



FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

IMPLEMENTACIÓN DE UN SISTEMA DE MONITOREO DEL CONSUMO
DE ACEITE DOMÉSTICO UTILIZADO POR ESTABLECIMIENTOS DE
BAJO IMPACTO AMBIENTAL A TRAVÉS DE UNA RED WIRELESS

AUTOR

FRANCISCOALEJANDRO DÍAZ CALVA

AÑO

2020



FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

IMPLEMENTACIÓN DE UN SISTEMA DE MONITOREO DEL CONSUMO DE
ACEITE DOMÉSTICO UTILIZADO POR ESTABLECIMIENTOS DE BAJO
IMPACTO AMBIENTAL A TRAVÉS DE UNA RED WIRELESS

Trabajo de titulación presentado en conformidad con los requisitos
establecidos para optar por el título de Ingeniero en Redes y
Telecomunicaciones

Profesor Guía

MSc. Iván Ricardo Sánchez Salazar

Autor

Francisco Alejandro Díaz Calva

Año

2020

DECLARACIÓN DEL PROFESOR GUÍA

Declaro haber dirigido el trabajo, “Implementación de un sistema de monitoreo del consumo de aceite doméstico utilizado por establecimientos de bajo impacto ambiental a través de una red Wireless”, a través de reuniones periódicas con el estudiante Francisco Alejandro Díaz Calva, en el semestre 202020, orientando sus conocimientos y competencias para un eficiente desarrollo del tema escogido y dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación.



Iván Ricardo Salazar Sánchez

Magister en Calidad, Seguridad y Ambiente

CI. 180345614-2

DECLARACIÓN DEL PROFESOR CORRECTOR

Declaro haber revisado este trabajo, "Implementación de un sistema de monitoreo del consumo de aceite doméstico utilizado por establecimientos de bajo impacto ambiental a través de una red Wireless, a través de reuniones periódicas con el estudiante Francisco Alejandro Díaz Calva, en el semestre 202020, orientando sus conocimientos y competencias para un eficiente desarrollo del tema escogido y dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación.



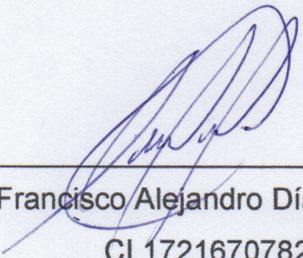
Nathaly Verónica Orozco Garzón

Doctora en Ingeniería Eléctrica en el Área de Telecomunicaciones y
Telemática

CI. 172093858-6

DECLARACION DE AUTORIA DEL ESTUDIANTE

"Declaro que este trabajo es original, de mi autoría, que se han citado todas las fuentes correspondientes y que en que su ejecución se respetaron todas las disposiciones legales que protegen los derechos de autor vigentes"



Francisco Alejandro Díaz Calva

CI.1721670782

AGRADECIMIENTOS

Agradezco a mi familia quienes son las personas más importantes en mi vida, a Dios por brindarme salud y poder culminar esta etapa de mi vida. Quiero agradecer a mis docentes quienes con sus conocimientos me han ayudado a formarme profesionalmente. Finalmente quiero agradecer a todas las personas que siempre han estado a mi lado brindándome su apoyo.

DEDICATORIA

Este trabajo va dedicado para mi familia que siempre ha estado a mi lado brindándome su apoyo y fortaleza. A mi madre en especial que siempre ha luchado por verme cumplir esta meta profesional, a ti Ani Elizabeth que fuiste la persona que con tu apoyo me impulsaste para comenzar con este sueño que en algún momento lo vi lejano. A mis amigos que cuando pasé por momentos difíciles me sostuvieron y no permitieron que renuncié a esta meta que estoy cumpliendo.

RESUMEN

La contaminación ambiental puede ocasionar graves daños al medio ambiente en el que vivimos. En la actualidad el uso inadecuado de desechos como residuos de aceite vegetal que son vertidos en cañerías públicas o domésticas pueden ocasionar taponamientos en las mismas y generar contaminación en el agua, así como gastos elevados de reparación; por tal motivo se plantea una solución que será de gran ayuda para generar una recolección responsable de los residuos de aceite vegetal producto de elaboración de frituras que producen los restaurantes y locales de comida rápida. Por este motivo se plantea la implementación de un sensor que se alojará en los contenedores autorizados para la recolección del aceite usado, el mismo que mediante pulsos ultrasónicos enviará datos acerca de la distancia del líquido dentro del contenedor en relación al sensor, para luego mediante un módulo Wireless enviar hacia una base de datos que estará asociada a una aplicación web que será monitoreada por los gestores del sistema y por el cliente, así también una aplicación móvil la cual será instalada en un smartphone Android del cliente para la visualización en vivo del estado del contenedor. Gracias a la visualización de los datos se podrá enviar una notificación tanto del cliente al gestor como viceversa y programar una cita de recolección oportuna. Por otro lado, la aplicación de este sistema permitirá el envío de notificaciones mediante correo electrónico a los usuarios y técnicos del sistema con el fin de mantener una comunicación bidireccional y coordinar la recolección de los residuos de aceite vegetal usado. Finalmente, gracias a los datos que se obtienen desde los sensores se puede establecer un análisis estadístico sobre la recolección de aceite vegetal usado de los contenedores y determinar el impacto ambiental que los mismos pueden provocar.

Palabras claves: Contaminación ambiental, sensor, red Wireless, contenedor, aplicación web y móvil, gestor.

ABSTRACT

Environmental pollution can cause serious damage to the environment in which we live. At present the inappropriate use of waste such as vegetable oil residues that are dumped in public or domestic pipes can cause plugging in them and generate contamination in the water, as well as high repair costs; For this reason, a solution is proposed that will be of great help to generate a responsible collection of waste vegetable oil produced from the preparation of fried foods produced by restaurants and fast food outlets. For this reason, the implementation of a sensor is proposed that will be housed in the authorized containers for the collection of used oil, the same one that will send data about the distance of the liquid inside the container in relation to the sensor by means of ultrasonic pulses, and then through a Wireless module send to a database that will be associated with a web application that will be monitored by the system managers and by the client, as well as a mobile application which will be installed on the client's Android smartphone for live status viewing of the container. Thanks to the visualization of the data, it will be possible to send a notification both from the client to the manager and vice versa and schedule a timely collection appointment. On the other hand, the application of this system will allow the sending of notifications by email to the users and technicians of the system in order to maintain two-way communication and coordinate the collection of used vegetable oil waste. Finally, thanks to the data obtained from the sensors, a statistical analysis can be established on the collection of used vegetable oil from the containers and determine the environmental impact that they may cause.

Keywords: Environmental pollution, sensor, Wireless network, container, web and mobile application, manager.

ÍNDICE

INTRODUCCIÓN	1
Antecedentes.....	1
Alcance	2
Justificación	3
Objetivos	3
Objetivo General.....	3
Objetivos Específicos	3
1. CONCEPTOS BÁSICOS	4
1.1. Medio Ambiente	4
1.2. Contaminación Ambiental.....	4
1.3. Contaminación Ambiental: Aceite Vegetal Usado	5
1.4. Gestores Ambientales	6
1.5. Contenedores	9
1.6. ASP.NET.....	11
1.7. C#.....	11
1.8. Visual Studio 2019.....	12
2. DETERMINACIÓN DE VARIABLES DEL SISTEMA DE MONITOREO.....	13
2.1. Variable de entrada	14
2.1.1. Variables de entrada del gestor	14
2.2. Variable de salida	15
2.1.2. Variables de salida del gestor.....	16
3. DISEÑO DEL PROTOTIPO PARA MONITOREO DEL USO DE ACEITE VEGETAL MEDIANTE UNA RED WIRELESS	16
3.1. Análisis de Requerimientos.....	16
3.1.1. Requerimientos de extracción y almacenamiento de datos	17
3.1.2. Requerimientos del servidor	19
3.1.3. Requerimientos de la base de datos.....	19

3.2. Elección de hardware y software del prototipo	19
3.2.1. Elección del sensor.....	20
3.2.2. Elección servidor base de datos	23
3.2.3. Elección del servidor de Base de datos	25
3.2.4. Elección del servidor de correo.....	28
3.2.5. Elección de <i>framework</i> para aplicación web	30
3.3. Requerimientos etapa de aplicación y monitoreo.....	33
3.3.1. Análisis de requerimientos aplicación web.....	33
3.3.2. Diagrama de casos de uso	35
3.3.3. Análisis de requerimientos aplicación móvil.....	39
3.4. Direccionamiento de red.....	45
4. IMPLEMENTACIÓN DEL SISTEMA DE MONITOREO DE NIVEL DE ACEITE USADO	45
4.1. Implementación del direccionamiento de red IPv4	45
4.1.1. Red interna	46
4.1.2. Configuración de red del servidor (Raspberry Pi 3 B+)	48
4.2. Implementación de fase para almacenamiento de datos	56
4.3. Implementación de la Aplicación Web	60
4.3.1. Creación de Aplicativo web en Visual Studio	60
4.4. Servidor DNS	65
4.5. Implementación de la Aplicación Móvil	66
4.5.1. Instalación de Android Studio en Windows 10	66
4.5.2. Desarrollo de la aplicación móvil del proyecto	68
4.5.3. Vistas aplicación móvil.....	69
5. PRUEBAS DE FUNCIONAMIENTO DEL SISTEMA.....	72
5.1. Funcionamiento del servidor	72
5.2. Base de Datos	74
5.3. Pruebas de funcionamiento fase de monitoreo y presentación de datos	76
5.3.1. Pruebas de verificación del aplicativo web.....	76
5.3.2. Pruebas de verificación del aplicativo móvil en Android.....	83
6. CONCLUSIONES Y RECOMENDACIONES	87

6.1. Conclusiones	87
6.2. Recomendaciones	89
REFERENCIAS	91
ANEXOS	94

ÍNDICE DE FIGURAS

Figura 1. Logotipo empresa ARC.	7
Figura 2. Logotipo empresa PROGEDE OIL-NATURA.	8
Figura 3. Contenedor homologado recolección aceite vegetal usado.	10
Figura 4. Contenedor almacenamiento masivo aceite vegetal usado.	10
Figura 5. Logotipo ASP.NET.	11
Figura 6. Inicio de Visual Studio 2019.	13
Figura 7. Diagrama variables proyecto.	14
Figura 8. Funcionamiento de proyecto Bi-Oil.	17
Figura 9. Diagrama extracción y almacenamiento de datos.	18
Figura 10. Requerimientos aplicativo web y móvil.	33
Figura 11. Diagrama caso de uso aplicación web.	36
Figura 12. Diagrama clases para aplicativo web.	37
Figura 13. Vista inicial del aplicativo web.	38
Figura 14. Inicio de sesión.	38
Figura 15. Registro de usuarios.	39
Figura 16. Informe de resultados.	39
Figura 17. Diagrama casos de uso.	41
Figura 18. Diagrama clases aplicación móvil.	42
Figura 19. Inicio aplicación.	43
Figura 20. Informe estado contenedor.	44
Figura 21. Notificación mediante correo electrónico.	44
Figura 22. Diagrama de red del proyecto.	46
Figura 23. Modem inalámbrico Calix 813Gv2-1.	48
Figura 24. Configuración interfaz Wireless Raspberry Pi 3 B+.	48
Figura 25. Raspberry Pi Modelo 3 B+.	49
Figura 26. Escritorio sistema operativo Raspbian.	50
Figura 27. Actualización paquetes sistema operativo.	51
Figura 28. Archivos de programa para ejecución de proyecto.	52
Figura 29. Importación de librerías para ejecución de programas.	52
Figura 30. Comunicación establecida entre la Base de datos y la Raspberry Pi.	53

Figura 31. Establecimiento de los puertos de comunicación y definición de variables.	53
Figura 32. Ejecución de la fórmula para obtener la distancia que existe entre el líquido y el sensor.	54
Figura 33. Almacenamiento de valores en base de datos.	54
Figura 34. Presentación de datos internos y limpieza de pantalla luego de ejecución.....	55
Figura 35. Formato configuración crontab.....	56
Figura 36. Archivo Crontab proyecto de Titulación.	56
Figura 37. Portal de Azure para crear base de datos.	57
Figura 38. Permiso de IP para acceso desde internet.	58
Figura 39. Acceso a SQL Server en Azure.....	58
Figura 40. Conexión base de datos en Azure.....	59
Figura 41. Creación Base de Datos BI-OIL.	59
Figura 42. Script para creación de tabla para proyecto.	60
Figura 43. Visualización del proyecto para aplicación web.....	61
Figura 44. Librerías importadas para desarrollo de proyecto.....	61
Figura 45. Código para vista de creación de usuarios.....	62
Figura 46. Lanzamiento del proyecto de manera local.	63
Figura 47. Página principal con usuario administrador.	63
Figura 48. Vista de reportería.....	64
Figura 49. Vista usuarios.....	64
Figura 50. Vista edición de usuario.	65
Figura 51. Administración de roles.	65
Figura 52. Nombres de servidores otorgados a dominio.	66
Figura 53. Mensaje de bienvenida instalación Android Studio.....	67
Figura 54. Ubicación de instalación Android Studio.	67
Figura 55. Inicio Android Studio, creación de proyecto.....	68
Figura 56. Conexión con Base de Datos SQL Server en Azure.	68
Figura 57. Código sincronización con correo electrónico para envío de notificaciones.	69
Figura 58. Vista inicial aplicación.	70

Figura 59. Vista Recuperación de contraseña.....	70
Figura 60. Vista generar gráfica o generar notificación de correo electrónico.	71
Figura 61. Vista correspondiente a gráfica de monitoreo.	71
Figura 62. Vista correspondiente a envío de notificación.	72
Figura 63. Raspberry Pi Pines Entrada/Salida GPIO.....	74
Figura 64. Comunicación entre sensor y base de datos.....	74
Figura 65. Base datos SQL Server.....	75
Figura 66. Roles del aplicativo web.....	77
Figura 67. Vista usuarios con perfil de administrador.....	78
Figura 68. Vista página con perfil de Técnico.....	78
Figura 69. Creación de usuario y asignación de roles perfil administrador.	79
Figura 70. Informe de aplicación web.....	80
Figura 71. Lectura de datos de contenedor desde cero.	80
Figura 72. Notificación mediante correo electrónico a usuario.	81
Figura 73. Notificación a Administrador mediante correo electrónico.	81
Figura 74. Inicio de sesión aplicación web.	82
Figura 75. Vista aplicación web con opción de cierre de sesión.	82
Figura 76. Inicio de Sesión Aplicación móvil.	84
Figura 77. Sección para visualizar informe o enviar notificación.	84
Figura 78. Presentación de informe en tiempo real de aplicación móvil.	85
Figura 79. Vista aplicación móvil para envío de notificaciones al sistema.	85
Figura 80. Notificación recibida en correo electrónico de Administrador.....	86

ÍNDICE TABLAS

Tabla 1 Comparación Sensores.	22
Tabla 2 Comparativo base de datos.	25
Tabla 3 Comparativo de servidores.	28
Tabla 4 Comparación servidor de correo.	30
Tabla 5 Comparación de Framework a elegir.	32
Tabla 6 Listado requerimientos funcionales.	34
Tabla 7 Listado de requerimientos no funcionales.	35
Tabla 8 Requerimiento funcional aplicación móvil.	40
Tabla 9 Requerimiento no funcional aplicación móvil.	40
Tabla 10 Direccionamiento de red de hardware utilizado.	45
Tabla 11 Direccionamiento de red interno.	47
Tabla 12 Requerimientos del servidor.	73
Tabla 13 Requerimientos servidor Base de Datos.	75
Tabla 14 Requerimientos funcionales aplicativo web.	76
Tabla 15 Requerimiento funcional Aplicativo Móvil.	83

INTRODUCCIÓN

Antecedentes

Con el aumento poblacional que se registra en las ciudades principales del país, de igual manera el aumento de consumo que generan las mismas, cada día se puede observar que se generan mayores volúmenes de desechos orgánicos e inorgánicos, la mayoría de estos peligrosos, que, cuando no existe un procesamiento adecuado, estos se convierten en residuos que son los generadores de las principales fuentes de contaminación en cada ciudad. Es así que, con el rápido proceso de urbanización experimentado también en todo el continente americano a ha requerido una inversión mayor y los proyectos de saneamiento que son atendidos por cada gobierno municipal tengan un costo elevado.

En Ecuador el tratamiento y la manera de procesar los desechos sólidos se encuentra a cargo de los Gobiernos Autónomos Descentralizados -GAD-; estas entidades son las encargadas, cuya planificación de gestión ambiental fomentan “el tratamiento de manera técnica con respecto a la disposición final de los desechos y la manera para desarrollar de alternativas de cómo sacar provecho de los residuos” (CONSULTORA, 2014) así poder aportar con la reducción del impacto ambiental que estos desperdicios ocasionan. Sin embargo, en lo que respecta al manejo de desperdicios líquidos como por ejemplo residuos de hidrocarburos, aceites, agua con detergente, aguas servidas o pesticidas no se manejan de igual manera, esto es debido a que no existen muchos proyectos dentro de los cabildos municipales, empresas privadas y los mismos ciudadanos para que estos tengan un adecuado tratamiento. “Algo importante que destacar es que los márgenes reportados de contaminación en el agua, ríos y otras fuentes de agua que se debe a distintos residuos, se puede interpretar con índices altos de contaminación por como lo estable el estándar internacional en algunas ciudades del país, lo cual afecta de manera evidente en la salud de ciudadanía y del medio ambiente” (Ley de Gestión Ambiental, 2004).

Es así como, unos de los principales contaminantes que encontramos es la presencia de residuos de aceites vegetales dentro del sistema de alcantarillado que posee la ciudad Quito; en lo que se conoce las estadísticas y datos que se presentan por medio de la Secretaría del Ambiente del municipio dan a conocer que: “los restos de aceite vegetal usado son considerados como la causa principal de contaminación de aguas en el perímetro urbano, de igual manera mediante estudios se estima que con los residuos de un litro de aceite puede llegar a afectar a 1000 litros de agua limpia” (Secretaria de Ambiente Quito, 2015). Así también, “se calcula cada persona puede desechar la cantidad de 2,76 litro durante un año. Lo cual significa que en la ciudad de Quito se produce alrededor de siete millones de litros de desperdicios de aceite vegetal que son desechados cada año”. (Secretaria de Ambiente Quito, 2015).

Por lo cual ante esta necesidad y la falta de un dispositivo que pueda ayudar a disminuir el impacto de contaminación producido por los aceites vegetales, se va a realizar el diseño y prototipo de un dispositivo que sea capaz de recolectar el aceite que es desechado por las cañerías y realice un monitoreo del consumo de aceite doméstico, el mismo que enviará detalles del comportamiento del líquido dentro del contenedor y se programe la recolección del envase.

Alcance

Diseñar un prototipo de un sistema automatizado que permita monitorear y determinar el nivel de aceite vegetal de cocina usado que se deposita en un contenedor de un establecimiento comercial, Para el efecto, el contenedor con el sistema estarán monitoreados remotamente gracias a la ayuda de sensores que midan la capacidad del contenedor y mediante el uso de un módulo WIRELESS, que será el encargado de transmitir la información que será monitoreada y almacenada en un servidor y así enviará una notificación tanto a los administradores a través de una plataforma web como a los usuarios a través de un aplicativo móvil para coordinar la recolección del líquido.

Justificación

El aceite vegetal puede ser utilizado para la producción de biodiesel, combustible que se está utilizando con mayor frecuencia con el propósito de reemplazar los combustibles fósiles. Recolectar todo el aceite de consumo de los hogares de la ciudad, centros educativos, restaurantes, hoteles, entre otros, permitirá reducir la contaminación de las aguas grises residuales y producir una cantidad de biocombustible considerable. La utilización de este biocombustible puede resultar de favorable en lo que respecta a la parte energética y mayor aún que es amigable con el medio ambiente, ya que este biocombustible emite en menor cantidad gases nocivos que afectan la atmósfera.

Es así, que con el avance de la tecnología y las facilidades que ésta brinda, se cuenta con servicios automatizados y controlados por una aplicación móvil.

Por otro lado, la Universidad de las Américas realiza campañas de concientización sobre el impacto ambiental que generan los desechos sólidos y líquidos, y mediante la realización de este proyecto podrá contribuir a la disminución del impacto de contaminación ambiental en la ciudad.

OBJETIVOS

Objetivo General

Implementar un sistema de monitoreo del consumo de aceite vegetal doméstico utilizado por establecimientos de bajo impacto ambiental a través de una red Wireless.

Objetivos Específicos

- Identificar las necesidades que se tienen frente a la situación actual del uso del aceite vegetal y el grado de impacto ambiental que generan por el deshecho erróneo de los mismos.
- Determinar las variables de entrada y salida a ser consideradas dentro del desarrollo del sistema de monitoreo.

- Diseñar un contenedor que recolecte aceite usado de cocina y que permita a través de sensores y módulos de comunicación Wireless enviar la información a una plataforma Web.
- Implementar el sistema de monitoreo de nivel de aceite domestico usado en pequeños negocios, considerando el diseño previamente establecido.
- Realizar pruebas del prototipo en diferentes escenarios en los cuales se pueda obtener datos reales para determinar datos estadísticos y realizar mejoras de este.

1. CONCEPTOS BÁSICOS

1.1. Medio Ambiente

El medio ambiente es el lugar o hábitat donde residen todos los seres vivos. El Ministerio del Ambiente lo define como “El conjunto de componentes orgánicos e inorgánicos, fenómenos biológicos, físicos y químicos que son de aporte para la vida, el desarrollo y la evolución de los elementos vivos”. (Moyococha, 2016).

1.2. Contaminación Ambiental

La contaminación que se produce en el medio ambiente es uno de los principales inconvenientes que ocasiona daños a la salud en personas, animales y plantas, es por esto que en la actualidad se emplean métodos para revertir el daño ocasionado por la mala gestión de los desechos de aceite vegetal usado a través de estudios, gracias a esto se puede obtener resultados que evidencian malas prácticas que desarrollan varias empresas locales así como a nivel mundial; estas son causantes de generar grandes cantidades de residuos lo cual alcanza altos porcentajes de peligro para el medio ambiente así también para la sociedad.

De igual manera el Ministerio de Ambiente señala que es el “la contaminación ambiental es el proceso en el cual se altera un ecosistema debido a la introducción de sustancias y/o elementos en el ambiente por parte del hombre lo cual puede llegar a perjudicar a la salud, ocasionar graves perjuicios en los

entornos ecológicos y en organismos vivientes, puede existir deterioro en las características y estructura del ambiente, así como reducir la posibilidad del aprovechamiento de los recursos naturales”. (Moyococha, 2016)

1.3. Contaminación Ambiental: Aceite Vegetal Usado

Los residuos de aceites vegetales son catalogados como un potencial peligro para el medio ambiente debido a los componentes que contiene, ya que al tratarse de un líquido se puede esparcir de una manera fácil en grandes extensiones de terreno, así como también en el agua, lo cual no permite el paso del oxígeno debido a una película que se forma en la superficie. Aunque se conocen los daños causados por el mal manejo del aceite vegetal usado, no se ha hecho una cuantificación de daños producidos por estos en Ecuador.

Los aceites de cocina son un producto muy utilizado hoy por la sociedad, donde sus ingredientes son producidos mediante la actividad agrícola, que una vez que se encuentran listos para ser procesados en su elaboración, luego son distribuidos y finalmente vendidos, las personas en sus hogares por lo general utilizan este aceite en la elaboración de frituras. Por lo que se denomina fritura al proceso culinario mediante el cual los alimentos se someten a altas temperaturas entre 150-200 °C (González Canal & González Ubierna, 2015) utilizando las grasas o aceites como medio de transmisión de calor.

