

**UNIVERSIDAD POLITÉCNICA SALESIANA**  
**SEDE CUENCA**

**CARRERA DE MECATRÓNICA**

*Trabajo de titulación previo a la  
obtención del título de  
Ingeniero en Mecatrónica*

**PROYECTO TÉCNICO:**

**“DISEÑO DE UN SISTEMA DUAL PARA LA CORTADORA LASER DGL-6060  
CON IMPRESIÓN 3D”**

**AUTORES:**

JUAN JOSÉ HURTADO ORDÓÑEZ  
JONATHAN AUGUSTO VIDAL DÍAZ

**TUTOR:**

ING. CHRISTIAN MAURICIO COBOS MALDONADO, MSc.

**CO-TUTORA:**

ING. MÓNICA ALEXANDRA ROMERO SACOTO, MSc.

CUENCA - ECUADOR

2021

## CESIÓN DE DERECHOS DE AUTOR

Nosotros, Juan José Hurtado Ordóñez con documento de identificación N° 0105502330 y Jonathan Augusto Vidal Díaz con documento de identificación N° 1104874035, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación: **“DISEÑO DE UN SISTEMA DUAL PARA LA CORTADORA LASER DGL-6060 CON IMPRESIÓN 3D”**, mismo que ha sido desarrollado para optar por el título de *Ingeniero en Mecatrónica*, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

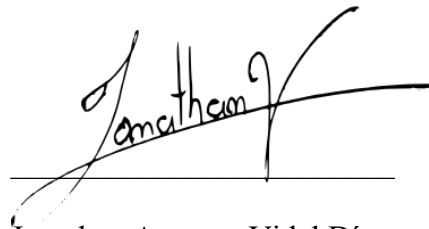
En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad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

Cuenca, noviembre de 2021.



Juan José Hurtado Ordóñez

C.I. 0105502330



Jonathan Augusto Vidal Díaz

C.I. 1104874035

## CERTIFICACIÓN

Yo, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: **“DISEÑO DE UN SISTEMA DUAL PARA LA CORTADORA LASER DGL-6060 CON IMPRESIÓN 3D”**, realizado por Juan José Hurtado Ordóñez y Jonathan Augusto Vidal Díaz, obteniendo el *Proyecto Técnico*, que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana.

Cuenca, noviembre de 2021.

A handwritten signature in black ink, reading "Christian Cobos", is written over a horizontal line. The signature is stylized and somewhat cursive.

Ing. Christian Mauricio Cobos Maldonado, MSc.

C.I. 0104015938

## DECLARATORIA DE RESPONSABILIDAD

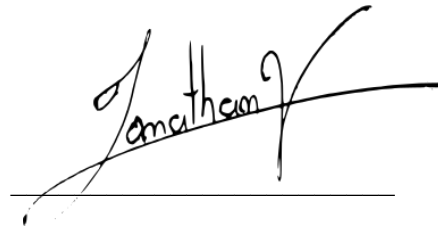
Nosotros, Juan José Hurtado Ordóñez con documento de identificación N° 0105502330 y Jonathan Augusto Vidal Díaz con documento de identificación N° 0105502330, autores del trabajo de titulación: “**DISEÑO DE UN SISTEMA DUAL PARA LA CORTADORA LASER DGL-6060 CON IMPRESIÓN 3D**”, certificamos que el total contenido del *Proyecto Técnico*, es de nuestra exclusiva responsabilidad y autoría.

Cuenca, noviembre de 2021.



Juan José Hurtado Ordóñez

C.I. 0105502330



Jonathan Augusto Vidal Díaz

C.I. 0105502330



## Dedicatoria

Este trabajo va dedicado, primeramente, a mi Dios quien ha sido el que me ha acompañado en cada momento de esta etapa y quien a bendecido mi vida y la de mi familia para llegar a este punto. También lo dedico a mis abuelitos y mis padres Johnny Hurtado y Ximena Ordoñez quienes han sido motor para mi vida y un ejemplo de esfuerzo, superación y dedicación. A mis hermanos Pedro Hurtado y Mateo Hurtado quien con cada momento de alegría me han brindado ánimos para nunca rendirme , a todos mis familiares quien de una u otra manera también forman parte de mi esfuerzo y ganas de ser mejor cada día.

También se lo dedico a a todas las personas que me han acompañado a lo largo de mi vida, de quien aprendí lo que es valentía, responsabilidad y el significado de perseverancia, pues sin estas cualidades tampoco podría haber llegado hasta este punto, ellas me han brindando, con cada momento de alegría y de tristeza, me motivo a seguir adelante y luchar por conseguir mis metas. Sería muy difícil ser la persona que soy ahora sin lo que cada uno de ellos ha aportado a mi vida y sin duda también a los grandes amigos que forman parte de mi vida y que todo este documento no alcanzaria para agradecer.

Juan Hurtado

El presente trabajo de titulación es la culminación de una utopía que empezó hace 5 años; va dedicado a Dios, quien me brindo esta oportunidad, a mis padres Jaime Vidal y Melva Díaz, por todo el esfuerzo que realizaron para que yo pudiera cumplir este objetivo con la dificultades de estar en otra ciudad. A mis hermanos Katuska y Josué, que me han brindado su apoyo incondicional; y demás familiares que me han aportado para que pueda alcanzar esta meta. Además, de a todos mis amigos que en esta ciudad me han hecho sentir como en casa.

Jonathan Vidal

## **Agradecimientos**

Primero agradezco a Dios quien me ha llenado de bendiciones a mí y a mis seres queridos ,a mis padres que con la ayuda de Dios siempre pudieron velar por mí y apoyarme en estudiar la carrera que yo más deseaba, gracias por siempre impulsarme a ser mejor. También quiero agradecer a la Empresa Dumenlaser SAS por todo el apoyo brindado para la realización de este trabajo de titulación y especialmente a Ing. Fernando Méndez quien dentro de la empresa nos apoyó con todo lo necesario. Agradezco a mi compañero Jonathan Vidal con quien conformamos un gran grupo de trabajo, pero sobre todo una gran amistad. Además, de a mi tutor el Ing. Cristian Cobos y a mi co-tutora la Ing. Mónica Romero que de igual manera supieron guiarnos y apoyarnos en el proceso de realización del trabajo de titulación y brindarnos el apoyo humano de su amistad.

### **Juan Hurtado**

A Dios por ser la luz que guía mi camino , a la Universidad, a nuestro apreciado director del trabajo de titulación, Ing. Cristian Cobos y a la Ing. Mónica Romero co-tutora, los cuales nos han brindado su apoyo y amistad, a todos mis profesores que me ayudaron en mi proceso de formación. Y a mis amigos y compañeros con los que he compartido gratas experiencias

A la empresa DumenLaser SAS, en la persona del Ing. Fernando Méndez, por la confianza brindada para poder realizar nuestro proyecto y apoyarnos en todo. Y a mi gran amigo Juan José Hurtado Ordoñez, con el cual hemos superado diferentes vicisitudes a lo largo de la carrera y de este trabajo de titulación, y que me brindo su sincera amistad.

Muchas Gracias

### **Jonathan Vidal**



Este documento fue realizado enteramente en L<sup>A</sup>T<sub>E</sub>X

# Índice

<b>Cesión de Derechos de Autor</b>	<b>I</b>
<b>Certificación</b>	<b>II</b>
<b>Declaratoria de Responsabilidad</b>	<b>III</b>
<b>Dedicatoria</b>	<b>IV</b>
<b>Agradecimientos</b>	<b>V</b>
<b>Resumen</b>	<b>XIV</b>
<b>Abstract</b>	<b>XV</b>
<b>1. Introducción</b>	<b>1</b>
<b>2. Problema</b>	<b>1</b>
2.1. Antecedentes . . . . .	2
2.2. Importancia y alcances . . . . .	2
2.3. Delimitación . . . . .	2
2.4. Problema General. . . . .	5
2.5. Problemas Específicos. . . . .	5
<b>3. Objetivos</b>	<b>5</b>
3.1. Objetivo General. . . . .	5
3.2. Objetivos Específicos. . . . .	5
3.3. Hipótesis General . . . . .	6
3.4. Hipótesis Específicas . . . . .	6
<b>4. Matriz de consistencia</b>	<b>6</b>
<b>5. Marco Teórico</b>	<b>8</b>
5.1. Maquinaria CNC . . . . .	8
5.1.1. Cortadora Laser . . . . .	9
5.1.2. Impresora 3D . . . . .	10
5.2. Arquitectura básica de maquinaria de impresión 3D de FFF . . . . .	11

5.2.1.	Hotend . . . . .	12
5.2.2.	Extrusor . . . . .	13
5.2.3.	Cama Caliente . . . . .	13
5.2.4.	Sistema De Movimiento de Ejes . . . . .	14
5.2.5.	Sistema De Control . . . . .	14
5.3.	Proceso de impresión 3D . . . . .	14
5.4.	Sistema de control . . . . .	16
5.4.1.	Sistemas de control discreto . . . . .	16
5.4.2.	Función de transferencia Z . . . . .	16
5.4.3.	Controlador PID . . . . .	17
5.5.	Internet de las cosas (IoT) . . . . .	18
<b>6.</b>	<b>Marco metodológico</b>	<b>18</b>
6.1.	Identificación de las características mecánicas y electrónicas de la cortadora laser DGL-6060 para implementar un extrusor para impresión 3D . . . . .	18
6.1.1.	Componentes Mecánicos de la maquina DGL-6060 . . . . .	18
6.1.2.	Componentes Electrónicos . . . . .	19
6.2.	Adaptación del diseño mecánico y electrónico de la cortadora laser DGL-6060, con un sistema dual con extrusor y cama caliente para la impresión 3D . . . . .	22
6.2.1.	Selección de sistema de cama caliente, sistema de extrusión. . . . .	22
6.2.2.	Diseño de acople para el hotend y extrusor en la cortadora laser . . . . .	24
6.2.3.	Evaluación de compatibilidad de las tarjetas electrónicas . . . . .	29
6.3.	Diseñar el sistema de control para la impresión 3D que permita leer los archivos generados, comandando los motores . . . . .	29
6.3.1.	Sistema de control para el movimiento en los 3 ejes. . . . .	29
6.3.2.	Adaptación de G code de impresión 3D para su funcionamiento en Linux CNC . . . . .	37
6.4.	Diseño de control de temperaturas que intervienen en la Impresión 3D . . . . .	43
6.4.1.	Aproximación del modelo del termistor . . . . .	44
6.4.2.	Obtención de los controladores de temperaturas . . . . .	47
6.5.	Diseñar un sistema dual y una interfaz IOT para inventario remoto; incorporando los software y controladores pertinentes del corte laser e impresión 3D. . . . .	64
6.5.1.	Diseño de la interfaz gráfica para la pantalla . . . . .	64
6.5.2.	Diseño de la plataforma iot . . . . .	67

<b>7. Resultados</b>	<b>70</b>
7.1. Resultado de la identificación de las características mecánicas y electrónicas de la cortadora laser DGL-6060 para implementar un extrusor para impresión 3D. . . . .	70
7.2. Resultado de la adaptación del diseño mecánico y electrónico de la cortadora laser DGL-6060; con un sistema dual que incorpora un extrusor y cama caliente para la impresión 3D. . . . .	70
7.3. Resultado del diseño del sistema de control para la impresión 3D, que permita leer los archivos generado, comandando los motores . . . . .	70
7.4. Resultado del diseño de control de temperaturas que intervienen en la Impresión 3D. . . . .	71
7.5. Resultado del diseño sistema dual y una interfaz IOT para inventario remoto; incorporando los software y controladores pertinentes del corte laser e impresión 3D. . . . .	72
<b>8. Conclusiones</b>	<b>75</b>
<b>9. Recomendaciones</b>	<b>76</b>
<b>Referencias</b>	<b>80</b>

## Lista de Tablas

1.	Cronograma de Actividades. . . . .	4
2.	Matriz de consistencia lógica. . . . .	7
3.	Tabla comparativa para selección de Hotend . . . . .	23
4.	Tabla comparativa para selección de Extrusor . . . . .	24
5.	Cálculo de parametros mediante Ziegler Nichols . . . . .	61



## Lista de Figuras

1.	Ubicación de la empresa Dumenlaser. . . . .	3
2.	Esquema de funcionamiento de una máquina de control numérico. . . . .	8
3.	Proceso del corte laser. . . . .	10
4.	Proceso de impresión 3D mediante FDM. . . . .	11
5.	Partes del hotend. . . . .	12
6.	Partes del Extrusor. . . . .	13
7.	Proceso de impresión 3D. . . . .	15
8.	Vista del diseño del acople realizado. . . . .	25
9.	Diseño modular. . . . .	26
10.	Diagrama de cuerpo libre del eje X . . . . .	27
11.	Diagrama Fuerza Cortante . . . . .	28
12.	Diagrama Momento Flector . . . . .	28
13.	Diagrama de bloques del sistema de control de LinuxCNC . . . . .	31
14.	Diagrama de bloques del control del motor a pasos . . . . .	32
15.	Configuración de parámetros básicos . . . . .	33
16.	Configuración de parámetros externos . . . . .	33
17.	Configuración de la interfaz gráfica . . . . .	34
18.	Configuración de las tarjetas Mesa . . . . .	35
19.	Configuración de las entradas y salidas en las tarjetas Mesa . . . . .	35
20.	Configuración de los motores, drivers y mecanismos de los ejes . . . . .	36
21.	Configuración de las dimensiones de los ejes . . . . .	37
22.	Curva de la ecuación característica de un termistor NTC 100K. . . . .	45
23.	Esquema del Divisor de tensión . . . . .	46
24.	Circuito propuesto de conexión USB tipo B. . . . .	48
25.	Circuito propuesto de conexión Atmega328p y reloj. . . . .	49
26.	Circuitos de entradas de sensores. . . . .	50
27.	Circuitos de control mediante PWM . . . . .	52
28.	Cruces por 0 en la señal AC . . . . .	53
29.	Circuito detector de cruce por 0 . . . . .	54
30.	Circuito de control Cama Caliente . . . . .	54
31.	Control de fase . . . . .	55
32.	Diagrama de bloques del control PID . . . . .	56
33.	Codigo del algoritmo de temperatura . . . . .	57

34.	Fenómeno Derivative Kick . . . . .	57
35.	Código algoritmo de control PID hotend . . . . .	58
36.	Código algoritmo de control PID cama caliente . . . . .	60
37.	Señal de salida y periodo crítico . . . . .	61
38.	Señal críticamente estable. . . . .	62
39.	Respuesta del sistema del hotend con constantes de $K_p=19,2$ $K_i=1.46$ $K_d=62.97$ . . . . .	62
40.	Respuesta del sistema del hotend con constantes de $k_p=6.5$ $k_i=2.3$ $k_d=62.57$ . . . . .	63
41.	Diagrama de flujos del proceso de la comunicación serial. . . . .	65
42.	Diagrama de flujos del proceso del programa M199. . . . .	69
43.	Modelo de placa controladora . . . . .	71
44.	Diagrama de flujo del bucle principal del algoritmo del microcontrolador . . . . .	73
45.	Diagrama de flujo del método PIDCama . . . . .	74
46.	Diagrama de procesos del método CalcularPID . . . . .	75
47.	Diagrama de procesos del método MedirTemperaturaCama y MedirTemperatura . . . . .	75

## Resumen

**E**n los últimos años, la fabricación aditiva ha cobrado gran relevancia para procesos de modelado y prototipado, la demanda de impresoras 3D se ha incrementado considerablemente. Por esta razón la empresa colombiana Dumen laser SAS, dedicada a la fabricación de máquinas de corte y grabado laser, en su afán de innovar y diversificar su línea de productos ha incursionado en el desarrollo de máquinas y modelos compatibles con la impresión 3D.

El presente trabajo de titulación sobre el Diseño de un Sistema dual para la cortadora laser DGL-6060 con impresión 3D; Dumen laser SAS ha colaborado con la información y recursos requeridos para el desarrollo de esta investigación.

Este proyecto presenta un enfoque mecatrónico ya que involucra la adaptación del diseño mecánico y electrónico que posee la cortadora laser antes mencionada para la función de impresión 3D. También del control tanto del movimiento de los ejes como de los parámetros de impresión, enfocándonos en el control de temperatura de extrusión, cama caliente. Además del diseño de una interfaz gráfica que nos permita la función dual de la máquina y que cuente con un sistema de IOT para inventario remoto.

**Palabras clave:** impresión 3D, corte laser, sistema dual, IOT, control temperatura de impresión.

## Abstract

In recent years, additive manufacturing has become very important for modeling and prototyping processes, the demand for 3D printers has increased considerably. For this reason, the Colombian company Dumen laser SAS, dedicated to the manufacture of laser cutting and engraving machines, in its eagerness to innovate and diversify its product line, has ventured into the development of machines and models compatible with 3D printing.

the present degree work on the Design of a Dual System for the DGL-6060 laser cutter with 3D printing; Dumen laser SAS has collaborated with the information and resources necessary for the development of this research.

This project presents a mechatronic approach since it involves the adaptation of the mechanical and electronic design that the aforementioned laser cutter has for the 3D printing function. also the control of both the movement of the axes and the printing parameters, focusing on the control of extrusion temperature, hot bed and closed environment. in addition to the design of a graphical interface that allows us the dual function of the machine and that has an IOT system for remote inventory.

**keywords:** 3D printing, laser cutting, dual system, IOT, printing temperature control.

# 1. Introducción

Desde el surgimiento de la tecnología de control numérico computarizado (CNC) los procesos de manufactura han tenido cambios considerables en cuanto a la presión y rapidez, mejorando la flexibilidad y productividad de los procesos de mecanizado; también el uso de CNC ha permitido realizar piezas más complejas y con distintos tipos de materiales. La tecnología CNC es muy versátil y se adapta a la mayoría de maquinas herramientas como fresadoras, tornos, taladros, corte laser e impresión 3D; básicamente se adaptan a los procesos que utilizan ejes cartesianos y polares.

Dumenlaser SAS es una empresa colombiana de la ciudad de Pasto dedicada a la innovación, desarrollo e implementación de tecnologías, esto le ha permitido sacar al mercado su propia gama de maquinaria CNC de corte y grabado laser, contando con diferentes modelos que se adaptan a las necesidades de sus clientes. Invirtiendo constantemente en I+D+I (Implementación + Desarrollo o+ Innovación) lo que ha generado el desarrollo también de nuevos productos como Routers e Impresoras 3D de tecnología SLS teniendo como mercado potencial Colombia, Ecuador, EEUU, España y otros países de América Latina.(Méndez, 2015).

El objetivo de Dumenlaser SAS es producir maquinaria con un alto valor tecnológico y de calidad, que les permita posicionarse en el mercado latinoamericano como una opción innovadora y así distinguirse de la competencia. Adicionalmente busca ser un referente de nuevos proyectos tecnológicos en el país y la fabricación de bienes de capital, con el mayor valor agregado nacional posible. (Méndez, 2015). (Banco de Ideas, 2012).

La empresa Dumenlaser, busca potenciar su maquinaria con un sistema dual de corte a laser e impresión 3D FFF(Fabricación de Filamento Fundido) ya que con los principios de fabricación que ellos emplean son muy similares; incorporando nuevas tecnologías como IOT para una mejor interacción entre el usuario con la máquina, además de tener un control de inventario, tiempos y procesos con el afán de mejorar el rendimiento de los equipos.

# 2. Problema

Las máquinas de manufactura aditiva tienen una gran demanda en el mercado, ya que esta tecnología brinda muchas ventajas en el área de diseño, especialmente en la parte de prototipado de piezas. En la actualidad la tendencia tecnológica obliga a las empresas desarrolladoras de maquinaria de prototipado, a desarrollar equipos versátiles, permitiendo al consumidor obtener máquinas que permitan dos tecnologías juntas (Impresión 3D/ Mecanizado

CNC). Por tal manera se desarrollará una propuesta para el acople de las tecnologías Corte laser e impresión 3D.

## 2.1. Antecedentes

Las cortadoras y grabadoras láser marca Dumen son fabricadas luego de años de un proceso de investigación, desarrollo y pruebas de campo, en las condiciones más exigentes de trabajo. Detrás de esto, hay un equipo multidisciplinario con más de 20 años de experiencia en maquinaria CNC que ha puesto todo su esfuerzo y conocimiento para lograr un equipo estable, confiable y que pueda cumplir las expectativas del mercado. La empresa Dumen laser cuenta con una cortadora laser DGL-6060 la cual sobresale por su gran rendimiento y sus excelentes características mecánicas enfocadas en movimientos precisos y en evitar las vibraciones; además han incursionado en el área de impresión 3D desarrollando proyectos de diseños de impresoras mediante SLS, que aún no están disponibles en el mercado.

## 2.2. Importancia y alcances

- El presente proyecto de titulación va dirigido a la empresa Dumenlaser SAS, quien lleva 8 años posicionándose como uno de los principales fabricantes de máquinas de corte laser, dentro de Colombia, y uno de los principales exportadores y distribuidores de esta maquinaria para nuestro país; busca potenciar su maquinaria con un sistema de impresión 3D ya que con los principios de fabricación que ellos emplean son muy similares. Lo que le permitirá mantenerse a la vanguardia de la maquinaria a nivel regional de Sudamerica, y continuar con su constante desarrollo e innovación que son un eje fundamental para la compañía y de esta manera desarrollar una maquina más competitiva en el mercado que satisfaga las necesidades de sus clientes, abriéndose campo también en la rama de la impresión 3D.

## 2.3. Delimitación

- a.- **Delimitación Espacial:** Este proyecto de titulación se realizó en la empresa Dumen laser SAS. Ubicada entre calle 20 y calle 21<sup>a</sup> de la ciudad de Pasto, del departamento de Nariño en la republica de Colombia, la dirección exacta se encuentra en la figura 1 .
- b.- **Delimitación Temporal:** El trabajo de titulación se elaboró en el año 2021, en un lapso de 6 meses acorde al cronograma mostrado en la tabla

**Figura 1:** Ubicación de la empresa Dumenlaser.



**Fuente:** Google Maps

**Tabla 1:** Cronograma de Actividades.

		CRONOGRAMA DE ACTIVIDADES																								
		MES 1				MES 2				MES 3				MES 4				MES 5				MES 6				
OBJETIVOS ESPECÍFICOS	ACTIVIDADES	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	
Identificar las características mecánicas y electrónicas de la cortadora laser DGL-6060 para implementar un extrusor para impresión 3D	Identificación de los componentes mecánicos presentes en la maquina DGL-6060	x	x																							
	Identificación de los componentes electrónicos presentes en la maquina DGL 6060			x	x																					
Adaptar el diseño mecánico y electrónico de la cortadora laser DGL-6060, para su potenciación, con un sistema dual con extrusor y cama caliente para la impresión 3D	Selección de sistema de cama caliente, sistema de extracción y ambiente de impresión							x																		
	Diseño de acople para el hotend y extrusor en la cortadora laser							x	x	x																
	Evaluación de compatibilidad de las tarjetas electrónicas											x	x													
Diseñar el sistema de control para la impresión 3D, que permita leer los archivos generado, comandando los motores y control de temperatura de la cama y el ambiente cerrado.	Revisión bibliográfica de controladores para máquinas de impresión 3D											x	x													
	Análisis de softwares que presenta la maquina DGL-6060 para interpretación de los códigos G											x	x	x												
	Implementación del interprete para código G para impresión 3D															x	x	x								
	Diseño del sistema de control para el movimiento en los 3 ejes																			x	x	x				
Diseñar un sistema dual y una interfaz IOT para inventario remoto; incorporando los software y controladores pertinentes del corte laser e impresión 3D.	Revisión bibliográfica de sistemas Iot																							x	x	
	Diseño de la interfaz grafica para la pantalla																							x	x	
	Diseño de la plataforma iot																							x	x	
	Redacción del documento	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	

Fuente: Autores



## **2.4. Problema General.**

- ¿Es posible diseñar una cortadora laser DGL-6060 implementando un sistema dual con impresión 3D?

## **2.5. Problemas Específicos.**

- ¿Se podrá identificar las características mecánicas y electrónicas de la cortadora laser DGL-6060 para la implementacion de un extrusor para impresión 3D?
- ¿Se podrá diseñar el sistema de control para la impresión 3D, que permita leer los archivos generados, comandando los motores y control de temperatura de extrusor, cama caliente ?
- ¿Es posible adaptar el diseño mecánico y electrónico de la cortadora laser DGL-6060; con un sistema dual que incorpora un extrusor y cama caliente para la impresión 3D?
- ¿Es posible diseñar un sistema dual y una interfaz IOT para inventario remoto; incorporando los software y controladores pertinentes del corte laser e impresión 3D?

# **3. Objetivos**

## **3.1. Objetivo General.**

- Diseñar un sistema dual para la cortadora laser DGL-6060 con impresión 3D

## **3.2. Objetivos Específicos.**

- Identificar las características mecánicas y electrónicas de la cortadora laser DGL-6060 para implementar un extrusor para impresión 3D.
- Diseñar el sistema de control para la impresión 3D, que permita leer los archivos generados, comandando los motores y control de temperatura de extrusor, cama caliente.
- Adaptar el diseño mecánico y electrónico de la cortadora laser DGL-6060; con un sistema dual que incorpora un extrusor y cama caliente para la impresión 3D.
- Diseñar un sistema dual y una interfaz IOT para inventario remoto; incorporando los softwares y controladores pertinentes del corte laser e impresión 3D.

### **3.3. Hipótesis General**

- Se diseñará una cortadora laser DGL-6060 implementando un sistema dual con impresión 3D.

### **3.4. Hipótesis Específicas**

- Se identificará las características mecánicas y electrónicas de la cortadora laser DGL-6060 para implementar un extrusor para impresión 3D
- Se diseñará el sistema de control para la impresión 3D, que permita leer los archivos generados, comandando los motores y control de temperatura de extrusor, cama caliente.
- Se adaptará el diseño mecánico y electrónico de la cortadora laser DGL-6060, con un sistema dual que incorpora un extrusor y cama caliente para la impresión 3D.
- Se diseñará un sistema dual y una interfaz IOT para inventario remoto, incorporando los softwares y controladores pertinentes del corte laser e impresión 3D.

## **4. Matriz de consistencia**

**Tabla 2:** Matriz de consistencia lógica.

MATRIZ DE CONSISTENCIA LÓGICA				
PROBLEMA GENERAL	OBJETIVO GENERAL	HIPÓTESIS GENERAL	VARIABLES	MARCO TEORICO
¿Es posible Diseñar una cortadora laser DGL-6060 implementando un sistema dual con impresión 3D?	Diseñar un sistema dual para la cortadora laser DGL-6060 con impresión 3D.	Se podrá diseñar una cortadora laser DGL-6060 implementando un sistema dual con impresión 3D.	VI: Diseño VD1:Sistema Dual VD2: Impresión 3D	Diseño Impresión 3D
<b>ESPECÍFICOS</b> ¿Se podrá identificar las características mecánicas y electrónicas de la cortadora laser DGL-6060 para implementar un extrusor para impresión 3D?	<b>ESPECÍFICOS</b> Identificar las características mecánicas y electrónicas de la cortadora laser DGL-6060 para implementar un extrusor para impresión 3D.	<b>ESPECÍFICAS</b> Se identificará las características mecánicas y electrónicas de la cortadora laser DGL-6060 para implementar un extrusor para impresión 3D	VI: Diseño VD1:Sistema Dual VD2: Impresión 3D	Cortadoras Laser
¿Se podrá diseñar el sistema de control para la impresión 3D, que permita leer los archivos generados, comandando los motores y control de temperatura de extrusor, cama caliente ?	Diseñar el sistema de control para la impresión 3D, que permita leer los archivos generado, comandando los motores y control de temperatura de extrusor, cama caliente.	Se diseñará el sistema de control para la impresión 3D, que permita leer los archivos generado, comandando los motores y control de temperatura de extrusor, cama caliente.	VI: Diseño VD1:Sistema Dual VD2: Impresión 3D	Control de temperatura en impresiones 3D Control de impresiones 3D
¿Es posible adaptar el diseño mecánico y electrónico de la cortadora laser DGL-6060; con un sistema dual que incorpora un extrusor y cama caliente para la impresión 3D?	Adaptar el diseño mecánico y electrónico de la cortadora laser DGL-6060; con un sistema dual que incorpora un extrusor y cama caliente para la impresión 3D.	Se adaptará el diseño mecánico y electrónico de la cortadora laser DGL-6060; con un sistema dual que incorpora un extrusor y cama caliente para la impresión 3D.	VI: Diseño VD1:Sistema Dual VD2: Impresión 3D	Impresión 3D y Sistemas de control
¿Es posible diseñar un sistema dual y una interfaz IOT para inventario remoto; incorporando los softwares y controladores pertinentes del corte laser e impresión 3D?	Diseñar un sistema dual y una interfaz IOT para inventario remoto; incorporando los software y controladores pertinentes del corte laser e impresión 3D.	Se diseñará un sistema dual y una interfaz IOT para inventario remoto; incorporando los software y controladores pertinentes del corte laser e impresión 3D.	VI: Diseño VD1:Sistema Dual VD2: Impresión 3D	interfaz Iot, Software, controladores

**Fuente:** Autores

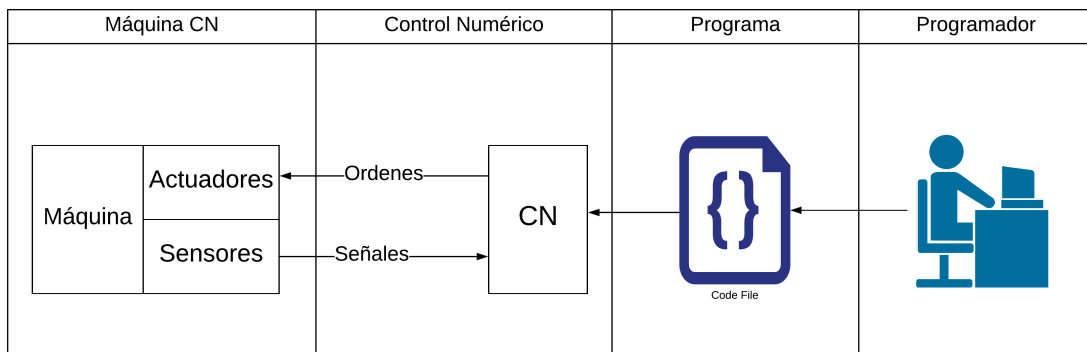
## 5. Marco Teórico

### 5.1. Maquinaria CNC

Dentro del área tecnológica de los procesos de manufactura se ha hecho más frecuente el uso de máquinas CNC, anteriormente, las maquinas-herramientas eran operadas de forma manual y la calidad y repetibilidad de las piezas dependía mayormente de la habilidad del operario.(Arriaga Segundo, 2004)

Gonzalez Contreras y Rosado Castellano (2015) definen el control numérico (CN) como “una forma de automatización programable en la que el proceso se controla con programas basados en números, letras y símbolos. La ejecución de una determinada maquina requiere de unos movimientos concretos de sus ejes”. Esta automatización permite controlar movimientos y velocidades, permitiendo cambiar de programación cada vez que se requiera una nueva pieza, en vez de cambiar la configuración de toda la estación de trabajo. En la figura 2 se muestra el esquema de funcionamiento de una máquina CN, en la cual constan los elementos básicos como son: la máquina-herramienta, el programa y el control numérico; y su interacción dentro de los procesos de fabricación.

