UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS
Acreditación Institucional de Alta Calidad

Autores:
Diego Leonardo Alvarez Hernández
Julian Andres Guavita Herrera

Diseño e implementación de un pulsioxímetro con registro en base de datos e interfaz de visualización para el monitoreo de signos vitales

Director:

PhD. Pablo Emilio Rozo García

Universidad Distrital Francisco José de Caldas

Facultad de Ingeniería

Bogotá D.C.

2025

ÍNDICE

TABLA DE FIGURAS 6
ÍNDICE DE TABLAS 15
ÍNDICE DE ECUACIONES 17
1. PREGUNTA-PROBLEMÁTICA 18
2. JUSTIFICACIÓN 20
3. OBJETIVOS 21
3.1. GENERAL 21
3.2. ESPECÍFICOS 21
4. MARCO TEÓRICO 22
4.1. SATURACIÓN DEL OXÍGENO EN LA SANGRE 22
4.2. PULSO CARDIACO 22
4.3. SENSORES 22
4.3.1. FOTOPLETISMOGRAFÍA 23
4.3.1.1. ESPECTROFOTOMETRÍA 23
4.3.1.1.1. MAX 30100 24
4.3.1.1.2. MAX 30102 24
4.3.1.1.3. MAX 30105 24
4.4. MICROCONTROLADORES 25
4.4.1. ARDUINO 25
4.4.1.1. ARDUINO UNO 25
4.4.1.2. ARDUINO MEGA 25
4.4.1.3. ARDUINO NANO 26
4.4.1.4. ARDUINO PRO 26
4.4.2. PSOC 26
4.4.3. ESP32 27
4.4.4. PIC 27
4.4.4.1. MICROCONTROLADORES 8 BITS 28
4.4.4.2. MICROCONTROLADORES 16 BITS 28
4.4.4.3. MICROCONTROLADORES 32 BITS 28
4.4.5. RASPBERRY 28
4.4.5.1. RASPBERRY PI 3 MODELO A+ 29
4.4.5.2. RASPBERRY PI 3 MODELO B+ 29
4.4.5.3. RASPBERRY PI 4 MODELO B+ 29
4.4.5.4. RASPBERRY PI ZERO 29
4.5. SISTEMAS DE COMUNICACIÓN 29
4.5.1. REDES INALÁMBRICAS 29
4.5.1.1. TIPOS DE REDES INALÁMBRICAS 30
4.5.1.1.1. REDES WPAN 30
4.5.1.1.2. REDES WLAN 30
4.5.1.1.3. REDES WMAN 31
4.5.1.1.4. REDES WIDE AREA 31
4.5.1.1.5. WIFI 31

4.5.1.1.6. BLUETOOTH 31
4.5.1.1.7. ZIGBEE 32
4.6. BASES DE DATOS 32
4.6.1. MySQL 33
4.6.2. MICROSOFT SQL SERVER 33
4.6.3. ORACLE 34
4.6.4. POSTGRESQL 34
4.6.5. DB2 35
4.7. TECLADO MATRICIAL 35
4.8. DISPLAY 36
4.8.1. DISPLAY 7 SEGMENTOS 36
4.8.2. LCD 36
4.8.3. TFT 37
4.8.4. OLED 37
4.9. SERVIDORES WEB 37
4.9.2. MICROSOFT IIS 38
4.9.3. GOOGLE WEB SERVER 38
4.9.4. NGINX 38
4.9.5. LIGHTTPD 38
4.10. PHPMYADMIN 38
5. ESTADO DEL ARTE 39
5.1. IMPLEMENTACIÓN DEL PROTOTIPO DE UN SISTEMA DE MONITOREO DE SIGNOS VITALES: PULSO, TEMPERATURA Y SATURACIÓN DE OXÍGENO PARA PACIENTES [30] 39
5.2. DISEÑO DE UN SISTEMA DE INSTRUMENTACIÓN WEARABLE PARA LA MEDIDA DE LA SATURACIÓN DE OXÍGENO EN LA SANGRE [31] 40
5.3. SISTEMA PARA LA MONITORIZACIÓN DE LA SATURACIÓN DE OXÍGENO EN SANGRE [32] 40
5.4. DISEÑO Y DESARROLLO DE UN DISPOSITIVO ELECTRÓNICO DE USO PERSONAL PARA LA MONITORIZACIÓN CONTINUADA DE LA SATURACIÓN DE OXÍGENO EN SANGRE [33] 41
5.5. DISPOSITIVO TELEMÉTRICO PARA MONITOREO DE FRECUENCIA CARDIACA Y SATURACIÓN DE OXÍGENO [34] 42
6. METODOLOGÍA 43
6.1. FASE 1 43
6.2. FASE 2 43
6.3. FASE 3 43
6.4. FASE 4 44
6.5. FASE 5 44
6.6. FASE 6 44
7. DISEÑO E IMPLEMENTACIÓN DEL SISTEMA 45
7.1. DISEÑO GENERAL DEL SISTEMA 45
7.1.1. CAPA DE ADQUISICIÓN Y PROCESAMIENTO 45
7.1.2. CAPA DE ALMACENAMIENTO 45
7.1.3. CAPA DE LÓGICA Y SERVICIOS 46

7.1.4. CAPA DE PRESENTACIÓN 46
7.2. ESPECIFICACIÓN TÉCNICA DEL SISTEMA 46
7.2.1. SENSOR ÓPTICO MAX30102 46
7.2.2. MICROCONTROLADOR ESP32 47
7.2.3. BASE DE DATOS SUPABASE (POSTGRESQL) 47
7.2.4. LÓGICA DE SERVICIOS (BACKEND) 48
7.2.5. INTERFAZ DE USUARIO (FRONTEND) 48
7.2.6. COMPONENTES AUXILIARES Y PERIFÉRICOS 49
7.2.6.1. TECLADO MATRICIAL 4X4 49
7.2.6.2. PANTALLA OLED (SH1106) 49
7.2.6.3. BATERÍA DE LITIO 18650 Y PORTA PILAS 50
7.2.6.4. MÓDULO TP4056 50
7.3. ESQUEMA ELÉCTRICO Y CONEXIONES 51
7.3.1. DIAGRAMA DE CONEXIÓN 51
7.3.2. ESPECIFICACIÓN DE PINES Y PROTOCOLOS 51
7.3.3. PRECAUCIONES ELÉCTRICAS Y TOLERANCIAS 54
7.3.4. IMPLEMENTACIÓN FÍSICA 54
7.4. DESARROLLO DEL DISPOSITIVO 56
7.4.1. VERIFICACIÓN DEL SENSOR Y CONECTIVIDAD 57
7.4.2. DESARROLLO DEL ALGORITMO DE BPM Y SPO_2 58
7.4.2.1. CÁLCULO DE LA FRECUENCIA CARDIACA 58
7.4.2.2. CÁLCULO DE LA SATURACIÓN DE OXÍGENO EN LA SANGRE 59
7.4.2.2.1. FILTRADO DE LA SEÑAL 60
7.4.2.2.2. CÁLCULO DEL FACTOR R 60
7.4.2.2.3. CONVERSIÓN FINAL A SPO_2 62
7.4.3. INTEGRACIÓN DE LA PANTALLA OLED 63
7.4.4. INTEGRACIÓN DEL TECLADO MATRICIAL 66
7.4.5. CONFIGURACIÓN Y CONEXIÓN WI-FI 71
7.4.6. FUNCIONES DE DETECCIÓN Y OPTIMIZACIÓN 74
7.4.6.1. MODO DE DETECCIÓN DEL SENSOR 74
7.4.6.2. GESTIÓN DE LA PRESENCIA O AUSENCIA DEL DEDO 75
7.4.6.3. REINICIO DE VARIABLES 75
7.4.6.4. MODO DE BAJO CONSUMO 76
7.4.7. BASE DE DATOS 78
7.4.7.1. TABLA USUARIOS 78
7.4.7.2. TABLA MEDICIONES 79
7.4.7.3. TABLA ALERTAS 80
7.4.7.4. INTERCONEXIÓN DEL ESQUEMA RELACIONAL 81
7.4.8. DESARROLLO DE LA INTERFAZ DEL SISTEMA 81
7.4.8.1. DESARROLLO DEL BACKEND 82
7.4.8.1.1. MÓDULO DE MEDICIONES 83
7.4.8.1.2. MÓDULO DE USUARIOS 85
7.4.8.1.3. MÓDULO DE ALERTAS 86
7.4.8.1.4. MECANISMOS DE ROBUSTEZ Y EXPERIENCIA DE USUARIO 87

7.4.8.2. DESARROLLO DEL FRONTEND 88
7.4.8.2.1. PANEL DEL PACIENTE 89
7.4.8.2.2. PANEL DEL MÉDICO 91
7.4.8.2.3. PANEL DEL ADMINISTRADOR 93
7.4.9. INTEGRACIÓN TECNOLÓGICA Y FLUJO DE DATOS 96
7.4.9.1. CICLO 1: REGISTRO Y PREPARACIÓN PARA LA INGESTA 96
7.4.9.2. CICLO 2: CONSULTA Y VISUALIZACIÓN SEGURA 97
8. PROTOCOLO DE PRUEBA 99
8.1. POBLACIÓN 99
8.1.1. TONO DE PIEL 99
8.2. PRUEBA DE PRECISIÓN Y ESTABILIDAD 101
8.3. ANÁLISIS DE RESULTADOS 102
8.3.1. DESVIACIÓN ESTÁNDAR 102
8.3.2. SESGO PROMEDIO 105
8.3.2.1. SESGO ABSOLUTO PROMEDIO 107
9. PRESUPUESTO 111
9.1. RECURSOS FÍSICOS 111
9.2. RECURSOS HUMANOS 111
9.3. PRESUPUESTO TOTAL 112
10. CONCLUSIONES 113
10.1. PRIMER OBJETIVO ESPECÍFICO 113
10.2. SEGUNDO OBJETIVO ESPECÍFICO 113
10.3. TERCER OBJETIVO ESPECÍFICO 113
10.4. CUARTO OBJETIVO ESPECÍFICO 114
10.5. QUINTO OBJETIVO ESPECÍFICO 114
11. TRABAJOS FUTUROS 115
12. BIBLIOGRAFÍA 115
13. WEBGRAFÍA 121
ANEXOS 123
A.1 MANUAL DEL USUARIO 123
PÁGINA WEB 123
PROCESO DE REGISTRO Y AUTENTICACIÓN 123
PAGINA PARA EL PACIENTE 126
PAGINA PARA EL MÉDICO 131
PAGINA PARA EL ADMINISTRADOR 146
PULSIOXIMETRO 163
MODO OFFILINE 165
MODO ONLINE 169
A.2 CÓDIGO PARA EL PROGRAMA OPERACIONAL DEL PULSIOXÍMETRO 177
A.3 RESULTADOS 220

TABLA DE FIGURAS

4.* MARCO TEÓRICO** ***22

Figura 1, Diagrama pulso cardíaco. [40] **22**

Figura 2, Estructura de tipos de sensores. [52] **23**

Figura 3, Esquema fotopletismografía. [39] **23**

Figura 4, Esquema general sistemas de espectrometría. [51] **23**

Figura 5, Sensor MAX 30102. [50] **24**

Figura 6, Dispositivo MCU ARDUINO. [49] **25**

Figura 7, Dispositivo MCU Psoc. [48] **26**

Figura 8, Dispositivo MCU ESP32. [22] **27**

Figura 9, Dispositivo MCU Pic. [47] **28**

Figura 10, Dispositivo MCU RASPBERRY. [46] **29**

Figura 11, Esquema de redes inalámbricas. [41] **30**

Figura 12, Tecnología WIFI. [45] **31**

Figura 13, Tecnología BLUETOOTH. [44] **32**

Figura 14, Tecnología ZIGBEE. [43] **32**

Figura 15, Diagrama de red de base de datos. [42] **33**

Figura 16, Arreglo de pulsadores de un teclado matricial. [53] **35**

Figura 17, Display 7 segmentos. [35] **36**

Figura 18, Pantalla LCD 16x2 con módulo I2C. [56] **36**

Figura 19, Pantalla TFT ILI9163. [55] **37**

Figura 20, Pantalla OLED 128x64 basado en SSD1306. [54] **37**

5.* ESTADO DEL ARTE** ***39

Figura 21, Estructura de funciones del sistema de monitoreo remoto. [30] **39**

Figura 22, Diagrama de bloques del sistema. [31] **40**

Figura 23, Diagrama de bloques del sistema. [32] **41**

Figura 24, Diagrama de bloques del sistema completo. [33] **41**

Figura 25, Diagrama de bloques del sistema completo. [34] **42**

7.* DISEÑO E IMPLEMENTACIÓN DEL SISTEMA** ***45

Figura 26, Arquitectura general del sistema en cuatro capas. [Elaboración propia] **45**

Figura 27, Diagrama de conexión. [Elaboración propia] **51**

Figura 28, Modelo 3D de la Placa de Circuito Impreso PCB con componentes. [Elaboración propia] **55**

Figura 29, Placa de Circuito Impreso PCB ensamblada con componentes. [Elaboración propia] 55

Figura 30, Dispositivo pulsioxímetro finalizado e integrado en la carcasa. [Elaboración propia] 56

Figura 31, Adquisición inicial de datos del algoritmo ejemplo. [Elaboración propia] 58

Figura 32, Variables de estado y búfer circular para el suavizado de BPM. [Elaboración propia] 58

Figura 33, Ecuación para el cálculo de la frecuencia cardíaca. [Elaboración propia] 59

Figura 34, Aplicación del promedio móvil para la mitigación de ruido y estabilización final del BPM. [Elaboración propia] 59

Figura 35, Obtención de muestras de fotopletismografía para el cálculo de spo2. [Elaboración propia] 60

Figura 36, Ecuación del filtro digital paso bajo exponencial para señales ppg. [Elaboración propia] 60

Figura 37, Cálculo simplificado de la componente ac de las longitudes de ondas. [Elaboración propia] 62

Figura 38, Cálculo del factor de proporción R. [Elaboración propia] 62

Figura 39, Filtro de suavizado exponencial para R. [Elaboración propia] 62

Figura 40, Ecuación de calibración lineal y restricción de rango para la spo2. [Elaboración propia] 63

Figura 41, Cálculo final de la saturación estable. [Elaboración propia] 63

Figura 42, Pantalla de bienvenida y presentación del sistema. [Elaboración propia] 64

Figura 43, Interfaz del portal de configuración Wi-Fi. [Elaboración propia] 64

Figura 44, Pantalla de verificación de cédula registrada. [Elaboración propia] 64

Figura 45, Visualización de datos biométricos en modo online. [Elaboración propia] 65

Figura 46, Visualización de datos biométricos en modo offline. [Elaboración propia] 65

Figura 47, Notificación de ausencia de dedo en sensor. [Elaboración propia] 66

Figura 48, Menú de decisión al salir del modo de bajo consumo. [Elaboración propia] 66

Figura 49, Asignación de pines y mapeo del teclado 4x4 en el ESP32. [Elaboración propia] 66

Figura 50, Menú principal de selección de modo online/offline. [Elaboración propia] 67

Figura 51, Interfaz de ingreso numérico de cédula. [Elaboración propia] 67

Figura 52, Opciones para cancelar o confirmar el número de cédula. [Elaboración propia] 68

Figura 53, Opciones para cancelar la configuración Wi-Fi. [Elaboración propia] 68

Figura 54, Opciones para cancelar o confirmar el número de cédula. [Elaboración propia] 68

Figura 55, Control rápido en modo operativo. [Elaboración propia] 69

Figura 56, Panel de opciones y control rápido en modo operativo. [Elaboración propia] 70

Figura 57, Opción para seleccionar el modo después del modo de bajo consumo. [Elaboración propia] 70

Figura 58, Interfaz de configuración Wi-Fi mediante portal cautivo. [Elaboración propia] 71

Figura 59, Proceso de configuración fallida por tiempo de espera o cancelación manual. [Elaboración propia] 72

Figura 60, Configuración de conexión Wi-Fi establecida, confirmación de SSID y dirección IP. [Elaboración propia] 72

Figura 61, Estructura para la consulta y verificación del número de cédula a la base de datos. [Elaboración propia] 73

Figura 62, Estructura del envío de datos fisiológicos a la base de datos. [Elaboración propia] 74

Figura 63, Rutina de verificación de la conexión del sensor y mensaje de error. [Elaboración propia] 74

Figura 64, Condición principal para la detección de la presencia del dedo. [Elaboración propia] 75

Figura 65, Lógica de activación del reinicio de variable. [Elaboración propia] 75

Figura 66, Función principal para el reinicio de variables. [Elaboración propia] 76

Figura 67, Lógica de activación por tiempo de inactividad del modo de bajo consumo. [Elaboración propia] 76

Figura 68, Secuencia de apagado del sensor con modo seleccionado y notificación al usuario. [Elaboración propia] 77

Figura 69, Secuencia de apagado del sensor sin modo seleccionado y notificación al usuario. [Elaboración propia] 77

Figura 70, Lógica de detección y función de reactivación del sistema. [Elaboración propia] 78

Figura 71, Estructura de la tabla Usuarios: Identificación y gestión de perfiles. [Elaboración propia] 78

Figura 72, Estructura de la tabla Mediciones: Registro de datos fisiológicos. [Elaboración propia] 79

Figura 73, Función y trigger para el filtro de ruido inicial de datos biométricos. [Elaboración propia] 80

Figura 74, Estructura de la tabla Alertas: Repositorio de eventos clínicos críticos. [Elaboración propia] 80

Figura 75, Función y trigger para la automatización de las alertas clínicas. [Elaboración propia] 81

Figura 76, Diagrama Entidad-Relación del esquema de la base de datos y sus interconexiones. [Elaboración propia] 81

Figura 77, Endpoint para la autenticación y generación de token JWT para el inicio de sesión. [Elaboración propia] 82

Figura 78, Endpoint de estructura de datos para el registro de usuarios nuevos. [Elaboración propia] 83

Figura 79, Endpoint para la eliminación de registros de mediciones. [Elaboración propia] 84

Figura 80, Endpoint para la actualización de datos de mediciones biométricas. [Elaboración propia] 84

Figura 81, Endpoint para la consulta jerarquizada de mediciones históricas. [Elaboración propia] 85

Figura 82, Endpoint para la creación de usuarios y asignación de roles. [Elaboración propia] 85

Figura 83, Endpoint para la actualización de perfiles y datos personales. [Elaboración propia] 86

Figura 84, Endpoint para la eliminación controlada de perfiles de usuario. [Elaboración propia] 86

Figura 85, Endpoint para la consulta de alertas críticas activas. [Elaboración propia] 87

Figura 86, Endpoint para la modificación del estado de la alerta. [Elaboración propia] 87

Figura 87, Endpoint para la eliminación definitiva de registros de alerta. [Elaboración propia] 87

Figura 88, Interfaz de inicio de sesión de usuarios. [Elaboración propia] 89

Figura 89, Interfaz de registro para nuevos usuarios pacientes. [Elaboración propia] 89

Figura 90, Dashboard del paciente: perfil de usuario y gráfica de tendencias biométricas. [Elaboración propia] 90

Figura 91, Dashboard del paciente: visualización del historial de mediciones recientes y listado de alertas. [Elaboración propia] 90

Figura 92, Panel de interfaz para la edición de datos personales del paciente. [Elaboración propia] 90

Figura 93, Dashboard del médico: vista general de gestión y listado de pacientes bajo supervisión. [Elaboración propia] 91

Figura 94, Dashboard del médico: panel de creación de nuevos usuarios. [Elaboración propia] 92

Figura 95, Dashboard médico: perfil de paciente y visualización de gráficas de tendencias clínicas. [Elaboración propia] 92

Figura 96, Dashboard médico: historial de mediciones del paciente y listado de alertas. [Elaboración propia] 93

Figura 97, Panel de edición del perfil de un paciente por el médico. [Elaboración propia] 93

Figura 98, Dashboard del administrador: vista general de gestión de todos los usuarios. [Elaboración propia] 94

Figura 99, Dashboard del administrador: panel de creación de nuevos usuarios. [Elaboración propia] 94

Figura 100, Dashboard del administrador: perfil de usuario y visualización de gráficas de tendencias clínicas. [Elaboración propia] 95

Figura 101, Dashboard del administrador: historial de mediciones de usuarios y listado de alertas. [Elaboración propia] 95

Figura 102, Panel de edición del perfil de un usuario por un administrador. [Elaboración propia] 96

Figura 103, Diagrama de flujo para el registro de datos. [Elaboración propia] 97

Figura 104, Diagrama de flujo para la visualización de datos. [Elaboración propia] 98

8.* PROTOCOLO DE PRUEBA *99

Figura 105, Escala de fototipos de Fitzpatrick. [Elaboración propia] 99

ANEXOS *123*

Figura 106, Pantalla de inicio resaltando el botón "Registrarse aquí". [Elaboración propia] 124

Figura 107, Pantalla de registro de usuario. [Elaboración propia] 124

Figura 108, Pantalla de registro mostrando mensaje de error por documento ya registrado. [Elaboración propia] 125

Figura 109, Pantalla de registro con mensaje de éxito. [Elaboración propia] 125

Figura 110, Pantalla de inicio de sesión mostrando mensaje de error por credenciales incorrectas. [Elaboración propia] 126

Figura 111, Pantalla general del perfil del paciente. [Elaboración propia] 126

Figura 112, Panel superior de bienvenida para el perfil de los pacientes y botón de cierre de sesión. [Elaboración propia] 127

Figura 113, Panel izquierdo del perfil del paciente. [Elaboración propia] 127

Figura 114, Pantalla de cambio de datos de perfil. [Elaboración propia] 128

Figura 115, Panel derecho con gráfica de tendencia de datos biométricos. [Elaboración propia] 128

Figura 116, Panel de tabla de registros médicos. [Elaboración propia] 129

Figura 117, Filtros de la tabla de registros médicos. [Elaboración propia] 129

Figura 118, Botón de contracción de la tabla de registros médicos. [Elaboración propia] 130

Figura 119, Panel inferior de tabla de alertas médicas. [Elaboración propia] 130

Figura 120, Filtros del panel inferior de la tabla de alertas médicas. [Elaboración propia] 131

Figura 121, Botón de contracción de la tabla de alertas médicas. [Elaboración propia] 131

Figura 122, Pantalla general del perfil de médico. [Elaboración propia] 132

Figura 123, Panel superior para el perfil de los médicos. [Elaboración propia] 132

Figura 124, Pantalla de registro de nuevos pacientes. [Elaboración propia] 133

Figura 125, Pantalla de registro mostrando mensaje de error por documento ya registrado. [Elaboración propia] 133

Figura 126, Pantalla de registro de nuevo paciente con mensaje de éxito. [Elaboración propia] 134

Figura 127, Panel izquierdo del perfil del médico. [Elaboración propia] 134

Figura 128, Pantalla de cambio de datos de perfil. [Elaboración propia] 135

Figura 129, Pantalla de edición de perfil mostrando mensaje de error por documento ya registrado. [Elaboración propia] 135

Figura 130, Pantalla de edición del perfil con mensaje de éxito. [Elaboración propia] 136

Figura 131, Pantalla de edición del perfil con mensaje de éxito y regreso al inicio de sesión. [Elaboración propia] 136

Figura 132, Panel derecho tabla de pacientes. [Elaboración propia] 137

Figura 133, Filtros de la tabla de pacientes. [Elaboración propia] 137

Figura 134, Botón para acceder a la información detallada de los pacientes. [Elaboración propia] 138

Figura 135, pantalla principal al acceder al botón de la vista de detalles de los pacientes. [Elaboración propia] 138

Figura 136, Panel superior del perfil de los pacientes y botón de regreso. [Elaboración propia] 139

Figura 137, Segundo panel del perfil del paciente. [Elaboración propia] 139

Figura 138, Pantalla de cambio de datos de perfil. [Elaboración propia] 140

Figura 139, Pantalla de edición de perfil mostrando mensaje de error por documento ya registrado. [Elaboración propia] 140

Figura 140, Pantalla de edición del perfil con mensaje de éxito. [Elaboración propia] 141

Figura 141, Pantalla de edición del perfil con mensaje de éxito y regreso al inicio de sesión. [Elaboración propia] 141

Figura 142, Tercer panel con gráfica de tendencia de datos biométricos. [Elaboración propia] 142

Figura 143, Panel de tabla de registros médicos. [Elaboración propia] 142

Figura 144, Filtros de la tabla de registros médicos. [Elaboración propia] 143

Figura 145, Botón para la edición del spo2 y bpm. [Elaboración propia] 143

Figura 146, Botón de contracción de la tabla de registros médicos. [Elaboración propia] 144

Figura 147, Panel inferior de tabla de alertas médicas. [Elaboración propia] 144

Figura 148, Filtros del panel inferior de la tabla de alertas médicas. [Elaboración propia] 145

Figura 149, Casilla para la edición del estado de alertas. [Elaboración propia] 145

Figura 150, Botón de contracción de la tabla de alertas médicas. [Elaboración propia] 146

Figura 151, Pantalla general del perfil de administrador. [Elaboración propia] 146

Figura 152, Panel superior para el perfil de los administradores. [Elaboración propia] 147

Figura 153, Pantalla de registro de nuevos usuarios. [Elaboración propia] 147

Figura 154, Pantalla de registro mostrando mensaje de error por documento ya registrado. [Elaboración propia] 148

Figura 155, Pantalla de registro de nuevo usuarios con mensaje de éxito. [Elaboración propia] 148

Figura 156, Panel izquierdo del perfil del administrador. [Elaboración propia] 149

Figura 157, Pantalla de cambio de datos de perfil. [Elaboración propia] 149

Figura 158, Pantalla de edición de perfil mostrando mensaje de error por documento ya registrado. [Elaboración propia] 150

Figura 159, Pantalla de edición del perfil con mensaje de éxito. [Elaboración propia] 150

Figura 160, Pantalla de edición del perfil con mensaje de éxito y regreso al inicio de sesión. [Elaboración propia] 151

Figura 161, Panel derecho tabla de usuarios. [Elaboración propia] 151

Figura 162, Filtros de la tabla de pacientes. [Elaboración propia] 152

Figura 163, Botón para la eliminación permanente de los usuarios. [Elaboración propia] 152

Figura 164, Confirmación para la eliminación permanente de los usuarios. [Elaboración propia] 153

Figura 165, Botón para acceder a la información detallada de los pacientes. [Elaboración propia] 153

Figura 166, pantalla principal al acceder al botón de la vista de detalles de los pacientes. [Elaboración propia] 154

Figura 167, Panel superior del perfil de los pacientes y botón de regreso. [Elaboración propia] 154

Figura 168, Segundo panel del perfil del paciente. [Elaboración propia] 155

Figura 169, Pantalla de cambio de datos de perfil. [Elaboración propia] 155

Figura 170, Pantalla de edición de perfil mostrando mensaje de error por documento ya registrado. [Elaboración propia] 156

Figura 171, Pantalla de edición del perfil con mensaje de éxito. [Elaboración propia] 156

Figura 172, Pantalla de edición del perfil con mensaje de éxito y regreso al inicio de sesión. [Elaboración propia] 157

Figura 173, Tercer panel con gráfica de tendencia de datos biométricos. [Elaboración propia] 157

Figura 174, Panel de tabla de registros médicos. [Elaboración propia] 158

Figura 175, Filtros de la tabla de registros médicos. [Elaboración propia] 158

Figura 176, Botón para la edición del spo2 y bpm. [Elaboración propia] 159

Figura 177, Botón para la eliminación de los registros biométricos. [Elaboración propia] 159

Figura 178, Confirmación para la eliminación permanente de las mediciones. [Elaboración propia] 160

Figura 179, Botón de contracción de la tabla de registros médicos. [Elaboración propia] 160

Figura 180, Panel inferior de tabla de alertas médicas. [Elaboración propia] 161

Figura 181, Filtros del panel inferior de la tabla de alertas médicas. [Elaboración propia] 161

Figura 182, Casilla para la eliminación de alertas. [Elaboración propia] 162

Figura 183, Confirmación para la eliminación permanente de las alertas. [Elaboración propia] 162

Figura 184, Botón de contracción de la tabla de alertas médicas. [Elaboración propia] 163

Figura 185, Switch de encendido o apagado del interruptor. [Elaboración propia] 163

Figura 186, Pulsador y mensaje de inicio. [Elaboración propia] 164

Figura 187, Teclado matricial y selección de modo. [Elaboración propia] 164

Figura 188, Mensaje de modo de bajo consumo y pulsador de inicio. [Elaboración propia] 165

Figura 189, Mensaje de inicio del modo offline. [Elaboración propia] 165

Figura 190, Calibración de datos biométricos. [Elaboración propia] 166

Figura 191, Medición de datos biométricos. [Elaboración propia] 166

Figura 192, Opciones de selección durante la medición de datos. [Elaboración propia] 167

Figura 193, Mensaje de detección del dedo. [Elaboración propia] 167

Figura 194, Mensaje de reinicio de mediciones. [Elaboración propia] 168

Figura 195, Mensaje de entrada a modo de bajo consumo. [Elaboración propia] 168

Figura 196, Opciones de reanudación y menú principal.[Elaboración propia] 169

Figura 197, Pantalla de Configuración de Wi-Fi. [Elaboración propia] 170

Figura 198, Modo de bajo consumo por inactividad. [Elaboración propia] 170

Figura 199, Confirmación de conexión Wi-Fi. [Elaboración propia] 171

Figura 200, Solicitud y escritura de la identificación. [Elaboración propia] 171

Figura 201, Reinicio por inactividad en ingreso de ID. [Elaboración propia] 172

Figura 202, Mensaje de ID no registrado y URL para realizar el registro. [Elaboración propia] 172

Figura 203, Inicio de medición y transmisión de datos. [Elaboración propia] 173

Figura 204, Opciones de selección durante la medición de datos. [Elaboración propia] 174

Figura 205, Mensaje de no detección del dedo. [Elaboración propia] 174

Figura 206, Mensaje de reinicio de mediciones. [Elaboración propia] 175

Figura 207, Mensaje de entrada a modo de bajo consumo. [Elaboración propia] 175

Figura 208, Opciones de reanudación y menú principal. [Elaboración propia] **176**
Figura 209, Protocolo de autoconexión Wifi. [Elaboración propia] **176**
Figura 210, Opciones de reanudación y menú principal. [Elaboración propia] **177**

ÍNDICE DE TABLAS

TABLA DE FIGURAS 6
ÍNDICE DE TABLAS 15
ÍNDICE DE ECUACIONES 16
1. PREGUNTA-PROBLEMÁTICA 17
2. JUSTIFICACIÓN 19
3. OBJETIVOS 20
4. MARCO TEÓRICO 21
5. ESTADO DEL ARTE 38
6. METODOLOGÍA 42
7. DISEÑO E IMPLEMENTACIÓN DEL SISTEMA 44
Tabla 1, Asignación de Pines del ESP32 para Control y Adquisición. [Elaboración propia] 52
8. PROTOCOLO DE PRUEBA 98
Tabla 2, Distribución según sexo. [Elaboración propia] 99
Tabla 3, Distribución según rango de edad. [Elaboración propia] 99
Tabla 4, Dispersión según el fototipo. [Elaboración propia] 99
Tabla 5, Desviación estándar de SpO_2 (global y estable). [Elaboración propia] 102
Tabla 6, Desviación estándar de BPM (global y estable). [Elaboración propia] 103
Tabla 7, Sesgo promedio de SpO_2 (global y estable). [Elaboración propia] 105
Tabla 8, Sesgo promedio de BPM (global y estable). [Elaboración propia] 106
Tabla 9, Sesgo absoluto promedio de SpO_2 (global y estable). [Elaboración propia] 107
Tabla 10, Sesgo absoluto promedio de BPM (global y estable). [Elaboración propia] 108
9. PRESUPUESTO 110
Tabla 11, Recursos físicos. [Elaboración propia] 110
Tabla 12, Recursos Humanos. [Elaboración propia] 110
Tabla 13, Presupuesto total. [Elaboración propia] 111
10. CONCLUSIONES 112
11. TRABAJOS FUTUROS 114
12. BIBLIOGRAFÍA 114
13. WEBGRAFÍA 120
ANEXOS 122
Tabla 14, Mediciones experimentales del sujeto de prueba 15. [Elaboración propia] 220
Tabla 15. Mediciones experimentales del sujeto de prueba 14. [Elaboración propia] 222
Tabla 16, Mediciones experimentales del sujeto de prueba 13. [Elaboración propia] 224
Tabla 17, Mediciones experimentales del sujeto de prueba 12. [Elaboración propia] 226
Tabla 18, Mediciones experimentales del sujeto de prueba 11. [Elaboración propia] 228
Tabla 19, Mediciones experimentales del sujeto de prueba 10. [Elaboración propia] 230
Tabla 20, Mediciones experimentales del sujeto de prueba 9. [Elaboración propia] 232
Tabla 21, Mediciones experimentales del sujeto de prueba 8. [Elaboración propia] 234
Tabla 22, Mediciones experimentales del sujeto de prueba 7. [Elaboración propia] 236
Tabla 23, Mediciones experimentales del sujeto de prueba 6. [Elaboración propia] 238
Tabla 24, Mediciones experimentales del sujeto de prueba 5. [Elaboración propia] 240
Tabla 25, Mediciones experimentales del sujeto de prueba 4. [Elaboración propia] 242

Tabla 26, Mediciones experimentales del sujeto de prueba 3. [Elaboración propia] 244
Tabla 27, Mediciones experimentales del sujeto de prueba 2. [Elaboración propia] 246
Tabla 28, Mediciones experimentales del sujeto de prueba 1. [Elaboración propia] 248

ÍNDICE DE ECUACIONES

Ecuación 1, Conversión del intervalo de pulso a BPM. 59

Ecuación 2, Ecuación de Suavizado Exponencial. 60

Ecuación 3, Valor Cuadrático Medio del componente AC RMS. 61

Ecuación 4, Relación de proporciones para el cálculo del factor R. 61

Ecuación 5, Factor r cálculo para la factorización de la constante. 61

Ecuación 6. Proporción para el cálculo del factor de forma simplificada. 61

Ecuación 7, Calibración lineal para la saturación de oxígeno. 62

Ecuación 8, Calibración lineal final para la saturación de oxígeno. 63

1. PREGUNTA-PROBLEMÁTICA

En Colombia la telemedicina o medicina virtual se comenzó a implementar y explorar a partir del 2020, a causa de la pandemia que ocurrió en dicho año. Como estrategia preventiva se planteó el desafío de fortalecer estos mecanismos de respuesta, teniendo en cuenta la pandemia y la recomendación de promover la virtualidad en todos los escenarios de la vida, para reducir los riesgos frente al COVID-19, comprendiendo que se debía garantizar la continuidad de la atención para la población en aislamiento. “El Sistema General de Seguridad Social en Salud realizó 63,1 millones de atenciones telefónicas y virtuales entre marzo de 2020 y el 31 de marzo de 2021, de este total, 20,7 millones correspondieron a atenciones telefónicas, mientras que 42,4 millones fueron teleconsultas” [1]. Siendo estas acciones un primer paso para Colombia en la medicina virtual o telemedicina, pero es necesario que este tipo de medicina se siga ampliando para que las personas de poblaciones de regiones apartadas, rurales y rurales dispersas, o personas con limitaciones económicas o de movilidad puedan tener acceso a la salud.[2]

Por otro lado el cambio a la transición de la medicina virtual o digital, es un proceso complicado, ya que en Colombia es un campo en el cual falta mucho desarrollo e investigación, pero es un avance necesario ya que esto tiene una connotación estratégica e innovadora, con las cuales se buscan implementar tecnologías digitales que faciliten procesos en las telemedicina como: análisis de macrodatos, herramientas para intercambiar y almacenar datos, herramientas para captar datos a distancia y un continuo enlace de información que permita una asistencia más enfocada en cada paciente, estas acciones con nuevas tecnologías podrían permitir resultados sanitarios más precisos, debido a que también ayudarán a mejorar los diagnósticos clínicos, las decisiones y la atención de los médicos al poder tener datos reales en los cuales se pueden apoyar. Como un segundo enfoque de interés en el desarrollo de las tecnologías que permitan mejorar el rendimiento en las telemedicinas o medicina virtual, se busca encontrar un factor que facilite la cobertura sanitaria universal, debido a que con la implementación de un plan estratégico que se enfoque en estas tecnologías y que cumpla con algunos de los 17 objetivos del Desarrollo Sostenible [3] se puede garantizar que 1000 millones más de personas [4]. también se beneficien y puedan disfrutar de una mejor salud y bienestar que previamente quizás no hubieran podido obtener o que hubieran sido inasequibles sin el constante desarrollo en las tecnologías en la atención de la salud [5].

Centrando el desarrollo de estas tecnologías dentro de la medicina virtual, en un solo sector de la población Colombiana, es necesario fijarse en las personas de tercera edad por varias razones, principalmente, son ellos quienes utilizan con mayor frecuencia los servicios de teleconsultas dado los requerimientos frecuentes de asistencia propios de la edad adulta [6], la segunda es que son más vulnerables a contraer enfermedades, y con respecto al virus COVID-19 es letal para este sector de la población, ya que si bien todos tienen el riesgo de contraer este virus, las personas de tercera edad tienen mayor probabilidad de enfermar gravemente si se infectan, con los mayores de 80 años muriendo a una tasa de cinco veces mayor que la media [7], la tercera razón es que este grupo de personas representa un porcentaje significativo de la sociedad Colombiana siendo aproximadamente un 13,5% de la población total nacional y un 15,5% de la población total de Bogotá [8] como último es sabido que los hospitales son los lugares con más riesgo de contagio, por lo que para este sector de la población es mejor evitar visitas a los hospitales de no ser estrictamente necesario. Atendiendo a estos factores, se evidencia que una gran parte de la población colombiana necesita esta alternativa de la medicina virtual, y de este mismo modo será uno de los sectores que emplearán en su mayoría todos los nuevos sistemas de salud que se construyan a partir de las nuevas tecnologías.