De acuerdo con (Navas, 2005): “en este proceso el aceite vegetal experimenta una gran cantidad de cambios físicos, químicos y sensoriales” (p. 14), los cuales se originan por la humedad que existe en los alimentos, el aire que conlleva oxígeno y las altas temperaturas entre los 160- 200 °C. Con estas temperaturas se puede ocasionar oxidación, reacciones como hidrólisis y polimerización que cambian las características de los aceites (Navas, 2005), esto ocasiona una degradación en los aceites a través de su tiempo de uso, lo que hace necesario que solo se pueda utilizar por una ocasión, ya que posterior a esto puede ocasionar perjuicios para la salud humana.

La mala gestión de los aceites residuales puede visualizarse como un problema ambiental y de salud (Sarno & Iuliano, 2019). Uno de los principales conflictos ambientales que representan los aceites es el deshacerse de ellos por medio de los drenajes, esto puede generar olores desagradables y contribuir con el incremento de microorganismos dañinos en la salud humana, así como taponamiento de los drenajes lo cual ocasionaría gastos extras en los hogares. Con respecto al agua, los aceites vegetales usados generan el agotamiento del oxígeno, causando la destrucción de flora y fauna en ecosistemas acuáticos; también pueden ocasionar obstrucción de tuberías y, en su etapa de depuración, es necesario una cantidad considerable de energía. (Greenhabit, 2013)

Finalmente, el proyecto propuesto pretende apoyar con las ordenanzas municipales que se refieren a la mejora Ambiental de la ciudad. Se gestionará de mejor manera mediante el uso tecnológico con un manejo integro de los residuos de aceite vegetal usado, alertando tanto a los administradores del sistema, así como al usuario final.

1.4. Gestores Ambientales

Un gestor ambiental es un ente especializado en la recolección de todo tipo de desechos ambientales. Recicla y recupera los mismos que son fuentes altamente contaminantes.

En el país existen un número bajo de gestores dedicados a la recolección de aceite usado vegetal o animal desechado por establecimientos de cocina.

En la ordenanza municipal dispuesta en 2007 se establece que los residuos que pertenecen a alimentos, aceite vegetal usados o grasas no pueden ser desechados en el sistema público de alcantarillas de la ciudad. Sino que estos residuos deben ser almacenados en contenedores sellados con tapa para luego entregarse a los gestores autorizados por el Distrito Metropolitano de Quito. (RESOLUCIÓN N° 001- DMMA-2007).

A continuación, se puede conocer dos de los principales gestores de recolección en la ciudad de Quito:

- **ARC & PIEPER RECOLECTORA Y EXPORTADORA S.A.**



Figura 1. Logotipo empresa ARC.

Tomado de: (ARC, 2020)

ARC, es una empresa ecuatoriana que se encuentra enfocada a recolectar los residuos de aceite vegetal usado para la preparación de frituras para posteriormente enviarlos hacia Europa para que sean transformados en biodiesel, lo cual convierte los residuos en una fuente de energía.

Esta empresa cuenta con *Licencia Ambiental Categoría II* la misma que cuenta con el registro No. *00353-13-2015-FA-CGZ4-DPAM-MAE* que le otorga la facultad para poder ejecutar dicha actividad, de igual manera cumple con la norma ambiental que se encuentra vigente. El Ministerio del Medio Ambiente es el ente encargado de emitir estas licencias.

La empresa cuenta al momento con más de 1500 clientes entre los cuales se mantienen locales de comida, sector hotelero y en sí toda industria que tiene relación con elaboración de alimentos varias ciudades del país, el 40% de los clientes los mantiene en la ciudad de Quito.

Por la recolección del residuo ARC paga por cada contenedor de 20 litros de y mínimo se debe poseer dos contenedores llenos, es decir 40 litros. La empresa ofrece transporte hasta la dirección que se indique para la recolección, ellos se encargan de llevar el contenedor vacío de 20 litros para reponer las que se llevan

en caso de que se necesiten. Cuentan con varias políticas para la recolección en lo que menciona que en los establecimientos.

Finalmente, la idea de negocio que ellos tienen produce un impacto que beneficia al medioambiente: ya que evita que los residuos de las frituras con aceite contaminen el agua y se pueda producir un bloqueo en el sistema de alcantarillas de la ciudad; así también la manera de evitar que estos residuos puedan ser comercializados de modo ilegal y los mismos puedan ser adulterados y vendidos como nuevos. Con esto también se evita que sean mezclados con alimentos para mascotas o animales en general u otras vituallas que puedan presentar un riesgo inminente en lo que respecta al estado de salud del consumidor humano. Al final el aceite es exportado para la elaboración de Biodiesel como se lo mencionó anteriormente. (ARC, 2020)

- **PROGEDE OIL NATURA**



Figura 2. Logotipo empresa PROGEDE OIL-NATURA.

Tomado de (OIL-NATURA, 2020)

PROGEDE se encuentra enfocado en concientizar a las empresas de la industria alimenticia y que generen residuos de aceite vegetal en cómo afecta al medio ambiente y a las personas la mala gestión de este.

La empresa PROGEDE de igual manera cuenta con el número de licencia ambiental MAE-SUIA-RA-DPAPCH-2017-236137, que es otorgado por el Ministerio de Medio Ambiente Ecuatoriano (MAE), y se encuentran autorizados a operar a nivel nacional.

Una vez que la empresa PROGEDE recoge el residuo de los establecimientos, ellos entregan un CERTIFICADO DE RECOLECCIÓN que acredita dicho servicio. Dicho certificado avala a los establecimientos que están cumpliendo la normativa publicada por el Ministerio de Ambiente Ecuatoriano y la Secretaría de Ambiente.

En lo que respecta la recolección del fluido se lo realiza dependiendo de las necesidades del negocio, es decir cada establecimiento se comunicó con la empresa y genera una cita para la recolección. Las agendas de recolección se generan con un plazo de 24 horas desde el aviso. (OIL-NATURA, 2020)

1.5. Contenedores

Un contenedor se define un recipiente, caja o transporte de carga que es construido para otorgar una facilidad en el traslado de objetos, como unidad de carga, se lo puede realizar por medio de cualquier transporte que posea una resistencia para poder realizar la acción de manera repetitiva para que sea llenado y vaciado con seguridad y facilidad, que posea accesorios que permitan que su manejo sea óptimo y rápido en lo que respecta a trasbordo, carga y descarga, de igual manera que cumpla las normas respectivas para el uso que se le vaya a dar. (Aduanas, 2020)

Para realizar la recolección de los aceites se podría utilizar un contenedor o recipiente con capacidad de 10 Gal., el mismo que se encuentra homologado con la Secretaría de Ambiente de la ciudad de Quito.

El contenedor presentará las siguientes características:



Figura 3. Contenedor homologado recolección aceite vegetal usado.

Tomado de: (Secretaria de Ambiente Quito, 2015)

Así mismo, se tiene diferentes opciones para realizar la recolección de este dependiendo del espacio a ser colocado.



Figura 4. Contenedor almacenamiento masivo aceite vegetal usado.

Tomado de: (Secretaria de Ambiente Quito, 2015)

Este contenedor tiene una altura de 90 cm que podría ser considerado para una mejor efectividad del sensor a utilizar.

1.6. ASP.NET

Es una herramienta gratuita que permite la creación de sitios y aplicativos webs mediante diferentes entornos como son HTML, CSS y JavaScript. También puede desarrollar API's web mediante el uso de Web Sockets que es una tecnología basada en tiempo real.

Dispone de tres marcos diferentes para la creación de aplicativos webs. Estos marcos proporcionan estabilidad y confiabilidad, sea cual sea el marco que se elija, se pueden crear excelentes aplicaciones web con estas opciones. (Microsoft, Información general de ASP.NET, 2019). En la Figura 5, se puede observar el logotipo correspondiente al *framework* ASP .NET.



Figura 5. Logotipo ASP.NET.

Tomado de: (Microsoft, Información general de ASP.NET, 2019)

1.7. C#

Lenguaje de programación que es orientado a objetos, posee seguridad de tipos. Este lenguaje de programación da la oportunidad que los programadores o desarrolladores puedan crear una gran variedad de aplicativos, que este

lenguaje de programación sea seguro, sólido y para su ejecución lo realicen mediante el Framework .NET.

A C# se lo puede utilizar para desarrollar aplicaciones nativas de Windows, así también como servicios web basados en XML, aplicaciones basadas entre cliente y servidor, aplicaciones específicas de base de datos, y muchos más usos en torno a programación. Mediante Visual C# se puede obtener un editor de código avanzado, la interfaz de desarrollo es amigable con el usuario, se puede depurar los programas y posee muchas otras que permite que los desarrolladores puedan realizar sus aplicativos. (Microsoft, Introducción al lenguaje C# y .NET Framework, 2015)

1.8. Visual Studio 2019

Es un software para programación propiedad de Microsoft el cual permite el desarrollo de aplicaciones multiplataforma.

El aplicativo es compatible con muchos de los lenguajes de programación que se manejan en la actualidad, los mismos que pueden ser editados y depurados directamente dentro de su editor de código. Este software se encuentra disponible para las plataformas de Windows y Mac, es gratis para estudiantes y usuarios individuales. (Pascual, 2019)

Este software es una de las herramientas más utilizadas hoy en día para el desarrollo de aplicativos web y móvil es así que su uso viene siendo impartido desde los centros educativos de secundaria lo que ayuda a que su uso sea totalmente amigable en todas sus funciones.

En la Figura 6, se puede observar el inicio de la aplicación Visual Studio 2019 la cual nos permitirá escribir la codificación para el aplicativo web.

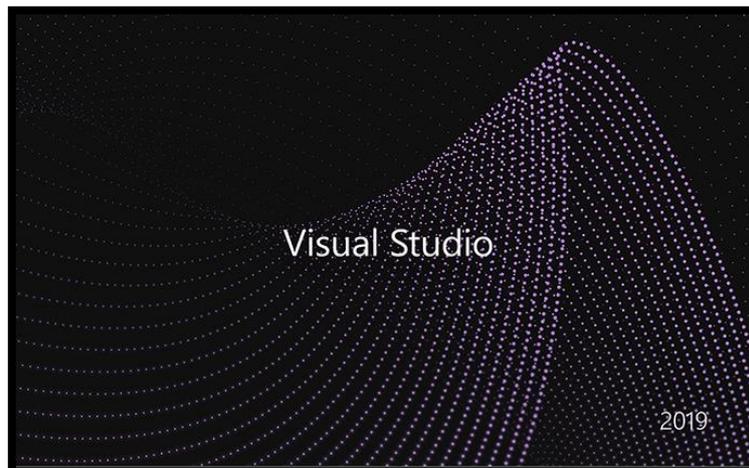


Figura 6. Inicio de Visual Studio 2019.

Tomado de: (Microsoft, 2019)

2. DETERMINACIÓN DE VARIABLES DEL SISTEMA DE MONITOREO

En este capítulo se va a realizar un análisis de las variables de entrada y salida que va a poseer el sistema, las mismas que serán necesarias para la recepción de los datos que se obtienen desde el sensor, así como las variables necesarias para la etapa de presentación de datos en el aplicativo web y móvil. Las variables de entrada serán todos los requerimientos de hardware y software necesarios para la obtención de los datos del estado del líquido dentro del contenedor, en cambio las variables de salida que se obtendrán serán las que permitan la presentación visual de los datos en el sistema.

En la Figura 7, se puede observar el diagrama a utilizar para la elaboración del sistema, el mismo que muestra los componentes necesarios para el desarrollo del proyecto, se puede observar el sensor junto con el contenedor, la base de datos necesaria para la recolección y almacenamiento de datos, así como el aplicativo web y móvil donde se presentaran los resultados obtenidos desde el contenedor.

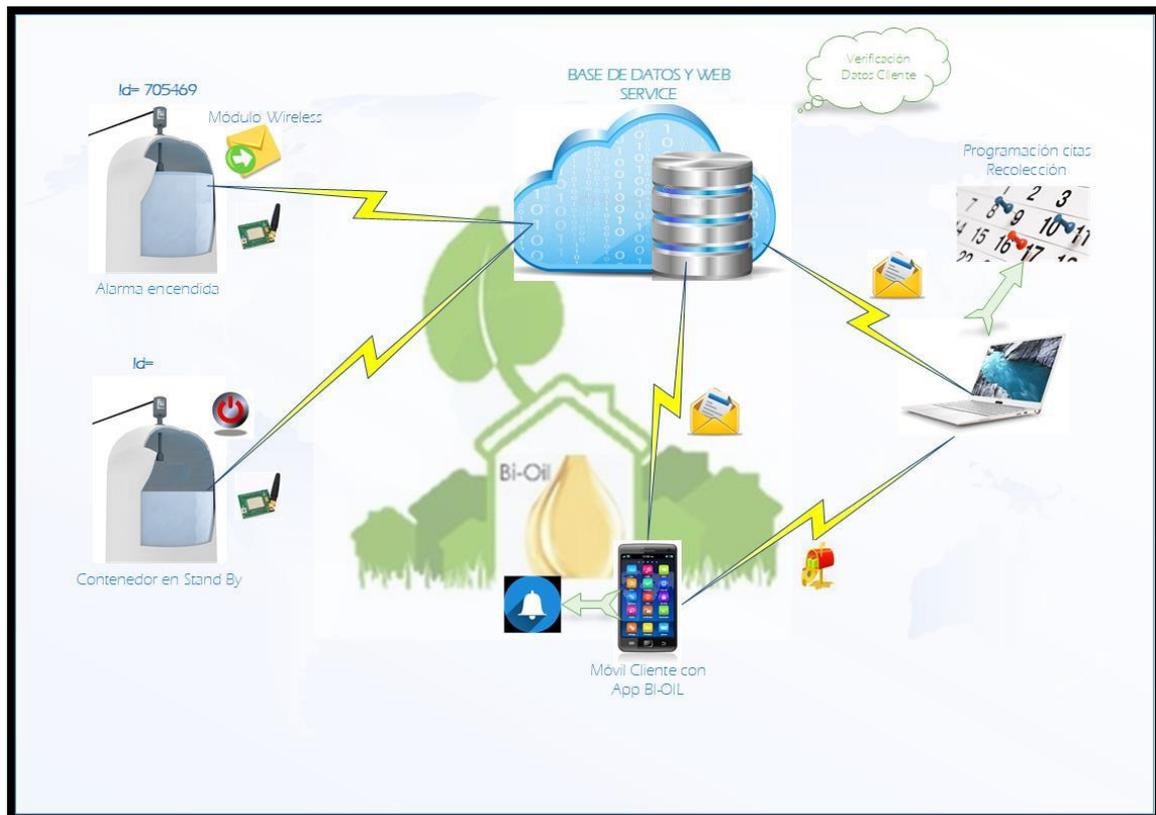


Figura 7. Diagrama variables proyecto.

2.1. Variable de entrada

Es una variable del sistema tal que una modificación de su magnitud o condición puede alterar el estado del sistema. (Sites Google, 2018)

2.1.1. Variables de entrada del gestor

Teniendo en cuenta la manera en que se realiza la recolección de los desechos de aceite vegetal usado en establecimientos que elaboran comidas rápidas así como los de bajo impacto ambiental hoy en día, que es mediante la comunicación de los establecimientos con 24 o 48 horas de anticipación, lo cual puede dificultar el almacenamiento de los residuos o a su vez que sean desechados de mala manera, ya que varios establecimientos al no tener un lugar para el almacenamiento de los desechos de aceite optan por verter este líquido por las cañerías o a su vez en los botan por el sistema de alcantarillado.

Se propone automatizar el procedimiento de recolección para que tanto gestores como establecimientos lo realicen de una manera más coordinada.

Para la creación del sistema se cuenta con las variables detalladas a continuación:

- **Página Web**

- Niveles de estado del contenedor: Nos permite conocer el estado de los contenedores con respecto al nivel del fluido que ingresa de acuerdo con la distancia que existe entre el líquido y el sensor ubicado en la parte superior, conociendo previamente que el contenedor tiene una altura de 37 centímetros.
- Ubicación: Permite conocer donde se encuentra instalado el contenedor, para su gestión, mediante un módulo GPS que se tiene instalado en la raspberry el cual nos dará las coordenadas de longitud y latitud presentadas en el aplicativo web.

- **Base de Datos**

- Datos del sensor: Se obtiene cada cierto tiempo la información referente a como se encuentra el contenedor con respecto al fluido.
- Almacenamiento: Espacio destinado para la recolección de datos provenientes del servidor el cuál se encontrará en la nube para un acceso a los mismos desde cualquier parte de internet.

- **Servidor de correo**

Recepción de correos: Permite obtener información acerca del estado del contenedor cuando este se encuentre cerca del límite de su capacidad.

2.2. Variable de salida

Es una variable del sistema cuya magnitud o condición se mide. (Sites Google, 2018)

2.1.2. Variables de salida del gestor

- **Página Web**
 - Indicadores: Muestran al gestor como tal estado de los contenedores para programar una cita de recolección y/o cambio de contenedor con fluido.
 - Gráficas: Muestra en tiempo real e histórico el comportamiento del fluido y los valores obtenidos desde el sensor.
- **Base de Datos**
 - Envío de valores: Los datos obtenidos se envían hacia la página web para la generación de indicadores y modelamiento de las gráficas para la apreciación del usuario final.
- **Servidor Correo**

Enviaré al usuario una notificación a un correo electrónico notificando el límite del contenedor y notificación para realizar cita de recolección.

3. DISEÑO DEL PROTOTIPO PARA MONITOREO DEL USO DE ACEITE VEGETAL MEDIANTE UNA RED WIRELESS

3.1. Análisis de Requerimientos

En este capítulo se presentarán las características que va a poseer el prototipo en su diseño e implementación, además de los componentes necesarios a utilizar. Se presentará también un esquema y requerimientos para el diseño del prototipo. En la Figura 8, se puede visualizar el esquema funcionamiento que debe poseer el sistema, para los cuales se van a listar una serie de requerimientos necesarios para el desarrollo de este.

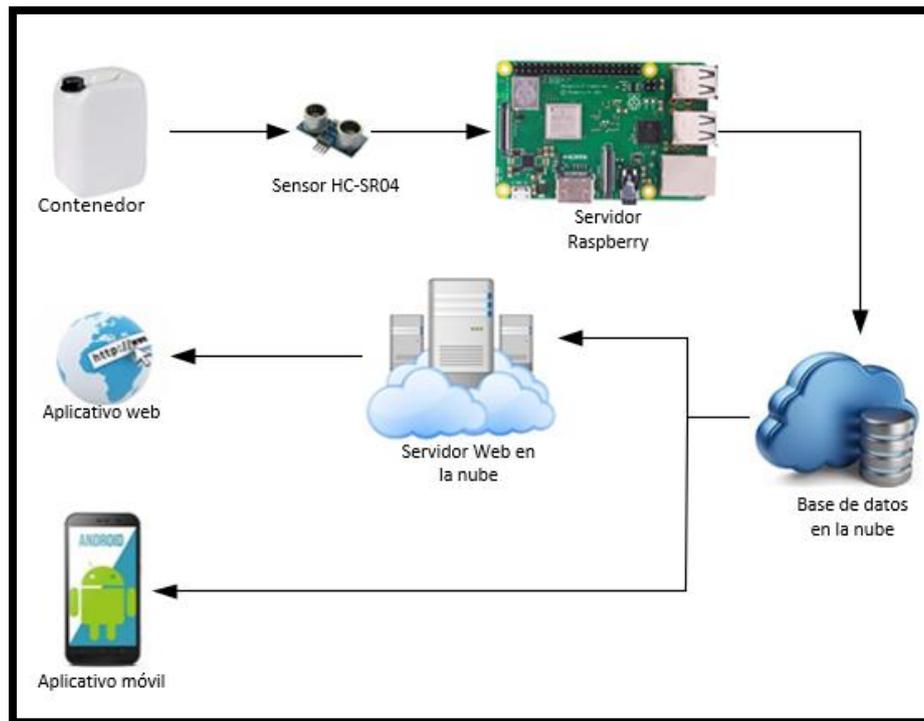


Figura 8. Funcionamiento de proyecto Bi-Oil.

Por lo cual, para un mejor entendimiento del proyecto planteado, el prototipo será analizado en dos etapas, las cuales son:

- Extracción y almacenamiento de datos.
- Monitoreo de datos y presentación de estos.

3.1.1. Requerimientos de extracción y almacenamiento de datos

Los requerimientos que va a poseer esta etapa van a constar de los siguientes componentes:

- Sensor ultrasónico (HC-SR04).
- Servidor (Raspberry)
- Servidor de base de datos.

En la Figura 9, se visualiza las partes utilizadas en esta etapa del diseño. Como es posible observar se inicia con el contenedor que va a almacenar el líquido

(residuos de aceite vegetal usado), mediante el sensor HC-SR04 se medirá la distancia entre el sensor y el líquido que será ejecutado por un script que se encuentra en el servidor (Raspberry) para su posterior envío y almacenamiento en la base de datos que se encontrará en la nube.

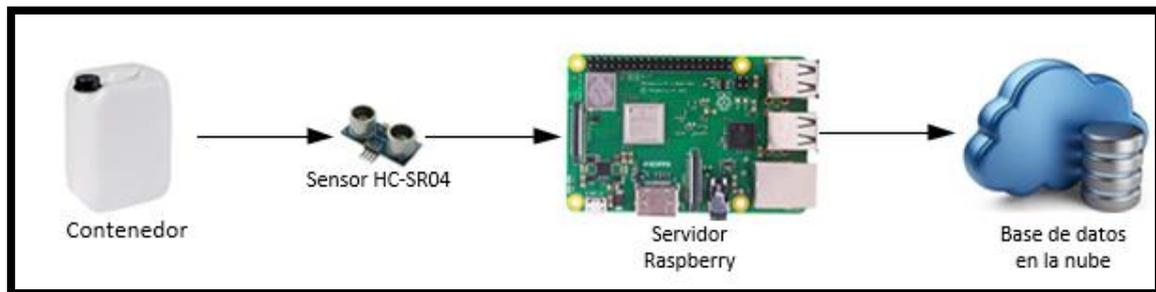


Figura 9. Diagrama extracción y almacenamiento de datos.

Para esta parte del diseño se eligieron componentes de acuerdo con el alcance del proyecto planteado y la funcionalidad que presenta cada uno para un desarrollo óptimo.

A los requerimientos se los detonarán con la letra R, y se obtendrá los siguientes:

R1: El sensor estará conectado directamente al servidor, con el fin de que los valores obtenidos sean los reales y lo más precisos posibles.

R2: Los valores que se capten mediante el sensor se almacenarán temporalmente en el servidor, los datos serán obtenidos en un período de tiempos de la siguiente manera: cada 2 minutos se ejecutará un script, así como también cada 5 minutos. Es decir, cuando el sistema se encuentre funcional se emitirán dos registros acordes a los tiempos indicados.

R3: Establecer la comunicación con el servidor

3.1.2. Requerimientos del servidor

R4: Establece la comunicación que se mantiene con el sensor para la lectura de datos que se van obteniendo en los diferentes tiempos que se plantearon anteriormente.

R5: Permitir que se establezca comunicación hacia el servidor de base de datos.

R6: Enviar los valores que se obtengan en las lecturas de ejecución del programa para almacenarlos en la base de datos.

R7: Generar diferentes registros con respecto a los tiempos planteados con el fin de poder visualizar de mejor manera los datos obtenidos durante el monitoreo.

R8: El consumo de energía es bajo y su instalación es adaptable ya que ocupa poco espacio.

R9: Se debe contar con al menos un módulo wireless para la comunicación hacia internet y la base de datos que se encontrará en la nube.

R10: Debe ser capaz de soportar la ejecución de varios scripts y/o programas.

3.1.3. Requerimientos de la base de datos

R11: Establecer conexión y comunicación con el servidor.

R12: Encargado de almacenar la información obtenida desde el servidor.

R13: Permitir que los datos obtenidos puedan ser publicados.

R14: Ser adaptable para desarrollo de aplicaciones, ya sean web o móviles.