**Figura 2:** Esquema de funcionamiento de una máquina de control numérico.



**Fuente:** Gonzalez Contreras y Rosado Castellano (2015)

Dentro de la evolución del control numérico se presenta actualmente la cuarta generación, en la cual se mantienen los principios del control numérico y varían las aplicaciones de la tecnología; con la llegada de los microprocesadores y la microtecnología se disminuyó el tamaño de las computadoras, lo que facilita la inserción de lógica de programación en computadoras lo que da paso al control numérico computarizado (CNC).(Arriaga Segundo, 2004)

Ventajas del uso de CNC:

- Reducción de los tiempos no productivos
- Aumento de la rentabilidad de las herramientas
- Mayor precisión y repetibilidad en las piezas
- Dejar de depender de la habilidad del operario y del número de operarios
- Posibilidad de mecanizar piezas con geometría compleja

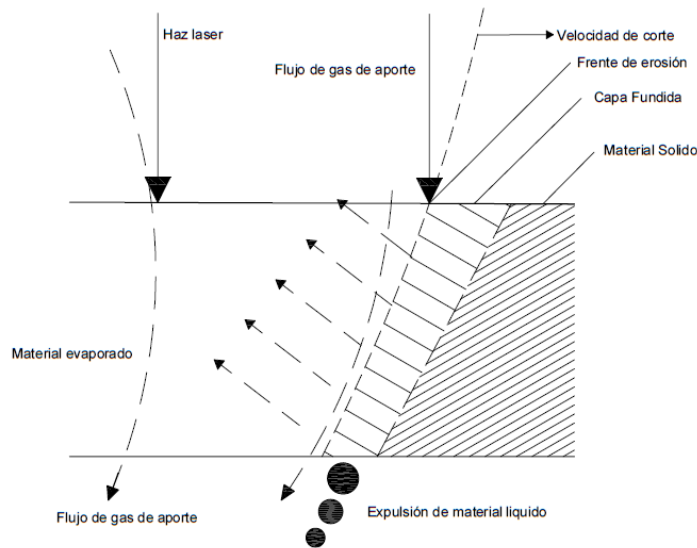
### **5.1.1. Cortadora Laser**

Dentro de las tecnologías de manufactura, el corte a laser aparece como un proceso de mecanizado con arranque de viruta (CAV) siendo uno de los métodos no convencionales ya que su aparición es relativamente reciente.

Para Bachs (2009) “El corte con láser es un proceso inducido térmicamente, en el cual la energía de un haz de luz focalizado es absorbida por el material, que se vaporiza formando así el corte.” Al ser un proceso térmico existen factores que facilitan el corte laser, estos factores son las propiedades físicas de los materiales a mecanizar como lo son las propiedades térmicas, eléctricas, ópticas y mecánicas; siendo importante también la capacidad de absorción de radiación por parte de los materiales a mecanizar, ya que hay una gran diferencia entre los materiales metálicos y aislantes, ya que para los aislantes la absorción aumenta con la longitud de onda, mientras que para los materiales metálicos disminuye con la longitud de onda.

El mecanismo de corte a laser se realiza por la evaporación del material, al ser inducido térmicamente, la energía necesaria para evaporar la pieza es obtenida de un sistema formado por un haz laser y una boquilla que permite el flujo de gas de aporte. La transmisión de calor se produce focalizando el haz de luz sobre la pieza, utiliza el flujo coaxial de gas de aporte que contribuye a la expulsión de material, potencia la oxidación del material y protege el laser de las partículas procedentes del corte. Para realizar el corte se requiere que el haz de luz y el gas de aporte incidan sobre la superficie de corte, parte de la luz del laser se refleja y no se aprovecha; por lo que la energía absorbida por el material tiene que subir la temperatura para provocar la evaporación del material como se puede observar en la figura 3.

**Figura 3:** Proceso del corte laser.



**Fuente:** Bachs (2009)

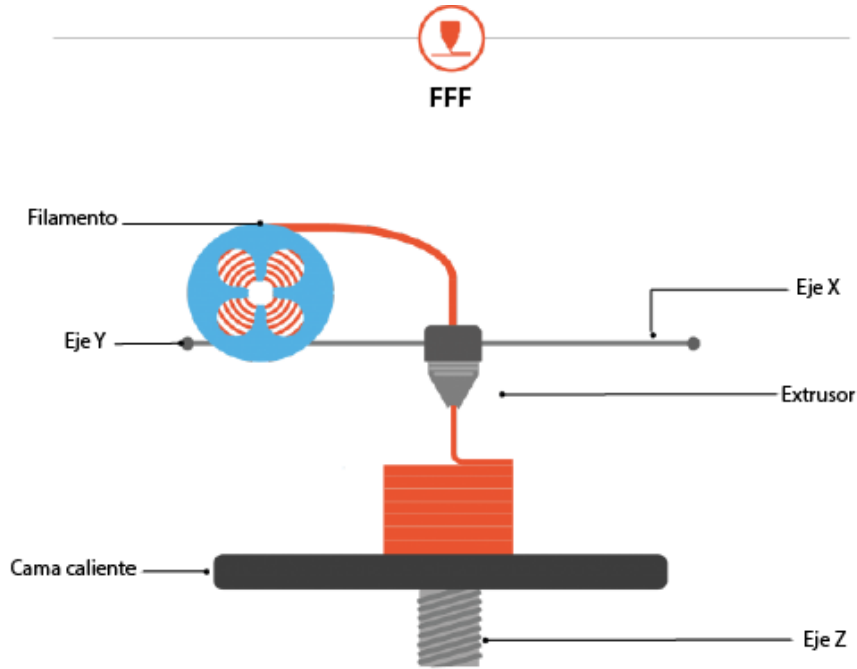
### 5.1.2. Impresora 3D

Las impresoras 3D son máquinas que nos ayudan a realizar el proceso de fabricación aditiva, estos procesos son recientes y más utilizados para prototipado. Para Jorquera Ortega (2017) “Fabricación aditiva es el nombre técnico que engloba todas las tecnologías de impresión 3D, se trata de la fabricación de objetos tridimensionales por aportación de material en vez de sustracción”. Las impresoras parten de un modelo en 3D y con ayuda de varios softwares, obtener un objeto físico.

Uno de los métodos de fabricación 3D más conocidos y utilizados es el método de Fabricación por Filamento Fundido(FFF), como se observa en la figura 4; consiste en depositar sucesivamente filamento del material, mediante una boquilla de extrusión, la cual se calienta a una temperatura de fusión del material utilizado de 185°C dependiendo el material de impresión, el material se deposita en capas muy finas las cuales inciden el acabado final de la pieza. Para el sistema de fabricación FFF es más común el uso de materiales termoplásticos como el ABS o PLA; aunque algunas impresoras también usan policarbonatos (PC), policaprolactonas (PCL), polifenilensulfonas (PPSF). También, el cabezal de impresión puede sustituirse por una jeringa que permite imprimir materiales como alimentos o materiales orgánicos como células.(Berchon y Luyt, 2016)

Con el fin de abaratar costos en 2005, Adran Bowyer creó el proyecto RepRap, que

**Figura 4:** Proceso de impresión 3D mediante FDM.



**Fuente:** Younext (2020)

consiste en diseñar impresoras 3D de código abierto y autoreplicante, así se permitió la accesibilidad de esta tecnología al público en general. Vale recalcar que el nombre de Modelado por deposición fundida y las siglas FDM, son marcas registradas por Stratasys, por lo que el proyecto RepRap creó el término Fused Filament Fabrication (FFF) que es de licencia pública. (Bordignon y colaboradores, 2018)

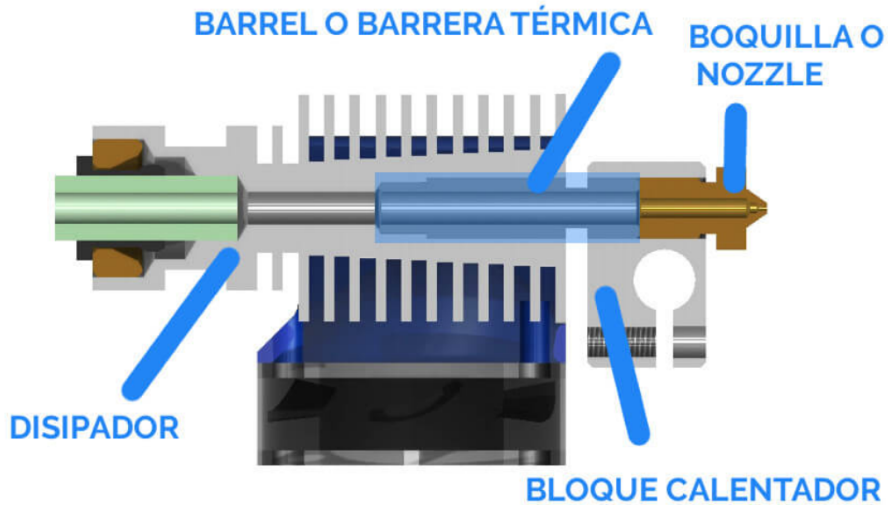
## **5.2. Arquitectura básica de maquinaria de impresión 3D de FFF**

Una impresora 3D tiene una configuración básica que consta de un cabezal extrusor que calienta y deposita el material, la parte mecánica encargada del movimiento de la boquilla y una cama caliente que mantenga la temperatura de la base, además de un sistema de control que nos permite comandar las acciones de la impresora. Estos elementos se combinan sinérgicamente para lograr materializar el diseño 3D en un objeto físico (Bordignon y colaboradores, 2018)

### 5.2.1. Hotend

El hotend se encarga de fundir el plástico para poder crear los finos hilos que formarán las capas de las piezas. Como se puede observar en la figura 5, un hotend se compone de las siguientes partes:

**Figura 5:** Partes del hotend.



**Fuente:** Lorenzo (2021)

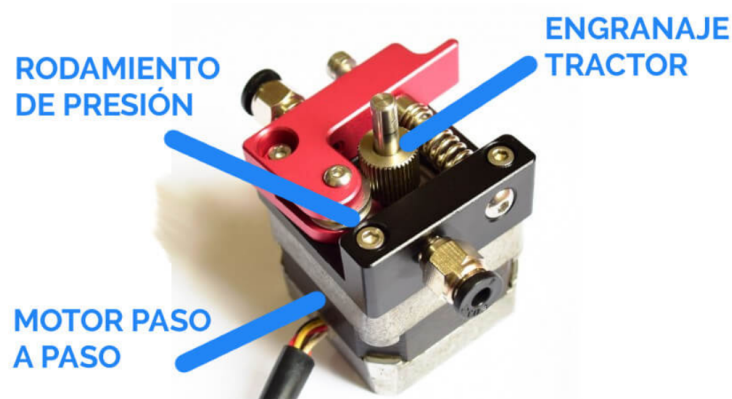
- **Boquilla o nozzle:** Generalmente es de 0,4mm de diámetro y es la que determina el diámetro del filamento fundido que creará las capas, existe una gran gama de medidas que van desde 0.20 mm, y también influye en la altura de las capas de impresión, mientras más alta sea la capa menor será el tiempo y la precisión de la impresión.
- **Bloque calentador:** Este bloque cuenta con una resistencia de cartucho la cual al aplicarle una corriente eléctrica se calienta transmitiendo calor al bloque y a la boquilla, también cuenta con un termistor que actúa como sensor de temperatura y nos ayuda al control de esta.
- **Barrera térmica o barrel:** Es un aislante que crea una barrera en el hotend que impide que el calor del bloque calentador se disipe rápidamente hacia arriba, pudiendo derretir el plástico antes de lo necesario.
- **Disipador:** permite disipar el calor del hotend de forma conveniente



### 5.2.2. Extrusor

El extrusor se encarga de empujar el filamento hacia el hotend, creando un flujo de material para la impresión como se puede ver en la figura 7 consta de 3 partes:

**Figura 6:** Partes del Extrusor.



**Fuente:** Lorenzo (2021)

- **Motor paso a paso:** se encarga de transmitir fuerza a la rueda dentada que agarra el filamento. A medida que gira, el filamento se extruirá más o menos rápido.
- **Engranaje tractor:** Convierte el movimiento circular del motor en uno lineal haciendo que el material de impresión avance, tiene forma dentada lo que facilita el agarre con el filamento.
- **Rodamiento de presión:** ayuda al empuje y adherencia del material con el engranaje tractor.

### 5.2.3. Cama Caliente

La cama caliente es una superficie que mediante una resistencia eléctrica se calienta y transfiere el calor a la base de la impresora 3D, para que las piezas pueden adherirse durante la impresión. De esta forma se evita o reduce el uno de los errores más frecuentes de la impresión como es el wrapping, el cual consiste en que las piezas se despegan y se deforman durante el proceso de impresión (Lorenzo, 2021). Al tener la cama caliente de la impresora a una temperatura adecuada, el material cuando es extruido no experimentará un cambio abrupto en la temperatura respecto a cuando sale del HotEnd y por tanto mejora la adherencia. (Javier y colaboradores, 2018)

#### 5.2.4. Sistema De Movimiento de Ejes

Generalmente las impresoras 3D del proyecto RepRap tienen la siguiente configuración de mecanismos para el movimiento del hotend en los diferentes ejes.

- **Eje X:** Genera el movimiento mediante el uso de un mecanismo de faja sincrónica que es impulsado por motores a pasos. Además, consta de varillas lisas o guías lineales y en este mecanismo se desliza el carro que contiene el hotend.
- **Eje Y:** Este movimiento viene dado por la plataforma que contiene la cama caliente, el movimiento se genera de los motores y se transmite mediante una faja sincrónica.
- **Eje Z:** Generado con los motores a pasos y transmitido mediante 2 tornillos de potencia que elevan-bajan el sistema del eje x; aunque en algunos casos que se eleva la cama caliente en vez del eje x.

#### 5.2.5. Sistema De Control

La impresora cuenta con una placa electrónica en la cual se encuentra el microcontrolador que recibe señales de los sensores, las procesa y envía órdenes a los actuadores para el proceso de impresión. Se cuenta también con drivers para el control de los motores a pasos. Estas placas pueden ser alimentadas con 12v o 24v.

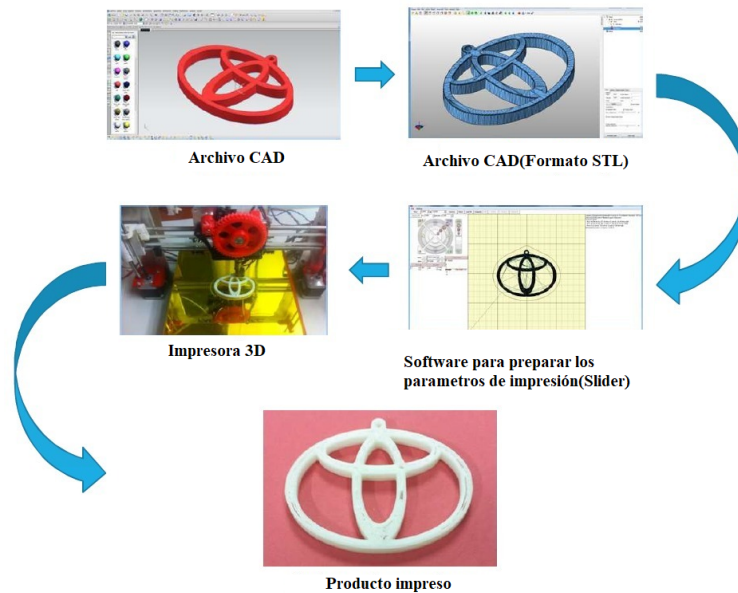
Para el software, la mayoría de impresoras cuentan con Marlin, el cual se puede editar con Arduino, este se encarga de recibir el código G para traducirlo a movimientos de ejes, el control de la temperatura tanto de la cama caliente como del hotend y el avance del material.

Para obtener las señales del entorno se cuenta con algunos sensores, los sensores de temperatura son termistores que toman las señales de la cama caliente y del hotend. Para obtener las señales de posiciones de los ejes, se utilizan sensores de fin carrera los cuales están para localizar el 0 máquina y el recorrido final del eje. (Lorenzo, 2021)

### 5.3. Proceso de impresión 3D

EL proceso de impresión 3D se fundamenta en pasos básicos que se describe en la imagen a continuación.

**Figura 7:** Proceso de impresión 3D.



**Fuente:** Cobos y colaboradores (2019)

Inicialmente, se debe obtener un modelo 3D del objeto mediante un programa CAD (Diseño Asistido por Computadora), el modelo debe ser exportado en formato STL.

STL (siglas provenientes del inglés stereolithography) es el formato estándar de archivos de prototipado rápido. Estos archivos utilizan mallas triangulares para definir la forma del objeto. Mientras más pequeños son los elementos de la malla, mejor es la aproximación del modelo a la superficie y la resolución, pero esto requerirá mayor tiempo de fabricación del prototipo. (Bordignon y colaboradores, 2018)

Posterior a esto, el archivo se lo traduce a GCode configurando previamente los parámetros de impresión mediante el software, estos softwares son llamados comúnmente Slicers. Cuando se transforma un archivo STL a código G, se está convirtiendo un modelo 3D en una serie de pasos que va a tener que seguir la impresora para construir ese objeto. (Bordignon y colaboradores, 2018)

El archivo GCode es enviado al firmware de la impresora donde se coordinan los motores y el extrusor para que vaya depositando el filamento capa a capa según las instrucciones del documento para así obtener el producto final.

## 5.4. Sistema de control

Los sistemas de control han tomado un rol importante en el desarrollo y avances tecnológicos, estando presentes en gran cantidad en los diferentes sectores de la industria. Los elementos básicos de un sistema de control se pueden describir mediante tres componentes como son: las entradas o señales actuantes, los elementos de control y las salidas o variables controladas. En general, el objetivo de estos sistemas es realizar el control de las salidas mediante las entradas a través de los elementos que componen el mismo .(Kuo, 1996)

### 5.4.1. Sistemas de control discreto

Los sistemas de control discretos son aquellos que utilizan señales discretas en lugar de las señales continuas. Un sistema de este tipo responde ante una secuencia a la entrada con otra secuencia a su salida, las cuales van a depender de las características del sistema y de la entrada. Se puede definir a esta secuencia como cualquier conjunto numerado de elementos, la cual generalmente está representada por  $\{x_k\}$  siendo  $K$  el índice del elemento dentro de la misma .

$$\{ \dots, x_{-2}, x_{-1}, x_0, x_1, x_2, x_3, \dots \} \Leftrightarrow \{ x_k \}$$

En un sistema continuo su comportamiento viene dado por las ecuaciones diferenciales, mientras que los sistemas discretos vienen caracterizados por una ecuación en diferencias.(García, 2003)

### 5.4.2. Función de transferencia $Z$

En un sistema continuo su comportamiento viene dado por las ecuaciones diferenciales, mientras que los sistemas discretos vienen caracterizados por una ecuación en diferencias. La función de transferencia en  $Z$  permite analizar el comportamiento de un sistema discreto. Si se tiene un sistema discreto definido por su ecuación en diferencias:

$$y_k + a_1 y_{k-1} + \dots + a_n y_{k-n} = b_0 u_k + b_1 u_{k-1} + \dots + b_m u_{k-m}$$

Donde  $y_k$  es la secuencia de salida del sistema discreto y  $u_k$  su secuencia de entrada, la función de transferencia en  $Z$  de este sistema viene dado por :

$$G(z) = \frac{Y_z}{U_z} = \frac{b_0 + b_1 z^{-1} + \dots + b_m z^{-m}}{1 + a_1 z^{-1} + \dots + a_n z^{-n}}$$

Esta ecuación de transferencia define el comportamiento por completo del sistema. (García, 2003)

### 5.4.3. Controlador PID

El controlador Proporcional-Integral-Derivativo (PID) es un controlador de tres términos que se han utilizado más ampliamente en la industria debido a su simplicidad, buen rendimiento de control y excelente solidez a las incertidumbres. Un control PID puede ser definido como un control retroalimentado que corrige el error entre un valor medido y el valor que se desea obtener o valor de referencia, mediante el cálculo y posteriormente realizando una acción correctiva que ajusta el proceso al objetivo de control. Todo esto mediante a 3 componentes: el proporcional, el integral y el derivativo. Se puede definir entonces la ecuación del controlador PID en tiempo continuo de la siguiente manera:

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (1)$$

donde:

$$K_i = \frac{K_p}{T_i} \quad (2)$$

$$K_d = K_p T_d \quad (3)$$

Siendo  $K_p$ ,  $K_i$  y  $K_d$  las constantes proporcional, integral y derivativa respectivamente y  $e(t)$  el error en estado estacionario en función del tiempo.

Para implementar el bloque PID digital es necesario convertir la ecuación (1) de una representación continua a una discreta. Existen varios métodos para llevarlo a cabo. Uno de ellos es utilizar las definiciones estudiadas en Cálculo para aproximar las operaciones involucradas. Por tanto, para la integral se utilizará la aproximación trapezoidal, y para la derivada la aproximación por diferencias finitas:

$$\int_0^t e(t) dt \approx \sum_{k=1}^n e(t_k) \Delta t \quad (4)$$

$$\frac{de(t)}{dt} \approx \frac{e(t_k) - e(t_{k-1})}{\Delta t} \quad (5)$$

donde :

$\Delta t$  es en intervalo de muestreo en segundos.(Ochoa Correa, 2013)

Por lo tanto, la ecuación 1 quedaría de la siguiente manera:

$$u(t_k) = K_p e(t_k) + K_i \sum_{k-1}^n e(t_k) \Delta t + K_d \frac{e(t_k) - e(t_{k-1})}{\Delta t} \quad (6)$$

,

## 5.5. Internet de las cosas (IoT)

IoT por sus siglas en inglés (Internet of Things) o internet de las cosas puede ser definido como una arquitectura basada en el uso del internet global para de esta manera facilitar el intercambio de bienes y servicios entre las cadenas de suministros, teniendo un papel fundamental en la seguridad y privacidad de los actores involucrados. Es una arquitectura emergente basada en la Internet global que facilita el intercambio de bienes y servicios entre redes de la cadena de suministro y que tiene un impacto importante en la seguridad y privacidad de los actores involucrados. De esta manera, la IoT se refiere a la interconexión en la red de distintos objetos cotidianos, que a menudo están equipados con algún tipo de inteligencia, por lo que también puede ser visto como una plataforma para dispositivos, los cuales se comunican electrónicamente y comparten información y datos específicos del mundo a su alrededor.(Weber, 2010)

## 6. Marco metodológico

### 6.1. Identificación de las características mecánicas y electrónicas de la cortadora laser DGL-6060 para implementar un extrusor para impresión 3D

#### 6.1.1. Componentes Mecánicos de la maquina DGL-6060

La cortadora laser DGL-6060 de la empresa Dumenlaser con 8 años en el mercado, ha brindado a sus clientes una máquina de corte a laser confiable y de gran rendimiento, en donde resaltan sus características mecánicas y de control ya que están optimizadas para evitar las vibraciones, mejorando la precisión de los cortes. Así que procedemos a describir los componentes que posee esta máquina de corte a laser.

- **Sistema de ejes** La máquina presenta un sistema de ejes cartesiano para el movimiento del láser en el cual el espacio de trabajo tenemos que la maquina DGL-6060 tiene una carrera de 600mm en el eje X, 600mm en el eje Y y 150mm en el eje Z.
  - Eje X y Y: Para el desplazamiento de estos ejes se usan motores a paso y faja sincrónica, en la cual se mueve el carro que contiene el hotend y el extrusor
  - Eje Z: Para el movimiento vertical se tienen 2 tornillos de potencia colocados a cada lado de la superficie de corte.

- **Motores**

Para generar el movimiento en los 3 ejes se cuenta con motores a pasos, los cuales tiene 1.2 grados por paso y trabajan con 5.8A, además cuenta con un par de torsión 6.8N.cm. Por las características de los motores a pasos son utilizados en sistemas de control, ya que pueden controlar la posición de un eje o piezas de mecanismos. El giro de los motores a pasos viene dado por cada pulso que recibe su unidad de control, mientras que la velocidad está dada por el número de pulsos que entran en una unidad de tiempo.(Chapman, 2012)

### 6.1.2. Componentes Electrónicos

- **CPU**

Para el sistema de CNC no es necesario una gran arquitectura computacional, porque no requiere un gran procesamiento de datos. Los procesos que se realizan en el CPU son la generación de trayectorias a partir de la lectura del diseño. Una parte importante del cálculo de las trayectorias debe ser en tiempo real ya que se debe emparejar con el monitoreo constante de las señales de los sensores y enviar los datos a los actuadores en tiempos precisos. Para la visualización de la interfaz gráfica si se requiere un mayor procesamiento de datos, en el caso de la cortadora DGL-6060, la interfaz es sencilla y con el procesador interno del CPU es capaz de trabajar con fluidez. La cortadora DGL-6060 cuenta con un procesador Intel Core 2 Duos, el cual posee 2 núcleos, con 2 subprocesos; trabaja a una frecuencia básica de 3 GHz y a una velocidad de bus de 1333MHz y una memoria caché de 6MB.(Intel Corporation, 2021)

- **Tarjeta Controladora**

Como parte del sistema CNC, se tiene la tarjeta controladora, que es una tarjeta de interconexión de componentes periféricos (PCI), la cual conecta el procesador con los actuadores y sensores de la máquina CNC; por lo que esta tarjeta debe de alta velocidad y acoplarse al sistema operativo en tiempo real.

La información que es generada en el procesador mediante la compilación del código g, es recibida por la tarjeta controladora mediante trenes de pulsos y esta a su vez es enviada a los actuadores para generar el movimiento. También, la tarjeta controladora recibe información de los sensores del sistema mecánico para comunicarlos con el procesador y así monitorear el correcto funcionamiento del sistema.

Una de las ventajas de utilizar la tarjeta PCI es que se evita que los puertos de la tarjeta madre se expongan a un sobre voltaje, puesto que, al conectar la tarjeta controladora, el CPU apaga las demás entradas de propósito general, esto funciona como un sistema de seguridad para evitar que las fuentes que alimentan los módulos de potencia puedan que entrar y quemar los puertos de la placa madre.

Las tarjetas controladoras que utiliza la cortadora laser DGL-6060 son las tarjetas de Mesa Electronics. Son tarjetas de gran versatilidad y de bajo costo en el mercado, por lo que son muy accesibles. Cuentan con 32 bits de entradas y salidas 17 por conector. Funcionan en sistemas de 3.3v y 5v, además todas las entradas y salidas cuentan con resistencias pull-up, dando un estado definido al momento del encendido.

Estas tarjetas presentan configuraciones para generar pasos, encoders, PWM, E/S analógicas, E/S digitales y para comunicación serial. Es compatible con tarjetas de conexión comunes y unidades de paso de varios ejes como Probotix-RF y Gecko G540. Además, Mesa Electronics cuenta con tarjetas de expansión compatibles. (Mesa Electronics, 2021)

La tarjeta de expansión utilizada en la máquina tiene la capacidad de 4 ejes de accionamiento con motores a pasos o servomotores; además cuenta con entrada para encoders de husillo y una entrada para control de husillo analógico aislado; para la comunicación con la tarjeta controladora con la tarjeta de expansión se cuenta con puertos de comunicación RS-422, para las entradas y salidas se cuenta con borneras para la facilidad de conexión y para la comunicación con un puerto paralelo. (Mesa Electronics, 2020)

- **Driver de motores a pasos**



El controlador de los motores a pasos recibe las señales en forma de pulsos de la tarjeta controladora y los transforma en señales de potencia como voltaje y corriente que se suministra al motor, de esta manera se puede controlar tanto la dirección del motor como su velocidad.

El DM556 es un controlador paso a paso totalmente digital y versátil basado en un DSP con algoritmo de control avanzado, esto permite a los motores funcionar con un ruido mucho menor, un calentamiento mas bajo, con una mayor suavidad y permitiéndoles alcanzar una velocidad mucho mas alta, por encima de los 3000 RPM; posee las siguientes características: (Leadshine Technology Company, 2009)

- Antirresonancia, proporciona un par óptimo y anula la inestabilidad de rango medio
  - La tecnología de autocomprobación del motor y configuración automática de parámetros ofrece respuestas óptimas con diferentes motores
  - Multi-Stepping permite una entrada de pasos de baja resolución para producir una salida de micropasos más alta para un rendimiento suave del sistema
  - Resoluciones de micropasos programables, desde paso completo hasta 102.400 pasos / rev.
  - Tensión de alimentación hasta +45 VCC
  - Corriente de salida programable, de 0.5A a 5.6A
  - Frecuencia de entrada de pulsos hasta 200 KHz
  - Entrada compatible con TTL y ópticamente aislada
  - Reducción automática de la corriente inactiva
  - Adecuado para motores de 2 y 4 fases
  - Soporta modos PUL / DIR y CW / CCW
  - Protecciones de sobretensión, sobrecorriente y error de fase
- **Sensores** La cortadora laser DGL-6060 cuenta con sensores inductivos de proximidad, estos sensores detectan los metales férricos y envían una señal al controlador. Estos sensores están ubicados en las partes iniciales y finales de los 3 ejes del sistema mecánico; en el sistema de la cortadora laser estos sensores se utilizan para identificar la posición de los ejes y el carro donde se encuentra el láser, así se puede tener una referencia de la posición del 0 Maquina y la posición final de cada uno de los ejes, con el fin de no sobres forzar el motor. El sensor inductivo de proximidad tiene algunas características: (OMRON Corporation, 2020)

- Distancia de detección: 5 mm  $\pm$  10
  - Distancia de establecimiento: 0 a 4 mm
  - Recorrido diferencial: 10
  - Objeto detectable: Metal ferroso
  - Objeto detectable estándar: Hierro, 18  $\times$  18  $\times$  1 mm
  - Frecuencia de respuesta: 1500 Hz
  - Voltaje de la fuente de alimentación: 12 a 24 VCC (10 a 30 VCC)
  - Corriente de fuga: 0,8 mA máx.
  - Salida de control:
    - ◇ Corriente de carga: 3 a 100 mA
    - ◇ Voltaje residual: 3,3 V máx.
- **HMI**

Los paneles Human Machine Interface (HMI) como su nombre lo indica son equipos electrónicos cuya función es servir de interfaz entre el hombre y la máquina. Estos dispositivos permiten observar los datos para que los operarios puedan monitorear los procesos que se desarrolla en la maquina; también brinda la posibilidad de ingresar datos de ajuste, visualizar mensajes y errores, revisar mediciones y cálculos, etc. (A. Daneri, 2009)

Una de las ventajas de los HMI es que le permite al operador modificar ciertos parámetros del proceso, de una manera fácil y rápida, sin tener que alterar el programa que se encuentra en la maquina y mientras esté en ejecución.

En la maquina DGL-6060 el HMI nos permite configurar las velocidades de corte y movimiento, intensidad de láser, configurar las posiciones del 0 pieza; además de monitorear el estado del proceso.

## **6.2. Adaptación del diseño mecánico y electrónico de la cortadora laser DGL-6060, con un sistema dual con extrusor y cama caliente para la impresión 3D**

### **6.2.1. Selección de sistema de cama caliente, sistema de extrusión.**

- Cama Caliente

Para la elección de la cama caliente se debe tomar en cuenta las dimensiones del espacio de trabajo disponible para impresión, la maquina DGL-6060 cuenta con una mesa de corte de 600mm x 600mm, los cuales se puede usar el mismo tamaño de superficie como área de impresión.

Al ser esta superficie de gran tamaño, se requiere una mayor potencia para poder calentar la cama caliente; por esta razón se opto por una cama caliente que funcione con corriente alterna.

