



PRACTICA EMPRESARIAL STERICLINIC S.A.S.

FABIO ANDRÉS HERNÁNDEZ RUEDA

UNIVERSIDAD PONTIFICIA BOLIVARIANA SECCIONAL BUCARAMANGA  
ESCUELA DE INGENIERIAS  
FACULTAD DE INGENIERIA ELECTRÓNICA  
BUCARAMANGA  
2016

PRACTICA EMPRESARIAL STERICLINIC S.A.S

FABIO ANDRÉS HERNÁNDEZ RUEDA

Práctica Empresarial Presentada Para Optar al Título De Ingeniero Electrónico

Docente Supervisor:  
PhD. Omar Pinzón Ardila

UNIVERSIDAD PONTIFICIA BOLIVARIANA SECCIONAL BUCARAMANGA  
ESCUELA DE INGENIERIAS  
FACULTAD DE INGENIERIA ELECTRÓNICA  
BUCARAMANGA  
2016

Nota de aceptación

---

---

---

---

Firma del presidente del jurado

---

Firma del jurado

---

Firma del jurado

---

Bucaramanga, Enero del 2016

## DEDICATORIA

Dedico este trabajo a mis padres, quienes con su trabajo diario me han sacado adelante , a mis hermanos que son mi inspiración para ser cada día mejor, a Andrea Ávila, por su apoyo y amor incondicional y al Ing. Holguer Becerra, sin sus enseñanzas, este proyecto no sería posible.

***Fabio Hernández.***

## **AGRADECIMIENTOS**

Agradezco a Dios, por ser el motor que impulsa mi vida y por ponerme siempre en el camino correcto, con las personas correctas.

Al colegio Santa Isabel de Hungría, por ser de gran ayuda en la realización de el sueño de estudiar ingeniería en tan prestigiosa universidad

A la facultad de Ingeniería Electrónica, al Ing. Sergio Salinas por sus consejos y motivación, al D. Omar Pinzón por su colaboración en la realización de este proyecto y al director de la facultad Ing. Raúl Restrepo Agudelo quien desde el primer momento, no dudo en darme todo su apoyo y confianza.

A la familia STERICLINIC S.A.S, por tan grata acogida en mi periodo de práctica, al Ingeniero Peter Gómez, por todas las enseñanzas recibidas a lo largo de la práctica, a Cristian Torres, por su amistad y apoyo, a Luis Arturo Moreno, por transmitirme conocimientos que solo se logran con años de experiencia y a Juan Diego Restrepo, por creer que en Colombia, se pueden hacer grandes cosas.

## CONTENIDO

INTRODUCCIÓN .....	14
1. DESCRIPCION DE LA EMPRESA. ....	15
1.1 DATOS DE LA EMPRESA.....	15
1.2 ESTRUCTURA ORGANIZACIONAL .....	15
1.3 ACTIVIDAD ECONÓMICA.....	16
2. OBJETIVOS.....	18
2.1 OBJETIVO GENERAL .....	18
2.2 OBJETIVOS ESPECÍFICOS.....	18
3. PLAN DE TRABAJO PROPUESTO.....	19
3.1 ACTIVIDADES A DESARROLLAR.....	19
3.2 CRONOGRAMA DE ACTIVIDADES .....	20
4. MARCO TEÓRICO .....	21
4.1 EL ESTERILIZADOR A VAPOR .....	21
4.1.1. HISTORIA DE LAS AUTOCLAVES.....	21
4.1.2. PROCESO DE ESTERILIZACIÓN DE UNA AUTOCLAVE .....	24
4.1.3. PARTES DE UNA AUTOCLAVE .....	24
4.2. ESTRUCTURA DE UN SISTEMA EMBEBIDO. ....	25
4.2.1. UNIDAD DE PROCESAMIENTO .....	26
4.2.1.1. MATRIZ DE COMPUERTAS PRORAMABLES EN CAMPO (FPGA)..	26
4.2.2. SISTEMA DE VISUALIZACIÓN.....	29
4.2.2.1. CONTROLADOR RA8875.....	29
4.2.3. MÓDULOS DE ENTRADAS Y SALIDAS .....	31
4.2.3.1. AMPLIFICADOR OPERACIONAL .....	31
4.2.3.2. TRIODO PARA CORRIENTE ALTERNA (TRIAC) .....	32

4.2.4.	TARJETA DE CIRCUITO IMPRESO (PCB).....	33
4.2.4.1.	HISTORIA .....	33
4.2.4.2.	COMPOSICIÓN.....	33
4.2.4.3.	DISEÑO ELECTRÓNICO AUTOMATIZADO (EDA).....	34
5.	DESARROLLO DEL PLAN DE TRABAJO.....	35
5.1.	METODOLOGÍA EMPLEADA PARA EL DESARROLLO DEL PROYECTO. ....	35
5.2.	IDENTIFICACIÓN DE ENTRADAS Y SALIDAS DEL SISTEMA .....	36
5.3.	CIRCUITOS DE ADECUACIÓN DE ENTRADAS Y SALIDAS .....	38
5.3.1.	ACONDICIONAMIENTO DE LAS SEÑALES ANALÓGICAS.....	38
5.3.2.	ACONDICIONAMIENTO DE ENTRADAS Y SALIDAS DIGITALES. ....	39
5.4.	DISEÑO Y CONSTRUCCIÓN DEL PCB.....	40
5.5.	IDENTIFICACIÓN DE LOS SISTEMAS DE CONTROL EN LOS PROCESOS DE UNA AUTOCLAVE .....	42
5.6.	PROGRAMACIÓN DE LOS PROCESOS DE CONTROL EN LA FPGA.....	43
5.6.1.	MÓDULO CALDERÍN.....	45
5.6.2.	MÓDULO PUERTA.....	46
5.6.3.	MÓDULO CICLO .....	46
5.6.4.	MÓDULO PROCESADOR.....	52
5.6.5.	MÓDULO ADC.....	53
5.6.6.	RESUMEN DEL SISTEMA HDL IMPLEMENTADO EN LA FPGA ALTERA MAX 10.. .....	55
5.7.	SISTEMA DE VISUALIZACIÓN.....	57
5.7.1.	REQUERIMIENTOS .....	57
5.7.2.	SELECCIÓN DEL SISTEMA DE VISUALIZACIÓN .....	58
5.8.	PROGRAMACIÓN DE LA INTERFAZ GRÁFICA DE USUARIO.....	59
5.8.1.	LIBRERIAS UTILIZADAS .....	59

5.8.2. PROGRAMACIÓN DE DIBUJO DE BOTONES, TEXTO Y FIGURAS GEOMETRICAS EN LA PANTALLA.....	61
5.8.3. RESULTADOS DE LA PROGRAMACIÓN DE LA GUI. ....	66
GLOSARIO.....	73
APORTE AL CONOCIMIENTO .....	75
CONCLUSIONES Y RECOMENDACIONES.....	76
BIBLIOGRAFIA .....	77

## LISTA DE TABLAS

<b>TABLA 1.</b> CRONOGRAMA DE ACTIVIDADES.....	20
<b>TABLA 2.</b> COMPONENTES DEL SISTEMA DE CONTROL.....	37
<b>TABLA 3.</b> CARACTERÍSTICAS DEL SISTEMA DE VISUALIZACIÓN ELEGIDO [41] .....	58
<b>TABLA 4.</b> FUNCIONES DE LA LIBRERIA CONTROL.H .....	72

## LISTA DE FIGURAS

<b>FIGURA 1.</b> ORGANIGRAMA DE STERICLINIC S.A.S.....	15
<b>FIGURA 2.</b> AUTOCLAVE A VAPOR DE 150 LITROS [1].....	17
<b>FIGURA 3.</b> AUTOCLAVE A VAPOR DE 250 LITROS[1].....	17
<b>FIGURA 4.</b> DIAGRAMA DE LA PRIMERA OLLA A PRESIÓN DESCRITA POR DENIS PAPIN [4]21	
<b>FIGURA 5.</b> PRIMERA AUTOCLAVE A VAPOR [6].....	22
<b>FIGURA 6.</b> AUTOCLAVE CHAMBERLAND [9] .....	22
<b>FIGURA 7.</b> SISTEMA DE PUERTA AUTOCLAVE UNDERWOOD [11].....	23
<b>FIGURA 8.</b> AUTOCLAVE MATACHANA Y TUTTNAUER [12][13] .....	23
<b>FIGURA 9.</b> ESTERILIZADOR A VAPOR [19] .....	25
<b>FIGURA 10.</b> DIAGRAMA DE BLOQUES DE UN SISTEMA EMBEBIDO.....	26
<b>FIGURA 11.</b> ESTRUCTURA INTERNA DE UNA MAX 10 [25] .....	27
<b>FIGURA 12.</b> ARQUITECTURA DE UN PROCESADOR NIOS II [26] .....	28
<b>FIGURA 13.</b> DIAGRAMA DE BLOQUES DEL RA8875 [38]. .....	30
<b>FIGURA 14.</b> CONEXIÓN DEL RA8875 CON UN MCU [38]. .....	30
<b>FIGURA 15.</b> DIAGRAMA DE CONEXIÓN LM358 [30] .....	31
<b>FIGURA 16.</b> SIMBOLOGÍA DEL TRIAC [22]. .....	32
<b>FIGURA 17.</b> CONFIGURACIÓN Y SÍMBOLO DEL BT138 [33] .....	33
<b>FIGURA 18.</b> COMPONENTES DE UN PCB DOBLE CAPA [36]. .....	34
<b>FIGURA 19.</b> ENTORNO DE ALTIUM DESIGNER.....	35
<b>FIGURA 20.</b> METODOLOGÍA DEL PLAN DE TRABAJO .....	35
<b>FIGURA 21.</b> COMPOSICIÓN DE UNA BEMICRO MAX 10 [35].....	37
<b>FIGURA 22.</b> ENTRADAS ANALÓGICAS EN EL KIT BEMICRO MAX10.....	37
<b>FIGURA 23.</b> ENTRADAS Y SALIDAS DIGITALES EN EL KIT BEMICRO MAX10.....	38
<b>FIGURA 24.</b> ACONDICIONAMIENTO SEÑAL 4-20 MA. ....	38
<b>FIGURA 25.</b> ACONDICIONAMIENTO SEÑAL DE 0-10 V. ....	39
<b>FIGURA 26.</b> ACONDICIONAMIENTO SEÑAL DE 0-5 V. ....	39
<b>FIGURA 27.</b> ACONDICIONAMIENTO DE ENTRADAS DIGITALES. ....	40
<b>FIGURA 28.</b> ACONDICIONAMIENTO DE SALIDAS DIGITALES.....	40
<b>FIGURA 29.</b> DISEÑO 2D DEL PCB DEL SISTEMA. ....	41
<b>FIGURA 30.</b> DISEÑO 3D DEL PCB DEL SISTEMA. ....	41
<b>FIGURA 31.</b> PCB CONSTRUIDO. ....	42
<b>FIGURA 32.</b> ETAPAS DE UN CICLO DE ESTERILIZACIÓN. ....	43

<b>FIGURA 33.</b> MÓDULOS IMPLEMENTADOS EN LA FPGA. ....	44
<b>FIGURA 34.</b> DIAGRAMA DE FLUJO MÓDULO CALDERÍN. ....	45
<b>FIGURA 35.</b> CIRCUITO DIGITAL DEL MÓDULO CALDERÍN.....	46
<b>FIGURA 36.</b> MÁQUINA DE ESTADOS MÓDULO PUERTA.....	47
<b>FIGURA 37.</b> CIRCUITO DIGITAL DEL MÓDULO CICLO.....	48
<b>FIGURA 38.</b> MÁQUINA DE ESTADOS MÓDULO PRE-VACÍOS. ....	49
<b>FIGURA 39.</b> MÁQUINA DE ESTADOS MÓDULO ALCANZA SET-POINTS. ....	50
<b>FIGURA 40.</b> MÁQUINA DE ESTADOS MÓDULO DESCARGA Y SECADO.....	51
<b>FIGURA 41.</b> MÁQUINA DE ESTADO MÓDULO FINALIZACIÓN.....	52
<b>FIGURA 42.</b> ELEMENTOS DEL PROCESADOR BASADO EN NIOS II.....	53
<b>FIGURA 43.</b> COMPONENTES QUE CONFORMAN EL MÓDULO ADC.....	54
<b>FIGURA 44.</b> CONFIGURACIÓN DEL IP ALTERA MODULAR ADC CORE. ....	54
<b>FIGURA 45.</b> CONEXIÓN DEL IP ALTERA MODULAR ADC CORE EN QSYS.....	55
<b>FIGURA 46.</b> CIRCUITO DIGITAL DEL MÓDULO ADC_DATA.....	55
<b>FIGURA 47.</b> REPORTE DE COMPILACIÓN DEL HDL IMPLEMENTADO.....	56
<b>FIGURA 48.</b> CONSUMO DE POTENCIA DEL HDL IMPLEMENTADO.....	56
<b>FIGURA 49.</b> PANTALLA KP-300 DE SIEMENS [40]. ....	57
<b>FIGURA 50.</b> WAVESHARE 7 INCH RESISTIVE TOUCH LCD [41]. ....	58
<b>FIGURA 51.</b> LECTURA Y ESCRITURA DE UN REGISTRO. [28].....	59
<b>FIGURA 52.</b> DIAGRAMA DE FLUJO DE LECTURA DEL PANEL TÁCTIL [28] .....	60
<b>FIGURA 53.</b> DIBUJO DE CUADRADO EN PANTALLA.....	62
<b>FIGURA 54.</b> DIBUJO DE ICONO EN PANTALLA. ....	63
<b>FIGURA 55.</b> DIBUJO DE TEXTO EN PANTALLA. ....	64
<b>FIGURA 56.</b> DIAGRAMA DE FLUJO DE LA GUI. ....	67
<b>FIGURA 57.</b> VENTANA DE INICIO.....	68
<b>FIGURA 58.</b> VENTANA ALCANZANDO PRESIÓN EN CHAQUETA. ....	68
<b>FIGURA 59.</b> VENTANA INTRODUCIR CARGA. ....	69
<b>FIGURA 60.</b> VENTANA DE SELECCIÓN DE CICLO. ....	69
<b>FIGURA 61.</b> VENTANA CONFIGURACIÓN DE CICLO. ....	70

## **RESUMEN GENERAL DE TRABAJO DE GRADO**

**TITULO:** PRÁCTICA EMPRESARIAL EN STERICLINIC S.A.S.

**AUTOR(ES):** FABIO ANDRES HERNÁNDEZ RUEDA.

**FACULTAD:** Facultad de Ingeniería Electrónica

**DIRECTOR:** Omar Pinzón Ardila

### **RESUMEN**

La práctica empresarial en STERICLINIC SAS consistió en el desarrollo de un sistema embebido de control para un esterilizador a vapor. El proyecto se centra en tres fases: en la primera fase se realizó el diseño de los circuitos esquemáticos y el PCB de la tarjeta electrónica con todo el hardware necesario para hacer la adecuación de las señales de entrada y salida del sistema, en la segunda fase del proyecto se realizaron todos los algoritmos de control de una autoclave en una FPGA a través de la programación en Verilog y en la tercera fase se realiza el diseño y desarrollo de una interfaz gráfica mediante una pantalla táctil LCD de siete pulgadas programada con un sistema integrado NIOS II.

### **PALABRAS CLAVE:**

FPGA NIOS II, GUI, VERILOG, AUTOCLAVE,  
ESTERILIZACIÓN, SISTEMA DE CONTROL.

**V° B° DIRECTOR DE TRABAJO DE GRADO**

## **GENERAL SUMMARY OF WORK OF GRADE**

**TITLE:** ENTERPRISE PRACTICE STERICLINIC S.A.S.

**AUTHOR(S):** FABIO ANDRES HERNÁNDEZ RUEDA.

**FACULTY:** Faculty of electronic engineering

**DIRECTOR:** Omar Pinzón Ardila

### **ABSTRACT**

The work done in the company STERICLINIC SAS was the development of an embedded system for controlling a steam sterilizer. The project focuses on three phases: the first phase design schematics and PCB circuits of the electronic card with all the required hardware was developed to handle the inputs and outputs of the system, in the second phase of the program was carried out control algorithms in an FPGA autoclave and in the third phase the design of a graphical user interface is made and implemented with a LCD touch screen seven inches by NIOS II embedded system.

### **KEYWORDS:**

FPGA NIOS II, GUI, VERILOG, AUTOCLAVE,  
STERILIZATION, CONTROL SYSTEM.

**V° B° DIRECTOR OF GRADUATE WORK**

## INTRODUCCIÓN

STERICLINIC S.A.S diseña y produce equipos de esterilización médica, uno de estos equipos es la autoclave a vapor de 150 litros, cuyo control electrónico se realiza mediante un PLC (*Programmable Logic Controller*) S7-1200 de Siemens, encargado de hacer la conmutación de las electro válvulas y las resistencias, tomar la medición de los sensores de presión y de temperatura, enviar señales al variador de velocidad para abrir y cerrar la compuerta de entrada del instrumental y realizar la comunicación con el panel básico KP-300 de Siemens que consiste en una pantalla monocromática de 3.6" y un teclado configurable que hace la función de HMI (*Human Machine Interface*) del esterilizador.

Aunque hacer el control del sistema con un PLC presenta ventajas como la robustez en el aislamiento eléctrico y control de ruido, ha ocasionado que la empresa se limite en sus diseños, ya que el hecho de implementar un nuevo sensor o cambiar el panel HMI significa comprar módulos Siemens de alto costo y en algunas ocasiones cambiar a un PLC de gama más alta, lo que se traduce en incrementar el valor final del producto y hacerlo menos competitivo en el mercado. Es por esto que STERICLINIC S.A.S. tiene como objetivo el diseño y desarrollo de su propio sistema de control, que le brinde autonomía al querer mejorar sus diseños sin tener que aumentar el precio final del producto.

Para realizar este objetivo fue necesario el diseño y la construcción de un PCB (*Printed Board Circuit*) apoyándose en el software *Altium Designer* que es un programa de diseño electrónico automatizado (EDA), que permite realizar los circuitos esquemáticos del sistema mediante librerías de los fabricantes y basarse en ellos para hacer el diseño del PCB, respetando las medidas de los componentes; en esta tarjeta se realizaron los acondicionamientos de las señales digitales y analógicas de entradas y salidas del sistema.

La lógica de control se implementó mediante una FPGA (*Field Program Gate Array*), dispositivo electrónico que incorpora bloques lógicos programables mediante algún lenguaje de descripción de hardware (HDL) como Verilog, VHDL o System Verilog esto le permite tener la capacidad de reproducir desde integrados sencillos de compuertas lógicas hasta sistemas digitales como un microprocesador de arquitectura compleja como ARM (*Advanced Risc Machine*), estas características son adecuadas para realizar un micro controlador, realizar la programación de una pantalla LCD táctil en lenguaje C y aprovechar todo el código que se encuentra en Internet para ello debido a que este lenguaje es mucho más popular que Verilog o cualquier otro HDL.

En el presente documento se explica con detalle las diferentes etapas que se llevaron a cabo para realizar el diseño y construcción de la tarjeta, la programación de la FPGA y el desarrollo de la HMI, así como los resultados obtenidos, las conclusiones y los posibles trabajos futuros.

## 1. DESCRIPCION DE LA EMPRESA.

### 1.1 DATOS DE LA EMPRESA

La empresa STERICLINIC S.A.S fue creada por Luis Arturo Moreno, Raúl Restrepo, Federico Restrepo y María Lucia Restrepo con el objetivo de establecer una empresa colombiana con equipos de alta calidad, con capacidad de competir con autoclaves importadas desde diferentes lugares del mundo. Inicia labores en el mes de Mayo del año 2013, en la calle 43 # 12-41 bajo la coordinación de Luis Arturo Moreno.

En el año 2014 se traslada a una bodega ubicada en el Barrio Gaitan, y se estructura una planta más organizada con puestos de trabajo delimitados y una línea de producción establecida, con el ánimo de mejorar cada día se realiza un área para validación y análisis de calidad de los equipos, logrando productos con altos estándares de calidad certificados por el INVIMA.

### 1.2 ESTRUCTURA ORGANIZACIONAL

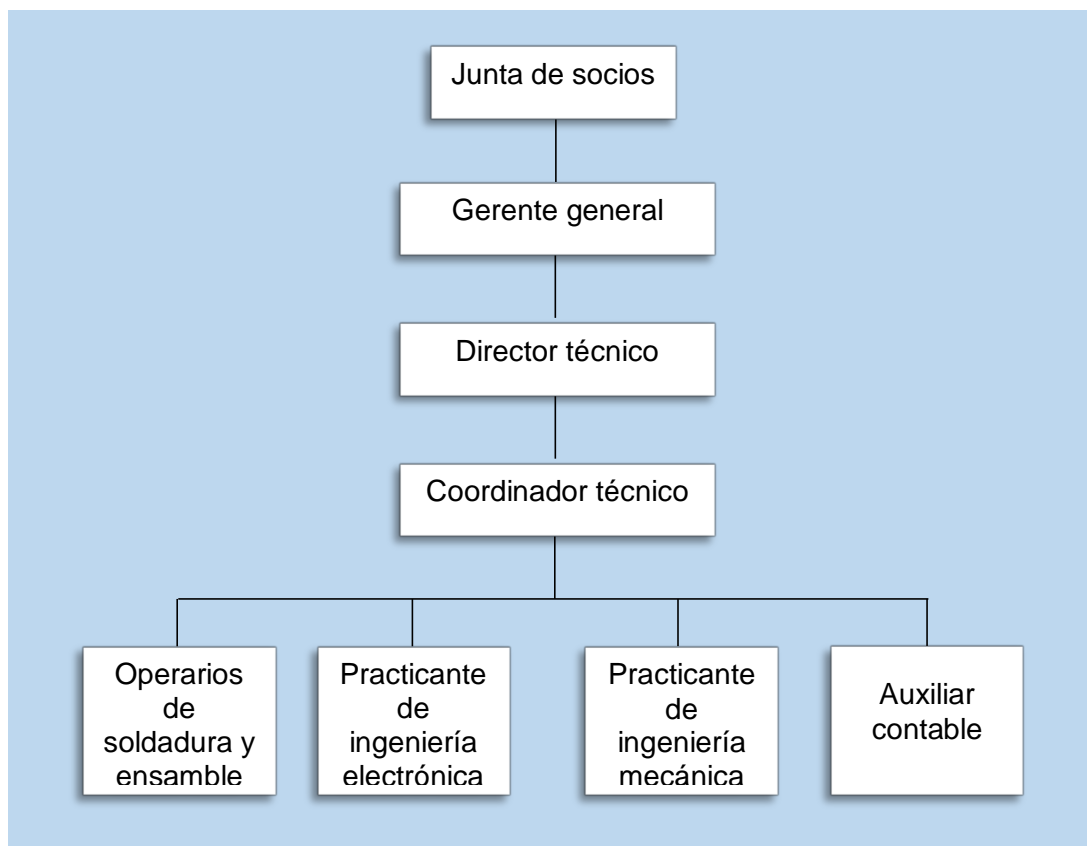


Figura 1. Organigrama de STERICLINIC S.A.S.

**Misión:**

Crear lealtad en nuestros clientes al tener a su disposición el más amplio portafolio de equipos de esterilización en todo el territorio nacional, trabajando siempre con responsabilidad profesional, eficiencia y calidez humana para presentar un servicio de excelente calidad.

**Visión:**

Ser en el año 2016 una compañía líder con reconocimiento a nivel nacional en el sector de la salud por la variedad, calidad e innovación de los productos y servicios ofrecidos, excelencia en la atención, compromiso con el desarrollo y crecimiento integro de su talento humano y con una infraestructura que brinde un ambiente de trabajo agradable y que satisfaga las experiencias de nuestros clientes.

**Valores corporativos:**

- Pasión por lo que se hace.
- Profesionalismo frente a la empresa.
- Humildad en las relaciones con nuestros clientes y compañeros de trabajo.
- Perseverancia con los objetivos y metas propuestas.
- Respeto hacia todos los clientes, empleados y hacia el propio ejercicio de la profesión.

