



**UNIVERSIDAD TECNOLÓGICA CENTROAMERICANA FACULTAD DE
INGENIERÍA
PROYECTO DE INVESTIGACIÓN**

**IMPLEMENTACIÓN DEL IOT PARA PROMOVER EL
DISTANCIAMIENTO SOCIAL DURANTE LA RECOLECCIÓN
DE CAFÉ**

**PREVIO A LA OBTENCIÓN DEL TÍTULO:
INGENIERO EN MECATRÓNICA**

PRESENTADO POR:

21611185

JOSÉ DARÍO ENAMORADO LEIVA

**ASESOR: JOSÉ LUIS ORDOÑEZ ÁVILA
CAMPUS: SAN PEDRO SULA; JUNIO, 2020**

Dedicatoria

A mis padres y hermanos los cuales siempre estuvieron apoyándome durante todo el trayecto académico y dándome consejo para poder dar mi mayor esfuerzo en todo lo que haga.

Agradecimiento

Le agradezco en primer lugar a Dios quien me ha dado fuerzas y me ha ayudado emocionalmente durante el largo recorrido tomado en busca de mi título de pregrado.

A mis amigos, Margie Madrid, Wolfgang Hanz, Adrián mora, David Duarte, Erick Rivera, Andrea Rojas, Leo Enamorado y mis compañeros de clase, con quienes se transitó este camino compartiendo tanto alegrías como tristezas.

Epígrafe

“Parate derecho con los hombros hacia atrás”

-Jordan B. Peterson

Resumen Ejecutivo

El Internet de las cosas (IoT) en combinación con una aplicación de página única (SPA) es un nuevo tipo de solución para el sistema de recolección de café utilizado en el Beneficio Rio Frio y que da paso a crear nuevas maneras de trabajos acordes a las medidas de distanciamiento tomadas por causa del COVID-19. Originalmente, el proceso de recolección se ha realizado a mano durante más de 100 años, por lo tanto este proyecto se enfoca en dar una nueva manera de abordar este trabajo. Este documento discute el uso IoT para enviar datos a un SPA. El dispositivo IoT de este proyecto es una báscula digital que ha sido modificada con un microcontrolador capaz de enviar datos de manera inalámbrica en tiempo real utilizando un protocolo IoT, la SPA está destinada a ser utilizada por cualquier tipo de dispositivo IoT y personas con bajo conocimiento técnico. Para medir las capacidades del protocolo de comunicación en combinación con la SPA, se ha creado un entorno para pruebas construido dentro del sistema central; Al usar este entorno, se muestra el tiempo entre las comunicaciones y podemos ver qué efectos tiene el tamaño del mensaje y la cantidad de clientes en el contexto de una aplicación de la vida real. Disponer de un sistema como este permite crear ambientes de trabajos en los que se reduzcan las aglomeraciones de personas que se generan durante la temporada de cosecha, reduciendo las probabilidades de propagación del virus en una región con escasos recursos médicos.

Palabras clave: MQTT, SPA; IoT; Café; Vue.js, COVID-19

Abstract

The Internet of Things (IoT) in combination with a single page application (SPA) is a new type of solution for the coffee recollection system used in the Beneficio Rio Frio thus creating new ways of working according to the distancing measures taken because of COVID-19. Originally, the recollection process has been carried out by hand for more than 100 years, therefore this project focuses on giving a new way of approaching this work. This document discusses the use of IoT to send data to a SPA. The IoT device of this project is a digital scale that has been modified with a microcontroller capable of sending data wirelessly in real time using an IoT protocol, the SPA is intended to be used by any type of IoT device and people with low knowledge technical. To measure the capabilities of the communication protocol in combination with the SPA, a test environment built within the central system has been created; Using this environment, the time between communications is displayed and we can see what effects message size and number of clients have in the context of a real life application. Having a system like this allows to create work environments in which the crowds of people that are generated during the harvest season are reduced, reducing the probability of spreading the virus in a region with scarce medical resources.

Key words: MQTT, SPA; IoT; Coffee; Vue.js, COVID-19

ÍNDICE DE CONTENIDO

CAPÍTULO I: INTRODUCCIÓN	15
CAPÍTULO II. PLANTEAMIENTO DEL PROBLEMA	16
2.1 PRECEDENTES DEL PROBLEMA	16
2.2 DEFINICIÓN DEL PROBLEMA	16
2.3 JUSTIFICACIÓN	17
2.4 PREGUNTAS DE INVESTIGACIÓN.....	17
2.5 OBJETIVOS	18
2.5.1 <i>Objetivo general</i>	18
2.5.2 <i>Objetivos específicos</i>	18
CAPÍTULO III. MARCO TEÓRICO.....	19
3.1 EL CAFÉ	19
3.1 <i>Historia del café en Honduras</i>	19
3.1.1 <i>Método de recolección del café hondureño en la actualidad</i>	20
3.1.2 <i>Métodos de recolección del café en otros países</i>	21
3.1.3 <i>Importancia del café en Honduras</i>	22
3.1.4 <i>El panorama del café en honduras respecto a la pandemia de Covid-19</i>	24
3.2 COVID-19.....	24
3.2.1 <i>Los Coronavirus</i>	25
3.2.2 <i>Síntomas</i>	25
3.2.3 <i>Propagación</i>	25
3.2.4 <i>Medidas de prevención</i>	26
3.2.5 <i>Tratamiento</i>	26
3.3 EL INTERNET DE LAS COSAS (IoT).....	26
3.3.1 <i>Protocolos del Internet de las cosas</i>	28
3.3.2 <i>Dispositivos en la red IoT</i>	32
3.4 INTERNET	34
3.4.1 <i>Hypertext Transfer Protocol (HTTP)</i>	34
3.5 PÁGINAS WEB.....	36
3.5.1 <i>Hyper Text Markup Language (HTML)</i>	36
3.5.2 <i>Cascading Style Sheets (CSS)</i>	37
3.5.3 <i>JavaScript</i>	39
3.5.4 <i>Hypertext Preprocessor (PHP)</i>	40
3.5.5 <i>Single-page application (SPA)</i>	40
3.6 BASES DE DATOS.....	41
3.6.1 <i>bases de datos digitales</i>	41
3.6.2 <i>Bases de datos relacionales</i>	42
3.6.3 <i>Structured Query Language (SQL)</i>	44
3.7 FRAMEWORKS	44
3.7.1 <i>Laravel</i>	45
3.7.2 <i>Node.js</i>	45
3.7.3 <i>Vue.js</i>	46

3.7.4	React.js.....	46
3.7.5	Angular.....	46
CAPÍTULO IV	METODOLOGÍA.....	48
4.1	ENFOQUE Y MÉTODO.....	48
4.2	VARIABLES DE INVESTIGACIÓN.....	48
4.3	TECNOLOGÍAS APLICADAS.....	49
4.4	MATERIALES.....	49
4.5	METODOLOGÍA DE ESTUDIO.....	50
4.5.1	<i>Elaboración de la tabla usuarios juntos a los formularios de registro.....</i>	<i>50</i>
4.5.2	<i>Implementación del módulo de autenticación, permisos y la creación del formulario de login y permisos.....</i>	<i>51</i>
4.5.3	<i>Creación la tabla ingresoFacturas y su formulario facturación de ingreso.....</i>	<i>52</i>
4.5.4	<i>Implementación del protocolo seleccionado para la comunicación de los dispositivos IoT.....</i>	<i>54</i>
4.5.5	<i>desarrollo de la báscula digital IoT.....</i>	<i>55</i>
4.6	METODOLOGÍA DE VALIDACIÓN.....	56
4.7	CRONOGRAMA DE ACTIVIDADES.....	56
CAPÍTULO V.	RESULTADOS Y ANÁLISIS.....	58
5.1	ANÁLISIS DE SUSTENTO DEL MARCO TEÓRICO.....	58
5.2	CONSTRUCCIÓN DE LA SPA.....	59
5.3	DESARROLLO DE LA ETAPA 1: ELABORACIÓN DE LA TABLA USUARIOS JUNTOS A LOS FORMULARIOS DE REGISTRO.....	63
5.3.1	<i>Creación de las migraciones.....</i>	<i>63</i>
5.3.2	<i>Creación del formulario de registro.....</i>	<i>63</i>
5.3.3	<i>Prueba de funcionamiento.....</i>	<i>64</i>
5.4	DESARROLLO DE LA ETAPA 2: IMPLEMENTACIÓN DEL MÓDULO DE AUTENTIFICACIÓN, PERMISOS Y LA CREACIÓN DEL FORMULARIO DE LOGIN Y PERMISOS.....	65
5.4.1	<i>Creación de las migraciones.....</i>	<i>65</i>
5.4.2	<i>Creación del formulario de permisos y login.....</i>	<i>66</i>
5.4.3	<i>Prueba de funcionamiento.....</i>	<i>67</i>
5.5	DESARROLLO DE LA ETAPA 3: CREACIÓN LA TABLA INGRESOFACTURAS Y SU FORMULARIO FACTURACIÓN DE INGRESO.....	67
5.5.1	<i>Creación de las migraciones.....</i>	<i>68</i>
5.5.2	<i>Creación del formulario de facturación.....</i>	<i>69</i>
5.5.3	<i>Prueba de funcionamiento.....</i>	<i>73</i>
5.6	DESARROLLO DE LA ETAPA 4: IMPLEMENTACIÓN DEL PROTOCOLO SELECCIONADO PARA LA COMUNICACIÓN DE LOS DISPOSITIVOS IOT.....	73
5.6.1	<i>Diseño de la ventana emergente.....</i>	<i>73</i>
5.6.3	<i>Pruebas.....</i>	<i>74</i>
5.7	DESARROLLO DE LA ETAPA 5: DESARROLLO DE LA BÁSCULA DIGITAL IOT.....	76
5.7.1	<i>Pruebas dentro de la SPA.....</i>	<i>77</i>
5.8	UBICACIÓN DE LOS PUNTOS DE RECEPCIÓN.....	77
CAPÍTULO VI:	CONCLUSIONES.....	81

CAPÍTULO VII: RECOMENDACIONES	82
CAPÍTULO VIII: ANEXOS.....	83
8.1 RESULTADO DE LAS PRUEBAS.....	83
8.1.1 tamaño del mensaje: 5bytes, usuarios :1.....	83
8.1.2 tamaño del mensaje: 5bytes, usuarios: 2.....	84
8.1.3 tamaño del mensaje: 5bytes, usuarios: 3.....	85
8.1.4 tamaño del mensaje: 5bytes, usuarios: 5.....	86
8.1.5 tamaño del mensaje: 50bytes, usuarios: 1.....	88
8.1.6 tamaño del mensaje: 100bytes, usuarios: 1.....	89
8.2 MANUAL DE USO.....	89
8.2.1 formulario de Login.....	89
8.2.2 Sección de noticias y mensajes.....	90
8.2.3 Menú desplegable.....	91
8.2.4 Indexador de formularios.....	92
8.2.5 Indexador de usuarios.....	93
8.2.6 Indexador de ingresoFacturas.....	94
8.2.7 formulario de ingresoFacturas.....	96
8.2.8 formulario de permisos.....	99
8.2.9 formulario de registro.....	100
8.2.10 ventana emergente de conexión MQTT.....	101
8.2.11 formulario de confirmación.....	101
CAPÍTULO IX: REFERENCIAS	103

ÍNDICE DE ILUSTRACIONES

FIG. 1 COMPARATIVO PRODUCCIÓN POR DEPARTAMENTO 2014-2015/2015-2016 (IHCAFE.S. F).....	20
FIG. 2 DESGLOSE DE LOS COSTOS DE PRODUCCIÓN DE UN QUINTAL ORO (JUAREZ,2018).....	22
FIG. 3 RESULTADO TOTAL DE LA ESCUELA DE CATADORES HASTA LA FECHA (IHCAFE).....	23
FIG. 4 COMPARACIÓN DE CASOS A LO LARGO DEL TIEMPO ENTRE DISTINTAS REGIONES. (OMS).....	24
FIG. 5 VISIÓN DEL INTERNET DE LAS COSAS(IOT)... (CAMA, 2012).....	27
FIG. 6 REPRESENTACIÓN DEL ESQUEMA PUB/SUB (AUTORÍA PROPIA).....	29
FIG. 7 RITMO DE CRECIMIENTO DE LOS DISPOSITIVOS IOT (GAOYANG SHAN ET AL., 2016).....	31
FIG. 8 ESQUEMA DE UNA RED ZIGBEE.....	32
FIG. 9 REPRESENTACIÓN DEL FUNCIONAMIENTO DEL PROTOCOLO HTML (AUTORÍA PROPIA).....	35
FIG. 10 EJEMPLO DEL LENGUAJE DE ETIQUETAS HTML (EGUILUZ,2008).....	37
FIG. 11 EFECTO DE DISTINTOS ATRIBUTOS SOBRE UN MISMO ELEMENTO (AUTORÍA PROPIA).....	37
FIG. 12 ESQUEMA DE LA SEPARACIÓN DE LOS CONTENIDOS Y SU PRESENTACIÓN (EGUÍLUZ, 2008).....	38
FIG. 13 CONOCIMIENTO, INTERÉS Y SATISFACCIÓN DE LOS USUARIOS CON LOS DISTINTOS PREPROCESADORES CSS (GREIF & BENITTE, 2019).....	39
FIG. 14 RANKIN DE LOS LENGUAJES DE PROGRAMACIÓN MÁS UTILIZADOS. (STACK OVERFLOW, 2019).....	39
FIG. 15 CONOCIMIENTO, INTERÉS Y SATISFACCIÓN DE LOS USUARIOS A LO LARGO DE LOS AÑOS. (GREIF & BENITTE, S. F.).....	47
FIG. 16 VARIABLES DEPENDIENTES E INDEPENDIENTES DE LA INVESTIGACIÓN (FUENTE PROPIA).....	49
FIG. 17 PRIMER INCREMENTO (FUENTE PROPIA).....	51
FIG. 18 SEGUNDO INCREMENTO (FUENTE PROPIA).....	52
FIG. 19 TERCER INCREMENTO (FUENTE PROPIA).....	54
FIG. 20 CUARTO INCREMENTO (FUENTE PROPIA).....	55
FIG. 21 QUINTO INCREMENTO (FUENTE PROPIA).....	56
FIG. 22 CRONOGRAMA DE ACTIVIDADES (FUENTE PROPIA).....	57
FIG. 23 TABLA COMPARATIVA DE PROTOCOLOS IOT (FUENTE PROPIA).....	58
FIG. 24 TABLA COMPARATIVA DE FRAMEWORKS FRONT-END (FUENTE PROPIA).....	59
FIG. 25 REPRESENTACIÓN VISUAL DEL ESQUEMA MVC (FUENTE PROPIA).....	60
FIG. 26 PROCEDIMIENTO PARA AGREGAR RUTAS (FUENTE PROPIA).....	61
FIG. 27 NAVEGACIÓN POR LA SPA (FUENTE PROPIA).....	62
FIG. 28 TABLA USUARIOS (FUENTE PROPIA).....	63
FIG. 29 FORMULARIO DE REGISTRO (FUENTE PROPIA).....	63
FIG. 30 FUNCIONAMIENTO DEL FORMULARIO DE REGISTRO (FUENTE PROPIA).....	64
FIG. 31 SEEDER DE LA TABLA USUARIOS (FUENTE PROPIA).....	65
FIG. 32 RELACIÓN VARIOS A VARIOS ENTRE LA TABLA PERMISSIONS Y LA TABLA USUARIOS.....	66
FIG. 33 FORMULARIO DE PERMISOS (FUENTE PROPIA).....	66
FIG. 34 FORMULARIO DE LOGIN (FUENTE PROPIA).....	67
FIG. 35 SEEDER DE LA TABLA PERMISSIONS (FUENTE PROPIA).....	67
FIG. 36 TABLA INGRESOFACTURAS (FUENTE PROPIA).....	68
FIG. 37 TRES RELACIONES UNO A VARIO ENTRE LA TABLA INGRESOFACTURAS Y LA TABLA USUARIOS (FUENTE PROPIA).....	69
FIG. 38 RESUMEN DEL FORMULARIO INGRESOFACTURAS (FUENTE PROPIA).....	69
FIG. 39 FUNCIONAMIENTO DEL FORMULARIO INGRESOFACTURAS CON LA INTENCIÓN "CREAR" (FUENTE PROPIA).....	71

FIG. 40 FUNCIONAMIENTO DEL FORMULARIO INGRESOFACTURAS CON LA INTENCIÓN "VER" (FUENTE PROPIA)	72
FIG. 41 FUNCIONAMIENTO DEL FORMULARIO INGRESOFACTURAS CON LA INTENCIÓN "ANULAR" (FUENTE PROPIA)	72
FIG. 42 VENTANA EMERGENTE PARA LA CONEXIÓN MQTT (FUENTE PROPIA)	74
FIG. 43 BÁSCULA DIGITAL A MODIFICAR (FUENTE PROPIA)	76
FIG. 44 UBICACIÓN DE LOS PUNTOS DE RECEPCIÓN (FUENTE PROPIA)	78
FIG. 45 RESULTADO FINAL	79
FIG. 46 PRUEBA 1 EN USUARIO 1	83
FIG. 47 PRUEBA 2 EN USUARIO 1	83
FIG. 48 PRUEBA 3 EN USUARIO 1	83
FIG. 49 PRUEBA 4 EN USUARIO 1	84
FIG. 50 PRUEBA 1 EN USUARIO 1	84
FIG. 51 PRUEBA 1 EN DISPOSITIVO 2	84
FIG. 52 PRUEBA 2 EN DISPOSITIVO 1	85
FIG. 53 PRUEBA 2 EN DISPOSITIVO 2	85
FIG. 54 PRUEBA 1 EN DISPOSITIVO 1	85
FIG. 55 PRUEBA 1 EN DISPOSITIVO 2	86
FIG. 56 PRUEBA 1 EN DISPOSITIVO 3	86
FIG. 57 PRUEBA 1 EN DISPOSITIVO 1	86
FIG. 58 PRUEBA 1 EN DISPOSITIVO 2	87
FIG. 59 PRUEBA 1 EN DISPOSITIVO 3	87
FIG. 60 PRUEBA 1 EN DISPOSITIVO 4	87
FIG. 61 PRUEBA 1 EN DISPOSITIVO 5	88
FIG. 62 PRUEBA 1 ENVÍO DE MENSAJE DE 50BYTES	88
FIG. 63 PRUEBA 2 ENVÍO DE MENSAJE DE 50BYTES	88
FIG. 64 PRUEBA 1 ENVÍO DE MENSAJE DE 100BYTES	89
FIG. 65 PRUEBA 2 ENVÍO DE MENSAJE 100BYTES	89
FIG. 66 PÁGINA INICIAL	89
FIG. 67 SECCIÓN DE NOTICIAS Y MENSAJES	90
FIG. 68 MENÚ DESPLEGABLE	91
FIG. 69 INDEXADOR DE FORMULARIOS	92
FIG. 70 INDEXADOR DE REGISTROS	94
FIG. 71 MENÚ DESPLEGABLE DE CAMPOS DE FILTRADO	95
FIG. 72 PASO 3 DEL FORMULARIO DE INGRESOFACTURAS CON INTENCIÓN CREAR	96
FIG. 73 FORMULARIO INGRESOFACTURAS CON INTENCIÓN CREAR	96
FIG. 74 PASO 4 DEL FORMULARIO DE INGRESOFACTURAS CON INTENCIÓN CREAR	97
FIG. 75 PASO 5 DEL FORMULARIO DE INGRESOFACTURAS CON INTENCIÓN CREAR	97
FIG. 76 FORMULARIO DE INGRESOFACTURAS CON INTENCIÓN VER	98
FIG. 77 FORMULARIO INGRESOFACTURAS CON INTENCIÓN ANULAR	99
FIG. 78 VENTANA EMERGENTE DE CONEXIÓN MQTT (FUENTE PROPIA)	101
FIG. 79 FORMULARIO DE CONFIRMACIÓN	101

CAPÍTULO I: INTRODUCCIÓN

En este capítulo se resume el contenido de todos los capítulos encontrados en este documento.

Capítulo II Planteamiento del problema: Se describe la importancia de este proyecto, la necesidad de digitalizar la información de beneficio Rio Frio y de tomar medidas para promover el distanciamiento social. Este capítulo termina explicando la manera en que estos problemas serán solucionados.

Capítulo III Marco teórico: En este capítulo se recopila una amplia gama de información que otorga contexto al proyecto por medio de la exploración de la teoría referente al café, el COVID-19, la creación de una SPA y la comunicación IoT.

Capítulo IV Metodología: Define la manera en la que este proyecto fue llevado a cabo, explicando el cronograma de trabajo, las variables a investigar y el enfoque con la que estas se analizan.

Capítulo V Resultados y análisis: Contiene la descripción detallada de la realización del proyecto en cada una de sus etapas. En las etapas finales se analizan las variables de investigación por medio del mismo sistema creado.

Capítulo VI Conclusiones: Se enfoca en dar respuesta a las preguntas planteadas en el capítulo II utilizando la información encontrada tanto en el capítulo III como en el capítulo V.

Capítulo VII Recomendaciones: Incluye sugerencias para mejorar la calidad de los resultados y para continuar con el desarrollo del proyecto en el futuro.

Capítulo VIII Anexos: Aquí se pueden encontrar las imágenes de las pruebas realizadas, y el manual de uso del sistema.

CAPÍTULO II. PLANTEAMIENTO DEL PROBLEMA

En este capítulo se detallará el problema de la investigación a profundidad, así como las razones que originaron este problema y la manera en la que se le dará una solución a este

2.1 Precedentes del problema

Con la pandemia de COVID-19 se ha visto una urgente necesidad de aumentar la digitalización de muchas de las actividades que tradicionalmente se han hecho con fuerza laboral humana. Como lo indica (Agudelo, 2020) tomando en cuenta la situación de desarrollo e la que se encuentra américa latina en términos de digitalización, se nos presenta la interrogante de si esta región dispone de los medios digitales necesarios para hacer frente a la pandemia del COVID-19. Es por esta razón que es importante reforzar el nivel de digitalización a lo largo de américa latina, incluyendo a Honduras. En honduras gran parte de su PIB agrícola proviene de la producción y exportación del café, y según (Bunn et al., 2018) alrededor de una 110,000 mil familias obtienen la mayoría de sus ingresos del rubro del café. Entre esta gran cantidad de familias, muchas de ellas son productoras o recolectores de café y es muy común en américa latina la migración de personas en época de cosecha de café, migración que puede acarrear junto a ella un contagio de COVID-19, es por esta razón que es imperativo el diseño de nuevas tecnologías que eviten lo más que se pueda las aglomeraciones de personas. La implementación del internet de la cosas según (Felipe et al., 2018) nos permite la obtención de información de nuevas formas y brinda muchas nuevas posibilidades a lo que puede ser creado en todos los sectores de la humanidad, con esto en mente es lógico pensar que el sector café se puede beneficiar mucho de esta tecnología también. Implementando dispositivos del internet de las cosas podemos reducir el contacto humano necesario para realizar muchas de las actividades que se llevan a cabo durante la recolección del café, en especial en esos momentos en los que muchos recolectores se reúnen para medir la cantidad de café recolectado en el día.

2.2 Definición del problema

El problema principal a tratar es el de buscar una manera de dar un trato eficiente a los clientes sin necesidad de que estos tengan que romper las medidas de distanciamiento social obligatorias para el control de la pandemia, la urgente necesidad de digitalizar y automatizar las tareas más comunes en el área de atención al cliente como el control de la cantidad de café entregado por cada cliente en específico y atender las necesidades de Beneficio Rio Frio de tener una base de

datos que contenga toda esta información disponible.

2.3 Justificación

Una Aplicación de página única (SPA) junto al Internet de las cosas (IoT), son tecnologías modernas que combinadas pueden resolver los problemas que Beneficio Rio Frio afronta en estos momentos, ya que el proceso de mejora de las redes de telecomunicación está puesto en marcha a su vez la popularidad de los smartphones nos da una base sólida sobre la cual trabajar.

2.4 Preguntas de investigación

- 1) ¿Cuánto tendría que extenderse la red para atacar el problema del distanciamiento social (m)
- 2) ¿Cuántos puntos de recepción se tendrán que colocar?
- 3) ¿Cómo cambia la latencia a medida que se aumenta la cantidad de dispositivos(ms)?
- 4) ¿Cómo cambia la latencia a medida que se aumenta el tamaño del mensaje IoT (ms)?

2.5 Objetivos

2.5.1 Objetivo general

Diseñar una SPA que pueda ser accesible por cualquier dispositivo conectado a la intranet, ya sean dispositivos operados por humanos como lo son los smartphones, tablets y computadoras; como dispositivos de adquisición de datos, por medio de básculas digitales y que al mismo tiempo la información generada por la SPA sea guardada en una base de datos

2.5.2 Objetivos específicos

- 1) Diseñar un SPA pensada tanto para dispositivos móviles como para computadoras tradicionales a la cual se pueda acceder en cualquier punto que se extienda la red local.
- 2) Proponer un mapa en el que se detalle la ubicación geográfica de los puntos de recepción
- 3) Crear un entorno de pruebas dentro de la SPA para ver el rendimiento del protocolo IoT seleccionado.
- 4) Modificar una báscula digital para convertirla en una báscula IoT, con el fin de comprobar la eficacia de su comunicación con la SPA.

CAPÍTULO III. MARCO TEÓRICO

Según lo dicho por (Liu et al., 2020) Desde el primero de diciembre del 2019, la propagación de Covid-19 está aumentando cada día. Esta enfermedad, ahora considerada pandemia, ha impactado fuertemente la manera en la que se desarrollan las actividades laborales en todo el mundo. Esta situación ha impactado incluso a al sector cafetalero de Honduras que se concentra mayormente en el occidente del país, por lo que es necesario tomar acción y modificar el esquema de trabajo en la producción de este rubro que según (IHCAFE, 2017) representa el 38% del PIB agrícola del país. Por ello soluciones tecnológicas para este rubro tan importante en el país deben ser llevadas a cabo con la mayor velocidad y eficacia posible.

3.1 El café

El café es un grano fruto del árbol del cafeto, es uno de los principales motores económicos de muchas regiones de América Latina. Su consumo elevado en todo el mundo y el amplio espectro de sabores y utilidades que se le dan a este grano es lo que lo ha llevado al punto en el que se encuentra ahora mismo

3.1 Historia del café en Honduras

Honduras en la actualidad es el quinto mayor productor de café del mundo, y destaca principalmente por su calidad de café que ha sido uno de los rubros principales del país por más de 100 años. Se pueden encontrar registro de producción de café en Honduras desde el año 1778, cuando el café entró a Honduras por medio de su hermano centroamericano Guatemala. Desde los primeros años la calidad del café hondureño ya se hacía notar, tanto que Don Ramón Auguiano en 1804 redactó un informe a los reyes de España que en ese entonces aún gobernaban sobre el territorio. En dicho informe se el gobernador intendente escribió "El café es de tan excelente calidad como el Moka. Se cree que haya sido sembrado por algunos curiosos, porque a pocos se conoce en esta provincia y aunque se cosecha en poca cantidad, pudiera cosecharse en mucha, porque está experimentado que rinde en esta provincia el duplicado de su siembra, que en los países donde más se ha cultivado" este fragmento fue encontrado en una artículo de la revista (FórumCafé, 2018) sin embargo fue hasta la década de 1950 que el café empezó a tener un alza significativa en el territorio. Impulsado por caficultores nativos del País se crearon las primeras instituciones como la oficina del café, la cual se dedicaba a brindar información técnica para el cultivo del café. Una década más tarde

se creó la Asociación de productores de café (AHPROCAFE), fundada por productores de la zona occidente del país y en 1970 se creó el Instituto Nacional del Café (IHCAFE) institución encargada a la investigación orientada al café que ha logrado la introducción de múltiples variedades de café dentro del territorio hondureño. Actualmente el café hondureño se produce en 15 de los 18 departamentos del país y tiene 300,000 hectáreas cultivadas, da trabajo a 120,000 familias según el IHCAFE y es considerado como uno de los principales motores del desarrollo económico del país.

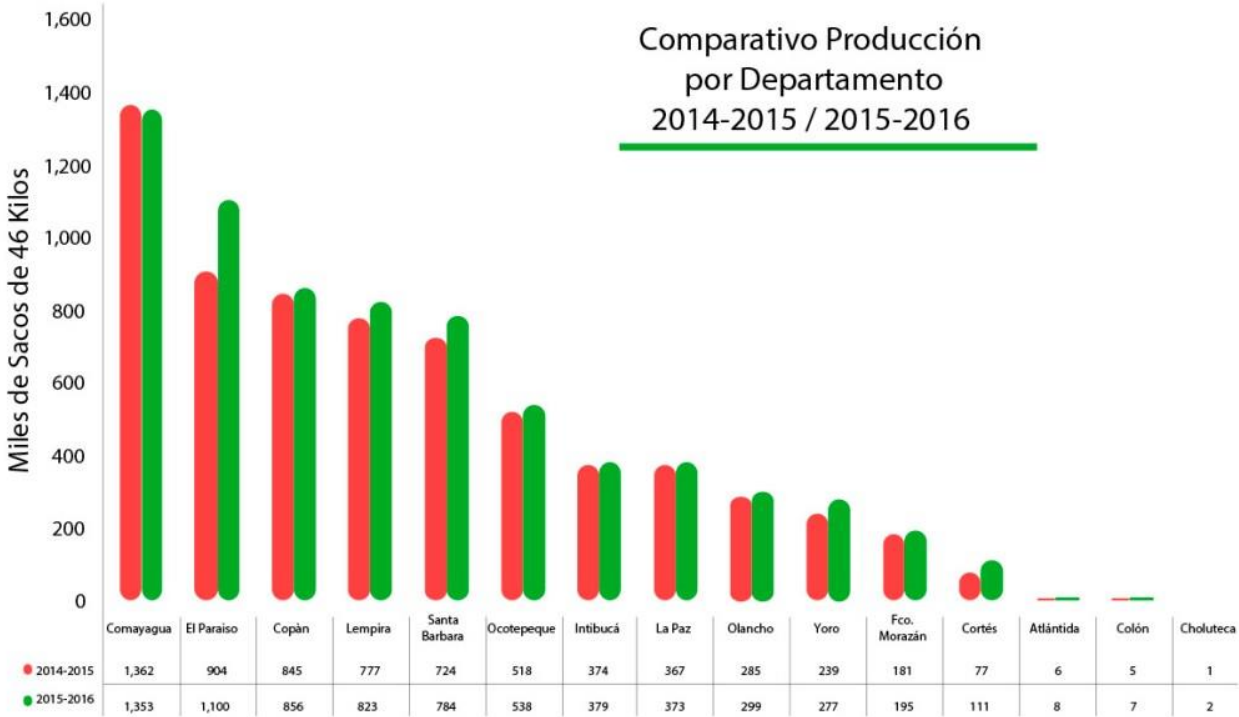


Fig. 1 Comparativo producción por departamento 2014-2015/2015-2016 (IHCAFE.s. f)

3.1.1 Método de recolección del café hondureño en la actualidad

En la actualidad el café producido en la mayoría de las zonas rurales es recolectado a mano en alturas de 1300mts en adelante, siendo la época alta de recolección en los meses de octubre a marzo; durante estos meses múltiples familias de distintas aldeas viajan a lugares como Rio Frio en busca de fincas que requieran mano de obra para la recolección. Esta recolección se hace en grupos de hasta 50 personas y cada individuo va depositando el café recolectado en sacos que pueden contener el equivalente a 50 Lb. Dichas personas al terminar el día laboral se reúnen junto al capataz de la finca para medir el volumen recolectado por cada individuo, este proceso se realiza a mano

y casi todos los productores de la zona realizan esta tarea todos los días junto a las personas que aportaron en la recolección. Realizada esta recolección y medición el productor tiene 2 alternativas, procesar su café o venderlo a alguna planta de beneficiado. Las plantas de beneficiado se encargan de procesar el café para lograr destacar las mejores características del grano; sin embargo, no todos los productores tienen la maquinaria o el conocimiento para llevar a cabo esta tarea y por ello muchos productores, viajan distancias de alrededor de 10Km para entregar su café a la planta de beneficiado que ellos prefieran. Una vez más, en dicha planta de beneficiado se mide el café por masa, colocándolos sacos en básculas que en su mayoría aún son análoga. Este proceso es igualmente realizado a mano y en algunas plantas de beneficiado se requieren alrededor de 5 personas dedicadas a mover y vaciar los sacos en sifones de selección. Junto a esas 5 personas hay una o dos personas encargadas de realizar notas de ingreso en el cual se anota, el nombre del productor, el peso total del café entregado, el tipo de café y el estado del café. En resumen durante estos procesos se produce un gran movimiento de personas de distintos lugares que terminan reuniéndose debido a la necesidad de entregar su café.