- Hotend

Para la selección del hotend, se hizo una comparativa entre 3 marcas comerciales conocidas como los son: Lite 6, V6 All Metal, Volcano. En la tabla 3 se puede observar la comparación de las características que se puso a consideración para la elección del hotend.

**Tabla 3:** Tabla comparativa para selección de Hotend

<b>Hotend</b>	<b>Lite 6</b>	<b>V6 All Metal</b>	<b>Volcano</b>
Temperatura Max	245°C	285°C	Depende la versión
Boquillas	0.1 a 0.5mm	0.1 a 0.5mm	0.6 a 1mm
Tubo de teflón	Si	No, metalico	Depende de la versión
Diámetros	1.75 a 3mm	1.75 a 3mm	1.75 a 3mm
Compatible con Bowden	Si	Si	Si

**Fuente:** Lorenzo (2021)

Las 3 opciones son compatibles con un sistema de impresión Bowden, y tienen el diámetro disponible entre 1.75mm a 3mm. Se seleccionó el hotend V6 de metal ya que alcanza una mayor temperatura y además el tubo interno no es de teflón, sino de metal lo que brinda mayor seguridad si es que las temperaturas se elevan más de lo debido. Trabajar con un rango mas elevado de temperatura nos permite tener mayor variedad de materiales para imprimir.

- Extrusor

Se escogió el sistema bowden, este sistema nos permite tener separados el hotend del extrusor, esto ayuda a reducir el peso y las vibraciones del sistema. En la tabla

4 se realizó una comparativa entre 3 marcas del mercado, con las características más importantes de cada una.(Lorenzo, 2021)

**Tabla 4:** Tabla comparativa para selección de Extrusor

<b>Extrusor</b>	<b>Mk8</b>	<b>Titan</b>	<b>Bondtech BMG</b>
Sistema de extrusión	Bowden	Bowden	Directo
Compatibilidad	<ul style="list-style-type: none"> <li>• Lite 6</li> <li>• All Metal V6</li> <li>• Volcano</li> </ul>	<ul style="list-style-type: none"> <li>• Lite 6</li> <li>• All Metal V6</li> <li>• Volcano</li> </ul>	<ul style="list-style-type: none"> <li>• Lite 6</li> <li>• All Metal V6</li> <li>• Volcano</li> </ul>
Velocidad Máxima	50 mm/s	80 mm/s	120 mm/s
Relación de transmisión	1/1	3/1	3/1
Doble engranaje	No	No	Si
Motor a pasos	Normal	Tipo Pancake	Tipo Pancake

**Fuente:** Lorenzo (2021)

Se seleccionó el extrusor E3D Titan, ya que es compatible con el sistema bowden y ofrece una mayor velocidad de impresión. Su relación de transmisión 3/1 permite que los engranes tengan un mejor agarre al filamento y además una mayor precisión con menor altura de capa; además mejora el rendimiento de los rodamientos. En caso de que el desgaste de los engranajes de plástico sea superior al esperado, se puede construir engranajes de metal en su lugar.

### 6.2.2. Diseño de acople para el hotend y extrusor en la cortadora laser

Para implementar un sistema dual, se pensó en un diseño que permita al operador o usuario, cambiar de funcionamiento la maquina de una manera fácil y rápida; analizando el espacio de trabajo y respetando la arquitectura ya existente en la maquina actual.

Como se mencionó anteriormente la maquina DGL-6060 ya cuenta con un sistema mecánico estable, que es compatible con las necesidades que se requiere para el movimiento de ejes de la impresión 3D; por lo que el diseño se enfocó en acoplar el hotend y extrusor al sistema de la máquina, por esta razón se optó por un diseño modular.

- **Diseño modular**

Para este diseño se planteó 2 módulos intercambiables, uno de los cuales será el que la cortadora posee para el láser, mientras que el otro se diseñó para impresión 3D.

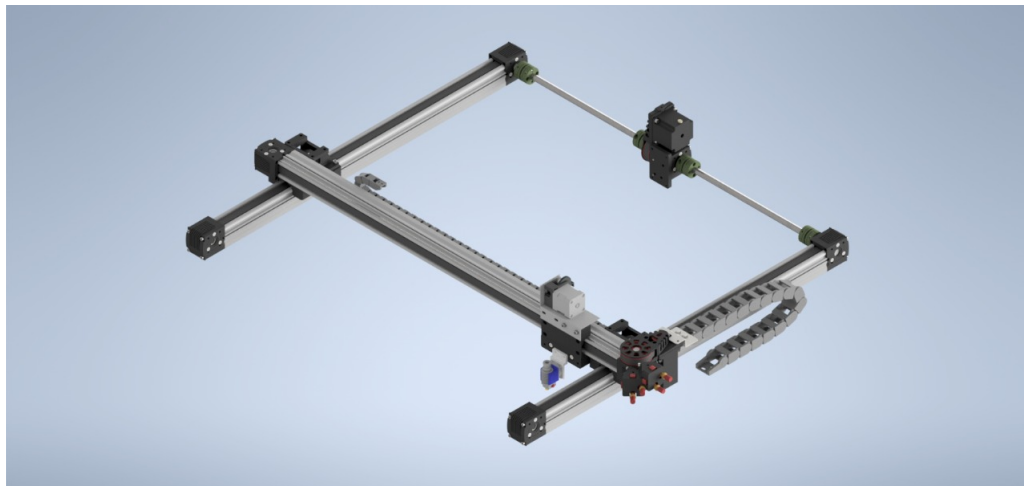
El módulo de impresión 3D, como se puede ver en la figura consta de:

- ⊗ Hotend V6 de metal
- ⊗ Pieza de acople del hotend al carro
- ⊗ Ventilador para el Hotend

Mientras que para el sistema de extrusor se diseñó una base que se sujeta en la parte de arriba del carro. Al estar el hotend alejado del extrusor se considera una extrusión de tipo bowden, como se observa en la figura 9 . Los planos de la pieza de acople y la base para el extrusor se pueden ver en los anexos.

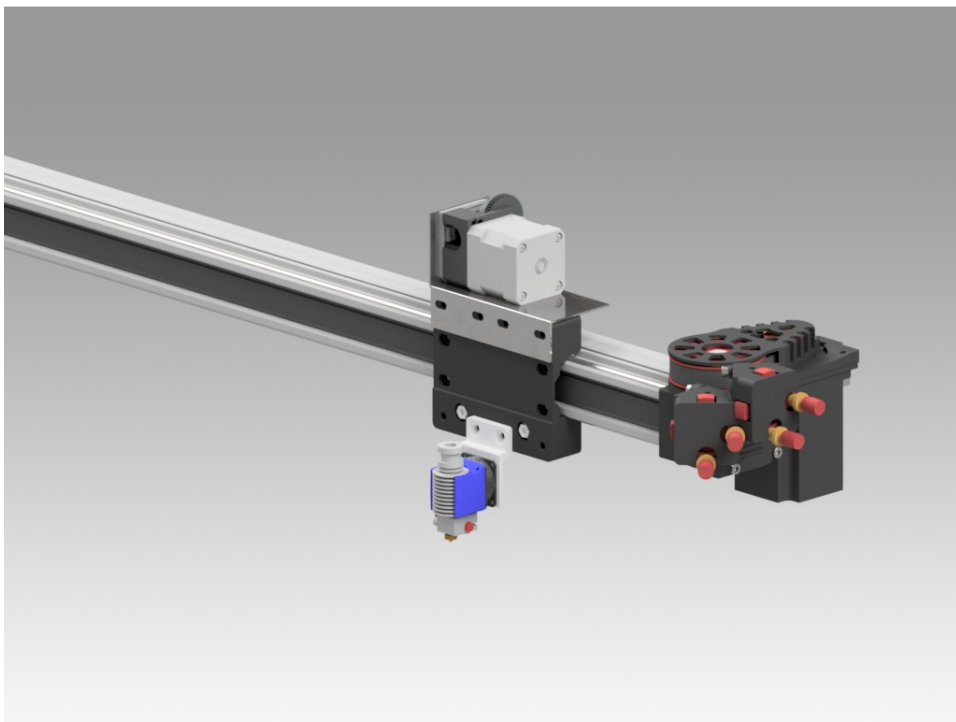
Para intercambiar estos 2 módulos solo es necesario quitar los 4 pernos que los sujetan, cambiar el módulo que se desee usar y empernarlo. Adicional a esto, es necesario configurar el nuevo 0 pieza y el 0 máquina como seguridad adicional para garantizar la presión de las diferentes herramientas en los 2 procesos.

**Figura 8:** Vista del diseño del acople realizado.



**Fuente:**Autores

**Figura 9:** Diseño modular.



**Fuente:**Autores

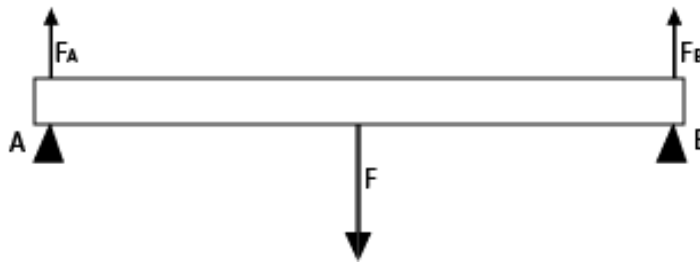
- **Cálculo de flexión en el Eje X**

Con la adición del módulo de impresión, hotend y extrusor, es necesario analizar si el peso adicional ocasiona alguna deformación en el eje X. Para esto se tomo en cuenta la situación más crítica que se pudiera afectar la estructura del eje. Mediante el análisis de diagrama de fuerzas y utilizando la gráfica de momento flector.

Para realizar el diagrama de fuerzas se analizó el peso de cada uno de los componentes del sistema modular de impresión y sumarlos para obtener una sola fuerza resultante de  $10,3N$ .

Mediante el diagrama de cuerpo libre que se puede observar en la figura 10, se pueden encontrar el valor de las fuerzas que ejercen los apoyos mediante el cálculo estático de las ecuaciones 7, 8, 9.

**Figura 10:** Diagrama de cuerpo libre del eje X .



**Fuente: Los Autores**

$$\sum Fy = 0 \quad (7)$$

$$FA + FB - F = 0 \quad (8)$$

Al ser las fuerzas  $FA$  y  $FB$  iguales y reemplazando el valor de  $F$  se obtiene como resultado:

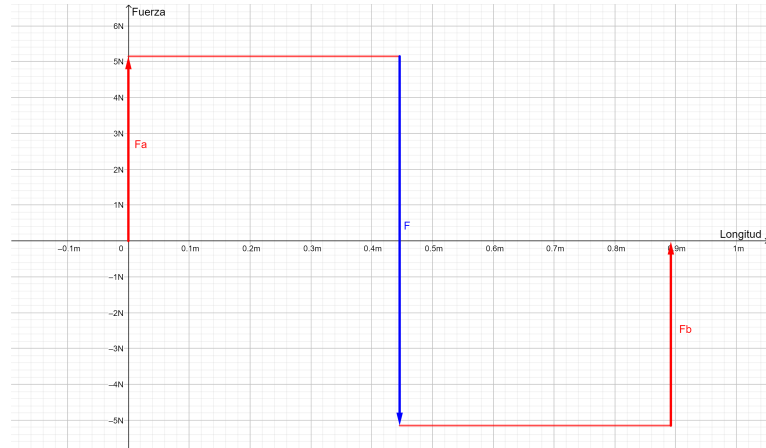
$$FA = FB = \frac{F}{2}$$

$$FA = FB = 5,15N \quad (9)$$

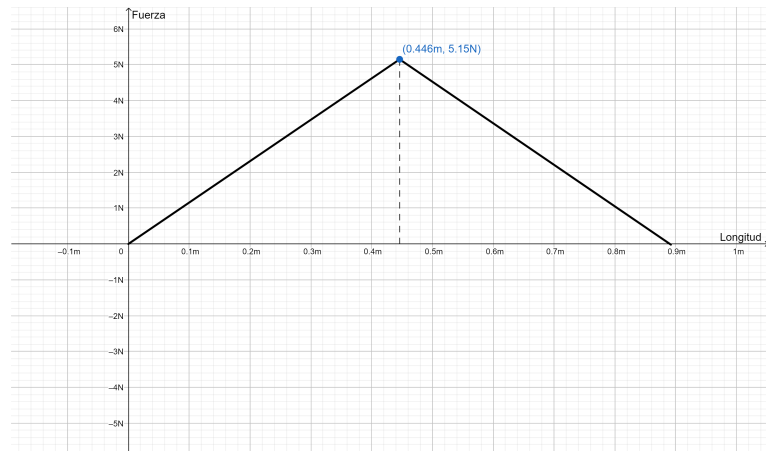
Una vez obtenidos los valores de cada una de las fuerzas que intervienen en el eje se procede a realizar el diagrama de Fuerza cortante y de momento flector como se

observan en la figura 11 y en la figura 12 respectivamente, obteniendo un momento flector  $M = 2,296Nm$ .

**Figura 11:** Diagrama Fuerza Cortante



**Figura 12:** Diagrama Momento Flector



**Fuente:** Los Autores

Para Budynas y Nisbett. (2014) el diagrama de momento flector es un método gráfico que nos facilita el cálculo del esfuerzo que viene dado por la ecuación 10.

$$\sigma_{max} = \frac{M * c}{I} \quad (10)$$

Donde:

$\sigma_{max}$ : Es el esfuerzo máximo en flexión.

$M$ : Momento flector máximo.



$c$ : Distancia desde donde se aplica la fuerza al eje neutro.

$I$ : Inercia de la viga.

$$\sigma_{max} = \frac{2,296Nm * 0,0299m}{2,40725x10^7m^4}$$
$$\sigma_{max} = 1,280MPa \quad (11)$$

Remplazando los datos tenemos que el máximo esfuerzo en flexión es de 1,280MPa, como podemos ver en la ecuación . Para la construcción del eje se tiene de material el aluminio 6063 T5, el cual consta con un de 695MPa por lo que sabemos que el material resiste el esfuerzo de flexión del sistema de impresión 3D.

### **6.2.3. Evaluación de compatibilidad de las tarjetas electrónicas**

Como se mencionó anteriormente, el sistema de la maquina DGL-6060 está compuesto por tarjetas controladoras, que interconectan el CPU con los periféricos; actuando como módulos de potencia para los actuadores. En el caso del sistema 3D, además de los actuadores con los que cuenta la máquina, se añade el motor a pasos encargado de la extrusión del filamento. Tanto, la tarjeta controladora que posee la máquina cuenta con la capacidad de añadir 2 tarjetas hijas con las cuales cada una puede controlar 2 motores a pasos. Lo que permite que las tarjetas electrónicas que controlan los actuadores sean utilizadas también para el proceso de Impresión 3D.

## **6.3. Diseñar el sistema de control para la impresión 3D que permita leer los archivos generados, comandando los motores**

### **6.3.1. Sistema de control para el movimiento en los 3 ejes.**

- **Linux CNC**

Es un software para computador que permite el control de máquinas herramienta tales como fresadoras, tornos, robots tipo puma o scara y cualquier otro tipo de máquina de hasta 9 ejes fue; creado por el Instituto Nacional de Normas y Tecnología de Estados Unidos (NIST) bajo el nombre de Controlador de máquina mejorado (EMC), posteriormente en 2011 se cambio de nombre al de LinuxCNC. Es software libre con un código fuente abierto, el cual cuenta con licencias de

dominio público PL y LGPL. Algunas de las características con las que cuenta el software son (LinuxCNC, 2021) :

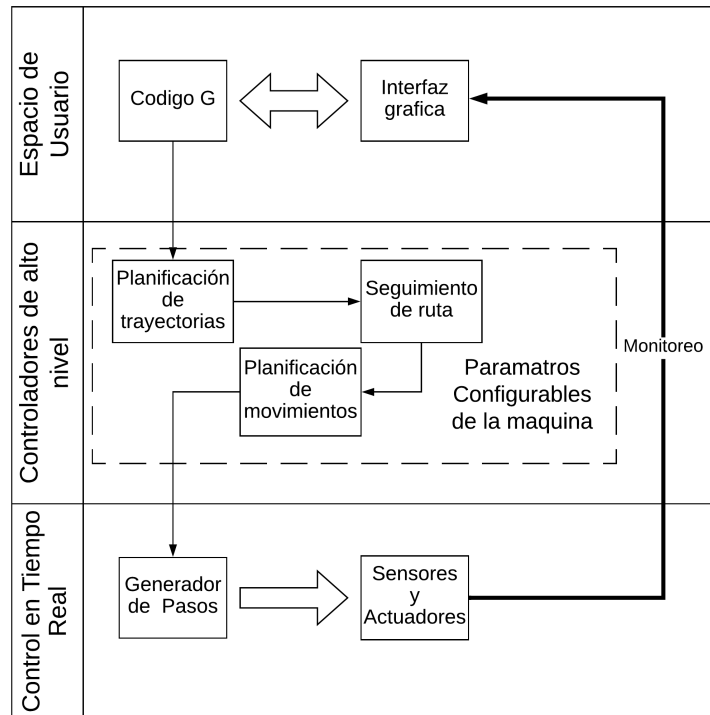
- ⊗ Sistema de interfaz gráfico compatible con varios tipos de pantallas.
- ⊗ Intérprete para código G.
- ⊗ Planeador de movimientos en tiempo real con análisis de instrucción siguiente.
- ⊗ Capacidad de operar la electrónica de máquina de bajo nivel como sensores y controladores.
- ⊗ Permite crear rápidamente configuraciones únicas para cada máquina.
- ⊗ No posee herramientas de dibujo CAD ( Dibujo asistido por computadora).
- ⊗ No posee herramientas generación de código G a partir de dibujos CAM( Manu-  
factura asistida por computadora).
- ⊗ Permite múltiples configuraciones para diferentes tipos de actuadores, ya sean motores a pasos o servomotores.

El sistema de LinuxCNC se compone de 3 partes principales (LinuxCNC, 2021):

- ⊗ Una interfaz gráfica de usuario (GUI), que forma la interfaz básica entre el operador, el software y la máquina CNC.
- ⊗ Una capa de abstracción de hardware (HAL), que proporciona un método para vincular las diversas señales virtuales internas generadas y recibidas por LinuxCNC con el mundo exterior.
- ⊗ Controladores de alto nivel que coordinan la generación y ejecución del control de movimiento de la máquina CNC, es decir, el controlador de movimiento (EMC-MOT), el controlador de entrada/salidas discretas (EMCIO) y el ejecutor de tareas (EMCTASK)

En la figura 13 se puede observar el diagrama de bloques simple, sobre el sistema de control que presenta LinuxCNC para el control mediante generador de pulsos para motores a pasos.

**Figura 13:** Diagrama de bloques del sistema de control de LinuxCNC



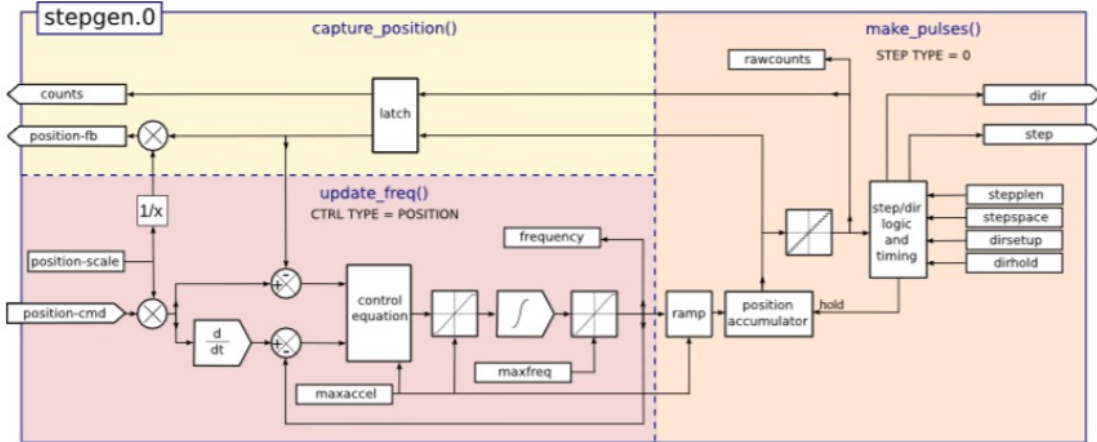
**Fuente: Los Autores**

El planificador de trayectorias sigue la ruta especificada por su programa de Código G, mientras opera dentro de los límites de la máquina, si se generan singularidades en donde se puede acceder a la misma ruta con distintos parámetros, generalmente esquinas, se genera un seguimiento de ruta que evalúa la velocidad y la longitud de la ruta para elegir la mejor opción. Finalmente, el planificador de movimiento controla las aceleraciones del sistema con relación a las distancias, materiales, parámetros de corte y la configuración de los componentes mecánicos establecidos en la máquina. (LinuxCNC, 2021).

Para el control de los motores a paso LinuxCNC genera un control en lazo cerrado mediante la señal de encoder que le brinda el driver del motor a pasos como se puede ver en el diagrama de bloques de la figura 14. El algoritmo funciona midiendo el error de posición y el error de velocidad, y calculando una aceleración que intentará reducir ambos a cero al mismo tiempo, el controlador se sintoniza de acuerdo a los parámetros de los motores y a la cinemática de la máquina; posee un bucle de posición preajustado, por lo que no se requiere sintonización PID

(LinuxCNC, 2021) .

**Figura 14:** Diagrama de bloques del control del motor a pasos



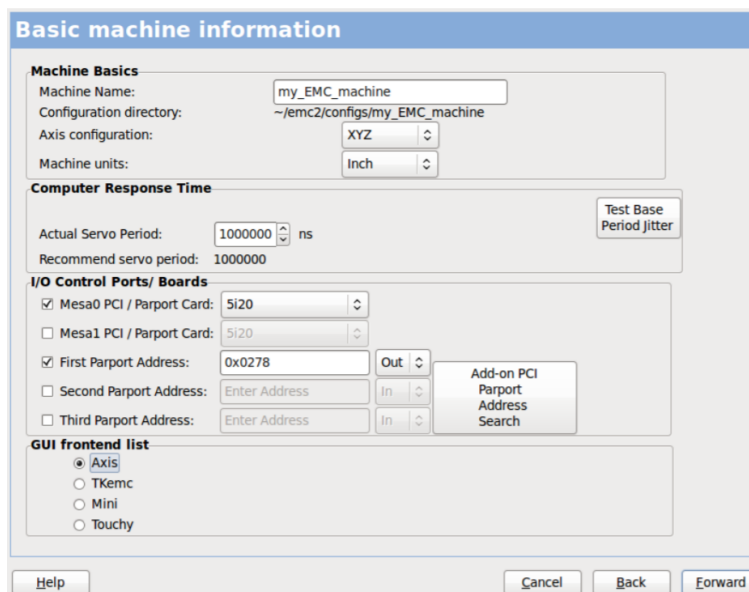
**Fuente:** LinuxCNC (2021)

- **Configuración de los parámetros de la máquina para impresión**

LinuxCNC cuenta con una herramienta llamada PNCconf, que facilita la configuración para máquinas que cuenten con tarjetas controladoras Mesa Anything I/O. Una de las ventajas de esta herramienta es que nos permite guardar la configuración para poder editar los valores en caso de ser necesario. Es necesario para empezar a configurar la máquina que las tarjetas controladoras mesa estén conectadas al CPU.

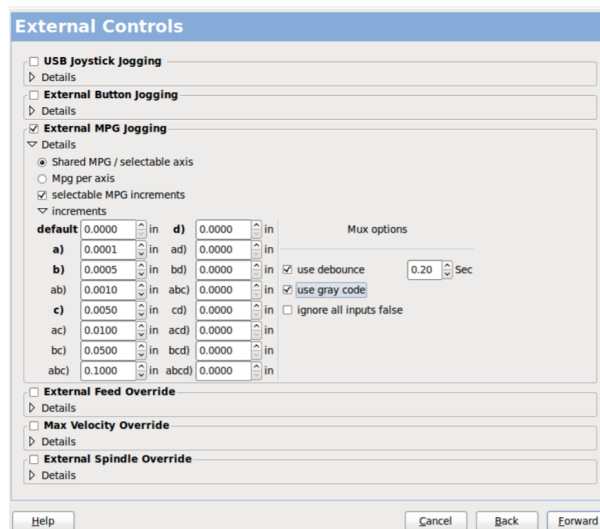
En la figura 15 se puede observar los parámetros básicos de la máquina, como son el nombre, las unidades en las que se trabajará y la configuración de los ejes que se necesite. En esta ventana se puede ejecutar la prueba del tiempo de respuesta del equipo, esto nos permitirá establecer los parámetros del sistema en tiempo real según la velocidad del CPU. Se configura el número de las tarjetas controladoras Mesa con las que trabajara la máquina y la interfaz de usuario GUI.

Figura 15: Configuración de parámetros básicos



Fuente: LinuxCNC (2021)

Figura 16: Configuración de parámetros externos



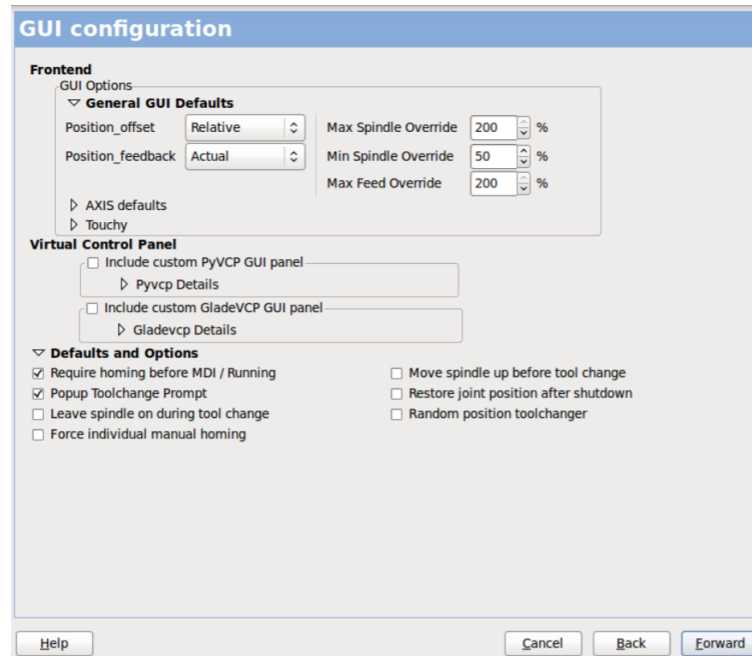
Fuente: LinuxCNC (2021)

Se puede configurar controles externos como Joystick jogging o botones como se ve en la figura 16.

Para la configuración del entorno GUI se puede observar en la figura 17 se configura

los parámetros básicos del entorno y se selecciona la opción de GladeVCP que permite generar paneles de control adicionales compatibles con la herramienta de programación Glade; con la cual se podrá diseñar entornos virtuales personalizados.

**Figura 17:** Configuración de la interfaz gráfica

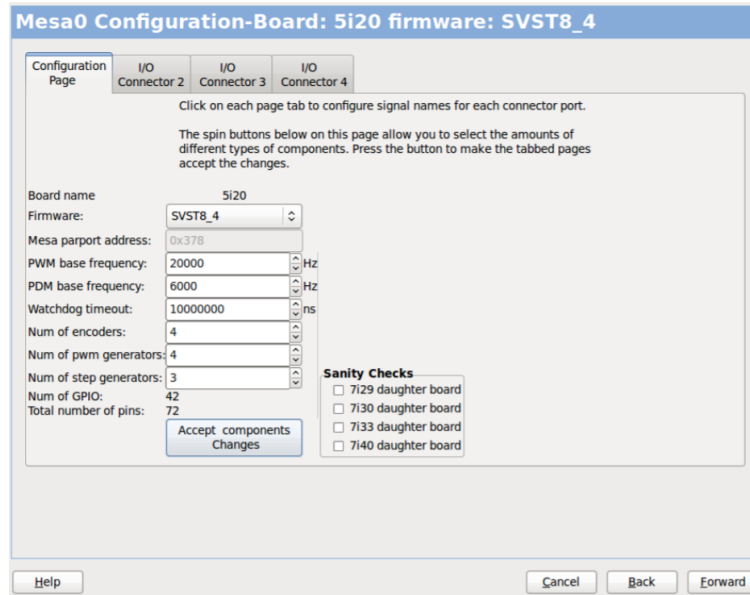


**Fuente:** LinuxCNC (2021)

En la figura 18 se observa la ventana en la que se selecciona el tipo de tarjetas controladoras Mesa, así como el número de tarjetas hijas y su forma de conexión, como la máquina posee 4 motores, se seleccionan 4 salidas de generación de pulsos y 2 tarjetas controladoras hijas.

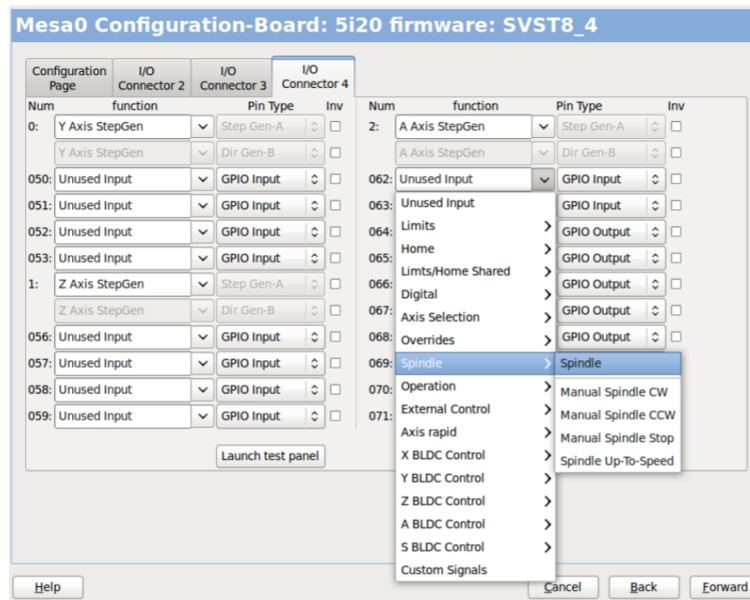
Para configurar las entradas y salidas de las tarjetas controladoras elegimos la opción de conexión 4 como se observa en la figura 19, que brinda una combinación entre pines GPIO y los generadores de pasos para los motores; en este espacio se seleccionó las salidas de los motores, a que ejes pertenece cada salida. En las entradas se pueden seleccionar las señales de los sensores que se van a conectar al sistema.

**Figura 18:** Configuración de las tarjetas Mesa



**Fuente:** LinuxCNC (2021)

**Figura 19:** Configuración de las entradas y salidas en las tarjetas Mesa

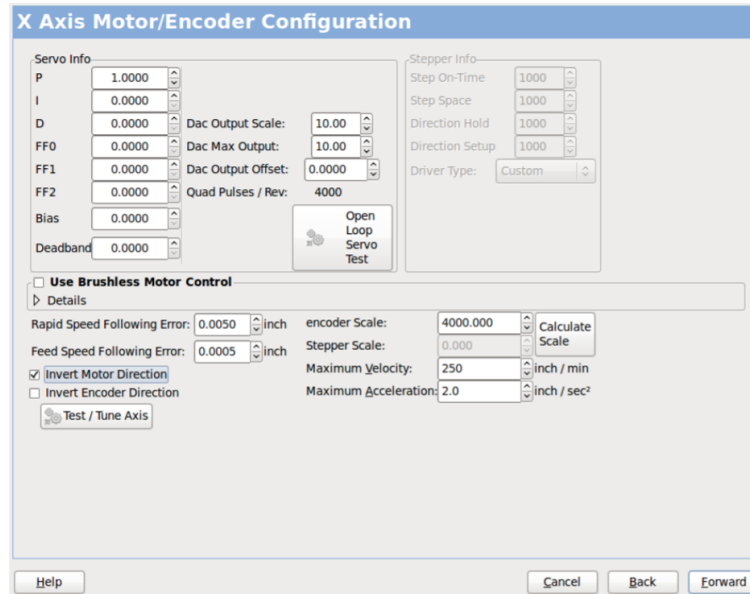


**Fuente:** LinuxCNC (2021)

En la ventana que se muestra en la figura 20 se observa la configuración de los motores, de los drivers y del mecanismo de transmisión en los ejes. Es importante

en este punto realizar la prueba de lazo abierto para comprobar la correcta sintonización del generador de pasos; al no contar con los motores se omitió este paso.

