



La versión digital de esta tesis está protegida por la Ley de Derechos de Autor del Ecuador.

Los derechos de autor han sido entregados a la "ESCUELA POLITÉCNICA NACIONAL" bajo el libre consentimiento del (los) autor(es).

Al consultar esta tesis deberá acatar con las disposiciones de la Ley y las siguientes condiciones de uso:

- Cualquier uso que haga de estos documentos o imágenes deben ser sólo para efectos de investigación o estudio académico, y usted no puede ponerlos a disposición de otra persona.
- Usted deberá reconocer el derecho del autor a ser identificado y citado como el autor de esta tesis.
- No se podrá obtener ningún beneficio comercial y las obras derivadas tienen que estar bajo los mismos términos de licencia que el trabajo original.

El Libre Acceso a la información, promueve el reconocimiento de la originalidad de las ideas de los demás, respetando las normas de presentación y de citación de autores con el fin de no incurrir en actos ilegítimos de copiar y hacer pasar como propias las creaciones de terceras personas.

***Respeto hacia sí mismo y hacia los demás.***

# **ESCUELA POLITÉCNICA NACIONAL**

**FACULTAD DE INGENIERÍA ELÉCTRICA Y  
ELECTRÓNICA**

**DESARROLLO DE UN PROTOTIPO DE SISTEMA DISTRIBUIDO  
PARA LA MEDICIÓN DE CONSUMO DE AGUA Y VISUALIZACIÓN  
DE INFORMACIÓN.**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE  
INGENIERO EN ELECTRÓNICA Y REDES DE INFORMACIÓN.**

**JORGE SEBASTIAN BAHAMONDE ACHIG**

**DIRECTOR: PhD. ANA MARÍA ZAMBRANO VIZUETE**

**Quito, marzo 2020**

## **AVAL**

Certifico que el presente trabajo fue desarrollado por Jorge Sebastian Bahamonde Achig, bajo mi supervisión.

---

**PhD. Ana María Zambrano**  
**DIRECTORA DEL TRABAJO DE TITULACIÓN**

## **DECLARACIÓN DE AUTORÍA**

Yo, Jorge Sebastian Bahamonde Achig, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración dejo constancia de que la Escuela Politécnica Nacional podrá hacer uso del presente trabajo según los términos estipulados en la Ley, Reglamentos y Normas vigentes.

---

Jorge Sebastian Bahamonde Achig

## **DEDICATORIA**

Dedico este Trabajo de Titulación a madre, hermanos y amigos que de una u otro forma me han apoyado en este largo camino; con el único objetivo de verme cumplir una de mis metas.

A mi madre Rosario del Pilar Achig Paucar por apoyarme, aconsejarme, alentarme y enseñarme que, aunque las cosas se vean difíciles siempre se puede superar cualquier adversidad.

A mis hermanos para que sepan que cualquier cosa se puede llegar a cumplir con esfuerzo y dedicación.

Por último, pero no menos importantes se la dedico a todos mis amigos que estuvieron en las buenas y en las malas. “El Combo” amigos con los que se puede contar para pasarla de lo mejor y que las penas se te olviden. A los “Niños Redes”: Erika Viteri, Fabian Olmedo y Ricardo Taipe con los que se compartí muchos momentos de alegrías, angustias, penas, etc. La compañía de ustedes hizo que el estudiar, realizar proyectos, hacer deberes, rendir pruebas no sea estresante sino un momento para compartir risas y conseguir buenos recuerdos.

Sebastian Bahamonde

## **AGRADECIMIENTO**

Agradezco a mi madre por darme su apoyo constante a lo largo de toda mi vida y por enseñarme a superar una a una las adversidades que se me han presentado. También, le agradezco por que supo guiarme por el camino correcto.

A mi padre Segundo Cesar Bahamonde Guerrero por mostrarme distintos caminos, distintas profesiones, ya que sin eso no hubiera podido elegir correctamente la carrera que hoy estoy culminando.

A mis tíos quienes me supieron darme una mano desinteresadamente en los momentos más difíciles, además, con sus consejos y palabras de aliento he logrado culminar esta etapa de mi vida.

Por último, quiero expresar mi más grande agradecimiento a PhD. Ana María Zambrano, quien me colaboro durante todo este proceso, y con su paciencia, su correcta dirección y conocimientos permitió el desarrollo de este Trabajo de Titulación.

## ÍNDICE DE CONTENIDO

AVAL .....	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA .....	III
AGRADECIMIENTO .....	IV
ÍNDICE DE CONTENIDO .....	V
ÍNDICE DE FIGURAS .....	VI
ÍNDICE DE TABLAS.....	IX
ÍNDICE DE SEGMENTOS DE CÓDIGOS.....	X
RESUMEN .....	XI
ABSTRACT.....	XII
1. INTRODUCCIÓN.....	1
1.1. OBJETIVOS.....	1
1.2. ALCANCE.....	2
1.3. MARCO TEÓRICO.....	3
2. METODOLOGÍA .....	28
2.1. ESTADO ACTUAL DEL SISTEMA DE MEDICIÓN DEL CONSUMO DE AGUA.....	28
2.2. DISEÑO .....	28
2.3. IMPLEMENTACIÓN .....	54
3. RESULTADOS Y DISCUSIÓN .....	81
3.1. ACTUALIZACIÓN DEL TABLERO KANBAN.....	81
3.2. PRUEBAS REQUERIMIENTOS FUNCIONALES .....	81
3.3. PRUEBAS REQUERIMIENTOS NO FUNCIONALES.....	99
3.4. ACTUALIZACIÓN DEL TABLERO KANBAN.....	103
4. CONCLUSIONES Y RECOMENDACIONES.....	105
4.1. CONCLUSIONES .....	105
4.2. RECOMENDACIONES.....	106
5. REFERENCIAS BIBLIOGRÁFICAS .....	107
6. ANEXOS.....	112
ORDEN DE EMPASTADO .....	155

## ÍNDICE DE FIGURAS

Figura 1.1 Arquitectura del servicio web. ....	5
<b>Figura 1.2.</b> Arquitectura de Android. ....	11
<b>Figura 1.3.</b> Arquitectura de iOS.....	13
<b>Figura 1.4.</b> Ventanas de Visual Studio.....	20
<b>Figura 1.5.</b> Arduino WEMOS D1.....	24
<b>Figura 2.1.</b> Sistema actual de medición de agua potable. ....	28
<b>Figura 2.2.</b> Diagrama de funcionamiento global del prototipo. ....	29
<b>Figura 2.3.</b> Tablero Kanban diseño. ....	30
<b>Figura 2.4.</b> Diagrama de Funcionamiento del Módulo Electrónico. ....	33
<b>Figura 2.5.</b> Algoritmo principal del medidor electrónico.....	35
<b>Figura 2.6.</b> Algoritmo de ingreso a la base de datos del valor de consumo. ....	36
<b>Figura 2.7.</b> Algoritmo de ingreso de valores a la memoria SD.....	37
<b>Figura 2.8.</b> Algoritmo de ingreso a la base de datos de valores de consumo atrasados.....	37
<b>Figura 2.9.</b> Diagrama entidad-relación de la base de datos. ....	38
<b>Figura 2.10.</b> Diagrama de casos de uso del sistema. ....	43
<b>Figura 2.11.</b> Diagrama de casos de uso del Módulo Electrónico. ....	44
<b>Figura 2.12.</b> Diagrama de casos de uso módulo de la Módulo Móvil.....	44
<b>Figura 2.13.</b> Diagrama de casos de uso del Módulo Web.....	45
<b>Figura 2.14.</b> Diagramas de clases del sistema.....	46
<b>Figura 2.15.</b> Funcionamiento de un servicio web.....	47
<b>Figura 2.16.</b> Bosquejo de la pantalla para consultar un medidor.....	48
<b>Figura 2.17.</b> Bosquejo de la pantalla principal de la aplicación móvil. ....	48
<b>Figura 2.18.</b> Bosquejo de la pantalla de los datos del usuario.....	49
<b>Figura 2.19.</b> Bosquejo para la consulta de consumos por meses. ....	49
<b>Figura 2.20.</b> Interfaz de consulta por días.....	50
<b>Figura 2.21.</b> Bosquejo de la pantalla de autenticación del Módulo Web.....	51
<b>Figura 2.22.</b> Bosquejo de la pantalla principal del Módulo Web.....	51
<b>Figura 2.23.</b> Bosquejo de la pantalla para la consulta de consumos. ....	52
<b>Figura 2.24.</b> Bosquejo de la pantalla para el ingreso de administradores.....	52
<b>Figura 2.25.</b> Bosquejo de la pantalla para editar administradores.....	53
<b>Figura 2.26.</b> Interfaz para el Registro de Usuarios.....	53
<b>Figura 2.27.</b> Interfaz para la modificación de clientes.....	54
<b>Figura 2.28.</b> Tablero Kanban implementación.....	55
<b>Figura 2.29.</b> Instalación MySQL Workbench.....	56
<b>Figura 2.30.</b> Cargas de trabajo instaladas en Visual Studio. ....	56
<b>Figura 2.31.</b> Ventana de preferencias del IDE de Arduino.....	57
<b>Figura 2.32.</b> Implantación de los elementos del medidor electrónico. ....	58
<b>Figura 2.33.</b> Servicios de la nube de Azure. ....	60
<b>Figura 2.34.</b> Configuración del servidor MySQL.....	61
<b>Figura 2.35.</b> Información general del servidor MySQL. ....	61
<b>Figura 2.36.</b> Cadenas de conexión hacia el servidor MySQL.....	62
<b>Figura 2.37.</b> Pantalla para la creación de una base de datos.....	62
<b>Figura 2.38.</b> Creación de un proyecto ASP.NET. ....	63



<b>Figura 2.39.</b> Tipo de Aplicación web ASP.NET.....	64
<b>Figura 2.40.</b> Métodos get y set. ....	64
<b>Figura 2.41.</b> Agregar nuevo controlador.....	65
<b>Figura 2.42.</b> Creación de aplicación tipo consola web.....	66
<b>Figura 2.43.</b> Creación del proyecto del tipo Aplicación web ASP.NET.....	68
<b>Figura 2.44.</b> Creación de la aplicación web.....	69
<b>Figura 2.45.</b> Implementación del elemento Formulario <i>Web Forms</i> .....	69
<b>Figura 2.46.</b> Interfaz de <i>login</i> de la aplicación web. ....	72
<b>Figura 2.47.</b> Interfaz principal de la aplicación web. ....	72
<b>Figura 2.48.</b> Interfaz para realizar la búsqueda de consumos de la aplicación web. ....	73
<b>Figura 2.49.</b> Interfaz para agregar un nuevo administrador.....	73
<b>Figura 2.50.</b> Interfaz para la consulta, modificación y eliminación de administradores. ....	73
<b>Figura 2.51.</b> Interfaz para el ingreso de nuevos clientes. ....	74
<b>Figura 2.52.</b> Interfaz para la búsqueda, modificación o eliminación de clientes.....	74
<b>Figura 2.53.</b> Creación del proyecto del tipo Aplicación móvil ( <i>Xamarin.Forms</i> ).....	75
<b>Figura 2.54.</b> Ventana de elección de la plataforma móvil. ....	75
<b>Figura 2.55.</b> Implementación del elemento del tipo Página de contenido. ....	76
<b>Figura 2.56.</b> Interfaz de ingreso y principal de la aplicación móvil.....	78
<b>Figura 2.57.</b> Interfaces de datos del cliente y consultas mensuales.....	79
<b>Figura 2.58.</b> Interfaz para consulta de consumo por días.....	79
<b>Figura 2.59.</b> Opción de publicar un proyecto. ....	80
<b>Figura 2.60.</b> Destino de la publicación del proyecto.....	80
<b>Figura 3.1.</b> Tablero Kanban pruebas. ....	81
<b>Figura 3.2.</b> Prueba de funcionamiento. Instalación del módulo electrónico.....	82
<b>Figura 3.3.</b> Prueba de funcionamiento. Prueba de conexión a la red inalámbrica.....	83
<b>Figura 3.4.</b> Prueba de funcionamiento. Comprobación de conexión a la red inalámbrica. ....	83
<b>Figura 3.5.</b> Prueba de funcionamiento. Ingreso del valor de consumo en la base de datos.....	83
<b>Figura 3.6.</b> Prueba de funcionamiento. Valor de consumo en la base de datos. ....	83
<b>Figura 3.7.</b> Prueba de funcionamiento. Ingreso del valor de consumo en la memoria SD. ....	84
<b>Figura 3.8.</b> Prueba de Funcionamiento. Alimentación al Sistema con energía AC.....	84
<b>Figura 3.9.</b> Prueba de funcionamiento. Respaldo de energía con batería. ....	85
<b>Figura 3.10.</b> Ingreso a la aplicación web.....	86
<b>Figura 3.11.</b> Prueba de funcionalidad. Ingreso de Administrador. ....	87
<b>Figura 3.12.</b> Prueba de funcionamiento. Registro de cliente y medidor.....	88
<b>Figura 3.13.</b> Prueba de funcionamiento. Validaciones registro cliente y medidor. ....	89
<b>Figura 3.14.</b> Prueba de funcionamiento. Ingreso del cliente y medidor en la base de datos. ....	89
<b>Figura 3.15.</b> Prueba de funcionamiento. Eliminación y Modificación de un cliente. ....	90
<b>Figura 3.16.</b> Pruebas de funcionamiento. Modificación y eliminación de medidores. ....	91
<b>Figura 3.17.</b> Pruebas de funcionamiento. Realización de consultas por días o meses.....	92
<b>Figura 3.18.</b> Prueba de funcionamiento. Recepción de la notificación. ....	93
<b>Figura 3.19.</b> Prueba de Funcionamiento. Información del cliente. ....	94
<b>Figura 3.20.</b> Prueba de funcionalidad. Ingreso aplicación móvil. ....	94
<b>Figura 3.21.</b> Prueba de funcionamiento. Consulta de consumo por día. ....	95
<b>Figura 3.22.</b> Prueba de funcionamiento. Consulta de consumos mensuales.....	96
<b>Figura 3.23.</b> Prueba de funcionamiento. Consulta de consumos por días. ....	97
<b>Figura 3.24.</b> Interfaz de inicio de sesión. ....	99
<b>Figura 3.25.</b> Nombre de usuario en el Módulo Web. ....	100

<b>Figura 3.26.</b> Mensaje de autenticación fallida.....	100
<b>Figura 3.27.</b> Módulo Web. Resultado de la dificultad de ingreso al Módulo Web.....	100
<b>Figura 3.28.</b> Módulo Web. Resultado de la operabilidad. ....	101
<b>Figura 3.29.</b> Módulo Móvil. Resultado de la dificultad de ingreso al Módulo Móvil.....	101
<b>Figura 3.30.</b> Módulo Móvil. Resultados de la operabilidad. ....	102
<b>Figura 3.31.</b> Tablero Kanban final. ....	104

## ÍNDICE DE TABLAS

<b>Tabla 1.1.</b> Tipos de Peticiones HTTP. ....	6
<b>Tabla 1.2.</b> Ventajas de las aplicaciones web. ....	7
<b>Tabla 1.3.</b> Ventajas y Desventajas de IaaS. ....	9
<b>Tabla 1.4.</b> Ventajas y Desventajas de PaaS. ....	9
<b>Tabla 1.5.</b> Ventajas y Desventajas de SaaS. ....	10
<b>Tabla 1.6.</b> Librerías de Android. ....	12
<b>Tabla 1.7.</b> Reglas Kanban. ....	14
<b>Tabla 1.8.</b> Comandos SQL. ....	17
<b>Tabla 1.9.</b> Cláusulas de SQL. ....	18
<b>Tabla 1.10.</b> Operadores Lógicos SQL. ....	18
<b>Tabla 1.11.</b> Operadores de comparación de SQL. ....	18
<b>Tabla 1.12.</b> Tabla de características de Xamarin. ....	23
<b>Tabla 1.13.</b> Características de la placa WEMOS D1. ....	25
<b>Tabla 1.14.</b> Características del Sensor YF-S201. ....	26
<b>Tabla 1.15.</b> Especificaciones Fuente de Alimentación Seco-Larm ST-2406-2A. ....	27
<b>Tabla 2.1.</b> Requerimientos Funcionales del sistema. ....	31
<b>Tabla 2.2.</b> Requerimientos Funcionales del Módulo Electrónico. ....	32
<b>Tabla 2.3.</b> Requerimientos Funcionales del Módulo Web. ....	32
<b>Tabla 2.4.</b> Requerimientos Funcionales del Módulo Móvil. ....	32
<b>Tabla 2.5.</b> Requerimientos No Funcionales del sistema. ....	33
<b>Tabla 2.6.</b> Elementos del Módulo Electrónico. ....	34
<b>Tabla 2.7.</b> Descripción de la entidad Medidor. ....	39
<b>Tabla 2.8.</b> Descripción de la entidad cliente. ....	39
<b>Tabla 2.9.</b> Descripción de la entidad medida. ....	40
<b>Tabla 2.10.</b> Descripción de la entidad administrador. ....	40
<b>Tabla 2.11.</b> Actividades del usuario dentro del sistema. ....	42
<b>Tabla 2.12.</b> Descripción de los atributos de la clase Medidor. ....	46
<b>Tabla 3.1.</b> Requerimientos Funcionales del Módulo Electrónico. ....	82
<b>Tabla 3.2.</b> Prueba de funcionamiento. Tabla de comparación de mediciones. ....	85
<b>Tabla 3.3.</b> Requerimientos Funcionales del Módulo Web. ....	86
<b>Tabla 3.4.</b> Requerimientos Funcionales del Módulo Móvil. ....	93
<b>Tabla 3.5.</b> Resumen del cumplimiento de los Requerimientos Funcionales. ....	98
<b>Tabla 3.6.</b> Requerimientos No Funcionales del sistema. ....	99
<b>Tabla 3.7.</b> Pruebas de requerimientos no funcionales. ....	103
<b>Tabla 6.1.</b> Descripción de la entidad cliente. ....	120
<b>Tabla 6.2.</b> Descripción de la entidad medida. ....	120
<b>Tabla 6.3.</b> Descripción de la entidad administrador. ....	121
<b>Tabla 6.4.</b> Descripción de los atributos de la clase Persona. ....	122
<b>Tabla 6.5.</b> Descripción de los atributos de la clase Cliente. ....	122
<b>Tabla 6.6.</b> Descripción de los atributos de la clase Administrador. ....	122
<b>Tabla 6.7.</b> Descripción de los atributos de la clase Medida. ....	123

## ÍNDICE DE SEGMENTOS DE CÓDIGOS

<b>Código 2.1.</b> Configuración de variables del módulo electrónico. ....	58
<b>Código 2.2.</b> Función para ingreso del valor de consumo del Módulo Electrónico. ....	59
<b>Código 2.3.</b> Creación de la tabla medidor.....	62
<b>Código 2.4.</b> Codificación de la clase medidor. ....	64
<b>Código 2.5.</b> Servicio web para ingreso de un medidor a la base de datos. ....	65
<b>Código 2.6.</b> Parte del código implementado para el envío de notificaciones. ....	67
<b>Código 2.7.</b> Servicio Web. Envió de notificaciones. ....	68
<b>Código 2.8.</b> Codificación de los controles de la aplicación web. ....	70
<b>Código 2.9.</b> Código para agregar un nuevo medidor. ....	70
<b>Código 2.10.</b> Uso del servicio web en la aplicación web. ....	71
<b>Código 2.11.</b> Codificación de los controles de la aplicación móvil. ....	76
<b>Código 2.12.</b> Código para el ingreso a la aplicación móvil.....	77
<b>Código 2.13.</b> Uso del servicio web en la aplicación móvil. ....	78

## RESUMEN

El presente trabajo de titulación propone un sistema para la lectura del valor del consumo de agua potable de manera remota; este sistema consta de tres módulos los cuales son: un Módulo Electrónico, Módulo Web y Módulo Móvil, los cuales se han desarrollado siguiendo a la arquitectura de 3 capas (Presentación, Negocio y Datos). Cada una de estas capas se encuentran representadas por un servidor alojado en la nube de Microsoft Azure para que se encuentren disponibles en cualquier lugar y momento. La Capa de Datos representada por un servidor MySQL, la Capa de Negocio representada por un servidor de API (*Application Programming Interface*) y la Capa de Presentación representada por un servidor web. Estos servidores son usados por los tres módulos según lo requieran para cumplir con su funcionamiento.

El Módulo Electrónico es el encargado de realizar la medición, la lectura y el registro del consumo de agua en la base de datos. El Módulo Web es el encargado de realizar la gestión de los usuarios, medidores y administradores del sistema, además, de poder realizar la consulta del consumo de agua de los distintos medidores. Por último, el Módulo Móvil es una aplicación móvil para el sistema operativo Android que es usada por los usuarios para realizar la consulta de los consumos de agua de uno o varios medidores.

Además, el sistema envía notificaciones a los usuarios mediante un correo electrónico; este correo se lo enviará una vez al final de cada mes, notificándole el valor de consumo total y el valor que ha consumido los tres últimos meses.

El sistema está desarrollado con diferentes herramientas de software como: IDE Arduino, Visual Studio, MySQL Workbench y diferentes lenguajes de programación como lo son C#, SQL, C.

**Palabras Clave:** Microsoft Azure, IDE Arduino, Visual Studio, MySQL Workbench, Android, SQL, C#, C.

## **ABSTRACT**

This degree work proposes a system for reading the value of water consumption remotely; this system consists of three modules which are: Electronic Module, Web Module and Mobile Module. These modules have been developed following the 3-tier architecture (Presentation, Application and Data). Each of these tiers had been represented by a server that are hosted in the Cloud Computing of Microsoft Azure so that the services are available in anywhere and anytime.

The Data tier is represented by a MySQL server, the Application tier is represented by an API server and the Presentation tier is render by a web server. Each of these servers are used by the modules to do their job.

The Electronic Module is responsible for measuring, reading and registering the water consumption in the Data Base. The Web Module is responsible for managing the users, water meter and administrators of the system also, in this module is possible to do queries about the water consumption of the different water meters. Finally, the Mobile Module is a mobile application for the Android operation system. This application is used for the clients to do queries of the water consumption of one or more water meters.

Also, the system sends notifications to the users through an email; this email will be sent once at the end of each month, this email has the total water consumption and the water consumption of the last three months.

The system is developed with different software tools as IDE Arduino, Visual Studio, MySQL Workbench and different programming languages such as: C#, SQL, C.

**Key Word:** Microsoft Azure, IDE Arduino, Visual Studio, MySQL Workbench, Android, SQL, C#, C.

# **1. INTRODUCCIÓN**

En los últimos años el uso de las redes de comunicaciones ha aumentado brindando ventajas a los usuarios con diferentes tipos de servicios que pueden manejarse mediante red; la mayor ventaja que han presenciado los usuarios al momento de usar una red de datos es la reducción de tiempo que emplean al realizar un trámite o consulta de cualquier servicio del cual estén interesados, sin la necesidad de que él tenga que desplazarse hacia un centro de ayuda o consulta del servicio, actualmente en el Ecuador, el servicio de agua potable no se ha visto beneficiado de la tecnología y de las comunicaciones.

La toma del valor del consumo del agua potable representa uno de los problemas que presentan las empresas que se dedican a la distribución de agua, ya que requiere una gran cantidad de tiempo el tener que pasar por cada una de las residencias tomando el valor del consumo.

Actualmente se tienen sistemas de recolección de datos de consumo de agua que se han estado implementado en diferentes lugares; como en Chile, donde ya se han hecho pruebas de funcionamiento de este tipo de sistemas con NB-IoT [1] y medidores Kamstrup [2]. De la misma manera es necesario dar evolución a estos servicios en el país y establecer un sistema que permita la recolección de datos del consumo de agua potable con las tecnologías e infraestructura que se tiene actualmente.

## **1.1. OBJETIVOS**

### **1.1.1. OBJETIVO GENERAL**

Desarrollo de un prototipo de sistema distribuido para la medición de consumo de agua y visualización de información.

### **1.1.2. OBJETIVOS ESPECÍFICOS**

- Analizar las tecnologías, herramientas y lenguajes para el desarrollo del sistema.
- Seleccionar los componentes electrónicos necesarios para el funcionamiento del módulo electrónico.
- Diseñar el software que será implementado en los tres módulos que intervendrán en el sistema de medición.
- Implementar los módulos siguiendo el diseño definido para el funcionamiento del medidor de consumo de agua, la aplicación web y la aplicación móvil.

## 1.2. ALCANCE

El prototipo se divide en tres partes. La primera, radica en un prototipo electrónico que realizará la medición del consumo del agua potable mediante un sensor del flujo que se encontrará conectado a una placa Arduino y del registro del valor de consumo en la base de datos.

La segunda parte diseñará e implementará una aplicación web para la administración del sistema donde se podrá gestionar a clientes y medidores; además de poder observar los consumos de los clientes.

La tercera parte diseñará e implementará una aplicación móvil en la cual el cliente podrá realizar consultas de consumo del o los medidores que tenga vinculado y podrá observar diferentes tipos de históricos.

El prototipo electrónico se compondrá de una placa Arduino, un sensor de flujo, una placa *wireless*, un banco de baterías como respaldo de energía y de una memoria externa.

La placa permitirá realizar una conexión Wifi con la red local del hogar, con lo cual se podrá realizar una conexión con una base de datos en donde se ingresarán los valores de consumo que proporcionará el sensor de flujo.

En la aplicación web se podrá realizar una administración del sistema la cual consiste en la creación, registro, actualización y eliminación de los clientes y medidores. Se podrá verificar el consumo de cada usuario y para la identificación de casos de extremo consumo; además, se podrá realizar consultas del consumo por fechas. También la aplicación realizará un reporte a final de mes del consumo que ha realizado el o los clientes.

En la aplicación móvil el cliente podrá dar seguimiento a distintos medidores, ver el consumo que ha ido realizando en el mes y observar históricos de consumo mes a mes.

El valor de consumo se almacenará una vez al día en un horario específico, y si no existe una conexión hacia internet en el horario establecido para el almacenamiento de éste, se almacenará en una memoria externa el valor de consumo y la fecha de cuando se realizó la medición. Este valor o valores serán enviados y almacenados en la base de datos cuando exista conexión a internet en el horario específico.

Así mismo, si existe falta de energía eléctrica el módulo electrónico dispondrá de una batería de respaldo para que el funcionamiento del medidor sea ininterrumpido, para esto la batería se la escogerá en la fase de diseño.

Una vez que se tengan los datos de consumo de los diferentes usuarios estos tendrán que ser leídos y presentados de una forma amigable para los usuarios, para lo cual se dispondrá de dos formas de presentación de los datos, las cuales son: la aplicación web



para la administración del sistema y de una aplicación móvil donde el cliente podrá realizar consultas de los valores de consumos por fechas.

### **1.3. MARCO TEÓRICO**

#### **1.3.1. ESTADO DEL ARTE DE SISTEMAS DE MEDICIÓN DE AGUA POTABLE**

El desarrollo de las comunicaciones electrónicas ha permitido realizar un envío de datos entre sistemas haciendo el uso de diferentes tipos de medios; estos datos pueden ser: voltajes, velocidades, flujos, presiones, temperaturas entre otros.

En la actualidad existen sistemas de mediciones remotas que hacen uso de radio frecuencias o del Internet por medio de la telefonía celular, fija o vía satelital. Con los sistemas de medición remota se llega a tener información de diferentes tipos en tiempo real como: perfiles de consumo, fugas, daños en instrumentos de medición, etc.

Además, si estos sistemas son aplicados a tareas como la medición del consumo de agua potable el usuario se asegurará de conocer lo que realmente ha consumido [3].

Entre las soluciones actuales para la mediciones del consumo de agua remotamente, existen soluciones que hacen uso de nuevas tecnologías de comunicación como es el caso de BMETERS [4], la cual implementa la tecnología de comunicación LoRaWan<sup>1</sup>. El funcionamiento de esta solución se basa en un módulo de radio el cual es agregado a un medidor de consumo de agua que posee una salida de pulsos eléctricos; el módulo de radio es el encargado de realizar la conexión a la red y de enviar el valor de consumo del usuario para ser almacenado [4].

Otra solución a la medición remota del consumo de agua potable la entrega Gignac Energie [5], cuya solución consta de un sensor iPERL, el cual es un sensor de medición con tecnología magnética, un sistema de comunicación por radio llamado SensusRF, el cual opera en una banda sin licencia y un portal web para tener acceso a los datos llamado H2OImes.

El funcionamiento de esta solución se basa en el sistema de comunicación SensusRF el cual consta de dos partes: una parte se encuentra en el medidor de agua y otra parte montado sobre un vehículo móvil. La parte del sistema de comunicación que se encuentra en el medidor de agua es la encargada de escuchar peticiones y enviar el valor de consumo de agua; por otro lado, la parte del sistema de comunicación que se encuentra en el vehículo móvil es el encargado de solicitar y de receptar la información de los medidores.

---

<sup>1</sup> LoRaWan (*Low Power Wide Area Network*) es una radio tecnología que nos permite la transmisión de datos con un bajo consumo de energía, opera la banda libre de 868MHz en Europa y 915 MHz en Norte América y Sur América.

Además, posee una antena de telefonía móvil la cual es la encargada de realizar la conexión hacia internet para enviar la información hacia el servidor. Por la manera del funcionamiento del sistema SensusRF el vehículo deberá desplazarse cerca de los medidores de agua para realizar la medición, ya que el alcance de la comunicación es máximo de un kilómetro [5].

El desarrollo de diferentes sistemas para realizar la medición remota de los medidores de agua dependerá de las tecnologías de comunicación usadas, así como también de las necesidades y/o prestaciones que se tengan para implementarlas. En la actualidad, existen tecnologías de comunicación para realizar este tipo de mediciones como: Wireless M-Bus<sup>2</sup>, Sigfox<sup>3</sup>, LoRaWan, NB-IoT<sup>4</sup>. Cada una de estas tecnologías tienen diferentes propósitos, así como también diferentes características como: el alcance de la comunicación, la capacidad de transmitir datos con bajo consumo de energía, inmunidad a las interferencias con otros dispositivos, etc.

Por lo tanto, es necesario involucrarse con el problema de la recolección del valor de consumo de agua de manera remota con un prototipo que haga uso de las tecnologías de comunicación e infraestructura existentes. Por tal motivo, nace este prototipo que propone dar solución al problema mediante el uso de la red WiFi del hogar del usuario como tecnología de comunicación.

### **1.3.2. SERVICIOS WEB, APLICACIONES WEB Y MÓVIL**

Los servicios web son aplicaciones que se pueden escribir, publicar, localizar e invocar a través de una red; además, nos permiten dar flexibilidad a los procesos que sean realizados en un negocio, ya que nos permiten integrar aplicaciones que de otra manera no podrían comunicarse. Los servicios web permiten la interoperabilidad lo cual significa que los servicios se pueden utilizar desde cualquier tipo de aplicación web o móvil independientemente del lenguaje de programación en el que se las haya desarrollado [6].

---

<sup>2</sup> Wireless M-bus es un protocolo de comunicación extendido y estandarizado para la comunicación y lectura inalámbricas. Esta tecnología de comunicación hace uso de bandas sin licencia con distintas frecuencias.

<sup>3</sup> Sigfox es una tecnología propietaria de la empresa del mismo nombre, la cual en vez de utilizar radiofrecuencia hace uso una tecnología de banda ultra estrecha para conectar de forma remota dispositivos inteligentes.

<sup>4</sup> NB-IoT es una tecnología de comunicación que hace el uso de la infraestructura de antenas para la comunicación móvil (LTE) y se encuentra optimizada para ofrecer una cobertura adecuada y para la transmisión de pequeñas cantidades de datos.

La comunicación que ofrece los servicios web entre aplicaciones se basa generalmente en dos tipos de arquitecturas: la arquitectura de Protocolo Simple de Acceso a Objetos o SOAP (*Simple Object Access Protocol*) y la arquitectura de Transferencia de Estado Representacional o REST (*Representational State Transfer*).

SOAP se encuentra basado en la arquitectura de un servicio web el cual se muestra en la Figura 1.1. En esta arquitectura se definen 3 entidades: Proveedor de Servicio, Servicio de Registro y Servicio de Petición.

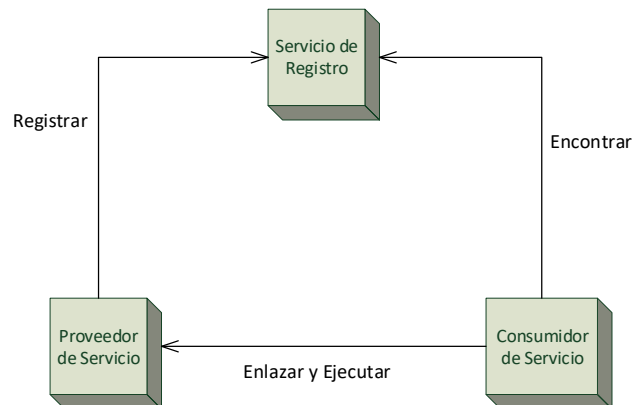


Figura 1.1 Arquitectura del servicio web.

El Proveedor de Servicio, es la entidad que acepta y ejecuta la solicitud del cliente; el Consumidor de Servicio es el cliente quien requiere el servicio; y el Servicio de Registro es un directorio que contiene todos los servicios disponibles.

SOAP realiza la comunicación entre las entidades mediante mensajes SOAP los cuales hacen uso el protocolo del mismo nombre y el lenguaje de marcado extensible o XML (*Extensible Markup Language*). Cada mensaje SOAP se encuentra compuesto por un sobre, cabecera y cuerpo. El sobre identifica al documento XML como un mensaje SOAP, la cabecera contiene la información para realizar peticiones y envío de respuestas, y el cuerpo contiene la información que es solicitada.

Por último, los mensajes SOAP y métodos de invocación como GET, POST, DELETE, PUT son definidos como documentos XML y son enviados mediante un protocolo de transporte como FTP o HTTP. Por otro lado, los servicios web del tipo REST son usados para el desarrollo de APIs<sup>5</sup> permitiendo la comunicación entre clientes y servidor mediante el uso

---

<sup>5</sup> Interfaz de Programación de Aplicaciones o API (Application Programming Interface) interfaz de programación de aplicaciones es un conjunto de funciones y procedimientos los cuales pueden ser utilizados por otras aplicaciones sin la necesidad de conocer como estos se encuentran implementados.

del protocolo de transporte HTTP. A diferencia de un servicio web del tipo SOAP, un servicio REST no requiere un mensaje con un formato de sobre, cabecera y cuerpo ya que la petición lo realiza con URIs<sup>6</sup>, por lo cual el ancho de banda para realizar las peticiones es menor [7].

Los datos que se retornan de las peticiones a servicios del tipo REST pueden ser: XML el cual hace el uso de un lenguaje de etiquetas, Notación de Objetos en JavaScript o JSON (*JavaScript Object Notation*) el cual es un formato de datos basado en el lenguaje de programación de JavaScript y tiene un formato de “nombre:valor” [8].

REST debe ser *STATELESS* (sin estado), es decir que cada petición que el servidor reciba deberá tener un tiempo limitado para ser procesada por el servidor, ya que de lo contrario la memoria RAM del servidor se agotaría. Para acciones como *logins* donde se requiere mantener un estado se hace uso de *TOKENS*, aunque en la mayoría de casos se hace el uso de *cookies*<sup>7</sup>.

Los servicios web del tipo REST implementan recursos los cuales permitirán realizar la comunicación. Estos recursos son las URIs únicas las cuales harán posible que el cliente entienda y haga uso de lo que se le retorna [7].

Los tipos de peticiones que puede realizar REST usando el protocolo HTTP se lo detalla en la **Tabla 1.1**.

**Tabla 1.1.** Tipos de Peticiones HTTP.

PETICIÓN	DESCRIPCIÓN
<b>GET (<i>read</i>)</b>	Usado para la lectura, por ejemplo, para listar usuarios de una base de datos; los parámetros de búsqueda son enviados en la URL.
<b>POST (<i>create</i>)</b>	Usado para mandar información. La información viaja en el cuerpo de la petición, por lo cual no serán visibles para el usuario.
<b>PUT (<i>update</i>)</b>	Usado para crear un registro.
<b>DELETE (<i>delete</i>)</b>	Usado para eliminar un registro.

El uso de cualquiera de estos tipos de comunicaciones dependerá de cómo se desarrolla la aplicación, sin embargo, el uso de SOAP para la comunicación entre aplicaciones causa

---

<sup>6</sup> Identificador de Recursos Uniforme o URI (*Uniform Resource Identifier*) es un identificador de recursos uniforme que es usado para identificar recursos del Internet, estos recursos pueden ser páginas web, servicios, imágenes, videos, etc.[49]

<sup>7</sup> *Cookie* es información que es almacenada en el navegador del usuario.

un mayor tráfico en la red, latencia y retrasos en el procesamiento. Para superar las limitaciones de SOAP se hace uso de REST [7].

Entre los tipos de aplicaciones se tienen aplicaciones web y aplicaciones móviles las cuales se detallan a continuación.

Una aplicación web es una herramienta informática a la cual se puede acceder a través del Internet o de una red local mediante un navegador Web. El desarrollo de aplicaciones web tiene varias ventajas como se las menciona en la **Tabla 1.2**.

**Tabla 1.2.** Ventajas de las aplicaciones web.

No necesita instalación.
Es multiplataforma y multidispositivo.
No depende de la potencia del dispositivo que se accede.
Se puede encontrar en la nube, por lo cual puede ser accesible desde cualquier lugar.
La aplicación es amigable para que el usuario la use fácilmente.

Este tipo aplicaciones se pueden clasificar debido al uso, por ejemplo: *Web Mail* usada para revisar correo electrónico como Gmail; *Wikis* es usada para compartir contenidos informativos como Wikipedia; tiendas *Online* para realizar compras mediante internet como Amazon o aplicaciones que permitan realizar la administración de un sistema [9].

Por otro lado, una aplicación móvil es un programa diseñado para ser ejecutado por dispositivos móviles como *smartphones*, *tablets*, etc. En estas aplicaciones el usuario puede realizar actividades, acceder a servicios o mantenerse informado. En el proceso de la instalación de las aplicaciones móviles se suele necesitar acceder a ciertos tipos de información como: lista de contactos, registro de llamadas, datos transmitidos por internet, datos de localización, entre otros; por lo cual se deberá permitir que la aplicación tenga acceso a la información del teléfono necesaria para que pueda funcionar correctamente [10].

### 1.3.3. CLOUD COMPUTING

El *Cloud Computing* es una tecnología que permite brindar una infraestructura informática, servicios o aplicaciones a cualquier usuario a través del Internet. La nube es un conjunto de recursos virtuales como velocidad de procesamiento o memoria de almacenamiento, los cuales se administran mediante un software de gestión y automatización. Para hacer uso de la nube o *Cloud Computing* es necesario sistemas operativos, software de

virtualización, software de gestión y automatización. Los sistemas operativos hacen uso de las redes e interfaces del usuario host, el software de virtualización extrae todos los recursos y los agrupa en la nube, el software de automatización asigna los recursos, y por último, las herramientas de gestión proporciona nuevos entornos [11].

En la actualidad existen diferentes tipos de implementación de nubes: públicas, privadas o híbridas.

Una nube pública es un depósito de recursos virtuales que son implementados a partir de un hardware cuya propiedad es de una empresa ajena a la compañía o al usuario. En este tipo de nube varias empresas pueden hacer uso de estos recursos, pero de una manera separada; y el proveedor es el encargado de realizar el mantenimiento y de la seguridad. Existen varios proveedores de nubes públicas, de los cuales los más conocidos son AWS (*Amazon Web Services*), GoogleCloud, AlibabaCloud y Microsoft Azure.

Una nube privada es un grupo de recursos software y hardware los cuales se ofrecen a través del internet y que son de uso exclusivo para una empresa u organización. Este tipo de nubes se pueden encontrar en el centro de datos de una empresa o alojarlas en proveedores externos, ya que a este tipo de nube la define los ajustes de privacidad y las responsabilidades de gestión [12].

La nube híbrida al igual que las anteriores nubes es un conjunto de recursos virtuales que son implementados a partir del hardware de una empresa externa y del hardware de la empresa que hace uso de la nube. El software de administración es el encargado de coordinar estos recursos y de permitir que los usuarios puedan tener acceso a ellos mediante una red de datos. En esta arquitectura el trabajo puede ser ejecutado tanto en la nube privada como en la nube pública con el objetivo de separar las cargas de trabajo en cada una de las nubes.

Además, las nubes ofrecen servicios como: Infraestructura como Servicio o IaaS (*Infrastructure as a Service*), Plataforma como Servicio o PaaS (*Platform as a Service*) y Software como Servicio o SaaS (*Software as a Service*) [11].

#### **1.3.3.1. INFRAESTRUCTURA COMO SERVICIO (IaaS)**

Un servicio IaaS es aquel en donde un proveedor brinda una infraestructura informática la cual posee un hardware y software mínimos requeridos para implementar una nube: la red, el almacenamiento, seguridad mediante firewalls, los servidores y la virtualización. El proveedor será el encargado de realizar la administración y el mantenimiento a estos recursos y el usuario puede hacer uso de estos recursos según lo requiera. Existen varios tipos de uso para este tipo de servicio los cuales pueden ser: de manera temporal, para experimentación, alojamiento de proyectos web grandes, para dar solución al

almacenamiento y copia de seguridad, análisis de *big data*, entre otros. Una de las características más importantes para el usuario que hace uso de este tipo de servicio es la de realizar el pago al proveedor solo por los recursos que él está usando[13].

El uso de IaaS presenta ventajas y desventajas como se detallan en la **Tabla 1.3**.

**Tabla 1.3.** Ventajas y Desventajas de IaaS.

Ventajas	Desventajas
No es necesario una inversión en hardware.	El proveedor es el responsable de la disponibilidad y seguridad del servicio.
Implementar proyectos de manera rápida.	El acceso es <i>online</i> , por lo cual cualquier problema de conexión incide en el acceso al servicio.
No existe gasto por instalación ni mantenimiento.	Se puede cambiar el proveedor del servicio de manera sencilla.

### 1.3.3.2. PLATAFORMA COMO SERVICIO (PaaS)

El servicio PaaS es una plataforma la cual se encuentra basada en la nube y en donde se puede desarrollar e implementar software; esto permite que el hardware y software del IaaS se pueda controlar junto con el sistema operativo, un middleware<sup>8</sup> y el entorno de tiempo de ejecución. Este tipo de servicio es especialmente útil para desarrolladores y empresas dedicadas al desarrollo de software, ya que con este tipo de servicio no se deben preocupar de la infraestructura necesaria para el funcionamiento del programa o aplicación [14].

El uso de PaaS tiene ventajas y desventajas como se detallan en la **Tabla 1.4**.

**Tabla 1.4.** Ventajas y Desventajas de PaaS.

Ventajas	Desventajas
El desarrollo de un programa o aplicación es rápido y fácil, ya que no es necesario adquirir o administrar una infraestructura propia.	No se tiene influencia en la infraestructura, por lo cual no puede agregar características que necesite el usuario.
Es escalable ya que se puede ampliar o reducir los recursos contratados.	Solo se puede hacer uso de los lenguajes de programación y herramientas que provea el proveedor.

<sup>8</sup> Middleware es una capa que se encuentra entre el sistema operativo y las aplicaciones. Esta es una capa que permite interactuar o comunicarse entre aplicaciones, redes, hardware o sistemas operativos.

### 1.3.3.3. SOFTWARE COMO SERVICIO (SaaS)

El servicio SaaS se centra en dar un acceso más fácil al usuario a un software a través de un navegador o un programa [11]. Esto hace referencia a que el software se encuentra disponible mediante una conexión a Internet y el cual no necesita ser instalado de forma local. El proveedor de este servicio proporciona el hardware y el software, mientras el cliente hace el uso únicamente de las funciones del software [15].

El uso de SaaS tiene ventajas y desventajas como se detallan en la **Tabla 1.5**.

**Tabla 1.5.** Ventajas y Desventajas de SaaS.

<b>Ventajas</b>	<b>Desventajas</b>
Rápido funcionamiento, el cliente solo debe conectarse.	Cesión de datos, aunque exista un contrato de confidencialidad de datos estos siguen siendo almacenados por el proveedor.
El proveedor es el encargado del mantenimiento.	Rescisión de los servicios, si el proveedor rescinde los servicios se puede perder los datos y documentos.
Se puede hacer uso del software desde diferentes dispositivos fijos o móviles.	Es necesario la compatibilidad, ya que las herramientas pueden diferir el rendimiento dependiendo del sistema operativo y el navegador que utilice.

Debido a las partes del prototipo que serán albergadas en la nube como: la base de datos, los servicios web y la página web se hará uso del servicio SaaS del *Cloud Computing*.

### 1.3.4. DISPOSITIVOS Y SISTEMAS OPERATIVOS MÓVILES

Un dispositivo móvil es aquel que posee ciertas capacidades de procesamiento, almacenamiento y/o conexión a redes; por otro lado, se tienen a los teléfonos móviles los cuales son dispositivos móviles inalámbricos, que tienen la misma función de un teléfono fijo, sin embargo, la principal característica es la movilidad que tiene y que no es necesario un cable para la conexión a una red telefónica [16].

Aunque la funcionalidad principal de un teléfono móvil es la comunicación de la voz, el desarrollo de estos ha hecho posible incorporarles otras funcionalidades como: envío de textos (SMS), acceso a internet, reproducción de video, etc.

Los *smartphones* son dispositivos electrónicos que tienen la funcionalidad de un teléfono móvil, pero que incluyen funciones similares a las de un ordenador. A estos dispositivos se



les permite la instalación de nuevos programas que permitan incrementar el procesamiento de los datos y la conectividad [16].

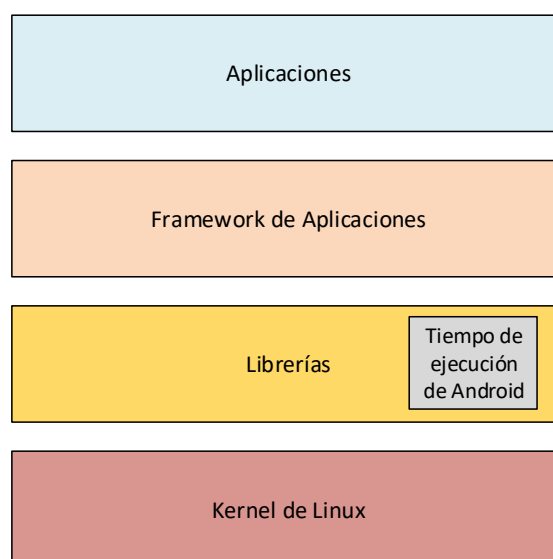
Un *smartphone* tiene entre sus características: un sistema operativo, conectividad a internet, acceso a correo electrónico; además, estos dispositivos vienen con tecnologías integradas como WiFi, *bluetooth*, redes 2G, 3G, etc. El uso de estas tecnologías en los dispositivos móviles permite el desarrollo de distinto tipo de aplicaciones [16].

Con el uso del internet en los *smartphones* se permite hacer uso de navegadores o de aplicaciones en las cuales se deban obtener datos o información de diferentes lugares.

Un *smartphone* posee un sistema operativo la cual es una capa entre el hardware y el usuario, que se lo puede considerar como una máquina virtual, lo cual le permite al usuario hacer uso las herramientas e interfaces para realizar las tareas informáticas sin la necesidad de tener complicados procesos para llevarlos a cabo [16].

Existen varias opciones de sistemas operativos móviles entre los más conocidos se encuentran: Android y iOS. Android es un sistema operativo de código abierto, lo que significa que cualquiera puede descargarlo, instalarlo o modificarlo de una manera gratuita. Este sistema operativo está basado en Linux el cual es un sistema operativo libre, gratuito y multiplataforma. La plataforma en la cual está basada Android permite la instalación de aplicaciones de diversas tiendas de aplicaciones o de aplicaciones propias del usuario. Esta plataforma contiene una pila de software en donde se encuentra el sistema operativo, el middleware y aplicaciones de código abierto.

La arquitectura de Android se compone de capas, cada una de estas capas ofrecen servicios a capas superiores y hace uso de servicios de capas inferiores. Esta arquitectura se lo puede observar en la **Figura 1.2.** [17].



**Figura 1.2.** Arquitectura de Android.

La capa de Aplicaciones contiene las aplicaciones tanto por defecto como las que sean instaladas por el usuario. Todas estas aplicaciones hacen uso de los servicios, API y librerías de las capas inferiores.

La capa de Framework de Aplicaciones en general representa a todas las herramientas para el desarrollo de una aplicación. En esta capa se encuentra un conjunto de APIs y el propio *framework*; entre las APIs más importantes que se encuentran en esta capa son: la de administración de actividades la cual gestiona el ciclo de vida de la aplicación, la de administración de ventanas la cual gestiona las ventanas de las aplicaciones, el sistema de vistas quien proporciona los elementos necesarios para la construcción de cada interfaz.

La capa de Librerías contiene todas las librerías usadas por Android, las cuales junto con el núcleo de Linux conforman la base de Android. Algunas de las librerías más importantes se detallan en la **Tabla 1.6.** [17].

**Tabla 1.6.** Librerías de Android.

<b>Librería</b>	<b>Descripción</b>
Libc	Es la que posee todas la cabeceras y funciones según el estándar del lenguaje C.
Surface Manager	Encargada de componer los elementos de navegación de la pantalla. Además, gestiona las ventanas de cada aplicación que se encuentre activa.
OpenGL/SL y SGL	Estas son librerías gráficas, por lo tanto, da la capacidad gráfica de Android. La librería OpenGL/SL maneja los gráficos 3D y la librería SGL los gráficos 2D.
FreeType	Esta librería es la encargada del manejo de diferentes tipos de fuentes.

La subcapa del Tiempo de ejecución de Android contiene las librerías *core* de Java y la máquina virtual Dalvik<sup>9</sup>.

Por último, se tiene el *Kernel* de Linux el cual tiene la función de abstraer el hardware disponible de los dispositivos móviles para que puedan ser usados por las capas

---

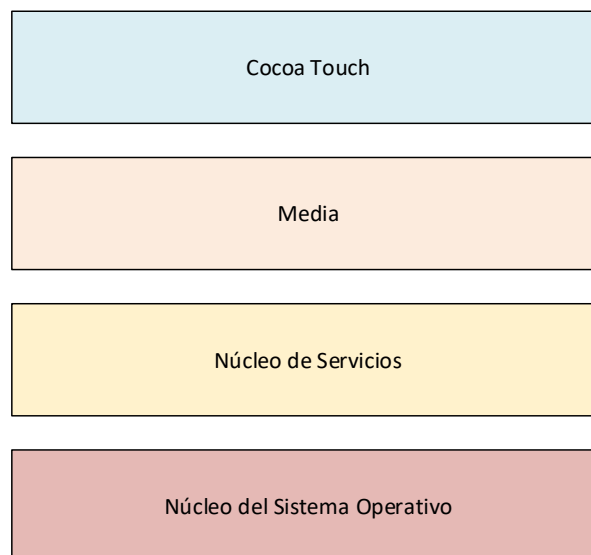
<sup>9</sup> Dalvik es una máquina virtual basada en Java que es usada por los dispositivos móviles debido a que se encuentra optimizada para dispositivos con poca memoria.

superiores. En esta capa se tiene todos los *drivers* necesarios para poder hacer uso del hardware.

