibles mejoras e implementaciones que se llevarían a cabo en desarrollos futuros para completar el funcionamiento del dispositivo.
- **Capítulo 11:** Bibliografía. Se listan las referencias bibliográficas empleadas para la documentación del presente TFG.

- **Capítulo 12:** Anexos. Se presenta una guía al usuario para indicar el uso del dispositivo y sus funciones.

### 3. OBJETIVOS

El objetivo principal que se persigue con este TFG es diseñar e implementar una herramienta para estimular la práctica de leer y lograr por consecuente una educación de calidad.

Esto se logrará permitiendo al usuario que registre sus lecturas actuales de manera sencilla y rápida a través de la cámara, obteniendo un análisis de su ritmo de lectura y almacenar los datos relativos a libros leídos o leyendo. El dispositivo también estará provisto de una pantalla capacitiva a color con la que el usuario podrá interactuar a través de una interfaz gráfica amigable y llamativa.

A continuación, se listan las fases que permitirán lograr el objetivo deseado:

- Diseño del dispositivo, priorizando la facilidad de uso y bajo coste.
- Reconocimiento del ejemplar de libro mediante la información obtenida de la cámara del dispositivo.
- Determinación de las métricas más adecuadas que fomenten la lectura de los usuarios.
- Almacenamiento y recuperación de la información desde una base de datos alojada en la nube.
- Introducción de la gamificación y la realimentación a los usuarios.
- Implementación del prototipo del dispositivo.

## 4. ESTADO DEL ARTE

A lo largo del tiempo, se han desarrollado multitud de aplicaciones, generalmente para teléfono móvil, para facilitar el seguimiento de lectura.

Estas aplicaciones permiten organizar los libros por categorías para facilitar la búsqueda y registrar el avance de lectura guardando la última página leída, inicio de lectura, fin de lectura, etc.

Una de las más empleadas es Goodreads, con más de 10 millones de descargas. Esta opción es una de las más completas, ya que permite localizar libros según el género, escribiendo el título, el ISBN y escaneando con la cámara del móvil la portada o el ISBN. Además, permite tener una biblioteca personal donde se guardan las lecturas pasadas, actuales y futuras.

La aplicación móvil podría ser una alternativa atractiva debido a la sencillez que presentaría para el usuario y por portabilidad que ofrece, ya que la mayoría de la población dispone de móvil con memoria suficiente para albergar una amplia cantidad de aplicaciones.

Sin embargo, el simple hecho de tener un dispositivo móvil en la mano favorece la distracción de la persona que lo posee por todas las utilidades que se le puede dar. En este proyecto se trata de desarrollar una nueva e innovadora herramienta, empleando tecnología hardware y software de bajo coste para ofrecer funcionalidades similares a las que ofrecen estas aplicaciones móviles.

En el mercado actual existen tres posibles alternativas para realizar esto. Estas son:

- **ESP32** (3; 4): chip SoC (System on a Chip) de bajo coste y bajo consumo de energía con conectividad Wifi y Bluetooth. Se suele utilizar en proyectos de IoT, para manejar sensores, actuadores o sistemas de domótica. Se puede programar en C, C++ o MicroPython (adaptación del lenguaje Python3, nueva versión de Python, que conserva sus principales elementos y se utiliza para programar microcontroladores).
- **Raspberry Pi** (5; 6): pequeño ordenador compacto y de bajo coste, que se conecta con un monitor, ratón y teclado. Su sistema operativo actualmente es Raspberry Pi OS (antes era Raspbian), el cual se basa en Debian (distribución

Linux de software libre). Es capaz de realizar las funciones típicas de un PC. Se puede programar en varios lenguajes de programación, tales como C/C++, Java y Python, entre otros.

- **Arduino UNO** (7; 8; 4; 6): placa de microcontrolador de código abierto basado en el microchip ATmega328P. Eso significa que integra este microcontrolador en una placa de desarrollo que incluye otros elementos, como pines de entrada/salida, oscilador para sincronizar el funcionamiento, regulador de voltaje, interfaz USB, etc. Es una placa de desarrollo sencilla que suelen utilizar las personas principiantes para proyectos sencillos, como control de sensores, LEDs, etc. Cuenta con su propio entorno de desarrollo para programar, Arduino IDE. Tiene su propio lenguaje de programación Wiring, basado en C++.

En la siguiente tabla se recoge una breve comparativa de las principales características de estos tres dispositivos:

| Característica            | RASPBERRY PI                        | ARDUINO UNO                                          | ESP32                      |
|---------------------------|-------------------------------------|------------------------------------------------------|----------------------------|
| Tipo de dispositivo       | Computadora de una sola placa (SBC) | Placa de desarrollo de microcontrolador (ATmega328P) | Microcontrolador           |
| Potencia de procesamiento | Alta                                | Baja                                                 | Media                      |
| Conectividad              | WiFi, Bluetooth, USB                | Sin conectividad inalámbrica                         | WiFi, Bluetooth            |
| Programación              | C/C++, Python, Java                 | C/C++                                                | C/C++, MicroPython         |
| Precio                    | Alto                                | Medio                                                | Bajo                       |
| Ideal para                | Proyectos complejos                 | Proyectos simples de electrónica algo más básica     | Proyectos de IoT, domótica |

*Tabla 4.1. Comparativa entre principales características de Raspberry Pi, ESP32 y Arduino UNO*

Finalmente, se ha elegido realizar el trabajo con ESP32, por las razones que se exponen a continuación (9; 10):

Sobre la conectividad inalámbrica, ESP32 proporciona conexión Wifi y Bluetooth, sin necesidad de módulos adicionales, mientras que muchas de las placas Arduino no poseen esa conectividad.

Raspberry Pi posee mayor capacidad de procesamiento y, por tanto, consume más energía. ESP32 proporciona las funciones que se necesitan para el proyecto consumiendo menos recursos.

La capacidad de procesamiento y memoria de Arduino respecto ESP32 está más limitada, ya que ESP32 cuenta con un microprocesador de 32 bits de doble o simple núcleo.

ESP32 tiene una gran cantidad de pines GPIO, lo que permite una amplia gama de conexiones de periféricos y poder conectar la cámara y pantalla que necesita el actual proyecto.

En conclusión, ESP32 es el dispositivo intermedio que se ajusta a los requerimientos de este proyecto, siendo eficiente y económico.

#### **4.1. ESP32**

El dispositivo que se explicará en detalle en este TFG está basado en el microcontrolador de bajo coste ESP32-S3, diseñado por el fabricante Espressif Systems.

##### *4.1.1. Espressif Systems*

Espressif Systems es una multinacional de semiconductores fundada en el año 2008. Tienen oficinas en distintos países como China, República Checa, India, Singapur y Brasil. Se encargan de diseñar soluciones de comunicación inalámbrica, bajo consumo y AIoT.

Han creado populares series de chips, módulos y placas de desarrollo, como ESP8266, ESP32, ESP32-S, ESP32-C y ESP32-H. Estos chips se definen por ser de bajo coste, inteligentes, adaptables y versátiles.

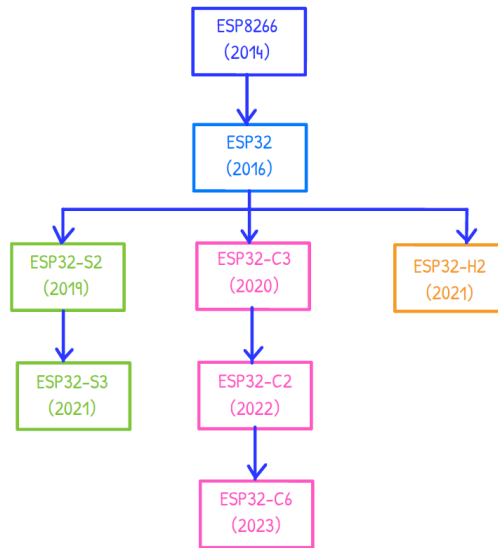


Figura 4.1. Dispositivos fabricados hasta el momento

Además, Espressif Systems también ofrece software de código abierto, como marcos de desarrollo para IoT (Internet of Things), audio, malla, reconocimiento facial y asistentes de voz inteligentes, con el fin de contribuir al desarrollo de la sociedad AIoT en el futuro (11).

#### 4.1.2. Inicio: ESP8266 y ESP32

ESP8266 es un SoC (System on Chip) y fue el primero de su familia en ser diseñado por Espressif Systems en 2014. Es conocido principalmente por la capacidad de conectividad WiFi integrada y el protocolo TCP/IP, lo que permite que se pueda utilizar en proyectos de IoT, dando acceso al dispositivo a la red.

Sin embargo, para lograr una mayor comprensión acerca de estos elementos se debe conocer la diferencia entre chip, módulo y placa o tarjeta de desarrollo, en el ámbito de estas herramientas de Espressif Systems (12).

- **Chip (12):** es el elemento principal, siendo un microcontrolador de bajo consumo de energía, bajo coste y gran capacidad de conectividad. Contiene el núcleo de procesamiento, memoria, interfaces de comunicación y otros periféricos necesarios para ejecutar programas. Se podría decir que es el “cerebro”.



Figura 4.2. SoC ESP8266EX (13)

- **Módulo (12):** suele llevar al chip integrado, además de componentes adicionales como un cristal de 40 MHz, memoria flash y una antena (dependiendo del modelo).



Figura 4.3. Módulo ESP8266 (ESP8266MOD) (14)

- **Placa o tarjeta de desarrollo (12):** integra el módulo en un PCB (Printed Circuit Board), añadiendo de esta manera otros componentes y circuitos adicionales, como conexión serie/USB, alimentación por USB, LEDs, botones de *boot* y *reset* y pines soldados a la placa.



Figura 4.4. Placa de desarrollo que integra el módulo ESP8266MOD (15)

#### 4.1.2.1. Comparativa ESP32 y ESP8266

ESP32 fue el sucesor de ESP8266, teniendo un coste ligeramente superior debido a las siguientes principales mejoras respecto su predecesor:

- Un núcleo adicional.
- Conectividad WiFi más rápida.
- Conectividad Bluetooth 4.2 y Bluetooth de baja energía (BLE)
- Más pines de entrada/salida (GPIO).

En la siguiente tabla se detallan las características técnicas más importantes de los dos de los módulos que integran ESP32 y ESP8266, para comprender mejor la comparativa entre ambos (3):

| CARACTERÍSTICAS          | ESP8266                                                                           | ESP32                                                                               |
|--------------------------|-----------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
|                          |  |  |
| Microprocesador          | Xtensa Single-core 32-bit L106                                                    | Xtensa Dual-Core 32-bit LX6                                                         |
| WiFi                     | 802.11 b/g/n (2.4 GHz), hasta 72.2 Mbps                                           | 802.11 b/g/n (2.4 GHz), hasta 150 Mbps                                              |
| Bluetooth                | No posee                                                                          | Bluetooth 4.2 y BLE                                                                 |
| Frecuencia máxima de CPU | 80 MHz                                                                            | 80/160/240 MHz                                                                      |
| ROM                      | No posee                                                                          | 448 KB                                                                              |
| SRAM                     | 160 KB                                                                            | 520 KB                                                                              |
| Flash                    | 1/2 MB                                                                            | 4 MB                                                                                |
| GPIO                     | 17                                                                                | 34                                                                                  |
| SPI                      | 2                                                                                 | 3                                                                                   |
| I <sup>2</sup> C         | 1                                                                                 | 2                                                                                   |
| I <sup>2</sup> S         | 2                                                                                 | 2                                                                                   |
| UART                     | 2                                                                                 | 2                                                                                   |
| ADC                      | 10 bits de resolución                                                             | 12 bits de resolución                                                               |

Tabla 4.2. Comparación entre ESP8266MOD y ESP32-WROOM-32 (3).

#### 4.1.2.2. Dispositivos ESP32

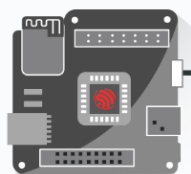
| PRINCIPALES SoCs                                                                    | PRINCIPALES MÓDULOS                                                                 | PRINCIPALES PLACAS DE DESARROLLO                                                      |
|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
|  |  |  |
| ESP32-D0WD-V3                                                                       | ESP32-WROOM-32E/32UE                                                                | ESP32-DevKitC                                                                         |
| ESP32-D0WDR2-V3                                                                     | ESP32--WROOM--DA                                                                    | ESP32-DevKitM-1                                                                       |
| ESP32-U4WDH                                                                         | ESP32-WROVER-E/IE                                                                   | ESP32-PICO-KIT                                                                        |
| ESP32-PICO-V3                                                                       | ESP32-MINI-1/1U                                                                     | ESP32-PICO-V3-ZERO-DevKit (ACK DevKit)                                                |
| ESP32-PICO-V3-02                                                                    | ESP32-PICO-MINI-02/02U                                                              | ESP-EYE                                                                               |
| ESP32-PICO-D4                                                                       | ESP32-PICO-V3-ZERO (ACK Module)                                                     | ESP32-Vaquita-DSPG                                                                    |

Tabla 4.3. Chips, módulos y placas de desarrollo más comunes de ESP32

#### 4.1.2.3. Características y especificaciones ESP32

A pesar de que anteriormente se detallaron las principales características técnicas del ESP32, es necesario indagar más en la hoja de características para ver más detalles de sus funciones y herramientas.

- **Conectividad inalámbrica (16):** este circuito integrado (SoC) está equipado con funcionalidad WiFi 802.11 b/g/n, también conocida como WiFi 4, que opera en la frecuencia de 2.4 GHz y puede alcanzar velocidades de hasta 150 Mbps. Es compatible con Bluetooth 4.2 y Bluetooth de baja energía (BLE).

El componente incluye un bloque de radio para enviar y recibir datos. El receptor de 2.4 GHz transforma la señal de RF de 2.4 GHz en señales de banda base, las cuales son digitalizadas por dos convertidores analógico-digitales (ADC) de alta velocidad y resolución, permitiendo así que los módulos WiFi y Bluetooth procesen la información recibida. Para ajustarse a las fluctuaciones del canal de señal, el chip integra filtros de RF, un control automático de ganancia (AGC), circuitos para la eliminación de desplazamiento de corriente continua (offset de CC) y filtros de banda base.

Por otro lado, el transmisor de 2.4 GHz convierte las señales de banda base en una señal de RF de 2.4 GHz y la amplifica a través de un potente amplificador de potencia CMOS antes de enviarla a la antena.

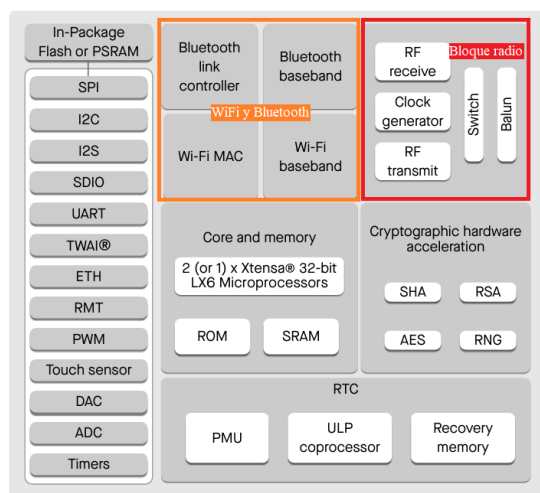


Figura 4.5. Bloques conectividad inalámbrica (16)

- **Núcleo (16):** ESP32 es capaz de trabajar con uno o dos de los núcleos Xtensa 32-bit LX6 que posee. Al tener la capacidad de dos núcleos y ya que cada uno

procesa datos de 32 bits a la vez, el ESP32 es capaz de ejecutar múltiples tareas de manera simultánea, aumentando así el rendimiento del sistema.

Sumado a esto, cuenta también con un coprocesador de ultra bajo consumo (ULP) que está dedicado a tareas sencillas que requieren un consumo mínimo de energía, tales como monitoreo de sensores, mediciones con el convertidor analógico-digital, etc. De esta manera se permite que los procesadores principales permanezcan en un estado de reposo profundo (Deep sleep).

La frecuencia normal de operación de estos procesadores es de 240 MHz.

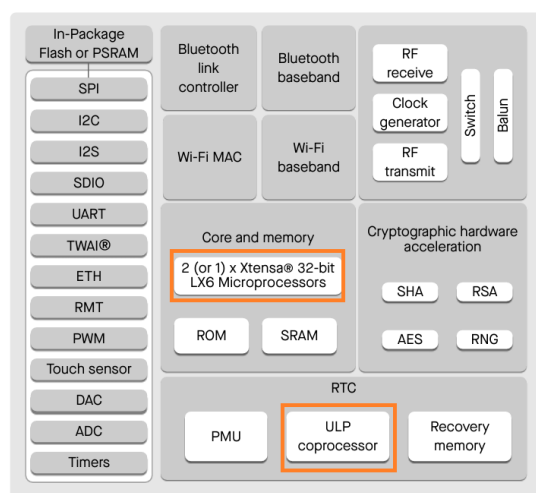


Figura 4.6. Bloques procesador y coprocesador (16)

- **Memoria (16):** la memoria en ESP32 se puede clasificar en memorias internas y memorias externas. Incluyéndose las memorias internas en el chip, y las externas, añadidas para expandir la capacidad del dispositivo.

En las memorias internas se encuentran:

- **ROM (448 KB):** almacena datos sobre el programa de arranque, configuración del chip e información básica que se necesita al encender el SoC. Esta memoria no puede ser modificada ni reprogramada.
- **SRAM (520 KB):** utilizada por el procesador para almacenar datos e instrucciones de manera temporal durante la ejecución del programa. Se puede leer y escribir.
- **Memoria rápida RTC (8 KB) (17):** empleada por el procesador principal para almacenar datos tras salir del modo *Deep Sleep*.

- **Memoria lenta RTC (8 KB) (17):** empleada por el coprocesador ULP para guardar las variables globales y estáticas del modo *Deep sleep* que emplea el código ejecutado desde la memoria RTC. Se siguen manteniendo los valores almacenados, incluso después de salir del modo *Deep sleep*.
- **Efuse (1 Kbit):** de los 1024 bits totales disponibles, 256 bits están dedicados al sistema para usos como la dirección MAC y la configuración del chip, mientras que los 768 bits restantes están asignados para el uso de aplicaciones específicas del cliente, tales como la encriptación de la memoria flash y la identificación del chip.

Las memorias externas que suele incluir (al conectar el chip a un módulo):

- **Flash externa (0/4/8/16 MB):** tipo de almacenamiento no volátil, es decir, no se pierden los datos al apagar el dispositivo, que sirve para guardar el código del programa y los datos del sistema. Algunos chips de ESP32 vienen con memoria flash ya incorporada, pero no es usual. Dependiendo del modelo de módulo puede tener un valor de 4/8/16 MB
- **PSRAM (0/2/4/8 MB):** memoria adicional de acceso aleatorio que se utiliza para expandir la capacidad de memoria SRAM, almacena datos en tiempo real (por ejemplo, búferes). Algunos chips la traen incorporada, aunque es poco común.

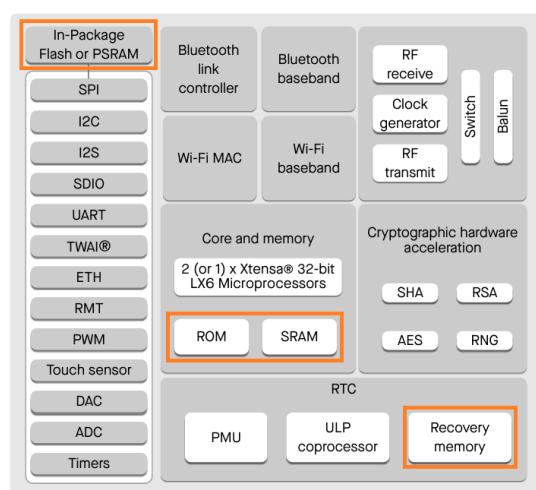


Figura 4.7. Bloques memorias (16)

- **Periféricos (16):** componentes adicionales o dispositivos externos que se pueden conectar al SoC ESP32 para ampliar sus capacidades y funcionalidades.

- **Pines entrada/salida de propósito general (GPIO):** pines que pueden configurarse para funcionar tanto como entradas como salidas digitales. Permiten que el microcontrolador se comunique con dispositivos externos, siendo capaz de controlar y leer las señales eléctricas. Tiene 34 GPIOs.
- **Conversor analógico digital (ADC):** ESP32 posee dos ADC de 12 bits cada uno. Con esto el chip es capaz de medir voltaje hasta en un máximo de 18 pines.
- **Conversor digital analógico (DAC):** para ESP32, existen otros dos canales DAC de 8 bits, lo cual permite transformar dos señales de entrada digitales en dos señales de salida analógicas.
- **Sensor táctil:** existen 10 GPIOs de alta sensibilidad y bajo nivel de ruido que detectan cambios al ser tocados, ya sea con una mano o un objeto.
- **Controlador SD/SDIO/MMC:** interactúa con tarjetas de memoria externas SD (*Secure Digital*), MMC (*MultiMedia Card*) y SDIO (*Secure Digital Input/Output*).
- **Universal Asynchronous Receiver Transmitter (UART):** 3 interfaces (UART0, UART1, UART2) que permiten la comunicación serie entre dispositivos. La velocidad llega hasta 5 Mbps. Las comunicaciones UART son asíncronas, lo que significa que no hay un reloj de sincronización compartido entre el transmisor y el receptor. En cambio, se utilizan bits de inicio y de parada para delimitar cada byte de datos.
- **I<sup>2</sup>C:** ESP32 tiene 2 interfaces I<sup>2</sup>C que sirven como maestro o esclavo a la hora de comunicarse. El maestro inicia y controla la comunicación, mientras que el esclavo responde las solicitudes de datos que le llegan del maestro.
- **I<sup>2</sup>S:** ESP32 dispone de 2 interfaces I<sup>2</sup>S estándar, que pueden funcionar también en modo maestro o esclavo, en modos de comunicación dúplex completo o semidúplex. Transmite datos de audio digital entre dispositivos.
- **Controlador remoto infrarrojo:** 8 canales de transmisión y recepción a distancia por infrarrojos.

- **PWM:** posibilidad de generar señales PWM de salida en los pines GPIO, controlando la cantidad de energía entregada.
- **SPI:** es un protocolo de comunicación que permite la transferencia de datos entre el microcontrolador y uno o varios dispositivos periféricos. Es un protocolo síncrono, lo que significa que utiliza un reloj para sincronizar la transferencia de datos (80 MHz).
  - El ESP32 dispone de tres SPIs hardware (SPI, HSPI y VSPI) que pueden funcionar en modo esclavo y maestro de comunicación. Utiliza pines GPIO predefinidos, no se pueden reasignar y poseen una alta velocidad.
  - También se puede disponer de SPI por software. En este caso el microcontrolador emula el protocolo SPI mediante software. Esto es más lento que el hardware SPI, ya que el microcontrolador tiene que manejar todos los detalles del protocolo SPI. Sin embargo, la ventaja es que se puede usar cualquier GPIO para la comunicación SPI, lo que proporciona más flexibilidad.

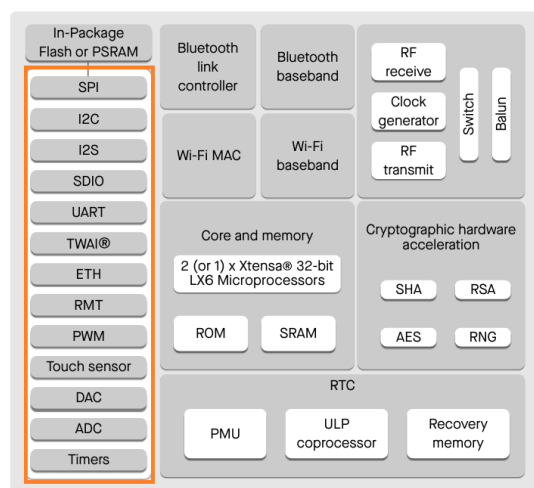


Figura 4.8. Bloques periféricos (16)

- **Seguridad (16):** esencial para proteger cualquier dispositivo de amenazas. Los principales mecanismos que utiliza ESP32 para garantizar esta seguridad son:
  - **Secure boot:** sistema que permite un inicio seguro del chip, permitiendo que solo se ejecute el firmware firmado y autenticado.

- **Flash encryption:** posibilidad de cifrar la memoria flash para proteger información sensible que pueda contener.
- **Aceleradores de hardware para criptografía:** capacidad de encriptar y desencriptar de manera rápida y eficiente empleando algoritmos específicos como: AES, Hash (SHA-2), RSA, ECC, *Random Number Generator* (RNG).

