Universidad de Oriente.

Nucleó Anzoátegui.

Escuela de Ingeniería y Ciencias Aplicadas.

Departamento de Ingeniería.

Arquitectura Avanzada del Computador.



Desarrollo de una Aplicación para el Control de Temperatura en Hornos de una Empresa ubicada en la zona norte del Estado Anzoátegui.

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, 22 de Junio de 2015

**Tabla de Contenidos**

**I.** **Introducción** 3

**II.** **Objetivos** 4

A. Objetivo general 4

**III.** **Diagramas a emplearse** 4

A. Diagrama de Casos de Uso 4

B. Diagrama de Clases 6

C. Diagrama de Componentes 7

D. Diagrama de Secuencia 8

E. Diagrama de Estados 9

F. Diagrama de Tiempo 9

**IV.** **Conclusiones** 11

**V.** **Lista de referencias** 12

**VI.** **Glosario de términos** 13

# **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 prenden bombillos). Además se afirma que en estos tiempos existe una orientación estratégica hacia el desarrollo de software para la codificación y control de los microcontroladores. Un ejemplo de ello es el Arduino IDE, que es un software con aplicaciones en las áreas antes mencionadas.

Un sistema de control basado en microcontroladores puede atender uno o varios dispositivos externos y sus respectivas funcionalidades. Ante esta situación una de las vías para el desarrollo del control es utilizar lazos independientes. Se utilizará los Diagramas de Casos de Uso para mostrar el empleo de estos en el Sistema propuesto (ver Figura N°1) y el Diagrama de Clases se muestra en la Figura N°2. Las Figuras N°3, 4, 5 y 6 muestran los diferentes diagramas del sistema.

Este trabajo está estructurado de la siguiente manera:

* Objetivos
* Diagramas a emplearse

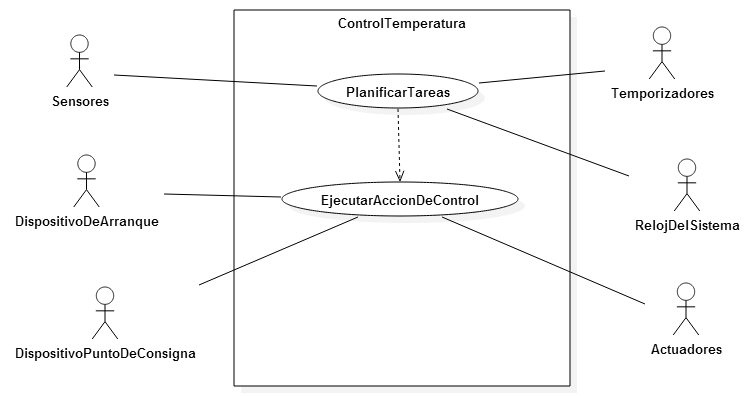
# **Objetivos**

### Objetivo general

Utilizar los Diagramas de Casos de Uso, Diagramas de Clase, Diagramas de Componentes, Diagramas de Secuencia, Diagramas de Maquina de Estados, en el estudio del Desarrollo de una Aplicación para el Control de Temperatura en Hornos de una Empresa ubicada en la zona norte del Estado Anzoátegui.

# **Diagramas a emplearse**

### Diagrama de Casos de Uso



**Figura N° 1. Diagrama de Casos de Uso.**

* **Planificar Tareas:**

Descripción:

Realiza la tarea de verificar la temperatura de los hornos a través de los sensores, por medio de los temporizadores y el reloj del sistema se puede controlar los tiempos de respuesta,.

Actores:

Sensores, Temporizadores, Reloj del Sistema.

Precondición:

Los Sensores deben estar conectados al Arduino Uno y tocar al Horno para poder revisar la temperatura y enviarla al Arduno Uno.

Flujo normal:

Los Sensores envían valores continuamente sobre las variaciones de temperatura.

Flujo alternativo:

Si el Sensor no está bien conectado o no toca al Horno no se obtendrán resultados correctos.

Poscondiciones:

Los sensores envían los valores obtenidos al Arduino Uno.

* **Ejecutar Acción de Control:**

Descripción:

Realiza la tarea de activar los actuadores, y de arrancar el Sistema.

Actores:

Dispositivo de Arranque, Actuadores, Dispositivo Punto de Consigna.

Precondición:

Los Actuadores, El dispositivo que inicia el Arranque del Sistema deben estar conectados al Arduino Uno.