Gracias a la arquitectura en la que se encuentra basada Android, es posible tener todas las facilidades para el desarrollo de aplicaciones y que éstas hagan uso de cualquier funcionalidad del teléfono como: llamadas, agenda, cámara, GPS, etc.[17].

En cambio, el sistema operativo iOS es desarrollado y comercializado por Apple Inc., que a diferencia de Android es de carácter propietario, esto quiere decir que solo puede ser comercializado y modificado por los propietarios. La instalación de aplicaciones no se la puede realizar desde cualquier tienda de aplicaciones, por lo cual las aplicaciones deberán estar en la tienda oficial de Apple llamada App Store.

La arquitectura de iOS tiene 4 capas las cuales se muestra en la **Figura 1.3.** [18].



**Figura 1.3.** Arquitectura de iOS.

La capa de *Cocoa Touch* realiza la abstracción del sistema operativo de los *iDevices*<sup>10</sup>. Esta capa hace uso de macros<sup>11</sup> para acceder a funciones del dispositivo, como contactos del usuario, localización, etc.

La capa Media es usada para mejorar las funciones de multimedia del móvil. Esta capa es usada siempre y cuando la capa *Cocoa Touch* no sea suficiente para dar la calidad de imagen o sonido que se requiera.

---

<sup>10</sup> *iDevices*: son dispositivos electrónicos comercializados por Apple como: *iPhones*, *iPods*, *iPads*, etc.

<sup>11</sup> Macros: es un conjunto de comandos los cuales son llamados mediante palabras clave. Estos macros pueden o no tener parámetros.

La capa de Núcleo de Servicios proporciona el acceso a los servicios principales del sistema operativo a las aplicaciones, con los macros de esta capa se tiene acceso a la ubicación del dispositivo, el almacenamiento en el iCloud, compartición de archivos, creación de redes para la compartición de datos móviles, entre otras.

La última capa es el Núcleo del Sistema Operativo la cual está encargada de controlar la memoria, hilos, ficheros del sistema, la red. Esta capa tiene el entorno del *kernel*<sup>12</sup>, controladores e interfaces básicas del sistema operativo [18].

Cada uno de estos sistemas operativos poseen una porción del mercado mundial siendo Android el que posee una mayor cantidad de usuarios con más del 80% del mercado; además, posee la plataforma para la descarga de aplicaciones Google Play desarrollada por Google Inc. la cual es una de la más relevantes del mercado. Después de Android se encuentra iOS con solo un poco más del 10% del mercado [19].

Por las facilidades de instalación, desarrollo de aplicaciones y también por la cantidad del mercado que posee; el desarrollo de la aplicación móvil planteado en el Proyecto de Titulación se lo realiza para el sistema operativo Android.

### 1.3.5. METODOLOGÍA ÁGIL KANBAN PARA EL DESARROLLO DE SOFTWARE

La metodología ágil Kanban es usada para la administración del progreso de las tareas necesarias para realizar un proyecto. Kanban consta de tres reglas para el desarrollo de software la cuales se detallan en la **Tabla 1.7.** [20]

**Tabla 1.7.** Reglas Kanban.

REGLA	DESCRIPCIÓN
<b>Regla 1</b>	Visualizar el trabajo y las fases del ciclo de producción o flujo de trabajo.
<b>Regla 2</b>	Determinar el límite del “trabajo en curso” ( <i>WIP – Work In Progress</i> ).
<b>Regla 3</b>	Medir el tiempo en completar una tarea ( <i>Lead time</i> ).

La Regla 1 de Kanban se basa en la división del trabajo en partes. Para esto se pueden utilizar ayudas visuales para ver la situación de cada parte del trabajo o tareas. Para empezar, se divide en partes el trabajo a realizar. Cada una de estas partes se la escribe

<sup>12</sup> Kernel es un software el cual es la parte fundamental de un sistema operativo.

en un *post-it* y se la pega en la pizarra. El *post-it* debe poseer la información necesaria como la descripción de la tarea y el tiempo de duración de la misma. En la pizarra deben estar los posibles estados por los que puede pasar una tarea.

La idea de este tablero es para que cada persona que se encuentra trabajando en el proyecto sepa qué persona está realizando cada tarea; y de que todos en el grupo tenga una tarea designada. Además, se puede conocer en qué estado se encuentra cada tarea y la prioridad que tiene cada una de ellas [21].

La Regla 2 de Kanban es la determinación del límite del trabajo en curso o WIP (*Work In Progress*). Una de las principales características es que el WIP debe tener un tiempo limitado para realizarlo. Además, se debe definir una cantidad máxima de tareas que se pueden realizar en cada fase del ciclo del trabajo. A este número de tareas se lo llama límite del WIP. Por ejemplo, en la fase de desarrollo se puede tener un máximo de 4 tareas y en pruebas máximo 1 tarea. Además, se tiene el concepto de que una tarea no puede ser iniciada si otra tarea previa aún no ha finalizado [21].

Por último, la Regla 3 de Kanban trata del tiempo que se demora una tarea en ser finalizada. A este tiempo se lo llama *Lead Time*. El *Lead Time* se cuenta desde el momento en que se realiza una petición de la tarea hasta el momento en que se la entrega [21].

### **1.3.6. DESARROLLO POR CAPAS DE SOFTWARE**

El desarrollo por capas de software es una técnica de programación la cual se conforma de 3 capas las cuales son: Capa de Presentación o Frontera, Capa de Lógica del Negocio o Control y Capa de Datos [22].

Debido a este modelo se podrá tener una programación ordenada y separada por módulos y capas que serán independientes entre sí, lo cual beneficia a la reducción en el tiempo de programación.

#### **1.3.6.1. CAPA DE PRESENTACIÓN O FRONTERA**

La presentación de la aplicación ante el usuario debe tener interfaces que realicen las funciones requeridas; además debe ser amigable y de fácil uso para los usuarios ya que el usuario final es el encargado de utilizar el sistema y de la retroalimentación al equipo de desarrollo en caso de que haya algo que mejorar.

Las interfaces deben ser coherentes con la información que se requiere la cual debe ser clara y concisa, no se debe hacer uso de componentes innecesarios, ni tampoco se debe mostrar información no solicitada por el usuario.

La Capa de Presentación tiene componentes encargados de realizar la comunicación del usuario con el sistema mediante el intercambio de información, capturando y desplegando los datos necesarios para realizar alguna tarea. En esta capa los datos se procesan de manera superficial [22].

#### **1.3.6.2. CAPA DE LÓGICA DEL NEGOCIO O CONTROL**

Es esta capa es donde se encuentra la lógica del programa, además de las estructuras de datos y objetos que serán los responsables de la manipulación de los datos existentes. También realizará el procesamiento de la información ingresada o solicitada por el usuario en la Capa de Presentación.

Esta capa se comunica con las otras capas para la realización de las tareas. Si la aplicación interactúa con otros sistemas que funcionan a la par, lo realiza mediante esta capa. También se realiza la comunicación con la Capa de Datos para la adquisición o ingreso de nuevos datos. En esta capa se recibe los datos que ingresó el usuario a través de la Capa de Presentación, los procesa y crea objetos según lo que se necesite hacer con estos datos. A esta acción se denomina encapsulamiento. Al encapsular los datos, el programa asegura mantener la consistencia de los datos y de esta manera se obtendrá la información exacta de las bases de datos [22].

#### **1.3.6.3. CAPA DE DATOS**

En esta capa es donde se encuentran almacenados los datos y es la encargada de acceder a los mismos. Los datos que se ingresan, así como los que se obtienen de la base de datos, deben ser consistentes y precisos.

Esta capa es donde se definen las consultas a realizar en la base de datos (consultas simples y complejas). También es la encargada de realizar el intercambio de la información directamente con la Capa de la Lógica del Negocio para que sea procesada e ingresada en objetos según se necesite [22].

### **1.3.7. TECNOLOGÍAS, HERRAMIENTAS Y LENGUAJES A UTILIZAR EN LA CAPA DE DATOS**

#### **1.3.7.1. MYSQL**

Es un Sistema de Gestión de Base de Datos Relacional o RDBMS (*Relational Database Management System*) de código abierto, el cual se encuentra basado en el lenguaje de consulta estructurado o SQL (*Structured Query Language*). El RDBMS es un programa que permite la creación, actualización y administración de una base de datos relacional.

El propósito de un modelo relacional de bases de datos es representar los datos en un conjunto de tablas y haciendo uso de punteros para poder navegar entre los datos de las mismas. Cada tabla en el modelo relacional representa a una entidad y las tablas podrán poseer tantas columnas como sean necesarias; cada columna representará a un atributo de la entidad. Además, una columna contendrá información única de cada tabla llamada Clave Primaria (*Primary Key*) y la información adicional o atributos para definir completamente la entidad se distribuirá en cada columna. Una fila es un conjunto de columnas que juntas describe a una entidad.

Además, algunas tablas pueden incluir información que podrá ser usada para la navegación entre otras tablas; estas columnas son conocidas como Claves Foráneas (*Foreign Keys*) y tiene el propósito de conectar a las entidades [23].

### 1.3.7.2. SQL

El lenguaje SEQUEL o SQL se encuentra estandarizado por la ANSI/ISO usado para la creación, manipulación y control de bases de datos. Este lenguaje se encuentra compuesto por comandos, cláusulas, operadores lógicos, y de comparación [24].

Existen tres tipos diferentes de comandos los cuales se detalla en la **Tabla 1.8**.

**Tabla 1.8.** Comandos SQL.

COMANDOS	DESCRIPCIÓN
<b>DDL (Data Definition Language)</b>	Este tipo de instrucciones son usadas para crear y modificar la estructura de los registros en la base de datos. A cada instrucción se le asocia una palabra clave con la instrucción DDL las cuales son: CREATE, ALTER y DROP respectivamente.
<b>DCL (Data Control Language)</b>	Mediante este tipo de instrucciones se controla el acceso a los registros de la base de datos. Mediante la instrucción GRANT se otorga acceso y con REVOKE se restringe el acceso a diferentes usuarios.
<b>DML (Data Manipulation Language)</b>	En este tipo de instrucción realizar la manipulación de datos como: seleccionar, agregar, modificar o borrar registros almacenados en la base de datos mediante las instrucciones: SELECT, INSERT, UPDATE y DELETE respectivamente.

Por otro lado, las cláusulas, operadores lógicos y de comparación son condiciones que permiten definir los datos que se desea seleccionar o manipular. Los diferentes tipos de cláusulas se detallan en la **Tabla 1.9**.

**Tabla 1.9.** Cláusulas de SQL.

Cláusula	Descripción
<b>FROM</b>	Es utilizado para seleccionar la tabla en la cual se va a realizar acciones.
<b>ORDER BY</b>	Utilizado para ordenar los registros en un orden específico.
<b>WHERE</b>	Utilizado para seleccionar registros donde se cumple cierta condición.

Los operadores lógicos utilizados en el lenguaje SQL se detallan en la **Tabla 1.10**.

**Tabla 1.10.** Operadores Lógicos SQL.

Operador Lógico	Descripción
<b>AND</b>	Representa al “y” lógico. Este operador evalúa dos condiciones y devuelve un valor de verdad siempre y cuando ambas sean verdaderas.
<b>OR</b>	Representa al “o” lógico. Este operador evalúa dos condiciones y devuelve un valor de verdad si alguna de las condiciones es verdadera.
<b>NOT</b>	Representa a la negación lógica. Devuelve un valor de verdad contrario al de la expresión.

Por último, los operadores de comparación utilizados en el lenguaje SQL se detallan en la **Tabla 1.11**.

**Tabla 1.11.** Operadores de comparación de SQL.

Operador de comparación	Descripción
<b>&lt;</b>	Menor que
<b>&gt;</b>	Mayor que
<b>&lt;&gt;</b>	Distinto de
<b>&lt;=</b>	Menor o igual que
<b>&gt;=</b>	Mayor o igual que
<b>BETWEEN</b>	Permite la comparación entre un intervalo.

### 1.3.7.3. MYSQL WORKBENCH

MySQL Workbench es una herramienta visual desarrollada para el uso administradores, desarrolladores y arquitectos de bases de datos. Además, permite realizar modelación de



datos, administración de usuarios, configuración de servidores y muchas otras más funcionalidades [25].

Esta herramienta es compatible con los sistemas operativos Windows, Linux y MAC OS X. Entre las principales funcionalidades están: desarrollo SQL, modelaje de datos y administración de servidores MySQL. La funcionalidad de desarrollo SQL permite realizar la creación de bases de datos relaciones SQL y la administración de servidores de bases de datos SQL. La funcionalidad del modelaje de datos permite la creación de diagramas de las bases de datos UML<sup>13</sup>. Con la funcionalidad de administración de servidores MySQL se realiza la creación de usuarios, asignación de privilegios a los usuarios sobre los esquemas de bases de datos [26]. MySQL Workbench posee un editor de tablas en el cual se realiza la edición tablas, columnas, índices, opciones, inserciones, etc. de las diferentes bases de datos. También tiene un editor SQL en el cual se realiza la ejecución de consultas SQL.

### **1.3.8. TECNOLOGÍAS, HERRAMIENTAS Y LENGUAJES A UTILIZAR EN LA CAPA DE NEGOCIO**

#### **1.3.8.1. FRAMEWORK .NET**

.NET es un software el cual ha sido diseñado y desarrollado por Microsoft. Es una máquina virtual en la cual se compila y ejecuta programas de diferentes lenguajes de programación como: C#, VB.Net, etc.

Este *framework* es usado para el desarrollo de aplicaciones basadas en ventanas para Windows, aplicaciones para la web, aplicaciones móviles, servicios web, etc.

Los principales componentes de .NET son: Entorno en Tiempo de ejecución de Lenguaje Común o CLR (*Common Language Runtime*) y la Librería de Clases o FCL (*Framework Class Library*). El CLR es el componente básico de la máquina virtual de .NET y es el entorno del tiempo de ejecución de .NET en el cual se ejecuta el código y ayuda al proceso de compilación dando servicios como administración de hilos, manejo de memoria, robustez, etc. Además, este componente es el responsable del manejo de la ejecución de los programas que se han desarrollado en .NET con cualquiera de los lenguajes de programación que este soporta. El FCL es una colección reusable de librerías de clases y métodos Orientados a Objetos que pueden ser integrados en el CLR.

---

<sup>13</sup> Lenguaje Unificado de Modelado o UML (*Unified Modeling Language*) Lenguaje unificado de modelado es un estándar usado para la creación de esquemas, diagramas y documentación referente al desarrollo de software.

La implementación de .NET se basa de la implementación de los componentes CLR y FCL en el sistema.

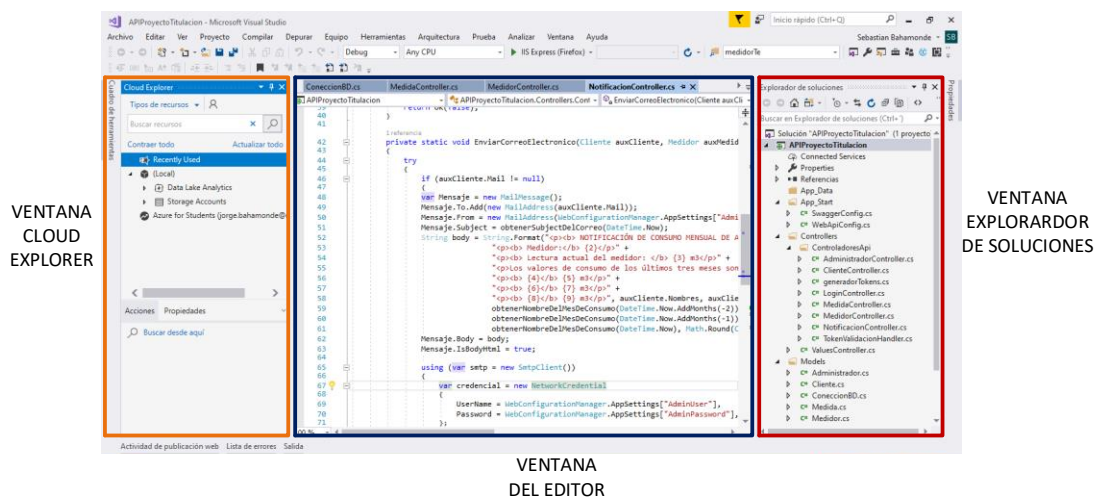
Una aplicación desarrollada con .NET solo se ejecutará en sistemas operativos basados en Windows ya que la aplicación depende del *framework* .NET [27].

### 1.3.8.2. VISUAL STUDIO

Visual Studio es un Entorno de Desarrollo Integrado o IDE (*Integrated Development Enviroment*) [28] el cual posee un panel donde se encuentran las opciones para editar, depurar y compilar código; además, permite la publicación de la aplicación en la nube de Azure.

Visual Studio a diferencia de otros IDEs que solo poseen editores de texto y el depurador, proporciona compiladores, herramientas de finalización de código, diseñadores gráficos, etc. [28].

El entorno general de Visual Studio proporciona diferentes tipos de ventanas como se muestra en la **Figura 1.4**, por ejemplo, la venta de Explorador de Soluciones en la cual se observa, navega y administran los diferentes archivos del código. La ventana del editor en la cual se muestra el contenido de cada uno de los archivos de código y en donde se edita el código o se diseña la interfaz del usuario. Además, con la ventana del *Cloud Explorer* del IDE se puede ver y administrar los recursos basados en la nube. Los recursos que se pueden administrar incluyen máquinas virtuales, tablas, bases de datos SQL, entre otras.



**Figura 1.4.** Ventanas de Visual Studio.

Entre algunas de las características avanzadas que posee Visual Studio se encuentra la instalación modular la cual permite elegir diferentes cargas de trabajo e instalarlas; las cargas de trabajo de Visual Studio son características necesarias para el desarrollo un software en un cierto lenguaje de programación o para una cierta plataforma (escritorio,

nube, web, etc.), esta característica de instalación por carga de trabajo ayuda a que la instalación de Visual Studio sea más ligera ya que solo se instalará lo necesario para el desarrollo de la aplicación y no de todas las funcionalidades de Visual Studio. Haciendo uso de esta característica se realiza la instalación de Xamarin la cual es usada para el desarrollo de aplicaciones móviles, además, se realiza la instalación del módulo desarrollo en la nube, el cual permite hacer uso de la nube de Azure, con la instalación de este módulo el IDE de Visual Studio permite la creación de aplicaciones para la nube de Microsoft Azure, ya que permite la configuración, compilación, depuración, empaquetamiento e implementación aplicaciones y servicios directamente en la nube [29].

### **1.3.8.3. LENGUAJE DE PROGRAMACIÓN C #**

C Sharp o C# es un lenguaje de programación Orientado a Objetos desarrollado por Microsoft que permite a desarrolladores realizar aplicaciones que se ejecutan en el *framework* de .NET. Entre las aplicaciones que se pueden desarrollar se tiene: Aplicaciones cliente de Windows, Servicios Web, Componentes distribuidos, Aplicaciones Cliente-Servidor, entre otras [30].

C# es un lenguaje de programación sencillo ya que elimina ciertos elementos que en otros lenguajes como C o C++ se incluyen y en .NET son innecesarios, por ejemplo, el tamaño de los tipos de datos básicos, en C# este tamaño es fijo e independiente del compilador, sistema operativo o máquina para quienes se compile la aplicación, por lo cual se facilita la portabilidad del código; además, no se incluyen elementos como macros o herencia múltiple.

Es un lenguaje moderno ya que agrega ciertos elementos útiles como el tipo de dato decimal el cual permite realizar operaciones de alta precisión. Agrega también la instrucción *foreach* la cual permite recorrer colecciones o listas con facilidad; el tipo de dato *String* para representar cadenas de caracteres y el tipo de dato *Boolean* el cual es usado para representar valores lógicos de verdad o falso [31].

Dado que C# es un lenguaje de programación Orientado a Objetos, éste soporta todas las características propias del paradigma de este tipo de programación: encapsulación, herencia y polimorfismo. La encapsulación es una propiedad que permite que la información de un objeto este oculta del mundo exterior, la cual consiste en que los atributos de un objeto tengan cierto nivel de acceso, para lo cual se tienen los modificadores: *public*, *private*, *protected*; C# añade además un cuarto modificador llamado *internal*, a este modificador se lo puede combinar con el modificador *protected* para indicar que el elemento solo puede ser accedido dentro de la definición del objeto.

La herencia es una propiedad en la cual se permiten crear objetos a partir de un objeto base heredando sus atributos, se tiene dos tipos de herencias: herencia simple cuando se hereda solo de un objeto base; la herencia múltiple cuando se hereda de dos o más objetos bases. C# sólo se admite herencia simple de clases. Por último, el polimorfismo es una propiedad que permite tener comportamientos diferentes de un método con el mismo nombre [32].

En este lenguaje de programación no se usa variables no inicializadas, para lo cual el compilador asigna a los campos un valor por defecto, este valor corresponderá al equivalente a cero de cada tipo de dato; además, no se lee ninguna variable local sin que se le haya asignado previamente algún valor. También comprueba que el acceso a los elementos de una lista o colección se realice con índices que se encuentren dentro del rango de la misma. C# permite definir métodos que admitan un número indefinido de parámetros de cualquier tipo de dato, además, comprueba que los valores que se les pasen a los métodos sean del tipo dato apropiado.

Por último, en C# todos los tipos de datos que se definan derivarán de manera implícita, de una clase base común llamada *System.Object*, por lo cual a cualquier tipo de dato se lo considera como un objeto y heredará todos los miembros definidos en esta clase [31].

Debido a las características y la flexibilidad que se tiene para el desarrollo de distinto tipo de aplicaciones el lenguaje de programación C# será usado para el desarrollo de las aplicaciones web y móvil.

### **1.3.9. TECNOLOGÍAS, HERRAMIENTAS Y LENGUAJES A UTILIZAR EN LA CAPA DE PRESENTACIÓN**

#### **1.3.9.1. ASP .NET**

*Active Server Pages (ASP)* es una tecnología desarrollada por Microsoft para la creación de páginas dinámicas. Generalmente usada por diseñadores para la construcción de sitios, aplicaciones dinámicas, REST APIs y microservicios. ASP.NET se encuentra dentro del *Framework .NET*, y está basado sobre un CLR, por lo cual permite a los desarrolladores escribir el código en cualquier lenguaje admitido por el .NET Framework como C#, F#, Visual Basic .NET.

Los sitios creados con ASP.NET son conocidos como “*web forms*”, los cuales tiene la extensión .ASPX. Además, tiene etiquetas HTML. Por otro lado, se puede tener etiquetas que definen Controles Web los cuales se ejecutan del lado del servidor y Controles de Usuarios en los cuales el programador ubica el código estático y dinámico que requiera la página web [33].

Las aplicaciones desarrolladas con ASP.NET corren sobre Windows, Linux, macOS, y Docker [34].

### 1.3.9.2. XAMARIN

Xamarin se extiende de la plataforma .NET con herramientas y librerías para desarrollar en C#, aplicaciones para iOS, Android, macOS, etc. Además de combinar todas las características de las plataformas nativas para cada sistema operativo agrega características propias para ayudar a desarrollo de aplicaciones para los sistemas operativos móviles antes mencionados. Las características de Xamarin se detallan en la **Tabla 1.12**. Las aplicaciones que son desarrolladas con Xamarin tienen acceso completo a todas las funcionalidades de la plataforma y del dispositivo, además, agrega librerías de terceros como servicios de Google Play, Facebook, Google APIs para iOS, etc.[35].

**Tabla 1.12.** Tabla de características de Xamarin.

CARACTERÍSTICA	DESCRIPCIÓN
<b>Enlaces completos para los SDK subyacentes.</b>	Xamarin contiene enlaces para casi todos los SDK de plataforma subyacentes en iOS y Android.
<b>Interoperabilidad con Objective-C, Java, C y C++</b>	Xamarin nos permite llamar directamente a las bibliotecas de Objective-C, Java, C y C++.
<b>Construcciones con lenguaje moderno.</b>	Las aplicaciones de Xamarin se escriben en el lenguaje de programación C#, un lenguaje que nos permite incluir las mejoras considerables en Objective-C y Java.
<b>Increíble biblioteca de clases base (BCL)</b>	Lo cual nos permite realizar un trabajo eficaz y compatibilidad con XML, bases de datos, serialización, E/S, cadenas y redes, etc.
<b>Compatibilidad multiplataforma móvil.</b>	Xamarin ofrece una compatibilidad multiplataforma para las tres principales plataformas móviles: iOS, Android. Lo cual nos permite realizar aplicaciones que compartan hasta un 90% del código, y la biblioteca de Xamarin.Mobile ofrece una API unificada para tener acceso a recursos comunes de las tres plataformas [36].

Las herramientas mencionadas en las distintas capas para el desarrollo del software, fueron seleccionadas debido a que se apegan a las necesidades requeridas para el



La placa WEMOS D1 es utilizada en este Proyecto de Titulación haciendo uso de las principales características de esta como es la conexión WiFi y de la seguridad para la conexión. Como la placa brinda varios tipos de seguridad para la conexión inalámbrica, y la seguridad depende de la red inalámbrica del usuario. Esta característica brinda una gran versatilidad para el uso del prototipo en diferentes ámbitos de seguridad de las redes inalámbricas. Así también se tiene la característica de manejar interrupciones<sup>15</sup> lo cual es necesario para el funcionamiento del flujómetro. Otras características de la placa WEMOS se detallan en la **Tabla 1.13**. [39].

**Tabla 1.13.** Características de la placa WEMOS D1.

<b>Característica</b>	<b>Definición</b>
<b>Redes Inalámbricas</b>	802.11 B/G/N
<b>Memoria Flash</b>	4MB
<b>Pines GPIO</b>	11
<b>Pines Analógicos</b>	1
<b>Puerto de Comunicación</b>	Serial
<b>Modos de Operación</b>	Cliente, <i>Access Point</i> y simultaneo.
<b>Seguridades Inalámbricas</b>	OPEN / WEP / WPA_PSK / WPA2_PSK / WPA_WPA2_PSK.
<b>Protocolos de Transporte</b>	TCP y UDP
<b>Interrupciones</b>	En todos los pines excepto en D0.
<b>Voltaje de entrada</b>	12-24 V
<b>Corriente de consumo</b>	150mA

### 1.3.10.2. SENSOR DE FLUJO YF-S201

El sensor de flujo o caudalímetro YF-S201 tiene una válvula plástica, un rotor y un sensor de efecto hall<sup>16</sup>; y es usado para líquidos de baja viscosidad como: bebidas gasificadas, bebidas alcohólicas, combustible, etc. El caudalímetro es compatible con la mayoría de los sistemas digitales como Arduino, Circuito Integrado Programable o PIC (*Programmable*

<sup>15</sup> Interrupción en el ámbito de hardware es un mecanismo el cual permite la comunicación entre el procesador y los dispositivos de entrada y salida.

<sup>16</sup> El efecto hall se basa en la aparición de un campo eléctrico por separación de cargas en el interior de un conductor y por el cual circula una corriente perpendicularmente al movimiento de las cargas.

*Integrated Circuited*), Raspberry Pi, Controlador Lógico Programable o PLC (*Programmable Logic Controller*), etc. [40].

El funcionamiento de este caudalímetro empieza con el flujo del agua a través del rotor haciéndolo girar a diferentes velocidades, el rotor se encuentra unido a un imán el cual activa al sensor de efecto hall, generando un pulso eléctrico el cual puede ser leído por una entrada digital.

Para calcular el volumen del líquido que pasa a través del flujómetro se debe conocer el factor de conversión  $K$  de frecuencia a caudal el cual será diferente dependiendo del diámetro de la tubería. El fabricante especifica tres ecuaciones para el cálculo del volumen del líquido que pasa por el flujómetro. Estas ecuaciones se las detallan a continuación [40]. La Ecuación (1.1) es usada para calcular el caudal a partir de la frecuencia en un flujómetro YF-S201.

$$f(Hz) = K * Q\left(\frac{L}{min}\right) \quad (1.1)$$

Una vez calculado el caudal que pasa a través del flujómetro se podrá calcular el volumen con la Ecuación (1.2) detallada a continuación:

$$V = V_0 + Q\Delta t \quad (1.2)$$

El valor de  $K$  dará la exactitud en la medida del flujómetro, por lo cual si se requiere una mayor exactitud en la medición se deberá realizar el cálculo del factor de conversión con la Ecuación (1.3).

$$K = \frac{N^{\circ} Pulsos}{Volumen * 60} \quad (1.3)$$

Las características de funcionamiento de flujómetro se detallan en la **Tabla 1.14**.

**Tabla 1.14.** Características del Sensor YF-S201.

Característica	Descripción
Mín. y Máx. voltaje de trabajo	4.5V a 24V DC
Voltaje normal de trabajo	5V a 18V DC
Máx. Corriente de trabajo	15mA (DC 5V)
Rango de flujo de agua	1 a 30 L/min
Temperatura del Líquido	≤ 120°C
Diámetro	½"



### 1.3.10.3. FUENTE DE ALIMENTACIÓN/CARGADOR SECO-LARM ST-2406-2A

Este tipo de fuente de alimentación es usada para sistemas de alarma o control de acceso para que tengan un respaldo de energía y se asegure el correcto funcionamiento de los sistemas antes mencionados. Este tipo de fuente de alimentación permite la recarga automática de baterías de respaldo, la conmutación entre el modo de batería y el suministro de energía. Además, permite una salida auxiliar programable de 6,12 o 24 VCC [41].

Esta fuente de alimentación posee especificaciones técnicas para el correcto funcionamiento de la fuente, detalladas en la **Tabla 1.15**.

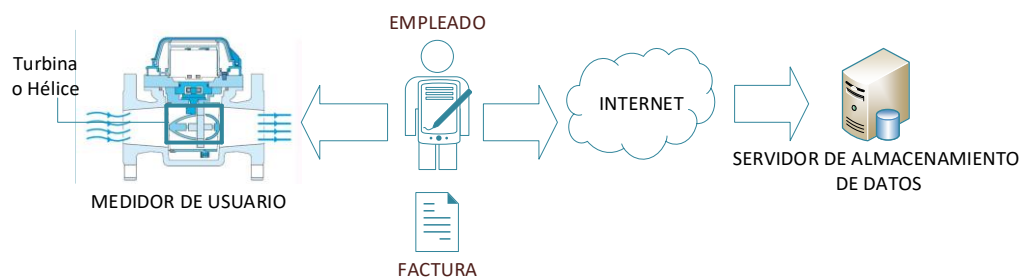
**Tabla 1.15.** Especificaciones Fuente de Alimentación Seco-Larm ST-2406-2A.

<b>CARACTERÍSTICA</b>	<b>DESCRIPCIÓN</b>
<b>Voltaje de salida CC</b>	6,12,24
<b>Voltaje de Operación</b>	12-24 VCA/VCC para 6VCC 18-24 VCA/VCC para 12VCC 24 VCA/VCC para 24VCC
<b>Corriente (Continua)</b>	1.5 <sup>a</sup>
<b>Corriente (Punto más alto)</b>	2.0A
<b>Corriente de Salida (carga máxima)</b>	6V = 100mA 12V = 220mA 24V = 450mA
<b>Temperatura de funcionamiento.</b>	-20 a 65° C

## 2. METODOLOGÍA

### 2.1. ESTADO ACTUAL DEL SISTEMA DE MEDICIÓN DEL CONSUMO DE AGUA

El funcionamiento del sistema actual de medición y toma del valor de consumo de agua potable de los usuarios se muestra en la **Figura 2.1**. Este sistema comienza por el medidor de agua potable del usuario, el cual tiene un flujómetro del tipo turbina tangencial. Este tipo de flujómetro es del tipo de “chorro único”, lo que quiere decir que el caudal de entrada del flujómetro es desviado para incidir de forma tangencial sobre una turbina o hélice. Esta hélice se encuentra en sentido perpendicular al flujo del agua girando a una velocidad directamente proporcional al caudal volumétrico del líquido. El giro de la turbina o hélice hace que el contador mecánico se mueva y muestre la cantidad de consumo.



**Figura 2.1.** Sistema actual de medición de agua potable.

Una vez que se tiene la forma de medir la cantidad de consumo de agua se tiene un método para recolectar el valor de consumo del agua potable de cada usuario, para lo cual el personal contratado por la empresa que ofrece el servicio de agua potable recorre las ubicaciones de todos los medidores tomando el valor de consumo de cada uno de estos medidores y registrándolos en una base de datos mediante un dispositivo móvil diseñado para este propósito. Este proceso de toma del valor de consumo del agua potable se lo realiza una vez al mes por lo cual puede presentar diferentes problemas como error en la toma del valor de consumo por falla humana, fallo en la comunicación para el registro del valor de consumo y que éste no pueda ser ingresado en ésta única ocasión en la cual se realiza la toma del valor de consumo.

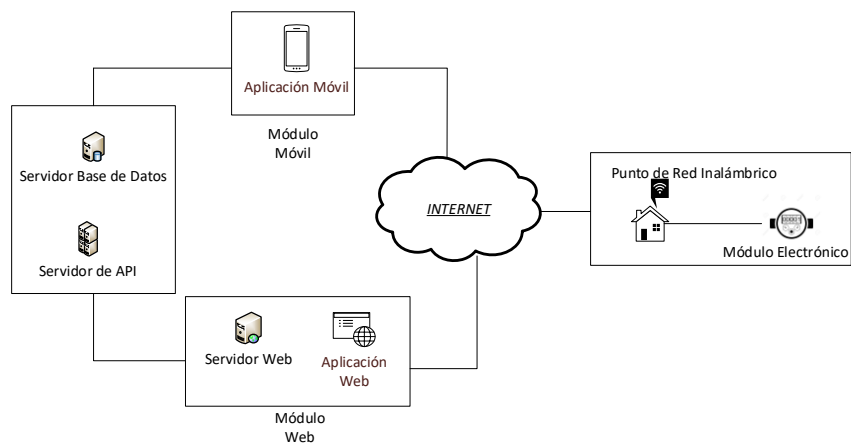
### 2.2. DISEÑO

En el siguiente apartado se detallará el diseño del prototipo del sistema distribuido para realizar la medición del valor del consumo de agua de manera remota y automática. El diseño se lo realiza siguiendo el desarrollo por capas de software las cuales son: Capa de Datos, Capa de Negocio y Capa de Presentación; y un diseño para la parte electrónica del prototipo. El primer paso para realizar el diseño del Sistema de Medición de Consumo de Agua Potable es realizar un tablero Kanban con las actividades a realizarse para el

cumplimiento de los Requerimientos Funcionales y No Funcionales del prototipo. El diseño de la parte electrónica constará de un diagrama de flujo; en la Capa de Datos se realizará un diagrama relacional de la base de datos; en el diseño de la Capa de Negocio se realizará un diagrama de clases y, diagramas de casos de uso para mostrar el funcionamiento del prototipo y el diseño de los servicios web; en la Capa de Presentación se realizará los *Mock-Ups*<sup>17</sup> de la aplicación web y móvil.

### 2.2.1. DIAGRAMA GLOBAL DEL PROTOTIPO

La **Figura 2.2** representa el funcionamiento del prototipo propuesto para el Sistema de Medición del Consumo de Agua Potable de manera general. El prototipo consta de tres módulos: Módulo Electrónico, Módulo Web y Módulo Móvil.



**Figura 2.2.** Diagrama de funcionamiento global del prototipo.

El prototipo propuesto hace uso de tres tipos de servidores: Servidor Web, Servidor API y un Servidor de Bases de Datos; estos servidores formarán parte de cada uno de los módulos.

El Módulo Electrónico dentro del sistema será el encargado de realizar la medición del consumo de agua potable del usuario. Este módulo hará uso de los Servidores de API y Base de Datos para almacenar el valor de consumo.

Por otro lado, el Módulo Web dentro del sistema tiene el papel de gestionar a los usuarios, medidores y administradores; por lo cual este módulo tiene una aplicación web, la cual será alojada en un Servidor Web para que sea accesible desde cualquier lugar. Además, este módulo hará el uso de los Servidores API y Base de Datos para obtener, almacenar, eliminar o editar información.

<sup>17</sup>Un *Mock-Up* es un fotomontaje el cual es usado por diseñadores gráficos para mostrar los diseños al cliente.

Por último, se tiene el Módulo Móvil, el cual consta de una aplicación móvil que se deberá instalar en un *Smartphone* con el sistema operativo Android. Esta aplicación hará el uso de los Servidores de API y Base de Datos para realizar consultas de consumo de agua potable de un medidor en específico.

## 2.2.2. TABLERO KANBAN

*KanbanTool* [42] es una herramienta en línea que sirve para realizar la gestión de tareas, pudiendo ser configuradas y modificadas con el fin de formar tableros Kanban. A cada tarea se le puede agregar la información necesaria para que el personal encargado pueda desarrollarla. En la **Figura 2.3** se muestra el Tablero Kanban correspondiente a esta fase.



**Figura 2.3.** Tablero Kanban diseño.

El tablero de la **Figura 2.3** consta de tres columnas las cuales son: “Lista Tareas”, “En Proceso” y “Completadas”; cada una de estas columnas representa el estado posible en el que se puede encontrar una tarea. En la columna “Lista Tareas” se encontrarán todas las tareas pendientes; en la columna “En Proceso” se encontrarán las tareas que se encuentren realizándose y por último en la columna “Completadas” se encontrarán las tareas que se encuentran terminadas.

### 2.2.3. REQUERIMIENTOS FUNCIONALES

Los Requerimientos Funcionales (RF) se los obtuvo después de haber realizado un estudio a 20 personas que hacen uso del actual sistema de medición de consumo de agua. Las encuestas realizadas y el análisis para la obtención de estos requerimientos se encuentran en el **ANEXO A**. Los Requerimientos Funcionales del sistema se lo detallan en la **Tabla 2.1**.

**Tabla 2.1.** Requerimientos Funcionales del sistema.

MÓDULO	RF	DESCRIPCIÓN
<b>Electrónico</b>	RF01	La placa Arduino debe ser capaz de conectarse a la red inalámbrica del usuario.
	RF02	Debe realizar el ingreso del valor de consumo a la base de datos.
	RF03	El valor de consumo debe ser almacenado en una memoria SD si no existe conexión a internet para el ingreso en la base de datos.
	RF04	El medidor debe funcionar de manera ininterrumpida.
<b>Aplicación Web</b>	RF05	Debe realizar una autenticación mediante un nombre de usuario y contraseña.
	RF06	Debe permitir la administración de los usuarios.
	RF07	Debe permitir la administración de los medidores.
	RF08	Debe permitir realizar consultas de consumos de cada medidor.
	RF09	Debe permitir el envío de notificaciones mediante correos electrónicos.
<b>Aplicación Móvil</b>	RF10	Debe mostrar la información del usuario del medidor.
	RF11	Debe realizar la autenticación mediante el número de serie del medidor.
	RF12	Se debe mostrar el consumo total y consumo de meses anteriores.
	RF13	Debe realizar consultas de los consumos por día, días y meses.

Los Requerimientos Funcionales para el Módulo Electrónico obtenidos y el tiempo para realizarlos se los presentan detallados en la **Tabla 2.2**.

**Tabla 2.2.** Requerimientos Funcionales del Módulo Electrónico.

<b>MÓDULO</b>	<b>RF</b>	<b>HORAS</b>
<b>Electrónico</b>	RF01	4
	RF02	4
	RF03	8
	RF04	4
	Total	20

Los Requerimientos Funcionales para el Módulo Web obtenidos se los presentan detallados en la **Tabla 2.3**.

**Tabla 2.3.** Requerimientos Funcionales del Módulo Web.

<b>MÓDULO</b>	<b>RF</b>	<b>HORAS</b>
<b>Aplicación Web</b>	RF05	5
	RF06	5
	RF07	6
	RF08	4
	RF09	4
	Total	24

Los Requerimientos Funcionales para el Módulo Móvil obtenidos se los presentan detallados en la **Tabla 2.4**.

**Tabla 2.4.** Requerimientos Funcionales del Módulo Móvil.

<b>MÓDULO</b>	<b>RF</b>	<b>HORAS</b>
<b>Aplicación Móvil</b>	RF09	3
	RF10	4
	RF11	8
	RF12	4
	Total	19

## 2.2.4. REQUERIMIENTOS NO FUNCIONALES

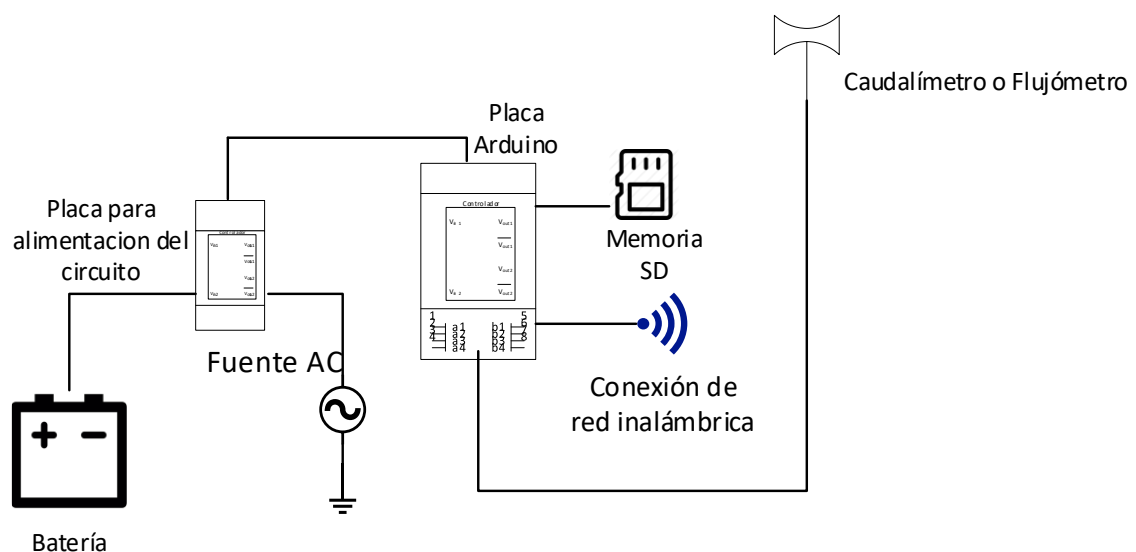
Los Requerimientos No Funcionales (RNF) son propiedades o cualidades que debe tener el sistema, por ejemplo: autenticación, seguridad, disponibilidad, operabilidad, etc. Este tipo de requerimientos para el sistema propuesto se lo detalla en la **Tabla 2.5**.

**Tabla 2.5.** Requerimientos No Funcionales del sistema.

RNF	NOMBRE	DESCRIPCIÓN
RNF01	Autenticación	El acceso al Módulo Web debe estar limitado por un nombre de usuario y una contraseña para cada usuario.
RNF02	Operabilidad	Las aplicaciones del prototipo deben ser de fácil manejo. Deben tener mensajes informativos orientados hacia el usuario.
RNF03	Acceso vía internet	Las aplicaciones del prototipo deben poder ser usadas desde cualquier lugar en el que se encuentren haciendo uso de Internet.
RNF04	Disponibilidad	El módulo y los servicios web deberán estar disponibles 24 horas los 7 días de la semana.

## 2.2.5. DISEÑO DEL MÓDULO ELECTRÓNICO

Para el Módulo Electrónico se determinarán los componentes necesarios para el funcionamiento de la parte de hardware y el diseño del software. El hardware de este módulo se encuentra compuesto de varios componentes los cuales se muestran en la **Figura 2.4**.



**Figura 2.4.** Diagrama de Funcionamiento del Módulo Electrónico.

Cada uno de estos elementos cumplen un propósito específico dentro del Módulo Electrónico; estos propósitos se detallan en la **Tabla 2.6**.

**Tabla 2.6.** Elementos del Módulo Electrónico.

Elemento	Función
Placa de Alimentación del Circuito	Encargada de realizar la alimentación al circuito y de realizar el intercambio entre la energía normal y la energía de la batería; además recarga la batería cuando exista la presencia de la energía normal.
Placa Arduino	Esta placa junto con el chip ESP8266 [37] será la encargada de tener la lógica de funcionamiento de todo el medidor electrónico.
Flujómetro	Este elemento será el encargado de producir pulsos eléctricos para que sean leídos y procesados por la placa Arduino con el fin de obtener el consumo de agua utilizado.
Memoria SD	Encargado de almacenar el valor de consumo cuando no exista conexión hacia internet.
Batería	Encargada de dar respaldo de energía al medidor.

Para el diseño de este módulo se hará uso de un Diagrama de Flujo o también conocido como flujograma, el cual es una forma de representar gráficamente un algoritmo o proceso. Este tipo de diagrama se lo realiza debido a que en el Sistema de Medición de Consumo de Agua Potable en esta parte hace uso de la programación estructurada<sup>18</sup>.

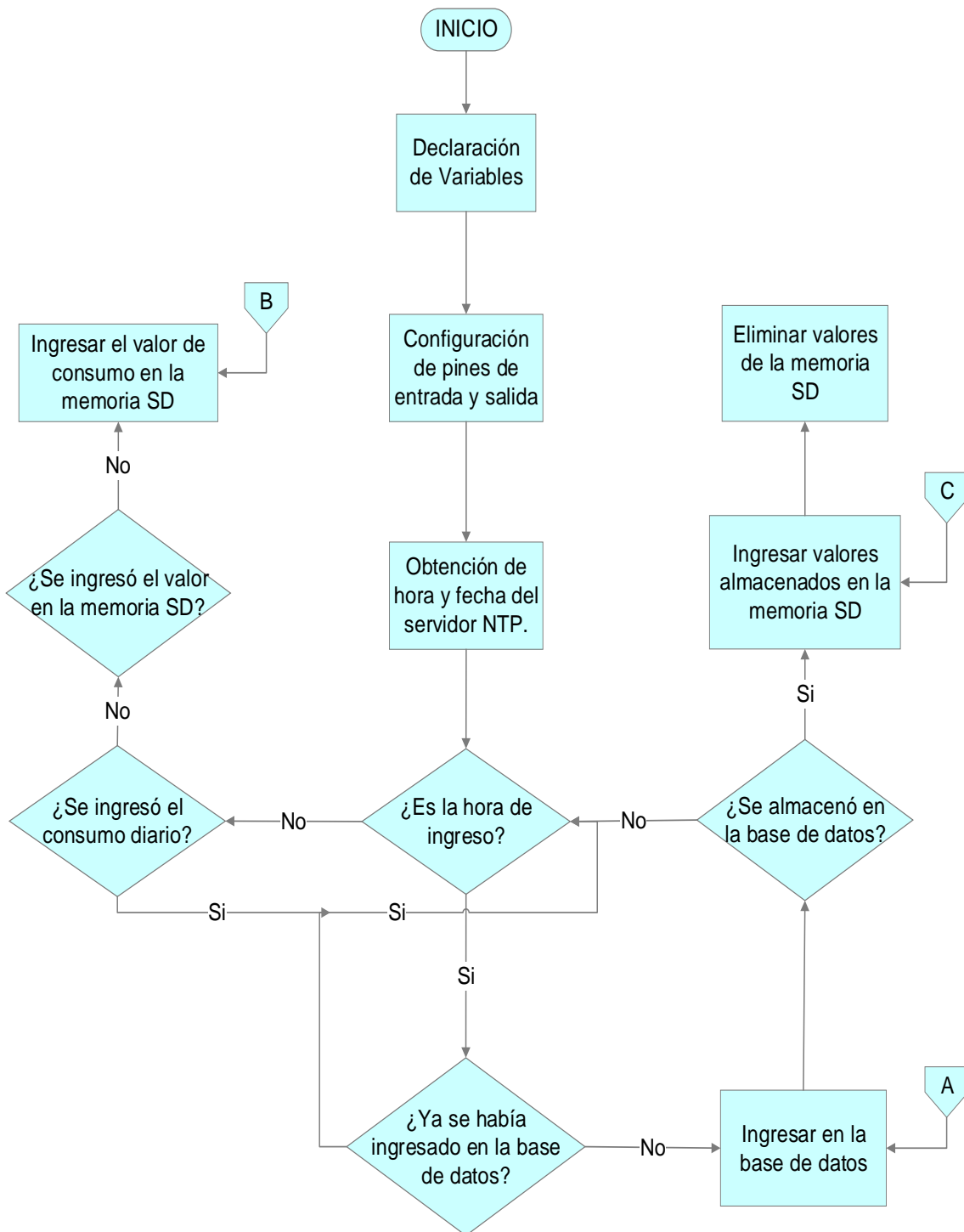
La **Figura 2.5** representa el proceso principal del funcionamiento del módulo, en el cual existen las etiquetas [A, B, C] las cuales representan a funciones cuyo algoritmo de funcionamiento se lo detallará en distintos diagramas de flujo. La etiqueta “A” dentro de la **Figura 2.5** representa al algoritmo para el ingreso del valor de consumo de agua potable en la base de datos; este algoritmo se encuentra detallado en la **Figura 2.6**. Por el contrario, la etiqueta “B” dentro de la **Figura 2.5** representa el algoritmo para almacenar en la memoria SD el valor de consumo que no se haya podido ingresar en la base de datos debido a que no existe conexión a internet, no se ha conectado a una red inalámbrica o si el servicio web para el almacenamiento de valor de consumo no se encuentra disponible durante lapso de una hora (que se encontrará configurado en el Módulo Electrónico para

---

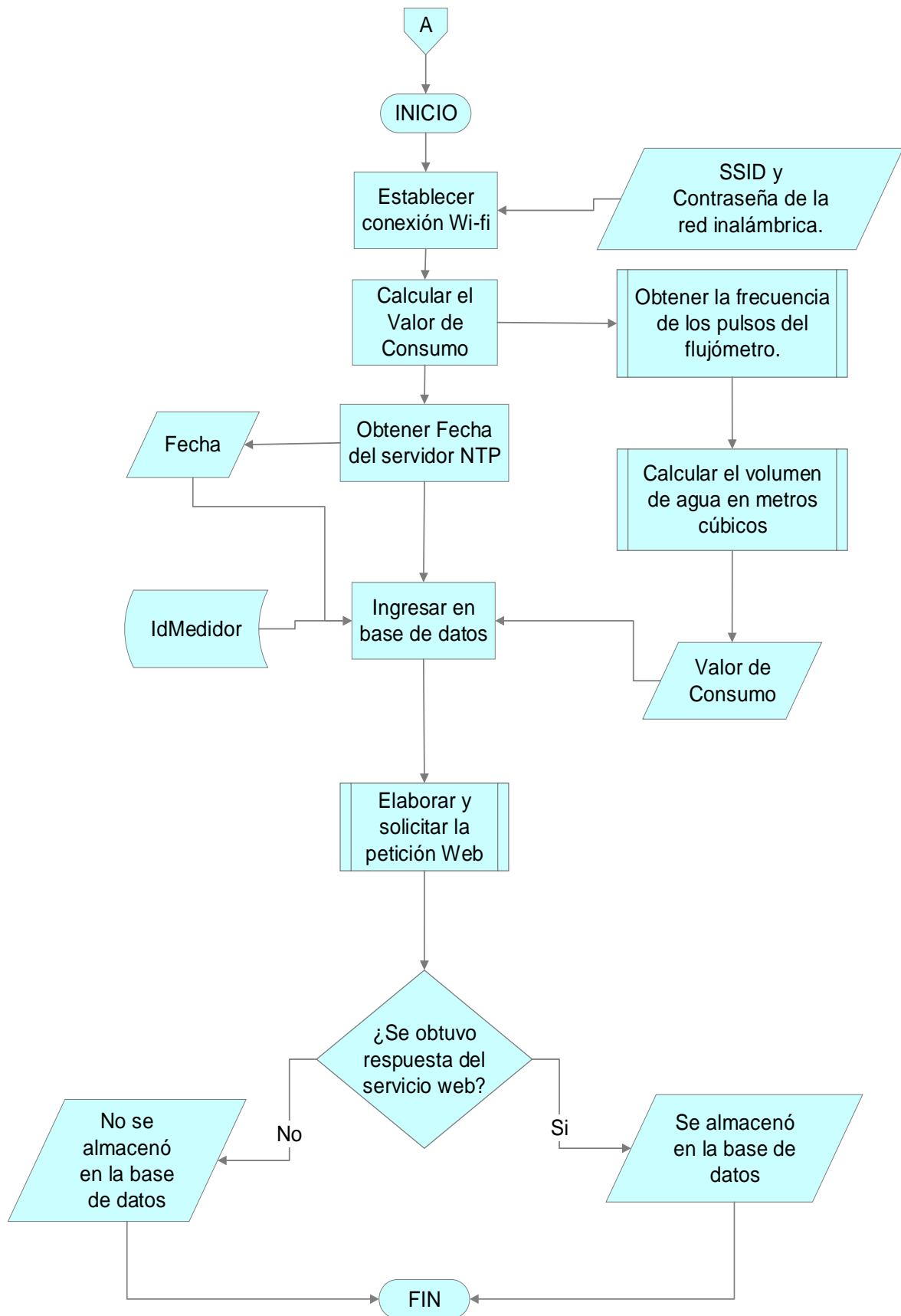
<sup>18</sup> Programación estructurada se refiere a la ejecución de instrucciones sucesivamente una tras otra, pero algunas partes del programa se ejecutan sin depender de que se cumpla alguna condición.



