PROYECTO FINAL SISTEMAS EMPOTRADOS.

Roberto Ballesteros Remesal.

Marta Caballero Rivera.

Andrea López Herradón.

Inés Pacheco Pérez.

Contenido

1. Introducción

El objetivo principal de nuestro proyecto es poder desarrollar un sistema de control inteligente utilizando el ESP32, un microcontrolador con capacidad para conexión WiFi y Bluetooth, que permite la interacción a través de una interfaz web y una pantalla LCD para mostrar información y que el usuario pueda saber el estado del sistema. Este sistema está diseñado para ofrecer un control remoto sobre diversos dispositivos, como un reproductor de audio, a través de un módulo DFPlayer Mini y un altavoz.

Este proyecto no solo sirve como ejemplo de la versatilidad del ESP32 para integrarse en sistemas de control remoto y dispositivos interactivos, sino también como una prueba de concepto para la creación de dispositivos IoT con interfaces de usuario accesibles y prácticas.

2. Descripción del proyecto

El objetivo de este proyecto es construir un asistente virtual similar a Alexa pero mucho más económico, acercándose así a todo el público posible. Utilizando una placa ESP32, que interactúe de manera inteligente con el usuario mediante comandos y respuestas. Este asistente dependerá de módulos WiFi o Bluetooth.

El sistema está compuesto por los siguientes componentes:

• ESP32: Este microcontrolador es el encargado de gestionar la conectividad WiFi, la comunicación entre los diferentes módulos y la interacción con el usuario. Se encarga de procesar los comandos recibidos desde la interfaz web, controlar el módulo DFPlayer Mini para la reproducción de música y gestionar la visualización de información en la pantalla LCD.

• Módulo DFPlayer Mini: Este pequeño módulo es el responsable de la reproducción de archivos MP3 almacenados en una tarjeta microSD. A través de los comandos enviados desde el ESP32, se pueden reproducir diferentes pistas de audio. El volumen de la música también se puede ajustar con los botones de la interfaz web.

• Pantalla LCD: Utilizada para mostrar información de estado, como la música que se está reproduciendo o cualquier otra información relevante sobre el sistema. También se utilizará para mostrar el resultado de las operaciones en el modo calculadora.

• LEDs: Se utilizan para indicar el estado del sistema. Un LED verde muestra que el sistema está funcionando correctamente, mientras que un LED azul se enciende cuando el sistema está procesando información, indicando que una

acción está en curso, como la ejecución de una operación matemática o la reproducción de música.

• Potenciómetro: Se utiliza para ajustar el brillo que le llega a la pantalla LCD.

• Altavoz: Se conecta al módulo DFPlayer Mini y es utilizado para reproducir el sonido de las pistas de música.

• Cables y resistencias: Se utilizan para la conexión de los diferentes componentes, como los LEDs, la pantalla LCD y otros módulos. Las resistencias se emplean para limitar el flujo de corriente y proteger los componentes.

• Protoboard: Para montar y conectar los diferentes componentes, facilitando el ensamblaje y las modificaciones del circuito.

La interacción del usuario con el sistema comienza cuando se conecta a la red WiFi y accede a la interfaz web, donde se presentan botones para controlar la reproducción de música, ajustar el volumen y activar diferentes modos del sistema. Al presionar un botón, el ESP32 recibe la acción y controla el módulo DFPlayer Mini para reproducir una pista específica.

2.1. Pasos para la realización del proyecto:

1. Planificación inicial: Para la correcta realización del proyecto, comenzamos analizando las diferentes opciones disponibles para los componentes y la estructura del sistema. Consideramos qué microcontrolador utilizar, decidiendo que el ESP32 sería la mejor opción debido a su conectividad WiFi y su capacidad para manejar diversos periféricos. También discutimos qué dispositivos utilizar, decidiendo incluir el altavoz, la pantalla LCD y los LEDs como componentes clave para la interacción y el control del sistema.