**Política de calidad:**

Alcanzar un posicionamiento sostenido en el mercado nacional, fundamentado en el mejoramiento continuo de los procesos y la capacitación permanente del recurso humano, ofreciendo a todos nuestros clientes productos y servicios de alta calidad a buen precio y en tiempos cortos.

**1.3 ACTIVIDAD ECONÓMICA**

Stericlinic S.A.S es una empresa que brinda servicio de mantenimiento a centrales de esterilización, además de ser fabricante de autoclaves a vapor de acuerdo a las necesidades de los clientes.

Entre sus productos principales se encuentran la autoclave a vapor con capacidad de 150 litros en su cámara cilíndrica, con pantalla de visualización de proceso y sistema automático para una puerta de referencia STE150L1P o dos puertas de referencia STE150L2P (ver Figura 2) y la autoclave a vapor de 250 litros con cámara cúbica y las características ya mencionadas en la autoclave de 150 litros (ver Figura 3).

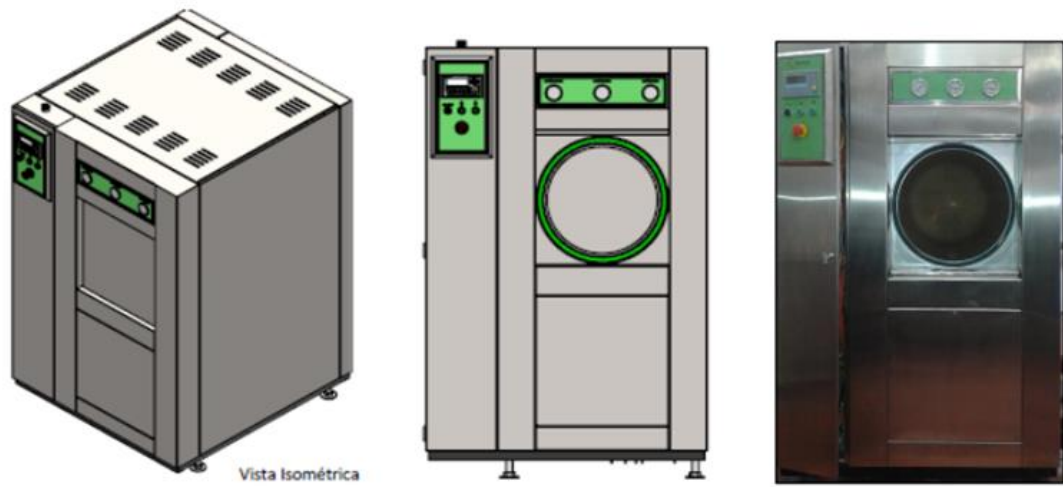


Figura 2. Autoclave a vapor de 150 litros [1]

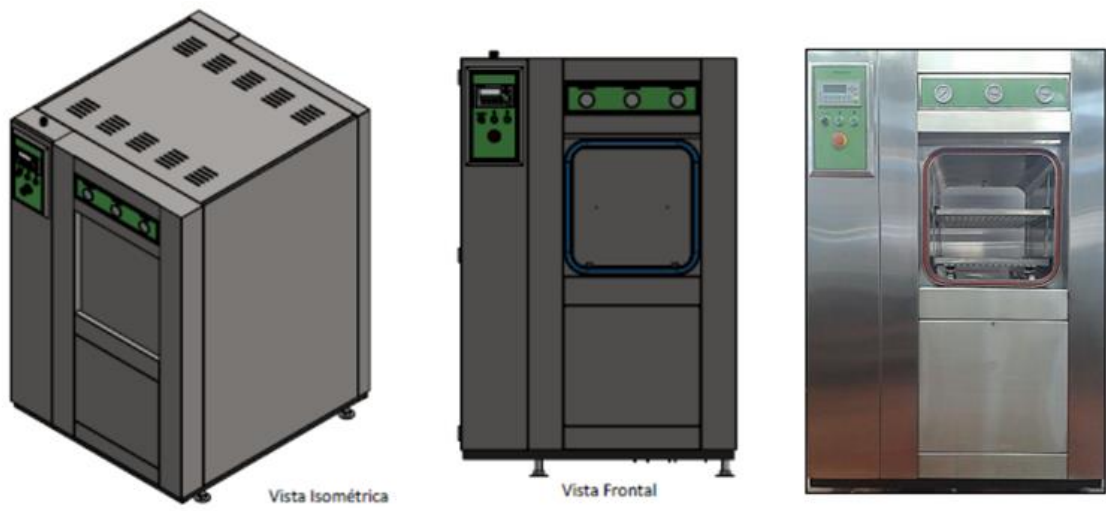


Figura 3. Autoclave a vapor de 250 litros[1]

## **2. OBJETIVOS**

### **2.1 OBJETIVO GENERAL**

Diseñar y construir un controlador, utilizando sistemas embebidos para esterilizadores a vapor.

### **2.2 OBJETIVOS ESPECÍFICOS**

- Realizar el marco teórico con los conceptos y normativas para el control del esterilizador.
- Establecer los requerimientos del sistema de control para el diseño esquemático del circuito.
- Seleccionar la arquitectura del controlador y realizar el diseño del controlador según los requerimientos establecidos.
- Diseñar la interfaz gráfica de usuario para el esterilizador a vapor.

### 3. PLAN DE TRABAJO PROPUESTO

#### 3.1 ACTIVIDADES A DESARROLLAR

Las actividades se dividieron en dos tipos como se muestra a continuación:

- Diseñar y construir un sistema de control utilizando sistemas embebidos para esterilizadores a vapor de uso médico.
  - Estado del arte y marco teórico sobre equipos de esterilización a vapor y normatividad para el control del proceso.
  - Identificar los requerimientos del sistema de control para el diseño esquemático del circuito.
  - Evaluación y socialización de requerimientos para selección de dispositivos de control e interfaz de usuario.
  - Diseñar del sistema de control según los requerimientos identificados.
  - Diseño de la interfaz gráfica de usuario para el esterilizador a vapor.
  
- Apoyo al proceso de fabricación de los esterilizadores a vapor.
  - Apoyo al cableado y montaje de sistema de control de los esterilizadores y programación de equipos.
  - Apoyo en las pruebas de calificación de los esterilizadores.
  - Documentación del proceso de fabricación.
  - Elaboración de planos eléctricos, electrónicos y constructivos de los equipos.

### 3.2 CRONOGRAMA DE ACTIVIDADES

<b>Semanas</b> Inicio 15/09/2015 fin 13/01/2016	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
<b>Actividades</b>																
Identificar los requerimientos del sistema de control para el diseño esquemático del circuito.	■	■	■	■												
Evaluación y socialización de requerimientos para selección de dispositivos de control e interfaz de usuario.					■	■	■									
Diseñar del sistema de control según los requerimientos identificados.								■	■	■	■	■				
Diseño de la interfaz gráfica de usuario para el esterilizador a vapor.													■	■	■	■
Documentación del proyecto			■	■	■	■	■	■	■	■	■	■	■	■	■	■
Apoyo al cableado y montaje de sistema de control de los esterilizadores y programación de equipos.															■	■
Apoyo en las pruebas de calificación de los esterilizadores.															■	■
Documentación del proceso de fabricación.			■	■	■	■	■	■	■	■	■	■	■	■	■	■
Elaboración de planos eléctricos, electrónicos y constructivos de los equipos.			■	■	■	■	■	■	■	■	■	■	■	■	■	■

**Tabla 1.** Cronograma de actividades



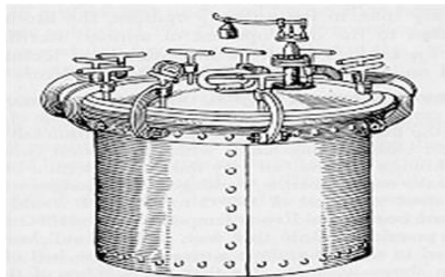
La relación entre los microorganismos y la infección se establece hacia 1878. La primera autoclave se fabrica en 1879: se trataba de un aparato portátil de 6 litros de capacidad calentado por alcohol (ver figura 5). [5]



**Figura 5.** Primera autoclave a vapor [6]

Koch y Wolffhugel en 1881 dan los fundamentos de la esterilización por calor seco y calor húmedo basándose en los descubrimientos de Pasteur en 1876: 30 minutos a 110° -120 °C de exposición al vapor que eran equivalentes a una hora de calor seco a 130 °- 150 °C. [7]

Charles Chamberland, colaborador de Louis Pasteur hizo la primera autoclave de vapor a presión en 1880 a raíz de la investigación de Koch y Wolffhugel, dicho aparato de esterilización lleva su nombre, Autoclave de Chamberland, que en la actualidad todavía es utilizado, con algunas modificaciones pero basado en el mismo principio.[8]



**Figura 6.**Autoclave Chamberland [9]

En 1924 Weeden Underwood promovió el concepto de central de esterilización y editó el libro "Texto de esterilización", además de hacer modificaciones como el sistema de puerta (figura 7) a las autoclaves que impulsaron la comercialización del esterilizador de vapor a presión. [10]

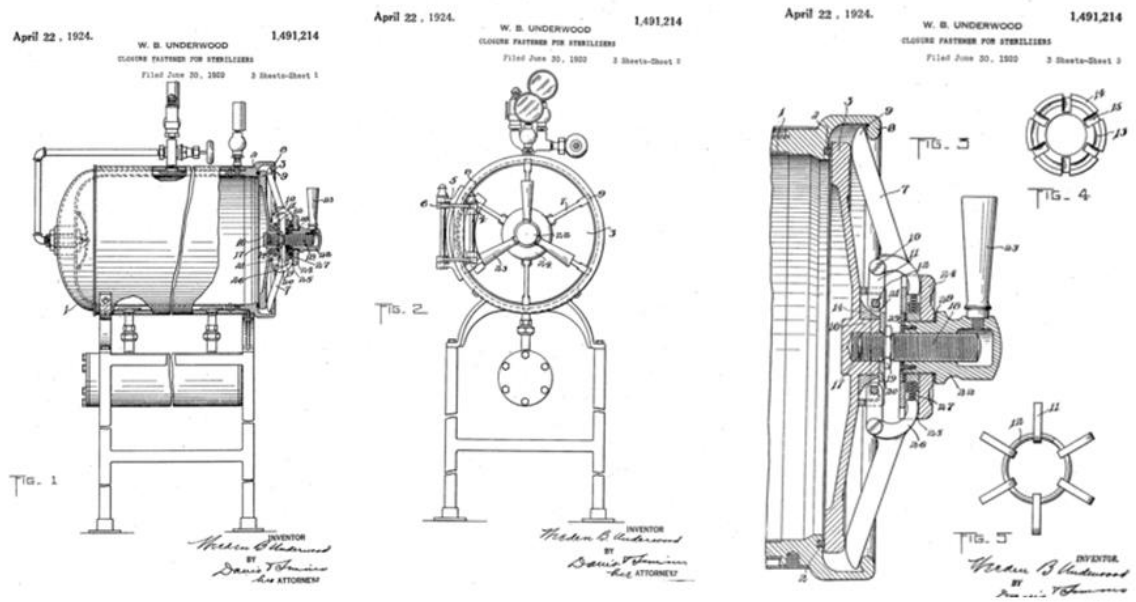


Figura 7. Sistema de puerta Autoclave Underwood [11]

Actualmente los esterilizadores son máquinas bastante complejas, con control automatizado, panel táctil, impresora y exportación de datos vía Internet, las empresas más reconocidas a nivel mundial son Tuttnauer, Matachana y Steris quienes mantienen la información acerca de la fabricación de sus productos muy reservada.



Figura 8. Autoclave Matachana y Tuttnauer [12][13]

#### **4.1.2. PROCESO DE ESTERILIZACIÓN DE UNA AUTOCLAVE**

El proceso de esterilización de una autoclave se fundamenta en exponer los microorganismos a la acción del calor mediante vapor saturado, principalmente hay dos temperaturas de exposición que son 121 °C y 134 °C. En caso de presentar una temperatura de exposición de 121 °C se expone el material a esterilizar a una presión de 15 psi y en uno de 134°C se tiene una presión de 30 psi, si no se mantienen estas relaciones se puede producir vapor sobresaturado (presión de vapor alta) y tener cargas mojadas o vapor sobrecalentado (presión de vapor baja) y producir cargas no esterilizadas en al finalizar el ciclo de esterilización.[14]

Un ciclo de esterilización por vapor se compone de las siguientes etapas:

- Acondicionamiento: En esta etapa se realiza la eliminación de aire de la cámara y de la carga, se prepara el vapor a la temperatura y presión adecuadas según el ciclo seleccionado.
- Exposición: La carga (material a ser esterilizado) se expone al vapor al vapor por un tiempo prolongado, manteniendo la temperatura y presión programada.
- Finalización: Última etapa en la que se evacua el vapor de la cámara y de la carga, y se realiza el secado de la misma. [15]

La eliminación de aire se puede hacer con dos sistemas: sistema gravitacional, en donde el vapor entra por la parte superior de la cámara y se elimina el aire por el drenaje gracias a la acción de la fuerza de gravedad sobre este; y por sistema de prevacío, que opera mediante una bomba de succión que permite que el ciclo de esterilización sea mucho más rápido. [16]

#### **4.1.3. PARTES DE UNA AUTOCLAVE**

De manera general las autoclaves están conformadas de una estructura, una parte mecánica y un control eléctrico.

La estructura está construida en acero inoxidable Clase AISI 316 principalmente, incluyendo las válvulas y cañerías y consiste en una cámara cuadrangular o cilíndrica, la cámara puede ser de doble fondo, esto le permite a la cámara que el control de temperatura no solo se haga con el vapor que le ingresa a la cámara, sino también con el vapor que circula por la parte externa de las paredes de la cámara. Las puertas tienen la misma forma de la cámara, pueden ser manuales o automáticas y de una o dos puertas: una autoclave consta de dos puertas si esta entre una área limpia y una área estéril para evitar la contaminación cruzada [17].

En la parte mecánica se tienen principalmente las válvulas encargadas de controlar la entrada y salida del flujo de vapor de la cámara y chaqueta, además de todos los motores que realizan la función de abrir y cerrar la puerta en caso de que el sistema de puerta sea automático y una bomba para realizar el vacío a la cámara si la autoclave funciona con sistema de vacío. [18]

El sistema de control electrónico consiste en una unidad de procesamiento que puede ser un PLC, un sistema embebido de control o cualquier otro sistema que permita enviar y recibir información del sistema y tomar decisiones al respecto.

Un ejemplo de una autoclave sencilla se muestra en la figura 4.

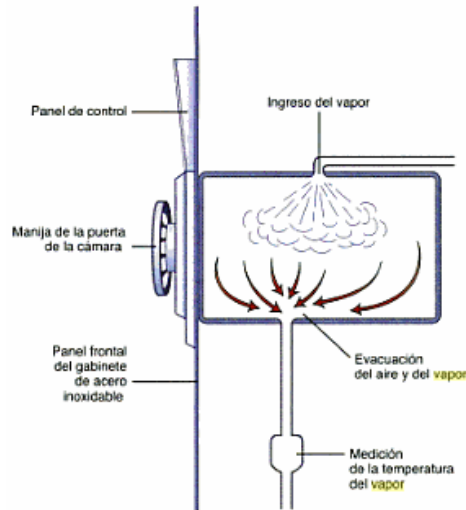


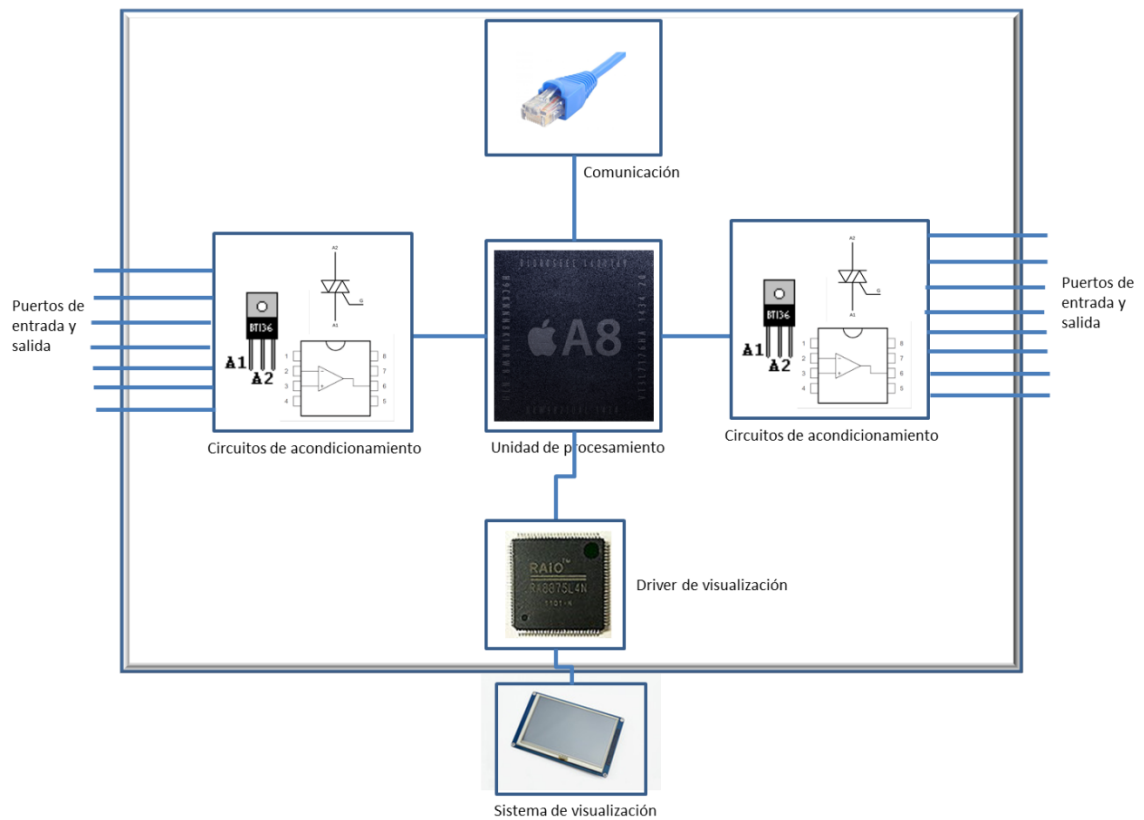
Figura 9. Esterilizador a vapor [19]

## 4.2. ESTRUCTURA DE UN SISTEMA EMBEBIDO.

Un sistema embebido consiste en un dispositivo electrónico que tiene integrado en su placa base todos los componentes necesarios para realizar algunas o pocas funciones dedicadas normalmente procesadas en tiempo real. El diseño de un sistema embebido muchas veces presenta un gran reto que involucra conocimientos de disciplinas como la electrónica, la informática y la física, es por esta razón que los equipos que desarrollan sistemas embebidos se componen de profesionales en varias disciplinas, de acuerdo con el sistema en cuestión a desarrollar.

Los sistemas embebidos tienen actualmente múltiples aplicaciones en la medicina, la automatización electrónica, el tratamiento de señales y aeronáutica, y se producen en gran cantidad superando a los computadores de uso general.

Un sistema embebido tiene básicamente cuatro componentes: Una unidad de procesamiento, una unidad de comunicación, un sistema de visualización y módulos de entrada y salidas analógicas y digitales todos integrados en circuito impreso [20]. La figura 10 presenta un diagrama de bloques de los componentes de un sistema embebido.



**Figura 10.**Diagrama de bloques de un sistema embebido

## 4.2.1. UNIDAD DE PROCESAMIENTO

Es la parte central del sistema, en ella se encuentra el microprocesador, microcontrolador, DSP, etc. Es decir es la capacidad de cómputo del sistema y el encargado de ejecutar las instrucciones mediante la realización de operaciones básicas aritméticas y lógicas de acuerdo a un programa previamente establecido. Un ejemplo de unidad de procesamiento son las FPGA ( *field programmable gate Array*) [21].

### 4.2.1.1. MATRIZ DE COMPUERTAS PRORAMABLES EN CAMPO (FPGA)

Los sistemas digitales se han presentado a lo largo del tiempo, como una solución a muchos problemas de la industria, en las cuales un error resulta bastante costoso y en algunos casos, riesgoso para el ser humano. Día a día las exigencias en desempeño y funcionamiento de dichos sistemas ha venido creciendo y se ha desarrollado tecnología que hace posible la realización de sistemas de control complejos con disponibilidad de diferentes dispositivos de hardware (FPGAs, CPLDs, IPs, ICs) de bajo costo, además de una gran cantidad de software de respaldo (librerías, drivers y sistemas operativos en tiempo real).[20]

Una FPGA ( *Fiel Programmable Gate Array*) es un dispositivo semiconductor que contiene bloques de lógica cuya interconexión y funcionalidad puede ser configurada 'in situ' mediante un lenguaje de descripción especializado como VHDL o Verilog, [22] lo que permite realizar prototipos de sistemas digitales de una manera rápida y económica para su futura implementación en ASIC (Circuito Integrado para Aplicaciones Específicas), aunque recientemente debido a su bajo costo, las FPGAs han pasado de ser sistemas para hacer prototipos, a implementarse en productos electrónicos finalizados.[23]

## ALTERA MAX 10

La FPGA MAX 10 es uno de los más recientes productos de Altera, cuenta con una avanzada tecnología de integración que incorpora una memoria Flash, una memoria RAM, hasta dos ADC y un DSP en el mismo chip, lo que lo hace un dispositivo de bajo costo.[24]

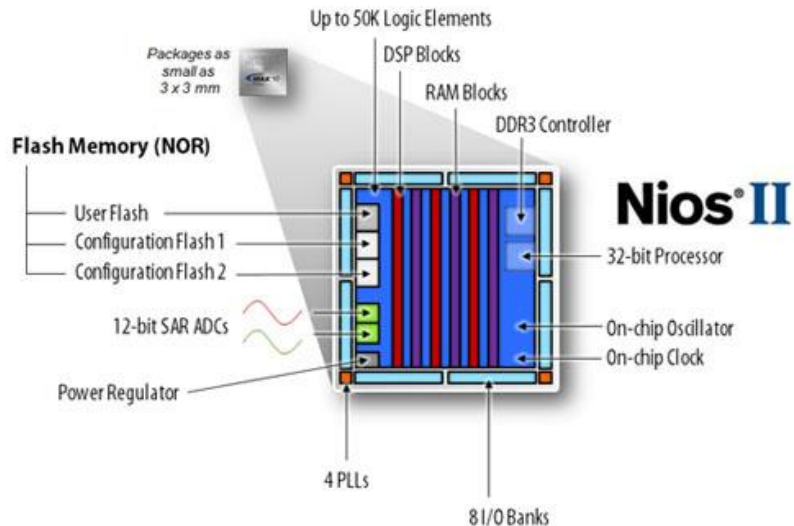


Figura 11. Estructura interna de una MAX 10 [25] .

Su arquitectura incluye:

- Hasta 50000 elementos lógicos (Les)
- Hasta 500 pines de entrada/salida
- Arquitectura no volátil de encendido instantáneo
- Un chip de hasta 3x3 mm<sup>2</sup>.
- SRAM embebida
- Bloques DSP
- Bucles de alto rendimiento para enganche de fase (PLL).
- Interfaz de memoria externa

- Soporte de procesadores de núcleo blando NIOS II
- ADC embebido de hasta 18 canales con sensor de temperatura
- Flash incorporado
- Oscilador interno

Las principales aplicaciones que tiene esta FPGA son: control de motores, sistemas de gestión de batería en automóviles, sistemas de información de entretenimiento avanzados de asistencia al conductor (ADAS), aplicaciones de comunicaciones inalámbricas de alta velocidad y procesamiento de imágenes en tiempo real para máquinas de visión artificial [25].

## PROCESADOR EMBEBIDO NIOS II

NIOS II es un procesador embebido de 32 bits de arquitectura RISC ( *Reduced Instruction Set Computer*), Altera desarrolló toda la descripción de hardware de NIOS II con el fin de incorporarse en las FPGAs de la misma empresa, este procesador puede adaptarse a sistemas hardware reconfigurables que van desde un sistema básico hasta sistemas operativos en tiempo real y sistemas tipo Linux [26].

