

# **ESCUELA POLITÉCNICA NACIONAL**

**FACULTAD DE INGENIERÍA ELÉTRICA Y ELECTRÓNICA**

**ANÁLISIS, DISEÑO Y SIMULACION DE UN SOFTWARE PARA  
PROTOTIPOS DE IMPRESORAS 3D DE COMIDA LISTA PARA EL  
CONSUMO**

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO  
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN  
ELECTRONICA Y AUTOMATIZACION**

**CRISTHIAN GIOVANNY CORREA CALDERON**

**[cristhian.correa@epn.edu.ec](mailto:cristhian.correa@epn.edu.ec)**


**DIRECTOR: JORGE ANDRÉS ROSALES ACOSTA**

**[andres.rosales@epn.edu.ec](mailto:andres.rosales@epn.edu.ec)**

**DMQ, octubre 2022**

## **CERTIFICACIONES**

Yo, CRISTHIAN GIOVANNY CORREA CALDERON declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.



---

**CRISTHIAN GIOVANNY CORREA CALDERON**

Certifico que el presente trabajo de integración curricular fue desarrollado por CRISTHIAN GIOVANNY CORREA CALDERON, bajo mi supervisión.



---

**JORGE ANDRÉS FOSALES ACOSTA**  
**DIRECTOR**

## **DECLARACIÓN DE AUTORÍA**

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

CRISTHIAN GIOVANNY CORREA CALDERON

JORGE ANDRÉS ROSALES ACOSTA

## **DEDICATORIA**

Este trabajo que ha sido producto de mi esfuerzo, dedicación y perseverancia se la dedico en primer lugar a Dios por darme el conocimiento, la energía y la oportunidad de soñar día a día con mi objetivo de vida. También se las dedico a mis padres, Giovanny Correa y Susana Calderón que durante toda mi vida me han estado apoyando y guiando de la forma correcta para convertirme en el ser humano que soy hoy en día, por eso y más, los amo con todo mi corazón. Finalmente agradecer a todos mis familiares y amigos, entre ellos a Henry, Vero, Andrés, Cristian, Esteban, Jonathan que me han acompañado en este duro camino, incluso cuando este aun no comenzaba y me brindaron tantos bonitos e inolvidables momentos.

## **AGRADECIMIENTO**

Dar el agradecimiento a la Facultad de Ingeniería Eléctrica y Electrónica, a la Escuela Politécnica Nacional y a todos los profesores que han colaborado en mi formación profesional gracias a sus conocimientos impartidos y las horas de enseñanza en las aulas de la universidad.

## ÍNDICE DE CONTENIDO

CERTIFICACIONES .....	I
DECLARACION DE AUTORIA .....	II
DEDICATORIA .....	III
AGRADECIMIENTO .....	IV
INDICE DE CONTENIDO .....	V
RESUMEN .....	VIII
ABSTRACT .....	IX
1. INTRODUCCION .....	1
1.1 Objetivo general .....	2
1.2 Objetivos especificos .....	2
1.3 Alcance .....	2
1.4 Marco teórico .....	3
1.4.1 Softwares dedicados al diseño 3D .....	3
1.4.1.1 TinkerCAD .....	3
1.4.1.2 3D Slash .....	4
1.4.1.3 123D Design .....	5
1.4.2 Formato de archivos impresión 3D .....	6
1.4.2.1 Archivos STL .....	6
1.4.2.2 Código G .....	7
1.4.3 Placa base impresora 3D .....	9
1.4.3.1 Bloque driver .....	10
1.4.3.2 Bloque heather .....	11
1.4.3.2 Bloque núcleo .....	11
1.4.3 Interfaz gráfica humano-máquina .....	12
1.4.3.2 App Designer .....	12
2 METODOLOGÍA .....	14
2.1 Método de estudio .....	14
2.1.1 Método inductivo .....	14
2.1.2 Método hipotético-deductivo .....	14
2.1.3 Método experimental .....	14

2.2 Enfoque .....	15
2.2.1 Enfoque cualitativo .....	15
2.3 Tipo de investigación .....	15
2.3.1 Investigación exploratoria .....	15
2.3.2 Investigación descriptiva .....	16
2.4 Técnicas de recolección de información .....	16
2.4.1 Revisión permanente y reducción de datos .....	17
2.4.2 Técnicas de análisis de información .....	17
2.4.3 Disposición y transformación de datos .....	17
2.4.4 Análisis de contenido .....	17
2.4.5 Obtención de resultados y conclusiones .....	17
2.5 Parámetros de control .....	17
2.5.1 Control de temperatura .....	17
2.5.2 Control de motor DC .....	20
2.5.2.1 Modelo matemático del motor DC .....	20
2.5.2.2 Parámetros del motor DC .....	23
2.5.2.3 Control en cascada .....	23
2.5.2.4 Implementación en Simulink .....	23
2.5.2.5 Diseño de controladores .....	26
2.6 Diseño HMI .....	27
2.6.1 Ventana de presentación .....	28
2.6.2 Ventana de software de control .....	29
2.6.2.1 Carga y manejo de archivos STL .....	29
2.6.2.2 Validación de temperatura de chocolate .....	32
2.6.2.3 Validación de cantidad de chocolate en almacenamiento ...	33
2.6.2.4 Estado de impresión .....	34
2.6.2.5 Botones de interacción .....	35
2.6.2.5 Mensajes de error y advertencia .....	35
2.6 Comunicación con placa base .....	37
2.7.1 Recepción de datos .....	38
2.7.2 Transmisión de datos .....	39

2.7 Archivo ejecutable .....	40
3 RESULTADOS. CONCLUSIONES Y RECOMENDACIONES .....	41
3.1 Resultados .....	41
3.1.2 Sintonización de controladores .....	41
3.1.2.1 Respuesta del control del lazo de corriente .....	41
3.1.2.2 Respuesta del control del lazo de velocidad .....	43
3.1.2.3 Respuesta del control del lazo de velocidad .....	46
3.1.3 Proceso de impresión .....	48
3.2 Conclusiones .....	51
3.3 Recomendaciones .....	52
4 REFERENCIAS BIBLIOGRAFICAS .....	53
5 ANEXOS .....	56
ANEXO I .....	57
ANEXO II .....	60



## RESUMEN

En los últimos años la tecnología ha avanzado a pasos agigantados haciendo que llegue a incluirse en diferentes campos de estudio y desarrollo con la finalidad de que se tenga un progreso más amplio y detallado en todo su contenido. De esta manera la tecnología se ve inmiscuida en temas relacionados a medicina, biología, química alimenticia, entre otros. Un medio no tan explorado, pero con visión en el futuro es el de obtener comida 3D impresa que pueda ser apta para el consumo humano, siendo el producto final de diferentes tipos de materiales comestibles.

El proyecto completo se enfoca en el desarrollo tanto de hardware como de software de una impresora 3D capaz de imprimir una figura o modelo propuesto por el usuario hecho de chocolate. Este trabajo que será descrito en los siguientes capítulos se enfoca principalmente en el software utilizado para el control y el manejo de la impresora 3D a través de una interfaz fácil de utilizar para el usuario y con elementos visuales del estado de impresión en tiempo real.

El software que se desarrolla basa su funcionamiento en la carga de archivos de figuras tridimensionales previamente diseñadas en aplicaciones dedicadas para esa tarea y la elaboración del producto comestible a través de la conectividad del software desarrollado con la parte de hardware correspondiente que realiza la actividad de impresión.

**PALABRAS CLAVE:** App Designer, controlador, impresora 3D, software, archivo STL.

## **ABSTRACT**

In recent years, technology has advanced by leaps and bounds, making it possible to include it in different fields of study and development in order to have a broader and more detailed progress in all its content. In this way, technology is involved in issues related to medicine, biology, food chemistry, among others. A not so explored, but forward-looking means is to obtain 3D printed food that can be fit for human consumption, being the final product of different types of edible materials.

The entire project focuses on the development of both hardware and software for a 3D printer capable of printing a figure or model made of chocolate proposed by the user. This work, which will be described in the following chapters, focuses mainly on the software used to control and manage the 3D printer through an easy-to-use interface for the user and visual elements of the printing status in real time.

The software that is developed bases its operation on the loading of previously designed three-dimensional figure files in dedicated applications for that task and the elaboration of the edible product through the connectivity of the developed software with the corresponding hardware part that performs the printing activity.

**KEYWORDS:** 3D printer App Designer, controller, software, STL file.

# 1 INTRODUCCIÓN

Este documento es el primer tomo correspondiente a la elaboración de una aplicación que permita controlar una impresora 3D de alimentos que tiene como material de impresión una pasta de chocolate que a la temperatura adecuada puede ser moldeado por el extrusor del dispositivo con el objetivo de tener como producto final un diseño personalizado tridimensional apto para el consumo humano acorde a los requisitos del usuario. El siguiente tomo elaborado por Steve Rojas se abarcará un estudio más exhaustivo en el diseño de controladores para el extrusor y manejo de las diferentes partes móviles del hardware de la impresora 3D.

En esta aplicación se incluye varias opciones de manejo por el usuario, mencionando como más importante la de poder cargar archivos con la extensión STL, para que el software pueda hacer el correspondiente análisis y que los datos contenidos en él puedan ser procesados por la aplicación y posteriormente sean ejecutados por un hardware adaptable a dicho aplicativo. Además de ello se incluye visualizadores digitales para apreciar los diferentes parámetros físicos aportados el hardware de la impresora 3D para el monitoreo de su funcionamiento en tiempo real, tales como los datos de coordenadas del movimiento de los componentes móviles, la temperatura a la que se está depositando el chocolate, el nivel de material que está almacenado en el hardware, tiempos de impresión y cantidad de pasta de chocolate necesaria para poder imprimir el diseño previamente cargado en el software, entre otros.

Adicional a ello se elaboran los controladores de temperatura de la resistencia de calentamiento que lleva a la pasta de chocolate a la temperatura adecuada para que esta sea moldeable y dar origen al diseño tridimensional cargado. Además, se hace el control de posición del extrusor, tomando en cuenta tiempos de respuesta y sobrepicos con la técnica de control de cascada utilizando la herramienta Simulink que viene incorporada en el software Matlab para la verificación de resultados y desempeño correcto del mismo.

Como último paso se detalla la manera de conectar el software con el hardware de la impresora 3D mediante comunicación serial que al igual que los otros apartados, se usara Simulink y sus librerías para lograr la recepción y envío de datos para el manejo del dispositivo. Adicional a todo ello, se adjunta en los anexos los enlaces de descarga hacia los diferentes archivos creados para que cualquier persona que desee hacer uso de este software tenga libre acceso a utilizarlos y además se incluye un video demostrativo de como está funcionando el software desde la carga del archivo STL hasta la obtención del producto final.

## **1.1 Objetivo general**

Desarrollar un software que permita controlar un prototipo de una impresora 3D de comida para la obtención de objetos tridimensionales diseñados en softwares dedicados para esa tarea, dando al usuario la pertinencia de visualizar el progreso de impresión en tiempo real y la apreciación de las variables suficientes que intervienen en este proceso.

## **1.2 Objetivos específicos**

1. Analizar 2 tipos de impresoras 3D, así como las partes constitutivas de ellas y su funcionamiento para poder adaptarlas a que el producto final de ellas pueda ser consumida por el ser humano.
2. Verificar el funcionamiento de 3 tipos de software que son utilizados en impresoras 3D comerciales que permitan controlar las partes mecánicas del dispositivo para la obtención del producto final.
3. Estudiar 5 tipos de ingredientes que sean aptos para el consumo humano y que puedan ser utilizados para la elaboración de alimentos tridimensionales al utilizar una impresora 3D.
4. Diseñar un software con una interfaz de fácil uso para el usuario donde se incorpore las herramientas necesarias y la visualización correspondiente de variables para el proceso de impresión 3D.
5. Simular el software diseñado adaptable a impresoras 3D para la verificación del funcionamiento de este y la visualización de las herramientas incorporadas.

## **1.3 Alcance**

1. Mediante uno de los softwares utilizados durante la carrera, diseñar una interfaz gráfica que sea de sencilla comprensibilidad y manejo para el usuario que haga uso de este. En esta interfaz se podrá cargar archivos de la extensión STL para que el software pueda realizar la interpretación de estos y enviar a la parte de hardware las órdenes para realizar la impresión.
2. Tener un software completamente funcional que permita realizar el control de las partes eléctricas y mecánicas de un prototipo de impresora 3D para la obtención de

modelos tridimensionales que tengan como material de elaboración chocolate y el producto final sea apto para el consumo humano.

3. Aportar un software completo, totalmente adaptable para el prototipo de una impresora 3D de productos alimenticios.
4. Explorar los diferentes productos alimenticios que pueden ser utilizados en la elaboración de productos que sean resultado de una impresión 3D para que puedan ser consumidas por el ser humano.
5. Proponer un modelo de impresora 3D que cumpla todas las funciones requeridas de dicho tipo de dispositivos y que pueda ser fácilmente implementada.
6. Permitir que los usuarios puedan obtener alimentos a partir de modelos creados por el usuario a su completa conveniencia y desarrollo.

## **1.4 Marco teórico**

El punto de partida para el proceso de impresión en 3D es un modelo digital tridimensional, el cual puede ser creado a través de diferentes softwares dedicados al diseño de objetos 3D. [1]

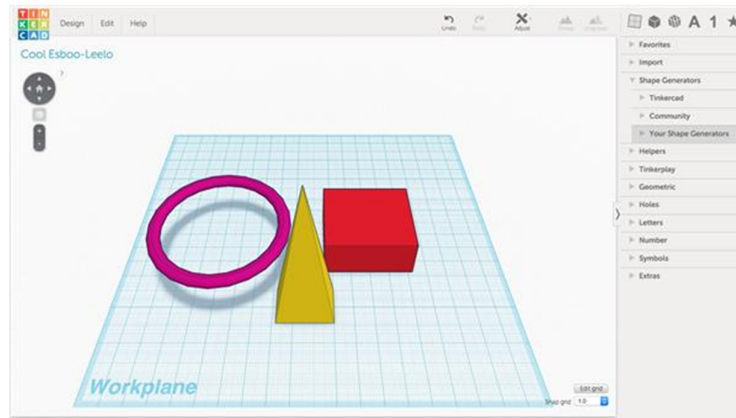
### **1.4.1 Softwares dedicados al diseño 3D**

En el mercado digital actual, hay varios softwares utilizados para la elaboración de modelos tridimensionales. A continuación, se hace un breve repaso de algunos de ellos y sus características más relevantes.

#### **1.4.1.1 TinkerCAD**

Esta herramienta desarrollada por Autodesk Esta destinada a principiantes y gente que está iniciando en el modelado en tres dimensiones. Mas que ser un software, es una plataforma virtual online ya que se puede encontrar en la web a través del enlace <https://www.tinkercad.com/>. Entre los sólidos que incluye esta plataforma son figuras preestablecidas en 3D como estrella, corazón, letras y números, además de cuerpos geométricos tridimensionales como cubo, esfera, prismas, etc. Estos solidos funcionan como bloques de construcción, es decir que el diseñador puede realizar una construcción gradual haciendo uso de las figuras simples antes descritas. Una vez que el usuario haya

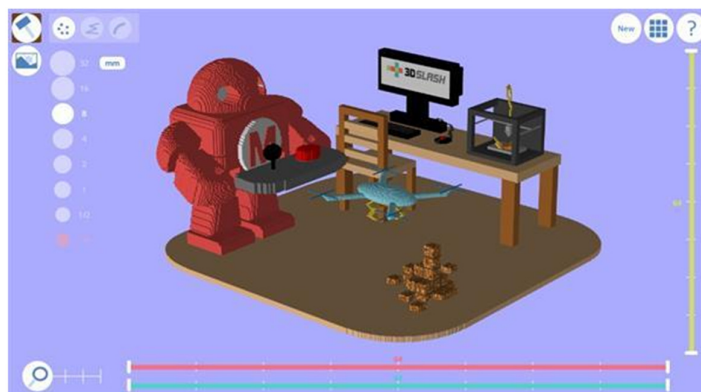
terminado su diseño, se podrá guardar el archivo en diferentes formatos como lo son stl, obj, entre otros.



**Fig. 1.1.** Ventana de trabajo de software TinkerCAD.

#### 1.4.1.2. 3D Slash

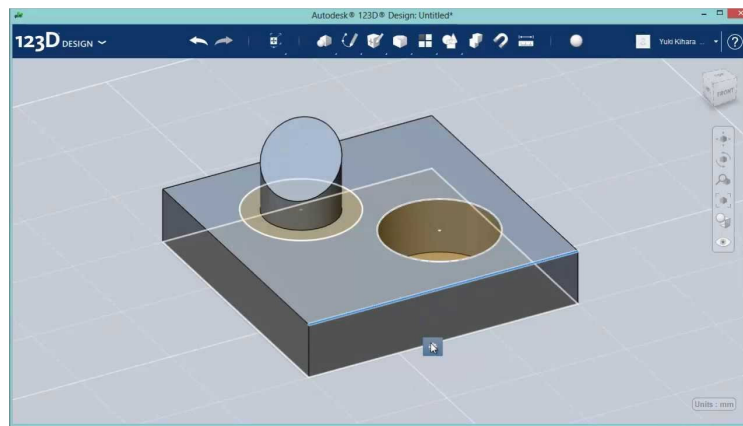
Al igual que TinkerCad, es una herramienta para diseñadores junior con la principal característica de que incluye un concepto de modelado de los videojuegos de Minecraft y hace uso de herramientas como martillos y cinceles para dar forma a los objetos que requiera el usuario. Al momento se ha lanzado la versión 2.0 de este software que incluye videos ilustrativos para el aprendizaje desde cero del uso de esta herramienta. También permite la exportación de los diseños en archivos de formato obj y stl. Esta herramienta se la puede ocupar mediante el sitio web <https://www.3dslash.net>, tiene funcionalidad online así como aplicación descargable de escritorio para dispositivos con sistema operativo Microsoft Windows y Linux.



