

Universidad de Oriente.  
Nucleó Anzoátegui.  
Escuela de Ingeniería y Ciencias Aplicadas.  
Departamento de Ingeniería.  
Arquitectura Avanzada del Computador.



Desarrollar un Sistema Basado en un Microcontrolador Aplicando  
Metodologías para los Sistemas Empotrados.

Tutor Profesor Ing. Alfonso Alfonsí

Grupo N° 1.

Integrantes:

María Cardoza C.I: 6.142.718

Luis Correa C.I: 19.840.230

Copyright © 2015 por Luis Correa & María Cardoza. Todos los derechos reservados.

Barcelona, 03 de Agosto de 2015

## Tabla de Contenidos

<b>I. Introducción .....</b>	<b>3</b>
<b>II. Subsistema de Hardware .....</b>	<b>4</b>
<b>III. Subsistema de Software.....</b>	<b>6</b>
A. Descripción de ejercicio .....	6
B. Metodologías y Herramientas.....	7
C. Control.....	9
D. Simulación en Software.....	9
E. Algoritmo Main .....	10
<b>IV. Desarrollo del Sistema de Control .....</b>	<b>12</b>
A. Requisitos Funcionales.....	12
B. Requisitos No Funcionales .....	12
C. Arquitectura del Sistema: Arquitectura Lógica .....	13
D. Arquitectura del Sistema: Arquitectura Física.....	13
E. Pruebas .....	13
<b>V. Diagrama de casos de uso .....</b>	<b>13</b>
<b>VI. Discusión de los resultados.....</b>	<b>15</b>
<b>VII. Conclusiones.....</b>	<b>16</b>
<b>VIII. Lista de Referencias .....</b>	<b>17</b>
<b>IX. Glosario de Términos .....</b>	<b>18</b>

## **I. Introducción**

En el desarrollo de los sistemas electrónicos, es evidente el impacto que representan los microcontroladores en áreas como la instrumentación, control, automatización industrial, robótica, domótica e incluso en dispositivos de la vida diaria (sensores que detectan movimientos y encienden bombillos). Además existe una orientación estratégica hacia el desarrollo de software para la codificación y control de los microcontroladores. Como ejemplo de ello se tiene el software Arduino IDE, que tiene aplicaciones en las áreas antes mencionadas.

Un sistema de control basado en microcontroladores puede atender uno o varios dispositivos externos y sus respectivas funcionalidades, como los circuitos eléctricos que funcionan como una planta. Ante esta situación una de las vías para el desarrollo de un sistema de control discreto empotrado, es utilizar lazos independientes. Se construirán los circuitos especificados en las Figuras N° 3 y 4, que muestran como realizar un circuito eléctrico como una planta económica.

**Este trabajo está estructurado de la siguiente manera:**

- Subsistema Hardware.
- Subsistema Software.
- Desarrollo del Sistema de Control.

## II. Subsistema de Hardware

Como plataforma de hardware para el desarrollo de la investigación se seleccionó Arduino Uno, tomando como referencia los intereses actuales en cuanto al Desarrollo e Implementación de Hardware Libre en Venezuela. El cómo software para el control del Arduino se usó el Arduino IDE.

Arduino Uno es la placa Arduino más popular. Su cerebro puede estar conformado por el microcontrolador Atmel AVR ATmega328P. En la Figura N° 1 se muestra la disposición de los pines del microcontrolador. Entre otras características el Arduino Uno posee una conexión USB, un conector para la alimentación, una cabecera ICSP (In-Circuit Serial Programming), y un botón de reset. Es un sistema de desarrollo sencillo y de bajo costo que permite la realización de múltiples diseños, en la tabla 1 se muestran las características técnicas del ATmega328P.

