



# **ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**Facultad de Ingeniería en Electricidad y Computación**

“Diseño mediante Sistema Arduino de controlador para Proyecto  
RepRap, aplicaciones de auto replicación en prototipo de  
impresión 3D”

## **TESINA DE SEMINARIO**

**Previo a la obtención del Título de:**

**INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES**

**Presentada por:**

**DANNY ALEXANDER HERAS RAMIREZ.**

**EDWIN OMAR CORREA LEON.**

**GUAYAQUIL – ECUADOR**

**AÑO: 2013**

## **AGRADECIMIENTO**

A nuestro tutor, el Ing. Carlos Valdiviezo, así como a las demás personas que han formado parte del seminario, con quienes hemos compartido esta gran experiencia de aprendizaje e investigación.

También debemos agradecer a todos los que han aportado en ayudarnos a cumplir con nuestra meta; amigos, profesores y autoridades de este prestigioso establecimiento educativo.

## DEDICATORIA

A nuestros padres por enseñarnos que si hay perseverancia no hay imposibles, además de su apoyo incondicional en toda nuestra vida académica.

## TRIBUNAL DE SUSTENTACIÓN



Ing. Carlos Valdivieso A.

**PROFESOR DEL SEMINARIO DE GRADUACIÓN**



Ing. Hugo Villavicencio V.

**PROFESOR DELEGADO POR LA UNIDAD ACADÉMICA**

## INDICE GENERAL

AGRADECIMIENTO .....	1
DECLARATORIA .....	2
<b>DECLARACIÓN EXPRESA .....</b>	<b>3</b>
INDICE GENERAL .....	7

"La responsabilidad del contenido de esta Tesina, nos corresponde exclusivamente; y el patrimonio intelectual de la misma, a la Escuela Superior Politécnica del Litoral".

1.1 ANTECEDENTES .....

1.2 MOTIVACIÓN .....

(Reglamento de Graduación de la ESPOL)

1.3.1 OBJETIVO GENERAL .....

1.3.2 OBJETIVOS ESPECÍFICOS .....

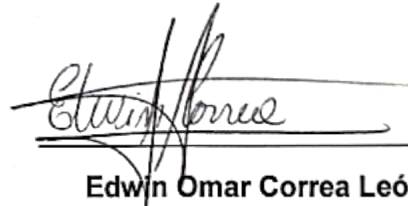
1.4 LIMITACIONES .....

CAPÍTULO 2

2.1



**Danny Alexander Heras Ramírez**



**Edwin Omar Correa León**

# ÍNDICE GENERAL

AGRADECIMIENTO .....	II
DEDICATORIA .....	III
DECLARACIÓN EXPRESA .....	V
ÍNDICE GENERAL.....	VI
ÍNDICE DE FIGURAS.....	IX
INTRODUCCIÓN.....	XII
<b>CAPÍTULO 1.....</b>	<b>1</b>
<b>DESCRIPCIÓN GENERAL DEL PROYECTO .....</b>	<b>1</b>
<b>1.1 ANTECEDENTES .....</b>	<b>2</b>
<b>1.2 MOTIVACIÓN .....</b>	<b>5</b>
<b>1.3 OBJETIVOS.....</b>	<b>6</b>
<b>1.3.1 OBJETIVO GENERAL.....</b>	<b>6</b>
<b>1.3.2 OBJETIVOS ESPECÍFICOS.....</b>	<b>6</b>
<b>1.4 LIMITACIONES .....</b>	<b>7</b>
<b>CAPÍTULO 2.....</b>	<b>9</b>
<b>SUSTENTACIÓN TEÓRICA.....</b>	<b>9</b>
<b>2.1 RESUMEN.....</b>	<b>9</b>
<b>2.2 PROYECTO REPRAP.....</b>	<b>10</b>
<b>2.2.1 IMPRESORAS 3D DESTACADAS.....</b>	<b>11</b>
<b>2.3 SOFTWARE A UTILIZAR .....</b>	<b>16</b>
<b>2.3.1 BLENDER .....</b>	<b>16</b>
<b>2.3.2 PROGRAMA AUTOCAD 3D.....</b>	<b>17</b>
<b>2.3.3 PROGRAMA REPLICATOR G.....</b>	<b>18</b>
<b>2.3 HARDWARE DEL CONTROLADOR.....</b>	<b>19</b>
<b>2.4.1 TARJETA ARDUINO UNO.....</b>	<b>20</b>

2.4 PARTES DE LA IMPRESORA 3D .....	27
2.5.1 ROBOT CARTESIANO .....	27
2.5.2 EXTRUSOR (THERMOPLASTIC EXTRUDER) .....	29
2.5.3 TABLERO DE IMPRESIÓN (HEATED PRINTBED).....	31
2.5.4 LINEAS DE MOVIMIENTO.....	32
<b>CAPÍTULO 3.....</b>	<b>33</b>
<b>DESARROLLO FINAL DEL PROYECTO .....</b>	<b>33</b>
3.1 RESUMEN DEL CAPITULO .....	33
3.2 EJERCICIO PRÁCTICO: CONTROL PARA MOTOR DE PASO .....	34
3.2.1 CODIGO FUENTE .....	36
3.2.2 IMPLEMENTACIÓN EN PROTOBOARD.....	37
3.3 DISEÑO DEL CONTROLADOR PARA EL PROYECTO REPRAP .....	39
3.3.1 CONEXIONES PARA LA RAMPS 1.2 .....	40
3.4 MODIFICACIÓN EN LA IMPRESORA 3D.....	44
3.4.1 FINALES DE CARRERA O END STOP .....	44
3.4.2 CONEXIONES DEL EXTRUSOR.....	46
3.4.3 CALIBRACIÓN DE DRIVERS .....	47
3.5 SOFTWARE DEL ARDUINO Y FIRMWARE .....	49
3.5.1 PROCESO PARA CARGAR FIRMWARE EN ARDUINO .....	50
3.5.2 SOFTWARE DE CONTROL .....	55
<b>CAPÍTULO 4.....</b>	<b>57</b>
<b>SIMULACIONES Y RESULTADOS .....</b>	<b>57</b>
4.1 RESUMEN DEL CAPÍTULO .....	57
4.2 SIMULACION SLICER.....	59
4.2.1 CONFIGURACION DE PARAMETROS .....	60
4.3 SIMULACION CON G-WIZARD .....	61
4.4 SIMULACION CON REPETIER HOST .....	64
4.4.1 SECCIONAMIENTOS.....	67
4.5 RESULTADOS DE LA SIMULACIÓN.....	69
4.6 RESULTADOS DEL PROCESO DE LA IMPRESIÓN.....	69

<b>CONCLUSIONES</b> .....	77
<b>RECOMENDACIONES</b> .....	80
<b>BIBLIOGRAFÍA</b> .....	82



## ÍNDICE DE FIGURAS

Figura 1.1a: Máquina madre y su primera máquina RepRap hijo.....	2
Figura 1.1b: DARWIN, Primera Impresora 3D Open Source .....	3
Figura 1.1c: Impresora 3D Prusa Mendel .....	4
Figura 2.2a: Proyecto RepRap.....	10
Figura 2.2b: Impresora Prusa Mendel.....	12
Figura 2.2c: Impresora Huxley 3D .....	13
Figura 2.2d: Impresora Ultimaker.....	14
Figura 2.2e: Tabla de comparación de impresoras 3D .....	15
Figura 2.3a: Interfaz principal del programa Blender .....	16
Figura 2.3b: Programa Autocad.....	17
Figura 2.3c: Programa Replicator G. ....	19
Figura 2.4a: Tarjeta Arduino Uno.....	20
Figura 2.4b: Microcontrolador Atmega328I.....	21
Figura 2.4c: Componentes Tarjeta Arduino .....	22
Figura 2.4d: Diagrama de Bloques Atmega328 .....	25
Figura 2.5a: Partes Plásticas RepRap PRUSA MENDEL.....	27
Figura 2.5b: Thermoplastic Extruder.....	29
Figura 2.5c: Heated PrintBed (Tablero de Impresión).....	31
Figura 2.5d: Rodamientos de bronce en un PRUSA MENDEL.....	32

Figura 3.2a: Circuito de Prueba de Motores, Componentes Electrónicos.....	35
Figura 3.2b: Controlador de motor de paso .....	36
Figura 3.2c: Implementación de Controlador de motor de paso .....	38
Figura 3.3a: Esquema del Arduino + Ramps .....	40
Figura 3.3b: Esquema de conexión para la Ramps .....	41
Figura 3.3c: Tarjeta Ramps .....	42
Figura 3.3d: Impresora Prusa Mendel con Tarjeta Ramps .....	43
Figura 3.4a: Finales de carrera Mecánicos.....	45
Figura 3.4b: Conexiones del extrusor .....	47
Figura 3.4c: Drivers en la tarjeta Ramps (POLOLUS) .....	48
Figura 3.5a: Instalación del Arduino.....	51
Figura 3.5b: Selección del puerto serial.....	52
Figura 3.5c: Selección del Firmware para el Arduino.....	53
Figura 3.5d: Compilación del Código .....	54
Figura 3.5e: Interfaz Software PronterFace .....	56
Figura 4.1a: Modelado 3D de Partes de Impresora .....	58
Figura 4.1b: Vertice o frame-vertex-foot .....	59
Figura 4.2a: Configuración de Parámetros SLICER .....	60
Figura 4.3a: Simulación G-Wizard .....	61
Figura 4.3b: Perspectiva Lateral plano XY.....	62
Figura 4.3c: Vista Superior, Componente Simulado .....	63
Figura 4.3d: Perspectiva Lateral plano XY.....	64

Figura 4.4a: Vista Superior, Capa 1 .....	65
Figura 4.4b: Vista Superior, Capa 33.....	66
Figura 4.4c: Vista Superior Modelo 3D .....	66
Figura 4.4.1a Seccionamiento de líneas.....	67
Figura 4.4.1b Seccionamiento Rectilíneo .....	67
Figura 4.4.1c Seccionamiento de Panal .....	67
Figura 4.4.1d Seccionamiento Concéntrico .....	67
Figura 4.4.1e Seccionamiento de Curva de Hilbert.....	68
Figura 4.4.1f Seccionamiento de Curva de Arquímedes.....	68
Figura 4.6a: Impresión de Cubo.....	70
Figura 4.6b: Impresión de Vértice, Capas Base .....	70
Figura 4.6c: Impresión de Vértice, Relleno Interior .....	71
Figura 4.6d: Impresión de Vértice, Objeto Completo .....	71
Figura 4.6e: Impresión Case Raspberry pi .....	72
Figura 4.6f: Impresión Invader .....	72

# INTRODUCCIÓN

Nuestro proyecto se enfoca en el desarrollo del controlador de la impresora 3D Prusa Mendel, utilizando la tarjeta Arduino la cual tiene integrada el microcontrolador ATMEGA 328, además esta se complementa con la Tarjeta RAMPS 1.2, la cual nos ayuda al control y funcionamiento de motores de paso, sensores y calentamiento del extrusor de la impresora, para esto pondremos en práctica todo lo aprendido en la materia de graduación de Microcontroladores Avanzados, además de los conocimientos previos adquiridos de microcontroladores del pregrado, es necesario el software de control de la tarjeta Arduino el cual se encargara del control de las funciones de la impresora 3D.

En el capítulo 1 se describe el proyecto de forma general, exponiendo los antecedentes, motivación, objetivos y limitaciones para su conclusión. El

capítulo 2 está enfocado a la fundamentación teórica y las herramientas utilizadas para el desarrollo de nuestro proyecto. En el capítulo 3 será explicado el diseño preliminar, el desarrollo y modificaciones de Hardware necesarias para la finalización del proyecto. Y en el capítulo 4, se mostrarán las pruebas, simulaciones además del análisis de resultados.

# **CAPÍTULO 1**

## **DESCRIPCIÓN GENERAL DEL PROYECTO**

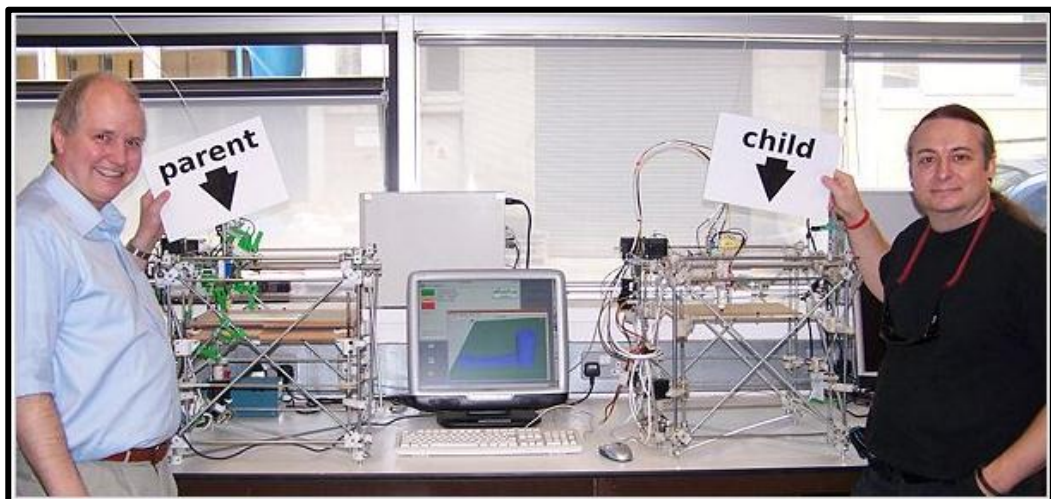
Este proyecto está dirigido al diseño de un controlador mediante el sistema Arduino con el que nos enfocaremos en la auto-replicación de partes mediante el proceso de impresión 3D, el cual consta de cinco etapas principales que son: Diseño del controlador, diseño del objeto, Codificación en código G, Transformación de código a órdenes mecánicas y finalmente llegando a la etapa de Impresión.

La idea de construir, diseñar y reproducir una nueva impresora 3D a partir de una ya existente es una forma eficiente de reducir costos. Esta es un herramienta muy poderosa y útil, además de tener un potencial interminable

que podría conducir a un mundo donde la gente pueda manipular y construir nuevos y creativos diseños, con la misma calidad que lo haría una empresa manufacturera desde la comodidad de nuestros hogares.

## 1.1 ANTECEDENTES

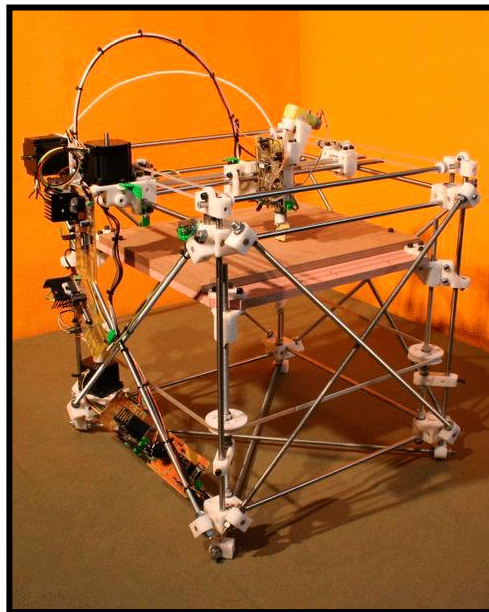
La palabra RepRap es el acrónimo para Replicating Rapid-prototyper (Prototipo de replicación rápida), el Proyecto RepRap tiene sus inicios en la universidad de Bath, Reino Unido, por el ingeniero Adrian Bowyer (**Ver Figura 1.1a**), quien tuvo la idea de crear impresoras auto replicables de código libre, es decir cualquiera puede tener acceso a este código bajo ningún costo, revolucionando la industria de la impresión 3D.



**Figura 1.1a: Máquina madre y su primera máquina RepRap hijo**

Esta tecnología ya existía mucho tiempo atrás pero era demasiado costosa, la máquina de menor valor en el mercado tiene un costo de alrededor de €30,000 y no estaba diseñada para ser manipulada por una sola persona.

La idea de auto replicación apareció por primera vez en la web en febrero del 2004, luego de años de trabajo el 9 de febrero del 2008 exactamente, apareció la primera impresora 3D la cual la llamaron DARWIN (**Ver Figura 1.1b**).