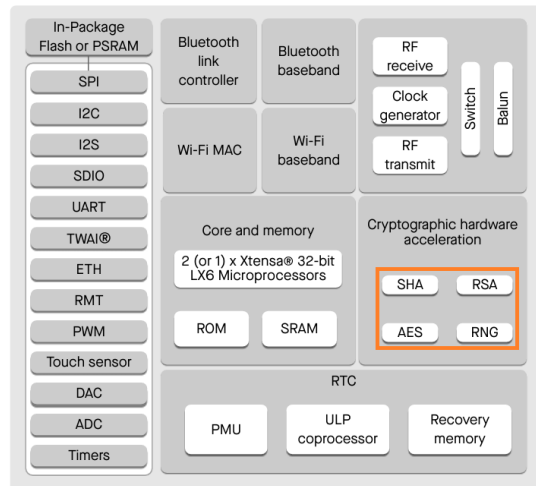


Figura 4.9. Bloques seguridad (16)

Una vez que se han explicado las características técnicas, se listan las aplicaciones que define el fabricante que tiene ESP32:

- Hogar inteligente
- Automatización industrial
- Sanidad
- Electrónica de consumo
- Agricultura inteligente
- Máquinas de punto de venta
- Robots de servicio
- Dispositivos de audio
- Concentradores de sensores IoT genéricos de bajo consumo
- Registradores de datos IoT genéricos de bajo consumo
- Cámaras de vídeo
- Reconocimiento de voz

- Reconocimiento de imágenes
- Tarjeta de red SDIO Wi-Fi + Bluetooth
- Sensores táctiles y de proximidad

## 5. DISPOSITIVO Y COMPONENTES EMPLEADOS

### 5.1. Comparación ESP32, ESP32-S2, ESP32-S3, ESP32-C3

Conocidas las principales características explicadas anteriormente del chip ESP32 y con el fin de encontrar el mejor modelo para llevar a cabo el trabajo que se expone en este TFG, se analizan los modelos más avanzados que existen el mercado actual, comparando sus principales características en la siguiente tabla a modo de resumen.

|                                       | ESP32                                                | ESP32-S2                                 | ESP32-S3                                 | ESP32-C3                                | ESP32-C6                                     |
|---------------------------------------|------------------------------------------------------|------------------------------------------|------------------------------------------|-----------------------------------------|----------------------------------------------|
| Procesador principal                  | Xtensa single core 32-bit LX6 (optionally dual core) | Xtensa single core 32-bit LX7            | Xtensa dual core 32-bit LX7 dual core    | RISC-V single core 32-bit               | RISC-V single core 32-bit                    |
| Frecuencia máxima de CPU              | 80/160/240 MHz                                       | 80/160/240 MHz                           | 80/160/240 MHz                           | 80/240 MHz                              | 80/160 MHz                                   |
| SRAM                                  | 520 KB                                               | 320 KB                                   | 512 KB                                   | 400 KB                                  | HP SRAM: 512 KB<br>LP SRAM: 16 KB            |
| ROM                                   | 448 KB                                               | 128 KB                                   | 384 KB                                   | 384 KB                                  | 320 KB                                       |
| Cache                                 | 32 KB                                                | 8/16 KB                                  | 32/64 KB                                 | 16 KB                                   | 32 KB                                        |
| WiFi                                  | WiFi 4                                               | WiFi 4                                   | WiFi 4                                   | WiFi 4                                  | WiFi 6                                       |
| Bluetooth                             | Bluetooth 4.2 y BLE                                  | X                                        | BLE 5.0                                  | BLE 5.0                                 | BLE 5.3                                      |
| Ethernet                              | ✓                                                    | X                                        | X                                        | X                                       | X                                            |
| Aceleración de Hardware Criptográfico | AES, Hash, RSA, ECC, RNG                             | AES, Hash, RSA, RNG, HMAC, Firma digital | AES, Hash, RSA, RNG, HMAC, Firma digital | AES, SHA, RSA, RNG, HMAC, Firma digital | AES, ECC, HMC, RSA, SHA, Firma digital, Hash |
| Secure boot                           | ✓                                                    | ✓                                        | ✓                                        | ✓                                       | ✓                                            |
| Flash encryption                      | ✓                                                    | ✓                                        | ✓                                        | ✓                                       | ✓                                            |
| SPI                                   | 4                                                    | 4                                        | 4                                        | 3                                       | 2                                            |
| I <sup>2</sup> C                      | 2                                                    | 2                                        | 2                                        | 1                                       | 1                                            |
| I <sup>2</sup> S                      | 2                                                    | 1                                        | 2                                        | 1                                       | 1                                            |
| UART                                  | 3                                                    | 2                                        | 3                                        | 2                                       | 2                                            |
| GPIO                                  | 34                                                   | 43                                       | 45                                       | 22                                      | 30                                           |
| USB                                   | X                                                    | USB OTG                                  | USB OTG<br>USB<br>Serial/JTAG            | USB<br>Serial/JTAG<br>controller        | USB<br>Serial/JTAG<br>controller             |
| ADC                                   | 12-bit SAR ADC up to 18 channels                     | 2×13-bit SAR ADCs, up to 20 channels     | 2×12-bit SAR ADCs, up to 20 channels     | 2×12-bit SAR ADCs, up to 6 channels     | 1×12-bit SAR ADC, up to 7 channels           |
| DAC                                   | 2×8-bit DAC                                          | 2×8-bit DACs                             | X                                        | X                                       | X                                            |
| Touch Sensor                          | 10                                                   | 14                                       | 14                                       | X                                       | X                                            |

Tabla 5.1. Comparativa de las características técnicas distintos modelos de ESP32. (18; 19; 20; 21; 22; 16)

Además, para una mejor comprensión de estos dispositivos y su complejidad se observará su diagrama de bloque funcional, según la hoja de características de cada uno de los modelos:

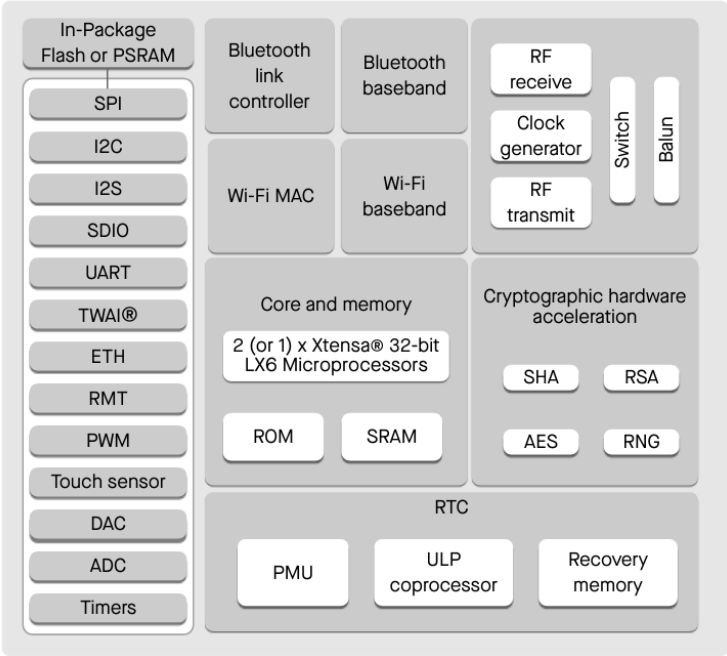


Figura 5.1. Diagrama de bloque funcional ESP32. (16)

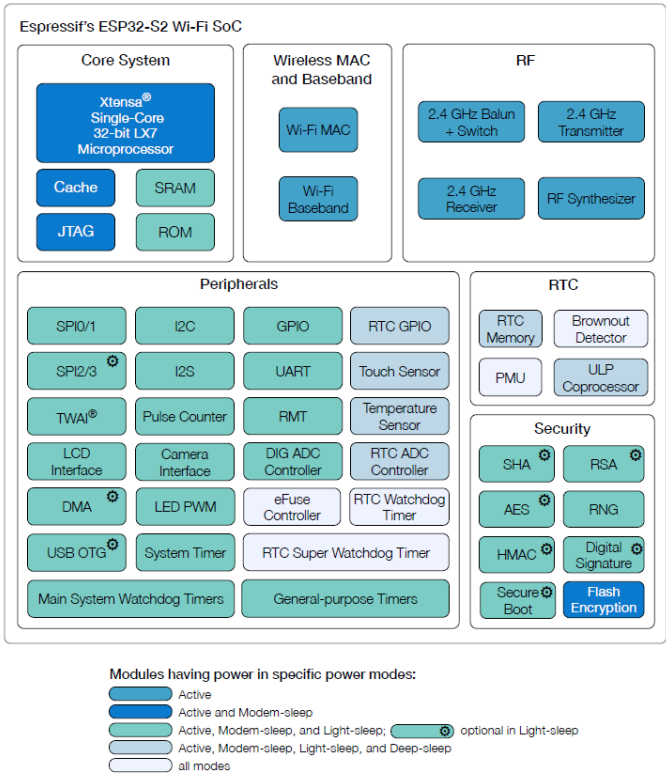


Figura 5.2. Diagrama de bloque funcional ESP32-S2. (21)

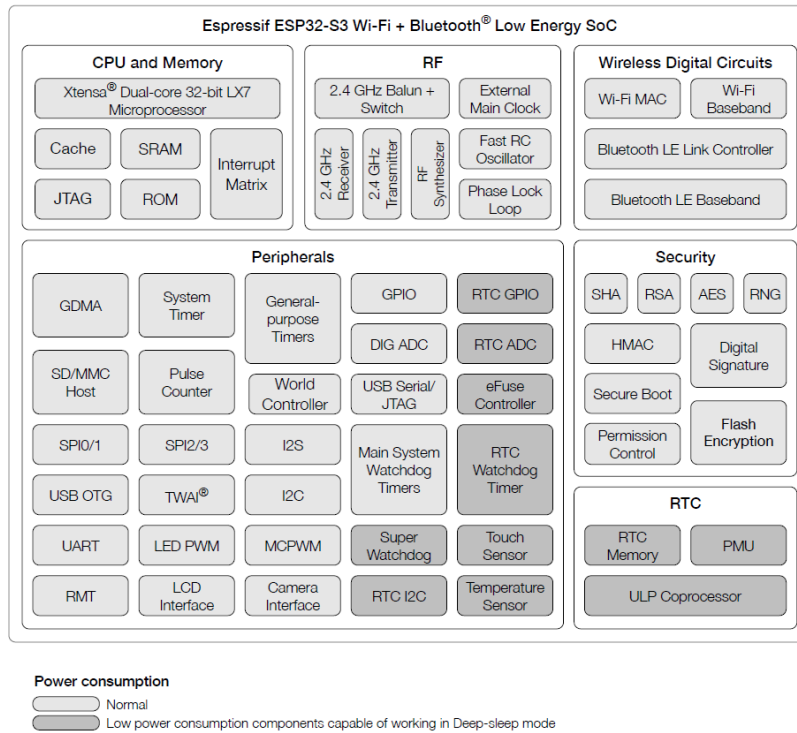


Figura 5.3. Diagrama de bloque funcional ESP32-S3. (22)

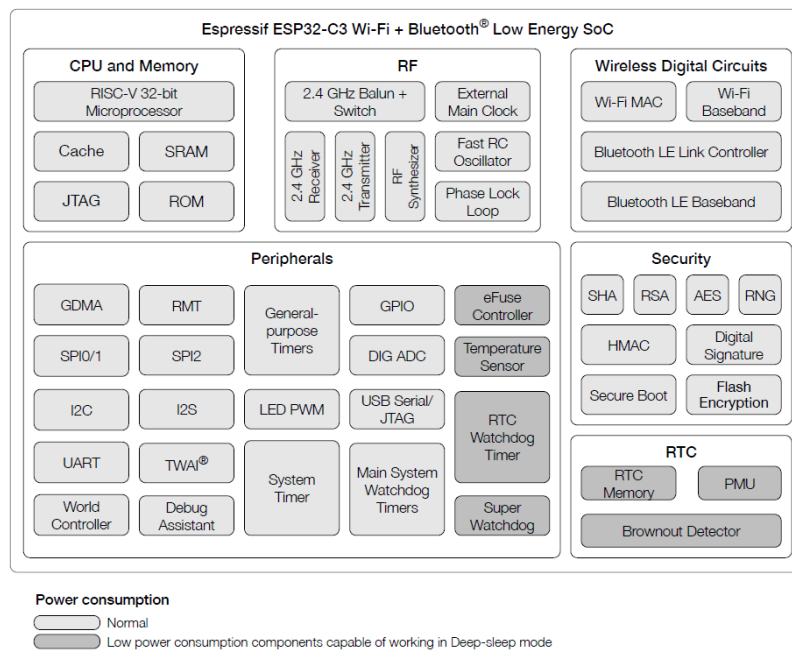


Figura 5.4. Diagrama de bloque funcional ESP32-C3. (19)

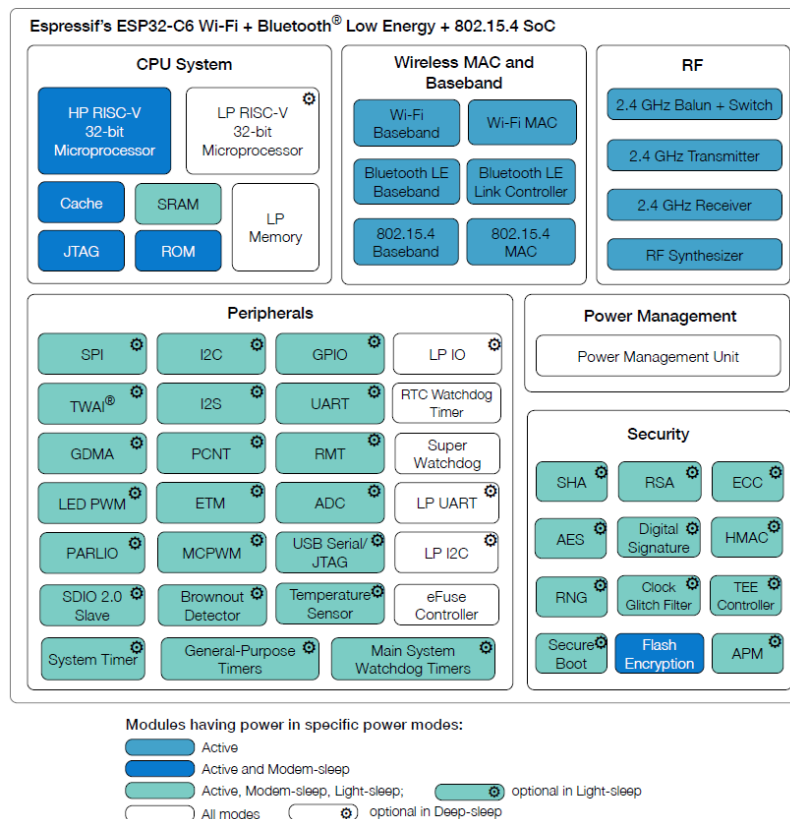


Figura 5.5. Diagrama de bloque funcional ESP32-C6. (20)

Finalmente, con toda esta información se ha llegado a la conclusión de que el dispositivo que presenta mejores características técnicas es **ESP32-S3**. Esto es debido a que posee un procesador principal de dos núcleos y que en general supera al resto de modelos respecto a la cantidad de SRAM, ROM, cache y pines GPIO, contando con un total de 45, lo cual permite conectar muchos más elementos para realizar funciones más avanzadas.

El procesador dual core Xtensa LX7 a 240 MHz del ESP32-S3 ofrece un rendimiento significativamente más rápido que los núcleos únicos de los demás modelos. Esto permite ejecutar aplicaciones más exigentes y proporcionará una mayor productividad para procesar las imágenes y realizar el reconocimiento del libro, para después garantizar al usuario una experiencia fluida al interactuar con la interfaz gráfica de la pantalla.

Ofrece 45 pines GPIO, más que cualquier otro modelo en la tabla. Esto brinda la posibilidad de conectar una mayor variedad de elementos y otros periféricos, ampliando el alcance de los proyectos. Con lo que se podrá conectar a la cámara para reconocer el ejemplar de lectura del usuario y a la pantalla, con la que el lector podrá interactuar sin problema gracias a los sensores táctiles.

Cuenta con 512 KB de SRAM, superando al resto de los modelos comparados. Esta mayor capacidad permite alojar aplicaciones más complejas con mayores conjuntos de datos.

Asimismo, posee conectividad inalámbrica WiFi que permitirá conectar el ESP32-S3 con una base de datos alojada en la nube para almacenar las lecturas del usuario de manera eficiente.

## 5.2. Estudio previo elección placa desarrollo ESP32-S3

A continuación, se detallan las posibles tarjetas de desarrollo ESP32-S3 a utilizar en el proyecto. Se desea buscar un kit adecuado que incluya además del módulo ESP32-S3, pantalla capacitiva a color y cámara.

Con el fin de una mejor comparación del presupuesto y características, se dividirán las posibles opciones según la plataforma de compra.

### MOUSER

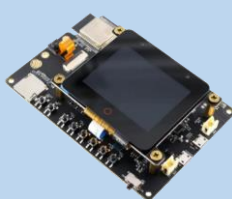
|                                            | ESP32-S3-EYE                                                                        | ESP32-S3-Korvo-2                                                                     | ESP32-S3-BOX-3                                                                        |
|--------------------------------------------|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| Imagen                                     |  |  |  |
| Módulo                                     | ESP32-S3-WROOM-1U-N8R8                                                              | ESP32-S3-WROOM-1-N16R8                                                               | ESP32-S3-WROOM-1-N16R16V                                                              |
| Pines GPIO                                 | 36                                                                                  | 36                                                                                   | 36                                                                                    |
| Capacidad de memoria Flash                 | 8 MB                                                                                | 16 MB                                                                                | 16 MB                                                                                 |
| PSRAM                                      | 8 MB                                                                                | 8 MB                                                                                 | 16 MB                                                                                 |
| Salida para altavoces                      | No                                                                                  | Sí                                                                                   | Sí                                                                                    |
| Con cámara incorporada                     | Sí (2 megapíxel, color, 1600x1200 OV2640)                                           | Sí (el fabricante no especifica el tipo)                                             | No                                                                                    |
| Con pantalla táctil capacitiva incorporada | Sí (LCD color 1.3", 240x240)                                                        | Sí (LCD color 2.4", 320x240)                                                         | Sí (LCD color 2.4", 320x240)                                                          |
| Presupuesto                                | 41.85 €                                                                             | 36,58 €                                                                              | 48,36 €                                                                               |

Tabla 5.2. ESP32-S3 en Mouser

## DIGIKEY

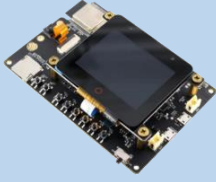
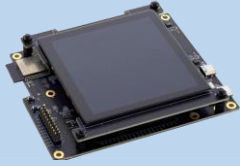
|                                            | ESP32-S3-Korvo-2                                                                  | ESP32-S3-LCD-EV-BOARD                                                               |
|--------------------------------------------|-----------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
| Imagen                                     |  |  |
| Módulo                                     | ESP32-S3-WROOM-1-N16R8                                                            | ESP32-S3-WROOM-1-N16R16V                                                            |
| Pines GPIO                                 | 36                                                                                | 36                                                                                  |
| Capacidad de memoria Flash                 | 16 MB                                                                             | 16 MB                                                                               |
| PSRAM                                      | 8 MB                                                                              | 16 MB                                                                               |
| Salida para altavoces                      | Sí                                                                                | Sí                                                                                  |
| Con cámara incorporada                     | Sí (el fabricante no especifica el tipo)                                          | No                                                                                  |
| Con pantalla táctil capacitiva incorporada | Sí (LCD color 2.4", 320x240)                                                      | Sí (LCD color 3.95", 480x480)                                                       |
| Presupuesto                                | 55,16 €                                                                           | 65,08 €                                                                             |

Tabla 5.3. ESP32-S3 en DigiKey

## ALIEXPRESS

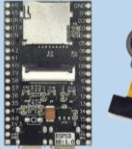
|                                            | ESP32-S3-EYE                                                                        | ESP32-S3 CAM                                                                        | ESP32-S3                                                                             | QS-ESP32-S3-EYE                                                                       |
|--------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| Imagen                                     |  |  |  |  |
| Módulo                                     | ESP32-S3-WROOM-1U-N8R8                                                              | ESP32-S3-WROOM-1-N16R8                                                              | Chip: ESP32-S3R8                                                                     | ESP32-S3-WROOM-1-N8R8                                                                 |
| Pines GPIO                                 | 36                                                                                  | 33                                                                                  | 13                                                                                   | No lo indica el fabricante                                                            |
| Capacidad de memoria Flash                 | 8 MB                                                                                | 16 MB                                                                               | 16 MB                                                                                | 8 MB                                                                                  |
| PSRAM                                      | 8 MB                                                                                | 8 MB                                                                                | 8 MB                                                                                 | 8 MB                                                                                  |
| Salida para altavoces                      | No                                                                                  | No                                                                                  | No                                                                                   | No                                                                                    |
| Con cámara incorporada                     | Sí (2 megapíxel, color, 1600x1200 OV2640)                                           | Sí (admite sensor OV2640, OV7725, OV3660)                                           | No                                                                                   | Sí (2 megapíxel)                                                                      |
| Con pantalla táctil capacitiva incorporada | Sí (LCD color 1.3", 240X240)                                                        | No                                                                                  | Sí (LCD color 1.14", 170x320)                                                        | Sí (LCD color 240X240)                                                                |
| Presupuesto                                | 71,36 €                                                                             | 12,08€                                                                              | 17,72€                                                                               | 19,80€                                                                                |

Tabla 5.4. ESP32-S3 en Aliexpress

## AMAZON

| ESP32-S3 Kit de Freenove                   |                                                                                             |
|--------------------------------------------|---------------------------------------------------------------------------------------------|
| Imagen                                     |          |
| Módulo                                     | Módulo ESP32-S3-WROOM-1 con chip ESP32-S3R8                                                 |
| Pines GPIO                                 | 36                                                                                          |
| Capacidad de memoria Flash                 | 8 MB                                                                                        |
| PSRAM                                      | 8 MB                                                                                        |
| Salida para altavoces                      | Sí                                                                                          |
| Con cámara incorporada                     | Sí (sensor OV2640 de 2 megapíxeles)                                                         |
| Con pantalla táctil capacitiva incorporada | Sí, (LCD color, 320X240)                                                                    |
| Presupuesto                                | 64,95 €                                                                                     |
| Información extra                          | También se puede encontrar el mismo producto en <b>Aliexpress:</b><br><b>Precio: 48,86€</b> |

Tabla 5.5. ESP32-S3 en Amazon

## ELECROW

| ESP32-S3 de Crowtail                       |                                                                                      |
|--------------------------------------------|--------------------------------------------------------------------------------------|
| Imagen                                     |  |
| Módulo                                     | ESP32-S3-WROOM-1-N16R8                                                               |
| Pines GPIO                                 | 31                                                                                   |
| Capacidad de memoria Flash                 | 16 MB                                                                                |
| PSRAM                                      | 8 MB                                                                                 |
| Salida para altavoces                      | Sí                                                                                   |
| Con cámara incorporada                     | Sí (sensor OV2640 de 2 megapíxeles)                                                  |
| Con pantalla táctil capacitiva incorporada | Sí, (LCD color de resolución 320x480)                                                |
| Presupuesto                                | 35,91 €                                                                              |

Tabla 5.6. ESP32-S3 en Elecrow

Tras analizar las tablas anteriores se llega a la conclusión de que las tres posibles mejores alternativas son ESP32-S3 de Crowtail, el kit de ESP32-S3 de Freenove y ESP32-S3-Korvo-2. Estas tres opciones poseen cámara y pantalla, que es lo primordial para este proyecto.

En términos de precio, Elecrow es la opción más económica, seguida de Mouser con un precio intermedio, y finalmente Amazon, que es la opción más costosa. Sin embargo, para el kit de Freenove en Amazon, existen alternativas para reducir este coste. Por ejemplo, para los centros educativos, se podría solicitar un presupuesto a través de Amazon, lo que podría resultar en un precio más bajo. Además, este kit también está disponible en Aliexpress, una plataforma que podría ofrecer un precio más competitivo.

En términos de especificaciones técnicas, aunque el kit de Freenove tiene menos memoria Flash (8 MB) en comparación con los kits de Mouser y Elecrow (16 MB), esta cantidad debería ser suficiente para el presente TFG. Todos ellos cuentan además con 8 MB de PSRAM, es decir, poseen la misma cantidad de memoria adicional a la RAM, para utilizarse en caso de que la aplicación requiera una mayor cantidad esta memoria para almacenar los datos temporales mientras la aplicación está en ejecución.

A pesar de que para el modelo ESP32-S3-Korvo-2 el fabricante no ha especificado qué sensor de cámara está utilizando, se conoce que el sensor OV2640 es compatible con ESP32-S3, por lo que se podría utilizar para cualquiera de estas tres opciones, con lo que en este aspecto no habría diferencias.

En cuanto a la conectividad inalámbrica, todos los kits cuentan con soporte para WiFi y Bluetooth.

En cuanto a la resolución de la pantalla, el modelo en Elecrow destaca con una pantalla LCD a color de resolución 320x480, mientras que Mouser y Freenove ofrecen una resolución más baja de 320x240, aunque esto debería ser suficiente también.

Finalmente, el kit de Freenove ofrece ventajas significativas en términos de documentación y soporte. Incluye un tutorial en PDF muy extenso y específico, así como archivos y sketches modificados para funcionar con este kit específico, con lo que se facilita ampliamente el desarrollo del trabajo. Además, la compra a través de Amazon asegura un proceso de compra rápido y fiable e incluye multitud de elementos en el kit que amplían las posibilidades de desarrollar funciones extra: tarjeta SD de 1 GB, dos

altavoces, convertidor de audio y amplificador, pulsioxímetro y pulsómetro, además de otras herramientas

Por todas estas especificaciones se ha tomado la decisión de utilizar el kit que ofrece la multinacional Amazon.

5.3. Desarrollo kit elegido y componentes

A continuación, se explican en detalle los componentes del kit elegido para el actual proyecto.

La placa de expansión permite la conexión de todos los componentes, proporcionando funcionalidades adicionales a ESP32-S3. En la parte posterior se detalla la conexión de cada GPIO.



Figura 5.6. Anverso placa del kit

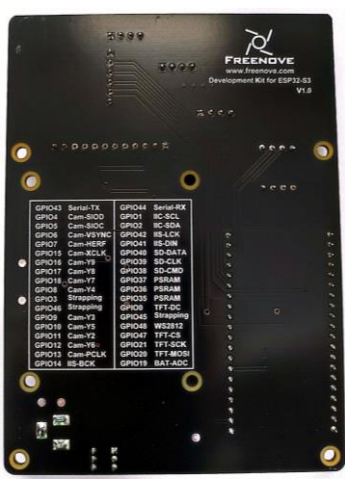


Figura 5.7. Reverso placa del kit

La placa de desarrollo de este kit incluye el módulo ESP32-S3-WROOM-1, cuyas características son las siguientes:

|                          | ESP32-S3-WROOM-1                      |
|--------------------------|---------------------------------------|
| Procesador principal     | Xtensa dual core 32-bit LX7 dual core |
| Frecuencia máxima de CPU | 80/160/240 MHz                        |
| SRAM                     | 512 KB                                |
| ROM                      | 384 KB                                |
| Flash                    | Hasta 16 MB                           |
| PSRAM                    | Hasta 16 MB                           |
| Cache                    | 32/64 KB                              |
| WiFi                     | WiFi 4                                |
| Bluetooth                | BLE 5.0                               |

|                                       |                                          |
|---------------------------------------|------------------------------------------|
| Ethernet                              | X                                        |
| Aceleración de Hardware Criptográfico | AES, Hash, RSA, RNG, HMAC, Firma digital |
| Secure boot                           | ✓                                        |
| Flash encryption                      | ✓                                        |
| SPI                                   | 4                                        |
| I <sup>2</sup> C                      | 2                                        |
| I <sup>2</sup> S                      | 2                                        |
| UART                                  | 3                                        |
| GPIO                                  | 36                                       |
| USB                                   | USB OTG<br>USB Serial/JTAG               |
| ADC                                   | 2×12-bit SAR ADCs, up to 20 channels     |
| DAC                                   | X                                        |
| Touch Sensor                          | 14                                       |

Tabla 5.7. Características técnicas del módulo ESP32-S3-WROOM-1 (23)

Sin embargo, es necesario concretar volviendo a los conceptos de chip, módulo y placa de desarrollo para indicar específicamente los de este kit de Freenove.

El SoC es el ESP32-S3R8, cuya memoria Flash es de 8 MB (no 16 MB como indicaba la tabla anterior).

| Ordering Code | SiP Flash       | SiP PSRAM        | Ambient Temperature (°C) | SPI Voltage |
|---------------|-----------------|------------------|--------------------------|-------------|
| ESP32-S3      | —               | —                | −40 ~ 105                | 3.3 V/1.8 V |
| ESP32-S3FN8   | 8 MB (Quad SPI) | —                | −40 ~ 85                 | 3.3 V       |
| ESP32-S3R2    | —               | 2 MB (Quad SPI)  | −40 ~ 85                 | 3.3 V       |
| ESP32-S3R8    | —               | 8 MB (Octal SPI) | −40 ~ 65                 | 3.3 V       |
| ESP32-S3R8V   | —               | 8 MB (Octal SPI) | −40 ~ 65                 | 1.8 V       |
| ESP32-S3FH4R2 | 4 MB (Quad SPI) | 2 MB (Quad SPI)  | −40 ~ 105                | 3.3 V       |

Tabla 5.8. Chip del kit del fabricante Freenove (22)

El módulo tal y como se indicó en la tabla 5.8, es ESP32-S3-WROOM-1, y más concretamente ESP32-S3-WROOM-1-N8R8. Conserva la PSRAM del chip y añade memoria Flash extra hasta los 8 MB.

| Ordering Code <sup>2</sup>            | Flash <sup>3,4</sup> | PSRAM <sup>5</sup> | Ambient Temp. <sup>6</sup><br>(°C) | Size <sup>7</sup><br>(mm)     |
|---------------------------------------|----------------------|--------------------|------------------------------------|-------------------------------|
| ESP32-S3-WROOM-1-N4                   | 4 MB (Quad SPI)      | -                  | −40 ~ 85                           | 18.0<br>×<br>25.5<br>×<br>3.1 |
| ESP32-S3-WROOM-1-N8                   | 8 MB (Quad SPI)      | -                  | −40 ~ 85                           |                               |
| ESP32-S3-WROOM-1-N16                  | 16 MB (Quad SPI)     | -                  | −40 ~ 85                           |                               |
| ESP32-S3-WROOM-1-H4                   | 4 MB (Quad SPI)      | -                  | −40 ~ 105                          |                               |
| ESP32-S3-WROOM-1-N4R2                 | 4 MB (Quad SPI)      | 2 MB (Quad SPI)    | −40 ~ 85                           |                               |
| ESP32-S3-WROOM-1-N8R2                 | 8 MB (Quad SPI)      | 2 MB (Quad SPI)    | −40 ~ 85                           |                               |
| ESP32-S3-WROOM-1-N16R2                | 16 MB (Quad SPI)     | 2 MB (Quad SPI)    | −40 ~ 85                           |                               |
| ESP32-S3-WROOM-1-N4R8                 | 4 MB (Quad SPI)      | 8 MB (Octal SPI)   | −40 ~ 65                           |                               |
| ESP32-S3-WROOM-1-N8R8                 | 8 MB (Quad SPI)      | 8 MB (Octal SPI)   | −40 ~ 65                           |                               |
| ESP32-S3-WROOM-1-N16R8                | 16 MB (Quad SPI)     | 8 MB (Octal SPI)   | −40 ~ 65                           |                               |
| ESP32-S3-WROOM-1-N16R16V <sup>6</sup> | 16 MB (Quad SPI)     | 16 MB (Octal SPI)  | −40 ~ 65                           |                               |

Tabla 5.9. Módulo del kit del fabricante Freenove (23)

Por último, la placa de desarrollo el fabricante indica que es ESP32S3 Dev Module y con todo estos datos se puede confirmar que las características correspondientes al dispositivo son las que indica la tabla 5.8, a excepción de las memorias Flash y PSRAM (8 MB).