3.2. Elección de hardware y software del prototipo

En este apartado se va a presentar los componentes físicos y software que se va a utilizar durante la implementación del prototipo, estos estarán considerados

dentro del análisis de requerimientos para las posteriores etapas, como son presentación y monitoreo.

3.2.1. Elección del sensor

Se requiere elegir un sensor que permita establecer comunicación con el servidor y a la vez enviar datos de manera fiable y que cumpla las siguientes características:

- Posea pines de entrada y salida para lectura de la información
- Debe permitir la comunicación con un servidor
- Sea conocido y confiable
- Sea fácil de configurar
- Los datos transmitidos sean garantizados

3.2.1.1. Opciones a consideración

En este apartado se presenta una serie de sensores que se tiene como candidatos a utilizar para la realización del sistema. Los cuales son los siguientes:

- Sensor ultrasónico HC-SR04
- Sensor de nivel de agua
- Sensor switch flotador para nivel agua

Sensor ultrasónico HC-SR04 (Indoware, 2013)

- Bajo consumo de energía
- Lecturas a nivel de pulsos
- Alto nivel de desempeño
- Rango de distancia: 2cm – 400cm
- Dimensiones: (45x20x15) mm

Sensor nivel de agua (DIY ROBOT KEYES, 2013)

- Bajo consumo de energía
- Contacto físico con el líquido para las lecturas
- Tipo Analógico
- Rango de distancia: 40mm
- Dimensiones: 60mm x 20mm

Sensor de nivel switch flotador agua (Gems Sensors, 2014)

- Bajo consumo de energía
- Detecta líquido dentro de un tanque
- Puede accionar una bomba, indicador, alarma u otro dispositivo

3.2.1.2. Comparación entre opciones de sensores

En la Tabla 1, se presenta una comparativa entre las opciones que se consideraron, donde se tiene énfasis en la relevancia del requerimiento, la misma que es definida entre las características que cada sensor posee, así como también si cumplen con las propiedades necesarias para la implementación.

En la misma se presentan como los sensores se ajustan al desarrollo del proyecto, esto permitirá que al momento de ejecutar el prototipo no se presenten inconvenientes de compatibilidad.

Los valores de cumplimiento se dan de acuerdo con la adaptabilidad que tiene el sensor con su respectivo funcionamiento, es así que 1 representa que el sensor es totalmente adaptable al prototipo.

Tabla 1

Comparación funcionalidades sensores.

Requerimientos	Importancia	HC-SR04		Sensor nivel de agua		Sensor nivel switch flotador agua	
		Cumple	Valor	Cumple	Valor	Cumple	Valor
Poseer pines de entrada y salida		Cumple	Valor	Cumple	Valor	Cumple	Valor
	1	Si	1	Si	1	Si	0,7
Permitir comunicación con servidor	1	Si	1	Si	1	Si	1
Conocido y confiable	0,7	Si	1	Si	0,8	Si	0,9
Fácil configuración	0,8	Si	1	Si	1	Si	1
Datos transmitidos garantizados	1	Si	1	Si	1	Si	0,9
Total		100%		96%		90%	

3.2.1.3. Selección del sensor en base a comparación

Mediante los resultados que se obtuvieron en la Tabla 1, se puede concluir que el sensor HC-SR04 fue el que mayor puntaje de cumplimiento obtuvo acorde a las características solicitadas para el desarrollo del proyecto, por lo cual este sensor es el seleccionado para el desarrollo del prototipo. De igual manera existe documentación suficiente que permite entender el funcionamiento del sensor e implementar de una manera correcta dentro del proyecto.

3.2.2. Elección servidor base de datos

Seleccionar servidor que alojará la base de datos, el cual nos permitirá alojar la información que se recibirá desde el sensor.

3.2.2.1. Requerimiento general

- El almacenamiento disponible de la base de datos debe ser mayor a 500MB.
- Almacenamiento en la nube
- Escalabilidad
- Disponible desde cualquier lugar
- Adaptabilidad para aplicaciones Web o móviles.
- Multiconexión
- Conveniencia en costos

3.2.2.2. Opciones consideradas para base de datos

Para la elección de una base de datos que sea adecuada en la elaboración del sistema, se presentan las opciones para revisión de sus características y poder elegir la más adecuada. A continuación, se presentan los motores de base de datos en consideración:

- SQL Server
- My SQL
- Cloud Firestore de Firebase

SQL Server (Microsoft, 2018b)

- Instalación *Standalone* o despliegue en la nube (Azure).
- Base de datos relacional.
- Ideal para aplicaciones que demanden constante modificación de escritura.
- Escalabilidad, estabilidad y seguridad.

- Capacidad de almacenamiento mayor a 1 GB.

MySQL (Mysql, 2013)

- Instalación *Standalone*.
- Multiplataforma.
- Base de datos relacional.
- No tiene costo de licenciamiento, ya que funciona mediante una Licencia Pública General (GPL).
- No tan eficaz con aplicaciones que posea constante modificación de datos.
- Capacidad de almacenamiento mayor a 2 GB.

Cloud Firestore de Firebase (Solanki, 2018)

- Despliegue en la nube.
- Base de datos NoSQL.
- Solución desarrollada por Google para desarrollo de aplicaciones Web y móviles.
- Capacidad de almacenamiento mayor a 1GB.
- Diferentes planes, “*Spark*” no tiene costo.

3.2.2.3. Cuadro comparativo entre opciones de base de datos

En la Tabla 2, se observa la comparación realiza entre las opciones para la elección de un motor de base de datos que será el encargado de almacenar la información que se obtenga del sensor, la misma que para tener un acceso desde cualquier parte de internet y no poseer hardware se debe encontrar en la nube. De igual manera debe cumplir los requisitos mencionados anteriormente, entre los más importantes para el desarrollo se encuentra que sea compatible con el servidor y el lenguaje Python, así que sea fácil de adaptar para el aplicativo web.

Tabla 2

Comparativo base de datos.

Requerimientos	Importancia	SQL Server		My SQL		Firestore de Firebase	
		Cumple	Valor	Cumple	Valor	Cumple	Valor
Capacidad de almacenamiento mayor a 500 MB	1	Si	1	Si	1	Si	1
Almacenamiento en la nube	1	Si	1	No	0	Si	1
Escalamiento	1	Si	1	Si	0,9	Si	0,7
Disponibilidad desde cualquier lugar	1	Si	1	Si	0,8	Si	1
Conexiones simultáneas	1	Si	1	Si	1	Si	1
Costo	1	Si	0,8	Si	1	Si	1
Total		97%		78%		95%	

3.2.2.4. Elección base de datos

El motor de base de datos seleccionado es SQL Server, el cual es elegido para el proyecto, en base al porcentaje más alto que obtuvo entre las opciones que se analizaron.

3.2.3. Elección del servidor de Base de datos

Para poder abordar todos los objetivos que se fijaron en el proyecto, se requiere un servidor que permita implementar la comunicación mediante el sensor seleccionado y SQL Server, el mismo será desplegado en la nube de Azure.

3.2.3.1. Requerimiento general

- Soporte conexión con el sensor
- Procesamiento mínimo 1GHz
- Conexión inalámbrica
- Consumo de energía bajo
- No ocupe espacios amplios
- Mantenimiento fácil
- Bajo costo

3.2.3.2. Opciones consideradas

- Lenovo ThinkCentre M710 Tiny
- Intel NUC
- Raspberry Pi 3B+

A continuación, se describen las características de las opciones consideradas para la elección del servidor.

Características Lenovo ThinkCentre M710 Tiny (Lock & Mount, 2020)

- No soporta comunicación directa con el sensor.
- Procesador core i5-7500 de 3.3GHz.
- Intel 8265 ac.
- Consumo de energía 65W.
- Tamaño: 170mm x 183mm x 34,5mm.
- RAM 8GB.
- Disco 256 SSD.
- Precio en el mercado: \$600 aprox.

Características Intel NUC 10 (Intel, 2018)

- No soporta comunicación directa con el sensor.
- Procesador core i5-10210U de 4.1GHz.

- Intel Wireless-AX MAC.
- Consumo de energía 25W.
- Tamaño: 117mm x 112mm x 51mm.
- RAM 16GB.
- Disco 512 SSD.
- Precio en el mercado: \$700 aprox.

Características Raspberry Pi 3 Model B+ (Raspberry Pi Foundation, 2016)

- Comunicación directa con el sensor puertos GPIO (40pines).
- Procesador Broadcom BCM2837B0, Cortex-A53 1,4GHz
- Tarjeta inalámbrica 2.4GHz y 5GHz
- Mínimo consumo de energía 5V y 2.5A DC.
- Tamaño: 82mm x 56mm x 19,5mm.
- RAM 2GB.
- Memoria MicroSD 16GB
- Precio en el mercado: \$60 aprox.

3.2.3.3. Cuadro comparativo para elección de servidor

En la Tabla 3 se puede observar una comparación realizada entre las opciones a considerar para usarlos como servidor. Del cual se puede visualizar que Raspberry PI es el servidor elegido para el desarrollo del proyecto, gracias a su bajo costo, cumple todas las funcionalidades requeridas y se puede adaptar en espacios pequeños.

Para realizar el comparativo entre las opciones seleccionadas previamente debemos tener en consideración que debe cumplir con los requerimientos en los que debe tener una comunicación directa con el sensor de distancia que se escogió previamente, de igual manera debe tener una compatibilidad con a base de datos SQL Server que se escogió en la sección anterior y que será la encargada de almacenar los datos que se genere mediante el script que se ejecutará en el servidor y sean captados desde el sensor.

Tabla 3

Comparativo de servidores.

Requerimientos	Importancia	Lenovo ThinkCentre M710 Tiny		Intel NUC 10		Raspberri Pi 3B+	
		Cumple	Valor	Cumple	Valor	Cumple	Valor
Soporta conexión con sensor		Cumple	Valor	Cumple	Valor	Cumple	Valor
	1	No	0	No	0	Si	1
Procesamiento mínimo 1GHZ	1	Si	1	Si	1	Si	0,9
Conexión inalámbrica	1	Si	1	Si	1	Si	1
Consumo de energía bajo	1	Si	0.7	Si	0,8	Si	1
No ocupe amplios espacios	1	Si	1	Si	1	Si	1
Bajo Costo	1	No	0	No	0	Si	1
Total		60%		63%		98%	

3.2.4. Elección del servidor de correo

Para enviar las notificaciones dentro del proyecto se requiere un servidor de correo, tanto desde los técnicos hacia los usuarios como viceversa. Por este motivo, a continuación, se presentan los requerimientos y consideraciones para una elección adecuada.

3.2.4.1. Requerimiento general

- Disponibilidad 24/7
- Fácil configuración

- Adaptabilidad con la aplicación web y móvil.
- Bajo costo
- Disponibilidad de información

3.2.4.2. Opciones consideradas

- Gmail
- Outlook
- hMailServer

A continuación, se describen las características de las opciones consideradas para la elección del servidor de correo.

Gmail (Centro de Apoyo Tecnológico a Emprendedores & Albacete, 2013)

- Despliegue en la nube
- Configuración sumamente fácil
- Existe documentación precisa para adaptar a aplicaciones web y móviles
- Cada licencia tiene un costo de \$8
- Amplia información

Outlook (Microsoft, 2018)

- Despliegue en la nube
- Configuración sumamente fácil
- Existe documentación precisa para adaptar a aplicaciones web y móviles
- Cada licencia tiene un costo de \$5
- Amplia información
-

hMailServer (IBM Corporation, 2009)

- Despliegue *standalone*.
- Configuración previo conocimiento de la herramienta.

- Experiencia intermedia para configurar con aplicaciones externas.
- Licenciamiento sin costo.
- Existe información en internet.

3.2.4.3. Cuadro comparativo entre servidores de correo

En la Tabla 4, se puede observar una comparación realizada entre las opciones a considerar para usarlos como servidor de correo. Por lo cual se puede deducir que la herramienta para correo electrónico que más se adapta a nuestro proyecto viene siendo OUTLOOK, por lo cual se trabajará con este servidor de correo.

Tabla 4

Comparación servidor de correo.

Requerimientos	Importancia	GMAIL		OUTLOOK		hMailServer	
		Cumple	Valor	Cumple	Valor	Cumple	Valor
Disponibilidad 24/7	1	Si	1	Si	1	Si	0,8
Fácil configuración	1	Si	1	Si	1	Si	0,9
Adaptabilidad	1	Si	1	Si	1	Si	0,7
Bajo costo	1	Si	0,9	Si	1	Si	1
Disponibilidad de Información	1	Si	1	Si	1	Si	0,9
Total		98%		100%		86%	

3.2.5. Elección de *framework* para aplicación web

Para la elaboración del aplicativo web y móvil, será necesario seleccionar un entorno de trabajo. Por este motivo, a continuación, se presentan los requerimientos y consideraciones para una elección adecuada.

3.2.5.1. Requerimiento general

- Fácil integración con SQL Server
- Permitir mantener organización en el proyecto
- Sea adaptable
- Información accesible
- Bajo costo

3.2.5.2. Opciones consideradas

- ASP.NET
- Angular
- PHP

A continuación, se describen las características de las opciones consideradas para la elección del *framework* para la aplicación web.

ASP.NET (Danysoft, 2017)

- Fácil integración con SQL Server
- Contenido y código, por separado
- Soporte de Lenguajes y compatibilidad con navegadores
- Amplia información en la web
- Versión gratuita y de pago

Angular (Jadhav et al., 2015)

- Experiencia intermedia para integración con SQL Server
- Organización en la estructura del proyecto
- Compatibilidad con navegadores, lenguajes de programación limitados
- Amplia información en la web
- Versión gratuita

PHP (Alvarez, 2016)

- Integración con SQL server, PHP recomienda el uso de My SQL
- Amplia información en la Web
- Compatibilidad con navegadores y lenguajes de programación
- Organización en el proyecto
- Versión gratuita

3.2.5.3. Cuadro comparativo entre opciones de *framework*

En la Tabla 5, se puede observar la comparación realizada para la elección del *framework* que más se adapte al proyecto a implementar. En base a la comparación realizada el *framework* que se va a utilizar es ASP.NET la cual se va a trabajar con la herramienta de desarrollo Visual Studio, posee una integración fácil con la base de datos y también con los componentes el desarrollo del proyecto.

Tabla 5

Comparación de Framework a elegir.

Requerimientos	Importancia	ASP.NET		Angular		PHP	
		Cumple	Valor	Cumple	Valor	Cumple	Valor
Fácil integración con Base de Datos	1	Si	1	Si	0,8	Si	0,9
Organización en proyecto	1	Si	1	Si	0,9	Si	1
Adaptable	1	Si	1	Si	1	Si	1
Información en la web	1	Si	1	Si	1	Si	1
Bajo costo	1	Si	1	Si	1	Si	1
Total		100%		93%		98%	

3.3. Requerimientos etapa de aplicación y monitoreo

En esta sección se presentarán los requerimientos que serán utilizados para la creación del aplicativo web y móvil, los mismos que se basan en el estudio previo para la elaboración del proyecto.

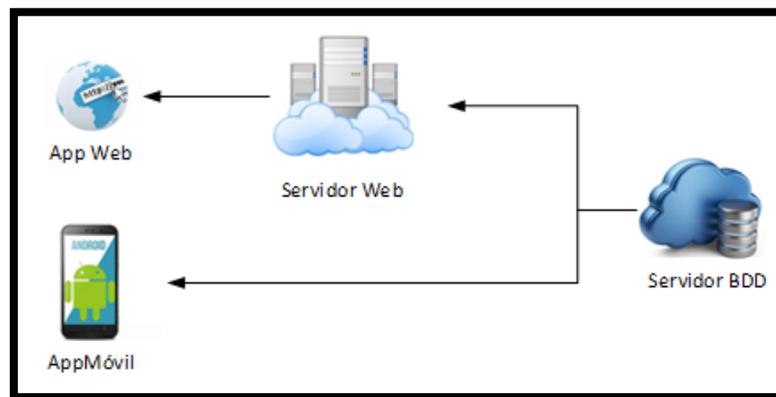


Figura 10. Requerimientos aplicativo web y móvil.

En la Figura 10, se puede observar los componentes necesarios para el funcionamiento del aplicativo web y móvil. Los mismos constan de una base de datos única que se encuentra en la nube y podrán obtener los datos para la presentación de los datos obtenidos desde el sensor.

3.3.1. Análisis de requerimientos aplicación web

Para poder desarrollar el aplicativo web se consideraron requerimientos acordes a la necesidad del proyecto, los cuales se detallarán a continuación

3.3.1.1. Requerimientos funcionales

Un requerimiento funcional, se puede decir que es la interacción entre el sistema y el ambiente, es decir los requerimientos funcionales son descritos como las características técnicas que va a poseer el sistema. (Cristia, 2011)

En la Tabla 6, se puede observar un listado de requerimientos funcionales que van a ser utilizados para la elaboración del aplicativo web.

Tabla 6

Listado requerimientos funcionales.

RF	DESCRIPCIÓN
RF01	La aplicación web manejará tres roles diferentes: administrador, técnico, usuario final.
RF02	La aplicación únicamente permitirá registrar usuarios con el perfil de administrador.
RF03	La aplicación permitirá que el usuario pueda recuperar su contraseña.
RF04	Los roles y los usuarios serán modificables únicamente por el administrador.
RF05	En la aplicación se podrá visualizar los informes emitidos por el sensor.
RF06	La aplicación generará notificaciones de alerta para un monitoreo más eficaz.
RF07	La aplicación permitirá enviar notificaciones tanto al usuario registrado como a técnicos y administrador.
RF08	La aplicación permitirá iniciar sesión con lo cual se identificará el rol del usuario que inicia sesión.
RF09	Se podrá finalizar sesión desde la aplicación.

3.3.1.2. Requerimientos no funcionales

Un requerimiento no funcional se puede decir que es un impedimento que existirá sobre el sistema.

En la Tabla 7, se presentan los requerimientos no funcionales que van a ser parte del aplicativo web.

Tabla 7

Listado de requerimientos no funcionales.

RNF	DESCRIPCIÓN
RNF01	No se permite duplicidad de usuarios.
RNF02	Las contraseñas de los usuarios no se almacenarán en texto plano.
RNF03	El aplicativo móvil se lo desarrollará únicamente para sistema operativo Android.
RNF04	La base de datos deberá estar acorde al número de usuarios que se disponga.

3.3.2. Diagrama de casos de uso

Es un método específico para la utilización del sistema, se describe una historia de uso particular del sistema. Aquí se puede definir los límites del sistema y como se encuentra relacionado con el sistema y su interfaz. (Gutiérrez, 2011)

Se elaboró el diagrama de caso de uso en el cual se puede analizar los requerimientos funcionales presentados previamente. En la Figura 11 se puede visualizar el diagrama en el cual cada uno de los roles que se implementará tendrá acceso a los recursos de la aplicación, es decir el administrador del sistema tendrá total acceso al aplicativo para la creación de usuarios, asignación de roles, visualización de reportes y envío de notificaciones; por otro lado, el técnico únicamente podrá iniciar sesión en el sistema, visualizar el estado del contenedor y envío de notificaciones al igual que el usuario.

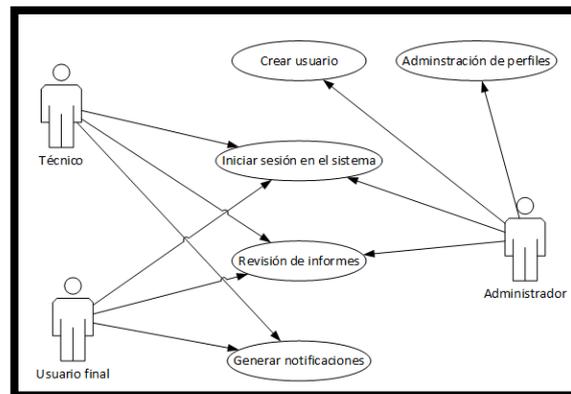


Figura 11. Diagrama caso de uso aplicación web.

3.3.2.1. Permisos y roles

Una vez que se ha definido el diagrama de casos de usos se implementará los siguientes permisos y roles para el desarrollo del sistema:

Permisos

- **Administrador:** Tendrá acceso a todo el sistema.
- **Técnico:** No puede realizar cambios al perfil de usuarios.
- **Usuario final:** Únicamente puede visualizar los informes y generar notificaciones (envío correo electrónico al sistema).

Roles

- **Administrador:** Se encarga de la creación, modificación y eliminación de usuarios, asigna los roles, tiene acceso a todos los módulos.
- **Técnico:** Puede visualizar los datos de los usuarios, revisión de informes, envío de notificaciones desde la plataforma para agenda de citas de recolección.
- **Usuario final:** Visualización de informe de estado de contenedor, envío de correo electrónico para solicitar recolección de contenedor.

3.3.2.2. Diagrama de clases

En esta sección se presentará el diagrama de clase que va a ser utilizado para el desarrollo del aplicativo web, en el mismo constan los roles definidos, acceso al aplicativos mediante usuario y contraseña, así como los datos de los usuarios que serán almacenados, esto se lo puede visualizar en la Figura 12.

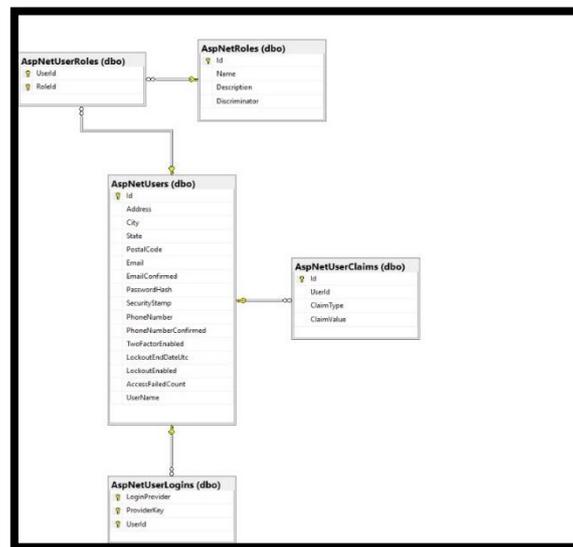


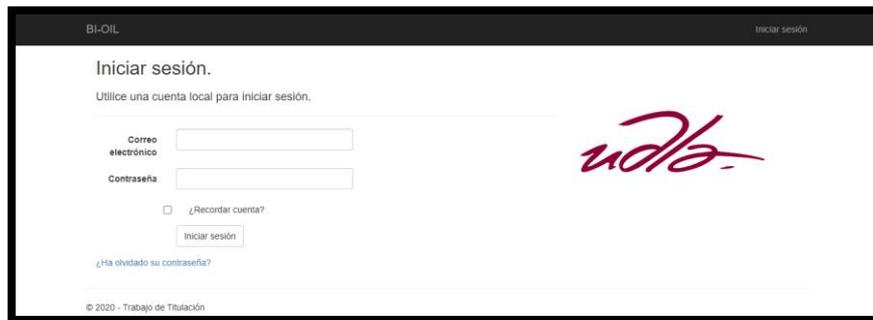
Figura 12. Diagrama clases para aplicativo web.

3.3.2.3. Vistas a ser implementadas en aplicativo web

A continuación, podemos observar las vistas correspondientes a ser implementadas dentro de la aplicación web a desarrollar las misma que se basan en los requerimientos establecidos anteriormente.

- **Inicio de sesión o *login***

En la Figura 13, se puede observar cómo deberá ser el diseño del inicio de la página inicial mostrada a los usuarios y gestores.