También se puede evidenciar que los adultos mayores son las personas menos tolerantes a los cambios y a la adherencia a las nuevas tecnologías en comparación a los pacientes en edades tempranas, además, en aquellos adultos mayores que requieren de personal cuidador, la atención resulta limitada debido a que los datos requeridos no se pueden recolectar de manera óptima por la dificultad en el uso de los dispositivos [6] por ende se ha de considerar este factor en el desarrollo de las nuevas tecnologías. Por lo tanto, estas herramientas deben empezar a implementar interfaces amigables para los usuarios, para que su uso sea el más práctico y adecuado para cualquier persona de la población y que el impacto tecnológico en las personas mayores no afecte los resultados clínicos que estos instrumentos puedan brindar

Ahora, buscando una opción viable para implementar el desarrollo en un instrumento que tenga facilidad de uso y un costo relativamente bajo, aparece la opción de los oxímetros de pulso de uso doméstico, ésta es una opción atractiva para monitorear a los pacientes. Este dispositivo médico empleado como un método no invasivo que de manera indirecta mide el porcentaje de saturación de oxígeno, la frecuencia respiratoria y la frecuencia cardíaca transportado por la hemoglobina en la sangre de un paciente. También es globalmente aceptado como el estándar para detectar y monitorear la hipoxemia, un nivel de oxígeno en la sangre inferior a lo normal. La hipoxemia puede presentarse con condiciones que afectan principalmente a los pulmones, como neumonía, bronquiolitis, asma, distrés respiratorio, entre otras; pero también se presenta a través de enfermedades sistémicas tales como sepsis y trauma. Adicionalmente, en conjunto con un suministro apropiado, la pulsioximetría es necesaria para el uso eficiente y seguro del oxígeno, así como para monitorear la respuesta a la terapia con oxígeno por parte del paciente, normalmente estos tratamientos son empleados en su mayoría por personas de tercera edad. [9]

En consecuencia para explorar este campo es necesario plantear estrategias e implementar ideas que aporten al desarrollo de las tecnologías en el campo de la medicina virtual y teniendo en cuenta todos los factores ya mencionados se hace viable una opción que satisfaga estas problemáticas y con el propósito de este trabajo se plantea la siguiente cuestión: ¿Cómo implementar el diseño de una mejora de un pulsioxímetro, para que permita la medición y registro de los signos vitales como lo son la frecuencia cardiaca y la saturación de oxígeno en la sangre de manera que facilite la comunicación entre médico y paciente, empleando interfaces simples para el fácil manejo del instrumento en personas de la tercera edad, dentro del marco de desarrollo de las tecnologías enfocadas a la medicina virtual en Bogotá?.

2. JUSTIFICACIÓN

Es necesario abordar el tema de las tecnologías que apoyan la telemedicina, ya que en Colombia este es un campo el cual aún sigue estando en una etapa muy temprana de desarrollo, y estos tiempos son los indicados para comenzar investigaciones, proponer ideas y diferentes alternativas para poder ir mejorando poco a poco este nuevo modelo de salud, para que la atención virtual que se ha ido implementando ya hace un par de años, tenga respaldos y apoyos en instrumentos y tecnologías que beneficien su funcionamiento y desarrollo, y de la misma manera para el bien general de las persona de la población Colombiana.

Como base teórica, el proyecto en telemedicina tiene muchas opciones desde la ingeniería que pueden ser estudiadas y aplicadas para que las herramientas y los instrumentos puedan mejorar en estos nuevos sistemas de salud virtual, además que las comunicaciones seguirán desarrollándose y de la misma manera, todas estas herramientas tendrán posibilidades de seguir siendo analizadas y mejoradas, dado que su estudio será constante en el área de las telecomunicaciones.

Una de las razones para desarrollar una investigación y un proyecto alrededor de un pulsioxímetro es que este es un dispositivo de muy bajo costo de fabricación en comparación a otras herramientas como lo puede ser un tensiómetro o un electrocardiógrafo. Además, al enfocarlo a las personas de tercera edad se está apuntando a las casas de retiros para los adultos mayores, puesto que con un solo equipo clínico se pueden llevar los registros de cada una de las personas que allí se encuentren de manera ordenada y sencilla, así es que solo es necesario tener uno de estos equipos para apoyar y reforzar las atenciones virtuales de la telemedicina.

Como ya se ha mencionado anteriormente el poder desarrollar la telemedicina contribuye al crecimiento de la cobertura social, puesto las personas que se beneficiaran y tendrán acceso a la salud aumentaran considerablemente [5], de forma que personas de poblaciones de regiones apartadas, rurales y rurales dispersas, o personas con limitaciones económicas o de movilidad puedan tener acceso a los nuevos sistemas de salud [2].

La telemedicina es uno de los varios sistemas y organizaciones de salud que buscan darle un giro verde a la medicina, con un enfoque en los beneficios ambientales y la mejora del estilo de vida. En Estados Unidos, uno de los sistemas hospitalarios sin ánimo de lucro más grandes, analizó el impacto que tuvo su plataforma de salud durante la pandemia de coronavirus y descubrió que la atención virtual no solo reduce los viajes hacia y desde el consultorio del médico o el hospital, sino que también reduce el uso y las emisiones de gas, por consiguiente, reduciendo el impacto negativo en nuestro planeta. Además, todo ese tiempo que no se pasó en la carretera redujo la congestión del tráfico y resultó en 15.092 toneladas métricas de emisiones de dióxido de carbono que no salieron del tubo de escape del automóvil, lo que a su vez se traduce en aproximadamente 18.490 acres de bosque salvado [10].

La causa principal para desarrollar este proyecto y empezar un proceso de investigación alrededor de las herramientas o instrumentos clínicos para la telemedicina, nace de las experiencias propias al ver las grandes falencias que la medicina virtual sigue teniendo en Colombia, puesto que no se han hecho cambios significativos a nivel nacional desde que este sistema se comenzó a implementar, por lo cual es necesario dar pie a nuevas ideas, nuevos proyectos, alternativas que permitan que este modelo de salud progrese en el país, es una necesidad empezar a establecer las bases del conocimiento en esta área y un deber aportar y contribuir en la construcción de un mejor país.

3. OBJETIVOS

3.1. GENERAL

Implementar un prototipo de pulsioxímetro que permita realizar la medición, registro, transmisión y comunicación de los signos vitales como lo son la frecuencia cardiaca y la saturación de oxígeno en la sangre.

3.2. ESPECÍFICOS

- Establecer un sistema de base de datos que cuente con campos para registrar los dos tipos de signos vitales medidos, la fecha; día, mes y año, y la hora en la que estos parámetros se evaluaron.

- Crear una interfaz simple para la base de datos, que cuente con un inicio de sesión para usuarios tanto como para el paciente como para el médico.

- Diseñar e implementar un instrumento clínico como lo es el pulsioxímetro.

- Instaurar un sistema de comunicación entre este dispositivo y la base de datos para que los registros se hagan con base a los datos de las mediciones de esta herramienta.

- Evaluar el funcionamiento del prototipo a través de pruebas con adultos voluntarios.

4. MARCO TEÓRICO

4.1. SATURACIÓN DEL OXÍGENO EN LA SANGRE

La saturación de oxígeno en la sangre tiene como definición la cantidad de oxígeno presente en el torrente sanguíneo, es decir, cuando la sangre se bombea desde el corazón hacia el resto del cuerpo, de esta manera la sangre recorre en cierto punto los pulmones, y en esta etapa del proceso es donde las moléculas de oxígeno se combinan con las células rojas de la sangre (eritrocitos), esta cantidad de células rojas que están saturados con oxígeno, es a lo que llamamos saturación de oxígeno en la sangre. De esta manera para saber que una persona tiene este parámetro de forma saludable y normal su valor debe oscilar entre 95% y un 100% de saturación de oxígeno en la sangre [11].

4.2. PULSO CARDIACO

La frecuencia cardíaca, ritmo cardíaco, ciclo cardíaco o pulso, es el número de latidos cardíacos por minuto que produce el corazón. En un ciclo la sangre con oxígeno procedente de los pulmones llega al corazón y es bombeada al resto del cuerpo, este proceso ocurre en muy poco tiempo ya que un latido tarda menos de un segundo en producirse [12].

Cuando se contrae el ventrículo izquierdo, la sangre pasa a través de las arterias y venas de todo el cuerpo. Esta onda de sangre es el pulso. Durante el reposo, el corazón sólo necesita bombear de 4 a 6 l/m de sangre. Este volumen aumenta hasta 5 veces en el ejercicio. Normalmente, cada ventrículo bombea 70 ml de sangre en cada contracción [13].

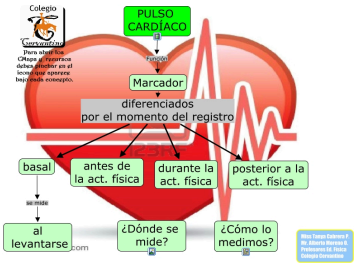


Figura 1, Diagrama pulso cardíaco. [40]

4.3. SENSORES

Los sensores son aquellos dispositivos que son sensibles a alguna forma de energía del ambiente, algunas de estas pueden ser luminoso, térmico, cinético, de forma que se relaciona información sobre una cantidad física de la cual se quiere obtener mediciones, como: temperatura, presión, velocidad, corriente, aceleración, posición, etc.

Un sensor no siempre tiene las características eléctricas necesarias para ser utilizado en un sistema de control, normalmente la señal de salida debe ser manipulada antes de su lectura, esto generalmente se logra con un circuito externo para producir un señal que puede ser leída por un controlador, en caso hipotético donde la salida de un sensor, cuando es sensibilizado por una energía externa, está dada por un nivel de tensión muy bajo, se hace necesaria su amplificación, por lo que este circuito externo necesariamente debe amplificar el nivel de la señal para su uso efectivo [14].



Figura 2, Estructura de tipos de sensores. [52]

4.3.1. FOTOPLETISMOGRAFÍA

La fotopletismografía (PPG) es un procedimiento no invasivo que se emplea para la detección de ondas cardiovasculares que se propagan alrededor del cuerpo humano, siendo consecuencia de las variaciones producidas en el flujo sanguíneo ocurridas en cada contracción cardiaca, este procedimiento tiene como base el irradiar un tejido con un diodo LED, normalmente se emplean diodos LED con longitudes de onda nominales de 660 nm y 940 nm, después de esto con un receptor se mide la intensidad lumínica reflejada, este receptor normalmente es un fotodetector selectivo, la señal recibida se simplificará y convertirá en voltaje para poder tratarla e interpretarla [12].

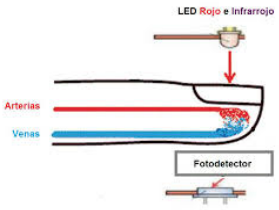


Figura 3, Esquema fotopletismografía. [39]

4.3.1.1. ESPECTROFOTOMETRÍA

La espectrofotometría es un método científico que permite realizar la medición de la cantidad de luz absorbida por una sustancia química, para realizar esto se mide la intensidad de la luz cuando un cuerpo luminoso pasa a través de dicha solución.

Esta medición se puede utilizar para conocer la cantidad de un elemento químico que compone la sustancia sometida a este método, en el caso de la sangre existen dos sustancias relevantes cuando se aplica este método, la hemoglobina y la oxihemoglobina, las cuales permitirán conocer el nivel de oxigenación, para poder medir estas dos sustancias es necesario usar dos longitudes de onda: una roja 660 nm y otra infrarroja 940 nm, la luz de estos dos haces atravesará la sangre oxigenada, la reducida y otros tejidos como huesos, uñas y piel [12].

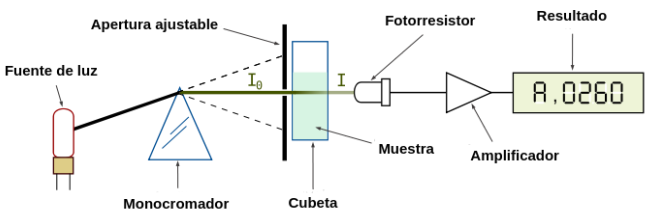


Figura 4, Esquema general sistemas de espectrometría. [51]

4.3.1.1.1. MAX 30100

El sensor MAX 30100, es un sensor que permite realizar la medición tanto como del ritmo cardíaco como de la saturación de oxígeno en la sangre, es un método no invasivo que normalmente se emplea en la pulsioximetría, este sensor captura dichos signos vitales con un circuito fotoeléctrico, de manera que el dispositivo integra dos emisores de luz, un led de luz roja y un led de luz infrarroja, de esta manera mide la cantidad de luz que se refleja a través del dedo, el sensor tiene integrado un pulsioxímetro y un monitor de frecuencia cardiaca, a su vez posee dos diodos Leds (led rojo y led infrarrojo), un fotodetector óptico especializado, filtro de luz ambiental entre 50 y 60Hz, y un conversor ADC [15].

Este tipo de sensores requieren dos voltajes para su funcionamiento: 1.8V y 3.3V, por lo cual incluye ambos reguladores de voltaje en su placa, además de que su consumo de corriente es mínimo [16].

4.3.1.1.2. MAX 30102

El circuito integrado MAX 30102, al igual que el circuito MAX 30100 puede medir la frecuencia del pulso cardiaco a la par de la saturación del oxígeno en la sangre, haciendo uso de dos diodos leds, un led rojo con frecuencia de 660 nm y un led infrarrojo con frecuencia de 920 nm, un fotodetector, óptica especializada, filtro de luz ambiental entre 50 y 60Hz, y un conversor ADC delta sigma de 16 bits y de hasta 1000 muestras por segundo en las cuales se mide la cantidad de luz reflejada [17].

Este circuito también opera con dos fuentes de alimentación a 1,8V y 3,3V, y adicionalmente cuenta con un módulo de comunicaciones que funciona mediante I2C [12].

4.3.1.1.3. MAX 30105

El sensor de partículas MAX 30105, es un sensor flexible y potente que permite detectar la distancia, la frecuencia cardíaca, la detección de partículas e incluso el parpadeo de un ojo, este circuito está equipado con tres Leds y un detector de fotones muy sensible, el funcionamiento de este sensor es muy similar al de los otros dos módulos revisados anteriormente, se refracta la luz a través del paciente y se medirá esa misma luz, dependiendo de lo que se detecte es posible tener diferentes tipos de partículas o materiales (como sangre oxigenada o humo de un incendio), el circuito cuenta con un LED rojo, un LED verde y un LED IR (infrarrojo) para detección de presencia, trazado de latidos cardíacos y monitorización de la frecuencia cardíaca entre una multitud de usos, incluida la oximetría de pulso, de igual manera este módulo está diseñado para trabajar a 5V y tiene la opción para comunicarse con microcontroladores de 3.3V y 5V[13].

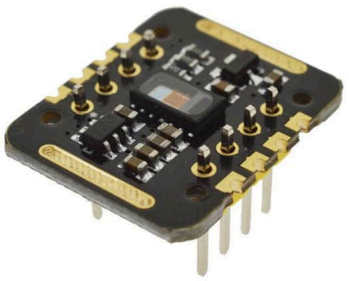


Figura 5, Sensor MAX 30102. [50]

4.4. MICROCONTROLADORES

Las tarjetas o placas de desarrollo, son útiles para el desarrollo de aplicaciones y sistemas embebidos, normalmente son empleadas como una herramienta de ayuda en el proceso del aprendizaje, debido a su amplia capacidad de complementos que permiten interactuar, explorar y crear nuevos proyectos, generando así nuevos conocimientos en los estudiantes e individuos que quieran profundizar en el estudio de los MCU.

De manera que el principal objetivo de las tarjetas de MCU es el de ayudar a estudiantes y personas que deseen incursionar en el proceso de aprendizaje, ya que contiene un hardware diseñado acorde a las necesidades del usuario, para que la persona que esté empleando esta herramienta se centre únicamente en la construcción del software sin tener que preocuparse en las conexiones de hardware y los periféricos que se desee utilizar[18]

4.4.1. ARDUINO

Arduino es un controlador que se basa en software y hardware fáciles de implementar, es de código abierto o open source, el hecho de ser una plataforma de código libre lo hace destacar frente a otras opciones comerciales, ya que los planos de los módulos están publicados bajo sus respectivas licencias, además de esto también ofrece la facilidad de ejecutarse en sistemas operativos varios, como lo son Windows, Linux o Macintosh OSX [19].

Mediante Arduino se puede programar y desarrollar elementos autónomos o puede ser empleado simplemente como captador de información.

Arduino ofrece una gran diversidad en los modelos de sus placas de desarrollo, algunas de los más conocidos como:

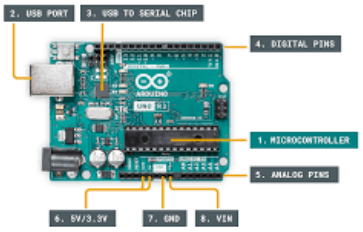


Figura 6, Dispositivo MCU ARDUINO. [49]

4.4.1.1. ARDUINO UNO

Se trata el primer modelo y más extendido, por ello la mayoría de placas restantes posee las características de este modelo, en cuanto a memoria es uno de los modelos más limitados, aunque resulta suficiente para muchos tipos de proyectos, consta de 14 Pines digitales de los cuales 6 pines pueden usarse como pwm y hasta seis pines analógicos [19].

4.4.1.2. ARDUINO MEGA

Se trata el modelo más potente Arduino, aunque es posible que tenga que sacrificar algo espacio, de todos los modelos es el que cuenta con más RAM, respecto a las propiedades eléctricas es muy similar a la placa Arduino uno, aunque posee 54 Pines digitales de los cuales 15 de ellos pueden ser pwm y 16 Pines analógicos [19].

4.4.1.3. ARDUINO NANO

Es una placa de pequeñas dimensiones tan solo mide 18.5 mm x 43.2 mm a pesar de su tamaño reducido se trata de un modelo muy completo, con conexión mini USB pero sin conector de alimentación externa, fue creado para aplicaciones de coste reducido y dónde importe el tamaño de la placa, sus propiedades eléctricas son como las de la placa Arduino uno con la diferencia que el modelo nano incluye dos Pines analógicos más [19].

4.4.1.4. ARDUINO PRO

Este modelo de placa fue creado para usuarios que buscan flexibilidad y precio bajo, no se trata de una placa más potente de Arduino ya que las características son muy similares a las de Arduino uno y tiene exactamente el mismo número de pines de esta placa, está destinada a la instalación semipermanente en objetos ya que la placa viene sin encabezados preinstalados y por ello se permite el uso de diferentes conectores o soldadura de cables [19].

4.4.2. PSOC

Las tarjetas y dispositivos Psoc son una serie de tarjetas creadas y desarrolladas por la empresa Cypress. El uso de este tipo de placa de desarrollo es muy utilizado en el acondicionamiento de señales, estos dispositivos cuentan con un conjunto de bloques digitales y analógicos programables, memorias y microcontroladores.[20]

La facilidad de programación de dichos bloques permite que el acondicionamiento de señales sea de manera fluida y sin problemas de adquisición, ya que emplean una arquitectura de sistema en chip altamente configurable para el diseño de control integrado.

Un solo dispositivo PSoC puede integrar hasta 100 funciones periféricas digitales y analógicas, lo que reduce el tiempo de diseño, el espacio en la placa, el consumo de energía y el costo del sistema, al tiempo que mejora la calidad del sistema. [21]

La arquitectura del PSoC 5LP mejora el rendimiento a través de:

- 32-bit ARM Cortex-M3 core plus, controlador DMA y procesador de filtro digital, hasta 80 MHz.
- Potencia ultra baja con el rango de voltaje más amplio de la industria.
- Los periféricos digitales y analógicos programables permiten funciones personalizadas.
- Enrutamiento flexible de cualquier función periférica analógica o digital a cualquier pin

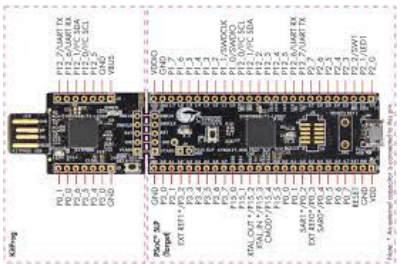


Figura 7, Dispositivo MCU Psoc. [48]

4.4.3. ESP32

La placa cuenta con un módulo ESP32-WROOM-32 de Espressif Systems, que combina memoria flash, un oscilador de cristal y una antena integrada en una PCB, así como un ESP32-SoC (System on a Chip). Una de las características que tiene esta placa es la disponibilidad de comunicación inalámbrica, ya sea vía WiFi o Bluetooth en la banda de 2.4 GHz. El ESP32 también cuenta con un CP2102 -Convertidor que permite establecer comunicaciones con las computadoras mediante el protocolo USB, lo que facilita la transferencia e intercambio de datos con diferentes programas. Alguna de las características técnicas del dispositivo son las siguientes[22]:

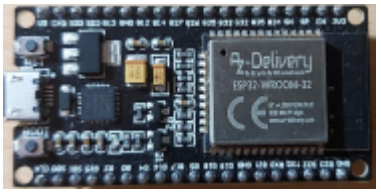


Figura 8, Dispositivo MCU ESP32. [22]

• Dimensiones: 56x28x13 mm.

• Voltaje de alimentación de 5V con mínima corriente de funcionamiento de 500 mA, ajustado a 3.3V

• Módulo de 38 pines, de los cuales 34 son I/O.

• Bajo consumo que permite alimentación mediante baterías.

• Memoria RAM de 512 kB.

• Memoria Flash externa de 4 MB.

• Conversor Analógico Digital (ADC) de 12 bits.

• Interfaces SPI, I2C, I2S, CAN y UART.

• Se puede programar con Arduino IDE, el framework de Espressif, Python, etc.

4.4.4. PIC

Los microcontroladores PIC de Microchip tienen las siguientes características

- Tienen una gran eficiencia del código.
- Funcionan a gran velocidad de ejecución.
- Posee una gran cantidad de herramientas de desarrollo de software y hardware y de bajo coste.
- Tiene compatibilidad de pines y código entre diferentes dispositivos de la misma familia.

Tiene una gran variedad de interrupciones que pueden ser empleadas para generar señales o para detectar cambios en las entradas.

Los microcontroladores se dividen en diferentes grupos según los bits utilizados para la memoria, que pueden ser 8, 16 y 32 bits.

Los microcontroladores de 8 bits son los más usados debido a su facilidad de uso, su precio bajo y su poco consumo de recursos. Los microcontroladores de 16 y 32 bits se utilizan en sistemas que utilizan redes y comunicaciones debido a que la mayoría de estos sistemas trabajan a 16 y 32 bits [23].



Figura 9, Dispositivo MCU Pic. [47]

4.4.4.1. MICROCONTROLADORES 8 BITS

Los microcontroladores de 8 bits de la gama media-mejorada y alta, son normalmente empleados en el ámbito del aprendizaje, el uso de este tipo de microcontroladores se debe a que son utilizados en la mayoría de las aplicaciones, son sencillos de programar y su precio es reducido.

Estos microcontroladores suelen incorporar una serie de módulos que permiten crear sistemas más complejos y con mayor funcionalidad [23].

4.4.4.2. MICROCONTROLADORES 16 BITS

Los microcontroladores de 16 bits de memoria se dividen en cuatro grupos diferentes, según las características de cada uno. pero a rasgos generales los cuatro grupos comparten las siguientes cualidades [23]:

- Los periféricos, software y herramientas de desarrollo son comunes para los cuatro grupos.
- La memoria de programa tiene una capacidad de 4KB a 256KB.
- Encapsulados de 14 a 100 pines.
- Utilizan arquitectura Harvard modificada con un bus de instrucciones de 24 bits.
- Ejecución de instrucciones en un solo ciclo.
- Respuesta a las interrupciones deterministas.

4.4.4.3. MICROCONTROLADORES 32 BITS

Los microcontroladores de 32 bits son dispositivos con un rendimiento muy elevado, dentro de esta gama de microcontroladores hay una gran variedad de tamaños de memoria, de velocidad de procesamiento y permiten conectar periféricos analógicos y digitales avanzados, también tienen opción de conectar CAN, CAN FD, Hi-Speed/FullSpeed USB y Ethernet [23]

4.4.5. RASPBERRY

Raspberry PI es una placa computadora (SBC) de bajo coste, en pocas palabras estas placas son ordenadores de pequeño tamaño, de dimensiones similares a una tarjeta, este sistema se desarrolló en el Reino Unido por la Fundación Raspberry PI (Universidad de Cambridge) en 2011, con el objetivo de estimular la enseñanza de la informática en las escuelas,

Está formada por una placa que soporta varios componentes necesarios en un ordenador común y es capaz de comportarse como tal. Este circuito se ha identificado como una maravilla en miniatura, que guarda en su interior un importante poder de cómputo en un tamaño muy reducido [24].

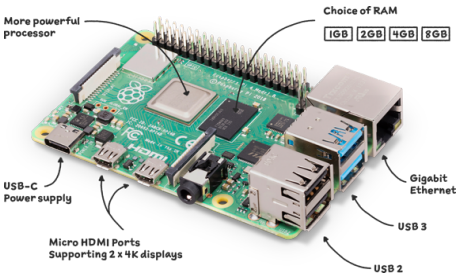


Figura 10, Dispositivo MCU RASPBERRY. [46]

4.4.5.1. RASPBERRY PI 3 MODELO A+

Los modelos A + se inclinan para aplicación con un menor precio. Cuenta con 512 MB de RAM compartidos con la GPU Video Core IV, un solo puerto USB y sin puerto de conexión de red por cable (RJ-45) [24].

4.4.5.2. RASPBERRY PI 3 MODELO B+

La Raspberry Pi 3 B+ cuenta con un nuevo procesador y mejor conectividad, así que pasa de tener 1.2Ghz a tener 1.4Ghz y en cuanto a la conectividad inalámbrica ahora incorpora doble banda a 2,4GHz y 5GHz, y su nuevo puerto Ethernet se triplica, pasa de 100 Mbits/s en el modelo anterior a 300 Mbits/s en el nuevo modelo, también cuenta con Bluetooth 4.2 (Low Energy) [24].

4.4.5.3. RASPBERRY PI 4 MODELO B+

La Raspberry Pi 4 B+ cambia los puertos HDMI de tamaño completo por dos puertos micro HDMI, también tiene la capacidad de manejar dos pantallas 4K a 60 Hz, ha diferencia de los modelos anteriores se ha incluido por primera vez USB 3.0, y el puerto Ethernet ya no está limitado a 300 Mbps, además de que cambia el procesador por un Broadcom nuevo hasta tres veces más eficiente que el anterior [24].

4.4.5.4. RASPBERRY PI ZERO

Este modelo tiene un tamaño mucho menor al de una Raspberry normal pero no tiene menor capacidad ya que es un 40% más potente que el primer modelo de Raspberry.

Tiene un microprocesador Broadcom BCM2835, que funciona a 1GHz con un solo núcleo. Posee 512MB de RAM, y comparte la gráfica Video Core IV, debido a su tamaño sustituye el puerto HDMI por Mini HDMI, manteniendo así las prestaciones, de la misma forma no emplea USB estándar, sino que tiene dos Micro USB, uno de alimentación y otro de datos, posee salida RCA, pero reemplaza los bornes por dos conectores integrados en la placa. Y cuenta con sistema de memoria práctico ya que su almacenamiento se emplea con memorias MicroSD [24].

4.5. SISTEMAS DE COMUNICACIÓN

4.5.1. REDES INALÁMBRICAS

Las redes inalámbricas se caracterizan principalmente por tener como canal de transmisión el aire, estas redes tienen ciertas ventajas sobre las redes físicas o cableadas, ya sea en cuanto a las técnicas o al apartado económico.

Una aplicación de estas ventajas en las redes inalámbricas, es el rango de movilidad del cual dispone el usuario, ya que estos pueden mantener un enlace con la red en una determinada zona, además de que estos sistemas tienen una banda determinada que frecuentemente no está regulada y es de uso libre, lo que les permite a los usuarios transmitir desde sus dispositivos en dicho rango.

Una de las principales ventajas que se deriva de la aplicación de redes inalámbricas, comparadas con redes cableadas, es que le permite al usuario un rango de movilidad; es decir los usuarios mantienen su conexión a la red por todo lo largo de una zona determinada.

Pero, así como presenta beneficios este tipo de redes presentan desventajas como lo son la calidad y la velocidad de la transmisión de la red, sin mencionar la vulnerabilidad en la seguridad de la información.

Las redes inalámbricas tienen fundamentos distintos ya sea por medios de transmisión por ejemplo microondas, ondas satelitales, ondas de radio, infrarrojo, y todas estas tecnologías son usadas actualmente. Normalmente su uso va directamente relacionado a la aplicación que se le desee dar a un sistema de comunicación. [25]



Figura 11, Esquema de redes inalámbricas. [41]

4.5.1.1. TIPOS DE REDES INALÁMBRICAS

Algunos de los beneficios que brindan las comunicaciones de tipo inalámbricas en sus diferentes presentaciones y tipos de redes a comparación de las redes físicas son las siguientes:

- Capacidad para un gran número de suscriptores
- Uso eficiente del espectro electromagnético debido a la utilización repetida de frecuencias Compatibilidad a nivel nacional e internacional, para que los usuarios móviles puedan utilizar sus mismos equipos en otros países o áreas
- Prestación de servicios para aplicaciones de datos, voz y video.
- Adaptación a la densidad de tráfico; dado que la densidad de tráfico es diferente en cada punto de la zona de cobertura.
- Calidad del servicio, en el caso de la voz, comparable a servicio telefónico tradicional y accesible al público en general. [26]

4.5.1.1.1. REDES WPAN

Esta red se define por sus siglas en inglés como Wireless Personal Área Network; este tipo de sistemas permiten la interconexión entre dispositivos de carácter portátil dentro de una zona determinada por el radio de cobertura, normalmente este radio tiene un rango de hasta 10 m, entre los dispositivos que comúnmente ocupan este tipo de redes tenemos: teléfonos inteligentes, laptops, tablets, impresoras, cámaras de fotos, etc., además poseen limitaciones en la velocidad de transmisión ya que alcanzan tan solo velocidades de 1 Mbps [25]

4.5.1.1.2. REDES WLAN

Esta red por sus siglas en inglés se define como Red de Área Local Inalámbrica (WLAN); es un tipo de sistema que permite la interconexión de diversos dispositivos como estaciones de trabajo, computadoras, impresoras, servidores, entre otros; estos dispositivos son capaces de comunicarse entre sí, por medio de unas direcciones IP asignadas a cada uno de ellos, ya sea fija o dinámica.

Este tipo de redes son muy utilizadas, ya que ofrecen ciertas ventajas de movilidad al usuario, debido a que permiten al usuario moverse dentro de un radio de cobertura establecido, sin perder conexión en la red [24],

4.5.1.1.3. REDES WMAN

Este tipo de redes se basa en el estándar IEEE 802.16, y su principal característica es que permiten dar cobertura a un área extensa; mediante el uso de esta tecnología, se puede por ejemplo realizar una interconexión entre redes que se encuentren alejadas. [25]

4.5.1.1.4. REDES WIDE AREA

Esta red se define por sus siglas en inglés (WWAN) como redes inalámbricas de área extendida, normalmente son empleadas para las comunicaciones en áreas geográficas extensas; la comunicación se establece mediante la utilización de antenas satelitales o antenas de microondas, permitiendo así el acceso a la red desde lugares remotos, por lo general muy lejanos. Su radio de cobertura es de miles de kilómetros, y sus velocidades de transmisión están en el rango de 56-170 kbps.

Estas redes emplean mayormente la tecnología UMTS, que se define por sus siglas como, Universal Mobile Telecommunications System, la cual es empleada en dispositivos móviles como celulares para redes 2G y 3G [25]

4.5.1.1.5. WIFI

La tecnología WIFI es la más aplicada en redes de área local inalámbricas, debido a que presentan ciertos beneficios de flexibilidad y adaptabilidad con respecto a las redes físicas de cables, comúnmente se utilizan en conjunto con redes cableadas para obtener una estructura más robusta para obtener un servicio más completo, esta tecnología provee a los usuarios de movilidad dentro de un radio de cobertura determinado por la capacidad del equipo instalado.

Disponen de una topología estructurada, donde intervienen dispositivos llamados puntos de acceso, los cuales permiten la conexión de cualquier tipo de dispositivos a redes existentes, o también permite la conexión entre redes, esto facilita a que se tengan radios de cobertura extensos, generalmente aplicados para instalaciones de gran tamaño, donde se requiere tener la misma red en todos los espacios, por facilidades de comunicación, almacenamiento y procesamiento de información

Estas tecnologías operan en bandas de frecuencia de libre operación, y son usadas principalmente para otorgar conexión a los dispositivos móviles de los usuarios en oficinas, hoteles, hospitales, aeropuertos, universidades, hogar [25].



Figura 12, Tecnología WIFI. [45]

4.5.1.1.6. BLUETOOTH

Bluetooth es una tecnología usada en las comunicaciones inalámbricas, su principal característica es generar comunicación entre dispositivos con el objetivo de poder transmitir voz y datos a través de un enlace de radiofrecuencia con un alcance no mayor a 10 m, entre dos dispositivos diferentes. [11]

El bluetooth en los Smartphones ha tenido un gran desarrollo, ya que es un dispositivo con alta presencia en el uso personal, se ha logrado conectar el celular a varios dispositivos como: reproductores de música, teclados, GPS, parlantes, etc. Y en general es una herramienta de gran uso y muy versátil. En la actualidad se ha logrado desarrollar esta tecnología hasta el punto de tener la versión "Bluetooth 5.0" mejorando su rango, la velocidad de transferencia de datos y su ancho de banda. [27]



Figura 13, Tecnología BLUETOOTH. [44]

4.5.1.1.7. ZIGBEE

Esta tecnología de comunicación se fundamenta en la utilización del estándar IEEE 802.15.4, surge con la idea de ser un protocolo normalizado para el funcionamiento adecuado de redes de área personal inalámbrica. Frecuentemente se centra en la interconexión de mecanismos electrónicos de pequeña señal o bajo consumo, por ello es mayormente utilizada en la comunicación de sensores en procesos industriales, en el campo de la medicina y la domótica [25].



Figura 14, Tecnología ZIGBEE. [43]

4.6. BASES DE DATOS

Las bases de datos son conjuntos de datos o información almacenada de forma estructurada. Diseñada para satisfacer las necesidades de manejo y almacenamiento de información por parte de organizaciones tales como empresas, hospitales, universidades, entre otras. Las bases de datos buscan combatir los problemas de duplicidad de información, inconsistencias y dificultades para realizar cambios a esta información, presente en los antiguos sistemas descentralizados de ficheros.

Las bases de datos modernas utilizan este modelo de repositorio único compartido dentro de la organización. Han evolucionado desde sistemas jerárquicos hasta los modernos sistemas objeto-relación. Cada avance ha mejorado la capacidad de modelar datos, ofreciendo soluciones más robustas y adaptadas a las necesidades modernas. [28]



Figura 15, Diagrama de red de base de datos. [42]

La administración de una base de datos se realiza a través de un Sistema de Gestión de Bases de Datos (SGBD), que es un software encargado de definir, crear, mantener y proporcionar acceso controlado a la base. Esto incluye funciones como la seguridad, integridad, control de concurrencia y recuperación ante fallos, lo que permite a múltiples usuarios interactuar con los datos de manera eficiente y segura.

4.6.1. MySQL

Este sistema de bases de datos es uno de los sistemas de gestión más populares para aplicaciones basadas en la web. Es gratuito y además está constantemente actualizado con nuevas características y medidas de seguridad, existen una variedad de ediciones de pago destinadas al uso comercial pero la versión gratuita ofrece más velocidad y seguridad al no llevar incluidas una gran cantidad de características y servicios, este último aspecto puede ser una característica positiva o negativa de acuerdo a la aplicación que le desee dar el usuario.

Este sistema permite seleccionar y manejar datos de una gran cantidad diferente de tipos de tablas, con una seguridad excelente y adicional a esto el sistema no desperdicia los recursos del servidor.

Sus principales ventajas son:

• Es gratuito.
• Ofrece una gran cantidad de características incluso en su versión gratuita.
• Existen multitud de interfaces de usuario y una gran comunidad detrás.
• Es compatible con otras bases de datos como Oracle.

Sus principales desventajas son:

• No existe soporte gratuito oficial para la versión gratuita.
• Puedes necesitar tiempo para automatizar funciones que otros sistemas llevan ya incluidos.

Ideal para desarrolladores independientes que necesitan un sistema robusto y un presupuesto muy limitado [28]

4.6.2. MICROSOFT SQL SERVER

Es un sistema de bases de datos, normalmente de los más empleados en los sistemas de gestión, el sistema permite la selección entre multitud de versiones de este software, puede funcionar tanto de forma local como en servidores basados en la nube e inclusive de ambas formas a la vez. Está también disponible en sistemas Linux desde el lanzamiento de la versión de Microsoft SQL Server 2016. Algunas de las características que incluye este sistema son: el soporte de la información temporal que hace posible rastrear los cambios que se han efectuado a los datos a lo largo del tiempo, la última versión de Microsoft SQL Server también incluye medidas de seguridad como los permisos de usuario, donde solo determinados usuarios con privilegios podrán acceder a la información más importante.

Sus principales ventajas son:

• Es rápido y estable.
• Ofrece la posibilidad de ver gráficas y datos del rendimiento de los recursos utilizados.
• Incluye soporte para los dispositivos móviles.
• Es totalmente compatible con los paquetes y programas de Microsoft.

Sus principales desventajas son:

• El precio puede ser elevado para muchas organizaciones.
• Puede llegar a consumir demasiados recursos, necesitando máquinas muy potentes.
• Muchos usuarios han notificado problemas a la hora de hacer tareas de mantenimiento.

Ideal para organizaciones grandes que utilizan muchos productos de la compañía Microsoft.[29]

4.6.3. ORACLE

Oracle probablemente sea el sistema gestor de bases de datos más utilizado, sus versiones más recientes permiten la gestión de bases de datos en servidores múltiples permitiendo fácilmente sostener miles de millones de datos al mismo tiempo.

Sus principales ventajas son:

• Innovación, generalmente Oracle va un paso más allá de todos los demás sistemas.
• Es increíblemente seguro, robusto y posee una gigantesca lista de características disponibles.

Sus principales desventajas son:

• Tiene un coste prohibitivo para pequeñas empresas y usuarios individuales.
• Este sistema requiere grandes cantidades de recursos para funcionar correctamente.

Ideal para organizaciones muy grandes que necesitan manejar gigantescas bases de datos y una gran cantidad de características. [29]

4.6.4. POSTGRESQL

Comúnmente usado para las bases de datos en sitios web, fue uno de los primeros sistemas de gestión de bases de datos en ser desarrollado y permite a los usuarios gestionar tanto datos estructurados como no estructurados, puede ser empleado en la mayoría de plataformas incluyendo los sistemas basados en Linux y, además, además tiene una interfaz práctica para la importación de bases de datos de otros sistemas, también puede ser utilizado en sistemas basados en la nube y en forma local.

Sus principales ventajas son:

• Puede soportar terabytes de información.
• Soporta JSON.
• Incluye una gran cantidad de funciones predefinidas.
• Tiene multitud de interfaces disponibles.

Las principales desventajas son:

• La documentación e información de este sistema no está muy clara por lo que para ejecutar funciones avanzadas el usuario promedio puede tener grandes dificultades.
• La configuración puede hacerse tediosa y complicada.
• No es muy veloz a la hora de hacer consultas muy grandes.

Es ideal para organizaciones con un presupuesto limitado que necesitan interfaces de usuario simples y usar JSON[29]

4.6.5. DB2

Este sistema de gestión fue creado por la compañía IBM, no utiliza las funciones SQL y puede leer los archivos XML y JSON. Fue diseñado para funcionar en los servidores iSeries de IBM, pero existen versiones que funcionan en Windows, Linux y Unix, sus versiones más actuales incluyen una larga lista de nuevas características, pero sobre todo destaca la mejora en su velocidad y en el uso de las memorias.

Sus principales ventajas son:

• Uso eficaz de los recursos del sistema.
• Puede correr varias tareas al mismo tiempo.
• Los códigos de error son muy descriptivos y permiten ahorrar tiempo a la hora de solucionar problemas.

Sus principales desventajas son:

• Alto coste monetario.
• Se requiere de software de terceros para ejecutar algunas funciones.
• El soporte básico solo está disponible durante tres años, después, hay que pagar.