3.1.2 Métodos de recolección del café en otros países

En Colombia según (Cenicafe, 2016) la mayoría de la recolección se hace a mano por lo que la cosecha y el beneficio representan el 60% de los costos de totales. En Colombia también es muy común la migración entre departamentos, las grandes cantidades de personas trabajando juntas y las recolecciones en pequeñas fincas. Otra característica que comparte con Honduras son los meses de trabajo siendo en promedio de 5 a 6 meses. Otros datos a destacar de la recolección en Colombia se pueden encontrar en el informe de (Federación nacional de cafetaleros de Colombia, 2016) el 85.8% de los recolectores de café son hombres, aunque estos porcentajes varían de departamento en departamento, el mayor porcentaje de mujeres recolectoras las podemos encontrar en el departamento de Nariño con un 27.4%; en cuanto a la edad solo el 53% de los recolectores son menores de 40 años, esto se traduce en que la experiencia promedio de los recolectores en Colombia es de 22 años en promedio para los hombres y 17.5 años para las mujeres; el porcentaje de migración en algunos sectores del país llega hasta el 30% y el 35% de los recolectores no han terminado su primaria.

En Brasil la recolección se hace mayoritariamente mecanizada cuando alrededor del 75% de la producción se encuentra completamente madura, esto significa que el porcentaje de mano de obra utilizada en la recolección es significativamente menor que la de otros países latinoamericanos.

En Guatemala también se opta por la recolección manual y puede llegar a representar hasta un 70% de los costos de producción del café oro del país según un informe de (Juarez, 2018) sin embargo es importa no confundir mecanización con calidad afirma el escritor del informe

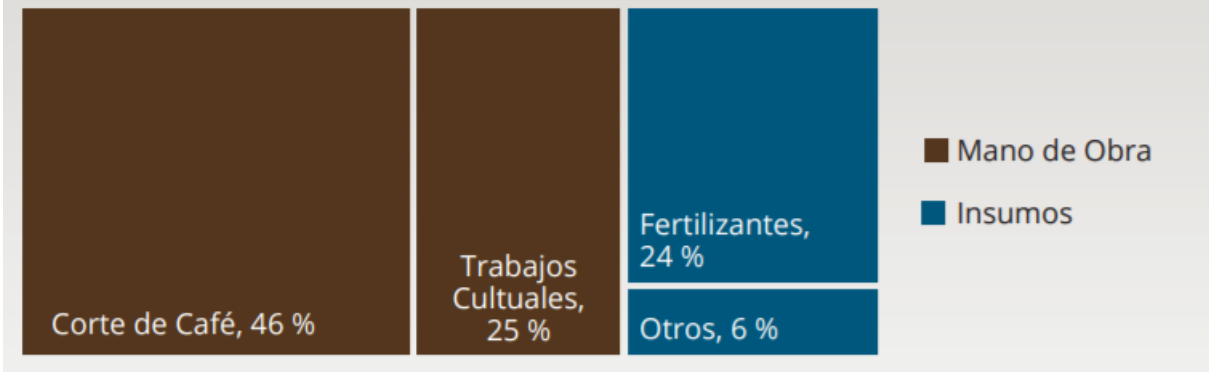


Fig. 2 Desglose de los costos de producción de un quintal oro (Juarez,2018)

En Costa Rica también comparte muchas características con sus hermanos centroamericanos y la mayor parte de su producción se hace de manera manual pero tienen algunos matices que la diferencian. Por ejemplo en la zona de los Santos, donde se produce alrededor del 30% del café de exportación con 9,336 hectáreas cultivadas, desde hace 3 décadas la recolección ha sido solventada por migrantes nicaragüenses e indígenas panameños según dice (Bolaños & Partanen, s. f.) en su investigación. Dichos extranjeros realizan migraciones todos los años y constituyen alrededor de 10mil y 12mil individuos en jornadas laborales de hasta 10 horas, siguiendo un esquema de recolección muy parecido al de Honduras actualmente. En costa rica se puede ver una clara diferenciación de clases sociales en el rubro del café y se ha planteado muchas veces la búsqueda de mejores condiciones laborales de dichos jornaleros pero no se ha tenido mucho éxito informa el investigador.

3.1.3 Importancia del café en Honduras

Como se mencionó con anterioridad el Instituto Hondureño del Café (IHCAFE) en el 2017 reportó que el producto interno bruto perteneciente al sector agrícola es constituido en un 38% por el sector

caficultor, da trabajo a más de 120,000 familias y es un sector en el que siempre se trata de innovar. En el 2003 se inició la primera Escuela de catadores patrocinada por IHCAFE, este programa instruyó a más de 50 catadores para que se empezara a desarrollar la caficultura de calidad dentro del país.

Nombre de la Escuela	Promoción	Módulos	Participantes Graduados
Catadores	IV°	24	35
Mecánicos Rurales	II°	10	12
Asistentes Administrativos	VII°	9	28
Gerentes Empresas	VII°	4	25
Administradores de Empresas	II°	9	15
Administradores de Finca	VII°	7	105
Diplomado Trinacional Caficultura Innovadora	I°	5	37
TUCCC	III°	24	12
TOTAL			269

Fig. 3 Resultado total de la escuela de catadores hasta la fecha (IHCAFE)

Dando la oportunidad a muchas personas desarrollar una carrera profesional en el rubro sin tener que dedicarse exclusivamente a la producción de café.

En el aspecto macroeconómico según indica (Diaz et al., 2018) en su tesis de pregrado de la EAP Zamorano: “Las ventas de café de Honduras generan 20-25% de las divisas del país. Dichas exportaciones de café se dirigen por un 62% hacia Europa (Alemania, Bélgica, Países Bajos) y por un 29% a los EE. UU, 4% a Asia y 5% a otros destinos. Los cafés diferenciados representan más del 20% de las exportaciones (2015/16) y crecieron hasta 30% (cifras preliminares ciclo 2016/17), con 7% FairTrade, 3.5% Orgánico (en parte combinado con otros sellos) y 4.7 % Utz/RFA. Además, el mercado interno del café demuestra marcado dinamismo, con el lanzamiento de nuevas marcas, la operación de (cadenas de) cafetines (Espresso americano con 180 lugares de venta) y el entrenamiento de una nueva generación de catadores y baristas” (Pxiii).

Respecto a esto es importante agregar que el café constituye el mayor producto de exportación de honduras, representando un 20.7% del producto total exportado según (Romero & Sibrian, 2016)

Es por estas razones que el café se mantiene como uno de los temas de mayor importancia en el país pero a pesar de que sea una gran fuente de investigación y desarrollo el enfoque que se le da a todos estos estudios es primordialmente desde un punto de vista agrícola y no relacionado con la creación de nuevas tecnologías que ayuden a impulsar la producción.

3.1.4 El panorama del café en honduras respecto a la pandemia de Covid-19

La pandemia de Covid-19 ha reducido la actividad humana en todos los rubros y en todos los países. Esto inevitablemente llevará a un impacto en la producción de café en Honduras. Según un informe de (International Coffee Organization, 2020) el coronavirus es una crisis en la salud pública que impactará las economías alrededor del mundo, por lo que es muy probable que esta crisis tenga un impacto profundo en el sector café a nivel mundial ya que se reducirá la demanda y las medidas de distanciamiento social tendrán un impacto negativo en la capacidad de producción.

Aunque el futuro aún es incierto y se tendrá que seguir investigando el tema a medida que este se desarrolle en el tiempo, son advertencias que exigirán que el país se prepare para respetar la medida de distanciamiento social mientras se intenta mantener la producción de café en la medida de lo posible.

3.2 Covid-19

Según (OMS, s. f.-b) es la enfermedad infecciosa causada por el coronavirus que se ha descubierto más recientemente. Tanto el virus como la enfermedad ya eran conocidos desde que los brotes empezaron en Wuhan (China) en diciembre del 2019. Actualmente la Covid-2019 es una pandemia que afecta a muchos países del mundo.

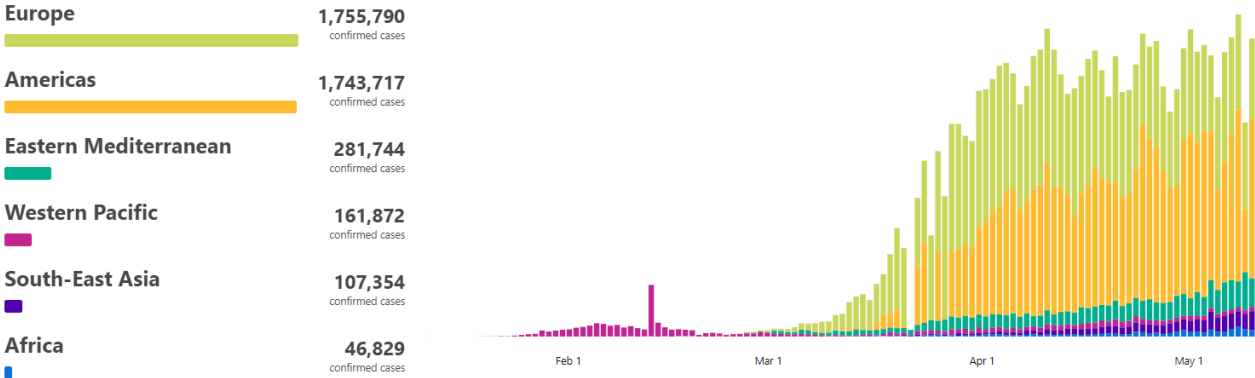


Fig. 4 Comparación de casos a lo largo del tiempo entre distintas regiones. (OMS)

3.2.1 Los Coronavirus

Son una extensa familia de virus en el que algunos son capaces de producir enfermedades en animales, incluidos los humanos. En el específico caso de los humanos, muchos coronavirus han sido los causantes de varias enfermedades que producen infecciones respiratorias que varían desde ser tan leves como el resfriado común hasta enfermedades más graves como el síndrome respiratorio de Oriente Medio (MERS) y el síndrome agudo severo (SRAS). Recientemente se ha descubierto que es un coronavirus el causante de la enfermedad COVID-19

3.2.2 Síntomas

Según (OMS, s. f.-b) Los síntomas más habituales son fiebre, tos seca y cansancio. Otros síntomas menos frecuentes son dolores y molestias, congestión nasal, dolor de cabeza, conjuntivitis, dolor de garganta, diarrea, pérdida del gusto u olfato y erupciones cutáneas o cambios de color en los dedos de manos o pies. Estos síntomas usualmente suelen ser leves y comienzan gradualmente.

Alrededor del 80% se recupera de la enfermedad sin necesidad de atender al hospital. Sin embargo; el otro 20% presenta un cuadro grave y experimentan enfermedades respiratorias. Las personas mayores y las que padecen de otras enfermedades previas como hipertensión arterial, problemas cardiacos o pulmonares, diabetes o cáncer tiene una mayor probabilidad de presentar los síntomas más graves. Es importante recordad que cualquier persona puede contraer COVID-19 y caer gravemente enferma aún si no presenta ningún agravante de los mencionados, es por eso por lo que sin importar la edad o la condición física; en el preciso momento en el que se empiece a experimentar fiebre, tos, dificultad al respirar o incapacidad hablar o moverse deberían de solicitar atención médica inmediatamente.

3.2.3 Propagación

Según (OMS, s. f.-b) la COVID-19 se puede contraer al tener contacto con otra que esté infectada. El principal medio de propagación son las gotículas que producen las personas infectadas al toser, estornudar y hablar. Estas gotículas son relativamente pesadas y no viajan grandes distancias y caen rápidamente al suelo. Una persona puede contraer la enfermedad al inhalar las gotículas proveniente de una persona infectada. Por eso se debe de mantener una distancia de mínimo un metro a cualquier otra persona. Las gotículas muchas veces caen en objetos y superficies como mesas, pomos y barandillas, de modo que otras personas se pueden infectar al tocar dichas superficies y luego llevar sus manos a sus ojos, nariz o boca. La OMS se encuentra investigando

otras probabilidades de contagio, sin embargo al momento de escribir esta tesis los anteriores medios de contagio son los conocidos y comprobados.

3.2.4 Medidas de prevención

Según (OMS, s. f.-a) las principales medidas de prevención que se pueden tomar son:

- Lavado frecuente de manos con agua y jabón o desinfectante ya que esto mata al virus en el caso de que se encuentre en sus manos.

- Adoptar medidas de higiene respiratoria, al toser o estornudar, cúbrase la boca y nariz con el codo flexionado o con un pañuelo, luego el pañuelo debe ser tirado inmediatamente y a continuación realizar un lavado de manos con un desinfectante o con agua y jabón. Esto es recomendado porque reduce el contacto de los fluidos del estornudo con el entorno.

- Mantener el distanciamiento social de mínimo un metro respecto a cualquier otra persona particularmente a aquellas que tosan, estornuden y tengan fiebre ya que es muy probable que estas personas tengan el virus y un contacto demasiado cercano con ellos puede resultar en contagio.

- Evite tocarse los ojos, nariz y la boca debido a la alta probabilidad de que nuestras manos toquen alguna superficie contaminada con el virus, y si luego de estar en contacto con estas superficies se llevan las manos a la cara se puede transferir el virus encontrado en la superficie a uno mismo.

- La organización mundial de la salud advierte que probablemente se encuentren nuevas medidas de prevención o se modifiquen las existentes debido a la investigación constante respecto a la enfermedad.

3.2.5 Tratamiento

Según la (OMS, s. f.-b) a pesar de que algunas soluciones médicas o remedios caseros pueden ayudar a aliviar los síntomas leves de la COVID-19, hasta ahora no hay medicamentos que hayan sido demostrados científicamente en la labor de prevenir, o curar la enfermedad. Por lo que se recomienda abstenerse de la automedicación de cualquier fármaco, sin embargo son muchos los ensayos clínicos que se están llevando a cabo en todos los ámbitos de la medicina ya sea para curar, prevenir o apalar las consecuencias de la enfermedad. Debido a la falta de un tratamiento eficaz lo más recomendable para mantenerse a salvo del virus es seguir las medidas de prevención.

3.3 El Internet de las cosas (IoT)

El internet de las cosas (IoT) nos muestra un mundo en el que los dispositivos que lo construyen pueden ser identificados en el Internet y por lo tanto crece con cada nuevo dispositivo que es agregado, debido a su popularidad la cantidad de dispositivos conectados crece a un ritmo

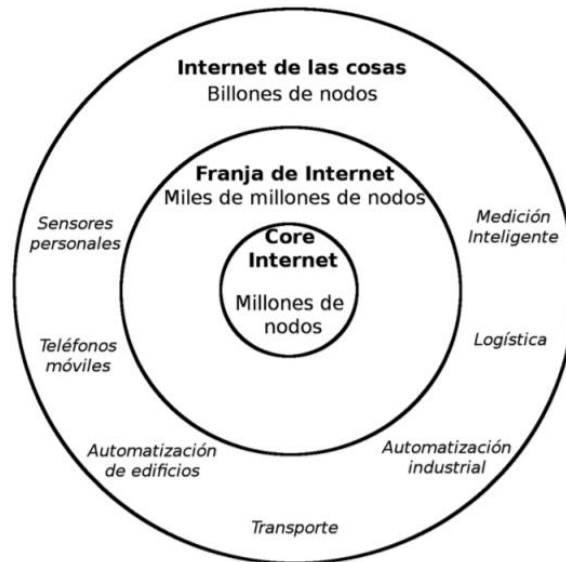


Fig. 5 Visión del internet de las cosas(IoT)... (Cama, 2012)

acelerado. De esta manera, las redes de sensores inalámbricos tienen un papel fundamental que incrementa la ubicuidad de las redes inteligentes de costos bajos e implementación fácil. Los protocolos desarrollados para el IoT están pensados para permitirnos la integración de nuevos elementos en nuestros lugares de trabajo, ciudades inteligentes y hogares también nos permiten mejorar la manera en la que obtenemos datos sobre la salud del usuario, brinda nuevas maneras de implementar seguridad dentro de las empresas, ayuda a recopilar datos del ambiente y de ciudades inteligentes.

Debido a que esta es una tecnología nueva, hay muchas cosas que aún queda por construir como indica (Cama, 2012) En la actualidad no hay un algo que determine que el internet de las cosas tenga que estar sujeta a ciertas tecnologías o protocolos, es por eso que muchos investigadores lo definen como la integración de dispositivos al internet para que estos se puedan comunicar y entregar sus datos recopilados sin necesidad de que un humano interactúe con ellos

Y es por esta razón que es posible que se genere algún tipo de confusión a la hora de hablar de los protocolos existentes, pues son muy variados y algunos cumplen funciones similares pero no idénticas.

3.3.1 Protocolos del Internet de las cosas

El internet de las cosas es un concepto que puede ser llevado a cabo de muchas maneras distintas, no es necesario seguir un camino en específico para llegar a obtener la comunicación M2M, por eso a medida que esta idea ha sido aplicada en distintos lugares, han surgido muchos protocolos de comunicación que llevan a cabo la misma tarea pero de maneras distintas. En este escenario podemos encontrar 4 protocolos de comunicación con los que se puede formar una red de dispositivos interconectados, algunos pueden conformar una red IoT por si solo y otros apoyan al concepto para permitir la construcción de redes IoT más sofisticadas.

3.3.1.1 Message Queuing Telemetry Transport (MQTT)

Es un protocolo de mensajería usada en la comunicación máquina-máquina (M2M) o también llamado IoT. MQTT está diseñado para anchos de banda bajos, de baja latencia y permite la comunicación de múltiples dispositivos. Este protocolo es particularmente útil porque como indica (Sadio et al., 2019) *“La mayoría de los dispositivos del internet de las cosas tienen limitaciones en su capacidad de procesamiento, energía y memoria, por lo tanto, es importante el uso de un protocolo que pueda ejecutarse bajo estas restricciones”*.

MQTT es un servicio de mensajería tipo push usando una lógica de publicador/suscripto, esto significa que los clientes se conectan con un servidor central llamado broker que se encarga de distribuir los mensajes.

Para filtrar los mensajes que son enviados a cada cliente los mensajes se disponen en tópicos organizados jerárquicamente. Un cliente puede publicar un mensaje en un determinado tópico. Otros clientes pueden suscribirse a este tópico, y el broker le hará llegar los mensajes suscritos.

Para lograr esto, los dispositivos se conectan utilizando TCP/IP con el broker, así este puede mantener registro sobre los dispositivos conectados. Esta conexión se mantiene abierta hasta que el dispositivo decide terminarla.

Estas características de mantener la sesión abierta y lista para la comunicación en cualquier momento es lo que nos permite tener una comunicación en tiempo real en el dispositivo pues a diferencia de otros protocolos el dispositivo no tiene que estar constantemente

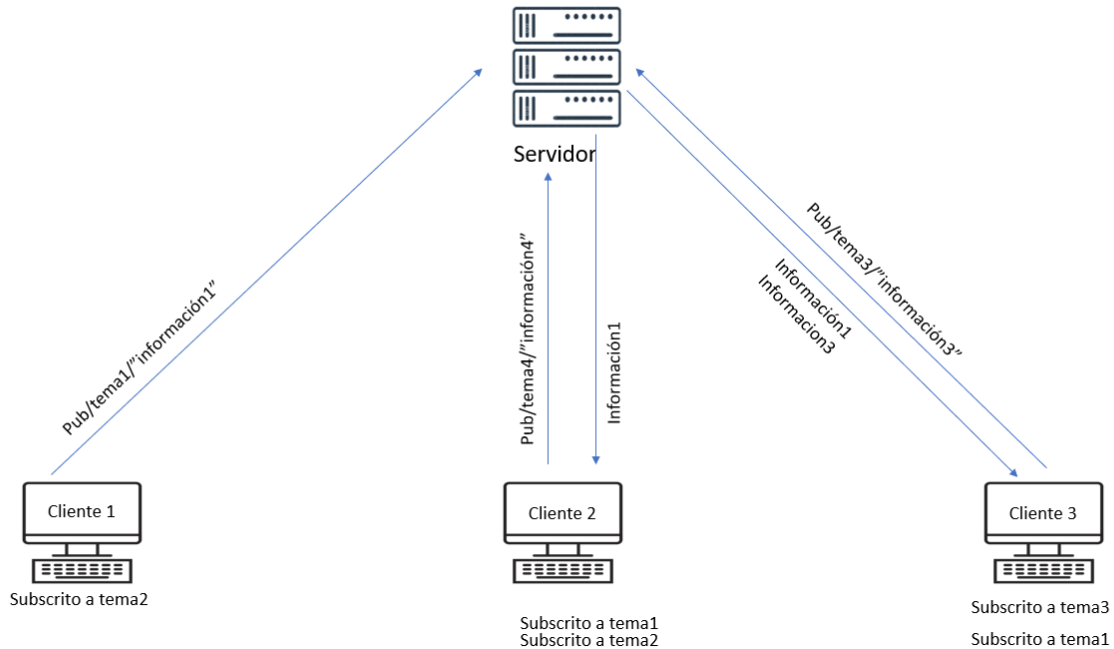


Fig. 6 Representación del esquema pub/sub (autoría propia)

La seguridad del protocolo MQTT es algo que despierta muchas interrogantes a la hora de construir un sistema robusto de IoT, de acuerdo a la documentación en línea y las palabras de (Andy et al., 2017) el protocolo MQTT solo provee de autenticación como mecanismo de seguridad por defecto, no encripta los datos ni el tráfico por lo tanto la privacidad de dichos datos, la integridad de los mismo e incluso el sistema de autenticación se pueden ver comprometidos. Por suerte Mosquitto agrega una pequeña capa de seguridad a sus mensajes, sin embargo a la hora de trabajar con datos extremadamente sensibles se recomienda implementar capas de seguridad adicionales en la medida que los dispositivos de la red IoT se lo permita.

3.3.1.1.1 Eclipse Mosquitto

Mosquitto es una implementación de código abierto de la versión 5.0 del protocolo MQTT, incluye las librerías necesarias para trabajar en C y las librerías con las utilidades mosquitto_pub y mosquitto_sub para publicaciones y suscripciones dentro del protocolo. Tiene la cualidad de tener librerías de terceros que facilita su implementación dentro de JavaScript y gracias a su soporte nativo en C también facilita su implementación en microcontroladores, otra ventaja de Mosquitto es la que menciona (Light, 2017) es muy útil para el envío de mensajes livianos entre dispositivos que se ven muy limitados en términos de capacidad de procesamiento y energía

3.3.1.2 Advanced Message Queuing Protocol (AMQP)

Es un protocolo de la capa de aplicación que se fundamenta en la comunicación proceso a proceso por medio de redes de IP. Esta idea permite que procesos que provienen de dos servidores diferentes se puedan comunicar sin importar la tecnología usada. En otras palabras el objetivo de AMQP es poder enviar mensajes entre brókeres por medio de TCP/IP.

Este protocolo de comunicación es una opción que compite directamente con MQTT pues ambos tratan de lograr el mismo objetivo, la diferencia clave entre estos dos protocolos es que AMQP envía los datos encriptados y funciona de manera asíncrona por lo que los mensajes pueden ser enviados en paralelo y se mantienen a la espera hasta que el cliente esté listo para recibir el mensaje.

3.3.1.2.1 RabbitMQ

RabbitMQ es un broker que integra el protocolo AMQP y nos permite tener acceso a toda la información que se transmite por medio de este protocolo, facilita la implementación de otras tecnologías, brinda soporte técnico y mejora la experiencia del usuario.

En el libro escrito por (Ayanoglu et al., 2016) define RabbitMQ como un broker de mensajería de código abierto que implementa AMQP y que asegura que los mensajes serán enviados y recibidos, los mensajes llegaran a su destinatario correcto, hace un guardado de seguridad de los mensajes, tiene un amplio potencial de escalado y puede extender sus funcionalidades por medio de plugins

3.3.1.3 Bluetooth Low-Energy (BLE)

Es una red inalámbrica personal pensada inicialmente para el uso en wearables de uso médico para compartir información entre dispositivos, este protocolo no es el mismo Bluetooth BR/EDR que es utilizado para dispositivos de entretenimiento, seguridad o de rendimiento; sin embargo ambas redes pueden coexistir pero no son compatibles. Este protocolo fue creado en el 2006 por Nokia pero implementado en el mercado hasta el 2009.

Comparado con el bluetooth original, esta versión de bluetooth de baja energía está diseñado precisamente para reducir considerablemente el consumo de energía del protocolo bluetooth original mientras se mantiene casi el mismo rango de comunicación. Los nuevos sistemas operativos como windows10, IOS12, Android10 y distribuciones recientes de bluetooth soportan nativamente este protocolo.

A pesar de sus interesantes características y su amplia adopción en dispositivos modernos BLE tiene ciertos problemas intrínsecos a modo de operación como lo dice (Gaoyang Shan et al., 2016) en el IoT, las puertas de enlace BLE deben ser descubiertas por los dispositivos BLE que lo rodean en poco tiempo. Cabe recalcar que esto es muy difícil de lograr cuando el número de dispositivos alrededor se vuelve demasiado grande. A medida que la cantidad de dispositivos crece, la densidad de señales se incrementa, y las colisiones se empiezan a volver un serio problema. Estas palabras conforman una gran advertencia a la hora de utilizar este protocolo como parte del IoT pues como se ve en la imagen a continuación encontrada en la investigación de (Mendieta et al., 2019).

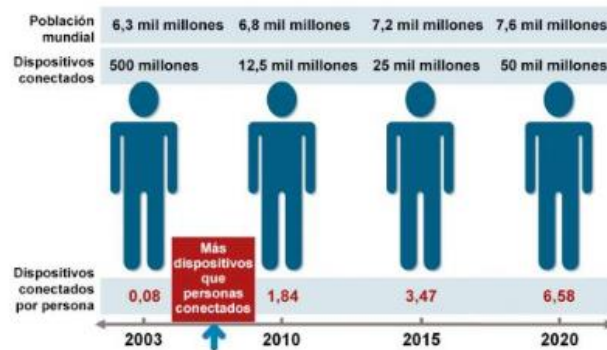


Fig. 7 Ritmo de crecimiento de los dispositivos IoT (Gaoyang Shan et al., 2016)

3.3.1.4 Zigbee

Según la traducción de la página oficial de Zigbee podemos definir este protocolo como un estándar global para las redes en malla de bajo consumo y regenerativa que ofrece una interoperabilidad para los dispositivos IoT pensados para la automatización de hogares y edificios. La red Zigbee puede soportar cientos de dispositivos y tiene características avanzadas de seguridad.

Dicho esto, a pesar de que el protocolo Zigbee no es un protocolo únicamente pensado para el IoT este es muy útil pues la conexión de cientos de dispositivos que comparten su información de manera coordinada y sin competir con los otros dispositivos de una misma red Wi-Fi reducen la congestión en la señales y la colisiones se vuelven menos comunes ya que el objetivo último de las redes Zigbee es llevar toda la información de cada nodo a un dispositivo final que es el único que necesita una conexión a internet directamente.

El aspecto en el que mejor se desarrolla Zigbee es en la creación de redes de sensores que nos permiten recolectar información del mundo real con una resolución nunca antes vistas, gracias a que su bajo consumo de energía, baja demanda de capacidad de procesamiento y la nula necesidad

de cables; nos permite desplegar una amplia cantidad de sensores simples que funcionan con baterías, dicha información luego puede ser enviada por el coordinador hacia al dispositivo final que comúnmente no está sujeto a las mismas limitaciones permitiendo así tomar decisiones útiles.

Se tiene que mencionar que Zigbee no se encarga de la comunicación del dispositivo final con el internet, así que no puede ser considerado como único protocolo para la generación de redes IoT pero permite potenciar las capacidades de estas redes al condensar la información de la gran cantidad de dispositivos que esta red soporta.

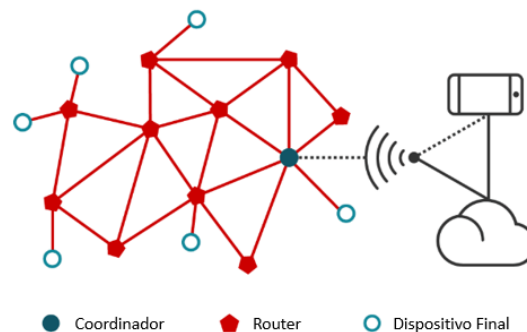


Fig. 8 Esquema de una red Zigbee

3.3.2 Dispositivos en la red IoT

La ventaja de las redes IoT es que cualquier dispositivo con compatibilidad con capacidad de conexión vía wifi puede unirse a la red pudiendo publicar y suscribirse a distintos temas, gracias a esto microcontroladores, computadoras de escritorio, computadoras portátiles, tablets y smartphones pueden comunicarse sin mucho esfuerzo siempre y cuando el tamaño de los datos a enviar sea razonable.

3.3.2.1 Microcontroladores

Los microcontroladores se encuentran en muchos de los dispositivos utilizados en el día a día y cada y son esenciales a la hora de construir un dispositivo que se pueda conectar a una red IoT. Estos elementos nos han permitido según (Dhanalaxmi & Naidu, 2017) Todos los sistemas de IoT tiene que ser gobernados por al menos un microcontrolador ya sea de propósito general o específico.

La explosión de microcontroladores en los últimos años es lo que nos ha permitido llegar al punto de desarrollo tecnológico en el que nos encontramos, es por eso por lo que ahora en el mercado nos encontramos con muchas opciones para desarrollar nuestros proyectos obligándonos a tener que realizar una búsqueda del microcontrolador que se adapte de mejor manera a las necesidades

de nuestro proyecto. Nos enfocaremos en los 2 microcontroladores para el desarrollo de dispositivos IoT más conocidos, el esp2866 y su sucesor esp32.

3.3.2.1.1 Microcontrolador ESP2688

Es un microcontrolador de bajo costo y con una modesta capacidad de procesamiento, bajo consumo de energía, altamente compacto, de alta durabilidad; por estas razones es ideal para el desarrollo de los nuevos proyectos en IoT como lo dice (Valderrama & Brea, 2018) "El ESP8266 es un microcontrolador perteneciente a una nueva generación de computadores embebidos que ya no sólo limita sus comunicaciones a medios alámbricos y radio sino que también dispone de conexión inalámbrica a la Internet por WIFI."

Gracias a estas características y su facilidad a la hora de ser programado en C, es un gran candidato para trabajar junto al broker Mosquitto del cual se habló anteriormente.

Para propósitos de investigación lo mejor a la hora de trabajar con el microcontrolador esp2866 es no partir del microcontrolador en bruto, en su lugar, lo más recomendable sería partir de una placa de prototipado como lo es NodeMCU que es un proyecto de código abierto desarrollado para facilitar la implementación del microcontrolador esp2866 al IoT, NodeMCU es una de las placas de prototipado más populares que nos evita la carga de tener que desarrollar nuestra propia placa desde 0, además agrega 4Mb de memoria EEPROM para compensar los problemas de memoria del esp2866

3.3.2.1.2 Microcontrolador Esp32

Es el sucesor del Esp2866 por lo que comparten muchas de sus características. Esta diseñado usando la tecnología de ultra bajo consumo TSMC e incluye en un solo chip la capacidad de utilizar Wi-Fi de 2.4Ghz y Bluetooth, tiene un mejor desempeño y versatilidad que su antecesor.

Su conexión Bluetooth está certificada para trabajar con el protocolo BLE y por lo tanto crear redes de sensores que pueden soportar comunicaciones a distancias de más de 50 metros. Sin embargo su precio puede llegar a ser el doble de alto que el del esp2688, característica que es clave a la hora de crear una amplia red de IoT.

A pesar de su elevado precio respecto a su antecesor, muchas veces su gran versatilidad juega un rol importante a la hora de decidir qué microcontrolador utilizar, es por eso por lo que las comunidades de desarrollo de pequeños proyectos de IoT tienden a preferir este modelo.

Este potente microcontrolador también puede ser encontrado integrado en muchas placas de

prototipado que facilitan su manejo y proveen circuitos de protección para evitar la destrucción del microcontrolador en caso de un error mientras se están probando ideas.

3.4 Internet

En palabras de (Gonge & Kandalkar, 2015) De manera reciente, la red interconectada de computadoras para transmitir datos conocida como internet, ha sido ampliamente adoptada en el uso cotidiano. Como lo dice (Jaimez-González & Vargas-Rodríguez, 2017) el inmenso impacto que ha tenido internet en el mundo permitiendo que los negocios puedan ofrecer sus productos o servicio, desarrollar nuevas plataformas de comunicación y modificar radicalmente la manera en la que obtenemos información.