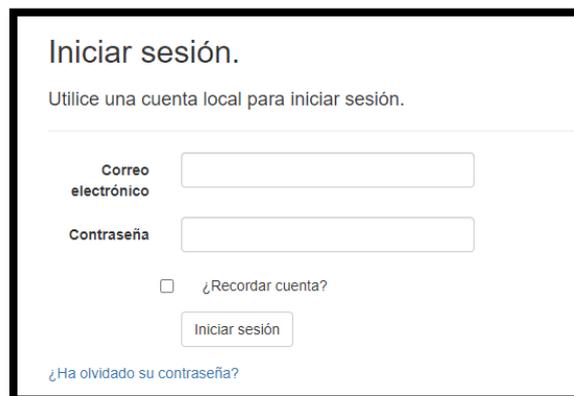


The screenshot shows the initial login page of the BI-OIL web application. The page has a dark header with 'BI-OIL' on the left and 'Iniciar sesión' on the right. The main content area is white and contains the following elements: the title 'Iniciar sesión.', the instruction 'Utilice una cuenta local para iniciar sesión.', two input fields for 'Correo electrónico' and 'Contraseña', a checkbox for '¿Recordar cuenta?', an 'Iniciar sesión' button, and a link for '¿Ha olvidado su contraseña?'. A red logo is visible on the right side of the page. At the bottom left, there is a copyright notice: '© 2020 - Trabajo de Titulación'.

Figura 13. Vista inicial del aplicativo web.

- **Ingreso al aplicativo web**

En la Figura 14, se puede observar cómo deberá ser el diseño del inicio de sesión para que los usuarios puedan acceder al aplicativo



This image provides a detailed view of the login form design. It features the title 'Iniciar sesión.' and the instruction 'Utilice una cuenta local para iniciar sesión.' above a horizontal line. Below the line are two input fields: 'Correo electrónico' and 'Contraseña'. Underneath these fields is a checkbox labeled '¿Recordar cuenta?' and an 'Iniciar sesión' button. At the bottom left, there is a link for '¿Ha olvidado su contraseña?'. The entire form is enclosed in a black border.

Figura 14. Inicio de sesión.

- **Registro para usuarios nuevos**

En la Figura 15, se puede observar cómo deberá estar el formulario para la creación de nuevos usuarios que tendrán acceso al aplicativo

Figura 15. Registro de usuarios.

- **Presentación de gráfico de estado del contenedor**

En la Figura 16, se puede visualizar como se presentarán los gráficos del estado del contenedor hacia los gestores para un monitoreo y posterior planificación de una cita de recolección, así como a los usuarios finales para conocer el estado del líquido dentro del contenedor.



Figura 16. Informe de resultados.

3.3.3. Análisis de requerimientos aplicación móvil

De igual manera que se realizó con el aplicativo web, para la creación del aplicativo móvil se presentan los requerimientos necesarios.

3.3.3.1. Requerimientos funcionales

Los requerimientos funcionales son las características propias que tendrá la aplicación para su correcto funcionamiento, es así que para la realización del aplicativo móvil del sistema se tienen los siguientes requerimientos detallados en la Tabla 8.

Tabla 8

Requerimiento funcional aplicación móvil.

RF	DESCRIPCION
RF01	La aplicación móvil solo manejará el rol de usuario final.
RF02	El rol de usuario permitirá ingresar al sistema, previo inicio de sesión.
RF03	La aplicación permitirá verificar los resultados de estado del contenedor.
RF04	La aplicación permitirá enviar notificación a los técnicos y administradores mediante correo electrónico.

3.3.3.2. Requerimientos no funcionales

Un requerimiento no funcional se describe como una parte que no tendrá afectación en las características del aplicativo, en la Tabla 9 se describe que la aplicación únicamente será desarrollada para sistema operativo Android.

Tabla 9

Requerimiento no funcional aplicación móvil.

RNF	DESCRIPCION
RNF01	Aplicación desarrollada para sistema operativo Android.

3.3.3.3. Diagrama de casos uso

En la Figura 17, se describe el diagrama de casos de uso planteado para el desarrollo del aplicativo móvil en el cual se puede visualizar que el usuario final podrá iniciar sesión dentro de la aplicación, podrá visualizar el estado del contenedor y a su vez enviar notificaciones dentro el aplicativo que llegará al correo electrónico del gestor.

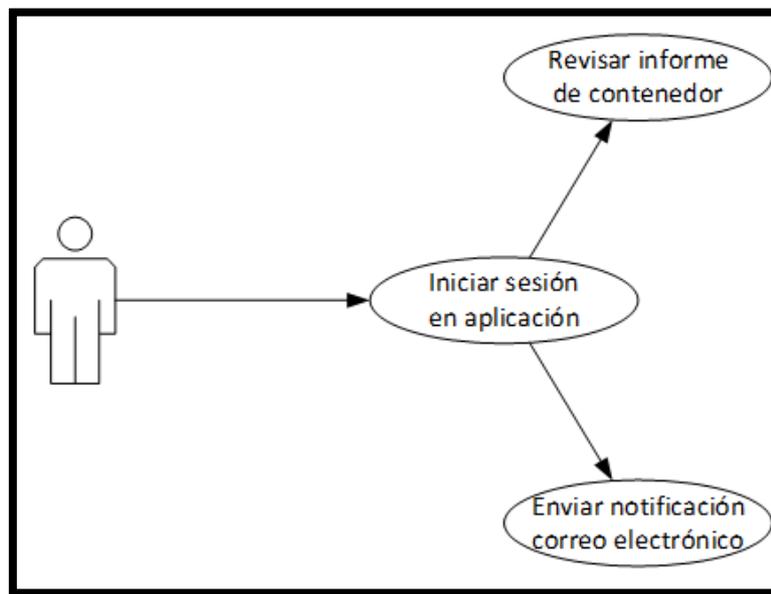


Figura 17. Diagrama casos de uso.

3.3.3.4. Diagrama de clases

El diagrama de clases nos permite visualizar el funcionamiento de la aplicación antes de implementarla dentro del *framework* que se vaya a desarrollar, es así como en la Figura 18 se puede observar las clases necesarias para desarrollar cada una de las vistas del aplicativo móvil.

3.3.3.5. Vistas aplicativo móvil

En esta sección se presentará las vistas de como tendrá que ser desarrollado el aplicativo móvil.

- **Inicio**

En la Figura 19, se puede observar cómo deberá ser implementada la vista inicial al momento que se ejecute la aplicación.

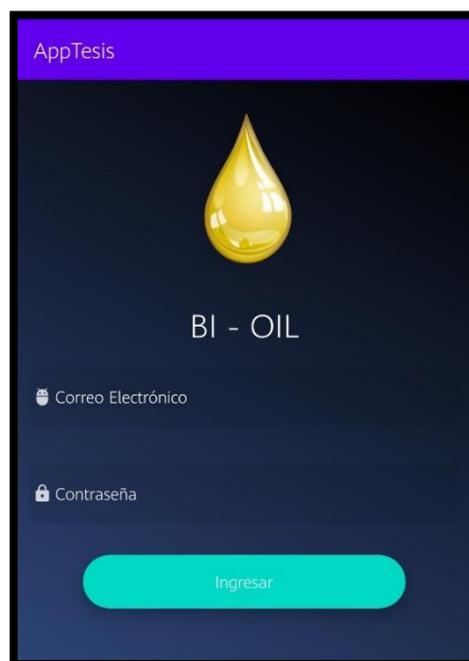


Figura 19. Inicio aplicación.

- **Generación de gráficos del estado del contenedor**

En la Figura 20, se puede observar que la aplicación móvil deberá poseer una vista en la cual se muestre el gráfico del comportamiento del líquido dentro del contenedor.

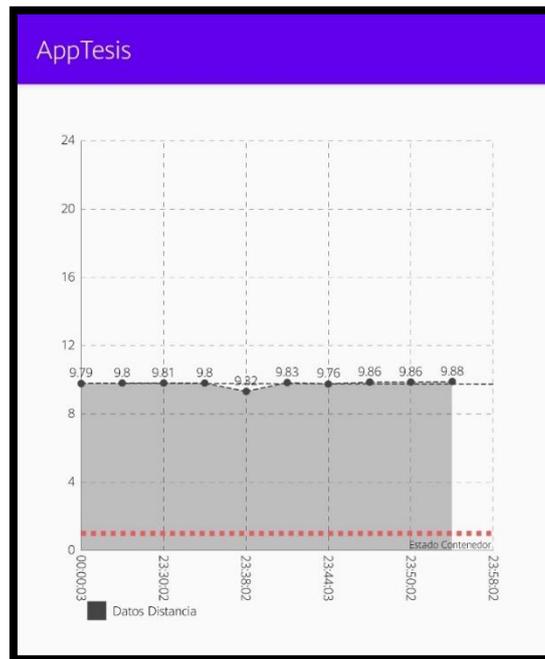


Figura 20. Informe estado contenedor.

- **Envío de notificación (correo electrónico)**

El usuario dentro del aplicativo deberá tener un espacio en el cuál pueda enviar notificaciones al gestor en caso de que el contenedor tuviese una falla o quisiera solicitar una cita de recolección.

Figura 21. Notificación mediante correo electrónico.

3.4. Direccionamiento de red

La Tabla 10 contiene la información con el direccionamiento IPv4 que se implementará en los dispositivos físicos para la ejecución del proyecto. El motor de base de datos no cuenta con un direccionamiento debido a que se encuentra en la nube.

Tabla 10

Direccionamiento de red de hardware utilizado.

Dispositivo de Red	Tipo de conexión	Dirección IP / Enlace	Máscara	Tipo de Asignación
Gateway	Cableada	192.168.1.158	/27	Estática
Raspberry Pi 3 B+	Inalámbrica	192.168.1.137	/27	Estática
Base de datos	Direccionamiento en la nube	bi-oilddbserver.database.windows.net	NA	NA

4. IMPLEMENTACIÓN DEL SISTEMA DE MONITOREO DE NIVEL DE ACEITE USADO

En esta sección se presenta la implementación de la red, la fase de obtención de datos y almacenamiento de estos en la base de datos SQL Server que se encuentra en la nube, de igual manera la presentación de los informes de monitoreo mediante la aplicación web y móvil.

4.1. Implementación del direccionamiento de red IPv4

A continuación, se observa el diagrama de red, la tabla de direccionamiento y las configuraciones de red necesarios en cada uno de los equipos, en los cuales el servidor Raspberry es el único que mantiene direccionamiento estático mientras que el equipo de pruebas y dispositivo móvil tienen direccionamiento por DHCP.

Mediante la Figura 22, se puede visualizar el detalle del diagrama de red necesario para la implementación del proyecto, el direccionamiento utilizado dentro de la red LAN, es decir el servidor Raspberry, laptop, teléfono y modem, así como también la dirección web que posee la base de datos SQL Server mediante el cual se van a poder comunicar todos los dispositivos.

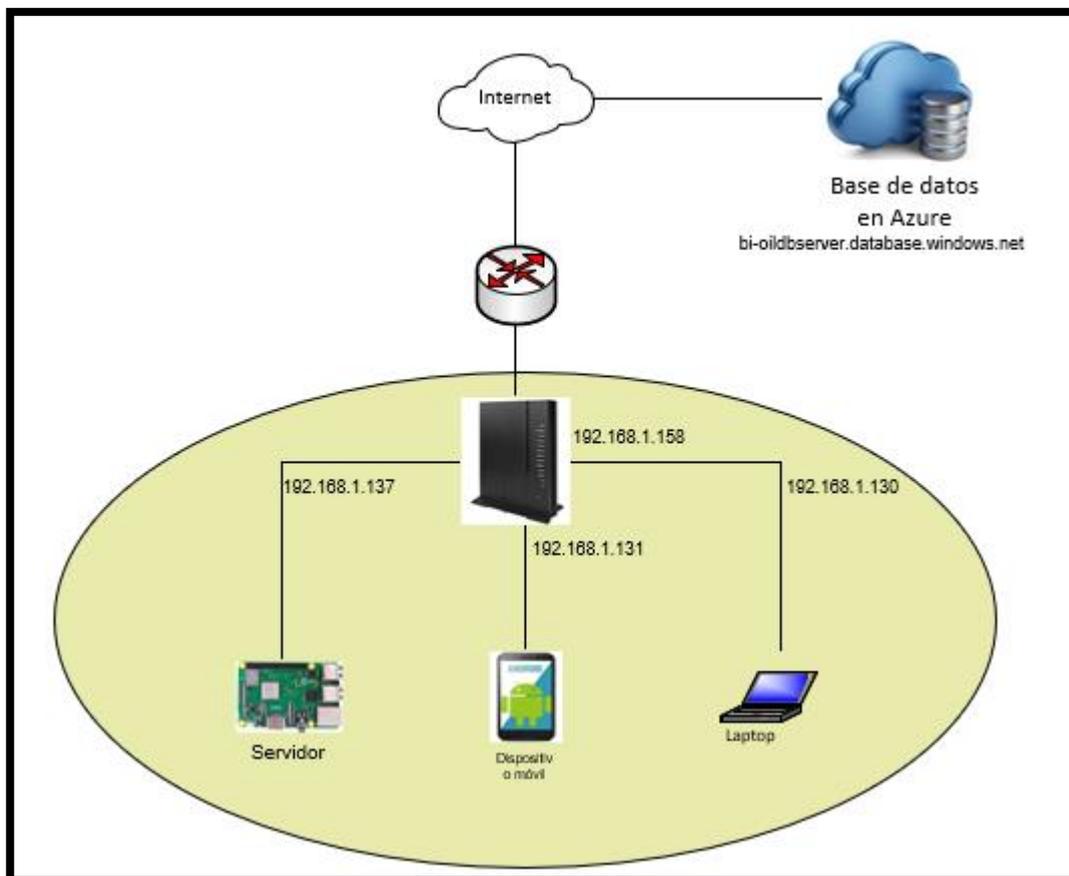


Figura 22. Diagrama de red del proyecto.

4.1.1. Red interna

Para la implementación del direccionamiento de red dentro del sistema de monitoreo será desde la conexión de red local en donde se vaya a implementar el prototipo, la misma que debe contar con salida hacia internet. Únicamente el servidor en este caso la Raspberry deberá contar con un direccionamiento estático para la comunicación con la base de datos. Es así que en la Tabla 11 contiene el direccionamiento necesario que se utilizó para la implementación del

proyecto, en el cual se detalla las direcciones IP que ocuparon cada uno de los dispositivos, el gateway que es el modem local que se lo tiene en los hogares o establecimientos comerciales, así como la dirección IP que tiene asignada la raspberry en su módulo Wireless y las direcciones asignadas a los dispositivos de manera dinámica..

Tabla 11

Direccionamiento de red interno.

Dispositivo de Red	Tipo de conexión	Dirección IP	Máscara	Tipo de Asignación
Gateway	Cableada	192.168.1.158	/27	Estática
Raspberry Pi3 B+	Inalámbrica	192.168.1.137	/27	Estática
Laptop pruebas	Cableada	192.168.1.130	/27	DHCP
Dispositivo móvil	Inalámbrica	192.168.1.131	/27	DHCP

4.1.1.1. Configuración de red

En esta sección se detallan las configuraciones de red realizadas en los dispositivos, gateway y raspberry.

4.1.1.2. Configuración de dirección IP del Gateway

El módem inalámbrico Calix modelo 813Gv2-1, es el dispositivo que se utilizó para la configuración de red LAN el cual es fundamental para poder desarrollar el proyecto. El mismo se debe tener en consideración que debe constar de un direccionamiento IP privado que es de uso local, por tal motivo se utilizará para la implementación la red local 192.168.1.128/27 que corresponde a la clase C y es de uso privado.



Figura 23. Modem inalámbrico Calix 813Gv2-1.

Tomado de: (Calix, 2015)

4.1.2. Configuración de red del servidor (Raspberry Pi 3 B+)

La Figura 24, muestra la configuración de la dirección IP que se realizó en la tarjeta inalámbrica, la misma que se encuentra dentro de la red local utilizada es decir 192.168.1.128/27, la misma que cuenta con salida hacia internet para comunicación con la base de datos y actualizaciones que requiera el dispositivo.

```
interface wlan0
static ip_address=172.45.93.137
static routers=172.45.93.158
static domain_name_servers=8.8.8.8
```

Figura 24. Configuración interfaz Wireless Raspberry Pi 3 B+.

4.1.2.1. Configuración del servidor (Raspberry pi 3 B+)

A continuación, se presentará el proceso realizado para la instalación del software requerido y las configuraciones realizadas para que el programa funcione de manera adecuada y poder adquirir los valores desde la Raspberry y almacenarlos la base de datos.

Características del Raspberry pi 3 B+

La Raspberry pi 3 B+ es el servidor en cual se implementó el proyecto, las características más relevantes se detallan a continuación:

- CPU Broadcom BCM2837B0, Cortex-A53 1.4GHz
- Memoria RAM de 1GB
- Módulo inalámbrico de 2.4GHz y 5GHz y Bluetooth
- 1 puerto Gigabit Ethernet
- 40 entradas GPIO para comunicación con hardware
- Puerto HDMI para conectar a una pantalla
- 4 puertos USB 2.0
- Interfaz Serial de cámara y una Interfaz serial de pantalla que permite la conexión para una cámara o una pantalla táctil
- 1 puerto de salida de audio
- Ranura Micro-SD para memoria de 16GB

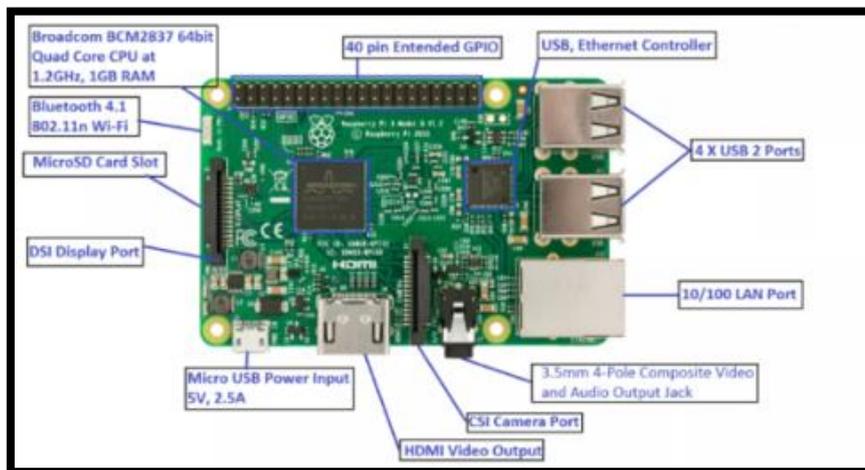


Figura 25. Raspberry Pi Modelo 3 B+.

Cuando se conecta la Raspberry Pi un indicador LED color rojo se enciende, esto significa que se encuentra energizada. Demora unos segundos y luego el sistema operativo Raspbian arranca, el mismo que previamente se debe instalar en una microSD, como se indicará en la sección 4.1.2.2.

En la Figura 25, se puede observar cada uno de los componentes que posee la Raspberry pi, para el desarrollo del proyecto los utilizados serán los puertos GPIO, módulo wireless, módulo microSD y la entrada de energía

4.1.2.2. Instalación de sistema operativo Raspbian

Para poder utilizar la Raspberry es necesario que se instale su sistema operativo dentro de una tarjeta microSD. El sistema operativo y el que se utilizó para la creación y desarrollo del proyecto es el software NOOBS, que se encuentra dentro de la página oficial de Raspberry Pi. Como se mencionó anteriormente el sistema operativo se instala dentro de una tarjeta microSD, la cual debe estar formateada. Una vez que se descargó el software NOOBS se debe descomprimir el archivo dentro de la tarjeta microSD.

En la Figura 26, se puede observar la ventana de inicio al momento que el sistema operativo carga en el servidor Raspberry.

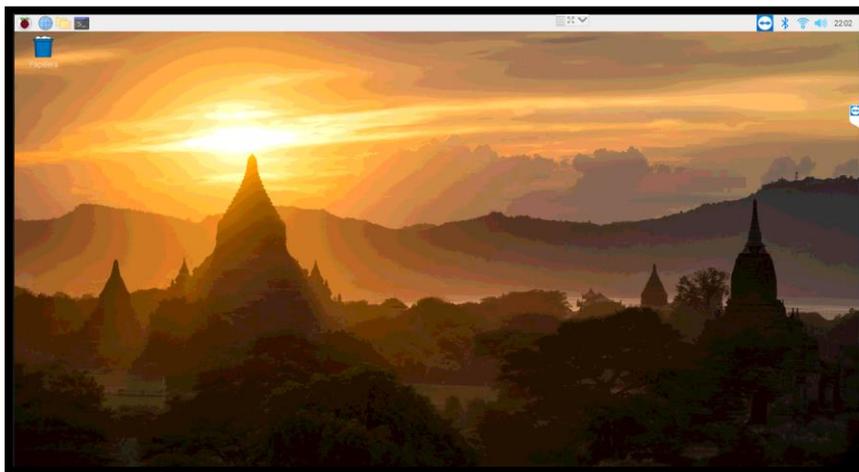


Figura 26. Escritorio sistema operativo Raspbian.

Cuando inicia el sistema operativo, se debe proceder con la actualización de este, como se puede visualizar en la Figura 27. Una vez que culmine la actualización el sistema operativo solicitará reinicio.

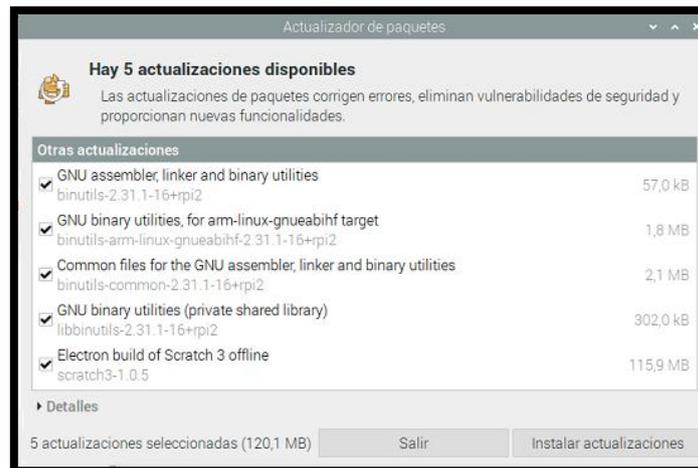


Figura 27. Actualización paquetes sistema operativo.

4.1.2.3. Instalación Python3

Cuando se inicia la Raspberry por defecto viene instalada la versión 2.7 de Python, para poder ejecutar nuestro programa de lectura de datos es necesario instalar la última versión de Python (Versión 3), ya que si se mantiene la versión 2.7 el programa tendrá errores de ejecución que no soporta varias librerías necesarias que funcionan en la versión 3, la misma se lo realiza mediante el comando mostrado a continuación:

sudo apt-get install Python-pip3.

La Figura 28, da detalle de los scripts que fueron creados para poder desarrollar el proyecto, así como su ubicación dentro del servidor donde la ruta de almacenamiento es /home/pi/Documents/Tesis.

Para la ejecución y desarrollo de todo el proyecto se utilizaron los scripts siguientes:

- ConexionBD.py: Script necesario para establecer comunicación entre el servidor Raspberry con la base de datos en Azure.
- Tesis2min.py: script encargado de ejecutar la medición en el sensor cada dos minutos y enviar los datos a ser almacenados.

- Tesis5min.py: script encargado de ejecutar la medición en el sensor cada dos minutos y enviar los datos a ser almacenados.

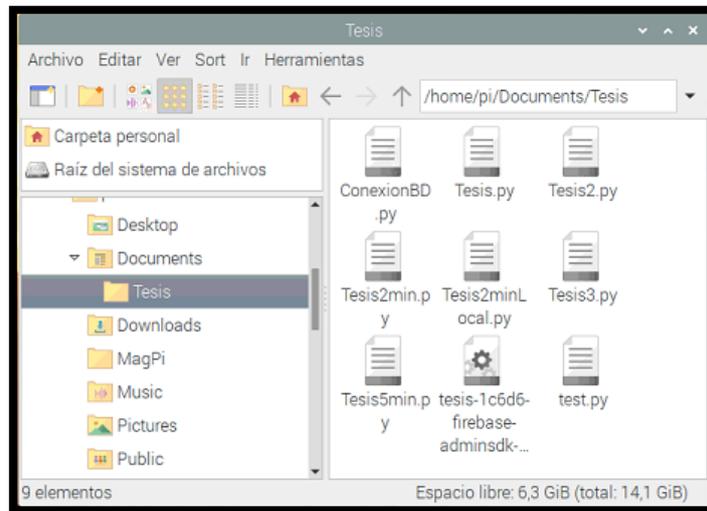


Figura 28. Archivos de programa para ejecución de proyecto.