Es ideal para organizaciones grandes que no tienen muchos recursos disponibles y necesitan manejar grandes bases de datos. [29]

4.7. TECLADO MATRICIAL

Un teclado matricial es un dispositivo que organiza un arreglo de múltiples pulsadores en disposición de filas y columnas, formando una matriz. Esta configuración permite reducir significativamente el número de conductores necesarios para su funcionamiento, en comparación con la conexión individual de cada pulsador. Estos dispositivos son utilizados ampliamente en la electrónica y la informática, siendo como ejemplos los teclados de nuestros computadores, dispositivos móviles e incluso nuestros mandos remotos. Su diseño facilita la interacción con microcontroladores en aplicaciones como introducción de datos, control de dispositivos, selección de opciones en interfaces de usuario e infinidad de utilidades. Dentro de sus beneficios, también encontramos su reducido tamaño añadiendo portabilidad a los dispositivos, un bajo costo y simplicidad.[50]

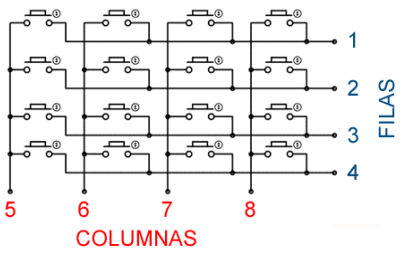


Figura 16, Arreglo de pulsadores de un teclado matricial. [53]

4.8. DISPLAY

4.8.1. DISPLAY 7 SEGMENTOS

Los displays de 7 segmentos son una herramienta clásica y versátil ampliamente usada para mostrar números y algunos caracteres alfanuméricos. Como su nombre lo dice, están compuestos por siete LEDs individuales dispuestos de forma de segmentos, que permiten formar los diferentes números y letras según la combinación encendida.
La versatilidad de estos dispositivos también radica en su capacidad para representar caracteres mediante códigos binarios, decimales o incluso hexadecimales, que especifican cuáles segmentos deben encenderse. Esto los convierte en una solución ideal para aplicaciones que van desde relojes digitales y medidores industriales hasta proyectos educativos y prototipos electrónicos. Su simplicidad, combinada con su eficiencia, ha mantenido a los displays de 7 segmentos como una opción popular en el diseño de sistemas electrónicos. [35]

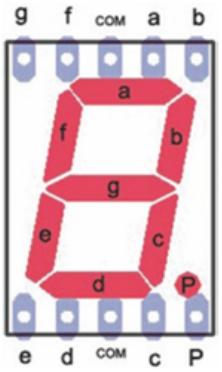


Figura 17, Display 7 segmentos. [35]

4.8.2. LCD

Las pantallas de cristal líquido, conocidas también como LCD, son una herramienta práctica y versátil para los proyectos electrónicos, pues facilitan la visualización de datos sin recurrir a los monitores o computadoras, haciendo los proyectos autónomos y portátiles. Se encuentran de variedad de tamaños según su número de filas y columnas, tales como el 16x2, 16x4, 20x2 y 20x4. Cuentan además, con un controlador HD44780 que organiza los caracteres a través de una matriz de 8x5 píxeles.



Figura 18, Pantalla LCD 16x2 con módulo I2C. [56]

La comunicación entre el microcontrolador y la pantalla se realiza mediante la conexión digital tradicional o el eficiente I2C, el cual reduce la cantidad de pines necesarios para su funcionamiento, solamente usa dos líneas para energizar el dispositivo y dos líneas de señal: datos en serie (SDA) y reloj en serie (SCL). El microcontrolador se comunica con todos los dispositivos y el mensaje incluye la dirección del dispositivo con el que se va a comunicar, de modo que solo el dispositivo correspondiente responde al microcontrolador. [35]

4.8.3. TFT

Las pantallas TFT (Thin-Film Transistor) representan una tecnología moderna de visualización con una gran flexibilidad y movilidad. Estas pantallas se comunican con Arduino mediante el protocolo SPI (Serial Peripheral Interface) o el protocolo I2C (Inter-Integrated Circuit), lo que permite una transmisión eficiente de datos. Su versatilidad les permite mostrar diversos elementos como texto, formas geométricas, imágenes y gráficos. Una característica notable es su capacidad para representar colores mediante códigos hexadecimales, solo en algunos modelos, permitiendo una amplia gama de opciones visuales. Se encuentran de diversos tamaños y resoluciones, como ejemplo la pantalla ST7735 1.8 pulgadas con una resolución de 160x128 píxeles y a color o la ILI9163 de 1.44 pulgadas con resolución de 128x128 monocromático. [35]



Figura 19, Pantalla TFT ILI9163. [55]

4.8.4. OLED

Las pantallas OLED (Organic Light-Emitting Diode) utilizan una película orgánica basada en carbono que emite luz cuando recibe corriente eléctrica. A diferencia de otras tecnologías, no requieren retroiluminación y son dispositivos de muy bajo consumo energético. Su uso es común en dispositivos como teléfonos móviles, cámaras digitales y laptops. En el ámbito de Arduino, se trabaja comúnmente con displays de 128x32 o 128x64 píxeles basados en el chip SSD1306 o en el chip SSH1106. Su lógica opera a 3.3V y permiten mostrar texto en diferentes tamaños, adaptándose a las necesidades del proyecto, con la particularidad de que cada carácter y espaciado requiere aproximadamente 6x8 píxeles. [35]



Figura 20, Pantalla OLED 128x64 basado en SSD1306. [54]

4.9. SERVIDORES WEB

Los servidores web desempeñan un papel fundamental en Internet, ya que son responsables de entregar las páginas a los usuarios. Funcionan como sistemas que reciben solicitudes de diversos clientes, tanto locales como en línea, y gestionan archivos como texto, imágenes y videos, permitiendo su visualización a través de un navegador. Estos servidores operan continuamente para proporcionar el servicio de la "World Wide Web" (WWW).[36]

4.9.1. APACHE

Apache es uno de los servidores web más populares, aunque su uso ha disminuido con el tiempo. Su versatilidad radica en su compatibilidad multiplataforma y su estructura modular, lo que permite la

integración de distintos lenguajes de programación en el lado del servidor, como PHP, Python y Perl. Además, ofrece características avanzadas como la optimización de datos mediante compresión, almacenamiento en caché y el uso de URLs más accesibles.[37]

4.9.2. MICROSOFT IIS

Microsoft IIS, aunque en sus inicios tuvo una reputación negativa por problemas de seguridad, ha evolucionado significativamente. Sin embargo, ha ido perdiendo presencia en el mercado. Se trata de uno de los servidores más utilizados y dispone de una amplia variedad de módulos, aunque su mayor limitación es su exclusividad para sistemas Windows.[37]

4.9.3. GOOGLE WEB SERVER

Google Web Server (GWS) no proporciona demasiados detalles sobre su funcionamiento. Se cree que es una versión personalizada de Apache, optimizada para sus servicios. Se emplea en plataformas como Blogger y App Engine, destacando por su capacidad para manejar grandes volúmenes de tráfico de manera eficiente. A diferencia de otros servidores como Apache o Nginx, GWS es exclusivo para los servicios de Google y no está disponible para el público en general.[37]

4.9.4. NGINX

Nginx es un servidor web ligero y de alto rendimiento, compatible con múltiples sistemas operativos, incluidos Windows, Linux y macOS. Su arquitectura optimizada le permite manejar un gran número de conexiones simultáneas con un consumo reducido de recursos, lo que lo hace ideal para plataformas de alto tráfico. Su eficiencia en la gestión de solicitudes lo ha posicionado como una de las mejores alternativas a servidores tradicionales.[37]

4.9.5. LIGHTTPD

Lighttpd es un servidor web ligero y eficiente, diseñado para minimizar el consumo de memoria y CPU sin sacrificar el rendimiento. Su arquitectura optimizada le permite gestionar un alto número de conexiones simultáneas de manera estable y rápida. Gracias a estas características, es una opción popular para sitios web de gran demanda. Además, su compatibilidad con tecnologías como FastCGI y SCGI facilita la integración con lenguajes de programación como PHP y Python, permitiendo un procesamiento dinámico eficiente.[37]

4.10. PHPMYADMIN

PHPMyAdmin es una aplicación web diseñada para la administración de servidores y bases de datos MySQL. Es una herramienta gratuita y de código abierto que permite realizar operaciones como la creación de bases de datos, modificación de estructuras de tablas, inserción y eliminación de datos, y ejecución de consultas SQL. Su interfaz intuitiva facilita la gestión de bases de datos incluso para usuarios con conocimientos limitados de programación.

Además, phpMyAdmin es ampliamente utilizado debido a su capacidad para administrar múltiples servidores y su soporte para funciones avanzadas como procedimientos almacenados, disparadores y claves foráneas. Es una solución versátil y eficiente para la administración de bases de datos en entornos web.[38]

5. ESTADO DEL ARTE

5.1. IMPLEMENTACIÓN DEL PROTOTIPO DE UN SISTEMA DE MONITOREO DE SIGNOS VITALES: PULSO, TEMPERATURA Y SATURACIÓN DE OXÍGENO PARA PACIENTES [30]

En este proyecto se desarrolló un sistema de monitoreo de signos vitales como lo son pulso, temperatura y saturación de oxígeno en la sangre, el cual se basa en la comparación y selección de componentes por medio de la metodología de la investigación y su validación a través de un análisis estadístico.

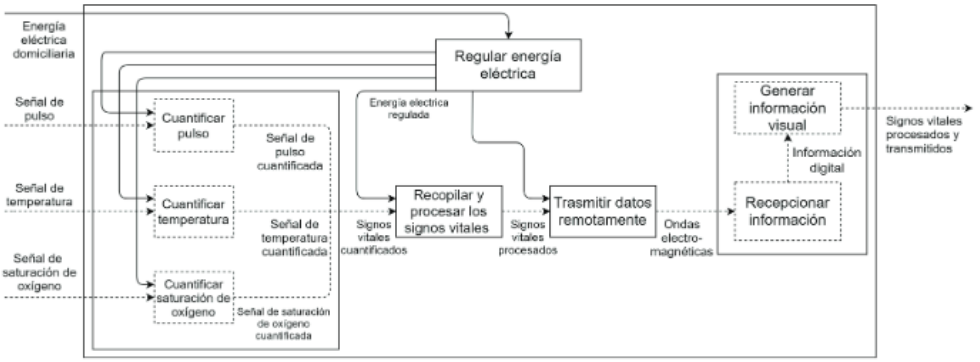


Figura 21, Estructura de funciones del sistema de monitoreo remoto. [30]

Para llevarlo a cabo se implementaron elementos como, el sensor MAX30102 que permite la adquisición del pulso cardíaco y la saturación del oxígeno en la sangre, ya que haciendo uso de un solo módulo de esta referencia se pueden medir los dos signos vitales, también se empleó el sensor MLX90014 en un módulo GY-900 para adquirir los valores de temperatura corporal, este sensor fue escogido por la razón de que tiene una alta precisión en el rango de trabajo de 30 a 40 °C, por otro lado se utilizó una placa Arduino Mega para la adquisición de los datos muestreados por los sensores, debido a su buena capacidad de memoria SRAM y a su alto grado de compatibilidad con los sensores en lo que respecta al interfaz de comunicación I2C, de la misma manera se seleccionó la SBC Raspberry Pi Zero W para la transmisión remota utilizando tecnologías de Cloud Computing e internet de las cosas, debido a que tiene la capacidad de trabajar con el estándar 802,11 en la banda de 2,4GHz y por último se empleó la aplicación y API web de internet de las cosas ThingSpeak, debido a su compatibilidad con la tarjeta de transmisión remota seleccionada, empleando el modelo de computación en la nube a manera de software como servicio (SaaS) y el protocolo de comunicación HTTP.

Los resultados que se obtuvieron en este proyecto se realizaron a través de la verificación de la correcta obtención de los datos de los signos vitales realizando un muestreo de los datos de acuerdo al juicio de expertos y validando a través de la implementación de la prueba “t” de Student para muestras pareadas.

Por último, es importante resaltar que el proyecto al tener las tecnologías de Cloud Computing e internet de las cosas puede llegar a integrar audio y video para recopilar más parámetros objetivos y subjetivos que proporciona el paciente al personal sanitario, además, cuenta con la capacidad de instalación de sistemas operativos que permitan la programación a distancia, como una ventaja en la coyuntura actual o similares, esto a manera de precedente en futuras investigaciones. [30]

5.2. DISEÑO DE UN SISTEMA DE INSTRUMENTACIÓN WEARABLE PARA LA MEDIDA DE LA SATURACIÓN DE OXÍGENO EN LA SANGRE [31]

En dicho trabajo se desarrolló tanto la parte de software como el diseño de hardware de un dispositivo médico encargado de calcular y estudiar la incógnita citada anteriormente, y que sea capaz de registrar sus mediciones, consiguiendo así una monitorización completa del sujeto

El sistema se encuentra conformado como se presenta en la Figura 1, funciona de tal manera que, el control PWM genera las señales de encendido del LED, es decir, a qué frecuencia se tomarán los datos y la amplitud que se tendrán, dicho control tiene una frecuencia de 500Hz y una amplitud de 5V, pasa a través de los drivers de LED que regularan los pulsos de los mismos, luego gracias a un fotodiodo, recibe toda la luz que no fue absorbida. Pasa a través de un amplificador de transimpedancia que la transforma en corriente, seguidamente llega al S&H que elimina cualquier variable que cause una mala filtración de señales durante la conversión, luego se filtra la señal, se amplifica, pasa por el ADC. Por último, de la salida del ADC, pasa por un microcontrolador para mostrar mediante un display, interfaz o lo que se programe, los datos que se establezcan.

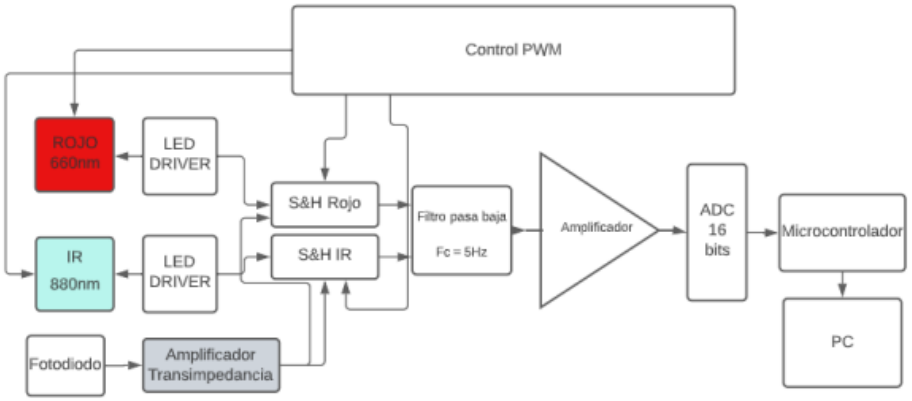


Figura 22, Diagrama de bloques del sistema. [31]

El resultado principal de este proyecto fue la visualización por medio de la interfaz gráfica de los valores de saturación de oxígeno en la sangre extraídos del sensor MAX30102 el cual fue implementado para elaborar un pulsioxímetro con un error aproximado de +/- 2.4 %. [31]

5.3. SISTEMA PARA LA MONITORIZACIÓN DE LA SATURACIÓN DE OXÍGENO EN SANGRE [32]

Este trabajo desarrolla la implantación y diseño de un prototipo electrónico basado en plataformas de hardware libre para el registro remoto de datos biomédicos de pacientes y su posterior envío a un servidor web para que se pueda analizar y representar los datos de los mismos de una manera gráfica.

El envío y recepción de los datos se lleva a cabo mediante un microcontrolador, y se envía de forma asíncrona a un servidor mediante Wifi.

Es en este servidor es donde se realizará un monitoreo de cada usuario registrado en la aplicación y su posterior almacenamiento en la base de datos.

Desde la aplicación web basada en Java se podrá acceder a cada información única para cada uno de los usuarios, se podrá visualizar los datos obtenidos por el sensor de manera alfanumérica o gráfica, así como vincular de manera única cada usuario con el hardware correspondiente

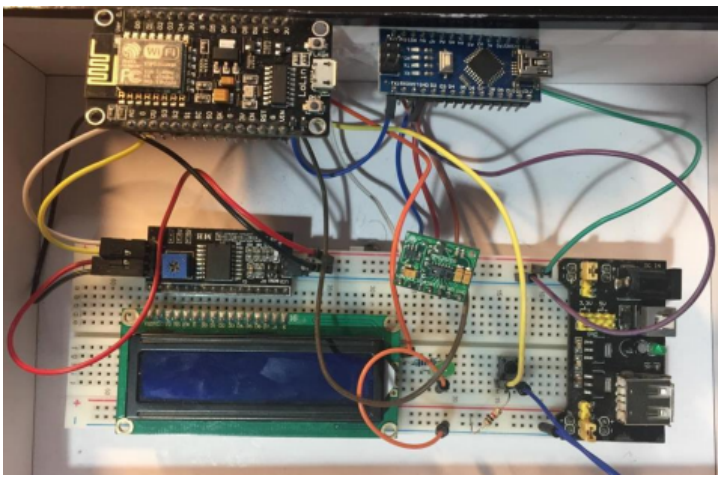


Figura 23, Diagrama de bloques del sistema. [32]

El prototipo desarrollado para este trabajo y el pulsioxímetro real se encuentra en un rango inferior al 3% de margen de error y adicional a esto dispone de capacidades de monitorización y procesado de datos.c

5.4. DISEÑO Y DESARROLLO DE UN DISPOSITIVO ELECTRÓNICO DE USO PERSONAL PARA LA MONITORIZACIÓN CONTINUADA DE LA SATURACIÓN DE OXÍGENO EN SANGRE [33]

En este proyecto se realizó el diseño de un sistema para la monitorización continuada de la saturación de oxígeno en sangre utilizando un sensor pulsioxímetro

La Figura 2 muestra el diagrama de bloques de las diferentes etapas que forman el sistema completo y el orden en el que se desarrolló este proyecto para el análisis y diseño de cada uno de los bloques.

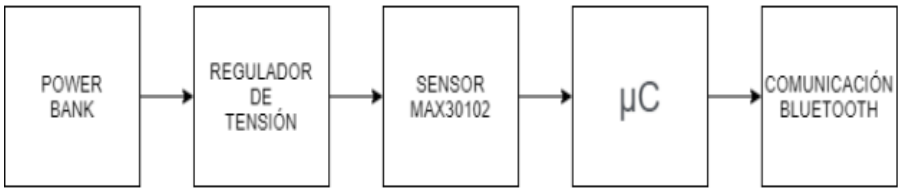


Figura 24, Diagrama de bloques del sistema completo. [33]

La primera etapa está formada por una Power Bank que carga una batería de 3.6V y eleva su tensión a 5V en su salida, a continuación, se regula la tensión de 5V a 3.3 y 1.8V utilizando dos reguladores de tensión. La etapa del sensor MAX30102 está formada por el propio sensor y el diseño de las resistencias de pull-up utilizadas para la comunicación con el µC. La penúltima etapa es el análisis del microcontrolador utilizado y la última etapa es el análisis y conexionado del módulo bluetooth del sistema a la aplicación en el teléfono móvil Android de cada usuario.

El dispositivo final tiene un diseño portátil y de fácil uso para cualquier persona, es un dispositivo autónomo debido a que cuenta con una carga de batería interna de 3.6 V utilizando un cargador de teléfono móvil común, presenta la inteligencia necesaria para el cálculo de la saturación de oxígeno en sangre y está dotado de un módulo bluetooth para enviar al móvil los datos registrados por el sensor. Se han utilizado componentes SMD normalizados de resistencias y condensadores para obtener la máxima miniaturización posible. [33]

5.5. DISPOSITIVO TELEMÉTRICO PARA MONITOREO DE FRECUENCIA CARDIACA Y SATURACIÓN DE OXÍGENO [34]

En este proyecto se realizó un sistema de telemetría para monitoreo de frecuencia cardíaca y saturación de oxígeno que permite almacenar los datos en una memoria SD y transmitirlos para su análisis a través de una interfaz desarrollada en LabVIEW.

El sensor empleado en este proyecto tiene un inconveniente ya que, la señal que entrega el fototransistor en conjunto con el led infrarrojo es muy tenue, por lo que, se necesitó un circuito encargado de amplificar la señal y posteriormente filtrar, la señal se procesa por la tarjeta de desarrollo Arduino UNO nano. Finalmente, los datos se envían por el módulo de RF de 915 MHz. Cuando el módulo de RF recibe los datos, los envía a una computadora a través de un cable USB y se presentan los datos en una interfaz gráfica programada en LabVIEW.



Figura 25, Diagrama de bloques del sistema completo. [34]

Las pruebas realizadas demuestran que los signos obtenidos por el sistema propuesto se asemejan a los signos obtenidos por un producto comercial y cuenta con un rango de alcance en la comunicación de aproximadamente 50 m, a esta distancia todavía se reciben los datos constantemente y sin interferencias [34].

6. METODOLOGÍA

6.1. FASE 1

Establecer un sistema de base de datos que cuente con campos para registrar los tres tipos de signos vitales medidos, la fecha; día, mes y año, y la hora en la que estos parámetros se evaluaron.

- Investigar los tipos de bases de datos existentes.
- Comparar las ventajas, beneficios y problemas que puedan tener las bases de datos indagadas.
- Elegir la opción que se acomode de mejor manera para la implementación del proyecto.
- Diseñar y elaborar una base de datos que cumpla con los requerimientos deseados.

6.2. FASE 2

Crear una interfaz simple para la base de datos, que cuente con un inicio de sesión para usuarios tanto como para el paciente como para el médico.

- Revisar los diferentes tipos de lenguaje de programación se pueden utilizar para implementar interfaces gráficas.
- Seleccionar el lenguaje de programación que presente mayor comodidad de implementación y una mejor comunicación con la base de datos.
- Diseñar una plantilla para una interfaz simple que cumpla con secciones para ingreso de usuarios, campos de información y ayuda para los usuarios.
- Enlazar la plantilla del interfaz a la base de datos para que los usuarios puedan acceder a la información desde un aplicativo.

6.3. FASE 3

Diseñar e implementar un instrumento clínico como lo es el pulsioxímetro.

- Examinar los tipos de sensores disponibles en el comercio para captar los datos requeridos en el instrumento.
- Comparar los sensores escogidos en cuanto a sus funciones y precios.
- Optar por el sensor más adecuado según los parámetros necesarios para implementar un pulsioxímetro de bajo costo.
- Investigar y comparar los diferentes tipos de microprocesadores para llevar a cabo la implementación del instrumento.
- Elegir un microprocesador que permita el desarrollo del instrumento clínico teniendo en cuenta el factor económico para una implementación de bajo costo.
- Diseñar un pulsioxímetro debidamente ajustado que se pueda implementar como un instrumento clínico.

6.4. FASE 4

Instaurar un sistema de comunicación entre este dispositivo y la base de datos para que los registros se hagan con base a los datos de las mediciones de esta herramienta.

- Indagar los sistemas de comunicación disponibles para hacer el respectivo proceso entre la base de datos y el microprocesador,
- Seleccionar uno de los sistemas de comunicación.
- Diseñar e implementar un sistema que permita la comunicación entre el microprocesador y la base de datos.

6.5. FASE 5

Evaluar el funcionamiento del prototipo a través de pruebas con adultos voluntarios.

- Desarrollar un encapsulado para tener un prototipo del dispositivo a evaluar.
- Realizar pruebas de medición y registros de los signos vitales en adultos, quienes participarán de manera voluntaria.
- Registrar y analizar los resultados obtenidos durante las mediciones.

6.6. FASE 6

Documentación y análisis.

- Registrar de forma escrita cada etapa y fase durante el desarrollo del proceso.
- Analizar y describir los resultados registrados que se tuvieron en cada momento durante la realización del proyecto.

7. DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

7.1. DISEÑO GENERAL DEL SISTEMA

El sistema consiste en un pulsioxímetro digital basado en un sensor óptico MAX30102 y un microcontrolador ESP32, diseñado para medir de forma no invasiva la saturación de oxígeno en la sangre (SpO2) y la frecuencia cardíaca (BPM). La arquitectura del sistema integra de forma distribuida las etapas de adquisición, procesamiento, almacenamiento y visualización de los datos, garantizando un flujo de información continuo, confiable y seguro.

La solución tecnológica se estructura en cuatro capas principales: adquisición y procesamiento, almacenamiento, lógica y servicios, y finalmente, presentación.

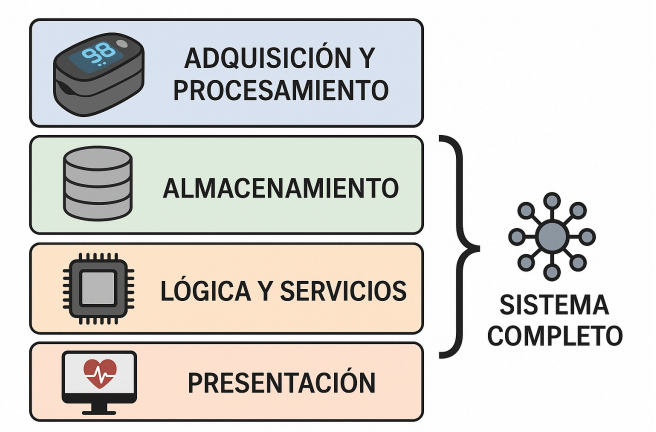


Figura 26, Arquitectura general del sistema en cuatro capas. [Elaboración propia]

7.1.1. CAPA DE ADQUISICIÓN Y PROCESAMIENTO

Esta capa se compone del hardware central del dispositivo. Incluye el microcontrolador ESP32 y el sensor óptico MAX30102. El sensor emplea la técnica de fotopletismografía (PPG) para registrar la variación de la luz, transmitiendo los datos al ESP32 mediante el protocolo I2C.

El ESP32 interpreta estas señales, aplicando algoritmos específicos para calcular los valores de SpO2 y BPM, y gestiona la visualización local en una pantalla integrada. El sistema también incluye un teclado matricial que permite al usuario ingresar su número de documento (cédula), el cual se emplea como identificador único para la correcta asociación de las mediciones al paciente. El ESP32 utiliza su conectividad Wi−Fi integrada para la transmisión inalámbrica de los datos procesados.

7.1.2. CAPA DE ALMACENAMIENTO

La segunda capa es la de persistencia de datos, gestionada por una solución de base de datos relacional en la nube . Esta capa es la autoridad responsable de almacenar tanto los perfiles de usuarios como los registros de mediciones enviados directamente por el dispositivo. Para asegurar la integridad referencial y la trazabilidad del dato clínico, esta capa valida que cada medición se asocie inequívocamente a una entidad de usuario preexistente.

Además de la persistencia, esta capa gestiona la validación inicial a nivel de datos y la generación de alertas automáticas. Mediante triggers implementados en la base de datos, una función ejecuta la

lógica para validar los valores recibidos y crear una alerta dedicada si las mediciones caen fuera de los umbrales clínicos preestablecidos.

7.1.3. CAPA DE LÓGICA Y SERVICIOS

La tercera capa es el Backend, un módulo de lógica central que actúa como el cerebro para la gestión, autenticación, autorización y consulta de datos. Este componente se despliega en un entorno de servidor en la nube y es responsable de validar las peticiones provenientes de la interfaz de usuario, procesar las consultas a la base de datos y aplicar las reglas de gestión del sistema, incluyendo el manejo de perfiles de usuario y sus permisos.

Para garantizar un acceso seguro, el Backend implementa mecanismos de autenticación y autorización basados en tokens de seguridad, asegurando que solo los usuarios con los permisos adecuados accedan a la información. Además, aplica filtros avanzados en sus puntos de acceso (*endpoints*) para optimizar la eficiencia de las consultas de datos históricos.

7.1.4. CAPA DE PRESENTACIÓN

La cuarta y última capa es el Frontend, una interfaz de usuario web que se despliega en una plataforma de alojamiento en la nube. Esta interfaz es responsable de traducir los datos recibidos del *backend* en representaciones visuales intuitivas y accesibles.

La interfaz contempla tres tipos de usuarios: pacientes (visualizan sus propias mediciones), médicos (acceso a todas las mediciones, con funciones de gestión básica), y administradores (con control total sobre el sistema). La visualización de los datos se realiza de dos formas: localmente, mediante una pantalla integrada en el dispositivo; y de forma remota, a través de la interfaz web, que permite consultar tanto el historial como las mediciones actuales. Esta estructura asegura que los datos se mantengan organizados, accesibles y protegidos según el nivel de permisos asignados.

7.2. ESPECIFICACIÓN TÉCNICA DEL SISTEMA

La especificación técnica del sistema es la sección que documenta y justifica las decisiones de diseño fundamentales que definen la estructura física y lógica de la solución. Aquí se detallan los componentes y tecnologías seleccionadas para cada capa de la arquitectura, demostrando cómo su elección garantiza la precisión, escalabilidad y seguridad operativa del sistema de monitoreo en la práctica.

7.2.1. SENSOR ÓPTICO MAX30102

El sensor MAX30102 fue seleccionado como el componente principal de adquisición de datos por su inigualable precisión y eficiencia energética en la medición simultánea de SpO2 y BPM mediante fotopletismografía. Esta elección se justifica por las siguientes ventajas sobre otras alternativas:

El MAX30102 integra LEDs rojo e infrarrojo con unas longitudes de onda de 660 nm y 880 nm respectivamente y un fotodiodo altamente sensible, lo que le permite captar con precisión la señal óptica reflejada en el tejido sanguíneo. Ofrece una precisión típica de ±2% en el rango de SpO2 y un consumo energético notablemente bajo de 600µA en modo activo, el cual es vital para prolongar la autonomía del dispositivo portátil alimentado por batería. Se evaluaron sensores de la misma familia, como el MAX30100, que se descartó por su menor precisión, mayor consumo y rango dinámico

limitado, factores que comprometen la calidad de las mediciones y la estabilidad de la señal. Asimismo, el MAX30105 se consideró inadecuado debido a que su enfoque principal está orientado a la detección ambiental, lo que implica un mayor costo y complejidad sin aportar beneficios directos a la medición de signos vitales requerida. El MAX30102 proporcionó el equilibrio óptimo en la relación costo-rendimiento, además de incluir filtros internos para la reducción de ruido y utilizar el protocolo I2C para una comunicación sencilla y robusta.

7.2.2. MICROCONTROLADOR ESP32

El ESP32 se eligió como el microcontrolador central del sistema debido a su potencia de procesamiento y su conectividad Wi−Fi integrada, requisitos indispensables para el monitoreo y transmisión de datos. Esta elección se tomó tras comparar sus capacidades con otras plataformas populares.

Microcontroladores de menor capacidad como la familia Arduino UNO o Arduino Nano fueron descartados por carecer de conectividad Wi−Fi y por tener una potencia de procesamiento limitada para ejecutar de manera eficiente el complejo algoritmo de procesamiento de señales ópticas y las tareas de comunicación de forma continua y prolongada.

El ESP32, con su doble núcleo Tensilica Xtensa LX6 que opera hasta a 240 MHz, resuelve estas deficiencias. Permite la ejecución eficiente de tareas concurrentes, dedicando un núcleo a la adquisición, filtrado y cálculo del algoritmo de SpO2, y el otro a la comunicación con la base de datos a través de Wi−Fi

Por otro lado, su conectividad Wi−Fi integrada simplifica significativamente el diseño al eliminar la necesidad de hardware adicional, reduciendo costos y complejidad. Finalmente, su arquitectura permite un bajo consumo energético, crucial para la portabilidad, y su amplia compatibilidad con entornos como Arduino IDE asegura un desarrollo ágil y facilita el mantenimiento y las futuras actualizaciones del sistema.

7.2.3. BASE DE DATOS SUPABASE (POSTGRESQL)

Para la capa de persistencia de datos, se seleccionó Supabase, que utiliza PostgreSQL como su sistema de gestión de base de datos relacional. Esta elección supera a otras soluciones relacionales y NoSQL debido a su confiabilidad operativa, capacidad de escalabilidad y habilidad para integrar lógica de backend. Se eligió PostgreSQL sobre otras bases de datos relacionales por ser un sistema con amplia trayectoria en la industria, de máxima fiabilidad transaccional y conocido por su probado rendimiento en ambientes de alta concurrencia. Su naturaleza relacional es crucial en un sistema médico, ya que facilita la aplicación de integridad referencial mediante llaves primarias y foráneas, asegurando que los datos de medición siempre estén asociados a un paciente existente, lo cual es un requisito de coherencia innegociable.

La plataforma Supabase optimiza esta base de datos para la arquitectura del sistema, ya que ofrece una solución nativa en la nube. Esto garantiza alta disponibilidad y capacidad para crecer junto con la cantidad de usuarios y datos de medición, superando las limitaciones de soluciones locales. Además de almacenar la información, Supabase permite implementar lógica programable directamente en la base de datos a través de triggers y funciones. Esta característica es fundamental, ya que permite que la base de datos ejecute automáticamente la validación de valores recibidos y la generación inmediata de alertas si las mediciones clínicas caen fuera de los umbrales predefinidos. En resumen, la

combinación de PostgreSQL con la infraestructura de Supabase proporciona la base más segura, escalable y lógicamente inteligente para gestionar la información médica.

7.2.4. LÓGICA DE SERVICIOS (BACKEND)

Para la capa de servicios, se eligió FastAPI, un framework moderno de Python, y se desplegó en la plataforma Render. Esta decisión se fundamenta en la necesidad de un sistema que ofrezca alto rendimiento, concurrencia eficiente y seguridad reforzada y estable para gestionar las reglas funcionales del sistema y la autenticación, con una infraestructura de despliegue confiable en la nube.

FastAPI se seleccionó por su diseño asíncrono, el cual le permite manejar peticiones concurrentes de manera más eficiente que otros frameworks de Python como Flask o Django. Es reconocido por ofrecer un rendimiento comparable al de Node.js en velocidad pura, lo cual es crucial para mantener la estabilidad del sistema bajo carga de usuarios y asegurar una respuesta rápida como gateway de seguridad y datos. El framework facilita la implementación de endpoints RESTful seguros y el uso de Pydantic, que proporciona validación estricta de datos en tiempo de ejecución. Esta característica es vital para la integridad de los datos médicos, ya que asegura que solo la información con el formato correcto pueda entrar al sistema. Además, FastAPI simplifica la implementación de la seguridad basada en tokens JWT y los mecanismos de autenticación y autorización, garantizando que el acceso a la información se gestione de forma segura y diferenciada según el perfil del usuario.

La elección de Render como plataforma de despliegue para el backend complementa las ventajas de FastAPI. Render ofrece un entorno gestionado que simplifica la operatividad y garantiza la disponibilidad continua de la API en la nube, liberando al equipo de la gestión de infraestructura. Su capacidad para escalar automáticamente en función de la demanda y su compatibilidad con los contenedores de Python aseguran que el backend pueda manejar picos de tráfico sin interrupciones. Además, el modelo de despliegue de Render permite una fácil integración con el control de versiones, facilitando la implementación rápida de actualizaciones y un mantenimiento eficiente.

7.2.5. INTERFAZ DE USUARIO (FRONTEND)

Para la capa de presentación, se seleccionó React.js, desplegado en la plataforma Vercel. Esta elección se fundamenta en la necesidad de ofrecer una interfaz de usuario dinámica, intuitiva y altamente responsiva, capaz de gestionar la complejidad de los datos médicos y adaptarse a los distintos perfiles de usuario, todo ello con una infraestructura de despliegue optimizada.

React.js fue elegido sobre otros frameworks de desarrollo como Angular o Vue.js por su enfoque en la estructura modular basada en componentes y la gestión eficiente del estado. Esto facilita la creación de una interfaz que se construye a partir de unidades de código reutilizables, lo que optimiza la escalabilidad y mantenibilidad del código a largo plazo. Su capacidad para crear una interfaz dinámica permite una experiencia de usuario fluida, esencial para la visualización de datos, la interacción con formularios de registro y autenticación. Además, la vasta comunidad de React y su ecosistema de librerías aceleran el desarrollo y la implementación de funcionalidades complejas, como la validación de campos en tiempo real.

La plataforma Vercel se seleccionó como entorno de despliegue para el frontend debido a sus beneficios operativos y de rendimiento. Vercel ofrece un sistema de entrega continua que automatiza el proceso de publicación, desde el código hasta la producción, facilitando la implementación rápida de actualizaciones y nuevas funcionalidades. Su infraestructura de red de entrega de contenido

garantiza una disponibilidad global y tiempos de carga reducidos para los usuarios, optimizando la experiencia independientemente de su ubicación geográfica. Además, la integración de Vercel con frameworks como React simplifica la gestión del proyecto y proporciona un rendimiento superior en comparación con hostings tradicionales o soluciones de servidor más complejas.

7.2.6. COMPONENTES AUXILIARES Y PERIFÉRICOS

La funcionalidad y la experiencia de usuario local del prototipo se complementan y optimizan con una cuidadosa selección de componentes auxiliares y periféricos. Si bien el sensor y el microcontrolador constituyen el núcleo de procesamiento y adquisición, estos componentes son esenciales para la interacción directa del usuario, la visualización inmediata de datos y la portabilidad del dispositivo.

7.2.6.1. TECLADO MATRICIAL 4X4

El teclado matricial 4×4 fue incorporado para facilitar la entrada manual del número de documento, lo cual es esencial para la correcta asociación de cada medición al paciente. Su selección principal se debe a su versatilidad, permitiendo que las teclas no numéricas funcionen como pulsadores de control para gestionar la interacción del usuario (como iniciar o detener la medición). Esta doble funcionalidad optimiza el diseño, reduciendo la necesidad de botones físicos separados y maximizando el espacio en el prototipo.

7.2.6.2. PANTALLA OLED (SH1106)

Para la visualización local de los datos, se seleccionó una pantalla de tipo OLED por ser la solución óptima para aplicaciones portátiles y dispositivos biométricos. Su elección se basa en que proporciona una visualización inmediata y nítida de los signos vitales SpO2, BPM y la información de estado, asegurando que el usuario pueda acceder a sus valores en tiempo real incluso sin la necesidad de una conexión a Internet.

Esta selección está sólidamente justificada por dos factores clave en la arquitectura electrónica del dispositivo: el bajo consumo energético y el diseño compacto. A diferencia de las pantallas LCD que requieren una retroiluminación constante, la tecnología OLED solo energiza los píxeles necesarios para mostrar el contenido. Este encendido selectivo minimiza drásticamente el consumo de energía del dispositivo, una característica crítica para prolongar la autonomía de la batería y mantener la portabilidad del sistema. Adicionalmente, su alto contraste y la calidad de los píxeles autoiluminados garantizan una lectura clara y sin reflejos bajo diversas condiciones de luz, facilitando su uso por parte del usuario o el personal médico.

Además de las ventajas de visualización y energía, la SH1106 es idónea por su método de comunicación. Utiliza el protocolo I2C, que requiere únicamente dos líneas de datos para la interacción. Esta característica permite un ahorro significativo de pines GPIO en el microcontrolador ESP32, dejando más puertos disponibles para la integración de otros periféricos clave del proyecto. A pesar de su resolución modesta, esta es más que suficiente para mostrar la información biométrica y los indicadores de estado en el formato uniforme que requiere el dispositivo. Finalmente, la gran compatibilidad de la SH1106 con el ecosistema Arduino y las librerías especializadas para ESP32, como las de Adafruit, simplifica enormemente el proceso de desarrollo e implementación del software de visualización.