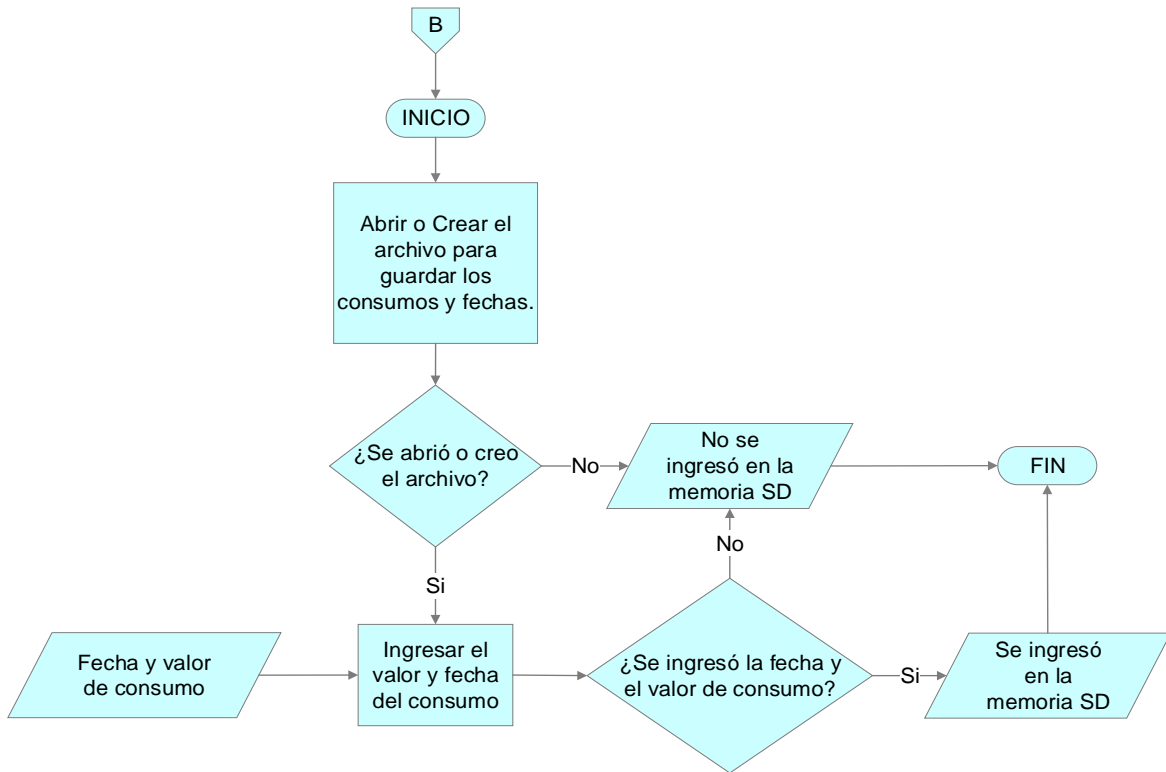
que realice el ingreso del valor de consumo). Este proceso se lo muestra en la **Figura 2.7**. Por último, la etiqueta “C” dentro de la **Figura 2.5** representa el algoritmo para el ingreso de valores de consumo que hayan sido almacenados en la memoria SD del medidor debido a la falta de conexión hacia internet; este proceso se lo muestra en la **Figura 2.8**.



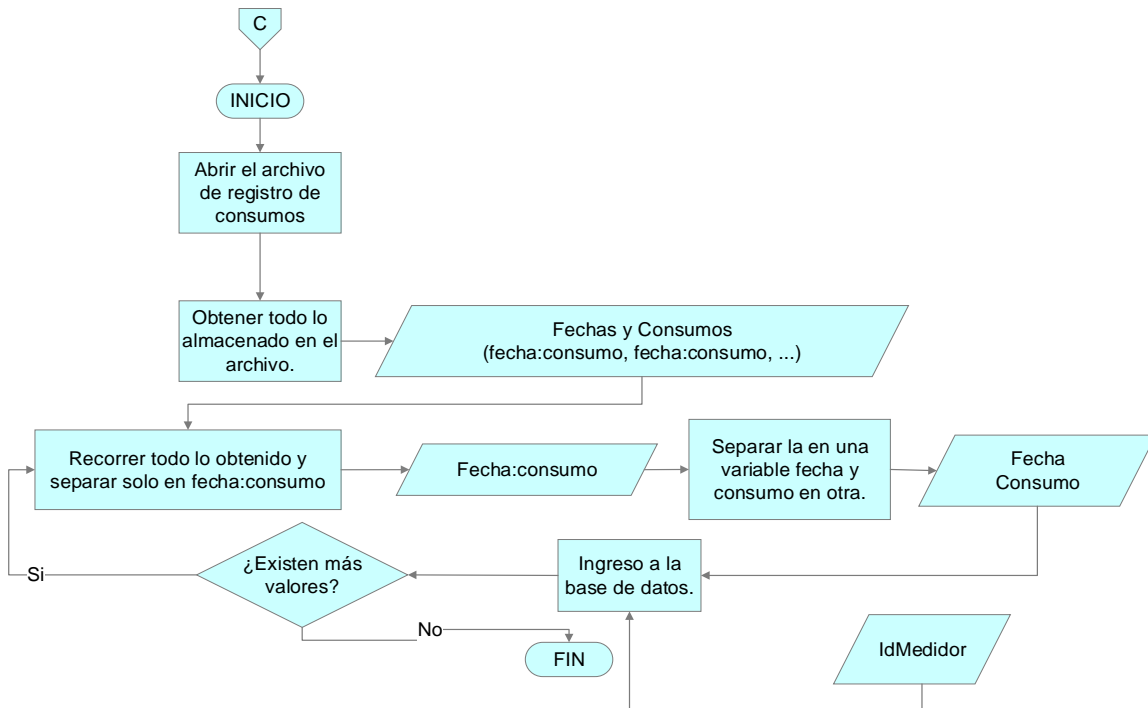
**Figura 2.5.** Algoritmo principal del medidor electrónico.



**Figura 2.6.** Algoritmo de ingreso a la base de datos del valor de consumo.



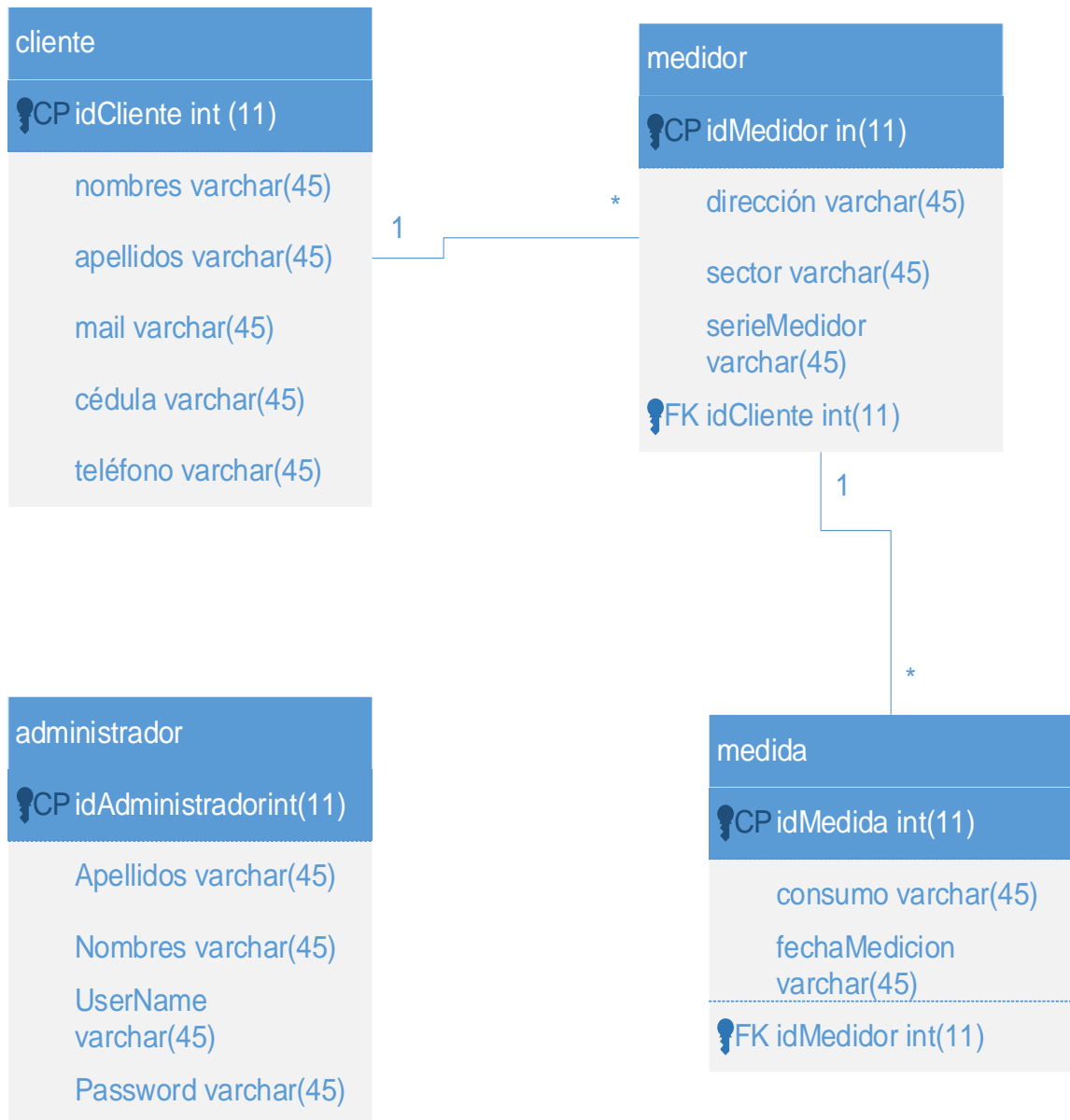
**Figura 2.7.** Algoritmo de ingreso de valores a la memoria SD.



**Figura 2.8.** Algoritmo de ingreso a la base de datos de valores de consumo atrasados.

## 2.2.6. DISEÑO DE LA CAPA DE DATOS

La capa de datos es representada a través del diseño de un diagrama relacional entre entidades. El diseño de la base de datos MySQL se muestra en la **Figura 2.9.**



**Figura 2.9.** Diagrama entidad-relación de la base de datos.

Como se puede observar en la **Figura 2.9** se tiene 4 entidades que intervendrán en el almacenamiento de datos. Cada una de estas entidades poseen campos, entre los cuales están los campos correspondientes a las claves primarias y claves foráneas; las claves primarias son atributos con valores únicos en cada entidad y las claves foráneas son usadas para la relación entre las entidades.

Por ejemplo, la entidad de medidor posee campos: ubicación (dirección, sector), identificación (idMedidor, serieMedidor) y la clave foránea (idCliente). Esta clave foránea se la tiene debido a que un cliente puede tener varios medidores, por lo cual, existe una cardinalidad de “uno a muchos” entre la entidad Cliente y Medidor. Por citar un ejemplo, los campos y el tipo de valor que posee la entidad Medidor se los detallan en la **Tabla 2.7**.

**Tabla 2.7.** Descripción de la entidad Medidor.

<b>CAMPO</b>	<b>TIPO</b>	<b>DESCRIPCIÓN</b>
<b>idMedidor</b>	Número Entero	Campo usado como identificador único dentro del sistema para cada medidor.
<b>Dirección</b>	Cadena de caracteres	Campo usado para determinar la dirección donde se encuentra ubicado el medidor.
<b>Sector</b>	Cadena de caracteres	Campo usado para determinar el sector específico donde se encuentra el medidor, este campo se requiere ya que la dirección puede repetirse.
<b>serieMedidor</b>	Cadena de caracteres	Campo usado como identificador único del medidor para el cliente.
<b>idCliente</b>	Número Entero	Campo usado para determinar el cliente propietario del medidor.

**Tabla 2.8.** Descripción de la entidad cliente.

<b>CAMPO</b>	<b>TIPO</b>	<b>DESCRIPCIÓN</b>
<b>idCliente</b>	Número entero	Campo usado como identificador único dentro del sistema del para cada cliente.
<b>Nombres</b>	Cadena de caracteres	Campo usado para almacenar los nombres del cliente.
<b>Apellidos</b>	Cadena de caracteres	Campo usado para almacenar los apellidos del cliente.
<b>Mail</b>	Cadena de caracteres	Campo usado para almacenar el correo electrónico del cliente, este correo es el que se usara para enviar las notificaciones.
<b>Cédula</b>	Número entero	Campo usado para almacenar el número de cedula del cliente.
<b>Teléfono</b>	Cadena de caracteres	Campo para almacenar un numero de contacto del cliente.

**Tabla 2.9.** Descripción de la entidad medida.

<b>CAMPO</b>	<b>TIPO</b>	<b>DESCRIPCIÓN</b>
<b>idMedida</b>	Número entero	Campo usado como identificador único dentro del sistema para cada medida realizada.
<b>idMedidor</b>	Número entero	Campo usado para determinar a qué medidor pertenece la medida ingresada.
<b>Consumo</b>	Cadena de caracteres	Campo usado almacenar el consumo realizado.
<b>fechaMedicion</b>	Cadena de caracteres	Campo usado determinar en qué fecha se ha realizado la medición.

**Tabla 2.10.** Descripción de la entidad administrador.

<b>CAMPO</b>	<b>TIPO</b>	<b>DESCRIPCIÓN</b>
<b>idAdministrador</b>	Número entero	Campo usado como identificador único dentro del sistema para cada administrador.
<b>Apellidos</b>	Cadena de caracteres	Campo usado para almacenar los nombres del administrador.
<b>Nombres</b>	Cadena de caracteres	Campo usado para almacenar los apellidos del administrador.
<b>userName</b>	Cadena de caracteres	Campo usado como identificador para el <i>login</i> del administrador.
<b>Password</b>	Cadena de caracteres	Campo usado para determinar la contraseña del administrador, este campo junto con el userName será usados para el <i>login</i> del administrador.

## **2.2.7. DISEÑO DE LA CAPA DE NEGOCIO**

El diseño de esta capa se basa en mostrar la lógica de los diferentes procesos que se aplicarán en el software a utilizarse en el Sistema de Medición de Consumo de Agua Potable. Para definir cada uno de los procesos se hará el uso de diagramas de casos de uso, diagramas de clases y diagrama del servicio web.

### **2.2.7.1. DIAGRAMAS DE CASOS DE USO**

Un diagrama de caso de uso es la descripción de las acciones o funciones que provee el sistema desde el punto de vista del usuario. A través de este tipo de diagramas se detalla el papel que desempeñan los usuarios dentro del Sistema de Medición de Consumo de Agua Potable. Este sistema divide a los usuarios en cuatro tipos de actores los cuales son: Administrador, Cliente, Usuario Sistema y Usuario Medidor.

El Administrador es aquel que se encuentra encargado de realizar la gestión (creación, lectura, actualización o eliminación) de usuarios, medidores y, además, de poder realizar consultas de consumo de cada uno de los medidores del sistema. Este usuario hace uso del Módulo Web.

El Cliente es aquel que hace uso del Módulo Electrónico o del Módulo Móvil para medir o verificar el consumo de agua respectivamente.

El Usuario Sistema es aquel que se encargará de realizar consultas de consumo, calcular el consumo de los meses de cada medidor o medidores de cada uno de los usuarios para el envío de las notificaciones de consumo al Cliente, a través de un correo electrónico el cual es enviado al final de cada mes.

El Usuario Medidor es aquel que interviene en el sistema realizando la medición y el ingreso del consumo de agua en la base de datos o también lo almacena en una memoria SD si existen fallas en la conexión hacia internet, disponibilidad de los servicios web, etc.

Los distintos tipos de usuarios antes detallados y las actividades que realizan dentro del Sistema de Medición de Consumo de Agua Potable se las detallan en la Tabla 2.11.

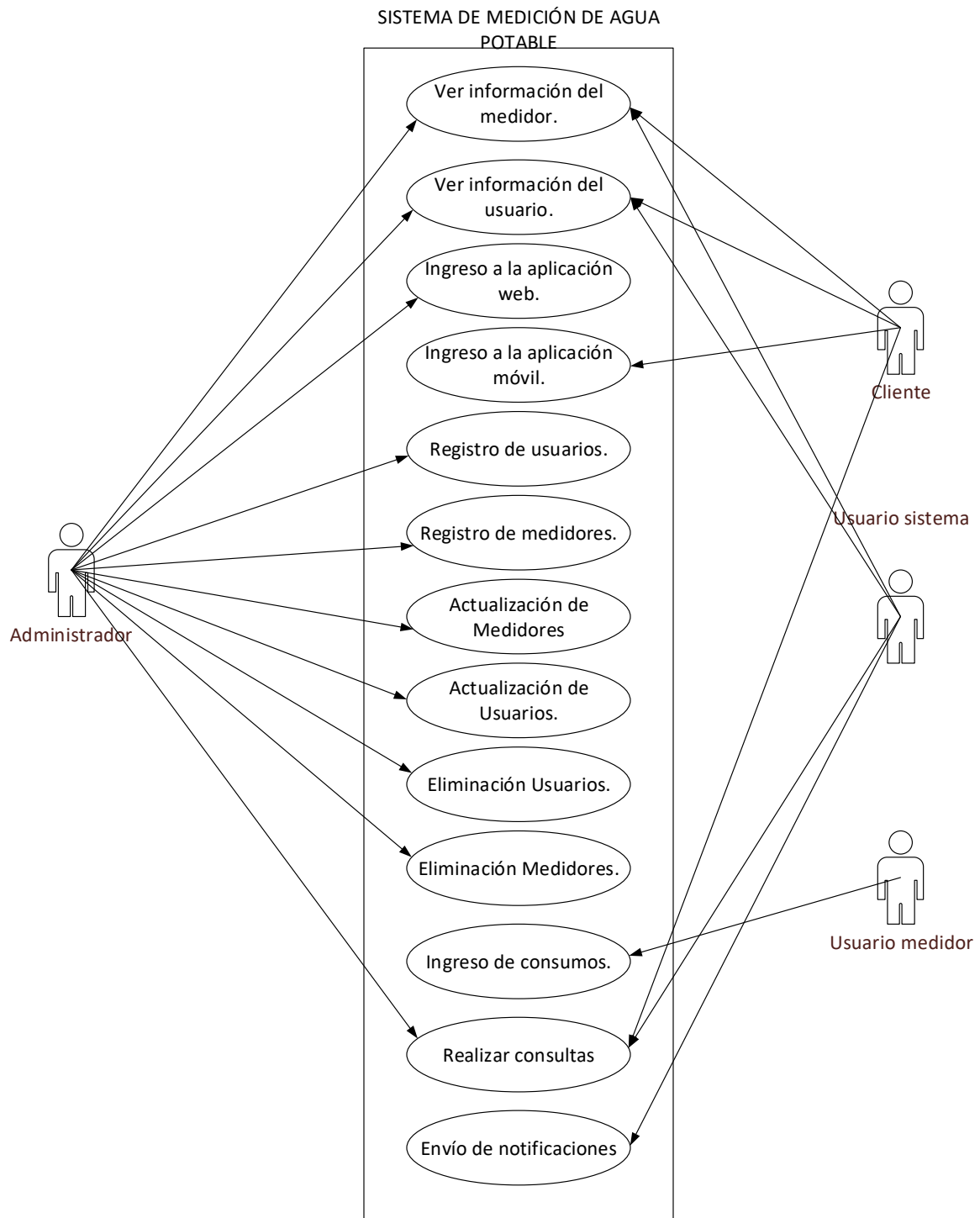
**Tabla 2.11.** Actividades del usuario dentro del sistema.

MÓDULO	ACTIVIDAD	CLIENTE	ADMINIS TRADOR	USUARIO SISTEMA	USUARIO MEDIDOR
Móvil	Instalación de la aplicación móvil.	X	X	X	-
	Uso de la Aplicación Móvil	X	X	X	-
	Realización de consultas.	X	X	X	-
Web	Uso de la aplicación web.	-	X	X	-
	Administración de Usuarios.	-	X	X	-
	Administración de Medidores.	-	X	X	-
	Realizar consultas.	-	X	X	-
	Envío de notificaciones.	-	-	X	-
Electrónico	Ingreso del consumo de agua.	-	-	X	X
	Medición del consumo de agua.	-	-	X	X

Los Diagramas de Casos de Uso del sistema, involucran a todos los tipos de usuarios y todas las acciones que se pueden realizar dentro del sistema. A cada usuario se lo relaciona únicamente con las actividades que estos pueden realizar.

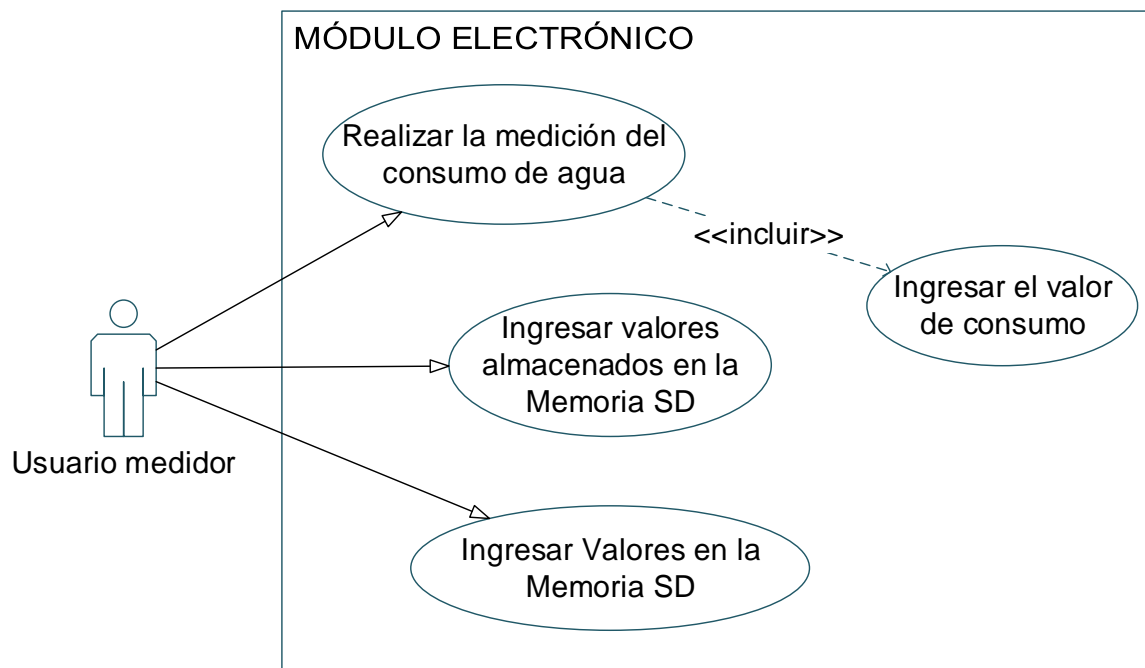
El Diagrama de Casos de Uso general del sistema se lo muestra en la **Figura 2.10**.





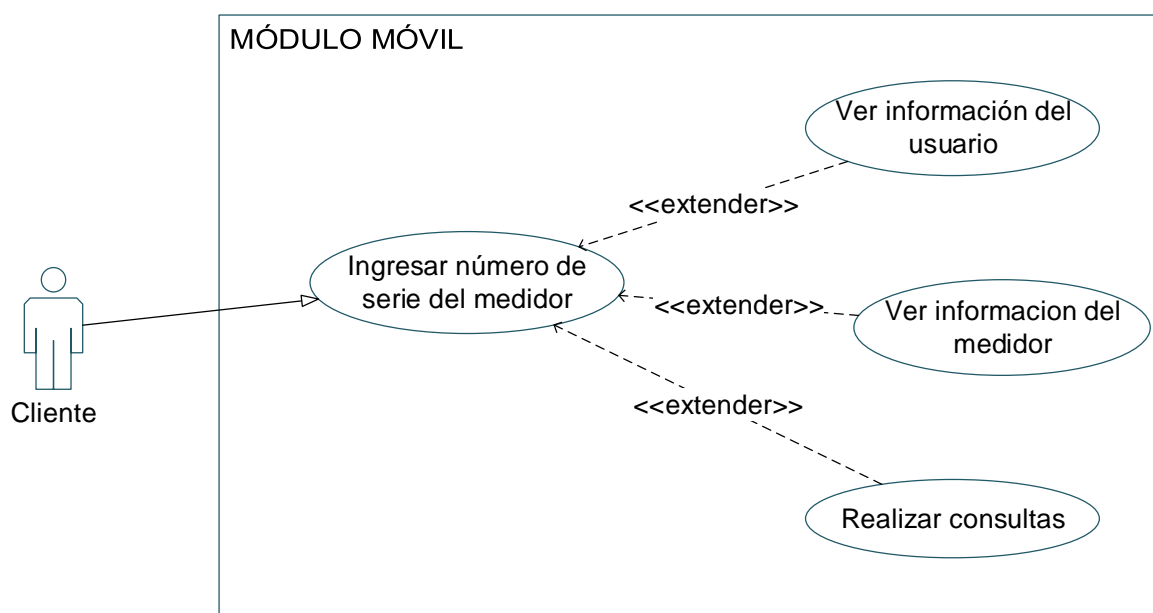
**Figura 2.10.** Diagrama de casos de uso del sistema.

Además de un diagrama de casos de uso general se realiza un diagrama más específico para cada uno de los módulos del prototipo. Para el Módulo Electrónico el diagrama de casos de uso se lo muestra en la **Figura 2.11**.



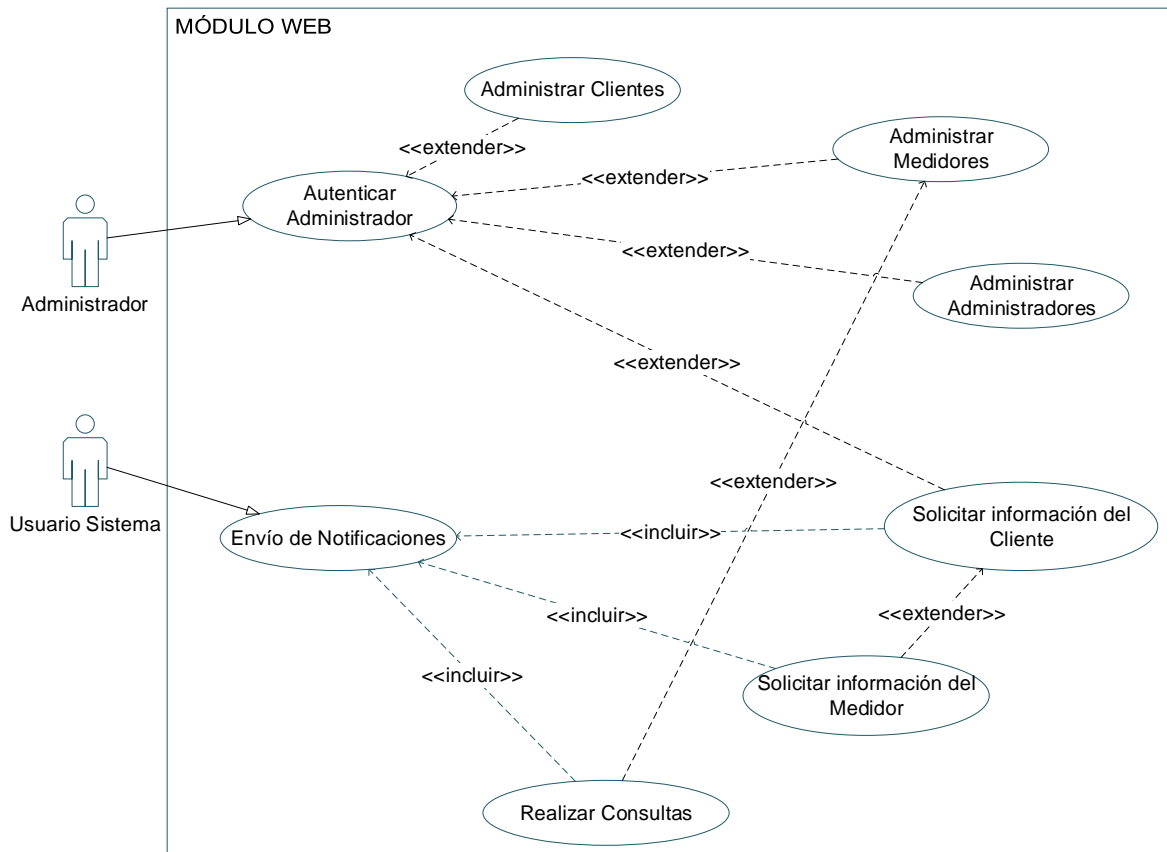
**Figura 2.11.** Diagrama de casos de uso del Módulo Electrónico.

Para el Módulo de la Aplicación Móvil el diagrama de casos de uso se los muestra en la **Figura 2.12.**



**Figura 2.12.** Diagrama de casos de uso módulo de la Módulo Móvil.

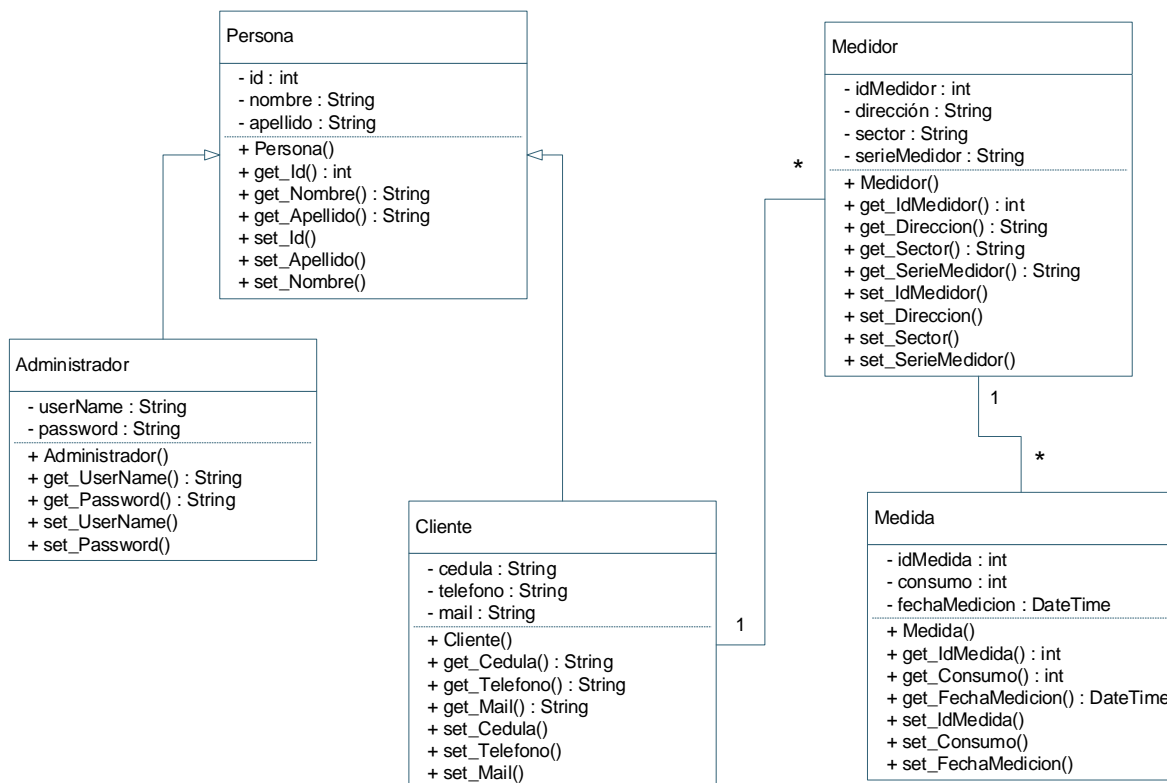
Por último, el diagrama de casos de uso para el Módulo Web se muestra en la **Figura 2.13.** En este módulo interactúan dos usuarios: el Usuario Administrador y el Usuario Sistema.



**Figura 2.13.** Diagrama de casos de uso del Módulo Web.

### 2.2.7.2. DIAGRAMA DE CLASES

El objetivo principal del diagrama de clases es representar la estructura general del sistema, mostrando cada una de las clases y la interacción entre ellas (herencias, asociaciones, etc.). Cada clase es un objeto, por lo cual debe poseer la información necesaria para describir al mismo. El Sistema de Medición de Consumo de Agua Potable dispone de varias clases para su funcionamiento, las cuales se muestran en la **Figura 2.14**. En el diagrama existen distintos tipos interacciones entre las clases, entre las cuales se tiene la relación de herencia de la clase Persona con las clases Administrador y Cliente; esta relación implica que estas dos clases poseerán los atributos y métodos de la clase Persona. Así también, se tiene una relación entre Cliente y Medidor de “1 a N” lo que indica que un Cliente puede tener varios Medidores; y, por último, se tiene la relación de “1 a N” entre las clases Medidor y Medición, lo cual implica que el medidor tendrá varias mediciones asociadas a él.



**Figura 2.14.** Diagramas de clases del sistema.

Cada una de estas clases tienen sus propios atributos y métodos que son diferentes a las otras clases, por ejemplo, la clase Medidor posee atributos de diferentes tipos de datos y diferentes tipos de acceso los cuales se detalla en la **Tabla 2.12**. Además, se tiene los métodos “*get*” y “*set*” para cada uno de los atributos, los cuales servirán para conseguir y establecer el valor del atributo respectivamente.

**Tabla 2.12.** Descripción de los atributos de la clase Medidor.

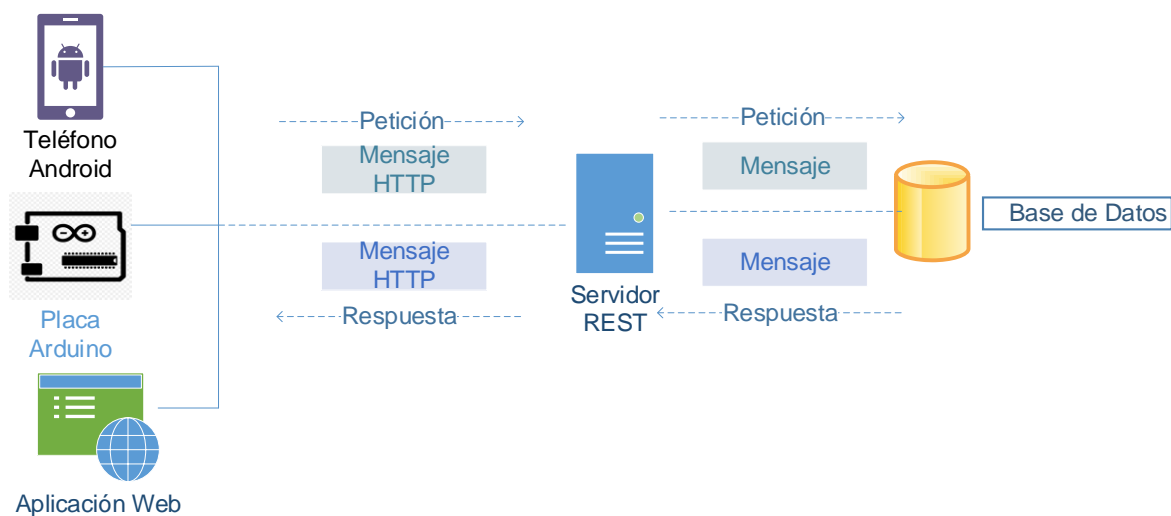
ATRIBUTO	TIPO DE DATO	TIPO DE ACCESO	DESCRIPCIÓN
<b>idMedidor</b>	Número entero	Privado (-)	Identificador único del medidor dentro del sistema.
<b>Dirección</b>	Cadena de caracteres.	Privado (-)	Atributo usado para identificar la dirección de la ubicación del medidor.
<b>Sector</b>	Cadena de caracteres.	Privado (-)	Atributo usado para identificar el sector de la ubicación del medidor.
<b>serieMedidor</b>	Cadena de caracteres.	Privado (-)	Identificador único del medidor para él usuario.

Las demás clases se las detallan en el **ANEXO B**.

### 2.2.7.3. DIAGRAMA DEL SERVICIO WEB

Los servicios web usados para el sistema de medición de consumo de agua son del tipo REST los cuales han sido detallados en el **Capítulo I Apartado 1.3.2**.

Para el funcionamiento de este tipo de servidor el usuario deberá conocer la dirección IP del servidor o el nombre de dominio; además de conocer la estructura de las URIs para la solicitud de los diferentes servicios. El funcionamiento del servicio tipo REST se presenta en la **Figura 2.15**.



**Figura 2.15.** Funcionamiento de un servicio web.

Cada uno de los servicios web serán diseñados siguiendo el funcionamiento de la **Figura 2.15**, y estos contendrán las acciones necesarias para el cumplimiento de los Requerimientos Funcionales del sistema, por lo cual, cada servicio web realizará una acción (leer, escribir, editar o borrar) en la base de datos.

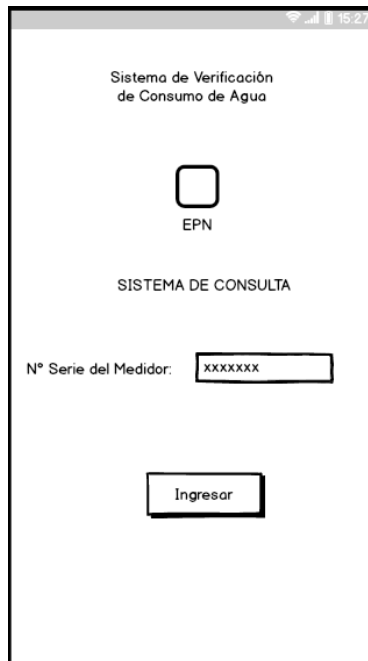
### 2.2.8. DISEÑO DE LA CAPA DE PRESENTACIÓN

En este apartado se diseñará las interfaces de las aplicaciones web y móvil, con las cuales interactúa el usuario para complementar los Requerimientos Funcionales detallados en el **Capítulo 2 Apartado 2.2.3**. Para lo cual se hará uso de la aplicación Balsamiq Wireframes, la cual es una herramienta que permite el diseño rápido de interfaces de usuario [43].

#### 2.2.8.1. INTERFAZ DEL MÓDULO MÓVIL

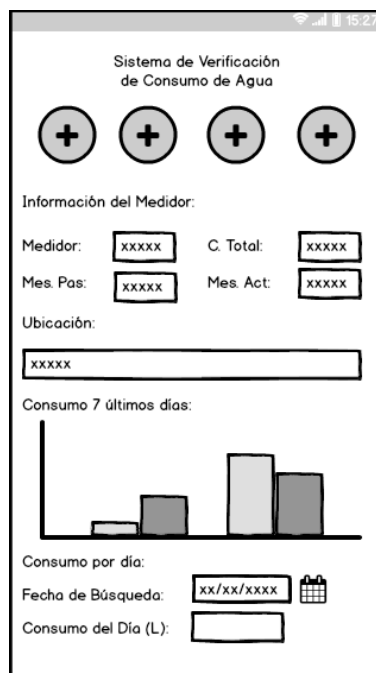
Para el Módulo Móvil se bosquejarán 5 pantallas con el fin de cumplir las acciones establecidas en los Requerimientos Funcionales. El primer bosquejo de la aplicación móvil se muestra en la **Figura 2.16**, el cual será usado para realizar el ingreso a las funcionalidades de la aplicación. En esta pantalla se requiere que el usuario haga el ingreso del número de serie del medidor, el cual será único. El objetivo de realizar este tipo de

ingreso es que aplicación pueda ser usada para realizar la consulta de varios medidores de los cuales se tenga en número de serie.



**Figura 2.16.** Bosquejo de la pantalla para consultar un medidor.

El siguiente bosquejo se muestra en la **Figura 2.17**, la cual es la pantalla principal de la aplicación en donde se mostrará la información básica del medidor y los últimos consumos que se han registrado en el sistema. Además, se podrá realizar consultas del consumo de agua potable por día.



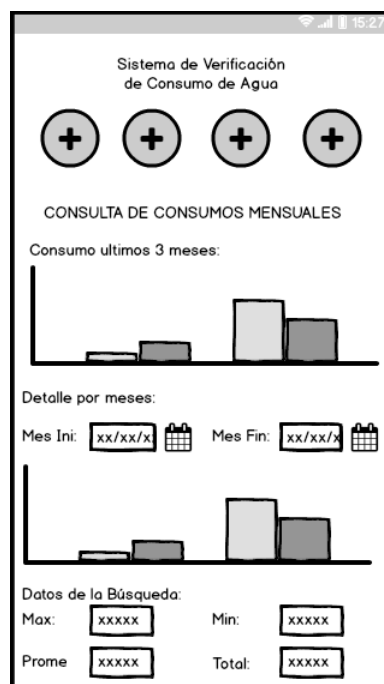
**Figura 2.17.** Bosquejo de la pantalla principal de la aplicación móvil.

El tercer bosquejo muestra la pantalla en donde se presenta la información del cliente propietario del medidor. Este bosquejo se la muestra en la **Figura 2.18**.



**Figura 2.18.** Bosquejo de la pantalla de los datos del usuario.

En el cuarto bosquejo se muestra la pantalla donde se realizarán las consultas mes a mes del consumo de agua del medidor y donde también, se observarán los valores como consumo máximo, mínimo, promedio y la suma total de los consumos realizados en los meses en los que se realice la consulta. Esta interfaz se muestra en la **Figura 2.19**.



**Figura 2.19.** Bosquejo para la consulta de consumos por meses.

Por último, el quinto bosquejo muestra la pantalla en donde el usuario podrá realizar consultas de una fecha a otra, escoger el tipo de diagrama (*Barchar*, *LineChart*, *PointChart*, *RadialGaugeChart*, *DonutChart*, *RadarChart*) para observar los consumos; y además de observar valores de consumo máximo, mínimo, promedio y la suma total de los consumos de los días que se haya realizado la consulta. Esta interfaz se muestra en la **Figura 2.20**.

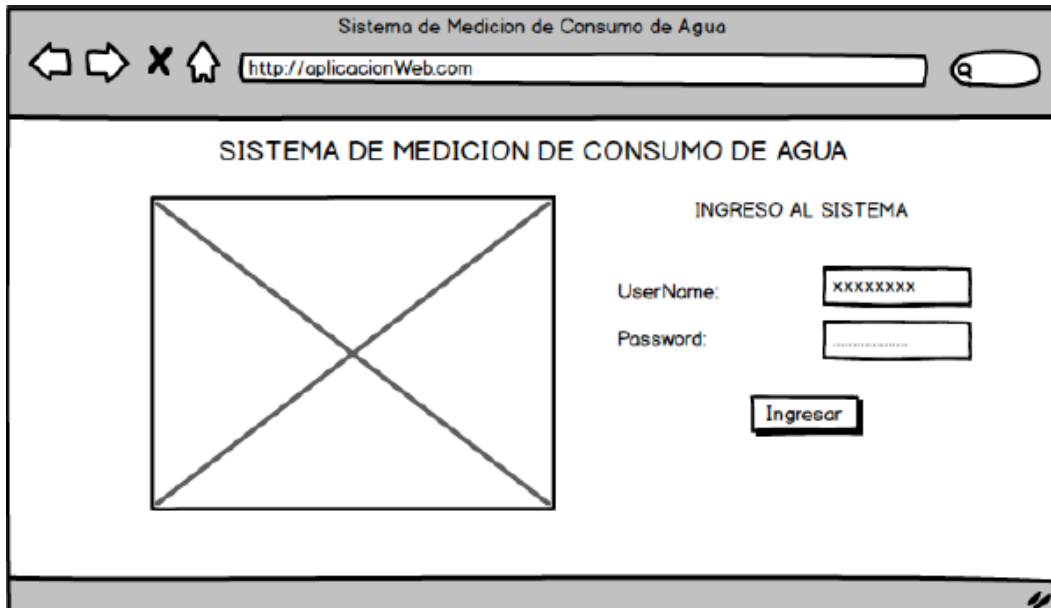


**Figura 2.20.** Interfaz de consulta por días.

### 2.2.8.2. INTERFAZ DEL MÓDULO WEB

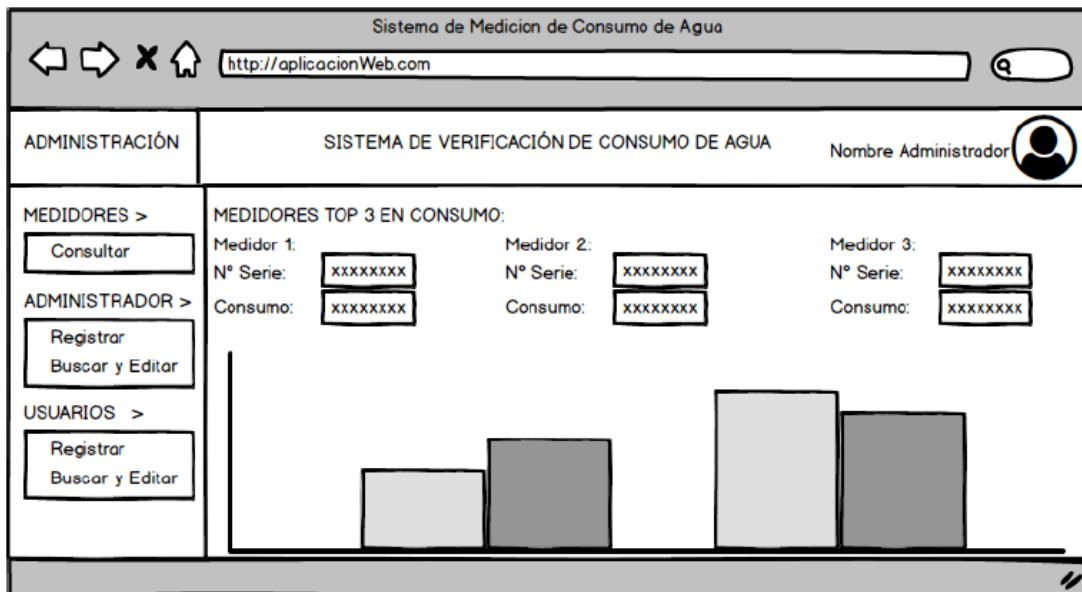
El Módulo web no requiere instalación, el usuario administrador puede hacer uso de esta aplicación a través de un navegador web, para lo cual el usuario debe conocer el nombre de dominio en donde se encuentre la aplicación. Una vez que el Administrador ingrese el nombre del dominio o URL, ejemplo, “*www.apliacionweb.com*”, se le presentará una página web en donde el Administrador deberá realizar el ingreso de los parámetros requeridos para realizar la autenticación e ingreso al sistema. El bosquejo de la pantalla para realizar la autenticación del usuario se muestra en la **Figura 2.21**.