**Figura 20:** Configuración de los motores, drivers y mecanismos de los ejes



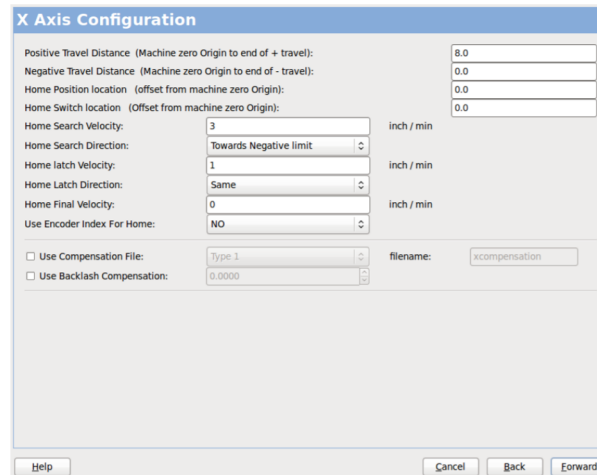
**Fuente: LinuxCNC (2021)**

La configuración de los ejes termina con los parámetros de dimensiones de la máquina, también en la ventana de la figura 21, se configura la posición del 0 Máquina.

Al finalizar esta configuración se crea una carpeta con los archivos necesarios para el funcionamiento, de los cuales el archivo .INI es el ejecutable del sistema cnc y contiene toda la información de la máquina. Mientras que el archivo .Hal tiene las configuraciones de las conexiones internas para el manejo de las señales de entrada y salida.



**Figura 21:** Configuración de las dimensiones de los ejes



**Fuente:** LinuxCNC (2021)

### 6.3.2. Adaptación de G code de impresión 3D para su funcionamiento en Linux CNC

- **Creación de códigos M necesarios para el funcionamiento de impresión 3D**

La mayoría de los comandos de LinuxCNC G Code comienzan con G o M (de General y Misceláneo). Las palabras para estos comandos se llaman códigos G y códigos M (Linuxcnc, 2021b). Entre los códigos M, existen ciertos que ya se encuentran programados para realizar una acción en concreto, pero el software de Linuxcnc también nos permite configurar nuestros propios códigos M para darles la función que nosotros deseamos, estos son los códigos M100 al M199 (Comandos definidos por el usuario) Para la creación de estos comandos definidos por el usuario , se debe crear un programa externo llamado M100 a M199 (sin extensión y M mayúscula) el cual realizará la acción deseada por el usuario (Linuxcnc, 2021a).

Dentro del archivo “.ngc” se agregará el código M definido por el usuario y además acompañarlo con los valores opcionales P y Q como sus dos argumentos si es necesario. Cuando el archivo de código G llega a un código M, la ejecución del mismo se detiene para realizar el programa externo y no continúa hasta que sale del programa externo.

Para crear el archivo externo, se puede usar cualquier archivo ejecutable válido pero el archivo debe estar ubicado en la ruta de búsqueda especificada en la

configuración del archivo ini. (Linuxcnc, 2021a)

Este conocimiento previamente mencionado es importante ya que los programas de Slicer 3D usan comandos M al generar el código G y de esta manera sestean los valores de temperatura del hotend, cama caliente y los valores de PWM de los ventiladores. Además, para la implementación de la conexión de la base de datos como parte del IoT, se usó uno de estos códigos M definidos por el usuario para realizar el envío de la información necesaria a la base de datos en la red. Actualmente los Slicer de impresión 3D tienen los mismos comandos M para las funciones básicas de la impresión como lo son :

- M104:sestea los valores de temperatura del hotend
- M140:sestea los valores de temperatura de cama caliente
- M106:sestea los valores de PWM de los ventiladores
- M107:Apagar ventiladores
- M109:confirmación de valor seteado de temperatura del extrusor
- M190:confirmación de valor seteado de temperatura de cama caliente

Por lo que nuestro archivo de código G obtenido de ellos ya tendrá estos comandos M agregados al programa. Entonces, nos toca crear los archivos M externos para que Linuxcnc pueda reconocer estos comandos como código M personalizados.

Cabe mencionar que los Slicer acompañaran a estos comandos M con la variable “S” seguido del valor que se hayan definido en el programa para cada una de estas acciones por ejemplo si dentro del Slicer definimos una temperatura del hotend de 210°C entonces el comando M que nos daría el programa serio:

M104 S210

Pero como se mencionó anteriormente los códigos M de Linuxcnc viene acompañados de los valores “P” “y” “Q” por lo que en lo posterior esta “S” debe ser remplazada por una “P” en cada uno de estos comandos. Es por esta razón que a continuación detallaremos la creación de nuestros propios archivos de código M realizados en lenguaje de programación de Python, con los cuales podemos dar la capacidad a Linuxcnc de realizar las acciones necesarias para la impresión 3D además de para la conexión IoT.

#### ⊗ Archivo M104

Este archivo está ubicado en la ruta de búsqueda especificada en la configuración del archivo ini y está escrito en el lenguaje de programación de Python y, siendo ejecutado cuando un comando M104 aparece dentro del archivo de código G en ejecución. Lo primero que hace el programa del archivo es obtiene el valor de seteo del hotend adquiriéndolo del código g que se está ejecutando y posterior a esto crea un archivo de texto de nombre “M104” con el texto "Temperatura hotend a setear:z el valor al que se quiere setear. Esto debido a que el programa donde se desarrolla la comunicación serial con el microprocesador (encargado de hacer el control de las temperaturas y ventiladores) envía el valor con el que se desea setear la temperatura del hotend al microprocesador solo cuando el archivo descrito anteriormente existe y luego de enviarlo por la comunicación serial, elimina dicho archivo para de esta manera no volver a enviar el mismo valor y realizar la acción cada vez que aparezca ese comando M dentro del código G. Para mayor comprensión de lo antes mencionado se puede apreciar el programa realizado en el Anexo 1.

#### ⊗ **Archivo M109**

Este archivo está ubicado en la ruta de búsqueda especificada en la configuración del archivo ini y está escrito en el lenguaje de programación de Python y, siendo ejecutado cuando un comando M109 aparece dentro del archivo de código G en ejecución. Lo primero que hace el programa del archivo es verificar la existencia del archivo “M109.txt”, el cual es creado por el programa donde se ejecuta la comunicación serial con el microprocesador cuando ha recibido al verificación de que el hotend se encuentra a la temperatura anteriormente seteada al momento de haberse ejecutado un código M104. Si el programa no encuentra este archivo entonces entra en un bucle a la espera de que el mismo exista, lo cual quiere decir que no se ha alcanzado la temperatura deseada, solo en el instante en el que el archivo existe se sale del bucle. Posteriormente elimina el archivo “M109.txt” y termina la ejecución del programa para que de esta manera retorne la ejecución archivo “.ngc” que se venía realizando. Para mayor comprensión de lo antes mencionado se puede apreciar el programa realizado en el Anexo 4.

#### ⊗ **Archivo M140**

Este archivo está ubicado en la ruta de búsqueda especificada en la configuración del archivo ini y está escrito en el lenguaje de programación de Python y, siendo ejecutado cuando un comando M104 aparece dentro del Archivo de código G en ejecución. Lo primero que hace el programa del archivo es obtener el valor de seteo de la cama caliente adquiriéndolo del código g que se está ejecutando y posterior a esto crea un Archivo de texto con el nombre “M140” con el texto "Temperatura de la cama caliente a setear:z el valor al que se quiere setear. Esto debido a que el programa donde se desarrolla la comunicación serial con el microprocesador (encargado de hacer el control de las temperaturas y ventiladores) envía el valor con el que se desea setear la temperatura de la cama caliente al microprocesador solo cuando el archivo descrito anteriormente existe y luego de enviarlo por la comunicación serial, elimina dicho archivo para de esta manera no volver a enviar el mismo valor y realizar la acción cada vez que aparezca ese comando M dentro del código G. Para mayor comprensión de lo antes mencionado se puede apreciar el programa realizado en el Anexo 5 .

#### ⊗ Archivo M190

Este archivo está ubicado en la ruta de búsqueda especificada en la configuración del archivo ini y está escrito en el lenguaje de programación de Python y, siendo ejecutado cuando un comando M190 aparece dentro del archivo de código G en ejecución. Lo primero que hace el programa del archivo es verificar la existencia del archivo “M190.txt”, el cual es creado por el programa donde se ejecuta la comunicación serial con el microprocesador cuando ha recibido la verificación de que la cama caliente se encuentra a la temperatura anteriormente saeteada al momento de haberse ejecutado un código M190. Si el programa no encuentra este archivo entonces entra en un bucle a la espera de que el mismo exista, lo cual quiere decir que no se ha alcanzado la temperatura deseada, solo en el instante en el archivo existe se sale del bucle. Posteriormente elimina entonces el archivo “M190.txt” y termina la ejecución del programa para que de esta manera retorne la ejecución archivo “.ngc” que se venía realizando Para mayor comprensión de lo antes mencionado se puede apreciar el programa realizado en el Anexo 6.

#### ⊗ Archivo M106

Este archivo está ubicado en la ruta de búsqueda especificada en la configuración del archivo ini y está escrito en el lenguaje de programación de Python y, siendo ejecutado cuando un comando M104 aparece dentro del archivo de código G en ejecución. Lo primero que hace el programa del archivo es obtiene el valor de seteo del PWM de los ventiladores adquiriéndolo del código g que se está ejecutando y posterior a esto crea un archivo de texto con el nombre “M106” con el texto "Setear ventilador (0-255):” y el valor al que se quiere setear. Esto debido a que el programa donde se desarrolla la comunicación serial con el microprocesador (encargado de hacer el control de las temperaturas y ventiladores) envía el valor con el que se desea setear los ventiladores al microprocesador solo cuando el archivo descrito anteriormente existe y luego de enviarlo por la comunicación serial, elimina dicho archivo para de esta manera no volver a enviar el mismo valor y realizar la acción cada vez que aparezca ese comando M dentro del código G. Para mayor comprensión de lo antes mencionado se puede apreciar el programa realizado en el Anexo 2.

#### ⊗ **Archivo M107**

Este archivo está ubicado en la ruta de búsqueda especificada en la configuración del archivo ini y está escrito en el lenguaje de programación de Python y, siendo ejecutado cuando un comando M104 aparece dentro del archivo de código G en ejecución. Lo que hace el programa del archivo es crear un archivo de texto con el nombre “M107” con el texto .^pagar ventilador :z el valor de 0 para apagar los ventiladores. Esto debido a que el programa donde se desarrolla la comunicación serial con el microprocesador (encargado de hacer el control de las temperaturas y ventiladores) envía el valor de 0 con el que se quiere setear los ventiladores al microprocesador solo cuando el archivo descrito anteriormente existe y luego de enviarlo por la comunicación serial, elimina dicho archivo para de esta manera no volver a enviar el mismo valor y realizar la acción cada vez que aparezca ese comando M dentro del código G. Para mayor comprensión de lo antes mencionado se puede apreciar el programa realizado en el Anexo 3.

#### ⊗ **Archivo M199**

Este archivo está ubicado en la ruta de búsqueda especificada en la configuración del archivo ini y está escrito en el lenguaje de programación de Python y, siendo ejecutado cuando un comando M199 aparece dentro del archivo de código G en ejecución. Como se había mencionado con anterioridad, este es un archivo necesario para realizar la conexión con la base de datos. Inicialmente se adquiere la fecha y hora exacta en la que se ejecutó el programa, posterior a esto obtiene el nombre del archivo “ngc” actualmente en ejecución y se conecta con la base de datos. Luego verifica que exista una tabla de datos dentro de la base con el mes y año respectivo, de no ser así la crea inmediatamente y luego agrega a la tabla del mes y año correspondiente el nombre del archivo y la fecha. Finalmente sierra la comunicación con la base de datos y termina de ejecutarse el programa, lo que hace a Linuxcnc continuar con su ejecución del archivo “ngc”. Para mayor comprensión de lo antes mencionado se puede apreciar el programa realizado en el Anexo 7 .Además, más adelante se ha realizado un apartado netamente para hablar a detalle de este código.

- **Traductor de G code a extensión ngc.**

El intérprete de código g que utiliza LinuxCNC es el RS274/NGC, el cual es el lenguaje básico basado en líneas de códigos llamados bloques, que son instrucciones que se envían para controlar sistemas CNC. Se presentaron algunos inconvenientes con la compatibilidad de los archivos que se generan en los softwares slicer para impresión 3D; por lo que es necesario hacer algunas modificaciones.

En primer lugar, el intérprete RS274/NGC solo permite leer archivos de código G con extensión .ngc y el archivo generado por la mayoría de los slicer tienen extensión .gcode. LinuxCNC no cuenta con una configuración para impresión 3D, por lo que no encontramos códigos G ni códigos M dentro de LinuxCNC que nos permitan controlar parámetros de impresión 3D, tales como la temperatura de la cama caliente, hotend y ventiladores. Estos comando si poseen los archivos .gcode dados por el slicer; anteriormente se mencionó la necesidad crear códigos compatibles con estas acciones, pero además es necesario comentar los códigos M que no ocuparemos en la configuración de nuestra impresora.

Para la denominación del comando del extrusor los slicer ocupan la letra “E” con el cual envían el avance del motor a pasos del extrusor, mientras que LinuxCNC solo permite controlar motores con los comandos de ejes los cuales son X, Y, Z,

A, B, C, U, V y W. Además, el comando para enviar datos a los códigos M en LinuxCNC es la letra “P”, mientras que en el slicer es la letra “S”.

Para corregir la compatibilidad de los 2 archivos se realizó la programación de un traductor de archivos .gcode; se utilizó como lenguaje de programación Python, el cual es compatible con LinuxCNC. Dentro de las funciones de este programa se encuentran crear un nuevo archivo con la extensión .ngc para que sea compatible con el intérprete RS274/NGC, en el cual se cambian las letras del control del extrusor de la letra “E” a la letra del eje escogido “A” y de la letra de datos pasar de “S” a “P” y comentar los códigos M mayores a 100 los cuales no utilizaremos con la excepción de 104, 106, 107, 109 , 140 y 190; que son los códigos M de los parámetros de impresión 3D. Cabe mencionar que el cambio de estas letras se realiza sin ocasionar cambios en los comentarios y demás partes del archivo; en el anexo 11 se puede observar el diagrama de flujo del traductor.

#### **6.4. Diseño de control de temperaturas que intervienen en la Impresión 3D**

La temperatura a la cual el hotend debe mantenerse es de gran importancia, puesto que de esta depende que el material con el que se quiere realizar la impresión 3D se pueda fundir adecuadamente, la mayoría de los fabricantes de los materiales de impresión informan al usuario los rangos óptimos para la impresión, ya que si la temperatura es menor a la sugerida, el material puede no depositarse adecuadamente al no haber alcanzado un valor óptimo de fusión.

Por otro lado, si la temperatura del mismo sobrepasa la sugerida, entonces el material se empezará a quemar y quedarse impregnado en el hotend lo que causará una futura obstrucción del mismo. La importancia del control también se debe a que en un cierto valor dentro del rango sugerido por el fabricante se obtendrá mejores resultados de impresión que en otros, por lo que aun dentro de ese rango se debe hacer un control de temperatura para mantenerlo lo más cercano al valor deseado.

El diseño propuesto para el control del hotend se validó mediante pruebas de funcionamiento.

En cuanto al control de la cama caliente, se debe mencionar que aunque no es tan preciso como el del hotend, se debe mantener la temperatura de la cama a un valor deseado ya que esto permite que el filamento pueda adherirse y mantenerse adherido durante toda la impresión, caso contrario, se nos puede despegar el objeto impreso durante el proceso.

#### 6.4.1. Aproximación del modelo del termistor

Existen diferentes elementos que nos permiten realizar el sensado de la temperatura de los distintos componentes de una impresora 3D, por lo general tanto la cama caliente como el hotend tienen incluido un termistor que permite sensar la temperatura de estos elementos. Para el presente proyecto se ha seleccionado tanto un hotend como una cama caliente que como elemento de medición de temperatura, vienen incorporado un termistor NTC de 100Kohms.

Un termistor es una resistencia sensible a la temperatura los cuales son fabricados con semiconductores. Los termistores NTC muestran un decrecimiento en la resistencia al incrementar la temperatura. Esta relación entre la resistencia  $R_x$  en función de la temperatura  $X$  puede ser descrita mediante la siguiente función. (Quintero, 2003)

$$R(x) = R_0 e^{\beta \left( \frac{1}{x} - \frac{1}{x_0} \right)} \quad (12)$$

Donde :

$R_0$  es la resistencia a 298 ° Kelvin del termistor

$\beta$  es el coeficiente de relación en ° Kelvin Dado que, tanto el hotend como la cama caliente tienen termistores de 100kohms, se puede utilizar los valores ideales que describen el comportamiento de estos termistores, los cuales son un  $\beta$  igual a 3799.42 ° Kelvin,  $x_0$  igual a 298° Kelvin y  $R_0$  igual a 100000.00 ohms, estos datos fueron adquiridos de la pagina web de Stanford Research Systems (2021) la cual nos deja obtener además la figura 22 de la curva la ecuación del termistor es la resistencia en relación a la temperatura.

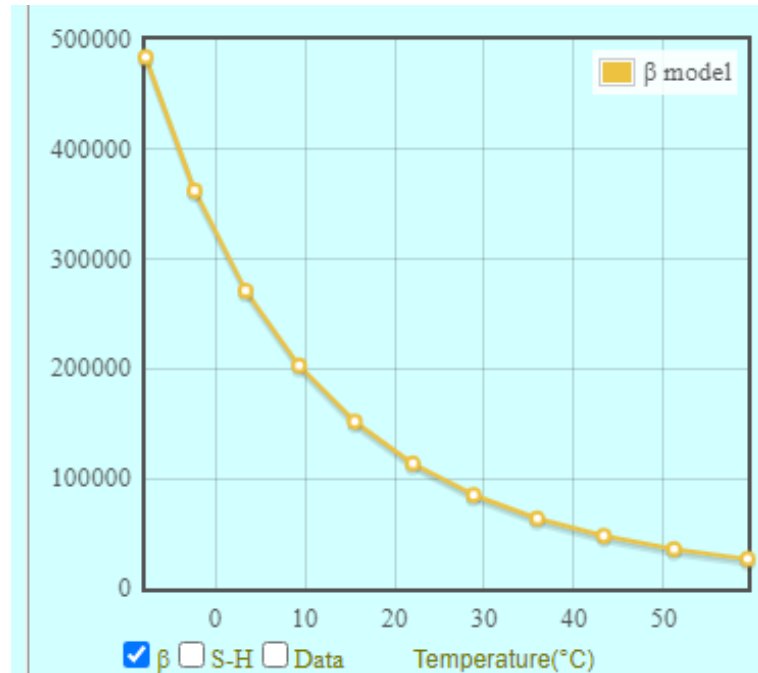
Entonces remplazando estos valores de  $R_0$  y  $\beta$  en la ecuación 12, nos queda:

$$R(x) = 100000 e^{3799,42 \left( \frac{1}{x} - \frac{1}{298} \right)} \quad (13)$$

Además, como se puede apreciar la función es exponencial decreciente por lo que para



**Figura 22:** Curva de la ecuación característica de un termistor NTC 100K.



Fuente: Stanford Research Systems (2021)

obtener una mejor aproximación a los valores de temperaturas reales, se ve conveniente linealizar la ecuación.

Quintero (2003) nos dice que:

"Linealizar el sistema consiste en establecer una metodología para diseñar el sistema de tal manera que la salida sea lineal."

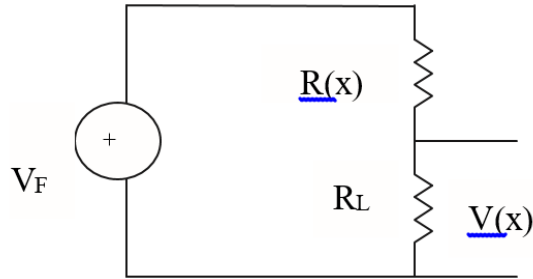
Por esta razón y para obtener los valores de nuestra resistencia del termistor se plantea un circuito divisor de tensión como el mostrado en la figura 23 ,el cual nos permite linealizar nuestra salida y obtener así también en función del voltaje la resistencia en el termistor.

Donde:

$R_{(x)}$  es la resistencia del termistor.

$R_L$  es la resistencia de linealización.

**Figura 23:** Esquema del Divisor de tensión



**Fuente:** Quintero (2003)

Es importante mencionar que la resistencia de linealización nos permitirá realizar la linealización de la salida entre cierto rango de valores por lo que para calcularla nos basamos en la siguiente ecuación:

$$R_L = \frac{R_a * (R_b - R_j) - pR_j * (R_b - R_a)}{p * (R_b - R_a) - R_b + R_j} \quad (14)$$

Donde :

$R_a$  es la resistencia a la temperatura inicial de linealización.

$R_b$  la resistencia a la temperatura final de linealización.

$R_j$  la resistencia entre cierto valor entre el rango de linealización.

Además el valor de  $p$  se encuentra mediante la ecuación mencionada a continuación:

$$p = \frac{b - x_j}{b - a} \quad (15)$$

Siendo:

$x_j$  la temperatura entre cierto valor entre el rango de linealización.

$a$  la temperatura inicial de linealización.

$b$  la temperatura final de linealización.

Con las ecuaciones 14 y 15 podemos calcular la resistencia  $R_L$  del divisor de voltaje del termistor del hotend. Sabiendo que la mayoría de los filamentos del material que se pretende utilice la impresora van entre un rango de 190°C a 230°C se selecciona un valor de  $x_j$  igual a 210°C, dado que se encuentra dentro del rango establecido.

Para proceder con el cálculo es necesario tener el valor de la resistencia del termistor NTC 100K a dichas temperaturas, para lo que utilizamos la pagina web de Stanford Research Systems (2021) que en base a la curva característica nos da los valores de resistencias siendo : 1067.47ohms, 556.06ohms y 760.91ohms respectivamente. Sustituyendo las temperaturas de 190 igual a  $a$ , 230 igual a  $b$  y 210  $x_j$  en la ecuación 15 obtenemos un valor de  $p=0.5$ , validando en función de estos valores la ecuación 14 obtenemos:

$$R_L=4.3\text{Kohms.}$$

Al no existir un valor comercial de resistencia se seleccionó una de 4.7Kohms que si es comercial y es cercana al valor calculado .

Por otro lado, para encontrar el valor de  $R_L$  del divisor de tensión con el termistor de la cama caliente, se realiza el mismo calculo para un rango de temperatura de 50 a 100° C, con los siguientes datos:

$$a=50.$$

$$b=100.$$

$$x_j=75.$$

$$R_a =37311.57.$$

$$R_b=7720.44.$$

$$R_j=16039.$$

$$p=0.5.$$

Remplazando los valores en la ecuación 14 obtenemos:

$$R_L=47.92\text{Kohms}$$

Selecionando una resistencia comercial cercana de 47Kohms.

#### **6.4.2. Obtención de los controladores de temperaturas**

Linuxcnc al ser un software especializado en el corte laser, es el encargado de realizar el control de movimiento de los motores para cada uno de los ejes así como del motor para

el extrusor, no obstante, no presenta las herramientas necesarias para realizar el control de temperatura, por esta razón se optó por diseñar un circuito con un microprocesador ATMEGA 328p como elemento principal que será el encargado de adquirir los datos de la temperaturas mendicantes sus ADC y realizar el control de las mismas utilizando sus salidas PWM y salidas digitales .

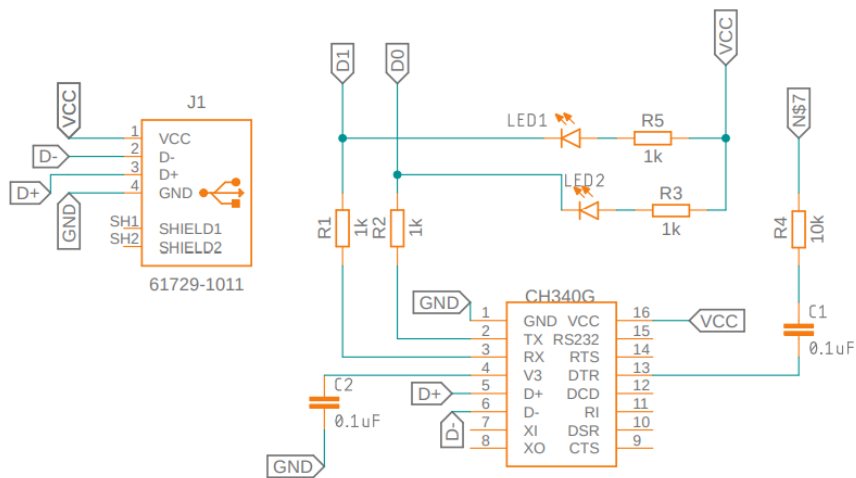
■ **Circuitos de los controladores**

⊗ **Circuito USB B**

En la figura 24 podemos ver el circuito que permite al microcontrolador comunicarse seriamente mediante USB tipo B con la tarjeta madre de la cortadora laser DGL6060 , además es mediante el mismo circuito que se alimenta a la placa del microcontrolador mediante los 5 voltios. Adicionalmente tenemos unos leds que nos permiten ver el envío y recepción de datos mediante la comunicación serial al microcontrolador.

**Figura 24:** Circuito propuesto de conexión USB tipo B.

### Circuito USB B



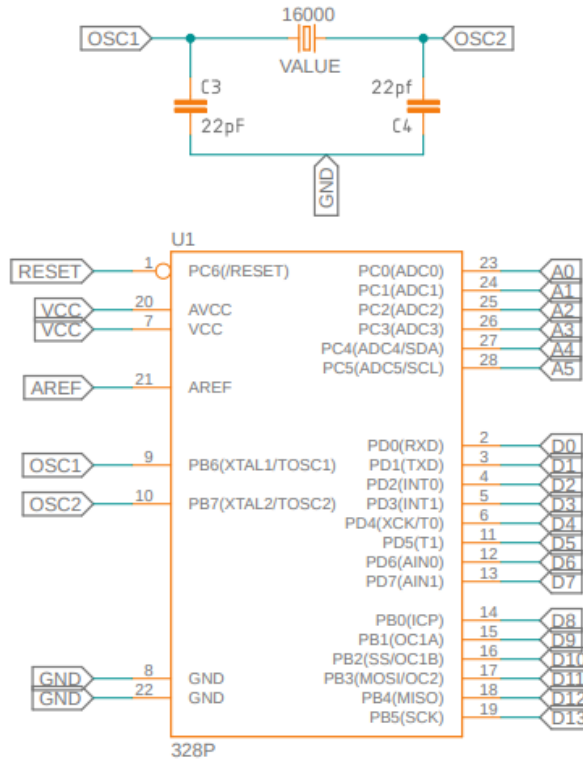
**Fuente: Autores**

⊗ **Circuito Atmega 328p y Clock**

En la figura 25 podemos ver el circuito del clock que necesita el microprocesador para funcionar, así también las respectivas líneas de conexión de cada pin del mismo.

**Figura 25:** Circuito propuesto de conexión Atmega328p y reloj.

## Circuito Atmega 328p y clock



Fuente: Autores

### ⊗ Circuitos Entrada de sensores

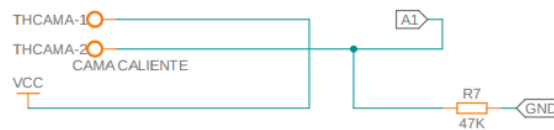
En la figura 26 podemos ver los circuitos divisores de tensión usados para adquirir el valor de resistencia de los termistores, como se mencionó anteriormente estos no solo sirven para obtener las resistencias del termistor sino también calcular el valor de temperatura de manera linealizada del hotend y cama cliente. Además, se deja incorporando en la placa la posibilidad de incorporar otro termistor para mediante el poder realizar en control de otra temperatura, ya sea que se desee incorporar la extrusión doble o el control de la temperatura ambiente en lo posterior.

**Figura 26:** Circuitos de entradas de sensores.

### Circuito Entrada Sensor Hottend



### Circuito Entrada Sensor Cama Caliente



### Circuito Entrada Sensor Extra



**Fuente: Autores**

## ⊗ Circuitos PWM de control de temperatura hotend y ventiladores

En la figura 27 se puede ver los diferentes circuitos mediante los cuales, utilizando las salidas de PWM (Pulse-Width Modulation) del microcontrolador, se realiza el control de la temperatura del hotend y también el seteo de las velocidades de los ventiladores. La modulación por ancho de pulso es una modulación donde se mantiene la frecuencia de la señal de salida constante (es decir, entre disparo de pulsos) mientras lo que varía es el ancho del pulso. El tiempo que la señal permanece encendida respecto al total del ciclo se denomina “ciclo de trabajo” y generalmente es expresada en porcentaje. (Luis Llamas, 2015)

La forma en que se realiza el control de temperatura del hotend se da mediante un controlador PID en lazo cerrado mediante el termistor, en un capítulo más adelante se hablara más a detalle de esto, pero básicamente, para el control se tendrá una salida de PWM que será el encargado de regular la potencia que se envía al bloque calefactor, el valor de PWM será mínimo cuando la temperatura se encuentre lejos de

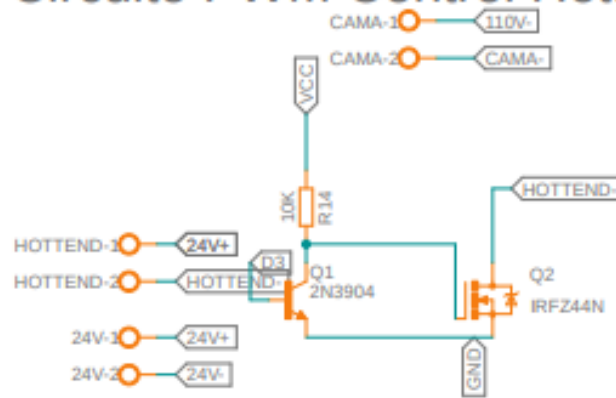
la deseada esto hará que la potencia que se entrega al hotend por medio del circuito sea la máxima y cuando la temperatura se va acercando, el valor del PWM disminuye lo que hace que se alimente con una potencia menor y de esta manera se mantendrá el control. El PWM controla la potencia mediante la variación del voltaje que se aplica al bloque calefactor, es importante mencionar que el máximo voltaje que tiene las salidas PWM del microcontrolador son de 5 voltios DC(Corriente Directa), por lo que se tiene el circuito mostrado en la figura 27 que mediante un mosfet y un transistor.