3.4.1 Hypertext Transfer Protocol (HTTP)

Es el protocolo utilizado para el envío de la mayoría de las páginas web en internet, y define el formato de estos documentos y como deben de ser transmitidos, de la misma manera este protocolo le dice a los navegadores y servidores cómo reaccionar ante ciertos comandos.

Http es un protocolo sin estado ya que cada comando es ejecutado sin tener conocimiento de los comandos ejecutados con anterioridad; por esta razón es tan difícil integrar interactividad dentro de los sitios web, por fortuna múltiples tecnologías se han implementado para solucionar este problema, los más populares Java, JavaScript, ActiveX y cookies.

Http funciona como un protocolo de pregunta-respuesta en el servidor y el cliente, siendo el cliente el dispositivo que ejecuta una aplicación y el servidor es el dispositivo en el que se aloja la página web, con esto en mente, el flujo de trabajo de HTTP es el siguiente: El cliente hace una petición HTTP al servidor, y de acuerdo al tipo de petición que se realizó el servidor ejecuta una acción u otra para luego devolver una respuesta que contiene el resultado de la petición, como lo dice (Aron, 2016) HTTP provee transporte, pero no define la presentación de la información. Así, un requerimiento HTTP puede contener HTML, JavaScript, JSON, XML y demás. Esta característica evita que el servidor pueda enviar información al cliente sin que este lo pida, en otras palabras, el servidor nunca enviará información que el cliente le haya pedido con anterioridad.

El ejemplo clásico de un cliente es el de un navegador web, como se mencionó con anterioridad, pero un cliente también puede ser un software de indexado, una aplicación móvil o cualquier pieza de código que es ejecutado por un microcontrolador.

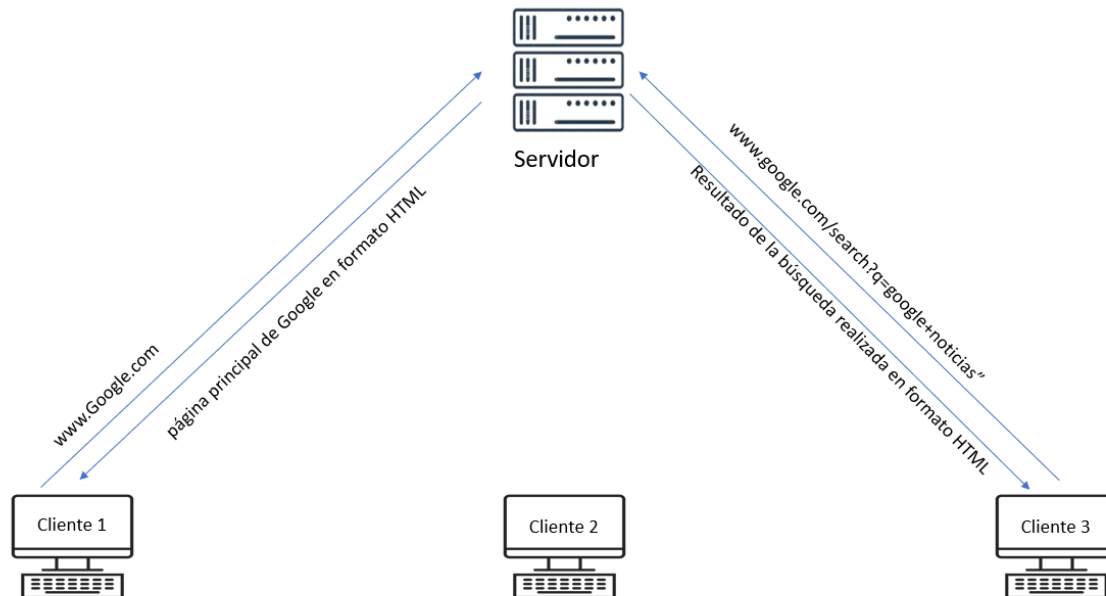


Fig. 9 Representación del funcionamiento del protocolo HTML (autoría propia)

3.4.1.1 Hypertext Transfer Protocol Secure (HTTPS)

Es una extensión del protocolo HTTP usado para las comunicaciones seguras entre computadoras dentro de la misma red y es por esta razón que es ampliamente utilizado ampliamente en internet. Para lograr la deseada comunicación segura, el protocolo utiliza un sistema de encriptado en la capa de transporte (*Transport Layer Security/ TLS*. Este protocolo es de gran importancia por la constante transmisión de información delicada a través de internet, sin embargo como dice (Krombholz et al., 2019) muchos usuarios comúnmente terminan confundidos con el tipo de encriptación utilizados y por lo tanto subestiman los beneficios de HTTPS.

3.4.1.2 Métodos de petición HTTP

HTTP define un grupo de métodos de petición para especificar la acción que se quiere que el servidor ejecute, de su misma manera distintos métodos de petición requieren de distintos parámetros que deben ser enviados por el cliente, los más utilizados son el método *GET*, *PUT*, *UPDATE*, y sus funciones son las de obtener una página HTML, enviar grandes datos al servidor y actualizar la información que el servidor tiene de la sesión HTML.

3.4.1.3 Cookies

Una *HTTP Cookie*, web cookie o simplemente cookie, como lo dice (Kwon et al., 2019) es una pequeña pieza de información enviada por la página web que es guardada dentro del cliente y es

utilizada para que la página web pueda recordad información útil sobre el usuario que se utilizan luego para personalizar aún más la experiencia del usuario o evitar el ingreso de información redundante cada vez que la página es refrescada. Estas son necesarias ya que como se mencionó anteriormente "Http es un protocolo sin estado ya que cada comando es ejecutado sin tener conocimiento de los comandos ejecutados con anterioridad" por lo que de no usarse estas a medida que se navega dentro de un mismo domino web se perdería la información cada vez que se va a una parte distinta de este domino.

3.5 Páginas web

Es un documento que contiene información en forma de texto, imágenes, video, enlaces y que se comparte por medio de la *World Wide Web* (WWW) y que puede ser accedida mediante un navegador web. Estos documentos están contruidos usando los lenguajes HTML, CSS Y JavaScript; cada uno de estos lenguajes se encarga de dar estructura, estilo y dinamismo a la página web.

Estos documentos comúnmente se encuentran alojados en un servidor que puede ser accedido desde cualquier parte del mundo siempre y cuando se disponga de una conexión a internet tanto por parte del cliente como por parte del servidor; sin embargo se puede restringir el acceso a estos documentos creando una intranet a la cual solo tienen accesos ciertos usuarios.

3.5.1 Hyper Text Markup Language (HTML)

Es el un lenguaje de marcado para la elaboración de páginas web. Consiste en una serie de elementos representados por etiquetas, y a través del uso de estas etiquetas se representa la estructura y el contenido estático de una página web. Cada elemento comúnmente está representado por una etiqueta de apertura, una etiqueta de cierre. son identificados por su contenido y el elemento al que afectan es el que se encuentra entre ambas etiquetas, por ejemplo para la etiqueta **h1** que es utilizada para dar énfasis a un texto las respectivas etiquetas de cierre y apertura serían: "`<h1> texto a resaltar </h1>`".

En su libro (Eguiluz, 2008) Hace una lista de las etiquetas actualmente aprobada para el uso en HTML y son: h1, h2, h3, h4, h5, h6, head, hr, html, i, iframe, img, input, ins, isindex, kbd, label, legend, li, link, map, meta, noframes, noscript, object, ol, optgroup, option, p, param, pre, q, s, samp, script, select, small, span, strike, strong, style, sub, sup, table, tbody, td, textarea, tfoot, th, thead, title, tr, tt, u, ul, var. Este conjunto de etiquetas son los que nos permiten crear nuestra página web, sin embargo estas muchas veces no logran dar con el resultado específico que estamos buscando, y

para eso nacen los atributos.

```
<parrafo>  
Contenido de texto con <importante>algunas palabras</importante> resaltadas de forma  
especial.  
</parrafo>
```

Fig. 10 Ejemplo del lenguaje de etiquetas HTML (eguiluz,2008)

3.5.1.1 Atributos de etiquetas

Los atributos de etiquetas proveen de información adicional a los elementos HTML, dicha información luego es interpretada por el navegador pudiendo tener diferentes resultados como darle un Identificador en específico a cada elemento HTML, modificar su clase CSS, modificar su estilo directamente o darle la capacidad de lanzar eventos JavaScript. Los atributos son escritos dentro de la de la etiqueta de apertura pero no en la de cierre siempre después del nombre del elemento, y el resultado se ve reflejado en el contenido del elemento

Elemento	Resultado
<p> Este es un párrafo </p>	Este es un párrafo
<p style="color:red"> Este es un párrafo en rojo </P>	Este es un párrafo en rojo
<p style="color:red;text-align:center" > Este es un párrafo en rojo centrado <p>	Este es un párrafo en rojo centrado

Fig. 11 Efecto de distintos atributos sobre un mismo elemento (autoría propia)

3.5.2 Cascading Style Sheets (CSS)

Cuando se empezó a explorar la construcción de páginas web en HTML estas solo se enfocaban en incluir el contenido en texto e imágenes. A medida que el estándar HTML, las páginas poco a poco empezaron a incluir su información dándole formato, cambiando el tipo de letra, colores y márgenes. Pero incluir en una misma página HTML el contenido, el diseño y la programación complica en gran manera el mantenimiento de dichas páginas ya que en gran parte de los casos esas 3 partes son trabajo de distintas personas, por lo que eventualmente se separaron.

En este contexto CSS es la manera en la que se logró separar el estilo del documento respecto a su contenido o programación.

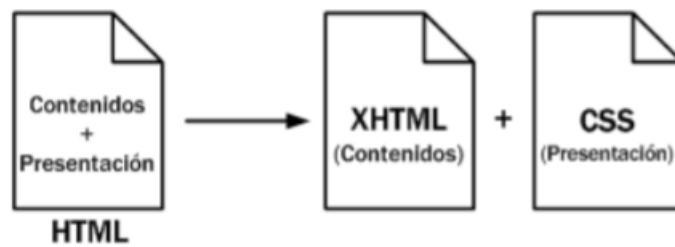


Fig. 12 Esquema de la separación de los contenidos y su presentación (Eguíluz, 2008)

Con CSS se le puede especificar a cada parte de un documento HTML un estilo en específico, se puede aplicar un estilo a todas las etiquetas de una página o se pueden aplicar solo a ciertos elementos. Con el avance del tiempo las páginas web se volvían más populares y se incrementó la complejidad del estilo, y se empezaron a crear frameworks y otras tecnologías alrededor de CSS para volver el trabajo de crear hojas de estilo más sencillo y reusable

3.5.2.1 stylesheet language (Sass)

Diseñado por Hampton Catlin y desarrollado por Natalie Weizanbaum, es un lenguaje de preprocesamiento basado en scripts que es compilado en CSS. Este lenguaje permite definir variables utilizando el signo "\$" y ":" para la asignación de valores a las variables.

Sass puede trabajar con números, cadenas de caracteres (Strings), colores (usando sus nombres) y booleanos. Dichas variables luego pueden ser usadas como argumentos para las distintas funciones que están disponibles que al ser ejecutada la compilación traduce dichas variables y funciones a lenguaje CSS original.

3.2.5.2 Leaner Style Sheets (Less)

Diseñado por Alexis Sellier, Less es un lenguaje de código abierto preprocesador de hojas de estilo dinámico que puede ser compilado en CSS y ser ejecutado desde el lado del cliente o del lado del servidor, otra característica destacable de Less es su capacidad de ser compilado en tiempo real por el navegador gracias al uso de less.js. Less hace uso de variables que son definidas con "@" y ":" para asignar un valor, utiliza mixins para reutilizar las propiedades de una clase dentro de otra y

también puede crear funciones que pueden ser interpretadas utilizando código JavaScript.

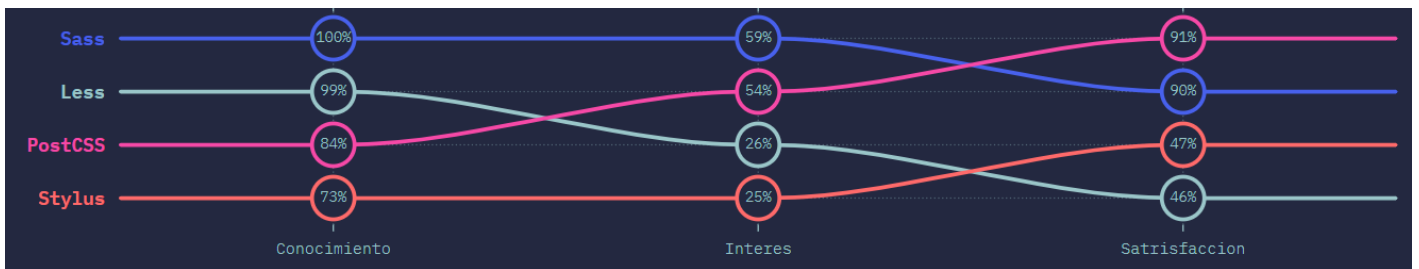


Fig. 13 Conocimiento, interés y satisfacción de los usuarios con los distintos preprocesadores CSS (Greif & Benitte, 2019)

3.5.3 JavaScript

Es un lenguaje interpretado de programación orientado a objetos. A nivel de sintaxis tiene muchas similitudes con C, C++ y Java. Sin embargo este es un lenguaje débilmente tipado por lo que las variables no tienen un tipo que las restrinja. Este lenguaje es muy utilizado en programación de páginas web, aunque muchas veces puede ser encontrado en la programación de aplicaciones externas.



Fig. 14 Rankin de los lenguajes de programación más utilizados. (Stack Overflow, 2019)

Según (Shen et al., 2018) JavaScript ha permitido que los desarrolladores pueda correr las líneas de código que ellos quieran en sus páginas web. Por eso en la actualidad todos los navegadores modernos pueden interpretar JavaScript y nos da la oportunidad de modificar el DOM de manera dinámica de acuerdo con la interacción del usuario con la página web. A medida que su popularidad aumentaba se empezó a utiliza JavaScript no solo dentro de las páginas operando del lado del cliente, siendo ahora común encontrarlo también operando del lado del servidor ayudándonos a enviar y recibir información entre ambas partes de la página. Las características principales del lenguaje son:

Es imperativo y estructurado, ya que usa sentencias *if*, bucles *for*, sentencias *switch*, etc. A partir del

2015 se permitió restringir el ámbito de las variables utilizando *let*, lo que permite que una variable solo tenga el alcance del bloque en el que está definido, de lo contrario la variable tendrá el alcance de toda la función en la que se definió.

Es de tipado dinámico como la mayoría de los lenguajes de scripting en los cuales el tipo no es asociado a la variable, en su lugar, el tipo es asignado al valor por lo que en cualquier momento una variable puede contener un valor de tipo *String* y en unas líneas de código después la misma variable puede contener un valor de tipo *entero*.

Es objetual, ya que está formado casi en su totalidad por objetos siendo estos definidos como arráis asociativos por lo que se pueden entender sus atributos como las llaves de dicho array, además dichos atributos pueden ser creados, eliminados o modificados en tiempo de ejecución.

3.5.4 Hypertext Preprocessor (PHP)

Es un lenguaje de programación de propósito general que opera del lado del servidor, fue uno de los primeros lenguajes de programación del lado del servidor ya que se podía incorporar directamente dentro de un documento HTML en lugar de tener que hacer llamadas a archivos ubicados de manera externa. Puede ser desplegado por la mayoría de los servidores web de la actualidad y a lo largo de todos los sistemas operativos sin incurrir en gasto alguno. Permitió por primera vez la implementación de páginas web dinámicas utilizando información almacenada en bases de datos, es bastante fácil de aprender, se puede conectar a la mayoría de los motores de bases de datos que se usan hoy en día, tiene una amplia documentación pues es uno de los lenguajes de programación más antiguos que se usan activamente cada día. Este último aspecto es muy importante pues al haber mantenido su popularidad durante mucho tiempo, es común encontrar líneas de código escritas en PHP en páginas muy antiguas. Con el tiempo se han desarrollado múltiples frameworks que han extendido mucho la utilización de PHP como lenguaje del lado del servidor, el verdadero inconveniente con esta tecnología es que sus características dinámicas son procesadas por el servidor, el cual evalúa el código PHP y lo convierte en elementos HTML por lo que el resultado es una página que no varía en sus resultados hasta que esta es recargada y por lo tanto es nuevamente evaluada por el servidor.

3.5.5 Single-page application (SPA)

Es una página web que trata de dar una experiencia lo más parecida a los programas nativos a los que se está acostumbrado a trabajar en dispositivos móviles y computadoras de escritorio. Para

lograr esto todo el contenido HTML, CSS y JavaScript se carga una sola vez o a medida que el usuario los vaya requiriendo. Esto resulta en que a la hora de tener que cargar nuevo contenido en el DOM o actualizar el ya existente no es necesario recargar la página completamente ya que dichas actualizaciones se hacen por medio de peticiones al servidor de manera asíncrona.

3.6 Bases de datos

Son un conjunto de datos que pertenecen al mismo contexto y son almacenados para darles uso en un evento futuro. Los primeros ejemplos de bases de datos son las conocidas bibliotecas, pues estas guardaban datos en forma de texto escrito y eran organizados para poder ser consultados cuando se les requiera. A medida que la tecnología se ha ido desarrollando, las bases de datos se han vuelto digitales lo que ha permitido construir múltiples técnicas a la hora de crearlas, organizarlas y trabajar con ellas

3.6.1 bases de datos digitales

Estas están formadas por registros que contienen información sobre un elemento base, en una base de datos utilizada como guía telefónica cada registro podría constar de número telefónico, nombre del dueño, llamadas realizadas, llamadas recibidas, etc. dichos campos son los que componen cada registro, estos campos guardan la información y muchas veces están restringidos a guardar solo un tipo de información. Una vez teniendo definidos que es un campo y que es un registro se puede definir el concepto de tabla que es simplemente un conjunto de registros que utilizan los mismos campos.

Y aunque estos conceptos por sí solos son simples, el conjunto de ellos y la relaciones que se pueden establecer entre campos, tablas y registros nos permiten diferenciar las bases de datos. Si clasificamos los datos de acuerdo con su contenido podemos encontrar:

Bases de datos bibliográficas, son las que estas construidas en base a una única fuente primaria que nos permite localizarla. Por ejemplo, se puede realizar una base de datos sobre una persona en la que se liste los logros de esta persona, el año en el que realizó dicho logro y el nivel de impacto social que ese logro produjo

Bases de datos de texto completo, estas almacenan fuentes primarias, por ejemplo, todos los artículos científicos que una universidad ha publicado

Directorios, estos proveen de información específica sobre empresas o personas.

Otra manera muy útil de clasificar las bases de datos es haciéndolo en base al modelo de

administración de datos.

- **Bases de datos jerárquicas**, de acuerdo con este modelo podemos organizar los datos en forma de árbol invertido en el que un nodo padre de información puede tener uno más hijos con información más específica. Este modelo es utilizado en casos en los que hay que guardar un gran volumen de información que comparten muchas características, pero se vuelve complicado a medida que se empiezan a mostrar redundancia en los datos.

- **Base de datos de red**, es bastante parecido al jerárquico pero en este caso es permitido que un mismo nodo tenga uno o más padres, esto soluciona el problema de la redundancia de los datos pero vuelve un poco más compleja su interpretación.

- **Bases de datos transaccionales**, su único fin es el de enviar y recibir datos a grandes velocidades, son muy poco comunes pero se utilizan principalmente en el entorno de análisis de calidad, datos de producción e industrial. En este tipo de base de datos se tiende a permitir alta redundancia y duplicación de datos pues no afectan a su objetivo principal de capturar y enviar datos a altas velocidades

- **Bases de datos relacionales**, este es el modelo más utilizado para representar problemas reales y administrar los datos dinámicamente, sus fundamentos fueron presentados en 1970 por Edgar Fran Codd y en poco tiempo se volvió el paradigma más popular en los modelos de bases de datos. Su idea principal es el uso de *relaciones* que pueden considerarse como un conjunto de datos denominado *tuplas*. Para poder sacar el máximo provecho de estas bases de datos se implementan las *consultas* que nos permiten recuperar la información de una manera muy flexible. Lo más común a la hora de crear bases de datos relacionales es utilizar el lenguaje SQL y se encuentra implementado en muchos motores de gestión de bases de datos

- **Bases de datos orientadas a objetos**, es uno de los modelos más recientes y se basa en el modelo informático de la programación orientada a objetos, y trata de guardar los datos como si fueran objetos e incorpora los conceptos importantes del paradigma de objetos como la encapsulación, herencia y polimorfismo.

3.6.2 Bases de datos relacionales

Son un tipo de base de datos digital que son llamadas así porque cumplen con el modelo relacional, y es el modelo más popular en la actualidad y llevó a la creación de sistemas de software dedicados a mantener bases de datos relacionales llamados, *relational database management system* (RDBMS)

que utilizan el lenguaje SQL para consultar y mantener las bases de datos. Todas estas bases de datos están formadas por múltiples tablas que no pueden tener un mismo nombre ni los mismos registros, además cada tabla necesita una llave primaria que identifica cada registro y debe de cumplir con la integridad de datos. Haciendo uso de las llaves primarias es que se logra crear una relación entre tablas.

3.6.2.1 Relaciones base

Para que una BDR cumpla su función necesita relaciones; las relaciones es la manera en la que se almacenan y se accede a los datos, en este contexto se puede entender a las tablas como *relaciones base*

3.6.2.2 Relaciones derivadas

Por otro lado tenemos las *relaciones derivadas* estas no almacenan datos, pero se calculan a partir de realizar operaciones relaciones y su implementación es llamada *vista* o *consulta*, esto es de especial utilidad porque representa el resultado de las operaciones como si fueran una sola tabla.

3.6.2.3 Restricciones y dominios

También llamadas reglas, nos permiten definir qué tipos de datos pueden ser guardados dentro de un campo y que tipo de datos no pueden ser guardados. Es una operación booleana pues, o se puede guardar o no se puede guardar. Las restricciones son parte fundamental de las bases de datos relacionales. Si nos enfocamos directamente en las restricciones podemos definir un nuevo concepto llamado *dominios* que representan el conjunto de valores que un campo puede aceptar, es importante notar que este concepto incluye mucho más que el concepto matemático de dominio, pues el *dominio* de un campo puede ser caracteres, cadenas de datos, fechas, etc.

3.6.2.4 Tipos de clave

Existen 2 tipos de clave, la clave primaria y la foránea. La primera es uno o más campos que son únicos para cada registro, y son utilizadas para identificar cada registro en específico además son la principal manera de relacionar datos de una tabla con otra. Este concepto de clave primaria es utilizado en otros ámbitos también, por lo tanto podríamos ver nuestro número de identidad como un tipo de clave primaria utilizada para nombrar a cada persona en específico y ninguna otra persona puede tener exactamente el mismo número de identidad que otra. La clave foránea surge cuando se crea una consulta que relaciona 2 tablas por medio de sus claves primarias, dichas claves primarias se ven reflejadas en el resultado y en dicho resultado ellas ya no son necesariamente

consideradas claves primarias, en su lugar pasan a llamarse claves secundarias y no es necesario que en la nueva tabla las claves foráneas cumplan el requisito de ser únicas.

3.6.3 Structured Query Language (SQL)

Es un lenguaje de dominio específico utilizados para administrar y recuperar información de los sistemas de gestión de bases de datos relacionales. Fue desarrollado por un grupo de la IBM en San José, California. Este lenguaje explota la flexibilidad y la potencia que tienen los sistemas relacionales, es considerado un lenguaje declarativo de alto nivel, lo que significa que en unas cuantas líneas de código se pueden crear programas altamente productivos que en otros lenguajes de programación serían necesarias muchas más instrucciones. Para lograr esto SQL integra, lenguaje de definición de datos, lenguaje interactivo de manipulación de datos, comandos para asegurar la integridad de los datos, comandos para la definición de vistas, comandos para especificar el comienzo y el final de una transacción, comandos para gestionar las autorizaciones de accesos a las relaciones o consulta y además SQL se puede integrar dentro de otros lenguajes de programación.

3.7 Frameworks

es una estructura conceptual de soporte definido que proveen de módulos de software concretos en la mayoría de los casos. Sirven como la base de desarrollo de un software facilitando la escritura de código. Esto significa que los frameworks nos dan partes de software que evita que tengamos que programar desde cero las funcionalidades básicas que muchos otros proyectos comparten, los frameworks también establecen buenas prácticas de programación como el MVC cuya principal función es ordenar el código y mantener ciertas partes de nuestro código centradas en tareas en específicas y no mezclarlas, puesto genera confusión a largo plazo.

Ya que los frameworks se construyen de manera colectiva y aumentan sus capacidades a lo largo del tiempo, permiten a sus usuarios crear aplicaciones que anteriormente no podrían haber creado por falta de tiempo o conocimiento, esto implica de igual manera que incluso los desarrolladores más avanzados pueden ver una considerable alza en la velocidad en la que terminan sus proyectos. En cuanto a la creación de aplicaciones web hay 2 grandes categorías en los que se clasifican los frameworks, front-end que son los que están dedicados a la construcción de una página web enfocándonos en el lado del cliente, y back-end que son los dedicados a la construcción de una página web enfocándose en el lado del servidor

3.7.1 Laravel

Es un framework back-end escrito en PHP, gracias a este framework la popularidad de PHP se ha mantenido a flote en el mundo del back-end ya que Laravel incluye características muy potentes respecto a la creación y manipulación de bases de datos gracias a su ORM integrado Eloquent quien se encarga de convertir código PHP en código SQL, facilitando su utilización con las clases PHP y optimizado con buenas prácticas SQL evitando en gran medida la inyección de código SQL que según (Sadeghian et al., 2013) la inyección SQL se ha vuelto de los ataques con el mayor rango en la lista de vulnerabilidades web. Laravel también implementa métodos de paginación y algoritmos de búsqueda de registros. Otro punto importante de Laravel es la implementación de un sistema robusto de seguridad, el cual ya toma en cuenta la necesidad de separar los permisos que tiene cada usuario a la hora de interactuar con la página web, incluye una buena integración con el framework front-end Vue.js y el framework CSS bootstrap4 diseñado para crear páginas web con la mentalidad "Móvil primero" en el cual nos aseguramos de que la página web sea fácil de usar en dispositivos móviles.

Laravel continua con su desarrollo y lanza una nueva versión cada año por lo que se mantiene siempre en la punta de la lanza en muchos aspectos, mejorando en seguridad, compatibilidad con otras tecnologías y brindando herramientas útiles a los desarrolladores para la creación de páginas web muy robustas del lado del servidor y facilitando la implementación de frameworks dedicados al lado del cliente.

3.7.2 Node.js

Es un framework back-end, aunque tiene muchas funcionalidades de parte del cliente también, que se basa en el lenguaje de programación JavaScript, funciona de manera asíncrona y utiliza una estructura de datos orientada a objetos provista por el motor V8 de Google. Node.js no incluye tantas funcionalidades como Laravel, pero es justo eso lo que le da una de sus principales características, y es facilidad de implementación de módulos de terceros y al ser escrito en JavaScript tiene una de las bases de programadores más grandes, y en la actualidad, la más popular. Otra de sus principales características es que este framework funciona tanto del lado del cliente como del lado del servidor lo que hace más fácil la programación full-Stack además constantemente se desarrollan nuevas tecnologías destinadas a trabajar con facilidad junto a Node.js.

3.7.3 Vue.js

Es un framework JavaScript front-end que está pensada para crear la parte visual de una página web funciona de manera modular usando un esquema de componentes, esto significa que cada componente se representa como un elemento HTML, pero que es más que ello ya que un componente Vue incluye su propia hojas de estilos y su propia lógica de programación JavaScript por lo que cada componente se vuelve una parte de la página por si sola. Además este esquema de trabajo permite que haya personas completamente dedicadas a la fabricación de componentes de manera aislada, que pueden ser reutilizadas muchas veces a lo largo del tiempo y aplicadas en páginas web completamente distintas siempre y cuando se use Vue.js como base

3.7.4 React.js

Creado por el equipo de Facebook sirve para desarrollar aplicaciones web de una manera ordenada y con reutilización de código, está escrito en JavaScript y su principal característica es la manipulación del DOM de una página web. React.js Asocia las vistas a los datos logrando que en caso de que ocurra una modificación en los datos, estas modificaciones tengan un efecto en la página web. Al igual que Vue.js utiliza componentes que encapsula el funcionamiento necesario para tener una experiencia completa, cada componente incluye su propia hoja de estilos, código HTML y lógica JavaScript. Otro punto importante a la hora de trabajar con React es su amplio ecosistema de trabajo como Redux o Flux que imbuyen de nuevas habilidades a React pero la más potente de sus aplicaciones externas es React Native que se encarga de llevar una aplicación web escrita en React como una aplicación nativa para dispositivos con Android o IOS, esto ayuda mucho al extender el alcance de nuestra aplicación pudiendo llegar a usuarios que de otra manera se quedarían de lado.

3.7.5 Angular

Es un framework desarrollado en TypeScript, de código abierto y mantenido por Google, su principal objetivo es del crear SPA utilizando el modelo Vista Controlador, al igual que Vue.js y React.js, basa su funcionamiento en componentes que funcionan de manera modular y por lo tanto independiente unos de otros. Aunque ha ido perdiendo su popularidad su importancia en el mundo de la

programación aún es relevante en muchos sentidos

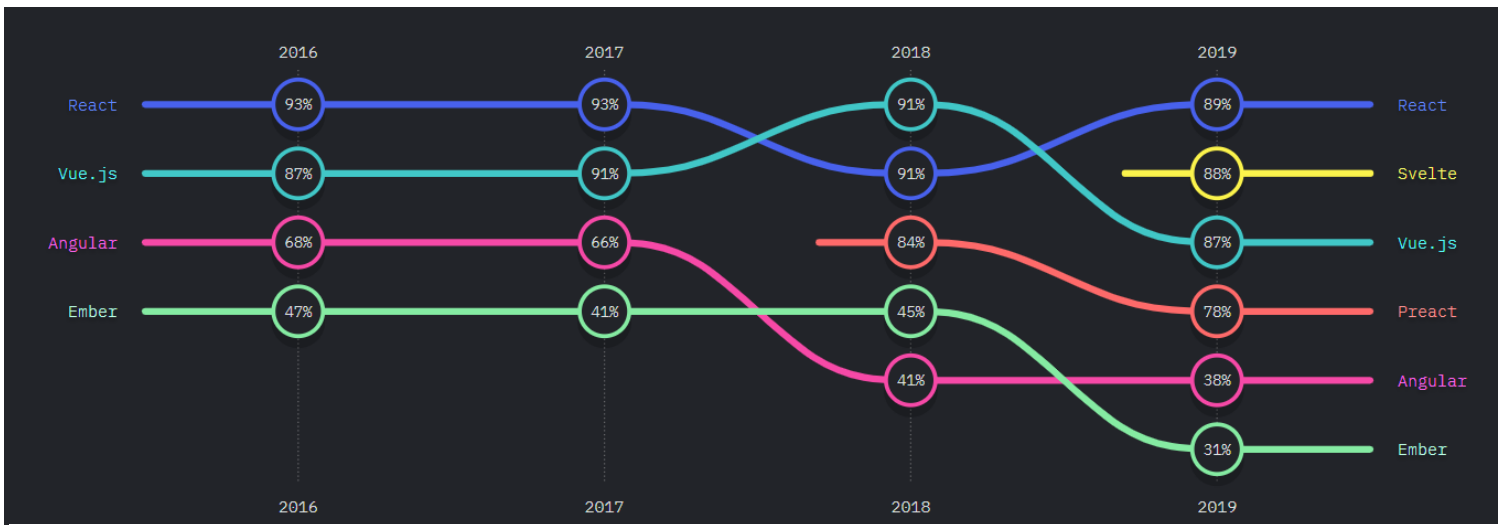


Fig. 15 Conocimiento, interés y satisfacción de los usuarios a lo largo de los años. (Greif & Benitte, s. f.)

CAPÍTULO IV METODOLOGÍA

Este capítulo tiene dos objetivos, el primero es la definición de la metodología y las actividades necesarias para que la creación de la SPA, la base de datos y la conexión de dispositivos IoT pueden ser completada. El segundo objetivo es definir el enfoque de la investigación que se llevará a cabo para responder las preguntas planteadas en esta tesis.

4.1 Enfoque y método

Las preguntas planteadas en esta investigación como "¿Cuántos operadores serán necesarios para su funcionamiento?" son preguntas que se concentran en mediciones numéricas las cuales son tratadas desde el enfoque cuantitativo según (Otero Ortega, s. f.).