Continuando con los pines, se proporciona una imagen que facilita la comprensión sobre sus funciones y distribución:

# FREENOVE ESP32-S3 WROOM Pinout

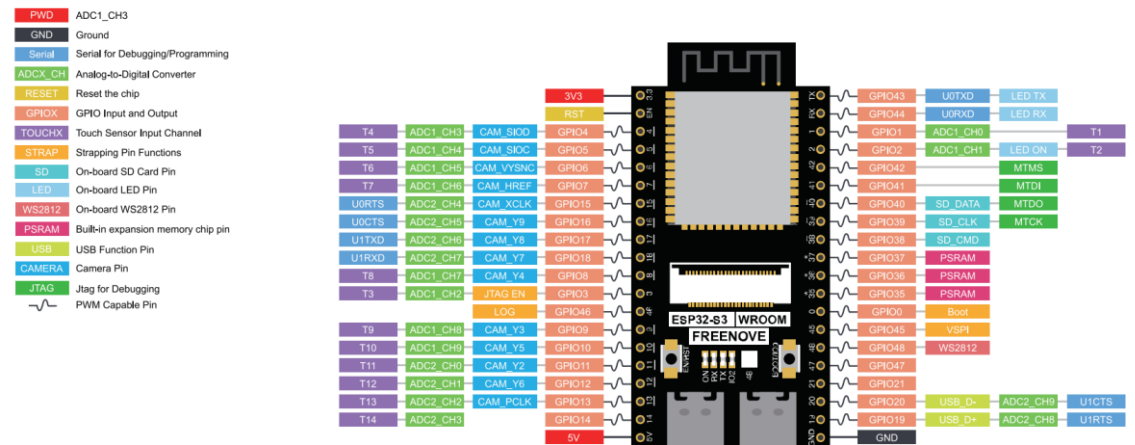


Figura 5.8. Pines de la placa de desarrollo de Freenove (24)

Por otra parte, el módulo cuenta con una antena PCB integrada con impedancia de salida de 50  $\Omega$ . Esto es una antena de radiofrecuencia diseñada y construida en el propio módulo del chip. Lo cual es cómodo, ya que evita tener que usar una antena externa y realizar las conexiones pertinentes. (24)



Figura 5.9. Módulo ESP32-S3-WROOM-1 (23)

Las interfaces de hardware de ESP32-S3 WROOM se distribuyen de la siguiente manera:

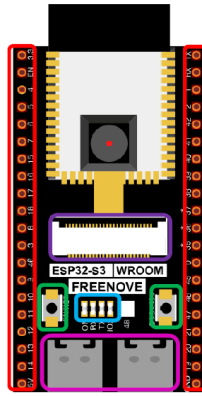


Figura 5.10. Interfaces de ESP32-S3-WROOM-1 señaladas (24)

| Box color                                                                         | Corresponding resources introduction     |
|-----------------------------------------------------------------------------------|------------------------------------------|
|  | GPIO pins                                |
|  | LED indicators                           |
|  | Camera interface                         |
|  | Reset button, Boot mode selection button |
|  | USB ports                                |

Tabla 5.10. Explicación interfaces señaladas (24)



Figura 5.11. Anverso placa desarrollo ESP32-S3 WROOM-1



Figura 5.12. Reverso placa desarrollo ESP32-S3 WROOM-1

La pantalla de 2,8 pulgadas es del fabricante Freenove. Incorpora un controlador táctil capacitivo FT6336U, fabricado por FocalTech, usualmente utilizado en teléfonos móviles. También incluye un controlador de pantalla TFT LCD ST7789V, del fabricante Sitronix, que permite mostrar la información deseada a color, RGB (24). Su resolución es de 240 (horizontal) x 320 (vertical).

TFT significa en inglés *Thin Film Transistor*, esto quiere decir que es una pantalla matriz en la que cada píxel se ilumina de forma individual, ya que cada píxel posee un transistor conmutador que recibe la señal eléctrica y determina el color y el brillo de su píxel correspondiente.

LCD significa en inglés *Liquid Crystal Display* y esto quiere decir que este tipo de pantalla está formado por dos láminas de vidrio con una sustancia de cristal líquido en medio, que al aplicar un voltaje cambia la orientación de esos cristales y permite que luz polarizada atraviese la pantalla en distintos grados.

Por tanto, la combinación de ambas permite crear una pantalla con mejor calidad de visualización, ya que al ser cada píxel controlado individualmente se controla mejor la luz que atraviesa los cristales líquidos y se pueden obtener mejores colores e imágenes con mayor nitidez.



Figura 5.13. Anverso pantalla



Figura 5.14. Reverso pantalla

La cámara del dispositivo incorpora el sensor de imagen OV2640. Esto es un sensor de 0,25 pulgadas y resolución de 2 megapíxeles, que captura imágenes en formato RGB y YUV, lo cual proporciona flexibilidad y calidad en la captura y procesamiento de imágenes. La resolución máxima de captura es de 1600 x 1200, conocida como UXGA (25).



Figura 5.15. Anverso cámara



Figura 5.16. Reverso cámara

Si se desea, se ofrece la opción de montar la cámara algo más alejada de ESP32-S3, empleando la placa extensora y el cable FPC para conectar la cámara con esta placa (24).



Figura 5.17. Cable FPC cámara y placa extensora cámara

Continuando con el convertidor y amplificador de audio, cuyas interfaces son (24):

- **Interfaz de altavoces:** La placa dispone de dos grupos de altavoces, Grupo L (L+ y L-) y Grupo R (R+ y R-), para conectar los altavoces de los canales izquierdo y derecho.
- **Interfaz de auriculares:** interfaz para conectar los auriculares.
- **Interfaz de entrada I<sup>2</sup>S:** se conecta a un dispositivo con I<sup>2</sup>S, transcodificando los datos de audio en señales de audio DAC.
- **Interfaz de entrada de audio externa:** se conecta a un equipo de audio externo. Amplifica las señales de audio de entrada externa.
- **Interfaz de alimentación:** se conecta a una fuente de alimentación externa. La fuente de alimentación externa seleccionada debe estar dentro del rango de 3,3V-5,0V.



Figura 5.18. Anverso convertidor y amplificador de audio

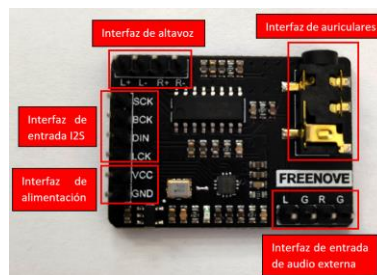


Figura 5.19. Reverso convertidor y amplificador de audio

También se tienen dos altavoces que se conectan a la placa:



Figura 5.20. Altavoces

Como componente adicional, el kit incluye un módulo MAX30102 para medir la frecuencia cardíaca humana y la saturación de oxígeno en sangre.

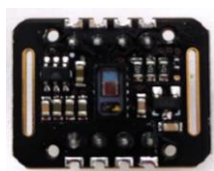


Figura 5.21. Anverso pulsómetro



Figura 5.22. Reverso pulsómetro

Cable para alimentar el circuito con una batería de 9 V:



Figura 5.23. Cable de batería de 9 V

ESP32-S3 proporciona dos formas de acceder a la tarjeta SD: mediante la interfaz SPI o la interfaz SDMMC.

La configuración SPI requiere 4 pines de E/S para acceder a la tarjeta SD, mientras que la interfaz SDMMC es compatible con los modos de bus de uno y cuatro bits. En el modo de un bit de SDMMC, solo se necesitan 3 pines de E/S para acceder a la tarjeta SD, mientras que en el modo de cuatro bits de SDMMC, se necesitan 6 pines de E/S.

Todos estos métodos permiten acceder a la tarjeta SD, pero difieren en la velocidad. El modo de bus de cuatro bits de SDMMC ofrece la velocidad de lectura y escritura más rápida, seguido por el modo de un bit de SDMMC, que es aproximadamente un 80% más lento que el modo de cuatro bits. El modo SPI es el más lento, proporcionando solo la mitad de la velocidad del modo de bus de cuatro bits de SDMMC.

Para la mayoría de las aplicaciones, se recomienda usar el modo de un bit, ya que requiere menos pines de E/S y aun así ofrece un rendimiento adecuado (24).



*Figura 5.24. Tarjeta SD y lector de tarjeta*

Por último, se incluyen algunos tornillos y separadores de latón para fijar correctamente algunos de los componentes a la placa.



*Figura 5.25. Tornillos y separadores de latón*

## 6. PROGRAMACIÓN Y SOFTWARE

### 6.1. Programación del dispositivo

El ESP32 es un microcontrolador muy versátil que admite varios lenguajes de programación. Los tres más comunes son Arduino, MicroPython y ESP-IDF, siendo este último un framework para ESP32, no un lenguaje de programación en sí.

#### 6.1.1. Arduino

Arduino es un lenguaje de programación para el desarrollo de aplicaciones en microcontroladores, que se basa en C/C++, que son lenguajes de programación compilados (26).

Un lenguaje de programación compilado es aquel en el que el código fuente escrito por los desarrolladores se traduce previamente a un código de máquina específico para una plataforma o sistema operativo concreto. Este proceso de traducción se realiza mediante un programa conocido como compilador. Una vez que el código ha sido compilado, puede ser ejecutado directamente por la máquina, lo que generalmente resulta en un rendimiento más rápido y eficiente en comparación con los lenguajes interpretados.

Es un lenguaje sencillo con una amplia variedad de librerías instalables para facilitar la interacción con diversos componentes electrónicos y para realizar funciones específicas. Cuenta además con una amplia comunidad de desarrolladores.

Su estructura básica se compone de dos funciones principales:

- **setup():** Esta función se ejecuta una sola vez al inicio del programa. Aquí se configuran los pines de entrada/salida, se inicializan las librerías necesarias y se preparan los recursos para el funcionamiento del programa.
- **loop():** Esta función se ejecuta repetidamente en un ciclo infinito, formando el núcleo de la lógica del programa.

#### 6.1.2. MicroPython

MicroPython es una implementación del lenguaje de programación Python 3, optimizada para ejecutarse en microcontroladores y entornos de sistemas embebidos con recursos limitados.

Al ser un lenguaje interpretado, similar a Python, el código se ejecuta directamente sin necesidad de compilación previa. Esto facilita la iteración rápida durante el desarrollo, ya que los cambios en el código se pueden probar inmediatamente sin tener que pasar por un proceso de compilación (26).

Incluye también una serie de bibliotecas y módulos específicos para ESP32, proporcionando funcionalidades para interactuar con su hardware (27).

Sin embargo, a pesar de estas ventajas, MicroPython puede no ser la opción más recomendable para programar el ESP32 en comparación con Arduino. Esto se debe a varias razones (26):

- **Rendimiento:** dado que MicroPython es un lenguaje interpretado, puede no ofrecer el mismo nivel de rendimiento que un lenguaje compilado como C/C++ utilizado en Arduino.
- **Compatibilidad de librerías:** aunque MicroPython tiene una serie de bibliotecas y módulos para ESP32, la cantidad y variedad de bibliotecas disponibles no es tan extensa como en Arduino.
- **Experiencia del entorno:** Arduino cuenta con un entorno de desarrollo más establecido, respaldado por una gran cantidad de usuarios, una documentación detallada y amplio soporte técnico.

### 6.1.3. ESP-IDF

ESP-IDF es un framework o marco de desarrollo oficial para el ESP32, proporcionado por *Espressif*, fabricante de ESP32. Ofrece un conjunto de herramientas para programar ESP32 en C/C++, consiguiendo un control más fino y un mejor rendimiento para todas las funcionalidades avanzadas del hardware (26).

Posee un soporte nativo para FreeRTOS, un sistema operativo en tiempo real para microcontroladores que proporciona los siguientes métodos (28):

- **Multitarea:** permite la ejecución de múltiples tareas (también conocidas como hilos) de manera simultánea. Esto es útil en aplicaciones donde varias operaciones independientes deben ser ejecutadas simultáneamente. En el caso de este proyecto, para que se ejecute la tarea de mostrar y cargar la pantalla con la interfaz de usuario independientemente de la tarea de la cámara y decodificación del libro.

- **Planificación de tareas:** incluye un planificador de tareas que decide qué tarea debe ejecutarse en un momento dado. Se basa en especificar las prioridades de tareas, con tareas de mayor prioridad que se ejecutan antes que las de menor prioridad.
- **Sincronización de tareas:** proporciona mecanismos para la sincronización de tareas, como semáforos y colas. Estos son útiles cuando las tareas deben coordinarse entre sí, por ejemplo, cuando una tarea depende de los resultados de otra.
- **Gestión de memoria:** incluye una gestión de memoria eficiente que es adecuada para sistemas con recursos limitados. Pudiendo elegir la cantidad de memoria asignada a una tarea en concreto.
- **Asignación de tareas a núcleos específicos:** en microcontroladores de múltiples núcleos, como el ESP32-S3, FreeRTOS permite asignar tareas a núcleos específicos, especialmente cuando las tareas son independientes. De manera que, si una tarea consume muchos recursos de CPU, asignarla a un núcleo separado evita que ralentice a las demás, permitiendo que las tareas menos intensivas se ejecuten sin interrupciones. En este caso, las tareas principales para la pantalla se ejecutan en el núcleo principal, es decir, el núcleo 1, mientras que la tarea de la cámara y decodificación de código QR se ejecutará en el núcleo 0.

Sin embargo, a pesar de todas estas ventajas, ESP-IDF puede ser más complejo y difícil de usar que Arduino y MicroPython, especialmente para los principiantes o aquellos que no están familiarizados con la programación en C/C++.

La razón principal de esta complejidad es que ESP-IDF proporciona un control más detallado sobre el hardware del ESP32, esto significa que los desarrolladores necesitan tener un conocimiento más profundo del hardware y de cómo interactuar con él. Además, ESP-IDF requiere la configuración manual de varios aspectos del sistema, como la gestión de la memoria y la multitarea, que son manejados automáticamente en Arduino y MicroPython. Esto puede aumentar la complejidad del desarrollo, especialmente para proyectos más grandes.

#### *6.1.3.1. Capa de compatibilidad con Arduino*

Por otro lado, ESP-IDF ofrece una capa de compatibilidad con Arduino que funciona como un puente entre Arduino y el framework ESP-IDF. Esta capa abstrae la complejidad del ESP-IDF, facilitando la programación del ESP32 con Arduino y a la vez brindando acceso a sus funciones avanzadas.

Funciona de la siguiente manera: cuando se compila el código escrito en Arduino para ESP32, este se transforma en código de máquina comprensible por el microcontrolador. Durante este proceso, la capa de compatibilidad se encarga de traducir las funciones y métodos de Arduino utilizados en el código a las funciones y métodos correspondientes de ESP-IDF (29).

Por lo tanto, al programar en Arduino y usar la capa de compatibilidad para que se traduzca a ESP-IDF, se puede aprovechar la simplicidad y facilidad de uso de Arduino, al tiempo que se tiene acceso a las características y capacidades avanzadas de ESP32 a través de ESP-IDF.

#### *6.1.4. Decisión sobre el lenguaje a utilizar*

Aunque ESP-IDF ofrece más control sobre el hardware, puede ser más difícil de usar.

Sin embargo, su capa de compatibilidad con Arduino permite un equilibrio entre facilidad de uso y acceso a funciones avanzadas. Por lo tanto, Arduino es la opción elegida por su facilidad de uso, librerías, gran comunidad de desarrolladores, documentación y acceso a las capacidades avanzadas del ESP32 a través de ESP-IDF.

Finalmente, tras decidir el lenguaje de programación, es necesario analizar las diferentes herramientas software posibles para encontrar aquella que se adecue satisfactoriamente a la programación de este TFG.

### **6.2. Entorno de desarrollo**

Entre las diferentes posibilidades se estudiarán las siguientes: Arduino IDE, CLion IDE y Visual Studio Code. Donde cabe destacar que se programará en lenguaje Arduino, con lo que esas tres plataformas son compatibles.

#### *6.2.1. Características comunes*

Todos estos entornos de desarrollo comparten una serie de aspectos favorables al programador y características comunes.

Todos estos IDEs ofrecen un espacio para escribir código con resaltado de sintaxis, sugerencias de código y otras funciones útiles para facilitar la escritura.

Permiten organizar proyectos de manera estructurada, con la opción de agregar varios archivos de código, bibliotecas y recursos adicionales, lo que hace más fácil trabajar en proyectos grandes.

Disponen de funcionalidades de depuración incorporadas que facilitan la detección y solución de fallos mediante la creación de puntos de interrupción, la revisión de variables y la ejecución secuencial del código.

Tanto CLion IDE como Visual Studio Code ofrecen compatibilidad con Git y otros sistemas de gestión de versiones, lo cual es esencial para monitorear modificaciones en el código y facilitar la colaboración entre desarrolladores.

Son compatibles con Windows, macOS y Linux, lo que permite desarrollar en el sistema operativo que prefieras.

### 6.2.2. Comparativa y elección de entorno

Para comprobar el funcionamiento de estas tres plataformas se propone el caso de realizar un programa sencillo que muestre la frase “Hello world” en la pantalla.



Figura 6.1. Resultado del programa que se realizará

Se presenta el mismo programa para estas tres plataformas. La programación para Arduino IDE difiere muy poco respecto CLion IDE y VSCode. Mientras que los programas en estos dos últimos softwares son iguales, solo cambia la interfaz gráfica.

La principal ventaja que representan CLion y VSCode respecto Arduino IDE es la integración de control de versiones mediante Git y la facilidad de manejar

librerías mediante PlatformIO. Esto es una plataforma que incluyen estos programas, de código abierto que facilita la creación, compilación, depuración y carga de código en diversas placas de desarrollo. En lo que concierne a las librerías del proyecto, PlatformIO permite buscar, instalar y gestionar las librerías directamente desde el entorno de desarrollo, ofreciendo una gran variedad de bibliotecas comúnmente utilizadas en proyectos de software embebido.

Arduino IDE es fácil y simple de usar, aunque carece de algunas características más avanzadas que requerirían proyectos más complejos.

CLion y VSCode son mejores alternativas para proyectos más completos, ya que poseen funcionalidades extras que Arduino IDE no tiene, como mayor análisis del código, soporte avanzado para depuración, mayor flexibilidad para organizar archivos, mayor personalización del entorno de trabajo y apoyo para la sintaxis.

Con estos motivos queda claro que CLion y VSCode son mejores opciones para proyectos de mayor envergadura.

Finalmente, para decidir el mejor camino entre estos dos softwares, cabe destacar que ambos son muy similares, sin embargo, la interfaz de CLion es más sencilla e intuitiva y muestra las librerías y partes del código que no se utilizan en color gris, aspecto que VSCode no proporciona. Además, indica los atributos que se deben añadir a las funciones.

Por ello, se tendría una mejor visión y control sobre el código realmente útil. CLion IDE será la plataforma utilizada en adelante para programar este proyecto.

```
1  #include "display.h"
2  #include <lvgl.h>
3  #include <TFT_eSPI.h>
4  #include "FT6336U.h"
5
6  Display screen;
7  void function_hello_world(void);
8
9  void setup() {
10     Serial.begin(115200);
11     screen.init();
12     function_hello_world();
13 }
14
15 void loop() {
16     screen.routine(); /* let the GUI do its work */
17     delay( 5 );
18 }
19
20 void function_hello_world(void)
21 {
22     /*Change the active screen's background color*/
23     lv_obj_set_style_bg_color(lv_scr_act(), lv_color_hex(0x13D4C0), LV_PART_MAIN);
24
25     /*Create a white label, set its text and align it to the center*/
26     lv_obj_t * label = lv_label_create(lv_scr_act());
27     lv_label_set_text(label, "Hello world");
28     lv_obj_set_style_text_color(lv_scr_act(), lv_color_hex(0xffffffff), LV_PART_MAIN);
29     lv_obj_align(label, LV_ALIGN_CENTER, 0, 0);
30 }
```

*Figura 6.2. Programa "Hola Mundo" en Arduino IDE*

```

1  #include <Arduino.h>
2  #include "display.h"
3  #include <lvgl.h>
4  #include <TFT_eSPI.h>
5  #include "FT6336U.h"
6
7  Display screen;
8  void function_hello_world(void);
9
10 void setup() {
11     Serial.begin(115200);
12     screen.init();
13     function_hello_world();
14 }
15
16 void loop() {
17     screen.routine(); /* let the GUI do its work */
18     delay( 5 );
19 }
20
21 void function_hello_world(void)
22 {
23     /*Change the active screen's background color*/
24     lv_obj_set_style_bg_color(lv_scr_act(), lv_color_hex(0x13D4C0), LV_PART_MAIN);
25
26     /*Create a white label, set its text and align it to the center*/
27     lv_obj_t * label = lv_label_create(lv_scr_act());
28     lv_label_set_text(label, "Hello world");
29     lv_obj_set_style_text_color(lv_scr_act(), lv_color_hex(0xffffffff), LV_PART_MAIN);
30     lv_obj_align(label, LV_ALIGN_CENTER, 0, 0);
31 }

```

Figura 6.3. Programa “Hola Mundo” en Visual Studio Code

```

1  #include <Arduino.h>
2  #include "display.h"
3  #include <lvgl.h>
4  #include <TFT_eSPI.h>
5  #include "FT6336U.h"
6
7  Display screen;
8  void function_hello_world(void);
9
10 void setup() {
11     Serial.begin( baud: 115200);
12     screen.init();
13     function_hello_world();
14     Serial.print("HOLA");
15 }
16
17 void loop() {
18     screen.routine(); /* let the GUI do its work */
19     delay( 5 );
20 }
21
22 void function_hello_world(void)
23 {
24     /*Change the active screen's background color*/
25     lv_obj_set_style_bg_color( obj: lv_scr_act(), value: lv_color_hex( c: 0x13D4C0), selector: LV_PART_MAIN);
26
27     /*Create a white label, set its text and align it to the center*/
28     lv_obj_t * label = lv_label_create( parent: lv_scr_act());
29     lv_label_set_text( obj: label, text: "Hello world");
30     lv_obj_set_style_text_color( obj: lv_scr_act(), value: lv_color_hex( c: 0xffffffff), selector: LV_PART_MAIN);
31     lv_obj_align( obj: label, align: LV_ALIGN_CENTER, x_ofs: 0, y_ofs: 0);
32 }

```

Figura 6.4. Programa “Hola Mundo” en CLion IDE

## 6.3. PlatformIO

### 6.3.1. Introducción

Como herramienta adicional para facilitar la programación se ha decidido utilizar el entorno de desarrollo integrado (IDE) de código abierto PlatformIO, cuyo núcleo está programado en lenguaje Python.

La principal ventaja que se obtiene de su uso en este trabajo es la gestión de dependencias automatizada que ofrece, lo cual permite instalar y actualizar de manera directa y sencilla las librerías necesarias.

Es compatible con múltiples plataformas y entornos de desarrollo y está ideado para utilizarse en proyectos con sistemas embebidos, tales como microcontroladores o placas de desarrollo.

También ofrece facilidades para compilar el código y cargarlo en el dispositivo, sin necesidad de utilizar línea de comandos, simplemente pulsando un botón (30).

De esta manera, se encargará de instalar el compilador y todas las librerías que se necesiten, facilitando su actualización cuando los desarrolladores actualicen la versión de la biblioteca.

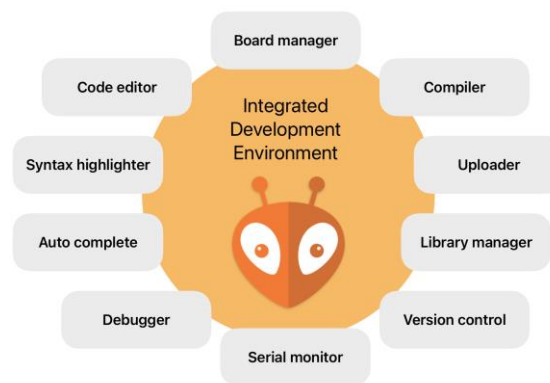


Figura 6.5. Funciones de PlatformIO

### 6.3.2. Configuración

Tras la instalación de PlatformIO en CLion IDE, aparece en la carpeta del proyecto su archivo de configuración, que se utiliza para especificar la configuración del proyecto.

```
[env:esp32-s3-devkitm-1]
platform = espressif32
board = esp32-s3-devkitm-1
framework = arduino
lib_deps =
    bodmer/TFT_eSPI@^2.5.43
    lvgl/lvgl@^8.3.11
    espressif/esp32-camera@^2.0.4
    mobizt/Firebase Arduino Client Library for
ESP8266 and ESP32@^4.4.14
    bblanchon/ArduinoJson@^7.0.4
    freenove/Freenove WS2812 Lib for ESP32@^1.0.6
    alvarowolfx/ESP32QRCodeReader@^1.1.0
monitor_speed = 115200
build_flags =
```

```

-I include/
-D ST7789_DRIVER=1
-D USER_SETUP_LOADED=1
-D TFT_WIDTH=320
-D TFT_HEIGHT=240
-D TFT_RGB_ORDER=TFT_BGR
-D TFT_INVERSION_OFF=1
-D TFT_MOSI=20
-D TFT_SCLK=21
-D TFT_CS=47
-D TFT_DC=0
-D LOAD_FONT2=1
-D LOAD_FONT4=1
-D SPI_FREQUENCY=80000000
-D I2C_SCL=1
-D I2C_SDA=2
-D RST_N_PIN=-1
-D INT_N_PIN=-1
-D LV_CONF_INCLUDE_SIMPLE=1
-D CORE_DEBUG_LEVEL=5
-D BOARD_HAS_PSRAM=1
-D TOUCH_CS=-1

```

La línea “[env:esp32-s3-devkitm-1]” define el entorno de construcción del proyecto. Ese entorno es la configuración específica para poder compilar y cargar el código en un tipo de placa de desarrollo específica. En el tutorial que proporciona el fabricante se indica que se elija la placa de desarrollo ESP32S3 Dev Module.