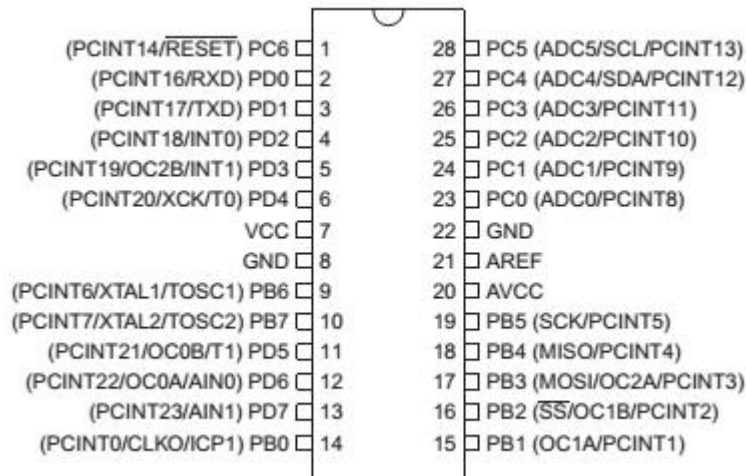


Figura N° 1. Arduino Uno ATmega328P

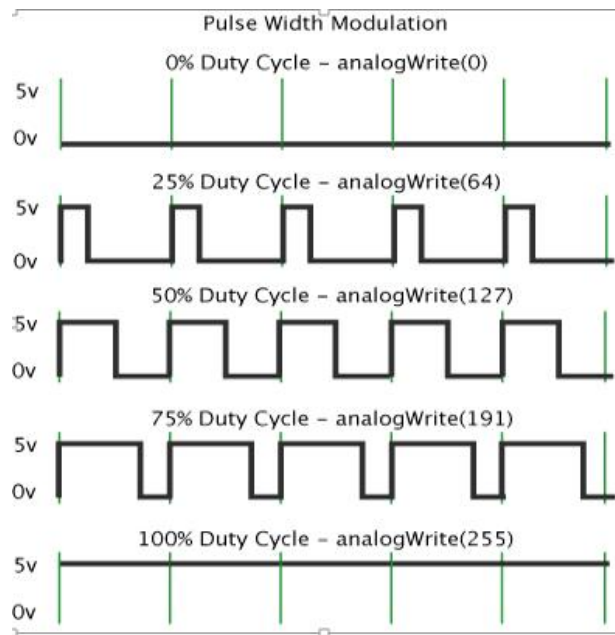


Logotipo de Arduino

**Tabla N° 1. Características técnicas  
de los microcontroladores ATmega328P**

PARÁMETROS	VALORES
Flash	32 Kbytes
RAM	2 Kbytes
Cantidad Pines	28
Frecuencia máxima de operación	20 MHz
CPU	8-bit AVR
Número de Canales variables	16
Pines máximos de E/S	26
Interrupciones externas	24

Como plataforma de hardware para el desarrollo de la investigación se seleccionó Arduino Uno, basado en el microcontrolador Atmel AVR ATmega328P, tomando como referencia los intereses actuales en cuanto al Desarrollo e Implementación de Hardware Libre en Venezuela. Arduino es una plataforma de Hardware Libre basada en una sencilla placa con entradas y salidas (E/S), analógicas y digitales, y en un entorno de Desarrollo que Implementa el lenguaje Processing/Wiring. El software de Arduino, Arduino IDE (Integrated Development Environment), se ejecuta en sistemas operativos Windows, OS X, Linux. Arduino Uno es la placa Arduino más popular. Su cerebro puede estar conformado por el microcontrolador Atmel AVR ATmega328P.



**Figura N° 2. Modulación Pulse Width Modulation (PWM).**

Tenemos las diferentes pulsos que puede enviar el Arduino Ide dependiendo de los valores que se escriban por los pines analógicos.

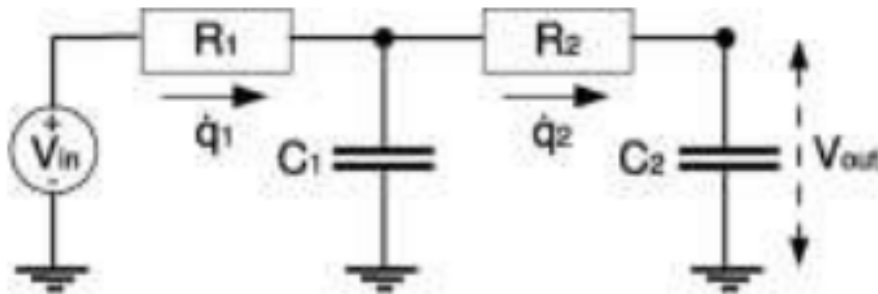
### III. Subsistema de Software

#### A. Descripción de Circuito para Simular una Planta

Dado el circuito RCRC mostrado en la fig. N°3, que representa la planta, cuyos valores de sus componentes son resistencias (R) y capacitores (C). Se debe cumplir:

$$R_1 = R_2 = 1\text{K}\Omega$$

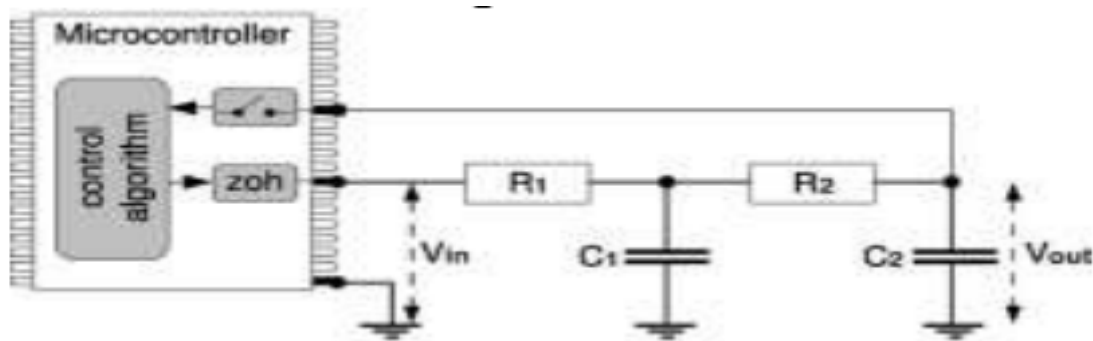
$$C_1 = C_2 = 33\text{ }\mu\text{F}$$



**Figura N° 3. Circuito Resistencia-Condensador y Resistencia-Condensador (RCRC).**

#### B. Descripción de Microcontrolador conectado al Circuito que simula una Planta

Desarrollar un algoritmo de control discreto empotrado en el sistema de desarrollo basado en el microcontrolador ( $\mu\text{C}$ ) ATmega328, usando para tal fin la metodología mikro\_STR con Extensión UML para el desarrollo de sistemas empotrados. El sistema empotrado debe visualizarse como en la fig. 4.



**Figura N° 4. Diagrama del Sistema Empotrado de Control Aimplementado.**

La selección del circuito eléctrico como una planta, Fig. N° 4, tiene importantes ventajas: dependen del circuito específico, pueden ser conectados directamente al  $\mu C$ , sin agregar otros componentes electrónicos. Muestra de ello se ilustra en la Fig. N° 4, donde el mantenedor de orden cero (zoh, zero-order hold) representa el actuador, y el conmutador, ubicado arriba al zoh, el muestreo. Es bueno mencionar que los niveles de señales Transistor-Transistor Logic (TTL) proporcionados por el  $\mu C$ , puede ser suficientes para llevar a cabo el control. Esta simplificación en términos de hardware reduce el esfuerzo de modelado para estudiar la planta y no se requieren modelos de actuadores o sensores. Los beneficios adicionales de este tipo de plantas son que los sistemas se pueden construir con facilidad, son baratos, tienen un peso ligero y pueden ser transportados con facilidad.

### C. Metodologías y Herramientas

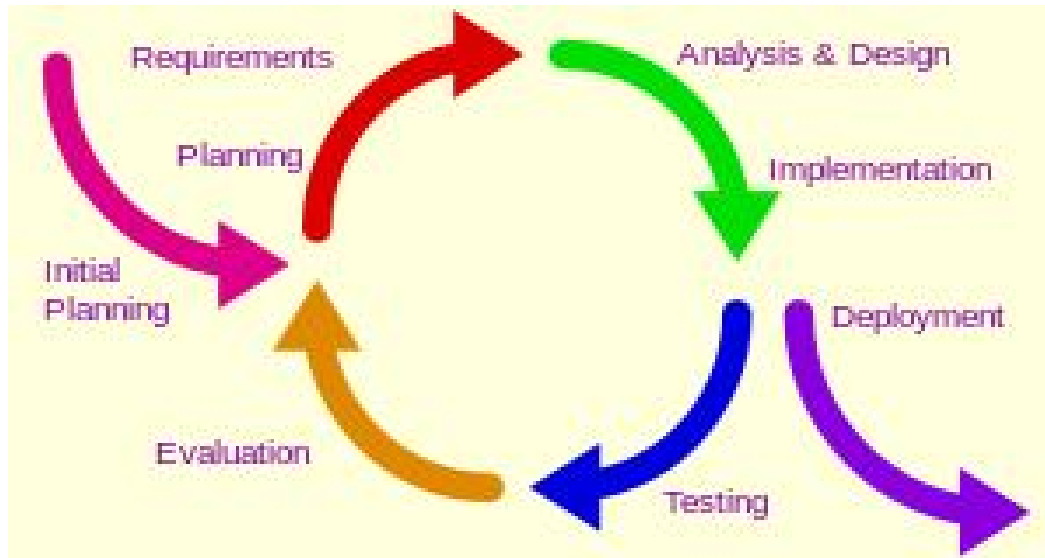
Desarrollo de una Aplicación para la creación de una planta económica para una empresa ubicada en la zona norte del estado Anzoátegui. Según la aplicación de metodologías para el desarrollo de un Sistema Empotrado (SE), dependerá del tipo de sistema y de los objetivos que se persiguen. En un sistema embebido estos procedimientos deben cumplir con las condicionantes en la implementación de este tipo de sistemas. La estructura del sistema se divide en dos partes fundamentales:

- Los requerimientos lógicos, los cuales están caracterizados por el Modelado del Comportamiento de Datos, Funciones y Control entre el Sistema y el ambiente.
- Los requerimientos temporales, que permiten predecir el comportamiento y garantizar la ejecución del sistema en el peor caso cuando se aplica un algoritmo.

