

**UNIVERSITAT  
JAUME I**

ESCUELA SUPERIOR DE TECNOLOGÍA Y CIENCIAS EXPERIMENTALES

UNIVERSIDAD JAUME I

**GRADO EN INGENIERÍA ELÉCTRICA**

---

**DESARROLLO DE UN SISTEMA DE CONTROL EN RED MEDIANTE  
EL ESTANDAR IEC-61499**

AUTOR: JOSÉ CARLOS MERCÉ GODOY

TUTOR: JULIO ARIEL ROMERO PÉREZ

CONVOCATORIA NOVIEMBRE DE 2015



*Quiero dedicar este proyecto a mis padres por haberme apoyado siempre en todos los objetivos que me he planteado a lo largo de mi vida y por la gran ayuda que siempre han estado y están dispuestos a brindarme. A mi hermana por estar siempre a mi lado incondicionalmente.*

*También quiero agradecer a Julio Ariel Romero el reto planteado con este proyecto que me ha permitido aumentar mis conocimientos y habilidades, así como la ayuda y formación que me ha ofrecido durante su desarrollo. A Pau Viciano por compartir su experiencia en las tareas de desarrollo de circuitos electrónicos. Y por último a Esteban Querol por la ayuda prestada en el uso de herramientas de software.*



## ÍNDICE

<b>1. MEMORIA</b> .....	3
<b>1.1 Objeto:</b> .....	3
<b>1.2 Alcance:</b> .....	3
<b>1.3 Antecedentes:</b> .....	3
1.3.1 Marco normativo y tecnológico.....	3
1.3.2 Estándar IEC-61499.....	4
1.3.2.1 <i>Objetivo</i> .....	4
1.3.2.2 <i>Estructura</i> .....	4
1.3.2.3 <i>Modelos</i> .....	5
1.3.2.4 <i>Herramientas de desarrollo de modelos</i> .....	9
1.3.2.5 <i>Plataformas de ejecución</i> .....	10
<b>1.4 Planteamiento del problema:</b> .....	11
<b>1.5 Solución adoptada:</b> .....	12
1.5.1 Selección de software.....	12
1.5.2 Selección de hardware.....	12
1.5.3 Desarrollo de capa interfaz de entradas analógicas y salidas EHRPWM.....	15
1.5.4 Sistema térmico con retardo y circuitos para sensor y actuador.....	26
1.5.4.1 <i>Amplificador ajustable para termopar</i> .....	27
1.5.4.2 <i>Amplificador de potencia para resistencias calefactoras</i> .....	30
1.5.5 Estructura física del sistema de control en red.....	31
1.5.6 Desarrollo e implementación del sistema de control distribuido.....	33
1.5.6.1 <i>Modelado de sistema, dispositivos y recursos</i> .....	33
1.5.6.2 <i>Modelado de aplicaciones</i> .....	34
1.5.6.3 <i>Mapeado de aplicaciones y configuración de comunicaciones</i> .....	37
1.5.6.4 <i>Implementación de los modelos</i> .....	40
1.5.7. Diseño de control PI y prueba de funcionamiento del sistema.....	44
<b>1.6 Conclusiones</b> .....	50
<b>1.7 Bibliografía</b> .....	51
<b>2. ANEXOS</b> .....	55
<b>2.1 Documentos de dispositivos desarrollados para el sistema</b> .....	55

2.1.1 Características e instrucciones de uso de capa interfaz de entradas analógicas y salidas EHRPWM. ....	55
2.1.2 Características e instrucciones de uso de sensor de temperatura analógico..	58
2.1.3 Características e instrucciones de uso de amplificador de potencia. ....	60
<b>2.2 Hojas de características de los componentes utilizados</b> .....	61
<b>3. PLANOS</b> .....	65
P01. Esquemático de capa interfaz de entradas analógicas y salidas EHRPWM. ....	66
P02. Esquemático de sensor de temperatura analógico.....	67
P03. Esquemático de amplificador de potencia. ....	68
P04. Diseño de PCB de capa interfaz de entradas analógicas y salidas EHRPWM. ....	69
P05. Diseño de PCB de sensor de temperatura analógico.....	70
P06. Diseño de PCB de amplificador de potencia. ....	71
P07. Detalle de PCB con componentes de capa interfaz de entradas analógicas y salidas EHRPWM. ....	72
P08. Detalle de PCB con componentes de sensor de temperatura analógico. ....	73
P09. Detalle de PCB con componentes de amplificador de potencia.....	74
<b>4. PRESUPUESTO</b> .....	77

# **1. MEMORIA**



# 1. MEMORIA

## 1.1 Objeto:

El presente proyecto tiene como objeto desarrollar un sistema de control distribuido según el estándar IEC-61499 con el que se puedan implementar diferentes algoritmos de control. Con ello se pretende probar el funcionamiento y comprobar las prestaciones que ofrecen este nuevo método de programación y las últimas versiones de software basadas en él disponibles actualmente.

Este estándar consiste en una arquitectura de referencia para el desarrollo de aplicaciones de control con lógica descentralizada. Está basada en bloques funcionales (FB) que se definen como la unidad estructural básica de los modelos. Una de las ventajas de usar este estándar es que permite el diseño centralizado de sistemas de control distribuidos.

## 1.2 Alcance:

El ámbito de aplicación de este proyecto es la investigación en laboratorio y uso académico en el área de ingeniería de sistemas y automática. En el caso que nos ocupa, se utiliza sobre un sistema de control térmico con retardo sobre el que se pretenden probar distintos algoritmos de control.

## 1.3 Antecedentes:

### 1.3.1 Marco normativo y tecnológico

Desde el lanzamiento en 1992 de la norma IEC-61131 se han conseguido estandarizar los lenguajes de programación en el ámbito de la automatización industrial. Esta norma ha contribuido a que los bloques de función (FB) estén cada día más presentes en este campo como elementos de programación, siendo su uso habitual entre ingenieros y demás profesionales del sector.

El uso de bloques funcionales permite la creación de programas más flexibles y reutilizables, ya que se entiende el bloque de función como una unidad de software en la que se encapsulan datos y código y que cuenta con entradas y salidas.

Con la creación de la norma IEC-61499 en el año 2005, por parte de la Comisión Electrotécnica Internacional (IEC), se consigue además la integración de protocolos de comunicación y modelos de información para facilitar la

implementación de sistemas de control distribuidos, proporcionando una metodología que permite obtener una aplicación y configuración de hardware independiente del proveedor.

### 1.3.2 Estándar IEC-61499

#### 1.3.2.1 Objetivo

El objetivo principal del estándar es conseguir un alto grado de control distribuido, pudiéndose utilizar sistemas embebidos existentes para descentralizar la lógica de control flexibilizando así los sistemas, y a la vez, normalizar el desarrollo de código de manera que este sea independiente de la máquina sobre la que se ejecuta.

Para ello el estándar trata los siguientes puntos principales:

- Portabilidad: El software debe soportar componentes y configuraciones creados por otras herramientas de desarrollo.
- Configurabilidad: Las herramientas de desarrollo de diferentes fabricantes deben ser capaces de configurar cualquier dispositivo y su software.
- Interoperabilidad: Los dispositivos pueden trabajar conjuntamente para llevar a cabo las funciones propias de las aplicaciones distribuidas.

#### 1.3.2.2 Estructura

El texto del estándar está estructurado en cuatro apartados:

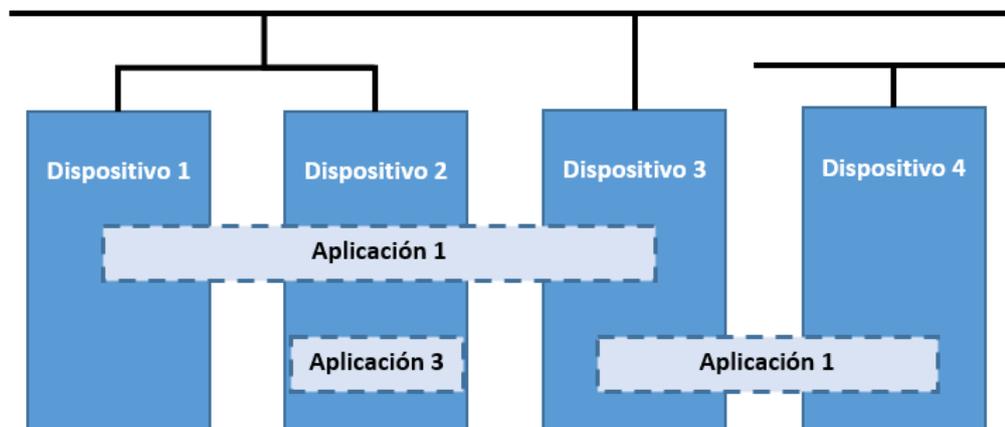
- Arquitectura: Se define los requisitos generales, definiciones y modelos de referencia. Reglas para la declaración de tipos de FB y reglas para su comportamiento.
- Requisitos de herramientas de software: Se definen los requisitos que deben cumplir las herramientas de software que ejecuten las arquitecturas definidas en el primer apartado.
- Manual informativo: Se describe, a través de ejemplos, la aplicabilidad de la arquitectura y de las herramientas de software que cumplen el estándar.
- Reglas y Perfiles de Conformidad: En este apartado se definen perfiles de compilación para los dispositivos compatibles y las herramientas de desarrollo.

### 1.3.2.3 Modelos

En este apartado se van a mostrar los principales modelos definidos en el primer apartado del estándar.

- Modelo de sistema:

Contiene a los demás modelos siendo el más alto en la estructura jerárquica. En él se encuentran los dispositivos y las diferentes aplicaciones que conforman el sistema de control, así como la relación existente entre estos de manera conforme a lo definido en el estándar.



*Figura 1. Modelo de sistema del estándar IEC-61499.*

Los dispositivos están comunicados mediante algún canal de comunicación de los permitidos y las aplicaciones se hayan distribuidas en ellos pudiendo encontrarse o no en dispositivos diferentes.

- Modelo de dispositivo:

Un dispositivo es capaz de albergar uno o varios recursos. Se entiende recurso como una unidad independiente de ejecución y control de redes de bloques de función.

Además proporciona una interfaz de proceso a los servicios que permite a los recursos intercambiar datos a través de las entradas y salidas físicas del dispositivo.

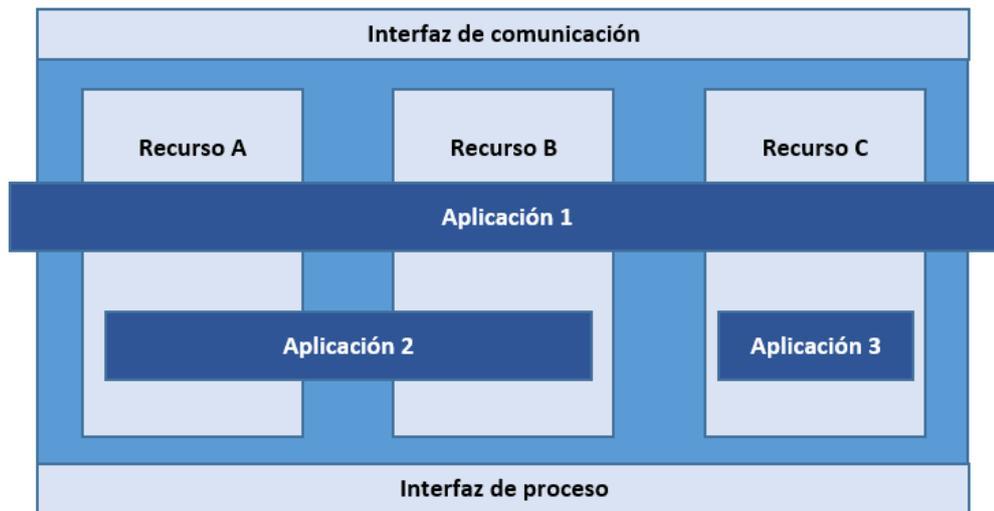


Figura 2. Modelo de dispositivo del estándar IEC-61499.

#### Modelo de recurso:

La finalidad del recurso es actuar de interfaz entre los sistemas de comunicación y los procesos específicos del dispositivo, como por ejemplo los subsistemas de entradas/salidas del dispositivo. Con ellos se logra el correcto funcionamiento del conjunto del sistema de control basado en aplicaciones cuyo comportamiento ha sido modelado independientemente de donde vaya a ser ejecutado.

Además los recursos son unidades que se ejecutan independientemente entre ellos aportando flexibilidad al sistema de control.

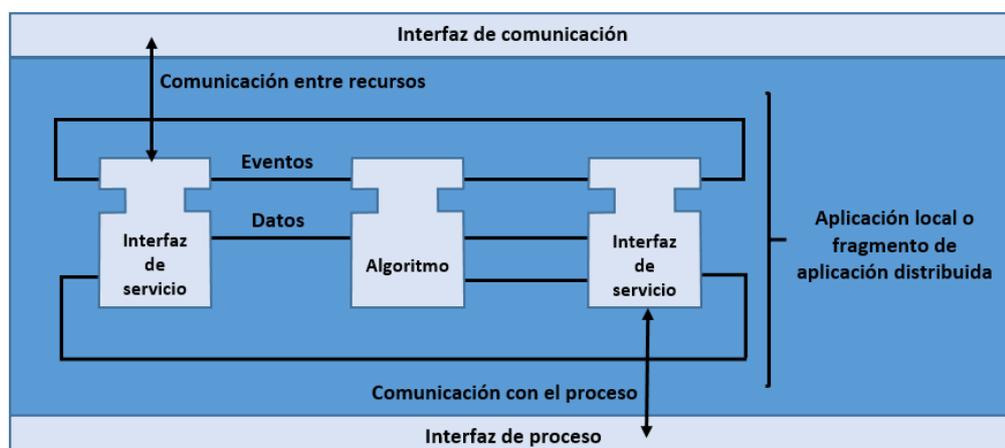
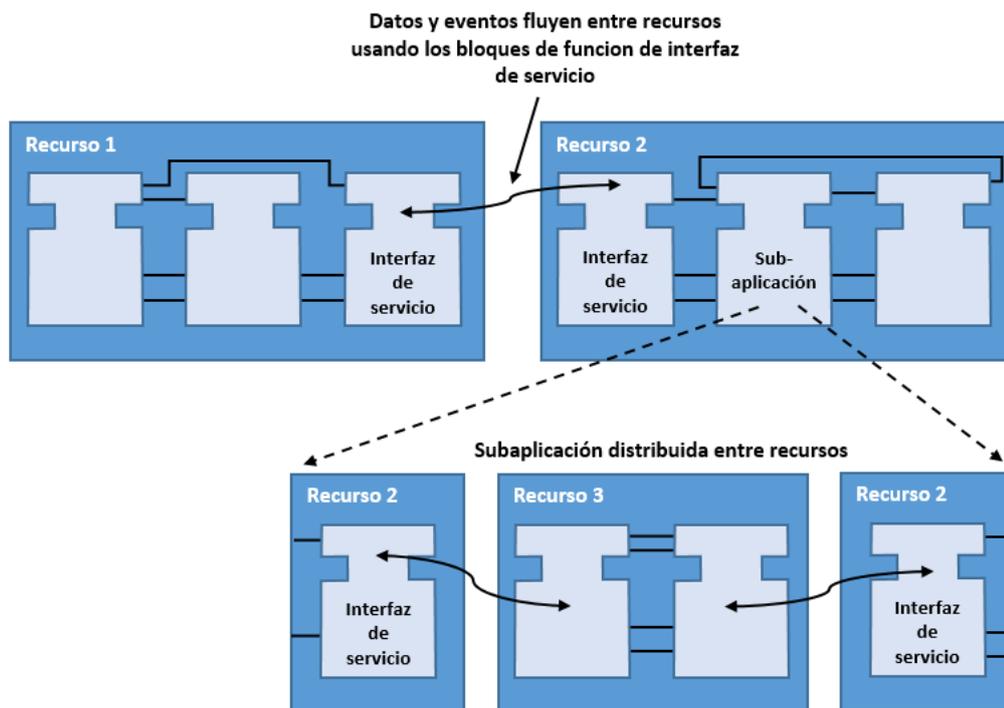


Figura 3. Modelo de recurso del estándar IEC-61499.

- Modelo de aplicación y subaplicación:

Una aplicación es una red de bloques de función conectados por flujos de eventos y datos que modela un comportamiento de un automatismo sin atender a donde se va a ejecutar. Posteriormente se distribuye esta aplicación entre los recursos disponibles para su correcta ejecución.

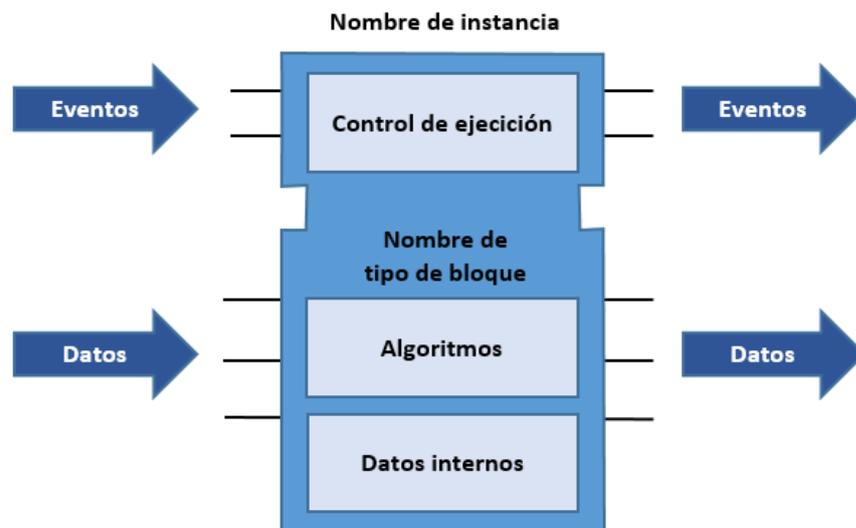
Además se pueden definir subaplicaciones, que son grupos de bloques de función interrelacionados que se comportan como un bloque de función, ya que tiene entradas y salidas de datos y eventos y además es replicable. En definitiva este modelo puede entenderse como una estructura fractal.



*Figura 4. Modelo de aplicación del estándar IEC-61499.*

- Modelo de bloque de función:

Un bloque de función es una unidad de código independiente con sus propios algoritmos y variables. Es la unidad de control básica del estándar IEC-61499, es decir, no es distribuible entre recursos diferentes.



*Figura 5. Modelo de bloque de función del estándar IEC-61499.*

En la figura se muestra la estructura de un bloque de función básico. Se observa que se divide en dos zonas.

En la superior se encuentran las entradas y salidas de eventos con los que se gestiona la ejecución del código. A la izquierda se encuentran los de entrada y a la derecha las de salida. El control de ejecución (ECC, Execution Control Chart) es el encargado de gestionar la ejecución del código del bloque. Su funcionamiento es similar al de una máquina de estados. En la parte inferior se encuentran las entradas y salidas de datos.

Puede tener cualquier número de entradas y salidas, pero como mínimo debe tener una salida que permita propagar la cadena de ejecución. Además el tipo de dato de cada una de ellas debe estar definido para evitar conflictos.

Internamente también dispone de conexiones "WITH", que son las que vinculan las señales de eventos con las de datos. Por ejemplo, en un bloque de función que realice una suma de dos entradas de datos, al obtenerse el resultado también se activaría una señal de evento confirmando que la operación ha sido realizada.

Los bloques de función básicos se pueden agrupar en un único encapsulado formando bloques de función compuestos. Su apariencia es como el básico con la diferencia de que no tienen un ECC propio sino que cada bloque que lo conforma tiene el suyo.

Por último comentaremos los bloques de función de servicio (SIFB, Service Interface Function Block). Son el vínculo entre los bloques de función o aplicaciones y el dispositivo en sí mismo. Su función es realizar los servicios necesarios para establecer la comunicación entre recursos de manera que las aplicaciones distribuidas entre estos funcionen correctamente. No forman parte de la aplicación, son propias del recurso y no se define exactamente en el estándar como deben ser ya que dependen del dispositivo en el que se usan.

#### *1.3.2.4 Herramientas de desarrollo de modelos*

En la parte dos del estándar se especifican las características que deben cumplir las herramientas de desarrollo para cumplir con el objetivo de la interoperabilidad. A continuación se muestran varias de estas herramientas con el fin de hacer una comparativa entre ellas.

- FBDK:

Functional Block Development Kit fué la primera herramienta de software desarrollada. Se trata de una aplicación Java que desarrollo uno de los creadores del estándar para representar gráficamente las aplicaciones de la norma, pero con el tiempo se ha desarrollado permitiendo actualmente testear los bloques e intercambiarlos en forma de XML.

Es una aplicación simple y poco robusta pero se sigue utilizando y distribuyéndose gratuitamente por la empresa Holobloc. Permite la portabilidad con 4DIAC\_IDE y NtxOne.

- 4DIAC-IDE:

Es una herramienta opensource basada en el entorno de desarrollo Eclipse. Además de ser robusta y sencilla de usar permite la creación de bloques de función básicos, sistemas, aplicaciones, etc. En sus últimas versiones incluye además las funciones de test y debug a través de las cuales se pueden visualizar y forzar las variables de forma remota durante la ejecución.

Es compatible con ntxOne y FBDK.

- nxtOne:

Esta herramienta permite programar según los estándares IEC-61131 e IEC-61499 y cuenta con la ventaja de disponer de un HMI con el que visualizar el proceso además de funciones de debugging y test.

Ha sido desarrollada por la compañía nxtControl y es compatible con todas las demás herramientas excepto con ISaGRAF.

- ISaGRAF:

Al igual que nxtOne, permite programar según los estándares IEC-61131 e IEC-61499 y tiene las funciones debugging y test.

Ha sido desarrollada por la compañía ICE Triplex perteneciente a Rockwell automation y tiene el inconveniente de no ser compatible con las demás herramientas de desarrollo.

#### *1.3.2.5 Plataformas de ejecución*

A continuación se describen brevemente varios entornos de ejecución disponibles para ejecutar los modelos desarrollados según la IEC-61499. La mayoría de ellos son compatibles con varias herramientas de desarrollo atendiendo al objetivo portabilidad del estándar.

- FBRT:

Es el primer entorno de ejecución que se desarrolló en Java embebido para probar los modelos desarrollados con FBDK. Al aparecer otros entornos de ejecución esta versión se dejó de desarrollar ya que al estar basada en Java dificulta la ejecución de sistemas en tiempo real, sin embargo la versión en Java normal se sigue actualizando ya que cuenta con un HMI siendo muy útil para visualizar el proceso si se utiliza algún entorno de desarrollo compatible que no cuente con esta función.

Las herramientas compatibles son FBDK, 4DIAC-IDE y nxtOne.

- FORTE:

Desarrollada por el mismo equipo que 4DIAC-IDE se trata de una plataforma de ejecución opensource que además sirve de base para otras versiones comerciales como es el caso de ntxRT61499F.

Permite la ejecución de sistemas en tiempo real en una amplia variedad de dispositivos, ya que se ha diseñado para ser independiente de la plataforma en la que se ejecute. La implementación se puede hacer desde 4DIAC, FBDK o nxtOne.

- nxtRT61499F:

Este entorno de ejecución pertenece a la compañía ntxControl. Está basado en FORTE pero incluye servicios y funciones adicionales. Se puede configurar con 4DIAC, ntxOne o FBDK.

- ISaGRAF Runtime:

Desarrollado por la compañía ICE Triplex este entorno de ejecución no usa el mismo perfil de compilación que los demás, de manera que solo es configurable desde la herramienta de desarrollo ISaGRAF.

Además en el modo de ejecución no cumple a rajatabla el estándar IEC-61499 basado en eventos, sino que funciona comprobando periódicamente el estado de las entradas.

#### **1.4 Planteamiento del problema:**

Se quiere diseñar un sistema de control en red que conste de varios nodos de ejecución y que permita la programación de algoritmos distribuidos de una forma sencilla y centralizada. Los sistemas que se desean controlar son procesos continuos de pocas variables.

El uso del estándar IEC-61499 nos brinda estas características de diseño. Por una parte la programación distribuida es sencilla gracias a los modelos de sistema, dispositivo, recurso y aplicación. Por otra parte el desarrollo de algoritmos es fácil ya que se basa en bloques funcionales.

Con este proyecto también se pretende probar las ventajas que aporta este tipo de programación y las posibilidades que ofrecen las herramientas de software actualmente existentes para el modelado e implementación de algoritmos según este estándar.

De estos requisitos de diseño surge la necesidad de encontrar los dispositivos adecuados para conformar esta red de controladores capaces de comunicarse mediante algún canal de comunicación y de gestionar señales de medida y de

control típicas en sistemas continuos, así como de seleccionar el software necesario y realizar las configuraciones necesarias de los dispositivos implicados.

### **1.5 Solución adoptada:**

Para el desarrollo del sistema de control en red basado en el estándar IEC-61499 se ha empezado por la selección de software a utilizar entre los mencionados en el apartado 1.3 de este proyecto.

#### 1.5.1 Selección de software

Como herramienta de desarrollo se ha optado por 4DIAC\_IDE, concretamente la versión 1.7.1. Esta aplicación, además de tener unas buenas prestaciones, es de código libre y compatible con la mayoría de las herramientas actualmente existentes, siendo este uno de los objetivos del estándar.