De ese tipo existen:

- **ESP32-S3-DevKitC-1 (31):** placa de desarrollo equipada con el módulo ESP32-S3-WROOM-1, ESP32-S3-WROOM-1U o ESP32-S3-WROOM-2. Estos módulos tienen hasta 36 GPIOs, flash hasta 16 MB, PSRAM hasta 16 MB, 512 KB de SRAM, 384 KB de ROM.
- **ESP32-S3-DevKitM-1 (32):** placa de desarrollo equipada con el módulo ESP32-S3-MINI-1 o ESP32-S3-MINI-1U. Estos módulos tienen hasta 39 GPIOs, flash hasta 8 MB, PSRAM hasta 2 MB, 512 KB de SRAM, 384 KB de ROM.

Ya que el módulo de la placa de desarrollo que se está utilizando es ESP32-S3-WROOM-1, se podría pensar que lo adecuado es elegir ESP32-S3-DevKitC-1. Aunque, pensando en que posteriormente se deberá realizar el reconocimiento del código del libro, es más recomendable elegir la opción que incluye PSRAM para disponer de más memoria en el momento de la decodificación, ya que es un proceso costoso, a pesar de que no se refiera al módulo que realmente se utiliza.

Siguiendo con la configuración del archivo de PlatformIO, se define la plataforma hardware en la que se desarrolla el proyecto, que en este caso es “*espressif32*”, es la que incluye los microcontroladores ESP32.

Luego se define la placa de desarrollo, que anteriormente se explicó la elección de ESP32-S3-DevKitM-1.

El *framework* es la estructura base para desarrollar software, de manera que proporciona herramientas y estructuras ya predefinidas para programar y no tener que crear todo el software desde el inicio. Los dos frameworks que se pueden seleccionar desde PlatformIO para programar ESP32 son Arduino y ESP-IDF. Al elegir el *framework* de Arduino y la plataforma de *espressif32*, el compilador de PlatformIO entiende que se está programando en Arduino un dispositivo ESP32 de Espressif y, por tanto, hace uso de la capa de compatibilidad de ESP-IDF con Arduino, emulando las características y funciones de Arduino mediante llamadas a ESP-IDF.

Posteriormente, se indican las librerías que se necesitan para el proyecto, la velocidad del monitor serie en baudios para la comunicación serie entre la placa de desarrollo y el ordenador que la programa (para enviar y recibir datos entre la placa y el ordenador).

En el apartado de “*build\_flags*” se establecen una serie de banderas o *flags*, que se utilizan para proporcionar información adicional al compilador durante el proceso de compilación.

La bandera “*-I include*” indica al compilador que busque archivos de cabecera en el directorio “*include*”.

La bandera “*-D*” se utiliza para definir un símbolo con un valor específico. Por ejemplo, “*-D ST7789\_DRIVER=1*” define el símbolo “*ST7789\_DRIVER*” con el valor “*1*”.

Las banderas “*-D*” están configurando varios parámetros, como las dimensiones de la pantalla TFT (“*TFT\_WIDTH*” y “*TFT\_HEIGHT*”), los pines de la interfaz SPI utilizada por la pantalla TFT (“*TFT\_MOSI*”, “*TFT\_SCLK*”, “*TFT\_CS*”, “*TFT\_DC*”), la frecuencia de la interfaz SPI (“*SPI\_FREQUENCY*”) y los pines de la interfaz I2C (“*I2C\_SCL*”, “*I2C\_SDA*”), entre otros.

Estas banderas son específicas para el proyecto y pueden variar dependiendo de las necesidades de la aplicación.

## 7. RECONOCIMIENTO Y DIFERENCIACIÓN DE LIBROS

### 7.1. Planteamiento inicial y alternativa

#### 7.1.1. Problema de enfoque

Uno de los fines de este trabajo es reconocer el ejemplar de libro que se muestre a la cámara del ESP32-S3. La manera más común y que se ideó en un primer momento es leer el código de barras que se encuentra en la parte posterior de todos los libros. Sin embargo, se ha encontrado el problema de que al leer este tipo de códigos con la cámara conectada al ESP32-S3 se obtiene una imagen con una calidad muy baja, lo cual dificulta gravemente la detección y decodificación del código.



*Figura 7.1. Imagen tomada del ISBN con la cámara del ESP32-S3*

Esto se debe a que, al ser un código pequeño formado por líneas y espacios verticales delgados en una dimensión, se necesita acercarlo mucho a la cámara, lo cual afecta considerablemente al enfoque con la lente usada.

En la hoja de características del sensor OV2640 se indica la distancia de enfoque y la profundidad de campo de este.

- **Distancia de enfoque (33):** es la distancia que hay entre la lente y el punto que se desea enfocar con clara nitidez. Para OV2640, esta distancia es de 120 cm a 150 cm.
- **Profundidad de campo (33):** es el rango de distancias en las que la imagen aparece nítida en general. Para OV2640 este rango es a partir de 60 cm.

Esto significa que para este sensor los objetos colocados a partir de 60 cm se verán generalmente nítidos y se enfocará claramente aproximadamente a 120 cm. No obstante, el fabricante no ha proporcionado información más específica sobre la lente, únicamente se tiene la información genérica sobre el sensor que utiliza. Por tanto, la

manera de comprobar la distancia de enfoque para esta lente en concreto es aproximada y debe ser de manera manual.

Tras realizar varios experimentos acercando objetos a la cámara se ha comprobado que su distancia de enfoque se encuentra en torno a 10 cm, por ello al tener que acercar el ISBN del libro mucho más, se obtiene una imagen tan desenfocada.

### 7.1.2. Solución al problema de enfoque: código bidimensional

Este problema se podría solucionar utilizando una lente con una distancia mínima de enfoque más corta o si se sigue trabajando con los elementos del kit ya adquiridos existe otra alternativa, escanear un código en dos dimensiones y cuadrado, en concreto uno de los más conocidos, el código QR o *Quick Response* en inglés.

Los códigos QR, al ser bidimensionales, pueden ser leídos desde distancias mayores en comparación con los códigos de barras. Además, su estructura matricial en módulos facilita la lectura con cámaras con resolución más baja. Esta es la opción con la que se trabajará.



Figura 7.2. Diferencia entre almacenaje de datos entre QR y código de barras (34)

#### 7.1.2.1. Códigos bidimensionales

Los códigos en dos dimensiones no están formados por barras, sino por otros elementos geométricos como puntos o cuadrados. La información se codifica en dos direcciones, horizontal y vertical. Surgen de la necesidad de almacenar una mayor cantidad de información (desde 25 caracteres hasta 7000, además de otros tipos de datos como direcciones de páginas web, fotografías y muchos más elementos) (35).

- **Data Matrix** (36): se utiliza en el área de producción y distribución para etiquetar principalmente elementos electrónicos y productos farmacéuticos.

## Data Matrix



- Para codificar texto o datos en masa en un solo código.
- Dígitos: hasta 3116.
- Alfanuméricos: hasta 2335.

*Figura 7.3. Ejemplo código Data Matrix (36)*

- **PDF 417** (36): se utiliza en numerosas aplicaciones, como transporte y seguimiento de paquetes, billetes de transporte, tarjetas de identificación, gestión de inventario y sistemas de pago móvil. Su flexibilidad permite ajustarlos a la cantidad de información requerida, expandiéndolos o reduciéndolos según lo que sea necesario.

## PDF 417



- Hasta 1800 caracteres ASCII por código de barras.
- Hasta 1100 caracteres binarios por símbolo.

*Figura 7.4. Ejemplo código PDF 417 (36)*

- **Aztec** (36): tienen un diseño eficiente que reduce su tamaño en comparación con otros tipos de códigos de barras bidimensionales. Constan de múltiples anillos concéntricos que rodean un núcleo central. La información se almacena en estos anillos mediante una combinación específica de puntos y líneas. El número de anillos y la concentración de datos se ajustan en función de los requisitos específicos de codificación.

## Aztec



- Dígitos: hasta 3832.
- Alfanuméricos: hasta 3067.

*Figura 7.5. Ejemplo código Aztec (36)*

- **Código QR (Quick Response)** (36): empleado en marketing, publicidad y aplicaciones móviles debido a su capacidad para almacenar una gran cantidad de información y enlaces a sitios web.



- Para codificar texto o datos en masa en un solo código.
- Dígitos: hasta 7089.
- Alfanuméricos: hasta 4296.
- Kanji: hasta 1817.

Figura 7.6. Ejemplo código QR (36)

#### 7.1.2.2. Códigos QR

Se compone de un conjunto de cuadrados blancos y negros que representan información. A la unidad de cuadrado más pequeña se le llama módulo. Contiene mucha más información que los códigos de barras y se puede leer en cualquier dirección, de arriba a abajo, abajo a arriba, izquierda a derecha o derecha a izquierda.

En función de la cantidad de módulos existen distintas versiones, a mayor número más cantidad de datos se pueden codificar y mayor es el QR.

Algunos ejemplos de la información que puede codificar: texto, enlaces web, teléfono, datos de contacto, PDF, WiFi, etc (34).

La estructura general se puede observar en la siguiente figura.

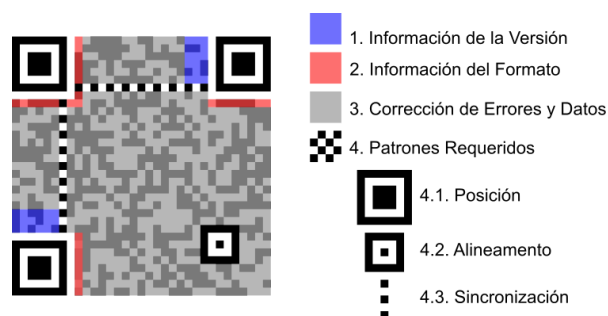


Figura 7.7. Composición QR versión 4, módulos 33x33 (34)

- **Protones de detección de la posición** (37): informan sobre dónde están los bordes del código y se pueda detectar.

- **Patrón de alineamiento** (37): referencia para asegurar que los módulos están alineados.
- **Patrones de sincronización** (37): muestra la posición de las filas y columnas para calcular el tamaño del código.
- **Información de la versión** (37): indica el número de la versión para saber la cantidad de módulos que hay (la versión 4 de la imagen tiene 33x33 módulos).
- **Información del formato** (37): indica cómo se toleran los errores y el patrón para enmascarar los datos.
- **Corrección de datos y errores** (37): parte que contiene el código para detectar y corregir errores. El código que se utiliza para corregir los errores si se encuentra dañado el QR es Reed-Solomon.

## 7.2. Conocer cómo se diferencian los libros

Aunque no se vaya a escanear el código de barras del libro, sino un QR generado para el libro, es necesario conocer sobre identificador de cada libro, conocido como ISBN (International Standard Book Number).

Un código de barras es una representación gráfica de datos numéricos o alfanuméricos que consta de una serie de barras y espacios paralelos de diferentes anchuras. Estos códigos se utilizan para identificar de forma única productos o artículos asignando a cada uno un código específico.

Su uso principal es facilitar y agilizar los procesos de identificación, seguimiento, inventario y ventas en diversos sectores comerciales, industriales y logísticos.

Son escaneados por un dispositivo óptico, obteniendo la información sobre el producto e identificándolo. Los detalles que proporciona el procesamiento e identificación del código pueden ser el precio, descripción, número de serie, fecha de fabricación y otros datos relevantes (36).

### 7.2.1. Tipos de códigos de barras

Los códigos de barras son lineales (la información solo se lee en un sentido) y están compuestos por líneas verticales paralelas de diferente espacio y grosor, que codifican la información generalmente en dirección horizontal. Representan información hasta un máximo de 25 caracteres (35).

- **Código de barras EAN (*European Article Number*)** (36): es el código más utilizado en Europa, principalmente en supermercados. Su fiabilidad en la lectura es superior a otras opciones. Este código se compone de elementos como el código de país, el del fabricante, el del producto y un dígito de verificación. Existen dos variantes:
  - EAN-13, con 13 dígitos, es la versión estándar.
  - EAN-8, con 8 dígitos, se usa para artículos pequeños.



Figura 7.8. Ejemplo código EAN-13 (36)

- **Código de barras UPC (*Universal Product Code*)** (36): se utiliza principalmente en el comercio al por menor, sobre todo en América del Norte, y es el precursor del EAN en Europa y del JAN en Japón. El formato UPC-A, que se usa únicamente en Estados Unidos y Canadá, no incluye un dígito para el país. En cambio, su primer número, el carácter del sistema numérico, define el tipo de producto. También hay dos variantes de este código.
  - UPC-A o UPC-12, la versión estándar de 12 dígitos.
  - UPC-E, una versión abreviada de 6 dígitos.



Figura 7.9. Ejemplo código UPC-A (36)

- **ITF (*Interleaved Two of Five*)** (36): se utiliza como el estándar de distribución impreso en cajas de cartón ondulado. El código "2 of 5" viene en varias variantes,

todas con una composición similar de 2 barras anchas (espacios) de entre 5 barras (espacios), pero cada una representa un carácter diferente.



Figura 7.10. Ejemplo código ITF-14 (36)

- **Code 39 (36):** es uno de los códigos de barras alfanuméricos más antiguos, se utiliza en múltiples sectores industriales. Su principal ventaja radica en su amplio abanico de caracteres y combinaciones posibles, que incluyen números, letras y símbolos diversos como comas, puntos, espacios, entre otros. Sin embargo, su desventaja principal es su baja tolerancia y su limitada densidad de información.



Figura 7.11. Ejemplo Code 39 (36)

- **Codabar (36):** es un código numérico de baja densidad de información, que incluye seis caracteres especiales adicionales, que van desde 0 hasta 9, además de signos como el guion, dos puntos, barra diagonal, punto, más y porcentaje. Cada carácter se representa mediante siete elementos, cuatro trazos y tres huecos. Se suele utilizar en aplicaciones que necesitan número de serie, como la administración de reservas de sangre, comprobantes para servicios de reparto a domicilio y carnés de socio.



Figura 7.12. Ejemplo Codabar (36)

- **Code 128 (36):** es un código de barras muy flexible y capaz de codificar todos los caracteres ASCII. Es comúnmente utilizado en la gestión de inventarios y transporte. Debido a su capacidad para representar una amplia gama de caracteres, exceptuando algunos específicos del japonés, es compatible con los sistemas informáticos. Dado que el CODE 128 utiliza cuatro tamaños de barra diferentes, se necesitan impresoras de alta calidad para imprimirlo correctamente. No es recomendable utilizar impresoras de matriz de puntos, de inyección de tinta FA o flexográficas en cartón corrugado para imprimir el CODE 128.



*Figura 7.13. Ejemplo Code 128 (36)*

### 7.2.2. Cómo funciona el código de barras

Todos los códigos de barras se componen de una serie de barras verticales negras y blancas de distintos grosores. Las líneas negras representan un “1”, mientras que los espacios en blanco representan un “0”. Así, las barras negras y los espacios blancos se suelen agrupar en conjuntos de siete en siete para formar un número binario, que se traduce a un número decimal del 0 al 9 (38).

A pesar de los distintos tipos de códigos, generalmente siguen una estructura común.

1. **Quiet zone (39):** es un margen en blanco que se debe dejar a los laterales del código para separar el código de otros elementos que puedan interferir en la lectura del escáner.
2. **Carácter inicial y final (39):** representan el comienzo y el fin de los datos. Pueden variar en función del tipo de código.
3. **Dígito de control:** sirve para validar el código de barras y comprobar que se ha producido una decodificación correcta. Por ejemplo, este dígito se calcula de la siguiente manera para EAN-13 (40):

- Se realiza la suma de los dígitos situados en posiciones pares.
- Se calcula la suma de los dígitos en posiciones impares.
- El total de las posiciones impares se multiplica por tres.
- Al resultado de multiplicar se le añade la suma de las posiciones pares.
- El número resultante se redondea al múltiplo de diez más cercano y superior.
- Del número redondeado se sustrae la suma de los resultados de los pasos “a” y “d”.

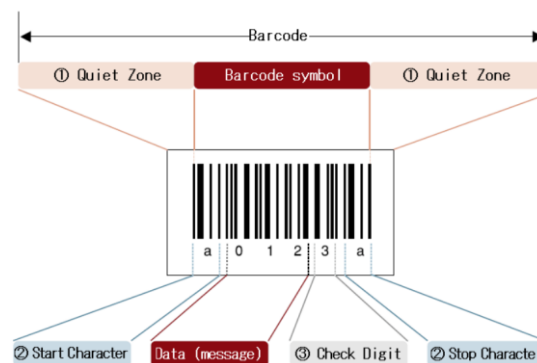


Figura 7.14. Estructura general código de barras (39)

### 7.2.3. ISBN (EAN-13)

En este caso, el tipo de código de barras que interesa es el ISBN. Este es el identificador para gestionar libros y demás productos editoriales y está homologado según el EAN-13 desde el año 2007 (posteriormente seguía la norma de EAN-10).

Está formado por 13 cifras y generalmente comienza por el prefijo 978, para indicar que se trata de un código de barras de productos de lectura.

Su estructura se compone de 5 partes bien diferenciadas (41):

- **Prefijo EAN:** longitud de 3 dígitos, es la cifra 978, que indica que se trata de un libro. También puede ser 979.
- **Grupo de registro:** longitud entre 1 y 3 dígitos. Indica la región geográfica o área lingüística a la que pertenece el artículo. En España se utiliza el 84.
- **Editor:** longitud entre 1 y 7 dígitos. Determina la editorial a la que pertenece.
- **Número de publicación:** longitud entre 1 y 6 dígitos. Indica la edición del libro.
- **Dígito de control:** un único dígito que verifica que el ISBN es válido.



Figura 7.15. Estructura ISBN de 13 dígitos (41)

Este código de barras tiene 95 áreas o módulos de igual anchura. Cada área puede ser blanca (representada por el 0) o negra (representada por el 1) y las áreas continuas forman una barra blanca o negra, estando compuesto así un código EAN-13 o ISBN por 59 barras (42).

Para saber cómo se decodificaría es importante conocer la distribución de estas áreas. De izquierda a derecha, se puede encontrar (42):

- 3 áreas para la guarda de inicio (101).
- 42 zonas a la izquierda (siete por cifra) para codificar las cifras 2ª a 7ª. Cada cifra está representada por cuatro barras. La primera cifra puede deducirse a partir de las 6 cifras.
- 5 áreas para la guarda central (01010).
- 42 zonas a la derecha (siete por cifra) para codificar las cifras 8ª a 13ª. Cada cifra está representada por cuatro barras.
- 3 áreas para la guarda final (101).

A continuación, se muestra de manera gráfica el proceso de decodificación de un código de barras EAN-13, mismo formato que el ISBN.

La codificación de los dígitos se puede apreciar en la siguiente tabla:

| Digit | Left-hand (Odd) | Left-hand (Even) | Right-hand |
|-------|-----------------|------------------|------------|
| 0     | 0001101         | 0100111          | 1110010    |
| 1     | 0011001         | 0110011          | 1100110    |
| 2     | 0010011         | 0011011          | 1101100    |
| 3     | 0111101         | 0100001          | 1000010    |
| 4     | 0100011         | 0011101          | 1011100    |
| 5     | 0110001         | 0111001          | 1001110    |
| 6     | 0101111         | 0000101          | 1010000    |
| 7     | 0111011         | 0010001          | 1000100    |
| 8     | 0110111         | 0001001          | 1001000    |
| 9     | 0001011         | 0010111          | 1110100    |

Tabla 7.1. Codificación dígitos parte derecha e izquierda EAN-13 (42)

| First digit | The parity of the 6 left-hand digits |
|-------------|--------------------------------------|
| 0           | O O O O O O                          |
| 1           | O O E O E E                          |
| 2           | O O E E O E                          |
| 3           | O O E E E O                          |
| 4           | O E O O E E                          |
| 5           | O E E O O E                          |
| 6           | O E E E O O                          |
| 7           | O E O E O E                          |
| 8           | O E O E E O                          |
| 9           | O E E O E O                          |

Tabla 7.2. Cálculo del primer número del código (42)

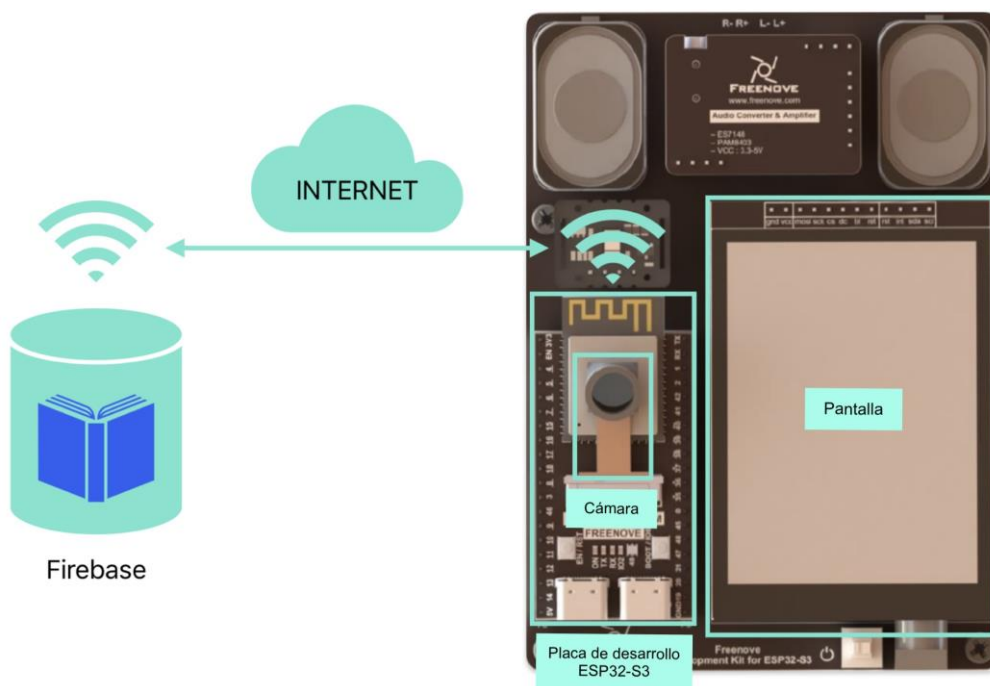


Figura 7.16. Código de barras EAN-13 decodificado

## 8. DISEÑO E IMPLEMENTACIÓN

### 8.1. Arquitectura general del software

El sistema está compuesto por varios componentes que interactúan entre sí para proporcionar la funcionalidad deseada. A continuación, se presenta un diagrama de la arquitectura general del sistema:



*Figura 8.1. Esquema de la arquitectura general del sistema*

En el diagrama presentado, se pueden observar los siguientes elementos:

- **Placa de desarrollo de ESP32-S3:** placa de desarrollo del fabricante Freenove que contiene al módulo empleado ESP32-S3-WROOM-1-N8R8, que a su vez incluye el chip microcontrolador ESP32-S3R8 que sirve como el “cerebro” del sistema. Se encarga de controlar todos los demás componentes y de ejecutar el software principal.
- **Cámara:** se encuentra conectado al ESP32 y se utiliza para capturar imágenes que luego se procesan para detectar y decodificar los códigos QR.
- **Conexión WiFi:** el ESP32 se conecta a la red local a través de WiFi. Esta conexión se utiliza para enviar y recibir datos a través de Internet.

- **Internet:** representada como una nube en el diagrama, Internet es la red global que permite al ESP32 comunicarse con el servidor Firebase y su base de datos en tiempo real.
- **Servidor Firebase:** este es el servidor en la nube donde se almacenan y se recuperan los datos sobre los libros del usuario. El ESP32 se comunica con este servidor para enviar y recibir la información deseada.

Cada uno de estos componentes juega un papel crucial en el funcionamiento del sistema. En las siguientes secciones, se describirá en detalle el funcionamiento del sistema.

### 8.1.1. Librerías

Como se indicó previamente el entorno de desarrollo es CLion junto con PlatformIO.

El proyecto se sirve de un conjunto de 7 librerías que aportan distintas funcionalidades para facilitar la programación del equipo:



Figura 8.2. Conjunto de librerías necesarias para el proyecto

- **ArduinoJson:** se utiliza para manejar los datos que se reciben de la base de datos en formato JSON (formato de datos ligero y fácil de leer que se suele utilizar en la web), facilitando la deserialización (proceso de convertir datos de un formato, JSON en este caso, a una estructura de datos que se pueda manejar en el código) para poder manejar y acceder correctamente a ellos.
- **Firebase Arduino Client Library for ESP8266 and ESP32:** se utiliza para que el ESP32-S3 pueda interactuar y comunicarse con la base de datos de Google Firebase. Con esta librería se permite la autenticación del usuario mediante correo electrónico y contraseña y el manejo de la base de datos en tiempo real que permite que se acceda a los datos de los libros cuando se desee y se modifique la página actual de lectura.

- **ESP32QRCodeReader:** se utiliza para capturar las imágenes del QR que se muestre a la cámara conecta al ESP32-S3 y se decodifique, siendo capaz de almacenar su valor y poder utilizarlo para buscar el libro en la base de datos.
- **esp32-camera:** proporciona el código para manejar gran cantidad de sensores de cámara, como OV2640, OV3660, OV5640, OV7670 y muchos más. Entre sus funciones se encuentran la de capturar imágenes en diferentes formatos (JPEG, PNG, RAW), ajustar la resolución, visualizar imágenes, etc.
- **lvgl:** se utiliza para crear y gestionar los elementos de la interfaz gráfica, tales como etiquetas, botones, colores, formas, mostrar imágenes. Es lo que permite crear la interfaz de usuario para que sea atractiva e interactiva.
- **TFT\_eSPI:** se utiliza para poder trabajar con la pantalla TFT utilizando la interfaz de periféricos en serie (SPI), para poder comunicar el microcontrolador con periféricos externos, como la pantalla TFT. Esto permite su inicialización, orientación o rotación de la pantalla, dibujar, etc.
- **Freenove WS2812 Lib for ESP32:** se utiliza para controlar el LED WS2812 que tiene incorporada la placa, permitiendo controlar su brillo y color. Este elemento es un LED RGB (rojo-verde-azul) que permite controlar cada color individualmente, además de su nivel de brillo, pudiendo establecerse desde 0 hasta 255 (256 niveles de brillo). Este será utilizado para facilitar la labor de escaneo de la cámara si hay poca iluminación, encendiéndose cuando la cámara empieza a capturar y apagándose posteriormente.

### 8.1.2. Código fuente

El resto del código del proyecto se encuentra en una carpeta fuente en la que se incluyen los archivos código fuente “.cpp” o “.c” y los correspondientes archivos de cabecera “.h”.

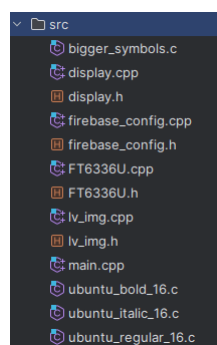


Figura 8.3. Archivos de código fuente

En concreto se tienen cuatro archivos escritos en lenguaje C para cargar distintos tipos de fuente a la interfaz de usuario: estilo regular, estilo negrita, estilo itálica y estilo regular más grande para representar símbolos. Para generarlas se ha utilizado un convertidor de fuente compatible con LVGL.

El archivo principal y punto de entrada al programa se denomina *main.cpp* y en él se declaran los pines de la cámara, la pantalla, el lector de código de QR y demás variables y funciones para desarrollar la interfaz gráfica que se muestra en la pantalla y con la que interacciona el usuario.

Finalmente, el resto de los archivos son dos de código fuente en C++ y sus respectivas cabeceras proporcionados por el fabricante para el manejo más específico de su modelo de pantalla. Estos archivos proporcionan métodos para controlar e interactuar con el panel táctil FT6336U de la pantalla y para iniciar la pantalla y actualizarla según las acciones del usuario.

## **8.2. Interfaz de usuario**

### **8.2.1. LVGL**

Toda la GUI ha sido creada y diseñada utilizando LVGL, por lo que antes de comenzar a crear la aplicación es necesario comprender bien esta librería.