**Figura 1.1b: DARWIN, Primera Impresora 3D Open Source**

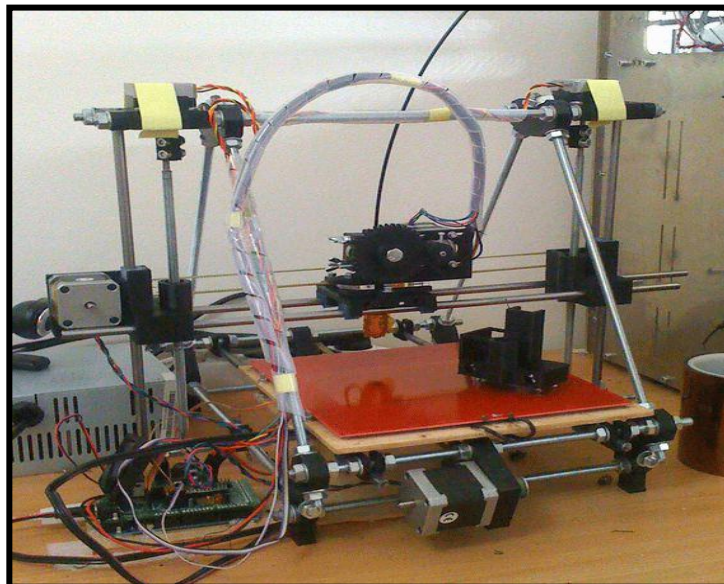
En Septiembre del mismo año, ya existían más de 100 de copias del modelo DARWIN en diferentes partes del mundo. Para octubre del 2009, diseñaron una nueva impresora 3D, basada en el modelo DARWIN, pero menos



sofisticada respecto a su montaje para así convertirla lo más autoreplicable posible, con esta nueva modificación nace la generación MENDEL.

En agosto del 2010, desarrollaron un modelo compacto de las impresoras MENDEL, denominada como impresora Huxley o "mini-Mendel", la cual requería un 30% menos de piezas para su construcción.

El mismo mes el joven llamado Josef Prusa, dejó asombrados a toda la comunidad RepRap, al presentar un nuevo diseño de las impresoras 3D, la RepRap Prusa Mendel (**Ver Figura 1.1c**), es una nueva evolución de la impresoras MENDEL, con la característica principal que serían mucho más sencillas de ensamblar, haciendo menos complejo su uso y permitiendo a más personas tener fácil acceso a esta tecnología.



**Figura 1.1c: Impresora 3D Prusa Mendel**

## 1.2 MOTIVACIÓN

Una de las principales aspiraciones y deseos que tenemos las personas es de tener la capacidad de crear objetos, pero esta siempre es limitada por muchos factores tales como costos, uso de software y dispositivos sofisticados para su implementación. La principal motivación es el uso del sistema Arduino para el diseño de un controlador el cual es necesario para el manejo y calibración de la impresora 3D.

La facilidad que nos da el uso de un sistema altamente utilizado en todo el mundo como lo es sistema Arduino, nos permite tener una mejor interacción con el controlador, teniendo un fácil acceso rutinas y procedimientos que hayan sido utilizados en prototipos ya implementados, los cuales pueden ser un base para la creación de nuevas rutinas que serán utilizadas para el control de los motores de la impresora 3D.

El uso de un software libre para la creación de objetos existentes y la capacidad de inventar nuevos diseños, lo hace un proyecto tentativo a que personas de todas las edades quieran involucrarse, además existen miles de diseños disponibles en la web, totalmente gratuitos haciendo que esto sea beneficioso para nuestro aprendizaje.

Las empresas manufactureras poseen grandes máquinas para producir objetos de una forma masiva y sofisticada, es casi imposible para cualquier persona adquirir una de estas, pero con la revolucionara impresora 3D cualquier puede hacerlo a un costo medianamente accesible.

## **1.3 OBJETIVOS**

### **1.3.1 OBJETIVO GENERAL**

- ✓ Como objetivo general de nuestro proyecto nos enfocaremos en el diseño de un controlador basado en el sistema Arduino como parte fundamental de nuestro proyecto RepRap, además el uso de herramientas de diseño y programas necesarios para el proceso de la impresión 3D.

### **1.3.2 OBJETIVOS ESPECÍFICOS**

- ✓ Diseño de un controlador basado en el sistema Arduino el cual se encarga del control de los motores y temperatura del extrusor de nuestra impresora 3D.
- ✓ Dominar la programación y uso de la plataforma de hardware libre basada en microcontroladores (tarjeta Arduino), programación basada en lenguaje C.

- ✓ Funcionamiento de los distintos programas utilizados en el proceso de la impresión 3D.
- ✓ Dominar la calibración exacta de la impresora Prusa Mendel, para el proyecto de auto replicación RepRap, debido a la precisión necesaria al momento de mover sus motores o el manejo de la temperatura a la cual se debe calentar el material utilizado en la impresión.

## **1.4 LIMITACIONES**

Una de las principales limitaciones en el proyecto sería, la capacidad para modelar piezas de un tamaño definido, básicamente la impresora no puede realizar impresiones fuera de los estándares previamente establecidos en sus especificaciones.

Otra de las limitaciones que se debe tomar en cuenta es, que la pieza a modelar es monocroma, debido a que el polímero con el que se fabricaría es de un solo color.

Como otra limitación podríamos considerar que para realizar una impresión debemos tener programas previamente instalados en nuestra computadora y

el uso de un controlador previamente diseñado, dado que no hay una interfaz directa entre la computadora y la impresora 3D.

Una limitación básica del microcontrolador Arduino sería que necesita tener una fuente de poder específica y la fragilidad de las conexiones a los pines que se encargan de controlar los motores de la impresora 3D.

## **CAPÍTULO 2**

### **SUSTENTACIÓN TEÓRICA**

#### **2.1 RESUMEN**

En este capítulo vamos a dar una descripción detallada de todos y cada uno de los elementos necesarios en el proceso de la impresión 3D, se detallará el uso del software especializado en diseño el cual nosotros vamos a utilizar en el proceso de nuestra impresión 3D, programas necesarios para la programación y calibración de nuestra impresora, haremos un breve resumen de lo que es el proyecto RepRap, se detallaran tres impresoras 3D ya existentes considerando sus características principales así como también

mencionaremos las partes y especificaciones técnicas de nuestro controlador Arduino.

Descripción del software de simulación, muy útil para saber cómo quedaría el diseño de nuestra impresión final, ya que sería necesaria una simulación previa del objeto a imprimirse, dado que en la transformación a código del archivo con extensión .STL se pueden generar algunos errores que conllevarían a una mala impresión.

## 2.2 PROYECTO REPRAP

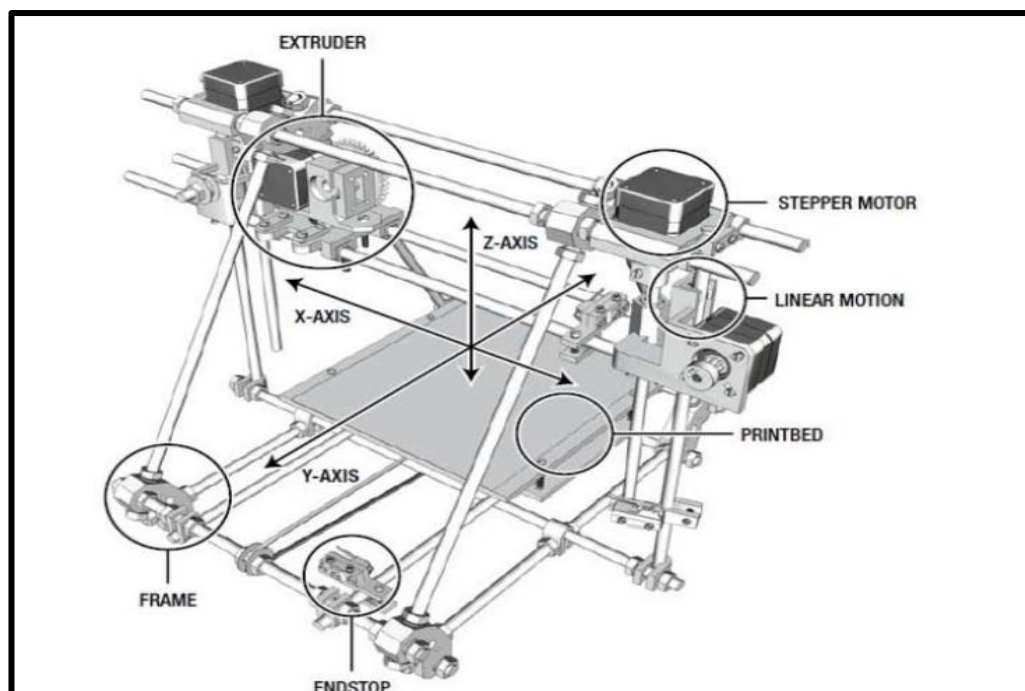


Figura 2.2a: Proyecto RepRap

El proyecto RepRap es una iniciativa para crear una máquina que sea capaz de auto replicarse, se la puede usar de una forma rápida para crear un prototipo y su respectiva manufactura, como ejemplo más óptimo sería el de una impresora 3D, esta es capaz de crear objetos en tres dimensiones a partir del modelado de dicho objeto previamente en una computadora.

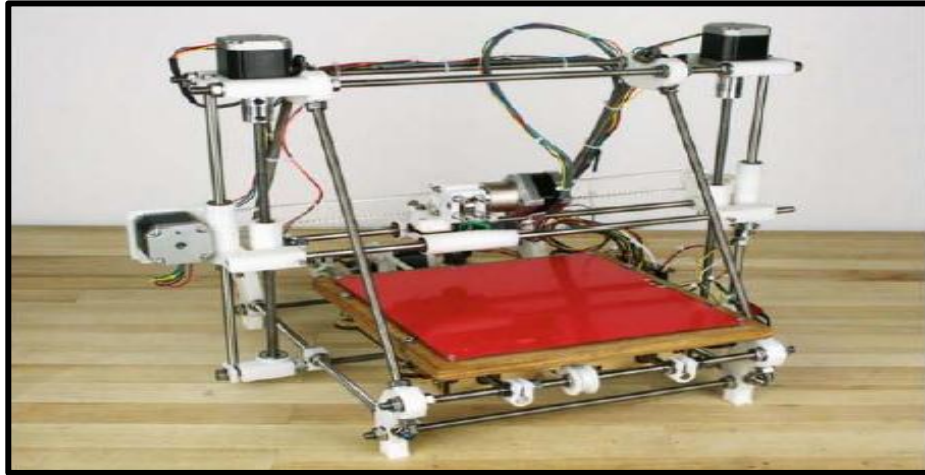
El creador del proyecto el proyecto de auto replicación como la posibilidad de crear componentes necesarias para construir otra versión de sí misma, debido al potencial de la auto réplica de la máquina su creador tuvo la visión de poder distribuir a bajo costo las maquinas RepRap.

### **2.2.1 IMPRESORAS 3D DESTACADAS**

En el mercado existen una gran variedad de impresoras 3D, se las divide generalmente por costos, calidad de impresión, velocidad de trabajo. Entre las principales que vamos a mencionar, se las tomo como referencia debido a su accesibilidad haciendo contraste hacia sus especificaciones técnicas.



### 2.2.1.1 MAKERGEAR PRUSA MENDEL

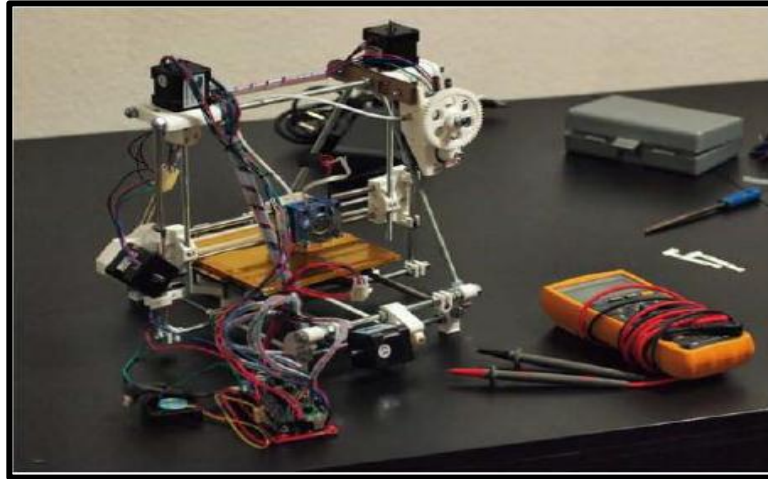


**Figura 2.2b: Impresora Prusa Mendel**

La impresora Prusa Mendel es la segunda generación de impresoras 3D, tiene un relativo bajo costo, está construida con pocas partes y además un volumen de 200mm x 200mm x 110mm, la Prusa Mendel utiliza 2 motores para el eje Z, la compañía MakerGear improviso con un diseño sencillo utilizando acero ligero, a continuación se detallan sus principales características.

- ✓ Rango medio en volumen de construcción.
- ✓ Con algunos trabajos de impresión se obtiene impresiones más rápidas.
- ✓ Asequible para su auto replicación, ya que sus partes son sencillas.

### 2.2.1.2 RERAP HUXLEY



**Figura 2.2c: Impresora Huxley 3D**

La Huxley fue diseñada más pequeña que su hermana la Prusa Mendel y mucho más portable fue diseñada por la compañía RepRapPro LTD ([www.RepRapPro.com](http://www.RepRapPro.com))

La Huxley es un poco más económica con respecto a su precio que la Prusa Mendel, su tamaño es de 140mm x 140mm x110mm, es una de las impresoras 3D más pequeñas que existen en el mercado actual, tiene una combinación de cojinetes lineales en el eje X y Y, también tiene una varilla enroscada de paso fino en el eje Z. a continuación se detallan sus características más esenciales.

- ✓ Una de las impresoras 3D más modestas por su pequeño tamaño.
- ✓ Tiene un sistema línea de movimiento preciso y confiable.

- ✓ Toma casi el mismo tiempo en construir y en configurar que la Prusa Mendel

### 2.2.1.3 ULTIMAKER



**Figura 2.2d: Impresora Ultimaker**

Tiene un tamaño de 210mm x 210mm x 220mm, la Ultimaker tiene el área de construcción más grande que todas las impresoras señaladas en esta reseña, sus ejes se pueden mover a una velocidad asombrosa.

Además de ser una de las impresoras 3D más precisas que existen debido a la calidad en el detalle de sus piezas, se la considera una de las impresoras 3D más caras del mercado actual, a continuación se detallan sus características esenciales.

- ✓ Mejor velocidad de impresión y precisión que otra impresora 3D
- ✓ Tiene una área de impresión grande a comparación de su tamaño
- ✓ El costo y la complejidad de su configuración limitan esta impresora la cual es adquirida en su mayoría por personas especializadas.

### 2.2.1.4 COMO ELEGIR LA IMPRESORA CORRECTA

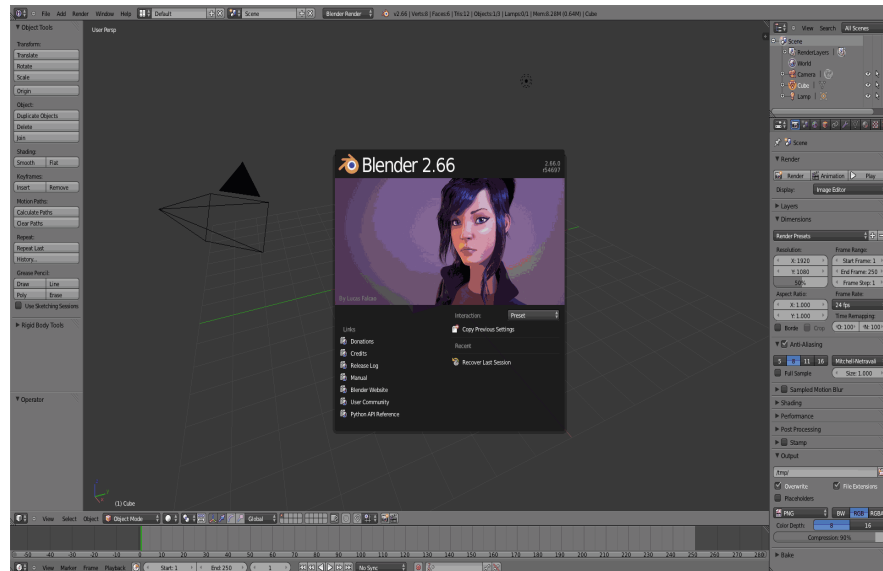
En el siguiente cuadro se detallan conjuntamente las características especiales de algunas de las impresoras existentes en el mercado con sus Pro y sus contras para así poder compáralas, cabe recalcar que la impresora que nosotros utilizaremos en nuestro proyecto RepRap **PRUSA MENDEL**

Printer	Print Volume (mm)	Resolution	Print Speed	Pref. Material	Price
RepRap Mendel	200 × 200 × 110	0.1mm	150mm/s	3mm PLA	\$830
RepRap Huxley	140 × 140 × 110	0.1mm	150mm/s	1.75mm PLA	\$600
MakerBot Replicator	225 × 145 × 150	0.2mm	45mm/s	1.75mm ABS	\$1,750
MakerGear Mosaic	127 × 127 × 127	0.15mm	75mm/s	1.75mm PLA	\$900
Ultimaker	210 × 210 × 220	0.04mm	300mm/s	3mm PLA	\$1,570
whiteAnt CNC	160 × 190 × 125	0.25mm	35mm/s	3mm ABS	NA
Aleph Objects AO-100	200 × 190 × 100	0.1mm	200mm/s	3mm PLA	\$1,500
Printbot	150 × 150 × 150	0.3mm	25mm/s	3mm ABS	\$550

**Figura 2.2e: Tabla de comparación de impresoras 3D**

## 2.3 SOFTWARE A UTILIZAR

### 2.3.1 BLENDER



**Figura 2.3a: Interfaz principal del programa Blender**

Blender es un programa que sirve para el modelado de objetos en 3D, ayudado por varias herramientas, está orientada para artistas y profesionales del diseño, usado para crear visualizaciones en 3D estáticas de alta calidad.

