



**DESARROLLO DE UN GESTOR PARA EL
CONTROL DE TAREAS DE
AUTOMATIZACIÓN DE CIRCUITOS
ELECTRÓNICOS HACIENDO USO DE UN
MICROCONTROLADOR INALÁMBRICO
A TRAVÉS DE UN SISTEMA WEB Y
MÓVIL**

Autor: Murillo Samuel

CI: 25.582.018

Urb. Yuma II, calle N° 3. Municipio San Diego
Teléfono: (0241) 8714240 (master) – Fax: (0241) 8712394



**REPÚBLICA BOLIVARIANA DE VENEZUELA
UNIVERSIDAD JOSÉ ANTONIO PÁEZ
FACULTAD DE INGENIERÍA
ESCUELA DE COMPUTACIÓN**

**DESARROLLO DE UN GESTOR PARA EL
CONTROL DE TAREAS DE
AUTOMATIZACIÓN DE CIRCUITOS
ELECTRÓNICOS HACIENDO USO DE UN
MICROCONTROLADOR INALÁMBRICO
A TRAVÉS DE UN SISTEMA WEB Y
MÓVIL**

**Trabajo de grado presentado como requisito para optar al título de
INGENIERO DE COMPUTACIÓN**

Autor: Murillo Samuel

C.I. 25.582.018

Tutor: Ing. Karen Ramírez

C.I. 14.571.240

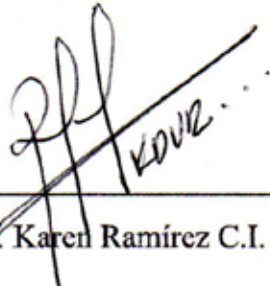


REPÚBLICA BOLIVARIANA DE VENEZUELA
UNIVERSIDAD JOSÉ ANTONIO PÁEZ
FACULTAD DE INGENIERÍA
ESCUELA DE COMPUTACIÓN

ACEPTACIÓN DEL TUTOR

Quien suscribe, Ingeniero Karen Ramírez portador de la cédula de identidad N° 14.571.240, en mi carácter de tutor del trabajo de grado presentado por el ciudadano Samuel José Murillo Rodríguez, portador de la cédula de identidad N° 25.582.018, titulado **DESARROLLO DE UN GESTOR PARA EL CONTROL DE TAREAS DE AUTOMATIZACIÓN DE CIRCUITOS ELECTRÓNICOS HACIENDO USO DE UN MICROCONTROLADOR INALÁMBRICO A TRAVÉS DE UN SISTEMA WEB Y MÓVIL** presentado como requisito parcial para optar al título de Ingeniero de Computación, considero que dicho trabajo reúne los requisitos y méritos suficientes para ser sometido a la presentación pública y evaluación por parte del jurado examinador que se designe.

En San Diego, a los dieciséis (16) días del mes de octubre del año dos mil diecisiete.



Ing. Karen Ramírez C.I. 14.571.240



Universidad José Antonio Páez
Facultad de Ingeniería

FI-I-047-2017-2

Valencia, 07 de Julio de 2017.

Ciudadano:
Murillo Samuel
C.I. 25.582.018
Presente.-

Cumplo con informarle que la Comisión de Trabajo de Grado y Pasantías de la Facultad de Ingeniería en su reunión N° 2-2017 de fecha **07/07/2017** aprobó el proyecto de trabajo de grado titulado **“DESARROLLO DE UN GESTOR PARA EL CONTROL DE TAREAS DE AUTOMATIZACIÓN DE CIRCUITOS ELECTRÓNICOS HACIENDO USO DE UN MICROCONTROLADOR INALÁMBRICO A TRAVÉS DE UN SISTEMA WEB Y MÓVIL”** Presentado por usted como requisito para optar al título de Ingeniero en Computación.

Se ratifica la designación de la Ing. Karen Ramírez, C.I. 14.571.240 y la Ing. Alicia Pizzella, C.I. 4.598.880 como Tutores Académicos que lo asesorarán en el desarrollo de este proyecto.

Atentamente,



Ing. José Gregorio Díaz
Decano de la Facultad de Ingeniería

c. c. Coordinación de Pasantías y Trabajo de Grado (2).

JGD/fr

AGRADECIMIENTOS

A mis padres, por formarme como persona y por su apoyo incondicional en todo momento.

A mis amigos y compañeros de clase, por siempre ofrecerme una escapada al estrés, para reír y compartir siempre que fuese necesario. A Juan, por brindarme la inspiración de este proyecto y a Erika por siempre estar ahí cuando la necesité.

A Omar Restrepo, por creer siempre en mí y en el proyecto.

A la comunidad de software libre, por colaborar en proyectos para el bien de todos los desarrolladores.

ÍNDICE GENERAL

CONTENIDO	Pp
RESUMEN	ix
INTRODUCCIÓN	1
CAPÍTULO	
I EL PROBLEMA	
1.1 Planteamiento del problema.....	3
1.2 Formulación del problema.....	4
1.3 Objetivos de la investigación.....	5
1.3.1 Objetivo general.....	5
1.3.2 Objetivos específicos.....	5
1.4 Justificación de la investigación.....	5
1.5 Alcance de la Investigación.....	6
II MARCO TEÓRICO	
2.1 Antecedentes.....	8
2.2 Bases teóricas.....	10
2.2.1 Automatización y control.....	10
2.2.2 Domótica e Inmótica.....	11
2.2.3 Internet de las cosas.....	13
2.2.4 Señales Analógicas y Digitales.....	14
2.2.5 Microcontroladores.....	15
2.2.6 Arduino.....	16
2.2.7 Raspberry Pi.....	17
2.2.8 ESP8266.....	17
2.2.9 ESP32.....	18
2.2.10 Angular.....	19
2.2.11 NativeScript.....	19
2.2.12 MongoDB.....	20
2.3 Definición de términos básicos.....	20

III	MARCO METODOLÓGICO	
	3.1 Tipo de investigación.....	22
	3.2 Diseño de la Investigación.....	22
	3.3 Nivel de la investigación.....	23
	3.4 Técnicas e instrumentos de recolección de datos.....	23
	3.5 Población y Muestra.....	24
	3.6 Técnicas de análisis de datos.....	24
	3.8 Fases Metodológicas.....	25
IV	RESULTADOS	
	4.1 Fase 1: Planificación.....	27
	4.1.1 Entrevista.....	27
	4.1.2 Definición de requerimientos del sistema.....	28
	4.1.3 Estudio comparativo descriptivo.....	29
	4.2 Fase 2: Diseño.....	32
	4.2.1 Componentes del sistema.....	32
	4.2.2 Casos de uso.....	33
	4.2.3 Modelo de base de datos.....	48
	4.3 Fase 3: Desarrollo.....	51
	4.4 Fase Pruebas.....	56
V	CONCLUSIONES Y RECOMENDACIONES	
	5.1 Conclusiones.....	59
	5.2 Recomendaciones.....	60
	REFERENCIAS	62
	ANEXOS	
	A Instrumento de investigación.....	65

ÍNDICE DE FIGURAS

FIGURA	Pp
1. Componentes del sistema.....	32
2. Casos de uso Usuario.....	34
3. Casos de uso Demonio.....	35
4. Casos de uso Microcontrolador.....	35
5. Modelo de base de datos.....	48
6. Página de Inicio de sesión.....	51
7. Formulario de registro.....	51
8. Lista de tareas.....	52
9. Ver tarea.....	52
10. Crear tarea.....	53
11. Lista de microcontroladores.....	53
12. Ver microcontrolador.....	54
13. Agregar microcontrolador.....	54
14. Listar reportes.....	55
15. Ver reporte.....	55

INDICE DE TABLAS

TABLA	Pp
1. Comparación de microcontroladores.....	30
2. Caso de uso: Iniciar sesión.....	36
3. Caso de uso: Recuperar contraseña.....	36
4. Caso de uso: Ver tarea.....	37
5. Caso de uso: Listar tareas.....	37
6. Caso de uso: Agregar tarea.....	38
7. Caso de uso: Modificar tarea.....	38
8. Caso de uso: Eliminar tarea.....	39
9. Caso de uso: Ver microcontrolador.....	39
10. Caso de uso: Listar microcontroladores.....	40
11. Caso de uso: Agregar microcontrolador.....	40
12. Caso de uso: Modificar microcontrolador.....	41
13. Caso de uso: Eliminar microcontrolador.....	41
14. Caso de uso: Ver reporte.....	42
15. Caso de uso: Listar reportes.....	42
16. Caso de uso: Conectar al servidor WebSocket.....	43
17. Caso de uso: Configurar nueva tarea.....	43
18. Caso de uso: Evaluar tarea.....	44
19. Caso de uso: Iniciar tarea.....	45
20. Caso de uso: Detener tarea.....	45
21. Caso de uso: Crear tarea.....	46
22. Caso de uso: Inicializar puertos.....	46
23. Caso de uso: Informar cambio de estado de un puerto.....	47
24. Caso de uso: Cambiar estado de un puerto.....	47
25. Descripción del modelo de base de datos.....	49
26. Caso de prueba: Agregar microcontrolador.....	56
27. Caso de prueba: Agregar tarea.....	56
28. Caso de prueba: Iniciar tarea.....	57
29. Caso de prueba Inicializar puertos.....	57
30. Caso de prueba: Informar cambio de estado de un puerto.....	58
31. Caso de prueba Evaluar tarea.....	58



**REPÚBLICA BOLIVARIANA DE VENEZUELA
UNIVERSIDAD JOSÉ ANTONIO PÁEZ
FACULTAD DE INGENIERÍA
ESCUELA DE COMPUTACIÓN**

**DESARROLLO DE UN GESTOR PARA EL CONTROL DE TAREAS DE
AUTOMATIZACIÓN DE CIRCUITOS ELECTRÓNICOS HACIENDO USO DE
UN MICROCONTROLADOR INALÁMBRICO A TRAVÉS DE UN SISTEMA
WEB Y MÓVIL**

Autor: Murillo Samuel

Tutor: Ing. Karen Ramírez

Fecha: Junio, 2017

RESUMEN

La necesidad de controlar, monitorear y ejecutar eventos ha tenido importancia desde la era industrial hasta nuestros días y al igual que cualquier otro sistema, los sistemas de control deben adaptarse a ciertas necesidades del usuario. Entre estas necesidades se resalta que sea: fácil de usar, informativo, y automatizado. El sentido de la automatización es alejar lo más posible el contacto del personal con los procesos por la simple razón de que un sistema automatizado disminuye o elimina por completo el indeseable error humano. Cada día son más las empresas que necesitan de un sistema de control y además que esté automatizado. Por lo tanto, la presente investigación tiene como objetivo desarrollar un gestor para el control de tareas de automatización de circuitos electrónicos haciendo uso de un microcontrolador inalámbrico a través de un sistema web y móvil a fin de facilitar con mayor eficiencia operativa a los usuarios el seguimiento y control de actividades con un costo mínimo de implantación. La investigación define como un proyecto especial, con un diseño de campo y nivel descriptivo. La población engloba a profesionales y usuarios del área de electrónica y automatización, la cual es una población con cantidad desconocida de individuos, por lo tanto se considera infinita. Posteriormente, se tiene una muestra intencional, escogida con criterios preestablecidos por el autor. Para esta investigación las técnicas de recolección de datos que se utilizarán son la observación directa y la entrevista semi-estructurada. El desarrollo del sistema seguirá las fases de la metodología XP, siendo estas Planificación, Diseño, Desarrollo y Pruebas.

Descriptores: Automatización, Microcontrolador inalámbrico, Sistema web y móvil, Metodología XP.

INTRODUCCIÓN

Desde la revolución industrial en el siglo XVIII e inicios del siglo XIX, la humanidad emprende un incremento en la producción agrícola, textil e industrial a la vez que disminuye el tiempo requerido para llevar a cabo procesos de producción; la posibilidad de realizar tareas repetitivas con estándares de calidad, sin alterar los costos de producción, representa una solución codiciada por grandes y pequeños empresarios. Ahora bien, debido a la introducción de cadenas de producción por Henry Ford, la necesidad de optimizar los procesos de producción se convirtió en un factor primordial en el triunfo y posicionamiento económico de una empresa en el mercado.

Con la llegada de la electrónica, dichos procesos dejaron de ser netamente mecánicos y se da paso al comienzo de la era de la automatización, que se caracteriza por ser un periodo de grandes cambios. Es allí cuando, a medida que la electrónica toma fuerza en la industria, con la aparición de computadores, la integración de tecnología de vanguardia y la electrónica digital, se perfeccionan y mejoran los procesos de automatización.

Posteriormente, la llegada de los microcontroladores permite un aumento en la producción industrial, con un decremento notable de horas hombre. Es necesario agregar, que no solo en el ramo de la automatización industrial estos componentes de tipo electrónicos permiten su desarrollo y mejora, sino que junto a la integración de sensores, actuadores y otros dispositivos se potenció el inicio de lo que se conoce como Domótica.

Por otra parte, la nanoelectrónica permitió que los microcontroladores fuesen más versátiles, no solo permitiendo la conectividad o la toma de información a través de sensores sino que, a su vez, concede la conectividad tanto alámbrica como inalámbrica, con la finalidad de realizar una tarea de manera eficiente, sin la supervisión del usuario.

Cabe destacar que, la automatización está ligada a la economía mundial, esto refleja un costo de inversión considerable, es allí cuando, la integración de la domótica,

usando microcontroladores y conocimiento en programación permite lograr la automatización de tareas o actividades cotidianas a cambio de un costo menor.

Esta investigación se conforma por cinco (5) capítulos descritos a continuación:

Capítulo I: El problema, se describe la problemática a profundidad, los objetivos a alcanzar para abordar las circunstancias descritas, la razón de ser de la presente investigación y el alcance de la misma.

Capítulo II: Marco teórico, se consultan investigaciones previas pertinentes al tema a tratar, adicionalmente contiene las bases teóricas que sustentan la investigación así como también la definición de los términos básicos.

Capítulo III: Marco metodológico, son presentadas las características investigativas que rigen el proyecto, siendo estas el tipo, diseño y nivel de investigación que encajan con el proyecto. También se define la población y muestra, donde se van a realizar dichos estudios con técnicas y herramientas que nos facilitarán la investigación, y se describen las fases metodológicas donde se detalla de forma precisa la realización en cada fase.

Capítulo IV: en el cual se exponen los resultados obtenidos en cada una de las fases metodológicas planteadas y desarrolladas de manera secuencial y cronológica en el capítulo III.

Capítulo V: Conclusiones y recomendaciones, se exponen las conclusiones obtenidas a partir del desarrollo, o, así como también recomendaciones para su posible continuación y la extensión del sistema desarrollado.

CAPÍTULO I

EL PROBLEMA

En este capítulo se describe la situación objeto de estudio, ubicándola en un contexto que permita comprender su origen y relaciones, a través de un enfoque metodológico, y conocer la definición del planteamiento y formulación del problema, los objetivos de estudio, la justificación y alcance del proyecto.

1.1 Planteamiento del problema

La necesidad de controlar, monitorear y ejecutar eventos ha tenido importancia desde la era industrial hasta nuestros días y al igual que cualquier otro sistema, los sistemas de control deben adaptarse a ciertas necesidades del usuario. Entre estas necesidades se resalta que sea: fácil de usar, de tal forma que entre menos conocimiento necesite el usuario mejor; informativo, siempre se debe llevar el control y estadística de las tareas realizadas, su razón y detalles; y automatizado, no basta con ejecutar las tareas sino que en lo posible debe saber cuándo hacerlas sin necesidad de la constante supervisión de un usuario.

El sentido de la automatización es alejar lo más posible el contacto del personal con los procesos por la simple razón de que un sistema automatizado disminuye o elimina por completo el indeseable error humano. Cada día son más las empresas que necesitan de un sistema de control y además que esté automatizado. En una industria, el sistema de instrumentación es de vital importancia para su correcto desenvolvimiento: facilita seguir con los estándares de calidad, permite detectar fallas antes de que ocurran y brinda estadísticas a los ingenieros para conocer el desempeño de un determinado proceso.