Como plataforma de ejecución para la implementación del sistema se ha elegido FORTE debido a la versatilidad que ofrece para elegir el hardware, ya que es independiente de la plataforma en la que se ejecute y funciona sobre varios sistemas operativos como Windows o Linux. También es compatible con FBDK y nxtOne y al igual que 4DIAC\_IDE también es de código libre.

Para el funcionamiento del HMI en el PC se usará el software FBRT, ya que se puede ejecutar directamente desde la perspectiva de implementación de 4DIAC\_IDE siendo cómodo su uso y se dispone de una librería de bloques de función diseñados para este propósito. No se trata de una aplicación muy sofisticada pero es suficiente para visualizar algunas variables del proceso durante la ejecución y también introducir algunos parámetros.

#### 1.5.2 Selección de hardware

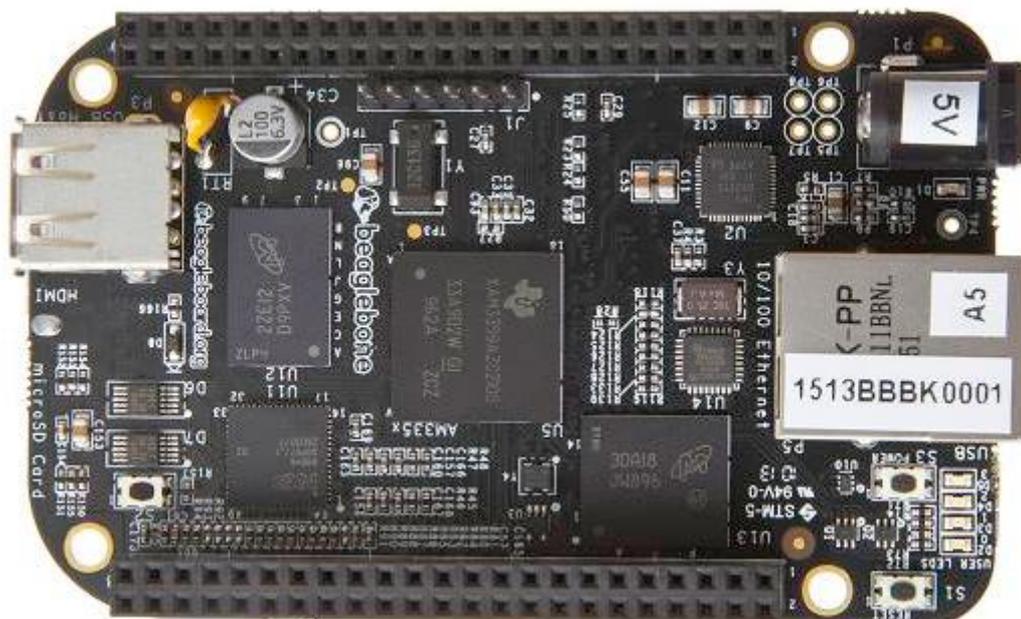
En la página web del desarrollador ([http://www.eclipse.org/4diac/en\\_com.php](http://www.eclipse.org/4diac/en_com.php)) se pueden encontrar diferentes sistemas embebidos válidos como soporte de ejecución de FORTE. Se pueden diferenciar dos grupos:

Boards	PLC
Digi Connect ME (ARM7)	Bachmann electronic M1 PLC
Lego Mindstorms nxt (ARM7)	WAGO PFC200
KIPR's CBC v2 robot controller	
RaspberryPI	
BeagleBone Black	

*Tabla 1. Sistemas embebidos recomendados por el desarrollador de 4DIAC\_IDE y FORTE.*

Puesto que se busca un dispositivo económico y no se requiere que soporte entornos industriales, directamente se descarta la opción de utilizar PLC para el sistema de control.

De entre las "boards" se opta por utilizar el mini-PC Beaglebone Black de Texas Instruments ya que es uno de los dispositivos más económicos y se adapta perfectamente a los requisitos de la aplicación desarrollada en el presente proyecto.



*Figura 6. Mini-PC Beaglebone Black.*

Beaglebone Black se trata de un dispositivo que lleva una CPU ARM Cortex-A8 a 1 GHz, 512 MB de RAM y 2 GB de almacenamiento interno, además de una ranura microSD. Como sistema operativo lleva por defecto Angstrom Linux, pero también se puede instalar Ubuntu o Android de manera sencilla.

En cuanto a puertos dispone de una conexión USB, una mini USB y otra Ethernet con la que poder conectar el dispositivo en red según permite el estándar. También cuenta con dos conectores de 46 pines cada uno entre los que se dispone de 65 entradas/salidas de propósito general (GPIO), 7 entradas analógicas, 8 salidas EHRPWM y varios pines para comunicaciones como I2C, SPI y CAN Bus.

Para la aplicación desarrollada son necesarias la conexión Ethernet, las entradas analógicas y las salidas EHRPWM. Las entradas/salidas digitales y el resto de comunicaciones no son tan interesantes ya que lo que se pretende es controlar sistemas continuos sencillos en el laboratorio mediante un sistema en red.

La elección de este dispositivo también añade nuevas condiciones de diseño al sistema de control, ya que estos dispositivos tienen limitaciones en cuanto a las tensiones que pueden manejar tanto en las entradas analógicas como en las salidas PWM. Las entradas analógicas admiten un rango de 0 a 1,8V y las señales PWM funcionan a 3,3V, por lo que se plantea la necesidad de usar una interfaz que permita la conexión de los equipos que conforman el sistema de control. Estos son sensores analógicos con rangos habituales de 0 a 5V y de 0 a 10V y actuadores que requieran señales PWM de 5V con un mínimo de potencia requerida.

Otros inconvenientes de estos dispositivos son limitaciones de uso que tienen como consecuencia la destrucción del procesador:

1. Superar el valor de 1,8V en las entradas analógicas.
2. Superar el valor de 3,3V en las entradas/salidas digitales.
3. Aplicar tensiones a cualquiera de los pines de entrada o salida mientras el dispositivo esté apagado o en proceso de arranque.

Como interfaz de conexión entre sensores, actuadores y el mini-PC para adaptar los niveles de tensiones se evalúa la posibilidad de usar una "cape" para Beaglebone Black. Las "capes", capas en español, son circuitos impresos diseñados para una aplicación específica y que permite una rápida conexión con la Beaglebone Black. Existe una comunidad en BeagleBoard.org en la que sus miembros desarrollan, comparten diseños y venden este tipo de productos.

Conseguir una capa que se adapte a las necesidades del proyecto no es la mejor opción por dos razones: 1) Los fabricantes son distintos para cada modelo y suelen ser particulares o empresas pequeñas. 2) No resulta barato costear el precio más los gastos de envío.

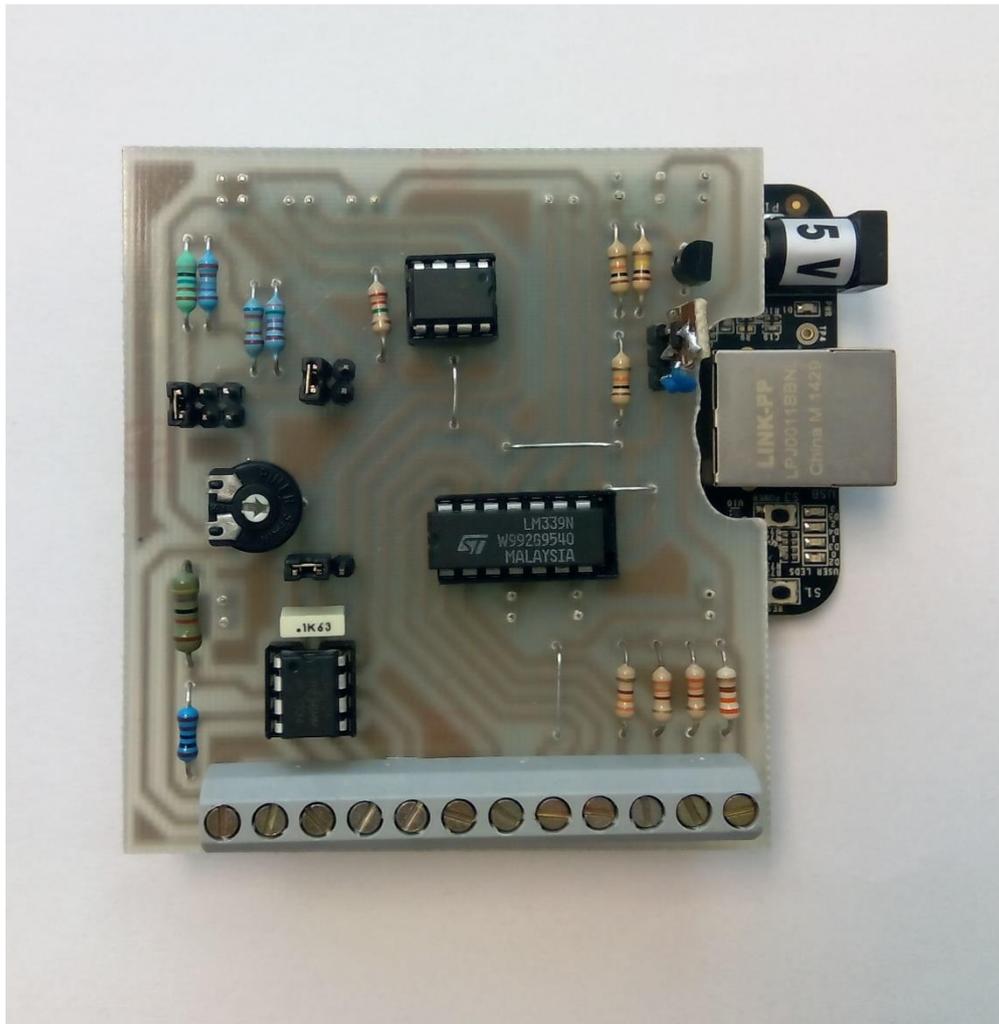
La mejor opción y más sencilla es diseñar y fabricar una capa con las características específicas para la aplicación a desarrollar.

### 1.5.3 Desarrollo de capa interfaz de entradas analógicas y salidas EHRPWM

La finalidad con la que se desarrolla la capa es poder usar el mini-PC Beaglebone Black como un nodo de control del sistema en red desarrollado en este proyecto.

Se quiere que permita el control de sistemas continuos por lo que se deben de poder conectarse sensores analógicos con rangos de medida de hasta 10V y actuadores que requieran señales PWM de 5V. También se pretende proteger el mini-PC frente a la situación de aplicar tensiones en sus pines mientras este está apagado, ya que esta situación es fácil que tenga lugar mientras se usan los diferentes equipos.

La alimentación de la capa se realiza desde la toma de 5V que posee la Beaglebone Black (VDD\_5V) disponible en los pines 5 y 6 del cabezal de expansión P9. Esta toma está conectada al jack de alimentación por lo que es capaz de suministrar hasta 1A de corriente mientras se use la fuente de alimentación adecuada.



*Figura 7. Capa interfaz de entradas analógicas y salidas EHRPWM.*

A continuación se describen uno a uno los diferentes circuitos que componen la capa:

a) Circuito de protección:

La Beaglebone Black dispone de una salida digital (SYS\_RESETn) para indicar cuando el sistema está en marcha. Esto es después de que se haya realizado el arranque del sistema. Hasta este momento no debe de haber tensiones en ninguno de sus pines, ya que esto dañaría el procesador. La toma de tensión VDD\_5V tiene tensión mientras la fuente de alimentación esté conectada aunque el mini-PC esté apagado, por lo que al alimentar la capa desde esta toma, puede haber tensiones en los pines con la Beaglebone Black apagada.

Para solucionar este problema se alimentan los amplificadores operacionales usados en las entradas analógicas y el comparador usado para las salidas EHRPWM desde un circuito que corta el suministro mientras la salida SYS\_RESETh esté en estado lógico 0. Al explicar los restantes circuitos se mostrará más detalladamente cómo se impide la existencia de tensiones al cortar el suministro de estos componentes.

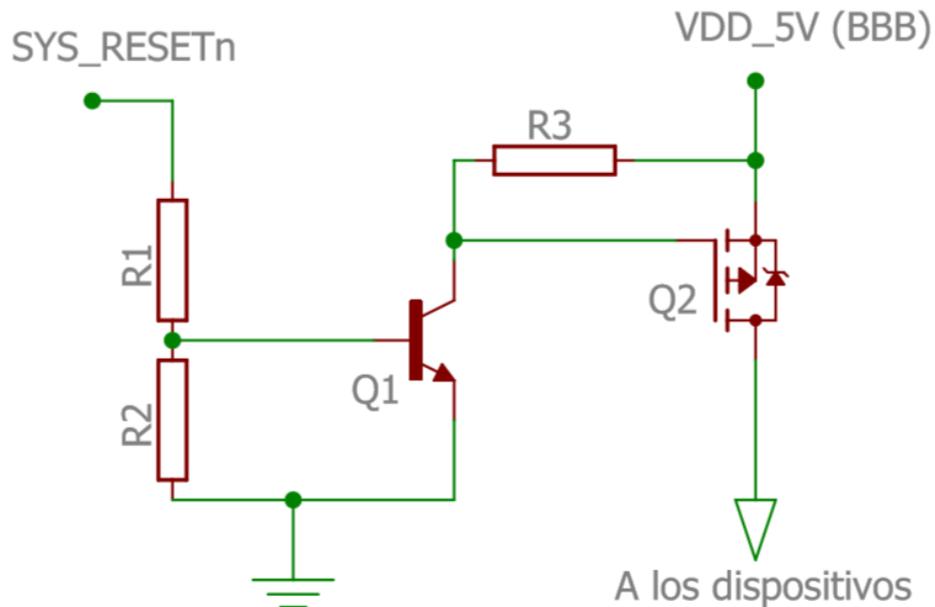


Figura 8. Circuito de protección de la capa.

El circuito, tal y como se muestra en la figura 8, consta de un transistor BJT con el que se polariza un MOSFET de canal P. Es necesario que este último sea de canal P, ya que se desea cortar el suministro de tensión en los componentes. Si se usase uno de canal N no se tendría esta garantía ya que este se montaría entre la carga y masa.

#### Cálculos y consideraciones:

Los componentes y sus valores son los siguientes:

- $R_1$        $10k\Omega$
- $R_2$        $100k\Omega$
- $R_3$        $10k\Omega$

- $Q_1$       *BC548C*
- $Q_2$       *RZR040P01*

La función de la resistencia  $R_2$  es simplemente fijar un valor lógico de 3,3V en la salida. Puesto que su valor es muy elevado en comparación con las otras resistencias no afecta a la polarización del BJT.

Según la hoja de características el valor de la tensión umbral del MOSFET está entre  $V_{GS(th),max} = -0,3V$  y  $V_{GS(th),min} = -1V$ , por lo que necesitamos una caída de tensión en  $R_3$  mayor de 1V.

En el caso más desfavorable en el estado lógico 1 de las salidas con tecnología LVCMOS se tienen 2,4V. Con esta tensión tenemos una intensidad de base en el BJT de:

$$I_B = \frac{(V_{LVCMOS.MIN} - V_{BE.BJT})}{R_1} = \frac{(2,4 - 0,7)V}{10k\Omega} = 0,17mA$$

Según la figura 3 del datasheet del BJT en la que se representa la curva  $V_{CE}(I_B)$ , para la intensidad de base calculada anteriormente se tiene una caída de tensión colector emisor  $V_{CE.ON} = 0,2V$  por lo que en la base del MOSFET conseguimos como mínimo una caída de tensión de:

$$V_{GS} = -(V_{DD\_5V} - V_{CE.ON}) = -(5 - 0,2) = -4,8V$$

Esta tensión está por debajo del umbral de corte del MOSFET, por lo tanto tenemos la garantía de que este está polarizado. Para este valor de tensión, según la figura 10 del datasheet solo se tiene una resistencia drenador fuente  $R_{DS.ON} = 25m\Omega$ .

Por otra parte, la corriente máxima en corte del BJT es  $I_{CE.OFF} = 4,0\mu A$ . Con esta corriente la caída de tensión en  $R_3$  cuando el BJT está en corte es de:

$$V_{GS.OFF.MAX} = I_{CE.OFF} \cdot R_3 = 4\mu A \cdot 10k\Omega = 0,04V$$

que está por encima de la tensión umbral del MOSFET, por lo tanto se tiene la garantía de que este también está en corte.

La salida digital SYS\_RESETn de la Beaglebone Black es capaz de suministrar 6mA. La corriente máxima suministrada se da con el valor máximo del estado lógico de tecnología LVCMOS de 3,3V. La corriente máxima es:

$$I_{SYS\_RESETn\_MAX} = \frac{(V_{LVCMOS\_MAX} - V_{BE})}{R_1} + \frac{V_{BE}}{R_2}$$

$$I_{SYS\_RESETn\_MAX} = \frac{(3,3 - 0,7)V}{10k\Omega} + \frac{0,7V}{100k\Omega} = 0,267mA$$

mucho menor de los 6mA que es capaz de suministrar.

Se hace una estimación de cuanta corriente debe suministrarse a través del MOSFET a los demás componentes de la capa.

$$I_{MOSFET\_MAX} = 2 \cdot I_{AMP\_OP\_MAX} + I_{COMP\_MAX} = 2 \cdot 1mA + 2mA = 4mA$$

Es una corriente mucho menor de la máxima que soporta el MOSFET.

Por último se considera la resistencia fuente drenador del MOSFET en saturación para evaluar la caída de tensión con el consumo máximo estimado anteriormente.

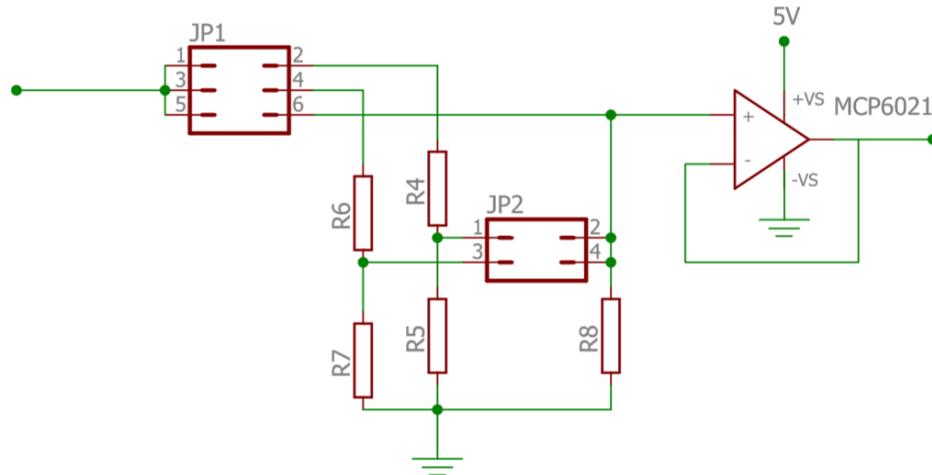
$$V_{DS\_MAX} = R_{DS\_ON} \cdot I_{MOSFET\_MAX} = 25m\Omega \cdot 4mA = 0,1mV$$

Es una caída de tensión despreciable frente a los 5V que se pretenden conmutar.

#### b) Circuito de adaptación de rangos de tensión para entrada analógica

El siguiente circuito tiene dos finalidades: 1) Adaptar el rango de tensión de medida de una entrada analógica de la Beaglebone Black que va de 0 a 1,8V a un rango de entrada seleccionable mediante jumpers de 5

o 10V. 2) Evitar que se aplique tensión a la entrada analógica mientras la Beaglebone Black esté apagada.



*Figura 9. Circuito de adaptación de rangos de tensión para entrada analógica.*

Tal y como se muestra en la figura el circuito consiste en dos divisores de tensión seleccionables mediante jumpers para poder conectar en la entrada sensores analógicos de 10V, de 5V o realizar la conexión sin divisor de tensión disponiendo del rango de 1,8V.

El amplificador operacional montado como seguidor de tensión está alimentado desde el circuito de protección explicado anteriormente. De esta manera el amplificador no está alimentado cuando la salida SYS\_RESETn está desactivada y resulta imposible que en la salida del operacional se tengan tensiones protegiendo así el procesador del mini-PC.

#### Cálculos y consideraciones:

Se calculan los divisores de tensión para que la suma de las dos resistencias esté entorno a los 10k $\Omega$  para que el consumo requerido al sensor sea pequeño. Las resistencias usadas tienen una tolerancia del 1% para realizar una conversión precisa.

Divisor de tensión para entrada analógica de 10V:

$$R_5 = 1,78\Omega$$

$$R_4 = R_5 \cdot \left[ \frac{V_i}{V_o} - 1 \right] = 1,78 \cdot \left[ \frac{10}{1,8} - 1 \right] = 8,1k\Omega \approx 8,06k\Omega$$

$$R_4 + R_5 = 8,06 + 1,78 = 9,84k\Omega \approx 10k\Omega$$

Divisor de tensión para entrada analógica de 5V:

$$R_7 = 3,57k\Omega$$

$$R_6 = R_7 \cdot \left[ \frac{V_i}{V_o} - 1 \right] = 3,57 \cdot \left[ \frac{5}{1,8} - 1 \right] = 6,34k\Omega$$

$$R_6 + R_7 = 3,57 + 6,34 = 9,91k\Omega \approx 10k\Omega$$

Si al cambiar los jumpers la entrada no inversora del amplificador operacional se quedara al aire, en la salida de este podrían aparecer hasta 3V, ya que no habría ninguna tensión de referencia que fije el valor de la salida. Esta tensión dañaría el procesador de la Beaglebone Black al sobrepasar el límite de 1,8V. Para solucionar este problema está la resistencia  $R_8$  de 2,2M $\Omega$ . Sirve para fijar una referencia de 0V en la entrada no inversora del operacional en el caso de que no haya ninguna otra, así en la salida nunca hay tensiones peligrosas. Al ser de un valor tan alto en comparación con las de los divisores el error que introduce es despreciable. Las resistencias equivalentes en cada caso son:

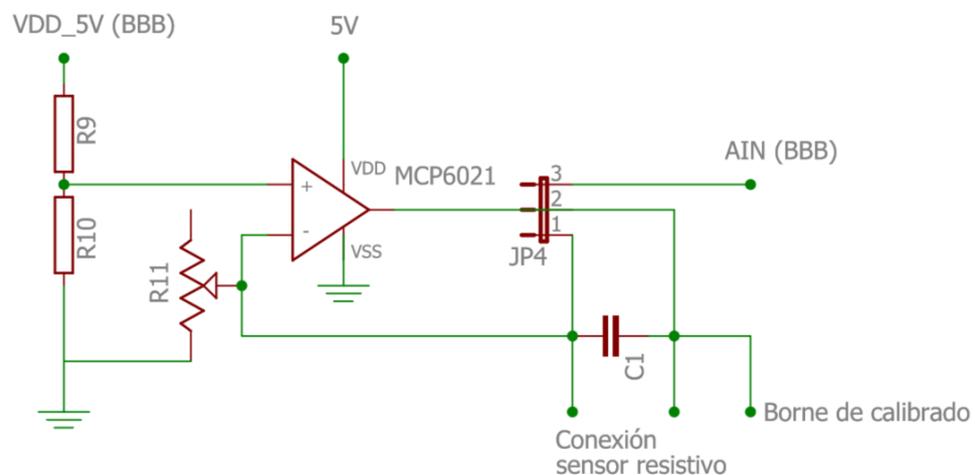
$$R_{5,8} = \frac{R_5 \cdot R_8}{R_5 + R_8} = \frac{1,78k\Omega \cdot 2,2M\Omega}{1,78k\Omega + 2,2M\Omega} = 1,778k\Omega \approx R_5$$

$$R_{7,8} = \frac{R_7 \cdot R_8}{R_7 + R_8} = \frac{3,57k\Omega \cdot 2,2M\Omega}{3,57k\Omega + 2,2M\Omega} = 3,56k\Omega \approx R_7$$

El amplificador operacional seleccionado es el MCP6021 de microchip. Se alimenta con tensión positiva de entre 2,5 y 5,5V, algo necesario ya que se alimenta desde la Beaglebone Black y no se disponen de tensiones negativas. Tiene un ancho de banda de 10MHz y es de bajo ruido y offset. Estas características lo hacen idóneo para esta aplicación, ya que no genera error de medida y se pueden medir señales con frecuencias muy elevadas. De esta manera no se limitan las características de la entrada analógica de la Beaglebone Black.

c) Circuito para conexión de sensores resistivos

Este circuito permite la conexión de sensores resistivos directamente a la capa convirtiendo esta variación de resistencia en variación de tensión para poder leer la medición en una entrada analógica de la Beaglebone Black.



*Figura 10. Circuito para conexión de sensores resistivos.*

La idea es disponer directamente desde la capa de unos bornes de conexión para un sensor resistivo y que en esta se integre la instrumentación necesaria, por lo que este circuito debe de ser ajustable para poder conectar un abanico más o menos amplio de sensores.

