Diseño e implementación de un prototipo de monitorización remota de pacientes que permita la recopilación de datos objetivos y subjetivos hacia la nube mediante el uso de Io-MT y Chatbot con el fin de optimizar el control de pacientes pertenecientes al grupo de atención prioritaria del centro de salud de la parroquia Antonio José Holguín

Torres Mata, Marlon Paúl

Departamento de Ciencias de la Energía y Mecánica

Carrera de Ingeniería Mecatrónica

Trabajo de Titulación, previo a la obtención del Título de Ingeniero Mecatrónico

Ing. Caizalitín Quinaluisa, Edwin Alejandro

3 de febrero de 2023

Latacunga

Reporte de Verificación de Contenido

Ouriginal by Turnitin

Document Information

Analyzed document	TESIS Torres Mata.pdf (D156838726)
Submitted	2023-01-24 20:34:00
Submitted by	
Submitter email	byron.corrales@utc.edu.ec
Similarity	2%
Analysis address	byron.corrales.utc@analysis.urkund.com

Sources included in the report

W	URL: https://prohealthware.com/es/difference-between-telemedicine-and-telehealth/ Fetched: 2023-01-24 21:23:00	3
W	URL: https://www.myrgroup.pe/blog/cual-es-la-diferencia-entre-telemedicina-teleasistencia-y-telesalud-6 Fetched: 2023-01-24 21:24:00	1
W	URL: https://www.muypymes.com/2020/10/12/curiosidades-origen-evolucion-historica-telemedicina Fetched: 2023-01-24 21:24:00	3
W	URL: https://rockcontent.com/es/blog/nginx/ Fetched: 2023-01-24 21:25:00	2
W	URL: https://scriptingmysql.wordpress.com/category/mysql-replication/ Fetched: 2023-01-24 21:25:00	2
SA	**TESIS-QUINTANILLA - URKUND.docx** Document TESIS-QUINTANILLA - URKUND.docx (D131554118)	1
W	URL: https://www.ilo.org/dyn/natlex/docs/ELECTRONIC/106626/130887/F-1857569878/RWA-106626.pdf Fetched: 2021-12-10 22:21:47	3
W	URL: https://www.healthit.gov/faq/what-telehealth-how-telehealth-different-telemedicine Fetched: 2023-01-24 21:23:00	1
W	URL: https://research-doc.credit-suisse.com/docView?language=ENG&format=PDF&sourceid=em&document_id... Fetched: 2023-01-24 21:24:00	1
W	URL: https://news.careinnovations.com/blog/what-is-telehealth-what-is-remote-patient-monitoring-how... Fetched: 2023-01-24 21:24:00	1

.....

Ing. Caizalitín Quinaluisa, Edwin Alejandro M.Sc.
Director

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

Departamento de Ciencias de la Energía y Mecánica

Carrera de Ingeniería Mecatrónica

Certificación

Certifico que el trabajo de titulación: **"Diseño e implementación de un prototipo de monitorización remota de pacientes que permita la recopilación de datos objetivos y subjetivos hacia la nube mediante el uso de Io-MT y Chatbot con el fin de optimizar el control de pacientes pertenecientes al grupo de atención prioritaria del centro de salud de la parroquia Antonio José Holguín"** fue realizado por el señor **Torres Mata, Marlon Paúl;** el mismo que cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, además fue revisado y analizado en su totalidad por la herramienta de prevención y/o verificación de similitud de contenidos; razón por la cual me permito acreditar y autorizar para que se lo sustente públicamente.

Latacunga, 3 de febrero de 2023

.....

Ing. Caizalitín Quinaluisa, Edwin Alejandro M.Sc.

C.C.: 0503351397

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

Departamento de Ciencias de la Energía y Mecánica

Carrera de Ingeniería Mecatrónica

Responsabilidad de Autoría

Yo, **Torres Mata, Marlon Paúl**, con cédula de ciudadanía n° 0503587537, declaro que el contenido, ideas y criterios del trabajo de titulación: **"Diseño e implementación de un prototipo de monitorización remota de pacientes que permita la recopilación de datos objetivos y subjetivos hacia la nube mediante el uso de Io-MT y Chatbot con el fin de optimizar el control de pacientes pertenecientes al grupo de atención prioritaria del centro de salud de la parroquia Antonio José Holguín"**, es de mi autoría y responsabilidad, cumpliendo los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Latacunga, 3 de febrero de 2023

.....

Torres Mata, Marlon Paúl

C.C.: 0503587537

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

Departamento de Ciencias de la Energía y Mecánica

Carrera de Ingeniería Mecatrónica

Autorización de Publicación

Yo, **Torres Mata, Marlon Paúl**, con cédula de ciudadanía N° 0503587537, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **"Diseño e implementación de un prototipo de monitorización remota de pacientes que permita la recopilación de datos objetivos y subjetivos hacia la nube mediante el uso de Io-MT y Chatbot con el fin de optimizar el control de pacientes pertenecientes al grupo de atención prioritaria del centro de salud de la parroquia Antonio José Holguín"** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Latacunga, 3 de febrero de 2023

.....

Torres Mata, Marlon Paúl

C.C.: 0503587537

Dedicatoria

A mis padres, Mery y Juan Carlos, sin su apoyo no hubiese podido estudiar esta carrera, este logro ha sido posible ya que desde muy pequeño me alentaron a cumplir mis objetivos y fueron un ejemplo de superación, día a día me hacen sentir afortunado de tenerlos como padres.

A mi abuela Dolores y a mi hermano Juan Carlos, por enseñarme que el amor es el regalo más grande que una generación puede dejar a otra, gracias a ustedes jamás me he sentido solo.

A mi amada Erika, por no dudar en brindarme su apoyo incondicional y por todo lo bueno que me ha dado, incluyendo nuestra pequeña familia que empieza a dar sus primeros pasos.

A mis abuelos, a mis tíos y primos por el amor que me demostraron y el esfuerzo que pusieron en cada una de las ocasiones en que necesité de su ayuda.

Marlon Paúl

Agradecimientos

A mis padres, por su disposición a verme cumplir mis sueños, brindándome todo cuanto ha estado a su alcance, muchos de mis logros se los debo a ustedes entre los cuales se incluye la culminación del presente proyecto.

A mi tutor de tesis Ing. Edwin Caizalitín que me brindó valiosos consejos demostrando genuino interés animándome y guiándome teórica y metodológicamente a lo largo del trabajo.

Al Dr. José Párraga, su papel como colaborador fue de vital importancia al poner a disposición su tiempo y conocimientos, sin lugar a duda su participación ha enriquecido el trabajo realizado.

A la Universidad de las Fuerzas Armadas ESPE-L, por todo el conocimiento que pusieron al alcance de mis manos a lo largo de estos años, todos quienes fueron participes de este proceso contribuyeron con un granito de arena en mi formación profesional, su apoyo se verá reflejado en la culminación de mi paso por la universidad.

Marlon Paúl

ÍNDICE DE CONTENIDOS

Carátula **1**

Reporte de Verificación de Contenido **2**

Certificación **3**

Responsabilidad de Autoría **4**

Autorización de Publicación **5**

Dedicatoria **6**

Agradecimientos **7**

Índice de Contenidos **8**

Índice de Figuras **13**

Índice de Tablas **24**

Resumen **26**

Abstract **27**

Capítulo I: Generalidades **28**

Introducción **28**

Antecedentes **30**

Planteamiento del Problema **32**

Justificación e Importancia **36**

Objetivos **38**

Objetivo General ***38***

Objetivos Específicos ***39***

Hipótesis **39**

Variables de Investigación **39**

Variable Independiente ***39***

Variable Dependiente ***39***

Capítulo II: Fundamentación Teórica **40**

Estado del Arte **40**

Monitorización Remota de Pacientes (RPM) **41**

Plataforma digital IoMT **42**

Visual Studio Code **43**

Google Cloud Platform **43**

Google Compute Engine ***44***

Protocolo MQTT **46**

EMQX **47**

MQTTX ***48***

Node.js **49**

NPM – Node Package Manager ***49***

Sequelize ***50***

NGINX **51**

ChatBot **52**

DialogFlow ***53***

Whatsapp-web.js ***55***

ESP 32 DevKit **56**

MySQL **57**

VUE.JS **58**

Parámetro Biométricos de interés **60**

Peso ***60***

Temperatura ***60***

Presión arterial ***61***

Frecuencia Cardiaca ***61***

Saturación de oxígeno en la sangre ***62***

Glucosa en sangre ***62***

Capítulo III: Diseño y Construcción **64**

Matriz QFD **64**

Análisis de la matriz QFD **68**

Diseño general del prototipo **69**

Diseño local de la plataforma web **70**

Selección de gestor de base de datos ***71***

Backend ***71***

Directorio assets. **71**

Directorio config. **72**

Directorio controllers. **73**

Directorio logs. **73**

Directorio Migrations. **74**

Directorio models. **75**

Directorio Routes. **76**

Directorio Seeders. **77**

Frontend ***78***

Directorio public. **78**

Directorio src. **79**

Configuración inicial del backend ***80***

Biblioteca MQTT en Node.js. **80**

Configuración inicial del frontend ***84***

Generación de tópicos para dispositivos. **86**

Framework para graficar datos. **88**

Bot de WhatsApp **96**

Integrar DialogFlow ***99***

Generar archivo de Credenciales. **102**

Migrar el proyecto al entorno de producción 108

Configurar la Máquina Virtual 110

Crear llaves SSH. 113

Conexión remota por medio de llaves SSH. 115

Levantar backend y chatbot 122

Elaboración de dispositivos para monitorización remota 138

Alimentación del circuito 145

Báscula 146

Diagrama flujo para el uso de la báscula 147

Generar caracteres especiales para la pantalla LCD de la báscula. 151

Diagrama de flujo de la programación de la báscula 153

Termómetro infrarrojo 156

Diagrama de flujo de la programación del termómetro infrarrojo 161

Oxímetro de pulso 164

Diagrama de flujo de la programación del oxímetro de pulso 169

Tensiómetro de muñeca 173

Diagrama de flujo de la programación del tensiómetro 177

Glucómetro 182

Diagrama de flujo de la programación del glucómetro 190

Capítulo IV: Implementación, pruebas y resultados 194

Resultados dela implementación de la báscula 194

Cálculo de errores para el peso en Kilogramos 197

Cálculo de errores para el peso en Libras 198

Resultados de la implementación del Termómetro infrarrojo 200

Cálculo de errores del termómetro infrarrojo con MLX90614 202

Cálculo de errores del termómetro infrarrojo comercial 204

Cálculo de repetibilidad de termómetro con MLX90614 205

Pruebas del sensor de temperatura en pacientes. 206

Resultados de la implementación del Oxímetro de pulso 209

Resultados de la implementación del Tensiómetro de muñeca 214

Resultados de la implementación del Glucómetro 222

Resultados de la implementación de la plataforma 226

Resultados de la encuesta: 227

Validación de la hipótesis 236

Análisis de costos 240

Capítulo V: Conclusiones y recomendaciones 241

Conclusiones 241

Recomendaciones 243

Bibliografía 245

Anexos 250

ÍNDICE DE FIGURAS

Figura 1 *Visual Studio Code* 43

Figura 2 *Servicios de Google Cloud Platform* 44

Figura 3 *Instancias de máquina virtual Google Compute Engine* 45

Figura 4 *Arquitectura PubSub del protocolo MQTT* 47

Figura 5 *EMQX dashboard* 48

Figura 6 *Interfaz de MQTTX* 48

Figura 7 *Documentación online del ORM Sequelize* 51

Figura 8 *Funcionamiento de NGINX* 52

Figura 9 *Ejemplo de funcionamiento de un Chatbot* 53

Figura 10 *Integración de servicios con DialogFlow* 54

Figura 11 *Generación de código QR* 56

Figura 12 *ESP 32 vs ESP 32 DevKit* 57

Figura 13 *Replicación asíncrona MySQL* 58

Figura 14 *Vue.js devtools* 59

Figura 15 *Matriz QFD* 67

Figura 16 *Diseño del prototipo sin especificar Software ni Hardware* 69

Figura 17 *Trabajar backend y frontend como proyectos distintos de VS Code* 70

Figura 18 *Carpeta assets* 72

Figura 19 *Archivo config.json* 72

Figura 20 *Archivo auth.js* 73

Figura 21 *Archivo error.log* 74

Figura 22 *Creación de tabla de usuarios bajo el sistema de migraciones* 74

Figura 23 *Modelo para la tabla devices* 75

Figura 24 *Tabla devices base de datos iot gestor MySQL* 76

Figura 25 *Archivo index.js* 76

Figura 26 *Creación y configuración de usuario Administrador* 77

Figura 27 *Archivos ubicados en el directorio raíz del backend* 77

Figura 28 *Index.html* 79

Figura 29 *Núcleo de trabajo frontend* 79

Figura 30 *Iniciar EMQX* 80

Figura 31 *Ejemplo de conexión de un cliente al broker Mosquitto con mqtt.js* 81

Figura 32 *Conexión del backend al broker EMQX con mqtt.js a nivel local* 81

Figura 33 *Previsualización archivo READM.md* 82

Figura 34 *Terminal integrado de VS Code* 82

Figura 35 *Construcción de tablas base* 83

Figura 36 *Arrancar el Backend* 83

Figura 37 *Previsualización archivo README.md* 84

Figura 38 *Dirección local e IP privada del proyecto a nivel local* 85

Figura 39 *Página de acceso a la plataforma* 85

Figura 40 *Página de administración de usuarios* 86

Figura 41 *Página de administración de dispositivos* 86

Figura 42 *Función a nivel global* 87

Figura 43 *Asignación de tópicos* 87

Figura 44 *Selección manual de versiones de Vue.js* 88

Figura 45 *Demos para visualización de datos amCharts 5* 89

Figura 46 *Código demo para Line Graph de amCharts 5* 89

Figura 47 *Ejemplo de función que genera números random de Amcharts 5* 90

Figura 48 *Line chart de valores random y Gauge chart con el último valor registrado* 90

Figura 49 *Display Breakpoints* 91

Figura 50 *Ejemplo de uso de breakpoints de Vuetify* 91

Figura 51 *Resultado del uso de breakpoints para el line chart* 92

Figura 52 *Resultado del uso de breakpoints para el Gauge chart* 92

Figura 53 *Resultado que se obtiene al no tomar en cuenta los breakpoints* 93

Figura 54 *Crear nuevo cliente en MQTTX* 93

Figura 55 *Configuración del cliente* 94

Figura 56 *Conectar cliente MQTTX* 94

Figura 57 *Configuración para intercambio de mensajes MQTTX* 95

Figura 58 *Envío de mensajes en formato JSON MQTTX* 96

Figura 59 *Página web oficial de WhatsApp-web.js* 96

Figura 60 *Abrir código QR en un navegador web* 97

Figura 61 *Generar código QR en URL estática* 98

Figura 62 *Ejemplo de intercambio de mensajes WhatsApp – Web .js* 99

Figura 63 *Página de descarga de Client Library de DialogFlow* 100

Figura 64 *Node.js Client Library* 100

Figura 65 *Intents para preguntas frecuentes de pacientes del centro de salud* 101

Figura 66 *Crear cuenta de Servicio* 102

Figura 67 *Otorgar permisos* 102

Figura 68 *Generar archivo de credenciales* 103

Figura 69 *Añadir imágenes a DialogFlow* 104

Figura 70 *Almacenamiento de imágenes para usar con DialogFlow* 105

Figura 71 *Obtener el link de una imagen de Pinterest* 105

Figura 72 *Envío de imágenes por WhatsApp con DialogFlow* 106

Figura 73 *Consulta de número de teléfono en Base de datos* 107

Figura 74 *Intercambio de información entre el Chatbot y el backend* 108

Figura 75 *Mensajes directos del bot de WhatsApp* 108

Figura 76 *Tecnologías involucradas en la elaboración de la plataforma* 109

Figura 77 *Instancias de Máquina Virtual* 110

Figura 78 *Configuración de la Máquina Virtual* 111

Figura 79 *Configuración del Disco de Arranque de la VM* 111

Figura 80 *Reglas de Firewall* 112

Figura 81 *Instancias de VM* 112

Figura 82 *Configuración de IP externa* 113

Figura 83 *IP externa estática* 113

Figura 84 *Claves SSH* 114

Figura 85 *Generar llaves SSH* 114

Figura 86 *Llaves SSH* 115

Figura 87 *Instancias de la máquina virtual “rpm”* 116

Figura 88 *Comando para establecer la conexión remota* 116

Figura 89 *Actualización del sistema operativo* 117

Figura 90 *Instalación de NGINX* 117

Figura 91 *Instalar npm en la instancia de VM* 117

Figura 92 *Extensión SFTP para VS Code* 118

Figura 93 *Crear directorios en la instancia de VM* 118

Figura 94 *Archivo de configuración SFTP* 119

Figura 95 *Cargar carpetas* 120

Figura 96 *Listar directorios cargados a la máquina virtual* 120

Figura 97 *Explorador de archivos de SFTP* 121

Figura 98 *Recuperar librerías en la máquina virtual* 121

Figura 99 *Instalar PM2* 122

Figura 100 *Levantar backend* 122

Figura 101 *Levantar servicios automáticamente* 122

Figura 102 *Compilación de producción* 123

Figura 103 *Página de bienvenida de Nginx* 124

Figura 104 *Abrir archivo de configuración de Nginx* 124
Figura 105 *Archivo de configuración por defecto de Nginx* 125
Figura 106 *Reiniciar Nginx* 125
Figura 107 *Acceso a la página web a través de la IP externa* 125
Figura 108 *Instalar EMQX en Ubuntu 20.04* 126
Figura 109 *Instalar MySQL en la instancia de VM* 126
Figura 110 *Configurar MySQL para escuchar tráfico externo* 127
Figura 111 *Reiniciar servicio de MySQL* 127
Figura 112 *Tabla de usuarios y Host default de MySQL* 128
Figura 113 Asignar contraseña a usuario root 128
Figura 114 *Modificar Host de usuario root en MySQL* 129
Figura 115 *Crear reglas de Firewall de Google Cloud* 129
Figura 116 *Apertura de puertos para MySQL* 130
Figura 117 *Extensión MySQL VS Code para administrar bases de datos* 130
Figura 118 *Configuración del administrador de base de datos* 131
Figura 119 *Crear base de datos en el entorno de producción* 131
Figura 120 *Migrar tablas en el entorno de producción* 131
Figura 121 *Crear usuario demo administrador* 132
Figura 122 *Acceso remoto a tabla users por medio de VS Code* 132
Figura 123 *Configurar CORS* 133
Figura 124 *Variables de ambiente* 133
Figura 125 *Abrir puerto MQTT* 134
Figura 126 *Abrir puerto API* 134
Figura 127 *Abrir puerto WebSocket* 135
Figura 128 *Reglas de Firewall de VPC* 135
Figura 129 *Reemplazar localhost por la IP externa en el archivo Chart.vue* 136

Figura 130 *Página web montada exitosamente en entorno productivo* 136
Figura 131 *Dispositivo de prueba para el Administrador* 137
Figura 132 *Configurar MQTTX para pruebas sobre IP pública* 137
Figura 133 *Simulación de sensores en el entorno de producción* 138
Figura 134 *Gráficas en tiempo real sobre instancia de VM de sensores simulados* 138
Figura 135 *Conectar ESP 32 a la red WiFi* 139
Figura 136 *Crear cliente MQTT* 140
Figura 137 *Definir dirección y puerto del Broker MQTT* 140
Figura 138 *Funciones del void setup()* 140
Figura 139 *Función callback* 141
Figura 140 *Parámetros que recibe la función callback* 141
Figura 141 *Definir arreglo para almacenar el mensaje* 142
Figura 142 *Deserialización de JSON a números enteros o flotantes* 142
Figura 143 *Deserialización de JSON a String* 142
Figura 144 *Función reconnect* 143
Figura 145 *Serialización con AruinoJSON.h* 143
Figura 146 *Publicar cadena JSON* 144
Figura 147 *Autenticación con JWT en EMQX* 144
Figura 148 *Parámetros para conexión con broker mediante autenticación JWT* 145
Figura 149 *Función para conexión con broker mediante autenticación JWT* 145
Figura 150 *ESP 32 alimentado por batería recargable* 145
Figura 151 *Circuito electrónico de la báscula* 148
Figura 152 *Tapa y base de báscula* 149
Figura 153 *Pata báscula* 149
Figura 154 *Orientación de construcción* 150
Figura 155 *Vista explosionada y lista de materiales báscula* 150

Figura 156 *Pantalla LCD 16X2* 151

Figura 157 *Representación en forma de matriz del carácter de una pantalla LCD* 151

Figura 158 *Generador de caracteres especiales para LCD alfanuméricas* 152

Figura 159 *Ejemplo de números formados a partir de caracteres especiales* 152

Figura 160 *Resultado de la elaboración de la báscula* 155

Figura 161 *Calibración de la báscula* 155

Figura 162 *Presentación de valor de peso con caracteres especiales* 155

Figura 163 *MLX90614* 156

Figura 164 *Codificación de sensores de la familia MLX90614* 157

Figura 165 *Ejemplos de modelos de sensores de la familia MLX90614* 158

Figura 166 *Circuito electrónico del sensor de temperatura infrarrojo* 159

Figura 167 *Modelo 3D termómetro infrarrojo* 160

Figura 168 *Vista explosionada y lista de materiales termómetro infrarrojo* 160

Figura 169 *Impresión de modelo 3D de termómetro infrarrojo* 163

Figura 170 *Resultado de la elaboración del termómetro infrarrojo* 163

Figura 171 *Señal de led IR para cálculo de oxígeno en la sangre* 165

Figura 172 *Circuito electrónico oxímetro de pulso* 166

Figura 173 *Modelo 3D oxímetro de pulso* 167

Figura 174 *Vista explosionada y lista de materiales del procesador del oxímetro* 167

Figura 175 *Vista explosionada y lista de materiales pinza oxímetro de pulso* 168

Figura 176 *Configuración para impresión 3D de la pinza del oxímetro* 168

Figura 177 *Prototipado rápido con impresión 3D* 170

Figura 178 *Verificar las ranuras del modelo 3D* 171

Figura 179 *Aplicar masilla plástica a impresión 3D* 171

Figura 180 *Pulir impresión 3D con lijas 400, 600 y 100* 171

Figura 181 *Resultado de pintar Impresión 3D con aerógrafo y pintura acrílica* 172

Figura 182 *Resultado de aplicar varias capas de barniz* 172

Figura 183 *Resultado final del ensamblaje del oxímetro de pulso* 172

Figura 184 *Curva de presión oscilatoria* 173

Figura 185 *Placa monitor de presión arterial comercial* 175

Figura 186 *Circuito electrónico del tensiómetro con IoT* 176

Figura 187 *Brazalete, válvula y bomba tensiómetro comercial* 179

Figura 188 *Vista explosionada y lista de materiales de la muñequera del tensiómetro* 179

Figura 189 *Vista explosionada y lista de materiales del procesador del tensiómetro* 180

Figura 190 *Impresión 3D y postprocesado del tensiómetro* 180

Figura 191 *Capa de pintura base sobre carcasa del tensiómetro* 180

Figura 192 *Pintura acrílica y laca protectora sobre carcasa del tensiómetro* 181

Figura 193 *Resultado del ensamble de la muñequera del tensiómetro* 181

Figura 194 *Resultado final del ensamblaje del tensiómetro* 181

Figura 195 *Diagrama de tira reactiva para medición de glucosa método de punción digital* ... 182

Figura 196 *Medida amperimétrica de glucosa mediante un circuito de potenciostato* 183

Figura 197 *Socket para tiras reactivas* 184

Figura 198 *Tira de prueba electroquímica en una configuración de contador* 184

Figura 199 *Circuito para adquisición de la señal de la tira reactiva* 185

Figura 200 *Curva cronoamperométrica típica de tira reactiva comercial* 186

Figura 201 *Circuito eléctrico Glucómetro IoT* 187

Figura 202 *Toma de muestras de glucosa* 188

Figura 203 *Curva para calibración de glucómetro* 189

Figura 204 *Vista explosionada y lista de materiales glucómetro* 191

Figura 205 *Resultado de la construcción del Glucómetro* 191

Figura 206 *Modelo 3D maletín de madera* 192

Figura 207 *Resultado de la elaboración del maletín de madera* 192

Figura 208 *Ensamble 3D del maletín de madera y los sensores* 193

Figura 209 *Resultado de la elaboración del maletín y los sensores* 193

Figura 210 *Báscula mecánica de columna del centro de salud* 194

Figura 211 *Toma de peso con báscula mecánica* 195

Figura 212 *Toma de peso con báscula de la plataforma* 195

Figura 213 *Pesajes en la escuela de la parroquia* 196

Figura 214 *Resultados del pesaje en la escuela de la parroquia* 196

Figura 215 *Precisión médica sensores versión MLX90614 DXX* 200

Figura 216 *Termómetro infrarrojo NX-200* 202

Figura 217 *Lectura de temperatura corporal vs termómetro comercial* 206

Figura 218 *Lecturas de temperatura corporal vs termómetro comercial* 206

Figura 219 *Toma de datos biométricos bebé de dos meses de edad* 207

Figura 220 *Toma de temperatura en el hogar del paciente el mismo día a las 10:00 pm* 207

Figura 221 *Toma de temperatura con termómetro infrarrojo comercial* 208

Figura 222 *Toma de temperatura en bebé de 6 meses con tos leve* 208

Figura 223 *Toma de temperatura en aulas con casos de infecciones respiratorias* 209

Figura 224 *Diagrama de dispersión métodos Ay B* 211

Figura 225 *Gráfica de Bland-Altman para oxímetro de pulso* 212

Figura 226 *Monitor de signos vitales con pedestal* 213

Figura 227 *Monitoreo de SpO2 en el centro de salud y en el hogar de la paciente* 213

Figura 228 *Presentación de parámetros biométricos del tensiómetro* 214

Figura 229 *Gráfica de dispersión presión sistólica* 216

Figura 230 *Gráfica de dispersión presión diastólica* 216

Figura 231 *Gráfica de Bland-Altman para presión sistólica* 217

Figura 232 *Gráfica de Bland-Altman para presión diastólica* 217

Figura 233 *Gráfica de dispersión de frecuencia cardiaca* 220

Figura 234 *Gráfica de Bland-Altman de frecuencia cardiaca* 220

Figura 235 *Toma de presión arterial por método auscultatorio en el Centro de Salud*..... 221

Figura 236 *Toma de presión arterial por método oscilométrico en el hogar del paciente* 222

Figura 237 *Tomada de concentración de glucosa en sangre*..... 225

Figura 238 *Pregunta 1. ¿Cuántos años tiene?* 227

Figura 239 *Pregunta 2. ¿Cuál es su género?* 227

Figura 240 *Pregunta 3. ¿Padece alguna enfermedad crónica?*..... 228

Figura 241 *Pregunta 4. ¿Forma parte del grupo de atención prioritaria?* 228

Figura 242 *Pregunta 5. ¿Tiene celular o computadora con conexión a internet?* 229

Figura 243 *Pregunta 6. ¿Por qué motivo utilizó el sistema de monitorización remota?* 230

Figura 244 *Pregunta 7. ¿El acceso a la página web le resultó fácil?*..... 230

Figura 245 *Pregunta 8. ¿Ha interactuado con un Chatbot con anterioridad?*..... 231

Figura 246 *Pregunta 9. ¿Cómo calificaría el Chatbot de la plataforma de monitorización remota?*..... 231

Figura 247 *Pregunta 10. ¿Recomendaría el uso de sistemas de monitorización remota a familiares o amigos basado en su experiencia?*..... 232

Figura 248 *Pregunta 11. ¿Le pareció fácil o intuitivo el uso de los sensores de la plataforma?*..... 232

Figura 249 *Pregunta 12. ¿Los sensores de la plataforma de monitorización remota tenían un aspecto confiable o de ser de buena calidad?* 233

Figura 250 *Pregunta 13. El tiempo empleado para tomar sus signos vitales con los sensores de la plataforma resultó ser:*..... 233

Figura 251 *Pregunta 14. El proceso de tomar sus signos vitales con los sensores de la plataforma fue:*..... 234

Figura 252 *Pregunta 15. ¿Ha recibido atención médica fuera de la parroquia?*..... 234

Figura 253 *Pregunta 16. ¿Basado en su experiencia con el uso de los sensores y sabiendo que se trata de prototipos, cree que el centro de salud debería intentar conseguir más sensores?* 235

Figura 254 *Pregunta 17. ¿Cómo afectó el uso de la plataforma de monitorización remota su opinión sobre la atención que brinda el centro de salud de la parroquia?*..... 235

Figura 255 *Entrada centro de salud Antonio José Holguín* 237

Figura 256 *Valor crítico para rechazar la hipótesis alternativa* 238

Figura 257 *Región de aceptación y de rechazo con respecto a* $H0$ 239

Figura 258 *Prueba t con herramienta de análisis de datos de Excel* 239

ÍNDICE DE TABLAS

Tabla 1 *Rangos normales de temperatura corporal para adultos y niños* 60

Tabla 2 *Valores de presión arterial* 61

Tabla 3 *Valores Normales de frecuencia cardíaca*..... 62

Tabla 4 *Niveles de glucosa en sangre por edades* 63

Tabla 5 *Lista de requerimientos de los clientes* 64

Tabla 6 *Especificaciones técnicas del sistema.*..... 65

Tabla 7 *Relación entre los requerimientos y las especificaciones técnicas* 66

Tabla 8 *Matriz de correlación*..... 66

Tabla 9 *Resultado matriz QFD*..... 68

Tabla 10 *Comparativa entre los principales gestores de bases de datos*. 71

Tabla 11 *Especificaciones técnicas módulo HX711*..... 146

Tabla 12 *Especificaciones técnicas MLX90614 ESF DCC* 158

Tabla 13 *Características Max30100 vs Max 30102* 164

Tabla 14 *Linternas ordenadas por longitud de onda*..... 165

Tabla 15 *Especificaciones técnicas MPS20N0040D*..... 174

Tabla 16 *Valores de voltaje del glucómetro IoT vs glucosa del glucómetro comercial* 188

Tabla 17 *Valores de peso para cálculos de error*..... 197

Tabla 18 *Valoración de repetibilidad báscula con HX711*..... 199

Tabla 19 *Valores de Ta y To para cálculo de error* 201

Tabla 20 *Valores de Ta y To tomados con termómetro infrarrojo comercial*..... 203

Tabla 21 *Datos para cálculo de repetibilidad MLX90614*..... 205

Tabla 22 *Tabla de datos para método de Bland-Altman y cálculo de errores del oxímetro*..... 210

Tabla 23 *Tabla de datos para método de Bland-Altman y cálculo de errores del oxímetro*..... 215

Tabla 24 *Datos para método de Bland-Altman y cálculo de errores de frecuencia cardiaca*... 219

Tabla 25 *Datos para método de Bland-Altman y cálculo de errores del glucómetro* 223

Tabla 26 *Gráfica de dispersión de concentración de glucosa en sangre* 224

Tabla 27 *Gráfica de Bland-Altman de concentración de glucosa en sangre* 224

Tabla 28 *Muestras para validación de hipótesis* 238

Tabla 29 *Análisis de costos* 240

Resumen

Este documento presenta la metodología usada para el diseño e implementación de un prototipo de monitorización remota de pacientes con la capacidad de recopilar datos objetivos y subjetivos en la nube a través del uso de Io-MT y Chatbot, con el prototipo se logró optimizar el control de pacientes pertenecientes al grupo de atención prioritaria del centro de salud de la parroquia Antonio José Holguín por medio de cinco sensores biométricos diseñados para medir la temperatura, peso, presión arterial, oxígeno en sangre y glucosa de los pacientes. Los sensores fueron seleccionados tomando en cuenta las afecciones crónicas más recurrentes en la población ecuatoriana. Los sensores permiten monitorear los parámetros fisiológicos de interés, es decir, obtener los datos objetivos, luego estos datos son enviados a un sistema montado en Google Cloud, se trata de una máquina virtual con Linux, los sensores envían los datos mediante WiFi a la máquina virtual que posee un bróker MQTT encargado de gestionar las conexiones a través de un modelo de publicación y suscripción, bajo este mismo modelo se comunica con un servidor web capaz de almacenar la información en base de datos, todo esto a excepción de los sensores compone el denominado backend del proyecto, el frontend por su parte está conformado por una página web diseñada para el manejo de los usuarios y sus dispositivos, cada usuario puede crear varios dispositivos para observar los datos que recibe el servidor web de los sensores que le han sido designados. Para obtener los datos subjetivos se creó un Chatbot de WhatsApp sobre la plataforma DialogFlow, la interacción con el usuario inicia cuando el servidor web envía por mensaje de texto los datos fisiológicos correspondientes junto con frases enfocadas en obtener la sintomatología del usuario. El Chatbot también tiene la capacidad de responder las dudas más frecuentes de los pacientes con respecto al uso del prototipo o los servicios del centro de salud.

Palabras clave: datos objetivos, datos subjetivos, MQTT, backend, frontend.

Abstract

This paper presents the methodology used to design and implement a prototype for remote monitoring of patients with the ability to collect objective and subjective data in the cloud through the use of Io-MT and Chatbot, the prototype allows for optimizing the control of patients in the priority attention group of the health center of the parish Antonio José Holguín through five biometric sensors designed to measure temperature, weight, blood pressure, blood oxygen level and blood sugar level. The sensors were selected taking into account the most recurrent chronic conditions in the Ecuadorian population. The sensors allow monitoring the physiological parameters of interest, the values shown by the sensors are known as objective data and these are sent to a system mounted on Google Cloud, this is a virtual machine with Linux, the sensors send the data over WiFi to the virtual machine that has an MQTT broker in charge of managing the connections through a publication and subscription model, the broker communicates with a web server capable of storing the information in a database through this same model, all this except for the sensors makes up the backend of the project, the frontend consists of a web page designed for the management of users and their devices, users can create several devices to observe the data received by the web server from the sensors that have been designated. A WhatsApp Chatbot implemented on the DialogFlow platform is responsible for collecting subjective data from patients, the interaction with the user starts when the web server sends by text message the corresponding physiological data along with phrases focused on obtaining the user's symptomatology. The Chatbot also has the ability to answer the most frequent doubts of patients regarding the use of the prototype or the services of the health center.

Key words: objective data, subjective data, MQTT, backend, frontend.

Capítulo I

Generalidades

Introducción

Los términos relacionados con la combinación de medicina y tecnología, es decir, telemedicina y telesalud, son indistinguibles pero la forma de entregar los servicios a través de diversos medios y su alcance permite diferenciarlos (ProHealth, 2018). La telemedicina se define como el uso de tecnologías de información y comunicaciones para brindar y respaldar la atención médica a larga distancia. Por otro lado, la telesalud es un concepto que abarca más áreas, es un fenómeno saludable que brinda educación, diagnóstico, tratamientos y también aumenta la conciencia sobre diversos problemas de salud (HealthIT, 2019).

La telemedicina solo define el uso de la tecnología para tratar a los pacientes, pero la telesalud también cubre servicios no clínicos (ProHealth, 2018). El término telesalud lo abarca todo ya que se refiere al desarrollo de información, educación y servicios sanitarios implementando todo tipo de tecnología. De hecho, la telemedicina generalmente está incluida en el alcance del término telesalud, debido a que la telesalud recurre al uso de una extensa gama de tecnologías y servicios para brindar atención a los pacientes y mejorar el sistema de atención médica en su totalidad (Romero, 2021).

La telesalud puede incluir servicios de salud como:

- Proveer educación y capacitación al personal de atención médica a través de interconsultas entre colegas profesionales que no estén disponibles a nivel local, en especial en zonas rurales
- Orientaciones para pacientes y cuidadores sobre un nuevo diagnóstico o nuevo medicamento
- Consejería en salud mental para ansiedad, depresión u otros problemas
- Campañas de salud

- Educación sanitaria
- Vigilancia
- Compra, venta y distribución de equipo especializado.

La distinción entre los dos términos también ha sido reconocida por la Organización Mundial de la Salud (OMS). Según la cual, la telemedicina es meramente responsable de examinar y tratar al paciente a través de tecnologías de información y comunicaciones. Claramente la telemedicina tiene un alcance más limitado al referirse específicamente a servicios clínicos remotos. La transmisión digital de imágenes médicas, el diagnóstico, las evaluaciones médicas a distancia y las video consultas con especialistas son ejemplos de telemedicina (Evisit, 2020).

Otro avance relevante es IoT o internet de las cosas, se refiere a cualquier elemento que posee sensores y conexión a internet con el objeto de intercambiar información con otros dispositivos o sistemas por medio de una dirección IP, un ejemplo de esto puede ser un automóvil con sensores que pueden informar del estado del vehículo por medio de una página web, otro ejemplo más moderno es el de los vehículos Tesla, un usuario puede abrir el auto usando la llave, un IPhone, un Apple Watch, un iPad, un iPod, entre otros, lo que demuestra el tipo de ecosistemas que IoT permite formar a través de la comunicación entre dispositivos. El número de dispositivos conectados a internet se incrementa cada año gracias a la reducción en el tamaño de los chips y al incremento del ancho de banda. Lo que reduce el costo de integrar objetos pequeños a la red (Gillis, 2022).

Con un serio retraso en comparación con otras industrias, la digitalización del sistema de atención médica se ha puesto en marcha, los programas piloto de Monitoreo Remoto de Pacientes comenzaron en la década de 1970 cuando el consorcio Kaiser Permanente creó sistemas de monitoreo para comunidades rurales con el fin de brindar una mejor atención médica a regiones aisladas. La literatura relacionada con RPM sugiere que las intervenciones basadas en modelos de comportamiento de salud, vías de atención y entrenamiento

personalizado conducen a los mejores resultados, sobre todo si se hace uso de IoT como herramienta de transferencia de datos y comunicación entre dispositivos lo que facilita el registro de datos y el desarrollo de interfaces de usuario más desarrolladas (Singh, Rice & Hernández, 2018).

Este método se utiliza principalmente para controlar enfermedades crónicas o afecciones específicas, como enfermedades cardíacas, diabetes mellitus o asma. Estos servicios pueden proporcionar resultados de salud comparables a los de un encuentro tradicional con el paciente en persona, proporcionan una mayor satisfacción a los pacientes y pueden ser rentables.

Antecedentes

Mediante revisión documental se encontró información en tesis que se relacionan con el tema investigativo, los cuales se muestran a continuación:

Sistema de ubicación y monitoreo de señales vitales en adultos mayores y personas con Alzheimer. En este trabajo se propone el diseño de un dispositivo electrónico portátil de bajo costo para apoyar en el cuidado y atención de adultos mayores, incluye un módulo GPS capaz de ubicar al usuario en cualquier instante, además de pequeños sensores que monitorean signos vitales como temperatura, presión arterial y frecuencia cardíaca. La información recolectada por los sensores es enviada a un servidor web en Linux Centos, se puede visualizar y gestionar a través de diferentes dispositivos terminales con WiFi. De esta manera se logra un control adecuado de las actividades del usuario (Jiménez, 2018).

Sistema ambulatorio para el control de signos vitales y prevención de la diabetes Mellitus. Propone el diseño de un dispositivo electrónico portátil para la prevención y el control de la Diabetes Mellitus en cualquier instante que el paciente requiera conocer su nivel de glucosa, el sistema ambulatorio incluye sensores, dispositivos y equipos para el control de signos vitales como presión arterial, pulso arterial y temperatura corporal. La información

recolectada es enviada a un servidor web implementado en una Raspberry Pi, la misma que se puede visualizar y gestionar a través de dispositivos con WiFi (Anchaluisa, 2018).

Prototipo e-Health basado en sistemas empotrados de bajo costo para monitoreo de signos vitales a través de internet. Prototipo armado con dispositivos low cost, específicamente un sensor de temperatura DS18B20 y un sensor de pulso, ambos conectados a una Raspberry Pi, capaz de ejecutar un proyecto desarrollado en Node.js que funciona como Backend y levantar un servidor MQTT encargado de transmitir información entre los publicadores y los suscriptores de un determinado tópico, así como también el consumo de una API de Twilio para enviar mensajes vía WhatsApp a los números registrados para alertas en la aplicación, por último para consumir todos estos servicios, una aplicación Web adaptativa para cualquier dispositivo desarrollado con el framework Angular 9, capaz de mostrar toda la información de los pacientes y permitir el monitoreo de los signos vitales (Haro, 2020).

Diseño y aplicación de un sistema de control y manejo telemático para personas con la enfermedad de diabetes. Efectúa el uso de una Raspberry Pi y un Arduino con el Shield e-Health Sensor V2.0, realizando monitoreo corporal mediante el uso de los diferentes sensores a disposición. El Shield posee diferentes tipos de conectividad disponibles, entre ellas Wi-Fi, 3G, GPRS, Bluetooth, 802.15.4 y ZigBee, presenta un módulo que integra las 3 placas y una pantalla táctil con ventilación y entradas para la conexión de los sensores. Los resultados se registran después de cada comida siguiendo el cronograma disponible en la página Web del proyecto (Castelo, 2021).

Sistema de monitoreo cardíaco portátil de una terminación para atención primaria o control médico remoto de pacientes con anomalías cardiacas. Presenta una alternativa económica de los ECG comerciales. El prototipo de sensor ECG utiliza el microcontrolador ESP32 de Espressif como núcleo de procesamiento y el módulo ADS1115 de Analog Devices como monitor cardíaco para detectar dos tipos de arritmias cardiacas como la bradicardia y taquicardia, para lo cual implementó un método de cálculo para la frecuencia cardiaca. La

interacción del usuario con el sistema de monitoreo cardiaco se realiza mediante una aplicación Android desarrollada específicamente para este proyecto. La App está implementada en lenguaje nativo para dar la fluidez necesaria en la gráfica de la señal ECG (Espinosa, 2021).

Sistema de telemedicina con monitoreo de signos vitales basado en IOT en un ambiente Smart TV. Se implementa un sistema de telemedicina utilizando IoT basándose en cuatro elementos: interfaz de usuario, administración, conectividad y sensores médicos. En la capa de interfaz de usuario se desarrolló una aplicación con el framework React-Native para el sistema Android TV la cual recibe los datos del sensor de BPMs y SpO2 MAX30100 a través de bluetooth embebido dentro de un circuito electrónico Wearable. Estos datos son enviados a través del protocolo TCP/IP y administrados mediante Odoo, alojado en una instancia EC2 de Amazon. La información almacenada puede ser desplegada a través de tablas o gráficas, alerta al usuario al recibir datos anormales y envía prescripciones a la aplicación de Android TV oportunamente (Garcés & Manzano, 2021).

Planteamiento del Problema

Según el INEC en el año 2019, las enfermedades isquémicas del corazón fueron la principal causa de muerte entre hombres y mujeres con 8574 muertes, debido a 35 diferentes variantes correspondientes al 11.7% del total de defunciones, debido en gran medida a la falta de monitoreo personalizado, a que los pacientes no pueden acudir constantemente a los chequeos y a que generalmente hacen dieta antes de cada cita médica lo que contamina los datos provocando que los diagnósticos sean susceptibles a equivocarse.

Le sigue en segundo lugar con el 6.7% del total de defunciones la Diabetes Mellitus con 4890 defunciones. En este caso en particular existen aparatos que permiten monitorear esta afección continuamente sin embargo se limitan a aplicaciones móviles o a registros integrados que son muy limitados, no le permiten al médico acceder a los datos en tiempo real y si los dispositivos se extravían o se dañan no existe la posibilidad de recuperar los registros.

La tercera causa de muertes en el país según el INEC en 2019 fueron las enfermedades cerebrovasculares con el 6.2% correspondiente a 4557 muertes. Los tensiómetros son frecuentemente usados con el fin de prevenir enfermedades cardiacas y tratar accidentes cerebrovasculares, al no implementar un sistema de IoMT que permita a los pacientes pertenecientes al grupo de atención prioritaria realizar este tipo de tareas desde la comodidad de sus hogares se ven obligados a acudir a hospitales y centros de salud únicamente para chequear su presión arterial perdiendo tiempo y dinero, por lo que empiezan a chequearse esporádicamente o cuando presentan complicaciones en su salud lo que desemboca en que el propio sistema de salud pierda recursos. En cuarto lugar, la influenza y neumonía representan el 5.6% del total de defunciones con 4096 y finalmente en quinto lugar se tiene que el 4.4% del total de defunciones se debieron a enfermedades hipertensivas con 3246 muertes.

Con estos datos está claro que las principales causas de muerte en el país son enfermedades crónicas que requieren monitoreo constante y acceso en tiempo real a los datos fisiológicos del paciente, pero estas prácticas no han sido integradas en el sistema de salud con graves consecuencias, el comportamiento de los registros de defunciones generales desde el año 2004 presenta una tendencia creciente hasta el año 2020, en este mismo año las enfermedades isquémicas del corazón continúan como la primera causa de muerte en el país con el 13.5% del total, pero este porcentaje en esta ocasión corresponde a 15639 defunciones.

A principios de 2020 inician los casos de COVID-19 en el país, la pandemia puso de manifiesto la necesidad de una rápida adopción de tecnologías sanitarias digitales innovadoras, que serían de utilidad frente a los escasos recursos disponibles que ocasionaron que para finales de 2020 el COVID-19 cuya cepa fue identificada se posiciona como la segunda causa de muerte entre hombres y mujeres a nivel nacional, con un total de 15490 defunciones que corresponden al 13.4% y en tercer lugar el COVID-19 cuya variante no pudo ser identificada con un 7.2% del total de defunciones es decir 8303 personas.

En cualquiera de los dos casos a principios de la pandemia se identificaban la mayoría de los infectados mediante los datos subjetivos de los pacientes, principalmente la pérdida del olfato y el gusto, sin embargo, por la falta de un sistema de Chatbot adecuado los pacientes tenían que asistir a una consulta médica en la que primero respondían preguntas de rutina con las cuales se podía identificar posibles casos de COVID 19, pero con el riesgo a contagiarse dentro del mismo hospital para los casos negativos y poniendo en riesgo su salud y la de los demás en los casos positivos, comportamiento que sigue vigente siendo uno de los motivos por los cuales no se logra superar del todo la pandemia.

La Diabetes Mellitus en cuarto lugar con un 6.8%, es decir, 7900 defunciones, en quinto lugar, la influenza y neumonía con 6930 muertes con el 6%, durante los últimos meses marcados por la pandemia de coronavirus, se han detectado casos de neumonía que todavía generan confusión en la comunidad médica, neumonías con la llamada hipoxia silenciosa. En la neumonía clásica, con frecuencia se da una falta de oxígeno conocida como hipoxia, que causa la falta de aliento, la fatiga e incluso la pérdida de conciencia. Sin embargo, ya son varios los casos de pacientes con coronavirus que, pese a tener niveles de oxígeno alarmantemente bajos, no presentan ninguna señal que alerte de su presencia. Dado que esta condición pasa inadvertida sin una oximetría, el paciente puede hablar, moverse y desenvolverse con normalidad, hasta que la falta de oxígeno produce el agotamiento de los pulmones y el fallo cardíaco, incluso sin haberse encontrado mal antes (Ratiopharm, 2020).

En sexto y séptimo lugar se encuentran las enfermedades hipertensivas con el 4.5% (5233 fallecidos) y las enfermedades cerebrovasculares con el 4.4% (5102 fallecidos). En resumen, para el año 2020 las siete primeras causas de muerte en el país son enfermedades que requieren monitoreo continuo y corresponden a un total de 64597 defunciones, número que se incrementa cada año debido a que no se implementan sistemas que permitan realizar el control de pacientes de manera más cómoda, económica y constante.

Con los datos estadísticos de los años 2019 y 2020 se deja en claro que la falta de un sistema que permita optimizar el control de pacientes producirá grandes pérdidas de los recursos del país y su población incluso superada la pandemia, además, la falta de sistemas tecnológicos de control de pacientes se refleja en la cultura de la población ecuatoriana ya que antes de la llegada del COVID-19 los índices de sobrepeso y obesidad ya eran alarmantes en las 24 provincias del Ecuador. Según el Programa Mundial de Alimentos (PMA), solo Napo tenía una tasa inferior al 50% en sobrepeso y obesidad en 2019.

Seis de cada diez adultos ecuatorianos tienen exceso de peso o son obesos y se prevé que hasta el 2030 el número de fallecidos por estas condiciones ascendería a 35671 por año en el país, 13000 más que la cifra actual, determinó el estudio. Sin embargo, estos índices empeorarían por el confinamiento, el teletrabajo, la teleeducación, restricciones como el cierre de gimnasios y parques y la mala alimentación provocados por la pandemia.

Si bien los datos y las estadísticas indican que el sobrepeso y obesidad ganan terreno en el país, la percepción ciudadana es otra. Según una encuesta realizada por Click Report, en enero de 2020, el 79% de los ecuatorianos consultados cree que su alimentación diaria es saludable. Además, solo el 30% dijo conocer a alguien con obesidad. Según el Instituto de Estadística y Censos (INEC), las principales causas de muerte en Ecuador se relacionan con malos hábitos alimenticios.

Todos los datos estadísticos y demás información presentada coincide con la situación actual del centro de salud de la parroquia Antonio José Holguín, los médicos a cargo corroboran que al igual que en el resto del país las afecciones crónicas son la principal causa de muerte en el sector, al ser una de las parroquias rurales con menor superficie territorial del cantón Salcedo, cuenta con muchas limitantes por la estructura de los suelos y sistemas de riego, diariamente la población económicamente activa se moviliza fuera del sector debido a sus empleos dejando en casa a adultos mayores, mujeres embarazadas, discapacitados, es decir, mayormente a personas pertenecientes al grupo de atención prioritaria, los cuales no es

de extrañarse que no acudan a sus chequeos de rutina, al no contar con el apoyo de una persona de confianza. Debido a que el centro de salud no cuenta con equipos que permitan la monitorización remota los médicos no pueden hacer más que comunicarse con los pacientes o sus familiares para recordarles que deben acudir con regularidad, pero con el tiempo gran parte de las llamadas empiezan a ser ignoradas.

Justificación e Importancia

El desarrollo del proyecto se justifica debido a que con la monitorización remota se podrá efectuar el control de los pacientes fuera de los entornos clínicos convencionales, como en el hogar o en un área distante, por lo que se puede aumentar el acceso y reducir los costos de atención médica. La implementación del prototipo implica el cuidado remoto constante de los pacientes por parte de sus médicos para rastrear síntomas físicos, afecciones crónicas o rehabilitación posterior a la hospitalización.

Anteriormente se detallaron las primeras causas de muerte general en el país, dejando de lado el COVID-19 se tratan de enfermedades crónicas que a su vez son el punto de interés de los sistemas de IoMT por lo que el prototipo sería de gran ayuda para los pacientes pertenecientes al grupo de atención prioritaria quienes tienen una mayor probabilidad de sufrir afecciones crónicas y a causa de la pandemia el valor de brindar este tipo de servicios a los pacientes que se esperaba redujeran los viajes y el contacto directo con los demás se hizo aún más evidente.

Ecuador tiene un alto porcentaje de zonas rurales marginales, al enfocar el proyecto en una parroquia rural se eliminarán las barreras más comunes que enfrenta este tipo de población, desde la proximidad, disponibilidad y acceso limitado a la atención médica hasta ingresos más bajos. La población que habita en estas zonas es campesina, se dedica a la ganadería y agricultura, vive agrupada en pequeñas comunidades y tiene un mayor porcentaje de adultos mayores, quienes son propensos a tener problemas de salud crónicos, más residentes sin seguro médico y limitados recursos que pueden ser optimizados gracias al

monitoreo constante de sus afecciones, también ayudaría a evitar y detectar otros problemas como el deterioro de su salud por el uso de fertilizantes o la crianza de animales sin las medidas de bioseguridad adecuadas. El centro de salud se encuentra ubicado en una parroquia de apenas 8 kilómetros cuadrados, el desarrollo del proyecto además de ser de utilidad para los pacientes, pondría en manifiesto los aspectos a tomar en cuenta a la hora de implementar tecnologías de IoMT en el sistema de salud pública del país.

La plataforma permitirá a los proveedores de atención médica diseñar un plan de control personalizado que facilite monitorear condiciones médicas específicas y comorbilidades. El plan de seguimiento se compone de actividades de salud de rutina programadas, lo que ayudará a los pacientes a aprender a reconocer los primeros síntomas de una exacerbación y a autocontrolarse. Al ejecutar actividades de salud los pacientes llegan a identificar las causas y los efectos de adoptar comportamientos más saludables, esto les ayuda a comprender sus signos o síntomas y a saber cuándo pedir ayuda, lo que les da una sensación de seguridad sobre su afección.

Mediante el prototipo el estado de salud se capturará desde el hogar del paciente y se actualiza en la plataforma, proporcionando información esencial para mejorar la toma de decisiones clínicas cuando los médicos ajusten y optimicen los tratamientos. Los cambios en los signos vitales pueden ser indicadores del deterioro de la salud de un paciente, mediante los sensores los signos vitales se monitorean y recopilan de cerca proporcionando a los pacientes y médicos indicadores de alerta temprana, lo que les da tiempo para revertir cualquier tendencia y prevenir el deterioro que conduce a hospitalizaciones y visitas evitables al departamento de emergencias.

En cuanto al alojamiento del proyecto, la mayoría de los médicos admiten el almacenamiento de datos basado en la nube. La transición del almacenamiento de datos a la nube significa que los datos estarán disponibles para los médicos en cualquier momento, independientemente del dispositivo o la ubicación. La flexibilidad para compartir los datos más

allá del cuidador y la institución actuales se volverá más importante a medida que la atención médica se vuelva cada vez más virtual. La nube también facilita una mejor integración, escalado y evolución de RPM a lo largo del tiempo, lo que ofrece la oportunidad de avanzar aún más en la digitalización de los procesos para el cuidado de los pacientes.

La integración del sistema en la nube ayudaría a disminuir la brecha entre la atención médica de un ecosistema descriptivo a uno en el que la atención médica sea predictiva, mediante la configuración de la nube, para lograr esto se necesitan grandes volúmenes de datos por lo que una vez que la afección del paciente sea diagnosticada mediante este sistema se podrían monitorear tanto los signos mediante RPM como los síntomas mediante Chatbot en momentos determinantes, como cuando la salud de los pacientes mejora o decae, con esto en un futuro se podrían implementar algoritmos de inteligencia artificial que permitan predecir el estado de salud de los pacientes e inclusive diagnosticarlos.

El prototipo también permitirá a los doctores familiarizarse con sistemas de IoMT, podrán elaborar protocolos para gestionar el tiempo que le dedican al monitoreo de sus pacientes a través de este tipo de servicios, la selección de uno o más dispositivos en función de las afecciones de cada paciente, a identificar a los pacientes que son elegibles para monitorización remota, a analizar los datos obtenidos y a establecer un plan de tratamiento a seguir en el paciente.

Objetivos

Objetivo General

Diseñar e implementar un prototipo de monitorización remota de pacientes que permita la recopilación de datos objetivos y subjetivos hacia la nube mediante el uso de Io-MT y Chatbot con el fin de optimizar el control de pacientes pertenecientes al grupo de atención prioritaria del centro de salud de la parroquia Antonio José Holguín.

Objetivos Específicos

- Recopilar información acerca de sistemas de IoMT actuales y procesos de control de pacientes basados en monitoreo continuo.
- Realizar pruebas de compatibilidad entre distintos tipos de comunicación inalámbrica y el protocolo MQTT.
- Seleccionar los sensores y el dispositivo MCU en base a sus características técnicas y la compatibilidad con los protocolos de comunicación establecidos.
- Diseñar la carcasa y los circuitos de los dispositivos.
- Intercambiar información entre los sensores y la base de datos localizada en la nube.
- Seleccionar un servicio de mensajería y elaborar el algoritmo del Chatbot sobre una plataforma que permita enviar los datos recopilados a la base de datos.
- Elaborar la página web con entornos diferentes para los administradores y los usuarios del prototipo.
- Implementar el prototipo y analizar los resultados obtenidos.

Hipótesis

¿El diseño e implementación de un prototipo de monitorización remota basado en el uso de IoMT y Chatbot permitirá optimizar el control de pacientes pertenecientes al grupo de atención prioritaria del Centro de Salud de la parroquia Antonio José Holguín a partir de la recopilación de sus datos objetivos y subjetivos?

Variables de Investigación

Variable Independiente

Prototipo para monitorización remota de pacientes

Variable Dependiente

Monitoreo y registro de los datos objetivos y subjetivos de los pacientes en tiempo real

Capítulo II

Fundamentación Teórica

Estado del Arte

Antes de que la Telemedicina existiera como tal, hubo algunos inventos que anticiparon la idea. Los primeros pasos de esta disciplina se dieron en 1883 con las experiencias de Étienne Marey, un médico francés que publicó el tratado “Mecanismos animales”. En ese tratado, Marey habla sobre un método para estudiar el vuelo de los pájaros basado en la transmisión a distancia de señales neumáticas y eléctricas (Olivares, 2021).

En 1890 Loeth inventa la mezcla entre un teléfono y un estetoscopio que se colocaba en el cuello de un enfermo y transmitía los ruidos de su laringe a un receptor que podía estar ubicado hasta a 600 millas del transmisor y distinguir ruidos de auscultación, este aparato se conoce como estetófono o estetotelefono (Pardell, 2021).

Williem Eithoven, se encontraba realizando estudios referentes a electrocardiografía para lo cual desarrolló un galvanómetro que aun que le permitió realizar aportes valiosos, ocupaba dos habitaciones, se calentaba mucho y pesaba 272 kilos por lo que lo instaló en el hospital más cercano y mediante un cable de 1.5 kilómetros transmitía electrocardiogramas al laboratorio de fisiología (Franceschini, 2020).

Los avances en el área de las telecomunicaciones y la tecnología cinematográfica de la época permitieron que para finales de 1950 y principios de 1960 fuera posible transmitir información más compleja como imágenes, videos y datos médicos relevantes, esto impulsó la entonces denominada video medicina, en 1959 los médicos de la universidad de Nebraska enviaban exámenes neurológicos por medio de televisión bidireccional y se acredita como el primer caso de telemedicina con video en tiempo real, tecnología que posteriormente se vinculó al hospital estatal. (Olivares, 2021).

El monitoreo remoto de pacientes como tal es esencialmente un producto de las tecnologías de la información y las telecomunicaciones del siglo XX. El monitoreo remoto ha

llegado al espacio, el estadounidense Alan Shepard voló en 1961 en la nave Freedom 7 la que poseía un ECG primitivo, un termómetro y un sensor de respiración conectado a un micrófono, para evaluar los datos se tenían registros previos al vuelo. Las primeras formas de RPM logradas se han complementado con videotelefonía, métodos de diagnóstico avanzados respaldados por aplicaciones distribuidas cliente / servidor y, ahora, RPM significa más que sólo exámenes, consultas o envío de información médica. Ahora los parientes pueden recibir atención médica desde sus hogares y seguir recibiendo atención de calidad (Wikimedia, 2019).

Monitorización Remota de Pacientes (RPM)

La monitorización remota de pacientes (RPM) es un método de prestación de atención médica, utiliza los últimos avances en tecnología de la información para recopilar datos de pacientes fuera de los entornos de atención médica tradicionales y se refiere al uso de tecnología para facilitar la interacción entre médicos y pacientes con un estado de salud estable. Sin embargo, no hay que dejarse engañar por el énfasis en la tecnología, los sistemas de monitoreo remoto de pacientes más efectivos, no se basan en los dispositivos médicos fríos, alienantes y cargados de cables del pasado, sino en productos de tecnología personal elegantes y amigables para el paciente (Hedges, 2021).

I**nternet of Medical Things (IoMT)**

Las aplicaciones de IoT en la atención médica van más allá, Healthcare IoT, también conocido como Internet of Medical Things (IoMT), permite la atención remota de pacientes, ha mejorado los procesos hospitalarios y farmacéuticos y ha aumentado la precisión de los datos. Se trata de una infraestructura conectada de dispositivos médicos, aplicaciones de software, sistemas y servicios de salud (Gates, 2021).

La adopción general de tecnologías IoT están beneficiando a muchas industrias, es una ola de herramientas basadas en sensores, que incluyen dispositivos portátiles y autónomos para el control remoto de pacientes y la combinación de dispositivos médicos conectados a

Internet con información del paciente lo que en última instancia distingue al ecosistema IoMT (Steger, 2020).

Plataforma digital IoMT

Una plataforma digital es una solución de software que permite realizar tareas, ofrecer distintos servicios o cubrir necesidades específicas de los usuarios a través de internet por medio de la ejecución de programas personalizados. De manera similar una plataforma digital IoMT es una solución de software que conecta hardware (dispositivos médicos), puntos de acceso y redes de datos, debe tener la capacidad de recolección, procesamiento, almacenamiento y visualización de datos en tiempo real.

Una plataforma Web IoMT debe tener las siguientes propiedades:

- Conectividad y normalización: hacer uso de diferentes tipos de formatos y protocolos que garanticen la transferencia de datos y conectividad de todos los dispositivos.
- Gestión de dispositivos: es necesario que se garantice que todos los dispositivos médicos conectados se encuentren funcionando correctamente.
- Base de datos: todos los datos obtenidos de los sensores son valiosos tanto para el sistema de salud como para cada uno de los pacientes, la base de datos debe almacenar y soportar el manejo de los datos.
- Procesamiento y sentencias de decisión: ejecutar eventos o acciones en base a los datos adquiridos por los sensores.
- Visualización: permite visualizar datos de interés a través de gráficas.
- Herramientas adicionales: puede ser análisis de datos, SDK, API, herramientas de testeo, inteligencia artificial, etc.

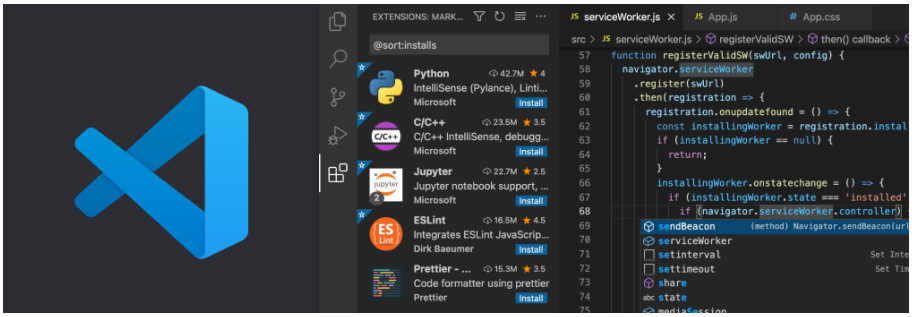
A continuación, se detallan las aplicaciones y tecnologías están involucradas en la implementación de la plataforma.

Visual Studio Code

También conocido como VS Code es un editor de código multiplataforma, se trata de un software gratuito y de código abierto que permite trabajar con diversos lenguajes de programación, entre sus características más destacables se tiene autocompletado de código, facilita trabajar con Git y otros proveedores SMC para el uso de control de versiones, es extensible y personalizable pueden agregar lenguajes de programación como python, C/C++, entre otros. Por medio de las extensiones se pueden instalar temas, depuradores y conectarse a servicios adicionales que se ejecutan en procesos separados por lo que el editor de texto no se ralentiza. Cuenta con una terminal integrada con capacidad para trabajar con cualquier Shell integrado como por ejemplo Bash o PowerShell.

Figura 1

Visual Studio Code



Nota. Sistema de búsqueda e instalación de extensiones de VS Code. Tomado de (VS Code, 2021).

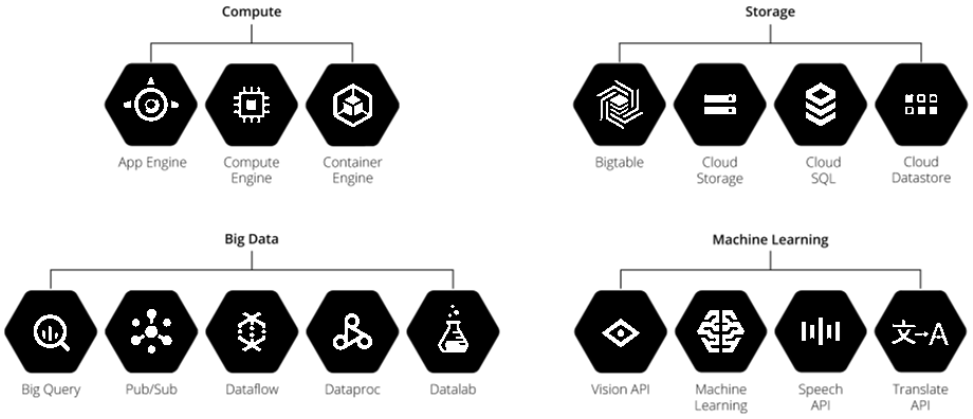
Google Cloud Platform

Es una plataforma que reúne el conjunto de servicios desarrollados por google y que se usaban dentro de la misma empresa, al contar con una gran infraestructura decidieron ofrecer estos recursos basados en la nube como una herramienta de gestión empresarial. Al enfocarse en estar siempre a la vanguardia, actualmente ponen a disposición más de 100 productos completamente dispares que van desde Machine Learning, DialogFlow, TensorFlow e inclusive Big Data, todo en un mismo sitio.

La computación en la nube permite almacenar y acceder a múltiples tipos de datos de forma remota, además, la infraestructura de Google ofrece altos niveles de seguridad y escalabilidad debido a sus porcentajes de fiabilidad y disponibilidad que se acercan al 100% al no contar con ningún periodo de inactividad programado.

Figura 2

Servicios de Google Cloud Platform



Nota. Listado de herramientas disponibles en Google Cloud según infraestructura, plataforma o software como servicio. Tomado de (Ordorica, 2020).

Google Compute Engine

Google Cloud ofrece a sus usuarios un amplio espectro de opciones de cómputo para correr aplicaciones, en algunas es suficiente con subir el código a la nube y por su gran nivel de gestión integrada la plataforma se encarga del resto, pero cuentan con un bajo nivel de personalización. Google Compute Engine por otro lado brinda total control y personalización sobre donde quiere el usuario que corran sus aplicaciones lo que permite sacar el máximo provecho a los recursos disponibles.

En Compute Engine Google ofrece máquinas virtuales, disco y red. Estos son los 3 pilares fundamentales del servicio, las máquinas virtuales pueden ser personalizadas a todos los niveles. Es posible configurar el CPU, RAM y disco de acuerdo a la plataforma que se vaya a implementar, todas ajustables de forma independiente, es posible incorporar tarjetas gráficas

e incluso tensor processing units para correr cargas de trabajo con machine learning y TensorFlow. Puede correr imágenes con los sistemas operativos Linux o Windows más habituales. Es responsabilidad del administrador gestionar la aplicación, Imágenes del SO, reglas de Firewall, balanceo y VPNs.

En cuanto a disco existen de dos tipos:

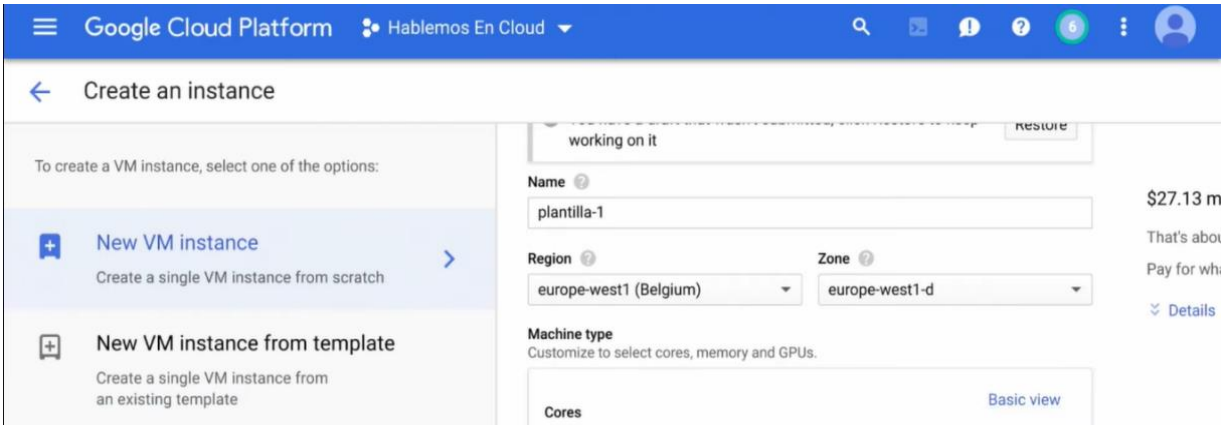
Discos persistentes: están disponibles en la red y se pueden configurar hasta un tamaño máximo de 64 Terabytes de almacenamiento, dependiendo de las necesidades de la carga de trabajo pueden ser magnéticos o de estado sólido, son independientes de las máquinas virtuales y en modo lectura varias máquinas virtuales pueden leer en la misma unidad y son redimensionables en cuanto a su rendimiento y capacidad.

Discos locales SSD: Se usan cuando la carga de trabajo necesita la menor latencia posible. Se pueden usar hasta 8 discos con hasta 375 Gigabytes de almacenamiento cada uno los cuales están atados al ciclo de vida de la máquina virtual.

Google Compute Engine conecta las máquinas virtuales a su red definida por software, que tiene alcance global y da dicho alcance a todas las aplicaciones.

Figura 3

Instancias de máquina virtual Google Compute Engine



Nota. Elaboración de plantilla de MV desde cero, personalización de la capacidad de la máquina y la ubicación en la que se va a montar la máquina.

Protocolo MQTT

En base al modelo de capas TCP/IP en donde cada capa realiza una tarea específica para mover los datos a través de la red, MQTT se trata de un protocolo da capa 5, se ha vuelto sumamente popular en aplicaciones de IoT debido a su eficiencia y ligereza que lo hace ideal para el uso en microcontroladores pequeños y de baja potencia, por ejemplo, el hecho de que requiere una cantidad mínima de código y energía.

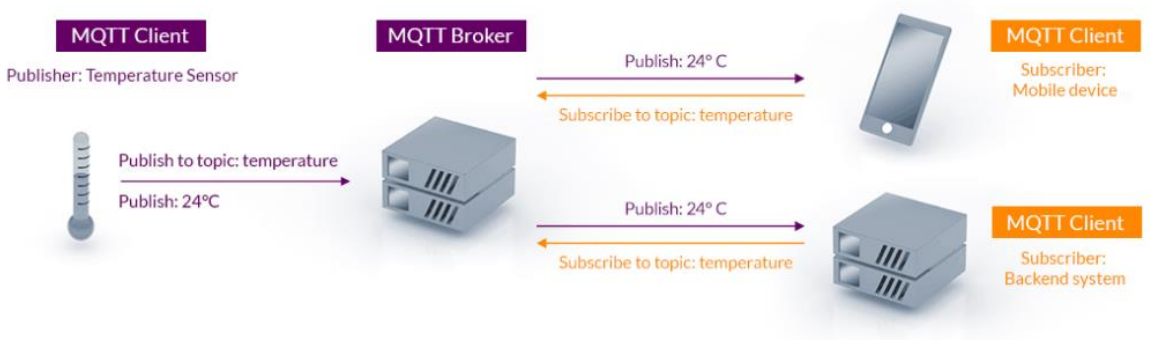
Gracias a que fue diseñado para conexiones con áreas remotas es ideal para lugares con un ancho de banda limitado, dispositivos con recursos restringidos o incluso aplicaciones, trabaja con el modelo publicador/suscriptor, un suscriptor puede convertirse en publicador y viceversa, por lo que es necesario un bróker que facilite las conexiones y gestione el manejo y los permisos de cada usuario mediante un filtro que se denomina Topic, básicamente el bróker es un servidor que recibe y redirige mensajes, para ello los mensajes cuentan con topics organizados jerárquicamente y el bróker hará llegar el mensaje a todos los clientes que se encuentren suscritos a un determinado topic.

MQTT permite el intercambio de información entre la nube y sensores ubicados en áreas remotas en tiempo real, esto también facilita la trasmisión de mensajes entre múltiples dispositivos y la escalabilidad, otra de sus características clave son las sesiones permanentes, cuando un cliente se desconecta de la red, almacena la información necesaria para que vuelva a conectarse, esto se puede usar en conjunto con el llamado testamento que se encarga de notificar al sistema cuando un cliente en específico se a desconectado de la red sin permiso.