El ancho de pulso de 5 voltios sirve como señal para enviar 24 voltios DC cuando el ancho del pulso sea alto. Además, este mismo circuito sirve para proteger al microcontrolador de que se pueda filtrar a él un voltaje de 24 voltios DC lo quemaría al mismo.

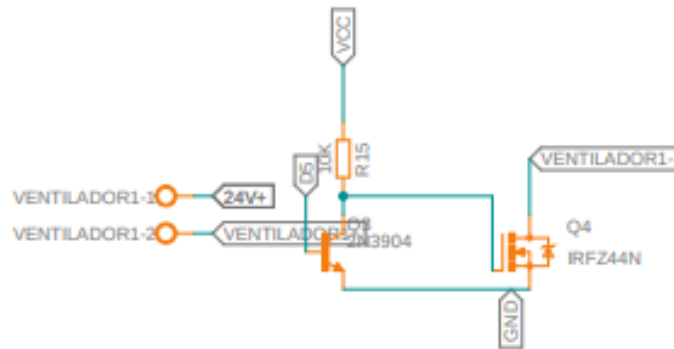
Para los ventiladores se tiene el mismo circuito pero no cuentan con retroalimentación al no ser de suma importancia, por lo que los valores de PWM serán constantes dependiendo de la programación.

**Figura 27:** Circuitos de control mediante PWM

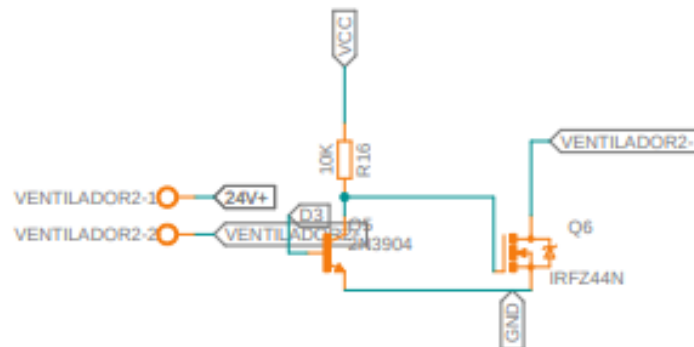
### Circuito PWM Control Hottend



### Circuito PWM Control Ventilador 1



### Circuito PWM Control Ventilador 2



Fuente:Autores

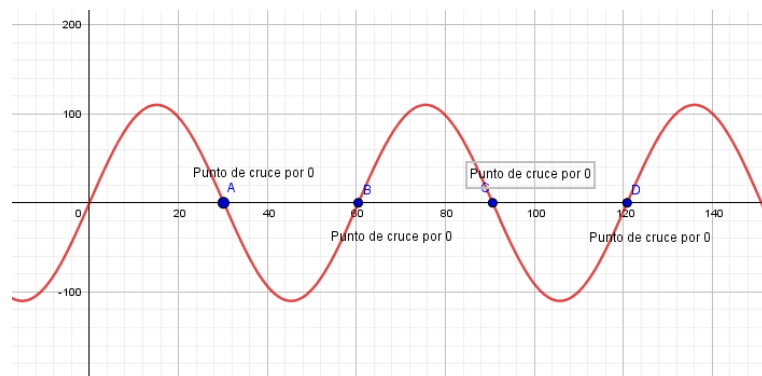


## ⊗ Circuitos para el control de temperatura de la cama caliente

Aunque el principio para el control que se propone para la temperatura de la cama caliente es el mismo que el del hotend, es decir utilizar un control PID digital con la señal de salida y un termistor como retroalimentación, los circuitos que se deben emplear para esta son diferentes. Principalmente porque a diferencia del hotend la cama caliente funciona con 110 voltios AC (Corriente Alterna), por lo que necesita un circuito que permita detectar el cruce por 0 de la señal AC y de igual manera un circuito que mediante la señal de salida 5 voltios DC podamos controlar la potencia de la señal AC que alimenta la cama caliente.

El circuito que se propone usar es aquel que nos permita detectar el cruce por 0 de la señal de 110 voltios AC, dado que como se observa en la figura 28 la señal periódicamente tiene cruces por el valor de 0 voltios, es necesario detectar el momento en que la onda pasa de un valor positivo a negativo o viceversa, dado que el detectar este cruce por 0 nos servirá para enviar en el momento adecuado el disparo del triac que se encuentra en el segundo circuito y también estar sincronizarlos con el controlador PID.

**Figura 28:** Cruces por 0 en la señal AC



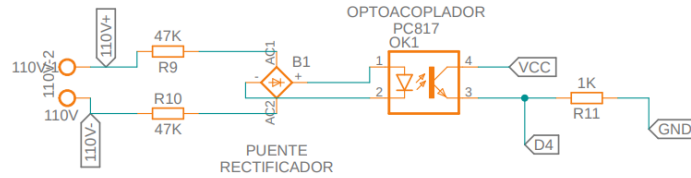
**Fuente:** Autores

Entonces, el circuito de la figura 29 lo que realiza es primero rectificar la señal mediante un puente rectificador de onda completa, esto ya que el microcontrolador no puede recibir señales de voltajes negativos. Además, también se cuenta con un opto acoplador para proteger al microcontrolador de voltajes mayores a 5 voltios que es el máximo voltaje permitido por el mismo, pero también nos sirve para detectar el cruce por 0 ya que el circuito va a enviar al microcontrolador pulsos de 5 voltios cada que la señal

rectificada tenga algún valor de voltaje .por lo que mediante la programación se detecta cuando en este circuito el voltaje es 0 para tener el instante en el que la señal AC tiene el cruce por este valor.

**Figura 29:** Circuito detector de cruce por 0

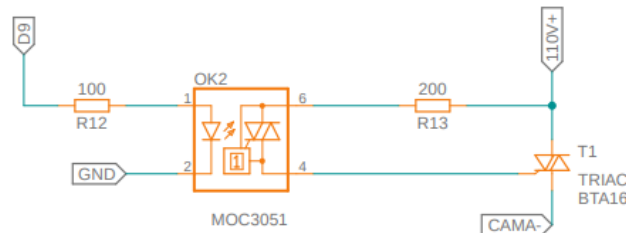
### Circuito Entrada Detector 0 Phase



**Fuente:Autores**

El Segundo circuito propuesto a usar ,el cual se puede ver en la figura 30 ,es el encargado de enviar un pulso de disparo al MOC3051 mediante un pin de salida y a su vez esto, es la señal que activa la puerta del triac.

**Figura 30:** Circuito de control Cama Caliente



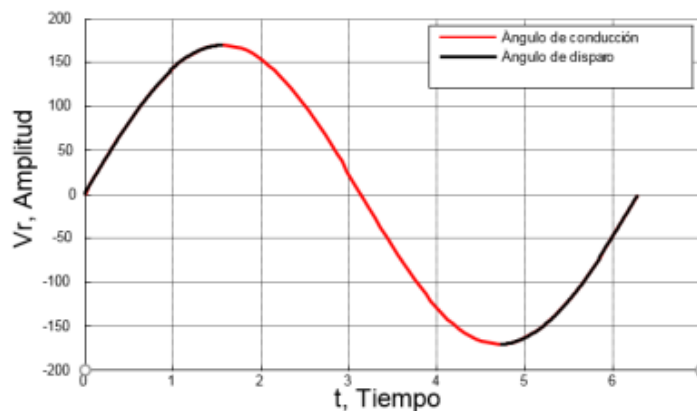
**Fuente:Autores**

Para entender de mejor manera como es que este circuito permite controlar la potencia que se alimenta a la cama caliente mediante un triac partimos de que:

“El triac es un interruptor semiconductor de CA de tres terminales que se activa en conducción cuando se aplica una señal de baja energía a su puerta. El cual conduce corriente en cualquier dirección cuando se enciende. Una señal de puerta positiva o negativa activará al dispositivo en la conducción, además el triac puede ser pensado como dos SCR complementarias en paralelo.“(Cabrera, 2019)

Gracias a su capacidad de poder controlar la activación de este elemento, es un método eficaz y ampliamente utilizado de control de la potencia media a una carga mediante el control de su fase. El control de fase es una forma de utilizar el triac para aplicar el suministro de la corriente alterna a una carga por una fracción controlada de cada ciclo. Cuando se realiza esta configuración, el triac permanece en codician abierta de cada ciclo positivo y negativo, luego mediante el circuito de control se activa en determinado momento del ciclo.

**Figura 31:** Control de fase



**Fuente:**Cabrera (2019)

En la figura 31 se puede apreciar dos ángulos, medidos en grados eléctricos, el de disparo y el de conducción:

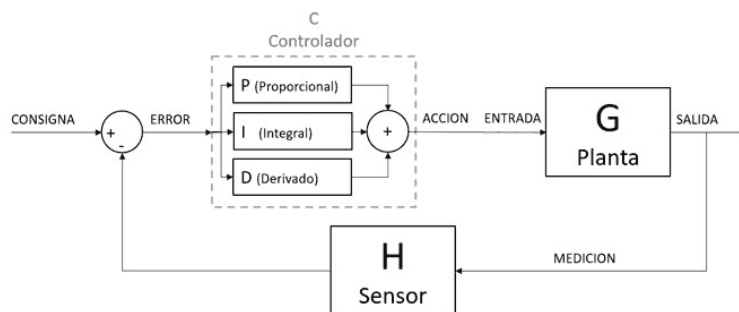
**Ángulo de disparo:** durante el cual el triac está bloqueando la tensión de red y no permite el flujo de corriente eléctrica.

**Ángulo de conducción:** El período durante el cual el triac permite el flujo de corriente.

- **Algoritmo de control**

Para desarrollar el algoritmo de control se debe tener en cuenta la definición de lo que es un controlador PID en tiempo discreto y de la ecuación 6, con el cual, se describe al sistema como lo muestra la figura 6 que es el diagrama de bloques de controlador y planta.

**Figura 32:** Diagrama de bloques del control PID



**Fuente:**Cabrera (2019)

El valor de referencia es adquirido mediante el código G ejecutado o el entorno virtual de la máquina, el cálculo del error lo realiza el algoritmo en el microcontrolador así como el valor de los términos proporcional, integral y derivativo, que mediante las salidas digitales y PWM regulan la potencia tanto del hotend como de la cama caliente que son la planta del sistema.

Inicialmente lo que el algoritmo debe realizar es el cálculo de la resistencia del termistor mediante una de las entradas ADC (Convertidor Análogo-Digital) el cual da un valor de 0 a 1023 (10 bits) dependiendo del voltaje de entrada que como máximo puede ser 5 voltios. Entonces con este valor y el valor de resistencia  $R_L$  de la cama caliente y hotend mencionado en la sección anterior respectivamente, se obtiene el valor de la resistencia del termistor.

Posterior a esto el algoritmo calcula la temperatura en Celsius mediante la ecuación 13 y finalmente se transforma de Kelvin a Celsius.

De esta manera el algoritmo para encontrar la temperatura mediante el termistor es el mostrado en la figura 33.

Cabe mencionar que en la medición del ADC se realiza una toma de 50 muestras en el mismo instante para de esta manera obtener un valor promedio más cercano al valor real, además se usa un método que permite tomar el valor de manera más rápida mediante el cambio del preescaler del timer en el microcontrolador.

En cuanto al algoritmo de control, mediante las ecuaciones 6,3 y 2 se podría generar ya un control básico PID, no obstante, se realizaron ciertas modificaciones para tener un controlador más robusto, teniendo en cuenta los siguientes efectos.

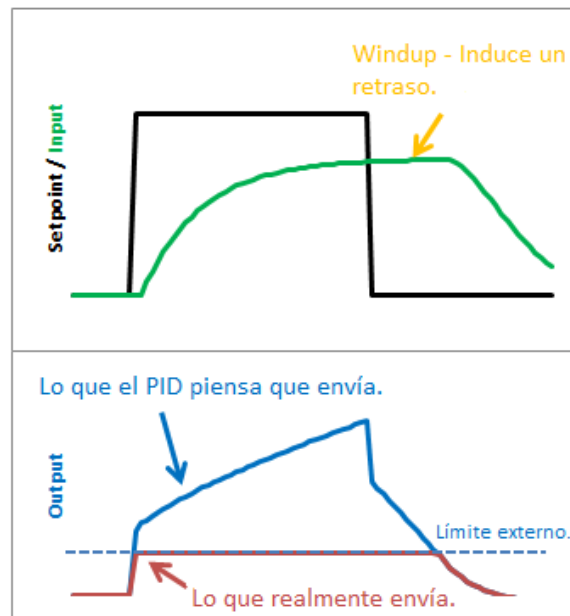
**Figura 33:** Código del algoritmo de temperatura

```
void MedirTemperatura() {  
  float a = 0;  
  for (int i = 0; i < 50; i++) {  
    a = analogReadFast1(termistor) + a; // calcular la resistencia electrica del  
    termistor  
  }  
  //usando la lectura del ADC  
  // cada 50 datos es 2 millis de retardo  
  medida = a / 50;  
  vm = (vcc / (1023 - 1)) * ( medida ); //Calcular tensión en la entrada  
  R1 = rAux * ((5.0 / vm) - 1); //Calcular la resistencia de la  
  NTC  
  temperaturaK = beta / (log(R1 / r0) + (beta / temp0)); //Calcular la temperatura  
  en Kelvin  
  //  
  Leer_temperatura = temperaturaK - 273; //Calcula la temperatura en Celcius  
}
```

**Fuente:Autores**

**Derivative Kick:** Este fenómeno se produce por variaciones rápidas en la señal de referencia las cuales se magnifican por la acción derivativa transformandose en componentes transitorios de gran amplitud en la señal de control. Para solucionar dicho efecto, se puede hacer un ligero cambio al cálculo del termino derivativo.(Beauregard, 2011)

**Figura 34:** Fenómeno Derivative Kick



**Fuente:Beauregard (2011)**

**Windup:** Este efecto se da al iniciar el control o en cualquier otra situación donde aparece un error muy grande durante un tiempo prolongado, esto hace que el termino integral aumente para corregir el error ,pero dado que nuestro microcontrolador tiene

un actuador limitado a 255(valor del PWM),este se saturara pero el termino integral seguirá aumentando. Pero, cuando el error se reduce, la parte integral también comenzará a reducirse, pero desde un valor muy alto, llevando mucho tiempo hasta que logre la estabilidad, generando fluctuaciones exageradamente grandes.(Beauregard, 2011)

**Figura 35:** Código algoritmo de control PID hotend

```
void CalculoPID() {
  now = millis();
  MedirTemperatura();
  Input = Leer_temperatura;
  //Calculamos todos los errores.
  error = Setpoint - Input;
  //P=kp*error;
  double dInput = (Input - lastInput);
  // codigo de restriccion de la constante I para
  // evitar su comportamiento al inicio del programa
  if (dInput==0) {
    ITerm += (ki * error);
  }
  else{
    ITerm -= (ki * dInput);
  }
  /-----
  // codigo para limitar la salida hasta 255 que es el rango pwm
  if (ITerm > outMax) {
    ITerm = outMax;
  }
  else if (ITerm < outMin) {
    ITerm = outMin;
  }
  Output = (kp * error) + ITerm - (kd * dInput);
  if (Output > outMax) {
    Output = outMax;
  }
  else if (Output < outMin) {
    Output = outMin;
  }
  /-----
  // Actualizamos valores y mandamos la señal pwm
  lastInput =Leer_temperatura;
  lastTime = now;
  analogWrite(PWM_hottend, 255 - Output); // salida del pwm del hottend
}
```

**Fuente:Autores**

Esto genera retrasos extraños como lo muestra la figura(citar) y para solucionarlo lo primero es limitar el valor del termino integral de 0 a 255 que es el rango en que trabajan

nuestras salidas PWM y de igual manera el valor de nuestra salida final, es decir la suma del termino proporcional, integral y derivativo. (Beauregard, 2011)

Cabe mencionar que el valor de PWM del microcontrolador va entre rangos de 0 a 255, un valor de 0 genera 0 voltios y el de 255 genera 5 voltios, pero cuando el valor está entre 0 a 255, el pin alterna rápidamente entre 0 y 5 Voltios, cuanto mayor sea el valor, más a menudo el pin se encuentra en “Alto” es decir su voltaje maximo. para explicar esto de mejor manera, se pone el siguiente ejemplo: un valor de 64 será 0 voltios tres cuartas partes del tiempo y 5 una cuarta parte. (dfists, 2020)

Entonces el algoritmo de control PID para la temperatura del hotend es el mostrado en la figura 35.

Es importante mencionar que la salida de PWM es inversa al valor de control del PID debido al diseño de los circuitos y a los transistores usados, por lo que el valor enviado al pin de PWM es 255- el valor de salida del controlador .

Aunque para el control de temperatura de cama caliente se propone el mismo algoritmo de control, existen algunas variaciones en el algoritmo de control de temperatura de la cama caliente.

Esto debido a que primeramente no se usa una señal PWM como salida de control sino una señal digital de 5V para dar el ángulo de disparo al triac, además esta señal estará en alto solo el tiempo que nos del valor de salida del PID en referencia a cuando se ha detectado el cruce por 0 . Es decir, el PID va a tener como límites de 0 a el valor del periodo de la señal .

En la figura 36 se observa el algoritmo de control propuesto para la temperatura de cama caliente.

En ambos casos tanto para control del hotend como el de la cama caliente, se ejecuta periódicamente el algoritmo cada cierto intervalo de tiempo, el controlador del hotend está dado por el tiempo de sintonización adecuado del PID y el otro está dado por el periodo de la señal AC.

- **Sintonizacion de los controladores.**

Como se menciona en la sesión de los algoritmos de control, el controlador PID depende

**Figura 36:** Código algoritmo de control PID cama caliente

```
////////////////////////////////////
////
//----- Funcion del PID de la Cama -----//
////////////////////////////////////
//
void PIDCama(){
  currentMillis = millis(); //guardamos el valor del tiempo antes del
  control
  if(currentMillis - previousMillis >= temp_read_Delay){
    previousMillis += temp_read_Delay; //Incrementamos el tiempo para el
    proximo control
    MedirTemperaturaCama(); // Calculamos la temperatura
    actual
    PID_error = setpoint_Cama - real_temperature; //Calculate the pid ERROR

    if(PID_error > 30) //la integral no afetara al control cuando la
    temperaturas es mayores a 30°C
    (PID_i = 0;)

    PID_p = kpc * PID_error; //Calculamos el valor P
    PID_i = PID_i + (kic * PID_error); //Calculamos el valor I
    timePrev = Time; // the previous time is stored before the
    actual time read
    Time = millis(); // actual time read
    elapsedTime = (Time - timePrev) / 1000;
    PID_d = kdc*(PID_error - previous_error)/elapsedTime; //Calculate the D
    value
    PID_value = PID_p + PID_i + PID_d; //Calculate total PID
    value

    //We define firing delay range between 0 and 7400. Read above why 7400!!!!!!
    if(PID_value < 0)
    { PID_value = 0; }
    if(PID_value > 7400)

    { PID_value = 7400;
    previous_error = PID_error; //Remember to store the previous error.
    }

    //If the zero cross interruption was detected we create the 100us firing pulse
    if (zero_cross_detected)
    {
      delayMicroseconds(maximum_firing_delay - PID_value); //This delay controls
      the power
      digitalWrite(Cama_pin,HIGH);
      delayMicroseconds(100);
      digitalWrite(Cama_pin,LOW);
      zero_cross_detected = false;
    }
  }
}
```

**Fuente:Autores**

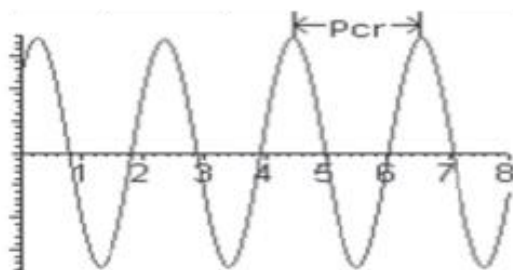
del valor de las constantes  $K_p$ ,  $K_i$  y  $K_d$  y la sintonización de las mismas hará que nuestro controlador se comporte de distintas maneras. Dado que no se conoce la función de transferencia de la planta, la mejor opción para encontrar los valores de sintonización de las constantes es mediante el segundo método de Ziegler Nichols que nos permite obtener el primer acercamiento a dichos valores mediante el valor de la constante  $k_p$  crítica.

Para realizar este método se aplica una señal escalón unitario a la entrada, se realimenta el lazo y se determina el valor de la ganancia ( $K_p$ ) que hace al sistema críticamente estable y se mide el periodo de salida ( $P_{cr}$ ). (Ruge, 2012)

En un sistema críticamente estable existe una oscilación con cierto periodo de salida como lo muestra la figura 37.



**Figura 37:** Señal de salida y periodo crítico



**Fuente:Ruge (2012)**

Una vez encontrado los valores de  $K_p$  crítico ( $K_{cr}$ ) y Periodo crítico ( $P_{cr}$ ) el método brinda la tabla 5 para encontrar los valores respectivos de  $K_p$ ,  $T_i$  y  $T_d$ .

**Tabla 5:** Cálculo de parametros mediante Ziegler Nichols

Tipo	$K_p$	$T_i$	$T_d$
<b>P</b>	$0.5K_{cr}$	inf	0
<b>PI</b>	$0.45K_{cr}$	$P_{cr}/1.2$	0
<b>PID</b>	$0,6K_{cr}$	$0,5P_{cr}$	$0.125P_{cr}$

**Fuente:Ruge (2012)**

Ademas, con la utilización de las formulas 2 y 3 se calcula los valores de las otras 2 constantes, es decir  $K_I$  y  $K_D$  mediante la sustitución de los valores de  $T_i$  y  $T_d$  en dichas ecuaciones. Una vez calculadas estas constantes va a depender de los parámetros de control el variar estos valores para acercarnos a las necesidades del control.

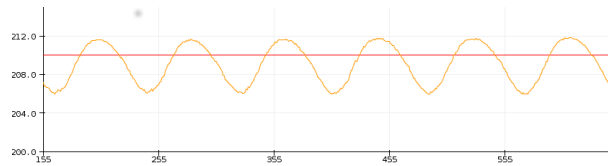
Como se mencionó anteriormente, la sintonización del controlador PID de la cama caliente mediante este método queda planteada para su futura realización.

Pero la sintonización del controlador PID del hotend se realizó de manera experimental, teniendo como parámetros de control el obtener una respuesta sobre amortiguada con un error en estado estable de  $\pm 2^\circ\text{C}$  y con un tiempo de estabilización no mayor a los 4 minutos dado que estos son los estándares de las impresoras 3D comerciales.

Entonces, se encuentra  $k_p$  crítico mediante la toma de muestras con el microcontrolador, dándole un escalón de 24 voltios al sistema, cerrando el lazo mediante el termistor y con el uso del algoritmo de control solo con el termino proporcional. Como se aprecia en la figura 38, con un valor de 32 en el  $K_p$  se encontró una respuesta críticamente

estable y el periodo de la misma es de 26.25 segundos dado que tiempo de muestreo es de 350 milisegundos.

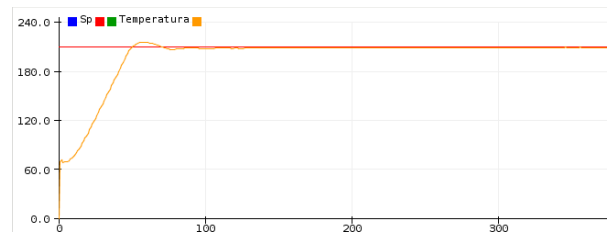
**Figura 38:** Señal críticamente estable.



**Fuente:Autores**

Con los datos mencionados y mediante el uso de la tabla 5 y las ecuaciones 2 y 3 se obtuvo como resultado los valores de:  $K_p=19,2$   $K_i=1.46$   $K_d=62.97$  como valores iniciales, los cuales al ser implementados en algoritmo de control podemos observar en la figura 39 el comportamiento de todo el sistema, el cual tiene un sobrepaso máximo del 3 % (218°C), un error en estado estable de  $\pm 3^\circ\text{C}$  y un tiempo de estabilización de 2.8 minutos.

**Figura 39:** Respuesta del sistema del hotend con constantes de  $K_p=19,2$   $K_i=1.46$   $K_d=62.97$ .

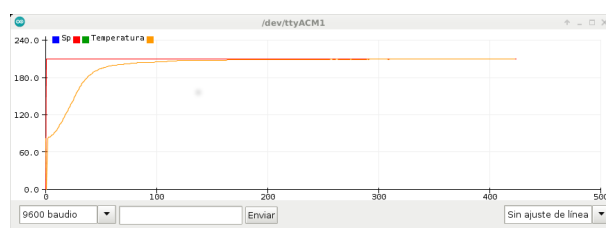


**Fuente:Autores**

Al existe un sobrepaso, esto incumple un parámetro de control, además de que el error tampoco es el deseado. Por lo que este resultado no es satisfactorio y se realizó algunas iteraciones mediante la variación de  $k_p$ ,  $k_i$  y  $k_d$ .

Las constantes finales implementadas en el algoritmo de control de temperatura de hotend son:  $k_p=6.5$   $k_i=2.3$   $k_d=62.57$ . Con lo cual, los resultados finales se observan en la figura 40. En dicha figura se puede ver un comportamiento sobre amortiguado con un tiempo de estabilización de aproximadamente 2 minutos y con un error en estado estable de  $\pm 1^\circ\text{C}$ , con lo cual se cumplen los objetivos de control.

**Figura 40:** Respuesta del sistema del hotend con constantes de  $k_p=6.5$   $k_i=2.3$   $k_d=62.57$  .



**Fuente:Autores**

## ■ Comunicación serial

Como bien se expuso en los apartados anteriores el control de las temperaturas se da en una placa diseñada específicamente para dicha tarea y separada de la tarjeta madre, por lo que al ser la tarjeta madre quien mediante linuxcnc va adquiriendo los valores de temperatura y ventiladores y siendo Linuxcnc quien durante la ejecución de ciertos códigos M espera la verificación de estos valores.

Es necesaria una comunicación entre estas 2 placas, viendo como la más adecuada la realización de una comunicación serial mediante 2 algoritmos, el primero programado en el microcontrolador y el segundo realizado en el programa principal de ejecución del entorno gráfico de la máquina.

En la figura 41 se presenta el diagrama de flujos para el desarrollo de cada uno de estos algoritmos, pero de manera general. Los dos algoritmos empiezan su funcionamiento comprobando que se encuentren comunicados mediante el envío de el carácter "Q" por parte de la tarjeta madre y devolviendo la cadena "listo" el microcontrolador, sin esta verificación el entorno gráfico no se procede a abrir. Posteriormente se genera un hilo dentro del programa que se ejecuta en la tarjeta madre donde constantemente esta verificándose la existencia de alguno de los archivos M104.txt, M140.txt, M106.txt o M107.txt, de existir uno de estos archivos, se envía el carácter respectivo al microcontrolador y el valor que se encuentre dentro de estos archivos. Finalmente borra el archivo de texto encontrado para de esta manera realizar este proceso una sola vez cada que se encuentre estos archivos.

El microcontrolador por su parte analiza el tipo de carácter que le enviaron y según eso cambia el valor ya sea de las variables de referencia de temperatura del hotend, como

caliente o el pwm de los ventiladores.

Una vez realizada esta acción, el microcontrolador si se cambio un valor de temperatura espera a que los valores reales sean iguales o los cambiados y envia un carácter de confirmación al programa de la tarjeta madre. Si al programa de la tarjeta madre recibe la confirmación antes mencionada entoces genera un archivo de texto M190 o M109 dependiendo de si es confirmación del hotend o de la cama caliente. Además, el mismo programa cada 35 segundos envia el carácter “R” al microcontrolador para que le devuelva el valor actual de las temperaturas, esto gracias a que el entorno grafico constantemente esta mostrando estos valores en su entorno.

## **6.5. Diseñar un sistema dual y una interfaz IOT para inventario remoto; incorporando los software y controladores pertinentes del corte laser e impresión 3D.**

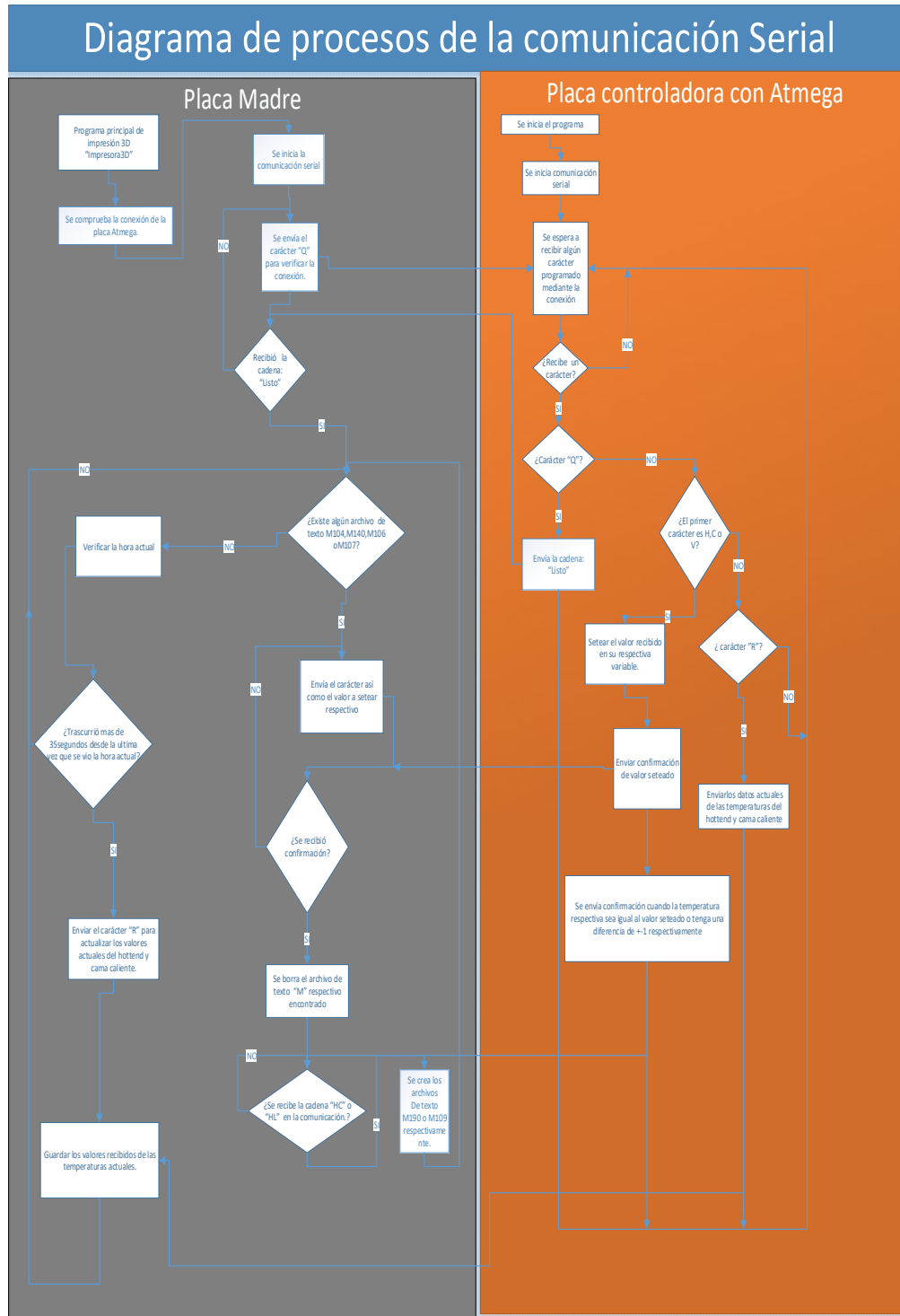
### **6.5.1. Diseño de la interfaz gráfica para la pantalla**

Para permitir que la Cortadora laser DGL-6060 pueda contar con un sistema dual, se debe crear un entorno grafico que permita la selección de la operación que se desea realizar, ya sea corte y grabado laser o impresión 3D, ya que cada una de estas funciones tiene sus archivos y configuraciones específicas para su correcto funcionamiento con Linuxcnc pertinentes. Además, Linuxcnc no puede tener 2 configuraciones de sus parámetros activos a la vez (archivos ini y hal) por lo que para poder cumplir con el objetivo del sistema dual debe existir una manera de cambiar las configuraciones, así como poder entrar y cerrar los entornos gráficos respectivos de cada función. Es por esta razón que se crea el entorno grafico principal dando al usuario la posibilidad de seleccionar la función de impresión 3D o corte laser y posterior a esto cargar las configuraciones y ejecutar en entorno grafico respectivo.

