

**UNIVERSIDAD POLITÉCNICA SALESIANA**  
**SEDE QUITO**

**CARRERA:**

**INGENIERÍA ELECTRÓNICA**

**Trabajo de titulación previo a la obtención del título de:**  
**INGENIEROS ELECTRÓNICOS.**

**TEMA:**

**SISTEMA DE CONTROL EN RED INALÁMBRICA (WNCS) IMPLEMENTADO AL  
CONTROL DE VOLTAJE DE UN GENERADOR DC DIDÁCTICO.**

**AUTORES:**

**ADRIÁN FRANCISCO LÓPEZ TERÁN**

**DIEGO PATRICIO ZAPATA NARVÁEZ**

**TUTOR:**

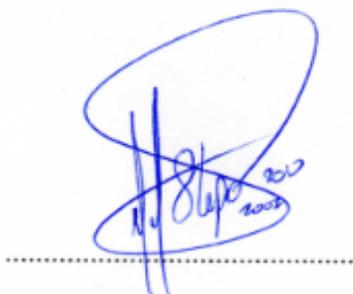
**CARLOS GERMÁN PILLAJO ANGOS**

**Quito, junio del 2018**

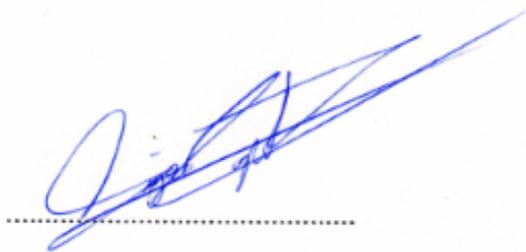
## CESIÓN DE DERECHOS DE AUTOR

Nosotros, Adrián Francisco López Terán con documento de identificación N° 172352501-8 y Diego Patricio Zapata Narváez con documento de identificación N° 172677062-9, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación intitulado: “SISTEMA DE CONTROL EN RED INALÁMBRICA (WNCS) IMPLEMENTADO AL CONTROL DE VOLTAJE DE UN GENERADOR DC DIDÁCTICO.”, mismo que ha sido desarrollado para optar por el título de Ingenieros Electrónicos, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribimos este documento en el momento que hacemos entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.



Adrián Francisco López Terán  
Cédula: 172352501-8



Diego Patricio Zapata Narváez  
Cédula: 172677062-9

Quito, junio 2018

## DECLARATORIA DE COAUTORÍA DEL DOCENTE TUTOR

Yo declaro que bajo mi dirección y asesoría fue desarrollado el trabajo de titulación, “SISTEMA DE CONTROL EN RED INALÁMBRICA (WNCS) IMPLEMENTADO AL CONTROL DE VOLTAJE DE UN GENERADOR DC DIDÁCTICO.”, realizado por Adrián Francisco López Terán y Diego Patricio Zapata Narváez, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana para ser considerado como trabajo final de titulación.

Quito, junio del 2018

A handwritten signature in blue ink, consisting of a large, stylized loop followed by several horizontal strokes, is written over a horizontal dotted line on a light-colored background.

Carlos Germán Pillajo Angos

C.I: 1709255119

## **DEDICATORIA**

Sin duda dedico este proyecto de titulación a la persona que siempre ha estado a mi lado, brindándome su amor, consejos y comprensión a lo largo de mi vida, su apoyo incondicional ha sido el pilar fundamental para enfrentar las adversidades que se han presentado y conseguir avanzar cada peldaño.

En este y en todos los logros que consiga, estrás presente.

Te amo MAMÁ.

**Adrián**

Esto va dedicado para todos aquellos que siempre creyeron en mi para todo aquel que supo levantarme y decirme que una caída no es derrota, que en la vida no hay atajos y sin obstáculos en la vida no es vida, es por esto que este trabajo va para para mis padres, mis hermanos, mi novia, y finalmente para mis compañeros.

**Diego**

## **AGRADECIMIENTO**

A Dios, a la vida, a mis padres, a mis abuelitos, a mi familia, a todos los profesores que a lo largo de mi vida estudiantil, inculcaron en mí no solo conocimientos, sino también valores, a mi novia, a mis amigos y compañeros de universidad con quienes compartí estos 5 años de carrera, agradezco a todas las personas que de una u otra forma estuvieron involucradas en este proceso y aportaron para lograr culminar una etapa más de mi vida.

**Adrián**

Primeramente, debo agradecer a Dios, por todo, por los medios para poder estudiar, por mi familia con la cual he llegado hasta aquí, por mis amigos, por mi novia y por todas las pruebas que me hicieron crecer hasta llegar a este punto por eso mucho más agradezco a Dios y a todos a los que puso en mi vida gracias.

**Diego**

Además, agradecemos a la Universidad Politécnica Salesiana, a nuestro tutor Ing. Carlos Germán Pillajo quien con sus conocimientos, consejos y apoyo supo guiarnos a lo largo de esta etapa, no solo en la parte profesional sino también en la parte humana, nos impulsó y motivo a seguir adelante, a desarrollar y concluir el presente proyecto de la mejor manera.

**Los Autores**

## ÍNDICE DE CONTENIDO

CESIÓN DE DERECHOS DE AUTOR.....	ii
DECLARATORIA DE COAUTORÍA DEL DOCENTE TUTOR.....	iii
DEDICATORIA .....	iv
AGRADECIMIENTO.....	v
RESUMEN.....	vi
ABSTRACT.....	vii
INTRODUCCIÓN .....	vii
CAPÍTULO I.....	1
ANTECEDENTES.....	1
1.1. Planteamiento del problema.....	1
1.2. Tema.....	1
1.3. Justificación.....	1
1.4. Objetivos .....	2
1.4.1. Objetivo general.....	2
1.4.2. Objetivos Específicos.....	2
1.5. Metodología .....	3
CAPÍTULO II .....	4
MARCO TEÓRICO.....	4
2.1. Sistemas Ciber-Físicos .....	4
2.2. Sistemas de control en red Inalámbrica WNCS .....	4
2.3. Internet de las cosas (IOT).....	5
2.4. Control basado en eventos .....	5
2.5. Control de Voltaje dc .....	6
2.5.1. Control por modulación de ancho de pulso - PWM.....	6
2.5.2. Control Proporcional Integral Derivativo - PID.....	6

2.6.	Control predictivo .....	7
2.7.	Herramientas de software.....	7
2.7.1.	Arduino IDE.....	7
2.7.2.	JavaScript .....	7
2.7.3.	Notación de Objetos de JavaScript (JSON) .....	8
2.7.4.	Lenguaje de Marcación de Hipertexto (HTML) .....	8
2.7.5.	Pre Procesador de Hypertexto (PHP).....	8
2.7.6.	Base de Datos MySQL.....	8
2.7.7.	Protocolo de Transferencia de Hipertexto (HTTP).....	9
2.7.8.	Plataforma No-IP .....	9
2.7.9.	WEBcam xp 5 .....	10
2.7.10.	Ident de Matlab .....	10
2.8.	Hardware utilizado .....	10
2.8.1.	Acondicionamiento De Señal.....	10
2.8.2.	Arduino Mega 2560 .....	11
2.8.3.	Módulo Dragino Yun Shield.....	11
2.8.4.	Raspberry Pi .....	12
2.8.5.	Cámara IP.....	12
2.9.	Maquinas Rotativas DC .....	12
2.9.1.	Motor DC .....	12
2.9.2.	Generador DC .....	13
CAPÍTULO III .....		14
IMPLEMENTACIÓN DEL SISTEMA DE CONTROL EN RED INALÁMBRICA .....		14
3.1.	Sistema a implementar .....	14
3.2.	Estructura del sistema .....	14
3.2.1.	Controlador Servidor.....	14
3.2.2.	Esquema De Comunicación Cliente-Servidor .....	15

3.2.3.	Cliente Sensor-Actuador .....	15
3.3.	Metodología de la implementación del generador de voltaje DC didáctico .....	16
3.3.1.	Proceso De Adquisición Y Control De La Señal .....	16
3.3.2.	Curva de comportamiento del sistema motor-generador DC.....	18
3.3.3.	Validación de la curva.....	20
3.4.	Modelo de la planta.....	21
3.4.1.	Modelo del motor DC .....	21
3.4.2.	Modelo del Generador.....	24
CAPÍTULO IV .....		27
CONTROLADOR PREDICTIVO EN WLAN .....		27
4.1.	Obtención del controlador Predictivo .....	27
4.1.1.	Estimador del Servidor-Controlador PID.....	27
4.1.2.	Estimador del Cliente – Planta.....	32
4.2.	Comunicación cliente-servidor .....	34
4.2.1.	Formato JSON para la comunicación, envío-recepción de datos del cliente-servidor.....	34
4.3.	Implementación del control basado en eventos .....	35
4.3.1.	Detector de Eventos .....	35
4.3.2.	Generador de Entrada de Control.....	35
CAPÍTULO V .....		36
CONTROLADOR PREDICTIVO EN WAN.....		36
5.1.	Configuración del Servidor en la nube.....	36
5.1.1.	Obtención de dominio y carga de archivos en el servidor .....	36
5.1.2.	Direccionamiento para la comunicación cliente-servidor.....	36
5.2.	Migración de servidor local a servidor en la nube .....	37
5.3.	Diferencias direccionamiento Servidor Local Vs Servidor en la Nube .....	37
5.4.	Configuración cámara IP.....	38

5.4.1.	Configuración software WEB CAM XP 5.....	38
5.4.2.	Configuración Software NO-IP.....	38
5.4.3.	Configuración de puertos de comunicación para la utilización de la cámara IP.....	38
CAPÍTULO VI.....		39
PRUEBAS Y RESULTADOS.....		39
6.1.	Funcionamiento PID en la planta, PID WLAN y PID WAN .....	39
6.1.1.	Pruebas y Resultados, Funcionamiento PID En Planta.....	39
6.1.2.	Pruebas y Resultados, Funcionamiento PID en Red WLAN.....	40
6.1.3.	Pruebas y Resultados, Funcionamiento PID en red WAN.....	40
6.1.4.	Comparación PID en la Planta, PID WLAN y PID WAN.....	41
6.2.	Comportamiento del controlador con y sin comunicación. ....	43
CONCLUSIONES .....		45
RECOMENDACIONES .....		46
LISTA DE REFERENCIAS .....		47
ANEXOS.....		51
	Elaboración placa de potencia y acondicionamiento de señal .....	51
	Configuración del Servidor en la nube.....	51
	Obtención de dominio y carga de archivos en el servidor .....	53
	Configuración software WEB CAM XP 5.....	56
	Configuración software NO-IP .....	60
	Configuración de puertos de comunicación para la utilización de la cámara IP. ....	66

## ÍNDICE DE FIGURAS

Figura 2.1: Diagrama de interacción Sistemas Ciber-Físicos .....	4
Figura 2.2: Arduino Mega 2560.....	11
Figura 2.3: Módulo Dragino Yun Shield .....	11
Figura 2.4: Raspberry Pi 3 Model B .....	12
Figura 3.1: Dispositivos utilizados en la implementación del WNCS.....	14
Figura 3.2: Estructura Del Sistema .....	15
Figura 3.3: Diagrama de Comunicación Cliente – Servidor .....	15
Figura 3.4: Sistema a Controlar .....	16
Figura 3.5: Diagrama de Control .....	18
Figura 3.6: Muestras Señal Vout VS PWM.....	19
Figura 3.7: Gráficas Señal Vout VS PWM .....	19
Figura 3.8: Gráfica Señal Vout VS PWM $t=0$ .....	19
Figura 3.9: Gráfica Señal Vout VS PWM sin tiempo de estabilización .....	20
Figura 3.10: Gráfica Señal Vout VS PWM, simulación y real. ....	21
Figura 3.11: Gráfica Señal Vout VS PWM, simulación y real. ....	21
Figura 3.12: Esquema Motor DC .....	22
Figura 3.13: Principio de funcionamiento de un generador DC .....	24
Figura 3.14: Circuito Equivalente de un generador CC.....	25
Figura 4.1: Ventana IDENT.....	27
Figura 4.2: Ventana Import Data .....	28
Figura 4.3: Número de polos y ceros del sistema .....	28
Figura 4.4: Estimación de función de transferencia .....	29
Figura 4.5: Modelamiento del sistema primera muestra .....	29
Figura 4.6: Modelamiento del sistema segunda muestra .....	29
Figura 4.7: Resultados presentados por Matlab .....	30
Figura 4.8: Algoritmo implementado en el Servidor .....	31
Figura 4.9: Algoritmo implementado en la Planta .....	33
Figura 4.10: Comunicación Cliente - Servidor .....	34
Figura 5.1: Direccionamiento Servidor Local (WLAN) .....	37
Figura 5.2: Direccionamiento Servidor en la Nube (WAN) .....	37
Figura 6.1: Voltaje OUT vs Tiempo (en Planta).....	39

Figura 6.2: Voltaje OUT vs Tiempo (en red WLAN).....	40
Figura 6.3: Voltaje OUT vs Tiempo (en red WAN) .....	41
Figura 6.4: Comparación Voltaje OUT vs Tiempo (bajada).....	41
Figura 6.5: Comparación Voltaje OUT vs Tiempo (subida).....	42
Figura 6.6: Comparación Voltaje OUT vs Tiempo (subida).....	43

## RESUMEN

Al hablar sobre Wireless Network Control Systems, se abordan temas de teoría de control combinados con teoría de comunicación fusionados en un sistema ciber-físico en el que interactúan sensores, actuadores y controladores a través de una red inalámbrica de comunicación. Esta arquitectura exhibe algunas ventajas entre las que destacan reducción de cableado y flexibilidad, por otro lado el medio por el que los datos son transmitidos no es ideal y se presentan limitaciones al compartir ancho de banda con otros dispositivos conectados a la red; en consecuencia existen retrasos en la transmisión (latencias).

Debido a la carencia de aplicaciones prácticas en esta área, en el presente trabajo se desarrolló un sistema de control para un generador de voltaje DC didáctico mediante control de velocidad PWM, en el cual el controlador se encuentra en un servidor WEB albergado en una tarjeta Raspberry Pi 3 en la que entrelaza programación HTML, PHP y JSCRIPT de forma armoniosa para la administración del flujo de datos entre cliente y servidor.

En la planta tanto el sensor como el actuador están conectados a una tarjeta Arduino Uno y un módulo Dragino, para permitir que la comunicación sea inalámbrica, cerrando el lazo de control a través de la red o de la nube. Se va a implementar un controlador PID basado en eventos para garantizar la regulación y estabilidad de la planta; demostrando su utilidad y las limitaciones del control inalámbrico (WNCS) en este tipo de plantas.

## **ABSTRACT**

When discussing about Wireless Network Control Systems, control and communication theory issues are combined and fused in a Cyberphysical System in which sensors, actuators and controllers interact through a wireless communication network. This architecture exhibits some advantages, flexibility and reduction of cabling are the most important, however the data are transmitted on not ideal media and shows some limitations when sharing bandwidth with other devices connected to the network; as consequence delays in transmission are presented (latencies).

Due to the shortage of practical applications in this area, In the present work a didactic DC voltage control system was developed, the controller is located on a WEB server housed in a Raspberry Pi 3 board, HTML, PHP and JAVASCRIPT programming, have an harmoniously interact for the administration of the flow of data between client and server.

At the plant, the sensor and the actuator are connected to an Arduino UNO board and a Dragino module who allow wireless communication, closing the control loop through the network or the cloud. To guarantee the stability of the plant; an implementation of an event-based PID controller is going to be implemented; demonstrating the utility and limits of WNCS in this type plants.

## INTRODUCCIÓN

La necesidad del hombre por la libertad, se ha trasladado a la cotidianidad de manejar dispositivos de forma inalámbrica, a medida del gran avance tecnológico y desarrollo del mundo, seguramente, en pocos años se verá implantado en varios procesos industriales. Los WNCS se encuentran en la intersección tanto de la teoría de control (sistemas dinámicos interconectados a través de canales de comunicación ideales) como de la teoría de la comunicación (transmisión de datos por canales imperfectos).

El presente trabajo trata de la implementación de un controlador predictivo PID basado en eventos, aplicado a un Sistema de Control de Voltaje DC en un Generador Didáctico. Estructurado de 6 capítulos que justifican los objetivos planteados.

El capítulo 1 aborda la metodología con la cual se realizó el presente proyecto. En el capítulo 2, se expone la parte conceptual, necesaria para comprender fundamentos teóricos del tema y punto de partida para desarrollar la simulación e implementación del Sistema de Control en Red Inalámbrica (WNCS) para el control de voltaje DC en un generador didáctico.

En el capítulo 3 se utiliza la herramienta IDENT del software Matlab para la obtención del modelo matemático, se compara la respuesta que se obtiene de la función de transferencia con la respuesta real del sistema, además se muestra el diagrama de bloques del sistema implementado. En el capítulo 4 y 5 se detalla la implementación del Sistema de Control para un Generador de Voltaje DC Didáctico mediante control de velocidad PWM, describiendo tanto diseño de hardware como funcionamiento de software. El capítulo 6 muestra los resultados obtenidos del sistema implementado, con gráficas se comparan los resultados de simulación e implementación del proyecto, el comportamiento del sistema al estar el servidor en una red local y al acceder al servidor desde un servicio de hosting. Conjuntamente se detalla conclusiones y recomendaciones que servirán para futuras investigaciones.

# CAPÍTULO I

## ANTECEDENTES

### 1.1. Planteamiento del problema

Los procesos industriales necesitan alta fiabilidad en su control, un factor muy importante es la velocidad de transmisión de señales para constatar en qué estado se encuentran las variables a manipular y aplicar un determinado control para que su comportamiento sea el adecuado. Las comunicaciones cableadas por la seguridad, fiabilidad y el bajo tiempo de retardo que ofrece son las idóneas para este tipo de procesos, pero la evolución del hombre lleva a buscar mecanismos que permitan realizar los mismos procesos con la libertad que ofrecen las comunicaciones vía Wireless. (de la Fuente López & Mazaeda Echevarría, 2016)

En los sistemas de control industrial en la actualidad comúnmente se requieren una gran distribución espacial de sensores y actuadores para monitorear y controlar el correcto funcionamiento de procesos físicos, pero a pesar de los avances tecnológicos en sensores y actuadores la industria recurre a infraestructura de comunicación cableada. Además se requiere controladores de gran capacidad para cada sistema o planta lo que eleva el costo de implementación de las mismas. (Monsalve Posada & Arias Londoño, 2015)

El problema al que se suele enfrentar sin duda son los retardos que se pueden presentar en la comunicación inalámbrica, lo cual será solventado mediante algoritmos predictivos que ayudaran a intuir el comportamiento futuro del proceso y mantener la estabilidad del sistema. (Monsalve Posada & Arias Londoño, 2015)

### 1.2. Tema

SISTEMA DE CONTROL EN RED INALÁMBRICA (WNCS) IMPLEMENTADO AL CONTROL DE VOLTAJE DE UN GENERADOR DC DIDÁCTICO.

### 1.3. Justificación

El desafío de implementar tecnologías de control industrial de manera inalámbrica está tomando forma gracias a la introducción a los WNCS (*Wireless Network Control System*). Una de las principales ventajas que presentan los WNCS es la reducción de

cableado lo que conlleva menores tiempos de instalación y la migración del control de hardware a software. (Marcosig, Felicioni, & Zanini, 2014)

El desarrollo de servicios en la nube permitirá que el control no esté ligado a un proyecto en específico, sino que un usuario capacitado podrá enlazar diferentes procesos; uno a la vez; al servidor y de esta forma se contribuirá con el desarrollo de nuevas aplicaciones físicas; además de la que se implementara en el presente trabajo; en el campo del WNCS.

La comunicación entre dispositivos inalámbricos requiere de cierto tiempo llamado “latencia”, en sistemas industriales de control esta comunicación debe garantizar fiabilidad por lo que es necesario un algoritmo de control robusto que no presente dificultades por las deficiencias que puedan presentarse en la red. (Monsalve Posada & Arias Londoño, 2015)

#### **1.4. Objetivos**

##### **1.4.1. Objetivo general**

Implementar un Sistema de control en red inalámbrica (WNCS) aplicado al control de voltaje de un generador DC didáctico, mediante algoritmos predictivos basado en eventos, monitoreado a través de la nube.

##### **1.4.2. Objetivos Específicos**

- Analizar la base teórica para la implementación de este tipo de controladores en redes inalámbricas.
- Implementar un sistema que realice el control de voltaje en un generador DC didáctico para la medición y obtención de curva de comportamiento del generador.
- Implementar un controlador basado en eventos para un WNCS mediante un sistema embebido en el cual estará el algoritmo predictivo para el control y monitoreo del sensor-actuador de la planta.
- Gestionar los datos obtenidos de la planta para su correcta interpretación y ejecución del controlador a través de la nube.
- Comparar el comportamiento del controlador PID en la nube en contraste con un PID local destinado al caso de estudio para determinar ventajas y desventajas del sistema.