Gracias a la popularidad que han tenido los microcontroladores entre la comunidad de electrónica e informática, ya suman una cantidad importante de usuarios en el mundo que deciden emplear microcontroladores para realizar tareas de control y automatización, bien sea para uso personal o para pequeños circuitos en sectores laborales, pero con la limitante de que se deben hacer de forma manual y local. Además

se debe disponer de ciertos conocimientos previos de electrónica y programación para configurar y realizar dichas tareas de control.

Si bien es cierto que existen varios desarrollos que han permitido la administración remota de microcontroladores, lo cual permite ejecutar ciertas tareas a distancia, aún persisten los inconvenientes de tener que desarrollar dichas tareas a ejecutar y programarlas bajo las condiciones de ejecución deseadas. Esto conlleva a una inversión de tiempo y esfuerzo importante así sea para la tarea más sencilla como por ejemplo el encendido de un diodo LED.

En contraste, existen también soluciones para industrias y procesos de mayor envergadura como por ejemplo sistemas con uso de un PLC (Controlador lógico programable), pero estas soluciones tienen un costo elevado para pequeñas empresas, comercios independientes o entusiastas; por lo tanto una configuración que sea asequible, sencilla de usar e implementar pero que al mismo tiempo cubra las necesidades de monitoreo y control que posee el usuario le brindaría un beneficio sustancial.

Evidentemente existe un vacío de herramientas para esta población que necesita hacer seguimiento de variables y accionar circuitos de forma remota, que también requiere que sean de bajo costo, de fácil implementación y uso. Es importante que estas personas puedan contar con un sistema de automatización y monitorización constante en tiempo real, por lo que se requiere de implementación de nuevas tecnologías para cubrir esta necesidad.

1.2 Formulación del problema

Por las razones descritas anteriormente se genera la siguiente interrogante, ¿De qué manera se podría garantizar la ejecución y monitoreo de procesos en circuitos electrónicos que hacen uso de microcontroladores inalámbricos en ambientes controlados con la mínima interacción humana, con costos asequibles y requiriendo baja configuración por parte del usuario?

1.3 Objetivos de la investigación

1.3.1 Objetivo general

Desarrollar un gestor para el control de tareas de automatización de circuitos electrónicos haciendo uso de un microcontrolador inalámbrico a través de un sistema web y móvil a fin de facilitar con mayor eficiencia operativa a los usuarios el seguimiento y control de actividades con un costo mínimo de implantación.

1.3.2 Objetivos específicos

ahorraría tiempo de desarrollo y el resultado final sería de mejor calidad, dejando al usuario encargado íntegramente del armado de los circuitos y sus condiciones de acción, sin necesidad de poseer conocimientos previos de programación, cumpliendo con estándares de seguridad y consumo de energía.

Adicionalmente, también se busca ofrecer una herramienta asequible y de fácil implementación a los usuarios con necesidad de controlar y monitorear sistemas. Esto es posible gracias a que los microcontroladores se han convertido en piezas de tecnología accesibles a todo público gracias a su bajo costo. Al ser inalámbricos lo hace ideal para trabajar con circuitos discretos y de bajo perfil. Este se podría instalar en un panel y conectarse directo a la red a través de un router sin necesidad de un computador local.

Dicho esto, la presente investigación también tiene como meta proporcionar un punto de partida para futuras investigaciones sobre la temática en cuestión. Las tecnologías utilizadas en este trabajo son de constante avance y mejoramiento; nuevas alternativas surgirán y por lo tanto posteriores investigaciones también lo harán.

1.5 Alcance de la Investigación

La arquitectura del sistema contará con un servidor que se encuentra entre la aplicación cliente y los microcontroladores, esto brindará ciertas mejoras en cuanto a rendimiento y funcionalidades extra. El servidor sería el encargado de gestionar todas las tareas, base de datos y generar reportes; dejando los microcontroladores a cargo solamente de abrir y cerrar los circuitos con su puertos de entrada/salida, así como también enviar datos de confirmación de dicho puerto, permitiendo así aligerar la carga al microcontrolador, el cual no es un computador por lo tanto no debe encargarse de demasiadas tareas.

Este servidor intermediario se encargará de alojar las tareas creadas por el usuario, los reportes generados por las tareas ejecutadas, la permisología y control de acceso de los usuarios y la información de los microcontroladores a los cuales se debe conectar.

La aplicación cliente que será manejada por el usuario tendrá la posibilidad de configurar múltiples microcontroladores, cada uno con una ilustración visual de la tarjeta para que el usuario pueda seleccionar los puertos que quiere disponer como entrada o salida, ver cuáles están ocupados por tareas creadas previamente y monitorear el estado de los pines. También podrá gestionar varias tareas al mismo tiempo y ver reportes generados por las tareas previamente agregadas. Esta será desarrollada para que trabaje en el sistema operativo móvil Android, así como también en la plataforma Web.

Para interconectar la base de datos y las aplicaciones cliente, se programará una API la cual también se alojará en el servidor. Ésta se encargará de estandarizar y homogeneizar la entrada y salida de información de la base de datos, facilitando la conectividad multiplataforma y aumentando la seguridad e integridad de la data.

Por último, desarrollará un script para instalar el sistema completo en un servidor. Este script contará con una interfaz por línea de comandos para facilitar la instalación en servidores que no cuenten con una interfaz gráfica, la cual es una práctica común entre los servidores.

Cabe destacar que todo el sistema será de código abierto, permitiendo el reuso con cualquier propósito. Se redactará un manual de uso que estará disponible tanto en la página Web del sistema y en la aplicación móvil con la opción de descargarlo de forma local.

CAPÍTULO II

MARCO TEÓRICO

En este capítulo permite ubicar el tema objeto de investigación dentro del conjunto de las teorías existentes. Está determinado por las características y necesidades de la investigación; centra la investigación en el problema, evitando desviaciones del planteamiento original e inspira nuevas líneas y áreas de investigación. Los antecedentes, bases teóricas y definición de términos básicos son los puntos a tratar en este capítulo.

2.1 Antecedentes

Con relación al presente tema de estudio, se ha podido encontrar trabajos a nivel internacional que sirvan no solo como referencia, sino también como guía para el desarrollo.

Kelly, A. (2016), presentó en la Universidad de las Américas (UDLA) de Ecuador el trabajo de grado para optar por el título de Ingeniero en Sistemas de Computación e Informática titulado: **“Desarrollo e implantación de aplicación móvil para el control y monitorización de calentador de agua de acumulación eléctrico”**. El objetivo principal es el desarrollo de una aplicación móvil para el control y monitorización de un calentador de agua de acumulación eléctrico, una herramienta de control remoto para el hogar, utilizando el protocolo de comunicación que más se adecue al problema. El proyecto se desarrolló en base a un modelo de prototipo y la investigación se concretó siguiendo un método experimental que dio como conclusión la utilización de MQ Telemetry Transport (MQTT) como protocolo de comunicación en vista que es robusto, adaptable, tiene poco consumo de ancho de banda y realiza una transmisión de datos homogénea, y que el método petición/respuesta es el más ideal para exponer los datos de estado. Adicionalmente especifican que la metodología experimental es idónea para sistemas que interactúan con hardware como microcontroladores.

Esta investigación proporciona el proceso de desarrollo de una aplicación móvil diseñada para el control remoto de un sistema, en este caso doméstico, que facilita su acceso y manejo mejorando la relación del usuario con el sistema. También se resalta y evidencia el concepto de internet de las cosas, concepto que engloba el desarrollo e interconexión de sistemas para su comunicación, control y automatización.

De la misma manera que el trabajo anterior, se puede destacar trabajos de grado que han innovado en el país y que además han ayudado mucho en pro a la comunidad de la misma forma en que aspira el presente proyecto de investigación.

Así mismo, compañeros de la Universidad José Antonio Páez realizaron trabajos de grado que comparten la misma visión con respecto a la metodología de la investigación. Navarro, J (2014) egresado de la Universidad José Antonio Páez (UJAP), realizó un trabajo especial de grado para optar al título de Ingeniero en Computación titulado: **“Desarrollo De Un Sistema De Información Bajo Plataforma Web Y Android Para El Soporte De La Gestión De Condominio”**. La investigación tiene como objetivo desarrollar un Sistema de Información bajo Plataforma Web y Android para el soporte de la gestión de condominios.

Para ello se realizó el diagnóstico de la situación actual para determinar los requerimientos del sistema, se diseñó las interfaces y base de datos, para su desarrollo de programación preprocesador hipertexto, se utilizó MySQL como gestor de base de datos, así mismo se utilizó el servidor Apache y se aplicó la metodología de desarrollo de software Programación Extrema (XP).

Esta investigación sirve como base para la metodología a seguir cuando se desarrolla un sistema multiplataforma con base de datos. El autor escogió Programación Extrema “Porque es una metodología diseñada para desarrollar aplicaciones en corto periodo de tiempo”. Así mismo, las fases de la metodología XP se ajustan perfectamente con las necesidades de la presente investigación.

Finalmente, Morales, R. y González J. (2013), egresados de la Universidad Rafael Urdaneta (URU), realizaron un trabajo especial de grado para optar al título de Ingeniero en Computación titulado: **“Control del tráfico vehicular por medio de**

semáforos inteligentes”. El propósito de la investigación se centra en desarrollar un sistema de semáforos inteligentes para el control del tráfico vehicular basado en hardware programado en lenguajes de alto nivel compilados enfocado para la vida cotidiana en la ciudad. La investigación se fundamentó en un diseño descriptivo mientras que para la fase de diseño y desarrollo se utilizó la metodología Open Up.

El sistema se desarrolló utilizando un Kit Arduino basado en un pequeño pic microcontrolador programable. El resultado obtenido fue la creación de un sistema de control de semáforos inteligentes para el control de tráfico vehicular implementado en la ciudad de Maracaibo que concluyó en una mejoría del flujo del tráfico gracias a la capacidad de detección de vehículos y toma de decisiones que dan prioridad a las avenidas con mayor flujo de tráfico.

La investigación referida transmite como aporte el manejo e implementación del Kit Arduino, herramienta importante en el desarrollo del sistema, y la interconexión de este con otros sistemas externos a él de forma remota. De igual forma permite visualizar la programación del microcontrolador con el fin de controlar y automatizar un sistema.

2.2 Bases teóricas

2.2.1 Automatización y control

La automatización de un sistema es el proceso de utilizar elementos tecnológicos para controlar y realizar tareas, generalmente de producción, sin la necesidad de un operador humano. Pere Ponsa y Toni Granollers en su documento titulado

nos expresan:

“La Real Academia de las Ciencias Físicas y Exactas define la automática como el conjunto de métodos y procedimientos para la sustitución del operario en tareas físicas y mentales previamente programadas. De esta definición original se desprende la definición de la automatización como la aplicación de la automática al control de procesos industriales.”

Los sistemas automatizados constan de dos partes principales: la parte operativa y la parte de mando. La parte operativa es la parte que acciona e interactúa directamente

con la máquina logrando que efectúe las operaciones deseadas, como por ejemplo: los motores. La parte de mando es el centro de procesamiento del sistema que se comunica y da las órdenes a los demás componentes del sistema.

El control se da con la manipulación de las magnitudes de un sistema denominado “Planta” a través de otro llamado “Sistema de control”, sin la intervención del operador directamente sobre los elementos de salida.

Los sistemas de control son el conjunto de dispositivos de un sistema que interactúan entre sí para lograr los objetivos de control en dicho sistema al producir una salida. La parte del sistema que proporciona las entradas al sistema de control es la que se denomina planta.

2.2.2 Domótica e Inmótica

Se conoce como domótica e inmótica al conjunto de tecnologías aplicadas para el control y automatización inteligente que permite una gestión eficiente del uso de la energía, que aporta seguridad y confort, además de comunicación entre el usuario y el sistema. La diferencia entre domótica e inmótica radica en el tamaño del local donde se realiza la instalación, siendo domótica el término para la aplicación en viviendas unifamiliares e inmótica para la aplicación en edificios no destinadas a la vivienda, como hoteles, hospitales, universidades, centros comerciales y el resto de edificios terciarios, así como también para industrias. La Asociación española de domótica e inmótica (CEDOM) nos indica que:

“La domótica es el conjunto de tecnologías aplicadas al control y la automatización inteligente de la vivienda, que permite una gestión eficiente del uso de la energía, que aporta seguridad y confort, además de comunicación entre el usuario y el sistema.”

Así mismo, CEDOM también agrega:

“La inmótica es el conjunto de tecnologías aplicadas al control y la automatización inteligente de edificios no destinados a vivienda, como hoteles, centros comerciales, escuelas, universidades, hospitales y todos los edificios terciarios, permitiendo una gestión eficiente del uso de la

energía, además de aportar seguridad, confort, y comunicación entre el usuario y el sistema.”

Un sistema domótico es capaz de recoger información proveniente de unos sensores o entradas, procesarla y emitir órdenes a unos actuadores o salidas. La red de control del sistema domótico se integra con la red de energía eléctrica y se coordina con el resto de redes con las que tenga relación: telefonía, televisión, tecnologías de la información, y redes exteriores de comunicación o información, cumpliendo con las reglas de instalación aplicables a cada una de ellas. La instalación interior eléctrica y la red de control del sistema domótico están reguladas por el Reglamento Electrotécnico para Baja Tensión (REBT). En particular, la red de control del sistema domótico está regulada por la instrucción ITC-BT-51 Instalaciones de sistemas de automatización, gestión técnica de la energía y seguridad para viviendas y edificios.

La domótica permite dar una mejor calidad de vida, facilitando el diseño de casas y hogares más humanos, más personales, polifuncionales y flexibles, además hoy en día, gracias a la evolución tecnológica, son más económicas y fáciles de usar e instalar.

Algunos beneficios de la domótica son: (a) ahorro energético, ya que gestiona el consumo de energía destinada a la iluminación, climatización y demás electrodomésticos; (b) seguridad personal y patrimonial, gracias a la instalación de cámaras de vigilancia, alarmas, sensores, y otros sistemas de seguridad como el cierre de puertas automático; (c) accesibilidad, debido a que permite el acceso remoto de las instalaciones y equipos domésticos para el envío y recepción de información por medio de dispositivos comunes como computadoras personales (PC) o teléfonos inteligentes de forma fácil e intuitiva para el usuario; (d) mantenimiento, ya que puede ordenar procesos tales como limpieza o lavado y monitorear el estado de los equipos y la infraestructura.

Por otro lado la inmótica emplea la gestión técnica de edificios (GTE) que proporciona información sobre el funcionamiento, el mantenimiento, los servicios y la gestión total de edificios; especialmente para la gestión de la energía, capacidad de

medición de registro de tendencias, y de generación de alarmas y diagnóstico del consumo de energía innecesario. Estos sistemas de gestión, automatización y control de edificios conducen en general a una mejora en la eficiencia energética y su implementación es recomendada para adquirir un conocimiento más profundo sobre el consumo energético del edificio y optimizar el funcionamiento de los sistemas energéticos.

Las funciones que tienen efecto sobre la eficiencia energética de los edificios se dividen en tres grupos: funciones de regulación automática, funciones de sistemas de automatización y control de edificios y funciones de gestión técnica de edificios.

Funciones de regulación automática: (a) regulación de calefacción y refrigeración, (b) regulación de la ventilación y del aire acondicionado, (c) control de iluminación, (d) control de persianas.

Automatización y control de edificios: (a) adaptación centralizada del sistema de automatización de edificios a las necesidades del usuario, por ejemplo, programas horarios o puntos de consigna, (b) optimización centralizada del sistema de automatización de edificios, por ejemplo, ajuste de reguladores o puntos de consigna.

Gestión técnica de edificios con funciones de eficiencia energética: (a) detección de fallos de los edificios y sus sistemas técnicos y prestación soporte para el diagnóstico de estos fallos, (b) presentación de la información sobre el consumo de energía, condiciones interiores y posibilidades de mejora.