- **Archivo Glade**

Glade es una herramienta que permite crear y editar diseños de interfaces de usuario para aplicaciones GTK+. Las bibliotecas GTK+ proporcionan una extensa

**Figura 41:** Diagrama de flujos del proceso de la comunicación serial.



Fuente: Autores

colección de bloques de construcción de interfaz de usuario tales como cajas, etiquetas de diálogo, entradas numéricas, casillas de selección y menús. Los interfaces de usuario diseñados en Glade se almacenan en un formato XML, permitiendo una integración fácil con herramientas externas, permitiendo usar Glade para añadir conexiones entre widgets y el código fuente de una aplicación.

Para el sistema dual de la maquina DGL-6060 se diseño una interfaz gráfica que permita seleccionar el modo de trabajo, entre el corte a laser y la impresión 3D; se tomo como referencia la interfaz que ya tenia la maquina para corte a laser y se diseño una compatible con el GUI de la configuración de LinuxCNC. Al hacer clic en alguna de las 2 opciones se ejecuta el archivo .ini de las configuraciones de la maquina en LinuxCNC

Al seleccionar el botón de impresión se abre la interfaz GUI; esta es la pantalla principal del proceso de impresión 3D, como se puede observar se cuenta en la parte superior con pestañas que nos llevan a las diferentes ventanas de la interfaz. También se cuenta con visualización de la posición de los ejes, que nos muestra en tiempo real el desplazamiento tanto en coordenadas relativas como absolutas y en la parte inferior se encuentra el monitoreo de los variables de impresión como son las temperaturas de la cama caliente y el hotend, la velocidad de los ventiladores, la comprobación de la comunicación serial y el avance de la impresión en porcentaje. Se tiene 2 botoneras principales que están fijadas en la interfaz las cuales controlan la maquina entre los botones que tenemos están:

- ⊗ Paro de emergencia
- ⊗ Encendido de la maquina
- ⊗ Mover a 0 Maquina
- ⊗ Iniciar: es el botón que da la orden de empezar a imprimir
- ⊗ Pausa: detiene el programa y puede volver a empezar en la línea de código en la que se detuvo
- ⊗ Stop: detiene el programa y vuelve al inicio del código.

En segunda ventana se muestra la interfaz de traducción en la cual se carga el archivo .gcode generado en los software slicer, el botón traducir ejecuta la función del traductor, creando el nuevo archivo ngc.

Luego de traducir el archivo se pasa a la ventana que carga el código para la

ejecución de la impresión.

También se tiene una ventana para la configuración manual del usuario, en la cual se puede mover los ejes de forma manual y modificar la ubicación del 0 Pieza, además de setear valores de los ventiladores, y las temperaturas de cama caliente y hotend.

- **Archivo Python**

Los Archivos glade son entornos gráficos que necesitan de un archivo en lenguaje de Python para poder programar las funciones pertinentes a cada elemento y a su vez son desde estos archivos donde se ejecutan las ventanas creadas mediante glade.

- ⊗ **Ventana de selección de funciones.**

Este es el programa desde donde se da inicio a todo el sistema dual, encargado de dar función a los botones de impresión 3D, el cual al ser clickeado ejecuta el archivo ".ini" el cual configura al Linuxcnc y procede a abrir el entorno virtual ejecutando el programa donde se encuentran programadas las funciones del glade. Dicho ejecutable se menciona en el apartado siguiente. Por otro lado también se programa el botón de Corte laser el cual dará a la maquina las configuraciones pertinentes a esta operación y a su vez ejecutara el entorno grafico respectivo.

- ⊗ **Archivo Ventana principal de impresión 3D**

Este archivo es el encargado de dar funcionalidad a todos los elementos dentro del entorno grafico creado para la impresión 3D. Además se puede considerar el programa principal dado que en el mismo se encuentra programado todo del proceso de la comunicación serial, así como ,mediante una de las funciones del mismo se invoca al archivo encargado de la traducción de "gcode" a "ngc".

### **6.5.2. Diseño de la plataforma iot**

Debido a que la empresa requiere que la maquina pueda llevar un inventario de los trabajos que va realizando, la solución que se plantea es la utilización de una base

de datos en la nube con la cual mediante un programa la maquina pueda guardar la información necesaria la misma.

Para ello se utilizó MySQL, el cual nos permite mediante una dirección IP crear nuestra propia base de datos en la nube para su posterior administración.

MySQL es un servicio de base de datos totalmente administrado que permite a los desarrolladores desarrollar e implementar de una manera rápida aplicaciones nativas en la nube seguras utilizando código abierto, cuenta con un motor de análisis en tiempo real integrado y escalable de forma masiva. (MySQL, 2021)

Además, MySQL permite al desarrollador crear sus propios entornos dentro de la nube para administrar de esta manera sus bases de datos, no obstante, en este proyecto se usó un entorno llamado phpMyAdmin.

phpMyAdmin es una herramienta gratuita escrito en lenguaje PHP, se encuentra destinada a manejar y administrar MySQL a través de la web. Admite una amplia gama de operaciones en MySQL y MariaDB. A través de una interfaz de usuario se pueden realizar operaciones como: administración de bases de datos, tablas, columnas, relaciones, índices, usuarios y permisos. No obstante, también tiene la capacidad de ejecutar los mismos procesos mediante declaraciones. (phpMyAdmin, 2021)

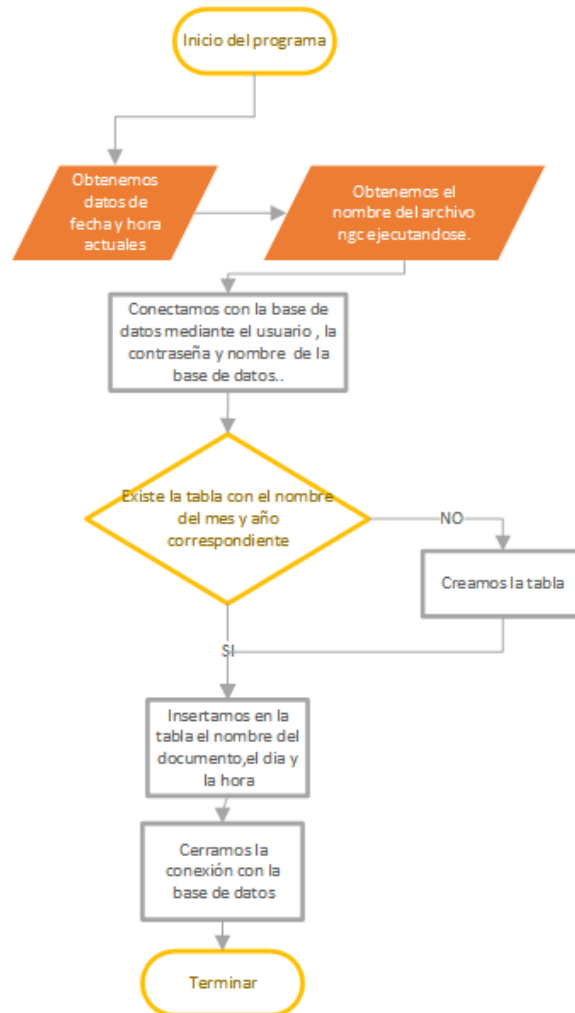
Una vez creada la base de datos en MySQL, se debe crear un usuario que administre toda la base de datos y es la cual maneja el dueño de la empresa y un usuario para la máquina que solo le permita acceder a las bases de datos y agregar información a las tablas mas no eliminarlas.

El programa desarrollado en Python con el nombre de M199, es el encargado de conectarse a la base de datos en la nube, adquirir los datos actuales de la fecha (día, mes, año) y posteriormente el nombre del archivo de código G que se está ejecutando en ese instante. Entonces el programa, procede a buscar la tabla correspondiente al mes y año, de no existir la crea. En esta tabla guarda en la fila los datos de nombre del archivo, día, hora y automáticamente también se graba la numeración correspondiente. Finalmente, el programa cierra su conexión con la base de datos y termina su ejecución.

El diagrama de flujo de este programa puede ser observado en la figura 42 la cual explica con detalles el algoritmo.



**Figura 42:** Diagrama de flujos del proceso del programa M199.



**Fuente: Autores**

Como se pudo observar el programa tiene como nombre "M199" ya que se utiliza como un código M programado por el usuario, de esta manera Linuxcnc ejecutara este programa cuando encuentre el código M mencionado dentro de sus líneas al realizar un archivo ".ngc". Pero al ser un código personalizado el programa encargado de traducir de gcode a ngc debe introducir esta línea de comando al final antes de la línea de comando "M02", para que así la maquina guarde esta información en la base de datos solo si se ha terminado de realizar la impresión.

## **7. Resultados**

### **7.1. Resultado de la identificación de las características mecánicas y electrónicas de la cortadora laser DGL-6060 para implementar un extrusor para impresión 3D.**

Como se mencionó anteriormente; se procedio a la identificación las partes mecánicas y eléctricas que ya posee actualmente la maquina DGL-6060, y se puedo comprobar su compatibilidad para un sistema dual con impresión 3D. el sistema mecánico en los 3 ejes cumple con las necesidades básicas para la realización de la impresión; mientras que, las targetas controladoras Mesa cuentan con una gran versatilidad para comandar las entradas y salidas del sistema, no es necesario modificar este esquema de control ya que cuenta con los puertos suficientes para añadir el motor a pasos del Hotend.

### **7.2. Resultado de la adaptación del diseño mecánico y electrónico de la cortadora laser DGL-6060; con un sistema dual que incorpora un extrusor y cama caliente para la impresión 3D.**

Se realizo un diseño para el acople modular que permita intercambiar las piezas del sistema de corte a laser y de impresión 3D de manera rápida y fácil para el operario además de fácil mantenimiento. Los planos del acople se pueden observar en los anexos.

Se seleccionó también el hotend y extrusor en base a las necesidades y características que posee el sistema mecánico de la maquina DGL6060; el hotend seleccionado es el E3D V6 que junto al Extrusor Titan Aero forman un sistema de impresión Bowden adecuado para las necesidades de la maquina.

### **7.3. Resultado del diseño del sistema de control para la impresión 3D, que permita leer los archivos generado, comandando los motores**

Además, se obtuvieron 6 programas nombrados con una “M” y su número respectivo, los cuales unos sirven para obtener los valores de temperaturas desde los comandos del código “ngc” y como bandera para que el programa de la comunicación serial sepa que debe actualizar los valores de los mismos. Por otro lado, otros se encargan de mantener en pausa a Linuxcnc hasta que exista un archivo que haya generado la comunicación serial, donde se confirme que

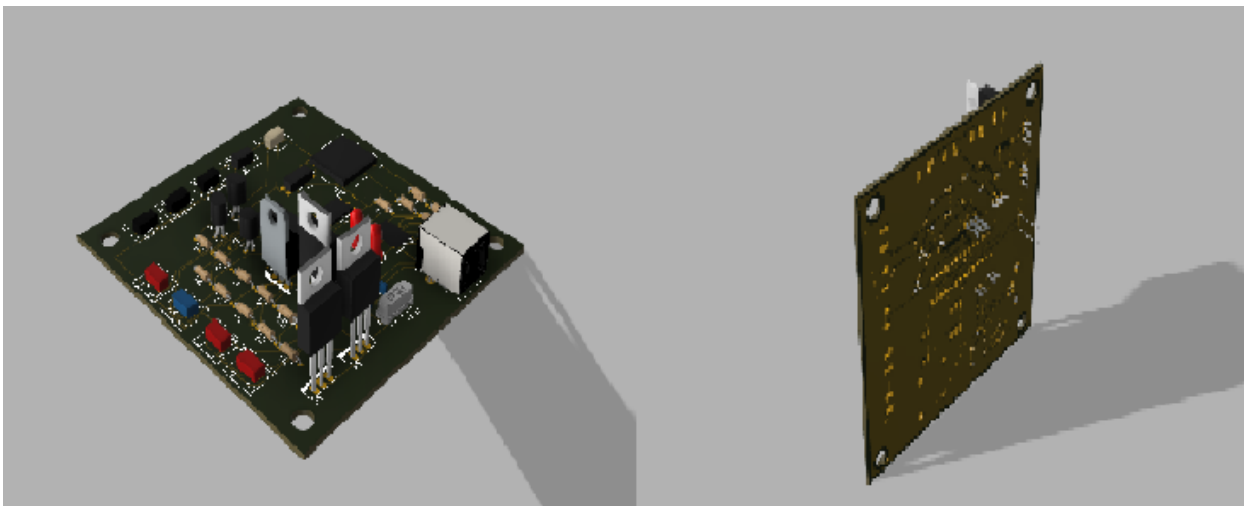
el valor de temperatura que se ha cambiado anteriormente ya sea el real. Los códigos de estos programas se pueden ver desde el anexo 1 hasta el 6.

#### 7.4. Resultado del diseño de control de temperaturas que intervienen en la Impresión 3D.

Dentro de este punto se puede recalcar 2 resultados obtenidos, el primero es el diseño de los circuitos necesarios para realizar los controles de temperaturas, la comunicación serial y los que permitan el correcto funcionamiento del microcontrolador.

Con estos circuitos se diseñó un modelo de placa controladora la cual se puede observar en la figura 43.

**Figura 43:** Modelo de placa controladora



**Fuente:**Autores

Por otro lado, también se desarrolló el algoritmo con el cual se programa en el microcontrolador y es el que permite realizar el control de las temperaturas y realiza una comunicación serial con la tarjeta madre de la máquina, en él se encuentran los cálculos de los PIDs necesarios, la adquisición de los valores de los sensores, los métodos para la obtención de temperatura a partir de los voltajes del A/D, entre otros.

A continuación, en la figura 44 se puede ver el diagrama de flujos del bucle principal de este algoritmo, en el podemos ver que existen métodos de los cuales, el método de

Cserial () es la programación respectiva sobre la comunicación serial lo cual fue explicado anteriormente, existe también el método de Verificarseteo () el cual usa la comunicación serial para notificar la tarjeta madre que los valores seteados son los mismos que el real medido.

Para comprender de mejor manera lo que realiza el método llamado PIDCama (), se puede observar la figura 45, este método se encarga de realizar el control PID de la cama caliente. Además, en la figura 46 podemos ver el diagrama de procesos del método CalculoPID () el cual realiza el cálculo del PID del hotend.

Cabe mencionar que existe también un método para adquirir los valores de temperatura actuales llamados MedirTemperaturaCama () y MedirTemperatura () utilizados para adquirir la temperatura de la cama caliente y el hotend respectivamente, el proceso es el mismo para ambos casos por lo que en la figura 47 se puede ver el diagrama de procesos que tienen estos métodos. El algoritmo completo se encuentra en el anexo 8.

## **7.5. Resultado del diseño sistema dual y una interfaz IOT para inventario remoto; incorporando los software y controladores pertinentes del corte laser e impresión 3D.**

Para el Sistema Dual, se obtuvo como resultado en primera instancia los entornos gráficos necesarios ,al ya existir uno desarrollado para el funcionamiento en el corte laser, primero se creó un entorno grafico para todas las funciones referentes a la impresión 3D , para ello se utilizó glade como la herramienta para crear la ventana y Python para instanciar la ventana y dale función a cada botón y herramienta dentro de esta ventana.

El código de este programa en Python se puede apreciar en el anexo 9, en el mismo también esta incorporado el algoritmo para la comunicación serial que debe existir en la tarjeta madre para comunicarse con el microcontrolador. No obstante también se creó otra ventana que es la que emerge al inicio y permite al usuario la posibilidad de seleccionar las operación a realizar, ya sea impresión 3D o corte laser, de igual manera esta fue desarrollada mediante Glade y programadas sus funciones mediante Python, en el anexo 10 se puede observar el código de este archivo.

En cuanto a la interfaz IOT de inventario remoto se pudo desarrollar un algoritmo que

Figura 44: Diagrama de flujo del bucle principal del algoritmo del microcontrolador

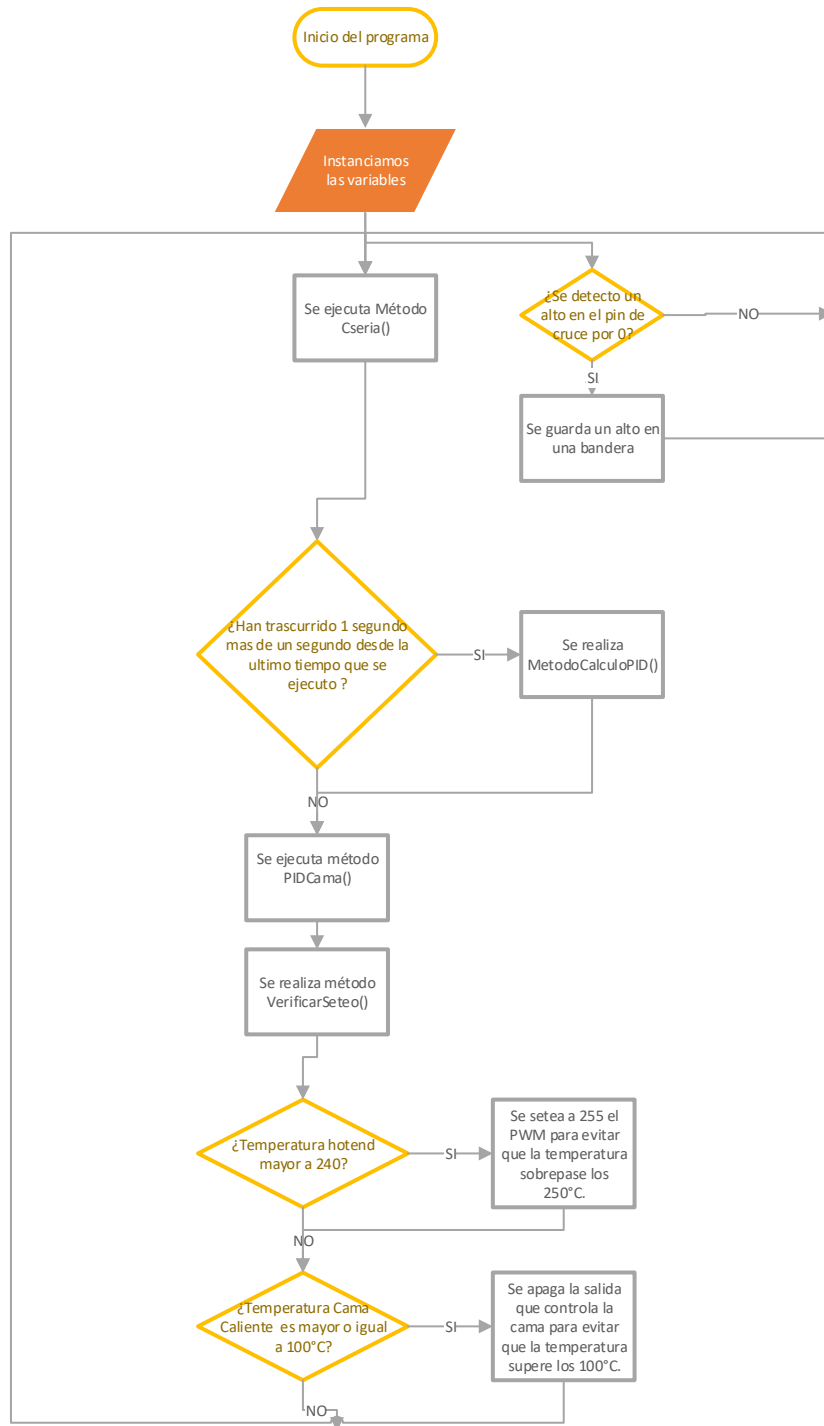
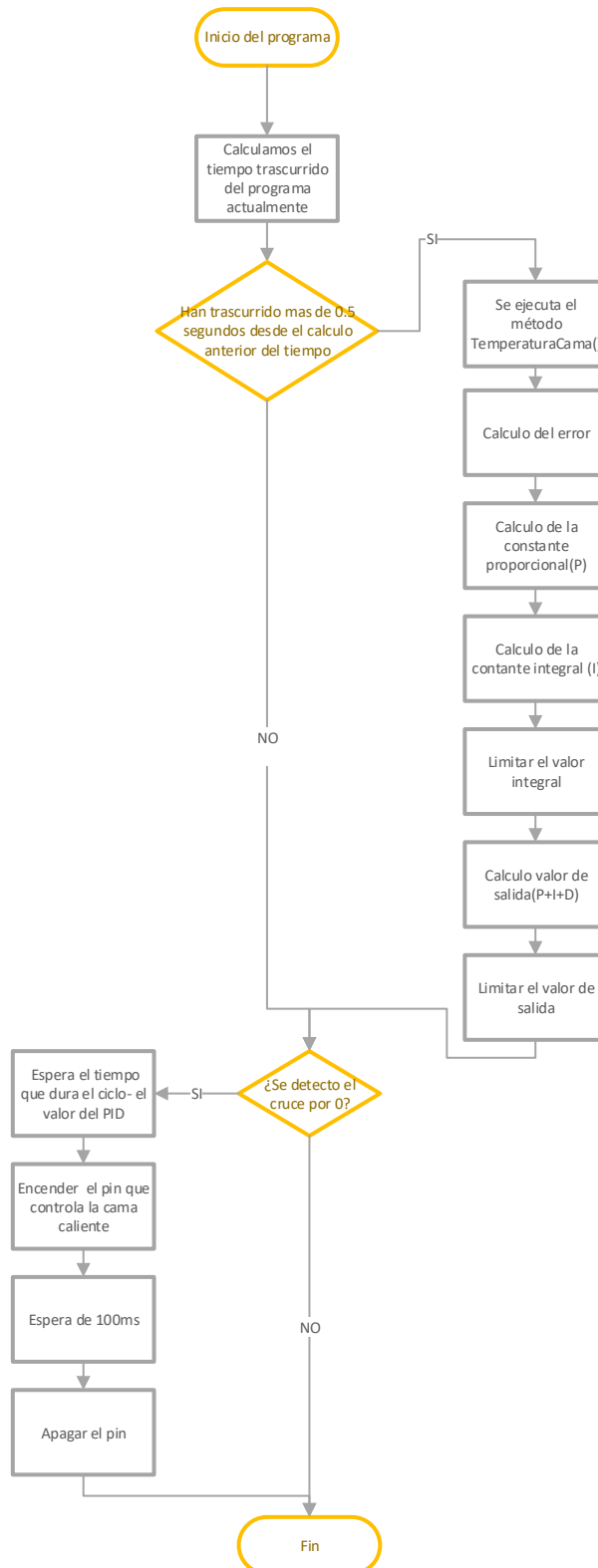
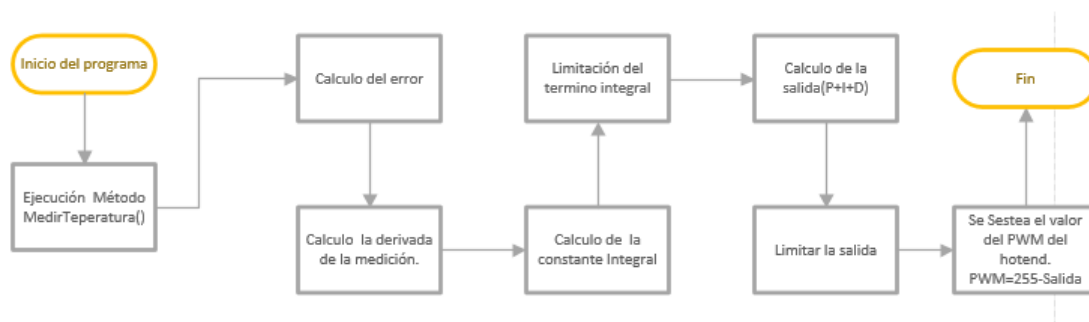


Figura 45: Diagrama de flujo del metodo PIDCama

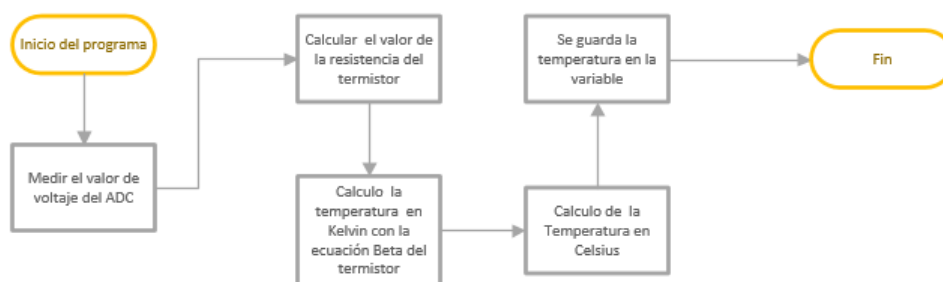


**Figura 46:** Diagrama de procesos del metodo CalcularPID



**Fuente:Autores**

**Figura 47:** Diagrama de procesos del metodo MedirTemperaturaCama y MedirTemperatura



**Fuente:Autores**

mediante un código M personalizado se conecta la maquina a una base de datos en la nube para en ella escribir en tablas la información necesaria sobre cada uno de los códigos que va ejecutando.

Este algoritmo se encuentra en el archivo llamado M199 y es el que se aprecia en el anexo 7.

## 8. Conclusiones

En este proyecto se pudo identificar los parámetros que le han permitido a la cortadora laser DGL-6060, posicionarse en el mercado como una maquina confiable y accesible; gracias a sus componentes mecánicos y electrónicos. Los cuales componen un sistema CNC para el corte laser que, además, presentan características básicas que se pueden adaptar para realizar

el proceso de impresión 3D.

En la parte mecánica la maquina DGL-6060 cuenta con sistemas móviles en los 3 ejes fundamentales para la impresión 3D, no obstante, es necesario establecer un diseño que permita añadir los componentes para realizar la impresión 3D mediante la técnica de fabricación de filamento fundido (FFF). Para esto se diseñó un acople modular donde se colocarán el extrusor de filamento como el hotend de impresión; este acople modular permite al operario intercambiar las piezas de corte a laser e impresión 3D de una manera fácil y rápida. Además de la selección de una cama caliente para el espacio de trabajo que es de 600x600mm, la cual por sus dimensiones debe trabajar con corriente alterna.

Para el control de movimiento de los 3 ejes se utilizó la herramienta de LinuxCNC la cual permite mediante el intérprete de código G realizar una planificación de trayectorias y movimientos que mediante un lazo de control cerrado permite enviar los pulsos para los motores a pasos. Para realizar el control de temperatura tanto de la cama caliente como del hotend de impresión, se diseño los circuitos que permiten la adquisición de las señales mediante termistores y realizar el mismo mediante las salidas PWM del microcontrolador, para modificar la potencia suministrada a estos componentes. En cuanto al control, se tuvo como elemento principal un microcontrolador el cual fue programado para que mediante la resistencia del termistor linealizar la variable de temperatura, establecer la comunicación serial y realizar el control mediante la sintonización de PID en tiempo discreto tanto para la cama caliente y hotend. Actualmente LinuxCNC no posee un módulo para impresión 3D por lo que fue necesario programar un traductor de código G, el cual nos permite traducir el código enviado por los softwares slicer de impresión 3D a un código ngc compatible con LinuxCNC.

Para completar el sistema dual es necesario una interfaz grafica que le permita al usurario elegir el modo de trabajo de la maquina y monitorear los parámetros de impresión, es por esto por lo que se desarrollo el diseño de esta mediante la plataforma Glade y programación en Python. Además, este sistema nos permite conectarnos a una base de datos para realizar el inventario remoto de la producción de la maquina mediante IOT.

## 9. Recomendaciones

- Aunque los componentes de CPU y procesador actuales de la cortadora DGL-6060 cumplen con las características necesarias para establecer el sistema dual, se recomienda una actualización de los mismo para evitar la discontinuación de la maquina en un



corto tiempo.

- Cabe mencionar que para el proyecto se optó por un servidor de base de datos gratuita, que brindaba las características necesarias para probar el sistema de comunicación IOT; se recomienda que, para su uso comercial, La empresa Dumenlaser SAS. Adquiera los servicios de un servidor privado, ya que esto permitirá crear mas usuarios y generar mas funciones dentro de la base de datos.
- Se recomienda que en futuras investigaciones se realice un control de la temperatura del ambiente cerrado para mejorar la calidad de la pieza de impresión, ya que la maquina DGL-6060 cuenta con un área de trabajo cerrada.

## Referencias

- A. Daneri, P. (2009). *PLC: automatización y control industrial*. Editorial Hispano Americana HASA.
- Arriaga Segundo, L. (2004). *Reconversion de las maquinas-herramienta convencionales a Control Numerico Computarizado (Metodologia)*. Instituto Politecnico Nacional.
- Bachs, L. (2009). *Aplicaciones industriales del laser*. Marcombo.
- Banco de Ideas (2012). Dumen Laser.
- Beauregard, B. (2011). Arduino PID - Guía de uso de la librería. pp. 1–28.
- Berchon, M. y Luyt, B. (2016). *La impresion 3D: guia definitiva para makers, disenadores, estudiantes, profesionales, artistas y manitas en general*. Editorial Gustavo Gili.
- Bordignon, F., Iglesias, A. A., y Hahn, Á. (2018). DISEÑO E IMPRESIÓN DE OBJETOS 3D Una guía de apoyo a escuelas. Technical report.
- Budynas, K. y Nisbett., R. (2014). Diseño en Ingeniería Mecánica de Shigley. *Igarss 2014*, (1).
- Cabrera, J. (2019). Variación Del Ángulo De Disparo De Un Sistema Senoidal Para Desarrollar Un Sistema Eléctrico De Ahorro De Potencia.
- Chapman, S. J. (2012). *MÁQUINAS ELÉCTRICAS 5ta Edición*, volume 5.
- Cobos, C., Conejero, A., Fenollar, O., y Ferrándiz, S. (2019). Influence of the addition of 0.5 and 1 carbon nanotubes (MWCNTs) in poly-lactic acid (PLA) for 3D printing. In *Procedia Manufacturing*, volume 41, pp. 875–881. Elsevier B.V.
- dfists (2020). Entradas y salidas digitales y analógicas.
- García, R. P. Ñ. (2003). Apuntes de sistemas de control. (May):379.
- Gonzalez Contreras, F. y Rosado Castellano, P. (2015). *Control numerico: marco y fundamentos (2a. ed.)*. Editorial de la Universidad Politecnica de Valencia.
- Intel Corporation (2021). Procesador Intel® Core™2 Duo E8400 (caché de 6 M, 3,00 GHz, FSB de 1333 MHz) Especificaciones de productos.

