



Aplicación para el Monitoreo de Consumo de Agua y Electricidad en el Hogar

Leandro Nicolás Cardozo

Ingeniería en Informática
Facultad de Ingeniería
Universidad Católica de Salta
2019

Título: Ingeniero en Informática

Alumno: Cardozo Leandro Nicolás

Firma:

Profesor Guía: Ing. Nancy del Carmen Lucero

Firma:

Tribunal Evaluador:

Nombre:

Firma:

Nombre:

Firma:

Nombre:

Firma:

Fecha de exposición del trabajo:

Agradecimientos

Quiero agradecer a mi padre que creyó en mí e hizo el esfuerzo de irme a buscarme a la Universidad cuando tenía tiempo disponible sin importar lo cansado que se encontraba, a mi madre por sus palabras de aliento cuando sacaba una alguna mala nota y por su apoyo incondicional.

A mis hermanos que siempre estuvieron presentes y en varias ocasiones prestaron su oído para que yo pueda practicar las exposiciones.

A mis abuelos y tíos siempre que estuvieron atentos a mi progreso en la facultad y comprensivos cuando no podía ir a visitarlos cada vez que tenía que estudiar para algún examen y en especial a mi tía Carmen que durante estos años hizo posible que curse la carrera.

Y por último un agradecimiento a mis profesores que me inspiraron a aprender y por todas las enseñanzas brindadas a lo largo de la carrera que no olvidaré.

Índice de contenidos

Agradecimientos	III
Índice de contenidos	IV
Índice de tablas	VI
Índice de figuras	VII
Abstract	IX
1. Introducción	1
1.1. Motivación	2
2. Estado de la cuestión	3
2.1. Antecedentes	3
2.1.1. Wabee	3
2.1.2. Smappee	3
2.1.3. Hydrao Smart Shower	3
2.2. Fundamentos teóricos	4
2.2.1. Android	4
2.2.2. Arduino	7
3. Definición del problema	8
3.1. Objetivo general	8
3.2. Objetivos específicos	8
3.3. Restricciones	9
3.4. Alcance	9
3.5. Recursos	9
3.6. Alternativas tecnológicas de solución al problema	10
4. Solución propuesta	12
4.1. Planificación del proyecto	12
4.1.1. Recursos de la solución	12
4.1.2. Gestión de riesgos	14
4.1.3. Plan de contingencia	15
4.1.4. Análisis FODA	15
4.1.5. Análisis de factibilidad	17
4.2. Aplicación de la Metodología Scrum	20
4.2.1. Roles	20
4.2.2. Product backlog	21
4.2.3. Historias de usuario	21
4.2.4. Planificación de Sprints	23
4.3. Desarrollo del proyecto	24
4.3.1. Caudalímetro	24
4.3.2. Sensor de corriente SCT-013	27
4.3.3. Caudalímetro y sensor de corriente en conjunto	30
4.3.4. Primer prototipo con AppInventor	31
4.3.5. Módulo esp8266	35

4.3.6.	API	37
4.3.7.	Conexión del esp8266 a la API	40
4.3.8.	Segundo prototipo de la aplicación	44
4.3.9.	Versión estable de la aplicación	47
4.3.10.	Finalización del proyecto	51
5.	Resultados y verificación experimental	53
5.1.	Caudalímetro	53
5.2.	Sensor de corriente	54
5.3.	Dificultades no previstas encontradas y sus soluciones	56
6.	Conclusiones	58
6.1.	Futuras líneas de investigación	58
	Bibliografía	59
	Anexos	62
	Anexo A: Como subir un programa a Arduino	62
	Anexo B: Código del caudalímetro	64
	Anexo C: Código del sensor SCT	66
	Anexo D: Código del caudalímetro y del sensor SCT	67
	Anexo E: Proceso de configuración del modulo esp8266	69
	Anexo F: Instalación de la aplicación de AppInventor y conexión con Arduino	71
	Anexo G: Configuración de phpMyAdmin	75
	Anexo H: Código del proyecto	77
	Anexo I: Construcción del circuito eléctrico	80
	Anexo J: Creación e instalación de la aplicación	84
	Anexo K: Archivo php	87

Índice de tablas

1.	Consumo de electrodomésticos	1
2.	Versiones de Android	4
3.	Riesgos	14
4.	Identificación de los riesgos en la matriz	15
5.	Matriz FODA	16
6.	Costos de software	18
7.	Costos de hardware	19
8.	Costo de los recursos humanos	19
9.	Costo total	19
10.	Product backlog	21
11.	Historia de usuario N°1	21
12.	Historia de usuario N°2	22
13.	Historia de usuario N°3	22
14.	Historia de usuario N°4	22
15.	Historia de usuario N°5	22
16.	Historia de usuario N°6	22
17.	Sprint N°1	23
18.	Sprint N°2	23
19.	Sprint N°3	24
20.	Sprint N°4	24
21.	Comandos AT	36
22.	Descripción de los archivos php	39
23.	Tabla de potencia de dispositivos	55

Índice de figuras

1.	Arquitectura de Android	5
2.	Caudalímetro YF-S201	12
3.	Placa Arduino UNO SMD R3	13
4.	Sensor de corriente alterna SCT-013-030.	13
5.	Módulo Esp8266 Esp01	13
6.	Fuente para la placa protoboard	14
7.	Diagrama de conexión del caudalímetro	25
8.	Primera conexión del caudalímetro con Arduino	26
9.	Mediciones del caudalímetro en Arduino	27
10.	Diagrama de conexión del sensor de corriente	28
11.	Conexión del sensor SCT-013	28
12.	Conexión del SCT	29
13.	Mediciones del sensor de corriente en Arduino	29
14.	Imagen de las conexiones del caudalímetro y el sensor SCT	30
15.	Muestra de datos con ambos dispositivos	31
16.	Primer diseño en AppInventor	32
17.	Estructura de bloques en AppInventor	32
18.	Conexión con el cable OTG al caudalímetro	33
19.	Muestra de datos del caudalímetro con cable OTG	34
20.	Conexión del cable OTG al sensor de corriente	34
21.	Muestra de datos del sensor de corriente con cable OTG	35
22.	Diagrama de conexión del esp8266 sin el microcontrolador	36
23.	Diagrama de conexión del esp8266 con el microcontrolador	36
24.	Resultados de los comandos AT	37
25.	Diagrama de conexión general	38
26.	Interfaz del programa Xampp	38
27.	Diagrama del proyecto	41
28.	Conexión de todos los componentes en protoboard	42
29.	Datos enviados desde esp8266 a la API	42
30.	Segunda captura de datos enviados del esp8266	43
31.	Datos del sensor de corriente en MySQL	44
32.	Datos del caudalímetro en MySQL	44
33.	Respuesta de la base de datos	45
34.	URLs de php	45
35.	Función de llamada a la API	46
36.	Primera captura de los datos en android	46
37.	Segunda captura de los datos en android	47
38.	Menú de la aplicación	47
39.	Tabla de consumo de agua	48
40.	Tabla de consumo de electricidad	48
41.	Página de estadísticas	49
42.	Página de datos en tiempo real	50
43.	Página de About	50
44.	Primera página de consejos	51
45.	Segunda página de consejos	51
46.	Componentes dentro del gabinete	52

47.	Vista con el gabinete cerrado	52
48.	Conexión al grifo	53
49.	Resultado de la prueba de la botella	54
50.	Vista de la zapatilla con los cuatro dispositivos	54
51.	Conexión del sensor de corriente	55
52.	Gráfica de potencia de dispositivos	55
53.	Potencia de los tres dispositivos en la aplicación	56
54.	Mensaje de error de sincronización	56
55.	Configuración de los parámetros del Arduino	57
56.	Selección de puerto	62
57.	Botón para compilar en Arduino	62
58.	Botón para subir el programa en Arduino	63
59.	Mensaje de programa subido correctamente	63
60.	Preferencias en Arduino IDE	69
61.	Gestor de tarjetas	70
62.	Configuración terminada	70
63.	Código QR para la instalación de AppInventor	71
64.	MIT	71
65.	Vista de instalación	72
66.	Vista de la aplicación instalada	72
67.	Cartel de bloqueo por Play Protect	73
68.	Cartel para dar los permisos de conexión USB	73
69.	Conexión exitosa con el Arduino	74
70.	Creación de la base de datos	75
71.	Creación de tabla en phpMyAdmin	75
72.	Creación de la tabla de consumo de agua	75
73.	Creación de la tabla de consumo de electricidad	76
74.	Diseño del circuito impreso	80
75.	Diseño del circuito impreso en hoja	80
76.	Marcado de los componentes con martillo	81
77.	Marcado con tinta indeleble	81
78.	Circuito en ácido ferrico	82
79.	Vista del circuito fuera del ácido	82
80.	Circuito con las pistas de cobre	82
81.	Circuito Soldado	83
82.	Vista superior del circuito terminado	83
83.	Manifiesto de Android	84
84.	Archivar la aplicación	84
85.	Cargando la aplicación	85
86.	Carga finalizada	85
87.	Archivo Apk	85
88.	Permitir instalar aplicaciones desconocidas	86
89.	Instalación finalizada	86

Abstract

El presente proyecto muestra el desarrollo de una aplicación móvil en la plataforma Android que permite a los usuarios obtener información acerca del consumo de agua y electricidad generado en sus hogares. Esto es posible por la utilización de un caudalímetro y un sensor de corriente eléctrica que están conectados a una placa Arduino por medio de un módulo de Wifi.

Palabras claves: Android, Arduino, Wifi.

Capítulo 1

Introducción

Con los avances tecnológicos de las últimas décadas la cantidad de dispositivos eléctricos aumentó de manera exponencial, provocando que el consumo de agua y electricidad en los hogares sea creciente. Y con la rápida expansión de los dispositivos móviles el acceso a la información mediante las aplicaciones es instantánea y forman parte del uso cotidiano de las personas.

Según Fundelec ¹ en el año 2011 un usuario residencial en Argentina consumía entre 550 kWh y 600 kWh durante un bimestre, no solo este recurso sufrió cambios, también el agua que cada año rompe récords de consumo en el mundo y conforme aumenta la población mundial la misma tiene más demanda. En Argentina según el Gobierno² el consumo estimado es de 318 litros por día, diez veces más de lo que recomienda la Organización Mundial de la Salud. ³

Un kilovatio-hora es equivalente a mantener un consumo de potencia de mil vatios durante una hora que es igual a tener diez lámparas de cien durante una hora, para tener más idea de los valores se presenta la Tabla 1 con la potencia de algunos aparatos eléctricos según Edenor. ⁴

Aparato electrodoméstico	Consumo en kilovatio-hora
Acondicionador de aire	1,013 kWh
Aspiradora	0,675 kWh
Computadora	0,720 kWh
Lavarropa automático	0,182 kWh
Freezer	0,090 kWh
Televisor 29"	0,140 kWh
Ventilador	0,090 kWh
Horno microondas	0,640 kWh
Heladera	0,063 kWh

Tabla 1: Consumo de electrodomésticos

El consumo de agua en los hogares se mide por metros cúbicos que es la unidad de caudal en el Sistema Internacional, el cual es equivalente al caudal de un cubo con lados de un metro de longitud moviéndose cada segundo, un metro cúbico por segundo es igual a mil litros por segundo.

Por lo general las personas no llevan un control sobre la cantidad de recursos que usan a diario, normalmente solo revisan la factura tienen que pagar a fin de mes, así surgió la idea de una solución mediante la aplicación de domótica y un software para monitorear en tiempo real el consumo de agua y electricidad para tomar decisiones con respecto a la información brindada. Además gran parte de la generación de energía eléctrica proviene de recursos no renovables como lo es el petróleo o el carbón, por lo que es esencial no desperdiciar energía para cuidar el medio ambiente.

¹Consumo de electricidad en el hogar y su impacto en la tarifa final. <http://www.fundelec.com.ar/informes/info0031.pdf> [último acceso, Agosto, 2018]

²Plan Nacional de Agua Potable y Saneamiento. https://www.argentina.gob.ar/sites/default/files/interior_agua_plan_agua_saneamiento.pdf [último acceso, Agosto, 2018]

³La cantidad de agua domiciliar, el nivel del servicio y la salud. http://www.who.int/water_sanitation_health/diseases/wsh0302/es/ [último acceso, Agosto, 2018].

⁴Consumo de algunos electrodomésticos. http://www.edenor.com.ar/cms/SP/CLI/HOG/USO_consumo.html# [último acceso, Agosto, 2018].

1.1. Motivación

Las tecnologías que están destinadas a proteger al medio ambiente que existen actualmente en el mercado no tienen suficiente publicidad como otras de consumo masivo, por ejemplo, los celulares o autos de última generación, por lo que pretendo con este proyecto es fomentar el uso de herramientas que posibilitan la fabricación de prototipos de productos que sirven para concientizar sobre el ahorro energía desde el hogar.

Otro motivo que me llevó a realizar el trabajo fue mi paso por Escuela Técnica “Albert Einstein” que me brindó los conocimientos en electrónica y electricidad que me condujeron a interesarme en el empleo de la domótica para resolver un problema.

Capítulo 2

Estado de la cuestión

2.1. Antecedentes

A continuación, se describen productos de las empresas Wabee, Smappe e Hydrao Smart shower que ofrecen una solución similar a la propuesta en este proyecto, que sirven para reflejar el estado actual de las tecnologías que pretenden concientizar sobre el ahorro de recursos ambientales.

2.1.1. Wabee

Wabee⁵ es un emprendimiento que incluye el tanto desarrollo de dispositivos como de software, creado a partir de la necesidad de conocer con anticipación el consumo eléctrico en hogares, comercios y empresas, el producto permite generar un cambio en la cultura del consumo energético, hacia un consumo más responsable y eficiente. Según la zona geográfica y el valor del vatio-hora, el ahorro podría llegar a un 20 % anual, según el fundador del emprendimiento.

2.1.2. Smappee

Los monitores de energía Smappee⁶ muestran información detallada sobre el consumo de energía. El producto realiza más de 4.000 mediciones por segundo según sus creadores, por lo que puede indicar en tiempo real cuánto es el gasto y cuáles son los dispositivos que más consumen. La empresa cuenta con una aplicación que tiene un panel de control donde se puede controlar y revisar en cualquier momento los flujos energéticos de la vivienda o empresa. Esta información es posible obtenerla tanto para la luz, la energía solar, el gas y para el agua.

2.1.3. Hydrao Smart Shower

La Hydrao Smart Shower⁷ es una invención que se utiliza para controlar el consumo del agua cuando nos bañamos. En sí, el producto es un cabezal de ducha con LEDs que va cambiando de color a verde, azul o rojo según el agua que consumamos. Cada color corresponde a uno de los tres niveles de alerta que se pueden configurar a través de una aplicación que está disponible tanto para iOS como Android. Aparte de poder configurar los colores y niveles de consumo, la aplicación que se vincula vía bluetooth, también permite al usuario configurar perfiles y registrar el uso de agua. Una ventaja de este producto es que no necesita de pilas o batería ni conexión a la corriente eléctrica gracias a la turbina que tiene incorporada capaz de producir energía eléctrica con la misma corriente del agua.

⁵Sitio Oficial de Wabee. <https://www.wabee.com.ar/> [último acceso, Agosto, 2018].

⁶Sitio oficial de Smappe https://www.smappee.com/eu_es/homepage [último acceso, Agosto, 2018].

⁷Sitio oficial de Hydrao Smart Shower. <https://www.hydrao.com/en/> [último acceso, Agosto, 2018].

2.2. Fundamentos teóricos

2.2.1. Android

Android según Ian F. Darwin⁸ es una tecnología móvil proporcionada a los teléfonos celulares, tablets, u otro dispositivo de mano, con la potencia y la portabilidad del sistema operativo Linux.

En septiembre del 2008 apareció la primera versión de Android y a partir de ella con cada nueva versión se presentaron nuevas funcionalidades y la solución a los problemas que presentaba anteriormente. Podemos ver en la siguiente tabla todas las versiones que existen hasta la fecha de redacción.

Letra	Nombre	Versión
A	Apple Pie	1.0
B	Banana Bread	1.1
C	Cupcake	1.5
D	Donut	1.6
E	Éclair	2.0 / 2.1
F	Froyo	2.2
G	Gingerbread	2.3
H	Honeycomb	3.0 / 3.1 / 3.2
I	Ice Cream Sandwich	4.0
J	Jelly Bean	4.1 / 4.2 / 4.3
K	KitKat	4.4
L	Lollipop	5.0 / 5.1
M	Marshmallow	6.0 / 6.0.1
N	Nougat	7.0 / 7.1 / 7.1.1 / 7.1.2
O	Oreo	8.0 / 8.1
P	Pie	9.0

Tabla 2: Versiones de Android

La información brindada a continuación sobre la estructura de la arquitectura de la plataforma Android fue extraída de su sitio web oficial de desarrolladores.⁹

Arquitectura

Android está desarrollada para una gran cantidad de dispositivos y factores de forma, en el siguiente diagrama se puede observar los componentes principales de la plataforma.

⁸Darwin, I. F. (2017). Android Cookbook: Problems and Solutions for Android Developers. "O'Reilly Media, Inc."

⁹Arquitectura de la plataforma. Android Developers <https://developer.android.com/guide/platform?hl=es-419> [último acceso, Agosto ,2018]

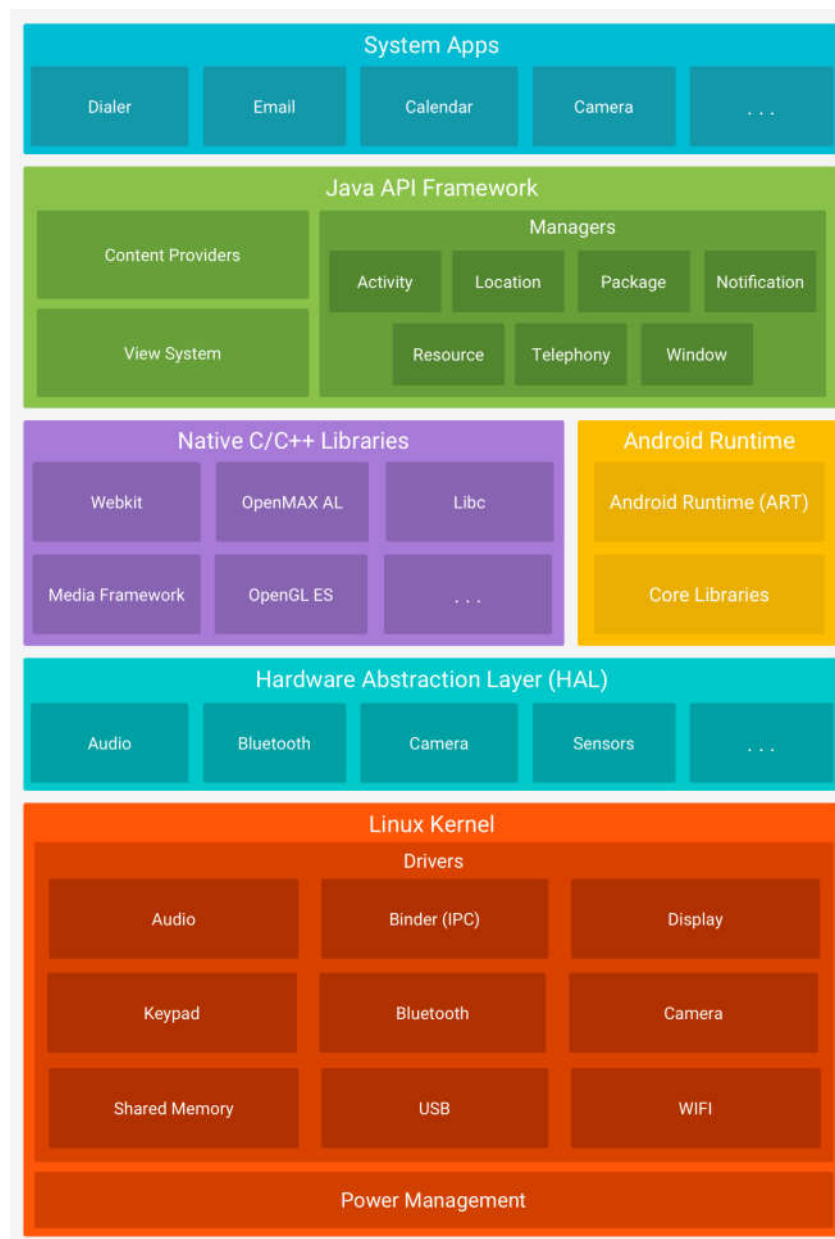


Figura 1: Arquitectura de Android

Kernel de Linux

El uso del kernel de Linux le permite a Android aprovechar las funciones de seguridad y desarrollar controladores de hardware. En los sistemas operativos de escritorio basados en GNU/Linux utilizan la última versión del mismo a diferencia de Android que solamente usa su versión más estable.

Capa de abstracción de hardware

Permite al sistema trabajar siempre con las mismas “instrucciones” aunque el hardware cambie y brinda las interfaces estándares que le dan las capacidades al soporte físico del dispositivo. La capa de abstracción consiste en varios módulos de bibliotecas y cada una implementa una interfaz para un tipo específico de componente, como puede ser el módulo de la cámara o del bluetooth.

Tiempo de ejecución de Android

Para los dispositivos con Android 5.0 o versiones posteriores, cada aplicación ejecuta sus propios procesos con sus instancias del tiempo de ejecución de Android (ART). El mismo está escrito para hacer funcionar varias máquinas virtuales en dispositivos de baja memoria, con el uso de archivos de extensión DEX diseñados especialmente para Android y optimizados para ocupar un espacio de memoria mínimo. Además, crea cadenas de herramientas como Jack y compila fuentes de Java que se pueden ejecutar en la plataforma Android.

Estas son algunas de las funciones principales del tiempo de ejecución de Android:

- Compilación anticipada y uso del método “justo a tiempo” que es un sistema de organización.
- Recolección de elementos no usados.
- Mejor compatibilidad con la depuración, como un generador de perfiles de muestras dedicado, excepciones de diagnóstico detalladas e informes de fallos, y la capacidad de establecer puntos de control para controlar campos específicos.

En Android también se incluye un conjunto de bibliotecas de tiempo de ejecución centrales que proporcionan la mayor parte de la funcionalidad del lenguaje de programación Java; se incluyen algunas funciones del lenguaje Java 8, que el framework de la Java API usa.

Bibliotecas C/C++ nativas

Muchos componentes y servicios centrales del sistema Android, como la capa de abstracción y el tiempo de ejecución, se basan en código nativo que requiere bibliotecas escritas en C y C++. La plataforma Android proporciona la API del framework de Java para exponer la funcionalidad de algunas de estas bibliotecas a las aplicaciones. Por ejemplo, se puede acceder a OpenGL ES a través de la Java OpenGL API del framework de Android para agregar a tu aplicación compatibilidad con los dibujos y la manipulación de gráficos 2D y 3D.

Si desarrollas una aplicación que requiere C o C++, puedes usar el NDK de Android para acceder a algunas de estas bibliotecas de plataformas nativas directamente desde tu código.

Framework de la Java API

Todo el conjunto de funciones del SO Android está disponible mediante las API escritas en el lenguaje Java. Estas API son los cimientos que se necesita para crear aplicaciones de Android simplificando la reutilización de los componentes del sistema, servicios centrales y modulares, como los siguientes:

- Un sistema de vista enriquecido y extensible que se usa para compilar la IU de una aplicación; se incluyen listas, cuadrículas, cuadros de texto, botones e incluso un navegador web integrable.
- Un administrador de notificaciones que permite que todas las aplicaciones muestren alertas personalizadas en la barra de estado.
- Un administrador de actividad que gestiona el ciclo de vida de las aplicaciones y proporciona una pila de retroceso de navegación común.
- Proveedores de contenido que permiten que las aplicaciones accedan a los datos de otras.

Los desarrolladores tienen acceso total a las mismas API del framework que usan las aplicaciones del sistema Android.

Apps del sistema

En Android se incluye un conjunto de aplicaciones centrales para el correo electrónico, mensajería SMS, calendarios, navegación en Internet y contactos, entre otros elementos. Las aplicaciones incluidas en la plataforma no tienen un estado especial entre las que el usuario elige instalar; por ello, una aplicación externa se puede convertir en el navegador web, el sistema de mensajería SMS o, incluso, el teclado predeterminado del usuario (existen algunas excepciones, como la configuración del sistema).

2.2.2. Arduino

Arduino ¹⁰ es una plataforma electrónica de código abierto con un hardware y software que son fáciles de usar y han servido en distintos proyectos a los aficionados que recién comienzan con la programación y la robótica como así también a los usuarios más avanzados.

Como lo explica la publicadora de Weblogs Xataka ¹¹ el Arduino es una placa basada en un microcontrolador ATMELE. Los microcontroladores son circuitos integrados en los que se pueden grabar instrucciones, las cuales las escribes con el lenguaje de programación que puedes utilizar en el entorno Arduino IDE. Estas instrucciones permiten crear programas que interactúan con los circuitos de la placa.

El microcontrolador de Arduino posee lo que se llama una interfaz de entrada, que es una conexión en la que podemos conectar en la placa diferentes tipos de periféricos. La información de estos periféricos que conectes se trasladará al microcontrolador, el cual se encargará de procesar los datos que le lleguen a través de ellos.

El tipo de periféricos que puedas utilizar para enviar datos al microcontrolador depende en gran medida de qué uso le estés pensando dar. Pueden ser cámaras para obtener imágenes, teclados para introducir datos, o diferentes tipos de sensores.

También cuenta con una interfaz de salida, que es la que se encarga de llevar la información que se ha procesado en el Arduino a otros periféricos. Estos periféricos pueden ser pantallas o altavoces en los que reproducir los datos procesados, pero también pueden ser otras placas o controladores

¹⁰Arduino, L. L. C. (2015). Arduino uno.