También incorpora un motor de 3D en tiempo real el cual permite la creación de contenido tridimensional interactivo que puede ser reproducido de forma independiente. Blender se desarrolla como Software Libre, con el código fuente disponible bajo la licencia GNU GPL, su descarga y su uso es completamente gratuito.

Entre sus principales características están:

- ✓ Modelado.
- ✓ Esculpido.

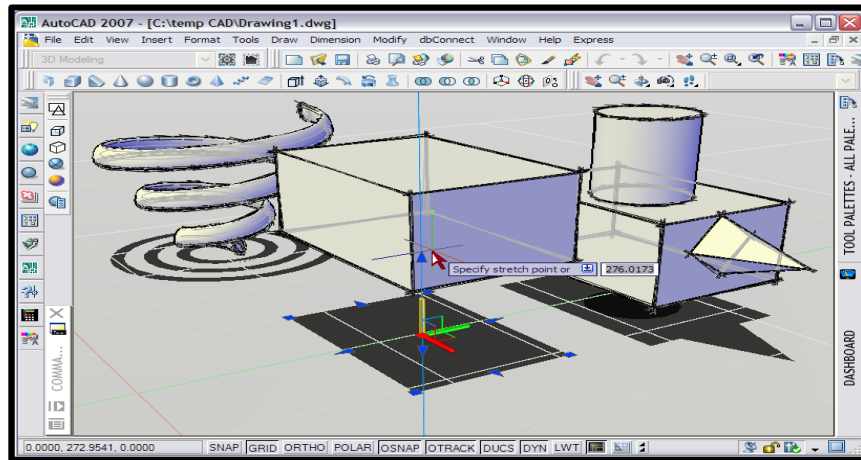
- ✓ Texturizado.
- ✓ Texturizado UV.
- ✓ Pintar UV sobre los modelos.
- ✓ Materiales.
- ✓ Sistema de nodos para las texturas y materiales para mayor complejidad y profesionalismo.
- ✓ Texturizados aplicados a diversos elementos.
- ✓ Sistema de Huesos.
- ✓ Sistema de partículas.

### **2.3.2 PROGRAMA AUTOCAD 3D**

AutoCAD es un programa de diseño mediante el uso de una computadora para realizar dibujos en dos y tres dimensiones. Es desarrollado y comercializado por la empresa Autodesk.

El término AutoCAD surge como creación de la compañía Autodesk, teniendo su primera aparición en 1982. AutoCAD es un software reconocido a nivel mundial por todas sus capacidades de edición, que hacen posible el dibujo digital de planos de edificios o la recreación de imágenes en 3D.

AutoCAD es uno de los programas más usados, elegido por arquitectos, Ingenieros y diseñadores industriales.

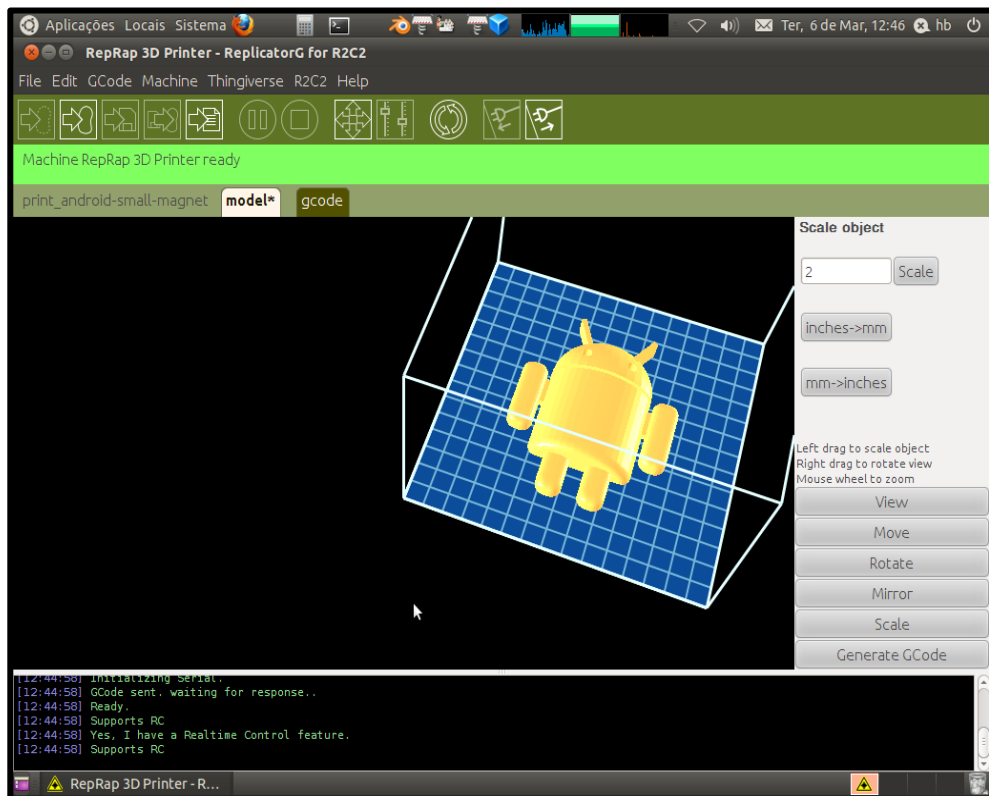


**FIGURA 2.3b: Programa Autocad**

### **2.3.3 PROGRAMA REPLICATOR G**

Este es el software que va a pilotar su replicador de MakerBot, Thing-O-Matic, CupCake CNC, máquina RepRap, o genéricos máquina CNC. Se le puede dar un archivo STL Gcode para procesar.

Es multiplataforma, fácil de instalar, y se basa en los conocidos procesos de Arduino y sus entornos. ReplicatorG es utilizado por miles de operadores MakerBot, y ha impreso decenas de miles de objetos en 3D y contando.



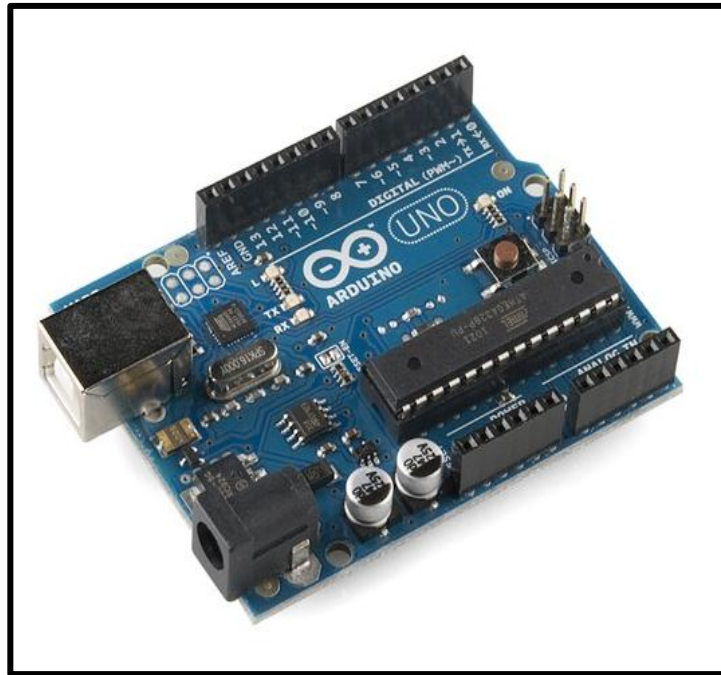
**FIGURA 2.3c: Programa Replicator G**

## 2.3 HARDWARE DEL CONTROLADOR

El controlador Arduino es la base de nuestro proyecto, está encargado de la traducción del código G a instrucciones mecánicas, mediante rutinas específicas que son diseñadas por un programador, consideramos importante dar un resumen de cada una de las partes que lo conforman.



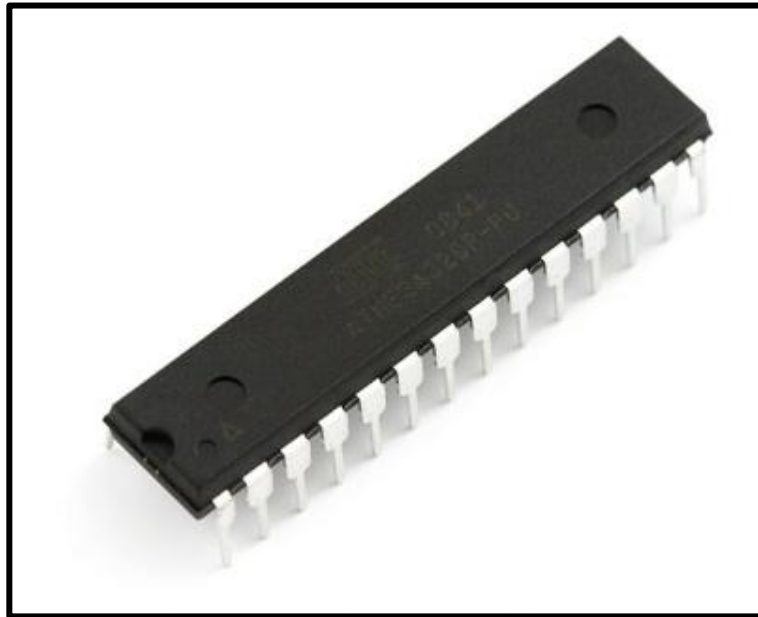
## 2.4.1 TARJETA ARDUINO UNO



**Figura 2.4a: Tarjeta Arduino Uno**

Arduino Uno (**Ver Figura 2.4a**) es una tarjeta electrónica altamente programable de software libre, fácil de usar y muy versátil para diseñar una gran variedad de proyectos electrónicos.

El cerebro de Arduino Uno está basado en el microcontrolador ATmega328, casi todos los demás componentes están encargados con la debida fuente de poder permitir la comunicación del microcontrolador con la computadora.



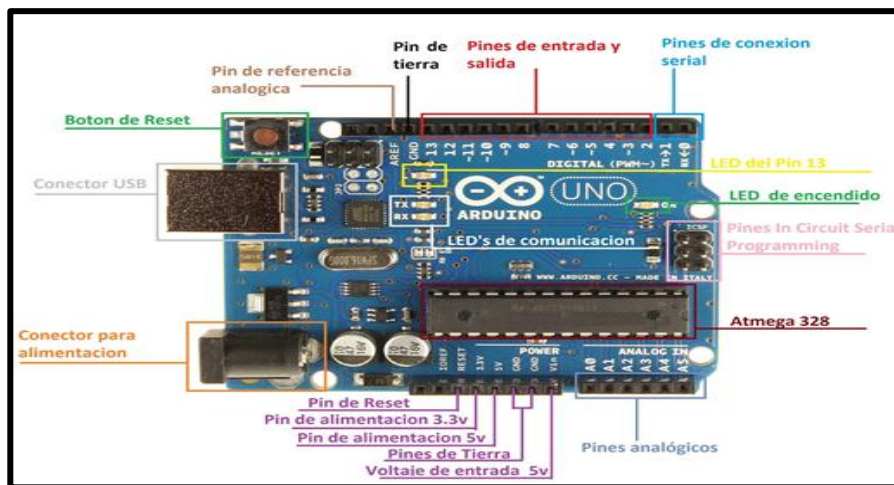
**Figura 2.4b: Microcontrolador Atmega328I**

El ATmega328 (**Ver Figura 2.4b**) es una pequeña computadora integrada en un chip, producido por la compañía Atmel, que posee un procesador mucho mayor que las primeras computadoras que existieron, consta de 28 pines colocado en centro de la tarjeta, posee uno o dos KB de RAM (Random Access memory), y unos cuantos KB de memoria programable (EPROM) que pueden ser utilizados para configurar nuestros programas. Posee entradas y salidas tanto digitales como analógicas, pueden ser programadas como un alto o un bajo (0 volts o 5 volts) y ser utilizadas para el encendido o apagado de diodos leds directamente.

Estos dispositivos no fueron creados para beneficio de amantes de la electrónica, fue adoptado por esta pequeña parte del mercado por su sencilla

programación y versatilidad. Su principal intención era utilizarlo en productos de alto consumo como vehículos, máquinas de lavado, DVD's, y juguetes.

Entre los principales componentes de la tarjeta Arduino podemos destacar los siguientes **(Ver Figura 2.4c)**.



**Figura 2.4c: Componentes Tarjeta Arduino**

### 2.4.1.1 FUENTE DE ALIMENTACIÓN

Se encuentra en la esquina inferior izquierda de la tarjeta, debajo del conector USB **(Ver Figura 2.4c)**, es alimentado por una fuente constante de 5V. Es importante mencionar al regulador de voltaje de 5 voltios (5V) que se encuentra justo encima del conector de alimentación, su función es de regular cualquier voltaje de entre un rango limitado entre 7V y 12V, también alimentado desde un conector constante de 5V.

### 2.4.1.2 CONEXIONES DE ALIMENTACIÓN

Referenciando a la **Figura 2.4c**, en la parte inferior central encontramos los conectores de alimentación. El primero es el botón de reset, sirve para resetear el microcontrolador. Los demás pines son de alimentación y tienen asignados diferentes voltajes (3.3V, 5V, GND y 5V).

### 2.4.1.3 ENTRADAS ANALÓGICAS

Los seis pines (A0...A5) (**Ver Figura2.4c**), son utilizados como entradas analógicas, pueden ser usados para medir voltajes y para la lectura de sensores analógicos, cabe recalcar que solo se puede medir voltaje mas no corriente ya que poseen una alta resistencia interna. Estos pines también cuentan con la capacidad de ser utilizados como entradas o salidas digitales.

### 2.4.1.4 CONEXIONES DIGITALES

En la parte superior de derecha a izquierda se encuentran las 14 entradas digitales (0...13) de la tarjeta (**Ver figura 2.4d**), pueden ser usadas tanto como entradas o salidas. Los pines por defecto son de entrada, por lo que es necesario que sean configurados por medio del comando `pinMode ( )`. En este modo los pines se encuentran en estado de alta impedancia, quiere

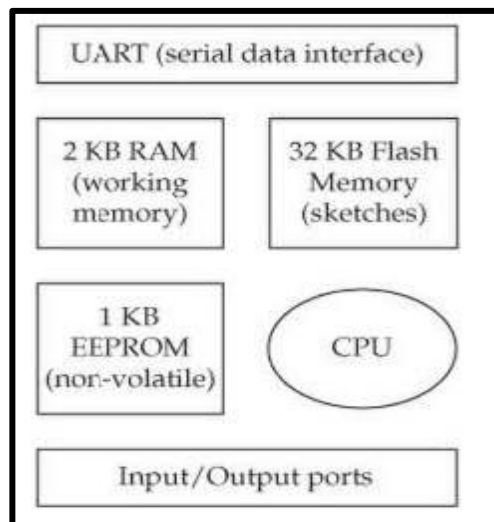
decir que se necesita muy poca corriente para para el pin de entrada de un estado a otro, lo que hace posible el uso de estos pines para tareas como lectura de un LED como un fotodiodo o un sensor capacitivo al tacto

Utilizando a los pines como salidas, estos se encuentran en un estado de baja impedancia. Esto quiere decir que son capaces de suministrar corriente a otros circuitos. Los pines pueden proporcionar una corriente positiva o negativa máxima de 40 mA. a otros dispositivos o circuitos, sin embargo es recomendable usar resistencias de 470  $\Omega$  o 1k, limitando la corriente máxima y así evitar danos a circuitos o transistores a la salida del pin.

#### **2.4.1.5 MICROCONTROLADOR**

El Atmmega323 es un dispositivo rectangular de 28 pines, ubicado en el centro de la tarjeta (**Ver Figura 2.4d**) de fácil reemplazo si es requerido. Este chip es considerado el corazón de la tarjeta, es la unidad central de proceso que controla todos los dispositivos que vienen con el dispositivo. Se encarga de ejecutar todas las instrucciones que estén grabadas en la memoria flash, así como también las que estén configuradas por el usuario dentro de la memoria EPROM. Toda información grabada en la EPROM no se perderá si el dispositivo es reseteado.