El diseño se ha realizado pensando en la conexión de termistores NTC y PTC y se trata básicamente de un amplificador no inversor en el que una de las resistencias es el propio sensor variando así la ganancia. La

tensión que se amplifica es una tensión constante definida por el divisor formado por  $R_9$  y  $R_{10}$  y alimentado directamente desde la toma de 5V del mini-PC (VDD\_5V). De esta manera se consigue una respuesta linealmente proporcional a la variación de la resistencia del sensor con un offset definido por la tensión de referencia tal y como se deduce de la función de transferencia:

$$V_o = V_{Ref} \left[ \frac{R_{sensor}}{R_{trimmer}} + 1 \right]$$

Se elige una tensión de referencia  $V_{Ref}=0,1V$  para minimizar el offset en la salida. Se calcula el divisor resistivo.

$$R_{10} = 1k\Omega$$

$$R_9 = R_{10} \cdot \left[ \frac{V_{DD\_5V}}{V_{Ref}} - 1 \right] = 3,57 \cdot \left[ \frac{5}{1,8} - 1 \right] = 49k\Omega \approx 47k\Omega$$

Las resistencias seleccionadas tienen una tolerancia del 1%.

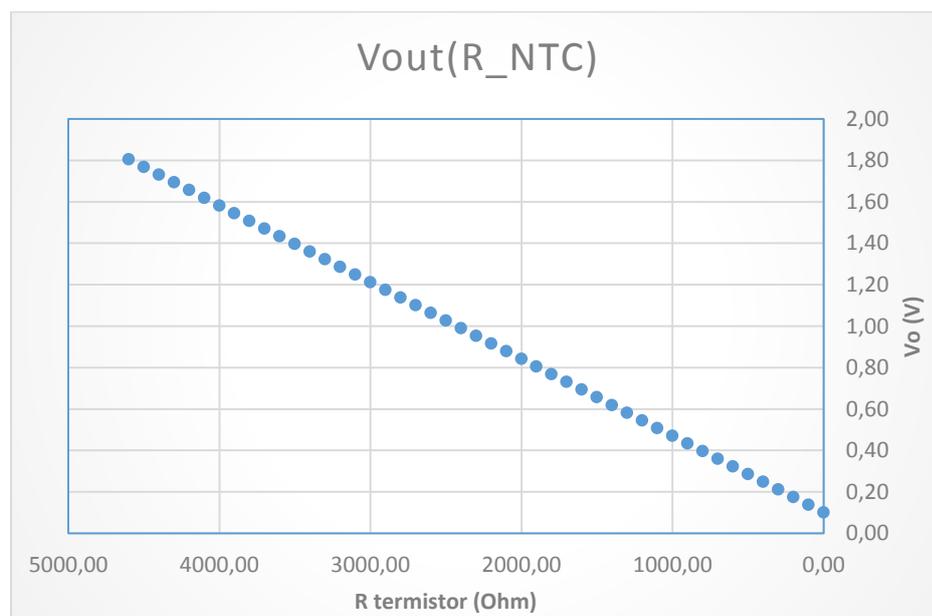
Con el trimmer se puede ajustar la ganancia para cada sensor, de manera que en su rango de variación de resistencia no se sobrepase el valor de 1,8V en la salida del operacional para no dañar la Beaglebone Black. Para esto se dispone del borne de calibrado y del jumper para poder hacer un ajuste del rango de tensión antes de conectar el circuito a la entrada analógica.

A menor resistencia del trimmer mayor ganancia, pudiéndose ajustar este hasta 0 saturando la salida del amplificador con ganancia infinita. El valor seleccionado para el trimmer es  $R_{trimmer}=1k\Omega$ . Con este valor la resistencia máxima del sensor que podemos conectar es:

$$R_{sensor.max} = R_{trimmer.max} \cdot \left[ \frac{V_o}{V_{Ref}} - 1 \right] = 1k\Omega \cdot \left[ \frac{1,8V}{0,1V} - 1 \right] = 17k\Omega$$

También se monta un condensador de  $0,1\mu\text{F}$  en paralelo con el sensor para filtrar ruidos de alta frecuencia.

A continuación se muestra un ejemplo de la respuesta para un termistor NTC de  $4,7\text{k}\Omega$  a  $20^\circ\text{C}$ . El trimmer se ajusta a un valor de  $270\Omega$  para tener  $1,8\text{V}$  en la salida a temperatura ambiente. Conforme aumenta la temperatura la salida baja hasta el offset de  $0,1\text{V}$ .



*Gráfico 1. Tensión de salida del amplificador en función de la resistencia del termistor NTC.*

d) Circuito de adaptación de tensión para salidas EHRPWM

Con este circuito se eleva la tensión de las salidas EHRPWM de la Beaglebone Black de  $3,3\text{V}$  a  $5\text{V}$ . Para ello se usa el circuito integrado LM339N que contiene cuatro comparadores capaces de conmutar a frecuencias muy elevadas con corrientes de entrada muy bajas haciéndolo compatible con salidas TTL, DTL, ECL, MOS y CMOS.

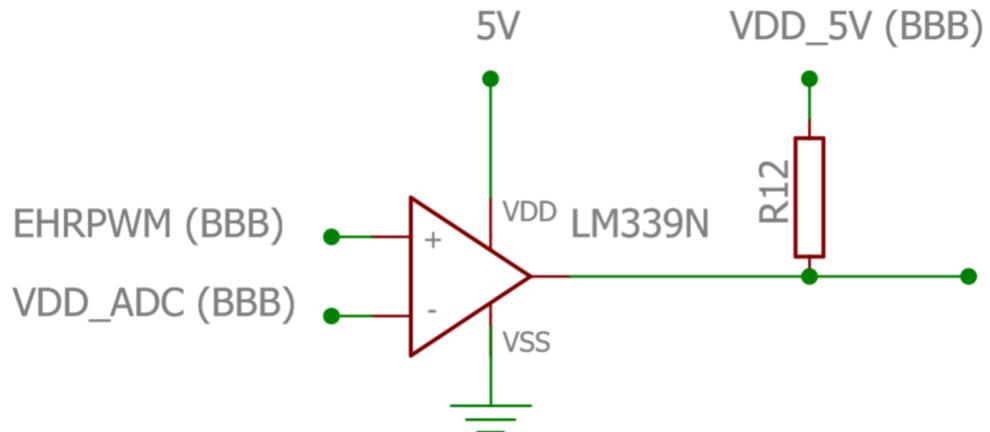


Figura 11. Circuito de adaptación de tensiones de salidas EHRPWM.

Cada comparador se conecta a una salida EHRPWM tal y como se muestran en la figura. En la entrada no inversora se conecta la señal EHRPWM de 3,3V. La tensión de referencia para el comparador se obtiene de la toma de tensión de la Beaglebone Black de 1,8V (VDD\_ADC) que está pensada para utilizarse como tensión de referencia para las señales analógicas. Puesto que los límites low y high de las salidas LVCMOS son 0,4V y 2,4V respectivamente, esta tensión de referencia es perfecta al estar dentro del rango.

El comparador se alimenta desde el circuito de protección descrito anteriormente, ya que si este está alimentado pueden aparecer tensiones en la entrada no inversora poniendo en peligro el mini-PC si este está apagado.

La salida, al ser en colector abierto, necesita una referencia de tensión a través de una resistencia que limita la corriente. El BJT del colector soporta como máximo 20mA. Se calcula un valor de resistencia para que limite esta corriente a 15mA para evitar sobrecalentamientos.

$$R_{12} = \frac{VDD\_5V}{I_{colector.max}} = \frac{5V}{15mA} = 333\Omega \approx 330\Omega$$

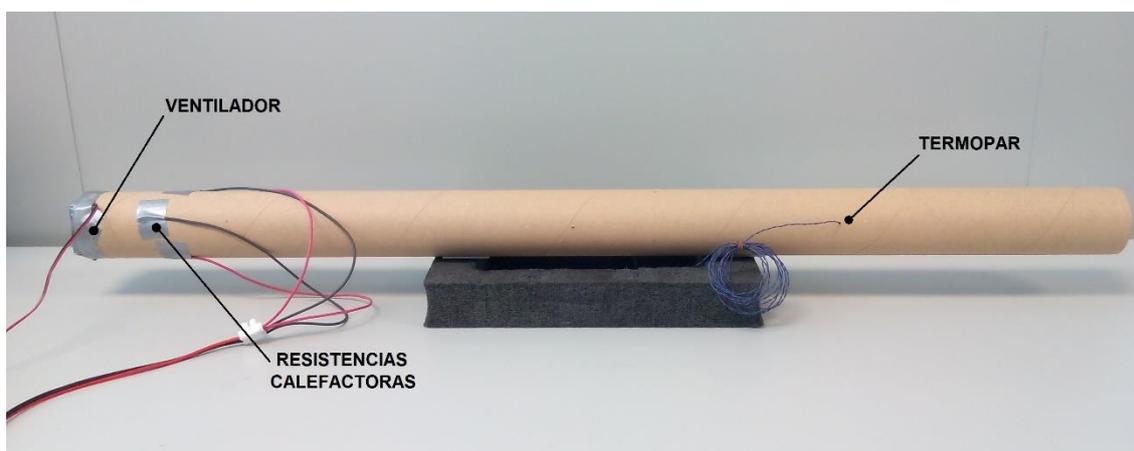
Puesto que las entradas PWM de los componentes electrónicos tienen resistencias muy altas, la caída de tensión en la resistencia de 330Ω es

despreciable frente a la caída de tensión en la resistencia de la entrada PWM que se conecte a las salidas de la capa. En el caso de conectar directamente una carga a estas salidas se debe considerar esta resistencia.

La tensión de referencia se toma directamente de la toma VDD\_5V capaz de suministrar hasta 1 A de corriente.

#### 1.5.4 Sistema térmico con retardo y circuitos para sensor y actuador

Se utiliza un sistema térmico con retardo tal y como el que se muestra en la imagen para probar el sistema de control en red para procesos continuos desarrollado en este proyecto.



*Figura 12. Sistema térmico con retardo.*

El sistema está formado por un tubo por el que circula una corriente de aire forzada mediante un pequeño ventilador situado en un extremo. A unos centímetros de este ventilador hay un par de resistencias calefactoras conectadas en paralelo que hacen la función de actuador del sistema.

Para medir la temperatura se utiliza un termopar situado en el extremo opuesto a las resistencias. El ventilador funciona de manera constante y tiene la finalidad de crear el retardo en la respuesta.

Es necesario desarrollar dos circuitos más para utilizar este sistema. Uno para amplificar la señal del termopar y otro para suministrar la potencia necesaria a las resistencias calefactoras. A continuación se detallan las características de estos circuitos:

### 1.5.4.1 Amplificador ajustable para termopar

Tal y como se muestra en la figura 14 se trata de un amplificador de señal en dos fases, uno de ellos de ganancia variable mediante un potenciómetro, y de un seguidor de tensión antes de la salida.

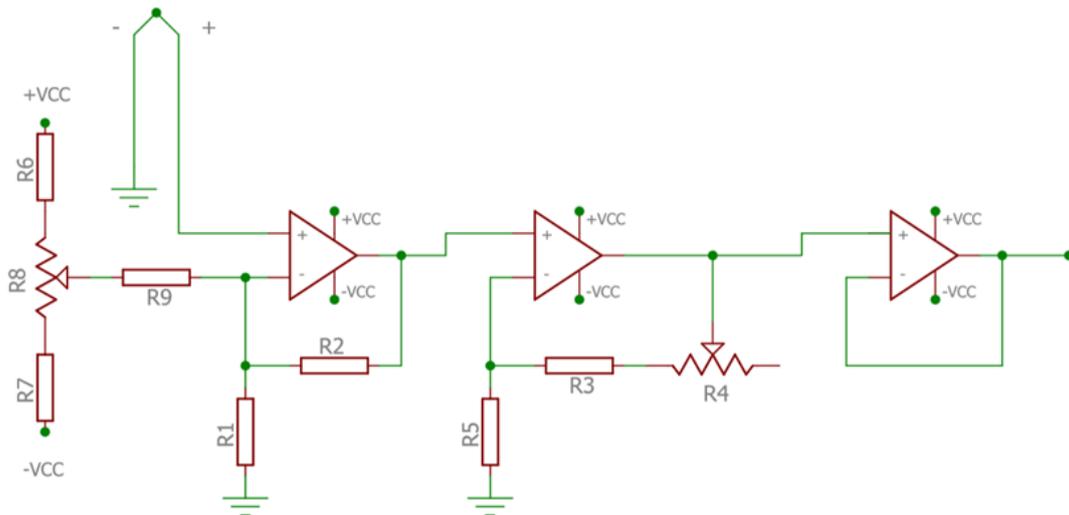


Figura 13. Circuito amplificador de señal para termopar.

Para el rango de temperaturas a medir, entre 20°C y 40°C aproximadamente, el termopar utilizado da una tensión entre  $V_{tp}=0,3\text{mV}$  y  $V_{tp}=0,5\text{mV}$  a la temperatura máxima según se conecten las resistencias calefactoras a 15V o 20V. Para tener una salida del orden de voltios se necesita una ganancia del orden de  $10^4$ . Se decide hacer la amplificación en dos fases con ganancias del orden de  $10^2$  cada una.

La función de transferencia del conjunto es la siguiente:

$$V_o = V_{tp} \cdot \left[ 1 + \frac{R_2}{R_1} \right] \cdot \left[ 1 + \frac{R_3 + R_4}{R_5} \right] = V_{tp} \cdot G_1 \cdot G_2$$

En primer lugar se eligen las resistencias  $R_1$  y  $R_2$  para tener una ganancia  $G_1$  de aproximadamente 100:

$$R_1 = 100\Omega ; R_2 = 10k\Omega ;$$

$$G_1 = \left[ 1 + \frac{R_2}{R_1} \right] = \left[ 1 + \frac{10000\Omega}{100\Omega} \right] = 101$$

Para dimensionar las resistencias y el rango de ajuste de la segunda fase de amplificación se hace una estimación de la ganancia necesaria en función del nivel de tensión en la salida del termopar. El objetivo es poder ajustar la salida a una tensión de 5V para el valor de temperatura máximo.

Estimación de ganancia G2 necesaria					
$V_{tp}$	$G_1$	$G_{2min}$	$G_{2max}$	$V_{o.min}$	$V_{o.max}$
0,0003	101	80	180	2,424	5,454
0,0005	101	80	180	4,04	9,09

Tabla 2. Estimación de ganancias para amplificador de señal para termopar.

Con una ganancia variable entre 80 y 180 se puede calibrar el circuito para tener una salida de 5V a temperatura máxima para el rango de tensiones de salida del termopar.

Con esta estimación ya se pueden dimensionar las resistencias restantes.

$$R_3 = 820\Omega ; R_4 = 1k\Omega ; R_5 = 100\Omega ;$$

Ganancias y tensiones de salida reales con las resistencias seleccionadas					
$V_{in}$	$G_1$	$G_{2min}$	$G_{2max}$	$V_{o.min}$	$V_{o.max}$
0,0003	101	83	183	2,5149	5,5449
0,0005	101	83	183	4,1915	9,2415

Tabla 3. Ganancias y valores de salida del amplificador de señal para termopar.

El termopar también tiene un offset de 1mV, que frente a los 0,4mV que entrega de media a temperatura máxima, es un valor muy elevado que nos limita el rango de medida. Para compensar este efecto se utiliza otro potenciómetro en modo de divisor de tensión ajustable para sumar un valor negativo igual a este offset a la entrada del amplificador.

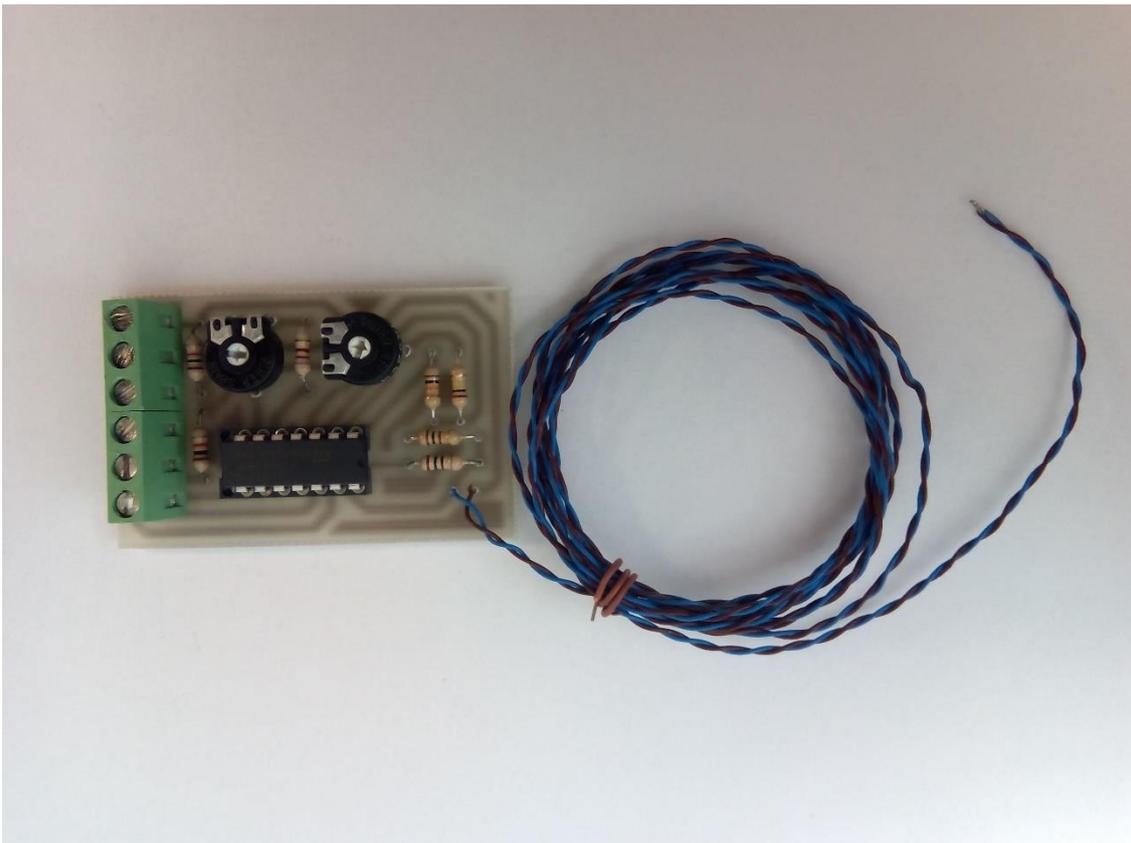
El valor de las resistencias seleccionadas es el siguiente:

$$R_6 = 1k\Omega ; R_7 = 1k\Omega ; R_8 = 1k\Omega ; R_9 = 100k\Omega$$

Las resistencias  $R_6$  y  $R_7$  sirven para mejorar la precisión a la hora de ajustar el offset con el potenciómetro. La resistencia  $R_9$  tiene la función de limitar las corrientes para aislar las dos zonas del circuito.

Se utiliza el circuito integrado LM324N, que contiene cuatro amplificadores operacionales capaces de entregar hasta 40mA cada uno. Se puede alimentar con tensión positiva o simétrica no sobrepasando los 30V de tensión entre  $+V_{cc}$  y  $-V_{cc}$ , y no siendo ninguna de estas menor a 3V.

Se desarrolla un circuito impreso para integrar el circuito y el termopar conformando así un sensor de temperatura analógico de offset y ganancia ajustable, alimentado con tensión simétrica entre  $\pm 3V$  y  $\pm 30V$  y capaz de entregar 40mA sin afectar a la medida gracias al seguidor de tensión en la salida.



*Figura 14. Sensor de temperatura analógico.*

#### 1.5.4.2 Amplificador de potencia para resistencias calefactoras

Las resistencias calefactoras consumen una potencia de 26W cuando se conectan a su tensión máxima de 20V, consumiendo una corriente de 1,3A. Es necesario utilizar un amplificador de potencia para poder controlar esta carga.

Se decide utilizar el circuito integrado ULN2003, que contiene 7 amplificadores Darlington de alta ganancia capaces de soportar 500mA cada uno. Conectando varios de ellos en paralelo se consigue la carga necesaria para alimentar las resistencias calefactoras. La solución adoptada se muestra en la figura 15.

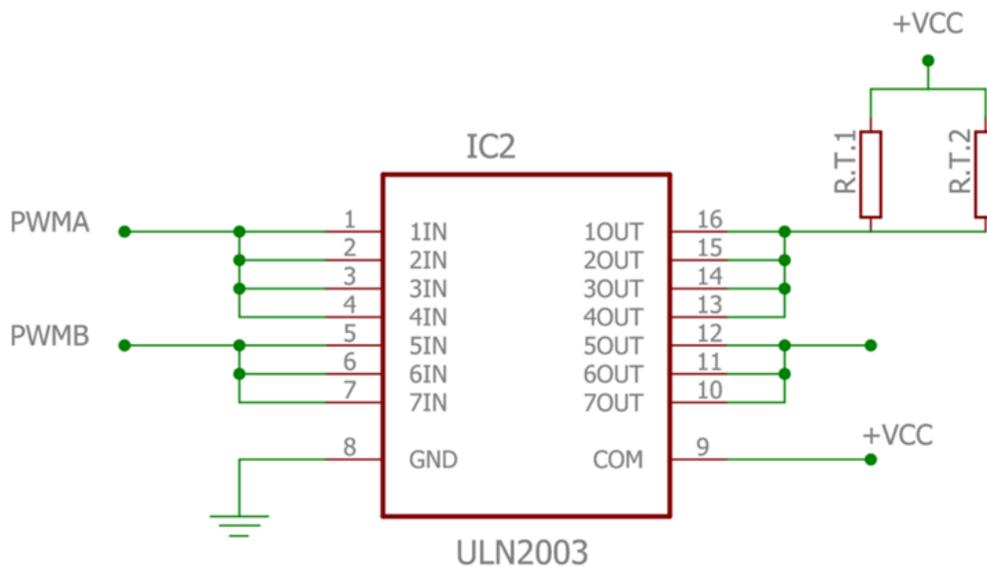


Figura 15. Circuito amplificador de potencia para resistencias calefactoras.

Se agrupan cuatro amplificadores para disponer de una carga de hasta 2A gestionados con la entrada PWMA y otros tres con los que se dispone de hasta 1,5A gestionados con la entrada PWMB. En esta aplicación se utiliza el PWMA en cuya salida se conectan las dos resistencias calefactoras en paralelo.

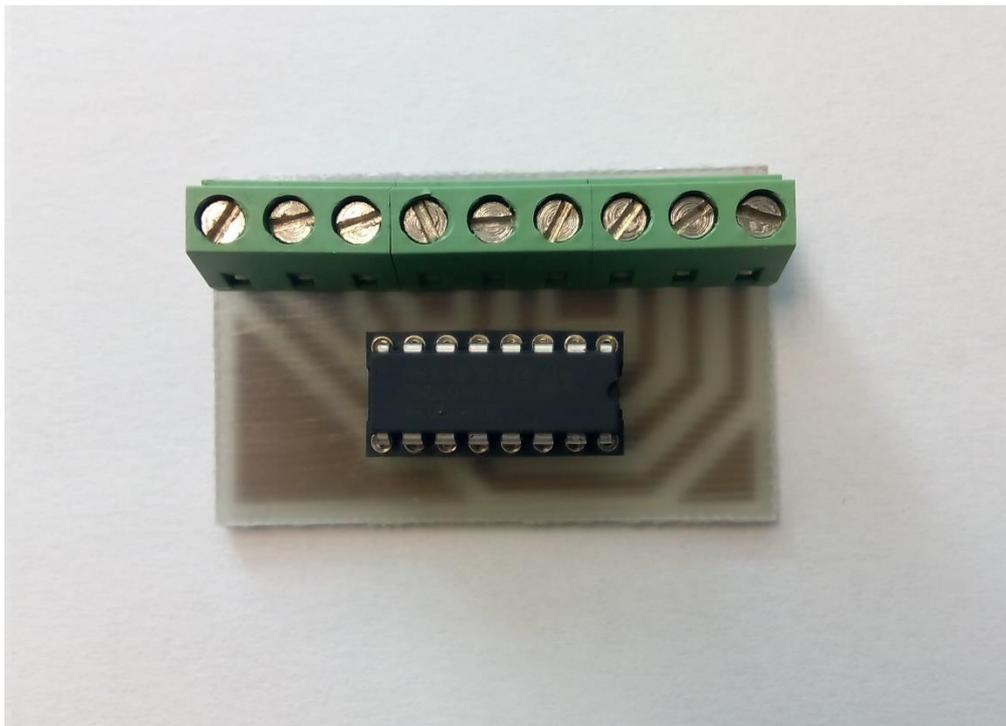
Cada transistor Darlington tiene una resistencia de base de 2,7kΩ. Hay que recordar que cada una de las salidas EHRPWM de la capa a la que se va a conectar tienen una resistencia de 330Ω conectada entre la toma VDD\_5V de la Beaglebone Black y el colector abierto del comparador LM339. La corriente de base de cada transistor es por lo tanto:

$$I_{b,Darlington} = \frac{VDD\_5V}{R_{o,EHRPWM} + R_{b,Darlington}} = \frac{5V}{330\Omega + 2,7k\Omega} = 1,65mA$$

corriente suficiente ya que estos transistores se saturan con 0,93mA.