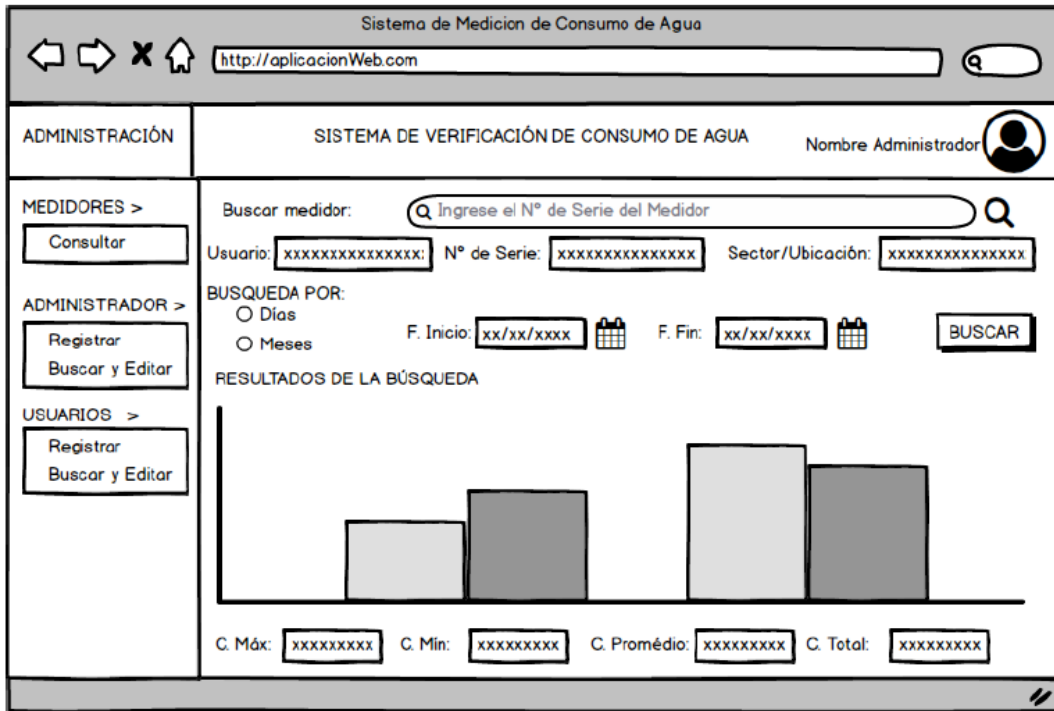
**Figura 2.21.** Bosquejo de la pantalla de autenticación del Módulo Web.

Después de haber realizado la autenticación se le presentará, al Administrador, la interfaz de la **Figura 2.22**; en la cual se muestra los medidores que tengan el consumo más elevado hasta ese momento.



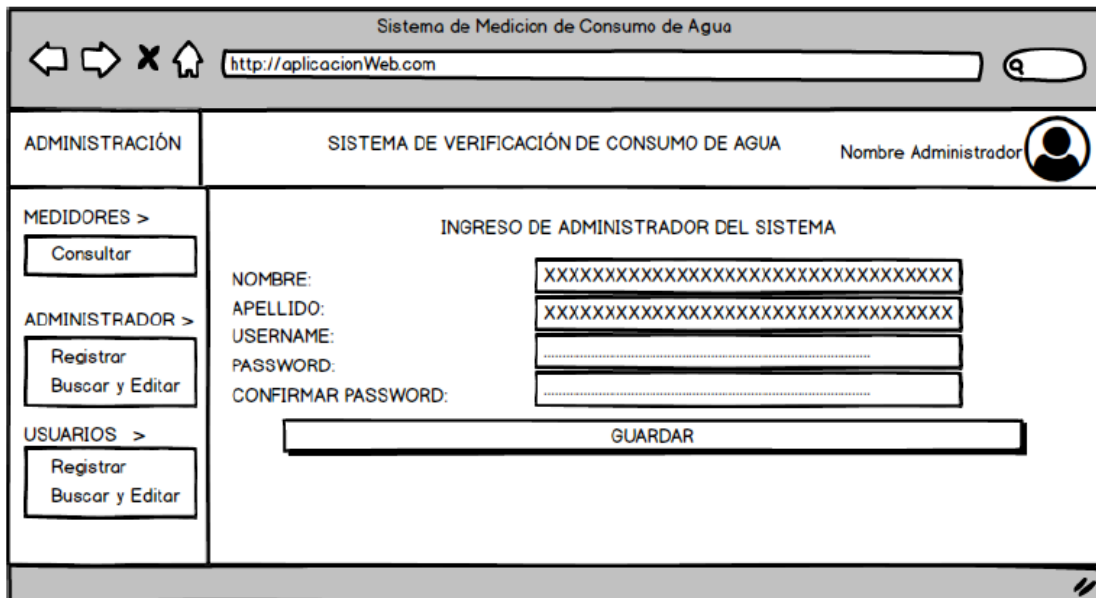
**Figura 2.22.** Bosquejo de la pantalla principal del Módulo Web.

Desde esta interfaz el Administrador podrá realizar la navegación a través del sistema mediante un panel situado en el lado izquierdo en donde se encuentra las diferentes opciones. Una de las opciones es “MEDIDORES” en donde se encuentra la subopción “Consultar” en la cual se podrá realizar la consulta de consumos de los distintos medidores registrados. Una vez se ingrese a esta opción se mostrará la interfaz de la **Figura 2.23**.



**Figura 2.23.** Bosquejo de la pantalla para la consulta de consumos.

La siguiente opción que se muestra es la de “ADMINISTRADOR” en la cual, se podrá realizar el registro de nuevos administradores, así como también de buscar, editar o eliminar los mismos. Para la subopción de “Registrar” se tiene la interfaz que se muestra en la **Figura 2.24.** y es en donde el Administrador registrará a nuevos Administradores.



**Figura 2.24.** Bosquejo de la pantalla para el ingreso de administradores.

Para la subopción de “Buscar y Editar” dentro de “ADMINISTRADOR” se tiene la interfaz que se muestra en la **Figura 2.25.** Aquí el Administrador buscará, editará o eliminará a Administradores que ya no formen parte del sistema.

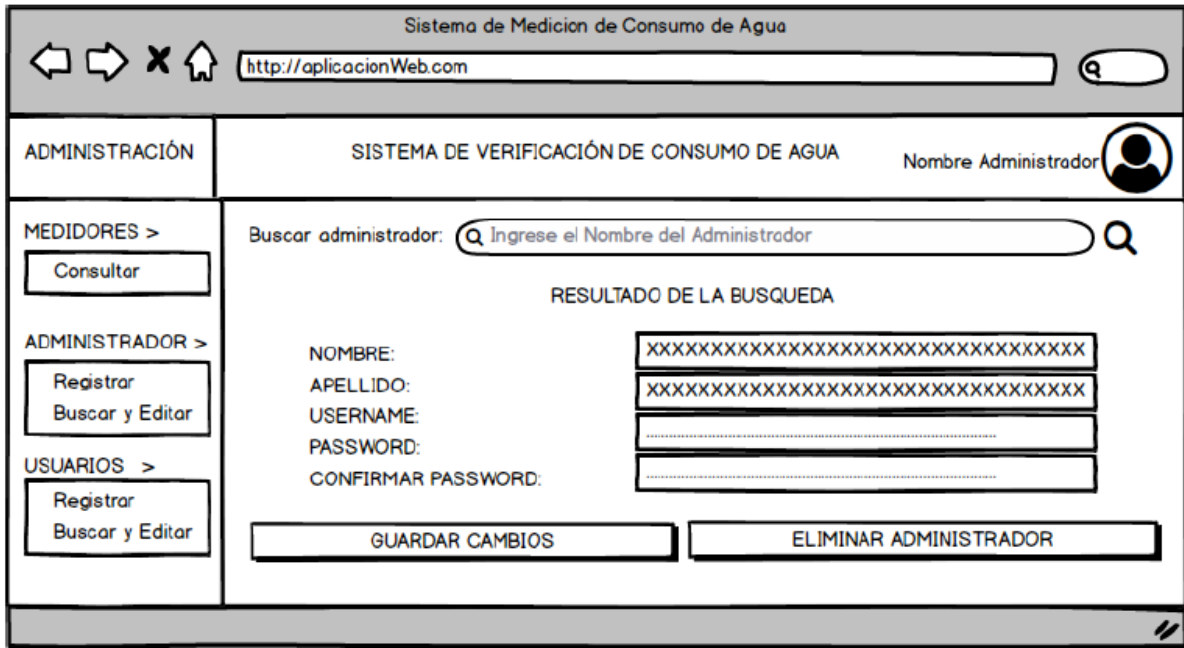


Figura 2.25. Bosquejo de la pantalla para editar administradores.

Por último, se tiene la opción de “USUARIOS” en la cual el Administrador tiene las acciones de registrar nuevos clientes, así como también la opción de buscar y editar los mismos. Para la subopción de “Registrar” se tiene la interfaz que se muestra en la **Figura 2.26**.

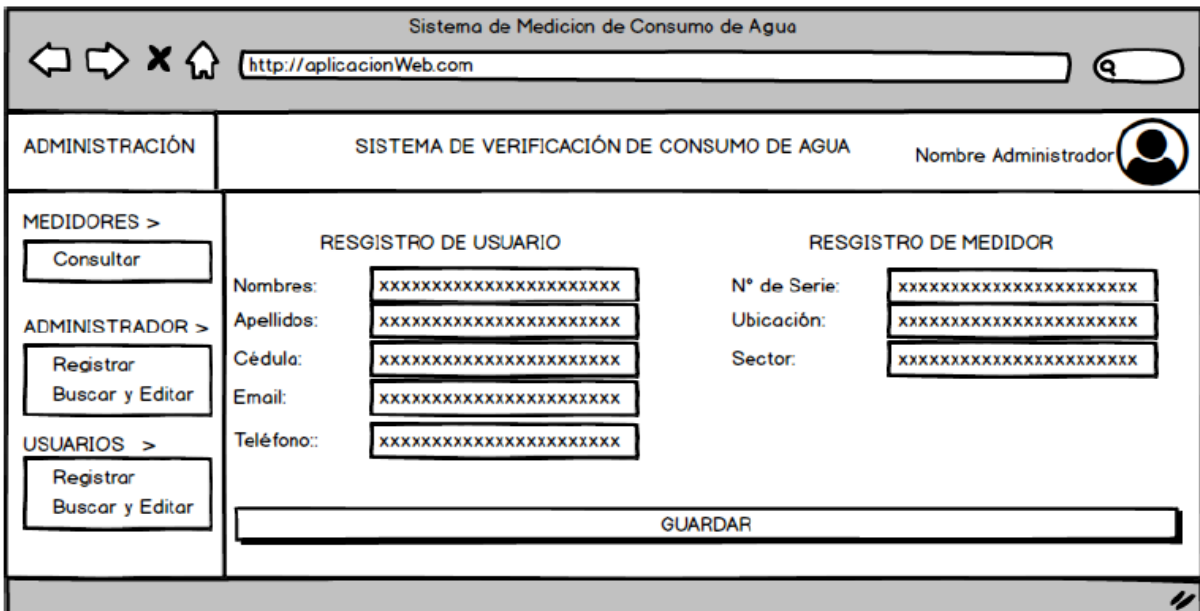
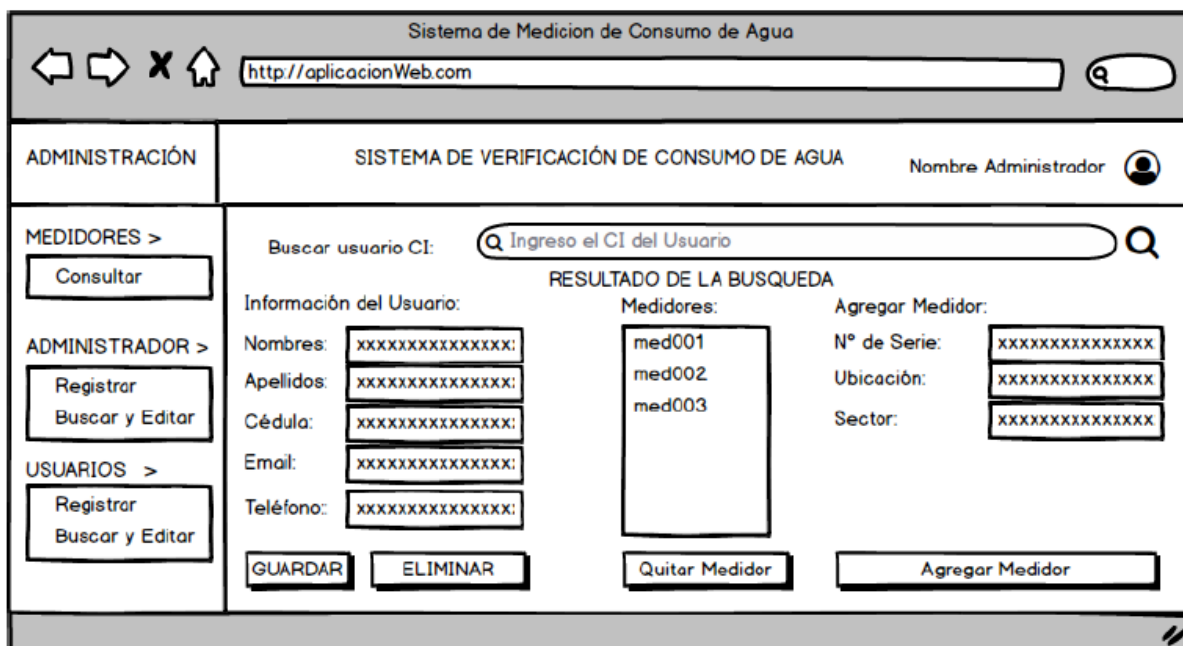


Figura 2.26. Interfaz para el Registro de Usuarios.

En esta interfaz el Administrador deberá realizar el ingreso de la información del usuario y del medidor electrónico asociado al usuario para completar el registro de un nuevo usuario. Por último, se tiene la subopción de “Buscar y Editar” dentro de la opción “USUARIOS” para lo cual el Administrador tiene la interfaz mostrada en la **Figura 2.27**.



**Figura 2.27.** Interfaz para la modificación de clientes.

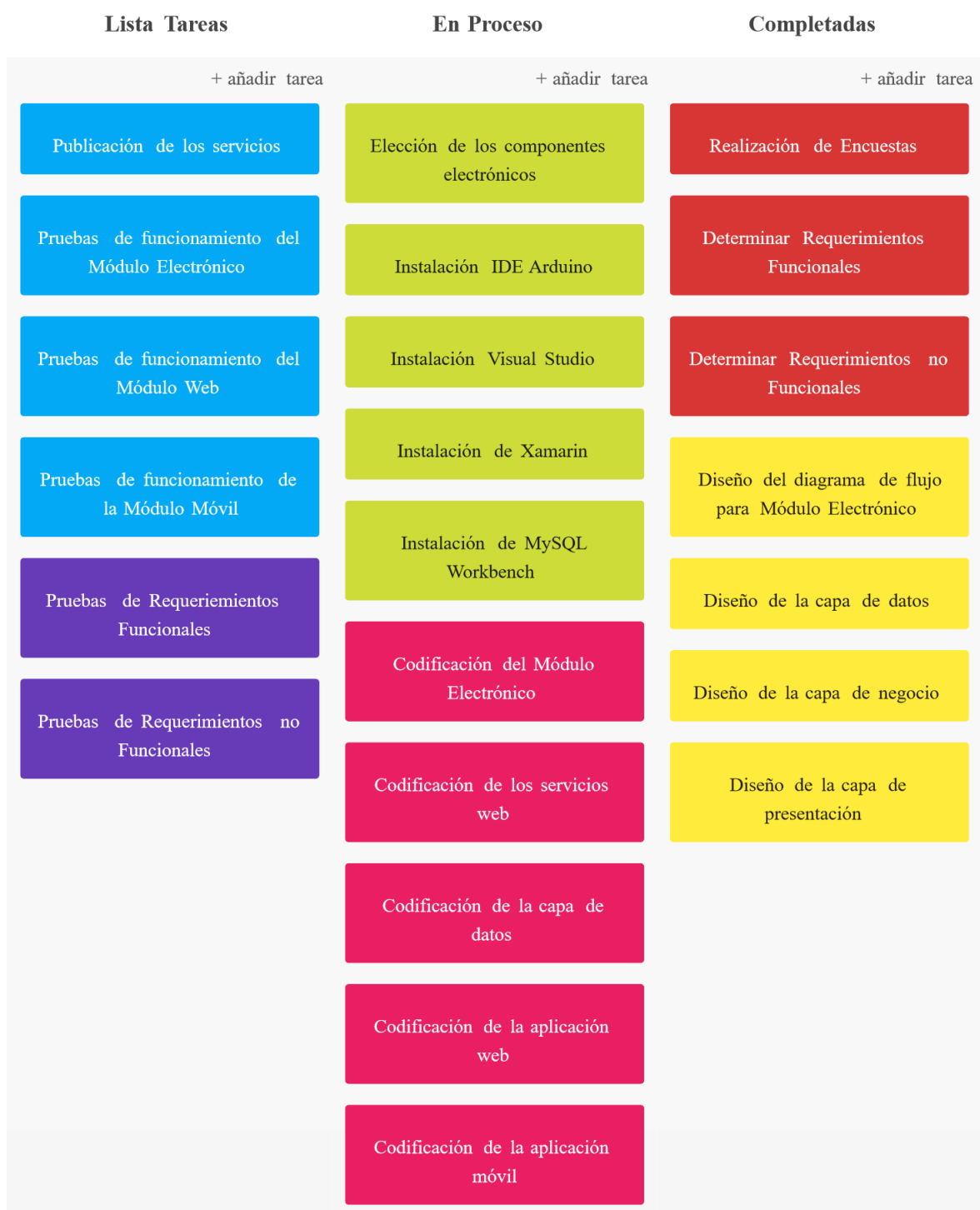
En esta interfaz el Administrador tiene las opciones de modificar la información del usuario, quitar o agregar medidores a él.

## 2.3. IMPLEMENTACIÓN

En este apartado se detallará el procedimiento de implementación de los tres módulos que intervienen en el sistema. En primer lugar, se realizará la instalación de las herramientas necesarias para el desarrollo de los módulos y posteriormente, siguiendo el diseño propuesto en el **Capítulo 2 Apartado 2.2.**, se realizará la implementación de los mismos.

### 2.3.1. ACTUALIZACIÓN DEL TABLERO KANBAN

El tablero Kanban de la **Figura 2.28**. muestra las tareas para la fase de implementación que se deben realizar para el cumplimiento de los Requerimientos Funcionales de los tres módulos.



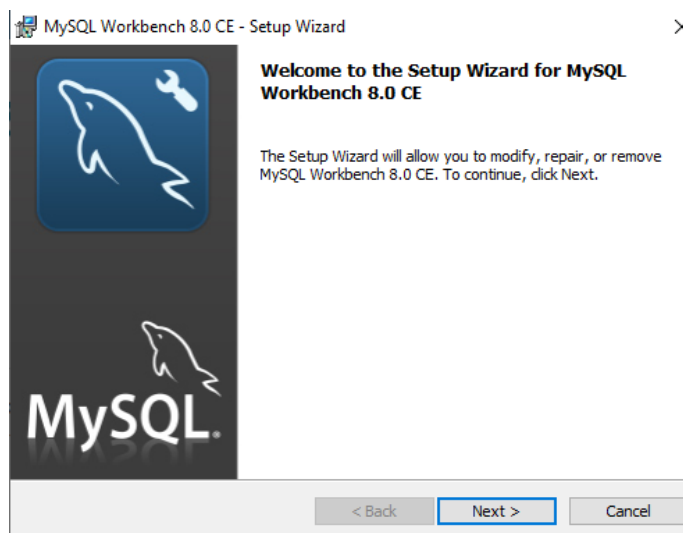
**Figura 2.28.** Tablero Kanban implementación.

## 2.3.2. INSTALACIÓN DE HERRAMIENTAS SOFTWARE REQUERIDAS

### 2.3.2.1. INSTALACIÓN DE MYSQL WORKBENCH

Para la instalación de la herramienta de MySQL Workbench se debe descargar el instalador desde su página oficial [44]. Una vez descargado se lo ejecuta y mostrará la pantalla de la

**Figura 2.29**, en la cual se deberá seleccionar la opción “Next” para proceder con la instalación. Se deberá seguir los pasos del “Asistente de Instalación” hasta finalizar con la instalación. Una vez finalizado el proceso se podrá hacer uso de las herramientas y los beneficios que se detallaron en el **Capítulo I Apartado 1.3.7.3**.



**Figura 2.29.** Instalación MySQL Workbench.

### 2.3.2.2. INSTALACIÓN DE VISUAL STUDIO Y XAMARIN

Para la instalación de Visual Studio se debe descargar el instalador de la página oficial de Microsoft [28]. Una vez descargado el instalador se lo ejecuta y siguiendo los pasos del “Asistente de Instalación” aparecerá la ventana de la **Figura 2.30**. donde se debe escoger las cargas de trabajo.



**Figura 2.30.** Cargas de trabajo instaladas en Visual Studio.

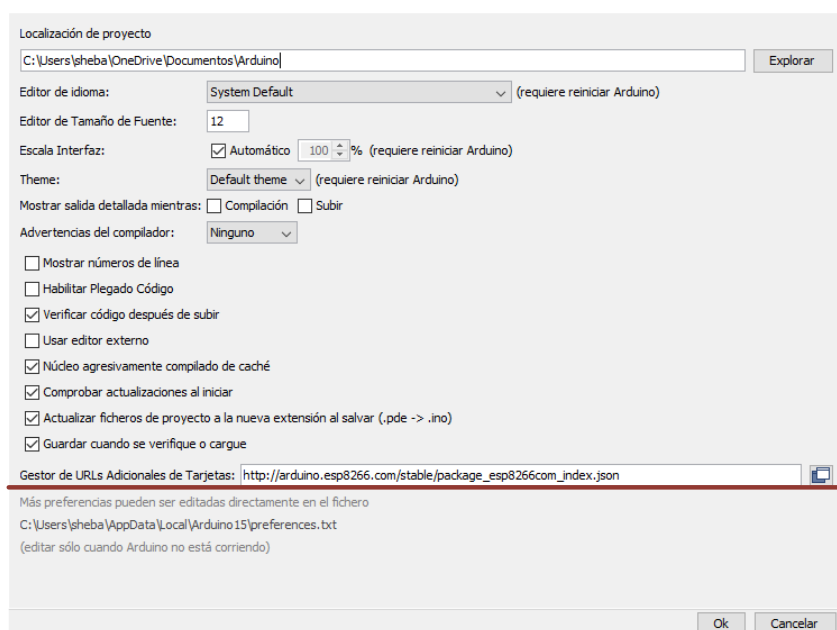
Las cargas de trabajo seleccionadas son: “Desarrollo de ASP.NET y web” las cuales serán usadas para el desarrollo de los servicios y la aplicación web respectivamente y; “Desarrollo de Azure” que será usada para subir las aplicaciones desarrolladas a la nube de Azure desde el IDE. El “Almacenamiento y procesamiento de datos” se usará para que el IDE pueda conectarse a distintas bases de datos; y, por último, “Desarrollo para dispositivos móviles con .NET” el cual será usado para el desarrollo de la aplicación móvil, a través de Xamarin.

Una vez seleccionadas las cargas de trabajo se escoge la opción “Instalar”. Los componentes seleccionados se instalarán y se encontrarán disponibles para el desarrollo del proyecto.

### 2.3.2.3. INSTALACIÓN DEL IDE DE ARDUINO

Para la instalación de IDE de Arduino se debe descargar el instalador directamente desde la página web oficial de Arduino [45]. Una vez descargado el instalador se lo ejecuta y se sigue los pasos del “Asistente de Instalación” hasta concluir con la instalación.

Una vez finalizado de instalar el IDE se deberá agregar opciones de tarjetas Arduino; éste paso se lo realiza debido a que la tarjeta WEMOS D1 usada para el desarrollo del Módulo Electrónico no se encuentra disponible en la configuración básica del IDE. Para agregar otras tarjetas Arduino se deberá ir a “Archivo” y seleccionar la opción de “Preferencias”. Una vez seleccionada se mostrará la pantalla de la **Figura 2.31.**, en la cual se debe realizar el ingreso de la dirección web donde se encuentren las descripciones de otras tarjetas Arduino [46].



**Figura 2.31.** Ventana de preferencias del IDE de Arduino.

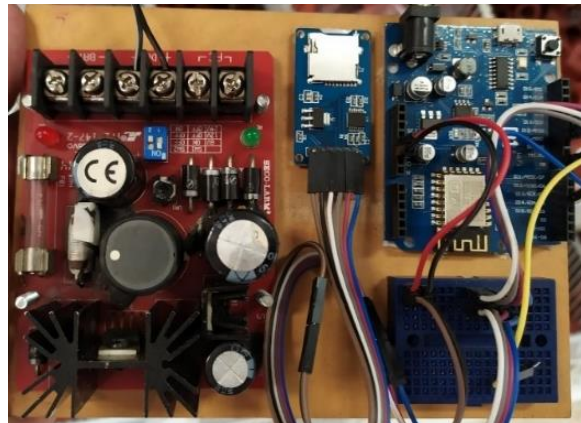


Una vez instaladas las tarjetas se puede hacer uso del IDE Arduino para el desarrollo de las funciones del módulo electrónico.

### 2.3.3. IMPLEMENTACIÓN DEL MEDIDOR ELECTRÓNICO

En este apartado se detalla el acople de los componentes electrónicos y la codificación del funcionamiento lógico que se implementarán sobre el Módulo Electrónico.

El acople de cada uno de los elementos que forman parte del medidor electrónico se muestra en la **Figura 2.32**. En esta figura se muestra a la tarjeta Arduino WEMOS D1, el lector de tarjetas Micro SD y el circuito para alimentación de energía del medidor.



**Figura 2.32.** Implantación de los elementos del medidor electrónico.

A continuación, se realizará la codificación de la parte del funcionamiento lógico del Módulo Electrónico; en el **Código 2.1**. se detalla la configuración de los parámetros iniciales para el funcionamiento del módulo.

```

1  //-----Declaración de Librerías-----
2  #include <SD.h>
3  #include <ESP8266WiFi.h> //Librería para la conexión WiFi
4  #include <WiFiUdp.h>
5  #include <NTPClient.h>
6  #include <SPI.h>
7  #include <String.h>
8  #include <Dns.h>
9  //-----Definición de variables para trabajo con la MicroSD-----
10 File archivoMedidas;
11 //-----Definición de variables para el ingreso de datos.-----
12 char fecha[126];
13 int ingresoDiarioDB = 0;
14 int ingresoDiarioSD = 0;
15 char serieMedidor[] = "med001";
16 int horalingreso =7;
17 //Declaración de variable para conexión WiFi.
18 char ssid[] = "SEBASTIAN BAHAMONDE"; //SSID de la red WiFi
19 char password[] = "1722706452"; //Password de la red WiFi
20 WiFiClient client;
21 //-----Declaración de variables para obtener la hora y fecha.-----
22 WiFiUDP ntpUDP; //Instancia para de la librería WiFiUDP.
23 //Instancia de la NTPClient (bufferUDP, servidor ntp, offset, tiempo de actualización)
24 NTPClient timeClient(ntpUDP, "ec.pool.ntp.org", -18000, 60000);
25 //-----Declaración de variable para el servicio web-----
26 const char* host = "apiprojectotitulacionsebas.azurewebsites.net";
27 //-----Variables para el sensor de flujo YF-S201-----
28 volatile int NumPulsos; //variable para la cantidad de pulsos recibidos
29 int PinSensor = 2; //Sensor conectado en el pin 2
30 float factor_conversion=7.5; //para convertir de frecuencia a caudal
31 float volumen=0;
32 long dt=0; //variación de tiempo por cada bucle
33 long t0=0; //millis() del bucle anterior

```

**Código 2.1.** Configuración de variables del módulo electrónico.



En el **Código 2.1** se puede observar que en las líneas desde la 2 a la 8 se realiza la declaración de las librerías necesarias para el funcionamiento del módulo; en la línea 10 se realiza la declaración de la variable para el manejo del archivo donde se guardarán los valores de consumo en la memoria SD. Las líneas desde la 12 a la 16 realizan la declaración de diferentes tipos de variables como la variable del número de serie del medidor. En las líneas 18, 19 y 20 se realiza la declaración de las variables para la conexión a la red inalámbrica del cliente; las variables de las líneas 20, 22 y 24 son del tipo *WiFiClient*, *WiFiUDP* y *NTPClient* respectivamente, que son usadas para la conexión a la red inalámbrica y para la conexión hacia un servidor NTP<sup>19</sup> para la obtención de la hora y fecha. La línea de código 26 declara el nombre del dominio donde se encontrarán alojados los servicios web y; por último, se tiene las variables de las líneas 28 a la 33, las cuales intervienen en el proceso del cálculo de consumo de agua.

Sobre la placa Arduino se han configurado varias funciones siguiendo el diseño del **Capítulo 2 Apartado 2.2.5**. Cada una de estas cumple un papel específico dentro del funcionamiento del medidor, por ejemplo: el **Código 2.2** muestra la función programada para el ingreso del valor de consumo en la base de datos mediante el uso de un servicio web.

```

1 //-----Funcion para insertar en la Base de Datos-----
2 int insertDB(char serieMedidor[], int valor, String Fecha)
3 {
4     Fecha.toCharArray (fecha, Fecha.length()+1); //Paso de String a Char, se grabara en la variable fecha.
5     Serial.print("Conectando a: ");
6     Serial.println(host);
7     // Use WiFiClient class to create TCP connections
8     WiFiClient client;
9     const int httpPort = 80;
10    if (!client.connect(host, httpPort)) {
11        Serial.println("Conexion fallida.");
12        return 0;
13    }
14    Serial.println("Conexion exitosa");
15    char peticionUrl[] = "/api/Login/ingresarConsumo?date=%s&valor=%d&serieMedidor=%s";
16    char peticion[256];
17    sprintf (peticion, peticionUrl, fecha, valor, serieMedidor);
18    String url = peticion;
19    Serial.print("Peticion URL: ");
20    Serial.println(url);
21    client.print (String ("GET ") + url + " HTTP/1.1\r\n" +
22                  "Host: " + host + "\r\n" +
23                  "Connection: close\r\n\r\n");
24    delay (10000);
25    String resultado;
26    while (client.available()) {
27        resultado = client.readStringUntil ('\r');
28    }
29    Serial.println("Imprimo el resultado.");
30    Serial.println(resultado);
31    int posTrue = resultado.indexOf("true");
32    Serial.println(posTrue);
33    if (posTrue >= 0)
34    {
35        Serial.print("Se ha ingresado en la base de datos.");
36        Serial.println();
37        Serial.println("Cerrando Conexion");
38        return 1;
39    }
40    Serial.print("No se ha ingresado en la base de datos.");
41    Serial.println();
42    Serial.println("Cerrando Conexion");
43    return 0;

```

**Código 2.2.** Función para ingreso del valor de consumo del Módulo Electrónico.

<sup>19</sup> Protocolo de tiempo de red o NTP (*Network Time Protocol*).

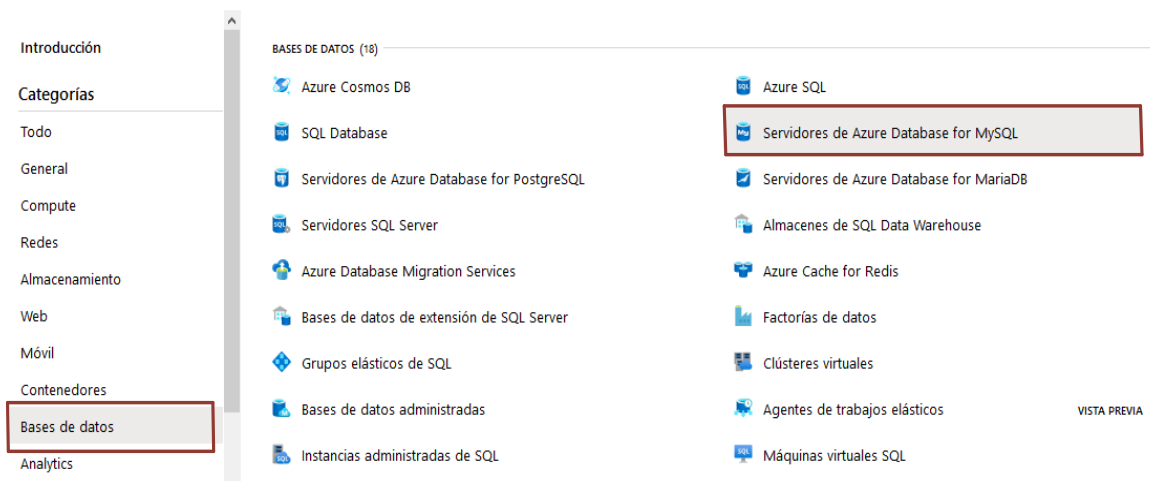
En la definición de la función del **Código 2.2.** se tiene como parámetros de entrada el número de serie del medidor, el valor del consumo y la fecha en la cual se ha tomado este valor, cómo se puede observar en la línea 2 del código. De la línea 15 a la 17 se arma la estructura de la petición del servicio web con los valores de entrada de la función; a continuación, en la línea 21 se realiza la petición al servicio web.

En las líneas 26 y 27 del código se lee la respuesta que es enviada por parte del servidor para posteriormente en la línea 31 realizar la búsqueda de la palabra “true” dentro de la respuesta para comprobar si se ha realizado o no el ingreso a la base de datos. Dependiendo de si se encuentra o no la palabra “true” dentro de la respuesta del servidor se retornará un valor de “1” si se ha encontrado la palabra true, por lo cual se ha ingresado el valor en la base de datos o se retornara el valor de “0” si no se ha encontrado la palabra true, por tal motivo no se ha logrado ingresar el valor. La lógica para el retorno del valor de “0” o “1” se encuentra desde la línea 35 a la 43 del código.

El código completo del Módulo Electrónico se encuentra en el **ANEXO C.**

### 2.3.4. IMPLEMENTACIÓN DE LA CAPA DE DATOS

La implementación de la capa de datos se lo realiza directamente en un servidor MySQL de la nube Azure, para lo cual se accede al panel de Azure y despliega todos los servicios que la nube ofrece. A continuación, se muestra la ventana que se observa en la **Figura 2.33.**



**Figura 2.33.** Servicios de la nube de Azure.

En esta ventana se deberá escoger la opción de “Bases de datos” y dentro de esta opción “Servidores de Azure *Database for MySQL*”. Una vez seleccionada esta opción se desplegará una pantalla donde se observarán los servidores MySQL que se encuentren configurados, para agregar uno nuevo se deberá ir a la opción de “Agregar” la cual desplegará una pantalla en la cual se realizará la configuración del tipo de suscripción de

Microsoft Azure usada para el servidor, el grupo de recursos y el nombre del servidor MySQL. Esta pantalla se la muestra en la **Figura 2.34**.

**Create MySQL server**  
Microsoft

[Básico](#) [Etiquetas](#) [Revisar y crear](#)

Cree un servidor de Azure Database for MySQL. [Más información](#)

**Detalles del proyecto**

Seleccione la suscripción para administrar recursos implementados y los costes. Use los grupos de recursos como carpetas para organizar y administrar todos los recursos.

Suscripción \* ⓘ (Deshabilitado) Azure for Students

Grupo de recursos \* ⓘ Seleccionar existentes...  
[Crear nuevo](#)

**Detalles del servidor**

Especifique la configuración necesaria para este servidor, incluida la selección de una ubicación y la configuración de los recursos de proceso y almacenamiento.

Nombre del servidor \* ⓘ Introduzca un nombre de servidor

Origen de datos \* ⓘ Ninguno Copia de seguridad

**Figura 2.34.** Configuración del servidor MySQL.

Una vez hecha la configuración del servidor, se da *click* en la opción de “Revisar y crear” para finalizar la configuración del servidor MySQL. Al finalizar la creación del servidor se despliega la pantalla que muestra la **Figura 2.35**, donde se observa las credenciales para el acceso al servidor.

Grupo de recursos...	: Tesis	Nombre del servidor	: aguabasedatos.mysql.database.azure.com
Estado	: Disabled	Nombre de inicio de se...	: adminTesis@aguabasedatos
Ubicación	: Centro-Sur de EE. UU.	Versión de MySQL	: 5.7
Suscripción (cambiar)	: Azure for Students	Configuración de rendi...	: Básico, 2 núcleos virtuales, 50 GB
Id. de suscripción	: ebc027b-3cf9-4c22-8261-6d33ed84e3b4	SSL enforce status	: DISABLED
Etiquetas (cambiar)	: Haga clic aquí para agregar etiquetas.		

**Figura 2.35.** Información general del servidor MySQL.

Por otro lado, también se tiene la opción de ver diferentes formatos de la cadena de conexión hacia el servidor MySQL como se muestra en la Figura 2.36. La cadena de conexión es usada para el desarrollo de los servicios web, ya que esta cadena será la encargada de realizar la conexión a la base de datos.

```

ADO.NET
Server=aguabasedatos.mysql.database.azure.com; Port=3306; Database={your_database}; Uid=adminTesis@aguabasedatos; Pwd={your_password}; SslMode=Preferred;

JDBC
String url = "jdbc:mysql://aguabasedatos.mysql.database.azure.com:3306/{your_database}?useSSL=true&requireSSL=false"; myDbConn = DriverManager.getConnection(url, "ad ...

Node.js
var conn = mysql.createConnection({host: "aguabasedatos.mysql.database.azure.com", user: "adminTesis@aguabasedatos", password: {your_password}, database: {your_datab ...

PHP
$con=mysql_init(); mysqli_ssl_set($con, NULL, NULL, {ca-cert filename}, NULL, NULL); mysqli_real_connect($con, "aguabasedatos.mysql.database.azure.com", "adminTesis@agua ...

Python
cnx = mysql.connector.connect(user="adminTesis@aguabasedatos", password={your_password}, host="aguabasedatos.mysql.database.azure.com", port=3306, database={you...

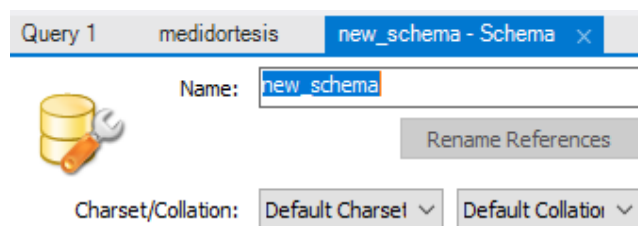
Ruby
client = Mysql2::Client.new(username: "adminTesis@aguabasedatos", password: {your_password}, database: {your_database}, host: "aguabasedatos.mysql.database.azure.com", ...

Aplicación web
Database={your_database}; Data Source=aguabasedatos.mysql.database.azure.com; User Id=adminTesis@aguabasedatos; Password={your_password}

```

**Figura 2.36.** Cadenas de conexión hacia el servidor MySQL.

A continuación, se debe crear la base de datos, para lo cual se hace uso de la herramienta MySQL Workbench. En esta herramienta se selecciona la opción “Add Schema” la cual despliega la pantalla de la **Figura 2.37** donde se debe ingresar el nombre de la base de datos y seleccionar “Apply” para guardar los cambios.



**Figura 2.37.** Pantalla para la creación de una base de datos.

Una vez creada la base de datos se realiza la codificación de cada una de las tablas siguiendo el diseño descrito en el **Capítulo 2 Apartado 2.2.6.** mediante el lenguaje SQL. Por citar un ejemplo, el **Código 2.3.** muestra las instrucciones necesarias para la creación y configuración de cada una de los campos de la tabla “medidor”.

```

1  --
2  -- Estructura de tabla para la tabla `medidor`
3  --
4  CREATE TABLE `medidor` (
5    `idMedidor` int(11) NOT NULL,
6    `direccion` varchar(45) NOT NULL,
7    `sector` varchar(45) NOT NULL,
8    `serieMedidor` varchar(45) NOT NULL,
9    `idCliente` int(11) NOT NULL
10 )
11 --
12 -- Indices de la tabla `medidor`
13 --
14 ALTER TABLE `medidor`
15   ADD PRIMARY KEY (`idMedidor`),
16   ADD KEY `idCliente` (`idCliente`);
17 --
18 -- AUTO_INCREMENT de la tabla `medidor`
19 --
20 ALTER TABLE `medidor`
21   MODIFY `idMedidor` int(11) NOT NULL AUTO_INCREMENT;
22 --
23 -- Clave foranea para la tabla `medidor`
24 --
25 ALTER TABLE `medidor`
26   ADD CONSTRAINT `medidor_ibfk_1` FOREIGN KEY (`idCliente`) REFERENCES `cliente` (`idCliente`);

```

**Código 2.3.** Creación de la tabla medidor.

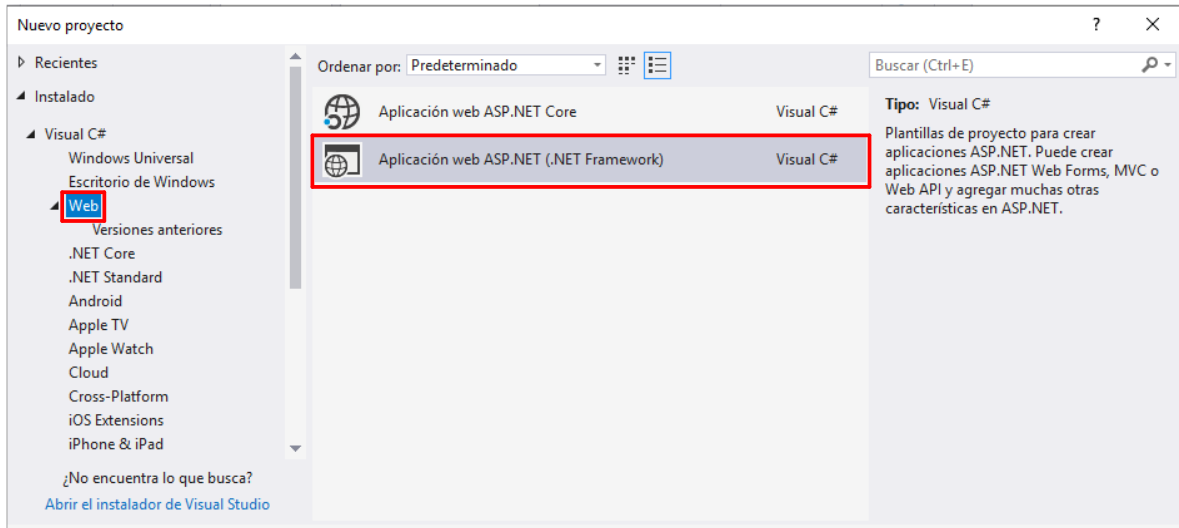
La línea 4 del código muestra la instrucción para la creación de la Tabla con el nombre de “medidor”; desde la línea 5 a la 9 se declaran las columnas que tendrá la tabla; a cada columna se lo configura con su tipo de valor como: un número entero (int) o como una cadena de caracteres (varchar); así también, cada columna de la tabla se la configura para que este no se guarde un campo vacío o nulo. La línea 14 del código muestra la sentencia necesaria para modificar la tabla. Las líneas 15 y 16 son las sentencias para agregar la clave primaria de la tabla. La línea 21 del código realiza la configuración para que el valor del campo “idMedidor” se autoincremente. Por último, la línea 26 del código indica la sentencia necesaria para establecer la clave foránea de la tabla.

La codificación completa de la base de datos se muestra en el **ANEXO D**.

### 2.3.5. IMPLEMENTACIÓN DE LA CAPA DE NEGOCIO

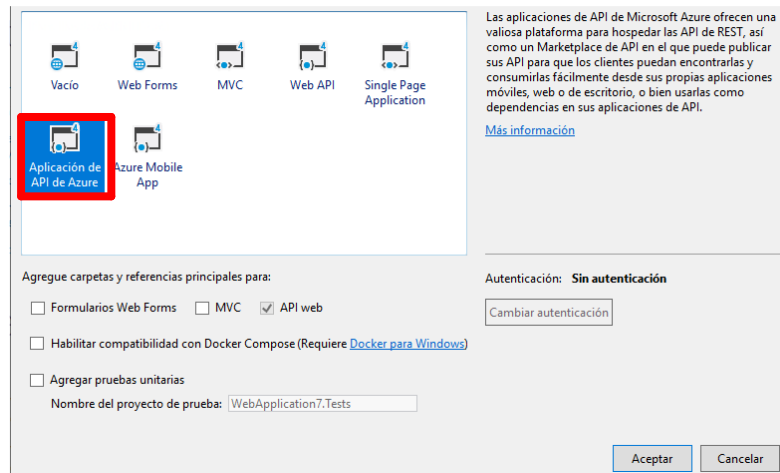
En el siguiente apartado se detallará la implementación de los diferentes servicios web que son implementados para el cumplimiento de los Requerimientos Funcionales del sistema, para lo cual, se seguirá el diseño detallado en el **Capítulo II Apartado 2.2.7** y se hará uso del IDE de Visual Studio junto con el lenguaje de programación C#.

La implementación de los servicios web se lo realiza en un proyecto del tipo Aplicación Web ASP.NET (.NET Framework) el cual se observa en la **Figura 2.38**.



**Figura 2.38.** Creación de un proyecto ASP.NET.

Una vez seleccionado el tipo de proyecto se muestra la pantalla de la **Figura 2.39**, donde se escoge la opción de Aplicación de API de Azure y se crea el proyecto.



**Figura 2.39.** Tipo de Aplicación web ASP.NET.

Para comenzar con la implementación de los servicios web se codifica las clases necesarias para el funcionamiento del sistema. El **Código 2.4** muestra la implementación de la clase “medidor” con los atributos y métodos necesarios de la clase.

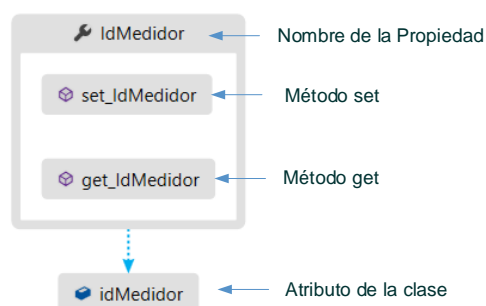
```

6 namespace APIProyectoTitulacion.Models
7 {
8     public class Medidor
9     {
10         private int idMedidor, idCliente;
11         private String direccion;
12         private String sector;
13         private String serieMedidor;
14         public int IdMedidor { get => idMedidor; set => idMedidor = value; }
15         public string Direccion { get => direccion; set => direccion = value; }
16         public string Sector { get => sector; set => sector = value; }
17         public string SerieMedidor { get => serieMedidor; set => serieMedidor = value; }
18         public int IdCliente { get => idCliente; set => idCliente = value; }
19     }
20 }

```

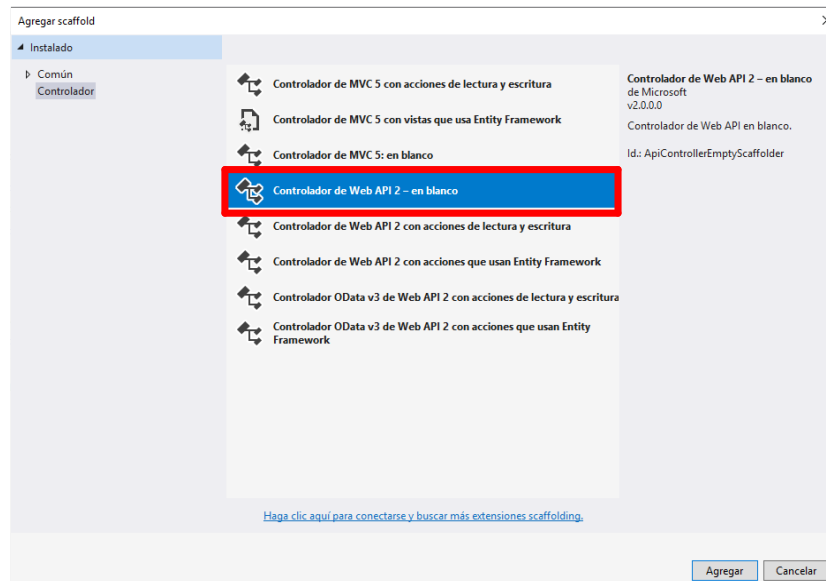
**Código 2.4.** Codificación de la clase medidor.

En esta porción de código se muestra desde la línea 10 a la 13 la declaración de los atributos del medidor; en las líneas de la 14 a la 18 se muestran las propiedades de cada uno de los atributos del medidor. Cada una de estas propiedades genera los métodos “Get” y “Set” de cada atributo, un ejemplo de esto se muestra en la **Figura 2.40**.



**Figura 2.40.** Métodos get y set.

Una vez codificadas las clases se debe agregar los controladores los cuales contienen los servicios web. La **Figura 2.41** muestra la creación de un controlador en blanco.



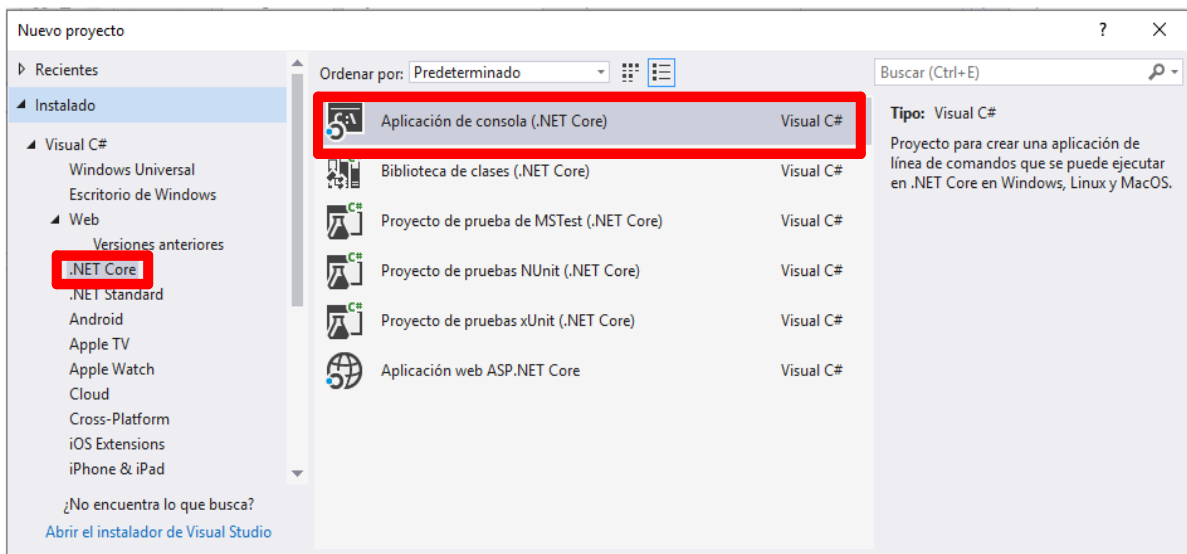
**Figura 2.41.** Agregar nuevo controlador.