¹¹¿Qué es Arduino, cómo funciona y qué puedes hacer con uno? <https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno> [último acceso, Agosto ,2018]

Capítulo 3

Definición del problema

Los consumidores finales por lo general no revisan los medidores de agua y electricidad que tienen en sus hogares tanto sea para comparar las variaciones en los costos mensualmente y/o para tener un control de este, una de las razones es que no resulta muy práctico. Por otro lado, los distribuidores no proveen una herramienta que brinde información del consumo diario en su página web o una aplicación móvil. Se puede apreciar la falta de un sistema que ofrezca un seguimiento de datos, que sea de rápido acceso y de tener otras alternativas de medir el consumo que no sea la brindada por las empresas que están a cargo de la distribución.

También hay que remarcar que Argentina se encuentra entre los países con más inflación en el mundo. Según las estadísticas brindadas por el Banco Central de la República Argentina¹² se estimó en Julio del 2018 mediante un análisis de expectativas que la inflación proyectada anualmente será de 31,8 % por lo que los servicios básicos como el agua y el de la electricidad inevitablemente aumentan y afectan a la economía hogareña, en consecuencia los consumidores finales buscarán distintas formas de ahorrar para reducir este costo.

Las funciones de gestión de los recursos hídricos del país son administrados por diferentes organizaciones. Para el año 2019 se planea tener un nivel de cobertura del 100 % de agua potable y 75 % de cloaca según la Subsecretaría de recursos hídricos¹³. Por lo cual es un motivo para cuidar de este recurso que es escaso, no solo siguiendo las recomendaciones que difunden diversos organismos si no que también buscando diversas tecnologías al alcance del público.

Este problema no es reciente, la falta de inversiones¹⁴ en cuanto a infraestructura, plantas hidroeléctricas y nucleares, planificación sectorial, tarifas igualitarias, el abandono de las políticas de integración y el subsidio de las personas que no lo necesitan hacen que la situación empeore. Se hace visible cuando se ven cortes de luz en distintos puntos del país y la dificultad de la expansión de los servicios de agua potable a las áreas urbano marginales.

3.1. Objetivo general

Es desarrollar una aplicación en la plataforma de Android para mostrar la cantidad de consumo en tiempo real de potencia eléctrica en kilovatios-hora con la utilización de un sensor de corriente alterna y litros de agua con un caudalímetro. La misma está pensada para su uso dentro del hogar de una familia tipo.

3.2. Objetivos específicos

- Desarrollar una infraestructura electrónica capaz de simplificar las conexiones entre los componentes del proyecto.

¹²Resultados del Relevamiento de Expectativas de Mercado (REM). <http://www.bcra.gov.ar/Pdfs/PublicacionesEstadisticas/REM180731%20Resultados%20web.pdf> [último acceso, Agosto, 2018].

¹³Plan Nacional de Agua Potable y Saneamiento https://www.argentina.gob.ar/sites/default/files/interior_agua_plan_agua_saneamiento.pdf [último acceso, Agosto, 2018]

¹⁴La falta de inversión en infraestructura impacta en el crecimiento. <https://www.lanacion.com.ar/economia/la-falta-inversion-infraestructura-impacta-crecimiento-nid2174951> [último acceso, Agosto, 2018].

- Desarrollar una API a fin de realizar las transferencias de datos entre la aplicación y el Arduino.

3.3. Restricciones

El trabajo está condicionado por diferentes factores, por lo cual no se contemplan algunos aspectos en concreto tales como:

- La aplicación está diseñada para el uso en el ámbito hogareño en zonas urbanas, no para empresas o industrias.
- La única plataforma móvil compatible es Android con una versión 5.0 o superior.
- Los datos brindados por la aplicación solo podrán ser mostrados cuando se tenga una conexión a Internet por medio de una red wifi.
- La limitación del funcionamiento brindado por las especificaciones técnicas del sensor de corriente y del caudalímetro.

3.4. Alcance

El trabajo se hizo en Visual Studio 2017 Community como un proyecto de Xamarin para dispositivos con una versión de Android mayor o igual a 5.0, el funcionamiento del mismo se limita dentro de una red de wifi doméstica que ejecute un servidor Apache, una base de datos MySQL y que tenga instalado los componentes del proyecto para recaudar los datos.

La cantidad de litros de agua que puede resistir el caudalímetro es de 30 por minuto según los datos brindados por el fabricante ¹⁵ y del sensor de corriente de 30 amper ¹⁶, ese el alcance de nuestros componentes teniendo en cuenta que existe un margen de error al no ser productos de precisión. Los costos estimados de consumo no tienen en cuenta todos los impuestos de la factura, solo la energía activa y el consumo por metros cúbicos de agua. La aplicación no estará disponible en una tienda online para su descarga.

Durante el desarrollo primero se especificó los requisitos que definen el funcionamiento del sistema, luego se procedió a experimentar con los componentes del proyecto y se estableció las bases para poder transmitir datos. A partir de aquí se realizaron dos prototipos y se definió las interfaces finales para terminar con la codificación para pasar a realizar la verificación experimental. También se diseñó un circuito eléctrico para facilitar las conexiones realizadas.

Se realizó el trabajo mediante la utilización de Scrum que es una metodología ágil que se puede emplear para el desarrollo de aplicaciones móviles, con un total de 4 sprints.

3.5. Recursos

Actualmente existen las siguientes formas para conocer el consumo de agua y electricidad en el hogar:

¹⁵Datasheet YF-S201. <https://www.hobbytronics.co.uk/datasheets/sensors/YF-S201.pdf> [último acceso, Junio, 2019].

¹⁶Datasheet SCT-013-030. <http://statics3.seeedstudio.com/assets/file/bazaar/product/101990028-SCT-013-030-Datasheet.pdf> [último acceso, Junio, 2019].

- La medición de la energía eléctrica se hace mediante un medidor de consumo eléctrico, uno de los parámetros que se miden son los kilovatios-hora, es instalado por la empresa encargada de la distribución para fines domésticos y comerciales.
- Mediante programas simuladores de consumo eléctrico se puede calcular el consumo de energía eléctrica en el hogar, pero estos no son muy precisos porque usan valores promedios de cada uno de los aparatos electrodomésticos que hay en el hogar. Estos programas no son usados en este proyecto.
- Un caudalímetro que es un instrumento que tiene el fin de medir el caudal de un fluido y por medio de una conexión eléctrica envía los datos.

3.6. Alternativas tecnológicas de solución al problema

La empresa Dio ¹⁷ vende distintos productos de domótica inalámbrica los cuales permiten prender y apagar luces o dispositivos eléctricos a distancia, también productos de calefacción, vigilancia y apertura y cierre de persianas.

Nimbus ¹⁸ es un lavarropas que no necesita agua para funcionar, solo requiere dióxido de carbono y un detergente reciclable, con una capacidad de 20 kilos de ropa y una duración de 30 min de lavado, es el método de limpieza más efectivo y cuidadoso.

Las Tecnologías verdes o Green Computing ¹⁹ contribuyen a la reducción del consumo de energía o emisión del dióxido de carbono minimizando su impacto mediante el uso eficiente de los recursos computacionales.

Algunos ejemplos de esta tecnología se pueden ver en los centros de datos que apagan los nodos ociosos y los encienden de nuevo cuando el sistema los necesita, además de la reducción del hardware mediante el estudio del porcentaje que se usa cada equipo. Asimismo de la posibilidad de reubicar el datacenter a un lugar donde puedan aprovechar las energías renovables de una forma más eficiente.

Según la multinacional de software Red Hat ²⁰ la virtualización permite crear servicios mediante recursos que normalmente se ejecutan en el hardware con el objetivo de usar toda la capacidad de procesamiento de una máquina, eso mismo permite reducir la huella de carbono al utilizar menos infraestructura computacional, además de lograr un aumento el uso de recursos y ahorro de espacio, otro beneficio es la disminución de los costos y el aumento de la velocidad de los programas.

Como lo explica la publicadora ACM ²¹ la computación en la nube es una forma de computación distribuida que permite a los usuarios acceder a una amplia red de recursos informáticos a través de Internet, al usarlas los equipos se vuelven más ecológicos al disminuir su capacidad de procesamiento con lo cual reducen su consumo de energía. La ventaja es que los usuarios no necesitan tener conocimientos o estar capacitados para emplear esta tecnología.

¹⁷Dio Connected Home <https://getdio.com/es/14-iluminacion-inteligente> [último acceso, Junio, 2019]

¹⁸Nimbus, el lavarropas sin agua que limpia hasta 20 kilos de ropa. <https://twenergy.com/ar/a/nimbus-el-lavarropas-sin-agua-que-limpia-hasta-20-kilos-de-ropa-2081> [último acceso, Junio, 2019]

¹⁹Martínez, A. N., & Porcelli, A. M. (2015). Impactos de las tecnologías informáticas en el Ambiente y nuevas tendencias en computación verde. *Diario DPI Suplemento Derecho y Tecnologías* N^o, 11-30.

²⁰¿Qué es la virtualización? <https://www.redhat.com/es/topics/virtualization/what-is-virtualization> [último acceso, Junio, 2019]

²¹A Nimbus Rises in the World of Cloud Computing. <https://cacm.acm.org/news/26083-a-nimbus-rises-in-the-world-of-cloud-computing/fulltext> [último acceso, Junio, 2019]

De acuerdo con la Universidad estatal a distancia²² el teletrabajo se fundamenta en el uso intensivo de Internet y de las tecnologías de la información. Con el uso de las mismas las empresas se convierten en entidades más ágiles ya que el trabajo que se hace desde el hogar y también contribuye minimizando la cantidad de gas que deberían usar los empleados al ir a la oficina, también es usado como una herramienta de retención para evitar la fuga de personal.

²²¿Qué es teletrabajo? <https://www.uned.ac.cr/viplan/teletrabajo/que-es-teletrabajo/que-es-teletrabajo> [último acceso, Junio, 2019]

Capítulo 4

Solución propuesta

Mediante la utilización de la aplicación desarrollada se propone facilitar la información a los usuarios finales para que ellos tomen una decisión con respecto a la modificación de su consumo de agua y electricidad para ser beneficiados económicamente y que tengan otra alternativa para comparar los datos que brindan los medidores de los proveedores de los servicios.

4.1. Planificación del proyecto

4.1.1. Recursos de la solución

Los recursos de hardware y software que se usaron en el proyecto son los siguientes:

- **Visual Studio 2017 Community:** Es el entorno de desarrollo integrado de Microsoft. Está disponible para los sistemas operativos de Windows y macOS.

La elección de este entorno se debe a que podemos desarrollar aplicaciones en Xamarin con una gran cantidad de herramientas y una comunidad activa en el mercado que brinda soporte a los problemas.

- **Caudalímetro YF-S201:** Contiene un sensor para medir la cantidad de líquido que transcurre por el mismo.



Figura 2: Caudalímetro YF-S201

- **IDE de Arduino:** El entorno de desarrollo de Arduino es una herramienta diseñada para distintas plataformas que está escrita en el lenguaje de programación Java. Tiene el propósito de crear programas que estén aplicados a la placa Arduino.

Tiene la ventaja que admite distintos lenguajes como ser C y C++ con la utilización de reglas especiales de estructuración de código, y tiene una biblioteca completa que brinda procedimientos de entrada y de salida que facilitan la programación.

- **Placa Arduino UNO R3:** Se destaca por la aparición de un puerto USB que facilita el uso, aumenta los ratios de transferencia como también la velocidad y elimina la necesidad de la instalación de drivers para los sistemas operativos de Linux y Mac.



Figura 3: Placa Arduino UNO SMD R3

- **Sensor de corriente alterna SCT-013-030:** Una característica del mismo es que no necesita que se corte o interrumpa la línea de corriente que se quiere medir, ya que tiene un núcleo partido como una pinza amperimétrica.



Figura 4: Sensor de corriente alterna SCT-013-030.

- **Cables** Se utilizó para realizar las conexiones de los sensores con la placa Arduino y con la placa protoboard.
- **Módulo ESP8266 ESP-01:** Su principal función es conectar a la placa Arduino a una red de wifi. Existen muchos modelos que integran el ESP8266, se eligió el esp-01 que contiene una completa documentación.

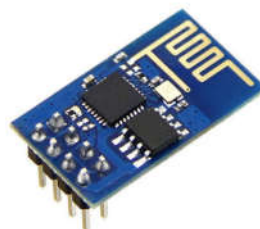


Figura 5: Módulo Esp8266 Esp01

- **Fuente para placa protoboard:** Brinda una alimentación de 5 y 3,3 volts a ambos lados de una placa protoboard.



Figura 6: Fuente para la placa protoboard

- **Fritzing:** Es un software que permite diagramar las conexiones de los componentes de una manera sencilla y eficiente.
- **PCB Wizard:** Es un software de electrónica para diseñar los circuitos impresos.

4.1.2. Gestión de riesgos

El objetivo es identificar, analizar y evaluar los posibles riesgos durante el desarrollo de la aplicación que puedan tener un impacto negativo.

Los riesgos que se identificaron son los siguientes:

- **Dificultad en la obtención del hardware:** La posible falta de stock en los negocios cercanos puede generar la necesidad de realizar compras en sitios de Internet, provocando una demora en el desarrollo del proyecto.
- **Retraso por compra de hardware con defectos:** Puede ocurrir que los componentes obtenidos tengan fallas que vienen de fábrica por lo que el reemplazo de los mismos producirá un aplazamiento en el proyecto.
- **Deserción de algún miembro fundamental del equipo:** Si un integrante del grupo abandona el proyecto y su participación es importante para avanzar en el trabajo, provocará una demora notable en la planificación, además que afectará al estado del ánimo de los miembros restantes.

Matriz de probabilidad e impacto

Es una herramienta que nos facilita determinar la prioridad de cada uno de los riesgos, la ocurrencia de cada uno de ellos y cual será el impacto en el proyecto una vez producida.

Riesgo	Ocurrencia	Impacto	Valoración	Prioridad
Dificultad de obtención del hardware	2	4	8	5
Retraso por compra de hardware con defectos	3	4	12	4
Deserción de algún miembro fundamental del equipo	2	3	6	3

Tabla 3: Riesgos

Impacto\Ocurrencia	Raro	Poco probable	Probable	Muy probable	Casi seguro
Despreciable	-	-	-	-	-
Menores	-	-	-	-	-
Moderadas	-	Deserción de algún miembro fundamental del equipo	-	-	-
Mayores	-	Dificultad de obtención del Hardware	Retraso por compra de hardware con defectos	-	-
Catastróficas	-	-	-	-	-

Tabla 4: Identificación de los riesgos en la matriz

Los valores que pueden tomar la ocurrencia de los riesgos está en un intervalo del uno al cinco donde el uno significa una ocurrencia rara y el cinco que es casi segura. Los valores del impacto de igual manera se encuentran en el intervalo del uno al cinco donde el mayor valor significa un impacto catastrófico y el menor que es despreciable. La valoración es el resultado del producto entre la ocurrencia y el impacto. El intervalo de la prioridad varía del uno al cinco donde el último representa una mayor prioridad.

4.1.3. Plan de contingencia

Son los procedimientos que se deben seguir con respecto a los potenciales riesgos del proyecto con el objetivo de evitarlos de ser posible o en caso contrario disminuir su impacto para garantizar la continuidad del funcionamiento del proyecto.

- **Dificultad de obtención del hardware:** Buscar diferentes proveedores en el exterior para la obtención de los componentes y verificar la planificación del proyecto para calcular la cantidad de días de retraso si ocurriese.
- **Retraso por compra de hardware con defectos:** Verificar el funcionamiento cuando recién se los obtenga y devolverlos en caso de que tengan fallas.
- **Deserción de algún miembro fundamental del equipo:** Es importante tener varios currículums de personas con las aptitudes y conocimientos necesarios para cada puesto del proyecto, por si es necesario reemplazar alguno.

4.1.4. Análisis FODA

El siguiente análisis estratégico permite identificar cuales son las fortalezas, oportunidades, debilidades y amenazas a las que se enfrenta el proyecto, para determinar cuál es su estado y permitir desarrollar estrategias que faciliten lograr los objetivos propuestos. Los factores externos del proyecto que analiza esta estrategia son las amenazas y oportunidades y los factores internos son las fortalezas y debilidades.

	Fortalezas	Debilidades
Oportunidades	Estrategias FO Estrategias que utilizan las fortalezas para maximizar las oportunidades	Estrategias DO Estrategias para minimizar las debilidades aprovechando las oportunidades
Amenazas	Estrategias FA Estrategias que utilizan las fortalezas para disminuir las amenazas	Estrategias DA Estrategias que minimizan las debilidades evitando las amenazas

Tabla 5: Matriz FODA

Teniendo en cuenta la estructura de la matriz FODA se van a definir sus componentes por separado.

Análisis interno

Los elementos que se analizan son las fortalezas y las debilidades del proyecto tales como la calidad del producto, el costo, la capacidad de los recursos humanos disponibles, la percepción de los usuarios, entre otros aspectos.

Fortalezas

- La capacidad de ofrecer al mercado una alternativa a las soluciones que existen actualmente.
- Reducción de los errores gracias al seguimiento de las buenas prácticas y a la correcta implementación de la metodología ágil.
- El soporte existente para el desarrollo de aplicaciones móviles en Android y para la creación de proyectos con la placa Arduino, en las páginas oficiales de los mismos.

Debilidades

- Incompatibilidad con otros sistemas operativos para móviles que no sean Android.
- La necesidad de estar conectados a una red Wifi donde se ubica la placa Arduino para que funcione de manera correcta la aplicación.

Análisis externo

Los elementos que se analizan son las oportunidades y las amenazas, son los factores externos del proyecto, se pueden determinar estableciendo eventos que tienen relación con el proyecto o comparando con otros productos similares.

Oportunidades

- Menor costo en el precio final, lo que permite atraer a los usuarios finales con mayor facilidad.
- Fomentar el cuidado de los recursos mediante una sección particular de la aplicación que brinda consejos de como hacerlo.

Amenazas

- Son los productos desarrollados por otras empresas están que consolidados en el mercado y tienen la confianza de los usuarios.
- Las fluctuaciones económicas que pueden provocar la disminución de potenciales usuarios de la aplicación por la dificultad de obtener los componentes necesarios para el desarrollo del trabajo.

Estrategias a implementar

Con la información del análisis interno y externo del proyecto se proponen las siguientes estrategias, FO donde se maximizan las fortalezas para aprovechar las oportunidades, FA donde se maximiza las fortalezas para enfrentar las amenazas, DO las cuales se intenta mejorar las debilidades ante las oportunidades, DA se denotan las debilidades del proyecto para resistir las amenazas.

Estrategias FO

- Con la gran cantidad de usuarios de la plataforma Android es posible llegar a una mayor cantidad de personas y hacer conocer a más personas que existe la tecnología para cuidar al medio ambiente.
- Por los bajos costos de desarrollo puede ser adquirido por una mayor cantidad de personas.

Estrategias FA

- Los beneficios de utilizar la aplicación hacen que los costos del desarrollo se amortizan con el transcurso de los meses con una futura implementación de publicidad.
- Desarrollar nuevos planes de contingencia lo suficientemente robusto como para amortiguar cualquier fluctuación negativa en el mercado.

Estrategias DO

- Ofrecer información, funcionalidades y estadísticas al igual que los productos similares de otras empresas.

Estrategias DA

- Buscar formas para permitir enviar los datos de la placa Arduino a través de internet para poder usar la aplicación sin necesidad de estar conectados a la red wifi.
- Permanecer actualizados e informados de nuevas tendencias en el mercado del ámbito de la tecnología.

4.1.5. Análisis de factibilidad

Factibilidad técnica

El proyecto si es factible técnicamente, porque se pueden conseguir todos los dispositivos tecnológicos necesarios para el trabajo, además los precios de los componentes no son demasiados elevados como para dificultar su obtención.

En cuanto a los requerimientos de software, las notebooks de cada miembro del equipo ya cuentan con las herramientas de desarrollo, las cuales son el Visual Studio Community, el entorno de desarrollo de Arduino, la herramienta de Fritzing para realizar los diagramas de conexión y PCB Wizard para diseñar el circuito impreso. No se detectaron problemas de compatibilidad de software con el sistema operativo Windows 10 instalado en cada máquina.

Factibilidad operativa

Es factible operativamente para el desarrollo del proyecto propuesto porque se cuentan con los recursos humanos necesarios en la Ciudad de Salta y ellos son:

- Product Owner
- Scrum Master
- Técnicos en informática
- Desarrolladores

Para los potenciales usuarios es factible operativamente porque para utilizar la aplicación desarrollada no se necesitan conocimientos que requieran una capacitación. Solamente necesitarán ayuda en la instalación del hardware, por lo que van a requerir el apoyo de un técnico, para realizar el montaje del caudalímetro y del sensor de corriente y la preparación de un servidor para poder almacenar los datos.

Respecto a lo que es obsolescencia del software, se garantiza que el sistema entregado al cliente estará actualizado permanentemente, lo que alargará su vida útil.

Factibilidad legal

En este trabajo se utilizaron herramientas de software libre y los componentes de hardware se obtuvieron de forma legal, con lo cual no se infringió ninguna norma legal o ley, por lo tanto, el proyecto es factible legalmente.

Factibilidad económica

Inversión Inicial

Se necesitarán seis personas en total para desarrollar el proyecto, entre ellos, tres programadores que se encargarán del diseño, desarrollo y testeo de la aplicación, un técnico para la construcción del gabinete con el circuito impreso incluido, el scrum master que facilitará la aplicación de la metodología de desarrollo y el product owner a cargo de la planificación de cada sprint y de la gestión del proyecto.

Para poder desarrollar el trabajo se requiere de los siguientes recursos de hardware y software.

Software	Subtotal
Visual Studio Community	\$ 0,00
IDE Arduino	\$ 0,00
Fritzing	\$ 0,00
PCB Wizard	\$ 0,00
Total	\$ 0,00

Tabla 6: Costos de software

Hardware	Cantidad	Subtotal
Caudalímetro Yf-s201	1	\$ 210,00
Sensor de corriente SCT-013	1	\$ 400,00
Módulo Esp8266 Esp-01	1	\$ 210,00
Arduino Uno R3	1	\$ 378,00
Costos de envío	1	\$ 810,00
Cables de Arduino	60	\$ 210,00
Fuente de alimentación para protoboard	1	\$ 82,00
Circuito integrado LM358	1	\$ 25,00
Zócalo para el circuito integrado	1	\$ 10,00
Placa de cobre 5x5	1	\$ 26,00
Ácido ferrico	1	\$ 90,00
Aerosol protector de circuitos	1	\$ 200,00
Gabinete	1	\$ 300,00
Total		\$ 2951,00

Tabla 7: Costos de hardware

Los costos de envío surgen por la adquisición del sensor de corriente y el caudalímetro.

Al equipo se le pagará por la cantidad de horas de trabajo. Los honorarios corresponden a los datos que brinda el Copaipa.²³

Se establece como honorario al rol de Product Owner como si fuera un líder de proyecto y al Scrum Master como un coach, ya que no existen referencias específicas para esos puestos.

Puesto de Trabajo	Cantidad	Costo por hora	Cant. hs por día	Días	Subtotal
Programador	3	\$ 112	4	28	\$ 37.632
Scrum Master	1	\$ 104	4	28	\$ 11.648
Product Owner	1	\$ 254	3	62	\$ 47.244
Técnico	1	\$ 84	2	50	\$ 8.400
				Total	\$ 104.924

Tabla 8: Costo de los recursos humanos

Costos	Subtotal
Costo de Hardware	\$ 2.951
Costo de Software	\$ 0,00
Costo de Recursos Humanos	\$ 104.924
Total	\$ 107.875

Tabla 9: Costo total

La inversión necesaria del proyecto es aproximadamente de 110 mil pesos argentinos.

Financiamiento

En Argentina mediante diferentes programas, planes y herramientas crediticias se pueden financiar proyectos tecnológicos, tanto para pequeños emprendimientos como para cooperativas o PyMEs, como lo explica el sitio Oficial del Gobierno Argentino.²⁴

²³Copaipa <http://www.copaipa.org.ar/informatica/> [último acceso, Junio ,2019]

²⁴Busco Financiamiento <https://www.argentina.gob.ar/tema/emprender/financiamiento> [último acceso, Agosto ,2019]

Además, existen concursos propuestos por diferentes entidades, como el organizado por el Instituto Balseiro llamado IB50K ²⁵ con una suma en premios que superan los 50 mil dólares que permitirían cubrir los gastos de la inversión necesaria para el desarrollo del prototipo propuesto en este proyecto, de la misma manera, el concurso realizado por la empresa Endeavor ²⁶ tiene premios de hasta de 60 mil euros para proyectos que generen un impacto social o ambiental con el uso de la tecnología. También el Gobierno Nacional realiza una convocatoria para participar del concurso nacional INNOVAR ²⁷ en donde universitarios, emprendedores, empresas o investigadores pueden participar.

4.2. Aplicación de la Metodología Scrum

Scrum es una metodología ágil para gestionar proyectos que fue escogida por la experiencia obtenida en distintas materias durante el transcurso de la carrera y se basa en los siguientes principios según Martín Alaimo. ²⁸

- **Individuos e interacciones por sobre los procesos y herramientas:** Scrum se apoya en la confianza hacia las personas, sus interacciones y los equipos que toman la responsabilidad de realizar las tareas removiendo los impedimentos que se encuentren en su camino.
- **Software funcionando por sobre documentación exhaustiva:** Scrum requiere que al final de cada sprint se entregue un producto funcionando.
- **Colaboración con el cliente por sobre la negociación de contratos:** El cliente forma parte del equipo y trabaja colaborando para que el producto tenga un valor al final de cada iteración.
- **Respuesta al cambio por sobre el seguimiento de un plan:** Scrum se asegura que todo el mundo dentro de un equipo tenga toda la información necesaria para poder tomar decisiones en cualquier momento.