El circuito integrado ULN2003 también incorpora un diodo de recirculación para cada transistor para poder conectar cargas inductivas. Estos se conectan a VCC a través del pin COM.

Para realizar las conexiones se diseña el circuito impreso mostrado en la figura 16. El resultado es un amplificador de potencia de dos canales controlados mediante señales PWM de 5V.



*Figura 16. Amplificador de potencia para resistencias calefactoras.*

#### 1.5.5 Estructura física del sistema de control en red

Llegados a este punto ya se dispone de todo el hardware e instrumentación necesarios para implementar el sistema de control en red mediante el estándar IEC-61499. Las conexiones de los distintos componentes se muestran en la figura 17.

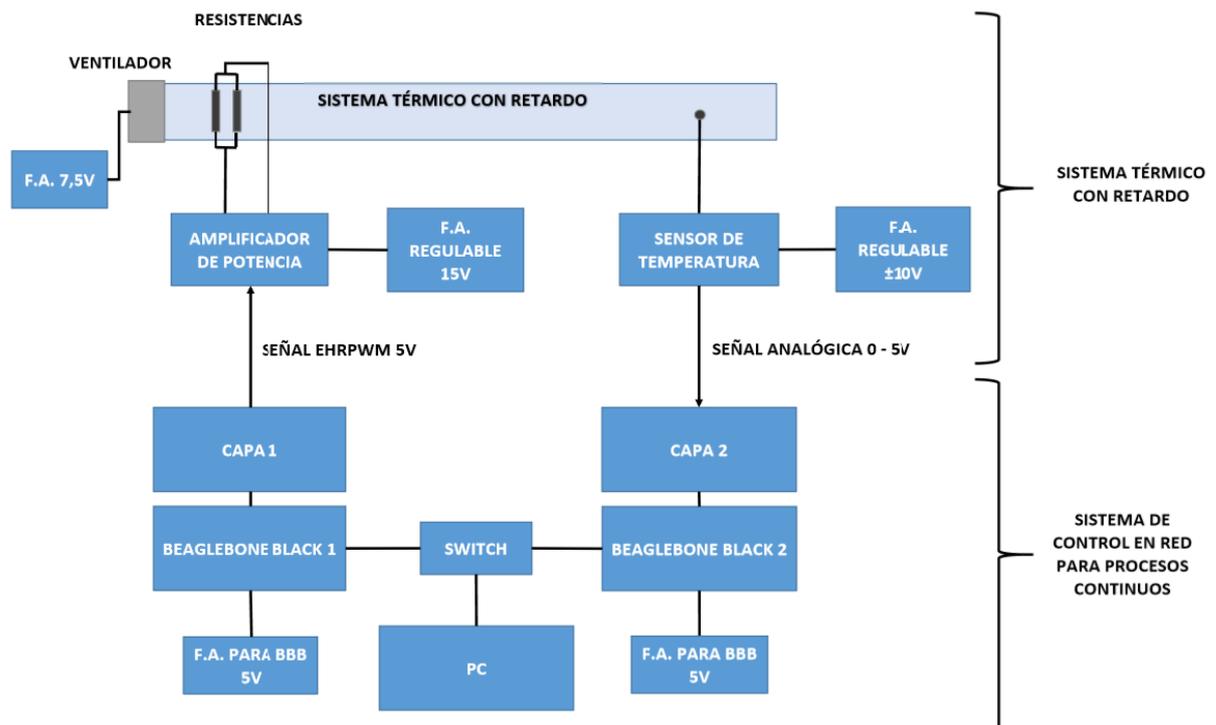


Figura 17. Esquema del sistema de control en red y del proceso continuo.

Se distinguen dos partes del montaje. Por un lado se tiene el sistema térmico con retardo, que es el proceso continuo sobre el que se va a utilizar el sistema de control en red desarrollado en este proyecto.

Del primero cabe comentar que el ventilador se alimenta a una tensión constante de 7,5V para crear el flujo de aire con el que se consigue el retardo en la respuesta. Para el actuador, en este caso las resistencias, se utiliza el amplificador de potencia desarrollado previamente, el cual está controlado con una de las salidas EHRPWM de 5V de la capa 1 y se alimenta a 15V con una fuente de alimentación regulable de laboratorio. Por último, para la medición de temperatura se utiliza el sensor desarrollado anteriormente alimentado con una tensión simétrica de  $\pm 10V$  mediante otra fuente regulable. La salida analógica de 0 a 5V del sensor de temperatura se conecta a la entrada analógica de rango seleccionable de la capa 2, configurada en este caso para el rango de 5V.

El sistema de control en red está formado por las dos Beaglebone Black y un PC conectados a través de Ethernet y las respectivas capas interfaz de entradas analógicas y salidas EHRPWM. Cada conjunto de mini-PC y capa conforman un nodo controlador del sistema equipado con una entrada analógica de rango seleccionable entre 1,8/5/10V, una entrada analógica con conexión directa de sensores resistivos y cuatro salidas EHRPWM. La idea de este diseño es aportar

versatilidad al sistema de manera que sirva para controlar un amplio abanico de procesos continuos con diferente número de sensores y actuadores.

La función del PC es en principio la de realizar la programación centralizada del sistema distribuido y también hacer la función de HMI, pero también puede usarse como nodo controlador ejecutando algoritmos de control en él aunque no disponga de entradas y salidas físicas con las que interactuar con el proceso a controlar.

#### 1.5.6 Desarrollo e implementación del sistema de control distribuido

En este apartado se va a describir el proceso de desarrollo de algoritmos de control según el estándar IEC-61499 utilizando la herramienta de desarrollo 4DIAC-IDE y la forma de implementarlos usando las plataformas de ejecución FORTE y FBRT. Las imágenes utilizadas para las explicaciones en este apartado han sido imprimidas directamente desde la herramienta de desarrollo 4DIAC-IDE.

El objetivo es implementar un control PI para el sistema térmico con retardo estando sensor y actuador conectados a dispositivos diferentes tal y como se aprecia en la figura 17. El PC se utilizará para ejecutar una aplicación HMI sencilla con la que visualizar las variables principales del proceso e introducir el parámetro de referencia para el control PI.

##### 1.5.6.1 Modelado de sistema, dispositivos y recursos

El primer paso en la implementación de un sistema de control distribuido según el estándar IEC-61499 es modelar el sistema con la herramienta de desarrollo determinando los dispositivos implicados y los canales de comunicación utilizados entre ellos, así como los recursos necesarios.

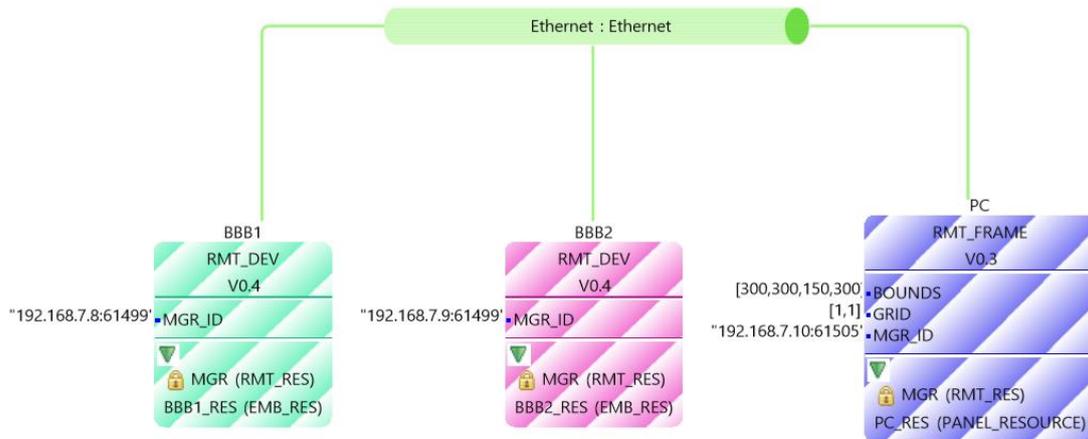


Figura 18. Modelo de sistema para control PID.

En el sistema se definen dos dispositivos remotos para las Beaglebone Black asignándoseles direcciones IP pertenecientes a la red de área local configurada. También se define un marco remoto para ejecutar en el PC la aplicación HMI. A este se le asigna la IP del PC. Estas direcciones y puertos son las que utilizará 4DIAC-IDE para descargar a cada dispositivo los algoritmos modelados. Para esta función cada dispositivo dispone de su recurso MGR\_ID, que no puede ser modificado.

Lo siguiente es definir los recursos, que son los marcos sobre los que se ejecutarán los bloques funcionales que se mapearan posteriormente, es decir, son el conjunto de recursos que dispondrá cada dispositivo para este fin. Aunque en este caso solo se define un recurso por dispositivo, hay que tener en cuenta que aunque pertenezcan al mismo dispositivo, la ejecución de cada recurso es independiente y requiere del uso de canales de comunicación entre ellos.

En este sistema estos recursos son del tipo "recurso embebido" en las Beaglebone Black, llamados BBB1\_RES y BBB2\_RES. El recurso PC\_RES es del tipo "recurso de panel".

#### 1.5.6.2 Modelado de aplicaciones

4DIAC-IDE ofrece una amplia selección de librerías con las que poder implementar prácticamente cualquier aplicación de control. Además también se pueden crear bloques funcionales y recompilar la plataforma de ejecución FORTE para que sea capaz de implementarlos en el dispositivo.

Las aplicaciones se modelan independientemente del sistema en el que se ejecutarán. Para este proyecto se desarrollan dos de ellas, PI y HMI que se explicarán a continuación.

En la figura 19 se aprecia la forma en la que se estructuran los algoritmos de control en forma de aplicaciones según se define en el estándar. Las líneas rojas simbolizan los eventos con los que se gestiona la ejecución de los diferentes bloques de función. Con las entradas/salidas de evento INIT/INITO se establece una cadena con la que se da la orden de inicializar los bloques de función que lo requieren. Cuando un bloque termina su inicialización se activa el evento INITO inicializándose así el siguiente.

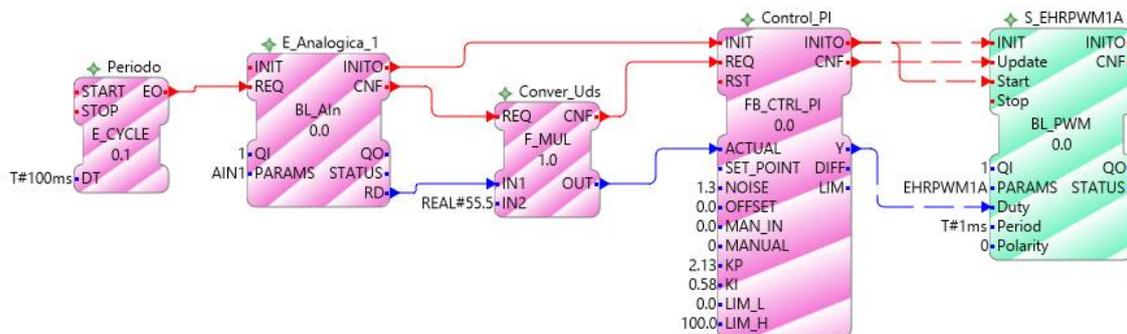


Figura 19. Aplicación PI.

El bloque funcional "Periodo", de tipo E\_CYCLE, da una señal de evento periódicamente a la frecuencia definida en su entrada de datos DT. Aquí se establece otra cadena de eventos en la que a cada bloque funcional se le solicita la ejecución de su función con la entrada REQ y posteriormente, confirmando este su ejecución con la salida de evento CNF, se le solicita la ejecución al siguiente bloque. De esta manera se obtiene un algoritmo de control que se ejecuta periódicamente.

Como se aprecia en la figura 19 la aplicación está formada por un bloque funcional de lectura de entrada analógica "E\_Analogica\_1" para medir la temperatura, seguido de otro bloque de tipo F\_MUL para ajustar el rango de la salida del primero, que está en voltios y va de 0 a 1,8V, al rango adimensional de 0 a 100%. Este flujo de datos está representado con líneas azules.

El siguiente bloque es el propio PI, que dispone de varias entradas de datos con las que se definen los parámetros necesarios para su ejecución (Kp, Ki,

Limites, etc...), así como la evolución de la variable medida del proceso o el valor de la referencia.

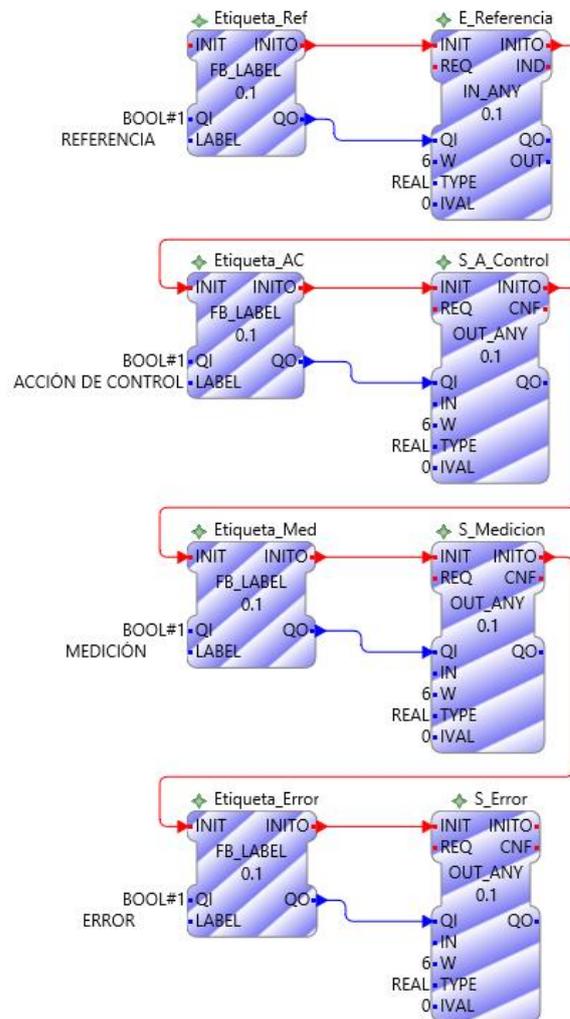
Por último, la salida de la acción de control del PI se le pasa al bloque funcional de la salida EHRPWM. Tanto el rango de la acción de control como el de la entrada del ciclo de trabajo del bloque EHRPWM van de 0 a 100, por lo que el algoritmo funciona con unidades adimensionales. Este porcentaje se traducirá físicamente a un rango de 0 a 5V al ejecutarse este bloque en la Beaglebone Black.

La variables booleana QO se utiliza para indicar que ha habido un error de ejecución en el bloque funcional y si la entrada Q1 es 0 el bloque no se ejecuta. Según en qué aplicación se trabaje puede ser interesante condicionar la ejecución del bloque a determinadas condiciones o simplemente encadenarlas de manera que si un bloque falla no se ejecuten los siguientes. En esta aplicación, al ser sencilla no es necesario por lo que directamente se les asigna el valor 1.

En la imagen se observa que faltan las señales de eventos de INIT en el bloque de entrada analógica y el de START en el bloque periodo. Estas señales se asignaran al "mapear" la aplicación, de manera que la orden de ejecución inicial provenga del dispositivo en el que se ejecuta dichos bloques. Mapear es la acción de distribuir aplicación en los recursos que pueden o no ser los mismos.

También falta el dato de la señal de referencia, entrada SET\_POINT del bloque "Control\_PI". Este dato se obtendrá de la aplicación HMI que se explicará a continuación.

El color de los bloques de función indica que estos se encuentran mapeados en distintos recursos. Por el mismo motivo las líneas de flujo de eventos y datos aparecen en discontinuo entre los bloques mapeados en uno y otro recurso. Esta distribución se verá con más detalle en el siguiente apartado.



*Figura 20. Aplicación HMI.*

La aplicación HMI consta de cuatro parejas de bloques funcionales, uno para que se muestre una etiqueta, y otro para introducir o mostrar un dato por pantalla. En cada uno de ellos se define el tipo de dato, el número de caracteres y el valor inicial. En la aplicación se encadenan los bloques para que se inicialicen en ese orden y en la pantalla aparezcan los recuadros ordenados. Los eventos REQ y los datos de entrada o salida se gestionaran desde el dispositivo una vez mapeada la aplicación.

### 1.5.6.3 Mapeado de aplicaciones y configuración de comunicaciones

Una vez diseñadas las aplicaciones el siguiente paso es mapearlas entre los recursos de los dispositivos definidos anteriormente. Se hace un esquema para

decidir esta distribución y para evaluar que datos se deben intercambiar entre ellos con el fin de establecer las comunicaciones necesarias.



Figura 21. Esquema de distribución de algoritmo de control y flujo de datos.

Puesto que el sensor está conectado a la Beaglebone Black 2 la lectura de la entrada analógica debe realizarse en su recurso. Lo mismo sucede con el bloque para salida EHRPWM que debe ir en la Beaglebone Black 1 ya que el amplificador de potencia para las resistencias calefactoras está conectado a este nodo.

El PI podría ejecutarse en cualquiera de los dos nodos, pero se decide mapearlo en la Beaglebone Black 2 ya que así el flujo de datos con el HMI es más sencillo.

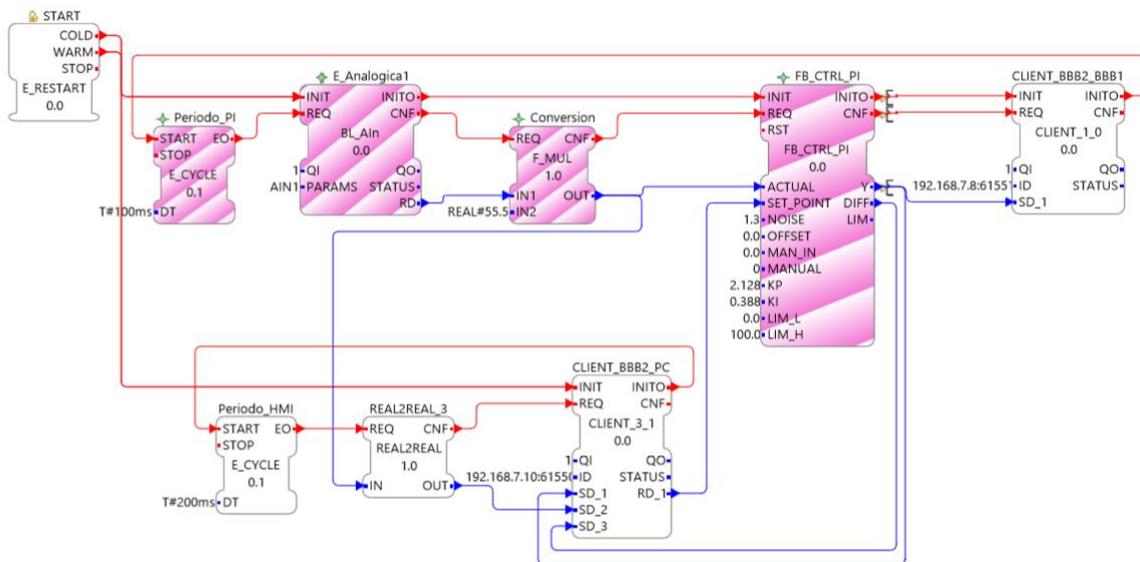


Figura 22. Recurso BBB2\_RES.

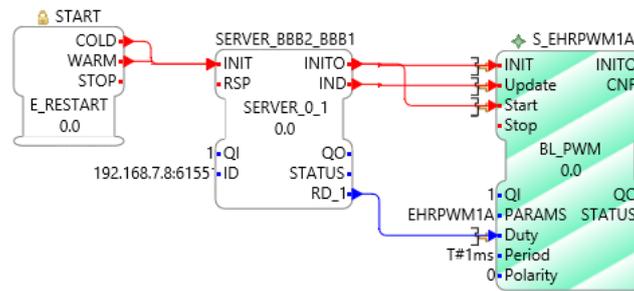


Figura 23. Recurso BBB1\_RES.

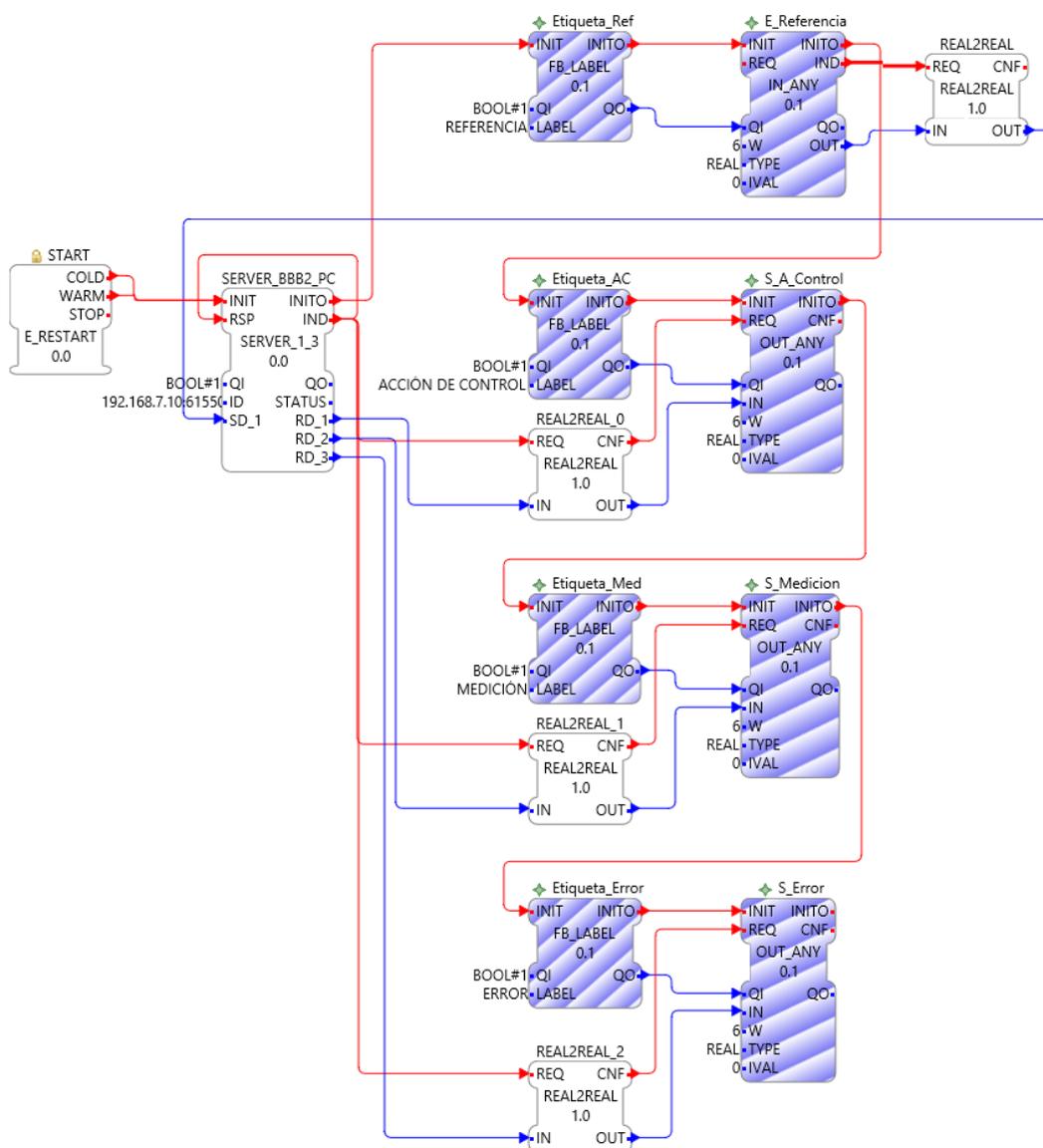


Figura 24. Recurso PC\_RES.

En las figuras 22, 23 y 24 se observa cómo quedan mapeadas las aplicaciones PI y HMI en los diferentes recursos. También se crean las comunicaciones que permitirán el intercambio de datos entre los recursos del sistema y algunos bloques funcionales más que son necesarios para que los algoritmos se ejecuten correctamente. Estos bloques, que aparecen en blanco, pertenecen directamente al recurso, a diferencia de las aplicaciones que existen como modelo independiente del sistema a la vez de como parte del algoritmo del recurso.

Cada recurso tiene un bloque START que no puede ser modificado y que da la señal de evento inicial para inicializar los bloques funcionales contenidos en él. También se usan varios bloques de tipo REAL2REAL que son necesarios cuando tanto la salida de un dato como la entrada al siguiente bloque funcional son de tipo ANY, por lo que el tipo de dato quedaría indeterminado si no se usasen.

Las comunicaciones se hacen con protocolo TCP/IP con dos parejas cliente/servidor, una entre los recursos BBB2\_RES y BBB1\_RES para pasar la acción de control del PI al bloque de EHRPWM y otra pareja entre BBB2\_RES y PC\_RES para compartir datos entre el PI y el HMI. Estos son, la referencia que se introduce por pantalla y la medición, la acción de control y el error que se muestran por pantalla.