Los procedimientos que se deben cumplir con las condicionantes en la implementación (ver figura 5) son:

- Planificación inicial.

- Planificación y Requerimiento.
- Análisis y Diseño, Implementación → Despliegue.
- Prueba.
- Evaluación → Iteración.



**Figura N° 5. Proceso de Desarrollo Iterativo.**

Entre las herramientas presentadas para cumplir con los requerimientos temporales del sistema se encuentran las pruebas y los temporizadores e interrupciones propias del microcontrolador, siendo estas dos últimas muy importantes, ya que como es bien sabido, la forma de trabajo de los microcontroladores es secuencial y la única manera que el planificador pueda en todo momento estar monitorizando y controlando la ejecución de las tareas, es a través de estas herramientas. Las herramientas computacionales utilizadas fueron el Arduino IDE, donde se realizó el código que fue almacenado en el microcontrolador, para la simulación de cada una de las etapas del sistema se utilizó ISIS Proteus, en su versión 7.2.

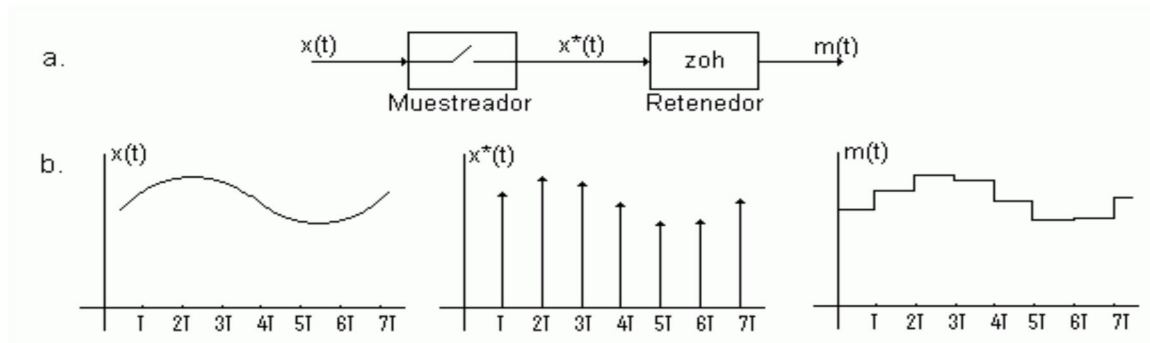
#### D. Explicación General del Funcionamiento del SE

El muestreador es el elemento fundamental en un sistema de control de tiempo discreto. Consiste simplemente en un interruptor que se cierra cada  $T$  segundos para admitir una señal de entrada. La función del muestreador es convertir una señal continua en el tiempo (análoga) en un tren de pulsos en los instantes de muestreo  $0, T, 2T \dots$  en donde  $T$  es el periodo de muestreo.

En la práctica, la señal en forma muestreada no se debe aplicar directamente a la planta por lo tanto es necesario incluir, después del muestreador, un dispositivo que reconstruya la señal. Este dispositivo se conoce con el nombre de retenedor y su finalidad es convertir la señal muestreada en una señal continua de tal forma que sea igual o lo más aproximada posible a la señal aplicada al muestreador. El retenedor más elemental convierte la señal muestreada en una señal que es constante entre dos instantes de muestreo consecutivos, este tipo de retenedor se conoce como “retenedor de orden cero” y es comúnmente el más utilizado. La exactitud del retenedor de orden



cero en la reconstrucción de la señal depende de la magnitud del periodo de muestreo. La figura 6 muestra un diagrama en bloques del conjunto muestreador-retenedor y la figura 6 da las formas de la señal de entrada y de salida en cada uno de estos dispositivos.