4.2.1. Roles

Kleer explica que intervienen tres roles en la metodología Scrum: el Product Owner, el equipo de desarrollo y el Scrum Master:

- **Product Owner:** Es la persona que se encarga de determinar la visión del producto, gestionar las expectativas de los clientes, recolectar los requerimientos y maximizar la rentabilidad del producto.
- **Equipo de desarrollo:** Está formado por los individuos que construyen el producto, son los únicos responsables por la calidad del mismo. En el equipo no existe un líder, cada miembro debe poseer las habilidades necesarias para realizar el trabajo requerido.

²⁵Concurso IB50K <http://www2.ib.edu.ar/ib50k/index.php> [último acceso, Agosto ,2019]

²⁶La Fundación everis busca el proyecto tecnológico más innovador de Argentina <https://www.endeavor.org.ar/la-fundacion-everis-busca-proyecto-tecnologico-mas-innovador-argentina/> [último acceso, Agosto ,2019]

²⁷Concurso Nacional de Innovaciones <https://mia.gob.ar/convocatorias/innovar> [último acceso, Agosto ,2019]

²⁸Alaimo, M. (2013). Proyectos ágiles con Scrum: flexibilidad, aprendizaje, innovación y colaboración en contextos complejos.

- **Scrum Master:** Es el líder del equipo, ayuda a que se alcance la mayor productividad posible. Además de proteger al equipo de distracciones y trabas externas al proyecto asegura la cooperación y comunicación.

En este proyecto el Scrum Master también adoptó el rol de Product Owner, por lo que no se aplicó la metodología de manera completa.

4.2.2. Product backlog

Contiene una lista con ítems que son considerados como historias de usuario y contienen la descripción de las características a construir del producto, además a cada uno se le asigna prioridades las cuales son cuantificadas usando la sucesión de Fibonacci, mientras más alto el valor mayor es su complejidad.

ID	Item	Prioridad
1	Visualización del consumo eléctrico en tiempo real	9
2	Visualización del consumo de agua en tiempo real	9
3	Cálculo de estadísticas a partir de los datos obtenidos	5
4	Visualización de registros por día, semana y mes	5
5	Visualización de consejos	3
6	Visualización de la información de aplicación	3

Tabla 10: Product backlog

4.2.3. Historias de usuario

Las historias de usuario son una representación de los requisitos que solicitan los usuarios, cada una cumple con un criterio de aceptación para determinar su terminación y además pueden contener requisitos no funcionales. El beneficio de usar historias de usuario es que permite mantener una relación cercana con los clientes, al conocer cuales son sus necesidades.

ID: 1	Nombre: Visualización del consumo eléctrico en tiempo real
Descripción: Como usuario necesito conocer la potencia que consume el hogar en tiempo real.	
Criterios de aceptación:	
<ul style="list-style-type: none"> • Se debe visualizar los datos obtenidos por el sensor de corriente que son la intensidad y la potencia de los componentes que se están midiendo. • Los datos se deben refrescar continuamente sin necesidad de volver a ingresar a la página. 	

Tabla 11: Historia de usuario N°1

ID: 2	Nombre: Visualización del consumo de agua en tiempo real
Descripción: Como usuario necesito ver el volumen de agua que estoy consumiendo durante el día.	
Criterios de aceptación:	
<ul style="list-style-type: none"> • Se debe visualizar los datos obtenidos por el caudalímetro, la cantidad de volumen de agua y la velocidad que fluye el caudal en litros por minuto. • Los datos se deben refrescar continuamente sin necesidad de volver a ingresar a la página. 	

Tabla 12: Historia de usuario N°2

ID: 3	Nombre: Cálculo de estadísticas a partir de los datos obtenidos
Descripción: Como usuario solicito saber el costo aproximado de la factura que vendrá el mes que viene a partir de los datos recaudados.	
Criterios de aceptación:	
<ul style="list-style-type: none"> • Calcular el costo del consumo del agua y la electricidad con los datos registrados mostrando el precio de los kilovatios-hora y los metros cúbicos de agua. 	

Tabla 13: Historia de usuario N°3

ID: 4	Nombre: Visualización de registros por día, semana y mes
Descripción: Como usuario me gustaría ver un registro de lo que gastó durante el día, la semana y hasta un mes.	
Criterios de aceptación:	
<ul style="list-style-type: none"> • Armar pestañas por cada intervalo de tiempo y que se actualicen cada vez que se presionen mostrando una animación. • Los datos deben estar en columnas separadas y teniendo como máximo dos decimales después de la coma y en el caso de la fecha tener un formato que sea legible. 	

Tabla 14: Historia de usuario N°4

ID: 5	Nombre: Visualización de consejos
Descripción: Como usuario me gustaría ver una página en donde se vean consejos para poder ahorrar energía eléctrica y reducir el gasto de agua.	
Criterios de aceptación:	
<ul style="list-style-type: none"> • Esta página debe ser la primera en abrirse cuando se inicia la aplicación. • Se deben agregar imágenes que acompañen a los consejos para una mejor presentación. 	

Tabla 15: Historia de usuario N°5

ID: 6	Nombre: Visualización de la información de aplicación
Descripción: Como usuario me gustaría ver una página en donde se pueda visualizar información acerca de la aplicación.	
Criterios de aceptación:	
<ul style="list-style-type: none"> • Esta página se debe ubicar en la última posición del menú. 	

Tabla 16: Historia de usuario N°6

4.2.4. Planificación de Sprints

El proyecto está planificado para que se desarrolle en 4 sprints con una duración de una semana cada uno. Un sprint es un conjunto de actividades que se deben cumplir en un determinado intervalo de tiempo y cada uno proporciona un producto potencialmente entregable.

- **Sprint 1:** Desarrollo de la visualización del consumo en tiempo real
- **Sprint 2:** Desarrollo de las estadísticas
- **Sprint 3:** Desarrollo de la visualización de registros
- **Sprint 4:** Desarrollo de las vistas con información

Sprint 1			
Tarea	Responsable	Duración	Estado
Planificación del Sprint	Product Owner	1 día	Finalizado
Desarrollo de los archivos php con las consultas a la base de datos	Programador 1	1 día	Finalizado
Desarrollo de las interfaz de “Tiempo real”	Programador 2	1 día	Finalizado
Desarrollo de la navegabilidad de la página de “Tiempo real”	Programador 1	1 día	Finalizado
Desarrollo del método para el caudalímetro que realiza la llamada a la API	Programador 2	1 día	Finalizado
Desarrollo del método para el sensor de corriente que realiza la llamada a la API	Programador 1	1 día	Finalizado
Verificación de resultados y testing	Programador 3	1 día	Finalizado
	Total	7 días	

Tabla 17: Sprint N°1

Sprint 2			
Tarea	Responsable	Duración	Estado
Planificación del Sprint	Product Owner	1 día	Finalizado
Desarrollo de los archivos php con las consultas a la base de datos	Programador 1	1 día	Finalizado
Desarrollo de las interfaz de “Estadística”	Programador 2	1 día	Finalizado
Desarrollo de la navegabilidad de la página de “Estadística”	Programador 1	1 día	Finalizado
Desarrollo del método que realiza la llamada a la API	Programador 2	2 días	Finalizado
Verificación de resultados y testing	Programador 3	1 día	Finalizado
	Total	7 días	

Tabla 18: Sprint N°2

Sprint 3			
Tarea	Responsable	Duración	Estado
Planificación del Sprint	Product Owner	1 día	Finalizado
Desarrollo de los archivos php con las consultas a la base de datos	Programador 1	1 día	Finalizado
Desarrollo de los archivos php	Programador 2	1 día	Finalizado
Desarrollo de las interfaces de “Consumo de agua” y “Consumo de electricidad”	Programador 1	1 día	Finalizado
Desarrollo de la navegabilidad de las páginas de “Consumo de agua” y “Consumo de electricidad”	Programador 2	1 día	Finalizado
Desarrollo de los métodos que realizan la llamada a la API	Programador 1	2 días	Finalizado
Verificación de resultados y testing	Programador 3	1 día	Finalizado
	Total	7 días	

Tabla 19: Sprint N°3

Sprint 4			
Tarea	Responsable	Duración	Estado
Planificación del Sprint	Product Owner	1 día	Finalizado
Desarrollo de los archivos php con las consultas a la base de datos	Programador 1	1 día	Finalizado
Desarrollo de los archivos php	Programador 2	1 día	Finalizado
Desarrollo de las interfaz de “Consejos” y “About”	Programador 1	1 día	Finalizado
Desarrollo de la navegabilidad de la página de “Consejos” y “About”	Programador 2	1 día	Finalizado
Desarrollo del método para el caudalímetro que realiza la llamada a la API	Programador 1	2 días	Finalizado
Verificación de resultados y testing	Programador 3	1 día	Finalizado
	Total	7 días	

Tabla 20: Sprint N°4

4.3. Desarrollo del proyecto

Este capítulo tiene el propósito de mostrar todo el proceso de desarrollo y experimentación del proyecto, primero comprendiendo el funcionamiento de cada componente hasta llegar a tener los suficientes conocimientos para poder cumplir con los objetivos planteados en la Definición del Problema.

4.3.1. Caudalímetro

El usado en este proyecto es un modelo YF-S201 que tiene tres cables de salida, cada uno con una funcionalidad diferente, el rojo que alimenta al dispositivo con una tensión de 5 voltios, el negro que realiza la conexión a masa y el amarillo que envía los pulsos que determina la cantidad de agua que pasa por el caudalímetro.

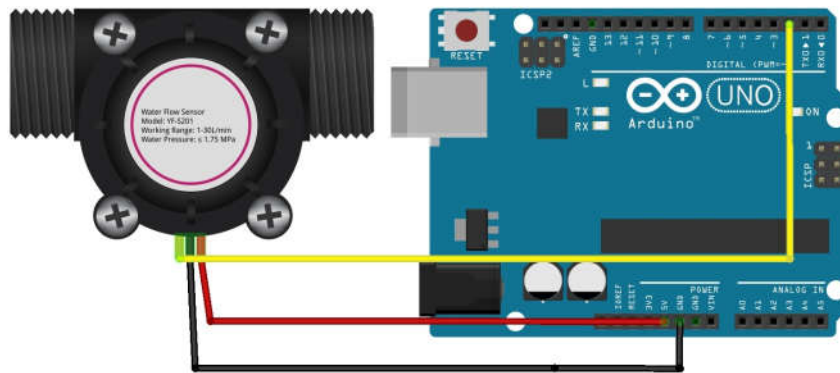


Figura 7: Diagrama de conexión del caudalímetro

El diagrama y el código que se usan para hacer funcionar el caudalímetro fueron extraídos del foro de Luis Llamas²⁹. Los pines del Arduino tienen diferentes funcionalidades, entre ellas tenemos los pines de interrupciones, que sirven para responder de forma inmediata a determinados eventos, en nuestro programa usamos el evento `RISING` el cual lanza una interrupción cuando los valores del pin varían de estados bajos a más altos, en nuestro caso lo aplicamos para calcular la frecuencia de la señal de salida para poder convertirlo en una unidad de caudal mediante un factor de conversión que nos da el fabricante del caudalímetro, el que empleamos es de 7,11 y lo representamos con la variable `K`. La velocidad del caudal en litros por minuto es directamente proporcional a la frecuencia e inversamente proporcional al factor de conversión.

$$f (Hz) = K \cdot Q (l/min) \Rightarrow Q (l/min) = \frac{f(Hz)}{K}$$

Representamos esta función mediante la siguiente línea de código, en donde frecuencia se obtiene de un método llamado `ObtenerFrecuencia()` que cuenta la cantidad de pulsos en un intervalo de un segundo.

```
1 float caudal_L_m=frecuencia / factor_conversion;
```

Para usar los pines de interrupciones en nuestro programa se debe usar la función `AttachInterrupt(pin, ISR, modo)` que es una función nativa del Arduino, en donde en el campo de pin se usa un valor ordinal, colocamos 0 si queremos utilizar el pin número 2 ó 1 si queremos usar el pin número 3. `ISR` es una abreviatura de *Interrupt Service Routine* y representa a la función o método que se llama cuando se produce la interrupción que es nuestro caso la usaremos para contar los pulsos y por último el modo es al evento que va a reaccionar el Arduino.

```
1 attachInterrupt(0, ContarPulsos , RISING);
```

Como el caudal es la variación del volumen con respecto al tiempo, podemos definir al volumen como la suma entre el volumen inicial y el producto entre el caudal por la variación del tiempo.

$$Q = \Delta V / \Delta t$$

$$V = V_0 + Q \Delta t$$

Y el volumen se calcula en el código con:

²⁹Medir caudal y consumo de agua con Arduino y caudalímetro. <https://www.luisllamas.es/caudal-consumo-de-agua-con-arduino-y-caudalimetro/> [último acceso, Mayo ,2019]

```
1 volumen=volumen+(caudal_L_m/60)*(dt/1000);
```

Una vez escrito el programa se procede a subirlo al Arduino como se muestra en el Anexo A.

En nuestra primera experiencia conectamos directamente el sensor al Arduino y no a una fuente de alimentación externa, además para simplificar el proceso se optó por soplar en la entrada del caudalímetro para simular una corriente de agua y para poder verificar que funciona, en toda la fase experimental se usó este método.

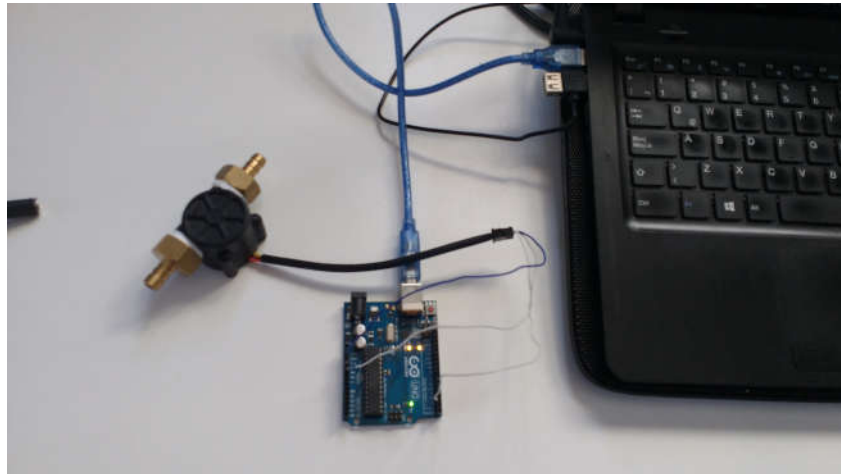


Figura 8: Primera conexión del caudalímetro con Arduino

Los resultados del sensor podemos verlos en el *Monitor Serie* que nos permite recibir y enviar mensajes de texto al Arduino, podemos acceder a esta funcionalidad desde la pestaña de herramientas del entorno de desarrollo. En la siguiente figura se ven dos columnas, la primera que muestra la velocidad del agua que transcurre en el caudalímetro y en la segunda el volumen estimado a partir del mismo, la explicación del programa cargado en el Arduino se puede encontrar en el Anexo B, los valores de la velocidad del caudal son nulos cuando sus hélices no se mueven, y tienen valores mayores a cero cuando el volumen total empieza a aumentar.

COM3 (Arduino/Genuino Uno)

Caudal: 0.000L/min	Volumen: 9.133 L
Caudal: 0.000L/min	Volumen: 9.133 L
Caudal: 0.000L/min	Volumen: 9.133 L
Caudal: 0.000L/min	Volumen: 9.133 L
Caudal: 1.125L/min	Volumen: 9.151 L
Caudal: 5.767L/min	Volumen: 9.248 L
Caudal: 5.204L/min	Volumen: 9.334 L
Caudal: 5.907L/min	Volumen: 9.433 L
Caudal: 2.532L/min	Volumen: 9.475 L
Caudal: 0.141L/min	Volumen: 9.477 L
Caudal: 0.000L/min	Volumen: 9.477 L
Caudal: 0.000L/min	Volumen: 9.477 L
Caudal: 0.000L/min	Volumen: 9.477 L
Caudal: 0.000L/min	Volumen: 9.477 L
Caudal: 0.000L/min	Volumen: 9.477 L

Figura 9: Mediciones del caudalímetro en Arduino

4.3.2. Sensor de corriente SCT-013

A diferencia del caudalímetro el sensor de corriente necesita un circuito integrado para poder funcionar correctamente, el que usamos es el LM358 que tiene una gran variedad de utilidades, entre ellas está la de amplificador operacional es decir, que la señal de salida del dispositivo puede ser cientos de veces mayor que la señal de entrada. Los datos de medición se envían a la entrada analógica cero (A0) del Arduino UNO que por dentro tiene un conversor analógico-digital que convierte la señal de entrada en un valor binario. El código usado para esta prueba se encuentra en el Anexo C y fue obtenido del foro de Luis Llamas.³⁰

Durante el proceso de experimentación se usó una placa protoboard que sirve para realizar pruebas colocando los componentes electrónicos en orificios que se encuentran conectados entre si y poder comprobar su funcionalidad sin tener que soldar las partes a un circuito impreso. Una vez terminada la etapa de experimentación se optó por crear uno para evitar que se produzcan falsos contactos en alguna conexión de cable o componente, además se reducen espacios y mejora la prolijidad que en la protoboard es difícil de conseguir, la construcción del mismo se detalló en la Solución Propuesta.

El diagrama de conexión se puede ver en la siguiente figura:

³⁰Sensor de corriente eléctrica no invasivo con Arduino y SCT-013. <https://www.luisllamas.es/arduino-sensor-corriente-sct-013/> [último acceso, Mayo ,2019]

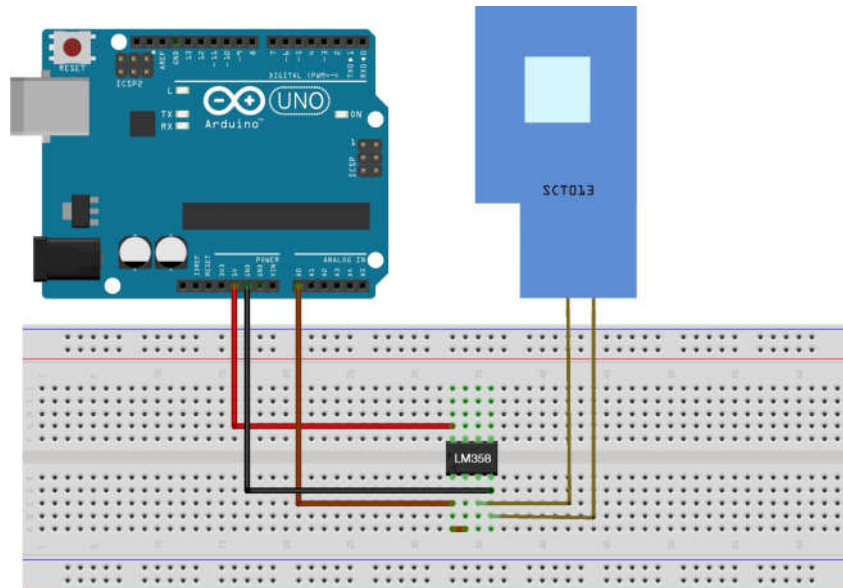


Figura 10: Diagrama de conexión del sensor de corriente

El sensor de corriente es de tipo no invasivo, es decir que no se necesita interrumpir la conexión para poder obtener información a partir de la misma, funciona por inducción electromagnética, por dentro tiene un núcleo ferromagnético que se une como una pinza, el cual permite enrollar el cable y poder obtener los valores de corriente.

El cable de salida del sensor de corriente tiene un conector jack de 3.5 milímetros por lo que se colocó una bornera que se conecta una extensión de cable con una ficha jack hembra para realizar la conexión y además se usó una fuente de alimentación externa diseñada para placas protoboard.

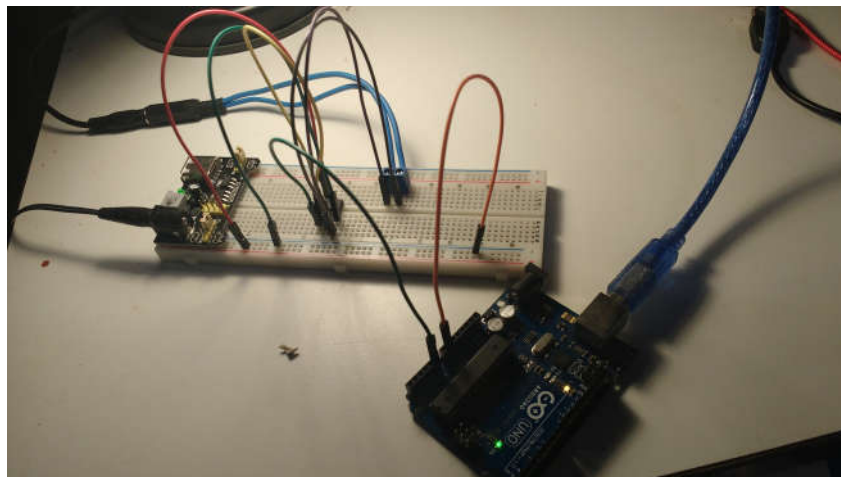


Figura 11: Conexión del sensor SCT-013

Para facilitar la conexión del sensor, se construyó una extensión de cable para evitar tener que desarmar el del dispositivo que íbamos a medir, que es el de una lámpara. Para esta prueba se usó una fuente de alimentación externa diseñada para ser usada en una placa protoboard.

4.3.3. Caudalímetro y sensor de corriente en conjunto

Luego de probar cada componente por separado se procedió a hacerlos funcionar a ambos con la placa Arduino y para lograrlo se tuvo que combinar los códigos de cada uno.

Hay que recordar que un programa de Arduino tiene dos métodos que son obligatorios que existan, los métodos *setup()* y *loop()* el primero sirve para establecer los criterios para la ejecución, por ejemplo, la velocidad de ejecución de la comunicación con el Arduino o para indicar cuales son los pines de entrada y de salida, y las líneas de código que se encuentren dentro del método *loop()* se van a ejecutar infinitamente.

Por lo tanto para combinar los códigos de ambos programas se tiene que juntar los métodos obligatorios en uno solo, antes de realizar esto se colocaron todas las variables tanto las del caudalímetro como las del sensor de corriente arriba del programa, los métodos para realizar los cálculos de frecuencia y de corriente se colocaron por debajo del método *loop()*.

La primera vez que se realizó la unión de los códigos surgieron errores ya que el código del caudalímetro para poder calcular la frecuencia del flujo utiliza una función llamada *interrupts()* y *noInterrupts()* los cuales sirven en el método para poder tomar una muestra de un segundo, estos métodos según la documentación de Arduino³¹ lo que hacen es deshabilitar y habilitar los pines de interrupción que tiene el Arduino, que son los 2 y 3 correspondientemente, también provoca que la comunicación entrante sea ignorada y además permiten realizar tareas en segundo plano por lo que las interrupciones pueden afectar al tiempo de ejecución del programa, al tener estas interrupciones provocaba una interferencia al calcular el valor de la frecuencia con el código siguiente para calcular el valor de la corriente, para solucionar este problema, coloqué los métodos de *Interrupts()* y *noInterrupts()* antes y después de ejecutar el método *get_corriente()* para que no sea interrumpido los cálculos realizados en este método.

Como se puede ver en la figura 16 tenemos las conexiones del circuito, el caudalímetro se conecta al pin 2, mientras que el sensor de corriente se conecta al pin analógico A0. La placa Arduino tiene que estar conectada a la masa de la placa protoboard para que funcione correctamente.

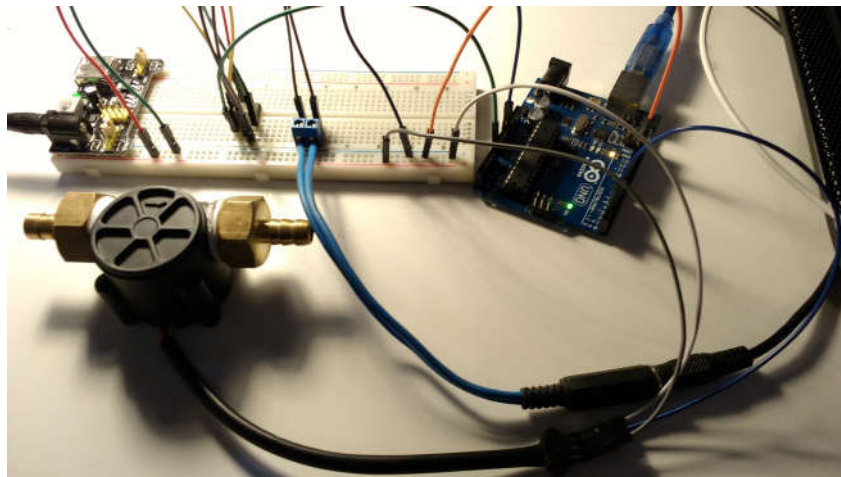


Figura 14: Imagen de las conexiones del caudalímetro y el sensor SCT

Una vez que el programa compilo y se subió correctamente, se pudo verificar su funcionamiento, revisando los valores que brindan los sensores. La primera verificación fue la de corriente, en donde se puede ver un valor de 71mA y una potencia de 15,560 vatios aproximadamente al tener una lámpara de 15 vatios.