La elaboración de las partes del proyecto se hará por medio de la metodología secuencial que es descrita por González, (s. f.) "La idea básica es desarrollar el sistema siguiendo etapas incrementales caracterizadas por generación de sucesivas versiones que van abarcando requerimientos hasta completar el sistema." (p.5) por esta razón se dividirá el proyecto en 5 incrementos y para una se evaluarán sus requisitos, se llevará a cabo la codificación del software necesario, se harán las pruebas para comprobar el correcto funcionamiento, para luego ser implementadas una a una de manera aditiva.

4.2 Variables de investigación

Las variables son fundamentales en la realización de una investigación y de acuerdo a lo escrito por (Villasís-Keever & Miranda-Novales, 2016) una variable se define como la parte de la investigación que constituye aquellos que se está midiendo, la información recolectada o los datos obtenidos usados con el fin de dar respuesta a las preguntas de la investigación. El concepto de variable se puede dividir en dependientes e independientes, siendo la variable dependiente afectada por una o más variables independientes.

Para el caso de este proyecto nos encontramos con 3 variables independientes y solo una dependiente, el cambio en el tiempo de respuesta al incrementar la cantidad de dispositivos, el cambio en el tiempo de respuesta al incrementar el tamaño del mensaje y la distancia que cubrirá la red local que depende de la cantidad de puntos de recepción colocados

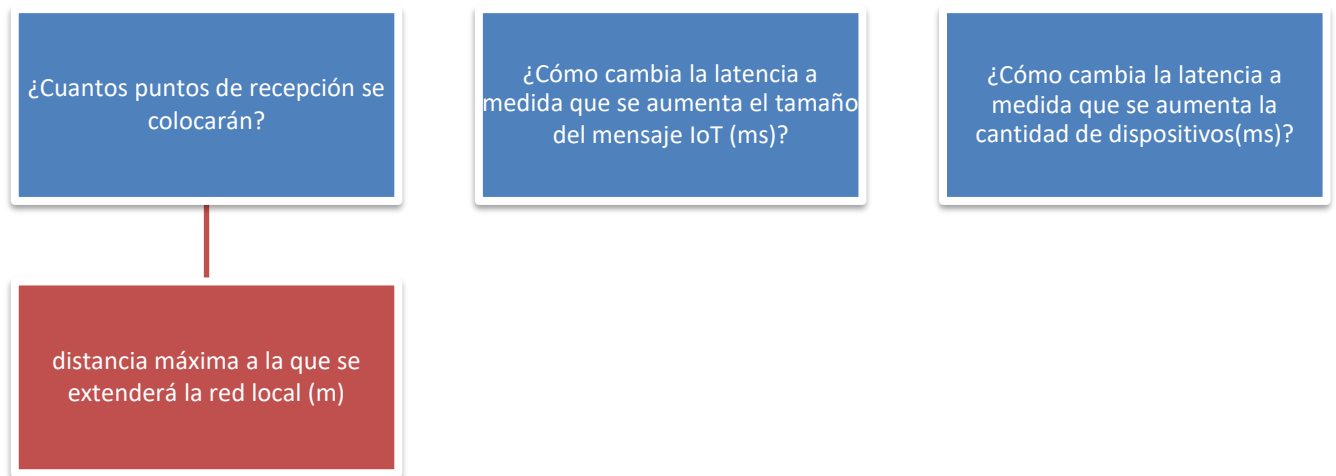


Fig. 16 Variables dependientes e independientes de la investigación (fuente propia)

4.3 Tecnologías aplicadas

Basándome en la investigación realizada en el marco teórico decidí utilizar las siguientes tecnologías:

- Un framework front-end para crear la SPA y el diseño del lado del cliente
- Un framework back-end para la administración de la base de datos y la lógica del lado del servidor
- Un protocolo de comunicación IoT
- Laragon, como servidor HTTP.
- PhpMyAdmin como administrado MySQL.
- Visual Estudio Code, como editor de código.

4.4 Materiales

La lista de materiales físicos necesarios para la realización de este proyecto es:

- **Microcontrolador**, con el cual se modificará la báscula digital para adquirir sus datos para luego enviarlos por medio del protocolo de comunicación IoT.
- **Báscula digital**, a la cual se le modificará su unidad de procesamiento para extraer su información y ser enviada usando el microcontrolador que mejor se ajuste a las necesidades

del proyecto

- **Smartphones**, con los que se comprobará que la SPA sea fácil de utilizar en un entorno móvil
- **Computadoras de escritorio**, con las que se comprobará que la SPA se fácil de utilizar en un entorno de escritorio
- **Servidor central**, Computadora en la que se ejecutarán la base de datos, el protocolo HTTP, y el protocolo de comunicación IoT elegido

4.5 Metodología de estudio

De acuerdo con lo mencionado en la sección 4.1 se utilizará la metodología Incremental, por lo que proyecto se dividirá en 5 incrementos en el que cada uno representa un producto terminado y a medida que estos incrementos se van uniendo aumentan la funcionalidad total de proyecto.

4.5.1 Elaboración de la tabla usuarios juntos a los formularios de registro

Análisis: Se iniciará con la tabla de usuarios dentro de la base de datos y la creación del formulario de registro para poder tener la estructura base en la cual se trabajará

Diseño: se crearán la tabla usuarios siguiendo este esquema:

La tabla usuarios conformada por las columnas y sus validaciones:

- **Id:** clave primaria, tipo Int, único, incremental, asignada por el sistema.
- **Nombre:** mínimo 8 caracteres, máximo 50 caracteres, tipo String, requerido.
- **Cédula:** máscara #####-####-#####, único, requerido.
- **Contraseña:** encriptada, mínimo 6 caracteres, requerido.
- **Fecha de creación:** tipo fecha, requerido.
- **Fecha de actualización:** tipo fecha, requerido.

La creación del formularios de registro es parte fundamental de este incremento, dicho formulario se creará en un componente separado para respetar la modularidad del proyecto.

Implementación: al ser esta la primera etapa, no tiene interacción con ninguna otra, por lo tanto su implementación es simplemente su creación.

Pruebas: rellenar la tabla usuarios con 100 registros, y comprobar que las validaciones se estén

ejecutando tanto en el lado del cliente, en el lado del servidor y dentro de la base de datos.

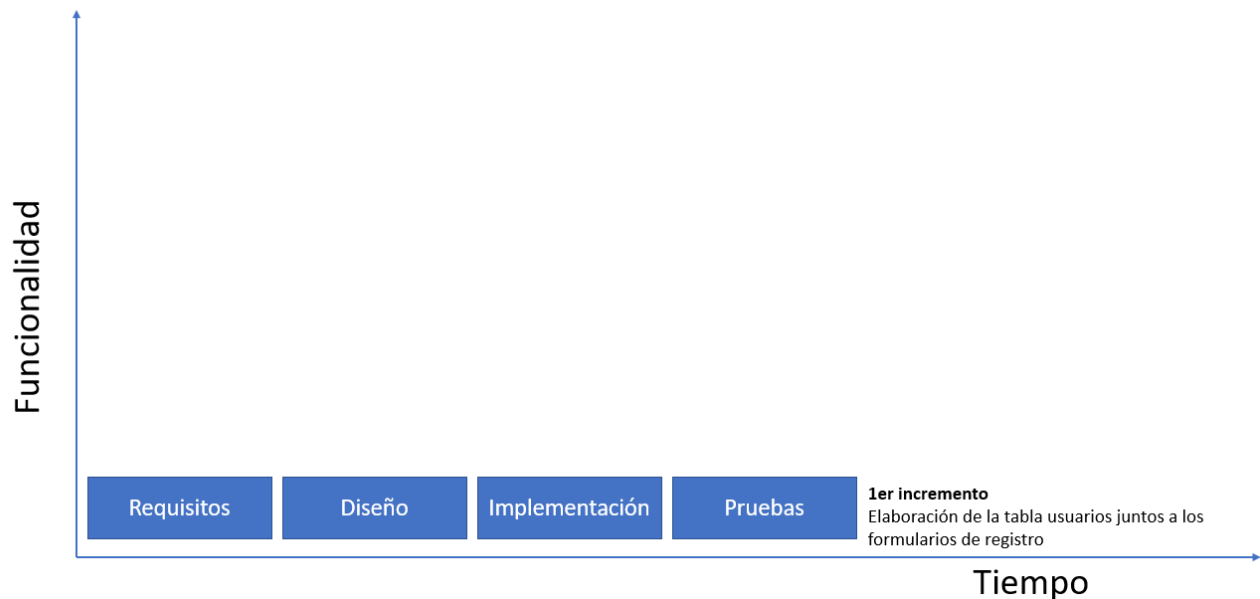


Fig. 17 Primer incremento (fuente propia)

4.5.2 Implementación del módulo de autenticación, permisos y la creación del formulario de login y permisos

Requisitos: se debe de crear un formulario de permisos y brindarle a cada usuario los permisos que le corresponden. En la parte del back-end se deben de crear las rutas necesarias para que el login se realice usando el campo cédula y el campo contraseña el cual deberá ser encriptada y comparada con la tabla de usuarios. En cuanto al formulario de permisos, este funcionará como un panel de control desde el cual se podrán adjudicar o revocar los permisos de cada usuario

Diseño: los formularios de login y permisos serán diseñados en componentes separados, se deben de generar los permisos necesarios para la creación, visualización y anulación para las tablas usuarios e ingresoFacturas. En la parte del back-end se deben de crear las rutas necesarias para que el login se realice usando el campo cédula y el campo contraseña el cual deberá ser encriptada y comparada con la tabla de usuarios.

Implementación: Esta etapa es la que brinda la seguridad a la SPA, por lo tanto cuando esta esté finalizada los usuarios solo deberían de ser capaces de ejecutar las acciones que tienen permitidas. En cambio, un usuario que no está logeado o registrado debería de ser incapaz de ejecutar cualquier acción dentro de la SPA.

Pruebas: brindar a cada usuario permisos al azar y verificar que solo puedan realizar aquellas

acciones para las cuales están autorizados, también se debe de verificar que un usuario no logeado o no registrado debería de ser incapaz de realizar cualquier acción dentro de la SPA. Se debe de comprobar el correcto funcionamiento de los formularios en dispositivos móviles y de escritorio.

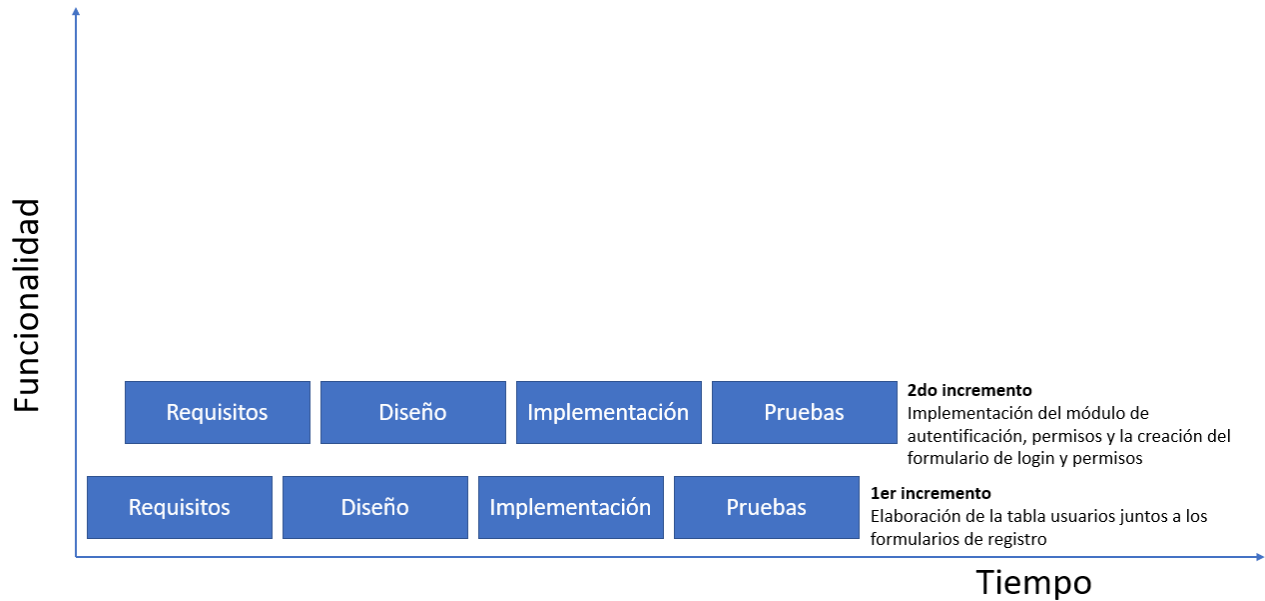


Fig. 18 Segundo incremento (fuente propia)

4.5.3 Creación la tabla ingresoFacturas y su formulario facturación de ingreso

Requisitos: Se creará la tabla ingresoFacturas que recibirá los datos de un formulario. Dicho formulario debe de ser fácil de entender y utilizar tanto en dispositivos móviles como de escritorio. Se deben de implementar validaciones de campo de lado del cliente. Se deben de crear las rutas y controladores necesarios para que la información del formulario sea correctamente guardada dentro de la tabla correspondiente.

Diseño: La tabla ingresoFacturas será creada siguiendo este esquema

La tabla ingresoFacturas:

- **Id:** clave primaria, tipo Int, único, incremental, asignada por el sistema, requerido.
- **IdPropietario:** clave foránea de tipo usuario, requerido.
- **IdCreador:** clave foránea de tipo usuario, requerido.
- **IdAnulador:** clave foránea de tipo usuario.
- **Fecha de creación:** tipo fecha, requerido.
- **Fecha de actualización:** tipo fecha, requerido.
- **Fecha de anulación:** tipo fecha.

- **PesoNeto:** tipo Int, requerido.
- **Sacos:** tipo Int, requerido.
- **TipoDeCafe:** tipo Enum (uva, mojado, seco, oreado), requerido.
- **Traído:** tipo bool, predeterminado falso, requerido.
- **Calidad:** tipo Enum (malo, normal, bueno), predeterminado normal, requerido.
- **Humedad:** tipo Int, rango 0-100, requerido.

Se realizarán las siguientes relaciones con la tabla usuarios:

- **Id(usuarios)->Id (ingresoFacturas):** uno a varios.
- **Id(usuarios)->IdCreador (ingresoFacturas):** uno a varios.
- **Id(usuarios)->IdPropietario(ingresoFacturas):** uno a varios.
- **Id(usuarios)->IdCreador(ingresoFacturas):** uno a varios.

Utilizando un componente separado para el formulario de ingresoFacturas este será dividido en dos secciones: primero, la selección de los usuarios, en el que se podrá identificar un usuario por cualquiera de los campos que lo definen y segundo, una vez seleccionado el usuario propietario se procederá a ingresar los datos restantes para la generación de un nuevo registro de ingresoFacturas.

Implementación: la primera sección del formulario de ingresoFacturas debé de poder obtener la información de todos los usuarios actualmente registrados, se deben de crear nuevos permisos para autorizar la creación de registros en la tabla ingresoFacturas.

Pruebas: comprobar que los permisos funcionen correctamente, comprobar que los registros se

estén guardando con la información introducida en el formulario dentro de la tabla ingresoFacturas.

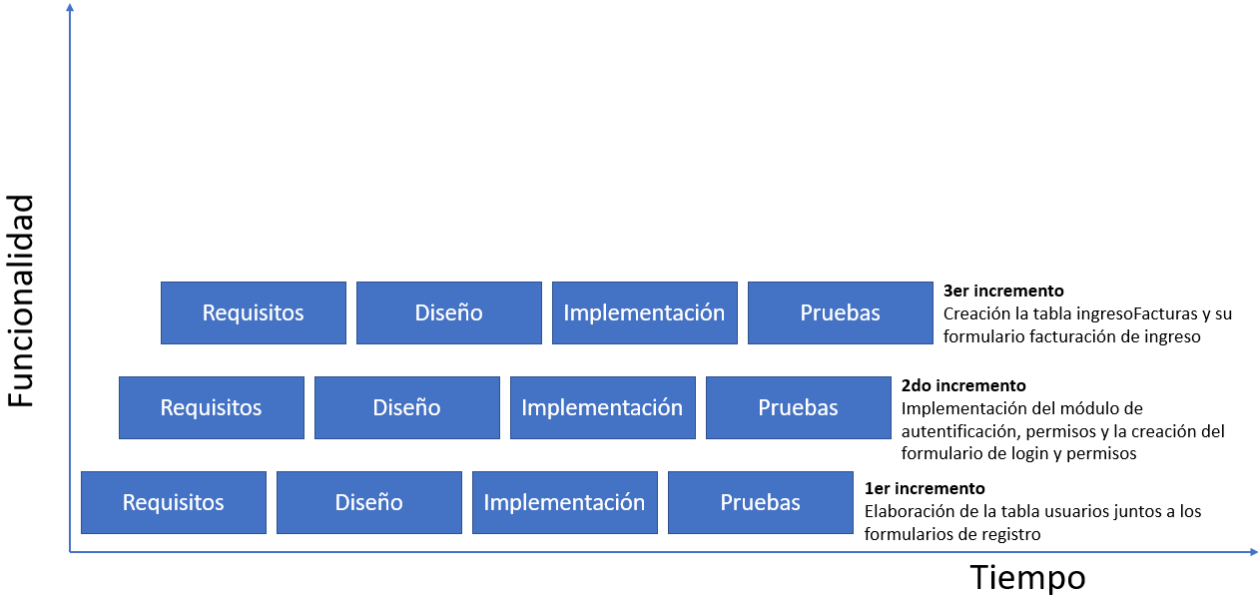


Fig. 19 Tercer incremento (fuente propia)

4.5.4 Implementación del protocolo seleccionado para la comunicación de los dispositivos IoT

Requisitos: crear una ventana emergente que permita seleccionar los dispositivos IoT que se encuentren la red, y que la información que estos recopilen sea mostrada correctamente dentro de la SPA.

Diseño: Utilizando JavaScript se suscribirá a cada usuario activo a un tópico que llevará su número de cédula, esto sirve para identificar a cada usuario en específico y que la información de los dispositivos IoT no esté disponible para todos los usuarios al mismo tiempo, solo para aquellos que se vinculen al dispositivo.

Implementación: en la realización de este proyecto el único dispositivo IoT que se desarrollará es la báscula digital modificada. Cada usuario tendrá la libertad de suscribirse o no al tópico en el cual el dispositivo está publicando su información, esto se hará por medio de un panel de control en el que se mostrarán los dispositivos IoT conectados a la SPA.

Pruebas: ya que Mosquitto tiene un panel de control en el que se pueden enviar mensajes bajo el tópico que se le especifique, se usará esa herramienta para comprobar que el usuario esté registrado bajo el tópico que tiene por nombre su número de cédula y que es capaz de recibir información

cuando esta se publica bajo ese tópico.

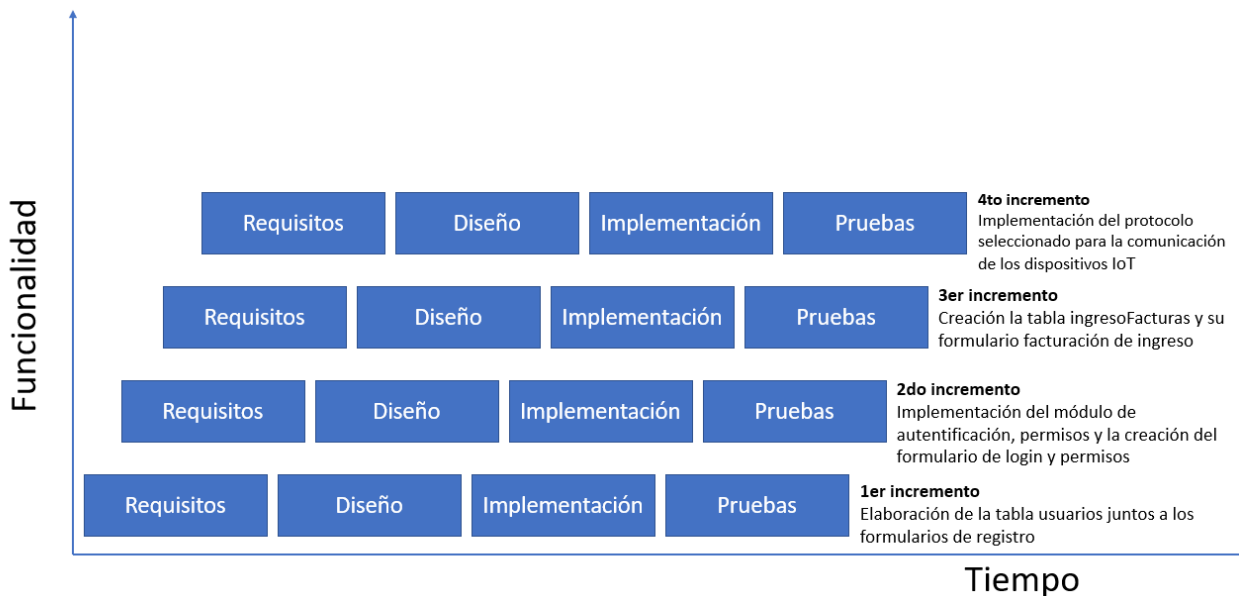


Fig. 20 Cuarto incremento (fuente propia)

4.5.5 desarrollo de la báscula digital IoT

Requisitos: Se modificará una báscula digital para poder extraer su información por medio de un microcontrolador esp32 o esp2866, este microcontrolador se conectará a la SPA por medio del protocolo de comunicación IoT seleccionado después de la investigación.

Diseño: Una vez la modificación sea realizada, el microcontrolador se registrará en el protocolo IoT con un nombre en específico, y publicará su información bajo ese nombre. Eso le permitirá a cualquier usuario con los permisos correspondientes suscribirse al mismo tópico y poder obtener la información

Implementación: ya que es una báscula digital su información será colocada automáticamente dentro del formulario de ingresoFacturas en el campo peso. En dicho campo se verá en tiempo real el resultado de la medición del peso del café que es entregado en el punto de recepción.

Pruebas: la parte fundamental de esta etapa es que la información que está midiendo la báscula digital sea la misma que es publicada en la SPA. Estas pruebas son fáciles de realizar pues no se está construyendo una báscula digital desde cero, sino que es una modificación. Dicha báscula será calibrada previamente y gracias a su display numérico se podrá comparar el dato obtenido por la

báscula con el dato que es enviado a la SPA.

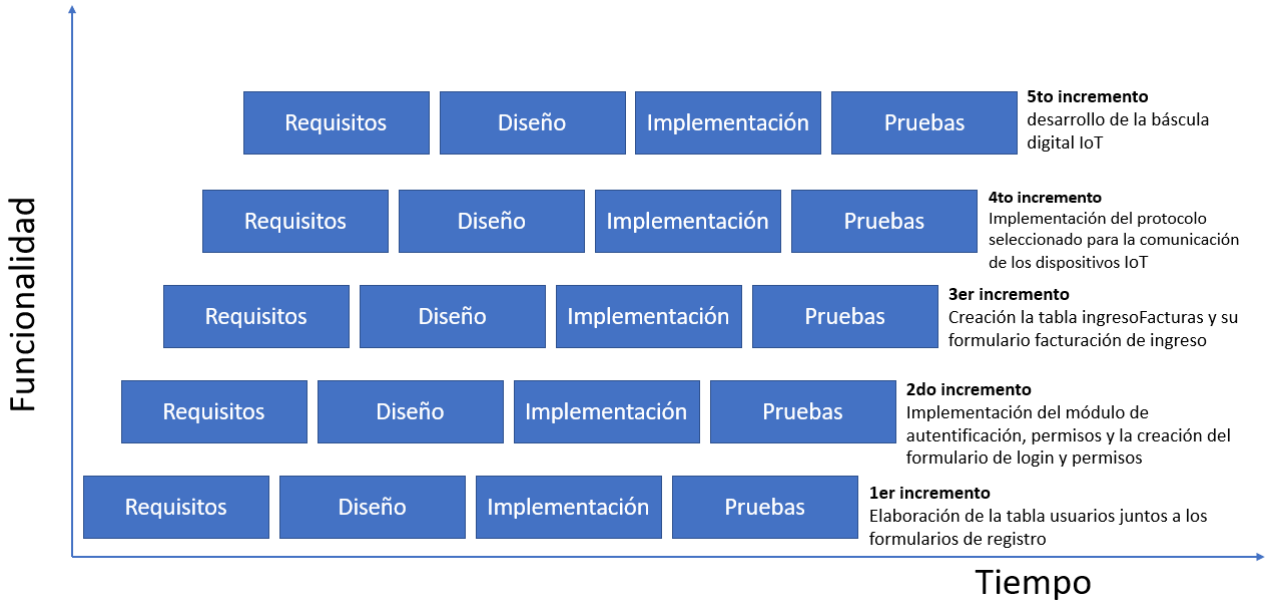


Fig. 21 Quinto incremento (fuente propia)

4.6 Metodología de validación

La investigación que se detalla en este documento será corroborada apoyándose en proyectos científicos, libros e investigaciones con enfoques relacionados al tema en cuestión. Por lo tanto, con los resultados de esta investigación también se pretende dar una nueva perspectiva a la hora de trabajar en café y como esto puede beneficiar a Honduras tanto durante la pandemia como fuera de esta.

4.7 Cronograma de actividades

Se elaboró un cronograma que especifica las fechas en las que se realizaran las actividades necesarias para completar este proyecto de investigación. Dicho proyecto está pensado para ser completa en 10 semanas, teniendo en cuenta los entregables y avances.

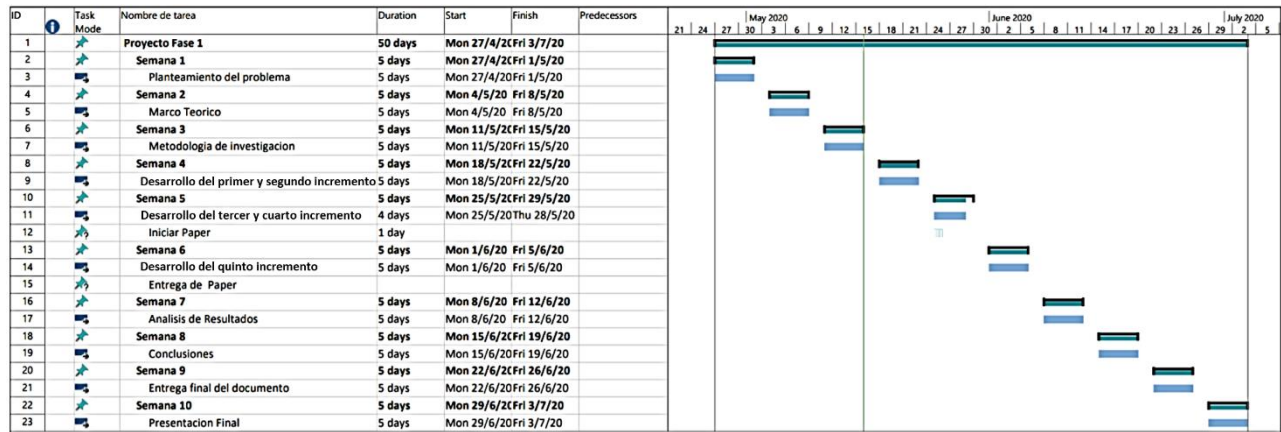


Fig. 22 Cronograma de actividades (fuente propia)

CAPÍTULO V. RESULTADOS Y ANÁLISIS

En este capítulo se muestra el resultado y análisis a través del desarrollo de la metodología del proyecto de investigación, donde se realizaron pruebas prácticas para comprobar el buen funcionamiento de las partes

5.1 Análisis de sustento del marco teórico

En el marco teórico se presentaron distintas tecnologías que serían capaz de llevar a cabo este proyecto. Para la realización de este proyecto es necesario elegir:

- Un protocolo de comunicación IoT
- Un framework front-end o una librería con capacidades de crear una SPA
- Un framework back-end

Los protocolos de comunicación IoT se representan en la siguiente tabla y sus características:

Características	MQTT	Zigbee	AMQP	BLE
Amplio soporte de la comunidad	X	X	-	X
Comunicación directa con Internet	X	-	X	
Comunicación en malla	-	X	-	X
Librerías JavaScript compatibles	X	X	-	-

Fig. 23 Tabla comparativa de protocolos IoT (fuente propia)

A pesar de que Zigbee presenta muchas de las características deseadas, para poder sacar el máximo potencial de esta tecnología es necesario utilizar más de un microcontrolador, uno que funcione como nodo y otro que funcione como coordinador; por otro lado MQTT no presenta ese problema y tiene las demás ventajas que presenta Zigbee. Por esto el protocolo de comunicación seleccionado es MQTT. En adición a esto se decidió utilizar el microcontrolador Node MCU ESP8266 como lo hizo (Kodali & Gorantla, 2017) quienes optaron por el mismo microcontrolador por su compatibilidad con el protocolo MQTT y su bajo consumo de energía, entre otros.

Los frameworks front-end y librerías capaces de crear una SPA son las siguientes:

Características	Vue.js	React.js	Angular
Amplio soporte de la comunidad	X	X	X
Enfocado en la creación de componentes	X	X	X
Actualización constante	X	X	X

Interés de la comunidad en la actualidad	alta	alta	baja
Curva de aprendizaje	baja	media	media

Fig. 24 Tabla comparativa de frameworks front-end (fuente propia)

La elección de un framework front-end es un poco complicada pues no está claro cuál de todas las tecnologías presenta mayores beneficios, pero es claro que React.js y Vue.js son las que presentan las mejores puntuaciones de los usuarios, ya que la satisfacción y el deseo de aprender a utilizarlos es extremadamente alta y se ha mantenido por muchos años. Sin embargo Vue.js presenta una documentación más fácil de entender a lo que se le suma lo dicho por (Silva & Alejandra, 2019) quien recomienda el uso de Vue.js por que mejora la productividad en el desarrollo de proyectos web, es por eso por lo que ha sido elegida por encima de React.js.

En cuanto a la selección del framework back-end, la decisión fue mucho más sencilla pues las características de Laravel son exactamente las necesarias para la creación de este proyecto. Esta decisión también fue tomada por (Yadav et al., 2019) quienes desarrollaron una web de comercio digital usando Laravel por gran ecosistema, su buena documentación y sus robustos paquetes de encriptación. Esto no quiere decir que Node.js no pueda ser utilizado para la realización de proyectos similares pues muchas de las características de las cuales carece puede ser agregadas utilizando paquetes de terceros.

5.2 construcción de la SPA

Se decidió por construir una SPA por la manera en la que enriquecen la interacción del usuario como lo indica (Deshmukh et al., 2019) quienes remarcan la facilidad de uso que reportan los usuarios respecto a las páginas web tradicionales.

El diseño back-end provisto por Laravel es bastante robusto. Laravel exige que se respete el diseño de programación llamado MVC (modelo vista controlador) las ventajas de este diseño de programación es que separa las funcionalidades de una página web para hacer el trabajo más eficaz, poder trabajar en grupos grandes y tener nuestro código ordenado. La manera en la que se procesa la información es la siguiente: El usuario hace una petición al controlador y este ejecuta acciones sobre el modelo luego el resultado de dichas acciones es renderizado en la vista la cual se encarga de presentar el resultado al usuario; una vez completado el ciclo el usuario dispondrá de nueva información con la cual podrá volver a llamar al controlador y así sucesivamente. Para el caso de la SPA la diferencia se encuentra en la que el renderizado de la página no requiere que se envíe una

página HTML completamente nueva, solo la información que ha sido modificada es refrescada cada vez que se ejecuta uno de estos ciclos.

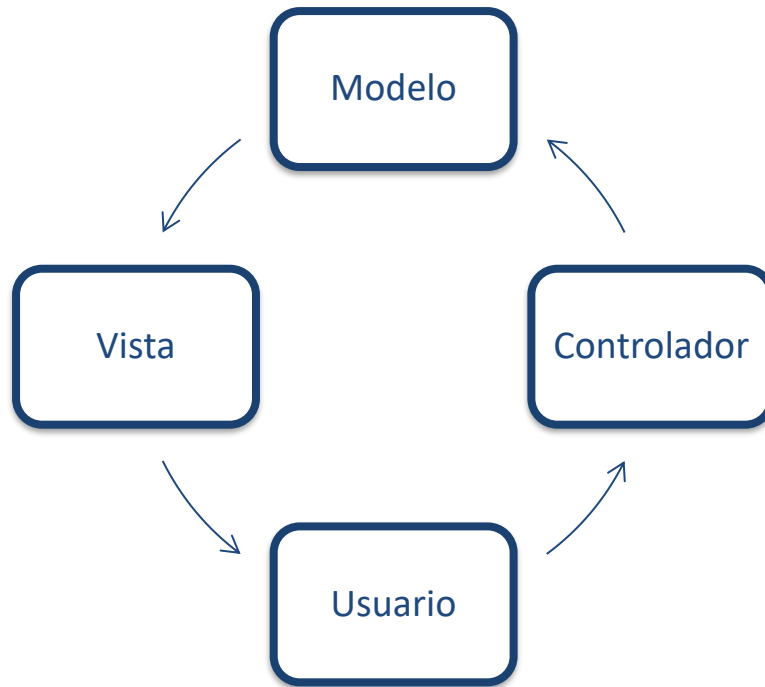


Fig. 25 Representación visual del esquema MVC (fuente propia)

Una vez conociendo la manera en la que se mueve la información, podemos definir cada una de las partes por separado.