Finalmente, el protocolo cuenta con 3 niveles de calidad de servicio que se encargan de verificar que los mensajes sean entregados y otra característica interesante es que mediante el método de mensajes retenidos permite configurar un mensaje para que se envíe a los clientes en el momento en que se suscriban a un nuevo tema.

Figura 4

Arquitectura PubSub del protocolo MQTT



Nota. Ejemplo de arquitectura Publicador/Suscriptor, un cliente no necesariamente debe ser un dispositivo, puede tratarse de interfaces o aplicaciones. Tomado de (MQTT, 2022).

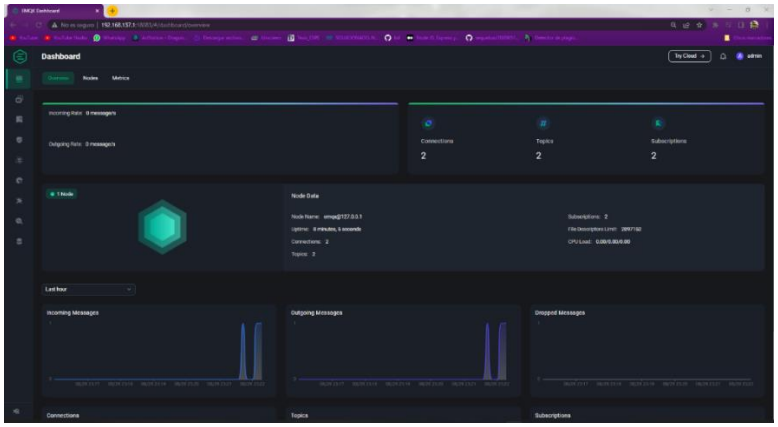
EMQX

Es una plataforma nativa de la nube que utiliza tanto tecnologías de agrupación como protocolos IoT de código abierto, se usa para gestionar el envío de datos de sensores a la nube y para activar desde la nube los algoritmos de control de los dispositivos. En su versión enterprise puede manejar hasta cien millones de clientes simultáneos en tiempo real con una latencia menor a un milisegundo. Es compatible con MQTT 5.0 lo que garantiza la escalabilidad y seguridad de los ecosistemas, pone a disposición varias funciones, entre las más importantes el soporte TCP/SSL, compatibilidad con el protocolo MQTT y WebSockets, soportes de suscripción, autenticación ID, dirección IP, nombre de usuario y contraseñas, compatibilidad con MySQL, límite de tasa de mensajes y conexión, etc.

Permite configurar el intervalo de caducidad de mensajes y sesiones, propiedades de usuarios y el motor de reglas que es uno de los parámetros fundamentales ya que se encarga de vigilar el envío y recepción de los mensajes, así como del control de eventos de cada dispositivo, de su correcta configuración depende la integración del proceso y simplifica mucho el uso de la plataforma por parte de los usuarios.

Figura 5

EMQX dashboard



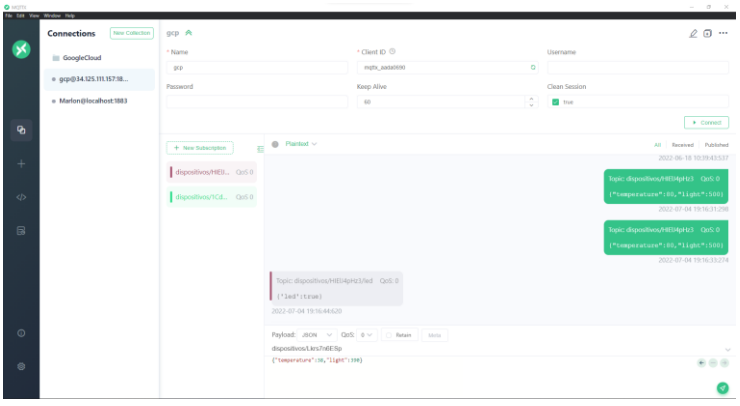
Nota. EMQX dashboard permite gestionar y monitorizar los dispositivos conectados y ofrece una interfaz desde la cual es fácil configurar el motor de reglas y se monta por defecto sobre el puerto 18083.

MQTTX

EMQX pone a disposición de sus usuarios una aplicación gratuita y multiplataforma que es básicamente un cliente de escritorio en forma de chat, sirve para probar conexiones bajo el protocolo MQTT y el envío y recepción de mensajes bajo el modelo Pub/Sub y es de utilidad para simular sensores ya que admite múltiples formatos de carga útil para intercambiar mensajes como JSON, Hex, Base64, etc.

Figura 6

Interfaz de MQTTX



Node.js

Es un entorno de tiempo de ejecución o runtime environment multiplataforma y de código libre creado para correr JavaScript, que a su vez se define como un lenguaje de programación orientado a objetos y que se encarga de dotar de un mayor dinamismo a las páginas web, lo que lo hace ideal para presentar datos en tiempo real y gracias a Node.js ahora se puede ejecutar tanto en un navegador como en un servidor, además es compatible con el protocolo MQTT que es usado por la mayoría de las aplicaciones de IoT.

Otro beneficio de usar Node.js es que es compatible con WebSockets, básicamente es una API sobre la que trabaja el protocolo MQTT para el transporte y encapsulamiento de mensajes al ser un método que proporciona comunicación full dúplex por medio de una única conexión TCP, esto también ayuda a desarrollar varias aplicaciones pequeñas que se comuniquen entre sí en lugar de un solo proceso con mucho código.

Se basa en V8 Engine de Google que convierte las instrucciones de JavaScript en código máquina que la computadora puede ejecutar más rápidamente. Sin embargo, casi ninguna función realiza I/O directamente lo que provoca que el proceso nunca se bloquee, haciéndolo ideal para manejo de datos en tiempo real, que al combinarse con callbacks que permiten múltiples conexiones simultáneas lo hacen propicio para desarrollar sistemas escalables, por cada conexión se activa un callback específico, pero si no existen conexiones activas Node.js se dormirá, esto es muy importante en aplicaciones de monitorización remota ya que solo se activar cuando un sensor se encuentre enviando información o cuando se requieran lanzar protocolos de control a los dispositivos lo que lo vuelve sumamente eficiente.

NPM – Node Package Manager

Facilita la instalación, actualización y eliminación de los más de 1.4 millones de paquetes disponibles, por medio de su repositorio con más de 27 mil millones de descargas semanales y gestiona el control de versiones lo que permite establecer la versión exacta de un

paquete que se requiere instalar únicamente aumentando un @ junto con la versión deseada, por lo que un paquete específico se puede instalar mediante una única línea de código.

NPM juega un papel fundamental en el desarrollo de aplicaciones de IoT con Node.js, contiene más de 80 paquetes relacionados con Arduino, Bluetooth Low Power y varios microcontroladores de baja potencia, posee un paquete de IoT que se usa para el cálculo de especificaciones de productos IoT, también tiene más de 30 paquetes que son útiles para varios sensores y otras herramientas de desarrollo de IoT, además de varias APIs que proporcionan transiciones de datos fluidas, volviéndolo excelente para desarrollos de alto nivel en tiempo real, permite instalar ORMs para el manejo de bases de datos y todo tipo de paquetes que son útiles para disminuir el tiempo de desarrollo de una aplicación.

Sequelize

Es un ORM o Object-Relational Mapping basado en Node.js, su trabajo es convertir los datos u objetos de una aplicación a un formato adecuado para ser almacenado en bases de datos relacionales y viceversa, también ofrece otros beneficios como comandos que permiten crear y actualizar tablas directamente desde la aplicación, posee mecanismos de migración de bases de datos, sincronización, asociaciones, reduce el tiempo de desarrollo de aplicaciones y previene a las páginas web de ataques a la base de datos por medio de sentencias SQL maliciosas.

Sequelize existe desde el año 2011 y actualmente es usado por un gran número de aplicaciones, esto es una prueba de su estabilidad, además de que cuenta con mucha documentación disponible, debido a que se basa en promesas facilita la administración de funciones asíncronas, es compatible con Node.js v10 o superior y está disponible a través de npm.

Figura 7

Documentación online del ORM Sequelize

Nota. Sequelize permite gestionar bases de datos desde cualquier aplicación desarrollada en Node.js. Tomado de (Sequelize, 2022).

NGINX

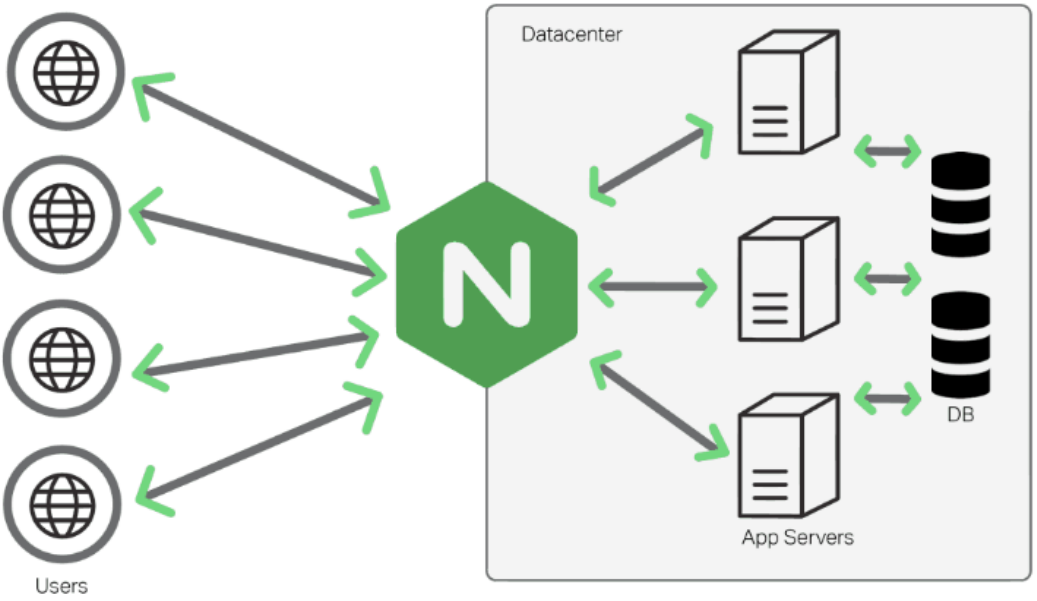
Es un servidor web de código abierto, trabaja con una arquitectura asíncrona y controlada por eventos, estas características lo convierten en uno de los servidores más deseables para aplicaciones que requieren velocidad y escalabilidad, tiene gran capacidad para manejar múltiples conexiones, algo que es recurrente en sitios web de alto tráfico.

Cuando un usuario solicita una página web el servidor web es el encargado de proporcionar la información que debe ser mostrada en dicha página al navegador. Los servidores tradicionales crean un hilo para cada solicitud, pero NGINX administra hilos similares bajo un proceso de trabajo que a su vez contiene unidades más pequeñas denominadas conexiones de trabajo y cada conexión de trabajo puede gestionar hasta 1024 solicitudes que sean similares sin ninguna dificultad, por lo que es muy usado para sitios web con mucho tráfico como almacenamiento en la nube.

Desde sus inicios se enfocó en la optimización de rendimiento bajo escala y un bajo uso de memoria y ahora es conocido por su rendimiento, estabilidad, funciones y fácil configuración. Por medio de la función de balanceó de carga si un servidor se satura es posible desviar las funciones a otros servidores previamente configurados. Es usado por grandes compañías como Netflix, GitHub, Heroku, entre otras.

Figura 8

Funcionamiento de NGINX



Nota. El software NGINX no posee un buen rendimiento en Windows, en comparación con Linux u otros sistemas operativos que soportan Unix. Tomado de (Higuerey, 2020).

ChatBot

Son programas capaces de simular una conversación con personas reales, hoy en día se utilizan en todo tipo de aplicaciones y a todo nivel, van desde asistentes capaces de responder a las inquietudes de los usuarios mediante mensajes automatizados hasta sistemas de IA que se construyen con palabras claves que se basan en las interacciones más comunes de los usuarios y que cuentan con herramientas de análisis de datos, aprendizaje profundo y acceso a bases de datos por medio de internet.

Figura 9

Ejemplo de funcionamiento de un Chatbot

Nota. Un Chatbot sofisticado con acceso a la base de datos de una empresa puede automatizar además de las conversaciones otros procesos como reservaciones, cancelaciones ventas, facturación, etc. Tomado de (Cornieles, 2019).

DialogFlow

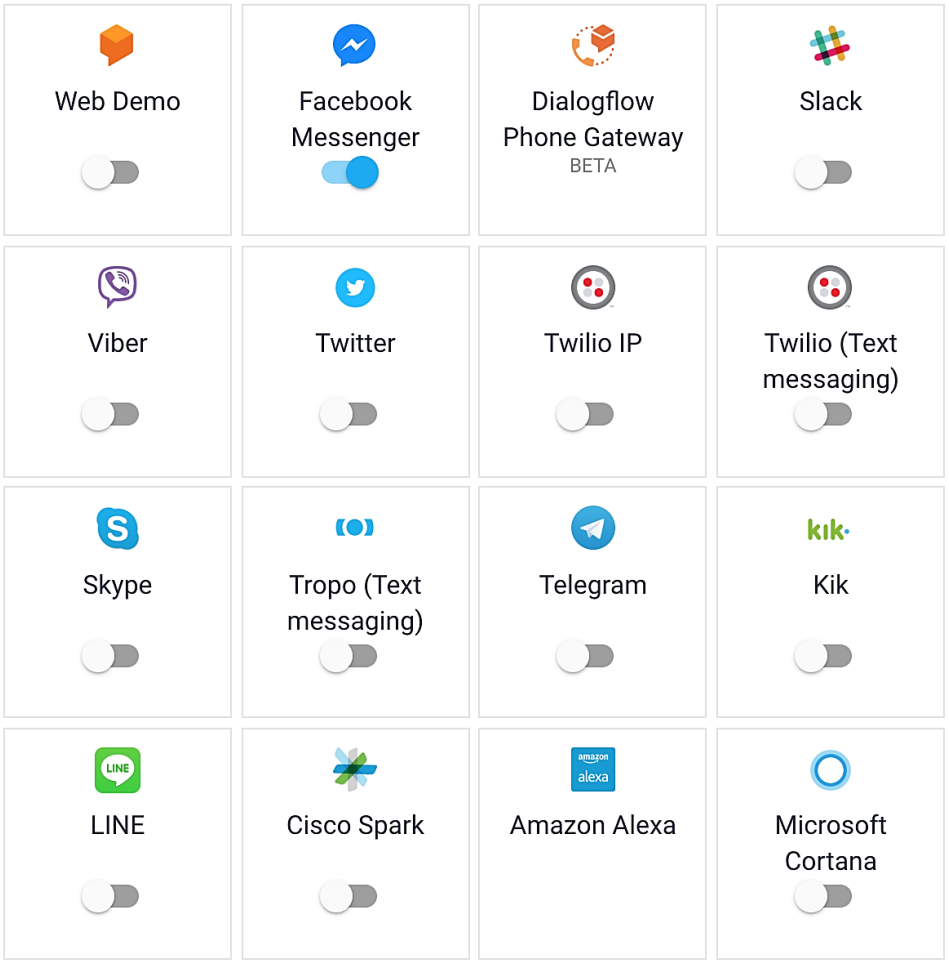
Es la herramienta que permite construir asistentes virtuales de voz y texto, es un servicio de Google que permite construir interfaces conversacionales de manera sencilla. Ayuda a configurar la construcción de un agente, desde la parte de gestión de la conversación mediante un motor de compresión de lenguaje natural hasta la parte de integración con diferentes canales y aplicaciones propias para obtener respuestas customizadas para usuarios específicos. Su IA de procesamiento de lenguaje natural se puede entrenar con muy pocos ejemplos, además, cuenta con agentes pre configurados que pueden usarse como base para construir un Chatbot lo que se ve reflejado en la velocidad con la que se puede implementar.

Un bot o agente se divide en intenciones, cada intención se entrena por separado y su objetivo es dar una respuesta clara a los usuarios sobre temas específicos. DialogFlow cuenta con múltiples opciones de integración para incorporar aplicaciones propias al asistente,

permitiendo monitorear datos relevantes en tiempo real y de este modo enviar respuestas más completas automáticamente o ejecutar algoritmos propios de la aplicación, esto se puede hacer llamando a un endpoint o utilizando Cloud Functions que esencialmente permiten ejecutar código sin tener que gestionar ningún servidor de modo que se pueden construir respuestas mucho más complejas incorporando un CRM o una base de datos.

Figura 10

Integración de servicios con DialogFlow



Nota. De la mayoría de servicios, DialogFlow requiere solo un token de acceso, varios servicios como WhatsApp que son compatibles no se muestran en la interfaz de DialogFlow pero se puede incorporar con herramientas como Node.js. Tomado de (Grabowski, 2018).

Dispone de herramientas embebidas de monitorización y análisis que permiten ver el uso que los usuarios le dan al Chatbot y tener un punto de partida para mejorarlo, se recogen todas las locuciones que hacen los usuarios con el agente e indica cuáles han sido mapeadas y cuáles no, de forma que podamos ver si es necesario volver a entrenar a un agente o si los usuarios se ven interesados en una intención que no se ha incluido. Además, muestra el flujo que siguen los usuarios a través de los diferentes pasos de una conversación.

Whatsapp-web.js

Es una librería de Node.js, básicamente es una API que se conecta a través de la versión de navegador de WhatsApp y que se apoya sobre otra API de alto nivel que se llama Puppeteer y que permite automatizar acciones sobre los navegadores de Google Chrome y Chromium, se integra para correr una instancia real de WhatsApp Web evitando así ser bloqueado, ya que dicha empresa no permite bots ni clientes no oficiales de la plataforma.

Requiere Node.js v12 o superior y debido a su relación con Puppeteer para poder instalarlo en un sistema operativo sin GUI como Linux server, al que solo se puede acceder por medio de un Shell, es necesario emular el navegador Chromium, por lo que es más fácil trabajar con Windows.

Whatsapp-web.js por defecto no mantiene abierta ninguna sesión, es decir que cada vez que el dispositivo se desconecte de la red es necesario volver a escanear el código QR que proporciona la versión de navegador de WhatsApp, de modo que si se desea conservar la sesión debe usar alguna de las estrategias disponibles que ofrece la misma biblioteca o crear unas propias que permitan guardar las credenciales de inicio de sesión, cabe recalcar que cada estrategia disponible para restaurar sesiones tiene sus ventajas y desventajas por lo que será necesario elegir en función del uso que se va a dar a la aplicación.

Figura 11

Generación de código QR

Nota. La librería de WhatsApp-Web.js funciona ejecutando la versión para navegador de WhatsApp en segundo plano por lo que los clientes se autorizan escaneando el código QR que genera para el inicio de sesión. Tomado de (López, 2022).

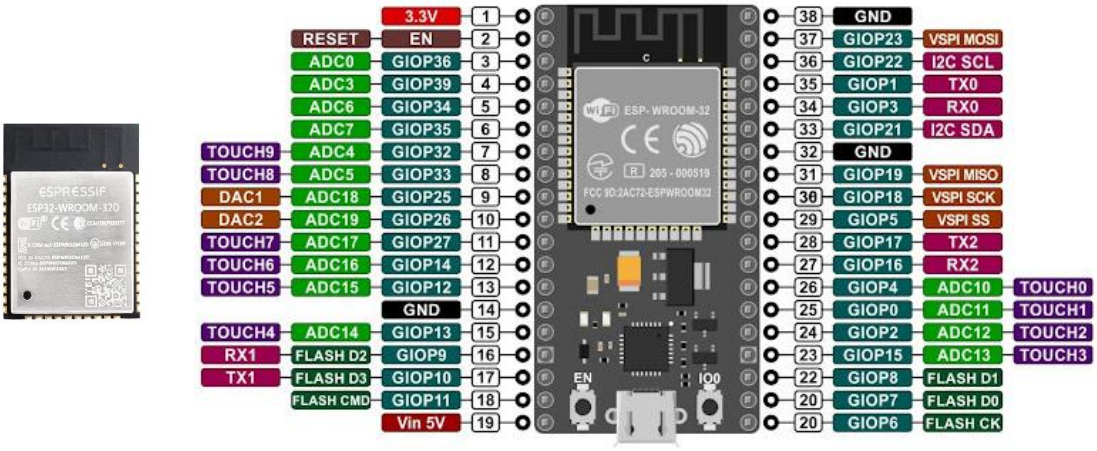
ESP 32 DevKit

Es la mejora de la placa de desarrollo ESP8266, tiene características superiores, se puede programar desde el IDE de Arduino por lo que una persona que esté familiarizada con este tipo de placas puede cambiarse a la ESP 32 de forma inmediata, al tener Wi-Fi y Bluetooth integrados con componentes RF prácticamente no necesita módulos ni componentes externos adicionales, es una placa diseñada a partir de una ESP 32 que facilita su uso para elaboración de prototipos (existen otros modelos con el mismo microprocesador pero con periféricos destinados a aplicaciones más específicas), no solo solventa las falencias de su predecesora sino que incluye características adicionales de última generación a bajo costo. Están disponibles con uno o dos núcleos por lo que pueden trabajar hasta en dos procesos en

paralelo, 4 tipos de comunicaciones como I2C o SPI, 16 salidas PWM, 2 ADC SAR DE 12 bits hasta 18 canales, 2 DAC de 8 bits, sensor de efecto Hall.

Figura 12

ESP 32 vs ESP 32 DevKit



a) b)

Nota. a) Microprocesador ESP 32 Wroom. b) Placa de desarrollo ESP 32 DevKit de 38 pines. Tomado de (Asanza, 2021).

Trabaja con Wi-Fi 802.11n a 2.4 GHz que admite hasta 150 Mbits/s y Bluetooth2 BR/EDR y BLE, consumo de energía ultra bajo, entre otras peculiaridades que la hacen ideal para aplicaciones de IoT y monitoreo remoto.

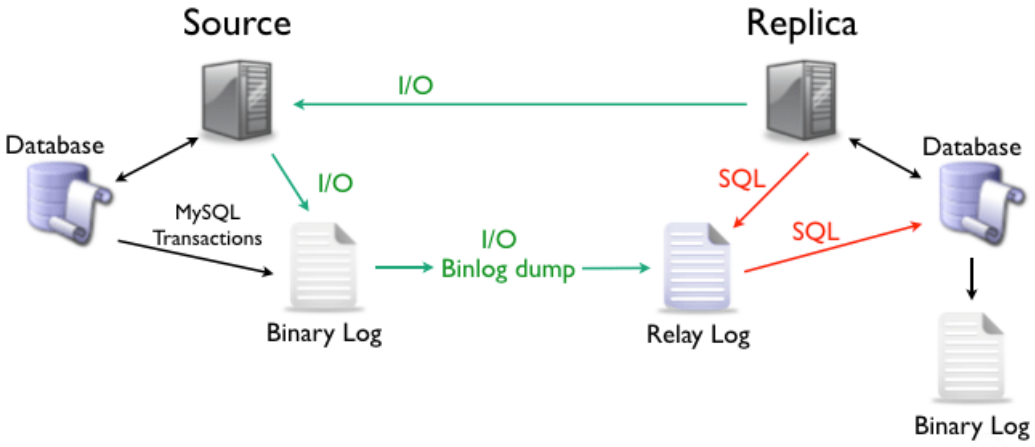
MySQL

Es una de las bases de datos de código abierto más usadas a nivel mundial, funciona bajo un modelo cliente/servidor, con un servidor multiproceso compatible con una gran cantidad de programas, bibliotecas de distintos clientes y muchas APIs mediante las cuales se puede acceder a la base de datos desde aplicaciones escritas en diferentes lenguajes lo que hace ideal a MySQL para su uso en máquinas virtuales montadas en la nube, al no tener necesidad de instalar ningún GUI o complementos es más rentable mantener en línea un sistema.

Se sabe que presenta problemas en entornos donde los datos se van modificando constantemente, sin embargo, esto es poco común en páginas web, es intensivo en lectura de datos por lo que es ideal para aplicaciones de monitoreo. Su arquitectura de replicación facilita la escalabilidad de las aplicaciones, la replicación es asíncrona y no hace falta que las réplicas permanezcan conectadas a la fuente para recibir actualizaciones, básicamente se trata de copiar información de una base de datos a otra lo que permite mejorar el rendimiento de una aplicación al distribuir la carga entre varias bases de datos y al mismo tiempo incrementa la velocidad de lectura y escritura de datos.

Figura 13

Replicación asíncrona MySQL



Nota. Las réplicas realizadas entre una o más bases de datos se guardan en un archivo especial denominado registro binario, el registro se crea incluso antes del hilo I/O que conecta la réplica con la fuente. Tomado de (Darnell, 2019).

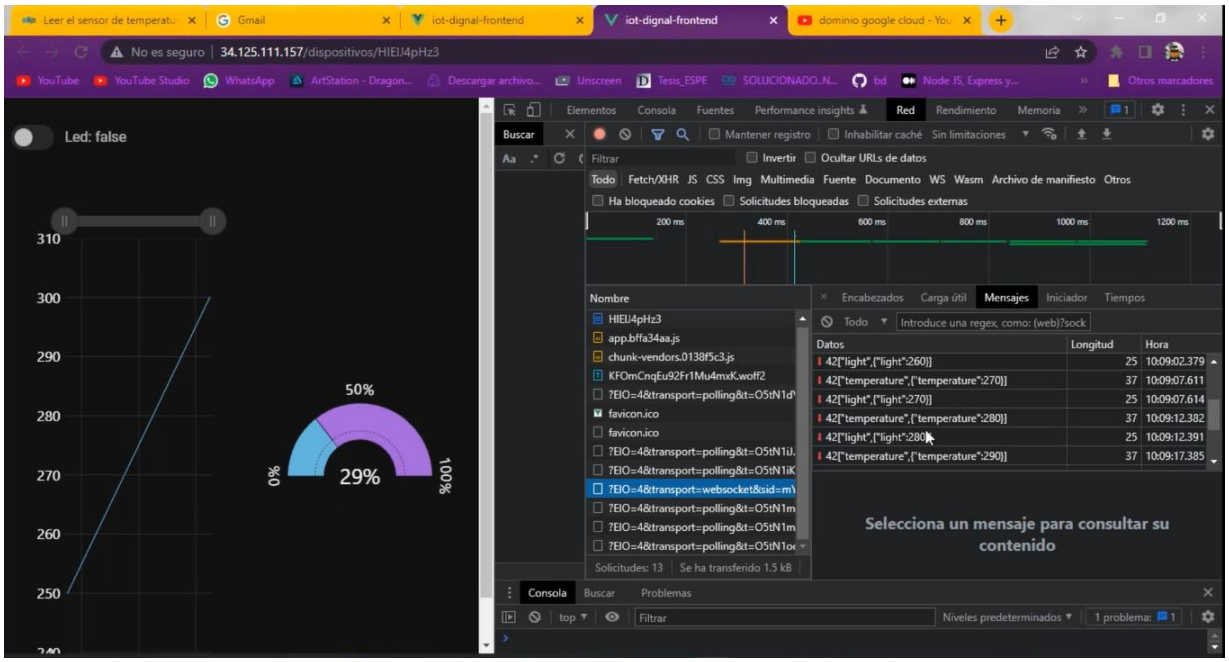
VUE.JS

Es un framework de código abierto, conocido como single-File Component para crear páginas web que junta la estructura de HTML, los estilos de CSS y la lógica de programación de JavaScript en un solo fichero. En Vue.js se habla de componentes, básicamente son archivos con código que permiten crear interfaces de usuario, se usa del lado del frontend. Existen otras

alternativas de framework con topologías similares, pero Vue.js destaca por su corta curva de aprendizaje y su alto desacoplamiento, mientras un proyecto crece se van instalando más librerías esto no afecta en el desempeño de Vue.js lo que es ideal al momento de extender funcionalidades, agregar módulos o utilizar herramientas similares como CSS ya que no existe ningún tipo de restricción al combinarlas, esto también facilita migrar y adaptar proyectos realizados con otros frameworks y pasarlos a Vue.js o simplemente incorporar características que se necesiten usando Vue.js, cada componente se puede importar y reutilizar. También es posible conectar el servidor a una base de datos para tener datos dinámicos lo que es ideal para presentar lecturas de sensores en tiempo real.

Figura 14

Vue.js devtools



Nota. Vue.js devtools es una extensión para el navegador Chrome que permite depurar código, representar el árbol de componentes, registrar eventos, inspeccionar componente y explorar eventos, facilita encontrar errores de programación en la etapa de desarrollo de la aplicación.

Parámetro Biométricos de interés

Peso

Es un indicador de la masa corporal, se usa para valorar el estado nutricional de una persona relacionándolo con otros parámetros como su edad, sexo y talla, se debe tener en cuenta el porcentaje de grasa y masa muscular. Se usa para la valoración antropométrica del paciente, monitorizar la variación en el peso del paciente, identificar efectos secundarios del tratamiento que sigue el paciente y es parte de la toma rutinaria de signos vitales.

Índice de masa corporal (ICM):

$$ICM = \frac{Peso\ [Kg]}{Altura^2\ [m^2]}$$

Temperatura

La temperatura corporal normal puede variar en el rango de 36.5 °C hasta 37.5 °C, es la capacidad del organismo de generar y eliminar calor para mantener su temperatura dentro de los límites seguros. Es un dato personal relativo al estado de salud de un individuo y se puede usar para verificar el estado de salud de una persona, monitorizar su termorregulación y controlar la evolución de una patología. Puede variar dependiendo del lugar del cuerpo en el que se tome la lectura, la edad, la hora del día y es muy sensible a los niveles hormonales. Normalmente es una reacción a infecciones, medicamentos, traumas graves o lesiones e incluso otras afecciones médicas como artritis, hipertiroidismo leucemia, etc.

Tabla 1

Rangos normales de temperatura corporal para adultos y niños

Tipo de lectura	0 a 2 años	3 a 10 años	11 a 65 años	Más de 65 años
Oral	35.5°C – 37.5°C	35.5°C – 37.5°C	36.4°C – 37.6°C	35.8°C – 36.9°C
Rectal	36.6°C – 38°C	36.6°C – 38°C	37°C – 38.1°C	36.2°C – 37.3°C
Axilas	34.7°C – 37.3°C	35.9°C – 36.7°C	35.2°C – 36.9°C	35.6°C – 36.3°C
Oído	36.4°C – 38°C	36.1°C – 37.8°C	35.9°C – 37.6°C	35.8°C – 37.5°C
Frente	36.4°C – 38°C	36.1°C –37.8°C	35.9°C – 37.6°C	35.8°C – 37.5°C

Presión arterial

La presión arterial es la fuerza que se aplica contra las paredes de las arterias cuando el corazón bombea sangre al cuerpo, la toma de presión arterial es el procedimiento mediante el cual se determina la presión máxima o sistólica y la presión mínima o diastólica. Siempre se registra primero la presión sistólica y luego la diastólica. La primera vez que una persona se toma la presión arterial debe hacerlo en ambos brazos para posteriormente usar el brazo que arroje la lectura más alta. Las fuentes de error más comunes suelen ser posición inapropiada de la extremidad, deflación inapropiada, observación de la primera presión arterial, brechas auscultatorias, factores del paciente, complicaciones como lesiones petequiales de la piel o el uso de un brazalete de tamaño equivocado.

Tabla 2

Valores de presión arterial

Categoría	Sistólica (mmHg)	Diastólica (mmHg)
Normal	Menos de 120	Menos de 80
Elevada	120 – 129	80 – 84
Alta grado 1	130 – 139	85 – 90
Alta grado 2	140 o más	90 o más
Alta grado 3	Superior a 180	Superior a 120

Frecuencia Cardiaca

Se define como el número de veces que el corazón se contrae en un minuto, para el correcto funcionamiento del organismo el corazón debe bombear sangre a una determinada presión y frecuencia, se puede sentir en lugares del cuerpo en los que haya una arteria cerca de la piel como en la muñeca o cuello, entre menor frecuencia cardíaca menor es la cantidad de sangre que el corazón bombea al cuerpo. Se usa para identificar anormalidades en el ritmo cardíaco, monitorear la evolución de patologías específicas de un paciente, identificar efectos secundarios de algunos medicamentos y forma parte de la toma rutinaria de signos vitales.

Tabla 3

Valores Normales de frecuencia cardíaca

Grupo	Edad	Latidos por minuto
Recién nacido	0 – 6 semanas	120 – 140
Infante	7 semanas – 1 año	100 – 130
Lactante Mayor	1 – 2 años	100 – 120
Pre-escolar	2 – 6 años	80 – 120
Escolar	6 – 13 años	80 – 100
Adolescente	13 – 16 años	70 – 80
Adulto	16 años en adelante	60 – 80

Saturación de oxígeno en la sangre

Se suele abreviar como SpO2 e indica la concentración de oxígeno en la sangre, los pulmones envían el oxígeno que inhalan al torrente sanguíneo a través del cual se dirige al corazón, cuando el corazón bombea sangre el oxígeno se une a los glóbulos rojos y se reparte por todo el cuerpo. Lo normal es que una persona saludable marque valores de SpO2 sobre el 95%, mientras que las personas que fueron diagnosticadas con COVID-19 deben marcar por lo menos un 92%, de otro modo se le deberá administrar oxígeno suplementario. Cuando se encuentra por debajo del 90% se produce una deficiencia de oxígeno en la sangre, células y tejidos que se denomina hipoxemia, provoca disminución del rendimiento cerebral y pérdidas de memoria, las personas con hipoxemia suelen experimentan dificultad para respirar, dolores de cabeza, confusión, agitación o mareos. Si se va a usar un oxímetro para medir el SpO2 es importante evitar el uso de esmalte de uñas y evitar estar agitado físicamente, para personas diagnosticadas con COVID-19 se sugiere se realicen una prueba cada 8 horas.

Glucosa en sangre

La glucosa es producida por el hígado, aunque también puede provenir de los azúcares que se ingieren con los alimentos y que al ser metabolizados se convierten en glucosa que se transporta por medio del torrente sanguíneo a todas las células del cuerpo, estas últimas usan

la glucosa como principal fuente de energía, el cuerpo regula la cantidad de glucosa en sangre por medio de la producción de insulina, cuando se incrementa la cantidad de insulina que el páncreas produce la glucosa en sangre disminuye. Al bajo nivel de glucosa en sangre se le llama Hipoglucemia, un nivel de glucosa más alto de lo normal se le llama Hiperglucemia. Las personas con diabetes deben mantener la glucosa en un nivel determinado por lo que suelen hacerse mediciones varias veces al día. Una de las pruebas más comunes para medir la glucosa es por medio de una prueba de sangre, se coloca una gota de sangre en una tirilla reactiva que por medio de una reacción electroquímica permite determinar la cantidad de glucosa en sangre.

Tabla 4

Niveles de glucosa en sangre por edades

Edad	En ayunas, al despertar o después del ayuno nocturno	Hasta dos horas luego de consumir alimentos	Antes de acostarse o 4 horas después de comer
0 – 5 años	100 – 180 mg/dl	Máximo 200 mg/dl	100 – 200 mg/dl
6 – 12 años	90 – 180 mg/dl	Máximo 200 mg/dl	100 – 180 mg/dl
13 – 19 años	90 – 130 mg/dl	Máximo 180 mg/dl	90 – 150 mg/dl
19 – 35 años	90 – 130 mg/dl	Máximo 140 mg/dl	90 – 150 mg/dl
35 – 60 años	90 – 100 mg/dl	Máximo 150 mg/dl	100 – 140 mg/dl
> 60 años	80 – 110 mg/dl	Máximo 160 mg/dl	100 – 140 mg/dl

Capítulo III

Diseño y Construcción

Matriz QFD

Es una herramienta que permite recoger la voz del cliente durante la etapa de diseño de un producto. El presente proyecto busca solventar las necesidades de una zona rural que consta de un centro de salud relativamente pequeño y se enfoca en los pacientes que pertenecen al grupo de atención prioritaria por lo que interrelacionar sus necesidades con las características técnicas del sistema permitirá aprovechar los limitados recursos del sector, el personal que trabaja en el centro de salud y de los mismos pacientes. En cuanto al nivel de importancia se clasifica entre 1 y 5, en donde 1 es poco importante y 5 muy importante.

Tabla 5

Lista de requerimientos de los clientes

No	Necesidad	Importancia
1	**Dispositivos recargables**	**5**
2	Indicador sonoro	3
3	Fuente de alimentación	5
4	Gráficos intuitivos	3
5	Dispositivos fáciles de colocar	4
6	Fácil de transportar	4
7	Chatbot amigable y cortés	1
8	Actualizar datos de los usuarios	2
9	Lecturas confiables	5
10	Pocos cables	3

Nota. Se han tomado en cuenta los requerimientos de los pacientes en torno al uso de los dispositivos para monitoreo diario y del personal del centro de salud en torno a la administración y visualización de datos en periodos de tiempo más significativos.

Tabla 6

Especificaciones técnicas del sistema

No	Especificaciones técnicas
1	Cada dispositivo cuenta con una batería de litio recargable
2	Cada dispositivo cuenta con un buzzer
3	Un conmutador deslizante permite desacoplar la batería para que el circuito se alimente a través de una fuente externa
4	Cada dispositivos cuenta con un registro de datos y una gráfica con el último dato adquirido
5	El modelo 3D y algoritmo de los dispositivos respeta los protocolos médicos de adquisición de cada parámetro biométrico
6	Todos los dispositivos usan el mismo tipo de fuente de carga y alimentación
7	Capacidad de comprensión de lenguaje natural
8	Permisos de administrador para modificar información
9	Sensores modernos bien calibrados
10	Uso de software compatible y de código abierto
11	Transferencia de información a través de Wi-Fi

Nota. Las especificaciones técnicas se tomaron a partir de las características generales de los sensores, a parte de las especificaciones enumeradas cada sensor tiene características propias que garantizar una correcta adquisición del parámetro biométrico destinado.

Tabla 7

Relación entre los requerimientos y las especificaciones técnicas

Tipos de relación necesidades cliente y requerimientos técnicos	Simbología	Calificación
Relación débil	O	1
Relación media	Δ	5
Relación fuerte	▉	9

Nota. También conocida como matriz de relaciones, permite saber que especificaciones técnicas afectan a las necesidades de los clientes y en qué medida.

Tabla 8

Matriz de correlación

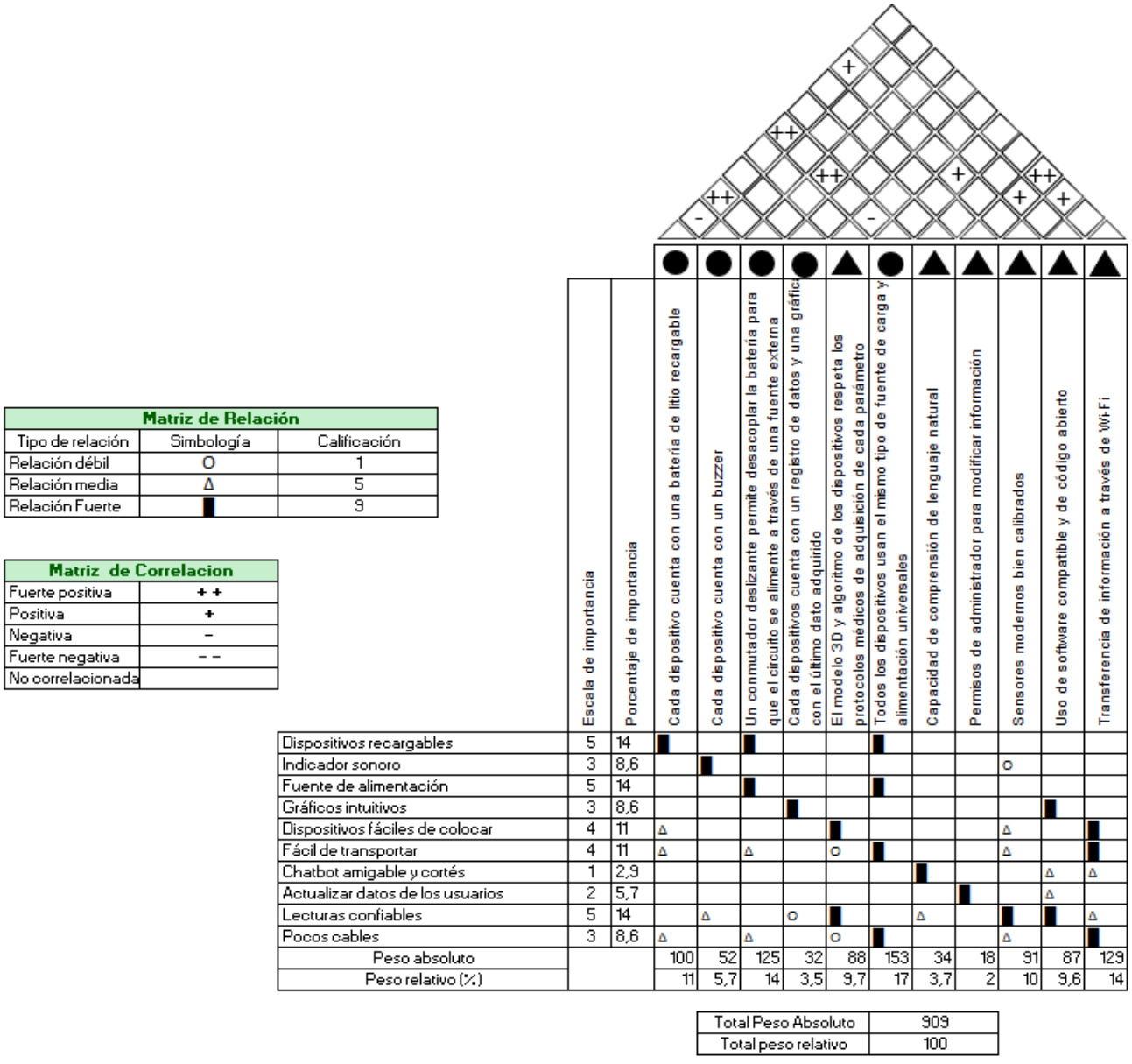
Matriz de correlación	
Fuerte positiva	++
Positiva	+
Negativa	-
Fuerte Negativa	--
No correlacionada	

Nota. Indica si las especificaciones de diseño se ayudan u obstaculizan entre sí.

Sobre cada especificación de diseño se colocan flechas que indican si la cantidad de dicho parámetro debe crecer o decrecer a fin de satisfacer la necesidad asignada, por ejemplo, si se tratara de peso y la flecha se coloca apuntando hacia abajo significa que se requiere reducir el peso en la medida de lo posible, si se requiere que el objeto sea pesado la flecha apuntará hacia arriba mostrando que a más peso el diseño es mejor o un punto si se requiere de un valor específico basado en alguna norma o algo similar.

Figura 15

Matriz QFD



Matriz de Relación		
Tipo de relación	Simbología	Calificación
Relación débil	O	1
Relación media	Δ	5
Relación Fuerte	▮	9

Matriz de Correlacion	
Fuerte positiva	+ +
Positiva	+
Negativa	–
Fuerte negativa	– –
No correlacionada	

	Escala de importancia	Porcentaje de importancia	Cada dispositivo cuenta con una batería de litio recargable	Cada dispositivo cuenta con un buzzer	Un conmutador deslizante permite desacoplar la batería para que el circuito se alimente a través de una fuente externa	Cada dispositivos cuenta con un registro de datos y una gráfica con el último dato adquirido	El modelo 3D y algoritmo de los dispositivos respeta los protocolos médicos de adquisición de cada parámetro	Todos los dispositivos usan el mismo tipo de fuente de carga y alimentación universales	Capacidad de comprensión de lenguaje natural	Permisos de administrador para modificar información	Sensores modernos bien calibrados	Uso de software compatible y de código abierto	Transferencia de información a través de Wi-Fi
Dispositivos recargables	5	14	▮		▮			▮					
Indicador sonoro	3	8,6		▮							O		
Fuente de alimentación	5	14			▮			▮					
Gráficos intuitivos	3	8,6				▮						▮	
Dispositivos fáciles de colocar	4	11	Δ				▮				Δ		▮
Fácil de transportar	4	11	Δ		Δ		O	▮			Δ		▮
Chatbot amigable y cortés	1	2,9							▮			Δ	Δ
Actualizar datos de los usuarios	2	5,7								▮		Δ	
Lecturas confiables	5	14		Δ		O	▮		Δ		▮	▮	Δ
Pocos cables	3	8,6	Δ		Δ		O	▮			Δ		▮
Peso absoluto			100	52	125	32	88	153	34	18	91	87	129
Peso relativo (%)			11	5,7	14	3,5	9,7	17	3,7	2	10	9,6	14

Total Peso Absoluto	909
Total peso relativo	100

Nota. Elaborar la matriz QFD es un trabajo que lleva una cantidad considerable de tiempo, pero es indispensable en proyectos cuyo objetivo principal es la satisfacción del cliente, en este caso los pacientes y personal del centro de salud.

Tabla 9

Resultado matriz QFD

Característica técnica	Importancia
Todos los dispositivos usan el mismo tipo de fuente de carga y alimentación	1
Transferencia de información a través de Wi-Fi	2
Un conmutador deslizante permite desacoplar la batería para que el circuito se alimente a través de una fuente externa	3
Cada dispositivo cuenta con una batería de litio recargable	4
Sensores modernos bien calibrados	5
El modelo 3D y algoritmo de los dispositivos respeta los protocolos médicos de adquisición de cada parámetro biométrico	6
Uso de software compatible y de código abierto	7
Cada dispositivo cuenta con un buzzer	8
Capacidad de comprensión de lenguaje natural	9
Cada dispositivos cuenta con un registro de datos y una gráfica con el último dato adquirido	10
Permisos de administrador para modificar información	11

Nota. Las características técnicas se ordenan de 1 que es la más importante hasta 11 para la menos importante.

Análisis de la matriz QFD

De las 4 especificaciones con más importancia 3 hacen referencia a la alimentación de los dispositivos:

- Todos los dispositivos usan el mismo tipo de fuente de carga y alimentación
- Transferencia de información a través de Wi-Fi
- Un conmutador deslizante permite desacoplar la batería para que el circuito se alimente a través de una fuente externa
- Cada dispositivo cuenta con una batería de litio recargable

Esto se debe a que se necesita dispositivos recargables para toma de parámetros biométricos fuera del centro de salud por ejemplo pacientes que se les dificulta abandonar su

hogar a causa de una discapacidad y a su vez un cargador de pared que permita usar los dispositivos por tiempo prolongado por ejemplo en caso de una campaña de vacunación o donación de sangre en colegios o universidades.

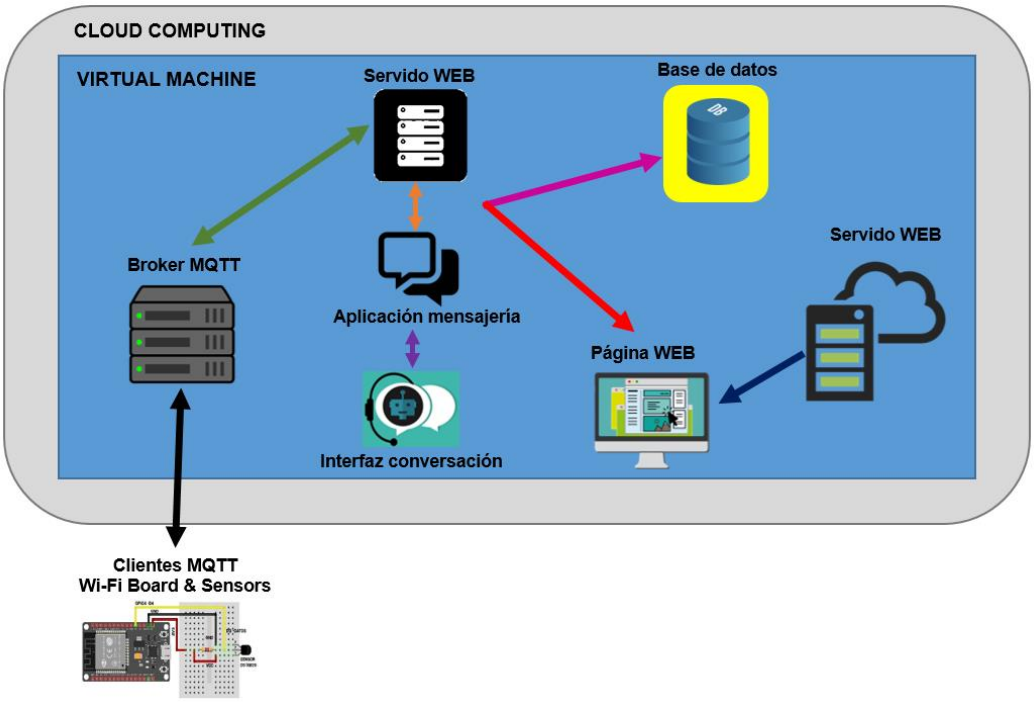
La transferencia de datos por Wi-Fi también tiene mucho peso debido a que se relaciona con varias de las necesidades de los usuarios como son la facilidad de transportar y usar los dispositivos. También se relaciona con la necesidad de pocos cables característica típica de los dispositivos de RPM.

Por lo tanto, dichas especificaciones deben ser cuidadosamente diseñadas e integradas para cumplir con los requerimientos de los usuarios.

Diseño general del prototipo

Figura 16

Diseño del prototipo sin especificar Software ni Hardware



Nota. Todo el software dentro del rectángulo azul se instalará en una máquina virtual que a su vez se montará sobre una plataforma de Cloud Computing representado por el rectángulo redondeado de color plomo.

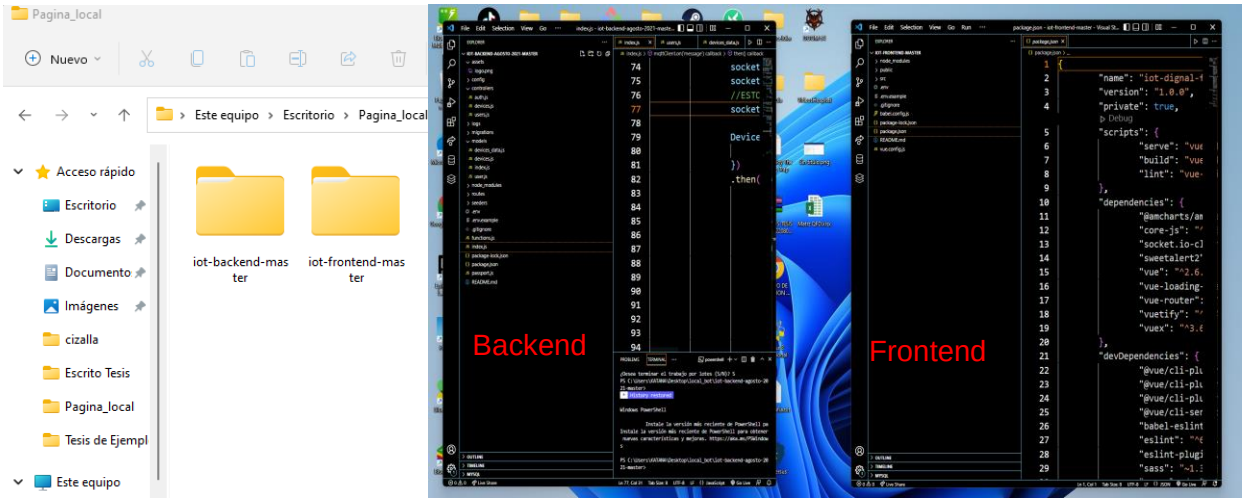
Diseño local de la plataforma web

La plataforma web se elabora primero a nivel local en cuanto al ingreso a la página web, la creación de usuarios, la creación y administración de dispositivos, la visualización de gráficas, el Chatbot, etc. Luego todo el software se migrará a un servicio de cloud computing para la integración de los dispositivos y el acceso por medio de internet.

Siguiendo la tendencia actual de elaboración de páginas web se divide en backend y frontend, Visual Studio Code los maneja como proyectos distintos, el backend tiene toda la información sensible, a la que no se requiere que el usuario tenga acceso directo, se encarga de gestionar el almacenamiento de datos y contiene todo lo que se encuentra del lado del servidor, el frontend toma los datos relevantes del backend y se los muestra a los usuarios, es la parte que interactúa con el usuario, comúnmente se denomina como el lado del cliente, backend y frontend se comunican a través de una API.

Figura 17

Trabajar backend y frontend como proyectos distintos de VS Code



Nota. Backend y frontend interactúan entre sí para asegurar la funcionalidad de la página web.

Selección de gestor de base de datos

Tabla 10

Comparativa entre los principales gestores de bases de datos

CRITERIOS DE SELECCIÓN	**GESTOR DE BASE DE DATOS**			
	Sybase	ACCESS	Oracle	MySQL
Multiplataforma	0	-	+	+
Velocidad de escritura, lectura y consultas	+	0	0	+
Seguridad	0	0	+	0
Escalabilidad	0	-	+	+
Costo	+	0	-	0
Disponibilidad de documentación	0	0	-	-
Orientado a internet	-	0	+	+
Facilidad de configuración	-	0	-	+
Suma +	2	0	4	5
Suma 0	4	6	1	2
Suma -	2	2	3	1
EVALUACIÓN NETA	0	-2	1	4
LUGAR	3	4	2	1
¿CONTINUAR?	No	No	No	Si

Nota. En cuanto a los lugares la puntuación más alta corresponde al gestor con mejores prestaciones y tiene el lugar 1, al gestor menos indicado le corresponde el lugar 4.

Para aplicaciones de monitoreo remoto el gestor de base de datos MySQL es el más indicado entre otras cosas por su fácil configuración, escalabilidad, precio, porque es multiplataforma, además de otras características especializadas que agregan valor, pero no se han tomado en cuenta al comparar únicamente los parámetros más relevantes.

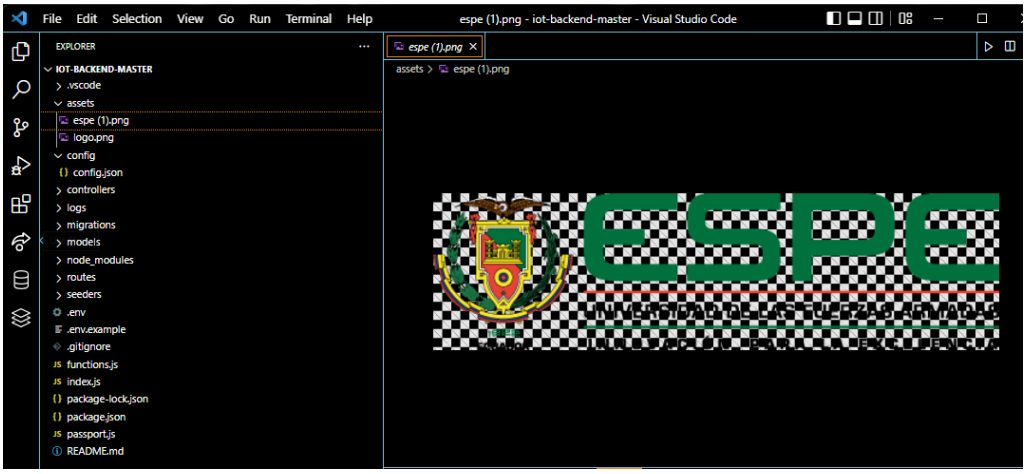
Backend

En la carpeta raíz del backend se encuentran:

Directorio assets. Almacena elementos estáticos. Contiene recursos que se quieren usar en el proyecto como imágenes o videos que se pueden visualizar en el frontend pero se guardan del lado del backend para resguardar dicha información.

Figura 18

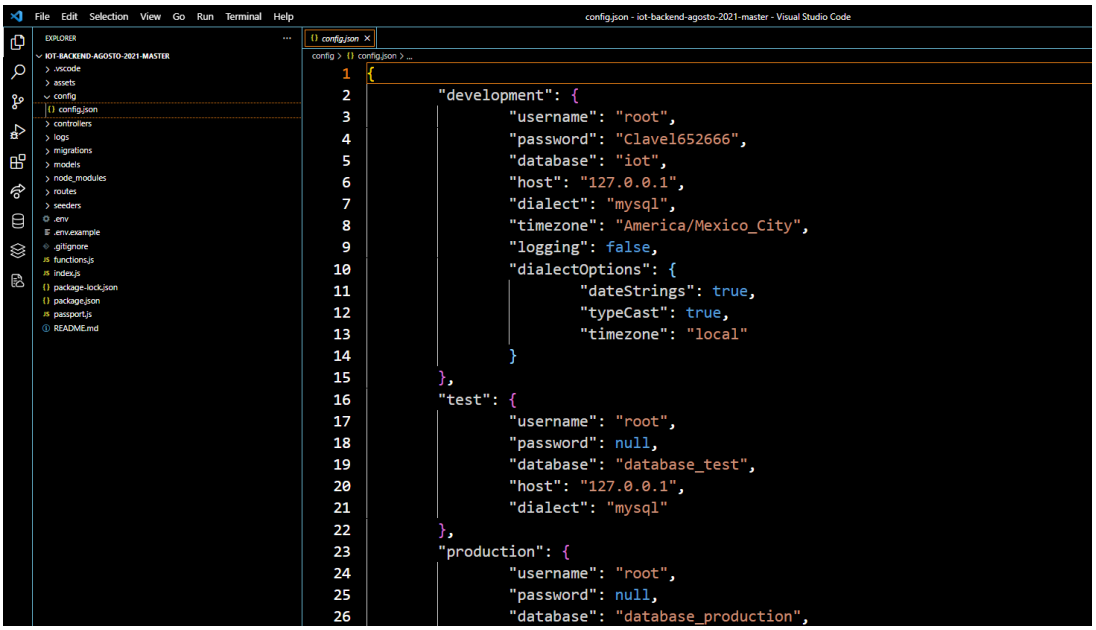
Carpeta assets



Directorio config. Contiene diversos tipos de configuración, por el momento solo contiene la configuración de la base de datos, es el que consulta el ORM para conectarse a la base de datos.

Figura 19

Archivo config.json

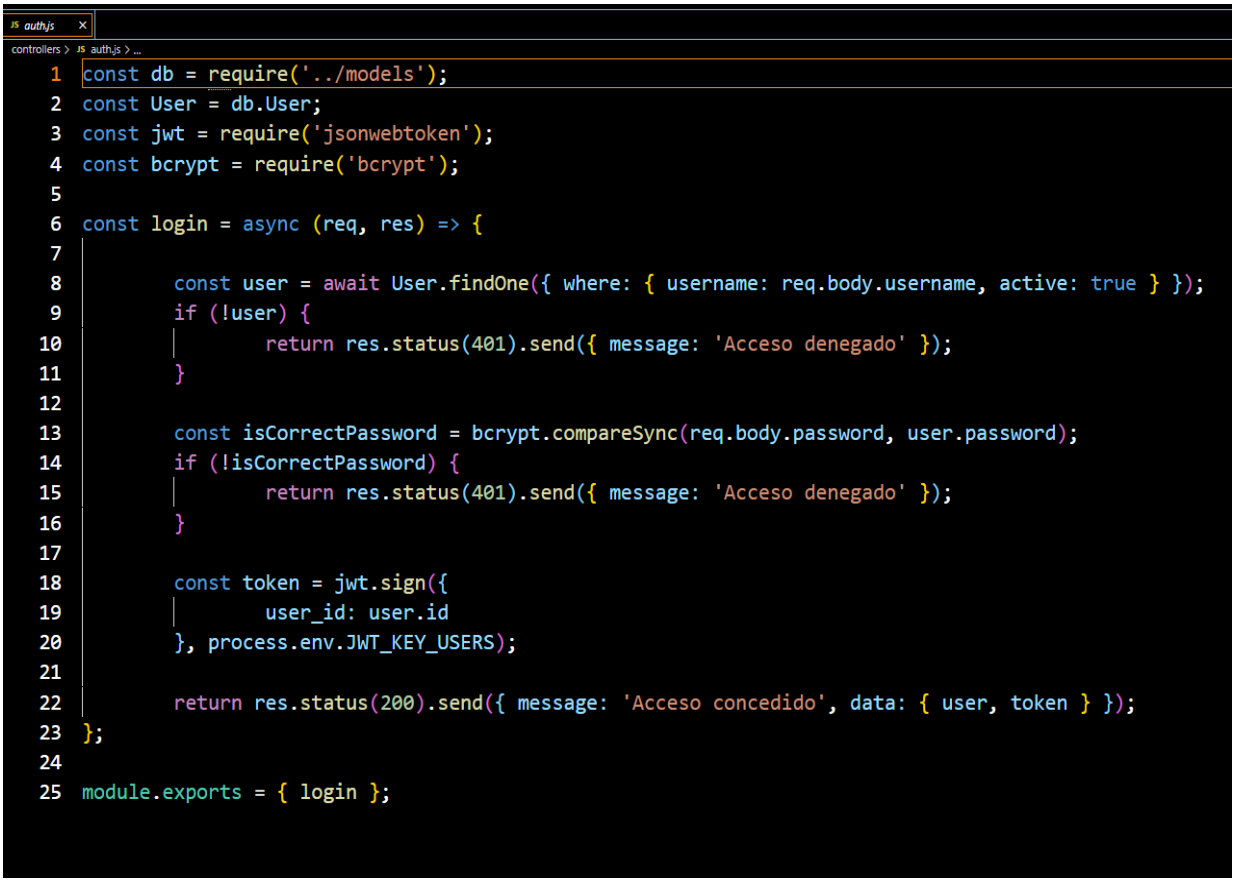


Nota. Contiene parámetros como el usuario, clave, zona horaria, entre otros datos importantes para conectarse a MySQL por medio de Sequelize.

Directorio controllers. Contiene los archivos auth.js, devices.js y users.js los que se encargan de controlar el inicio de sesión de usuarios, los dispositivos y los parámetros de los usuarios respectivamente.

Figura 20

Archivo auth.js

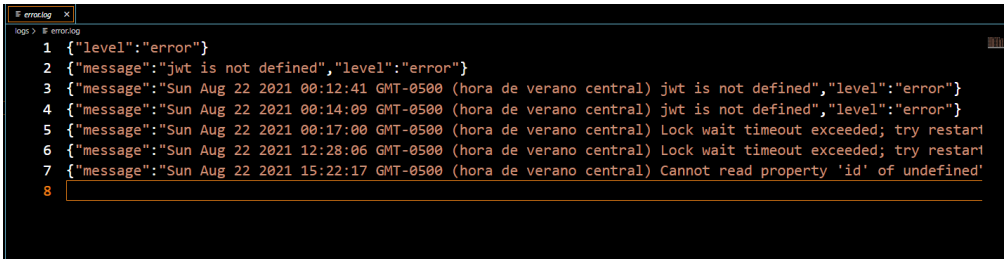


Nota. Proceso de autenticación de usuarios mediante JSON Web Tokens, biblioteca que crea y verifica tokens de seguridad, se utiliza para compartir información entre cliente y servidor.

Directorio logs. Contiene un registro secuencial de los eventos que afectan el funcionamiento de la aplicación a manera de evidencia, en el archivo error.log se registran los errores que se generan al correr la aplicación.

Figura 21

Archivo error.log

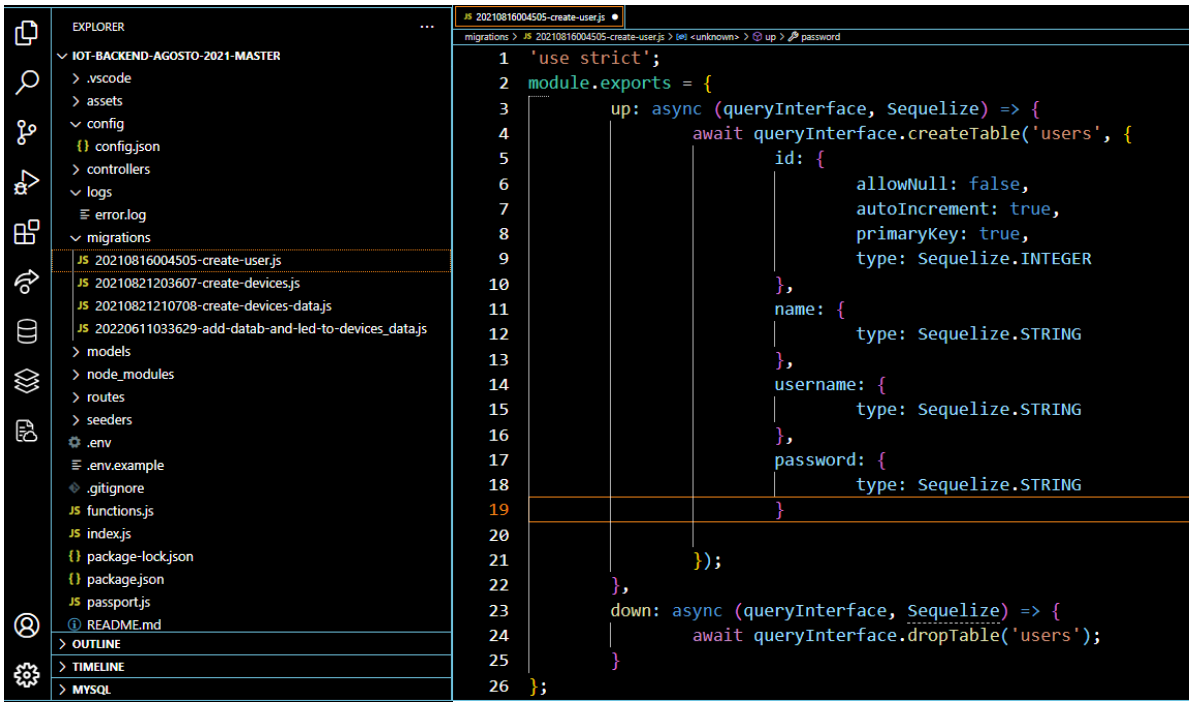


Nota. Se muestran errores de variables no definidas y en la definición de la zona horaria en el archivo de configuración de la base de datos.

Directorio Migrations. Permite generar tablas a nivel de base de datos, anteriormente el archivo config.js contenía los parámetros para que el ORM se conecte a la base de datos, ahora el directorio migrations contiene archivos .js con la información necesaria para crear tablas que permitan posteriormente almacenar datos en ellas.

Figura 22

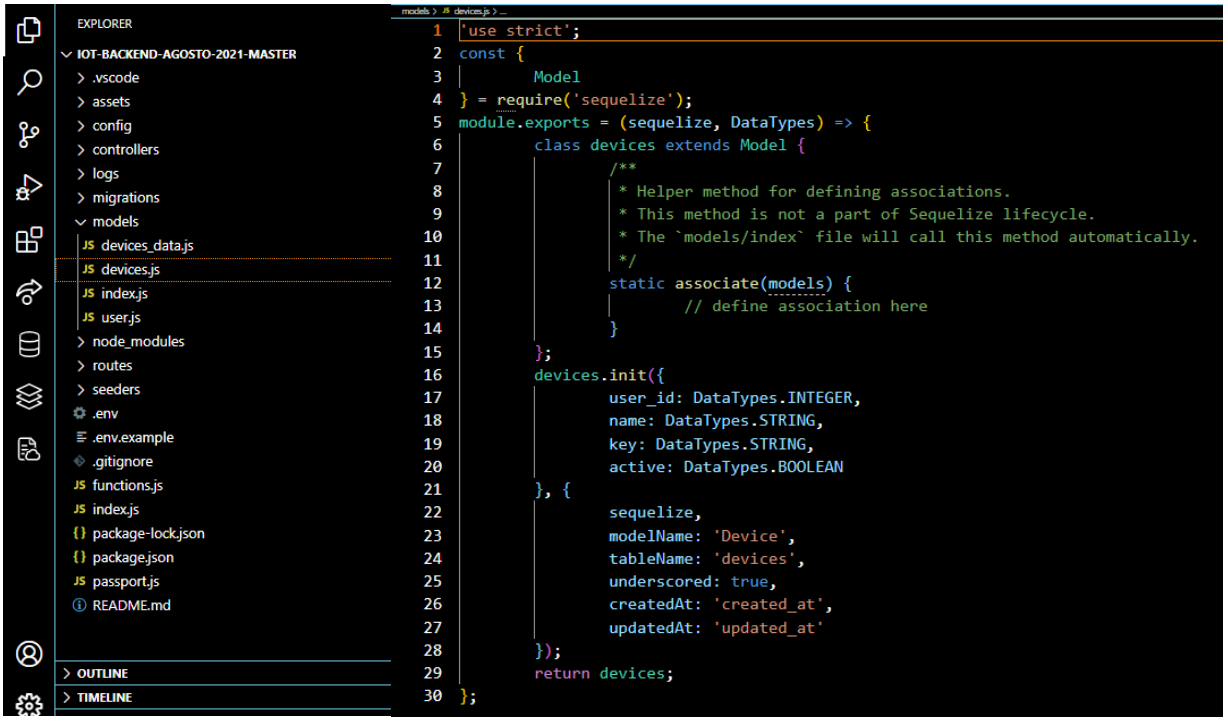
Creación de tabla de usuarios bajo el sistema de migraciones



Directorio models. Contiene archivos que transforman las tablas de la base de datos a un lenguaje de programación orientado a objetos para hacer uso de ellos en consultas, escritura, relación entre tablas, etc.

Figura 23

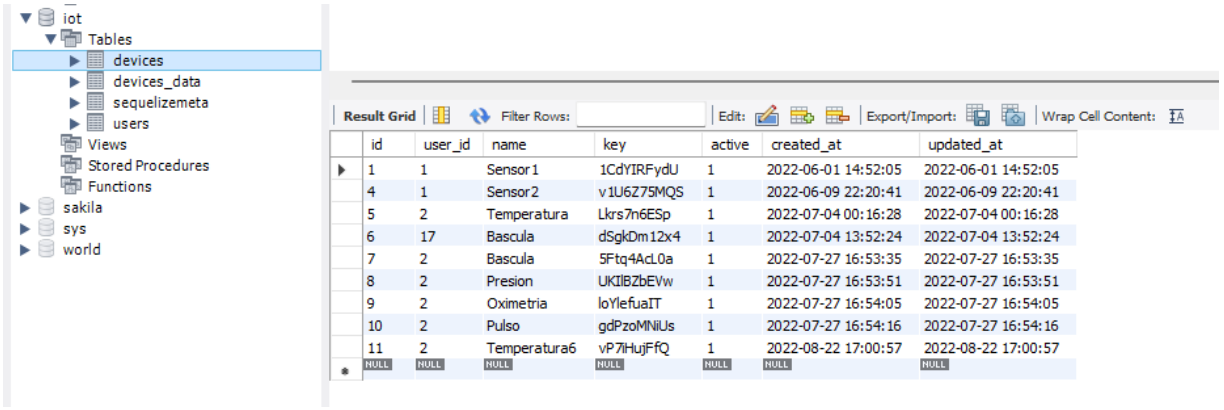
Modelo para la tabla devices



Nota. Un modelo es una abstracción que representa una tabla en la base de datos e incluye información que Sequelize necesita como el nombre de la tabla y sus columnas.