El diagrama de bloques (**Ver Figura 2.4d**) muestra las principales características de este dispositivo.



**Figura 2.4d: Diagrama de Bloques Atmega328**

### 2.4.1.6 OTROS COMPONENTES

Posee un oscilador con cristal de cuarzo de 16 MHz, es rectangular y se encuentra por encima del microcontrolador. A la derecha del cristal está ubicado el conmutador de reset, sirve para empezar de nuevo el programa y limpiar la memoria flash.

A la derecha del botón de reset, se puede observar el conector de programación serial (**Ver Figura 2.4d**), es otra de las formas mediante las cuales se puede programar el Arduino. Por el último, en la esquina superior

izquierda se encuentra el conector USB que hace que su uso más sencillo y conveniente para el usuario.

#### **2.4.1.7 ASPECTOS DESTACADOS DEL ARDUINO UNO**

- ✓ Microcontrolador ATmega328
- ✓ 14 entrada/salidas digitales, 6 pueden ser usadas como salidas PWM
- ✓ Posee 6 entradas analógicas.
- ✓ Oscilador de cristal de 16 MHz
- ✓ Conector USB.
- ✓ Microcontrolador ATmega 328.
- ✓ Voltaje de Operación 5V.
- ✓ Voltaje de entrada (recomendado) 7-12 V.
- ✓ Voltaje de entrada (limites) 6-20 V.
- ✓ Corriente máx. por pines de 40 mA.
- ✓ Corriente máx. por el pin de 3.3V de 50 mA.
- ✓ Memoria Flash de 32KB.
- ✓ Memoria estática de acceso aleatorio (SRAM) de 2KB.
- ✓ Memoria EEPROM de 1KB.

## 2.4 PARTES DE LA IMPRESORA 3D

La impresora RepRap PRUSA MENDEL consta de piezas tanto metálicas, como de plástico. Es importante hacer énfasis que todas las piezas de plástico (**Ver figura 2.5a**), son impresas en la misma impresora que estaremos construyendo. Las piezas se construyen con densidades variables para adaptarse a su aplicación



**Figura 2.5a: Partes Plásticas RepRap PRUSA MENDEL**

Entre las principales partes que constituyen a la impresora RepRap PRUSA MENDEL. Tenemos:

### 2.5.1 ROBOT CARTESIANO

La impresora 3D tiende a moverse como un robot cartesiano, esto significa que puede moverse en las tres direcciones lineales, a lo largo del eje x-, y- y

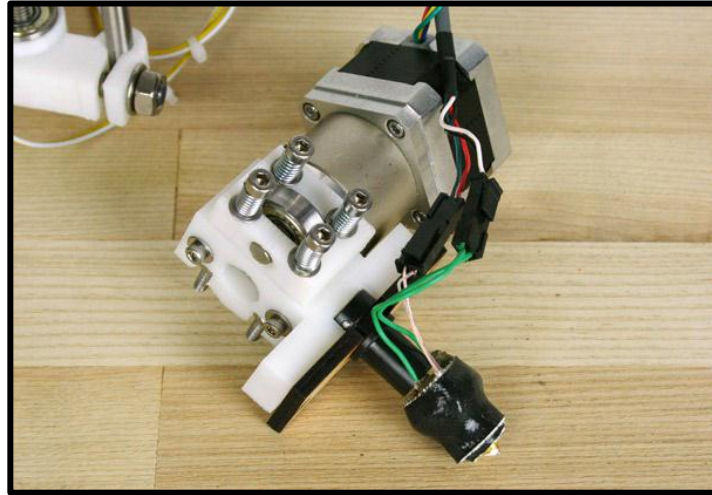


z- también conocidos como planos cartesianos. Para realizar estos movimientos, la impresora 3D utiliza pequeños motores de gran precisión.

El movimiento que realiza un robot de tres ejes, es como cualquier robot que es controlado por códigos numéricos (G-Code), es por esto que es utilizado en las impresoras 3D, siendo capaz de posicionar la máquina de extrusión a lo largo de cada uno de estos ejes lineales de movimiento para establecer capas y capas de plástico caliente.

Los movimientos al realizar el proceso de impresión parecen muy complejos, pero en realidad no es tan complicado, ya que todas las impresoras están estandarizadas. Gracias a las comunidades abiertas de intercambio de información, nuevos y mejores diseños se han compartido libremente generando una mejora notable en esta tecnología.

## 2.5.2 EXTRUSOR (THERMOPLASTIC EXTRUDER)



**Figura 2.5b: Thermoplastic Extruder**

Este dispositivo es capaz de recibir hilos finos de Thermoplastic, un tipo de plástico que se ablanda a un estado semilíquido al extrusor (**Ver Figura 2.5b**), es sin duda una de las partes más complejas de la Impresora 3D que sigue teniendo un desarrollo impresionante, actualmente una de las piezas claves: el filamento de plástico y el extremo caliente termal. La unidad de filamento es jalado a través de los carretes de 3mm o 1,75 mm de diámetro, usando un mecanismo de accionamiento con engranajes. La mayoría de los conductores de filamento, usan un motor de paso para mejorar el flujo de plástico a través del extremo caliente. Estos motores se encuentran dirigidos hacia abajo con engranajes impresos, como se muestra en la **Figura 2.5b**, estos le dan la fuerza necesaria para continuar con la extrusión.

El filamento es arrastrado hacia el extrusor por el controlador de incandescencia, es cuando se alimenta el extremo caliente. El extremo caliente esta térmicamente asilado del resto del extrusor, está compuesto de un bloque de aluminio con un calentador embebido, junto a un sensor de temperatura.

El plástico al atravesar el extremo caliente, es sometido a una temperatura de alrededor de 170 ° C a 220 ° C, dependiendo de las características del plástico. Una vez que le plástico alcanza un estado semilíquido, es forzado a atravesar por una boquilla de impresión de unos 0,35 milímetros a 0,5 milímetros de diámetro, antes de colocar esta delgada extrusión caliente sobre el tablero de impresión, se describe una capa que dará forma a lo que se desee imprimir utilizando algún patrón de relleno.

### 2.5.3 TABLERO DE IMPRESIÓN (HEATED PRINTBED)



**Figura 2.5c: Heated PrintBed (Tablero de Impresión)**

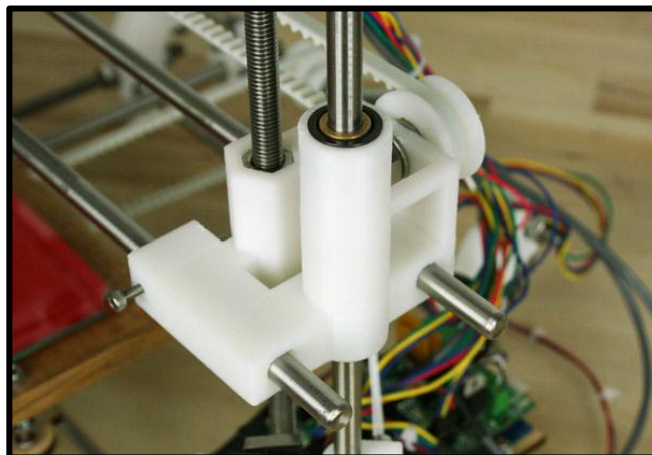
La superficie donde se realizan las impresiones, es llamada PrintBed (**Ver Figura 2.5c**), se usan a menudo dos materiales para su fabricación, vidrio o aluminio, para mejorar la difusión de calor a través de zona y para obtener una superficie lisa y nivelada.

El vidrio ofrece una superficie más lisa mientras que el aluminio conduce mejor el calor a través de una superficie climatizada. Para evitar deformaciones en la superficie de impresión, a menudo se cubren con un tipo de cinta de un material que sea barato cambiarlo periódicamente.

## 2.5.4 LINEAS DE MOVIMIENTO

El sistema de movimiento lineal, permite a los ejes del sistema mecánico moverse, a menudo es quien determina que tan precisa es la impresora 3d, o la velocidad de su impresión o la cantidad de mantenimiento que necesitara la impresora a largo plazo.

La mayoría de las impresoras utilizan sistemas lisos, varillas de tierra para cada eje y los rodamientos de bolas de plástico o bronce para deslizarse a través de cada varilla. Los rodamientos de bolas han ganado mucha popularidad por su longevidad y funcionamiento suave sobre la vida útil de la impresora, son más fuertes durante la impresión los casquillos de bronce **(Ver Figura 2.5d)**, pero requieren mayor trabajo para alinearse al momento del proceso de construcción.



**Figura 2.5d: Rodamientos de bronce en un PRUSA MENDEL**

## **CAPÍTULO 3**

### **DESARROLLO FINAL DEL PROYECTO**

#### **3.1 RESUMEN DEL CAPITULO**

En este capítulo detallaremos el proceso con el cual se ha completado nuestro proyecto, combinando todas las etapas necesarias para la construcción de nuestro controlador, para manejar el proceso de impresión.

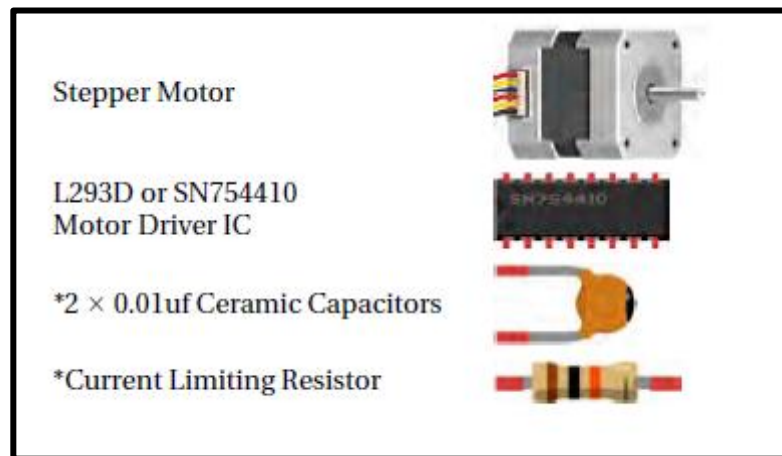
Se definirá el proceso del diseño del controlador basado en la tarjeta Arduino con la combinación de la tarjeta Ramp, los drivers encargados del control de los motores de paso, además la configuración en la tarjeta Ramp de los pines

de control para los diferentes sensores de los finales de carrera y el control de temperatura del extrusor.

Comprenderemos mejor al Arduino mediante un ejercicio práctico, con el cual se nos facilitara el manejo de esta tarjeta, para poder interpretar de una mejor manera la configuración de las conexiones entre el Arduino y la impresora 3D.

### **3.2 EJERCICIO PRÁCTICO: CONTROL PARA MOTOR DE PASO**

Mediante el uso de la tarjeta Arduino se puede construir un controlador para un motor de paso de manera sencilla. A continuación se detalla el proceso, se puede crear fácilmente un controlador con el cual se pueda mover los motores en diferentes direcciones, manejar su velocidad y establecer el número de pasos del motor.



**Figura 3.2a: Circuito de Prueba de Motores, Componentes Electrónicos**

La tarjeta Arduino se conecta mediante una fuente DC externa, el uso de los capacitores en esta configuración es opcional pero debido a que estos ayudan a evitar interferencias con el microcontrolador es recomendable no omitirlos, estos están conectados entre +5v y Gnd, además se requiere de una resistencia entre el pin de Vin del Arduino y el pin de entrada de voltaje del L293D, los pines 4,5,6 y 7 del Arduino van conectadas con los pines 1, 2,3 y 4 del driver que controla el motor, los pines de salida 1 y 2 del driver del motor deben ser conectados con la bobina 1 del motor y las salidas 3 y 4 deben ir conectadas con la bobina 2.

Se recomienda leer el respectivo datasheet del motor para verificar los colores correspondientes de los cables a la bobina 1 o 2.



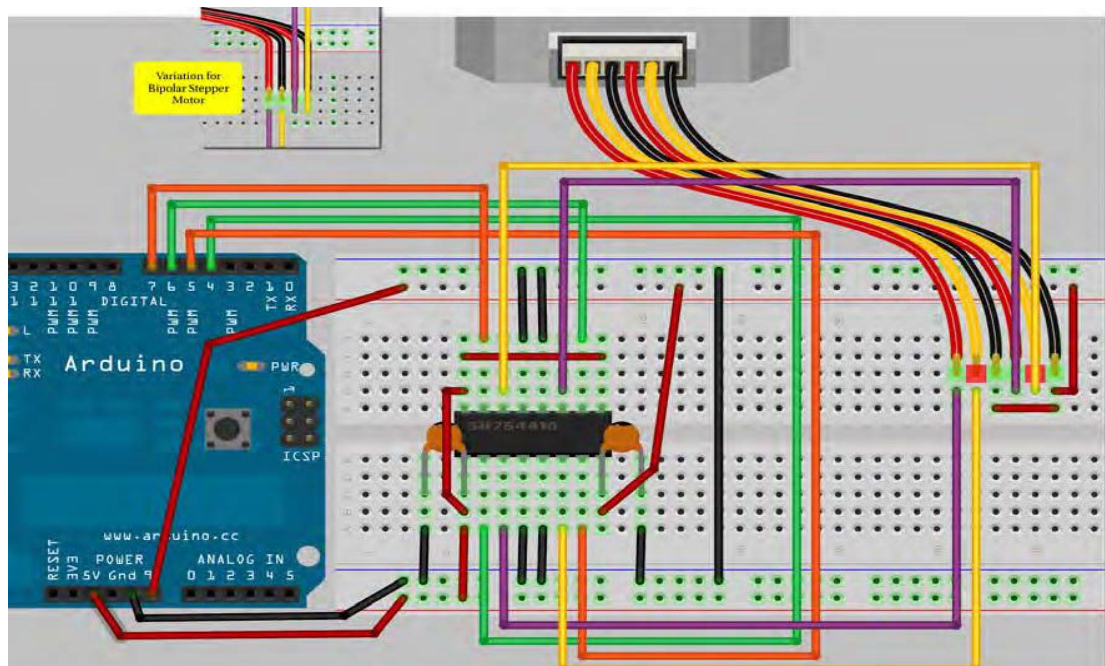


Figura 3.2b: Controlador de motor de paso

### 3.2.1 CODIGO FUENTE

```
#include <Stepper.h>

// steps value is 360 / degree angle of motor

#define STEPS 200
// create a stepper object on pins 4, 5, 6 and 7

Stepper stepper(STEPS, 4, 5, 6, 7);

void setup()

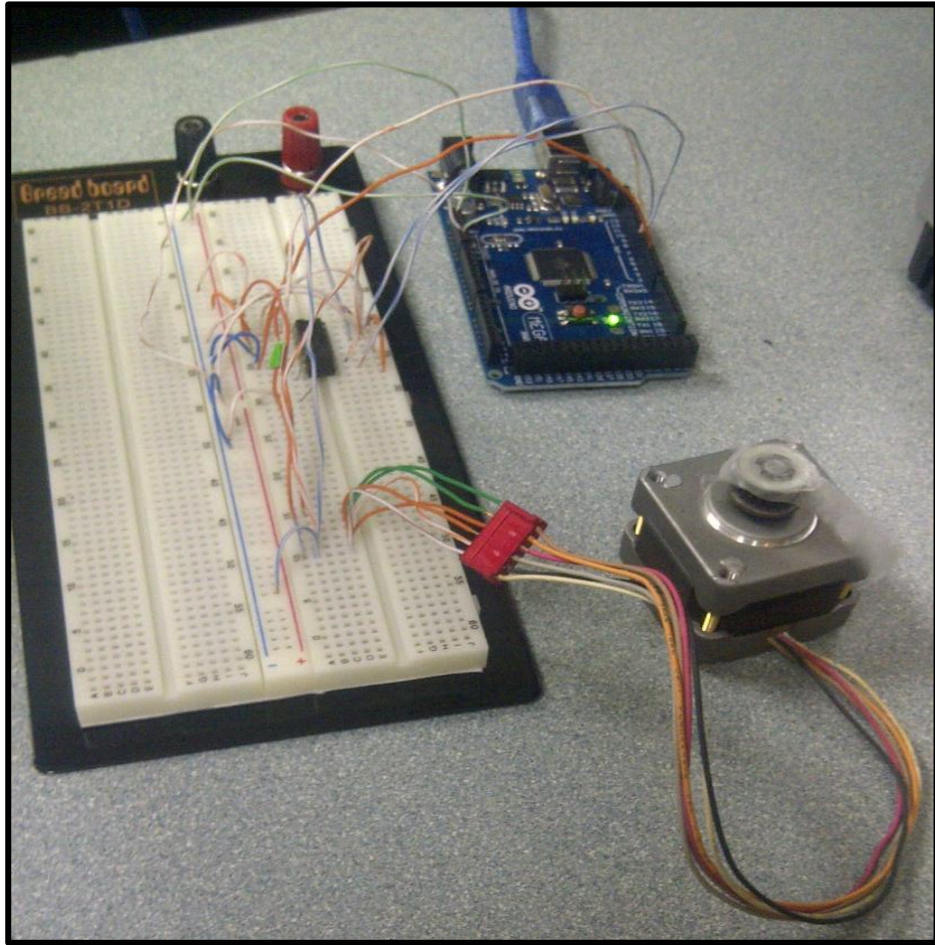
{

}
```

```
void loop()
{
  stepper.setSpeed(60);
    stepper.step(200);
      delay(100);
        stepper.setSpeed(20);
          stepper.step(-50);
            delay(100);
```

### 3.2.2 IMPLEMENTACIÓN EN PROTOBOARD

A continuación mostramos el diseño del circuito implementando en protoboard (**Ver Figura 3.1c**), con el cual pudimos realizar la prueba de los 4 motores de paso, verificando la funcionalidad de cada uno de ellos, así mismo se realizaron pruebas para medir la intensidad de corriente necesaria para que los motores puedan operar, evitando riesgos de tener sobrecargas y el deterioro de alguno de los circuitos internos de la tarjeta controladora.