2. Pruebas iniciales de los componentes: Una vez definido el enfoque del proyecto, comenzamos probando cada componente de manera individual. Esto incluyó probar el LEDs para verificar su correcto funcionamiento como indicadores de estado, probar el altavoz con el módulo DFPlayer Mini para la reproducción de música y realizar pruebas sencillas con el código para asegurar que cada componente respondiera adecuadamente antes de integrarlos en el proyecto.

3. Montaje del hardware: Tras las pruebas iniciales, comenzamos la fase de montaje del hardware. Conectamos el ESP32 con los otros componentes, como la pantalla LCD, los LEDs, el DFPlayer Mini y el potenciómetro para el control de volumen. Utilizamos una protoboard para realizar las conexiones, asegurándonos de que cada componente estuviera correctamente conectado y funcionando como se esperaba.

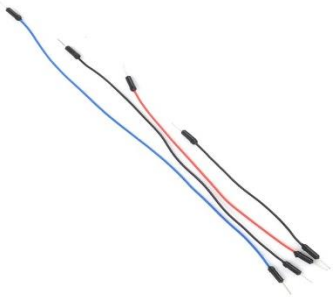
4. Desarrollo del software: Con el hardware montado, iniciamos la programación del ESP32. El software consistió en la creación de un servidor web para permitir la interacción remota con el sistema, la gestión de los comandos de reproducción de música, la visualización en la pantalla LCD y la activación de los diferentes modos de operación. Utilizamos los códigos previos que habíamos probado para ajustar y optimizar el funcionamiento de cada componente dentro del sistema. Del mismo modo hemos creado audios adaptados a nuestras respuestas por parte del altavoz inteligente para cada una de las opciones.

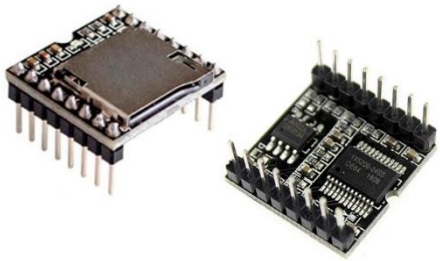
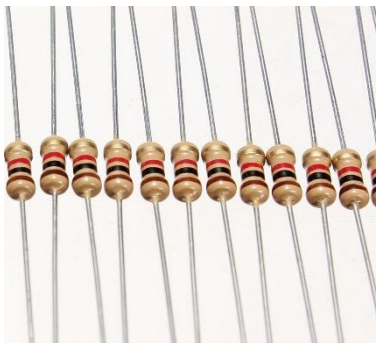
5. Integración de hardware y software: Finalmente, combinamos el hardware con el software. Realizamos diversas pruebas para asegurarnos de que todas las funcionalidades operaran correctamente, desde la correcta visualización en la pantalla LCD hasta la reproducción de música en el altavoz y la respuesta de los botones. Tras realizar las pruebas, ajustamos los detalles finales y confirmamos que el sistema cumpliera con los objetivos establecidos. Realizando así el empotrado del sistema en nuestra caja reestructurada y adaptada a nuestros componentes.

2.2. Reparto de tareas

Las tareas se han repartido de manera que todos hemos participado en todas las áreas de forma equitativa, ya que ha sido esencial para el éxito del proyecto. Para avanzar con el hardware, hemos realizado reuniones en la biblioteca y en las clases prácticas de la asignatura, mientras que, para el software, lo que hemos hecho ha sido tenerlo todo en un ordenador y todos teníamos el código, para poder modificarlo cuando quisiéramos. De igual manera, tanto la memoria, el blog y la presentación se han realizado de forma colaborativa, compartiendo el progreso del proyecto a medida que cada uno iba modificándolo y haciendo cambios.