2.2.3 Internet de las cosas

1999 Kevin Ashton inventa “Internet of Things”: El IoT es el mundo en el que cada objeto tiene una identidad virtual propia y capacidad potencial para integrarse e interactuar de manera independiente en la Red con cualquier otro individuo, ya sea una máquina (M2M) o un humano.

El Internet de las cosas se refiere a un concepto que describe la cualidad de los objetos que nos rodean para conectarse a la red y transferir información de forma autónoma y así poder establecer una comunicación con las personas u otros dispositivos a su alrededor.

Según el Grupo de soluciones empresariales basadas en Internet (IBSG, Internet Business Solutions Group) de Cisco, IdC (Internet de las Cosas) es sencillamente el punto en el tiempo en el que se conectaron a Internet más “cosas u objetos” que personas.

Esto nos dice que el concepto de Internet de las Cosas no solo está relacionado con la capacidad de los objetos comunes para tener conexión a internet, sino que también entra en juego la cantidad de estos objetos conectados al unísono en comparación con la cantidad de usuarios o personas conectadas.

El Internet de las cosas tiene como única finalidad conectar el máximo de objetos que nos rodean, entre ellos y con nosotros.

2.2.4 Señales Analógicas y Digitales

Una señal analógica es un tipo de señal generada por algún tipo de fenómeno electromagnético y que es representable por una función matemática continua en la que es variable su amplitud y periodo en función del tiempo formando una onda sinusoidal.

En la naturaleza, las señales que predominan son analógicas como por ejemplo: la variación de temperatura, la presión, la velocidad, la luz, el sonido, la energía, etc. Estas señales a menudo suelen ser transformadas en señales eléctricas, mediante el adecuado transductor, para su tratamiento electrónico.

La señal eléctrica analógica es aquella en la que los valores de la tensión o voltaje varían constantemente en forma de corriente alterna, incrementando su valor con signo eléctrico positivo (+) durante medio ciclo y disminuyendo a continuación con signo eléctrico negativo (-) en el medio ciclo siguiente.

Por otro lado, una señal digital es aquella que presenta una variación discontinua con el tiempo y que sólo puede tomar ciertos valores discretos. La señal básica es una onda cuadrada (pulsos) y las representaciones se realizan en el dominio del tiempo.

Las señales digitales no se producen en el mundo físico como tales, sino que son creadas por el hombre y utilizadas principalmente para transmitir información aplicándola en función del número de estados distintos que pueda tener o según su

naturaleza eléctrica. Si son dos los estados posibles, se dice que son binarias, si son tres, ternarias, si son cuatro, cuaternarias y así sucesivamente.

Los sistemas digitales, como por ejemplo el ordenador, usan la lógica de dos estados representados por dos niveles de tensión eléctrica, uno alto y otro bajo (High y Low, respectivamente, en inglés). Por abstracción, dichos estados se sustituyen por ceros y unos, lo que facilita la aplicación de la lógica y la aritmética binaria. Si el nivel alto se representa por 1 y el bajo por 0, se habla de lógica positiva y en caso contrario de lógica negativa.

2.2.5 Microcontroladores

Un microcontrolador es un circuito integrado que se programa para gobernar los procesos de los dispositivos conectados a él. Es un componente electrónico que se emplea para funciones de detección, regulación, control y monitoreo de aquellos dispositivos que gobierna, así como también para tareas de automatización en todo tipo de sistemas. Consiste en un sistema cerrado que está formado por una arquitectura que contiene las tres unidades básicas de un computador: una CPU, unidades de memoria, y unidades de E/S, generalmente tanto digitales como analógicas.

Un microcontrolador es capaz de realizar la tarea de muchos circuitos lógicos, conversores A/D, D/A, temporizadores, decodificadores, etc..., simplificando todo el diseño a una placa de reducido tamaño y pocos elementos. Los usuarios pueden programar estas tareas utilizando una variedad de lenguajes de programación tales como Assembly, C y C + +. Estos programas de software pueden desarrollarse para controlar y adquirir ciertas entradas, realizar cálculos a alta velocidad y analizar y generar salidas para controlar una variedad de dispositivos tales como pantallas LCD, actuadores y más.

2.2.6 Arduino

Uno de los microcontroladores más famosos por su facilidad de uso y costo es el Arduino. Según la comunidad chilena de Arduino, tenemos que

“Consiste en una plataforma electrónica de código abierto (open-source) basada en hardware y software fáciles de usar. Está pensado para

artistas, diseñadores, como hobby y para cualquiera interesado en crear objetos o entornos interactivos. Arduino puede ‘sentir’ el entorno mediante la recepción de entradas desde una variedad de sensores y puede afectar a su alrededor mediante el control de luces, motores y otros artefactos.”

Nació en el Ivrea Interaction Design Institute como una herramienta sencilla para la creación rápida de prototipos, dirigido a estudiantes sin experiencia en electrónica y programación. A medida que creció la comunidad, la junta de Arduino comenzó a adaptarse a las nuevas necesidades y desafíos, cambiando su oferta de simples tableros de 8 bits a productos para aplicaciones Internet of things (IoT), portátiles, impresión 3D y entornos incrustados.

Todas las placas de Arduino son totalmente de código abierto, lo que permite a los usuarios construirlas independientemente y modificarlas. El hardware de código abierto brinda la oportunidad de estudiar el hardware para entender cómo funciona, hacer cambios en él y compartir esos cambios. Arduino lanza todos los archivos de diseño original (Eagle CAD) para su hardware. Estos archivos están licenciados bajo una licencia Creative Commons Attribution Share-Alike, que permite tanto las obras derivadas personales como las comerciales, siempre y cuando acrediten a Arduino y liberen sus diseños bajo la misma licencia.

Las tarjetas Arduino son capaces de detectar el ambiente recibiendo entradas de muchos sensores y afectar su entorno controlando luces, motores y otros actuadores. Las tareas se realizan enviando un conjunto de instrucciones al microcontrolador en el tablero. Para ello se utiliza el lenguaje de programación Arduino (basado en el cableado), y el software Arduino (IDE), basado en el procesamiento.

El software Arduino es fácil de usar para los principiantes, pero lo suficientemente flexible para los usuarios avanzados. Se puede ejecutar en sistemas operativos Windows, Macintosh OSX y Linux. El software también es de código abierto, y crece gracias a las contribuciones de los usuarios. El código fuente para el entorno Java se publica bajo la General Public License (GPL), una de las licencias open-source ampliamente usadas en el mundo, y las bibliotecas de microcontroladores

C/C ++ están bajo la Lesser General Public License (LGPL), fundada por la Free Software Foundation que vela por la libertad de compartir y modificar el software.

Arduino proporciona un entorno de programación sencillo y claro, así como también simplifica el proceso de trabajo con los microcontroladores, todo esto a un precio relativamente barato en comparación con otras plataformas de microcontroladores. Los maestros y los estudiantes lo utilizan para construir instrumentos científicos de bajo costo, para probar los principios de química y física, o para comenzar con la programación y la robótica. Diseñadores y arquitectos construyen prototipos interactivos, músicos y artistas lo utilizan para instalaciones y para experimentar con nuevos instrumentos musicales.

2.2.7 Raspberry Pi

Consiste en un computador de placa única de bajo coste desarrollado en Reino Unido por la Fundación Raspberry Pi, con el objetivo de estimular la enseñanza de ciencias de la computación en las escuelas, así como también brindar una plataforma sencilla de utilizar para proyectos de electrónica a aficionados y entusiastas del área.

2.2.8 ESP8266

Entre los diferentes microcontroladores que se encuentran en el mercado está el ESP8266, el cual es un chip de bajo costo Wi-Fi con una capacidades TCP/IP completas y un microcontrolador, es creado y desarrollado por Espressif Systems, una empresa china con sede en Shanghai. La empresa colombiana Electronilab, explica que:

“El ESP8266 es un chip altamente integrado diseñado para las necesidades de un nuevo mundo conectado. Ofrece una solución completa y autónoma de redes Wi-Fi, lo que le permite alojar la aplicación o servir como puente entre Internet y un microcontrolador. El ESP8266 tiene potentes capacidades a bordo de procesamiento y almacenamiento que le permiten integrarse con sensores y dispositivos específicos de aplicación a través de sus GPIOs con un desarrollo mínimo y carga mínima durante el tiempo de ejecución. Su alto grado de integración en el chip permite una circuitería externa mínima, y la totalidad de la solución, incluyendo el módulo está diseñado para ocupar el área mínima en un PCB.”

Este microcontrolador empezó a ganar popularidad alrededor del año 2014, pero no existía demasiada documentación disponible. Aún así, entusiastas del área lo aclamaron por su bajo consumo de energía, costo asequible y propiedades inalámbricas.

2.2.9 ESP32

Luego en el año 2016, Espressif lanzó al mercado el ESP32, un microcontrolador de bajo costo y consumo con Bluetooth y Wi-Fi integrados que emplea un microprocesador Tensilica Xtensa LX6 de doble núcleo. Es el sucesor del microcontrolador ESP8266, ofreciendo mejor rendimiento de potencia y radiofrecuencia (RF), mayor robustez, versatilidad y confiabilidad.

El ESP32 se puede adquirir a precios muy bajos y cuenta con un consumo de energía extremadamente eficiente gracias a la combinación de varios programas propietarios y hardware optimizado. Las funciones de ahorro de energía de última generación incluyen la sincronización de reloj de resolución fina, modos de alimentación y escalado dinámico de potencia. Opera con voltajes de 2.2V a 3.6V y es capaz de funcionar confiablemente en entornos industriales con un rango de temperatura de operación de -40 °C a 125 °C.

Este microcontrolador está fabricado con un microcontrolador de 240 MHz dual core Tensilica LX6 de doble núcleo que trabaja a 240 Mhz, SRAM integrada de 520 KB, memoria flash de 16 Mbyte, dual-mode Bluetooth integrado y Wi-Fi integrado de 2.4 GHz con soporte de protocolos WEP y WPA/WPA2 PSK/Enterprise.

El ESP32 puede funcionar como un sistema autónomo completo o como un dispositivo esclavo a una MCU de host para reducir la sobrecarga de la pila de comunicación en el procesador principal de la aplicación. ESP32 puede interactuar con otros sistemas para proporcionar Wi-Fi y funcionalidad Bluetooth a través de las interfaces SPI/SDIO o I2C / UART.

2.2.10 Angular

Se trata de un framework JavaScript de código abierto, mantenido por Google, que se utiliza para crear y mantener aplicaciones web de una sola página. Su objetivo es aumentar las aplicaciones basadas en navegador con capacidad de Modelo Vista

Controlador (MVC), en un esfuerzo para hacer que el desarrollo y las pruebas sean más fáciles.

La biblioteca lee el HTML que contiene atributos de las etiquetas personalizadas adicionales, entonces obedece a las directivas de los atributos personalizados, y une las piezas de entrada o salida de la página a un modelo representado por las variables estándar de JavaScript. Los valores de las variables de JavaScript se pueden configurar manualmente, o recuperados de los recursos JSON estáticos o dinámicos.

Angular 2 recomienda el uso del lenguaje de programación de Microsoft TypeScript, el cual presenta ciertas ventajas por encima de JavaScript: Programación orientada a objetos, Tipado estático, Genéricos, entre otros. Este lenguaje no pretende ser uno separado de JavaScript, sino una capa encima del mismo, ya que el código en TypeScript es compilado a ECMAScript 6.

2.2.11 NativeScript

Otro de los framework importantes es NativeScript, un framework de código abierto para desarrollar aplicaciones para las plataformas iOS y Android. Aplicaciones construidas con NativeScript usan lenguajes de programación independientes de una plataforma, como JavaScript o TypeScript. Estas aplicaciones resultan ser totalmente nativas, ya que usan las mismas APIs como si hubiesen sido hechas en Xcode o Android Studio.

2.2.12 MongoDB

Para finalizar con las herramientas, MongoDB es un sistema de base de datos NoSQL orientado a documentos, desarrollado bajo el concepto de código abierto.

MongoDB forma parte de la nueva familia de sistemas de base de datos NoSQL. En lugar de guardar los datos en tablas como se hace en las base de datos relacionales, MongoDB guarda estructuras de datos en documentos similares a JSON con un esquema dinámico (MongoDB utiliza una especificación llamada BSON), haciendo que la integración de los datos en ciertas aplicaciones sea más fácil y rápida.

2.3 Definición de términos básicos

Condición: En el contexto de esta investigación, una condición es una serie de eventos y estados que se deben cumplir para llevar a cabo o no una tarea. Estos eventos son configurados a conveniencia por el usuario.

Demonio: En el ámbito de la informática, un demonio (también conocido como servicio) es un programa que está constantemente ejecutándose en segundo plano y no requiere interacción directa por parte del usuario.

GPIO: General Purpose Input/Output (GPIO) o Entrada/Salida de Propósito General en español, es un pin genérico en un circuito integrado cuyo comportamiento, tanto de los pines de entrada como de salida, se puede controlar por el usuario en tiempo de ejecución.

Como su nombre lo indica, al ser de propósitos generales los pines GPIO no tienen ningún propósito especial definido, y no se utilizan de forma predeterminada. Los periféricos GPIO varían muy ampliamente. En algunos casos pueden ser un grupo de pines que se pueden cambiar en grupo ya sea de entrada o de salida. En otros, cada pin puede configurarse con flexibilidad para aceptar o entregar diferentes voltajes lógicos.

Un puerto GPIO es un grupo de pines GPIO (normalmente 8 pines) dispuestos en un grupo, y se tratan como un único puerto.

Reporte: Son todos aquellos informes que se generan al momento de que el servidor evalúa una condición y toma una decisión. Estos le brindan información útil al usuario para tener una especie de bitácora donde se registran todas las acciones realizadas por el sistema.

REST API: Una Application Programming Interface (interfaz de programación de aplicaciones), es un conjunto de subrutinas, funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

REST, REpresentational State Transfer (Transferencia de Estado Representacional), es un tipo de arquitectura de desarrollo web que se apoya totalmente

en el estándar HTTP. REST nos permite crear servicios y aplicaciones que pueden ser usadas por cualquier dispositivo o cliente que entienda HTTP.

Conociendo estas dos definiciones, una REST API es un conjunto de funciones y procedimientos que emplean los métodos HTTP para proveer información de una base de datos a aplicaciones cliente.

Tarea: Conjunto de acciones a tomar cuando una condición dada se cumple. Las acciones consisten en activar o desactivar puertos de un microcontrolador según la configuración deseada por el usuario.

WebSockets: Es una tecnología que proporciona un canal de comunicación bidireccional y full-duplex sobre un único puerto TCP. Está diseñada para ser implementada en navegadores y servidores web, pero puede utilizarse por cualquier aplicación cliente/servidor.

CAPÍTULO III

MARCO METODOLÓGICO

El marco metodológico constituye el eje del trabajo de investigación donde se describen el conjunto de métodos, procedimientos, técnicas y estrategias para llevar a cabo una investigación según Arias, F. (2012).

3.1 Tipo de investigación

La investigación realizada es un proyecto especial. García, L. y Mijares, H. (2006) en su trabajo **“Normas para la elaboración y presentación de los Anteproyectos, proyectos y Trabajos de grado”**, definen un proyecto especial como:

“Creaciones tangibles, susceptibles de ser utilizadas como soluciones a problemas demostrados, o que respondan a necesidades e intereses de tipo cultural. Se incluyen en esta categoría los trabajos de elaboración de libros de textos y de materiales de apoyo educativo, el desarrollo de software y hardware, prototipos y de productos tecnológicos en general.”

3.2 Diseño de la Investigación

Según Arias, F. (2012), “El diseño de investigación es la estrategia general que adopta el investigador para responder al problema planteado. En atención al diseño, la investigación se clasifica en: documental, de campo y experimental.” (P.27).

Conforme al criterio del autor, esta investigación es de campo, Arias, F. (2012) dice al respecto “La investigación de campo es aquella que consiste en la recolección de datos directamente de los sujetos investigados, o de la realidad donde ocurren los hechos (datos primarios), sin manipular o controlar variable alguna, es decir, el investigador obtiene la información pero no altera las condiciones existentes.” (P.34). Dicho esto, se define la investigación como de campo al estar en contacto directo al punto de estudio, analizando sus causas y variables, con el fin de llegar a una conclusión.