**Figura 3.2c: Implementación de Controlador de motor de paso**

Los motores de paso trabajan a una frecuencia de 60hz y son controlados por instrucciones previamente configuradas en el Arduino, este práctico ejercicio se enfoca en los movimientos bidireccionales de los motores.

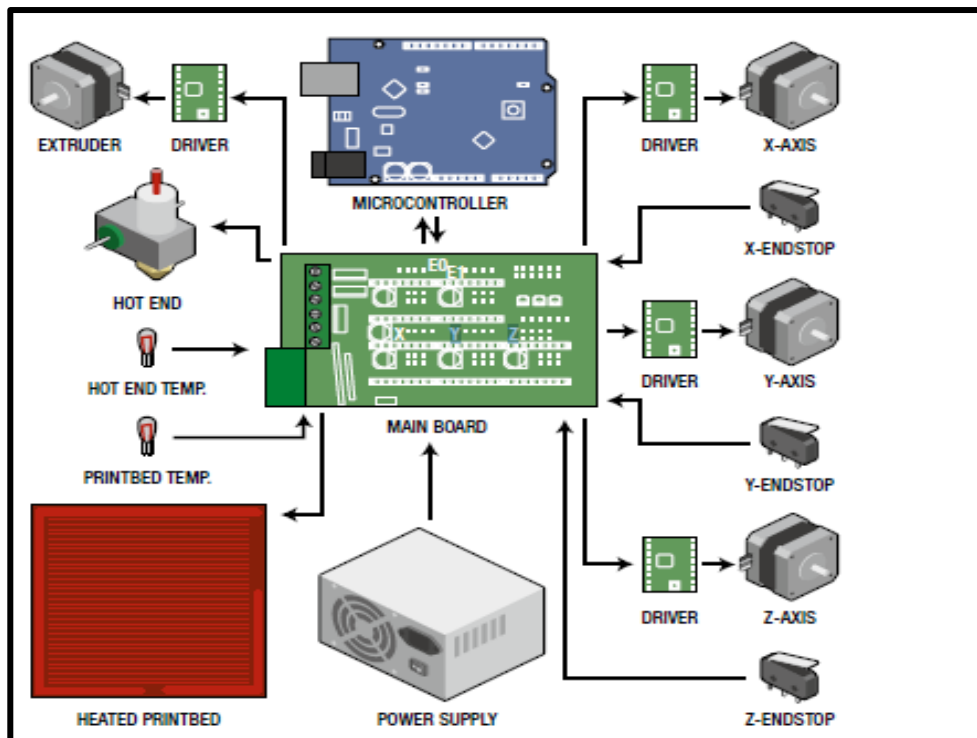
También nos ayuda a tener una mejor percepción de la programación de las rutinas que usa el Arduino y la configuración que se debe usar entre los motores de paso y la tarjeta Ramps.

### 3.3 DISEÑO DEL CONTROLADOR PARA EL PROYECTO REPRAP

La parte fundamental de nuestro tema se enfoca en este capítulo, el diseño del controlador que se divide en varias etapas. A continuación se detallara la implementación que se debe realizar entre la tarjeta Arduino y la Ramps, el proceso de programación previo que se debe que realizar a la tarjeta Arduino, la instalación del software de Arduino y el firmware llamado **Sprinter** el cual nos ayudara con la comunicación entre el Arduino y la Impresora 3D.

Para el control de los motores, finales de carrera y temperatura del extrusor se utiliza la combinación de las dos tarjetas Arduino + ramp 1.2, esta combinación es necesaria para completar la configuración necesaria de nuestro controlador, en la tarjeta Ramps están los pines destinado para conectar los drivers los cuales son microcontroladores más pequeños que se encargan del control de los motores paso a paso.

### 3.3.1 CONEXIONES PARA LA RAMPS 1.2

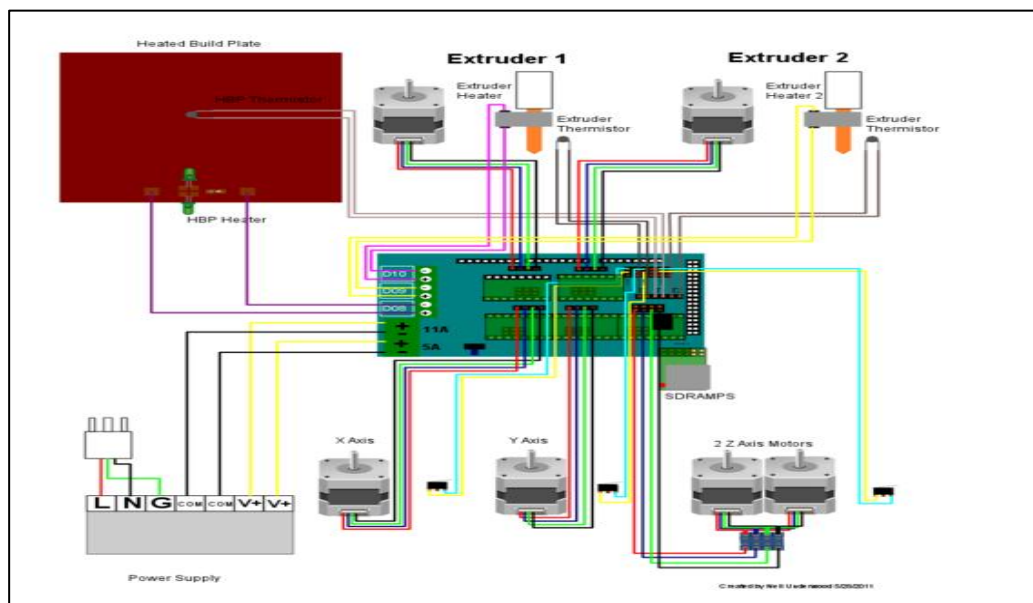


**Figura 3.3a: Esquema del Arduino + Ramps**

En la **Figura 3.3a** podemos observar el esquema de conexión para el controlador de la impresora 3D, está basada en el uso de la tarjeta Ramps 1.2, esta es la alternativa propuesta a diseñar, con los respectivos cambios en la configuración de las conexiones, el uso de finales de carrera mecánicos por los ópticos que vienen por defecto y el uso de un software especializado para crear una interface con el usuario, nos permite el uso de nuestra impresora 3D.

Tenemos cuatro drivers para motores de paso, comúnmente conocidos como Pololus, al lado de estos tenemos cuatro pines donde se conectan los cables que salen de cada motor.

Los cuatro drivers nos permiten controlar el movimiento del motor del X, el del eje Y, los dos del eje Z y el motor del extrusor, es el encargado de mover el filamento de plástico para que este se caliente, los motores del eje Z tienen que ir, como se muestra en el esquema (**Ver Figura 3.3b**), debido a que en la tarjeta Ramps se muestran dos sockets de pines para conectarlo, si son conectados en una posición errónea los motores no desempeñarán su función correctamente, se debe tener cuidado al momento de realizar las conexiones en cuanto a la polaridad, ya que esto causaría que los motores giren al lado contrario del requerido.

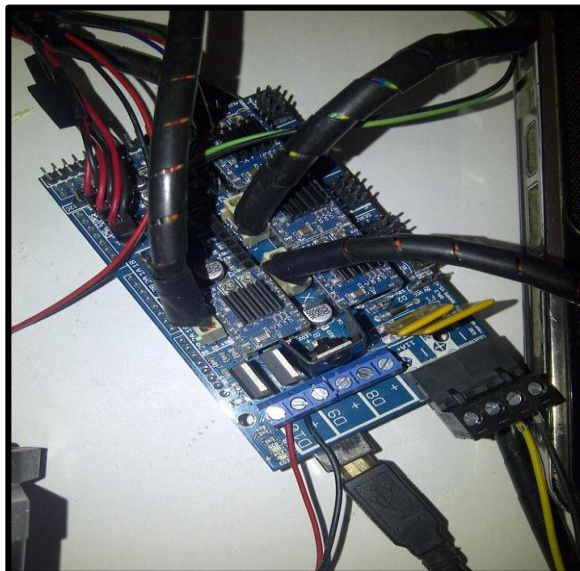


**Figura 3.3b: Esquema de conexión para la Ramps**

También se muestran cuatro pines los cuales dos son para cada termistor, uno de ellos es el de la cama caliente (**heatedbed**) y el otro corresponde al extrusor, pero en este caso nosotros no contamos con una (**heatedbed**) así que solo usamos los pines para el extrusor.

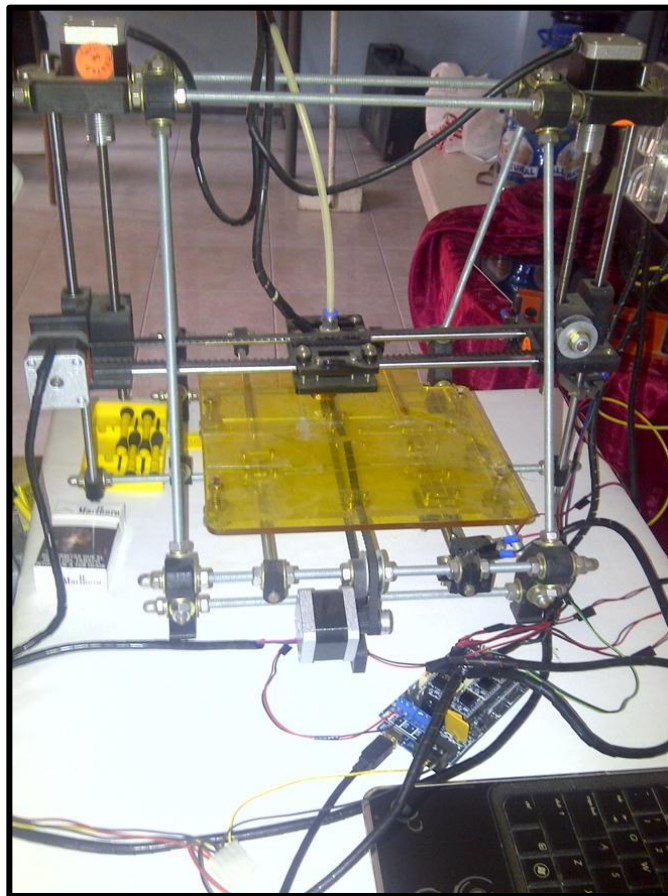
Para las conexiones de los termistores se utilizan también conectores del tipo servo, pero para nuestro caso particular usamos los finales de carrera, tenemos un conector de tres terminales y para los termistores, se pueden usar dos conectores de dos terminales cada uno.

La tarjeta Ramps opera con 12 VDC, y el rango de corriente para que opere es de 5-6 amperios, se debe tener mucho cuidado al realizar las conexiones y no dejar ningún cable su debido asilamiento para evitar accidentes.



**Figura 3.3c: Tarjeta Ramps**

Podemos apreciar en la **Figura 3.3c**, la tarjeta Ramps en operación con sus respectivas conexiones de voltaje y de corriente, también podemos observar los conectores de cada motor en su respectivo socket, y el conector USB que nos brinda la interface con el computador.



**Figura 3.3d: Impresora Prusa Mendel con Tarjeta Ramps**

Podemos observar en la **Figura 3.3d** la impresora ensamblada en funcionamiento con la tarjetas Ramps.

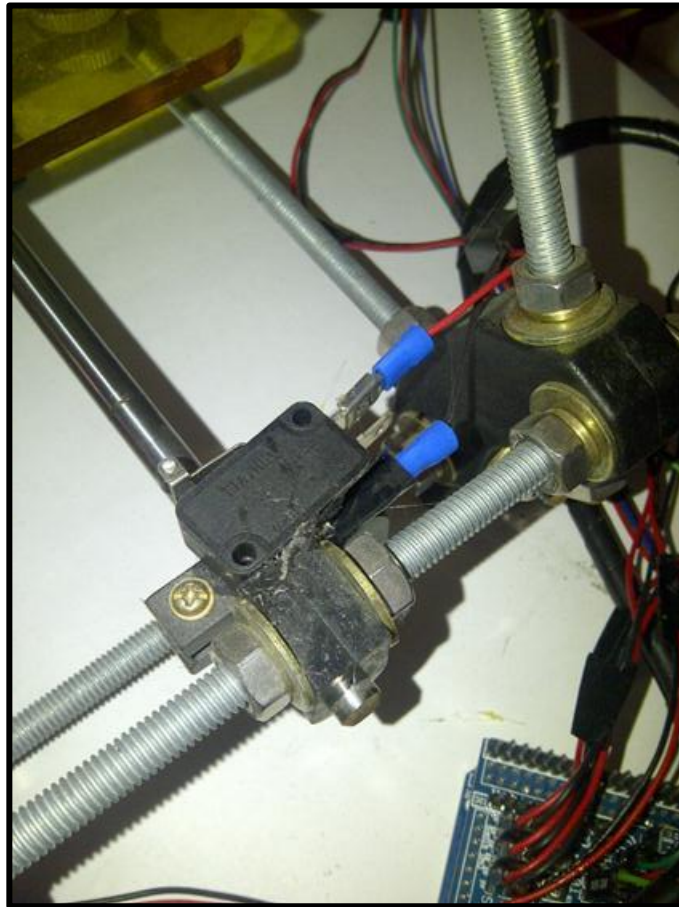


### **3.4 MODIFICACIÓN EN LA IMPRESORA 3D**

Al utilizar un nuevo controlador, se debe tener en cuenta que se deben realizar modificaciones en las conexiones y la estructura de la impresora Prusa Mendel, que sean capaces de acoplarse al nuevo controlador, estos son cambios necesarios debido a que las conexiones por defecto de fábrica no se ajustan al nuevo controlador, diseñado en base al microcontrolador Arduino y la Tarjeta Ramps. A continuación se detallan los cambios realizados para que la impresora opere correctamente.

#### **3.4.1 FINALES DE CARRERA O END STOP**

Los finales de carrera que vienen con la impresora de fábrica son sensores ópticos, estos con su controlador original funcionan perfectamente, pero no se acoplan con el nuevo controlador diseñado en nuestro proyecto, ya que los ópticos constan de cuatro pines y en la tarjeta Ramps utilizada solo posee dos pines, este es el principal motivo por el cual se deben reemplazar los finales de carrera ópticos por finales de carrera mecánicos, estos constan con el número de pines correcto para la conexión con la tarjeta Ramps.



**Figura 3.4ª: Finales de carrera Mecánicos**

Hay que tomar en cuenta que para los finales de carrera mecánicos debemos hacer una pequeña adaptación en la estructura de la impresora, los finales de carrera mecánicos se remplazan en los ejes X, Y, Z también debemos tomar en cuenta que estos estén correctamente calibrados debido a que estos son menos sensibles que los ópticos y podrían causar problemas si no son correctamente instalados.

### 3.4.2 CONEXIONES DEL EXTRUSOR

Otro cambio necesario está en el extrusor debido a que originalmente este cuenta directamente con cuatro pines, pero en el diseño de nuestro controlador se dividen las conexiones dos pines para sensar la temperatura y dos pines que calientan la punta del extrusor.