7.2.6.3. BATERÍA DE LITIO 18650 Y PORTA PILAS

La elección de la fuente de alimentación recae en una batería recargable de iones de litio de formato 18650, con una capacidad nominal específica de 2200mAh. Esta selección es crítica para la autonomía del dispositivo, ya que su alta densidad energética garantiza un suministro continuo y estable para el microcontrolador ESP32 y los periféricos durante períodos prolongados de uso, manteniendo la portabilidad del sistema. La batería opera a un voltaje nominal de 3.7v, compatible con la circuitería de alimentación del sistema. La estandarización del formato 18650 es una decisión intencional de diseño que, a diferencia de las baterías soldadas o cableadas, permite al usuario final reemplazar la celda de manera sencilla y segura al final de su ciclo de vida útil o en caso de fallo. Esta característica elimina la dependencia de asistencia técnica, reduce los costos de mantenimiento a largo plazo y mejora significativamente la sostenibilidad del producto.

El alojamiento de esta celda se realiza mediante un porta pilas dedicado con terminales de resorte o presión. Este componente cumple funciones técnicas esenciales: en primer lugar, asegura una conexión eléctrica firme y sin vibraciones entre la celda y la PCB, previniendo microcortes que podrían reiniciar el microcontrolador o corromper los datos durante el monitoreo activo. En segundo lugar, facilita la inserción y remoción segura de la batería, eliminando el riesgo de daño a los terminales de la placa o el peligro asociado a la manipulación directa de celdas de litio. Este peligro se mitiga porque el porta pilas aísla físicamente los terminales positivo y negativo expuestos de la celda, previniendo el riesgo de cortocircuito accidental que podría ocurrir si un objeto metálico o el propio usuario, al manipular la celda desnuda, puentea los terminales. Esto es crítico dada la alta corriente que pueden entregar las baterías de litio. De este modo, su estructura plástica proporciona una capa de aislamiento físico dentro de la carcasa, aumentando la seguridad operativa del sistema.

7.2.6.4. MÓDULO TP4056

El Módulo TP4056 fue seleccionado como el circuito dedicado para la gestión de la celda 18650, proporcionando una solución eficiente, segura y optimizada para el proceso de recarga. Esta decisión técnica se tomó tras descartar la implementación de un circuito de carga discreto, cuya complejidad en el diseño de la PCB y el riesgo de imprecisión en la limitación de corriente y voltaje comprometerían la fiabilidad de la solución. El TP4056, en cambio, ofrece una alta integración que resuelve estas deficiencias al incorporar el circuito de carga junto a los chips de protección necesarios, lo que reduce drásticamente el tiempo de desarrollo y minimiza el riesgo de fallos por diseño, asegurando una alta fiabilidad operativa. Su función principal es implementar el perfil de carga de Corriente Constante/Voltaje Constante, el método óptimo y obligatorio para garantizar la seguridad y extender la vida útil de las baterías de iones de litio, asegurando que la tensión se limite con precisión a 4.2v.

Adicionalmente, el módulo incorpora las salvaguardas esenciales contra sobrecarga y cortocircuito, así como una protección crítica contra sobredescarga. Esta última se ejecuta mediante una desconexión automática del consumo al caer el voltaje por debajo de 2.4v. Dado que el módulo debe estar cableado permanentemente a la batería para garantizar su monitoreo, este mecanismo de protección se establece como la última línea de defensa de seguridad. El umbral de 2.4v es vital, ya que si el voltaje de la celda cayera a valores inferiores, como 1v, la celda se consideraría irreversiblemente dañada debido a la formación de cortocircuitos internos, lo que impide una recarga segura e incrementa significativamente el riesgo de fallo catastrófico o incendio. Aunque el módulo no cuenta con un indicador visual específico para el umbral de 2.4v, la luz roja de carga se convierte

en el indicador funcional indirecto: si el módulo acepta la carga y el LED se ilumina, la celda está en el rango operativo seguro y por encima del límite de sobredescarga crítica.

Finalmente, la integración de un puerto Mini-USB en el módulo estandariza el proceso de recarga, contribuyendo a la universalidad y facilidad de uso del producto. Por consiguiente, esta gestión eficiente de la energía establece una arquitectura altamente confiable y segura para la alimentación del sistema, que no solo optimiza la autonomía del dispositivo, sino que también garantiza su viabilidad como solución biomédica portátil.

7.3. ESQUEMA ELÉCTRICO Y CONEXIONES

El diseño electrónico integra un microcontrolador ESP32 con cuatro periféricos clave: una pantalla OLED, un sensor biométrico MAX30102, un teclado matricial y un pulsador de activación. La alimentación se gestiona mediante el módulo TP4056, encargado de la carga y la regulación de voltaje para el sistema. Para la comunicación, el MAX30102 y la OLED comparten el protocolo en serie I²C, optimizando el uso de pines. El teclado matricial 4x4 y el pulsador se conecta directamente a pines GPIO digitales.

7.3.1. DIAGRAMA DE CONEXIÓN

A continuación, se presenta un diagrama simplificado que detalla las conexiones entre el microcontrolador ESP32 y los periféricos principales. Este esquema ilustra cómo se combinan diferentes tipos de conexiones para integrar todos los componentes.

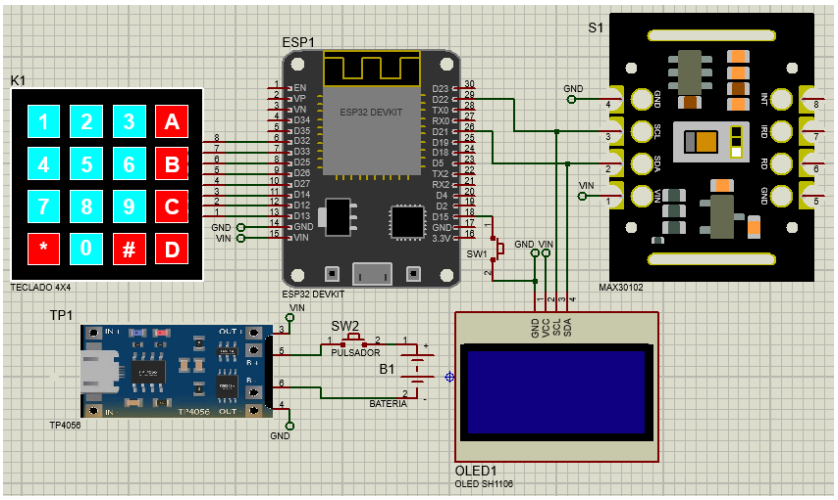


Figura 27, Diagrama de conexión. [Elaboración propia]

7.3.2. ESPECIFICACIÓN DE PINES Y PROTOCOLOS

El microcontrolador ESP32 asume el rol de maestro en el sistema, orquestando la comunicación con todos los periféricos. Este rol implica que el ESP32 controla las transferencias de datos, configura los estados de los dispositivos esclavos y utiliza protocolos de interfaz adaptados a la naturaleza y requisitos de cada componente. Esta arquitectura centralizada es clave para la eficiencia, permitiendo al ESP32 gestionar tareas simultáneas como la lectura biométrica, la visualización en la OLED y la entrada de comandos desde el teclado.

El ESP32 se comunica con el sensor MAX30102 y la pantalla OLED mediante la interfaz I²C, que permite que ambos dispositivos compartan las mismas líneas de datos y reloj. Esto minimiza el número de pines del microcontrolador requeridos.

- **ESP32 GPIO22 (SCL):** Pin de reloj serial. Este pin sincroniza la transferencia de datos.
- **ESP32 GPIO21 (SDA):** Pin de datos serial. Por aquí se transmiten y reciben los datos.

Cada dispositivo conectado a estas líneas posee una dirección única que permite al microcontrolador dirigirse a ellos de manera individual:

- **Pantalla OLED (SH1106):** Dirección I²C 0x3C.
- **Sensor MAX30102:** Dirección I²C 0x57.

El teclado matricial por otra parte requiere una conexión dedicada de pines digitales para su funcionamiento. El ESP32 escanea las filas y columnas del teclado para detectar qué tecla se ha presionado.

- **Pines de Filas (Filas R1 a R4):** Se conectan a los pines GPIO del ESP32 que se configurarán como salidas.
- **Pines de Columnas (Columnas C1 a C4):** Se conectan a los pines GPIO del ESP32 que se configurarán como entradas, con resistencias pull-up internas activadas.

El pulsador de activación se conecta a un pin de propósito general del ESP32. Su función principal no es controlar la lógica del programa, sino activar una interrupción externa para despertar al microcontrolador del estado de bajo consumo. La conexión del pulsador utiliza una configuración *pull-up* para asegurar un estado de reposo definido y evitar activaciones erróneas. Esta conexión permite que el dispositivo ahorre energía de manera eficiente y sea activado por el usuario sólo cuando sea necesario.

- **ESP32 GPIO15**: Se conecta al pulsador para activar la interrupción externa para salir del modo de bajo consumo.

El módulo TP4056 cumple la función crítica de gestión de energía y protección. El módulo está conectado a la batería para supervisar su estado , y sus terminales de salida se utilizan para alimentar todo el circuito del microcontrolador.

- **VIN:** Se conecta a la salida positiva para suministrar el voltaje de la batería a la ESP32.
- **GND:** Se conecta a la salida negativa para establecer la referencia de tierra del circuito principal.

El switch de apagado es un interruptor físico dimensionado para manejar la corriente del sistema y se intercala en la línea de alimentación. Su función es garantizar el aislamiento total del sistema de la batería de litio cuando no está en uso, asegurando un consumo de energía cero y simplificando los procedimientos de emergencia o mantenimiento.

A continuación, se detalla la asignación de pines para todos los componentes:

Tabla 1, Asignación de Pines del ESP32 para Control y Adquisición. [Elaboración propia]

Componente	Pin del Componente	Pin del ESP32	Función del Pin
Pantalla OLED	VCC	3.3V	Alimentación del módulo de pantalla.
	GND	GND	Conexión a tierra.
	SDA	GPIO21	Línea de datos I²C.
	SCL	GPIO22	Línea de datos I²C.
Sensor MAX30105	VCC	3.3V	Alimentación del módulo del sensor.
	GND	GND	Conexión a tierra.
	SDA	GPIO21	Línea de datos I²C.
	SCL	GPIO22	Línea de datos I²C.
Teclado Matricial	Fila 1	GPIO13	Pin de salida para escanear filas.
	Fila 2	GPIO12	Pin de salida para escanear filas.
	Fila 3	GPIO14	Pin de salida para escanear filas.
	Fila 4	GPIO27	Pin de salida para escanear filas.
	Columna 1	GPIO26	Pin de entrada para leer columnas.
	Columna 2	GPIO25	Pin de entrada para leer columnas.
	Columna 3	GPIO33	Pin de entrada para leer columnas.
	Columna 4	GPIO32	Pin de entrada para leer columnas.
Pulsador	Pin 1	GPIO15	Activa la interrupción para salir de bajo consumo.
	Pin 2	GND	Conexión a tierra.
TP4056	Vout+	VIN	Salida positiva del módulo al pin de entrada del ESP32.
	Vout-	GND	Salida negativa del módulo al común de tierra.
	B+	Batería positivo	Recibe el polo positivo conmutado de la batería.
	B-	Batería negativo	Conexión al polo negativo de la batería.
Switch	Pin 1	Batería positivo	Suministra la alimentación conmutada al módulo TP4056.
	Pin 2	B+	Recibe el polo positivo directo de la batería.

7.3.3. PRECAUCIONES ELÉCTRICAS Y TOLERANCIAS

La naturaleza portátil y la sensibilidad del sensor biométrico del proyecto exigieron la implementación de medidas estrictas en el diseño de hardware para garantizar la estabilidad eléctrica y la integridad de la señal. Estas precauciones fueron fundamentales para mitigar interferencias, asegurar una alimentación confiable de la batería y establecer estados lógicos definidos en las entradas del ESP32, elevando así la fiabilidad del dispositivo fuera de un entorno de laboratorio.

- **Tolerancia de Voltaje y Compatibilidad:** Se verificó la compatibilidad de la lógica de I²C a 3.3v en el sensor MAX30102 y la pantalla OLED, evitando daños por sobretensión en los pines de datos SDA y SCL.
- **Gestión y Protección de Energía:** El módulo TP4056 actúa como cargador de la batería de litio y como sistema de protección, ofreciendo protección contra sobrecarga y sobredescarga para alargar la vida útil de la batería. Adicionalmente, el módulo protege todo el circuito de picos de voltaje peligrosos provenientes de la batería, asegurando el suministro de energía estable y seguro.
- **Ahorro y Alargamiento de Energía:** Se incorporó un switch de apagado físico para aislar completamente la batería del circuito principal, garantizando un consumo de energía nulo cuando el dispositivo no está en uso y optimizando la duración de la carga.
- **Definición de Estado Lógico:** Se utilizaron las resistencias *pull-up* internas del ESP32 en las conexiones de entrada del pulsador de activación y las líneas de columnas del teclado matricial. Esta práctica asegura que las entradas digitales tengan un estado lógico definido, previniendo lecturas flotantes o erróneas.
- **Integración y Mitigación de Ruido:** El circuito fue implementado en Placa de Circuito Impreso. Esta implementación evita el ruido eléctrico asociado al cableado largo en las protoboard, reduce las impedancias y las inductancias parásitas, mejorando la integridad de la señal y la precisión de las lecturas del sensor óptico, resultando en un sistema final robusto y compacto.

7.3.4. IMPLEMENTACIÓN FÍSICA

El diseño lógico del circuito y la verificación de sus protocolos fueron materializados en una Placa de Circuito Impreso PCB, lo que marca la creación de una estructura electrónica permanente. Esta fase de diseño se centró rigurosamente en la disposición física de los componentes, priorizando la proximidad de elementos críticos como el ESP32 y el sensor MAX30102. Este diseño de trazado de pistas se realizó específicamente para minimizar la longitud de las trazas de comunicación I²C, lo cual es vital para lograr una mejor inmunidad al ruido comparado con cualquier cableado. De esta manera, se refuerza la calidad y la integridad de las señales adquiridas.

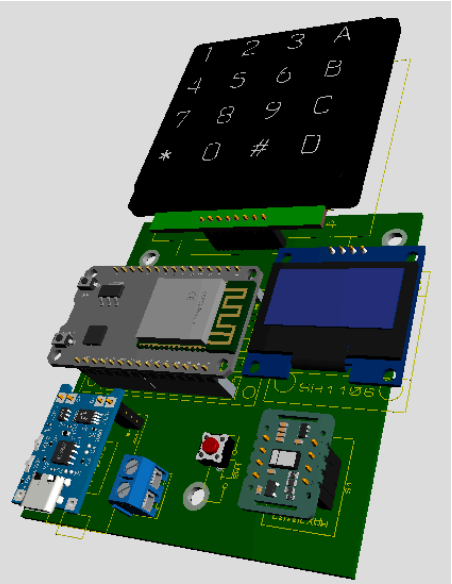


Figura 28, Modelo 3D de la Placa de Circuito Impreso PCB con Componentes. [Elaboración propia]

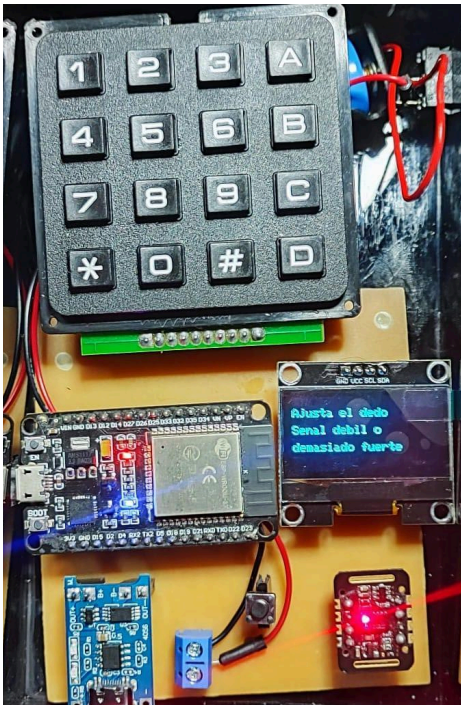


Figura 29, Placa de Circuito Impreso PCB ensamblada con Componentes. [Elaboración propia]

La PCB, al ser el soporte estructural, permitió una alta densidad de componentes, esencial para cumplir con los requisitos de tamaño de un dispositivo portátil. Tras el ensamblaje, la placa completa se integró en un encapsulado o carcasa diseñado a medida. Este componente mecánico es fundamental, ya que otorga al sistema la resistencia mecánica y la protección física necesaria contra el polvo, la humedad y los impactos. Además, la carcasa fue diseñada para alojar de forma ergonómica el teclado matricial y la pantalla OLED, garantizando la correcta interacción del usuario y la accesibilidad del sensor. Este proceso de integración asegura que el sistema electrónico sea estable y apto para su uso.



Figura 30, Dispositivo pulsioxímetro finalizado e integrado en la carcasa. [Elaboración propia]

7.4. DESARROLLO DEL DISPOSITIVO

La arquitectura adoptada para el sistema sigue un modelo de cuatro capas, en el cual se distinguen claramente los niveles de adquisición, procesamiento, almacenamiento, lógica y presentación de datos. En la primera capa se encuentra el hardware, compuesto por el microcontrolador ESP32 y el sensor óptico MAX30102, que tiene la función de registrar la saturación de oxígeno en la sangre y la frecuencia cardíaca del usuario.

El software constituye el núcleo funcional del pulsioxímetro, ya que se encarga de interpretar las señales ópticas del sensor MAX30102, procesarlas para calcular el ritmo cardíaco y la saturación de oxígeno en la sangre, y posteriormente gestionar su envío a una base de datos para almacenamiento y consulta remota. El código se implementó en C++ sobre la plataforma de desarrollo Arduino, optimizado para ejecutarse en el microcontrolador ESP32 para aprovechar su capacidad de procesamiento, conectividad Wi-Fi y eficiencia energética.

Este capítulo describe en detalle la estructura y el funcionamiento del sistema, presentando la arquitectura completa de la solución y las tecnologías utilizadas. Se explica el papel de cada componente principal y las decisiones de diseño adoptadas, abarcando desde la lógica implementada para el cálculo de BPM y SpO2, hasta el procedimiento de transmisión de los datos procesados a la base de datos, garantizando su integridad. Adicionalmente, se detalla cómo esta información es luego recuperada y presentada de forma segura en una interfaz de usuario intuitiva, ofreciendo una representación visual comprensible para el monitoreo por parte de los usuarios finales.

El diseño del prototipo se estructuró por fases con mejoras progresivas hasta alcanzar el objetivo final. El proceso se llevó a cabo de la siguiente manera:

1. **Comprobación de Hardware:** Verificación inicial del funcionamiento del sensor y la conexión con el microcontrolador.

2. **Diseño del Algoritmo:** Desarrollo del algoritmo de medición de BPM y SpO2 empleando el sensor MAX30102.

3. **Integración de Pantalla:** Incorporación de una pantalla OLED para la visualización local de las mediciones en tiempo real.

4. **Integración del Teclado:** Implementación de un teclado matricial para la entrada de datos y la interacción con el sistema.

5. **Conectividad y Configuración:** Establecimiento de la conexión Wi−Fi del dispositivo y adición de opciones de configuración como la selección de modo Wi−Fi/offline y el cambio de red.

6. **Optimización Funcional:** Implementación de funcionalidades como la detección del dedo, la verificación constante de la comunicación entre sensor y microcontrolador, el reinicio de variables y el modo de bajo consumo.

7. **Registro de Datos en Base de Datos:** Implementación del registro del número de documento para asociar mediciones al usuario, diseño de las tablas en Supabase y enlace del dispositivo.

8. **Visualización Remota:** Creación de una interfaz web para la visualización y consulta remota de los registros almacenados.

7.4.1. VERIFICACIÓN DEL SENSOR Y CONECTIVIDAD

La primera etapa del desarrollo de *software* se centró en la validación funcional de la Capa de Adquisición para confirmar la correcta interacción entre el microcontrolador ESP32 y el sensor MAX30102.

En primer lugar, se estableció la conexión física del sensor siguiendo el protocolo I²C especificado en la sección 7.3, enlazando los pines de datos SDA y reloj SCL a los pines GPIO21 y GPIO22. Se verificó la correcta inicialización del dispositivo en el bus utilizando la SparkFun MAX3010x Sensor Library como *driver* de bajo nivel. Un programa de ejemplo fue compilado para verificar que se obtenían las señales crudas de fotopletismografía de forma continua, junto con un cálculo inicial de BPM y SpO2.

Si bien este proceso confirmó la operatividad básica del sensor, el procesamiento simplificado provisto por el código de ejemplo resultó en valores de BPM y SpO2 altamente variables e inconsistentes. Esta inestabilidad demostró que el tratamiento genérico de la señal era insuficiente para cumplir con la precisión clínica requerida. Por lo tanto, esta etapa preliminar justificó la necesidad de implementar un algoritmo de procesamiento de señales sólido y dedicado que mitigara el ruido inherente y garantizara la fiabilidad de las mediciones, tal como se detalla en la siguiente sección.

```
09:32:56.161 -> red=164784, ir=214176, HR=250, HRvalid=1, SPO2=96, SPO2Valid=1
09:32:56.231 -> red=163949, ir=214028, HR=250, HRvalid=1, SPO2=96, SPO2Valid=1
09:32:56.377 -> red=164727, ir=216006, HR=214, HRvalid=1, SPO2=100, SPO2Valid=1
09:32:56.445 -> red=164840, ir=215813, HR=214, HRvalid=1, SPO2=100, SPO2Valid=1
09:32:56.563 -> red=164858, ir=216392, HR=214, HRvalid=1, SPO2=100, SPO2Valid=1
09:32:56.641 -> red=165163, ir=217164, HR=214, HRvalid=1, SPO2=100, SPO2Valid=1
09:32:56.753 -> red=165261, ir=215443, HR=214, HRvalid=1, SPO2=100, SPO2Valid=1
09:32:56.828 -> red=164326, ir=214986, HR=214, HRvalid=1, SPO2=100, SPO2Valid=1
09:32:56.974 -> red=165073, ir=217240, HR=214, HRvalid=1, SPO2=100, SPO2Valid=1
09:32:57.056 -> red=165349, ir=217318, HR=214, HRvalid=1, SPO2=100, SPO2Valid=1
09:32:57.160 -> red=165402, ir=217743, HR=214, HRvalid=1, SPO2=100, SPO2Valid=1
09:32:57.230 -> red=165643, ir=218309, HR=214, HRvalid=1, SPO2=100, SPO2Valid=1
09:32:57.377 -> red=165617, ir=216577, HR=214, HRvalid=1, SPO2=100, SPO2Valid=1
09:32:57.453 -> red=164752, ir=216015, HR=214, HRvalid=1, SPO2=100, SPO2Valid=1
09:32:57.576 -> red=165367, ir=218092, HR=214, HRvalid=1, SPO2=100, SPO2Valid=1
09:32:57.646 -> red=165676, ir=218097, HR=214, HRvalid=1, SPO2=100, SPO2Valid=1
09:32:57.775 -> red=165710, ir=218575, HR=214, HRvalid=1, SPO2=100, SPO2Valid=1
09:32:57.856 -> red=166072, ir=219229, HR=214, HRvalid=1, SPO2=100, SPO2Valid=1
09:32:57.965 -> red=165829, ir=216845, HR=214, HRvalid=1, SPO2=100, SPO2Valid=1
09:32:58.037 -> red=164883, ir=216803, HR=214, HRvalid=1, SPO2=100, SPO2Valid=1
09:32:58.153 -> red=165979, ir=219333, HR=214, HRvalid=1, SPO2=100, SPO2Valid=1
09:32:58.232 -> red=166159, ir=218742, HR=214, HRvalid=1, SPO2=100, SPO2Valid=1
09:32:58.343 -> red=165735, ir=218491, HR=214, HRvalid=1, SPO2=100, SPO2Valid=1
09:32:58.451 -> red=165802, ir=218855, HR=214, HRvalid=1, SPO2=100, SPO2Valid=1
09:32:58.563 -> red=166156, ir=217444, HR=214, HRvalid=1, SPO2=100, SPO2Valid=1
09:32:58.634 -> red=164995, ir=215609, HR=214, HRvalid=1, SPO2=100, SPO2Valid=1
09:32:58.757 -> red=165396, ir=218156, HR=214, HRvalid=1, SPO2=100, SPO2Valid=1
09:32:58.824 -> red=166021, ir=218969, HR=250, HRvalid=1, SPO2=99, SPO2Valid=1
09:32:58.972 -> red=167788, ir=223600, HR=250, HRvalid=1, SPO2=99, SPO2Valid=1
09:32:59.056 -> red=169103, ir=226481, HR=250, HRvalid=1, SPO2=99, SPO2Valid=1
09:32:59.174 -> red=170593, ir=227651, HR=250, HRvalid=1, SPO2=99, SPO2Valid=1
09:32:59.257 -> red=170541, ir=228338, HR=250, HRvalid=1, SPO2=99, SPO2Valid=1
```

Figura 31, Adquisición inicial de datos del algoritmo ejemplo. [Elaboración propia]

7.4.2. DESARROLLO DEL ALGORITMO DE BPM Y SPO_2

Tras la validación inicial de la conectividad I2C, se procedió al desarrollo del algoritmo de procesamiento de señales avanzado, esencial para mitigar la inestabilidad observada en los datos preliminares. Este algoritmo constituye el núcleo de la capa de Adquisición y se encarga de transformar las señales crudas de fotopletismografía en mediciones clínicas estables de frecuencia cardíaca y saturación de oxígeno.

Para la implementación, se utilizaron librerías clave como la librería *MAX30105.h* de SparkFun que permite la configuración de parámetros esenciales del sensor, tales como el brillo de los LED´s, el promedio de lecturas, el modo de operación y el tamaño de los *buffers* internos. Paralelamente, la librería *heartRate.h* fue fundamental para la detección eficiente de picos de pulso gracias a su función *checkForBeat*(), sentando la base para el cálculo preciso del BPM.

7.4.2.1. CÁLCULO DE LA FRECUENCIA CARDIACA

Para el cálculo de la frecuencia cardíaca, se implementó una estrategia de suavizado y promedio móvil, esencial para contrarrestar la inestabilidad de la señal cruda y proporcionar una lectura clínica fiable. Esto se logró mediante la definición de un conjunto de variables de estado: la matriz *rates* actúa como un búfer circular que almacena un historial de los últimos valores de BPM para el cálculo del promedio; la variable *rateSpot* es el índice que asegura que los nuevos valores reemplacen a los más antiguos de forma secuencial; *lastBeat* registra el *timestamp* del pulso anterior; y *beatAvg* almacena el promedio móvil reciente, que constituye la lectura final estable.

```
byte rates[RATE_SIZE];
byte rateSpot = 0;
long lastBeat = 0;
float beatAvg = 0;
```

Figura 32, Variables de estado y búfer circular para el suavizado de BPM. [Elaboración propia]

El proceso de cálculo de la frecuencia cardíaca se basa en la determinación precisa del período de tiempo entre latidos. De forma que el ciclo se inicia con la función *particleSensor.getIR()*, la cual es responsable de la lectura continua de la señal de fotopletismografía generada por el LED infrarrojo, cuyos valores son almacenados en la variable *ir*. La detección de un latido se realiza mediante la función *checkForBeat(ir)* provista por la librería, que identifica el pico de pulso en la señal PPG.

Al detectar un pulso válido, se registra el tiempo actual del sistema utilizando la función *millis()*. El cálculo de la duración de un ciclo cardíaco, denominado tiempo interlatido *Δt*, se obtiene como la diferencia entre el *timestamp* actual y el tiempo registrado del pulso anterior *lastBeat*.

Una vez determinado este intervalo, la frecuencia cardíaca se obtiene al aplicar la siguiente relación matemática, la cual transforma el período de tiempo a la unidad de latidos por minuto:

$$BPM = \frac{60}{\Delta t/1000} \quad (1)$$

Ecuación 1, Conversión del intervalo de pulso a BPM.

Donde Δt representa el intervalo de tiempo entre latidos en milisegundos. La división de Δt por 1000 normaliza el valor a segundos, permitiendo que la constante 60 convierta el resultado a la unidad estándar BPM, facilitando su comparación con rangos fisiológicos.

```
// Procesar latidos
if (checkForBeat(ir) == true) {
  long delta = millis() - lastBeat;
  lastBeat = millis();

  float beatsPerMinute = 60 / (delta / 1000.0);
```

Figura 33, Ecuación para el cálculo de la frecuencia cardíaca. [Elaboración propia]

Una vez determinado, el valor de BPM se somete a una validación dentro de un rango fisiológico seguro de 20 a 255 BPM para descartar lecturas que sean inconsistentes o valores fuera de rango. Este filtro de software previene la incorporación de errores de lectura al sistema.

Los valores validados se almacenan inmediatamente en la matriz *rates*, la cual opera como un búfer circular que mantiene un historial de los últimos seis latidos. Finalmente, se calcula un promedio móvil de estos seis valores. Este mecanismo de suavizado es esencial para obtener una lectura más estable y confiable, reduciendo drásticamente el impacto de fluctuaciones erráticas y el ruido inherente en la señal PPG.

```
  if (beatsPerMinute > 20 && beatsPerMinute < 255) {
    rates[rateSpot++] = (byte)beatsPerMinute;
    rateSpot %= RATE_SIZE;

    beatAvg = 0;
    for (byte i = 0; i < RATE_SIZE; i++) {
      beatAvg += rates[i];
    }
    beatAvg /= RATE_SIZE;
  }
}
```

Figura 34, Aplicación del promedio móvil para la mitigación de ruido y estabilización final del BPM. [Elaboración propia]

7.4.2.2. CÁLCULO DE LA SATURACIÓN DE OXÍGENO EN LA SANGRE

El cálculo de la saturación de oxígeno en el sistema se fundamenta en la ley de Lambert-Beer, específicamente, en la relación de las amplitudes de las señales de fotopletismografía obtenidas de las

mediciones de los LED´s rojo e infrarrojo. Al igual que en el cálculo de BPM, las funciones *particleSensor.getIR*() y *particleSensor.getRed*() captan las mediciones del sensor y las almacenan en las variables *ir* y *red*. Para que estas lecturas sirvan como base para el cálculo de SpO2, es imprescindible aplicar un proceso de filtrado digital que minimice el ruido y las interferencias externas.

```
// Lectura de datos del sensor
long ir = particleSensor.getIR();
long red = particleSensor.getRed();
```

Figura 35, Obtención de muestras de fotopletismografía para el cálculo de spo2. [Elaboración propia]

7.4.2.2.1. FILTRADO DE LA SEÑAL

Para suavizar la señal y minimizar el ruido provocado por movimientos o interferencias eléctricas, se implementó un filtro paso bajo de tipo exponencial. Este filtro opera atenuando las variaciones de alta frecuencia y preservando la tendencia real de los datos, lo que resulta esencial para la fiabilidad de las mediciones de SpO2.

La actualización del valor promedio se realiza ponderando la lectura previa y la nueva muestra según la siguiente ecuación de suavizado exponencial:

$$B_{nuevo} = B_{previo} \cdot f + A \cdot (1 - f) \quad (2)$$

Ecuación 2, Ecuación de Suavizado Exponencial.

Donde:

- Bnuevo es el valor filtrado (promediado) actual (*aveir* o *avered*).
- Bprevio es el valor filtrado de la muestra anterior.
- A es la nueva muestra de lectura (*ir* o *red*).
- f es el factor de atenuación (*frate*), establecido en 0.95.

Este factor de atenuación establece que el 95% del valor filtrado proviene de la medición anterior y solo el 5% de la nueva muestra. Al dar un mayor peso a las mediciones pasadas, se reduce drásticamente la influencia de valores atípicos y se produce una señal promediada que refleja la tendencia de la señal de manera estable. Este proceso garantiza que el cálculo posterior de SpO2 se base en datos filtrados y confiables además de que refleja la componente DC de la fotopletismografía.

```
// Calcular SpO2
avered = avered * frate + red * (1.0 - frate);
aveir = aveir * frate + ir * (1.0 - frate);
```

Figura 36, Ecuación del filtro digital paso bajo exponencial para señales ppg. [Elaboración propia]

7.4.2.2.2. CÁLCULO DEL FACTOR R

El factor de proporción R, representado como *rawR* en el código, se obtiene al determinar las componentes AC y DC y calcular su cociente.

La componente DC representa la absorción de luz constante por los tejidos como el hueso y la piel, además del flujo sanguíneo no pulsátil. Este flujo es el volumen de sangre base que permanece estacionario en las venas y capilares y no se ve afectado por la sístole cardíaca. Se obtiene del valor suavizado del filtro anterior *avered* y *aveir*. La Componente AC, la porción pulsátil, representa la

variación de volumen sanguíneo arterial causada por cada sístole cardíaca. El factor R se basa en el cociente de las relaciones AC/DC; esta división anula los factores de absorción constantes y aísla la señal arterial pura, que es la única que transporta información sobre la saturación.

Para cuantificar esta componente AC, se determina su valor eficaz (RMS). Esta medida es ideal para representar la potencia o energía efectiva de la pulsación arterial, proporcionando una evaluación de variación que es altamente resistente al ruido al atenuar la influencia de picos esporádicos. El RMS se define formalmente como la raíz cuadrada de la media de las diferencias al cuadrado entre la lectura cruda y su promedio suavizado:

$$AC_{RMS} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(L_{cruda} - DC)^2} \tag{3}$$

Ecuación 3, Valor Cuadrático Medio del componente AC RMS.

Donde N es el número de muestras acumuladas.

La determinación del factor R se realiza calculando el cociente entre la relación AC/DC de la luz roja y la relación AC/DC de la luz infrarroja:

$$R = \frac{AC_{RED}/DC_{RED}}{AC_{IR}/DC_{IR}} \tag{4}$$

Ecuación 4, Relación de proporciones para el cálculo del factor R.

Para optimizar el rendimiento del microcontrolador, la división por N se omite en el cálculo del factor R. Esto se debe a que el factor N es una constante que afecta por igual a los valores de RMSRojo y RMSIr y, por lo tanto, se anula en la relación de cocientes:

$$R = \frac{\left(\sqrt{\frac{1}{N}\sum_{i=1}^{N}(L_{cruda}-DC_{RED})^2}\right)/DC_{RED}}{\left(\sqrt{\frac{1}{N}\sum_{i=1}^{N}(L_{cruda}-DC_{IR})^2}\right)/DC_{IR}} \rightarrow R = \frac{\left(\frac{1}{\sqrt{N}}\sqrt{\sum_{i=1}^{N}(L_{cruda}-DC_{RED})^2}\right)/DC_{RED}}{\left(\frac{1}{\sqrt{N}}\sqrt{\sum_{i=1}^{N}(L_{cruda}-DC_{IR})^2}\right)/DC_{IR}} \tag{5}$$

Ecuación 5, Factor r cálculo para la factorización de la constante.

Como se observa, el término $\frac{1}{\sqrt{N}}$ se cancela al ser la misma constante en ambas componentes. Por lo tanto, el cálculo para el Factor R se simplifica a:

$$R = \frac{\left(\sqrt{\sum_{i=1}^{N}(L_{cruda}-DC_{RED})^2}\right)/DC_{RED}}{\left(\sqrt{\sum_{i=1}^{N}(L_{cruda}-DC_{IR})^2}\right)/DC_{IR}} \tag{6}$$

Ecuación 6. Proporción para el cálculo del factor de forma simplificada.

Por lo que para calcular la componente AC de ambas longitudes de onda se utiliza la raíz de la sumatoria de las diferencias al cuadrado entre las lecturas instantáneas y sus respectivos valores promediados

```
sumredrms += pow((red - avered), 2);
sumirrms += pow((ir - aveir), 2);
```

Figura 37, Cálculo simplificado de la componente ac de las longitudes de ondas. [Elaboración propia]

Una vez simplificadas estas expresiones y después de acumular una cantidad específica de muestras (N=100, representado como *Num* en el código), se calcula el factor R. Este valor es una de las métricas más importantes, ya que representa el cociente de las relaciones AC/DC para las dos longitudes de onda, comparando la absorción de luz pulsátil y constante. Esta comparación permite deducir el porcentaje de hemoglobina oxigenada en la sangre. Esto es fundamental porque se asocia de forma inversa con la saturación de oxígeno: a mayor oxigenación en la sangre, menor tiende a ser R.

```
if ((i % Num) == 0) {
  double rawR = (sqrt(sumredrms) / avered) / (sqrt(sumirrms) / aveir);
```

Figura 38, Cálculo del factor de proporción R. [Elaboración propia]

Para minimizar las fluctuaciones rápidas e indeseadas en el valor de rawR, se implementa un segundo filtro de suavizado exponencial con el mismo modelo utilizado para las componentes DC. Esta vez, el factor de atenuación se establece en 0.7.

```
smoothedR = alpha * smoothedR + (1 - alpha) * rawR;
```

Figura 39, Filtro de suavizado exponencial para R. [Elaboración propia]

Este factor de atenuación establece que el 70% del valor filtrado proviene de la medición anterior y el 30% proviene de la nueva muestra *rawR*. Al dar un peso considerable a la nueva muestra, el filtro es lo suficientemente sensible para reaccionar a los cambios reales en el factor R, mientras que sigue atenuando los picos espurios. Este procesamiento genera *smoothedR*, un valor de R con estabilidad mejorada, el cual es el dato de entrada utilizado en el siguiente paso del algoritmo para la conversión final a SpO2.

7.4.2.2.3. CONVERSIÓN FINAL A SPO$_2$

El cálculo de la Saturación de Oxígeno representa la etapa final del algoritmo, donde el factor R estabilizado (*smoothedR*) se transforma en un valor clínicamente relevante y procesable. Si bien el principio de funcionamiento se fundamenta en la Ley de Lambert-Beer. La relación directa entre el factor R y el SpO2 no es puramente teórica en la práctica de la oximetría de pulso. En su lugar, se emplea un modelo de calibración lineal empírica. Este modelo es un estándar en la industria porque, históricamente, ha demostrado la correlación más sólida y fiable entre la señal óptica del sensor y la saturación real de oxígeno en sangre. Este proyecto adopta dicho modelo pre-establecido para correlacionar el factor R obtenido del sensor con el porcentaje de saturación de oxígeno, lo que permite la obtención de un resultado consistente. La conversión se expresa de forma general como:

$$SpO^2 = - M \cdot R + B \tag{7}$$

Ecuación 7, Calibración lineal para la saturación de oxígeno.

Para el dispositivo desarrollado, la conversión de R a SpO2 se realiza utilizando la siguiente ecuación lineal, con los coeficientes específicos:

$$SpO^2 = -25 \cdot smoothedR + 110 \quad (8)$$

Ecuación 8, Calibración lineal final para la saturación de oxígeno.

En esta ecuación, la pendiente M se establece en −25 y la intersección B en 110. La constante M=−25 es crítica porque confirma la relación inversa entre el factor R y la saturación. La constante B=110 ajusta el punto de inicio de la curva de saturación para que la salida caiga dentro del rango clínicamente esperado. Es importante señalar que estos parámetros M y B pueden ser modificados para realizar ajustes finos o recalibraciones del dispositivo, lo que permite compensar variaciones entre diferentes sensores o adaptarse a rangos específicos de saturación.