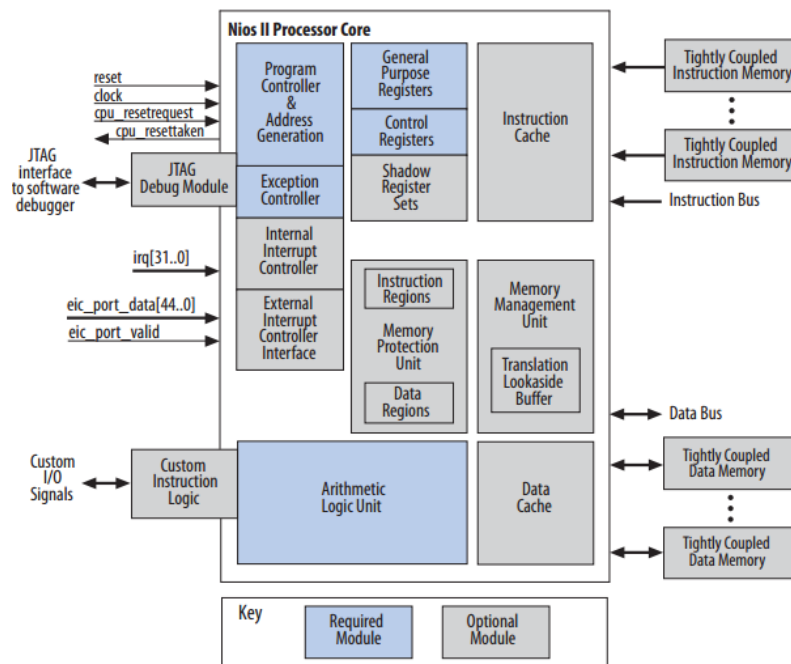


Figura 12.Arquitectura de un procesador NIOS II [26] .

## ESTRUCTURA BÁSICA DE UN SISTEMA BASADO EN NIOS II.

Los siguientes son los elementos que se deben añadir para una correcta realización de un sistema basado en NIOS II:

- **JTAG:** es un mecanismo de transporte que permite al programador acceder módulo de depuración encontrado en el NIOS lo que permite programar y supervisar al procesador [27].
- **Instrucion Master Port:** Es el puerto encargado de llevar el cache de instrucciones hacia el procesador [27].
- **Data Master Port:** Esta encargado de supervisar, enviar y recibir los datos de los puertos, la memoria y demás periféricos conectados al Nios II [27].
- **Reset:** Reinicia el sistema, de gran importancia para un bloqueo total del sistema [27].
- **Clock:** Esta encargado de recibir la frecuencia de reloj del oscilador externo para que puedan completarse los ciclos de máquina [27].
- **Irq:** Recibe las interrupciones de los periféricos conectados según la prioridad establecida [27].

### 4.2.2. SISTEMA DE VISUALIZACIÓN

Es un programa informático que actúa de interfaz de usuario, utilizando un conjunto de recursos gráficos proyectado en una pantalla con el objetivo de representar de manera entendible al usuario lo que está sucediendo en el sistema [21].

Actualmente el sistema de visualización puede ser a su vez un sistema de interacción con el usuario mediante paneles táctiles que funcionan como interfaces gráficas de usuario. Estas interfaces requieren a menudo de un driver externo o GPU ( *Graphics Processor Unit*) encargado de quitar carga de trabajo al procesador central y procesar en paralelo los diferentes gráficos que se muestran en el sistema de visualización [21]. Un ejemplo de ello es el controlador RA8875

#### 4.2.2.1. CONTROLADOR RA8875

El RA8875 es un potente controlador tipo texto/gráfico para pantallas tipo LCD, que permite mediante la ejecución de registros dibujar sobre una pantalla LCD de hasta 800 X 480 pixeles de resolución. Cuenta con 768 KB de RAM que permite dibujar hasta dos cuadros superpuestos. El RA8875 también puede controlar pantallas táctiles resistivas de cuatro hilos. Tiene la capacidad de dibujar figuras geométricas como una línea, un cuadrado o un círculo mediante su hardware además de tener funciones especiales como ventana flotante, girar ventana, desplazar texto, lo que hace que se le exija menos al controlador principal y se pueda acelerar la visualización en pantalla [28].

La siguiente figura muestra el diagrama de bloques del controlador RA8875.

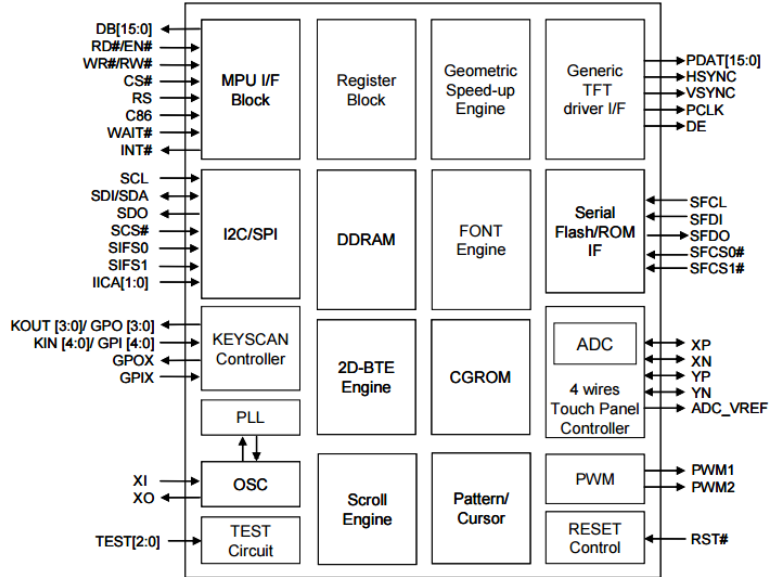


Figura 13. Diagrama de bloques del RA8875 [38].

Para hacer la conexión del controlador con algún dispositivo microprocesado ( Arduino, PIC, Raspberry, entre otros) , se debe tener en cuenta el siguiente diagrama de conexión.

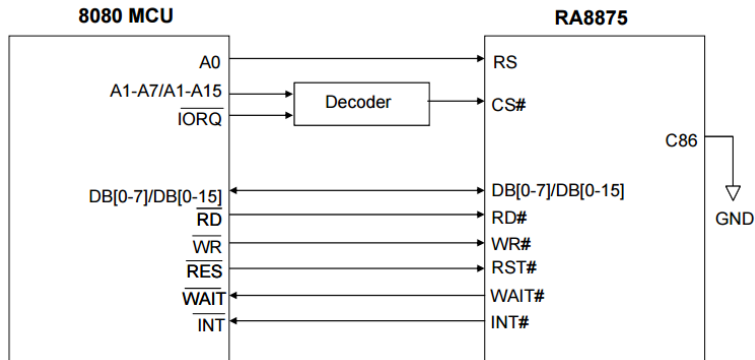


Figura 14. Conexión del RA8875 con un MCU [38].

### 4.2.3. MÓDULOS DE ENTRADAS Y SALIDAS

El módulo de entradas y salidas analógicas y digitales suele emplearse para procesar señales transmitidas por sensores, activar diodos LED, conmutar relés, activar resistencias, controlar motores, etc. [21] A menudo se requiere una etapa de acondicionamiento para tratar estas señales. A continuación se explicaran algunos componentes frecuentemente empleados para realizar estas etapas de acondicionamiento.

#### 4.2.3.1. AMPLIFICADOR OPERACIONAL

Se le atribuye a George Philbrick la construcción del primer amplificador operacional y su comercialización en 1948 bajo su propia compañía Philbrick Associate, estos amplificadores fueron destinados a realizar operaciones matemáticas, de allí proviene la palabra operacional. Entre los años de 1964 y 1967 la compañía Fairchild desarrolló los amplificadores operacionales en circuitos integrados 702,709 y 741, mientras que National Semiconductor introdujo el 101/301, al ser circuitos integrados su costo y tiempo de fabricación se redujo, además de ocupar menos espacio. Con el pasar del tiempo se han desarrollado amplificadores para funciones complejas, hoy en día se utilizan en muchas aplicaciones y son de gran importancia en el tratamiento de señales analógicas [29].

#### CIRCUITO INTEGRADO LM358

El LM358 tiene dos amplificadores operacionales de alta ganancia de manera independiente, que opera con alimentación simple, es utilizado en áreas como transductores y bloques de ganancia DC [30], a continuación se muestra su diagrama de conexión.

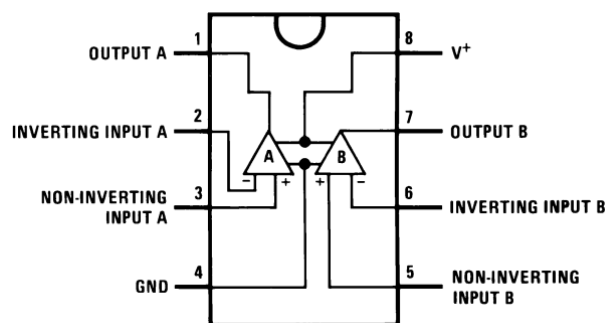


Figura 15. Diagrama de conexión LM358 [30]

#### 4.2.3.2. TRIODO PARA CORRIENTE ALTERNA (TRIAC)

El triac es un dispositivo de tres terminales usado para controlar la corriente promedio que fluye hacia una carga, tiene capacidad de producir corriente en cualquier dirección, de MT2 hacia MT1 o de MT1 hacia MT2 de acuerdo a la polaridad del voltaje cuando pasa una corriente de pocos miliamperios en la puerta (ver figura 11), se puede decir que el triac es un interruptor electrónico de corriente alterna normalmente abierto [31].

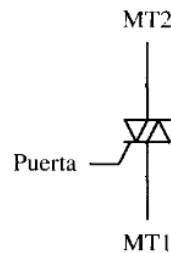


Figura 16. Simbología del TRIAC [22].

Algunas características eléctricas más importantes del triac que se deben tener en cuenta para el momento de su selección son:

- $I_{gt}$ : Corriente de puerta necesaria para el disparo, la mayoría de triacs tienen un  $I_{gt}$  de 0.1 a 20 mA [32].
- $V_{gt}$ : Voltaje de puerta necesario para disparar el triac, comúnmente  $V_{gt}$  es de 0.6 a 2.0 V [32].
- $I_{HO}$ : Es la corriente mínima que debe pasar entre MT1 y MT2 para que triac se encuentre encendido cuando hay corriente de disparo, se le conoce como corriente de mantenimiento y en la mayoría de triacs es menor a 100 mA [32].
- $I_{T_{rms}}$ : Corriente rms máxima permitida entre MT1 y MT2.
- $V_{DROM}$ : Voltaje de rompimiento, si se excede este voltaje entre las terminales MT1 y MT2 el triac entrara en constante conducción de corriente sin posibilidad de control [32].
- $V_{TM}$ : Es el voltaje de estado encendido, un valor bajo significa que el triac replica cercanamente la acción de un interruptor mecánico, aplicando el voltaje de alimentación completo a la carga. Por lo general este valor esta entre 1 y 2 V [32].

#### TRIAC BT138

El triac BT138, tiene uso en diferentes aplicaciones que requieren de cargas de potencia alta y alto rendimiento en altas temperaturas, sus aplicaciones incluyen control de motores, control de voltajes en la industria e iluminación doméstica [33].

Sus parámetros típicos más importantes son:  $V_{Drom} = 800$  V,  $I_{T_{rms}} = 12$  A,  $I_{gt} = 5 \sim 22$  mA,  $V_{gt} = 0.7$  V. A continuación se muestra las configuraciones y el símbolo del BT138 [33].

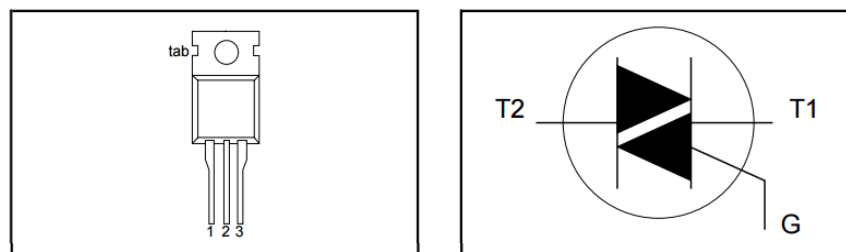


Figura 17. Configuración y símbolo del BT138 [33] .

## 4.2.4. TARJETA DE CIRCUITO IMPRESO (PCB)

### 4.2.4.1. HISTORIA

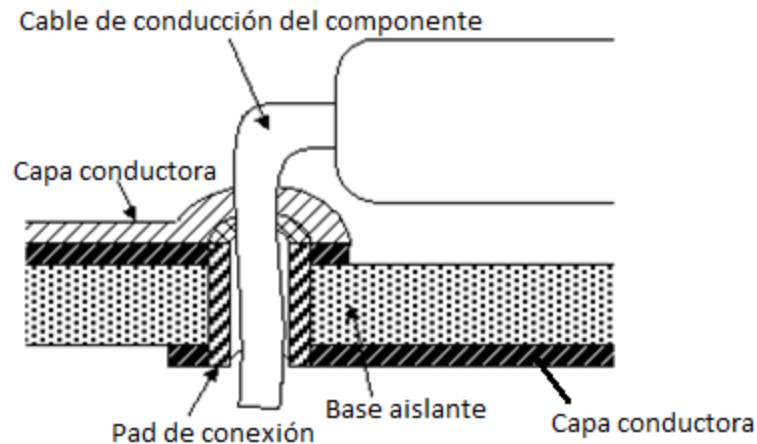
El ingeniero austriaco Paul Eisler es quizá el inventor del circuito impreso, cuando diseñó un circuito impreso para una radio en la compañía en la cual trabajaba con el fin de demostrar que su invento era totalmente viable en 1936, este invento revolucionó la industria tecnológica y en la segunda guerra mundial se empleó esta nueva tecnología para fabricar radios robustos en masa [34].

En 1956 se publica la patente de Moe Abramson y Stanilus F. Danko, de la *United States Army Signal*, donde se proporciona un método de montaje que combina las ventajas de unir componentes prefabricados con los circuitos conductores mediante patrones de interconexión para luego soldarlos [35].

Este último desarrollo es la base de la fabricación de los circuitos actuales con la diferencia que la soldadura se puede hacer automáticamente y el diseño se puede hacer mediante un software de diseño asistido por computador (CAD).

### 4.2.4.2. COMPOSICIÓN

Un PCB (*Printed Circuit Board*), es una superficie constituida por tres partes principales: La base aislante que sirve de soporte a los componentes, de una a dieciséis capas conductoras con caminos elaborados de algún material conductor se interconectan y los *pads* que sirven para hacer la conexión entre el componente y el camino conductor, existen dos tipos de *pads*; los *thru-hole* y los SMD (*Surface Mount Technology*) [36] .



**Figura 18.** Componentes de un PCB doble capa [36].

#### 4.2.4.3. DISEÑO ELECTRÓNICO AUTOMATIZADO (EDA)

Consiste en diseñar el circuito impreso con la ayuda de algún software, para distribuir e interconectar los componentes, estos programas ya tienen toda la información acerca de los componentes, como la distancia entre pines, el tipo de montaje y el tipo de empaquetado. Además ofrecen la posibilidad de automatizar ciertas tareas como el ruteado y la posición de los componentes, aclarando que esto no reemplaza la experiencia y conocimiento del diseñador [37].

Los siguientes son los pasos que comúnmente se realizan para hacer un PCB por medio de un EDA:

- Crear los circuitos esquemáticos (opcional): estos son una representación pictórica de un circuito electrónico y muestra la conexión real mediante cables entre los símbolos de los dispositivos [37].
- Convertir el circuito esquemático en una lista de pines y nodos (*net list*): en este paso se pueden observar los pines de los componentes (*Footprints*) y, si se realizó el paso anterior, líneas guías de la manera como se hacen las conexiones entre los componentes[37].
- Determinar la posición de cada componente[37].
- Ruteo[37].

Actualmente uno de los programas más usados a nivel profesional para el diseño electrónico automatizado es *Altium Designer*, que unifica el diseño de la plataforma física con el diseño de tarjeta de circuito impreso además de tener un avanzado gestor de librerías que permite descargar los componentes actualizados de acuerdo a su fabricante [38].

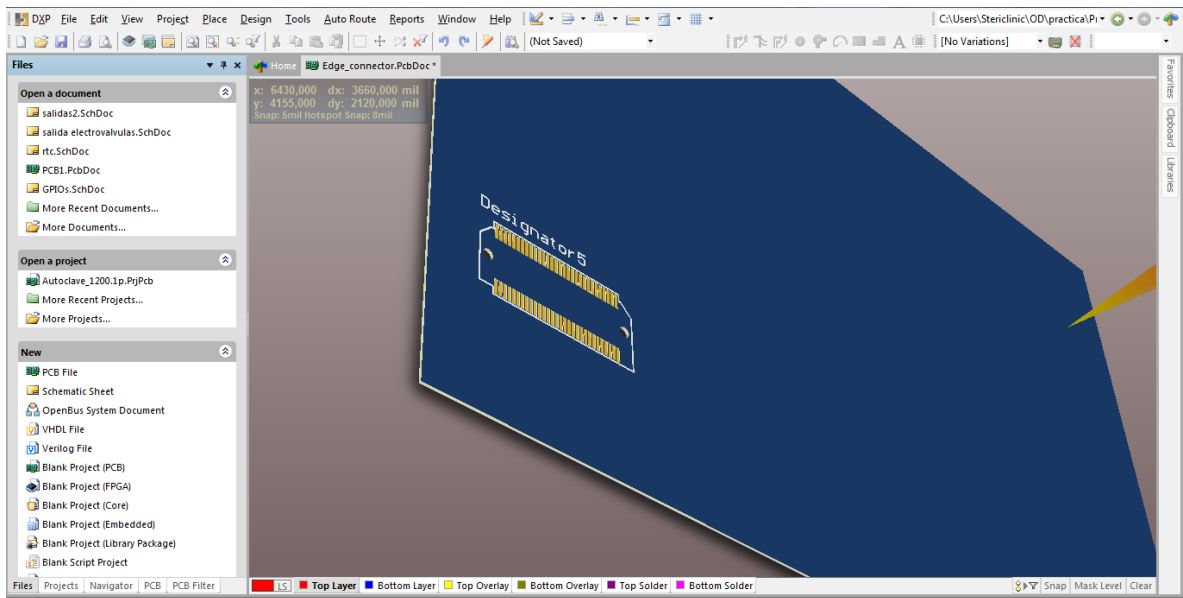


Figura 19. Entorno de Altium Designer.

## 5. DESARROLLO DEL PLAN DE TRABAJO

### 5.1. METODOLOGÍA EMPLEADA PARA EL DESARROLLO DEL PROYECTO.

En la siguiente figura se muestra la metodología que se empleó a lo largo del proyecto,

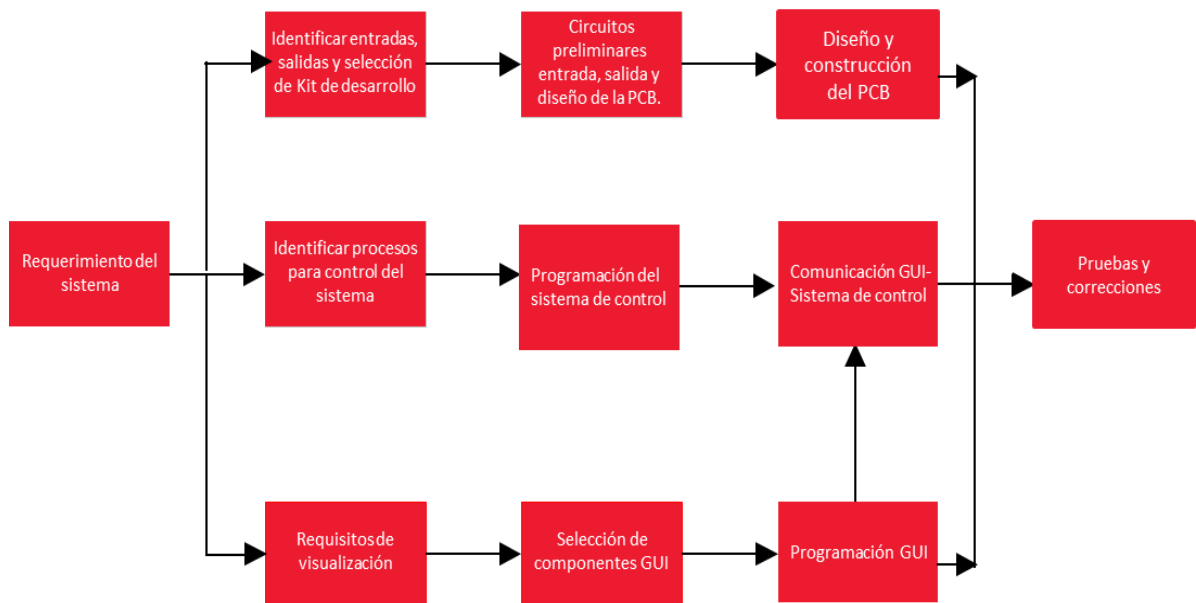


Figura 20. Metodología del plan de trabajo

A continuación se describe cada paso empleado en la metodología.

## 5.2. IDENTIFICACIÓN DE ENTRADAS Y SALIDAS DEL SISTEMA

Para lograr la identificación de entradas y salidas del sistema, se recibió una capacitación acerca del funcionamiento de una autoclave de 150 litros con compuerta automática y los diferentes elementos que conforman el sistema de control (actuadores, sensores, comunicación, interfaz gráfica, entre otros) y se plantearon las mejoras que debía tener el equipo en el nuevo sistema de control.

Los siguientes son los datos acerca de los componentes del sistema planteado:

- Número de entradas digitales: 9.
- Número de entradas analógicas: 4 (RTD, sensor de presión cámara, sensor presión generador de vapor), además de dejar una entrada analógica para cualquier labor de mantenimiento o validación a futuro.
- Número de salidas digitales: 12 de control.
- Controlador: Kit de desarrollo BEMICRO MAX 10.
- Pantalla LCD táctil de 7 pulgadas: utiliza 23 puertos como entrada-salida.

En la siguiente tabla se explica con detalle los elementos que conforman el sistema planteado:

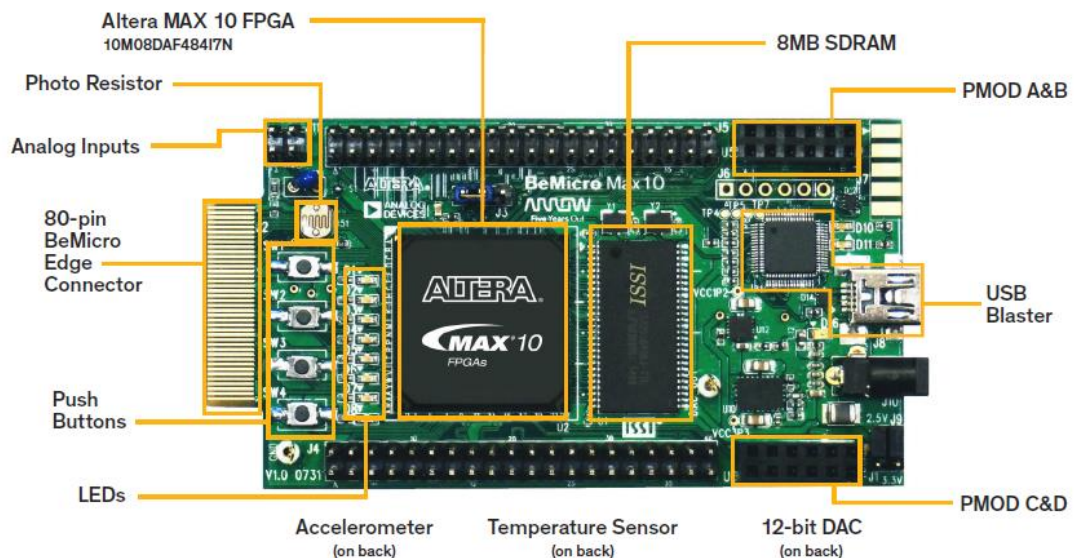
#	Elemento	Descripción
1	Kit de desarrollo "BEMICRO MAX 10"	FPGA
1	Pantalla LCD táctil de 7"	Interfaz HMI ( 23 E/S)
1	Socket para SD	Memoria del sistema
1	Pulsadores	Entrada digital (parada emergencia)
1	Interruptor	Entrada digital ( seguridad de puerta)
1	Sensor de nivel generador de vapor.	2 entradas digitales (nivel mínimo y máximo)
1	RTD PT-100	Entrada analógica ( temperatura cámara)
2	Sensor de presión.	Entradas analógicas ( presión cámara y presión generador de vapor)
3	Presostato	Entradas digitales ( presión de chaqueta, cámara y empaque)
2	Final de carrera	Entradas digitales (puerta abierta y puerta cerrada)
7	Electroválvulas	Salida con triacs.
1	Contactador	Salida con triacs. (Resistencias de calentamiento)
1	Alarma	Salida con triacs.
1	Variador de velocidad	2 Salidas con triacs ( contactores de abrir

		puerta y bajar puerta)
2	Contactor	Salida con triacs ( contactor de bomba de agua y bomba de vacío)

**Tabla 2.** Componentes del sistema de control

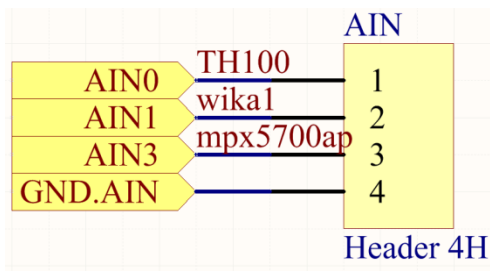
El kit de desarrollo “BEMICRO MAX 10” es una tarjeta de la empresa Arrow Electronics, con la principal característica de que a pesar de ser una tarjeta de bajo costo, tiene todo lo necesario para realizar sistemas de mediana complejidad [39].