## **1.5. Metodología**

- Se tomarán artículos científicos referenciales, publicados en repositorios certificados, para analizar el comportamiento de sistemas que integren motores y control por WNCS y así determinar los recursos necesarios para su implementación.
- Para realizar la planta a controlar se utilizarán moto-reductores DC acoplados al eje, de tal manera que el motor que será alimentado con una Fuente DC de 12 voltios, al girar permita que el motor utilizado como generador entregue un voltaje de salida, el cual deberá ser medido a través de acondicionamiento electrónico y entradas análogas, dicho voltaje de salida será controlado a voluntad del usuario.
- Para efecto de simulaciones se determinará el modelo matemático del sistema por lo cual se utilizará la herramienta IDENT de MATLAB.
- Mediante el modelo matemático experimental que se obtendrá, se determinará el Algoritmo Predictivo propio del sistema que permitirá evitar errores por latencias presentes en la red inalámbrica.
- Se desarrollará un servidor en el cual estará establecido tanto el controlador como el Algoritmo Predictivo y mostrará al usuario el estado y comportamiento actual e histórico del Sistema.

## CAPÍTULO II

### MARCO TEÓRICO

#### 2.1. Sistemas Ciber-Físicos

Sistemas Ciber-Físicos (CPS) es la asociación del mundo físico y cibernético, que busca mejorar la calidad de vida a través del monitoreo y control, del mundo físico en el que vivimos utilizando como herramienta las capacidades del mundo cibernético enlazando teorías de control, comunicación y computación. (Pillajo Angos C. , 2014)

Figura 2.1: Diagrama de interacción Sistemas Ciber-Físicos

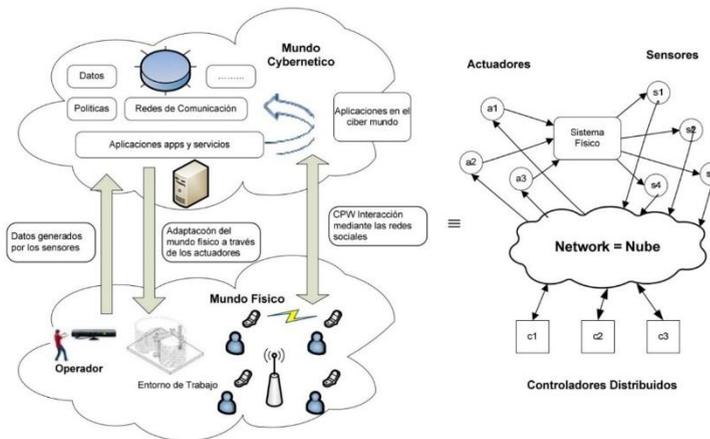


Diagrama gráfico interacción de un Sistema Ciber-Físico. (Pillajo Angos C. , 2014)

Se llama Sistema Ciber-Físico a aquel que reúne procesos físicos, hardware, software y se encuentra conectado mediante una infraestructura de telecomunicaciones, este campo se encuentra estrechamente relacionado con la denominada industria 4.0, tanto la comunidad científica como la industrial concuerdan en que este tipo de sistemas y específicamente hablando el de la producción pueden ocasionar la cuarta revolución industrial. Un sistema Ciber-Físico requiere la integración de ingeniería electrónica, ingeniería de software e ingeniería de telecomunicaciones. (Sobrevilla, Hernández, Velasco-Elizondo, & Soriano, 2017) (Beruvides López, Haber Guerra, & Castaño Romero, 2016)

#### 2.2. Sistemas de control en red Inalámbrica WNCS

Los sistemas de control en red NCS de sus siglas en inglés, Network Control Systems, han aparecido gracias al desarrollo de las tecnologías de computación,

comunicación y control. Su lazo de control se encuentra cerrado a través de un canal de comunicación, a diferencia de los controles convencionales el canal de comunicación utilizado no es ideal. Cuando un NCS se aplica a través de redes inalámbricas se denomina Wireless Network Control System por sus siglas en inglés WNCS. Entre sus ventajas se encuentra flexibilidad, movilidad, reducción y ahorro de cableado, lo que implica menor tiempo de instalación. Los desafíos a los que se enfrenta un WNCS al comunicarse por un medio no ideal, son sin duda fiabilidad y retardos de comunicación. (Zhang, Wang, & Yu, 2017) (Hernandez Bonilla & Vargas Plazas, 2013)

### **2.3. Internet de las cosas (IOT)**

El internet de las cosas denomina como “cosas” a cualquier dispositivo que se encuentre conectado a internet y que ofrezca información, como ejemplo se puede tomar las Smart cities como Barcelona-España en la cual en las paradas de bus se tiene información del tiempo de espera entre cada unidad. El internet de las cosas también está migrando hacia la industria IIoT (de sus siglas en inglés Industrial Internet of Things), como se comenta en el libro “El crecimiento no convencional a través de la Internet Industrial de las Cosas en la Industria”, las empresas utilizan la IIoT para mejorar servicios de mantenimiento que ya ofrecen, ciertas empresas están innovando y utilizando la IIoT para actualizar y ofrecer nuevos servicios, mejorar productos e ingresar a nuevos mercados como Michelin que utiliza sensores en los neumáticos y mediante análisis, ayudan a ahorrar combustible a los conductores de las flotas de camiones. (Daugherty, Banerjee, Negm, & Alter, 2017) (Callicó & Celma, 2016)

### **2.4. Control basado en eventos**

Su origen data de los años 50, Phillip H. Ellis planteo que el método más apropiado de muestreo se debe realizar únicamente cuando exista un cambio significativo en la señal, así la muestra estará justificada. Los sistemas de control en red demandan un considerable intercambio de información, es así que el control basado en eventos es una solución para reducir la carga de tráfico y hacer más flexible el control de procesos. (Guinaldo, Sánchez, & Dormido, 2017) (Velásquez, Pinto, & Rosales, 2017)

## **2.5. Control de Voltaje dc**

### **2.5.1. Control por modulación de ancho de pulso - PWM**

El control PWM, es el método más utilizado para controlar las revoluciones de un motor DC, su funcionamiento se basa en el control del intervalo de tiempo en que la señal se encuentra en 1 lógico, intervalo de tiempo conocido como  $T_{ON}$ , siendo  $T_{OFF}$  el intervalo en el que la señal se encuentra en 0 lógico. (Petru & Mazen, 2015)

En el presente proyecto, se utiliza el control PWM en el motor, al controlar su tiempo de activación, se controlará sus revoluciones y al estar acoplado por el eje al generador, se controlará el voltaje de salida ( $V_{out}$ ), teniendo una relación directamente proporcional: a mayores revoluciones del motor, mayor diferencial de voltaje en los terminales.

### **2.5.2. Control Proporcional Integral Derivativo - PID**

El control PID (Proporcional, Integral, Derivativo), se utiliza en varias aplicaciones en la industria, este tipo de control es retroalimentado, su función es reducir a cero el error que existe entre la señal de salida de la planta y la señal referencial o Setpoint (SP) (GIL GARCES & RINCÓN GAVIRIA , 2014). A continuación se detallan sus acciones:

- Acción Proporcional.- Como su nombre lo indica, esta acción de control será proporcional al error, normalmente en un controlador proporcional existe un error en estado estacionario.
- Acción Integral.- Se encarga de asegurar que la salida del sistema, concuerde con el valor deseado (SP). Al existir un “error +” la señal de control incrementara, de igual manera, un “error -” generará una señal menor, sin tomar en cuenta el valor del error.
- Acción Derivativa.- Su propósito es mejorar la estabilidad del lazo cerrado, produciendo en la salida del proceso un efecto predictivo.

En la planta utilizada para el sistema implementado en el presente trabajo, la señal de referencia (SP), es el valor deseado en el generador ( $V_{out}$ ), la red de comunicación inalámbrica será quien cierre el lazo de control.

## **2.6. Control predictivo**

Estrategia de control, con un extenso conjunto de métodos, cuyo desarrollo parte de ideas comunes. En el artículo “Control Predictivo Generalizado en un PLC Siemens Simatic S7-1200” (Neila, Roshany-Yamchi, Gnyszka, & Limón, 2015), se detallan los aspectos a tomar en cuenta para desarrollar un control predictivo, los cuales son:

- Para predecir la salida en un intervalo de tiempo futuro, se utiliza un modelo.
- Para minimizar la función objetivo, desarrollar cálculos óptimos del conjunto de señales de control.
- Uso de una estrategia deslizante.

En el control de motores DC este tipo de control ayudará a proporcionar una regulación amplia y progresiva de la velocidad. (López & Sierra, 2015)

## **2.7. Herramientas de software**

En la presente sección se presenta el conocimiento teórico básico de los elementos, programas y lenguajes de programación utilizados para el presente proyecto

### **2.7.1. Arduino IDE**

Entorno de desarrollo integrado (de sus siglas en inglés Integrated Development Environment), es la plataforma de programación de Arduino, la cual es adecuada tanto para principiantes o expertos, esta plataforma es de software libre, nació en Italia en el instituto de diseño interactivo Ivrea en el año 2005, con la ayuda de esta plataforma es posible programar las placas Arduino las cuales también son hardware libre. (Moran, Teragni, & Zabala, 2017) (Reimann & Maday , 2017)

### **2.7.2. JavaScript**

Desarrollado por Netscape Communications, se trata de un lenguaje de programación de scripts, derivado de C++. Su objetivo es dar vida a los navegadores desde el lado del cliente. En la actualidad Netscape pertenece a la fundación Mozilla, quienes han incrementado la popularidad de este lenguaje, ya que permite transformar paginas HTML estáticas en dinámicas sin tener que recurrir al servidor (Castillo, 2017) (Vigroux, 2015). Dentro de la página WEB implementada en el presente proyecto, JavaScript tiene la función de visualizar y presentar los datos monitoreados, a su vez informará en caso de existir pérdida de comunicación.

### **2.7.3. Notación de Objetos de JavaScript (JSON)**

Especificado en el año 2001 por Douglas Crockford, se trata de un formato eficaz para codificar datos, comprensible para humanos y fácil interpretación para computadoras, una de sus principales virtudes es el intercambio rápido de pequeñas cantidades de datos entre navegadores y servicios WEB (Mora Castillo, 2016). En el presente trabajo los datos tanto del sensor como del controlador son enviados en formato JSON.

### **2.7.4. Lenguaje de Marcación de Hipertexto (HTML)**

Fue creado por Tim Berners-Lee en 1989, convirtiéndose en 1995 en un estándar para compartir documentos, su función es la gestión y organización del contenido que se desea mostrar en una página WEB, como imágenes, texto, entre otros (Romero Alamán, 2017) (Contreras Rodríguez, 2015). En conjunto JavaScript y HTML, son la apariencia y presentación de la interfaz de la página WEB del presente proyecto.

### **2.7.5. Pre Procesador de Hipertexto (PHP)**

Diseñado por Rasmus Lerdorf en el año 1994, originalmente para aplicaciones WEB del lado del cliente, puede ser insertado en documentos HTML, entre sus principales características se tienen: robustez, velocidad, orientado a objetos, software gratuito y de código abierto. Al ser un lenguaje modular, es ideal para el uso e instalación en servidores WEB. Según analistas, se cree que más del 80% de sitios WEB utilizan PHP. (Arias, Aprende Programacion WEB con PHP y MySQL, 2015) (Heurtel, 2015)

Anteriormente se indicó que HTML y JavaScript son la interfaz de presentación de la página WEB, PHP es el servidor tras la página, encargado de realizar cálculos, operaciones del controlador y responder solicitudes del cliente (Sistema a Controlar).

### **2.7.6. Base de Datos MySQL**

Es un Sistema de Gestión de Base de Datos Relacional (SGBDR) capaz de conectarse con PHP, es la base de datos más utilizada en internet debido a que en aplicaciones WEB, la modificación de datos no es frecuente, pero a su vez la búsqueda y lectura de datos es muy habitual (Arias, Aprende Programacion Web con

PHP y MySQL, 2017) (Natsys, 2014). MySQL almacena usuarios y contraseñas para el uso del servidor.

### **2.7.7. Protocolo de Transferencia de Hipertexto (HTTP)**

El Protocolo de Transferencia de Hipertexto de sus siglas en inglés HTTP, es el tratado usado por los programas para la comunicarse con World Wide WEB (WWW) o internet. HTTP es muy utilizado por dos características, conversación entre navegadores y servidores WEB (Gourley, Totty, Sayer, Aggarwal, & Reddy, 2002). A continuación se presentan los métodos usados por HTTP:

- Get.- Usado para solicitar un servicio o para enviar un recurso. HTTP/1.1 requiere servicios para implementar estos métodos. EL método GET es usado por el cliente para solicitar la respuesta del controlador.
- Post.- Diseñado para enviar entrada de datos para un servidor. Es usado frecuentemente para soportar formularios HTML. Los datos llenados en formularios HTML son típicamente enviados al servidor. El método POST permite al usuario cambiar parámetros de control como SP, kp, ki y kd.
- Put.- Escribe documentos para un servidor. Algunos sistemas de publicación que permiten crear páginas WEB usan PUT. El método PUT es usado por el cliente para enviar los datos de la planta en este caso el voltaje medido.

### **2.7.8. Plataforma No-IP**

Se trata de una plataforma WEB que proporciona servicio DNS dinámico, con el cual se asigna un nombre de Host a una dirección IP dinámica, entendiéndose por dinámica a una dirección que cambiará con cierta frecuencia, NO-IP se encarga de actualizar y relacionar dicha dirección con el nombre de Host seleccionado por el usuario.

Este servicio se utiliza en el presente proyecto para el monitoreo de la planta al estar conectada al controlador WAN, relacionando la Dirección IP “Pública” de la red, a un nombre de dominio dinámico y agregando un número de puerto se puede acceder únicamente a la cámara WEB y mostrar en “tiempo real” el estado de la planta.

### **2.7.9. WEBcam xp 5**

Es un software que permite convertir una cámara WEB en una cámara IP, se utilizó para los controladores WLAN y WAN, en el caso del controlador WLAN, la cámara WEB toma la dirección IP del computador al que se encuentre conectado dentro de la misma red, accediendo al video de la cámara por un puerto específico establecido por el usuario, de la misma manera debe configurar en el controlador WAN, su dirección será la IP Pública de la red o el nombre de Host establecido en el servicio NO-IP.

### **2.7.10. Ident de Matlab**

Se trata de una herramienta que permite la obtención de modelos matemáticos a partir de la toma de datos de entrada y de salida, estos datos pueden ser tomados en el dominio del tiempo o en el dominio de la frecuencia y permiten identificar funciones de transferencia en tiempo continuo o en tiempo discreto. El modelo identificado ayuda en la predicción de la respuesta del sistema. (The MathWorks, Inc., 1994-2018)

## **2.8. Hardware utilizado**

En esta sección se detalla la teoría acerca de herramientas y componentes utilizados en la implementación y construcción del hardware utilizado en el presente trabajo.

### **2.8.1. Acondicionamiento De Señal**

- **Conversión ADC.-** La misión de estos convertidores es obtener una representación digital de la magnitud analógica presentada en su entrada través de procesos de cuantificación, la señal analógica de entrada es codificada en un conjunto discreto de estados asignándolos a un conjunto de bits. (Pérez, 2007)
- **Circuitos Operacionales.-** Es un dispositivo electrónico de una salida y dos entradas, en la salida se obtiene la diferencia de las entradas multiplicada por un factor de ganancia (G), su nombre nace por su utilización en calculadoras análogas para desarrollar operaciones matemáticas. Este dispositivo posee diversas configuraciones, una de ellas es seguidor de voltaje, proporciona a la salida la misma tensión que a la entrada, esta configuración está recomendada a la hora de tomar mediciones de voltaje lo más exactas posibles, otra de sus

ventajas es que actúa como un buffer, eliminando efectos de carga y acoplando impedancias. (Juárez Sanchez, Morales Sánchez, & Sandoval Smith, 2017)

### 2.8.2. Arduino Mega 2560

Es una plataforma de desarrollo de hardware libre, posee un microcontrolador ATmega2560 y varios pines que se los puede utilizar como entradas o salidas tanto digitales como análogas (Campos Verdi, 2016), algunas de sus características se encuentran en la siguiente cita. (Campos Verdi, 2016)

Figura 2.2: Arduino Mega 2560



Tarjeta Arduino Mega 2560 Utilizada en la implementación del proyecto, Adrián López & Diego Zapata

### 2.8.3. Módulo Dragino Yun Shield

El Arduino Yún combina la potencia de Linux y la sencillez característica de Arduino. Una de las características más interesantes es que dispone de red ethernet y Wifi, con la posibilidad de configurarse como cliente o punto de acceso. (Reina & Vélez, 2015)

Figura 2.3: Módulo Dragino Yun Shield



Módulo Dragino Yun Shield utilizado en la implementación del proyecto, Adrián López & Diego Zapata

#### **2.8.4. Raspberry Pi**

Se trata de un computador embebido, que cabe en la palma de la mano, su desarrollo comienza en Reino Unido en el año 2011 por la fundación Raspberry PI, el Raspberry Pi 3 Model B fue lanzado al mercado en febrero del 2016 (RASPBERRY PI FOUNDATION) sus características se pueden encontrar en la siguiente cita. (Raspberry, 2016)

Figura 2.4: Raspberry Pi 3 Model B



Tarjeta Raspberry Pi 3 Model B utilizada en la implementación del proyecto, Adrián López & Diego Zapata

La tarjeta Raspberry PI 3 Model B, está configurada como servidor local, para realizar pruebas en red WLAN, estas pruebas son de gran ayuda antes de alojar el servidor en un servicio de Hosting.

#### **2.8.5. Cámara IP**

Las cámaras IP permiten acceder mediante aplicaciones propias del desarrollador o mediante un navegador WEB al archivo de video en tiempo real, en este caso utilizamos una cámara WEB del fabricante Genius, adaptada a cámara Ip gracias al Software WEBCAM XP 5, mencionado anteriormente, esta cámara servirá para monitoreo del estado de la planta.

### **2.9. Maquinas Rotativas DC**

#### **2.9.1. Motor DC**

Los motores DC son máquinas rotativas eléctricas que convierten energía eléctrica DC en energía mecánica. El motor DC consta de dos partes principales, el rotor y el estator, al aplicar corriente dc en sus terminales, las bobinas en el rotor, son magnetizadas y al interactuar el campo magnético inducido por la corriente con un

campo magnético externo (imanes permanentes en el estator), se produce una fuerza perpendicular al plano que forman el campo magnético y el flujo de corriente, llamada fuerza de Lorentz, lo que provoca el movimiento del rotor. (Chapman, 2012)

### **2.9.2. Generador DC**

Los generadores de cd son máquinas que se utilizan como generador, la única diferencia real entre un generador DC y un motor DC es la dirección del flujo de potencia. (Chapman, 2012)

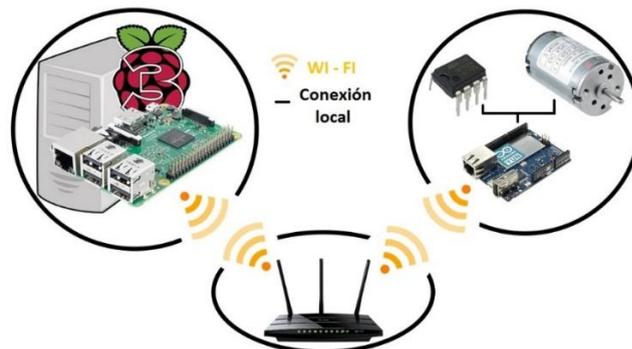
## CAPÍTULO III

### IMPLEMENTACIÓN DEL SISTEMA DE CONTROL EN RED INALÁMBRICA

#### 3.1. Sistema a implementar

El sistema de control de voltaje presentado se comunica dentro de una red WAN. El cliente (sensor-actuador) está localizado en la planta, en este nodo se usa ArduinoUNO + Dragino. En el servidor se encuentra el controlador, implementado en un sistema embebido como es el Raspberry Pi 3. En la Figura 3.1 se muestran los dispositivos utilizados en la implementación del Sistema De Control En Red Inalámbrica.

Figura 3.1: Dispositivos utilizados en la implementación del WNCS



Dispositivos utilizados en la implementación del WNCS, Adrián López & Diego Zapata

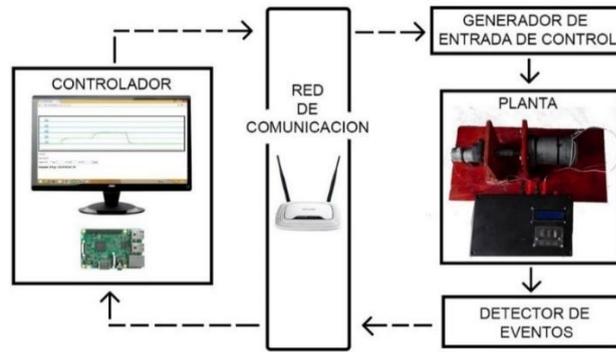
#### 3.2. Estructura del sistema

En esta sección se detalla el sistema compuesto por el Controlador – Servidor, Cliente – Actuador y el esquema de comunicación entre ambos.

##### 3.2.1. Controlador Servidor

El controlador que en este caso es un PID (servidor WEB: recepción y presentación de datos, servidor PHP: procesa y almacena los datos de la planta, servidor SQL: base de datos) está albergado en una tarjeta Raspberry PI 3, a este servidor se puede acceder desde cualquier computador conectado en la misma red de comunicación WAN que cuente con los permisos necesarios. En la Figura 3.2 se detalla la estructura del sistema a implementar.