**Fig. 1.2.** Ventana de trabajo de software 3D Slash.

### 1.4.1.3. 123D Design

Este software desarrollado por Autodesk al igual que TinkerCad, contiene una biblioteca con modelos tridimensionales 3D de diferentes objetos que pueden ser editados por el usuario. Las herramientas que se incluyen en la primera plataforma descrita en este apartado están presentes en 123D Design y adicional a ellas está la de que los proyectos creados se pueden subir de forma directa al servicio de Shapeways, el cual es un servicio de impresión en 3D. 123D Design puede ser descargado del sitio web: <https://www.autodesk.com/solutions/123d-apps>. [2]



**Fig. 1.3.** Ventana de trabajo de software 123D Design.

Todas las herramientas que se han descrito hasta ahora para el diseño y modelado 3D tienen una cosa en común y es que permiten la extracción de los proyectos elaborados en diferentes formatos que pueden ser ejecutados por los controladores de las impresoras 3D.

La característica fundamental de los formatos de archivo de proyectos 3D es que codifican la geometría de las figuras del proyecto, lo cual se hace de 3 maneras que son:

- Malla aproximada
- Malla precisa
- Geometría solida constructiva (CSG)

Al realizar la impresión en 3D, se hace uso de una malla aproximada dado que las impresoras 3D tiene una resolución muy baja de impresión, mientras que las otras maneras de codificación mencionas lo hacen en una resolución media y alta. La codificación de malla aproximada utiliza la técnica de “teselado” que consta de formar la superficie de un objeto mediante figuras geométricas básicas sin que existan huecos ni solapamientos. Lo cual da origen a que existan 3 formatos habitualmente utilizados que son:

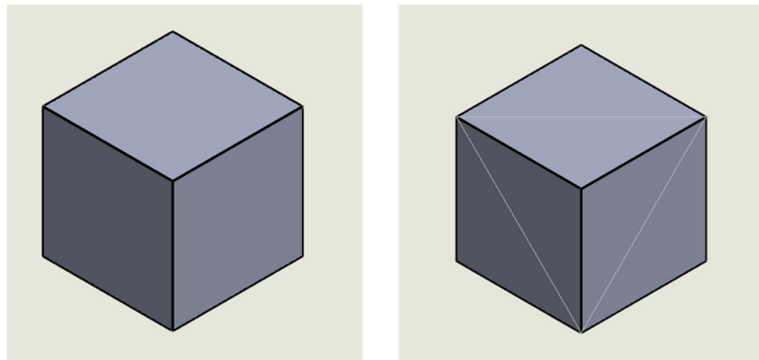
- STL
- OBJ
- PLY

De los 3 formatos mencionados, el archivo STL, que por sus siglas en ingles es Standart Triangle Language es el menos pesado en cuanto al espacio ocupado en memoria, siendo habitual que su peso sea en el orden los kilobits y esto hace que su procesamiento se realice con mayor velocidad. Este formato de archivo excluye varia información del objeto como su textura, color y propiedades físicas que en los otros formatos si se incluyen. [3]

## 1.4.2 Formato de archivos impresión 3D

### 1.4.2.1 Archivos STL

El fichero STL hace una codificación del modelo 3D mediante un mosaico donde se divide toda la superficie en figuras geométricas sin que quede huecos entre ellas, dando como resultado algo muy parecido a lo que se aprecia cuando se observa una pared embaldosada. En el procedimiento de desmenuzar el modelado 3D generalmente se utilizan triángulos, por lo que a manera de ejemplo se podría imaginar un cubo que mediante esta técnica se dividiría su cara en 2 triángulos y al poseer 6 caras, daría como resultado un total de 12 triángulos.



**Fig. 1.4.** Segmentación de cuerpos 3D al convertirlo en archivo STL.

A estos triángulos se los conoce como facetas y son los que se guardan en el fichero con extensión stl, con lo cual se reduce la información para retener los datos del cubo ya que solamente se guardan la posición de cada uno de los puntos que forman los triángulos y un vector normal a cada superficie que conforman al modelo 3D. Cuando este modelo es más complejo la técnica sigue funcionando al utilizar triángulos para cubrir toda superficie.



A mayor cantidad de triangulo que contenga el archivo STL, el objeto esta más definido, haciendo de esta manera un símil a lo que es una cámara digital y la resolución de pixelaje con la cual captura ésta las imágenes.

Una impresora 3D hace el procesamiento de datos de un archivo STL mediante un slicer que es una herramienta capaz de convertir los datos del archivo en instrucciones para la impresora. Su procedimiento se basa en la división en capas del objeto para que el dispositivo pueda realizar la formación del objeto en 3 dimensiones de abajo hacia arriba.

Existen 2 condiciones que debe cumplir el archivo STL para que pueda ser procesado por el software, las cuales son que:

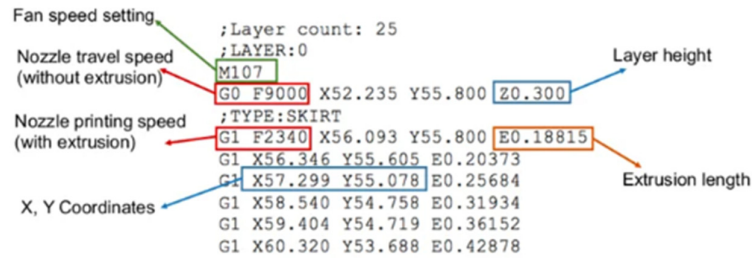
- El archivo tenga un grosor de pared diferente de cero.
- La malla que se forma con triángulos este totalmente cerrada. [4]

Finalmente, las instrucciones que son enviadas a cada uno de los componentes físicos de la impresora 3D son traducidos por un ensamblador mediante la conversión del fichero STL u otro formato de archivo de imagen tridimensional (OBJ o PLY) en un código de interpretación denominado como G-Code o código G que es el lenguaje utilizado por las CNC (máquina de control numérico por ordenador). Estas instrucciones son almacenadas en un archivo en formato de texto con ciertos comandos que determinan los parámetros de velocidad, temperatura, posición, entre otros, para llevar a cabo la elaboración del producto final. [5]

#### **1.4.2.2 Código G**

La codificación en código G que es el lenguaje utilizado por las CNC (máquina de control numérico por ordenador). Estas instrucciones son almacenadas en un archivo en formato de texto con ciertos comandos que determinan los parámetros de velocidad, temperatura, posición, entre otros, para llevar a cabo la elaboración del producto final.

En la figura 1.4 se muestra un ejemplo de estructura del código G, donde las letras X, Y Z representan los tres ejes de la impresora 3D. Un valor de X positivo 0 moverá el extrusor hacia la derecha, mientras que valores positivos de Y moverá el extrusor hacia atrás y con valores positivos de Z moverá el extrusor hacia arriba. La letra F en cambio representa la velocidad con la cual se moverá el extrusor y el valor que lo acompaña será el módulo de esta velocidad representado en unidades de milímetros por minuto. La letra E representa la longitud del movimiento en milímetros.



**Fig. 1.5.** Ejemplo de estructura del código G.

G1 es el comando más básico, encargándose del depósito de material cuando el extrusor se esté moviendo, por el contrario, cuando se coloca el comando G0, lo que se esté ordenando a la impresora es que no deposite material en ese lugar. [6]

En la tabla 1.1, se menciona la mayoría de los comandos utilizados al elaborar un código G.

**Tabla. 1.1.** Instrucciones básicas del código G.

Instrucción	Acción
G0	Movimiento lineal rápido
G1	Movimiento lineal con extrusión de material
G4	Tiempo de espera en la ejecución de comandos
G28	Autoinicio de los ejes de coordenadas
G90	Uso de coordenadas absolutas
G91	Uso de coordenadas relativas
G92	Ajuste de la posición actual a las coordenadas proporcionadas.
M104	Ajuste de la temperatura del extrusor
M105	Lectura del valor de temperatura actual
M17	Habilitación de los motores a pasos
M18	Deshabilitación de los motores a pasos
M92	Establecimiento de los pasos por unidad de los ejes
M140	Establecimiento de la temperatura objetivo
M200	Configuración del diámetro del filamento
M302	Realizar una extrusión en frío

Dependiendo de la impresora 3D utilizada, estos comandos pueden cambiar ya que existen diferentes modelos estructurales de impresoras en lo que se refiere a hardware.

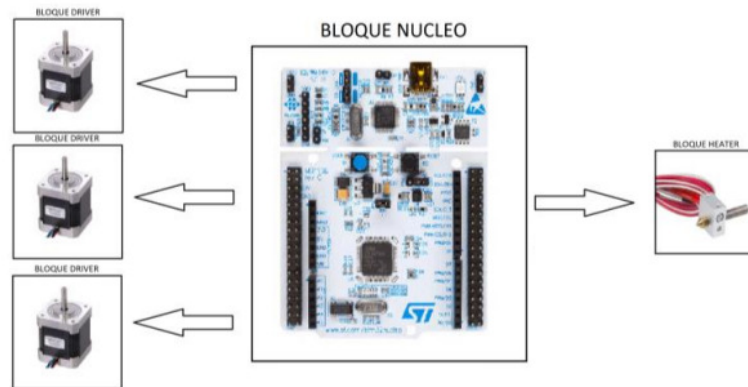
### 1.4.3 Placa base impresora 3D

Todos los dispositivos electrónicos que se usan de manera cotidiana o que se visualizan en el diario vivir están constituidos por componentes mecánicos y electrónicos que funcionan en conjunto, ya sea que se refiera a un smart TV, un celular, una computadora personal, una impresora, entre otros. En estos dispositivos se tiene un núcleo o componente clave que es la tarjeta madre, también conocido como placa base que está compuesto por un microprocesador, memoria y demás circuitos que son fundamentales para la operación de todas las partes del PC. [7]

Las tarjetas madre de una impresora 3D tiene 3 bloques que se detallan a continuación

- 1) El bloque núcleo: está constituido por el microcontrolador que este puede ser de cualquier clase y especificaciones, según el fabricante. En él, generalmente viene cargado de fábrica el firmware con los algoritmos de control para los motores y la temperatura del extrusor. Este firmware es capaz de realizar la conversión de las instrucciones que están codificadas en los diferentes tipos de archivos de diseño 3D que han sido descritos anteriormente y llevar a una instrucción para realizar la construcción del modelo 3D.
- 2) El bloque driver: su funcionamiento se basa en la transformación de las señales de control que son producidas en el microcontrolador en señales eléctricas, generalmente corriente, que sean capaces de energizar a las bobinas de los motores a pasos que constituyen la impresora 3D para el control de su movimiento, ya sea su posición o velocidad.
- 3) El bloque heater: al igual que el bloque driver, este bloque se encarga de la transformación de las señales digitales en eléctricas, pero en este caso son enviadas al calentador del material para el control de la temperatura del extrusor.

En la figura 1.6 se muestra la constitución del hardware de la impresora 3D que está siendo elaborada en este proyecto, pero la cantidad de bloques podría diferir dependiendo del dispositivo que se esté elaborando o que ya se tenga y se lo esté utilizando.



**Fig. 1.6.** Estructura física de impresora 3D a partir de placa base.

### 1.4.3.1 Bloque driver

Como se ha mencionado anteriormente, este bloque tiene cumplir la función de convertir las señales digitales en movimiento para el eje del motor a pasos mediante la energización de las bobinas de forma correcta. En este bloque se tienen diferentes entradas y salidas.

#### - Entradas:

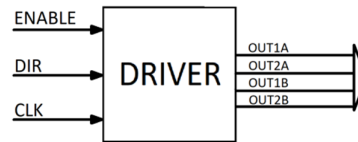
**ENABLE:** se tiene una señal digital de niveles alto o bajo para indicar si los motores están habilitados para ser utilizados. Un nivel bajo indicará que el eje del motor esta libre, mientras que un nivel alto indicará que la bobina del motor esta energizada y podrá ser controlado.

**DIR:** la señal es de tipo digital y con ella se podrá controla la dirección en la que gira el motor. Si el nivel de la señal es alto, entonces el motor girará en un sentido, mientras que si el nivel de la señal es bajo girará en sentido contrario.

**CLK:** es una señal digital que realiza el giro del eje del motor en los flancos de subida que se detecte.

#### - Salidas:

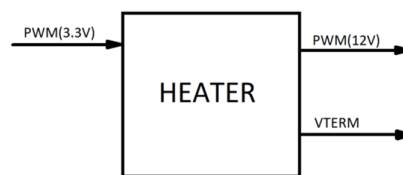
**OUT1A, OUT2A, OUT1B Y OUT2B:** estas son las salidas hacia las fases de los motores a paso, su salida es de tipo eléctrica, específicamente de corriente y ejecutan el movimiento de ellos. La simbología de letras A y B es para diferencias las 2 fases. Este bloque se lo coloca de manera individual para cada bloque, pero la señal de ENABLE si es común de los 3 motores que se tiene.



**Fig. 1.7.** Diagrama E/S bloque driver.

### 1.4.3.2 Bloque heater

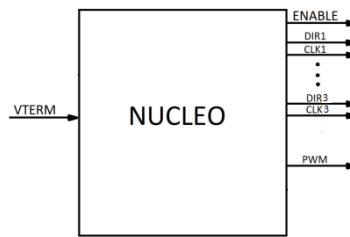
El bloque visto anteriormente las entradas digitales eran unidireccionales, llevaban una señal del microcontrolador hacia el motor a pasos y estos no enviaban ninguna información al micro. En cambio, en este bloque, si se tiene una señal bidireccional. Este cumple la función de la transformación de una entrada digital PWM de 0 a 3.3 V en otra señal de tipo PWM que se aplica al calentador con valor de voltaje de entre 0 a 12 V, logrando de esta manera realizar un control de la potencia que se suministra. Por otra parte, se obtiene una respuesta con respecto al calentamiento mediante una señal del sensor de temperatura en el extrusor, esta señal VTERM es analógica con valores ente 0 a 3.3 V.



**Fig. 1.8.** Diagrama E/S bloque heater.

### 1.4.3.3 Bloque núcleo

Es el cerebro de la tarjeta madre, se encarga de controlar a los otros bloques. Posee una señal de salida digital común a los drivers que es la ENABLE y las salidas digitales a los bloques drivers que ya se ha descrito anteriormente. Para el bloque heater tiene la entrada analógica proporcionada por el sensor de temperatura y la salida PWM para el control de la potencia que se suministra al calentador del material. [8]



**Fig. 1.9.** Diagrama E/S bloque núcleo.

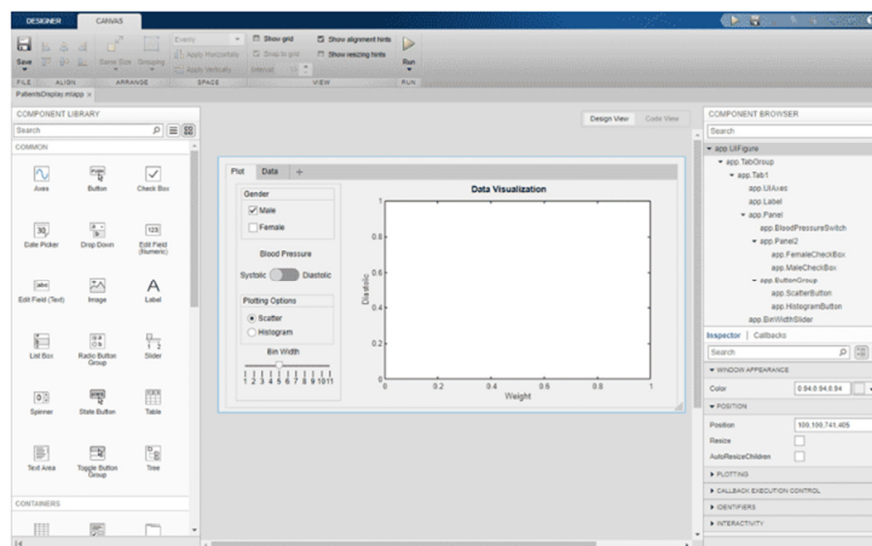
### 1.4.4 Interfaz gráfica humano-máquina

Para el diseño del HMI que sea capaz de controlar a la impresora 3D se hará uso del software Matlab mediante su herramienta App Designer.

#### 1.4.4.1 App Designer

Este componente de Matlab permite diseñar aplicaciones de tipo profesional a un desarrollador amateur mediante componentes visuales para la elaboración de la interfaz con la cual interaccionara el usuario, conocido como GUI por sus siglas en inglés, y el uso de un editor de código integrado para programar el comportamiento.

El diseño de la interfaz se lo realiza en un espacio de trabajo donde desde una paleta de objetos se arrastra y coloca estos según los requerimientos del diseñador como se lo puede apreciar en la figura 1.10.