LVGL significa Biblioteca de Gráficos Ligera y Versátil, en inglés *Ligth and Versatile Graphics Library*, es una librería de código abierto escrita en C que permite crear interfaces de usuario atractivas para dispositivos embebidos, siendo dispositivos embebidos todos aquellos que están diseñados para realizar una o pocas funciones específicas, estando generalmente integrado en un dispositivo más grande, pudiendo considerarse entonces que ESP32 es un sistema embebido y que LVGL es apto para ello.

Ofrece una amplia gama de widgets (como botones, listas, gráficos, etc.) que se pueden personalizar modificando el color, tamaño, fuente, estilo, etc.

También utiliza un sistema de coordenadas cartesianas para posicionar los elementos en la pantalla (el origen, cero en X y cero en Y, se encuentra en la esquina superior izquierda).

Es independiente del hardware, se puede usar con cualquier microcontrolador y es escalable, capaz de funcionar con poca memoria (64 KB de Flash, 16 KB de RAM)

Además, proporciona un sistema de manejo de eventos para responder a las interacciones del usuario, como por ejemplo pulsaciones de botones o desplazamientos de listas, siendo capaz de registrar funciones de devolución de llamada (*callbacks*) que se ejecutarán cuando se produzcan estos eventos.

También permite cargar nuevas pantallas y limpiar el contenido de la anterior para no agotar los recursos con cada nueva pantalla que se va creando, en función de las interacciones del usuario a través de las distintas pestañas.

En conclusión, en este TFG se utiliza LVGL para crear la interfaz de usuario incluyendo la creación de varias pantallas, la adición de elementos a esas pantallas y la configuración de las funciones de devolución de llamada para manejar las interacciones del usuario.

### *8.2.2. Funcionamiento general de la interfaz gráfica*

Una parte crucial de este proyecto es la interfaz gráfica de usuario (GUI), que proporciona una forma intuitiva y eficiente para los usuarios de interactuar con el sistema. Es el punto de contacto entre el usuario y el sistema, y su diseño y funcionalidad pueden tener un impacto significativo en la facilidad de uso y la eficiencia.

En las siguientes secciones, se describe en detalle cómo se ha diseñado e implementado la GUI para este TFG y cómo interactúa con el resto del sistema.

#### *8.2.2.1. Introducción a la GUI*

En la interfaz gráfica de este proyecto, se ha implementado una barra de navegación que consta de cuatro secciones distintas. Esto se ha realizado creando un objeto *lv\_obj\_t* con la vista de pestañas *lv\_tabview* de LVGL, para posteriormente crear cada una de estas pestañas como otro objeto asignándolo a la vista de pestañas *tabview*.

El contenido de cada pestaña se genera mediante funciones específicas. Estas funciones toman como argumento el objeto de la pestaña correspondiente y añaden a este objeto los elementos de la interfaz de usuario que deben aparecer en la pestaña.

Al seleccionar una sección en la barra de navegación, el usuario es llevado a la pestaña correspondiente, activando un manejador de eventos que limpia el contenido de la pestaña seleccionada para volver a cargarlo y actualizar dinámicamente así algún dato que haya podido cambiar.

Dentro de cada pestaña, existen varias pantallas por las que se pueden navegar. Cada una de estas pantallas tiene una función específica, permitiendo al usuario interactuar con diferentes aspectos de la aplicación dependiendo de la pantalla que esté visualizando en ese momento.

Para mejorar la experiencia del usuario y hacer la navegación más intuitiva, cada sección en la barra de navegación está representada por un color distinto. Esto no solo hace que la interfaz sea más atractiva visualmente, sino que también ayuda a identificar y recordar más fácilmente las diferentes secciones de la aplicación.

En la siguiente tabla se ofrece a modo de resumen la estructura que sigue cada pestaña y la función genérica de cada una de las pantallas que ofrece. Todos los apartados tienen dos pantallas, excepto la tercera sección, que se compone de tres pantallas distintas.

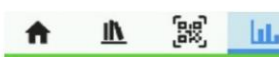
| Interfaz de usuario                                                                 |           |            |                                                                                                                                |
|-------------------------------------------------------------------------------------|-----------|------------|--------------------------------------------------------------------------------------------------------------------------------|
|  | Pestaña 1 | Pantalla 1 | Bienvenida al usuario (indica a modo de juego en qué nivel está el lector)                                                     |
|                                                                                     |           | Pantalla 2 | Breve explicación al usuario sobre interfaz                                                                                    |
|  | Pestaña 2 | Pantalla 1 | Lista con todos los libros del usuario que se encuentran en la base de datos                                                   |
|                                                                                     |           | Pantalla 2 | Información personalizada sobre el libro seleccionado de la lista                                                              |
|  | Pestaña 3 | Pantalla 1 | Introducción previa a escanear el libro                                                                                        |
|                                                                                     |           | Pantalla 2 | Texto e imagen descriptivos para indicar que ya se puede escanear el libro                                                     |
|                                                                                     |           | Pantalla 3 | Información sobre libro escaneado y teclado numérico para modificar página actual                                              |
|  | Pestaña 4 | Pantalla 1 | Estadísticas básicas de usuario (lectura actual, páginas leídas, libros empezados, finalizados, sin empezar)                   |
|                                                                                     |           | Pantalla 2 | Gráfica de barras para mostrar el porcentaje leído de cada libro (con leyenda para indicar a qué libro corresponde cada barra) |

Tabla 8.1. Breve resumen sobre el funcionamiento general de la interfaz

#### 8.2.2.2. Gamificación

Un aspecto central de este proyecto es la incorporación de elementos de gamificación para fomentar la lectura.

La gamificación se refiere al uso de elementos de diseño de juegos en contextos no lúdicos, en cualquier ámbito, con el fin de mejorar la realización y comprensión de tareas, aumentar la motivación de las personas, obtener mejores resultados en el sector en el que se aplique, aumento de la participación y productividad del usuario (43).

La gamificación se puede aplicar a una gran variedad de ámbitos:

- **Educación:** para hacer el aprendizaje más divertido y atractivo para los estudiantes.
- **Formación empresarial:** para formar a los empleados en nuevas habilidades o aumentar la motivación.
- **Marketing:** para crear campañas de marketing más atractivas y efectivas.
- **Salud:** para motivar a las personas a adoptar hábitos de vida saludables.

Para incorporar la gamificación, algunas de las técnicas más empleadas son las que se muestran en la siguiente imagen:



*Figura 8.4. Principales técnicas para aplicar la gamificación (43)*

En este caso, se ha implementado un sistema de escalado de niveles por acumulación de páginas leídas.

Las personas se sienten motivadas por el deseo de demostrar competencia y lograr el éxito, por ello un sistema de niveles puede satisfacer este deseo al proporcionar

una serie de metas (es decir, niveles) que el usuario puede esforzarse por alcanzar y demostrar así su éxito.

Este método funciona de la siguiente manera: a medida que los usuarios leen más páginas, avanzan a niveles superiores. Este progreso se refleja en la interfaz de usuario, proporcionando a los usuarios una retroalimentación visual de su progreso y logros en el inicio en la pantalla principal. El objetivo de este sistema es motivar a los usuarios a leer más, a utilizar la aplicación de manera más frecuente y a fomentar un sentido de logro y progreso.

Son en total 6 niveles. Se comienza en el nivel aprendiz, hasta llegar al máximo nivel, nivel sabio, tras alcanzar 1500 páginas leídas. Este umbral para un lector habitual puede ser relativamente bajo, pero ya que este TFG está enfocado a lectores poco habituales para incentivar su hábito de lectura, estos niveles están adecuados a ellos.

Debajo del nivel se indica el usuario el número de páginas que le quedan para subir al siguiente nivel, de manera que se le proporciona un objetivo claro. Y para motivar aún más, en la pestaña 3, si se introduce el número correspondiente a la página final aparece una felicitación al lector animándolo a que siga leyendo.

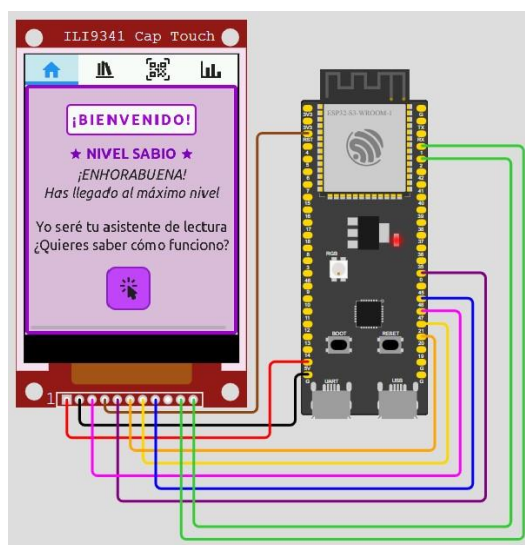


Figura 8.5. Máximo nivel que se puede alcanzar (a partir de 1500 páginas)

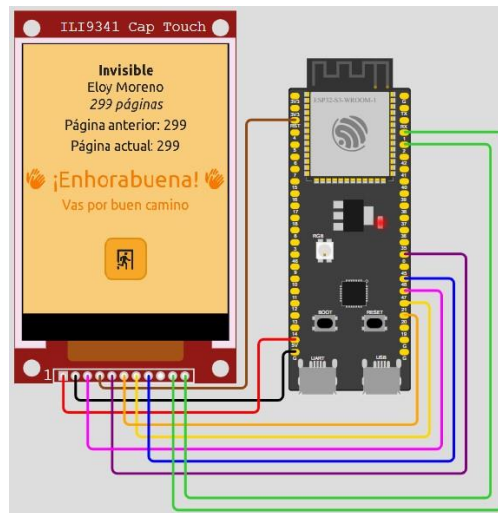


Figura 8.6. Felicitación motivadora tras terminar un libro

En conclusión, se ha propuesto utilizar un sistema de niveles basado en el número de páginas leídas en lugar del número de libros leídos, que podría ser otra alternativa. Sin embargo, a continuación, se analizará la decisión de utilizar el número de páginas para implementar la gamificación:

**Mayor precisión en la medición del progreso:** el número de páginas proporciona una medida más precisa del progreso del usuario en comparación con el número de libros leídos. Esto se debe a que la longitud de los libros varía considerablemente, generando distorsiones en la percepción del avance si se basa únicamente en la cantidad de libros completados, ya que un usuario que lee libros cortos podría avanzar más rápido en términos de cantidad de libros, sin haber dedicado el mismo tiempo o esfuerzo que otro usuario que lee libros más extensos.

**Equidad en la gamificación:** al basarse en páginas leídas, se establece un sistema de gamificación más equitativo, donde todos los usuarios tienen las mismas oportunidades de avanzar, independientemente de la velocidad a la que leen o la longitud de los libros que eligen.

**Fomenta la lectura continua:** enfocarse en el número de páginas incentiva la lectura constante y regular, independientemente de la longitud de los libros seleccionados. Esto promueve un hábito de lectura más sólido y consistente, ya que el progreso se ve recompensado de manera continua, sin importar el tamaño del libro que se esté leyendo en ese momento.

**Adaptabilidad a diferentes ritmos de lectura:** el sistema basado en páginas se adapta mejor a los diferentes ritmos de lectura de los usuarios. Aquellos que leen más

rápido pueden avanzar más rápidamente en los niveles, mientras que los usuarios que leen a un ritmo más lento pueden progresar a su propio ritmo sin sentirse presionados o desalentados.

#### *8.2.2.3. Realimentación al usuario*

La realimentación al usuario es un componente esencial en cualquier sistema interactivo. Se refiere a la información que un sistema proporciona a los usuarios en respuesta a sus acciones. Además, permite motivar y reforzar la labor del usuario, que en este caso es la lectura.

A través del incremento de niveles que proporciona la herramienta previa de gamificación se aporta una realimentación positiva para reforzar los logros y avances del usuario, destacando en la pantalla principal de la primera pestaña 1 de la interfaz el nivel actual del lector.

Otro tipo de realimentación al usuario es aquella informativa, es decir, se proporciona al usuario en la pestaña 4 información sobre su progreso personal, como el último libro que está leyendo, las páginas leídas, el libro más largo leído, los libros completados, sin empezar o a medias, pudiendo comprobar esta última información de manera gráfica a través de un gráfico de barras que indica el porcentaje de páginas leídas para todos sus libros.

Con todo esto se estimula el interés y la continuidad del usuario.

#### *8.2.3. Pestaña 1 de la interfaz (Inicio)*

Esta parte podría considerarse el menú introductorio o el menú *home*, donde se obtiene una explicación sobre el funcionamiento general de todo el dispositivo y las funciones que ofrece al usuario.

Las pantallas de esta pestaña se representan con el color morado y en la primera de ellas se ofrece la bienvenida con un cartel llamativo que destaca sobre el resto de elementos. Justo debajo se muestra el método de gamificación explicado anteriormente, indicando al usuario su nivel actual, resaltado también de un color diferente al resto de textos para que la vista de la persona se dirija directamente ahí.

Para ofrecer la información correspondiente al nivel, primero se busca el parámetro de la página actual del usuario para cada libro en la base de datos. Una vez se tienen todos esos parámetros, se suman para calcular el total de páginas leídas incluyendo todos los libros, se establece una comparativa con los umbrales

programados para determinar el nivel y se plasma la información en la etiqueta en la pantalla.

Además, se calcula también las páginas restantes para llegar al siguiente nivel para que el usuario lo tenga en cuenta en cualquier momento.

Abajo aparece un botón con un símbolo que indica que se pinche en él. Este botón activa un manejador de eventos que reacciona al tacto del usuario, se oscurece y se desplaza ligeramente para indicar que ha sido pulsado, llevando a una nueva pantalla.

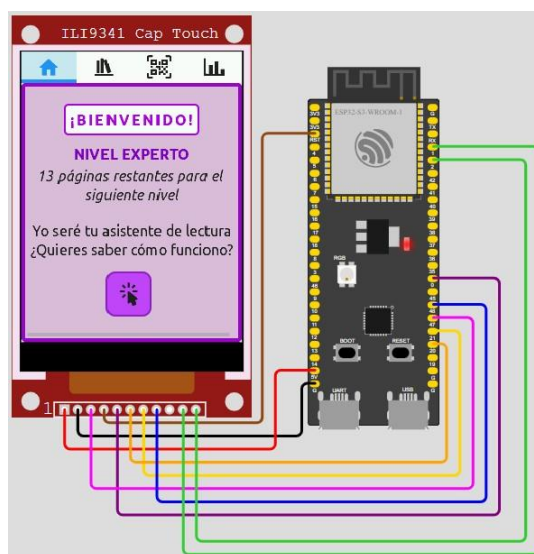


Figura 8.7. Pantalla principal de la pestaña 1

La nueva pantalla secundaria que se crea sigue el mismo estilo de color de esa pestaña y es meramente informativa.

Informa al lector sobre los niveles que puede alcanzar y de las funciones que tiene cada una de las distintas secciones de la aplicación, representada por un símbolo diferente cada una.

Para terminar, al final se incluye otro botón que al ser pulsado dispara el evento para regresar de nuevo a la pantalla principal y limpiar en la que se encontraba el usuario, para que vuelva a aparecer el menú de navegación para moverse entre las distintas pestañas.

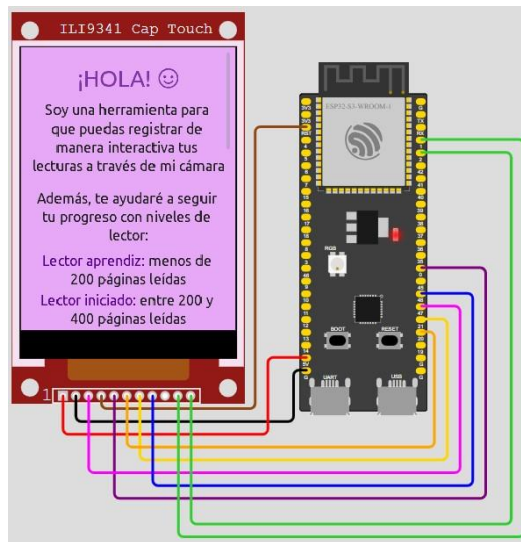


Figura 8.8. Pantalla secundaria explicativa de la pestaña 1

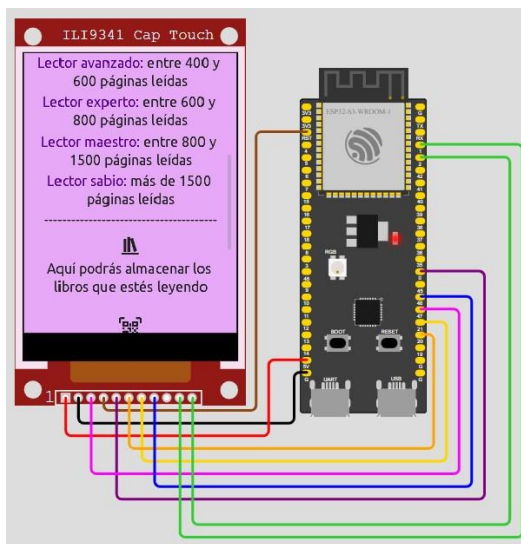


Figura 8.9. Pantalla secundaria explicativa de la pestaña 1

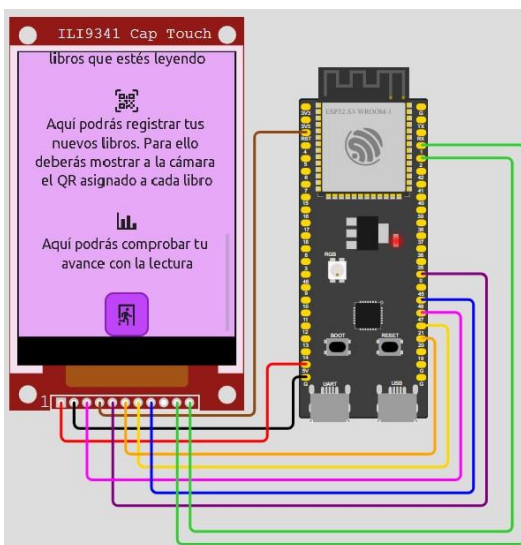


Figura 8.10. Pantalla secundaria explicativa de la pestaña 1

Con el fin de comprender de manera más sencilla e intuitiva esta parte se incluye un ordinograma. Este diagrama sirve como una representación visual de la estructura y la lógica de la aplicación, facilitando la comprensión de su funcionamiento y desarrollo, a cualquier persona independientemente de que esté familiarizado o no con el lenguaje de programación empleado y la implementación de LVGL.

Los símbolos que aparecen en el esquema de esta sección y de las siguientes y su significado son:

- **Óvalo** (44): representa el inicio o el final de un proceso.
- **Rectángulo** (44): representa un proceso o acción específica.
- **Rombo** (44): representa un punto de decisión a partir del cual el camino se puede dividir en dos o más.
- **Cilindro** (44): representa los datos almacenados en la base de datos.
- **Flechas** (44): muestra la dirección del flujo y conecta procesos.

Este ordinograma es una herramienta valiosa para entender cómo se interconectan las diferentes partes de la aplicación y cómo fluye la interacción del usuario a través de la misma. Además, puede servir como una guía útil para la implementación del código.

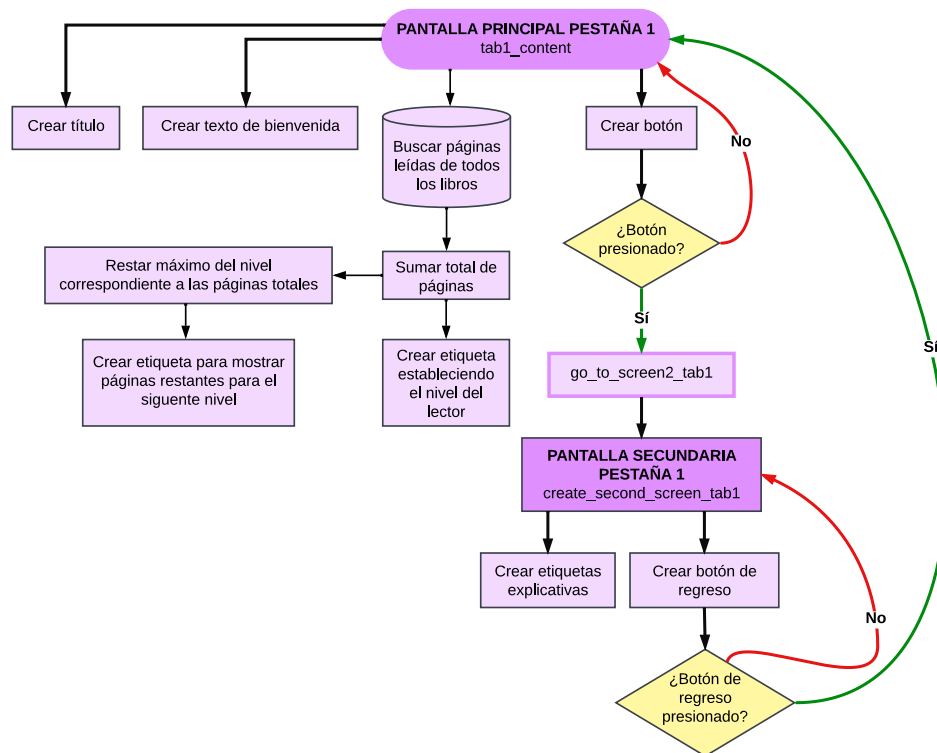


Figura 8.11. Esquema de funcionamiento de la pestaña 1

#### 8.2.4. Pestaña 2 de la interfaz (Biblioteca)

Este apartado se representa con el color azul y la pantalla principal muestra la biblioteca del usuario en un objeto tipo lista de LVGL con los títulos de todos los libros que el lector tiene registrados en la base de datos. Para ello previamente a mostrarlos se debe realizar la consulta a la base de datos y guardar la información en un objeto, para posteriormente acceder a ella.

Pensando en una interacción más cómoda del usuario con todos sus libros, esta lista se ordena según una marca temporal, es decir, desde arriba hacia abajo aparecen desde el más reciente hasta el más antiguo. Así podrá acceder rápidamente sus lecturas actuales.

La marca temporal se actualiza cuando el usuario introduce la página por la que se ha quedado leyendo (en la pestaña 3) y con eso se entiende que ese es el libro que está leyendo actualmente, con lo que es el más reciente y pasaría a mostrarse el primero de la lista.

Además, para poder consultar el título completo, si alguno es muy largo, se ha programado una función de desplazamiento circular para verlo completo de manera dinámica.

Cada título de la lista es un botón que, al ser pulsado, llama al manejador de eventos que lleva al lector a una segunda pantalla mostrando la información acerca de ese libro.

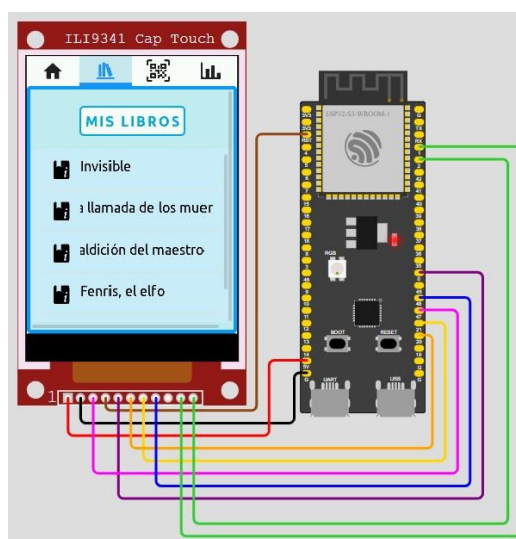


Figura 8.12. Pantalla principal de la pestaña 2

En la pantalla secundaria aparecen los detalles del libro seleccionado, donde se aprecia el título, autor, número total de páginas y página actual. La distinta información se muestra con diferentes estilos de fuente para resaltar cada parte.

Para volver a la sección principal de la pestaña 2 y seguir navegando entre las distintas partes solo se debe pulsar el botón que aparece en la parte baja, que también se oscurece y mueve ligeramente su posición, para indicar que ha sido pulsado. Todos los botones siguen esa dinámica.

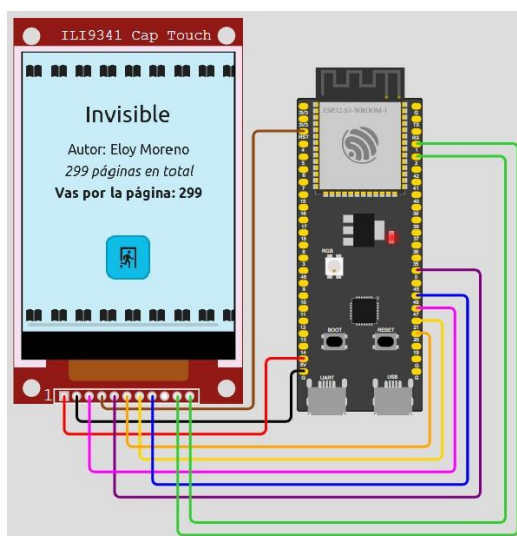


Figura 8.13. Pantalla secundaria de la pestaña 2

Si el título es muy largo y no cabe en la pantalla, también se le ha aplicado el desplazamiento circular para poder visualizarlo por completo. En la siguiente figura se observa el dinamismo de ese largo encabezado.

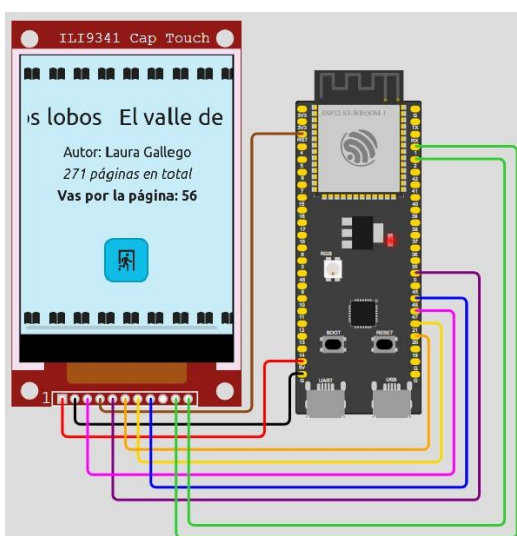


Figura 8.14. Pantalla secundaria de la pestaña 2 (título largo con scroll circular)

Siguiendo la misma filosofía que en el apartado anterior de la pestaña 1, para facilitar el entendimiento de las funciones y creación de estas secciones se proporciona un ordinograma explicativo, siguiendo los mismos colores de la pestaña 2 de la GUI.