#### *1.5.6.4 Implementación de los modelos*

Se usan las plataformas de ejecución FORTE en las Beaglebone Black y FBRT en el PC para implementar los modelos desarrollados con 4DIAC-IDE. Estos softwares se deben estar ejecutando en los dispositivos implicados para poder implementar desde 4DIAC-IDE los algoritmos.

Para este fin el primer paso es configurar las Beaglebone Black con una IP estática y compatible con la red de área local configurada. Recordemos que estos dispositivos llevan instalado el sistema operativo Angstrom Linux, por lo que es posible conectarse a ellos desde el PC a través de puerto serie utilizando su puerto mini-USB para cambiar su configuración de red. Para realizar esta tarea se utiliza el software de código libre PuTTY.

Una vez se realiza la conexión remota vía puerto serie se modifica el archivo "interfaces" del mini-PC que es el que contiene la configuración de red. Para reeditar este archivo se utiliza el editor de texto de Linux nano.

```

COM6 - PuTTY
RX bytes:14602 (14.2 KiB) TX bytes:10059 (9.8 KiB)

root@beaglebone:~# cd /etc/network
root@beaglebone:/etc/network# nano interfaces
GNU nano 2.2.6 File: interfaces

This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
#allow-hotplug eth0
auto eth0
iface eth0 inet static
address 192.168.7.9
netmask 255.255.255.0
# Example to keep MAC address between reboots
#hwaddress ether DE:AD:BE:EF:CA:FE

# The secondary network interface
#allow-hotplug eth1
iface eth1 inet dhcp

[ Read 34 lines ]
^C Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell

```

Figura 25. Reconfiguración de red de la Beaglebone Black con el editor de texto de Linux nano.

Ejecutando el comando “ifconfig” de Linux comprobamos que la configuración de red es correcta. Una vez configuradas ya se puede establecer conexión vía Ethernet con las dos Beaglebone Black.

```

192.168.7.9 - PuTTY
Last login: Wed Apr 23 20:20:44 2014
root@beaglebone:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 78:a5:04:bc:97:1c
          inet addr:192.168.7.9  Bcast:192.168.7.255  Mask:255.255.255.0
          inet6 addr: fe80::7aa5:4ff:febc:971c/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:59 errors:0 dropped:0 overruns:0 frame:0
          TX packets:78 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:6784 (6.6 KiB)  TX bytes:14063 (13.7 KiB)
          Interrupt:40

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

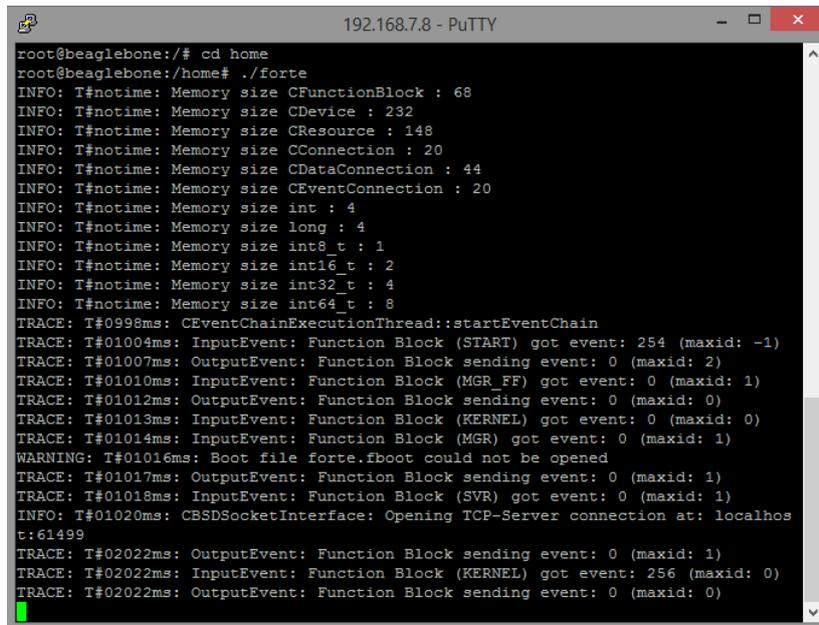
usb0     Link encap:Ethernet  HWaddr 8a:4f:e5:e2:bf:21
          inet addr:192.168.7.2  Bcast:192.168.7.3  Mask:255.255.255.252
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@beaglebone:~#

```

Figura 26. Detalle de configuración de red de la Beaglebone Black.

El siguiente paso es transferir el software FORTE a los mini-PC que se realiza con el software Fillezilla que es un cliente FTP multiplataforma de código abierto y software libre. La ejecución FORTE se realiza conectándose con PuTTY a las Beaglebone Black vía SSH.



```
192.168.7.8 - PuTTY
root@beaglebone:/# cd home
root@beaglebone:/home# ./forte
INFO: T#notime: Memory size CFunctionBlock : 68
INFO: T#notime: Memory size CDevice : 232
INFO: T#notime: Memory size CResource : 148
INFO: T#notime: Memory size CConnection : 20
INFO: T#notime: Memory size CDataConnection : 44
INFO: T#notime: Memory size CEventConnection : 20
INFO: T#notime: Memory size int : 4
INFO: T#notime: Memory size long : 4
INFO: T#notime: Memory size int8_t : 1
INFO: T#notime: Memory size int16_t : 2
INFO: T#notime: Memory size int32_t : 4
INFO: T#notime: Memory size int64_t : 8
TRACE: T#0998ms: CEventChainExecutionThread::startEventChain
TRACE: T#01004ms: InputEvent: Function Block (START) got event: 254 (maxid: -1)
TRACE: T#01007ms: OutputEvent: Function Block sending event: 0 (maxid: 2)
TRACE: T#01010ms: InputEvent: Function Block (MGR_FF) got event: 0 (maxid: 1)
TRACE: T#01012ms: OutputEvent: Function Block sending event: 0 (maxid: 0)
TRACE: T#01013ms: InputEvent: Function Block (KERNEL) got event: 0 (maxid: 0)
TRACE: T#01014ms: InputEvent: Function Block (MGR) got event: 0 (maxid: 1)
WARNING: T#01016ms: Boot file forte.fboot could not be opened
TRACE: T#01017ms: OutputEvent: Function Block sending event: 0 (maxid: 1)
TRACE: T#01018ms: InputEvent: Function Block (SVR) got event: 0 (maxid: 1)
INFO: T#01020ms: CBSocketInterface: Opening TCP-Server connection at: localhos
t:61499
TRACE: T#02022ms: OutputEvent: Function Block sending event: 0 (maxid: 1)
TRACE: T#02022ms: InputEvent: Function Block (KERNEL) got event: 256 (maxid: 0)
TRACE: T#02022ms: OutputEvent: Function Block sending event: 0 (maxid: 0)
```

*Figura 27. Ejecución de FORTE en la Beaglebone Black.*

A partir de este punto las Beaglebone Black ya están listas para descargarles los modelos desde 4DIAC-IDE. Para implementar los algoritmos contenidos en el recurso PC\_RES es necesario ejecutar FBRT en el PC. Esta plataforma de ejecución basada en Java es necesaria ya que soporta los bloques funcionales HMI de su librería. Esto se puede hacer cómodamente desde la vista de implementación de 4DIAC-IDE, tal y como se muestra en la figura 28, simplemente indicando previamente la ruta donde se encuentra guardado.

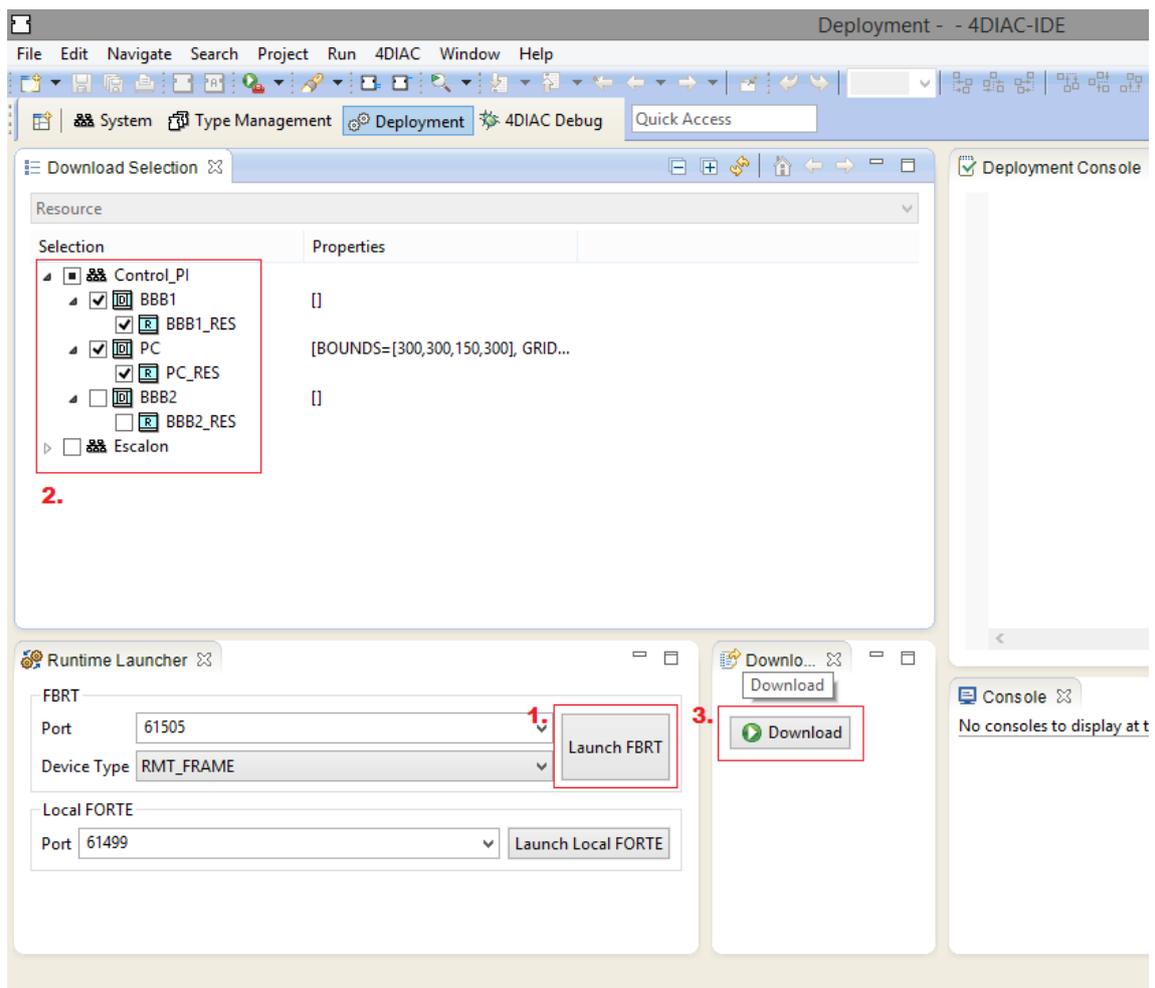


Figura 28. Detalle de la vista de implementación de 4DIAC-IDE con los pasos a seguir para la implementación de los modelos.

Una vez funcionando FORTE en las Beaglebone Black y FBRT en el PC se seleccionan los recursos BBB1\_RES y PC\_RES y se descargan pulsando "Download". Estos dos recursos se descargan antes porque son los que tienen los servidores de las comunicaciones, así cuando se descargan los algoritmos del BBB2\_RES estas ya están inicializados.

En la figura 29 se muestra cómo, una vez descargados los programas, se abre una ventana de Java con el HMI mostrando las señales del control PI. En la consola de implementación aparecen las acciones realizadas durante la descarga y también los errores si los hubiera.

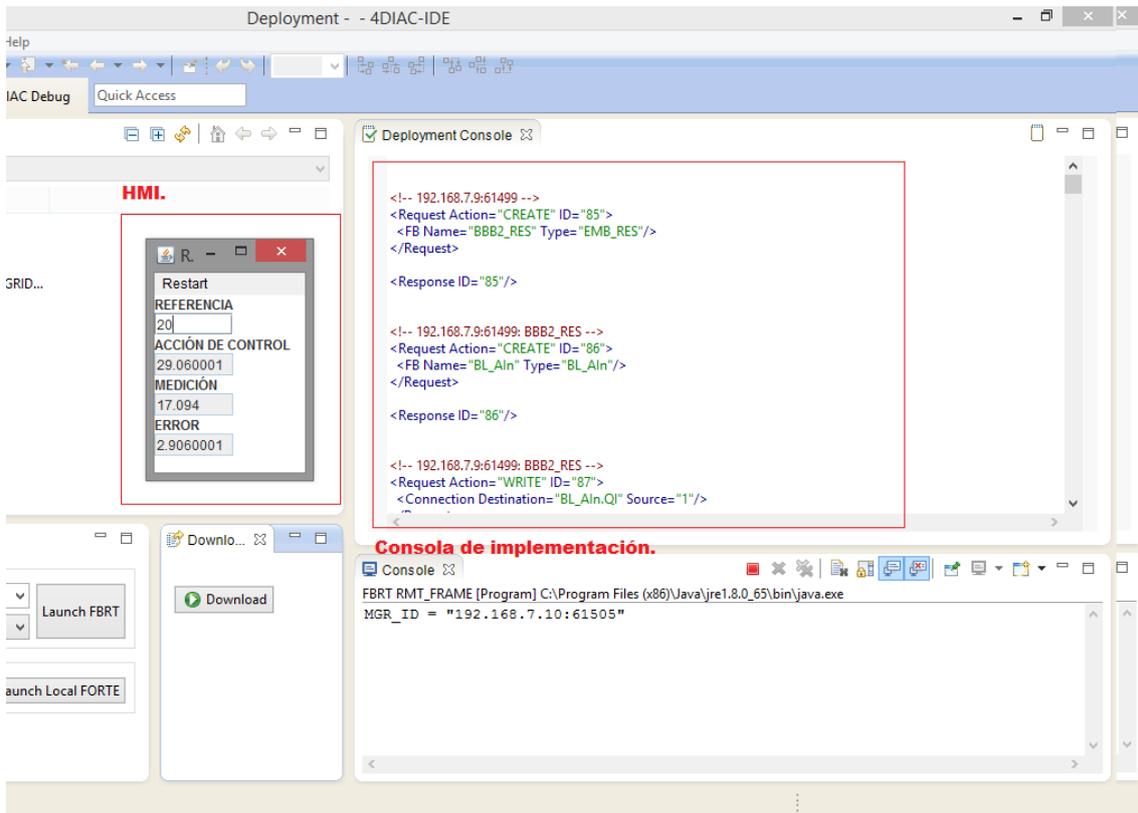


Figura 29. Detalle de la vista de implementación de 4DIAC-IDE tras la descarga de los programas a los dispositivos.

### 1.5.7 Diseño de control PI y prueba de funcionamiento del sistema

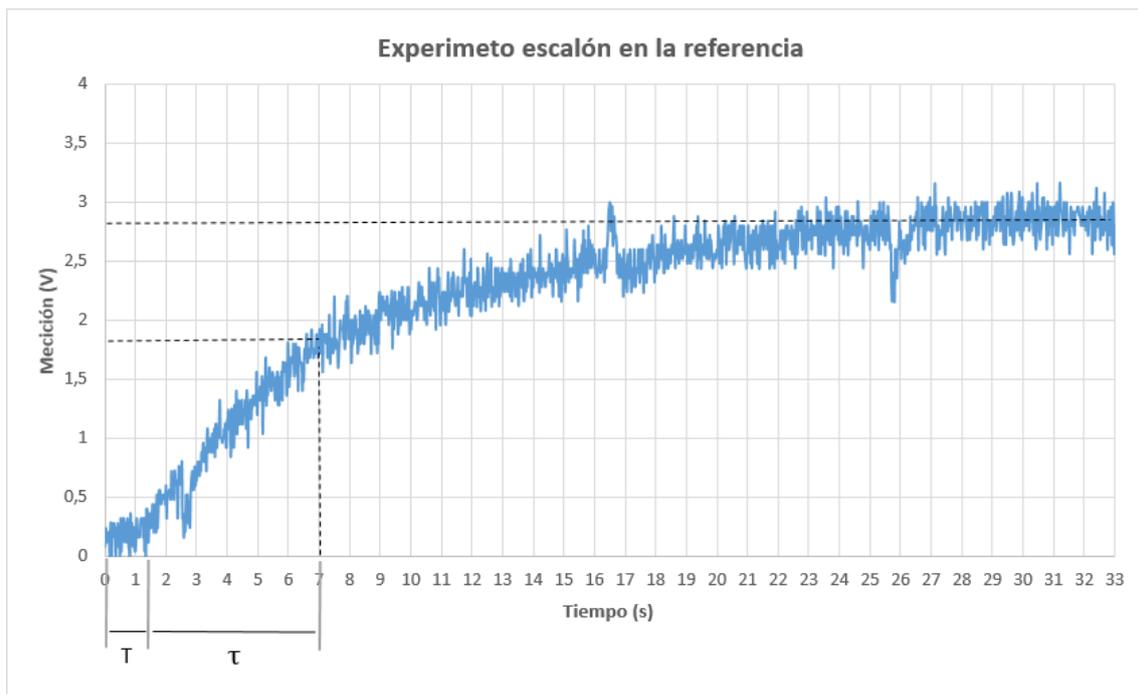
Con el fin de probar el funcionamiento del sistema de control en red desarrollado en este proyecto se diseña un control PI para el proceso térmico con retardo explicado anteriormente. Para ello previamente se desarrolla un programa con 4DIAC-IDE para hacer un experimento escalón, que por su sencillez, no se va a hacer hincapié en su explicación. Simplemente cabe comentar que se trata de una aplicación HMI con la que se le pasa el dato del ciclo de trabajo a un bloque funcional de EHRPWM para poder controlar la potencia de las resistencias en su rango de 0 a 100%.

Llegados a este punto es conveniente aclarar los rangos de las señales de medida y del actuador en las diferentes partes del sistema.

	Proceso	Capa	BBB	Controlador
Señal de medida	20°C – 40°C	0 - 5 V	0 – 1,8 V	0 – 100%
Señal de actuador	0 – 15 V	0 – 5 V	0 – 3,3 V	0 – 100%

*Tabla 4. Rangos de las señales de sensor y actuador.*

Se lleva a cabo el experimento escalón. Para capturar los datos de la respuesta se mide directamente la señal del sensor con un osciloscopio modelo Tektronix TDS 1001B que permite guardar los datos en formato .CSV para poder posteriormente interpretarlos con Excel. En el gráfico 2 se muestran los resultados.



*Gráfico 2. Respuesta del proceso térmico con retardo ante escalón en la referencia.*

El escalón en la referencia es de 0 a 60%. Tomando medidas en el gráfico de la respuesta se obtiene la función de transferencia del proceso térmico.

$$T = 1,5s ; \tau = 7 - 1,5 = 5,5s$$

$$\Delta y_V = 2,8 - 0,1 = 2,7V ; \Delta y = (2,7V/5V) \cdot 100 = 54 \%$$

$$K = \frac{\Delta y}{\Delta u} = \frac{54}{60} = 0,9$$

$$G(s) = \frac{e^{-T \cdot s} \cdot K}{\tau \cdot s + 1} = \frac{e^{-1,5 \cdot s} \cdot 0,9}{5,5 \cdot s + 1}$$

Conocida la función de transferencia del proceso se determinan los parámetros del PI utilizando una aplicación Java de diseño de PID gratuita disponible en la dirección web:

<https://sites.google.com/a/uji.es/freepidtools/home/download-herramienta-de-diseno-de-pid>

Esta aplicación calcula los parámetros del controlador a partir de las constantes de tiempo y la ganancia.

The screenshot shows a web application titled "Ajuste de PID en frecuencia" by Roberto Sanchis. The interface includes several sections for configuring a PID controller:

- Características del sensor:** Ruido (pico-pico) = 0.100
- Características del actuador:** umin = 0.0, umax = 5.0
- Periodo de muestreo digital:** T (seg) = 0.00.100
- Define el modelo del proceso de una de las 3 formas siguientes:**
  - 1. Introduce los coeficientes del numerador y denominador de G(s):**

numG = 0.9000 + 0.0000 s + 0.0000 s<sup>2</sup> + 0.0000 s<sup>3</sup> + 0.0000 s<sup>4</sup> + 0.0000 s<sup>5</sup>

denG = 1.0000 + 5.5000 s + 0.0000 s<sup>2</sup> + 0.0000 s<sup>3</sup> + 0.0000 s<sup>4</sup> + 0.0000 s<sup>5</sup> + 0.0000 s<sup>6</sup>

Retardo = 1.5000
  - 2. O bien define el modelo introduciendo las constantes de tiempo:**

K = 0.9,  tau1 = 5.5,  tau2 = 1.000,  tau3 = 1.000,  1/wn = 0.000, psi = 0.000,  beta = 0.000

Retardo = 1.5

Integrador,  K negativa,  beta negativa,
  - 3. O bien carga el modelo desde un fichero:**

Figura 30. Introducción de constantes del proceso en la aplicación Java de diseño de PID.

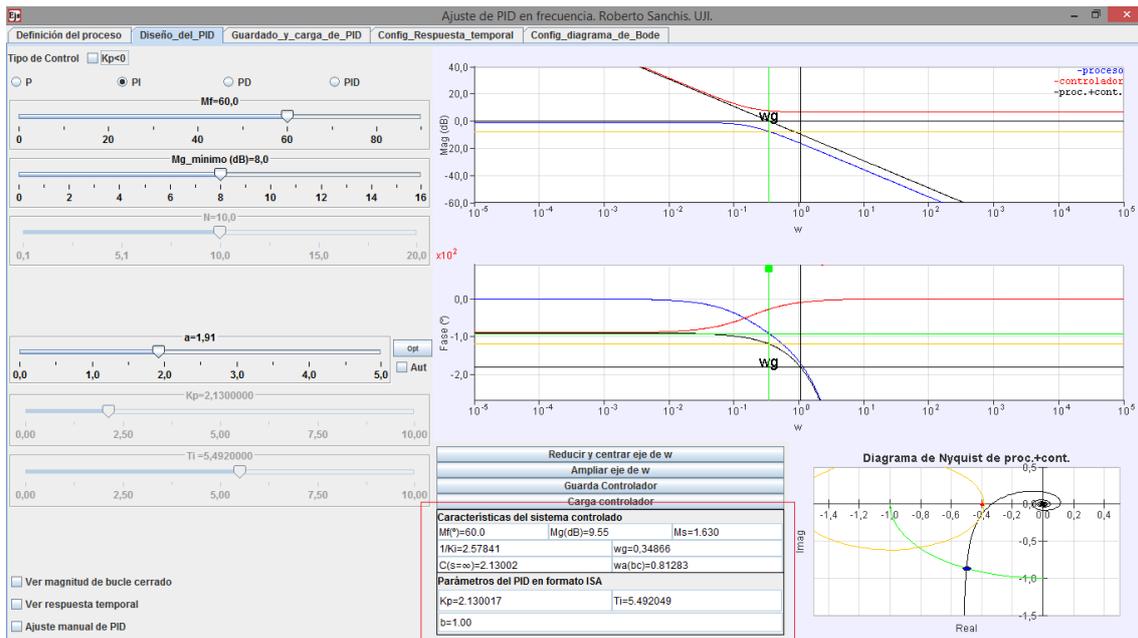


Figura 31. Parámetros del controlador PI calculados con la aplicación Java de diseño de PID.

Una vez introducidos los datos necesarios se obtienen los siguientes parámetros del control PI.

$$K_p = 2,130 ; T_i = 5,492$$

Para comprobar el funcionamiento del controlador es necesario observar, además de la evolución de la respuesta, la evolución de la acción de control. La medición se realiza entre la salida EHRPWM de la capa con rango de 0 a 5V y el amplificador de potencia encargado de aumentar esta señal al rango de 0 a 15V para atacar a las resistencias calefactoras.

Es necesario realizar un filtrado de esta señal EHRPWM para obtener el valor medio y ver la evolución en el tiempo de la acción de control. Para ello se diseña un filtro RC paso bajo que se conecta en paralelo al circuito a modo de sonda de medida para poder captar los datos con el osciloscopio.

El periodo del EHRPWM que queremos filtrar es 1ms. Para hacer un buen filtrado se elige una constante de tiempo del filtro 10 veces mayor ( $\tau_{RC}=10\text{ms}$ ). El controlador se ejecuta con un periodo de 100ms, de manera que con este filtro no perdemos precisión en la medida de la evolución de la acción de control en el tiempo.