**Fig. 1.10.** Entorno de trabajo App Designer.

Y para la definición del comportamiento de la app diseñada, esta aplicación tiene una versión de MATLAB Editor integrada donde se escribe el código a ejecutarse, además de poseer en la misma ventana de trabajo el componente de Code Analyzer que brinda una visualización de mensajes de advertencia y errores en el código. [9]

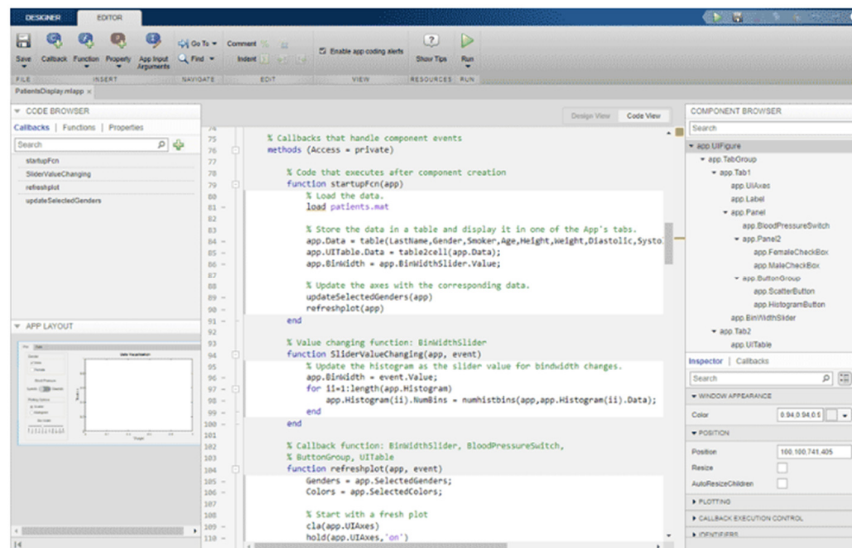


Fig. 1.11. Ventana de programación de código de App Designer.

## **2 METODOLOGÍA**

La elaboración de este trabajo de integración curricular ha sido desarrollado mediante los métodos inductivo, hipotético-deductivo y experimental. [10]

### **2.1 Método de estudio**

#### **2.1.1 Método inductivo**

En el punto de partida de este proyecto se cuenta con los conocimientos adoptados y la formación adquirida durante toda la carrera universitaria, dando lugar a unas bases sólidas en lo que respecta al funcionamiento y el manejo de diferentes softwares de simulación, el estudio y medición de variables físicas y la operatividad y control de una gran variedad de componentes eléctricos y electrónicos.

#### **2.1.2 Método hipotético-deductivo**

Mediante varios procesos de indagación, búsqueda y análisis realizados acerca del tema principal establecido en este trabajo, se ha llegado a la conclusión de que en la Escuela Politécnica Nacional no se ha realizado un trabajo de tal magnitud donde se ponga a prueba el nivel de investigación del estudiante en lo que respecta a la tecnología de las impresoras 3D, su funcionamiento y constitución. Resultando un desafío para el estudiante en aseverar las hipótesis que se tienen hasta dar comienzo a este trabajo y convertirlas en tesis para una implementación funcional del software desarrollado que complemente a un prototipo físico del dispositivo en mención.

#### **2.1.3 Método experimental**

Con la elaboración de un software capaz de poder interactuar con un hardware destinado a la impresión 3d de comida lista para el consumo humano se trata de contrarrestar la hipótesis planteada donde se permita poner a prueba los conocimientos del estudiante en el desarrollo aplicaciones de control, además de brindar al lector o cualquier persona interesada, una herramienta de tipo informática que abarque los aspectos más relevantes de los parámetros de funcionamiento de las impresoras 3D de comida mediante una interfaz intuitiva para el usuario.



## **2.2 Enfoque**

A través del enfoque cualitativo se ha diseñado este planteamiento metodológico debido a que este tipo de enfoque es el que mejor se adapta y confabula con las características que serán presentadas, dando una visión holística del trabajo presentado.

### **2.2.1 Enfoque cualitativo**

Este tipo de enfoque hace énfasis en la realidad del contexto natural y el cómo están sucediendo los acontecimientos, además de conocer la función de los instrumentos utilizados para la recolección de la información como puede ser entrevistas, artículos, observaciones, entre otros que ayuden al investigador en la recolección de información que produzca datos descriptivos. Según los investigadores Taylor y Bogdan (1897), este tipo de enfoque es inductivo ya que la persona que realiza la tarea de investigación desarrolla los conceptos a partir de datos e información ya establecida, mas no de hipótesis o teorías preconcebidas.

Debido a una investigación exhaustiva acerca de las características físicas de diferentes tipo de impresoras 3D y como estas operan, tanto en lo que respecta a hardware como ha software se ha llegado a establecer como un modelo de este dispositivo a la impresoras tridimensionales que únicamente con el movimiento del extrusor de material en tres ejes, se llega a tener un producto final apto para el consumo humano y partir de dicha premisa, se desarrolla el software que controle la impresora.[11]

## **2.3 Tipo de investigación.**

En el transcurso del desarrollo de este trabajo se ha hecho algunos tipos de investigaciones para poder lograr los objetivos planteados, entre estos tipos de investigación tenemos de tipo exploratoria, descriptiva

### **2.3.1 Investigación exploratoria**

Este tipo de investigación es realizado cuando el objetivo planteado es un tema poco estudiado o que no se haya abordado antes, siendo fuente de consulta guías no oficiales e información vagamente relacionada al tema a tratarse. La finalidad de este tipo de investigación es incrementar el grado de familiaridad con contenido que de cierta manera es desconocido, además de compilar la mayor cantidad de ideas que aporten con una explicación puntual acerca de una idea en concreto.

En el caso de las impresoras de comida en 3D, no se tiene antecedentes de que se haya abordado un tópico de tal magnitud en la Escuela Politécnica Nacional y a nivel de las universidades del Ecuador se tiene pocas referencias, por lo cual se ha trazado un objetivo exploratorio a través de la web para poder desarrollar este trabajo.

### **2.3.2 Investigación descriptiva**

La meta del investigador es describir los eventos y situaciones en el ámbito de cómo se están manifestando. Por medio de la investigación descriptiva se busca a través de diferentes procedimientos llegar a especificar las propiedades más importantes que logren una exposición completa de la idea tratada.

Durante la ejecución y desarrollo del software de control para una impresora 3D de comida se hace énfasis en cada uno de los aspectos que comprenden su funcionamiento a través de explicaciones y conceptos elementales para que el lector discierna la información de forma sencilla. [12]

## **2.4 Técnicas de recolección de información**

Como se ha mencionado anteriormente en los tipos de investigación, este trabajo de integración curricular tiene su base de consulta en la web a través de papers, documentos informativos, registros, videos explicativos y observaciones con el afán de recolectar la mayor cantidad de datos e información que permitan cumplir los objetivos planteados para el desarrollo del software de control de las impresoras 3d de comida apta para el consumo humano.

### **2.4.1 Técnicas de análisis de información**

El análisis es un conjunto de operaciones, reflexiones, transformaciones y comprobaciones que se realizan con los datos con la finalidad de extraer las ideas más significantes y relevantes con respecto a un tema de investigación. Esta actividad parte de examinar sistemáticamente un grupo de elementos informativos descubriendo que cual es la relación entre dichos elementos y la relación que tienen con el todo.

Las técnicas que han sido utilizadas en el análisis de información para el desarrollo de este trabajo de integración curricular son:

### **2.4.2 Revisión permanente y reducción de datos**

La primera tarea que se ha realizado es simplificar la información, realizar un resumen de todo lo consultado para que los datos se puedan manejar de una mejor manera y sea abarcable para el análisis del investigador.

### **2.4.3 Disposición y transformación de datos**

Se ha organizado la información de manera grafica luego de haber sido recolectada para una interpretación fácil y análisis secuencial de la misma, logrando de esta manera la conexión entre los conceptos y la transición de los procesos para el desarrollo del tema.

### **2.4.4 Análisis de contenido**

Mediante esta técnica se descubren los significados de los documentos que se han objeto de consulta poniendo de esta forma en manifiesto todo el abanico de información para poder codificar y clasificar los diferentes elementos en categorías donde se represente de manera clara cada uno de los ítems a tratar.

### **2.4.5 Obtención de resultados y conclusiones**

Por medio de las conclusiones, que no son más que afirmaciones de los conocimientos adquiridos en el desarrollo del trabajo del investigador en el contexto del problema estudiado, se puede llegar a la comparación con la actividad en la investigación llevada a cabo, determinando semejanzas y diferencias, dando lugar a nuevas conclusiones que aportan un valor significativo ya sea en la apertura para nuevos puntos de investigación referentes al tema principal o la conclusión de dicho trabajo. [13]

## **2.5 Parámetros de control**

Son 2 principales aspectos que controlar para la impresión de figuras 3D. El primero de ellos es el control sobre los motores a pasos que forman parte de la impresora y también la temperatura de extrusión del chocolate, ya que es una variable critica al momento de obtener el producto final.

### **2.5.1 Control de temperatura**

La materia prima utilizada para la impresión de alimentos en 3D es el chocolate, al realizar una investigación en la web, se ha determinado que la temperatura que debe estar la pasta

de chocolate puramente hecha de cacao para ser moldeado es de aproximadamente entre 32 a 35 °C. [14]

Para poder llevar el chocolate a la temperatura antes descrita, se hará uso de una resistencia eléctrica comercial, la cual se puede visualizar en la figura 2.1 y será implementada en la parte del hardware.



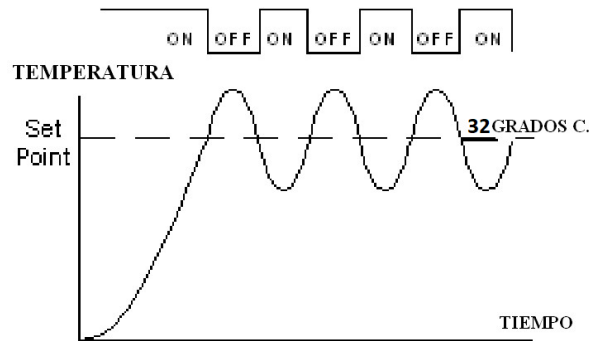
**Fig. 2.1.** Resistencia de calentamiento MITZU de 900W de potencia.

Esta resistencia comercial es de la marca MITZU de 900W y a pesar de que se la oferta en el mercado para calentar agua, en este proyecto será utilizada para calentar el chocolate. En la tabla 2.1 se muestran sus características eléctricas.[15]

**Tabla. 2.1.** Características físicas y eléctricas de resistencia de calentamiento MITZU.

Característica	Valor o cualidad
Tipo de resistencia	Resistencia calentadora de agua
Número de vueltas	4
Longitud de cable	0.75 m
Alimentación	127 V
Potencia	900 W
Corriente	7 A
Resistencia	18 $\Omega$

Para el control de temperatura de chocolate mediante esta resistencia se hará uso de un control on-off, es decir, que el sensor de temperatura implementado en el hardware cuando detecte que la temperatura del material es baja se encenderá la resistencia hasta que el chocolate llegue a la temperatura deseada, la cual es entre los 32 a 35 °C. Se establece como set point los 32 °C, tal como se lo visualiza en la figura 2.2.



**Fig. 2.2.** Control on-off para resistencia de calentamiento MITZU.

A continuación, se presenta el cálculo del tiempo que le tome a la resistencia llevar al chocolate de una temperatura ambiente hasta la temperatura deseada basado en los datos que se colocaron en la tabla previamente vista.

Para dicho calculo se comienza con el cálculo de la energía utilizada para calentar el chocolate mediante la ecuación 2.1 que relaciona la cantidad de calor que intercambia una masa de una cierta sustancia para llevar a la temperatura deseada.

$$Q = m * c * \Delta T \quad (2.1)$$

Donde  $Q$  es la cantidad de calor,  $m$  la masa de la sustancia, que según el tamaño del contenedor implementado en el hardware es de  $350 \text{ cm}^3$  que guarda una cantidad de  $290 \text{ g}$  de chocolate aproximadamente,  $c$  el calor específico de dicha sustancia (en este caso,  $c$  del chocolate es  $2300 \text{ cal/kg/}^\circ\text{C}$ ) y  $\Delta T$  la variación de temperatura que esta experimenta, partiendo de la temperatura de conservación  $15 \text{ }^\circ\text{C}$  a la que se almacena el chocolate, hasta la temperatura donde ya se lo puede moldear que es  $32^\circ\text{C}$ . De esta manera y reemplazando los datos en la ecuación 2.1, se tiene: [16]

$$Q = 0.225(2300)(32 - 15) = 879.75 \text{ cal}$$

Ahora, el tiempo en que le toma en calentarse con la resistencia implementada se lo hace con la ecuación 2.2.

$$Q = 0.24 * I^2 * R * t \quad (2.2)$$

Donde,  $Q$  es la energía,  $I$  la corriente,  $R$  la resistencia y  $t$  el tiempo que le toma al elemento final de control en llegar a la temperatura indicada.

Despejando el tiempo en la ecuación 2.2, se obtiene la siguiente expresión:

$$t = \frac{Q}{0.24 * I^2 * R} \quad (2.3)$$

Reemplazando los datos en la ecuación 2.3

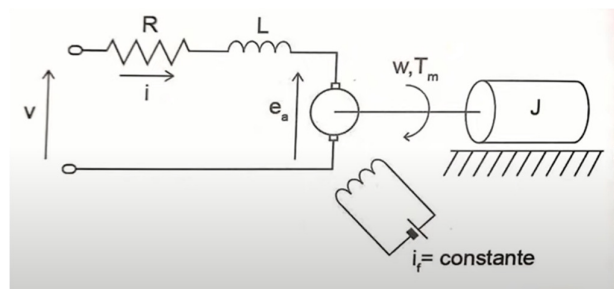
$$t = \frac{879.75}{0.24 * I^2 * R} = \frac{879.75}{0.24 * 7^2 * 18.14} = 2.47 \text{ seg}$$

## 2.5.2 Control motor DC

### 2.5.2.1 Modelo matemático del motor DC

La parte de hardware de la impresora 3D está constituida por 5 motores de corriente continua, en el cual se controlará la corriente, velocidad y posición de este para una correcta operación.

Para ello primero se determinará el modelo del motor a utilizar. En la siguiente imagen se muestra el modelo electromecánico el motor DC



**Fig. 2.3.** Modelo electromecánico del motor DC.

A partir de este modelo gráfico se realiza un análisis de mallas del circuito eléctrico para obtener su modelo matemático, partiendo del análisis de mallas en la parte eléctrica del modelo en mención se establece la ecuación 2.4.

$$v(t) = Ri(t) + L \frac{di(t)}{dt} + E_a(t) \quad (2.4)$$

Despejando el voltaje en el inductor de la ecuación 2.4 se obtiene:

$$L \frac{di(t)}{dt} = v(t) - Ri(t) - E_a(t) \quad (2.5)$$

Donde  $E_a(t)$  es un voltaje generado cuando los conductores de la armadura están en movimiento a través del flujo de campo producido por la corriente de campo representada como  $i_f$ .

Luego en la sección mecánica se realiza su modelado representado en la ecuación 2.6.

$$T_m(t) = J \frac{d\omega(t)}{dt} + B\omega(t) \quad (2.6)$$

Donde  $T_m(t)$  es el torque del motor,  $B$  es el coeficiente de fricción que existen entre el motor y la carga que está colocada en su eje,  $J$  es el momento de inercia del rotor y la carga con relación al eje del motor y  $\omega(t)$  es la velocidad angular a la que gira el eje del motor.

De la ecuación 2.6, si se despeja el momento de inercia para representar la dinámica del movimiento se obtiene:

$$J \frac{d\omega(t)}{dt} = T_m(t) - B\omega(t) \quad (2.7)$$

Por último, se asume que existe una relación proporcional entre el voltaje inducido en la armadura y la velocidad angular del eje del motor, la cual se representa como  $K_a$  y una relación electromecánica proporcional entre el torque y la corriente, denotada como  $K_m$ , tal como se lo hace en las siguientes 2 ecuaciones:

$$E_a(t) = K_a\omega(t) \quad (2.8)$$

$$T_m(t) = K_m i(t) \quad (2.9)$$

Aplicando la transformada de Laplace en las ecuaciones 2.4 y 2.5, se tiene las ecuaciones 2.10 y 2.11:

$$Lsi(s) = v(s) - Ri(s) - E_a(s) \quad (2.10)$$

$$E_a(s) = K_a\omega(s) \quad (2.11)$$

Mientras que de las ecuaciones 2.6 y 2.7 al aplicar la transformada de Laplace se tiene las ecuaciones 2.12 y 2.13.

$$Js\omega(s) = T_m(s) - B\omega(s) \quad (2.12)$$

$$T_m(s) = K_{mi}i(s) \quad (2.13)$$

Sustituyendo las ecuaciones 2.11 y 2.13 en la ecuación 2.10, se tiene:

$$Ls \frac{T_m(s)}{K_m} = v(s) - R \frac{T_m(s)}{K_m} - K_a\omega(s) \quad (2.14)$$

Despejando el voltaje en la ecuación 2.14:

$$v(s) = \frac{(R + Ls)T_m(s)}{K_m} + K_a\omega(s) \quad (2.15)$$

Luego de la ecuación 2.12 se despeja la velocidad angular:

$$\omega(s) = \frac{T_m(s)}{Js + B} \quad (2.16)$$

Y reemplazando la ecuación 2.16 en la ecuación 2.14 se obtiene:

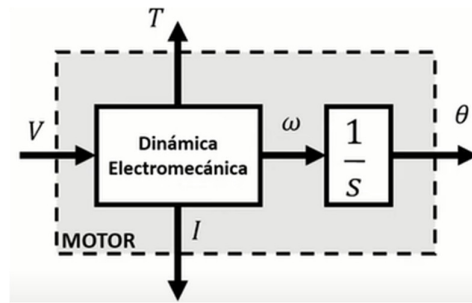
$$v(s) = \frac{(R + Ls)T_m(s)}{K_m} + K_a \frac{T_m(s)}{Js + B} \quad (2.17)$$

Finalmente, para obtener la función de transferencia de relación entre la salida que es el torque y la entrada el voltaje se lo hace mediante despejes en la ecuación 2.17 y se obtiene dicha función de transferencia tal como se observa en la ecuación 2.18.