Figura 3.2: Estructura Del Sistema

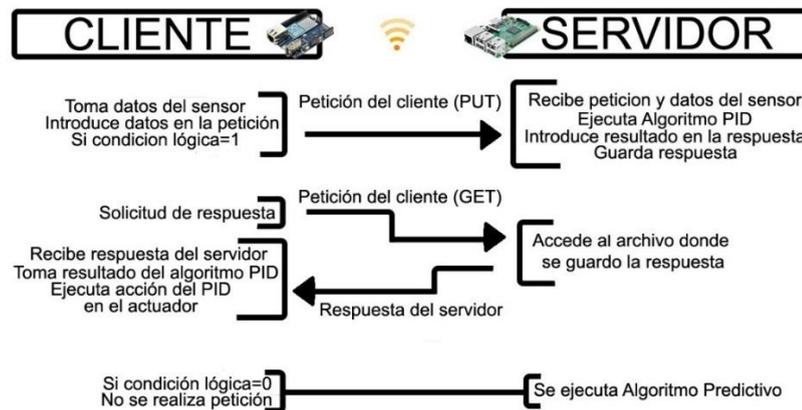


Sistema motor-generator acoplado por ejes a controlar, Adrián López & Diego Zapata

### 3.2.2. Esquema De Comunicación Cliente-Servidor

A continuación, en la Figura 3.3, se muestra el diagrama de cómo se realiza la comunicación entre el Cliente y el Servidor.

Figura 3.3: Diagrama de Comunicación Cliente – Servidor



Sistema motor-generator acoplado por ejes a controlar, Adrián López & Diego Zapata

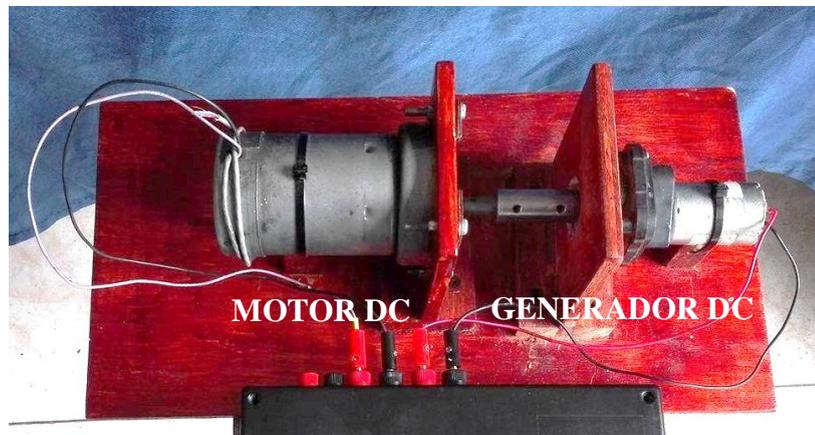
### 3.2.3. Cliente Sensor-Actuador

El valor medido en la planta pasa por un proceso de acondicionamiento de señal y es enviado al controlador y comparado con el SP (SP) especificado; el error presente en la planta es procesado en un algoritmo PID discreto en el servidor el cual retorna un valor comprendido entre 0 - 255 (255→0%; 0→100% del voltaje entregado al motor), este valor activará una salida PWM en el Dragino el cual a su vez provocará la conmutación en un Mosfet encargado de controlar al motor.

En caso de que el sistema llegue a estabilizarse o que la comunicación llegue a fallar, el cliente (Placa Arduino) tiene conFigurado un controlador estimador el cual se

encargará de realizar el control en lazo abierto en lugar del controlador PID del servidor.

Figura 3.4: Sistema a Controlar



Sistema motor-generator acoplado por ejes a controlar, Adrián López & Diego Zapata

En la Figura 3.4, se aprecia el sistema Motor – Generador acoplado por ejes a controlar, el cual consta de un motor DC que será alimentado por una fuente de 12V, este motor transforma la energía eléctrica en energía mecánica, al estar acoplado al eje de otro motor, este realizara el proceso contrario transformando así la energía mecánica, movimiento del rotor, en energía eléctrica.

Debido a defectos de la máquina y sus constantes de fabricación el sistema presentado es no lineal e inestable por lo que se encontraran perturbaciones en el orden de los milivoltios.

### **3.3. Metodología de la implementación del generador de voltaje DC didáctico**

En esta sección se detalla los pasos a seguir para la obtención del modelo matemático y función de transferencia del generador DC, necesarios para efectos de simulaciones y parametrización de valores para el control del sistema.

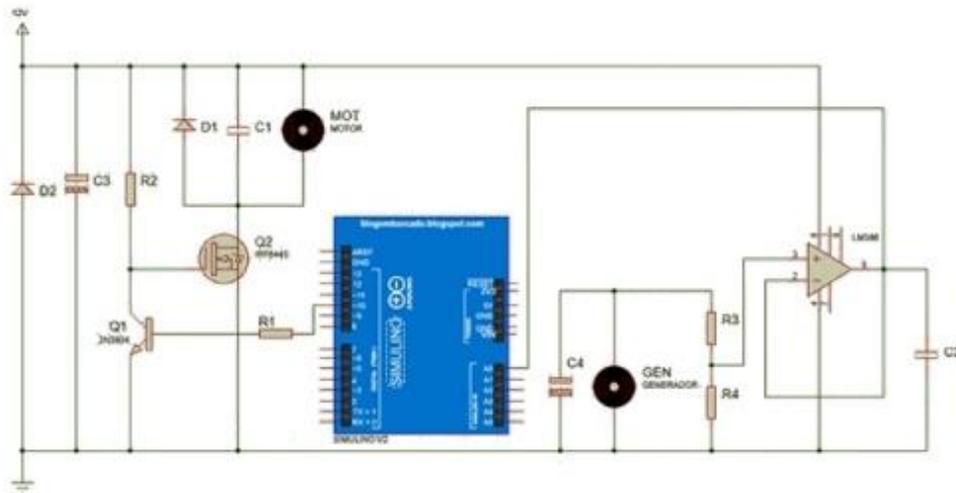
#### **3.3.1. Proceso De Adquisición Y Control De La Señal**

- Identificación de Variables de Entrada-Salida.- Dentro del sistema presentado, la variable a controlar, directamente se trata del voltaje entregado en los terminales del Generador DC, de aquí en adelante se referirá a esta variable como  $V_{out}$ ; indirectamente se controlará la velocidad del Generador DC, ya que la variable manipulada es el voltaje de entrada suministrado al

Motor DC, mediante una señal PWM controlada por Arduino, de aquí en adelante se referirá a esta variable como PWM.

- Circuito de Control de Velocidad.- El control de velocidad para el sistema se lo efectúa con la ayuda de una salida PWM de Arduino, esta salida no tiene la capacidad para alimentar el Motor DC por lo que fue necesario utilizar un transistor NPN el cual convierte la señal PWM de 5V de Arduino en una señal PWM de 12V, la misma que ayuda a controlar el “switched” del MOSFET encargado de entregar el voltaje al Motor DC.
- Circuito de Medición de Voltaje.- Como se explicó anteriormente la variable a controlar es  $V_{out}$ , esta variable se encuentra en el rango de 10V a 30V, la placa Arduino no es capaz de recibir estos valores de voltaje, por lo que es necesario acondicionar esta señal, para lograrlo se utilizó como primer punto un condensador con el objetivo de filtrar y reducir rizado de la señal, posterior a este paso se dimensionó un par de resistencias para implementar un divisor de voltaje, el cual permite que el voltaje máximo sea 5V (voltaje máximo para una entrada análoga de Arduino), finalmente se utiliza un Amplificador Operacional en configuración seguidor de voltaje, el cual permite adaptar impedancias e impedir que se vea afectada la medición por parte del ADC de Arduino.
- Esquema del Circuito de Control Implementado.- La Figura 3.5 muestra el diagrama completo del circuito de control (Circuito Control de Velocidad + Circuito medición de Voltaje), donde además se integraron los diodos rectificadores D1 y D2, los cuales se encargan de controlar y disipar corrientes libres por efecto de las bobinas tanto del Motor DC como del Generador DC, de igual manera los capacitores C3 y C1 se encargaran de los picos y armónicos introducidos en la red de alimentación por las inductancias del circuito.

Figura 3.5: Diagrama de Control



Simulación del diagrama de control (Circuito Control de Velocidad + Circuito medición de Voltaje), Adrián López & Diego Zapata.

### 3.3.2. Curva de comportamiento del sistema motor-generator DC

Para la obtención del modelo matemático se procedió a tomar muestras de las señales de entrada y salida de la planta con un tiempo de muestreo de 10 ms y un retardo en el comportamiento de la planta de 450 ms, con un voltaje mínimo en el generador de 10v, voltaje necesario para que el Motor DC logre romper sus condiciones iniciales de inercia y consiga mover el eje acoplado para producir voltaje en los terminales del Generador DC, como voltaje máximo se estableció 30v, ya que al energizar al motor con el voltaje máximo, el generador en sus terminales registraba un valor de 33 voltios aproximadamente.

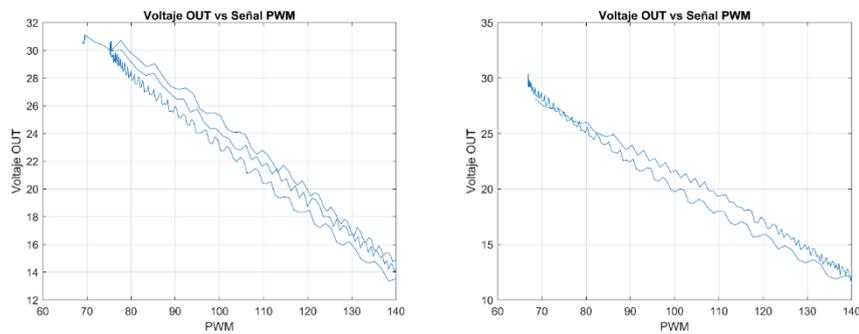
Los datos han sido obtenidos con ayuda del monitor de Arduino IDE, estos datos se pueden observar en la Figura 3.6, en donde se muestran 4 columnas en las cuales, la primera columna muestra la señal PWM, la segunda columna muestra el voltaje de salida en el Generador DC, la tercera columna muestra el error entre el Set Point y el voltaje de salida, la cuarta columna muestra el tiempo que se tarda la placa Arduino Mega 2560 en procesar los datos y ejecutar el algoritmo de control, posteriormente se los almacena en una tabla de Excel donde se comprobó la validez de los datos mediante gráficas de la Entrada vs Salida como se muestra en la Figura 3.7.

Figura 3.6: Muestras Señal Vout VS PWM

PWM	Vout	PWM	Vout
100.47	9.82	10.18	98.00
95.08	10.56	9.44	95.00
92.25	11.09	8.91	96.00
90.67	11.82	8.18	90.00
86.84	12.04	7.96	94.00
83.96	12.62	7.38	90.00
81.09	12.93	7.07	91.00
79.81	13.57	6.43	92.00
75.82	13.57	6.43	94.00
72.20	13.94	6.06	93.00

Toma de muestras señal de entrada PWM y señal de salida Vout, Adrián López & Diego Zapata.

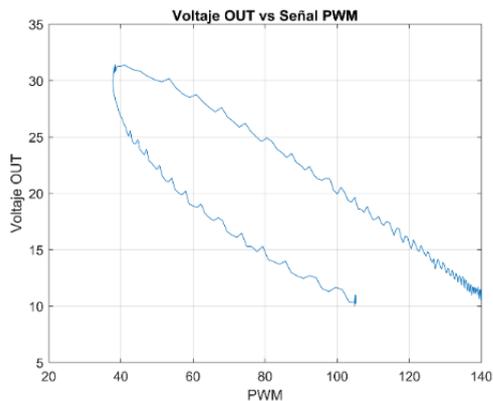
Figura 3.7: Gráficas Señal Vout VS PWM



Gráficas señal PWM y señal Vout, Adrián López & Diego Zapata.

Los gráficos presentados anteriormente se obtuvieron una vez que la planta ya se encontraba en funcionamiento debido a que al tomar muestras partiendo desde el reposo, los datos no convergen en una curva característica debido a la inercia de arranque del motor como se visualiza en la Figura 3.8.

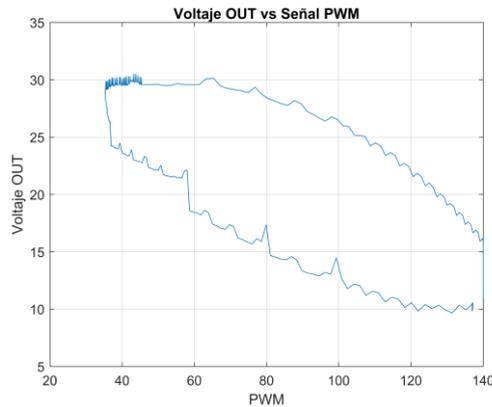
Figura 3.8: Gráfica Señal Vout VS PWM t=0



Gráfica señal PWM y señal Vout, partiendo desde el reposo, Adrián López & Diego Zapata.

Por otra parte al no dar tiempo de retardo al sistema para que este se estabilice se presentaban errores por efecto de inercias e histéresis en el motor como se puede visualizar en la Figura 3.9.

Figura 3.9: Gráfica Señal Vout VS PWM sin tiempo de estabilización



Gráfica señal PWM y señal Vout, partiendo desde el reposo, Adrián López & Diego Zapata.

### 3.3.3. Validación de la curva

Para la obtención de la curva de comportamiento se utilizó la ayuda de gráficas de dispersión, donde se puede agregar líneas de tendencia, una herramienta de Microsoft Excel, de tal forma que al promediar el grupo de muestras obtenido se obtuvo la siguiente ecuación con una aproximación del 98,79%:

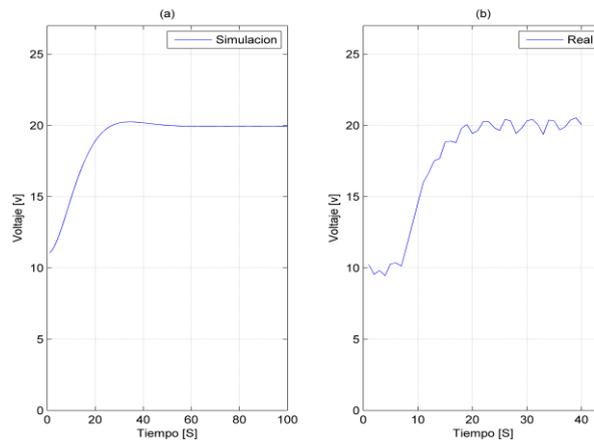
$$y = -0,2318x + 44,327 \quad EC(3.1)$$

$$R^2 = 0,9879 \quad EC(3.2)$$

Donde “y” representa el voltaje de salida del generador DC, “x” la señal PWM que ingresa al motor (0 a 255) y “R<sup>2</sup>” el coeficiente de determinación.

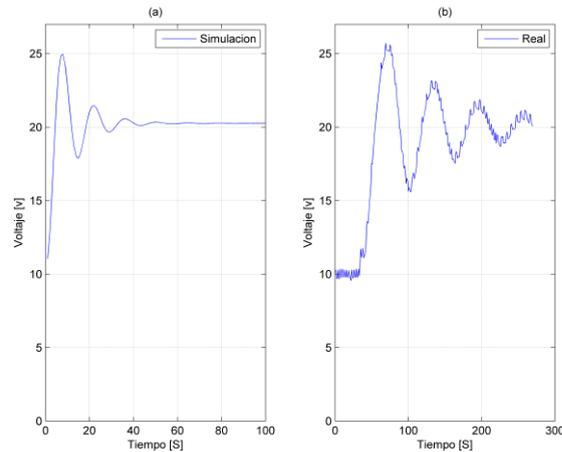
Al simular el comportamiento de la planta con la curva de comportamiento obtenida, se obtuvo los resultados mostrados en las Figuras 3.10 y 3.11.

Figura 3.10: Gráfica Señal Vout VS PWM, simulación y real.



Gráficas señal PWM y señal Vout , (a) simulación, (b) real,  $k_p=3$ , Adrián López & Diego Zapata.

Figura 3.11: Gráfica Señal Vout VS PWM, simulación y real.



Gráficas señal PWM y señal Vout, (a) simulación, (b) real,  $k_p=5$ , Adrián López & Diego Zapata.

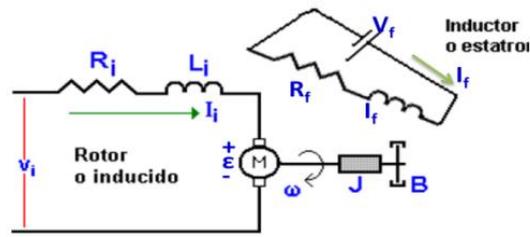
### 3.4. Modelo de la planta

Para determinar el modelo de la planta se toma en cuenta la composición de la misma, la cual consta de un motor DC acoplado a una carga, en este caso un generador DC.

#### 3.4.1. Modelo del motor DC

Un motor DC consta de rotor y estator, donde el campo generado por el estator es creado por la corriente que pasa por las bobinas del mismo mientras el rotor (inducido) es la parte giratoria del motor, de esta configuración se saca el esquema del motor presentado en la Figura 3.12.

Figura 3.12: Esquema Motor DC



Esquema Motor DC, (Alvarez Alvarado, 2011)

Dónde:

\$R\_f\$ = Resistencia bobinado del estator

\$L\_i\$ = Inductancia del bobinado del rotor

\$L\_f\$ = Inductancia bobinado del estator

\$I\_i\$ = Corriente en el bobinado del rotor

\$I\_f\$ = Corriente en el bobinado del estator

\$V\_i\$ = Voltaje de alimentación del rotor

\$V\_f\$ = Voltaje alimentación del estator

\$W\$ = Velocidad angular

\$R\_i\$ = Resistencia del bobinado del rotor

\$J\$ = Momento de inercia

\$B\$ = Coeficiente de rozamiento viscoso

En base al esquema del motor DC se determina la ecuación EC(3.3)

$$k_p I_i(t) = J \times \frac{d\omega(t)}{dt} + B \times \omega(t) \quad EC(3.3)$$

Despejando \$I\_i(t)\$ de la EC(3.3) y luego derivando con respecto al tiempo da como resultado:

$$I_i(t) = \frac{J \times \frac{d\omega(t)}{dt} + B \times \omega(t)}{k_p} \quad EC(3.4)$$

$$\frac{dI_i(t)}{dt} = \frac{J \times \frac{d^2\omega(t)}{dt^2} + B \times \frac{d\omega(t)}{dt}}{k_p} \quad EC(3.5)$$

En base a las ecuaciones EC(3.4) y EC(3.5) obtiene la EC(3.6), que describe el modelo matemático para un motor DC:

$$V_i - k_b \times \omega(t) = R_i \times \frac{J \times \frac{d\omega(t)}{dt} + B \times \omega(t)}{k_p} + L_i \times \frac{J \times \frac{d^2\omega(t)}{dt^2} + B \times \frac{d\omega(t)}{dt}}{k_p} \quad EC(3.6)$$

Nota. Para ver todos los pasos de la obtención del modelo matemático remitirse a Máquinas Eléctricas, Fraile M. J. 5ta Edición. (2003). España, Mc Graw Hill.

### 3.4.2. Modelo del Generador

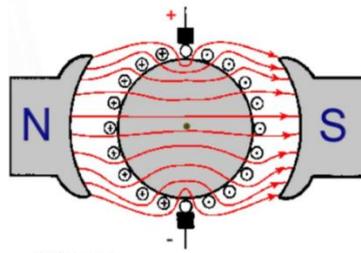
El principio de funcionamiento de un generador DC es la ley de Faraday en la que postula que “La fuerza electromotriz inducida en un circuito es proporcional a la variación de flujo magnético en la superficie limitada por dicho circuito.” (Costa Quintana & López Aguilar, 2007).

Lo cual en fórmula es:

$$\varepsilon = -\frac{d\Phi}{dt} \quad EC(3.7)$$

De aquí se tiene que la variación de flujo magnético en las espiras del rotor de un generador depende del ángulo de estas con la dirección del campo magnético, la longitud de las espiras y del número de espiras.

Figura 3.13: Principio de funcionamiento de un generador DC



Principio de funcionamiento de un generador DC, (Armengol Blanco, 2014)

De lo cual se tiene:

$$\varepsilon = v \times B * l * N \quad EC(3.8)$$

Dónde:

$\varepsilon$  = voltaje inducido en el conductor

$l$  = longitud del conductor

$v$  = velocidad del conductor

$N$  = número de espira

$B$  =vector densidad del flujo magnético

Una bobina está conformada por un cierto número de espiras y una cierta longitud de conductor de cobre, al circular una corriente por la bobina existe una caída de tensión, por tanto, la bobina tiene una inductancia y resistencia. Las espiras del inducido constituyen una bobina, por tanto, se representa por una inductancia y resistencia.

En la Figura 3.14, se muestra el circuito equivalente de un generador DC modelado en base a las ecuaciones de la FEM.