³¹Documentación de Interrupts. <https://www.arduino.cc/en/Reference/Interrupts> [último acceso, Mayo ,2019]

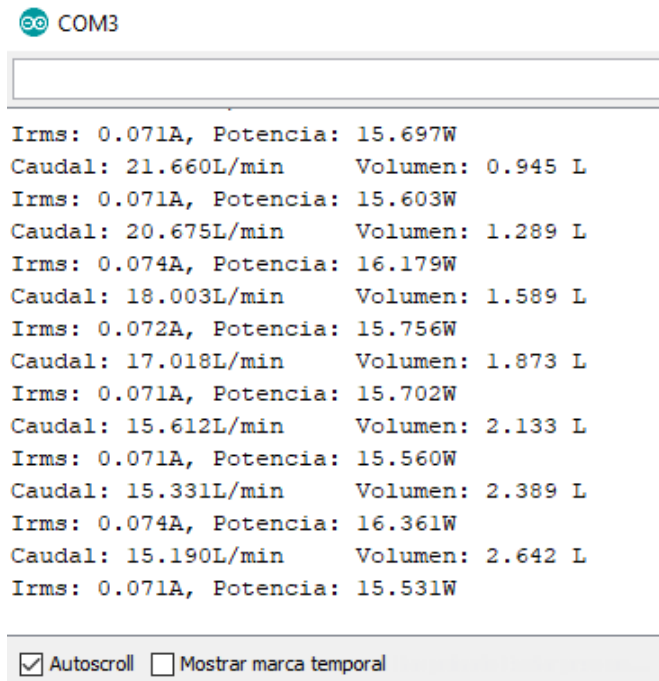


Figura 15: Muestra de datos con ambos dispositivos

4.3.4. Primer prototipo con AppInventor

Antes de conseguir el módulo esp8266 de wifi se realizó una aplicación con una herramienta online llamada AppInventor³² con el fin de pasar los datos de los sensores a un dispositivo móvil. Para tal fin se requirió de una fuente externa para alimentar a la placa Arduino, ya que el modelo de celular usado es un Moto G de tercera generación y no cuenta con la característica de poder abastecer a la placa Arduino por medio del cable OTG que requiere de 5 voltios.

La plataforma de desarrollo cuenta con una sección para construir la interfaz visual de la aplicación y otra para la construcción de la parte lógica en donde se establece el funcionamiento del mismo mediante la utilización de bloques, que representan sentencias de programación que brindan el comportamiento deseado.

En la figura 18 se puede ver el diseño realizado, en donde tenemos tres botones, el primero con el nombre de “Conectarse” que sirve para establecer y verificar la conexión de la placa Arduino, el segundo es el botón “Mostrar datos” que permite leer los datos que envía el Arduino y el tercer botón cierra la conexión con el Arduino.

³²MIT App Inventor 2 <http://ai2.appinventor.mit.edu/> [último acceso, Mayo ,2019]

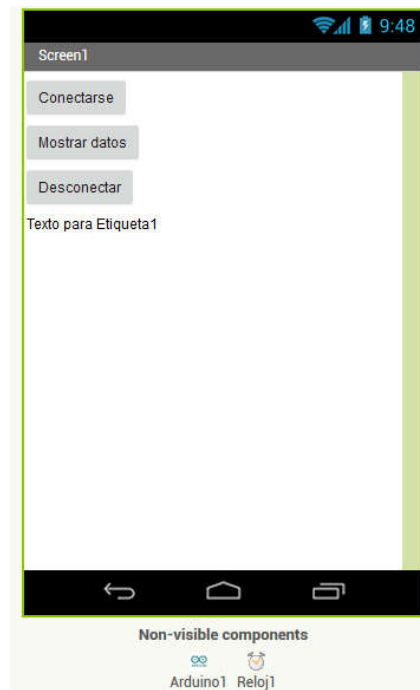


Figura 16: Primer diseño en AppInventor

En el diagrama de bloques el “Button1” hace referencia al botón de “Conectarse”, que luego de iniciar la conexión con el Arduino establece el Baudrate a 9600 que representa la velocidad de transmisión de los datos para tomar muestras y decodificar la señal.

El segundo botón, llamado como “Button2” hace referencia al botón de “Mostrar Datos” que en la estructura de bloques llama a la función *Read* para poder leerlo y los coloca dentro del atributo Text de un componente llamado “Etiqueta1”.

El tercer botón, denominado “Button3” que corresponde al botón de “Desconectar” cierra la conexión con la placa Arduino con la función *Close*.

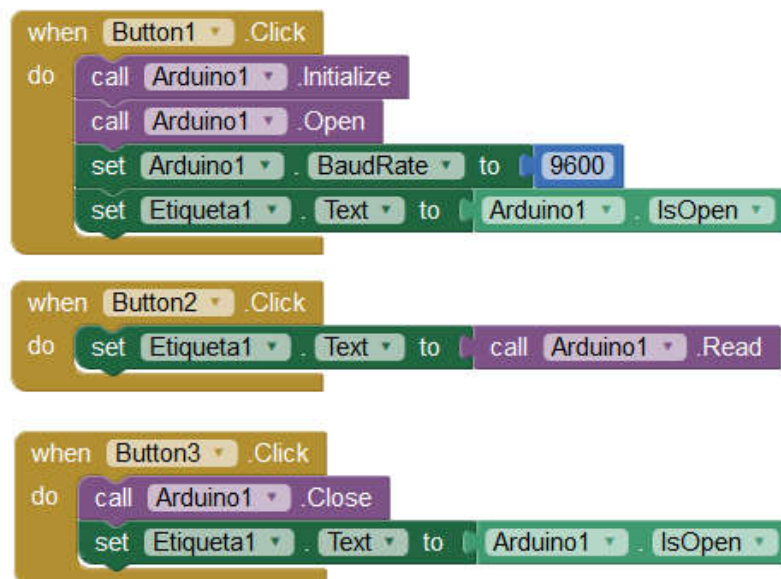


Figura 17: Estructura de bloques en AppInventor

Caudalímetro con cable OTG

Antes de leer los datos en el móvil por medio del cable OTG hay que instalar la aplicación creada en AppInventor, la explicación de todo este procedimiento se detalla en el Anexo G.

A continuación establecemos la conexión con el botón de conectar, una vez realizada esta acción se cambiará el valor que tiene la etiqueta a “true”, luego se presiona el botón de “Mostrar Datos” y veremos en la aplicación los datos como en el *Monitor Serial* de Arduino. En la primera imagen tenemos una velocidad y cantidad de litros de agua en cero, pero cuando hacemos funcionar el caudalímetro empezamos a ver los valores en la segunda imagen. La etiqueta al no ser un componente que tenga un Scroll, es decir, un desplazamiento hacia abajo tenemos que presionar nuevamente en “Mostrar Datos” para ver los nuevos resultados.

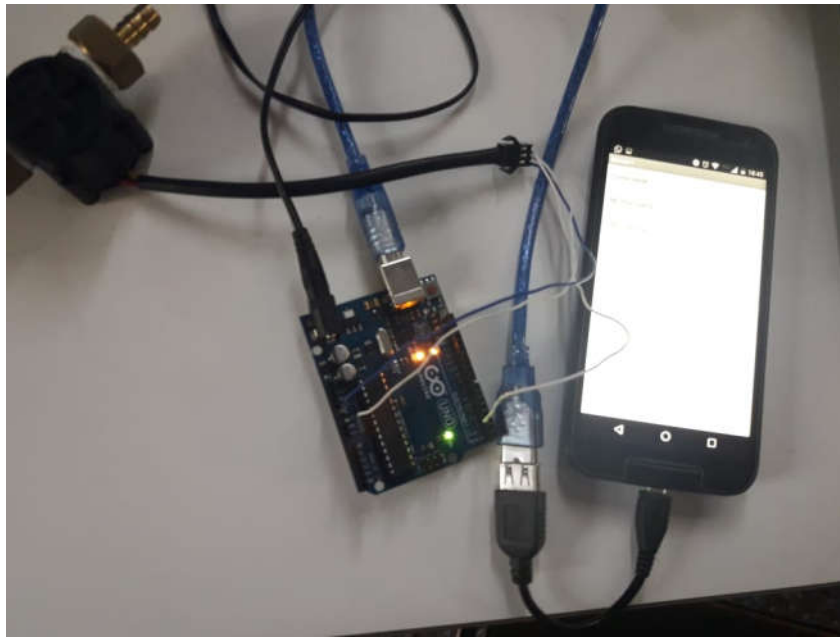


Figura 18: Conexión con el cable OTG al caudalímetro

Los resultados de la prueba se ven en la siguiente imagen, en donde se registra un volumen de agua de 0,3 litros aproximadamente.



Figura 19: Muestra de datos del caudalímetro con cable OTG

Sensor de corriente con cable OTG

Este programa sirve tanto para enviar los datos del caudalímetro como del sensor de corriente, ya que lee los valores por igual y el programa que los calcula se encuentran en el Arduino. En la primera muestra de datos de la figura tenemos la mayoría de los valores en 16 vatios de potencia y en la figura 22 se ven las conexiones realizadas.

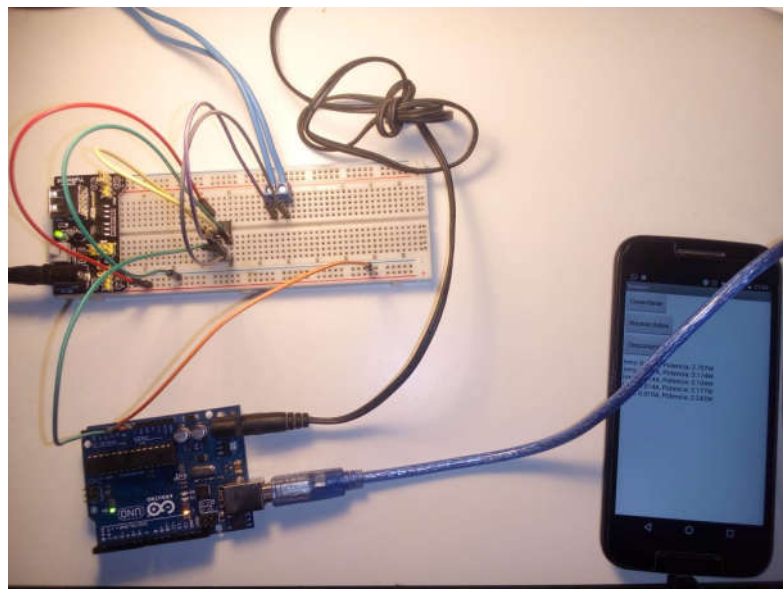


Figura 20: Conexión del cable OTG al sensor de corriente



Figura 21: Muestra de datos del sensor de corriente con cable OTG

4.3.5. Módulo esp8266

Para poder usar este componente se tiene que realizar la instalación de las librerías del mismo, este proceso se detalla en el Anexo E. Existen varios modelos de esp8266 el que se usa en este proyecto es el ESP-01, que tiene dos versiones una de color azul con una memoria flash de 512 Kb y la negra con 1Mb.

Existen 3 formas de conectar el esp8266, la primera es mediante la placa Arduino sin el microcontrolador que es el circuito integrado de color negro de mayor tamaño, en donde el programa se carga directamente al módulo como se ve en la 22, la segunda es con el microcontrolador del Arduino, en este caso para comunicarse tiene que usar una librería llamada *Software Serial* con la cual se indican los pines en donde se coloca el módulo, a diferencia de la conexión anterior no se necesita tener conectado el pin CPIO0 con la masa para realizar la subida de datos, la conexión se muestra en la 23. La tercera forma es mediante un adaptador USB-UART, que no usamos en este proyecto.

A diferencia del Arduino la velocidad de comunicación del módulo de WiFi puede resultar diferente, generalmente para este modelo usado corresponde usar 115200 baudios si se usa uno diferente los resultados pueden tener símbolos extraños que se denominan como basura o ruido, además el módulo trabaja con un voltaje de 3.3 voltios, usar una tensión de mayor amplitud puede provocar que el circuito se quemé y al ser un valor muy bajo se lo conecta al suministro de voltaje de la fuente para placa protoboard.

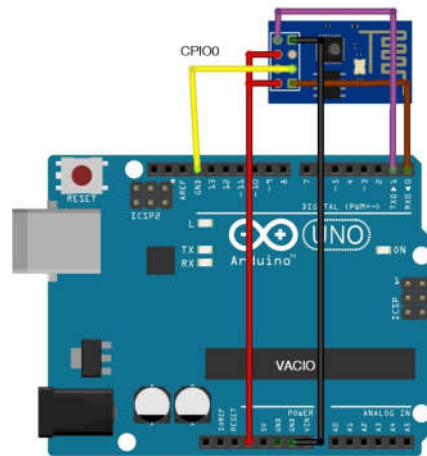


Figura 22: Diagrama de conexión del esp8266 sin el microcontrolador

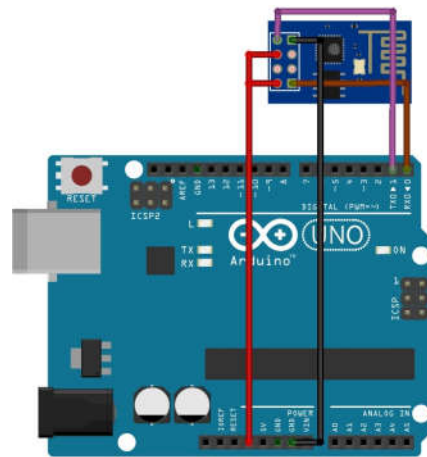


Figura 23: Diagrama de conexión del esp8266 con el microcontrolador

Comandos AT

Son instrucciones codificadas para comunicarse con el módulo de wifi u otro dispositivo, en la primera experiencia con el módulo se realizó la conexión sin el microcontrolador del Arduino y sin cargar algún programa se abrió el Monitor Serial seleccionando la opción “Ambos NL & CR” para poder enviar los comandos por medio de la caja.

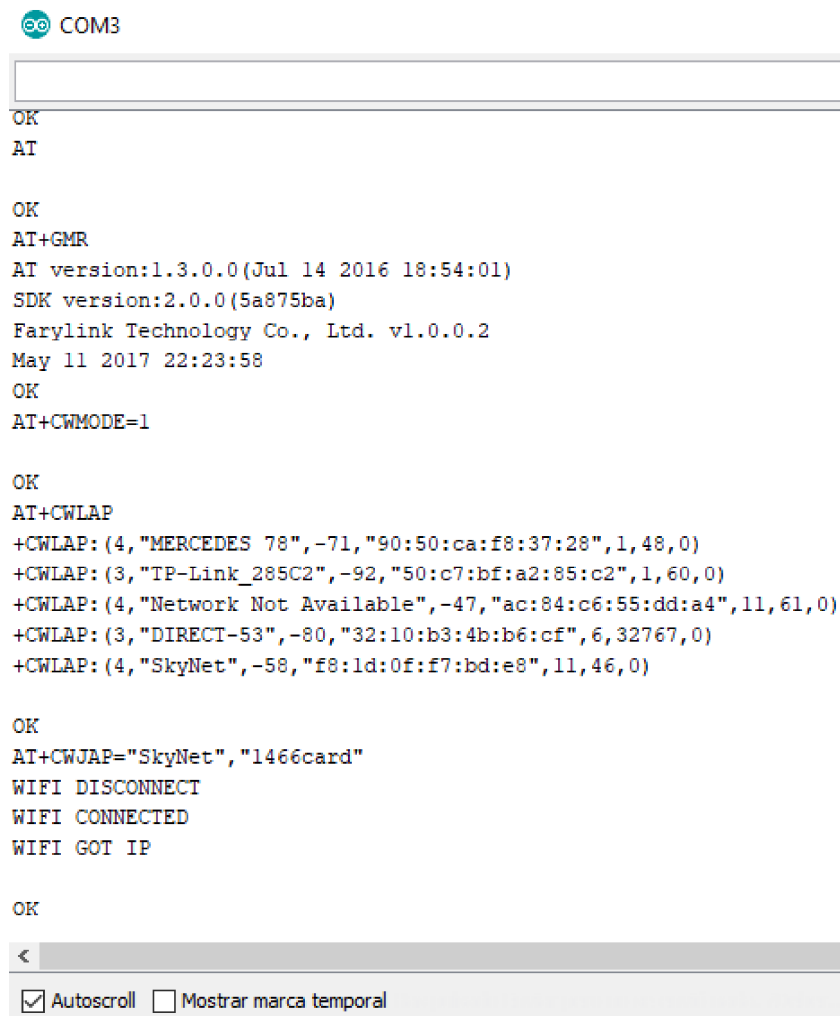
Los comandos tienen diferentes funcionalidades colocamos algunas en la siguiente tabla:

Comandos	Descripción
AT	Comprueba el estado del módulo
AT+RST	Reinicia el módulo
AT+GMR	Muestra la versión
AT+CWMODE	Cambia el modo
AT+CWLAP	Lista las redes cercanas
AT+CWJAP	Se conecta a una red

Tabla 21: Comandos AT

Enviándolas por el Monitor Serie tenemos los resultados de la figura 26, en donde logramos

conectarnos a la red llamada “SkyNet”, el modo número 1 del módulo es el modo de estación, es decir que será usado de cliente.



```

COM3
OK
AT

OK
AT+GMR
AT version:1.3.0.0 (Jul 14 2016 18:54:01)
SDK version:2.0.0 (5a875ba)
Farylink Technology Co., Ltd. v1.0.0.2
May 11 2017 22:23:58
OK
AT+CWMODE=1

OK
AT+CWMLAP
+CWLAP: (4, "MERCEDES 78", -71, "90:50:ca:f8:37:28", 1, 48, 0)
+CWLAP: (3, "TP-Link_285C2", -92, "50:c7:bf:a2:85:c2", 1, 60, 0)
+CWLAP: (4, "Network Not Available", -47, "ac:84:c6:55:dd:a4", 11, 61, 0)
+CWLAP: (3, "DIRECT-53", -80, "32:10:b3:4b:b6:cf", 6, 32767, 0)
+CWLAP: (4, "SkyNet", -58, "f8:1d:0f:f7:bd:e8", 11, 46, 0)

OK
AT+CWJAP="SkyNet", "1466card"
WIFI DISCONNECT
WIFI CONNECTED
WIFI GOT IP

OK

```

Figura 24: Resultados de los comandos AT

4.3.6. API

Una vez que se comprendió el funcionamiento de cada componente, se planteó la creación de una Web-API que es una API que se invoca mediante un protocolo HTTP, una API es un conjunto de reglas y especificaciones que las aplicaciones pueden seguir para comunicarse entre ellas como lo explica la publicadora TICbeat.³³

Creamos esta Web-API que a partir de aquí se nombrará simplemente como API empleando PHP y Mysql, para esto se necesitó usar un servidor Web llamado Apache que se lo escogió por ser gratuito y de código abierto, además de que es muy utilizado y existe soporte brindado por la comunidad. Como lo explica el sitio de Hosinger³⁴ Apache no es un servidor físico, sino que es un software que se ejecuta en un servidor. Un servidor es una aplicación capaz de atender a las peticiones de un cliente y devolverle una respuesta en concordancia. El trabajo del mismo es establecer una conexión entre un servidor y un navegador, mientras se envían información entre

³³¿Qué es una API y para qué sirve? <https://www.ticbeat.com/tecnologias/que-es-una-api-para-que-sirve/> [último acceso, Mayo ,2019]

³⁴¿Qué es Apache? Descripción completa del servidor web Apache. <https://www.hostinger.com.ar/tutoriales/que-es-apache/> [último acceso, Mayo ,2019]

ellos. Cuando una persona solicita información de un sitio, el mismo le devuelve una respuesta con los datos solicitados a través del protocolo HTTP, y apache se asegura de garantizar una comunicación fluida y segura.

Se usó un programa llamado Xampp, el cual es un paquete de software que gestiona bases de datos MySQL y un servidor apache con los intérpretes para lenguajes de script PHP y Perl, se decidió usar este software por la facilidad de uso y por la ventaja de no estar instalando cada componente por separado. Para poder gestionar la base de datos xampp provee de phpMyAdmin que está creado en lenguaje php y es open source, para acceder a esta herramienta se tiene que tener los servicios de la base de datos y del servidor apache e ingresar a la siguiente dirección: <http://localhost/phpMyAdmin>, con esta herramienta se crearon la base de datos y las tablas en donde se almacenaron los valores del caudalímetro y del sensor de corriente y se detalla el procedimiento en el Anexo F.

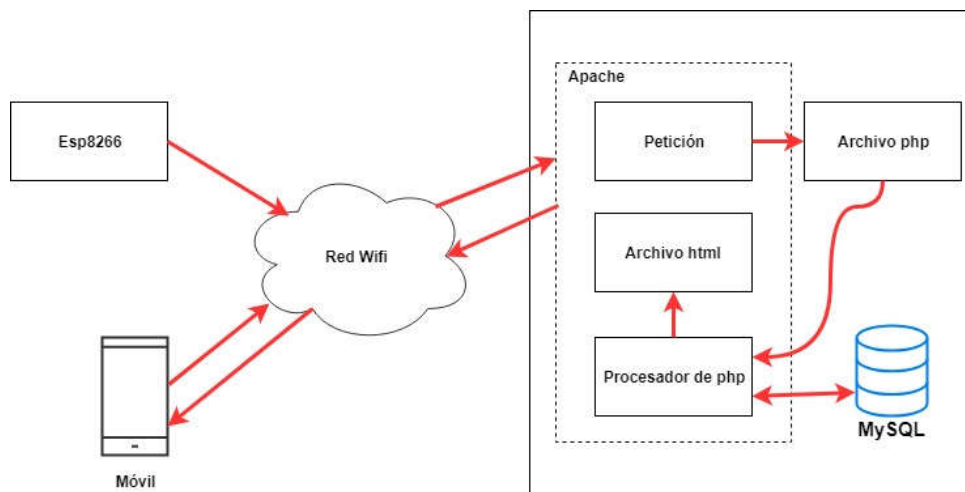


Figura 25: Diagrama de conexión general

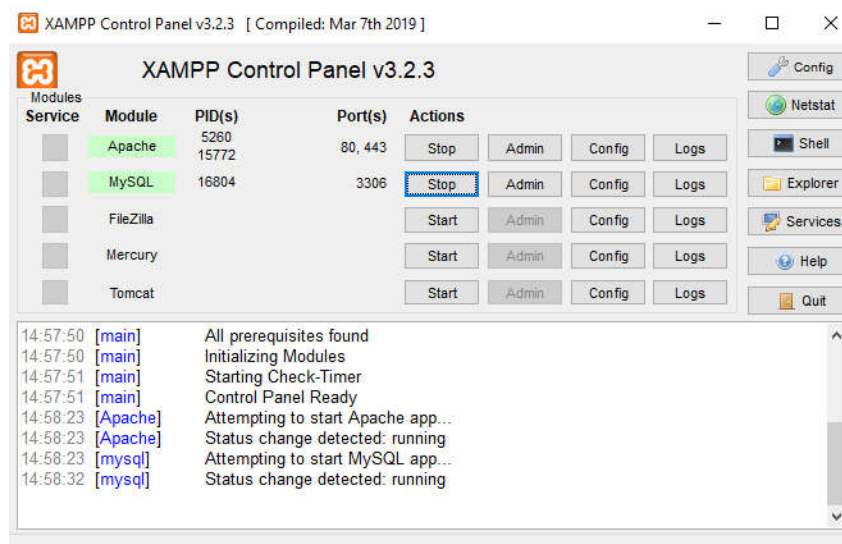


Figura 26: Interfaz del programa Xampp

Para poder realizar las peticiones a la API se usan archivos escritos en PHP, los cuales se realizan las consultas a la base de datos, así como también las operaciones de inserción, modificación y eliminación. Estos archivos se tienen que almacenar en la carpeta htdocs en la ubicación donde se instaló Xampp.

Archivo	Funcionalidad
quyelectricday.php	Consulta en la tabla de consumo eléctrico del día actual.
quyelectricmes.php	Consulta en la tabla de consumo eléctrico de los últimos 30 días
quyelectricsemana.php	Consulta en la tabla de consumo eléctrico de los últimos 7 días.
quypromelectric.php	Consulta del consumo promedio de los últimos 30 días.
quysumwater.php	Consulta del consumo total de agua de los últimos 30 días.
quywaterday.php	Consulta en la tabla del consumo de agua del día actual.
quywatermes.php	Consulta en la tabla del consumo de agua de los últimos 30 días.
quywatersemana.php	Consulta en la tabla del consumo de agua de los últimos 7 días.
realelectric.php	Consulta en la tabla de consumo eléctrico en tiempo real.
realwater.php	Consulta en la tabla de consumo de agua en tiempo real.
updatedata.php	Inserta los valores en las tablas de consumo de agua y electricidad.
updatedatareal.php	Actualiza la tabla de consumo y agua en tiempo real.

Tabla 22: Descripción de los archivos php

Cada archivo php realiza una conexión a la base de datos mediante la siguiente sentencia, en donde tenemos que especificar en qué base de datos nos vamos a conectar, la ip del servidor, cual es el usuario y la contraseña, usamos las que vienen por defecto.

```

1  define ("DB_SERVER", "192.168.0.16");
2  define ("DB_USER", "root");
3  define ("DB_PASSWORD", "root");
4  define ("DB_DATABASE", "arduino");
5  $conexion = mysqli_connect(DB_SERVER , DB_USER, DB_PASSWORD, DB_DATABASE
    ) or die ("Unable to connect to MySQL") ;

```

La consulta en la base de datos se realiza de la siguiente manera y transformar la respuesta en formato json, usamos este formato porque es mucho más fácil de leer a diferencia del xml. La aplicación móvil tendrá que deserializar este formato para poder leerlo. En este ejemplo traemos el consumo eléctrico del mismo día.

```

1  mysqli_query ($conexion ,DB_DATABASE);
2  $query="Select * from 'arduino'.'consumoelectrico' where DATE(fecha) =
    DATE(NOW())";
3  $resultado= $conexion->query ($query);
4
5  if ($resultado->num_rows > 0)
6  {
7      while ($r = mysqli_fetch_assoc ($resultado))
8      {
9          $rows []= $r;
10     }
11
12     echo json_encode ($rows);
13 }
14 else
15 {
16     $data->ok = false;
17     $data->error = "Error en el servidor, por favor intente
    nuevamente en unos minutos";
18     echo json_encode ($data);
19     exit ();
20 }

```

4.3.7. Conexión del esp8266 a la API

Para poder enviar datos a la API los archivos en formato php tendrán que recibir los parámetros, hay dos formas de enviarlos una es mediante el método GET en donde se transmite la información a través de la dirección URL, la estructura de la misma se puede ver a continuación:

http://localhost/Archivophp?parametro1="valor1"¶metro2="valor2"

Y la otra forma es mediante el método POST en donde se transmiten los datos por medio del protocolo HTTP, siendo los datos más difíciles de ver para el usuario. Usamos el método GET porque es mucho más simple y práctico para los fines del proyecto.