Los componentes que conforman este kit de desarrollo se pueden ver en la siguiente imagen.



**Figura 21.** Composición de una BEMICRO MAX 10 [35].

Las figuras 22 y 23 muestran la manera como fueron utilizados los puertos del kit de desarrollo Bemicro MAX 10 en el proyecto, se debe tener en cuenta que la tarjeta cuenta con un puerto edge de 80 pines que se puede utilizar para expandir las posibilidades para desarrollos futuros.



**Figura 22.** Entradas analógicas en el kit Bemicro MAX10

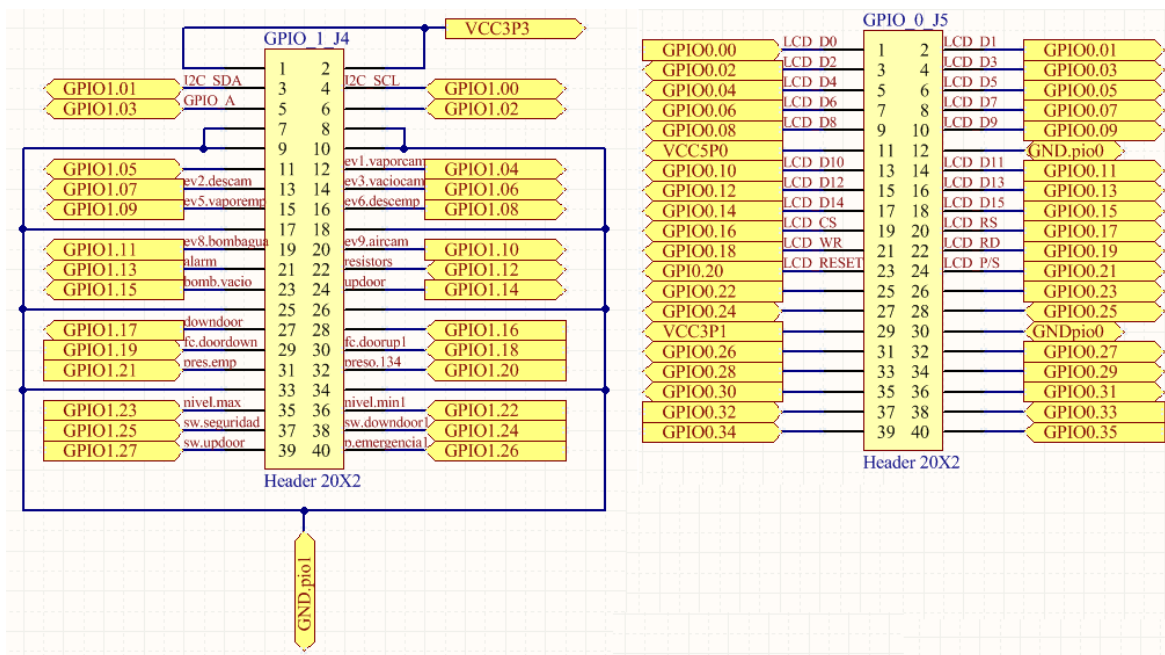


Figura 23. Entradas y salidas digitales en el kit Bemicro MAX10.

### 5.3. CIRCUITOS DE ADECUACIÓN DE ENTRADAS Y SALIDAS

#### 5.3.1. ACONDICIONAMIENTO DE LAS SEÑALES ANALÓGICAS

Se debe tener en cuenta que el rango de voltajes del ADC (*Analog to digital converter*) de la FPGA es de 0 a 3.3 Vdc, el siguiente es el circuito de acondicionamiento para la señal del TH100 que es el transmisor de la RTD PT-100 de la autoclave y envía una señal de 4-20 [mA], dicha señal pasa a través de una resistencia para obtener un voltaje en la entrada no inversora del amplificador operacional, en la entrada inversora llega la tensión de referencia como cero en tensión del sistema que se puede ajustar mediante RPOT.

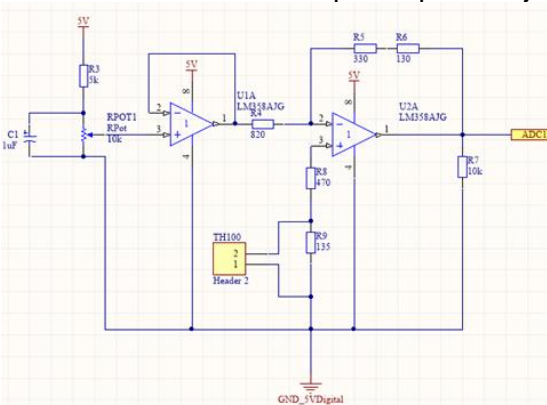


Figura 24. Acondicionamiento señal 4-20 mA.

Las siguientes imágenes muestran los circuitos de acondicionamiento de las señales de voltaje de los sensores de presión tanto de 0-10 [V] (sensor Wika A-10) como de 0-5 [V] (sensor MPX5700-ap) de salida. Estos circuitos constan de un divisor de voltaje que reduce el voltaje de entrada a un rango de 0 a 3.3 Vdc y de un buffer seguidor de voltaje para hacer acople de impedancia que evita que el circuito interno del ADC de la FPGA se vea como una carga para el divisor de voltaje y se entre en lecturas erróneas del dato de presión, por lo que si la señal del sensor de presión llegara con poca corriente, el circuito seguidor compensaría esa pérdida con la fuente de alimentación del amplificador operacional.

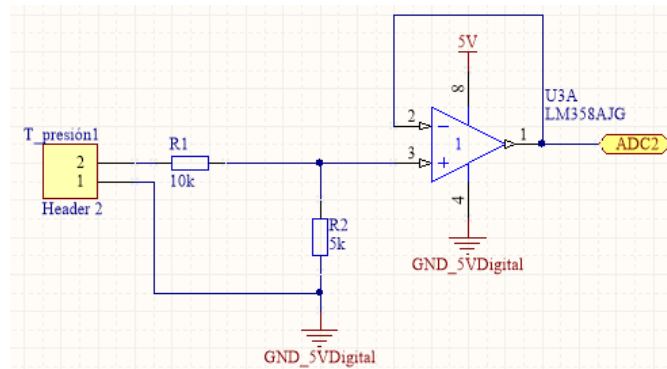


Figura 25. Acondicionamiento señal de 0-10 V.

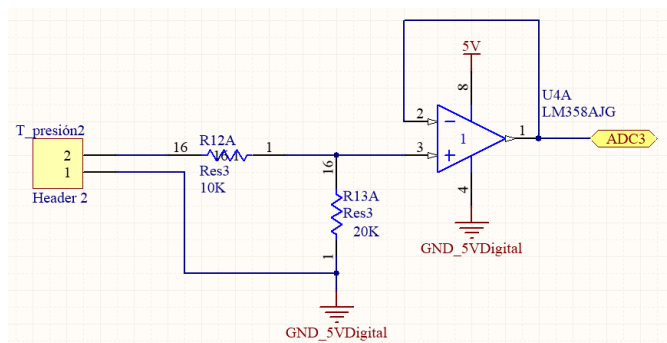


Figura 26. Acondicionamiento señal de 0-5 V.

### 5.3.2. ACONDICIONAMIENTO DE ENTRADAS Y SALIDAS DIGITALES.

El circuito que se utiliza para acondicionar las señales de entrada consta de un divisor de voltaje que se encarga de reducir las señales de entrada de 5 [V] en señales de 3.3 [V] que es el voltaje soportado por los puertos de entrada digitales de la FPGA garantizando que no hallan sobretensiones en sus entradas.

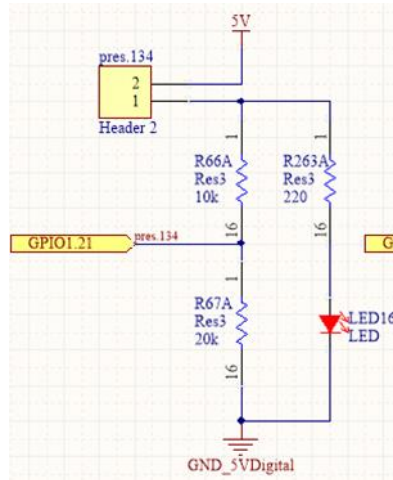


Figura 27. Acondicionamiento de entradas digitales.

El circuito de acondicionamiento de salidas digitales es un circuito con salida TRIAC el cual permite hacer conmutación con pequeñas corrientes de cargas de voltaje alterno de hasta 12 A, debido a que todas las cargas son inductivas (electroválvulas y contactores) este circuito cuenta con una red SNUBBER con el fin de suprimir los picos de voltajes y corrientes que se puedan generar al conmutar las cargas.

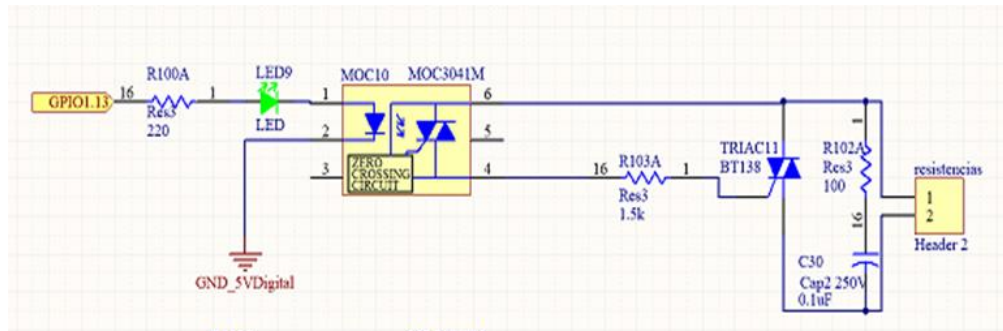


Figura 28. Acondicionamiento de salidas digitales.

#### 5.4. DISEÑO Y CONSTRUCCIÓN DEL PCB.

El diseño del PCB se realizó en el software *Altium Designer* con licencia de evaluación, de una capa para montajes tipo agujero pasante (THD) ya que esto permitió que a pesar de tener una tarjeta de tamaño grande, la fabricación de la tarjeta y el montaje de los componentes se hiciera en corto tiempo, sin embargo se plantea como trabajo futuro la posibilidad de usar un diseño de PCB doble capa con componentes de montaje superficial (SMD) para reducir el tamaño de la tarjeta.

El ancho de las pistas se calculó según la norma IPC-2221 ó IPC-D-275, tomando como base el valor de la corriente que va a circular por ella.

A continuación se muestran el diseño 2D y 3D de la tarjeta.

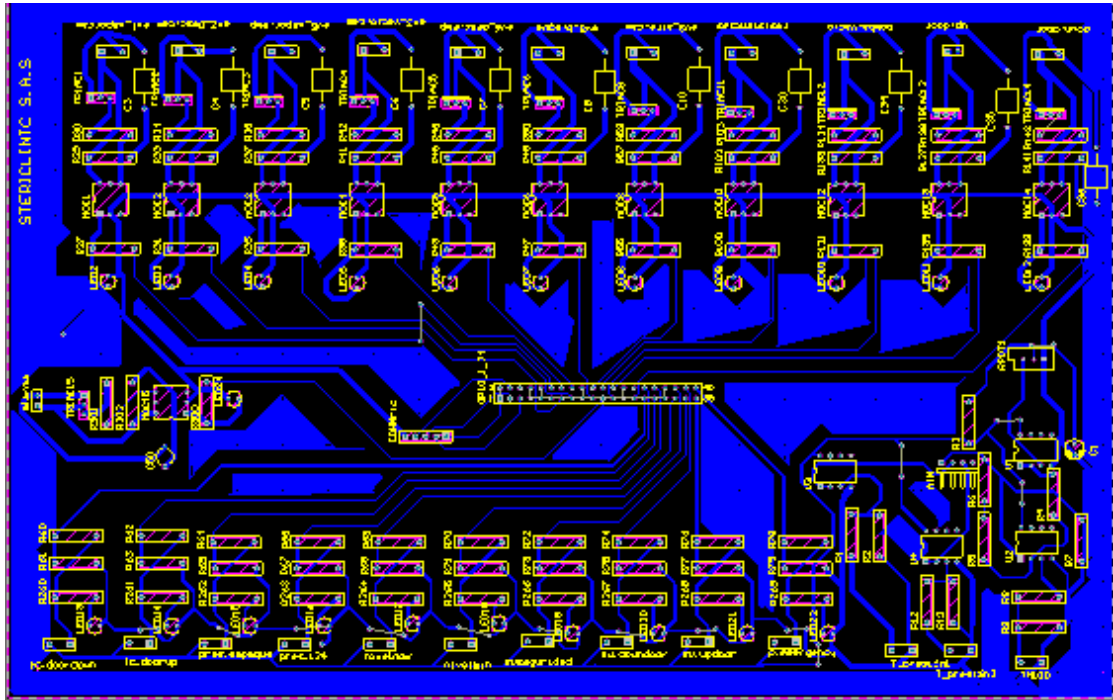


Figura 29. Diseño 2D del PCB del sistema.

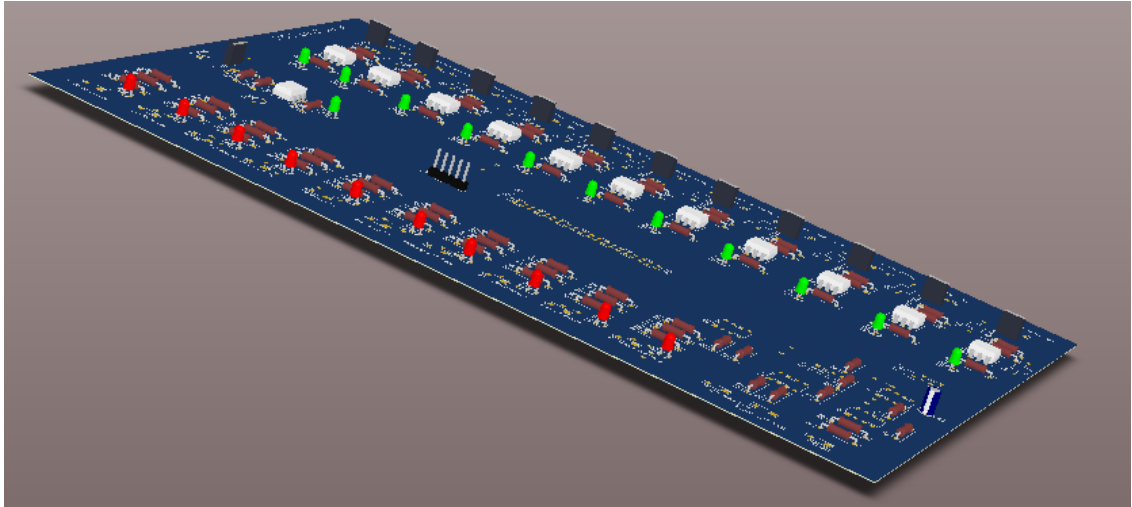


Figura 30. Diseño 3D del PCB del sistema.

Para la construcción del circuito impreso se contrató el servicio a una empresa local, la cual recomendó utilizar materiales de bajo costo para su realización, ya que al ser el primer diseño está sujeto a la posibilidad de realizarle muchos cambios.

La figura 31 muestra el resultado final de la construcción del PCB.

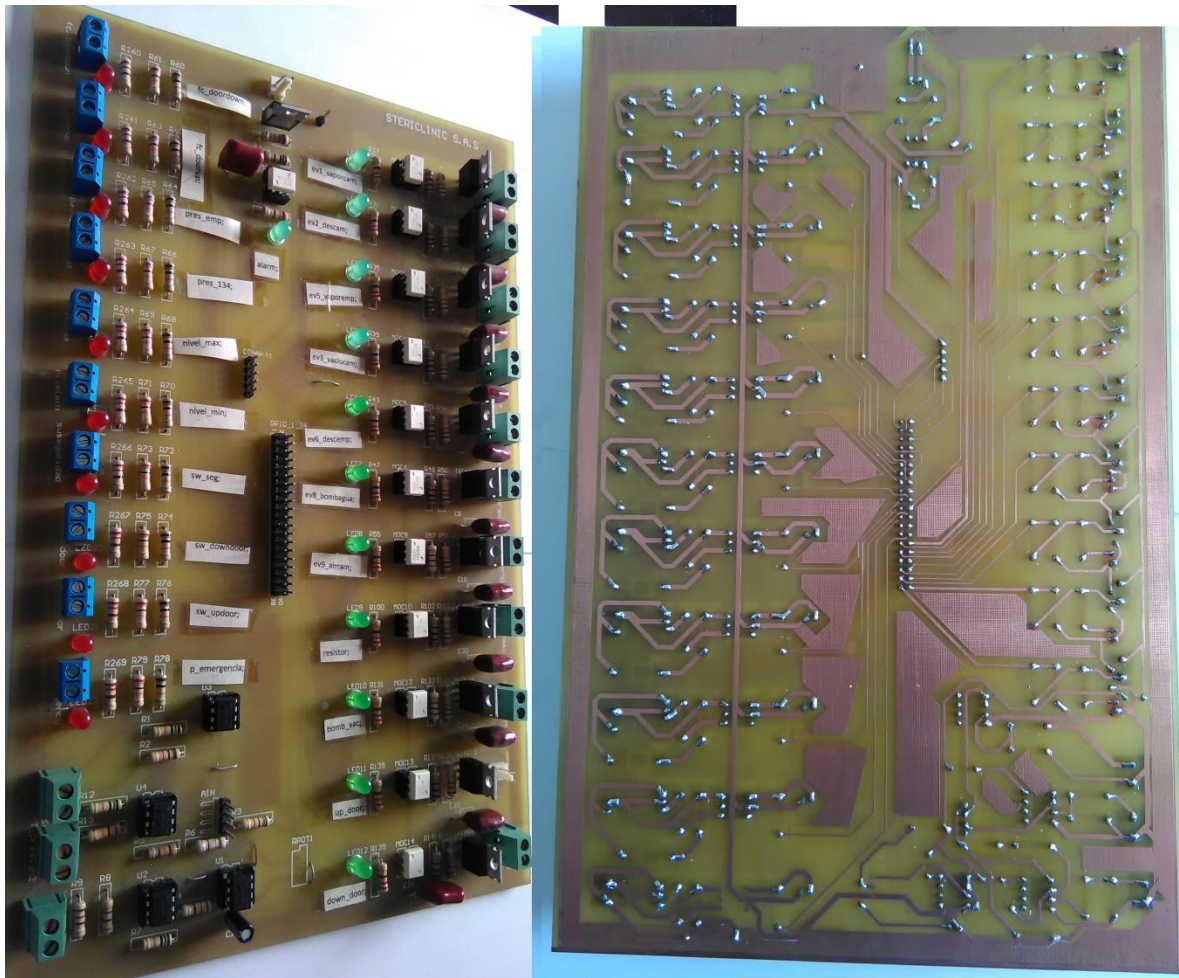


Figura 31.PCB construido.

## 5.5. IDENTIFICACIÓN DE LOS SISTEMAS DE CONTROL EN LOS PROCESOS DE UNA AUTOCLAVE

Al encender el equipo, se realiza el llenado del generador de vapor con agua limpia mediante una motobomba, el nivel de agua del generador de vapor es controlado por un sensor de nivel. Al alcanzar un nivel de agua adecuado se encienden las resistencias de calentamiento para realizar la generación de vapor, dicho vapor es transmitido a la chaqueta hasta que se alcancen 40 psi de presión, luego de esto el equipo queda habilitado para que el operario esterilice el material que requiera.

Una vez seleccionado el ciclo e introducido el material a esterilizar, el equipo sella la puerta mediante una inyección de vapor y realiza pre-vacíos mediante una bomba de vacío, una vez finalizada la etapa de pre-vacíos se inyecta vapor a la cámara hasta alcanzar la temperatura (121°C / 134 °C) y presión establecida según el ciclo elegido, esta temperatura y presión se mantienen durante un tiempo.

Al finalizar el tiempo de esterilización, se realiza un vacío con el fin de realizar la extracción del vapor de la cámara y se inyecta aire filtrado para realizar el secado de la carga. Finalmente se activa una alarma y se abre la puerta para realizar la extracción del material estéril.

La figura 32 muestra el comportamiento de la temperatura y presión en el proceso descrito anteriormente.