Figura 24

Tabla devices base de datos iot gestor MySQL

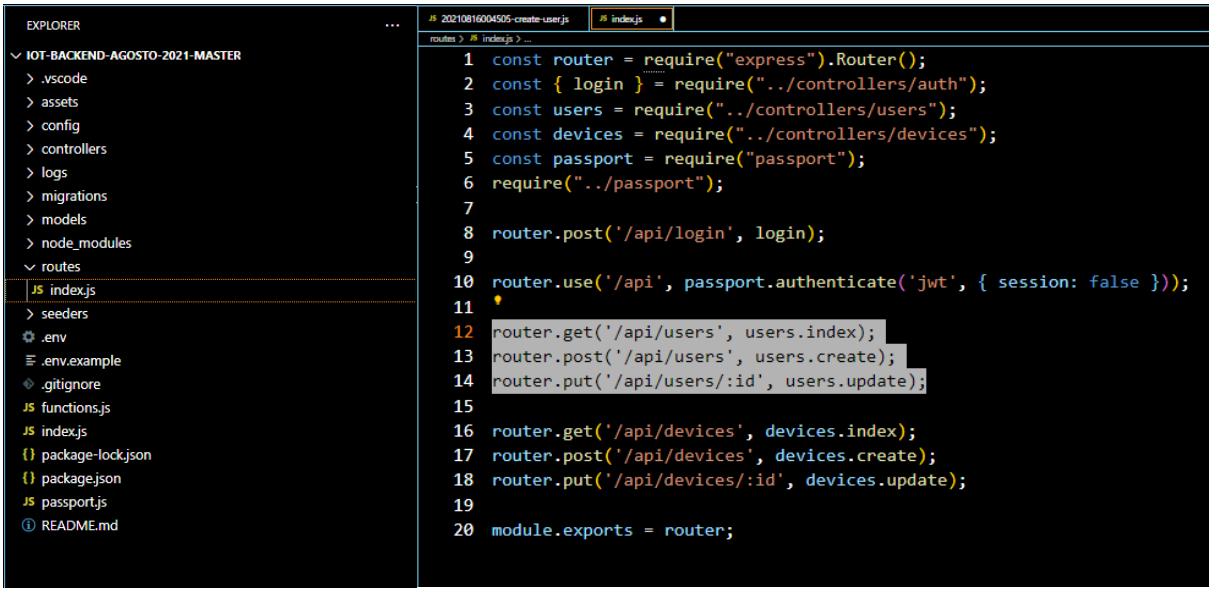


id	user_id	name	key	active	created_at	updated_at
1	1	Sensor1	1CdYIRFydU	1	2022-06-01 14:52:05	2022-06-01 14:52:05
4	1	Sensor2	v1U6Z75MQS	1	2022-06-09 22:20:41	2022-06-09 22:20:41
5	2	Temperatura	Lkrs7n6ESp	1	2022-07-04 00:16:28	2022-07-04 00:16:28
6	17	Bascula	dSgkDm12x4	1	2022-07-04 13:52:24	2022-07-04 13:52:24
7	2	Bascula	5Ftq4AcL0a	1	2022-07-27 16:53:35	2022-07-27 16:53:35
8	2	Presion	UKIlBZbEVw	1	2022-07-27 16:53:51	2022-07-27 16:53:51
9	2	Oximetria	loYlefuaIT	1	2022-07-27 16:54:05	2022-07-27 16:54:05
10	2	Pulso	gdPzoMNiUs	1	2022-07-27 16:54:16	2022-07-27 16:54:16
11	2	Temperatura6	vP7iHujFfQ	1	2022-08-22 17:00:57	2022-08-22 17:00:57
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Directorio Routes. En su interior se encuentra el archivo index.js, este archivo contiene endpoints o URLs, cuando una persona entra a la página web el frontend solicita la información necesaria para interactuar con el usuario al backend mediante una API, las URLs señalan la localización exacta de los archivos, datos o recursos del backend que el frontend requiere.

Figura 25

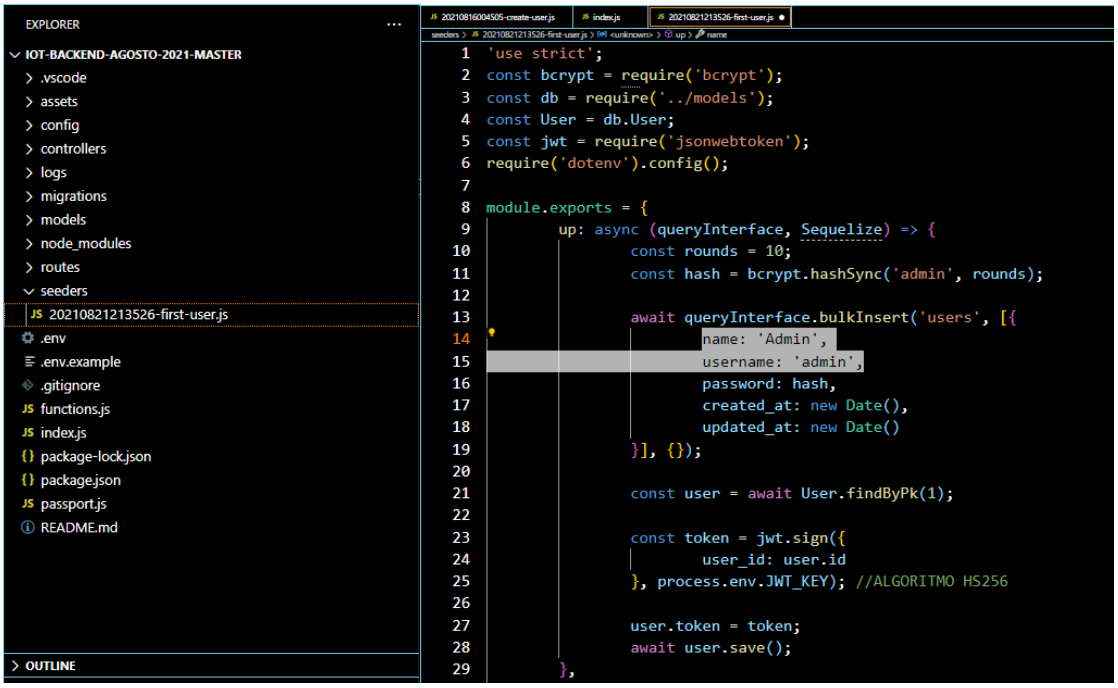
Archivo index.js



Directorio Seeders. Se utiliza para cargar datos de prueba mediante el ORM, contiene un archivo .js que crea un usuario para poder ingresar a la plataforma mediante un nombre de usuario y contraseña demo que será el encargado de administrar la plataforma.

Figura 26

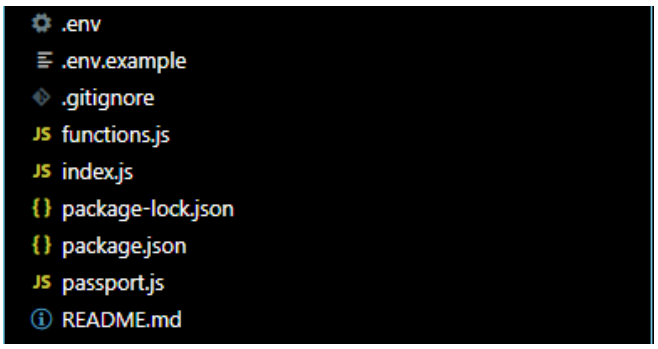
Creación y configuración de usuario Administrador



Además de estas carpetas también se almacenan varios archivos en el directorio raíz que contienen entre otras cosas funciones universales y variables de ambiente. A continuación, se detalla la función de cada uno:

Figura 27

Archivos ubicados en el directorio raíz del backend



- **.env** Contiene variables de ambiente. Permite definir si se está desplegando el proyecto en un entorno de producción o desarrollo.
- **.env.example** Da a entender como hay que configurar el archivo de variables de ambiente, hace referencia al archivo .env.
- **.gitignore** Permite omitir ciertos elementos del repositorio, en este caso omite la carpeta node_modules que es el que almacena todas las librerías insertadas por medio de npm para que no se descarguen junto con el resto del proyecto.
- **Readme.me** Contiene indicaciones para instalar el proyecto como tal.
- **functions** Contiene funciones que se van a usar frecuentemente a lo largo del proyecto y facilita llamarlas por medio del nombre del archivo.
- **index.js** Punto de entrada al proyecto, donde se inicializa la plataforma.
- **package.json** Almacena la versión de las librerías instaladas y algunas configuraciones adicionales. Usa el denominado versionado semántico
- **package_lock.json** Almacena las librerías que se han instalado para recuperarlas cuando alguien descargue el proyecto y quiera correrlo. Evita tener que actualizar versiones, cuando alguien ejecuta el comando npm install permite examinar e instalar la versión exacta de una librería.
- **passport.js** Librería que se incluye para el inicio de sesión a través de jwt o Json Web Token.

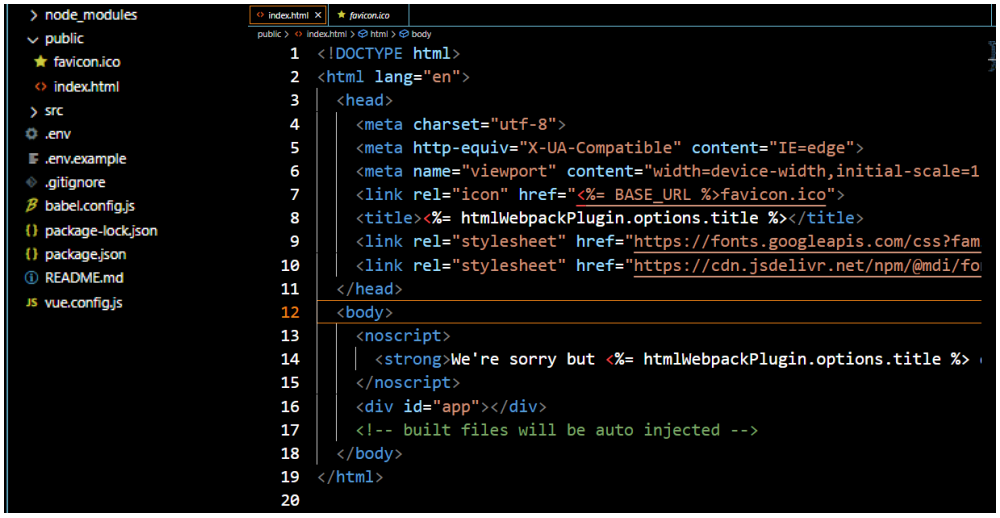
Frontend

En el directorio raíz del frontend se encuentran:

Directorio public. Contiene el icono de la página web y el HTML que va a consumir el navegador para visualizar la página. Los navegadores solo reconocen 3 elementos, HTML a nivel de estructura, CSS a nivel de diseño y JavaScript a nivel de lógica de programación.

Figura 28

Index.html



Directorio src. Se trata de la fuente, contiene el código en un lenguaje de alto nivel, es decir antes de convertirlo a código máquina. Se usa principalmente para editar código fuente.

Figura 29

Núcleo de trabajo frontend

Nota. En proyectos extensos se trabaja por medio de componentes, todos los archivos con extensión .vue que se muestran en la imagen son ejemplos de componentes usados para elaborar la interfaz de usuario.

Además de estos directorios también cuenta con varios archivos que tienen la misma función que para el backend, sin embargo, posee archivos adicionales relacionados con la compatibilidad entre la plataforma y el navegador web:

- **babel.config.js** Es un compilador que interpreta los archivos .vue y lo transforma a código JavaScript que cualquier navegador pueda entender.
- **vue.config.js** Es un archivo de configuración opcional que se carga automáticamente cuando está ubicado en la raíz del proyecto. Permite incluir detalles de configuración adicionales de Vue o sus plugins.

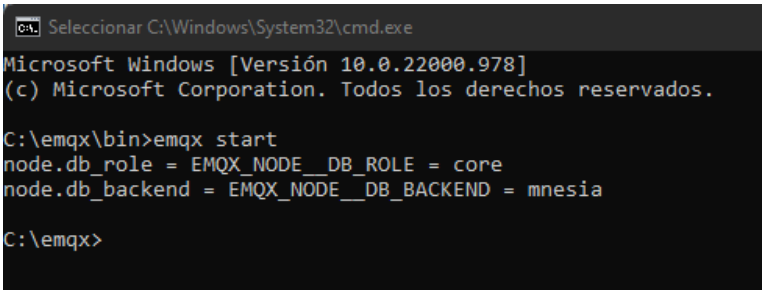
Configuración inicial del backend

Antes de ejecutar el proyecto es necesario instalar Node.js versión 12 o superior, MySQL versión 8 o superior y finalmente el bróker MQTT llamado EMQX versión 5.0.7 o superior.

La instalación de bróker es sumamente sencilla, solo descargar la carpeta y moverla al disco local C. Es necesario iniciar el bróker antes de correr el proyecto.

Figura 30

Iniciar EMQX



Nota. Abrir el command prompt en la siguiente ruta: *C:\emqx\bin* y ejecutar *emqx start.*

Biblioteca MQTT en Node.js. La biblioteca mqtt.js contiene las funciones que permiten conectarse por medio de WebSocket a distintos broker mqtt que funcionan como gestores que permiten enviar y recibir mensajes como por ejemplo mosquitto, aedes-cli, EMQX, etc. Se instala por medio del comando *npm install mqtt*.

Figura 31

Ejemplo de conexión de un cliente al broker Mosquitto con mqtt.js

```
const mqtt = require('mqtt')
const client  = mqtt.connect('mqtt://test.mosquitto.org')

client.on('connect', function () {
  client.subscribe('presence', function (err) {
    if (!err) {
      client.publish('presence', 'Hello mqtt')
    }
  })
})

client.on('message', function (topic, message) {
  // message is Buffer
  console.log(message.toString())
  client.end()
})
```

Nota. La conexión se realiza sobre un servicio de pruebas online de Mosquitto que no es estable y se recomienda que no se use para efectuar pruebas con información sensible. Tomado de (Mosquitto, 2021).

Figura 32

Conexión del backend al broker EMQX con mqtt.js a nivel local

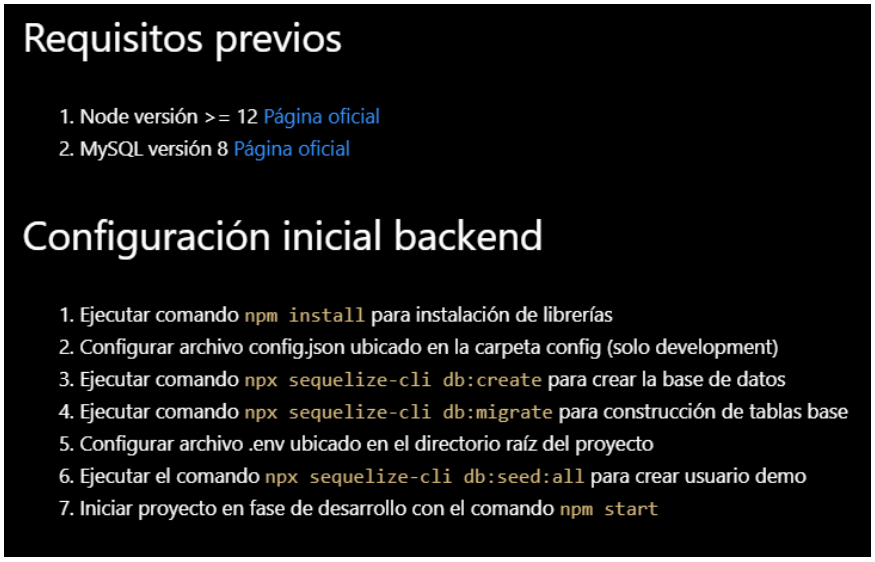
```
const mqttClient=mqtt.connect('mqtt://localhost');
mqttClient.on('error',function (error) {
        console.log(`Error al conectarse al broker: ${error}`);
});

mqttClient.on('connect', function(){
        console.log('Conectado a broker');

//Para recibir la información es necesario suscribirse a un topico
        mqttClient.subscribe('dispositivos/+');

});
```

Nota. Se configura en el archivo index.js ubicado en la raíz del backend, la configuración se realiza para la versión gratuita de EMQX.

Las instrucciones para ejecutar el backend en fase de desarrollo se encuentran en el directorio raíz del backend en el archivo README.md junto con enlaces para la descarga de Node y MySQL para lo cual es necesario activar la previsualización de VS Code.

Figura 33

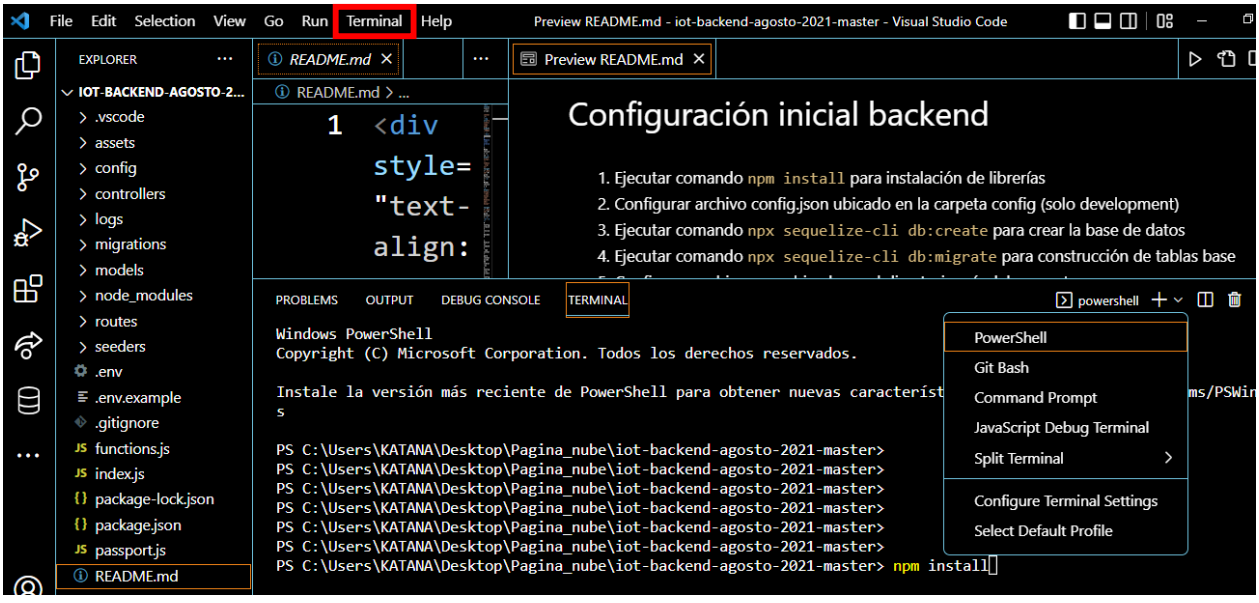
Previsualización archivo READM.md



Nota. Los comandos se deben ejecutar en el terminal que VS Code trae integrado.

Figura 34

Terminal integrado de VS Code

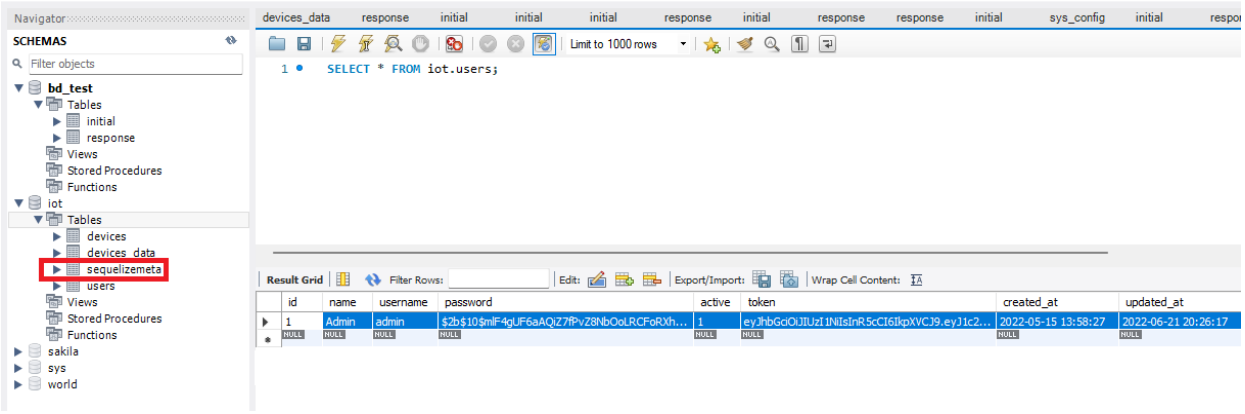


Nota. En la terminal de VS Code es posible seleccionar el intérprete de preferencia, en este caso se utiliza PowerShell.

Antes de iniciar el proyecto, si todos los pasos previos fueron ejecutados adecuadamente al abrir MySQL Workbench se podrán visualizar las tablas y la migración del usuario encargado de administrar la plataforma junto con su contraseña.

Figura 35

Construcción de tablas base

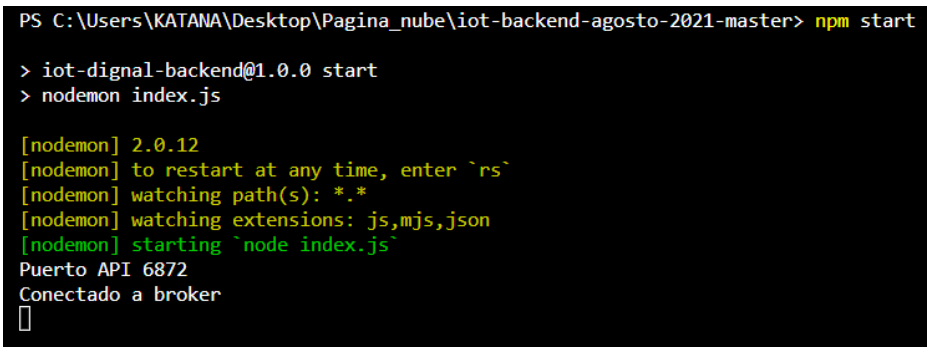


Nota. La tabla sequelizemeta se crea por defecto para llevar el control de las migraciones.

Ahora ya se puede iniciar el proyecto para que cualquier proyecto lo pueda contactar a nivel local. La herramienta nodemon de Node.js es de mucha ayuda en proyectos en fase de desarrollo, al efectuar y guarda algún cambio en cualquier archivo del directorio raíz reinicia automáticamente la aplicación.

Figura 36

Arrancar el Backend



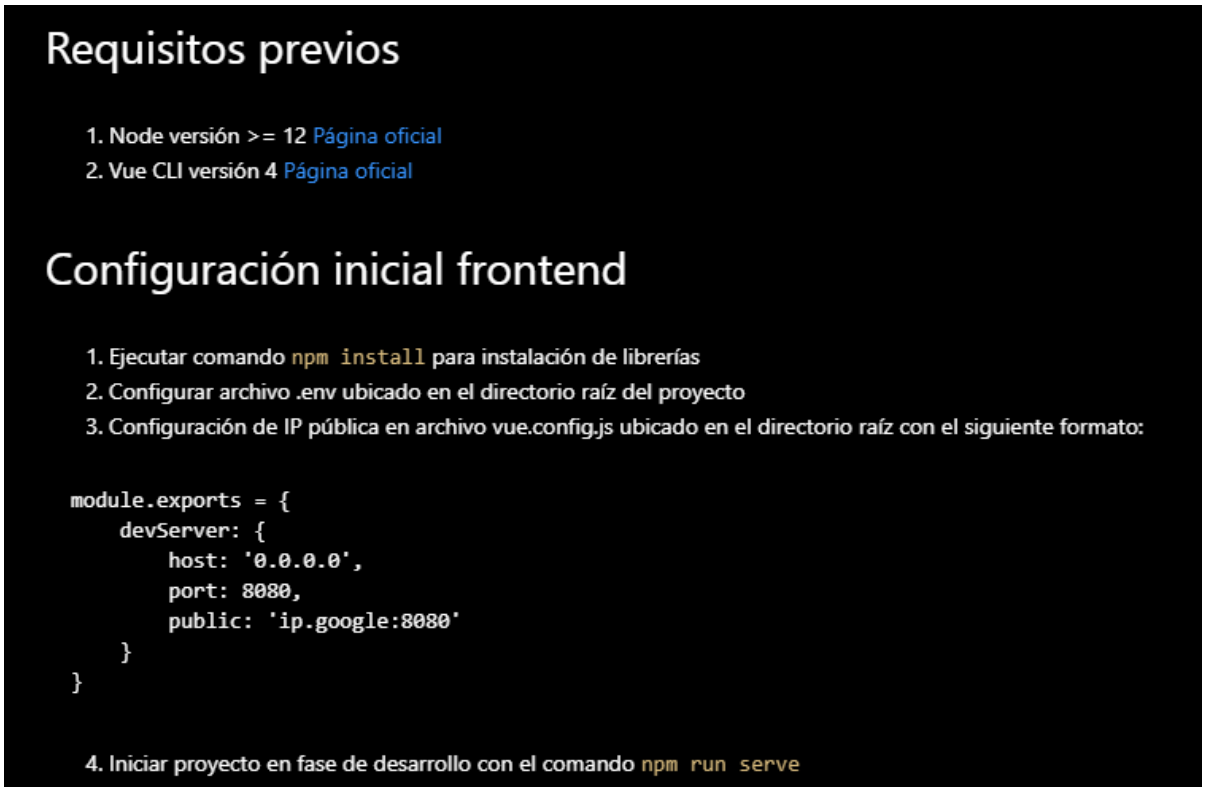
Nota. Se imprime por consola el puerto que se tiene configurado para realizar las conexiones por medio de Websocket y el estado del bróker EMQX.

Configuración inicial del frontend

Se debe dejar corriendo el backend y abrir otra vez VS Code pero en el directorio del frontend que viene siendo un cliente y se ejecutan en el terminal los comandos que especifica el archivo README.md omitiendo el paso 3 que se usa para montar el proyecto en etapa de producción.

Figura 37

Previsualización archivo README.md



Nota. El archivo README.md se encuentra en el directorio raíz del frontend.

Al ejecutar el frontend se presentan en la terminal la dirección local y la dirección IP, con cualquiera de las dos direcciones es posible acceder a la plataforma desde un navegador web.

Figura 38

Dirección local e IP privada del proyecto a nivel local

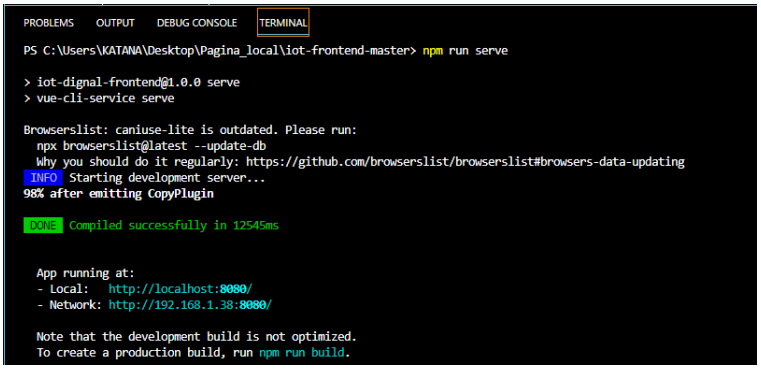
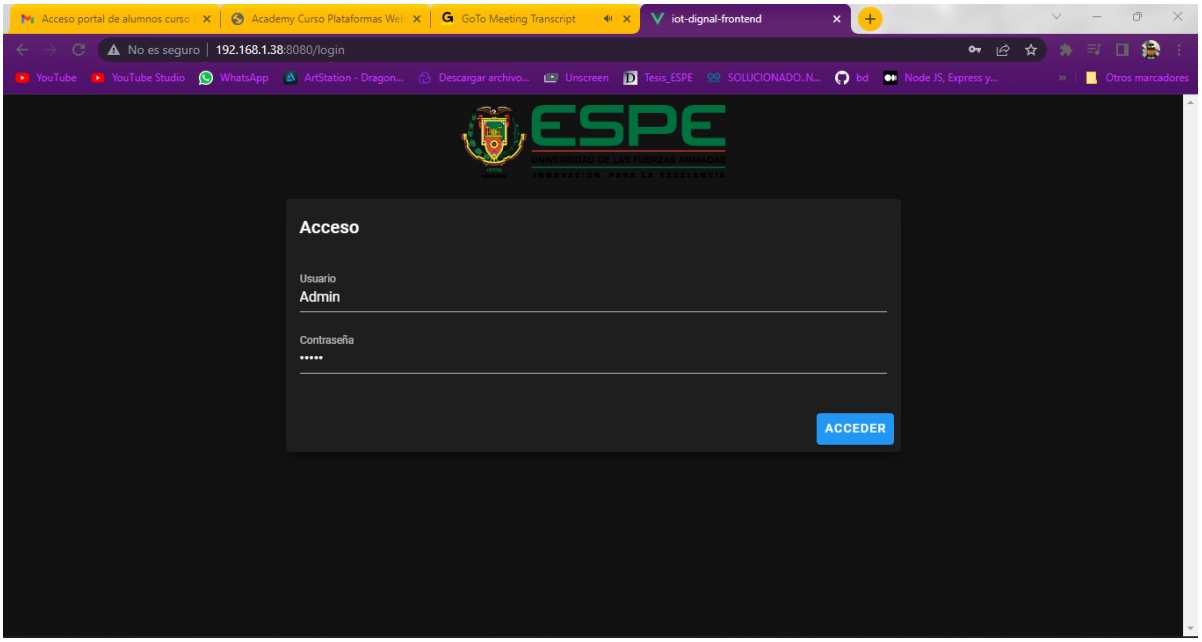


Figura 39

Página de acceso a la plataforma

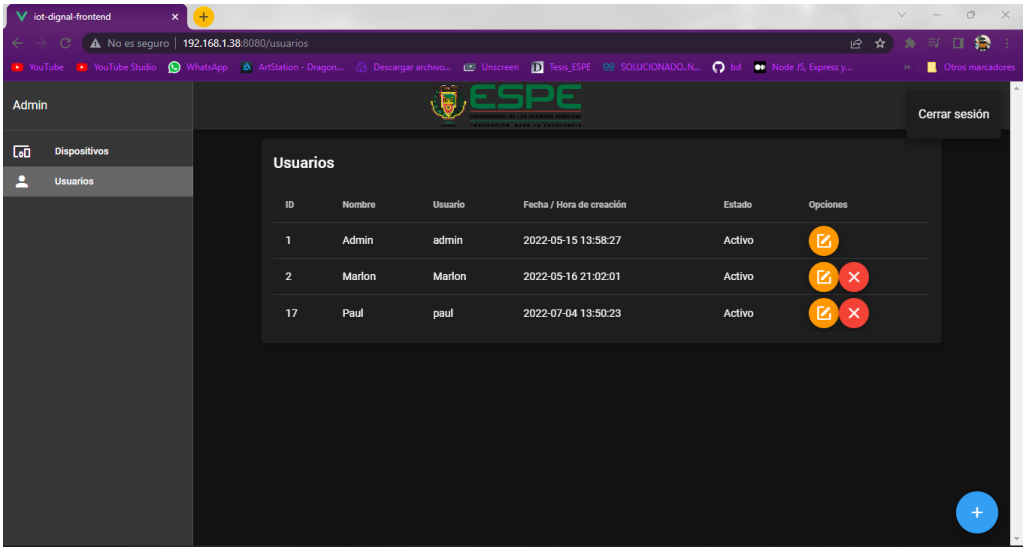


Nota. Como se mencionó con anterioridad por medio de un fichero se carga por defecto un usuario administrador junto con su clave de acceso, para este ejemplo el usuario Admin con su contraseña admin.

Una vez que el administrador accede, la plataforma permite crear nuevos usuarios, para fines específicos de la aplicación además del nombre, usuario y contraseña es necesario ingresar un número de teléfono enlazado a una cuenta de WhatsApp.

Figura 40

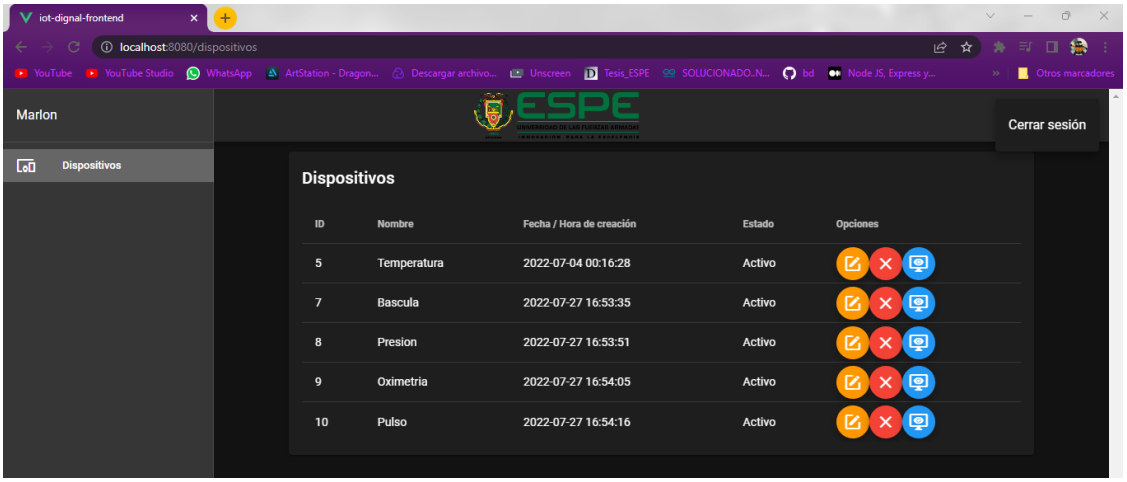
Página de administración de usuarios



Cada usuario puede administrar sus propios dispositivos en base a los parámetros biométricos que se deseen registrar, claro está, con ciertas limitaciones que lo reducen a editar el nombre o el estado de un dispositivo previamente configurado por el administrador.

Figura 41

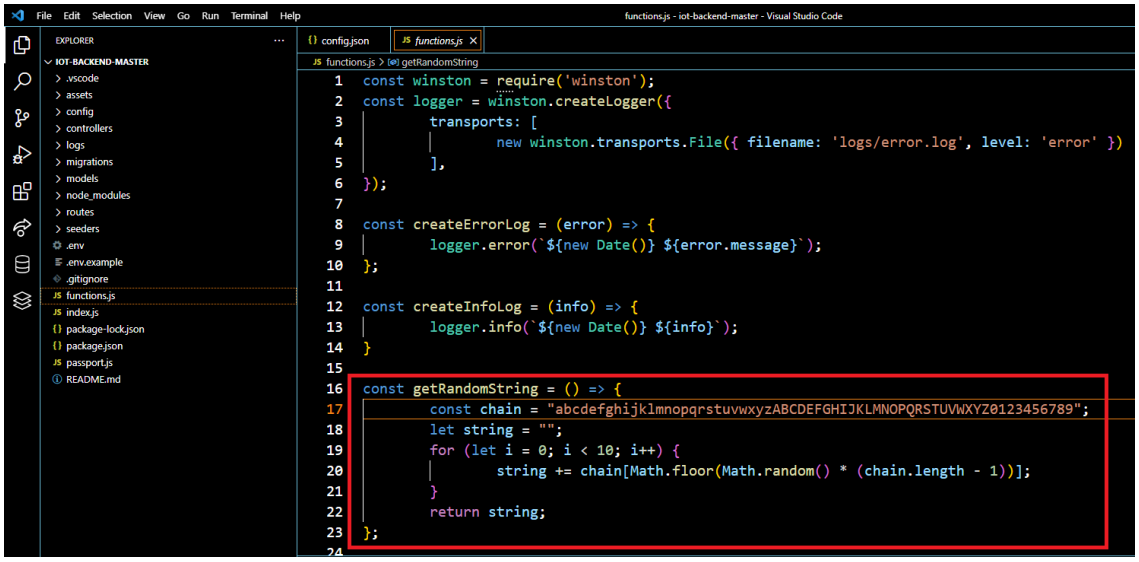
Página de administración de dispositivos



Generación de tópicos para dispositivos. Una función flecha *getRandomStrin*g del directorio functions.js forma un string de 10 caracteres compuestos por números y letras mayúsculas y minúsculas de forma aleatoria.

Figura 42

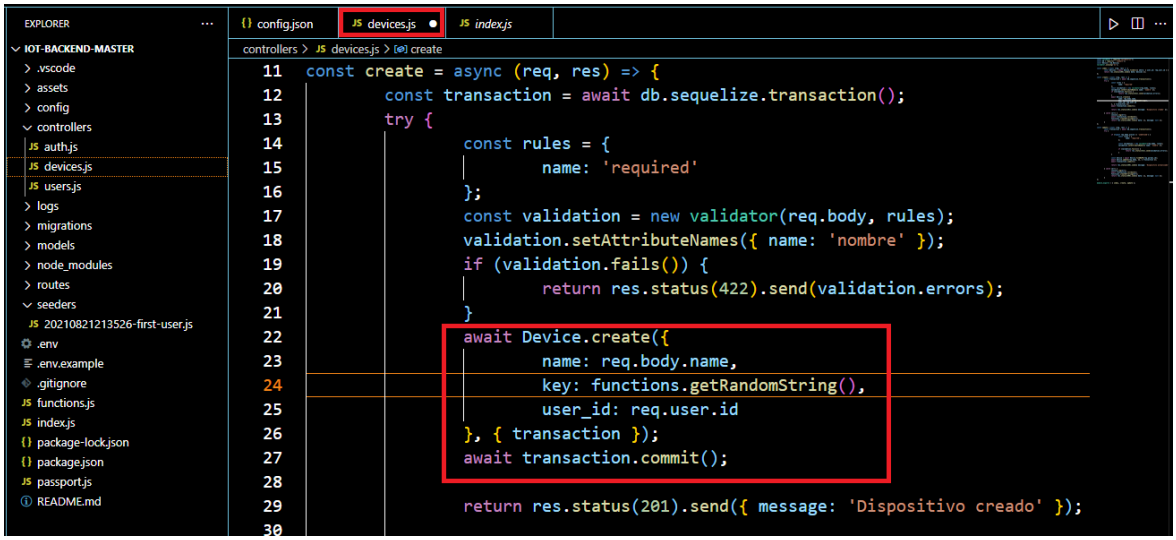
Función a nivel global



El directorio controllers contiene el archivo devices que contiene toda la programación necesaria para crear dispositivos, una vez que el usuario ingresa el nombre del nuevo dispositivo el programa se encarga de generar un tópico mediante la función *getRandomString*, si el tópico se repite genera otro, de modo que a cada dispositivo se le asigne un tópico único e irrepetible. Una vez que se asigna el tópico la información se envía a base de datos.

Figura 43

Asignación de tópicos



Framework para graficar datos. Para el diseño de la plataforma se usó el framework Vue.js y varias de sus bibliotecas, hay un pequeño problema específicamente en la biblioteca Vuetify debido a que su versión 3 es una beta para pruebas y no está diseñada para aplicaciones productivas, por lo tanto, es necesario instalar Vuetify versión 2 y para que no se presenten problemas de compatibilidad se requieren instalar Vue.js versión 2 y sus bibliotecas Vue Router versión 3, Vuex versión 3 y Vuetify versión 2.

Figura 44

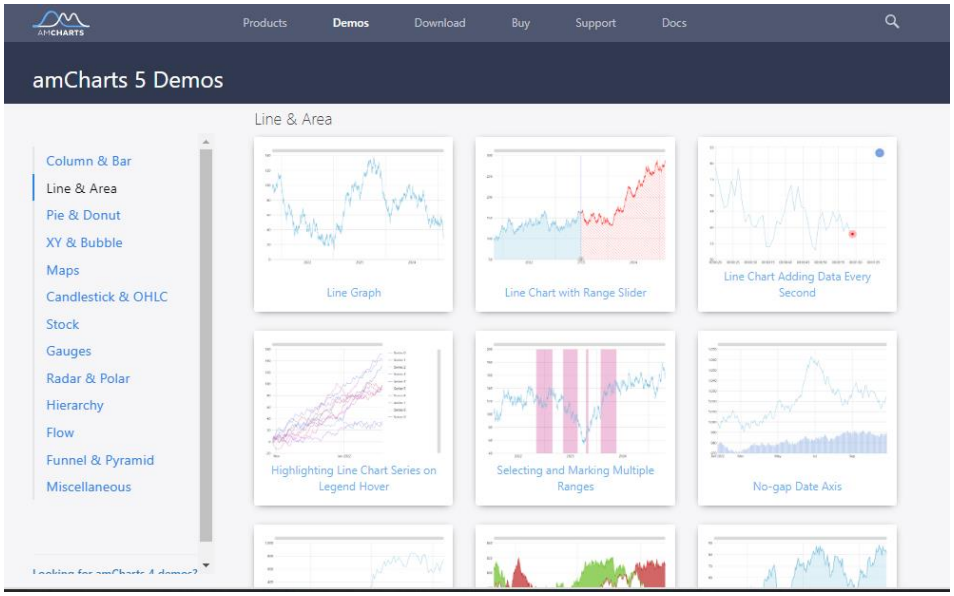
Selección manual de versiones de Vue.js

El uso de las bibliotecas permite extender las funcionalidades de Vue.js y además los componentes son reutilizables, esto es especialmente útil a la hora de presentar los datos que cada sensor obtiene mediante una página con gráficas que se generan automáticamente al crear un dispositivo.

Para crear las gráficas se van a usar los demos gratuitos que ofrece la biblioteca amCharts 5, las gráficas se dibujan inicialmente a partir de números random que se generan por defecto, para instalarla basta con ejecutar el comando *npm install @amcharts/amcharts5* en el terminal del frontend.

Figura 45

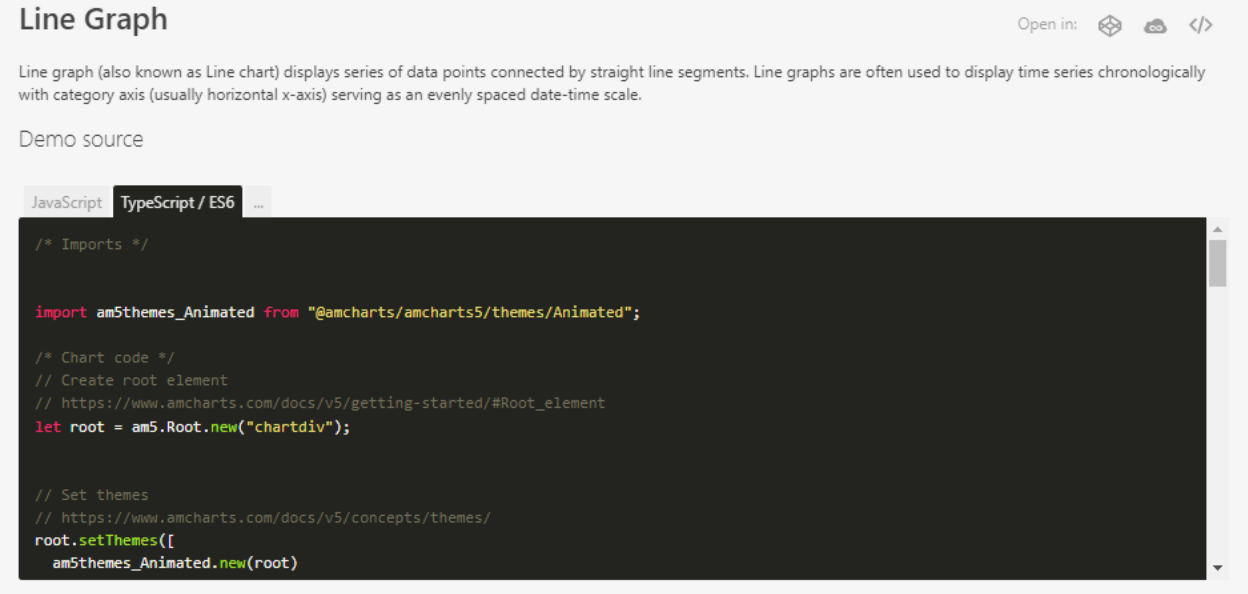
Demos para visualización de datos amCharts 5



Nota. Demos para graficar curvas. Tomado de (amCharts, 2022).

Figura 46

Código demo para Line Graph de amCharts 5



Nota. El código debe ser descargado siempre en el estándar de JavaScript ECMAScript 6, en la página se abrevia como ES6. Tomado de (amCharts, 2022).

Figura 47

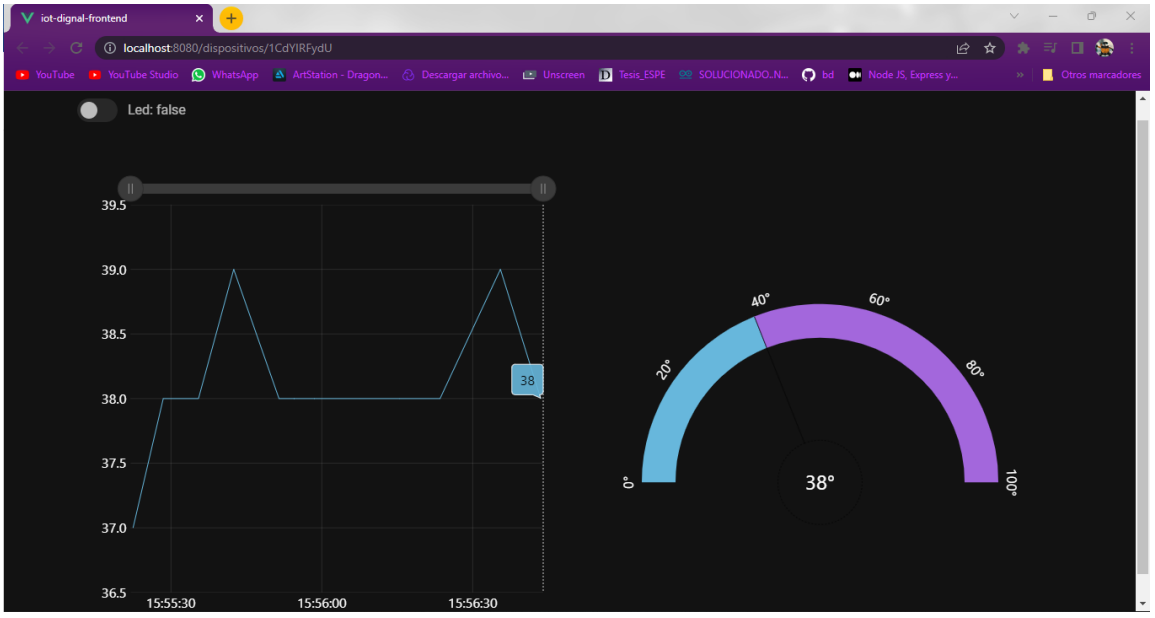
Ejemplo de función que genera números random de Amcharts 5

```
// Generate random data
let date = new Date();
date.setHours(0, 0, 0, 0);
let value = 100;

function generateData() {
  value = Math.round((Math.random() * 10 - 5) + value);
  am5.time.add(date, "day", 1);
  return {
    date: date.getTime(),
    value: value
  };
}
```

Nota. Posteriormente los números aleatorios deben ser reemplazados por los valores obtenidos por los sensores. Tomado de (amCharts, 2022).

Figura 48

Line chart de valores random y Gauge chart con el último valor registrado



Nota. La gráfica corresponde al dispositivo destinado a medir temperatura que se muestra en la Figura 41.

La distribución de los componentes se realiza con el sistema de cuadrícula de 12 puntos de Vuetify que ayudan a organizar el contenido de la página a manera de plantilla que está ligada a breakpoints los cuales hacen referencia al tamaño y orientación de la pantalla de un dispositivo como por ejemplo una computadora o un celular.

Figura 49

Display Breakpoints

Device	Code	Type	Range
Extra small	xs	Small to large phone	< 600px
Small	sm	Small to medium tablet	600px > < 960px
Medium	md	Large tablet to laptop	960px > < 1264px*
Large	lg	Desktop	1264px > < 1904px*
Extra large	xl	4k and ultra-wide	> 1904px*

Nota. Permite controlar el aspecto de la aplicación dependiendo del tamaño de la pantalla.

Figura 50

Ejemplo de uso de breakpoints de Vuetify

```
<v-row>
    <v-col cols="12" md="6">
        <LineChart :socket="socket"/>
    </v-col>
    <v-col cols="12" md="6">
        <GaugeChart :socket="socket" />
    </v-col>
</v-row>
```

Como resultado las gráficas se posicionan una debajo de la otra en dispositivos cuya pantalla tiene un tamaño que se clasifique como médium, por lo que la página funcionará bien si se accede desde una computadora, Tablet, laptop y celulares grandes. Los resultados pueden verse con ayuda de las herramientas del navegador.

Figura 51

Resultado del uso de breakpoints para el line chart

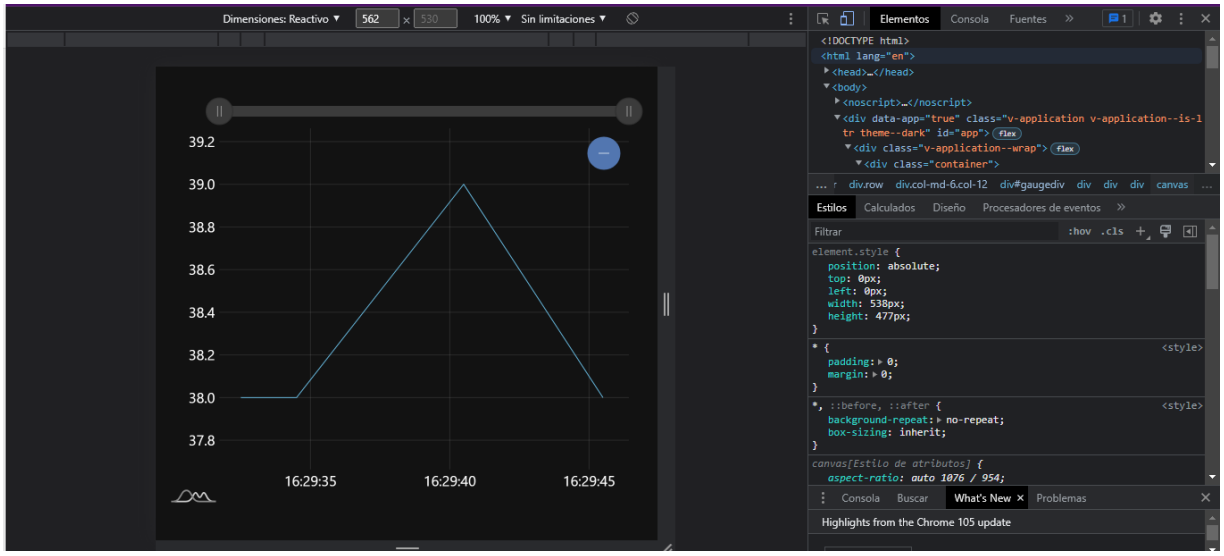


Figura 52

Resultado del uso de breakpoints para el Gauge chart

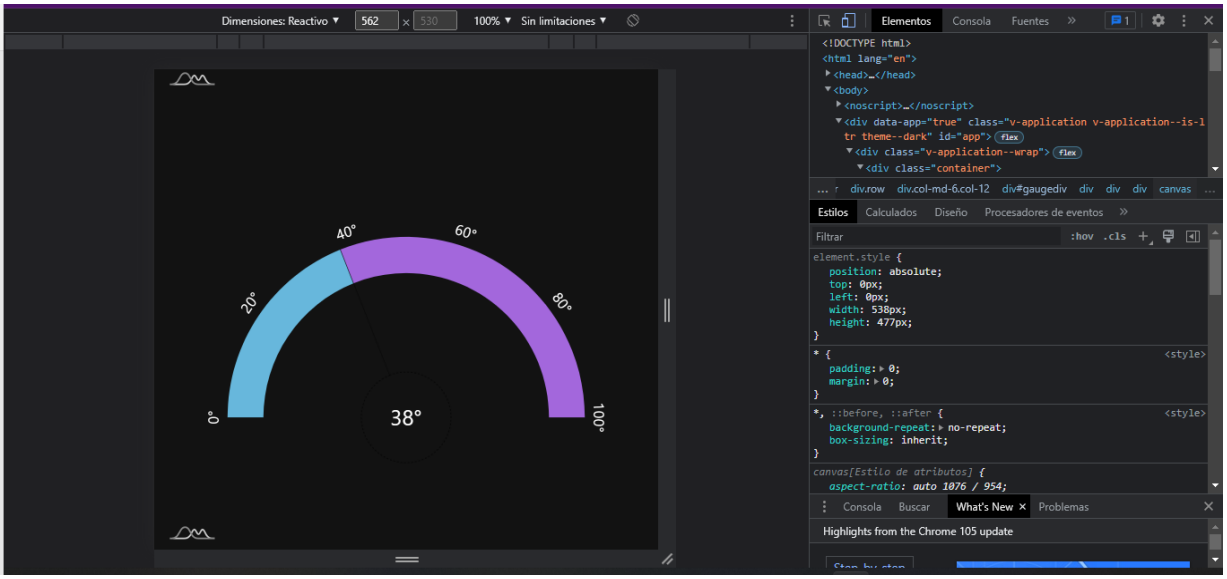
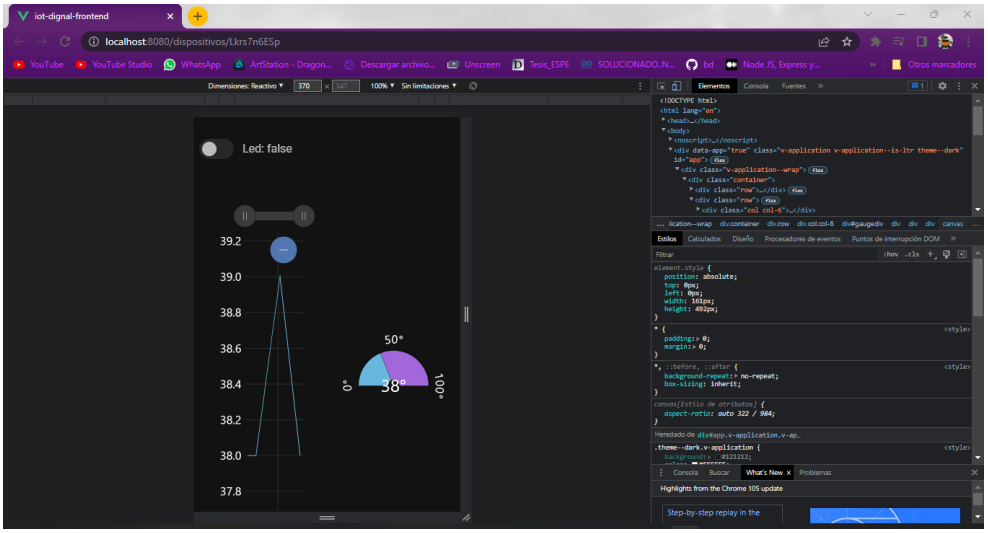


Figura 53

Resultado que se obtiene al no tomar en cuenta los breakpoints



Nota. Las gráficas se ven demasiado pequeñas debido a que ambas se ubican en la misma fila.

Es hora de usar MQTTX para simular sensores, para el intercambio de información se usará el formato JSON que es el que mejor se adapta para aplicaciones de RPM debido a que es ligero, es independiente del lenguaje de programación por lo que es fácil de integrar tanto en la plataforma como en los dispositivos de IoT, permite mejoras en la IU al facilitar la implementación de páginas web dinámicas y las máquinas no ocupan muchos recursos para interpretar y generar estos datos.

Figura 54

Crear nuevo cliente en MQTTX

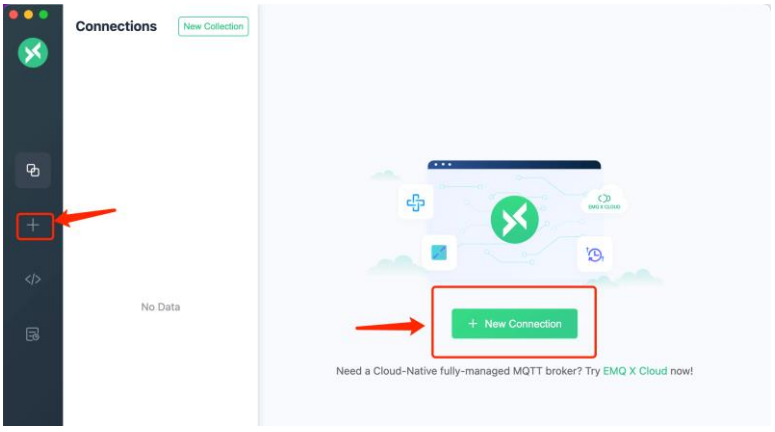
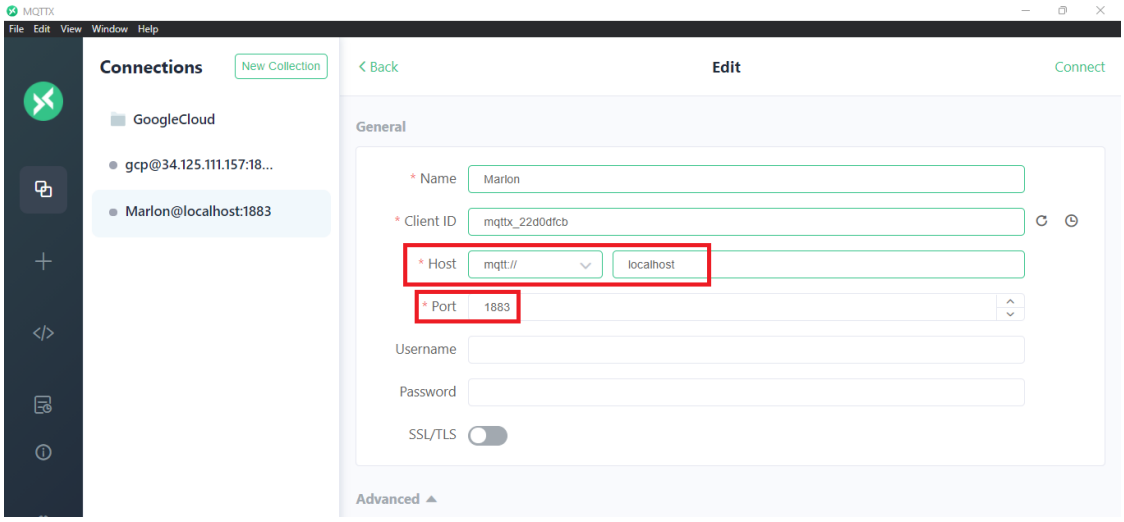


Figura 55

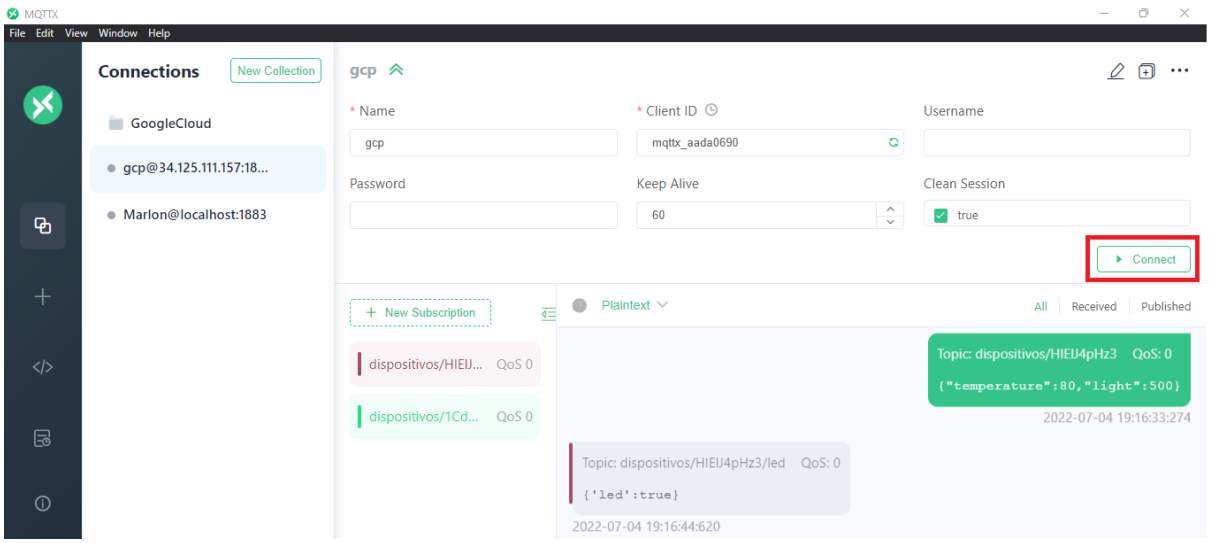
Configuración del cliente



Nota. Para las pruebas es suficiente con configurar el host y el puerto, por defecto se usa el puerto TCP 1883 para MQTT, garantiza la entrega de paquetes de datos en el orden en que fueron enviados.

Figura 56

Conectar cliente MQTTX

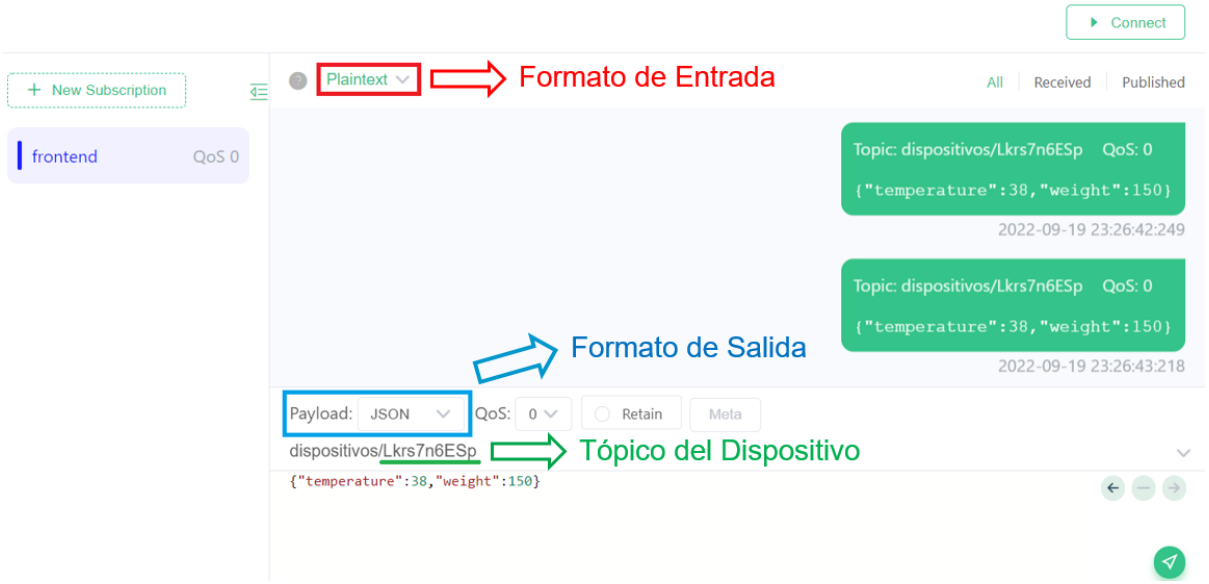


Nota. Es necesario ejecutar el broker EMQX como se muestra en la Figura 30 para que el cliente pueda conectarse.

Para el intercambio de información hay que introducir una URL, en este caso *dispositivos/* que será la misma para todos los sensores y a continuación el tópico que los diferencia, con el menú de flecha superior se puede especificar el formato en el que se recibe información y en el Payload de la parte inferior se selecciona el formato de envío de información.

Figura 57

Configuración para intercambio de mensajes MQTTX



Para el envío de información en formato JSON los elementos principales al trabajar con objetos son la clave y su valor correspondiente, la clave será definida por medio del nombre del parámetro biométrico que mide el sensor al que corresponde el tópico e irán entre comillas dobles. Las llaves son obligatorias y señalan el inicio y fin de un objeto, además, se usan algunos otros símbolos como los dos puntos, que separan la clave del valor que se le ha asignado o la coma, que separa los elementos. Otra regla importante es que los valores true, false y null no van entre comillas, si bien las reglas sintácticas del formato JSON son sencillas solo se han señalado las de relevancia para la implementación de la plataforma.

Figura 58

Envío de mensajes en formato JSON MQTTX

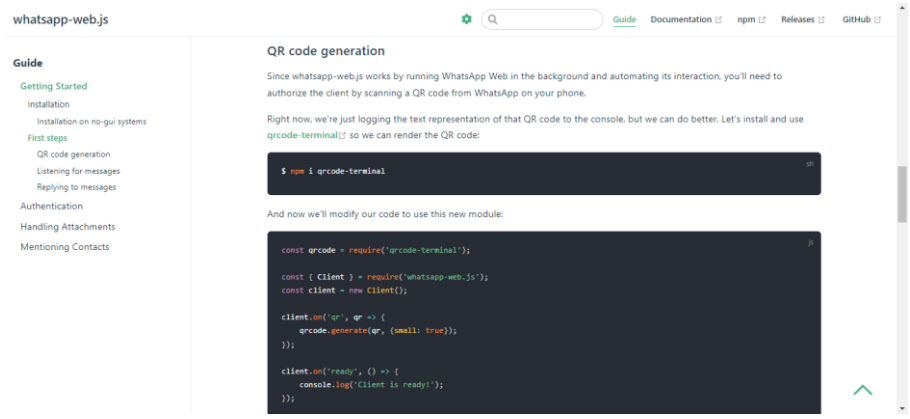
Nota. Simulación de transferencia de información para los sensores de temperatura y peso con MQTTX utilizando el formato JSON para el envío de información.

Bot de WhatsApp

Se trabaja como un nuevo proyecto a partir de la librería WhatsApp-web.js para ser ejecutado como cliente del backend, de manera muy similar al frontend el intercambio de información se realizará por medio de WebSocket. En el directorio de este nuevo proyecto se instala WhatsApp-web.js mediante el comando *npm i whatsapp-web.js,* la librería genera una representación en formato texto del código QR, para solucionar esto hay que usar el comando *npm install -g qrcode-terminal,* este comando instala una librería capaz de generar el código QR, tal y como se observa en la Figura 11.

Figura 59

Página web oficial de WhatsApp-web.js



Nota. La página contiene una guía de configuración de los servicios que ofrece la librería. Tomado de (López, 2022).

Como el proyecto se va a montar en una máquina virtual con una versión de Linux sin GUI y hay que escanear el código QR que se genera para poder conectarse a la aplicación para navegador de WhatsApp, se añade una función que muestra el código QR en una dirección URL estática, cuando se monte el proyecto en internet se podrá acceder sin problema desde cualquier navegador.

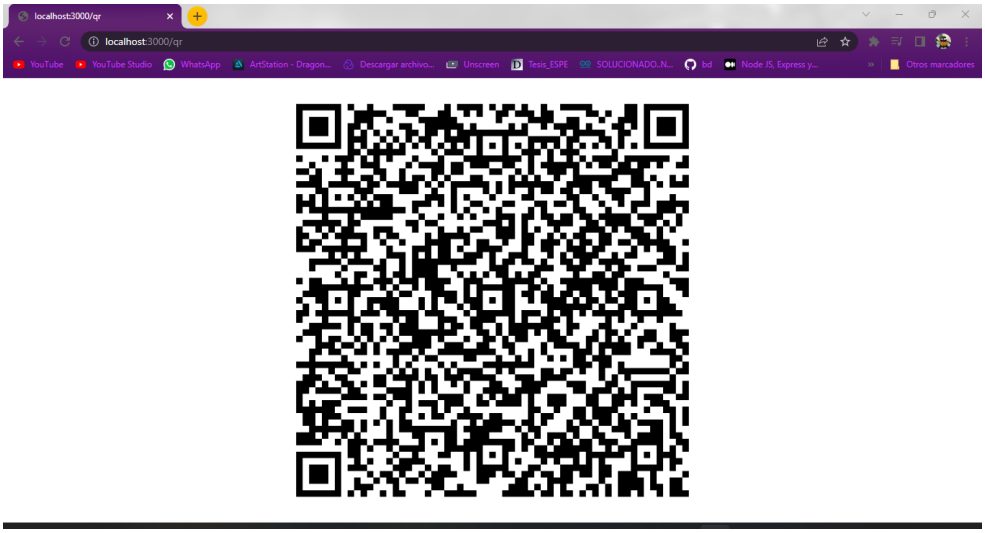
Figura 60

Abrir código QR en un navegador web

Nota. Función y URL necesarios para abrir el código QR en un navegador web.

Figura 61

Generar código QR en URL estática



Para guardar la sesión la librería ofrece varios métodos, de este modo no será necesario volver a escanear el código cuando se desconecte el cliente, es decir, cuando se deje de correr el programa. La opción más atractiva es LocalAuth Strategy, su único inconveniente es que no funciona adecuadamente con plataformas de Cloud Hosting que proporcionan técnicas de archivos efímeros como Heroku, pero al no ser el caso del Host de Google se puede implementar sin inconvenientes. Al iniciar sesión se genera un directorio *.wwebjs_auth* que almacena toda la información necesaria para recuperar la sesión cuando se vuelve a ejecutar el programa.

Luego de escanear el código, mediante las funciones de escuchar y recibir mensajes se pueden efectuar las primeras interacciones con el Chatbot por medio de condicionales. En la siguiente imagen se presenta un ejemplo que abarca varias características útiles de la librería, por medio de la función listenMessage se obtienen el número de celular del usuario que envía un mensaje, el número del usuario que lo recibe (el dispositivo que escaneó el código QR) y el texto recibido. El condicional ejecuta la función sendMessage que corresponde a la solicitud que se ha realizado. Si se ha solicitado un reporte, el bot responde con el valor actual de un

sensor de luz al mismo número que ha realizado la solicitud, también es posible enviar mensajes a números específicos, si se solicita una alarma, el bot envía un mensaje directo al número configurado con el texto *Emergencia!!*.

Figura 62

Ejemplo de intercambio de mensajes WhatsApp – Web .js

```
const listenMessage =() =>{
        client.on('message',(msg) =>{
                const {from, to, body}=msg;
                console.log(from, to ,body);

                switch(body){
                        case 'Reporte':
                                sendMesaage(from, `Luz es igual a ${Luz}`)
                                break;
                        case 'Alarma':
                                sendMesaage('593992996190@c.us' , 'Emergencia!!');
                                break;
                }

        })
}
const sendMesaage =(to, message)=>{
        client.sendMessage(to, message)
}
```

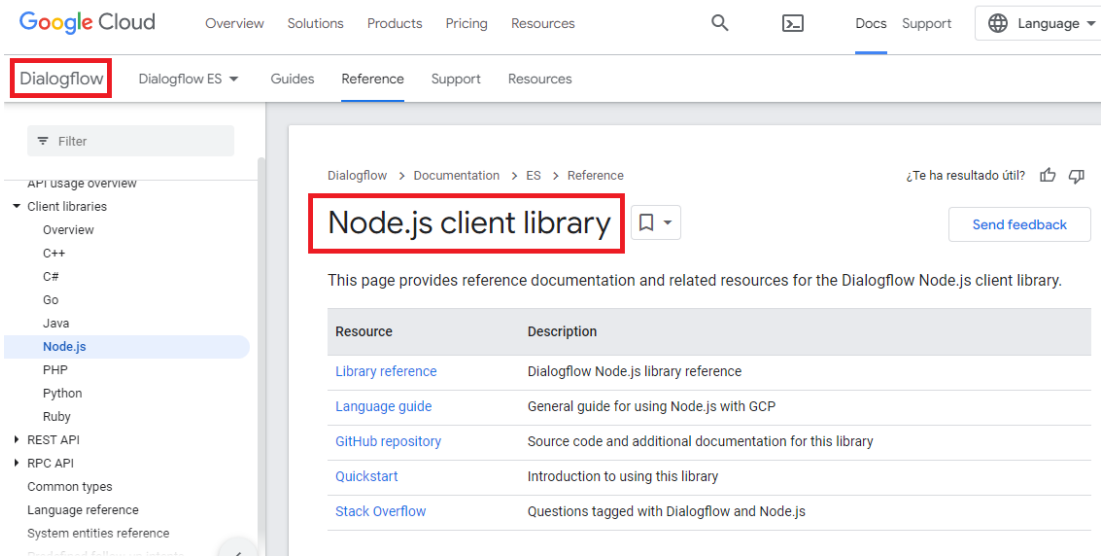
Cabe recalcar que al trabajar con condicionales los caracteres deben ser exactamente los mismos para obtener respuesta y es prácticamente imposible escalar el proyecto, para solucionar este inconveniente se hace uso del servicio de Chatbot de Google que se conoce como DialogFlow.

Integrar DialogFlow

Se hace uso de la librería que el mismo DialogFlow pone a disposición desde su página oficial para elaborar proyectos en node.js, junto con documentación de referencia y recursos relacionados. Las aplicaciones integradas se observan en la Figura 10, para cualquier otra aplicación si se quiere que el usuario final interactúe con el bot de DialogFlow hay que configurar una API que realiza consultas al agente en cada turno de la conversación.