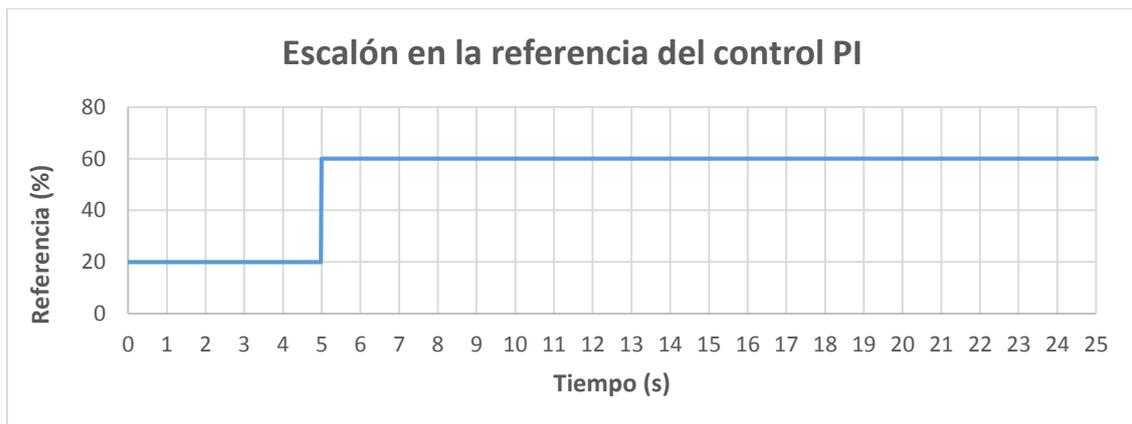
Cálculos del filtro RC paso bajo:

La función de transferencia del filtro RC es la siguiente:

$$\frac{V_o(s)}{V_i(s)} = \frac{1}{R \cdot C \cdot s + 1}$$

La constante de tiempo es por lo tanto  $\tau_{RC}=R \cdot C$ . Se selecciona una resistencia de 100k $\Omega$  para no añadir consumos de corriente que afecten al funcionamiento del circuito, por lo que el condensador debe tener un valor de 0,1 $\mu$ F.

Por último solo queda probar el controlador PI sobre el proceso térmico. Para ello se realiza un escalón en la referencia y una perturbación. A continuación se muestran los resultados de dichas pruebas.



*Gráfico 3. Escalón en la referencia del control PI.*

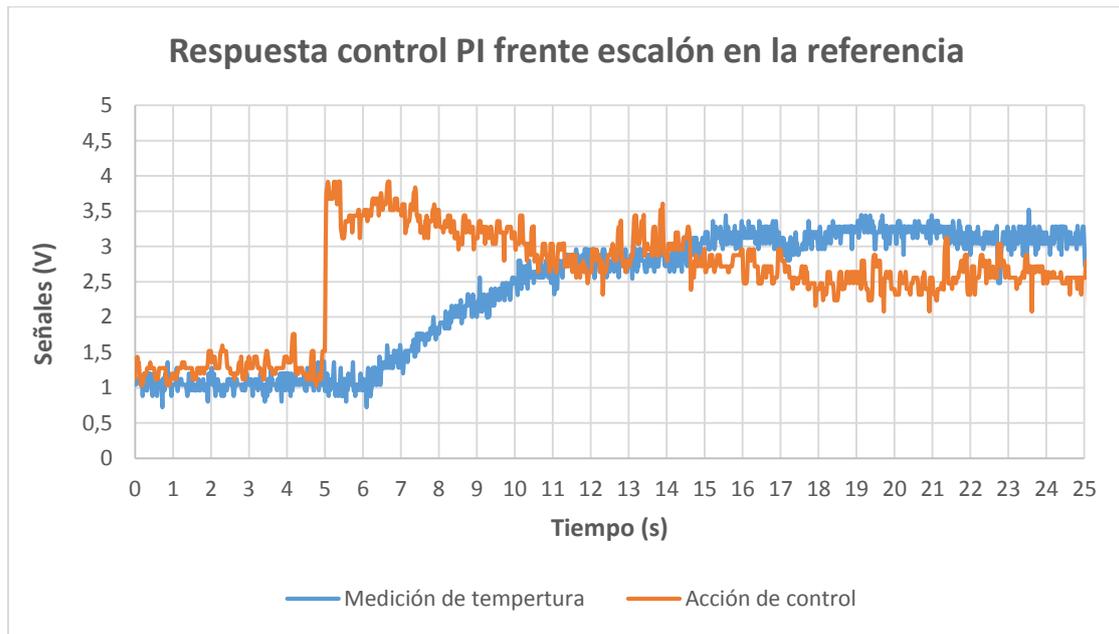


Gráfico 4. Respuesta del control PI frente a escalón en la referencia.

En el segundo 5 se cambia la referencia de 20 a 60%. En el gráfico 4 se observa como la acción de control aumenta en ese instante aumentando el valor de temperatura hasta alcanzar los 3V de medida en el sensor, lo que corresponde con el 60% de su rango de medida.

Estando la referencia en 60% se aumenta la tensión de alimentación del ventilador que crea el flujo de aire constante a través del tubo para crear una perturbación.

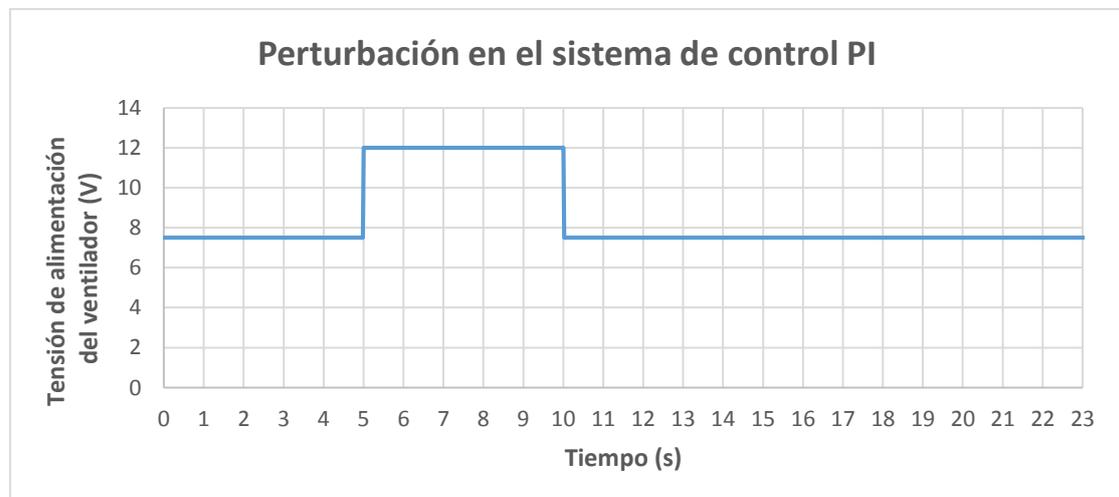
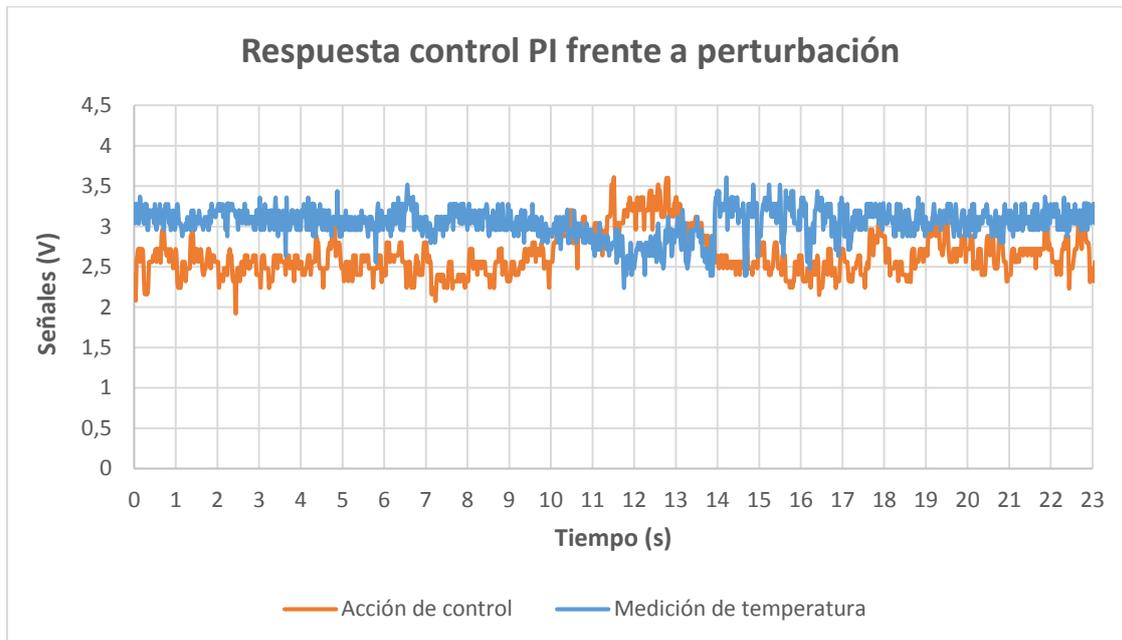


Gráfico 5. Perturbación en el sistema de control PI.



*Gráfico 6. Respuesta de control PI frente a perturbación*

El sistema percibe la perturbación unos segundos más tarde ya que, además de tratarse de un sistema con retardo, la inercia térmica de las masas acentúa este efecto. En el gráfico 6 se observa cómo la acción de control aumenta para compensar la bajada de temperatura creada por el flujo de aire mayor.

## 1.6 Conclusiones

Con la solución adoptada en este proyecto se cumplen los objetivos deseados. Se ha conseguido un sistema de control en red de dos nodos de control fácilmente programable de una forma centralizada. El sistema ofrece versatilidad y flexibilidad para ser reconfigurado y gracias a la capa interfaz permite controlar un amplio abanico de procesos continuos.

También se trata de un sistema fácilmente ampliable, ya que simplemente conectando nuevos dispositivos en la red Ethernet, que soporten los softwares compatibles según la norma, se crean nuevos nodos de control en el sistema.

Llevando a cabo este proyecto también se ha demostrado la gran utilidad y buenas prestaciones que ofrecen el estándar IEC-61499 junto con las herramientas de software desarrolladas hasta la fecha para programar algoritmos de control de forma sencilla, rápida y eficaz.

Sí que cabe comentar que con la utilización de la herramienta de desarrollo 4DIAC-IDE, aunque permite desarrollar sistemas de control muy capaces y

variados, se tiene el inconveniente de no contar con un entorno HMI desarrollado, razón por la que ha sido necesario utilizar un osciloscopio y algún circuito auxiliar para comprobar el funcionamiento del controlador. Hay que tener en cuenta que tanto la forma de implementar modelos según el estándar IEC-61499 como las herramientas de software actualmente existentes para este fin, están en proceso de mejora. Otras herramientas de pago, a diferencia de 4DIAC-IDE, sí que disponen de un entorno HMI más desarrollado, como es el caso de ntxOne.

Se espera que en un futuro las diferentes herramientas de modelado e implementación de estos sistemas ofrezcan cada vez más prestaciones, cosa de la que se puede estar seguro ya que con este proyecto y muchos otros se demuestra el gran potencial que ofrece este estándar.

## 1.7 Bibliografía

- [1] Robert Lewis, R. 2001. "Modelling control systems using IEC 61499".
- [2] Mark A. Yoder and Jason Kridner, R. 2015. "BeagleBone Cookbook".
- [3] 4DIAC Consortium. 2000. "4DIAC - Framework for Distributed Industrial Automation and Control, IEC-61499 Implementation for Distributed Devices of the Next Generation".  
Disponible en: <http://www.eclipse.org/4diac/>
- [4] HOLOBLOC, Inc. "Resources for the New Generation of Automation and Control Software".  
Disponible en: <http://www.holobloc.com>
- [5] The BeagleBoard Foundation. "BeagleBone Foundation webpage"  
Disponible en: <http://beagleboard.org>
- [6] The BeagleBoard Foundation. "BeagleBone Black wiki"  
Disponible en: <http://elinux.org/Beagleboard:BeagleBoneBlack/>

- [7] Apuntes de la asignatura EE1018 Electrónica
  
- [8] Apuntes de la asignatura EE1035 Instrumentación, Medida y Tratamiento de la Señal.
  
- [9] Apuntes de la asignatura EE1023 Sistemas Automáticos.
  
- [10] Apuntes de la asignatura EE1038 Regulación Automática.

## **2. ANEXOS**

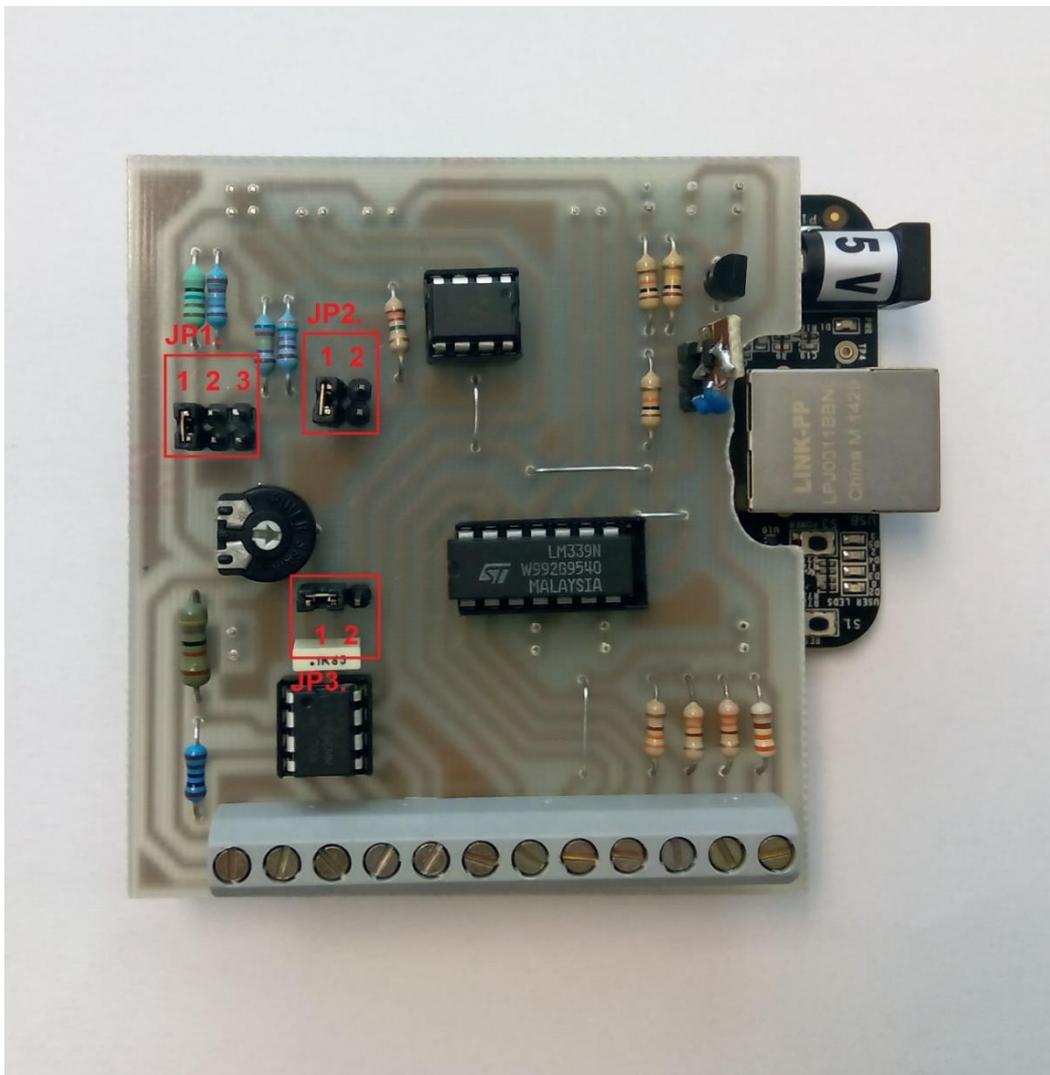


## 2. ANEXOS

### 2.1 Documentos de dispositivos desarrollados para el sistema

#### 2.1.1 Características e instrucciones de uso de capa interfaz de entradas analógicas y salidas EHRPWM.

Este documento tiene la finalidad de describir las características y condiciones de uso de la capa interfaz para el control de procesos continuos desarrollada en este proyecto.



*Figura 32. Capa interfaz de entradas analógicas y salidas EHRPWM.*

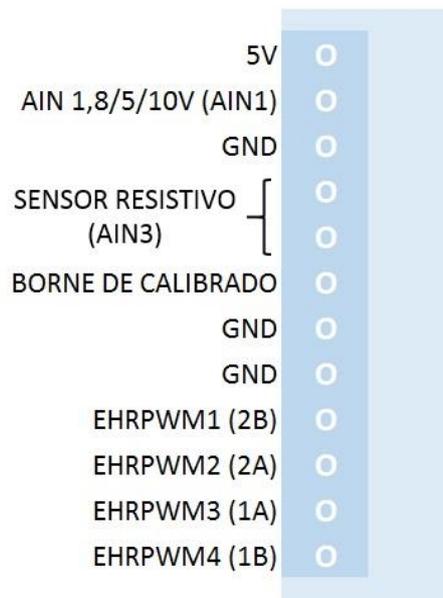


Figura 33. Detalle de conexiones en los terminales de la capa.

#### Resumen de características:

- 1 Entrada analógica configurable 1,8/5/10V.
- 1 Entrada analógica con conexión directa de sensor resistivo.
- 4 Salidas EHRPWM de 5V.
- Toma de 5V (900mA).
- Protección de mini-PC Beaglebone Black en estado de desconexión.

#### Entrada analógica configurable 1,8/5/10V:

El rango de tensión de entrada es seleccionable mediante los jumpers JP1 y JP2. Las distintas configuraciones y sus características se muestran en la siguiente tabla:

Rango de entrada	JP1	JP2	R entrada	I entrada
10V	1	1	9,84k $\Omega$	-
5V	2	2	9,91k $\Omega$	-
1,8V	3	-	-	2 $\mu$ A

Tabla 5. Configuración de jumpers para entrada analógica configurable.

Para la programación del dispositivo se ha de saber que esta entrada se corresponde con la entrada analógica AIN1 de la Beaglebone Black.

#### Entrada analógica con conexión directa de sensor resistivo

Esta entrada permite la conexión directa de sensores resistivos con un valor máximo de  $17k\Omega$ . Se trata de un amplificador no inversor que da una salida de tensión directamente proporcional al valor de la resistencia del sensor con un valor de offset de  $0,1V$ . Para ajustar el rango de medida se dispone de un borne de calibrado y de un trimmer que permite ajustar el rango de medida de manera que, con el valor máximo del sensor, en este borne se midan  $1,8V$ .

El circuito cuenta con un jumper (JP3) que en la posición 1 cortocircuita los bornes de conexión del sensor llevando el valor de salida al mínimo de  $0,1V$ , ya que la resistencia pasa a valer  $0\Omega$ . Además mientras el jumper no esté en la posición 2 la salida del amplificador no inversor no está conectada a la entrada analógica AIN3, de manera que aun sobrepasándose el valor de  $1,8V$  no se dañaría esta entrada analógica.

**PRECAUCIÓN:** Nunca se debe superar el valor de  $1,8V$  en el borne de calibrado mientras el jumper JP3 esté en la posición 2. El valor de la resistencia del sensor junto con el del trimmer determinan la ganancia de la salida del amplificador al pin AIN3 de la Beaglebone Black, por lo que con valores de resistencia del sensor inferiores a  $17k\Omega$  es posible superar los  $1,8V$  si el trimmer no está ajustado a su valor máximo de  $1k\Omega$ .

Los pasos a seguir para la utilización de esta entrada son los siguientes:

1. Retirar el jumper JP3. Es decir, no conectar ni en la posición 1 ni en la 2.
2. Conectar un multímetro al borne de calibrado para medir la tensión.
3. Conectar el sensor resistivo en sus bornes.
4. Con el valor máximo de resistencia del sensor, ajustar con el trimmer la ganancia del amplificador hasta alcanzar un valor próximo a  $1,8V$ .
5. Una vez se esté seguro de que no se vaya a superar este valor, poner el jumper JP3 en la posición 2 para conectar la salida del amplificador a la entrada analógica AIN3 de la Beaglebone Black.

#### Salidas EHRPWM de 5V

La capa dispone de 4 salidas EHRPWM de 5V que se corresponden con las de la Beaglebone Black según se indica en la tabla:

Salida EHRPWM capa	Salida EHRPWM BBB	R salida
EHRPWM1	EHRPWM2B	330Ω
EHRPWM2	EHRPWM2A	330Ω
EHRPWM3	EHRPWM1A	330Ω
EHRPWM4	EHRPWM1B	330Ω

*Tabla 6. Correspondencia y características de las salidas EHRPWM.*

#### Lista de componentes:

- 2 Resistencias 10kΩ ±5% 1/4W.
- 1 Resistencia 100kΩ ±5% 1/4W.
- 1 Resistencia 2,2MΩ ±5% 1/4W.
- 4 Resistencias 330Ω ±5% 1/4W.
- 1 Resistencia 8,06kΩ ±1% 1/4W.
- 1 Resistencia 1,78kΩ ±1% 1/4W.
- 1 Resistencia 6,37kΩ ±1% 1/4W.
- 1 Resistencia 3,57kΩ ±1% 1/4W.
- 1 Resistencia 47kΩ ±1% 1/4W.
- 1 Resistencia 1kΩ ±1% 1/4W.
- 1 Trimmer 1kΩ ±5% 1/2W.
- 1 Condensador 0,1μF ±5% PET 63 V.
- 1 Transistor BJT BC548C.
- 1 Transistor MOSFET RZR040P01.
- 2 Amplificadores operacionales MCP6021.
- 1 Comparador LM339N.
- 1 Bloque de terminales de 12 contactos AWG 24-12.
- 37 Conectores placa a placa 6mm de largo, 2,54mm de ancho de paso.
- 3 Jumpers de 2 posiciones y ancho de paso 2,54mm.

#### 2.1.2 Características e instrucciones de uso de sensor de temperatura analógico.

En este documento se especifican las características y condiciones de uso del sensor de temperatura analógico desarrollado en este proyecto.

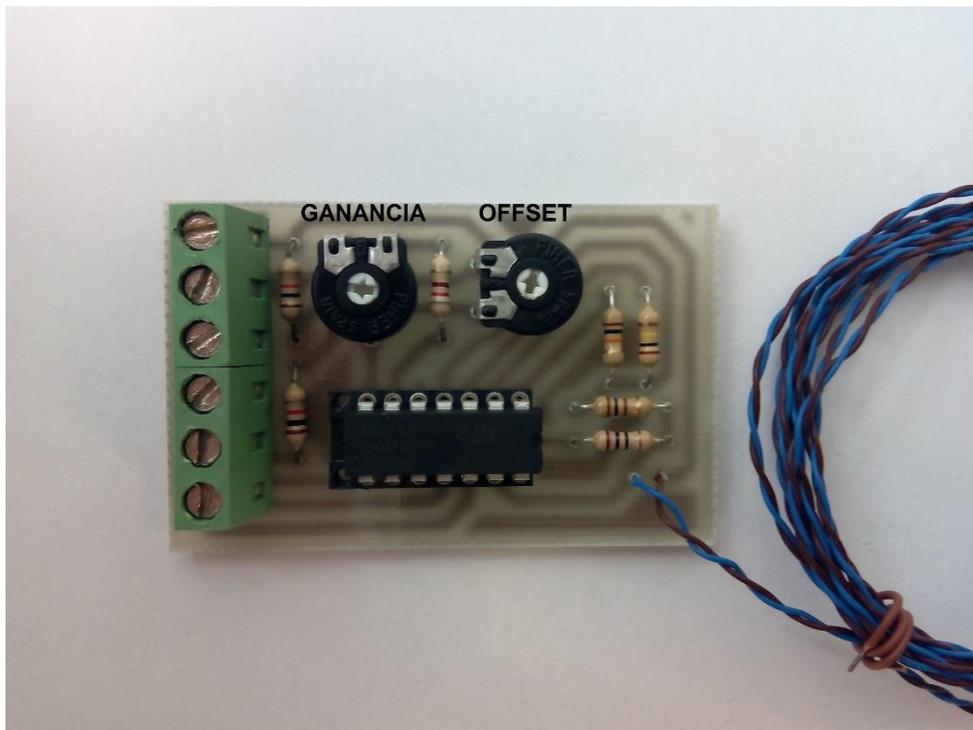


Figura 34. Sensor de temperatura analógico.

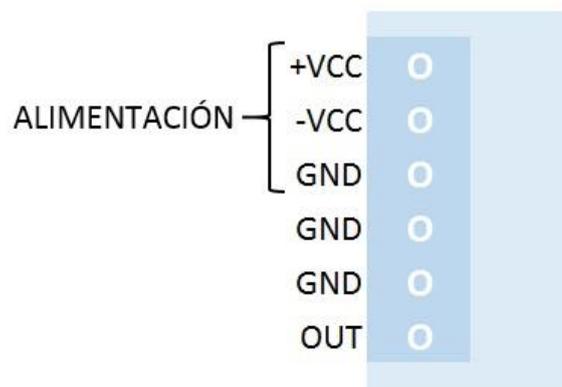


Figura 35. Detalle de conexiones en los terminales del sensor de temperatura analógico.