- **Modelo:** es el elemento central y es una estructura de datos dinámica, en la cual se maneja la lógica los datos y las reglas de la aplicación. Laravel permite que en sus modelos se definan funciones específicas que hagan operaciones con los datos contenidos dentro del registro al que se accede.
- **Vista:** cualquier representación de la información
- **Controlador:** es quien recibe las peticiones del usuario y las convierte en comandos para el modelo o la vista. Laravel permite que un controlador pueda acceder a varios modelos al mismo tiempo dentro de una misma función.

La creación de estas 3 partes es fundamental para cada tabla, pues Laravel trata cada registro de las tablas creadas como una instancia de su modelo asociado y junto a su ORM Eloquent es que podemos interactuar con dichos registros de manera fácil y segura pues incluye muchas de las funciones comúnmente utilizadas a la hora de ver, modificar, crear o eliminar registros.

La presentación de las vistas se comúnmente por medio de rutas, sin embargo en el caso de una SPA la vista es única y las rutas no son utilizadas en la manera en la que Laravel las plantea en su documentación oficial. Ya que estamos utilizando Vue.js es por medio de esta librería que se renderizan las vistas de acuerdo a la ruta entregada en la barra de dirección de nuestro navegador, hay que aclarar que esto no significa que se estén entregando distintas vistas con cada dirección, sin importar cual sea la dirección a la cual nos queremos dirigir Laravel siempre responde con la misma página HTML que contienen todo lo necesario para funcionar, el único requisito es que la dirección empiece por "www.nucelczc.test/ " cualquier cosa escrita a continuación de la pleca será procesada por Vue.js para decidir qué componentes se renderizaran en el DOM. Pero es importante notar que Vue.js solo se encarga de la renderización de las vistas por lo que las rutas de Laravel siguen siendo utilizadas para enviar información al controlador, y teniendo en mente la meta de generar una SPA capaz de ser fácilmente escalable se automatizó la creación de rutas de esta manera.

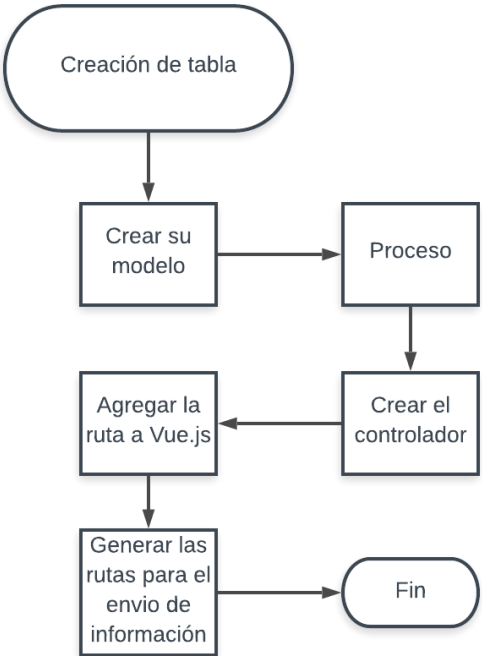


Fig. 26 Procedimiento para agregar rutas (fuente propia)

Siguiendo ese diagrama se asegura que, para cada tabla creada, se cree su modelo, su vista, su controlador con sus funciones y las rutas utilizadas para enviar información para ser procesada. La información se envía al controlador por medio de instancias *Request* que son aceptadas por

Laravel y se construyen el front-end por medio de peticiones asíncronas de Ajax, esto es lo más recomendado pues permite que información sensible sea enviada sin que está sea expuesta en la barra de dirección del navegador sin embargo esto no significa que información extremadamente sensible como las contraseñas no deban ser encriptadas antes de ser enviadas.

El diseño front-end usa componentes prefabricados por Vuetfy.js, dichos componentes agregan muchas funcionalidades sin necesidad de tener que ser programadas por cuenta propia y cuentan con un estilo muy consistente entre cada componente, su documentación oficial es muy detallada, fácil de entender y se mantiene en constante actualización. En adición a los componentes provistos por Vuetify.js muchos otros componentes fueron creados de manera personalizada para poder que el funcionamiento de la página sea el óptimo.

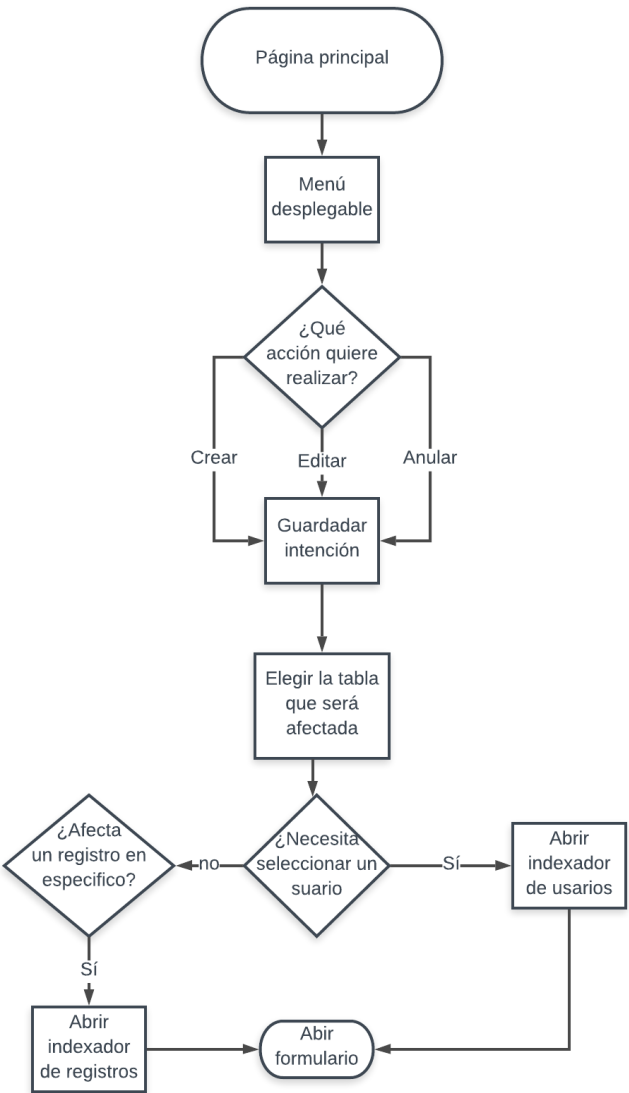


Fig. 27 Navegación por la SPA (fuente propia)

El diagrama expuesto arriba demuestra la navegación básica dentro de la SPA, en él se pueden ver 3 decisiones que son importantes a la hora de abrir el formulario, puestas estas decisiones son guardadas para posteriormente ser enviadas por medio de Ajax al controlador antes de abrir el formulario y cambia la manera en que el formulario es renderizado.

5.3 Desarrollo de la etapa 1: Elaboración de la tabla usuarios juntos a los formularios de registro

La tabla de usuarios es la tabla principal necesaria para el funcionamiento del proyecto y el formulario de registro será el encargado de agregar nuevos registros a dicha tabla.

5.3.1 Creación de las migraciones

En Laravel, la manera de crear una tabla dentro de una base de datos es por medio de "migraciones". Estas piezas de código están pensadas para mantener un registro de como las tablas se van modificando a lo largo del desarrollo del proyecto. Esta migración fue creada utilizando la plantilla predeterminada de Laravel, ya que es lo recomendado según la documentación oficial.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1 id	int(10)		UNSIGNED	No	None		AUTO_INCREMENT
<input type="checkbox"/>	2 nombre	varchar(255)	utf8mb4_unicode_ci		No	None		
<input type="checkbox"/>	3 cedula	varchar(255)	utf8mb4_unicode_ci		No	None		
<input type="checkbox"/>	4 contraseña	varchar(255)	utf8mb4_unicode_ci		No	NULL		
<input type="checkbox"/>	5 created_at	timestamp			Yes	NULL		
<input type="checkbox"/>	6 updated_at	timestamp			Yes	NULL		

Fig. 28 Tabla usuarios (fuente propia)

5.3.2 Creación del formulario de registro

En un componente separado, se creó el formulario de registro. Dicho formulario permite que se agreguen registros de manera rápida, está validado tanto del lado del cliente como del lado del servidor.



Fig. 29 Formulario de registro (fuente propia)

5.3.3 Prueba de funcionamiento

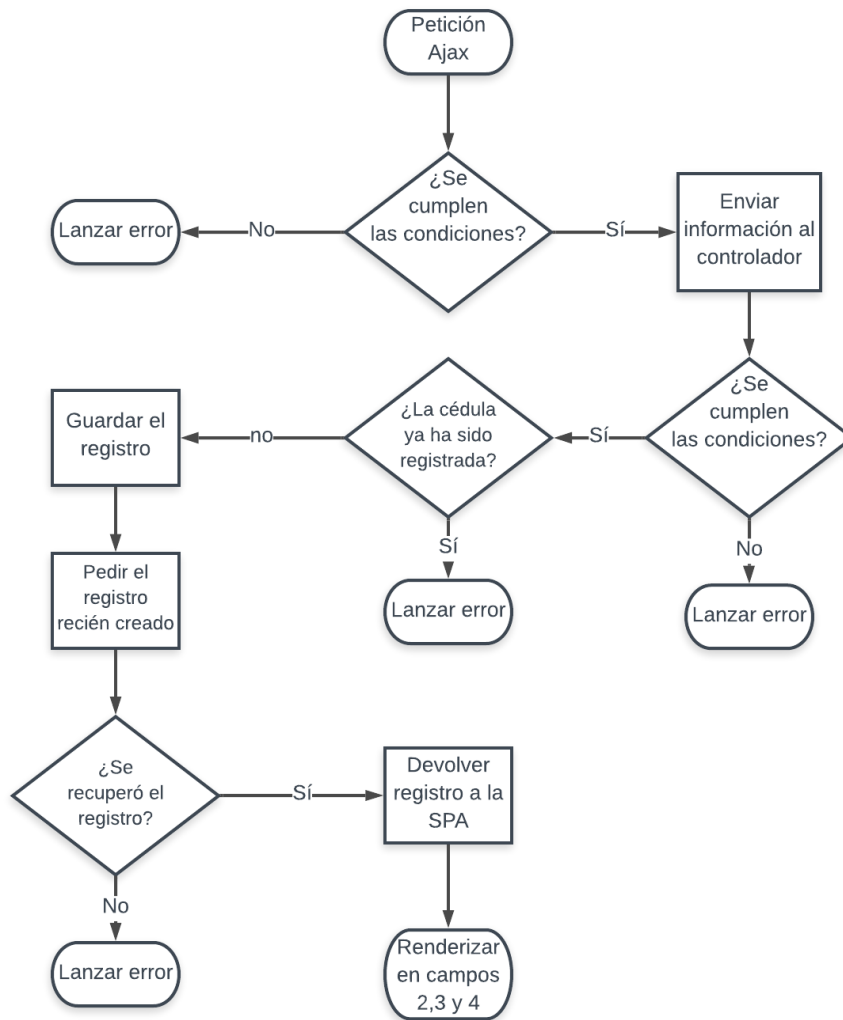


Fig. 30 Funcionamiento del formulario de registro (fuente propia)

Se comprobó que el formulario de registro funcione correctamente y que fuera seguro tanto del lado del cliente como del lado del servidor.

Utilizando los Seeders de Laravel, se rellenó la tabla de usuarios con 100 registros generados

aleatoriamente y el registro de administrador.

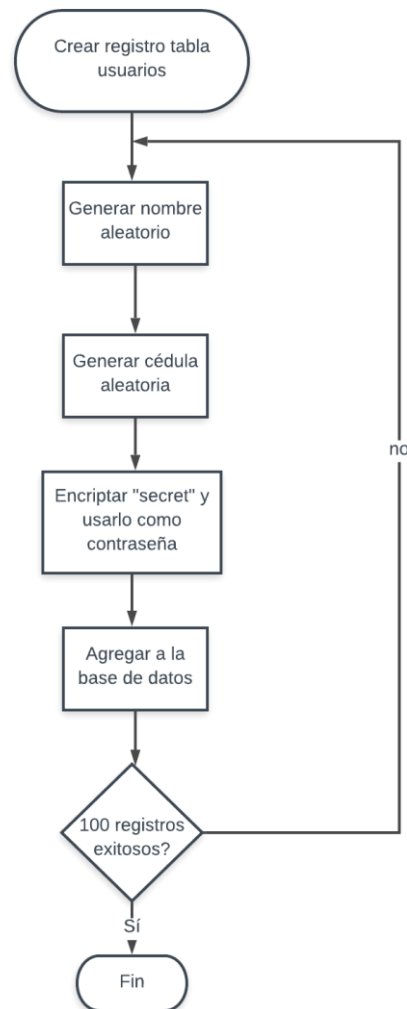


Fig. 31 Seeder de la tabla usuarios (fuente propia)

Los 100 registros fueron creados exitosamente.

5.4 Desarrollo de la etapa 2: Implementación del módulo de autenticación, permisos y la creación del formulario de login y permisos

El módulo de autenticación y el módulo de permisos son los que brindan la seguridad a la SPA, la autenticación es la acción de iniciar sesión dentro de la SPA y los permisos son los que controlan las acciones que un usuario puede o no hacer dentro de la SPA.

5.4.1 Creación de las migraciones

Los permisos son guardados en la tabla llamada "permissions" y tienen una relación varios a varios con la tabla "usuarios". En estos casos la documentación de Laravel indica que debe de generarse

una tercera tabla que sirva como puente para que la relación varios a varios sea llevada a cabo exitosamente.

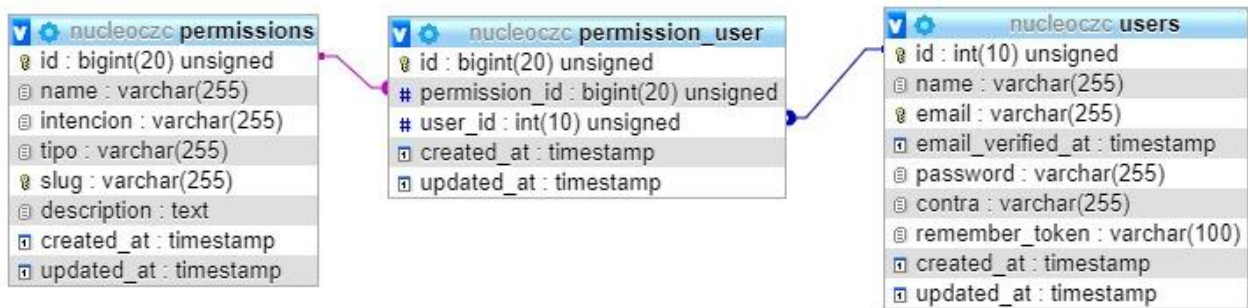


Fig. 32 relación varios a varios entre la tabla permissions y la tabla usuarios

5.4.2 Creación del formulario de permisos y login

En un componentes separados, se creó el formulario de permisos y el de login. El formulario de Este formulario tiene la particularidad de que al abrirse presenta los permisos actualmente para lograr esto, al seleccionar el usuario se hace una petición Ajax para conseguir los permisos de dicho usuario para poder renderizarlos con anticipación, luego cualquier cambio realizado será guardado automáticamente.

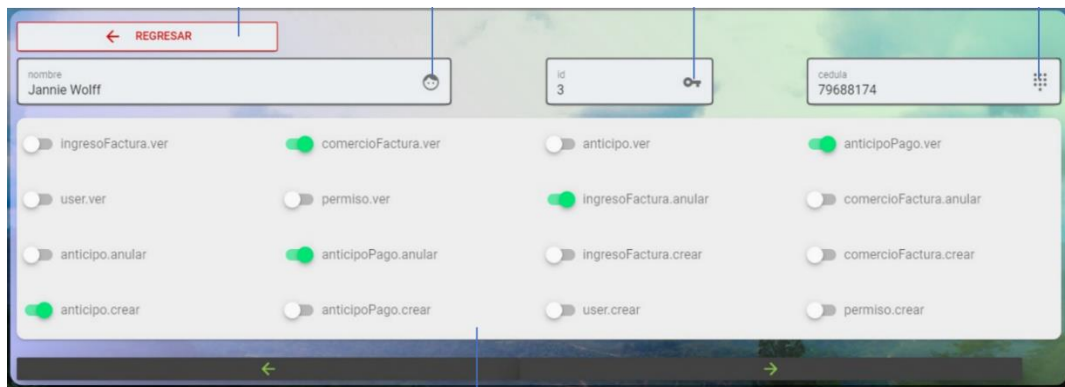


Fig. 33 formulario de permisos (fuente propia)

El formulario de login utiliza la cedula y la contraseña para que el usuario inicie sesión dentro de la SPA, es enviada al controlador de login donde es encriptada y se compara con la contraseña, también encriptada, que coincide con la cédula introducida en el formulario. Luego de que se comprueba la coincidencia tanto de la cédula como de la contraseña, Laravel se encarga de iniciar la sesión de ese usuario y la mantiene iniciada siempre y cuando el navegador no sea cerrado. Laravel no tiene ningún problema manejando múltiples sesiones iniciadas al mismo tiempo.

The image shows a login form with two input fields: 'Cédula' containing the number '1622199800209' and 'Contraseña' with a masked password '.....'. Below the fields is a green 'ENTRAR' button.

Fig. 34 formulario de login (fuente propia)

5.4.3 Prueba de funcionamiento

Ya que en el desarrollo de la etapa anterior se crearon 100 usuarios aleatorios y a todos se les dio la misma contraseña, se seleccionaron 10 usuarios aleatorios y se comprobó que el sistema de permisos funcionara de manera correcta y que los permisos se actualizarán correctamente dentro del panel de control de formulario de permisos. Esto se logró utilizando los Seeders de Laravel.

Una prueba similar se llevó a cabo para comprobar el funcionamiento del sistema de login, intentar realizar cualquier acción dentro de la SPA sin estar logeado resulta en un rechazo por parte del servidor.

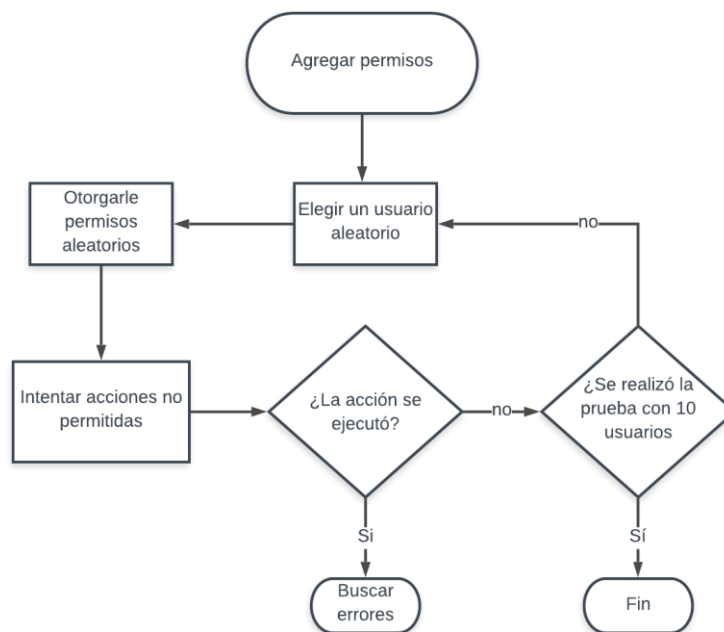


Fig. 35 Seeder de la tabla permissions (fuente propia)

5.5 Desarrollo de la etapa 3: Creación la tabla ingresoFacturas y su formulario facturación de

ingreso

La tabla de ingresoFacturas es donde se guarda la información de cada factura de ingreso realizada. Las facturas de ingreso sirven para llevar el registro de las características del café recolectado por cada productor, luego ellos pueden utilizar dichas facturas para decidir cuándo vender el café recolectado. Tradicionalmente dichas facturas siempre se han realizado a mano por lo tanto el formulario de facturación debe de ser fácil de entender tanto como para el que crea la factura como para el que la recibe.

5.5.1 Creación de las migraciones

La tabla de ingresoFacturas tiene menos columnas que las que tiene el talonario de facturación comúnmente utilizado, se hizo así porque muchas de las columnas son el resultado de operaciones matemáticas realizadas usando otras columnas por lo tanto se vuelven redundantes.

Es importante notar que se incluye dos campos relacionados con la anulación de la factura, ya que por deseos de la empresa ningún registro debe de ser borrado de la base de datos incluso si el registro fue creado con información errónea, por eso, en caso de que una factura sea creada con información errónea esta solo es anulada en lugar de eliminada de la base de datos.





#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1 id 	int(10)		UNSIGNED	No	None		AUTO_INCREMENT
<input type="checkbox"/>	2 idcreador 	int(10)		UNSIGNED	No	None		
<input type="checkbox"/>	3 idanulador 	int(10)		UNSIGNED	Yes	NULL		
<input type="checkbox"/>	4 idpropietario 	int(10)		UNSIGNED	No	None		
<input type="checkbox"/>	5 pesoNeto	double			No	None		
<input type="checkbox"/>	6 sacos	double			No	None		
<input type="checkbox"/>	7 tipo	varchar(255)	utf8mb4_unicode_ci		No	None		
<input type="checkbox"/>	8 traído	tinyint(1)			No	None		
<input type="checkbox"/>	9 calidad	varchar(255)	utf8mb4_unicode_ci		No	None		
<input type="checkbox"/>	10 humedad	double			No	None		
<input type="checkbox"/>	11 fechaAnulado	datetime			Yes	NULL		
<input type="checkbox"/>	12 created_at	timestamp			Yes	NULL		
<input type="checkbox"/>	13 updated_at	timestamp			Yes	NULL		

Fig. 36 tabla ingresoFacturas (fuente propia)

Estas facturas tienen tres relaciones varios a uno con la tabla de usuarios, ya que un mismo usuario puede ser dueño de múltiples facturas, un mismo usuario puede ser el creador de múltiples facturas y un mismo usuario puede ser el anulador de múltiples facturas

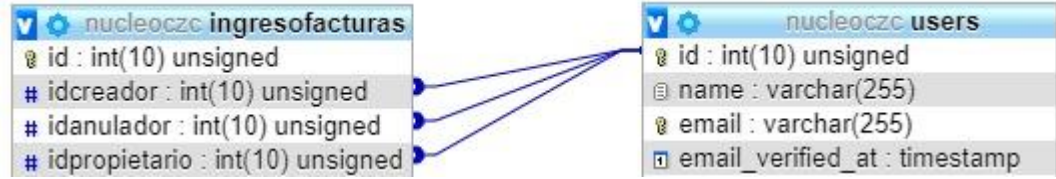


Fig. 37 tres relaciones uno a varios entre la tabla ingresoFacturas y la tabla usuarios (fuente propia)

5.5.2 Creación del formulario de facturación

Como se mencionó anteriormente, el formulario de facturación debe ser fácil de utilizar y estar validado en el lado del cliente, en el lado del servidor y en la base de datos. Para lograr que la interfaz fuera fácil de usar en dispositivos móviles el formulario fue diseñado para que encaje en una sola pantalla.

The screenshot shows a mobile application interface for an invoice summary. It features a dark grey background with white text. At the top, there is a blue header with the text 'Toque para guardar:'. Below this, the summary is presented in a list of items, each with a bold label and a value: 'Propietario: Mortimer Kub', 'Peso Total: 123', 'Tara: 10', 'Tipo: uva', 'Calidad: bueno', 'Humedad: 23', 'Total Neto: 113', and 'Total seco: 87.01'.

Fig. 38 resumen del formulario ingresoFacturas (fuente propia)

- **Peso Total:** es la suma de las pesadas que realice la báscula digital
- **Tara:** es el peso que se le asocia a un saco vacío, en caso de que el saco esté húmedo el peso es considerado como una libra y en caso de que el saco se encuentre seco, solo se le atribuye un peso de 0.5 libras
- **Tipo:** detalla el estado en el que es entregado el café, este puede ser entregado en Uva, Verde, Seco y húmedo
- **Calidad:** es un calificativo de que tan bueno se encuentra el café, este valor es importante a la hora de realizar el pago de la factura
- **Humedad:** el café se paga de acuerdo con su peso cuando se encuentra seco al 13% de humedad, cualquier porcentaje de humedad arriba del 13% es considerado como peso extra añadido por el agua. En cuanto al peso añadido del café en uva se considera que solo el 20% del peso en uva será su peso cuando se encuentre seco al 13%
- **Total Neto:** es el resultado de restar el peso total menos la tara, este número no es guardado en la tabla de ingresoFacturas pero es presentado al creador y al propietario para corroborar los números
- **Total Seco:** es el resultado de restar todo el peso extra atribuida a la humedad arriba del 13% del café, este número tampoco es guardado dentro de la tabla ingresoFacturas solo es para que el creador y el propietario puedan corroborar los números

Las operaciones necesarias para realizar los cálculos del total neto y el total seco están escritas dentro del modelo de ingresoFactura esto es de utilidad porque este formulario es el que más cambia de acuerdo con la intención con la que es abierto por lo tanto es importante explicar cómo se comporta el formulario de acuerdo con las distintas intenciones

5.5.2.1 Crear

Para crear un registro de ingresoFactura se tiene que elegir un usuario por medio del indexador de usuarios, luego rellenar los espacios necesarios y por último validar la creación del registro ingresando la contraseña de login una vez más.

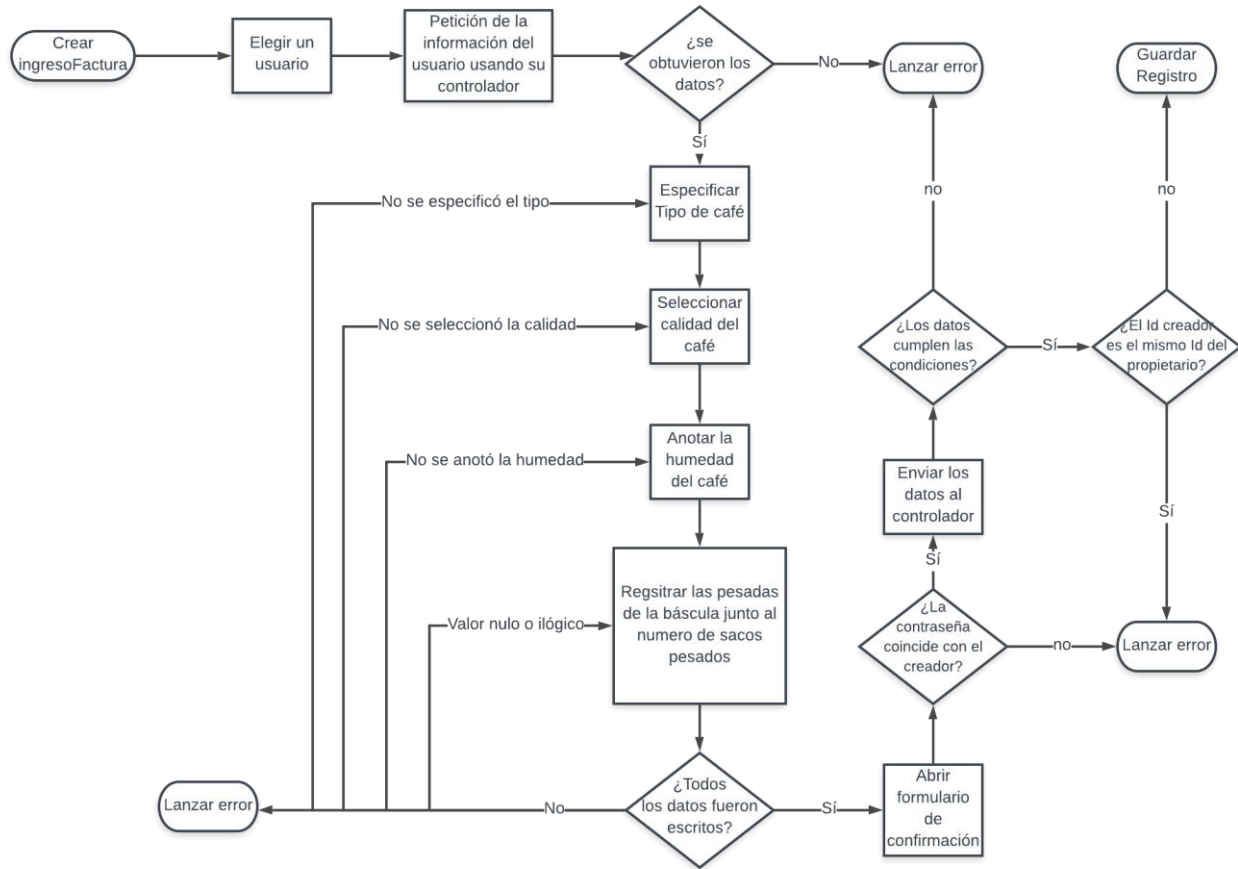


Fig. 39 funcionamiento del formulario ingresoFacturas con la intención "crear" (fuente propia)

5.5.2.2 Ver

La acción de ver es la más simple de todas pues solo presenta un resumen de los datos de la ingresoFactura a la cual queremos acceder. Para elegir la ingresoFactura entre las miles que se generan en un año común es necesario filtrar de acuerdo con los parámetros conocidos, los métodos más directos son: Buscar entre las ingresoFacturas de un usuario en particular o ingresar el id de una ingresoFactura conocida

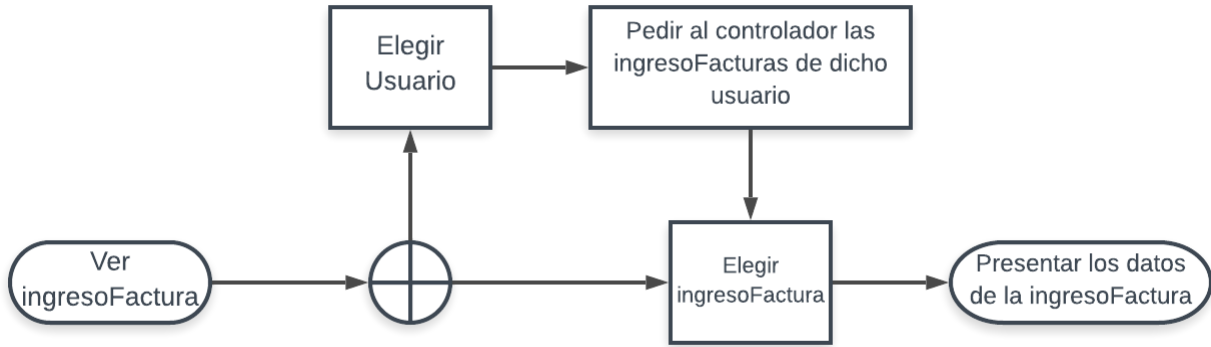


Fig. 40 funcionamiento del formulario ingresoFacturas con la intención "ver" (fuente propia)

En cuanto a los permisos, este caso sería un muy especial pues cada persona es capaz de ver las ingresoFacturas de la cual es dueño, pero no puede ver las de los demás. Para hacer esto, el controlador de ingresoFacturas tiene una ruta extra que se salta el permiso de *ver.ingresoFacturas* siempre y cuando el usuario sea el propietario de dicha ingresoFactura. Un caso parecido se da con los creadores de dichas ingresoFacturas pues ellos deben de ser capaces de ver las ingresoFacturas que han creado por si quieren verificar la información introducida o en caso de un reclamo tener un respaldo, una vez más, esto significa crear una ruta en específico para saltar el permiso de *ver.ingresoFacturas*.