Figura 63

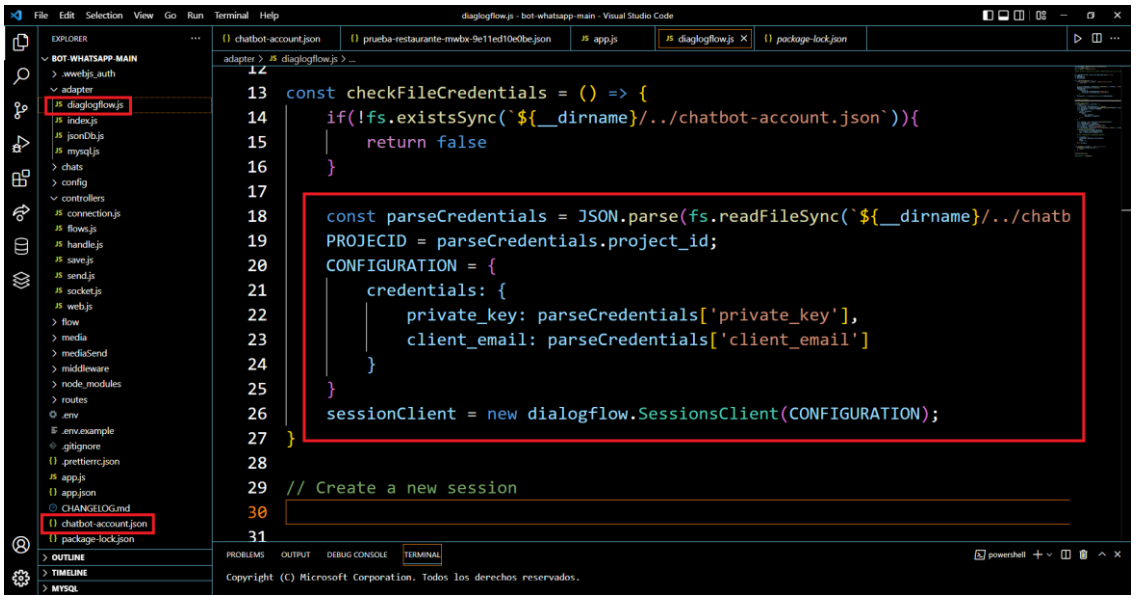
Página de descarga de Client Library de DialogFlow



La programación que permite al usuario final interactuar con el agente se encuentra en la raíz del proyecto, dentro del directorio adapter y en el archivo dialogflow.js, trabaja en conjunto con el archivo *chatbot-account.json,* también se ubica en la raíz y contiene las credenciales que se generan al crear la cuenta de servicio de Google.

Figura 64

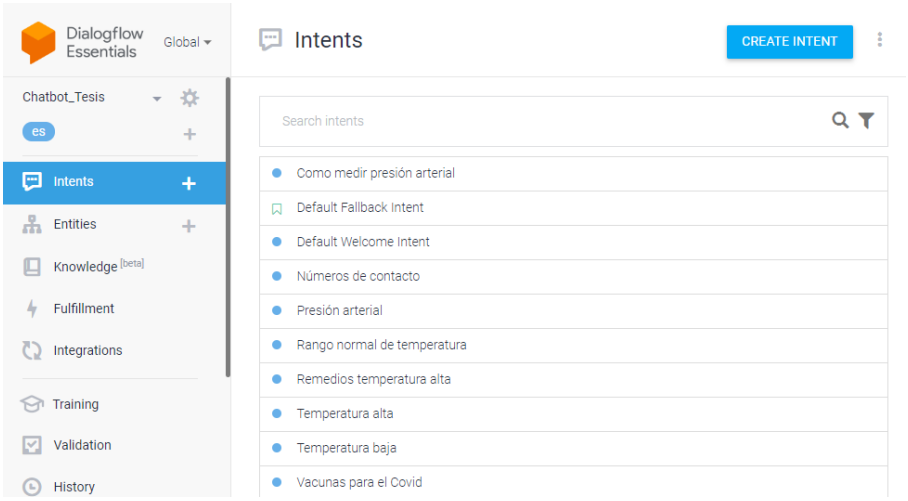
Node.js Client Library



En DialogFlow lo primero es crear un agente, además del nombre se configura el idioma y la zona horaria. Lo siguiente es crear las Intenciones o Intents, lo común es crear una para cada objetivo que tenga el chatbot, por defecto vienen dos, una en caso de que se produzca algún error y la otra es una intención de bienvenida, su objetivo por ejemplo sería responder a saludos típicos como hola y buenos días que son expresiones que usan los usuarios al no saber cómo iniciar la conversación, de no existir esta intención el bot iría directamente a la intención de error y la interacción no fluiría con normalidad.

Figura 65

Intents para preguntas frecuentes de pacientes del centro de salud

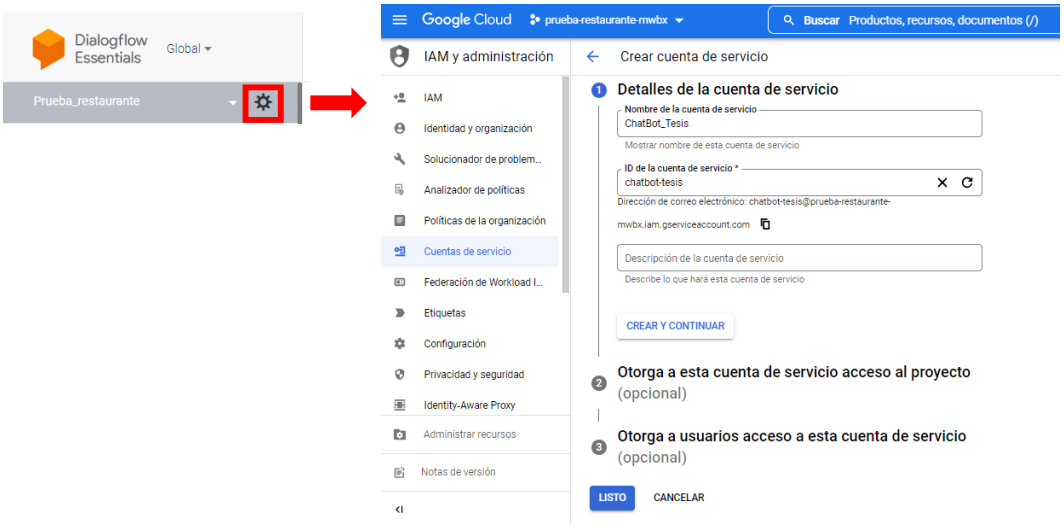


El servicio cuenta con más herramientas que facilitan las interacciones y permiten dar respuestas más precisas y personalizadas, pero se requieren más líneas de código para su integración a node.js y los errores se vuelven más frecuentes por lo que es recomendable usarlos solo si el sistema ha estado funcionando previamente y los resultados han sido satisfactorios, la propia plataforma ofrece datos sobre las interacciones que muestran si se requiere reentrenar al bot o en qué circunstancias se presentan más errores o cuales son las preguntas más frecuentes de los usuarios. Esta información es la que permite escalar el Chatbot, de otro modo se estaría invirtiendo tiempo en un sistema que no se sabe si al final va a funcionar o no.

Generar archivo de Credenciales. Para conectar al proyecto con DialogFlow hay que crear el archivo con las credenciales.

Figura 66

Crear cuenta de Servicio



Nota. Solo llenar el nombre y luego clicar en Listo.

Figura 67

Otorgar permisos

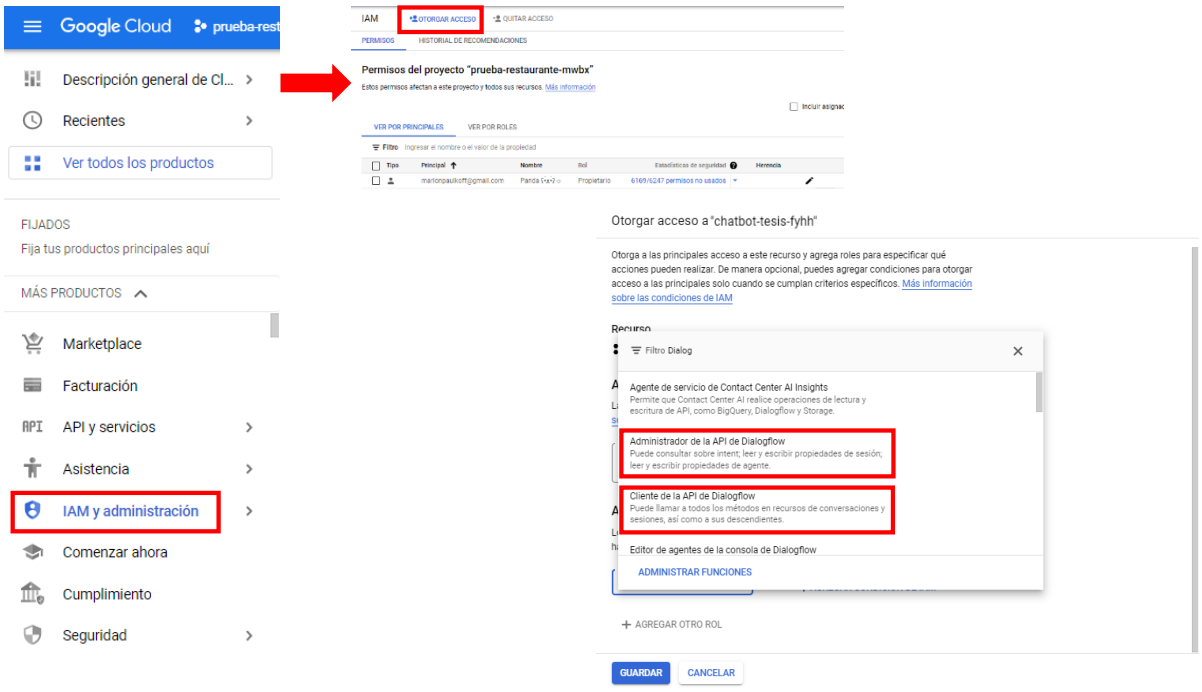
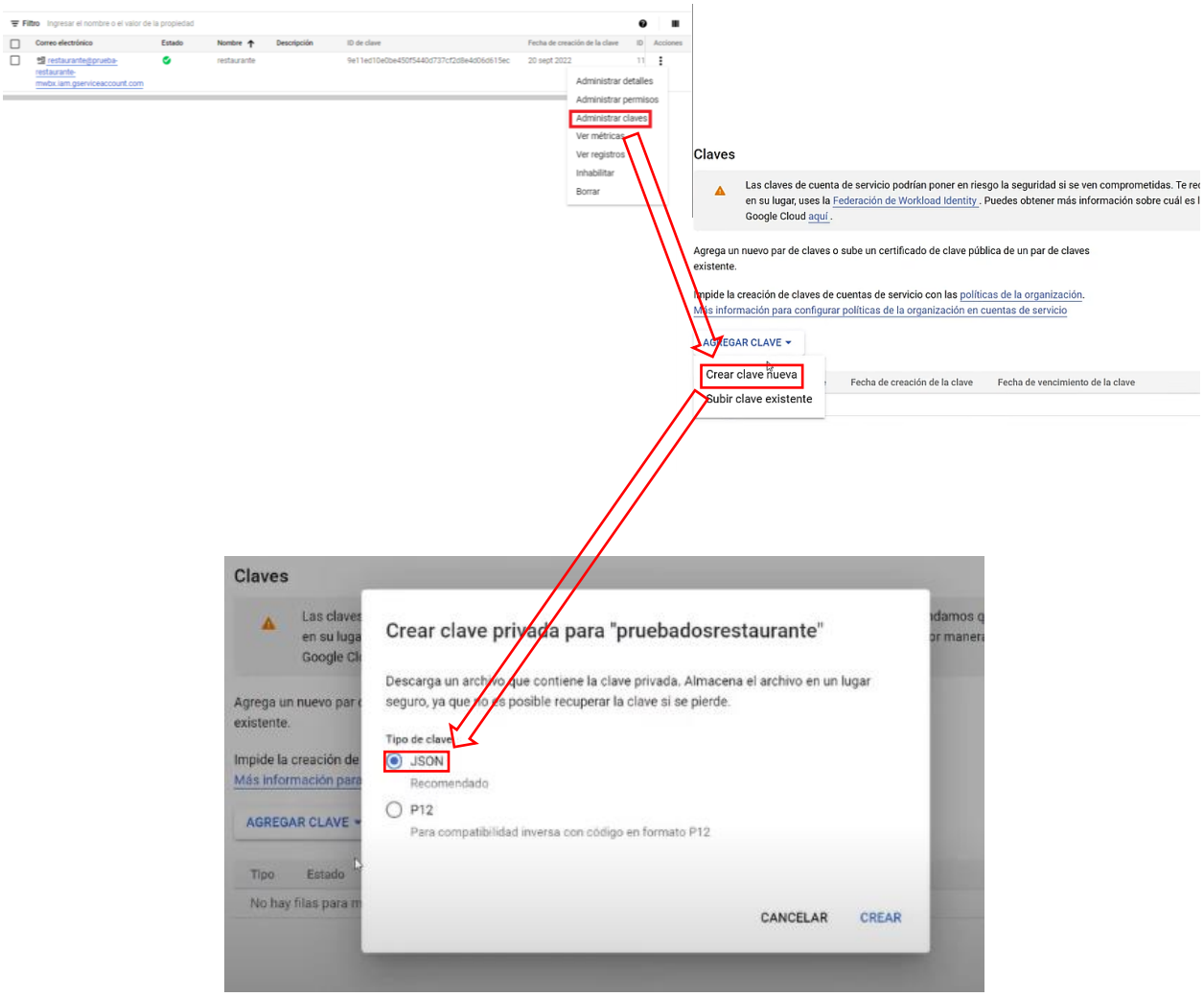


Figura 68

Generar archivo de credenciales

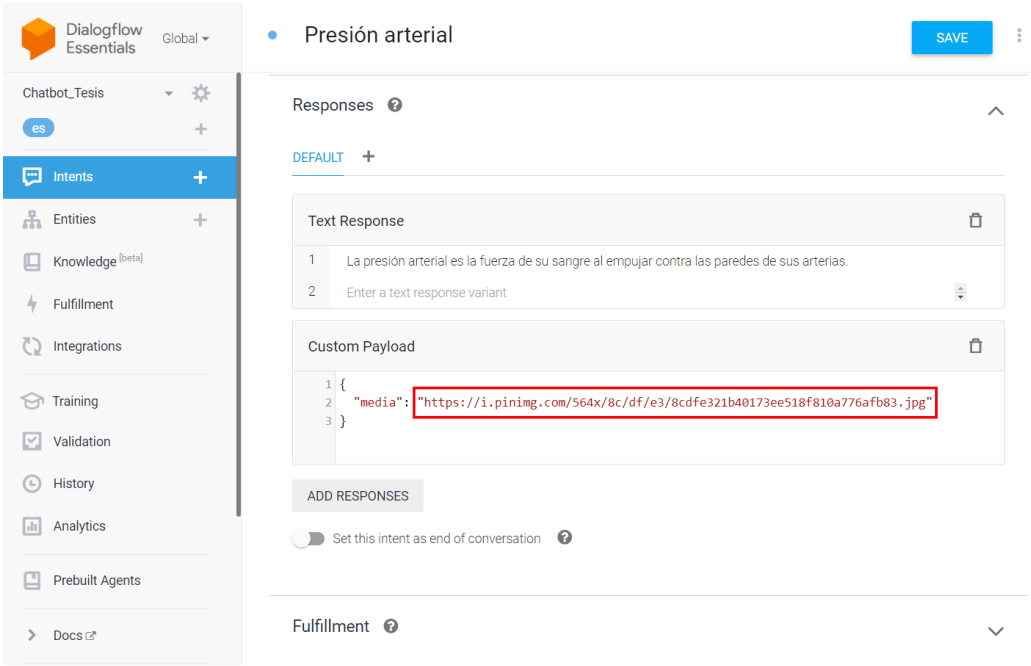


Nota. Al finalizar el proceso se descarga automáticamente el archivo con las credenciales.

Una vez que se actualizan las credenciales en el archivo chat-bot-account.json ya se puede usar el agente. Para responder preguntas frecuentes como conceptos, el uso de sensores, o números de contactos se han elaborado varias imágenes que pueden ser añadidas al contenido multimedia disponible desde el mismo DialogFlow pero debe estar disponible en la web para acceder mediante el link.

Figura 69

Añadir imágenes a DialogFlow

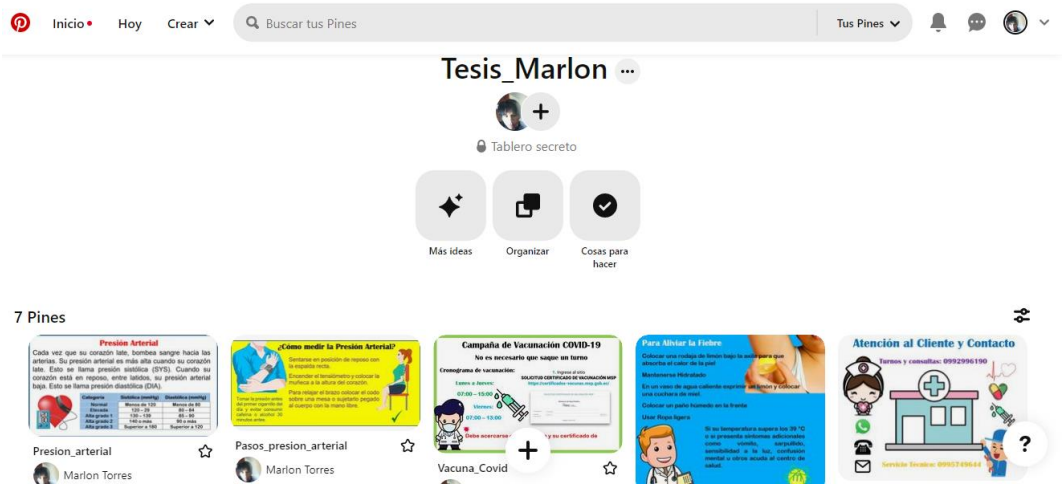


Nota. El link del recurso se especifica en formato JSON.

Los servicios de almacenamiento en la nube más populares como Google Drive, Mega, OneDrive, entre otros, no permiten hacer públicos en la web los archivos como hace un par de años, una forma segura y económica por la que se ha optado es subir las imágenes a Pinterest, se define como una red social visual en la que es posible compartir y buscar imágenes, videos y todo tipo de contenido visual, ofrece la opción de colocar las imágenes que un usuario sube en privado, esto evita que puedan ser encontradas mediante una búsqueda en internet pero cualquier persona con el link podrá acceder a la imagen, se hace uso de esta característica para enviar imágenes mediante DialogFlow, gracias a esto una vez que se elabora la imagen se sube y configura en menos de un minuto.

Figura 70

Almacenamiento de imágenes para usar con DialogFlow



Nota. Las imágenes que se guarden dentro del tablero secreto solo se podrán visualizar únicamente mediante su link de acceso.

La dirección que se ingresa en el Custom Payload de DialogFlow corresponde al link que muestra el navegador al abrir la imagen en una pestaña nueva.

Figura 71

Obtener el link de una imagen de Pinterest

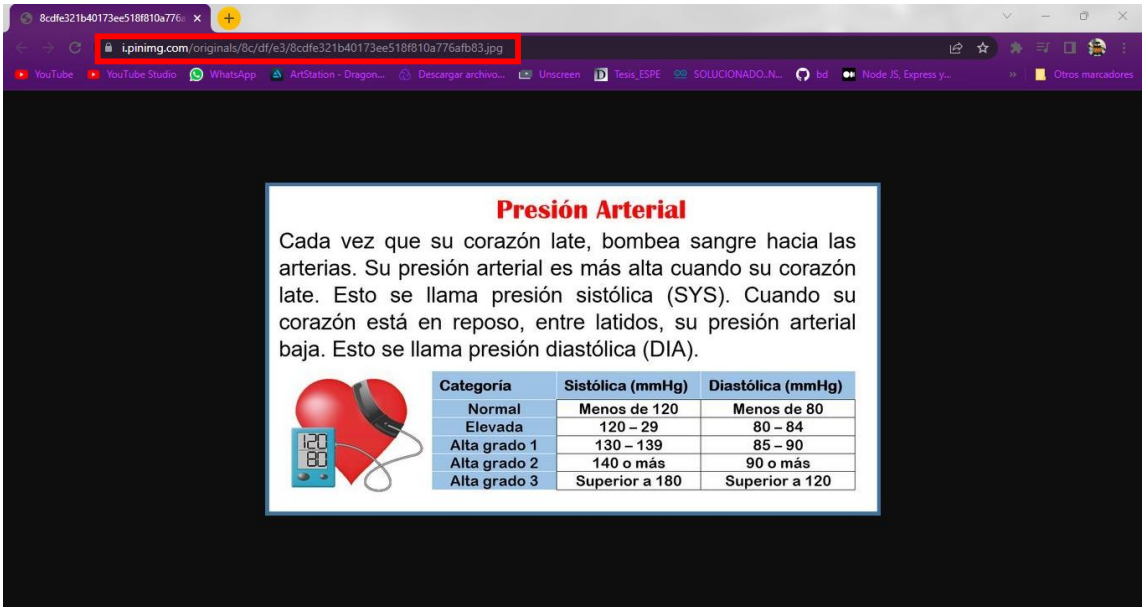
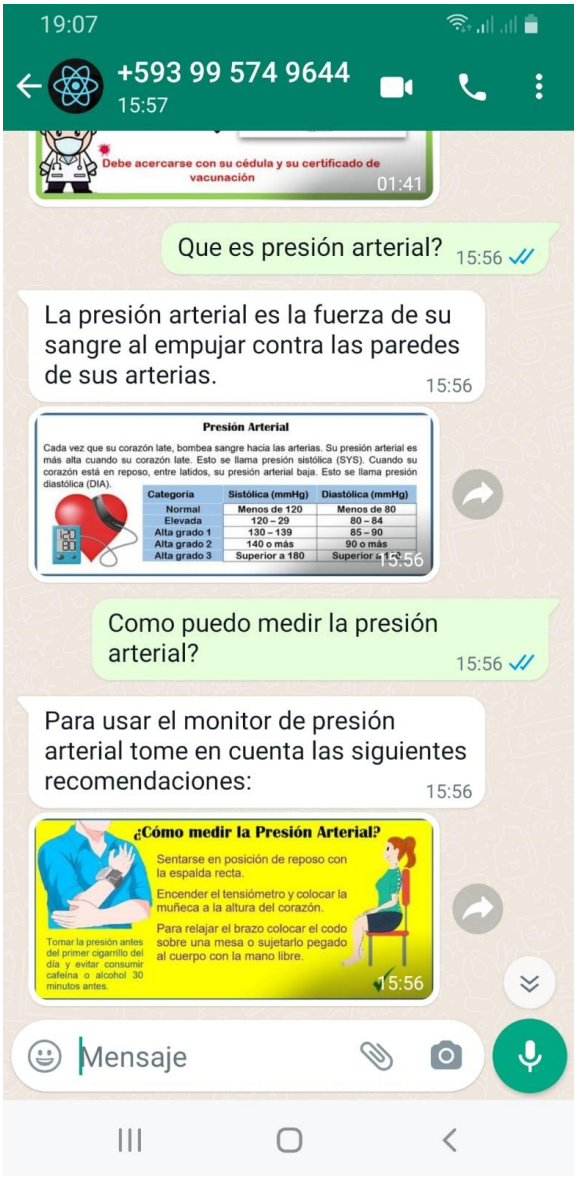


Figura 72

Envío de imágenes por WhatsApp con DialogFlow



Un Chatbot básicamente se encarga de responder a preguntas frecuentes de los usuarios de una plataforma, para el sistema de monitorización se busca obtener los datos subjetivos de los pacientes en momentos clave, los usuarios no siempre saben cómo iniciar la interacción o están dispuestos a suministrar información de su día a día, para solucionar este inconveniente los datos registrados por los sensores se envían vía WhatsApp a los usuarios junto con preguntas del tipo ¿Cómo se siente hoy? o ¿Desearía reportar algún síntoma?. Para

poder enviar este tipo de mensajes lo fundamental es conocer a qué persona corresponde un dispositivo, justo después de que registra la lectura de un parámetro biométrico se ejecuta una función que efectúa una búsqueda en la base de datos, por medio del id del dispositivo localiza el id del usuario al que pertenece y con eso encuentra su respectivo número de celular. Toda esta información se encuentra del lado del backend por lo tanto es el encargado de esta tarea, una vez que localiza el número de teléfono lo envía por medio de WebSockets al proyecto con el bot de Whatsapp, no hay que olvidar que tanto el frontend como el Chatbot son proyectos separados que funcionan como clientes del backend.

Figura 73

Consulta de número de teléfono en Base de datos

```
socket.emit('watsapp',temperature);
Device.findOne({where:{
        id:device.id}
})
.then((devices) => {
        console.log(devices.id);
        console.log(devices.user_id);
        User.findOne({where:{
                id:devices.user_id}
        })
        .then((cell) => {
                console.log(cell.id);
                console.log(cell.cellphone);
                socket.emit('celu',cell.cellphone);

        });
});
```

Nota. La imagen corresponde al archivo index.js ubicado en la raíz del backend.

Como se puede Observar en la Figura 73 se emiten dos mensajes mediante WebSockets, el primero contiene el valor registrado por el sensor y ejecuta una función de lado del Chatbot que lo imprime por consola, con esto se verifica que la información ha sido recibida por el Chatbot, el segundo mensaje contiene el número de celular y dispara una función del Chatbot que envía un mensaje al Usuario en cuestión con el valor que ha medido el dispositivo y el texto *Hay alguna novedad que desee reportar?.* Eso es todo, cualquier respuesta del usuario o inquietud pasará a ser respondida por el agente de DialogFlow.

Figura 74

Intercambio de información entre el Chatbot y el backend

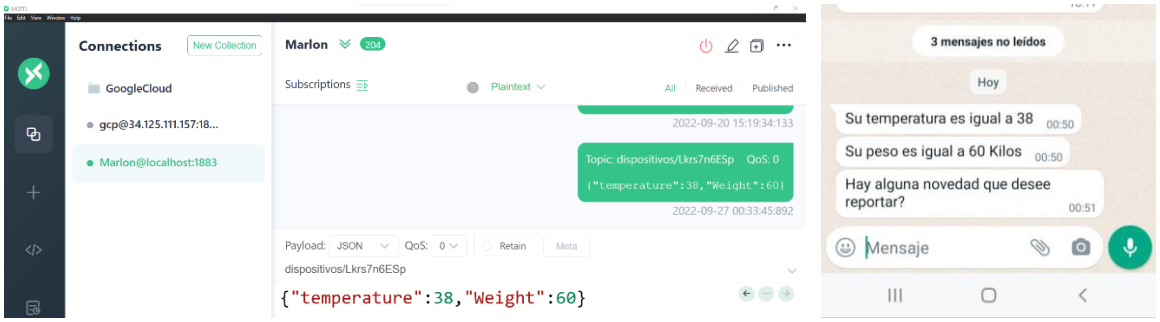
```
socket.on('watsapp', function(watsapp){
    console.log(`Valor de temperatura ${watsapp}`);
    console.log(`Valor de peso ${watsapp_dos}`);
    sensortemp=watsapp;

});

socket.on('celu', function(celu){
    console.log(`El numero de telefono es: ${celu}`);
    sendMesaage(`593${celu}@c.us`, `Su temperatura es igual a ${sensortemp}`);
    sendMesaage(`593${celu}@c.us`, `Su peso es igual a ${Weight}`);
    sendMesaage(`593${celu}@c.us` , `Hay alguna novedad que desee reportar?`);
});
```

Nota. La imagen corresponde al archivo app.js ubicado en la raíz del bot de WhatsApp.

Este procedimiento se puede verificar con ayuda de MQTTX simulando sensores, se pueden hacer pruebas con diferentes usuarios y múltiples dispositivos a la vez. No existe ningún tipo de problema o interferencia entre los mensajes preconfigurados y el bot.

Figura 75

Mensajes directos del bot de WhatsApp



Migrar el proyecto al entorno de producción

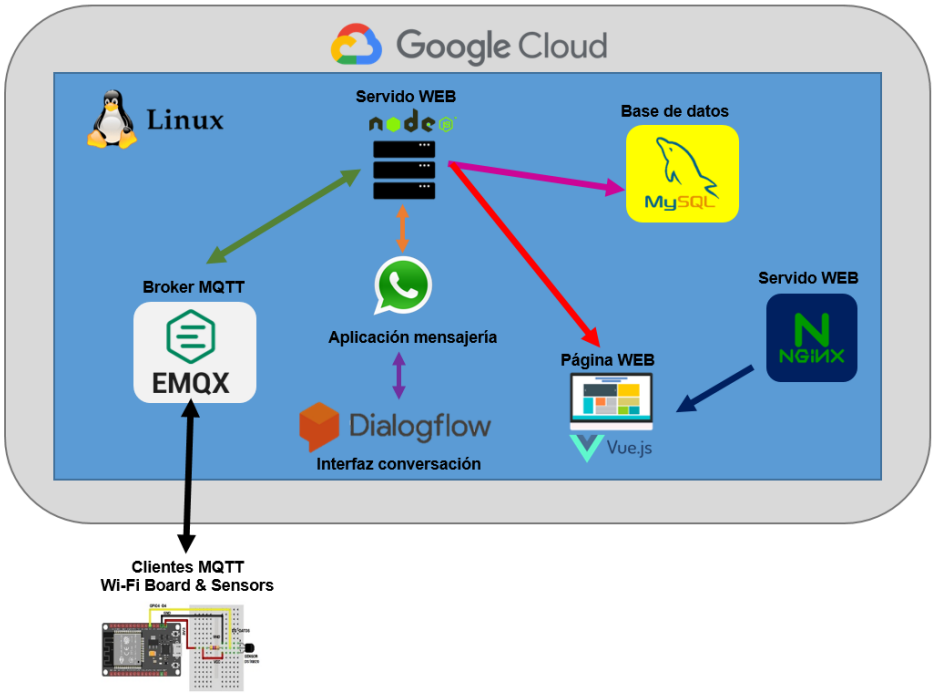
Para el desarrollo se ha tomado en cuenta un concepto llamado ciclo de vida del software, todo el proyecto funciona adecuadamente a nivel local, tanto el backend, frontend y chatbot se comunican sin inconvenientes, no existen problemas de compatibilidad y todos los métodos usados son apropiados por lo que se espera que al subir todo a la nube no hagan faltan numerosas rectificaciones, esto es vital porque a partir de este punto es necesario pagar por los servicios de Google, si la plataforma no trabaja adecuadamente o la corrección de

errores toma demasiado tiempo se generarán pérdidas económicas, por este motivo ahora se habla de que el proyecto se va a ejecutar en un entorno productivo, este entorno es básicamente una máquina virtual montada en los servidores de Google.

Es importante señalar que en la etapa de desarrollo se usó en todo momento el sistema operativo de Windows y ahora se va a implementar sobre una máquina virtual con una versión de Linux sin GUI, esto fue tomado en cuenta desde el inicio por lo que se espera que no se presente ningún impedimento, todo lo que se encuentra dentro de la máquina virtual se probó a nivel local a excepción del Servidor web NGINX que sirve para levantar páginas web, en la fase de desarrollo el frontend se ejecuta con npm por medio de un servicio que Vue.js trae integrado pero cuando la página se cae no se levanta de nuevo de forma automática como lo hace NGINX sino que hay que ejecutar el comando de npm manualmente, la imagen siguiente es una representación de la plataforma en entorno de producción, tiene una mayor infraestructura, mayor capacidad de manejo de tráfico y conexiones simultaneas.

Figura 76

Tecnologías involucradas en la elaboración de la plataforma

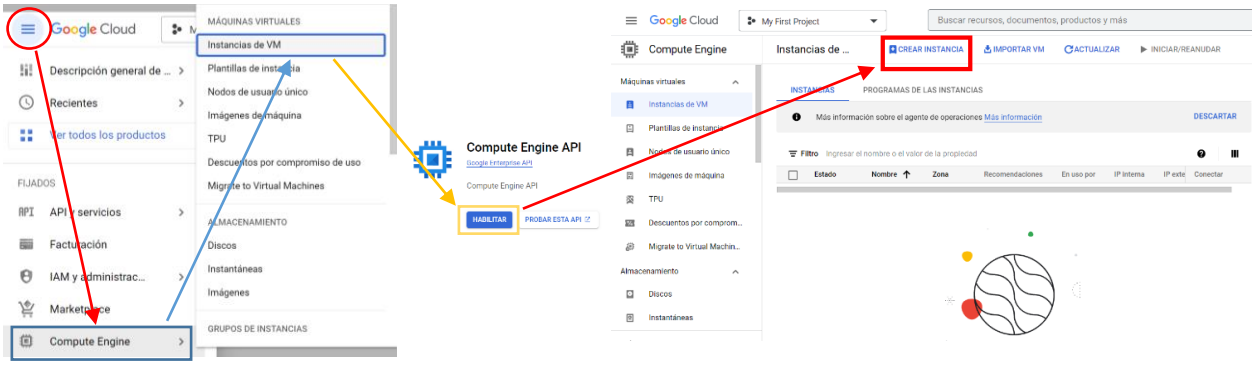


Configurar la Máquina Virtual

Luego de ingresar a la plataforma de Google Cloud hay que habilitar el servicio de compute Engine API y crear una nueva instancia de VM, se accede por medio del menú hamburguesa de la página principal de Google Cloud como se muestra en la Figura 77.

Figura 77

Instancias de Máquina Virtual

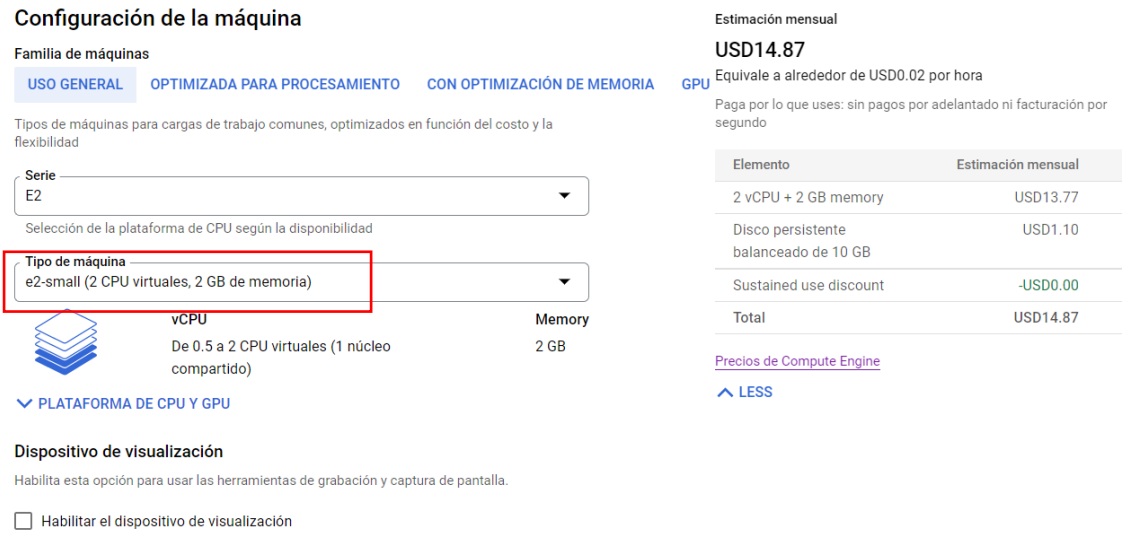


De inmediato se muestra la pantalla de configuración de la máquina virtual junto con una estimación del precio a pagar por mes e inclusive por hora, acompañado por un enlace en el que se pueden visualizar todos los precios relacionados con el servicio de compute engine como el tipo de máquina, el tamaño del disco, balanceo de carga, etc. Hay opciones preconfiguradas que dependen del enfoque que se va a dar a la máquina virtual y que no se van a usar para el presente proyecto ya que su función es optimizar recursos y ninguna opción es adecuada para el sistema de rpm que se busca implementar.

Uno de los apartados relevantes es el tipo de máquina, cambiar de tipo médium a small, este cambio baja la memoria de 4GB a 2GB, el cambio se refleja inmediatamente en la estimación mensual en aproximadamente 15 dólares.

Figura 78

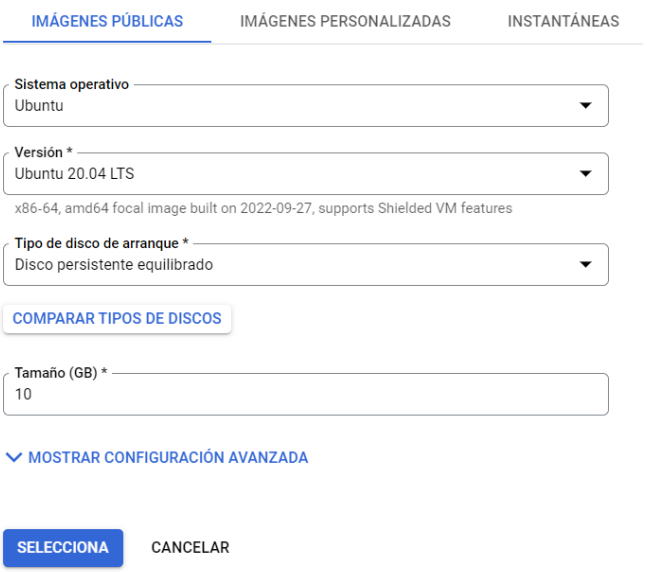
Configuración de la Máquina Virtual



Lo siguiente es cambiar el disco de arranque, desde un inicio se planteó usar Linux, se usa la versión 20.04 LTS, el usar una versión más reciente produciría problemas de compatibilidad con el Broker EMQX.

Figura 79

Configuración del Disco de Arranque de la VM

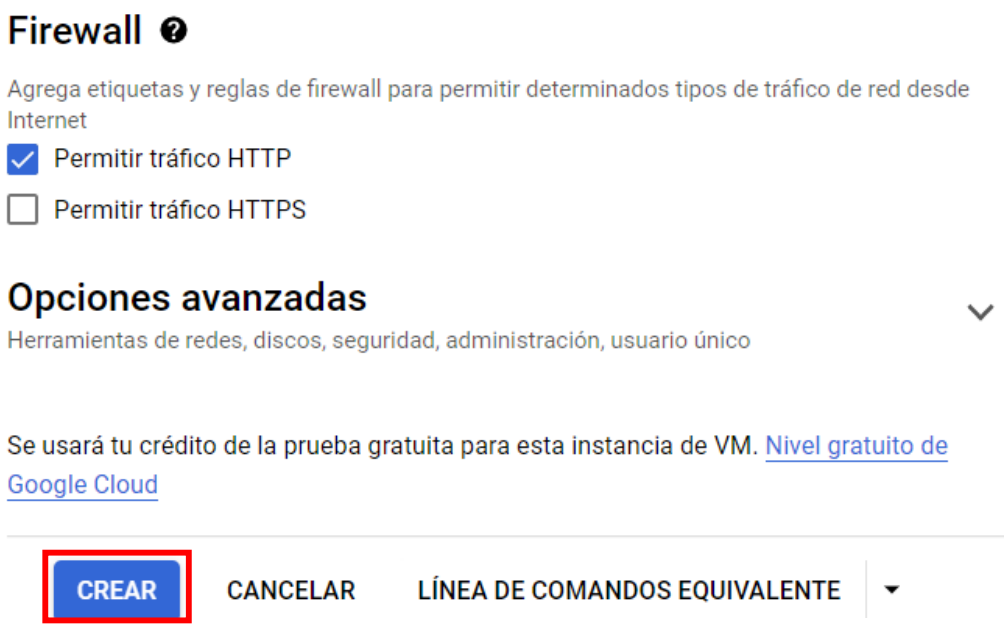


Nota. Esta configuración no afecta en la estimación mensual, el valor será el mismo.

Por último, hay que seleccionar el tipo de tráfico que se quiere permitir, se pueden marcar ambas opciones o únicamente HTTP de no contar con un dominio ni certificados SSL, Google Cloud también cuenta con estos servicios y son fáciles de configurar, pero tienen un valor adicional.

Figura 80

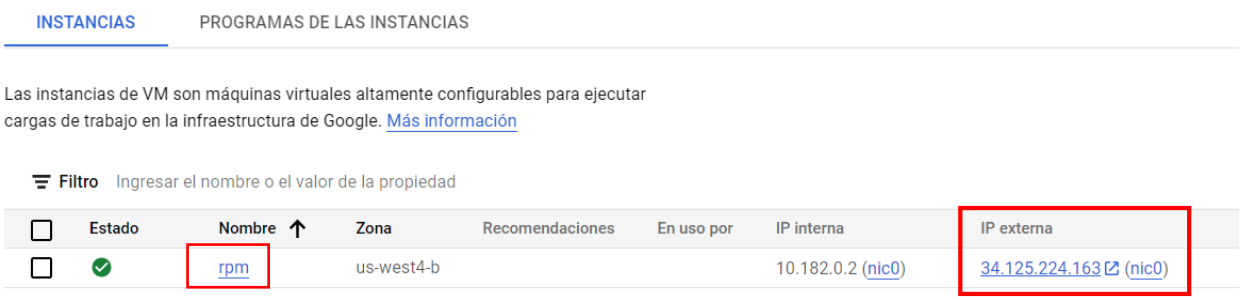
Reglas de Firewall



Se crea la instancia y se muestra junto con las direcciones IP interna y externa, ambas direcciones son efímeras, cambian cada vez que se reinicia el servidor.

Figura 81

Instancias de VM



Para acceder a la plataforma por internet se va a hacer uso de la dirección IP externa, en el apartado del nombre de la VM contiene un enlace que permite cambiar la configuración del servidor, en la sección de interfaz de red.

Figura 82

Configuración de IP externa

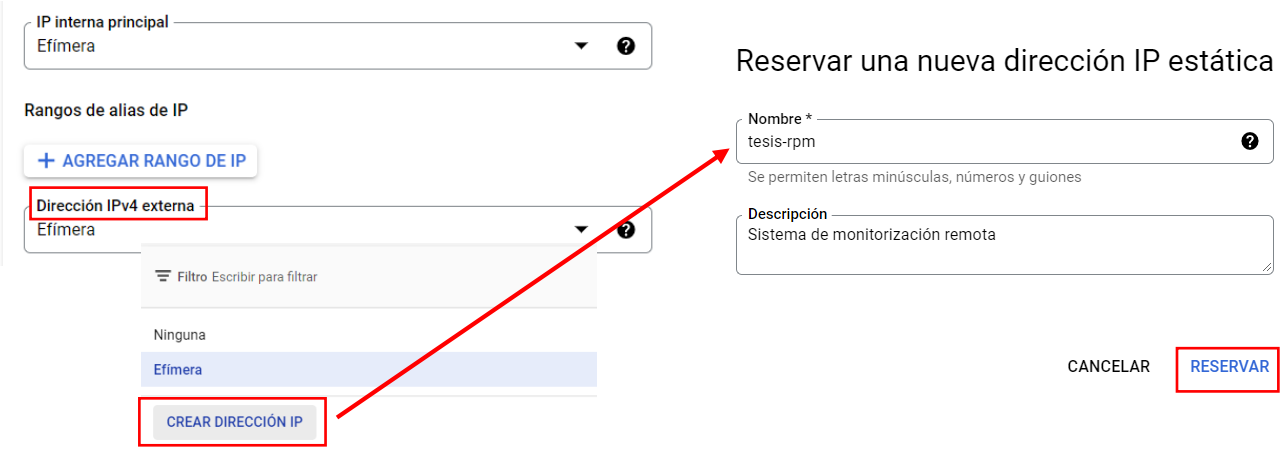


Figura 83

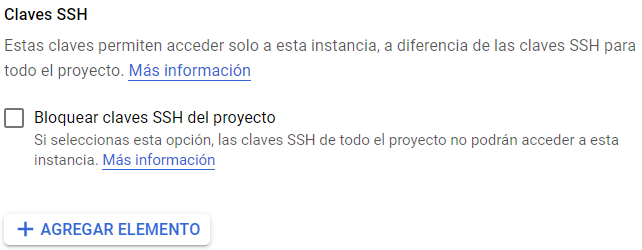
IP externa estática

Nota. Luego de ingresar el nombre se obtiene una dirección IP externa estática que por cierto tiene un costo adicional.

Crear llaves SSH. En la misma página en donde se edita la configuración del servidor, un poco más debajo de la dirección IP externa se encuentran las claves SSH, por medio de llaves SSH que se genera en el equipo local que se utiliza como entorno de desarrollo de la plataforma se puede acceder a la instancia, en pocas palabras permite conectarse de forma remota al servidor, este método es más seguro que usar un usuario y contraseña.

Figura 84

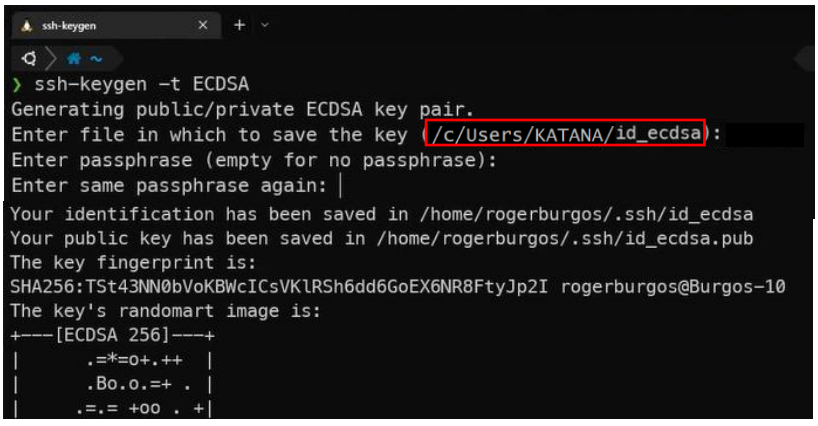
Claves SSH



Para generar las credenciales de acceso y poder autenticarse hay que crear las llaves SSH, consta de dos partes una llave privada que hay que almacenarla en la computadora, luego se configura el servidor para que la reconozca y acepte mediante la llave pública. Las llaves se generan abriendo la terminal, powershell, cmd o Git bash. Para la creación de las llaves existen dos algoritmos muy populares, SRA y ECDSA, ambos con sus ventajas y *desventajas, se elige el algoritmo ECDSA porque es más seguro que SRA y ofrece mayor* rendimiento y escalabilidad debido a sus claves de menor longitud.

Figura 85

Generar llaves SSH

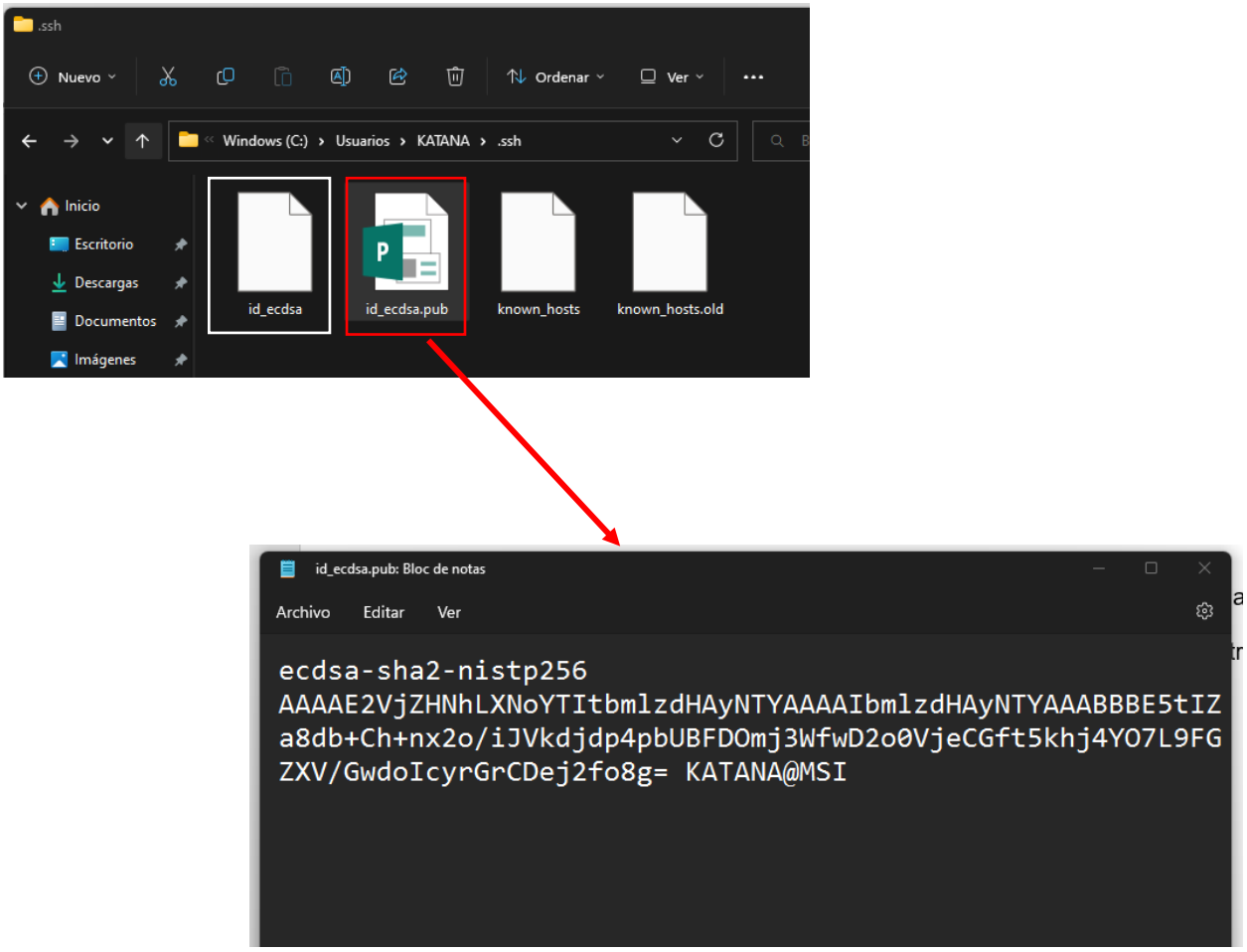


Nota. Se imprime por consola el directorio en donde se van a guardar los archivos con la llave pública y privada, adicionalmente se puede proteger los archivos con contraseña, si no se requiere simplemente se presiona la tecla enter dos veces.

En el directorio se crean dos archivos, uno sin extensión que es la llave privada y el otro con extensión “.pub” que contiene la llave pública, este archivo se puede abrir con cualquier editor de texto como el Bloc de Notas, hay que copiar su contenido y pegarlo en el apartado de la configuración del servidor que contiene las claves SSH y se muestra en la Figura 84.

Figura 86

Llaves SSH



Conexión remota por medio de llaves SSH. Para establecer la conexión se necesitan la IP externa y el nombre de usuario que se asigna a la clave SSH generada anteriormente, ambos parámetros se encuentran en la página de instancia de la máquina virtual “rpm”.

Figura 87

Instancias de la máquina virtual “rpm”

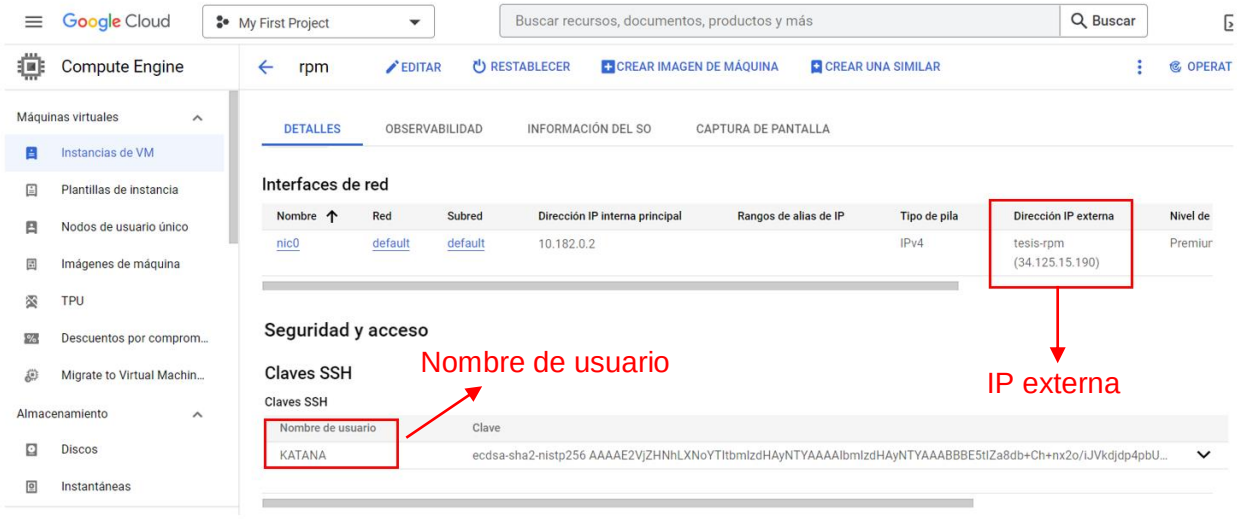


Figura 88

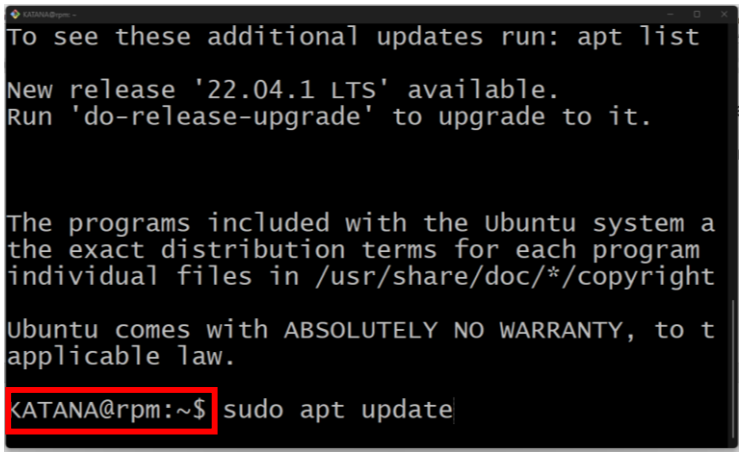
Comando para establecer la conexión remota

Nota. El comando es *ssh KATANA@34.125.15.190 -i "/c/Users/KATANA/.ssh/id_ecdsa”,* luego de ejecutar el comando y confirmar que se desea establecer la conexión se puede trabajar con el servidor sin la necesidad de entrar a la plataforma de Google Cloud.

Como se está trabajando con la versión 20.04.5 LTS de Ubuntu se realizan las actualizaciones típicas mediante los comandos “sudo apt update”, “sudo apt upgrade” y “sudo apt autoclean” y el servidor queda listo para la instalación de las aplicaciones y los proyectos.

Figura 89

Actualización del sistema operativo



Nota. Al conectarse al servidor el terminal empieza a mostrar el nombre de usuario y la instancia de VM.

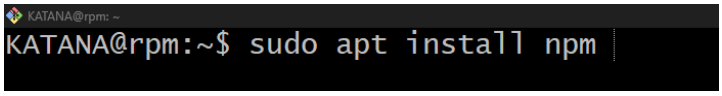
Figura 90

Instalación de NGINX

Nota. Se puede verificar la instalación de NGINX abriendo una pestaña en el navegador con la dirección IP externa, se mostrará el mensaje “Welcome to nginx”.

Figura 91

Instalar npm en la instancia de VM



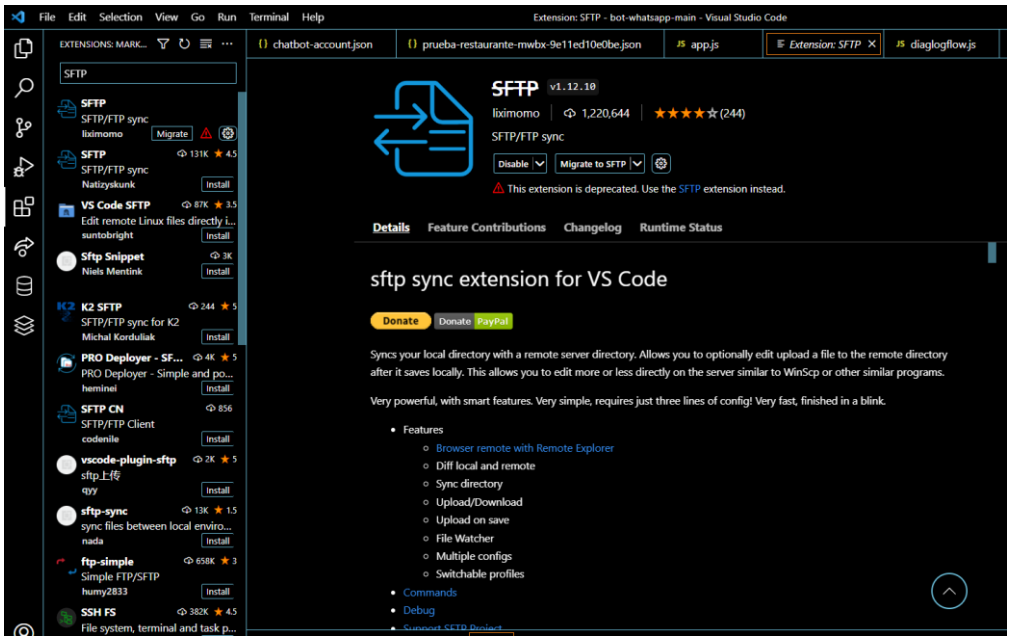
Nota. Node.js se instala automáticamente, se puede consultar la versión de npm y node que se han instalado con los comandos “npm -v” y “node -v” respectivamente.

Para cargar archivos a la instancia de VM se utiliza una extensión de VS Code que se llama SFTP y permite sincronizar un directorio local con un servidor remoto. Es posible configurarlo para cargar y actualizar archivos modificados localmente en el momento en que se

guarden, de este modo al editar un archivo del backend o el chatbot en el directorio local, se reflejarán en el servidor remoto al momento de guardar lo cambios, con respecto al frontend es un poco diferente, aunque los cambios se van subir al servidor remoto estos no se reflejarán en la página web a menos de que se reinicie el servicio.

Figura 92

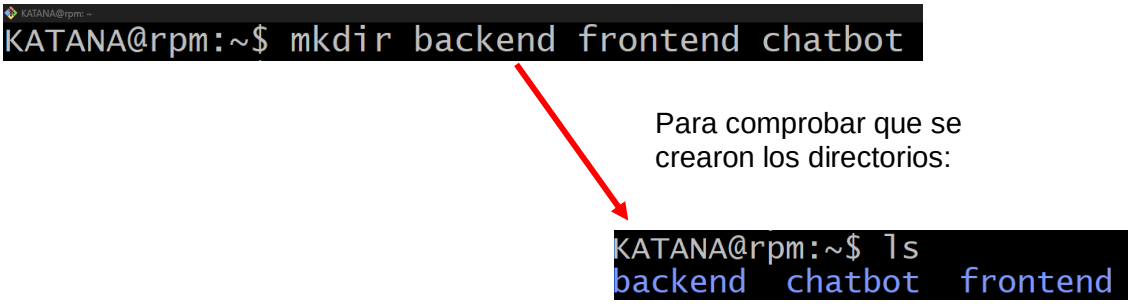
Extensión SFTP para VS Code



Previo a cargar los archivos es necesario crear los directorios en los que se va a almacenar cada proyecto.

Figura 93

Crear directorios en la instancia de VM

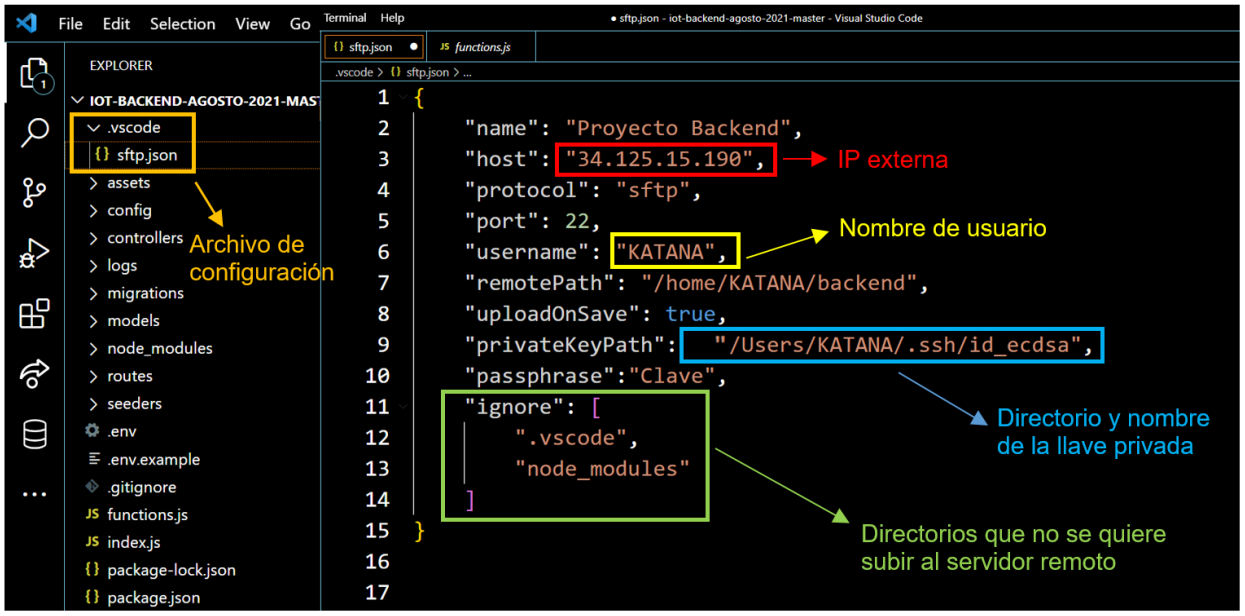


Para cargar los archivos se abre cualquiera de los proyectos en VS Code y se presiona la tecla F1 para abrir la lista de comandos, se busca el comando FSTP: Config y al clicar sobre él se crea y se abre el archivo sftp.config, luego se puede acceder al archivo de configuración mediante el directorio ".vscode" que lo contiene y que se encuentra en la raíz del proyecto, se puede subir los proyectos en cualquier orden, en este caso se empieza por el backend.

En cuanto al apartado "name", se puede colocar cualquier nombre que permita identificar el proyecto mientras que en "username" hay que colocar el nombre de usuario que se genera al configurar la clave SSH que se muestra en la Figura 87.

Figura 94

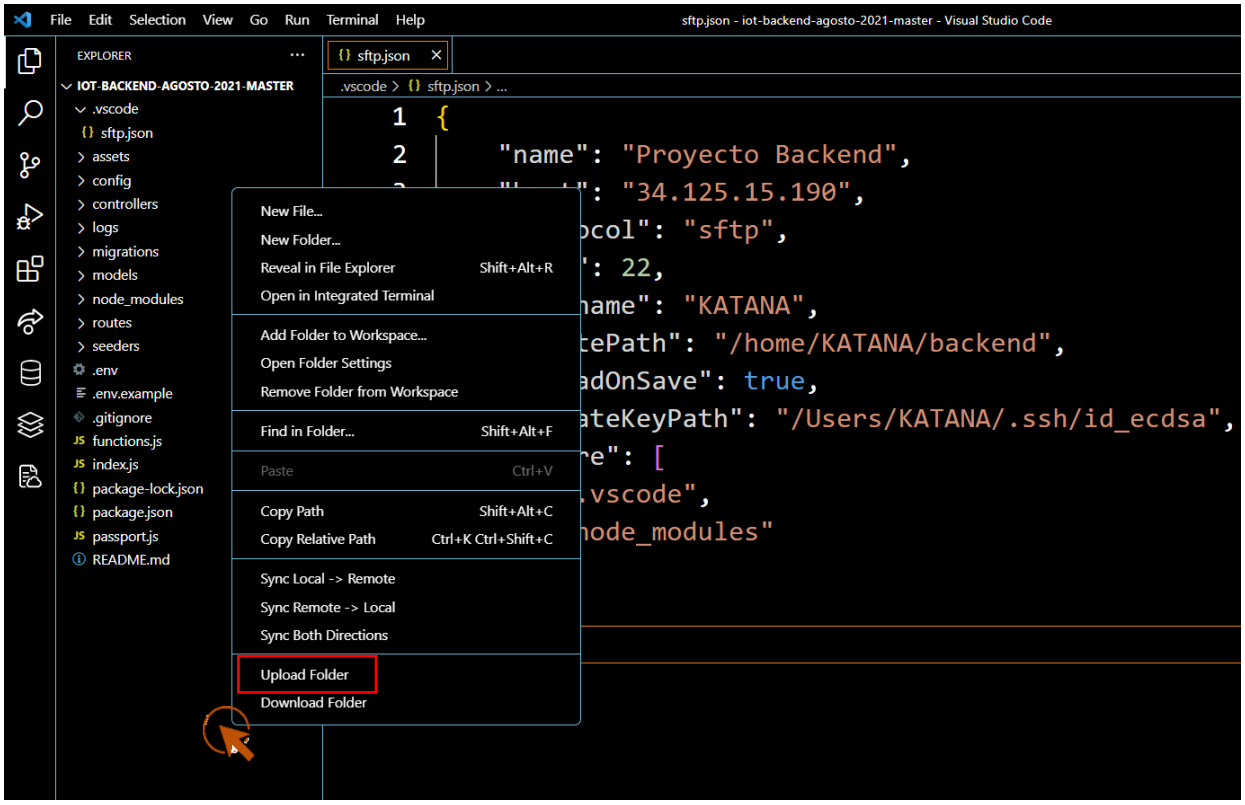
Archivo de configuración SFTP



Nota. El apartado de "passphrase" se debe llenar con la clave asignada a la llave SSH privada, si no tiene una clave asignada se borra esa sección. En el apartado "remotePath" se coloca la ruta de la instancia de VM en la que se van a subir los archivos.

Figura 95

Cargar carpetas



Nota. Presionar el botón derecho del mouse en una parte vacía del explorador de VS Code y luego seleccionar la opción Upload Folder.

Se realiza el mismo procedimiento para los otros dos proyectos editando el "remotePath" con el directorio de la instancia de VM correspondiente. Luego de cargar todas las carpetas se puede verificar el contenido de cada uno de los directorios mediante la terminal y como se encuentran sincronizados también se pueden visualizar desde VS Code.

Figura 96

Listar directorios cargados a la máquina virtual

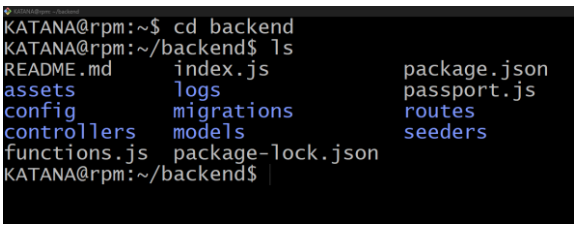
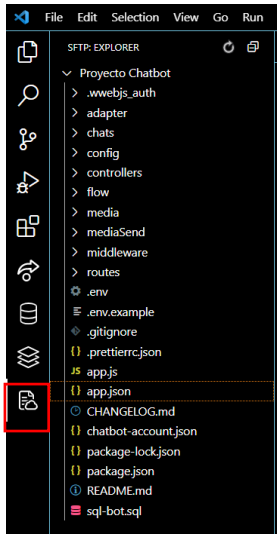


Figura 97

Explorador de archivos de SFTP

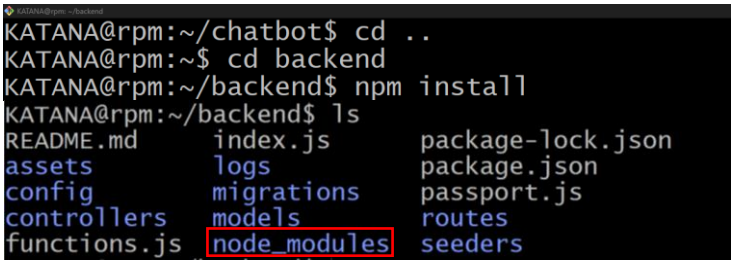


Nota. Se observa que no se cargaron los directorios ".vscode" y "node_modules" tal y como se configuró previamente.

El directorio ".vscode" no tiene ninguna utilidad para el proyecto cargado en la instancia de VM, por otro lado, el directorio "node-modules" contiene todas las librerías descargadas por medio de npm que no se cargan por que tomaría demasiado tiempo y es mucho más fácil volver a instalarlas, para ello simplemente se ejecuta el comando *npm install* en la terminal, esto crea el directorio "node_modules" e instala todas las librerías a partir del archivo "package_lock.json" en la máquina virtual, proceso que se debe repetir para los 3 proyectos.

Figura 98

Recuperar librerías en la máquina virtual

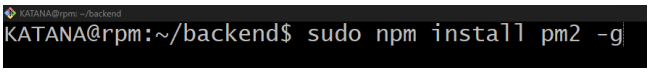


Levantar backend y chatbot

Se mencionó que existe una gran diferencia entre implementar el proyecto en un entorno de desarrollo y un entorno productivo, por este motivo se optó por instalar Nginx para levantar la página web, este servidor web no tiene la capacidad de ejecutar JavaScript y no es correcto usar npm start para levantar el backend y el chatbot, este servicio de npm se centra en el desarrollo, reinicia el servidor automáticamente cuando se guarda cualquier cambio en el proyecto para reflejarlo lo antes posible por medio de nodemon, en un entorno de producción no se esperan realizar cambios constantemente, es mejor usar un administrador de procesos como PM2 que está preparado para mantener los servicios en línea 24/7, optimizar recursos y que cuenta con herramientas que facilitan las tareas comunes de gestión del sistema con la capacidad de levantar los servicios automáticamente cuando se reinicia el servidor.

Figura 99

Instalar PM2



Nota. Se usa –g para instalar PM2 a nivel global y utilizarlo en backend y chatbot.

Figura 100

Levantar backend

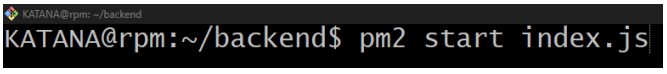
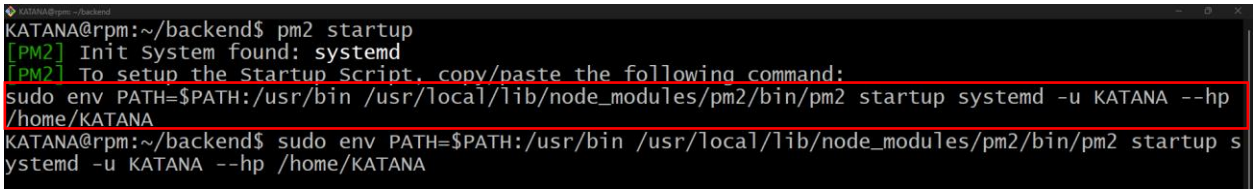


Figura 101

Levantar servicios automáticamente

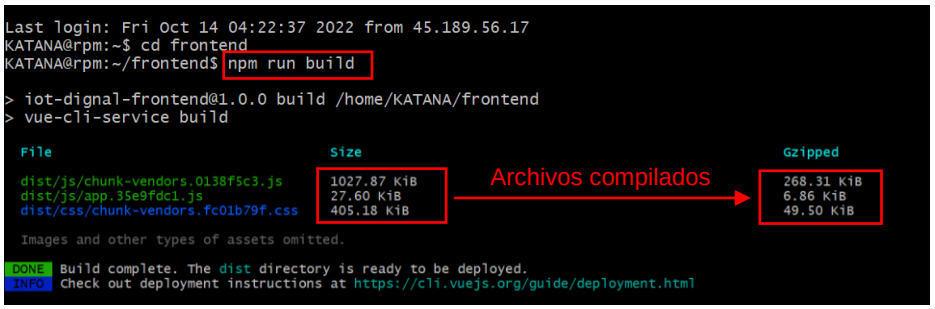


Nota. El comando *pm2 startup* genera un nuevo comando que hay que copiar y ejecutar, con esto el backend se levantará automáticamente cada vez que se reinicie el servidor.