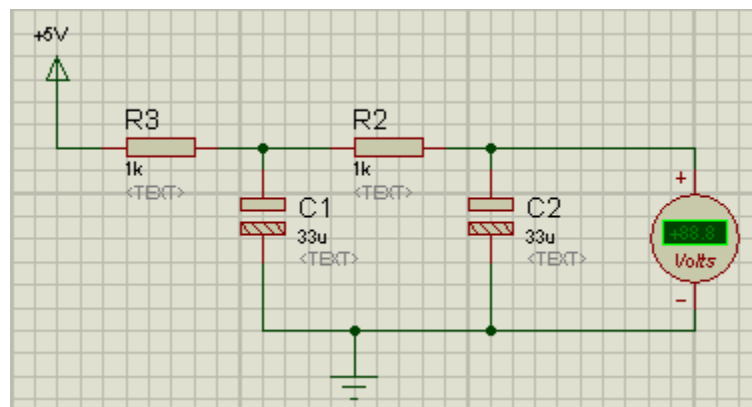


**Figura N° 6. Conjunto Muestreador-Retenedor B) Señales Entrada-Salida en el Muestreador y en el Retenedor**

#### E. Control

El Control es una herramienta que permite modificar el comportamiento del sistema, para que se apegue o lo que se desea.

#### F. Simulación en Software



**Figura N° 7. Simulación de circuito Proteus 7.2 de la figura N° 3.**

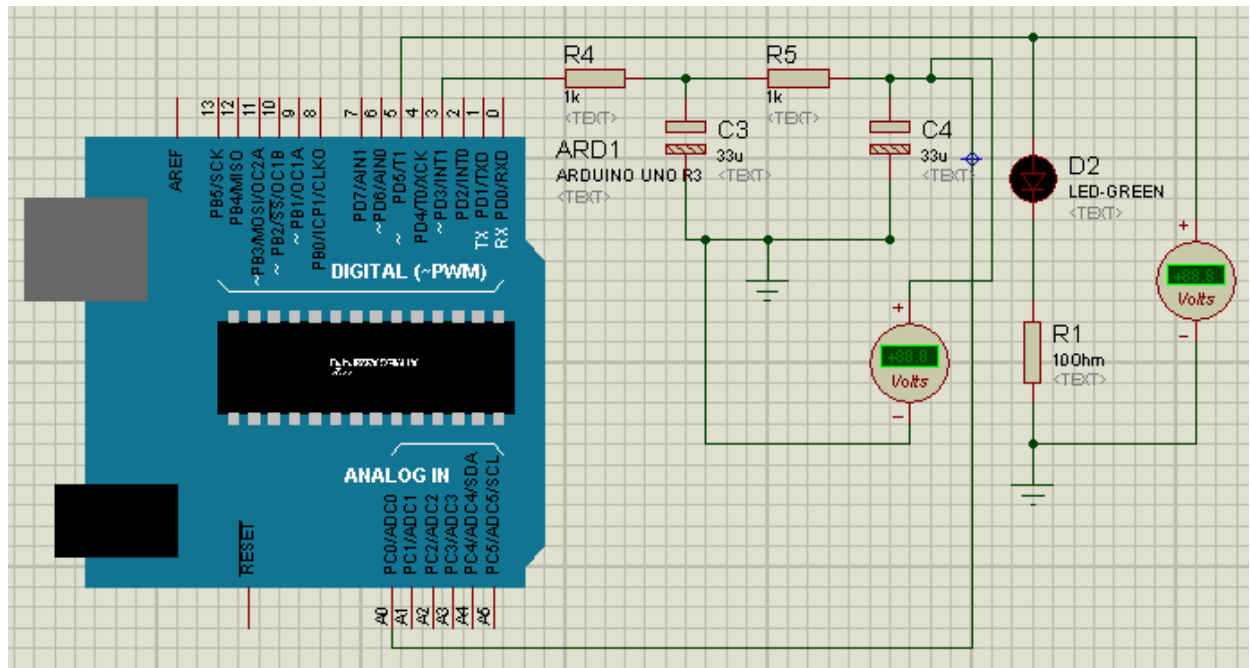


Figura N° 8. Simulación de circuito de la figura N° 4.

#### G. Algoritmo Main

Desarrollar un algoritmo de control discreto empotrado en el sistema de desarrollo basado en el microcontrolador ( $\mu C$ ) ATmega328, usando para tal fin la metodología mikro\_STR con Extensión UML para el desarrollo de sistemas empotrados.

El diseño de un algoritmo de control requiere de un modelo discreto del sistema a controlar, el cual es normalmente de naturaleza analógica.

- Por ello es necesario encontrar una representación discreta del sistema que pueda ser manejada por el sistema de procesamiento. Esta representación de señales discretas tiene la forma de ecuaciones de diferencias.
- Su análisis se facilita considerablemente por el método de transformada en Z, que es un método operacional aplicado a ecuaciones de diferencia, similar al de la transformada de Laplace.