En el diseño de este controlador con el microcontrolador Arduino y la tarjeta Ramps nos damos cuenta que los pines del termistor y del calentamiento del extrusor son diferentes por lo tanto tuvimos que establecer que cables pertenecían a cada una de estas operaciones y con sumo cuidado dividir los cables para sus respectivos pines en la tarjeta Ramps. **(Ver Figura 3.3b).**

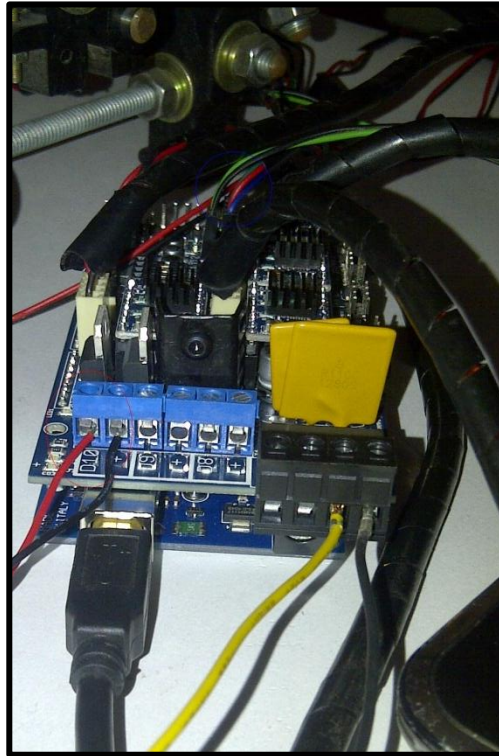
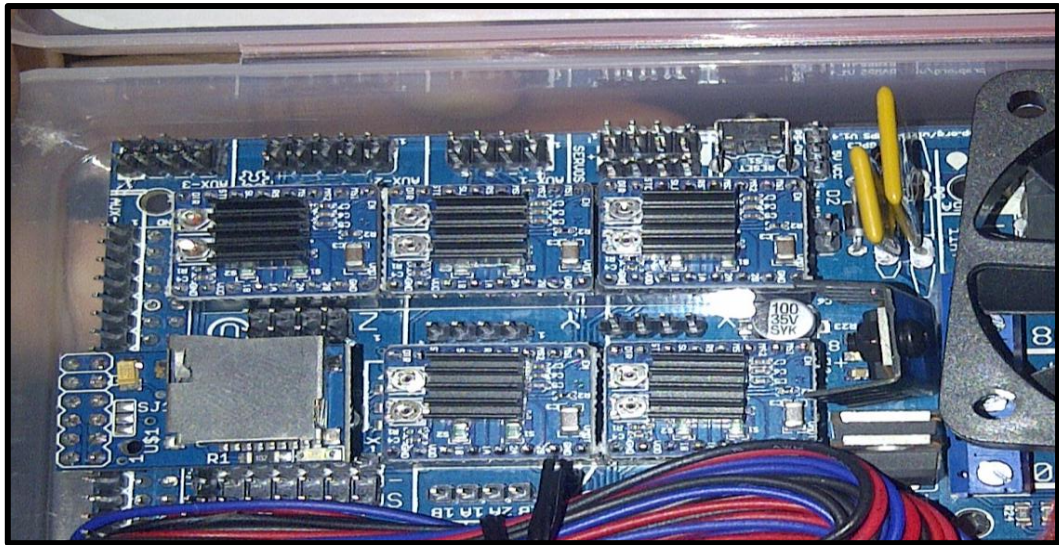


Figura3.4b: Conexiones del extrusor

### 3.4.3 CALIBRACIÓN DE DRIVERS

Esta debe ser la principal calibración ya que los drivers (**POLOLUS**) se encargan del movimiento de los motores, estos tienen que moverse de manera precisa en el momento de la impresión.



**Figura 3.4c: Drivers en la tarjeta Ramps (POLOLUS)**

En la **Figura 3.4c** podemos observar los drivers, estos constan con dos potenciómetros, el primero controla la cantidad de corriente que le llega al motor, es decir si el potenciómetro no permite el paso la cantidad de corriente necesaria para que el motor opere, las bobinas internas de los motores no tendrán la fuerza suficiente para moverse.

Se hicieron las respectivas pruebas para medir la intensidad de corriente que llega a la tarjeta Ramps sin tener ningún motor conectado, se registraron valores de 80 mA. Con un motor conectado se obtuvo un valor de 280 mA. para que opere sin problema alguno, es decir se determinó que cada bobina se alimenta con 200 mA. para estar en su rango de operación. El segundo potenciómetro se encarga de controlar la frecuencia del motor.

### **3.5 SOFTWARE DEL ARDUINO Y FIRMWARE**

Una vez que se ha concluido con el ensamblado de la impresora 3D y el diseño del controlador con toda su electrónica vamos a proceder con la programación del Arduino y la instalación del software necesario para la calibración e interpretación.

Se debe realizar la instalación del firmware adecuado al microcontrolador Arduino y así realizar la correcta configuración, también es necesario que instalemos en el computador con la cual vamos a dirigir los movimientos y rutinas de la impresora 3D, el software correspondiente y un programa que se encargue de generar lo que se denomina “capeado” o modelado por capas es decir, que transforme los diseños 3D previamente diseñados que vamos a imprimir a un formato que represente ese diseño por capas para que pueda ser creado por la impresora ese formato se denomina código G, más conocido como G-code.

## 3.5.1 PROCESO PARA CARGAR FIRMWARE EN ARDUINO

### 3.5.1.1 DESCRIPCIÓN

El firmware, entre los que podemos elegir, para instalar en el microcontrolador de nuestra Prusa Mendel están el **Sprinter** y **Marlín**, estos son programas libres es decir que nosotros podemos descargarlos y editarlos libremente. Podemos encontrarlos en diversos repositorios y páginas de internet, en este caso nosotros optamos por el **Sprinter** como firmware de nuestro controlador ya que ofrece múltiples posibilidades que con el **Marlin** no se podrían lograr.

Entre estas tenemos como las principales: establecer un control PID sobre las temperaturas del termistor de la impresora, para que así esta no esté permanentemente oscilando alrededor de la temperatura de referencia que le hayamos puesto en la configuración de software, también nos permite un control de la velocidad de los ejes con aceleración gradual.

Este Firmware nos permite calibrar los ejes de manera que se muevan lo más precisos posible y la cantidad que les indiquemos desde la computadora de forma manual y, por tanto, las cantidades que deban moverse durante una impresión y la cantidad de material que se utiliza, así el detalle de la impresión es de una calidad más alta.

### 3.5.1.2 INSTALACIÓN

Para instalación del firmware en el microcontrolador, necesitamos previamente descargarnos el software de Arduino, que lo podemos conseguir gratuitamente en la página de Arduino, el procedimiento a seguir es similar para cualquier firmware utilizando cualquier sistema operativo (Windows, MAC o Linux).

Una vez tengamos en nuestra computadora el firmware como el software de Arduino debemos seguir las siguientes instrucciones:

- 1 Descomprimos el archivo que hemos descargado de la web de Arduino
- 2 Arrancamos el programa.
- 3 Seleccionamos en la pestaña “Tools/Board” de la parte superior, el modelo de Arduino que estemos utilizando

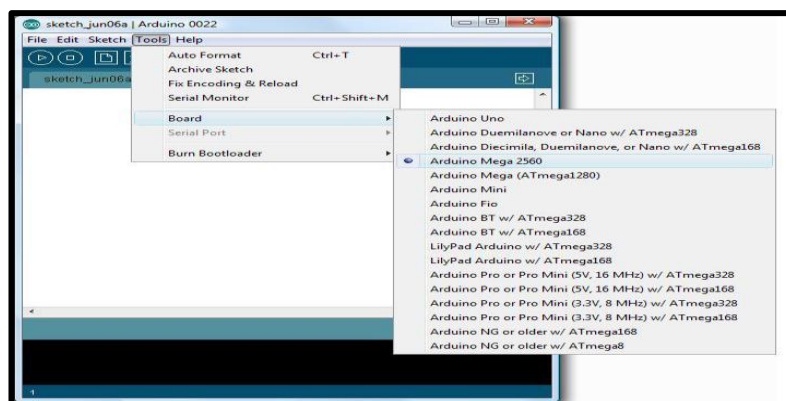
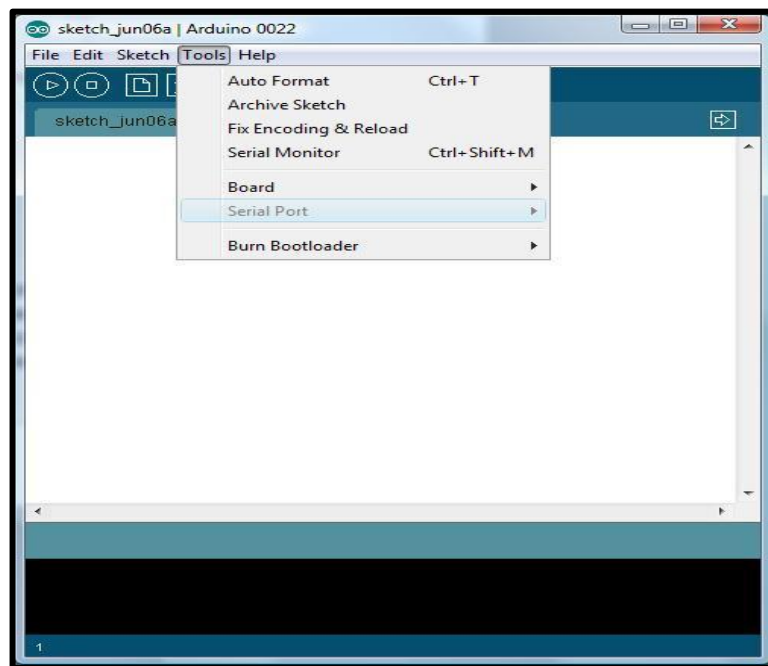


Figura 3.5a: Instalación del Arduino

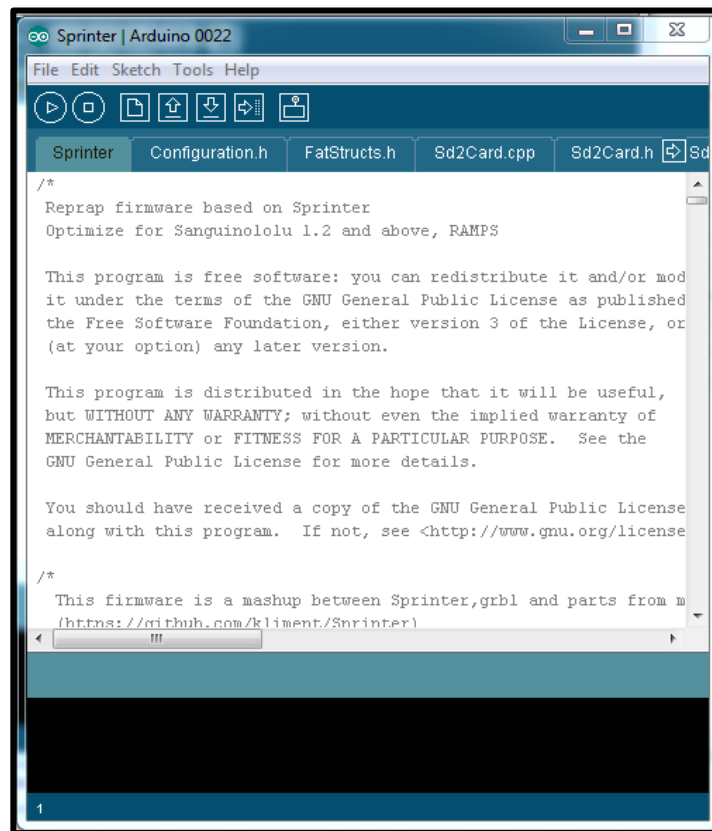


- 4 Seleccionamos en la pestaña “Tools/ Serial Port”, el puerto USB en el que está conectado el microcontrolador.



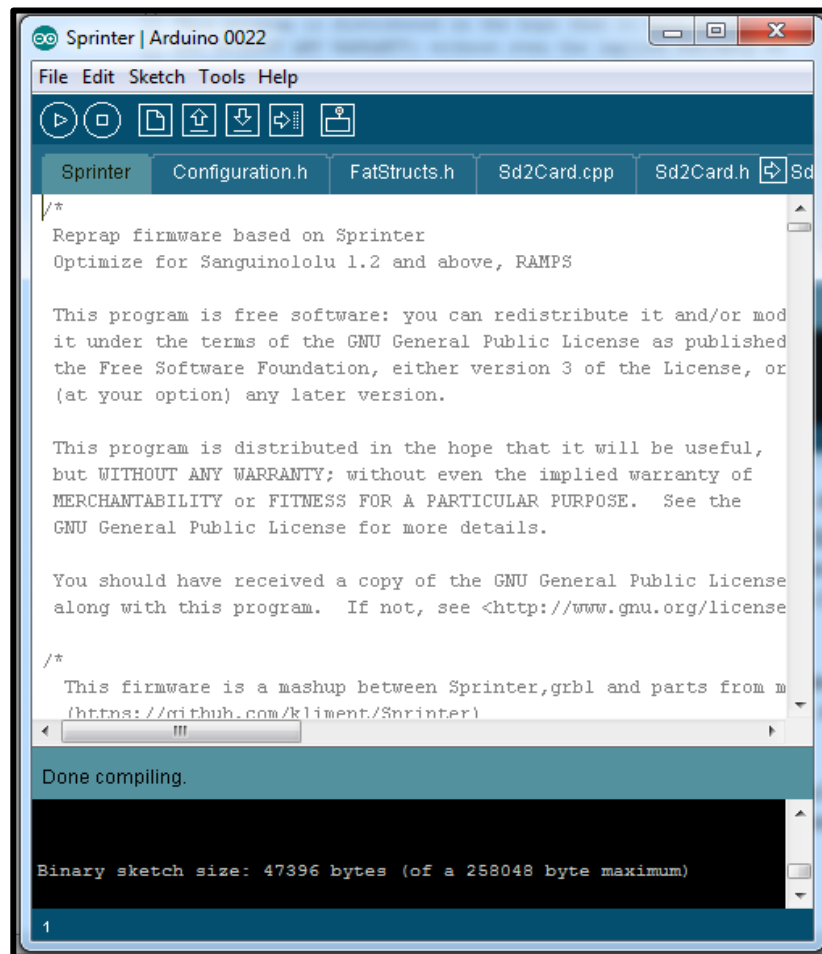
**Figura 3.5b: Selección del puerto serial**

- 5 Hacemos click en “File/Open” y abrimos el archivo .pde, que se debe encontrar en la carpeta descargada del repositorio web con el firmware.



**Figura 3.5c: Selección del Firmware para el Arduino**

- 6 Configurado adecuadamente el firmware debemos compilar el código para ver que no hayamos cometido errores. Para ello podemos hacer click en “Sketch/Verify /Compile”, podemos hacer click en el icono con la flecha de play o, simplemente, podemos usar el comando rápido, que es pulsar las teclas Ctrl+R.



**Figura 3.5d: Compilación del Código**

- 7 Con el código compilado ya podemos cargarle el firmware a nuestro microcontrolador. Para ello podemos hacer click en “File/Upload to I/O Board”, podemos hacer click en el icono con una flecha que indica hacia la derecha, y que al pasar el puntero por él aparece un mensaje a la derecha que dice “Upload”; o, simplemente, podemos usar el comando rápido, que es pulsar las teclas Ctrl+U.

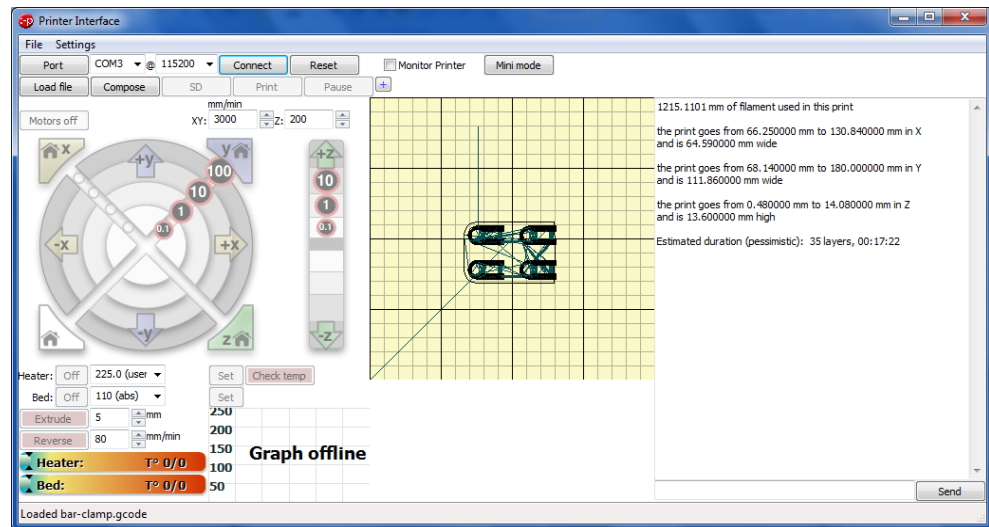
### 3.5.2 SOFTWARE DE CONTROL

Teniendo el firmware instalado, el siguiente paso es instalar el software de control en el computador, con la cual se controlara la impresora, en nuestro proyecto se utiliza principalmente el **Pronterface**, el nuestro es el **3Dstuffmaker Easy Print** el cual funciona perfectamente con los modelos Prusa Mendel.