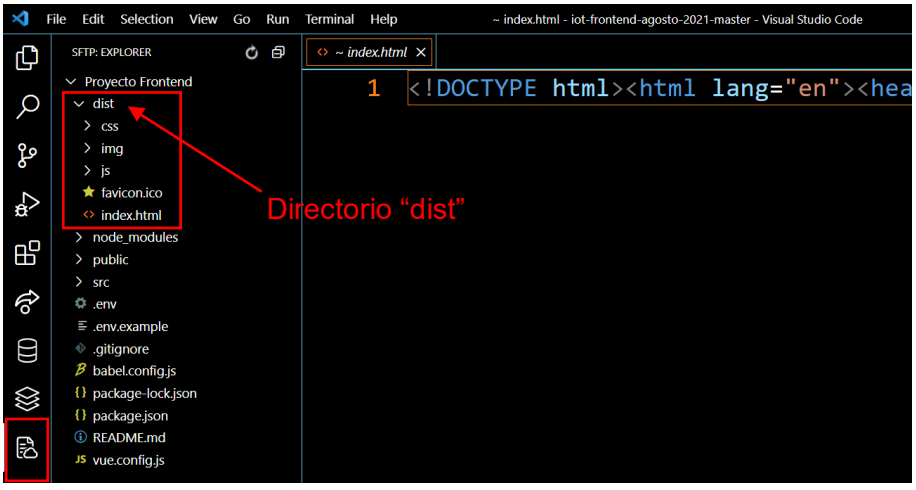
El backend ya se encuentra en línea, antes de ejecutar en frontend hay que realizar ciertas tareas de preparación para el uso de Nginx, el comando *npm run build* compila todos los archivos JavaScript verificando que la aplicación se ejecuta localmente y la empaqueta en archivos estáticos para trabajar en entornos de producción, estos archivos que han sido compilados y optimizados los guarda en el directorio “dist”, los archivos que contiene este directorio pueden ser interpretados por cualquier navegador y es lo único que requiere Nginx para poner en línea la página web.

Figura 102

Compilación de producción



a)



b)

Nota. a) Comando para ejecutar la compilación de producción para Nginx. b) Directorio “dist”. El comando debe ser ejecutado dentro de la carpeta del Frontend, los archivos compilados tienen un tamaño menor al estar optimizados.

De momento cualquier persona que intente acceder a la dirección IP externa verá el mensaje de bienvenida de Nginx, para poder visualizar la página web del proyecto hay que editar el archivo de configuración de Nginx.

Figura 103

Página de bienvenida de Nginx

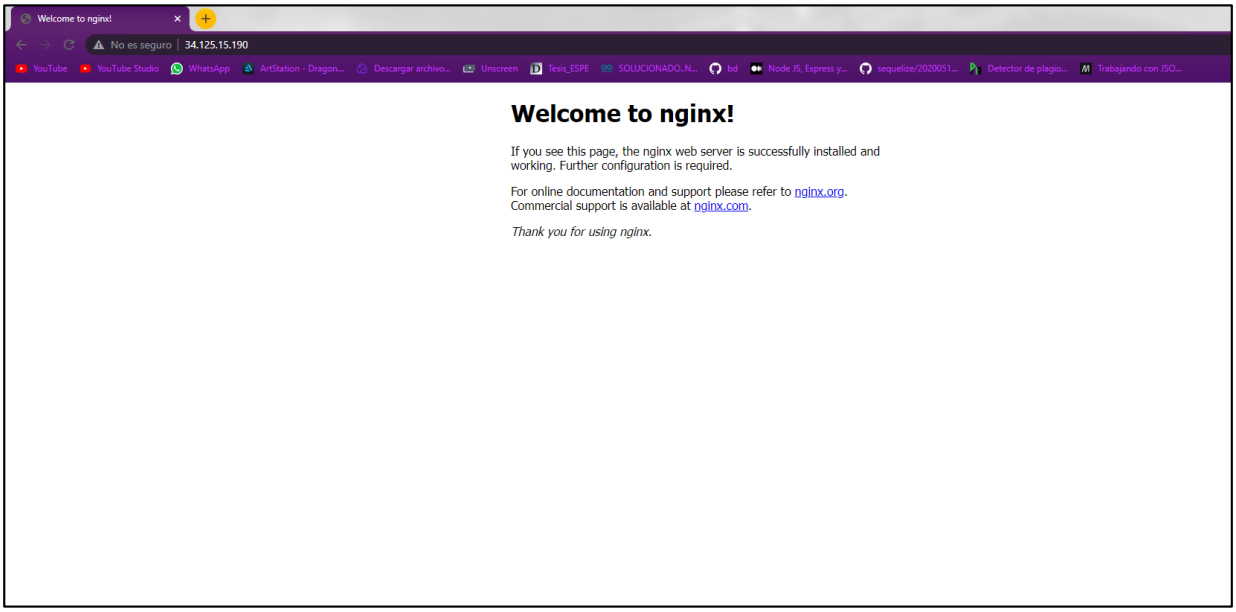
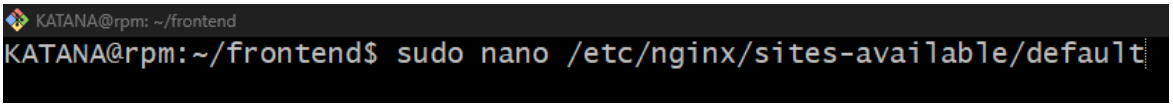


Figura 104

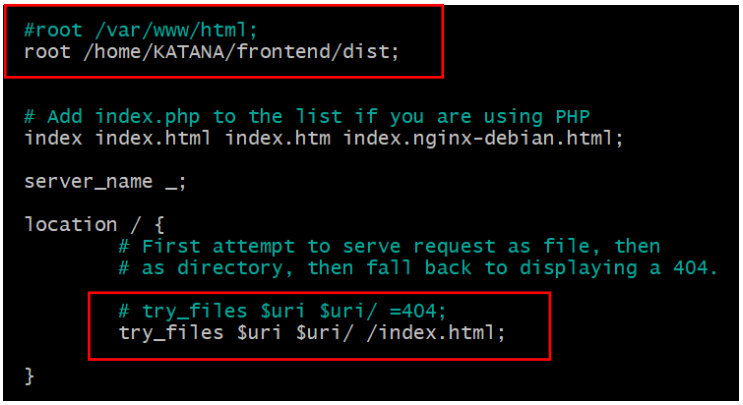
Abrir archivo de configuración de Nginx



Dentro del archivo de configuración se modifica la ruta de la página de bienvenida de nginx por la ruta que contiene la versión compilada del proyecto, es decir, el directorio “disp”. También trae configurado el mensaje que se debe mostrar en caso de un error 404, debido a la forma en que trabaja Vue.js, al estar destinado a aplicaciones de una sola página hay que modificar la configuración para que todo pase a través del archivo index.html y que sea el mismo Vue.js el que se encargue de disparar el error 404 en caso de que ocurra.

Figura 105

Archivo de configuración por defecto de Nginx



Nota. A la hora de guardar los cambios hay que sobrescribir el archivo de configuración default.

Estos cambios no se reflejan porque el servidor necesita volver a cargar las configuraciones, esto se consigue reiniciando el servicio de Nginx. Una vez que se reinicie la IP externa mostrará la página de acceso a la plataforma.

Figura 106

Reiniciar Nginx

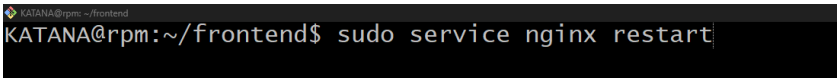
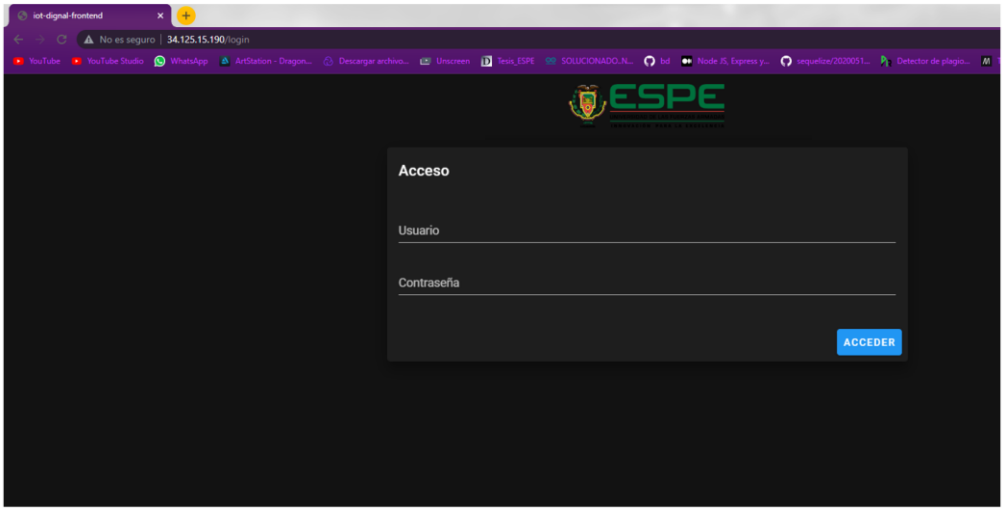


Figura 107

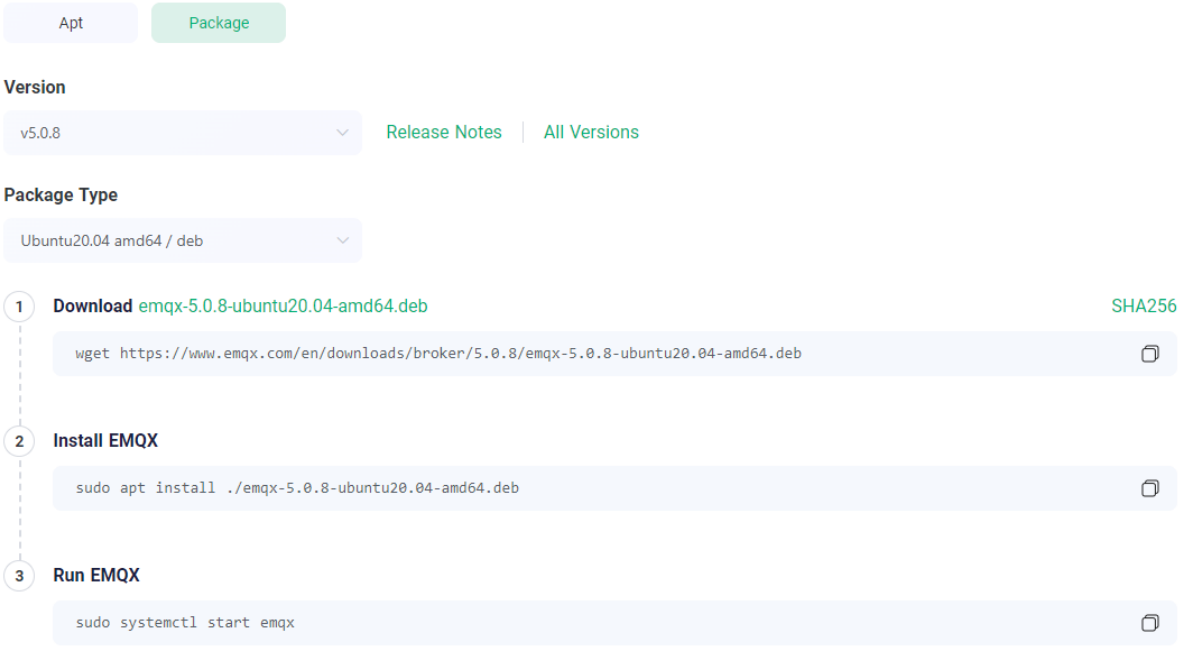
Acceso a la página web a través de la IP externa



Por el momento solo se encuentra levantado el frontend pero para que todos los servicios de la plataforma funcionen adecuadamente hay que instalar un par de aplicaciones adicionales en la máquina virtual. Para la comunicación con los sensores se necesita el servidor MQTT, el proceso de instalación para Ubuntu 20.04 se encuentra en la página oficial de EMQX. Los comandos se deben ejecutar en la ruta /home.

Figura 108

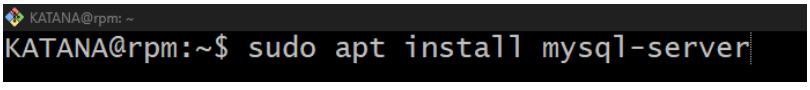
Instalar EMQX en Ubuntu 20.04



Nota. En el apartado “Package Type” hay que seleccionar amd, la otra opción es arm pero a la hora de crear la instancia no es posible elegir esa opción por problemas de compatibilidad entre la arquitectura del disco de arranque y el tipo de máquina virtual. Tomado de (EMQTech, 2022).

Figura 109

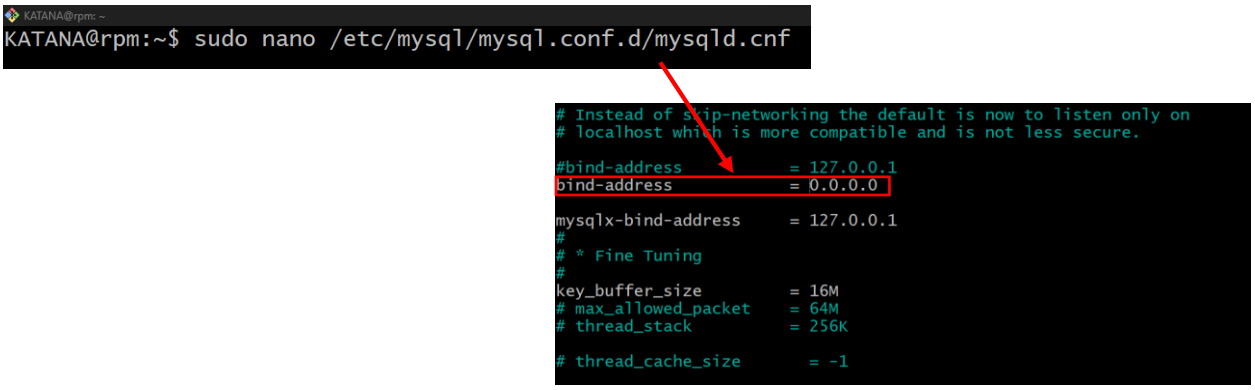
Instalar MySQL en la instancia de VM



Entre los archivos de configuración de MySQL hay que modificar el parámetro bin-address para que permita acceder a la base de datos por medio de un servidor remoto, por defecto MySQL se encuentra configurada para escuchar únicamente conexiones a nivel local, habilitar este servicio es beneficioso en casos en los que se cuenta con un equipo que trabaja con Windows mientras que el código se ejecuta sobre una VM con Linux y que además no cuenta con una interfaz gráfica de usuario.

Figura 110

Configurar MySQL para escuchar tráfico externo



Nota. La IP 0.0.0.0 es un comodín para escuchar solicitudes desde cualquier IP externa.

Luego de editar la bin-adress hay que guardar y cerrar el archivo, para que los cambios tengan efecto hay que reiniciar el servicio de MySQL.

Figura 111

Reiniciar servicio de MySQL

Al conectarse a MySQL se puede hacer una consulta de los usuarios y el host que trae por defecto, como se observa el usuario root solo escucha solicitudes a nivel de localhost.

Figura 112

Tabla de usuarios y Host default de MySQL

```
mysql> SELECT User, Host FROM mysql.user;
+------------------+-----------+
| User             | Host      |
+------------------+-----------+
| debian-sys-maint | localhost |
| mysql.infoschema | localhost |
| mysql.session    | localhost |
| mysql.sys        | localhost |
| root             | localhost |
+------------------+-----------+
5 rows in set (0.00 sec)
```

Lo primero es asignarle una contraseña nativa de MySQL al usuario root para que el mismo MySQL se encargue de la encriptación de la contraseña, la cual debe ser la misma contraseña que se asignó en el backend en el archivo config.js

Figura 113

Asignar contraseña a usuario root

```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'Clave1652666';
```

a)

b)

Nota. a) Comando para asignar contraseña al usuario root. b) Contenido archivo config.js.

Para reiniciar la tabla de usuarios se ejecuta el comando *FLUSH PRIVILEGES*, con esto se aplican los cambios y ahora para ingresar a MySQL se debe ejecutar el comando *mysql –u root –p* e ingresar la contraseña.

Luego de haber modificado la configuración de MySQL hay que modificar el host del usuario root y se hace de manera similar, mediante el comodín “%” que significa que el usuario admitirá conexiones desde cualquier IP externa. Luego de modificar la tabla de usuarios hay que ejecutar el comando *FLUSH PRIVILEGES*.

Figura 114

Modificar Host de usuario root en MySQL

```
mysql> UPDATE mysql.user SET Host='%' WHERE User='root'
```

Al tratarse de una instancia de VM de Google Cloud hay que crear las reglas de Firewall de VPC, contienen componentes que definen lo que hace la regla que es básicamente orientar tráfico según el protocolo, puertos de destino y orígenes.

Figura 115

Crear reglas de Firewall de Google Cloud

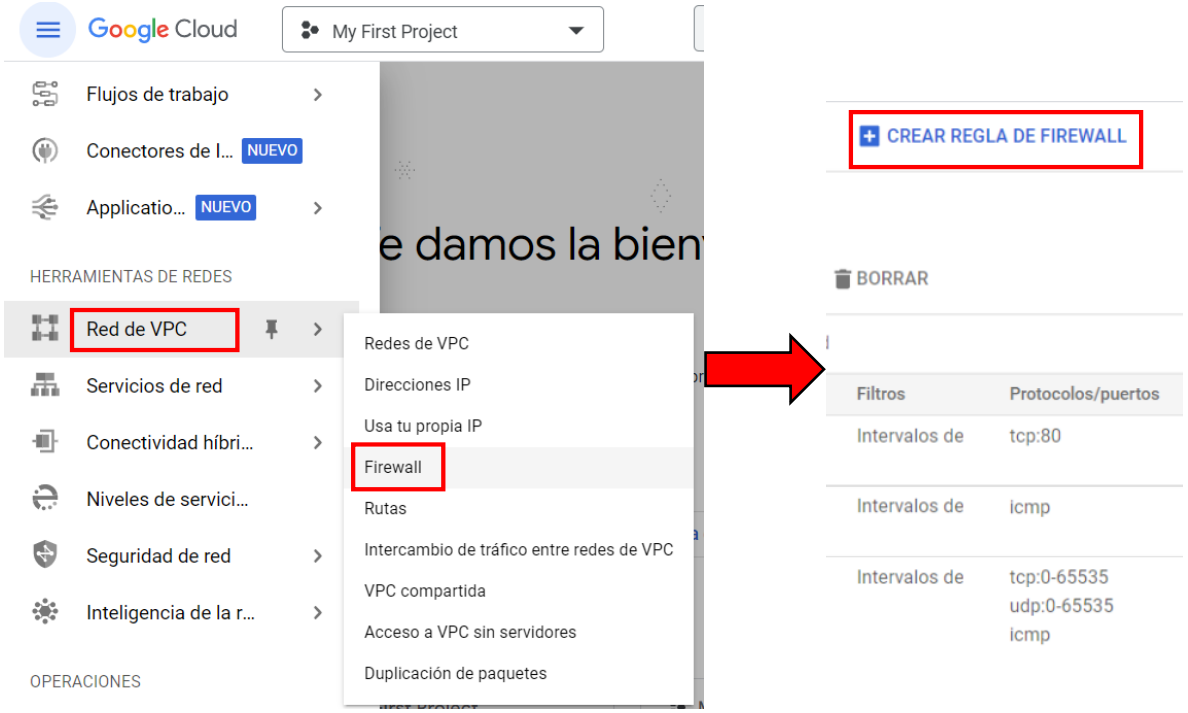


Figura 116

Apertura de puertos para MySQL

Nota. No confundir el puerto 3389 que permite conectarse a instancias mediante escritorio remoto con el puerto 3306 que se usa para conexiones de servidor a base de datos MySQL.

La apertura del puerto 3306 da acceso VS Code para que se conecte a MySQL de forma remota por medio de una extensión para administrar bases de datos.

Figura 117

Extensión MySQL VS Code para administrar bases de datos

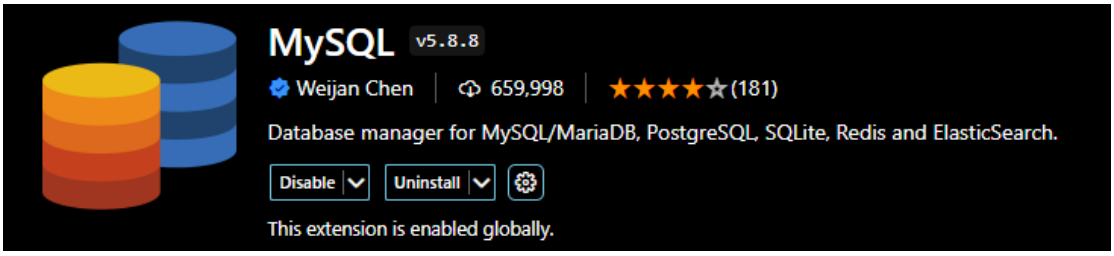
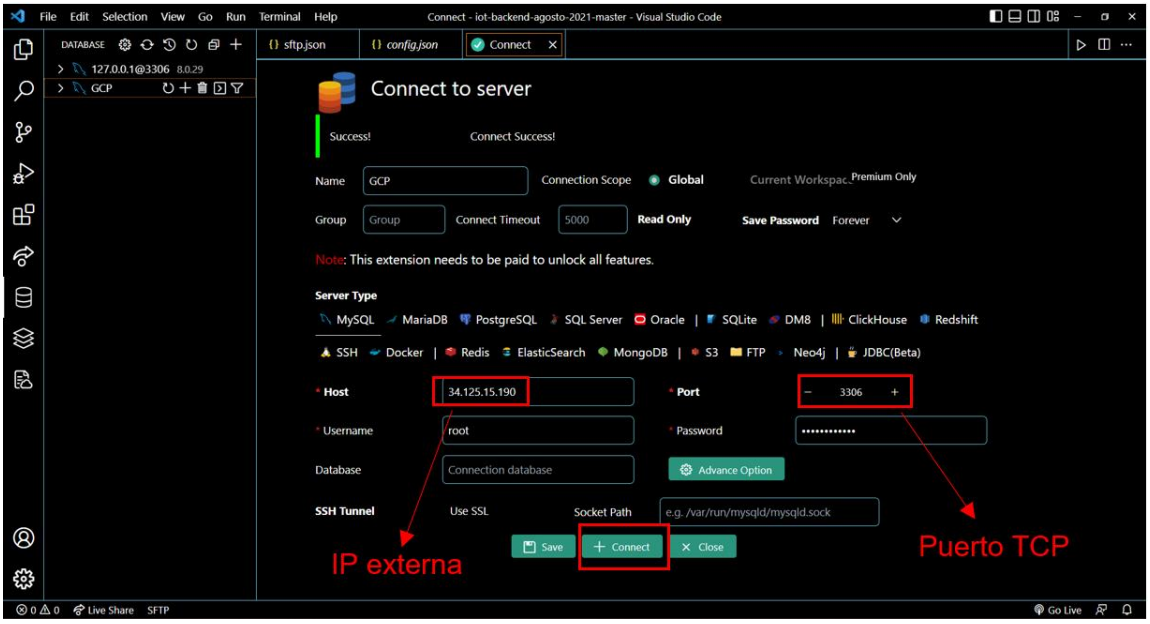


Figura 118

Configuración del administrador de base de datos



Hay que crear la base de datos y migrar las tablas con sequelize.

Figura 119

Crear base de datos en el entorno de producción

```
KATANA@rpm:~$ cd backend
KATANA@rpm:~/backend$ npx sequelize-cli db:create

Sequelize CLI [Node: 10.19.0, CLI: 6.2.0, ORM: 6.6.5]

Loaded configuration file "config/config.json".
Using environment "development".
Database iot created.
```

Figura 120

Migrar tablas en el entorno de producción

```
KATANA@rpm:~/backend$ npx sequelize-cli db:migrate

Sequelize CLI [Node: 10.19.0, CLI: 6.2.0, ORM: 6.6.5]

Loaded configuration file "config/config.json".
Using environment "development".
== 20210816004505-create-user: migrating =======
== 20210816004505-create-user: migrated (0.259s)

== 20210821203607-create-devices: migrating =======
== 20210821203607-create-devices: migrated (0.243s)

== 20210821210708-create-devices-data: migrating =======
== 20210821210708-create-devices-data: migrated (0.253s)

== 20220622005200-numerotelefono: migrating =======
== 20220622005200-numerotelefono: migrated (0.241s)
```

Figura 121

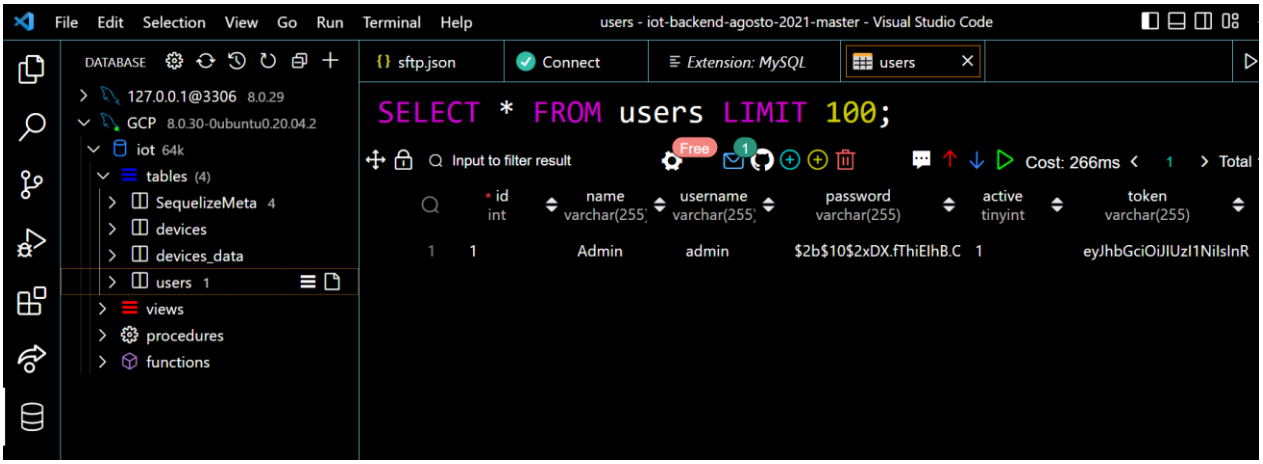
Crear usuario demo administrador

```
KATANA@rpm:~/backend$ npx sequelize-cli db:seed:all

Sequelize CLI [Node: 10.19.0, CLI: 6.2.0, ORM: 6.6.5]

Loaded configuration file "config/config.json".
Using environment "development".
== 20210821213526-first-user: migrating =======
== 20210821213526-first-user: migrated (1.724s)
```

La ejecución de los comandos para crear la base de datos, migrar las tablas y crear el usuario demo se puede corroborar desde VS Code.

Figura 122

Acceso remoto a tabla users por medio de VS Code



Por medio del uso compartido de recursos entre dominios o CORS un servidor permite o rechaza automáticamente ciertas solicitudes dependiendo de su origen, en este caso el origen corresponde a la IP externa, no hace falta especificar el puerto, por defecto se encuentra habilitado el puerto 80 en Google Cloud para el destino correspondiente al http-server como se muestra en la Figura 115.

Figura 123

Configurar CORS

Volviendo al tema de habilitar puertos, además del puerto para MySQL hay que habilitar 3 puertos más que corresponden al puerto 1883 para MQTT, el puerto 9874 para la API y el puerto 6872 para Websocket, el puerto 1883 viene configurado por defecto en el broker MQTT mientras que los otros dos se configuran en el archivo de variables de ambiente.

Figura 124

Variables de ambiente

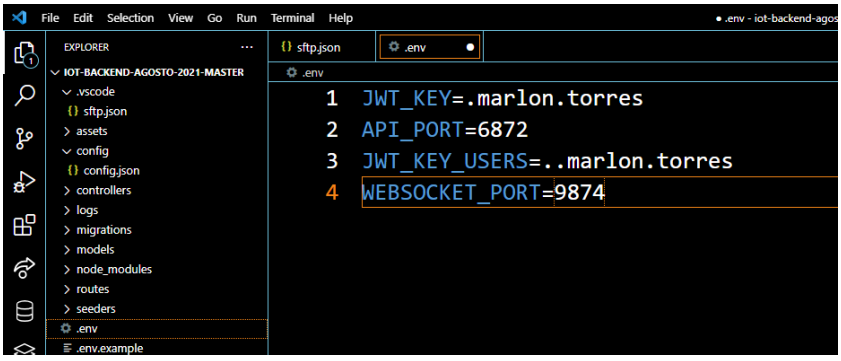


Figura 125

Abrir puerto MQTT

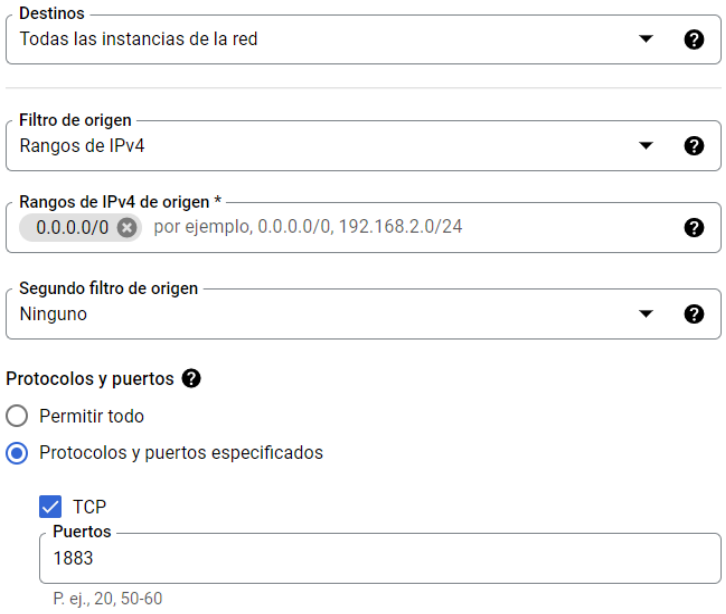


Figura 126

Abrir puerto API

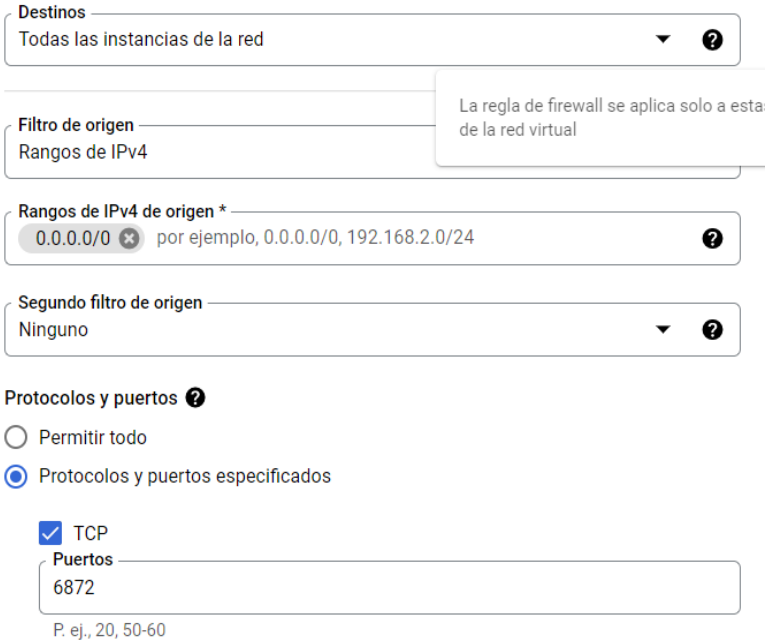


Figura 127

Abrir puerto WebSocket

Figura 128

Reglas de Firewall de VPC

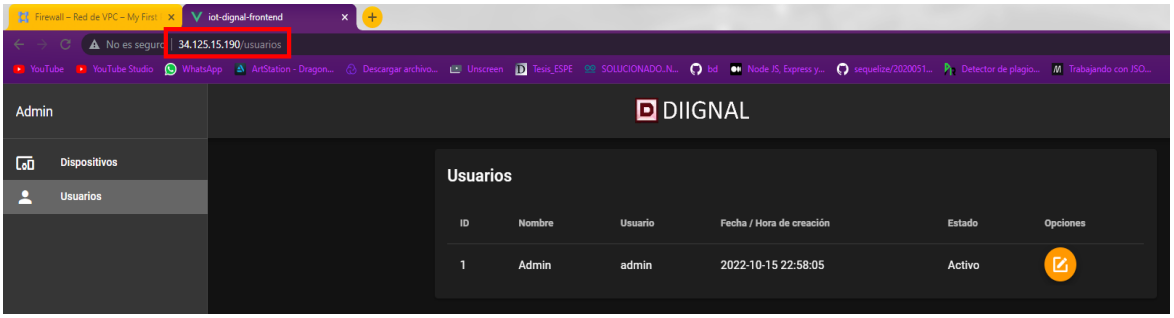
Figura 129

Reemplazar localhost por la IP externa en el archivo Chart.vue

El desarrollo de la plataforma hasta este punto corresponde a lo mostrado en la Figura 76, a excepción de la parte de los sensores que al igual que en el desarrollo a nivel local durante la implementación en el entorno productivo también pueden ser simulados.

Figura 130

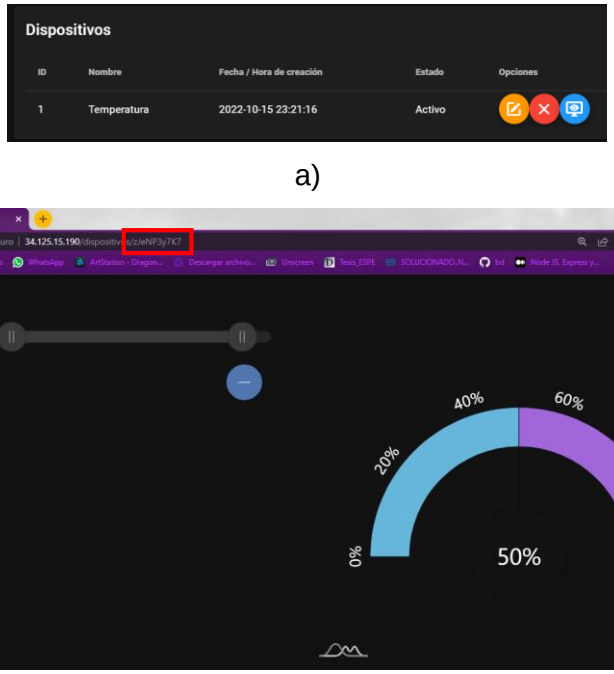
Página web montada exitosamente en entorno productivo



Antes de configurar MQTTX para simular los sensores hay que crear un nuevo dispositivo en la plataforma y abrir la gráfica para copiar el tópico que se le ha asignado.

Figura 131

Dispositivo de prueba para el Administrador



a)

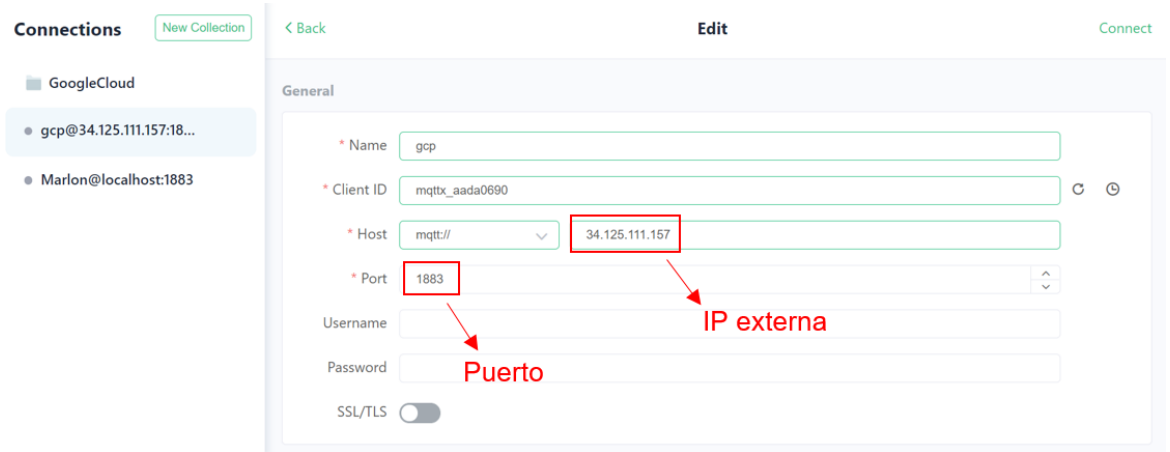
b)

Nota. a) Crear dispositivo Temperatura. b) Gráfica del dispositivo Temperatura. La ruta de la gráfica se asigna por medio del tópico que le corresponde a dicho dispositivo.

En MQTT presionar sobre el botón New Collection y asignar un nombre, para realizar pruebas solo se configura el HOST con la IP externa y se verifica que apunte al puerto 1883.

Figura 132

Configurar MQTTX para pruebas sobre IP pública



El envío de datos es exactamente igual a como se realiza en el entorno de desarrollo, lo único que hay que cambiar es el tópico del dispositivo de prueba.

Figura 133

Simulación de sensores en el entorno de producción

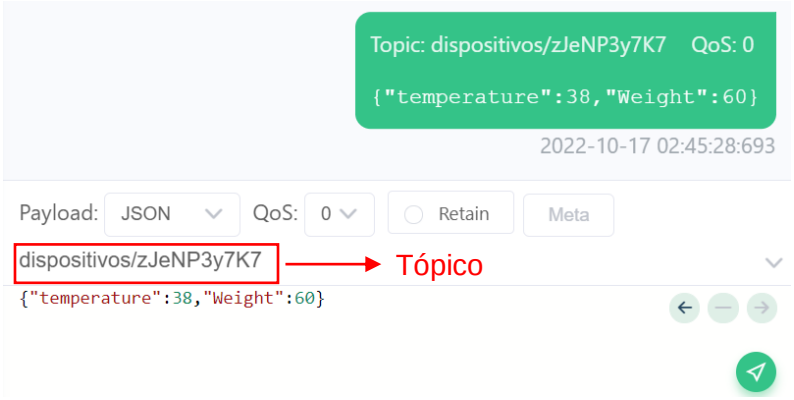
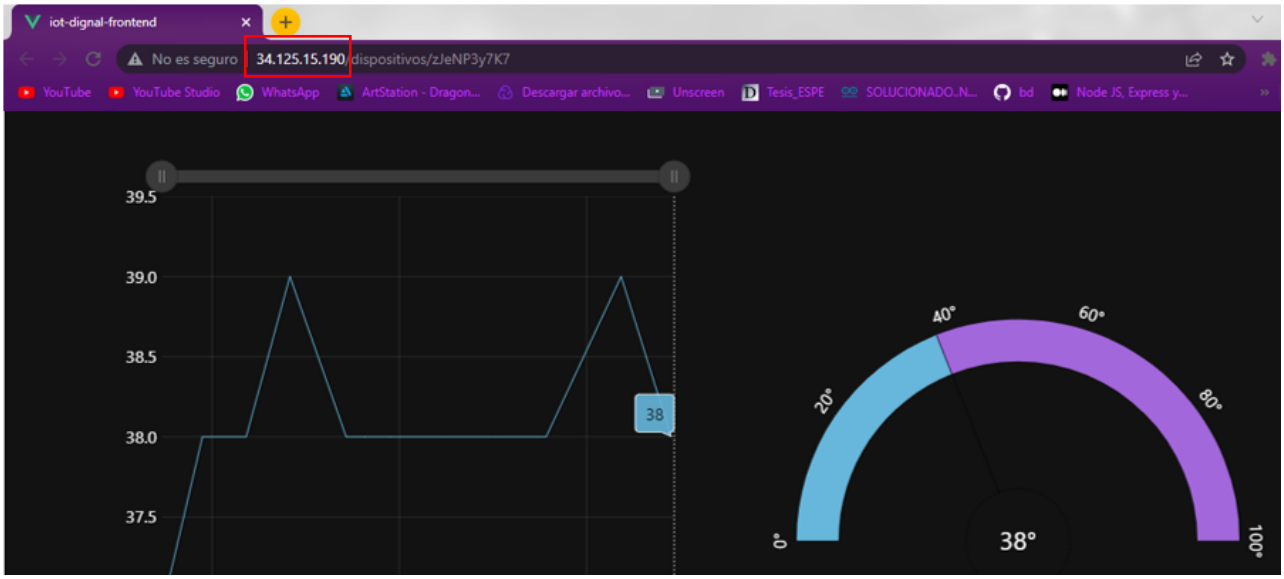


Figura 134

Gráficas en tiempo real sobre instancia de VM de sensores simulados



Elaboración de dispositivos para monitorización remota

Se elaboran cinco dispositivos típicos de un sistema de RPM que se ajustan a las necesidades de la población en base al registro estadístico de defunciones del país por lo que el sistema se centra en el monitoreo de enfermedades crónicas.

Todos los sensores se elaboran a partir de la placa de desarrollo ESP 32 que se programa con el IDE de Arduino. Cada dispositivo se encarga de obtener parámetros biométricos específicos y sus algoritmos son diferentes, principalmente para facilitar su uso, lo que no cambia es el envío de datos del dispositivo al broker MQTT, posteriormente el broker envía los datos al backend y este a su vez los envía a la base de datos y genera las gráficas.

El envío de datos se hace en tres pasos, primero cualquiera de los dispositivos al encenderse intenta conectarse a la red WiFi mediante el nombre y contraseña de la red. Para que esto sea posible se debe incluir la librería “WiFi.h”, se instala automáticamente cuando se instala el complemento de ESP 32 para Arduino.

Figura 135

Conectar ESP 32 a la red WiFi

Luego de que se establece la conexión a la red WiFi, se usa la librería “PubSubClient.h” para generar un cliente MQTT a través del cual el dispositivo se conecte al broker EMQX bajo la arquitectura Pub/Sub.

Figura 136

Crear cliente MQTT

```
5 WiFiClient esp32Client;
6 PubSubClient mqttClient(esp32Client);
```

Para que el cliente MQTT se pueda conectar al broker hay que definir la dirección y el puerto, la dirección corresponde a la IP externa con la que se ha venido trabajando. Además, se define una variable a la que se asignarán los mensajes MQTT entrantes, los mensajes se reciben serializados en formato JSON.

Figura 137

Definir dirección y puerto del Broker MQTT

```
11 char *server = "34.125.15.190";
12 int port = 1883;
13
14 //En esta variable se almacena mensaje MQTT entrante
15 String resultS = "";
```

En el void setup() el paso 1 es llamar a la función WifiInit() que inicia la conexión a la red WiFi como se muestra en la Figura 135, en el paso 2 se configura el broker MQTT y en el paso 3 se establece la función callback.

Figura 138

Funciones del void setup()

```
80 void setup(){
81     Serial.begin(115200);
82     while (!Serial) continue;
83     delay(10);
84
85  1  wifiInit();
86
87  2  mqttClient.setServer(server, port);
88
89  3  mqttClient.setCallback(callback);
90 }
```

Un callback es una función que se ejecuta después de un evento predeterminado, en este caso cuando se reciba un mensaje desde el tópico al que el dispositivo se encuentre suscrito.

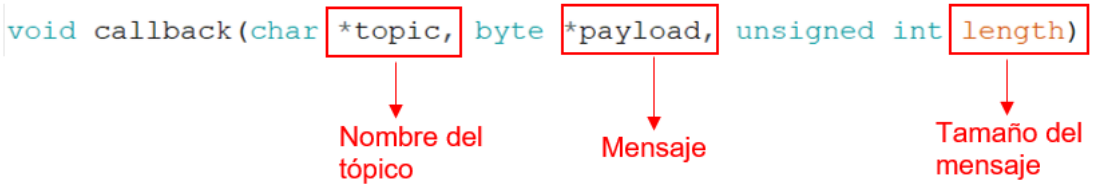
Figura 139

Función callback

```
void callback(char *topic, byte *payload, unsigned int length) {

  char payload_string[length + 1];

  int resultI;

//Convertir los mensajes a enteros o flotantes
  memcpy(payload_string, payload, length);
  payload_string[length] = '\0';
  resultI = atoi(payload_string);
  var = resultI;
  Serial.println(resultI);

//Limpiar variable resultS
  resultS = "";

//Convertir los mensajes a tipo String
  for (int i=0;i<length;i++) {
    resultS= resultS + (char)payload[i];
  }
  Serial.println(resultS);
}
```

Figura 140

Parámetros que recibe la función callback



Los mensajes se reciben en formato JSON, para poder trabajar más cómodamente en el IDE de Arduino hay que convertirlos en objetos este proceso se conoce como deserialización, para deserializar los datos como enteros o flotantes se define un arreglo del tamaño del mensaje más 1, esto para tener suficiente espacio.

Figura 141

Definir arreglo para almacenar el mensaje

```
char payload_string[length + 1];
```

Con la función nemcpy se indica en donde se van a guardar los datos recibidos (payload_string), de donde obtener el mensaje (payload) y el tamaño del mensaje (length). Luego se vacía el arreglo y se convierte el mensaje a enteros o flotantes con la función atoi y se guardan en la variable resultI para posteriormente imprimirla por consola.

Figura 142

Deserialización de JSON a números enteros o flotantes

```
//Convertir los mensajes a enteros o flotantes
  memcpy(payload_string, payload, length);
  payload_string[length] = '\0';
  resultI = atoi(payload_string);
  var = resultI;
  Serial.println(resultI);
```

Para deserializar los datos a tipo String lo primero es limpiar la variable resultS para que no se produzcan errores al leer el dato serializado ya que esta variable irá guardando la cadena de caracteres tal y como se muestra en la Figura 143.

Figura 143

Deserialización de JSON a String

```
//Limpiar variable resultS
  resultS = "";

//Convertir los mensajes a tipo String
  for (int i=0;i<length;i++) {
    resultS= resultS + (char)payload[i];
  }
  Serial.println(resultS);
}
```

La función void reconnect () revisa si el cliente se encuentra conectado al broker, si no lo está intenta conectarse cada 5 segundos, el dispositivo se suscribe al tópico una vez que el

cliente logra conectarse al broker, estar suscrito a un tópico permite emitir alertas o señales de control.

Figura 144

Función reconnect

```
void reconnect() {
  while (!mqttClient.connected()) {
    Serial.print("Intentando conectarse MQTT...");

//Se conecta al broker:
    if (mqttClient.connect("arduinoClient")) {
      Serial.println("Conectado");

      //Una vez que se conecte se suscribe al dispositivo
      mqttClient.subscribe("dispositivos/zJeNP3y7K7/led");

    } else {
      Serial.print("Fallo, rc=");
      Serial.print(mqttClient.state());
      Serial.println(" intentar de nuevo en 5 segundos");
      // Wait 5 seconds before retrying
      delay(5000);
    }
  }
}
```

En el caso contrario, cuando se quiere enviar un mensaje al broker primero se debe serializar un objeto en una cadena JSON, esto es más sencillo ya que se usa la librería ArduinoJson.h, en su versión 6 por lo que no hay mucho que explicar, en la Figura 145 se toma un valor entero “temperatura” y se publica en el tópico correspondiente a las gráficas del dispositivo.

Figura 145

Serialización con AruinoJSON.h

```
StaticJsonDocument<256> doc;

//Preparar el dato
doc["temperature"] = 10;

char out[128];
//Serializar el dato
int b =serializeJson(doc, out);
Serial.println(out);
```

Figura 146

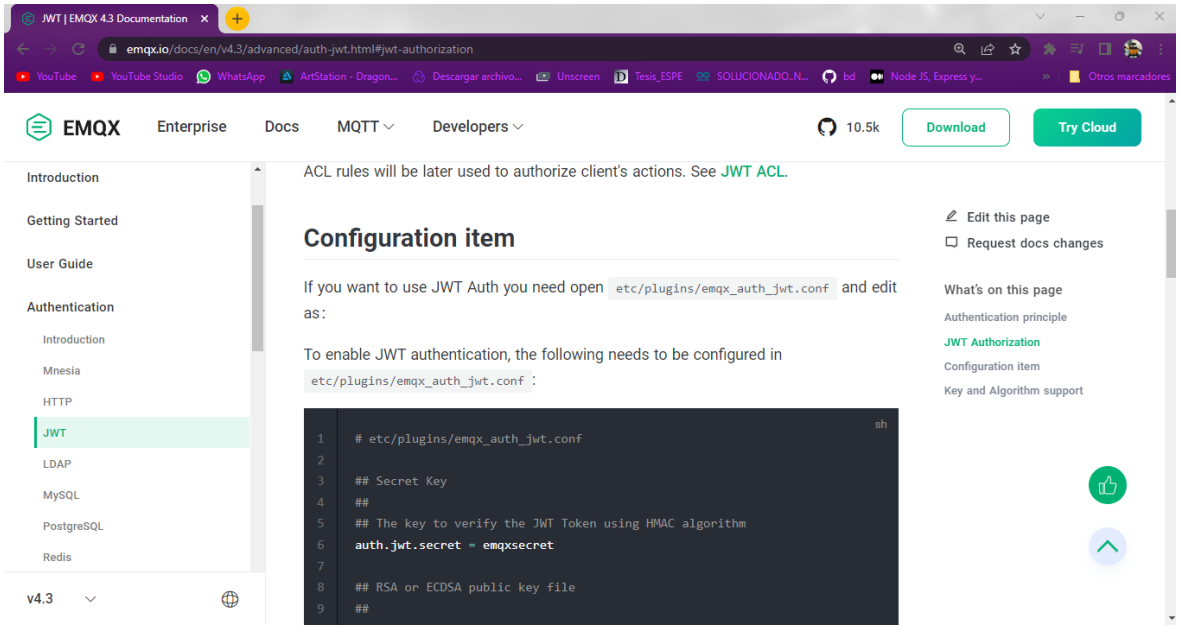
Publicar cadena JSON

```
//Publicar la cadena en el tópico correspondiente
boolean rc = mqttClient.publish("dispositivos/zJeNP3y7K7", out);
```

Cualquier dispositivo puede conectarse al broker únicamente con la IP externa y el tópico lo que puede representar un problema de seguridad del sistema, para evitar esto se instala y configura el complemento de EMQX para autenticación por JWT como se indica en su página oficial.

Figura 147

Autenticación con JWT en EMQX



Nota. La documentación cambia dependiendo de la versión de EMQX que se seleccione, es importante revisar la documentación de versiones anteriores que es más detallada. Tomado de (EMQX, 2020).

Ahora para que el dispositivo se pueda conectar además de la dirección y el tópico se deben incluir el usuario y la clave, estos parámetros se entregan a función mqttClient.connect() que se encarga de conectar la ESP 32 con el broker.

Figura 148

Parámetros para conexión con broker mediante autenticación JWT

```
char *server = "34.125.15.190";
int port = 1883;
const char *mqtt_username = "emqx";
const char *mqtt_password = ".!marlon.torres";
```

Figura 149

Función para conexión con broker mediante autenticación JWT

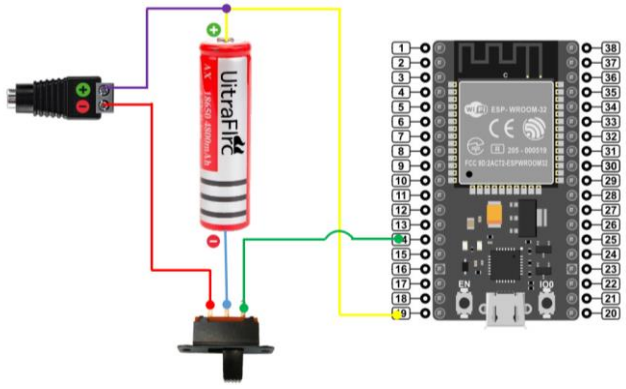
```
while (!mqttClient.connected()) {
  String client_id = String(WiFi.macAddress());

    if (mqttClient.connect(client_id.c_str(), mqtt_username, mqtt_password)) {
        Serial.println("Connected to Public MQTT Broker");
    } else {
        Serial.print("Failed to connect with MQTT Broker");
        Serial.print(mqttClient.state());
        delay(2000);
    }
}
```

Alimentación del circuito

Todos los dispositivos se alimentan por medio de una batería recargable 18650 de 3.7V y 4800mAh, como las baterías no se pueden cargar mientras se encuentran conectadas al circuito se usa un interruptor conmutador que las desacopla del circuito, además, mientras la batería se encuentra desacoplada se puede alimentar la ESP 32 por su puerto micro USB.

Figura 150

ESP 32 alimentado por batería recargable



Báscula

Para la báscula se usa el módulo HX711 y cuatro celdas de carga, contiene el circuito de almacenamiento y un convertidor analógico/digital compatible con una variedad de microcontroladores como Arduino y la ESP 32, utiliza comunicación serial de dos pines (clock y data), se usan para obtención de datos, configurar la ganancia, reinicio y apagado, los pines se pueden conectar a cualquier pin digital del microcontrolador y es similar a I2C.

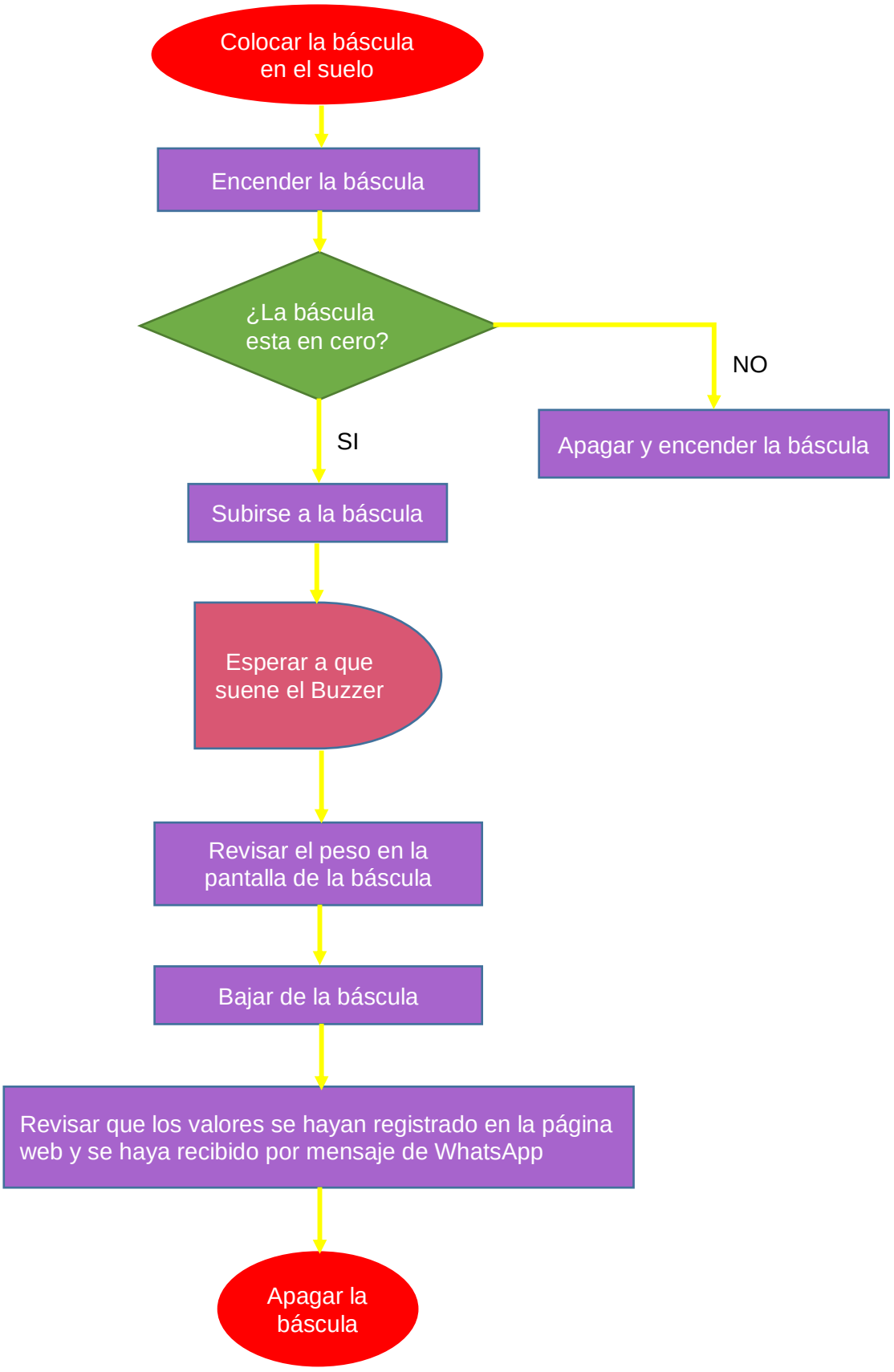
Tabla 11

Especificaciones técnicas módulo HX711

Especificaciones técnicas HX711	
Voltaje de alimentación	2.7 a 5.5 V
Consumo de corriente	< 10 mA
Resolución ADC	24 bits
Frecuencia de lectura	80 Hz
Dimensiones	38*21*10 mm

Para los dispositivos en general es fundamental garantizar que los datos se suban a la nube y que sean fáciles de usar, para esto se incluye un indicador sonoro que se activa en el momento en que los datos han sido enviados al broker, también se ha hecho una investigación previa de los procedimientos que hay que seguir para obtener lecturas correctas, por ejemplo, en el caso de la báscula hay que colocar la báscula en un lugar firme, parase en el centro con la espalda recta y la vista al frente, todas estas consideraciones han sido tomadas en cuenta para el diseño de cada dispositivo, a continuación se muestra el diagrama de flujo con los pasos a seguir para el uso de una báscula adaptado a una báscula con IoT que se ajuste a las características de la plataforma web.

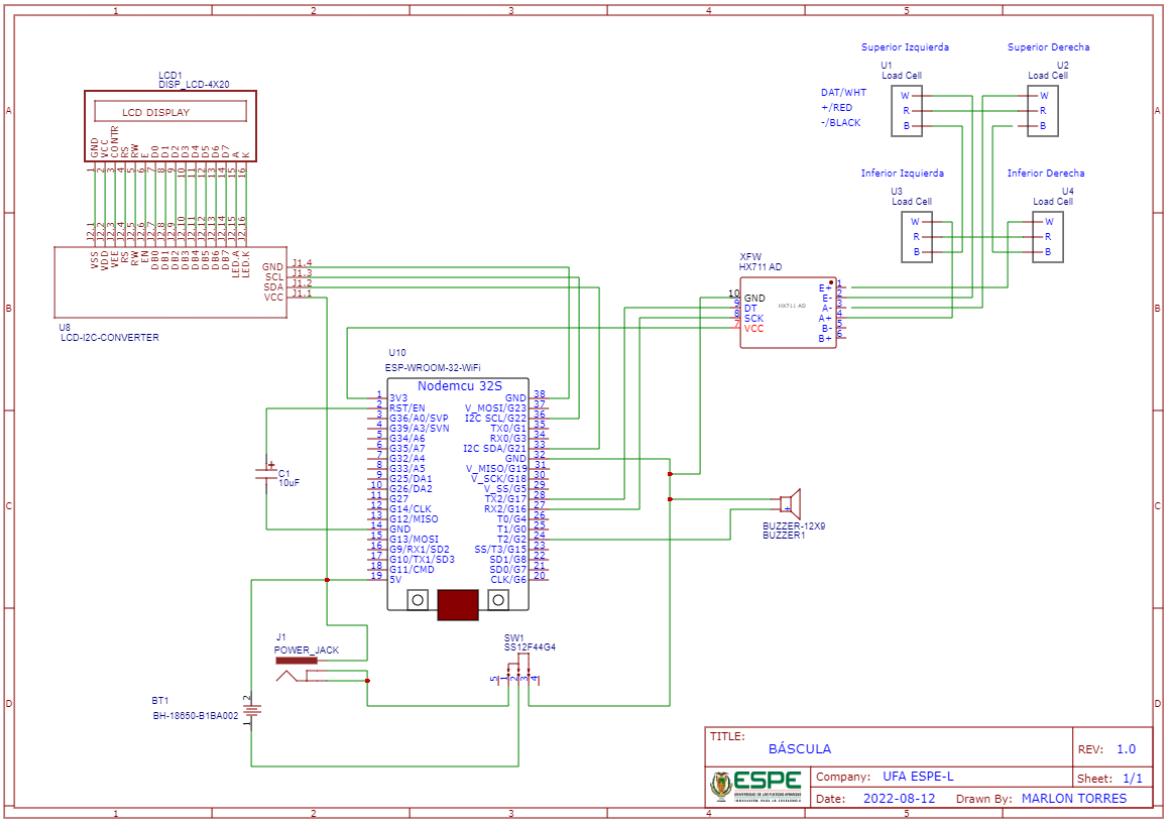
Diagrama flujo para el uso de la báscula



Las básculas comerciales más comunes para uso personal soportan un peso máximo de 180 Kg, la báscula se diseña para soportar un peso de 200 Kg, ambos valores superan por mucho el peso de una persona promedio, esta elección es sencilla ya que se encuentra condicionada por las celdas de carga disponibles en el mercado y la compatibilidad con el módulo HX711, además, no influye significativamente en el costo del dispositivo.

Figura 151

Circuito electrónico de la báscula



El circuito con todos los componentes facilita el diseño de la carcasa, se trata de una estructura plástica impresa en 3D, no es necesario ni útil un análisis estructural al no ser posible tomar en cuenta aspectos propios de la impresión 3D y de la manufactura aditiva en general como el relleno, la retícula, el número de capas, etc. Además, la estructura va a estar sometida únicamente a cargas alternadas de compresión, se cuenta con ensayos mecánicos de elementos impresos en 3D que fácilmente resisten cargas de tensión perpendiculares a las

capas de más de 250 Kg con rellenos del 30%, y con paredes de apenas dos líneas, por lo que el mayor problema de la estructura es la presión, para evitar que la estructura falle al aplicar demasiada presión en algún punto de la base o la tapa se añade un patrón cúbico en el interior de la base de la báscula.

Figura 152

Tapa y base de báscula

Las celdas de carga se ubican en las patas de la báscula, como el peso se va a concentrar en las cuatro patas se imprimen en ABS que tiene mejores características mecánicas y en condiciones similares resiste como mínimo 50 Kg más que el PLA en cargas de tensión como se ha registrado en varios ensayos de empresas dedicadas a vender impresoras y rollos de filamento.

Figura 153

Pata báscula

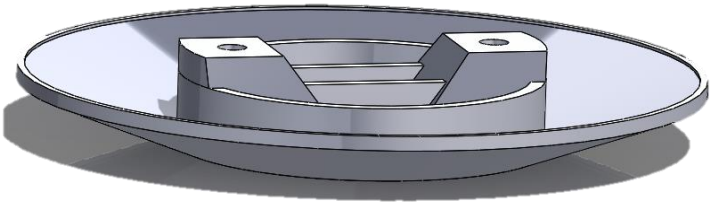
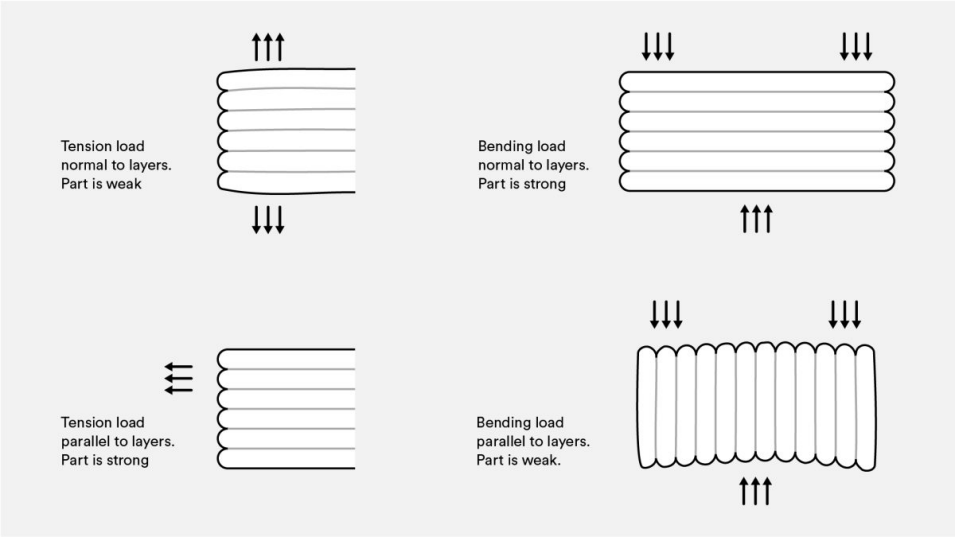


Figura 154

Orientación de construcción

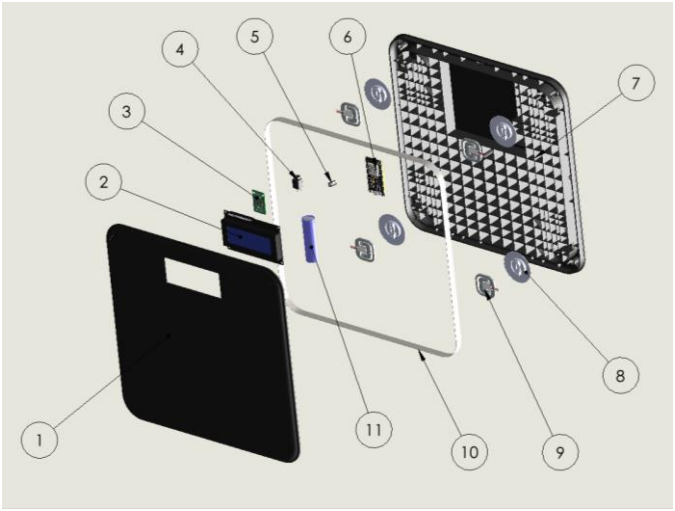


Nota. De naturaleza anisotrópica, son más débiles en una dirección debido a la orientación de las capas Tomado de (VanHome, 2021).

Por la forma de la tapa y la base se pueden imprimir con un relleno del 100% sin gastar mucho material extra, se añade una lámina metálica para que ayude a mantener la integridad de la báscula, todos los elementos se muestran y enlistan en la Figura 155.

Figura 155

Vista explosionada y lista de materiales báscula

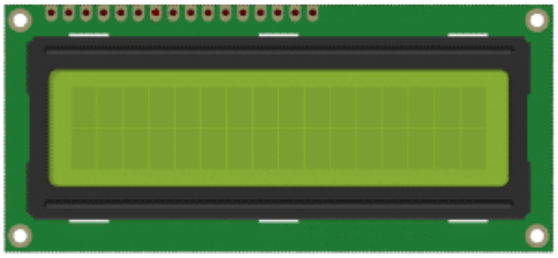


N.º DE ELEMENTO	N.º DE PIEZA	CANTIDAD
1	Tapa	1
2	LCD 20x4	1
3	Driver HX711	1
4	Jack 3.5 mm	1
5	Capacitor electrolítico	1
6	ESP 32	1
7	Cuerpo	1
8	Pata de soporte	4
9	Celda de carga	4
10	Lámina metálica	1
11	Batería 18650	1

Generar caracteres especiales para la pantalla LCD de la báscula. Las pantallas LCD 16x2 y 20x4 cuentan con 16 y 20 caracteres por fila respectivamente, cada carácter se compone de una matriz de píxeles de 5 columnas y 8 filas.

Figura 156

Pantalla LCD 16X2



Para que los usuarios puedan ver su peso en la báscula sin tener que inclinarse hacia la pantalla o bajarse de la báscula para acercar la pantalla a su rostro se generan caracteres especiales por medio de su representación en forma de matriz.

Figura 157

Representación en forma de matriz del carácter de una pantalla LCD

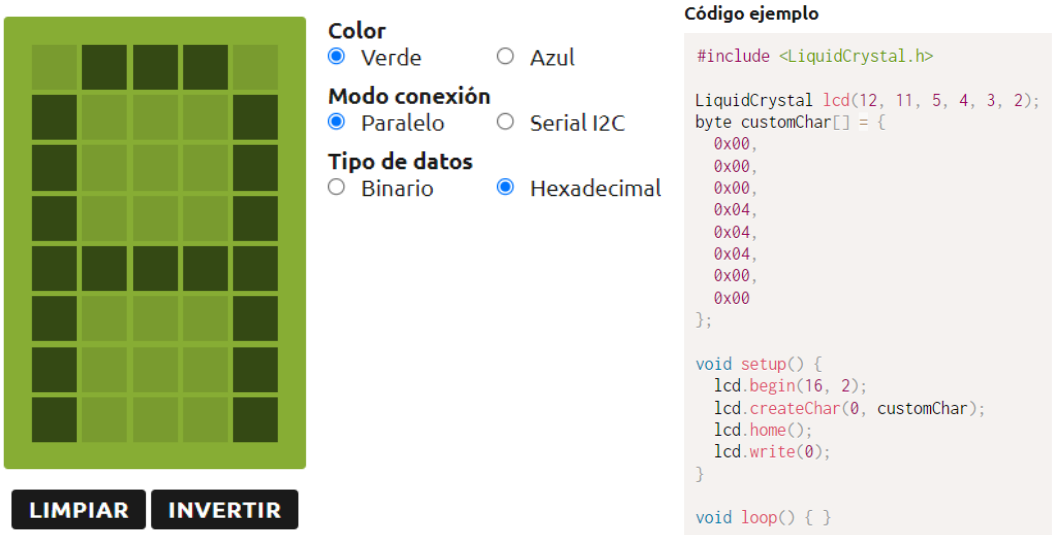
	d7	d6	d5	d4	d3	d2	d1	d0
byte0	-	-	-	0	0	1	0	0
byte1	-	-	-	0	1	0	1	0
byte2	-	-	-	1	0	0	0	1
byte3	-	-	-	1	0	0	0	1
byte4	-	-	-	1	0	0	0	1
byte5	-	-	-	1	1	1	1	1
byte6	-	-	-	1	0	0	0	1
byte7	-	-	-	0	0	0	0	0

Nota. Las pantallas LCD capaces de mostrar textos, números y símbolos se denominan alfanuméricas. Tomado de (Aladuino, 2021).

Los caracteres personalizados se crean con una página web, solo se marcan los pixeles que se quieren activar y se escogen el color de la pantalla, el modo de conexión y el tipo de datos, el resultado es un código de ejemplo que se puede usar en el IDE de Arduino.