5.5.2.3 Anular

Esta acción es la que sustituye a la acción de eliminar, esto se ha decidido así por la manera en la que Beneficio Rio Frio ha manejado sus facturas durante mucho tiempo. Esta acción funciona casi de la misma manera que la acción ver, pero en este caso es necesario poder ir al formulario de contraseña para asegurar que la persona que anula dicha factura es consciente de que la está anulando

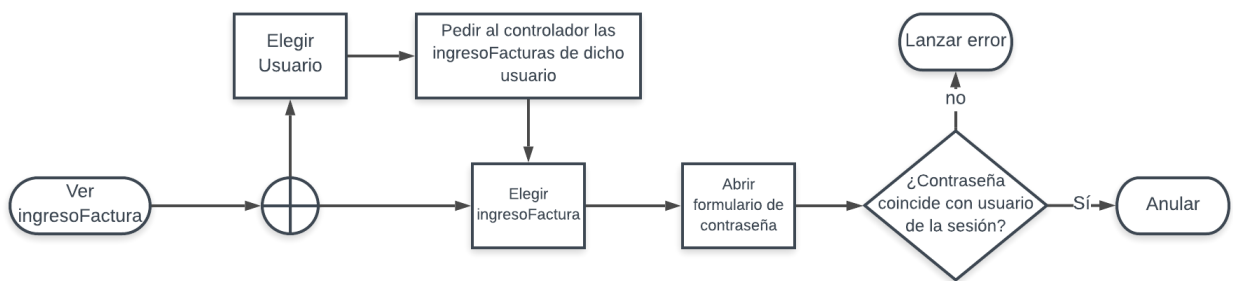


Fig. 41 funcionamiento del formulario ingresoFacturas con la intención "anular" (fuente propia)

En este caso la diferencia principal respecto al caso de la intención *ver*, es que el propietario de la *ingresoFactura* no puede anular su factura, solo el creador puede por medio de una ruta especial que se salta el permiso de *anular.ingresoFacturas*. Explicado esto, cualquier persona que disponga de ese permiso es capaz de anular cualquier factura sin importar si es el propietario, creador o ninguno de los dos. La información de quién fue la persona que anuló el registro así como la fecha y hora en la que se ejecutó la acción se guardan actualizando el estado del registro.

5.5.3 Prueba de funcionamiento

Se agregaron 1000 registros a la tabla de *ingresoFacturas* con los seeders de Laravel, esto se hizo tomando 2 usuarios aleatorios, en el que uno es elegido propietario y el otro como creador. Luego de realizar esta prueba se permitió a personas del beneficio interactuar con la página utilizando sus propios smartphones y todo funcionó correctamente. Luego de estas 2 pruebas se comprobó que los permisos relacionados a la creación de facturas funcionaran correctamente siguiendo el procedimiento utilizado en la etapa 2

5.6 Desarrollo de la etapa 4: Implementación del protocolo seleccionado para la comunicación de los dispositivos IoT

Como se mencionó al inicio de este capítulo se eligió MQTT como el protocolo a utilizar para la comunicación con la SPA. Esto se logró utilizando el broker Mosquitto el cual se encarga de brindar algunas configuraciones relacionadas a la seguridad. La seguridad que implementa Mosquitto se basa en pedir una contraseña para comunicarse dentro de la red, de esta manera un dispositivo que no disponga de dicha contraseña no podrá ser conectado. Otra de las razones por la cual se utilizó Mosquitto es porque dispone de compatibilidad con la librería MQTT.js para JavaScript, dicha librería contiene los métodos para suministrar la contraseña, identificar a cada cliente con un id único. Es oportuno mencionar que utilizar una SPA en ambientes industriales permite que los datos se desplieguen al usuario de manera rápida como lo menciona (Salgado et al., 2019) quienes crearon un prototipo SCADA para controlar el nivel de agua en contenedores utilizando una SPA.

5.6.1 Diseño de la ventana emergente

Como se explicó a inicios de este capítulo, la SPA cuenta con un menú desplegable desde el cual se puede elegir la acción que se quiere realizar con los registros. Utilizando dicho menú se agregó una nueva opción que permite la aparición de una ventana emergente desde la cual se inicia la sesión MQT. Esta ventana emergente puede ser abierta desde cualquier parte de la SPA pues se espera que

se agreguen dispositivos con finalidades distintas en el futuro.

Justo en el momento en el que la ventana es abierta, el cliente usuario es suscrito a la tópicos MQTT "pagina".

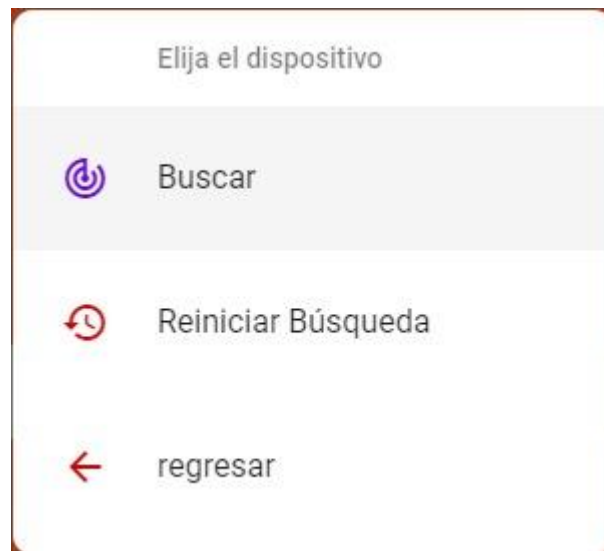


Fig. 42 ventana emergente para la conexión MQTT (fuente propia)

5.6.3 Pruebas

En esta fase se consideró óptimo responder las preguntas de investigación pues, todas están enfocadas a evaluar el funcionamiento de la comunicación IoT. Para hacerlo se construyó un componente por separado dedicado a medir el tiempo que requiere una comunicación de doble sentido en el protocolo MQTT, esta comunicación se da entre el microcontrolador esp8266 utilizado para crear la báscula IoT y uno o más usuarios de la SPA. En la investigación de (Sadeq et al., 2019) encontraron que el envío de grandes cantidades de datos por MQTT puede resultar en colisiones,

5.6.3.1 Cantidad de dispositivos

En este caso, un solo usuario de la SPA hace la pide al microcontrolador esp2866 la entrega de 20 mensajes de un 5 bytes a una cantidad variable de dispositivos, el tamaño fue decido basado en la entrega de un número menor a 1,099,511,627,776, esto es así porque la función más común que realizarán los dispositivos dentro de la red es la de enviar números pequeños.

- **tamaño del mensaje: 5bytes, usuarios: 1**

En esta primera prueba se pudo observar un comportamiento bastante uniforme, con un tiempo de respuesta considerablemente bueno para las aplicaciones que se tienen pensadas

- **tamaño del mensaje: 5bytes, usuarios: 2**

Esta prueba tiene la particularidad de tener un usuario conectado desde un dispositivo móvil y otro conectado desde una computadora de escritorio, esto tampoco resultó en ningún cambio significativo en el tiempo de respuesta.

- **tamaño del mensaje: 5bytes, usuarios: 3**

En esta prueba se conectaron 2 dispositivos de escritorio y un dispositivo móvil, una vez más la latencia fue mínima para todos los dispositivos. Viendo estos resultados se consideró que se debería proseguir conectando todos los dispositivos que se tenían cerca.

- **tamaño del mensaje: 5bytes, usuarios: 5**

A pesar de haber conectado todos los dispositivos que se encontraron dentro de las instalaciones, no se vio un incremento en el tiempo de respuesta, esto da un poco más de firmeza a dicho por (Tantitharanukul et al., 2017) quien remarca la mejora considerable que tiene MQTT respecto a HTTP a la hora de enviar mensajes a múltiples dispositivos sin consumir demasiados recursos. Por lo tanto, se puede concluir que la red es capaz de soportar suficientes dispositivos para los requerimientos de cualquier trabajo hecho en Beneficio Rio Frio en el futuro cercano.

5.6.3.2 Tamaño del mensaje

Este tipo de pruebas se enfocaron en enviar mensajes cada vez más grandes a un solo dispositivo, aunque la SPA no está pensada para compartir datos de gran tamaño por medio del protocolo MQTT, saber la cantidad máxima que puede soportar permitiría plantear la implementación de aplicaciones más complicadas

- **tamaño del mensaje: 5bytes, usuarios: 1**

Esta es idéntica a la primer prueba de la sección anterior pero se incluyó para dar un punto de referencia.

- **tamaño del mensaje: 50bytes, usuarios: 1**

Multiplicar por 10 el tamaño del mensaje enviado al parecer no tuvo ningún impacto en el tiempo de respuesta una vez más, sin embargo un incremento de este tamaño abre nuevas posibles aplicaciones pero no aplicaciones que sean radicalmente distintas, un ejemplo de esto es el proyecto realizado por (Alqinsi et al., 2018) en el que realizó un sistema de monitoreo para una fuente de alimentación interrumpible, el cual solo requería que se

enviaran mensajes de 15bytes

- **tamaño del mensaje: 100bytes, usuarios: 1**

En esta prueba a pesar de que una vez más no se vieron cambio significativos, se llegó al límite que el microcontrolador puede soportar aunque investigando en la documentación de la librería que permite la comunicación con el protocolo MQTT no hay una explicación. Por lo tanto la duda queda abierta.

5.7 Desarrollo de la etapa 5: desarrollo de la báscula digital IoT

Haciendo uso de una báscula digital preexistente, se soldó el microcontrolador al driver de leds utilizado para iluminar los displays de la báscula. Dicho driver se encarga de iluminar los 6 dígitos que se encuentran en la báscula, debido a que el peso máximo posible es de 999 libras, solo es necesario obtener la información de los segmentos 1,2 y 3.



Fig. 43 báscula digital a modificar (fuente propia)

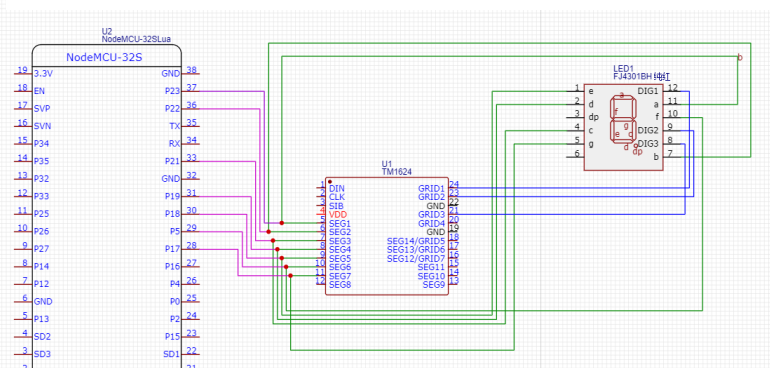


Fig. 66 diagrama de conexión del microcontrolador esp8266 entre el driver de leds y el display de 7 segmentos (fuente propi)

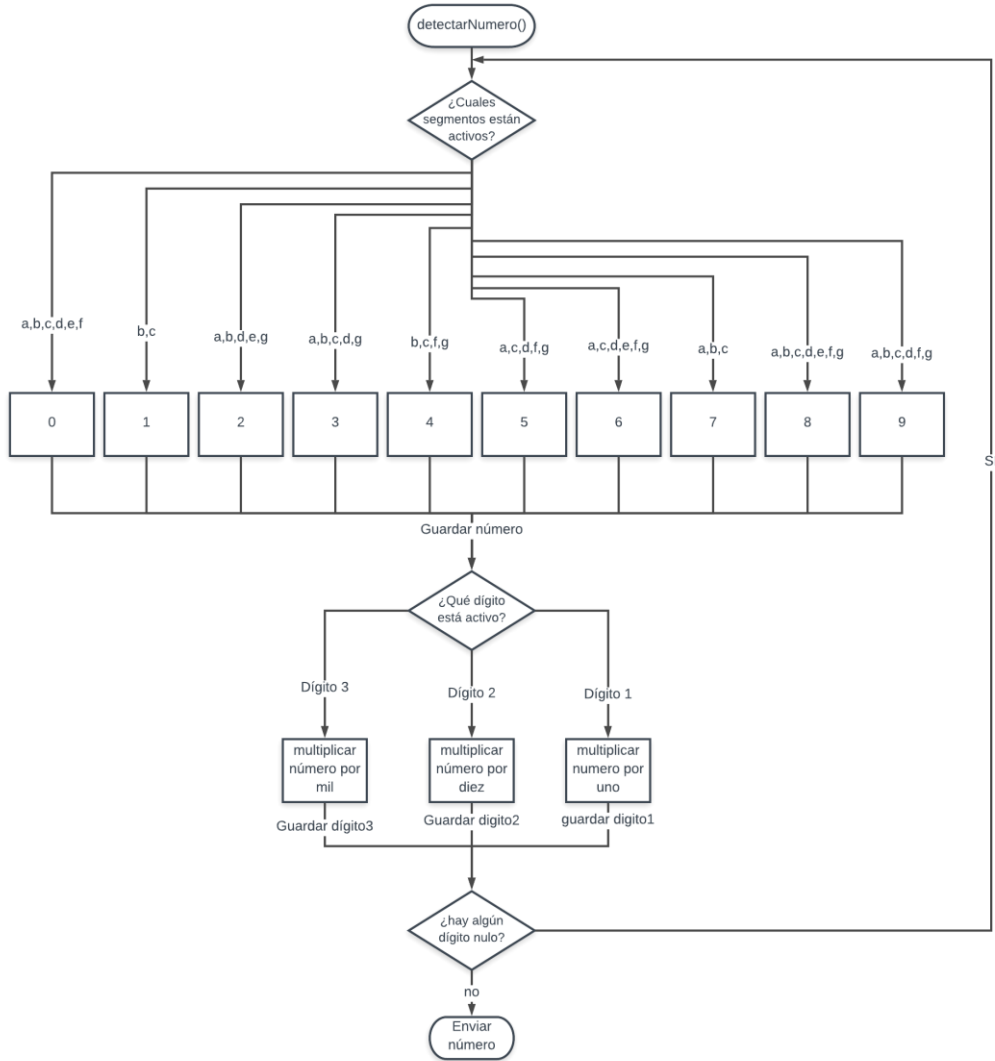


Fig. 67 algoritmo para obtener el número enviado al display de 7 segmentos por la báscula digital (fuente propia)

5.7.1 Pruebas dentro de la SPA

Primero se probó el funcionamiento en conjunto a la SPA. Los números presentados por la báscula y los números vistos en el formulario de ingresoFacturas coincidían prácticamente al mismo tiempo, y la distancia de comunicación no resultó ser ningún problema, ya que mientras se disponga de una red local la comunicación puede ser ejecutada, esto implica que la distancia máxima a la que puede ser conectado un dispositivo es determinada por la capacidad de cobertura de la red.

5.8 Ubicación de los puntos de recepción

Beneficio Rio Frio brinda sus servicios a las siguientes aldeas ubicadas en el departamento de Santa Bárbara y el departamento de Lempira:

- Río frío,
- Los hoyos,
- Las marías,
- Hierba buena,
- Plan del higo,
- Joconal,
- Choloma,
- El porvenir,
- El pacayal,
- El zapotillo,
- San Benito,
- Montes de oro,
- La balastrera,
- El terrero,
- San bartolo,
- Camoterias,
- La ceibita,
- La zona,
- El filo,

De esta lista de aldeas se seleccionó, Río Frio, San Bartolo, Las Marías y Choloma como las primeras en la que se implementará un punto de recepción basado en la SPA ya que son las aldeas más grandes y pueden brindar sus servicios a las aldeas aledañas. Con dichas localidades definidas se realizó el siguiente mapa y se calculó la distancia máxima entre los puntos.

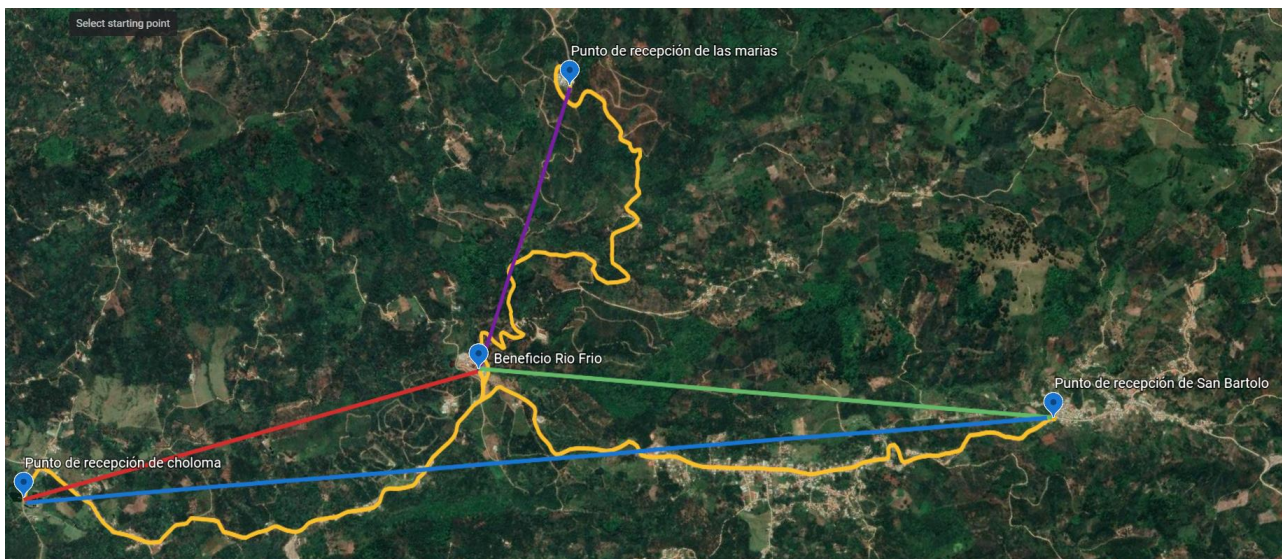


Fig. 44 Ubicación de los puntos de recepción (fuente propia)

En este mapa se pueden ver en amarillo el camino recorrido en carro desde cualquiera de los 4 puntos a otro. Sin embargo, extender una red local utilizando radio frecuencia requiere una línea directa de visión, dichas líneas directas de visión están señalizadas por las líneas en morado, rojo y verde, la línea celeste es la conexión más larga dentro de este mapa:

- Celeste- Línea directa entre punto de recepción de San Bartolo y Punto de recepción de Choloma. (consta de 5,681.91 m aproximadamente).
- Morado- Línea directa entre punto de recepción de las Marías y Beneficio Río Frío. (consta de 1,607.42 m aproximadamente).

- Rojo- Línea directa entre Beneficio Río Frío y punto de recepción de Choloma. (Consta de 2,638.32 m aproximadamente.)
- Verde- Línea directa entre Beneficio Río Frío y punto de recepción de San Bartolo. (Consta de 3,127.82 m aproximadamente)

Con estos datos presentados, y siendo Beneficio Río Frío el lugar donde se encontrarán los equipos necesarios para crear la red local, se necesita conectar Beneficio Río Frío al punto de recepción San Bartolo (3,127.82 m), al punto de recepción de Choloma (2,638.32 m) y al punto de recepción de Las Marías (1,607.42 m). Por lo tanto siempre y cuando la red Local se mantenga en línea se podría comunicar un dispositivo IoT con cliente de la SPA a un máximo aproximado de 5,681.91 m.

En cada plantel de los puntos de recepción es posible albergar hasta 5 vehículos pickup al mismo tiempo respetando las medidas de distanciamiento social, esto mejora a la situación actual de Beneficio Río Frío en el que se llega a tener hasta 15 vehículos altamente cargados, tanto de café como de personas desde las 4pm hasta la 1am 6 días a la semana sin importar las condiciones meteorológicas.

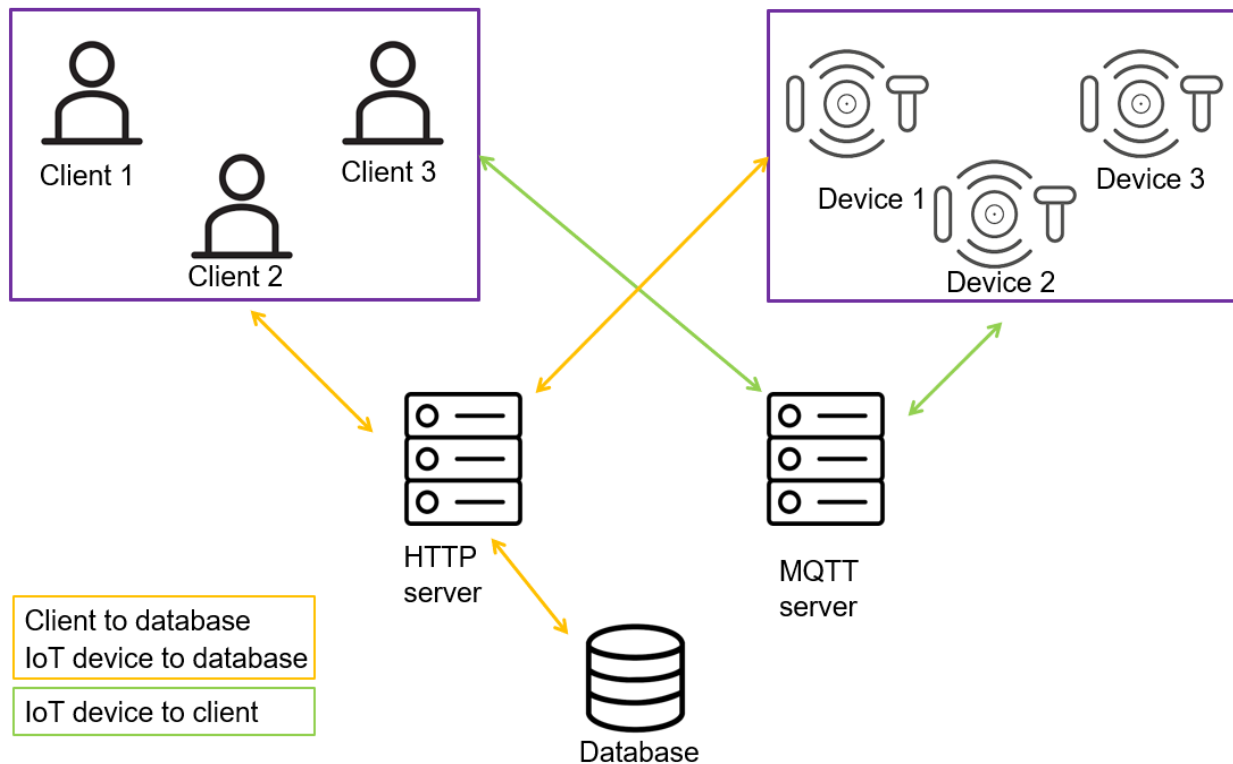


Fig. 45 Resultado Final

Con estas nuevas estaciones se obtiene este esquema de trabajo, en el que tres clientes simultáneos

pueden ser atendidos gracias a las básculas IoT instaladas, Cada uno de los puntos de recepción tiene acceso a la misma base de datos y se puede comunicar con cualquiera de las básculas permitiendo que se reduzca la migración de personas y abriendo las puertas a la recepción remota pues los 3 puntos de recepción pueden ser atendidos por una sola persona en el plantel de Beneficio Rio Frio.

CAPÍTULO VI: CONCLUSIONES

El poder agregar puntos de recepción en cualquier lugar donde la red Local se extienda le da la oportunidad a Beneficio Río Frio de reducir la aglomeración de personas colocando los primeros 3 puntos de recepción en las afueras de Beneficio Río Frío, en la aldea Las Marías y la aldea Choloma, estos 4 lugares constan de suficiente espacio para albergar a 5 vehículos pick up al mismo tiempo sin que se tengan que romper las medidas de distanciamiento social y reduce el desplazamiento de personas.

De acuerdo con la distancia máxima entre los 4 puntos, la red local se tendrá que extender un total de 5,681.91 m que permitirá visualizar una SPA que es fácil de utilizar, que no presenta errores de diseño al ser vista en dispositivos móviles o de escritorio, cada formulario fue construido en un componente completamente separado del resto para respetar la modularidad de la página y se tomaron los pasos necesarios para que la SPA pueda seguir creciendo sin afectar ninguna otra parte.

El protocolo MQTT ha dejado demostrada la baja latencia a la hora de enviar mensajes de pocos bytes. Así como en esta investigación se probó que es capaz de enviar datos obtenidos de una báscula digital en tiempo real, en el trabajo de (Aroon, 2016) lograron controlar un robot usando rastreo por GPS y enviando las comandos de movimiento por medio un servicio MQTT alojado en la nube.

El uso del protocolo MQTT hace que la adquisición de datos del mundo real sea mucho más fácil, permitiendo crear amplias redes de sensores, pues de acuerdo con lo visto en los resultados se pueden conectar múltiples dispositivos sin afectar significativamente el rendimiento del protocolo de comunicación

CAPÍTULO VII: RECOMENDACIONES

El campo de IoT es relativamente nuevo cuando se compara con otras tecnologías, es por eso por lo que aún hay un amplio margen de mejora.

La primer recomendación sería lograr descongestionar al broker MQTT de mensajes publicados bajo un tópico con cero suscriptores, en trabajo parecido fue hecho por (Tanomwong & Jaikaeo, 2018) en el que utilizó el sistema de tópicos utilizado en MQTT para crear un middleware que permite especificar si no se quiere enviar el mensaje si este tópico no tiene suscriptores.

Una segunda recomendación surge por parte de lo presentado en la investigación de (Harsha et al., 2018) quien demuestra que el protocolo MQTT al momento de realizar una desconexión envía las credenciales del usuario en texto plano, dichas credenciales pueden ser utilizadas para acceder a información sensible o dañar el funcionamiento de la red.

Este proyecto constituye las bases para crear un sistema de facturación más complejo que pueda conectarse con Impresoras, cámaras, altavoces y micrófonos por medio de la red local mientras que dispositivos como botones, básculas y sensores de movimiento por medio del IoT