$$\frac{T_m(s)}{v(s)} = \frac{K_m(Js + B)}{LJs^2 + (RJ + LB)s + RB + K_mK_a} \quad (2.18)$$

La descripción grafica del motor DC en diagrama de bloques, se observa en la figura 2.4 donde el bloque que se observa como su dinámica electromecánica es la representación del motor alimentado por un voltaje y a la salida de este se tienen diferentes variables como la corriente de armadura, el torque generado por el motor y la velocidad angular, además si esta última variable es integrada, se obtiene la posición angular





**Fig. 2.4.** Diagrama de bloques de variables de interés del motor DC.

### 2.5.2.2 Parámetros del motor DC

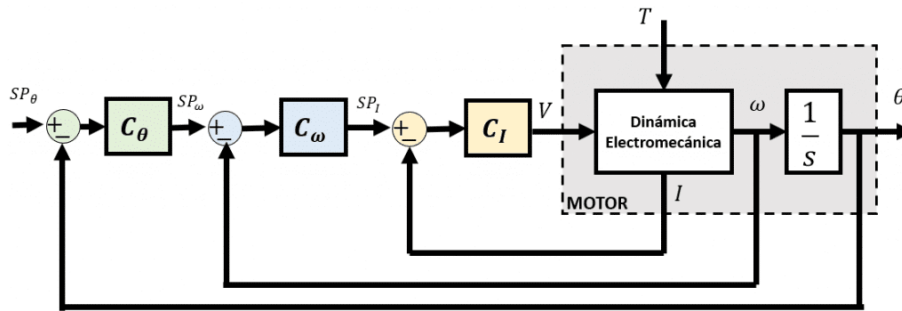
El motor para implementar es el NEMA size 17 1.8° 2-phase stepper motor de la marca Schneider Electric, del cual en la tabla 2.2 se visualiza los datos más relevantes y necesarios para realizar la sintonización del controlador de los motores. [18]

**Tabla. 2.2.** Características físicas y eléctricas de motor DC NEMA size 17.

Parámetro	Valor
Momento de inercia (J)	0.03 kgm <sup>2</sup> /s <sup>2</sup>
Coefficiente de fricción (B)	0.15 Nm
Resistencia (R)	2.1 Ω
Inductancia (L)	0.05 H
Relación entre voltaje inducido en la armadura y la velocidad angular del eje (Ka)	0.01
Relación entre el torque mecánico y la corriente eléctrica (Km)	0.01

### 2.5.2.3 Control en cascada

Para el control del motor DC se hará uso de un control realimentado en cascada dado que para la movilidad de las partes físicas de la impresora se necesita mucha precisión en la posición de ellas y debido a que se tiene como variables secundarias la corriente de armadura y la velocidad angular que también se las puede controlar y si se logra cumplir este objetivo, se podría realizar un control más eficiente en la variable principal que es la posición angular del motor de corriente directa



**Fig. 2.5.** Diagrama de bloques control en cascada motor DC.

Las variables mostradas en la figura 2.5 son:  $V$ : voltaje,  $I$ : corriente,  $\omega$ : velocidad angular,  $\vartheta$ : posición angular,  $T$ : torque de carga sobre el motor.

La primera variable que controlar es la corriente, la cual es muy rápida, dándose cambios en ella en cuestión de milisegundos, por ello esta se encuentra en el lazo más interno y su control se da por el controlador  $C_I$  que tiene la funcionalidad de rechazar cualquier perturbación que exista a nivel de corriente en la armadura. Seguido a ello se hace la medición de la velocidad angular y de igual manera que con la corriente, se coloca un controlador,  $C_\omega$ , capaz de rechazar las perturbaciones que pueda sufrir en velocidad teniendo en consideración la dinámica interna de la corriente que fue previamente controlada. Y por último se tiene la variable de interés que es la posición, la cual más interesa controlar y se encuentra en el lazo más externo, controlada por  $C_\vartheta$  y llevando el motor a la posición deseada y tomando en cuenta la dinámica de las otras variables que son más rápidas que ya están siendo controladas. [19]

#### 2.5.2.4. Implementación en Simulink

Para realizar la implementación en Simulink y simular, se hará uso de la transformación a espacio de estados, tomando como estados la velocidad angular del motor y la corriente de armadura que circula por el circuito, de la siguiente manera:

$$\begin{aligned} x_1 &= \omega & \dot{x}_1 &= \dot{\omega} \\ x_2 &= i & \dot{x}_2 &= \dot{i} \end{aligned}$$

Y se sustituirá en las ecuaciones diferenciales, previamente establecidas:

$$\dot{x}_1 = -\frac{B}{J}x_1 + \frac{K_m}{J}x_2$$

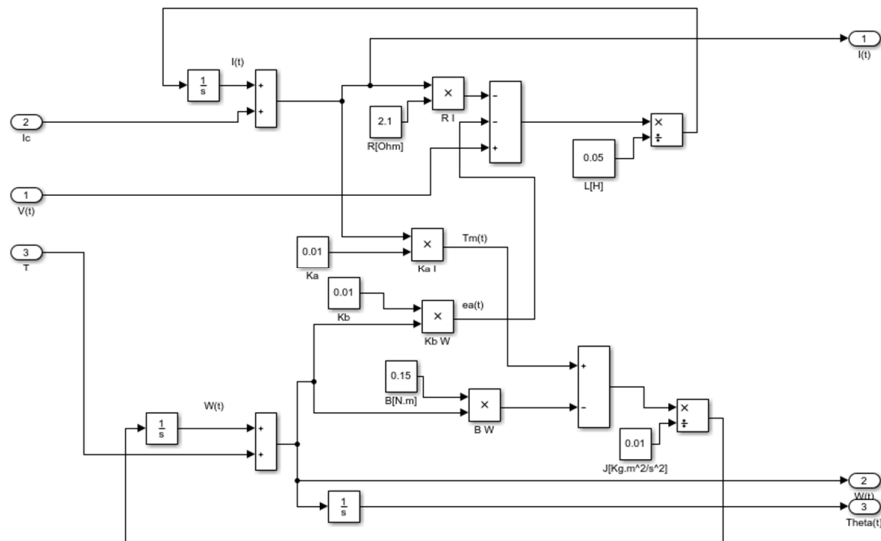
$$\dot{x}_2 = -\frac{R}{L}x_2 - \frac{K_a}{L}x_1 + \frac{1}{L}v$$

Y ya representado en matriz de espacio de estados:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -\frac{B}{J} & \frac{K_m}{J} \\ -\frac{K_a}{L} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix} v$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Por lo cual, en la figura 2.6 se hace la implementación del motor en Simulink



**Fig. 2.6.** Implementación en Simulink mediante variables de estado del motor DC.

Donde en la entrada;  $V(t)$  es el voltaje inyectado al motor,  $I_c$  es una perturbación en la corriente de armadura y  $T$  es el torque de carga. En cambio, a la salida tenemos las variables medidas que son  $I(t)$  corriente en la armadura,  $\omega(t)$  la velocidad angular del motor y  $\theta(t)$  su posición angular.

### 2.5.2.5 Diseño de los controladores

Para el diseño de cada uno de los controladores que forman parte del lazo realimentado en cascada se partirá desde la sintonización del control PID del lazo más interno, es decir, el de corriente, luego el de velocidad angular y finalmente el de posición. Se hará uso del autosintonizador que posee el bloque PID Controller de Simulink y un ajuste fino para poder llegar a los parámetros deseados para el control de las diferentes variables del motor. La herramienta antes mencionada tiene el nombre de PID Tuner que brinda un método para el ajuste de PID, logrando de esta manera un diseño robusto con el tiempo de respuesta que se desee. En las figuras 2.7 y 2.8 se muestra las opciones dentro de dicho bloque que permite la autosintonización y el cuadro de dialogo que permite realizar el ajuste fino de acuerdo con los parámetros deseados, respectivamente. [20]

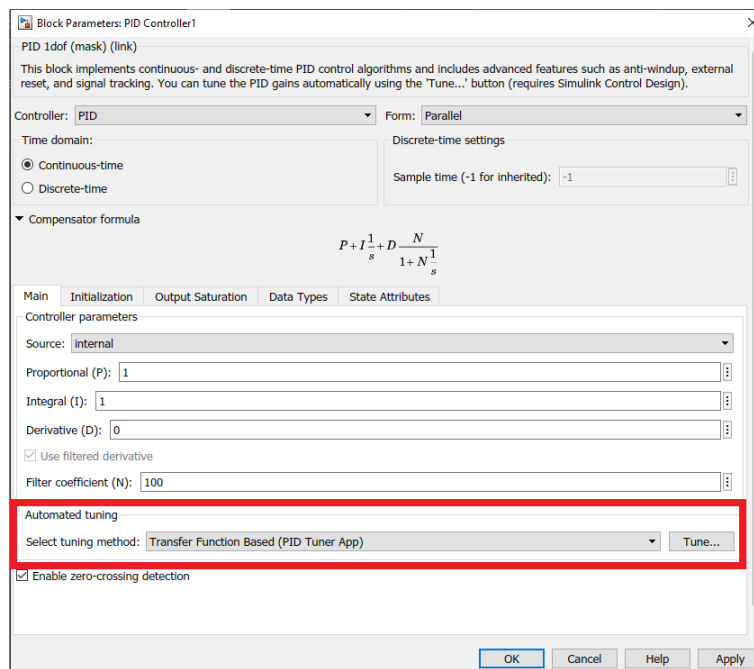
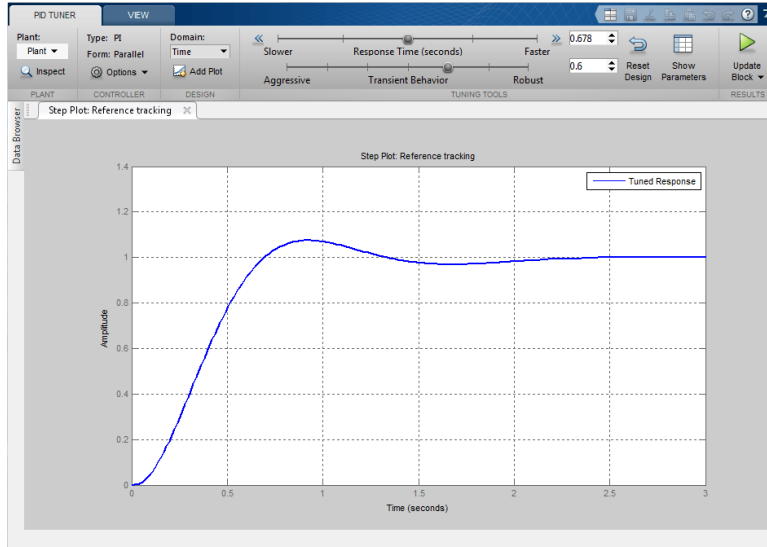


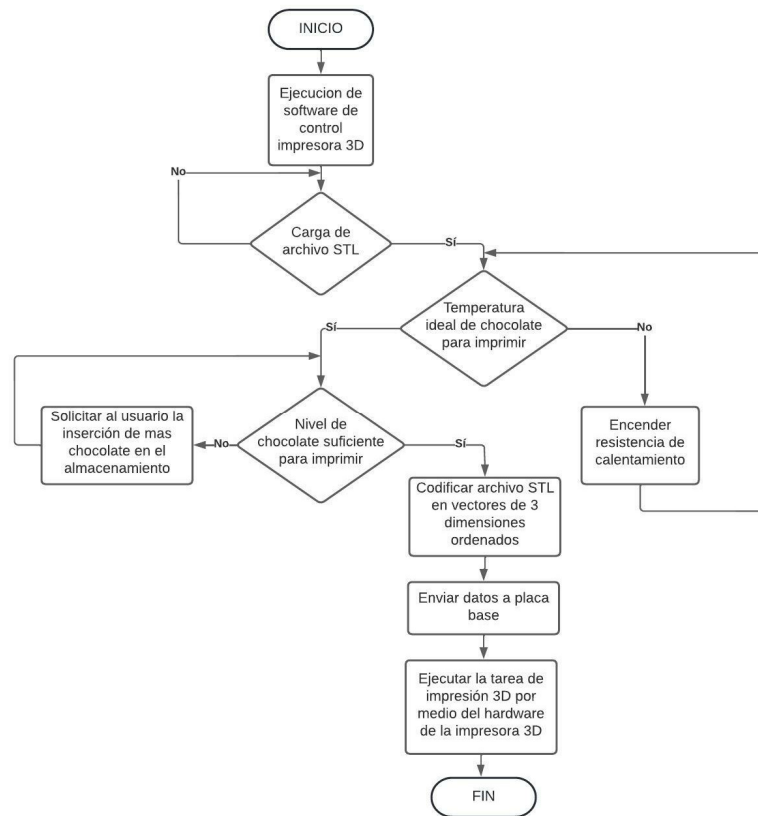
Fig. 2.7. Subventana de configuración de parámetros P, I, D del controlador.



**Fig. 2.8.** Subventana de ajuste de variables mediante grafico de respuesta del sistema.

## 2.6 Diseño HMI

La elaboración del HMI se lo ha realizado en el componente App Designer del software Matlab, este contiene varios aspectos de visualización y control para mantener al usuario al tanto del estado actual del proceso de impresión, las variables que intervienen durante dicho proceso y la carga de archivos STL, los cuales incluyen el modelo de la figura tridimensional que será impresa en pasta de chocolate. Para lograr el objetivo de imprimir objetos tridimensionales con pasta de chocolate se ha planteado la secuencia mostrada en el diagrama de flujo de la figura 2.9.



**Fig. 2.9.** Diagrama de flujo de funcionamiento del software diseñado.

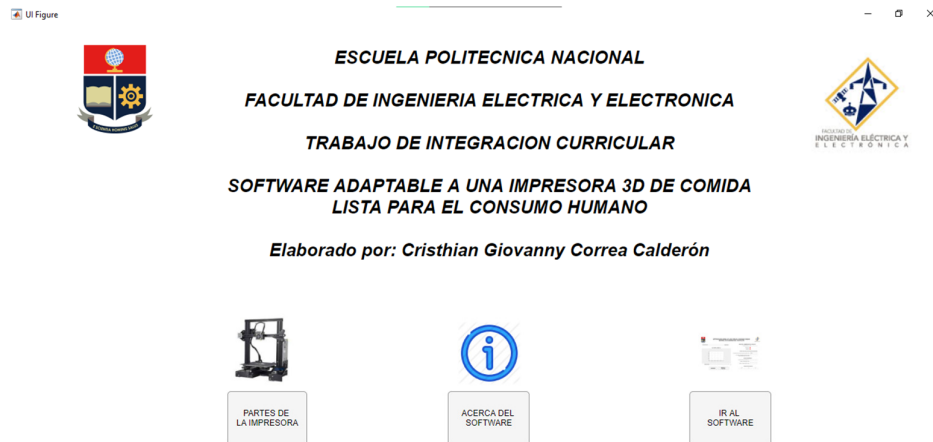
Los aspectos que contiene la interfaz se detallan a continuación:

### 2.6.1 Ventana de presentación

En la primera ventana que se tiene al ejecutar el programa es la presentación de este proyecto, haciendo uso de las herramientas de diseño de App Designer se ha colocado texto, imágenes y 3 botones en la interfaz que hacen la apertura de otras ventanas. Estos 3 botones son:

- **Partes de la impresora:** Muestra la información de los componentes de hardware más relevantes de la impresora 3D.
- **Acerca del software:** Hace la apertura de una ventana informativa con detalle de las características visuales que compone el software y los campos de visualización que posee para un seguimiento del comportamiento de la parte física.
- **Ir al software:** Este botón ejecuta el software de control como tal, donde se tiene las opciones de carga de archivos y la visualización de las variables que conforman el estado de la impresión.

En la figura 2.10 se tiene la ventana de presentación



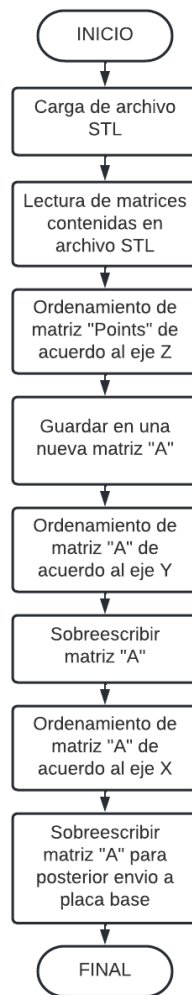
**Fig. 2.10.** Ventana de presentación de software diseñado.

## 2.6.2 Ventana de software de control

En esta ventana se tratan diferentes aspectos para la obtención del producto final siendo el punto de partida el poder cargar el modelo 3D que previamente se ha diseñado en cualquier software destinado para ello, la validación de parámetros que abarcan el inicio de la impresión y la visualización del estado de la impresión en tiempo real. A continuación, se hace un detalle más profundo y la forma en que se ha logrado dicha acción.

### 2.6.2.1 Carga y manejo de archivos STL

En la ejecución de esta tarea se sigue un orden desde que se carga el archivo hasta que se envíe las coordenadas para ser interpretada por la placa base de la parte física de la impresora 3D de comida y que se realice la elaboración de la figura 3D de pasta de chocolate. Por otra parte, se muestra una visualización 3D del objeto mediante el proyector mallado tridimensional que posee Matlab y las dimensiones del modelo. El proceso por seguir es el mostrado en el diagrama de flujo de la figura 2.11, donde la matriz "A" es donde se almacenará la información final de los vectores y en la figura 2.12 se observa el visualizador de figuras 3D incorporado en el software diseñado



**Fig. 2.11.** Diagrama de flujo de manejo de archivos STL.





**Fig. 2.12.** Sitio de carga y visualización archivos STL en ventana de control.

Para esta función se ha hecho uso del comando *uigetfile* para poder aperturar una ventana emergente para la navegación por los archivos guardados en el ordenador del usuario que este haciendo uso del software y posteriormente el comando *stlread* de Matlab para poder leer los archivos de extensión STL que contienen las coordenadas de los puntos del archivo 3D para que el inyector de chocolate pueda realizar la instalación.