Figura 158

Generador de caracteres especiales para LCD alfanuméricas



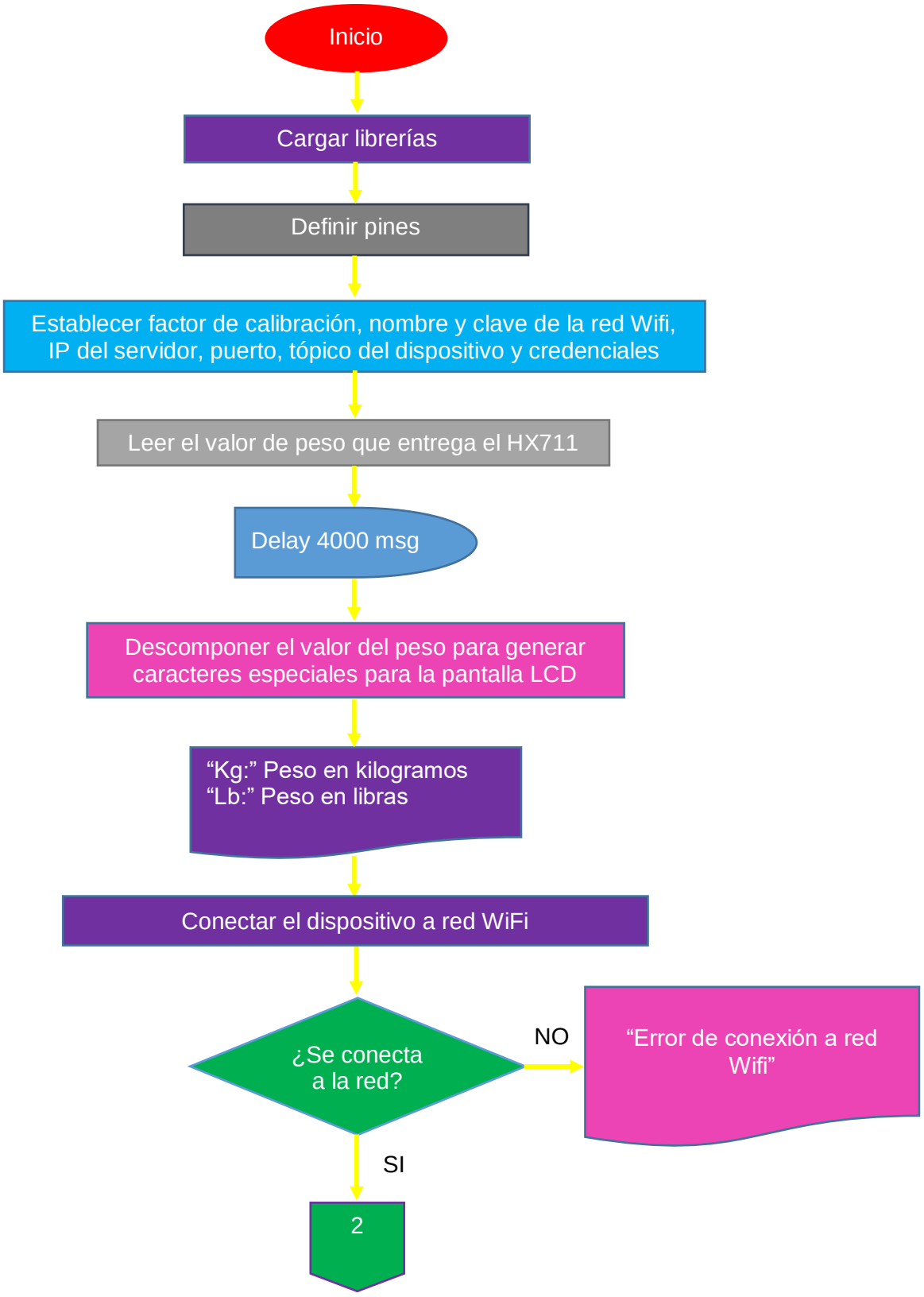
Nota. El código de ejemplo se genera para una LCD 16x2, para el proyecto se usa el módulo adaptador para I2C. Tomado de (Aladuino, 2021).

Para representar un número se necesita un total de cinco caracteres especiales, se forman con tres filas y dos columnas de la pantalla LCD, la báscula da el valor de peso en kilogramos y en libras por medio de tres números enteros y dos decimales, cada número se descompone y se imprime uno por uno en una determinada posición de la pantalla LCD por medio de un switch case.

Figura 159

Ejemplo de números formados a partir de caracteres especiales

Diagrama de flujo de la programación de la báscula



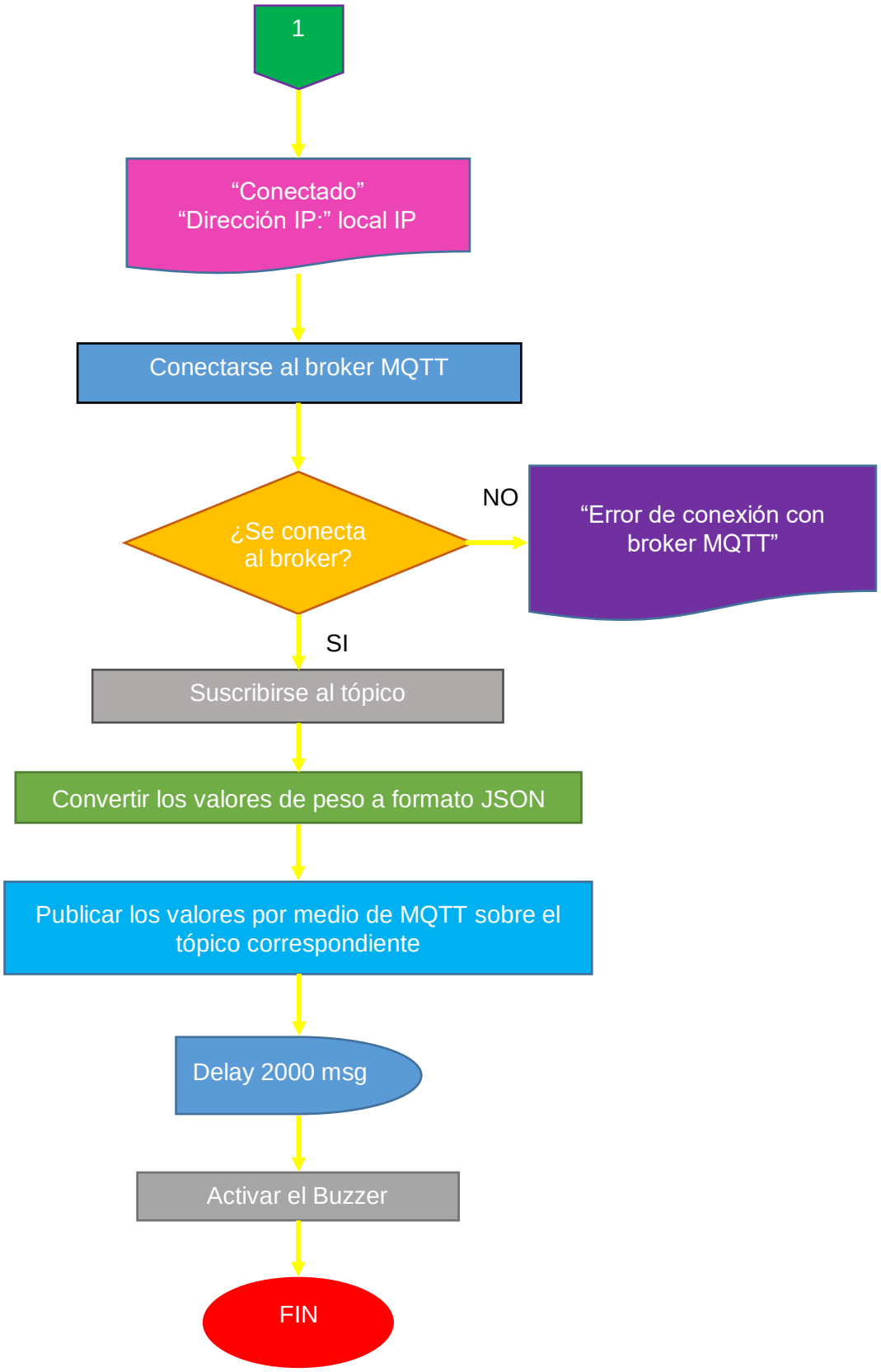

1
“Conectado”
“Dirección IP:” local IP
Conectarse al broker MQTT
¿Se conecta al broker?
NO
“Error de conexión con broker MQTT”
SI
Suscribirse al tópico
Convertir los valores de peso a formato JSON
Publicar los valores por medio de MQTT sobre el tópico correspondiente
Delay 2000 msg
Activar el Buzzer
FIN

Figura 160

Resultado de la elaboración de la báscula

Figura 161

Calibración de la báscula

Figura 162

Presentación de valor de peso con caracteres especiales

Termómetro infrarrojo

Debido a la pandemia el uso de los termómetros infrarrojos se ha popularizado, comúnmente dispositivos muy sofisticados se usan para medir superficies a las que no se puede acceder con facilidad, pero en la actualidad dispositivos económicos se usan de forma muy recurrente en todo tipo de establecimientos desde centros comerciales hasta hospitales.

Su principio de funcionamiento se basa en que todos los objetos irradian energía, la temperatura del objeto y la intensidad de energía radiante son proporcionales, el cuerpo humano emite principalmente radiación infrarroja, para obtener la temperatura de una persona el sensor de temperatura infrarrojo mide con precisión la débil energía de radiación infrarroja liberada por su cuerpo.

Para la elaboración del termómetro se usa el sensor de temperatura MLX90614ESF DCC, forma parte de la familia de sensores MLX90614 que traen integrados una termopila sensible a IR y un circuito acondicionador de señal ASSP en el encapsulado TO-39.

Figura 163

MLX90614

Al sensor se le añaden lentes ópticos para incrementar el rango en el que pueden medir la temperatura, dependiendo del tipo y tamaño del lente sube el precio del sensor, por lo tanto el sensor más económico es el que no posee ningún tipo de lente, como el que se muestra en

la Figura 163, también se integra sobre placas con un regulador de voltaje, protecciones o filtros económicos, no agregan mucho valor al sensor en comparación con el lente óptico.

En la familia de los MLX90614, los dispositivos se diferencian mediante códigos con los que se identifica su rango de temperatura, encapsulado y el código de opciones que consta de tres dígitos que hacen referencia al voltaje de alimentación, número de termopilas y opciones de paquete.

Figura 164

Codificación de sensores de la familia MLX90614

Part No.	Temperature Code	Package Code	- Option Code - X X X	Standard part	Packing form
MLX90614	E (-40 °C...85°C) K (-40°C...125°C)	SF (TO-39)	(1) (2) (3)	-000	- T U

(1) Supply Voltage/ Accuracy	(2) Number of thermopiles:	(3) Package options:
A - 5V	A – single zone	A – Standard package
B - 3V	B – dual zone	B – Reserved
C - Reserved	C – gradient compensated*	C – 35° FOV
D - 3V medical accuracy		D/E – Reserved
		F – 10° FOV
		G – Reserved
		H – 12° FOV (refractive lens)
		I – 5° FOV

Nota. El tipo de lente óptico no se especifica en ninguna parte de la codificación del sensor de temperatura. Tomado de (Melexis, 2019).

Para la elaboración del termómetro infrarrojo se selecciona el modelo MLX90614 ESF DCC con lente óptico DCC3863110 y placa con regulador de voltaje integrado, la diferencia de precio en comparación con el modelo sin lente óptico es mínima y escoger sensores con lentes ópticos de mayor tamaño no es viable ya que cuestan aproximadamente el triple.

Figura 165

Ejemplos de modelos de sensores de la familia MLX90614

Tabla 12

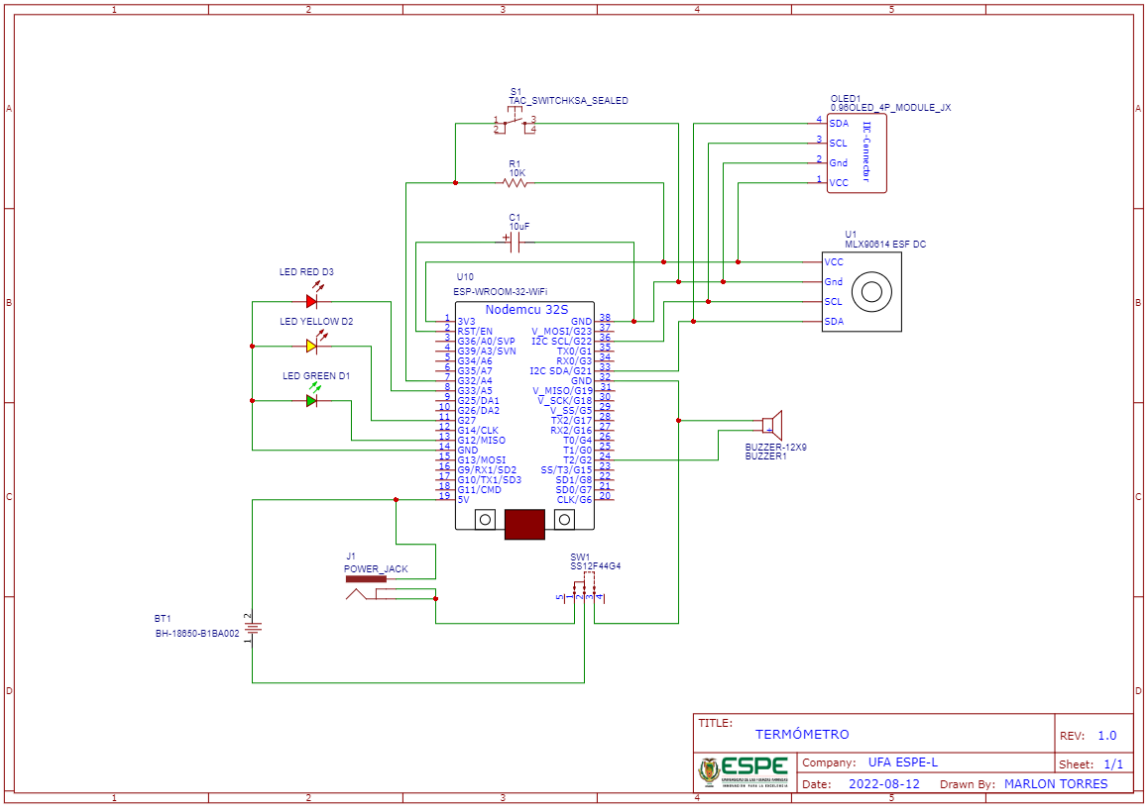
Especificaciones técnicas MLX90614 ESF DCC

Especificaciones técnicas MLX90614 ESF DCC	
Familia	MLX90614
Rango temperatura	E = (- 40°C…85 °C)
Encapsulado	SF = TO-39
Alimentación/precisión	D = 3V D = Precisión médica
Número de termopilas	C = Gradiente compensado
Opciones de encapsulado	C = - 35 °C FOV
Resolución	0.14 °C

Los valores se presentan por medio de una pantalla oled SSD 1306 que permite mantener un diseño compacto para el termómetro, además del circuito de alimentación se incluyen 3 leds de distintos colores que facilitan la interpretación de los valores medidos y un botón pulsador ubicado de tal forma que sea cómodo para los usuarios medirse la temperatura a sí mismos o a otra persona.

Figura 166

Circuito electrónico del sensor de temperatura infrarrojo



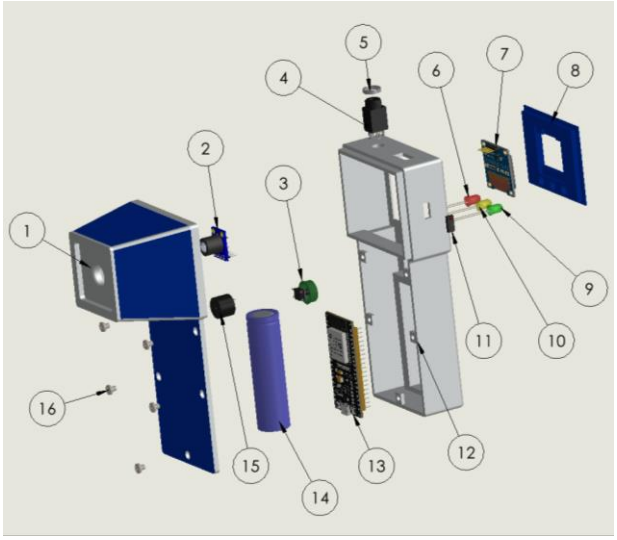
El termómetro consta de tres partes, la tapa que contiene únicamente el sensor infrarrojo, un soporte para la pantalla oled y una base que contiene el resto de componentes cuya disposición se muestra en la vista explosionada. El modelo no debe interferir con el campo de visión del sensor o la distancia al objeto que se desea medir, ambos parámetros afectan directamente a la precisión del dispositivo, el sensor promedia la temperatura de todos los objetos que se encuentren dentro de su campo de visión y cualquier obstáculo puede afectar los valores de temperatura, especialmente en modelos con un amplio ángulo de visión.

Figura 167

Modelo 3D termómetro infrarrojo

Figura 168

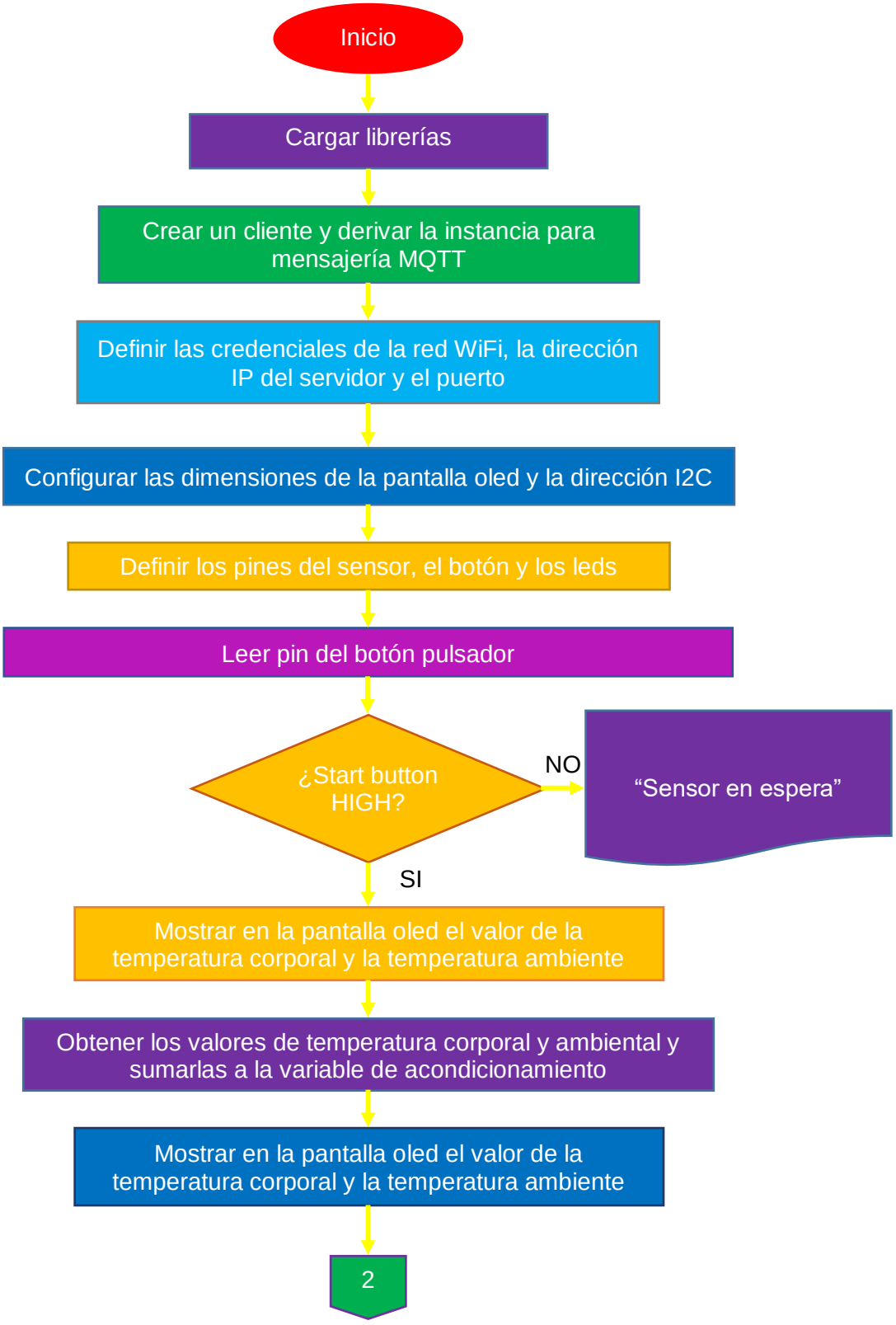
Vista explosionada y lista de materiales termómetro infrarrojo



N.º DE ELEMENTO	N.º DE PIEZA	CANTIDAD
1	Tapa	1
2	MLX90614 ESF DCC	1
3	Push Button	1
4	Jack 3.5 mm	1
5	Tuerca Jack 3.5 mm	1
6	Led rojo	1
7	SSD 1306	1
8	Soporte SSD 1306	1
9	Led verde	1
10	Led amarillo	1
11	Conmutador deslizante	1
12	Base	1
13	ESP 32	1
14	Batería 18650	1
15	Buzzer	1
16	Tornillo rosca métrica	5

Para la programación del dispositivo se usa la librería compatible con todos los sensores de temperatura infrarrojo de Melexis MLX90614.h, el sensor de temperatura y la pantalla oled se conectan mediante I2C por lo que basta con especificar la dirección de la pantalla a la 0X3C, el sensor tiene una dirección I2C predeterminada 0x5A.

Diagrama de flujo de la programación del termómetro infrarrojo



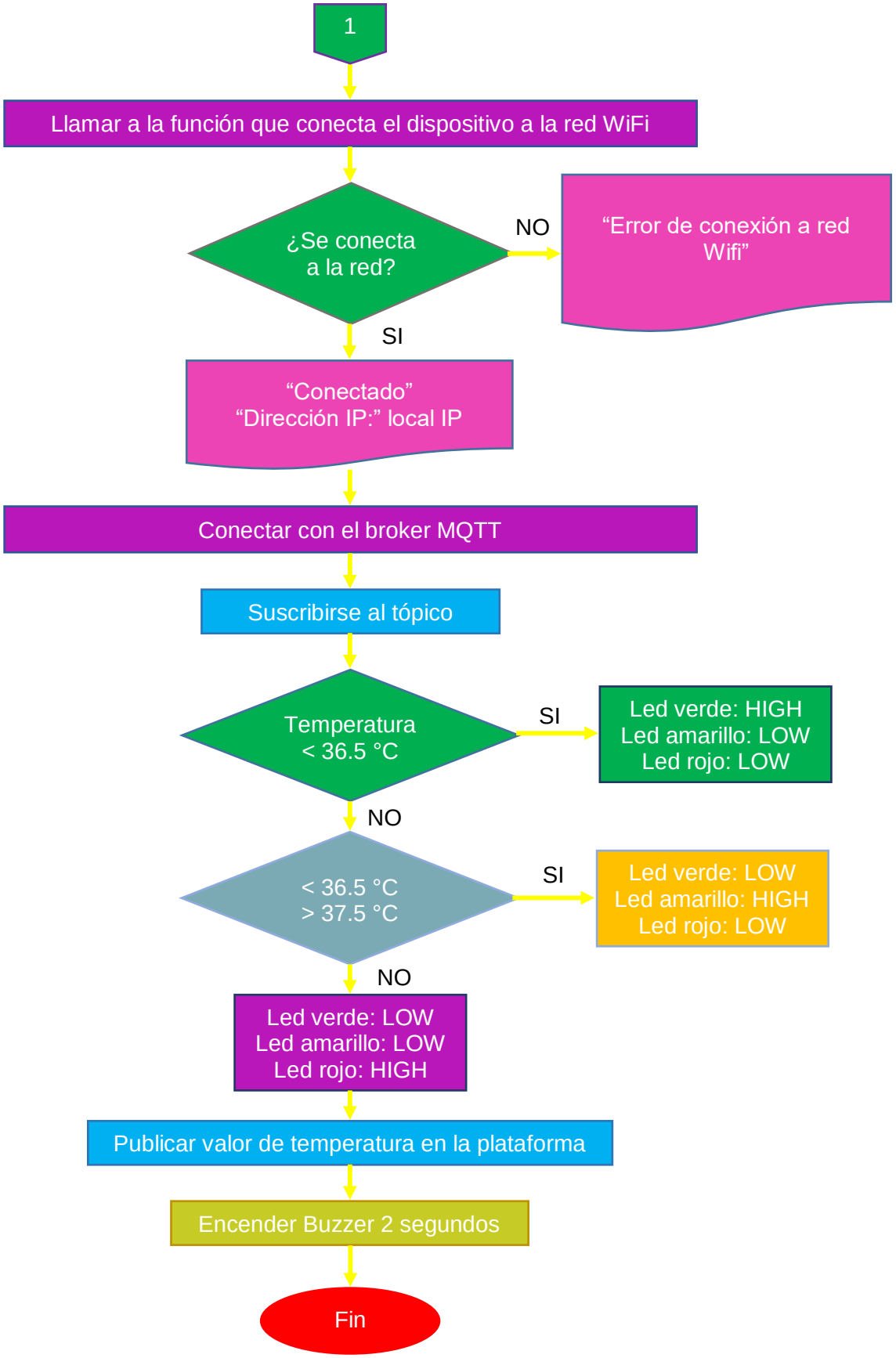

1
Llamar a la función que conecta el dispositivo a la red WiFi
¿Se conecta a la red?
NO
“Error de conexión a red Wifi”
SI
“Conectado”
“Dirección IP:” local IP
Conectar con el broker MQTT
Suscribirse al tópico
Temperatura < 36.5 °C
SI
Led verde: HIGH
Led amarillo: LOW
Led rojo: LOW
NO
< 36.5 °C
> 37.5 °C
SI
Led verde: LOW
Led amarillo: HIGH
Led rojo: LOW
NO
Led verde: LOW
Led amarillo: LOW
Led rojo: HIGH
Publicar valor de temperatura en la plataforma
Encender Buzzer 2 segundos
Fin

Figura 169

Impresión de modelo 3D de termómetro infrarrojo

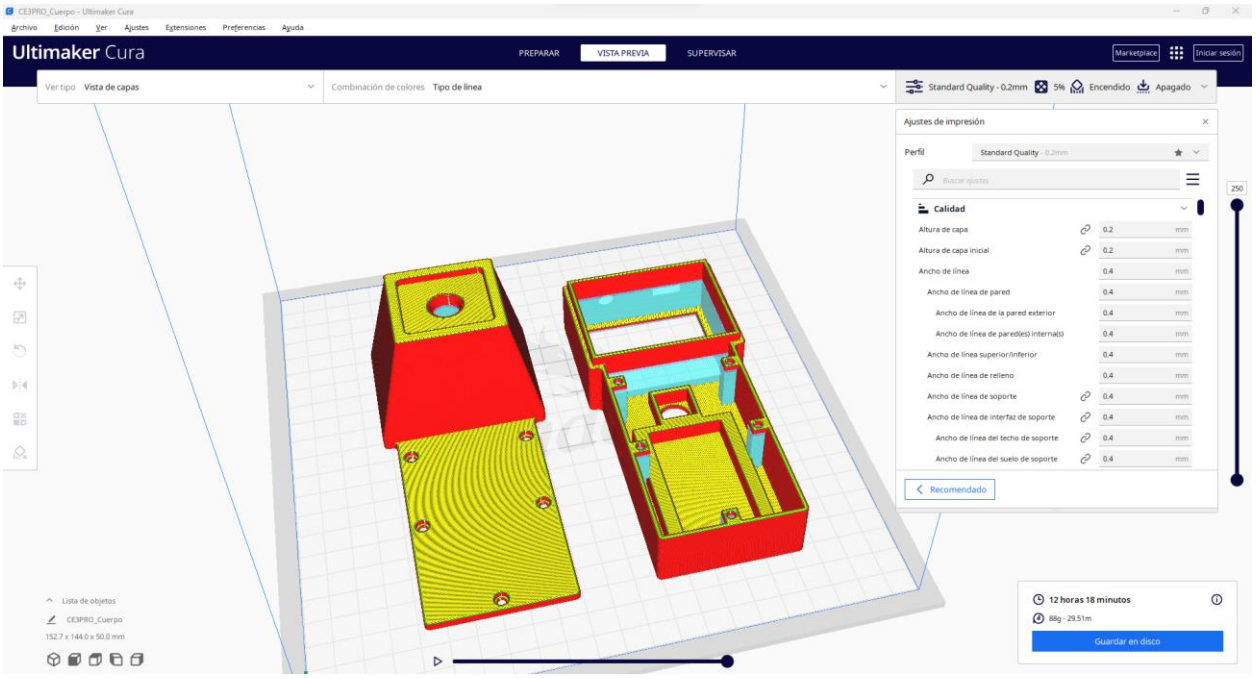
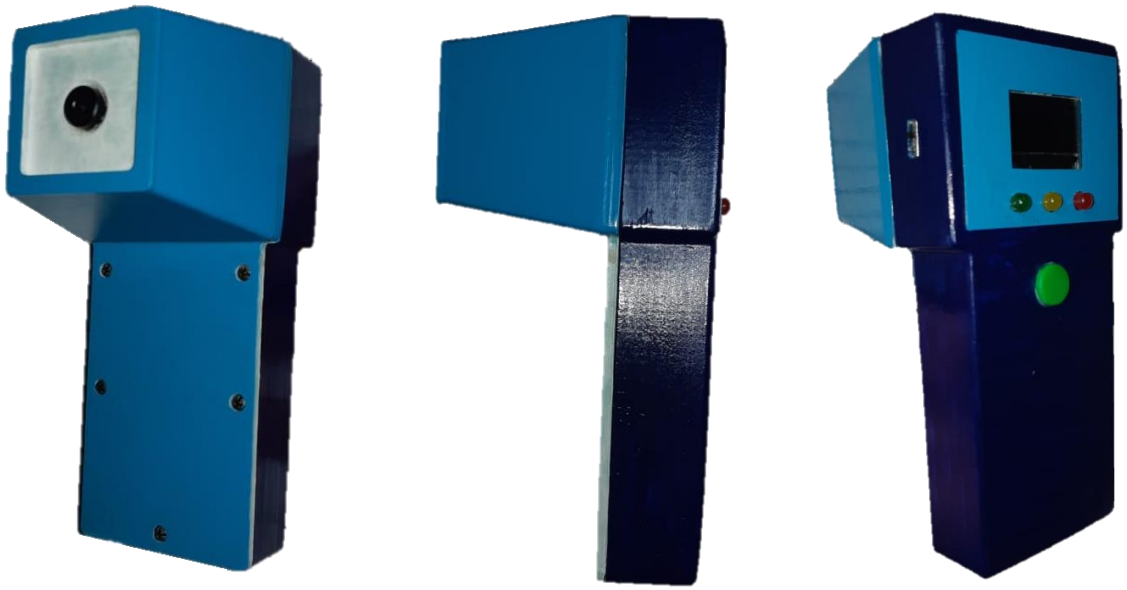


Figura 170

Resultado de la elaboración del termómetro infrarrojo



Oxímetro de pulso

El diseño se efectúa a partir de la selección de un sensor de oximetría, los sensores compatibles con Arduino más populares son el max30100 y max30102, no hay que confundirlos con el max30105 cuyo sensor óptico de alta sensibilidad está diseñado para aplicaciones de detección de humo. Los sensores max30100 y max30102 están diseñados para integrarlos en dispositivos portátiles, dispositivos de monitoreo médico, teléfonos inteligentes y monitoreo de actividad física. Si bien el max30102 es la versión mejorada del max30100, no presenta un incremento significativo en su precio.

Tabla 13

Características Max30100 vs Max 30102

Característica	Max30100	Max30102
Voltaje de alimentación	1.8 V - 3.3 V	1.8 V - 3.3 V
Corriente de apagado	0.7 µA	0.7 µA
ADC	14 bits	18 bits
Protocolo de comunicación	I2C	I2C
FIFO	16 bits	32 bits
Ancho de pulso	200 µs – 1.6 ms	69 µs – 411 µs
Corriente máxima	1200 µA	1200 µA
Frecuencia reloj I2C	400 kHz	400kHz

La sangre y la sangre saturada de oxígeno no absorben luz con la misma longitud de onda, la sangre con una pobre cantidad de oxígeno absorbe mayor cantidad de luz roja y la sangre oxigenada absorbe mayormente luz infrarroja, el sensor de oximetría tiene un led que emite luz roja con un espectro es de 660nm y un led que emite luz infrarroja con un espectro de 880nm, un fotodiodo mide la cantidad de luz roja e infrarroja reflejada y calcula el porcentaje de oxígeno en la sangre.

Tabla 14

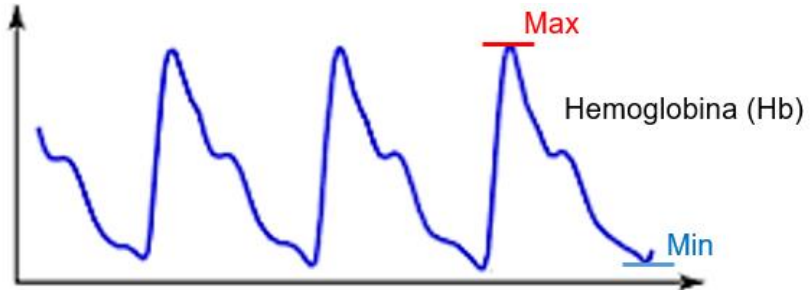
Linternas ordenadas por longitud de onda

Radiación		Longitud de onda
Ultravioleta 100 nm – 400 nm	Ultravioleta C	100 nm – 280 nm
	Ultravioleta B	280 nm – 315 nm
	Ultravioleta A	315 nm – 400 nm
Visible 400 nm – 780 nm	Violeta	400 nm – 455 nm
	Azul	455 nm – 490 nm
	Verde	490 nm – 570 nm
	Amarillo	570 nm – 590 nm
	Anaranjado	590 nm – 620 nm
	Rojo	620 nm – 780 nm
Infrarroja 780 nm – 1 mm	Infrarrojo A	780 nm – 1400 nm
	Infrarrojo B	1400 nm – 3000 nm
	Infrarrojo C	3000 nm – 1 mm

El sensor max30102 tiene un monitor de frecuencia cardíaca de bajo consumo, alta relación señal/ruido, por su ADC de 18 bits es más sensible a los cambios de voltaje del receptor IR y el ancho de pulso del led se puede programar. Cuando el corazón de una persona late bombea sangre a todo el cuerpo, el líquido circula a través de los vasos sanguíneos de tal modo que el volumen de los capilares aumenta y posterior a la contracción del corazón el volumen disminuye, los cambios del volumen de sangre en los vasos capilares provocan variaciones en lecturas del led infrarrojo lo que permite obtener la frecuencia cardiaca a partir de la frecuencia de la señal del led IR.

Figura 171

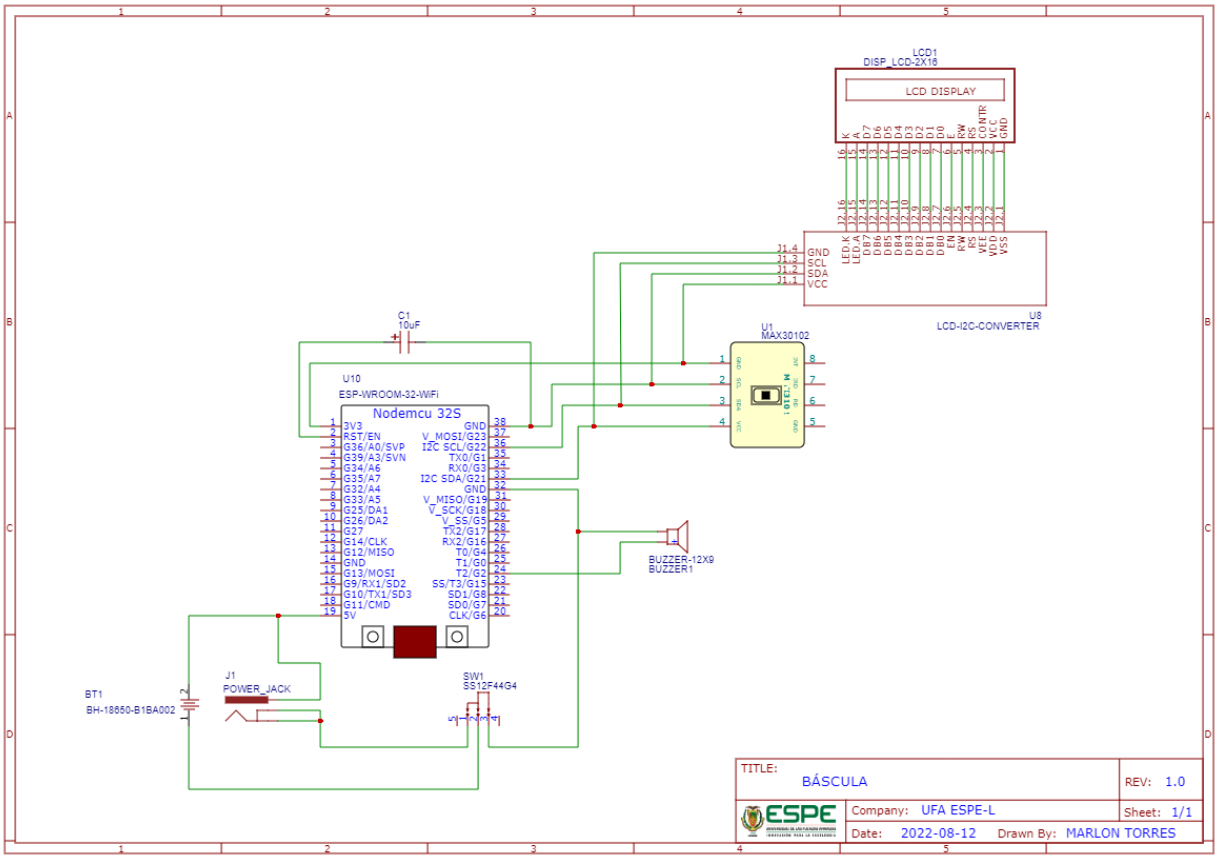
Señal de led IR para cálculo de oxígeno en la sangre



Nota. Los máximos y mínimos de las señales del led rojo e infrarrojo se pueden usar para obtener el %SPO2 con el método pico-valle. Tomado de (Figueroa et al., 2016).

El sensor Max30102 es de tipo reflectante, el emisor y receptor se encuentran uno al lado del otro y funciona cuando la luz se refleja en la punta del dedo de una persona, si no se coloca ninguna superficie reflexiva el receptor no recibe ninguna señal, gracias a esto se puede elaborar un algoritmo en el que los datos se registren únicamente en el momento en que alguien coloca su dedo sobre el sensor por lo que el dispositivo necesita únicamente un botón de encendido que da como resultado un dispositivo fácil de usar y un circuito más sencillo.

Figura 172 *Circuito electrónico oxímetro de pulso*



Los leds y el fotodetector del max30102 se encuentran cubiertos por un pequeño encapsulado de plástico transparente que sobresale de la placa, el encapsulado es frágil y dificulta instalar el sensor en la carcasa al montar el oxímetro además de que dificulta dar mantenimiento al mismo, para diseñar el modelo 3D es conveniente separar el sensor del resto de los componentes.

Para programar el oxímetro se usa la librería Sparkfun_MAX3010x, el módulo se comunica con el microcontrolador mediante I2C, tiene pre configurada la dirección 0XAE para escritura y la 0XAF para lectura.

Figura 173

Modelo 3D oxímetro de pulso

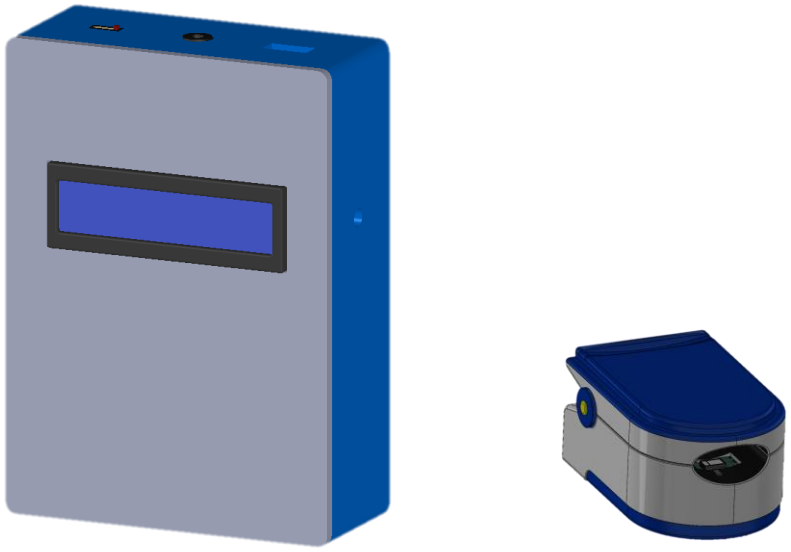
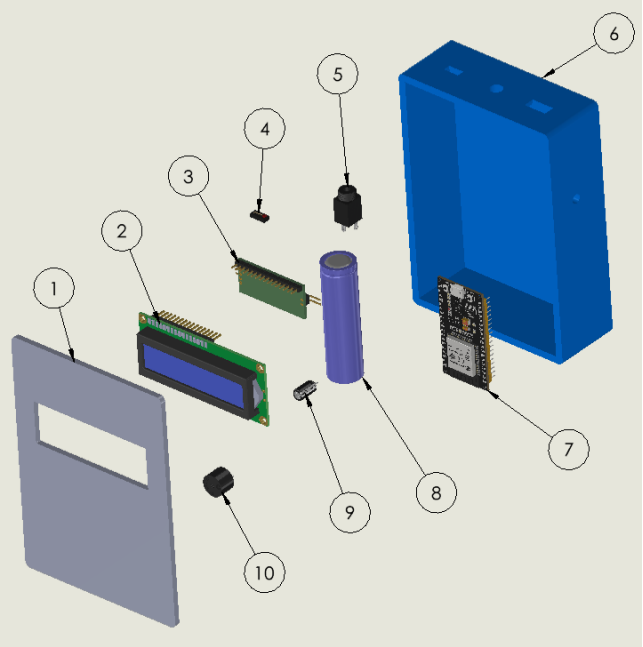


Figura 174

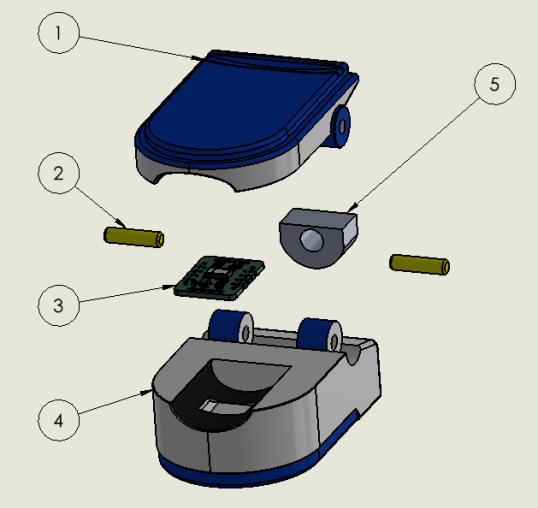
Vista explosionada y lista de materiales del procesador del oxímetro



N.º DE ELEMENTO	N.º DE PIEZA	CANTIDAD
1	Tapa	1
2	LCD 16X2	1
3	Módulo I2C	1
4	Conmutador deslizante	1
5	Jack 3.5 mm	1
6	Base	1
7	ESP 32	1
8	Batería 18650	1
9	Capacitor	1
10	Buzzer	1

Figura 175

Vista explosionada y lista de materiales pinza oxímetro de pulso



N.º DE ELEMENTO	N.º DE PIEZA	CANTIDAD
1	Mandíbula superior	1
2	Acople	1
3	Max30102	1
4	Mandíbula inferior	1
5	Tapón	1

Es bien sabido en la comunidad de la impresión 3D que una impresora bien calibrada puede imprimir hasta 4 centímetros en voladizo, por la forma y tamaño de la pinza de dedo se puede imprimir sin soportes, pero es aconsejable eliminar los soportes solo en la parte del sensor, si la impresora no se calibra bien quitar los soportes provocaría deformaciones en las que el sensor podría chocar y romperse al ensamblar la pinza.

Figura 176

Configuración para impresión 3D de la pinza del oxímetro

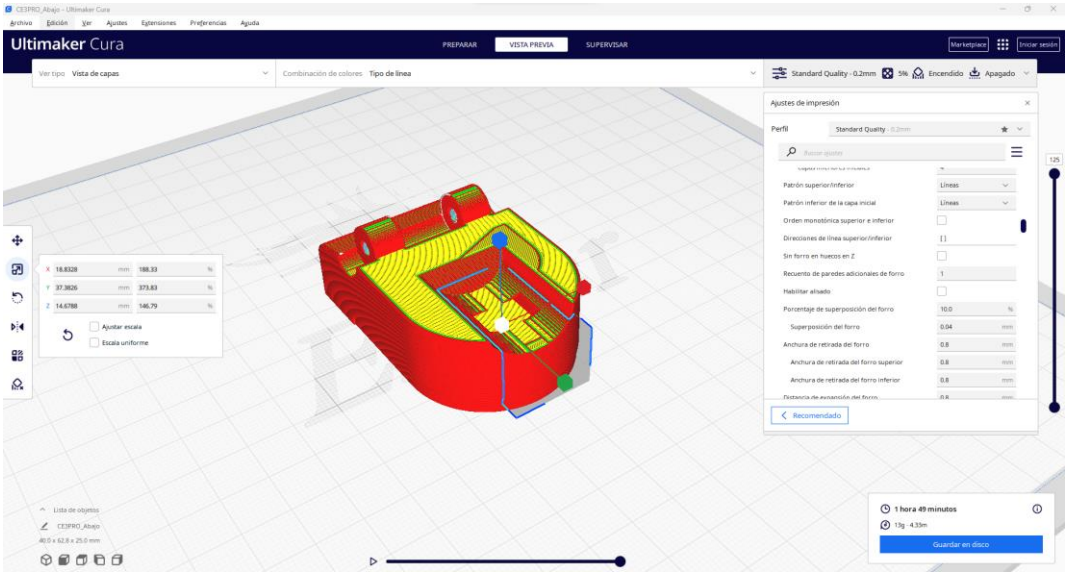
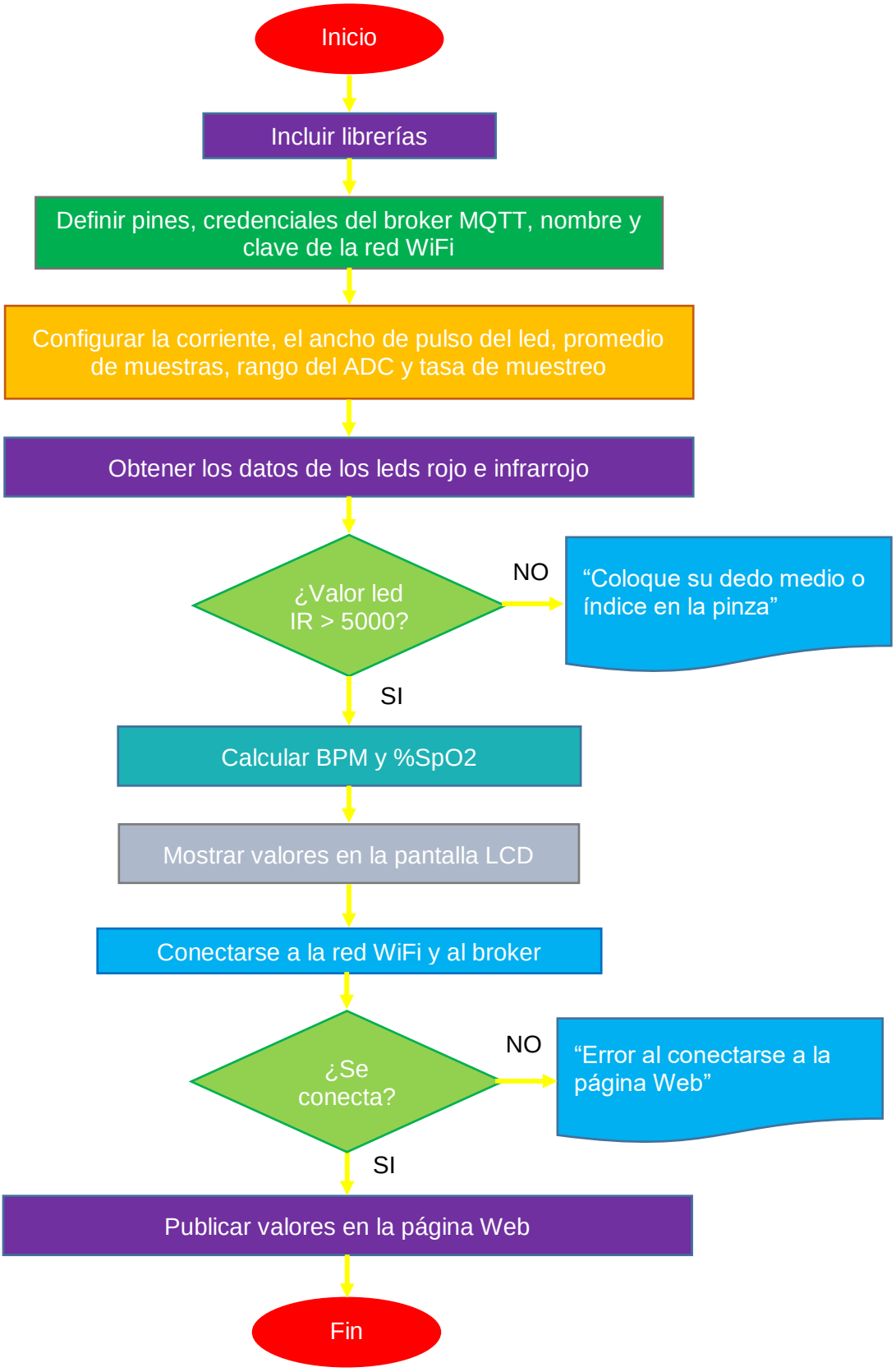


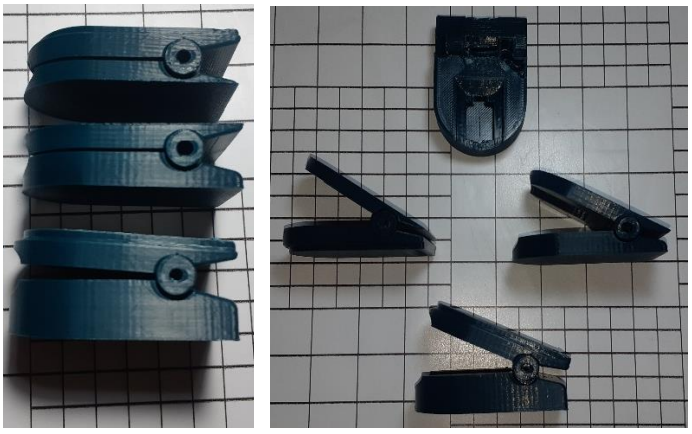
Diagrama de flujo de la programación del oxímetro de pulso



No se ha detallado con anterioridad, pero la impresión 3D de los dispositivos requiere muchas veces correcciones que aseguren su funcionamiento adecuado, por ejemplo, la pinza de dedo del oxímetro debe garantizar una presión constante del dedo del paciente sobre los leds del sensor Max30102, el mecanismo cuenta con un resorte y es más fácil ajustar el modelo 3D que fabricar un resorte a medida. No es necesario imprimir un modelo completo ni con una excelente calidad, se puede imprimir solo la parte de interés con calidad baja o media y realizar correcciones hasta llegar al diseño final para ahorrar tiempo y dinero.

Figura 177

Prototipado rápido con impresión 3D



La impresión 3D en filamento puede dar un aspecto poco confiable a los dispositivos provocando que los pacientes y el personal del centro de salud se muestren reacios a usar el sistema de monitoreo, mediciones correctas y un buen acabado en los dispositivos ayudaría a que a prefieran usar el sistema sobre los dispositivos más antiguos y a que exista un mayor interés en las tecnologías de monitoreo remoto.

Para obtener mejorar el aspecto de la impresión 3D lo primero es pasar una lija 300 sobre toda su superficie, luego se aplica una capa de masilla plástica cubriendo todas las imperfecciones, una vez que la masilla se seca se usa una lija 400 para el pulido, lija 600 para el pulido fino y lija 1000 para el pulido final, luego se aplica una capa de pintura acrílica y por último se aplican varias capas de barniz para pintura acrílica.

Figura 178

Verificar las ranuras del modelo 3D

Figura 179

Aplicar masilla plástica a impresión 3D

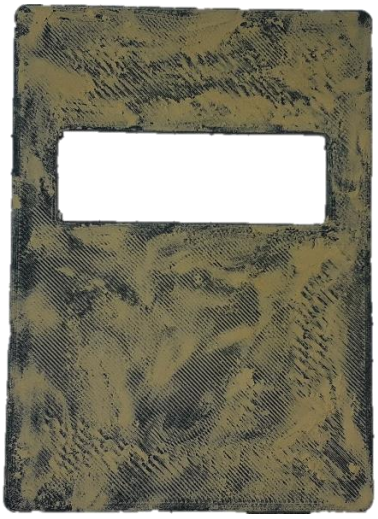


Figura 180

Pulir impresión 3D con lijas 400, 600 y 100

Figura 181

Resultado de pintar Impresión 3D con aerógrafo y pintura acrílica

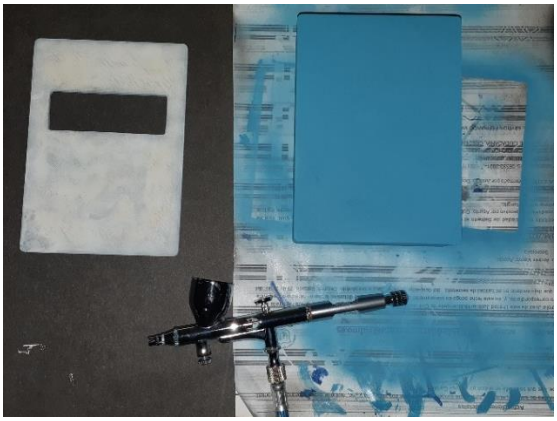


Figura 182

Resultado de aplicar varias capas de barniz

Figura 183

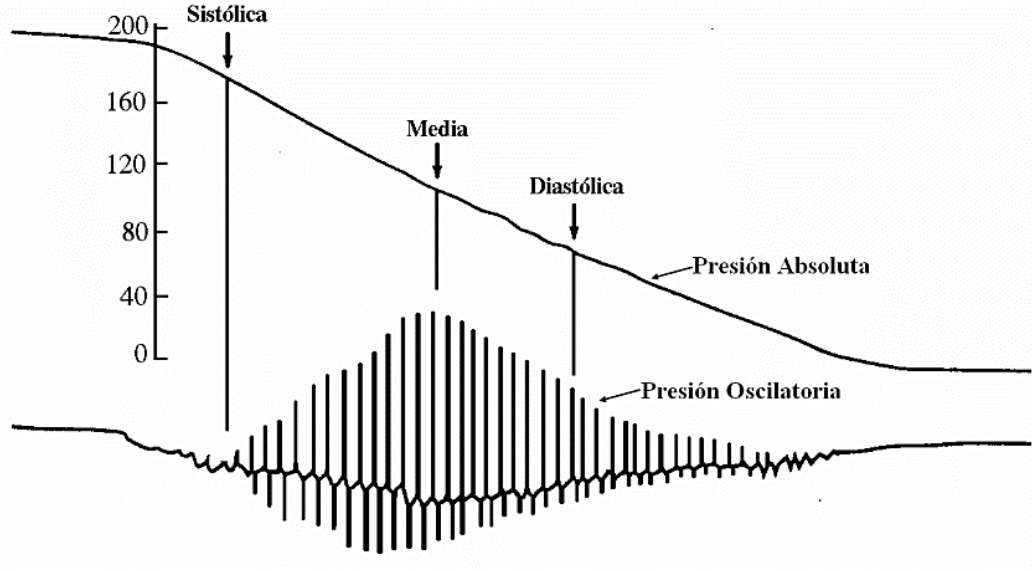
Resultado final del ensamblaje del oxímetro de pulso

Tensiómetro de muñeca

Para medir la presión arterial se pueden usar métodos invasivos y aunque son menos precisos también métodos no invasivos, dentro de los no invasivos la más usada es la medición oscilométrica, se coloca un brazalete en el brazo o muñeca del paciente y se posiciona a la altura del corazón, se infla hasta garantizar la oclusión de la arteria radial, posteriormente el brazalete se desinfla lentamente, cuando la arteria comienza a destaparse el flujo sanguíneo produce oscilaciones en las paredes de la arteria que se reflejan en los valores de presión medidos por el sensor, estos valores son amplificados y la señal resultante se conoce como presión absoluta, al aplicar un filtro pasa-banda se obtiene la presión oscilatoria, a partir de esta señal se pueden estimar los valores de la presión sistólica y diastólica.

Figura 184

Curva de presión oscilatoria



Nota. Para que un monitor de presión haga una estimación adecuada necesita una serie regular de latidos, si el pulso de un paciente es irregular el monitor podría no tener los datos suficientes, el filtro pasa-bandas se calcula en función del rango normal de latidos. Tomado de *GMISDN Blood pressure measurement webinar*, por (Lewis, 2021).

Para la integración del tensiómetro a la plataforma se ha optado por adaptar un tensiómetro comercial, esta es una tarea más sencilla pero que permite ahorrar mucho tiempo y dinero, los modelos más distribuidos a nivel mundial integran un sensor de presión MPS20N0040D apto para aplicaciones médicas como instrumentos de diagnóstico, también tienen una mini válvula solenoide y una mini bomba que trabajan en conjunto con el sensor permitiendo obtener la presión absoluta y la presión oscilatoria, estas dos señales se envían a un multiplexor que deja pasar solo la señal que solicita el microcontrolador hacia un ADC.

Tabla 15

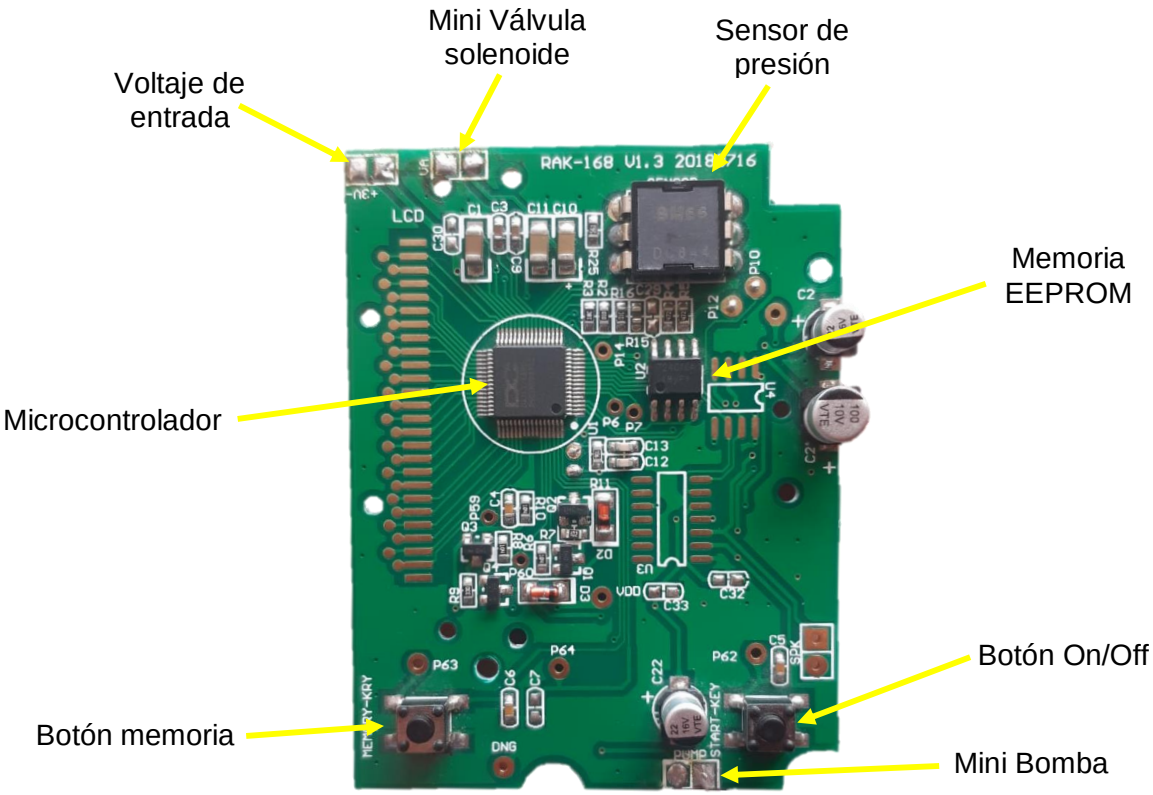
Especificaciones técnicas MPS20N0040D

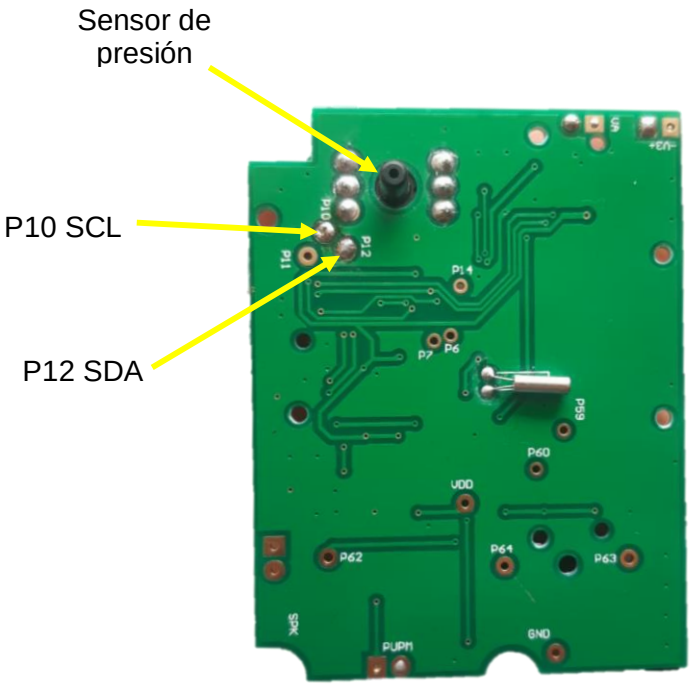
Especificaciones técnicas MPS20N0040D	
Alimentación	3.3 – 5 V
Rango de presión	0 – 300 mmHg
Encapsulado	SF = TO-39
Alimentación/precisión	D = 3V
Voltaje de salida	50 – 100 mV
Histéresis	± 0,7 % FS
Medios accesibles	Gases limpios, secos y no corrosivos
Encapsulado	DIP6
Linealidad	± 0,3 % FS

La placa del dispositivo contiene el sensor de presión, el microcontrolador, el filtro pasa-bajos y una memoria EEPROM T24C16A con interfaz I2C, si se configura un Arduino nano para que trabaje como esclavo del monitor de presión se pueden obtener los valores de presión absoluta y oscilatoria para controlar la mini válvula solenoide y la mini bomba. Cuenta con pulsadores NO para encender el dispositivo y acceder a la memoria EEPROM, los pines de SCL y SDA correspondientes a la interfaz I2C se pueden ubicar en la parte posterior de la placa.

Figura 185

Placa monitor de presión arterial Comercial





La librería del IDE de Arduino para configurar al ESP 32 como esclavo tiene varios errores y la configuración es poco intuitiva por lo que se usa un Arduino nano en su lugar, este a su vez se comunica con la ESP 32 mediante UART, los pines Rx y Tx del Arduino nano funcionan a 5 voltios mientras que los pines de Rx y Tx del ESP 32 trabajan a 3.3 voltios, para conectarlos de forma segura y bidireccional se usa un conversor de niveles lógicos que funciona con cualquier señal digital, de este modo se consigue enviar los valores de presión arterial del Arduino al ESP 32 que finalmente los envía a la página web. Para evitar errores y disminuir la latencia en la lectura de la memoria EEPROM del tensiómetro comercial se ha decidido no usar el módulo I2C para la pantalla LCD.

Figura 186

Circuito electrónico del tensiómetro con IoT

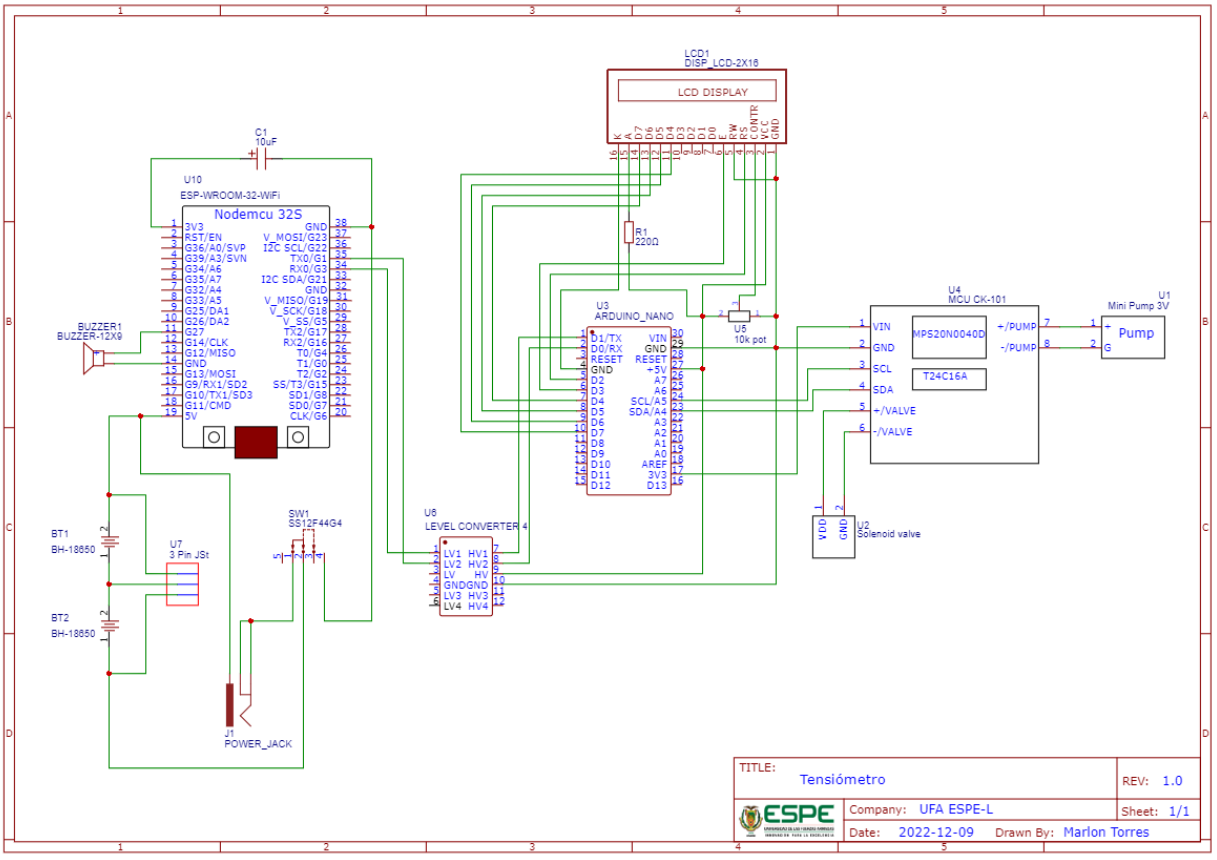
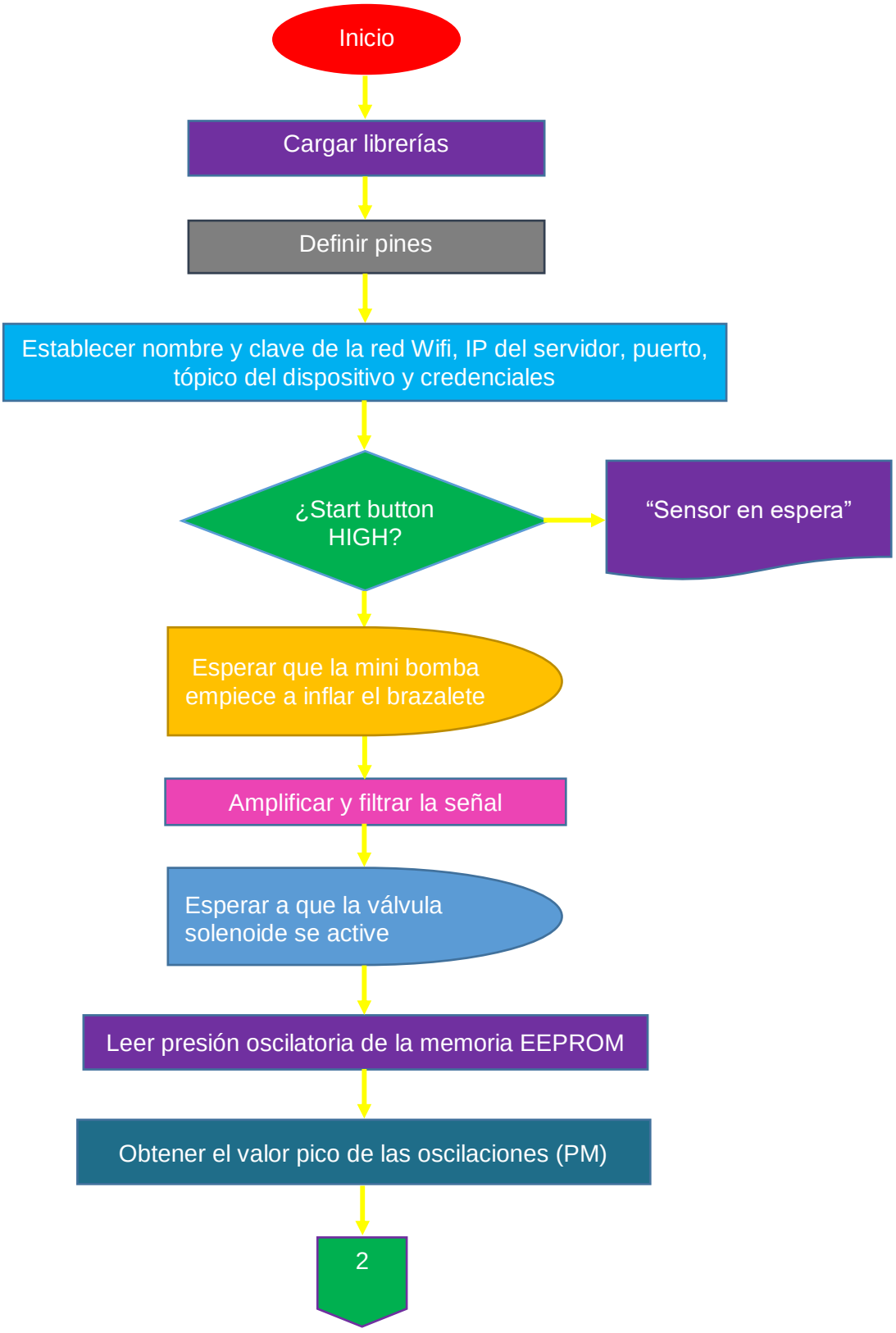
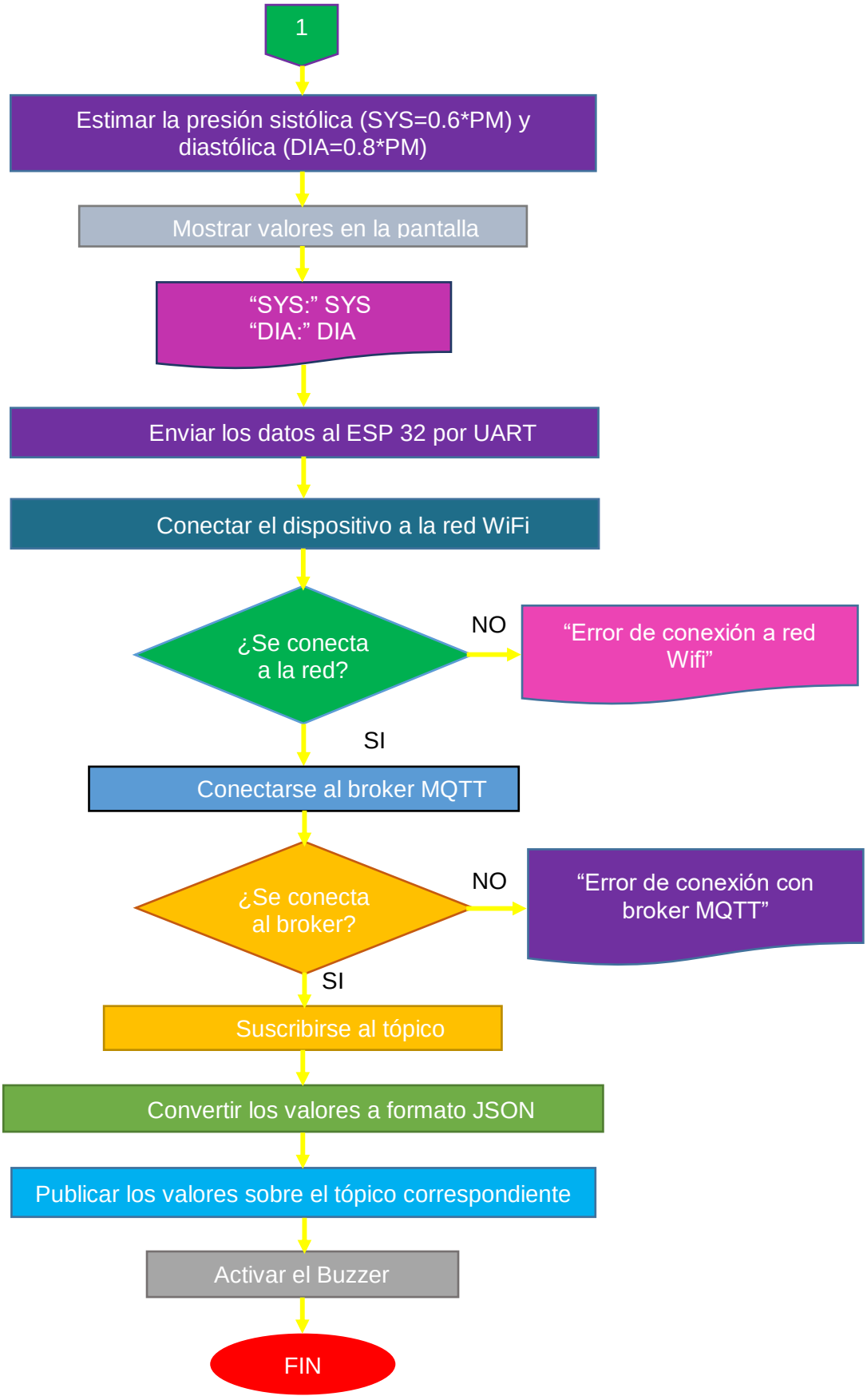


Diagrama de flujo de la programación del tensiómetro




1
Estimar la presión sistólica (SYS=0.6*PM) y diastólica (DIA=0.8*PM)
Mostrar valores en la pantalla
“SYS:” SYS
“DIA:” DIA
Enviar los datos al ESP 32 por UART
Conectar el dispositivo a la red WiFi
¿Se conecta a la red?
NO
“Error de conexión a red Wifi”
SI
Conectarse al broker MQTT
¿Se conecta al broker?
NO
“Error de conexión con broker MQTT”
SI
Suscribirse al tópico
Convertir los valores a formato JSON
Publicar los valores sobre el tópico correspondiente
Activar el Buzzer
FIN

El circuito se alimenta con un pack de baterías 2S, el pack se arma con dos baterías 18650 de 6800mAh por lo que a la salida se obtiene 7.4 voltios para alimentar al Arduino nano, con esto se consiguen los 5 voltios para la comunicación serial del lado de alto voltaje y los 5 voltios para la pantalla LCD, se extraen de un tensiómetro comercial una mini bomba y una mini válvula que trabajan a 3.3 voltios junto con el brazalete y se encargan de automatizar el dispositivo, además, se parte de las medidas de estos tres componentes para el diseño 3D del tensiómetro. La ESP 32 también se alimenta con el pack de baterías a 7.4 voltios, los pines de comunicación serial se conectan al lado de bajo voltaje del conversor lógico de nivel.