Una vez obtenido el valor instantáneo de SpO2, es crucial limitarlo a un rango fisiológicamente aceptable para garantizar la fiabilidad. El valor de SpO2 se restringe a un rango de 70% a 100%. Esta restricción se aplica para acotar cualquier valor extremo que podría ser generado por ruido residual o artefactos de movimiento, asegurando que la salida se mantenga dentro de un límite clínicamente coherente.

```
float SpO2 = -25 * (smoothedR) + 110;
// Asegurar que SpO2 esté en un rango válido (70-100%)
SpO2 = constrain(SpO2, 70, 100);
```

Figura 40, Ecuación de calibración lineal y restricción de rango para la spo2. [Elaboración propia]

Finalmente, el valor instantáneo de SpO2 se somete a un tercer filtro de suavizado exponencial para generar la lectura final, ESpO2. Este suavizado final es esencial para proporcionar una estabilidad visual superior y mitigar el ruido residual que pudiera persistir. El filtro utiliza el factor de atenuación FSpO2=0.7, lo cual significa que el 70% del valor final proviene del histórico de medición anterior y el 30% proviene de la nueva SpO2 instantánea. Esta ponderación garantiza que el valor de ESpO2 reaccione apropiadamente a los cambios reales en la saturación, mientras que suprime el ruido de alta frecuencia, facilitando así la interpretación clínica y la consistencia en el monitoreo.

```
ESpO2 = FSpO2 * ESpO2 + (1.0 - FSpO2) * SpO2;
```

Figura 41, Cálculo final de la saturación estable. [Elaboración propia]

7.4.3. INTEGRACIÓN DE LA PANTALLA OLED

Para complementar la captura y el procesamiento de datos, se integró una pantalla OLED basada en el controlador *SH1106*, esencial para la visualización inmediata de los valores biométricos y la interacción bidireccional con el usuario. La conexión física al microcontrolador ESP32 se realiza mediante el protocolo I2C, con la dirección del dispositivo establecida en *0x3C*. La gestión de la visualización se soporta en las librerías *Adafruit_SH1106_ESP32*.h y *Adafruit_GFX*.h, las cuales facilitan la configuración de texto y la gestión de la memoria de la pantalla. La inicialización del display ocurre al inicio de la ejecución, estableciendo un formato base del texto que se mantiene uniforme a lo largo de todas las fases operativas del dispositivo.



Figura 42, Pantalla de bienvenida y presentación del sistema. [Elaboración propia]

Tras la inicialización, la pantalla gestiona el flujo inicial de interacción y la configuración del sistema. Inmediatamente, guía al usuario para que seleccione el modo de operación, presentando las opciones de conexión online u offline a través de *displayModeSelection*(). Si se selecciona el modo Online, la pantalla es crucial en la siguiente fase porque requiere que la función *configureWiFi*() comunique el proceso de conexión, mostrando las instrucciones para la configuración de la red, y confirmando el éxito con los datos de la conexión establecida. Solo después de la conexión, el sistema procede a solicitar la identificación, donde la pantalla guía al usuario a través de la función *solicitarIDPaciente*() para la entrada y verificación de la cédula mediante el teclado matricial.



Figura 43, Interfaz del portal de configuración Wi-Fi. [Elaboración propia]

Una vez establecida la conexión Wi-Fi, la pantalla asume el papel de feedback durante la fase de validación del ID. Tras ingresar la cédula, la pantalla muestra un mensaje de espera mientras la función *verificarIDenSupabase*() realiza la consulta HTTP a la base de datos remota. Este paso es fundamental para asegurar que solo los usuarios registrados puedan avanzar al modo Online. En caso de éxito, la pantalla confirma el ID; si el ID no está registrado, presenta opciones para reintentar, volver al menú principal o para entrar al modo de bajo consumo.



Figura 44, Pantalla de verificación de cédula registrada. [Elaboración propia]

Finalmente, la pantalla notifica que la configuración Wi-Fi y la verificación de ID son exitosas, lo cual permite el avance a la fase de medición. Al finalizar la fase de configuración, el dispositivo inicia el monitoreo en tiempo real de los valores biométricos, gestionado mediante la función *displayMeasurements*(bpm, spo2). La cual aplica una lógica condicional estricta para optimizar la visualización según el modo de operación seleccionado. En el modo online, la pantalla prioriza la información de conectividad, mostrando el indicador del estado de la red Wi-Fi. Por el contrario, en el modo offline, la prioridad es la legibilidad local; por ello, los valores de BPM y SpO2 se muestran utilizando un tamaño de texto ampliado.

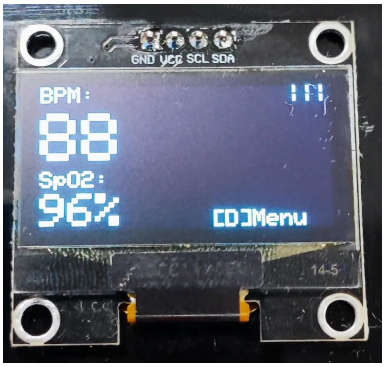


Figura 45, Visualización de datos biométricos en modo online. [Elaboración propia]

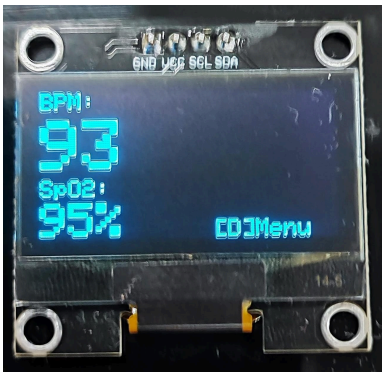


Figura 46, Visualización de datos biométricos en modo offline. [Elaboración propia]

Complementando el monitoreo continuo, la pantalla gestiona y comunica todas las transiciones de estado y notificaciones críticas del sistema. Durante el ciclo de medición, se utiliza para alertar sobre la ausencia del dedo en el sensor junto a las opciones de control disponibles. La pantalla también es el canal para las funciones de ahorro de energía: *enterLowPowerMode*() *enterLowPowerModeNoMode*() y gestionando las transiciones desde y hacia el modo de bajo consumo, mostrando instrucciones para el despertar y después, al reanudar, presentando las opciones de decisión "Continuar online o Continuar offline". De esta forma, la pantalla OLED se integra de manera continua en todo el ciclo operativo del dispositivo, consolidándose como un periférico indispensable para la interacción, la retroalimentación y la funcionalidad global del sistema.



Figura 47, Notificación de ausencia de dedo en sensor. [Elaboración propia]



Figura 48, Menú de decisión al salir del modo de bajo consumo. [Elaboración propia]

7.4.4. INTEGRACIÓN DEL TECLADO MATRICIAL

La incorporación del teclado matricial 4x4 establece el canal de comunicación esencial y único con el usuario, fundamental para la trazabilidad de las mediciones y la gestión dinámica del comportamiento del dispositivo. Su implementación requirió la definición de sus parámetros físicos, lo que incluye las dimensiones de la matriz, la asignación de los pines GPIO del ESP32 para la lectura de filas y columnas, y la construcción de la matriz keys que correlaciona la ubicación física de cada tecla con su valor alfanumérico. Para gestionar eficientemente la lectura de las pulsaciones, se utilizó la librería Keypad.h, que inicializa el objeto keypad. Este sistema de entrada de bajo nivel es indispensable y se utiliza en complemento con la pantalla OLED, la cual provee la guía visual y el *feedback* necesario al usuario en cada etapa del proceso operativo.

```
const byte ROWS = 4;
const byte COLS = 4;
byte rowPins[ROWS] = {13, 12, 14, 27};
byte colPins[COLS] = {26, 25, 33, 32};
char keys[ROWS][COLS] = {
  { '1', '4', '7', '*' },
  { '2', '5', '8', '0' },
  { '3', '6', '9', '#' },
  { 'A', 'B', 'C', 'D' }
};
Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);
```

Figura 49, Asignación de pines y mapeo del teclado 4x4 en el ESP32. [Elaboración propia]

El teclado se activa en las fases iniciales de configuración, siendo el medio principal por el cual el usuario establece el comportamiento operativo y su identidad. En la selección del modo operacional, el teclado es requerido por *displayModeSelection*() para que el usuario elija entre el modo [1] Online o [0] Offline. Esta decisión es crítica y se realiza bajo un tiempo limitado, tras el cual la inactividad induce al modo de bajo consumo *enterLowPowerModeNoMode*(). Si se opta por el modo Online, el

teclado facilita el Ingreso y Verificación de ID a través de *solicitarIDPaciente*(), donde sus funciones específicas son:

- Entrada Numérica: Para ingresar los dígitos de la cédula.
- Corrección: La tecla * permite la función de borrado de dígitos.

Confirmación: La tecla # finaliza la entrada e inicia la consulta a la base de datos remota con *verificarIDenSupabase*().

```
displayModeSelection();
updateActivity();
Serial.println("=== PASO 1: Seleccionar modo ===");
Serial.println("Esperando modo (1=Online, 0=Offline)");

while (!modeSelected) {
  char key = keypad.getKey();
  if (key) {
    updateActivity();

    if (key == '1') {
      isOnline = true;
      modeSelected = true;
      Serial.println("✓ Modo ONLINE seleccionado");
    } else if (key == '0') {
      isOnline = false;
      modeSelected = true;
      Serial.println("✓ Modo OFFLINE seleccionado");
    }
  }

  // Timeout en selección de modo
  if (checkMenuTimeout(MODE_SELECT_TIMEOUT)) {
    displayModeSelection();
    updateActivity();
  }

  delay(50);
}
```

Figura 50, Menú principal de selección de modo online/offline. [Elaboración propia]

```
char key = keypad.getKey();
if (key) {
  updateActivity();// Ingresar dígitos
  if (key >= '0' && key <= '9') {
    if (idBuffer.length() < MAX_ID_LENGTH) {
      idBuffer += key;
      display.fillRect(0, 25, 128, 8, BLACK);
      display.setCursor(0, 25);
      display.print("ID: ");
      display.print(idBuffer);
      display.print(" (");
      display.print(idBuffer.length());
      display.print("/");
      display.print(MAX_ID_LENGTH);
      display.print(")");
      display.display();
      Serial.println("ID actual: " + idBuffer + " (" + String(idBuffer.length()) + " dígito
    } else { // Máximo alcanzado
      display.fillRect(0, 62, 128, 2, BLACK);
      display.setCursor(0, 62);
      display.print("Max ");
      display.print(MAX_ID_LENGTH);
      display.print(" dígitos!");
      display.display();
      delay(800);
      display.fillRect(0, 62, 128, 2, BLACK);
      display.display();
    }
  }
```

Figura 51, Interfaz de ingreso numérico de cédula. [Elaboración propia]

```
// Borrar último dígito
else if (key == '*') {
  if (idBuffer.length() > 0) {
    idBuffer.remove(idBuffer.length() - 1);
    display.fillRect(0, 25, 128, 8, BLACK);
    display.setCursor(0, 25);
    display.print("ID: ");
    display.print(idBuffer);
    display.print(" (");
    display.print(idBuffer.length());
    display.print("/");
    display.print(MAX_ID_LENGTH);
    display.print(")");
    display.display();
    Serial.println("Dígito borrado. ID actual: " + idBuffer);
  }
}
// Confirmar
else if (key == '#') {
  if (idBuffer.length() >= MIN_ID_LENGTH) {
    Serial.println("✓ Cédula ingresada: " + idBuffer);
    ingresando = false; // Salir del loop de ingreso
```

Figura 52, Opciones para cancelar o confirmar el número de cédula. [Elaboración propia]

La funcionalidad del teclado se extiende al manejo de interrupciones y errores. Durante el proceso de Control del Portal WiFi en *configureWiFi*(), la tecla D funciona como un comando de anulación inmediata, que detiene el portal cautivo y provoca un reinicio completo del sistema , permitiendo al usuario volver a intentar la configuración. De igual forma, en la Decisión Post-Verificación, si la función *verificarIDenSupabase*() falla, es decir el ID no está registrado, el teclado se activa para ofrecer al usuario dos opciones: * para reintentar el ingreso de la cédula, o D para reiniciar el sistema y volver al menú principal.

```
unsigned long wifiStartTime = millis();
while (WiFi.status() != WL_CONNECTED) {
  char key = keypad.getKey();
  if (key == 'D') {
    Serial.println("Config WiFi cancelada");
    display.clearDisplay();
    display.setCursor(0, 20);
    display.println("Configuracion");
    display.setCursor(0, 35);
    display.println("cancelada");
    display.display();
    delay(2000);
    wifiManager.stopConfigPortal();
    return false;
  }
```

Figura 53, Opciones para cancelar la configuración Wi-Fi. [Elaboración propia]

```
// Mostrar opciones
display.clearDisplay();
display.setCursor(0, 10);
display.println("ID no registrado");
display.setCursor(0, 30);
display.println("[*] Reintentar ID");
display.setCursor(0, 45);
display.println("[D] Volver al menu");
display.display();
Serial.println("\n[*] Reintentar ingreso de ID");
Serial.println("[D] Volver al menú principal");
// Esperar decisión del usuario
unsigned long startTime = millis();
while (millis() - startTime < 30000) { // 30 segundos timeout
  char key = keypad.getKey();
  if (key) {
    updateActivity();

    if (key == '*') {
      Serial.println("✓ Usuario eligió reintentar");
      return true; // Reintentar
    } else if (key == 'D') {
      Serial.println("✓ Usuario eligió volver al menú");
      return false; // Volver al menú
    }
  }

  delay(100);
}
```

Figura 54, Opciones para cancelar o confirmar el número de cédula. [Elaboración propia]

Una vez que el dispositivo entra en el ciclo de operación y medición, el teclado permanece activo para la gestión y el control rápido del dispositivo sin interrumpir la adquisición de la señal biométrica. La

función *processKeyDuringMeasurement*() dirige el control mantienendo dos comandos críticos como atajos directos y establece la puerta de enlace a la configuración:

- Ahorro de Energía (#): Activa de forma directa la función de bajo consumo *enterLowPowerMode*().

- Reinicio de Variables (*): Ejecuta un *resetVariables*() inmediato, forzando la limpieza de los buffers de señal para asegurar que la próxima medición comience desde un estado limpio.

- Menú de Configuración (D): La tecla 'D' redirige el flujo de control hacia una nueva interfaz *displayConfigMenu*().

```
//============================================================
void processKeyDuringMeasurement(char key) {
  updateActivity();

  if (key == '#') {
    // Sleep
    Serial.println("   Tecla [#] - Entrando en sleep");
    enterLowPowerMode();
  } else if (key == '*') {
    // Reset
    Serial.println("   Tecla [*] - Reset variables");
    resetVariables();
  } else if (key == 'D') {
    // Menú de configuración
    Serial.println("   Tecla [D] - Menú");
    displayConfigMenu();
  }
}
```

Figura 55, Control rápido en modo operativo. [Elaboración propia]

Esta interfaz centraliza las funcionalidades críticas en un menú jerárquico visible en la pantalla OLED, mejorando la robustez operativa y el control. Las opciones disponibles dentro del menú son:

- [1] Ver info: Permite acceder al estado de diagnóstico del sistema, mostrando el modo operativo, el ID del paciente actual y, si está conectado, la SSID y la intensidad de la señal Wi-Fi.

- [2] Reset variables: Ejecuta el restablecimiento de variables y buffers de forma controlada.

- [3] Cambiar modo: Inicia el proceso para alternar el estado del dispositivo entre los modos ONLINE y OFFLINE, lo cual culmina con un *ESP.restart*() para aplicar el cambio de modo de forma limpia.

- [4] Reset WiFi: Esta opción implementa una lógica condicional que solo es visible si el estado isOnline es verdadero. Permite borrar las credenciales almacenadas por el WiFiManager, seguido de un reinicio.

- [#]Sleep y [D]Salir: Opciones de navegación que permiten al usuario activar el modo de bajo consumo desde el menú o retornar a la pantalla de medición, respectivamente.

```
void displayConfigMenu() {
  bool inMenu = true;
  unsigned long menuStartTime = millis();

  while (inMenu) {
    display.clearDisplay();
    display.setTextSize(1);
    display.setCursor(0, 0);
    display.println("=== MENU ===");
    display.setCursor(0, 12);
    display.println("[1] Ver info");
    display.setCursor(0, 22);
    display.println("[2] Reset variables");
    display.setCursor(0, 32);
    display.println("[3] Cambiar modo");

    // Solo mostrar Reset WiFi si está en modo Online
    if (isOnline) {
      display.setCursor(0, 42);
      display.println("[4] Reset WiFi");
    }

    display.setCursor(0, 52);
    display.println("[#]Sleep [D]Salir");
    display.display();

    char key = keypad.getKey();
```

Figura 56, Panel de opciones y control rápido en modo operativo. [Elaboración propia]

Finalmente, el teclado es esencial para la gestión del modo de bajo consumo. Al despertar del modo de bajo consumo mediante el botón externo, el sistema invoca a *esp_sleep_wakeup_cause_t wakeup_reason* y presenta un menú de decisión. En este punto, el teclado se utiliza para:

- Pulsar 0: Continuar la operación en el Online.
- Pulsar 1: Continuar la operación en el Offline.

```
// Pantalla de bienvenida
display.clearDisplay();
display.setTextSize(2);
display.setCursor(10, 10);
display.println("BpmSp02");
display.setTextSize(1);
display.setCursor(20, 35);
display.println("Version 1.9");
display.setCursor(10, 50);
display.println("Bienvenido!");
display.display();
delay(WELCOME_DELAY);

// Verificar si hay despertar desde deep sleep
esp_sleep_wakeup_cause_t wakeup_reason = esp_sleep_get_wakeup_cause();

if (wakeup_reason == ESP_SLEEP_WAKEUP_EXT0) {
  Serial.println("═══════════════════════════════════");
  Serial.println("  DESPERTANDO DESDE DEEP SLEEP");
  Serial.println("═══════════════════════════════════");

  // Si estaba en Online, volver a pedir cédula
  waitingForWakeDecision = true;

  display.clearDisplay();
  display.setTextSize(1);
  display.setCursor(0, 10);
  display.println("Sistema despertado");
  display.setCursor(0, 30);
  display.println("[1] Continuar Online");
  display.setCursor(0, 45);
  display.println("[0] Continuar Offline");
  display.display();

  // ← IMPORTANTE: Usar timer independiente para wake decision
  unsigned long wakeStartTime = millis();
```

Figura 57, Opción para seleccionar el modo después del modo de bajo consumo. [Elaboración propia]

7.4.5. CONFIGURACIÓN Y CONEXIÓN WI-FI

La conexión a una red inalámbrica es el pilar fundamental para la funcionalidad de telemetría del dispositivo, habilitando el envío continuo de datos fisiológicos a una base de datos remota. Para optimizar este proceso, el microcontrolador ESP32 utiliza la librería *WiFiManager.h*, la cual implementa un portal cautivo para la gestión automática de credenciales, eliminando la necesidad de codificarlas en el firmware.

Este proceso se inicia inmediatamente después de que el usuario selecciona el modo online en el menú principal, momento en que el sistema invoca la función *configureWiFi*(). Al ejecutarse, la pantalla OLED guía al usuario para que se conecte a un punto de acceso temporal, "ESP32_AutoConnect", y acceda a la IP local 192.168.4.1 desde un navegador para ingresar las credenciales de su red.

```
bool configureWiFi() {
  WiFiManager wifiManager;

  display.clearDisplay();
  display.setTextSize(1);
  display.setCursor(0, 0);
  display.println("Configurando WiFi");
  display.setCursor(0, 15);
  display.println("Conectate a:");
  display.setCursor(0, 25);
  display.println("ESP32_AutoConnect");
  display.setCursor(0, 35);
  display.println("Abre: 192.168.4.1");
  display.setCursor(0, 50);
  display.println("[D] Cancelar");
  display.display();

  wifiManager.setConfigPortalTimeout(WIFI_CONFIG_TIMEOUT / 1000);
  wifiManager.setConfigPortalBlocking(false);
  wifiManager.autoConnect("ESP32_AutoConnect");
```

Figura 58, Interfaz de configuración Wi-Fi mediante portal cautivo. [Elaboración propia]

El proceso de configuración de la red Wi-Fi está sujeto a un tiempo de espera predefinido por *WIFI_CONFIG_TIMEOUT*. Si el usuario no completa la introducción de las credenciales dentro de este límite, o si decide anular manualmente la configuración pulsando la tecla 'D', el sistema lo interpreta como un fallo. En ambos casos, el ESP32 desactiva inmediatamente el portal cautivo y ejecuta un reinicio completo *ESP.restart*(). Este reinicio asegura que el usuario retorne directamente al menú principal, donde podrá intentar la configuración nuevamente.

```
unsigned long wifiStartTime = millis();

while (WiFi.status() != WL_CONNECTED) {
  char key = keypad.getKey();
  if (key == 'D') {
    Serial.println("Config WiFi cancelada");
    display.clearDisplay();
    display.setCursor(0, 20);
    display.println("Configuracion");
    display.setCursor(0, 35);
    display.println("cancelada");
    display.display();
    delay(2000);
    wifiManager.stopConfigPortal();
    return false;
  }

  if (millis() - wifiStartTime > WIFI_CONFIG_TIMEOUT) {
    display.clearDisplay();
    display.setCursor(0, 10);
    display.println("Tiempo agotado");
    display.setCursor(0, 25);
    display.println("[D] Volver al menu");
    display.display();

    while (true) {
      char key = keypad.getKey();
      if (key == 'D') {
        wifiManager.stopConfigPortal();
        return false;
      }
      delay(100);
    }
  }

  wifiManager.process();
  delay(100);
}
```

Figura 59, Proceso de configuración fallida por tiempo de espera o cancelación manual. [Elaboración propia]

De otra forma una vez que el proceso de gestión de credenciales finaliza correctamente y el ESP32 logra establecer la conectividad con la red, el dispositivo lo confirma como una conexión exitosa. Para notificar al usuario, el sistema utiliza la pantalla OLED para mostrar la SSID de la red a la que se ha conectado y la dirección IP que le ha sido asignada. Adicionalmente, el éxito de la conexión se señaliza activando el LED indicador correspondiente.

```
  display.clearDisplay();
  display.setCursor(0, 0);
  display.println("Conectado a WiFi");
  display.setCursor(0, 15);
  display.println("SSID: " + WiFi.SSID());
  display.setCursor(0, 25);
  display.println("IP: " + WiFi.localIP().toString());
  display.setCursor(0, 40);
  display.println("Supabase: Listo");
  display.display();
  Serial.println("☁️ WiFi OK - Conectado a Supabase");
  Serial.println("IP: " + WiFi.localIP().toString());
  digitalWrite(LED_PIN, HIGH);
  delay(3000);
  return true;
}
```

Figura 60, Configuración de conexión Wi-Fi establecida, confirmación de SSID y dirección IP. [Elaboración propia]

Con la conectividad Wi-Fi asegurada, el sistema procede a la fase de validación de identidad, un requisito fundamental introducido para asegurar la trazabilidad y seguridad de los datos. La función *verificarIDenSupabase*() toma la cédula ingresada por el usuario y construye una consulta HTTP GET hacia el *endpoint* de Supabase dedicado a la tabla usuarios.

El sistema utiliza esta consulta para determinar si el ID proporcionado existe dentro de la base de datos:

- ID Válido: Si el ID es encontrado, se almacena como *idPacienteActual* y se permite el avance a la fase de telemetría.

- ID Inválido: Si el ID no existe o la conexión falla, se muestra una pantalla de error que sugiere al usuario reintentar el ingreso del ID (*) o reiniciar el sistema (D), impidiendo el inicio de la medición y el envío de datos.



```
bool verificarIDenSupabase(String cedula) {
  if(WiFi.status() != WL_CONNECTED) {
    Serial.println("❌ Error: WiFi no conectado para verificación");
    return false;
  }

  HTTPClient http;
  // Construir URL de consulta
  String url = String(SUPABASE_USUARIOS_URL) + "?cedula=eq."
  + cedula + "&select=cedula";

  Serial.println("🔍 Verificando ID en base de datos...");
  Serial.println("URL: " + url);

  http.begin(url);
  http.addHeader("apikey", SUPABASE_KEY);
  http.addHeader("Authorization", String("Bearer ") + SUPABASE_KEY);

  int httpResponseCode = http.GET();

  if (httpResponseCode > 0) {
    String response = http.getString();
    Serial.println("📡 Response Code: " + String(httpResponseCode));
    Serial.println("📄 Response: " + response);

    http.end();
    // Si la respuesta es "[]" significa que no existe
    // Si la respuesta contiene datos, existe
    if (httpResponseCode == 200) {
      if (response == "[]" || response.length() < 3) {
        Serial.println("❌ ID no encontrado en base de datos");
        return false;
      } else {
        Serial.println("✅ ID verificado exitosamente");
        return true;
      }
    } else {
      Serial.println("⚠️ Código de respuesta inesperado");
      return false;
    }
  } else {
    Serial.println("❌ Error HTTP en verificación: " + String(httpResponseCode));
    http.end();
    return false;
  }
}
```

Figura 61, Estructura para la consulta y verificación del número de cédula a la base de datos. [Elaboración propia]

Con la conectividad Wi-Fi y el ID del paciente asegurados y validados, el sistema procede a la fase de telemetría. Para implementar la comunicación, se utiliza la librería *HTTPClient.h* del ESP32, la cual gestiona la transferencia de datos mediante el protocolo HTTP. Esta operación es delegada a la función *sendToSupabase*(), que prepara la información encapsulando las métricas fisiológicas como el ID del paciente, la frecuencia cardíaca y la saturación de oxígeno, dentro de un objeto en formato JSON. Este objeto se envía como una solicitud POST al *endpoint* de la API REST de Supabase definido por *SUPABASE_URL*. Para un monitoreo efectivo, esta comunicación se realiza de forma periódica, en intervalos de 5 segundos por *DATA_SEND_INTERVAL*, asegurando así un flujo constante y oportuno de datos vitales.

```
void sendToSupabase(float bpm, float spo2) {
  if(WiFi.status() != WL_CONNECTED) {
    Serial.println("Error: WiFi no conectado");
    lastDataSent = false;
    return;
  }

  HTTPClient http;

  http.begin(SUPABASE_URL);
  http.addHeader("Content-Type", "application/json");
  http.addHeader("apikey", SUPABASE_KEY);
  http.addHeader("Authorization", String("Bearer ") + SUPABASE_KEY);
  http.addHeader("Prefer", "return=minimal");

  StaticJsonDocument<256> doc;
  doc["cedula_paciente"] = idPacienteActual;
  doc["ritmo_cardiaco"] = bpm;
  doc["spo2"] = spo2;

  String jsonData;
  serializeJson(doc, jsonData);

  Serial.println("Enviando a Supabase:");
  Serial.println(jsonData);

  int httpResponseCode = http.POST(jsonData);

  if (httpResponseCode > 0) {
    Serial.println("📨 Response Code: " + String(httpResponseCode));
    if (httpResponseCode == 201 || httpResponseCode == 200) {
      Serial.println("✅ Datos enviados exitosamente");
      lastDataSent = true;
    } else {
      Serial.println("⚠️ Respuesta inesperada");
      lastDataSent = false;
    }
  } else {
    Serial.println("❌ Error HTTP: " + String(httpResponseCode));
    lastDataSent = false;
  }

  http.end();
}
```

Figura 62, Estructura del envío de datos fisiológicos a la base de datos. [Elaboración propia]

7.4.6. FUNCIONES DE DETECCIÓN Y OPTIMIZACIÓN

Una vez establecidas las funciones principales de muestreo y procesamiento, se integraron mecanismos de control basados en eventos para aumentar la fiabilidad y la eficiencia energética del dispositivo. Estas optimizaciones incluyen la verificación inicial del *hardware* mediante la detección del sensor MAX30102 y la gestión inteligente del estado del sistema basada en la detección de la presencia o ausencia del dedo.

7.4.6.1. MODO DE DETECCIÓN DEL SENSOR

Como parte del desarrollo, se implementó un mecanismo crucial para la detección del sensor MAX30102 en el inicio del sistema. Esta rutina se ejecuta una única vez en la fase de inicialización del *setup()* para garantizar que el componente de *hardware* esté conectado y listo para operar. El proceso establece la comunicación a través del protocolo I2C en la dirección 0x57 y verifica la respuesta del sensor mediante la función *particleSensor.begin()*. Si el dispositivo no es detectado, el programa detiene su ejecución inmediatamente con un bucle *while(1)*, y el sistema envía el mensaje de error "MAX30102 no encontrado", evitando fallos de lectura y un comportamiento indefinido.

```
if (!particleSensor.begin(Wire, 0x57)) { // Cambia la dirección I2C del
  display.clearDisplay();
  display.setCursor(0, 0);
  display.println("ERROR:");
  display.setCursor(0, 15);
  display.println("MAX30102 no encontrado");
  display.setCursor(0, 30);
  display.println("Verifica conexiones");
  display.display();
  Serial.println("MAX30102 no encontrado. Verifica la conexión.");
  while (1);
}
```

Figura 63, Rutina de verificación de la conexión del sensor y mensaje de error. [Elaboración propia]

Este método de autodetección es fundamental para la fiabilidad, ya que evita que el sistema intente procesar datos de un sensor que no está presente. Esto garantiza un comportamiento estable y una respuesta clara ante fallas de *hardware* desde el primer momento.

7.4.6.2. GESTIÓN DE LA PRESENCIA O AUSENCIA DEL DEDO

La función de detección de dedo es un componente esencial para asegurar la validez de la medición. El *software* monitorea continuamente el valor de intensidad de luz infrarroja *ir* capturada por el sensor y lo compara con un umbral predefinido *FINGER_ON*. Si el valor registrado cae por debajo de este límite, el sistema interpreta que el dedo ha sido retirado y, por lo tanto, la medición biométrica no es válida. Cuando se cumple la condición de ausencia *ir* < *FINGER_ON*, el programa detiene los cálculos de biometría y notifica al usuario. Para evitar la saturación del bus de comunicación y el desperdicio de ciclos de procesamiento, además se implementó una lógica de temporización para controlar la frecuencia de esta notificación.

```
if (ir < FINGER_ON) {
    if (millis() - lastPrintTime > NO_FINGER_PRINT_INTERVAL) {
      display.clearDisplay();
      display.setCursor(10, 10);
      display.println("No hay dedo");
      display.setCursor(10, 25);
      display.println("detectado");
      display.display();
      lastPrintTime = millis();
    }
```

Figura 64, Condición principal para la detección de la presencia del dedo. [Elaboración propia]

Esta lógica funciona con dos elementos: un registro del último momento en que se envió el mensaje *lastPrintTime* y un intervalo mínimo de espera *NO_FINGER_PRINT_INTERVAL*, fijado en 5000 milisegundos. El sistema comprueba si han transcurrido más de 5 segundos desde la última vez que se mostró la alerta. Solo si se ha superado este tiempo, se envía un nuevo mensaje. De esta manera, se minimizan las operaciones de escritura en la interfaz de usuario, optimizando el rendimiento del sistema y reduciendo el consumo energético.

La persistente ausencia del dedo no solo dispara esta notificación, sino que también activa, tras un tiempo determinado, otros dos mecanismos de optimización de inactividad, operando bajo dos umbrales temporales distintos. Si la inactividad del dedo excede los 60 segundos se activa la función de reinicio de variables. Por otro lado, si la inactividad se prolonga más allá de los 120 segundos, el sistema inicia el modo de bajo consumo.

7.4.6.3. REINICIO DE VARIABLES

El reinicio de variables es una estrategia de optimización esencial diseñada para mantener la integridad y fiabilidad de las mediciones futuras. Esta función se activa mediante dos mecanismos: automáticamente, una vez que el tiempo de ausencia del dedo excede el límite temporal *RESET_VARS_TIMEOUT*, de 60 segundos; o manualmente, en cualquier momento de la medición, al pulsar la tecla * en el teclado matricial.

```
// Reiniciar variables si no hay detección durante el tiempo configurado
if (isFingerDetected && millis() - lastDetectedTime > RESET_VARS_TIMEOUT) {
  resetVariables();
  isFingerDetected = false;
  Serial.println("Reinicio de variables debido a inactividad prolongada.");
}
```

Figura 65, Lógica de activación del reinicio de variable. [Elaboración propia]

Una vez activada, la función procede a limpiar y neutralizar todos los registros históricos del sistema. Su propósito principal es poner a cero los *buffers* de señal y restablecer los promedios de la frecuencia cardíaca *beatAvg* y la saturación de oxígeno *ESpO2*. Al ejecutar este reinicio, se garantiza que las nuevas mediciones comiencen desde un estado completamente limpio. Esta medida es fundamental para prevenir cualquier posible distorsión de los resultados causada por la contaminación de datos residuales de sesiones anteriores, asegurando la fiabilidad de la próxima lectura biométrica.

```
void resetVariables() {
  for (byte j = 0; j < RATE_SIZE; j++) {
    rates[j] = 0;
  }
  rateSpot = 0;
  lastBeat = 0;
  beatAvg = 0;
  avered = 0;
  aveir = 0;
  sumirrms = 0;
  sumredrms = 0;
  i = 0;
  ESpO2 = 0;
  smoothedR = 1.0;

  display.clearDisplay();
  display.setCursor(10, 20);
  display.println("Reiniciando");
  display.setCursor(10, 35);
  display.println("variables...");
  display.display();
  Serial.println("Variables reiniciadas");
  delay(1000);
}
```

Figura 66, Función principal para el reinicio de variables. [Elaboración propia]

7.4.6.4. MODO DE BAJO CONSUMO

El modo de bajo consumo es una función crítica diseñada para maximizar la eficiencia energética del dispositivo. Este modo se inicia automáticamente cuando el tiempo de inactividad excede el límite definido por *LOW_POWER_TIMEOUT*, de 120 segundos, o es activado manualmente por el usuario mediante la tecla # del teclado matricial.

```
  if (!lowPowerMode && millis() - lastDetectedTime > LOW_POWER_TIMEOUT) {
    enterLowPowerMode();
  }

  return;
}
```

Figura 67, Lógica de activación por tiempo de inactividad del modo de bajo consumo. [Elaboración propia]

La activación de este modo es gestionada por la función *EnterLowPowerMode*(), la cual optimiza el consumo siguiendo una secuencia estricta.

La función procede directamente con la desactivación de periféricos: el sensor MAX30102 se desactiva, el LED indicador se apaga, y un mensaje en la pantalla OLED informa al usuario que el dispositivo entrará en suspensión.

Finalmente, el microcontrolador ESP32 se configura para despertar mediante una interrupción externa con la pulsación del botón y entra en el estado de Light Sleep, donde su consumo de energía es mínimo.

Un modo alternativo, *enterLowPowerModeNoMode*(), se utiliza si el usuario no selecciona un modo de operación en la configuración inicial dentro del tiempo límite, llevando el dispositivo directamente al Light Sleep.



```
//============================================================
void enterLowPowerMode() {
  lowPowerMode = true;

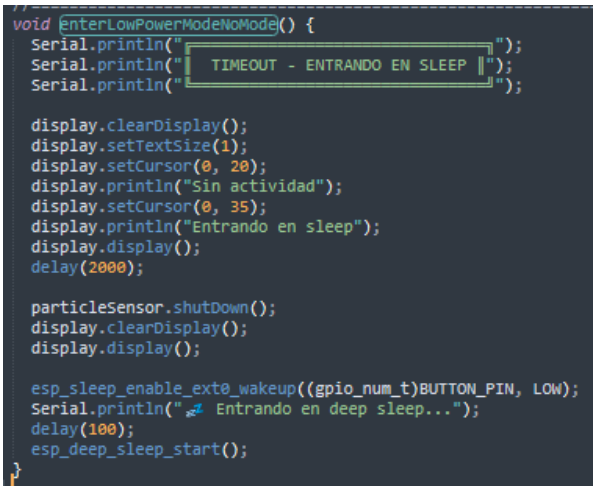
  Serial.println("╔═══════════════════════════════╗");
  Serial.println("║  ENTRANDO EN MODO SLEEP       ║");
  Serial.println("╚═══════════════════════════════╝");

  display.clearDisplay();
  display.setTextSize(1);
  display.setCursor(0, 15);
  display.println("Modo bajo consumo");
  display.setCursor(0, 30);
  display.println("Presiona boton");
  display.setCursor(0, 45);
  display.println("para despertar");
  display.display();
  delay(LOW_POWER_INFO_DELAY);

  particleSensor.shutDown();
  display.clearDisplay();
  display.display();

  esp_sleep_enable_ext0_wakeup((gpio_num_t)BUTTON_PIN, LOW);
  Serial.println("💤 Entrando en deep sleep...");
  delay(100);
  esp_deep_sleep_start();
}
```

Figura 68, Secuencia de apagado del sensor con modo seleccionado y notificación al usuario. [Elaboración propia]



```
//============================================================
void enterLowPowerModeNoMode() {
  Serial.println("╔═══════════════════════════════╗");
  Serial.println("║  TIMEOUT - ENTRANDO EN SLEEP  ║");
  Serial.println("╚═══════════════════════════════╝");

  display.clearDisplay();
  display.setTextSize(1);
  display.setCursor(0, 20);
  display.println("Sin actividad");
  display.setCursor(0, 35);
  display.println("Entrando en sleep");
  display.display();
  delay(2000);

  particleSensor.shutDown();
  display.clearDisplay();
  display.display();

  esp_sleep_enable_ext0_wakeup((gpio_num_t)BUTTON_PIN, LOW);
  Serial.println("💤 Entrando en deep sleep...");
  delay(100);
  esp_deep_sleep_start();
}
```

Figura 69, Secuencia de apagado del sensor sin modo seleccionado y notificación al usuario. [Elaboración propia]

La salida del modo de bajo consumo se activa al detectar la interrupción externa provocada por la pulsación del botón.

Al despertar, el sistema automáticamente reactiva el sensor de medición y presenta un menú en la pantalla para que el usuario tome una decisión. Este menú permite dos acciones inmediatas, controladas por el teclado matricial:

- Pulsar la tecla 0 para continuar la operación en el modo Online.
- Pulsar la tecla 1 para forzar un reinicio completo del dispositivo, lo que permite al usuario ingresar una nueva identificación o seleccionar un modo de operación distinto.