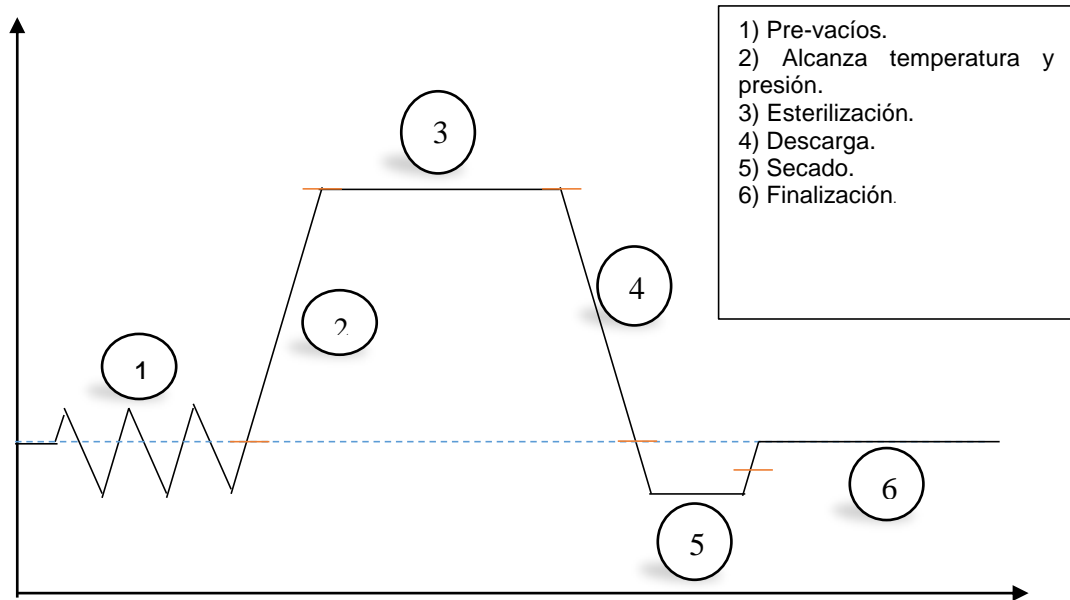
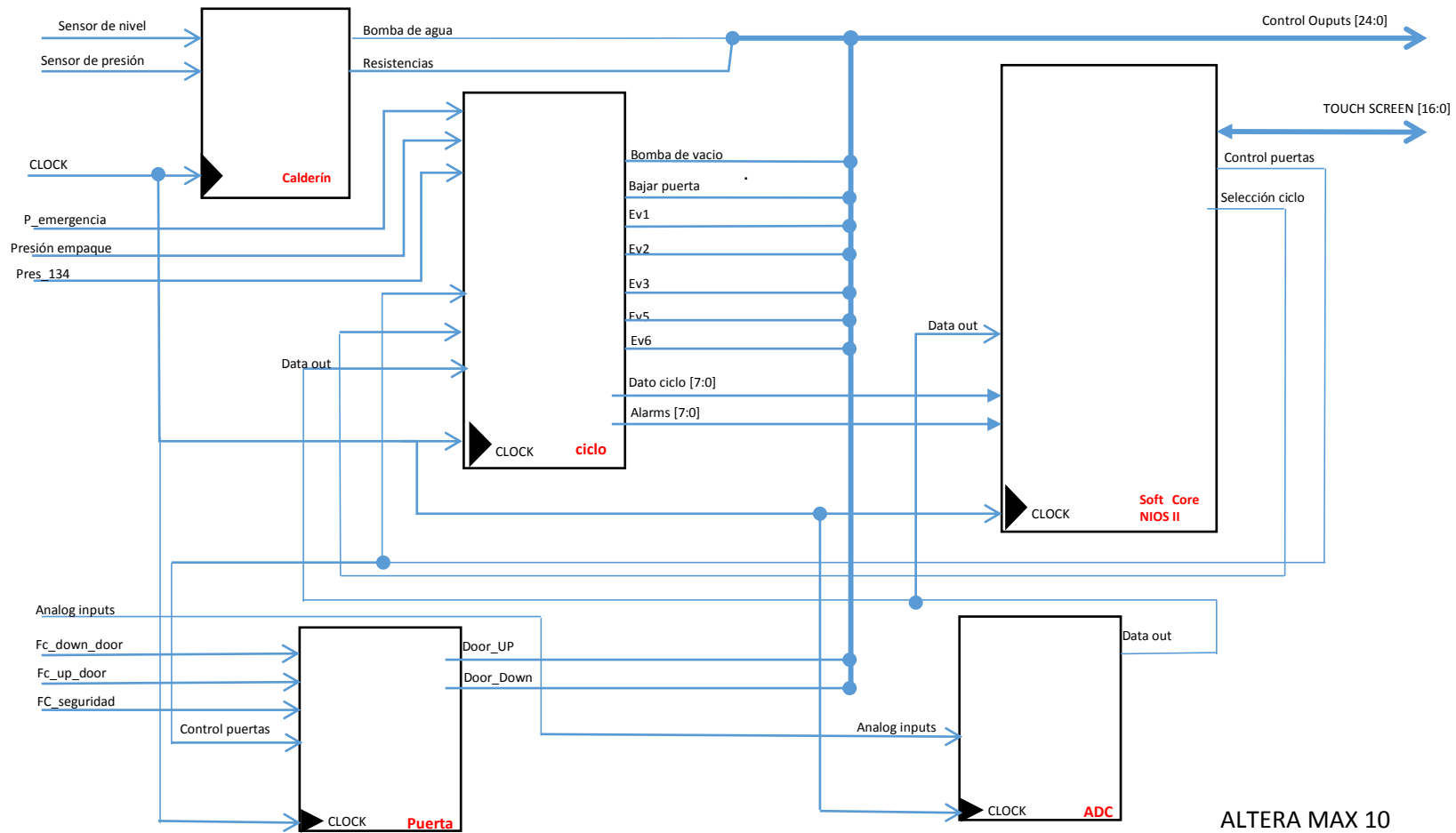


Figura 32. Etapas de un ciclo de esterilización.

## 5.6. PROGRAMACIÓN DE LOS PROCESOS DE CONTROL EN LA FPGA.

Para la programación del sistema de control, se implementaron cinco módulos principales: módulo calderín, módulo ciclo, módulo puerta, procesador y módulo ADC. Por motivos de confidencialidad de la empresa el código de programación de la FPGA no puede ser mostrado y por ello se utilizan herramientas como diagramas flujo y máquinas de estado para explicar el funcionamiento de los módulos.

El siguiente esquema muestra de una forma general, la manera como se implementaron los módulos en la FPGA.



ALTERA MAX 10

Figura 33. Módulos implementados en la FPGA.

Como se muestra en la figura 33, cada módulo está conectado con otro, además de compartir un reloj. A continuación se explica con detalle cómo funciona cada módulo.

### 5.6.1. MÓDULO CALDERÍN

Este módulo está encargado de hacer el llenado del calderín, su funcionamiento se describe en el siguiente diagrama de flujo.

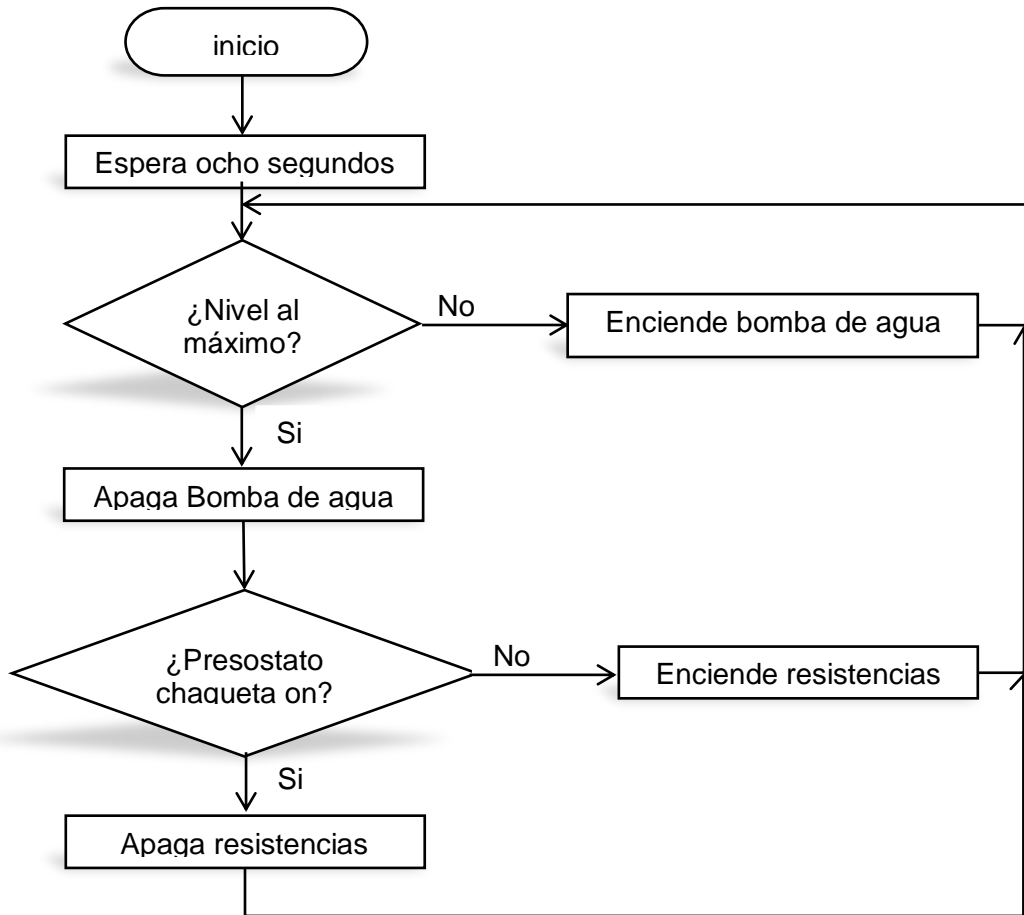


Figura 34. Diagrama de flujo módulo calderín.

El módulo calderín está encargado de hacer el llenado del calderín o generador de vapor con agua proveniente de un sistema de osmosis inverso, el llenado se controla mediante un sensor de nivel y la activación de las resistencias de calentamiento con un presostato que también es el encargado de hacer el control de histéresis, teniendo la precaución de no activar la bomba de agua y las resistencias al tiempo. Este módulo es completamente independiente y funciona en paralelo con el resto del sistema. El siguiente, es el circuito digital del módulo.

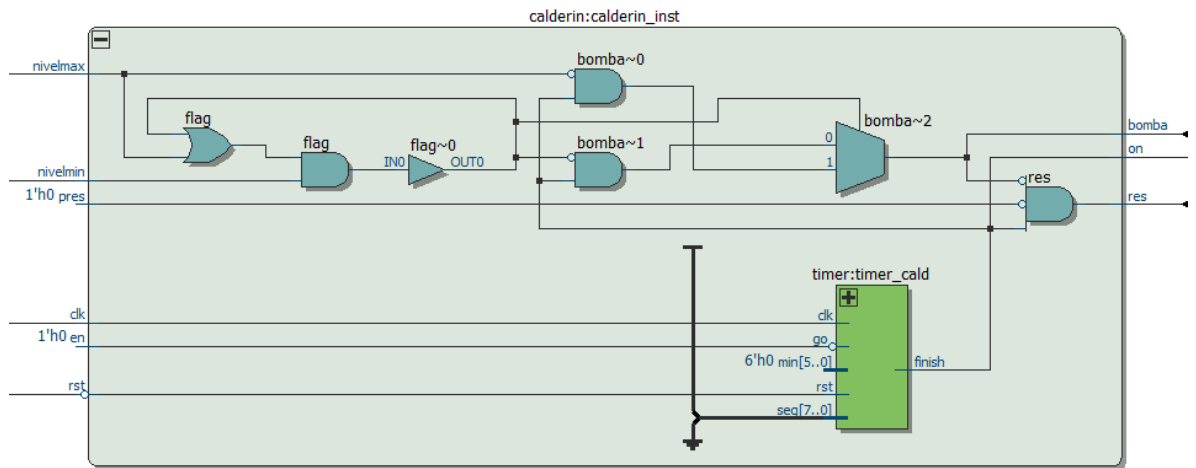


Figura 35. Circuito digital del módulo calderín.

### 5.6.2. MÓDULO PUERTA

Este módulo se encarga de hacer el control de la puerta (ver figura 36), la cual se puede abrir o cerrar cuando el operario lo desee a través de la interfaz gráfica siempre y cuando la autoclave no se encuentre realizando un ciclo, y se abre automáticamente al finalizar el ciclo. El siguiente es el diagrama de flujo del sistema.

Debido a la gran cantidad de elementos lógicos de este módulo, es difícil mostrar su circuito digital.

### 5.6.3. MÓDULO CICLO

Este módulo incorpora las diferentes etapas de un ciclo de esterilización (ver figura 31), controlando los tiempos, las entradas y salidas involucradas en el control de temperatura y presión de la cámara. Dentro del módulo ciclo, cada etapa del ciclo de esterilización es un módulo con una señal de inicio (*Idle*) y *finish*, de acuerdo a un orden pre-establecido.

La figura 37 muestra de manera general el circuito digital de este módulo.

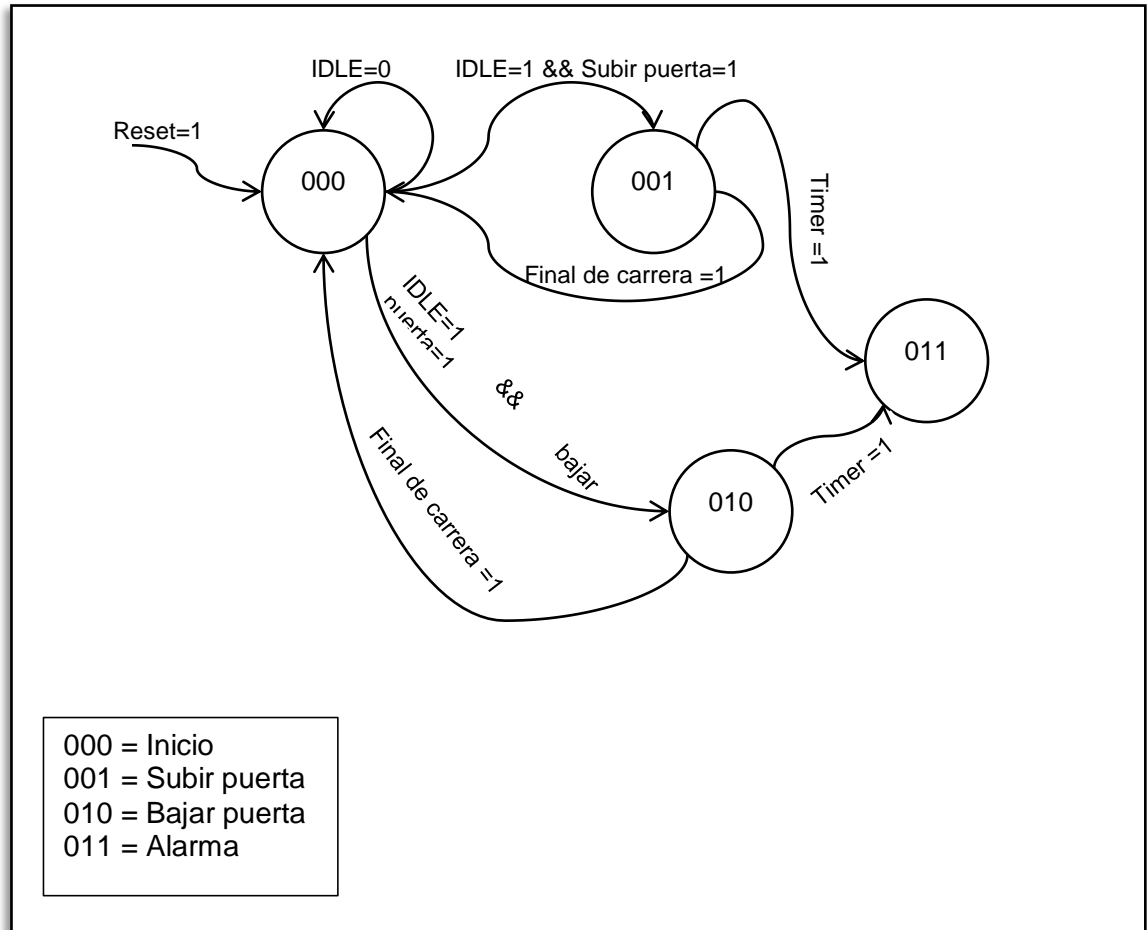


Figura 36. Máquina de estados módulo puerta

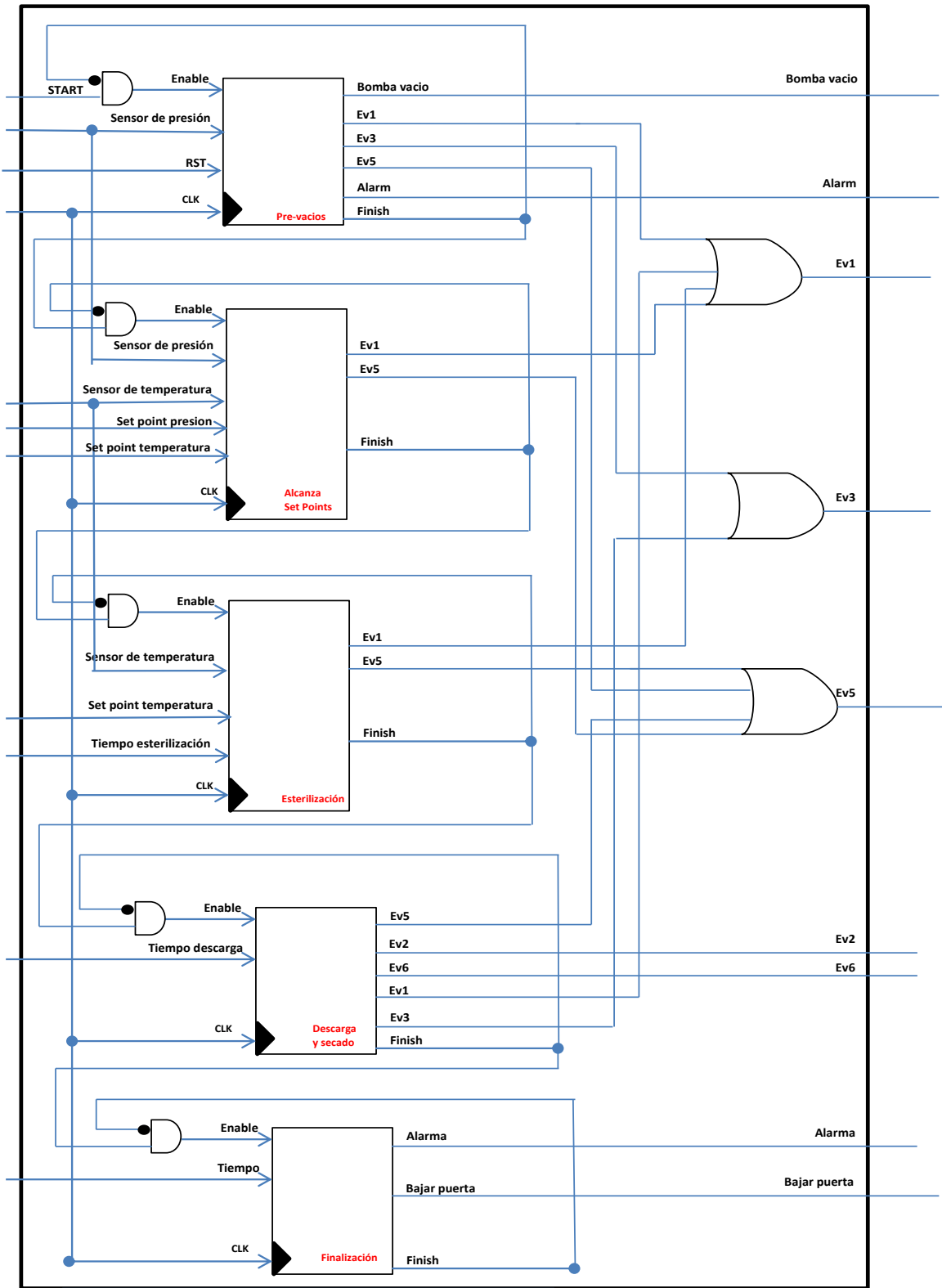


Figura 37. Circuito digital del módulo ciclo.

Cada módulo que se encuentra en la figura 36, tiene una lógica propia de acuerdo a las características de un ciclo de esterilización, a continuación se describe el comportamiento de cada módulo:

- **Módulo pre-vacíos:** con este módulo inicia el ciclo de esterilización, en primer lugar se inyecta presión al empaque hasta una presión establecida, si esta presión no llega durante un tiempo se activa una alarma, luego se hace un vacío mediante una bomba de vacío para después inyectar vapor a la cámara, el vacío y la inyección de vapor se hace tres veces seguidas y por último se envía una señal de finalización para indicar que los pre-vacíos han terminado. La figura 38 muestra la máquina de estados del módulo.

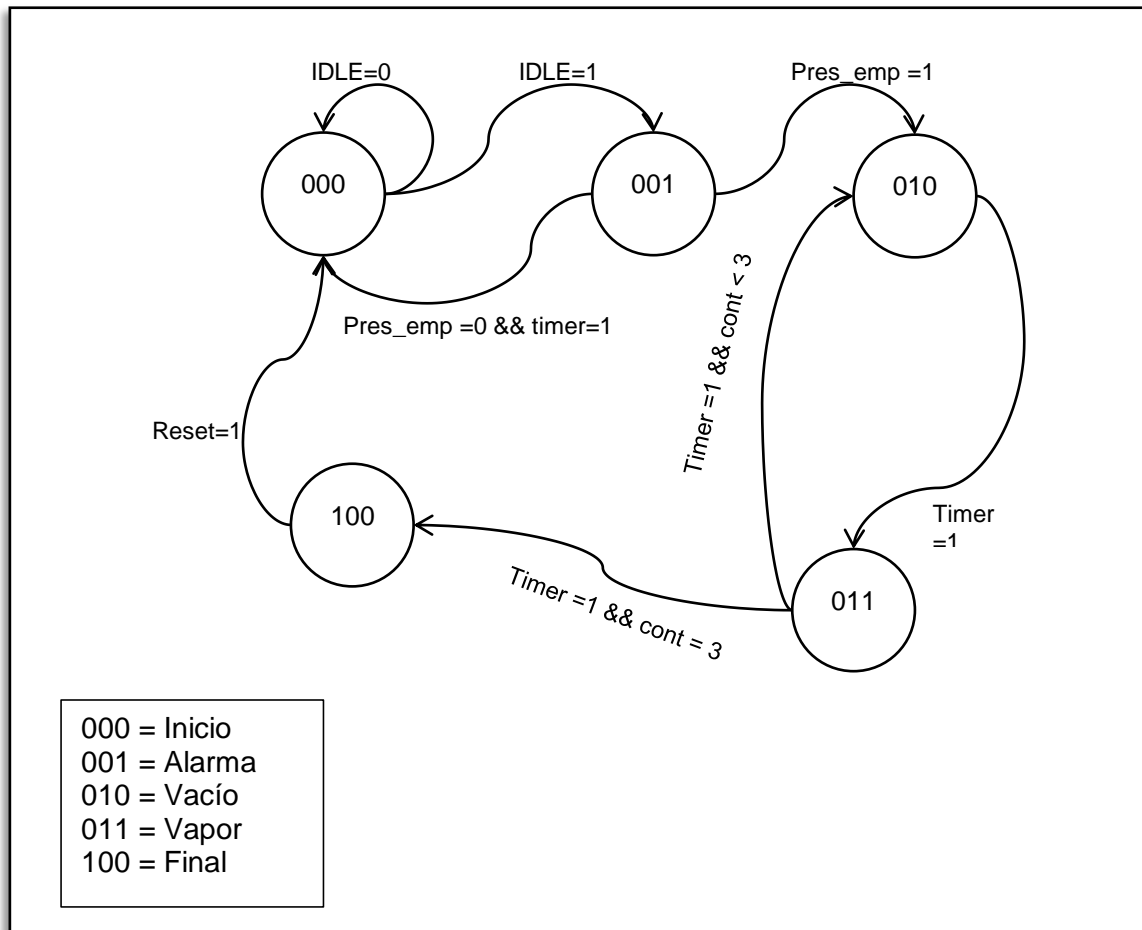


Figura 38. Máquina de estados módulo pre-vacíos.

- **Módulo Alcanza set-point:** este módulo se encarga de llevar la cámara a la temperatura y presión de esterilización, para ello se inyecta vapor constante a la cámara hasta alcanzar valores de temperatura y presión un poco menores a los finales, a partir de este momento se inyecta vapor mediante un control de PWM hasta alcanzar los valores apropiados para esterilizar la carga. La figura 39 muestra la máquina de estados del módulo.