- Javier, F., Somet, P., José, J., Fuentes, M., Leiva, F. G., Ángel, M., y Esteve, P. (2018). *Control de temperatura para sistema de impresión 3D*. PhD tesis.
- Jorquera Ortega, A. (2017). *Fabricacion Digital: introduccion al modelado e impresion 3D*. Ministerio de Educacion de Espana.
- Kuo, B. (1996). *Sistemas de Control Automático*, 7ed, 931p.pdf.
- Leadshine Technology Company (2009). User's Manual For DM556 Fully Digital Stepping Driver. Technical report.
- Linuxcnc (2021a). Códigos M.
- Linuxcnc (2021b). Descripción general del código G.
- LinuxCNC (2021). LinuxCNC V2.8.2-19.
- Lorenzo, J. (2021). Guía de compra extrusores y hotends impresora 3d.
- Luis Llamas (2015). Salidas analógicas PWM en Arduino.
- Mesa Electronics (2020). 7I78 STEP/DIR PLUS I/O DAUGHTERCARD. Technical report.
- Mesa Electronics (2021). 5i25 anything i/o manual.
- MySQL (2021). MySQL :: White Papers.
- Méndez (2015). Dumen Laser – Slogan.
- Ochoa Correa, D. (2013). Control Pid Digital. pp. 76–81.
- OMRON Corporation (2020). Flat Inductive Proximity Sensor TL-W. Technical report.
- phpMyAdmin (2021). phpMyAdmin.
- Quintero, R. (2003). Método para linealizar la salida de un sensor. *Ingeniería (Bogotá)*, 8(1):82–85.
- Ruge, I. (2012). Optimización de señal de control en reguladores PID con arquitectura antireset Wind-Up. *Revista Tecnura*, 15(30):24.
- Stanford Research Systems (2021). SRS Thermistor Calculator.

Weber, R. H. (2010). Internet of Things – New security and privacy challenges. *Computer Law Security Review*, 26(1):23–30.

Younext (2020). Férulas 3D personalizadas: Younext y su tecnología diferenciadora - Younext Healthcare, Innovación en calidad de vida.

CONFIDENTIAL

## ANEXOS

## Anexo 1: Código Archivo M104

```
#!/usr/bin/env python

import sys, time, os
from os import system

P,Q = map(lambda x: float(x), sys.argv[1:])
#creamos el documento
Archivo=open( "/home/dumen/linuxcnc/nc_files/M104.txt", "w")
Texto=["Temperatura hottend a setear:", str(P),]
Archivo.writelines("\n".join(Texto))
Archivo.close()
```

CONFIDENTIAL

## Anexo 2: Código Archivo M106

```
#!/usr/bin/env python

import sys, time, os
from os import system

P,Q = map(lambda x: float(x), sys.argv[1:])
#creamos el documento
Archivo=open( "/home/dumen/linuxcnc/nc_files/M106.txt", "w")
Texto=["Setear ventilador (0-255):", str(P)]
Archivo.writelines("\n".join(Texto))
Archivo.close()
```

CONFIDENTIAL

### Anexo 3: Código Archivo M107

```
#!/usr/bin/env python

import sys, time, os
from os import system

P,Q = map(lambda x: float(x), sys.argv[1:])
#creamos el documento
Archivo=open( "/home/dumen/linuxcnc/nc_files/M107.txt", "w")
Texto=["Apagar ventilador :", str(P)]
Archivo.writelines("\n".join(Texto))
Archivo.close()
```

CONFIDENTIAL



## Anexo 4: Código Archivo M109

```
#!/usr/bin/env python

import sys, time, os
from os import system
import linuxcnc

#verificamos que se a creado el doc que se crea cuando la temperatura del
hottend es la seteada
UM109="/home/dumen/linuxcnc/nc_files/M109.txt"
Verificacion=os.path.isfile(UM109)
a=0
while Verificacion==False:
    linuxcnc.AUTO_PAUSE
    Verificacion=os.path.isfile(UM109)
    if Verificacion==True:
        linuxcnc.AUTO_RESUME
        os.remove(UM109)
        a=1
if Verificacion==True and a==0:
    os.remove(UM109)
```

## Anexo 5: Código Archivo M140

```
#!/usr/bin/env python

import sys, time, os
from os import system

P,Q = map(float, sys.argv[1:])
#creamos el documento
Archivo=open( "/home/dumen/linuxcnc/nc_files/M140.txt", "w")
Texto=["Temperatura de Cama Caliente a setear: ", str(P)]
Archivo.writelines("\n".join(Texto))
Archivo.close()
```

CONFIDENTIAL

## Anexo 6: Código Archivo M190

```
#!/usr/bin/env python

import sys, time, os
from os import system
import linuxcnc

#verificamos que se a creado el doc que se crea cuando la temperatura del
hottend es la seteada
UM190="/home/dumen/linuxcnc/nc_files/M190.txt"
Verificacion=os.path.isfile(UM190)
a=0
while Verificacion==False:
    linuxcnc.AUTO_PAUSE
    Verificacion=os.path.isfile(UM190)
    if Verificacion==True:
        linuxcnc.AUTO_RESUME
        os.remove(UM190)
        a=1
if Verificacion==True and a==0:
    os.remove(UM190)
```

## Anexo 7: Código Archivo M199

```
#!/usr/bin/env python

# -*- coding: utf-8 -*-

import mysql.connector as sql
from datetime import datetime

import linuxcnc

def ObtenerMes():
    global mes
    if (now.month==1):
        mes="Enero"
    elif (now.month==2):
        mes="Febrero"
    elif (now.month==3):
        mes="Marzo"
    elif (now.month==4):
        mes="Abril"
    elif (now.month==5):
        mes="Mayo"
    elif (now.month==6):
        mes="Junio"
    elif (now.month==7):
        mes="Julio"
    elif (now.month==8):
        mes="Agosto"
    elif (now.month==9):
        mes="Septiembre"
    elif (now.month==10):
        mes="Octubre"
    elif (now.month==11):
        mes="Noviembre"
    elif (now.month==12):
        mes="Diciembre"
```

```

def ConsultarTabla(mes,a):
    cursor=db.cursor()#
    tabla=mes+str(a)
    accion="SELECT * FROM {}".format(tabla)
    cursor.execute(accion)
    datos=cursor.fetchall()#obtiene todos los datos
    cursor.close()#serramos el cursor
    #db.close()#serramos la coneccion
    #return datos

def InsertarEnTabla(nombre,mes,a,dia,hora):
    cursor=db.cursor()
    tabla=mes+str(a)
    """ INSERT INTO "NOMBRE DE TABLA"
        (CAMPO1.CAMPO2.. TODOS LOS CAMPOS)
        VALUES
        (VALUE1,VALUE2,VALUE_N) # CAMPOS AUTONUMERICOS NO SE PONEN
    """
    Isql='''INSERT INTO {}(Documento,Dia,Hora)
        Values('{}','{}','{}')'''.format(tabla,str(nombre),str(dia),hora)
    cursor.execute(Isql)#seleccionar de la tabla
    #n=cursor.rowcount()#contar columnas afectadas
    db.commit()# para hacer los cambios
    cursor.close()#serramos el cursor

def CrearTabla(mes,a):
    print("no existe tabla")
    cursor=db.cursor()
    tabla=mes+str(a)
    Ctabla='''CREATE TABLE {}(
    NArchivo int (255) NOT NULL auto_increment,
    Documento VARCHAR(50) NOT NULL,
    Dia VARCHAR(50) NOT NULL,
    Hora VARCHAR(50) NOT NULL,
    PRIMARY KEY(NArchivo)
    )
    '''.format(tabla)
    cursor.execute(Ctabla)#seleccionar de la tabla
    #n=cursor.rowcount()#contar columnas afectadas
    db.commit()# para hacer los cambios
    cursor.close()#serramos el cursor

```

```

#####Inicio del programa #####

now = datetime.now()#obtenemos fecha y hora actuales
year=now.year
mes=""
dia=now.day
hora=str(now.hour)+"h:"+str(now.minute)+"m:"+str(now.second)+"s"
s=linuxcnc.stat()
s.poll()
DireccionArchivo=s.file
c=DireccionArchivo.rfind("/")
NombreArchivo=str(DireccionArchivo[c+1:])
ObtenerMes()

#####
## Conctar con la base de datos ##
#####
db=sql.connect(host="sql10.freesqldatabase.com",
               user="sql10415416",
               password="LAafExirUf",
               database="sql10415416")

try:
    ConsultarTabla(mes,year)
except:
    CrearTabla(mes,year)
InsertarEnTabla(NombreArchivo,mes,year,dia,hora)
db.close()#cerramos la coneccion

```

## Anexo 8: Algoritmo del microcontrolador

```
/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////
//////          PROGRAMA DE ATMEGA328P
PARA          ////
////          EL CONTROL DE DE TEMPERATURA Y VENTILADORES DE
LA IMPRESORA 3D          ////
/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////

//===== Pins
Utilizados=====//
int PWM_hottend = 3;//Pin PWM hottend
int termistor = A0;//pin adc thermistor hottend

int PWM_ventilador1 = 6;//Pin PWM
int PWM_ventilador2 = 9;//Pin PWM

int Cama_pin = 5;//Pin Cama Caliente
int zero_cross = 4;// Pin de cruce cero cama
int termistorC = A1;//pin adc thermistor hottend
//=====
=====//

//=====Variables
ventiladores=====//
int pwm=0;// variable usada para el pwm de los ventiladores
//=====
=====//

//=====Variables sensor
Temperatura=====//
float Setpoint=23;
float Leer_temperatura =0.0;
float medida= 0;
//=====
=====//

//=====Variables de la ecuacion Beta del termistor
Hottend=====//
float rAux = 4700;
float vcc = 5.0;
```

```

float beta = 3799.42;
float temp0 = 298.0;
float r0 = 100000.0;
float vm = 0.0;
float R1 = 0.0;
float temperaturaK = 0.0;
//=====
=====//

//=====Variables utilizadas en el controlador PID
Hottend=====//
unsigned long lastTime=0, now=0;
double Input, Output;
double ITerm, lastInput;
double kp, ki, kd;
unsigned long SampleTime = 1000; //Tiempo de muestreo 1segundo.
1000
double outMin, outMax;
double I, D, P;
double error;
//=====
=====//

//=====Variables utilizadas en 0 phase de la Cama
caliente=====//
//Variables Cama caliente
int last_CH1_state = 0; // guarda el ultimo cambio de estado
detectado
bool zero_cross_detected = false; // variable deteccion del
cruce por 0 de la corriente
////////////////////////////////////
int maximum_firing_delay = 7400;
//Later in the code you will see that the maximum delay after
the zero detection
is 7400. Why? Well, we know that the 220V AC voltage has a
frequency of around 50-60HZ so
the period is between 20ms and 16ms, depending on the
country. We control the firing
delay each half period so each 10ms or 8 ms. To amke sure we
wont pass thsoe 10ms, I've made tests
and the 7400us or 7.4ms was a good value. Measure your

```



```

frequency and change that value later */
//=====
=====//

//=====Variables sensor Temperatura Cama
Caliente=====//
float medidaC = 0;
int real_temperature = 0.0;
int setpoint_Cama = 23;
//=====
=====//

//=====Variables de la ecuacion Beta del termistor Cama
Caliente=====//
float rAuxC = 4700;
float betaC = 3799.42;
float vmC = 0.0;
float R1C = 0.0;
float temperaturaKCama = 0.0;

//=====
=====//

//=====Variables de control PID de la Cama
caliente=====//
unsigned long previousMillis = 0; //tiempo anterior
unsigned long currentMillis = 0; //tiempo actual
int temp_read_Delay = 500; //tiempo de sampleo
float PID_error = 0;
float previous_error = 0; //error anterior
float elapsedTime, Time, timePrev;
int PID_value = 0;
//PID constants
int kpc = 0; int kic = 0; int kdc = 0;
int PID_p = 0; int PID_i = 0; int PID_d = 0;
//=====
=====//

//=====Variables confirmacion de valor seteado de la
temperatura respectiva=====//
int confH[2] = {0, 0}; // Variable confirmacion de seteo hottend,

```

```

confH[0] es seteada cuando se da un cambio en la temp seteada
//          confH[1] se setea cuando luego de
setearse confH[0] se estabiliza nuevamente la temperatura
hottend
int confC[2]={0,0}; // Variable confirmacion de seteo cama ,
confC[0] es seteada cuando se da un cambio en la temp seteada
//          confC[1] se setea cuando luego de
setearse confC[0] se estabiliza nuevamente la temperatura
cama
float VTH[4]; // variable usada para verificar el seteo de
hottend
int contTH=0; // contador de VTH
float VTC[4]; // variable usada para verificar el seteo de la
cama
int contTC=0; // contador de VTC
//=====
=====//

////////////////////////////////////
////////////////////////////////////
////
METODOS (FUNCIONES)                                     ////
////          UTILIZADOS EN EL
PROGRAMA          ////
////////////////////////////////////
////////////////////////////////////

////////////////////////////////////
////////////////////////////////////
//=====          Funcion del PID del
Hottend          =====//
////////////////////////////////////
////////////////////////////////////

void CalculoPID() {
    now = millis();
    MedirTemperatura();
    Input = Leer_temperatura;
    //Calculamostodosloserrores.
    error = Setpoint - Input;
    //P=kp*error;
    double dInput = (Input - lastInput); //derivada de la

```

```

medición
// codigo de restriccion de la constante I para
// evitar su comprtamiento al inicio del programa
if (dInput==0){
    ITerm += (ki * error);
}
else{
    ITerm -= (ki * dInput);
}

//-----
// codigo para limitar la salida hasta 255 que es el rango
pwm
if (ITerm > outMax) {
    ITerm = outMax;
}
else if (ITerm < outMin) {
    ITerm = outMin;
}
Output = (kp * error) + ITerm - (kd * dInput);
if (Output > outMax) {
    Output = outMax;
}
else if (Output < outMin)
    Output = outMin;
}

//-----

// Actualizamos valores y mandamos la señal pwm
lastInput =Leer_temperatura;
nowTime = now;
analogWrite(PWM_hottend, 255 - Output);// salida del pwm
del hottend
}

//=====
=====//

////////////////////////////////////

```

```

////////////////////////////////////
//=====          Funcion  Seterar KD,KIyKP
Hotend          =====//
////////////////////////////////////
////////////////////////////////////
void SetK(double Kp, double Ki, double Kd) {
    double SampleTimeInSec = ((double)SampleTime) / 1000;
    kp = Kp;
    ki = Ki * SampleTimeInSec;
    kd = Kd / SampleTimeInSec;
}
//=====
=====//

////////////////////////////////////
////////////////////////////////////
//=====          Funcion  Seterar KD,KIyKP
Cama          =====//
////////////////////////////////////
////////////////////////////////////
void SetKC(double Kp, double Ki, double Kd) {
    double SampleTimeInSec = ((double)temp_read_Delay) / 1000;
    kpc = kpc;
    kic = kic * SampleTimeInSec;
    kdc = kdc / SampleTimeInSec;
}
//=====
=====//

////////////////////////////////////
////////////////////////////////////
//=====          Setear limites de la salida
iniciales          =====//
////////////////////////////////////
////////////////////////////////////
void SetSalidalimites(double Min, double Max) {
    if (Min > Max) {
        return;
    }
    outMin = Min;
}

```

```

outMax = Max;
if (Output > outMax) {
    Output = outMax;
}
else if (Output < outMin) {
    Output = outMin;
}
if (ITerm > outMax) {
    ITerm = outMax;
}
else if (ITerm < outMin) {
    ITerm = outMin;
}
}
//=====
=====//

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//===== Cambio de la velocidad de muestreo del adc TH
hotten =====//
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
int analogReadFast1(int ADCpin) {
    // if defined( __AVR_ATmega328P__ ) {
    byte ADCregOriginal = ADCSRA;
    ADCSRA = (ADCSRA & B11111000) | 5; // 5 32 prescaler
    int adc = analogRead(ADCpin);
    ADCSRA = ADCregOriginal;
    return adc;
}
//=====
=====//

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//Calculo de la temperatura del hottend mediante la ecuacion
de beta del termistor //
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

```

```

void MedirTemperatura() {
    float a = 0;
    for (int i = 0; i < 50; i++) {
        a = analogReadFast1(termistor) + a; // calcular la
resistencia electrica del termistor
//usando la lectura del
ADC
    } // cada 50 datos es 2
millis de retardo
    medida = a / 50;
    vm = (vcc / (1023 - 1)) * ( medida ); //Calcular
tensión en la entrada
    R1 = rAux * ((5.0 / vm) - 1); //Calcular la
resistencia de la NTC
    temperaturaK = beta / (log(R1 / r0) + (beta / temp0));
//Calcular la temperatura en Kelvin
//
    Leer_temperatura = temperaturaK - 273; //Calcular la
temperatura en Celcius
}

//=====
=====//

////////////////////////////////////
////////////////////////////////////
//Calculo de la temperatura de la cama mediante la ecuacion
de beta del termistor //
////////////////////////////////////
////////////////////////////////////

void MedirTemperaturaCama() {
    float a = 0;
    for (int i = 0; i < 50; i++) {
        a = analogRead(termistorC) + a; // calcular la resistencia
electrica del termistor
//usando la lectura del ADC
    } // cada 50 datos es 2 millis
    medidaC = a / 50;
    vmC = (vcc / (1023 - 1)) * ( medidaC ); //Calcular
tensión en la entrada

```

```

    R1C = rAuxC * ((5.0 / vmC) - 1); //Calcular
la resistencia de la NTC
    temperaturaKCama = betaC / (log(R1 / r0) + (betaC /
temp0)); //Calcular la temperatura en Kelvin
    real_temperature= temperaturaKCama - 273;//Calcula la
temperatura en Celcius
}

//=====
=====//

//////////////////////////////////////
//////////////////////////////////////
//===== Funcion de prender y apagar
Ventiladores =====//
//////////////////////////////////////
//////////////////////////////////////

void Ventiladores() {
    analogWrite(PWM_ventilador1,pwm);
    analogWrite(PWM_ventilador2,pwm);
}

//=====
=====//

//////////////////////////////////////
//////////////////////////////////////
//===== Funcion Comunicacion serial Atmega con
python =====//
//////////////////////////////////////
//////////////////////////////////////

void SerialC() {
    //detectamos si existe un dato envio del python al atmega
    if(Serial.available()){
        /*python envia una letra clave antes del numero para asi
poder
    identificar el valor (munero) a que corresponde,por esta
razon al
    string enviado le dividimos en 2 variables,clave y

```

```

valor      */
    String SerialResivido=Serial.readString();
    String Clave=SerialResivido.substring(0,1);
    String Valor=SerialResivido.substring(1);
    /* Cambio de temperatura segun el comando:
    Q-->comprobar seteo serial
    H--> hottend
    C--> cama caliente
    V-->ventiladores
    R--> enviar valores de temperaturas actuales
    S--> creado el codigo m de confirmacion seteo temperatura
hottend
    U--> creado el codigo m de confirmacion seteo temperatura
cama
    */
    if(Clave=="Q"){
        delay(6);
        Serial.println("Listo");
    }
    if(Clave=="H"){
        Setpoint=Valor.toInt();
        Serial.println(Setpoint);
        confH[0]=1; /*variable que habilita la verificacion del
valor seteado del hottend*/
    }
    if(Clave=="C"){
        setpoint_Cama=Valor.toInt();
        Serial.println(setpoint_Cama);
        confC[0]=1; /*variable que habilita la verificacion del
valor seteado del hottend*/
    }
    if(Clave=="V"){
        pwm=Valor.toInt();
        Serial.println(pwm);
        Ventiladores(); /* se setea el pwm respectivo de los
ventiladores*/
    }
    if(Clave=="R"){
        String th=String((int)Input);
        String tc=String(real_temperature);
        String enviar=th+" "+tc;

```



```
Serial.println(Enviar);/* envial al python el valor
actual de las temperaturas*/
```

```
    }
    if(Clave=="S"){
        //se reinicion las variables de confirmacion
        confH[0]=0;
        confH[1]=0;
    }
    if(Clave=="U"){
        //se reinicion las variables de confirmacion
        confC[0]=0;
        confC[1]=0;
    }
}
/* parte del codigo donde luego de ser verificado el valor
seteado respectivo
* (se realiza en la funcion Verificar>Seteo() ) envia un
comando al pyton
para avisarle que ya esta seteada la temperatura
recivida
```

```
comandos:
HL-->Hottend I esto
// CL-->Cama Caliente Lista
*/
//temperatura hottend a alcanzado el setpoint
else if( confH[0]==1 && confH[1]==1){
    Serial.println("HL");
}
//temperatura de la cama a alcanzado el setpoint
else if( confC[0]==1 && confC[1]==1){
    Serial.println("CL");
}
}
```

```
//=====
=====//

////////////////////////////////////
////////////////////////////////////
//===== Verificacion temperatura alcanzo
```

```

setpoint          =====//
////////////////////
////////////////////

voidVerificarSeteo(){
/* se verifica que se pida verificacion de la temperatura
del hottend */
  if(confH[0]==1){
    /* si la temperatura esta entre un rango +-0.7 del
setpoint del hottend ,entoces
    * se empieza a guardad las 4 siguientes temperaturas
del hotend en VTH,una vez
    * guardadas esas temperaturas, se verifica que la
diferencia entre estas 4
    * temperaturas y el set no sea mayor a + 1 grado, esto
porque pudo solo haberse
    * dado un sobrepaso en el control y no realmente se
estabilizo, si esta
    * entre ese rango de 1 grado , entoces se setea
confTH[1] para que el serial pueda
    * enviar la verificacion al python.
    */
    if ((Setpoint-0.7)<Input<(Setpoint+0.7)){
      if (contTH==0){
        VTH[contTH]=Input;
        contTH=1;
      }
      else if (contTH==1){
        VTH[contTH]=Input;
        contTH=2;
      }
      else if (contTH==2){
        VTH[contTH]=Input;
        contTH=3;
      }
      else if (contTH==3){
        VTH[contTH]=Input;
        float prom=(VTH[0]+VTH[1]+VTH[2]+VTH[3])/4;
        if((-1<=(Setpoint-prom)) and (Setpoint-prom)<=1){
          confH[1]=1;
        }
      }
    }
  }
}

```

```

        contTH=0;
    }
}
}
/* se verifica que se pida verificación de la temperatura
del hottend */
if(confC[0]==1){
    /* si la temperatura esta entre un rango +-0.7 del setpoint
de la cama ,entonces
    * se empieza a guardar las 4 siguientes temperatura de
la cama en VTC, una vez
    * guardadas esas temperaturas, se verifica que la
diferencia entre estas 4
    * temperaturas y el set no sea mayor a +- 1 grado, esto
porque pudo solo haberse
    * dado un sobrepaso en el control y no realmente se
estabilizo, si esta
    * entre ese rango de 1 grado, entonces se setea
confTC[1] para que el serial pueda
    * enviar la verificación al python
    */
    if ((setpoint_Cama-0.7)<real_temperature<(setpoint_Cama+0.
7)){
        if (contTC==0){
            VTC[contTC]=real_temperature;
            contTC=1;
        }
        else if (contTC==1){
            VTC[contTC]=real_temperature;
            contTC=2;
        }
        else if (contTC==2){
            VTC[contTC]=real_temperature;
            contTC=3;
        }
        else if (contTC==3){
            VTC[contTC]=real_temperature;
            float prom=(VTC[0]+VTC[1]+VTC[2]+VTC[3])/4;
            if((-1<=(setpoint_Cama-prom)) and
(setpoint_Cama-prom)<=1){
                confC[1]=1;
            }
        }
    }
}

```

```

    }
    contTC=0;
  }
}
}
}
//=====
=====//

////////////////////////////////////
////////////////////////////////////
//===== Funcion del PID de la
Cama =====//
////////////////////////////////////
////////////////////////////////////
void PIDCama () {
  currentMillis = millis(); //guardamos el valor
del tiempo antes del control
  if(currentMillis - previousMillis >= temp_read_Delay){
    previousMillis += temp_read_Delay;
//Incrementamos el tiempo para el proximo control
    MedirTemperaturaCama (); // Calculamos
la temperatura actual
    PID_error = setpoint_Cama - real_temperature;
//Calculate the pid ERROR

    if(PID_error > 30) //la integral no afetara al control
cuando la temperaturas es mayores a 30°C
    {PID_i = 0;}

    PID_p = kpc * PID_error;
//Calculamos el valor P
    PID_i = PID_i + (kic * PID_error);
//Calculamos el valor I
    timePrev = Time; // the previous time
is stored before the actual time read
    Time = millis(); // actual time read
    elapsedTime = (Time - timePrev) / 1000;
    PID_d = kdc*((PID_error - previous_error)/elapsedTime);
//Calculate the D value
    PID_value = PID_p + PID_i + PID_d;

```

```

//Calculate total PID value

//We define firing delay range between 0 and 7400. Read
above why 7400!!!!!!!
if(PID_value < 0)
{
    PID_value = 0;
}
if(PID_value > 7400)
{
    PID_value = 7400;
}
previous_error = PID_error; //Remember to store the
previous error.
}

//If the zero cross interruption was detected we create the
100us firing pulse
if (zero_cross_detected)
{
    delayMicroseconds(maximum_firing_delay - PID_value);
//This delay controls the power
digitalWrite(Cama_pin,HIGH);
delayMicroseconds(100);
digitalWrite(Cama_pin,LOW);
zero_cross_detected = false;
}
}
}
//=====
=====//

////////////////////////////////////
////////////////////////////////////
////////////////////////////////////
Initialization //

//
ATMEGA //
////////////////////////////////////
////////////////////////////////////

void setup() {
    Serial.begin(9600); //Inicio comunicacion serial
    pinMode(PWM_hottend, OUTPUT);
    TCCR2B = TCCR2B & B1111000 | B00000011; //ajuste frecuencia
    pwm3

```

```

pinMode(termistor, INPUT);
SetK(6.5,2.3,62.57); //seteo constantes K hottend
SetKC(203,7.2 ,1.04 ); //seteo constantes K Cama
SetSalidalimites(0, 255); //seteo limites PID hottend
Ventiladores(); //inicio los ventiladores
pinMode (Cama_pin,OUTPUT);
pinMode (zero_cross,INPUT);
// Activo las interrupciones el el pin D4
PCICR |= (1<<PCIE2); //enable PCMSK2
scan
PCMSK2 |= (1 << PCINT20); //Set pin D4 (zero cross input)
trigger an interrupt on state change.
}
//=====
=====//

////////////////////////////////////
////////////////////////////////////
/////                               Bucla
principal                               ////
////////////////////////////////////
////////////////////////////////////
void loop() {
  SerialC();
  //Determina si hay que ejecutar el PID de hottend o
  retornar de la función.
  if (millis() - lastTime >= SampleTime) {
    //Serial.println(confH[0]);
    CalculoPID();
  }
  PIDCama();
  VerificaSeteo();
  ////////////////////////////////////seguridades para las temperaturas
  if (Input>=240){
    analogWrite(PWM_hottend, 255);
  }
  if (real_temperature>100){
    digitalWrite(Cama_pin,LOW);
  }
}

```

```

//Interrupcion que detecta el Ophase
//-----
ISR(PCINT2_vect){
    ////////////////////////////////////////////Input from
    optocoupler
    if(PIND & B00010000){          //We make an AND with the
state register, We verify if pin D4 is HIGH???
        if(last_CH1_state == 0){    //If the last state was 0,
then we have a state change...
            zero_cross_detected = true; //We have detected a state
change! We need both falling and rising edges
        }
        else if(last_CH1_state == 1){ //If pin 8 is LOW and
the last state was HIGH then we have a state change
            zero_cross_detected = true; //We have detected a state
change! We need both falling and rising edges
            last_CH1_state = 0;      //Store the current state
into the last state for the next loop
        }
    }
}

```

CONFIDENTIAL

## Anexo 9: Algoritmo de Python de la ventana de impresión 3D y comunicación serial

```
#!/usr/bin/env python

# import libraries
import pygtk
pygtk.require("2.0")
import gtk
import sys,os
import linuxcnc, hal
import gladevcp.makepins
import subprocess
import gobject
from time import time
import time
import math
import serial
import threading
from interprete import Traducir
from os import system
from time import clock

# set up paths to files
BASE = os.path.abspath(os.path.join(os.path.dirname(sys.argv[0]), ".."))
libdir = os.path.join(BASE, "lib", "python")
sys.path.insert(0, libdir)
datadir = os.path.join(BASE, "share", "linuxcnc")
xmlname = os.path.join(datadir, "GUI4.glade")

#DIRECCION DE LOS ARCHIVOS DE TEXTO M
# M104 ----> SETEAR HOTTEND
# M140 ----> SETEAR CAMA CALIENTE
# M106 ----> SETEAR VENTILADOR
# M109---->ESPERAR A QUE TEMPERATURA HOTTEND SE ESTABILICE
# M190---->ESPERAR A QUE TEMPERATURA CAMA CALIENTE SE ESTABILICE
# M107 ---->Apagar ventilador
UM104="/home/dumen/linuxcnc/nc_files/M104.txt"
UM140="/home/dumen/linuxcnc/nc_files/M140.txt"
UM109="/home/dumen/linuxcnc/nc_files/M109.txt"
UM190="/home/dumen/linuxcnc/nc_files/M190.txt"
UM106="/home/dumen/linuxcnc/nc_files/M106.txt"
UM107="/home/dumen/linuxcnc/nc_files/M107.txt"
temperatura="/home/dumen/linuxcnc/nc_files/temperatura.txt"

#Variables de la comunicacion serial con atmega

serialPort="no hay ninguna coneccion"#variable de cofirmacion de coneccion usb
atmega
check=0#variable de cofirmacion de coneccion serial
arduinoPort=None #variable global del puerto serial
salir=1
THottend=0
TCama=0
PWMventiladores=0
```



```

SetCama=0
SetHottend=0
a = 0#variable de tiempo
Setmanual=0# variable para identificar que el seteo es manual
CSerial="Desactivado"
error="ninguno"
Tlineas=0
#Funcion para ver que este conectado el atmega
def ComprobarConeccionAtmega():
    global serialPort ,arduinoPort
    n=0
    while serialPort=="no hay ninguna coneccion" and n<100:
        x=system("ls /dev/ttyACM0")
        if x==0:
            serialPort = '/dev/ttyACM0'
            arduinoPort = serial.Serial(serialPort, 9600,timeout=2)
            time.sleep(1)
        else:
            y=system("ls /dev/ttyACM1")
            if y==0:
                serialPort = '/dev/ttyACM1'
                arduinoPort = serial.Serial(serialPort, 9600,timeout=2)
                time.sleep(1)
            n=n+1
#Funcion para ver que este establecida la coneccion serial
def ComprobarConeccionSerial():
    global check, CSerial, error
    check=0
    CSerial="Desactivado"
    error="ninguno"
    if serialPort!="no hay ninguna coneccion" :
        while check==0:
            var= "Q"
            arduinoPort .open()
            time.sleep(0.8)
            arduinoPort.flushInput()# limpiamos el bufer
            arduinoPort.write(var.encode('ascii'))#enviamos al
arduino
            serial=arduinoPort.readline().strip().decode('ascii')
            print(serial)#mostrar en pantalla el estado del serial
            if serial=="Listo":
                check=1
                CSerial="Activo"
            else:
                CSerial="Desactivado"
                error="intente otra vez o vuelva a conectar
Atmega"
        if serialPort=="no hay ninguna coneccion" :
            error="no hay ninguna coneccion con atmega" #poner cuadro de
            alerta que no deje continuar porque serial no esta conectado
            ### #####
            ### Funcion para enviar la temperatura a setear cuando ###
            ### un codigo M sea ejecutado ###