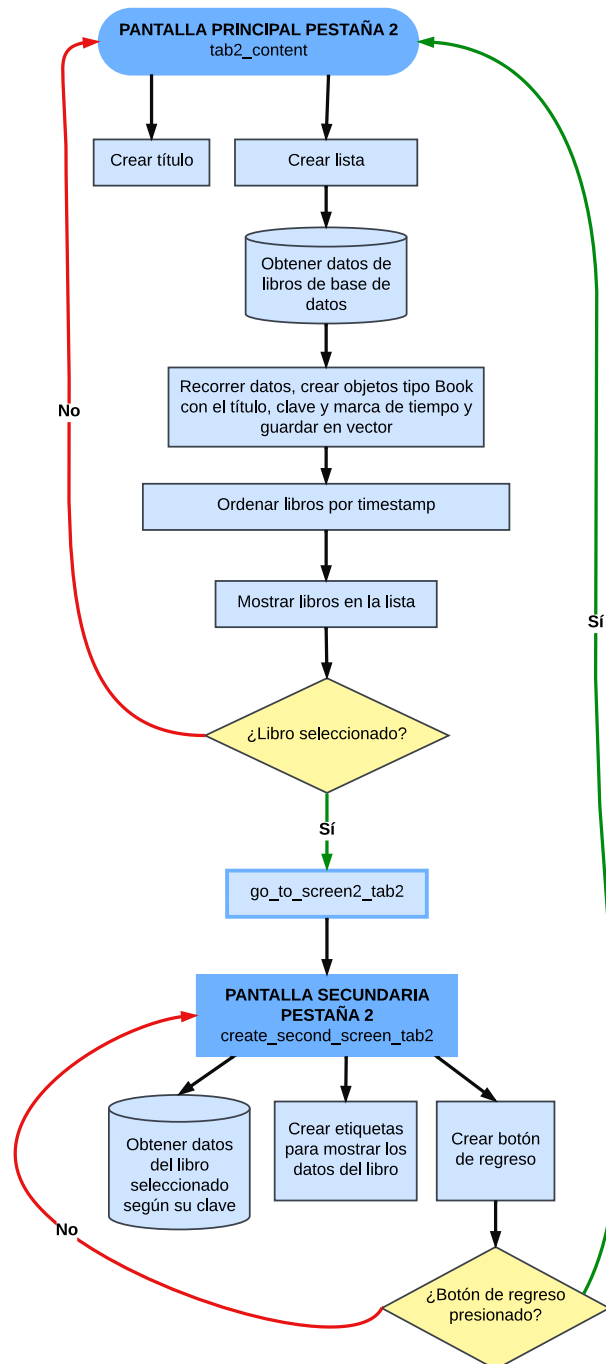


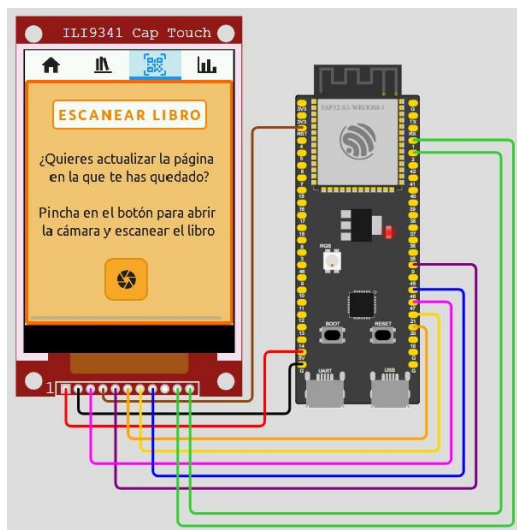
Figura 8.15. Esquema de funcionamiento de la pestaña 2

### 8.2.5. Pestaña 3 de la interfaz (Reconocimiento del libro)

Esta es la parte más compleja de la aplicación, tiene más funciones y en concreto posee una pantalla más que las demás pestañas.

Estas pantallas siguen se muestran siguiendo un patrón naranja de color, al igual que su ordinograma, que se expondrá al final de este apartado.

La pantalla principal desde la que se observa el menú de navegación entre todas las partes de la interfaz, es meramente informativa. Contiene el título, un par de textos explicativos para guiar al usuario y un botón para acceder a la siguiente parte.



*Figura 8.16. Pantalla principal de la pestaña 3*

Al acceder a esta segunda pantalla se activa la cámara del dispositivo, se inicia la tarea de escaneo de código QR, ejecutándose en un núcleo aparte del resto de funcionalidades (se ejecuta en el 0, mientras que el resto de aplicación se ejecuta en el 1, el principal) y se muestra información relevante al usuario, incluyendo una imagen explicativa. Además, se proporciona un botón al deslizar la pantalla hacia abajo para que el usuario pueda regresar a la pantalla principal de esta sección si no desea escanear un QR finalmente.

La tarea de escaneo del código se encarga de decodificar el contenido utilizando su librería correspondiente y almacenarlo en una variable global. Y una vez que se ha decodificado, se para la tarea de escaneo y se guarda la información en una variable para poder realizar una búsqueda del número del ISBN posteriormente en la base de datos.

Por otro lado, como funcionalidad adicional, cuando se inicia el proceso de escaneo del código, el LED WS2812 se enciende automáticamente para iluminar el código QR, para ayudar a mejorar la visibilidad del QR, si se encontrara en un ambiente con poca luz.

Una vez que el escaneo del código QR se ha completado con éxito, el LED se apaga automáticamente, para conservar la energía y no consumir recursos incensarios.

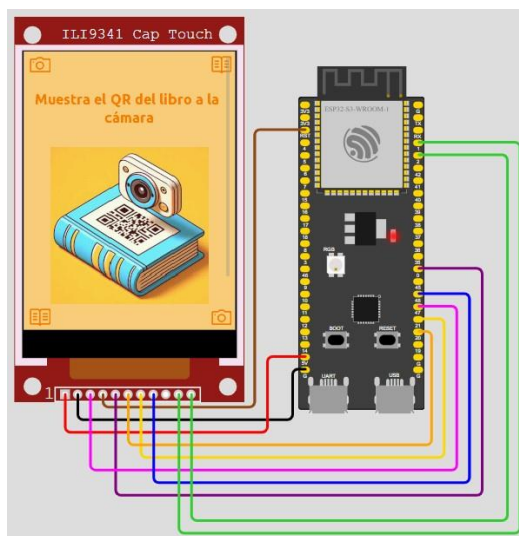


Figura 8.17. Pantalla secundaria de la pestaña 3 (escaneo de QR)

Posteriormente, se realiza una búsqueda en la base de datos para verificar si el ISBN decodificado del código QR existe. Tras finalizar esta búsqueda, se haya encontrado el libro o no, aparece automáticamente la tercera pantalla de la pestaña 3.

Si el libro correspondiente al ISBN existe, se muestra al usuario los detalles del libro y se le proporciona un teclado numérico para que pueda introducir la página actual en la que se encuentra. Tras esto, se debe pulsar el botón que corresponde al símbolo del *tick* y el teclado desaparecerá, para mostrar un mensaje con la página introducida y un botón para devolver al lector a la pantalla principal de la pestaña 3.

Una vez registrada la página actual, ese dato se actualiza automáticamente en la base de datos en tiempo real y la información actualizada podría verse desde las otras pestañas que muestren ese tipo de información. Además, ese libro pasaría a posicionarse en primer lugar en la lista de la biblioteca de la pestaña 2 porque es el más reciente.

Si el usuario introduce por error un número que supera a la cantidad de páginas total que contiene el libro, al pulsar en el botón del *tick*, ese número se cambia por el máximo número de páginas, entendiendo así que el usuario quería introducir la última página del libro porque ya lo ha finalizado.

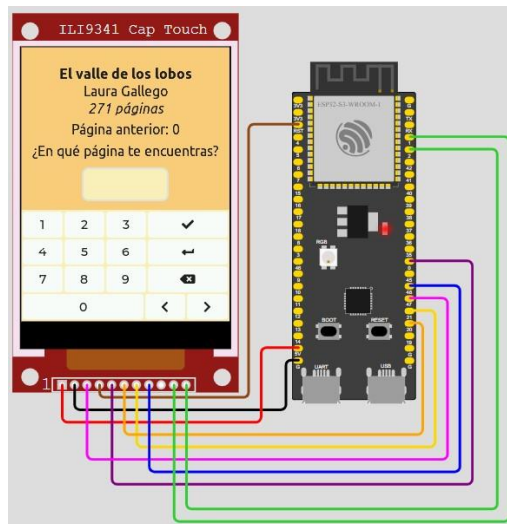


Figura 8.18. Tercera pantalla de la pestaña 3

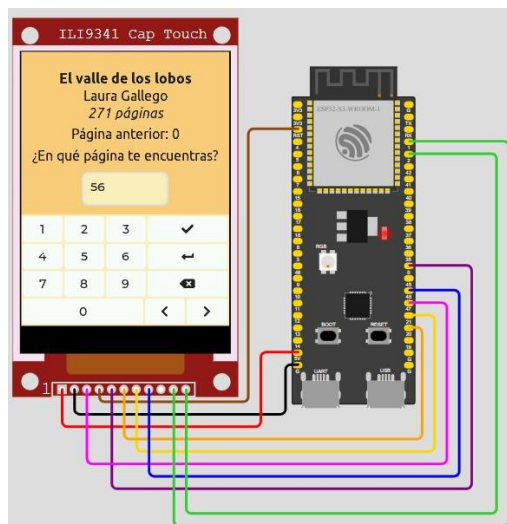


Figura 8.19. Tercera pantalla de la pestaña 3

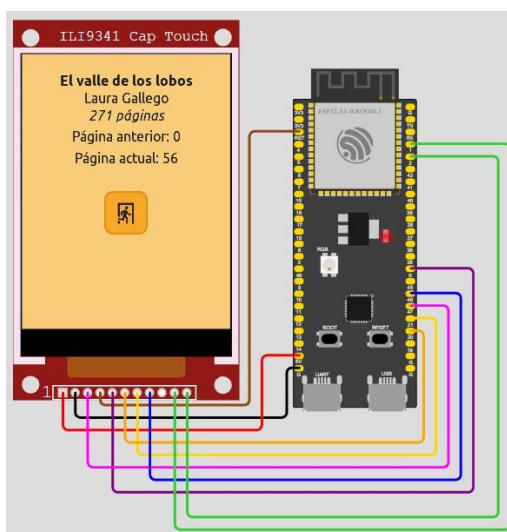
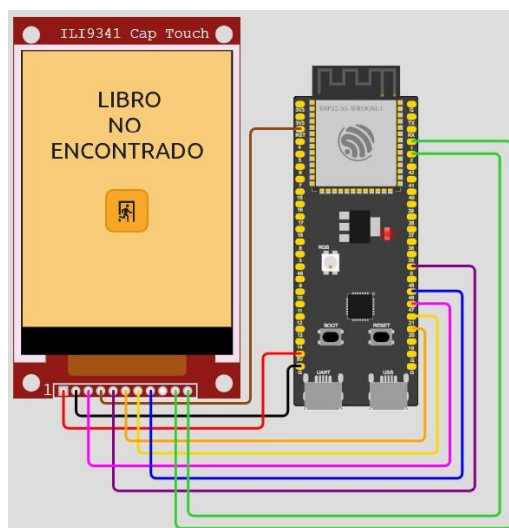


Figura 8.20. Tercera pantalla de la pestaña 3

Si, por el contrario, el libro no existe se muestra un mensaje de error al usuario junto con el botón de regreso a la pantalla principal, indicando que ese libro no se encuentra en la base de datos.



*Figura 8.21. Tercera pantalla de la pestaña 3*

Por último, se muestra en la siguiente imagen el ordinograma correspondiente a todo el funcionamiento de la aplicación en esta tercera pestaña.

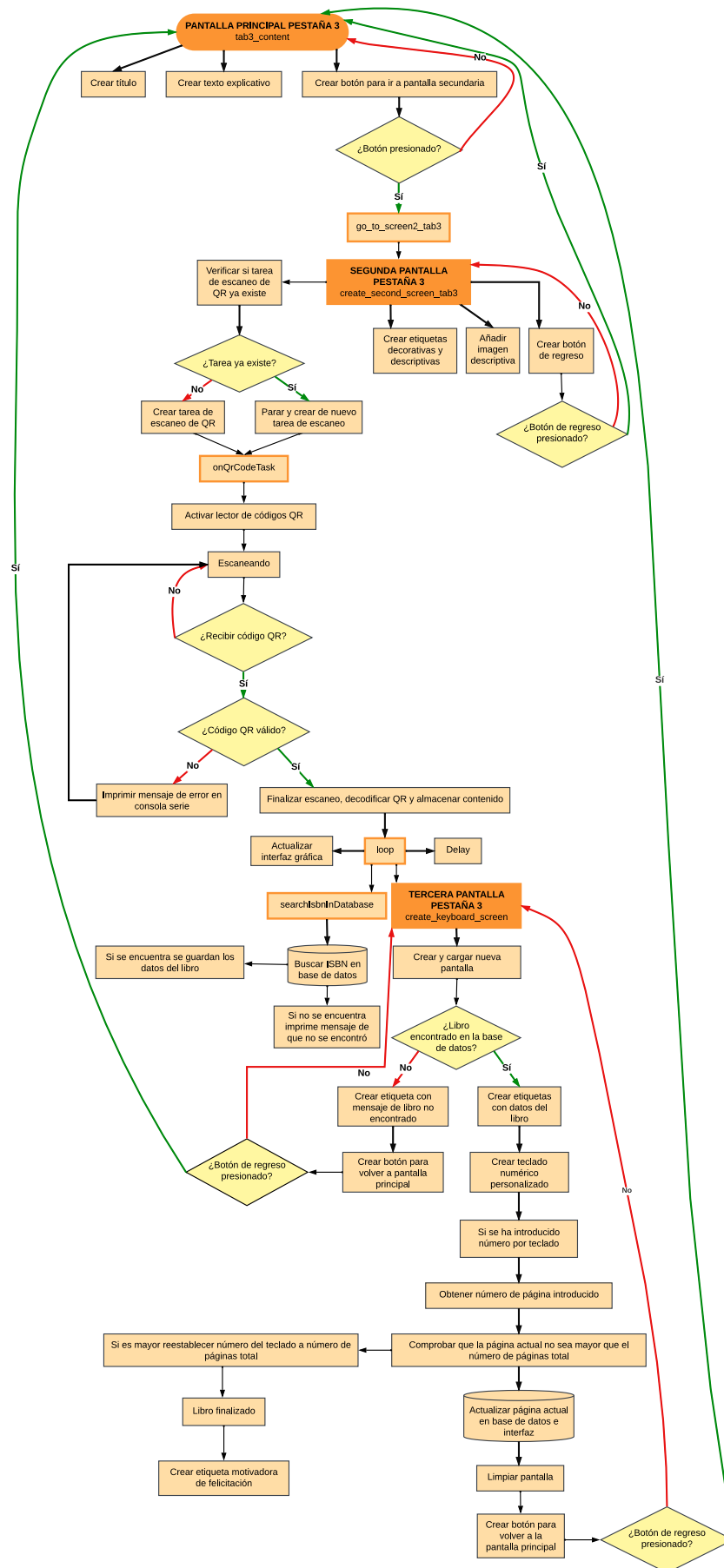


Figura 8.22. Esquema de funcionamiento de la pestaña 3

#### 8.2.6. Pestaña 4 de la interfaz (Estadísticas)

Esta es la última parte de la GUI, que sigue el patrón de color verde y muestra datos relacionados con el avance del lector. Esta funcionalidad permite a los usuarios visualizar información relevante sobre su actividad de lectura.

Los datos que se muestran incluyen aspectos como el número total de páginas leídas, el libro más reciente que han leído, o el libro en el que han leído más páginas, los libros comenzados, sin comenzar y finalizados.

Además, se incluyen varios botones para que el desplazamiento por la pantalla sea más rápido.

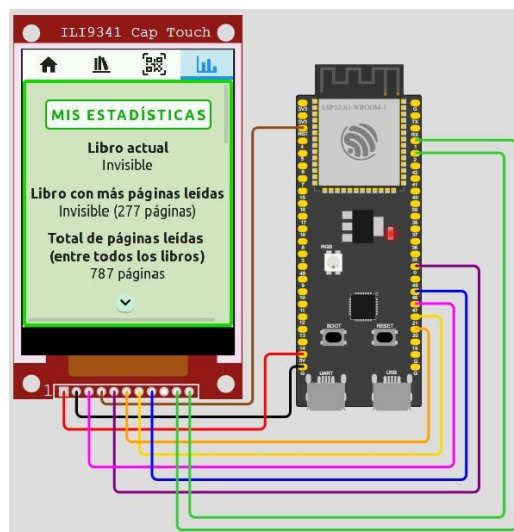


Figura 8.23. Pantalla principal de la pestaña 4

Todos los libros que se encuentren en la base de datos en tiempo real son clasificados en tres categorías (libros iniciados, sin comenzar o finalizados), y cada una de ellas se muestra en un contenedor exclusivo.

Para mostrar los libros que el lector no ha usado todavía se comprueba el número de página actual. Si el usuario se encuentra en la página cero se almacena el título de ese libro en su contenedor correspondiente según su categoría.

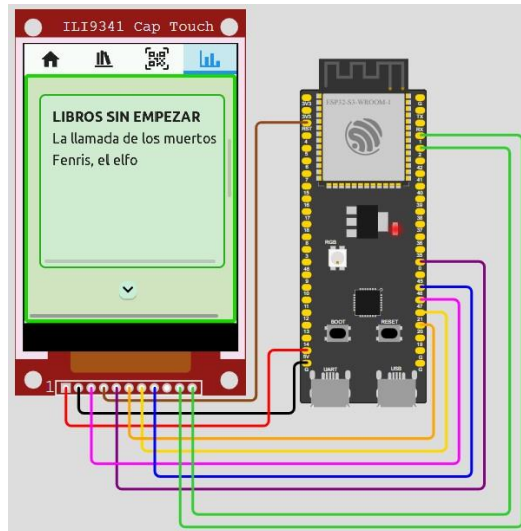


Figura 8.24. Pantalla principal de la pestaña 4

Por el contrario, si la página actual es distinta de cero, pero menor que las páginas totales del libro se clasifica en los libros que ya están comenzados, calculando el porcentaje de páginas leídas. Así se tiene una clara visión sobre el avance en esa categoría.

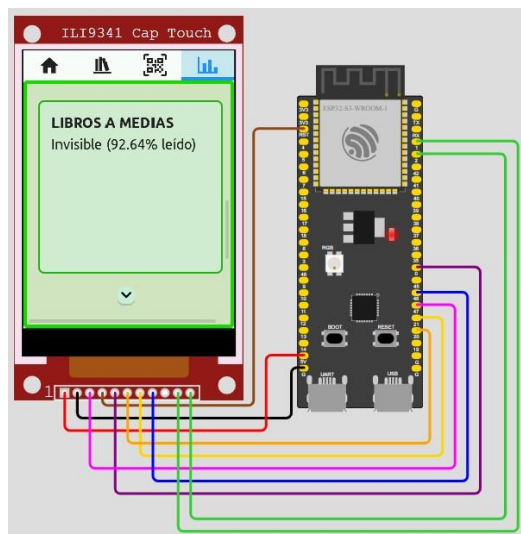


Figura 8.25. Pantalla principal de la pestaña 4

Finalmente, si la página actual es igual a las páginas totales, se entiende que el libro ya ha sido completado y se almacena su título en el último contenedor para la categoría de libros finalizados.

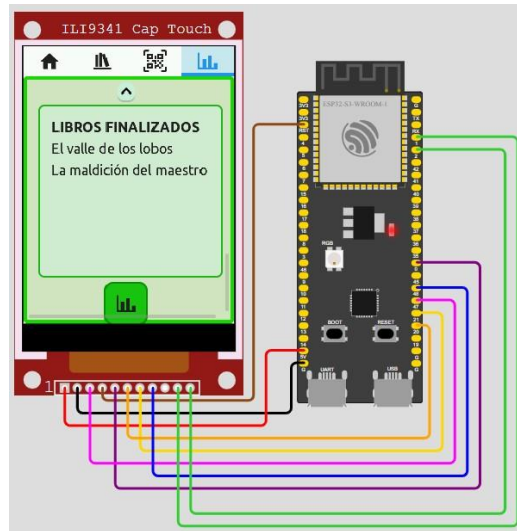


Figura 8.26. Pantalla principal de la pestaña 4

Justo debajo del último contenedor de la pantalla principal se encuentra un botón que cambia a una nueva pantalla en la que se mostrará un gráfico de barras que permite comprobar el porcentaje de páginas leídas para cada uno de los libros de la biblioteca del usuario.

En la parte de arriba de la pantalla secundaria se debe mostrar una leyenda que indica a qué libro se corresponde cada etiqueta del eje X del gráfico, ya que los títulos son mucho más largos y sería ineficiente mostrarlos en ese eje, debido a que se solaparían unos con otros o las barras tendrían que estar demasiado separadas.

También aparece dos botones de navegación rápida para acceder a la leyenda o al gráfico.

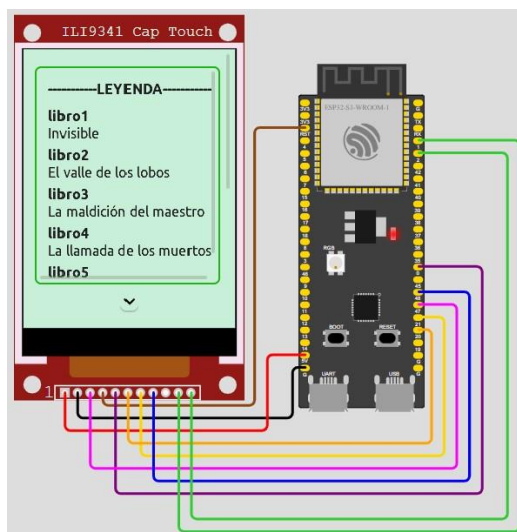


Figura 8.27. Pantalla secundaria de la pestaña 4

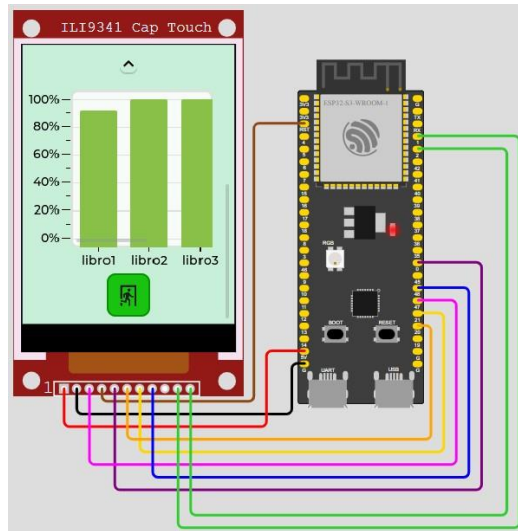


Figura 8.28. Pantalla secundaria de la pestaña 4

Al final del todo, el usuario interactuará con el botón verde oscuro para volver a la pantalla principal de la pestaña 4.

El ordinograma de todas estas funciones se aprecia en la siguiente figura.

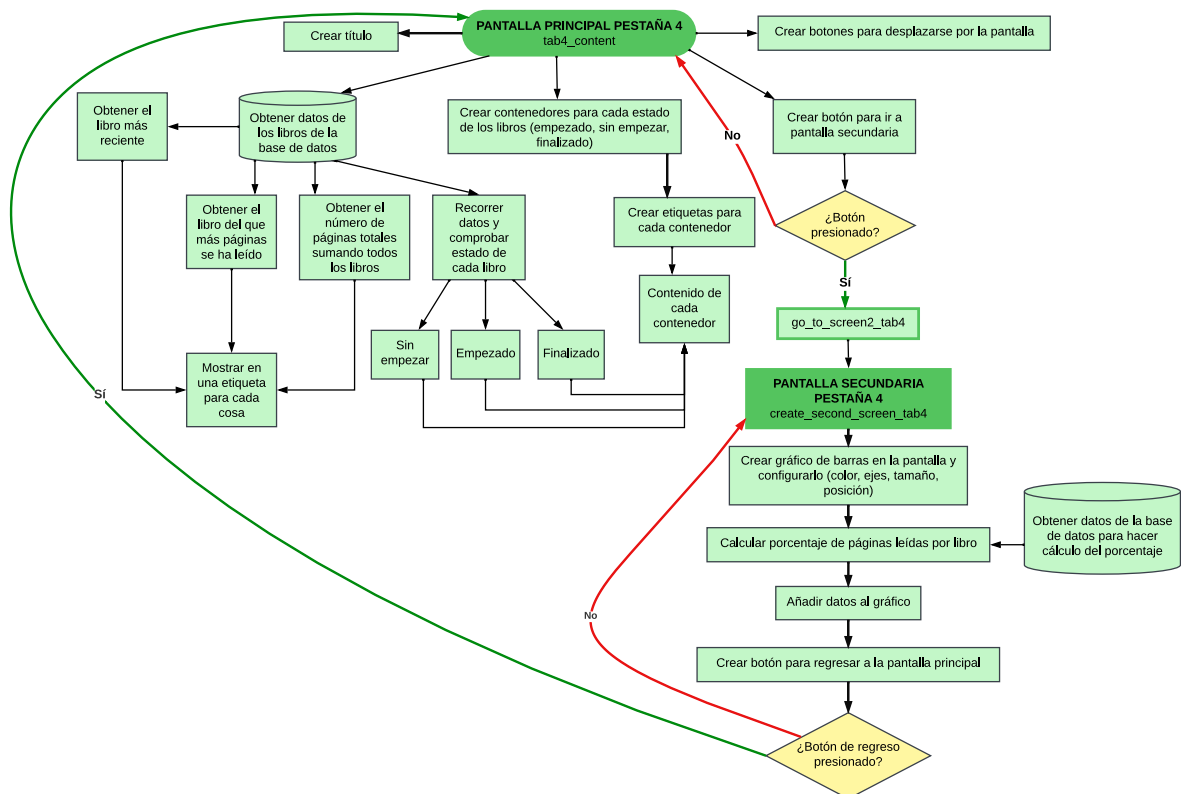


Figura 8.29. Esquema de funcionamiento de la pestaña 4

### 8.3. Base de datos

#### 8.3.1. Introducción a Firebase

Se ha seleccionado el almacenamiento en la nube para guardar la información de las lecturas de los usuarios, eligiendo específicamente *Firebase Realtime Database* para este propósito.

Esta base de datos NoSQL, hospedada en la nube de Google, es reconocida por su sencillez operativa. Proporciona una plataforma amigable que facilita a los usuarios la ejecución de operaciones CRUD (crear, leer, actualizar y eliminar) con los datos, incluso sin poseer una experiencia profunda en el manejo de bases de datos (45).

Esta plataforma ofrece sincronización en tiempo real, lo que significa que cualquier actualización en la información de un libro se refleja de inmediato en todos los dispositivos conectados (45).

En términos de seguridad y confiabilidad, proporciona una serie de reglas de seguridad y mecanismo de autenticación y control de acceso (45).

Por último, es una opción rentable, ya que ofrece un plan gratuito que cubre ampliamente las necesidades comunes, con 100 conexiones simultáneas, 1 GB de almacenamiento y 10 GB de transferencia de datos por mes, todo esto es más que suficiente para el actual TFG (46).

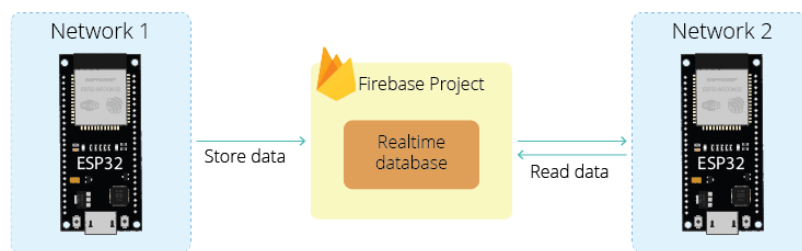


Figura 8.30. Esquema de comunicación con la base de datos (47)

#### 8.3.2. Formato de almacenamiento de los datos

Continuando con el almacenamiento de datos, estos se guardan en formato JSON (*JavaScript Object Notation*). Es un tipo de formato de almacenamiento de información para el intercambio de datos, ampliamente utilizado ya que es ligero y su sintaxis es entendible para los humanos (48).

Este formato de datos siempre comienza y termina con llaves y está formado por un conjunto de pares de claves y valores. Las claves siempre son cadenas de caracteres

y deben ir entre comillas dobles. Los valores pueden ser de varios tipos (objeto, *array*, cadena de caracteres, booleano, número o nulo) (48).

La clave se separa del valor mediante dos puntos, cada par de clave y valor se separa del siguiente con una coma, las llaves agrupan objetos y los corchetes se usan para agrupar *arrays* o vectores (48).