Resumen de características:

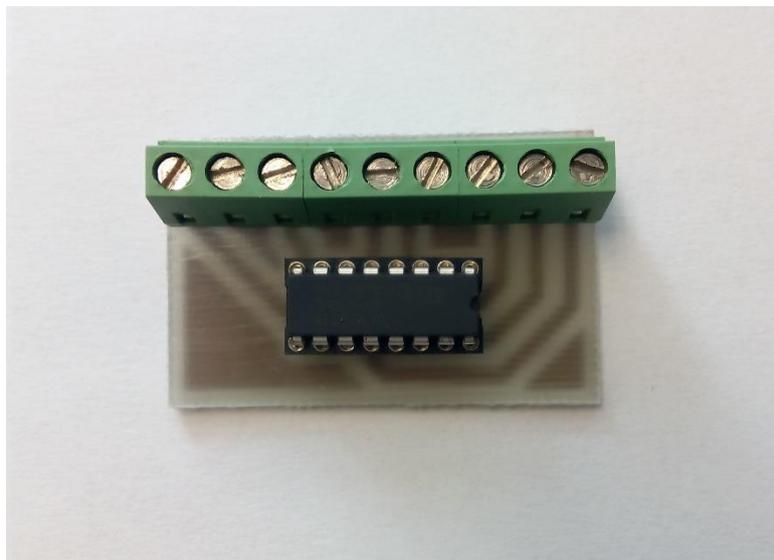
- Rango de temperatura diferencial de 30°C aproximadamente.
- Tensión de alimentación simétrica entre  $\pm 3$  y  $\pm 30$ V.
- Ajuste de ganancia y offset para rango de tensión de salida de 0 a 5V.
- Corriente máxima de salida 40mA.

Lista de componentes:

- 1 Resistencia  $10\Omega \pm 5\%$  1/4W.
- 1 Resistencia  $100\Omega \pm 5\%$  1/4W.
- 1 Resistencia  $820\Omega \pm 5\%$  1/4W.
- 2 Resistencias  $1k\Omega \pm 5\%$  1/4W.
- 1 Resistencia  $10k\Omega \pm 5\%$  1/4W.
- 1 Resistencia  $100k\Omega \pm 5\%$  1/4W.
- 1 Trimmer  $1k\Omega \pm 5\%$  1/2W.
- 1 Amplificador operacional LM324N.
- 1 Termopar tipo K (Ni/Cr) Cable de 1m.
- 2 Bloques de terminales de 3 contactos AWG 24-12.

2.1.3 Características e instrucciones de uso de amplificador de potencia.

En este documento se especifican las características y condiciones de uso del amplificador de potencia desarrollado en este proyecto.



*Figura 36. Amplificador de potencia.*

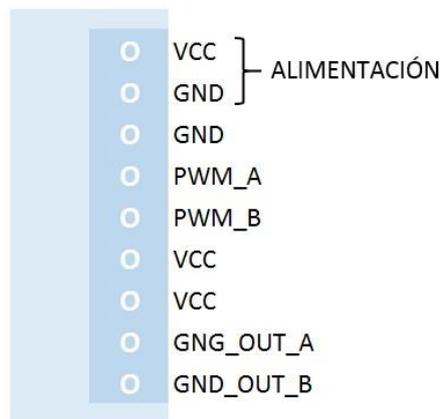


Figura 37. Detalle de conexiones en los terminales del amplificador de potencia.

Se trata de un amplificador de dos canales para amplificar la potencia de señales PWM de 5V aumentando la tensión y la corriente en la salida. Las cargas se conectan entre las tomas VCC disponibles y sus respectivas masas para cada canal.

#### Resumen de características:

- Amplificador de potencia de dos canales para señales PWM.
- Señales PWM de entrada de 5V TTL o CMOS.
- VCC máxima 50V.
- I canal A 2A.
- I canal B 1,5A.

#### Lista de componentes:

- 1 Cadena de amplificadores Darlington ULN2003A.
- 3 Bloques de terminales de 3 contactos AWG 24-12

## 2.2 Hojas de características de los componentes utilizados

1. Hoja de características del BJT BC548C disponible en:

<http://pdf1.alldatasheet.es/datasheet-pdf/view/2897/MOTOROLA/BC548C.html>

2. Hoja de características del MOSFET RZR040P01 disponible en:

<http://pdf1.alldatasheet.es/datasheet-pdf/view/338891/ROHM/RZR040P01.html>

3. Hoja de características del amplificador operacional MCP6021 disponible en:

<http://pdf1.alldatasheet.es/datasheet-pdf/view/99098/MICROCHIP/MCP6021.html>

4. Hoja de características del comparador LM339N disponible en:

<http://pdf1.alldatasheet.es/datasheet-pdf/view/22764/STMICROELECTRONICS/LM339N.html>

5. Hoja de características del amplificador operacional LM324N disponible en:

<http://pdf1.alldatasheet.es/datasheet-pdf/view/22756/STMICROELECTRONICS/LM324N.html>

6. Hoja de características de la cadena de amplificadores Darlington ULN2003A disponible en:

<http://pdf1.alldatasheet.es/datasheet-pdf/view/25566/STMICROELECTRONICS/ULN2003A.html>

# **3. PLANOS**



### **3. PLANOS**

P01. Esquemático de capa interfaz de entradas analógicas y salidas EHRPWM.

P02. Esquemático de sensor de temperatura analógico.

P03. Esquemático de amplificador de potencia.

P04. Diseño de PCB de capa interfaz de entradas analógicas y salidas EHRPWM.

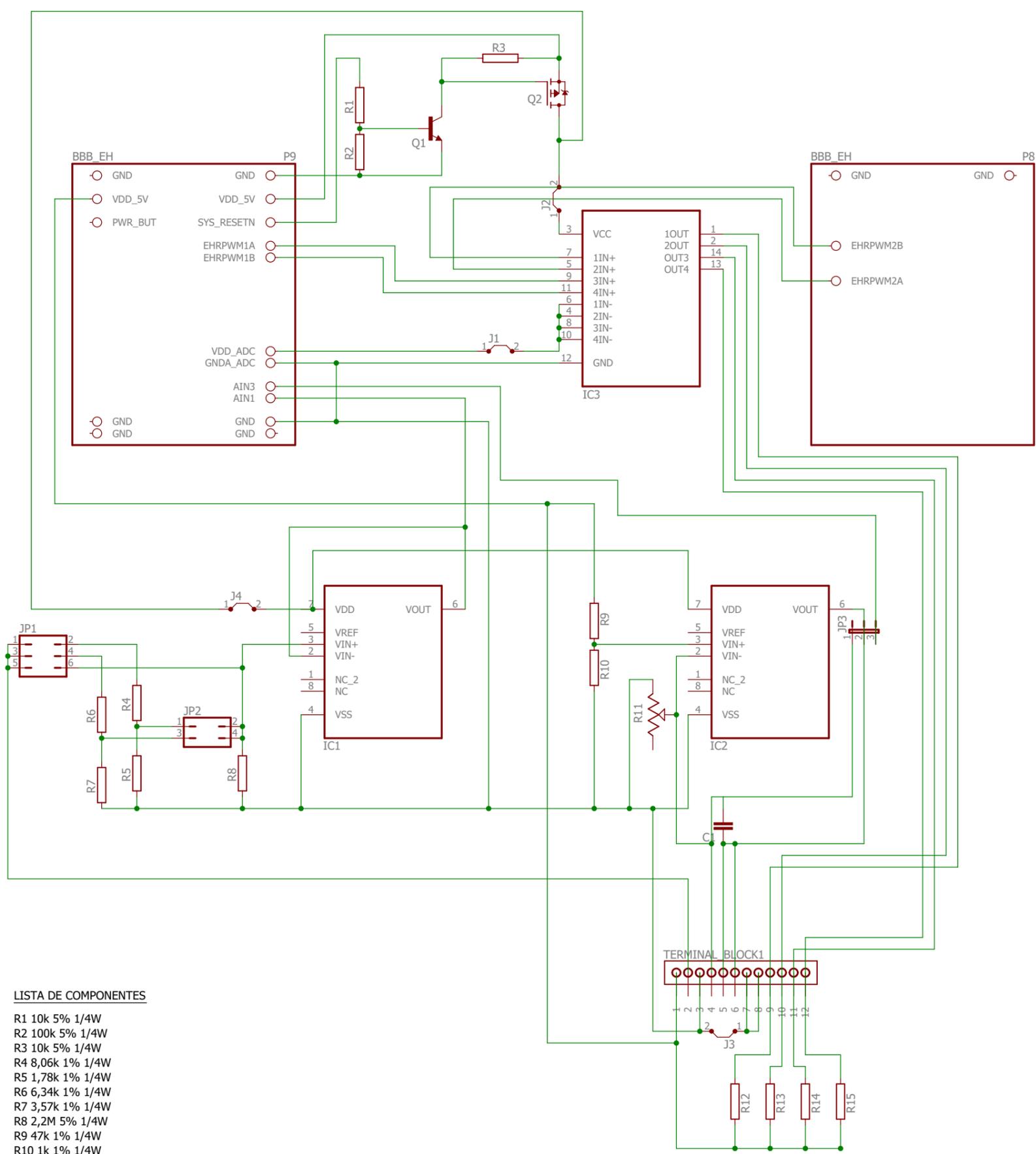
P05. Diseño de PCB de sensor de temperatura analógico.

P06. Diseño de PCB de amplificador de potencia.

P07. Detalle de PCB con componentes de capa interfaz de entradas analógicas y salidas EHRPWM.

P08. Detalle de PCB con componentes de sensor de temperatura analógico.

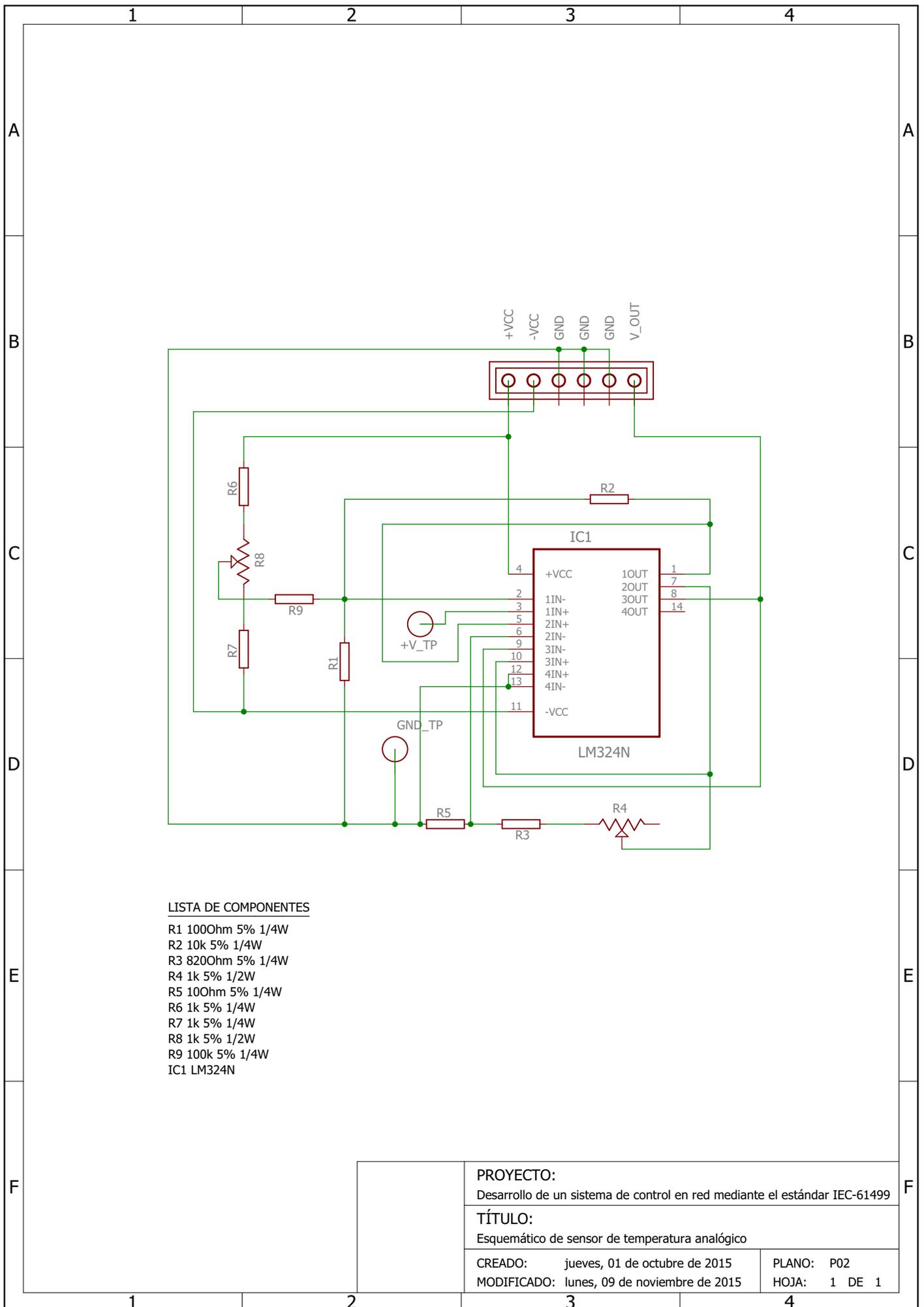
P09. Detalle de PCB con componentes de amplificador de potencia.



**LISTA DE COMPONENTES**

- R1 10k 5% 1/4W
- R2 100k 5% 1/4W
- R3 10k 5% 1/4W
- R4 8,06k 1% 1/4W
- R5 1,78k 1% 1/4W
- R6 6,34k 1% 1/4W
- R7 3,57k 1% 1/4W
- R8 2,2M 5% 1/4W
- R9 47k 1% 1/4W
- R10 1k 1% 1/4W
- R11 1k 5% 1/2W
- R12 3300hm 5% 1/4W
- R13 3300hm 5% 1/4W
- R14 3300hm 5% 1/4W
- R15 3300hm 5% 1/4W
- C1 0,1uF 5% 63V
- Q1 BC548C
- Q2 RZR040P01
- IC1 MCP6021
- IC2 MCP6021
- IC3 LM339N

<b>PROYECTO:</b> Desarrollo de un sistema de control en red mediante el estándar IEC-61499	
<b>TÍTULO:</b> Esquemático de capa interfaz de entradas analógicas y salidas EHRPWM	
CREADO: miércoles, 16 de septiembre de 2015	PLANO: P01
MODIFICADO: lunes, 09 de noviembre de 2015	HOJA: 1 DE: 1



**LISTA DE COMPONENTES**

- R1 100Ohm 5% 1/4W
- R2 10k 5% 1/4W
- R3 820Ohm 5% 1/4W
- R4 1k 5% 1/2W
- R5 10Ohm 5% 1/4W
- R6 1k 5% 1/4W
- R7 1k 5% 1/4W
- R8 1k 5% 1/2W
- R9 100k 5% 1/4W
- IC1 LM324N

**PROYECTO:**

Desarrollo de un sistema de control en red mediante el estándar IEC-61499

**TÍTULO:**

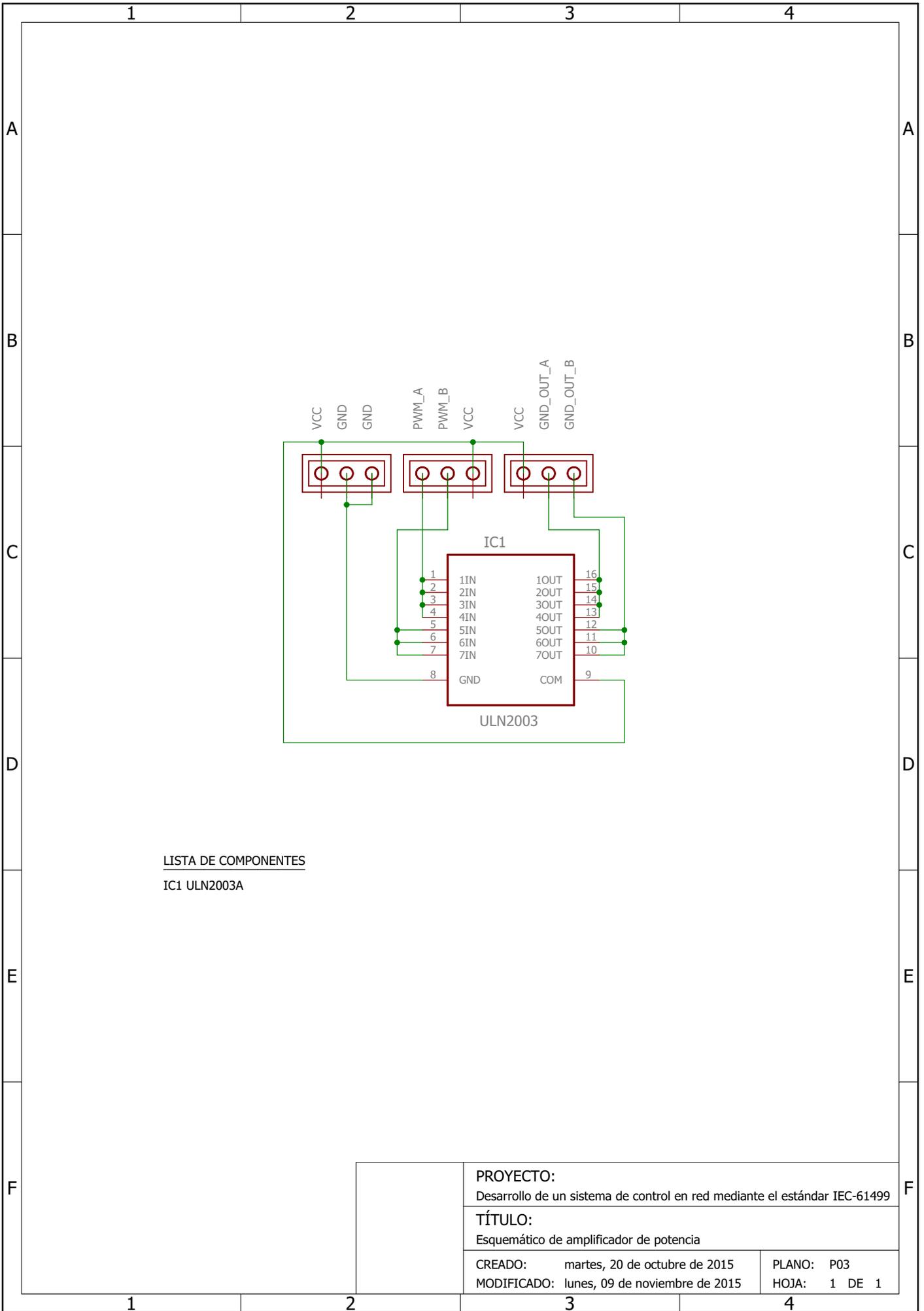
Esquemático de sensor de temperatura analógico