Figura 3.14: Circuito Equivalente de un generador CC



Circuito Equivalente de un generador CC , (Armengol Blanco, 2014)

Donde:

$V_a$  = Voltaje en los terminales

$L_a$  = Inductancia bobina de armadura

$I_a$  = Corriente de armadura

$E_a$  = Fuerza contra electromotriz

$R_a$  = Resistencia bobina de armadura

generada

En base a la segunda ley de Kirchhoff se tiene:

$$V_a = E_a - I_a \times R_a - L_a \frac{dI_a}{dt} \quad EC(3.9)$$

Sabiendo que

$$\varepsilon = E_a \quad EC(3.10)$$

Se tiene reemplazando EC(3.8) en EC(3.9):

$$V_a = v \times B * l * N - I_a \times R_a - L_a \frac{dI_a}{dt} \quad EC(3.11)$$

De donde la EC(3.11) representa el comportamiento del generador en base a  $v(t)$  y  $I_a(t)$  mientras que  $B, l, N, L_a$  son constantes.

## CAPÍTULO IV

### CONTROLADOR PREDICTIVO EN WLAN

Para este capítulo, el controlador se ha implementado dentro de una red privada, la tarjeta Raspberry Pi 3 Model B será el servidor, el mismo que albergará al controlador.

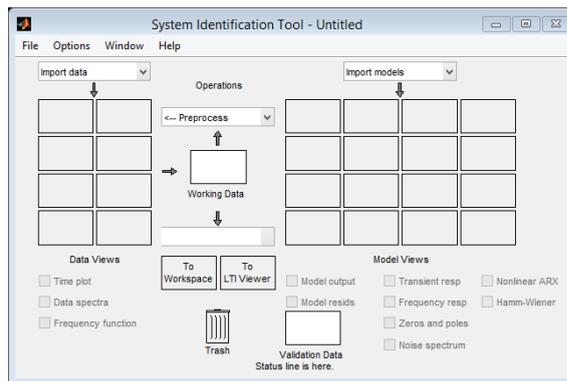
#### 4.1. Obtención del controlador Predictivo

El objetivo de utilizar un controlador estimador se presenta cuando existe pérdida de comunicación, su función principal a desarrollar es proveer de soporte al sistema y entregar información similar a la real hasta que la comunicación se restablezca. Tanto el controlador situado en el servidor, así como predictor en la planta, necesitan estimar el valor del otro extremo para que una vez retomada la comunicación tengan una respuesta cercana a la real.

##### 4.1.1. Estimador del Servidor-Controlador PID

Los datos obtenidos en la sección 3.3.2 (Obtención de la Curva de Comportamiento), son importados al software Matlab para utilizar una herramienta llamada IDENT, la cual permitirá determinar la función de transferencia de cada gráfica, con el propósito de obtener aquella que se apegue de mejor manera al sistema, siguiendo el proceso que se observará desde la Figura 4.1 hasta la Figura 4.8.

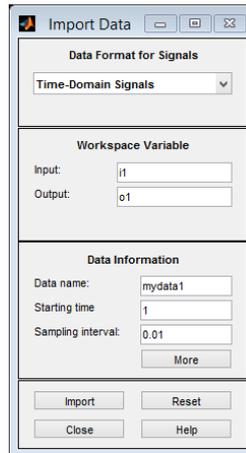
Figura 4.1: Ventana IDENT



Estimación Función de Transferencia herramienta IDENT de Matlab, Adrián López & Diego Zapata.

En la ventana de la Figura 4.1 se debe ingresar los datos exportados en el dominio del tiempo con un “Sampling Interval” de 0.01, ya que es el tiempo de muestreo propio de la placa Arduino Mega, como se muestra en la Figura 4.2.

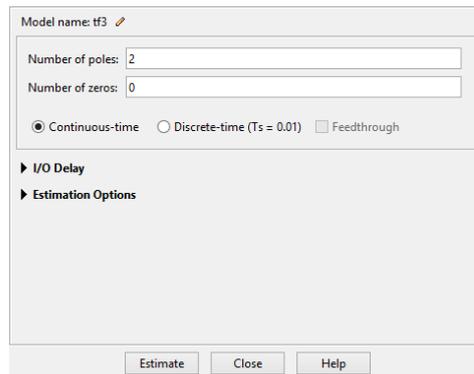
Figura 4.2: Ventana Import Data



Estimación Función de Transferencia herramienta IDENT de Matlab, Adrián López & Diego Zapata.

Con los datos importados se realiza un estimado de función de transferencia para cada grupo de datos importados.

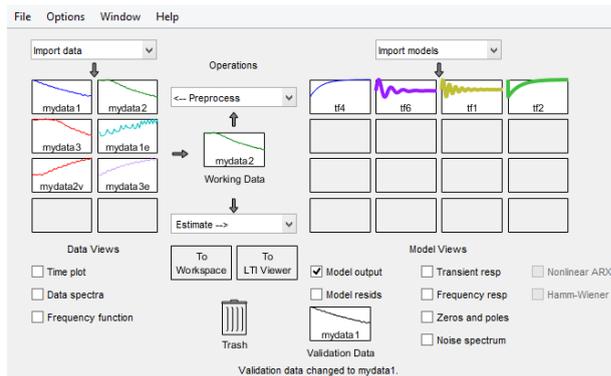
Figura 4.3: Número de polos y ceros del sistema



Numero de polos y ceros del sistema herramienta IDENT de Matlab, Adrián López & Diego Zapata.

El siguiente paso es establecer el número de polos y ceros del sistema, como se visualiza en la Figura 4.3, en el cual se eligió 2 polos y 0 zeros en base al modelo matemático del motor presentado en la sección 3.4.1. En la Figura 4.4 se presenta la ventana de la herramienta IDENT de Matlab, en la cual se realiza la estimación de función de transferencia.

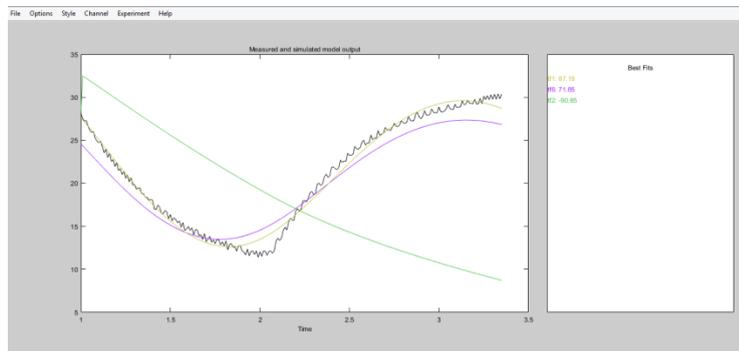
Figura 4.4: Estimación de función de transferencia



Estimación de Función de transferencia herramienta IDENT de Matlab, Adrián López & Diego Zapata.

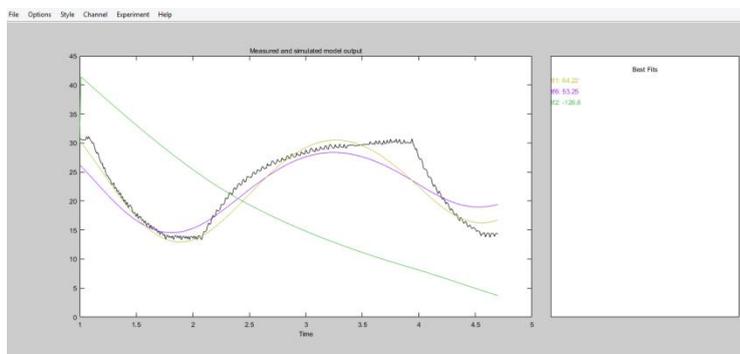
De todas las funciones de transferencia conseguidas se escoge la que presenta mejor aproximación comparándolas respecto a las muestras importadas en la herramienta IDENT, con ayuda de la opción “Model Output”, como se muestra en las Figuras 4.5, 4.6 y 4.7.

Figura 4.5: Modelamiento del sistema primera muestra



Modelamiento del sistema herramienta IDENT de Matlab, Adrián López & Diego Zapata.

Figura 4.6: Modelamiento del sistema segunda muestra



Modelamiento del sistema herramienta IDENT de Matlab, Adrián López & Diego Zapata.

De las funciones de transferencia calculadas se determina la de mejor estimación, en este caso es “tf6”, como se puede apreciar en la Figura 4.7, donde se detalla la función de transferencia que mejor se adapta al sistema, presentando una aproximación de 85.91%.

Figura 4.7: Resultados presentados por Matlab

```
tf6 =

    From input "u1" to output "y1":
    0.5537
    -----
    s^2 + 0.5606 s + 2.355

Name: tf6
Continuous-time identified transfer function.

Parameterization:
  Number of poles: 2   Number of zeros: 0
  Number of free coefficients: 3
  Use "tfdata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Estimated using TFEST on time domain data "mydata3".
Fit to estimation data: 85.91% (stability enforced)
FPE: 0.7556, MSE: 0.73
```

Resultados presentados por Matlab, Modelamiento del sistema tercera muestra herramienta IDENT, Adrián López & Diego Zapata.

$$\frac{V}{pwm} = \frac{0,5537}{s^2 + 0,5606s + 2,355} \quad EC(4.1)$$

La Función de Transferencia es discretizada con el comando “c2d” de Matlab:

$$\frac{V}{pwm} = \frac{0,00002763z^{-1} + 0,00002758z^{-2}}{1 - 1,994z^{-1} + 0,9944z^{-2}} \quad EC(4.2)$$

Con la función discretizada se obtiene la ecuación en diferencias, despejando V se tiene lo siguiente:

$$V = 1,994 V_{i-1} - 0,9944 V_{i-2} - 0,00002763 PWM_{i-1} + 0,00002758 PWM_{i-2} \quad EC(4.3)$$

La cual es el estimador de la planta que será usado por el controlador del servidor para estimar la respuesta de la planta real.

#### 4.1.1.1. Sintonización del PID

Para la sintonización de los valores óptimos del PID (kp, ki, kd) para disminuir el tiempo de estabilización y mejorar la respuesta al error, se usan algoritmos genéticos tomados del trabajo “Implementación de un Controlador Predictivo Basado en Eventos para un Sistema de Control en Red Inalámbrico” (Pillajo Angos C. G., 2017), usando como función de evaluación la ecuación estimadora de la planta

EC(4.3) en la sección 4.1.1, con lo cual se determina que los valores para el controlador son:

$$k_p = 0,0065$$

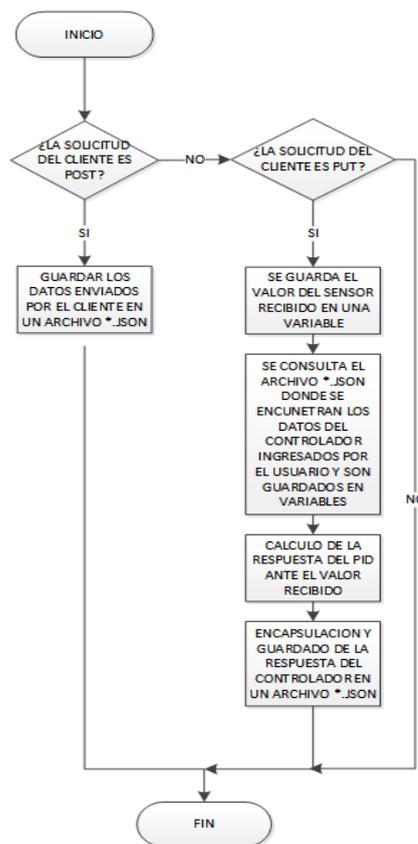
$$k_i = 45,2361$$

$$k_d = 0,001$$

#### 4.1.1.2. Algoritmo implementado en el Servidor

A continuación se presenta en la Figura 4.8 el algoritmo implementado en el servidor.

Figura 4.8: Algoritmo implementado en el Servidor



Algoritmo implementado en el servidor, Adrián López & Diego Zapata.

- Cuando el usuario cambia los valores de SP,  $k_p$ ,  $k_i$  o  $k_d$  envía una solicitud POST, ante esta solicitud el servidor únicamente guardará estos datos en un archivo para consultarlo posteriormente.
- Si la solicitud es PUT, significa que son datos enviados por el cliente (Placa Arduino), los datos recibidos por el servidor (dato del sensor y errores

anteriores) son decodificados y guardados en variables, a su vez se consulta los datos guardados en el archivo de la solicitud POST con lo cual se tendrá SP, kp, ki, kd, errores anteriores y el valor del sensado en la planta; estas variables son necesarias para el cálculo del PID.

- Una vez calculado la respuesta del PID se guarda la respuesta (valor PWM) y los 2 últimos errores (SP - valor sensado) de la planta en otro archivo al cual el cliente puede consultar para obtener la respuesta del servidor.

#### 4.1.2. Estimador del Cliente – Planta

En base al comportamiento de la planta, al aplicarse un controlador PID local con los parámetros de sintonización (comportamiento ideal de la planta), se obtienen datos muestreados y repitiendo los pasos de la sección 4.1.1, Figuras 4.1 a 4.7, las cuales detallan el proceso para la obtención de la función de transferencia discretizada utilizando la herramienta IDENT de Matlab; el resultado es el siguiente:

$$\frac{PWM}{E} = \frac{-1,641 z^{-1} + 0,6786 z^{-2}}{1 - z^{-1}} \quad EC(4.4)$$

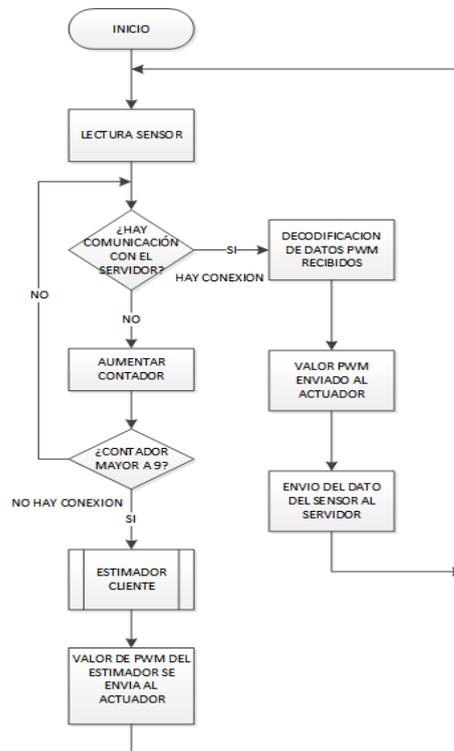
De esa ecuación se obtiene la ecuación en diferencia la cual es el estimador en la parte de la planta:

$$PWM = PWM_{i-1} - 1,641 E_{i-1} + 0,6786 E_{i-2} \quad EC(4.5)$$

##### 4.1.2.1. Algoritmo implementado en la planta

A continuación se presenta en la Figura 4.9 el algoritmo implementado en la planta.

Figura 4.9: Algoritmo implementado en la Planta



Algoritmo implementado en la planta, Adrián López & Diego Zapata.

En el cliente el algoritmo desarrolla los siguientes pasos:

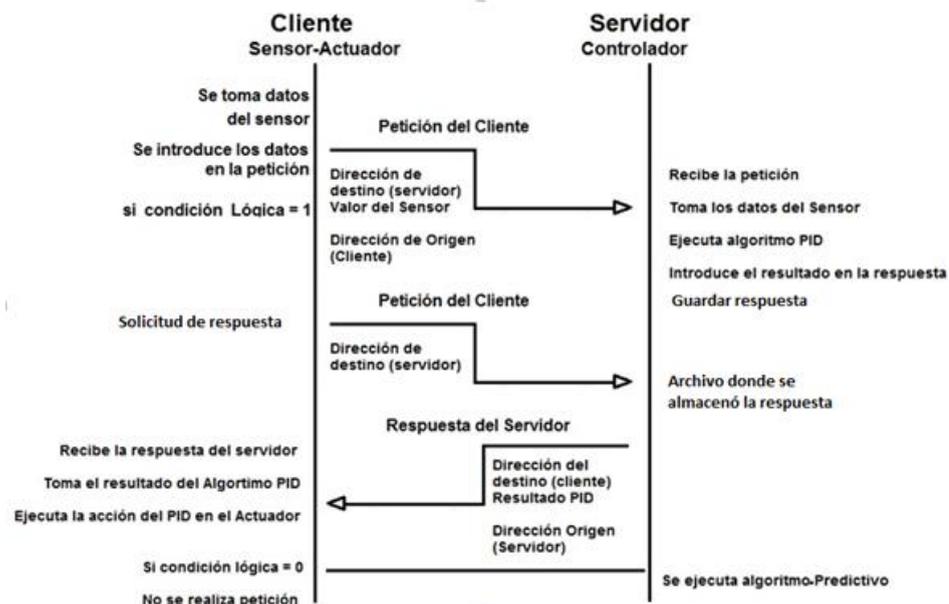
- Medición del voltaje en el generador ( $V_{out}$ ).
- Conexión con el servidor, se tiene dos opciones:
  - Al establecer conexión, solicita la respuesta del controlador (valor PWM), lo decodifica y envía el valor del PWM al actuador, por último envía el valor sensado al inicio al servidor para un nuevo cálculo.
  - Si la conexión con el servidor falla, incrementa un contador y lo vuelve a intentar, esta operación será realizada 9 veces (el valor de 9 es un valor escogido empíricamente en el cual espera la conexión con el servidor por un tiempo prudente), una vez que el contador es mayor a 9 considera que la conexión con el servidor ha fallado y usa el estimador del cliente para controlar la planta y el proceso empieza nuevamente.

## 4.2. Comunicación cliente-servidor

Para la comunicación cliente – servidor, que se detalla en la Figura 4.10, se debe tomar en cuenta que es el cliente quien siempre empieza la comunicación, por lo cual, el cliente sigue los siguientes pasos:

- Primero reserva un espacio de memoria para almacenar todos los datos del formato JSON
- Segundo transforma el dato a ser enviado en una cadena de caracteres.
- Tercero se da el formato JSON a la cadena de caracteres.
- Mediante método PUT se envía los datos al servidor.
- Se hace la lectura de la respuesta mediante el método GET el cual nos retorna una cadena de caracteres.
- A esta cadena se la convierte en formato JSON y se decodifica los datos necesarios en este caso la respuesta del controlador.

Figura 4.10: Comunicación Cliente - Servidor



Comunicación Cliente – Servidor, (Pillajo Angos C. G., 2017)

### 4.2.1. Formato JSON para la comunicación, envío-recepción de datos del cliente-servidor

El formato JSON ordena los datos y permite extraerlos fácilmente sin métodos de discriminación, para esto el formato JSON tiene la siguiente forma:

```
{"sensor":0,"error":10,"resp":0,"Out_a1":0,"Sensor_a1":0,"SP":10,"Sensor_a2":0}
```

Donde todos los datos son enviados entre corchetes ({}), el nombre de cada valor enviado se encuentra entre comillas (“ ”), y el valor de esta variable después de dos puntos (:), separando de las demás variables por una coma(,), de esta manera se escoge el valor deseado de cada variable sin importar el tipo de dato que sea (Int, Float, String).

### **4.3. Implementación del control basado en eventos**

Para evitar la monopolización del ancho de banda y reducir su tiempo de utilización, el controlador basado en eventos determina cuando es necesario el uso de la red de comunicación, para esto se vale de dos recursos los cuales son “Detector de eventos” y “Generador de entrada de control”.

#### **4.3.1. Detector de Eventos**

Para realizar el detector de eventos se compara el valor recibido de la respuesta del servidor (valor PWM), con el generado por el estimador, al comparar con una constante predefinida por el usuario en base a las necesidades y limitaciones de la planta, se determina si es necesario enviar datos al servidor y optar por dos opciones, de ser necesario, solicitar el funcionamiento del controlador, caso contrario trabajar únicamente con el estimador programado en la planta. Este recurso identificará cuando el estimador de la planta funciona igual o similar que el controlador en la nube (servidor), en este caso no es necesario el uso de la red de comunicación, tampoco del controlador en la nube (servidor), ya que el estimador se puede encargar momentáneamente de la planta.

#### **4.3.2. Generador de Entrada de Control**

En el generador de entrada de control, el servidor es quien determina si es necesario seguir enviando una respuesta, ya que si el error en la planta es mínimo o menor a un error tolerado, relega la labor de controlar la planta al estimador, hasta que se detecte que el error supera el valor tolerado, de esta forma se reduce el uso de la red por ciertos lapsos de tiempo y por consiguiente el cliente dejará de solicitar la intervención del servidor.

## CAPÍTULO V

### CONTROLADOR PREDICTIVO EN WAN

En este capítulo se detallan configuraciones de las herramientas de software necesarias para albergar el controlador en la “nube” y el monitoreo de la planta, con el propósito de acceder al control y monitoreo del sistema desde cualquier parte del mundo.

#### 5.1. Configuración del Servidor en la nube

Para albergar el servidor en la nube se requiere contratar un servicio de hosting, el cual proveerá de un espacio para alojar archivos, para esto se utiliza la plataforma SmarterASP.net, la cual proporciona un periodo de 60 días gratuitos con opción a renovación, su configuración se detalla en los anexos.