Un ejemplo sencillo de JSON podría ser:

```
{
  "Nombre": "Ainhoa",
  "Edad": 21
}
```

*Figura 8.31. Ejemplo de JSON*

El JSON de esta aplicación en particular es el siguiente y ha sido creado desde la interfaz de la web de Firebase:

```
{
  "libros": {
    "libro1": {
      "autor": "Eloy Moreno",
      "isbn": "9788416588435",
      "pagina_actual": 0,
      "paginas_total": 299,
      "titulo": "Invisible",
      "ultima_modificacion": 1718191606046
    },
    "libro2": {
      "autor": "Laura Gallego",
      "isbn": "9788467539677",
      "pagina_actual": 0,
      "paginas_total": 271,
      "titulo": "El valle de los lobos",
      "ultima_modificacion": 1717925378107
    },
    "libro3": {
      "autor": "Laura Gallego",
      "isbn": "9788467539684",
      "pagina_actual": 0,
      "paginas_total": 239,
      "titulo": "La maldición del maestro",
      "ultima_modificacion": 1718129698326
    },
    "libro4": {
      "autor": "Laura Gallego",
      "isbn": "9788467539691",
      "pagina_actual": 0,
      "paginas_total": 239,
      "titulo": "La llamada de los muertos",
      "ultima_modificacion": 1718182563901
    },
    "libro5": {
      "autor": "Laura Gallego",
      "isbn": "9788467539707",
      "pagina_actual": 0,
      "paginas_total": 299,
      "titulo": "Fenris, el elfo",
      "ultima_modificacion": 1718191654227
    }
  }
}
```

*Figura 8.32. JSON del proyecto completo*



Figura 8.33. Interfaz de Firebase donde se maneja el JSON

### 8.3.3. Creación de Realtime Database de Firebase

Para empezar a trabajar con *Realtime Database* de Firebase primero se debe crear una cuenta e ir a la consola de Firebase.

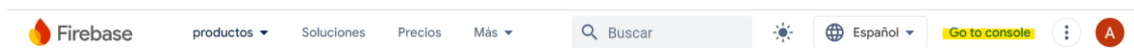


Figura 8.34. Cómo acceder a la consola de Firebase

Para después crear un nuevo proyecto asignándole un nombre descriptivo que indique la funcionalidad que se le va a dar y posteriormente se van siguiendo los pasos que indica la página.

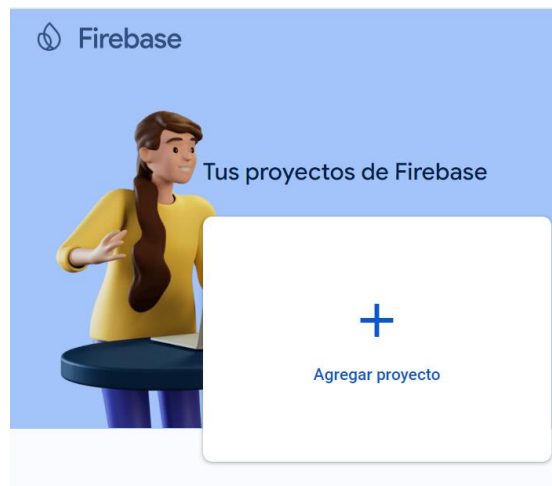


Figura 8.35. Creación de proyecto de Firebase

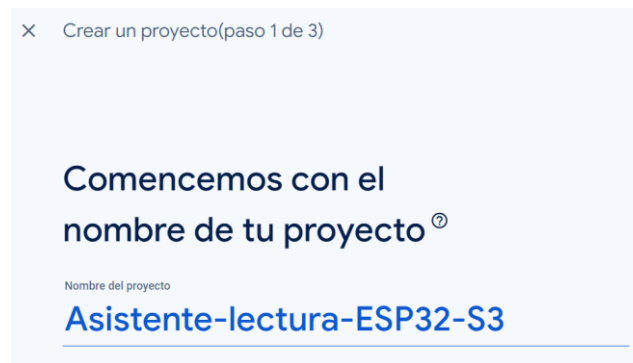


Figura 8.36. Asignación de nombre descriptivo

Durante el proceso de creación del proyecto se van generando los siguientes elementos esenciales para la comunicación entre el ESP32-S3 y la base de datos.

La URL es la dirección web donde se encuentra la *Realtime Database* de Firebase. Cada proyecto de Firebase tiene una URL de base de datos única. El ESP32-S3 necesita esta URL para saber a dónde enviar las solicitudes cuando quiera leer o escribir los datos.

La clave de la API es un identificador único que se utiliza para autenticar las solicitudes que ESP32-S3 hace a Firebase. Funciona de manera similar a una contraseña, proporcionando al ESP32-S3 la capacidad de demostrar su autorización para acceder a los servicios de Firebase. En otras palabras, la clave de la API es un mecanismo de seguridad que verifica la identidad del ESP32-S3 y garantiza que solo las entidades autorizadas puedan interactuar con la base de datos de Firebase.

Desde Firebase también se deben generar las credenciales de usuario, correo electrónico y contraseña, para autenticar al cliente y que solo accedan a la comunicación con la base de datos los usuarios autorizados.

Finalmente, se configuran las reglas de seguridad para determinar quién puede leer y escribir los datos. Las reglas de este proyecto se han diseñado para que solo realicen estas operaciones aquellos que estén autenticados.

```
1 {  
2   "rules": {  
3     ".read": "auth != null",  
4     ".write": "auth != null"  
5   }  
6 }
```

Figura 8.37. Reglas de seguridad para escritura y lectura de usuarios autenticados

### 8.3.4. Información a almacenar

#### 8.3.4.1. General

La información de la base de datos se almacena en el JSON visto anteriormente en la figura 8.32. En este caso, el objeto JSON principal contiene un objeto "libros", que a su vez contiene varios objetos que representan los libros individuales.

Cada uno de estos libros contiene un objeto con seis pares clave-valor con la siguiente información:

- **"autor"**: el nombre del autor del libro. Se almacena en forma de cadena de caracteres.
- **"isbn"**: el número ISBN del libro, identificador único para los libros. Se almacena en forma de cadena de caracteres.
- **"pagina\_actual"**: la página actual en la que se encuentra el lector. Este valor se actualiza a medida que el lector avanza en la lectura del libro. Se almacena en forma de número.
- **"paginas\_total"**: el número total de páginas del libro. Se almacena en forma de número.
- **"titulo"**: el título del libro. Se almacena en forma de cadena de caracteres.
- **"ultima\_modificacion"**: una marca de tiempo que indica cuándo se realizó la última modificación en los datos del libro. Se almacena en forma de número.

#### 8.3.4.2. Para la gamificación

En cuanto a la gamificación, se ha implementado un sistema de niveles basado en el número de páginas leídas en total. Para ello, se necesitan los datos correspondientes a la página actual de cada libro, de manera que se suman para calcular cuánto ha leído el usuario en cómputo global. A medida que lee más páginas, sube de nivel. Además, se le informa sobre cuántas páginas necesita leer para alcanzar el siguiente nivel. Los niveles son los siguientes:

- Nivel Aprendiz: 0-199 páginas leídas.
- Nivel Iniciado: 200-399 páginas leídas.
- Nivel Avanzado: 400-599 páginas leídas.
- Nivel Experto: 600-799 páginas leídas.

- Nivel Maestro: 800-1499 páginas leídas.
- Nivel Sabio: 1500 o más páginas leídas.

También, cuando el usuario termina de leer un libro, se le muestra un mensaje de felicitación y se le anima a seguir leyendo. Para ello, se necesitan los datos correspondientes a la página actual de cada libro y al número de páginas total de ese libro en concreto, para realizar la comparativa de ambas y averiguar si el lector ha llegado a la última página y por tanto ha finalizado el libro.

#### 8.3.4.3. Marca temporal

La última modificación es una marca temporal que se utiliza para ordenar los libros desde el más reciente hasta el más antiguo y para mostrarle al usuario cuál es su lectura actual. Se ha creado escribiendo para cada objeto: "ultima\_modificacion": {"sv": "timestamp"} (49). La etiqueta de tiempo es un valor del servidor, esto quiere decir que es un valor que se genera en el servidor y no en el cliente. Por tanto, al escribir {"sv": "timestamp"} Firebase lo reemplaza por el tiempo actual en milisegundos del servidor desde la época UNIX (1 de enero de 1970) y esto es el número largo que se aprecia en el JSON.

El usuario provoca una actualización del *timestamp* del libro correspondiente, desde el dispositivo cada vez que actualiza el número de página actual del libro desde la pantalla 3 de la pestaña 3. Para ello desde el código, de manera interna, se llama a la función *setTimestamp()*.

En resumen, la base de datos almacena información detallada sobre cada libro que el usuario está leyendo o ha leído. Esta información se utiliza para implementar un sistema de gamificación que motiva a leer más y para proporcionar estadísticas útiles.

#### 8.3.5. Configuración a nivel de código

Para poder trabajar con Firebase desde el ESP32-S3, primero se debe instalar su librería desde PlatformIO, que se denomina *Firebase Arduino Client Library for ESP8266 and ESP32*.

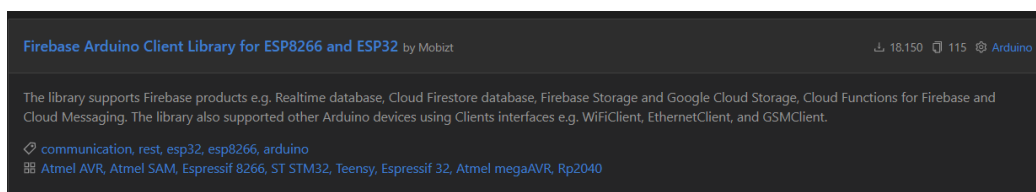


Figura 8.38. Instalación de la librería de Cliente Firebase para ESP32

Una vez que se tiene instalada la librería, se deben definir las constantes (variables cuyos valores no se ven alterados) de los siguientes elementos esenciales.

```
#define DATABASE_URL "https://asistente-lectura-esp32-s3-default-rtdb.europe-west1.firebaseio.com/"
#define API_KEY "AIzaSyDy16HV9yF2pW5C4llrmu9VVGicfL9JrtE"
#define USER_EMAIL "amm00384@red.ujaen.es"
#define USER_PASSWORD "frbs_4"
#define WIFI_SSID "telema2"
#define WIFI_PASSWORD "teleco2015"
```

Tras esto se declaran el resto de variables para interactuar con la base de datos, tales como el objeto *FirebaseData* para almacenar los datos, *FirebaseAuth* para la autenticación del usuario mediante el email y contraseña, *FirebaseConfig* para configuración general.

```
// Definición de las variables de Firebase
FirebaseData fbdo;
FirebaseAuth auth;
FirebaseConfig config;

// Definición de otras variables relacionadas con Firebase
unsigned long count = 0;
bool signupOK = false;
String path = "/libros";
```

Se inicia la configuración de Firebase en una función a la que luego se llamará desde la función de inicialización *setup()* del código fuente principal.

```
// Definición de la función para configurar Firebase
void setup_firebase() {

    // Inicializar la conexión Wi-Fi
    WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
    Serial.print("Conectando a la Red Wi-Fi "); // Imprime un mensaje de conexión a Wifi
    while (WiFi.status() != WL_CONNECTED) { // Bucle que espera a que la conexión sea exitosa
        Serial.print(".");
        delay(300);
    }
    Serial.println("Conectado con la IP: "); // Imprime un mensaje de conexión exitosa y la IP asignada
    Serial.println(WiFi.localIP());
    Serial.println();

    // Imprime la versión del cliente Firebase
    Serial.printf("Cliente Firebase v%s\n\n", FIREBASE_CLIENT_VERSION);

    // Asigna la clave API a la configuración
    config.api_key = API_KEY;

    // Asignar las credenciales de inicio de sesión del usuario
    auth.user.email = USER_EMAIL;
    auth.user.password = USER_PASSWORD;

    // Asigna la URL de la base de datos a la configuración
    config.database_url = DATABASE_URL;
```

```

// Registrarse
if (Firebase.signUp(&config, &auth, USER_EMAIL, USER_PASSWORD)) { // Intenta registrarse con
Firebase
    Serial.println("REGISTRO CORRECTO"); // Imprime mensaje de registro exitoso
    signupOK = true;
} else {
    Serial.printf("%s\n", config.signer.signupError.message.c_str()); // Imprime mensaje de error si falla
    el registro
}

//Asignar la función de devolución de llamada para la tarea de generación de token de larga duración
config.token_status_callback = tokenStatusCallback; //see addons/TokenHelper.h

Firebase.begin(&config, &auth); // Inicia la comunicación con Firebase
Firebase.reconnectWiFi(true); // Habilita la reconexión automática a Wifi
}

```

Finalmente se crean las funciones que permiten leer los datos personalizados del JSON de este proyecto y la función que permite modificar el número de página actual del usuario a la vez que se actualiza la marca temporal.

```

// Función para obtener los datos de un libro
DynamicJsonDocument get_book_data(const std::string & key) {
    // Define el path inicial del JSON en la base de datos Firebase
    String path = "/libros";
    // Si se proporciona una clave, se añade al path para apuntar a un libro específico
    if (!key.empty()) {
        path += "/" + String(key.c_str());
    }

    // Intenta obtener los datos del libro de la base de datos Firebase
    if (Firebase.RTDB.get(&fbdo, path.c_str())) {
        // Comprueba si los datos obtenidos son de tipo JSON
        if (fbdo.dataType() == "json") {
            // Obtiene un puntero al objeto JSON
            FirebaseJson *json = fbdo.jsonObjectPtr();
            // Crea una cadena de texto para almacenar el JSON
            String jsonString;
            // Convierte el objeto JSON a una cadena de texto
            json->toString(jsonString);

            // Crea un documento JSON para almacenar los datos
            DynamicJsonDocument doc(1024);
            // Deserializa la cadena de texto JSON en el documento JSON
            deserializeJson(doc, jsonString);

            // Devuelve el documento JSON con los datos del libro
            return doc;
        }
    }
    // Si ocurre algún error, devuelve un documento JSON vacío
    DynamicJsonDocument doc(1024);
    return doc;
}

```

```
// Función para actualizar la página actual de un libro
void update_current_page(const std::string & key, int page) {
    // Construye la ruta al campo "pagina_actual" del libro en la base de datos Firebase
    // La clave del libro se pasa como argumento a la función
    String path = "/libros/" + String(key.c_str()) + "/pagina_actual";
    // Actualiza el valor de "pagina_actual" en la base de datos con el número de página proporcionado
    Firebase.RTDB.setInt(&fbdo, path.c_str(), page);

    // Construye la ruta al campo "ultima_modificacion" del libro en la base de datos Firebase
    path = "/libros/" + String(key.c_str()) + "/ultima_modificacion";
    // Actualiza el valor de "ultima_modificacion" en la base de datos con la marca de tiempo actual
    Firebase.RTDB.setTimestamp(&fbdo, path.c_str());
}
}
```

## 8.4. Lectura y decodificación de QR

### 8.4.1. Creación del código

Para reconocer cada libro y poder acceder a su información en la base de datos, previamente se crea un QR para cada uno que codifica su ISBN (en texto). Ese número se encuentra junto al código de barras en el reverso del libro.

Debido al auge que ha experimentado este tipo de códigos, en internet existen multitud de páginas donde generarlo en cuestión de segundos. En concreto, para este caso se ha generado un QR versión 1 de dimensión 21 x 21 módulos, uno de los más sencillos, ya que no se necesita codificar mucha información, sino únicamente un texto con 13 dígitos.



Figura 8.39. Pasar de código de barras (ISBN) a QR que contiene el número del ISBN

### 8.4.2. Instalación de la librería de decodificación y funcionamiento

Una vez se ha creado, solo hay que mostrarlo a la cámara para que el ESP32-S3 lo procese y decodifique. Para realizar esto se ha instalado a través de PlatformIO, la librería ESP32QRCodeReader.

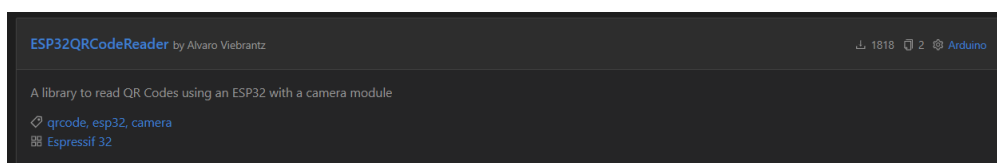


Figura 8.40. Librería para decodificar QR desde PlatformIO

Esta biblioteca posee una serie de limitaciones. Necesita que el dispositivo ESP32 que la utilice tenga PSRAM (en este proyecto la placa de desarrollo sí cuenta con PSRAM) y no está programada para el módulo de cámara empleado (CAMERA\_MODEL\_ESP32S3\_EYE).

Los módulos de cámara que el desarrollador especifica son:

- CAMERA\_MODEL\_WROVER\_KIT
- CAMERA\_MODEL\_ESP\_EYE
- CAMERA\_MODEL\_M5STACK\_PSRAM
- CAMERA\_MODEL\_M5STACK\_V2\_PSRAM
- CAMERA\_MODEL\_M5STACK\_WIDE
- CAMERA\_MODEL\_AI\_THINKER

Pero esto se puede solucionar de manera sencilla. En el archivo principal de la aplicación se declaran los pines del modelo de cámara que se está utilizando, previamente a declarar el archivo cabecera de la librería. De esta manera no se modifica la librería y se adapta a los requerimientos.

```
// El fabricante me indica que este es el modelo de cámara del kit
#define CAMERA_MODEL_ESP32S3_EYE \
{ \
    .PWDN_GPIO_NUM = -1, \
    .RESET_GPIO_NUM = -1, \
    .XCLK_GPIO_NUM = 15, \
    .SIOD_GPIO_NUM = 4, \
    .SIOC_GPIO_NUM = 5, \
    .Y9_GPIO_NUM = 16, \
    .Y8_GPIO_NUM = 17, \
    .Y7_GPIO_NUM = 18, \
    .Y6_GPIO_NUM = 12, \
    .Y5_GPIO_NUM = 10, \
    .Y4_GPIO_NUM = 8, \
    .Y3_GPIO_NUM = 9, \
    .Y2_GPIO_NUM = 11, \
    .VSYNC_GPIO_NUM = 6, \
    .HREF_GPIO_NUM = 7, \
    .PCLK_GPIO_NUM = 13, \
}

#include <ESP32QRCodeReader.h>
```

*Figura 8.41. Declaración del modelo de cámara empleado y sus pines*

El creador de la librería indica que internamente utiliza una versión ligeramente modificada de la biblioteca Quirc y algo de código de OpenMV que ha sido portado para funcionar en el proyecto MaixPy (50).

La biblioteca Quirc es una biblioteca de código abierto, escrita en C, diseñada específicamente para decodificar códigos QR. Es rápida, puede utilizarse en tiempo real y su algoritmo de reconocimiento permite detectar y decodificar códigos QR en diversas posiciones. En este proyecto, se ha modificado para que utilice directamente la PSRAM del ESP32, lo que ayuda a manejar la alta demanda de memoria requerida para el proceso de decodificación de códigos QR (51).

OpenMV es una plataforma de código abierto que utiliza Python para la visión artificial en microcontroladores, permitiendo el reconocimiento de imágenes a través de la cámara. Parte del código de OpenMV se adapta para funcionar con MaixPy (52).

MaixPy es un sistema operativo basado en MicroPython (implementación del lenguaje Python3 adaptado para funcionar con microcontroladores) que permite el uso de algoritmos de inteligencia artificial para procesar imágenes. En este caso, se utiliza para proporcionar funcionalidades adicionales para el manejo de imágenes y la interacción con la cámara (53).

En resumen, el reconocimiento de códigos QR en este proyecto se basa en la combinación de varias tecnologías y bibliotecas, la biblioteca Quirc modificada para utilizar la PSRAM, código de OpenMV y de MaixPy. Esta combinación de tecnologías permite que el proyecto maneje eficientemente la alta demanda de memoria requerida para el reconocimiento y decodificación de códigos QR.

Además, la biblioteca tiene sus propios archivos para configurar el modelo de la cámara, inicializar y declarar los pines de la cámara, configurar la frecuencia del reloj para capturar y procesar los datos, y establecer el formato de pixel al capturar la imagen, entre demás configuraciones.

## 9. CONCLUSIONES Y RESULTADOS

### 9.1. Resultados obtenidos

El proyecto ha desarrollado con éxito un prototipo funcional de un asistente de lectura que emplea gamificación utilizando ESP32-S3 que cumple con los objetivos planteados en el TFG. Los logros más significativos incluyen:

#### 9.1.1. Diseño del dispositivo

**Priorización de la facilidad de uso:** diseñado con un enfoque en la simplicidad y la intuición, permite a los usuarios interactuar con el dispositivo de manera eficiente y sin complicaciones. Los elementos visuales y las funciones se presentan de manera clara, lo que facilita la comprensión y el uso de las características del dispositivo. Además, el tamaño compacto del dispositivo contribuye a su facilidad de uso.

**Estudio exhaustivo de alternativas:** se llevó a cabo un análisis exhaustivo de diferentes opciones de dispositivos en diversos sitios de compra online. Se evaluaron aspectos como el precio y las características técnicas.

**Selección del dispositivo:** tras un análisis meticuloso, se optó por el kit de Amazon. A pesar de no ser la opción más económica, este dispositivo cumplía con los requisitos técnicos esenciales y, además, ofrecía una ventaja significativa: un amplio tutorial de ayuda a la programación de ese dispositivo en concreto, proporcionado por el fabricante.

#### 9.1.2. Reconocimiento del ejemplar de libro

**Implementación del sistema de códigos QR:** se ha desarrollado un sistema de reconocimiento de códigos QR para identificar los libros de forma sencilla y eficaz. Los códigos QR se pueden imprimir en las portadas o contraportadas de los libros, o incluso en etiquetas adhesivas. El sistema es capaz de leer los códigos QR incluso en condiciones de iluminación difíciles, gracias a la iluminación del LED de la placa de desarrollo.

#### 9.1.3. Métricas de fomento de la lectura (gamificación e interfaz)

Se han implementado diversas métricas para fomentar la lectura de los usuarios, como:

**Gamificación con niveles de lectura:** el usuario progresa a través de diferentes niveles a medida que lee más páginas, y por consiguiente más libros. Esta gamificación motiva a los usuarios a seguir leyendo y alcanzar nuevos niveles. También se muestra el número de páginas restantes para alcanzar el próximo nivel.

**Número de total páginas leídas:** se muestra el número total de páginas leídas por el usuario, lo que le permite visualizar su progreso de forma cuantitativa.

**Libro más largo leído:** se muestra cuál es el libro del que más páginas ha leído.

**Lectura actual:** se proporciona al usuario el título del libro más reciente que está leyendo, además de que los libros de su biblioteca se ordenan desde el más reciente al más antiguo.

**Porcentaje de libros empezados:** se muestra el porcentaje de cada libro que tiene comenzado el usuario, de manera que tenga una idea clara del avance sobre cada uno de esos libros.

**Gráficos de progreso:** se utiliza un gráfico de barra para visualizar el progreso del usuario de forma gráfica e intuitiva (porcentaje de páginas leídas para todos los libros de la biblioteca personal del lector). Esto facilita al usuario comprender su evolución y mantenerse motivado.

**Retroalimentación personalizada:** se proporciona a los usuarios una retroalimentación personalizada sobre su progreso. Se muestran mensajes de felicitación cuando se completa cada libro, animándolo a que siga leyendo. La retroalimentación positiva y motivadora ayuda a los usuarios a mantenerse comprometidos y a alcanzar sus objetivos de lectura.

**Presentación atractiva y fácil de entender:** Las métricas se muestran en una interfaz gráfica atractiva y fácil de entender, utilizando colores vivos, iconos y animaciones. Esto hace que la información sea más accesible y atractiva para los usuarios, especialmente para niños y jóvenes.

#### *9.1.4. Almacenamiento de la información*

**Desarrollo de una base de datos en la nube:** se ha creado una base de datos en tiempo real de Firebase en la nube para almacenar de forma segura la información de los usuarios y los libros. La base de datos está protegida por protocolos de seguridad y autenticación para garantizar la privacidad de los datos.

**Acceso seguro y eficiente:** El dispositivo puede acceder a la base de datos en la nube de forma segura y eficiente, utilizando una conexión a Internet. Esto permite que el dispositivo se mantenga actualizado con la información más reciente y que los usuarios puedan acceder a sus datos desde cualquier lugar, siempre y cuando tenga conexión a internet.

#### *9.1.5. Gamificación y realimentación*

## **9.2. Limitaciones**

Durante el desarrollo del proyecto, se encontraron varios desafíos y limitaciones.

### *9.2.1. Lectura de QR y no de código de barras (ISBN)*

Uno de los principales desafíos fue la necesidad de crear un código QR para cada libro, ya que la cámara no tenía suficiente calidad de enfoque para leer el ISBN del libro. Sin embargo, este desafío se superó con éxito mediante la implementación de la decodificación de códigos QR, que demostró ser una solución robusta y eficiente.

### *9.2.2. Mostrar imagen de cámara en pantalla y decodificar simultáneamente*

Otra limitación encontrada fue la incapacidad de mostrar la imagen de la cámara en la pantalla al mismo tiempo que se captura y decodifica el QR debido a conflictos entre las tareas de FreeRTOS. FreeRTOS es un sistema operativo en tiempo real para microcontroladores que maneja tareas para manejar distintos hilos de funciones en la aplicación. El framework de Arduino utilizado incluye FreeRTOS.

Las operaciones con la cámara se realizan con tareas FreeRTOS debido a que la captura de imágenes y decodificación del código son procesos costosos que consumen tiempo, por lo que al realizarlo con este tipo de tareas se permite que se ejecuten a la vez otras tareas, como la actualización de la GUI o la comunicación con la base de datos, y así se gestionan mejor los recursos del sistema.

El problema de no poder mostrar la imagen en la pantalla desde la cámara a la vez que se captura y decodifica el QR es porque ambas tareas necesitan acceder al mismo recurso compartido, el búfer de la cámara y eso provoca conflicto entre ambos hilos, lo cual provoca el reinicio en bucle del dispositivo. No se ha logrado encontrar una solución a esto. Simplemente se ha optado por desechar la tarea de mostrar la imagen de la cámara en la pantalla.

### *9.2.3. Actualización de pantalla al cambiar datos desde la base de datos*

Se encontró también un desafío en la actualización de las pantallas principales de las pestañas de la interfaz gráfica de usuario cuando se cambia algún dato en la base de datos. Este desafío se superó con éxito, permitiendo que las pantallas se actualicen sin necesidad de reiniciar el dispositivo (antes de solucionarlo se tenía que pulsar el botón de *reset*).

Si el usuario se desplaza a otra pestaña o pantalla, al volver a la que necesita actualizar su contenido, comprobará que se ha actualizado automáticamente.

### *9.2.4. Gestión de recursos en un sistema embebido*

El desarrollo de la actual aplicación de asistente a la lectura en un microcontrolador ESP32-S3 presenta una serie de desafíos y limitaciones inherentes a la naturaleza de los sistemas embebidos. Estos dispositivos, aunque potentes para su tamaño, tienen recursos limitados en comparación con los sistemas informáticos tradicionales. Esto implica que la gestión eficiente de estos recursos, como la memoria y la capacidad de procesamiento, es crucial para el rendimiento y la funcionalidad de la aplicación.

### *9.2.5. Integración y sincronización de componentes de hardware*

La aplicación requiere la integración y sincronización de varios componentes de hardware, incluyendo una cámara y una pantalla. Esto implica la gestión de la comunicación entre estos dispositivos y el ESP32-S3, así como la programación de la lógica para que funcionen de manera conjunta y fluida.