#### Código Wiring de Arduino IDE.

```
const int led=3;
const int led2=5;
const int pot=0;
void setup(){
  pinMode(led,OUTPUT);
  pinMode(led2,OUTPUT);
}void loop(){
  for(int brillo=0;brillo<256;brillo++){
    analogWrite(led,brillo);
    int valor=analogRead(pot);
```

```

int brilloled = map(valor, 0, 1023, 0, 255);
analogWrite(led2, brilloled);
delay(10);
}
for(int brillo2=255;brillo2>=0;brillo2--){
  analogWrite(led, brillo2);
  int valor=analogRead(pot);
  int brilloled = map(valor, 0, 1023, 0, 255);
  analogWrite(led2, brilloled);
  delay(10);
}
}

```

### Código Zero-Order Hold

```

/*
brillo=map(temperatura,10,40,0,255);
brillo = constrain(brillo,0,255);
*/
const int MAX=500;
int x[MAX];
const int T=2;
int wc=2;
int n=0;
const int led=5;
const int pot=0;

void setup(){
  Serial.begin(9600);
  int cont=0;
  for(int brillo=0;brillo<256;brillo++){
    x[cont++]=brillo;
  }
  for(int brillo2=255;brillo2>=0;brillo2--){
    x[cont++]=brillo2;
  }
}

void loop(){
  n=0;
  for(int i=0;i<MAX;i++){
    zeroorderhold(i);
  }
}

void zeroorderhold(int t){
  int valor=x[n]*(T*wc/PI)*(sin(wc*(t-T*n))/wc*(t-T*n));

```

```

Serial.println(valor);
// valor = constrain(valor,0,255);
//analogWrite(led,valor);
// int muestra =analogRead(pot);
n++;
}

```

#### IV. Desarrollo del Sistema de Control

##### A. Requisitos Funcionales

- El sistema debe enviar señales que simularan el comportamiento de una planta
- El sistema podrá leer los datos generados por la planta.

##### B. Requisitos No Funcionales

- La operación sobre dispositivos es realizada por manejadores implementados como tareas.

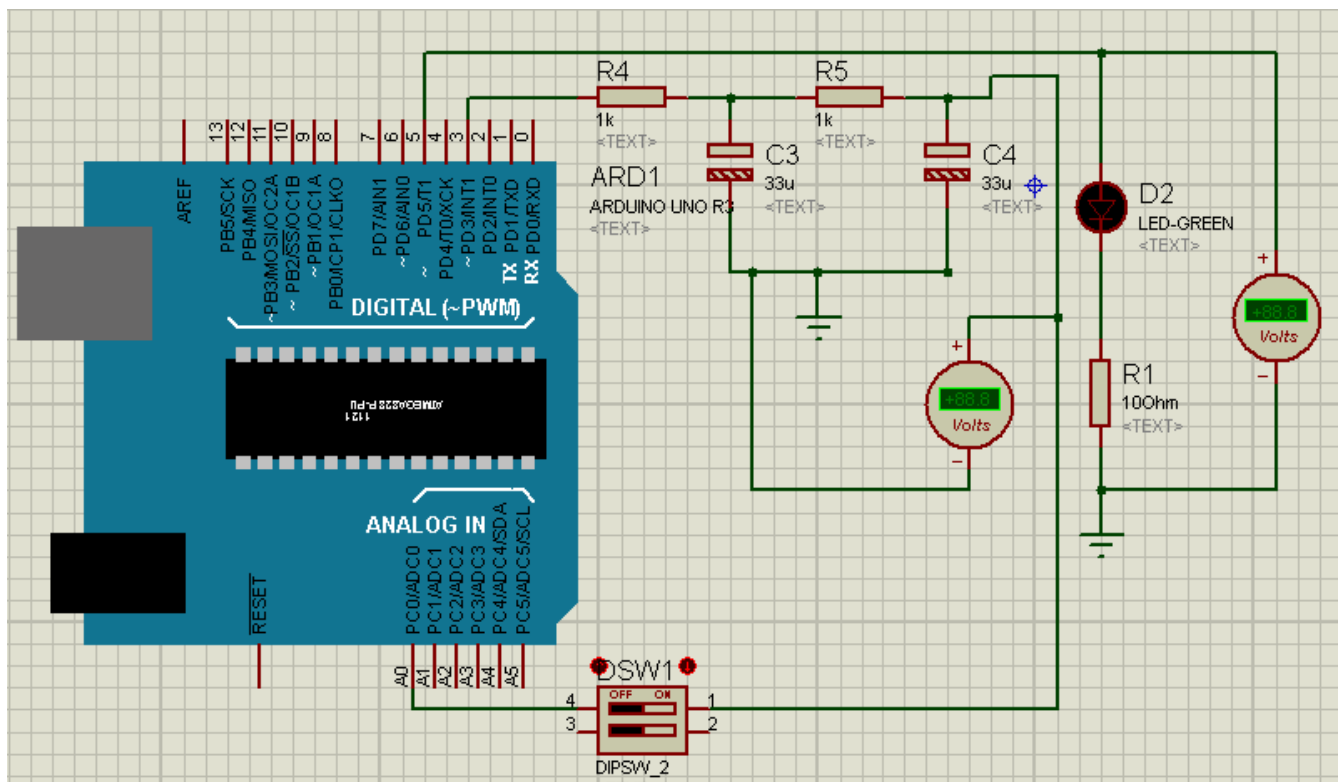


Figura N° 9. Diagrama Esquemático del Sistema de Simulación de una Planta.