Flujo normal:

El Arduino activara los actuadores cuando se detecte alguna temperatura no deseada.

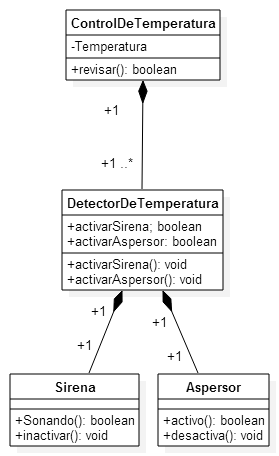
Flujo alternativo:

Si el Actuador no está bien conectado no se podrá bajar la temperatura cuando este muy alta, si el Arranque del Sistema no está bien conectado el Sistema jamás iniciara.

Poscondiciones:

Los Actuadores son activados por el Arduino Uno mientras baja la temperatura.

### Diagrama de Clases

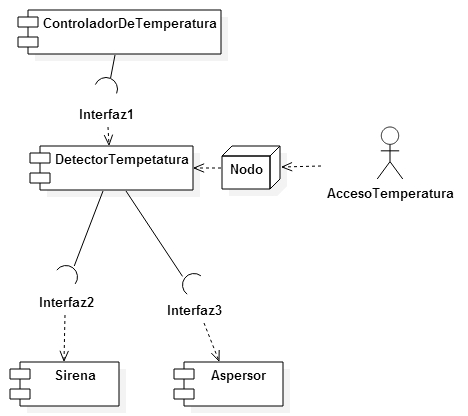


**Figura N°2. Diagrama de Clases**

La clase ControlDeTemperatura tiene que siempre tener una lista con al menos una clase DetectorDeTemperatura, también puede tener muchas.

La clase DetectorDeTemperatura tiene siempre que tener una Sirena y un Aspersor.

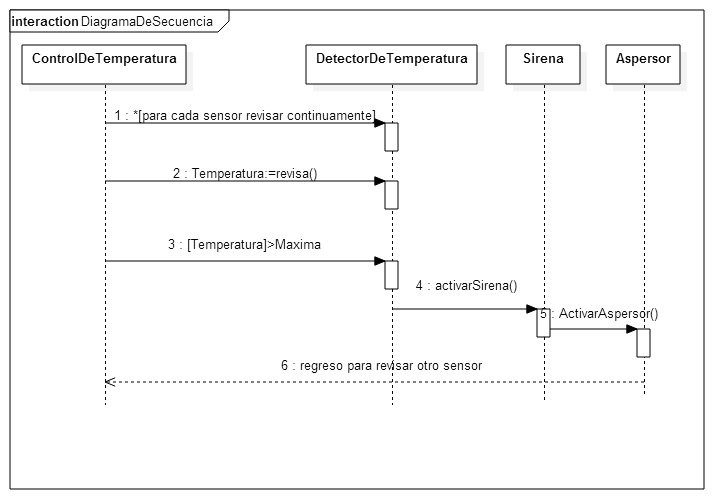
### Diagrama de Componentes



**Figura N°3. Diagrama de Componentes del Sistema**

En el Diagrama de visualiza que se tiene un acceso para la temperatura y las interfaces de conección de los componentes DetectorTemperatura, Sirena, Aspersor, ControlDeTemperatura.

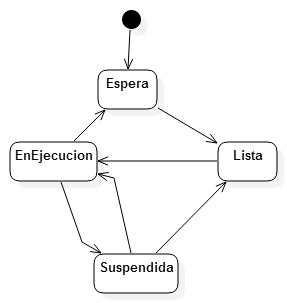
### Diagrama de Secuencia



**Figura N°4. Diagrama de Secuencia**

Diagrama de Secuencia para el control de Temperatura en hornos en el cual se hacen continuas revisiones por cada sensor de temperatura para detectar si se superó la temperatura máxima para activar la alarma y los aspersores, si no se supera la temperatura máxima en un sensor se revisa el siguiente y así susecivamente.