3.3 Nivel de la investigación

Conforme a Arias, F. (2012) el nivel de investigación se refiere “al grado de profundidad con que se aborda un fenómeno u objeto de estudio” (P.23). El nivel de investigación utilizado en esta investigación es descriptivo, y esta según Arias, F (2012) consiste en “la caracterización de un hecho, fenómeno, individuo o grupo, con el fin de establecer su estructura o comportamiento.” (P.24).

3.4 Técnicas e instrumentos de recolección de datos

Arias, F. (2012) define las técnicas de recolección de datos como: “El procedimiento o forma particular de obtener datos o información” (P.67). De igual forma también define los instrumentos como: “cualquier recurso, dispositivo o formato (en papel o digital), que se utiliza para obtener, registrar o almacenar información.” (P.68). Para esta investigación las técnicas de recolección de datos que se utilizarán son: la observación directa y la entrevista.

Arias, F. (2012) describe que: “La observación es una técnica que consiste en visualizar o captar mediante la vista, en forma sistemática, cualquier hecho, fenómeno o situación que se produzca en la naturaleza o en la sociedad, en función de unos objetivos de investigación preestablecidos” (P.69). En base a esto se optará por una observación directa simple para obtener información sobre los diversos sistemas de control de circuitos electrónicos actuales en el mercado y sobre el desempeño que tienen las personas manipulando este tipo de sistemas. Como instrumento se utilizará una libreta o cuaderno de notas para registrar los datos obtenidos.

Arias, F. (2012) menciona que: “La entrevista, más que un simple interrogatorio, es una técnica basada en un diálogo o conversación ‘cara a cara’, entre el entrevistador y el entrevistado acerca de un tema previamente determinado, de tal manera que el entrevistador pueda obtener la información requerida” (P.73). Adicionalmente, contempla la una entrevista semi-estructurada: “Aun cuando existe una guía de preguntas, el entrevistador puede realizar otras no contempladas inicialmente. Esto se debe a que una respuesta puede dar origen a una pregunta adicional o extraordinaria. Esta técnica se caracteriza por su flexibilidad” (P.74). El

autor plantea una entrevista semi-estructurada para recolectar datos de expertos de la materia que incumbe esta investigación para poder así conocer opiniones y casos de usos deseados por posibles usuarios del sistema.

3.5 Población y Muestra

De la palabra de Arias, F. (2012) “La población, o en términos más precisos población objetivo, es un conjunto finito o infinito de elementos con características comunes para los cuales serán extensivas las conclusiones de la investigación. Ésta queda delimitada por el problema y por los objetivos del estudio.” (P.81). Así mismo define población infinita como “aquella en la que se desconoce el total de elementos que la conforman, por cuanto no existe un registro documental de éstos debido a que su elaboración sería prácticamente imposible.” (P.82). El objetivo de esta investigación engloba a profesionales y usuarios del área de electrónica y automatización, la cual es una población con cantidad desconocida de individuos, por lo tanto se considera infinita.

Adicionalmente, Arias, F. (2012) define la muestra como “un subconjunto representativo y finito que se extrae de la población accesible” (P.83). El autor considera prudente utilizar un muestreo no probabilístico, la cual es definida por Arias, F. (2012) como “un procedimiento de selección en el que se desconoce la probabilidad que tienen los elementos de la población para integrar la muestra” (P.85). Para esto, el autor aplicará un muestreo intencional, el cual Arias, F. (2012) establece como “los elementos son escogidos con base en criterios o juicios preestablecidos por el investigador.” (P.85).

3.6 Técnicas de análisis de datos

Arias, F. (2012) describe esta fase como: “las técnicas lógicas (inducción, deducción, análisis, síntesis), o estadísticas (descriptivas o inferenciales), que serán empleadas para descifrar lo que revelan los datos que sean recogidos” (P.111). Para el análisis y procesamiento de los datos se utilizará la técnica análisis.

Primero se analizarán los datos obtenidos en la observación que deberán dar un mejor punto de vista de la población y el desarrollo del sistema para luego, en base a

un juicio personal, establecer los diseños y formación de controles más convenientes para la población. Posteriormente se procederá a analizar los resultados que se obtendrán en la entrevista y así conocer el trato que presenta la población con respecto al desarrollo de la solución para plantear posibles mejoras.

3.7 Fases Metodológicas

La metodología que será aplicada para el desarrollo de la presente investigación es la metodología ágil XP. Según Joskowicz, J. (2008) en su trabajo Reglas y Prácticas en eXtreme Programming, esta metodología “surge como una nueva manera de encarar proyectos de software, proponiendo una metodología basada esencialmente en la simplicidad y agilidad”.

Esta metodología fue introducida por Kent Beck a finales del año 1998 mientras trabajaba en Chrysler Comprehensive Compensation System (C3). Luego en 1999 publicó el primer libro sobre esta metodología titulado Extreme Programming Explained: Embrace Change.

La metodología XP tiene un conjunto importante de reglas y prácticas. En forma genérica, se pueden agrupar en: Planificación, Diseño, Desarrollo y Pruebas.

- Fase 1: Planificación

Diagnosticar la necesidad del seguimiento y control de tareas de automatización a través de circuitos electrónicos utilizando microcontroladores inalámbricos para el establecimiento de información útil sobre el uso de esta tecnología en ambientes controlados utilizando una entrevista semi-estructurada dirigida a profesionales del área.

Así mismo, se requiere identificar los requerimientos para el control de tareas de automatización que hacen uso de circuitos electrónicos utilizando microcontroladores inalámbricos en ambientes controlados.

Por último, determinar la factibilidad del desarrollo de un gestor para el control de tareas de automatización de circuitos electrónicos utilizando microcontroladores inalámbricos previamente seleccionando el modelo más pertinente a través de un estudio comparativo descriptivo.

- **Fase 2: Diseño**

Diseñar el sistema de gestión y monitoreo para la automatización de circuitos electrónicos haciendo uso de herramientas de la ingeniería de software; como lo son los casos de uso, diagramas de estado, diseño y desarrollo de componentes, entre otros.

- **Fase 3: Desarrollo**

Desarrollar un gestor para el control de tareas de automatización de circuitos electrónicos empleando microcontroladores inalámbricos en las plataformas web y móvil, basándose en el diseño generado en las fases anteriores.

- **Fase 4: Pruebas**

Despliegue del sistema en un caso de uso controlado, para probar la integración de los componentes y diagnosticar alguna falla o comportamiento no deseado. Adicionalmente, corregir estas situaciones no deseadas en el sistema.

CAPÍTULO IV

RESULTADOS

En este capítulo se describen los resultados obtenidos durante las cuatro etapas planteadas en el desarrollo metodológico de la investigación.

4.1. Fase 1: Planificación

En esta fase se realizaron diferentes estudios para determinar necesidades y posibles casos de uso de los usuarios finales del sistema, para comprender su complejidad y tenerlos presente durante el diseño de las diferentes partes que conforman la arquitectura del sistema, como lo son la aplicación web y móvil, servidor, lógica del microcontrolador, etc.

4.1.1. Entrevista

Se llevó a cabo una entrevista semi-estructurada dirigida a profesionales del área de automatización industrial y domótica para obtener cierta información relevante para la investigación, con una muestra de 10 individuos. A continuación se presentan un análisis cualitativo acorde a la similitud de las respuestas dadas en cada una de las preguntas realizadas en las entrevistas, obteniendo lo siguiente:

Pregunta 1: ¿Considera necesario el seguimiento y control de tareas utilizando instrumentos de fácil acceso y bajo costo? ¿Por qué?

Las respuestas plasmadas en el instrumento, reveló que la totalidad de los individuos entrevistados respondieron de manera positiva. Siempre se busca el balance económico en un proyecto.

Pregunta 2: ¿Qué variables maneja usted comúnmente en sus sistemas de medición y automatización? Indique su naturaleza e instrumentos de medición o sensores utilizados.

Se obtuvo de acuerdo a una tendencia significativa, que los profesionales en el área de automatización de procesos por lo general realizan mediciones de temperatura utilizando termocuplas de tipo K, medidas de presión utilizando manómetros de tipo

Bourdon y transmisores de presión capacitivo, medición de nivel utilizando ultrasonido y medición de caudal utilizando transmisores de diferencial flujo de placa de orificio.

Pregunta 3: ¿Cuál es el presupuesto promedio de que dispone para un proyecto de automatización y monitoreo?

Un gran número de los entrevistados, indicaron que los costos varían dependiendo del alcance y las necesidades del proyecto, sin embargo, proporcionaron posibles costos al momento de realizar una automatización que oscilaron entre ochocientos a un millón de dólares.

Pregunta 4: Comente brevemente un caso de un sistema de automatización y monitoreo que haya usado en su entorno laboral.

De acuerdo a los expresados por la mayoría de los entrevistados, se puede inferir que estas personas emplean sistemas automatizados que puedan realizar el control y monitoreo de procesos a fin de garantizar el buen funcionamiento de los equipos involucrados en las fases de un sistema de producción, así como también garantizar la seguridad e integridad del personal.

Pregunta 5: ¿Qué conocimientos posee de ingeniería de software? (Por ejemplo desarrollo de software multiplataforma, diseño de base de datos, manejo de servidores, etc)

Realizado el análisis de la entrevista, se evidenció que la totalidad de los individuos entrevistados no poseen conocimientos de ingeniería de software. Muchos manejan el área de programación a nivel de PLC o electrónica.

4.1.2 Definición de requerimientos del sistema

Tras haber realizado la entrevista y observaciones a se procedió a la elaboración de los requerimientos funcionales y no funcionales del sistema, con el objetivo de responder a la necesidad de los profesionales del área.

Requerimientos funcionales:

Agregar, modificar, editar y eliminar microcontroladores al sistema

Agregar, modificar, editar y eliminar tareas de automatización al sistema

Generar y mostrar reportes en base a las tareas ejecutadas

Realización de una aplicación cliente en las plataformas web y móvil para visualizar y manejar la información del sistema

Conectar microcontroladores de forma inalámbrica al sistema para la ejecución de tareas

Definición y escritura de lógica del microcontrolador para llevar a cabo las tareas de automatización y monitoreo

Requerimientos no funcionales:

Interfaz cómoda, ágil e intuitiva

Rapidez y estabilidad del servicio

Fácil instalación y configuración del servicio en el microcontrolador

Seguridad, integridad y confiabilidad de la data

4.1.3 Estudio comparativo descriptivo

Para garantizar el óptimo rendimiento del sistema, es de gran importancia estudiar la selección del microcontrolador a usar. A continuación se presentan las 4 opciones evaluadas:

Tabla 1. Comparación de microcontroladores

	ESP8266	ESP32
Plataforma	RTOS/Arduino	ESP-IDF/Arduino
Costo	\$9	\$12
Beneficios	<ul style="list-style-type: none"> - Compacto - Gran cantidad de documentación disponible - Bajo consumo de energía - Opción de menor costo 	<ul style="list-style-type: none"> - Compacto - Incluye conectividad bluetooth - Gran cantidad de puertos disponibles - Bajo consumo de energía
Desventajas	<ul style="list-style-type: none"> - Fabricado por un tercero 	<ul style="list-style-type: none"> - Poco soporte disponible - Fabricado por un tercero
	Arduino Uno WiFi	Raspberry Pi Zero W
Plataforma	Arduino	Linux
Costo	\$36,85	\$25
Beneficios	<ul style="list-style-type: none"> - Gran cantidad de documentación disponible - Producto oficial de Arduino 	<ul style="list-style-type: none"> - Compacto - Gran cantidad de documentación disponible - Incluye conectividad bluetooth - Gran cantidad de puertos disponibles
Desventajas	<ul style="list-style-type: none"> - Voluminoso - Alto consumo de energía - Poca cantidad de puertos disponibles - Opción de mayor costo 	<ul style="list-style-type: none"> - Alto consumo de energía

Fuente: Samuel Murillo (2017)

El ESP8266 y ESP32 son desarrollados por la empresa Espressif Systems. Ambos son microcontroladores de bajo consumo que buscan incluir la mayor cantidad de funciones en el menor tamaño posible y a bajo costo. Una gran ventaja que estos ofrecen es su compatibilidad con la plataforma Arduino, donde constantemente desarrolladores contribuyen extensiones y construyen una comunidad con millones de usuarios activos.

Lo que diferencia a estos dos microcontroladores es su soporte y comunidad: el ESP8266 tiene una mayor cantidad de extensiones disponibles y desarrolladores trabajando con él en comparación al ESP32. Ciertas tecnologías no están disponibles aún para este último o aún están en fase de prueba, sin una fecha en la que estará disponible. Esto se debe a la cantidad de tiempo que cada uno lleva disponible en el mercado: El ESP8266 fue lanzado a mediados del año 2014 mientras que el ESP32 a finales del 2016.

El Arduino Uno WiFi es una placa usada comúnmente para proyectos educativos. Su tamaño la hace de fácil manejo para inexpertos, pero poco práctica para proyectos comerciales. Este microcontrolador está diseñado ensamblar prototipos y el aprendizaje de electrónica, mas no para proyectos comerciales.

Por último, se evaluó el Raspberry Pi W. Si bien no es un microcontrolador sino un computador de placa única, muchos proyectos realizados en una placa Arduino son fácilmente trasladables a un Raspberry Pi. Es compacto en comparación al Arduino Uno, ofreciendo la misma variedad de interfaces que el Raspberry Pi original en una reducida presentación.

Luego de llevar a cabo el estudio comparativo, se pudo determinar que la opción más adecuada es el ESP8266, ya que ofrece un balance entre versatilidad, cualidades positivas y soporte, además de ser la opción de menor costo. Haber escogido este microcontrolador brindó la posibilidad de reducir la cantidad de horas-hombre empleadas en el desarrollo de la lógica del mismo, permitiendo así invertir más horas en la interfaz gráfica destinada al usuario.

4.2. Fase 2: Diseño

4.2.1 Componentes del sistema

La fase de diseño comprendió la estructuración de la arquitectura del sistema y la definición de los componentes que integran el mismo. A continuación se muestra un diagrama que ilustra los componentes y su interacción entre ellos:

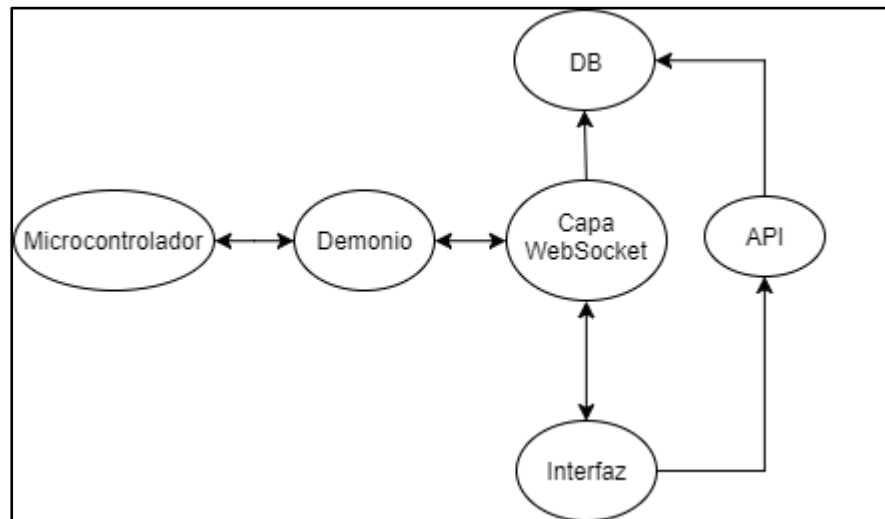


Figura 1. Componentes del sistema

Fuente: Samuel Murillo (2017)

La ilustración anterior refleja los 6 componentes que integran el sistema:

la cual se encarga de notificar un cambio de estado de algún puerto y de activar o desactivar los puertos designados en las tareas creadas por el usuario.

un componente crucial para el funcionamiento del sistema. Este se encarga de la intercomunicación del servidor y el microcontrolador. Está corriendo en segundo plano en el mismo servidor donde se aloja la base de datos y la API. Éste se encarga de evaluar si una tarea se debe ejecutar si las condiciones son las apropiadas.

un componente habitual en cualquier sistema moderno. Este se encarga de interconectar las interfaces gráficas con la base de datos mediante el protocolo HTTP siguiendo los estándares REST.

que se encarga de recibir las solicitudes del demonio y del microcontrolador y redirigirlas donde sea necesario. Esta reside en el mismo servidor que la base de datos y la API. Cabe destacar que tanto el demonio como el microcontrolador se comunican usando el protocolo WebSocket.

componente típico de cualquier sistema. Se encarga de alojar la data de forma permanente.

comprende la aplicación web y móvil. Esta se encarga de brindar un entorno amigable para toda la configuración del sistema. Hace solicitudes a la API y a la capa WebSocket.