CREADO: jueves, 01 de octubre de 2015

PLANO: P02

MODIFICADO: lunes, 09 de noviembre de 2015

HOJA: 1 DE 1



LISTA DE COMPONENTES

IC1 ULN2003A

**PROYECTO:**

Desarrollo de un sistema de control en red mediante el estándar IEC-61499

**TÍTULO:**

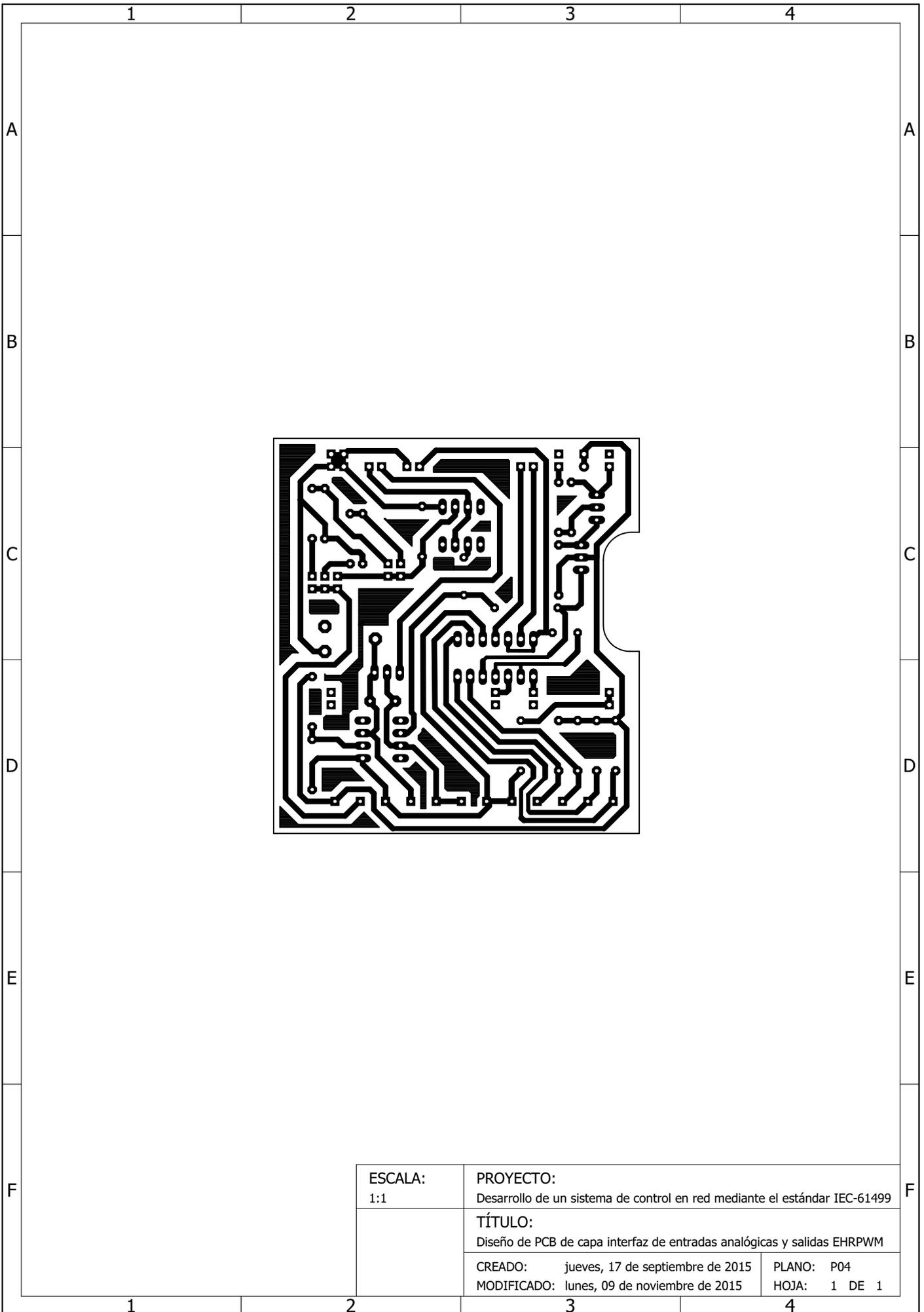
Esquemático de amplificador de potencia

CREADO: martes, 20 de octubre de 2015

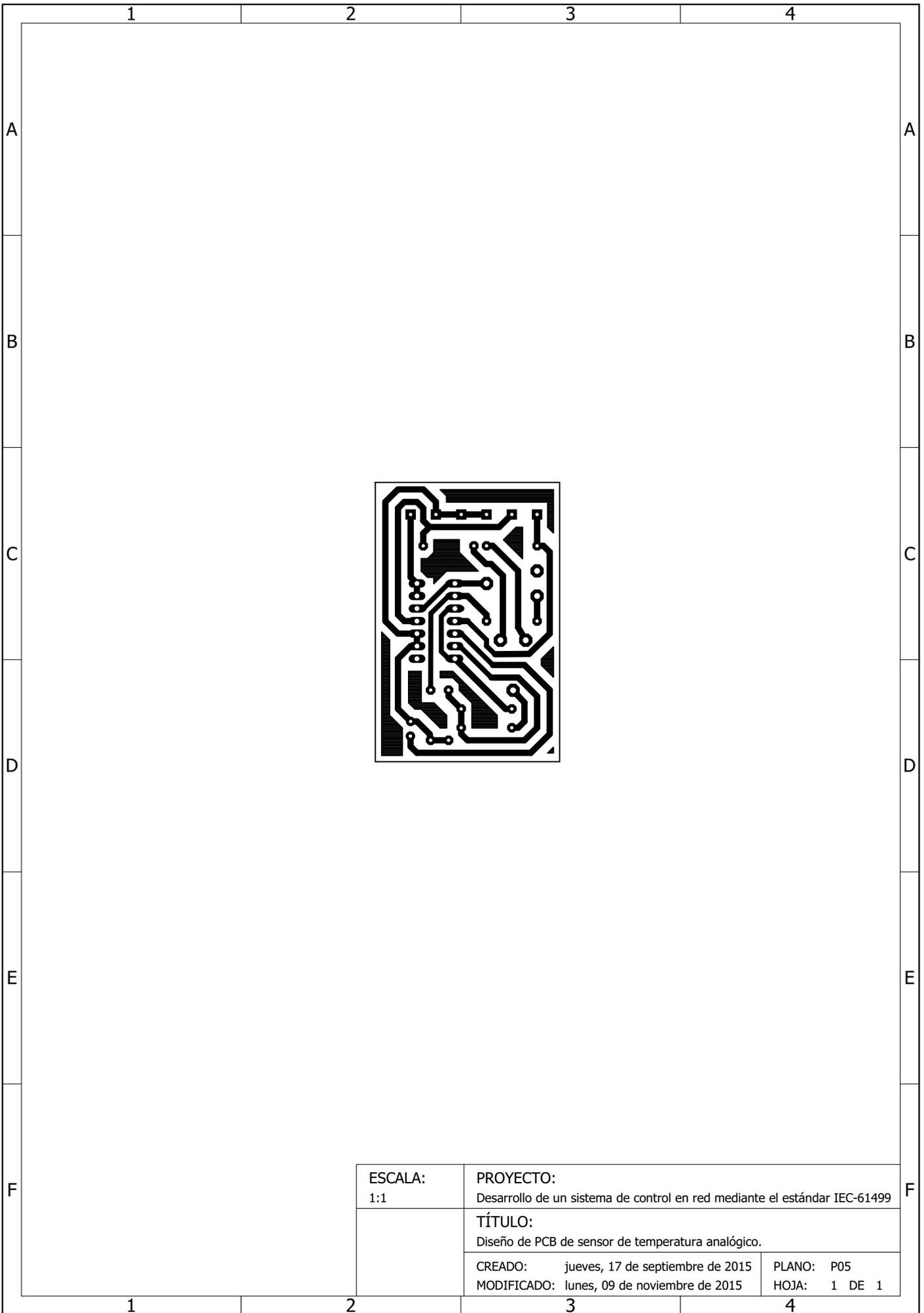
PLANO: P03

MODIFICADO: lunes, 09 de noviembre de 2015

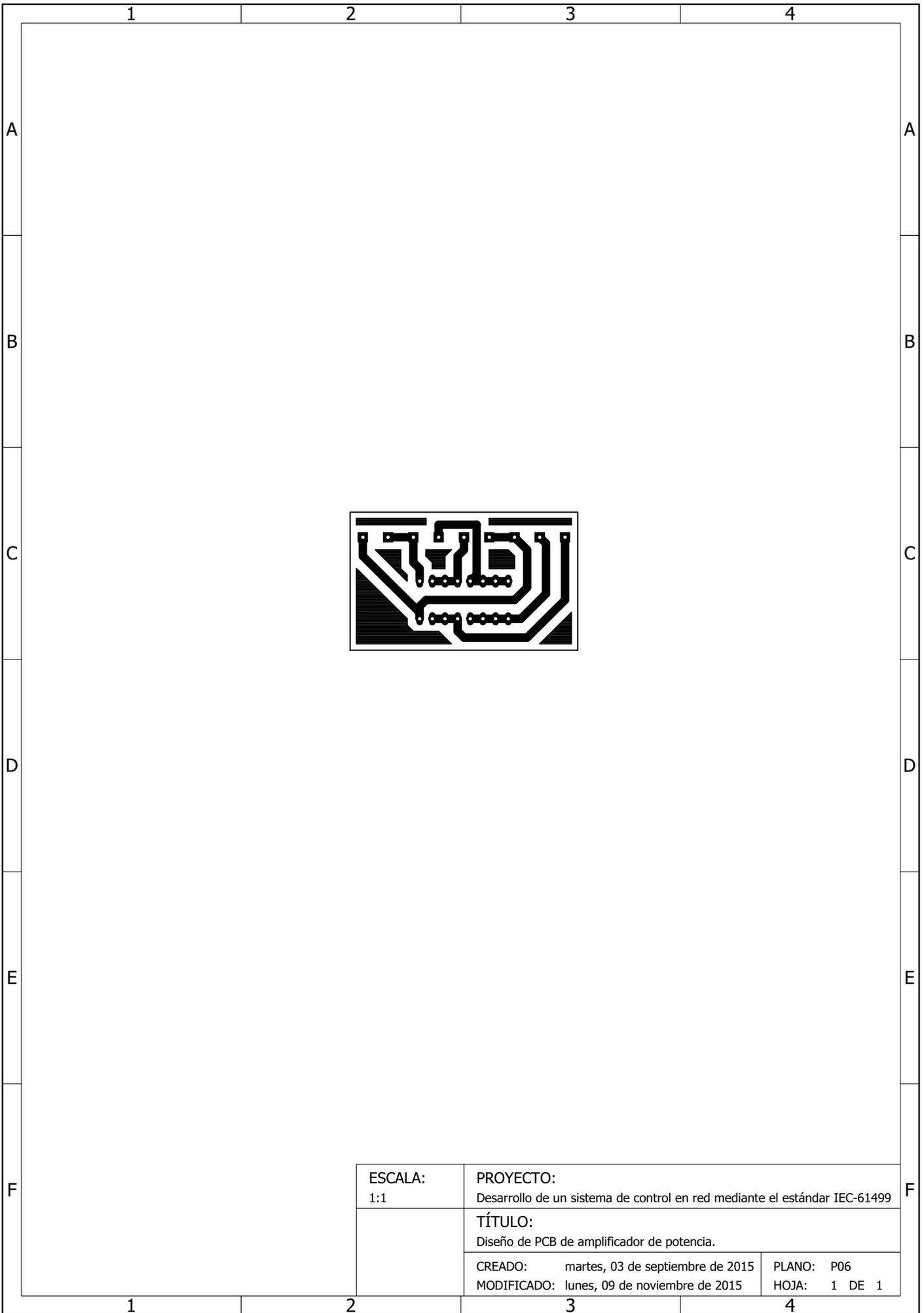
HOJA: 1 DE 1

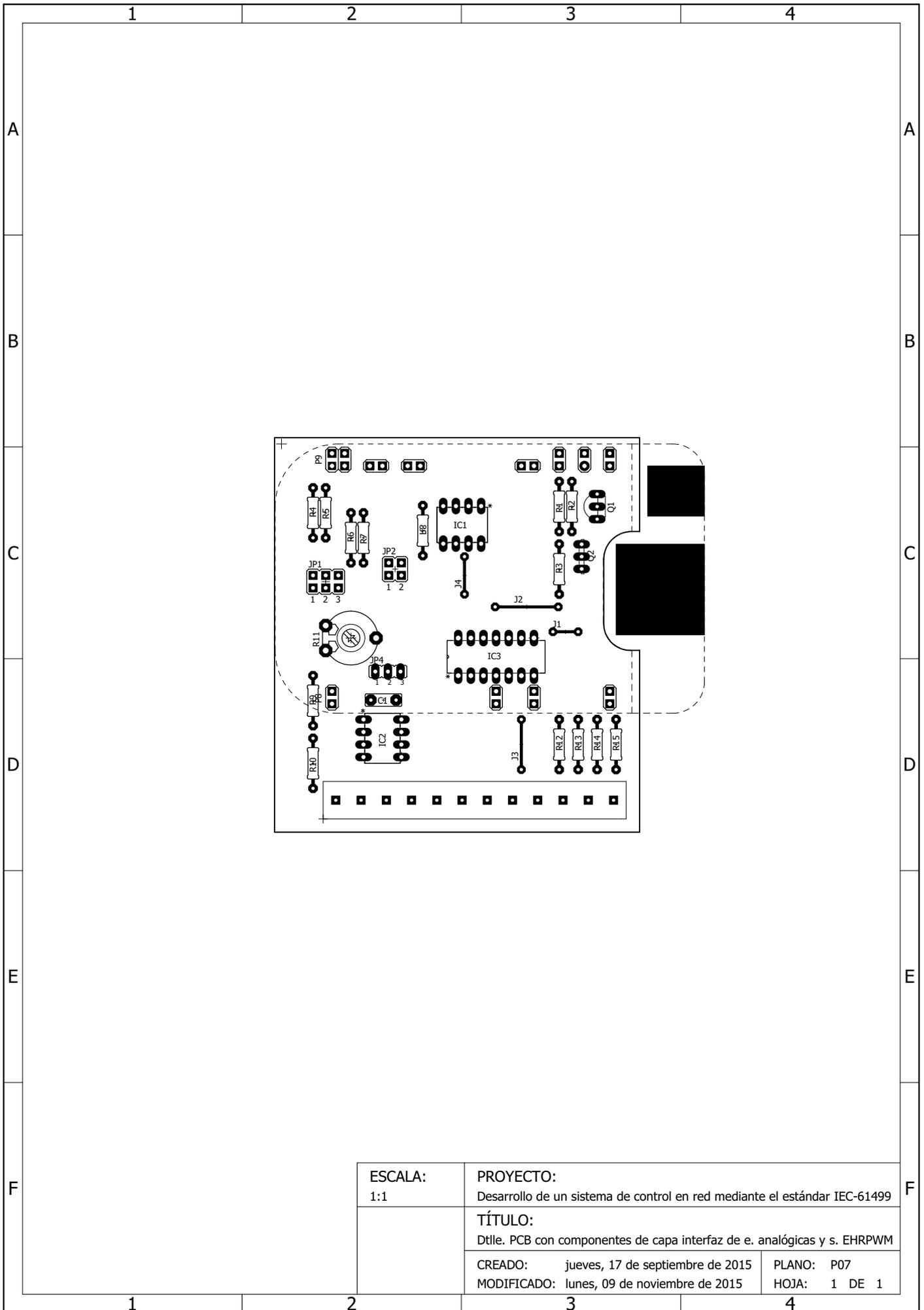


ESCALA: 1:1	PROYECTO: Desarrollo de un sistema de control en red mediante el estándar IEC-61499	
	TÍTULO: Diseño de PCB de capa interfaz de entradas analógicas y salidas EHRPWM	
	CREADO:    jueves, 17 de septiembre de 2015	PLANO:    P04
	MODIFICADO: lunes, 09 de noviembre de 2015	HOJA:    1 DE 1

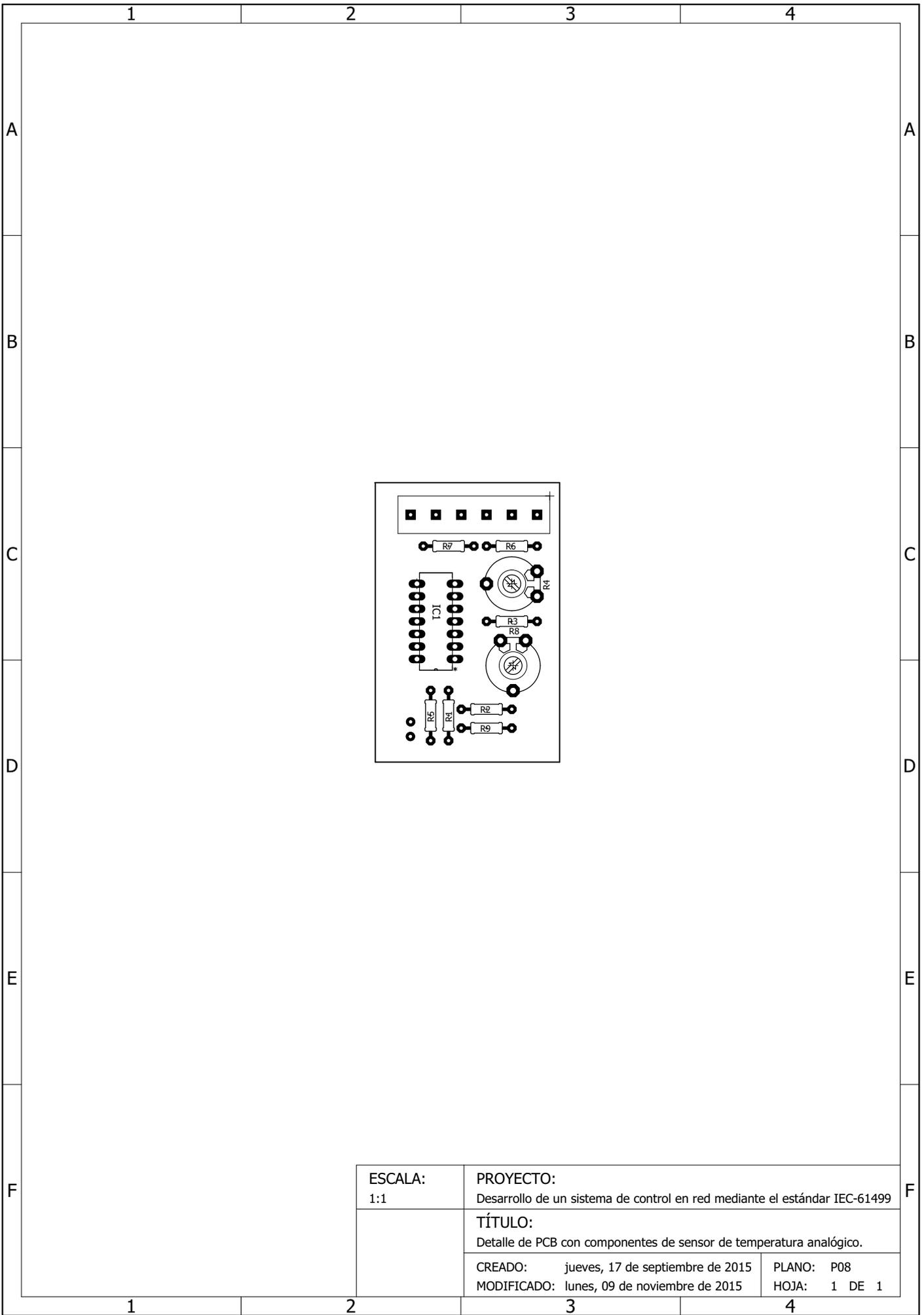


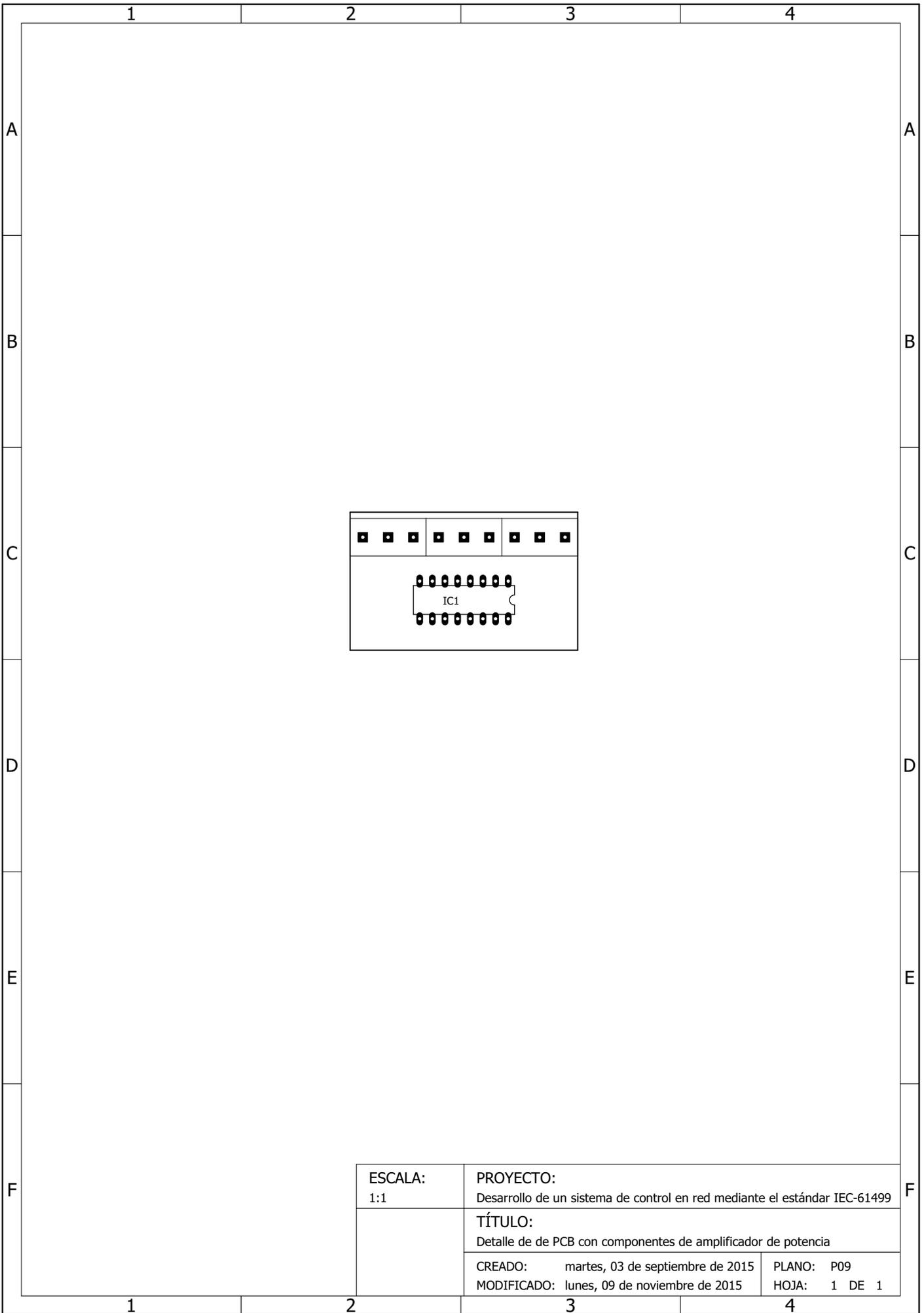
ESCALA: 1:1	PROYECTO: Desarrollo de un sistema de control en red mediante el estándar IEC-61499
	TÍTULO: Diseño de PCB de sensor de temperatura analógico.
	CREADO:    jueves, 17 de septiembre de 2015
	MODIFICADO: lunes, 09 de noviembre de 2015
	PLANO:   P05
	HOJA:    1 DE 1





ESCALA: 1:1	PROYECTO: Desarrollo de un sistema de control en red mediante el estándar IEC-61499	PLANO: P07 HOJA: 1 DE 1
	TÍTULO: DtIle. PCB con componentes de capa interfaz de e. analógicas y s. EHRPWM	
	CREADO: jueves, 17 de septiembre de 2015	
	MODIFICADO: lunes, 09 de noviembre de 2015	





# **4. PRESUPUESTO**



## 4. PRESUPUESTO

En este apartado se estudia el coste que supone la ejecución del presente proyecto. Para ello se calcula por separado el coste de la implementación del sistema de control en red y el sistema térmico en concepto de materiales utilizados. Por otra parte se estiman los gastos asociados a las horas que se han empleado para el desarrollo del mismo.

En primer lugar se estima el coste del sistema de control en red.

<b>CAPÍTULO 1: Sistema de control en red</b>			
<b>Mini-PC</b>			
Cantidad	Referencias	Descripción	Precio/Ud.
2	BBB1, BBB2	Beaglebone Black	45,74 €
2		Fuente de alimentación 5V para BBB	7,50 €
<b>Subtotal</b>			<b>106,48 €</b>
<b>Red Ethernet</b>			
Cantidad	Referencias	Descripción	Precio/Ud.
1		Switch Ethernet	14,87 €
3		Cable de Red UTP RJ45 Cat 6 2m	2,44 €
<b>Subtotal</b>			<b>22,19 €</b>
<b>Resistencias</b>			
Cantidad	Referencias	Descripción	Precio/Ud.
8	P01: R12, R13, R14, R15	330Ω ±5% 1/4W	0,02 €
2	P01: R10	1kΩ ±1% 1/4W	0,05 €
2	P01: R5	1,78kΩ ±1% 1/4W	0,05 €
2	P01: R7	3,57kΩ ±1% 1/4W	0,05 €
2	P01: R6	6,37kΩ ±1% 1/4W	0,05 €
2	P01: R4	8,06kΩ ±1% 1/4W	0,05 €
4	P01: R1, R3	10kΩ ±5% 1/4W	0,02 €
2	P01: R9	47kΩ ±1% 1/4W	0,05 €
4	P01: R2	100kΩ ±5% 1/4W	0,02 €
2	P01: R8	2,2MΩ ±5% 1/4W	0,02 €
<b>Subtotal</b>			<b>0,96 €</b>
<b>Potenciómetros Trimmer</b>			
Cantidad	Referencias	Descripción	Precio/Ud.
2	P01: R11	1kΩ ±5% 1/2W	1,55 €
<b>Subtotal</b>			<b>3,10 €</b>

<b>Condensadores</b>			
Cantidad	Referencias	Descripción	Precio/Ud.
2	P01: C1	0.1 $\mu$ F $\pm$ 5% PET 63 V	0,09 €
<b>Subtotal</b>			<b>0,18 €</b>
<b>Transistores</b>			
Cantidad	Referencias	Descripción	Precio/Ud.
2	P01: Q1	BC548C BJT NPN	0,35 €
2	P01: Q2	RZR040P01TL MOSFET Canal P	0,53 €
<b>Subtotal</b>			<b>1,76 €</b>
<b>Circuitos integrados</b>			
Cantidad	Referencias	Descripción	Precio/Ud.
4	P01: IC1, IC2	MCP6021-E/P Amp. Operacional	1,04 €
2	P01: IC3	LM339N Comparador Analógico	0,31 €
<b>Subtotal</b>			<b>4,78 €</b>
<b>Otros</b>			
Cantidad	Referencias	Descripción	Precio/Ud.
2		Bloque terminales 12xAWG 24-12	8,88 €
2		Conect. Placa a Placax38, 6mm, 2,54mm	2,40 €
2		Placa baquelita cobre 75x100mm	1,40 €
6	P01: JP1, JP2, JP3	Jumper, 2 posiciones, 2,54 mm	0,12 €
<b>Subtotal</b>			<b>26,08 €</b>
<b>Subtotal CAPÍTULO 1</b>			<b>165,53 €</b>

*Tabla 7. Presupuesto del sistema de control en red.*

Respecto al coste del sistema de control en red cabe comentar que no se han incluido costes relativos a licencias de software. Hay que recordar que todos los softwares utilizados en el desarrollo del proyecto son de condigo libre, por lo que no supone ningún coste adicional.

A continuación se muestran los costes derivados del desarrollo de instrumentación para el sistema térmico con retardo utilizado para implementar el control en este proyecto.

## **CAPÍTULO 2: Instrumentación para sistema térmico con retardo**

### **Resistencias**

Cantidad	Referencias	Descripción	Precio/Ud.
1	P02: R5	10 $\Omega$ $\pm$ 5% 1/4W	0,02 €
1	P02: R1	100 $\Omega$ $\pm$ 5% 1/4W	0,02 €
1	P02: R3	820 $\Omega$ $\pm$ 5% 1/4W	0,02 €

2	P02: R6, R7	1k $\Omega$ $\pm$ 5% 1/4W	0,02 €
1	P02: R2	10k $\Omega$ $\pm$ 5% 1/4W	0,02 €
1	P02: R9	100k $\Omega$ $\pm$ 5% 1/4W	0,02 €
<b>Subtotal</b>			<b>0,14 €</b>
<b>Potenciómetros Trimmer</b>			
Cantidad	Referencias	Descripción	Precio/Ud.
2	P02: R4, R8	1k $\Omega$ $\pm$ 5% 1/2W	1,55 €
<b>Subtotal</b>			<b>3,10 €</b>
<b>Circuitos integrados</b>			
Cantidad	Referencias	Descripción	Precio/Ud.
1	P02: IC1	LM324N Amp. Operacional	0,41 €
1	P03: IC1	ULN2003A Cadena de Darlington	0,62 €
<b>Subtotal</b>			<b>1,03 €</b>
<b>Termopares</b>			
Cantidad	Referencias	Descripción	Precio/Ud.
1		Termopar tipo K (Ni/Cr) Cable de 1m	5,64 €
<b>Subtotal</b>			<b>5,64 €</b>
<b>Otros</b>			
Cantidad	Referencias	Descripción	Precio/Ud.
5		Bloque terminales 3xAWG 24-12	2,33 €
1		Placa baquelita cobre 75x100mm	1,40 €
<b>Subtotal</b>			<b>13,05 €</b>
<b>Subtotal CAPÍTULO 2</b>			<b>22,96 €</b>

Tabla 8. Presupuesto de instrumentación para el sistema térmico con retardo.

Por último al coste de los materiales es necesario añadirle el coste de la mano de obra, que en este caso incluirá el salario del ingeniero encargado de la realización del proyecto.

Para la realización de este presupuesto se ha considerado que el coste neto que supone el trabajo del ingeniero es de 30€/h. También se ha considerado el cómputo total de horas que supone el desarrollo de este proyecto por parte de un ingeniero experimentado. En la siguiente tabla se detallan las horas necesarias para la realización de cada una de las partes del proyecto.

<b>CAPÍTULO 3: Costes de ingeniería</b>		
Horas	Concepto	Coste
28	Documentación	840,00 €
32	Diseño de instrumentación	960,00 €

18	Montaje de instrumentación	540,00 €
6	Configuración de dispositivos	180,00 €
20	Desarrollo de algoritmos de control	600,00 €
10	Experimentación	300,00 €
28	Redacción del documento	840,00 €
<b>Subtotal CAPÍTULO 3</b>		<b>4.260,00 €</b>

*Tabla 9. Presupuesto de costes de ingeniería.*

Al coste de los diferentes capítulos se le suma el 21% de IVA obteniéndose el coste total de desarrollo del proyecto.

<b>CAPÍTULO</b>	<b>Descripción</b>	<b>Coste</b>
1	Sistema de control en red	165,53 €
2	Instrumentación para sistema térmico con retardo	22,96 €
3	Costes de ingeniería	4.260,00 €
Subtotal		4.448,49 €
IVA (21%)		934,18 €
<b>TOTAL</b>		<b>5.382,67 €</b>

*Tabla 10. Coste total del proyecto.*