3. Materiales

Material	Total utilizados
Fuente de alimentación (4,5 V) 	3
Plug para la pila 	1
Placa base - protoboard (630 puntos) 	2
Cables (macho – macho, hembra – hembra) 	37
Altavoz 	1

DFPlayer Mini		1
Resistencias (220kΩ)		2
Pantalla LCD		1
ESP32		1
Potenciómetro		1

Caja de cartón		1
Dispositivo móvil		1

3.1. Costes de materiales

MATERIAL	PRECIO
Fuente de alimentación	2
Plug para la pila	Caja
Placa base-protoboard	Caja
Cable tipo macho macho	Caja
Altavoz	Reutilizado de un antiguo altavoz
DFPlayer Mini	5,94
Resistencias	Caja
Pantalla LCD	Caja
ESP32	10,19
Potenciometro	Caja
Caja de cartón	Reutilizada
Dispositivo móvil	Reutilizado
Total	18,13

4. Hardware

LCD	ESP32
GND	GND
5V	5V
V0	POTENCIOMETRO
RS	G22
RW	GND
E	G23
D4	G18
D5	G19
D6	G21
D7	G5
A	5V
K	GND

DFPLAYER	ESP32
VCC	5V
GND	GND
RX	G26
TX	G27

DFPlayer cuenta con dos conexiones adicionales al altavoz en las entradas SPK_1.

Contamos con un potenciómetro conectado a 5V, LCD y GND.

5. Software

```
#include <WiFi.h>
#include <WebServer.h>
#include <DFRobotDFPlayerMini.h>
#include <HardwareSerial.h>
#include <LiquidCrystal.h>

// Pines del LCD
#define RS 22
#define E  23
#define D4 18
#define D5 19
#define D6 21
#define D7 5

// Pines DFPlayer
#define RX_PIN 27
#define TX_PIN 26

// LEDs
#define LED_VERDE 2
#define LED_ROJO 4

// WiFi
const char* ssid = "ESP32_Audio";
const char* password = "123";
WebServer server(80);

// Objetos
HardwareSerial mySerial(1);
DFRobotDFPlayerMini player;
LiquidCrystal lcd(RS, E, D4, D5, D6, D7);

// Variables para calculadora
bool modoCalculadora = false;
String expresion = "";

// Variables para cuestionario
bool esperandoRespuesta = false;
int preguntaActual = 0;  // 1 a 3

void encenderLedRojoBreve() {
  digitalWrite(LED_ROJO, HIGH);
  delay(300);
  digitalWrite(LED_ROJO, LOW);
}

void reproducirPregunta(int numero) {
```

```
  lcd.clear();
  lcd.print("Pregunta " + String(numero));
  switch (numero) {
    case 1: player.play(6); break;   // 0006.mp3
    case 2: player.play(9); break;   // 0009.mp3
    case 3: player.play(10); break;  // 0010.mp3
  }
}

void handleRoot() {
  server.send(200, "text/html", R"rawliteral(
    <html><head>
      <title>ESP32 Control</title>
      <style>
        body { font-family: Arial; text-align: center; background: #f4f4f4; }
        button { font-size: 20px; padding: 15px; margin: 5px;
                 background-color: #4CAF50; color: white; border: none;
                 border-radius: 5px; cursor: pointer; }
        button:hover { background-color: #45a049; }
      </style>
      <script>
        function sendInput(c){ fetch('/calc?input='+encodeURIComponent(c)); }
      </script>
    </head><body>
      <h2>Panel de Control</h2>
      <button onclick="fetch('/btn1')">1</button>
      <button onclick="fetch('/btn2')">2</button>
      <button onclick="fetch('/btn3')">3</button>
      <button onclick="fetch('/btn4')">4</button>
      <button onclick="fetch('/btn5')">5</button>
      <button onclick="fetch('/btn6')">6</button><br><br>
      <button onclick="fetch('/pause')">Pausar/Reanudar</button><br><br>
      <button onclick="fetch('/volup')">Volumen +</button>
      <button onclick="fetch('/voldown')">Volumen -</button><br><br>
      <h3>Calculadora</h3>
      <div>
        <button onclick="sendInput('1')">1</button>
        <button onclick="sendInput('2')">2</button>
        <button onclick="sendInput('3')">3</button>
        <button onclick="sendInput('+')">+</button><br>
        <button onclick="sendInput('4')">4</button>
        <button onclick="sendInput('5')">5</button>
        <button onclick="sendInput('6')">6</button>
        <button onclick="sendInput('-')">-</button><br>
        <button onclick="sendInput('7')">7</button>
        <button onclick="sendInput('8')">8</button>
        <button onclick="sendInput('9')">9</button>
        <button onclick="sendInput('*')">*</button><br>
        <button onclick="sendInput('0')">0</button>
```

```
      <button onclick="sendInput('C')">C</button>
      <button onclick="sendInput('=')">=</button>
      <button onclick="sendInput('/')">/</button>
    </div><br>
    <button onclick="fetch('/verdadero')">Verdadero</button>
    <button onclick="fetch('/falso')">Falso</button>
  </body></html>
)rawliteral");
}

int evaluarExpresion(String expr) {
  expr.replace(" ", "");
  int res = 0;
  char op = 0;
  String num = "";
  for (char c : expr) {
    if (isdigit(c)) num += c;
    else {
      if (num.length()) {
        int v = num.toInt();
        if (!op) res = v;
        else if (op == '+') res += v;
        else if (op == '-') res -= v;
        else if (op == '*') res *= v;
        else if (op == '/') res /= v;
        num = "";
      }
      op = c;
    }
  }
  if (num.length()) {
    int v = num.toInt();
    if (!op) res = v;
    else if (op == '+') res += v;
    else if (op == '-') res -= v;
    else if (op == '*') res *= v;
    else if (op == '/') res /= v;
  }
  return res;
}
// Calculadora
void handleCalcInput() {
  if (!modoCalculadora) {
    server.send(200, "text/plain", "Calculadora desactivada");
    return;
  }
  if (!server.hasArg("input")) {
    server.send(400, "text/plain", "Sin entrada");
    return;
```