Estos archivos STL al ser importados a Matlab se lee como 2 matrices de orden  $m \times 3$ , es decir,  $m$  vectores tridimensionales. Estos datos son una matriz de puntos llamada *Points* y otra matriz que contiene la conectividad entre los triángulos que forman parte del modelo 3D llamado *Connectivity List*. La primera matriz mencionada es la que se utilizará para poder enviar la orden de movimiento hacia los motores, se hace un recorrido por la matriz fila a fila, donde se extraen los valores de los vectores de 3 dimensiones y se transmite a la placa base donde esta interpretará la locación hacia donde el inyector ira haciendo la extrusión del chocolate en pasta en su recorrido. Cabe mencionar que esta matriz, por defecto Matlab lo lee de manera desordenada y mediante el comando *sortrows* se realiza la organización de los vectores en función de la tercera columna que abarca los valores del eje Z, haciendo que estos tengan un orden ascendente en función de dicho eje, dado que la impresión va desde la parte inferior hasta la parte superior. Luego se procederá a organizar los vectores en función de los demás ejes coordenados.

Una muestra del código de presentación para la importación del archivo en formato stl y el ordenamiento de los vectores del eje Z se muestra a continuación:

```
[file, pathToFile] = uigetfile('*.stl','Seleccione un archivo con extension STL');
%Busqueda del archivo stl en el ordenador
FileName = strcat(pathToFile,file);
imagen3d = stlread(strcat(pathToFile, file)); %lectura del archivo STL
orden_seguimiento = sortrows(seguimiento,3) %Ordenamiento de vectores acorde al eje Z
```

Y al momento de ejecutar el procedimiento de impresión se tendrá un campo de visualización en la interfaz, como se lo aprecia en la figura 2.13, que permita ver donde se encuentra el extrusor realizando la tarea de evacuación de pasta de chocolate.

**ORIENTACION EXTRUSOR**

<b>X</b>	<b>Y</b>	<b>Z</b>
0	0	0

**Fig. 2.13.** Sitio de visualización de locación del extrusor.

### 2.6.2.2 Validación de temperatura de chocolate

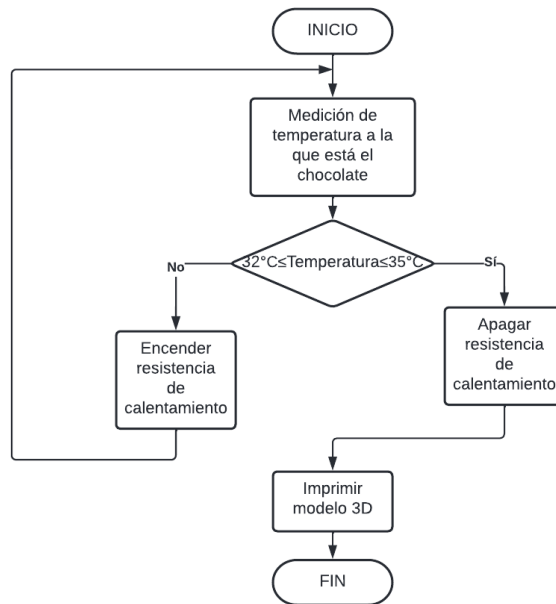
Como se ha mencionado anteriormente, la temperatura óptima para que la pasta de chocolate sea moldeable es de entre 32 a 35 °C, esta es validada por el sensor que forma parte del hardware de la impresora 3D, donde si esta variable se encuentra en el valor adecuado, el proceso de impresión se llevara a cabo, caso contrario se encenderá la resistencia de calentamiento hasta que la temperatura este en la temperatura adecuada, se podrá iniciar la impresión. En la figura 2.13 se observa el visualizador incorporado en el software diseño, mientras que la figura 2.14 muestra la lógica de validación de temperatura mediante un diagrama de flujo.

**ESTADO DE LA TEMPERATURA DEL CHOCOLATE**

T° CHOCOLATE [°C]

RESISTENCIA DE CALENTAMIENTO ●

**Fig. 2.13.** Sitio de visualización de estado de temperatura de chocolate.



**Fig. 2.14.** Diagrama de flujo de gestión de validación de temperatura de chocolate.

### 2.6.2.3 Validación de cantidad de chocolate en almacenamiento

Otra variable crítica para realizar la impresión del modelo 3D es conocer que cantidad de material hay disponible para que pueda ser impreso el objeto. En el hardware de la impresora se tiene un recipiente que almacena hasta 350 cm<sup>3</sup> (3500000 mm<sup>3</sup>) de pasta de chocolate, la cual es una cantidad considerable para poder lograr la impresión de los modelos que se ha definido como objetivo de materializarlos en este proyecto.

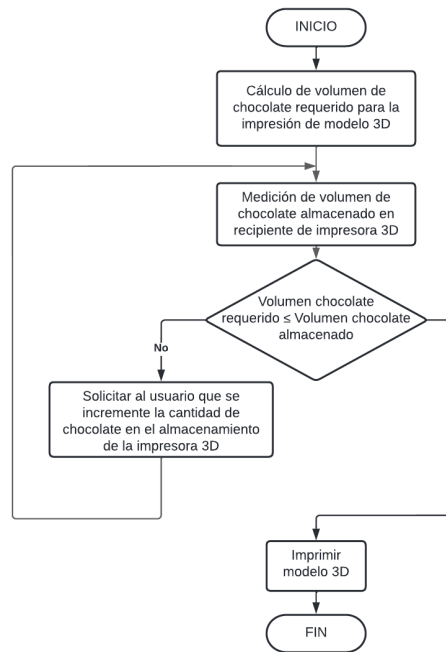
Además de eso, se hace un cálculo de cuanta cantidad de chocolate necesita el modelo 3D que se está deseando imprimir para realizar la comparación con el volumen de chocolate que se dispone en el almacenamiento de la impresora y si este bastara para poder imprimir completamente el objeto deseado, caso contrario se solicitara que se coloque tanta cantidad de pasta de chocolate como sea suficiente para realizar la tarea de impresión 3D. Al igual que en el apartado anterior donde se validaba la temperatura, en esta ocasión, también se incluye un visualizador de esta variable en el software diseñado, tal como se observa en la figura 2.15 y para ello se hace la validación de esta como se observa en el diagrama de flujo de la figura 2.16.

**ESTADO DEL NIVEL DE CHOCOLATE EN EXTRUSOR**

VOLUMEN DE CHOCOLATE EN EXTRUSOR (mm<sup>3</sup>)

VOLUMEN APROXIMADO QUE SE USARA PARA IMPRESION ACTUAL (mm<sup>3</sup>)

**Fig. 2.15.** Sitio de visualización de estado de nivel de cantidad de chocolate.



**Fig. 2.16.** Diagrama de flujo de gestión de validación de nivel de chocolate almacenado.

#### 2.6.2.4. Estado de impresión

En estos campos de visualización se tiene 2 datos relacionados a la situación de impresión en tiempo real como son el porcentaje de impresión y el tiempo que ha transcurrido en segundos de esta tarea. Adicional a ello se tiene un campo que muestra el tiempo aproximado en segundos que tardara en realizarse la impresión, esto está basado en los componentes físicos de la impresora 3D y la velocidad a la que actúan los motores de la extrusora y demás partes móviles del dispositivo. Este cálculo se verá contrarrestado en la muestra de los resultados que se observa en el siguiente capítulo de este trabajo.

**ESTADO DE IMPRESION**

TIEMPO APROXIMADO (seg)

PORCENTAJE DE IMPRESION (%)

TIEMPO TRANSCURRIDO (seg)

**Fig. 2.17.** Sitio de visualización de estado de impresión en tiempo real.

### 2.6.2.5 Botones de interacción

Dado que esta aplicación está destinada para la impresión de modelos 3D alojados en un archivo de formato stl, solamente se ha colocado el botón que permita ejecutar la impresión mediante el botón *IMPRIMIR*, adicionalmente se ha agregado otro botón que permita regresar al menú anterior llamado *MENU PRINCIPAL* en la que se encuentra la información previamente descrita.



**Fig. 2.18.** Botones en ventana de control.

### 2.6.2.6 Mensajes de error y advertencia






Durante la ejecución del software desde la carga del archivo stl hasta la obtención del producto final se muestran ciertos mensajes en diferentes circunstancias haciendo uso del comando *uialert* de Matlab para desplegar ventanas de alerta que permitan al usuario conocer el estado actual de la impresión, si ha ocurrido algún error en la ejecución de las tareas o si se ha producido una advertencia que debe ser corregida para que el procedimiento de impresión se siga realizando con normalidad.

El comando *uialert* tiene el siguiente formato:

```
uialert(fig,message,title)
```

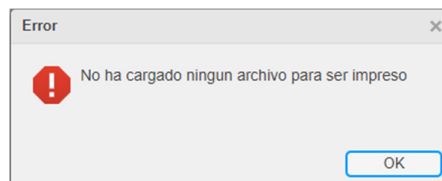
Donde *fig* hace referencia al icono que se va a mostrar dependiendo de la característica del mensaje que se desplegará según la tabla 2.3, *message* será el mensaje de aviso y *title* el título del cuadro de texto emergente. [21]

**Tabla. 2.3.** Tipos de mensajes e iconos para anuncios emergentes en App Designer.

VALOR	ICONO
error	
advertencia	
información	
pregunta	
cumplido	

A continuación, se presentan y describen algunos de estos anuncios emergentes que se ha colocado a lo largo de la ejecución del software y cuando estos se muestran al usuario

- Error al presionar imprimir sin haber cargado un archivo stl previamente, lo cual sugiere al usuario que se suba un modelo 3D para que pueda ser impreso.



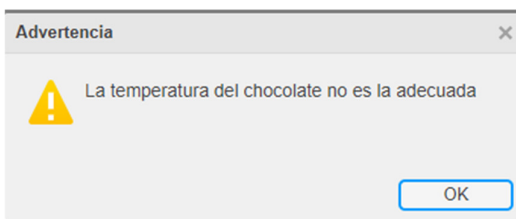
**Fig. 2.19.** Anuncio emergente de error.

- Información de estado en tiempo real al subir un archivo stl al software.



**Fig. 2.20.** Anuncio emergente de progreso de acción.

- Advertencia cuando la temperatura del chocolate no es la adecuada para comenzar la impresión consecuentemente al mostrarse este cuadro de dialogo se enviará a encender la resistencia de calentamiento que haga elevar la temperatura de la pasta de chocolate

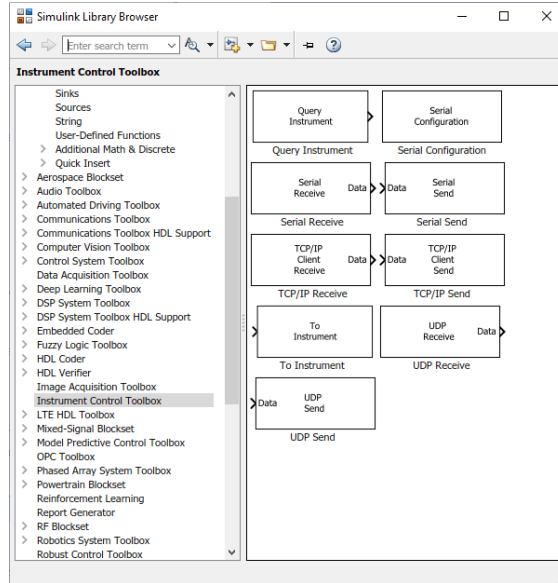


**Fig. 2.21.** Anuncio emergente de advertencia.

## **2.7 Comunicación con placa base**

Para poder hacer el envío y recepción de datos hacia y desde la placa base, respectivamente, se hará uso de comunicación serial entre el computador o dispositivo que se esté ejecutando el software de control de la impresora 3D de comida y la placa base del hardware. En la investigación y el desarrollo del complemento de este trabajo de integración curricular, como lo es el hardware de la impresora 3D, se implementa como placa base la Geeetech GT2560 que está basada su construcción en un microcontrolador de la familia de procesadores que se ha estado manejando en algunas materias de la carrera universitaria como lo es el ATmega2560. Al ser esta placa ya destinada para el control de impresoras 3D posee el firmware que interpreta las coordenadas en 3 ejes de cada punto para hacer que el extrusor se mueva hasta cada punto para realizar la extrusión del chocolate.

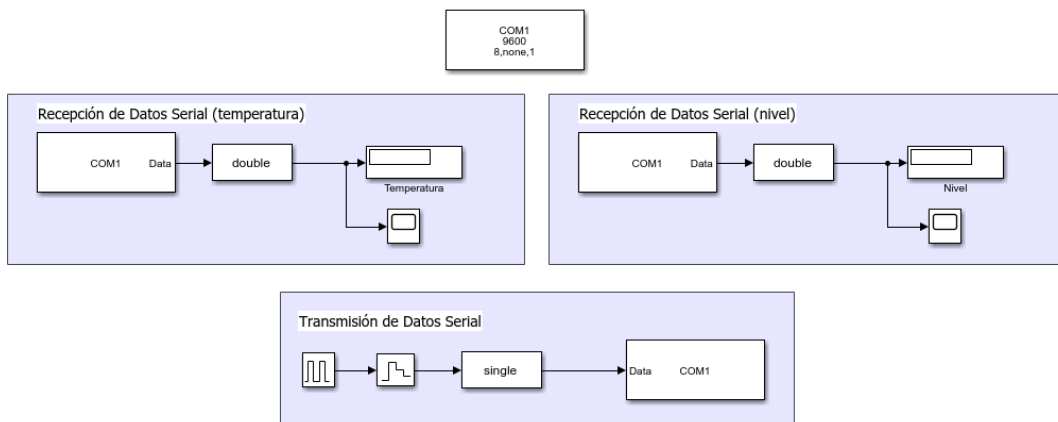
Debido a que la aplicación que se ha desarrollado se encuentra hecha mediante Matlab, se hará uso de Simulink y la librería Instrument Control Toolbox para poder realizar la comunicación serial entre la computadora donde se aloja el software y la placa base. Una vez que se tenga abierto la aplicación de Simulink, en su librería se encontrara dicha biblioteca con los bloques necesarios para esta comunicación, como se muestra en la figura siguiente.



**Fig. 2.22.** Librería de Simulink.

Cabe recalcar que Matlab trabaja con números de punto flotante de precisión doble por defecto que utiliza 8 bytes de espacio de memoria, mientras que el microcontrolador de la placa a ser utilizada para el hardware de la impresora 3D ocupa 4 bytes de memoria.

A través de los bloques que tiene la librería de Matlab se ha implementado los diagramas de bloques mostrados en la figura 2.23 que realizan la recepción y transmisión de datos.

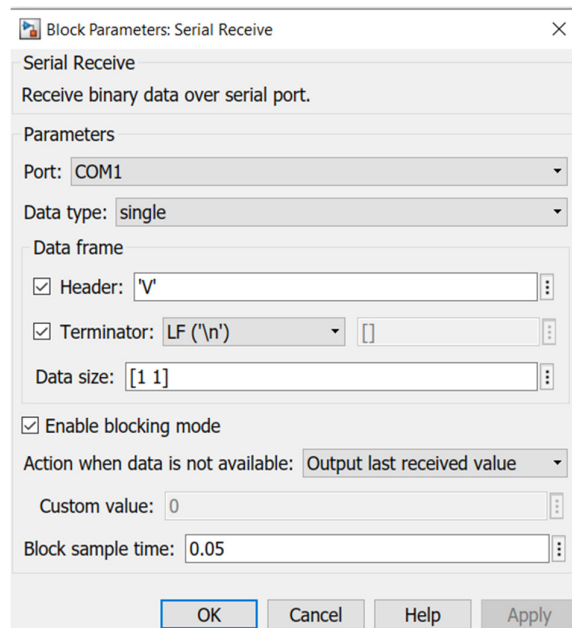


**Fig. 2.23.** Diagrama de bloques de configuración para comunicación serial.



### 2.7.1 Recepción de datos

En la sección de la recepción de señal se tiene el bloque Data Type Conversion que se encarga de realizar la conversión de tipos de datos entre Simulink y la placa base como se ha mencionado previamente que se lo deberá realizar. Además, se coloca el bloque Serial Receive configurado como se observa en la figura 2.24, donde el campo Data type se coloca el valor de Single dado la naturaleza de los datos de recepción, en la cabecera de la trama de datos se ha puesto una 'V' haciendo referencia al carácter inicial y como finalización de la trama se ha puesto el símbolo '\n', por último, un valor de 0.05s para el tiempo de muestreo.



**Fig. 2.24.** Subventana de configuración del bloque para la recepción de datos.

### 2.7.2. Transmisión de datos

En cambio, en la sección de transmisión de datos, el bloque Pulse generator se configura con un periodo de 4 s para el tren de pulso y un ancho del 50%, el bloque Zero-Order Hold se ocupa de la discretización de datos con periodo de muestreo de 0.05 s, el bloque Data Type Conversion realiza la conversión de datos de tipo double manejado por Matlab a de tipo single que interpreta la placa base y el bloque Serial send se configura con los parámetros de la placa base para realizar la comunicación serial. [22]

## 2.8 Archivo ejecutable

Una vez que ya se tenga todos los archivos o componentes que forman parte del software de control de la impresora 3D que han sido creados ya sea en App Designer, Simulink o un script de Matlab se lo empaqueta en un solo archivo que puede ser ejecutado en cualquier computador y pueda ser manejado fácilmente. Para obtener este archivo se hace uso de la opción Package App Toolstrip que posee Matlab. Además, a través de este componente mencionado se puede dar ciertos retoques a la aplicación como la ventana de presentación, nombre de la ventana que se presentara al ejecutar el servicio, su icono, entre otras características que harán de la aplicación creada se vea como profesional. En la siguiente ventana, se muestra la ventana de este componente donde se visualiza las diferentes características que le dan este retoque ya mencionado [23].