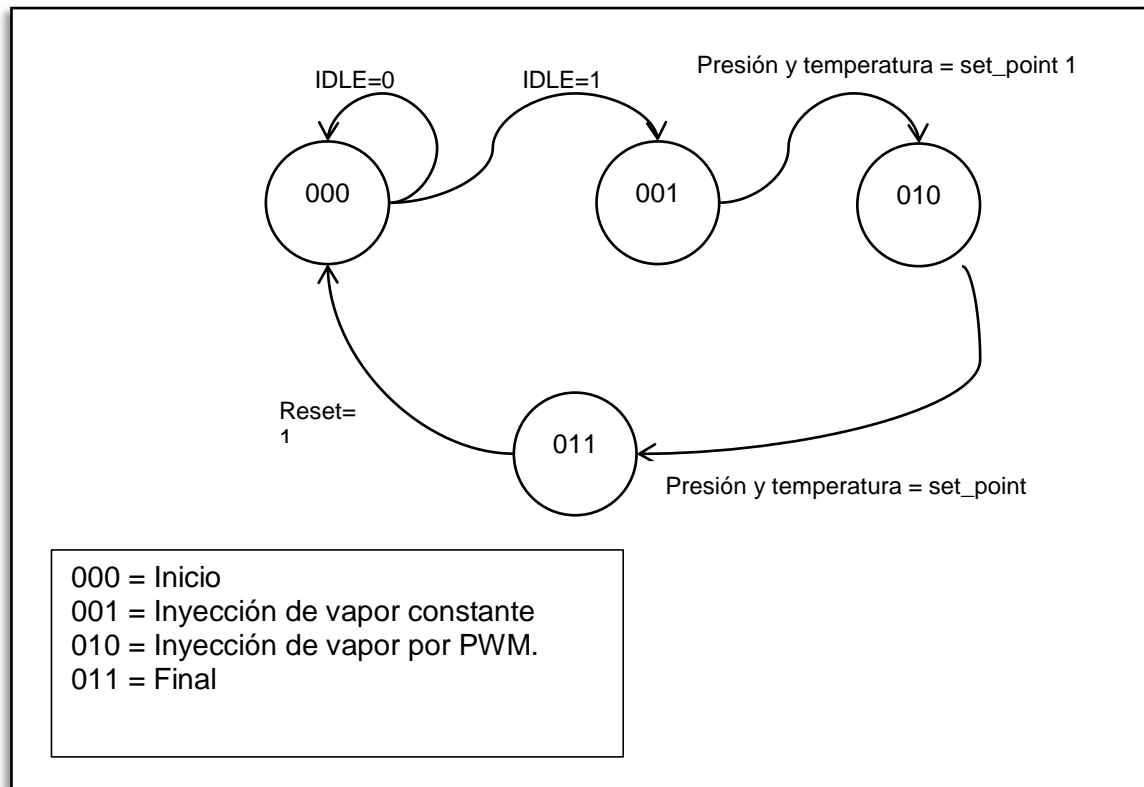
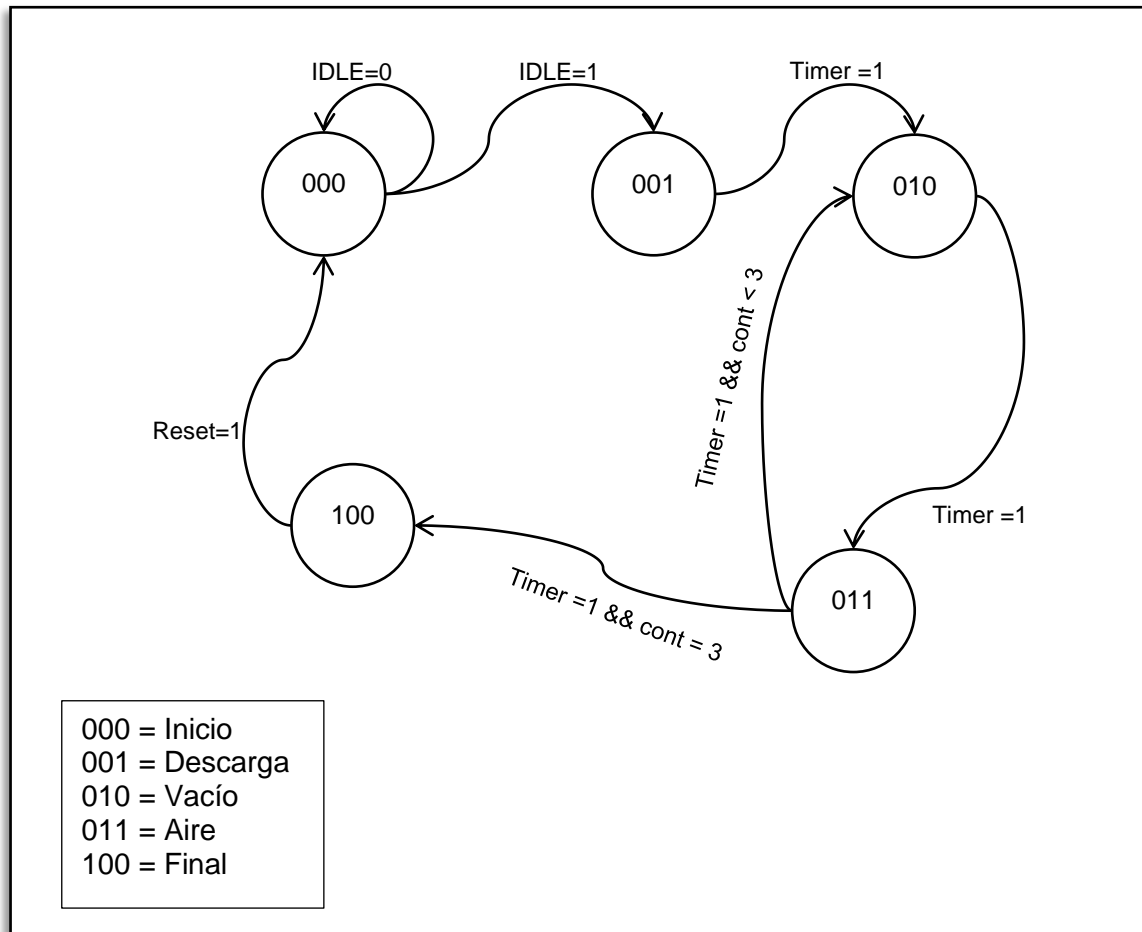


Figura 39. Máquina de estados módulo alcanza set-points.

- **Módulo esterilización:** este módulo lleva a cabo el control para mantener los valores de temperatura y presión estables durante el tiempo de esterilización.
- **Módulo de descarga y secado:** inicialmente este módulo realiza la descarga de la cámara y el empaque con el fin de liberar la presión positiva presente, luego realiza un vacío en la cámara con el fin de absorber la humedad de la carga y seguido inyecta aire limpio a la cámara para secarla, las operaciones de vacío e inyección de aire se repiten tres veces o más hasta dejar la carga completamente seca como lo muestra la figura 40.



**Figura 40.** Máquina de estados módulo descarga y secado.

- **Módulo de finalización:** inicialmente se realiza una última inyección de aire seguido por un vacío profundo para garantizar que la carga este completamente seca, seguido de esto se realiza la descarga de la cámara, empaque y se inyecta aire con el objetivo de igualar la presión de la cámara con la presión atmosférica, finalmente se entre abre la puerta y se deja lista la carga para ser retirada. La figura 41 muestra la máquina de estados del módulo.

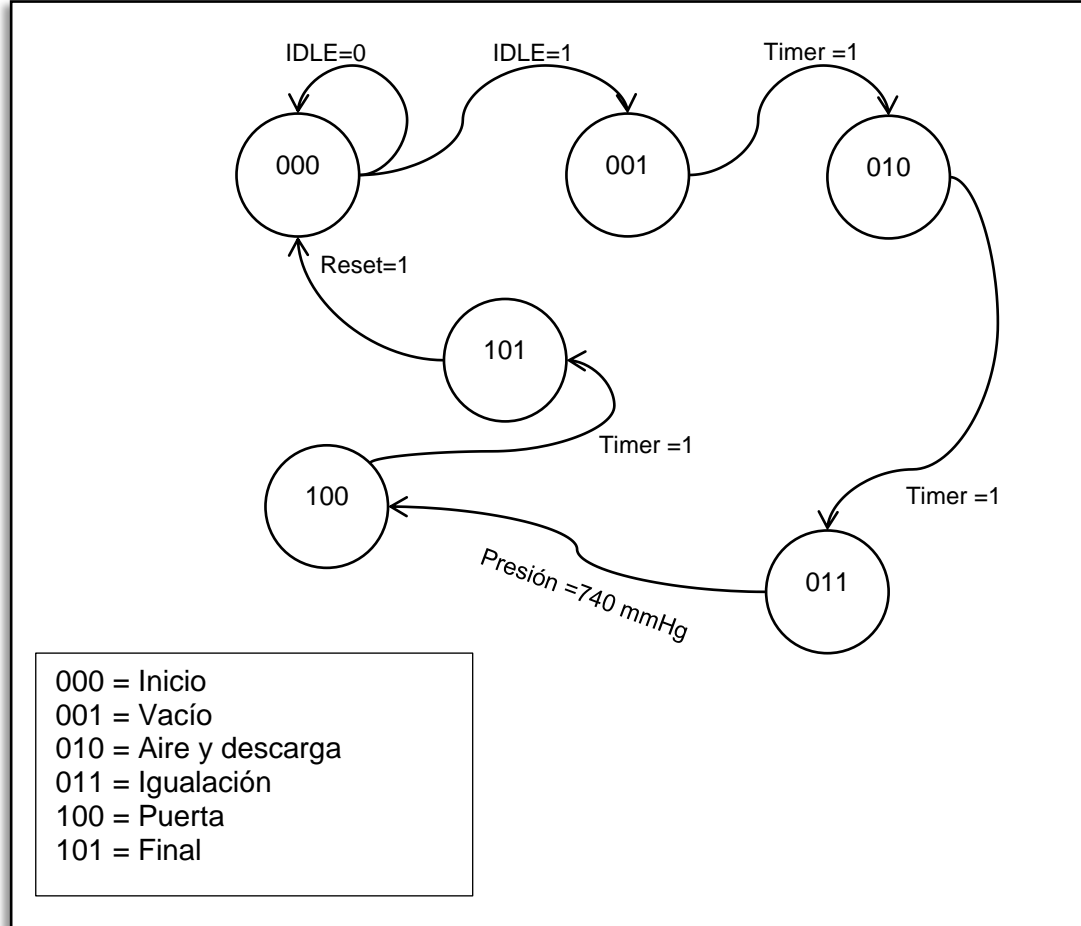


Figura 41. Máquina de estado módulo finalización.

#### 5.6.4. MÓDULO PROCESADOR

Este módulo consta de una CPU basada en un procesador NIOS II y se desarrolla bajo la herramienta QSYS del software Quartus II, Integrar un procesador en el diseño, hace que se tengan los beneficios que se tendrían utilizando cualquier otro procesador ( Arduino, PIC, entre otros) pero con la ventaja de poder implementar hardware en paralelo que nos ayude a bajar el consumo de energía y distribuir de mejor manera todos los procesos que requieran una cantidad mínima de tiempo de ejecución.

El procesador implementado, contiene los elementos que conforman un procesador normal, una CPU, una memoria RAM tipo SDRAM, una memoria ROM tipo Flash, un temporizador para controlar tiempos e interrupciones, un módulo de comunicación de clase JTAG tipo UART y un PLL para generar las diferentes frecuencias de reloj que necesita el sistema. La siguiente figura muestra cada elemento que conforma el procesador implementado.

Use	C...	Name	Description	Export	Clock	Base
<input checked="" type="checkbox"/>		<input type="checkbox"/> clk_50	Clock Source		<i>exported</i>	
<input checked="" type="checkbox"/>		<input type="checkbox"/> pll	Avalon ALTPLL		clk_50	
<input checked="" type="checkbox"/>		<input type="checkbox"/> cpu	Nios II Processor		pll_c0	0x0100_1800
<input checked="" type="checkbox"/>		<input type="checkbox"/> sdram	SDRAM Controller		pll_c0	0x0080_0000
<input checked="" type="checkbox"/>		<input type="checkbox"/> flash	BeMicro Max 10 - Serial Flash Controller		pll_c0	0x0100_1000
<input checked="" type="checkbox"/>		<input type="checkbox"/> LED	PIO (Parallel I/O)		clk_50	0x0100_20f0
<input checked="" type="checkbox"/>		<input type="checkbox"/> SW	PIO (Parallel I/O)		clk_50	0x0100_20e0
<input checked="" type="checkbox"/>		<input type="checkbox"/> timer	Interval Timer		clk_50	0x0100_2020
<input checked="" type="checkbox"/>		<input type="checkbox"/> jtag	JTAG UART		clk_50	0x0100_2108
<input checked="" type="checkbox"/>		<input type="checkbox"/> sysid	System ID Peripheral		pll_c0	0x0100_2100
<input checked="" type="checkbox"/>		<input type="checkbox"/> address	PIO (Parallel I/O)		pll_c0	0x0100_20d0
<input checked="" type="checkbox"/>		<input type="checkbox"/> data_read	PIO (Parallel I/O)		pll_c0	0x0100_20c0
<input checked="" type="checkbox"/>		<input type="checkbox"/> data_write	PIO (Parallel I/O)		pll_c0	0x0100_20b0
<input checked="" type="checkbox"/>		<input type="checkbox"/> lcd_cs	PIO (Parallel I/O)		pll_c0	0x0100_20a0
<input checked="" type="checkbox"/>		<input type="checkbox"/> lcd_rs	PIO (Parallel I/O)		pll_c0	0x0100_2080
<input checked="" type="checkbox"/>		<input type="checkbox"/> touch_irq	PIO (Parallel I/O)		pll_c0	0x0100_2070
<input checked="" type="checkbox"/>		<input type="checkbox"/> reset	PIO (Parallel I/O)		pll_c0	0x0100_2060
<input checked="" type="checkbox"/>		<input type="checkbox"/> lcd_wr	PIO (Parallel I/O)		pll_c0	0x0100_2050
<input checked="" type="checkbox"/>		<input type="checkbox"/> lcd_data	PIO (Parallel I/O)		pll_c0	0x0100_2040
<input checked="" type="checkbox"/>		<input type="checkbox"/> lcd_rd	PIO (Parallel I/O)		pll_c0	0x0000_0000

Figura 42. Elementos del Procesador basado en NIOS II.

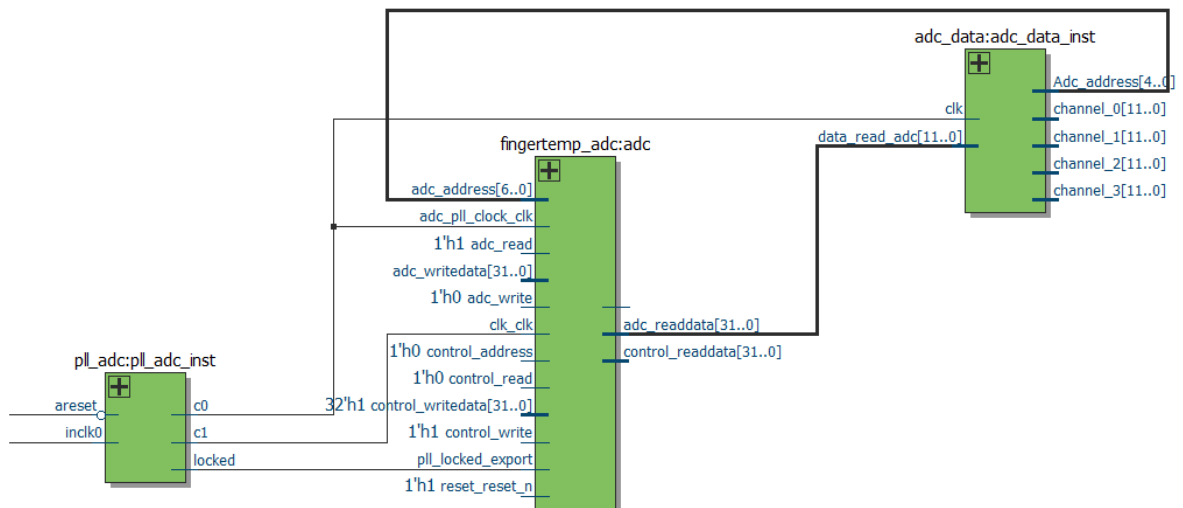
El software que se ejecutará en el sistema Nios II debe ser capaz de mostrar en la pantalla LCD diferentes parámetros del proceso de esterilización, además debe ser capaz de recibir las señales dadas por el panel táctil para hacer interacción con el usuario por medio de los botones puestos en pantalla, según estos requerimientos se adecuan los siguientes periféricos al sistema Nios II:

- Comunicación con los procesos de control: el sistema NIOS se comunica con los procesos de control mediante la ejecución de registros que envían a través del puerto address, dependiendo del registro el sistema NIOS escribirá un dato en el sistema de control o lo leerá por medio de los puertos data\_read o data\_write.
- Comunicación con la pantalla: se realiza la comunicación de la pantalla por medio de los periféricos lcd\_data, lcd\_rd, lcd\_cs, lcd\_rs, lcd\_wr, como lo requiere el chip RA8875 en la figura 12.

### 5.6.5. MÓDULO ADC

Este módulo es el encargado de recibir las señales analógicas de los sensores de presión de la cámara, de presión del calderín y de temperatura de la cámara, estas señales han sido previamente acondicionadas para ser recibidas de una manera adecuada en la FPGA.

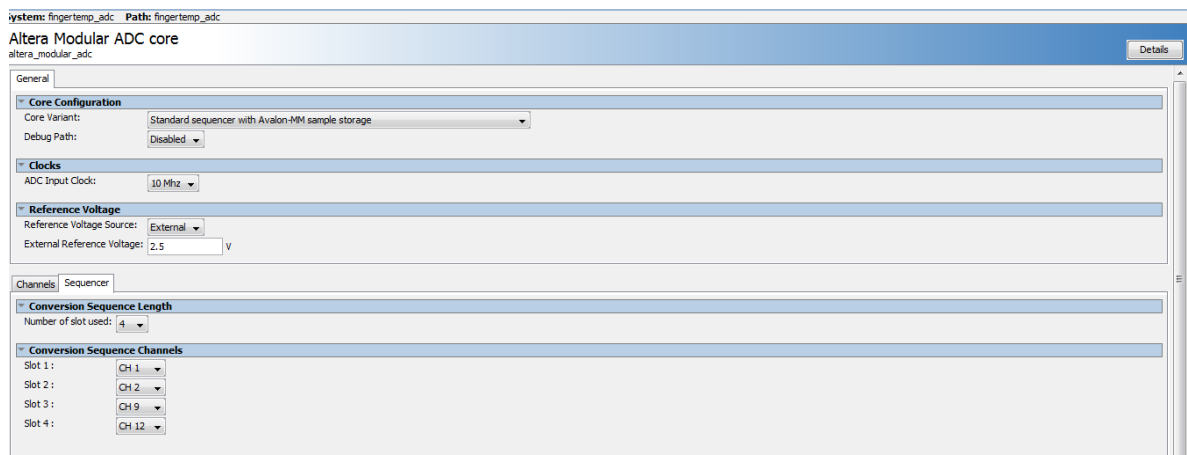
El módulo ADC está compuesto de tres componentes principales que son: un PLL (*Phase Locked Loop*), un sistema desarrollado bajo la herramienta de integración de sistemas QSYS, que utiliza el IP Altera modular ADC core y un módulo encargado de enviar los registros de instrucción ADC core y de-multiplexar los datos provenientes del mismo en los canales deseados. A continuación muestra el circuito de conexión de estos módulos.



**Figura 43.**Componentes que conforman el módulo ADC.

El PLL se encarga de recibir una señal de reloj de 50 Mhz y enviar dos señales de reloj enganchadas en fase al core ADC, una de 50 Mhz que es la señal de funcionamiento de los módulos y la otra de 10 Mhz equivalente a la frecuencia de muestreo del ADC.

El módulo llamado fingertem\_adc:adc contiene el Ip\_Core de altera llamado Altera Modular ADC core, este módulo se encarga de configurar el ADC interno de la FPGA Altera Max 10 de acuerdo a los parámetros que el usuario establezca, para este caso se utiliza una frecuencia de muestreo de 10 Mhz y el modo de secuencia estándar con almacenamiento de muestras Avalon MM, lo que nos permite utilizar la memoria interna de la FPGA para hacer un almacenamiento de las muestras que tome el ADC y luego hacer una lectura de estas muestras mediante la lógica implementada en el módulo adc\_data. La figura 44 muestra la configuración del ADC y los canales implementados en el diseño y la figura 45 su implementación en Qsys.



**Figura 44.** Configuración del IP Altera Modular ADC core.

Component	Name	Description	Export	Signal
clk_0	clk_in	Clock Source	clk	<i>exported</i>
	clk_in_reset	Reset Input	reset	
	clk	Clock Output	<i>Double-click to export</i>	clk_0
	clk_reset	Reset Output	<i>Double-click to export</i>	
ADC_CORE	clock	Clock Input	<i>Double-click to export</i>	clk_0
	reset_sink	Reset Input	<i>Double-click to export</i>	[clock]
	adc_pll_clock	Clock Input	adc_pll_clock	<i>exported</i>
	adc_pll_locked	Conduit	pll_locked	
	sequencer_csr	Avalon Memory Mapped Slave	control	[clock]
	sample_store_csr	Avalon Memory Mapped Slave	adc	[clock]
	sample_store_irq	Interrupt Sender	adc_irq	[clock]

Figura 45. Conexión del IP Altera Modular ADC Core en Qsys.

El módulo `adc_data` consta de un divisor de frecuencia que se encarga de enviar un reloj con una frecuencia 12 veces menor llamado `clk_output`, cada flanco positivo de este reloj cambia el canal del que se toma la muestra, esto se hace cíclicamente y mediante una de-multiplexación se envía a una salida diferente la muestra dependiendo del canal que se esté tomando. La siguiente imagen muestra el circuito digital del módulo

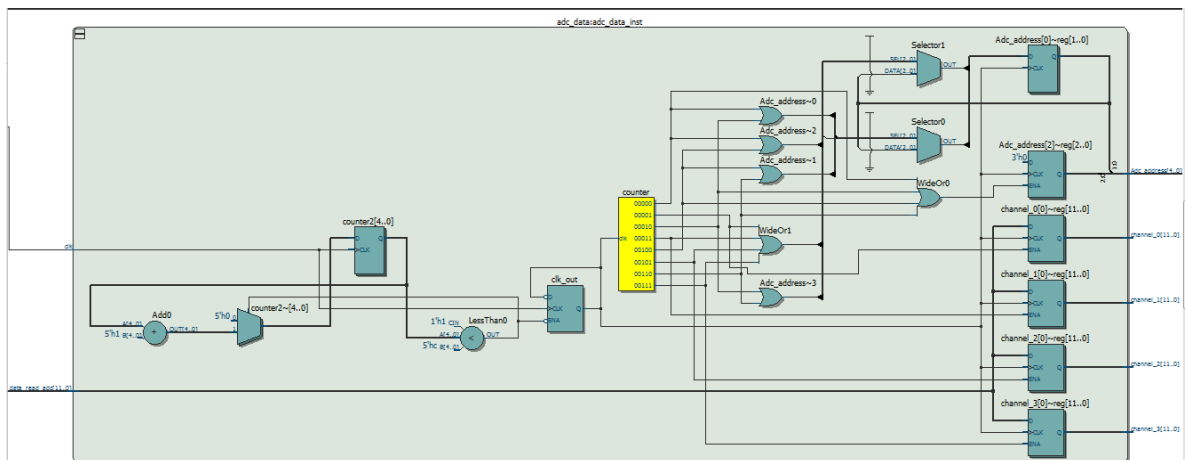


Figura 46. Circuito digital del módulo `adc_data`.

### 5.6.6. RESUMEN DEL SISTEMA HDL IMPLEMENTADO EN LA FPGA ALTERA MAX 10

La herramienta Quartus II permite obtener diferentes reportes acerca de todos los módulos descritos anteriormente al ser implementados en la FPGA, debido a la importancia de conocer cuántos recursos pueden llegar a ocupar la descripción de hardware descrita anteriormente. La figura 47 nos muestra el número de elementos lógicos necesarios para implementar el sistema que en este caso fue del 83%, utilizando el 83% de los pines y el 11% de la memoria total embebida.

Family	MAX 10
Device	10M08DAF484C8GES
Timing Models	Preliminary
Total logic elements	6,686 / 8,064 ( 83 % )
Total combinational functions	5,612 / 8,064 ( 70 % )
Dedicated logic registers	3,661 / 8,064 ( 45 % )
Total registers	3729
Total pins	208 / 250 ( 83 % )
Total virtual pins	0
Total memory bits	41,472 / 387,072 ( 11 % )
Embedded Multiplier 9-bit elements	0 / 48 ( 0 % )
Total PLLs	2 / 2 ( 100 % )
UFM blocks	0 / 1 ( 0 % )
ADC blocks	1 / 1 ( 100 % )

**Figura 47.** Reporte de compilación del HDL implementado.

*Quartus II* nos permite conocer el estimado del consumo de potencia de nuestro sistema, las condiciones de frecuencia, el número de elementos lógicos activos y la disipación de potencia en la FPGA puede variar y afectar la temperatura interna del sistema que suministra la energía a la FPGA. Según la herramienta PowerPlay Analyser de Altera el consumo de potencia del sistema incorporado es de aproximadamente 4W como se muestra en la figura 48.