### *9.2.6. Creación de la interfaz gráfica de usuario (GUI)*

La creación de la interfaz gráfica de usuario (GUI) también presentó desafíos. Se utilizó la biblioteca LVGL, que está diseñada para sistemas embebidos con recursos limitados. Sin embargo, crear una GUI atractiva y funcional con esta biblioteca requiere un conocimiento profundo de su funcionamiento y capacidades.

### *9.2.7. Conectividad WiFi y comunicación con la base de datos en la nube*

La aplicación incluye conectividad WiFi para comunicarse con una base de datos en la nube. Esto permite a la aplicación leer y escribir datos del lector, pero también añade complejidad al proyecto. La programación de la lógica de red, la gestión de la conexión WiFi y la sincronización de los datos entre el dispositivo y la nube son tareas que consumen recursos en el microcontrolador.

A pesar de estos desafíos y limitaciones, el proyecto ha logrado cumplir con los objetivos planteados inicialmente. Sin embargo, se reconoce que aún existen áreas de mejora y se espera que futuros trabajos puedan abordar estas limitaciones para mejorar aún más la eficacia y la experiencia del usuario del asistente de lectura.

## 10. LÍNEAS FUTURAS

En el desarrollo del presente proyecto se han identificado diversas áreas de mejora y expansión que podrían ser abordadas en investigaciones futuras. A continuación, se presentan las líneas propuestas para el enriquecimiento y la evolución del asistente de lectura basado utilizando ESP32-S3.

### 10.1. Mejora en la captura y procesamiento de imágenes

Una de las limitaciones actuales radica en la capacidad de la cámara para enfocar y leer correctamente el ISBN de los libros. Para superar este obstáculo, se propone la incorporación de una lente con un sensor de mayor calidad que permita un enfoque más preciso.

Los principales sensores que soporta ESP32-S3 son:

| model  | max resolution | color type | output format                                                                         | Len Size |
|--------|----------------|------------|---------------------------------------------------------------------------------------|----------|
| OV2640 | 1600 x 1200    | color      | YUV(422/420)/YCbCr422<br>RGB565/555<br>8-bit compressed data<br>8/10-bit Raw RGB data | 1/4"     |
| OV3660 | 2048 x 1536    | color      | raw RGB data<br>RGB565/555/444<br>CCIR656<br>YCbCr422<br>compression                  | 1/5"     |
| OV5640 | 2592 x 1944    | color      | RAW RGB<br>RGB565/555/444<br>CCIR656<br>YUV422/420<br>YCbCr422<br>compression         | 1/4"     |

Tabla 10.1. Sensores de cámara compatibles con ESP32-S3 (54)

En este proyecto se ha utilizado OV2640, sin lograr un buen enfoque a distancias cortas. Ahora se propone usar el sensor OV5640 en lugar del OV2640 o el OV3660, ya que al tener una resolución más alta permitirá capturar imágenes con mayor detalle, incluso de cerca. Además, cuenta con una función de autoenfoco, proporcionando así un enfoque mucho más preciso.

Sin embargo, dado que no existe una librería específica para ESP32 que cumpla con la función de decodificar códigos de barras, se sugiere la implementación de un

servidor externo que, utilizando Python y OpenCV, realice la decodificación del ISBN y envíe la información resultante al ESP32.

Se propone el uso de Python y OpenCV porque ya existe documentación que sirve de guía para usar estas herramientas y decodificar el código de barras. OpenCV es una librería de código abierto con la que se permite realizar procesamiento de imágenes y es compatible con Python, de esta manera se podría lograr el objetivo de leer el ISBN del libro.

## **10.2. Nuevas funcionalidades para el usuario**

Se contempla la posibilidad de añadir una pestaña adicional dedicada a retos, con el fin de incrementar la interactividad y el compromiso del usuario con la lectura. Esta pestaña podría incluir desafíos y metas que motiven al usuario a incrementar su frecuencia y volumen de lectura.

Una opción sería que el lector se pueda imponer su propio límite de reto de lectura por día. Para hacer esto se debería añadir un nuevo objeto en el JSON de la base de datos, cuya clave sea “reto” y que contenga el mínimo de páginas diarias que se proponer leer por día por ejemplo (ese número lo podría introducir el propio usuario desde un teclado numérico en la nueva pestaña), una marca de tiempo que se reinicie cada 24 horas para poder reiniciar otra variable nueva que contenga el número de páginas leídas por día.

También, para mejorar la interacción con la base de datos se plantea que el usuario tenga también la posibilidad de introducir libros nuevos en la base de datos en tiempo real de Firebase, desde el ESP32-S3. Así su interacción con sus datos de la nube no se limitaría únicamente a leerlos y a actualizar el número de página actual. Sin embargo, esto presenta el desafío del tamaño reducido de la pantalla para mostrar un teclado completo para introducir los datos como el título del libro. Por lo tanto, se requeriría una pantalla de mayor resolución y tamaño para facilitar esta interacción.

## **10.3. Reconocimiento avanzado de libros**

Otra línea de mejora consiste en entrenar al dispositivo para que no solo reconozca el código de barras o el QR asociado a un libro, sino que también sea capaz de identificar el libro por su portada. Esto proporcionaría una experiencia de usuario más fluida y natural, alineada con las interacciones cotidianas con libros físicos.

Aunque, es necesario destacar que esto podría ser un desafío complicado que podría requerir uso de redes neuronales o aprendizaje automático para realizar un algoritmo de inteligencia artificial y que se pueda llevar a cabo el entrenamiento del dispositivo para reconocer correctamente las portadas.

Primero se debería recopilar datos de imágenes de las portadas de libros que se quieran reconocer. Las imágenes deben sean claras y estar tomadas desde diferentes ángulos y condiciones de iluminación.

Después habría que procesar las imágenes para que todas tengan el mismo tamaño y características similares.

Entonces es cuando se debería utilizar un algoritmo de aprendizaje automático para entrenar un modelo con las imágenes recopiladas.

Una vez que se tiene el modelo entrenado, se cargaría en el ESP32-S3 para realizar finalmente la predicción de la portada del libro y reconocerlo.

#### **10.4. Interacción por voz con el usuario**

Una propuesta de mejora adicional consiste en la integración de un sistema de Síntesis de Voz (*Text-to-Speech*) que permita al dispositivo interactuar verbalmente a través del altavoz con el usuario. Esta funcionalidad podría emplearse para comunicar al usuario información relevante como el libro identificado, el número de páginas leídas, así como para motivar al usuario a proseguir con su lectura, entre otras posibilidades.

La implementación de TTS enriquecería la experiencia del usuario y aumentaría la accesibilidad del dispositivo, ya que permitiría la interacción con el mismo sin la necesidad de visualizar la pantalla. Esta característica podría resultar especialmente beneficiosa para usuarios con discapacidades visuales.

Para la implementación de esta funcionalidad, se podría recurrir a una biblioteca de TTS compatible con ESP32-S3. Dicha biblioteca sería responsable de convertir el texto en voz y reproducirla a través del altavoz conectado al ESP32-S3.

Es importante destacar que la implementación de esta funcionalidad requeriría tener en cuenta diversos aspectos, como la calidad de la voz generada y la velocidad de habla, entre otros, con el objetivo de garantizar una experiencia de usuario satisfactoria.

## 11. BIBLIOGRAFÍA

1. **Ministerio de Educación, Formación Profesional y Deportes.** *PISA 2022. Programa para la Evaluación Internacional de los Estudiantes. Informe español.* s.l.: Instituto Nacional de Evaluación Educativa, 2023.
2. **España, Ministerio de Cultura.** Los índices de lectura de los españoles se mantienen estables tras la pandemia y crecen cinco puntos respecto a 2012. [En línea] 31 de 01 de 2024. [Citado el: 10 de 04 de 2024.] <https://www.cultura.gob.es/actualidad/2024/01/240131-barometro-habitos-lectura.html>.
3. **Carmenate, José Guerra.** ESP32 Wifi y Bluetooth en un solo chip. [En línea] [Citado el: 11 de 04 de 2024.] <https://programarfacil.com/esp8266/esp32/>.
4. **Tecnotec.** ESP32 VS ARDUINO UNO: ¿CUÁL DEBERÍA OBTENER? [En línea] [Citado el: 10 de 06 de 2024.] <https://tecnotec.es/esp32-vs-arduino-uno-cual-deberia-obtener/>.
5. **MCI, electronics.** ¿Que es Raspberry Pi? [En línea] [Citado el: 20 de 05 de 2024.] <https://raspberrypi.cl/que-es-raspberry/>.
6. **Elsinga, Roelof Jan.** What is the difference between an Arduino/ESP and a Raspberry Pi? [En línea] 16 de 12 de 2021. [Citado el: 10 de 06 de 2024.] <https://www.espforbeginners.com/guides/differences-between-arduino-raspberry-pi/>.
7. **Mecna, Programación básica.** Lenguaje de programación en Arduino: Guía completa. [En línea] 07 de 11 de 2023. [Citado el: 20 de 05 de 2024.] <https://mecna.es/lenguaje-de-programacion-en-arduino-guia-completa/>.
8. **Wikipedia.** Arduino Uno. [En línea] 11 de 11 de 2023. [Citado el: 20 de 05 de 2024.] [https://es.wikipedia.org/wiki/Arduino\\_Uno](https://es.wikipedia.org/wiki/Arduino_Uno).
9. **Aufranc, Jean-Luc.** Know the Differences between Raspberry Pi, Arduino, and ESP8266/ESP32. [En línea] 24 de 03 de 2020. [Citado el: 20 de 05 de 2024.] <https://www.cnx-software.com/2020/03/24/know-the-differences-between-raspberry-pi-arduino-and-esp8266-esp32/>.
10. **Minibots.** Pruebas de rendimiento de Raspberry Pi Pico y comparativa con plataformas Arduino, ESP8266 y ESP32. [En línea] 06 de 04 de 2021. [Citado el: 20 de 05 de 2024.] <https://minibots.wordpress.com/2021/04/06/pruebas-de-rendimiento-de-raspberry-pi-pico-y-comparativa-con-plataformas-arduino-esp8266-y-esp32/>.
11. **Systems, Espressif.** About Espressif. [En línea] [Citado el: 10 de 04 de 2024.] <https://www.espressif.com/en/company/about-espressif>.
12. **Herranz, Álvaro Benito.** Trabajo Fin de Grado . *Desarrollo de aplicaciones para IoT con el módulo ESP32.* Alcalá de Henares : s.n., 2019.
13. **Electronics, Mouser.** Espressif Systems ESP8266EX. [En línea] [Citado el: 28 de 04 de 2024.] [https://www.mouser.es/ProductDetail/Espressif-Systems/ESP8266EX?qs=chTDxNqvsynorwcu2ADV3g%3D%3D&mgh=1&vip=1&utm\\_id=19105062887&gad\\_source=1&gclid=CjwKCAjw57exBhAsEiwAalxaZmN-HCQddD\\_dwYVDw5zRCYMeWjoQjLs1jEwKK0UXPU-YfJ4bQAY29xoCoBMQAvD\\_BwE](https://www.mouser.es/ProductDetail/Espressif-Systems/ESP8266EX?qs=chTDxNqvsynorwcu2ADV3g%3D%3D&mgh=1&vip=1&utm_id=19105062887&gad_source=1&gclid=CjwKCAjw57exBhAsEiwAalxaZmN-HCQddD_dwYVDw5zRCYMeWjoQjLs1jEwKK0UXPU-YfJ4bQAY29xoCoBMQAvD_BwE).

14. **NAFCOM.** Módulos Bluetooth/WiFi/RF/LAN. [En línea] [Citado el: 28 de 04 de 2024.] <https://nafcom.es/modulos-bluetooth-wifi-rf/esp12s-esp-12s-esp8266-modulo-wifi/gmx-niv509-con10158.htm>.
15. **IndiaMART.** Wifi Model Vendor ESP8266MOD, For Automation, Model Name/Number: Esp8266 Development Board. [En línea] [Citado el: 28 de 04 de 2024.] <https://www.indiamart.com/proddetail/model-vendor-esp8266mod-2849145496630.html>.
16. **Systems, Espressif.** ESP32Series Datasheet.
17. —. ESP-IDF Programming Guide. [En línea] [Citado el: 13 de 04 de 2024.] <https://docs.espressif.com/projects/esp-idf/en/v4.3/esp32s2/api-guides/memory-types.html>.
18. **Moonrinta, Sirawit.** Comparison table for ESP8266/ESP32/ESP32-S2/ESP32-S3/ESP32-C3/ESP32-C6 (Github). [En línea] 2021. [Citado el: 12 de 04 de 2024.] <https://gist.github.com/sekcompSci/2bf39e715d5fe47579fa184fa819f421>.
19. **Systems, Espressif.** ESP32-C3 Series Datasheet.
20. —. ESP32-C6 Series Datasheet.
21. —. ESP32-S2 Series Datasheet.
22. —. ESP32-S3 Series Datasheet.
23. —. ESP32-S3-WROOM-1 Datasheet.
24. **Freenove.** Tutorial kit ESP32-S3 Freenove.
25. **Technologies, OmniVision.** OV2640 2 MPixel product brief.
26. **Moragomez, Erick Roch.** Todo sobre ESP32: Guía y Aplicaciones Prácticas. [En línea] [Citado el: 18 de 06 de 2024.] <https://lovtechnology.com/todo-sobre-esp32-guia-y-aplicaciones-practicas/>.
27. **TRSPoS.** ¿Qué lenguaje de programación utiliza ESP32? [En línea] [Citado el: 18 de 06 de 2024.] [https://trspos.com/lenguajes-de-programacion-esp32-uso/#arduino\\_language](https://trspos.com/lenguajes-de-programacion-esp32-uso/#arduino_language).
28. **FreeRTOS.** What is An RTOS? [En línea] [Citado el: 18 de 06 de 2024.] <https://www.freertos.org/about-RTOS.html>.
29. **Espressif.** Arduino as an ESP-IDF component. [En línea] [Citado el: 18 de 06 de 2024.] [https://docs.espressif.com/projects/arduino-esp32/en/latest/esp-idf\\_component.html](https://docs.espressif.com/projects/arduino-esp32/en/latest/esp-idf_component.html).
30. **Workshop, DroneBot.** Getting Started with PlatformIO. [En línea] 16 de 01 de 2021. [Citado el: 11 de 06 de 2024.] <https://dronebotworkshop.com/platformio/>.
31. **Espressif.** ESP32-S3-DevKitC-1. [En línea] [Citado el: 15 de 06 de 2024.] <https://docs.espressif.com/projects/esp-idf/en/stable/esp32s3/hw-reference/esp32s3/user-guide-devkitc-1-v1.0.html>.
32. —. ESP32-S3-DevKitM-1. [En línea] [Citado el: 15 de 06 de 2024.] <https://docs.espressif.com/projects/esp-idf/en/stable/esp32s3/hw-reference/esp32s3/user-guide-devkitm-1.html>.
33. **Omnivision.** OV2640 Camera Module Software Application Notes. 2007.

34. **Cataluña, Universidad Politécnica.** ¿Qué son los códigos QR? [En línea] [Citado el: 11 de 06 de 2024.] <https://inlab.fib.upc.edu/es/uncategorized-ca-es/que-son-los-codigos-qr/2012/>.
35. **SOLGE.** Estas son las diferencias entre los códigos de barras 1D y 2D. [En línea] 18 de 10 de 2022. [Citado el: 28 de 04 de 2024.] <https://www.solge.es/sin-categorizar/estas-son-las-diferencias-entre-los-codigos-de-barras-1d-y-2d/>.
36. **Nebot, Marc.** Tipos de códigos de barras: Una guía completa para elegir el adecuado. [En línea] [Citado el: 28 de 04 de 2024.] <https://stockagile.com/blog/tipos-de-codigos-de-barras/>.
37. **Keyence.** What is the QR code? [En línea] [Citado el: 11 de 06 de 2024.] [https://www.keyence.com.mx/ss/products/auto\\_id/barcode\\_lecture/basic\\_2d/qr/](https://www.keyence.com.mx/ss/products/auto_id/barcode_lecture/basic_2d/qr/).
38. **Brampton.** Barcode + QR Code Reader using Arduino & QR Scanner Module. [En línea] 29 de 05 de 2023. [Citado el: 01 de 05 de 2024.] <https://how2electronics.com/barcode-qr-code-reader-using-arduino-qr-scanner-module/>.
39. **Luong, Irene.** What is a barcode and how does it work? [En línea] 19 de 12 de 2023. [Citado el: 30 de 04 de 2024.] <https://www.magestore.com/blog/what-is-a-barcode/#s6>.
40. **QR, CODIGO.** Calcular dígito de control EAN-13. [En línea] [Citado el: 01 de 05 de 2024.] <https://www.codigos-qr.com/calcular-digito-de-control-ean13-gtin13/#:~:text=%C2%BFC%C3%B3mo%20distinguir%20el%20d%C3%ADgito%20de,acompa%C3%B1ar%20al%20c%C3%B3digo%20de%20barras..>
41. **Ortega, Adriana.** Descifrando la anatomía de un ISBN. [En línea] 14 de 06 de 2022. [Citado el: 01 de 05 de 2024.] <https://metabooks.com/es/servicio/blog-mvb-america-latina/descifrando-la-anatomia-de-un-isbn>.
42. **Xu, Lihang.** Locating and Decoding EAN-13 Barcodes using Python and OpenCV. [En línea] 26 de 08 de 2021. [Citado el: 20 de 05 de 2024.] <https://www.dynamsoft.com/codepool/locating-and-decoding-ean13-python-opencv.html>.
43. **Gaitán, Virginia.** Gamificación: el aprendizaje divertido. [En línea] [Citado el: 13 de 06 de 2024.] <https://www.educativa.com/blog-articulos/gamificacion-el-aprendizaje-divertido/>.
44. **Harry.** What is a programming flowchart and how is it made? [En línea] 07 de 12 de 2023. [Citado el: 13 de 06 de 2024.] <https://boardmix.com/es/knowledge/programming-flowchart/>.
45. **Firestore.** Realtime Database Firestore. [En línea] 20 de 03 de 2024. [Citado el: 13 de 06 de 2024.] <https://firebase.google.com/docs/database?hl=es>.
46. —. **Firestore Pricing.** [En línea] [Citado el: 13 de 06 de 2024.] <https://firebase.google.com/pricing?hl=es-419>.
47. **Tutorials, Random Nerd.** ESP32: Getting Started with Firestore (Realtime Database). [En línea] [Citado el: 11 de 06 de 2024.] <https://randomnerdtutorials.com/esp32-firebase-realtime-database/>.
48. **A., Deyimar.** ¿Qué es JSON? [En línea] 10 de 01 de 2023. [Citado el: 13 de 06 de 2024.] <https://www.hostinger.es/tutoriales/que-es-json>.
49. **Firestore.** Save data Firestore. [En línea] [Citado el: 13 de 06 de 2024.] <https://firebase.google.com/docs/database/rest/save-data?hl=es-419>.

50. **Viebrantz, Álvaro.** A library to read QR Codes using an ESP32 with a camera module (Github). [En línea] 2023. [Citado el: 11 de 06 de 2024.] <https://github.com/alvarowolff/ESP32QRCodeReader?tab=readme-ov-file>.
51. **Beer, Daniel.** QR decoder library (Github). [En línea] 2023. [Citado el: 11 de 06 de 2024.] <https://github.com/dlbeer/quirc>.
52. **OpenMV.** Machine Vision with Python. [En línea] [Citado el: 11 de 06 de 2024.] <https://openmv.io/>.
53. **Sipeed.** Introduction of MaixPy-v1. [En línea] [Citado el: 11 de 06 de 2024.] <https://wiki.sipeed.com/soft/maixpy/en/>.
54. **Espressif.** ESP32 Camera Driver. [En línea] [Citado el: 16 de 06 de 2024.] <https://github.com/espressif/esp32-camera>.
55. **Ernstzen, Lee.** Types Of Barcode Scanners – A Detailed Guide. [En línea] 07 de 03 de 2023. [Citado el: 30 de 04 de 2024.] <https://tritonstore.com.au/types-of-barcode-scanners/>.
56. **K., Suwatchai.** Async Firebase Client for Arduino. Supports Realtime Database, Cloud Firestore Database, Firebase Storage, Cloud Messaging, Google Cloud Functions and Google Cloud Storage (Github). [En línea] [Citado el: 11 de 06 de 2024.] <https://github.com/mobizt/FirebaseClient>.
57. **ESPBoards.** ESP-IDF vs Arduino Core: Which Framework to Choose in 2023. [En línea] 18 de 03 de 2023. [Citado el: 18 de 06 de 2024.] <https://www.espboards.dev/blog/esp-idf-vs-arduino-core/>.

## 12. ANEXOS

### 12.1. Manual de usuario

#### 12.1.1. Introducción

El presente manual está destinado a proporcionar una guía detallada para el uso del Asistente de Lectura con ESP32-S3. Siguiendo las instrucciones aquí descritas, se podrá navegar fácilmente por las distintas secciones del dispositivo y aprovechar todas sus funcionalidades, mejorando así la experiencia de lectura.

Este dispositivo ha sido diseñado para mejorar la experiencia de lectura mediante una interfaz gráfica intuitiva y elementos de gamificación.

Primero, es imprescindible que el dispositivo esté adecuadamente alimentado. Esto se puede lograr conectándolo a un ordenador mediante un cable USB o a una fuente de alimentación externa, como una pila de 9 V, tal y como se muestra en la imagen.

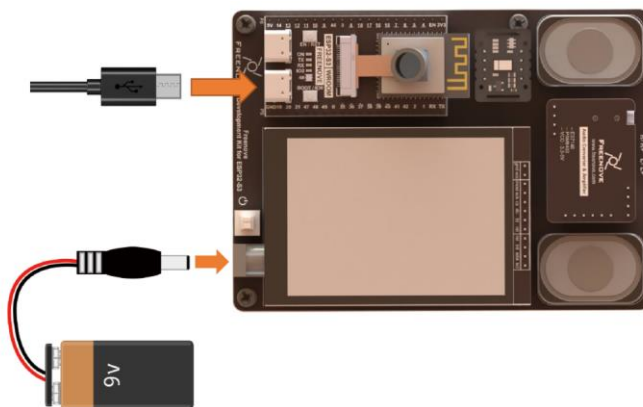


Figura 12.1. Alimentación del dispositivo (24)

A continuación, se detallan las diversas funciones y características del dispositivo.

#### 12.1.2. Interfaz de Usuario

La interfaz se divide en cuatro secciones, cada una identificada por un color distinto para facilitar la memoria visual y recordar mejor el funcionamiento de cada sección:

- Pestaña 1 (Morado): Inicio y sistema de niveles de lector para motivar
- Pestaña 2 (Azul): Biblioteca

- Pestaña 3 (Naranja): Escaneo de QR del libro y modificación de página actual del libro
- Pestaña 4 (Verde): Progreso del lector

#### 12.1.2.1. Pestaña 1: Inicio y Gamificación

### Pantalla Principal

Bienvenida al usuario → Muestra el nivel de lectura actual.

Botón → Permite acceder a una pantalla secundaria con información adicional sobre los niveles y funcionalidades del dispositivo.

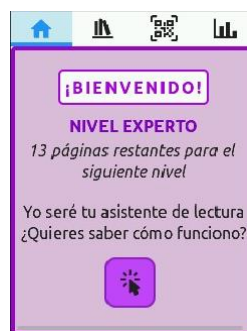


Figura 12.2. Pantalla principal de la pestaña 1

### Pantalla Secundaria (informativa)

Información de niveles → Describe los diferentes niveles que se pueden alcanzar.

Información de aplicación → Explica las distintas secciones de la aplicación.

Botón de Regreso → Permite volver a la pantalla principal.



Figura 12.3. Pantalla secundaria explicativa de la pestaña 1



Figura 12.4. Pantalla secundaria explicativa de la pestaña 1

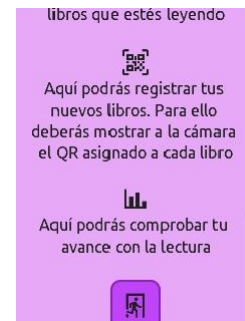


Figura 12.5. Pantalla secundaria explicativa de la pestaña 1

### 12.1.2.2. Pestaña 2: Biblioteca

#### Pantalla Principal

Lista de libros → Presenta todos los libros registrados del usuario, ordenados del más reciente al más antiguo.

Botón de libro → Al pulsar sobre un título, se accede a más detalles del libro en una pantalla nueva.



Figura 12.6. Pantalla principal de la pestaña 2

#### Pantalla Secundaria

Detalles del libro → Muestra el título, autor, número total de páginas y la página actual del libro seleccionado.

Botón de regreso → Permite volver a la lista de libros.



Figura 12.7. Pantalla secundaria de la pestaña 2

### 12.1.2.3. Pestaña 3: Escaneo de QR y modificación de página actual del libro

#### Pantalla Principal

Instrucciones → Proporciona una guía sobre cómo utilizar la función de escaneo de QR.

Botón de escaneo → Activa la cámara para escanear el código QR del libro y lleva al usuario a una nueva pantalla.



Figura 12.8. Pantalla principal de la pestaña 3

### Pantalla de Escaneo

Escaneo de QR → La cámara se activa para escanear el código QR del libro. El LED se enciende para mejorar la visibilidad.

Botón de regreso al deslizar la pantalla hacia abajo → Permite volver a la pantalla principal de la pestaña 3.



Figura 12.9. Pantalla secundaria de la pestaña 3 (escaneo de QR)

### Pantalla de resultado tras escaneo

Libro Encontrado → Muestra los detalles del libro escaneado y un teclado para introducir la página actual por la que el usuario se ha quedado leyendo.

Libro No Encontrado → Presenta un mensaje de error si el usuario no lo tiene registrado en su biblioteca.

Botón de regreso → Permite volver a la pantalla principal de la pestaña 3.

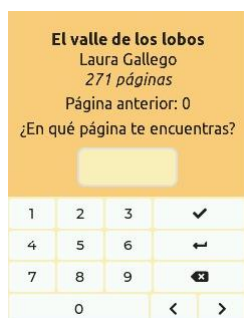


Figura 12.10. Tercera pantalla de la pestaña 3

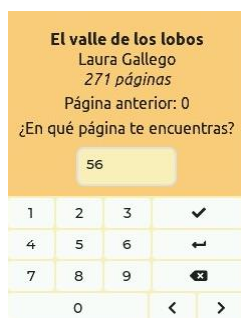


Figura 12.11. Tercera pantalla de la pestaña 3



Figura 12.12. Tercera pantalla de la pestaña 3



Figura 12.13. Tercera pantalla de la pestaña 3

#### 12.1.2.4. Pestaña 4: Progreso del lector

### Pantalla Principal

Estadísticas → Muestra el número total de páginas leídas, el libro más reciente, el libro más largo leído y la clasificación de libros (sin comenzar, comenzados, finalizados).

Botón de gráfico → Permite acceder a otra pantalla y visualizar un gráfico de barras con el porcentaje de páginas leídas de todos los libros de la biblioteca del lector.



Figura 12.14. Pantalla principal de la pestaña 4



Figura 12.15. Pantalla principal de la pestaña 4



Figura 12.16. Pantalla principal de la pestaña 4



Figura 12.17. Pantalla principal de la pestaña 4

### Pantalla de Gráfico

Gráfico de Barras → Representa el porcentaje de páginas leídas para cada libro.

Leyenda → Indica a qué libro corresponde cada barra del gráfico.

Botón de Regreso → Permite volver a la pantalla principal de la pestaña 4.



Figura 12.18. Pantalla secundaria de la pestaña 4

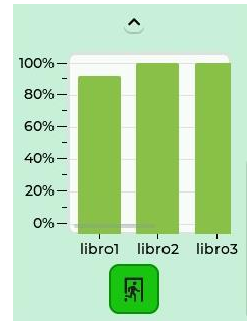


Figura 12.19. Pantalla secundaria de la pestaña 4