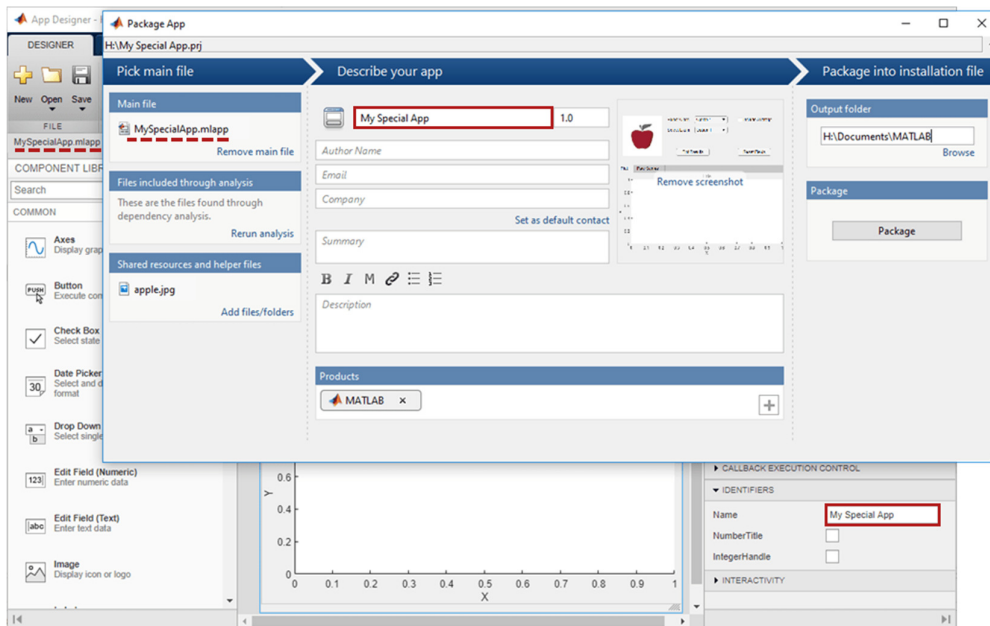


Fig. 2.25. Ventana de conversión de proyecto en App Designer a archivo ejecutable.

### 3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

#### 3.1 Resultados

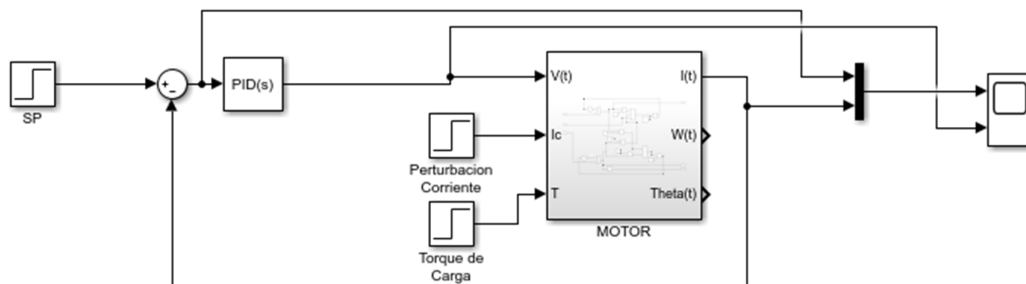
##### 3.1.2 Sintonización de controladores

Como se mencionó en el capítulo anterior, se hizo uso del autosintonizador de Matlab y posteriormente se dio un ajuste fino para poder obtener el mejor resultado sin tener que hacer un esfuerzo mayor para llevarlo hasta el punto tomado como referencia.

En control hecho es en cascada, controlando así del lazo más interno, el cual posee la variable más rápida, hacia el lazo más externo con la variable más lenta. A continuación, se verifica los resultados obtenidos de realizar el control sobre cada variable:

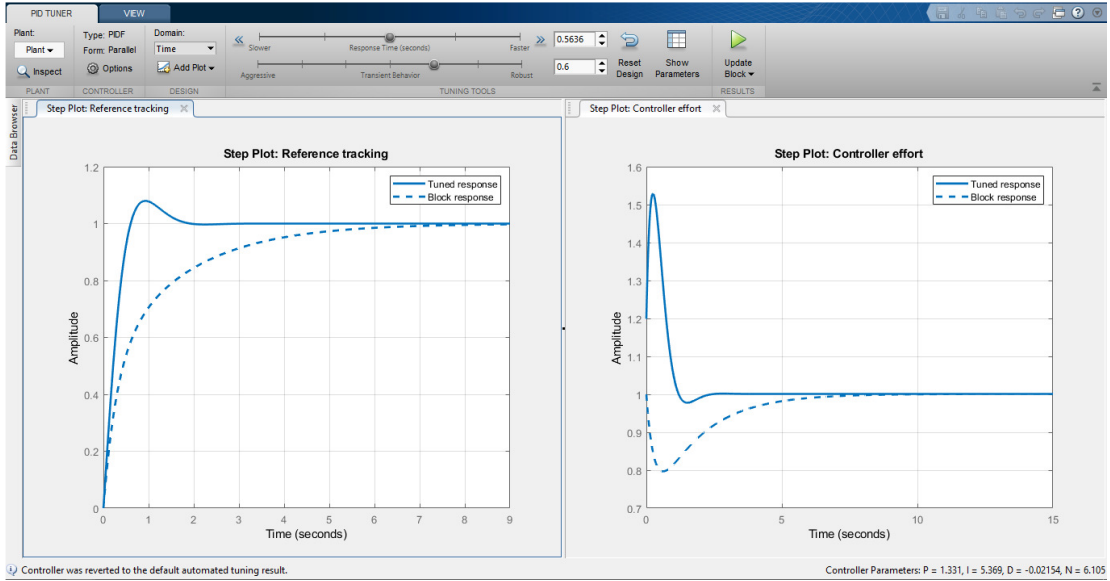
##### 3.1.2.1 Respuesta del control del lazo de corriente

En la siguiente imagen se arma el diagrama de bloques con la realimentación de corriente y el controlador, además se añade perturbaciones para poder verificar que el controlador está actuando sobre el motor.

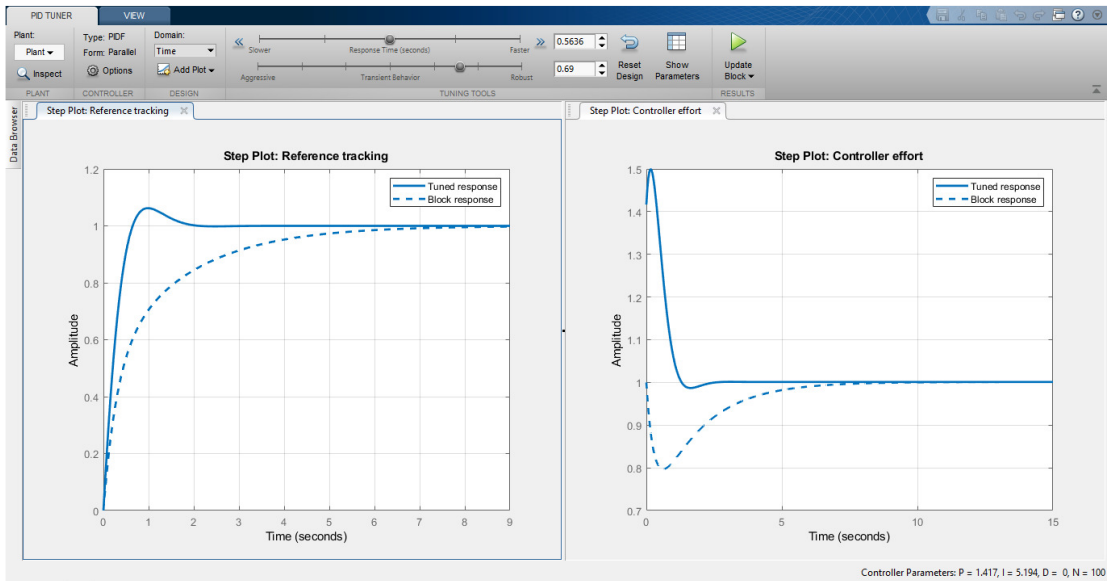


**Fig. 3.1.** Diagrama de bloques en Simulink del lazo de corriente.

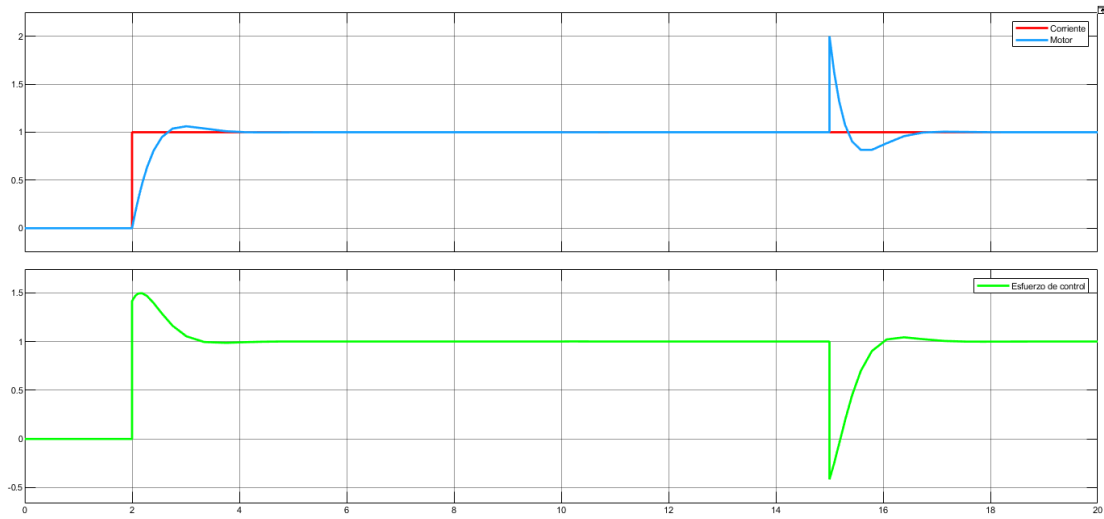
En las figuras 3.2 se aprecia la respuesta de la variable con el controlador PID incorporado al utilizar el autosintonizador de Simulink, en la figura 3.3 se visualiza la respuesta del controlador PID, pero con un ajuste más fino utilizando las opciones de tiempo de respuesta y comportamiento transitorio, añadiendo en ellas también el esfuerzo realizado por el controlador para cumplir su objetivo. Finalmente, se muestra la respuesta del control de corriente ante perturbaciones y cambio de referencia en la figura 3.4.



**Fig. 3.2.** Respuesta del control del lazo de corriente mediante el autosintonizador.



**Fig. 3.3.** Respuesta del control del lazo de corriente mediante al realizar un ajuste fino.



**Fig. 3.4.** Respuesta del control del lazo de corriente ante cambios de referencia y perturbaciones.

En la tabla 3.1 se muestran los valores de P, I y D para el controlador con cada una de las técnicas utilizadas.

**Tabla. 3.1.** Valores P, I y D para el controlador de corriente.

Parte	Autosintonizador	Ajuste fino
Proporcional (P)	1.331	1.417
Integral (I)	5.369	5.194
Derivativa (D)	-0.02154	0

### 3.1.2.2 Respuesta del control del lazo de velocidad

De igual manera, se hace la realimentación para la variable velocidad tomando en cuenta la dinámica de la corriente que previamente se analizó y ya fue controlado. En este caso, el diagrama de bloques se lo visualiza en la figura 3.5 y en las figuras 3.6, 3.7 y 3.8 se visualizan las respuestas del controlador al autosintonizar, luego al darle un ajuste fino y en la última la respuesta ante cambios de referencia y perturbaciones.

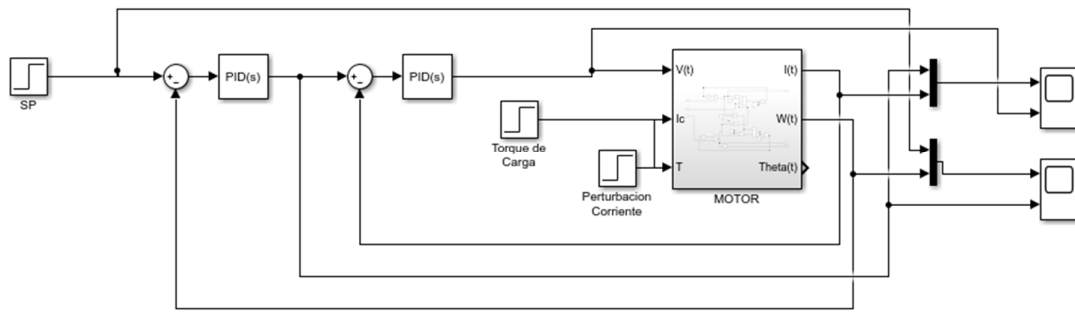


Fig. 3.5. Diagrama de bloques en Simulink del lazo de velocidad.

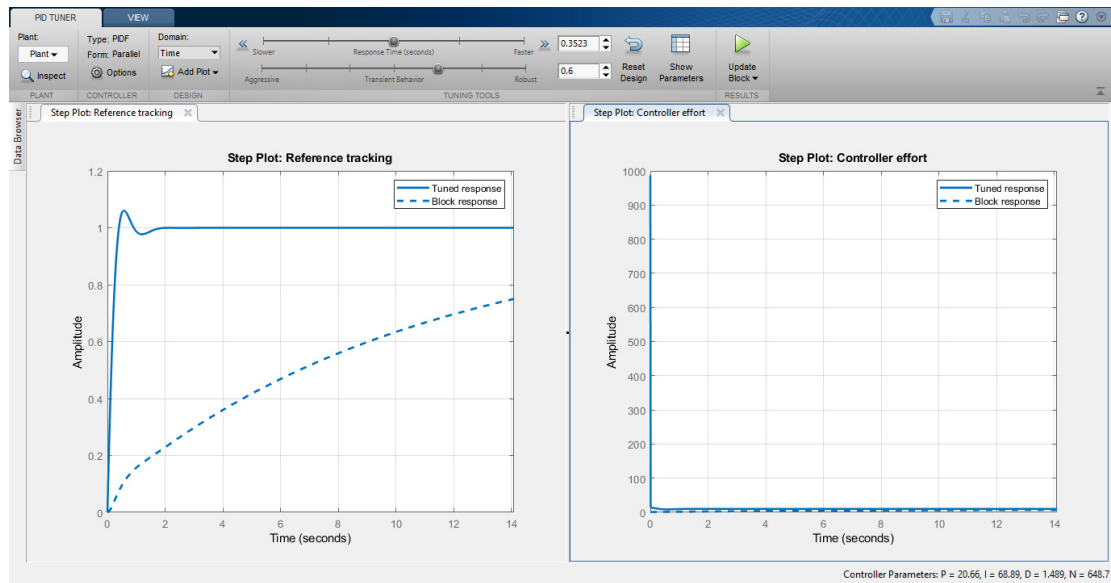
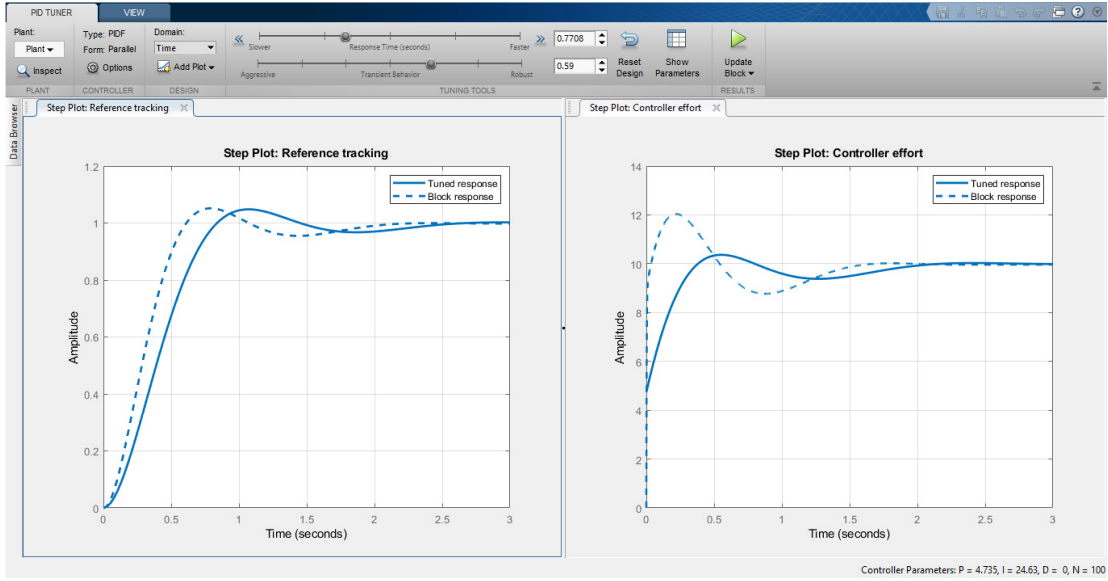
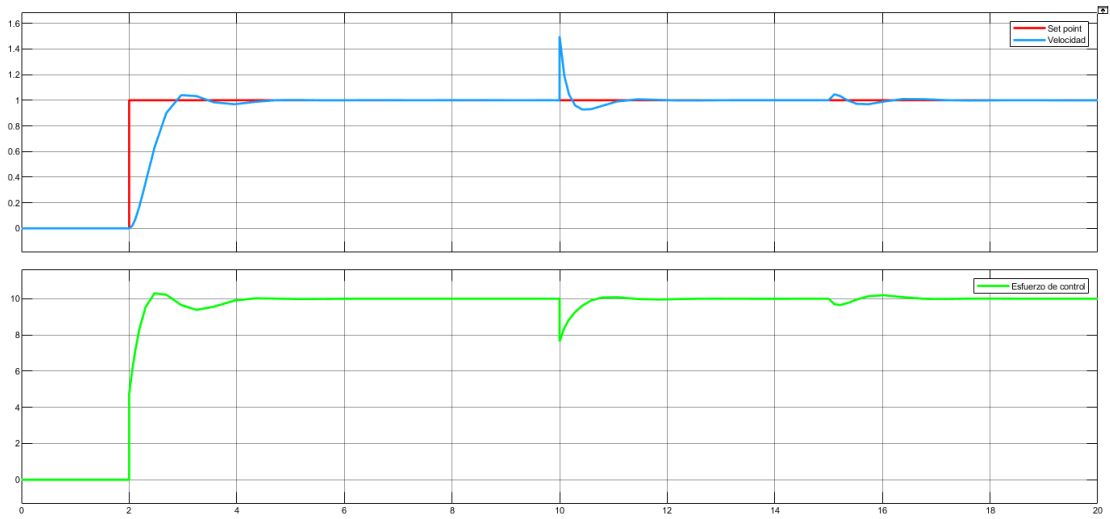


Fig. 3.6. Respuesta del control del lazo de velocidad mediante el autosintonizador.