El **Pronterface** es un software que nos permite acceder directamente a la configuración del programa de creación de archivos Gcode, para editarlo y calibrar las impresiones a nuestro antojo, tiene un entorno gráfico que nos permitirá manipular los ejes de nuestra impresora, calentar el extrusor, monitorear la temperatura, además se puede controlar la velocidad de la impresión y muestra el tiempo de duración de cada impresión.

#### 3.5.2.1 INSTALACION DEL SOFTWARE DE CONTROL

La instalación de cualquiera de este software es instantánea, simplemente lo descargamos, se corre el archivo ejecutable y ya lo tenemos. Aunque se requiere una buena versión de Phyton, sobre todo para usar **Pronterface**.



**Figura 3.5e: Interfaz Software PronterFace**

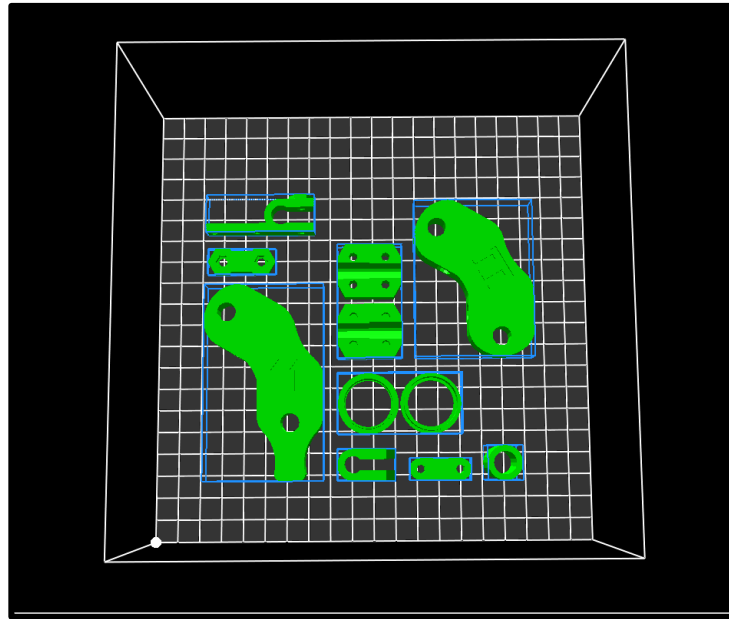
## **CAPÍTULO 4**

### **SIMULACIONES Y RESULTADOS**

#### **4.1 RESUMEN DEL CAPÍTULO**

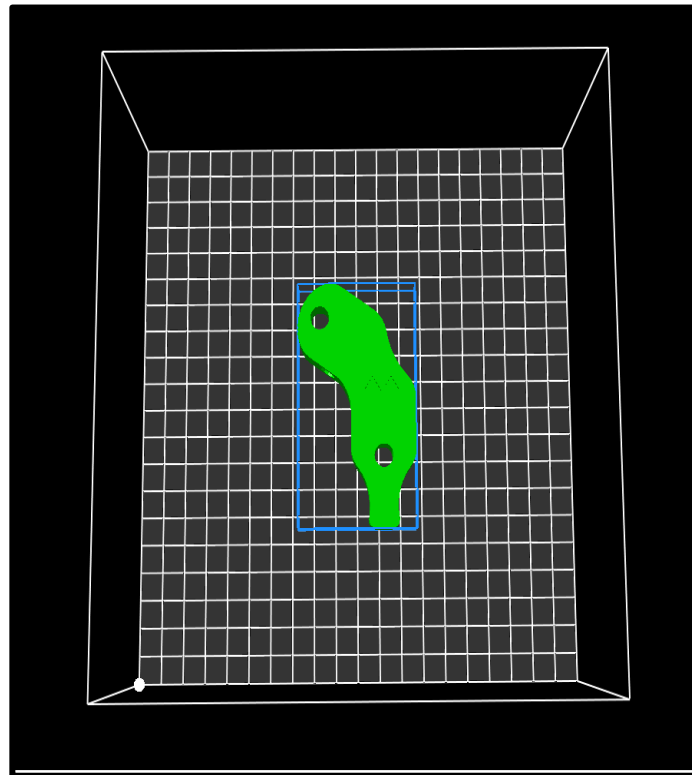
En este capítulo mostraremos gráficamente el proceso de impresión mediante el uso de los programas previamente mencionados, nos enfocaremos en la impresión de las partes principales de una impresora, como fue con anterioridad nuestro proyecto tiene como principal aplicación la replicación de las partes, también conoceremos paso a paso los procedimientos y etapas que se realizan antes y durante la impresión para obtener como resultado el objeto 3D impreso.

A continuación se muestran una porción de las partes de la impresora 3D que pueden ser replicadas.



**Figura 4.1a: Modelado 3D de Partes de Impresora**

El proceso de impresión es el mismo para todas las partes de la impresora así como también para cualquier diseño que fuese modelado en 3D, para nuestro estudio y análisis hemos seleccionado una parte fundamental de nuestra impresora, uno de los soportes o mejor conocido como frame-vertex-foot, la impresora 3D consta de cuatro, uno en cada esquina y nos sirve para fijar a la impresora a una superficie plana.



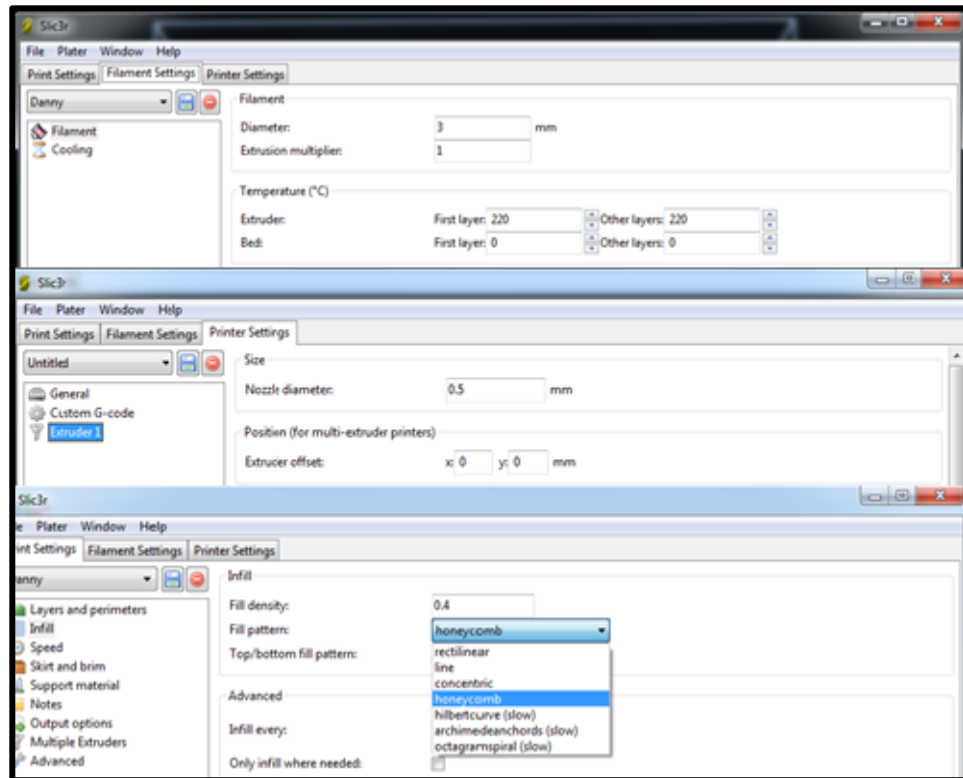
**Figura 4.1b: Vertice o frame-vertex-foot**

## **4.2 SIMULACION SLICER**

Es importante mencionar que para realizar cualquier simulación es necesario tener el modelado 3D en archivos .STL o a su vez tener el código G del objeto. Estos archivos son fáciles de reconocer por su extensión .gcode, se los genera a partir de un archivo .CAD o .STL, el software que usamos para realizar este procedimiento es conocido como SLICER.



## 4.2.1 CONFIGURACION DE PARAMETROS



**Figura 4.2a: Configuración de Parámetros SLICER**

Para obtener una buena simulación y por consiguiente una buena impresión, se deben establecer de forma correcta los parámetros de impresión, tales como diámetro de filamento, diámetro de la boquilla del extrusor, temperatura a la cual se va a trabajar y por último el tipo de relleno que se utilizara en la impresión **Ver Figura 4.2a**, todos estos parámetros determinaran el número de capas que se debe seccionar el objeto y la cantidad de material requerida para la impresión.

### 4.3 SIMULACION CON G-WIZARD

G-Wizard es uno de los principales simuladores de impresión 3D, altamente utilizado por sus características técnicas, nos permite tener una simulación precisa y detallada. Su precisión se basa fundamentalmente en su perfecta utilización de algoritmos para la lectura del código G del modelo 3D, haciendo que los usuarios seamos capaces de tener una predicción bastante exacta de cómo será nuestra impresión.

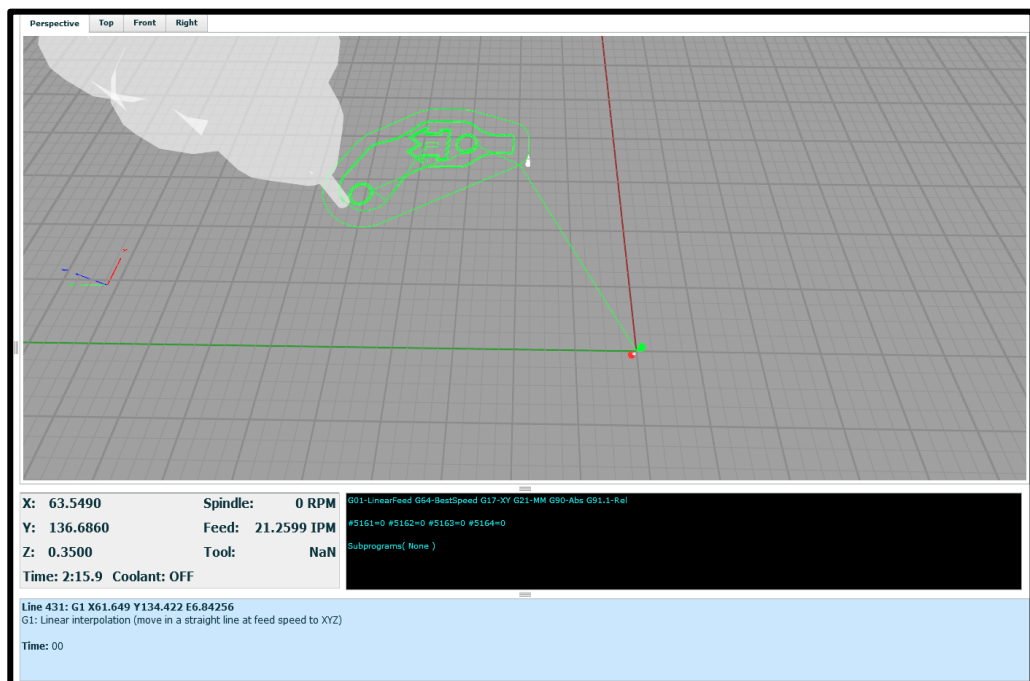
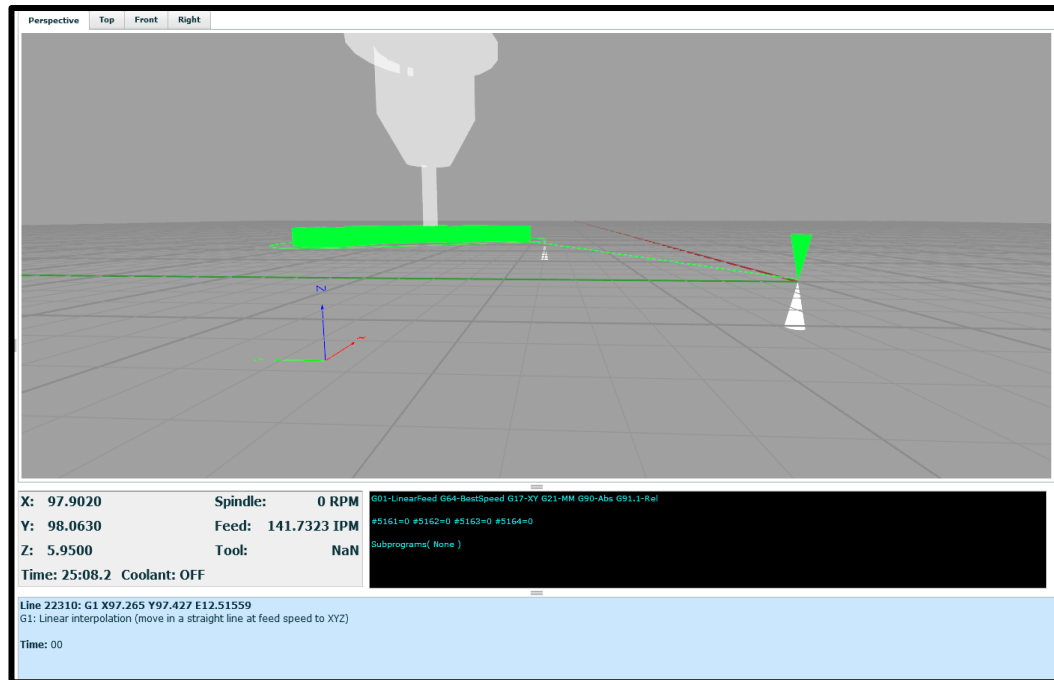


Figura 4.3a: Simulación G-Wizard

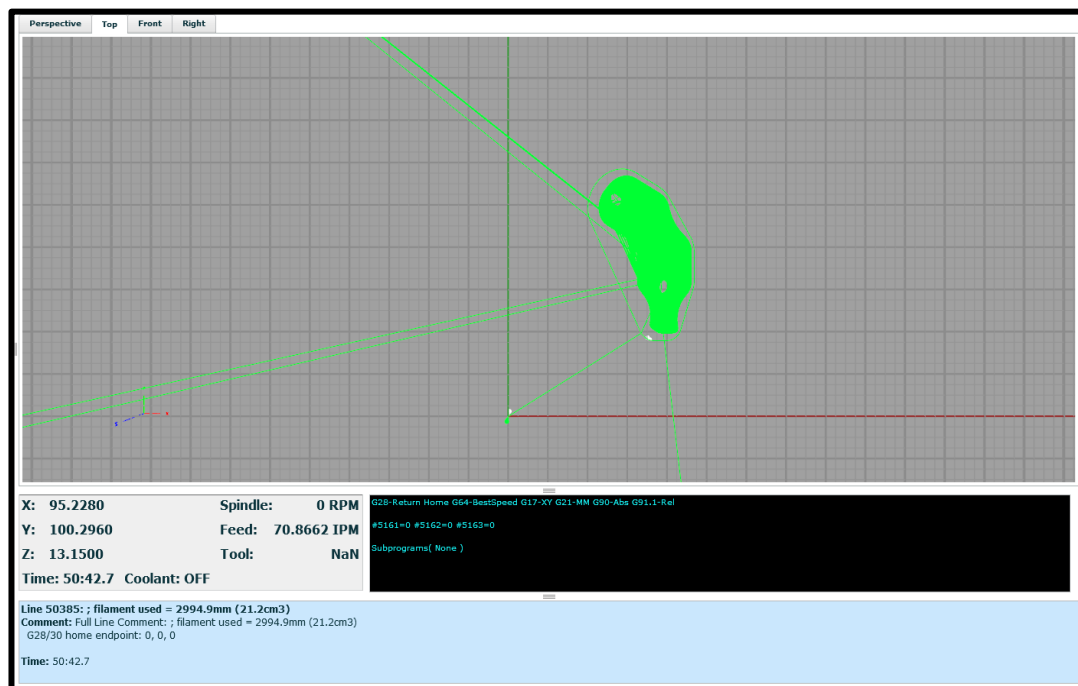
En la **Figura 4.3a**, podemos apreciar las primeras capas de impresión, también se puede visualizar la posición exacta del extrusor en sus coordenadas X, Y y Z, en un tiempo estimado que también es mostrado por el simulador.



**Figura 4.3b: Perspectiva Lateral plano XY**

Esta vista lateral nos permite observar el desarrollo de la impresión en el plano XY **Ver Figura 4.3b**, las capas van aumentando conforme el plano Z aumente, el código G para este modelo es de aproximadamente de unas 50k líneas, por lo que podemos notar que está en 40% de su impresión.