##### 5.1.1. Obtención de dominio y carga de archivos en el servidor

Con el espacio de almacenamiento de la página WEB es necesario un dominio para asociar la página y poder acceder a ella, en la plataforma SmarterASP.net se puede comprar un dominio y cargar los archivos en el servidor siguiendo los pasos detallados en anexos.

##### 5.1.2. Direccionamiento para la comunicación cliente-servidor

En la comunicación en una red local WLAN se asigna una dirección IP estática para el servidor, donde todos los dispositivos que interaccionan en el sistema de control WNCS deben estar dentro del mismo rango de direcciones IP, al ser una red privada, todos los dispositivos tendrán los 3 primeros octetos iguales los cuales son 192.168.0.XXX. En el caso del controlador WAN, la dirección del servidor es la que se ha asignado en el servicio de HOST, en este caso el nombre del dominio es <http://ine4c.com/>; En el código del cliente, esta dirección se debe configurar para que pueda enlazarse a él.

La dirección que tiene el cliente sigue estando dentro de la red privada, es decir tendrá una dirección como 192.168.0.XXX, esta dirección no es visualizada por el servidor, ya que el router se encarga de traducir la dirección IP privada a una dirección IP pública, de esta manera el servidor recibe la petición y envía una respuesta a la IP pública, por su parte el router realiza el proceso contrario identificando el paquete y direccionando al cliente.

## 5.2. Migración de servidor local a servidor en la nube

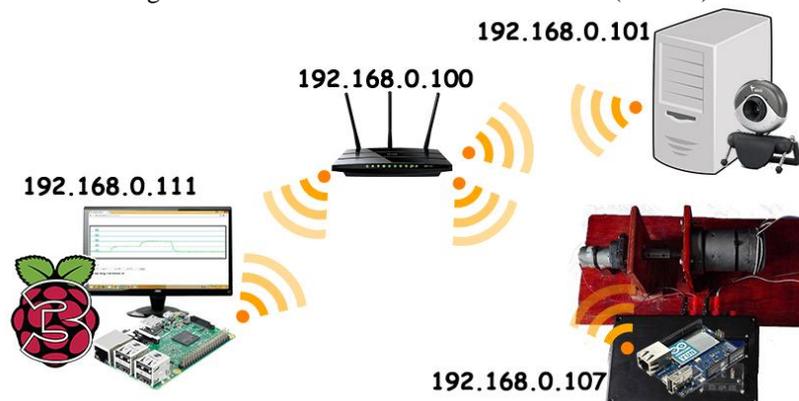
Al momento de migrar del servidor local (WLAN) a uno albergado en la nube (WAN), surgen dos cambios principales:

- El cambio de dirección IP como ya se habló en la anterior sección.
- La lectura de archivos por parte del cliente se ve entorpecida por procesos de seguridad, para esto es necesario añadir una etiqueta <html> al inicio de todo archivo de solo lectura que se desee leer ya que sin este cambio los datos recibidos llegarán en blanco.

## 5.3. Diferencias direccionamiento Servidor Local Vs Servidor en la Nube

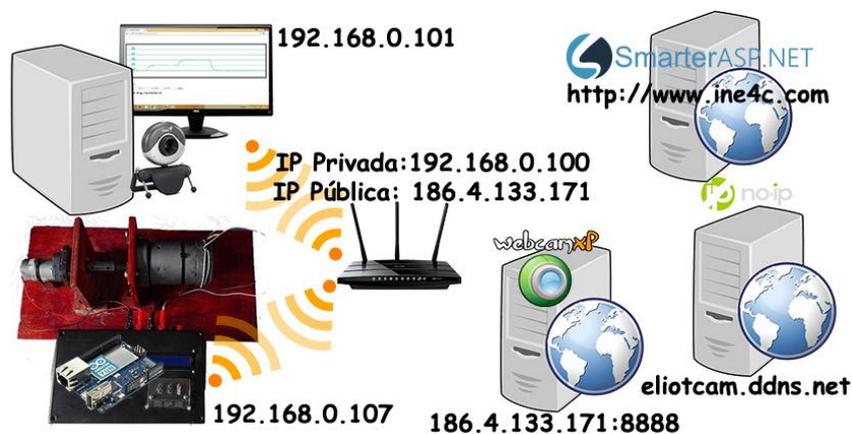
El direccionamiento de los dispositivos y servidores que intervienen en el sistema de control en red inalámbrica, tanto en el controlador WLAN como en el WAN se detalla en las Figuras 5.1 (Servidor WLAN) y 5.2 (Servidor WAN)

Figura 5.1: Direccionamiento Servidor Local (WLAN)



Comunicación Cliente – Servidor WLAN, Adrián López & Diego Zapata.

Figura 5.2: Direccionamiento Servidor en la Nube (WAN)



Comunicación Cliente – Servidor WAN, Adrián López & Diego Zapata.

## **5.4. Configuración cámara IP**

En el proyecto se utilizó una cámara WEB adaptada a cámara IP, la cual servirá para monitoreo de la planta, para su configuración se utilizó dos herramientas de Software.

### **5.4.1. Configuración software WEB CAM XP 5**

Al no disponer de cámaras IP se usa una cámara web como cámara IP usando el software WEB CAM XP5 el cual asocia la dirección IP de la máquina a la cual está conectada y acceder desde un navegador web, para configurar la cámara con el software se siguen los pasos detallados en anexos.

### **5.4.2. Configuración Software NO-IP**

El software NO-IP, provee un servicio de DNS Dinámico, este servicio se encarga de mantener un nombre de dominio fijo, el cual está relacionado con una dirección IP. Los pasos para su configuración se detallan en anexos.

### **5.4.3. Configuración de puertos de comunicación para la utilización de la cámara IP.**

Para realizar la configuración de puertos de comunicación para la utilización de la Cámara WEB adaptada a Cámara IP, se debe acceder al router, lo cual se logra ingresando la dirección del Router en un navegador o también con la Dirección IP Pública de la red, el usuario y la contraseña se lo consigue llamando al proveedor del Servicio de Internet, los pasos para configurar los puertos de comunicación son detallados en anexos.

## CAPÍTULO VI

### PRUEBAS Y RESULTADOS

A continuación, se presenta las pruebas realizadas y resultados obtenidos al trabajar con el PID y la planta, utilizando controladores diferentes (Planta, WLAN, WAN), lo que generó diferentes retardos por efecto del albergue del controlador.

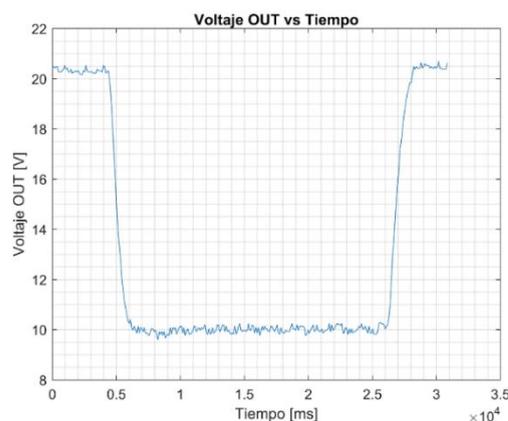
#### 6.1. Funcionamiento PID en la planta, PID WLAN y PID WAN

Para las pruebas que se realizarán a continuación, se toman los parámetros de sintonización obtenidos en la sección 4.1.1, el siguiente paso es establecer cambios de SP en la planta entre 10 voltios (mínima velocidad del motor al romper inercia) y 20 voltios (para evitar forzar al motor), adicionalmente se toman datos del tiempo requerido en el sistema para completar todo un proceso (sensado, envío y recepción de datos, cálculo del controlador y control de la planta), para el cálculo de la latencia en cada controlador.

##### 6.1.1. Pruebas y Resultados, Funcionamiento PID En Planta.

Para las pruebas con el controlador en la planta (sin red WLAN) se programa a la planta para trabajar inicialmente en un SP de 20 V, cuando el controlador considere que se ha estabilizado en este punto cambia el SP a 10 V y nuevamente una vez estabilizado lo vuelve a cambiar a 20 V el resultado obtenido es el siguiente:

Figura 6.1: Voltaje OUT vs Tiempo (en Planta)



Comportamiento de la planta con controlador en la misma, Adrián López & Diego Zapata.

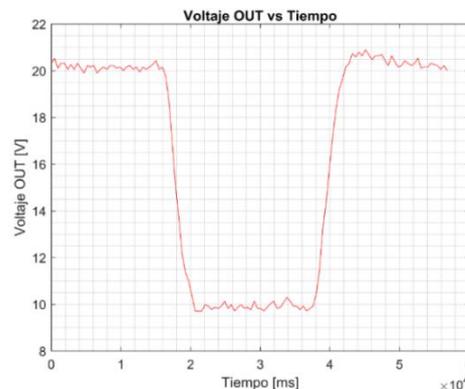
En la Figura 6.1, se aprecia la prueba del funcionamiento de la planta, donde se inicia con un voltaje de 20v, al cabo de 4 segundos el SP cambia a 10v, tomándole al sistema 2s para lograr estabilizarse y mantenerse en este nivel de voltaje en el

intervalo de 6s – 26s (20 segundos), posteriormente retornara al SP inicial que fue de 20v, la estabilización en el cambio de SP nuevamente es de 2 segundos.

### 6.1.2. Pruebas y Resultados, Funcionamiento PID en Red WLAN.

Para las pruebas con el controlador en red WLAN se programa a la planta en un servidor local en este caso se lo hace en un sistema embebido como es la tarjeta Raspberry PI 3 Model B, donde desde la interfaz programada en el servidor se manipula el SP de forma similar al de la sección 6.1.1., el resultado obtenido es el siguiente:

Figura 6.2: Voltaje OUT vs Tiempo (en red WLAN)



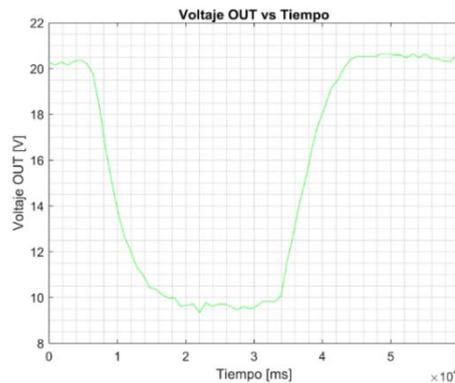
Comportamiento de la planta con controlador en red WLAN, Adrián López & Diego Zapata.

En la Figura 6.2, se puede observar la prueba del funcionamiento de la planta en red WLAN, donde se inicia con un voltaje de 20v, después de 14 segundos el SP cambia a 10v, tomándole al sistema 4s para lograr estabilizarse y mantenerse en este nivel de voltaje en el intervalo de 18s – 36s (18 segundos), posteriormente retornara al SP inicial que fue de 20v, la estabilización en el cambio de SP ahora es de 5 segundos.

### 6.1.3. Pruebas y Resultados, Funcionamiento PID en red WAN.

Para las pruebas con el controlador en red WAN se programa a la planta en un servidor albergado en la nube, la manipulación del SP se lo hace de la misma forma que en la sección 6.1.2., el resultado obtenido es el siguiente:

Figura 6.3: Voltaje OUT vs Tiempo (en red WAN)



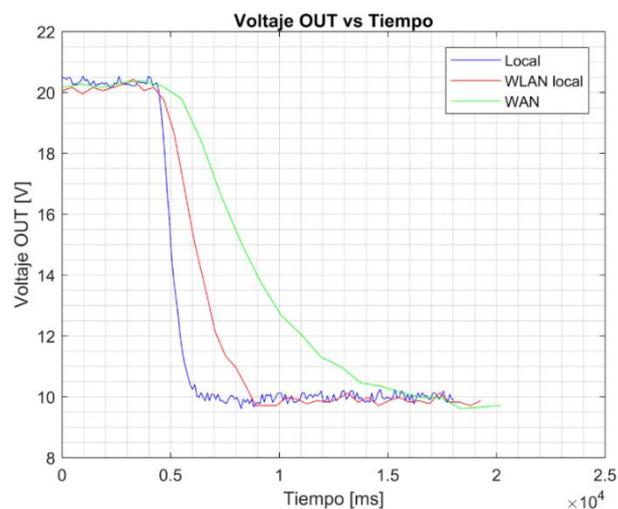
Comportamiento de la planta con controlador en red WAN, Adrián López & Diego Zapata.

En la Figura 6.3, se muestra la prueba del funcionamiento de la planta en red WAN, donde se inicia con un voltaje de 20v, al pasar 4 segundos el SP cambia a 10v, tomándole al sistema aproximadamente 12s para lograr estabilizarse y mantenerse en este nivel de voltaje en el intervalo de 16s – 30s (14 segundos), posteriormente retornara al SP inicial que fue de 20v, la estabilización en el cambio de SP se mantiene en 12 segundos.

#### 6.1.4. Comparación PID en la Planta, PID WLAN y PID WAN.

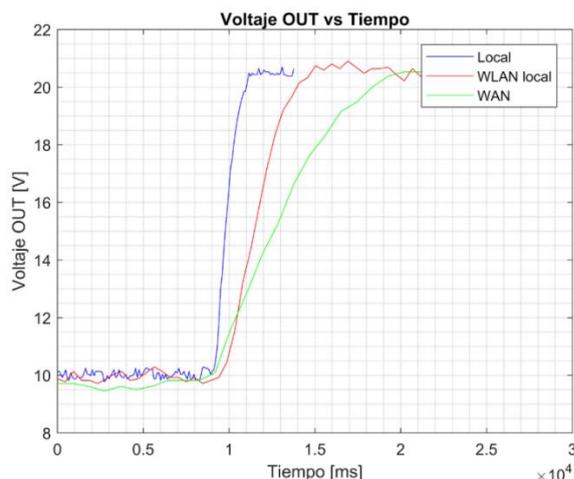
Al comparar el comportamiento de los sistemas (PID en la Planta, PID WLAN y PID en la nube), al momento de cambiar de SP entre los valores de voltaje, 20 y 10 se obtuvo los siguientes resultados:

Figura 6.4: Comparación Voltaje OUT vs Tiempo (bajada)



Comparación del comportamiento de la planta con diferentes albergues del controlador (Planta, WLAN, WAN), Adrián López & Diego Zapata.

Figura 6.5: Comparación Voltaje OUT vs Tiempo (subida)



Comparación del comportamiento de la planta con diferentes albergues del controlador (Planta, WLAN, WAN), Adrián López & Diego Zapata.

Como se observa en la Figura 6.4 y Figura 6.5 la mejor respuesta es la del controlador en la planta (Local) ya que el tiempo de estabilización es de tan solo 2 segundos con una latencia entre cada muestra de 97 ms promedio.

En cambio, el tiempo de estabilización del controlador en red WLAN es de 5 segundos, más del doble que en el controlador de la planta y con una latencia de 470 ms.

Finalmente, el controlador en la nube, tiene un tiempo de estabilización de 12 segundos, más del doble que el tiempo en el controlador WLAN y 6 veces el tiempo del controlador en la Planta con una latencia de 918 ms.

Al realizar diferentes pruebas en red WAN, desde diferentes lugares no se ve afectado la respuesta obtenida ni las latencias presentes, como se pudo apreciar al realizar las pruebas en el “IV Congreso Internacional de Ciencia, Tecnología e Innovación para la Sociedad (CITIS 2017)”, desarrollado en la ciudad de Guayaquil, en el cual fue expuesto el presente trabajo con la planta localizada en la ciudad de Quito.

Tabla 1. Valores comparativos entre controladores

	PID local	PID WLAN	PID WAN
latencia [ms]	97	470	918
ts [ms]	2000	5000	12000

Comparación de latencia y tiempo de estabilización entre controladores (Planta, WLAN, WAN), Adrián López & Diego Zapata.

Con los resultados obtenidos se aprecia el gran efecto causado por la latencia en la red y la aparente desventaja e impactó a la respuesta del sistema al usar sistemas en red inalámbrica.

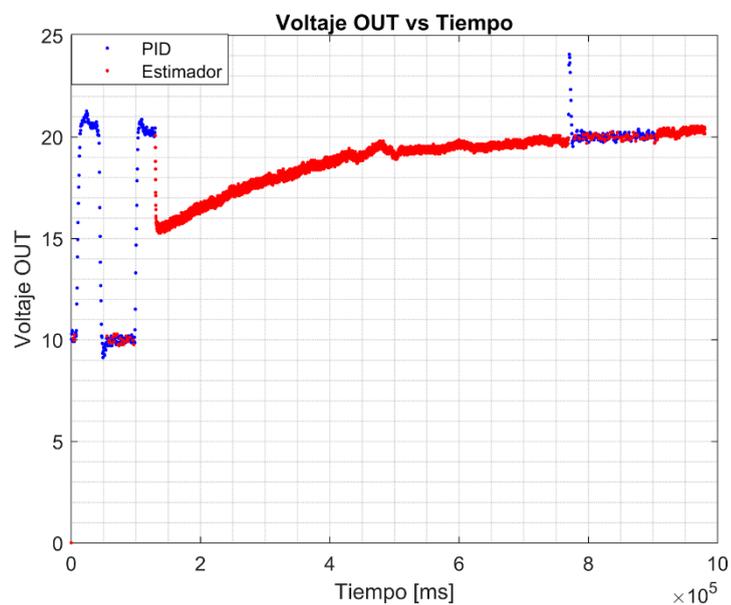
## 6.2. Comportamiento del controlador con y sin comunicación.

Para pruebas de comunicación se dispone de un botón encargado de simular la pérdida total de comunicación, de esta forma se hace comportar a la planta de la siguiente manera:

- Inicia estabilizando en un SP de 20, posteriormente cambiar a 10 y nuevamente a 20 para comprobar el buen funcionamiento del controlador en la nube.
- Se simula la pérdida total de la comunicación donde el estimador se encarga del control.
- Pasado un tiempo apropiado para observar el comportamiento del estimador se retorna la comunicación y por ende el control en la nube.

Al realizar dicho comportamiento el resultado obtenido es el siguiente:

Figura 6.6: Comparación Voltaje OUT vs Tiempo (subida)



Comportamiento de la planta con y sin comunicación, Adrián López & Diego Zapata.

De la Figura 6.6 en ciertos intervalos en los cuales la planta se estabiliza en 10V se aprecia que el estimador entra en funcionamiento esto debido al control por eventos donde considera innecesario el uso del controlador en la planta.

En un segundo intervalo; desde aproximadamente el segundo 105 hasta el segundo 705; donde se simula la pérdida de comunicación y entra en funcionamiento el estimador, este reduce drásticamente el voltaje de salida, alrededor de un 25% pero se ve que esa reducción sucede por un breve tiempo y retorna al nivel del SP deseado.

A Partir del segundo 705 hasta el 900 se retoma la comunicación y el controlador en la nube retoma el control, pero al momento de la transición de estimador a controlador se da una desestabilización por lo que se presenta un pico que llega a 25 V esto es debido a la que el controlador no conoce el valor en el que se encuentra la planta y tan solo lo estima.

Finalmente, en el último tramo desde el segundo 900 hasta el 1000 se simula una vez más la pérdida de comunicación donde el estimador vuelve a funcionar, pero ya no presenta la gran disminución de voltaje como lo hizo la primera vez y en su lugar mantiene el nivel del voltaje oscilando alrededor del SP (dentro de un rango de 0.5 voltios ya que son los valores de las perturbaciones propias de la planta).

## CONCLUSIONES

Para el proceso de obtención de la Curva de Comportamiento, se obtuvo datos de entrada (Voltaje PWM) y salida (Vout en los terminales del Generador DC), fue necesario establecer periodos de tiempo, ya que ante una velocidad constante el generador varia su voltaje de salida hasta lograr estabilizarse; por efecto de la histéresis; por lo que se concluyó que la respuesta del generador ante una entrada de velocidad no es inmediata y la necesidad de un tiempo para su estabilización es evidente.

El estimador presenta un muy buen rendimiento donde este es capaz de mantener controlado el nivel de voltaje dentro de los niveles de tolerancia establecidos por el usuario en base a las perturbaciones de la planta, a pesar de que en un inicio presenta un comportamiento inestable su funcionamiento se corrige al funcionar por un periodo de tiempo donde sus parámetros se ajustan al de la planta como se observó en las pruebas con y sin comunicación.

Si bien el control en la nube se lo puede hacer desde cualquier parte del mundo a través de la interfaz web, la ubicación desde donde se lo haga no afecta el comportamiento o las latencias como se lo nota en las pruebas realizadas en red WAN, debido a que las latencias corresponden a la comunicación entre el cliente y el servidor, el cual está ubicado en un lugar fijo y no importa desde donde se manipule la interfaz de control, por lo que las latencias no dependen del lugar donde se controle sino de la ubicación del cliente (la planta) y el servidor. Lo cual se corroboró al presentar el proyecto en el congreso CITIS 2017, realizado en la ciudad de Guayaquil.

Se pudo observar las limitaciones y desventajas reflejadas al comparar los resultados de los controladores local, red WLAN y red WAN; pero la viabilidad de la utilización de estos sistemas sigue siendo muy rentable y variable según la aplicación, ya que en sistemas donde el tiempo no sea un parámetro crítico, puede funcionar correctamente y en caso de que se requiera una respuesta más rápida a la del control WAN, se puede aplicar el concepto de control en neblina que no es más que usar servidores locales para el control (Control WLAN) en lugar del control WAN y estos servidores también pueden enviar los datos al servidor en la nube y tener monitoreo desde cualquier parte del mundo.