Total Thermal Power Dissipation	4454.99 mW
Core Dynamic Thermal Power Dissipation	1.14 mW
Core Static Thermal Power Dissipation	62.49 mW
I/O Thermal Power Dissipation	4391.36 mW
Power Estimation Confidence	Low: user provided insufficient toggle rate data

**Figura 48.** Consumo de potencia del HDL implementado.

Estos análisis nos permiten hacer un análisis del sistema y obtener conclusiones acerca de que modificaciones se pueden hacer a futuro para mejorar el sistema, que nuevos módulos se pueden añadir y como acelerar el código implementado para obtener respuestas más rápidas con menos disipación de potencia.

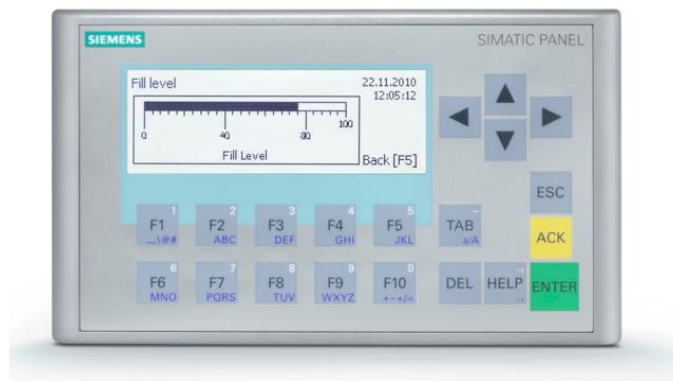
Una vez realizada toda la descripción de hardware de los sistemas de control en la FPGA, se procede a programar la FPGA por medio de la interfaz JTAG utilizando el programador de Quartus II, y con esto tenemos el sistema listo, con un procesador basado en NIOS II con el cual se puede ejecutar cualquier software desarrollado en lenguaje C/C++ en la interfaz de desarrollo NIOS II Software Build Tools For Eclipse.

## 5.7. SISTEMA DE VISUALIZACIÓN

### 5.7.1. REQUERIMIENTOS

Se realizó la identificación del sistema actual de visualización la cual es un panel KP300 de Siemens (ver figura 49) con las siguientes características [40] :

- Resolución de 240 x 80 pixeles.
- 10 teclas de sistema y 10 teclas de función.
- Iluminación posterior de colores blanco, verde, rojo y amarillo.
- Interfaz PROFINET.
- Programable con TIA portal versión 11 en adelante.
- Calidad y robustez con un grado de protección IP65.
- Conexión con PLC S7-300 en adelante.



**Figura 49.** Pantalla KP-300 de Siemens [40].

Con base en esta identificación, se realizó el estudio con la ayuda de personal de la empresa acerca de los requerimientos que debe tener el nuevo sistema de visualización y se hicieron los siguientes requerimientos:

- Tamaño de la pantalla de 7 pulgadas para adecuarla al panel actual que tiene el mismo tamaño (pantalla KP300 de Siemens).
- Pantalla a color con el objetivo de mejorar el actual sistema que es monocromático.
- Resolución superior a la actual.
- Panel táctil ya que el actual sistema de botones es anticuado y difícil de utilizar.
- Económico con el fin de reducir costos actuales de fabricación del equipo.
- Con unidad de procesamiento propio para no incrementar el gasto de elementos lógicos en la FPGA al tener que acelerar por hardware los algoritmos de visualización.

### 5.7.2. SELECCIÓN DEL SISTEMA DE VISUALIZACIÓN

Teniendo en cuenta los requerimientos que debe tener el sistema de visualización se hace la selección de una pantalla de siete pulgadas de la empresa WaveShare (ver figura 50) con las características mostradas en la tabla 3.



Figura 50. Waveshare 7 inch Resistive touch LCD [41].

Tipo de LCD	TFT
Interfaz	8080 Series Interface
Controlador del LCD	RA8875
Tipo de panel táctil	Resistivo
Profundidad de color	256/16K
Luz de fondo	LED
Tamaño del display (mm)	154.08 (W) × 85.92 (H)
Paso de punto (mm)	0.0642 (W) × 0.1790 (H)
Relación de aspecto	8:5
Resolución	800 * 480 (Píxeles)
Alimentación	3.0V~3.6V
Corriente de luz de fondo	No definido
Temperatura de operación	No definido
Compatibilidad	MCU sin controlador LCD

Tabla 3. Características del sistema de visualización elegido [41].

## 5.8. PROGRAMACIÓN DE LA INTERFAZ GRÁFICA DE USUARIO

La interfaz gráfica de usuario es la encargada de hacer la interacción entre el operario y la autoclave mediante botones, datos y gráficas previamente programadas. Para ello se utilizó el sistema basado en NIOS II implementado en el capítulo 4.2.1.1 mediante los periféricos `lcd_data`, `lcd_rd`, `lcd_cs`, `lcd_rs`, `lcd_wr`.

En la web del fabricante de la pantalla es posible encontrar las librerías del chip RA8875 que nos permiten realizar todas las funciones permitidas por este chip, como dibujar figuras geométricas, escribir texto, dibujar gráficas, leer la posición donde la membrana táctil fue pulsada entre otras. Estas librerías son descritas en el lenguaje de programación C/C++ y por ello fueron implementadas en el sistema NIOS II que se programa en el mismo lenguaje.

### 5.8.1. LIBRERIAS UTILIZADAS

A continuación se describen las librerías utilizadas para realizar la interfaz gráfica de usuario:

- **LCD\_Interface** : Es la encargada de realizar la escritura de los comandos y registros mediante los puertos de la FPGA, contiene las instrucciones `LCD_CmdWrite()` y `LCD_DataWrite()` para realizar la escritura y configuración de los registros y `LCD_CmdRead()` y `LCD_DataRead()` para la lectura de la configuración de algún registro en el RA8875 de acuerdo al siguiente diagrama:

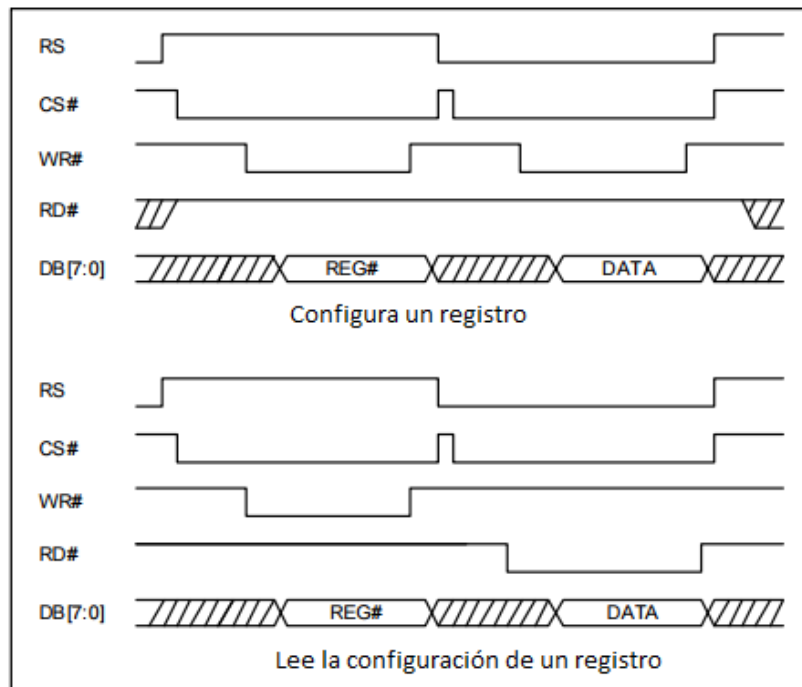


Figura 51. Lectura y escritura de un registro. [28]

- **RA8875\_Driver:** Contiene la configuración de todas las instrucciones que se pueden realizar en el RA8875 de acuerdo a la hoja de datos como dibujar un círculo (ver figura 52), encender la pantalla, configurar la pantalla, mediante diferentes funciones. Por ejemplo, en el siguiente código se muestra la función encender pantalla y apagar:

```

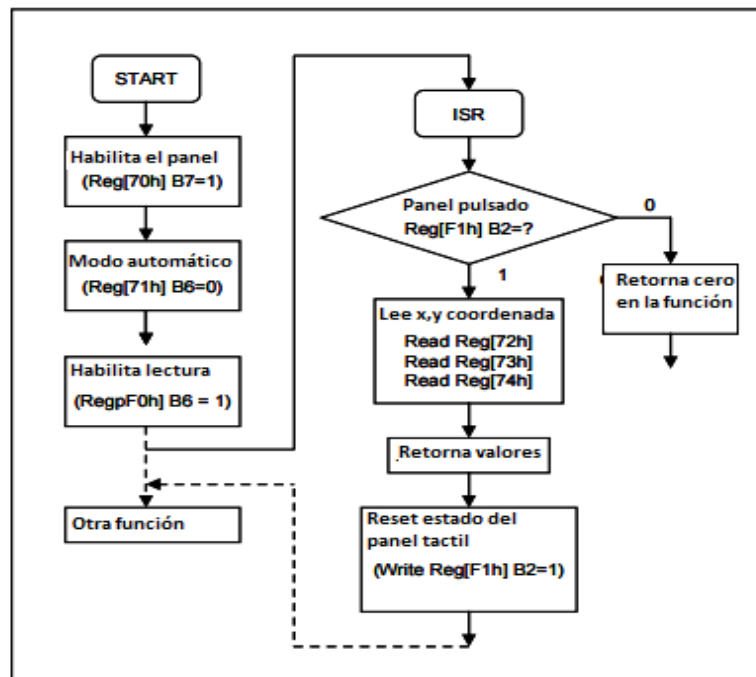
void Display_ON(void)
{
    LCD_CmdWrite(0x01); //Registro encender/apagar pantalla
    LCD_DataWrite(0x80); //Configura el registro en 0x80 para encenderla
}

void Display_OFF(void)
{
    LCD_CmdWrite(0x01); //Registro encender/apagar pantalla
    LCD_DataWrite(0x00); //Configura el registro en 0x00 para apagarla
}

```

**Código 1.** Encender y apagar pantalla.

- **Res\_Touch:** Contiene diferentes funciones que permiten realizar la lectura del estado de la membrana táctil. El siguiente diagrama muestra la manera como se realiza la lectura del estado de alguna coordenada del panel táctil.



**Figura 52.** Diagrama de flujo de lectura del panel táctil [28] .

## 5.8.2. PROGRAMACIÓN DE DIBUJO DE BOTONES, TEXTO Y FIGURAS GEOMETRICAS EN LA PANTALLA.

Debido a que los algoritmos más utilizados a lo largo de la programación de la GUI fueron el dibujo de figuras geométricas, botones y texto, se procederá a describir la manera como ellos se programan en la pantalla.

Antes que todo se debe inicializar la pantalla, para ello se dan los parámetros de los diferentes registros encargados de configurar el RA8875 de acuerdo al tipo de pantalla que se esté utilizando, para este caso es una pantalla de resolución 800x480 pixeles con 16k de profundidad de color, esta configuración se da en la función *LCD\_Initializtion()* de la librería RA8875\_Driver.

Una vez inicializada la pantalla se procede a escribir las diferentes funciones de acuerdo a lo que se desee dibujar en la pantalla. Para dibujar un cuadrado el usuario debe proceder a encender la pantalla con la función *Display\_on()*, luego se activa la zona en la que se desee realizar el dibujo y se procede a escribir la función de la figura geométrica a dibujar. El siguiente es el código para dibujar un cuadrado de tamaño 200\*480 en las coordenadas 0,0 con relleno de color negro.

```
Display_OFF(); // apaga la pantalla

    delay_ms(500);
    LCD_Initializtion(); // inicializa la pantalla
    delay_ms(500);
    Display_ON(); // enciende la pantalla

    Active_Window(0, 799, 0, 479); // habilita toda la pantalla
    LcdClear(White); // realiza un barrido de color blanco en toda la
pantalla

    delay_ms(50);
    Chk_Busy(); // espera a que el RA8875 procese la información
    Graphic_Mode(); // entra en modo dibujo
    LCD_DrawSquare(0, 0, 200, 479, Black, 1); // dibuja el cuadrado
    Chk_Busy(); // espera a que el RA8875 procese la información
```

**Código 2.** Dibujo de un cuadrado.

El resultado de este código se muestra en la siguiente figura.



**Figura 53.** Dibujo de cuadrado en pantalla.

Si se desea dibujar un icono especial se debe proceder a guardar los datos RGB de cada pixel de la imagen en un vector y luego realizar la escritura de cada posición de este vector en la pantalla como se muestra en el siguiente código.

```
LCD_Initializtion(); // inicializa la pantalla
delay_ms(500);
Display_ON(); // enciende la pantalla

Active_Window(0, 799, 0, 479); // habilita toda la pantalla
LcdClear(White); // realiza un barrido de color blanco en toda la
pantalla
delay_ms(50);
Chk_Busy(); // espera a que el RA8875 procese la información

Active_Window(270, 339, 350, 425); // habilita el espacio donde se
dibuja el icono
XY_Coordinate(270, 350); // se dan las coordenadas de dibujo del icono
Graphic_Mode();
LCD_CmdWrite(0x02); // comando de escritura de RGB en pixel
for (i = 0; i < 10640; i++) {
    LCD_DataWrite(go[i]); // escribe dato RGB guardado en el vector
de imagen
    Chk_Busy(); // espera a que el RA8875 procese la información
}
```

**Código 3.** Dibujo de icono en la pantalla.

El resultado del código anterior se muestra en la siguiente figura:



**Figura 54.** Dibujo de icono en pantalla.

Para escribir un texto se utiliza la función `LCD_Internal_Font_ROM()`, donde se describe el tamaño y color de texto además de las coordenadas donde se quiere ubicar el texto como se muestra en el siguiente código

```
delay_ms(500);  
LCD_Initializtion(); // inicializa la pantalla  
delay_ms(500);  
Display_ON(); // enciende la pantalla  
  
Active_Window(0, 799, 0, 479); // habilita toda la pantalla  
LcdClear(White); // realiza un barrido de color blanco en toda la pantalla  
delay_ms(50);  
Chk_Busy(); // espera a que el RA8875 procese la información  
  
LCD_Internal_Font_ROM(10, 10, Red, Black, 1, 1, 3, 0);  
Show_String(" HOLA MUNDO ", 1);  
delay_ms(1);
```

**Código 4.** Dibujo de texto en la pantalla.

La siguiente imagen muestra el resultado del código anterior.

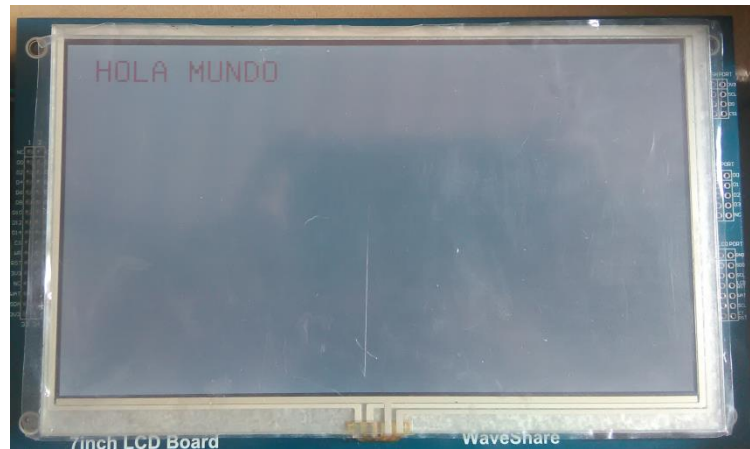


Figura 55. Dibujo de texto en pantalla.

Realizar un botón consiste en dibujar la figura que se quiere utilizar como botón, sea un cuadrado, un círculo o la imagen de un icono y siguiente a esto leer la membrana táctil, si las coordenadas donde es pulsada la membrana táctil coincide con la coordenada donde se encuentra dibujado el botón, el botón es activo, de lo contrario el botón no está siendo pulsado, el siguiente código muestra la manera como se implementó esta lógica en el RA8875 a través del NIOS II.

```
int n=0;
Active_Window(0,799,0,479);
TP_auto_mode(); //
LCD_CmdWrite(0x70);//
LCD_DataWrite(0xc2);
Enable_TP();
while(n==0)
{
    detect_touch();

    //Trace_coordinate();

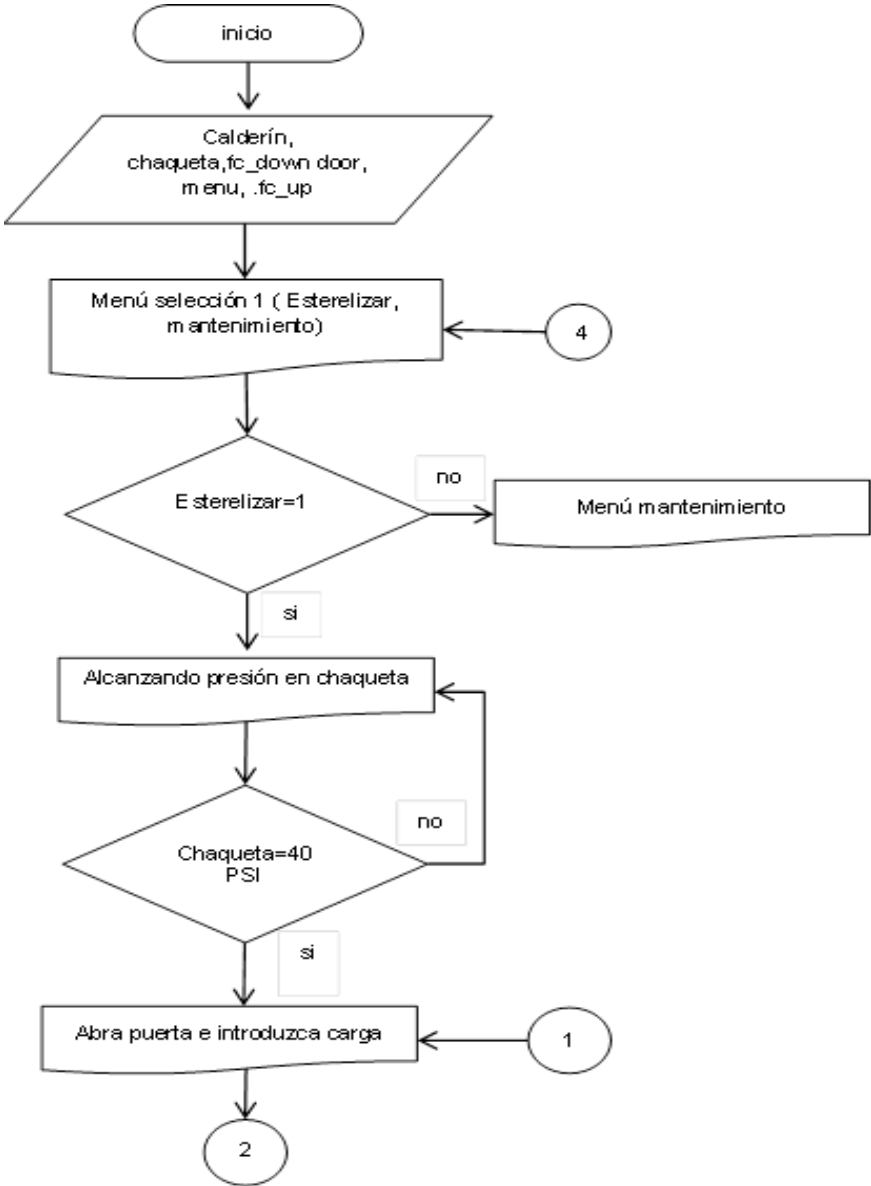
    if(LCD_X>270&&LCD_X<333)
    {
        if(LCD_Y>350&&LCD_Y<413)
        {
            n=1;
        }
    }
}
```

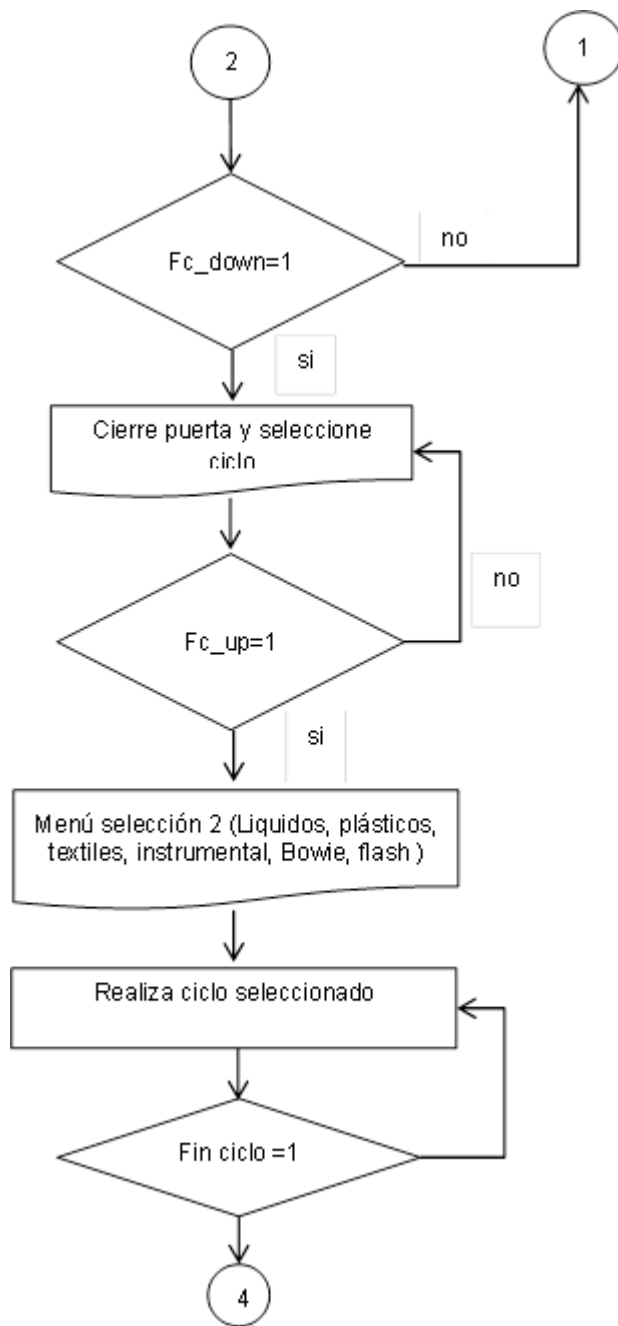
```
    }
}
if(LCD_X>404&&LCD_X<467)
    {
        if(LCD_Y>350&&LCD_Y<413)
        {
            n=2;
        }
    }
    Active_Window(538,601,350,413);
    if(LCD_X>538 &&LCD_X<601)
    {
        if(LCD_Y>350&&LCD_Y<413)
        {
            n=3;
        }
    }
}
return n;
```

**Código 5.** Programación de la membrana táctil para un botón.

### 5.8.3. RESULTADOS DE LA PROGRAMACIÓN DE LA GUI.

Debido a que por cuestiones de confidencialidad de la empresa no se puede mostrar el código utilizado para realizar las diferentes ventanas de la interfaz gráfica, se muestra la lógica empleada para realizar la interacción con el usuario a través del siguiente diagrama de flujo.





**Figura 56.**Diagrama de flujo de la GUI.

Algunos resultados obtenidos fueron: la ventana de inicio que se muestra en la figura 56 , la ventana alcanzando presión de la figura 57, la ventana de introducir carga de la figura 58, la ventana de selección de ciclo de la figura 59 y la ventana de configuración de ciclo de la figura 60.