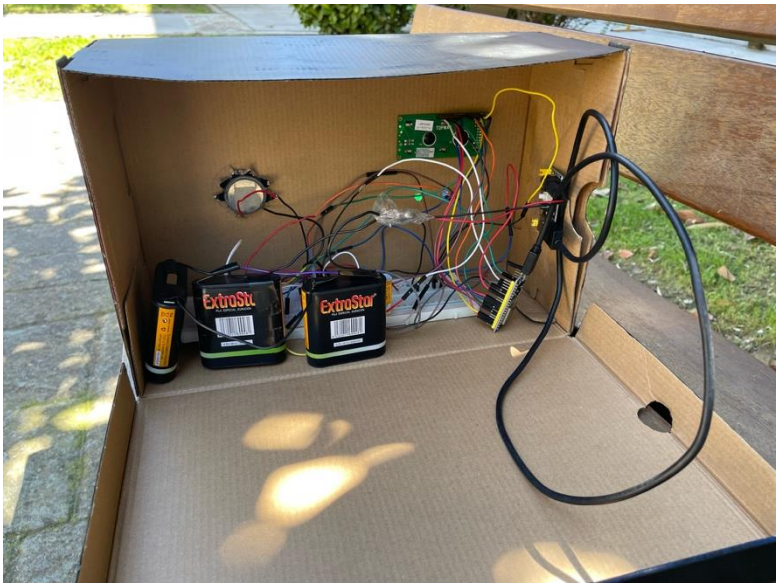
```
  }
  String in = server.arg("input");
  if (in == "C") {
    expresion = "";
    lcd.clear(); lcd.print("Calculadora");
  } else if (in == "=") {
    int r = evaluarExpresion(expresion);
    lcd.clear();
    lcd.print("Resultado:");
    lcd.setCursor(0,1);
    lcd.print(r);
    expresion = "";
  } else {
    expresion += in;
    lcd.clear(); lcd.print(expresion);
  }
  server.send(200, "text/plain", "OK");
}

// Preguntas
void handleBtn5() {
  encenderLedRojoBreve();
  preguntaActual = 1;
  esperandoRespuesta = true;
  reproducirPregunta(1);
  server.send(200, "text/plain", "Inicia quiz");
}

void handleVerdadero() {
  if (!esperandoRespuesta) {
    server.send(200, "text/plain", "No hay pregunta");
    return;
  }
  bool correcta = (preguntaActual == 1);
  if (correcta) {
    lcd.clear(); lcd.print("Correcto!");
    player.play(7);
    delay(5000);
    preguntaActual++;
    if (preguntaActual <= 3) reproducirPregunta(preguntaActual);
    else { esperandoRespuesta=false; lcd.clear(); lcd.print("Fin preguntas"); }
  } else {
    lcd.clear(); lcd.print("Incorrecto");
    player.play(8);
    delay(5000);
    reproducirPregunta(preguntaActual);
  }
  server.send(200, "text/plain", "OK");
}
```