## RECOMENDACIONES

Para mejorar el comportamiento del algoritmo predictivo se recomienda hacerlo funcionar en paralelo al controlador en la nube para mantenerlo actualizado al momento de entrar a funcionar.

Usar un transistor PNP para el control del MOSFET para evitar que el control sea invertido 0-100% y 255-0%.

Poner una dirección IP estática a la Raspberry para evitar cambiar las direcciones en la programación del cliente cada vez q el servidor cambia de IP.

Dar permisos de lectura y escritura a los archivos del servidor para evitar que el cliente no pueda acceder a ellos.

Al usar JSON en el cliente (Arduino) liberar el espacio en la memoria cada vez q se recibe información.

Para la implementación del control WAN, se recomienda que la planta esté directamente enlazada al Router de frontera entregado por el proveedor del servicio de Internet, ya que al añadir en la red Routers como repetidores, se presentan inconvenientes al momento de la apertura de puertos.

Para trabajos futuros se recomienda probar diferentes tipos de envío de datos en lugar del formato JSON como puede ser SOAP para buscar reducir las latencias en el envío y recepción de datos.

Se puede implementar todo un proceso donde intervengan varios actuadores y sensores a la vez con el fin de estudiar el consumo del ancho de banda de la red y cómo interfieren los demás dispositivos entre si además del impacto de todos los procesos en el servidor.

## LISTA DE REFERENCIAS

- Alvarez Alvarado, M. S. (11 de diciembre de 2011). Modelo matemático de un motor de corriente continua separadamente excitado: Control de velocidad por corriente de armadura. Guayaquil, Ecuador: Instituto de Ciencias Físicas, Escuela Superior Politécnica del Litoral.
- Arias, M. Á. (2015). *Aprende Programacion WEB con PHP y MySQL*. IT Campus Academy.
- Arias, M. Á. (2017). *Aprende Programacion Web con PHP y MySQL*. IT Campus Academy.
- Armengol Blanco, B. (2014). *Máquinas Eléctricas*. Bolivia: Facultad Nacional de Ingeniería Ingeniería Eléctrica e Ingeniería Electrónica.
- Beruvides López, G., Haber Guerra, R., & Castaño Romero, F. (16 de junio de 2016). Sistemas ciberfísicos basados en métodos de inteligencia computacional para la monitorización de procesos de fabricación. Estado actual y líneas futuras. Madrid, España: Universidad Politecnica de Madrid.
- Callicó, G., & Celma, J. (2016). El nuevo fetichismo del internet de las cosas en la sociedad kidult. *Aloma: revista de psicologia, ciències de l'educació i de l'esport Blanquerna*, 77-86.
- Campos Verdi, K. J. (noviembre de 2016). DISEÑO DE ACOUPLE MECATRÓNICO PARA. *ARDUINO MEGA 2560 R3*. (L. T. Ojeda, Ed.) Lima, Peru: PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ. Recuperado el 10 de 12 de 2017, de <http://arduino.cl/arduino-mega-2560/>
- Castillo, A. A. (2017). *Curso de Programación Web: JavaScript, Ajax y jQuery*. It Campus Academy.
- Chapman, S. J. (2012). *MÁQUINAS ELECTRICAS*. Mexico D.F.: MCGRAW-HILL/INTERAMERICANA EDITORES, S.A.
- Contreras Rodríguez, I. M. (20 de mayo de 2015). DISEÑO DE SITIO WEB PARA PRESENTAR LAS ACTIVIDADES BENÉFICAS A LOS PATROCINADORES REALES Y POTENCIALES DE LA

ORGANIZACIÓN NO LUCRATIVA ASOCIACIÓN DESAFÍO.  
Guatemala, Guatemala: Universidad Galileo.

Costa Quintana, J., & López Aguilar, F. (2007). *Interacción Electromagnética, Teoría Clásica*. Barcelona - España: Reverté.

Daugherty, P., Banerjee, P., Negm, W., & Alter, A. E. (2017). El crecimiento no convencional. . *Petrotecnia*, 51-64.

de la Fuente López, E., & Mazaeda Echevarría, R. (2016). *Industria 4.0*. Valladolid, España: Universidad de Valladolid. Escuela de Ingenierías Industriales.

Fraile, M. J. (2003). *Máquinas Eléctricas 5ta Edición*. España: Mc Graw Hill.

GIL GARCÉS, L. A., & RINCÓN GAVIRIA, J. L. (2014). CONTROL PID PARA EL CONTROL DE VELOCIDAD DE UN MOTOR DC. Pereira, Colombia: UNIVERSIDAD TECNOLÓGICA DE PEREIRA.

Gourley, D., Totty, B., Sayer, M., Aggarwal, A., & Reddy, S. (2002). *HTTP: The Definitive Guide: The Definitive Guide*. Reilly media.

Guinaldo, M., Sánchez, J., & Dormido, S. (2017). Control en red basado en eventos: de lo centralizado a lo distribuido. *Revista Iberoamericana de Automática e Informática Industrial RIAI*, 16-30.

Hernandez Bonilla, A., & Vargas Plazas, F. (2013). *Diseño e implementación de una tarjeta de adquisición de datos inalámbrica para el control del helicóptero 2-DOF de Quanser®*. Bogotá, Colombia : Pontificia Universidad Javeriana.

Heurtel, O. (2015). *PHP 5.6: desarrollar un sitio web dinámico e interactivo*. Barcelona, España: Ediciones ENI.

Ingeniería MCI . (s.f.). *ARDUINO MEGA 2560 R3*. (L. T. Ojeda, Ed.) Santiago, Chile. Recuperado el 10 de 12 de 2017, de <http://arduino.cl/arduino-mega-2560/>

Juárez Sanchez, M., Morales Sánchez, D., & Sandoval Smith, C. (03 de abril de 2017). *Diseño e implementación de un sintetizador analógico, utilizando amplificadores operacionales*. México D.F, México: Instituto Politécnico Nacional.

- López, E. A., & Sierra, J. D. (2015). Control de un motor de inducción utilizando algoritmo predictivo:¿ Formar la cultura audiovisual y multimedial?. . *REVISTA POLITÉCNICA*,, 73-84.
- Marcosig, E. P., Felicioni, F. E., & Zanini, A. (2014). Self-Trigging Based on Lyapunov with Adaptive Control Law for WNCS. *IEEE Latin America Transactions*, 1242-1249.
- Monsalve Posada, J. F., & Arias Londoño, A. (2015). Redes inalámbricas para el control de procesos industriales en tiempo real. *REVISTA POLITÉCNICA*, 83-93.
- Mora Castillo, J. A. (2016). Serialización/deserialización de objetos y transmisión de datos con JSON: una revisión de la literatura. *Tecnología en Marcha Vol. 29*, 118-125.
- Moran, R., Teragni, M., & Zabala, G. (2017). A Concurrent Programming Language for Arduino and Educational Robotics. *XXIII Congreso Argentino de Ciencias de la Computación*. La Plata: Universidad Nacional de La Plata.
- Natsys. (24 de febrero de 2014). *Todo sobre MySQL*. Obtenido de [https://books.google.com.ec/books?id=GS3kAgAAQBAJ&printsec=frontcover&hl=es&source=gbs\\_ge\\_summary\\_r&cad=0#v=onepage&q&f=false](https://books.google.com.ec/books?id=GS3kAgAAQBAJ&printsec=frontcover&hl=es&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false)
- Neila, O., Roshany-Yamchi, S., Gnyszka, B., & Limón, D. (2015). Control Predictivo Generalizado en un PLC Siemens Simatic S7-1200. *Actas de XXXVI Jornadas de Automática*, (págs. 983-988). Bilbao.
- Pérez, J. A. (2007). *Electrónica digital*. Madrid: Delta.
- Petru, L., & Mazen, G. (2015). PWM Control of a DC Motor Used to Drive a Conveyor Belt. En P. Engineering (Ed.), *International Symposium on Intelligent Manufacturing and Automation*. 25, págs. 299 - 304. ELSEVIER.
- Pillajo Angos, C. (01 de julio de 2014). CPS Cyber-Physical Systems. *Introducción a Sistemas Cyber Físicos* . Medellin, Colombia: Universidad Pontificia Bolivariana.
- Pillajo Angos, C. G. (2017). *IMPLEMENTACIÓN DE UN CONTROLADOR PREDICTIVO BASADO EN EVENTOS PARA UN SISTEMA DE CONTROL*

*EN RED INALÁMBRICO*. Medellín: Universidad Pontificia Bolivariana.  
Obtenido de <https://repository.upb.edu.co/handle/20.500.11912/3111>

RASPBERRY PI FOUNDATION. (s.f.). *RASPBERRY PI 3 MODEL B*. (R. P. FOUNDATION, Ed.) Recuperado el 10 de 12 de 2017, de RASPBERRY PI 3 MODEL B

Reimann, D., & Maday , C. (2017). Enseñanza y aprendizaje del modelado computacional en procesos creativos y contextos estéticos. *EDUCATION IN THE KNOWLEDGE SOCIETY(EKS)*, 87 - 97.

Reina, G. A., & Vélez, R. M. (2015). TELECONTROL DOMÓTICO DE CASA DE CAMPO. Guayaquil, Guayas, Ecuador.

Romero Alamán, J. (20 de julio de 2017). *M-flashcarding como recurso de aprendizaje del HTML*. Cataluña, España: Universitat Jaume I.

Sobrevilla, G., Hernández, J., Velasco-Elizondo, P., & Soriano, S. (2017). Aplicando Scrum y Prácticas de Ingeniería de Software para la Mejora Continua del Desarrollo de un Sistema Ciber-Físico-Applying Scrum and Software Engineering Practices to Continu. *ReCIBE*, 1-15.

The MathWorks, Inc. (1994-2018). *MathWorks*. Obtenido de <https://la.mathworks.com/help/ident/index.html>

Velásquez, S., Pinto, L., & Rosales, L. (2017). CONTROL DE SISTEMA DE PRIMER ORDEN CON RETARDO BASADO EN EVENTOS. *UNEXPO Universidad Ciencia y Tecnología*, 36-43.

Vigoroux, C. (2015). *Aprender a desarrollar con JavaScript*. Barcelona, España: Ediciones ENI.

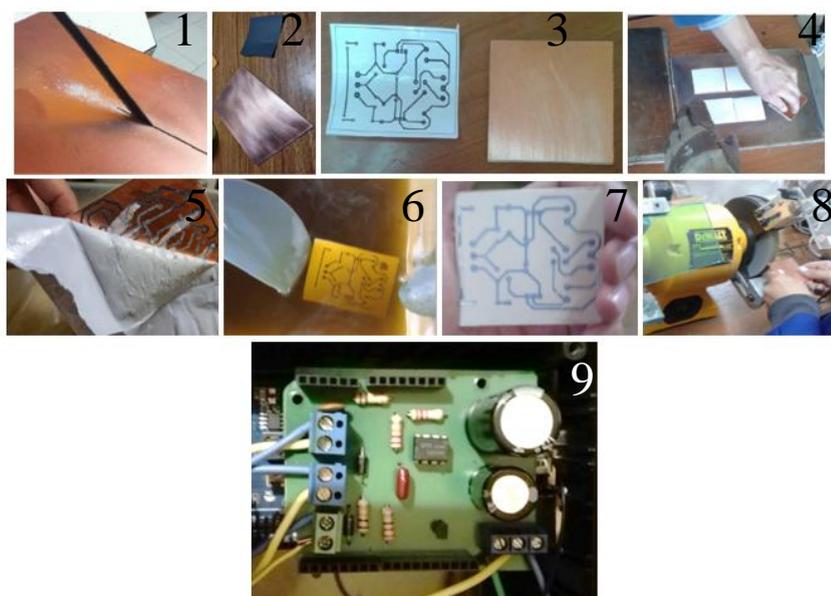
Zhang, D., Wang, Q.-G., & Yu, L. (2017). *Filtering and Control of Wireless Networked Systems*. Suiza: Springer.

## ANEXOS

### Elaboración placa de potencia y acondicionamiento de señal

En la Figura A.1 se observa el proceso de creación de la placa para el control y medición del sistema.

Figura A.1: Elaboración placa de potencia y acondicionamiento de señal



Creación (Circuito Control de Velocidad + Circuito medición de Voltaje), 1) Recorte de la placa, 2) Lijado de la placa, 3) Posicionamiento del papel Transfer, 4) Transferencia del circuito impreso a la placa, 5) Remoción del papel Transfer, 6) y 7) Proceso de degradado de la placa, 8) Esmerilado de la placa, 9) Placa Soldada y aplicada máscara antisoldante, Adrián López & Diego Zapata.

### Configuración del Servidor en la nube

- Primero en la página de la plataforma se crea una nueva cuenta presionando el botón “Get Started”. Como se muestra en la Figura A.2.

Figura A.2: Página principal de SmarterASP.net



Página principal de SmarterASP.net, Adrián López & Diego Zapata.

- Se selecciona el paquete que se desea y se llena los datos de la nueva cuenta, la cual estará activa una vez que se se confirme el correo electrónico ingresado, como se detalla en la Figura A.3.

Figura A.3: Formulario de registro

The registration form is divided into several sections:

- Your Order:**
  - Hosting Plan: 60 Days Free Trial
  - Pricing:
 

Hosting Fee (50 x 60Days)	\$0
Setup Fee	\$0
Coupon Code	<input type="text"/> <input type="button" value="Apply"/>
<b>Your Total</b>	<b>\$0 (Credit Card NOT Required!)</b>
- Domain Name:**

All of our domain names start at \$12/year. You can purchase them in your hosting control panel after you setup your Trial or Paid hosting account.
- Member Info:**
  - Login Name:
  - Email:  \*You CANNOT change login name once you sign up.
  - Confirm Email:  \*Valid Email Required to Receive Verification Email!
  - Password:
  - Confirm Password:

I have read and agree to the [terms and conditions of use](#)

Formulario con los datos necesarios para crear una nueva cuenta, Adrián López & Diego Zapata.

- En la cuenta creada se ingresa al PC para ver las páginas WEB creadas y los dominios asociados, como se muestra en la Figura A.4.

Figura A.4: Página principal de la cuenta SmarterASP.net

The dashboard includes the following elements:

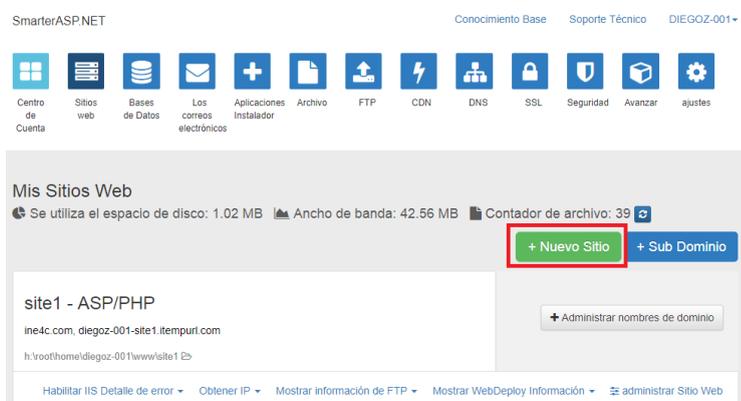
- Navigation icons: hostings, Dominios, Complementos/Afiliado, Facturas, ajustes, Soporte Técnico.
- Balance: \$0.00
- Section: **Mis Cuentas de Hosting** (+ Añadir nueva cuenta de alojamiento)
- Account list:
 

ine4c.com (Basic Plan)	<input type="button" value="ENTRAR AL PC"/>
Fecha de caducidad: Jul 21, 2018	<a href="#">Renovar Cuenta</a> <a href="#">Actualizar</a> <a href="#">Ver planos de Información</a>

Página principal donde se administra los hostings, Adrián López & Diego Zapata.

- En la página principal del usuario se puede crear sitios WEB y configurar la organización de carpetas y archivos del sitio, con lo que el hosting queda configurado, como se presenta en la Figura A.5.

Figura A.5: Administración de sitios WEB

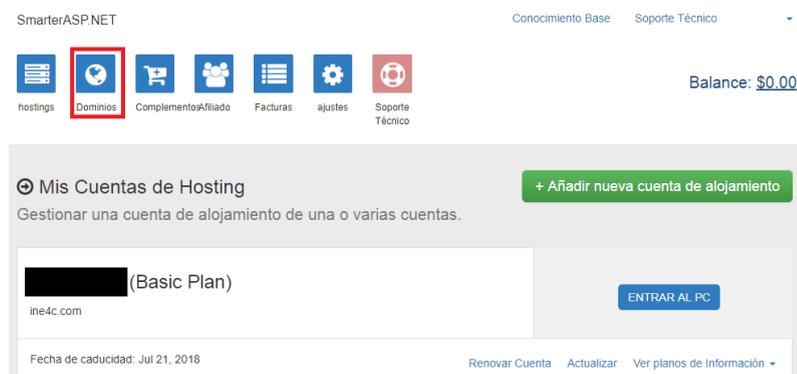


Página para la administración de los sitios WEB, Adrián López & Diego Zapata.

### Obtención de dominio y carga de archivos en el servidor

- En la página principal del usuario se ingresa en Dominios, como se muestra en la Figura A.6.

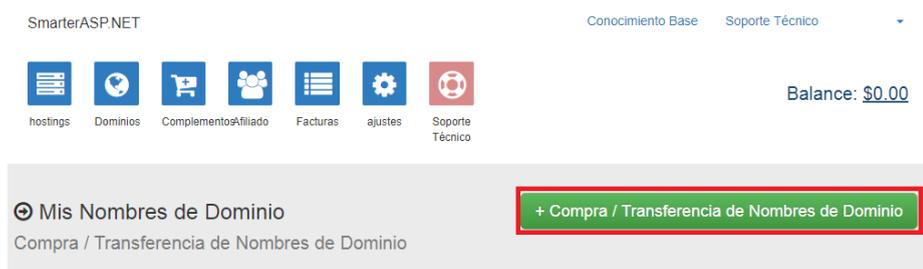
Figura A.6: Pestaña Dominios Página Principal del Usuario



Pestaña Dominios Página Principal del Usuario, Adrián López & Diego Zapata.

- Se selecciona Compra/Transferencia de Nombres de Dominios, La Figura A.7 detalla este paso.

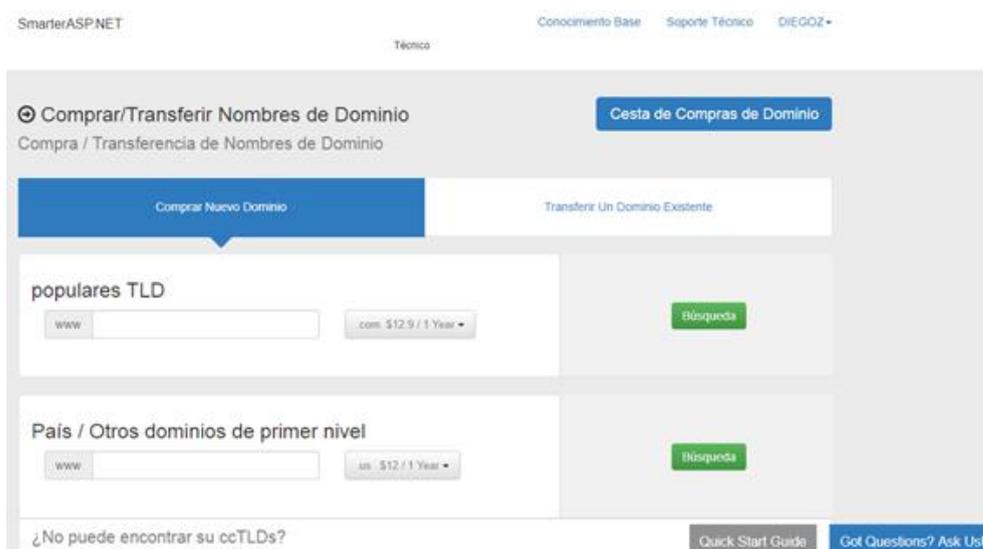
Figura A.7: Compra/Transferencia de Nombres de Dominios



Compra/Transferencia de Nombres de Dominios, Adrián López & Diego Zapata.

- En la Figura A.8 se selecciona el tipo de dominio y el nombre del mismo.

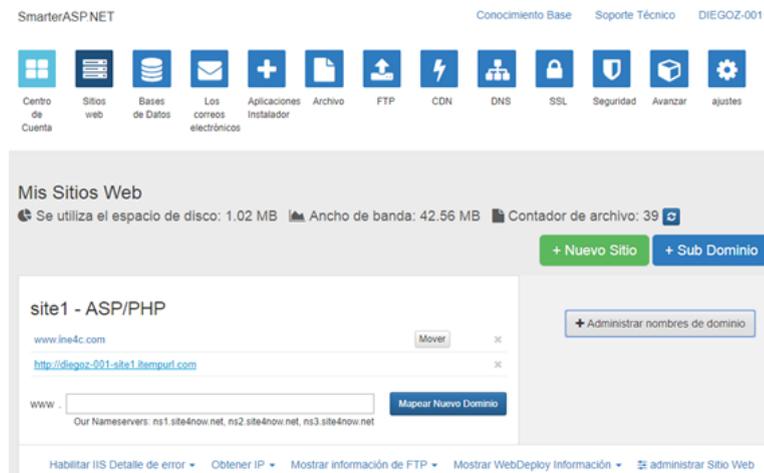
Figura A.8: Tipo y Nombre de Dominio



Tipo y Nombre de Dominio, Adrián López & Diego Zapata.