Si el usuario no pulsa ninguna tecla dentro del tiempo límite de decisión establecido, el sistema asume por defecto la acción de continuar la operación en el modo que estaba activo antes de la suspensión.

```
// Pantalla de bienvenida
display.clearDisplay();
display.setTextSize(2);
display.setCursor(10, 10);
display.println("BpmSp02");
display.setTextSize(1);
display.setCursor(20, 35);
display.println("Version 1.9");
display.setCursor(10, 50);
display.println("Bienvenido!");
display.display();
delay(WELCOME_DELAY);

// Verificar si hay despertar desde deep sleep
esp_sleep_wakeup_cause_t wakeup_reason = esp_sleep_get_wakeup_cause();

if (wakeup_reason == ESP_SLEEP_WAKEUP_EXT0) {
  Serial.println("═══════════════════════════════════");
  Serial.println("  DESPERTANDO DESDE DEEP SLEEP");
  Serial.println("═══════════════════════════════════");

  // Si estaba en Online, volver a pedir cédula
  waitingForWakeDecision = true;

  display.clearDisplay();
  display.setTextSize(1);
  display.setCursor(0, 10);
  display.println("Sistema despertado");
  display.setCursor(0, 30);
  display.println("[1] Continuar Online");
  display.setCursor(0, 45);
  display.println("[0] Continuar Offline");
  display.display();

  // ← IMPORTANTE: Usar timer independiente para wake decision
  unsigned long wakeStartTime = millis();
```

Figura 70, Lógica de detección y función de reactivación del sistema. [Elaboración propia]

7.4.7. BASE DE DATOS

Para la persistencia de los datos del sistema, se utilizó PostgreSQL sobre la infraestructura de Supabase, garantizando un ambiente relacional sólido y escalable en la nube. Esta elección se basa en un diseño relacional compuesto por tres tablas interconectadas: Usuarios, Mediciones y Alertas, cuya estructura se diseñó para asegurar la integridad referencial y la trazabilidad de la información clínica.

La estructura de la base de datos se fundamenta en la integridad de las relaciones entre sus tres entidades, cada una con una función crítica dentro del flujo de datos del sistema.

7.4.7.1. TABLA USUARIOS

La función primordial de esta tabla es la gestión centralizada de perfiles, roles y seguridad. Almacena la información demográfica esencial para la identificación del paciente y define los permisos de acceso mediante el campo *rol*. Su llave primaria, *cedula*, actúa como el identificador único y principal punto de anclaje para el resto de la base de datos.

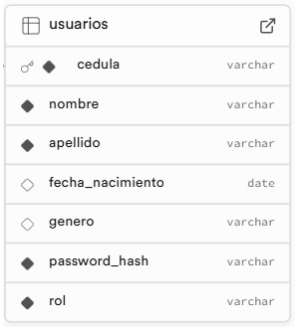


Figura 71, Estructura de la tabla Usuarios: Identificación y gestión de perfiles. [Elaboración propia]

Los registros que se ingresan a esta tabla, lo hacen exclusivamente a través de la capa de lógica y servicios (*Backend*). El proceso se inicia mediante una solicitud desde la interfaz web (*Frontend*) tras el registro de un nuevo usuario. El Backend gestiona la solicitud, asegurando la integridad de los datos mediante la validación estricta de campos, la generación del *hash* seguro para la contraseña, y la autorización final para la inserción en la base de datos.

7.4.7.2. TABLA MEDICIONES

Esta tabla tiene la función esencial de almacenar y trazar los datos fisiológicos adquiridos en tiempo real por el dispositivo. Cada registro individual documenta el estado de los signos vitales en un momento preciso . Su llave foránea, *cedula_paciente*, asegura que cada medición esté inequívocamente asociada a un usuario registrado en el sistema.



Figura 72, Estructura de la tabla Mediciones: Registro de datos fisiológicos. [Elaboración propia]

El registro de datos en esta tabla se origina en la capa de adquisición y procesamiento. Específicamente, el dispositivo ESP32 utiliza su módulo de conectividad Wi−Fi para iniciar una conexión segura y transmitir directamente los valores calculados de SpO2 y BPM, junto con la cédula del paciente ingresada localmente, a la base de datos. El sistema de base de datos recibe el registro y automáticamente inicia un proceso de validación de integridad referencial: verifica la existencia de la cédula en la tabla Usuarios para cumplir con la restricción de la llave foránea. Solo si la identidad es válida y se cumplen las restricciones del esquema, se permite la inserción del registro de medición en la tabla.

Para asegurar la integridad de la información clínica y eliminar los artefactos de inestabilidad inherentes al proceso de calibración y estabilización inicial del sensor, se implementó un filtro preventivo en la capa de la base de datos, ejecutado antes de la inserción definitiva de cada registro.

Este proceso es gestionado por el *trigger tgr_filtro_ruido*, el cual se configura para ejecutarse inmediatamente antes de la inserción de cada fila en la tabla mediciones. Dicho *trigger* invoca a la función *filtro_ruido_inicial*, que contiene la lógica de validación de estabilidad.

La función *filtro_ruido_inicial* actúa como un validador de estabilidad, evaluando si el registro ingresado cumple simultáneamente las dos condiciones que definen el "ruido de carga" que son los valores atípicos de ascenso lento de los datos biométricos:

- Saturación de Oxígeno: El valor es igual o inferior a 68%, SpO2<68.
- Ritmo Cardíaco: El valor es igual o inferior a 45, ritmo cardiaco<45.

Solo si ambas condiciones son verificadas como verdaderas, el registro es clasificado como artefacto inicial y la inserción se cancela. Este diseño asegura que solo se descarte el ruido de inicio de sesión, permitiendo la persistencia de lecturas reales que podrían ser bajas como bradicardia o hipoxemia severa, si superan al menos uno de los umbrales.

```sql
-- 1. FUNCIÓN: filtro_ruido_inicial() - Contiene la lógica de descarte.
CREATE OR REPLACE FUNCTION filtro_ruido_inicial()
RETURNS TRIGGER AS $$
DECLARE
    UMBRAL_SPO2_MINIMO INT := 68;
    UMBRAL_BPM_MINIMO INT := 45;

BEGIN
    IF NEW.spo2 <= UMBRAL_SPO2_MINIMO AND NEW.ritmo_cardiaco <= UMBRAL_BPM_MINIMO THEN

        RAISE NOTICE 'Noise filter activated.';

        RETURN NULL;
    END IF;

    RETURN NEW;
END;
$$ LANGUAGE plpgsql SECURITY DEFINER;


-- 2. TRIGGER: tgr_filtro_ruido - Se activa antes de la inserción en 'mediciones'.
CREATE OR REPLACE TRIGGER tgr_filtro_ruido
BEFORE INSERT ON public.mediciones
FOR EACH ROW EXECUTE FUNCTION filtro_ruido_inicial();
```

Figura 73, Función y trigger para el filtro de ruido inicial de datos biométricos. [Elaboración propia]

7.4.7.3. TABLA ALERTAS

La función de esta tabla es servir como el registro histórico y repositorio de eventos clínicos críticos. Se utiliza exclusivamente para almacenar las notificaciones de anomalías generadas automáticamente por el *trigger* de la base de datos. El campo *leida* y *tipo_alerta* permite al personal médico priorizar la atención y gestionar el seguimiento de estos incidentes.

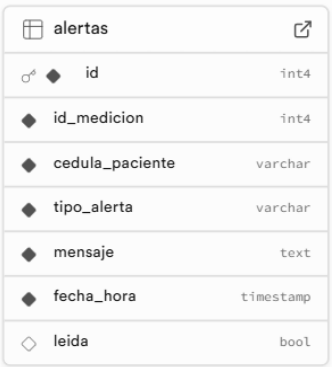


Figura 74, Estructura de la tabla Alertas: Repositorio de eventos clínicos críticos. [Elaboración propia]

La inserción y registro de datos en la tabla Alertas es gestionada y originada de manera automatizada y asíncrona a través del trigger t*rg_generar_alerta*.

Este *trigger* está definido para ejecutarse inmediatamente después de cada inserción de fila en la tabla Mediciones. El *trigger* invoca a la función *generar_alerta*, la cual actúa como un validador clínico en tiempo real.

La lógica contenida en esta función evalúa los valores recién ingresados y dispara la inserción en Alertas si se cumplen las siguientes condiciones:

- Alerta Roja: Se genera si la saturación de oxígeno (SpO2) es menor a 90% (indicando hipoxemia crítica).

- Alerta Amarilla: Se genera si el ritmo cardíaco (BPM) es mayor a 120 BPM (indicando taquicardia).

Este diseño convierte a la base de datos en la fuente de autoridad para la certificación de un evento crítico, asegurando que el registro de la alerta sea inmediato y preciso.

```
CREATE OR REPLACE FUNCTION generar_alerta()
RETURNS TRIGGER AS $$
BEGIN
    -- Alerta roja si SpO₂ < 90%
    IF NEW.spo2 < 90 THEN
        INSERT INTO alertas(id_medicion, cedula_paciente, tipo_alerta, mensaje, fecha_hora)
        VALUES (NEW.id, NEW.cedula_paciente, 'ROJA', 'SpO₂ por debajo de 90%', NEW.fecha_hora);
    END IF;

    -- Alerta amarilla si ritmo cardíaco > 120
    IF NEW.ritmo_cardiaco > 120 THEN
        INSERT INTO alertas(id_medicion, cedula_paciente, tipo_alerta, mensaje, fecha_hora)
        VALUES (NEW.id, NEW.cedula_paciente, 'AMARILLA', 'Ritmo cardíaco mayor a 120 bpm', NEW.fech
    END IF;

    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER trg_generar_alerta
AFTER INSERT ON mediciones
FOR EACH ROW
EXECUTE FUNCTION generar_alerta();
```

Figura 75, Función y trigger para la automatización de las alertas clínicas. [Elaboración propia]

7.4.7.4. INTERCONEXIÓN DEL ESQUEMA RELACIONAL

El diseño relacional logra su máxima funcionalidad al establecer una cadena de dependencia clara a través de las llaves foráneas. La tabla Usuarios actúa como el punto de origen de la identidad y la seguridad del sistema. Desde allí, la tabla Mediciones se ancla al paciente mediante su cédula, garantizando que cada valor fisiológico registrado sea trazable. Finalmente, la tabla Alertas se encuentra en el nivel más dependiente, ya que cada registro de anomalía debe hacer referencia a una medición específica (*id_mediciones*). Este anclaje jerárquico es lo que permite que el mecanismo de eliminación en cascada mantenga la integridad absoluta de la base de datos, reflejando con precisión la actividad y la historia clínica de cada usuario.

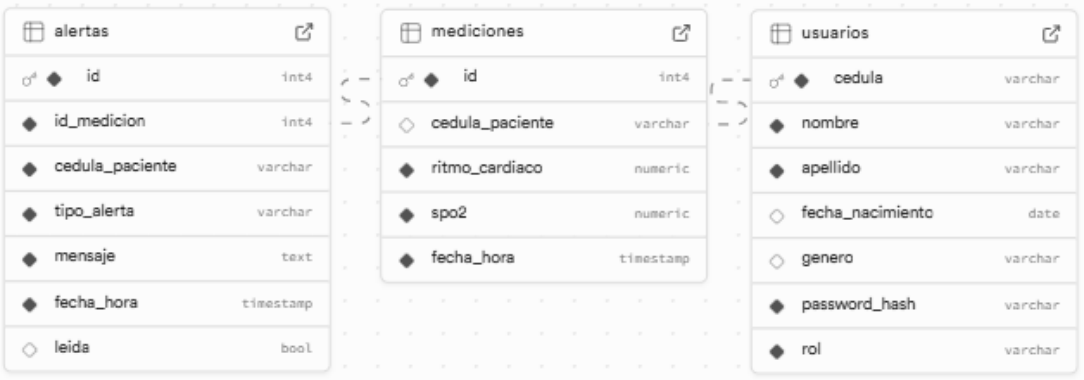


Figura 76, Diagrama Entidad-Relación del esquema de la base de datos y sus interconexiones. [Elaboración propia]

7.4.8. DESARROLLO DE LA INTERFAZ DEL SISTEMA

El sistema de pulsioximetría desarrollado se diseñó no sólo como una herramienta de medición biomédica, sino también como una plataforma integral de gestión de información médica. Para cumplir con este objetivo, se construyó una interfaz compuesta por un backend eficiente y un frontend

accesible y amigable, que permiten la interacción con los datos obtenidos a través del dispositivo ESP32 y el sensor MAX30102.

7.4.8.1. DESARROLLO DEL BACKEND

La tercera capa de la arquitectura del sistema está compuesta por el Backend, implementado en Python utilizando el framework FastAPI. Este módulo central, desplegado en un servidor en Render, actúa como el cerebro lógico para la gestión, autenticación, autorización y consulta de datos. La elección de Render como plataforma de despliegue se justifica por su capacidad para manejar peticiones de forma concurrente y por asegurar que la lógica crucial del sistema se mantenga operativa y escalable en la nube, independientemente del entorno de desarrollo local.

El Backend es el responsable de recibir, validar, procesar y almacenar información sensible, como los perfiles de los usuarios y las mediciones. Esto se logra a través de la definición de endpoints RESTful, lo que facilita la interacción segura para la consulta de mediciones y para todas las funciones de gestión de usuarios y alertas. La construcción con FastAPI aprovecha su rapidez, la integración con OpenAPI para la documentación automática y la validación estricta de datos mediante Pydantic, lo que garantiza la integridad de la información registrada, especialmente en el momento del registro de usuarios.

La seguridad del sistema se aborda de manera sólida a través de un módulo de autenticación dedicado. Este módulo gestiona el registro e inicio de sesión de usuarios, utilizando el algoritmo bcrypt para el hashing seguro de contraseñas. De esta forma, se garantiza que la información sensible de acceso nunca se almacene en texto plano en la base de datos de Supabase.

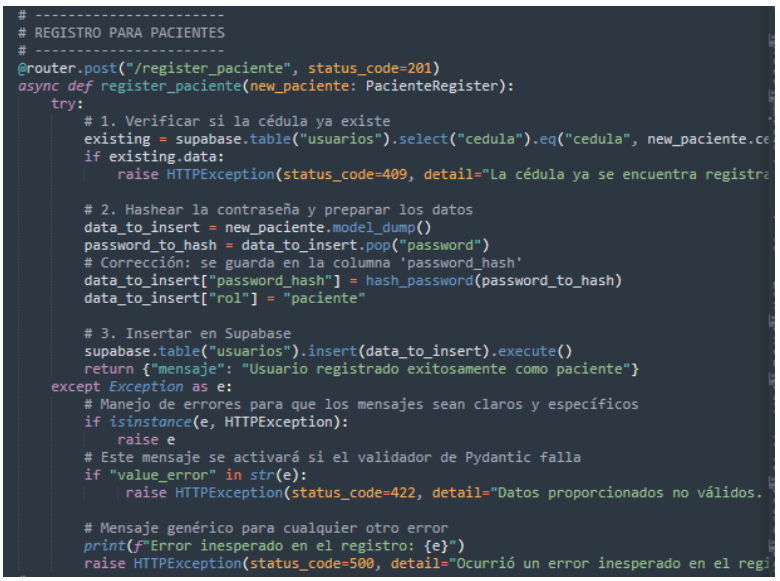
```python
# ----------------------
# LOGIN
# ----------------------
@router.post("/login")
async def login(form_data: OAuth2PasswordRequestForm = Depends()):
    try:
        cedula = int(form_data.username)
    except ValueError:
        raise HTTPException(status_code=400, detail="La cédula debe ser numérica")

    result = supabase.table("usuarios").select("*").eq("cedula", cedula).execute()
    if not result.data:
        raise HTTPException(status_code=400, detail="Usuario no encontrado")

    user = result.data[0]
    if not verify_password(form_data.password, user["password_hash"]):
        raise HTTPException(status_code=400, detail="Contraseña incorrecta")

    access_token = create_access_token({"sub": user["cedula"], "rol": user["rol"]})
    return {"access_token": access_token, "token_type": "bearer"}
```

Figura 77, Endpoint para la autenticación y generación de token JWT para el inicio de sesión. [Elaboración propia]



```python
# -----------------------
# REGISTRO PARA PACIENTES
# -----------------------
@router.post("/register_paciente", status_code=201)
async def register_paciente(new_paciente: PacienteRegister):
    try:
        # 1. Verificar si la cédula ya existe
        existing = supabase.table("usuarios").select("cedula").eq("cedula", new_paciente.ce
        if existing.data:
            raise HTTPException(status_code=409, detail="La cédula ya se encuentra registra

        # 2. Hashear la contraseña y preparar los datos
        data_to_insert = new_paciente.model_dump()
        password_to_hash = data_to_insert.pop("password")
        # Corrección: se guarda en la columna 'password_hash'
        data_to_insert["password_hash"] = hash_password(password_to_hash)
        data_to_insert["rol"] = "paciente"

        # 3. Insertar en Supabase
        supabase.table("usuarios").insert(data_to_insert).execute()
        return {"mensaje": "Usuario registrado exitosamente como paciente"}
    except Exception as e:
        # Manejo de errores para que los mensajes sean claros y específicos
        if isinstance(e, HTTPException):
            raise e
        # Este mensaje se activará si el validador de Pydantic falla
        if "value_error" in str(e):
            raise HTTPException(status_code=422, detail="Datos proporcionados no válidos.

        # Mensaje genérico para cualquier otro error
        print(f"Error inesperado en el registro: {e}")
        raise HTTPException(status_code=500, detail="Ocurrió un error inesperado en el regi
```

Figura 78, Endpoint de estructura de datos para el registro de usuarios nuevos. [Elaboración propia]

Una vez autenticado, el sistema emite Tokens de Web JSON, que no solo verifican la identidad del usuario en cada solicitud, sino que también transportan la información de autorización relativa al rol. Esta información es crucial para la estructura modular del backend, diseñada para ser aprovechada mediante una lógica estricta en cada endpoint que habilita o restringe el acceso a las funcionalidades. Así, el sistema garantiza una separación clara de responsabilidades, estableciendo la organización de la funcionalidad en los siguientes módulos principales.

7.4.8.1.1. MÓDULO DE MEDICIONES

Este módulo se enfoca en la gestión y consulta de los registros de saturación de oxígeno y frecuencia cardíaca que han sido previamente ingresados y persistidos en la base de datos. Su función primordial es la consulta de registros históricos, y la accesibilidad a esta información se jerarquiza estrictamente por rol para asegurar la privacidad y el cumplimiento normativo. Los pacientes pueden visualizar su propio historial para el seguimiento personal; los médicos tienen acceso a los datos de todos los pacientes registrados en el sistema, lo cual es esencial para el análisis clínico y diagnóstico; y los administradores poseen acceso irrestricto a la totalidad de los datos para fines de auditoría y gestión sistémica. Las operaciones de actualización de mediciones están estrictamente limitadas a los roles de médico y administrador, ya que requieren una validación clínica o de auditoría para corregir cualquier anomalía. Por último, la eliminación definitiva de mediciones está autorizada únicamente para el rol de administrador, debido a que esta acción puede comprometer la trazabilidad histórica y la integridad de los registros clínicos, y solo debe ser ejecutada por el personal con la máxima autoridad sobre la gestión del sistema de datos.



```python
# -----------------------
# ELIMINAR MEDICIONES
# -----------------------
@router.delete("/mediciones/{id}")
async def delete_medicion(id: int, token: str = Depends(oauth2_scheme)):
    usuario = decode_token(token)
    if not usuario:
        raise HTTPException(status_code=401, detail="No autenticado")

    rol_actual = usuario["rol"]
    if rol_actual != "admin":
        raise HTTPException(status_code=403, detail="No autorizado para eliminar registros")

    result = supabase.table("mediciones").select("id").eq("id", id).execute()
    if not result.data:
        raise HTTPException(status_code=404, detail="Medición no encontrada")

    supabase.table("mediciones").delete().eq("id", id).execute()
    return {"mensaje": f"Medición con id {id} eliminada exitosamente"}
```

Figura 79, Endpoint para la eliminación de registros de mediciones. [Elaboración propia]



```python
# -----------------------
# ACTUALIZAR MEDICIONES
# -----------------------
@router.put("/mediciones/{id}")
async def update_medicion(id: int, datos: dict, token: str = Depends(oauth2_scheme)):
    usuario = decode_token(token)
    if not usuario:
        raise HTTPException(status_code=401, detail="No autenticado")

    rol = usuario["rol"]
    cedula = usuario["sub"]

    if rol == "paciente":
        raise HTTPException(status_code=403, detail="No autorizado para actualizar mediciones")

    campos_prohibidos = ["id", "cedula_paciente"]
    for campo in campos_prohibidos:
        if campo in datos:
            raise HTTPException(status_code=400, detail=f"No se puede actualizar el campo '{campo}

    result = supabase.table("mediciones").select("*").eq("id", id).execute()
    if not result.data:
        raise HTTPException(status_code=404, detail="Medición no encontrada")

    medicion = result.data[0]

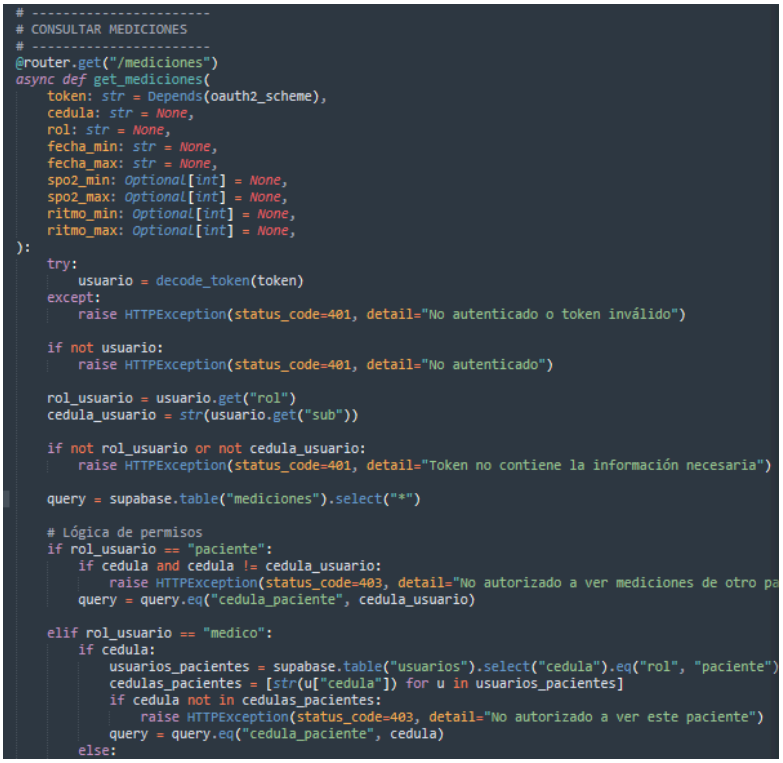
    if rol == "medico":
        if medicion["cedula_paciente"] != cedula and supabase.table("usuarios").select("rol").eq("
            raise HTTPException(status_code=403, detail="No autorizado para actualizar esta medici

    campos_permitidos = ["ritmo_cardiaco", "spo2", "fecha_hora"]
    datos_filtrados = {k: v for k, v in datos.items() if k in campos_permitidos}

    if not datos_filtrados:
        raise HTTPException(status_code=400, detail="No hay campos válidos para actualizar")

    supabase.table("mediciones").update(datos_filtrados).eq("id", id).execute()
    return {"mensaje": "Medición actualizada exitosamente"}
```

Figura 80, Endpoint para la actualización de datos de mediciones biométricas. [Elaboración propia]



```python
# ------------------------
# CONSULTAR MEDICIONES
# ------------------------
@router.get("/mediciones")
async def get_mediciones(
    token: str = Depends(oauth2_scheme),
    cedula: str = None,
    rol: str = None,
    fecha_min: str = None,
    fecha_max: str = None,
    spo2_min: Optional[int] = None,
    spo2_max: Optional[int] = None,
    ritmo_min: Optional[int] = None,
    ritmo_max: Optional[int] = None,
):
    try:
        usuario = decode_token(token)
    except:
        raise HTTPException(status_code=401, detail="No autenticado o token inválido")

    if not usuario:
        raise HTTPException(status_code=401, detail="No autenticado")

    rol_usuario = usuario.get("rol")
    cedula_usuario = str(usuario.get("sub"))

    if not rol_usuario or not cedula_usuario:
        raise HTTPException(status_code=401, detail="Token no contiene la información necesaria")

    query = supabase.table("mediciones").select("*")

    # Lógica de permisos
    if rol_usuario == "paciente":
        if cedula and cedula != cedula_usuario:
            raise HTTPException(status_code=403, detail="No autorizado a ver mediciones de otro pa
        query = query.eq("cedula_paciente", cedula_usuario)

    elif rol_usuario == "medico":
        if cedula:
            usuarios_pacientes = supabase.table("usuarios").select("cedula").eq("rol", "paciente")
            cedulas_pacientes = [str(u["cedula"]) for u in usuarios_pacientes]
            if cedula not in cedulas_pacientes:
                raise HTTPException(status_code=403, detail="No autorizado a ver este paciente")
            query = query.eq("cedula_paciente", cedula)
        else:
```

Figura 81, Endpoint para la consulta jerarquizada de mediciones históricas. [Elaboración propia]

7.4.8.1.2. MÓDULO DE USUARIOS

Este módulo es crítico para la seguridad y la estructura administrativa, gestionando el ciclo de vida completo de las cuentas y la jerarquía de roles dentro del sistema. De forma básica, todos los usuarios tienen permitido consultar y actualizar sus propios datos personales. Sin embargo, las funcionalidades de gestión de terceros se encuentran jerarquizadas de forma estricta: los pacientes están limitados exclusivamente a la administración de su propio perfil. Por su parte, los médicos poseen una visibilidad más amplia, lo que les permite consultar la información de todos los pacientes bajo su supervisión y actualizar los datos personales de estos. Adicionalmente, los médicos están autorizados a crear nuevos usuarios, aunque restringiendo esta capacidad a la asignación exclusiva del rol de paciente, y no tienen permitido eliminar ninguna cuenta existente. Finalmente, el administrador ejerce un control irrestricto sobre la estructura completa, con la potestad de ver y editar todos los perfiles, crear cualquier tipo de usuario nuevo incluidos médicos y otros administradores, y eliminar usuarios definitivamente, asegurando el control centralizado de la arquitectura de accesos del sistema.



```python
# ------------------------
# CREAR USUARIOS
# ------------------------
@router.post("/usuarios")
async def create_usuario(new_user: dict, token: str = Depends(oauth2_scheme)):
    usuario = decode_token(token)
    if not usuario:
        raise HTTPException(status_code=401, detail="No autenticado")

    rol_actual = usuario["rol"]

    if rol_actual == "paciente" or (rol_actual == "medico" and new_user.get("rol") != "paciente"):
        raise HTTPException(status_code=403, detail="No autorizado para crear este usuario")

    existing = supabase.table("usuarios").select("*").eq("cedula", new_user["cedula"]).execute()
    if existing.data:
        raise HTTPException(status_code=400, detail="Cédula ya registrada")

    new_user["password_hash"] = hash_password(new_user.pop("password"))
    supabase.table("usuarios").insert(new_user).execute()
    return {"mensaje": "Usuario creado exitosamente"}
```

Figura 82, Endpoint para la creación de usuarios y asignación de roles. [Elaboración propia]



```python
# ----------------------
# ACTUALIZAR USUARIO
# ----------------------
@router.put("/usuarios/{cedula_actualizar}")
async def update_usuario(cedula_actualizar: str, datos: dict, token: str = Depends(oauth2_scheme)
    usuario = decode_token(token)
    if not usuario:
        raise HTTPException(status_code=401, detail="No autenticado")

    rol_actual = usuario["rol"]
    cedula = usuario["sub"]

    if rol_actual == "paciente" and cedula != cedula_actualizar:
        raise HTTPException(status_code=403, detail="No autorizado")

    if rol_actual == "medico":
        result = supabase.table("usuarios").select("rol").eq("cedula", cedula_actualizar).execute
        if not result.data:
            raise HTTPException(status_code=404, detail="Usuario a actualizar no encontrado")
        rol_target = result.data[0]["rol"]
        if cedula_actualizar != cedula and rol_target != "paciente":
            raise HTTPException(status_code=403, detail="No autorizado para actualizar este usuar
        if "rol" in datos:
            raise HTTPException(status_code=403, detail="No autorizado para cambiar el rol de un

    if rol_actual == "paciente" and "rol" in datos:
        raise HTTPException(status_code=403, detail="No autorizado para cambiar el rol de un usua

    if "cedula" in datos:
        nueva_cedula = datos["cedula"]
        existente = supabase.table("usuarios").select("*").eq("cedula", nueva_cedula).execute().d
        if existente and existente[0]["cedula"] != cedula_actualizar:
            raise HTTPException(status_code=400, detail="La cédula que intenta asignar ya está re

    if "password" in datos:
        datos["password_hash"] = hash_password(datos.pop("password"))

    supabase.table("usuarios").update(datos).eq("cedula", cedula_actualizar).execute()
    return {"mensaje": "Usuario actualizado exitosamente"}
```

Figura 83, Endpoint para la actualización de perfiles y datos personales. [Elaboración propia]



```python
# ----------------------
# ELIMINAR USUARIO
# ----------------------
@router.delete("/usuarios/{cedula_eliminar}")
async def delete_usuario(cedula_eliminar: int, token: str = Depends(oauth2_scheme)):
    usuario = decode_token(token)
    if not usuario:
        raise HTTPException(status_code=401, detail="No autenticado")

    rol_actual = usuario["rol"]
    if rol_actual != "admin":
        raise HTTPException(status_code=403, detail="No autorizado")

    # Paso 1: Verificar si el usuario existe
    result = supabase.table("usuarios").select("cedula").eq("cedula", cedula_eliminar).execute()

    # Paso 2: Si la lista de datos está vacía, el usuario no existe.
    if not result.data:
        raise HTTPException(status_code=404, detail="Usuario no encontrado")

    # Paso 3: Si el usuario existe, se procede con la eliminación
    supabase.table("usuarios").delete().eq("cedula", cedula_eliminar).execute()
    return {"mensaje": "Usuario eliminado exitosamente"}
```

Figura 84, Endpoint para la eliminación controlada de perfiles de usuario. [Elaboración propia]

7.4.8.1.3. MÓDULO DE ALERTAS

Diseñado para la gestión proactiva de eventos críticos, este módulo implementa la lógica necesaria para generar y gestionar las alertas desencadenadas automáticamente por mediciones de SpO2 o BPM que superan o caen por debajo de los umbrales clínicos preestablecidos. La consulta inmediata de alertas activas es esencial para el personal sanitario. Por ello, solo los usuarios con rol de médico están autorizados para realizar la actualización del estado de una alerta, por ejemplo, marcándola como "leída", asegurando que solo el personal con la autoridad clínica pueda validar su seguimiento. Adicionalmente, la eliminación definitiva de los registros de alerta está reservada exclusivamente para el administrador, lo que permite mantener la trazabilidad histórica de los eventos, mientras se asegura la depuración controlada de registros redundantes o erróneos del sistema.

```python
# ----------------------
# CONSULTAR ALERTAS
# ----------------------
@router.get("/alertas")
async def get_alertas(
    token: str = Depends(oauth2_scheme),
    cedula: str = None,
    tipo_alerta: str = None,
    fecha_min: str = None,
    fecha_max: str = None,
    leida: Optional[bool] = None
):
    usuario = decode_token(token)
    if not usuario:
        raise HTTPException(status_code=401, detail="No autenticado")

    rol_usuario = usuario["rol"]
    cedula_usuario = str(usuario["sub"])

    # Iniciar la consulta a la tabla de alertas
    query = supabase.table("alertas").select("*")

    # Lógica de permisos y filtros por cédula
    if rol_usuario == "paciente":
        # Se añade un control explícito para evitar que los pacientes filtren por cédula
        if cedula and cedula != cedula_usuario:
            raise HTTPException(status_code=403, detail="No tiene permiso para filtrar por
        query = query.eq("cedula_paciente", cedula_usuario)
    elif rol_usuario in ["medico", "admin"]:
        if cedula:
            query = query.eq("cedula_paciente", cedula)
```

Figura 85, Endpoint para la consulta de alertas críticas activas. [Elaboración propia]

```python
# ----------------------
# ACTUALIZAR ALERTA
# ----------------------
class AlertaUpdate(BaseModel):
    leida: bool

@router.put("/alertas/{id}")
async def update_alerta(id: int, datos: AlertaUpdate, token: str = Depends(oauth2_scheme)):
    usuario = decode_token(token)
    if not usuario:
        raise HTTPException(status_code=401, detail="No autenticado")

    rol = usuario["rol"]
    if rol not in ["medico"]:
        raise HTTPException(status_code=403, detail="No autorizado para actualizar alertas")

    result = supabase.table("alertas").select("id").eq("id", id).execute()
    if not result.data:
        raise HTTPException(status_code=404, detail="Alerta no encontrada")

    supabase.table("alertas").update({"leida": datos.leida}).eq("id", id).execute()
    return {"mensaje": f"Alerta con ID {id} actualizada exitosamente."}
```

Figura 86, Endpoint para la modificación del estado de la alerta. [Elaboración propia]

```python
# ----------------------
# ELIMINAR ALERTA
# ----------------------
@router.delete("/alertas/{id}")
async def delete_alerta(id: int, token: str = Depends(oauth2_scheme)):
    usuario = decode_token(token)
    if not usuario:
        raise HTTPException(status_code=401, detail="No autenticado")

    rol_actual = usuario["rol"]
    if rol_actual != "admin":
        raise HTTPException(status_code=403, detail="No autorizado para eliminar alertas"

    result = supabase.table("alertas").select("id").eq("id", id).execute()
    if not result.data:
        raise HTTPException(status_code=404, detail="Alerta no encontrada")

    supabase.table("alertas").delete().eq("id", id).execute()
    return {"mensaje": f"Alerta con ID {id} eliminada exitosamente"}
```

Figura 87, Endpoint para la eliminación definitiva de registros de alerta. [Elaboración propia]

7.4.8.1.4. MECANISMOS DE ROBUSTEZ Y EXPERIENCIA DE USUARIO

La solidez del *backend* se complementa con la implementación de rigurosos mecanismos de validación de datos y manejo de errores estandarizado en todas las interacciones que gestiona. La construcción sobre FastAPI permite la validación estricta a través de *Pydantic* y la captura centralizada de excepciones.

Por ejemplo, al intentar un registro o una actualización de perfil, el sistema verifica la unicidad de la cédula. Si un usuario intenta registrarse con una cédula ya existente, el *backend* interrumpe la ejecución del proceso y devuelve una respuesta estructurada con el código de estado 400 Bad Request y el detalle específico del error "Cédula ya registrada". De manera similar, si se violan las reglas de formato, como intentar iniciar sesión con una cédula no numérica, se devuelve un código de estado específico, asegurando que el *frontend* reciba información clara para el usuario final.

Finalmente, para optimizar la eficiencia y la experiencia del usuario durante la consulta de datos, el *backend* incorpora filtros avanzados en sus *endpoints* de consulta, como:

- Consulta de Mediciones: Filtros por SpO2, Ritmo Cardíaco y rangos de fecha
- Consulta de Alertas: Filtros por estado de lectura, tipo de alerta y rangos de fecha.

Estos parámetros permiten al *frontend* realizar peticiones específicas y acotadas a la base de datos, lo que resulta en una optimización crítica del rendimiento de las consultas y permite a los usuarios sanitarios ubicar rápidamente la información crítica dentro del historial extenso de los pacientes.

7.4.8.2. DESARROLLO DEL FRONTEND

Finalmente, la cuarta capa del sistema está constituida por el Frontend, desarrollado en React.js. Esta elección tecnológica se fundamenta en la necesidad de mantener un código organizado, escalable y con alta capacidad de renderización dinámica, aprovechando la estructura basada en componentes reutilizables.

Para la publicación y operatividad del sistema, se tomó la decisión crítica de desplegar la aplicación en la plataforma Vercel. Esta estrategia no solo simplifica el flujo de trabajo, sino que asegura el despliegue continuo de la interfaz, lo cual facilita la implementación rápida de actualizaciones, y garantiza la disponibilidad global de la aplicación, permitiendo a los usuarios acceder a la información de manera rápida y constante, sin interrupciones.

La interfaz de usuario se diseñó para ser intuitiva, priorizando la claridad en la presentación de la información médica. Este componente es el responsable de traducir los datos recibidos del *backend* en representaciones visuales intuitivas. Para ello, se utilizaron componentes que destacan las métricas clave y facilitan la interpretación de los datos por parte de los usuarios. El diseño se basó en principios de usabilidad, buscando reducir la curva de aprendizaje y minimizar los errores de interacción. Como resultado de este enfoque, se diseñaron paneles diferenciados según los roles definidos en el sistema: paciente, médico y administrador.

La interacción segura con el sistema inicia con la implementación de formularios de autenticación y registro que incluyen validación de campos en tiempo real, asegurando que los datos ingresados sean correctos antes de ser enviados al *backend*. La comunicación con el *backend* se establece mediante peticiones HTTP asíncronas utilizando Axios. Cada solicitud de datos incorpora el *token* JWT correspondiente, el cual se valida en el *backend* antes de permitir el acceso a la información. Esta integración asegura la coherencia y seguridad de la información a lo largo de todo el sistema. Una vez autenticados, los usuarios acceden a un *dashboard* que se adapta según su rol, lo que garantiza una sincronización constante entre la interfaz y la información médica centralizada.

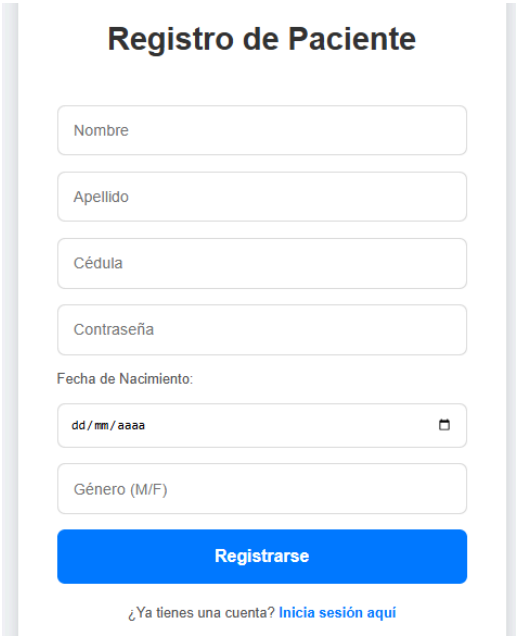


Figura 88, Interfaz de inicio de sesión de usuarios. [Elaboración propia]



Figura 89, Interfaz de registro para nuevos usuarios pacientes. [Elaboración propia]

7.4.8.2.1. PANEL DEL PACIENTE

El *Dashboard* del Paciente es una vista concisa y personal enfocada en el autoservicio y la monitorización individual. Muestra el perfil de usuario, incluyendo un botón dedicado a la edición de sus datos personales básicos. La sección principal se centra en el monitoreo de su estado de salud, ofreciendo una gráfica de tendencias con sus mediciones históricas de SpO2 y BPM. Este panel también presenta las mediciones más recientes. El seguimiento de eventos críticos se realiza a través de la lista de Alertas asociadas a su historial, donde se presentan aquellas mediciones que han caído fuera de los rangos clínicos normales. Cada alerta en la lista detalla la fecha, el ID de la medición, el código de la alerta y su estado de lectura por parte del personal médico, permitiendo al paciente un seguimiento rápido y personal de cualquier evento.