Cada controlador creado tendrá los servicios web relacionados a las acciones que se pueden realizar con cada clase, por ejemplo, el controlador “Medidor” contiene el servicio web “agregarMedidor”. Este servicio es usado para ingresar la información de un nuevo medidor a la base de datos teniendo como parámetro de entrada al medidor que se va a ingresar y retornando un valor de “verdadero” si se ha ingresado la información del medidor o “falso” si no se ha ingresado. El código empleado para la creación de este servicio se muestra en el **Código 2.5**.

```
113 [HttpPost]
114 //Metodo para crear un medidor.
115 0 referencias
116 public IHttpActionResult agregarMedidor(Medidor medidor)
117 {
118     String query = "INSERT INTO medidor (idCliente,direccion,sector,serieMedidor) " +
119                 "VALUES (@idCliente,@direccion,@sector,@serieMedidor)";
120     ConexionBD coneccion = new ConexionBD();
121     if (coneccion.abrirConexion())
122     {
123         MySqlCommand command = new MySqlCommand(query, coneccion.coneccion);
124         command.Parameters.AddWithValue("@idCliente", medidor.IdCliente);
125         command.Parameters.AddWithValue("@direccion", medidor.Direccion);
126         command.Parameters.AddWithValue("@serieMedidor", medidor.SerieMedidor);
127         command.Parameters.AddWithValue("@sector", medidor.Sector);
128         int retorno = command.ExecuteNonQuery();
129         if (retorno != 0)
130         {
131             coneccion.cerrarConexion();
132             return Ok(true);
133         }
134     }
135     coneccion.cerrarConexion();
136     return Ok(false);
137 }
```

**Código 2.5.** Servicio web para ingreso de un medidor a la base de datos.

La línea 113 del código establece el tipo de servicio web que se desarrollará, el cual en este caso es del tipo “POST”. La línea 117 del código arma la sentencia SQL para el ingreso de un medidor en la base de datos; a continuación, la línea 119 se establece una instancia de un objeto del tipo “ConeccionBD”, el cual es usado para abrir o cerrar la conexión con la base de datos. Si se establece la conexión a la base de datos se ejecuta la línea 122 del código la cual instancia un objeto del tipo “ MySqlCommand” el cual representa una sentencia SQL, que será ejecutada en un servidor MySQL. A este tipo de objeto se lo instancia con la sentencia SQL de la línea 117 y la conexión a la base de datos. Desde la línea 123 a la 126 se establece el valor de los distintos parámetros de la sentencia SQL que serán almacenados. En la línea 127 se realiza la ejecución la sentencia SQL; por último, desde la línea 128 a la 135 se establece la lógica para el retorno del valor de verdadero si se ha realizado el ingreso, o el valor de falso si no se ha realizado el ingreso. Por otro lado, para el envío de notificaciones a los clientes se crea un nuevo tipo de proyecto que es de tipo consola en Visual Studio. Para crear este proyecto se deberá elegir el tipo de proyecto mostrado en la **Figura 2.42**.



**Figura 2.42.** Creación de aplicación tipo consola web.

Una vez creado el proyecto se realiza la codificación del mismo, por lo cual, el **Código 2.6** muestra una parte del código implementado para el envío de notificaciones mensuales al cliente.



```

10 static void Main(string[] args)
11 {
12     bool notificacionMensual = false;
13
14     while (true)
15     {
16         //Obtenemos el primer dia del mes actual para usarlo para calcular el ultimo dia del mes.
17         DateTime primerDiaDelMes = new DateTime(DateTime.Now.Year,DateTime.Now.Month,1);
18         //DateTime.Now.Date
19         if (DateTime.Now.Date == primerDiaDelMes.AddMonths(1).AddDays(-1).Date)
20         {
21             if (notificacionMensual == false)
22             {
23                 Console.WriteLine("Notifico los consumos mensuales.");
24                 //La solicitud al servicio me regresa true= si notifico o false= no notifico.
25                 //Autentico al administrador principal.
26                 //Deserealizo el token de autenticacion.
27                 string token = JsonConvert.DeserializeObject<String>(autenticarAdministrador());
28                 if (token != String.Empty || token != null)
29                 {
30                     bool respuesta = enviarCorreos(token);
31                     if (respuesta == true)
32                     {
33                         notificacionMensual = true;
34                     }
35                 }
36             }
37             else
38             {
39                 Console.WriteLine("Ya notifique los consumos mensuales.");
40             }
41         }
42         if (DateTime.Now.Date != primerDiaDelMes.AddMonths(1).AddDays(-1).Date)
43         {
44             Console.WriteLine(DateTime.Now.Date);
45             Console.WriteLine(primerDiaDelMes.AddMonths(1).AddDays(-1).Date);
46             Console.WriteLine("No notifico los consumos mensuales.");
47             notificacionMensual = false;
48         }
49     }
50 }

```

**Código 2.6.** Parte del código implementado para el envío de notificaciones.

El funcionamiento de este programa se basa en un bucle *while* el cual se lo declara en la línea de código 14; después se realiza la comparación de la fecha actual y el último día del mes en la línea 19, debido a que el envío de notificaciones se lo realizará únicamente al finalizar el mes. Si la condición se cumple se realizará el envío de la notificación al usuario; esta notificación es un correo electrónico en el cual consta el nombre del cliente, el número de serie del medidor, la última lectura del medidor, y los valores de consumo de los tres últimos meses.

El servicio web implementado para realizar el envío de las notificaciones se lo muestra en el **Código 2.7**. En las líneas de la 51 a 54 se realiza el armado de la cabecera del correo electrónico en donde se ha agregado el correo del destinatario y remitente. Desde la línea 54 a la 67 se arma el cuerpo del correo; en el cuerpo del correo se encuentra el número de serie del medidor, el consumo total del medidor de agua, y los consumos de los últimos tres meses. Por último, desde la línea 69 a la 81 se realiza las configuraciones necesarias para hacer uso de la cuenta de correo electrónico como las credenciales de la cuenta, el nombre del servidor de correo electrónico y en la línea 81 se realiza el envío del mensaje.

```

45     private static void EnviarCorreoElectronico(Cliente auxCliente, Medidor auxMedidor, List<int> medidasMeses, Medida auxUltimaMedida)
46     {
47     {
48     {
49         if (auxCliente.Mail != null)
50         {
51             var Mensaje = new MailMessage();
52             Mensaje.To.Add(new MailAddress(auxCliente.Mail));
53             Mensaje.From = new MailAddress(WebConfigurationManager.AppSettings["AdminUser"]);
54             Mensaje.Subject = obtenerSubjectDelCorreo(DateTime.Now);
55             String body = String.Format("<p><b> NOTIFICACIÓN DE CONSUMO MENSUAL DE AGUA POTABLE.</b></p><p><b> Usuario:</b> {0} {1}</p>" +
56                 "<p><b> Medidor:</b> {2}</p>" +
57                 "<p><b> Lectura actual del medidor: </b> {3} m3</p>" +
58                 "<p><b> Los valores de consumo de los últimos tres meses son:" +
59                 "<p><b> {4}</b> {5} m3</p>" +
60                 "<p><b> {6}</b> {7} m3</p>" +
61                 "<p><b> {8}</b> {9} m3</p>", auxCliente.Nombres, auxCliente.Apellidos, auxMedidor.SerieMedidor,
62                 Math.Round(Convert.ToDouble(auxUltimaMedida.Cantidad) / 1000, 1),
63                 obtenerNombreDelMesDeConsumo(DateTime.Now.AddMonths(-2)), Math.Round(Convert.ToDouble(medidasMeses[0]) / 1000, 1).ToString(),
64                 obtenerNombreDelMesDeConsumo(DateTime.Now.AddMonths(-1)), Math.Round(Convert.ToDouble(medidasMeses[1]) / 1000, 1).ToString(),
65                 obtenerNombreDelMesDeConsumo(DateTime.Now), Math.Round(Convert.ToDouble(medidasMeses[2]) / 1000, 1).ToString());
66             Mensaje.Body = body;
67             Mensaje.IsBodyHtml = true;
68         }
69         using (var smtp = new SmtplibClient())
70         {
71             var credencial = new NetworkCredential
72             {
73                 Username = WebConfigurationManager.AppSettings["AdminUser"],
74                 Password = WebConfigurationManager.AppSettings["AdminPassword"],
75             };
76             smtp.Credentials = credencial;
77             smtp.Host = WebConfigurationManager.AppSettings["SMTPName"];
78             smtp.Port = int.Parse(WebConfigurationManager.AppSettings["SMTPPort"]);
79             smtp.EnableSsl = true;
80         }
81         smtp.Send(Mensaje);
82     }
83     }
84 }

```

**Código 2.7.** Servicio Web. Envío de notificaciones.

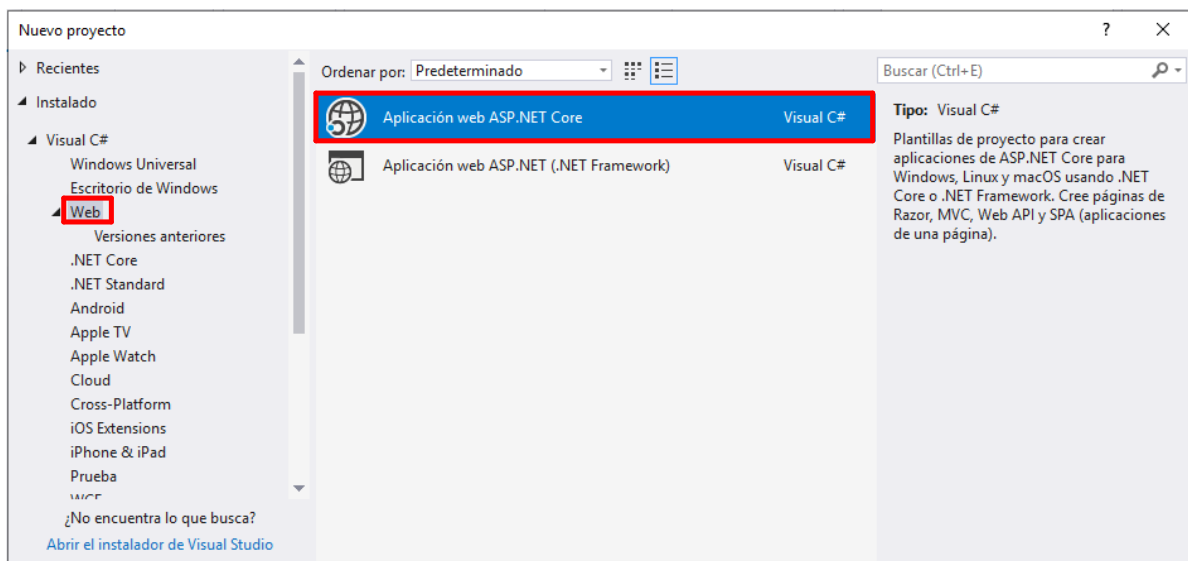
La implementación de todos los servicios web y el programa para el envío de notificaciones se encuentran en el **ANEXO E**.

### 2.3.6. IMPLEMENTACIÓN DE LA CAPA DE PRESENTACIÓN

En el presente apartado se detallará la implementación de las interfaces para las aplicaciones web y móvil siguiendo el diseño del **Capítulo II Apartado 2.2.8**.

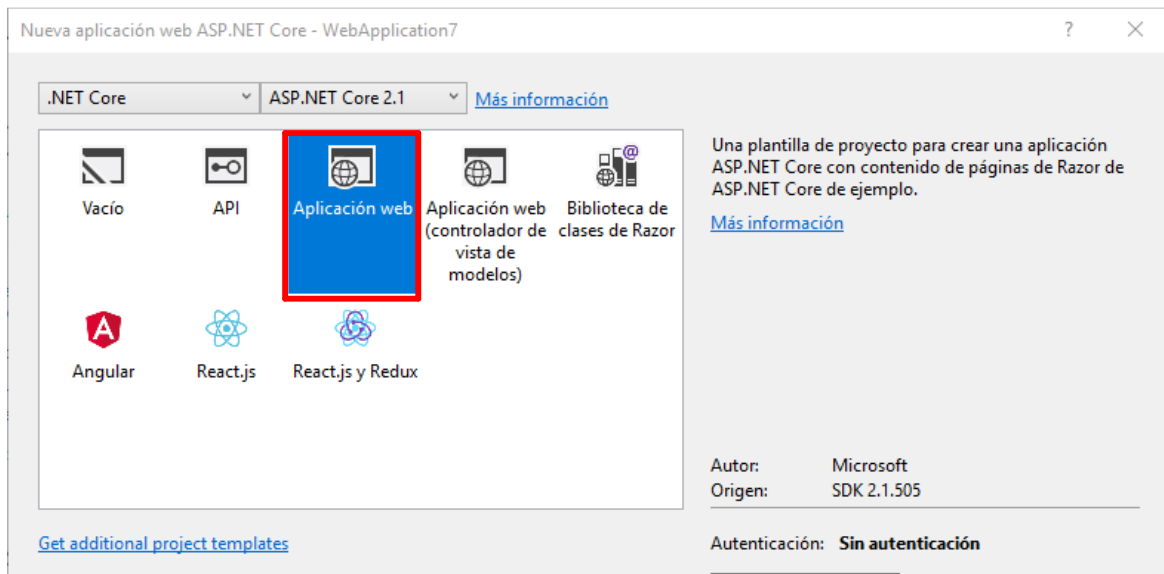
#### 2.3.6.1. IMPLEMENTACIÓN DE LA APLICACIÓN WEB

Para la creación de la interfaz de la aplicación web se crea un proyecto del tipo Aplicación web ASP.NET Core en Visual Studio. La **Figura 2.43** muestra la creación de este tipo de proyecto para la aplicación web.



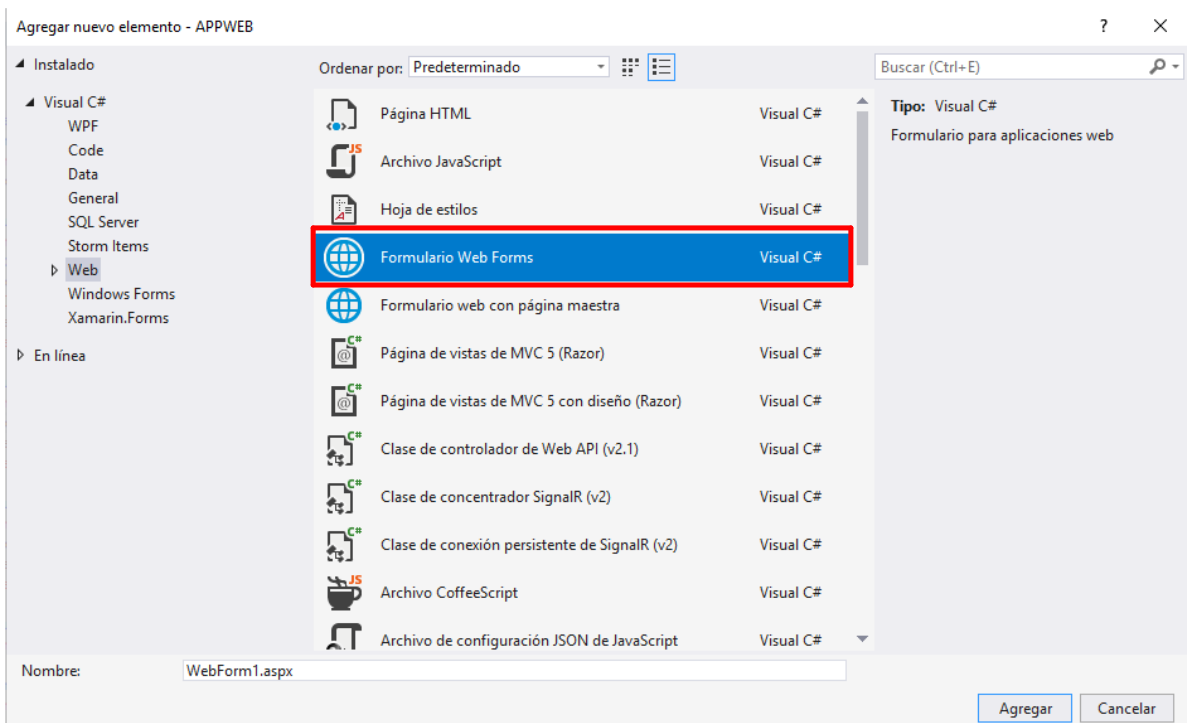
**Figura 2.43.** Creación del proyecto del tipo Aplicación web ASP.NET.

Una vez elegido el tipo de proyecto se mostrará la ventana de la **Figura 2.44**, donde se elige la opción de “Aplicación web” y se creará el proyecto.



**Figura 2.44.** Creación de la aplicación web.

Lo siguiente es agregar nuevos elementos del tipo Formulario “*Web Forms*” los cuales representan a cada pantalla que observará el usuario; la agregación de estos elementos en el proyecto se muestra en la **Figura 2.45**.



**Figura 2.45.** Implementación del elemento Formulario *Web Forms*.

El agregar el elemento Formulario *Web Forms* genera archivos con la extensión .aspx y .aspx.cs. El archivo con la extensión .aspx es donde se realiza la codificación de los

controles de la interfaz del usuario. Un ejemplo de codificación de este tipo de elementos se muestra en el **Código 2.8.** donde se puede observar la implementación de los controles *Label*, *Textbox* y de un validador de campo. El control *Textbox* posee la propiedad "ID" que es un identificador del control el cual permite que el control pueda ser llamado desde el archivo con la extensión .aspx.cs; por otro lado, el validador de campo se encuentra ligado al *Textbox* para verificar que el valor del control no se encuentre vacío y si esto ocurre se muestre un mensaje al usuario informándole de que es necesario que ingrese un valor.

```
<div class="input-group">
  <asp:Label runat="server" CssClass="form-control border-white border-left-0 bg-white small">Serie:
  </asp:Label>
  <asp:TextBox runat="server" CssClass="form-control border-left-0 bg-light w-50" ID="txtNSerie"
    placeholder="N° de Serie."></asp:TextBox>
  <asp:RequiredFieldValidator runat="server" ID="RequiredFieldValidator10" ControlToValidate="txtNSerie"
    ErrorMessage="*" ForeColor="Red" ValidationGroup="form_ejm"></asp:RequiredFieldValidator>
</div>
```

**Código 2.8.** Codificación de los controles de la aplicación web.

Por otro lado, el archivo con la extensión .aspx.cs es donde se realiza la codificación de la lógica del funcionamiento de cada uno de los controles que se encuentran en la interfaz de la aplicación. Por ejemplo, la codificación del botón que permite guardar un Medidor se muestra en el **Código 2.9.**

```
286 protected void btnGuardarMedidor_Click(object sender, EventArgs e)
287 {
288     if (txtNSerie.Text != String.Empty && txtSector.Text != String.Empty && txtDireccion.Text != String.Empty)
289     {
290         Medidor aux = new Medidor();
291         aux.Direccion = txtDireccion.Text;
292         aux.SerieMedidor = txtNSerie.Text;
293         aux.Sector = txtSector.Text;
294         aux.IdCliente = Convert.ToInt32(txtIdCliente.Text);
295         bool verificacionMedidor = verificarMedidor(txtNSerie.Text);
296         if (verificacionMedidor == false)
297         {
298             bool guardado = agregarMedidor(aux);
299             if (guardado == true)
300             {
301                 lblMensajeMedidor.Text = "Ingreso con éxito.";
302                 lblMensajeMedidor.ForeColor = System.Drawing.Color.Green;
303                 mostrarMedidores(buscarMedidoresCliente(Convert.ToInt32(txtIdCliente.Text)));
304                 btnGuardarMedidor.Visible = false;
305                 txtIdMedidor.Visible = true;
306                 lblIdMedidor.Visible = true;
307                 deshabilitarCamposMedidor();
308             }
309         }
310     }
311 }
```

**Código 2.9.** Código para agregar un nuevo medidor.

Este código sirve para el registro de un nuevo medidor en la base de datos. La línea 288 del código verifica que los campos de información del medidor a ingresarse no se encuentren vacíos. En las líneas 290 a la 294 se realiza la instancia de un objeto del tipo "Medidor" y se establece el valor de sus atributos. La línea 295 verifica si el medidor que se está ingresando no haya sido ingresado anteriormente. Si el medidor no se encuentra registrado se procede a su almacenamiento en la base de datos para lo cual se ejecuta la sentencia de la línea 298, esta línea ejecutará el método para solicitar el servicio web necesario para el registro del medidor, este método retornará un valor "true" si el medidor ha sido ingresado o "false" sino ha sido ingresado a la base de datos; si el resultado es "true" se ejecutan las líneas 301 a 307 del código, donde se muestra un mensaje al usuario de que el medidor ha sido ingresado.

Las acciones de obtener, guardar, editar o eliminar que se realicen en la aplicación web se encuentran ligadas a un servicio web; por lo cual estos deben ser invocados. El

Código 2.10. muestra cómo se realiza la invocación de un servicio web que permite el registro de un nuevo medidor en la base de datos.

```
154     protected bool agregarMedidor(Medidor medidor)
155     {
156         var client = new RestClient(ConfigurationManager.AppSettings["URLServidor"]);
157         var request = new RestRequest("api/Medidor/agregarMedidor", Method.POST);
158         request.AddParameter("serieMedidor", medidor.SerieMedidor);
159         request.AddParameter("direccion", medidor.Direccion);
160         request.AddParameter("sector", medidor.Sector);
161         request.AddParameter("idCliente", medidor.IdCliente);
162         request.AddHeader("Authorization", ad.Token);
163         IRestResponse response = client.Execute(request);
164         bool respuesta = new JavaScriptSerializer().Deserialize<bool>(response.Content);
165         return respuesta;
166     }
```

**Código 2.10.** Uso del servicio web en la aplicación web.

En el **Código 2.10** la línea 156 se realiza la instancia de un objeto del tipo `RestClient` el cual alberga la dirección URL del servidor donde se encuentran los servicios web; además, desde la línea 157 a la 162 se realiza la instancia de un objeto del tipo `RestRequest` en el cual se establecen los valores de la URI del servicio web, el tipo de método HTTP y los parámetros que se desean enviar en la petición. A continuación, en la línea 163 se realiza la solicitud al servicio web y se guarda la respuesta del servidor en un objeto del tipo `IRestResponse`, el cual en la siguiente línea de código será deserializado a un tipo de dato `bool`, para que este valor sea retornado por el método.

Una vez realizada la codificación de los controles (Botones, Etiquetas, Campos de Texto, etc.) necesarios para cada una de las interfaces de la aplicación web y el funcionamiento de los mismos, se han obtenido las interfaces mostradas a continuación.



Figura 2.46. Interfaz de *login* de la aplicación web.

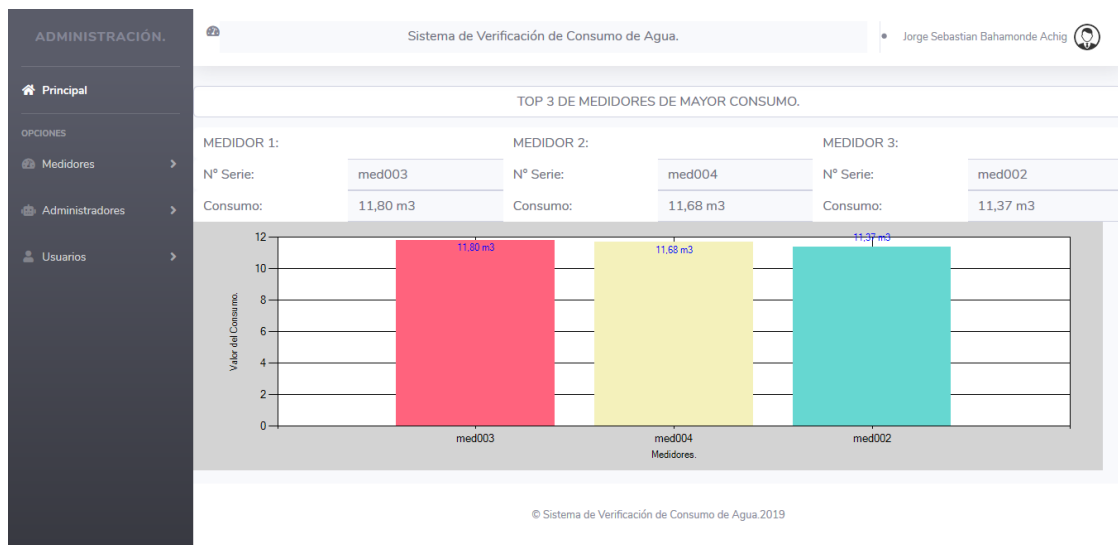


Figura 2.47. Interfaz principal de la aplicación web.



Figura 2.48. Interfaz para realizar la búsqueda de consumos de la aplicación web.

ADMINISTRACIÓN. Sistema de Verificación de Consumo de Agua. Jorge Sebastian Bahamonde Achig

Principal

OPCIONES

- Medidores
- Administradores
- Usuarios

INGRESO DE ADMINISTRADORES DEL SISTEMA

Nombre: Nombre.

Apellido: Apellido.

Username: Username.

Password: Password.

Conf. Password: Password.

© Sistema de Verificación de Consumo de Agua 2019

Figura 2.49. Interfaz para agregar un nuevo administrador.

ADMINISTRACIÓN. Sistema de Verificación de Consumo de Agua. Jorge Sebastian Bahamonde Achig

Principal

OPCIONES

- Medidores
- Administradores
- Usuarios

Buscar: Ingrese el userName del administrador.

RESULTADO DE LA BÚSQUEDA DE ADMINISTRADORES.

Id Administrador: Ingrese el UserName del Administrador.

UserName: UserName del Administrador.

Nombres: Nombres del Administrador.

Apellidos: Apellidos del Administrador.

Password: Ingrese Password Anterior.

Nuevo Password: Ingrese el nuevo password.

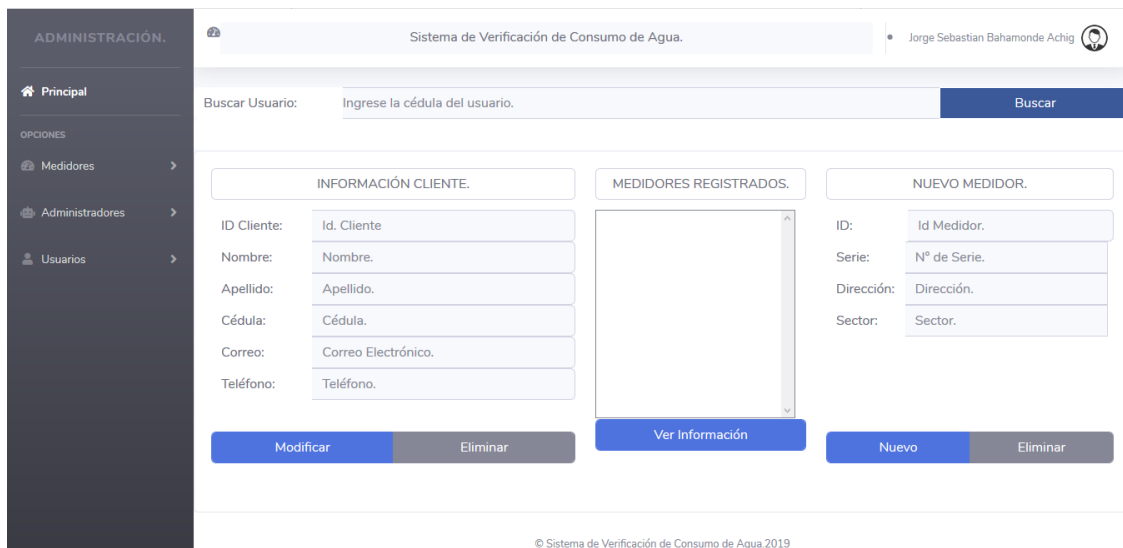
Confirmar Password: Ingrese de nuevo el password.

© Sistema de Verificación de Consumo de Agua 2019

Figura 2.50. Interfaz para la consulta, modificación y eliminación de administradores.



**Figura 2.51.** Interfaz para el ingreso de nuevos clientes.



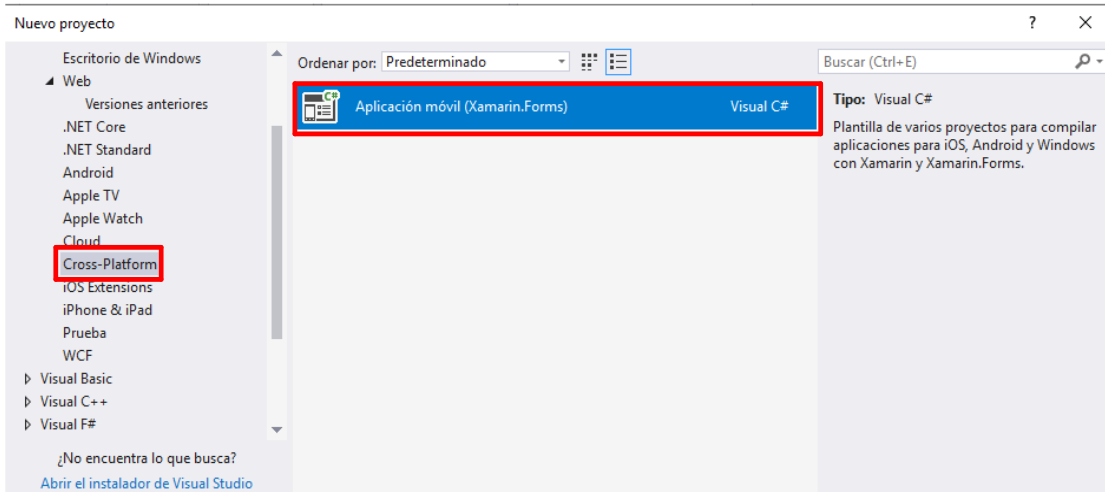
**Figura 2.52.** Interfaz para la búsqueda, modificación o eliminación de clientes.

El código completo para la implementación de las interfaces de la aplicación web se encuentra en el **ANEXO F**.

### 2.3.6.2. IMPLEMENTACIÓN DE LA APLICACIÓN MÓVIL

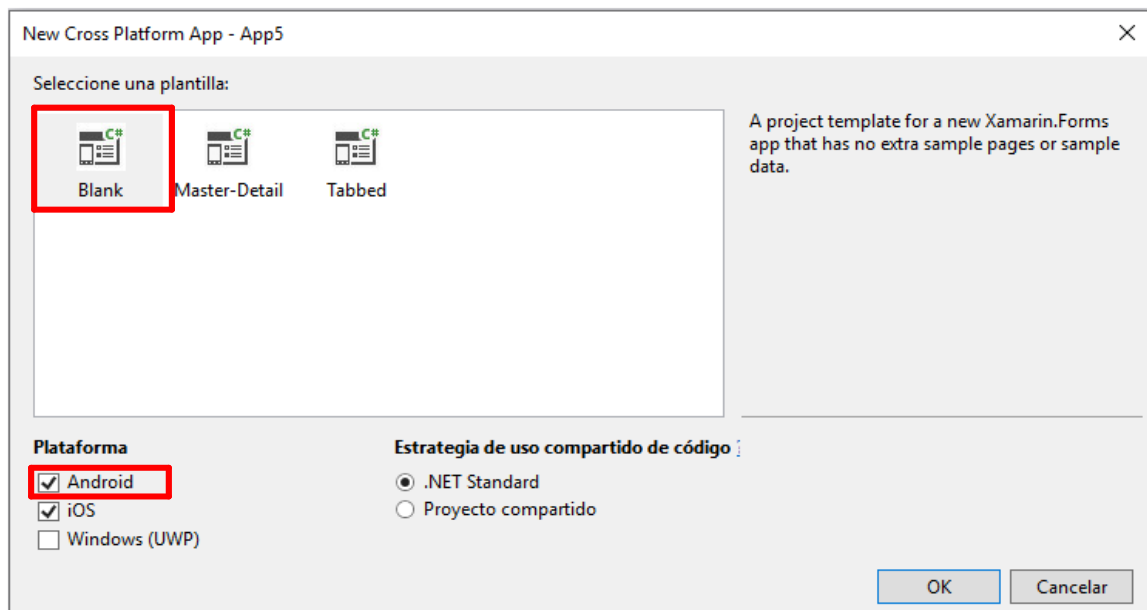
Para la creación de la aplicación móvil y sus interfaces se crea en Visual Studio un proyecto del tipo Aplicación móvil (Xamarin.Forms), el cual se encuentra en la opción de “Cross-Plataform”. La **Figura 2.53** muestra la elección para la creación de este tipo de proyecto.





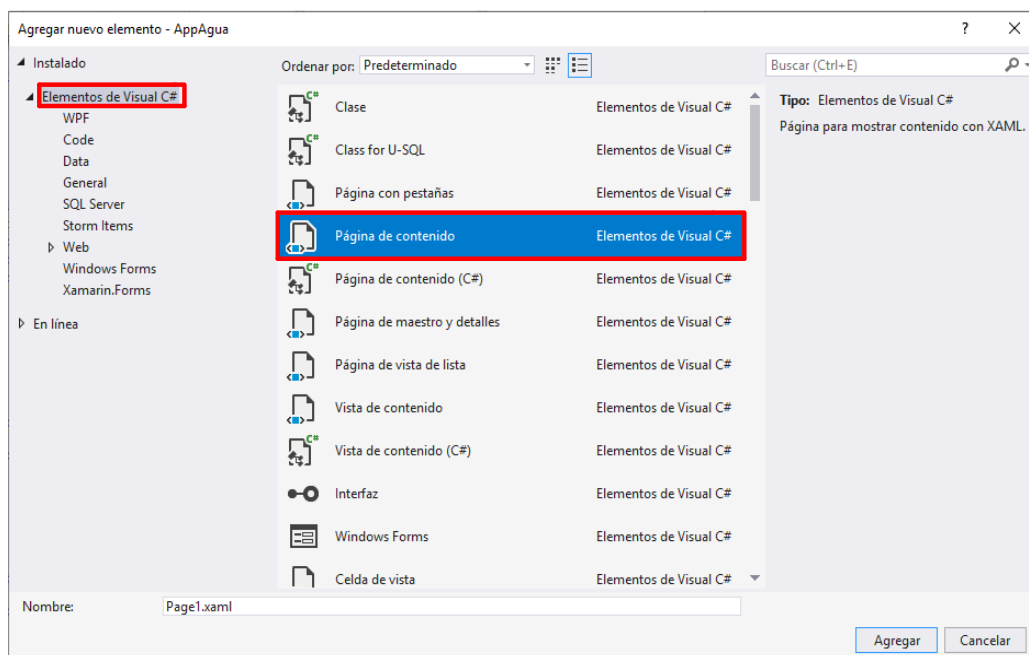
**Figura 2.53.** Creación del proyecto del tipo Aplicación móvil (*Xamarin.Forms*).

Una vez seleccionado el tipo de proyecto se muestra la pantalla de la **Figura 2.54**, en la cual se debe escoger la plataforma para la que se desarrolla la aplicación móvil; en este caso Android y el tipo de plantilla la cual es “En blanco” o “Blank”. Presionando “OK” se crea el proyecto.



**Figura 2.54.** Ventana de elección de la plataforma móvil.

Una vez creado el proyecto se debe agregar el elemento del tipo “Página de contenido”, el cual representa una interfaz o *layout* de la aplicación móvil. La **Figura 2.55**. muestra la pantalla en la cual se hace la elección de este tipo de elemento.



**Figura 2.55.** Implementación del elemento del tipo Página de contenido.

Al agregar este elemento se genera dos tipos de archivos los cuales tienen las extensiones .xaml y .xaml.cs. El archivo con la extensión .xaml es donde se realiza la codificación de los controles que posee la interfaz del usuario. Un ejemplo de codificación de los controles se muestra en el **Código 2.11**.

```

48      <Entry x:Name="txtIdMedidor"
49            Placeholder="Serie Del Medidor."
50            PlaceholderColor="White"
51            FontSize="13"
52            HorizontalTextAlignment="Center"
53            TextColor="White"
54            FontFamily="Consolas"
55            HorizontalOptions="FillAndExpand"/>
56      </StackLayout>
57      <BoxView WidthRequest="10"/>
58      <BoxView HeightRequest="150"/>
59      <Button Text="Ingresar"
60            BackgroundColor="#121e35"
61            WidthRequest="100"
62            HorizontalOptions="Center"
63            VerticalOptions="Center"
64            CornerRadius="100"
65            FontFamily="Consolas"
66            FontAttributes="Bold"
67            TextColor="White"
68            x:Name="btnIngresar"
69            Clicked="btnIngresar_Click"/>
70      <ActivityIndicator Color="Red"
71            HorizontalOptions="Center"
72            VerticalOptions="Center"
73            x:Name="cargando"
74            />

```

**Código 2.11.** Codificación de los controles de la aplicación móvil.

Cada uno de los controles que se agregan a la interfaz de la aplicación poseen características las cuales pueden ser modificados para dar estilo al control; por ejemplo,

en la línea 54 del código se tiene la etiqueta *FontFamily* la cual es usada para dar el tipo de letra; en la línea 67 se configura al atributo *TextColor* que da el color al texto del control; otro atributo es el que se configura en la línea 68 el cual es *x:Name* el cual es utilizado para identificar al control y que este pueda ser usado desde el archivo con extensión *.xaml.cs*. El archivo con extensión *.xaml.cs* es donde se codifica la lógica de funcionamiento de cada uno de los controles de la interfaz, por ejemplo, el **Código 2.12** muestra el código implementado para el ingreso a la aplicación.

```
21 public async void btnIngresar_Click(Object sender, EventArgs e)
22 {
23     cargando.IsRunning = true;
24     Cliente auxCliente = new Cliente();
25     Medidor auxMedidor = new Medidor();
26     String token = String.Empty;
27     try
28     {
29         //Hacer la solicitud del token de autorizacion.
30         token = JsonConvert.DeserializeObject<String>(await Task.Run(()
31             => autorizarMedidor(Convert.ToString(txtIdMedidor.Text))));
32         if (token != null)
33         {
34             //Hacer la solicitud para la obtencion del medidor.
35             auxMedidor = await Task.Run(()
36                 => obtenerMedidor(Convert.ToString(txtIdMedidor.Text), token));
37             //Hacer la solicitud para la obtencion del cliente.
38             auxCliente = await Task.Run(()
39                 => obtenerCliente(Convert.ToInt32(auxMedidor.IdMedidor), token));
40
41             await this.Navigation.PushModalAsync(new Principal(token, auxCliente, auxMedidor));
42         }
43         else
44         {
45             await this.DisplayAlert("Alerta", "El Medidor no esta Registrado..", "Aceptar");
46         }
47     }
48     catch (Exception ex) { }
49
50     cargando.IsRunning = false;
51
52 }
```

**Código 2.12.** Código para el ingreso a la aplicación móvil.

En el **Código 2.12** en la línea 30 y 31 realiza la petición del *Token* de autenticación el cual es agregado en la cabecera de la solicitud de un servicio web. En las líneas 35 y 38 se realiza la ejecución de los métodos para la obtención de la información del medidor y del cliente respectivamente. Cada uno de estos métodos hace uso de un servicio web para cumplir con su propósito. Por ejemplo, el **Código 2.13** muestra el código necesario para obtener la información de un medidor mediante el uso de un servicio web.

```

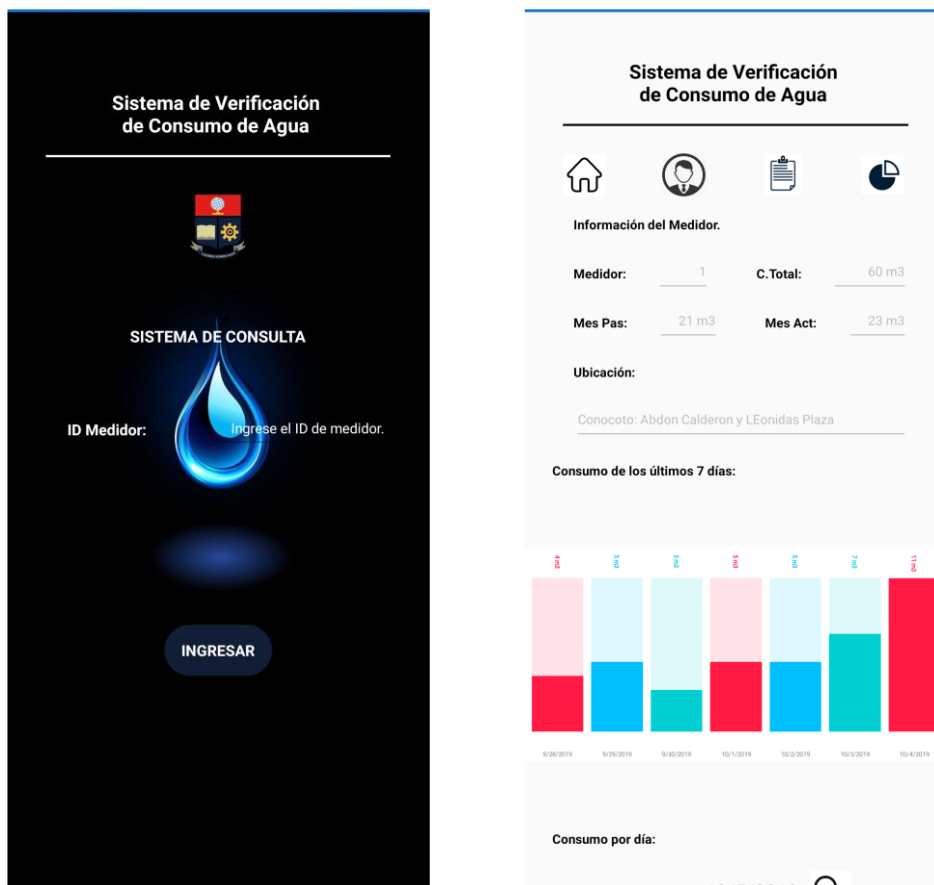
81     async Task<Medidor> obtenerMedidor(string serieMedidor, string token)
82     {
83         Medidor aux = new Medidor();
84
85         var cliente = new RestClient(hostServicio);
86         var request = new RestRequest("api/Medidor/obtenerM", Method.POST);
87         request.AddParameter("serieMedidor", serieMedidor, ParameterType.QueryString);
88         request.AddHeader("Authorization", String.Format("Bearer " + token));
89         IRestResponse<Medidor> response = cliente.Execute<Medidor>(request);
90         aux.Direccion = response.Data.Direccion;
91         aux.IdMedidor = response.Data.IdMedidor;
92         aux.Sector = response.Data.Sector;
93         aux.SerieMedidor = response.Data.SerieMedidor;
94         return aux;
95     }
96 }
97

```

**Código 2.13.** Uso del servicio web en la aplicación móvil.

En el **Código 2.13** desde la línea 85 a la 88 se arma la solicitud del servicio web, en la línea 88 se realiza la petición y se almacena la respuesta del servidor. Desde la línea 90 a la 93 del código se extrae los valores que fueron retornados por el servidor y se los establece en los atributos de la instancia de un objeto del tipo “Medidor”, por último, en la línea 94 se retorna el objeto con la información del medidor solicitado.

Una vez realizada la codificación de los controles (Botones, Etiquetas, Campos de Texto, Iconos, etc.) necesarios para cada una de las interfaces de la aplicación móvil y del funcionamiento de los mismos se ha obtenido las interfaces mostradas a continuación.



**Figura 2.56.** Interfaz de ingreso y principal de la aplicación móvil.

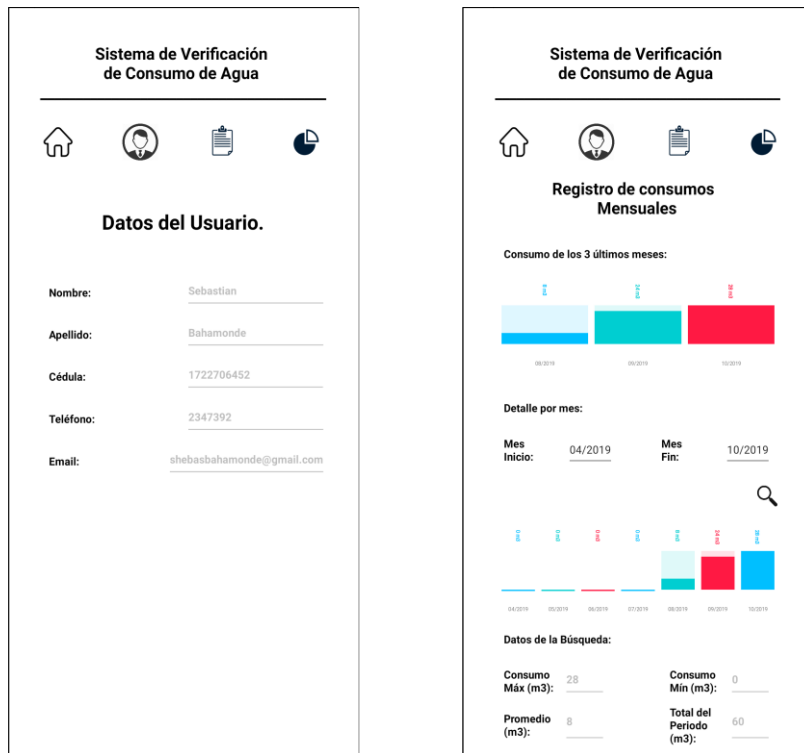


Figura 2.57. Interfaces de datos del cliente y consultas mensuales.

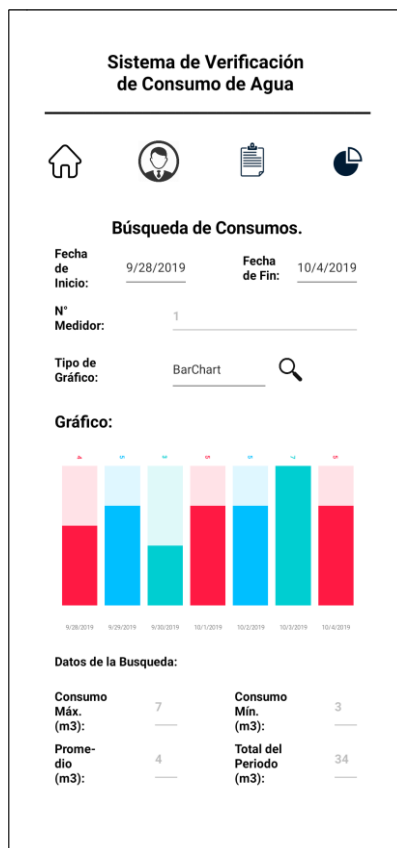


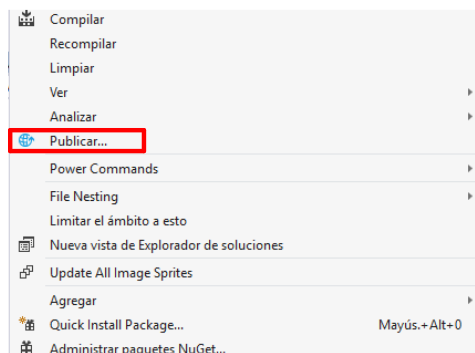
Figura 2.58. Interfaz para consulta de consumo por días.

El código completo para la implementación de las interfaces de la aplicación móvil se encuentra en el **ANEXO G**.

### 2.3.7. PUBLICACIÓN DE LA APLICACIÓN Y SERVICIOS WEB EN LA NUBE AZURE

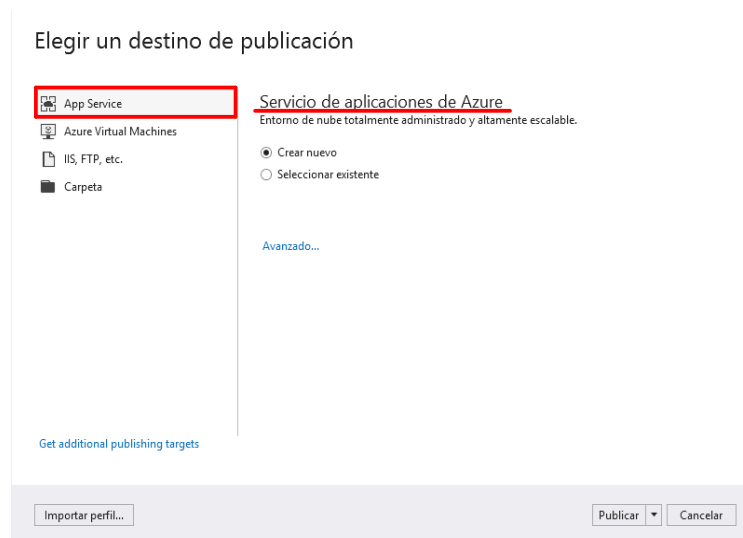
En el siguiente apartado se detalla la publicación de los servicios web y de la aplicación móvil en la nube de Azure. En cualquiera de los dos casos se lo realiza directamente desde el IDE de Visual Studio.

Un requisito para que se pueda realizar la publicación de cualquier proyecto desde el IDE de Visual Studio es tener registrada en el IDE la cuenta con la que se haya creado la suscripción a la nube de Azure. Cumpliendo este requisito se debe ir a las opciones del proyecto y se elige la opción de “Publicar”, esta opción se muestra en la **Figura 2.59**.



**Figura 2.59.** Opción de publicar un proyecto.

Una vez ingresado a esta opción se mostrará la ventana de la **Figura 2.60**, en la cual se debe elegir el destino de la publicación. En este caso se elegirá el “Servicio de aplicaciones de Azure” y se selecciona “Publicar” para continuar con el asistente hasta finalizar el proceso.



**Figura 2.60.** Destino de la publicación del proyecto.

### 3. RESULTADOS Y DISCUSIÓN

En esta sección se detallan las pruebas a las que ha sido sometido el Sistema Distribuido para la Medición de Consumo de Agua; además, se realiza el análisis de los resultados adquiridos. El sistema será puesto a pruebas de funcionalidad de cada uno de los módulos con el objetivo de verificar el cumplimiento de los requerimientos tanto funcionales y no funcionales del sistema.

#### 3.1. ACTUALIZACIÓN DEL TABLERO KANBAN

El tablero Kanban de la **Figura 3.1.** muestra las actividades que se realizaron para la fase de pruebas del prototipo.