```

```

#####
def SerialP():
    global check ,a
    if serialPort!="no hay ninguna coneccion" and check==1:
        while(salir==1):
            comunicacionMcodes()
            time.sleep(0.8)
            b = clock()
            if b - a >0.35:
                ActualizarTemperaturas()
                a = b

#####
##          Funcion para ver los valores actuales de temperaturas ##
#####
def ActualizarTemperaturas():
    global THottend,TCama
    try:
        var= "R"
        arduinoPort.flushInput()# limpiamos el bufer
        arduinoPort.write(var.encode('utf-8'))#enviamos al arduino
        time.sleep(0.5)
        serial=arduinoPort.readline().strip().decode('utf-8')
        ValoresT=str(serial).split("+")
        THottend=ValoresT[0]
        TCama=ValoresT[1]
    except UnicodeDecodeError: # catch error and ignore it
        print('uh oh')
def comunicacionMcodes():
    global Setmanual
    if os.path.isfile(UM104)==True:#selear temperatura hottend
        Archivo=open( UM104,"r")
        Texto=Archivo.readlines()
        Archivo.close()
        try:
            var= "H"+Texto[1]
            arduinoPort.flushInput()# limpiamos el bufer
            arduinoPort.write(var.encode('utf-8'))#enviamos al arduino
            time.sleep(0.8)
            os.remove(UM104)
        except:
            os.remove(UM104)
    elif os.path.isfile(UM140)==True:#setear temperatura cama caliente
        Archivo=open( UM140,"r")
        Texto=Archivo.readlines()
        Archivo.close()
        try:
            var= "C"+Texto[1]
            arduinoPort.flushInput()# limpiamos el bufer
            arduinoPort.write(var.encode('utf-8'))#enviamos al arduino
            time.sleep(0.8)
            os.remove(UM140)
        except:

```

```

        os.remove(UM140)
elif os.path.isfile(UM106)==True:#setear ventilador velocidad
    Archivo=open( UM106,"r")
    Texto=Archivo.readlines()
    Archivo.close()
    try:
        var= "V"+Texto[1]
        arduinoPort.flushInput()# limpiamos el bufer
        arduinoPort.write(var.encode('utf-8'))#enviamos al arduino
        time.sleep(0.8)
        os.remove(UM106)
    except:
        os.remove(UM106)
elif os.path.isfile(UM107)==True:#apagar ventilador
    Archivo=open( UM107,"r")
    Texto=Archivo.readlines()
    Archivo.close()
    try:
        var= "V"+Texto[1]
        arduinoPort.flushInput()# limpiamos el bufer
        arduinoPort.write(var.encode('utf-8'))#enviamos al arduino
        time.sleep(0.8)
        os.remove(UM107)
    except:
        os.remove(UM107)
elif (arduinoPort.inWaiting()!=0):
    serial=arduinoPort.readline().strip().decode('ascii')
    if Setmanual==0:
        if serial=="HL":
            var= "S"
            arduinoPort.flushInput()# limpiamos el bufer
            arduinoPort.write(var.encode('utf-8'))#enviamos al
arduino
            time.sleep(0.2)
            Archivo=open( UM109,"w")
            Texto=Archivo.write(" temperatura hottend lista")
            Archivo.close()
        elif serial=="CL":
            var= "U"
            arduinoPort.flushInput()# limpiamos el bufer
            arduinoPort.write(var.encode('utf-8'))#enviamos al
arduino
            time.sleep(0.2)
            Archivo=open( UM190,"w")
            Texto=Archivo.write(" temperatura CamaCaliente lista")
            Archivo.close()
    if Setmanual==1:
        if serial=="HL":
            Setmanual=0
        elif serial=="CL":
            Setmanual=0
# main class
class gui(object):

```

```

def __init__(self, infile):
    self.builder = gtk.Builder()
    self.builder.add_from_file(xmlname)
    self.builder.connect_signals(self)
    self.window = self.builder.get_object("window1")
    mensaje=self.builder.get_object("Verror")
    self.window.connect("delete-event",gtk.main_quit)
    self.visualizador = self.builder.get_object("visualizador")
    self.halcomp = hal.component("gui")
    self.cnc = linuxcnc
    self.s = self.cnc.stat()
    self.c = self.cnc.command()
    self.ec = self.cnc.error_channel()
    self.panel = gladevcp.makepins.GladePanel(self.halcomp, xmlname,
self.builder, None)
    self.halcomp.ready()
    self.window.show()
    self.tcl_dir = os.environ["LINUXCNC_TCL_DIR"] + "/bin/"
    self.ini_file = os.environ["INI_FILE_NAME"]
    self.segundos = 0
    self.minutos = 0
    self.horas = 0
    self.tiempo = 0
    self.numero = 0
    self.bnumero = False
    self.archivo = 'gjhjdss'
    self.estadoant = 0
    self.bcero = False
    self.corriendo = False
    self.cicloChiller = 0
    gobject.timeout_add(1000, self.Datos)
    self.progreso = self.builder.get_object("progreso")

def General_Malt():
    text = _("Do you really want to close linuxcnc?")
    if not root_window.tk.call("nf_dialog", ".error", _("Confirm
Close"), text, "warning", 1, _("Yes"), _("No")):
        root_window.destroy()

def str2bool(self, v):
    return v.lower() in ("yes", "true", "1")

def on_Run_clicked(self, widget, data=None):
    global Tlineas
    self.s.poll()
    self.c.unhome(0)
    self.c.unhome(1)
    self.c.unhome(2)

```

```

        texto =str(self.s.file)
        leer= open(texto,"r")
        l=leer.readlines()
        leer.close()
        Tlineas =len(l)+1
        self.bcerero = False

    def on_tbLimpiar_clicked(self, widget, data=None):
        self.visualizador.clear_live_plotter()

    def on_tbDetener_clicked(self, widget, data=None):
        self.visualizador.clear_live_plotter()

# jogging
    def on_bDerecha_pressed(self, widget, data=None):
        self.s.poll()
        if self.builder.get_object("hJog").get_value() > 0 and
self.s.axis[0]['homed']:
            self.c.mode(linuxcnc.MODE_MANUAL)
            self.c.jog(self.cnc.JOG_CONTINUOUS, 0,
self.builder.get_object("hJog").get_value()/60)

    def on_bDerecha_released(self, widget, data=None):
        self.c.jog(self.cnc.JOG_STOP, 0)

    def on_bIzquierda_pressed(self, widget, data=None):
        self.s.poll()
        if self.builder.get_object("hJog").get_value() > 0 and
self.s.axis[0]['homed']:
            self.c.mode(linuxcnc.MODE_MANUAL)
            self.c.jog(self.cnc.JOG_CONTINUOUS, 0,
-self.builder.get_object("hJog").get_value()/60)

    def on_bIzquierda_released(self, widget, data=None):
        self.c.jog(self.cnc.JOG_STOP, 0)

    def on_bArriba_pressed(self, widget, data=None):
        self.s.poll()
        if self.builder.get_object("hJog").get_value() > 0 and
self.s.axis[1]['homed']:
            self.c.mode(linuxcnc.MODE_MANUAL)
            self.c.jog(self.cnc.JOG_CONTINUOUS, 1,
self.builder.get_object("hJog").get_value()/60)

    def on_bArriba_released(self, widget, data=None):
        self.c.jog(self.cnc.JOG_STOP, 1)

    def on_bAbajo_pressed(self, widget, data=None):
        self.s.poll()
        if self.builder.get_object("hJog").get_value() > 0 and
self.s.axis[1]['homed']:
            self.c.mode(linuxcnc.MODE_MANUAL)

```

```

        self.c.jog(self.cnc.JOG_CONTINUOUS, 1,
-self.builder.get_object("hJog").get_value()/60)

    def on_bAbajo_released(self, widget, data=None):
        self.c.jog(self.cnc.JOG_STOP, 1)

    def on_btTrazo_clicked(self, widget, data=None):
        self.s.poll()
        conjunto1 = str(self.s.file).split('/')
        conjunto2 = conjunto1[len(conjunto1)-1].split('.')
        self.soloArchivo = conjunto2[0]
        cadena = self.soloArchivo + '_trazo'
        print cadena
        if not self.s.estop and self.s.enabled and self.s.homed and
(self.s.interp_state == self.cnc.INTERP_IDLE):
            self.c.mode(self.cnc.MODE_MDI)
            self.c.wait_complete()
            self.c.mdi("<math>O</math>%s> call" % cadena)

    def on_bOrigen_pressed(self, widget, data=None):
        self.s.poll()
        if (not self.s.estop and self.s.enabled and self.s.homed and
(self.s.interp_state == self.cnc.INTERP_IDLE)):
            varx = self.s.axis[0]['output']
            vary = self.s.axis[1]['output']
            if abs(varx) < 1:
                varx = 0
            if abs(vary) < 1:
                vary = 0
            cadena = 'G10 L2 P1 X'+str(varx)+' Y'+str(vary)+' Z0'
            self.c.mode(linuxcnc.MODE_MDI)
            self.c.wait_complete() # wait until mode switch executed
            self.c.mdi(cadena)
            self.c.wait_complete()
            self.c.mode(linuxcnc.MODE_MANUAL)
            self.c.wait_complete()

    def on_bBomba_clicked(self, widget, data=None):
        self.ser.write("B\r")

    def on_Mostrador_clicked(self, widget, data=None):
        self.s.poll()
        texto = 'Archivo: '+str(self.s.file)
        self.builder.get_object("larchivo").set_text(texto)

    def on_Salir_clicked(self, widget, data=None):
        self.s.poll()
        print "Salir"
        self.ser.close()
        Gtk.main_quit()

    def on_bError_clicked(self, widget, data=None):
        self.ser.write("R\r")

```

```

def on_okVentiladores_clicked(self, widget, data=None):
    self.s.poll()
    AVentiladores= self.builder.get_object('AVentiladores')
    global PWMventiladores
    SetPWMventiladores= AVentiladores.get_text()
    var= "V"+SetPWMventiladores
    arduinoPort.flushInput()# limpiamos el bufer
    arduinoPort.write(var.encode('utf-8'))#enviamos al arduino
    time.sleep(0.5)
def on_OKCama_clicked(self, widget, data=None):
    self.s.poll()
    labelTCama= self.builder.get_object('TCama')
    global SetCama
    SetCama= labelTCama.get_text()
    Setmanual=1
    var= "C"+SetCama
    arduinoPort.flushInput()# limpiamos el bufer
    arduinoPort.write(var.encode('utf-8'))#enviamos al arduino
    time.sleep(0.5)
def on_OKHottend_clicked(self, widget, data=None):
    self.s.poll()
    labelTHottend= self.builder.get_object('THottend')
    global SetHottend
    SetHottend= labelTHottend.get_text()
    Setmanual=1
    var= "H"+SetHottend
    arduinoPort.flushInput()# limpiamos el bufer
    arduinoPort.write(var.encode('utf-8'))#enviamos al arduino
    time.sleep(0.5)
def on_Traducir_clicked(self, widget, data=None):
    global Tlineas
    self.s.poll()
    texto =str(self.s.file)
    self.builder.get_object("larchivo0").set_text('Archivo: '+
texto)
    Traducir(texto)
def on_window1_destroy(self, widget, data=None):
    print "Salir"
    self.ser.close()
    gtk.main_quit()
def Datos(self):
#barra de progreso
    self.s.poll()
    linea=self.s.current_line
    if Tlineas>0:
        Porcentaje=(linea*100)/Tlineas
self.builder.get_object("progreso").set_text(str(Porcentaje)+"%")
self.builder.get_object("HottendAc").set_text(str(THottend)+"
Celsius")
self.builder.get_object("CamaAC").set_text(str(TCama)+"
Celsius")

```

```

        self.builder.get_object("lserial").set_text(str(CSerial))

self.builder.get_object("VentiladoresAc").set_text(str(PWMventiladores))

        return True

    def postgui(self):
        #inifile = linuxcnc.ini(sys.argv[2])
        inifile = linuxcnc.ini(self.ini_file)
        postgui_halfile = inifile.find("HAL", "POSTGUI_HALFILE")
        return postgui_halfile,sys.argv[2]

# run the program
if __name__ == "__main__":
    if len(sys.argv) > 2 and sys.argv[1] == '-ini':
        print "2", sys.argv[2]
        hwg = gui(sys.argv[2])
    else:
        hwg = gui()
    # load a postgui file if one is present in the INI file
    postgui_halfile,inifile = gui.postgui(sys.argv[2])
    print "GUI postgui filename:",postgui_halfile
    if postgui_halfile:
        res = os.spawnvp(os.P_WAIT, "halcmd", ["halcmd",
        "-i",inifile,"-f", postgui_halfile])
        if res: raise SystemExit, res
    ComprobarConeccionAtmega()
    if serialPort!="no hay ninguna coreccion":
        ComprobarConeccionSerial()
        hilo= threading.Thread(name='Serial',target=SerialP)
        hilo.daemon = True
        hilo.start()
        gtk.main()
        arduinoPort .close()
    else:
        print( "\n NO HAY NINGUNA CONEXION CON EL USB DE ATMEGA \n\n")#
        A=0/0

```



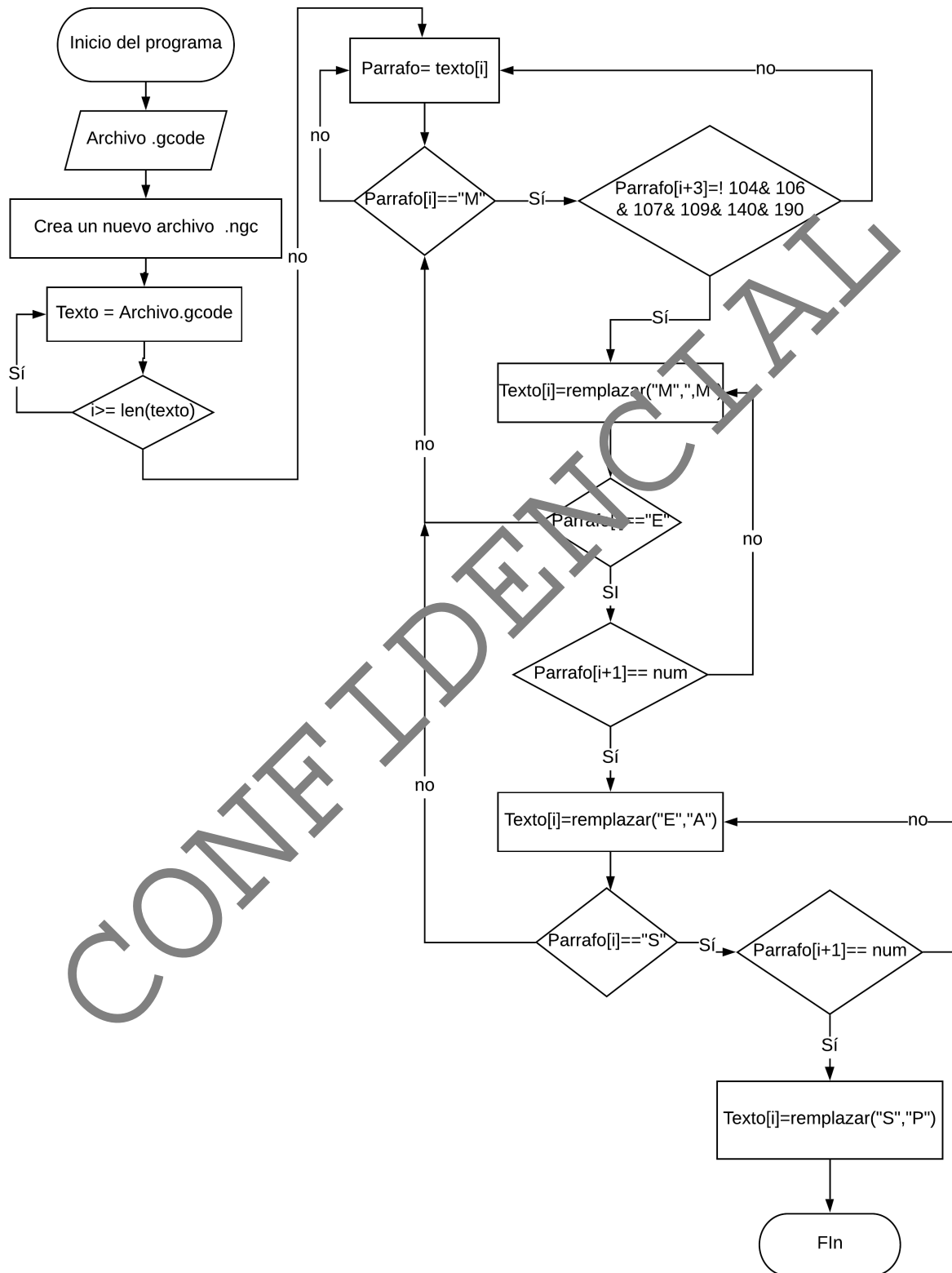
## Anexo 10: Algoritmo de Python de la ventana de selección de funciones(Corte laser o Impresión 3D)

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
"""
Created on Tue May 11 12:16:13 2021

@author: ASUS
"""
import os
import gi
from gi.repository import Gtk

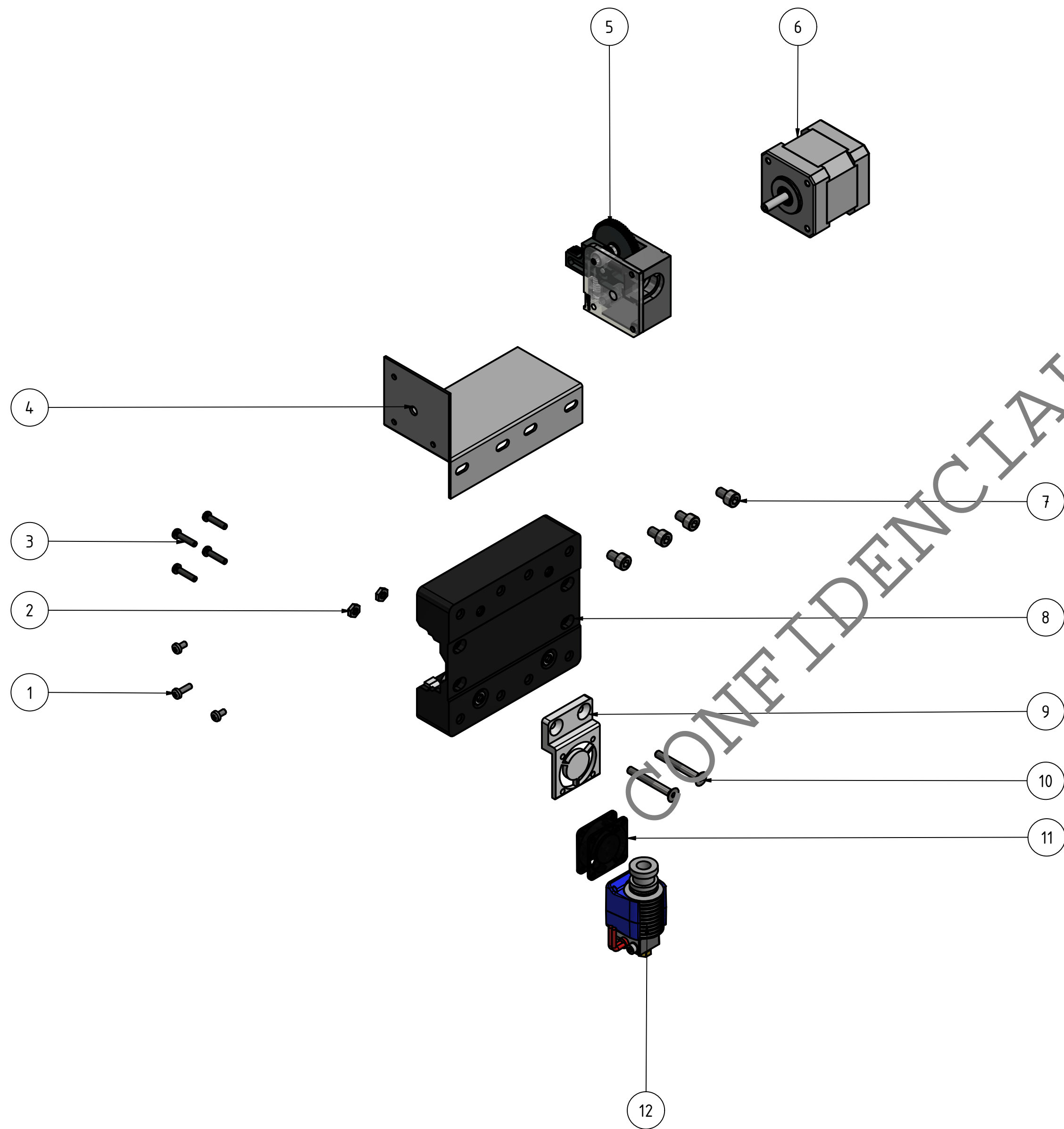
def imprimir(button):
    print("se aplasto imprimir")
    os.system(r"linuxcnc
/home/dumen/linuxcnc/configs/Impresora3D_2/Impresora3D_2.ini")
def corte(button):
    print("corte")
    os.system(r"linuxcnc /home/dumen/linuxcnc/configs/CNCLaser/CNCLaser.ini")
Builder=Gtk.Builder()
Builder.add_from_file("inicio.glade")
handlers={
    "cerrar":Gtk.main_quit,
    "terminar":Gtk.main_quit,
    "impresora":imprimir,
    "corte":corte,
}
Builder.connect_signals(handlers)
ventana=Builder.get_object("ventana")
ventana.show_all()
Gtk.main()
```

## Anexo 11: Diagrama de flujo del traductor



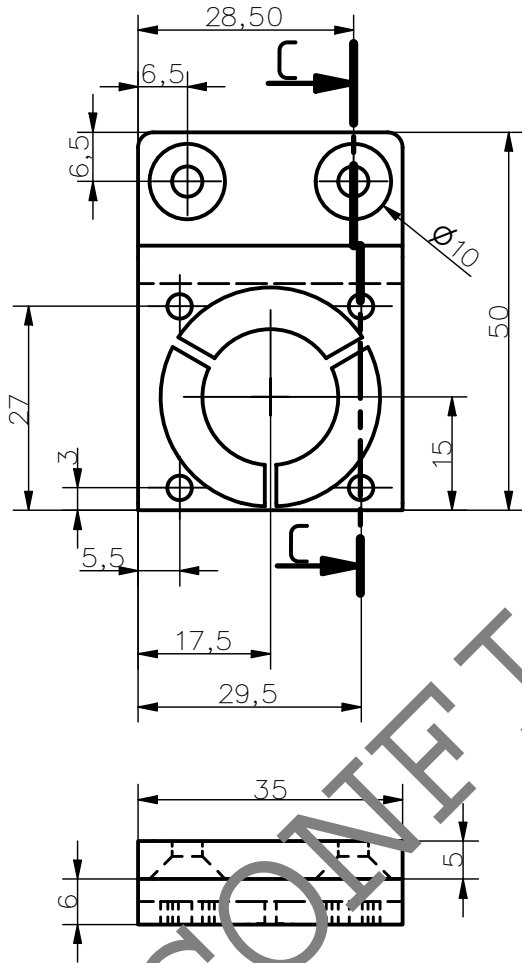
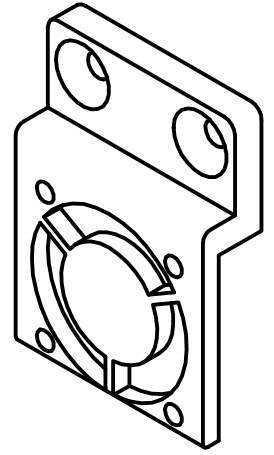
**Anexo 12:** Planos del diseño mecánico

CONFIDENCIAL

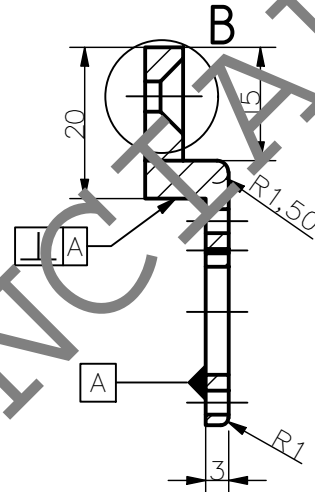


LISTA DE PIEZAS			
ELEMENTO	CTDAD	Nº DE PIEZA	DESCRIPCIÓN
1	3	DIN 7985 (H) - M3x10-H	Tornillos de cabeza cilíndrica abombada con hueco cruciforme - Tipo H
2	2	DIN EN 24 036 - M4	Tuercas hexagonales estrechas. Productos de clase B (sin chaflán)
3	4	ISO 7045 - M3 x 14 - 4.8 - H	Tornillo para maquinaria de cabeza cilíndrica redondeada con ranura y hueco cruciforme
4	1	Placa de sujeción del extrusor	
5	1	Extrusor E3D Titan AERO	
6	1	Motor Nema 17	
7	4	ISO 4762 - M5 x 8	Tornillo de cabeza cilíndrica con hueco hexagonal
8	1	Carro del Eje	
9	1	Acople modular del hotend	
10	2	Unbrako - M4 x 35	Tornillo avellanado
12	1	E3D v6 HotEnd	

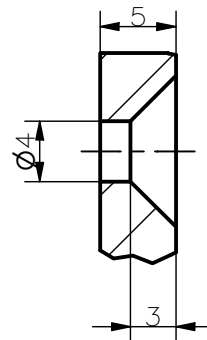
10º ciclo	Fecha	Nombre	<b>UNIVERSIDAD POLITÉCNICA SALESIANA</b> SEDE MATRIZ CUENCA INGENIERÍAS Carrera de Ingeniería Mecatrónica
Dibujado:	20/06/21	Hurtado & Vidal	
Comprobado:	20/07/21	Ing. C. Cobos	
Trabajo de titulación			
Escala:	Tol. gen.:		
Lámina:	1	<b>SISTEMA DE IMPRESIÓN 3D</b>	



C-C ( 1 : 1 )



B ( 2 : 1 )



N8

10º ciclo

Fecha

Nombre



**UNIVERSIDAD POLITÉCNICA  
SALESIANA**

Dibujado: 20/06/21

Hurtado & Vidal

SEDE MATRIZ CUENCA

Comprobado: 20/07/21

Ing. C. Cobos

Trabajo de titulación

**INGENIERÍAS**  
Carrera de Ingeniería Mecatrónica

Escala:

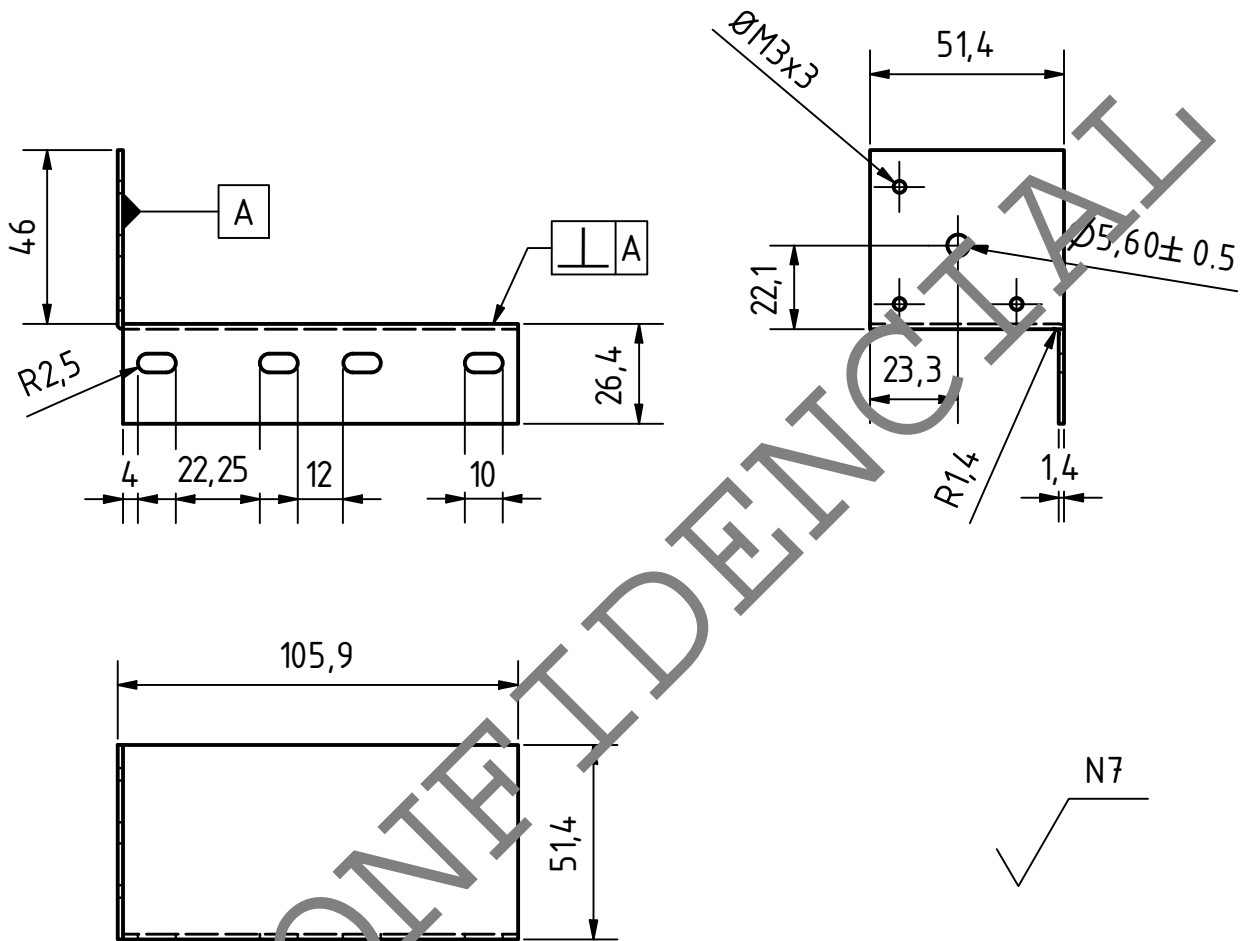
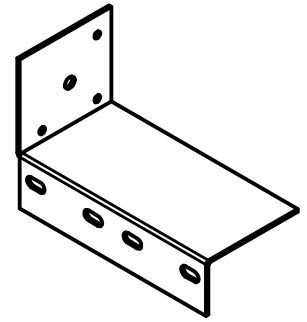
1:1

Tol. gen.:

Lámina:

2

**ACOPLE MODULAR DE HOTEND**



CONFIDENTIAL

<b>10º ciclo</b>	<b>Fecha</b>	<b>Nombre</b>		<b>UNIVERSIDAD POLITÉCNICA SALESIANA</b> <b>SEDE MATRIZ CUENCA</b>
<b>Dibujado:</b>	20/06/21	Hurtado & Vidal		
<b>Comprobado:</b>	20/07/21	Ing. C. Cobos		
<b>Trabajo de titulación</b>			<b>INGENIERÍAS</b> <b>Carrera de Ingeniería Mecatrónica</b>	
<b>Escala:</b>	<b>Tol. gen.:</b>	<b>PLACA DE SUJECIÓN DE EXTRUSIÓN</b>		
<b>Lámina:</b>	3			