**Fig. 3.7.** Respuesta del control del lazo de velocidad mediante al realizar un ajuste fino.



**Fig. 3.8.** Respuesta del control del lazo de corriente ante cambios de referencia y perturbaciones.

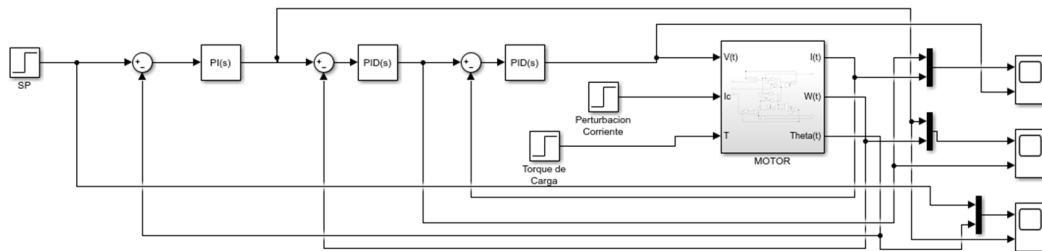
**Tabla. 3.2.** Valores P, I y D para el controlador de velocidad.

Parte	Autosintonizador	Ajuste fino
Proporcional (P)	20.66	4.375
Integral (I)	68.89	24.63
Derivativa (D)	1.489	0

A diferencia del diseño del controlador de corriente, en el controlador de la velocidad se observa un cambio significativo de los valores al realizar un ajuste fino, esto ha sido producto de buscar el menor esfuerzo posible que se encargue de realizar dicho controlador.

### 3.1.2.3 Respuesta del control del lazo de posición

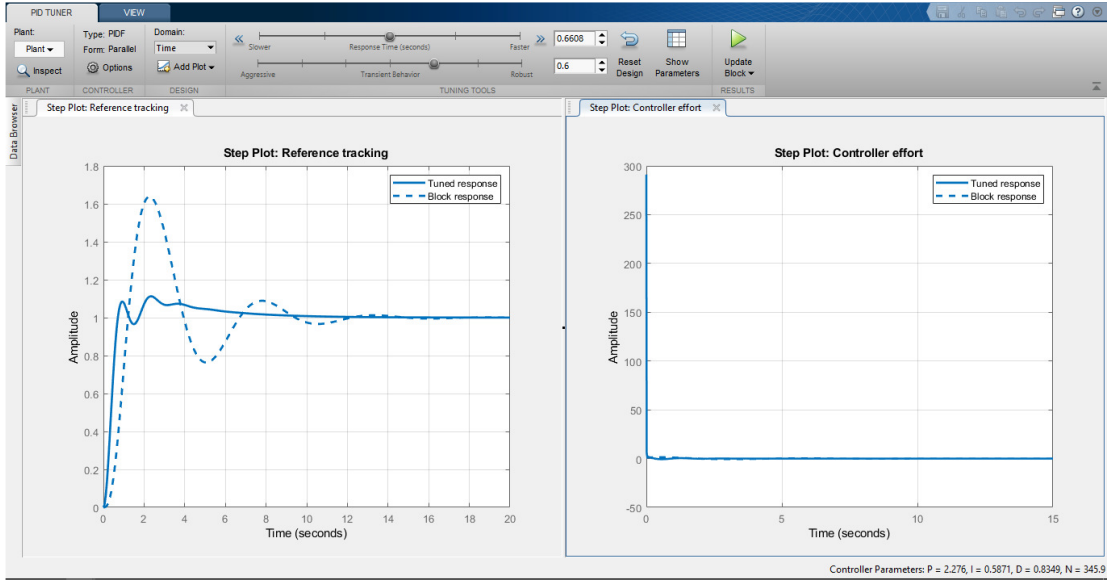
Ahora la variable que es de interés, la posición, se hará realimentando junto con la dinámica de control de las 2 anteriores variables mencionadas, así también se añaden perturbaciones en la corriente y el torque de carga para poder evaluar el comportamiento de los controladores diseñados, quedando el diagrama de bloques el mostrado en la figura 3.9.



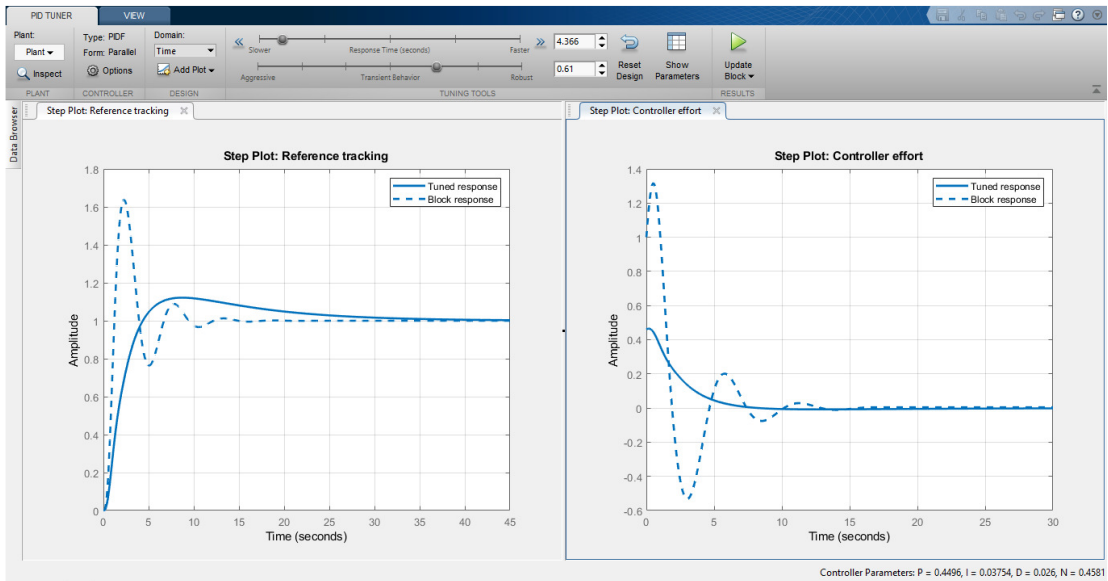
**Fig. 3.9.** Diagrama de bloques en Simulink del lazo de posición

Posterior a lo tratado, se hará el mismo análisis apreciado en los 2 controladores anteriores, es decir, autosintonización, ajuste fino y comportamiento ante perturbaciones como se lo muestra a continuación:

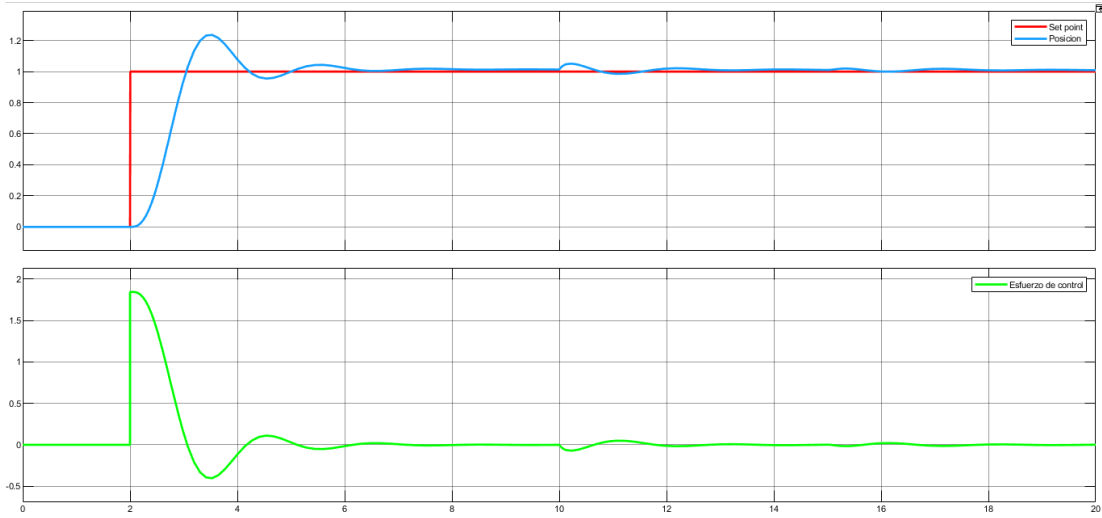




**Fig. 3.10.** Respuesta del control del lazo de posición mediante el autosintonizador.



**Fig. 3.11.** Respuesta del control del lazo de posición mediante al realizar un ajuste fino.



**Fig. 3.12.** Respuesta del control del lazo de posición ante cambios de referencia y perturbaciones.

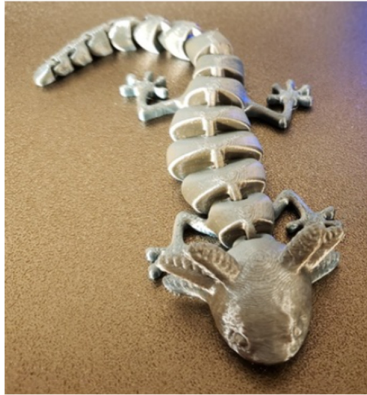
**Tabla. 3.3.** Valores P, I y D para el controlador de posición.

Parte	Autosintonizador	Ajuste fino
Proporcional (P)	2.276	0.4496
Integral (I)	0.5871	0.03754
Derivativa (D)	0.8349	0.026

En la tabla 3.3 se observan los valores determinados para los 2 diferentes técnicas de sintonización y la variación de valores, al igual que en el controlador de velocidad, se ve una alteración de valores muy grandes, debido a que se buscará el menor esfuerzo del controlador y obtener la respuesta adecuada como se lo ha venido mencionado anteriormente.

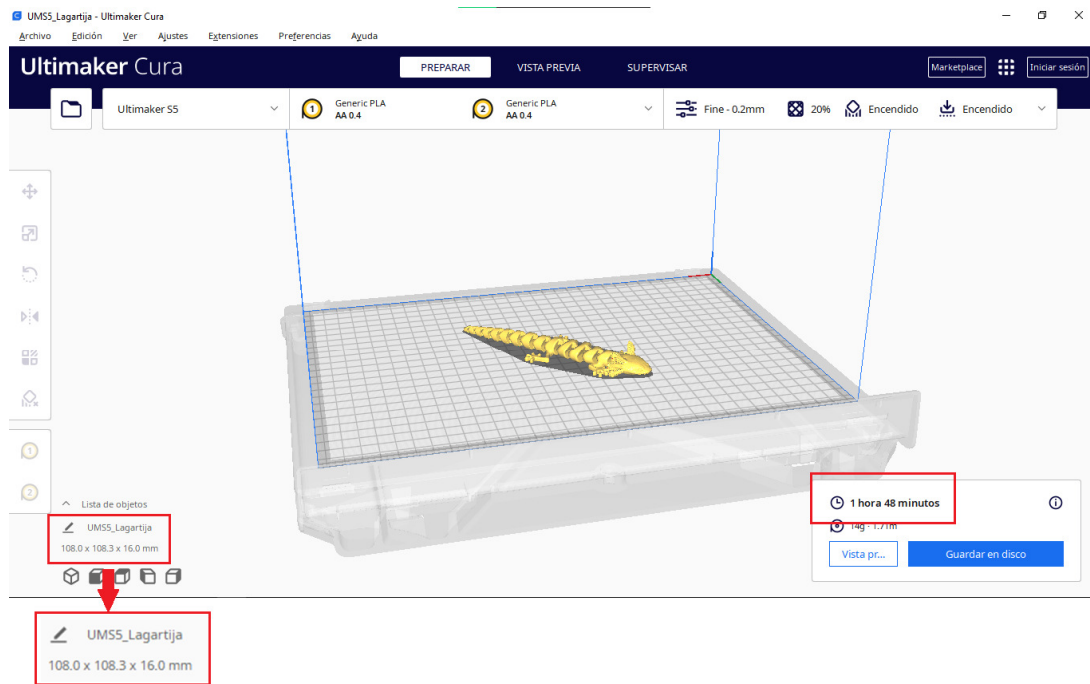
### 3.1.3 Proceso de impresión

Para poder lograr el cometido de llegar la impresión 3D se ha decidido partir de un objeto 3D real a ser escaneado, produciendo de esta manera el archivo con extensión STL, el objeto en cuestión es una estatua de una lagartija como se lo visualiza en la figura 3.13.



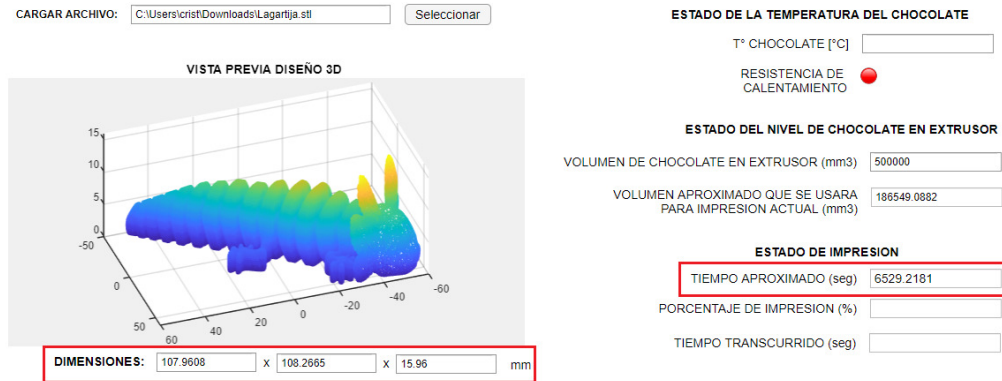
**Fig. 3.13.** Figura de una lagartija de metal.

Al realizar un escaneo del objeto en mención, se puede visualizar a este en el software CURA, como se lo aprecia en la figura 3.14, el cual es una aplicación profesional de manejo de impresoras 3D que ha sido como referencia para la elaboración de este trabajo. Este software extrae la figura en archivo STL al ser escaneado para poder ser cargada en la aplicación diseñada, obteniendo la figura siguiente, además de poder visualizar sus dimensiones y el tiempo de impresión.



**Fig. 3.14.** Representación del modelo de la figura 3.13 en el software CURA.

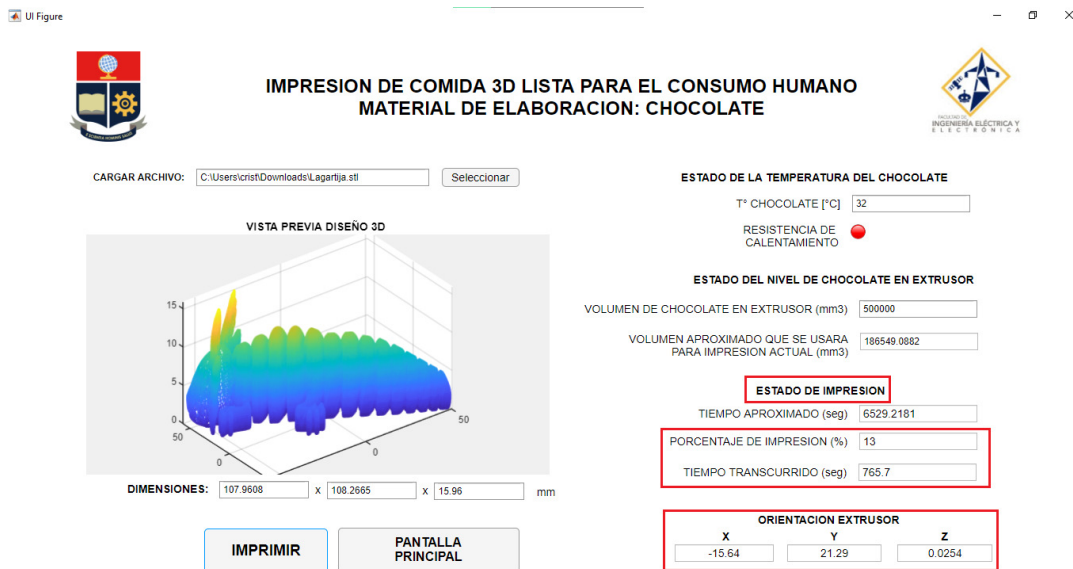
Al ser ya cargado en el software desarrollado se obtiene el resultado de la figura 3.15 y de igual manera con las dimensiones de la figura original y el cálculo del tiempo aproximado que tomará la figura en ser impresa.