4.2.2 Casos de uso

Los casos de uso son una herramienta usada en el Lenguaje Unificado de Modelado (UML) que permite mostrar las diferentes acciones que pueden tomar los diferentes entidades (actores) dentro del sistema. A continuación se muestran los diagramas correspondientes a cada actor y sus respectivas descripciones:

Persona que desea monitorear y/o automatizar una tarea. Interactúa con el sistema por medio de una aplicación cliente, sea en la plataforma web o móvil. Se encarga de agregar la información necesaria de las tareas y microcontroladores en el sistema.

Programa que monitorea la actividad de los microcontroladores y toma decisiones a partir de sus estados y cambio de los mismos. Se encarga de iniciar o detener tareas e informar al servidor de dichas actividades.

Este está programado para activar o desactivar físicamente sus puertos cuando el demonio se lo indique, así como también informarle cuando un puerto cambió de estado.

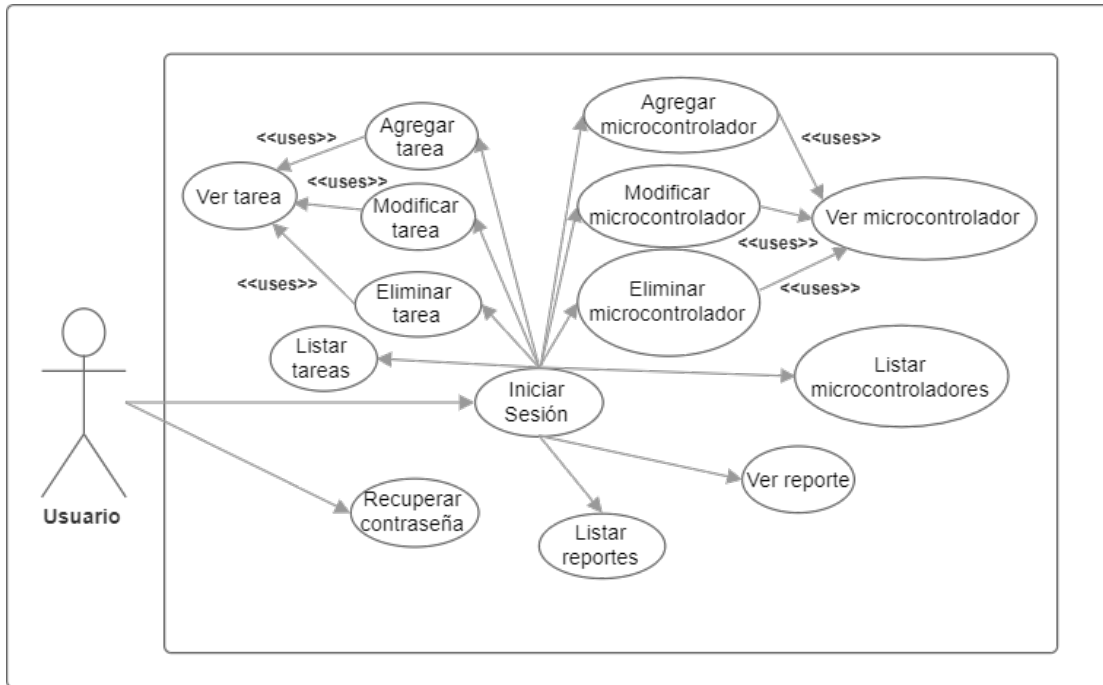


Figura 2. Casos de uso Usuario

Fuente: Samuel Murillo (2017)

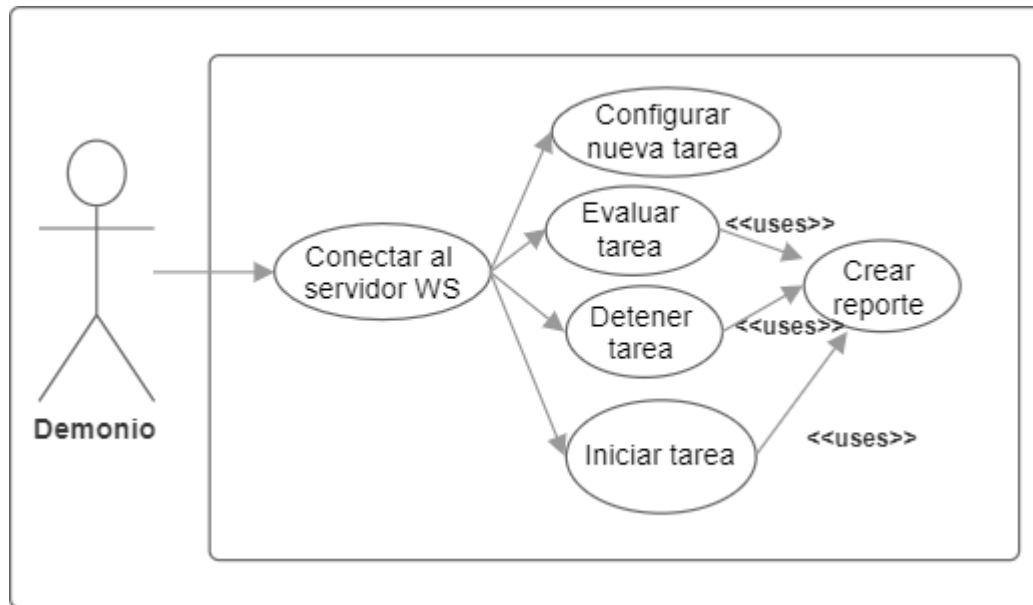


Figura 3. Casos de uso Demonio

Fuente: Samuel Murillo (2017)

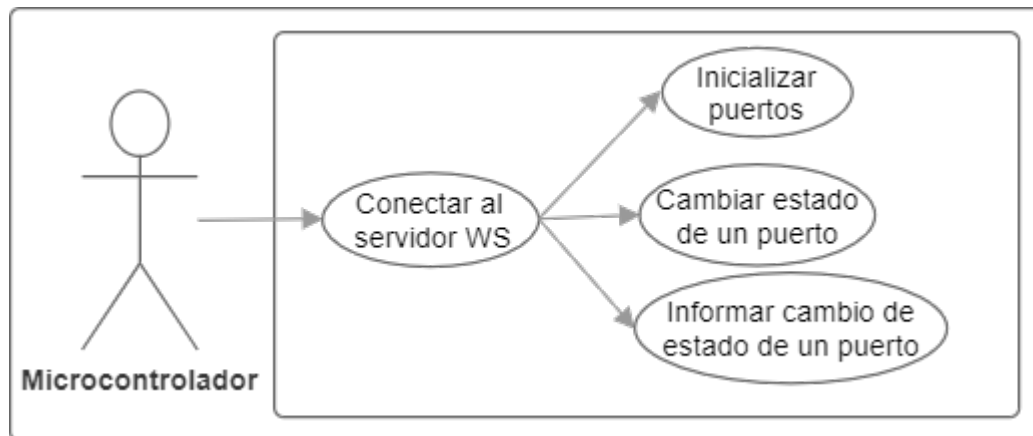


Figura 4. Casos de uso Microcontrolador

Fuente: Samuel Murillo (2017)

Tabla 2. Caso de uso: Iniciar sesión

Iniciar sesión	
Actor: Usuario	
Objetivo: Autenticarse en el sistema	
Precondición: El servidor debe de estar disponible	
Flujo normal: 1. Ingresar un correo electrónico y contraseña en los campos mostrados 2. El sistema verifica los datos 3. Se permite el ingreso al sistema	Flujo alternativo: 1. Los datos ingresados en los campos son incorrectos o poseen errores 2. Se muestran los errores al usuario

Fuente: Samuel Murillo (2017)

Tabla 3. Caso de uso: Recuperar contraseña

Recuperar contraseña	
Actor: Usuario	
Objetivo: Cambiar la contraseña en caso de no recordarla	
Precondición: El servidor debe de estar disponible	
Flujo normal: 1. Ingresar un correo electrónico en el campo mostrado 2. El sistema verifica los datos 3. El sistema envía un email al correo introducido si fue encontrado en la base de datos	Flujo alternativo: 1. Los datos ingresados en los campos son incorrectos o poseen errores 2. Se muestran los errores al usuario

Fuente: Samuel Murillo (2017)

Tabla 4. Caso de uso: Ver tarea

Ver tarea	
Actor: Usuario	
Objetivo: Visualizar los detalles de una tarea	
Precondición: Estar autenticado	
Flujo normal: 1. Seleccionar la categoría “Tareas” 2. Seleccionar una tarea de la lista mostrada	Flujo alternativo:

Fuente: Samuel Murillo (2017)

Tabla 5. Caso de uso: Listar tareas

Listar tareas	
Actor: Usuario	
Objetivo: Visualizar todas las tareas disponibles	
Precondición: Estar autenticado	
Flujo normal: 1. Seleccionar la categoría “Tareas” 2. El sistema muestra la lista de tareas en el servidor	Flujo alternativo:

Fuente: Samuel Murillo (2017)

Tabla 6. Caso de uso: Agregar tarea

Agregar tarea	
Actor: Usuario	
Objetivo: Adicionar una tarea en el microcontrolador deseado	
Precondición: Estar autenticado	
<p>Flujo normal:</p> <ol style="list-style-type: none"> 1. Seleccionar la categoría “Tareas” 2. Seleccionar la opción “Nueva tarea” 3. El usuario provee la información referente de la tarea: nombre, microcontrolador, condiciones y los puertos involucrados 4. El servidor procesa, valida y guarda la información recibida 	<p>Flujo alternativo:</p> <ol style="list-style-type: none"> 1. Los datos ingresados en los campos son incorrectos o poseen errores 2. Se muestran los errores al usuario

Fuente: Samuel Murillo (2017)

Tabla 7. Caso de uso: Modificar tarea

Modificar tarea	
Actor: Usuario	
Objetivo: Cambiar la información de una tarea	
Precondición: Estar autenticado	
<p>Flujo normal:</p> <ol style="list-style-type: none"> 1. Seleccionar la categoría “Tareas” 2. Seleccionar una tarea de la lista mostrada 3. Seleccionar la opción de modificar 4. Modificar la información de la tarea por medio del formulario de Agregar tarea 	<p>Flujo alternativo:</p> <ol style="list-style-type: none"> 1. Los datos ingresados en los campos son incorrectos o poseen errores 2. Se muestran los errores al usuario

Fuente: Samuel Murillo (2017)

Tabla 8. Caso de uso: Eliminar tarea

Eliminar tarea	
Actor: Usuario	
Objetivo: Remover una tarea	
Precondición: Estar autenticado	
Flujo normal: <ol style="list-style-type: none">1. Seleccionar la categoría “Tareas”2. Seleccionar una tarea de la lista mostrada3. Seleccionar la opción de eliminar4. Confirmar la eliminación	Flujo alternativo: <ol style="list-style-type: none">1. El usuario cancela la eliminación2. Se retorna a la vista de Ver tarea

Fuente: Samuel Murillo (2017)

Tabla 9. Caso de uso: Ver microcontrolador

Ver microcontrolador	
Actor: Usuario	
Objetivo: Visualizar información de un microcontrolador	
Precondición: Estar autenticado	
Flujo normal: <ol style="list-style-type: none">1. Seleccionar la categoría “Microcontroladores”2. Seleccionar un microcontrolador de la lista mostrada	Flujo alternativo:

Fuente: Samuel Murillo (2017)

Tabla 10. Caso de uso: Listar microcontroladores

Listar microcontroladores	
Actor: Usuario	
Objetivo: Visualizar todos los microcontroladores en el sistema	
Precondición: Estar autenticado	
Flujo normal: 1. Seleccionar la categoría “Microcontroladores” 2. El sistema muestra la lista de microcontroladores en el servidor	Flujo alternativo:

Fuente: Samuel Murillo (2017)

Tabla 11. Caso de uso: Agregar microcontrolador

Agregar microcontrolador	
Actor: Usuario	
Objetivo: Agregar un nuevo microcontrolador al sistema	
Precondición: Estar autenticado	
Flujo normal: 1. Seleccionar la categoría “Microcontroladores” 2. Seleccionar la opción “Nuevo microcontrolador” 3. El usuario provee la información referente del microcontrolador: nombre, IP y descripción (opcional) 4. El servidor procesa, valida y guarda la información recibida	Flujo alternativo: 1. Los datos ingresados en los campos son incorrectos o poseen errores 2. Se muestran los errores al usuario

Fuente: Samuel Murillo (2017)

Tabla 12. Caso de uso: Modificar microcontrolador

Modificar microcontrolador	
Actor: Usuario	
Objetivo: Cambiar la información de un microcontrolador	
Precondición: Estar autenticado	
<p>Flujo normal:</p> <ol style="list-style-type: none"> 1. Seleccionar la categoría “Microcontrolador” 2. Seleccionar un microcontrolador de la lista mostrada 3. Seleccionar la opción de modificar 4. Modificar la información del microcontrolador por medio del formulario de Agregar microcontrolador 	<p>Flujo alternativo:</p> <ol style="list-style-type: none"> 1. Los datos ingresados en los campos son incorrectos o poseen errores 2. Se muestran los errores al usuario

Fuente: Samuel Murillo (2017)

Tabla 13. Caso de uso: Eliminar microcontrolador

Eliminar microcontrolador	
Actor: Usuario	
Objetivo: Remover un microcontrolador	
Precondición: Estar autenticado	
<p>Flujo normal:</p> <ol style="list-style-type: none"> 1. Seleccionar la categoría “Microcontroladores” 2. Seleccionar un microcontrolador de la lista mostrada 3. Seleccionar la opción de eliminar 4. Confirmar la eliminación 	<p>Flujo alternativo:</p> <ol style="list-style-type: none"> 1. El usuario cancela la eliminación 2. Se retorna a la vista de Ver microcontrolador

Fuente: Samuel Murillo (2017)

Tabla 14. Caso de uso: Ver reporte

Ver reporte	
Actor: Usuario	
Objetivo: Visualizar información de un reporte	
Precondición: Estar autenticado	
Flujo normal: 1. Seleccionar la categoría “Reportes” 2. Seleccionar un reporte de la lista mostrada	Flujo alternativo:

Fuente: Samuel Murillo (2017)

Tabla 15. Caso de uso: Listar reportes

Listar reportes	
Actor: Usuario	
Objetivo: Visualizar todos los reportes en el sistema	
Precondición: Estar autenticado	
Flujo normal: 1. Seleccionar la categoría “Reportes” 2. El sistema muestra la lista de reportes en el servidor	Flujo alternativo:

Fuente: Samuel Murillo (2017)

Tabla 16. Caso de uso: Conectar al servidor WebSocket

Conectar al servidor WebSocket	
Actor: Demonio, Microcontrolador	
Objetivo: Establecer conexión con el servidor mediante el protocolo WebSocket	
Precondición: Ninguna	
Flujo normal: <ol style="list-style-type: none"> 1. Se define la URL en donde está el servidor WebSocket 2. El cliente WebSocket establece conexión con el servidor 3. El servidor WebSocket responde con un identificador único para la sesión 	Flujo alternativo: <ol style="list-style-type: none"> 1. No se puede establecer conexión con el servidor WebSocket 2. El cliente WebSocket entra en estado de reposo