Para que los archivos php reciban parámetros necesitan de la siguiente sentencia:

```
1 $Variable1 = $_GET["variable1"]
2 $Variable2 = $_GET["variable2"]
```

Esto solo se aplica en los archivos de updatedata.php y updatedatareal.php que van a insertar los valores del caudalímetro y del sensor de corriente en la base de datos MySQL.

El código del archivo updatedata.php se lo puede ver en el Anexo K. Para que el módulo de wifi envíe los datos primero tenemos que conectarnos con los comandos AT a la red local, mediante el uso de la librería de *Software Serial* y establecer la velocidad de comunicación del módulo a 115200 baudios y la del Arduino a 9600 baudios. Como el método *Loop()* se ejecuta infinitamente no existía la forma de enviar datos a las tablas en tiempo real y a las tablas en donde se almacenan los datos cada media hora ya que se ejecutaban continuamente y una función de espera como la de *delay()* no era la solución. Para resolverlo se usó una librería que se obtuvo de GitHub ³⁵ para simular el uso de hilos, es decir, que se ejecuten de manera simultánea dos métodos. Para lograr este comportamiento se define el hilo al comienzo del programa y en el método de *setup()* se establece que el método de *senddata* se ejecutará cada 1.800.000 segundos, es decir cada media hora, luego en el método *loop()* preguntará constantemente si el hilo debe iniciarse, cuando pase el tiempo establecido se ejecutará.

```
1 thr.enabled = true;
2 thr.setInterval(1800000);
3 thr.onRun(senddata);
```

En el método *loop()* del programa:

```
1 interrupts();
2 senddatareal(velocidad, volumen, irms, potencia);
3 noInterrupts();
4 if(thr.shouldRun())
5 {
6   thr.run();
7 }
```

La diferencia del método *senddata()* y *senddatareal()* es que el primero no se puede recibir los parámetros, al usar la librería de hilos, por lo tanto dentro del mismo realizó los cálculos necesarios del caudalímetro y del sensor de corriente.

Dentro de los métodos mencionados por medio del se establece una conexión TCP al localhost por medio del comando AT, como se ve en el código a continuación:

³⁵Arduino Thread. <https://github.com/ivanseidel/ArduinoThread> [último acceso, Junio ,2019]

```

1 String cmd = "AT+CIPSTART=\"TCP\", \"";
2 cmd += "192.168.0.16";
3 cmd += "\",80";
4 espSerial.println(cmd);

```

En donde espSerial es el objeto que representa al módulo de WiFi, luego se envía la URL en donde hacemos la inserción de los datos:

```

1 String getRequest =
2 "GET " + uri;
3 getRequest += "?velocidad=";
4 getRequest += String(velocidad);
5 getRequest += "&volumen=";
6 getRequest += String(volumen);
7 getRequest += "&irms=";
8 getRequest += String(irms);
9 getRequest += "&potencia=";
10 getRequest += String(potencia);
11 getRequest += "\r\n\r\n";
12 delay(10);
13 cmd = "AT+CIPSEND=";
14 cmd += String(getRequest.length());
15 espSerial.println(cmd);
16 Serial.print(getRequest);

```

La variable uri representa la ubicación del archivo php que cargará los datos en la base de datos, en el caso de las tablas en tiempo real será el archivo updatedatareal.php y de las tablas restantes será la de updatedata(). De esta manera se envían los datos del caudalímetro y del sensor de corriente, siempre que el servidor apache y la base de datos se estén ejecutando.

Las conexiones de cada componente en particular no varía cuando se los une, solo ocurrió un inconveniente al juntar las masas de las fuentes de 5 y 3.3 voltios ya que dejaba de funcionar uno de los componentes, la solución fue colocar los cables negativos por separado, cerrando el circuito con los cables de masa de su correspondiente fuente.

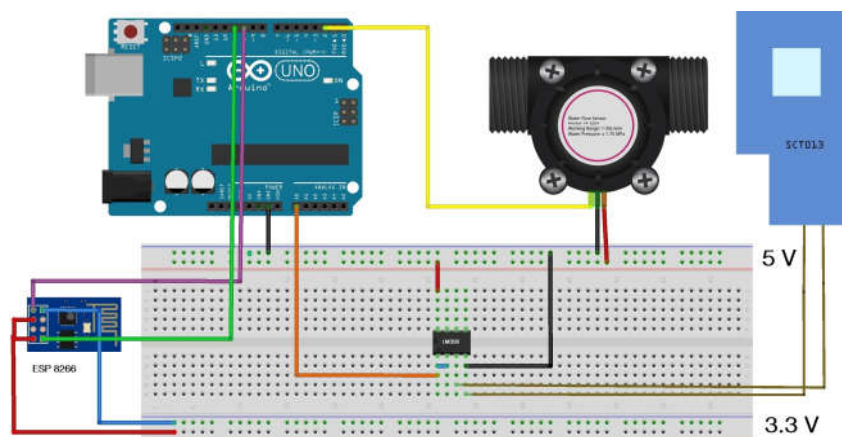


Figura 27: Diagrama del proyecto

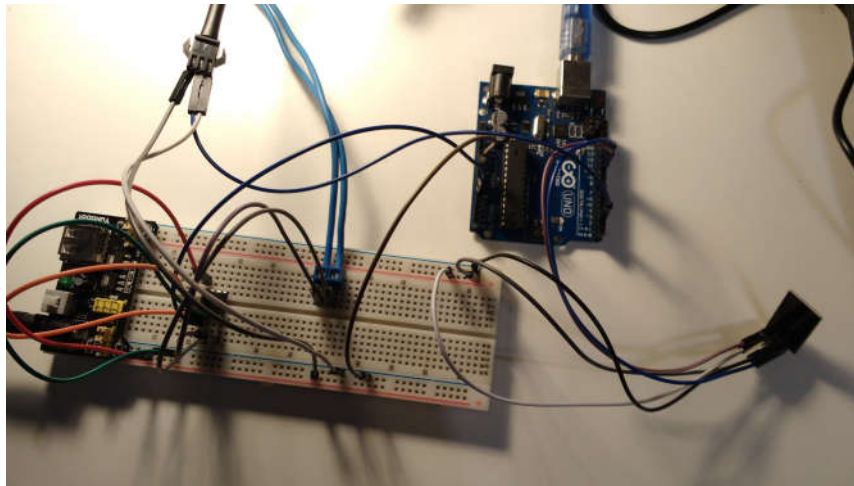


Figura 28: Conexión de todos los componentes en protoboard

Datos enviados

En las figuras 31 y 32 se observa lo que el programa muestra en el *Monitor Serial* los datos de los sensores y luego establece la conexión a la red local, luego se envía la sentencia GET y actualizar la base de datos.

```

Caudal: 0.000L/min      Volumen: 0.000 L
Irms: 0.067A, Potencia: 14.793W
AT+CIPSTART="TCP", "192.168.0.16", 80
GET /updatedata.php?velocidad=0.00&volumen=0.00&irms=0.07&potencia=14.79

Caudal: 0.000L/min      Volumen: 0.000 L
Irms: 0.067A, Potencia: 14.678W
AT+CIPSTART="TCP", "192.168.0.16", 80
GET /updatedata.php?velocidad=0.00&volumen=0.00&irms=0.07&potencia=14.68

Caudal: 0.000L/min      Volumen: 0.000 L
Irms: 0.067A, Potencia: 14.775W
AT+CIPSTART="TCP", "192.168.0.16", 80
GET /updatedata.php?velocidad=0.00&volumen=0.00&irms=0.07&potencia=14.77

Caudal: 0.000L/min      Volumen: 0.000 L
Irms: 0.066A, Potencia: 14.493W
AT+CIPSTART="TCP", "192.168.0.16", 80
GET /updatedata.php?velocidad=0.00&volumen=0.00&irms=0.07&potencia=14.49

Caudal: 16.034L/min      Volumen: 0.802 L
Irms: 0.067A, Potencia: 14.638W
AT+CIPSTART="TCP", "192.168.0.16", 80
GET /updatedata.php?velocidad=16.03&volumen=0.80&irms=0.07&potencia=14.64

Caudal: 31.927L/min      Volumen: 2.398 L
Irms: 0.067A, Potencia: 14.639W
AT+CIPSTART="TCP", "192.168.0.16", 80
GET /updatedata.php?velocidad=31.93&volumen=2.40&irms=0.07&potencia=14.64

Caudal: 21.941L/min      Volumen: 3.495 L
Irms: 0.067A, Potencia: 14.764W
AT+CIPSTART="TCP", "192.168.0.16", 80
GET /updatedata.php?velocidad=21.94&volumen=3.50&irms=0.07&potencia=14.76

```

Figura 29: Datos enviados desde esp8266 a la API

```
Caudal: 7.173L/min      Volumen: 3.854 L
Irms: 0.065A, Potencia: 14.402W
AT+CIPSTART="TCP", "192.168.0.16", 80
GET /updatedata.php?velocidad=7.17&volumen=3.85&irms=0.07&potencia=14.40

Caudal: 0.000L/min      Volumen: 3.854 L
Irms: 0.065A, Potencia: 14.312W
AT+CIPSTART="TCP", "192.168.0.16", 80
GET /updatedata.php?velocidad=0.00&volumen=3.85&irms=0.07&potencia=14.31

Caudal: 0.000L/min      Volumen: 3.854 L
Irms: 0.000A, Potencia: 0.000W
AT+CIPSTART="TCP", "192.168.0.16", 80
GET /updatedata.php?velocidad=0.00&volumen=3.85&irms=0.00&potencia=0.00

Caudal: 0.000L/min      Volumen: 3.854 L
Irms: 0.000A, Potencia: 0.000W
AT+CIPSTART="TCP", "192.168.0.16", 80
GET /updatedata.php?velocidad=0.00&volumen=3.85&irms=0.00&potencia=0.00

Caudal: 0.000L/min      Volumen: 3.854 L
Irms: 0.000A, Potencia: 0.000W
AT+CIPSTART="TCP", "192.168.0.16", 80
GET /updatedata.php?velocidad=0.00&volumen=3.85&irms=0.00&potencia=0.00

Caudal: 21.941L/min     Volumen: 4.585 L
Irms: 0.000A, Potencia: 0.000W
AT+CIPSTART="TCP", "192.168.0.16", 80
GET /updatedata.php?velocidad=21.94&volumen=4.59&irms=0.00&potencia=0.00

Caudal: 7.736L/min      Volumen: 4.972 L
Irms: 0.000A, Potencia: 0.000W
AT+CIPSTART="TCP", "192.168.0.16", 80
GET /updatedata.php?velocidad=7.74&volumen=4.97&irms=0.00&potencia=0.00
```

Figura 30: Segunda captura de datos enviados del esp8266

Si revisamos la base de datos se puede apreciar que si se guardaron los valores enviados desde el módulo de wifi, en la figura 33 se pueden ver la potencia de 14 vatios y la corriente de 0.07 amper que se almacenaron y en la figura 34 se puede ver el caudal de 21,94 litros por minuto con un volumen cercano a los 5 litros.

<input type="checkbox"/>	 Editar	 Copiar	 Borrar	260	0.07	14.75	2019-05-18 00:13:16
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	261	0.07	14.52	2019-05-18 00:13:19
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	262	0.07	14.75	2019-05-18 00:13:23
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	263	0.07	14.63	2019-05-18 00:13:27
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	264	0.07	14.72	2019-05-18 00:13:30
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	265	0.07	14.68	2019-05-18 00:13:46
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	266	0.07	14.77	2019-05-18 00:13:50
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	267	0.07	14.49	2019-05-18 00:13:54
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	268	0.07	14.64	2019-05-18 00:13:57
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	269	0.07	14.64	2019-05-18 00:14:01
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	270	0.07	14.76	2019-05-18 00:14:05
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	271	0.07	14.4	2019-05-18 00:14:09
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	272	0.07	14.31	2019-05-18 00:14:12
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	273	0	0	2019-05-18 00:14:16
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	274	0	0	2019-05-18 00:14:20
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	275	0	0	2019-05-18 00:14:23
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	276	0	0	2019-05-18 00:14:26
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	277	0	0	2019-05-18 00:14:30
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	278	0.07	14.41	2019-05-18 00:14:34
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	279	0.07	14.31	2019-05-18 00:14:37

Figura 31: Datos del sensor de corriente en MySQL





























































<input type="checkbox"/>	 Editar	 Copiar	 Borrar	281	0	0	2019-05-18 00:13:16
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	282	0	0	2019-05-18 00:13:19
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	283	0	0	2019-05-18 00:13:23
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	284	0	0	2019-05-18 00:13:27
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	285	0	0	2019-05-18 00:13:30
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	286	0	0	2019-05-18 00:13:46
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	287	0	0	2019-05-18 00:13:50
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	288	0	0	2019-05-18 00:13:54
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	289	16.03	0.8	2019-05-18 00:13:57
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	290	31.93	2.4	2019-05-18 00:14:01
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	291	21.94	3.5	2019-05-18 00:14:05
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	292	7.17	3.85	2019-05-18 00:14:09
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	293	0	3.85	2019-05-18 00:14:12
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	294	0	3.85	2019-05-18 00:14:16
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	295	0	3.85	2019-05-18 00:14:20
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	296	0	3.85	2019-05-18 00:14:23
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	297	21.94	4.59	2019-05-18 00:14:26
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	298	7.74	4.97	2019-05-18 00:14:30
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	299	0	4.97	2019-05-18 00:14:33
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	300	0	4.97	2019-05-18 00:14:37

Figura 32: Datos del caudalímetro en MySQL

4.3.8. Segundo prototipo de la aplicación

Para las tablas del caudalímetro y del sensor de corriente tenemos dos clases que tienen como propiedades las mismas columnas que se encuentran en la base de datos MySQL. Si rea-

lizamos una consulta a la API nos devolverá como respuesta un archivo en formato *json* como el siguiente:

```

▼ 0:
  id_agua:      "565"
  velocidad:    "0"
  volumen:      "0"
  fecha:        "2019-05-23 17:33:52"
▼ 1:
  id_agua:      "566"
  velocidad:    "0"
  volumen:      "0"
  fecha:        "2019-05-23 17:36:51"
▼ 2:
  id_agua:      "567"
  velocidad:    "0"
  volumen:      "0"
  fecha:        "2019-05-23 17:37:04"

```

Figura 33: Respuesta de la base de datos

Para poder recibir los datos de la API, se creó una clase llamada *Network* en donde tenemos todas las URLs de los archivos PHP, en donde tenemos las consultas a la base de datos, y por cada archivo tenemos un método en la aplicación que hace la llamada a la API cuando inicia la página y en el caso de los datos en tiempo real se llaman cada medio segundo para revisar si se actualizaron los datos.

```

public class Network
{
    const String URLwaterday = "http://192.168.0.16/querywaterday.php";
    const String URLwatersemana = "http://192.168.0.16/querywatersemana.php";
    const String URLwatermes = "http://192.168.0.16/querywatermes.php";
    const String URLelectricday = "http://192.168.0.16/queryelectricday.php";
    const String URLelectricsemana = "http://192.168.0.16/queryelectricsemana.php";
    const String URLelectricmes = "http://192.168.0.16/queryelectricmes.php";
    const String URLrealwater = "http://192.168.0.16/realwater.php";
    const String URLrealelectric = "http://192.168.0.16/realelectric.php";
    const String URLquerysumwater = "http://192.168.0.16/querysumwater.php";
    const String URLquerypromelectric = "http://192.168.0.16/querypromelectric.php";

    private HttpClient getClient()
    {
        HttpClient client = new HttpClient();
        client.DefaultRequestHeaders.Add("Accept", "application/json");
        client.DefaultRequestHeaders.Add("Connection", "close");
        return client;
    }
}

```

Figura 34: URLs de php

Se muestra uno de los métodos que realiza una llamada a la API, el mismo es asíncrono, es decir que se ejecuta en un segundo plano y trae el contenido de la URL en un string y luego deserializarlo en un tipo *IEnumerable* que implementa la clase Caudalímetro, es decir que cada fila del archivo json se corresponderá con la misma propiedad que le corresponde en . Si no trae los datos de manera satisfactoria del archivo php la aplicación devolverá un *Enumerable* vacío. Se requiere instalar el paquete *Newtonsoft* para deserializar el contenido.

```
public async Task<IEnumerable<Caudalimetro>> getDatosCaudalimetroDia()
{
    HttpClient client = getClient();
    var res = await client.GetAsync(URLwaterday);

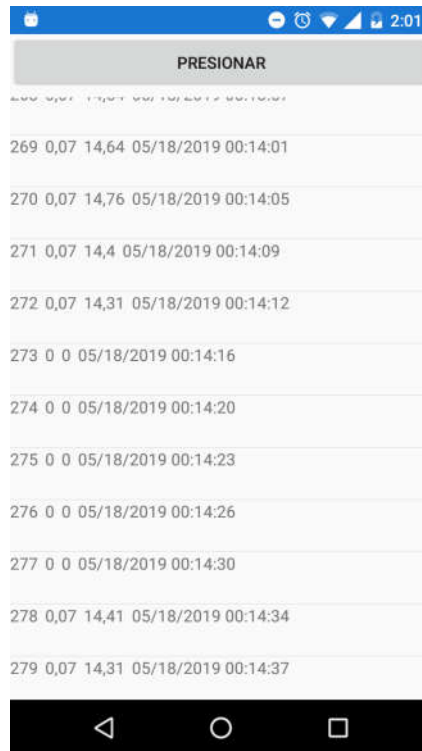
    if (res.IsSuccessStatusCode)
    {
        string content = await res.Content.ReadAsStringAsync();
        return JsonConvert.DeserializeObject<IEnumerable<Caudalimetro>>(content);
    }
    return Enumerable.Empty<Caudalimetro>();
}
```

Figura 35: Función de llamada a la API

La primera versión de la aplicación solo consistía en una página que contenía una lista y desplegaba la información al iniciarse. Los datos que se muestran en la figura 39 son los del caudalímetro y en la figura 38 se ven los del sensor de corriente. Son los mismos valores que se almacenaron en la base de datos MySQL.



Figura 36: Primera captura de los datos en android



The screenshot shows a mobile application interface with a blue header bar containing the text "PRESIONAR". Below the header is a list of data points, each on a separate row. The data points are as follows:

ID	Value 1	Value 2	Date	Time
269	0,07	14,64	05/18/2019	00:14:01
270	0,07	14,76	05/18/2019	00:14:05
271	0,07	14,4	05/18/2019	00:14:09
272	0,07	14,31	05/18/2019	00:14:12
273	0	0	05/18/2019	00:14:16
274	0	0	05/18/2019	00:14:20
275	0	0	05/18/2019	00:14:23
276	0	0	05/18/2019	00:14:26
277	0	0	05/18/2019	00:14:30
278	0,07	14,41	05/18/2019	00:14:34
279	0,07	14,31	05/18/2019	00:14:37

Figura 37: Segunda captura de los datos en android

4.3.9. Versión estable de la aplicación

Luego de realizar el segundo prototipo de la aplicación se crearon las siguientes interfaces de la versión estable, cada una con diferentes funciones, desde el menú se accede a las páginas que serán explicadas a continuación.

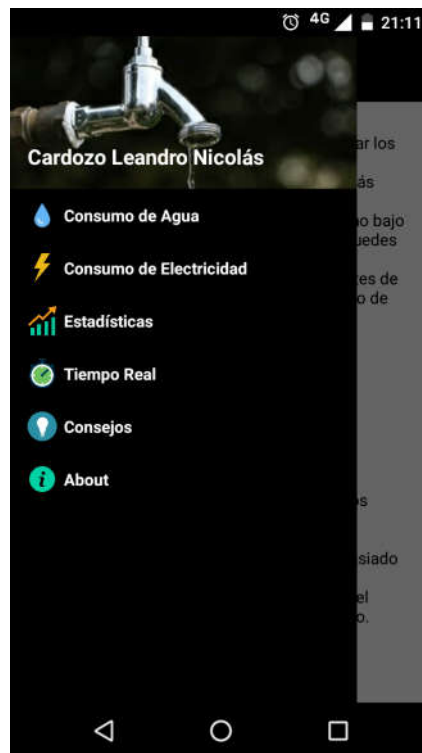


Figura 38: Menú de la aplicación

En la sección de “Consumo de Agua” y “Consumo de electricidad” se pueden ver los registros por día semana y mes, para la captura de pantalla se tomaron redujo el tiempo de muestra a 20 segundos, pero luego se revirtió a su configuración inicial para tomar los datos cada media hora.

Consumo de Agua			
HOY		SEMANA	MES
L/min	Litros	Fecha	
0	0,66	05:50 p. m. 23-05-19	
0	1,12	05:50 p. m. 23-05-19	
0,84	1,53	05:51 p. m. 23-05-19	
0,7	1,77	05:51 p. m. 23-05-19	
1,97	2,27	05:51 p. m. 23-05-19	
0	2,35	05:51 p. m. 23-05-19	
11,81	3,56	05:52 p. m. 23-05-19	
15,05	5,32	05:52 p. m. 23-05-19	
10,13	7,36	05:52 p. m. 23-05-19	
3,52	8,56	05:52 p. m. 23-05-19	

Figura 39: Tabla de consumo de agua

Consumo de Electricidad			
HOY		SEMANA	MES
Irms	Potencia	Fecha	
0,06	13,99	05:47 p. m. 23-05-19	
0,06	13,85	05:47 p. m. 23-05-19	
0,07	14,65	05:47 p. m. 23-05-19	
0,07	14,49	05:48 p. m. 23-05-19	
0,06	14,22	05:48 p. m. 23-05-19	
0,06	14,16	05:48 p. m. 23-05-19	
0,06	14,22	05:48 p. m. 23-05-19	
0,06	14,24	05:49 p. m. 23-05-19	
0,06	14,14	05:49 p. m. 23-05-19	
0,06	14,1	05:49 p. m. 23-05-19	

Figura 40: Tabla de consumo de electricidad

En la página de “Estadísticas” se calculó el precio estimado que se pagaría por consumir los servicios, en caso de la energía eléctrica obtenemos el costo de la Energía Activa y sin tener en cuenta otros impuestos como el IVA, la incidencia por el alumbrado público entre otros.

Se decidió que el caudalímetro comience a contar la cantidad de litros desde cero por cada día, y para poder obtener el total se realizó una consulta a la base de datos que traiga el máximo valor del volumen por cada día durante los últimos treinta días y a ese valor lo dividimos sobre mil para obtener la cantidad de metros cúbicos de agua que es la unidad usada por Aguas del Norte para calcular el precio a pagar, este valor varia dependiendo de la zona de la ciudad, en Barrio Mitre el costo es de \$8,3 por metros cúbicos consumidos. Por ejemplo, si solo tenemos 20 litros consumidos en el mes, equivalentes a 0,02 metros cúbicos, el costo del mismo es de 17 centavos, en cambio si el consumo es de 10.000 litros al mes tendremos un costo de 83 pesos.

Para calcular el precio estimado del consumo de electricidad se hace un promedio de todos los valores de potencia de los últimos treinta días y se los multiplica por veinticuatro y por treinta que es el tiempo de uso y lo dividimos sobre mil para obtener la cantidad de kilovatios por hora. Como tenemos un promedio de 14,7W, por usar solo una lámpara de 15W, la cantidad de kilovatios por hora es de 93 vatios y si lo multiplicamos por el precio del kilovatio por hora que varia dependiendo de la zona residencial, valor que aplica EDESA (Salta) a la energía activa (Barrio Mitre) cuyo costo es de \$3,57, nos da un total de \$39,04.



Figura 41: Página de estadísticas

En la página de “Tiempo Real” tenemos los valores de velocidad del caudal con el volumen y del sensor de corriente la intensidad con su respectiva potencia, estos datos se actualizan en intervalos de 3 a 6 segundos. Los valores mostrados en esta página se almacenan temporalmente en la base de datos, siempre se actualizan en una sola fila, ya que tenemos 4 tablas en total, dos para cada sensor, en la primera se actualizan los datos cada 30 minutos y en la segunda es casi instantáneo, estos datos no se almacenan en la base de datos para ahorrar espacio y no tener redundancia.

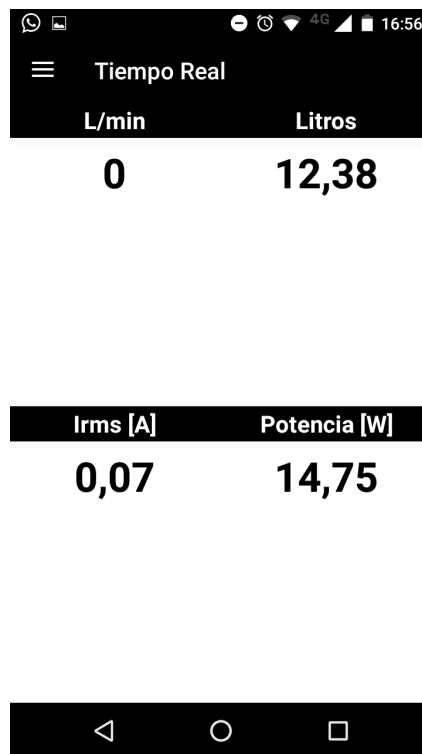


Figura 42: Página de datos en tiempo real

La pantalla de “About” de la aplicación solo tiene el propósito de dar información sobre el autor de la misma.