A continuación, se procederá analizar el script “Tesis2min.py”, el mismo que se ejecuta dentro del servidor Raspberry y es el encargado de poner en funcionamiento al sensor cada dos minutos y este envíe los datos hacia la base de datos. De igual manera en el Anexo se incluirán todos los scripts que se ejecutan dentro del servidor.

En la Figura 29, se observa las que se van a ejecutar para establecer comunicación con el sensor y base de datos.

```
import RPi.GPIO as GPIO
import time
import sys
import os
from datetime import date
from datetime import datetime
from os import getenv
import pymysql
import subprocess
```

Figura 29. Importación de librerías para ejecución de programas.

En la Figura 30, se puede visualizar la línea de código necesaria para que el servidor se autentique dentro de la base de datos y esta le permita almacenar los datos que se obtienen desde el sensor.

```
#Conexión base de datos
server = 'bi-oilobserver.database.windows.net'
database = 'BI-OIL_db'
username = 'vm@bi-oilobserver.database.windows.net'
password = 'Udl@2020.'
conn = pymssql.connect(server, username, password, database)
cursor = conn.cursor()
```

Figura 30. Comunicación establecida entre la Base de datos y la Raspberry Pi.

En la Figura 31 tenemos los puertos GPIO que son utilizados para la comunicación con el sensor, en este caso el puerto GPIO 4 es utilizado para enviar un sonido por 10 μ s y mediante el puerto GPIO 18 se receptorá el tiempo que demora dicho sonido en su viaje de ida y vuelta.

```
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)

TRIG = 4
ECHO = 18
num = 2

GPIO.setup(TRIG,GPIO.OUT)
GPIO.setup(ECHO,GPIO.IN)

GPIO.output(TRIG, True)
time.sleep(0.00001)
GPIO.output(TRIG, False)
```

Figura 31. Establecimiento de los puertos de comunicación y definición de variables.

En la Figura 32, se describe la fórmula utilizada para determinar la distancia entre el sensor y el líquido, la cual se detalla a continuación:

$$distancia (cm) = \frac{tiempo (\mu s) * 1 (cm)}{velocidad del sonido * 2}$$

(Ecuación 1)

Mediante la (Ecuación 1, se puede observar que las variables “start2min” que es el tiempo de ida y “end2min” el tiempo de vuelta se tiene el tiempo total del sonido emitido al cual implícitamente se lo multiplica por uno y se divide para “0,000058” que es la velocidad del sonido en condiciones normales (343 m/s) multiplicado por dos y el mismo realizado la conversión a centímetros/microsegundos.

```

fecha = datetime.now().strftime("%X")
hora = datetime.now().strftime("%X")

while GPIO.input(ECHO) == False:
    start2min = time.time()

while GPIO.input(ECHO) == True:
    end2min = time.time()

sig_time2min = end2min - start2min

#CM:
dist2min = sig_time2min / 0.000058

total2min = 27 - dist2min

```

Figura 32. Ejecución de la fórmula para obtener la distancia que existe entre el líquido y el sensor.

En la Figura 33, se observa la parte del script necesario para almacenar dentro de la base de datos la distancia obtenida por el sensor, así como la fecha y hora que se emiten los datos.

```

cursor.execute("INSERT INTO Tesis2min (Fecha, Hora, Distancia) VALUES ('%s', '%s', %s)" %(fecha, hora, total2min))
conn.commit()
conn.close()

```

Figura 33. Almacenamiento de valores en base de datos.

En la Figura 34, se puede observar la parte del script que imprimirá los resultados para poder ser observados de manera local, los mismo que se presentarán únicamente con dos decimales, de igual manera el comando “GPIO.cleanup()” que es el encargado de limpiar los datos que emite el sensor para cada lanzamiento del programa.

```

print('Total: {0:.2f}'.format(total2min))
print(sig_time2min)
print(start2min)
print('Fecha: ', fecha)
print('Hora: ', hora)
print('Distancia: {0:.2f} cm.'.format(dist2min))

GPIO.cleanup()

```

Figura 34. Presentación de datos internos y limpieza de pantalla luego de ejecución.

A continuación, se presentará un breve detalle de la función que cumple cada script:

- **ConexiónBD:** este script hace referencia a la conexión establecida mediante el servidor y la BDD donde se tendrá almacenados los valores obtenidos.
- **Tesis2min:** este script contiene la ejecución para receptor los datos entre el sensor y la distancia del líquido, luego los enviará para el almacenamiento en el servidor.
- **Tesis5min:** el script cumple con la misma funcionalidad que el script “Tesis2min”, la única diferencia que existe es que este se ejecuta en un periodo repetitivo de cada 5 minutos.

Es de gran importancia comentar que los scripts descritos anteriormente se ejecutan en los tiempos mencionados, gracias a una tarea programada del sistema operativo que se cumple mediante el archivo de configuración *crontab*.

4.1.2.4. Tarea programada mediante *crontab*

Crontab se trata de un archivo de texto que posee contenido especial. Este posee una lista de scripts que se ejecutan con una periodicidad diferente. Para poder configurar este archivo se debe cumplir el siguiente formato que se encuentra en la ruta de configuración **/etc/crontab** y mediante el editor de texto nano u otro de preferencia se puede realizar las modificaciones con los parámetros que se puede ver en la Figura 35, en donde se puede visualizar los

Se la puede administrar directamente desde el portal de Azure cuya URL es la siguiente, <https://azure.microsoft.com/es-es/account/>, de igual manera se puede obtener acceso desde cualquier parte de internet mediante el *SQL Server Management Studio*. En la Figura 37, se puede observar el panel de control de la cuenta de Azure para poder crear la instancia de base de datos para ser administrada desde internet.

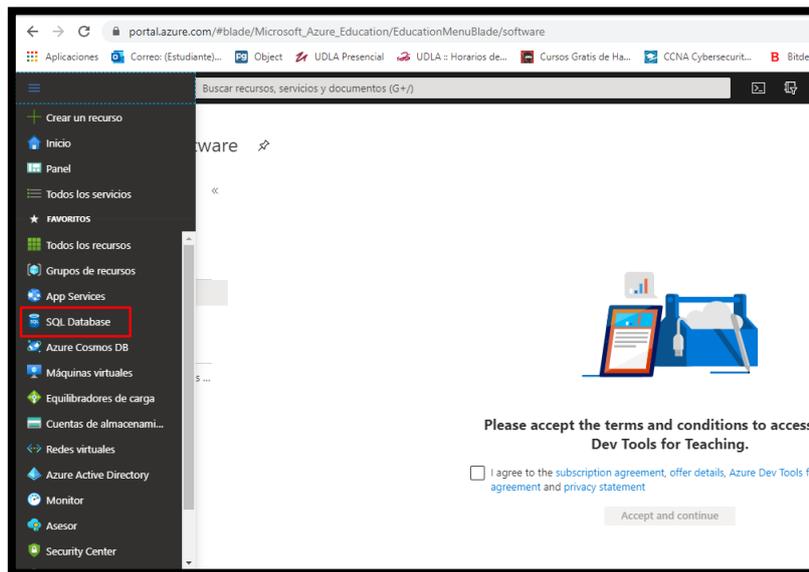


Figura 37. Portal de Azure para crear base de datos.

Una vez que se ingresa dentro de la instancia de base de datos que se creó, se procede a otorgar permisos en el firewall para poder acceder desde internet, en la Figura 38 se visualiza de mejor manera como se asigna las direcciones IP públicas para que estas tengan acceso al portal y realizar cambios dentro del servidor de base de datos.

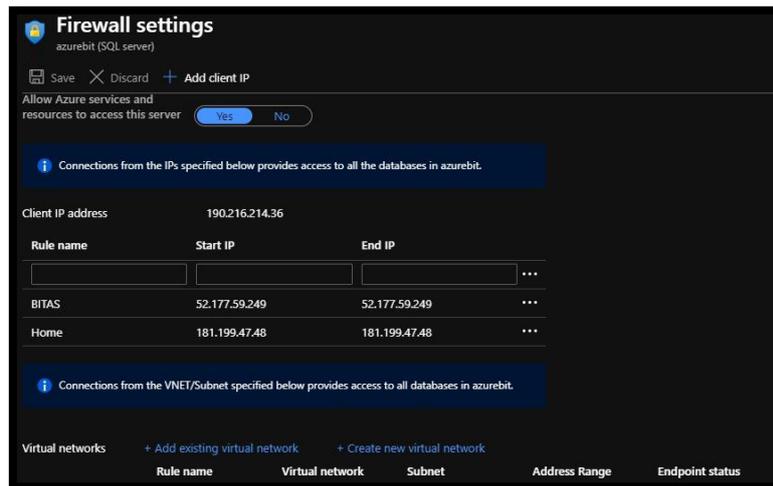


Figura 38. Permiso de IP para acceso desde internet.

Luego que ya se han asignado los permisos, mediante el SQL Server Management Studio se procede a ingresar con las credenciales de acceso, tal cual se lo puede visualizar en la Figura 39.

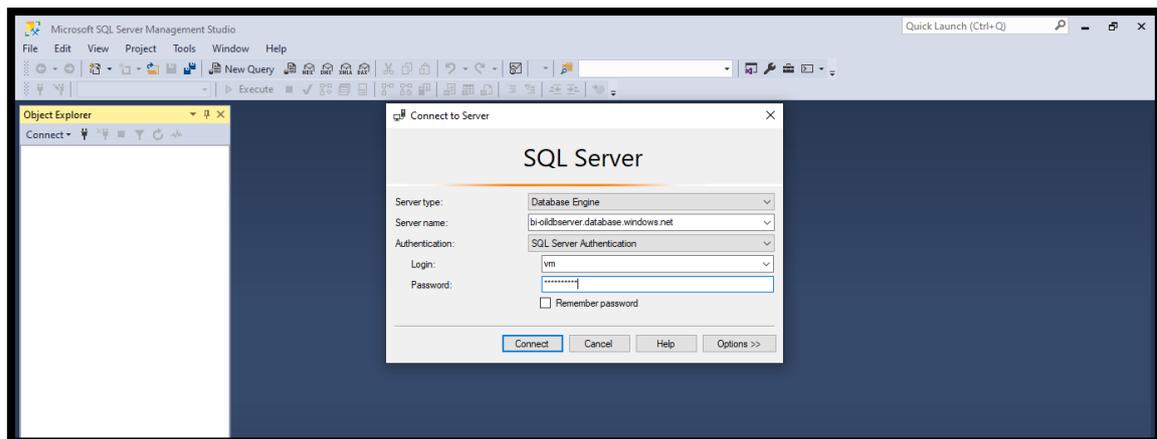


Figura 39. Acceso a SQL Server en Azure.

A continuación, una vez que se accede a la consola de administración se puede observar que se ha conectado correctamente a la Base de Datos, esto, se visualiza en la Figura 40.

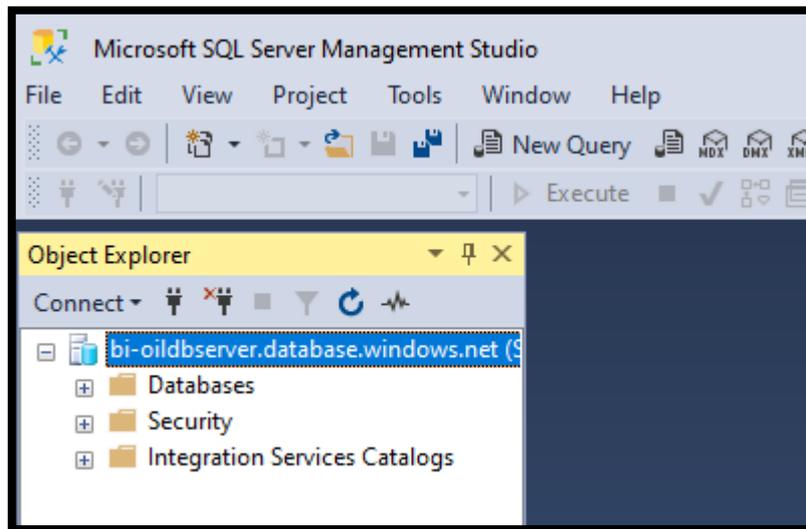


Figura 40. Conexión base de datos en Azure.

Luego, dentro de la instancia procedemos a crear la base de datos que será utilizada para la realización del proyecto. En la Figura 41 se puede observar que la base de datos “BI-OIL” se ha creado la cual contendrá todas las tablas necesarias para la implementación del sensor y el aplicativo web.

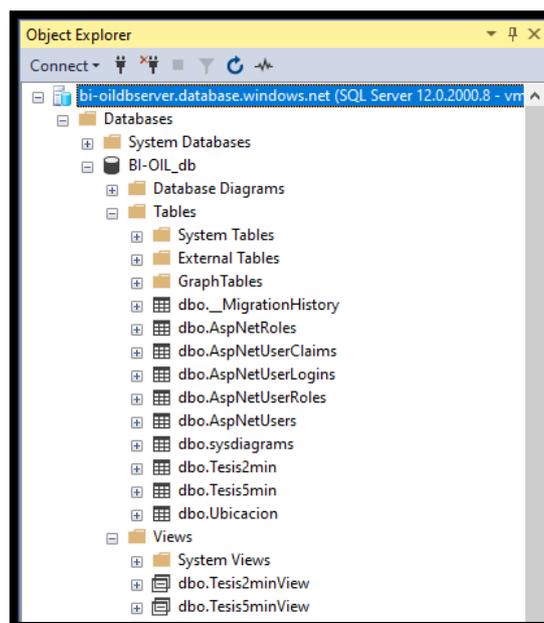


Figura 41. Creación Base de Datos BI-OIL.

Para luego proceder con la creación de las tablas necesarias tanto para la página web como para la aplicación desde donde se leerán los datos que se almacenan desde el sensor, la Figura 42 detalla la sintaxis utilizada para la creación de la Tala que almacenará los datos enviados cada cinco minutos desde el sensor.

```
create table Tesis5(  
    IdTesis5 int identity(1,1) not null,  
    Fecha date not null,  
    Hora time not null,  
    Distancia float not null,  
    constraint PK_Status5 primary key clustered (IdTesis5)  
)
```

Figura 42. Script para creación de tabla para proyecto.

Finalmente, una vez que tenemos creadas todas las tablas utilizando la misma sintaxis que la Figura 42, estarán listas para ser llamadas desde la aplicación Web y móvil desde los diferentes lenguajes de programación que se esté usando, por ejemplo: C#, Java.

4.3. Implementación de la Aplicación Web

4.3.1. Creación de Aplicativo web en Visual Studio

Para el inicio de la creación del proyecto se procederá a realizar lo siguiente: Se inicia el editor visual de Microsoft Visual Studio 2019 y luego nos solicitará el tipo de proyecto que se realizará y la ruta de almacenamiento. Una vez que se ha seleccionado los parámetros se creará el proyecto como se visualiza en la Figura 43.

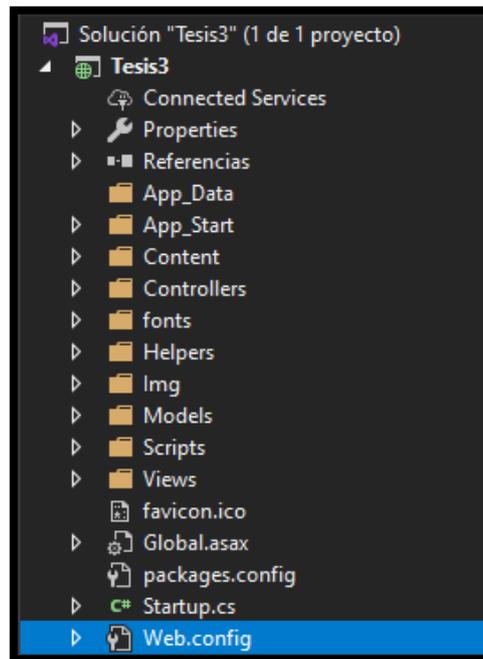


Figura 43. Visualización del proyecto para aplicación web.

Mientras se va desarrollando la aplicación se van importando librerías necesarias que permitirán que la aplicación funcione sin inconvenientes, algunas librerías las podemos ver a continuación en la Figura 44

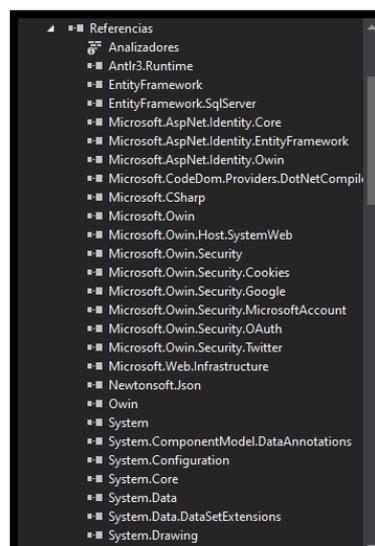


Figura 44. Librerías importadas para desarrollo de proyecto.

Conforme se va desarrollando el proyecto se crearán las diferentes clases para el funcionamiento correcto de las vistas que se implementará para dentro del aplicativo web.

En la Figura 45 se visualiza una fracción del código implementado para la elaboración del proyecto en donde se detalla la parte que es necesaria para mostrar la vista de creación de usuarios.

```
public class RegisterViewModel
{
    [Required]
    [EmailAddress]
    [Display(Name = "Email")]
    public string Email { get; set; }

    [Required]
    [StringLength(100, ErrorMessage = "La {0} debería tener como mínimo {2} de largo.", MinimumLength = 6)]
    [DataType(DataType.Password)]
    [Display(Name = "Password")]
    public string Password { get; set; }

    [DataType(DataType.Password)]
    [Display(Name = "Confirmar password")]
    [Compare("Password", ErrorMessage = "The password and confirmation password do not match.")]
    public string ConfirmPassword { get; set; }

    // Add the new address properties:
    [Display(Name = "Direccion")]
    public string Address { get; set; }

    [Display(Name = "Ciudad")]
    public string City { get; set; }

    [Display(Name = "Departamento")]
    public string State { get; set; }
}
```

Figura 45. Código para vista de creación de usuarios.

Luego, se procede a ejecutar el proyecto desde la herramienta de Visual Studio el cual se inicia de manera local en un computador desde donde se está realizando la programación, se visualiza el resultado en la Figura 46.

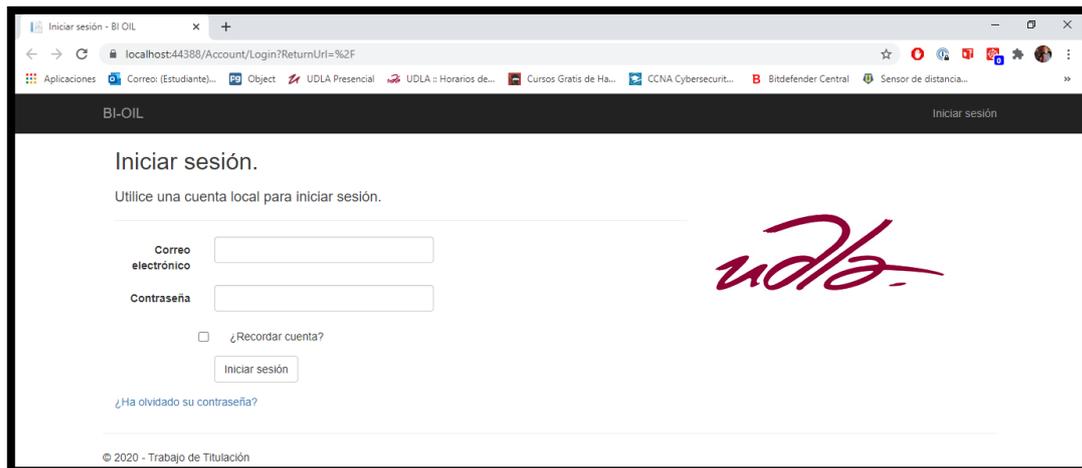


Figura 46. Lanzamiento del proyecto de manera local.

4.3.1.1. Vistas de la aplicación Web

En la Figura 47, se observa la página principal del aplicativo web una vez que hemos iniciado sesión y nos permitirá visualizar las opciones acordes al perfil que se tenga

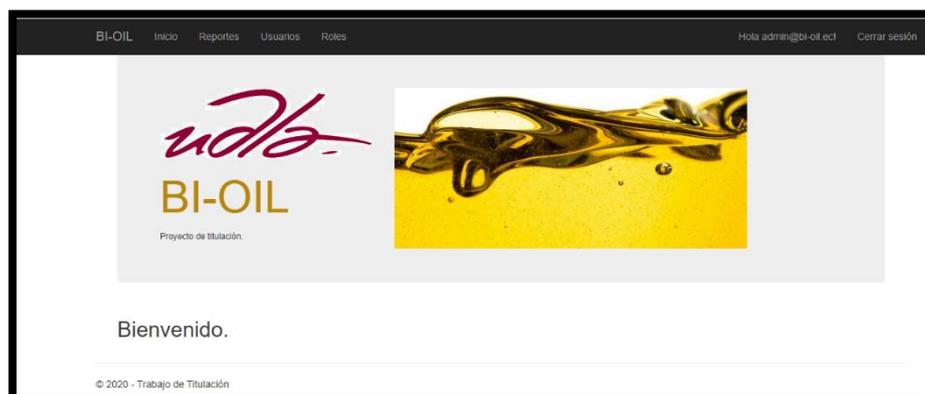


Figura 47. Página principal con usuario administrador.

En la Figura 48 se puede observar la pestaña de “Reportes” en la cual se muestran los gráficos del estado del contenedor con respecto a la distancia que existe entre el líquido y el sensor.

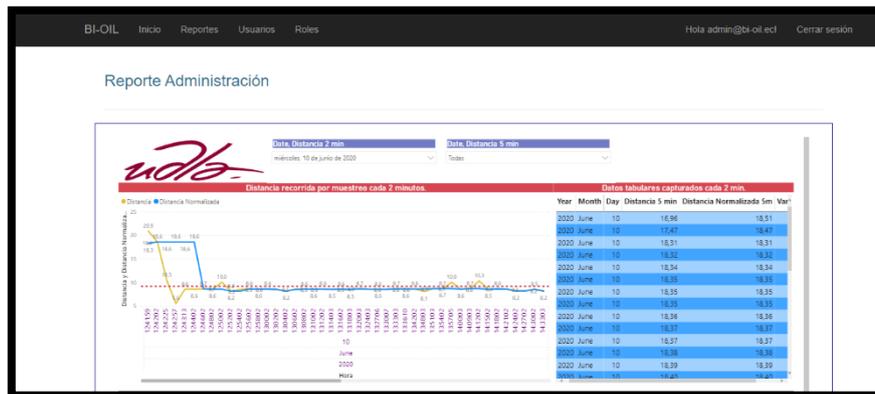


Figura 48. Vista de reportería.

En la Figura 49 se observa la pestaña para creación, edición y eliminación de los usuarios que van a tener acceso en el aplicativo web.

The screenshot displays the 'Lista de usuarios' page. It features a navigation bar at the top with 'BI-OIL Inicio Reportes Usuarios Roles' and a user session indicator 'Hola admin@bi-oil.ec Cerrar sesión'. The main content area is titled 'Lista de usuarios' and contains a 'Create New' button and a table of users.