Fuente: Samuel Murillo (2017)

Tabla 17. Caso de uso: Configurar nueva tarea

Configurar nueva tarea	
Actor: Demonio	
Objetivo: Agregar la información necesaria en el servidor para la ejecución de una tarea	
Precondición: El demonio debe estar conectado al servidor WebSocket y el microcontrolador debe estar activo	
Flujo normal: <ol style="list-style-type: none"> 1. El demonio es notificado de que se ha agregado una nueva tarea 2. El demonio procede a establecer las condiciones fecha-hora de la tarea 3. El demonio informa al microcontrolador que un nuevo puerto debe ser configurado 	Flujo alternativo: <ol style="list-style-type: none"> 1. El microcontrolador no está activo 2. Se envía una notificación de error a la aplicación cliente

Fuente: Samuel Murillo (2017)

Tabla 18. Caso de uso: Evaluar tarea

Evaluar tarea	
Actor: Demonio	
Objetivo: Verificar que una tarea cumple con las condiciones para ser ejecutada	
Precondición: El demonio debe estar conectado al servidor WebSocket y el microcontrolador debe estar activo	
<p>Flujo normal:</p> <ol style="list-style-type: none"> 1. El demonio es notificado de que un puerto ha cambiado en el microcontrolador o que una condición fecha-hora se ha cumplido 2. El demonio procede a evaluar las demás condiciones de la tarea 3. Si las condiciones se cumplen <ol style="list-style-type: none"> 3.1. El demonio prepara una solicitud para iniciar o detener la tarea <p>De lo contrario</p> <ol style="list-style-type: none"> 3.2. Se crear un reporte con detalles del porqué no hubo un cambio de estado en la tarea 	<p>Flujo alternativo:</p>

Fuente: Samuel Murillo (2017)

Tabla 19. Caso de uso: Iniciar tarea

Iniciar tarea	
Actor: Demonio	
Objetivo: Enviar solicitud a microcontrolador para iniciar una tarea	
Precondición: El demonio debe estar conectado al servidor WebSocket y el microcontrolador debe estar activo. Se ha evaluado y aprobado la tarea anteriormente.	
Flujo normal: <ol style="list-style-type: none"> 1. El demonio envía una solicitud al microcontrolador para ejecutar la tarea 2. Se crear un reporte con detalles de la ejecución de la tarea 	Flujo alternativo:

Fuente: Samuel Murillo (2017)

Tabla 20. Caso de uso: Detener tarea

Detener tarea	
Actor: Demonio	
Objetivo: Enviar solicitud a microcontrolador para detener una tarea	
Precondición: El demonio debe estar conectado al servidor WebSocket y el microcontrolador debe estar activo. Se ha evaluado y aprobado la tarea anteriormente.	
Flujo normal: <ol style="list-style-type: none"> 1. El demonio envía una solicitud al microcontrolador para detener la tarea 2. Se crear un reporte con detalles de la detención de la tarea 	Flujo alternativo:

Fuente: Samuel Murillo (2017)

Tabla 21. Caso de uso: Crear tarea

Crear reporte	
Actor: Demonio	
Objetivo: Generar un reporte a partir de información acerca de una tarea	
Precondición: El demonio debe estar conectado al servidor WebSocket.	
Flujo normal: <ol style="list-style-type: none">1. El demonio genera un reporte a partir de un evento que lo amerite2. El reporte es guardado en la base de datos	Flujo alternativo:

Fuente: Samuel Murillo (2017)

Tabla 22. Caso de uso: Inicializar puertos

Inicializar puertos	
Actor: Microcontrolador	
Precondición: El microcontrolador debe estar conectado al servidor WebSocket	
Flujo normal: <ol style="list-style-type: none">1. El microcontrolador recibe el conjunto de tareas que tiene asignadas2. El microcontrolador itera sobre las tareas para configurar los puertos de entrada y salida según las tareas lo establezcan3. El microcontrolador informa al demonio de que este está listo para ejecutar tareas	Flujo alternativo:

Fuente: Samuel Murillo (2017)

Tabla 23. Caso de uso: Informar cambio de estado de un puerto

Informar cambio de estado de un puerto	
Actor: Microcontrolador	
Precondición: El microcontrolador debe estar conectado al servidor WebSocket	
Flujo normal: 1. Uno o más puertos de entrada en el microcontrolador cambian de estado 2. El microcontrolador informa al demonio del cambio de estado del puerto, señalando el número del puerto y el nuevo estado	Flujo alternativo:

Fuente: Samuel Murillo (2017)

Tabla 24. Caso de uso: Cambiar estado de un puerto

Cambiar estado de un puerto	
Actor: Microcontrolador	
Precondición: El microcontrolador debe estar conectado al servidor WebSocket	
Flujo normal: 1. El microcontrolador recibe una solicitud del demonio para cambiar el estado de un puerto 2. El microcontrolador procesa la información recibida 3. Se procede a cambiar el estado del puerto señalado	Flujo alternativo:

Fuente: Samuel Murillo (2017)

4.2.3 Modelo de base de datos

A continuación se presenta el modelo de base de datos utilizado. Cabe destacar que se presenta un modelo no relacional. Una de las diferencias principales entre el

modelo no relacional presentado por MongoDB es que sus entidades no son llamadas “tablas” como en bases de datos SQL, sino que son representadas como “colecciones”.

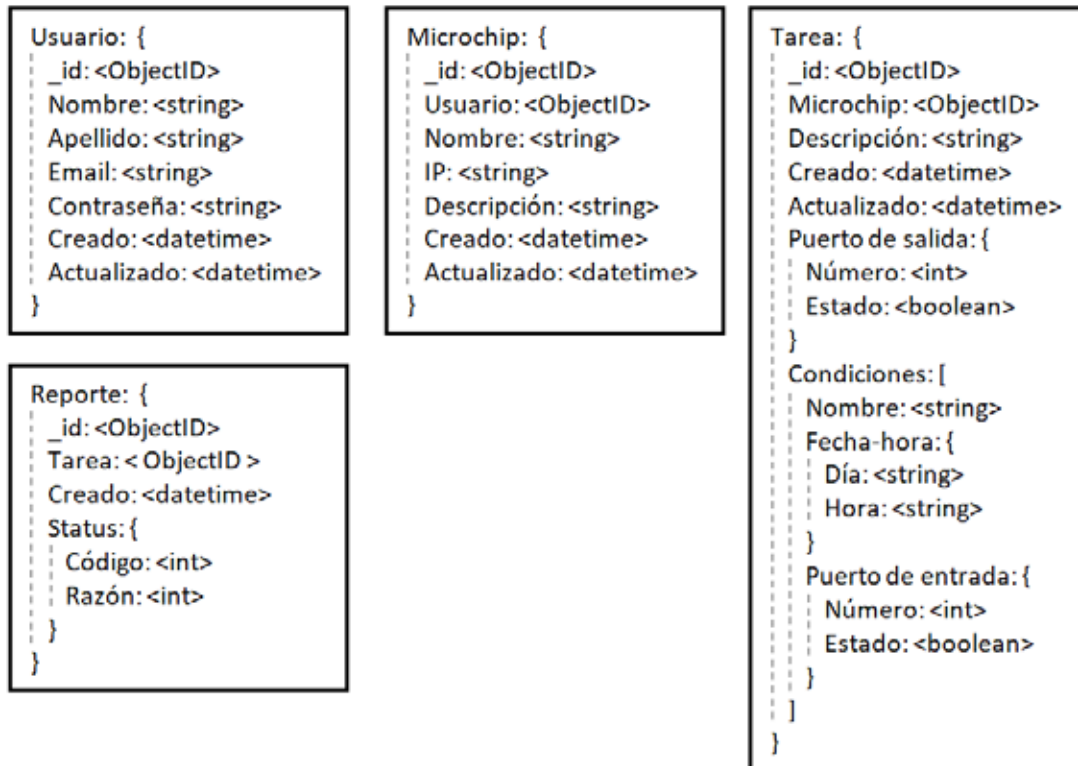


Figura 5. Modelo de base de datos

Fuente: Samuel Murillo (2017)

Tabla 25. Descripción del modelo de base de datos

Nombre de del documento	Contenido del documento	Descripción del documento
Usuario	<pre> Usuario: { _id: <ObjectID> Nombre: <string> Apellido: <string> Email: <string> Contraseña: <string> Creado: <datetime> Actualizado: <datetime> } </pre>	<p>Se almacena la información básica del usuario.</p>
Microchip	<pre> Microchip: { _id: <ObjectID> Usuario: <ObjectID> Nombre: <string> IP: <string> Descripción: <string> Creado: <datetime> Actualizado: <datetime> } </pre>	<p>Se recopila la información pertinente del microcontrolador, como el usuario quien lo agregó y la IP que lo identifica en la red local.</p>

Nombre de del documento	Contenido del documento	Descripción del documento
Tarea	<pre> Tarea: { _id: <ObjectID> Microchip: <ObjectID> Descripción: <string> Creado: <datetime> Actualizado: <datetime> Puerto de salida: { Número: <int> Estado: <boolean> } Condiciones: [Nombre: <string> Fecha-hora: { Día: <string> Hora: <string> } Puerto de entrada: { Número: <int> Estado: <boolean> }] } </pre>	<p>Información que describe la actividad a realizar en una tarea, así como también sus condiciones de acción. Pertenece a un sólo microcontrolador.</p>
Reporte	<pre> Reporte: { _id: <ObjectID> Tarea: <ObjectID > Creado: <datetime> Status: { Código: <int> Razón: <int> } } </pre>	<p>Reporte generado por el servidor a partir de algún hecho significativo en un microcontrolador.</p>

Fuente: Samuel Murillo (2017)

4.3. Fase 3: Desarrollo

En esta fase se tomó la información recaudada en las fases anteriores para llevar a la realidad el sistema propuesto. A continuación se muestran capturas de pantalla del resultado.

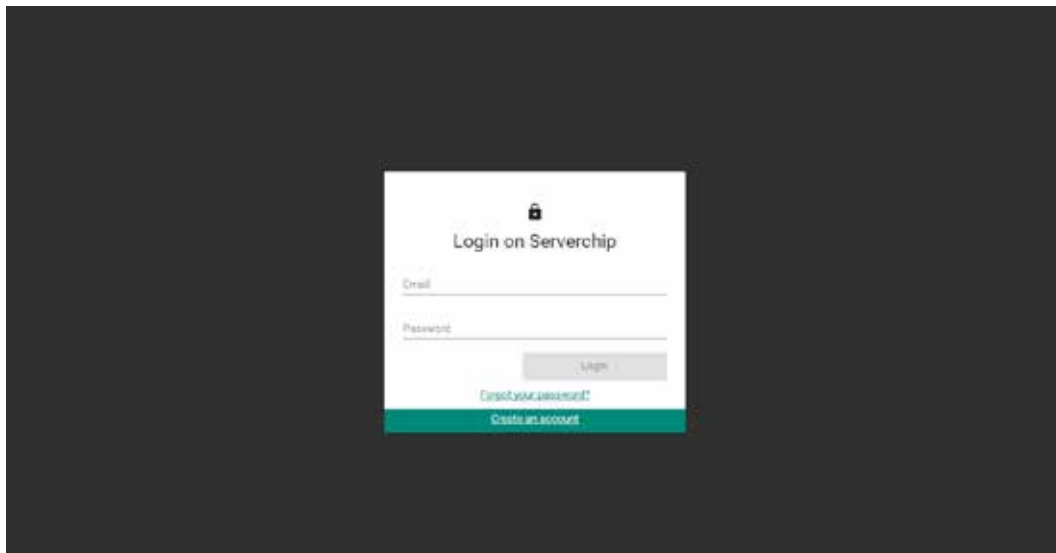


Figura 6. Página de Inicio de sesión

Fuente: Samuel Murillo (2017)

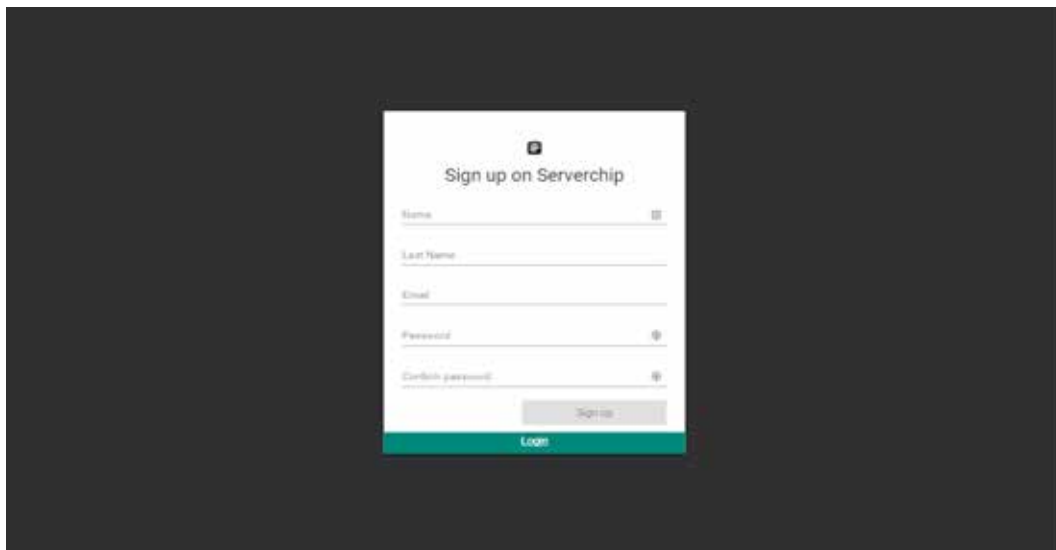


Figura 7. Formulario de registro

Fuente: Samuel Murillo (2017)

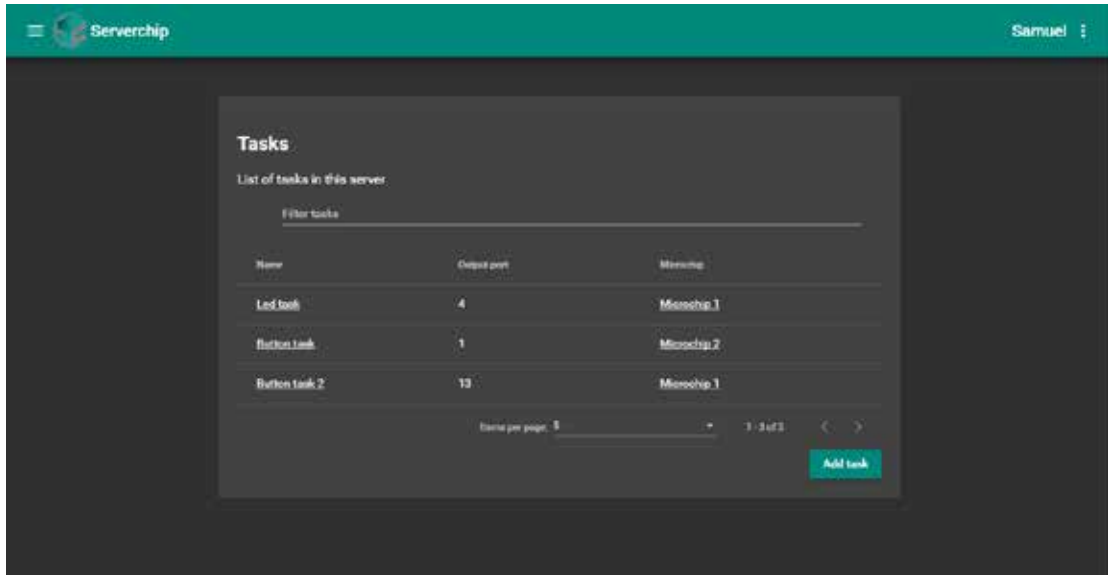


Figura 8. Lista de tareas

Fuente: Samuel Murillo (2017)