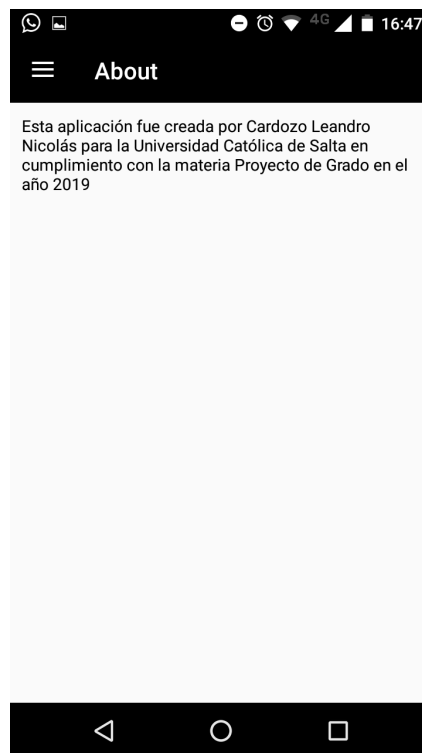


Figura 43: Página de About

Al iniciar la aplicación la primera página que se inicia es la de “Consejos”, que sirve como recordatorio para el usuario con el fin de reducir el consumo de agua y electricidad.

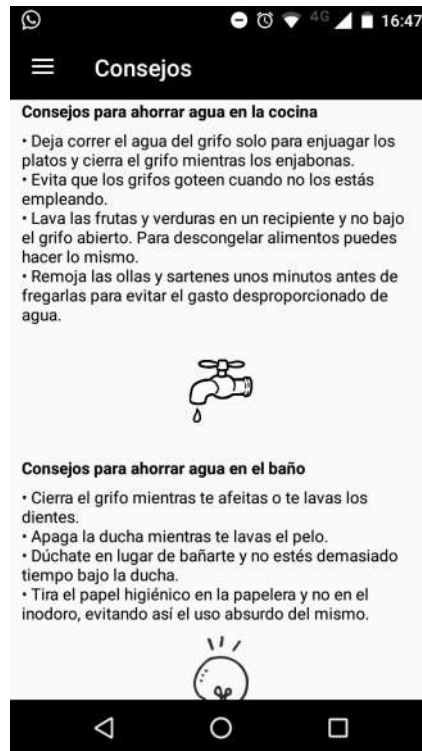


Figura 44: Primera página de consejos

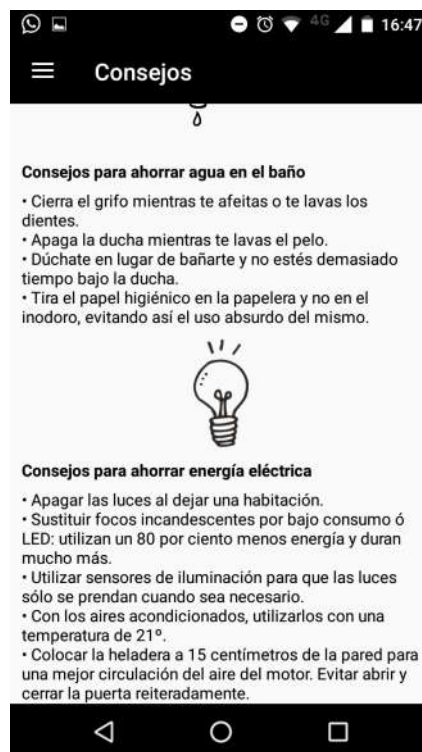


Figura 45: Segunda página de consejos

4.3.10. Finalización del proyecto

Al finalizar la creación del circuito eléctrico se montaron todos los componentes en un gabinete de metal para protegerlos de posibles daños externos y para una mejor presentación.

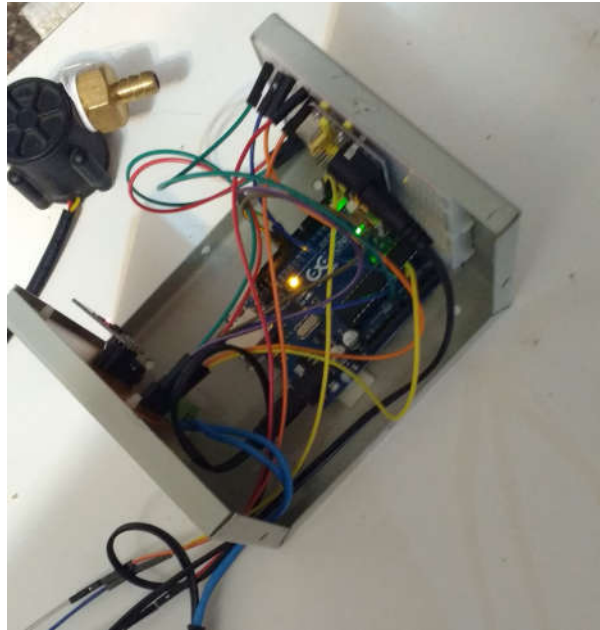


Figura 46: Componentes dentro del gabinete



Figura 47: Vista con el gabinete cerrado

Durante el desarrollo del proyecto la aplicación se instaló cada vez que se hacía una nueva prueba, por lo tanto para poder instalarla en otros dispositivos fue necesario crear el archivo con extensión apk para tal motivo. El proceso se encuentra en el Anexo J.

Capítulo 5

Resultados y verificación experimental

Una vez terminado el desarrollo de la aplicación y el armado del gabinete se realizaron pruebas para verificar el correcto funcionamiento del proyecto.

5.1. Caudalímetro

La primera verificación se hizo con el caudalímetro, en donde la prueba consistió en llenar una botella de 1,5 litros para compararlo con el valor que se registra en la aplicación.



Figura 48: Conexión al grifo

Se conectó el caudalímetro mediante una manguera flexible (para usos en sanitarios) al grifo y con una espiga de 1/2" con salida al recipiente (botella). Concluido el llenado del contenedor la lectura registrada en la aplicación fue de 1,58 litros, donde se puede apreciar un error mínimo, por lo tanto, el usuario puede confiar en los resultados que muestra la aplicación.

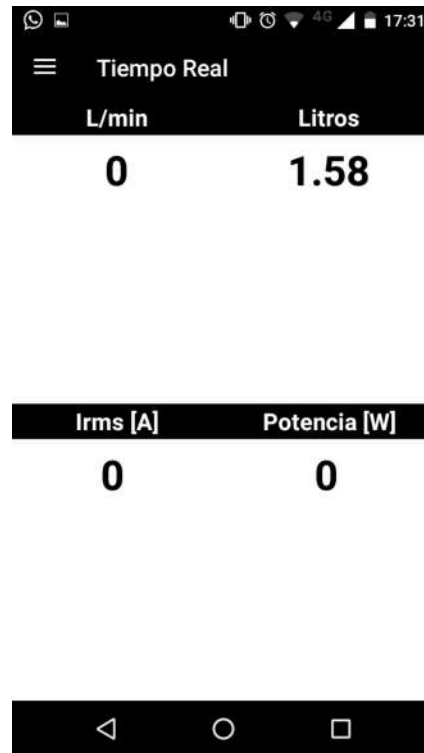


Figura 49: Resultado de la prueba de la botella

5.2. Sensor de corriente

La segunda verificación fue realizada con el sensor de corriente. Se conectó varios dispositivos eléctricos a una zapatilla: una lámpara, una notebook, un monitor y el cargador para alimentar al gabinete del circuito, el sensor se colocó en la extensión de cable que es intermedio entre el enchufe y la zapatilla eléctrica como se ve en la figura 52. La suma de potencia entre todos los componente es de 80 vatios por los valores que brindan sus fabricantes, en nuestra aplicación se registró un valor de 78 vatios que se aprecia en la figura 54, se tuvo un error de medición de 1.8 vatios producidos por el desgaste de los aparatos eléctricos. Se considera un resultado válido por lo que se da por concluida la verificación del sensor de corriente.

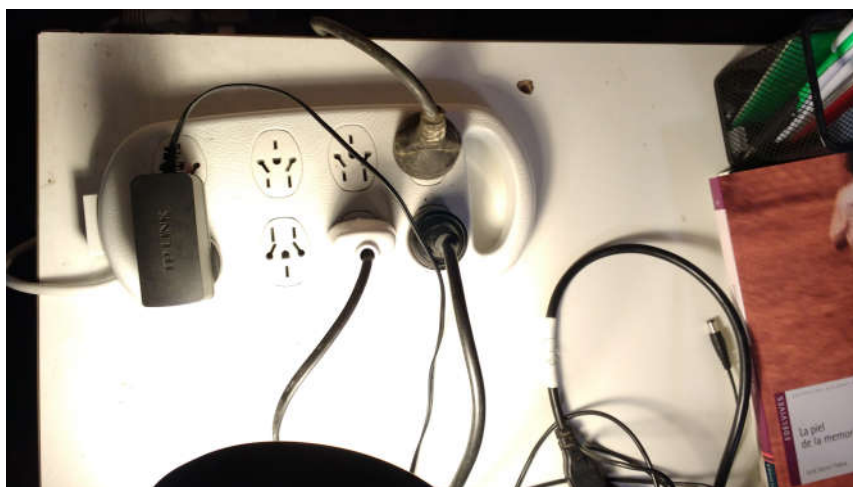


Figura 50: Vista de la zapatilla con los cuatro dispositivos



Figura 51: Conexión del sensor de corriente

Dispositivo	Potencia
Lámpara	15 W
Notebook	40 W
Monitor LCD	20 W
Cargador para el gabinete	5 W
Total	80 W

Tabla 23: Tabla de potencia de dispositivos

En la siguiente gráfica se puede ver la potencia por cada dispositivo eléctrico.

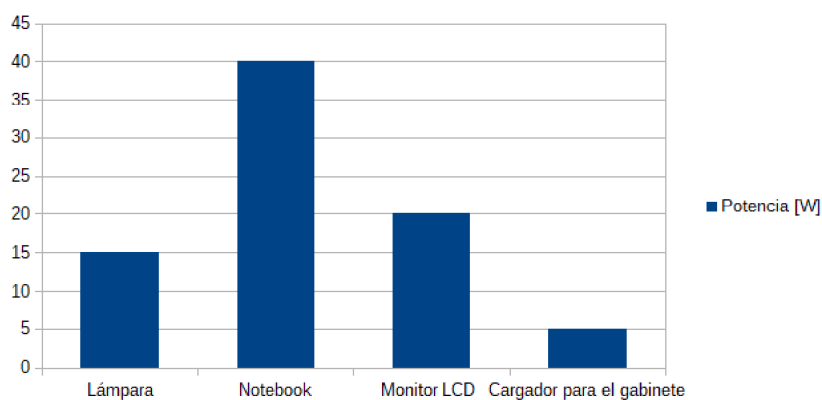


Figura 52: Gráfica de potencia de dispositivos

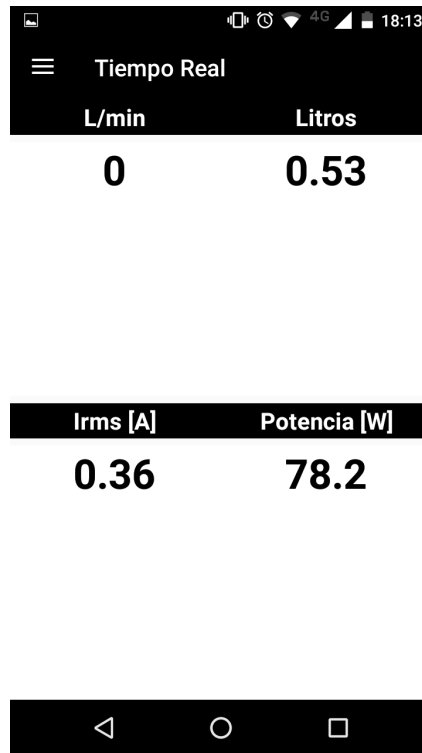


Figura 53: Potencia de los tres dispositivos en la aplicación

El beneficio que tiene el usuario de conocer los consumos en tiempo real es que puede detectar si existe una fuga en la instalación de agua, al verificar que todos los grifos están cerrados pero el volumen de la aplicación aumenta conforme pasa el tiempo. Con respecto a la corriente eléctrica, el usuario puede descubrir cuáles son los aparatos eléctricos que consumen más energía si apaga cada uno y comprueba, de esa manera, cuánto varía la potencia total en la aplicación.

5.3. Dificultades no previstas encontradas y sus soluciones

Durante la experiencia de usar el módulo se encontró un error de sincronización, este tipo de error sucede cuando hay un cable de algún componente eléctrico conectado a un pin no correspondiente de la placa (Arduino), también aparece cuando el módulo de wifi no está conectado al puerto asignado.

```

error: espcomm_upload_mem failed
El Sketch usa 258604 bytes (51%) del espacio de memoria
Las variables Globales usan 26696 bytes (32%) de la memoria
warning: espcomm_sync failed
error: espcomm_open failed
error: espcomm_upload_mem failed
error: espcomm_upload_mem failed

```

Figura 54: Mensaje de error de sincronización

Otro inconveniente que surgió en un principio fue que se tenía configurado el valor de *Crystal Frequency* a 40 Mhz, por lo tanto los resultados que se obtenían estaban llenos de ruidos y no se podía diferenciar las respuestas. Solo cuando se configuraba el módulo de wifi en 74.880 baudios

se podía diferenciar algo cuando en realidad no se usa esa velocidad, en el momento que se cambió el valor a 26 Mhz se corrigió este error y se logró leer las respuestas en 115.200 baudios como correspondía. Esta solución se encontró en un foro de GitHub ³⁶ en un problema con otro modelo de módulo de wifi. En una de las respuestas recomiendan usar el valor por defecto de 26 Mhz. La configuración de todos los parámetros se puede ver en la siguiente imagen:

```
Placa: "Generic ESP8266 Module"  
Upload Speed: "115200"  
CPU Frequency: "80 MHz"  
Crystal Frequency: "26 MHz"  
Flash Size: "1M (64K SPIFFS)"  
Flash Mode: "QIO (fast)"  
Flash Frequency: "40MHz"  
Reset Method: "ck"  
Debug port: "Disabled"  
Debug Level: "Ninguno"  
lwIP Variant: "v2 Lower Memory"  
VTables: "Flash"  
Exceptions: "Disabled"  
Builtin Led: "2"  
Erase Flash: "Only Sketch"  
Puerto: "COM3"  
Obtén información de la placa
```

Figura 55: Configuración de los parámetros del Arduino

También hubo inconvenientes realizando la conexión de todos los componentes del proyecto, se realizó una conexión de los cables negativos tanto del módulo de wifi y del sensor de corriente a la masa del Arduino, lo que generó un mal funcionamiento de ambos componentes, la solución a este problema fue conectar la masa del módulo de wifi a la masa de la fuente de energía de 3,3 volts.

³⁶NodeMCU ESP8266 12E - serial monitor garbage output. <https://github.com/esp8266/Arduino/issues/4005> [último acceso, Mayo ,2019]

Capítulo 6

Conclusiones

Los objetivos formulados en la Definición del Problema se lograron cumplir satisfactoriamente obteniendo así la transmisión de datos de los sensores a una aplicación en la plataforma Android. La utilización de la metodología Scrum facilitó el desarrollo del proyecto porque al finalizar cada sprint se logró obtener un avance en el proyecto.

A partir de este trabajo pueden surgir proyectos similares, por ejemplo, utilizando otros tipos de sensores como el de proximidad para informar sobre una actividad sospechosa o con un sensor de gas para detectar una fuga y enviar la correspondiente notificación al móvil.

En cuanto a la experiencia me sirvió mucho porque empecé experimentando con los componentes haciéndolos funcionar uno a la vez, luego realizando su integración y a pesar de que se encontraron dificultades se detectó la solución investigando y probando distintas alternativas.

Realizar la tesis ha resultado un complemento indispensable en mi educación como alumno, debido a que me permitió poder aplicar y ampliar los conocimientos teóricos y prácticos adquiridos durante la formación en la Universidad.

6.1. Futuras líneas de investigación

Una posible mejora al proyecto es usar una base de datos en la nube como lo es *Firebase*, que pertenece a Google, para poder acceder a los datos de la aplicación sin estar limitados por la red de wifi. De igual forma, con la utilización de *Webs sockets* podemos aumentar la performance utilizando una comunicación bidireccional con la API y no estar realizando llamadas a la misma a cada momento. Este proyecto podría implementarse en casas ecológicas para ayudar al cuidado de medio ambiente.

Bibliografía

A Nimbus Rises in the World of Cloud Computing. [https://cacm.acm.org/news/26083\ -a-nimbus-rises-in-the-world-of-cloud-computing/fulltext](https://cacm.acm.org/news/26083\-a-nimbus-rises-in-the-world-of-cloud-computing/fulltext) [último acceso, Junio, 2019]

Alaimo, M. (2013). Proyectos ágiles con Scrum: flexibilidad, aprendizaje, innovación y colaboración en contextos complejos.

Arduino, L. L. C. (2015). Arduino uno.

Arduino Thread. <https://github.com/ivanseidel/ArduinoThread> [último acceso, Junio, 2019]

Arquitectura de la plataforma. Android Developers <https://developer.android.com/guide/platform?hl=es-419> [último acceso, Agosto, 2018]

Busco financiamiento <https://www.argentina.gob.ar/tema/emprender/financiamiento> [último acceso, Agosto, 2019]

Concurso IB50K <http://www2.ib.edu.ar/ib50k/index.php> [último acceso, Agosto, 2019]

Concurso Nacional de Innovaciones <https://mia.gob.ar/convocatorias/innovar> [último acceso, Agosto, 2019]

Consumo de algunos electrodomésticos. http://www.edenor.com.ar/cms/SP/CLI/HOG/USO_consumo.html# [último acceso, Agosto, 2018]

Consumo de electricidad en el hogar y su impacto en la tarifa final. <http://www.fundelec.com.ar/informes/info0031.pdf> [último acceso, Agosto, 2018]

Copaipa <http://www.copaipa.org.ar/informatica/> [último acceso, Junio, 2019]

Darwin, I. F. (2017). Android Cookbook: Problems and Solutions for Android Developers. “O’Reilly Media, Inc.”.

Datasheet SCT-013-030. <http://statics3.seeedstudio.com/assets/file/bazaar/product/101990028-SCT-013-030-Datasheet.pdf> [último acceso, Junio, 2019]

Datasheet YF-S201. <https://www.hobbytronics.co.uk/datasheets/sensors/YF-S201.pdf> [último acceso, Junio, 2019]

Dio Connected Home <https://getdio.com/es/14-iluminacion-inteligente> [último acceso, Junio, 2019]

Documentación de Interrupts. <https://www.arduino.cc/en/Reference/Interrupts> [último acceso, Mayo, 2019]

La cantidad de agua domiciliaria, el nivel del servicio y la salud. <http://www.who.int/>

water_sanitation_health/diseases/wsh0302/es/ [último acceso, Agosto, 2018]

La falta de inversión en infraestructura impacta en el crecimiento. <https://www.lanacion.com.ar/economia/la-falta-inversion-infraestructura-impacta-crecimiento-nid\2174951> [último acceso, Agosto, 2018]

La Fundación everis busca el proyecto tecnológico más innovador de Argentina <https://www.endeavor.org.ar/la-fundacion-everis-busca-proyecto-tecnologico-mas-innovador-argentina/> [último acceso, Agosto, 2019]

Martínez, A. N., & Porcelli, A. M. (2015). Impactos de las tecnologías informáticas en el Ambiente y nuevas tendencias en computación verde. Diario DPI Suplemento Derecho y Tecnologías N°, 11-30.

Medir caudal y consumo de agua con Arduino y caudalímetro. <https://www.luisllamas.es/caudal-consumo-de-agua-con-arduino-y-caudalimetro/> [último acceso, Mayo, 2019]

MIT App Inventor 2 <http://ai2.appinventor.mit.edu/> [último acceso, Mayo, 2019]

Nimbus, el lavarropas sin agua que limpia hasta 20 kilos de ropa. <https://twenergy.com/ar/a/nimbus-el-lavarropas-sin-agua-que-limpia-hasta-20-kilos-de-ropa-2081> [último acceso, Junio, 2019]

NodeMCU ESP8266 12E - serial monitor garbage output. <https://github.com/esp8266/Arduino/issues/4005> [último acceso, Mayo, 2019]

Plan Nacional de Agua Potable y Saneamiento. https://www.argentina.gob.ar/sites/default/files/interior_agua_plan_agua_saneamiento.pdf [último acceso, Agosto, 2018]

Resultados del Relevamiento de Expectativas de Mercado (REM). <http://www.bcra.gov.ar/Pdfs/PublicacionesEstadisticas/REM180731%20Resultados%20web.pdf> [último acceso, Agosto, 2018]

Sensor de corriente eléctrica no invasivo con Arduino y SCT-013. <https://www.luisllamas.es/arduino-sensor-corriente-sct-013/> [último acceso, Mayo, 2019]

Sitio oficial de Hydrao Smart Shower. <https://www.hydrao.com/en/> [último acceso, Agosto, 2018]

Sitio oficial de Smappee https://www.smappee.com/eu_es/homepage [último acceso, Agosto, 2018]

Sitio Oficial de Wabee. <https://www.wabee.com.ar/> [último acceso, Agosto, 2018]

¿Qué es Apache? Descripción completa del servidor web Apache. <https://www.hostinger.com.ar/tutoriales/que-es-apache/> [último acceso, Mayo, 2019]

¿Qué es Arduino, cómo funciona y qué puedes hacer con uno? <https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno> [último acceso, Agosto, 2018]

¿Qué es la virtualización? <https://www.redhat.com/es/topics/virtualizati\on/what-is-virtualization> [último acceso, Junio, 2019]

¿Qué es teletrabajo? <https://www.uned.ac.cr/viplan/teletrabajo/que-es-tele\trabajo/que-es-teletrabajo> [último acceso, Junio, 2019]

¿Qué es una API y para qué sirve? <https://www.ticbeat.com/tecnologias/que-\es-una-api-para-que-sirve/> [último acceso, Mayo ,2019]

Anexos

Anexo A: Como subir un programa a Arduino

Los programas de Arduino constan de dos métodos, el método *setup()* en donde se definen configuraciones de ejecución, por ejemplo, que la velocidad de comunicación con el Arduino sea de 9600 baudios y el método *loop()* es en dónde el código se ejecuta infinitamente. Si alguno de estos métodos se encuentra ausente el programa no compila.

El primer paso es conectar el Arduino por medio del cable y verificar que el puerto este seleccionado.

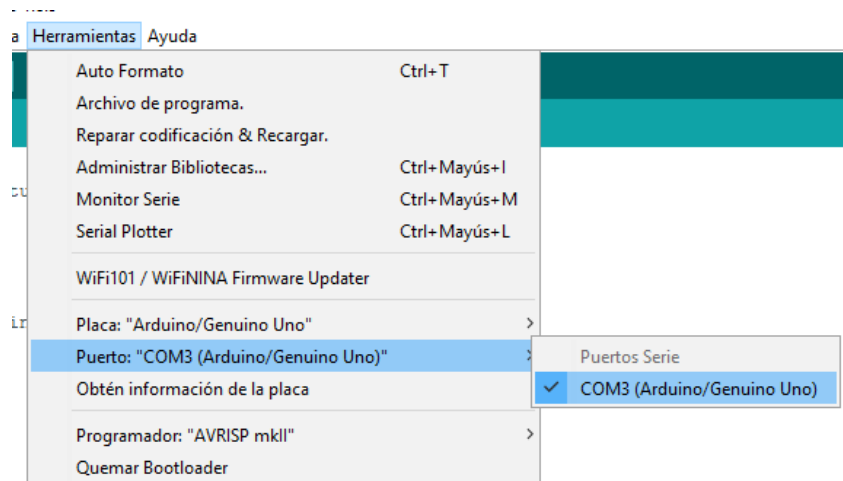


Figura 56: Selección de puerto

Para poder subir un programa se puede “verificar” el código presionando el botón como muestra la figura 59 y luego seleccionando el botón de “subir” como se ve en la figura 60, también se puede optar por presionar solamente el botón “subir” porque también cumple la funcionalidad de compilar el código. Al finalizar se muestra un mensaje de “subido” cuando se realizó el proceso de manera satisfactoria.