Figura 90, Dashboard del paciente: perfil de usuario y gráfica de tendencias biométricas. [Elaboración propia]

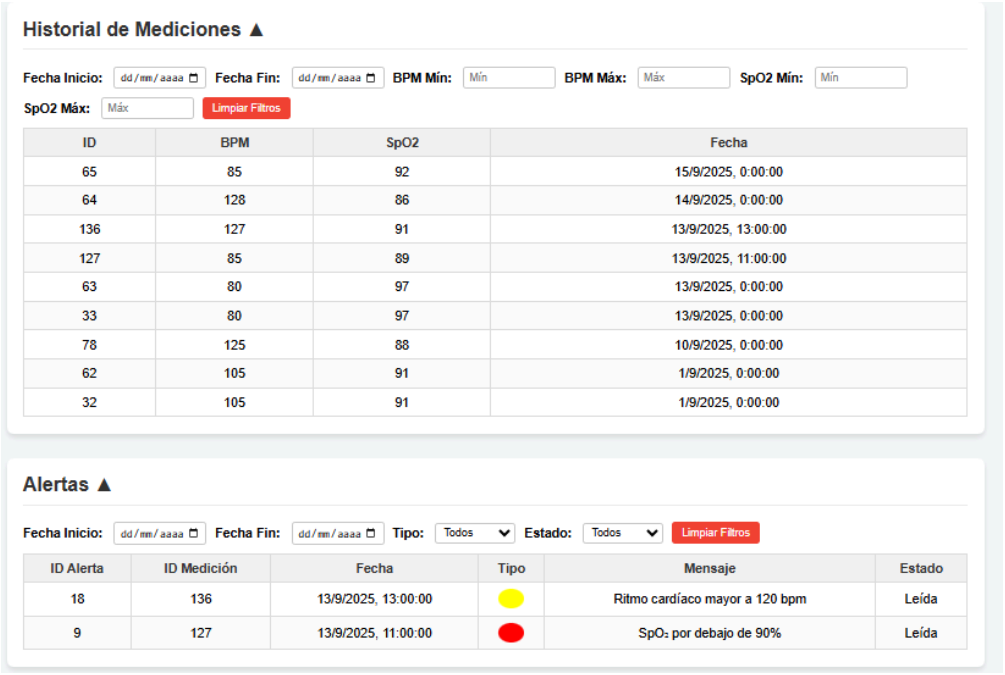


Figura 91, Dashboard del paciente: visualización del historial de mediciones recientes y listado de alertas. [Elaboración propia]

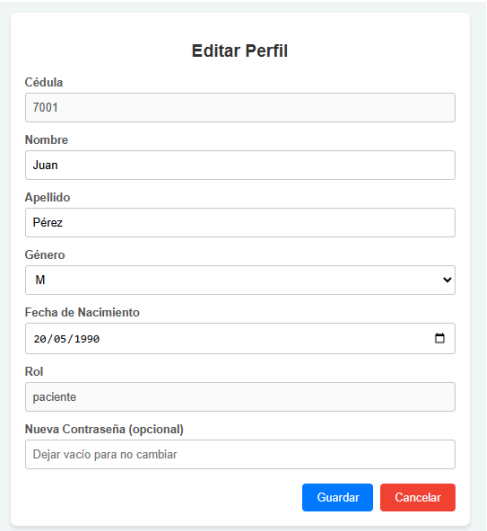


Figura 92, Panel de interfaz para la edición de datos personales del paciente. [Elaboración propia]

7.4.8.2.2. PANEL DEL MÉDICO

El *Dashboard* del Médico está diseñado para la gestión clínica y el análisis de la población bajo su cuidado. Al ingresar, el profesional visualiza su perfil y una tabla que lista a todos los usuarios asignados con el rol de Paciente. Este panel incluye la funcionalidad de creación de cuentas nuevas, restringida a la asignación exclusiva del rol de paciente, y un botón de acción que permite la edición de datos personales tanto en su propio perfil como en el de los pacientes bajo su supervisión. Al seleccionar un paciente, el médico accede a un dashboard de detalle que no solo facilita el análisis de la gráfica de mediciones, las alertas y el historial completo, sino que también habilita funcionalidades de edición clínica: puede actualizar la información básica del paciente, modificar los registros de mediciones existentes específicamente los valores de SpO2 y BPM y cambiar el estado de las alertas marcándolas como "leídas" o "no leídas". Estas capacidades de edición, sujetas a las restricciones de rol definidas en el *backend*, son esenciales para la corrección de anomalías, la validación clínica de los datos y el seguimiento de la trazabilidad de los eventos críticos, facilitando la identificación de tendencias clínicas y la toma de decisiones diagnósticas.

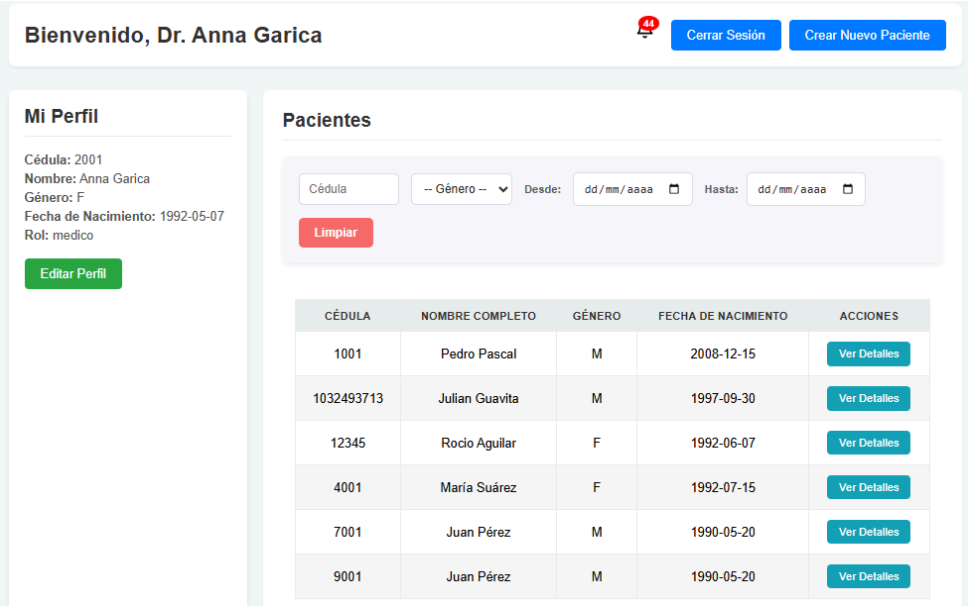


Figura 93, Dashboard del médico: vista general de gestión y listado de pacientes bajo supervisión. [Elaboración propia]

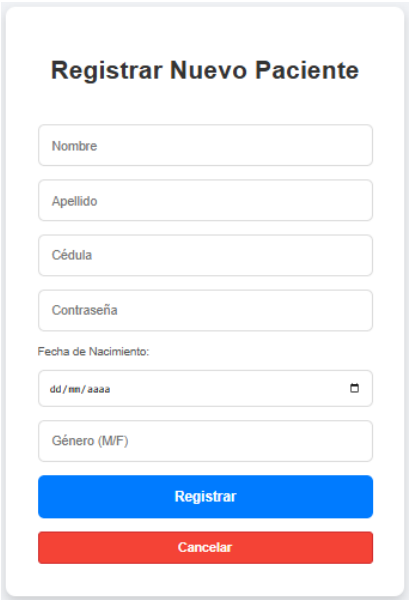


Figura 94, Dashboard del médico: panel de creación de nuevos usuarios. [Elaboración propia]



Figura 95, Dashboard médico: perfil de paciente y visualización de gráficas de tendencias clínicas. [Elaboración propia]

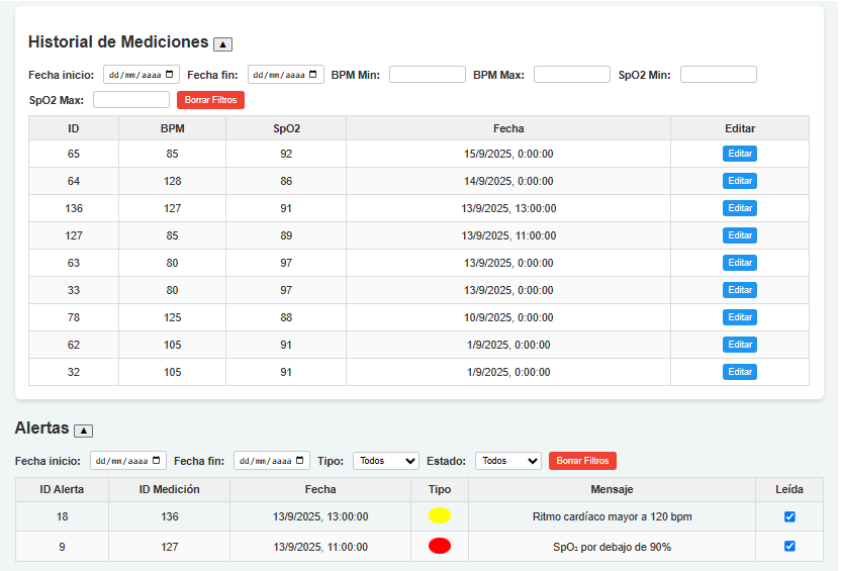


Figura 96, Dashboard médico: historial de mediciones del paciente y listado de alertas. [Elaboración propia]

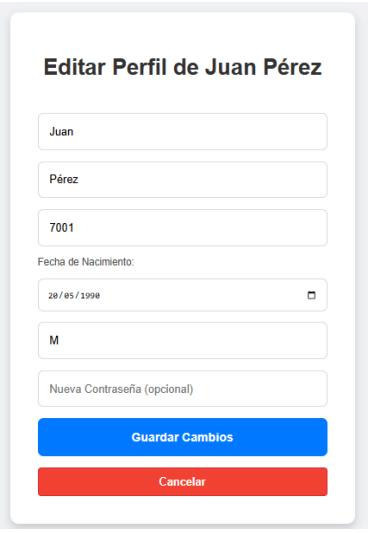


Figura 97, Panel de edición del perfil de un paciente por el médico. [Elaboración propia]

7.4.8.2.3. PANEL DEL ADMINISTRADOR

El *Dashboard* del administrador es esencialmente una versión de gestión del panel del médico, que opera con el nivel máximo de permisos del sistema. Dispone de su perfil, la tabla de todos los usuarios registrados, y el acceso a la creación de nuevas cuentas incluyendo todos los roles. El administrador posee acceso irrestricto para la edición de perfiles de cualquier usuario, sin las restricciones de rol aplicadas al médico o paciente. Adicionalmente, hereda todas las facultades de edición clínica del médico: puede modificar los valores de SpO2 y BPM en los registros de mediciones. La diferencia crítica radica en la presencia de botones de acción adicionales que habilitan las funcionalidades más sensibles: se incluyen botones para la eliminación definitiva de registros de mediciones, para la eliminación de alertas y para la eliminación de usuarios, asegurando que solo el personal con la máxima autoridad pueda ejecutar estas operaciones que afectan la integridad de los datos y la estructura del sistema.

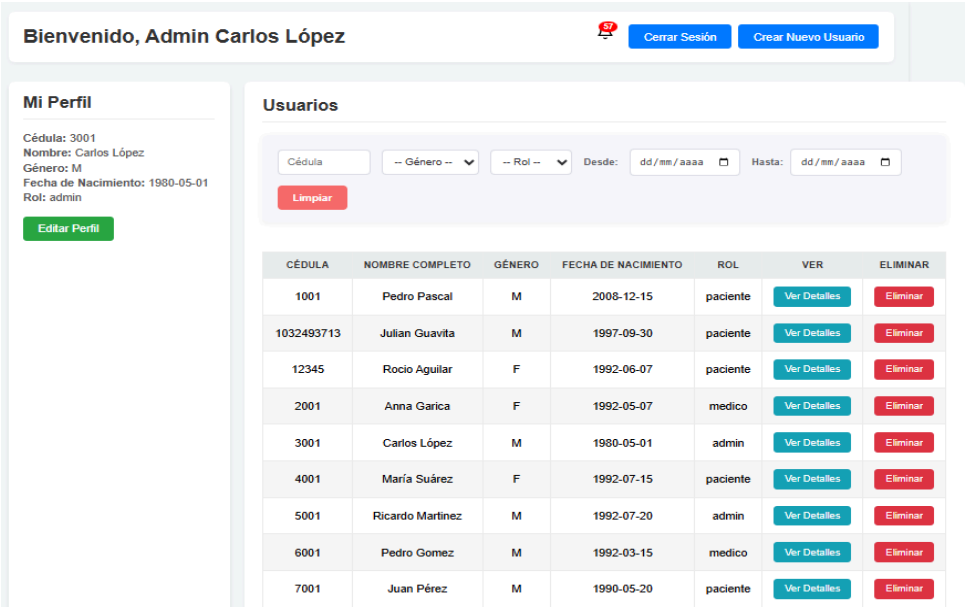


Figura 98, Dashboard del administrador: vista general de gestión de todos los usuarios. [Elaboración propia]

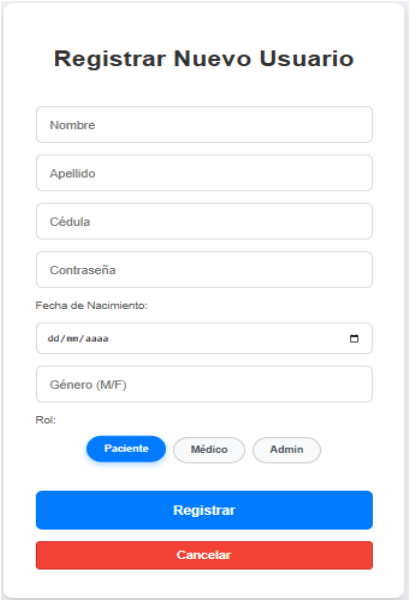


Figura 99, Dashboard del administrador: panel de creación de nuevos usuarios. [Elaboración propia]



Figura 100, Dashboard del administrador: perfil de usuario y visualización de gráficas de tendencias clínicas. [Elaboración propia]

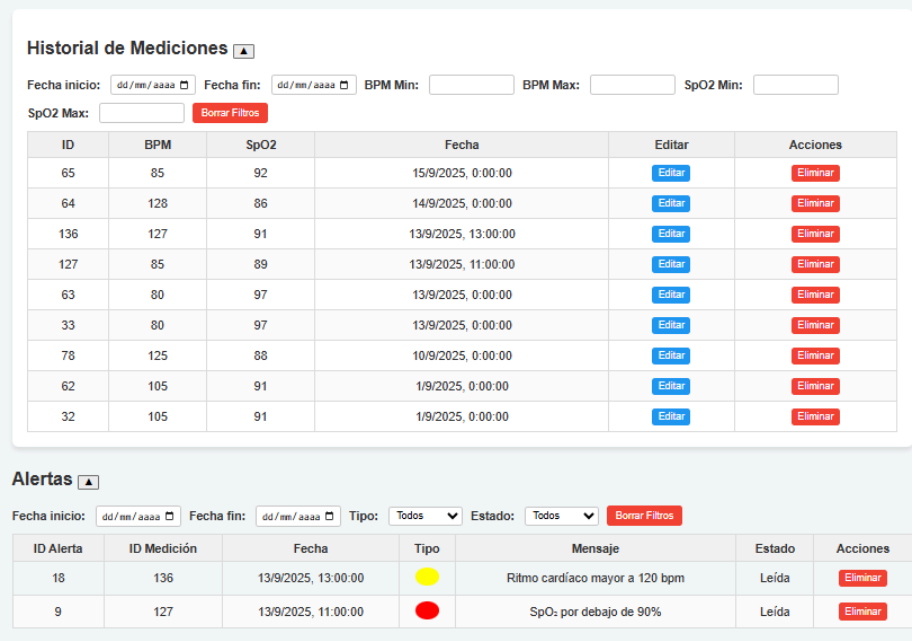


Figura 101, Dashboard del administrador: historial de mediciones de usuarios y listado de alertas. [Elaboración propia]

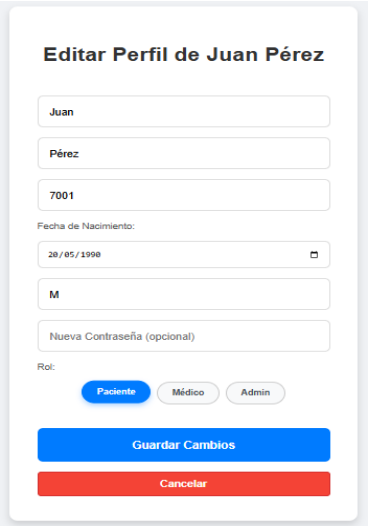


Figura 102, Panel de edición del perfil de un usuario por un administrador. [Elaboración propia]

7.4.9. INTEGRACIÓN TECNOLÓGICA Y FLUJO DE DATOS

El sistema se construyó sobre un stack de tecnologías modernas y de alto rendimiento, diseñado para garantizar la velocidad y la escalabilidad. La arquitectura se organiza bajo un modelo distribuido compuesto por cuatro pilares: un dispositivo de adquisición que utiliza el microcontrolador ESP32 con sensor MAX30102; una base de datos centralizada gestionada por Supabase con PostgreSQL; un módulo de lógica y seguridad implementado en el *backend* con FastAPI desplegado en Render; y una interfaz de presentación desarrollada en el *frontend* con React desplegado en Vercel. Esta configuración asegura que la gestión y el procesamiento de la información se mantengan operativos en la nube, optimizando la concurrencia y la disponibilidad del servicio.

7.4.9.1. CICLO 1: REGISTRO Y PREPARACIÓN PARA LA INGESTA

El primer ciclo, denominado registro y preparación para la ingesta, se articula en torno a una estricta dependencia de autenticación. Es crucial que el usuario sea registrado y autenticado a través de la interfaz web compuesta por el *frontend* y *backend*, asegurando que sus datos sean almacenados correctamente. Esta dependencia garantiza la integridad referencial, dado que la cédula del usuario debe existir en la tabla principal antes de que cualquier medición se vincule a ella como llave foránea. Cualquier intento de enviar datos biométricos con una cédula no registrada resultará en un error directo de la base de datos, impidiendo la escritura.

El proceso de registro de mediciones inicia cuando el usuario selecciona el modo Online en el menú principal y configura la conexión inalámbrica. Una vez conectado, el usuario introduce su número de identificación utilizando el teclado. En este punto, el dispositivo realiza una consulta de seguridad a la base de datos central para confirmar que la identificación es válida. Si la identificación es válida, el sistema procede sin interrupción a la fase de adquisición de datos. Por el contrario, si la identificación no es reconocida, el sistema bloquea la medición y el envío de datos. En su lugar, presenta una pantalla de error que permite al usuario reintentar el ingreso de la identificación o reiniciar el dispositivo para volver al menú de inicio. Solo tras la validación exitosa de la identificación, el dispositivo inicia la medición de saturación de oxígeno y frecuencia cardíaca. Estos valores, ya vinculados a un usuario confirmado, son enviados directamente a la base de datos remota. La base de datos recibe los datos y, a través de sus procesos automáticos de triggers, genera alertas si los valores clínicos caen fuera de los rangos preestablecidos.

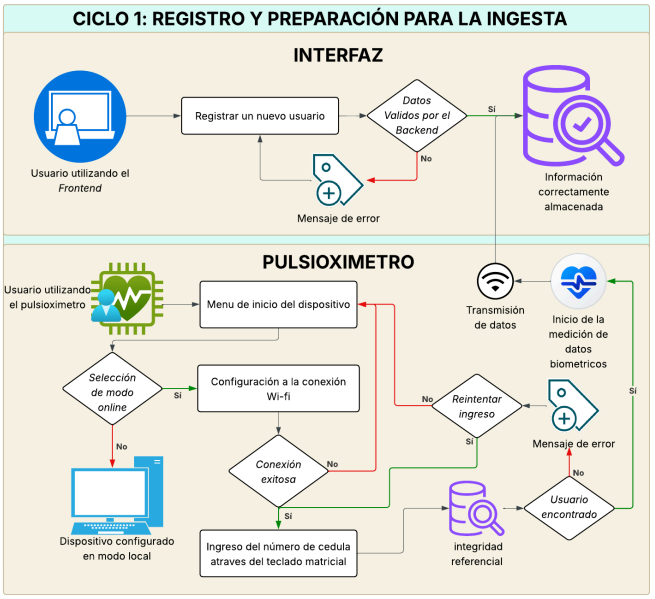


Figura 103, Diagrama de flujo para el registro de datos. [Elaboración propia]

7.4.9.2. CICLO 2: CONSULTA Y VISUALIZACIÓN SEGURA

El segundo ciclo, de consulta y visualización segura, abarca tanto la presentación de datos en tiempo real en el dispositivo como la recuperación de datos históricos desde la nube. La pantalla OLED del dispositivo asegura una visualización constante e inmediata de las mediciones de BPM y SpO2, ofreciendo *feedback* directo al usuario durante la adquisición del dato, sin importar si el modo operativo es Online u Offline.

El flujo de consulta remota se inicia cuando un usuario accede a la interfaz. El *frontend* envía una solicitud de datos al módulo *backend*. Este componente de lógica opera como una puerta de

seguridad, verificando la validez del *token* de autenticación del usuario y sus permisos de acceso antes de procesar la solicitud. Durante el procesamiento, la aplicación de servicios implementa mecanismos eficaces de manejo de errores y consulta la base de datos, aplicando los filtros necesarios en sus *endpoints* para optimizar la comodidad y eficiencia del usuario al realizar consultas. Finalmente, el servidor devuelve los registros solicitados en formato JSON, y el *frontend* transforma estos datos en tablas, gráficos y notificaciones visuales, adaptándose al perfil del usuario que inició sesión. Esta dinámica de servicios distribuidos asegura que la información médica fluya de manera segura, confiable y en tiempo real desde el dispositivo de adquisición hasta la visualización final.

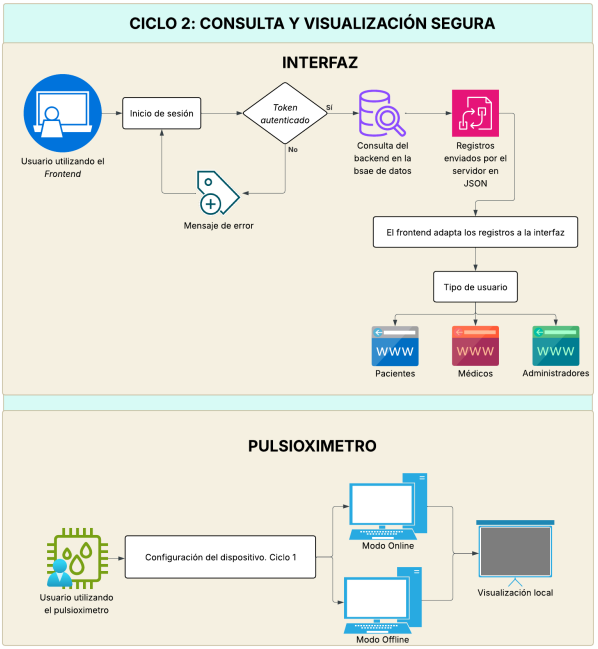


Figura 104, Diagrama de flujo para la visualización de datos. [Elaboración propia]

8. PROTOCOLO DE PRUEBA

Para validar la funcionalidad y precisión del prototipo se llevó a cabo una prueba preliminar. El objetivo principal fue validar parámetros fundamentales de funcionamiento como la precisión, estabilidad y respuesta del dispositivo así como verificar la confiabilidad de las mediciones obtenidas para el BPM y el SpO2.

Con este propósito se diseñó un protocolo de prueba en el cual se utilizó el dispositivo en un pequeño grupo poblacional. Para obtener un rango de datos representativo, la prueba se realizó en quince adultos sanos.

8.1. POBLACIÓN

Durante la conformación de la muestra se consideraron variables demográficas y fisiológicas como el sexo, la edad y el tono de piel, con el objetivo de obtener una población representativa de la variabilidad presente en usuarios potenciales. Esta decisión metodológica buscó fortalecer la validez externa del estudio, permitiendo evaluar el desempeño del prototipo bajo condiciones fisiológicas diversas y, por tanto, su posible generalización a un uso más amplio. Sin embargo, dichos factores no fueron incluidos como variables de análisis estadístico, dado que el propósito central de esta etapa fue examinar la estabilidad, repetibilidad y concordancia de las mediciones del prototipo frente al dispositivo de referencia, priorizando la caracterización técnica de su comportamiento por encima de las diferencias individuales de los participantes.

8.1.1. TONO DE PIEL

De los criterios anteriores es importante destacar el tono de piel, ya que, si bien el protocolo de prueba no tuvo como objetivo evaluar de forma directa la validez del dispositivo en distintos tonos de piel, se reconoció la relevancia de esta característica para garantizar su funcionamiento de manera universal. Por esta razón, se incluyeron en las pruebas participantes con la mayoría de los fototipos descritos por la escala de Fitzpatrick, pues la consideración de esta característica podría dar luz sobre las variaciones en la absorción y reflexión de la luz que pueden presentarse en función del tono cutáneo.

La escala de Fitzpatrick es un sistema de clasificación dermatológica que describe la respuesta de la piel a la radiación ultravioleta y agrupa los tonos de piel en seis categorías, que van desde la piel muy clara (tipo I) hasta la piel muy oscura (tipo VI). Esta clasificación es ampliamente utilizada en estudios biomédicos y en el desarrollo de dispositivos ópticos o basados en fotometría, ya que permite evaluar la eficacia de sensores o equipos que dependen de la interacción de la luz con la piel.

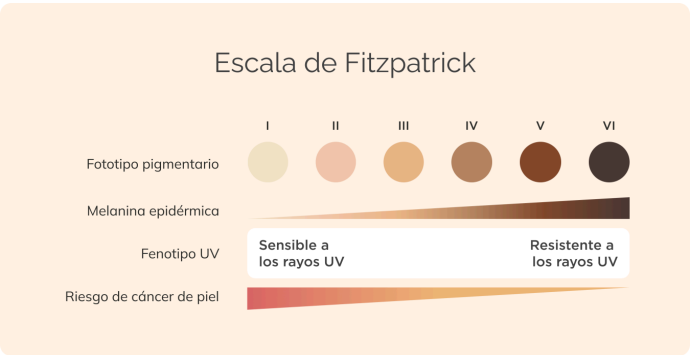


Figura 105, Escala de fototipos de Fitzpatrick. [Elaboración propia]

8.1.1.2 DISTRIBUCIÓN DE PARTICIPANTES

La población final participante en las pruebas del prototipo estuvo conformada por quince adultos sanos, seleccionados conforme a los criterios descritos previamente. A continuación, se presenta la distribución de esta población según sexo, grupo etario y fototipo de piel.

Tabla 2, Distribución según sexo. [Elaboración propia]

Sexo	Número de participantes
Femenino	2
Masculino	13
Total	15

Tabla 3, Distribución según rango de edad. [Elaboración propia]

Grupo etario	Rango de edad	Nombre del rango	Número de participantes
Adolescentes	≤18	A	4
Adultos jóvenes	>18	J	11
Total	15		

Tabla 4, Dispersión según el fototipo. [Elaboración propia]

Fototipo	Número de participantes
I	2
II	4
III	5
IV	3
V	1
VI	-
Total	15

Si bien la información de cada participante y los datos obtenidos en sus respectivas lecturas se encuentran reunidos en la base de datos, se creó un sistema de etiquetado que permitió mantener la trazabilidad de los registros experimentales fuera del portal, además de agrupar y analizar los resultados de las pruebas según variables. En este sistema de etiquetados todos los participantes fueron registrados mediante un sistema de etiquetado que permitió identificar de forma rápida y estructurada la información básica relevante para el estudio. La estructura del código de identificación utilizada fue la siguiente:

[Nombre]–[Sexo]–[Grupo etario]–[Fototipo]

1. Leonardo-M-J-I
2. Julián-M-J-V
3. Paula-F-A-I
4. David-M-J-II
5. Andrés-M-J-IV
6. Alvaro-M-J-II
7. Jean-M-J-III
8. Diego-M-J-III
9. Nicolás-M-A-III
10. Mariana-F-A-II
11. Sergio-M-J-II
12. Javier-M-A-III
13. Juan-M-J-IV
14. Alejandro-M-A-III
15. Jimmy-M-J-IV

8.2. PRUEBA DE PRECISIÓN Y ESTABILIDAD

El objetivo de esta prueba fue verificar la precisión y estabilidad de las mediciones obtenidas por el prototipo, comparándolas con las de un pulsioxímetro comercial de referencia. Para ello, se realizaron registros simultáneos de los valores de SpO_2 y BPM obtenidos por ambos dispositivos bajo las mismas condiciones.

Para la validación de las mediciones del prototipo se utilizó un pulsioxímetro digital ChoiceMMed, modelo MD300C11, fabricado por ChoiceMMed. Este dispositivo presenta una precisión de ±2% en las lecturas de saturación de oxígeno (SpO_2), lo que asegura un nivel de confiabilidad adecuado para su uso como instrumento de referencia. Está equipado con una pantalla LED y funciona mediante baterías tipo AAA.

La prueba se llevó a cabo en un entorno controlado y con iluminación estable. Cada participante permaneció sentado y en reposo, con el fin de minimizar las variaciones fisiológicas derivadas del movimiento o la actividad física. Previo a cada medición, se siguió el siguiente procedimiento estandarizado:

A. Reposo: el participante descansó en una posición cómoda durante cinco minutos antes de iniciar la medición.

B. Colocación: el prototipo y el pulsioxímetro comercial se colocaron de manera simultánea en dedos adyacentes de la misma mano, garantizando así que las condiciones de perfusión y temperatura fueran equivalentes.

C. Posición: el participante se sentó en una posición cómoda para evitar movimientos que pudieran afectar la lectura y se le solicitó permanecer en dicha posición hasta que los dispositivos terminaran la toma de datos.

Se realizaron 59 mediciones consecutivas en intervalos regulares, registrando los valores de SpO_2 y BPM obtenidos por ambos dispositivos. Con estos datos, se efectuó un análisis comparativo para evaluar la concordancia entre las mediciones y la consistencia temporal de los valores reportados por el prototipo.

Las variables analizadas en esta prueba fueron:

1. SpO_2: saturación de oxígeno medida por ambos dispositivos.

2. BPM: frecuencia cardíaca registrada simultáneamente.

3. Desviación estándar: empleada para determinar la estabilidad y repetibilidad de las mediciones del prototipo.

4. Sesgo**:** diferencia promedio entre los valores obtenidos por el prototipo y el dispositivo de referencia, empleada para estimar la tendencia sistemática del error.

8.3. ANÁLISIS DE RESULTADOS

Los datos registrados durante la prueba fueron organizados y procesados de manera que permitieran identificar tanto la variabilidad inherente a las mediciones como las tendencias sistemáticas entre dispositivos. En este análisis se consideraron indicadores descriptivos y comparativos, entre los cuales destacan la desviación estándar, utilizada para evaluar la estabilidad y repetibilidad de las lecturas, y el sesgo, empleado para estimar la tendencia del error sistemático entre ambos equipos. Estas métricas constituyen la base del análisis que se presenta a continuación, en el que se examinan de forma detallada los resultados obtenidos en cada una de las pruebas realizadas.

8.3.1. DESVIACIÓN ESTÁNDAR

La desviación estándar es una medida estadística que refleja el grado de variabilidad o dispersión de un conjunto de datos respecto a su promedio. En términos simples, permite conocer qué tan estables o consistentes son las mediciones obtenidas. Cuando la desviación estándar es baja, significa que las lecturas se mantienen próximas al valor medio, mientras que una desviación alta indica fluctuaciones considerables entre los datos registrados.

En el marco de la validación del prototipo, este indicador es fundamental para evaluar su estabilidad y repetibilidad. Un pulsioxímetro confiable debe producir resultados coherentes bajo las mismas condiciones de medición, de modo que una baja desviación estándar es un signo de precisión técnica y de una correcta detección de la señal fisiológica. Además, este parámetro permite determinar el grado de consistencia de las lecturas del prototipo en comparación con el dispositivo de referencia.

En este estudio, la desviación estándar se calculó a partir de la diferencia entre cada valor registrado y el promedio general de las mediciones, considerando la magnitud de esas variaciones para obtener una estimación de la dispersión. Durante las mediciones experimentales del SpO_2 se observó que las lecturas iniciales presentaban una estabilidad insuficiente, atribuible al comportamiento del modelo de cálculo lineal implementado en el código del prototipo. Dicho modelo tiende a generar una respuesta transitoria más lenta, ya que requiere varios ciclos de actualización antes de ajustarse correctamente a las fluctuaciones reales de la señal. Con base en esta observación, se decidió realizar el análisis a partir de dos valores de desviación estándar por participante, con el fin de distinguir entre la variabilidad total y la estabilidad intrínseca del dispositivo.

La desviación global se obtuvo considerando el total de mediciones registradas por participante (59 lecturas), proporcionando una estimación general de la dispersión a lo largo de toda la prueba. En contraste, la desviación estable se calculó tras remover las primeras quince mediciones de cada serie,

para analizar únicamente el comportamiento del dispositivo una vez alcanzada la estabilidad de la señal.

La tabla 5 muestra los valores de desviación estándar de la saturación de oxígeno correspondientes a cada sujeto de prueba, diferenciando entre la desviación estándar global y la desviación estándar estable. En la mayoría de los casos se observa una disminución en la variabilidad de los datos al considerar únicamente el periodo estable, lo que confirma que las lecturas iniciales influyen de manera significativa en la dispersión total de las mediciones.

Las diferencias entre ambas desviaciones oscilan entre 0.006% y 6.53%, lo que indica que el impacto de las mediciones iniciales varía entre participantes. Los sujetos con diferencias mayores (por ejemplo, David, Diego y Jean) reflejan un comportamiento menos estable durante el arranque de la medición, mientras que aquellos con diferencias mínimas muestran una respuesta más consistente del sistema.

En conjunto, estos resultados permiten inferir que, una vez superada la fase inicial de ajuste, el prototipo mantiene niveles adecuados de estabilidad y repetibilidad, lo que respalda su desempeño para aplicaciones de monitoreo continuo.

Tabla 5, Desviación estándar de SpO_2 (global y estable). [Elaboración propia]

Sujeto De Prueba	D.E. Global Spo2 (%)	D.E. Estable Spo2 (%)	Diferencia De D.E. Spo2 (%)
Jimmy	0,909	0,903	0,006
Alejandro	4,436	0,000	4,436
Juan	3,712	3,717	-0,005
Javier	4,249	1,631	2,617
Sergio	3,036	1,171	1,866
Mariana	1,630	1,427	0,203
Nicolás	6,078	1,022	5,056
Diego	6,482	2,491	3,990
Jean	0,880	0,505	0,376
Alvaro	4,645	2,365	2,280
Andrés	1,454	1,190	0,264
David	7,209	0,679	6,530
Paula	5,650	2,018	3,631
Julián	4,336	1,116	3,220
Leonardo	6,285	0,973	5,312
Promedio	4,066	1,414	2,652

En la tabla 6, se presentan los valores de desviación estándar global y desviación estándar estable para cada participante. En promedio, la desviación global fue de 4.45 BPM, mientras que la desviación estable fue de 4.28 BPM, lo cual evidencia una consistencia general entre ambas métricas. La diferencia promedio de apenas 0.17 BPM indica que la exclusión de las lecturas iniciales no produjo un cambio significativo en la dispersión de los datos, sugiriendo que las mediciones del prototipo son estables desde las primeras lecturas.

Este comportamiento puede explicarse por el algoritmo de validación de pulsos implementado en el código del dispositivo, que filtra automáticamente valores fuera del rango fisiológicamente plausible (40–180 BPM). Dicha validación reduce la influencia de lecturas erróneas estabilizando la señal desde etapas tempranas de la medición.

Las variaciones más notables entre ambas desviaciones se observaron en los participantes Mariana y Juan, con diferencias superiores a 1 BPM. Estos casos pueden atribuirse a ligeros desplazamientos del sensor, movimientos del sujeto o interferencias lumínicas que introducen perturbaciones temporales en la detección de los pulsos. Aun así, el comportamiento general muestra que el algoritmo logra compensar eficazmente dichas perturbaciones, manteniendo una alta repetibilidad y coherencia en las mediciones de frecuencia cardíaca.

Tabla 6, Desviación estándar de BPM (global y estable). [Elaboración propia]

Sujeto De Prueba	D.E. Global BPM	D.E. Estable BPM	Diferencia De D.E. BPM
Jimmy	3,765	3,348	0,417
Alejandro	2,846	2,316	0,530
Juan	5,302	3,997	1,305
Javier	6,628	7,200	-0,573
Sergio	3,603	3,999	-0,396
Mariana	8,100	5,921	2,178
Nicolás	5,574	6,305	-0,731
Diego	4,459	4,473	-0,014
Jean	4,429	4,466	-0,037
Alvaro	3,552	3,638	-0,086
Andrés	3,844	3,663	0,180
David	4,403	4,534	-0,131
Paula	5,161	4,982	0,179
Julián	3,088	3,345	-0,257
Leonardo	2,027	2,114	-0,087
Promedio	4,452	4,287	0,165

En conjunto, los resultados obtenidos del análisis de la desviación estándar para las variables SpO_2 y BPM evidencian un desempeño estable y repetible del prototipo bajo condiciones controladas de prueba. En ambos casos, las diferencias entre las desviaciones globales y estables fueron mínimas, lo que sugiere que el comportamiento del dispositivo alcanza rápidamente una condición de equilibrio en la medición y mantiene lecturas consistentes a lo largo del tiempo. Las fluctuaciones observadas en algunos participantes se asocian principalmente a factores externos, como el movimiento o pequeñas variaciones en la colocación del sensor, más que a deficiencias del sistema de procesamiento. En términos generales, estos resultados respaldan la fiabilidad del prototipo para registrar señales fisiológicas con una variabilidad reducida, y constituyen un indicador favorable de su precisión, estabilidad y consistencia operativa.

8.3.2. SESGO PROMEDIO

El sesgo es una medida estadística que refleja la diferencia sistemática promedio entre los valores obtenidos por el prototipo y los registrados por el dispositivo de referencia. En otras palabras, indica si el prototipo tiende a sobreestimar o subestimar las mediciones respecto al estándar utilizado como comparación.

En el contexto de la validación de un pulsioxímetro, el sesgo es un indicador esencial, ya que permite evaluar la precisión absoluta del dispositivo y determinar si las discrepancias observadas obedecen a un patrón constante o a variaciones aleatorias. Un sesgo cercano a cero indica una alta exactitud, mientras que valores positivos o negativos sostenidos pueden señalar la necesidad de ajustes en la calibración o en el procesamiento de señal.

Para su cálculo, el sesgo se obtuvo a partir del promedio de las diferencias entre las lecturas del prototipo y las del dispositivo de referencia en cada medición, expresado en las mismas unidades que la variable analizada (SpO_2 o BPM). Este procedimiento permite identificar la tendencia sistemática del error y establecer si el prototipo mantiene coherencia con las mediciones del instrumento de comparación a lo largo del tiempo.

En la tabla 7 se presentan los valores de sesgo global y sesgo estable correspondientes a las mediciones de SpO_2 obtenidas para cada participante. En términos generales, los resultados muestran que no existe un sesgo sistemático significativo entre ambas condiciones, lo cual sugiere que el algoritmo implementado logra mantener una precisión promedio adecuada durante todo el registro. El valor promedio del sesgo global fue de –1.366 %, mientras que el sesgo estable promedió 0.305 %, con una diferencia media de –1.671 %, lo que indica una ligera tendencia del prototipo a subestimar los valores de saturación durante los primeros instantes de la medición.