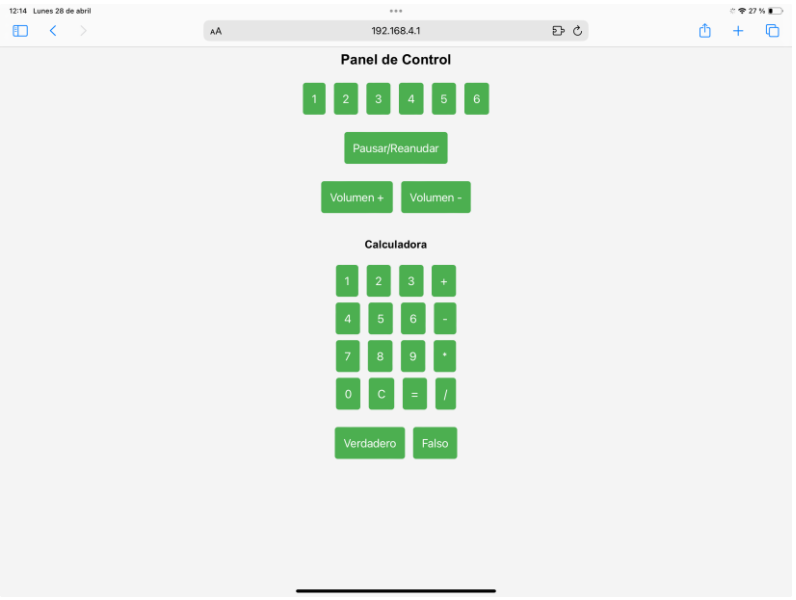
```
200.
201.void handleFalso() {
202.  if (!esperandoRespuesta) {
203.    server.send(200, "text/plain", "No hay pregunta");
204.    return;
205.  }
206.  bool correcta = (preguntaActual==2 || preguntaActual==3);
207.  if (correcta) {
208.    lcd.clear(); lcd.print("Correcto!");
209.    player.play(7);
210.    delay(5000);
211.    preguntaActual++;
212.    if (preguntaActual <= 3) reproducirPregunta(preguntaActual);
213.    else { esperandoRespuesta=false; lcd.clear(); lcd.print("Fin preguntas"); }
214.  } else {
215.    lcd.clear(); lcd.print("Incorrecto");
216.    player.play(8);
217.    delay(5000);
218.    reproducirPregunta(preguntaActual);
219.  }
220.  server.send(200, "text/plain", "OK");
221.}
222.
223.// Musica
224.void handleBtn1() {
225.  encenderLedRojoBreve();
226.  player.play(3);
227.  lcd.clear(); lcd.print("Musica Alegre");
228.  server.send(200, "text/plain", "Play 003");
229.}
230.
231.void handleBtn2() {
232.  encenderLedRojoBreve();
233.  player.play(4);
234.  lcd.clear(); lcd.print("Musica Ambiente");
235.  server.send(200, "text/plain", "Play 004");
236.}
237.
238.void handleBtn3() {
239.  encenderLedRojoBreve();
240.  player.play(5);
241.  lcd.clear(); lcd.print("Modo Estudio");
242.  server.send(200, "text/plain", "Play 005");
243.}
244.
245.void handlePause() {
246.  encenderLedRojoBreve();
247.  player.pause();
248.  server.send(200, "text/plain", "Pause/Resume");
```

```
}

void handleVolUp() {
  encenderLedRojoBreve();
  player.volumeUp();
  server.send(200, "text/plain", "Vol +");
}

void handleVolDown() {
  encenderLedRojoBreve();
  player.volumeDown();
  server.send(200, "text/plain", "Vol -");
}

void handleBtn4() {
  encenderLedRojoBreve();
  modoCalculadora = true;
  expresion = "";
  lcd.clear(); lcd.print("Calculadora");
  server.send(200, "text/plain", "Calc ON");
}

// Apagar Sistema
void handleBtn6() {
  digitalWrite(LED_VERDE, LOW);
  digitalWrite(LED_ROJO, LOW);
  lcd.clear(); lcd.print("Sistema apagado");
  server.send(200, "text/plain", "Shutdown");
  delay(2000);
  while (true);
}

void setup() {
  Serial.begin(115200);
  mySerial.begin(9600, SERIAL_8N1, RX_PIN, TX_PIN);
  lcd.begin(16, 2);

  pinMode(LED_VERDE, OUTPUT);
  pinMode(LED_ROJO, OUTPUT);
  digitalWrite(LED_VERDE, HIGH);
  digitalWrite(LED_ROJO, LOW);

  lcd.clear(); lcd.print("Iniciando...");

  if (!player.begin(mySerial)) {
    lcd.clear(); lcd.print("Error DFPlayer");
    while (1);
  }