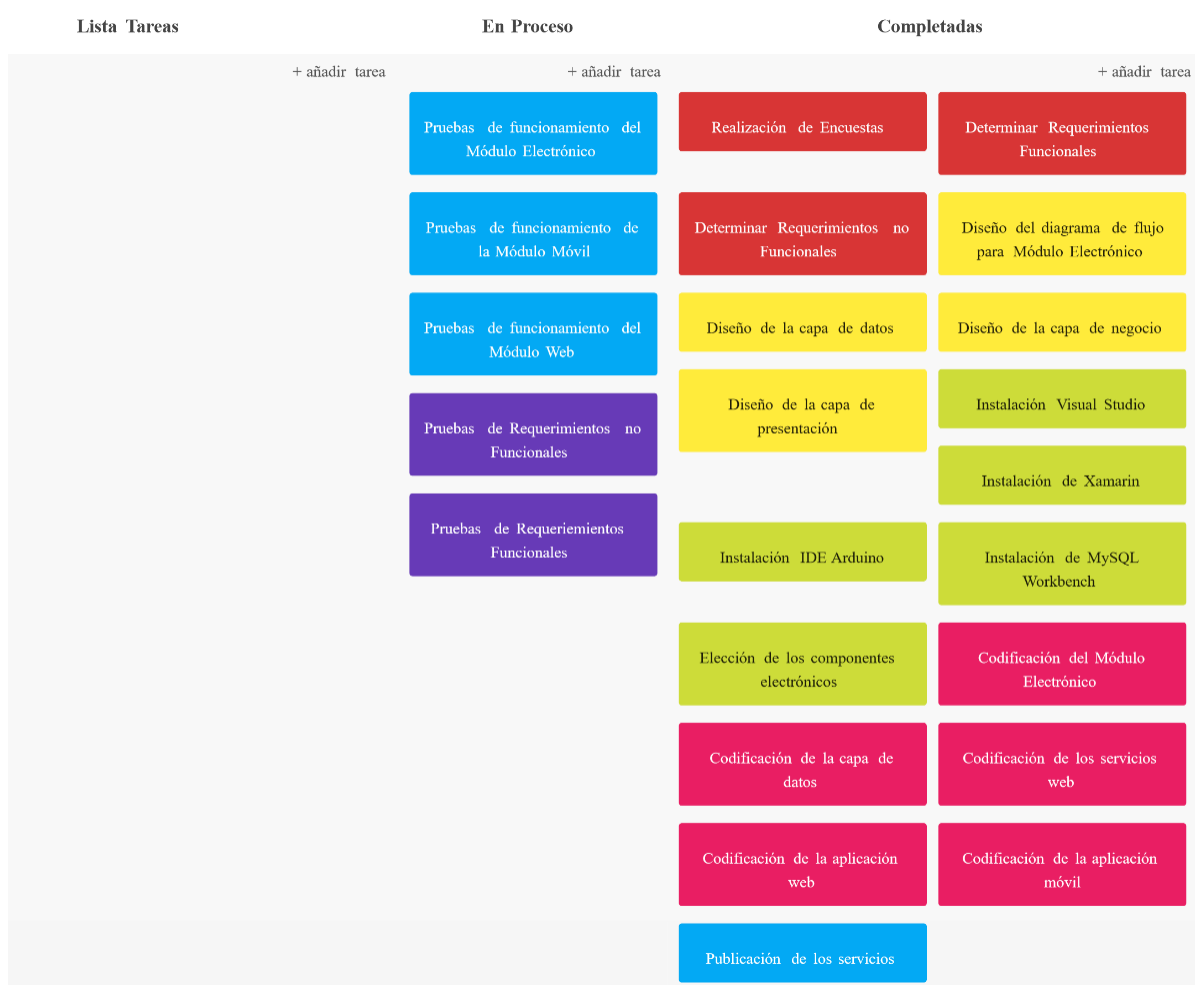


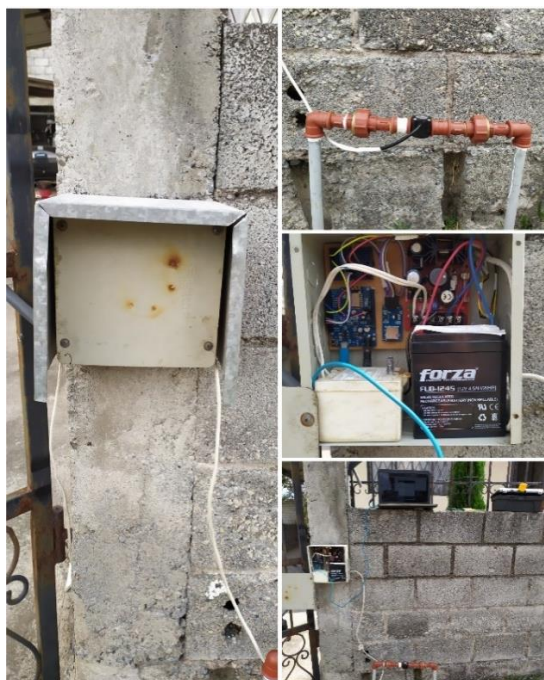
Figura 3.1. Tablero Kanban pruebas.

#### 3.2. PRUEBAS REQUERIMIENTOS FUNCIONALES

Para realizar las pruebas de este tipo de requerimientos se evalúa cada requerimiento descrito en el **Capítulo 2 Apartado 2.2.3.**, especificando cada uno de los requerimientos pertenecientes a cada módulo.

### 3.2.1. PRUEBAS DEL MÓDULO ELECTRÓNICO

Para la realización de las pruebas del Módulo Electrónico se ha realizado la integración del mismo a la red de agua potable de una casa. Integrado este módulo se procedió a realizar la verificación de los Requerimientos Funcionales, los cuales fueron planteados en el **Capítulo II Apartado 2.2.3**. En la **Figura 3.2** muestra la instalación del Módulo Electrónico.



**Figura 3.2.** Prueba de funcionamiento. Instalación del módulo electrónico.

Los Requerimientos Funcionales para este módulo se los detalla en la Tabla 3.1.

**Tabla 3.1.** Requerimientos Funcionales del Módulo Electrónico.

RF	DESCRIPCIÓN
RF01	La placa Arduino debe ser capaz de conectarse a la red inalámbrica del usuario.
RF02	Debe realizar el ingreso del valor de consumo a la base de datos.
RF03	El valor de consumo debe ser almacenado en una memoria SD si no existe conexión a internet para el ingreso en la base de datos.
RF04	El medidor debe funcionar de manera ininterrumpida.

Para el cumplimiento del RF01 se realiza pruebas de conexión del medidor a la red inalámbrica de una casa. La **Figura 3.3** muestra el mensaje de confirmación que el medidor envía por el puerto serial de la placa Arduino. Además, la **Figura 3.4** muestra el uso de un *scanner* de red para corroborar que el dispositivo se ha conectado.

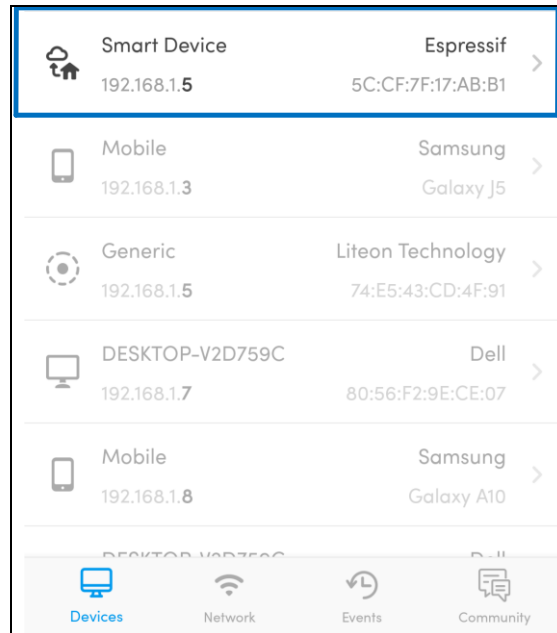


```

Se ha conectado a la red inalambrica.
Iniciado el SD
Hora: 9
Valor Consumido: 0 L

```

**Figura 3.3.** Prueba de funcionamiento. Prueba de conexión a la red inalámbrica.



**Figura 3.4.** Prueba de funcionamiento. Comprobación de conexión a la red inalámbrica.

Para el cumplimiento del RF02 la **Figura 3.5** muestra el mensaje para conocer si se ingresó el valor de consumo a la base de datos; y la **Figura 3.6** muestra el ingreso del valor de consumo en la base de datos.

```

Se ha conectado a la red inalambrica.
Iniciado el SD
Hora: 17
Valor Consumido: 0 L
Hora de ingreso.
Conectando a: apiprojectotitulacionsebas.azurewebsites.net
Conexion exitosa
Petición URL: /api/Login/ingresarConsumo?date=2019-11-12&valor=0&serieMedidor=med001
Se ha ingresado en la base de datos.
Cerrando Conexion

```

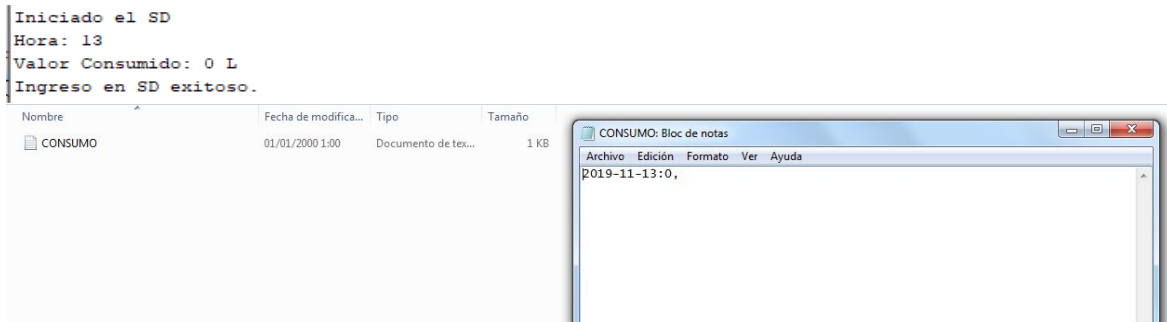
**Figura 3.5.** Prueba de funcionamiento. Ingreso del valor de consumo en la base de datos.

	idMedida	idMedidor	consumo	fechaMedicion
▶	1	1	0	2019-11-12
*	NULL	NULL	NULL	NULL

**Figura 3.6.** Prueba de funcionamiento. Valor de consumo en la base de datos.

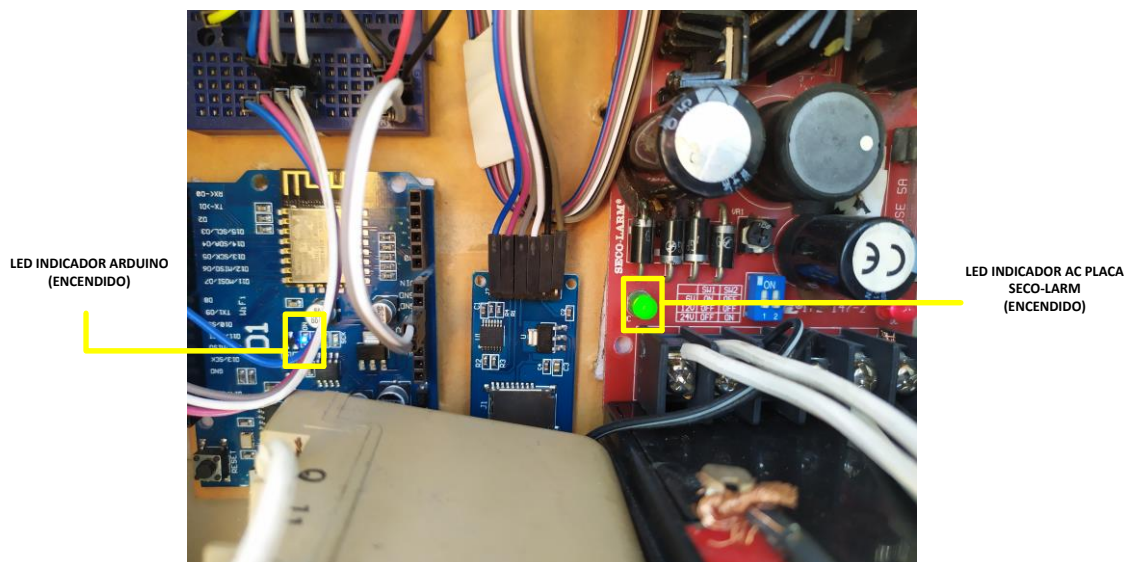
Para el cumplimiento del RF03, al módulo se le ha negado el acceso a la red inalámbrica de la casa momentáneamente. Por lo cual el módulo procedió a realizar el ingreso del valor de consumo en la memoria SD. La **Figura 3.7.** muestra los mensajes de confirmación para

conocer si el valor de consumo se ha ingresado en la memoria SD, así como también la vista del archivo donde se guarda el valor de consumo junto con la fecha en que se ha realizado la toma del valor.



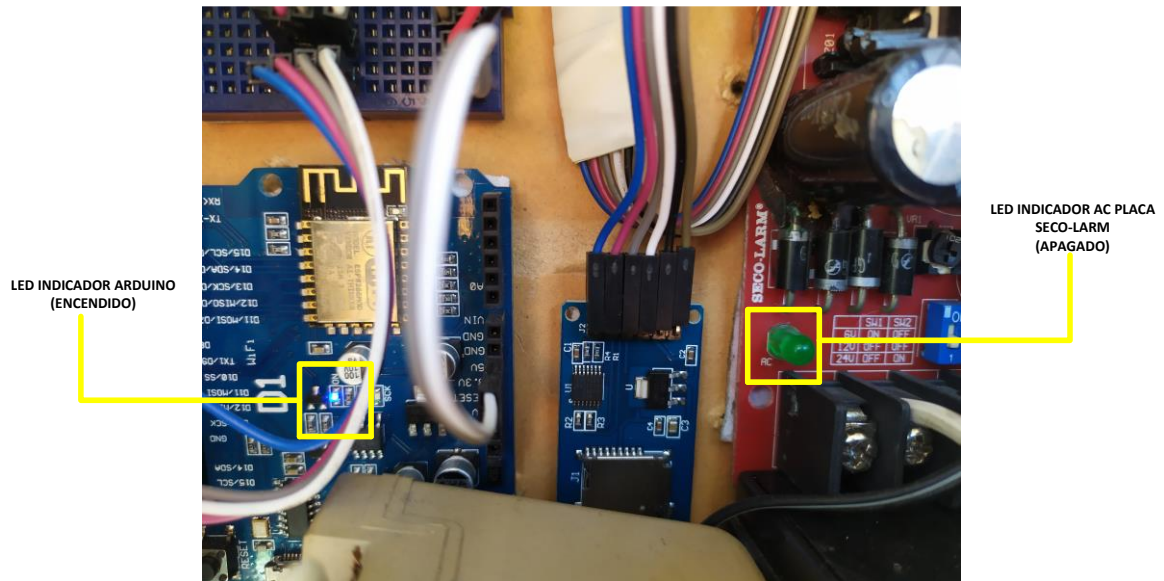
**Figura 3.7.** Prueba de funcionamiento. Ingreso del valor de consumo en la memoria SD. Por último, para el cumplimiento de RF04 se realizó pruebas de funcionamiento de la placa SECO-LARM que es la encargada de energizar al Módulo Electrónico con la energía eléctrica de la casa o con la de una batería.

En la **Figura 3.8** se muestra el funcionamiento del Módulo Electrónico con energía eléctrica de la casa. En esta figura se observa que el led alimentación de energía “AC” de la placa SECO-LARM y el led de “ON” de la placa Arduino se encuentran encendidos.



**Figura 3.8.** Prueba de Funcionamiento. Alimentación al Sistema con energía AC.

A continuación, se procedió a desconectar momentáneamente de la energía eléctrica de la casa a la placa SECO-LARM y en la **Figura 3.9** se muestra que el led de alimentación AC la placa SECO-LARM se encuentra apagado, pero el led de “ON” de la placa Arduino aún se encuentra encendido; esto es debido a que se encuentra funcionando con la energía de la batería.



**Figura 3.9.** Prueba de funcionamiento. Respaldo de energía con batería.

Además, también se hizo pruebas del valor medido por Módulo Electrónico. Estas pruebas se las realizó haciendo pasar un litro de agua por el medidor actual y observando el valor medido por el Módulo Electrónico, los resultados de estas pruebas se presentan en la **Tabla 3.2.**

**Tabla 3.2.** Prueba de funcionamiento. Tabla de comparación de mediciones.

Medición	Medidor Actual (L)	Prototipo (L)
1	1.2	1,1
2	1	1,3
3	1	0,8
4	1	0,9
5	1	0,8
6	1	0,9
7	1	1,1
8	1	0,9
9	1	1,2
10	1	1,1
<b>Total</b>	10,2	10,1

Como se puede observar en la **Tabla 3.2.** existen diferencias entre los valores obtenidos por el prototipo y el medidor de agua actual. Esto se puede ser debido a la apreciación que se tuvo al momento de observar el valor en el medidor de agua actual, pero concluidas las

mediciones el margen de error se acerca al 0,98% por lo cual las mediciones entre el prototipo y el medidor actual son semejantes.

El registro de las medidas tomadas entre el medidor de agua actual y el Módulo Electrónico desarrollado se encuentra en el **ANEXO H**.

### 3.2.2. PRUEBAS DEL MÓDULO WEB


En este apartado se realizó las pruebas necesarias para el cumplimiento de los Requerimientos Funcionales que se muestran en la **Tabla 3.3**. Para realizar estas pruebas se ha ingresado la información de un administrador para el uso del sistema.

**Tabla 3.3.** Requerimientos Funcionales del Módulo Web.

RF	DESCRIPCIÓN
RF05	Debe realizar una autenticación mediante un nombre de usuario y contraseña.
RF06	Debe permitir la administración de los usuarios.
RF07	Debe permitir la administración de los medidores.
RF08	Debe permitir realizar consultas de consumos de cada medidor.
RF09	Debe permitir el envío de notificaciones mediante correos electrónicos.

Para el cumplimiento del Requerimiento Funcional RF05, la aplicación web desplegará un formulario el cual se muestra en la **Figura 3.10** en donde se debe realizar el ingreso de la información solicitada; así también en la **Figura 3.11** se muestra los mensajes dirigidos hacia el usuario en casos de existir errores al ingresar a la aplicación.

SISTEMA DE VERIFICACION DE CONSUMO DE AGUA



Ingreso al Sistema

**Figura 3.10.** Ingreso a la aplicación web.

## SISTEMA DE VERIFICACION DE CONSUMO DE AGUA

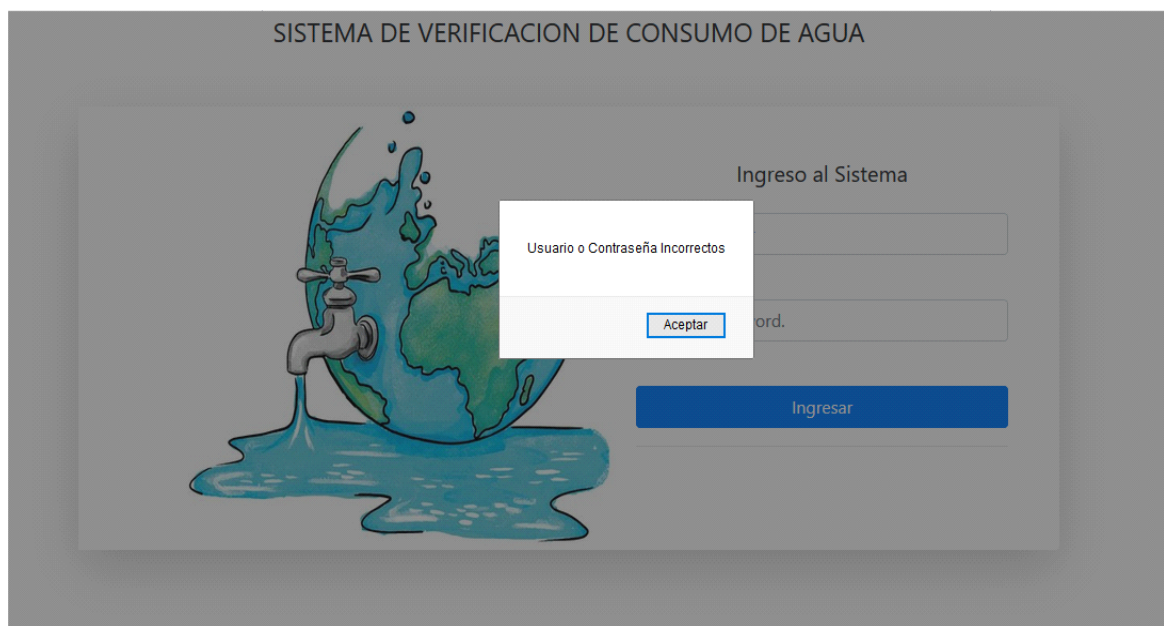
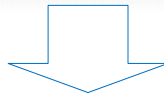


Ingreso al Sistema

Ingrese userName  
\* Campo Requerido

Ingrese password.  
\* Campo Requerido

Ingresar



SISTEMA DE VERIFICACION DE CONSUMO DE AGUA

Ingreso al Sistema

Usuario o Contraseña Incorrectos

Aceptar

Ingresar

**Figura 3.11.** Prueba de funcionalidad. Ingreso de Administrador.

Los Requerimientos Funcionales RF06 y RF07 del Módulo Web consisten en la administración (creación, eliminación, modificación y lectura) de clientes y medidores. En la **Figura 3.12** se muestra el cumplimiento de la creación de un cliente y un medidor. Además, la **Figura 3.13**. muestra la validación que se realiza para verificar si el usuario ya ha sido, o no registrado en el sistema al igual que el medidor.

ADMINISTRACIÓN. Sistema de Verificación de Consumo de Agua. Jorge Sebastian Bahamonde Achig

Principal

OPCIONES

- Medidores
- Administradores
- Usuarios

INGRESO DE USUARIOS.

DATOS DEL CLIENTE.		DATOS DEL MEDIDOR.	
Nombre:	Nombre.	Nº de Serie:	Nº de Serie.
Apellido:	Apellido.	Dirección:	Dirección.
Cédula:	Cédula.	Sector:	Sector.
Correo Electrónico:	Correo Electrónico.		
Teléfono:	Teléfono.		

Guardar

© Sistema de Verificación de Consumo de Agua 2019

ADMINISTRACIÓN. Sistema de Verificación de Consumo de Agua. Jorge Sebastian Bahamonde Achig

Principal

OPCIONES

- Medidores
- Administradores
- Usuarios

INGRESO DE USUARIOS.

DATOS DEL CLIENTE.		DATOS DEL MEDIDOR.	
Nombre:	Nombre. *requerido	Nº de Serie:	Nº de Serie. *requerido
Apellido:	Apellido. *requerido	Dirección:	Dirección. *requerido
Cédula:	Cédula. *requerido	Sector:	Sector. *requerido
Correo Electrónico:	Correo Electrónico. *requerido		
Teléfono:	Teléfono. *requerido		

Guardar

© Sistema de Verificación de Consumo de Agua 2019

ADMINISTRACIÓN. Sistema de Verificación de Consumo de Agua. Jorge Sebastian Bahamonde Achig

Principal

OPCIONES

- Medidores
- Administradores
- Usuarios

INGRESO DE USUARIOS.

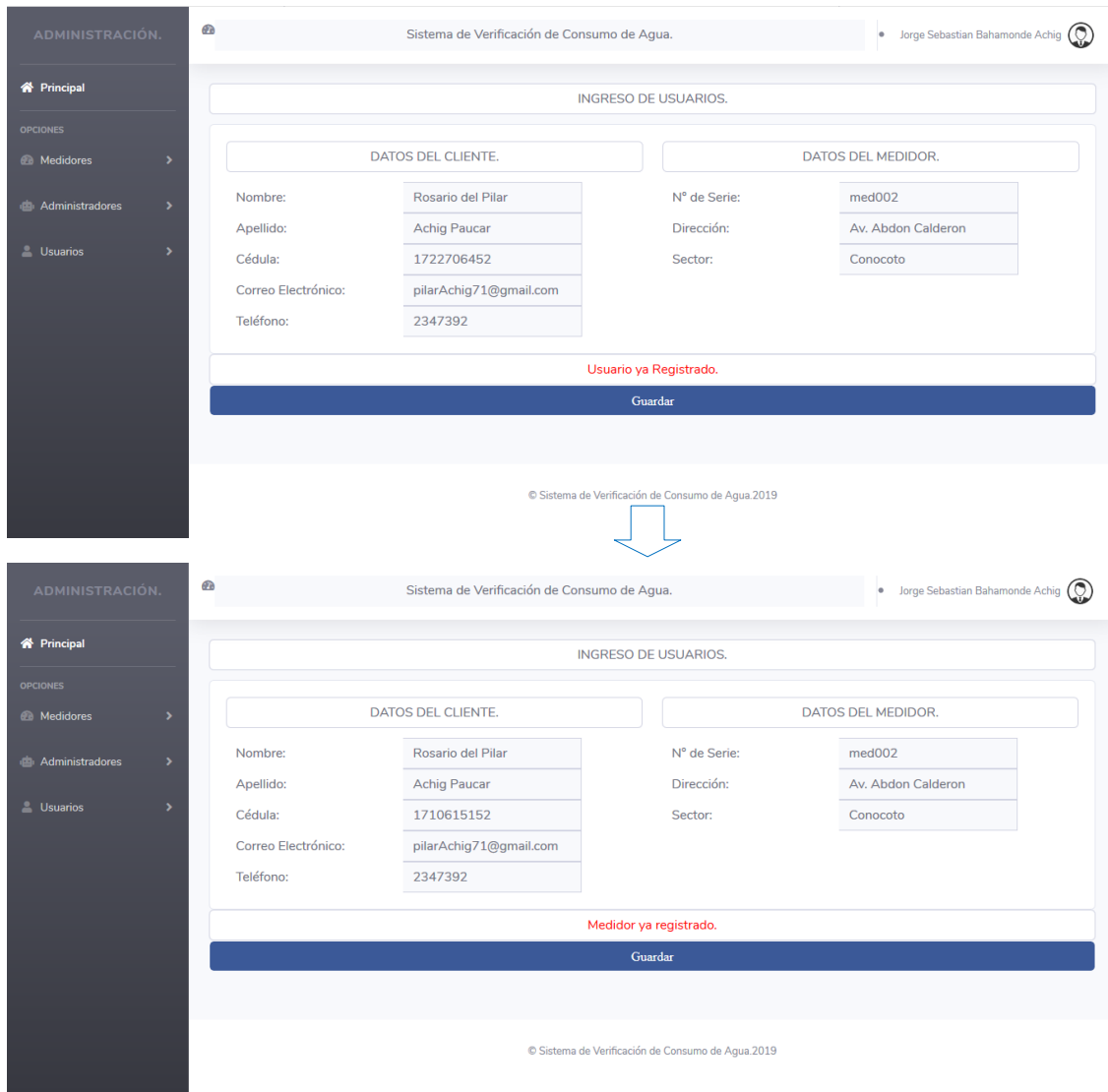
DATOS DEL CLIENTE.		DATOS DEL MEDIDOR.	
Nombre:	Nombre.	Nº de Serie:	Nº de Serie.
Apellido:	Apellido.	Dirección:	Dirección.
Cédula:	Cédula.	Sector:	Sector.
Correo Electrónico:	Correo Electrónico.		
Teléfono:	Teléfono.		

Registro exitoso.

Guardar

© Sistema de Verificación de Consumo de Agua 2019

Figura 3.12. Prueba de funcionamiento. Registro de cliente y medidor.



**Figura 3.13.** Prueba de funcionamiento. Validaciones registro cliente y medidor. Como verificación, el registro del cliente y medidor en la base de datos se lo muestra en la **Figura 3.14.**

	idCliente	nombres	apellidos	mail	cedula	telefono
▶	8	Rosario del Pilar	Achig Paucar	pilarAchig71@gmail.com	1710615152	2347392
*	NULL	NULL	NULL	NULL	NULL	NULL

	idMedidor	direccion	sector	serieMedidor	idCliente
▶	11	Av. Abdon Calderon	Conocoto	med006	8
*	NULL	NULL	NULL	NULL	NULL

**Figura 3.14.** Prueba de funcionamiento. Ingreso del cliente y medidor en la base de datos.

La siguiente parte de estos requerimientos es la modificación de la información; la **Figura 3.15** muestra el cumplimiento de la modificación de la información del usuario que a su vez muestra cuadros de dialogo para la confirmación de las acciones de “Modificar” o “Eliminar”.



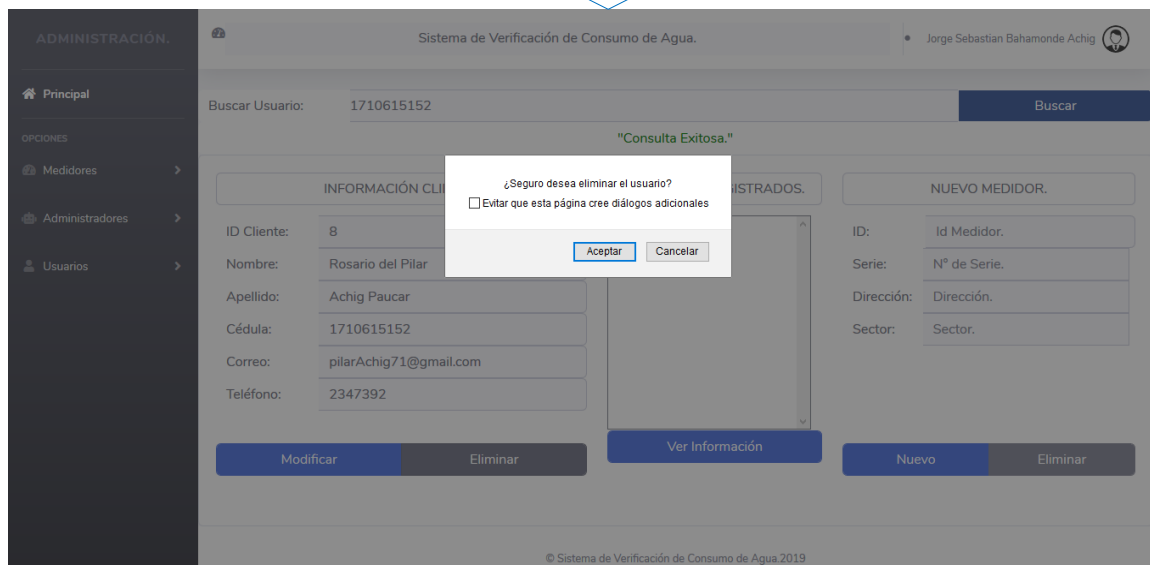
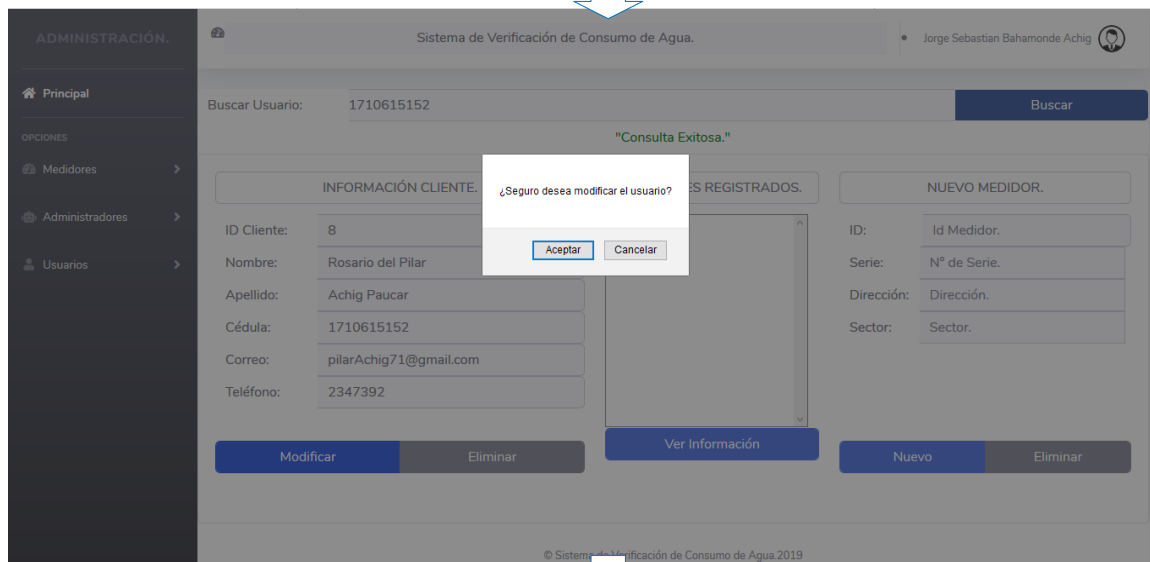
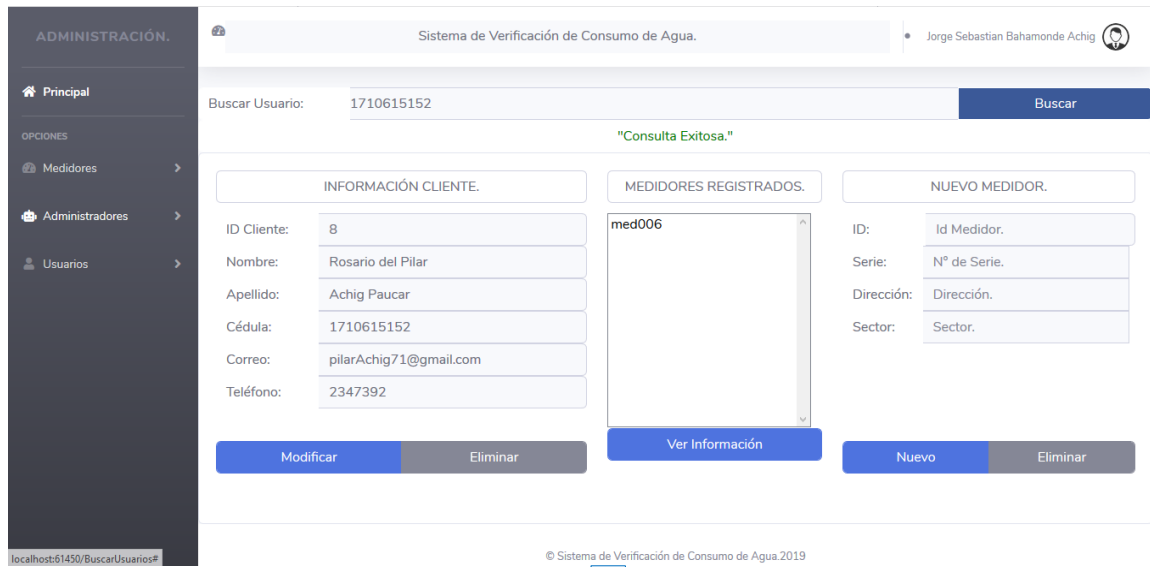


Figura 3.15. Prueba de funcionamiento. Eliminación y Modificación de un cliente.



Así también se puede realizar la modificación de los medidores que se encuentran ligados al cliente. La Figura 3.16. muestra el cumplimiento de lo antes detallado.

NUEVO MEDIDOR.

Serie:	N° de Serie.	*
Dirección:	Dirección.	*
Sector:	Sector.	*

Nuevo Eliminar

Guardar

"Todos los campos son necesarios."

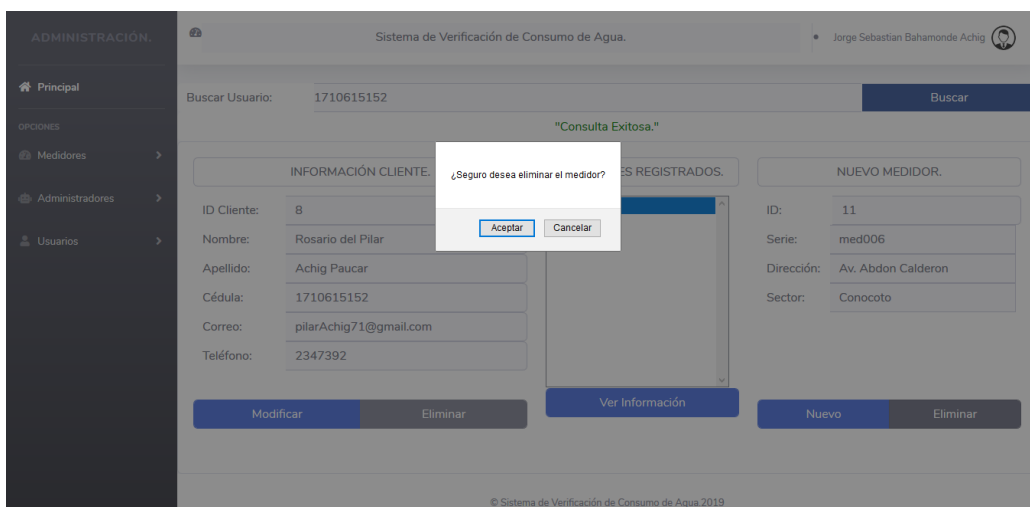
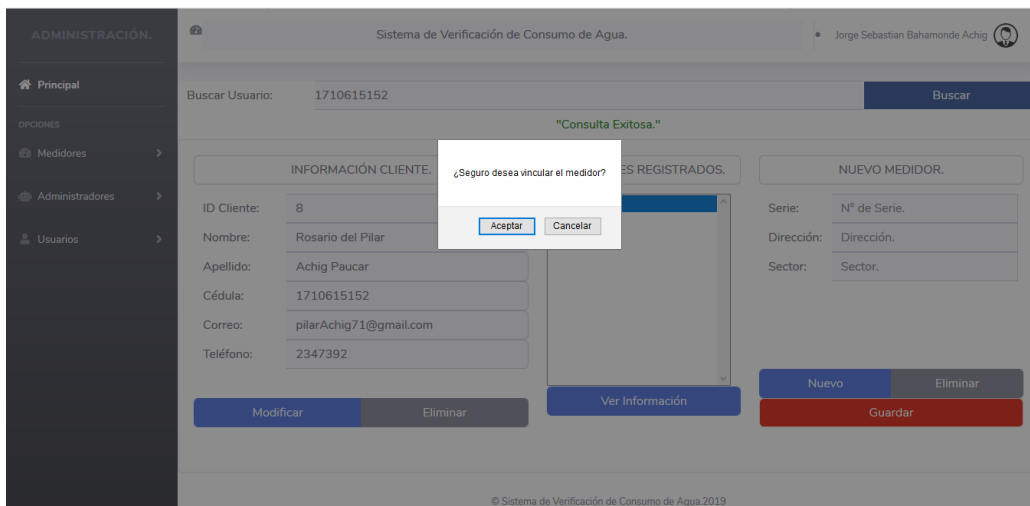


Figura 3.16. Pruebas de funcionamiento. Modificación y eliminación de medidores.

Por último, se tiene las pruebas para el cumplimiento del requerimiento RF08. La **Figura 3.17.** muestra la interfaz en donde se realiza la consulta de consumos.

The figure displays three sequential screenshots of a web application interface for water consumption verification. The interface is titled "Sistema de Verificación de Consumo de Agua." and shows the user "Jorge Sebastian Bahamonde Achig".

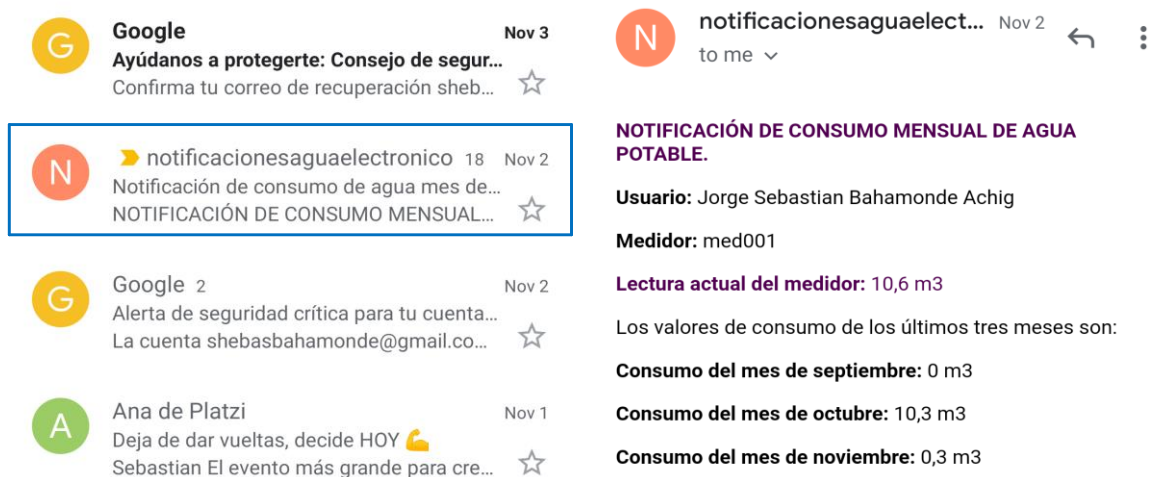
**Screenshot 1 (Top):** Shows the search results for meter ID "med001". The status is "Consulta Exitosa." and the general information includes: Usuario: Jorge Sebastian Bahamonde A..., Medidor: med001, Ubicación: Conocoto Abdon Calderon y Le. Summary statistics: Consumo octubre: 10,3 m<sup>3</sup>, Consumo noviembre: 1,7 m<sup>3</sup>, Consumo Total: 12 m<sup>3</sup>. The search parameters are set to "Días" with no specific date range.

**Screenshot 2 (Middle):** Shows the search results for meter ID "med001" with specific date ranges. The search parameters are: Tipo de Consulta:  Días, Fecha Inicial: 04/11/2019, Fecha Final: 10/11/2019. The bar chart shows consumption values in Liters (L) for the following dates: 4/11/2019 (120 L), 5/11/2019 (120 L), 6/11/2019 (150 L), 7/11/2019 (200 L), 8/11/2019 (200 L), 9/11/2019 (250 L), and 10/11/2019 (44 L). Summary statistics: C. Máx: 253 L, C. Mín: 44 L, Promedio: 157 L, Total: 1100 L.

**Screenshot 3 (Bottom):** Shows the search results for meter ID "med001" with specific date ranges. The search parameters are: Tipo de Consulta:  Meses, Fecha Inicial: 04/09/2019, Fecha Final: 10/11/2019. The bar chart shows consumption values in m<sup>3</sup> for the following months: 09/2019 (0 m<sup>3</sup>), 10/2019 (10,4 m<sup>3</sup>), and 11/2019 (1,7 m<sup>3</sup>). Summary statistics: C. Máx: 10,3 m<sup>3</sup>, C. Mín: 0 m<sup>3</sup>, Promedio: 4 m<sup>3</sup>, Total: 12 m<sup>3</sup>.

**Figura 3.17.** Pruebas de funcionamiento. Realización de consultas por días o meses.

El cumplimiento del Requerimiento Funcional RF09 se muestra en la **Figura 3.18.**, donde se muestra el correo que llega al usuario, en donde se detalla de los consumos que ha realizado en los últimos tres meses, además, de la última lectura del medidor.



**Figura 3.18.** Prueba de funcionamiento. Recepción de la notificación.

### 3.2.3. PRUEBAS DEL MÓDULO MÓVIL

Para realizar las pruebas de los Requerimientos Funcionales del Módulo Móvil los cuales se muestran en la **Tabla 3.4**, se ha registrado un cliente con su medidor mediante el Módulo Web.

**Tabla 3.4.** Requerimientos Funcionales del Módulo Móvil.

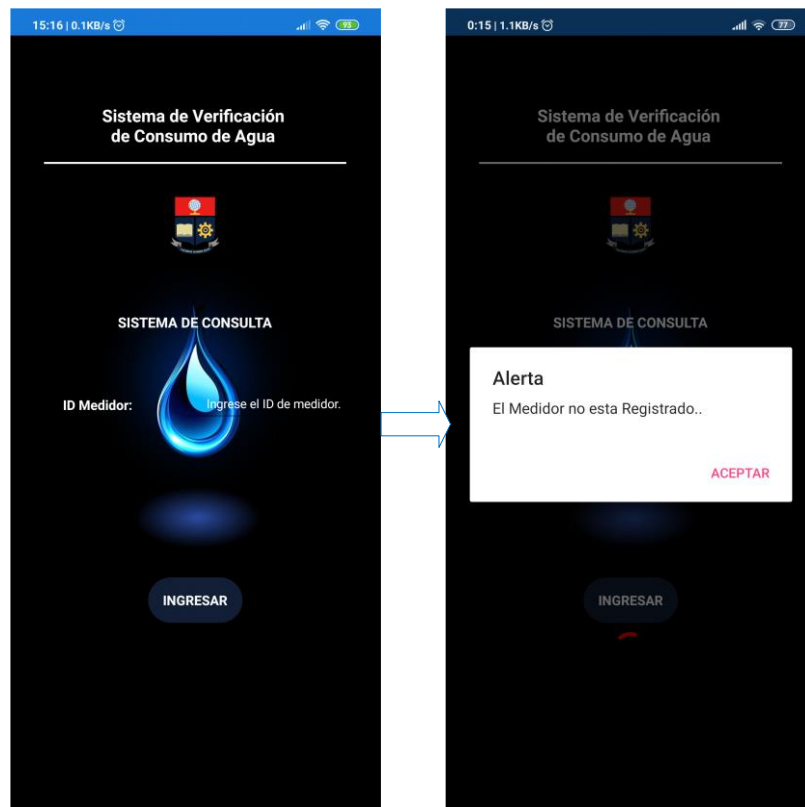
RF	DESCRIPCIÓN
RF10	Debe mostrar la información del usuario del medidor.
RF11	Se debe realizar el ingreso a la aplicación móvil mediante el número de serie del medidor.
RF12	Se debe mostrar el consumo total y consumo de meses anteriores.
RF13	Debe realizar consultas de los consumos por día, días y meses.

El cumplimiento del Requerimiento Funcional RF10, se presenta la pantalla de la **Figura 3.19** en donde se muestra la información del usuario propietario del medidor.



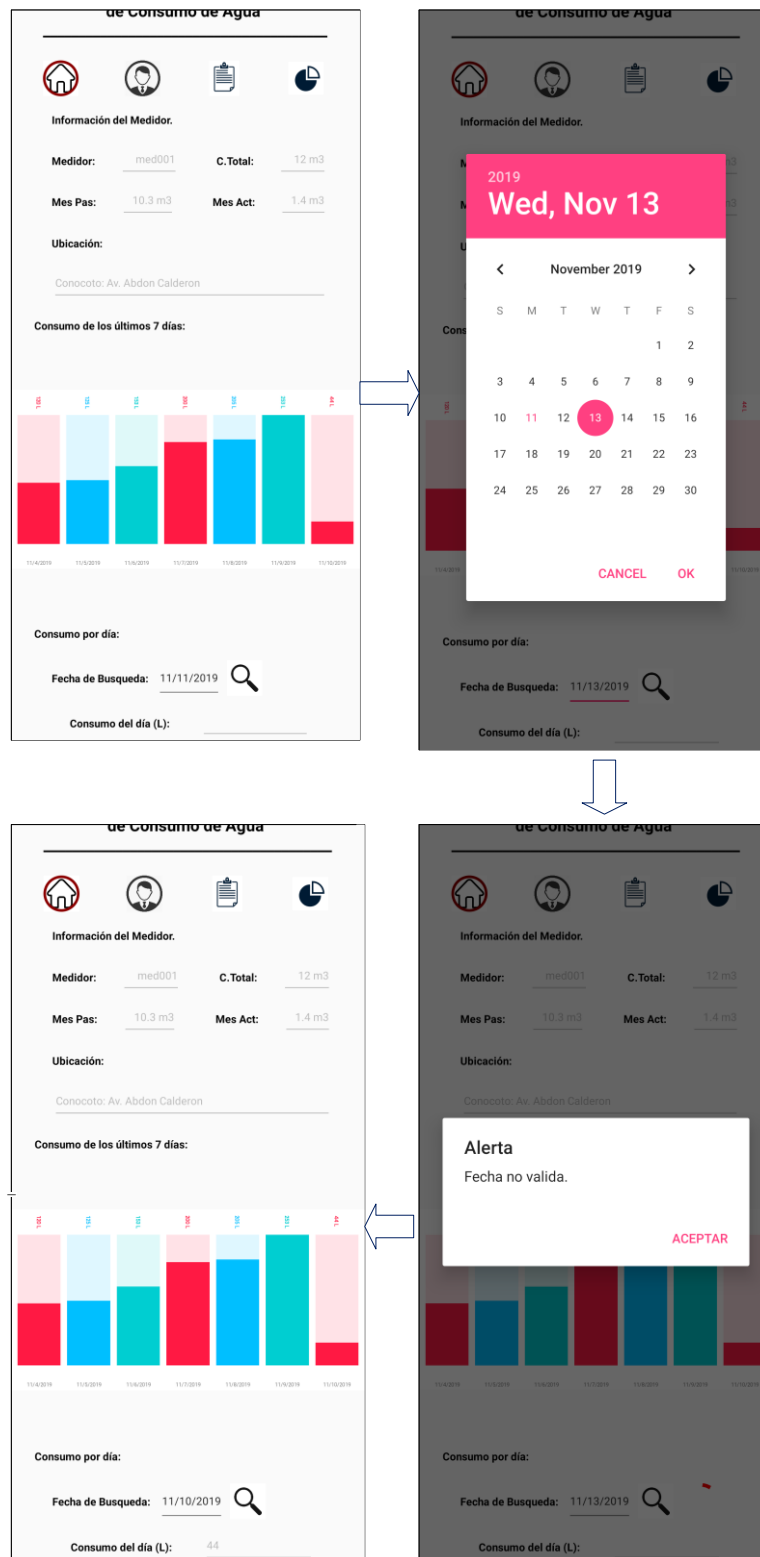
**Figura 3.19.** Prueba de Funcionamiento. Información del cliente.

Para mostrar el cumplimiento del Requerimiento Funcional RF11, se debe realizar el ingreso del número de serie del medidor, el cual es único. La **Figura 3.20** muestra la interfaz en donde se debe realizar el ingreso de un número de serie valido para acceder a la aplicación, ya que caso contrario se mostrará un mensaje dirigido hacia el usuario con el mensaje de que el número de serie ingresado no está registrado.