Figura 187

Brazalete, válvula y bomba tensiómetro comercial

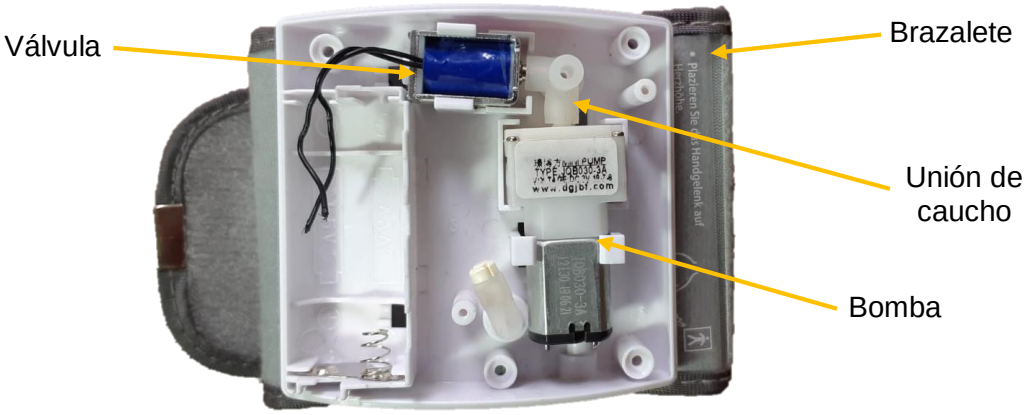
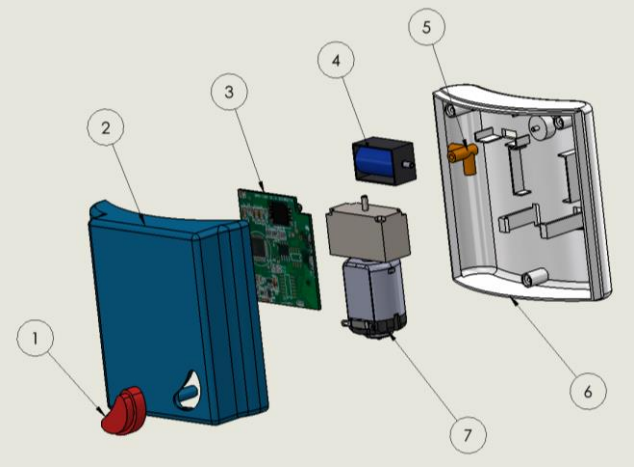


Figura 188

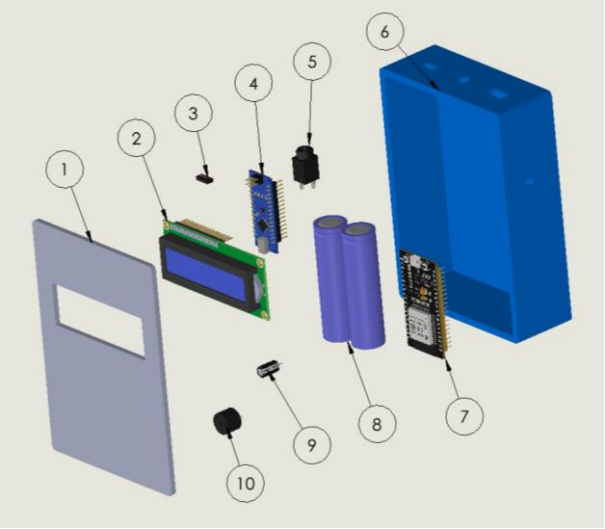
Vista explosionada y lista de materiales de la muñequera del tensiómetro



N.º DE ELEMENTO	N.º DE PIEZA	CANTIDAD
1	Push Button	1
2	Tapa	1
3	Microcontrolador tensiómetro	1
4	Mini válvula	1
5	Unión de caucho	1
6	Base	1
7	Mini bomba	1

Figura 189

Vista explosionada y lista de materiales del procesador del tensiómetro



N.º DE ELEMENTO	N.º DE PIEZA	CANTIDAD
1	Tapa	1
2	LCD 16X2	1
3	Conmutador deslizante	1
4	Arduino Nano	1
5	Jack 3.5 mm	1
6	Base	1
7	ESP 32	1
8	Pack Baterías	1
9	Capacitor	1
10	Buzzer	1

Figura 190

Impresión 3D y postprocesado del tensiómetro

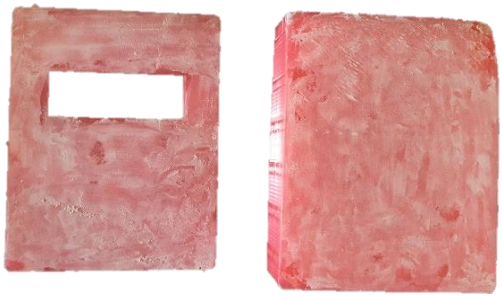


Figura 191

Capa de pintura base sobre carcasa del tensiómetro

Figura 192

Pintura acrílica y laca protectora sobre carcasa del tensiómetro

Figura 193

Resultado del ensamble de la muñequera del tensiómetro

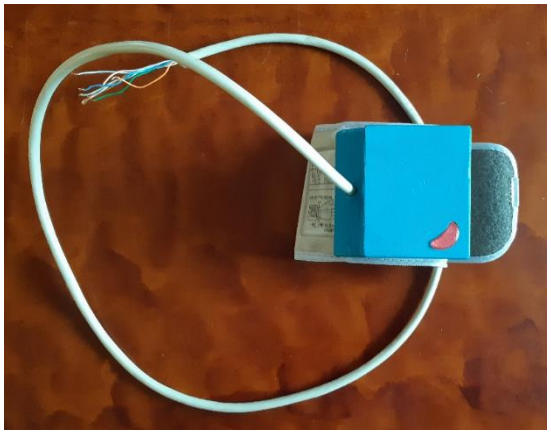


Figura 194

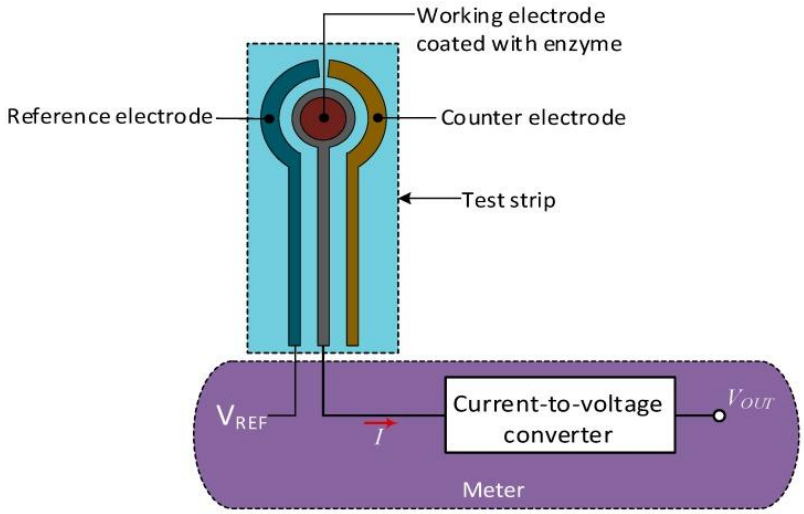
Resultado final del ensamblaje del tensiómetro

Glucómetro

El diseño del glucómetro parte de la selección de una tira reactiva comercial en configuración de contador que se puede identificar porque tiene tres terminales, cuando se coloca una gota de sangre en el área reactiva la glucosa presente se oxida por la presencia de la enzima glucosa oxidasa (GOx) colocada en el terminal Working, es una oxidorreductasa que cataliza la oxidación electroquímica de la glucosa formando peróxido de hidrógeno y ácido glucónico, finalmente el peróxido de hidrógeno se oxida liberando electrones que se hacen circular aplicando voltaje a los terminales Working y Counter, como resultado se obtiene una señal de corriente.

Figura 195

Diagrama de tira reactiva para medición de glucosa método de punción digital



Nota. La cantidad de corriente que produce la oxidación de la glucosa es proporcional al nivel de glucosa en sangre, esta proporcionalidad se mantiene luego de convertir la corriente en voltaje. Tomado de (Gonzales et al., 2019).

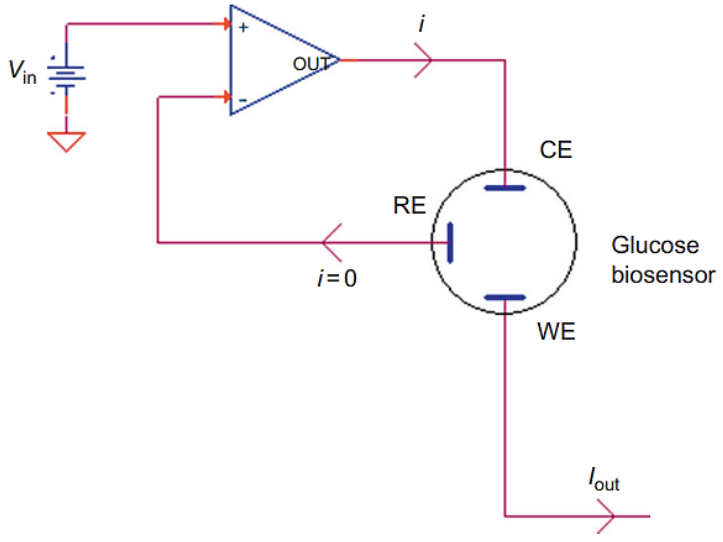
Para que la tira reactiva funcione hay que aplicar un voltaje fijo entre los terminales de Working y Counter, para posteriormente monitorear la corriente resultante, el voltaje aplicado puede ser constante o en una secuencia conocida, en este caso se aplicará un voltaje de

400mV, que será suficiente para hacer circular los electrones. El terminal Working detecta la corriente de la reacción, el terminal Reference mantiene el voltaje constante con respecto al electrodo Working y el terminal Counter suministra corriente al terminal Working.

Para controlar el voltaje del electrodo de trabajo y medir la corriente producida por la tira reactiva sin tener en cuenta los cambios de la carga se utiliza un potenciostato, se diferencia de las fuentes de corriente constante comunes porque es capaz de medir y suministrar corriente en un rango que va desde los picoamperios hasta los amperios en ambas polaridades. Para el circuito se utiliza un amplificador operacional que presenta una alta impedancia de entrada y muy alta ganancia por lo que tanto el flujo de electrones como la diferencia de voltaje entre los pines de entrada son casi nulos.

Figura 196

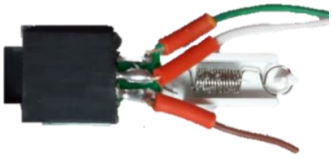
Medida amperimétrica de glucosa mediante un circuito de potenciostato



Nota. Un voltaje Vin aplicado al pin “+” resulta en una corriente de salida que fluye del electrodo Counter al electrodo Working, con esto se consigue que la diferencia de potencial que se produce entre los electrodos Reference y Working sea la misma que se aplica al pin “+”, manteniendo una diferencia de 0 voltios entre los pines de entrada. Tomado de (Fitzpatrick, 2014).

Figura 197

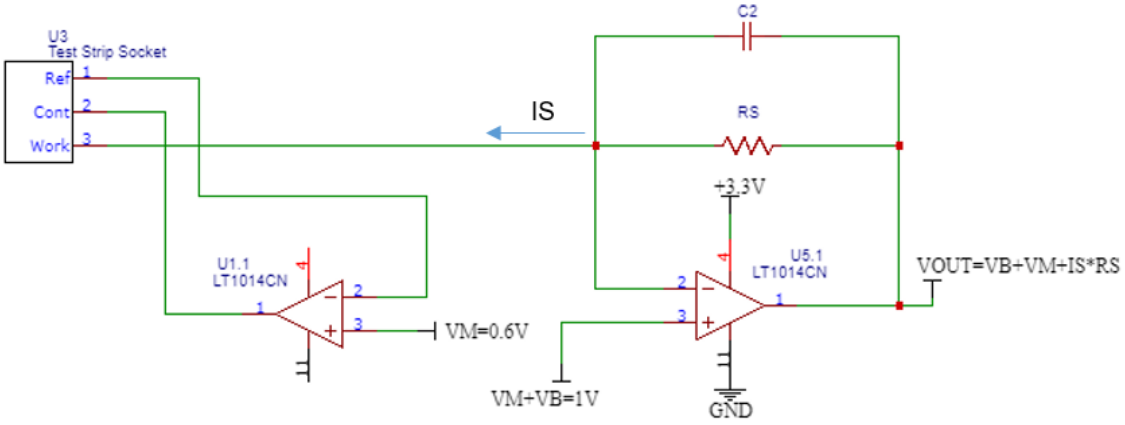
Socket para tiras reactivas



Para alimentar la tira reactiva se usan divisores de voltaje de los que se obtiene 0.6V y 1V, estos voltajes se conectan a las entradas no inversoras de los amplificadores operacionales del potenciostato y del conversor de corriente a voltaje respectivamente, el resultado es que entre los terminales de Working y Counter se aplica un voltaje de 0.4V, o lo que es lo mismo, los 400Mv necesarios para que los electrones circulen entre ambos terminales.

Figura 198

Tira de prueba electroquímica en una configuración de contador



Nota. El voltaje de polarización en el sitio de reacción de la tira reactiva se establece y mantiene con mayor precisión durante la medición al usar esta configuración.

Los valores de corriente que arroja un glucómetro están entre 10µA y 50µA, la ESP 32 trabaja a 3.3V, con estos valores se puede calcular el valor de RS para el conversor de corriente-voltaje, luego se selecciona un valor de resistencia comercial que para este caso corresponde a una resistencia de 44kΩ.

Para $I_S = 50\ \mu A$:

$$Vout = V_B + V_M + I_S * R_s$$

$$Vout = 1V + 50x10^{-6}A * 44x10^3\Omega$$

$$Vout = 3.2V$$

Para $I_S = 10\ \mu A$:

$$Vout = V_B + V_M + I_S * R_s$$

$$Vout = 1V + 10x10^{-6} * 44x10^3\Omega$$

$$Vout = 1.44\ V$$

El circuito también incluye un capacitor para filtrar el ruido de alta frecuencia pero que no es de vital importancia, para una frecuencia de corte de 8Hz el valor del capacitor es:

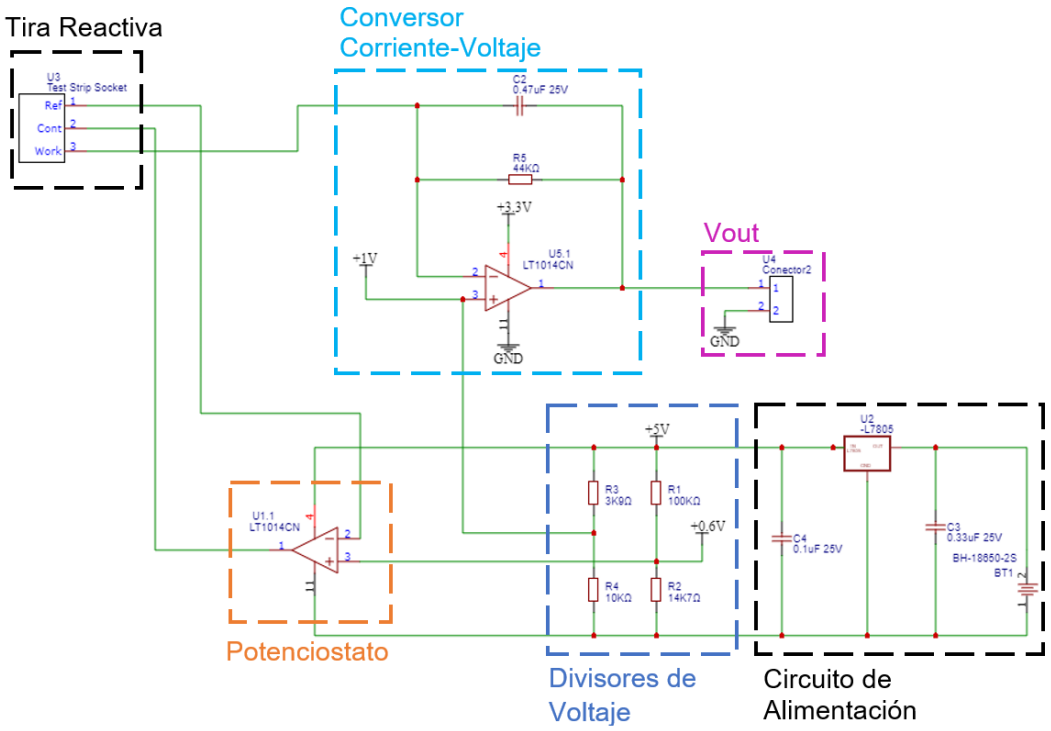
$$C = \frac{1}{2\pi * 44k\Omega * 8\text{Hz}}$$

$$C = 0.45\mu F$$

El capacitor comercial con el valor más próximo es de 0.47 µF

Figura 199

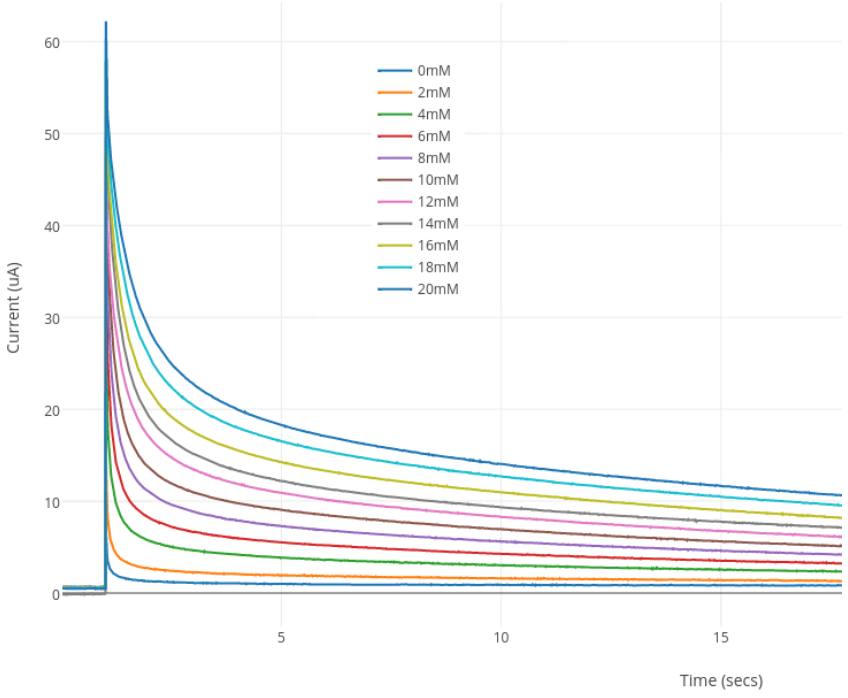
Circuito para adquisición de la señal de la tira reactiva



La señal que sale de la tira reactiva por el terminal Working pasa por un conversor de corriente-voltaje que tiene un offset de 1V y una resistencia de $R5 = 44K\Omega$, si se trabaja con un valor de resistencia lo suficientemente alto no hay necesidad de amplificar la señal de la tira reactiva que se encuentra en el orden de los µA, pero se corre el riesgo de saturar el amplificador operacional, especialmente porque la tira reactiva no produce una señal de corriente constante como se observa en la Figura 200, este problema se puede ignorar por que el valor de voltaje será medido una vez que la corriente se haya estabilizado.

Figura 200

Curva cronoamperométrica típica de tira reactiva comercial



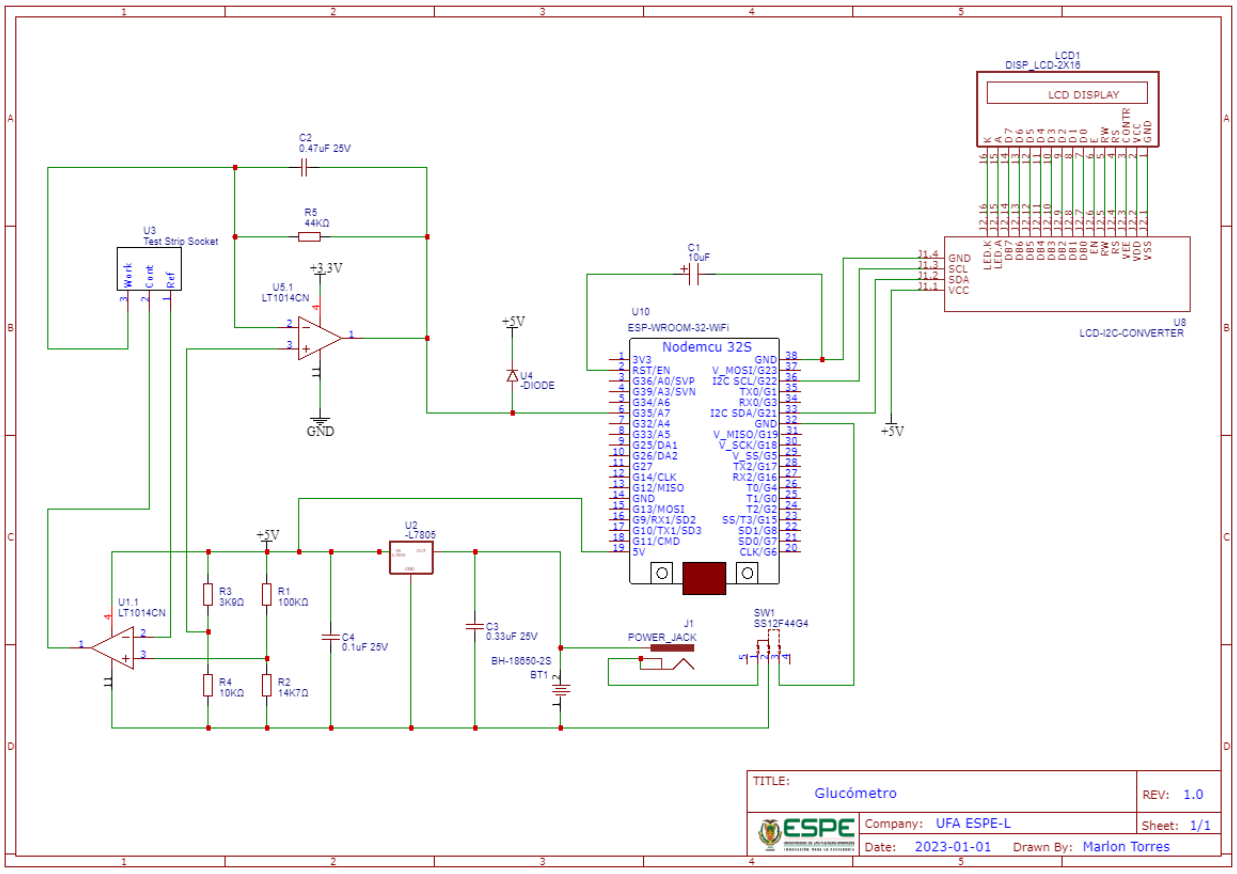
Nota. La gráfica corresponde a tiras reactivas alimentadas con 500Mv durante 29 segundos. Tomado de (Long, 2018).

A continuación, se usa el circuito de la Figura 199 para adquirir los valores de voltaje por medio de una ESP 32, el circuito resultante se muestra en la Figura 201, con la resistencia de 44KΩ se obtuvieron voltajes de entre 1.2 y 3.2 voltios para valores de 58 mg/dl y 215 mg/dl

respectivamente, estos valores abarcan los rangos de glucosa en sangre que indican las tablas por lo que el circuito es adecuado para la elaboración de un glucómetro.

Figura 201

Circuito eléctrico Glucómetro IoT



Para la mayoría de las pruebas se solicitó la ayuda de algunos pacientes del centro de salud y de algunos habitantes de la parroquia, sin embargo, por la dificultad de conseguir valores relativamente bajos y altos de glucosa en sangre que permitan garantizar el funcionamiento del glucómetro en un rango aceptable de valores se optó por elaborar una solución de glucosa 100mM en agua destilada con ayuda del colaborador científico. Para la solución se usó el producto D-(+)-Glucose, >=99.5% (GC) código G8270-100G de la marca Sigma-Aldrich, tanto la solución como las tiras reactivas son costosas por lo que los valores de los que se dispone para calibrar el sensor son limitados.

Figura 202

Toma de muestras de glucosa

Luego de medir varias muestras con el glucómetro comercial y con el glucómetro IoT que corresponde al circuito de la Figura 201 se obtuvieron los siguientes resultados:

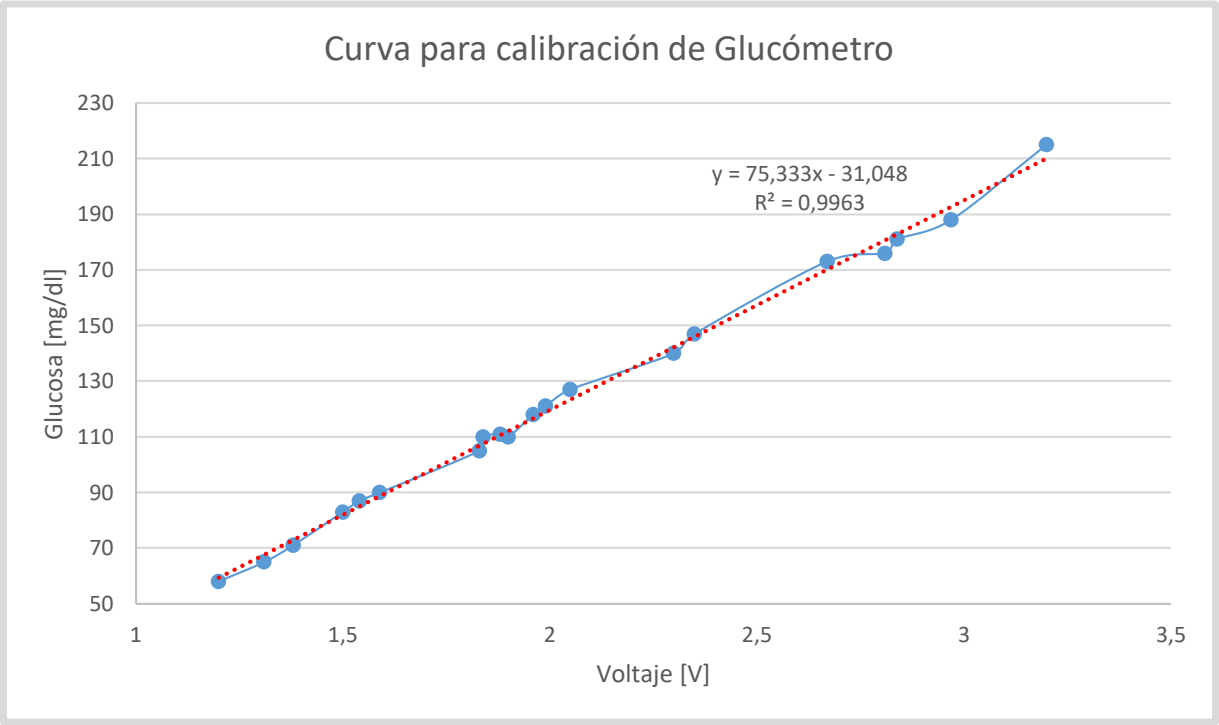
Tabla 16

Valores de voltaje del glucómetro IoT vs glucosa del glucómetro comercial

Ítem	Voltaje [V]	Glucosa [mg/dl]
1	1,2	58
2	1,31	65
3	1,38	71
4	1,5	83
5	1,54	87
6	1,59	90
7	1,83	105
8	1,84	110
9	1,88	111
10	1,9	110
11	1,96	118
12	1,99	121
13	2,05	127
14	2,3	140
15	2,35	147
16	2,67	173
17	2,81	176
18	2,84	181
19	2,97	188
20	3,2	215

Figura 203

Curva para calibración de glucómetro



De la gráfica anterior se obtiene la ecuación que modela la relación entre el valor de glucosa en sangre que arroja el glucómetro comercial y el voltaje correspondiente que mide el glucómetro IoT.

$$Glucosa\ \left[\frac{mg}{dl}\right] = 75.333 * Voltaje[V] - 31.048$$

Los valores de voltaje se ubican entre 1V y 3.3V debido al offset del amplificador operacional del conversor de corriente a voltaje y al voltaje de que se aplica en su terminal de alimentación positiva. A partir del coeficiente de determinación se calcula el coeficiente de Pearson:

$$R = \sqrt{R^2} = \sqrt{0.9963}$$

$$R = 0.9981$$

El coeficiente de Pearson muestra una fuerte correlación positiva por lo que se procede con la calibración del glucómetro IoT mediante la ecuación de regresión lineal.

Diagrama de flujo de la programación del glucómetro

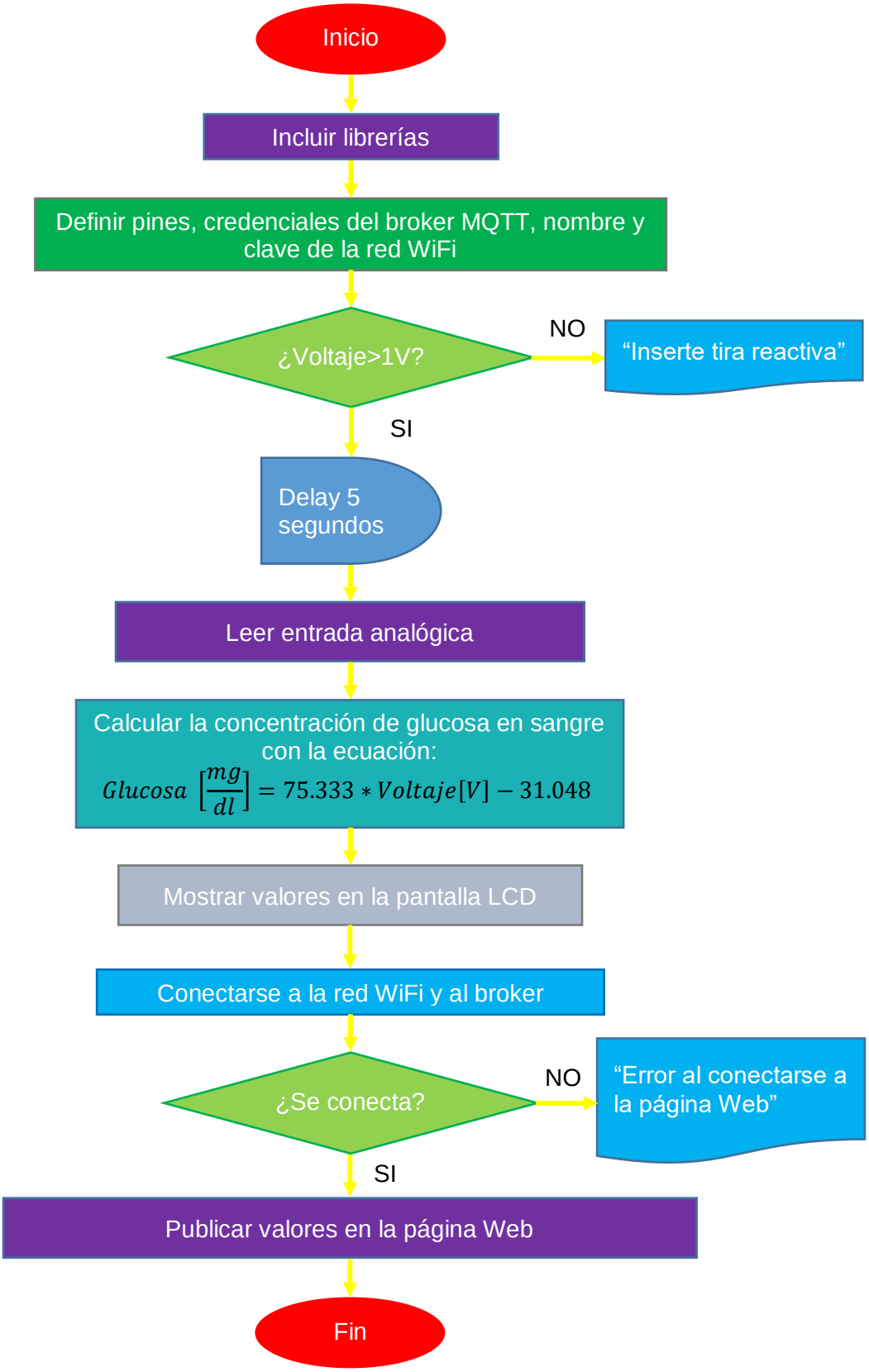
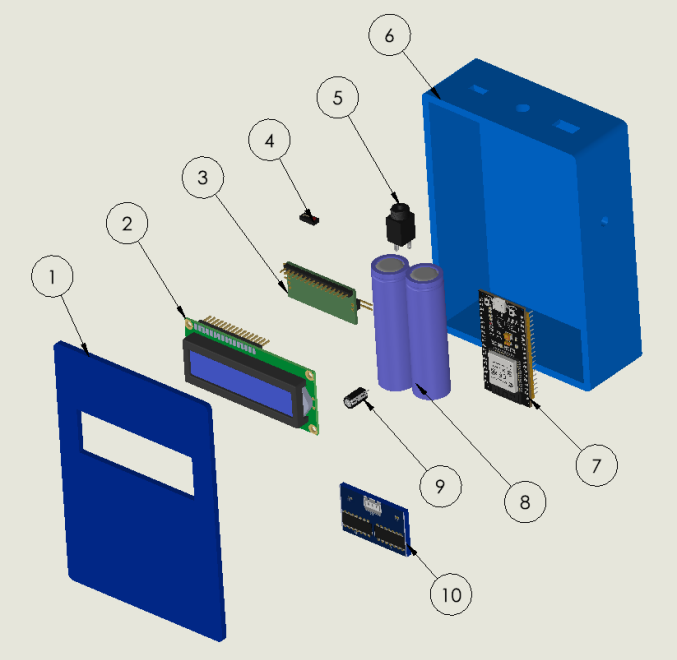


Figura 204

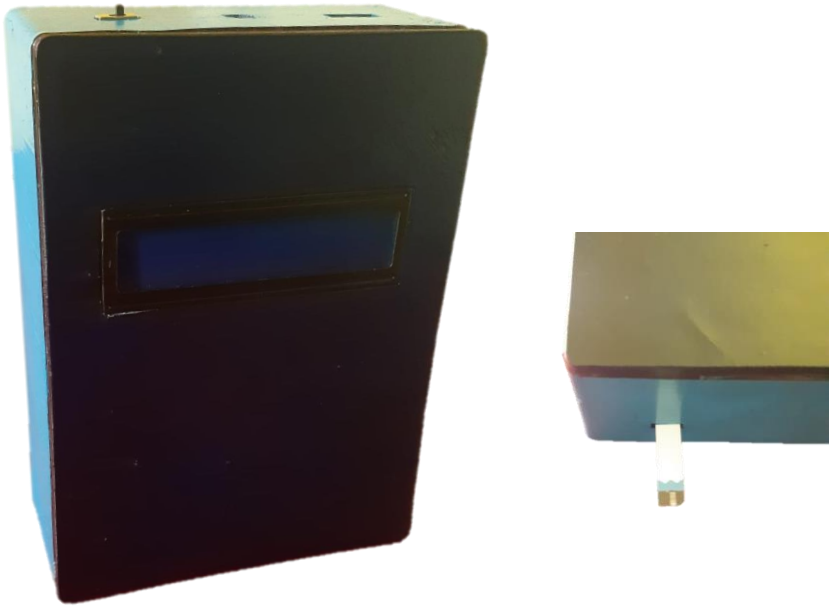
Vista explosionada y lista de materiales glucómetro



N.º DE ELEMENTO	N.º DE PIEZA	CANTIDAD
1	Tapa	1
2	LCD 16X2	1
3	Módulo I2C	1
4	Conmutador deslizante	1
5	Jack 3.5 mm	1
6	Base	1
7	ESP 32	1
8	Pack Batería 2S	1
9	Capacitor	1
10	Placa conversor	1

Figura 205

Resultado de la construcción del Glucómetro



Para transportar los sensores, un par de cables y consumibles con facilidad de modo que el sistema cause una buena impresión a los pacientes se elaboró un maletín de madera con compartimentos hechos a la medida y forrado de terciopelo.

Figura 206

Modelo 3D maletín de madera

Figura 207

Resultado de la elaboración del maletín de madera

Figura 208

Ensamble 3D del maletín de madera y los sensores

Figura 209

Resultado de la elaboración del maletín y los sensores

Capítulo IV

Implementación, pruebas y resultados

Resultados dela implementación de la báscula

El módulo HX711 da buenos resultados en la construcción de básculas digitales, sin embargo, se han realizado múltiples pruebas para garantizar el correcto funcionamiento del dispositivo y la validez del dato que se envía a la plataforma, una vez que un paciente se sube a la báscula se toman los valores del peso durante cuatro segundos, pero, se envía únicamente el último valor registrado para asegurar que el valor se guarda una vez que el paciente ha subido ambos pies a la báscula y se ha colocado en la posición correcta. Los datos registrados en la página web se comparan con valores medidos con la báscula del centro de salud en pacientes de distintas edades y pesos.

Figura 210

Báscula mecánica de columna del centro de salud

Figura 211

Toma de peso con báscula mecánica

Figura 212

Toma de peso con báscula de la plataforma

Figura 213

Pesajes en la escuela de la parroquia

Figura 214

Resultados del pesaje en la escuela de la parroquia

Tabla 17

Valores de peso para cálculos de error

Báscula con módulo HX711		
Peso real = 82,45 Kg = 181.77 Lb		
Ítem	**Peso en Kilogramos**	**Peso en libras**
1	82,4	181,66
2	82,47	181,82
3	82,46	181,79
4	82,47	181,82
5	82,46	181,79
6	82,62	182,15
7	82,45	181,77
8	82,43	181,73
9	82,45	181,77
10	82,31	181,46
11	82,43	181,73
12	82,55	181,99
13	82,5	181,88
14	82,42	181,7
15	82,43	181,73
16	82,68	182,28
17	82,37	181,59
18	82,43	181,73
19	82,42	181,7
20	82,39	181,64
21	82,43	181,73
22	82,47	181,82
23	82,45	181,77
24	82,4	181,66
25	82,42	181,7
26	82,44	181,75
27	82,41	181,68
28	82,42	181,7
29	82,35	181,55
30	82,41	181,68

Cálculo de errores para el peso en Kilogramos

Valor promedio:

$$\overline{To} = \frac{\sum_{i=1}^{n} To_i}{n}$$

$$\overline{To} = 82{,}446\ Kg$$

Erro absoluto:

$$Ea_{To} = \frac{\sum_{i=1}^{n} |To_i - \overline{To}|}{n}$$

$$Ea_{To} = 0{,}04626\ Kg$$

Error relativo:

$$Er_{To} = \frac{Ea_{To}}{\overline{To}}$$

$$Er_{To} = 0{,}0005611$$

Error porcentual:

$$Ep_{To} = Er_{To} * 100$$

$$Ep_{To} = \ 0{,}0561\ \%$$

La precisión del sensor de peso en Kilogramos es de $82.45 \pm 0.0561\%$

Cálculo de errores para el peso en Libras

Valor promedio:

$$\overline{To} = \frac{\sum_{i=1}^{n} To_i}{n}$$

$$\overline{To} = 181{,}759\ Lb$$

Erro absoluto:

$$Ea_{To} = \frac{\sum_{i=1}^{n} |To_i - \overline{To}|}{n}$$

$$Ea_{To} = 0{,}1004\ Lb$$

Error relativo:

$$Er_{To} = \frac{Ea_{To}}{\overline{To}}$$

$$Er_{To} = 0{,}00055$$

Error porcentual:

$$Ep_{To} = Er_{To} * 100 = 0.055\ \%$$

La precisión del sensor de peso en Kilogramos es de $181.77 \pm 0.055\%$

Tabla 18

Valoración de repetibilidad báscula con HX711

	Peso en Kilogramos				Peso en Libras			
Ítem	Variable	Indicación	Diferencia	Cuadrado diferencia	Variable	Indicación	Diferencia	Cuadrado diferencia
1	5	5,1	-0,1	1,00E-02	11,0231	11,243562	-0,220462	4,86E-02
2	10	10,03	-0,03	9,00E-04	22,0462	22,1123386	-0,0661386	4,37E-03
3	15	15,02	-0,02	4,00E-04	33,0693	33,1133924	-0,0440924	1,94E-03
4	20	19,98	0,02	4,00E-04	44,0924	44,0483076	0,0440924	1,94E-03
5	25	25,07	-0,07	4,90E-03	55,1155	55,2698234	-0,1543234	2,38E-02
6	30	30	0	0,00E+00	66,1386	66,1386	0	0,00E+00
7	35	35,02	-0,02	4,00E-04	77,1617	77,2057924	-0,0440924	1,94E-03
8	40	39,99	0,01	1,00E-04	88,1848	88,1627538	0,0220462	4,86E-04
9	45	45,06	-0,06	3,60E-03	99,2079	99,3401772	-0,1322772	1,75E-02
10	50	50,04	-0,04	1,60E-03	110,231	110,319185	-0,0881848	7,78E-03
11	55	55	0	0,00E+00	121,2541	121,2541	0	0,00E+00
12	60	59,94	0,06	3,60E-03	132,2772	132,144923	0,1322772	1,75E-02
13	65	64,97	0,03	9,00E-04	143,3003	143,234161	0,0661386	4,37E-03
14	70	69,68	0,32	1,02E-01	154,3234	153,617922	0,7054784	4,98E-01
15	75	75,08	-0,08	6,40E-03	165,3465	165,52287	-0,1763696	3,11E-02
16	80	80,02	-0,02	4,00E-04	176,3696	176,413692	-0,0440924	1,94E-03
17	85	84,96	0,04	1,60E-03	187,3927	187,304515	0,0881848	7,78E-03
18	90	90	0	0,00E+00	198,4158	198,4158	0	0,00E+00
19	95	94,92	0,08	6,40E-03	209,4389	209,26253	0,1763696	3,11E-02
20	100	99,94	0,06	3,60E-03	220,462	220,329723	0,1322772	1,75E-02
21	105	105,04	-0,04	1,60E-03	231,4851	231,573285	-0,0881848	7,78E-03
22	110	109,99	0,01	1,00E-04	242,5082	242,486154	0,0220462	4,86E-04
23	115	114,96	0,04	1,60E-03	253,5313	253,443115	0,0881848	7,78E-03
24	120	120	0	0,00E+00	264,5544	264,5544	0	0,00E+00
25	125	125,06	-0,06	3,60E-03	275,5775	275,709777	-0,1322772	1,75E-02
26	130	129,95	0,05	2,50E-03	286,6006	286,490369	0,110231	1,22E-02
27	135	135,02	-0,02	4,00E-04	297,6237	297,667792	-0,0440924	1,94E-03
28	140	139,98	0,02	4,00E-04	308,6468	308,602708	0,0440924	1,94E-03
29	145	145,01	-0,01	1,00E-04	319,6699	319,691946	-0,0220462	4,86E-04
	Suma de cuadrados diferencias/N			5,26E-03	Suma de cuadrados diferencias/N			2,56E-02
	Repetibilidad			0,072548834	Repetibilidad			0,159942611

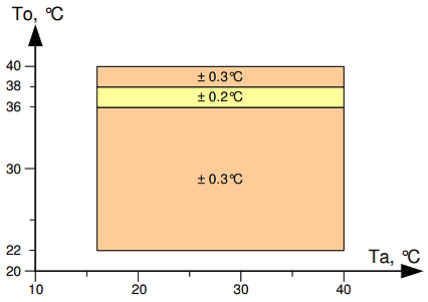
Para calibrar la báscula se tomaron en cuenta las normas ASTM E74-18e1, INEN 1 502 e ISO 9001, los resultados corresponden a una balanza de clase III (precisión media), que se puede usar en entornos médicos para aplicaciones como cálculo de índice de grasa corporal, seguimiento de progresión de embarazos y seguimiento a pacientes con insuficiencia cardiaca, como referencia en el mercado se encuentran a la venta básculas mecánicas de columna con capacidad de 200 Kg y precisión de 50g.

Resultados de la implementación del Termómetro infrarrojo

Los sensores de la familia MLX 90614 varían su precisión dependiendo de la temperatura ambiente (Ta) y de la temperatura del objeto al que se apunta el sensor (To), las versiones MLX90614 DXX están diseñadas para aplicaciones médicas y tienen una mayor precisión entre 16°C y 40°C para Ta y entre 22°C y 40°C para To.

Figura 215

Precisión médica sensores versión MLX90614 DXX



Según reportes del portal de internet Freemeteo la temperatura ambiente promedio de la parroquia de Antonio José Holguín es de 12°C durante lo que va del año 2022, la temperatura rara vez pasa de 23°C o baja de 6°C, pero, aunque sean pocos, se tienen registros de temperaturas de hasta 0°C. Es evidente que las bajas temperaturas terminarán por influir en el desempeño del sensor, a continuación, se muestran los resultados obtenidos, cabe mencionar que se aplicaron los métodos para mejorar el rendimiento del sensor que el fabricante sugiere, por ejemplo, incrementar la ganancia del sensor mediante un amplificador operacional o minimizar la dependencia entre las lecturas y el voltaje de alimentación.

Tabla 19

Valores de Ta y To para cálculo de error

	Sensor Infrarrojo con MLX90614			
Medida	**Fecha**	**Hora**	**Ta (°C)**	**To (°C)**
1	07/11/2022	6:00	5	36,84
2	07/11/2022	7:00	8	37,12
3	07/11/2022	8:00	11	36,96
4	07/11/2022	9:00	13	37,06
5	07/11/2022	10:00	15	37,11
6	07/11/2022	11:00	17	37,02
7	07/11/2022	12:00	19	37
8	07/11/2022	13:00	19	36,98
9	07/11/2022	17:00	13	37,05
10	07/11/2022	18:00	12	37,12
11	07/11/2022	21:00	11	36,97
12	07/11/2022	22:00	10	37,08
13	08/11/2022	6:00	9	36,94
14	08/11/2022	8:00	12	36,96
15	08/11/2022	9:00	14	37,12
16	08/11/2022	17:00	12	37,07
17	08/11/2022	21:00	10	36,97
18	09/11/2022	6:00	7	37,22
19	09/11/2022	7:00	8	37,11
20	09/11/2022	8:00	10	37,08
21	09/11/2022	9:00	13	37,03
22	09/11/2022	20:00	12	36,95
23	10/11/2022	6:00	7	36,87
24	10/11/2022	7:00	8	36,93
25	10/11/2022	8:00	11	36,88
26	10/11/2022	9:00	13	37,06
27	10/11/2022	20:00	12	37,08
28	10/11/2022	22:00	11	37,12
29	11/11/2022	6:00	9	36,93
30	11/11/2022	7:00	9	36,91

Nota. La tabla se elabora midiendo la temperatura de un objeto a 37 °C a diferentes horas del día durante 5 días, la temperatura ambiente de la parroquia casi siempre se encuentra dentro del rango de precisión médica del sensor desde las 11:00 am hasta las 16:00 pm, la mayoría de estos datos no se toman en cuenta.

Cálculo de errores del termómetro infrarrojo con MLX90614

Valor promedio:

$$\overline{To} = \frac{\sum_{i=1}^{n} To_i}{n}$$

$$\overline{To} = 37.02\ °C$$

Erro absoluto:

$$Ea_{To} = \frac{\sum_{i=1}^{n} |To_i - \overline{To}|}{n}$$

$$Ea_{To} = 0{,}0778\ °C$$

Error relativo:

$$Er_{To} = \frac{Ea_{To}}{\overline{To}}$$

$$Er_{To} = 0{,}0021$$

Error porcentual:

$$Ep_{To} = Er_{To} * 100$$

$$Ep_{To} = \ 0{,}211\ \%$$

La precisión del sensor es de $To \pm 0.211\%$

Los resultados se pueden comparar con un termómetro infrarrojo comercial.

Figura 216

Termómetro infrarrojo NX-200

Tabla 20

Valores de Ta y To tomados con termómetro infrarrojo comercial

	Termómetro Infrarrojo comercial			
Medida	**Fecha**	**Hora**	**Ta (°C)**	**To (°C)**
1	07/11/2022	6:00	5	36,8
2	07/11/2022	7:00	8	37,2
3	07/11/2022	8:00	11	36,9
4	07/11/2022	9:00	13	37,1
5	07/11/2022	10:00	15	37
6	07/11/2022	11:00	17	36,8
7	07/11/2022	12:00	19	37
8	07/11/2022	13:00	19	37,4
9	07/11/2022	17:00	13	37,3
10	07/11/2022	18:00	12	36,8
11	07/11/2022	21:00	11	37,2
12	07/11/2022	22:00	10	36,8
13	08/11/2022	6:00	9	36,6
14	08/11/2022	8:00	12	37,2
15	08/11/2022	9:00	14	36,8
16	08/11/2022	17:00	12	37
17	08/11/2022	21:00	10	36,6
18	09/11/2022	6:00	7	37,2
19	09/11/2022	7:00	8	37,4
20	09/11/2022	8:00	10	36,8
21	09/11/2022	9:00	13	37
22	09/11/2022	20:00	12	36,7
23	10/11/2022	6:00	7	36,8
24	10/11/2022	7:00	8	36,9
25	10/11/2022	8:00	11	36,6
26	10/11/2022	9:00	13	37,4
27	10/11/2022	20:00	12	37,3
28	10/11/2022	22:00	11	37,2
29	11/11/2022	6:00	9	36,6
30	11/11/2022	7:00	9	36,8

Cálculo de errores del termómetro infrarrojo comercial

Valor promedio:

$$\overline{To} = \frac{\sum_{i=1}^{n} To_i}{n}$$

$$\overline{To} = \ 36{,}97\ °C$$

Erro absoluto:

$$Ea_{To} = \frac{\sum_{i=1}^{n} |To_i - \overline{To}|}{n}$$

$$Ea_{To} = 0{,}2222\ °C$$

Error relativo:

$$Er_{To} = \frac{Ea_{To}}{\overline{To}}$$

$$Er_{To} = 0{,}00601\ °C$$

Error porcentual:

$$Ep_{To} = Er_{To} * 100$$

$$Ep_{To} = \ 0{,}601\ \%$$

La precisión del termómetro infrarrojo comercial es de $To \pm 0.601\%$

Cálculo de repetibilidad de termómetro con MLX90614

Tabla 21

Datos para cálculo de repetibilidad MLX90614

Repetibilidad termómetro con MLX90614				
ítem	**Temp (°C)**	**Indicación**	**Diferencia**	**Cuadrado diferencia**
1	16	16,02	-0,02	4,00E-04
2	17	16,97	0,03	9,00E-04
3	18	17,99	0,01	1,00E-04
4	19	19,05	-0,05	2,50E-03
5	20	19,97	0,03	9,00E-04
6	21	20,96	0,04	1,60E-03
7	22	21,95	0,05	2,50E-03
8	23	23	0	0,00E+00
9	24	24,03	-0,03	9,00E-04
10	25	24,99	0,01	1,00E-04
11	26	26,01	-0,01	1,00E-04
12	27	27,02	-0,02	4,00E-04
13	28	27,98	0,02	4,00E-04
14	29	29	0	0,00E+00
15	30	30,01	-0,01	1,00E-04
16	31	31,08	-0,08	6,40E-03
17	32	32,1	-0,1	1,00E-02
18	33	32,94	0,06	3,60E-03
19	34	34,06	-0,06	3,60E-03
20	35	35,09	-0,09	8,10E-03
21	36	36	0	0,00E+00
22	37	37,01	-0,01	1,00E-04
23	38	37,93	0,07	4,90E-03
24	39	39,08	-0,08	6,40E-03
25	40	40,1	-0,1	1,00E-02
26	41	41,12	-0,12	1,44E-02
27	42	41,86	0,14	1,96E-02
28	43	43,08	-0,08	6,40E-03
29	44	44,13	-0,13	1,69E-02
30	45	45,14	-0,14	1,96E-02
Suma de cuadrados de las diferencias				1,41E-01
Suma de cuadrados de las diferencias/N				4,70E-03
Repetibilidad				**0,068532231**

Figura 217

Lectura de temperatura corporal vs termómetro comercial

Figura 218

Lecturas de temperatura corporal vs termómetro comercial

Pruebas del sensor de temperatura en pacientes. Se realizan varias pruebas en bebés, luego de las vacunas de los dos, cuatro y seis meses es común que presenten fiebre unas cuantas horas después de la vacuna, aunque la fiebre es ligera puede ocasionar estrés y preocupación en madres y padres primerizos. También se realizaron pruebas en la escuela de la parroquia que se encuentra frente al centro de salud por los incrementos de infecciones respiratorias a finales de noviembre y principios de diciembre del año 2022. Todos los valores de temperatura registrados se comprueban con un termómetro comercial.

Figura 219

Toma de datos biométricos bebé de dos meses de edad

Nota. La fiebre comúnmente se presenta de 6 a 7 horas luego de la vacuna y puede durar entre 24 a 48 horas.

Figura 220

Toma de temperatura en el hogar del paciente el mismo día a las 10:00 pm

Figura 221

Toma de temperatura con termómetro infrarrojo comercial

Figura 222

Toma de temperatura en bebé de 6 meses con tos leve

Figura 223

Toma de temperatura en aulas con casos de infecciones respiratorias

Resultados de la implementación del Oxímetro de pulso

Para calibrar y posteriormente validar el oxímetro de pulso se usa como referencia un oxímetro comercial, los resultados se evalúan mediante el método de Bland-Altman para evaluar la concordancia entre ambos métodos, con 25 observaciones cada uno. Los porcentajes normales de saturación de oxígeno en la sangre suelen estar por encima del 95%, para obtener valores por debajo del rango normal fue necesario buscar pacientes con infecciones respiratorias, un caso particular fue de gran ayuda, se trató de un paciente ex-fumador, hipertenso y con diabetes tipo 2 de 80 años que tenía pulmonía debido a las secuelas de la enfermedad COVID-19 que había contraído con anterioridad, su porcentaje de saturación en la sangre se encontraba entre el 60% y 70%, con oxígeno suplementario a 2 L/min el oxímetro marcaba entre 70% y 80%, con el pasar de los días empezó a marcar entre 80% y 90% con 1 L/min de oxígeno suplementario, luego de un tratamiento con bromuro de ipratropio el porcentaje de oxígeno en la sangre del paciente se normalizó.

Tabla 22

Tabla de datos para método de Bland-Altman y cálculo de errores del oxímetro

Ítem	Método A LMT-01 %SPO2 (%)	Método B MAX 30102 %SPO2 (%)	Media	Diferencia	Error Absoluto	Error Relativo	Error Porcentual
1	60	62	61	-2	2	0,03333	3,333
2	62	63	62,5	-1	1	0,01612	1,612
3	64	65	64,5	-1	1	0,01562	1,562
4	66	66	66	0	0	0	0
5	68	65	66,5	3	3	0,04411	4,411
6	70	71	70,5	-1	1	0,01428	1,428
7	72	72	72	0	0	0	0
8	74	71	72,5	3	3	0,04054	4,054
9	76	76	76	0	0	0	0
10	78	76	77	2	2	0,02564	2,564
11	80	81	80,5	-1	1	0,0125	1,25
12	82	83	82,5	-1	1	0,01219	1,219
13	84	84	84	0	0	0	0
14	86	86	86	0	0	0	0
15	88	87	87,5	1	1	0,01136	1,136
16	90	89	89,5	1	1	0,01111	1,111
17	92	93	92,5	-1	1	0,01086	1,086
18	93	93	93	0	0	0	0
19	94	94	94	0	0	0	0
20	95	95	95	0	0	0	0
21	96	96	96	0	0	0	0
22	97	96	96,5	1	1	0,01030	1,030
23	98	98	98	0	0	0	0
24	99	98	98,5	1	1	0,01010	1,010
25	100	100	100	0	0	0	0
Media (d)				0,16		Promedio	1.07
desviación estándar (s)				1,21380943			

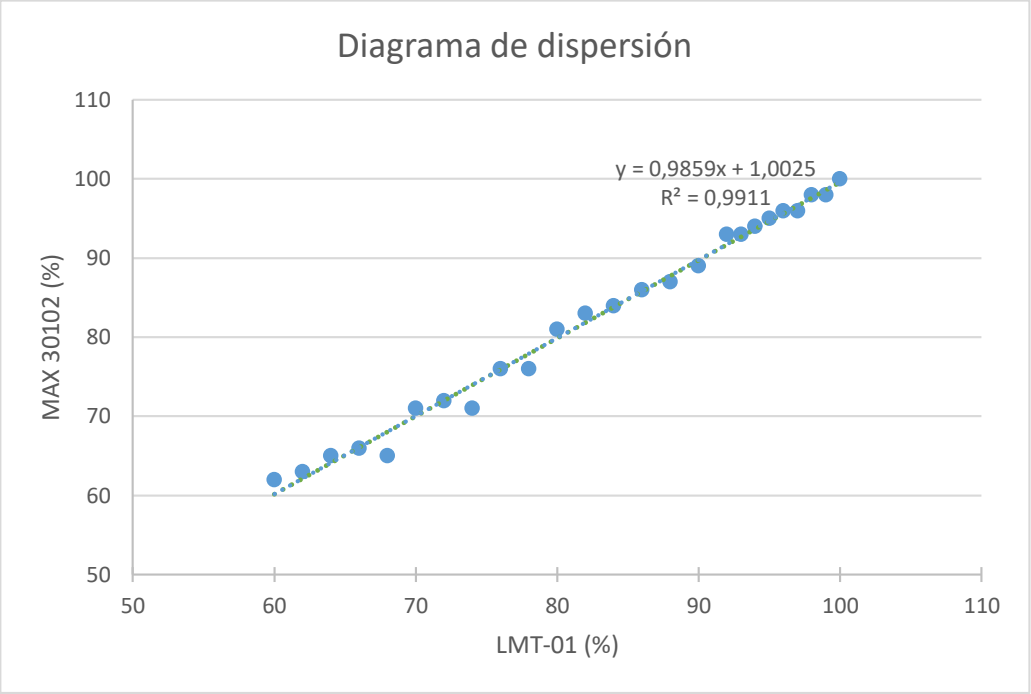
El promedio de los valores en la columna diferencia resulta ser 0.16

La desviación estándar de los valores en la columna Diferencia es 1.2138

Las observaciones del oxímetro de pulso se pueden inspeccionar mediante su representación en un diagrama de dispersión.

Figura 224

Diagrama de dispersión métodos Ay B



Del diagrama de dispersión se obtienen los siguientes resultados:

- Regresión lineal: y=0.9589x+1.0025
- Intersección: 1.0025
- Coeficiente de correlación de Pearson: 0,9955
- Coeficiente de determinación: 0.9911

El coeficiente de correlación de Pearson muestra una fuerte relación positiva entre los valores de ambos oxímetros de pulso. Aunque tengan una alta correlación no significa que exista concordancia entre ambos métodos ya que no se toma en cuenta la diferencia entre los valores, la pendiente de la regresión lineal indica que los valores del Método A sobreestiman al método B, además, al no encontrarse los valores sobre la bisectriz en el diagrama de dispersión se interpreta que existe discrepancia entre ambos métodos.

Los límites superior e inferior del intervalo de confianza para la diferencia promedio se calculan como:

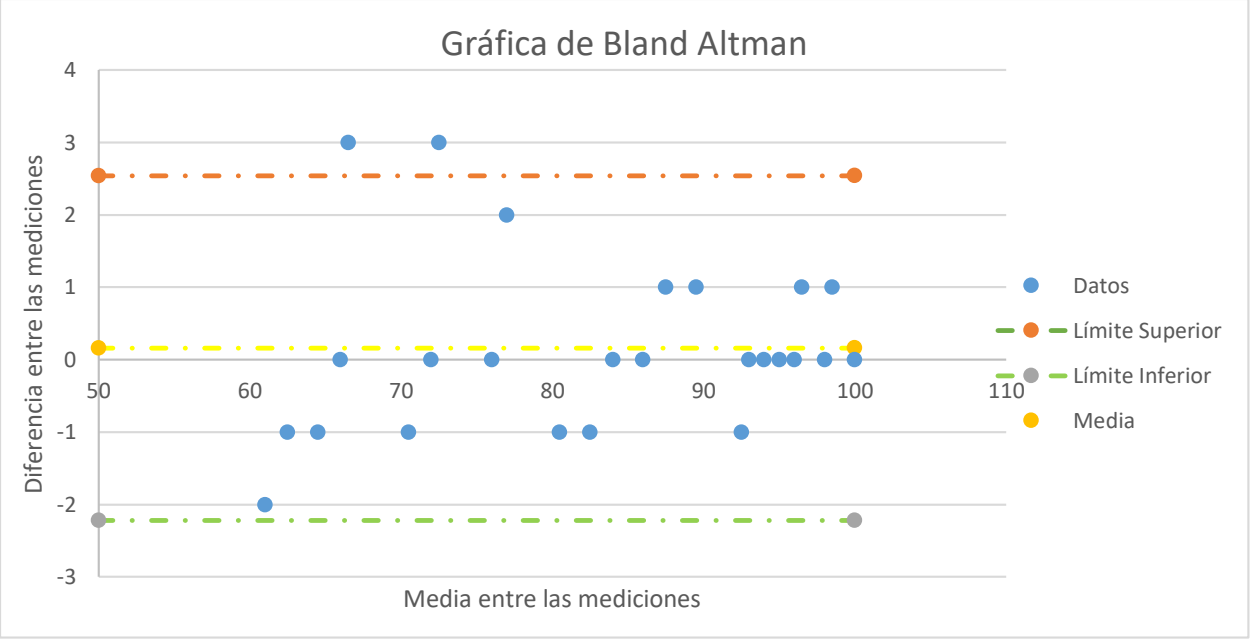
Límite Superior: $\bar{d} + 1.96 * s = 0.16 + 1.96 * 1.2138 = 2.539$

Límite inferior: $\bar{d} - 1.96 * s = 0.16 - 1.96 * 1.2138 = -2.219$

Estos valores se visualizan en el diagrama de Bland-Altman.

Figura 225

Gráfica de Bland-Altman para oxímetro de pulso



Los resultados de la regresión lineal muestran según su pendiente que existe una buena correlación entre ambos métodos y que el oxímetro comercial sobreestima al oxímetro con módulo Max30102, además, es muy similar al valor de exactitud igual a %SpO2 ±1.072% calculado por medio del error porcentual. De la gráfica de Bland-Altman se puede concluir que el 92% de los valores se encuentran dentro de los límites de concordancia, en promedio el método A mide 0.16 unidades más que el método B, mientras que los datos se muestran más dispersos para valores menores al 80%. Por lo que se puede concluir que existe concordancia entre los valores medidos con el método A y el método B.

Figura 226

Monitor de signos vitales con pedestal

Figura 227

Monitoreo de SpO2 en el centro de salud y en el hogar de la paciente

Resultados de la implementación del Tensiómetro de muñeca

Un tensiómetro que ocupa el método oscilométrico no mide directamente la presión arterial, sino que la estima a partir de la presión del brazalete, esta presión pasa por filtro pasa-banda, tomando en cuenta que los pulsos por minuto de una persona varían de 60 a 180, por lo que si se calcula la frecuencia se obtienen valores de 1 Hz a 3 Hz, para garantizar un buen resultado se puede ampliar el rango de frecuencias de 0.8 Hz a 4 Hz. Con estos mismos valores se puede calcular la frecuencia de muestreo, según el teorema de muestreo debe ser por lo menos 2 veces la frecuencia de la señal, tomando un factor de seguridad de 10 se obtiene una frecuencia de muestreo de 30Hz.

A la salida del filtro se obtiene la presión oscilatoria que permite obtener los valores de presión sistólica, presión diastólica y frecuencia cardiaca del paciente por lo que estos valores se relacionan entre sí, pero, en la tabla estos datos no corresponden a la misma muestra, es decir, la presión sistólica, diastólica y la presión de una misma fila son de tomas distintas y a distintos pacientes y se correlacionan únicamente a la medida respectiva tomada con el tensiómetro comercial, se han organizado los valores de este modo para facilitar la construcción y análisis de las gráficas.

Figura 228

Presentación de parámetros biométricos del tensiómetro

Tabla 23

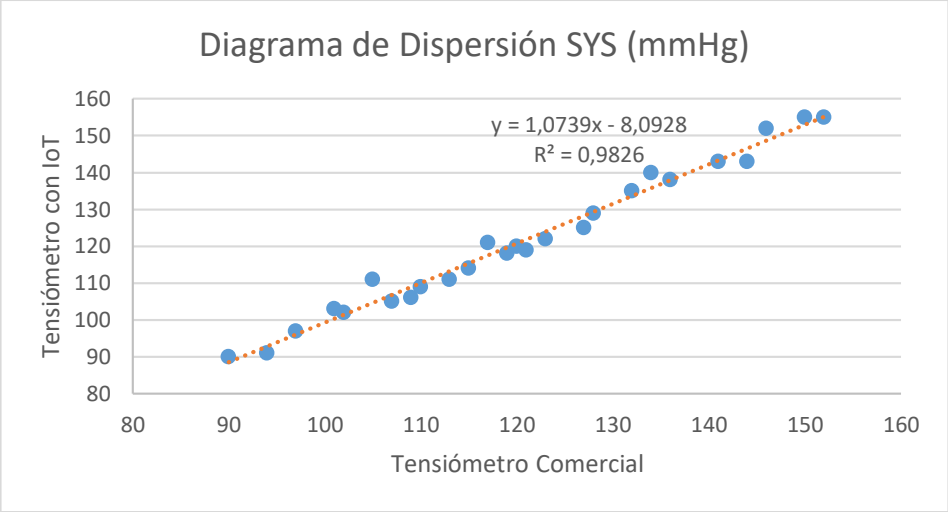
Tabla de datos para método de Bland-Altman y cálculo de errores del tensiómetro

	Método A Tensiómetro comercial		Método B Tensiómetro IoT		SYS				DIA			
Ítem	SYS (mmHg)	DIA (mmHg)	SYS (mmHg)	DIA (mmHg)	Media	Diferencia	Error Absoluto	Error Porcentual	Media	Diferencia	Error Absoluto	Error Porcentual
1	90	61	90	62	90	0	0	0,00	61,5	-1	1	1,64
2	94	62	91	62	92,5	3	3	3,19	62	0	0	0,00
3	97	64	97	63	97	0	0	0,00	63,5	1	1	1,56
4	101	65	103	70	102	-2	2	1,98	67,5	-5	5	7,69
5	102	68	102	69	102	0	0	0,00	68,5	-1	1	1,47
6	105	69	111	68	108	-6	6	5,71	68,5	1	1	1,45
7	107	70	105	72	106	2	2	1,87	71	-2	2	2,86
8	109	72	106	72	107,5	3	3	2,75	72	0	0	0,00
9	110	74	109	79	109,5	1	1	0,91	76,5	-5	5	6,76
10	113	75	111	78	112	2	2	1,77	76,5	-3	3	4,00
11	115	76	114	80	114,5	1	1	0,87	78	-4	4	5,26
12	117	80	121	82	119	-4	4	3,42	81	-2	2	2,50
13	119	81	118	81	118,5	1	1	0,84	81	0	0	0,00
14	120	83	120	86	120	0	0	0,00	84,5	-3	3	3,61
15	121	86	119	84	120	2	2	1,65	85	2	2	2,33
16	123	87	122	82	122,5	1	1	0,81	84,5	5	5	5,75
17	127	88	125	85	126	2	2	1,57	86,5	3	3	3,41
18	128	89	129	87	128,5	-1	1	0,78	88	2	2	2,25
19	132	91	135	94	133,5	-3	3	2,27	92,5	-3	3	3,30
20	134	96	140	96	137	-6	6	4,48	96	0	0	0,00
21	136	97	138	97	137	-2	2	1,47	97	0	0	0,00
22	141	98	143	95	142	-2	2	1,42	96,5	3	3	3,06
23	144	100	143	98	143,5	1	1	0,69	99	2	2	2,00
24	146	102	152	105	149	-6	6	4,11	103,5	-3	3	2,94
25	150	105	155	100	152,5	-5	5	3,33	102,5	5	5	4,76
26	152	106	155	103	153,5	-3	3	1,97	104,5	3	3	2,83
						Promedio	2,27	1,84	Promedio		2,27	2,75
			Media (d)			-0,81	Media (d)			-0,19		
			Desviación estándar (s)			2,84	desviación estándar (s)			2,84		

Las observaciones del monitor de presión arterial se pueden inspeccionar mediante su representación en un diagrama de dispersión.