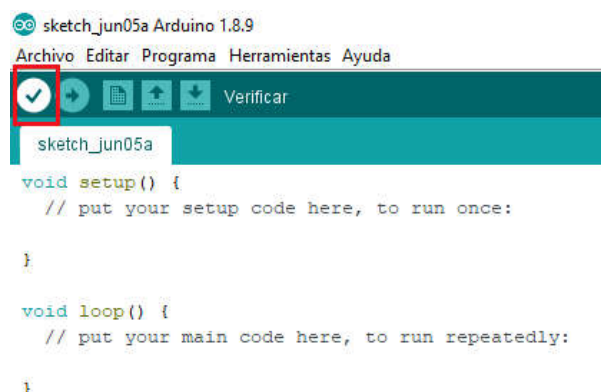


Figura 57: Botón para compilar en Arduino



Figura 58: Botón para subir el programa en Arduino

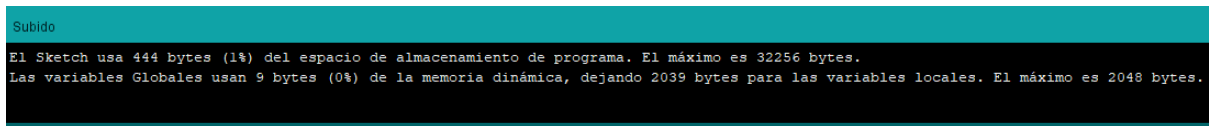


Figura 59: Mensaje de programa subido correctamente

Anexo B: Código del caudalímetro

```

1  volatile int NumPulsos; //variable para la cantidad de pulsos recibidos
2  int PinSensor = 2;      //Sensor conectado en el pin 2
3  float factor_conversion=7.11; //para convertir de frecuencia a caudal
4  float volumen=0;
5  long dt=0; //variación de tiempo por cada bucle
6  long t0=0; //millis () del bucle anterior
7
8  //---Función que se ejecuta en interrupción-----
9  void ContarPulsos ()
10 {
11  NumPulsos++; //incrementamos la variable de pulsos
12 }
13
14 //---Función para obtener frecuencia de los pulsos-----
15 int ObtenerFrecuecia ()
16 {
17  int frecuencia;
18  NumPulsos = 0; //Ponemos a 0 el número de pulsos
19  interrupts (); //Habilitamos las interrupciones
20  delay (1000); //muestra de 1 segundo
21  noInterrupts (); //Deshabilitamos las interrupciones
22  frecuencia=NumPulsos; //Hz(pulsos por segundo)
23  return frecuencia;
24 }
25
26 void setup ()
27 {
28
29  Serial.begin(9600);
30  pinMode(PinSensor , INPUT);
31  attachInterrupt(0,ContarPulsos ,RISING); //(Interrupción 0(Pin2),función,
    Flanco de subida)
32  Serial.println ("Envie 'r' para restablecer el volumen a 0 Litros");
33  t0=millis ();
34 }
35
36 void loop ()
37 {
38  if (Serial.available ()) {
39  if (Serial.read()=='r')volumen=0;//restablecemos el volumen si recibimos
    'r'
40 }
41 float frecuencia=ObtenerFrecuecia (); //obtenemos la frecuencia de los
    pulsos en Hz
42 float caudal_L_m=frecuencia/ factor_conversion; //calculamos el caudal en
    L/m
43 dt=millis ()-t0; //calculamos la variación de tiempo
44 t0=millis ();
45 volumen=volumen+(caudal_L_m/60)*(dt/1000); // volumen(L)=caudal(L/s)*
    tiempo (s)
46
47 //-----Enviamos por el puerto serie-----
48 Serial.print ("Caudal: ");
49 Serial.print (caudal_L_m,3);
50 Serial.print ("L/min\tVolumen: ");
51 Serial.print (volumen,3);

```

```
52 Serial.println (" L");  
53 }
```

Anexo C: Código del sensor SCT

```

1
2 void setup() {
3
4   Serial.begin(9600);
5   analogReference(INTERNAL);
6   //analogReference(INTERNAL1V1); //solo Arduino Mega
7   }
8
9 void loop() {
10
11  float Irms=get_corriente(); //Corriente eficaz (A)
12  float P=Irms*220.0; // P=IV (Watts)
13
14  Serial.print("Irms: ");
15  Serial.print(Irms,3);
16  Serial.print("A, Potencia: ");
17  Serial.print(P,3);
18  Serial.println("W");
19  //delay(100);
20  }
21
22 float get_corriente()
23 {
24  float voltajeSensor;
25  float corriente=0;
26  float Sumatoria=0;
27  long tiempo=millis();
28  int N=0;
29  while(millis()-tiempo<500)//Duración 0.5 segundos(Aprox. 30 ciclos de 60
    Hz)
30  {
31  voltajeSensor = analogRead(A0) * (1.1 / 1023.0);////voltaje del sensor
32  corriente=voltajeSensor*30.0; //corriente=VoltajeSensor*(30A/1V)
33  Sumatoria=Sumatoria+sq(corriente);//Sumatoria de Cuadrados
34  N=N+1;
35  delay(1);
36  }
37  Sumatoria=Sumatoria*2;//Para compensar los cuadrados de los semiciclos
    negativos.
38  corriente=sqrt((Sumatoria)/N); //ecuación del RMS
39  return(corriente);
40  }

```

Anexo D: Código del caudalímetro y del sensor SCT

```

1  volatile int NumPulsos; //variable para la cantidad de pulsos recibidos
2  int PinSensor = 2;      //Sensor conectado en el pin 2
3  float factor_conversion=7.11; //para convertir de frecuencia a caudal
4  float volumen=0;
5  long dt=0; //variación de tiempo por cada bucle
6  long t0=0; //millis() del bucle anterior
7
8  void setup() {
9  Serial.begin(9600);
10 analogReference(INTERNAL);
11 pinMode(PinSensor, INPUT);
12 attachInterrupt(0, ContarPulsos, RISING); //(Interrupción 0(Pin2), función,
    Flanco de subida)
13 //Serial.println("Envie 'r' para restablecer el volumen a 0 Litros");
14 t0=millis();
15 }
16
17 void loop() {
18 if (Serial.available()) {
19 if (Serial.read()=='r') volumen=0; //restablecemos el volumen si recibimos
    'r'
20 }
21
22 float frecuencia=ObtenerFrecuecia();
23 float caudal_L_m=frecuencia/factor_conversion; //calculamos el caudal en
    L/m
24 dt=millis()-t0; //calculamos la variación de tiempo
25 t0=millis();
26 volumen=volumen+(caudal_L_m/60)*(dt/1000); // volumen(L)=caudal(L/s)*
    tiempo(s)
27 Serial.print("Caudal: ");
28 Serial.print(caudal_L_m,3);
29 Serial.print("L/min\tVolumen: ");
30 Serial.print(volumen,3);
31 Serial.println(" L");
32 interrupts();
33 float Irms=get_corriente(); //Corriente eficaz (A)
34 noInterrupts();
35 float P=Irms*220.0; // P=IV (Watts)}
36 Serial.print("Irms: ");
37 Serial.print(Irms,3);
38 Serial.print("A, Potencia: ");
39 Serial.print(P,3);
40 Serial.println("W");
41 }
42
43 void ContarPulsos ()
44 {
45 NumPulsos++; //incrementamos la variable de pulsos
46 }
47
48 int ObtenerFrecuecia()
49 {
50 int frecuencia;
51 NumPulsos = 0; //Ponemos a 0 el número de pulsos
52 interrupts(); //Habilitamos las interrupciones

```

```
53 delay(1000); //muestra de 1 segundo
54 noInterrupts(); //Deshabilitamos las interrupciones
55 frecuencia=NumPulsos; //Hz(pulsos por segundo)
56 return frecuencia;
57 }
58
59 float get_corriente ()
60 {
61 float voltajeSensor;
62 float corriente=0;
63 float Sumatoria=0;
64 long tiempo=millis();
65 int N=0;
66 while(millis()-tiempo<500)//Duración 0.5 segundos(Aprox. 30 ciclos de 60
    Hz)
67 {
68 voltajeSensor = analogRead(A0) * (1.1 / 1023.0);//// voltaje del sensor
69 corriente=voltajeSensor*30.0; // corriente=VoltajeSensor*(30A/1V)
70 Sumatoria=Sumatoria+sq(corriente);//Sumatoria de Cuadrados
71 N=N+1;
72 // delay(1);
73 }
74 Sumatoria=Sumatoria*2;//Para compensar los cuadrados de los semiciclos
    negativos.
75 corriente=sqrt((Sumatoria)/N); //ecuación del RMS
76 return(corriente);
77 }
```

Anexo E: Proceso de configuración del módulo esp8266

Primero se debe instalar las librerías del módulo, que se pueden encontrar en github. <https://github.com/esp8266/Arduino>.

Se debe colocar en el apartado de preferencias la siguiente URL: http://arduino.esp8266.com/stable/package_esp8266com_index.json para que se puedan descargar los paquetes necesarios y usar el módulo esp8266, como se muestra en la figura 62.

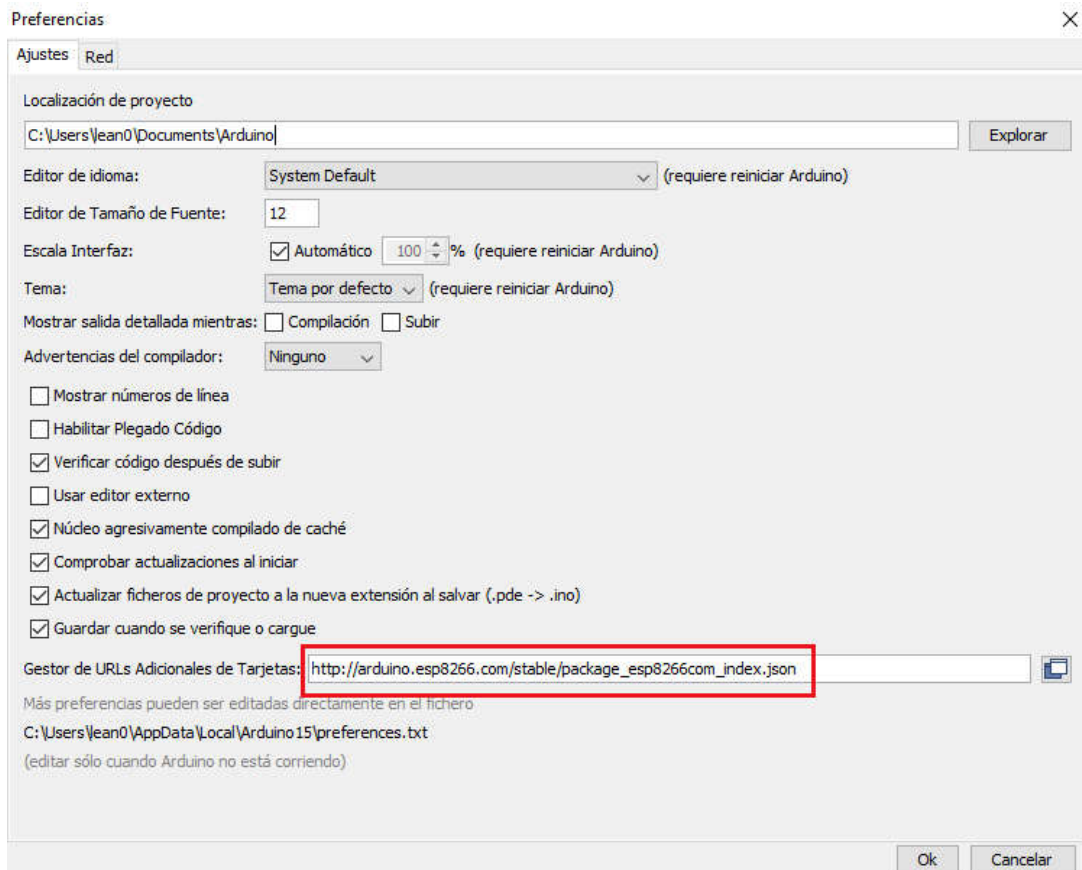


Figura 60: Preferencias en Arduino IDE

Una vez que se tiene la url configurada se debe ir a "Herramientas>Placa Arduino UNO ->Gestor de Tarjetas y buscar los paquetes de esp8266 e instalar la última versión como muestra en la figura 63.

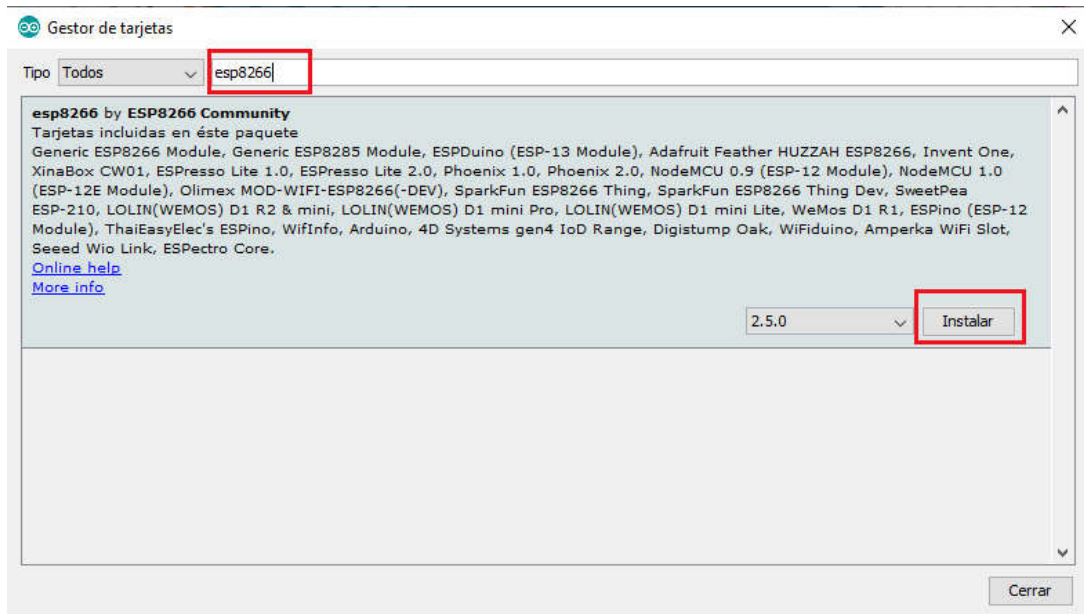


Figura 61: Gestor de tarjetas

Terminado el proceso de instalación se procede a verificar que se ejecutó de manera correcta en "Herramientas>Placa Arduino UNO" y ver que se agregaron nuevos módulos Arduino, entre ellos el genérico del esp8266 que se utilizó.

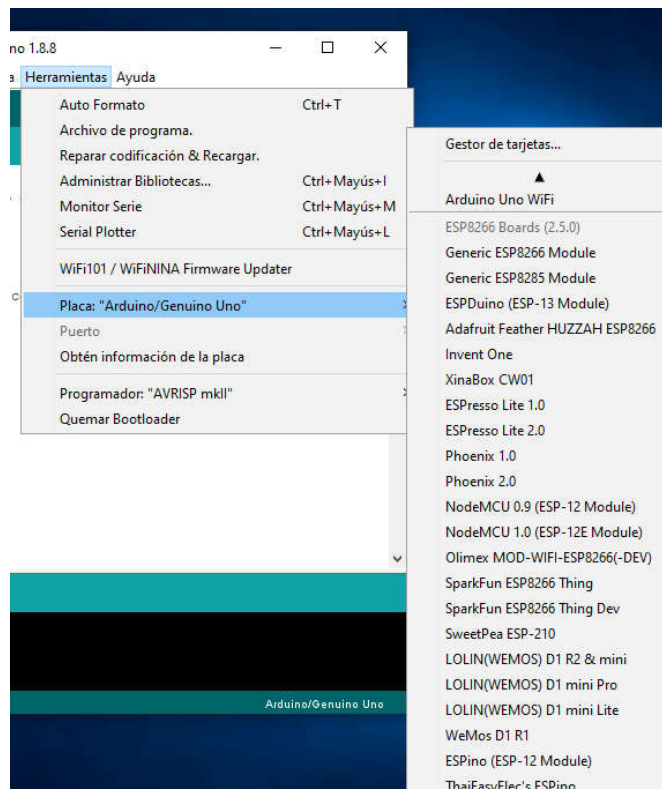


Figura 62: Configuración terminada

Anexo F: Instalación de la aplicación de AppInventor y conexión con Arduino

El primer paso es ir al sitio de AppInventor <http://ai2.appinventor.mit.edu/> e iniciar sesión con nuestra cuenta de Google.

Una vez iniciada la sesión y con la aplicación de prueba construida se procede con la instalación de la misma y para realizar esto, debemos tener instalado en nuestro móvil la aplicación MIT AI Companion. A continuación, vamos a la pestaña de Build y seleccionamos App (proviene QR code for .apk), mientras tanto iniciamos el MIT en nuestro móvil como se ve en la figura 66 y usamos scan QR e instalamos la aplicación.



Figura 63: Código QR para la instalación de AppInventor

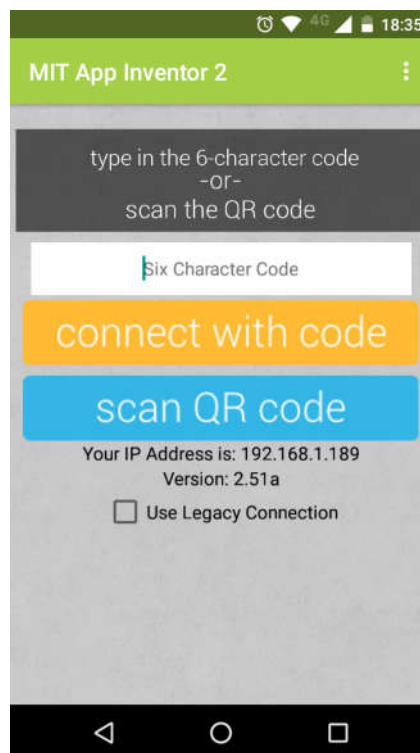


Figura 64: MIT

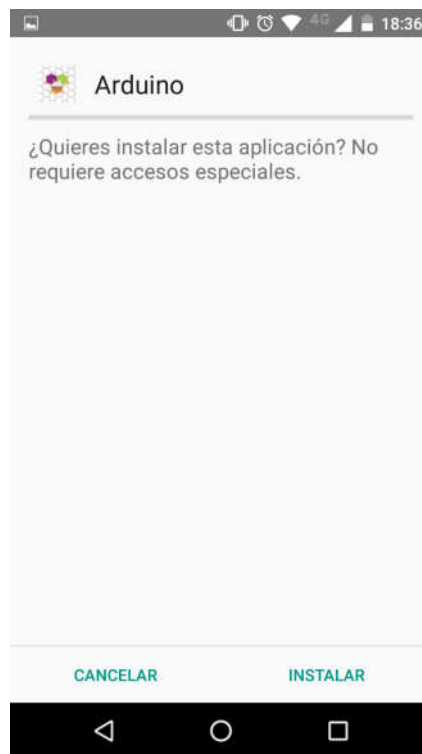


Figura 65: Vista de instalación

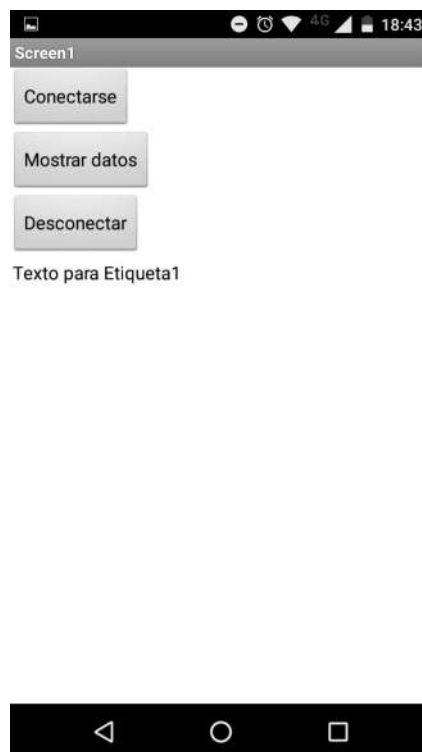


Figura 66: Vista de la aplicación instalada

Durante la instalación nos saldrá un cartel de bloqueo por parte de Play Protect que es una herramienta de seguridad que ofrece Google para los dispositivos, advirtiéndonos que el desarrollador es desconocido y que no pueden asegurarnos que la aplicación sea segura, ya que no pertenece a la PlayStore, por lo tanto vamos a seleccionar la opción de “instalar de todas formas”.

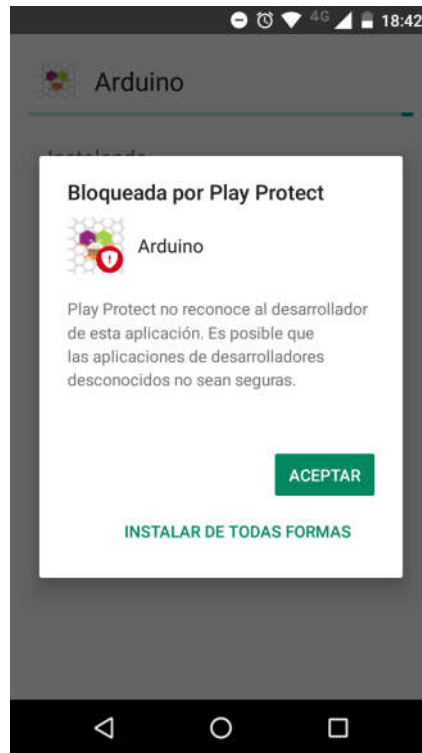


Figura 67: Cartel de bloqueo por Play Protect

Una vez que presionemos el botón “Conectar” el Arduino intentará acceder a nuestro dispositivo, por lo que debemos aceptar el siguiente cartel.



Figura 68: Cartel para dar los permisos de conexión USB

Por último se verá debajo de los botones la etiqueta de “true” que significa que se conectó el Arduino de manera satisfactoria.

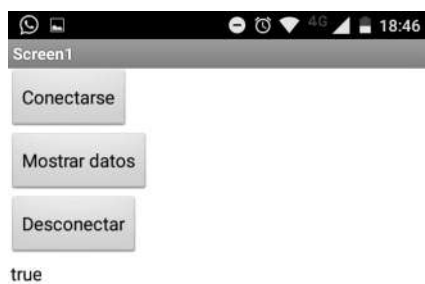


Figura 69: Conexión exitosa con el Arduino

Anexo G: Configuración de phpMyAdmin

Una vez instalado Xampp, con un navegador nos dirigimos a la url que tiene nuestra dirección IP en nuestro caso es la 192.168.0.16 o se puede acceder mediante `http://localhost/`.

Una vez dentro de la interfaz de phpMyAdmin creamos una base de datos con el nombre “Arduino”.

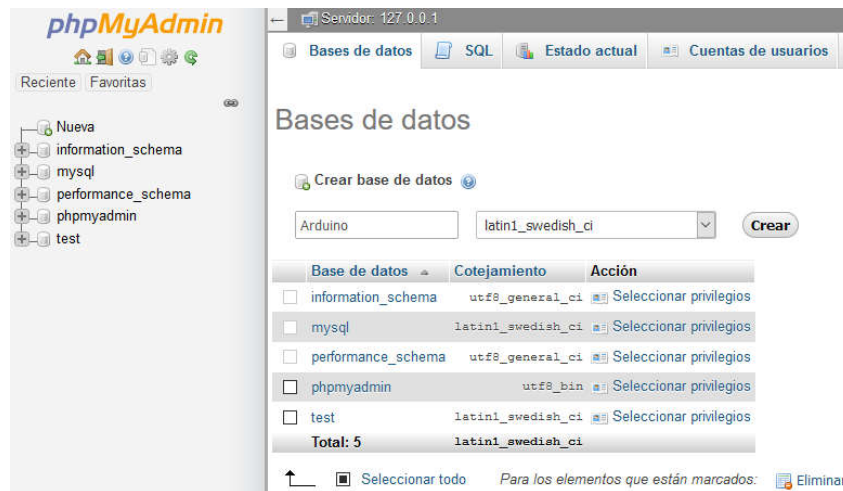


Figura 70: Creación de la base de datos

Ahora se procede a crear las tablas para almacenar los datos de los sensores seleccionando la cantidad de columnas que necesita, creamos las tablas “consumodeagua” y “consumoelectrico”.

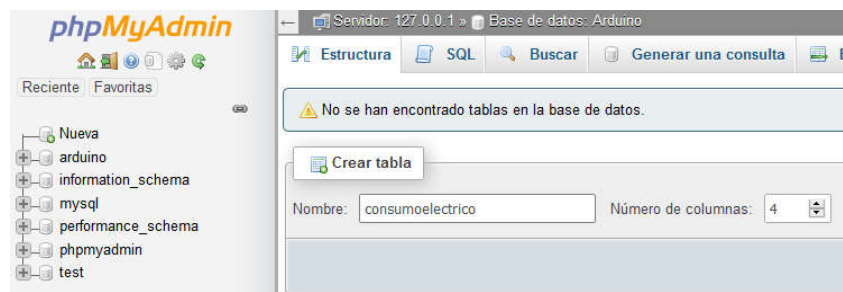


Figura 71: Creación de tabla en phpMyAdmin

En las siguientes figuras se muestra los atributos que se asignan a cada columna de las tablas, para los identificadores de cada una se escoge el tipo de dato “integer”, con respecto a la fecha seleccionamos “timestamp”, para la velocidad, el volumen, la intensidad, la potencia se escoge “float” y se marca la opción para que el identificador sea autoincremental.

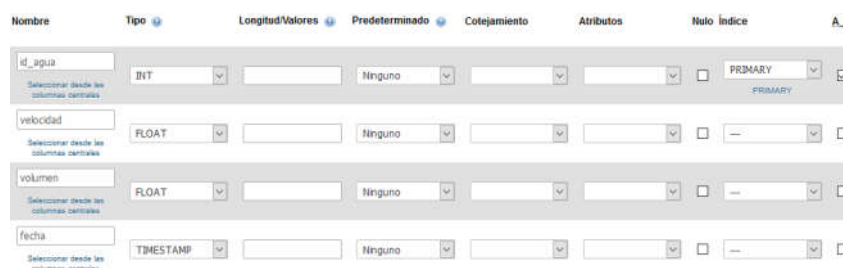


Figura 72: Creación de la tabla de consumo de agua

Nombre	Tipo	Longitud/Valores	Predeterminado	Cotejamiento	Atributos	Índice	A.I.
id_electricidad	INT		Ninguno			PRIMARY	<input checked="" type="checkbox"/>
ems	FLOAT		Ninguno			--	<input type="checkbox"/>
potencia	FLOAT		Ninguno			--	<input type="checkbox"/>
fecha	TIMESTAMP		Ninguno			--	<input type="checkbox"/>

Figura 73: Creación de la tabla de consumo de electricidad

Este proceso se realizó también para crear las tablas de “realwater” y “realelectric” que sirven para mostrar los valores de los sensores en tiempo real.