- Se efectiviza el pago del dominio y de vuelta en la página de administración de los sitios WEB se asocia el sitio WEB con el dominio creado en “Administrar nombres de dominio”. Por último, se mapea el dominio creado con el sitio WEB con lo cual se puede acceder a la página WEB, ver Figura A.9.

Figura A.9: Mapeo el dominio creado con el sitio WEB

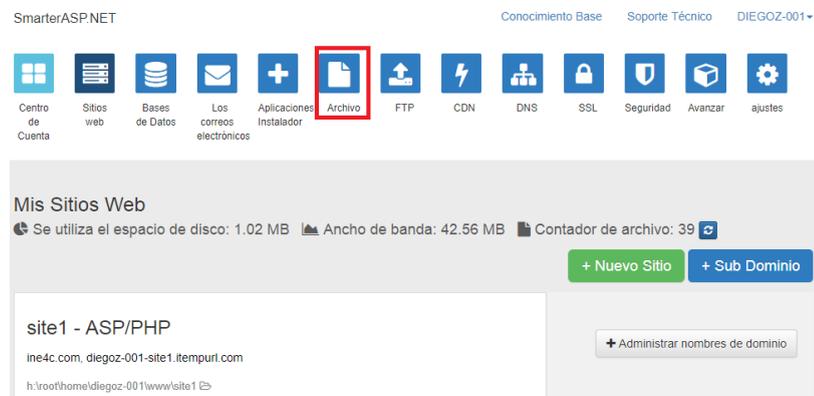


Mapeo el dominio creado con el sitio WEB, Adrián López & Diego Zapata.

A continuación se presenta los pasos para la carga de archivos en el servidor:

- En la página de administración de páginas WEB se ingresa en Archivos, como se ve en la Figura A.10.

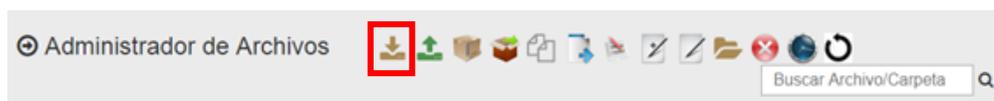
Figura A.10: Pestaña para la Subida de Archivos al Servidor



Pestaña para la Subida de Archivos al Servidor, Adrián López & Diego Zapata.

- En administrador de archivos se selecciona subir.

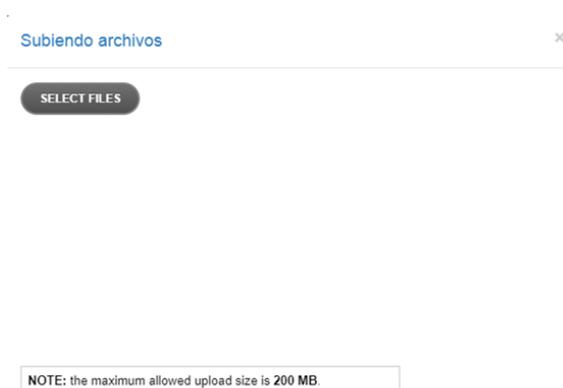
Figura A.11: Icono para subir Archivos



Icono para subir Archivos, Adrián López & Diego Zapata.

- En el recuadro abierto se selecciona los archivos de la página WEB y se suben, como se muestra en la Figura A.12.

Figura A.12: Espacio para Importar los Archivos que se Subirán



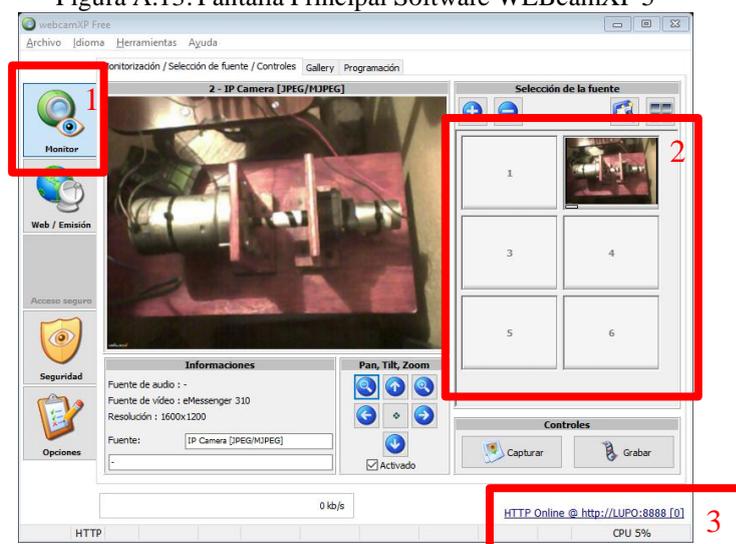
Espacio para Importar los Archivos que se Subirán, Adrián López & Diego Zapata.

Con lo cual la página WEB se encuentra lista para trabajar.

### Configuración software WEB CAM XP 5

Al inicializar el Software la pantalla principal mostrará todas las Cámaras WEB conectadas al ordenador, como se visualiza en la Figura A.13.

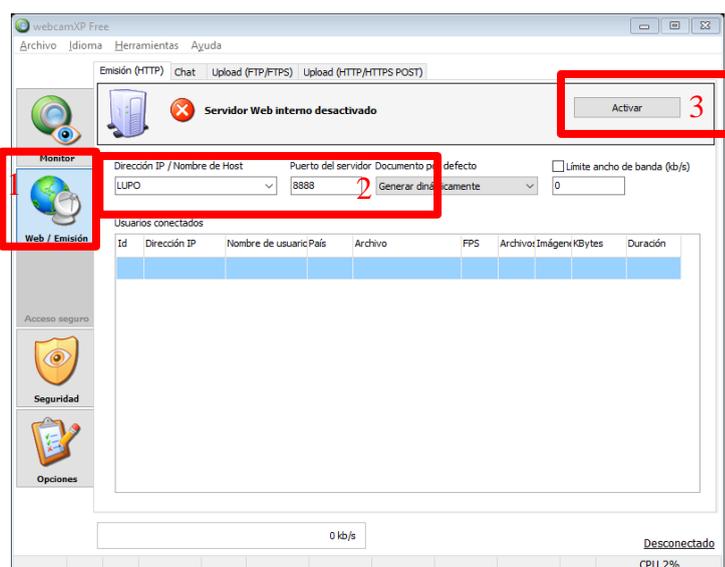
Figura A.13: Pantalla Principal Software WEBcamXP 5



Software WEBcamXP 5, 1) Pestaña Monitor, 2) Cámaras WEB conectadas, 3) Dirección del Servidor Interno, Adrián López & Diego Zapata.

A continuación, en la parte izquierda de la pantalla que se muestra en la Figura A.14, se selecciona la pestaña “WEB / Emisión”, en donde se establece el número de puerto por el cual se comunicará, así como se muestra el nombre o la dirección del equipo al cual está conectada la Cámara WEB.

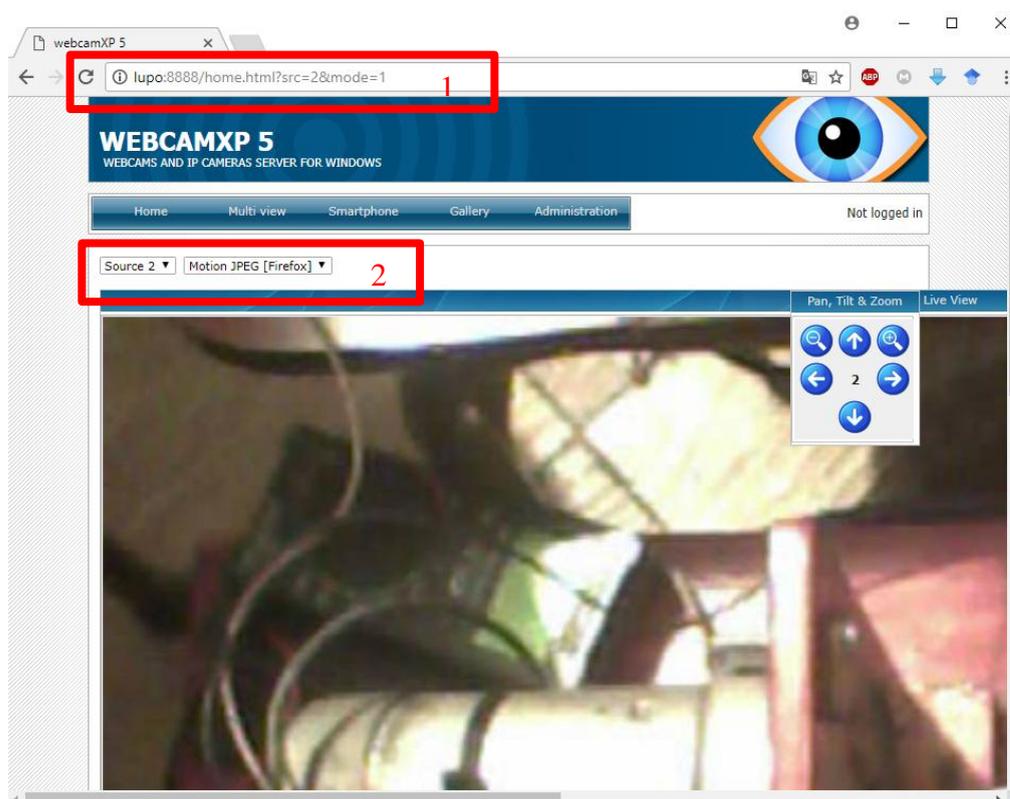
Figura A.14: Selección Número de Puerto de la Cámara WEB



Selección número de puerto de la cámara WEB, 1) Pestaña WEB / Emisión, 2) Dirección IP/ Nombre de Host y Puerto del servidor, 3) Activación del Servidor WEB Interno, Adrián López & Diego Zapata.

Una vez que esté activado el servidor interno, se retorna a la pestaña Monitor (Figura A.13 bloque 1) y se accede a la dirección del Servidor Interno (Figura A.13 bloque 3), con lo cual se redirigirá al navegador en el que se muestra la dirección del Servidor Interno (Figura A.15 bloque 1) y el video de la Cámara WEB, el mismo que puede ser visualizado accediendo mediante el navegador, desde cualquier equipo conectado a la red, como se ve en la Figura A.15. Adicional se debe seleccionar la opción “Motion JPEG [Firefox]” (Figura A.15 bloque 2).

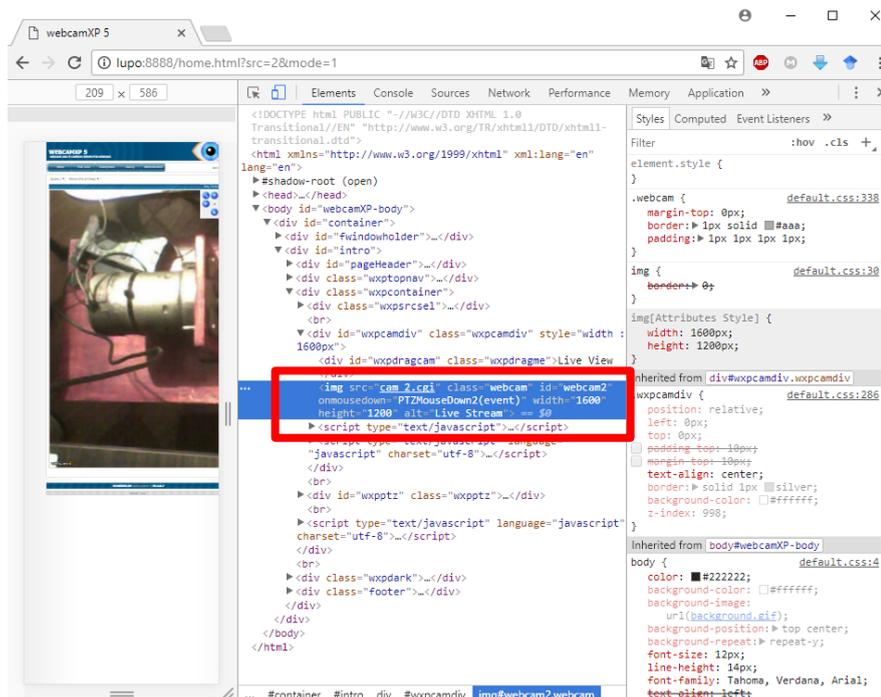
Figura A.15: Servidor Interno Cámara WEB



Servidor Interno Cámara WEB, 1) Dirección Servidor Interno, 2) opción “Motion JPEG [Firefox]”, Adrián López & Diego Zapata.

El Servidor Interno de la cámara muestra información que no se desea visualizar tanto en el Servidor WLAN, como en el Servidor WAN, por lo tanto, se debe aislar únicamente el archivo que guarda el video, lo cual se logra al dar click derecho en el video y seleccionando la opción Inspeccionar (esto se realizó en el navegador Google Chrome), como se visualiza en la Figura A.16.

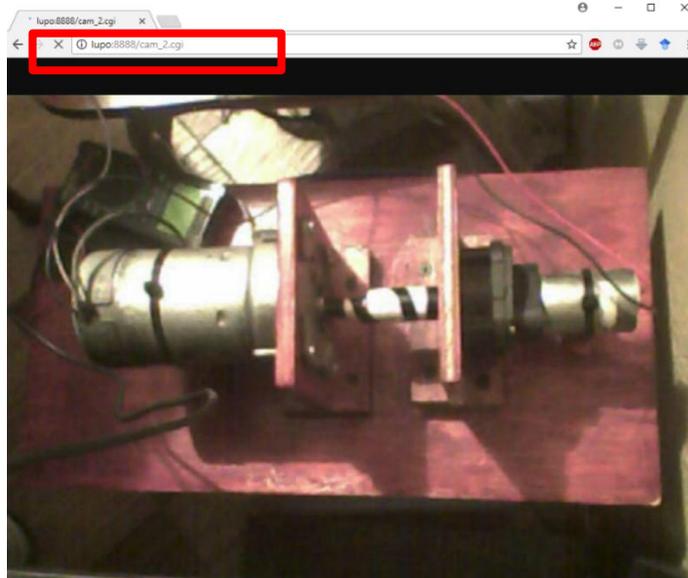
Figura A.16: Archivo de video Servidor Interno Cámara WEB



Archivo de video Servidor Interno Cámara WEB, Adrián López & Diego Zapata.

Al archivo en el cual se guarda el video es “cam2\_ cgi”, en la Figura A.17 se comprueba el acceso al archivo de video del Servidor Interno de la Cámara WEB, cambiando la dirección a la que se debe acceder.

Figura A.17: Dirección Archivo de video Servidor Interno Cámara WEB

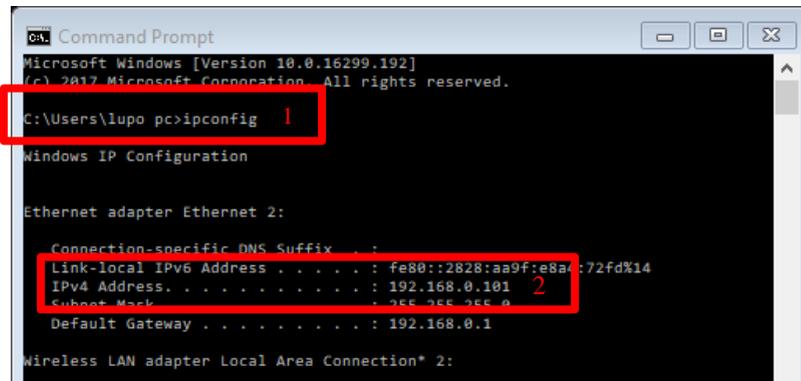


Dirección archivo de video Servidor Interno Cámara WEB, Adrián López & Diego Zapata.

Como se aprecia en las Figuras A.13 – A.17, en la dirección se muestra el nombre del equipo en el que se encuentra conectada la Cámara WEB, para relacionar este

nombre con una Dirección IP, se ejecuta el comando “ipconfig” en el símbolo de sistema de Windows, como se detalla en la Figura A.18.

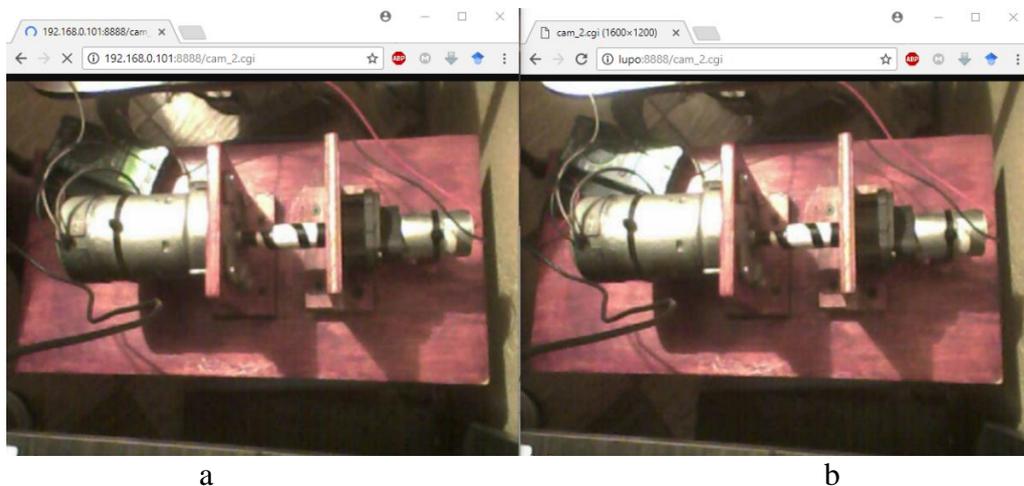
Figura A.18: Dirección IP del Ordenador de la Cámara WEB



Dirección IP del Ordenador al cual se encuentra conectada la Cámara WEB, 1) Nombre del ordenador “lupo”, 2) Dirección IP del ordenador 192.168.0.101, Adrián López & Diego Zapata.

Como se visualiza en la Figura A.18, la Dirección IP del ordenador “lupo”, es 192.168.0.101, lo cual se comprueba en la Figura A.19.

Figura A.19: Dirección IP y Nombre del Ordenador



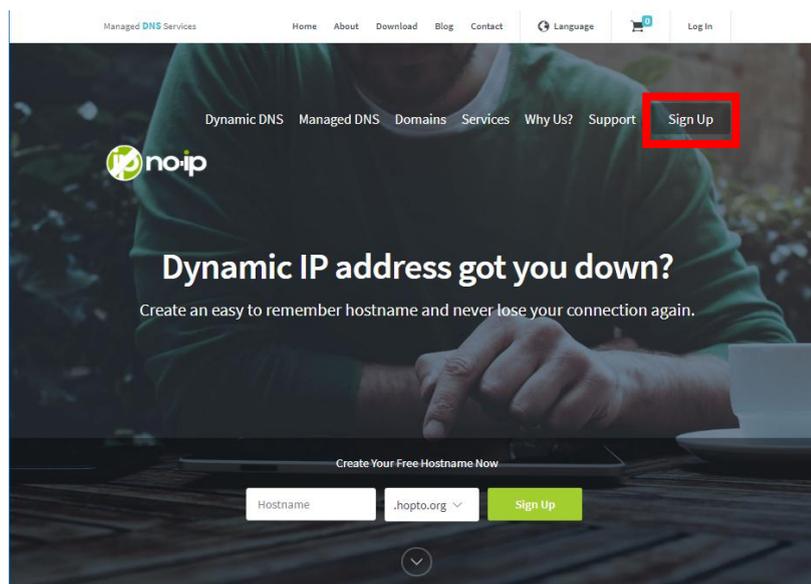
Dirección IP del Ordenador al cual se encuentra conectada la Cámara WEB, a) Dirección IP del Ordenador, b) Nombre del Ordenador. Adrián López & Diego Zapata.

En el caso del servidor WLAN, para visualizar la planta, se debe acceder a la dirección mostrada en la Figura A.19.a, por otra parte, para el servidor WAN se necesita la ayuda de otro Software el cual se detalla a continuación.

### Configuración software NO-IP

Ingresar a la página y crear una cuenta, presionando el botón “Sing Up”, como se detalla en la Figura A.20.