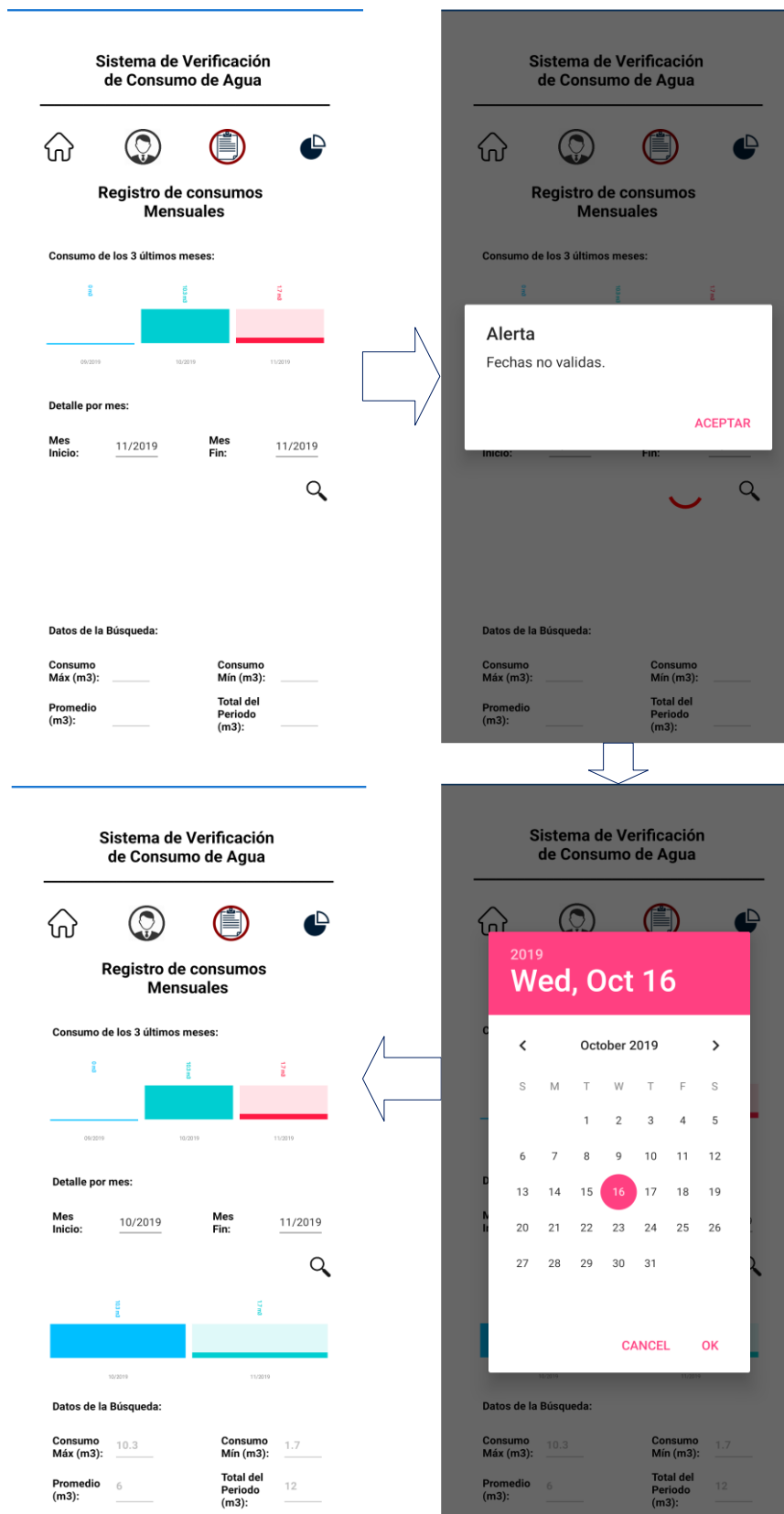
**Figura 3.20.** Prueba de funcionalidad. Ingreso aplicación móvil.

Una vez que se haya logrado realizar la autenticación del medidor se tiene tres tipos de pantallas donde se realizan las acciones necesarias para el cumplimiento de los requerimientos RF12 y RF13. En la **Figura 3.21** se observa el valor de consumo total del medidor; además, de poder realizar la consulta del consumo en un día específico.



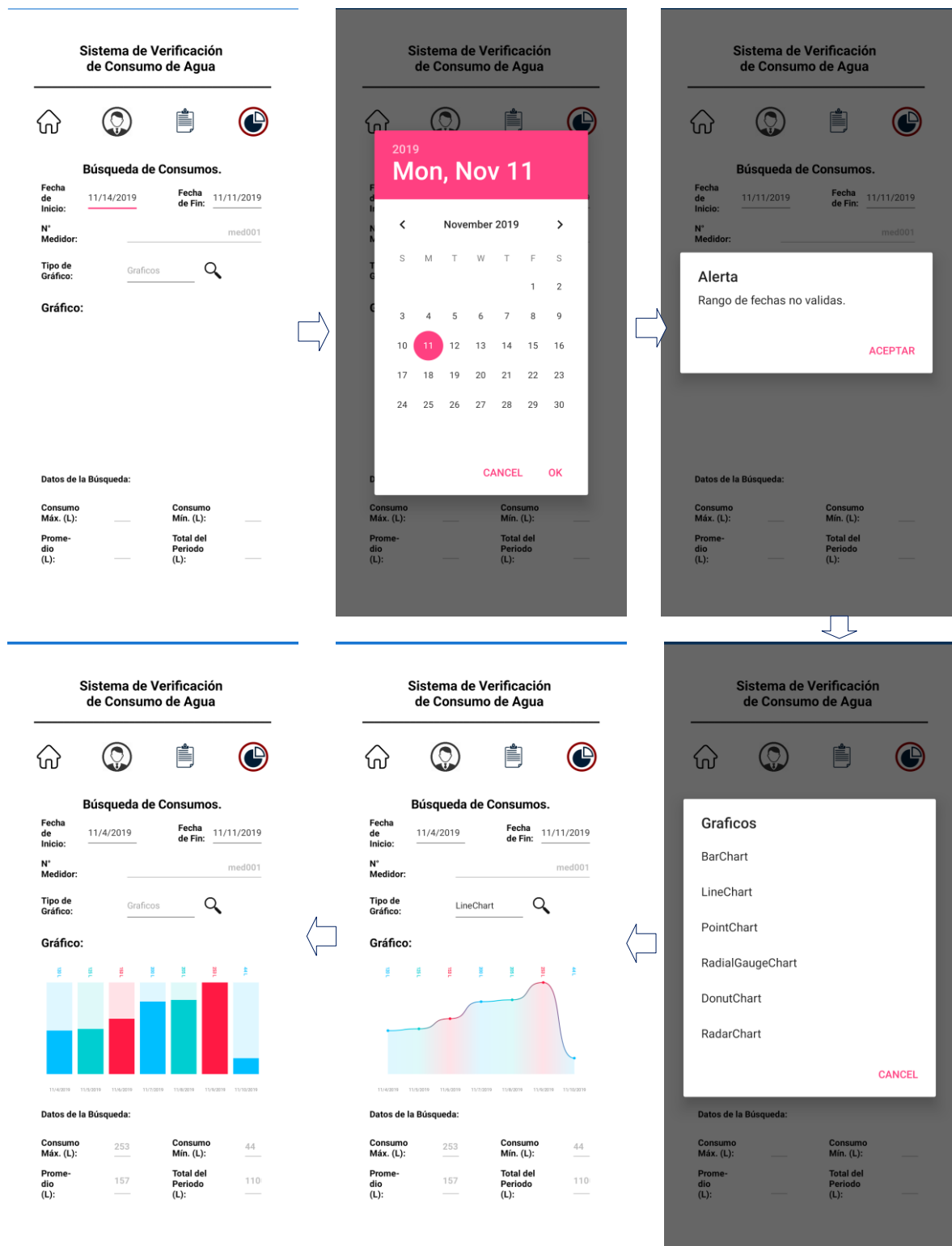
**Figura 3.21.** Prueba de funcionamiento. Consulta de consumo por día.

La **Figura 3.22** muestra la consulta del consumo por meses, y la información del consumo de los tres últimos meses.



**Figura 3.22.** Prueba de funcionamiento. Consulta de consumos mensuales.

La **Figura 3.23** muestra la consulta de consumos por días. En este tipo de consulta se puede observar diferentes tipos de gráficos de los consumos.



**Figura 3.23.** Prueba de funcionamiento. Consulta de consumos por días.

Por último, en la **Tabla 3.5** se muestra un resumen del cumplimiento de los Requerimientos Funcionales del sistema en su totalidad.

**Tabla 3.5.** Resumen del cumplimiento de los Requerimientos Funcionales.

<b>MÓDULO</b>	<b>RF</b>	<b>DESCRIPCIÓN</b>	<b>HORAS</b>	<b>ESTADO</b>
<b>Electrónico</b>	RF01	La placa Arduino debe ser capaz de conectarse a la red inalámbrica del usuario.	4	OK
	RF02	Debe realizar el ingreso del valor de consumo a la base de datos.	4	OK
	RF03	El valor de consumo debe ser almacenado en una memoria SD si no existe conexión a internet para el ingreso en la base de datos.	8	OK
	RF04	El medidor debe funcionar de manera ininterrumpida.	4	OK
<b>Aplicación Web</b>	RF05	Debe realizar una autenticación mediante un nombre de usuario y contraseña.	5	OK
	RF06	Debe permitir la administración de los usuarios.	5	OK
	RF07	Debe permitir la administración de los medidores.	6	OK
	RF08	Debe permitir realizar consultas de consumos de cada medidor.	4	OK
	RF09	Debe permitir el envío de notificaciones mediante correos electrónicos.	4	OK
<b>Aplicación Móvil</b>	RF10	Debe mostrar la información del usuario del medidor.	3	OK
	RF11	Debe realizar la autenticación mediante el número de serie del medidor.	4	OK
	RF12	Se debe mostrar el consumo total y consumo de meses anteriores.	8	OK
	RF13	Debe realizar consultas de los consumos por día, días y meses.	4	OK



### 3.3. PRUEBAS REQUERIMIENTOS NO FUNCIONALES

En el siguiente apartado de realizará la comprobación de todos los Requerimientos No Funcionales, los cuales se muestran en la **Tabla 3.6.**

**Tabla 3.6.** Requerimientos No Funcionales del sistema.

RNF	NOMBRE	DESCRIPCIÓN
RNF01	Autenticación	El acceso al Módulo Web debe estar limitado por un nombre de usuario y una contraseña para cada usuario.
RNF02	Operabilidad	Las aplicaciones del prototipo deben ser de fácil manejo. Deben tener mensajes informativos orientados hacia el usuario.
RNF03	Acceso vía internet	Las aplicaciones del prototipo deben poder ser usadas desde cualquier lugar en el que se encuentren haciendo uso de Internet.
RNF04	Disponibilidad	El módulo y los servicios web deberán estar disponibles 24 horas los 7 días de la semana.

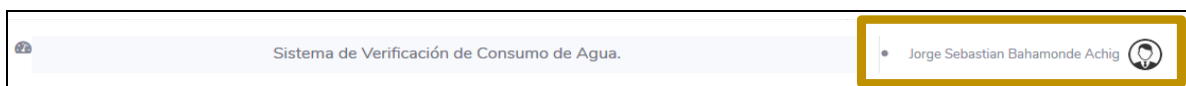
#### 3.3.1. RNF01: AUTENTICACIÓN

El acceso al Módulo Web se encuentra limitado por un “nombre de usuario” o “*userName*” y contraseña para cada Administrador. La **Figura 3.24.** muestra la interfaz donde el Administrador debe realizar el ingreso de las credenciales antes mencionadas, validación de los campos cuando no se haya ingresado ningún valor.

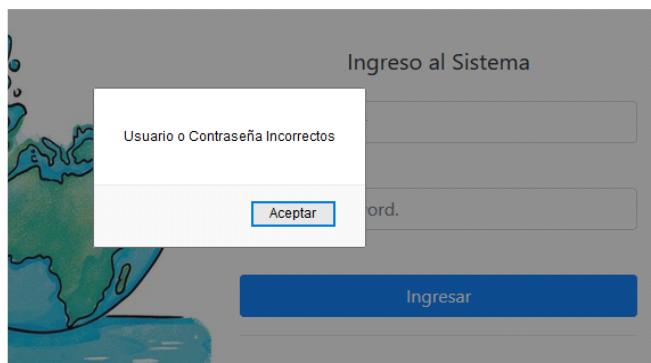
The figure displays two versions of the login interface. Both are titled 'Ingreso al Sistema'. The left version shows a functional form with two input fields: 'Ingrese userName' and 'Ingrese password.', and a blue 'Ingresar' button. The right version shows the same form but with validation errors. Below each input field, there is a red asterisk and the text '\* Campo Requerido', indicating that the fields are required and currently empty.

**Figura 3.24.** Interfaz de inicio de sesión.

Una vez se haya ingresado las credenciales correctamente y se presione el botón ingresar, se ingresará a la pantalla principal del Módulo Web donde se mostrará el nombre del Administrador como en la **Figura 3.25**. Por lo contrario, se mostrará el mensaje de la **Figura 3.26**, indicando que el nombre del usuario o la contraseña son incorrectos.



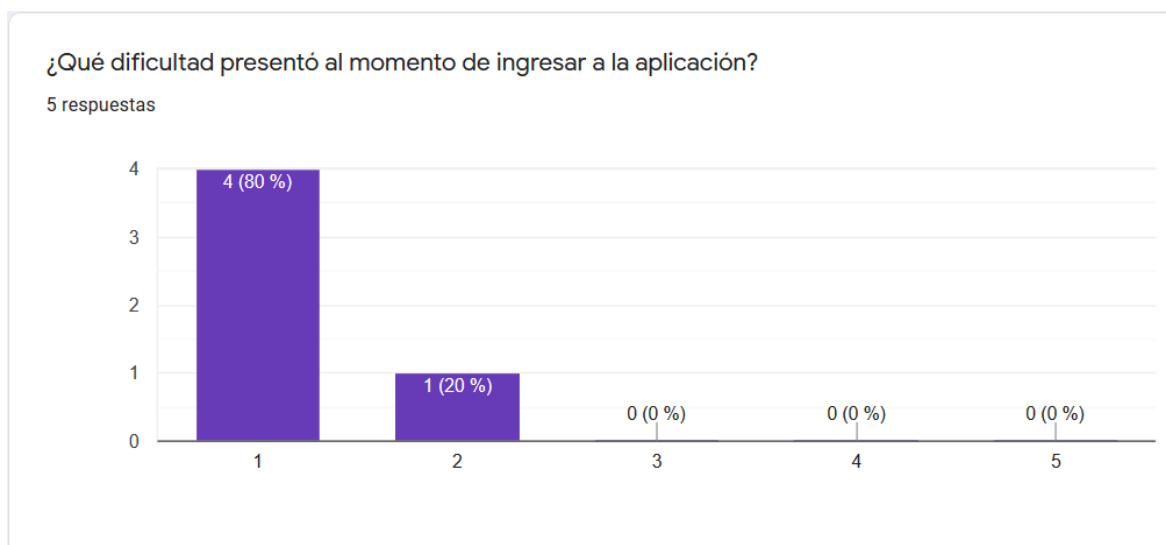
**Figura 3.25.** Nombre de usuario en el Módulo Web.



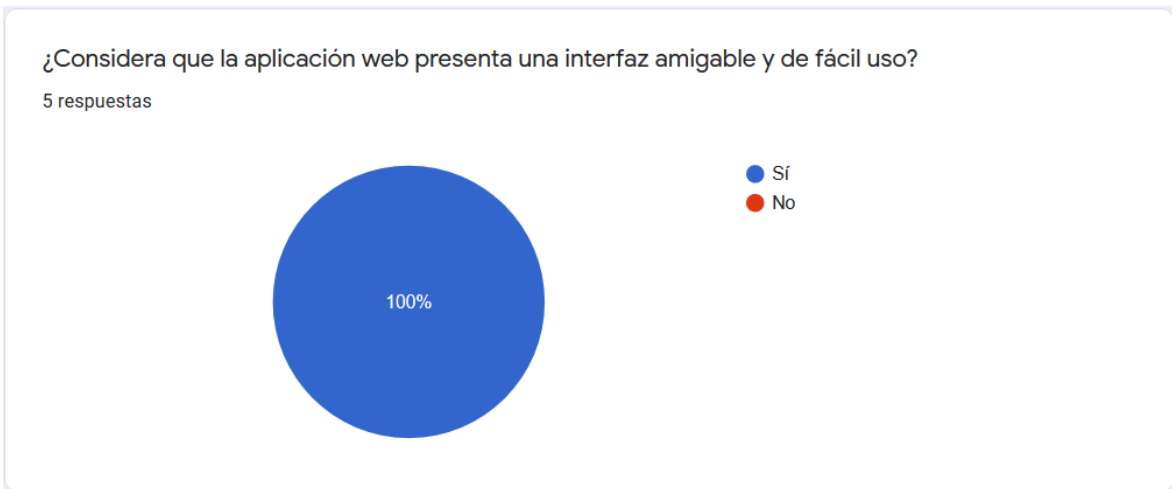
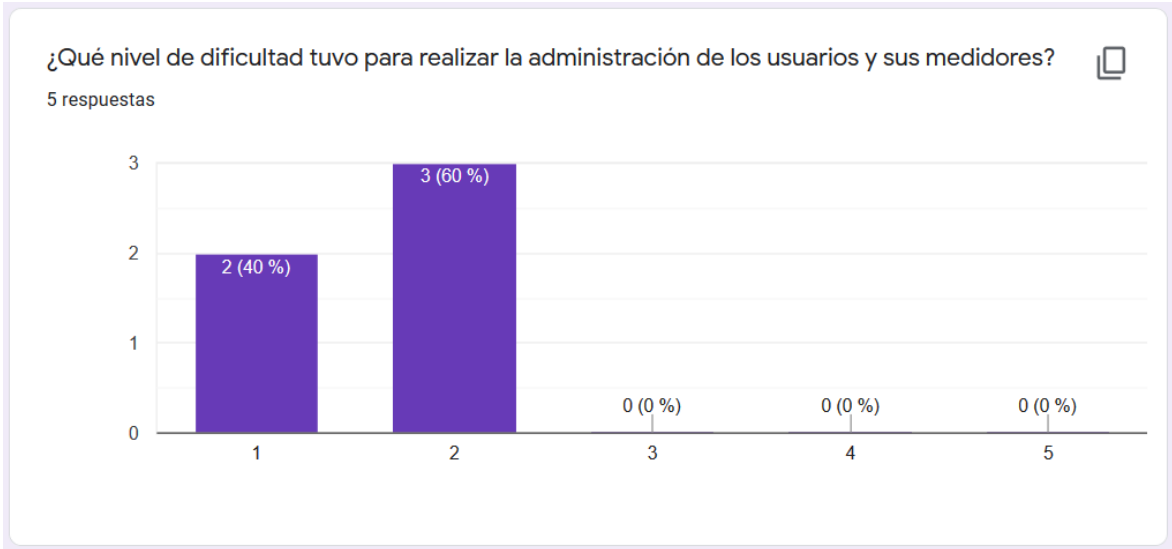
**Figura 3.26.** Mensaje de autenticación fallida.

### 3.3.2. RNF02: OPERABILIDAD

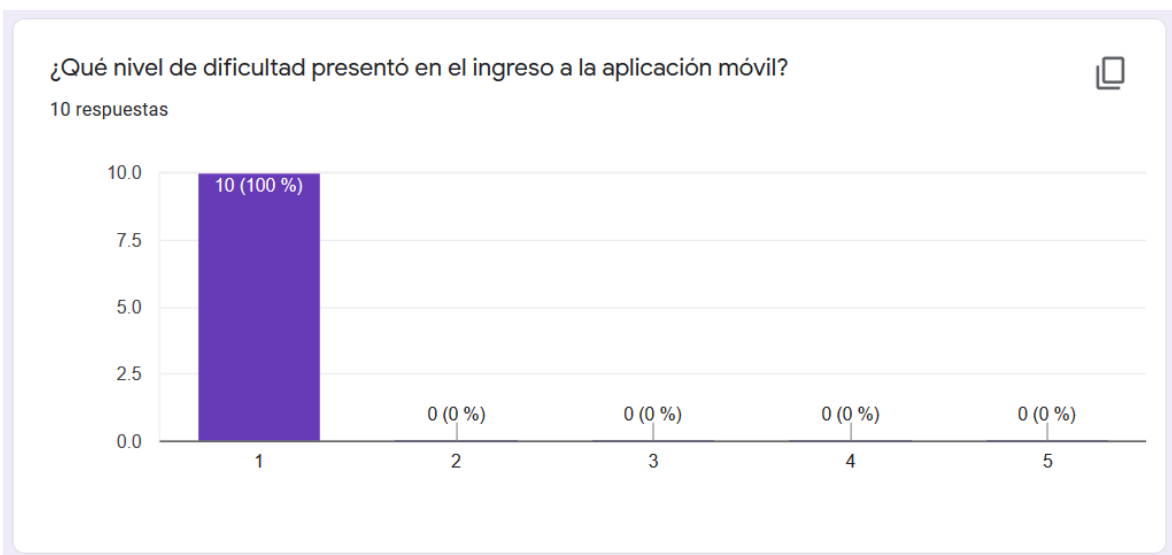
Para el cumplimiento de este Requerimiento No Funcional se realizó una encuesta de satisfacción a 10 usuarios que hicieron uso de la aplicación móvil y de 5 usuarios que hicieron uso de la aplicación web mediante la aplicación de “*Google Forms*” [47]; además, también se hace uso de esta aplicación para analizar los resultados de las encuestas, siendo algunos de los resultados los mostrados en las: Figura 3.27, Figura 3.28, Figura 3.29, Figura 3.30. En la escala de dificultad 1 - 5, “1” representa a un nivel fácil y “5” a un nivel difícil.



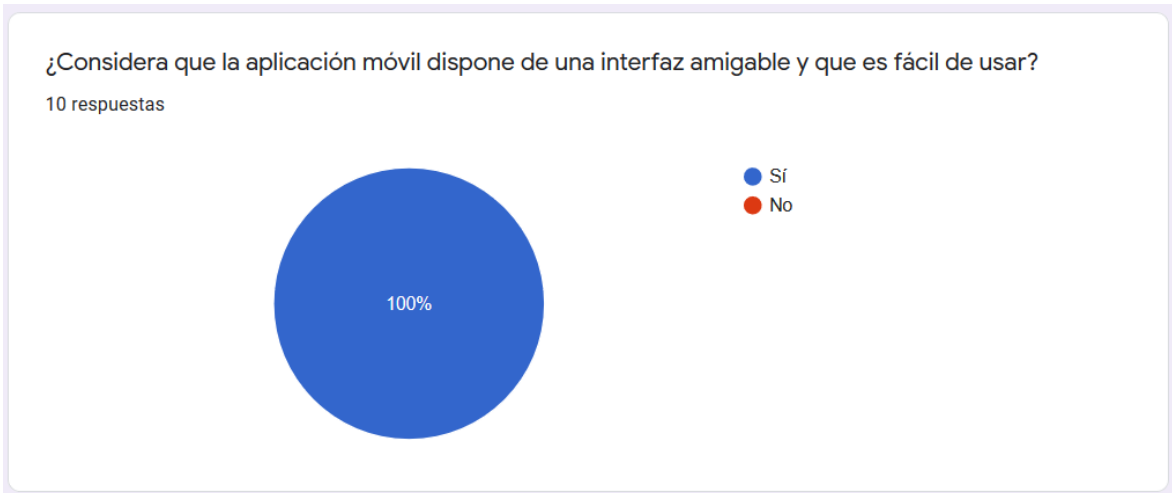
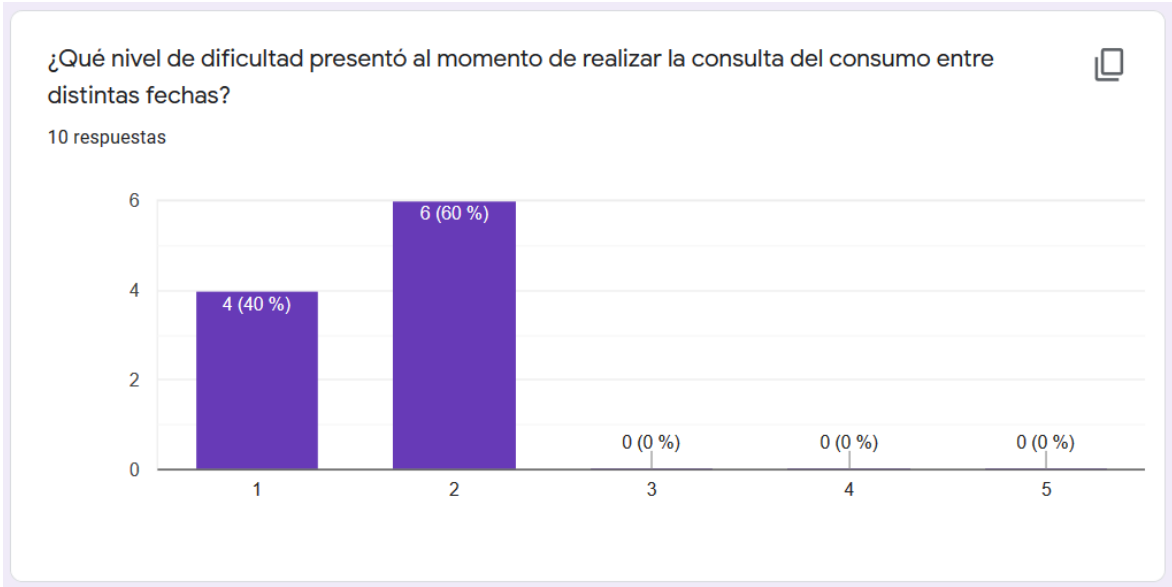
**Figura 3.27.** Módulo Web. Resultado de la dificultad de ingreso al Módulo Web.



**Figura 3.28.** Módulo Web. Resultado de la operabilidad.



**Figura 3.29.** Módulo Móvil. Resultado de la dificultad de ingreso al Módulo Móvil.



**Figura 3.30.** Módulo Móvil. Resultados de la operabilidad.

Como se puede observar en los resultados la dificultad del manejo de los Módulos Web y Móvil es muy bajo. Además, se observa que los dos módulos presentan una interfaz amigable y de fácil uso para los usuarios.

Las encuestas realizadas a los clientes y los resultados se muestran en el **ANEXO I**.

### 3.3.3. RNF03: ACCESIBILIDAD y RNF04: DISPONIBILIDAD

Estos requerimientos se orientan a que los servicios y la información se encuentre disponible desde cualquier lugar y en cualquier momento para diferentes usuarios.

Como el sistema se encuentra alojado en la nube de Microsoft Azure, se observa que en el contrato del servicio o *Service Level Agreement (SLA)* de la nube determina que: “*En el caso de los Servicios en la nube, garantizamos que cuando se implementen dos o más instancias de rol en diferentes Dominios de actualización, al menos una instancia de rol tendrá Conectividad de instancia de rol al menos el 99,95 % del tiempo.*” [48]. Por lo

anteriormente mencionado, se comprueba el cumplimiento de los Requerimientos No Funcionales correspondientes a Accesibilidad y Disponibilidad.

Una vez terminado de realizar la validación de los RNF, se presenta a continuación la **Tabla 3.7.** donde se presenta el cumplimiento de cada uno de los mismos.

**Tabla 3.7.** Pruebas de requerimientos no funcionales.

RNF	NOMBRE	RESULTADOS	ESTADO
RNF01	Autenticación	El Módulo Web proporciona una interfaz para el <i>login</i> del administrador del sistema.	OK
RNF02	Operabilidad	El resultado de la encuesta realizada a los usuarios, muestra que los Módulo Web y Móvil son amigables y sencillas de usar.	OK
RNF03	Acceso vía internet	Debido al acuerdo de servicio o ACL de Microsoft Azure se garantiza que el acceso a los servicios web y a la	OK
RNF04	Disponibilidad	Módulo Web se lo pueda realizar desde cualquier lugar. Además, también se garantiza se encuentren disponibles el 99,95% del tiempo.	OK

### 3.4. ACTUALIZACIÓN DEL TABLERO KANBAN

La **Figura 3.31** muestra el tablero Kanban para confirmar si se ha cumplido con todas las actividades establecidas para el desarrollo del prototipo.

Lista Tareas

En Proceso

Completadas

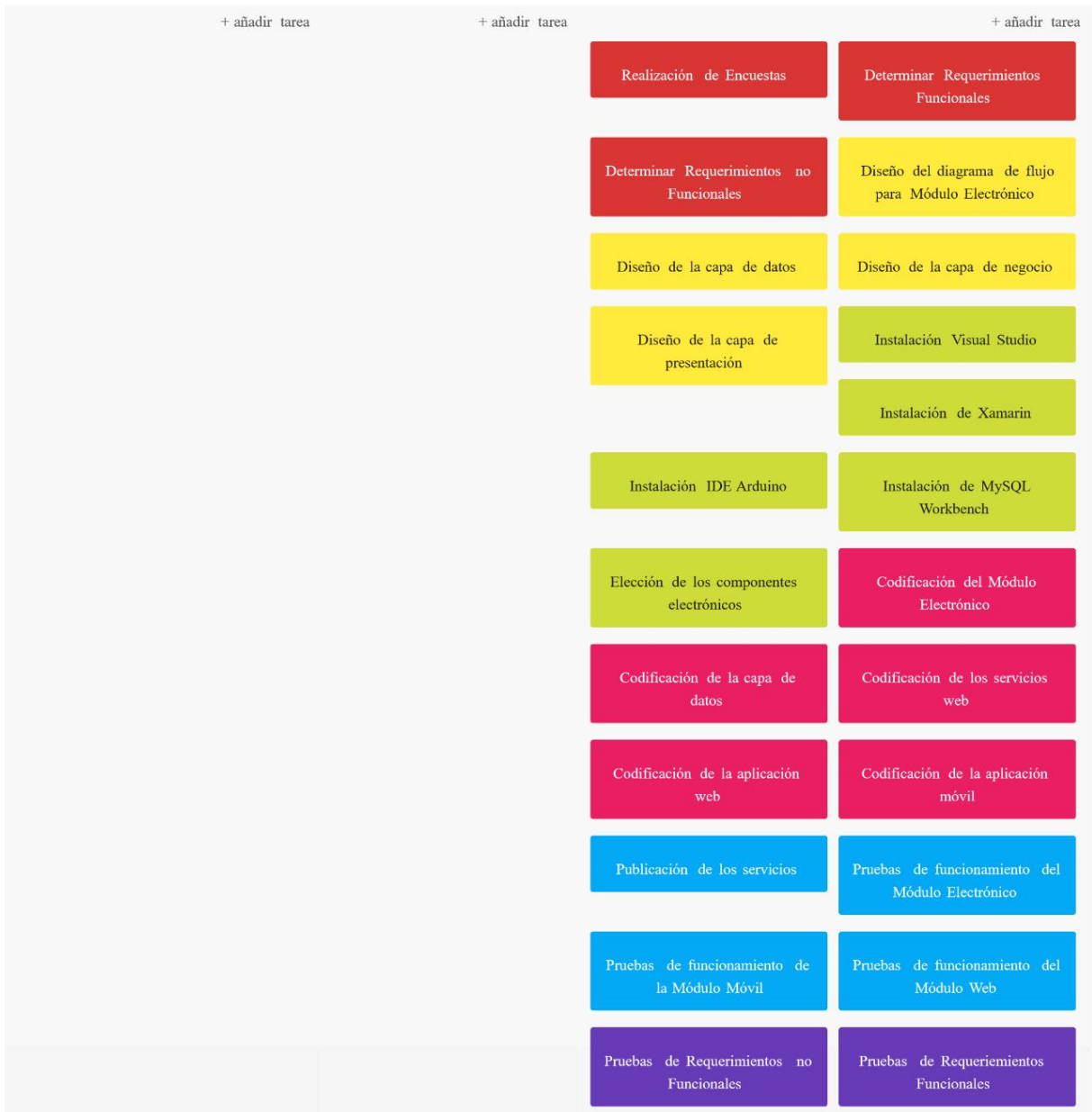


Figura 3.31. Tablero Kanban final.

## 4. CONCLUSIONES Y RECOMENDACIONES

### 4.1. CONCLUSIONES

- Este Trabajo de Titulación presenta un prototipo de un sistema opcional a la lectura del valor de consumo de agua. El prototipo consta de un Módulo Electrónico, Módulo Web y Módulo Móvil los cuales permiten realizar la lectura remota, la medición del valor de consumo de agua, la gestión del sistema y la ejecución de consultas de los valores de consumo de agua a un cliente de una manera sencilla.
- Las tecnologías, herramientas y lenguajes de programación que se utilizaron para el desarrollo del prototipo se las eligió en base a los conocimientos referentes al desarrollo de software como: Bases de Datos Relacionales, Programación Orientada a Objetos y Aplicaciones Distribuidas. Las herramientas seleccionadas fueron: C#, C, SQL como lenguajes de programación y MySQL Workbench, Visual Studio, Arduino IDE, Microsoft Azure como las herramientas y tecnologías para el desarrollo del prototipo. Todas estas herramientas presentan un entorno amigable para el desarrollador y permiten que el desarrollo de una aplicación sea más entendible.
- El uso de la metodología ágil Kanban para el desarrollo del prototipo fue de gran utilidad, debido a que se logró organizar el trabajo en distintas tareas, evitando que se desarrollen varias tareas al mismo tiempo y que estas queden inconclusas. Además, esta metodología permitió que se conozca el estado en el que se encontraba el desarrollo del prototipo en cualquier momento.
- El diseño de cada uno de los módulos se lo realizó en base a los Requerimientos Funcionales y No Funcionales que fueron adquiridos mediante encuestas realizadas a usuarios del actual sistema de medición de agua potable. Con estos requerimientos se procedió a realizar diagramas de flujo, diagramas de clases, diagramas de casos de uso, un diagrama UML para la base de datos, con el fin de satisfacer los requerimientos obtenidos de las encuestas.
- La implementación del prototipo se lo realizó siguiendo el diseño y el desarrollo por capas de software (Datos, Lógica y Presentación); lo cual, permite que el funcionamiento del prototipo se divida en tres distintas partes independientes, además, también permite que puedan existir distintos grupos de desarrolladores y que cada uno de estos equipos trabajen en una sola capa sin afectar a los otros.
- El Módulo Electrónico dispone de distintos tipos de elementos electrónicos y eléctricos con el fin de corregir posibles fallas como: la falta de comunicación para el almacenamiento del valor de consumo de agua en la base de datos, por lo cual,

este módulo posee una memoria microSD para que el valor de consumo sea registrado en ella y posteriormente sea ingresado en la base de datos. Además, posee una batería y placa electrónica que permiten que el módulo no deje de funcionar ante una posible falla de energía eléctrica.

- Los resultados adquiridos en la fase de pruebas de los Requerimientos Funcionales y No Funcionales del prototipo indicaron que el prototipo presenta un correcto funcionamiento. El módulo electrónico presenta un error en la medida de consumo, esto puede deberse a que el medidor mecánico actual presenta agujas analógicas para revisar el consumo realizado y por tal motivo se pudo haber cometido un error de apreciación del valor medido. Además, se estableció que las aplicaciones que se desarrollaron son amigables para el usuario, ya que presentan formularios claros para el ingreso de información, así como también, gráficos que facilitan al usuario el entendimiento de los datos que se muestran.

## 4.2. RECOMENDACIONES

- Es recomendable el uso de la metodología ágil Kanban para el desarrollo de este tipo proyectos, ya que esta metodología permite la gestión de las diferentes tareas necesarias para completar un proyecto. Además, de permitir que los involucrados en el desarrollo del proyecto conozcan las tareas que se encuentran pendientes, realizándose o terminadas.
- Es recomendable el desarrollo de un módulo que permita acoplarse al sistema actual y que este permita realizar la lectura de los valores de consumo remotamente de otros servicios básicos como el de energía eléctrica, debido a que en la actualidad se presentan problemas debido a errores cometidos en la lectura de estos valores o también que los valores no sean leídos perjudicando a los usuarios.
- Es recomendable el uso del *Cloud Computing* en este tipo de proyectos, ya que sus diferentes tipos de servicios permiten que se elija y pague solamente los recursos necesarios para el funcionamiento de cualquier aplicación; evitando que se requieran grandes inversiones para el desarrollo de nuevas aplicaciones.
- Las placas Arduino son recomendables para el desarrollo de los distintos tipos de proyectos que requieran automatizar algún proceso con un consumo bajo de energía. Además, estas placas en sus diferentes versiones poseen distintos tipos de tecnologías de comunicación como: WiFi, GSM, Ethernet, Bluetooth, etc., por lo cual, cualquier proyecto que se pueda desarrollar con este tipo de placas pueden formar parte del internet de las cosas o *Internet of Things (IoT)*.



## 5. REFERENCIAS BIBLIOGRÁFICAS

- [1] A. System, “La nueva revolución del mundo conectado”, 2018. [En línea]. Disponible en: <https://accent-systems.com/es/nb-iot/>. [Consultado: 17-dic-2018].
- [2] C. Rehbein, “Nueva tecnología para medidores de agua dará mayor información a los clientes.”, 2017. [En línea]. Disponible en: <https://www.publimetro.cl/cl/noticias/2017/02/15/nueva-tecnologia-medidores-agua-dara-mayor-informacion-clientes.html>. [Consultado: 15-nov-2018].
- [3] M. Quishpe, “Telemetria O Telectura Para Medidores De Agua Potable En El Centro Del Dmq”, INSTITUTO DE ALTOS ESTUDIOS NACIONALES, 2008.
- [4] BMETERS, “Dispositivi per reti LoRaWAN”, 2018.
- [5] iagua, “Lectura remota de contadores con la solución innovadora SensusRF”, 2017. [En línea]. Disponible en: <https://www.iagua.es/noticias/espana/sensus/17/10/19/lectura-remota-contadores-solucion-innovadora-sensusrf>. [Consultado: 06-sep-2019].
- [6] IBM, “Servicios Web”. [En línea]. Disponible en: [https://www.ibm.com/support/knowledgecenter/es/SS7K4U\\_9.0.0/com.ibm.websphere.zseries.doc/ae/cwbs\\_wbs2.html](https://www.ibm.com/support/knowledgecenter/es/SS7K4U_9.0.0/com.ibm.websphere.zseries.doc/ae/cwbs_wbs2.html). [Consultado: 22-abr-2019].
- [7] A. R. García Núñez, “Restful : un caso de uso de gestión de bibliotecas”, 2017. [En línea]. Disponible en: [http://oa.upm.es/45203/12/TFG\\_ADOLFO\\_RODRIGO\\_GARCIA\\_NUNEZ.pdf](http://oa.upm.es/45203/12/TFG_ADOLFO_RODRIGO_GARCIA_NUNEZ.pdf). [Consultado: 02-nov-2019].
- [8] JSON, “Introducción a JSON”. [En línea]. Disponible en: <https://www.json.org/json-es.html>. [Consultado: 22-abr-2019].
- [9] C. Mateu, *Desarrollo de aplicaciones web*. Barcelona: Fundació per a la Universitat Oberta de Catalunya.
- [10] J. Garrido Cobo, *TCF Desarrollo de Aplicaciones Móviles*, vol. 3, núm. 1. 2013.

- [11] RedHat, "Cloud Computing". [En línea]. Disponible en: <https://www.redhat.com/es/topics/cloud-computing/what-is-private-cloud>. [Consultado: 25-sep-2019].
- [12] "Cloud Computing - Aplicaciones en un solo tacto". [En línea]. Disponible en: <https://www.salesforce.com/mx/cloud-computing/>. [Consultado: 26-sep-2019].
- [13] D. G. IONOS, "IaaS", 2019. [En línea]. Disponible en: <https://www.ionos.es/digitalguide/servidores/know-how/que-es-iaas/>. [Consultado: 17-oct-2019].
- [14] D. G. IONOS, "PaaS", 2019. [En línea]. Disponible en: <https://www.ionos.es/digitalguide/servidores/know-how/paas/>. [Consultado: 17-oct-2019].
- [15] D. G. IONOS, "SaaS", 2019. [En línea]. Disponible en: <https://www.ionos.es/digitalguide/servidores/know-how/que-es-saas/>. [Consultado: 17-oct-2019].
- [16] A. B. Alonso, I. F. Artime, M. Á. Rodríguez, y R. G. Baniello, "Dispositivos móviles". [En línea]. Disponible en: [http://isa.uniovi.es/docencia/SIGC/pdf/telefonía\\_movil.pdf](http://isa.uniovi.es/docencia/SIGC/pdf/telefonía_movil.pdf). [Consultado: 08-may-2019].
- [17] S. de Comunicaciones, "Arquitectura Android". [En línea]. Disponible en: <https://sites.google.com/site/swcuc3m/home/android/generalidades/2-2-arquitectura-de-android>. [Consultado: 17-oct-2019].
- [18] S. de Comunicaciones, "iOS". [En línea]. Disponible en: <https://sites.google.com/site/swcuc3m/home/iphone/2-generalidades-1/2-2-arquitectura>. [Consultado: 17-oct-2019].
- [19] Softcorp, "Definición y cómo funcionan las aplicaciones móviles", 2019. [En línea]. Disponible en: <https://www.servisoftcorp.com/definicion-y-como-funcionan-las-aplicaciones-moviles/>. [Consultado: 07-sep-2019].
- [20] A. Alvarez Garcia y C. Lasa Gomez, *Métodos Ágiles: Scrum, Kanban, Lean*

(2ª Ed.). 2018.

- [21] M. J. Pérez, “Guía Comparativa de Metodologías Ágiles”, *Univ. Valladolid*, pp. 3–117, 2012.
- [22] R. J. Vargas y J. P. M. Granados, “Programación en Capas.”, Costa Rica, 2013.
- [23] A. Beaulieu, *Learning SQL*. United States: O'REILLY, 2009.
- [24] J. Segovia, “Diferencias entre DDL, DML y DCL”, *Todopostgresql*, 2017. [En línea]. Disponible en: <https://todopostgresql.com/diferencias-entre-ddl-dml-y-dcl/>. [Consultado: 01-oct-2019].
- [25] MySQL, “MySQL Workbench”. [En línea]. Disponible en: <https://www.mysql.com/products/workbench/>. [Consultado: 05-dic-2019].
- [26] Oracle, “MySQL Workbench Manual”. [En línea]. Disponible en: <https://dev.mysql.com/doc/workbench/en/wb-intro.html>. [Consultado: 16-sep-2019].
- [27] A. Aggarwal, “Introduction to .NET Framework”. [En línea]. Disponible en: <https://www.geeksforgeeks.org/introduction-to-net-framework/>. [Consultado: 16-sep-2019].
- [28] Microsoft, “Visual Studio”. [En línea]. Disponible en: <https://visualstudio.microsoft.com/es/vs/>. [Consultado: 02-dic-2019].
- [29] Microsoft, “Características de Visual Studio”, 2018. [En línea]. Disponible en: <https://docs.microsoft.com/es-es/visualstudio/ide/advanced-feature-overview?view=vs-2019>. [Consultado: 17-sep-2019].
- [30] Microsoft, “Introducción al lenguaje C# y .NET Framework”, 2015. [En línea]. Disponible en: <https://docs.microsoft.com/es-es/dotnet/csharp/getting-started/introduction-to-the-csharp-language-and-the-net-framework>. [Consultado: 03-nov-2019].
- [31] J. A. González Seco, “El lenguaje de programación C#”, 2001. [En línea]. Disponible en: [https://programacion.net/articulo/el\\_lenguaje\\_de\\_programacion\\_c\\_167/3](https://programacion.net/articulo/el_lenguaje_de_programacion_c_167/3).

- [Consultado: 03-nov-2019].
- [32] L. Álvarez *et al.*, “Programación Orientada a Objetos”. [En línea]. Disponible en: [https://ferestrepoca.github.io/paradigmas-de-programacion/poo/poo\\_teoría/concepts.html](https://ferestrepoca.github.io/paradigmas-de-programacion/poo/poo_teoría/concepts.html). [Consultado: 03-nov-2019].
- [33] Wikipedia, “ASP.NET”, 2019. [En línea]. Disponible en: <https://es.wikipedia.org/wiki/ASP.NET>. [Consultado: 17-sep-2019].
- [34] Windows, “What is ASP.NET?”, 2019. [En línea]. Disponible en: <https://dotnet.microsoft.com/learn/aspnet/what-is-aspnet>. [Consultado: 17-sep-2019].
- [35] “What is Xamarin?”, 2019. [En línea]. Disponible en: <https://dotnet.microsoft.com/learn/xamarin/what-is-xamarin>. [Consultado: 17-sep-2019].
- [36] Windows, “Cross-Platform Guidance”, 2019. [En línea]. Disponible en: <https://docs.microsoft.com/en-us/xamarin/cross-platform/get-started/introduction-to-mobile-development>. [Consultado: 17-sep-2019].
- [37] G. Factory, “WEMOS D1”. [En línea]. Disponible en: <https://www.geekfactory.mx/tienda/arduino/wemos-d1-esp8266/>. [Consultado: 17-sep-2019].
- [38] Arduino, “What is Arduino?” [En línea]. Disponible en: <https://www.arduino.cc/en/Guide/Introduction#>. [Consultado: 05-sep-2019].
- [39] W. Electronics, “D1”. [En línea]. Disponible en: <https://wiki.wemos.cc/products:d1:d1>. [Consultado: 17-sep-2019].
- [40] M. Naylamp, “Sensor de flujo de agua 1/2" YF-S201”. [En línea]. Disponible en: <https://naylampmechatronics.com/sensores-liquido/108-sensor-de-flujo-de-agua-12-yf-s201.html>. [Consultado: 18-sep-2019].
- [41] S. Alarm, “ST-Series Power Supplies / Chargers Fuentes de Alimentación / Cargadores Manual Features : Introduction : Parts List ”:
- [42] S. Labs, “KanbanTool”. [En línea]. Disponible en: <https://kanbantool.com/es/>. [Consultado: 13-nov-2019].

- [43] B. Studios, “Balsamiq”. [En línea]. Disponible en:  
<https://balsamiq.com/wireframes/>. [Consultado: 11-may-2020].
- [44] MySQL, “MySQL Workbench”. [En línea]. Disponible en:  
<https://www.mysql.com/products/workbench/>. [Consultado: 07-sep-2019].
- [45] Arduino, “Arduino - Software”. [En línea]. Disponible en:  
<https://www.arduino.cc/en/main/software>. [Consultado: 02-dic-2019].
- [46] E. Crespo, “IDE Arduino y Configuración”. [En línea]. Disponible en:  
<https://aprendiendoarduino.wordpress.com/tag/gestor-tarjetas/>. [Consultado: 27-ene-2020].
- [47] Google, “Crea atractivos formularios”. [En línea]. Disponible en:  
<https://www.google.com/intl/es-419/forms/about/>. [Consultado: 13-feb-2020].
- [48] Microsoft Azure, “SLA para Cloud Services”. [En línea]. Disponible en:  
[https://azure.microsoft.com/es-es/support/legal/sla/cloud-services/v1\\_5/](https://azure.microsoft.com/es-es/support/legal/sla/cloud-services/v1_5/).  
[Consultado: 13-feb-2020].
- [49] L. Castro, “¿Qué es un URI?”, 2016. [En línea]. Disponible en:  
<https://www.aboutspanol.com/que-es-uri-157626>. [Consultado: 18-sep-2019].

## **6. ANEXOS**

ANEXO A. Encuestas para obtención de requerimientos del sistema.

ANEXO B. Descripción de las clases del sistema.

ANEXO C. Código implementado en el módulo electrónico.

ANEXO D. Código implementado para la base de datos.

ANEXO E. Código implementado para los servicios web y envío de notificaciones.

ANEXO F. Código implementado para el desarrollo de la aplicación web.

ANEXO G. Código implementado para el desarrollo de la aplicación móvil.

ANEXO H. El registro de las medidas del medidor de agua actual y el Módulo Electrónico.

ANEXO I. Encuestas de satisfacción de la aplicación web y móvil.

## ANEXO A

### ENCUESTAS PARA OBTENCIÓN DE REQUERIMIENTOS DEL SISTEMA

Preguntas realizadas:

## ENCUESTA DE LOS REQUERIMIENTOS DEL SISTEMA DE MEDICIÓN DE CONSUMO DE AGUA.

Obtención de requerimientos.

**\*Obligatorio**

¿Dispone de una manera sencilla de conocer el valor de consumo de agua potable que ha realizado en cualquier fecha o fechas? \*

Sí

No

¿Dispone de algún dispositivo que realice la medición del consumo de agua a parte del proporcionado por la empresa encargada del servicio de agua potable, con el fin de corroborar si el valor marcado por el medidor de la empresa de agua potable es correcto? \*

Sí

No

¿Se le notifica de manera puntual el consumo de agua potable que ha realizado en cada mes? \*

Sí

No

¿Conoce usted alguna solución que permita realizar la medición del consumo de agua de manera remota? \*

- Sí
- No

¿Si se aplicaría una solución para la toma del valor de consumo de agua de manera remota permitiría que este haga uso de la red Wifi de su hogar para reducir valores en la implementación de esta solución? \*

- Sí
- No

¿Haría uso de una aplicación móvil en donde usted pueda verificar el valor del consumo de agua que ha realizado en distintas fechas de uno o varios medidores de agua? \*

- Sí
- No

¿Qué tipo de datos cree que debería presentar la aplicación móvil al usuario? \*

- El consumo de diferentes fecha o fechas.
- El consumo de diferentes meses.
- El último valor de consumo registrado.
- Todos los anteriores.



¿Qué tipos de consultas acerca del consumo de agua potable debería realizar la aplicación móvil? \*

- De una fecha en específico.
- Entre distintas fechas.
- Entre distintos meses y fechas.
- Todas las anteriores.