Id	Correo	Dirección	Opciones
962d816b-3df9-4db2-a503-01ec4a9b3389	viady@hotmail.com		Editi Details Delete
ba6cf4b6-6040-4203-b7c1-88a8b69baf58	fadiaz@udanet.ec	Quito Quito 170409	Editi Details Delete
bdb75f05-7e17-45cd-bbe4-3e5499f96590	francesco-007@hotmail.com	La Vicentina 170409	Editi Details Delete
e0c3cd01-b952-4706-b656-589b3c1c9047	admin@bi-oil.ec		Editi Details Delete
ea509467-a814-4692-9629-2d9e4742fb33	alexldiaz1406@gmail.com	quito quito	Editi Details Delete

© 2020 - Trabajo de Titulación

Figura 49. Vista usuarios.

En la Figura 50 se observa los campos que se pueden editar en la creación de usuarios o bien si se necesita modificar alguna información.

BI-OIL Inicio Reportes Usuarios Roles Hola admin@bi-oil.ec Cerrar sesión

Edición de Usuario

Correo Dirección

Provincia Código Postal

Seleccione los Roles Técnico Usuario Administrador

© 2020 - Trabajo de Titulación

Figura 50. Vista edición de usuario.

En la Figura 51 se tiene la vista de administración de roles que va a tener cada usuario que se cree dentro del aplicativo, los mismos definirán que permisos tendrán los usuarios.

BI-OIL Inicio Reportes Usuarios Roles Hola admin@bi-oil.ec Cerrar sesión

Lista de Roles

Name	Description	
Técnico	Usuario técnico controlador de nivel.	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
Usuario	Usuario final con vista de reportería de control de flujo.	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
Administrador	Usuario que administra todos los requisitos del sistema	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

© 2020 - Trabajo de Titulación

Figura 51. Administración de roles.

4.4. Servidor DNS

El dominio bi-oil.ec fue adquirido mediante la empresa NIC.EC que es la encargada en el país de registrar los dominios que terminen en **.ec**, por tal motivo se utilizan los DNS de esta cuenta. En la Figura 52 se muestra el registro NS (*Name Server*), que permite ser ubicado desde internet.

List of NS Records					
Sr No	Record Id	Name	Value	Status	
1	111486607	bi-oil.ec	nicec.mars.orderbox-dns.com	Active	
2	111486608	bi-oil.ec	nicec.earth.orderbox-dns.com	Active	
3	111486609	bi-oil.ec	nicec.venus.orderbox-dns.com	Active	
4	111486610	bi-oil.ec	nicec.mercury.orderbox-dns.com	Active	

Figura 52. Nombres de servidores otorgados a dominio.

4.5. Implementación de la Aplicación Móvil

Se conoce que Android es un sistema operativo que fue creado originalmente para dispositivos móviles al que igual que iOS para iPhone, Symbian para Nokia o Blackberry OS para móviles Blackberry. Android es diferente a los mencionados anteriormente ya que su kernel se encuentra basado en Linux, el mismo que es parte de un núcleo de sistema operativo libre y gratuito. (Gonzalez, 2011).

4.5.1. Instalación de Android Studio en Windows 10

Para poder instalar el software Android Studio se debe verificar previamente que se cuente con el compilador de Java *JDK*, caso contrario se lo puede obtener e instalar desde el sitio web de Oracle, a través del siguiente URL: <https://www.oracle.com/technetwork/es/java/javase/downloads/index.html>

Una vez que se ha verificado que se cuente con JDK o se haya realizado la respectiva instalación, nos redirigirá hacia la página de Android Studio para realizar la descarga del instalador en el siguiente enlace:

<https://developer.android.com/studio#downloads>

Luego que se haya realizado la descarga del instalador, se continuará con los siguientes pasos para su instalación:

- Primero se debe ejecutar el instalador de Android Studio como se observa en la Figura 53.

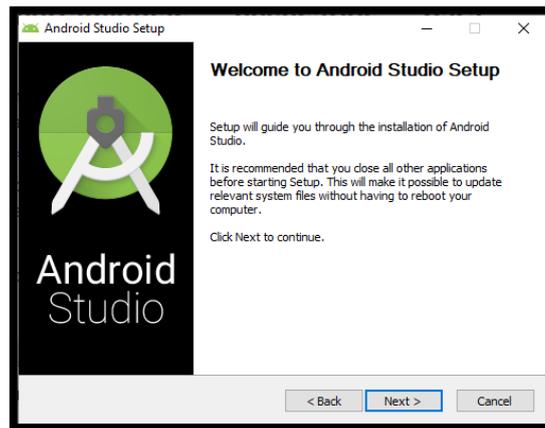


Figura 53. Mensaje de bienvenida instalación Android Studio.

- Luego por defecto se debe seleccionar *Next*, luego nos aparecerá la ubicación en donde se instalará la aplicación, también la dejamos por defecto como se observa en la Figura 54.

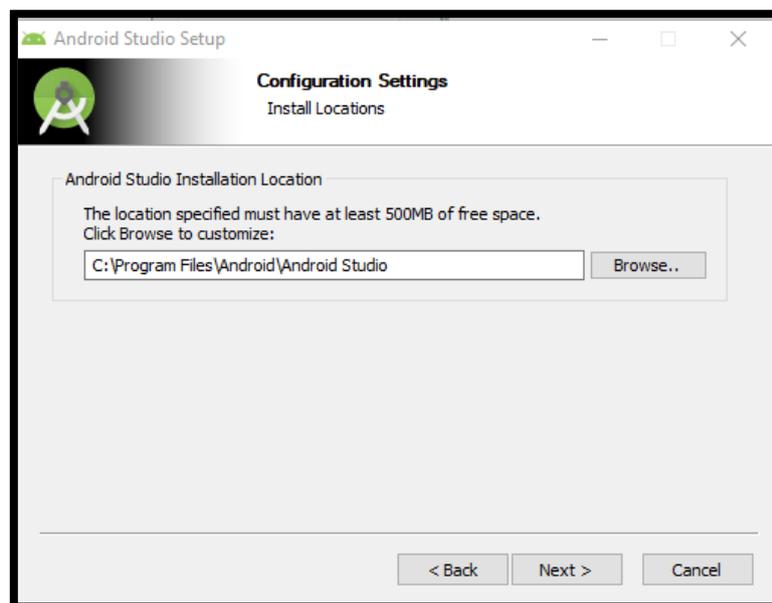


Figura 54. Ubicación de instalación Android Studio.

- Finalmente, cuando haya concluido su instalación, se procederá a iniciar la aplicación y aparecerá una ventana para crear un nuevo proyecto como se puede observar en la Figura 55.

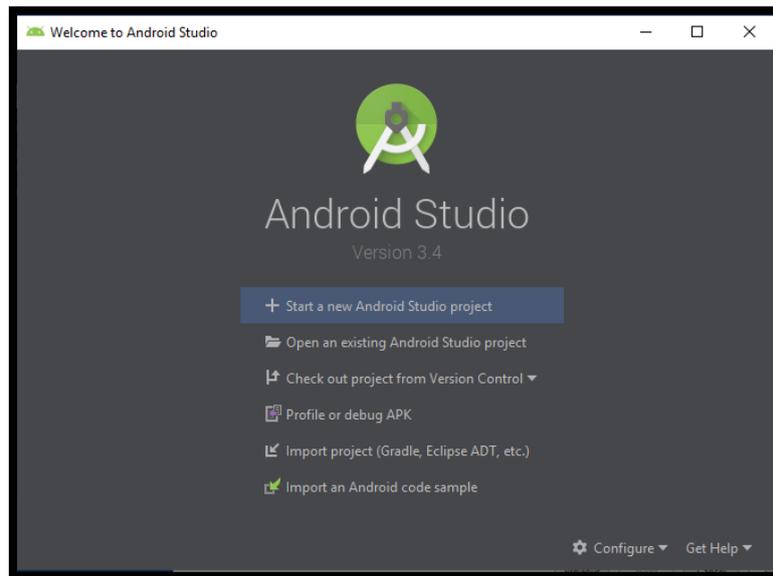


Figura 55. Inicio Android Studio, creación de proyecto.

4.5.2. Desarrollo de la aplicación móvil del proyecto

En esta sección se presentará los componentes necesarios para el desarrollo del aplicativo móvil con el uso de la herramienta Android Studio.

En la Figura 56 se presenta la codificación utilizada para realizar la conexión entre la base de datos para obtener los datos y la aplicación móvil.

```

public class ConexionDB {

    public static Connection ConnectionHelper() {

        StrictMode.ThreadPolicy policy = new StrictMode.ThreadPolicy.Builder().permitAll().build();
        StrictMode.setThreadPolicy(policy);
        Connection connection = null;
        String ConnectionURL = null;

        try {
            Class.forName("net.sourceforge.jtds.jdbc.Driver");
            //ConnectionURL = "jdbc:jtds:sqlserver://172.45.93.130;databaseName=TESIS;user=sa;password=Pacho@lejoi408;";
            ConnectionURL = "jdbc:jtds:sqlserver://bi-oidbserver.database.windows.net;databaseName=BI-OID_db;user=vm;password=Udl@2020;";
            connection = DriverManager.getConnection(ConnectionURL);
        } catch (SQLException se) {
            Log.e( tag: "ERROR", se.getMessage());
        } catch (ClassNotFoundException e) {
            Log.e( tag: "ERROR", e.getMessage());
        } catch (Exception e) {
            Log.e( tag: "ERROR", e.getMessage());
        }
        return connection;
    }
}

```

Figura 56. Conexión con Base de Datos SQL Server en Azure.

En la Figura 57 se puede observar el código necesario para establecer conexión con el servidor de correo en este caso con Outlook, el mismo que permitirá el envío y recepción de correos por parte de los usuarios desde el aplicativo web, así como desde la aplicación móvil.

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_correo);
    xCorreo = "admin@bi-oil.ec";
    //xCorreo = (EditText) findViewById(R.id.txtCorreoDestino);
    xAsunto = (EditText) findViewById(R.id.txtAsunto);
    xMensaje = (EditText) findViewById(R.id.txtMensaje);
    xEnvio = (TextView) findViewById(R.id.txtEnviarCorreo);
    //xEnviar = (Button) findViewById(R.id.btnEnviar);
    //xEnviar.setOnClickListener(this);
    xEnvio.setOnClickListener(this);
    xRegresar = (TextView) findViewById(R.id.txtRegresar);

    xRegresar.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intentReg = new Intent( packageContext: Correo.this, Usuario.class);
            Correo.this.startActivity(intentReg);
        }
    });
}
```

Figura 57. Código sincronización con correo electrónico para envío de notificaciones.

4.5.3. Vistas aplicación móvil

- **Página inicial de inicio de sesión**

En la Figura 58 se puede observar la vista inicial al momento que la aplicación móvil es abierta desde un *smartphone* Android.

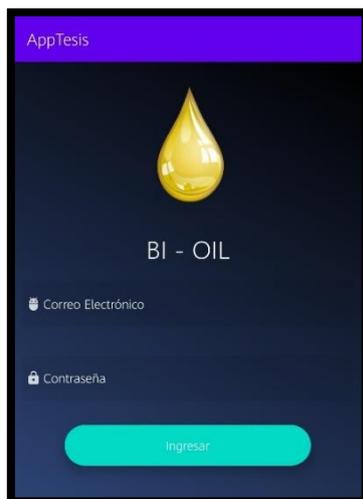


Figura 58. Vista inicial aplicación.

- **Vista para recuperación de contraseña**

En la Figura 59 se puede observar que el usuario final tiene la opción de regenerar su contraseña ingresando su usuario es decir su correo electrónico y escribiendo la nueva contraseña.



Figura 59. Vista Recuperación de contraseña.

- **Vista para generar gráficas o notificaciones**

En la Figura 60 se presenta la vista del aplicativo móvil que realiza la actualización de los gráficos del estado del contenedor, así como también

permite la opción de enviar una notificación mediante correo electrónico directo a los técnicos o administradores del sistema

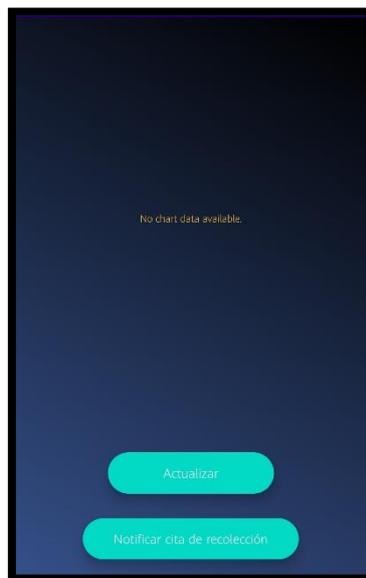


Figura 60. Vista generar gráfica o generar notificación de correo electrónico.

- **Vista correspondiente a la gráfica**

En la Figura 61 se puede observar la pestaña de gráficos diseñada para que el usuario pueda realizar un seguimiento del estado del contenedor.

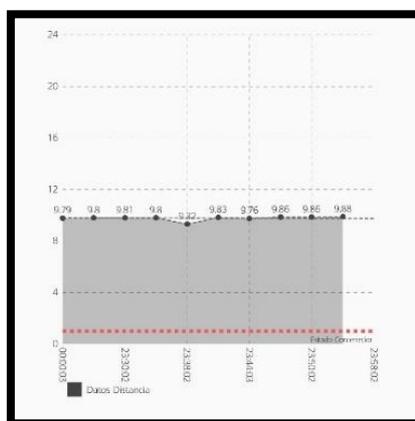


Figura 61. Vista correspondiente a gráfica de monitoreo.

- **Vista para enviar correo de notificación**

En la Figura 62 se puede observar la vista para envío de notificaciones diseñada para que el usuario ingrese su nombre de usuario y el mensaje que desea que el gestor conozca y mediante correo electrónico se procederá a enviar la notificación.

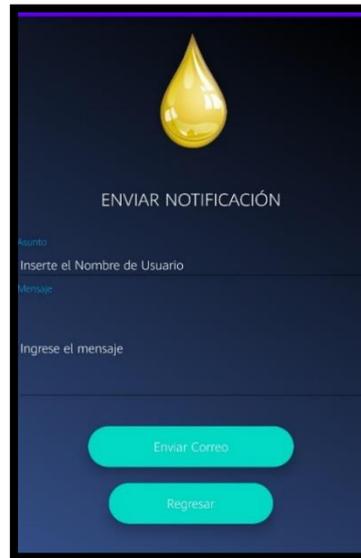


Figura 62. Vista correspondiente a envío de notificación.

5. PRUEBAS DE FUNCIONAMIENTO DEL SISTEMA

El siguiente capítulo se presenta el funcionamiento adecuado del sistema propuesto y como se cumplieron los objetivos que fueron planteados.

5.1. Funcionamiento del servidor

En la Tabla 12 se presentan los requerimientos que hacen relación con el servidor, en el cual se tiene que el servidor tiene una comunicación directa con el sensor de distancia HC-SR04 que realiza la medición dentro del contenedor para conocer el estado del líquido, de igual manera tiene una conexión con la base de datos para poder enviar los datos que se recojan desde el sensor, para ser almacenados en la nube, tiene una tarjeta inalámbrica para que se pueda conectar a internet.

Tabla 12

Requerimientos del servidor.

N°	Detalle Requerimiento
1	Poseer pines de entrada y salida para conexión de sensor
2	Permitir una comunicación entre el sensor y el servidor
3	Permitir comunicación hacia la Base de Datos
4	Permitir generación de informes en base a tiempos diferentes
5	Poseer una tarjeta inalámbrica
6	Permitir la ejecución de scripts o programas

En la Figura 63, se puede observar que el servidor cumple con el Requerimiento 1, en donde se observa que existen los pines para que se pueda establecer la comunicación respectiva entre el servidor y el sensor. De igual manera en la Figura 64 se muestra el cumplimiento del requerimiento 3 donde se puede verificar la comunicación entre el sensor con la BDD, se puede visualizar los datos registrados cada 2 minutos en la fecha 22/06/2020 de 22h00 a 22h14.

Para los Requerimientos 2, 4, 5 y 6 finalmente se cumplen los requerimientos ya que el sensor genera los datos en el servidor, el cual envía los datos de distancia que se almacenan en el servidor de base de datos como se visualiza en la Figura 64, permite la generación de informes en base a las tareas programadas del Crontab y Cron, posee una tarjeta inalámbrica, la misma que permite la conexión hacia internet y de igual manera el sistema operativo Raspbian permite la ejecución de varios scripts simultáneos.

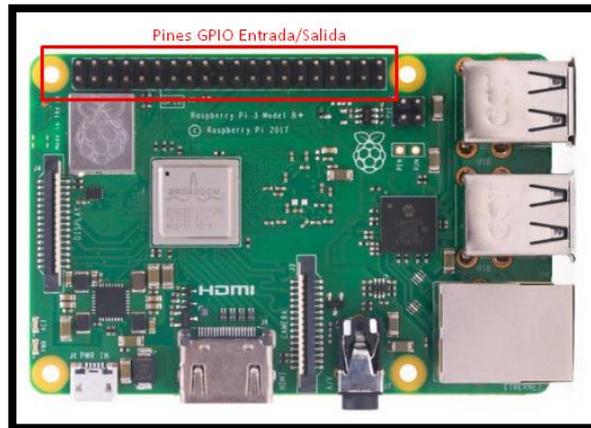


Figura 63. Raspberry Pi Pines Entrada/Salida GPIO.

SQLQuery1.sql - bi-...BI-OIL_db (vm (67))*