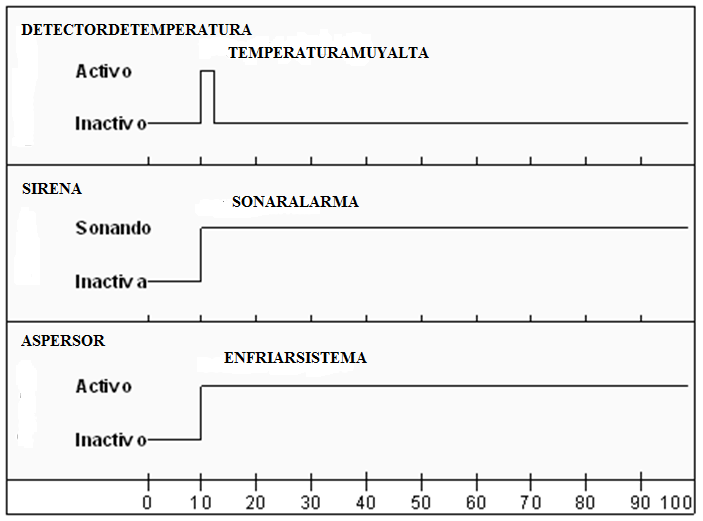
### Diagrama de Estados



**Figura N°5. Diagrama de Estado**

•Espera: estado inicial de la tarea. La tarea se encuentra a la espera hasta su activación, código 00.  
•Lista: estado indica que Ti terminado que cada vez que se produce una nueva activación, código 10. Es a partir  
de este momento, cuando el planificador toma en cuenta a la tarea en cuestión.  
•En ejecución: indica que Ti está haciendo uso del procesador en ese momento, código 01.  
•Suspendida: se utiliza para indicar que la tarea fue expulsada por el planificador. Tiene  
dos posibles transiciones, código 01 o 10.

### Diagrama de Tiempo



**Figura N°6. Diagrama de Tiempo**

# **Conclusiones**

* El Diagrama de Casos de Uso, en el presente trabajo es un tipo de Diagrama para modelar la interacción de los Actores (entidades externas u otros sistemas) con el sistema.
* Diagrama de Clases, en el presente trabajo es un tipo de Diagrama para modelar la estructura de las Clases y su relación con otras Clases en nuestro Sistema.
* Diagrama de Componentes, en el presente trabajo es un tipo de Diagrama para modelar como nuestro sistema de [software](https://es.wikipedia.org/wiki/Software) es dividido en [componentes](https://es.wikipedia.org/wiki/Componente_de_software) y muestra las [dependencias](https://es.wikipedia.org/w/index.php?title=Dependencia_%28UML%29&action=edit&redlink=1) entre estos componentes. Los componentes físicos incluyen [archivos](https://es.wikipedia.org/wiki/Archivo_%28computaci%C3%B3n%29), cabeceras, [bibliotecas compartidas](https://es.wikipedia.org/wiki/Biblioteca_%28programaci%C3%B3n%29), [módulos](https://es.wikipedia.org/wiki/M%C3%B3dulo_%28programaci%C3%B3n%29), [ejecutables](https://es.wikipedia.org/wiki/Ejecutable), o [paquetes](https://es.wikipedia.org/wiki/Paquete_de_software).
* El Diagrama de Secuencia, en el presente trabajo es un tipo de Diagrama para modelar la interacción entre objetos.
* Diagrama de Estado, en el presente trabajo nos permite mostrar los diferentes estados que puede adquirir una clase en nuestro Sistema.

# **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.

# **Glosario de términos**

* Arduino Uno: Es una plataforma de [hardware libre](https://es.wikipedia.org/wiki/Hardware_libre), basada en una [placa](https://es.wikipedia.org/wiki/Circuito_impreso) con un [microcontrolador](https://es.wikipedia.org/wiki/Microcontrolador) y un [entorno de desarrollo](https://es.wikipedia.org/wiki/Entorno_de_desarrollo), diseñada para facilitar el uso de la electrónica en proyectos multidisciplinares, 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 Develoment Enviromment): 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](https://es.wikipedia.org/w/index.php?title=Circuito_Integrado&action=edit&redlink=1) [microcontrolador](https://es.wikipedia.org/wiki/Microcontrolador) creado por [Atmel](https://es.wikipedia.org/wiki/Atmel) y pertenece a la serie [megaAVR](https://es.wikipedia.org/w/index.php?title=MegaAVR&action=edit&redlink=1).
* ISIS Proteus: Es una [compilación](https://es.wikipedia.org/wiki/Suite_ofim%C3%A1tica) de [programas](https://es.wikipedia.org/wiki/Software) de diseño y simulación [electrónica](https://es.wikipedia.org/wiki/Electr%C3%B3nica), desarrollado por [Labcenter Electronics](https://es.wikipedia.org/w/index.php?title=Labcenter_Electronics&action=edit&redlink=1) que consta de los dos programas principales: Ares e Isis, y los módulos VSM y Electra.