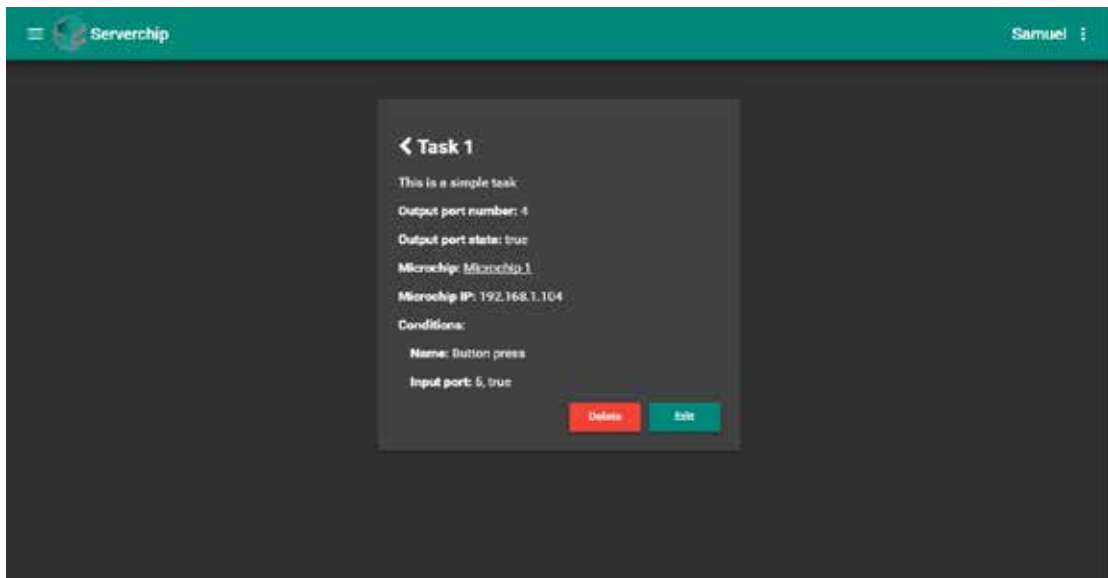


Figura 9. Ver tarea

Fuente: Samuel Murillo (2017)

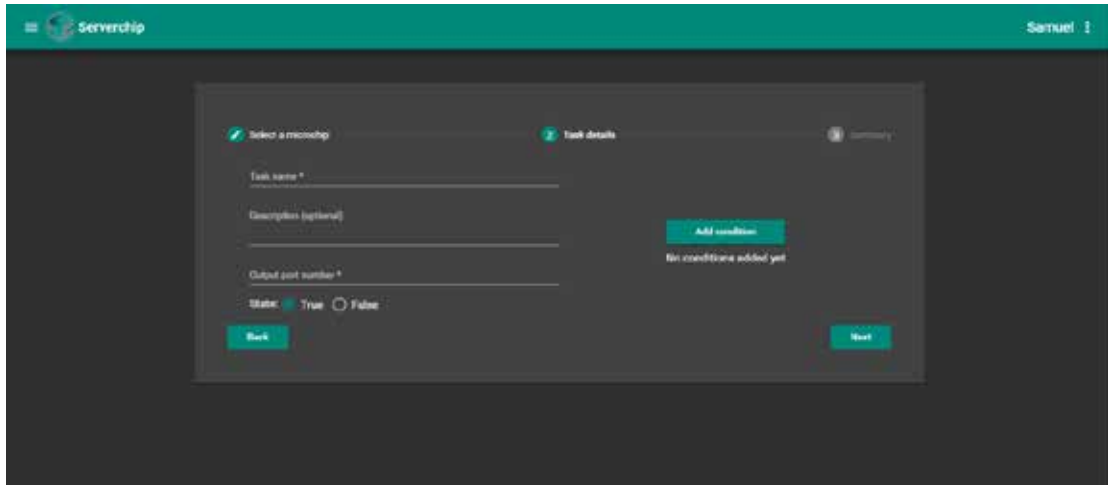


Figura 10. Crear tarea

Fuente: Samuel Murillo (2017)

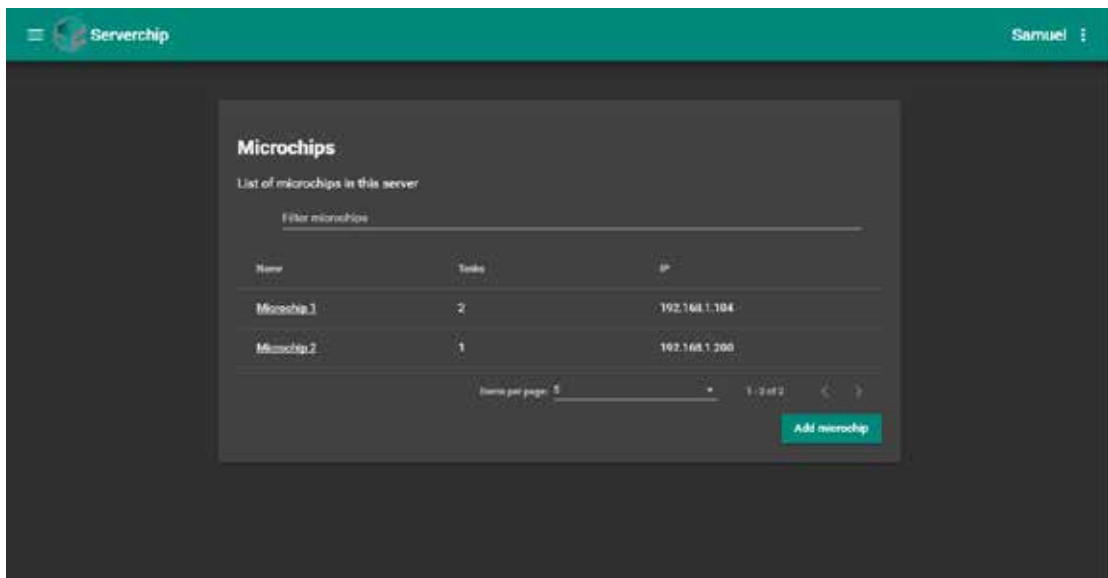


Figura 11. Lista de microcontroladores

Fuente: Samuel Murillo (2017)

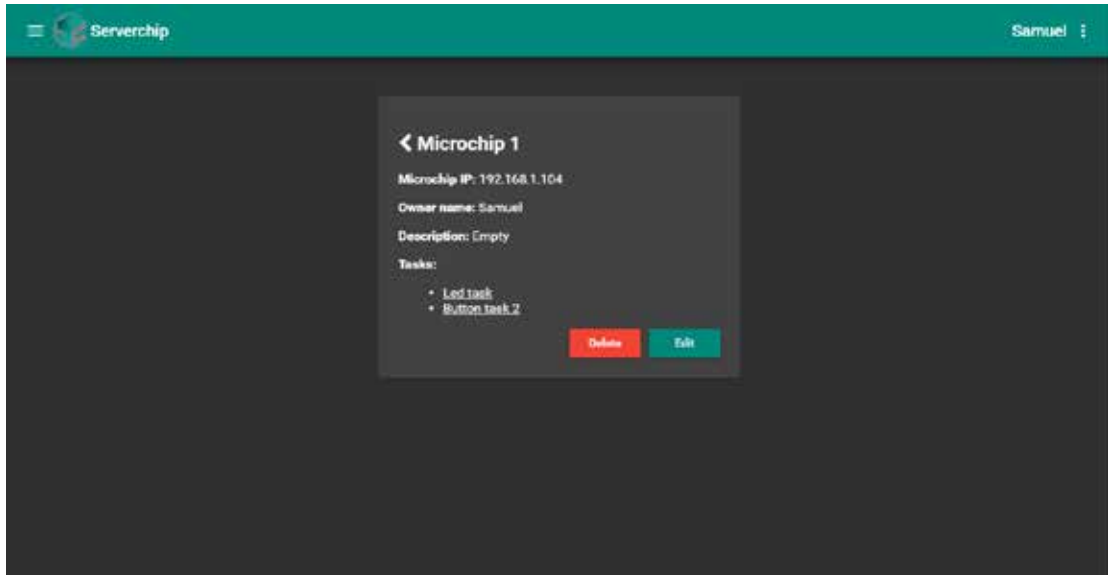


Figura 12. Ver microcontrolador

Fuente: Samuel Murillo (2017)

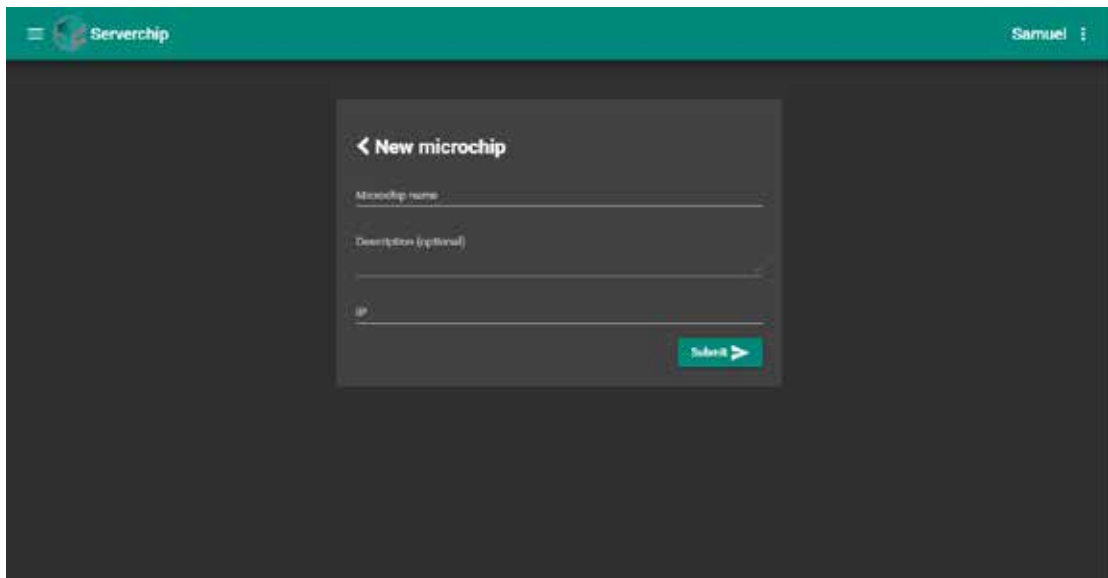


Figura 13. Agregar microcontrolador

Fuente: Samuel Murillo (2017)

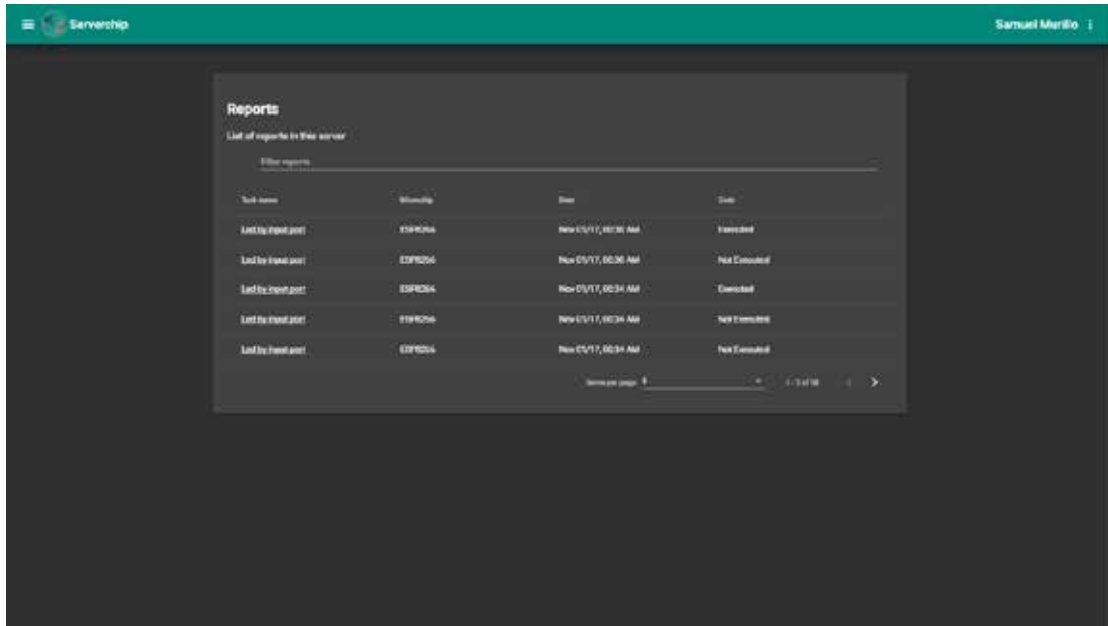


Figura 14. Listar reportes

Fuente: Samuel Murillo (2017)

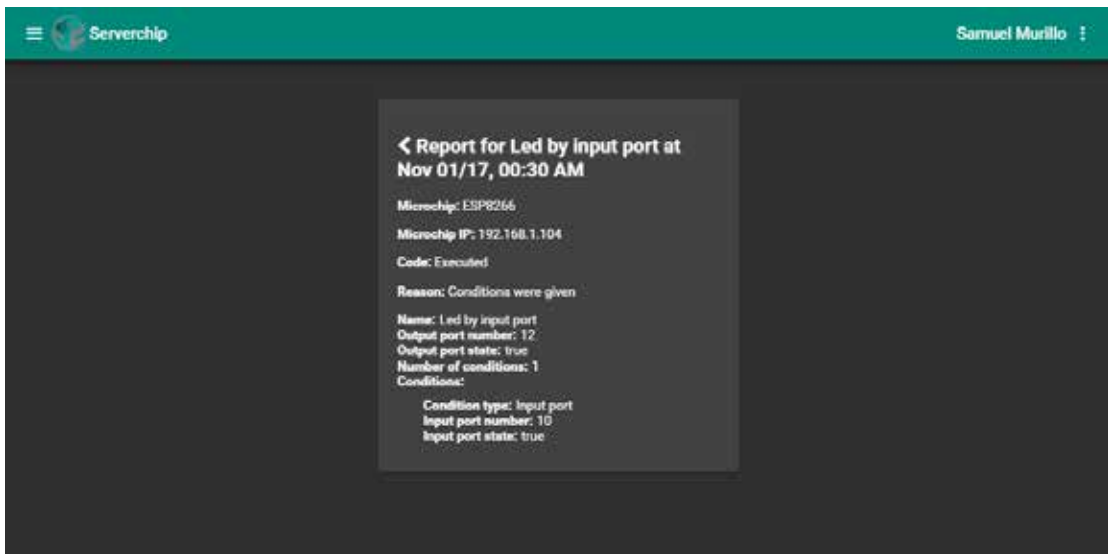


Figura 15. Ver reporte

Fuente: Samuel Murillo (2017)

4.4. Fase 4: Pruebas

En esta última fase se probó el sistema al finalizar su codificación, asegurándose de que cumpla con los requerimientos recolectados en la fase de planificación y corrigiendo imperfecciones si es necesario.

Tabla 26. Caso de prueba: Agregar microcontrolador

Caso de prueba: Agregar microcontrolador	
Descripción: Se comprobó que el usuario puede agregar un microcontrolador con sus detalles asociados.	
Tipo: Caja negra	Caso de uso: Agregar microcontrolador
Resultados: Se muestra los detalles del microcontrolador agregado	
Observaciones: Ninguna	

Fuente: Samuel Murillo (2017)

Tabla 27. Caso de prueba: Agregar tarea

Caso de prueba: Agregar tarea	
Descripción: Se comprobó que el usuario puede agregar una tarea con una condición por puerto de entrada o por condición de día-hora.	
Tipo: Caja negra	Caso de uso: Agregar tarea
Resultados: Se muestra los detalles de la tarea agregada	
Observaciones: Ninguna	

Fuente: Samuel Murillo (2017)

Tabla 28. Caso de prueba: Iniciar tarea

Caso de prueba: Iniciar tarea	
Descripción: Se comprobó que el microcontrolador informa correctamente al demonio y al servidor cuando se debe ejecutar una tarea, luego el demonio envía la señal para ejecutar dicha tarea.	
Tipo: Caja negra	Caso de uso: Iniciar tarea
Resultados: Se ejecutó la tarea en cuestión.	
Observaciones: Se puede notar una pequeña latencia mientras las condiciones se cumplen y la tarea se ejecuta.	