¿Se debería realizar el uso de distintos tipos de histogramas para representar los valores de consumos consultados, de tal forma que se mejore la visualización de los valores de mayor y menor consumo de agua potable? \*

- Sí
- No

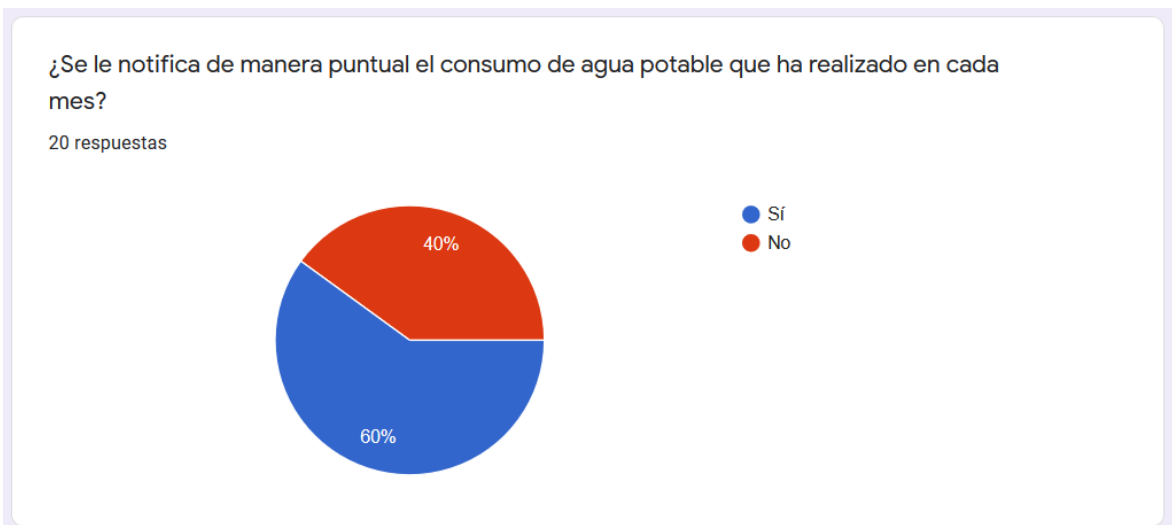
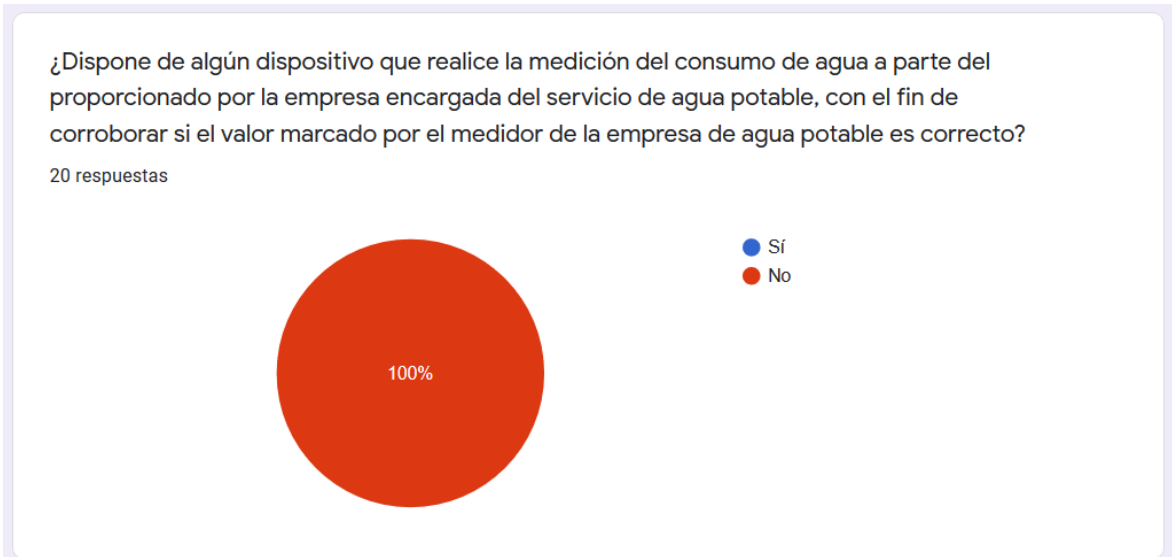
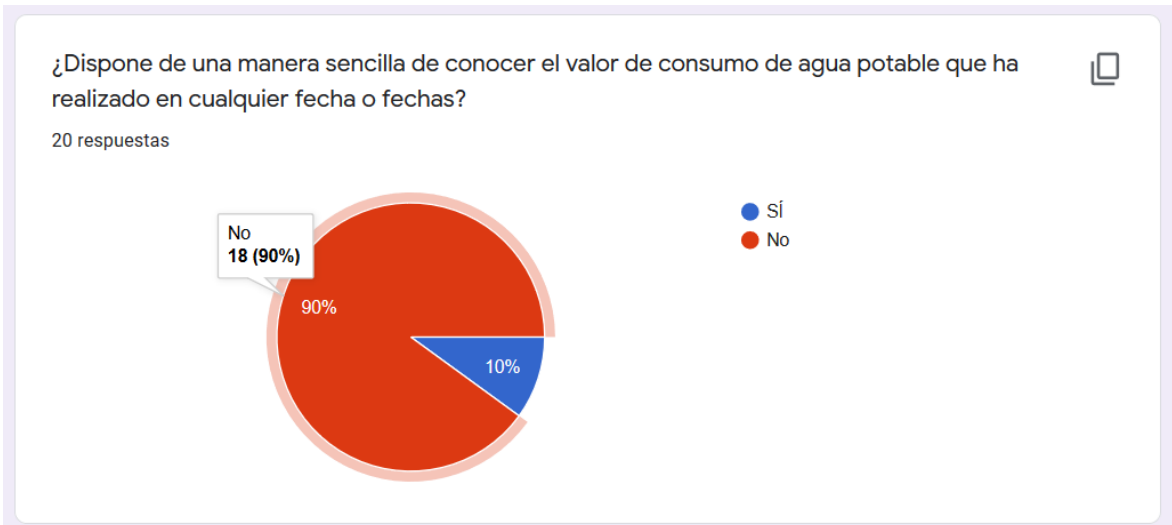
¿Está de acuerdo en que el cliente deba ser registrado previamente en el sistema, con el objetivo de identificar la pertenencia y la ubicación de cada nuevo medidor de agua? \*

- Sí
- No

¿Está de acuerdo en que se le asigne un número de serie único a cada medidor y que este sea usado para el ingreso a la aplicación móvil para que solo se pueda realizar consultas del consumo de ese medidor? \*

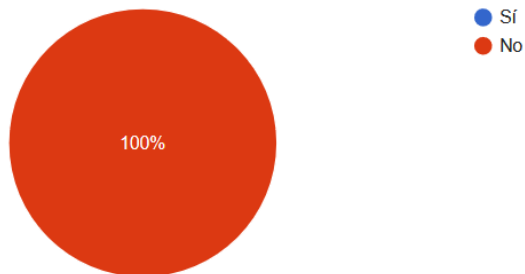
- Sí
- No

Resultados de las encuestas:



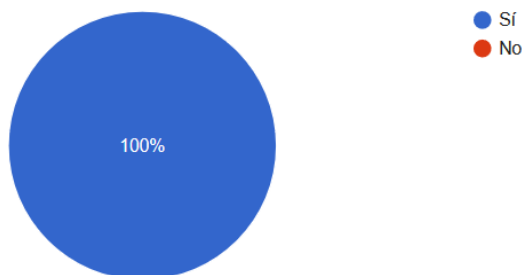
¿Conoce usted alguna solución que permita realizar la medición del consumo de agua de manera remota?

20 respuestas



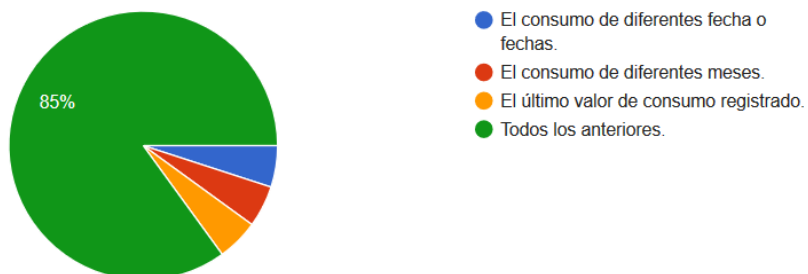
¿Si se aplicaría una solución para la toma del valor de consumo de agua de manera remota permitiría que este haga uso de la red Wifi de su hogar para reducir valores en la implementación de esta solución?

20 respuestas



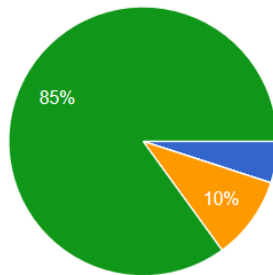
¿Qué tipo de datos cree que debería presentar la aplicación móvil al usuario?

20 respuestas



¿Qué tipos de consultas acerca del consumo de agua potable debería realizar la aplicación móvil?

20 respuestas



- De una fecha en específico.
- Entre distintas fechas.
- Entre distintos meses y fechas.
- Todas las anteriores.

¿Se debería realizar el uso de distintos tipos de histogramas para representar los valores de consumos consultados, de tal forma que se mejore la visualización de los valores de mayor y menor consumo de agua potable?

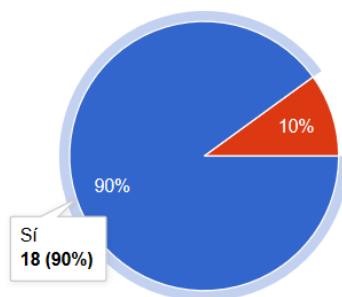
20 respuestas



- Sí
- No

¿Está de acuerdo en que el cliente deba ser registrado previamente en el sistema, con el objetivo de identificar la pertenencia y la ubicación de cada nuevo medidor de agua?

20 respuestas

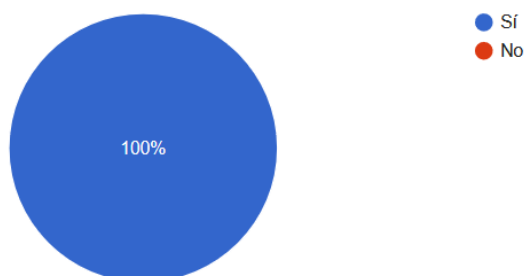


- Sí
- No

¿Está de acuerdo en que se le asigne un número de serie único a cada medidor y que este sea usado para el ingreso a la aplicación móvil para que solo se pueda realizar consultas del consumo de ese medidor?



20 respuestas



## ANEXO B

### DESCRIPCIÓN DE LAS ENTIDADES QUE INTERVIENEN EN EL DISEÑO DE LA CAPA DE DATOS

**Tabla 6.1.** Descripción de la entidad cliente.

<b>CAMPO</b>	<b>TIPO</b>	<b>DESCRIPCIÓN</b>
<b>idCliente</b>	Número entero	Campo usado como identificador único dentro del sistema del para cada cliente.
<b>Nombres</b>	Cadena de caracteres	Campo usado para almacenar los nombres del cliente.
<b>Apellidos</b>	Cadena de caracteres	Campo usado para almacenar los apellidos del cliente.
<b>Mail</b>	Cadena de caracteres	Campo usado para almacenar el correo electrónico del cliente, este correo es el que se usara para enviar las notificaciones.
<b>Cédula</b>	Número entero	Campo usado para almacenar el número de cedula del cliente.
<b>Teléfono</b>	Cadena de caracteres	Campo para almacenar un numero de contacto del cliente.

**Tabla 6.2.** Descripción de la entidad medida.

<b>CAMPO</b>	<b>TIPO</b>	<b>DESCRIPCIÓN</b>
<b>idMedida</b>	Número entero	Campo usado como identificador único dentro del sistema para cada medida realizada.
<b>idMedidor</b>	Número entero	Campo usado para determinar a qué medidor pertenece la medida ingresada.
<b>Consumo</b>	Cadena de caracteres	Campo usado almacenar el consumo realizado.
<b>fechaMedicion</b>	Cadena de caracteres	Campo usado determinar en qué fecha se ha realizado la medición.

**Tabla 6.3.** Descripción de la entidad administrador.

<b>CAMPO</b>	<b>TIPO</b>	<b>DESCRIPCIÓN</b>
<b>idAdministrador</b>	Número entero	Campo usado como identificador único dentro del sistema para cada administrador.
<b>Apellidos</b>	Cadena de caracteres	Campo usado para almacenar los nombres del administrador.
<b>Nombres</b>	Cadena de caracteres	Campo usado para almacenar los apellidos del administrador.
<b>userName</b>	Cadena de caracteres	Campo usado como identificador para el <i>login</i> del administrador.
<b>Password</b>	Cadena de caracteres	Campo usado para determinar la contraseña del administrador, este campo junto con el userName será usados para el <i>login</i> del administrador.

## ANEXO B

### DESCRIPCIÓN DE LAS CLASES DEL SISTEMA

Tabla 6.4. Descripción de los atributos de la clase Persona.

ATRIBUTO	TIPO DE DATO	TIPO DE ACCESO	DESCRIPCIÓN
nombres	Cadena de caracteres.	Privado (-)	Atributo designado para los dos nombres de la persona.
apellidos	Cadena de caracteres.	Privado (-)	Atributo designado para los dos apellidos de la persona.

Tabla 6.5. Descripción de los atributos de la clase Cliente.

ATRIBUTO	TIPO DE DATO	TIPO DE ACCESO	DESCRIPCIÓN
idCliente	Número entero	Privado (-)	Atributo designado para el identificador único de cada cliente dentro del sistema.
mail	Cadena de caracteres.	Privado (-)	Atributo designado para la dirección de correo electrónico del cliente.
Cédula	Cadena de caracteres.	Privado (-)	Atributo designado para el número de cédula del cliente.
teléfono	Cadena de caracteres.	Privado (-)	Atributo designado para el número telefónico del cliente.

Tabla 6.6. Descripción de los atributos de la clase Administrador.

ATRIBUTO	TIPO DE DATO	TIPO DE ACCESO	DESCRIPCIÓN
idAdministrador	Número entero	Privado (-)	Atributo designado para el identificador único de cada administrador dentro del sistema.
userName	Cadena de caracteres.	Privado (-)	Atributo designado para una credencial del administrador para el <i>login</i> .
password	Cadena de caracteres.	Privado (-)	Atributo designado para la contraseña del administrador usada en el <i>login</i> .



**Tabla 6.7.** Descripción de los atributos de la clase Medida.

<b>ATRIBUTO</b>	<b>TIPO DE DATO</b>	<b>TIPO DE ACCESO</b>	<b>DESCRIPCIÓN</b>
<b>idMedida</b>	Número entero	Privado (-)	Atributo designado para el identificador único de cada medida dentro del sistema.
<b>consumo</b>	Número entero	Privado (-)	Atributo designado para el valor de consumo de agua.
<b>fechaMedicion</b>	Fecha y hora.	Privado (-)	Atributo designado para la fecha en que se ha realizado el ingreso del consumo de agua.

## ANEXO C

### CÓDIGO IMPLEMENTADO EN EL MÓDULO ELECTRÓNICO

```
//-----Declaracion de Librerias-----.  
#include <SD.h>  
#include <ESP8266WiFi.h> //Libreria para la conexion WiFi  
#include <WiFiUdp.h>  
#include <NTPClient.h>  
#include <SPI.h>  
#include <String.h>  
#include <Dns.h>  
  
//-----Definicion de variables para trabajo con la MicroSD-----  
File archivoMedidas;  
  
//-----Definicion de variables para el ingreso de datos.-----  
char fecha[126];  
int ingresoDiarioDB = 0;  
int ingresoDiarioSD = 0;  
int idMedidor = 1;  
int horaIngreso = 18;  
  
//-----Declaracion de variable para conexion WiFi.-----  
char ssid[] = "SEBASTIAN BAHAMONDE"; //SSDI de la red WiFi  
char password[] = "1722706452"; //Password de la red WiFi  
WiFiClient client;  
  
//-----Declaracion de variables para obtener la hora y fecha.-----  
WiFiUDP ntpUDP; //Instancia para de la libreria WiFiUDP.  
//Instancia de la NTPClient (bufferUDP,servidor ntp,offset,tiempo de actualizacion)  
NTPClient timeClient(ntpUDP, "ec.pool.ntp.org", -18000, 60000);  
  
//-----Declaracion de variable para el servicio web-----  
const char* host = "apimedicion20191002124840.azurewebsites.net";  
  
//-----Variables para el sensor de flujo YF-S201-----  
volatile int NumPulsos; //variable para la cantidad de pulsos recibidos  
int PinSensor = 2; //Sensor conectado en el pin 2  
float factor_conversion = 7.5; //para convertir de frecuencia a caudal
```

```

float volumen = 0;
long dt = 0; //variación de tiempo por cada bucle
long t0 = 0; //millis() del bucle anterior

void setup()
{
  Serial.begin(9600);

  //Configuración de I/O para el flujometro
  pinMode(PinSensor, INPUT);
  attachInterrupt(0, ContarPulsos, RISING); //(Interrupción 0(Pin2),función,Flanco de
subida)
  t0 = millis();
  //Conexion a la red WiFi
  conectarAWifi();
  //Inicialización de NTP.
  timeClient.begin();
  while (!timeClient.update())
  {
    timeClient.update();
  }
  if (!SD.begin(4))
  {
    Serial.println("No se pudo iniciar el SD");
  }
  Serial.println("Iniciado el SD");
}
//-----Lazo de repeticion general.-----
void loop()
{
  int hora = timeClient.getHours();
  Serial.print("Hora: ");

```

```

Serial.println(hora);
Serial.print("Valor Consumido: ");
Serial.print(valorConsumo());
Serial.println(" L");
if (hora == horaIngreso)
{
    ingresoDiarioSD = 0;
    if (ingresoDiarioDB == 0)
    {
        Serial.println("Hora de ingreso.");
        int i = insertDB(idMedidor, valorConsumo(), obtenerFecha());
        Serial.println(i);
        if (i == 1)
        {
            Serial.println("Solo debo repetirme una vez.");
            ingresoDiarioDB = 1;
            lecturaSD();
            i = 0;
        }
    }
}
if (ingresoDiarioDB == 0 && hora != horaIngreso && ingresoDiarioSD == 0)
{
    char consumo[128];
    obtenerFecha().toCharArray(fecha, obtenerFecha().length() + 1); //Paso de String a
Char, se grabara en la variable fecha.
    sprintf(consumo, "%s:%d,", fecha, valorConsumo());
    insertarEnSD(consumo);
    ingresoDiarioSD = 1;
}
if (hora != horaIngreso)
{
    ingresoDiarioDB = 0;

```

```

}
delay(5000);
}
//-----Funcion a ejecutarse en la interrupcion.-----
void ContarPulsos()
{
  NumPulsos++; //Se incrementa los pulsos
}

//-----Funcion para la obtencion del valor-----
int valorConsumo()
{
  float frecuencia = ObtenerFrecuecia(); //obtenemos la frecuencia de los pulsos en Hz
  float caudal_L_m = frecuencia / factor_conversion; //calculamos el caudal en L/m
  dt = millis() - t0; //calculamos la variación de tiempo
  t0 = millis();
  volumen = volumen + (caudal_L_m / 60) * (dt / 1000); // volumen(L)=caudal(L/s)*tiempo(s)
  return volumen;
}

//-----Función para obtener frecuencia de los pulsos-----
int ObtenerFrecuecia()
{
  int frecuencia;
  NumPulsos = 0; //Ponemos a 0 el número de pulsos
  interrupts(); //Habilitamos las interrupciones
  delay(1000); //muestra de 1 segundo
  noInterrupts(); //Deshabilitamos las interrupciones
  frecuencia = NumPulsos; //Hz(pulsos por segundo)
  return frecuencia;
}

```

```

//-----Funcion para insertar en la Base de Datos-----
int insertDB(int idMedidor, int valor, String Fecha)
{
    Fecha.toCharArray(fecha, Fecha.length() + 1); //Paso de String a Char, se grabara en la
variable fecha.
    Serial.print("connecting to ");
    Serial.println(host);
    // Uso WiFiClient para crear la conexion TCP.
    WiFiClient client;
    const int httpPort = 80;
    if (!client.connect(host, httpPort)) {
        Serial.println("Conexion Fallida");
        return 0;
    }
    Serial.println("Conexion exitosa");
    char peticionUrl[] = "/api/Login/ingresarConsumo?date=%s&valor=%d&idMedidor=%d";
    char peticion[256];
    sprintf(peticion, peticionUrl, fecha, valor, idMedidor);
    // We now create a URI for the request
    String url = peticion;
    Serial.print("Requesting URL: ");
    Serial.println(url);
    // Petincion al servidor web.
    client.print(String("GET ") + url + " HTTP/1.1\r\n" +
        "Host: " + host + "\r\n" +
        "Connection: close\r\n\r\n");
    delay(10000);
    // Leo la respuesta del servidor web a mi peticion.
    Serial.println("Respuesta:");
    while (client.available()) {
        String linea = client.readStringUntil('\r');
        Serial.print(linea);
    }
}

```

```

Serial.println("Cierro la conexión.");
return 1;
}

//-----Funcion para obtener Fecha-----
String obtenerFecha()
{
String formattedDate = timeClient.getFormattedDate();
int splitT = formattedDate.indexOf("T");
String dayStamp = formattedDate.substring(0, splitT);
return dayStamp;
}

//-----Funcion para obtener Hora-----
String obtenerHora()
{
String formattedTime = timeClient.getFormattedTime();
int splitT = formattedTime.indexOf("T");
String timeStamp = formattedTime.substring(splitT + 1, formattedTime.length() - 1);
return timeStamp;
}

//-----Funcion para el ingreso en la SD-----
int insertarEnSD(String cadena)
{
archivoMedidas = SD.open("consumo.txt", FILE_WRITE);
if (archivoMedidas)
{
archivoMedidas.print(cadena);
archivoMedidas.close();
Serial.println("Ingreso en SD exitoso.");
return 1;
} else

```

```

{
    Serial.println("Error al abrir el archivo.");
    return 0;
}
}
//-----Funcion para la lectura de la SD-----
void lecturaSD()
{
    archivoMedidas = SD.open("consumo.txt");
    if (archivoMedidas)
    {
        String valor;
        while (archivoMedidas.available())
        {
            valor = archivoMedidas.readString();
        }
        char *strtok(char *str1, const char *str2);
        char delimitador[] = ",";

        char *resultado = NULL; //Variable a guardar fecha y valor de consumo yyyy-mm-dd
        char result[valor.length()]; // Variable para guardar como char lo leido de la SD.
        valor.toCharArray(result, valor.length() + 1); //Paso de String a Char, se grabara en la
        variable result.
        resultado = strtok(result, delimitador); // Obtengo el primer dato a registrarse. yyyy-mm-
        dd:consumo
        //Recorro todos los valores.
        while ( resultado != NULL)
        {
            resultado = strtok(NULL, delimitador);
            //Ingreso de la funcion para el ingreso del valor de datos.
            Serial.println(resultado);
            ingresarValoresAtrasados(resultado);
        }
    }
}

```



```

    archivoMedidas.close();
    SD.remove("consumo.txt");
} else
{
    Serial.println("No existe el archivo consumo.txt");
}
}
//-----Funcion para ingresar valores atrasados.-----
int ingresarValoresAtrasados(char fechaConsumo[])
{
    int i = 0;
    String datoProcesar = String(fechaConsumo);
    if (datoProcesar != NULL)
    {
        Serial.println("Ingreso de los valores a la base de datos.");
        String date = datoProcesar.substring(0, 10);
        int consumo = (datoProcesar.substring(11, datoProcesar.length())).toInt();
        i = insertDB(idMedidor, consumo, date);
    }
}
}

```

## ANEXO D

### CÓDIGO IMPLEMENTADO EN LA BASE DE DATOS

```
--  
-- Estructura de tabla para la tabla `administrador`  
--  
CREATE TABLE `administrador` (  
  `idAdministrador` int(11) NOT NULL,  
  `apellidos` varchar(45) NOT NULL,  
  `nombres` varchar(45) NOT NULL,  
  `userName` varchar(45) NOT NULL,  
  `password` varchar(45) NOT NULL  
)  
--  
-- Indices de la tabla `administrador`  
--  
ALTER TABLE `administrador`  
  ADD PRIMARY KEY (`idAdministrador`);  
--  
-- AUTO_INCREMENT de la tabla `administrador`  
--  
ALTER TABLE `administrador`  
  MODIFY `idAdministrador` int(11) NOT NULL AUTO_INCREMENT;  
  
-----  
  
--  
-- Estructura de tabla para la tabla `cliente`  
--  
CREATE TABLE `cliente` (  
  `idCliente` int(11) NOT NULL,  
  `nombres` varchar(45) NOT NULL,  
  `apellidos` varchar(45) NOT NULL,
```

```

`mail` varchar(45) NOT NULL,
`cedula` varchar(45) NOT NULL,
`telefono` varchar(45) NOT NULL
)
--
-- Indices de la tabla `cliente`
--
ALTER TABLE `cliente`
  ADD PRIMARY KEY (`idCliente`);
--
-- AUTO_INCREMENT de la tabla `cliente`
--
ALTER TABLE `cliente`
  MODIFY `idCliente` int(11) NOT NULL AUTO_INCREMENT;

-----

--
-- Estructura de tabla para la tabla `medida`
--
CREATE TABLE `medida` (
  `idMedida` int(11) NOT NULL,
  `idMedidor` int(11) NOT NULL,
  `consumo` int(11) NOT NULL,
  `fechaMedicion` date NOT NULL
)
--
-- Indices de la tabla `medida`
--
ALTER TABLE `medida`
  ADD PRIMARY KEY (`idMedida`),
  ADD KEY `idMedidor` (`idMedidor`);
--

```

```

-- AUTO_INCREMENT de la tabla `medida`
--
ALTER TABLE `medida`
  MODIFY `idMedida` int(11) NOT NULL AUTO_INCREMENT;
--
-- Clave foranea para la tabla `medida`
--
ALTER TABLE `medida`
  ADD CONSTRAINT `medida_ibfk_1` FOREIGN KEY (`idMedidor`) REFERENCES
`medidor` (`idMedidor`);

-----

--
-- Estructura de tabla para la tabla `medidor`
--
CREATE TABLE `medidor` (
  `idMedidor` int(11) NOT NULL,
  `direccion` varchar(45) NOT NULL,
  `sector` varchar(45) NOT NULL,
  `serieMedidor` varchar(45) NOT NULL,
  `idCliente` int(11) NOT NULL
)
--
-- Indices de la tabla `medidor`
--
ALTER TABLE `medidor`
  ADD PRIMARY KEY (`idMedidor`),
  ADD KEY `idCliente` (`idCliente`);
--
-- AUTO_INCREMENT de la tabla `medidor`
--

```

```
ALTER TABLE `medidor`  
  MODIFY `idMedidor` int(11) NOT NULL AUTO_INCREMENT;  
--  
-- Clave foranea para la tabla `medidor`  
--  
ALTER TABLE `medidor`  
  ADD CONSTRAINT `medidor_ibfk_1` FOREIGN KEY (`idCliente`) REFERENCES  
  `cliente` (`idCliente`);  
COMMIT;
```

## **ANEXO E**

### **CÓDIGO IMPLEMENTADO PARA LOS SERVICIOS WEB Y ENVÍO DE NOTIFICACIONES**

El código implementado de los servicios web se encuentra en el CD adjunto.

## **ANEXO F**

### **CÓDIGO IMPLEMENTADO PARA EL DESARROLLO DEL MÓDULO WEB**

El código implementado de la aplicación web se encuentra en el CD adjunto.

## **ANEXO G**

### **CÓDIGO IMPLEMENTADO PARA EL DESARROLLO DEL MÓDULO MÓVIL**





El código implementado de la aplicación móvil se encuentra en el CD adjunto.



## ANEXO H

### REGISTRO DE LAS MEDIDAS DEL MEDIDOR DE AGUA ACTUAL Y EL MÓDULO ELECTRÓNICO

Las pruebas de comparación de los valores medidos por el medidor de agua actual y el Módulo Electrónico se lo realizó haciendo pasar un litro de agua por el medidor de agua actual y observando el valor medidor por el Módulo Electrónico.

Cant. (L)	Medidor	Cant. (L)	Módulo Electrónico	Cant. (L)
1	Antes: 	1.2	Antes: Valor Consumido: 0.0 L	1.1
	Después: 		Después: Valor Consumido: 1.1 L	
2	Antes: 	1	Antes: Valor Consumido: 1.1 L	1.3
	Después: 		Después: Valor Consumido: 2.4 L	

23	Antes:	1	Antes: Valor Consumido: 2.4 L	0.8
	Después:		Después: Valor Consumido: 3.2 L	
4	Antes:	1	Antes: Valor Consumido: 3.2 L	0.9
	Después:		Después: Valor Consumido: 4.1 L	

5	Antes:	1	Antes: Valor Consumido: 4.1 L	0.8
	Después:		Después: Valor Consumido: 4.9 L	
6	Antes:	1	Antes: Valor Consumido: 4.9 L	0.9
	Después:		Después: Valor Consumido: 5.8 L	

7	Antes:	1	Antes: Valor Consumido: 5.8 L	1.1
	Después:		Después: Valor Consumido: 6.9 L	
8	Antes:	1	Antes: Valor Consumido: 6.9 L	0.9
	Después:		Después: Valor Consumido: 7.8 L	

9	Antes:	1	Antes: Valor Consumido: 7.8 L	1.2
	Después:		Después: Valor Consumido: 9.0 L	
10	Antes:	1	Antes: Valor Consumido: 9.0 L	1.1
	Después:		Después: Valor Consumido: 10.1 L	

# ANEXO I

## ENCUESTAS DE SATISFACCIÓN DE LOS MÓDULOS WEB Y MÓVIL

Preguntas para el Módulo Web:

### SISTEMA DE MEDICIÓN DE CONSUMO DE AGUA

ENCUESTA DE SATISFACCIÓN DE LA APLICACIÓN WEB

\*Obligatorio

¿Qué dificultad presento al momento de ingresar a la aplicación? \*

	1	2	3	4	5	
Fácil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Difícil

¿Qué nivel de dificultad tuvo para realizar la búsqueda de clientes? \*

	1	2	3	4	5	
Fácil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Difícil

¿Qué nivel de dificultad tuvo para realizar la búsqueda del consumo de un día? \*

	1	2	3	4	5	
Fácil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Difícil

¿Qué nivel de dificultad tuvo para realizar la consulta del consumo por meses? \*

	1	2	3	4	5	
Fácil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Difícil

¿Qué nivel de dificultad tuvo para realizar la consulta del consumo por fechas? \*

	1	2	3	4	5	
Fácil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Difícil

¿Qué nivel de dificultad tuvo para realizar el ingreso de nuevos usuarios? \*

	1	2	3	4	5	
Fácil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Difícil

¿Qué nivel de dificultad tuvo para realizar la administración de los usuarios y sus medidores? \*

	1	2	3	4	5	
Fácil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Difícil

¿Qué nivel de dificultad tuvo para realizar la administración de Administradores?

\*

Fácil      1      2      3      4      5      Dificil

¿Considera que la aplicación web presenta una interfaz amigable y de facil uso?

Sí

No

Recomendaciones:

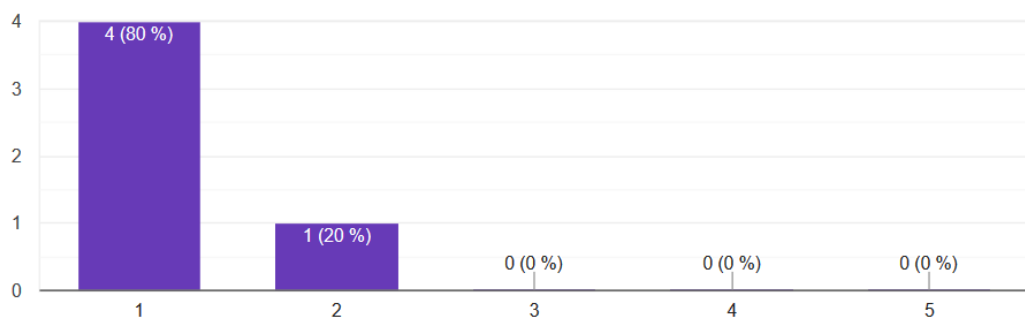
Tu respuesta

---

Resultados:

¿Qué dificultad presento al momento de ingresar a la aplicación?

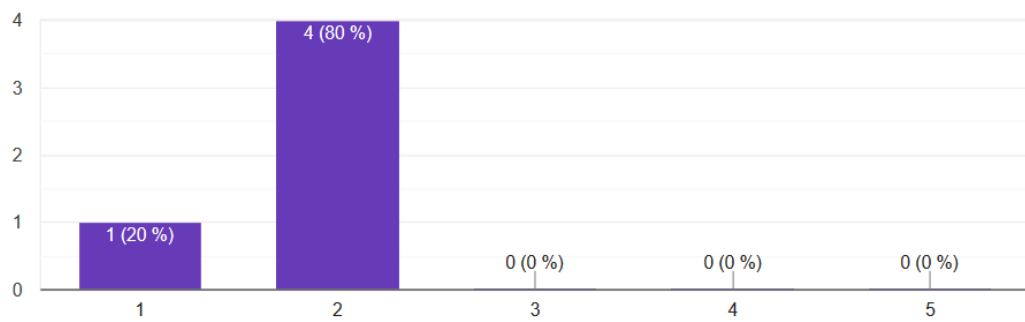
5 respuestas





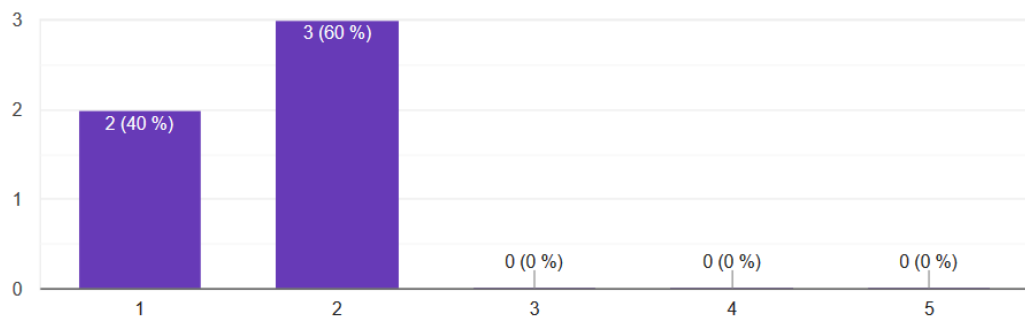
¿Qué nivel de dificultad tuvo para realizar la búsqueda de clientes?

5 respuestas



¿Qué nivel de dificultad tuvo para realizar la búsqueda del consumo de un día?

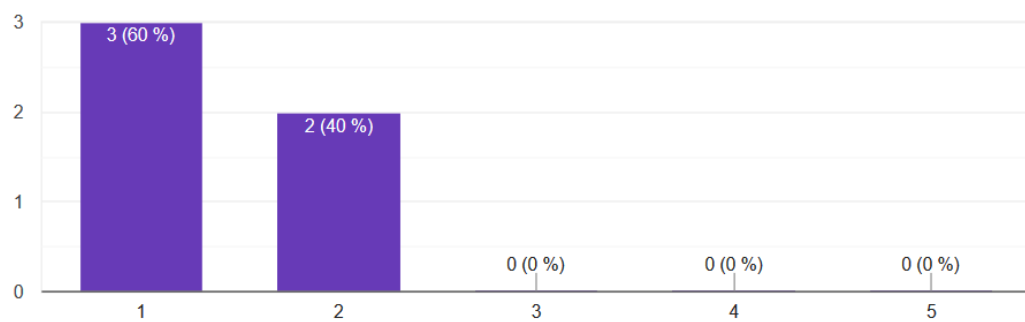
5 respuestas



¿Qué nivel de dificultad tuvo para realizar la consulta del consumo por meses?

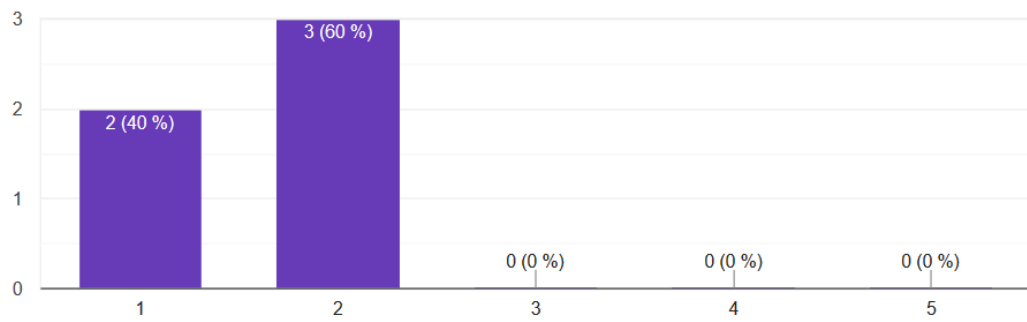


5 respuestas



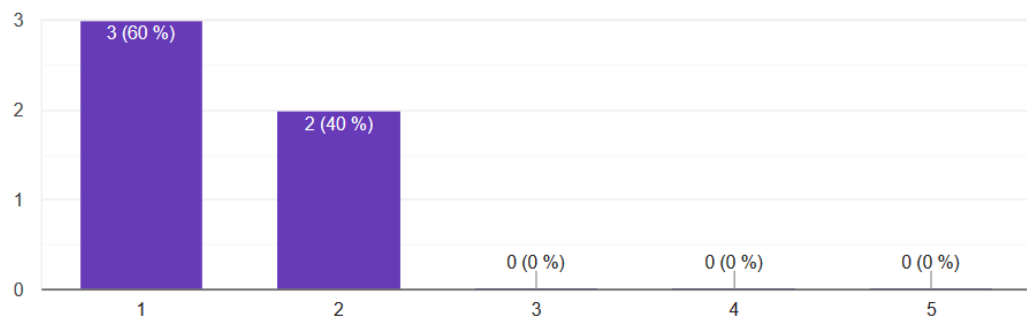
¿Qué nivel de dificultad tuvo para realizar la consulta del consumo por fechas?

5 respuestas



¿Qué nivel de dificultad tuvo para realizar el ingreso de nuevos usuarios?

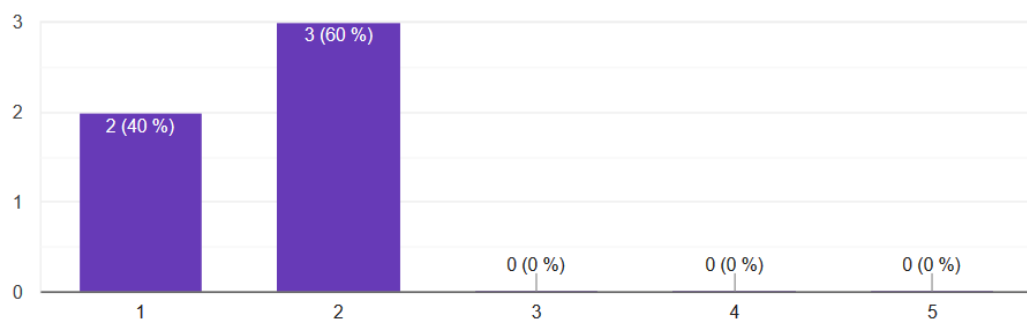
5 respuestas



¿Qué nivel de dificultad tuvo para realizar la administración de los usuarios y sus medidores?

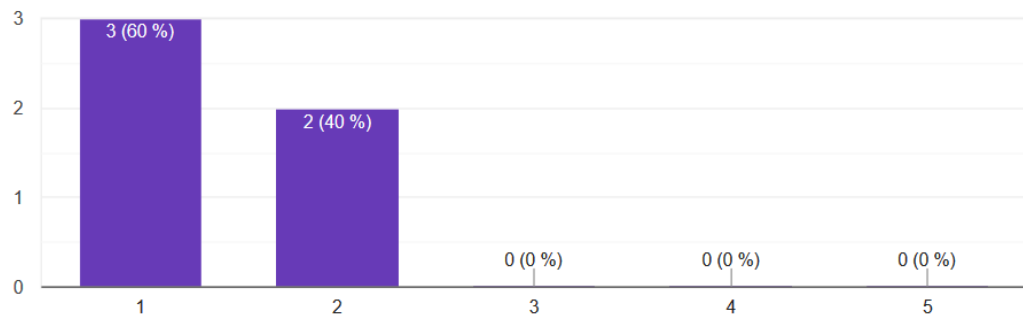


5 respuestas



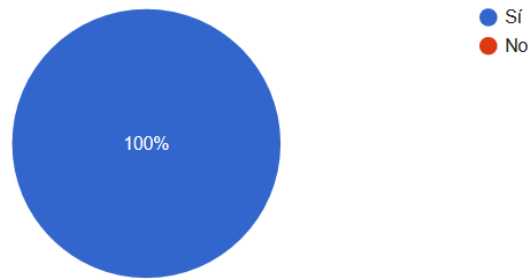
¿Qué nivel de dificultad tuvo para realizar la administración de Administradores?

5 respuestas



¿Considera que la aplicación web presenta una interfaz amigable y de fácil uso?

5 respuestas



Preguntas para el Módulo Móvil:

# SISTEMA DE MEDICIÓN DE CONSUMO DE AGUA

ENCUESTA DE SATISFACCIÓN DE LA APLICACIÓN MÓVIL

\*Obligatorio

¿Qué nivel de dificultad presentó en el ingreso a la aplicación móvil? \*

1

2

3

4

5

2. ¿Cree usted que la información presentada en la pantalla principal le es útil? \*

Sí

No

¿Qué nivel de dificultad presentó al momento de realizar la búsqueda del consumo de una fecha específica? \*

1

2

3

4

5

Fácil

Difícil

¿Qué nivel de dificultad presentó al momento de realizar la consulta del consumo por meses? \*

	1	2	3	4	5	
Fácil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Difícil

¿Qué nivel de dificultad presentó al momento de realizar la consulta del consumo entre distintas fechas? \*

	1	2	3	4	5	
Fácil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Difícil

¿El uso de los diferentes histogramas para observar los valores de consumo de la consulta de consumo por fechas le fué fácil usar? \*

- Sí
- No

¿Considera que la aplicación móvil dispone de una interfaz amigable y que es fácil de usar? \*

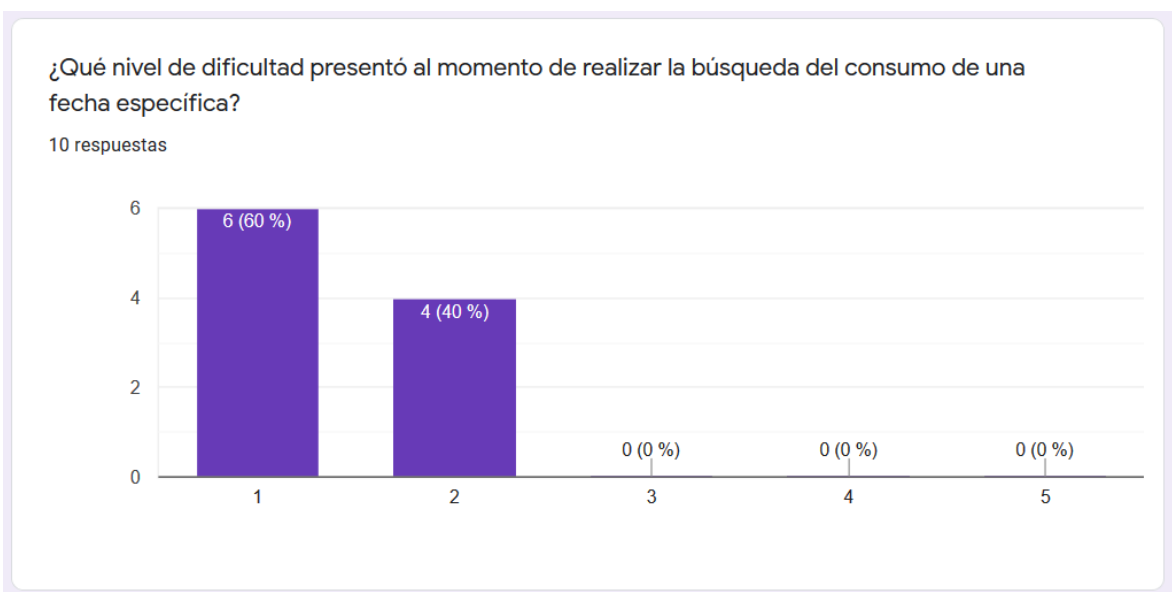
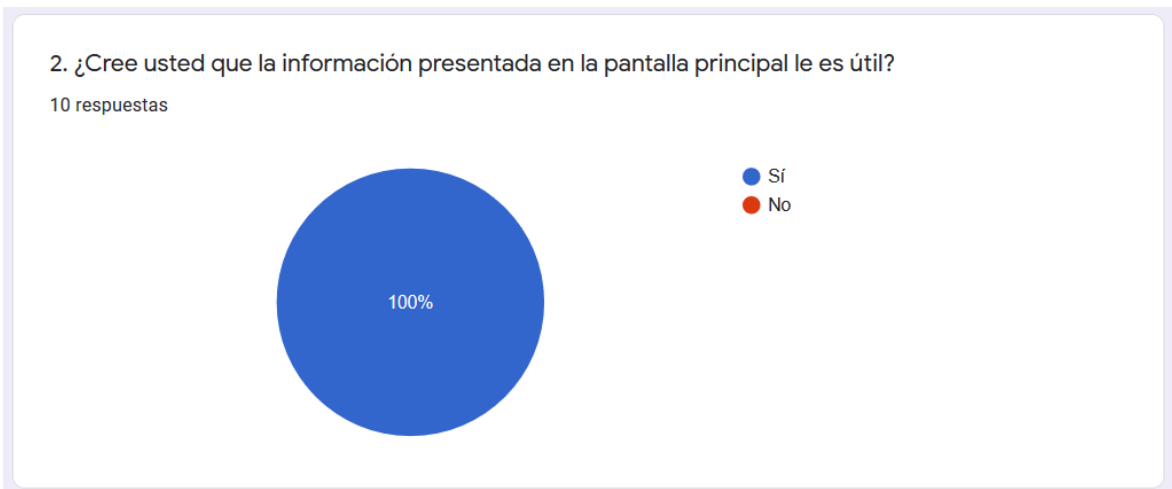
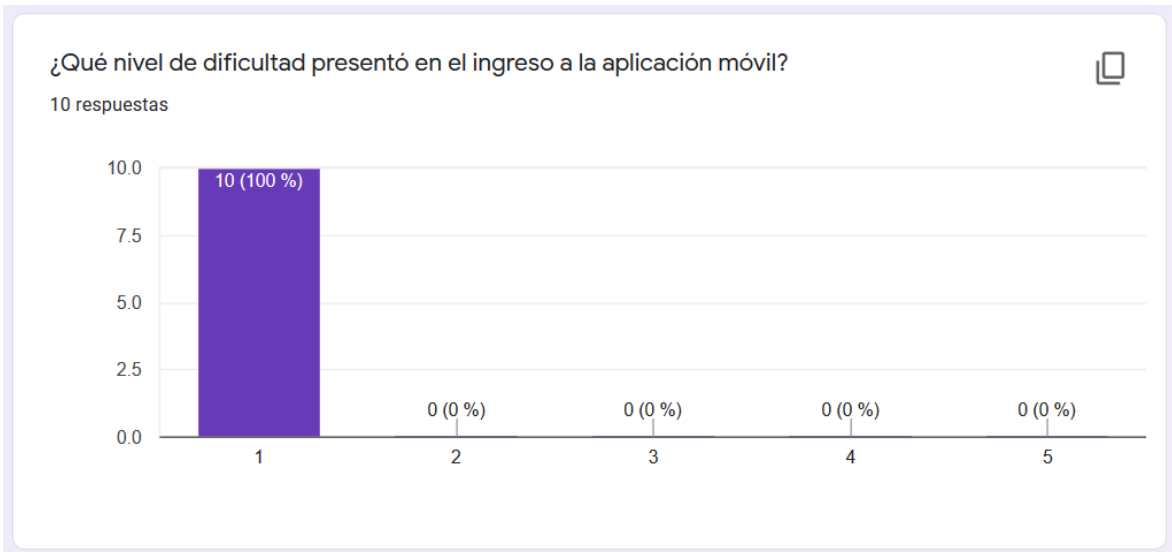
- Sí
- No

Recomendaciones:

Tu respuesta

---

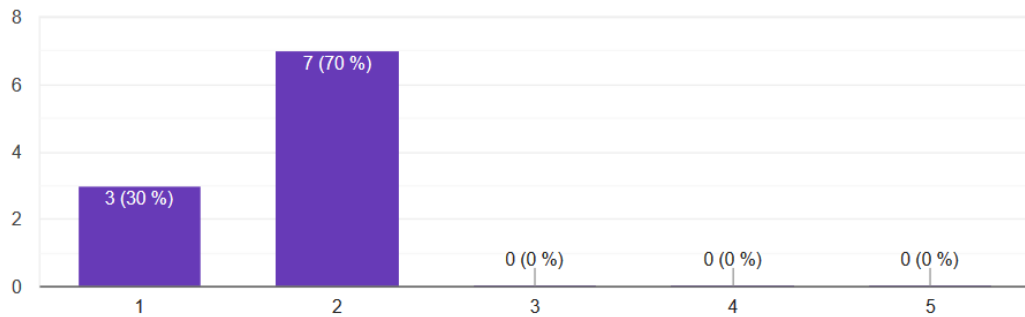
Resultados:



¿Qué nivel de dificultad presentó al momento de realizar la consulta del consumo por meses?



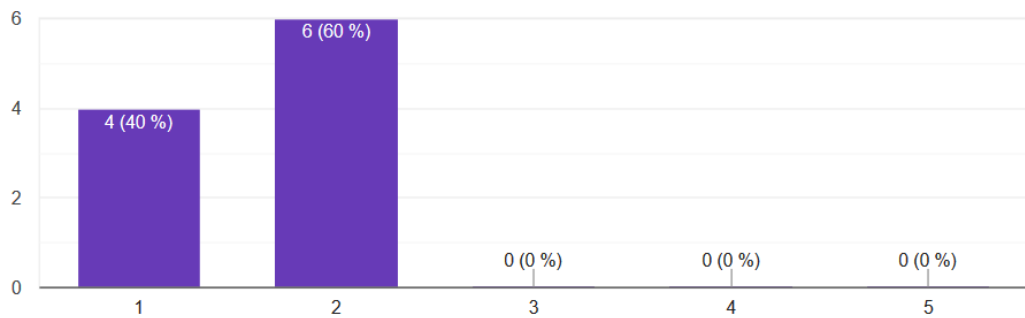
10 respuestas



¿Qué nivel de dificultad presentó al momento de realizar la consulta del consumo entre distintas fechas?

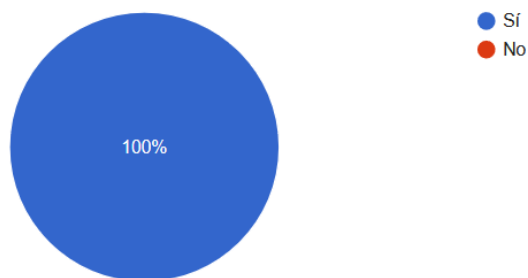


10 respuestas



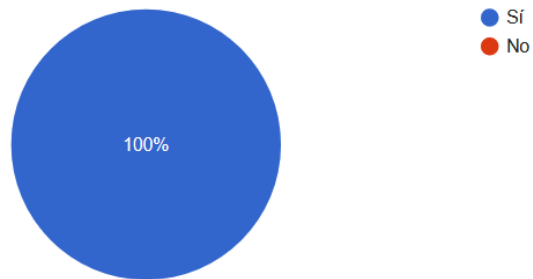
¿El uso de los diferentes histogramas para observar los valores de consumo de la consulta de consumo por fechas le fué fácil usar?

10 respuestas



¿Considera que la aplicación móvil dispone de una interfaz amigable y que es fácil de usar?

10 respuestas





## **ORDEN DE EMPASTADO**