Anexo H: Código del proyecto

```

1  ...# include <SoftwareSerial.h>
2  #include <Thread.h>
3  SoftwareSerial espSerial(11,10);
4  Thread thr = Thread();
5  String ssid="SkyNet";
6  String password = "1466card";
7  boolean DEBUG=true;
8  volatile int NumPulsos;
9  int PinSensor = 6;
10 float factor_conversion=7.11;
11 float volumen=0;
12 long dt=0;
13 long t0=0;
14 String data;
15 String uri = "/updatedata.php";
16 String urir = "/updatedatareal.php";
17 float frecuencia;
18 float velocidad;
19 float irms;
20 float potencia;
21 boolean senddata()
22 {
23 frecuencia=ObtenerFrecuecia();
24 velocidad=frecuencia / factor_conversion;
25 dt=millis()-t0;
26 t0=millis();
27 volumen=volumen+(velocidad/60)*(dt/1000);
28 interrupts();
29 irms=get_corriente();
30 noInterrupts();
31 potencia=irms*220.0;
32 interrupts();
33
34 String cmd = "AT+CIPSTART=\\"TCP\",";
35 cmd += "192.168.0.16";
36 cmd += "\",80";
37 espSerial.println(cmd);
38 if (DEBUG) Serial.println(cmd);
39 if(espSerial.find("Error")){
40 if (DEBUG) Serial.println("AT+CIPSTART error");
41 return false;
42 }
43
44 String getRequest =
45 "GET " + uri;
46 getRequest += "?velocidad=";
47 getRequest += String(velocidad);
48 getRequest += "&volumen=";
49 getRequest += String(volumen);
50 getRequest += "&irms=";
51 getRequest += String(irms);
52 getRequest += "&potencia=";
53 getRequest += String(potencia);
54 getRequest += "\r\n\r\n";
55 delay(10);
56 cmd = "AT+CIPSEND=";

```

```

57 cmd += String(getRequest.length());
58 espSerial.println(cmd);
59 Serial.print(getRequest);
60 espSerial.print(getRequest);
61 if (espSerial.find(">")){
62   espSerial.print(getRequest);
63   if ( espSerial.find("SEND OK")) {
64     espSerial.println("AT+CIPCLOSE");
65   }
66 }
67 return true;
68 }
69
70 boolean senddatareal(float velocidad, float volumen, float irms, float
    potencia){
71   String cmd = "AT+CIPSTART=\\"TCP\",";
72   cmd += "192.168.0.16";
73   cmd += "\",80";
74   espSerial.println(cmd);
75   if (DEBUG) Serial.println(cmd);
76   if(espSerial.find("Error")){
77     if (DEBUG) Serial.println("AT+CIPSTART error");
78     return false;
79   }
80   String getRequest =
81   "GET " + urir;
82   getRequest += "?velocidad=";
83   getRequest += String(velocidad);
84   getRequest += "&volumen=";
85   getRequest += String(volumen);
86   getRequest += "&irms=";
87   getRequest += String(irms);
88   getRequest += "&potencia=";
89   getRequest += String(potencia);
90   getRequest += "\r\n\r\n";
91   delay(10);
92   cmd = "AT+CIPSEND=";
93   cmd += String(getRequest.length());
94   espSerial.println(cmd);
95   Serial.print(getRequest);
96   espSerial.print(getRequest);
97   if (espSerial.find(">")){
98     espSerial.print(getRequest);
99     if ( espSerial.find("SEND OK")) {
100       espSerial.println("AT+CIPCLOSE");
101     }
102   }
103   return true;
104 }
105
106 void ContarPulsos () {
107   NumPulsos++;
108 }
109
110 int ObtenerFrecuecia () {
111   int frecuencia;
112   NumPulsos = 0;
113   interrupts ();
114   delay(1000);
115   noInterrupts ();

```

```

116 frecuencia=NumPulsos;
117 return frecuencia;
118 }
119
120 float get_corriente ()
121 {
122 float voltajeSensor;
123 float corriente=0;
124 float Sumatoria=0;
125 long tiempo=millis ();
126 int N=0;
127 while ( millis ()-tiempo<500)
128 {
129 voltajeSensor = analogRead(A0) * (1.1 / 1023.0);
130 corriente=voltajeSensor *30.0;
131 Sumatoria=Sumatoria+sq ( corriente );
132 N=N+1;
133 }
134 Sumatoria=Sumatoria *2;
135 corriente=sqrt (( Sumatoria )/N);
136 return ( corriente );
137 }
138
139 void setup () {
140 Serial .begin (9600);
141 espSerial .begin (115200);
142 analogReference (INTERNAL);
143 espSerial .println ("AT+CWMODE=1");
144 espSerial .println ("AT+CWJAP=\"\"+ssid+\"\", \"\"+password+\"\"");
145 pinMode (PinSensor , INPUT);
146 attachInterrupt (0, ContarPulsos ,RISING);
147 Serial .println ("Envie 'r' para restablecer el volumen a 0 Litros");
148 t0=millis ();
149 if (DEBUG) Serial .println ("Setup completed");
150 thr .enabled = true;
151 thr .setInterval (1800000);
152 thr .onRun (senddata);
153 }
154
155 void loop () {
156 float frecuencia=ObtenerFrecuecia ();
157 float velocidad=frecuencia / factor_conversion;
158 dt=millis ()-t0;
159 t0=millis ();
160 volumen=volumen+( velocidad /60)*( dt/1000);
161 interrupts ();
162 float irms=get_corriente ();
163 noInterrupts ();
164 float potencia=irms *220.0;
165 interrupts ();
166 senddatareal ( velocidad , volumen , irms , potencia);
167 noInterrupts ();
168
169 if ( thr .shouldRun ())
170 {
171 thr .run ();
172 }
173 noInterrupts ();
174 }

```

Anexo I: Construcción del circuito eléctrico

Con el uso del software PCB Wizard versión 3.5 se diseñó el circuito electrónico para el facilitar las conexiones del módulo de wifi esp8266 y del circuito integrado LM358.

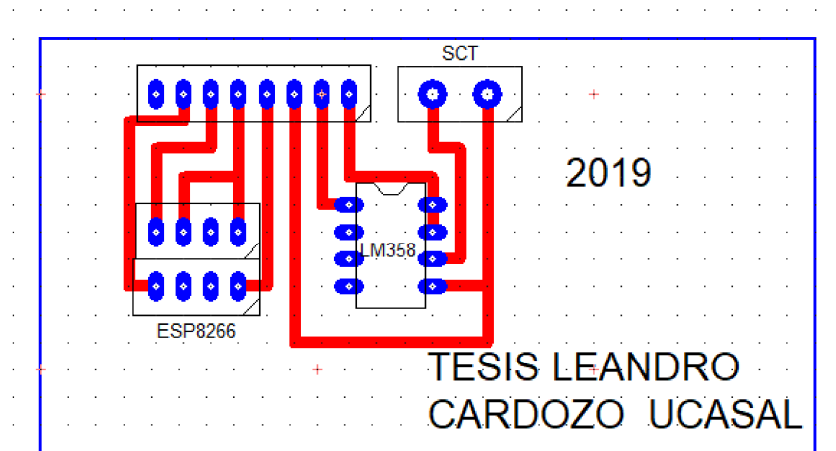


Figura 74: Diseño del circuito impreso

Para poder armar la placa con los componente primero pasamos el diseño a una hoja de papel cuadriculado y remarcamos las dimensiones de la placa, como se ve en la siguiente figura.

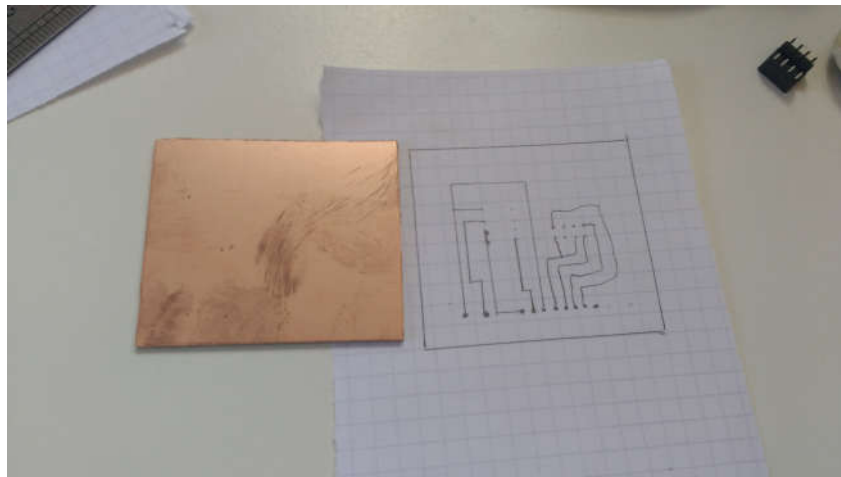


Figura 75: Diseño del circuito impreso en hoja

Para ser más precisos se marca con un martillo y un clavo los lugares en donde se ubicarán los componentes, también se puede realizar con una aguja.

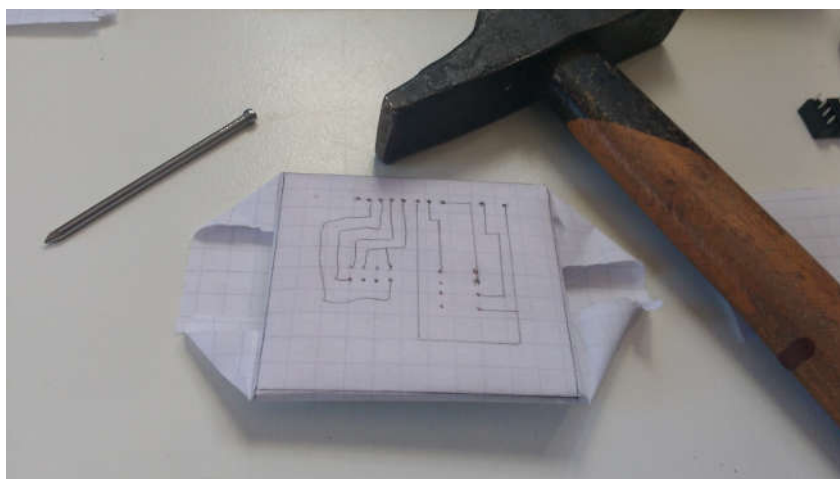


Figura 76: Marcado de los componentes con martillo

Se marcan las pistas con un marcador indeleble, para que el ácido ferrico quite de la placa todo el cobre que no necesitamos, se demora entre 10 a 15 minutos este proceso, al terminar se quita la tinta con una virulana o esponja de acero y para evitar que se oxide el cobre se limpia la placa para eliminar la tinta de los agujeros hechos por el clavo y se echa un aerosol para ese proposito.

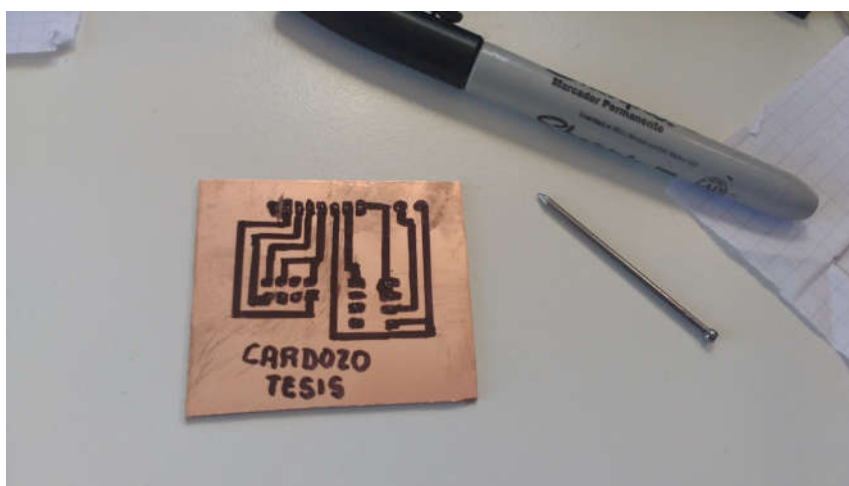


Figura 77: Marcado con tinta indeleble

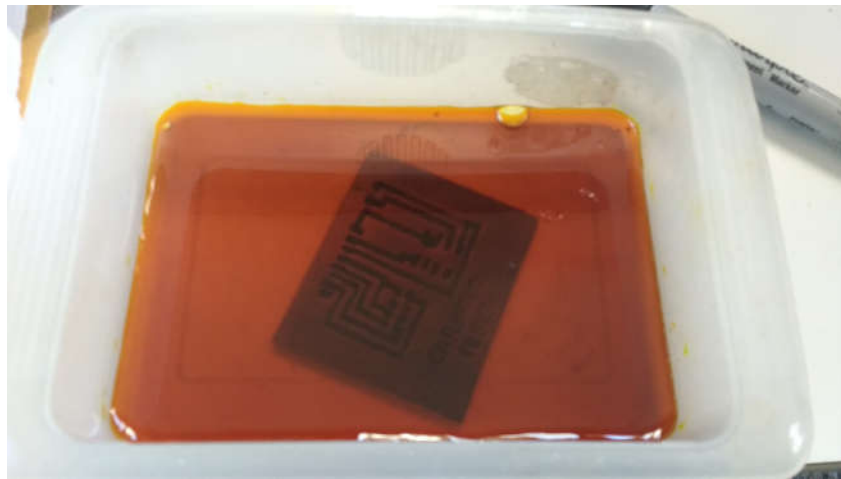


Figura 78: Circuito en ácido ferrico

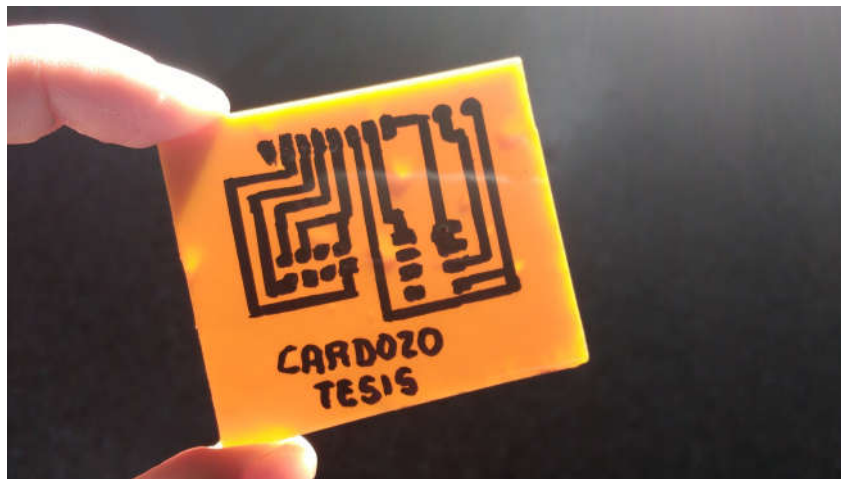


Figura 79: Vista del circuito fuera del ácido



Figura 80: Circuito con las pistas de cobre

Por último con una soldadora se fijan los componentes usando estaño.

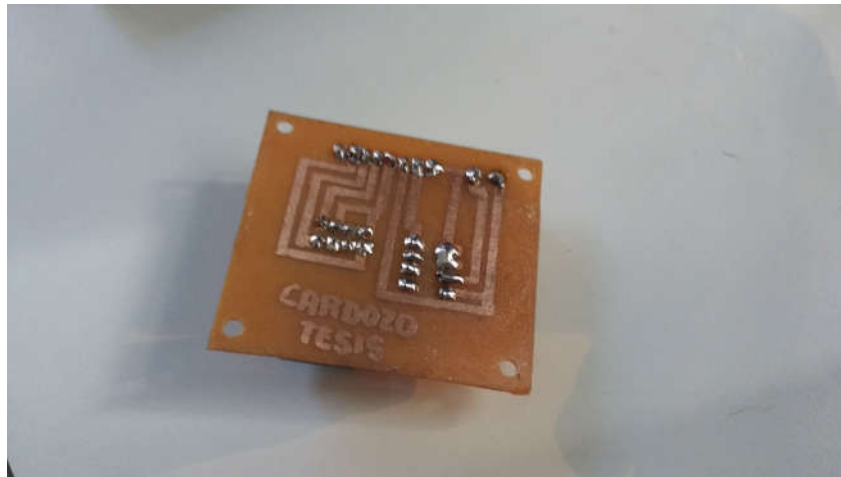


Figura 81: Circuito Soldado



Figura 82: Vista superior del circuito terminado

Anexo J: Creación e instalación de la aplicación

Para poder instalar la aplicación hay que crear un archivo con extensión apk para tal motivo. El primer paso es acceder al manifiesto de Android seleccionando en las propiedades del proyecto. marcamos 5.0 como la versión mínima de Android.

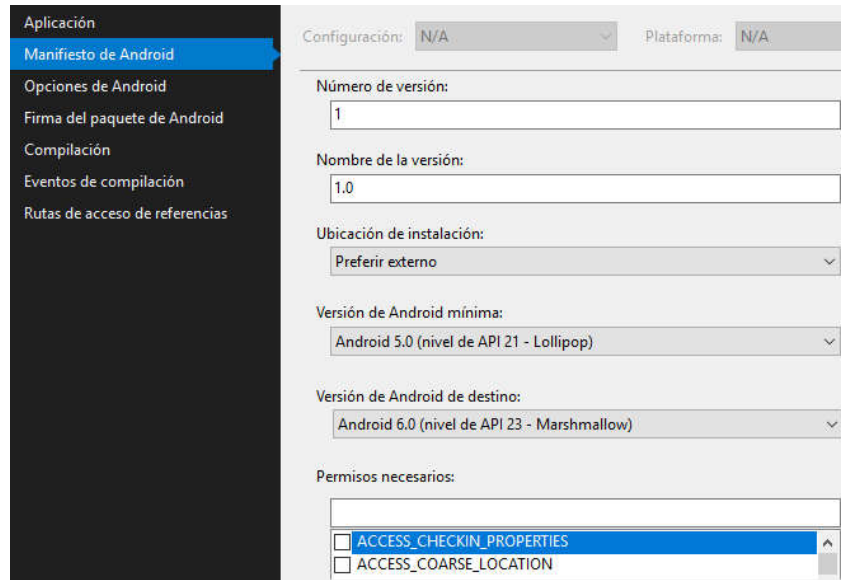


Figura 83: Manifiesto de Android

A continuación con el click derecho sobre el proyecto de Android seleccionamos la opción de “Archivar”.

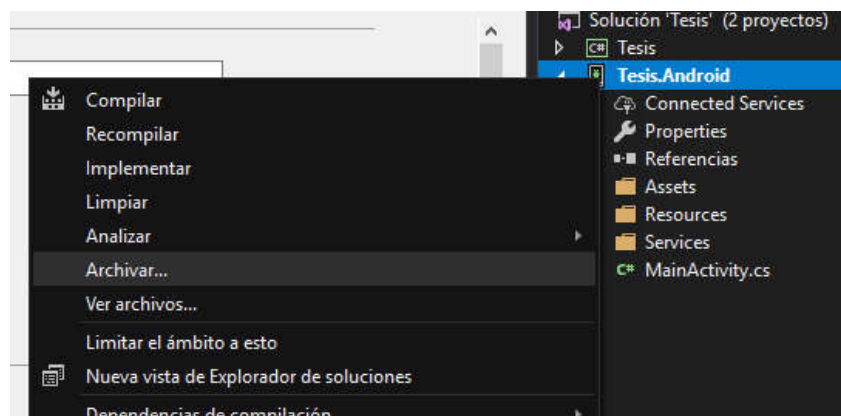


Figura 84: Archivar la aplicación

Seleccionamos la opción para compilar el proyecto y comenzará a cargar, esto demora aproximadamente 5 minutos.

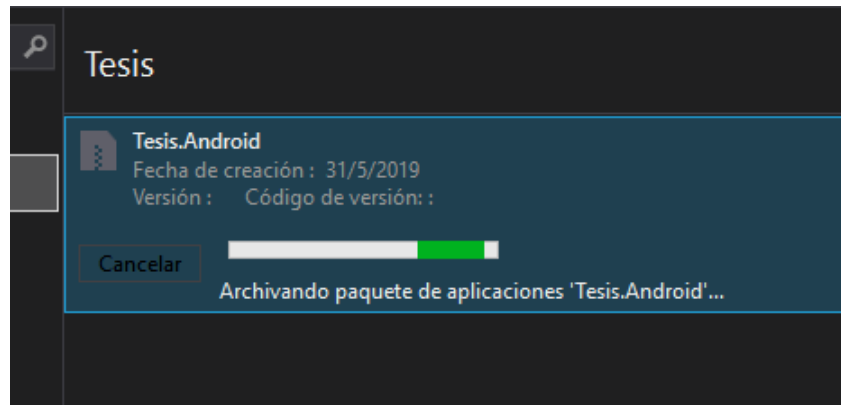


Figura 85: Cargando la aplicación

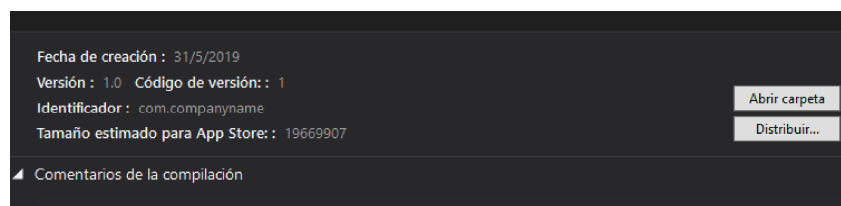


Figura 86: Carga finalizada

Al finalizar nos dirigimos a la carpeta dónde se ubica el archivo creado.

Nombre	Fecha de modifica...	Tipo	Tamaño
app-icons	31/5/2019 23:43	Carpeta de archivos	
mSYM	31/5/2019 23:43	Carpeta de archivos	
archive.xml	31/5/2019 23:43	Documento XML	1 KB
com.companyname.apk	31/5/2019 23:42	Archivo APK	19.209 KB

Figura 87: Archivo Apk

Para poder instalarlo debemos configurar la seguridad de nuestro móvil para permitir la instalación de fuentes desconocidas.

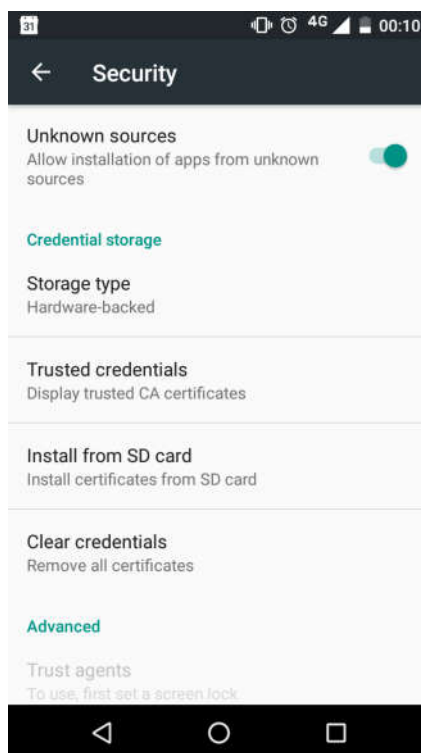


Figura 88: Permitir instalar aplicaciones desconocidas

Una vez que tenemos el archivo apk en nuestro dispositivo procedemos a instalarlo, al terminar veremos la aplicación en nuestro móvil.

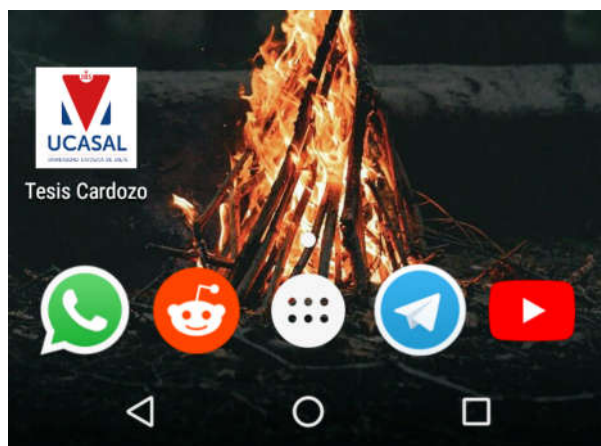


Figura 89: Instalación finalizada

Anexo K: Archivo php

En este Anexo se muestra solo uno de los archivos con extensión php, el que sube los datos de los sensores a la base de datos MySQL.

```

1  <?php
2  define ("DB_SERVER", "192.168.0.16");
3  define ("DB_USER", "root");
4  define ("DB_PASSWORD", "root");
5  define ("DB_DATABASE", "arduino");
6  $velocidad = $_GET["velocidad"];
7  $volumen = $_GET["volumen"];
8  $irms = $_GET["irms"];
9  $potencia = $_GET["potencia"];
10
11  try
12  {
13  $conexion = mysqli_connect(DB_SERVER , DB_USER, DB_PASSWORD, DB_DATABASE
14  ) or die("Unable to connect to MySQL") ;
15  mysqli_select_db($conexion,"arduino") or die("No se pudo seleccionar la
16  base de datos"); ;
17  mysqli_query($conexion,DB_DATABASE);
18  $query="INSERT INTO 'arduino'.'consumodeagua' ('id_agua', 'velocidad', '
19  volumen', 'fecha') VALUES (NULL, $velocidad,$volumen,
20  CURRENT_TIMESTAMP);";
21  $resultado= $conexion->query($query);
22
23  if($resultado)
24  {
25  echo "Registro exitoso";
26  }
27  else
28  {
29  echo "Registro no exitoso";
30  }
31  }
32  catch(PDOException $e)
33  {
34  echo $sql . "<br>" . $e->getMessage();
35  }
36  try
37  {
38  $conexion = mysqli_connect(DB_SERVER , DB_USER, DB_PASSWORD, DB_DATABASE
39  ) or die("No se puede conectar a MySQL") ;
40  mysqli_select_db($conexion,"arduino") or die("No se pudo seleccionar la
41  base de datos"); ;
42  mysqli_query($conexion,DB_DATABASE);
43  $query="INSERT INTO 'arduino'.'consumoelectrico' ('id_electricidad', '
44  irms', 'potencia', 'fecha') VALUES (NULL, $irms,$potencia,
45  CURRENT_TIMESTAMP);";
46  $resultado= $conexion->query($query);
47
48  if($resultado)
49  {
50  echo "Registro exitoso";
51  }
52  else
53  {
54  echo "Registro no exitoso";
55  }
56  }

```

```
47     }
48     }
49     catch (PDOException $e)
50     {
51     echo $sql . "<br>" . $e->getMessage ();
52     }
53 ?>
```