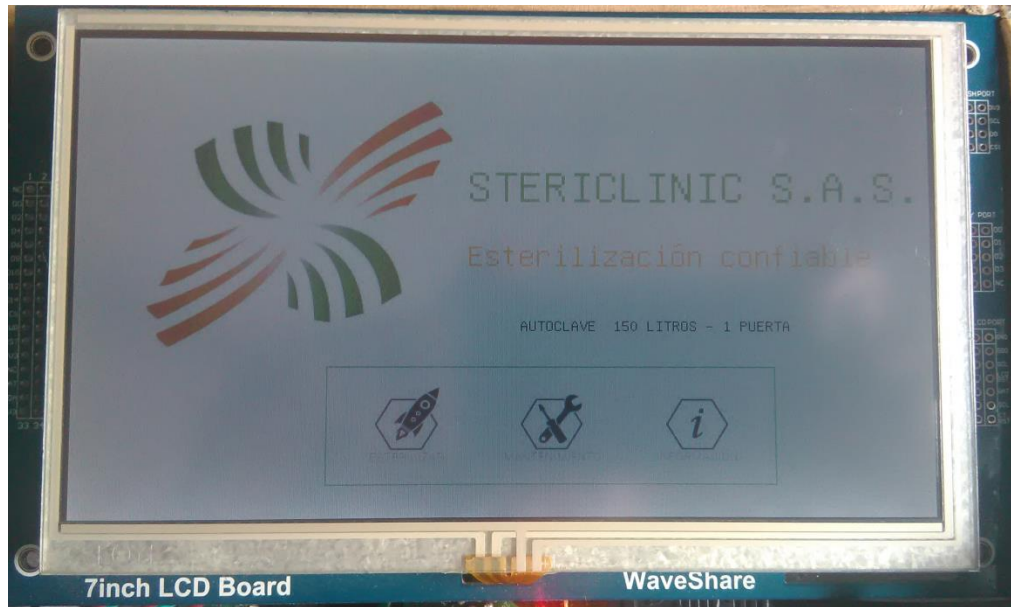


Figura 57. Ventana de inicio

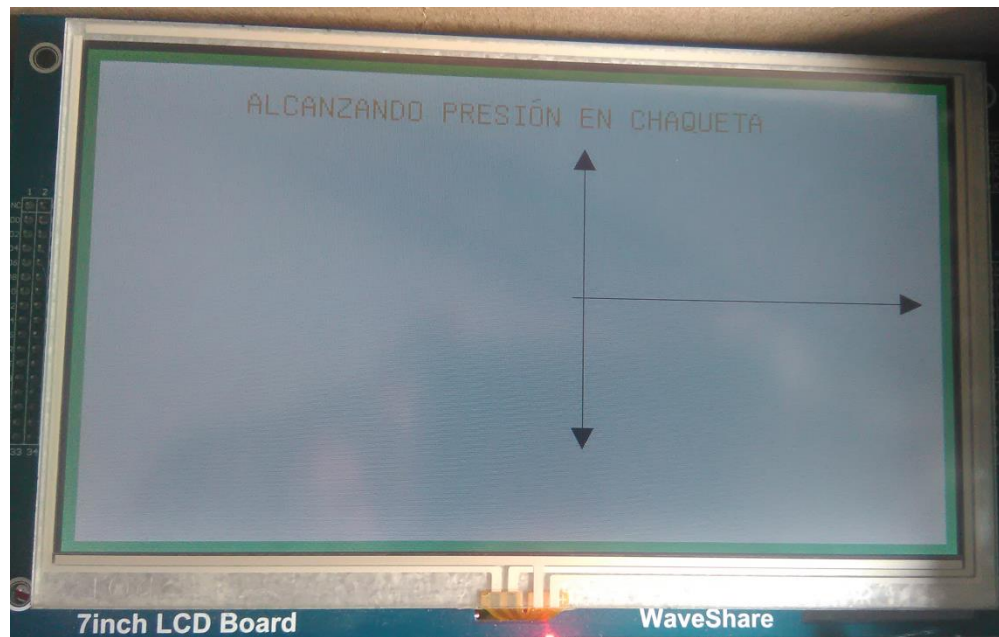
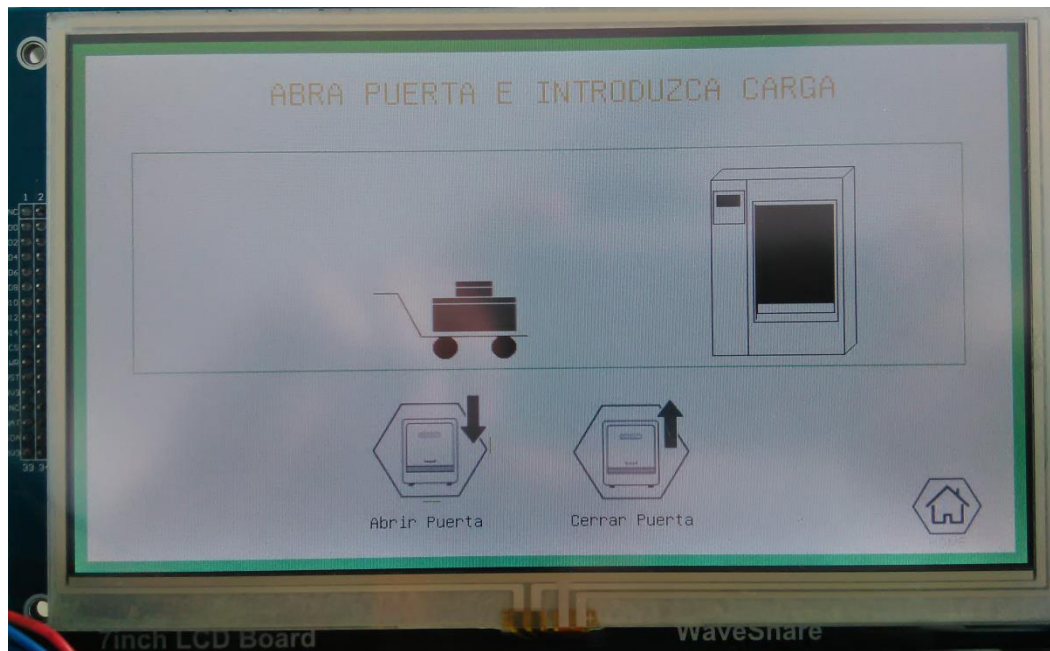
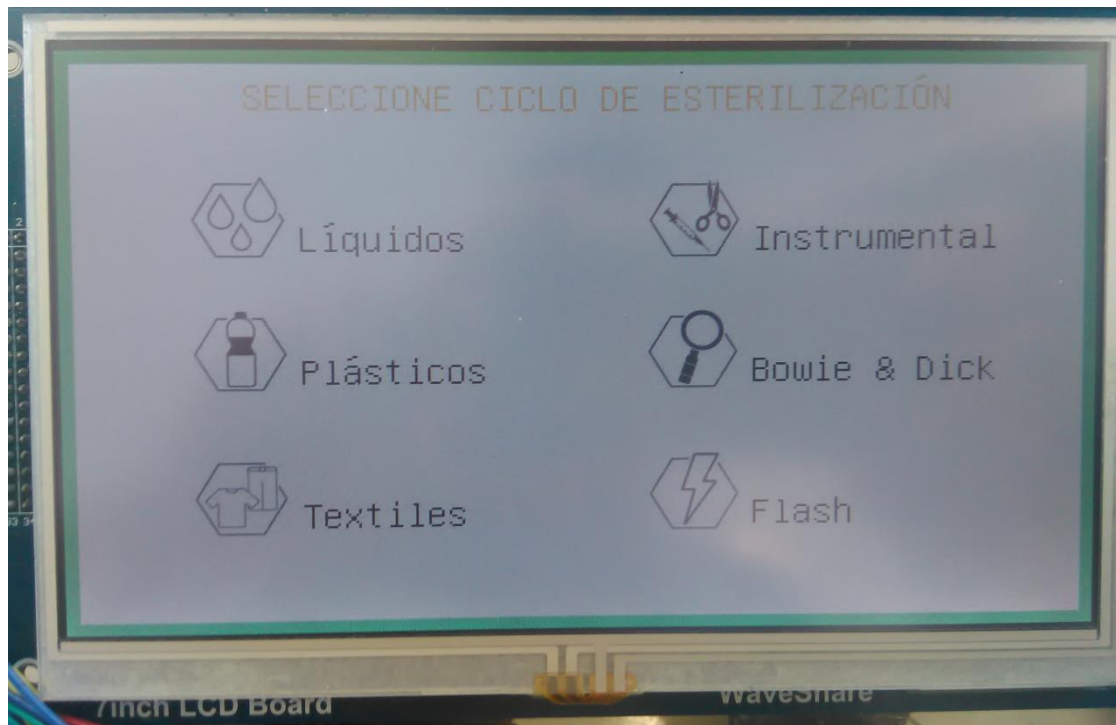


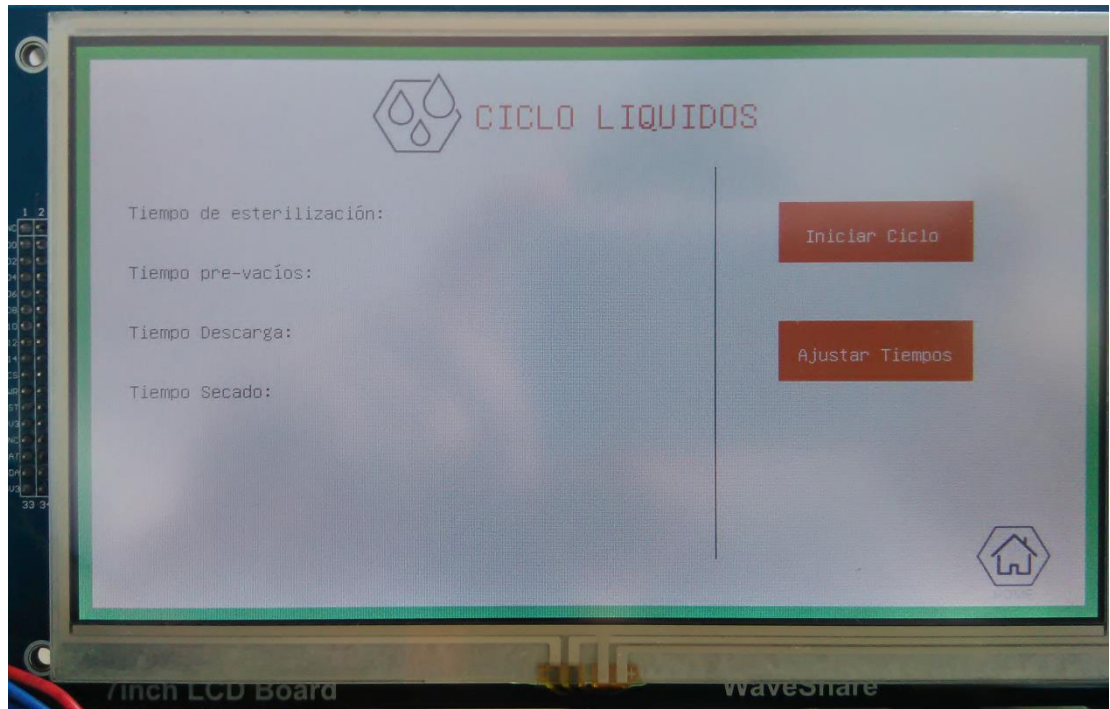
Figura 58. Ventana alcanzando presión en chaqueta.



**Figura 59.** Ventana introducir carga.



**Figura 60.** Ventana de selección de ciclo.



**Figura 61.** Ventana configuración de ciclo.

## 5.9. COMUNICACIÓN DE LA INTERFAZ GRÁFICA CON SISTEMA DE CONTROL

Se realizó la creación de la librería control.h, donde se implementaron las funciones de configuración de los parámetros de control que se muestran en la tabla 4.

ADDRESS	W/R	DATA
000000	0	Lee estado de EV1
000001	1	Configura estado de EV1
000010	0	Lee estado de EV2
000011	1	Configura estado de EV2
000100	0	Lee estado de EV3
000101	1	Configura estado de EV3
000110	0	Lee estado de Ev6
000111	1	Configura estado de Ev6
001000	0	Lee estado de Ev8
001001	1	Configura estado de Ev8
001010	0	Lee estado de Ev9
001011	1	Configura estado de Ev9
001100	0	Lee estado bomba vacio
001101	1	Configura estado bomba vacio
001110	0	Lee estado puerta arriba
001111	1	sube puerta
010000	0	Lee estado puerta abajo
010001	1	Baja puerta
010010	0	Lee sensor de presion cámara
010100	0	Lee sensor de temperatura
010110	0	Lee tiempo prevacío
010111	1	Establece tiempo prevacío
011000	0	Lee tiempo de esterilización
011001	1	Establece tiempo de esterilización
011010	0	Lee tiempo de descarga
011011	1	Establece tiempo de descara
011100	0	Lee tiempo de secado
011101	1	Establece tiempo de secado
011110	0	Lee dato ciclo
011111	1	inicia ciclo
100000	0	Lee dato calderín
100010	0	Lee dato presion chaqueta
100100	0	Lee presostato inicializar prevacios

100110	0	Lee alarma final
101000	0	Lee estado de EV5
101001	1	Configura estado de ev5
101010	0	Lee alarma pre vacio

**Tabla 4.** Funciones de la libreria control.h

En la descripción de hardware se realiza la de-multiplexación de los datos y de acuerdo a la instrucción requerida se ejecuta la descripción de hardware indicada. Esto se realizó con el fin de optimizar recursos de la FPGA.

## GLOSARIO

ALTERA: Marca productora de FPGAs ubicada en California, Estados Unidos.

CAD: Acrónimo para *computer-aided design*. Un sistema CAD es una combinación de hardware y software que permite a los ingenieros diseñar desde muebles hasta aviones.

DRIVER: Controlador, rutina o programa que enlaza un dispositivo periférico al sistema operativo.

FPGA: Del inglés *Field Program Gates Arrays*, arreglo de compuertas lógicas programables.

GUI: es un programa informático que actúa de interfaz de usuario, utilizando un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz.

HDL: Un lenguaje de descripción de hardware (HDL, *Hardware Description Language*) es un lenguaje de programación especializado que se utiliza para definir la estructura, diseño y operación de circuitos electrónicos, y más comúnmente, de circuitos electrónicos digitales.

IP-CORE: Modulo de descripción de hardware con propiedad intelectual.

JTAG: un acrónimo para *Joint Test Action Group*, es el nombre común utilizado para la norma IEEE 1149.1 titulada *Standard Test Access Port and Boundary-Scan Architecture para test access ports* utilizada para testear PCBs utilizando escaneo de límites.

LE: Elemento lógico dentro de un sistema digital.

NIOS II: Procesador embebido de 32 bits de arquitectura RISC.

PCB: Tarjeta de circuito impreso.

QSYS: Programa que permite realizar la descripción de hardware de una FPGA basándose en la interconexión de bloques con lógica definida.

QUARTUS II: Programa que permite realizar la descripción de hardware de las FPGAs de Altera.

RAM: Memoria de acceso aleatorio de características volátiles, conserva la información mientras se encuentre con energía.

ROM: Circuito integrado de memoria de solo lectura que almacena instrucciones y datos de forma permanente.

RTDpt100: Del inglés *resistance temperature detector*, es un sensor de temperatura resistivo, consiste en un alambre de platino a 0 °C tiene 100 ohms y que al aumentar la temperatura aumenta su resistencia eléctrica.

UART: Del inglés de *Universal Asynchronous Receiver-Transmitter*, en español: Transmisor-Receptor Asíncrono Universal, es el dispositivo que controla los puertos y dispositivos serie.

VERILOG: Verilog es un lenguaje de descripción de hardware (HDL, *del Inglés Hardware Description Language*) usado para modelar sistemas electrónicos.

## **APORTE AL CONOCIMIENTO**

En la empresa STERICLINIC S.A.S, se han realizado autoclaves basados en PLCs convencionales como lo son el S7-1200 o el Logo de Siemens, apoyándose en diferentes módulos que ofrece Siemens para añadirle características al sistema de control como lo es la visualización de parámetros mediante la pantalla KP-300 o la lectura de datos analógicos mediante módulos de expansión analógicos, esto ha traído el inconveniente de que se dependa de lo que ofrezca Siemens actualmente para añadirle características a las autoclaves lo que le quita autonomía de diseño a la empresa.

Debido a esto STERICLINIC S.A.S vio la necesidad de implementar un sistema electrónico de control propio, y se plantea hacer uso de sistemas embebidos basados en FPGAs con circuitos de acondicionamiento que le permitan tener autonomía en sus diseños e implementar elementos como pantallas táctiles sin necesidad de comprar módulos costosos y limitados como la pantalla KP-300 utilizada actualmente.

Esta tecnología no había sido utilizada en la empresa por lo que el proyecto presentado presenta un gran aporte al conocimiento, ya que el uso de un sistema propio puede ofrecer a la empresa un mejor posicionamiento en el mercado al poder implementar componentes como una impresora térmica, módulos de monitoreo remoto, módulos de exportación de datos, entre otros, sin depender del portafolio de productos de una empresa externa como Siemens.

También realizó aporte al conocimiento acerca de la manera como se diseña una interfaz gráfica de usuario más amigable, utilizando nuevas tecnologías como lo son las pantallas táctiles mediante librerías propias de la empresa, las cuales pueden ser mejoradas y utilizadas en otros proyectos.

## CONCLUSIONES Y RECOMENDACIONES

Con el trabajo realizado durante la práctica en STERICLINIC S.A.S, se logró cumplir con los objetivos planteados en el plan de trabajo propuesto, llegando a realizar una interfaz gráfica de usuario para una autoclave de 150 litros basado en sistemas embebidos como los son las FPGAs con programación propia así como también el sistema de control y la PCB lo que le permite a la empresa ganar autonomía y rentabilidad frente al sistema empleado actualmente además de mejorar la apariencia de la autoclave con la implementación de una pantalla táctil a color.

La práctica plantea el inicio de una actualización tecnológica de la empresa y demuestra la viabilidad de usar la tecnología empleada en el proyecto para los demás productos que se fabrican, además de dejar abierta la posibilidad de realizar nuevos tipos de autoclaves, con sistemas de control más complejos sin que esto signifique un incremento significativo en el producto.

Como trabajos futuros se plantea, realizar las pruebas finales en una autoclave real con el fin de analizar el comportamiento del nuevo sistema de control y realizar las respectivas correcciones y mejoras.

## BIBLIOGRAFIA

- [1] J. P. Gomez Vargas, *PRACTICA EMPRESARIAL STERICLINIC S.A.S.* Bucaramanga: Universidad Pontificia Bolivariana, 2014.
- [2] G. Bioeditores, *Esterilización, desinfección y limpieza hospitalaria. TOMO I. Métodos, procedimientos, materiales y prácticas de desinfección y limpieza.* Bioediciones.
- [3] C. Silvestre, L. Fagoaga, M. J. Garciandía, I. Lanzeta, M. C. Mateo, and M. C. Zapata, "Esterilización," in *Canales del Sistema Sanitario de Navarra*, 2000, vol. 23, pp. 95–103.
- [4] J. Dejean and C. G. Herrero, *La esencia del estilo.* Editorial Nerea, S.A., 2009.
- [5] D. A. Murillo, *Gestión de enfermería y los servicios generales de organizaciones sanitarias.* Editorial Díaz de Santos, S.A., 2012.
- [6] J. CORTES MUNGUÍA and U. SUÁREZ JUÁREZ, "DISEÑO DEL SISTEMA DE CONTROL DE PROCESOS PARA UNA AUTOCLAVE HÍBRIDA," 2015.
- [7] Francisco and F. J. [et al. ] C. Márquez, "TCAE en el servicio de esterilización." p. 192, 2007.
- [8] F. A. Oyawale and A. E. Olaoye, "Design and Construction of an Autoclave," *Pac J Sci Technol*, vol. 8, pp. 224–230, 2007.
- [9] J. J. Perkins, *Principles and Methods of Sterilization in Health Sciences.* Thomas, 1969.
- [10] W. B. Underwood, "Bedpan cleaner and sterilizer." Google Patents, 1927.
- [11] W. B. Underwood, "Closure fastener for sterilizers." Google Patents, 1924.
- [12] Tuttnauer, "Tuttnauer Autoclave Manufacturer," 2015. [Online]. Available: <http://es.tuttnauer.com/>. [Accessed: 20-Dec-2015].
- [13] Matachana, "Esterilizadores," 2008. [Online]. Available: <http://www.matachanagroup.com/productos/esterilizadores.html>. [Accessed: 20-Dec-2015].
- [14] E. U. S. Riveros, "Esterilizadores a vapor I."
- [15] L. Romero, *Apuntes de Cátedra, Esterilización.* España: Universidad Nacional de San Juan, 2008.
- [16] M. C. O. Vargas and M. G. S. Vazquez, *Manual de evaluación de la calidad del servicio en enfermería / Manual Evaluation of Service Quality in Nursing: Estrategias para su aplicación / Strategies for its Application.* Editorial Medica Panamericana Sa de, 2009.
- [17] A. E. de Normalización y Certificación, *UNE-EN 285/AC: esterilización: esterilizadores de vapor: esterilizadores grandes.* AENOR, 1998.
- [18] A. R. Gennaro, *Remington Farmacia*, no. v. 1. Editorial Médica Panamericana, 2003.
- [19] J. K. Fuller, J. R. Fuller, and E. Ness, *Instrumentación quirúrgica: teoría, técnicas y procedimientos.* Panamericana, 2007.
- [20] C. Camargo, J. Cortés, and A. Jiménez, "Implementación de sistemas digitales complejos utilizando sistemas embebidos," *Ingenium*, vol. 13, no. 25, pp. 5–15, 2012.
- [21] R. Cayssials, *Sistemas embebidos FPGA.* Marcombo, S.A., 2014.
- [22] S. Hauck, "The roles of FPGAs in reprogrammable systems," *Proc. IEEE*, vol. 86, no. 4, pp. 615–638, 1998.
- [23] I. Kuon and J. Rose, "Measuring the gap between FPGAs and ASICs," *Comput.*

- Des. Integr. Circuits Syst. IEEE Trans.*, vol. 26, no. 2, pp. 203–215, 2007.
- [24] ALTERA Corp, *MAX 10 HANDBOOK*. California, EEUU, 2015.
- [25] Altera.com, “MAX 10 - Overview.” [Online]. Available: <https://www.altera.com/products/fpga/max-series/max-10/overview.html>. [Accessed: 07-Jan-2016].
- [26] Altera.com, “Nios II Processor - Overview.” [Online]. Available: <https://www.altera.com/products/processors/overview.html>. [Accessed: 07-Jan-2016].
- [27] P. P. Chu, *Embedded SoPC Design with Nios II Processor and Verilog Examples*. Wiley, 2012.
- [28] R. Semiconductor, “RA8875 TFT DRIVER,” 2014.
- [29] R. F. Coughlin and F. F. Driscoll, *Amplificadores operacionales y circuitos integrados lineales*. Pearson Educación, 1999.
- [30] National Semiconductor, “Low Power Dual Operational Amplifiers,” 2005.
- [31] D. W. Hart, *Electrónica de potencia*. Pearson Educación, SA, 2001.
- [32] T. J. Maloney, *Electrónica industrial moderna*. Pearson Educación, 2006.
- [33] Philips Semiconductors, “Triacs,” 1997.
- [34] P. Eisler and M. E. W. Williams, *My Life with the Printed Circuit*. Lehigh University Press, 1989.
- [35] S. F. Danko and A. Moe, “Process of assembling electrical circuits.” Google Patents, 1956.
- [36] W. C. Bosshart, *Printed Circuit Boards: Design and Technology*. Tata McGraw-Hill, 1983.
- [37] D. Jansen, *The Electronic Design Automation Handbook*. Springer, 2003.
- [38] Altium.com, “PCB Design Tools - EDA Software - ECAD Data Management - Altium,” 2016. [Online]. Available: <http://www.altium.com/>. [Accessed: 07-Jan-2016].
- [39] A. Electronics, “Bemicro Max 10,” 2015. [Online]. Available: <https://www.arrow.com/en/products/bemicromax10/arrow-development-tools/#page-1>. [Accessed: 20-Sep-2015].
- [40] Support.industry.siemens.com, “SIOS,” 2016. [Online]. Available: <https://support.industry.siemens.com/cs/document/49864804/liberaci%C3%B3n-para-el-suministro%3A-simatic-hmi-kp300-basic-mono-pn?dti=0&lc=es-WW>. [Accessed: 08-Jan-2016].
- [41] Waveshare.com, “7inch Resistive Touch LCD (C) 7 inch 800\*480 Multicolor Graphic LCD, with resistive touch screen,” 2016. [Online]. Available: <http://www.waveshare.com/7inch-Resistive-Touch-LCD-C.htm>. [Accessed: 09-Jan-2016].