```
select * from Tesis2min order by IdTesis2 desc
```

100 %

Results Messages

	IdTesis2	Fecha	Hora	Distancia
109	1609	2020-06-22	22:14:02.0000000	8,68698672590585
110	1608	2020-06-22	22:12:03.0000000	8,64588007433661
111	1607	2020-06-22	22:10:03.0000000	9,08983191128435
112	1606	2020-06-22	22:08:03.0000000	8,69109739106277
113	1605	2020-06-22	22:06:04.0000000	9,07749991581358
114	1604	2020-06-22	22:04:03.0000000	8,67054406527815
115	1603	2020-06-22	22:02:03.0000000	9,11449590222589
116	1602	2020-06-22	22:00:04.0000000	6,08493568157328

Figura 64. Comunicación entre sensor y base de datos.

5.2. Base de Datos

En esta sección se realizará la validación del funcionamiento de la base de datos para el funcionamiento correcto del sistema de monitoreo en el cual se evidencia el cumplimiento de los requerimientos que se los puede observar en la Tabla 13, los mismos que corresponden a los detallados en la sección 3.1.3.

Tabla 13

Requerimientos servidor Base de Datos.

N°	Detalle Requerimiento
1	Permitir conexión con el Servidor
2	Permitir almacenar la información enviada del servidor
3	Permitir la publicación de datos almacenados
4	Permitir adaptabilidad con aplicaciones web y móviles

En la Figura 65, se puede visualizar la base de datos que se encuentra implementada en la que cumplen los requerimientos 1 y 2, en la cual se observa las distancias almacenadas que se envían directamente desde el servidor por medio del sensor implementado en el contenedor.

Para validar los requerimientos 3 y 4, se lo puede visualizar desde la página web y móvil las mismas que se encuentran publicadas gracias a la facilidad de la integración con ASP.NET y Android Studio.

IdTesis2	Fecha	Hora	Distancia	DistanciaNorm
109	2020-06-22	22:14:02.0000000	8.68698672590585	8.68698672590585
110	2020-06-22	22:12:03.0000000	8.64588007433661	8.64588007433661
111	2020-06-22	22:10:03.0000000	9.08983191128435	9.08983191128435
112	2020-06-22	22:08:03.0000000	8.69109739106277	8.69109739106277
113	2020-06-22	22:06:04.0000000	9.07749991581358	9.07749991581358
114	2020-06-22	22:04:03.0000000	8.67054406527815	9.11449590222589
115	2020-06-22	22:02:03.0000000	9.11449590222589	9.11449590222589
116	2020-06-22	22:00:04.0000000	6.08493568157328	9.9037436123552
117	2020-06-22	21:58:02.0000000	8.67465473043508	9.9037436123552
118	2020-06-22	21:54:03.0000000	9.9037436123552	9.9037436123552
119	2020-06-14	00:00:03.0000000	9.74753833639211	9.74753833639211
120	2020-06-13	23:58:02.0000000	8.87907962141366	9.87907962141366
121	2020-06-13	23:54:02.0000000	9.86263696078596	9.86263696078596
122	2020-06-13	23:50:02.0000000	9.86263696078596	9.86263696078596
123	2020-06-13	23:46:03.0000000	9.75870331862388	9.75870331862388
124	2020-06-13	23:44:03.0000000	9.82564097437365	9.82564097437365
125	2020-06-13	23:40:02.0000000	9.31591849491514	9.31591849491514
126	2020-06-13	23:38:02.0000000	9.80097698343211	9.80097698343211
127	2020-06-13	23:34:03.0000000	9.80919831374596	9.80919831374596
128	2020-06-13	23:30:02.0000000	9.79686631827519	9.79686631827519
129	2020-06-13	23:28:02.0000000	9.79275565311827	9.79275565311827
130	2020-06-13	23:26:02.0000000	9.86263696078596	9.86263696078596
131	2020-06-13	23:24:02.0000000	9.73520634092134	9.73520634092134

Figura 65. Base datos SQL Server.

5.3. Pruebas de funcionamiento fase de monitoreo y presentación de datos

En la siguiente sección se procederá con la validación del funcionamiento tanto de la fase de monitoreo como presentación de datos.

5.3.1. Pruebas de verificación del aplicativo web

En esta sección se valida el funcionamiento correcto del aplicativo web con respecto a los requerimientos funcionales que se presentaron en el capítulo 3, en la sección 3.3.1.1 A continuación, se visualiza los requerimientos respectivo:

Tabla 14

Requerimientos funcionales aplicativo web.

RF	DESCRIPCION
RF01	La aplicación web manejará tres roles diferentes: administrador, técnico, usuario final.
RF02	La aplicación únicamente permitirá registrar usuarios con el perfil de administrador.
RF03	La aplicación permitirá que el usuario pueda recuperar su contraseña.
RF04	Los roles y los usuarios serán modificables únicamente por el administrador.
RF05	En la aplicación se podrá visualizar los informes emitidos por el sensor.
RF06	La aplicación generará notificaciones de alerta para un monitoreo más eficaz.
RF07	La aplicación permitirá enviar notificaciones tanto al usuario registrado como a técnicos y administrador.
RF08	La aplicación permitirá iniciar sesión con lo cual se identificará el rol del usuario que inicia sesión.
RF09	Se podrá finalizar sesión desde la aplicación.

5.3.1.1. Validación de requerimientos funcionales

En la Figura 66, se puede observar el cumplimiento del requerimiento funcional 1 en donde se valida los roles aplicados dentro del aplicativo web.

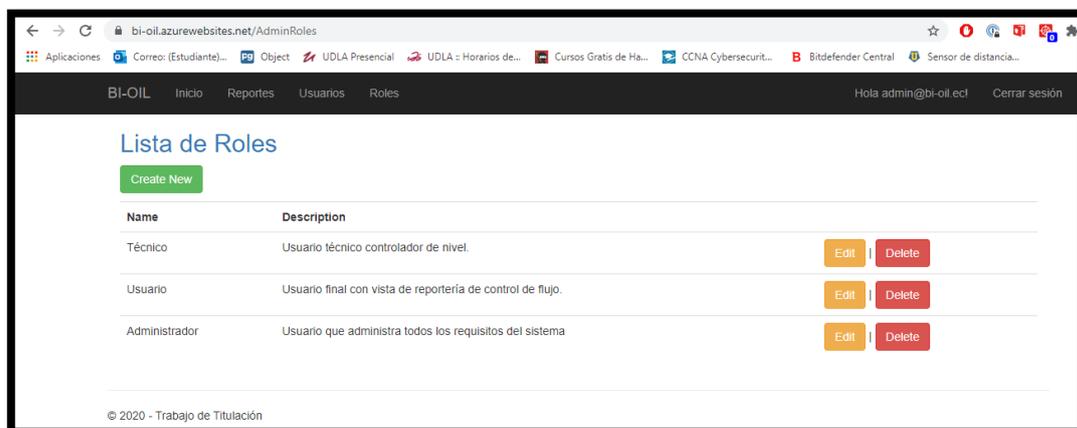


Figura 66. Roles del aplicativo web.

De acuerdo a la Figura 67, se valida el cumplimiento del requerimiento funcional 2 el cual con el perfil de Administrador puede crear usuarios y en la Figura 68 con un perfil de Técnico se puede validar que no permite la creación de usuarios.

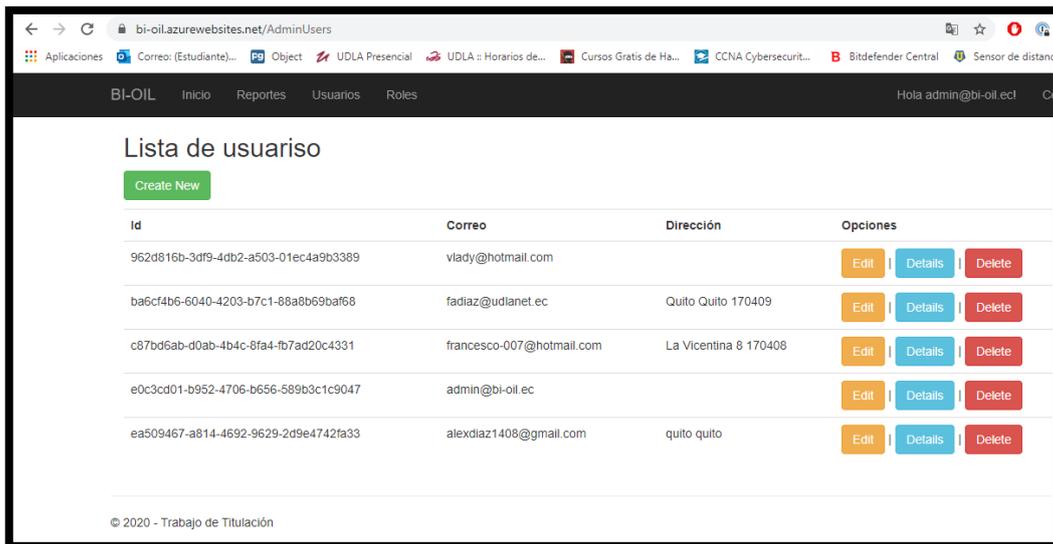


Figura 67. Vista usuarios con perfil de administrador.

En la Figura 68 se puede observar que un usuario con perfil de técnico tiene el acceso limitado a las funcionalidades del aplicativo web.

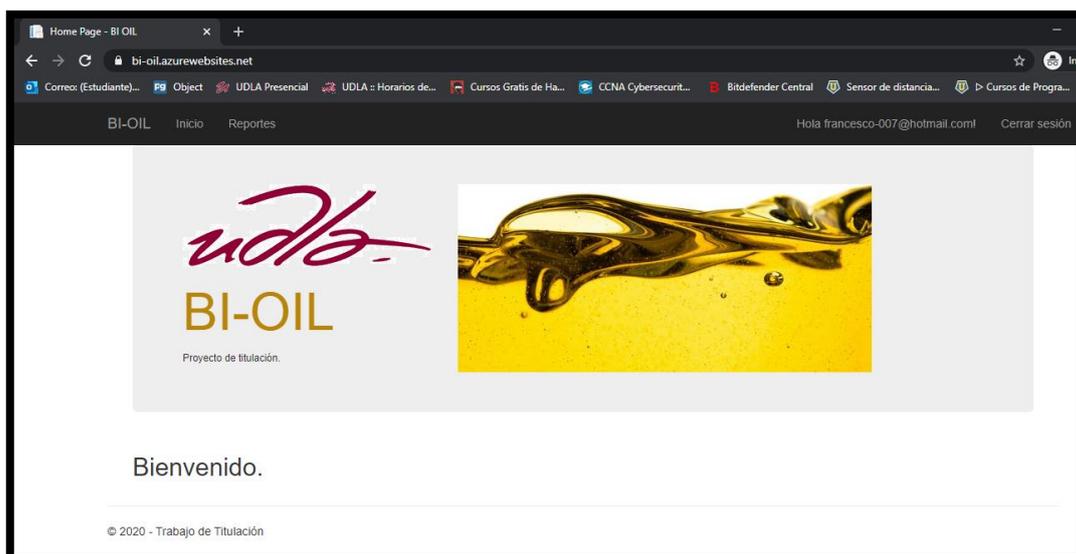


Figura 68. Vista página con perfil de Técnico.

El requerimiento funcional 4 se valida en base a la Figura 69 en la cual el usuario Administrador al momento de la creación del usuario puede establecer el rol al cual pertenecerá.

bi-oil.azurewebsites.net/AdminUsers/Create

Aplicaciones | Correo: (Estudiante)... | Object | UDLA Presencial | UDLA - Horarios de... | Cursos Gratis de Ha... | CCNA Cybersecurit... | Bitdefender Central | Sensor de distancia...

BI-OIL | Inicio | Reportes | Usuarios | Roles | Hola admin@bi-oil.ec | Cerrar sesión

Create

ApplicationUser

Email

Direccion

Departamento

Codigo Postal

Password

Confirmar password

Seleccione los Roles Técnico Usuario Administrador

© 2020 - Trabajo de Titulación

Figura 69. Creación de usuario y asignación de roles perfil administrador.

Para la validación del requerimiento funcional 5, se puede observar en la Figura 70 en la cual se muestra la generación de informes que se presenta dentro de la aplicación web con la obtención de los datos que se generan en el servidor con la distancia entre el líquido y el sensor. De igual manera en la Figura 71 se puede observar como el sensor fue recolectando datos el 28 de junio a las 17:35 con un valor de cero, luego de dos minutos se agrega una cantidad de líquido y el sensor muestra nuevos datos que la distancia de la base del contenedor hasta la superficie del líquido son aprox. 5 centímetros, luego a las 17:45 le agrega un poco más de líquido y se obtiene una nueva lectura. Más tarde luego de 1 hora 15 minutos se ha ido agregando líquido y el sensor ha ido mostrando los datos obtenidos, a las 19:00 se retira cierta cantidad de líquido y el sensor toma lectura de la distancia de manera correcta; hasta las 19:45 se vuelve a agregar líquido y se obtienen datos crecientes en la gráfica mostrada en donde a esta hora se vuelve a retirar un poco de líquido y se vuelve a tomar lectura en la cual se observa que se obtiene un decrecimiento en la gráfica de monitoreo. A partir de las 20:00 se mantiene estable la gráfica ya que no se agrega más líquido y los valores se mantienen constante.

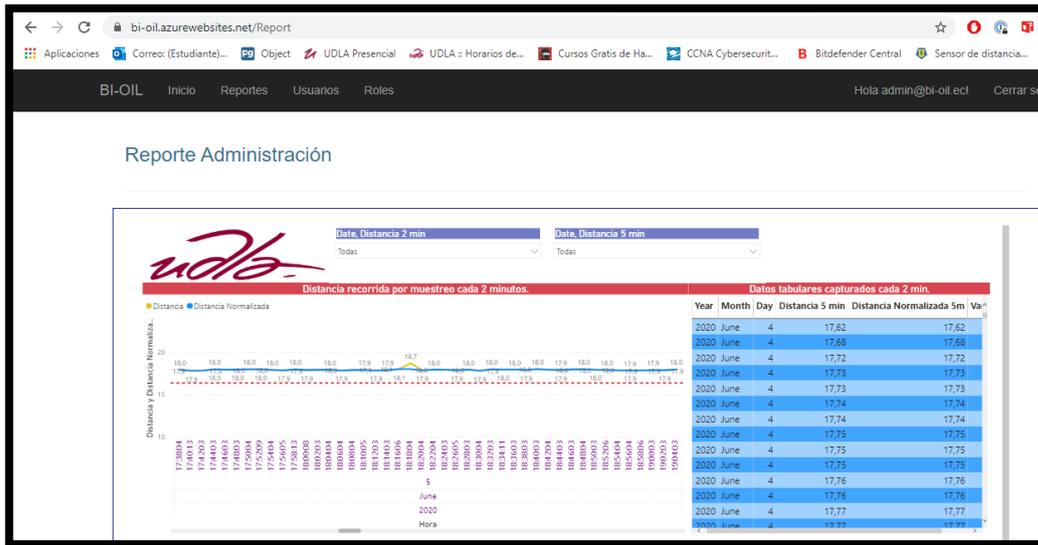


Figura 70. Informe de aplicación web.



Figura 71. Lectura de datos de contenedor desde cero.

Para el requerimiento funcional 6 y 7 se puede validar en la Figura 72 como llegan las notificaciones al correo del usuario del estado del líquido en el contenedor, mientras que en la Figura 73 se puede observar cómo llegan las notificaciones al correo del Administrador con relación al cambio de líquido en el contenedor del usuario.

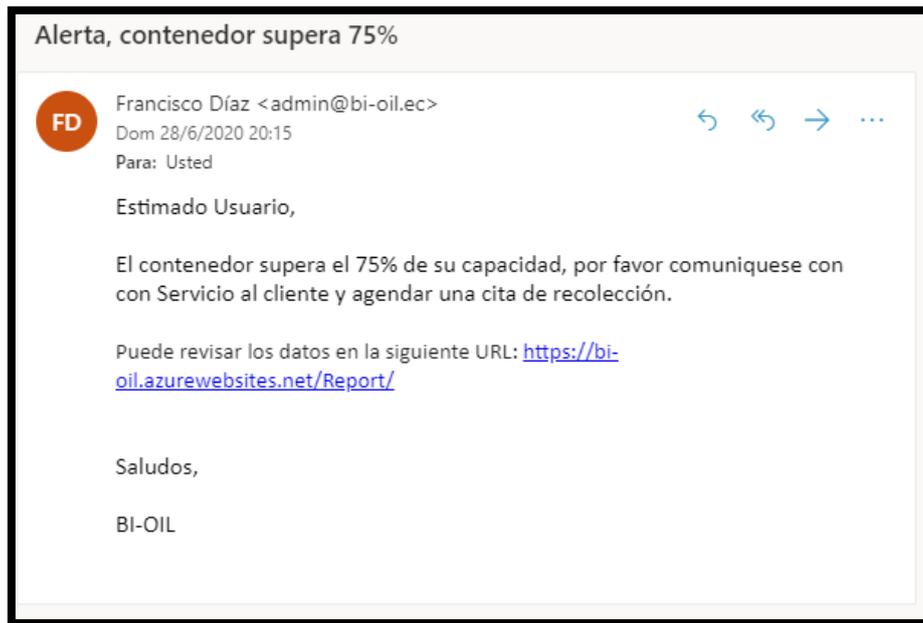


Figura 72. Notificación mediante correo electrónico a usuario.



Figura 73. Notificación a Administrador mediante correo electrónico.

En la Figura 74, se observa que para acceder al aplicativo web se debe autenticar con usuario asignado y contraseña enviada por el administrador, con lo cual se valida el requerimiento funcional 8.

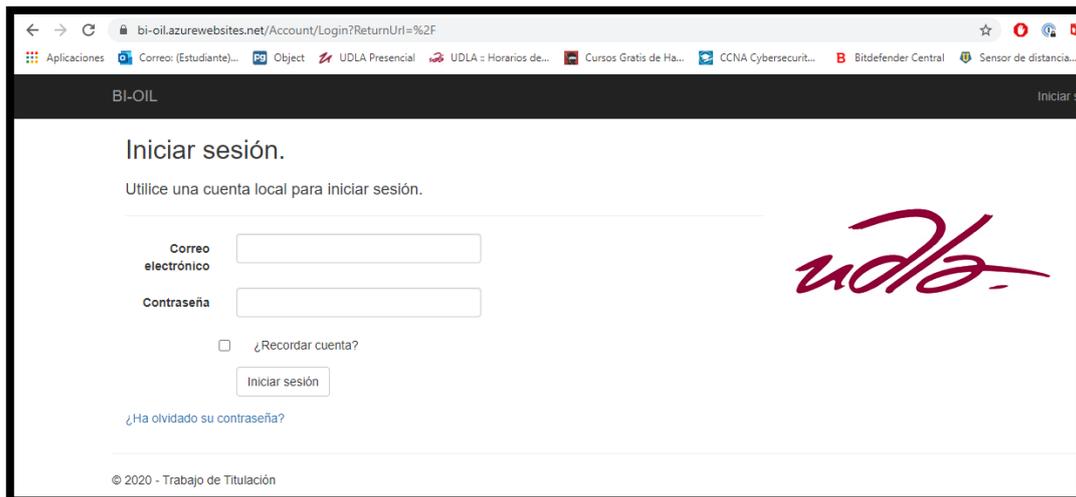


Figura 74. Inicio de sesión aplicación web.

Finalmente, para validar el requerimiento funcional 9 se puede observar en la Figura 75 la esquina superior derecha que la aplicación web permite el cierre de sesión.

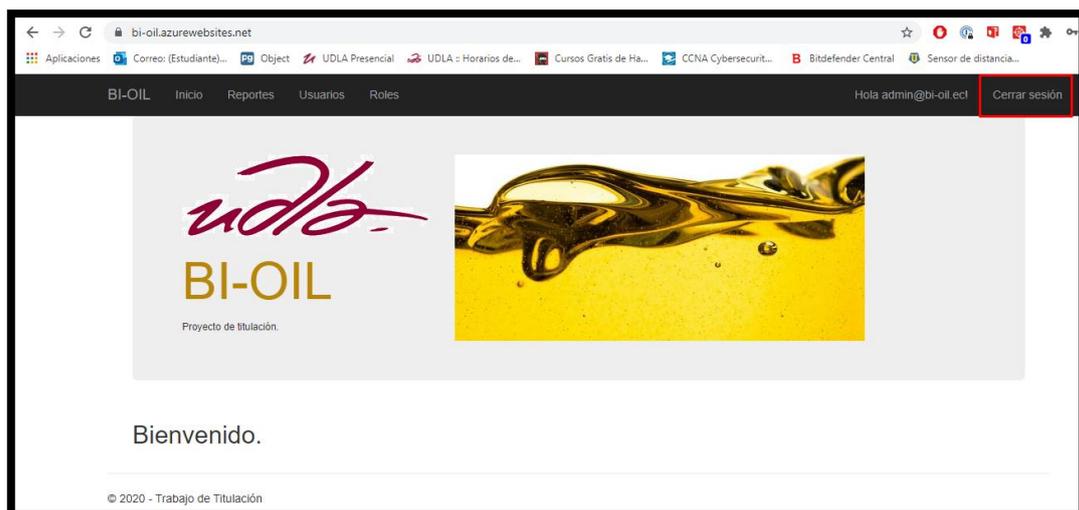


Figura 75. Vista aplicación web con opción de cierre de sesión.

5.3.2. Pruebas de verificación del aplicativo móvil en Android

En esta sección valida el funcionamiento correcto relacionado a la Aplicación Móvil que es desarrollada para Android con respecto a los requerimientos funcionales que se presentaron en el capítulo 3, subsección 3.4.3.1.

En la Tabla 15 se podrá visualizar la funcionabilidad de los requerimientos necesarios para la implementación del aplicativo móvil.

Tabla 15

Requerimiento funcional Aplicativo Móvil.

RF	DESCRIPCIÓN
RF01	La aplicación móvil solo manejará el rol de usuario final.
RF02	El rol de usuario permitirá ingresar al sistema, previo inicio de sesión.
RF03	La aplicación permitirá verificar los resultados de estado del contenedor.
RF04	La aplicación permitirá enviar notificación a los técnicos y administradores mediante correo electrónico.

5.3.2.1. Validación de requerimientos funcionales

En la Figura 76 se puede observar que para poder acceder a la aplicación móvil se lo deber realizar mediante el ingreso de un usuario y contraseña, por lo que se cumple con los requerimientos funcionales 1 y 2 en los cuales el usuario puede establecer sesión con el correo electrónico y clave proporcionada por el Administrador.

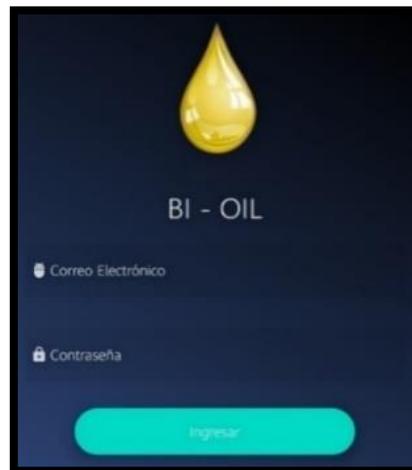


Figura 76. Inicio de Sesión Aplicación móvil.

En esta Figura 77 se observa como la aplicación permite al usuario final escoger entre la vista del informe o enviar una notificación. Por otro lado, en la Figura 78 se puede validar el requerimiento funcional 3 acerca de la generación de informe con datos en tiempo real, en la cual se muestra el comportamiento del líquido dentro del contenedor en vivo, es decir en la aplicación se reflejarán los últimos diez datos que se obtiene desde el sensor los cuales son mostrados en una gráfica lineal.

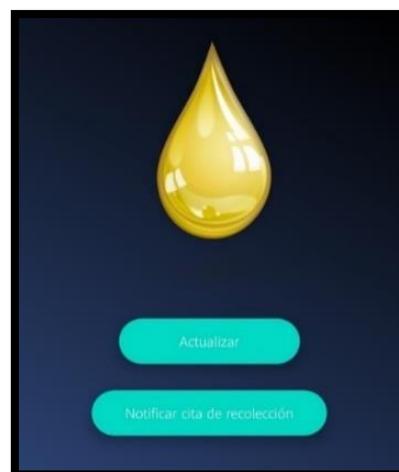


Figura 77. Sección para visualizar informe o enviar notificación.

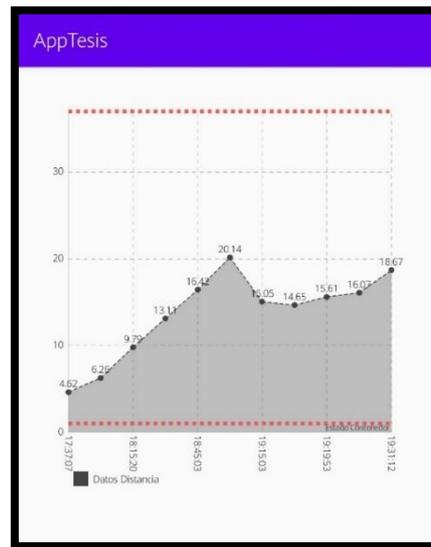


Figura 78. Presentación de informe en tiempo real de aplicación móvil.

Finalmente, en la Figura 79 se visualiza la manera como envía notificaciones la aplicación mediante correo electrónico hacia los administradores de la plataforma con el fin de generar una cita o notificar algún inconveniente. Por otro lado, en la Figura 80 se puede observar cómo llena la notificación por parte del usuario al correo electrónico de Administración.

ENVIAR NOTIFICACIÓN

Asunto
 Usuario: Francisco Díaz

Mensaje
 Buenas noches,
 Por favor su gentil ayuda con la recolección del contenedor,
 el mismo se encuentra lleno.
 Saludos.

Enviar Correo
Regresar
Mensaje enviado

Figura 79. Vista aplicación móvil para envío de notificaciones al sistema.

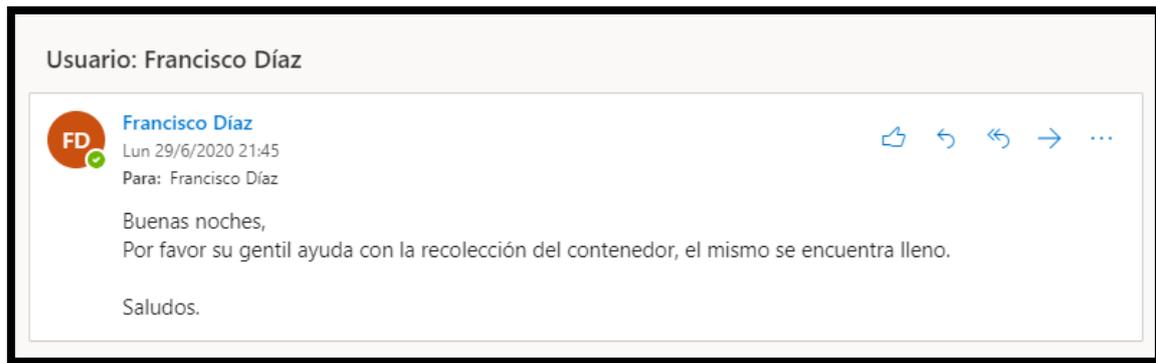


Figura 80. Notificación recibida en correo electrónico de Administrador.

6. CONCLUSIONES Y RECOMENDACIONES

6.1. Conclusiones

- El presente proyecto consiste en la implementación de un sistema de adquisición y monitoreo que permite medir los niveles de desechos de aceite vegetal de uso doméstico por medio de un sistema de adquisición de datos compuesto por un sensor ultrasónico y un microcontrolador que luego de procesar envía los datos a una BD alojada en la nube para luego ser presentados al usuario a través de un aplicativo web. Los valores obtenidos se presentan en tiempo real, lo que permite optimizar y tomar decisiones oportunas al momento de establecer una cita de recolección de los residuos de aceite vegetal usado.
- El desarrollo del sistema automatizado permitirá que los establecimientos de bajo impacto ambiental que preparan alimentos en especial de frituras puedan gestionar de mejor manera la recolección de desechos de aceite vegetal, así evitar la evacuación de estos desperdicios por cañerías y contribuir con la protección del Medio Ambiente.
- Mediante el sistema de monitoreo los gestores de recolección pueden evitar las visitas innecesarias a los establecimientos de sus clientes, ya que podrán visualizar el estado de los contenedores y planificar de una manera ordenada las visitas de recolección.
- El uso de un servidor de base de datos SQL permitió mayor compatibilidad con el desarrollo del proyecto, ya que este se sincronizó de manera adecuada con el programa realizado en Python en el servidor (Raspberry pi 3 B+) y la extracción de datos fue óptima para la presentación de este en la aplicación web y móvil.
- ASP.NET permitió que el proyecto se desarrolle de una manera esperada debido a su alta compatibilidad con la base de datos, aplicación móvil y lo más importante en cualquier navegador que se quiera iniciar la aplicación.
- Se puede asegurar que los valores almacenados en el servidor de Datos serán íntegros ya que mediante reglas de *firewall* que se configuran en el

portal de Azure únicamente se establecen direcciones IP conocidas para acceso total a la información.

- El uso del servidor de datos SQL Server en Azure está sobre la nube de Microsoft, permite trabajar en tiempo real con los datos, además que los datos se encuentran organizados, se puede establecer sentencias SQL que permitieron desarrollar el trabajo con mayor facilidad.
- En la puesta a marcha del servidor de datos y el aplicativo no se tuvo que hacer uso de hardware, ya que gracias al portal de Azure permite mantener un hosting en la nube al igual que la base de datos. Esto es de gran importancia al momento que se publicó el aplicativo web, lo que permitió el proyecto no solo de manera local sino desde cualquier lugar.
- Para la implementación de informes técnicos de gráficos, se tomó la decisión de normalizar los datos obtenidos con el fin de presentar un gráfico más estable y entendible a la vista del usuario, sin descuidar los datos en bruto que se obtienen directo desde el sensor con el fin de validar el crecimiento del líquido dentro del contenedor. El porcentaje de normalización que se eligió es de +/- 0,99 cm con el fin de llevar la gráfica estable.
- Trabajar con un sensor como el ultrasónico permitió una mayor estabilidad de los datos que se obtienen, ya que se emite una onda de sonido y mediante un cálculo matemático se puede obtener un valor real de la distancia con el que se puede trabajar de una manera correcta y tener un alto grado de certeza en lo que ocurre dentro del contenedor con el líquido.
- El usuario final cuenta con acceso desde su dispositivo móvil, el cual le permite visualizar el estado del contenedor en tiempo real, así como enviar notificaciones al sistema las mismas que llegarán a la cuenta de correo del administrador.
- Para la creación e implementación del proyecto se hizo uso de los recursos que el portal de Azure brinda que es un bono de \$200 que proporciona Microsoft en la cuenta de manera gratuita por un año, lo cual permitió un ahorro en los costos de operatividad (Base de datos y

Hosting), de igual manera teniendo una garantía en la integridad del desarrollo.

- Las notificaciones mediante correo electrónico que se encuentran implementadas en el sistema permiten que el usuario final pueda conocer que la capacidad del contenedor está llegando a su límite y las notificaciones que recibe el administrador muestra si hay una variación del líquido que se encuentra dentro del contenedor.
- El proyecto no solamente puede ser ejecutado para la medición de residuos de aceite vegetal, sino también puede ser empleados para cualquier tipo de líquidos que sean almacenados en un contenedor y se necesite conocer el estado de estos de una manera remota y en tiempo real.
- De igual manera el proyecto consta de un amplio estudio para un posible escalamiento en sus diferentes módulos, como puede ser generación de módulos de citas, temperatura del contenedor, alarma de verificación en caso de que el sensor sea removido, etc.

6.2. Recomendaciones

- Con el avance tecnológico se recomienda hacer uso de aplicaciones, bases de datos que se encuentren en la nube, ya que con esto se evita el uso de hardware y se tiene una accesibilidad desde cualquier lugar y no tener inconvenientes como el uso de una IP pública que puede llegar a ser más costosa que todo el proyecto realizado.
- Se recomienda que el servidor Raspberry Pi 3 B+ trabaje sus scripts con el lenguaje Python, ya que este viene pre-instalado por defecto en el sistema operativo, haciendo falta una actualización de la versión.
- En situaciones como la que pasar un cable ethernet para la conexión del servidor con internet resulta dificultoso, se recomienda el uso de tecnologías como Wi-Fi que permite la comunicación de manera eficiente y nos libra del uso de cables.

- Mantener tareas programadas para tener un mejor control de lo que sucede con el dispositivo y se puede presentar mediante gráficas para llevar controles fiables.
- Para una mayor confianza del sitio que se desarrolla es recomendable el uso de certificados SSL, ya que mediante estos certificados se avala que la página web a la que se visita es real y se da un plus adicional con el usuario final.
- Es recomendable mantener los componentes de los proyectos como base de datos, desarrollo del aplicativo web y móvil de manera ordenada con el fin de cumplir estándares y brindar mayor eficiencia en el código y que al momento de publicar las aplicaciones no se tenga errores o a su vez sea fácil identificar dicha fuente.

REFERENCIAS

- Alvarez, R. (2016). *Manual de PHP - Descargar*. 1–142. <http://manual-de-php.uptodown.com/>
- Castañeda Solís, J. I. (2018). *Análisis de la generación de residuos de aceite de uso domiciliario para promover la correcta gestión y sensibilización ambiental de la población del Cantón Esmeraldas* (Tesis de maestría). Pontificia Universidad Católica de Ecuador – Sede Esmeraldas.
- Centro de Apoyo Tecnológico a Emprendedores, F. P., & Albacete, C. y T. de. (2013). *Manual de uso avanzado de la aplicación*. 1–74. https://www.bilib.es/fileadmin/user_upload/oficinamovil/Documentos/Ofimatrica/Googledrive/manualavanzado/Google_Drive_-_Manual_avanzado.pdf
- CONSULTORA, A. (2014). *Secretaría de Ambiente Municipio de Quito*.
- Danysoft. (2017). *Asp.net*. 1–4.
- DIY ROBOT KEYES. (2013). *Water Sensor Module User ' s Manual*. 1–3. <https://pandoralab.com.br/wp-content/uploads/2016/01/sensor-nivel-agua.pdf>
- Gems Sensors. (2014). *LS-7 Series (Plastic) Compact , Side-Mounted Level Switches*. 78803(78803), 2–3.
- IBM Corporation. (2009). *Install and configure e-mail server and client*. 1–15.
- Indoware. (2013). *Ultrasonic Ranging Module HC - SR04. Datasheet*, 1–4. <http://www.micropik.com/PDF/HCSR04.pdf>
- Intel. (2018). *Intel ® NUC Board / Kit NUC7i7DN. February*. https://www.intel.com/content/dam/support/us/en/documents/mini-pcs/nuc-kits/NUC7i7DN_TechProdSpec.pdf

Jadhav, M. A., Sawant, B. R., Deshmukh, A., Mumbai, N., Jain, N., Mangal, P., Mehta, D., Basalo, A., Álvarez, M. Á., Hurtado, P., Cerdá, X. J., Wahlin, D., & Cuartero, R. G. (2015). Manual de Angular JS. *JavaScript Library*, 5(3), 2876–2879.

http://oa.upm.es/44420/3/TFG_RAFAEL_GUTIERREZ_CUARTERO.pdf
<http://www.jgrcs.info/index.php/jgrcs/article/view/952>
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.736.4771&rep=rep1&type=pdf>

Ley de Gestión Ambiental. (2004). *Glosario de definiciones*. 6(7–8), 53–55.

LIBRO, V. A. 6. *Norma de Calidad Ambiental para el Manejo y Disposición Final de Desechos Sólidos no Peligrosos* [en línea]. 2015. SI: sn [Consulta: 20 mayo 2020].

Lock, K. C., & Mount, V. (2020). *ThinkCentre M710 Tiny*.

López Montenegro, S. A. (2014). *Formulación del plan estratégico 2014-2018 para el mejoramiento de la gestión empresarial de la empresa pública metropolitana de gestión integral de residuos sólidos emgirs ep del distrito metropolitano de Quito* (Tesis de maestría). Universidad Israel

Microsoft. (2018). *Curso intensivo de Office 365* :

Microsoft. (2018). *Microsoft SQL Server 2019 White Paper. September 2018*.
<https://info.microsoft.com/rs/157-GQE-382/images/EN-US-CNTNT-white-paper-DBMod-Microsoft-SQL-Server-2019-Technical-white-paper.pdf>

Moyococha, S. (2016). *Estudio De Impacto Ambienta Expost “Sistema De Gestión Integral De Desechos Hospitalarios Del Distrito Metropolitano De Quito.”* 1–240.

Mysql. (2013). MySQL 5 . 0 Reference Manual. *MySQL 5.0 Reference Manual*, 1, 1692. dev.mysql.com

Raspberry Pi Foundation. (2016). Raspberry Pi 3 Model B+ Datasheet.

Datasheet, 5. <https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-Model-Bplus-Product-Brief.pdf>

Sarno, M., & Iuliano, M. (2019). Biodiesel production from waste cooking oil. *Green Processing and Synthesis*, 8(1), 828–836. <https://doi.org/10.1515/gps-2019-0053>

Solanki, D. (2018). Aprendizaje Firebase. *Contributors, STACK OVERFLOW*, 79. <https://riptutorial.com/Download/firebase-es.pdf>

ANEXOS

ANEXO 1

Código Raspberry

Código utilizado para la comunicación entre el servidor Raspberry y el sensor ultrasónico HC-SR04.