CAPÍTULO VIII: ANEXOS

8.1 Resultado de las pruebas

8.1.1 tamaño del mensaje: 5bytes, usuarios :1

Promedio (ms): 28.80, minimo: 24, maximo: 36

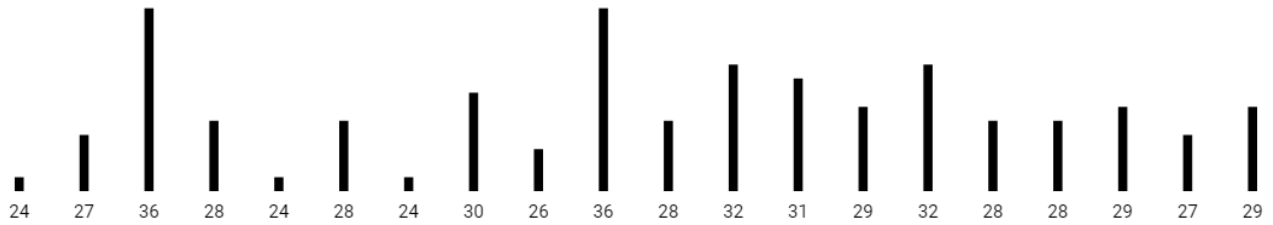


Fig. 46 prueba 1 en usuario 1

Promedio (ms): 26.75, minimo: 23, maximo: 32

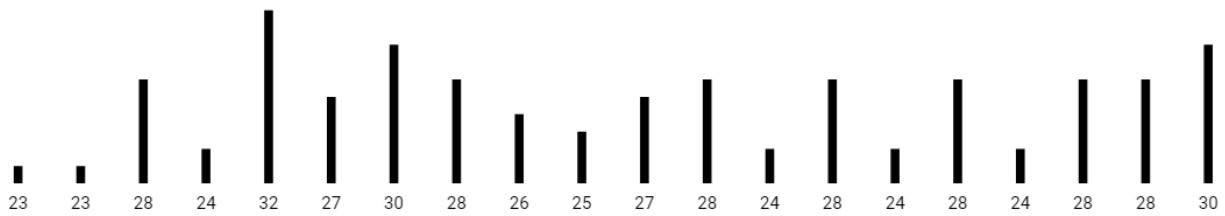


Fig. 47 prueba 2 en usuario 1

Promedio (ms): 26.00, minimo: 23, maximo: 32

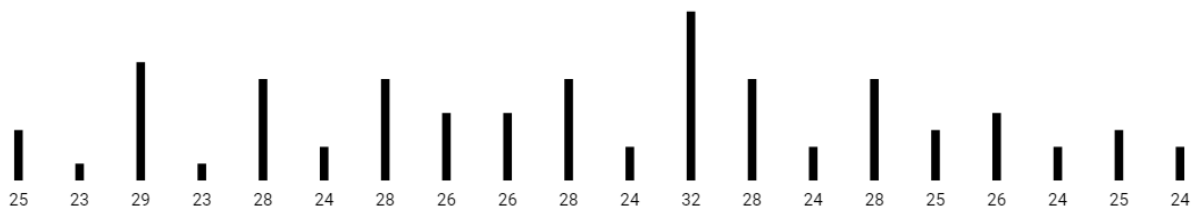


Fig. 48 prueba 3 en usuario 1

Promedio (ms): 27.30, minimo: 23, maximo: 48

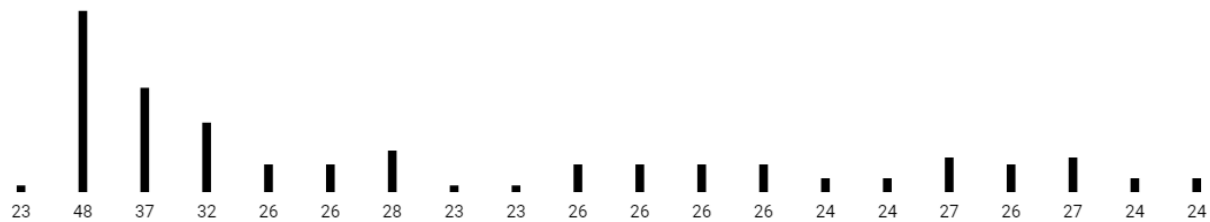


Fig. 49 prueba 4 en usuario 1

8.1.2 tamaño del mensaje: 5bytes, usuarios: 2

Promedio (ms): 19.30, minimo: 16, maximo: 25

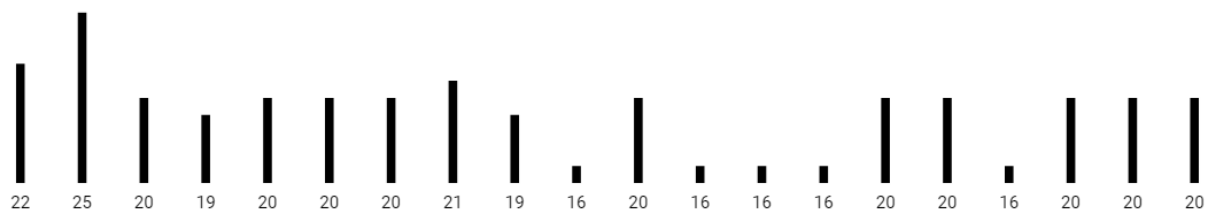


Fig. 50 prueba 1 en usuario 1

Promedio (ms): 16.25, minimo: 3, maximo: 33

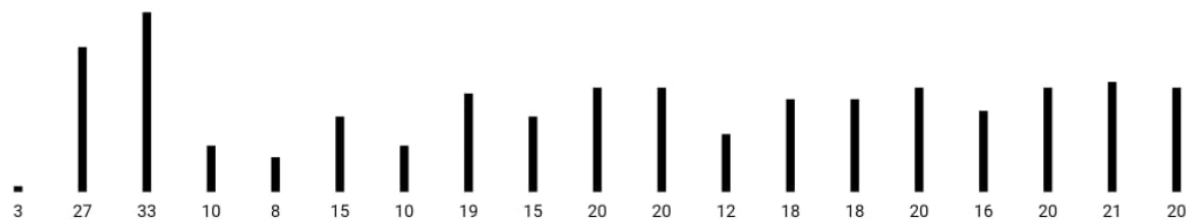


Fig. 51 prueba 1 en dispositivo 2

Promedio (ms): 19.70, minimo: 16, maximo: 24

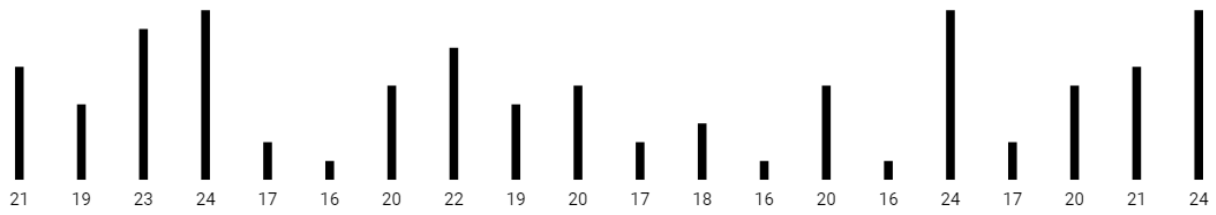


Fig. 52 prueba 2 en dispositivo 1

Promedio (ms): 18.45, minimo: 5, maximo: 51

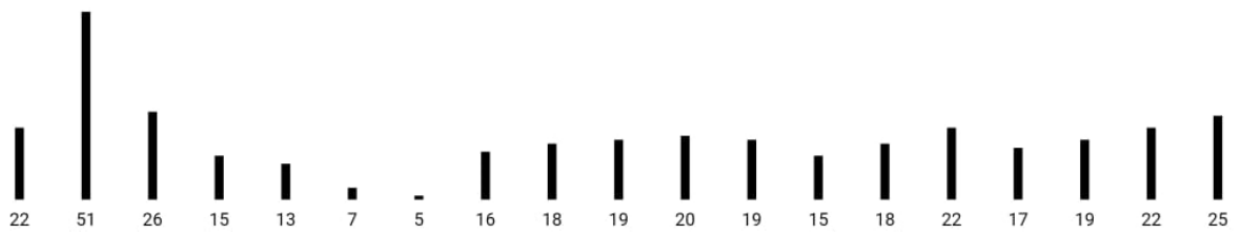


Fig. 53 prueba 2 en dispositivo 2

8.1.3 tamaño del mensaje: 5bytes, usuarios: 3

Promedio (ms): 16.95, minimo: 14, maximo: 22

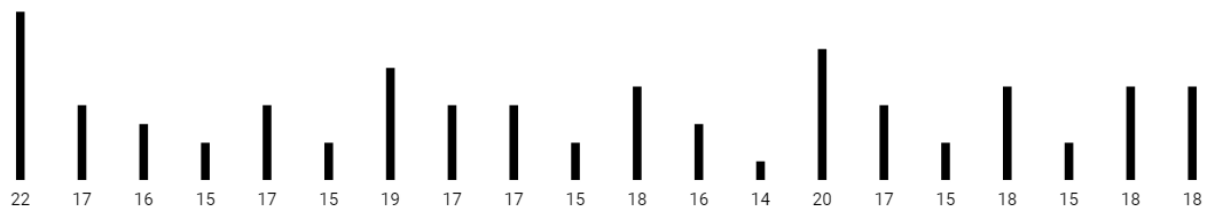


Fig. 54 prueba 1 en dispositivo 1

Promedio (ms): 11.10, minimo: 4, maximo: 49

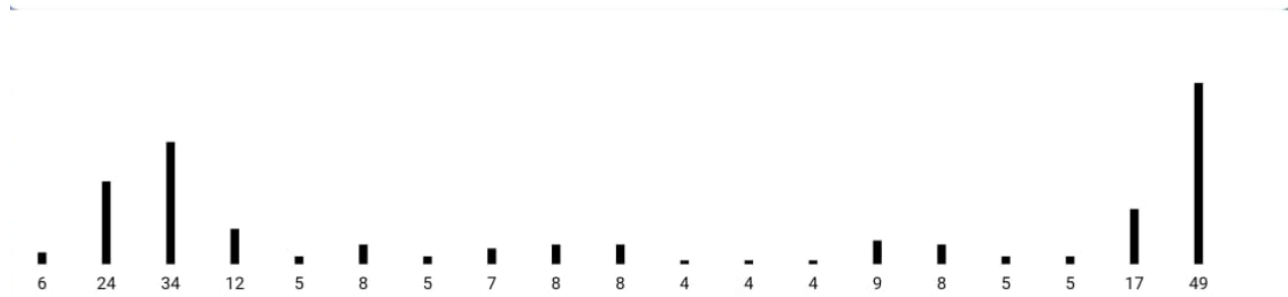


Fig. 55 prueba 1 en dispositivo 2

Promedio (ms): 15.35, minimo: 9, maximo: 20

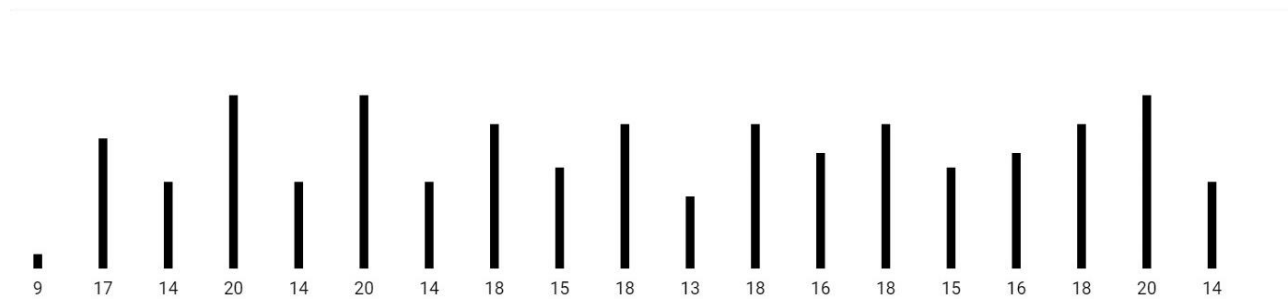


Fig. 56 prueba 1 en dispositivo 3

8.1.4 tamaño del mensaje: 5bytes, usuarios: 5

Promedio (ms): 21.30, minimo: 11, maximo: 60

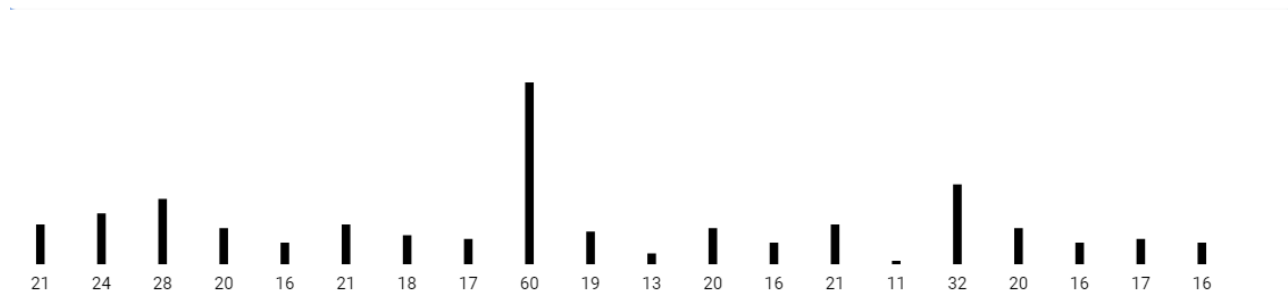


Fig. 57 prueba 1 en dispositivo 1

Promedio (ms): 16.10, minimo: 1, maximo: 49

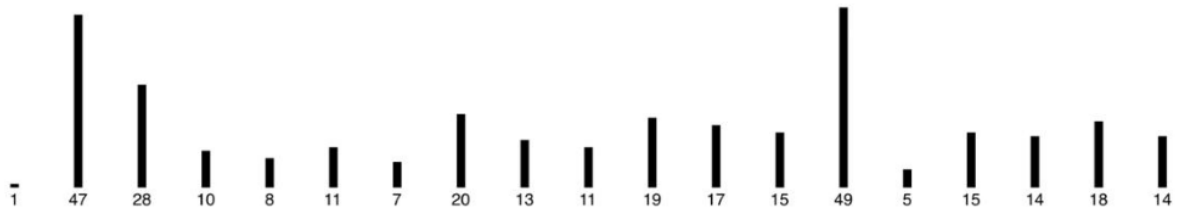


Fig. 58 prueba 1 en dispositivo 2

Promedio (ms): 16.10, minimo: 1, maximo: 49

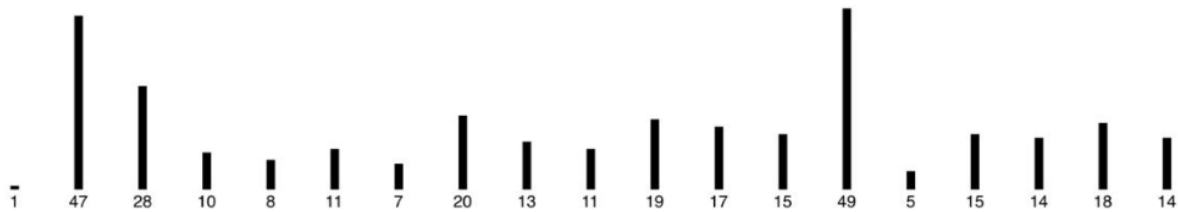


Fig. 59 prueba 1 en dispositivo 3

Promedio (ms): 20.15, minimo: 15, maximo: 45

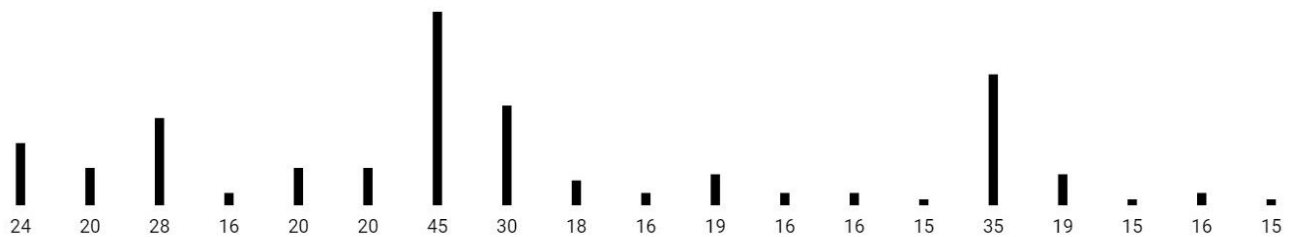


Fig. 60 prueba 1 en dispositivo 4

Promedio (ms): 16.70, minimo: 10, maximo: 35

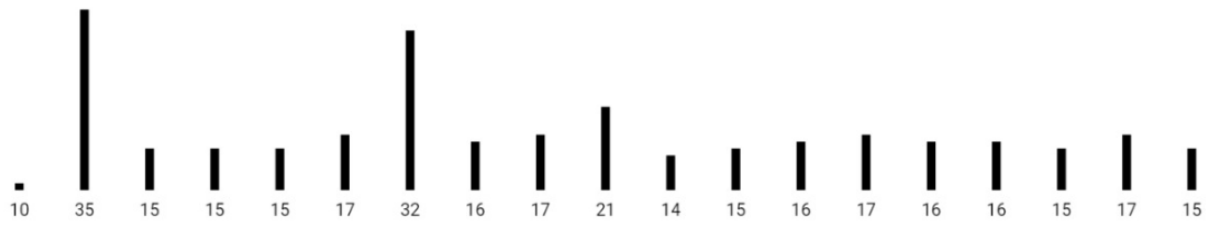


Fig. 61 prueba 1 en dispositivo 5

8.1.5 tamaño del mensaje: 50bytes, usuarios: 1

Promedio (ms): 16.90, minimo: 9, maximo: 23

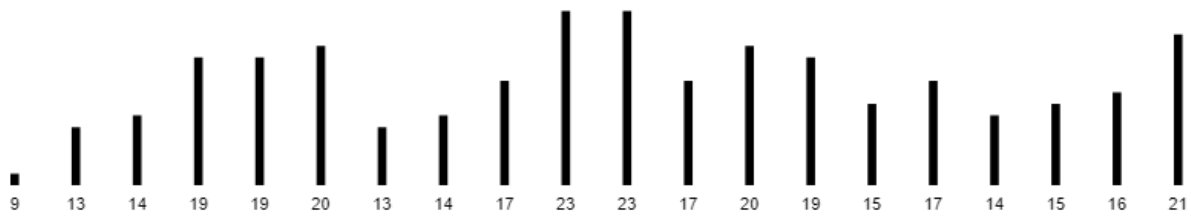


Fig. 62 prueba 1 envío de mensaje de 50bytes

Promedio (ms): 15.65, minimo: 8, maximo: 26

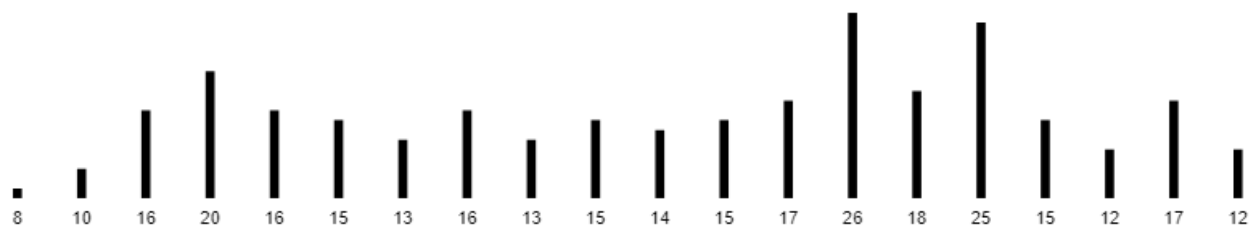


Fig. 63 prueba 2 envío de mensaje de 50bytes

8.1.6 tamaño del mensaje: 100bytes, usuarios: 1

Promedio (ms): 20.65, minimo: 12, maximo: 35

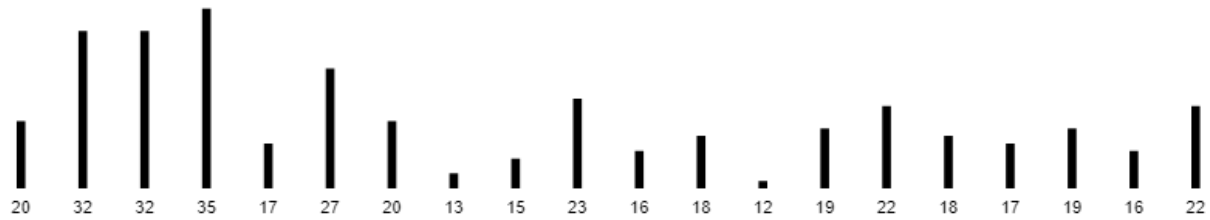


Fig. 64 prueba 1 envío de mensaje de 100bytes

Promedio (ms): 17.40, minimo: 10, maximo: 28

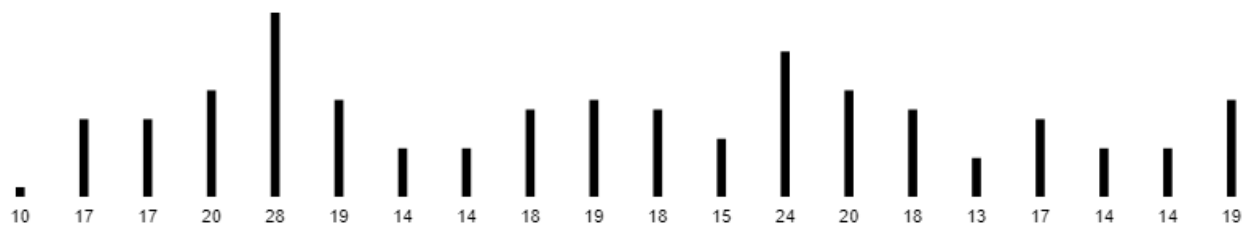


Fig. 65 prueba 2 envío de mensaje 100bytes

8.2 Manual de uso

8.2.1 formulario de Login

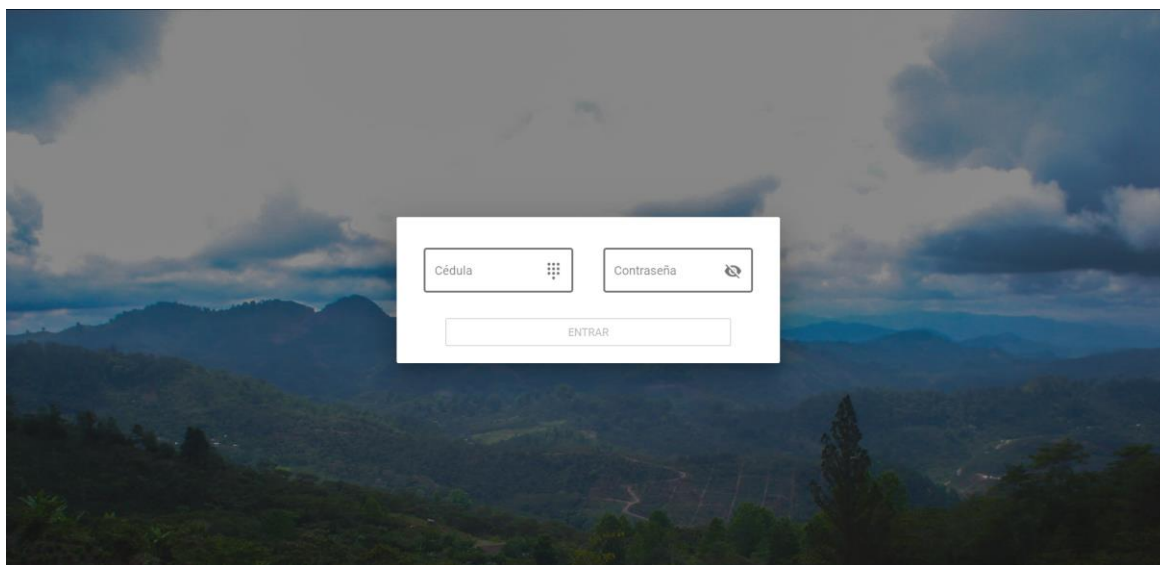


Fig. 66 Página inicial

Para ingresar al sistema se requiere coincidir la cédula con su contraseña. Está validado tanto del lado del front-end como del back-end, se envía la información al servidor por medio de petición tipo POST al servidor.

8.2.2 Sección de noticias y mensajes

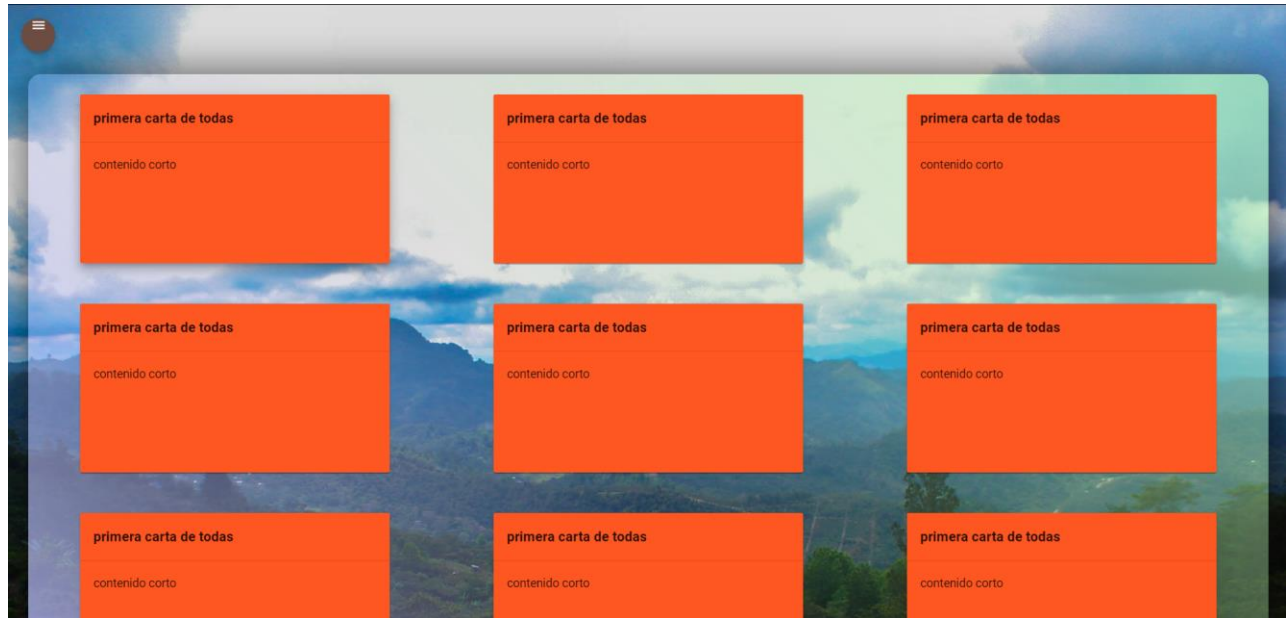


Fig. 67 Sección de noticias y mensajes

Esta sección está pensada para presentar noticias y mensajes, por el momento no están conectados a la base de datos

8.2.3 Menú desplegable

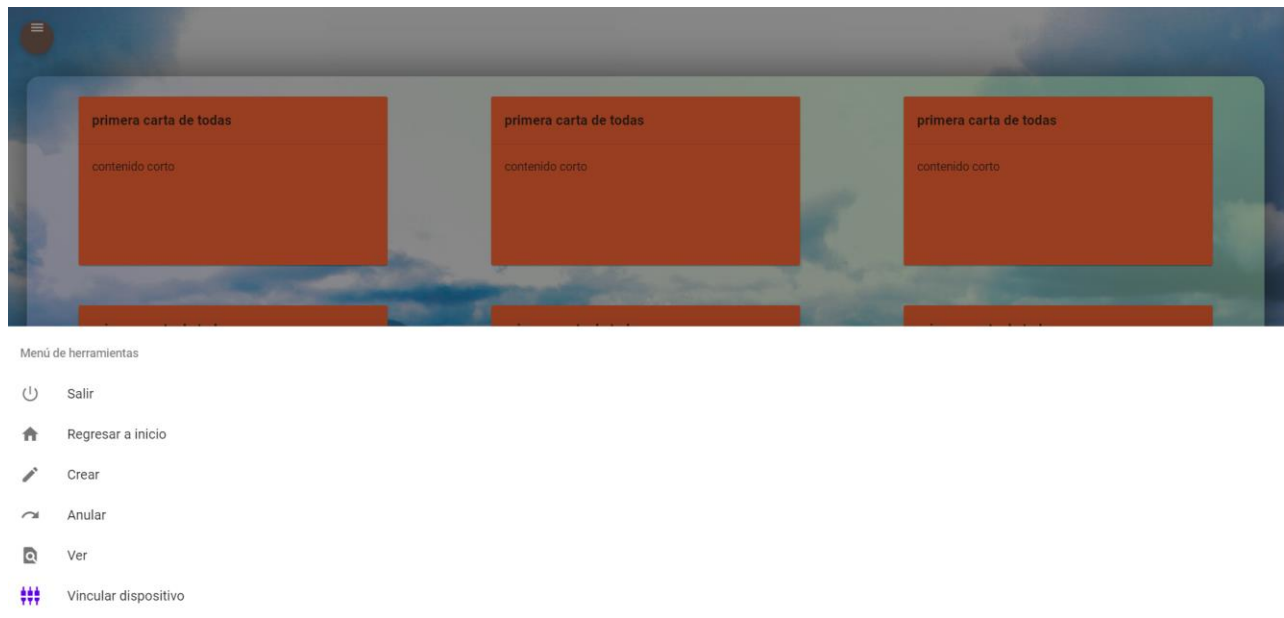


Fig. 68 menú desplegable

- **Botón Salir:** Cierra la sesión
- **Botón regresar a inicio:** retorna a la sección de noticias y mensajes
- **Botón crear:** ingreso al indexador de formularios con la intención de crear
- **Botón anular:** ingreso al indexador de formularios con la intención de anular
- **Botón ver:** ingreso al indexador de formularios con la intención de ver
- **Botón vincular dispositivos:** Abre la ventana emergente de conexión a los dispositivos IoT

8.2.4 Indexador de formularios

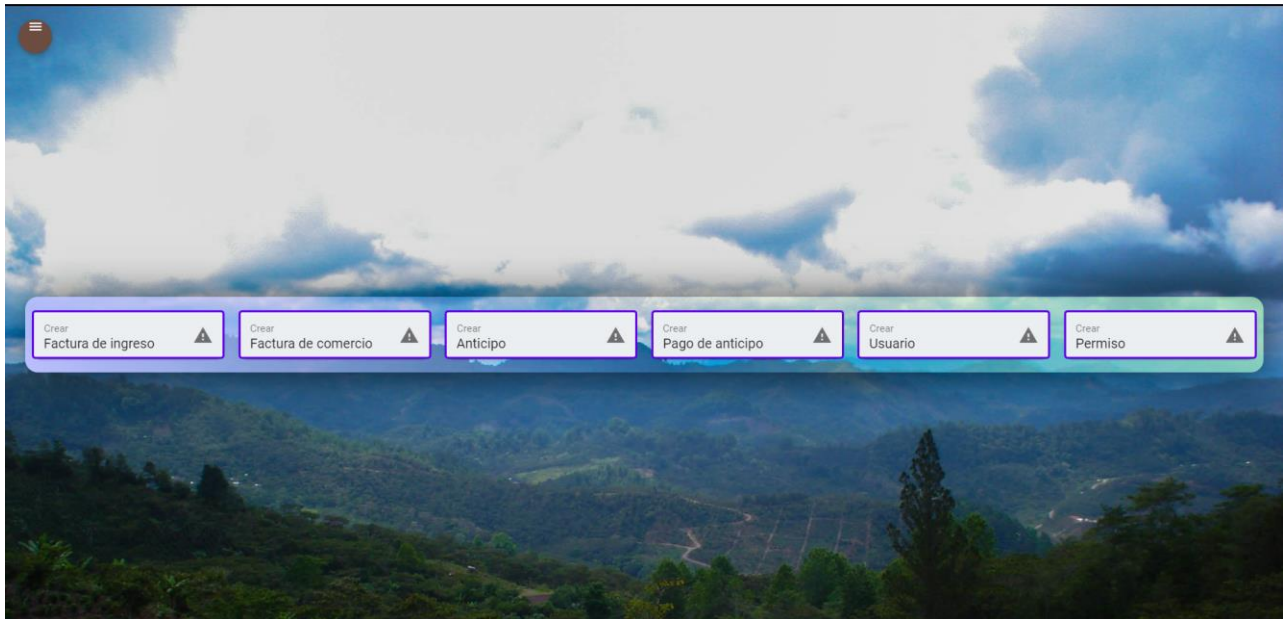


Fig. 69 Indexador de formularios

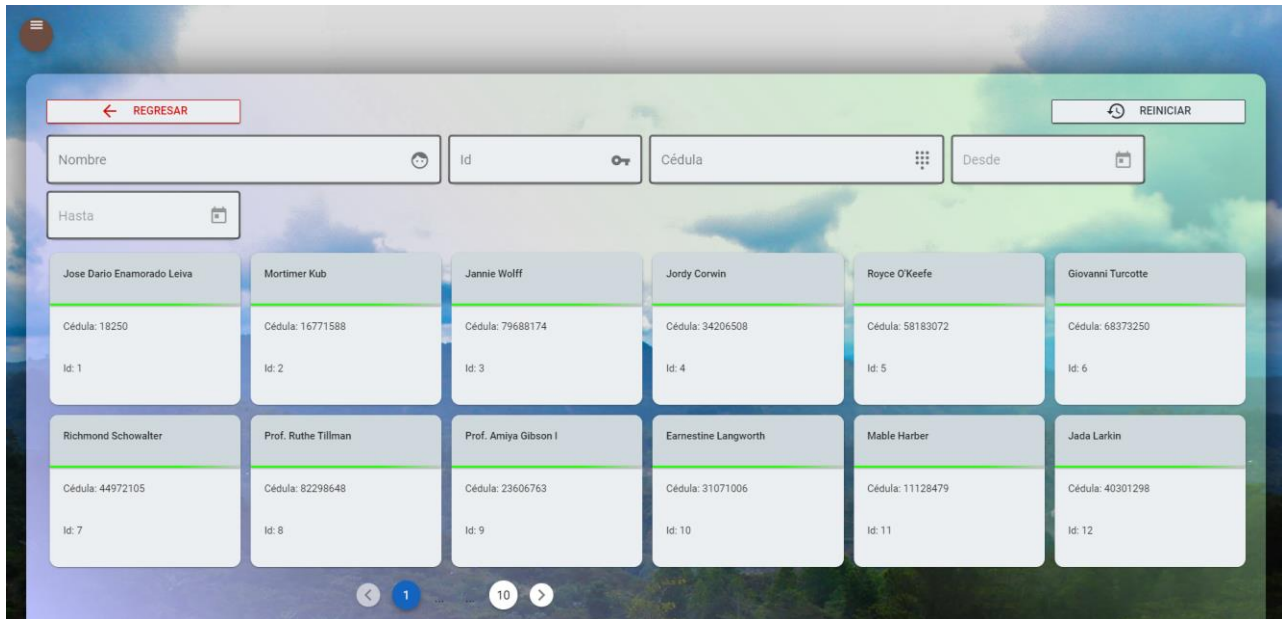
anular y ver del menú desplegable. De acuerdo con la intención con la que se accede al indexador de formulario el contorno de las etiquetas de texto toman un color u otro:

Crear: Morado.

Anular: Rojo

Ver: Verde

8.2.5 Indexador de usuarios



Se accede a esta sección cada vez que se requiere elegir un usuario en específico. Debido a que no se espera una cantidad de usuarios mayor a 1,500 usuarios al ingresar a esta sección se carga a la memoria del cliente el nombre, cédula e id de todos los usuarios registrados, esto agiliza el proceso de búsqueda pues utilizando los campos búsqueda el usuarios se filtran con cada actualización en la información que estos contengan.

- **Campo nombre:** filtra los usuarios seleccionando aquellos que contengan en cualquier parte de su nombre el texto escrito dentro del campo.
- **Campo id:** filtra los usuarios seleccionando aquellos que contengan en cualquier parte de su id el número escrito dentro del campo.
- **Campo cédula:** filtra los usuarios seleccionando aquellos que contengan en cualquier parte de su cédula el número escrito dentro del campo
- **Campo Desde:** filtra los usuarios seleccionando aquellos que hayan sido registrados después de la fecha ingresada.
- **Campo Hasta:** filtra los usuarios seleccionando aquellos que hayan sido registrados antes de la fecha ingresada.
- **Cartas de resultados:** Cada usuario seleccionado a partir del proceso de filtrado es representado en forma de carta, hacer clic en una carta selecciona al usuario que esa carta representa

- **Barra de paginación:** Cada página solo presenta hasta 12 cartas de resultados, en caso de que hayan más de 12 cartas de resultados la barra de paginación nos permite ver el siguiente grupo de 12 cartas de resultados

8.2.6 Indexador de ingresoFacturas

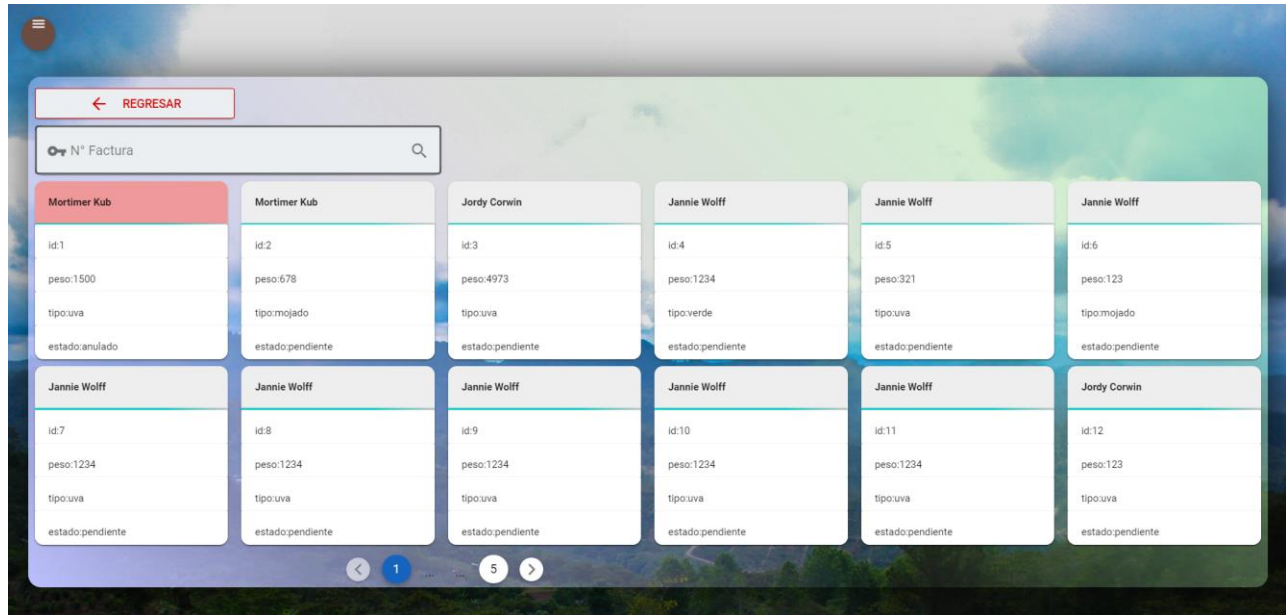


Fig. 70 Indexador de registros

Esta sección nos permite elegir un registro en concreto de una tabla de la base de datos, pero para este proyecto se limita al caso de las ingresoFacturas.