### C. Arquitectura Lógica del Sistema

Para la elaboración de la estructura del software del Sistema que simula el comportamiento de una planta, fue necesario desarrollar la Arquitectura Lógica del Sistema en este caso siguiendo los lineamientos de las metodologías citadas en la sección tres.

### D. Arquitectura Física del Sistema

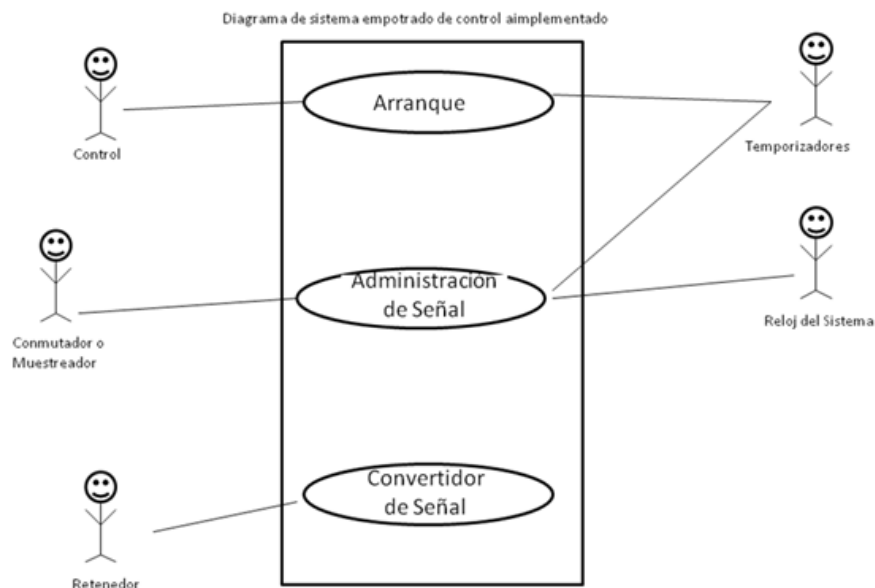
Como primer paso se asoció a un pin analógico la generación de señal para la planta y otro pin analógico para recibir la señal enviada por la planta. Se puede visualizar Figura número 9.

### E. Pruebas

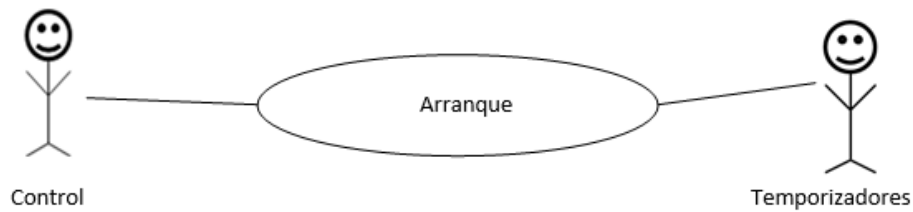
Se confeccionaron dos experiencias.

- La primera generando el hex en el Arduino IDE y luego realizando la simulación en el software ISIS Proteus.
- La segunda generando el hex en el Arduino IDE y luego cargando el programa en el Arduino Uno para poder comprobar físicamente el funcionamiento del código y del circuito.

## V. Diagrama de casos de uso



**Figura N° 10. Diagrama de Casos de Uso General del Sistema.**



**Figura N° 11. Diagrama de Casos de Uso Específico del Arranque.**

Caso N° 1: Arranque

<b>Nombre del caso de uso:</b>	Arranque
<b>Actores:</b>	Control y temporizadores
<b>Descripción:</b>	Permite el encendido y apagado del circuito.