```

```
  player.volume(20);
  player.play(1); delay(5000);
  player.play(2); delay(5000);
  lcd.clear(); lcd.print("Listo para usar");

  WiFi.softAP(ssid, password);
  Serial.println("WiFi AP iniciado: 192.168.4.1");

  server.on("/",       handleRoot);
  server.on("/btn1",   handleBtn1);
  server.on("/btn2",   handleBtn2);
  server.on("/btn3",   handleBtn3);
  server.on("/btn4",   handleBtn4);
  server.on("/btn5",   handleBtn5);
  server.on("/btn6",   handleBtn6);
  server.on("/pause",  handlePause);
  server.on("/volup",  handleVolUp);
  server.on("/voldown",handleVolDown);
  server.on("/calc",   handleCalcInput);
  server.on("/verdadero", handleVerdadero);
  server.on("/falso",     handleFalso);

  server.begin();
}

void loop() {
  server.handleClient();
}
```


Sistema apagado


ExtroStar
ExtroStar


12:14 Lunes 28 de abril
27 %
AA
192.168.4.1
Panel de Control
1 2 3 4 5 6
Pausar/Reanudar
Volumen +
Volumen -
Calculadora
1 2 3 +
4 5 6 -
7 8 9 *
0 C = /
Verdadero
Falso

Bibliografía:

Imagen fuente de alimentación

Imagen plug para pila

Imagen Placa base

Imagen cables

Imagen resistencias

Imagen altavoz

Imagen DFPlayer Mini

Imagen Pantalla LCD

Imagen ESP32

Imagen potenciómetro

Imagen caja de cartón

Imagen dispositivo móvil

Video ESP32 con altavoz

Video ESP32 bluetooth