Las variaciones más notables entre ambos sesgos se observan en los sujetos Nicolás, Diego, David, Leonardo, Álvaro y Paula, cuyas diferencias superan el –2 %. Estas discrepancias pueden atribuirse nuevamente al comportamiento del modelo de cálculo lineal implementado en el código del prototipo, el cual tiende a presentar una respuesta más lenta en la convergencia de los valores iniciales. Este fenómeno genera una subestimación temporal de los niveles de SpO_2 durante las primeras lecturas, antes de que el algoritmo logre ajustarse adecuadamente al comportamiento real de la señal. En consecuencia, las diferencias observadas entre el sesgo global y el sesgo estable no reflejan un error estructural del sensor, sino el efecto transitorio del modelo de estimación lineal sobre los datos iniciales.

En conjunto, los resultados evidencian que el modelo de estimación empleado para SpO_2 presenta un comportamiento estable y reproducible, con un nivel de sesgo bajo y dentro de los márgenes aceptables para mediciones ópticas no invasivas.

Tabla 7, Sesgo promedio de SpO_2 (global y estable). [Elaboración propia]

Sujeto De Prueba	Sesgo Global Spo2 (%)	Sesgo Estable Spo2 (%)	Diferencia De Sesgo Spo2 (%)
Jimmy	2,034	1,956	0,078
Alejandro	1,339	3,000	-1,661
Juan	-5,017	-4,156	-0,861
Javier	-4,136	-2,444	-1,691
Sergio	0,492	1,756	-1,264
Mariana	-0,793	-0,311	-0,482
Nicolás	0,542	3,333	-2,791
Diego	-3,492	-0,556	-2,936
Jean	2,186	2,533	-0,347
Alvaro	-1,898	0,333	-2,232
Andrés	-1,559	-1,244	-0,315
David	-1,559	1,756	-3,315
Paula	-6,339	-4,289	-2,050
Julián	-0,542	1,600	-2,142
Leonardo	-1,746	1,311	-3,057
Promedio	-1,366	0,305	-1,671

En la tabla 8 se observa que el sesgo promedio global fue de 0,649 BPM, mientras que el sesgo estable promedio fue de 0,684 BPM, lo que evidencia una diferencia mínima (–0,035 BPM) entre ambas condiciones. Esta similitud sugiere que el comportamiento del modelo lineal implementado en el código del prototipo mantiene una estimación consistente incluso tras eliminar las mediciones iniciales.

Las variaciones más notables entre ambos sesgos se presentan en los sujetos Mariana, Juan y Paula, cuyas diferencias superan ±1 BPM. Estas discrepancias pueden atribuirse a la sensibilidad del modelo de cálculo lineal durante los primeros segundos de registro, cuando la señal aún no ha alcanzado una convergencia estable. Este comportamiento se alinea con lo observado en las mediciones de SpO_2, donde el modelo lineal tiende a responder de forma más lenta ante los cambios iniciales en la señal óptica.

No obstante, el promedio general de sesgo cercano a cero indica que no existe una tendencia sistemática significativa de sobre o subestimación, lo que demuestra que el algoritmo logra un ajuste adecuado al valor real de la frecuencia cardíaca y garantiza una buena precisión promedio en las mediciones del prototipo.

Tabla 8, Sesgo promedio de BPM (global y estable). [Elaboración propia]

Sujeto De Prueba	Sesgo Global BPM	Sesgo Estable BPM	Diferencia Sesgo BPM
Jimmy	0,610	1,556	-0,945
Alejandro	-0,271	-0,667	0,395
Juan	1,169	-0,422	1,592
Javier	1,932	1,444	0,488
Sergio	0,322	0,222	0,100
Mariana	-1,638	1,067	-2,705
Nicolás	-0,712	-0,533	-0,179
Diego	1,814	2,244	-0,431
Jean	0,797	0,765	0,032
Alvaro	0,153	0,644	-0,492
Andrés	1,051	0,889	0,162
David	1,576	1,622	-0,046
Paula	2,237	0,844	1,393
Julián	-0,136	-0,244	0,109
Leonardo	0,831	0,822	0,008
Promedio	0,649	0,684	-0,035

8.3.2.1. SESGO ABSOLUTO PROMEDIO

En el apartado anterior se analizó el sesgo promedio de SPO_2 y BPM, con el objetivo de identificar si el prototipo tiende a sobrestimar o subestimar las mediciones respecto al valor de referencia. No obstante, evaluar únicamente el sesgo promedio puede ocultar la magnitud real de los errores, ya que desviaciones positivas y negativas podrían compensarse entre sí. Por ello, es igualmente importante considerar el sesgo absoluto promedio, que refleja cuánto se desvían en promedio las mediciones del valor real sin importar la dirección del error, proporcionando una evaluación más completa de la precisión del dispositivo.

En la tabla 9 se observa que el sesgo promedio global fue de 3,506 %, mientras que el sesgo promedio en condiciones estables fue de 2,317 %, con una diferencia de 1,189 %. Esto evidencia que el

dispositivo es ligeramente menos preciso cuando la señal no es estable, aunque mantiene una tendencia general a sobreestimar la SPO_2. El sesgo absoluto promedio, también de 3,506 %, indica que, en promedio, las mediciones se desvían aproximadamente 3,5 % del valor real, lo que indica que la magnitud de los errores individuales se mantiene dentro de rangos aceptables.

Las diferencias más destacadas entre el sesgo global y el estable se presentan en los sujetos Paula, David, Diego y Javier, con discrepancias superiores a 2 %, lo que sugiere que la precisión se ve afectada principalmente en condiciones de señal inestable. En contraste, la mayoría de los sujetos presentan diferencias menores, evidenciando un comportamiento consistente del pulsioxímetro en mediciones estables.

En síntesis, la combinación del análisis de sesgo promedio y sesgo absoluto promedio permite afirmar que el dispositivo diseñado tiende a sobreestimar ligeramente la SPO_2, manteniendo un error promedio total aceptable, lo que demuestra que el algoritmo de estimación proporciona mediciones confiables y precisas.

Tabla 9, Sesgo absoluto promedio de SpO_2 (global y estable). [Elaboración propia]

Sujeto De Prueba	Sesgo Global Spo2 (%)	Sesgo Estable Spo2 (%)	Diferencia De Sesgo Spo2 (%)
Jimmy	2,034	1,956	0,078
Alejandro	3,610	3,000	0,610
Juan	5,051	4,200	0,851
Javier	4,136	2,444	1,691
Sergio	2,254	1,756	0,499
Mariana	1,237	0,844	0,393
Nicolás	4,542	3,333	1,209
Diego	4,542	1,933	2,609
Jean	2,220	2,533	-0,313
Alvaro	3,695	2,022	1,673
Andrés	1,661	1,378	0,283
David	4,237	1,756	2,482
Paula	6,339	4,289	2,050
Julián	3,119	1,778	1,341
Leonardo	3,915	1,533	2,382
Promedio	3,506	2,317	1,189

Al igual que en el análisis de la saturación de oxígeno, evaluar únicamente el sesgo promedio de la frecuencia cardíaca (BPM) permite identificar si el pulsioxímetro tiende a sobrestimar o subestimar las mediciones frente al valor de referencia. Sin embargo, el sesgo promedio por sí solo no refleja la magnitud total de los errores, ya que desviaciones positivas y negativas pueden compensarse entre sí.

Por esta razón, es igualmente importante calcular el sesgo absoluto promedio, que refleja cuánto se desvían en promedio las mediciones del valor real sin importar la dirección del error, proporcionando una evaluación más completa de la precisión del dispositivo.

En la tabla 10 se observa que el sesgo promedio global fue de 3,616 BPM, mientras que el sesgo promedio en condiciones estables fue de 3,483 BPM, con una diferencia de 0,133 BPM. Esto indica que el dispositivo mantiene un comportamiento consistente entre las mediciones generales y las estables, con una ligera variación que evidencia que la precisión puede verse mínimamente afectada por condiciones de señal inestable.

Las diferencias más significativas entre el sesgo global y el estable se presentan en los sujetos Juan, Javier, Mariana y Nicolás, cuyas discrepancias superan ±0,7 BPM, lo que sugiere que el prototipo puede verse afectado por variaciones temporales de la señal durante los primeros segundos de registro. En contraste, la mayoría de los sujetos presentan diferencias menores, evidenciando que el dispositivo proporciona mediciones consistentes en condiciones estables.

En conclusión, el análisis de sesgo promedio y sesgo absoluto promedio indica que el pulsioxímetro diseñado tiende a sobreestimar ligeramente la frecuencia cardíaca, pero mantiene un error promedio total aceptable. Estos resultados demuestran que el algoritmo de estimación proporciona mediciones confiables y precisas, especialmente bajo condiciones de señal estable.

Tabla 10, Sesgo absoluto promedio de BPM (global y estable). [Elaboración propia]

Sujeto De Prueba	Sesgo Global BPM	Sesgo Estable BPM	Diferencia Sesgo BPM
Jimmy	2,847	2,711	0,136
Alejandro	2,136	1,956	0,180
Juan	4,085	3,356	0,729
Javier	5,322	5,844	-0,522
Sergio	2,661	2,978	-0,317
Mariana	5,746	4,178	1,568
Nicolás	4,068	4,800	-0,732
Diego	3,610	3,622	-0,012
Jean	3,610	3,933	-0,323
Alvaro	2,492	2,600	-0,108
Andrés	2,881	2,622	0,259
David	3,746	3,889	-0,143
Paula	4,610	3,956	0,655
Julián	4,610	3,956	0,655

Leonardo	1,814	1,844	-0,031
Promedio	3,616	3,483	0,133

El análisis de sesgo permitió corroborar la coherencia del prototipo en la estimación de las variables fisiológicas medidas, evidenciando un margen de error reducido respecto al dispositivo de referencia. Si bien se observaron ligeras discrepancias en algunos participantes durante las mediciones iniciales —atribuibles al comportamiento del modelo de cálculo lineal implementado—, estas diferencias no representan un error sistemático significativo. El promedio general de sesgo próximo a cero en ambos parámetros confirma que el prototipo no presenta una tendencia consistente a sobrestimar ni subestimar los valores medidos. Asimismo, el análisis del sesgo absoluto promedio evidencia que, independientemente de la dirección del error, la magnitud de las desviaciones se mantiene dentro de rangos aceptables, lo que respalda la precisión general del prototipo frente al dispositivo de referencia.

9. PRESUPUESTO

Para el costeo de este proyecto es necesario tener en cuenta todos los elementos que se emplearon para el desarrollo del mismo, tales como microprocesadores, componentes varios de conexiones. computador, sensores entre otros. Adicionalmente también se incluirá un salario que refleje el trabajo y tiempo de las personas que se vean involucradas en la producción del trabajo como lo son los autores, asesores o directores.

Teniendo esto en mente se construyen las siguientes tablas para tener un presupuesto aproximado de lo que cuesta realizar el proyecto planteado en este documento.

9.1. RECURSOS FÍSICOS

Tabla 11, Recursos físicos. [Elaboración propia]

Elemento	Cantidad	Costo
MAX 30102	1	$ 22,000
ESP32	1	$ 34,900
COMPUTADOR	1	$ 1,699,000
IMPRESIÓN 3D	4	$ 200,000
PANTALLA OLED	1	$ 35,000
PCB	1	$ 17,000
REGLETA 1X40	2	$ 5,000
TECLADO MATRICIAL	1	$ 12.000
CABLES DE CONEXIÓN	16	$ 10,000
MÓDULO TP4056	1	$ 4,000
BATERIA Y PORTA PILAS	1	$ 20,500
	TOTAL	$ 2,059,400

9.2. RECURSOS HUMANOS

Tabla 12, Recursos Humanos. [Elaboración propia]

RECURSO HUMANO	COSTO HORA	COSTO SEMANAL	COSTO TOTAL
ASESOR	$ 100,000	$ 200,000	$ 4,800,000

AUTOR 1	$	6,000	$	600,000	$ 18,000,000
AUTOR 2	$	6,000	$	600,000	$ 18,000,000
TOTAL	$	112,000	$	1,400,000	$ 40,800,000

9.3. PRESUPUESTO TOTAL

Tabla 13, Presupuesto total. [Elaboración propia]

PRESUPUESTO		
RECURSO HUMANO	$	40,800,000
RECURSO FÍSICO	$	2,034,900
TOTAL	$	42,834,900

10. CONCLUSIONES

10.1. PRIMER OBJETIVO ESPECÍFICO

Se estableció un sistema de base de datos utilizando Supabase con PostgreSQL, garantizando confiabilidad operativa, escalabilidad y alto rendimiento en entornos de alta concurrencia, cumpliendo con los requerimientos de registro de signos vitales y tiempos de medición.

La estructura relacional permite aplicar integridad referencial mediante llaves primarias y foráneas, asegurando que cada medición esté correctamente asociada a un paciente existente y manteniendo la coherencia y trazabilidad de la información clínica.

La base de datos puede ejecutar lógica programable mediante triggers y funciones, lo que permite validar automáticamente los valores ingresados y generar alertas inmediatas ante mediciones fuera de los umbrales predefinidos, asegurando un manejo de datos seguro, inteligente y preparado para el crecimiento del sistema.

10.2. SEGUNDO OBJETIVO ESPECÍFICO

Se desarrolló una interfaz de usuario web dinámica e intuitiva utilizando React.js, que permite a pacientes, médicos y administradores acceder a la información de manera diferenciada según su perfil, optimizando la visualización de mediciones actuales e históricas.

La implementación de un backend con FastAPI, desplegado en Render, garantiza la gestión segura y eficiente de la autenticación, autorización y reglas funcionales del sistema, permitiendo el manejo de múltiples usuarios concurrentes sin comprometer la estabilidad del servicio.

La integración del frontend y el backend con plataformas de despliegue en la nube (Vercel y Render) asegura alta disponibilidad, escalabilidad y facilidad de mantenimiento, permitiendo la actualización rápida de funcionalidades y el acceso confiable a los datos médicos desde cualquier ubicación.

10.3. TERCER OBJETIVO ESPECÍFICO

El pulsioxímetro fue diseñado con una arquitectura de cuatro capas (adquisición y procesamiento, almacenamiento, lógica y servicios, y presentación), integrando el sensor óptico MAX30102 y el microcontrolador ESP32, lo que permite un flujo continuo de datos desde la adquisición hasta la visualización, garantizando estabilidad y confiabilidad en las mediciones de SpO_2 y BPM.

El diseño de la PCB y la carcasa priorizó la proximidad de componentes críticos y la minimización de la longitud de las trazas I^2C, aumentando la inmunidad al ruido y asegurando la integridad de las señales, mientras que la carcasa personalizada proporciona protección mecánica y accesibilidad ergonómica para el teclado y la pantalla OLED.

La integración de componentes de interacción como la pantalla OLED y el teclado matricial permite al usuario visualizar en tiempo real las mediciones y controlar el dispositivo de manera intuitiva, garantizando una interfaz física funcional y accesible dentro del prototipo.

10.4. CUARTO OBJETIVO ESPECÍFICO

Se implementó un sistema de comunicación eficiente entre el pulsioxímetro y la base de datos utilizando el microcontrolador ESP32, que combina alta capacidad de procesamiento y conectividad Wi-Fi integrada, lo que permite el envío continuo de datos fisiológicos sin necesidad de hardware adicional.

La arquitectura de doble núcleo del ESP32 permite ejecutar tareas concurrentes, dedicando un núcleo a la adquisición y procesamiento de señales de SpO_2 y BPM, y el otro a la transmisión de datos, garantizando eficiencia y estabilidad en la operación del dispositivo.

La conexión inalámbrica se optimizó mediante la librería WiFiManager.h, que facilita la gestión automática de credenciales a través de un portal cautivo, asegurando un registro seguro y continuo de las mediciones en la base de datos, esencial para la funcionalidad de telemetría del sistema.

10.5. QUINTO OBJETIVO ESPECÍFICO

La prueba con quince adultos permitió validar la funcionalidad del prototipo, evidenciando que los parámetros medidos por el pulsioxímetro se obtienen de manera consistente y confiable bajo condiciones controladas, lo que respalda su estabilidad operativa y repetibilidad.

El análisis de la desviación estándar mostró que las mediciones son estables y presentan baja dispersión una vez superadas las lecturas iniciales. Los valores promediados fueron de 1,414 % para SpO_2 y 4,287 BPM, lo que indica que las fluctuaciones registradas se mantienen dentro de rangos reducidos y que el dispositivo logra una medición precisa incluso ante pequeñas variaciones externas, como movimientos del usuario o ligeros cambios en la colocación del sensor.

La evaluación del sesgo promedio y sesgo absoluto promedio, tomando como referencia el pulsioxímetro de modelo MD300C11 de la marca CHOICEMMED, evidenció que el prototipo tiende a sobreestimar ligeramente los valores, sin presentar un error sistemático significativo. Los valores promediados fueron de 3,506 % de sesgo global para SpO_2 y 3,616 BPM para la frecuencia cardíaca. Los sesgos absolutos promedios fueron de $\pm 2.317\%$ para SpO_2 y ± 3.956 BPM para la frecuencia cardiaca lo que confirma que la magnitud de las desviaciones se mantiene dentro de rangos aceptables. Esto demuestra que el algoritmo de estimación implementado permite obtener mediciones precisas y confiables para ambos parámetros fisiológicos.

11. TRABAJOS FUTUROS

El desarrollo del prototipo de pulsioxímetro ha permitido establecer una base sólida para la medición confiable de la saturación de oxígeno y la frecuencia cardíaca; sin embargo, existen diversas oportunidades de mejora y expansión que podrían implementarse en futuras versiones del dispositivo. En primer lugar, se sugiere explorar la aplicación de un modelo exponencial para el cálculo de la saturación de oxígeno en lugar del modelo lineal actualmente utilizado. Este cambio podría proporcionar una interpretación más realista del comportamiento fisiológico, minimizando los valores atípicos observados durante las primeras mediciones y mejorando la precisión inicial del dispositivo.

Asimismo, se recomienda optimizar el diseño del encapsulado para reducir la incidencia de factores externos que puedan afectar la medición, tales como la presión ejercida por el dedo sobre el sensor, el movimiento involuntario de la extremidad o la interferencia de luz ambiental. Un diseño ergonómico y mejor protegido aumentaría la consistencia de las lecturas y la experiencia del usuario.

Otra línea de mejora corresponde a la reducción de costos mediante la reevaluación de los componentes seleccionados, buscando alternativas que mantengan la calidad del dispositivo pero optimicen el presupuesto de producción, lo que facilitaría su escalabilidad y accesibilidad.

Finalmente, se considera la integración de nuevos sensores, como termómetros o tensiómetros, que permitan la medición de parámetros adicionales y la eventual creación de una estación médica integral. Esta expansión tecnológica ampliaría las funcionalidades del prototipo, ofreciendo un monitoreo más completo y versátil de los signos vitales, con potencial aplicación en entornos clínicos y domésticos.

12. BIBLIOGRAFÍA

[1] MINISTERIO DE SALUD Y PROTECCIÓN SOCIAL, “Sistema de Salud ha realizado 63,1 millones de atenciones virtuales y telefónicas,” *Boletín de Prensa No 590* , Accessed: Mar. 27, 2025. [Online]. Available: https://www.minsalud.gov.co/Paginas/Sistema-de-Salud-ha-realizado-63.1-millones-de-atenciones-virtuales-y-telefonicas.aspx

[2] MINISTERIO DE SALUD Y PROTECCIÓN, “ORIENTACIONES PARA LA IMPLEMENTACIÓN DE LA ESTRATEGIA DE TRATAMIENTO DIRECTAMENTE OBSERVADO (TDO) CON SOPORTE EN LAS TECNOLOGÍAS DE INFORMACIÒN Y COMUNICACIONES. ,” Colombia, Apr. 2020. Accessed: Mar. 27, 2025. [Online]. Available: https://www.minsalud.gov.co/sites/rid/Lists/BibliotecaDigital/RIDE/VS/PP/ET/10orientaciones-tdo-virtual-2020.pdf

[3] NACIONES UNIDAS, “Objetivos De Desarrollo Sostenible.” Accessed: Mar. 27, 2025. [Online]. Available: https://www.un.org/sustainabledevelopment/es/sustainable-development-goals/

[4] Organización mundial de la Salud, “Estrategia mundial sobre salud digital en 2020-2025,” Aug. 2021. Accessed: Mar. 27, 2025. [Online]. Available: https://iris.who.int/bitstream/handle/10665/344251/9789240027572-spa.pdf?sequence=1

[5] Asamblea General de Las Naciones Unidas, “Documento final de la reunión de alto nivel de la Asamblea General sobre el examen general de la aplicación de los resultados de la Cumbre Mundial sobre la Sociedad de la Información ,” Nueva York , Feb. 2016. Accessed: Mar. 27, 2025. [Online]. Available: https://unctad.org/es/system/files/official-document/ares70d125_es.pdf

[6] L. T. Grajales Becerra, Y. G. Angel Bello, and J. D. Tapiero Rojas, “La edad como factor determinante en la atención por teleconsulta,” *Revista Colombiana de Medicina Física y Rehabilitación*, Aug. 03, 2021. Accessed: Mar. 27, 2025. [Online]. Available: https://revistacmfr.org/index.php/rcmfr/article/view/298/337

[7] Organización Panamericana de Salud, “Las personas mayores de 60 años han sido las más afectadas por la COVID-19 en las Américas,” Washington D.C., Sep. 2020. Accessed: Mar. 27, 2025. [Online]. Available: https://www.paho.org/es/noticias/30-9-2020-personas-mayores-60-anos-han-sido-mas-afectadas-por-covid-19-americas#:~:text=Si%20bien%20todos%20est%C3%A1n%20el,veces%20mayor%20que%20la%20media.

[8] Julio Cesar Cubillos Álzate, Mariana Matamoros Cárdenas, and Santiago Alberto Perea Caro, “Boletines Poblacionales1 : Personas Adultas Mayores de 60 años,” Bogotá D.C., Dec. 2020. Accessed: Mar. 27, 2025. [Online]. Available: https://www.minsalud.gov.co/sites/rid/Lists/BibliotecaDigital/RIDE/DE/PS/280920-boletines-poblacionales-adulto-mayorI-2020.pdf

[9] Organización Panamericana de Salud, “Aspectos técnicos y regulatorios sobre el uso de oxímetros de pulso en el monitoreo de pacientes con COVID-19,” Aug. 2020. Accessed:

Mar. 27, 2025. [Online]. Available: https://iris.paho.org/bitstream/handle/10665.2/52551/OPSHSSMTCOVID-19200029_spa.pdf?sequence=1&isAllowed=y#:~:text=El%20dispositivo%20m%C3%A9dico%20empleado%20en,la%20sangre%20inferior%20al%20normal.

[10] Common Spirit, "Vital Virtual Care Offered During COVID-19 Comes with 'Green' Benefits," Accessed: Mar. 27, 2025. https://www.commonspirit.org/news-articles/vital-covid-19-virtual-care-comes-with-green-benefits

[11] O. I. Arias Juárez, "DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE MONITOREO PARA LA MEDICIÓN DEL PULSO CARDÍACO Y SATURACIÓN DE OXÍGENO EN LA SANGRE," Universidad de Las Américas , Ecuador, 2017. Accessed: Mar. 27, 2025. [Online]. Available: https://dspace.udla.edu.ec/bitstream/33000/8359/1/UDLA-EC-TIRT-2017-36.pdf

[12] F. González Romero, "Diseño e implementación de un pulsioxímetro reflexivo y estudio de su funcionamiento en diferentes zonas del cuerpo," Escuela Técnica Superior de Ingeniería Universidad de Sevilla, Sevilla, 2019. Accessed: Mar. 27, 2025. [Online]. Available: https://idus.us.es/server/api/core/bitstreams/b26f1c02-d9e7-4e62-9a5b-9b1bee819fb3/content

[13] M. A. Zea Guaqueta and O. G. Nimisica Garcia, "Sistema de control médico remoto," Universidad Distrital Francisco José De Caldas, Colombia , 2017. Accessed: Mar. 27, 2025. [Online]. Available: https://repository.udistrital.edu.co/server/api/core/bitstreams/10a13bed-2b04-41be-8bdb-436ef116cc2b/content

[14] Patrício Amorim. Carlos Augusto, "Sensores," Guaratinguetá , 2010. Accessed: Mar. 27, 2025. [Online]. Available: https://www.feg.unesp.br/Home/PaginasPessoais/ProfMarceloWendling/4---sensores-v2.0.pdf

[15] A. E. Sánchez De La Cueva, "DESARROLLO DE UN PROTOTIPO DE PRÓTESIS DE PIERNA DERECHA CON UN SISTEMA DE MEDICIÓN DE SIGNOS VITALES EN UN DISPOSITIVO SMARTPHONE," UNIVERSIDAD TECNOLÓGICA ISRAEL, ECUADOR, 2019. Accessed: Mar. 27, 2025. [Online]. Available: http://repositorio.uisrael.edu.ec/bitstream/47000/2138/1/UISRAEL-EC-ELDT-378.242-2019-069.pdf

[16] Marcos Gabriel Sein-Echaluce García, "Pulsioxímetro con registro de datos en IOT y generación de alertas," Escuela de Ingeniería y Arquitectura UNIVERSIDAD DE ZARAGOZA, Zaragoza, 2019. Accessed: Mar. 27, 2025. [Online]. Available: https://core.ac.uk/download/pdf/290000514.pdf

[17] M. Cruz Luna, J. Torres Jiménez, A. Hernández Luna, and R. Luna Santos, "Monitor de temperatura, ritmo cardiaco y ubicación enviando mensajes de texto a un teléfono celular," *RITI Journa*, vol. Vol. 9, 18, no. doi: https://doi.org/10.36825/RITI.09.18.007, May 2021, Accessed: Mar. 27, 2025. [Online]. Available: https://dialnet.unirioja.es/servlet/articulo?codigo=7955982

[18] C. G. MARCATOMA PALTA, “APLICACIONES DE MICROCONTROLADORES PARA INTERACCIÓN ENTRE ACTUADORES Y SENSORES UTILIZANDO UN MÓDULO EASYPIC V8,” UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA, Ecuador, 2020. Accessed: Mar. 27, 2025. [Online]. Available: https://repositorio.upse.edu.ec/handle/46000/5362

[19] I. VILA CASTELLA, “DISEÑO Y DESARROLLO DEL PROTOTIPO DE UN BRAZO ROBOTICO ARTICULADO ANTROPOMORFICO CONTROLADO MEDIANTE UN MICROCONTROLADOR ARDUINO MEGA 2560,” UNIVERSIDAD POLITÉCNICA DE VALENCIA, España, 2018. Accessed: Mar. 27, 2025. [Online]. Available: http://hdl.handle.net/10251/109875

[20] R. E. Gómez, S. L. C. Álvarez, G. C. Chávez, and G. F. G. Flores, “Circuito de acondicionamiento con amplio rango dinámico para sensores resistivos basado en PSoC,” *Ingeniería Investigación y Tecnología*, vol. XXIII (número 3), no. ISSN 2594-0732, pp. 4–5, Jun. 2022, Accessed: Mar. 27, 2025. [Online]. Available: https://www.revistaingenieria.unam.mx/numeros/2022/v23n3-06.pdf

[21] D.A. Román-Landeros, N. Alvarado-Tovar, and V.E. Manqueros-Avilés, “Microcontrolador PSOC 5 para la operación de servo sistemas SURESERVO instalados en un prototipo de seguidor solar de dos ejes,” *Revista Ciencia, Ingeniería y Desarrollo Tec Lerdo*, vol. Volúmen: 1 Número 4, no. 2448–623X, pp. 2–3, 2018. Accessed: Mar. 27, 2025. Available: http://repocid.itslerdo.edu.mx/coninci2018/CID037.pdf

[22] D. Domínguez Castillejo, “Desarrollo de un pulsioxímetro para la medida del monóxido de carbono en sangre,” ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE TELECOMUNICACIÓN, Madrid, 2022. Accessed: Mar. 27, 2025. [Online]. Available: https://oa.upm.es/71784/1/TFG_DiegoDominguezCastillejo.pdf

[23] M. Aliaga Igea and C. Ruiz Zamarreño, “DESARROLLO DE SISTEMAS BASADOS EN UN MICROCONTROLADOR PIC,” Universidad Pública de Navarra , España, 2021. Accessed: Mar. 27, 2025. [Online]. Available: https://academica-e.unavarra.es/server/api/core/bitstreams/59c734d8-8204-4329-b3f3-464c8715d7c8/content

[24] CRUZ ZAMBRANO MIGUEL ÁNGEL, “IMPLEMENTACIÓN DE CIRCUITOS PROGRAMABLES CON MICROCONTROLADORES MEDIANTE TARJETA RASPBERRY PI PARA PRÁCTICAS EN EL LABORATORIO DE ELECTRÓNICA, DE LA CARRERA INGENIERÍA EN COMPUTACIÓN Y REDES.,” UNIVERSIDAD ESTATAL DEL SUR DE MANABÍ, Ecuador, 2021. Accessed: Mar. 27, 2025. [Online]. Available:https://repositorio.unesum.edu.ec/bitstream/53000/2676/1/CRUZ%20ZAMBRANO%20MIGUEL%20ANGEL.pdf

[25] C. R. VALENCIA TUQUERES, “EVALUACIÓN DE TECNOLOGÍAS INALÁMBRICAS EN REDES DE ÁREA DOMÉSTICA PARA OBTENER LA CURVA CARACTERÍSTICA DE CARGA EN EDIFICIOS INTELIGENTES,” UNIVERSIDAD POLITÉCNICA SALESIANA SEDE QUITO, Quito, 2019. Accessed: Mar. 27, 2025. [Online]. Available: https://dspace.ups.edu.ec/bitstream/123456789/17531/1/UPS%20-%20ST004133.pdf

[26] C. A. Pico Gutiérrez, “SISTEMA DE COMUNICACIÓN INALÁMBRICA PARA LA LECTURA Y TRANSMISIÓN DE DATOS EN EL CONTROL DE LAS RUTAS Y TIEMPOS DEL RECORRIDO DE BUSES URBANOS DE LA CIUDAD DE AMBATO,” UNIVERSIDAD TÉCNICA DE AMBATO, Ecuador, 2014.[Online]. Available: Accessed: Mar. 27, 2025. https://repositorio.uta.edu.ec/server/api/core/bitstreams/a2690324-c40c-4924-bd84-4fcac69a6d73/content

[27] L. A. Torres, “INTERFAZ DE SISTEMAS DE RADIOFRECUENCIA A BLUETOOTH,” UNIVERSIDAD INTERNACIONAL SEK, Ecuador, 2020. Accessed: Mar. 27, 2025. [Online]. Available: https://repositorio.uisek.edu.ec/bitstream/123456789/3844/1/Tesis.pdf

[28] Mercedes Marqués, *Bases de datos*. Universitat Jaume I Departament d’Enginyeria i Ciència dels Computadors, 2011. Accessed: Mar. 27, 2025. [Online]. Available: https://libros.metabiblioteca.org/server/api/core/bitstreams/bc680058-e43a-43f0-a857-78fe22f94e6b/content

[29] J. M. Lozano Banqueri, “CREACIÓN Y GESTIÓN DE UNA BASE DE DATOS CON MYSQL Y PHPMYADMIN,” Universidad de Jaén, España, 2018. Accessed: Mar. 27, 2025. [Online]. Available: https://crea.ujaen.es/server/api/core/bitstreams/12d06233-dde9-4058-8aa9-6971a0e1e6b6/content

[30] M. A. Bejarano Reyes and E. A. Manzano Ramos, “Implementación del prototipo de un sistema de monitoreo de signos vitales: pulso, temperatura y saturación de oxígeno para pacientes,” Perú, Jul. 2021. Accessed: Mar. 27, 2025. [Online]. Available: https://revistas.ulima.edu.pe/index.php/Interfases/article/view/5168/5452

[31] C. D. Tinoco Moreno and J. D. Velásquez Zora, “Diseño de un Sistema de Instrumentación Wearable para la Medida de la Saturación de Oxígeno en la Sangre,” Universidad del Norte, Colombia, 2022. Accessed: Mar. 27, 2025. [Online]. Available: https://manglar.uninorte.edu.co/bitstream/handle/10584/10644/Artículo.pdf?sequence=1&isAllowed=y

[32] D. P. Alvarado, “SISTEMA PARA LA MONITORIZACIÓN DE LA SATURACIÓN DE OXÍGENO EN SANGRE,” UNIVERSIDAD DE JAÉN Escuela Politécnica Superior de Linares, España, 2019. Accessed: Mar. 27, 2025. [Online]. Available: https://crea.ujaen.es/server/api/core/bitstreams/c01ebfb8-5279-43c8-8347-f825f5c67e53/content

[33] Reboll Meliá. Fernando, “Diseño y desarrollo de un dispositivo electrónico de uso personal para la monitorización continuada de la saturación de oxígeno en sangre,” UNIVERSIDAD POLITÉCNICA DE VALENCIA, España, 2021. Accessed: Mar. 27, 2025. [Online]. Available: https://riunet.upv.es/handle/10251/170478?show=full

[34] L. Lomelí Herrera, Aguayo Ríos. Federico, and Martínez Peláe. Rafael, “DISPOSITIVO TELEMÉTRICO PARA MONITOREO DE FRECUENCIA CARDIACA Y SATURACIÓN DE OXÍGENO,” *Pistas Educativas, Tecnológico Nacional de México en*

Celaya, vol. No. 128 (SENIE 2017), Feb. 2015, Accessed: Mar. 27, 2025. [Online]. Available:https://pistaseducativas.celaya.tecnm.mx/index.php/pistas/article/view/1145/990

[35] Neil Cameron, Arduino Applied Comprehensive Projects for Everyday Electronics, vol. 1. APRESS, 2018. Accessed: Mar. 27, 2025. [Online]. Available:https://link.springer.com/book/10.1007/978-1-4842-3960-5

[36] Toto Ronald Sánchez Cari, "Análisis y Proceso de Hardening de Servidor Virtual Web," *Universidad Mayor de San Andrés*, 2021.Accessed: Mar. 27, 2025. [Online]. Available:https://ojs.umsa.bo/ojs/index.php/inf_fcpn_pgi/article/view/82

[37] BRUNO CHAVARRIA NEIRA and EDISSON GUDIÑO DE LA A, "IMPLEMENTACIÓN DE UN SERVIDOR WEB Y UN DISEÑO DE UNA PÁGINA UTILIZANDO HERRAMIENTAS DE SOFTWARE LIBRE PARA EL DISPENSARIO 'SAGRADA FAMILIA' DE LA CIUDAD DE GUAYAQUIL," UNIVERSIDAD POLITECNICA SALESIANA, GUAYAQUIL, 2017. Accessed: Mar. 27, 2025. [Online]. Available:https://dspace.ups.edu.ec/bitstream/123456789/14162/1/GT001840.pdf

[38] Rawat, B., Purnama, S., & Mulyati, MySQL Database Management System (DBMS) On FTP Site LAPAN Bandung. International Journal of Cyber and IT Service Management (IJCITSM), 2021. Accessed: Mar. 27, 2025. [Online]. Available:https://media.neliti.com/media/publications/407145-mysql-database-management-system-dbms-on-9953a0f6.pdf

13. WEBGRAFÍA

[39] "FOTOPLETISMOGRAFÍA." https://repositorio.unal.edu.co/bitstream/handle/unal/62943/1012325930.2018.pdf?sequence=1&isAllowed=y (accessed Mar. 27, 2025).

[40] T. Cabrera, A. Moreno, and Colegio Cervantino, "Pulso Cardiaco." https://cmapspublic.ihmc.us/viewer/rid=1J9ZHPS9C-2C76R72-17D6 (accessed Mar. 27, 2025).

[41] ConceptoABC, "Redes inalámbricas." https://conceptoabc.com/redes-inalambricas/ (accessed Mar. 27, 2025).

[42] 247tecno, "¿Cómo funciona una base de datos?" https://247tecno.com/como-funciona-una-base-de-datos/ (accessed Mar. 27, 2025).

[43] COMUNICACIONES INALAMBRICASHOY, "ZigBee Green Power," Jul. 22, 2015. https://www.comunicacionesinalambricashoy.com/zigbee-green-power/ (accessed Mar. 27, 2025).

[44] MOKOBlue: Bluetooth original / BLE IoT & Fabricante de dispositivos inteligentes, "Diferentes versiones de Bluetooth:La aparición de Bluetooth 5.0," Mar. 19, 2021. https://www.mokoblue.com/es/bluetooth-versions-5-0/ (accessed Mar. 27, 2025).

[45] G. Chavarri. Movistar, "WiFi 6 y WiFi 5: en qué se diferencian y cuáles son sus ventajas," 2020, Accessed: Mar. 27, 2025. [Online]. Available: https://www.movistar.es/blog/amplificador-smart-wifi/diferencias-tecnologia-wifi-6/

[46] Raspberry Pi, "Raspberry Pi 4", Accessed: Dec. 08, 2022. [Online]. Available: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/ (accessed Mar. 27, 2025).

[47] Blog Microcontroladores, "Microcontroladores PIC y sus variedades," Nov. 18, 2012. https://microcontroladoressesv.wordpress.com/ (accessed Mar. 27, 2025).

[48] ELECTRONICLAB, "PSoC 5LP Prototyping Kit (CY8CKIT-059)." https://electronilab.co/tienda/psoc-5lp-prototyping-kit-cy8ckit-059/ (accessed Mar. 27, 2025).

[49] ARDUINO.CC, "Getting Started with Arduino," Dec. 07, 2022. https://docs.arduino.cc/learn/starting-guide/getting-started-arduino (accessed Mar. 27, 2025).

[50] UNIT ELECTRONICS, "MAX30102 Sensor Pulso Concentracion Oxigeno." https://uelectronics.com/producto/max30102-sensor-pulso-concentracion-oxigeno/ (accessed Mar. 27, 2025).

[51] LABSTER THEORY, "Principio de la espectrofotometría," Sep. 20, 2021. https://theory.labster.com/spectrophotometry_principle-es/ (accessed Mar. 27, 2025)..

[52] Oscarbui, "Robotica introducción," Dec. 2, 2013. https://es.slideshare.net/slideshow/robotica-introduccin-28823976/28823976 (accessed Mar. 27, 2025).

[53] Luis Llamas, Usar un teclado matricial con Arduino, https://www.luisllamas.es/arduino-teclado-matricial/ (accessed Dec. 07, 2022).

[54] MEGATRONICA ROBÓTICA ELECTRÓNICA COMPUTACIÓN INGENIERÍA, “Pantalla Oled 128×64 I2c Display Lcd 0.96 Azul VCC GND Arduino.”, https://megatronica.cc/producto/pantalla-oled-128x64-i2c-display-lcd-0-96-azul-vcc-gnd-arduino/ (accessed Mar. 27, 2025).

[55] Nettigo logo head upsc, “LCD TFT Display 1.8" SPI ILI9163.”, https://nettigo.eu/products/lcd-tft-display-1-8-spi-ili9163 (accessed Mar. 27, 2025).

[56] robotluk, “16x2 LCD Ekran I2C Modülü ile 1602 IIC/I2C.”, https://www.robotluk.com/(accessed Mar. 27, 2025).