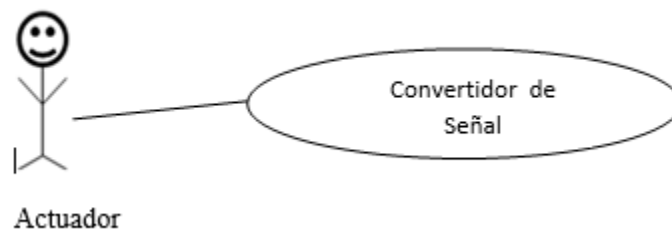


**Figura N° 12. Diagrama de Casos de Uso Específico de Administración de Señal.**

Caso N° 2: Administración de señal

<b>Nombre del caso de uso:</b>	Administración de señal
<b>Actores:</b>	Conmutador y Reloj del Sistema
<b>Descripción:</b>	Convierte una señal continua en el tiempo (análoga) en un tren de pulsos en los instantes de muestreo $0, T, 2T \dots$ en donde $T$ es el periodo de muestreo.

Actuador



**Figura N° 13. Diagrama de Casos de Uso Específico de Convertidor de señal.**

Caso N°3: Convertidor de señal

<b>Nombre del caso de uso:</b>	Convertidor de señal
<b>Actores:</b>	Actuador
<b>Descripción:</b>	Convierte la señal muestreada en una señal que es constante entre dos instantes de muestreo consecutivos, este tipo de retenedor se conoce como “retenedor de orden cero” y es comúnmente el más utilizado.

## VI. Discusión de los resultados

Se pueden apreciar los resultados del análisis digital realizado en el simulador, incorporando en el algoritmo de control. La respuesta del sistema para el circuito eléctrico como planta se muestra en la Figura N° 10. El controlador fue creado y cumplido en su totalidad.

De igual manera, la rutina de visualización en la pantalla del computador se llevó a cabo de acuerdo a lo previsto. Por tratarse de una tarea de visualización de datos, se pudo realizar un análisis profundo para corroborar lo propio al poder cargar el programa en el Arduino Uno y mostrar que funcionaba lo simulado.

Se pueden apreciar los resultados del análisis digital realizado en el simulador, incorporando el Planificador Main. La respuesta del sistema para los cuatro Hornos (análisis interactivo) se muestra en la Figura N° 9. El plan cíclico creado para el conjunto de tareas fue cumplido en su totalidad, respetando las restricciones temporales y el orden de activación de cada tarea.

En cuanto al análisis de la respuesta del sistema a lazo cerrado, las cuatro temperaturas lograron posicionarse en el valor deseado transcurrido cierto tiempo, se puede decir entonces que además de cumplir con las restricciones temporales, se lograron concretar todas las acciones de control del sistema.

De igual manera, la rutina de visualización en la pantalla del computador se llevó a cabo de acuerdo a lo previsto. Por tratarse de una tarea de visualización de datos, se pudo realizar un análisis profundo para corroborar lo propio al poder cargar el programa en el Arduino Uno y mostrar que funcionaba lo simulado.

## **VII. Conclusiones**

- Se desarrollo una aplicación para la generación de señales para una planta y que pueda leer los valores generados por la planta, lo cual permitirá un ahorro ya que la planta utilizada es una simulación a través de un hardware.
- Mediante la construcción del circuito se pudo cargar el programa en el Arduino Uno y realizar las conexiones necesarias para que el circuito funcionara correctamente.
- Mediante el circuito eléctrico de la Figura N°8 se comprobó el funcionamiento de una planta económica.
- Mediante el algoritmo de control discreto se demostró que se pueden leer y controlar los valores de voltaje de la planta.



## **VIII. Lista de Referencias**

- Alfonsí, A. (2013). Técnica Dinámica para Ajustar las Necesidades Energéticas de los Sistemas Empotrados de Control de Tiempo Real Autónomos. Trabajo de Ascenso Profesor Titular. Departamento de Computación y Sistemas. Universidad de Oriente, Núcleo Anzoátegui, Barcelona, Venezuela.
- James Rumbaugh, Ivar Jacobson, Grady Booch 2007. El Lenguaje de Modelado Unificado Ed 2. Madrid. Addison Wesley.

## **IX. Glosario de Términos**

- **Arduino Uno:** Es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios, basado en el microcontrolador Atmel AVR ATmega328P, implementa el lenguaje Processing/Wiring, con el cual se puede crear el circuito (hardware para detectar temperatura) y cargar el hex creado con el programa Arduino IDE.
- **Arduino IDE (Integrated Development Environment):** Entorno de Desarrollo Integrado, se ejecuta en sistemas operativos Windows, OS X, Linux, para programar con el lenguaje Processing/Wiring para las placas de Arduino.
- **Atmega328P:** Es un chip microcontrolador creado por Atmel y pertenece a la serie megaAVR.
- **ISIS Proteus:** Es una compilación de programas de diseño y simulación electrónica, desarrollado por Labcenter Electronics que consta de los dos programas principales: Ares e Isis, y los módulos VSM y Electra.