```
import RPi.GPIO as GPIO

import time

import sys

import os

from datetime import date

from datetime import datetime

from os import getenv

import pymysql

import subprocess

#Permite la conexión con la base de datos

server = 'bi-oildbserver.database.windows.net'

database = 'BI-OIL_db'

username = 'vm@bi-oildbserver.database.windows.net'

password = 'Vlady1995.'

conn = pymysql.connect(server, username, password, database)

cursor = conn.cursor()
```

```
GPIO.setmode(GPIO.BCM)

GPIO.setwarnings(False)

# Establecimiento de los pines a ser utilizados

TRIG = 4

ECHO = 18

num = 2

GPIO.setup(TRIG,GPIO.OUT)

GPIO.setup(ECHO,GPIO.IN)

GPIO.output(TRIG, True)

time.sleep(0.00001)

GPIO.output(TRIG, False)

fecha = datetime.now().strftime("%x")

hora = datetime.now().strftime("%X")

while GPIO.input(ECHO) == False:

    start2min = time.time()

while GPIO.input(ECHO) == True:

    end2min = time.time()

sig_time2min = end2min - start2min

# Determinación de la distancia de l sensor en CM:
```

```
dist2min = sig_time2min / 0.000058
```

```
total2min = 47 - dist2min
```

```
cursor.execute("INSERT INTO Tesis2min (Fecha, Hora, Distancia) VALUES  
(%s', '%s', %s)" %(fecha, hora, total2min))
```

```
#query = ("INSERT INTO Tesis2min ")
```

```
#query = ("INSERT INTO  
billed_items(item_name,billed_qty,price,item_bill_series)  
VALUES(%s,%s,%s,%s)")
```

```
#c.execute(query,((name),(no),(price),(series))
```

```
conn.commit()
```

```
conn.close()
```

```
print('Total: {0:.2f}'.format(total2min))
```

```
print(sig_time2min)
```

```
print(start2min)
```

```
print('Fecha: ', fecha)
```

```
print('Hora: ', hora)
```

```
print('Distancia: {0:.2f} cm.'.format(dist2min))
```

```
GPIO.cleanup()
```

ANEXO 2

Código Aplicación Web

En este anexo se detalla el código utilizado para la realización del aplicativo web como tal.

```
using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

namespace Tesis3.Models

{

    public class ExternalLoginConfirmationViewModel

    {

        [Required]

        [Display(Name = "Correo electrónico")]

        public string Email { get; set; }

    }

    public class ExternalLoginListViewModel

    {

        public string returnUrl { get; set; }

    }

    public class SendCodeViewModel

    {

        public string SelectedProvider { get; set; }

        public ICollection<System.Web.Mvc.SelectListItem> Providers { get; set; }

    }

}
```

```
public string returnUrl { get; set; }

public bool RememberMe { get; set; }

}

public class VerifyCodeViewModel

{

    [Required]

    public string Provider { get; set; }

    [Required]

    [Display(Name = "Código")]

    public string Code { get; set; }

    public string returnUrl { get; set; }

    [Display(Name = "¿Recordar este explorador?")]

    public bool RememberBrowser { get; set; }

    public bool RememberMe { get; set; }

}

public class ForgotViewModel

{

    [Required]

    [Display(Name = "Correo electrónico")]
```

```
    public string Email { get; set; }  
  
}  
  
public class LoginViewModel  
  
{  
  
    [Required]  
  
    [Display(Name = "Correo electrónico")]  
  
    [EmailAddress]  
  
    public string Email { get; set; }  
  
    [Required]  
  
    [DataType(DataType.Password)]  
  
    [Display(Name = "Contraseña")]  
  
    public string Password { get; set; }  
  
    [Display(Name = "¿Recordar cuenta?")]  
  
    public bool RememberMe { get; set; }  
  
}  
  
public class RegisterViewModel  
  
{  
  
    [Required]  
  
    [EmailAddress]
```

```
[Display(Name = "Email")]
```

```
public string Email { get; set; }
```

```
[Required]
```

```
[StringLength(100, ErrorMessage = "La {0} deberia tener como minimo {2}  
de largo.", MinimumLength = 6)]
```

```
[DataType(DataType.Password)]
```

```
[Display(Name = "Password")]
```

```
public string Password { get; set; }
```

```
[DataType(DataType.Password)]
```

```
[Display(Name = "Confirmar password")]
```

```
[Compare("Password", ErrorMessage = "The password and confirmation  
password do not match.")]
```

```
public string ConfirmPassword { get; set; }
```

```
// Add the new address properties:
```

```
[Display(Name = "Direccion")]
```

```
public string Address { get; set; }
```

```
[Display(Name = "Ciudad")]
```

```
public string City { get; set; }
```

```
[Display(Name = "Departamento")]
```

```
public string State { get; set; }
```

```
// Use a sensible display name for views:

[Display(Name = "Codigo Postal")]

public string PostalCode { get; set; }

}

public class ResetPasswordViewModel

{

    [Required]

    [EmailAddress]

    [Display(Name = "Correo electrónico")]

    public string Email { get; set; }

    [Required]

    [StringLength(100, ErrorMessage = "El número de caracteres de {0} debe ser al menos {2}.", MinimumLength = 6)]

    [DataType(DataType.Password)]

    [Display(Name = "Contraseña")]

    public string Password { get; set; }

    [DataType(DataType.Password)]

    [Display(Name = "Confirmar contraseña")]

    [Compare("Password", ErrorMessage = "La contraseña y la contraseña de confirmación no coinciden.")]
```

```

        public string ConfirmPassword { get; set; }

        public string Code { get; set; }

    }

    public class ForgotPasswordViewModel

    {

        [Required]

        [EmailAddress]

        [Display(Name = "Correo electrónico")]

        public string Email { get; set; }

    }

}

```

Vista Administrador

Código que permite la realización de la vista principal del aplicativo web

```

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

using System.Web.Mvc;

namespace Tesis3.Models

{

    public class RoleViewModel

```

```
{  
  
    public string Id { get; set; }  
  
    [Required(AllowEmptyStrings = false)]  
  
    [Display(Name = "RoleName")]  
  
    public string Name { get; set; }  
  
    public string Description { get; set; }  
  
}  
  
public class EditUserViewModel  
  
{  
  
    public string Id { get; set; }  
  
    [Required(AllowEmptyStrings = false)]  
  
    [Display(Name = "Correo")]  
  
    [EmailAddress]  
  
    public string Email { get; set; }  
  
    // Add the Address Info:  
  
    [Display(Name = "Dirección")]  
  
    public string Address { get; set; }  
  
    [Display(Name = "Ciudad")]  
  
    public string City { get; set; }  
  
}
```

```

[Display(Name = "Provincia")]

public string State { get; set; }

// Use a sensible display name for views:

[Display(Name = "Código Postal")]

public string PostalCode { get; set; }

public IEnumerable<SelectListItem> RolesList { get; set; }

}

}

```

Vista Correo electrónico

Este Código permitirá la conexión entre el aplicativo web y el servidor de correo electrónico.

```

using System.ComponentModel.DataAnnotations;

namespace Tesis3.Models

{

    public class Correo

    {

        [Display(Name = "Destinatario")]

        [Required(ErrorMessage = "Por favor ingrese el correo del destinatario.")]

        public string MailTo { get; set; }

        [Display(Name = "Asunto")]

```

```
[Required(ErrorMessage = "Por favor ingrese el asunto del correo.")]

public string MailSubject { get; set; }

[Display(Name = "Mensaje")]

[Required(ErrorMessage = "Por favor ingrese el contenido del correo")]

public string MailBody { get; set; } }
```

Anexo 3

Código Aplicativo Móvil

Vista Login

Código necesario para la realización de la vista principal del aplicativo móvil.

```
public class MainActivity extends AppCompatActivity {

    TextView tv_registrar, tv_olvido, tv_enviar, tv_ingresar;

    EditText et_user, et_pass;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);

        tv_olvido = findViewById(R.id.txtOlvido);

        tv_ingresar = findViewById(R.id.txtIngresar);

        et_user = (EditText) findViewById(R.id.txtUser);
```

```

et_pass = (EditText) findViewById(R.id.txtPass);

tv_olvido.setOnClickListener(new View.OnClickListener() {

    @Override

    public void onClick(View v) {

        Intent intentOlv = new Intent(MainActivity.this,
OlvidoContrasena.class);

        MainActivity.this.startActivity(intentOlv);

    }

});

tv_ingresar.setOnClickListener(new View.OnClickListener() {

    @Override

    public void onClick(View v) {

        ingreso();

    }

});

}

private void attemptLogin() {

}

public Connection conexionBD() {

    Connection conexion = null;

```

```

try {

    StrictMode.ThreadPolicy policy = new
StrictMode.ThreadPolicy.Builder().permitAll().build();

    StrictMode.setThreadPolicy(policy);

    Class.forName("net.sourceforge.jtds.jdbc.Driver").newInstance();

    //conexion =
DriverManager.getConnection("jdbc:jtds:sqlserver://172.45.93.130;databaseNa
me=TESIS;user=sa;password=Pacho@lejo1408;");

    conexion = DriverManager.getConnection("jdbc:jtds:sqlserver://bi-
oildbserver.database.windows.net;databaseName=BI-
OIL_db;user=vm;password=Vlady1995.");

} catch (Exception e){

    Toast.makeText(getApplicationContext(),e.getMessage(),
Toast.LENGTH_SHORT).show();

}

return conexion; }

public void ingreso() {

    String Usuario = et_user.getText().toString();

    String Contraseña = et_pass.getText().toString();

    try {

        PreparedStatement pstL = conexionBD().prepareStatement("select
Correo, Contraseña from Usuarios where Correo=? and Contraseña=?");

```

```

pstL.setString(1, et_user.getText().toString());

pstL.setString(2, et_pass.getText().toString());

ResultSet rs = pstL.executeQuery();

String ms = "";

if (rs.next()) {

    Intent login = new Intent(MainActivity.this, Usuario.class);

    MainActivity.this.startActivity(login);

    Toast.makeText(getApplicationContext(),"Ingreso
Exitoso",Toast.LENGTH_SHORT).show();

}else{

    Toast.makeText(getApplicationContext(),"Usuario o contraseña
incorrecto",Toast.LENGTH_LONG).show();

}

pstL.close();

rs.close();

}catch (SQLException e) {

    Toast.makeText(getApplicationContext(),"Usuario o contraseña
incorrecto",Toast.LENGTH_LONG).show();

} } }

```

Código Conexión Base de datos

Este código permitirá que se establezca una conexión entre el aplicativo móvil con la base de datos para inicio de sesión y obtener los datos del sensor.

```
public class ConexionDB {

    public static Connection ConnectionHelper() {

        StrictMode.ThreadPolicy policy = new
        StrictMode.ThreadPolicy.Builder().permitAll().build();

        StrictMode.setThreadPolicy(policy);

        Connection connection = null;

        String ConnectionURL = null;

        try {

            Class.forName("net.sourceforge.jtds.jdbc.Driver");

            //ConnectionURL =
            "jdbc:jtds:sqlserver://172.45.93.130;databaseName=TESIS;user=sa;password=
            Pacho@lejo1408;";

            ConnectionURL = "jdbc:jtds:sqlserver://bi-
            oildbserver.database.windows.net;databaseName=BI-
            OIL_db;user=vm;password=Udl@2020;";

            connection = DriverManager.getConnection(ConnectionURL);

        } catch (SQLException se) {

            Log.e("ERROR", se.getMessage());

        } catch (ClassNotFoundException e) {

            Log.e("ERROR", e.getMessage());

        }

    }

}
```

```
    } catch (Exception e) {  
  
        Log.e("ERROR", e.getMessage());  
  
    }  
  
    return connection;    }}
```

Vista Gráficas aplicativo móvil

Mediante este Código se puede realizar la visualización de las gráficas en tiempo real del aplicativo móvil mediante los complementos de LineChart.

```
public class LineChartActivity extends AppCompatActivity {  
  
    private LineChart mChart;  
  
    @Override  
  
    protected void onCreate(Bundle savedInstanceState) {  
  
        super.onCreate(savedInstanceState);  
  
        //setContentView(R.layout.activity_usuario);  
  
        setContentView(R.layout.activity_line_chart);  
  
        mChart = findViewById(R.id.chart);  
  
        mChart.setTouchEnabled(true);  
  
        mChart.setPinchZoom(true);  
  
        MyMarkerView mv = new MyMarkerView(getApplicationContext(),  
R.layout.custom_marker_view);  
  
        mv.setChartView(mChart);
```

```
mChart.setMarker(mv);

renderData();

}

public void renderData() {

    LimitLine lXAxis = new LimitLine(10f, "Index 20");

    lXAxis.setLineWidth(4f);

    lXAxis.enableDashedLine(10f, 10f, 0f);

    lXAxis.setLabelPosition(LimitLine.LimitLabelPosition.RIGHT_BOTTOM);

    lXAxis.setTextSize(10f);

    XAxis xAxis = mChart.getXAxis();

    xAxis.enableGridDashedLine(10f, 10f, 0f);

    xAxis.setAxisMaximum(10f);

    xAxis.setAxisMinimum(0f);

    xAxis.setDrawLimitLinesBehindData(true);

    xAxis.setPosition(XAxis.XAxisPosition.BOTTOM);

    LimitLine l1 = new LimitLine(37f, "Limite Maximo");

    l1.setLineWidth(4f);

    l1.enableDashedLine(10f, 10f, 0f);
```

```
ll1.setLabelPosition(LimitLine.LimitLabelPosition.RIGHT_TOP);

ll1.setTextSize(10f);

LimitLine ll2 = new LimitLine(1f, "");

ll2.setLineWidth(4f);

ll2.enableDashedLine(10f, 10f, 0f);

ll2.setLabelPosition(LimitLine.LimitLabelPosition.RIGHT_BOTTOM);

ll2.setTextSize(10f);

YAxis leftAxis = mChart.getAxisLeft();

leftAxis.removeAllLimitLines();

leftAxis.addLimitLine(ll1);

leftAxis.addLimitLine(ll2);

leftAxis.setAxisMaximum(37f);

leftAxis.setAxisMinimum(0f);

leftAxis.enableGridDashedLine(10f, 10f, 0f);

leftAxis.setDrawZeroLine(false);

leftAxis.setDrawLimitLinesBehindData(false);

mChart.getAxisRight().setEnabled(false);

setData();

}
```

```

private void setData() {

    GraficaDAO graficaDAO = new GraficaDAO();

    List<Ploteo> ploteoList= graficaDAO.ObtenerDatos();

    Iterator i = ploteoList.iterator();

    ArrayList<Entry> values = new ArrayList<>();

    String xAxisLables[] = new String[11];

    int ii =0;

    while(i.hasNext()){

        Ploteo target = (Ploteo)i.next();

        System.out.println("valor    x:"+target.getEntryObject().getX()+"    valor
Y:"+target.getEntryObject().getY());

        xAxisLables[ii]=target.getHora();

        // values.add(((Ploteo)i.next()).getEntryObject());

        values.add(target.getEntryObject());

        ii++;

    }

    mChart.getDescription().setText("Estado Contenedor");

    mChart.getXAxis().setValueFormatter(new
IndexAxisValueFormatter(xAxisLables));

    mChart.getXAxis().setLabelRotationAngle(90);

```

```
LineDataSet set1;

if (mChart.getData() != null &&

    mChart.getData().getDataSetCount() > 0) {

    set1 = (LineDataSet) mChart.getData().getDataSetByIndex(0);

    set1.setValues(values);

    mChart.getData().notifyDataSetChanged();

    mChart.notifyDataSetChanged();

    set1.setValueFormatter(new MyDecimalValueFormatter());

} else {

    set1 = new LineDataSet(values, "Datos Distancia");

    set1.setDrawIcons(false);

    set1.enableDashedLine(10f, 5f, 0f);

    set1.enableDashedHighlightLine(10f, 5f, 0f);

    set1.setColor(Color.DKGRAY);

    set1.setCircleColor(Color.DKGRAY);

    set1.setLineWidth(1f);

    set1.setCircleRadius(3f);

    set1.setDrawCircleHole(false);

    set1.setValueTextSize(10f);
```

```
set1.setDrawFilled(true);

set1.setFormLineWidth(1f);

set1.setFormLineDashEffect(new DashPathEffect(new float[]{10f, 5f},
0f));

set1.setFormSize(15.f);

set1.setValueFormatter(new MyDecimalValueFormatter());

if (Utils.getSDKInt() >= 37) {

    Drawable drawable = ContextCompat.getDrawable(this,
R.drawable.fade_blue);

    set1.setFillDrawable(drawable);

} else {

    set1.setFillColor(Color.DKGRAY);

}

ArrayList<ILineDataSet> dataSets = new ArrayList<>();

dataSets.add(set1);

LineData data = new LineData(dataSets);

mChart.setData(data);

} }
```