Figura 229

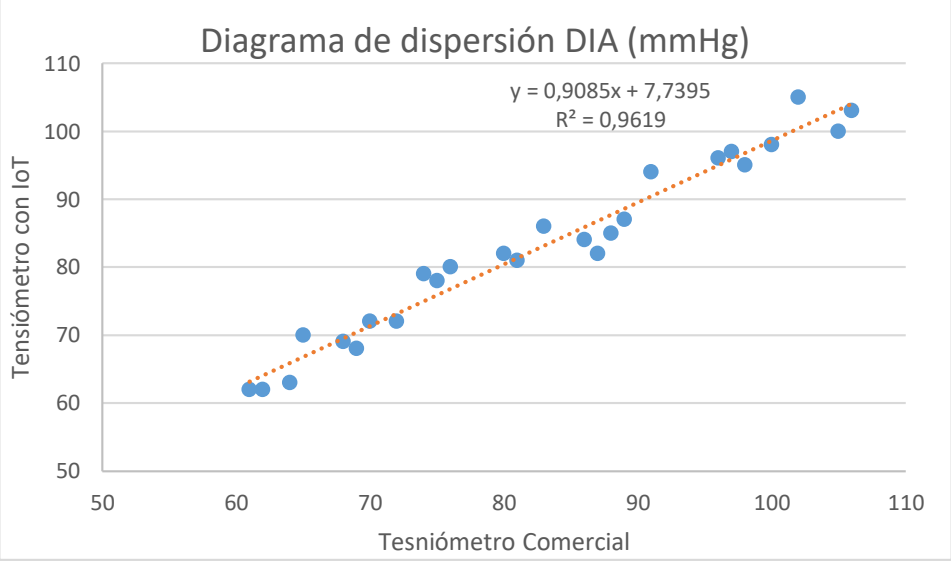
Gráfica de dispersión presión sistólica



Regresión lineal: y=1.0739x-8.0928
Coeficiente de Pearson=0.9912

Figura 230

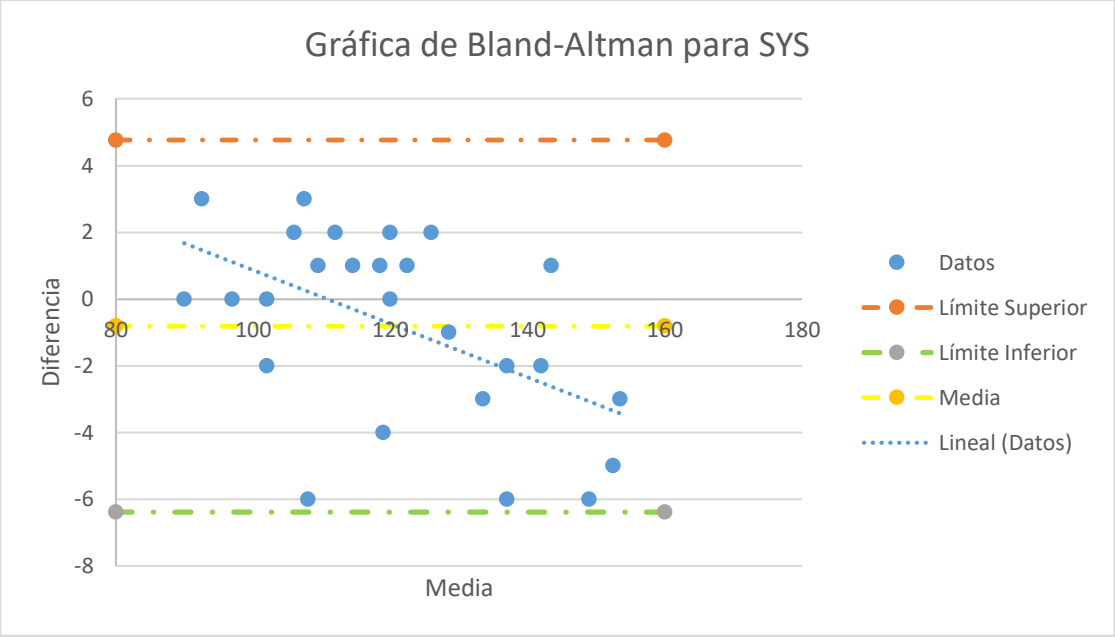
Gráfica de dispersión presión diastólica



Regresión lineal: y=0.9085x+7.7395
Coeficiente de Pearson=0.9807

Figura 231

Gráfica de Bland-Altman para presión sistólica

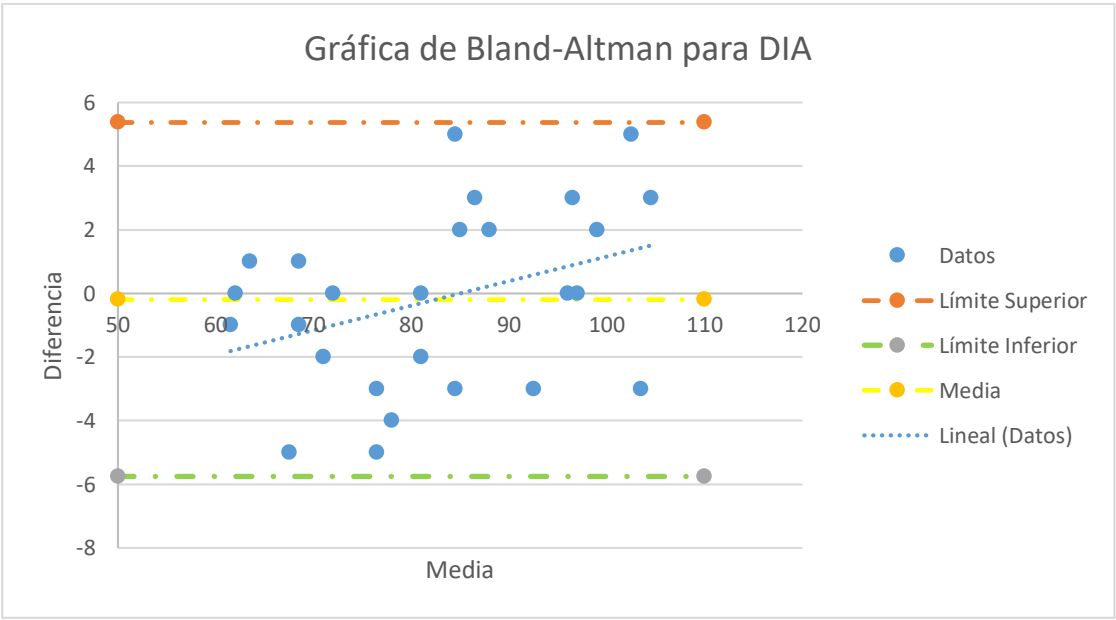


Límite Superior: $\bar{d} + 1.96 * s = -0.81 + 1.96 * 2.84 = 4.764$

Límite inferior: $\bar{d} - 1.96 * s = -0.81 - 1.96 * 2.84 = -6.379$

Figura 232

Gráfica de Bland-Altman para presión diastólica



Límite Superior: $\bar{d} + 1.96 * s = -0.19 + 1.96 * 2.84 = 5.379$

Límite inferior: $\bar{d} - 1.96 * s = -0.19 - 1.96 * 2.84 = -5.764$

El diagrama de dispersión indica que en ambos casos existe una fuerte correlación positiva entre los métodos A y B, se comprueba calculando el coeficiente de Pearson que es ligeramente mayor para la presión sistólica cuyos datos se encuentran menos dispersos lo que apunta a que la ecuación de regresión modela los datos con mayor precisión. Por la pendiente que arroja la regresión lineal de la presión sistólica se deduce que el tensiómetro IoT tiende a dar valores ligeramente mayores a los que arroja el tensiómetro comercial, mientras que para la presión diastólica es lo contrario, arroja valores ligeramente menores.

Las gráficas de Bland-Altman muestran valores dispersos en todo el rango de valores de presión sistólica y diastólica, como era de esperar, al ser un método de estimación. Por otro lado, todos los valores se encuentran dentro de los límites de concordancia, es decir, dentro del rango de confianza del 95%.

Como los valores se encuentran muy dispersos se agregó una línea de regresión a las gráficas de Bland-Altman que denota un bias no constante, en el caso de la presión sistólica el bias varía proporcionalmente con una tendencia negativa de las diferencias conforme aumenta la magnitud de la variable medida, en el caso de la presión diastólica se observa un bias no constante que varía proporcionalmente con una tendencia positiva conforme aumenta la magnitud de la variable medida.

El error absoluto es exactamente igual para ambos valores de presión, el error porcentual es mayor para la presión diastólica, lo que comprueba las conclusiones sacadas a partir de la gráfica de dispersión.

Exactitud de la presión sistólica: $SYS \pm 1.84\%$

Exactitud de la presión diastólica: $DIA \pm 2.75\%$

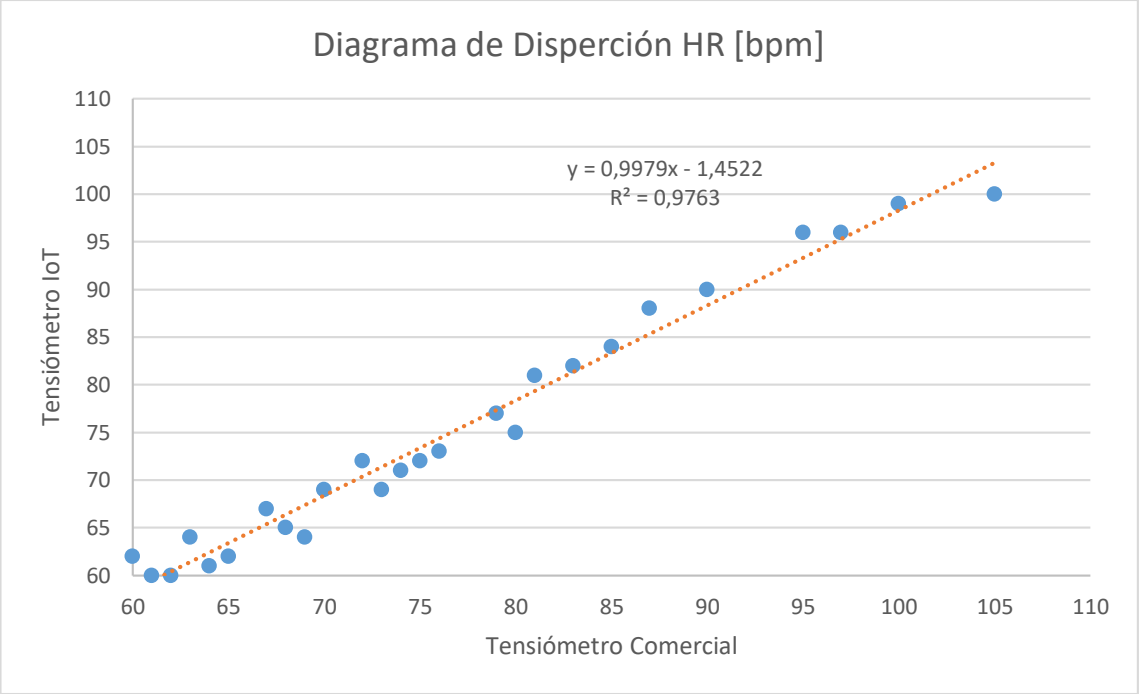
Tabla 24

Datos para método de Bland-Altman y cálculo de errores de frecuencia cardiaca

Ítem	Método A Tensiómetro Comercial HR [bpm]	Método B Tensiómetro IoT HR [bpm]	Media	Diferencia	Error Absoluto	Error Relativo	Error Porcentual
1	60	62	61	-2	2	0,0333	3,3333
2	61	60	60,5	1	1	0,0163	1,6393
3	62	60	61	2	2	0,0322	3,2258
4	63	64	63,5	-1	1	0,0158	1,5873
5	64	61	62,5	3	3	0,0468	4,6875
6	65	62	63,5	3	3	0,0461	4,6153
7	67	67	67	0	0	0	0
8	68	65	66,5	3	3	0,0441	4,4117
9	69	64	66,5	5	5	0,0724	7,2463
10	70	69	69,5	1	1	0,0142	1,4285
11	72	72	72	0	0	0	0
12	73	69	71	4	4	0,0547	5,4794
13	74	71	72,5	3	3	0,0405	4,0540
14	75	72	73,5	3	3	0,04	4
15	76	73	74,5	3	3	0,0394	3,9473
16	79	77	78	2	2	0,0253	2,5316
17	80	75	77,5	5	5	0,0625	6,25
18	81	81	81	0	0	0	0
19	83	82	82,5	1	1	0,0120	1,2048
20	85	84	84,5	1	1	0,0117	1,1764
21	87	88	87,5	-1	1	0,0114	1,1494
22	90	90	90	0	0	0	0
23	95	96	95,5	-1	1	0,0105	1,0526
24	97	96	96,5	1	1	0,0103	1,0309
25	100	99	99,5	1	1	0,01	1
26	105	100	102,5	5	5	0,0476	4,7619
				Promedio	2	0,0268	2,6851
Media (d)				1,61538			
desviación estándar (s)				1,98145			

Figura 233

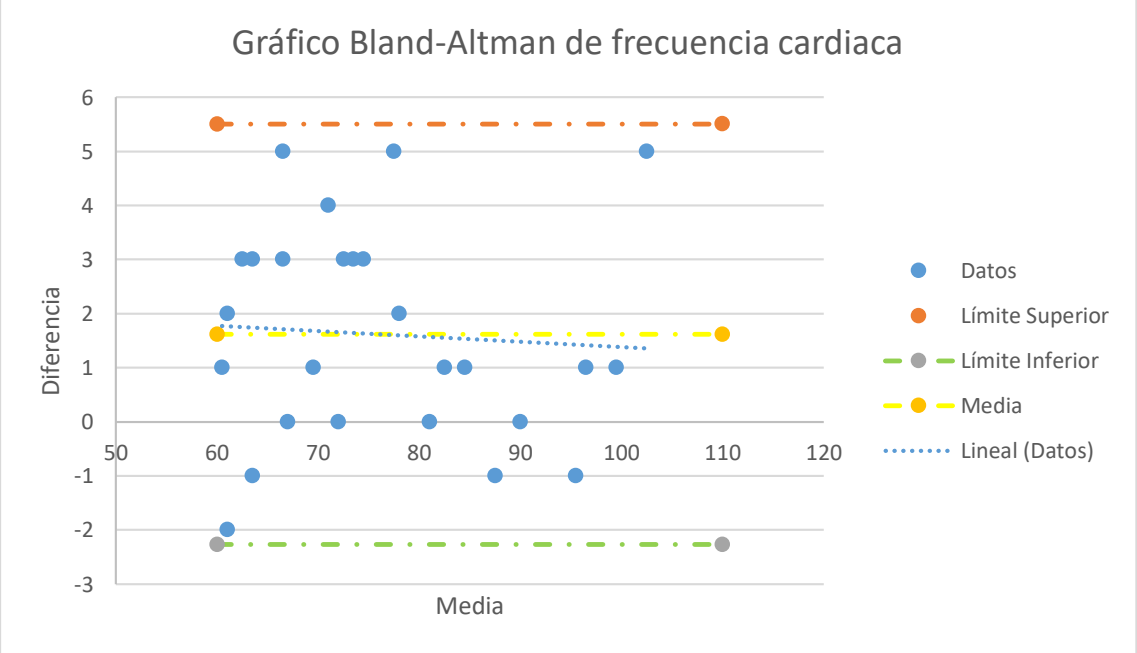
Gráfica de dispersión de frecuencia cardiaca



Coeficiente de Pearson=0.988

Figura 234

Gráfica de Bland-Altman de frecuencia cardiaca



Los límites superior e inferior del intervalo de confianza para la diferencia promedio se calculan como:

Límite Superior: $\bar{d} + 1.96 * s = 1.6153 + 1.96 * 1.9814 = 5.498$

Límite inferior: $\bar{d} - 1.96 * s = 1.6153 - 1.96 * 1.9814 = -2.268$

El diagrama de dispersión de la frecuencia cardíaca indica una fuerte relación positiva entre ambos métodos, se comprueba a través del coeficiente de correlación de Pearson que la estimación es casi tan buena como en el caso de la presión sistólica. El valor de la media de las diferencias señala que el método A sobrestima al método B, es decir el tensiómetro comercial da valores ligeramente mayores, por otro lado, todas las muestras se encuentran dentro de los límites de concordancia, la línea de regresión de la gráfica de Bland-Altman muestra un bias casi constante, pero, que en realidad varía proporcionalmente con una tendencia negativa conforme aumenta la frecuencia cardiaca.

Figura 235

Toma de presión arterial por método auscultatorio en el Centro de Salud

Figura 236

Toma de presión arterial por método oscilométrico en el hogar del paciente

Resultados de la implementación del Glucómetro

Aunque la mayoría de muestras fueron tomadas a pacientes del centro de salud no se obtuvieron valores en todo el rango de manera que fue necesario recurrir nuevamente a la solución de glucosa 100mM y agua destilada, los valores de glucosa de cada muestra se compararon con un glucómetro comercial.

Para prevenir problemas a la hora de tomar la muestra es necesario que los pacientes se laven las manos antes de pincharse el dedo, la muestra se puede tomar en cualquier dedo, tomar la muestra en diferentes dedos evita que aparezcan cayos, se recomienda no pinchar en el centro de la yema sino en un costado debido a que esta zona tiene con mayor disposición de vasos sanguíneos y la gota se obtiene con mayor facilidad, además, tiene menos nervios por lo que el proceso es menos doloroso. Tomando en cuenta estas consideraciones se obtuvieron los siguientes resultados.

Tabla 25

Datos para método de Bland-Altman y cálculo de errores del glucómetro

	Método A Glucómetro Safe-Accu	Método B Glucómetro IoT					
Ítem	**mg/dl**	**mg/dl**	**Media**	**Diferencia**	**Error Absoluto**	**Error Relativo**	**Error Porcentual**
1	65	68	66,5	-3	3	0,0461	4,6153
2	67	70	68,5	-3	3	0,0447	4,4776
3	76	75	75,5	1	1	0,0131	1,3157
4	80	80	80	0	0	0	0
5	86	87	86,5	-1	1	0,0116	1,1627
6	100	98	99	2	2	0,02	2
7	110	107	108,5	3	3	0,0272	2,7272
8	112	114	113	-2	2	0,0178	1,7857
9	116	112	114	4	4	0,0344	3,4482
10	121	124	122,5	-3	3	0,0247	2,4793
11	124	122	123	2	2	0,0161	1,6129
12	138	140	139	-2	2	0,0144	1,4492
13	147	145	146	2	2	0,0136	1,3605
14	150	153	151,5	-3	3	0,02	2
15	158	160	159	-2	2	0,0126	1,2658
16	161	164	162,5	-3	3	0,0186	1,8633
17	170	169	169,5	1	1	0,0058	0,5882
18	173	171	172	2	2	0,0115	1,1560
19	182	183	182,5	-1	1	0,0054	0,5494
20	201	197	199	4	4	0,0199	1,9900
				Promedio	2,2	0,0189	1,8923
Media (d)				-0,1			
desviación estándar (s)				2,4899			

Tabla 26

Gráfica de dispersión de concentración de glucosa en sangre

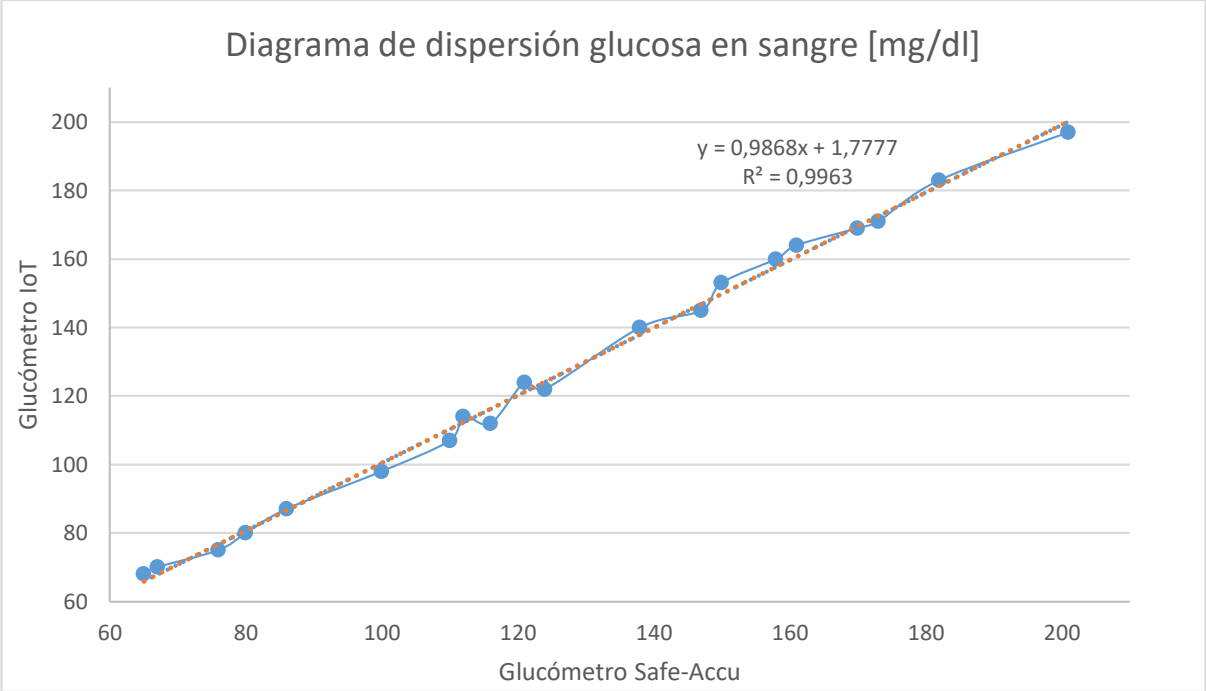
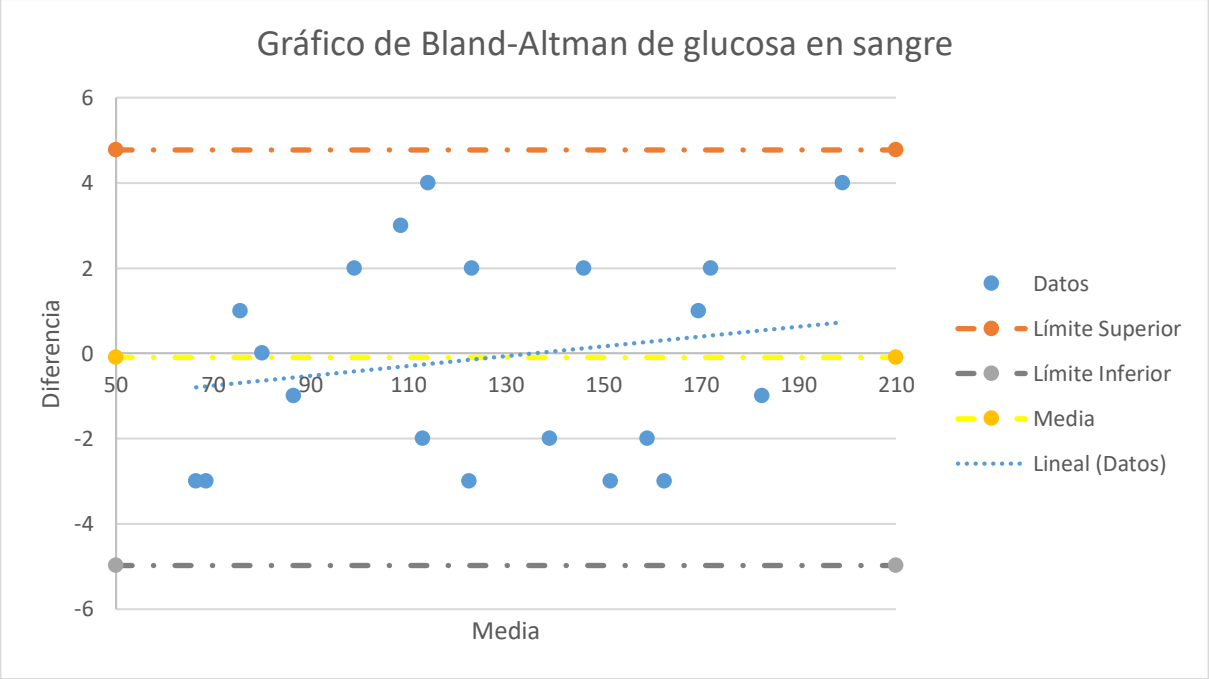


Tabla 27

Gráfica de Bland-Altman de concentración de glucosa en sangre



Los límites superior e inferior para un intervalo de confianza del 95% se calculan como:

Límite Superior: $\bar{d} + 1.96 * s = -0.1 + 1.96 * 2.4899 = 4.7803$

Límite inferior: $\bar{d} - 1.96 * s = -0.1 - 1.96 * 2.4899 = -4.9803$

Del diagrama de dispersión de la concentración de glucosa en sangre se infiere que ambos dispositivos mantienen una fuerte relación positiva debido a que el coeficiente de correlación de Pearson es casi igual a 1, también se observa que los valores empiezan a estar más dispersos a partir de los 100 mg/dl. El bias entre métodos es de -0.1 unidades lo que significa que en promedio el método A arroja valores menores a los del método B, la gráfica de Bland-Altman muestra que todos los valores se encuentran dentro del rango de concordancia establecido, la línea de regresión lineal calculada para las diferencias denota un bias proporcional con tendencia positiva conforme aumenta la concentración de glucosa en sangre.

Figura 237

Tomada de concentración de glucosa en sangre

Resultados de la implementación de la plataforma

Se realiza una encuesta al personal y a los pacientes del centro de salud para evaluar el desempeño de la plataforma, la correcta selección de los parámetros biométricos a monitorizar, la coherencia entre las herramientas que dispone la plataforma y las necesidades del sector, la accesibilidad de la población a la plataforma, el interés que pone la población en su estado de salud física y si el sistema mejora o empeora el control de pacientes pertenecientes al grupo de atención prioritaria, además, es indispensable medir el impacto que produce la implementación de la plataforma en el criterio de los pacientes con respecto a la imagen del centro de salud , con respecto a la calidad de los servicios prestados en el centro de salud y con respecto a los sistemas de monitorización remota de pacientes.

Cálculo del tamaño de la muestra:

- $n =$ Tamaño de la muestra
- $N = 3000$ Tamaño de población
- $Z = 1.96$ Nivel de confianza del 95%
- $e = 0.1$ Margen de error del 10%
- $p = 0.5$ Probabilidad de que el evento ocurra $50\ \%$
- $q = 0.5$ Probabilidad de que el evento no ocurra $50\ \%$

$$n = \frac{NZ^2pq}{(N-1)e^2 + Z^2pq}$$

$$n = \frac{3000 * (1.96)^2 * 0.5 * 0.5}{(3000-1) * 0.1^2 + 1.96^2 * 0.5 * 0.5}$$

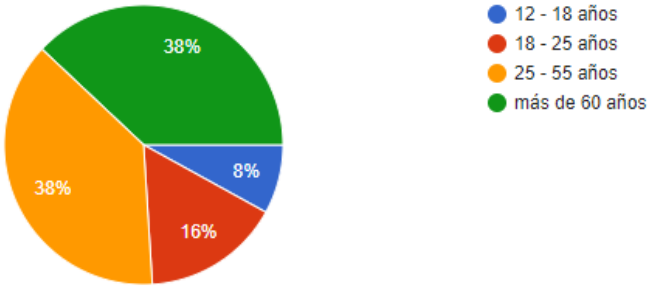
$$n = 93.09$$

Resultados de la encuesta:

Figura 238

Pregunta 1. ¿Cuántos años tiene?

Alternativas	Respuesta	Porcentaje
12-18 años	6	6.4 %
18-25 años	11	11.7 %
25-55 años	28	29.8 %
Más de 60 años	49	52.1 %

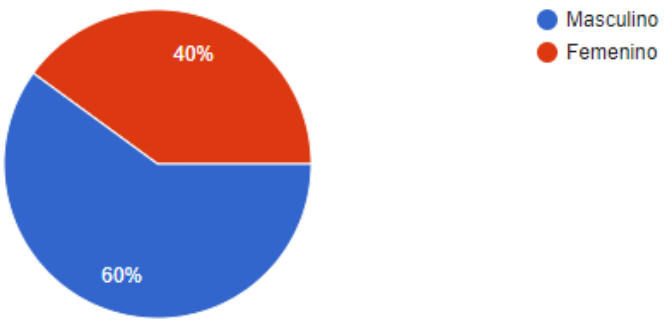


La mayoría de personas que acuden al centro de salud a monitorizar sus parámetros biométricos pasan de los 60 años, los pacientes de entre 25 a 50 años acuden únicamente cuando se enferman o porque tienen o tuvieron un hábito que deteriora la salud, los pacientes de entre 18 a 25 años solicitan exámenes médicos para aplicar a vacantes de empleo y los pacientes menores de 18 años rara acuden a chequeos médicos.

Figura 239

Pregunta 2. ¿Cuál es su género?

Alternativas	Respuesta	Porcentaje
Masculino	56	40.4 %
Femenino	38	59.6 %

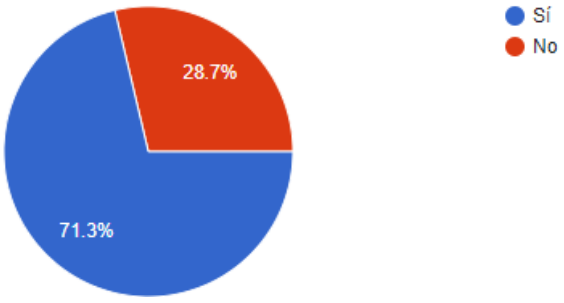


Se tomaron los parámetros biométricos de un número significativo de hombres y mujeres en vista de que el género de una persona es un factor de riesgo, en comparación con las mujeres los hombres suelen evitar ir al médico, pero la gráfica muestra lo contrario, esto se debe a que los hombres son más propensos a sufrir enfermedades crónicas por lo que acuden con más frecuencia a monitorizar sus parámetros biométricos.

Figura 240

Pregunta 3. ¿Padece alguna enfermedad crónica?

Alternativas	Respuesta	Porcentaje
Si	67	71.3 %
No	27	28.7 %

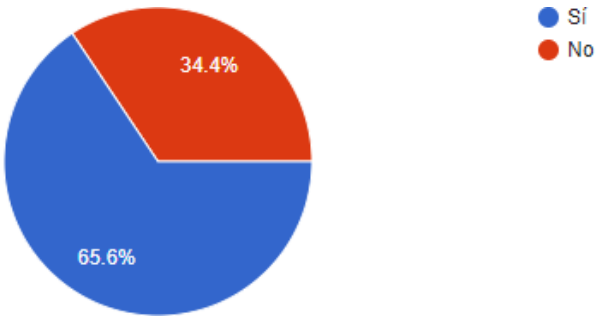


La mayoría de personas que acuden al centro de salud a monitorizar sus parámetros biométricos tienen enfermedades crónicas.

Figura 241

Pregunta 4. ¿Forma parte del grupo de atención prioritaria?

Alternativas	Respuesta	Porcentaje
Si	61	65.6 %
No	32	34.4 %

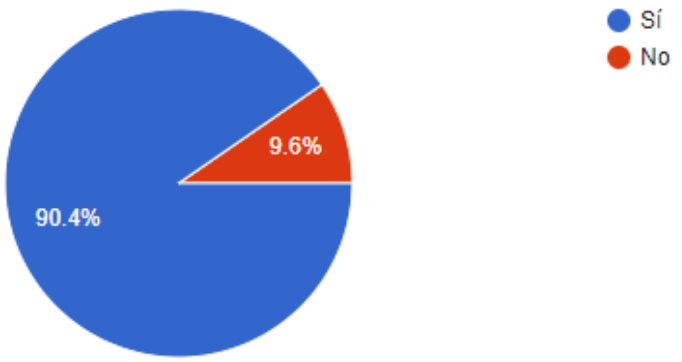


La mayoría de los pacientes del centro de salud pertenecen al grupo de atención prioritaria, esto se debe a que la agricultura y la ganadería son de las pocas actividades económicas que se realizan en la parroquia y se practica en su mayoría por miembros del seguro campesino, la población más joven y saludable se desplaza a las parroquias aledañas que tienen un mayor número de habitantes y actividad económica más variada.

Figura 242

Pregunta 5. ¿Tiene celular o computadora con conexión a internet?

Alternativas	Respuesta	Porcentaje
Si	85	90.4 %
No	9	9.6 %

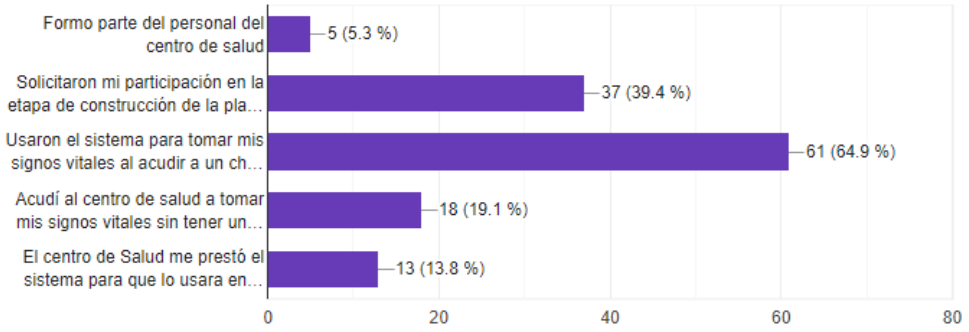


Casi todos los pacientes tienen medios para acceder a la plataforma, debido a la pandemia todas las familias hicieron un esfuerzo por contratar planes de internet y adquirir por lo menos una computadora o celular inteligente debido a que las escuelas, colegios y universidades adoptaron un modelo de educación virtual.

Figura 243

Pregunta 6. ¿Por qué motivo utilizó el sistema de monitorización remota?

Alternativas	Respuesta	Porcentaje
Formo parte del personal del centro de salud	5	5.3 %
Solicitaron mi participación en la etapa de construcción de la plataforma	37	39.4 %
Usaron el sistema para tomar mis signos vitales al acudir a un chequeo	61	64.9 %
Acudí al centro de salud a tomar mis signos vitales sin tener un chequeo programado	18	19.1 %
El centro de Salud me prestó el sistema para que lo usara en mi hogar	13	13.8 %

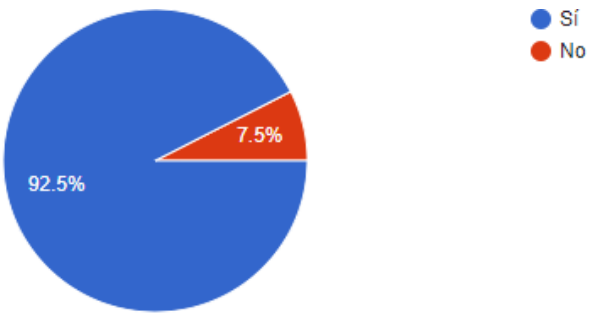


Las encuestas se aplicaron a pacientes que toman sus signos vitales regularmente, pacientes que acuden solo si su estado de salud desmejora, al personal del centro de salud y a algunas personas que colaboraron durante la etapa de desarrollo del prototipo.

Figura 244

Pregunta 7. ¿El acceso a la página web le resultó fácil?

Alternativas	Respuesta	Porcentaje
Si	86	92.5 %
No	7	7.5 %

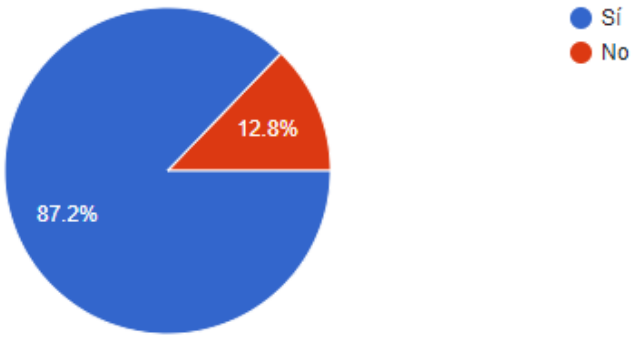


Los pacientes no experimentaron dificultades para ingresar a la plataforma, los gráficos y listas se despliegan bien tanto en celulares como en ordenadores.

Figura 245

Pregunta 8. ¿Ha interactuado con un Chatbot con anterioridad?

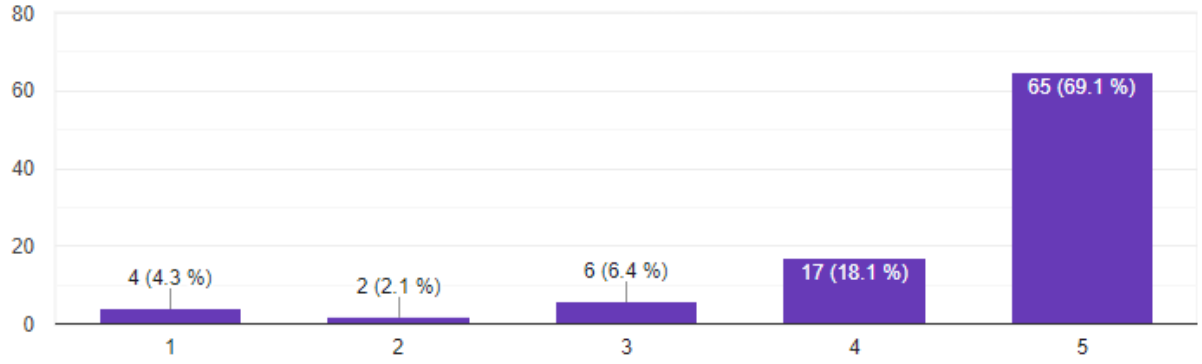
Alternativas	Respuesta	Porcentaje
Si	82	87.2 %
No	12	12.8 %



La mayoría de pacientes ya tiene experiencia interactuando con Chatbots.

Figura 246

Pregunta 9. ¿Cómo calificaría el Chatbot de la plataforma de monitorización remota?



El bot envía información de los sensores, respuestas a preguntas frecuentes e imágenes acordes a los temas consultados, como la mayoría de pacientes tienen malas experiencias tratando con bots de tiendas online, interactuar con un bot más sofisticado genera una impresión positiva de la plataforma.

Figura 247

Pregunta 10. ¿Recomendaría el uso de sistemas de monitorización remota a familiares o amigos basado en su experiencia?

Alternativas	Respuesta	Porcentaje
Si	88	93.6 %
No	6	6.4 %

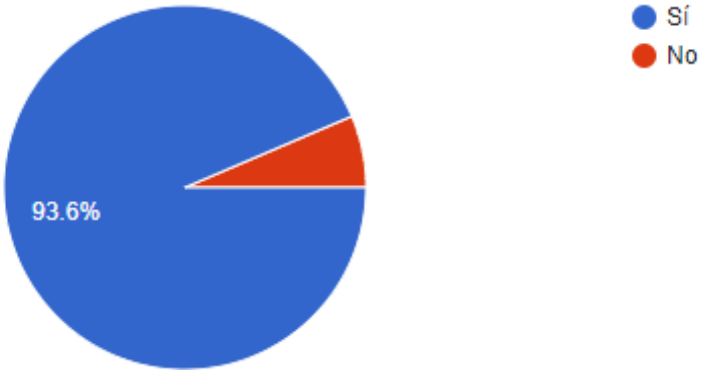
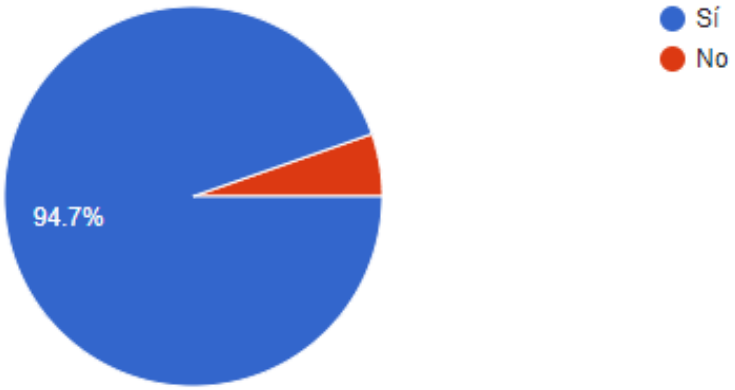


Figura 248

Pregunta 11. ¿Le pareció fácil o intuitivo el uso de los sensores de la plataforma?

Alternativas	Respuesta	Porcentaje
Si	89	94.7 %
No	5	5.3 %

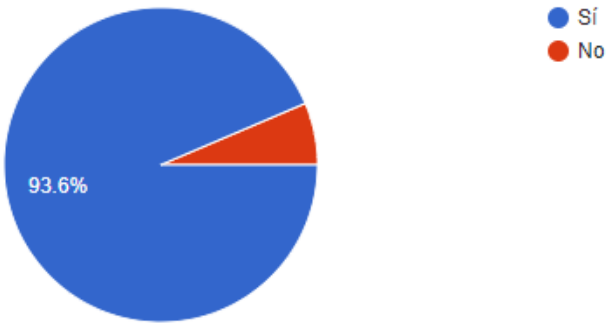


El 94.7% de los encuestados indica que es sencillo o intuitivo usar los sensores de la plataforma, esto indica que tomar como referencia sensores comerciales para la base de su funcionamiento y el circuito de alimentación dieron buenos resultados.

Figura 249

Pregunta 12. ¿Los sensores de la plataforma de monitorización remota tenían un aspecto confiable o de ser de buena calidad?

Alternativas	Respuesta	Porcentaje
Si	88	93.6 %
No	6	6.4 %

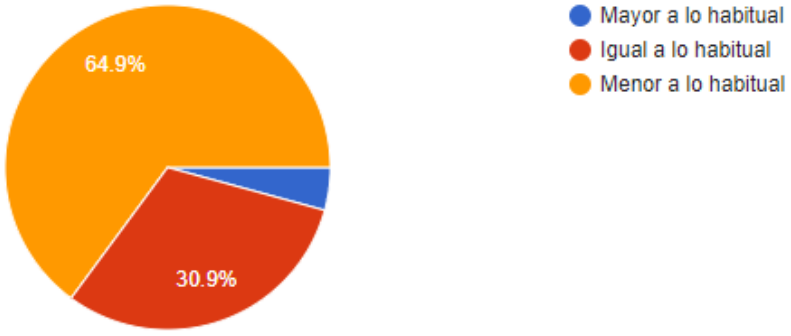


El postprocesado de la impresión 3D dio buenos resultados.

Figura 250

Pregunta 13. El tiempo empleado para tomar sus signos vitales con los sensores de la plataforma resultó ser:

Alternativas	Respuesta	Porcentaje
Mayor a lo habitual	5	5.3 %
Igual a lo habitual	37	39.4 %
Menor a lo habitual	61	64.9 %

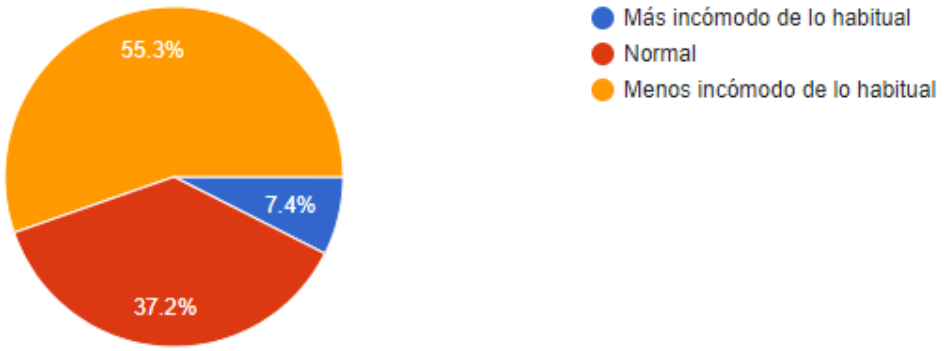


El 64.9% de los pacientes indica que se redujo el tiempo en la toma de signos vitales, esto se debe en gran medida al tensiómetro porque en el centro de salud se acostumbra a usar el método auscultatorio de Korotkoff.

Figura 251

Pregunta 14. El proceso de tomar sus signos vitales con los sensores de la plataforma fue:

Alternativas	Respuesta	Porcentaje
Más incómodo de lo habitual	7	7.4 %
Normal	35	37.2 %
Menos incómodo de lo habitual	52	53.3 %

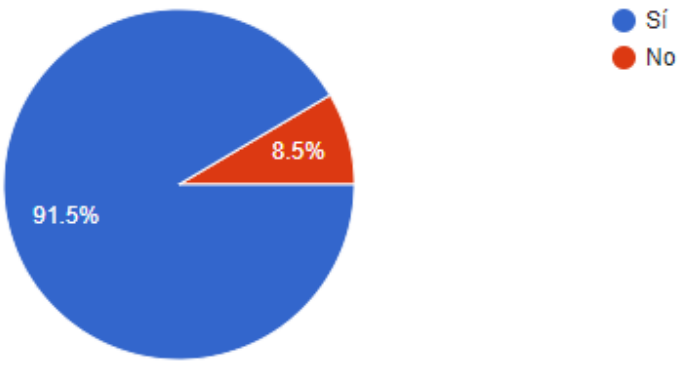


Las pruebas fueron realizadas tanto en el centro de salud como en el hogar de algunos pacientes, el porcentaje de pacientes que sintieron que los sensores eran incómodos es de apenas el 7.4% lo que apunta a un buen diseño, algoritmo de programación y bajo porcentaje de errores de medición o conectividad. Además de que el Chatbot y la página web no se cayeron en ningún momento luego de correr los programas en la máquina virtual.

Figura 252

Pregunta 15. ¿Ha recibido atención médica fuera de la parroquia?

Alternativas	Respuesta	Porcentaje
Si	86	91.5 %
No	8	8.5 %



El 91.5% de la población acude a chequeos médicos fuera de la parroquia, lo más común es que acudan al hospital de Salcedo por medio de una referencia o ingresando por emergencias debido a que el centro de salud solo cuenta con médicos familiares.

Figura 253

Pregunta 16. ¿Basado en su experiencia con el uso de los sensores y sabiendo que se trata de prototipos, cree que el centro de salud debería intentar conseguir más sensores?

Alternativas	Respuesta	Porcentaje
Si	87	92.6 %
No	7	7.4 %

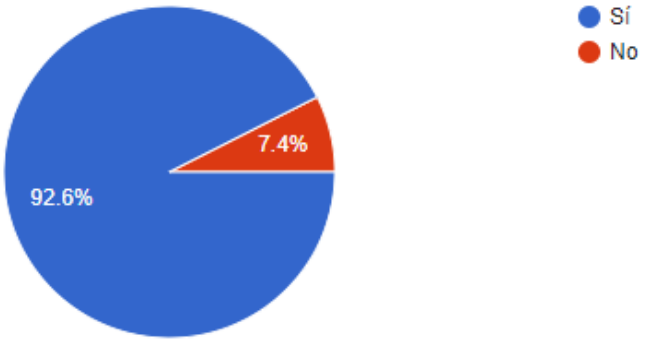
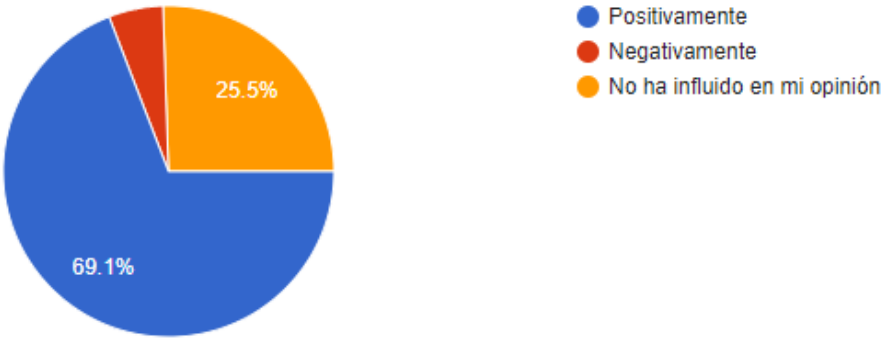


Figura 254

Pregunta 17. ¿Cómo afectó el uso de la plataforma de monitorización remota su opinión sobre la atención que brinda el centro de salud de la parroquia?

Alternativas	Respuesta	Porcentaje
Positivamente	65	5.3 %
Negativamente	24	25.5 %
No ha influido en mi opinión	5	69.1 %



El 69.1% de los encuestados indica que el desarrollo e implementación del prototipo ha influido positivamente en la imagen del centro de salud, no ha influido en la opinión del 25.5% y ha causado una mala impresión en el 5.3% de encuestados.

Con todos los datos de la encuesta se puede concluir que la plataforma web ha causado una buena impresión en la población de la parroquia que acude al centro de salud y sus herramientas responden a las necesidades del sector, además, ha despertado interés en el uso de sistemas de monitorización remota, de computadoras, de celulares y de aplicaciones como apoyo en el cuidado de su salud y el tratamiento de enfermedades crónicas.

Validación de la hipótesis

Al inicio del desarrollo del proyecto se fundamentó la hipótesis de la siguiente forma: ¿El diseño e implementación de un prototipo de monitorización remota basado en el uso de IoMT y Chatbot permitirá optimizar el control de pacientes pertenecientes al grupo de atención prioritaria del Centro de Salud de la parroquia Antonio José Holguín a partir de la recopilación de sus datos objetivos y subjetivos?

Previamente se han validado los dispositivos, el Chatbot y la página web, pero para validar la hipótesis como tal hay que comprobar que el sistema optimiza el control de pacientes pertenecientes al grupo de atención prioritaria, cualquier persona que quiera acceder a los servicios del centro de salud para sí misma o para un familiar debe sacar un turno ese mismo día, los turnos se entregan a las 07:00 am por lo que no es raro ver un grupo de personas haciendo fila en la entrada del centro de salud, a las personas que solicitaban un turno para una toma de signos vitales de rutina de un paciente perteneciente al grupo de atención prioritaria o que tuviera alguna dificultad para trasladarse al centro de salud se les mencionó la posibilidad de una visita médica a sus hogares que generalmente solo se realiza en casos de emergencia con kits de primeros auxilios, en el transcurso de 4 semanas se lograron conseguir 12 muestras, para la validación se trabaja con el tiempo empleado para la toma de signos

vitales en el hogar del paciente con el prototipo y el tiempo que llevaría la toma de signos vitales de no contar con el prototipo.

Figura 255

Entrada centro de salud Antonio José Holguín

La validación se hace por medio de la prueba t de Student para muestras relacionadas unilateral con cola a la derecha, por ende, se plantean la hipótesis nula e hipótesis alternativa.

H_0: El prototipo de monitorización remota no permite optimizar el control de pacientes pertenecientes al grupo de atención prioritaria del Centro de Salud de la parroquia Antonio José Holguín.

H_1: El prototipo de monitorización remota permite optimizar el control de pacientes pertenecientes al grupo de atención prioritaria del Centro de Salud de la parroquia Antonio José Holguín.

Estadístico de prueba: $t = \frac{\bar{d}}{s_d/\sqrt{n}}$

Donde:

$S_d = \sqrt{\frac{(d_i - \bar{d})^2}{n-1}}$

$\bar{d}$: Promedio de las diferencias

S_d: Desviación estándar

Tabla 28

Muestras para validación de hipótesis

Ítem	Tiempo A [min]	Tiempo B [min]	Diferencia
1	45	30	15
2	35	28	7
3	38	25	13
4	45	30	15
5	35	28	7
6	40	35	5
7	35	20	15
8	42	30	12
9	45	27	18
10	40	30	10
11	38	25	13
12	40	24	16
		$\bar{d}$	12,16666667
		S_d	4,086192567

$$n = 12$$

$$t = \frac{\bar{d}}{S_d/\sqrt{n}} = \frac{12{,}16}{4.08/\sqrt{12}}$$

$$t = 10.3144$$

$$gl = (n-1) = 11$$

Para un nivel de confianza del 95%:

$$P(t = 1.7959) = \alpha = 0.05$$

Figura 256

Valor crítico para rechazar la hipótesis alternativa

Grados de libertad	0.25	0.1	0.05	0.025	0.01	0.005
1	1.0000	3.0777	6.3137	12.7062	31.8210	63.6559
2	0.8165	1.8856	2.9200	4.3027	6.9645	9.9250
3	0.7649	1.6377	2.3534	3.1824	4.5407	5.8408
4	0.7407	1.5332	2.1318	2.7765	3.7469	4.6041
5	0.7267	1.4759	2.0150	2.5706	3.3649	4.0321
6	0.7176	1.4398	1.9432	2.4469	3.1427	3.7074
7	0.7111	1.4149	1.8946	2.3646	2.9979	3.4995
8	0.7064	1.3968	1.8595	2.3060	2.8965	3.3554
9	0.7027	1.3830	1.8331	2.2622	2.8214	3.2498
10	0.6998	1.3722	1.8125	2.2281	2.7638	3.1693
11	0.6974	1.3634	1.7959	2.2010	2.7181	3.1058
12	0.6955	1.3562	1.7823	2.1788	2.6810	3.0545
13	0.6938	1.3502	1.7709	2.1604	2.6503	3.0123

$$t_{(1-\alpha),(n-1)} = 1.7959$$

$$P(t = 1.7959) = 0.05$$

$$P(t = 10.3144) = 3{,}18x10^{-8}$$

Escenario:

$$H_0: \mu_d \leq 0$$

$$H_1: \mu_d > 0$$

Se rechaza la hipótesis nula (H_0) si:

$$t > t_{(1-\alpha),(n-1)}$$

Figura 257

Región de aceptación y de rechazo con respecto a H_0

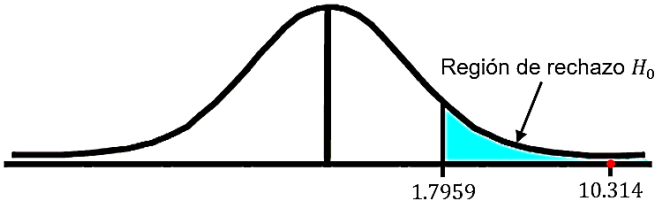


Figura 258

Prueba t con herramienta de análisis de datos de Excel

	Tiempo A [min]	Tiempo B [min]
Media	39,83333333	27,66666667
Varianza	14,6969697	14,78787879
Observaciones	12	12
Coeficiente de correlación de Pearson	0,433712236	
Diferencia hipotética de las medias	0	
Grados de libertad	11	
Estadístico t	10,31438655	
P(T<=t) una cola	2,71071E-07	
Valor crítico de t (una cola)	1,795884819	
P(T<=t) dos colas	5,42142E-07	
Valor crítico de t (dos colas)	2,20098516	

En base a los resultados de la prueba t de Student no se debe aceptar la hipótesis nula con lo que se concluye que $H_1: \mu_d > 0$, la diferencia promedio del tiempo que tarda cada toma de signos vitales es mayor que cero, es decir que se emplea menos tiempo al usar el prototipo.

Análisis de costos

En la Tabla 29 se presenta el desglose de los gastos que se llevaron a cabo para la elaboración del presente proyecto.

Tabla 29

Análisis de costos

Cantidad	Detalle	Valor Unitario USD	Valor Total USD
5	Placa ESP-WROOM-32	17.00	85.00
7	Batería Recargable 18650 8800mah 3.7V	04.50	31.50
5	Conmutador deslizante 2P 24V 0.5A 3 pines	00.35	01.75
5	Jack 5.5x2.1mm Hembra a bornera	00.50	02.50
1	Jack 5.5x2.1mm Macho	00.50	00.50
3	Pantalla LCD 16X2 backlight verde	04.50	13.50
1	Pantalla LCD 20X4 backlight azul	11.00	11.00
4	Módulo I2C para LCD	03.00	12.00
1	Pantalla oled I2C display LCD 128x64 0.96	06.50	06.50
1	Módulo para celda de carga HX711	03.50	03.50
4	Celda de carga 50Kg	06.50	26.00
1	Sensor temperatura infrarrojo MLX90614 DCC	15.00	15.00
2	Buzzer activo	00.50	01.00
3	Diodo led	00.05	00.15
1	Botón pulsador 4 pines 12x12x7.3mm	00.25	00.25
1	Barniz de secado rápido en aerosol	10.00	10.00
1	Sensor de concentración de oxígeno Max30102	12.00	12.00
1	Set de pintura acrílica	09.00	09.00
1	Conversor de nivel lógico de 4 canales	02.50	02.50
1	MCU tensiómetro digital ck-101	17.00	17.00
1	Potenciómetro 10 KΩ	00.10	00.10
2	Amplificador operacional de precisión cuádruple	01.00	02.00
1	Diodo 1N4728A	00.25	00.25
1	Regulador L7805CV	00.50	00.50
1	Arduino NANO	15.00	15.00
1	Cable USB Macho – USB Macho	05.00	05.00
1	Paquete de tiras reactivas	23.00	23.00
1	Maletín de madera	25.00	25.00
67	Impresión 3D	02.00	134.00
1	Página Web	17.27	17.27
10	Resistencias	00.10	01.00
10	Capacitores	00.10	01.00
1	Masilla plástica	06.00	06.00
6	Lijas	00.75	04.50
1	Varios (Cables, estaño, termoflex, , etc.)	25.00	25.00
		Total	520.27

Capítulo V

Conclusiones y recomendaciones

Conclusiones

- Al finalizar el presente trabajo se logró diseñar e implementar un prototipo de monitorización remota de pacientes que permite la recopilación de datos objetivos y subjetivos hacia la nube mediante el uso de Io-MT y Chatbot que optimiza el control de pacientes pertenecientes al grupo de atención prioritaria del centro de salud de la parroquia Antonio José Holguín.
- A través de la investigación previa se garantiza que la plataforma no solo se centra en satisfacer las necesidades del centro de salud, sino que, además, toma en cuenta la definición y características de los sistemas de IoMT y RPM modernos para digitalizar y transformar los procesos del centro de salud.
- Elaborar sensores independientes que se conecten a una red WiFi para enviar los datos a la plataforma es la mejor opción, conectar los sensores por Bluetooth no ofrece ningún beneficio considerable y por otro lado los protocolos diseñados para trabajar en redes de IoT como ZigBee, Zwave o Thread, están pensadas para aplicaciones de domótica o requieren de hardware demasiado costoso para administrar pocos sensores.
- Tal y como se ha comprobado, las especificaciones técnicas de la ESP 32 superan a varias placas de desarrollo populares, además, es compatible con la mayoría de librerías del IDE de Arduino por lo que no se presentan problemas para conectarse a la red WiFi, para ejecutar el protocolo MQTT ni para trabajar con los sensores.
- Los resultados de la encuesta indican que los dispositivos tienen un buen acabado y sus formas no entorpecen la toma de datos, junto con los resultados de los cálculos de error y gráficas de dispersión se corrobora que los dispositivos de la plataforma podrían reemplazar a los dispositivos comerciales del centro de salud de ser necesario.

- Es posible intercambiar información entre los sensores y la base de datos alojada en la nube sin problemas de latencia o errores de comunicación una vez que se encuentren conectados a internet.
- El entrenamiento del Chatbot sobre la plataforma DialogFlow y su posterior integración con WhatsApp se realizó de manera exitosa, los resultados de la encuesta señalan una correcta personalización del agente, la eficiencia de las interacciones y su capacidad de responder a las preguntas más frecuentes de los pacientes del centro de salud que finamente almacena de forma exitosa en la base de datos.
- Al finalizar el desarrollo de la página web se lograron incorporar dos entornos distintos, uno para el personal del centro de salud que facilita tareas administrativas como crear usuarios, cambiar de claves o modificar datos y otro entorno para usuarios que se centra en la visualización de datos en tiempo real mediante gráficas.
- Con la validación de los sensores, el Chatbot, la página web y la hipótesis se concluye que el sistema optimiza el control de pacientes pertenecientes al grupo de atención prioritaria del Centro de Salud de la parroquia Antonio José Holguín.

Recomendaciones

- Reunir una cantidad considerable de signos y síntomas de distintos pacientes para desarrollar algoritmos de IA para detectar patrones que permitan usar la plataforma para diagnosticar enfermedades y no solo para monitorear pacientes que ya han sido diagnosticados.
- Adaptar la plataforma para el control de un mayor número de dispositivos conectados a una misma red de forma simultánea como por ejemplo el mismo centro de salud o un hospital mediante la integración del protocolo Zigbee.
- Todos los dispositivos además de su respectivo sensor integran baterías de litio, para no reducir la vida útil de las baterías se recomienda no dejar que los dispositivos se descarguen por completo y usar el cable de alimentación cuando se requiera un uso prolongado.
- Para disminuir los gastos de impresión 3D es recomendable orientar las piezas de tal forma que se reduzca el tiempo de impresión al mínimo y evitar usar soportes en voladizos de menos de 4 centímetros, para garantizar buenos resultados es necesario realizar test de temperatura y nivelar la cama de la impresora.
- Instalar la extensión Vue.js devtools en el navegador, es un depurador que permite detectar y diagnosticar fallos errores en aplicaciones hechas en Vue.js por medio de su supervisión y edición de datos de componentes en tiempo real, supervisión del rendimiento, estadísticas de tiempo para los métodos del ciclo de vida de los componentes y manejo del enrutamiento.
- Desplegar en Chatbot en múltiples plataformas mediante APIs o mediante alguna de las 14 integraciones single-click pre construidas de DialogFlow como Telegram o Facebook Messenger para que el usuario pueda interactuar desde la plataforma que elija.

- Tomando como base la plataforma presentada en el presente escrito, elaborar un algoritmo que permita personalizar la página web según los gustos o la enfermedad de cada paciente y que se puedan añadir herramientas como alarmas y conversores de unidades.
- Los usuarios deben respetar los protocolos para toma de signos vitales para evitar que se produzcan errores y garantizar que se obtenga la información necesaria y precisa sobre la condición actual de un paciente.

Bibliografía

VS Code. (2021, 3 noviembre). *Code Editing. Redefined*. Visual Studio Code. Recuperado 21 de agosto de 2022, de https://code.visualstudio.com/

ProHealth. (2018, 18 junio*). Difference between Telemedicine and Telehealth. Pro Healthcare Services*. Recuperado 3 de febrero de 2022, de https://prohealthware.com/es/difference-between-telemedicine-and-telehealth/

Healthit. (2019, 17 octubre). *What is telehealth? How is telehealth different from telemedicine?* | *HealthIT.gov*. Recuperado 29 de diciembre de 2021, de https://www.healthit.gov/faq/what-telehealth-how-telehealth-different-telemedicine

Romero, C. (2021, 15 febrero). *Diferencia entre Telemedicina y Telesalud*. Martin&Romero. Recuperado 3 de febrero de 2022, de https://www.myrgroup.pe/blog/cual-es-la-diferencia-entre-telemedicina-teleasistencia-y-telesalud-6#:%7E:text=El%20t%C3%A9rmino%20telesalud%20lo%20abarca,m%C3%A1s%20amplio%20del%20t%C3%A9rmino%20telesalud

Gillis, A. S. (2022b, marzo 4). *What is the internet of things (IoT)?* IoT Agenda. Recuperado 21 de agosto de 2022, de https://www.techtarget.com/iotagenda/definition/Internet-of-Things-IoT

Singh, J., Rice, A., Ron, E., & Harris, C. (2018, 8 octubre). *A High-Growth and Cost-Saving Industry Shaping the Future of Healthcare Delivery*. CreditSuisse. https://research-doc.credit-suisse.com/docView?language=ENG&format=PDF&sourceid=em&document_id=1080795741&serialid=1M5MIeePDC5H3SxI1ny8An6ynJvupL%2ftN3dnXA7dfNI%3d

Jiménez, D. (2018). *Sistema de ubicación y monitoreo de señales vitales en adultos mayores y personas con Alzheimer* [Tesis de pregrado, Universidad Técnica de Ambato].

Anchaluisa, D. (2018). *Sistema ambulatorio para el control de signos vitales y prevención de la diabetes Mellitus* [Tesis de pregrado, Universidad Técnica de Ambato].

Haro, E. (2020). *Prototipo e-Health basado en sistemas empotrados de bajo costo para monitoreo de signos vitales a través de internet* [Tesis de maestría, Universidad de las Fuerzas Armadas ESPE].

Castelo, J. (2021). *Diseño y aplicación de un sistema de control y manejo telemático para personas con la enfermedad de diabetes* [Tesis de pregrado, Universidad Politécnica Salesiana Sede Guayaquil].

Espinosa, J. (2021*). Sistema de monitoreo cardíaco portátil de una terminación para atención primaria o control médico remoto de pacientes con anomalías cardiacas* [Tesis de pregrado, Universidad de Cuenca].

Garcés, R. & Villafuerte V. (2021*). Sistema de telemedicina con monitoreo de signos vitales basado en IOT en un ambiente Smart TV* [Tesis de pregrado, Universidad Técnica de Ambato].

Wikimedia. (2022, 4 julio). *Telemedicine. Wikiversity*. Recuperado 21 de agosto de 2022, de https://en.wikiversity.org/wiki/Telemedicine

Olivares, D. (2021, 31 mayo). *Curiosidades sobre el origen y la evolución histórica de la telemedicina*. MuyPymes. Recuperado 3 de febrero de 2022, de https://www.muypymes.com/2020/10/12/curiosidades-origen-evolucion-historica-telemedicina

Pardell, X. (2021, 9 diciembre). *TICs, Telemedicina y eSalud - Apuntes de Electromedicina.* Pardell. Recuperado 3 de febrero de 2022, de https://www.pardell.es/tics,-telemedicina-y-esalud.html

Franceschini, C. (2020, 11 junio). *Pandemia covid-19 ¿la telemedicina llegó para quedarse?* AAMR. Recuperado 3 de febrero de 2022, de http://www.aamr.org.ar/lagaceta/andemia-covid-19-la-telemedicina-llego-para-quedarse/#:%7E:text=Willem%20Einthoven%2C%20que%20fue%20premio,realiz%C3%B3%20por%20electrodos%20de%20inmersi%C3%B3n

Hedges, K. (2021, 28 julio). *What Is Telehealth? What Is Remote Patient Monitoring? How Are They Different?* Care Innovations. Recuperado 30 de diciembre de 2021, de https://news.careinnovations.com/blog/what-is-telehealth-what-is-remote-patient-monitoring-how-are-they-different

Gates, L. (2021, 30 abril). *Future of Healthcare: Internet of Medical Things.* Insight. Recuperado 5 de enero de 2022, de https://www.insight.com/en_US/content-and-resources/2021/future-of-healthcare--internet-of-medical-things.html

Steger, A. (2020, 6 mayo). *How the Internet of Medical Things Is Impacting Healthcare.* Technology Solutions That Drive Healthcare. Recuperado 6 de enero de 2022, de https://healthtechmagazine.net/article/2020/01/how-internet-medical-things-impacting-healthcare-perfcon

Ratiopharm. (2020, 5 noviembre). *Abordaje de la neumonía silenciosa en la farmacia*. Recuperado 7 de enero de 2022, de https://ratiopharm.es/en-la-botica/actualidad-farmaceutica/abordaje-de-la-neumonia-silenciosa-en-la-farmacia

Ordorica, I. (2020, 20 agosto). *Qué es y para qué sirve Google Cloud Platform*. Incentro. Recuperado 22 de agosto de 2022, de https://www.incentro.com/es-ES/blog/que-es-google-cloud-platform

MQTT. (2022, 5 agosto). *MQTT - The Standard for IoT Messaging*. Recuperado 29 de agosto de 2022, de https://mqtt.org/

Higuerey, E. (2020, 17 febrero). *NGINX: entiende cómo funciona este servidor web y cómo se diferencia a Apache*. Rockcontent. Recuperado 30 de agosto de 2022, de https://rockcontent.com/es/blog/nginx/

Grabowski, M. & Netguru. (2018, 15 octubre). *What Is DialogFlow and Why Do You Need It?* Netguru. Recuperado 31 de agosto de 2022, de https://www.netguru.com/blog/what-is-dialogflow

López, P. (2022, 14 agosto). *Getting Started | whatsapp-web.js*. Whatsapp-Web.Js. Recuperado 31 de agosto de 2022, de https://wwebjs.dev/guide/#installation

Cornieles, P. (2019, 1 abril). *El uso de chatbots en Latinoamérica viene creciendo a pasos agigantados*. IA Latam. Recuperado 31 de agosto de 2022, de https://ia-latam.com/2019/04/01/el-uso-de-chatbots-en-latinoamerica-viene-creciendo-a-pasos-agigantados/

Sequelize.org. (2022, 23 agosto). *Getting Started | Sequelize*. Sequelize. Recuperado 1 de septiembre de 2022, de https://sequelize.org/docs/v6/getting-started/

Asanza, V. (2021, 6 julio). *Especificaciones del módulo ESP32*. Aiware. Recuperado 7 de septiembre de 2022, de https://vasanza.blogspot.com/2021/07/especificaciones-del-modulo-esp32.html

Darnell, T. (2019, 9 abril). *Adding a replicated MySQL database instance using a Group Replication server as the source.* Scripting MySQL. Recuperado 7 de septiembre de 2022, de https://scriptingmysql.wordpress.com/category/mysql-replication/

Mosquitto. (2022, 6 julio). *mosquitto*. test mosquitto. Recuperado 15 de septiembre de 2022, de https://test.mosquitto.org/

amCharts. (2022, 16 septiembre). *amCharts 5 Demos*. Recuperado 16 de septiembre de 2022, de https://www.amcharts.com/demos/

amCharts. (2022, 10 marzo). Line Graph. Recuperado 16 de septiembre de 2022, de https://www.amcharts.com/demos/line-graph/

EMQTech. (2022, 6 octubre). EMQX The Most Scalable MQTT Broker for IoT. EMQX. Recuperado 15 de octubre de 2022, de https://www.emqx.io/downloads?os=Ubuntu

EMQX. (2020, 7 febrero). *EMQX Authentication JWT*. EMQX 4.3 Documentation. Recuperado 20 de octubre de 2022, de https://www.emqx.io/docs/en/v4.3/advanced/auth-jwt.html

VanHome, M. (2021, 17 febrero). *How to design parts for FDM 3D printing*? Hubs. https://www.hubs.com/knowledge-base/how-design-parts-fdm-3d-printing/

Aladuino. (2021, 11 noviembre). *Generador de caracteres especiales para LCD alfanuméricas*. Electrónica Especializada. https://www.aladuino.com.mx/blog/generador-de-caracteres-especiales-para-lcd-alfanumericas/

Melexis. (2019). *MLX90614 family: Single and Dual Zone*. https://html.alldatasheet.com/html-pdf/885637/MELEXIS/MLX90614ESF-DCC-000-TU/309/1/MLX90614ESF-DCC-000-TU.html

Figueroa, A., Rojas, S., Sánchez, E., Ramírez, M. & Cabrera, A. (2016). *Oxímetro de pulso no invasivo aplicado en el monitoreo atlético*. https://www-optica.inaoep.mx/~tecnologia_salud/2016/documentos/memorias/MyT2016_024_E.pdf

Freemeteo. (2022, 14 noviembre). *Tiempo Antonio José Holguín*. https://freemeteo.ec/mobile/eltiempo/antonio-jose-holguin/historia/historial-mensual/?gid=3660591&station=23040&month=11&year=2022&language=spanishar&country=ecuador

Lewis, P. (2021, 12 febrero). *GMISDN Blood pressure measurement webinar* [Vídeo]. YouTube. Recuperado 8 de diciembre de 2022, de https://www.youtube.com/watch?v=m6N0FTciKrk

Gonzales, V., Ahmed, M. & Abbosh, A. (2019, 15 febrero). *The Progress of Glucose Monitoring-A Review of Invasive to Minimally and Non-Invasive Techniques, Devices and Sensors.* Abstract - Europe PMC. Recuperado 1 de enero de 2023, de https://europepmc.org/article/PMC/6412701

Fitzpatrick, D. (2014). Implantable Electronic Medical Devices. Academic Press.

Long, J. (2018, 20 agosto). *Chronoamperometry with glucose test strips*. IO Rodeo Resources. Recuperado 10 de noviembre de 2022, de https://blog.iorodeo.com/chronoamperometry-with-glucose-test-strips/

Anexos