Fuente: Samuel Murillo (2017)

Tabla 29. Caso de prueba: Inicializar puertos

Caso de prueba: Inicializar puertos	
Descripción: Se comprobó que el microcontrolador se conecte de forma correcta al servidor WebSocket, reciba las tareas a su cargo e inicialice los puertos de dichas tareas.	
Tipo: Caja negra	Caso de uso: Inicializar puertos
Resultados: Se inicializaron los puertos de las tareas	
Observaciones: Ninguna.	

Fuente: Samuel Murillo (2017)

Tabla 30. Caso de prueba: Informar cambio de estado de un puerto

Caso de prueba: Informar cambio de estado de un puerto	
Descripción: Se comprobó que el microcontrolador informe al demonio de forma correcta cuando un puerto cambia de estado, para evaluar si se debe ejecutar una tarea o no.	
Tipo: Caja negra	Caso de uso: Informar cambio de estado de un puerto
Resultados: Se informó al demonio correctamente al cambiar un puerto de estado.	
Observaciones: Ninguna.	

Fuente: Samuel Murillo (2017)

Tabla 31. Caso de prueba Evaluar tarea

Caso de prueba: Evaluar tarea	
Descripción: Se comprobó que el demonio evalúe si una tarea se debe ejecutar cuando un puerto cambia de estado o se cumple una condición día-fecha.	
Tipo: Caja negra	Caso de uso: Evaluar tarea
Resultados: El demonio evaluó las tareas de forma correcta, declinando solicitudes cuando las condiciones no son dadas y aceptándolas cuando sea necesario.	
Observaciones: Ninguna.	

Fuente: Samuel Murillo (2017)

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

Una vez concluida la investigación de acuerdo con cada uno de los objetivos específicos se efectúan las siguientes conclusiones:

Con respecto a la fase de planificación se pudo identificar las necesidades del seguimiento y control de tareas de automatización a través de circuitos electrónicos utilizando microcontroladores inalámbrico, a fin de establecer cuál es realmente el beneficio al momento de diseñar la propuesta para el usuario final.

Dentro del mismo contexto, se determinaron los requerimientos del sistema de información, los cuales consistió en la búsqueda de una solución informática capaz de facilitar con mayor eficiencia operativa a los usuarios el seguimiento y control de actividades con un costo mínimo de implantación, minimizando tiempos de respuesta y además de mantener la integridad de la información por medio de controles de acceso al sistema.

Luego se diseñó el sistema de información automatizado donde se procedió a la creación de las interfaces, además, se establecieron como se realizaron cada uno de los proceso, así como la estructura del gestor para el control de tareas en ambientes controlados.

Con respecto a implementar y evaluar el gestor para el control de tareas de automatización de circuitos electrónicos haciendo uso de un microcontrolador inalámbrico a través de un sistema web y móvil, se logró construir un sistemas que cumpliera los requerimientos obtenidos en fases anteriores, capaz de fin de facilitar con mayor eficiencia operativa a los usuarios el seguimiento y control de actividades en ambientes controlados, por último, se realizaron las pruebas pertinentes para corroborar la efectividad y evaluar su funcionamiento.

Por otra parte, se optó usar una metodologías XP ya que a través de ella se pudo adaptar los tiempos en el desarrollo de cada unas de las fases, lo que permitió trabajar

de manera simultánea en cada una de ellas y al no encasillarse en un plan de trabajo, esto debido a los constantes cambios que se fueron presentando a lo largo del desarrollo de la propuesta.

Para finalizar, las tecnologías que más se utilizan para automatizar procesos continuos que requieren control, así como de la manipulación y del monitoreo de variables en tiempo real a través de la web, permiten apoyar a los usuarios en la interpretación de los procesos y en la detección de fallas y solución inmediata de los problemas en el sistema, mediante alarmas programadas; sin embargo, también traen como consecuencia la necesidad y los requerimientos de personal altamente calificado en éstas tecnologías para su aplicación, pero sobre todo, para desarrollar los algoritmos tanto de control y protección como los de aquellos eventos a realizarse en las GUI (interfaz gráfica de usuario), las cuales también deben desarrollarse principalmente en similitud con la realidad de los sistemas y procesos a controlar.

5.2 Recomendaciones

Durante la realización de la presente propuesta de un gestor para el control de tareas de automatización de circuitos electrónicos haciendo uso de un microcontrolador inalámbrico a través de un sistema web y móvil, se fueron evaluando y analizando las siguientes recomendaciones:

Se recomienda implantar el gestor para el control de tareas de automatización diseñado para que mejoren los procesos que se efectúan en ambientes controlados, ya que a través de su uso se puede lograr importantes avances, pues automatizan los procesos, haciendo más fácil y poco frecuente las interacciones con el usuario.

Garantizar el buen funcionamiento del sistema y la vida útil de sus componentes a través de una planificación adecuada:

Elaborar un programa de inspecciones a los sensores con una frecuencia trimestral para evaluar las condiciones de los equipos electrónicos. Dependiendo de los hallazgos y evaluaciones, ajustar los periodos de inspección. Esto constituirá el mantenimiento preventivo (solo inspección).

Elaborar un plan de mantenimiento preventivo en base a los resultados obtenidos en las inspecciones, donde se realicen los ajuste necesarios para el buen funcionamiento de los equipos electrónicos, También, mantenimiento eléctrico general donde se ejecute: ajuste de terminales de conexión, pruebas de consumo y envío de señal.

En cuanto a sistema web y móvil realizar actualizaciones pertinentes a medidas que el usuarios aumente sus requerimientos al momento de automatizar y monitorear tareas cotidianas.

Así mismo, se recomienda profundizar en estudios que implementen nuevas metodologías para el análisis de la respuesta de los sistemas de circuitos electrónicos haciendo uso de un microcontrolador inalámbrico, acorde con las nuevas tendencias mundiales dirigidas a la utilización de herramientas computacionales.

Para investigaciones futuras sobre el tema, sería muy recomendable contactar a organismos que están encargados de llevar a cabo estudios o proyectos de automatización del quehacer diario como un plan de incentivos a nuevos emprendedores.

REFERENCIAS

- Álvarez, M. y Anatole, J. (2016). **El Internet de las Cosas y la responsabilidad**.
Extraído el 13 de diciembre de 2016 desde <http://www.libremercado.com/2016-11-26/maria-alvarez-y-javier-anatole-el-internet-de-las-cosas-y-la-responsabilidad-80633/>
- Arduino (n.d.). **¿QUÉ ES ARDUINO?** Extraído el 13 de diciembre de 2016 desde <http://arduino.cl/que-es-arduino/>
- Arias, F (2012). **El Proyecto de Investigación: Introducción a la metodología científica**. Sexta edición. Editorial Episteme.
- Canto, C. (n.d.). **Arquitectura de microcontroladores**. Facultad de Ciencias. UASLP.
Extraído el 11 de diciembre de 2016 desde http://galia.fc.uaslp.mx/~cantocar/microcontroladores/SLIDES_8051_PDF/2_INTROD.PDF
- CEDOM (n.d.). **Qué es Domótica**. Extraído el 13 de diciembre de 2016 desde <http://www.cedom.es/sobre-domotica/que-es-domotica>
- CEDOM (n.d.). **Qué es Inmótica**. Extraído el 13 de diciembre de 2016 desde <http://www.cedom.es/sobre-domotica/que-es-inmotica>
- Electronilab (n.d.). **NodeMCU**. Extraído el 24 de mayo de 2017 desde <https://electronilab.co/tienda/nodemcu-board-de-desarrollo-con-esp8266-wifi-y-lua/>
- Evans, D. (2011).

http://www.cisco.com/c/dam/global/es_mx/solutions/executive/assets/pdf/internet-of-things-iot-ibsg.pdf
- González, J. (2016). **Angular 2 Wiki**. Extraído el 14 de diciembre de 2016 desde <https://www.barbarianmeetscoding.com/wiki/angular-2/>
- Joskowicz, J. (2008). **Reglas y Prácticas en eXtreme Programming**. Universidad de Vigo. <https://iie.fing.edu.uy/~josej/docs/XP%20-%20Jose%20Joskowicz.pdf>

- Kelly, Rafael (2016). **Desarrollo e implantación de aplicación móvil para el control y monitorización de calentador de agua de acumulación eléctrico**. Trabajo de titulación. Universidad de las Américas de Ecuador.
- Morales, R. y González J. (2013). **Control del tráfico vehicular por medio de semáforos inteligentes**. Trabajo de grado. Universidad Rafael Urdaneta.
- Navarro, Johan (2014). **Desarrollo de un sistema de información bajo plataforma web y android para el soporte de la gestión de condominio**. Trabajo de grado. Universidad José Antonio Páez.
- Pere Ponsa, Toni Granollers (n.d.). **Diseño y automatización industrial**. Extraído el 24 de mayo de 2017 desde <https://www.epsevg.upc.edu/hcd/material/lecturas/interfaz.pdf>
- Ramirez, F. (2008). **Señales analógicas y digitales**. Extraído el 11 de diciembre de 2016 desde <http://www.tuelectronica.es/tutoriales/telecomunicaciones/senales-analogicas-y-digitales.html>
- UPV/EHU (2001). **Automatización**. Extraído el 11 de diciembre de 2016 desde <http://www.sc.ehu.es/sbweb/webcentro/automatica/WebCQMH1/PAGINA%20PRINCIPAL/Automatizacion/Automatizacion.htm>
- Wikimedia Foundation (n.d.). **Raspberry Pi**. Extraído el 24 de mayo de 2017 desde https://en.wikipedia.org/wiki/Raspberry_Pi
- Wikimedia Foundation (n.d.). **ESP8266**. Extraído el 24 de mayo de 2017 desde <https://en.wikipedia.org/wiki/ESP8266>
- Wikimedia Foundation (n.d.). **ESP32**. Extraído el 11 de diciembre de 2016 desde <https://en.wikipedia.org/wiki/ESP32>
- Wikimedia Foundation (n.d.). **Interfaz de programación de aplicaciones**. Extraído el 15 de diciembre de 2016 desde https://es.wikipedia.org/wiki/Interfaz_de_programaci%C3%B3n_de_aplicaciones
- Wikimedia Foundation (n.d.). **GPIO**. Extraído el 11 de diciembre de 2016 desde <https://es.wikipedia.org/wiki/GPIO>

Wikimedia Foundation (n.d.). **NativeScript**. Extraído el 14 de diciembre de 2016 desde <https://en.wikipedia.org/wiki/NativeScript>

Wikimedia Foundation (n.d.). **WebSocket**. Extraído el 15 de diciembre de 2016 desde <https://es.wikipedia.org/wiki/WebSocket>

Wikimedia Foundation (n.d.). **Programación extrema**. Extraído el 24 de mayo de 2017 desde https://es.wikipedia.org/wiki/Programaci%C3%B3n_extrema

Wikimedia Foundation (n.d.). **Daemon (computing)**. Extraído el 24 de mayo de 2017 desde [https://en.wikipedia.org/wiki/Daemon_\(computing\)](https://en.wikipedia.org/wiki/Daemon_(computing))

ANEXO A

Instrumento de investigación



REPÚBLICA BOLIVARIANA DE VENEZUELA
UNIVERSIDAD JOSÉ ANTONIO PÁEZ
FACULTAD DE INGENIERÍA
ESCUELA DE COMPUTACIÓN

ENTREVISTA

Objetivo: Conocer información relevante para el usuario final del sistema, así como también la utilidad que le puede dar un profesional del área de automatización.

1. ¿Considera necesario el seguimiento y control de tareas utilizando instrumentos de fácil acceso y bajo costo? ¿Por qué?

2. ¿Qué variables maneja usted comúnmente en sus sistemas de medición y automatización? Indique su naturaleza e instrumentos de medición o sensores utilizados.

3. ¿Cuál es el presupuesto promedio de que dispone para un proyecto de automatización y monitoreo?

4. Comente brevemente un caso de un sistema de automatización y monitoreo que haya usado en su entorno laboral.

5. ¿Qué conocimientos posee de ingeniería de software? (Por ejemplo desarrollo de software multiplataforma, diseño de base de datos, manejo de servidores, etc)



REPÚBLICA BOLIVARIANA DE VENEZUELA
UNIVERSIDAD JOSÉ ANTONIO PÁEZ
FACULTAD DE INGENIERÍA
ESCUELA DE COMPUTACIÓN

CARTA DE VALIDACIÓN DEL INSTRUMENTO

Quien suscribe, **Ing. Milagros Mejías**, Portador de la cédula de identidad **Nro: 7.225.362**, por medio de la presente hago constar que he leído y evaluado el instrumento de recolección de datos correspondiente al Proyecto: **DESARROLLO DE UN GESTOR PARA EL CONTROL DE TAREAS DE AUTOMATIZACIÓN DE CIRCUITOS ELECTRÓNICOS HACIENDO USO DE UN MICROCONTROLADOR INALÁMBRICO A TRAVÉS DE UN SISTEMA WEB Y MÓVIL**, presentado por el ciudadano **Samuel Murillo**, Portador de la cédula de identidad **25.582.018**, para optar al título Ingeniero de Computación, el cual apruebo en calidad de validador.

En San Diego, a los 11 días del mes de Octubre del año dos mil diecisiete.

Ing. Milagros Mejías C.I. 7.225.362



**REPÚBLICA BOLIVARIANA DE VENEZUELA
UNIVERSIDAD JOSÉ ANTONIO PÁEZ
FACULTAD DE INGENIERÍA
ESCUELA DE COMPUTACIÓN**

CARTA DE VALIDACIÓN DEL INSTRUMENTO

Quien suscribe, **Ing. Jesús Castellano**, Portador de la cédula de identidad **Nro: 18.867.869**, por medio de la presente hago constar que he leído y evaluado el instrumento de recolección de datos correspondiente al Proyecto: **DESARROLLO DE UN GESTOR PARA EL CONTROL DE TAREAS DE AUTOMATIZACIÓN DE CIRCUITOS ELECTRÓNICOS HACIENDO USO DE UN MICROCONTROLADOR INALÁMBRICO A TRAVÉS DE UN SISTEMA WEB Y MÓVIL**, presentado por el ciudadano **Samuel Murillo**, Portador de la cédula de identidad **25.582.018**, para optar al título Ingeniero de Computación, el cual apruebo en calidad de validador.

En San Diego, a los 11 días del mes de Octubre del año dos mil diecisiete.

Ing. Jesús Castellano C.I. 18.867.869




REPÚBLICA BOLIVARIANA DE VENEZUELA
UNIVERSIDAD JOSÉ ANTONIO PÁEZ
FACULTAD DE INGENIERÍA
ESCUELA DE COMPUTACIÓN

CARTA DE VALIDACIÓN DEL INSTRUMENTO

Quien suscribe, **Ing. Karen Ramírez**, Portador de la cédula de identidad **Nro: 14.571.240**, por medio de la presente hago constar que he leído y evaluado el instrumento de recolección de datos correspondiente al Proyecto: **DESARROLLO DE UN GESTOR PARA EL CONTROL DE TAREAS DE AUTOMATIZACIÓN DE CIRCUITOS ELECTRÓNICOS HACIENDO USO DE UN MICROCONTROLADOR INALÁMBRICO A TRAVÉS DE UN SISTEMA WEB Y MÓVIL**, presentado por el ciudadano **Samuel Murillo**, Portador de la cédula de identidad **25.582.018**, para optar al título Ingeniero de Computación, el cual apruebo en calidad de validador.

En San Diego, a los 11 días del mes de Octubre del año dos mil diecisiete.



Ing. Karen Ramírez C.I. 14.571.240