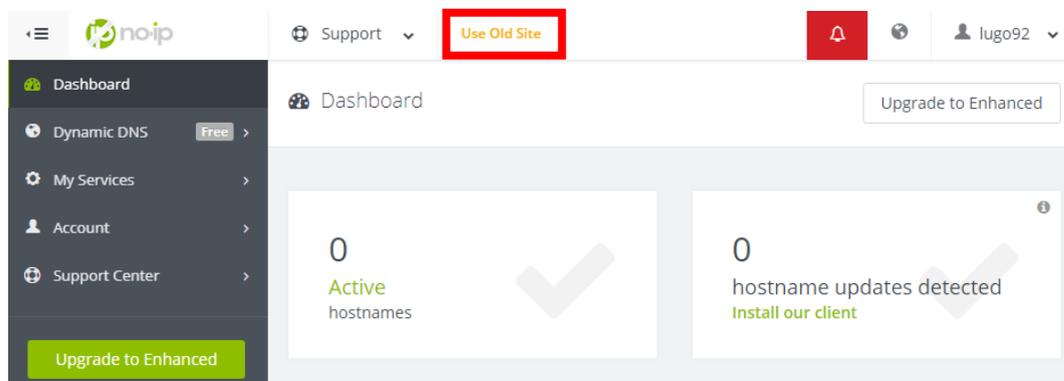
Figura A.20: Página principal No-IP



Página principal NO-IP, Adrián López & Diego Zapata.

Una vez que se ha registrado, en la pantalla principal, se selecciona la opción “Use Old Site”, como se visualiza en la Figura A.21, ya que el sitio antiguo de la página, proporciona mayor información.

Figura A.21: Pantalla principal No-IP



Pantalla principal NO-IP, Adrián López & Diego Zapata.

Al estar en el sitio antiguo se selecciona la opción “Add Host”, en la pestaña que se despliega se introduce el nombre del host, seguido de ddns.net, esta opción viene por defecto, la dirección IP a la cual estará relacionada, será la IP Pública de la red, por lo cual es necesario que se configure en una máquina que se encuentre dentro de la red en la que esté conectada la planta, como se detalla en la Figura A.22.

Figura A.22: Creación del Host

Hosts/Redirects

> Add Host 1

Manage Hosts

Manage Groups

Download Client

Upgrade to Enhanced

Need Help?

Support Center

Troubleshooting Guide

Dynamic Update Client

Support Ticket

Contact Us

Upgrade to Priority Support

### Add a host

Fill out the following fields to configure your host. After you are done click 'Create Host' to add your host.

Own a domain name?  
Use your own domain name with our DNS system. [Add](#) or [Register](#) your domain name now or read more for pricing and features.

#### Hostname Information

Hostname: eliotcam ddns.net 2

Host Type:  DNS Host (A)  DNS Host (Round Robin)  DNS Alias (CNAME)

Port 80 Redirect  Web Redirect  AAAA (IPv6)

IP Address: 186.4.133.171 3

Assign to Group: - No Group - [Configure Groups](#)

Enable Wildcard: Wildcards are a Plus / Enhanced feature. [Upgrade Now!](#)

Creación de Host, NO-IP, 1) Add Host, 2) Nombre del Host, 3) Dirección IP Pública de la red, Adrián López & Diego Zapata.

A continuación, se dirige al final de la página como se muestra en la Figura A.23, y se presiona el botón “Add Host”, con lo cual la creación del Host estará terminada.

Figura A.23: Confirmación del Host

#### Mail Options

MX Record	MX Priority
Enter the name of your external mail exchangers (mx records) as hostnames not IP addresses. <input type="text"/>	5

If you would like a more MX records, please upgrade to [No-IP Plus](#) or [Enhanced](#).

Restablecer Add Host

Confirmación de Host, NO-IP, Adrián López & Diego Zapata.

Una vez que se pulsa Add Host, la pantalla se actualizará y mostrará los hosts que estén activos en la cuenta, este servicio se oferta de dos maneras, el primero que es gratuito, permite un máximo de 3 servidores DNS dinámicos, estos servidores tienen una vigencia de 30 días, después de este plazo se debe acceder nuevamente para volver a habilitarlos por otro periodo de 30 días. El segundo es de pago, la ventaja que ofrece es que los servidores no tienen caducidad.

Figura A.24: Host Activo

The screenshot shows the 'Manage Hosts' interface. On the left is a sidebar with navigation options: 'Hosts/Redirects', 'Add Host', 'Manage Hosts' (selected), 'Manage Groups', 'Download Client', 'Upgrade to Enhanced', 'Need Help?', 'Support Center', 'Troubleshooting Guide', and 'Dynamic Update Client'. The main content area has a green notification bar at the top: 'Host eliotcam.ddns.net created. Update will be applied within 1 minute.' Below this is a blue bar: 'Current Hosts: 1' and 'Need More Hosts? Enhance Your Account!' with an 'Enhance Your Account' button. A table titled 'Hosts By Domain' is shown with columns 'Host', 'IP/URL', and 'Action'. The table contains one entry: 'eliotcam.ddns.net' with IP '186.4.133.171' and actions 'Modify' and 'Remove'. A red box highlights this row. At the bottom right is an 'Add A Host' button.

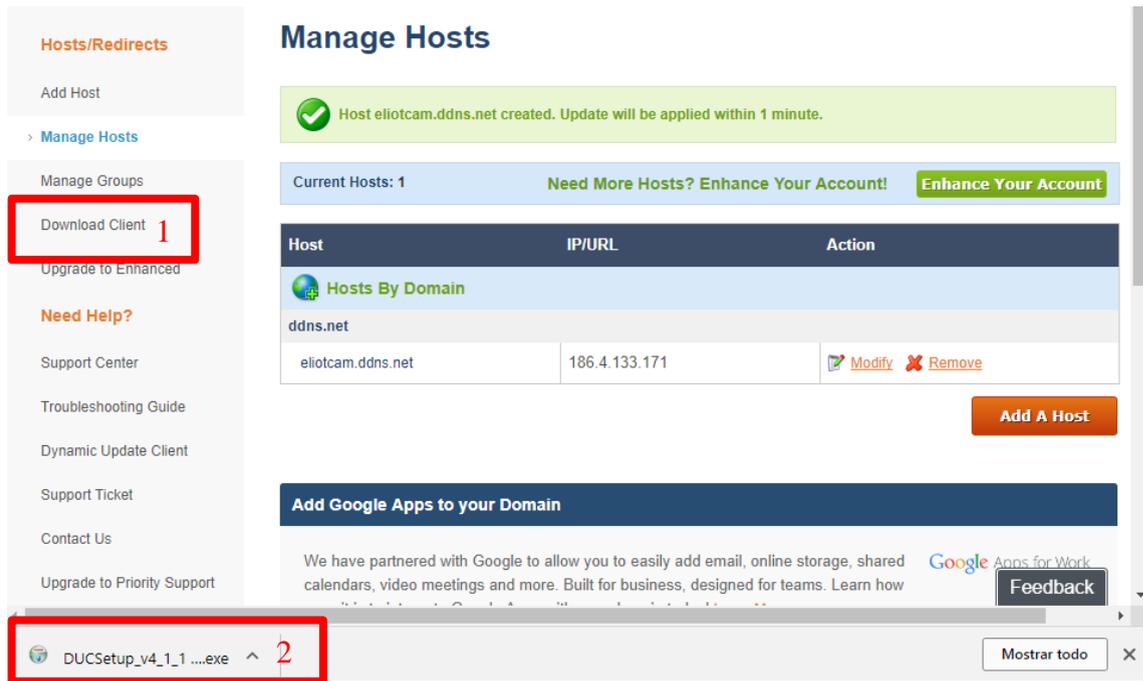
Host	IP/URL	Action
eliotcam.ddns.net	186.4.133.171	<a href="#">Modify</a> <a href="#">Remove</a>

Host Activo, NO-IP, Adrián López & Diego Zapata.

En la Figura A.24, se muestra el Host “eliotcam.ddns.net”, asociado a la dirección IP Pública de red a la cual se encuentra enlazada la planta (Cliente), que en este caso es 186.4.133.171.

El siguiente paso es dirigirse a la opción Download Client, como se detalla en la Figura A.25, y se descargará el software Dynamic DNS Update Client (DUC), la instalación de este software es muy sencilla únicamente se debe aceptar los términos y condiciones del programa, posteriormente se ejecuta el Software y un proceso en segundo plano, el cual permite mantener actualizada la Dirección IP Pública de la red al nombre de dominio creado en la cuenta de NO-IP.

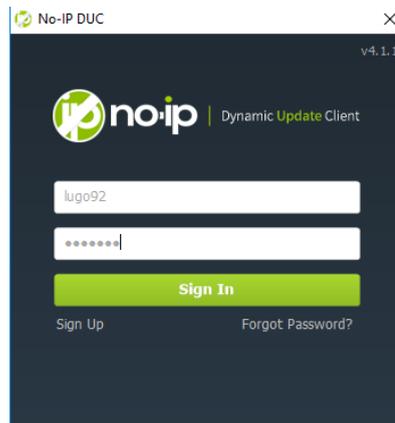
Figura A.25: Descarga Software Dynamic DNS Update Client



Descarga Software Dynamic DNS Update Client, 1) Download Client, 2) Software DUC descargado, NO-IP, Adrián López & Diego Zapata.

Al iniciar el Software DUC, pedirá que inicie sesión con la cuenta de NO-IP, como se muestra en la Figura A.26.

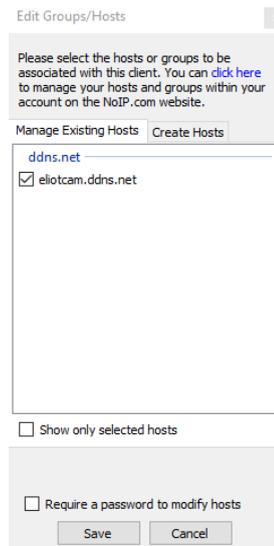
Figura A.26: Inicio de sesión Dynamic DNS Update Client



Inicio de sesión Software DUC, NO-IP, Adrián López & Diego Zapata.

Al iniciarse el software se despliegan dos ventanas, la primera indica los Host asociados a la cuenta de NO-IP, en la cual se debe seleccionar los host que se desean mantener activos en la red y presionar el botón "Save" como se indica en la Figura A.27.

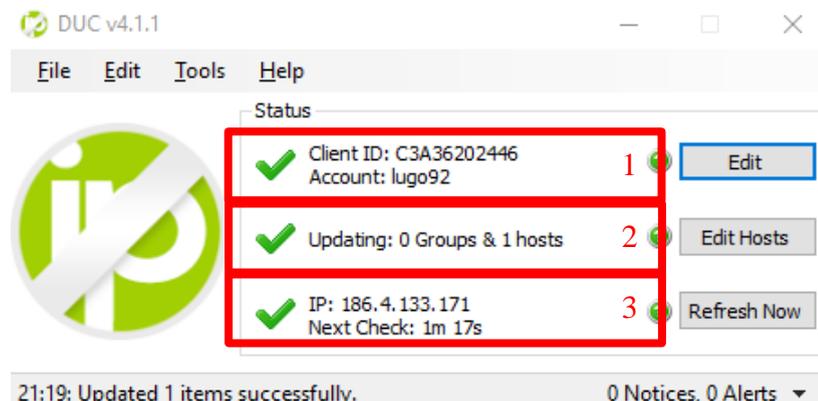
Figura A.27: Software DUC, activación de host en la red



Software DUC, activación de host en la red, NO-IP, Adrián López & Diego Zapata.

Una vez que se haya seleccionado los Host que estarán activos en la red, se visualiza la segunda ventana del software DUC, la cual muestra el estado de los Host de la cuenta NO-IP, la información que detalla es la cuenta NO-IP asociada al servicio, cuantos hosts se encuentran activos y la dirección IP Pública actual de la red, tal como se muestra en la Figura A.28.

Figura A.28: Estado de host en la red



Estado de host en la red, 1) Cuenta NO-IP, 2) Numero de Host activos, 3) dirección IP Pública actual de la red, NO-IP, Adrián López & Diego Zapata.

Este servicio se utilizó para realizar el monitoreo de la planta por medio de una cámara WEB, adaptada a cámara IP, la cual tomara una dirección IP privada dentro de la red. Para que el video pueda ser observado en el controlador WAN, se debe acceder a la cámara mediante la dirección IP pública y la asignación de un puerto, el router es el encargado de direccionar los paquetes a cada dispositivo.

## Configuración de puertos de comunicación para la utilización de la cámara IP.

El fabricante del Router de la red es Huawei Figura A.29, y el proveedor del Servicio de Internet es Netlife.

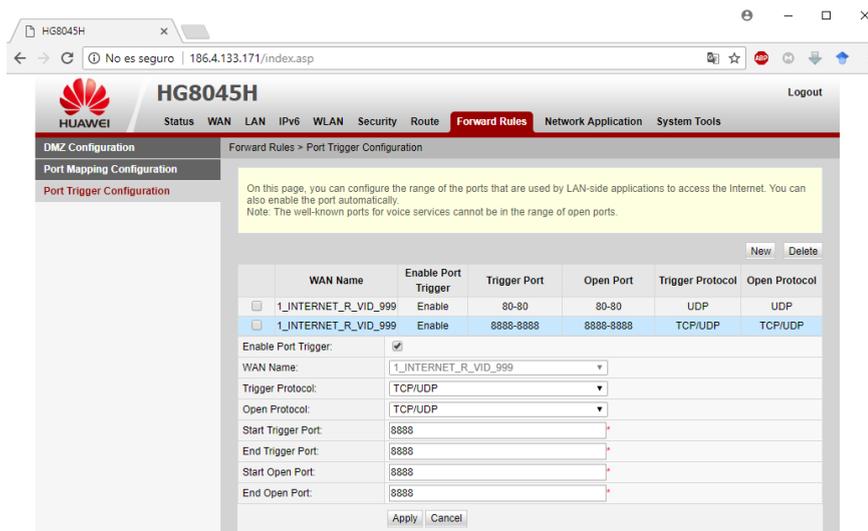
Figura A.29: Router de la red



Router de la red, Adrián López & Diego Zapata.

Una vez que se tiene acceso al Router, se selecciona la pestaña “Forward Rules”, la cual presenta 3 pestañas en el lado izquierdo que son “Port Trigger ConFfiguration”, “Port Mapping ConFfiguration” y “DMZ ConFfiguration”, lo cual se detallara en las Figuras A.30 – A.32.

Figura A.30: Configuración “Port Trigger ConFfiguration”

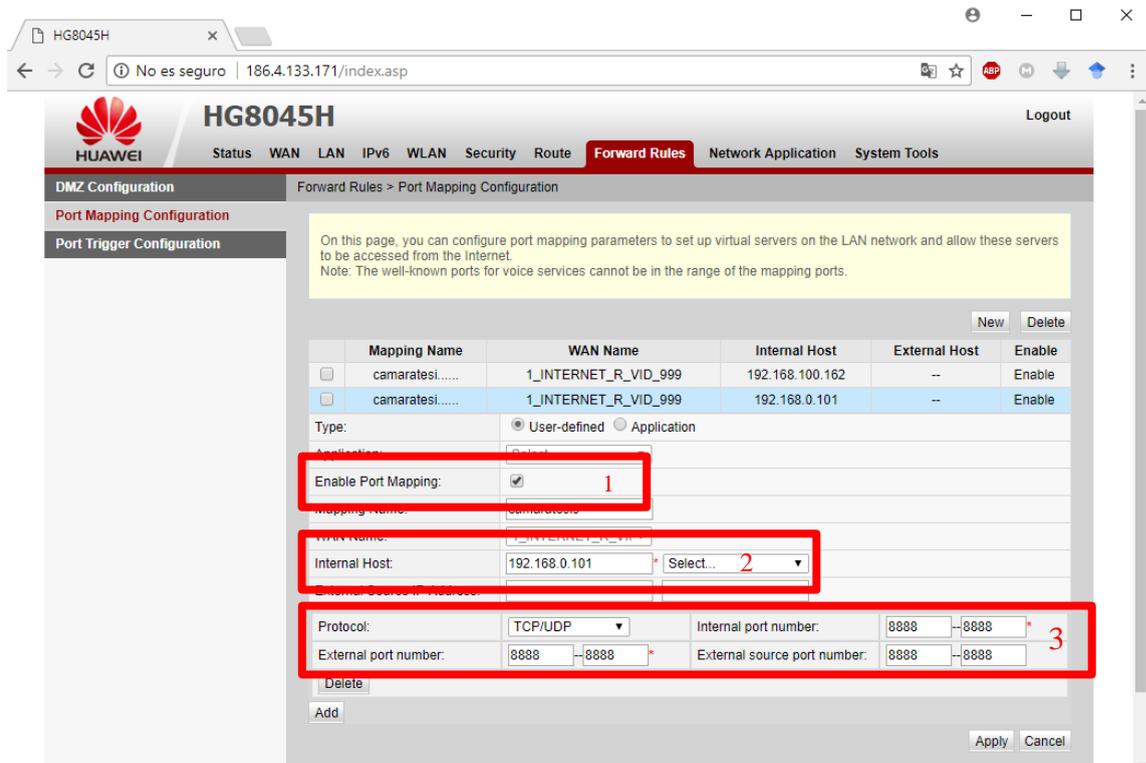


Configuración “Port Trigger ConFfiguration” en el Router de la red, Adrián López & Diego Zapata.

En la pestaña que se configura en la Figura A.30, se establece los puertos por los cuales una aplicación o un dispositivo accede a Internet, en este caso el puerto que se

configura será 8888, el mismo que se estableció en el software WEBcamXP 5 en la Figura A.14. En la Figura A.31 se realiza el mapeo de puertos, en otras palabras se ingresa la dirección IP el dispositivo que se comunicara por el puerto establecido.

Figura A.31: Configuración “Port Mapping ConFiguration”

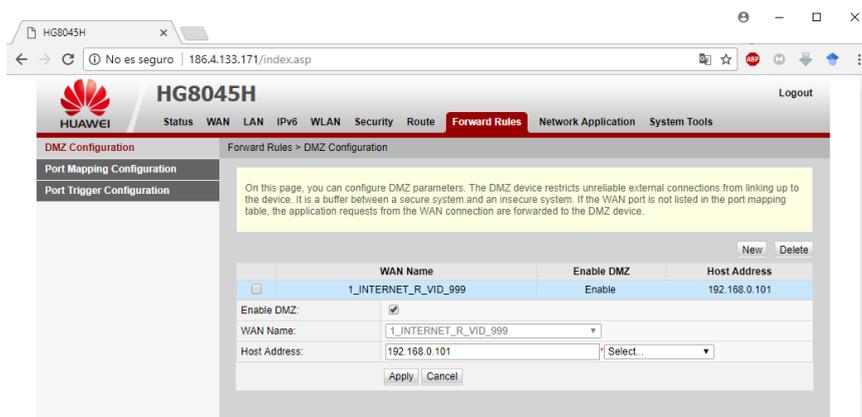


Configuración “Port Mapping ConFiguration”, en el Router de la red, 1) Habilitación mapeo de puerto, 2) Dirección IP del dispositivo, 3) Número de puerto establecido previamente para la comunicación, Adrián López & Diego Zapata.

El siguiente paso es crear un mapeo del Puerto que se estableció para que una aplicación o dispositivo acceda al internet, que en este caso será la cámara WEB, en la cual se habilita el mapeo del puerto, se define un nombre para el mapeo, se selecciona la dirección del dispositivo (en este caso del ordenador al que está conectada la Cámara WEB), y el puerto que seleccionado en la Figura A.30.

El último paso es habilitar el parámetro “DMZ” De-Militarized Zone o Zona Desmilitarizada, al dispositivo que accede a Internet, para que no exista ningún impedimento o futuros problemas de puertos al acceder al video de la cámara WEB. Lo cual se logra ingresando la dirección del dispositivo y dar click en “Apply”. Como se muestra en la Figura A.32. Cabe destacar que este parámetro se puede otorgar únicamente a un dispositivo en la red.

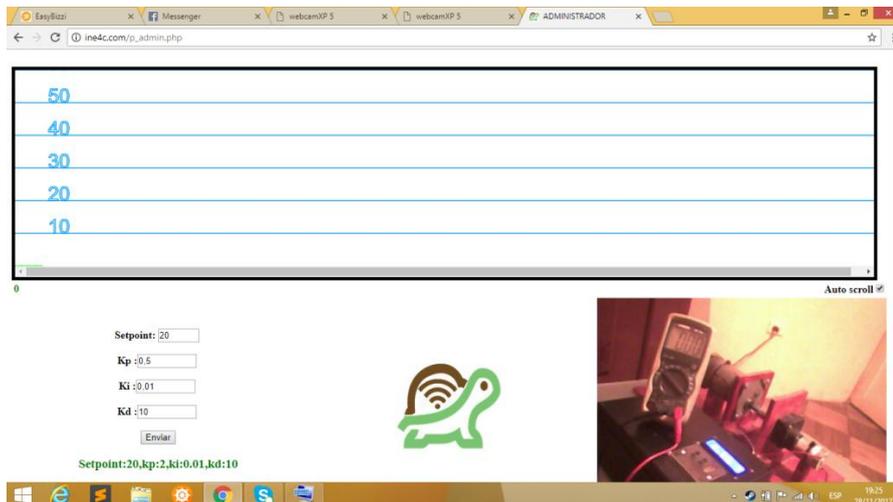
Figura A.32: Configuración “DMZ ConFiguration”



Configuración “DMZ ConFiguration” en el Router de la red, Adrián López & Diego Zapata.

El resultado final del servidor WAN se muestra en la Figura A.33.

Figura A.33: Página Servidor – Controlador WAN



Página Servidor – Controlador WAN, Adrián López & Diego Zapata.