**Fig. 3.15.** Representación del modelo de la figura 3.13 en el software desarrollado.

Las dimensiones varían en un margen muy pequeño, el programa CURA está redondeando a la décima más cercana mientras que el software desarrollado muestra un valor más exacto. Por otra parte, en el tiempo aproximado para impresión, el software calcula 1 hora con 48 minutos que, transformando a segundos, el valor sería 6480 segundos, mientras que el software diseñado calcula un valor de 6529.2181 segundos, teniendo una diferencia de 49 segundos, siendo un error porcentual de 0.76% el que se tendría al comparar ambos softwares.

Al presionar el botón "IMPRIMIR" se lleva a cabo el proceso, donde el porcentaje de impresión avanza mientras el tiempo también va transcurriendo y adicional a ello, se observa la posición actual del extrusor donde deposita la pasta de chocolate en tiempo real, esto se muestra en la figura 3.16.



**Fig. 3.16.** Proceso de impresión de la figura 3.13 en el software desarrollado.

En los anexos de este trabajo se adjunta un enlace que dirige hacia un video donde se puede visualizar todo el proceso de impresión visto desde el software diseñado, así como se tiene acceso a los archivos generados en este trabajo.

## 3.2 Conclusiones

Indicar las recomendaciones formuladas a partir del desarrollo de este trabajo de integración curricular.

- App Designer, complemento de Matlab, es una gran herramienta para el desarrollo de nuevas aplicaciones ya que brinda muchos estilos de personalización y diseño acorde a lo que el usuario desee crear, partiendo de opciones simples de manejo.
- El manejo de archivos STL es el más sencillo en lo que respecta al uso de modelos tridimensionales digitales ya que este excluye características del color del modelo, aspecto no necesario en la elaboración de figuras de chocolate.
- El diseño de controladores puede convertirse en un arduo trabajo, pero a través de la herramienta de autosintonización que posee Simulink, se puede llegar a transformar en una tarea sencilla, aparte de que se puede ir cambiando diferentes propiedades como son el tiempo de respuesta y/o la respuesta en estado transitorio para obtener la respuesta deseada.
- Los tiempos en ejecutarse una impresión 3D pueden ser muy largos dependiendo de la complejidad y el tamaño del diseño, pudiendo llevar hasta horas en que se finalice este proceso, dicho esto, el tiempo de cálculo que se demoraría en realizar

la tarea de impresión está basado principalmente en las características de las partes móviles del hardware de la impresora.

- Los ficheros con extensión STL incluyen información acerca de las figuras tridimensionales mediante 2 matrices, una de ellas es la malla de puntos la cual es la que brinda la localización donde el extrusor hará el depósito de chocolate para formar el modelo 3D.
- El intercambio de datos entre el software desarrollado en Matlab con la placa base del hardware de la impresora se hace de forma sencilla dado que a través de Simulink y su librería de comunicación serial donde con la configuración correcta de las características del microprocesador se puede llegar a lograr. En este trabajo de integración curricular no se pudo verificar dicha comunicación, pero en asignaturas tomadas en la carrera como sistemas microprocesados o control embebido se ha realizado diferentes proyectos que abarcan esta temática. La placa base utiliza un microprocesador ATmega, dispositivo que ya ha sido utilizado en las asignaturas mencionadas anteriormente teniendo éxito en la comunicación.
- Se ha contrarrestado los resultados obtenidos del software de simulación con los del software CURA, el cual es un software profesional que se acopla a impresoras 3D fabricadas por grandes empresas de tecnología y en lo que concierne a tiempos de impresión, dimensiones de modelos 3D, entre otros, se obtienen resultados muy similares.

### **3.3 Recomendaciones**

- Cuando se realice el manejo de los archivos STL se debe tener en cuenta que la matriz que contiene este fichero con el mapeo de puntos viene desordenada y se debe realizar un reordenamiento de esta de acuerdo con el mandato del programador.
- Para el diseño de controladores tener en cuenta los valores de sobrepico y tiempos de establecimiento requerido, es importante que para una verificación de funcionamiento correcto de estos se añadan cambios de referencia y perturbaciones para visualizar un comportamiento acorde a lo que sucedería en la vida real.
- En el uso del componente App Designer es de fácil manejo, pero se debe tener un orden para llevar a cabo la programación de todos los objetos que se vayan utilizando a lo largo de la elaboración de la lógica ya que en la generación de nuevas funciones generalmente se hacen uso de variables globales que deben ser

declaradas como tales en el encabezado del programa, aparte de dentro de cada una de las funciones generadas.

- Utilizar los diferentes componentes de Matlab como Simulink, App Designer y scripts ayudan mucho en la elaboración de un proyecto grande ya que se pueden realizar diferentes tareas en cada uno de estos e interconectar para tener un solo proyecto funcional acorde a los requerimientos del programador.

## 4 REFERENCIAS BIBLIOGRÁFICAS

- [1] 3D Printing Industry, "The Free Beginner's Guide To 3d Printing," Jul. 2014. [Online]. Available: <https://3dprintingindustry.com/wp-content/uploads/2014/07/3D-Printing-Guide.pdf>, Accessed on: Jul. 21, 2022.
- [2] Educación 3.0, "10 Programas De Diseño 3D Para Seguir Aprovechando Tu Impresora 3D," [www.educaciontrespuntocero.com](http://www.educaciontrespuntocero.com), Jan. 12, 2017. [Online]. Available: <https://www.educaciontrespuntocero.com/noticias/programas-diseno-impresoras-3d/>, Accessed on: Jul. 21, 2022.
- [3] R3ALD, "¿Qué es un fichero STL?," [www.r3ald.com](http://www.r3ald.com), Dec. 01, 2014. [Online]. Available: <http://www.r3ald.com/que-es-un-fichero-stl>, Accessed on: Jul. 21, 2022).
- [4] 3DFILAMENTS, "¿Qué es el stl? Los archivos por excelencia para impresión 3D explicados de forma fácil," [www.3dfils.com](http://www.3dfils.com), 2021. [Online]. Available: <https://www.3dfils.com/es/blog/entrada/que-es-stl.html>, Accessed on: Jul. 22, 2022.
- [5] A. Lucero and E. Vásconez, "implementación De Un Prototipo De Impresora 3D Controlada Inalámbicamente Por Bluetooth Mediante Una Plataforma Android," Jun. 2017. [Online]. Available: <https://dspace.ups.edu.ec/bitstream/123456789/14619/1/UPS-GT001984.pdf>, Accessed on: Jul. 23, 2022.
- [6] A. Molina, "¿Qué es un G-Code y cuál es su función en el proceso de impresión 3D?," 3Dnatives, Sep. 22, 2021. [Online]. Available: <https://www.3dnatives.com/es/g-code-proceso-impresion-3d-230920212/>, Accessed on: Jul. 23, 2022.
- [7] B. Florida, "Mantenimiento de Computadores Hardware Interno," Universidad de Cundinamarca, Mar. 2011. [Online]. Available:

- <https://informaticafloridablanca.files.wordpress.com/2011/03/hardware-interno.pdf>, Accessed on: Jul. 27, 2022.
- [8] C. Lima, "Diseño de placa de expansión para control de impresora 3D," Escuela Técnica Superior de Ingeniería Universidad de Sevilla, 2017. [Online]. Available: [https://biblus.us.es/bibing/proyectos/abreproy/91555/fichero/TFG\\_Carlos+Lima+Lopez.pdf](https://biblus.us.es/bibing/proyectos/abreproy/91555/fichero/TFG_Carlos+Lima+Lopez.pdf), Accessed on: Jul. 27, 2022.
- [9] MathWorks, "MATLAB App Designer," [la.mathworks.com](http://la.mathworks.com), 2016. [Online]. Available: <https://la.mathworks.com/products/matlab/app-designer.html> Accessed on: Jul. 27, 2022.
- [10] B. Castillo, "6 Tipos De Métodos De Investigación," Guía Universitaria, Oct. 14, 2020. [Online]. Available: <https://guiauniversitaria.mx/6-tipos-de-metodos-de-investigacion/>, Accessed on: Jul. 31, 2022).
- [11] E. Hernández, "Enfoques Cuantitativo, Cualitativo Y Mixto," [www.eumed.net](http://www.eumed.net), 2007. [https://www.eumed.net/tesis-doctorales/2012/mirm/enfoque\\_cualitativo.html](https://www.eumed.net/tesis-doctorales/2012/mirm/enfoque_cualitativo.html) (accessed Jul. 31, 2022).
- [12] R. Javier, "Tipos de investigación," Psicología Psico Zoebisch, 2010. [Online]. Available: <https://sites.google.com/site/metodelainvest1/unidad-iii-tipos-de-investigacion>, Accessed on: Jul. 31, 2022).
- [13] R. Saiz, "Técnicas De Análisis De Información" Unican.es, Jan. 23, 2021. [Online]. Available: [https://grupos.unican.es/mide/masterinnova/rincon\\_alumnos/resumenes/TECNICAS%20DE%20AN%C3%81LISIS%20DE%20INFORMACI%C3%93N.doc](https://grupos.unican.es/mide/masterinnova/rincon_alumnos/resumenes/TECNICAS%20DE%20AN%C3%81LISIS%20DE%20INFORMACI%C3%93N.doc) Accessed on: Jul. 31, 2022.
- [14] Chocolates Artesanos Isabel, "¿Cómo hacer el templado manual del chocolate?," Chocolates Artesanos Isabel, Jun. 24, 2021. [Online]. <https://chocolatesartesanosisabel.com/templado-manual-del-chocolate/#:~:text=La%20manteca%20de%20cacao%20est%C3%A1>, Accessed on: Aug. 1, 2022.
- [15] La Electrónica, "Resistencia Mitsu de 900W Para Calentar Agua," La Electrónica, 2018. [Online]. Available: <https://laelectronica.com.gt/resistencia-mitsu-de-900w-para-calentar-agua>. Accessed on: Aug. 01, 2022).
- [16] Grupo Disco, "Calores Específicos," [www.grupodisco.com](http://www.grupodisco.com), 2013. [Online]. <https://www.grupodisco.com/wp-content/uploads/FileManager/docs/DATOS%20T%C3%89CNICOS/REFRIGERACI%C3%93N/Calores%20espec%C3%ADficos.pdf>, Accessed on: Aug. 01, 2022.



- [17] AREATECNOLOGIA, "Resistencia Eléctrica," [www.areatecnologia.com](http://www.areatecnologia.com), 2013. [Online]. Available: <https://www.areatecnologia.com/electricidad/resistencia-electrica.html> . Accessed on: Aug. 02, 2022.
- [18] Schneider Electric, "NEMA Size 17 1.8° 2-Phase Stepper Motor," Schneider Electric, 2017. [Online]. Available: [http://www.autoflexible.com/file\\_upload/product/attach/NEMA%2017.pdf](http://www.autoflexible.com/file_upload/product/attach/NEMA%2017.pdf), Accessed on: Aug. 02, 2022.
- [19] S. Castaño, "Control en Cascada," Control Automático Educación, Oct. 20, 2019. [Online]. Available: <https://controlautomaticoeducacion.com/control-realimentado/control-en-cascada/>, Accessed on: 02, 2022.
- [20] MathWorks, "PID Controller Tuning in Simulink," [la.mathworks.com](http://la.mathworks.com), 2016. [Online]. Available: <https://la.mathworks.com/help/slcontrol/gs/automated-tuning-of-simulink-pid-controller-block.html#:~:text=using%20PID%20Tuner.->, Accessed on: Aug. 03, 2022.
- [21] MathWorks, "Create warning dialog box," [la.mathworks.com](http://la.mathworks.com), 2015. [Online]. Available: <https://la.mathworks.com/help/matlab/ref/warndlg.html> Accessed on: Aug. 04, 2022.
- [22] S. Castaño, "Comunicación Serial Arduino Simulink/Matlab," Control Automático Educación, [Online]. Available: Jul. 07, 2021. <https://controlautomaticoeducacion.com/arduino/comunicacion-serial-arduino-simulink-matlab/>, Accessed on: Aug. 04, 2022.
- [23] MathWorks, "Package Apps in App Designer - MATLAB & Simulink - MathWorks América Latina," [la.mathworks.com](http://la.mathworks.com), 2017. [Online]. Available: [https://la.mathworks.com/help/matlab/creating\\_guis/package-apps-in-app-designer.html](https://la.mathworks.com/help/matlab/creating_guis/package-apps-in-app-designer.html), Accessed on: Aug. 05, 2022.

## **5 ANEXOS**

ANEXO I. Código desarrollado en Matlab en el diseño del software de control.

ANEXO II. Enlaces.

## ANEXO I. Código desarrollado en Matlab en el diseño del software de control.

A continuación, se muestra parte del código desarrollado, particularmente la carga y lectura de archivos STL:

```
% Button pushed function: cargar_archivo
function cargar_archivoButtonPushed(app, event)
global orden_seguimiento;
global FileName;
[file, pathToFile] = uigetfile('*.stl','Seleccione un archivo
con extension STL'); %Busqueda del archivo stl en el ordenador
FileName = strcat(pathToFile,file);
imagen3d = stlread(strcat(pathToFile, file)); %lectura del
archivo STL
espera = uiprogresdgl(uifigure, 'Title', 'Subiendo archivo');
for i = 1:10
    if espera.CancelRequested
        break
    end
    espera.Value = (i-1)/10;
    espera.Message = sprintf ('Por favor, espere...') ;
    pause(2);
end
trimesh(imagen3d, 'Parent', app.mostrar_imagen);
app.texto_archivo.Value = FileName;
seguimiento = imagen3d.Points;
orden_seguimiento = sortrows(seguimiento,3) %Ordenamiento de
vectores acorde al eje Z
[fil com] = size (orden_seguimiento);
maximos = max (seguimiento);
minimos = min (seguimiento);
par_vol = maximos - minimos;
largo = par_vol(1);
ancho = par_vol(2);
prof = par_vol(3);
volumen = largo*ancho*prof;
app.nivelaproximado.Value = num2str(volumen);
app.tiempoaproximado.Value = num2str(volumen*0.035); %el valor
0.035 dependera de la velocidad con la que el extrusor de la impresora
expulse chocolate
app.largo.Value = num2str(largo);
app.ancho.Value = num2str(ancho);
app.profundidad.Value = num2str(prof);

end

% Button pushed function: IMPRIMIR
function IMPRIMIRPushed(app, event)
global orden_seguimiento
global chocofrio
```

```

global chococaliente
global tempchocolate
global FileName
[fil com] = size (orden_seguimiento);
tempchocolate = str2num(app.tchocolate.Value);
arch_cargado = isempty(app.texto_archivo.Value);

if arch_cargado == 1
    uialert(uifigure,'No ha cargado ningun archivo para ser
impreso','Error','Icon','error');
else
    if (str2num(app.nivelextrusor.Value) <
str2num(app.nivelaproximado.Value))
        uialert(uifigure,'La cantidad de chocolate actual no es
suficiente para imprimir','Advertencia','Icon','warning');
    else
        if (tempchocolate >= 32) && (tempchocolate <=35)
            espera =
uiprogressdlg(uifigure,'Title','IMPRIMIENDO','Cancelable',"on");
            for i = 1:fil
                matriz3d = orden_seguimiento(i,:);
                app.X.Value = matriz3d(1);
                app.Y.Value = matriz3d(2);
                app.Z.Value = matriz3d(3);
                app.statusimpresion.Value =
num2str(round(espera.Value*100));
                app.tiempotrancurrido.Value = num2str(i*3.538);
                %Cambiar el 0.34 segun corresponda)
                espera.Value = i/fil;
                espera.Message = sprintf ('Por favor, espere...');
                if espera.CancelRequested
                    break
                end
            end
            uialert(uifigure,'Su diseño 3D ha sido
impreso','Listo','Icon','success');
        else
            uialert(uifigure,'La temperatura del chocolate no es la
adecuada','Advertencia','Icon','warning');
            if tempchocolate >= 35
                chocofrio = 1;
                app.ENFRIARLamp.Color = [0.00,1.00,0.00];
            end
            if tempchocolate <= 35
                chococaliente = 1;
                app.RESISTENCIADECALENTAMIENTOLamp.Color =
[0.00,1.00,0.00];
            end
        end
    end
end
end

```

```
end
```

```
end
```

```
% Value changed function: tchocolate  
function tchocolateValueChanged(app, event)  
    value = str2num(app.tchocolate.Value);  
    if (value >= 32) && (value <=35)  
        app.RESISTENCIADECALENTAMIENTOLamp.Color = 'red';  
        uialert(uifigure, 'La temperatura del chocolate esta lista  
para imprimir. Presione el boton "IMPRIMIR" para  
comenzar', 'Listo', 'Icon', 'success');  
    end
```

## **ANEXO II. Enlaces**

En el siguiente enlace se tiene acceso a todos los archivos utilizados para la realización de este trabajo de integración curricular que se ha obtenido de la programación hecha en Matlab.

<https://drive.google.com/drive/folders/1M9leDLfGJbQI4Ba2dheX6IVoFhJSFyTx?usp=sharing>

El siguiente enlace conduce hacia un video de YouTube donde se aprecia una simulación y el funcionamiento del software diseñado:

<https://youtu.be/W19i3uJLs8>