- **Campos de filtrado:** presenta el último campo de filtrado relleno, al hacer clic en la ícono de la izquierda aparece un menú desplegable. En dicho menú se selecciona el campo por el cuál se quiere filtrar, se puede seleccionar más de un filtro y la opción "Reiniciar Filtros"

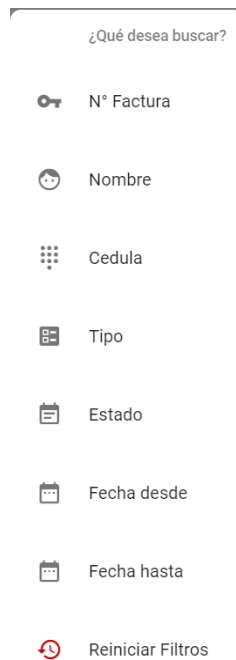


Fig. 71 Menú desplegable de campos de filtrado

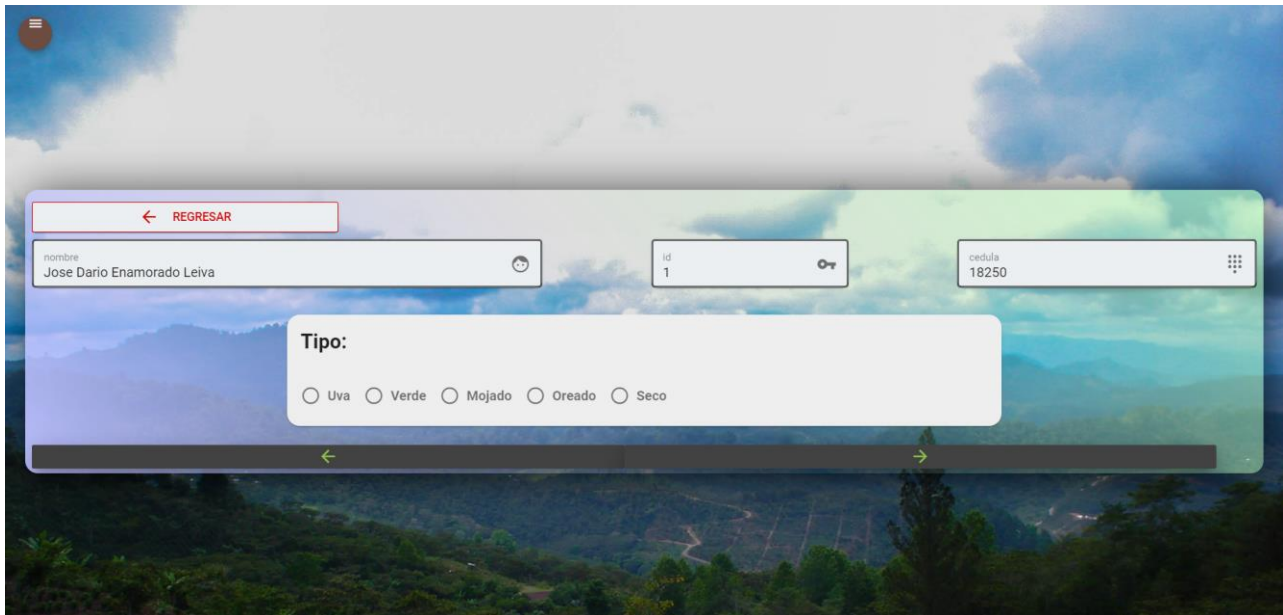
facilita el borrado de todos los filtros seleccionados

- **Cartas de resultados:** Cada usuario seleccionado a partir del proceso de filtrado es representado en forma de carta, hacer clic en una carta selecciona al usuario que esa carta representa
- **Barra de paginación:** Cada página solo presenta hasta 12 cartas de resultados, en caso de que hayan más de 12 cartas de resultados la barra de paginación nos permite ver el siguiente grupo de 12 cartas de resultados

8.2.7 formulario de ingresoFacturas

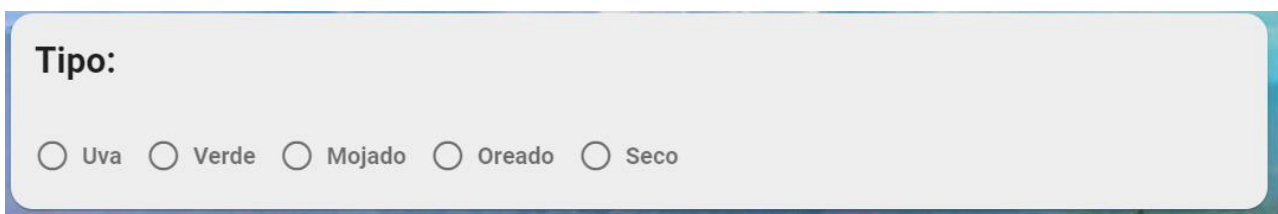
8.2.7.1 formulario de ingresoFacturas con intención Crear

Para mantener todo el formulario visible sin necesidad de barras de desplazamiento se dividió el formulario en 5 pasos.



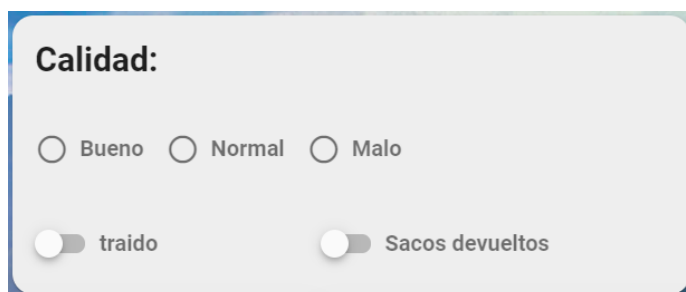
The screenshot shows a mobile application interface for creating an invoice entry. At the top left, there is a red button labeled 'REGRESAR' with a left-pointing arrow. Below it, there are three input fields: 'nombre' containing 'Jose Dario Enamorado Leiva', 'id' containing '1', and 'cedula' containing '18250'. A central white box titled 'Tipo:' contains five radio button options: 'Uva', 'Verde', 'Mojado', 'Oreado', and 'Seco'. At the bottom of the form, there is a dark grey bar with a left-pointing arrow on the left and a right-pointing arrow on the right.

Fig. 73 formulario ingresoFacturas con intención Crear



This is a close-up of the 'Tipo:' section from the previous screenshot. It shows the title 'Tipo:' in bold, followed by five radio button options: 'Uva', 'Verde', 'Mojado', 'Oreado', and 'Seco'.

Fig. 72 Paso 1 del formulario ingresoFacturas con intención crear



This is a close-up of the 'Calidad:' section. It features the title 'Calidad:' in bold, followed by three radio button options: 'Bueno', 'Normal', and 'Malo'. Below these are two toggle switches: 'traido' (which is currently turned on) and 'Sacos devueltos' (which is currently turned off).

Fig. 73 Paso 2 del formulario ingresoFacturas con intención crear



This is a close-up of the bottom section of the form. It shows three input fields: 'Humedad' followed by a percentage sign, 'finca', and 'observaciones'.

Fig. 72 Paso 3 del formulario de ingresoFacturas con intención crear


+ AGREGAR		- QUITAR	
Sacos		Peso	
1			
peso:2			
sacos:1			

Fig. 74 Paso 4 del formulario de ingresoFacturas con intención crear

Toque para guardar:

Propietario: Mortimer Kub

Peso Total: 123

Tara: 10

Tipo: uva

Calidad: bueno

Humedad: 23

Total Neto: 113

Total seco: 87.01

Fig. 75 Paso 5 del formulario de ingresoFacturas con intención crear

- **Cabecera:** son los campos de texto ubicados en la parte superior y presentan al usuario seleccionado con anterioridad en el indexador de usuarios
- **Paso 1:** selecciona el tipo de café ingresado, es obligatorio
- **Paso2:** selecciona la calidad del café ingresado, si fue traído o no y si se le devolvieron los sacos o no
- **Paso3:** contiene 3 campos de texto en los que se especifica la humedad del café, la finca de la cual proviene y observaciones que se pueden agregar
- **Paso4:** Requiere rellenar el campo sacos y peso para luego presionar agregar, al presionar

agregar se genera una carta que guarda la información de dicha pesada, múltiples pesadas se pueden agregar y son sumadas al final. El botón quitar elimina la última pesada agregada

- **Paso5:** Contiene el resumen de la información ingresada en el formulario. Al hacer clic en esta carta, se abre el formulario de confirmación

8.2.7.2 formulario de ingresoFacturas con intención ver

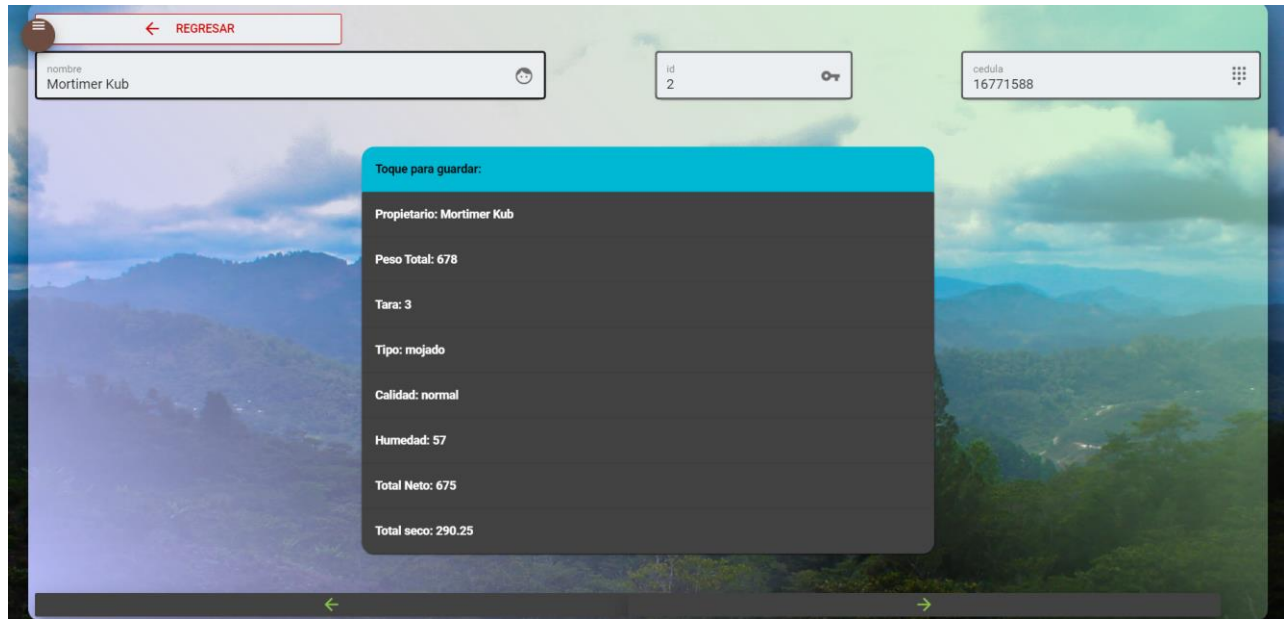


Fig. 76 formulario de ingresoFacturas con intención ver

Al abrir un registro con la intención ver solo se presenta el resumen de dicho registro. Este formulario no tiene ninguna interacción extra

8.2.7.2 formulario ingresoFacturas con intención anular

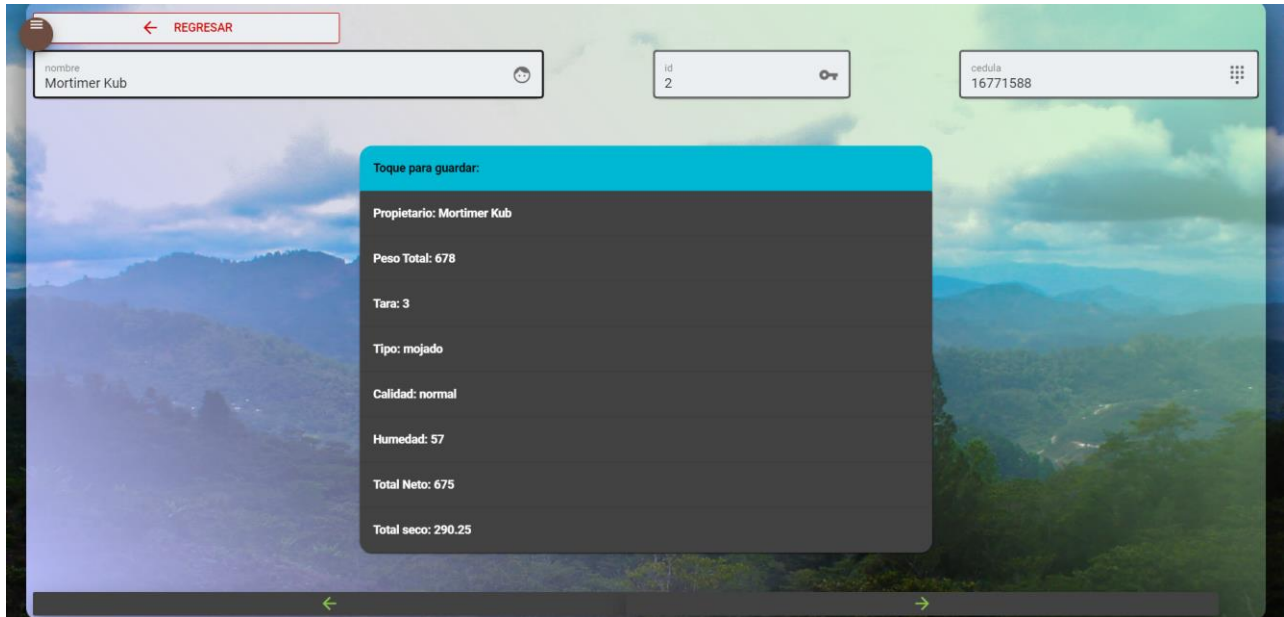
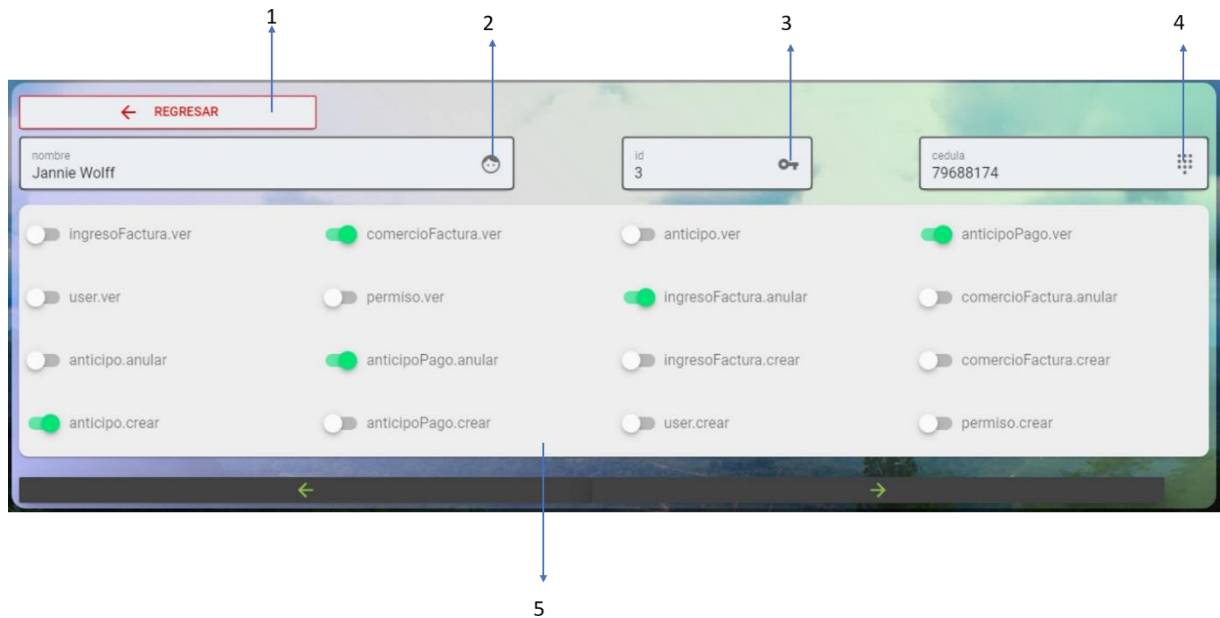


Fig. 77 formulario ingresoFacturas con intención anular

Al abrir un registro con la intención anular solo se presenta el resumen de dicho registro, pero al hacer clic en la carta del resumen se abre el formulario de confirmación. Visualmente no presenta gran cambio respecto al formulario de Ver

8.2.8 formulario de permisos

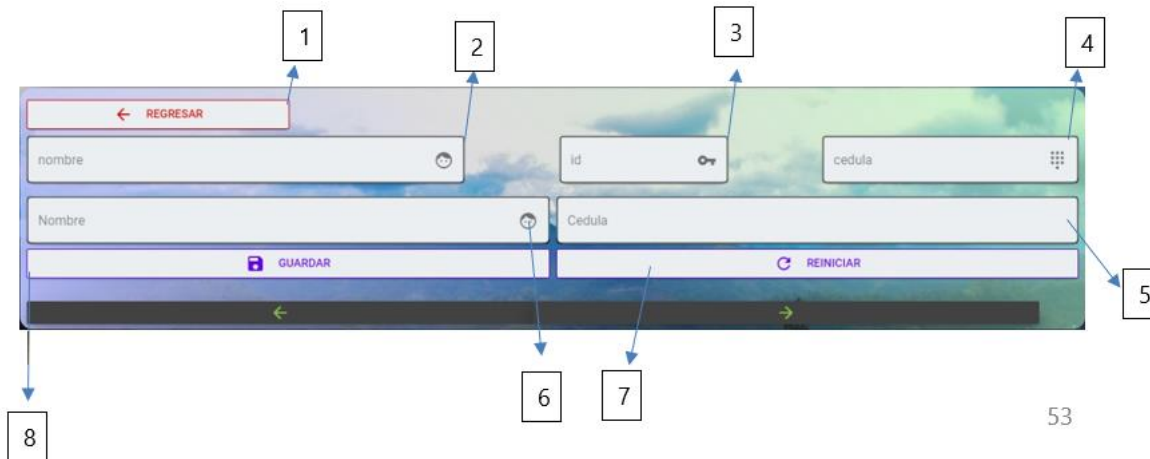


- **Botón 1 "regresar"**: regresa a la página de selección de tabla
- **Campos 2, 3, 4**: Exponen los datos del usuario al cual se le están asignando los permisos

- **Bloque 5 “Permisos”:** Renderiza un seleccionador booleano por cada permiso guardado en la tabla permisos

Este formulario solo está disponible con las intenciones de Crear y Ver.

8.2.9 formulario de registro



- **Botón 1 “regresar”:** elimina los datos del formulario y regresa a la página de selección de tabla
- **Campos 2, 3, 4:** Exponen los datos del último usuario registrado, esto tiene como objetivo comprobar que el registro se haya hecho de manera correcta, pues esos datos son obtenidos directamente de la tabla usuarios
- **Campo 5 “Cédula”:** obligatorio, debe contener exactamente 13 dígitos.
- **Campo 6 “Nombre”:** obligatorio, debe contener el nombre completo del usuario.
- **Botón 7 “Reiniciar”:** elimina los datos en campo 5 y 6 sin guardar nada.
- **Botón 8 “Guardar”:** realiza una petición Ajax para guardar los datos del campo 5 y 6 en la tabla usuarios.

8.2.10 ventana emergente de conexión MQTT

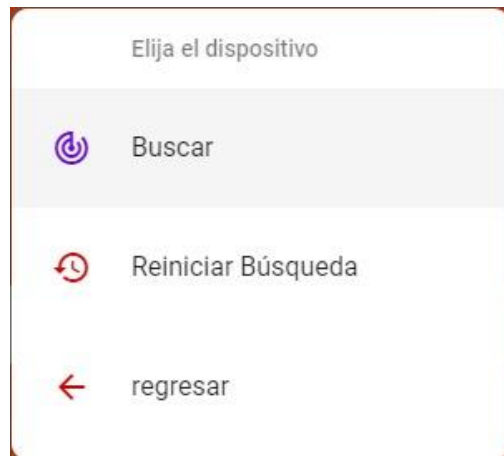


Fig. 78 Ventana emergente de conexión MQTT (fuente propia)

- **Opción "buscar":** Al ser pulsado este botón se publica un mensaje bajo el tópico MQTT "dispositivos", cuando cualquiera de los dispositivos recibe este mensaje debe responder publicando su nombre bajo el tópico "página" de esta manera, la SPA sabe que ese dispositivo existe.
- **Opción "Reiniciar Búsqueda":** ya que solo es necesario que el dispositivo responda 1 vez para que este sea detectado, se puede dar el caso que el dispositivo sea desconectado mientras esta ventana sigue activa. Al presionar este botón todos los dispositivos que han sido detectados son eliminados para realizar una búsqueda nueva.
- **Opción "regresar":** cierra la ventana, y elimina a los dispositivos.

Al ser detectado un dispositivo solo hace falta seleccionarlo para que la SPA se suscriba al nombre del dispositivo.

8.2.11 formulario de confirmación

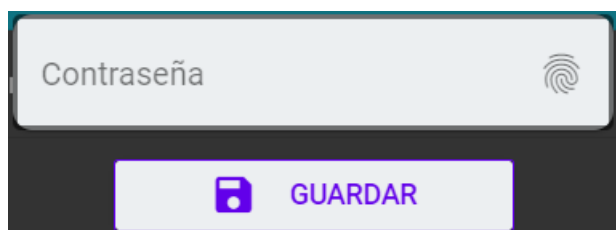


Fig. 79 formulario de confirmación

El formulario de confirmación tiene como único objetivo agregar una capa extra de seguridad,

solo requiere la contraseña del usuario con la sesión iniciada para funcionar

CAPÍTULO IX: REFERENCIAS

- Agudelo, M. (2020). *El estado de la digitalización de América Latina frente a la pandemia del COVID-19* EDITOR: CAF. 40.
- Alqinsi, P., Matheus Edward, I. J., Ismail, N., & Darmalaksana, W. (2018). IoT-Based UPS Monitoring System Using MQTT Protocols. *2018 4th International Conference on Wireless and Telematics (ICWT)*, 1-5. <https://doi.org/10.1109/ICWT.2018.8527815>
- Andy, S., Rahardjo, B., & Hanindhito, B. (2017). Attack scenarios and security analysis of MQTT communication protocol in IoT system. *2017 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, 1-6. <https://doi.org/10.1109/EECSI.2017.8239179>
- Aron, S. (2016). *Protocolos IIoT para considerar*. AADECA. https://editores-srl.com.ar/sites/default/files/aa2_semle_protocolos_ilot.pdf
- Aroon, N. (2016). Study of using MQTT cloud platform for remotely control robot and GPS tracking. *2016 13th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, 1-6. <https://doi.org/10.1109/ECTICon.2016.7561390>
- Ayanoglu, E., Aytas, Y., & Nahum, D. (2016). *Mastering RabbitMQ*. Packt Publishing Ltd.
- Bolaños, R. L., & Partanen, T. (s. f.). *LA RECOLECCIÓN DEL CAFÉ, UNA LABOR POR VISIBILIZAR*. 10.
- Bunn, C., Lundy, M., Läderach, P., Castro-Llanos, F., & Girvetz, E. H. (2018). *Café sostenible adaptado al clima en Honduras*. <https://cgspace.cgiar.org/handle/10568/105436>
- Cama, A. (2012). *Las redes de sensores inalámbricos y el Internet de las cosas*. 10.
- Cenicafe. (2016, marzo 31). *Cultivemos Café, Recolección* | www.cenicafe.org. https://www.cenicafe.org/es/index.php/cultivemos_cafe/recoleccion

- Deshmukh, S., Mane, D., & Retawade, A. (2019). Building a Single Page Application Web Front-end for E-Learning site. *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, 985-987. <https://doi.org/10.1109/ICCMC.2019.8819703>
- Dhanalaxmi, B., & Naidu, G. A. (2017). A survey on design and analysis of robust IoT architecture. *2017 International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, 375-378. <https://doi.org/10.1109/ICIMIA.2017.7975639>
- Diaz, N., Sanders, A., & Rendón, J. (2018). *Análisis de factibilidad para establecer una central de beneficio húmedo de café en El Paraíso, El Paraíso, Honduras* [Eap Zamorano]. <https://library.wur.nl/WebQuery/wurpubs/fulltext/450336>
- Eguiluz, J. (2008). *Introducción a XHTML*. 168.
- Federación nacional de cafetaleros de Colombia. (2016). *La recolección de café en Colombia mercado laboral.pdf*. https://federaciondecafeteros.org/static/files/La_recoleccion%C3%B3n_de_cafe_en_Colombiamercado_laboral.pdf
- Felipe, M. del R. C., Ortega, J. M. P., Ureta, G. D., & Zambrano, D. M. Z. (2018). Tecnologías de internet de las cosas en la obtención de información. *Dominio de las Ciencias*, 4(2), 147-160.
- FórumCafé. (2018). Café de Honduras. *Revista Fórum Café*. <https://www.revistaforumcafe.com/cafe-de-honduras>
- Gaoyang Shan, Sun-young Im, & Byeong-hee Roh. (2016). Optimal AdvInterval for BLE scanning in different number of BLE devices environment. *2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 1031-1032. <https://doi.org/10.1109/INFOCOMW.2016.7562238>

- Gonge, S. S., & Kandalkar, G. M. (2015). Combination of internet technology and education technology used for improvement in examination system of engineering. *2015 IEEE 3rd International Conference on MOOCs, Innovation and Technology in Education (MITE)*, 360-364. <https://doi.org/10.1109/MITE.2015.7375345>
- González, A. J. (s. f.). *Ingeniería de Software*. 32.
- Harsha, M. S., Bhavani, B. M., & Kundhavai, K. R. (2018). Analysis of vulnerabilities in MQTT security using Shodan API and implementation of its countermeasures via authentication and ACLs. *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2244-2250. <https://doi.org/10.1109/ICACCI.2018.8554472>
- IHCAFE. (2017, mayo 30). Regiones cafetaleras. *IHCAFE - Instituto Hondureño del Cafe*. <https://www.ihcafe.hn/regiones-cafetaleras/>
- International Coffee Organization. (2020, abril). *Coffee-break-series-1e.pdf*. <http://www.ico.org/documents/cy2019-20/coffee-break-series-1e.pdf>
- Jaimez-González, C., & Vargas-Rodríguez, R. (2017). Editor web visual para HTML, CSS y JavaScript de apoyo a la docencia. *Virtualidad, Educación y Ciencia*, 8(14), 136-152.
- Juarez, F. (2018, diciembre). *2018.-DICE-El-café-guatemalteco-un-enfonque-en-el-mercado-mundial-y-su-productividad-1.pdf*. <http://camcig.org/userfiles/2019/01/2018.-DICE-El-caf%C3%A9-guatemalteco-un-enfonque-en-el-mercado-mundial-y-su-productividad-1.pdf>
- Kodali, R. K., & Gorantla, V. S. K. (2017). Weather tracking system using MQTT and SQLite. *2017 3rd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*, 205-208. <https://doi.org/10.1109/ICATCCT.2017.8389134>
- Krombholz, K., Busse, K., Pfeffer, K., Smith, M., & von Zezschwitz, E. (2019). «If HTTPS Were Secure, I Wouldn't Need 2FA»—End User and Administrator Mental Models of HTTPS. *2019 IEEE*

- Symposium on Security and Privacy (SP)*, 246-263. <https://doi.org/10.1109/SP.2019.00060>
- Kwon, H., Nam, H., Lee, S., Hahn, C., & Hur, J. (2019). (In-)Security of Cookies in HTTPS: Cookie Theft by Removing Cookie Flags. *IEEE Transactions on Information Forensics and Security*, 1-1. <https://doi.org/10.1109/TIFS.2019.2938416>
- Liu, Q., Liu, Z., Li, D., Gao, Z., Zhu, J., Yang, J., & Wang, Q. (2020). *Assessing the Tendency of 2019-nCoV (COVID-19) Outbreak in China*. 27.
- Mendieta, T. P., Herrera, J., & Peña, A. J. (2019). La Capacidad del IOT de Transformar el Futuro. *Revista Avenir*, 1(1), 15-18.
- OMS. (s. f.-a). *Orientaciones para el público*. Recuperado 12 de mayo de 2020, de <https://www.who.int/es/emergencias/diseases/novel-coronavirus-2019/advice-for-public>
- OMS. (s. f.-b). *Preguntas y respuestas sobre la enfermedad por coronavirus (COVID-19)*. Recuperado 12 de mayo de 2020, de <https://www.who.int/es/emergencias/diseases/novel-coronavirus-2019/advice-for-public/q-a-coronaviruses>
- Otero Ortega, A. (s. f.). *ENFOQUES DE INVESTIGACIÓN*. 35.
- Romero, J. R., & Sibrian, F. J. (2016). *Estudio de la estructura de mercado de la comercialización del café en Honduras* [Eap Zamorano]. <https://bdigital.zamorano.edu/bitstream/11036/5874/1/AGN-2016-T043.pdf>
- Sadeghian, A., Zamani, M., & Ibrahim, S. (2013). SQL Injection Is Still Alive: A Study on SQL Injection Signature Evasion Techniques. *2013 International Conference on Informatics and Creative Multimedia*, 265-268. <https://doi.org/10.1109/ICICM.2013.52>
- Sadeq, A. S., Hassan, R., Al-rawi, S. S., Jubair, A. M., & Aman, A. H. M. (2019). A Qos Approach For Internet Of Things (Iot) Environment Using Mqtt Protocol. *2019 International Conference on Cybersecurity (ICoCSec)*, 59-63. <https://doi.org/10.1109/ICoCSec47621.2019.8971097>

- Sadio, O., Ngom, I., & Lishou, C. (2019). Lightweight Security Scheme for MQTT/MQTT-SN Protocol. *2019 Sixth International Conference on Internet of Things: Systems, Management and Security (IOTSMS)*, 119-123. <https://doi.org/10.1109/IOTSMS48152.2019.8939177>
- Salgado, H. R., Mendoza, E. C., Bravo, M. H., & Montero, J. A. (2019). *Desarrollo de prototipo SCADA para controlar el nivel de agua en dos contenedores de agua*. 1(1), 7.
- Shen, V. R. L., Wei, C.-S., & Juang, T. T.-Y. (2018). Javascript Malware Detection Using A High-Level Fuzzy Petri Net. *2018 International Conference on Machine Learning and Cybernetics (ICMLC)*, 2, 511-514. <https://doi.org/10.1109/ICMLC.2018.8527036>
- Silva, O., & Alejandra, M. (2019). *Desarrollo del geoportal para la gestión del sistema de información ALPA de la ESPOCH utilizando el framework Vue.js*.
<http://dspace.esPOCH.edu.ec/handle/123456789/11783>
- Stack Overflow. (2019). *Stack Overflow Developer Survey 2019*. Stack Overflow.
https://insights.stackoverflow.com/survey/2019/?utm_source=social-share&utm_medium=social&utm_campaign=dev-survey-2019
- Tanomwong, N., & Jaikaeo, C. (2018). Adaptive middleware for costly data generation over MQTT. *2018 5th International Conference on Business and Industrial Research (ICBIR)*, 63-68.
<https://doi.org/10.1109/ICBIR.2018.8391167>
- Tantitharanukul, N., Osathanunkul, K., Hantrakul, K., Pramokchon, P., & Khoenkaw, P. (2017). MQTT-Topics Management System for sharing of Open Data. *2017 International Conference on Digital Arts, Media and Technology (ICDAMT)*, 62-65.
<https://doi.org/10.1109/ICDAMT.2017.7904935>
- Valderrama, J., & Brea, E. (2018). (PDF) *ESP8266: Un microcontrolador para el Internet de las Cosas*.
<http://www.ing.ucv.ve/jifi2018/documentos/electronica/ERT-006.pdf>

Villasís-Keever, M. Á., & Miranda-Navales, M. G. (2016). El protocolo de investigación IV: Las variables de estudio. *Revista Alergia México*, 63(3), 303.

<https://doi.org/10.29262/ram.v63i3.199>

Yadav, N., Rajpoot, D. S., & Dhakad, S. K. (2019). LARAVEL: A PHP Framework for E-Commerce Website. *2019 Fifth International Conference on Image Information Processing (ICIIP)*, 503-

508. <https://doi.org/10.1109/ICIIP47207.2019.8985771>