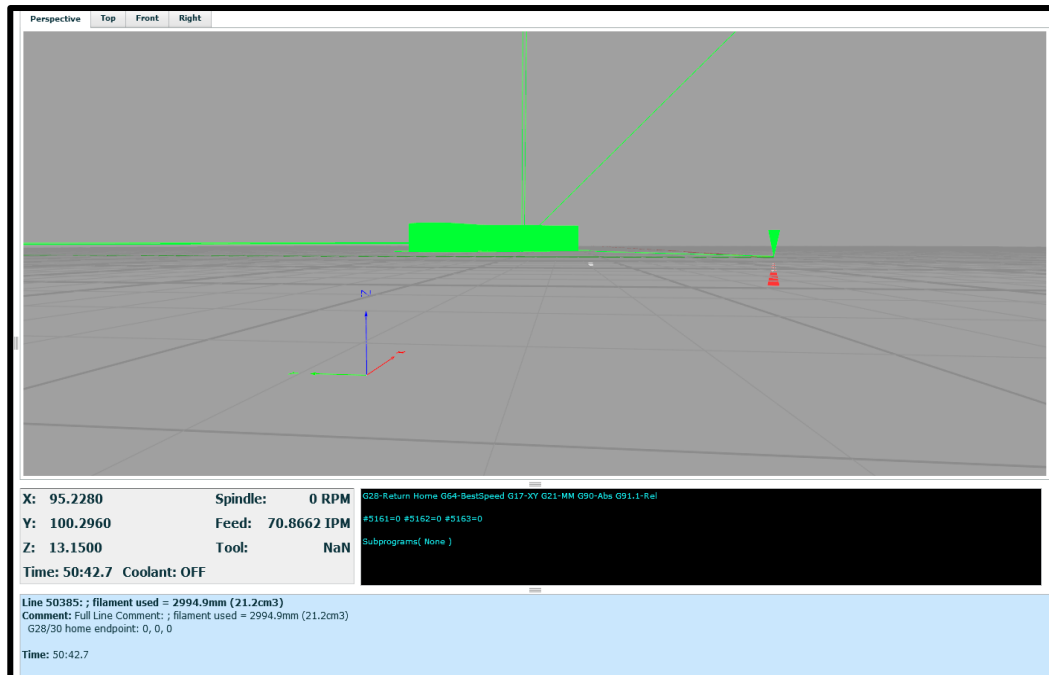
La simulación se puede realizar en tiempo real o aumentarla la velocidad de impresión hasta una velocidad 100 veces de lo normal, podemos notar que el tiempo registrado por el simulador es de 25 minutos, cabe recalcar que son tiempos estimados mas no son valores exactos.



**Figura 4.3c: Vista Superior, Componente Simulado**

Una vez terminada la simulación, G-Wizard nos estima un aproximado de cuanto material será necesario y el tiempo que tomara realizar la impresión, para este caso particular, el material requerido para la impresión de este componente será de 29.94 cm de filamento de plástico y se llevara a cabo en un tiempo estimado de 50 minutos y 42 segundos.

Para contrastar los porcentajes de impresión, mostraremos una vista lateral en perspectiva de la simulación del componente al 100%.

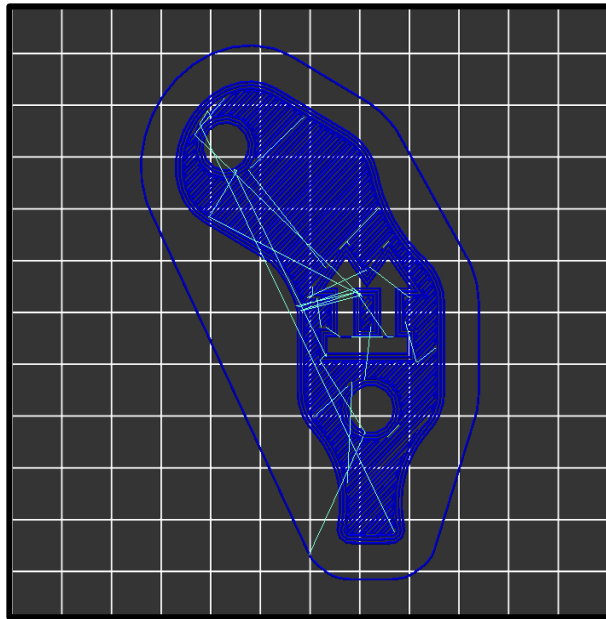


**Figura 4.3d: Perspectiva Lateral plano XY**

## 4.4 SIMULACION CON REPETIER HOST

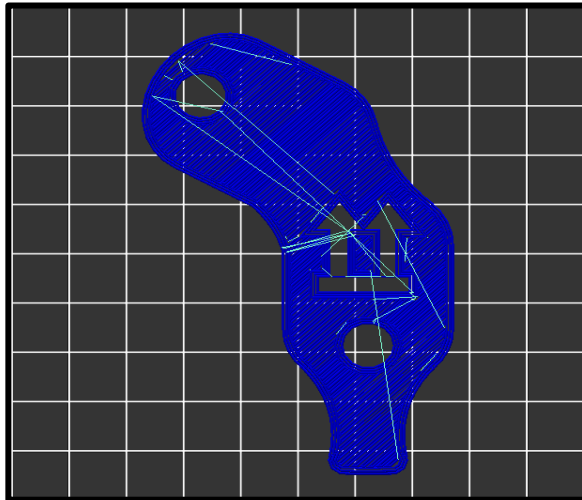
Con Repetier Host podemos tener una simulación menos precisa que la de G-Wizard pero mucho más práctica y conveniente para realizar un análisis respecto a cada una de las capas de nuestro objeto a imprimir. Con los parámetros establecidos anteriormente en el programa SLICER, nos arrojó

que nuestro objeto constara de 33 capas. A continuación analizaremos diferentes etapas y el comportamiento de cada una de ellas.



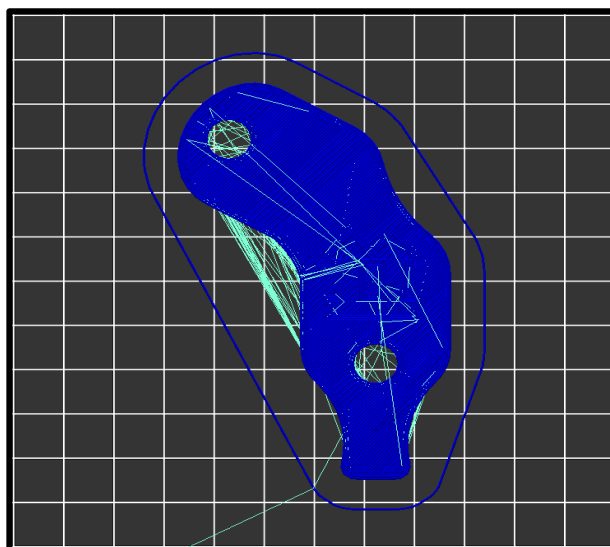
**Figura 4.4a: Vista Superior, Capa 1**

La primera capa mostrada en este simulador nos permite tener una mejor apreciación **Ver Figura 4.4a**, de cómo será la base del componente, nos damos cuenta que el relleno de las primeras capas es lineal, debido a que deben a que todas las caras exteriores del objeto deben ser macizas y sin espacios para que no existan imperfecciones.



**Figura 4.4b: Vista Superior, Capa 33**

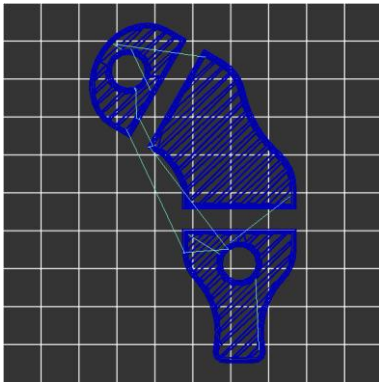
Como podemos notar en la **Figura 4.4b**, el relleno de las últimas capas es igual al de las capas iniciales, haciendo que el objeto se torne de macizo y compacto en su exterior.



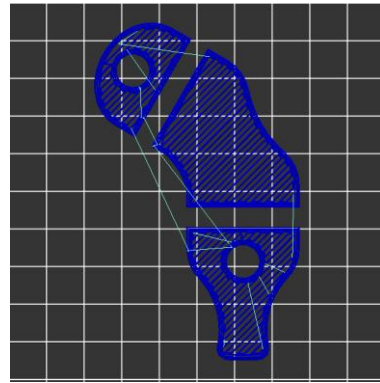
**Figura 4.4c: Vista Superior Modelo 3D**

#### 4.4.1 SECCIONAMIENTOS

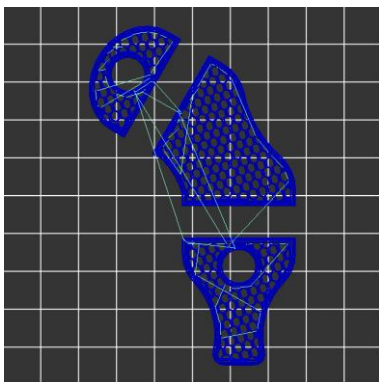
Existen varios tipos de seccionamientos que se pueden usar para rellenar el interior del componente. Realizamos la simulación con los diferentes seccionamientos que usa el software SLICER, haciendo un análisis con respecto a la capa número 15 obtuvimos lo siguiente:



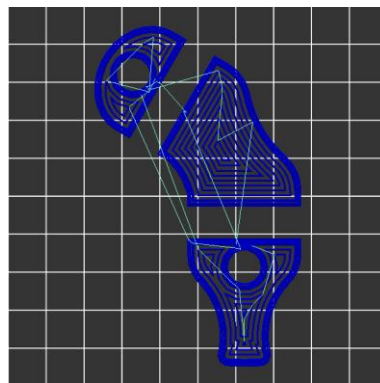
**Figura 4.4.1a Seccionamiento de líneas**



**Figura 4.4.1b Seccionamiento Rectilíneo**

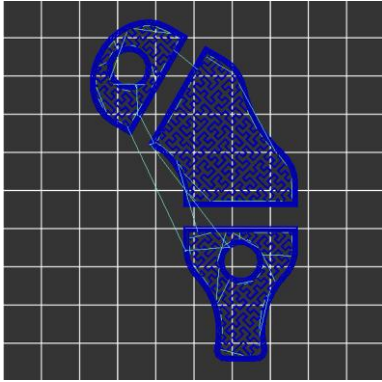


**Figura 4.4.1c Seccionamiento de Panal**

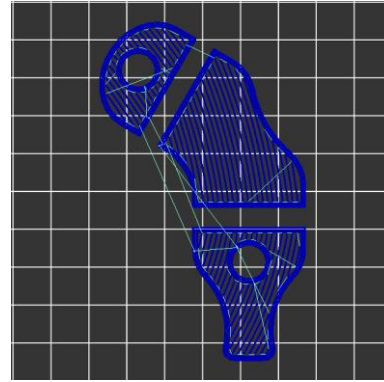


**Figura 4.4.1d Seccionamiento Concéntrico**





**Figura 4.4.1e Seccionamiento de  
Curva de Hilbert**



**Figura 4.4.1f Seccionamiento de  
Curva de Arquímedes**

Los seccionamientos mostrados en las **Figuras 4.4.1a a 4.4.1f**, quedan a elección de cada usuario, pero por lo general los más usados son el rectilíneo, el lineal y el de forma de panel.

Si queremos darle estilo a nuestra impresión, podemos usar otros estilos tales como:

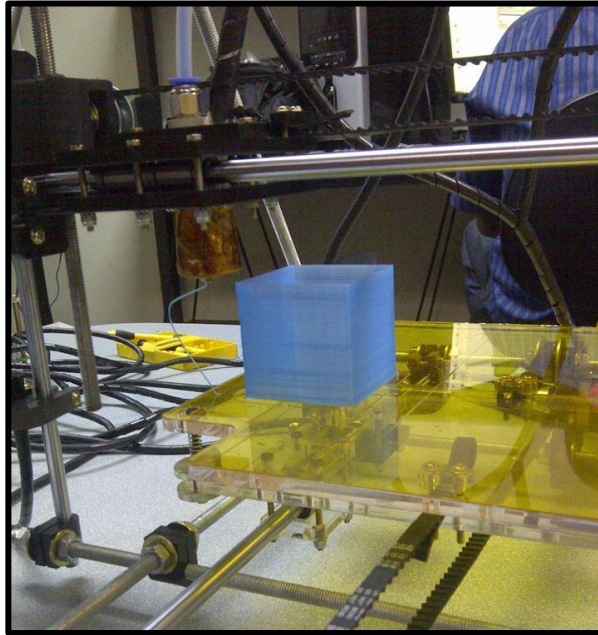
- Seccionamiento de Curva de Hilbert,
- Seccionamiento Concéntrico
- Seccionamiento de Curva de Arquímedes

## 4.5 RESULTADOS DE LA SIMULACIÓN

Por medio de cada una de las simulaciones que se han realizado en este capítulo, tenemos una amplia perspectiva sobre el funcionamiento del proyecto y los requerimientos necesarios para su implementación, en cada una de las etapas de simulación pudimos estimar valores óptimos de configuración, material de filamento indispensable y tiempo aproximado de impresión, todas estas variables debidamente registradas nos conducirán a tener un proceso de impresión impecable y cumplir con todos los objetivos planteados en nuestro proyecto.

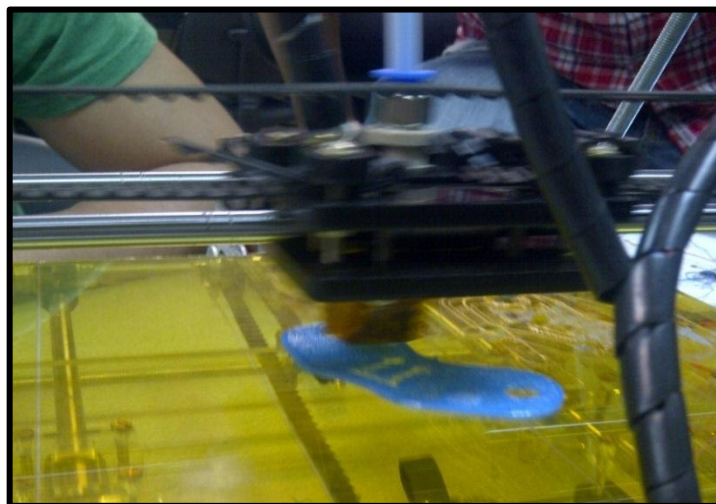
## 4.6 RESULTADOS DEL PROCESO DE LA IMPRESIÓN

Calibrada la impresora y con las configuraciones debidas, se realizó la primera impresión, usamos un modelo 3D que nos da por defecto el Slicer, el archivo STL usado fue el cube.STL, este es un cubo de interior hueco, se generó su código G luego comenzó el proceso de impresión, nos permitió ver claramente paso a paso los movimientos precisos que realiza el extrusor y al finalizar tuvimos un cubo con muy pocas imperfecciones. **Ver Figura 4.6a**



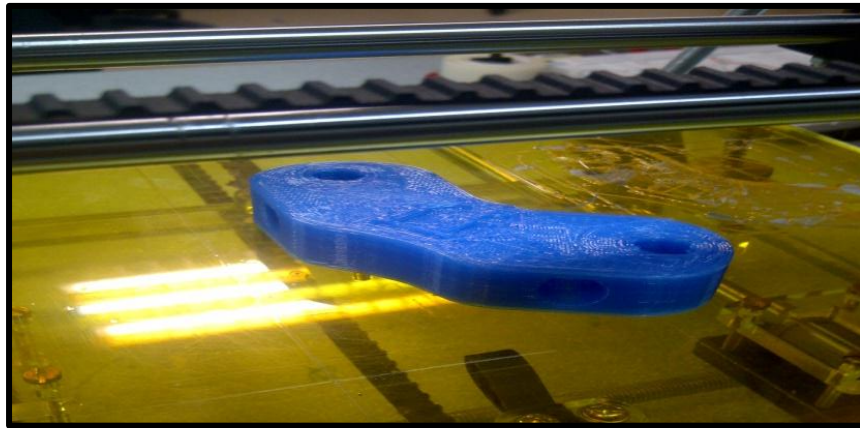
**Figura 4.6a: Impresión de Cubo**

En la **Figura 4.6b**, podemos visualizar la impresión de una de las partes de la impresora, es un vértice superior en sus primeras capas.



**Figura 4.6b: Impresión de Vértice, Capas Base**

En la **Figura 4.6c** se puede observar como el extrusor va formando las capas detalladamente y realiza un relleno de seccionamiento de panel.



**Figura 4.6c: Impresión de Vértice, Relleno Interior**

En la **Figura 4.6d** podemos observar la impresión finalizada, hay que tomar en cuenta que por el número de capas el tiempo en el que se realizó la impresión aproximadamente fue de una hora, esta es una pieza funcional de la estructura de la Prusa Mendel.



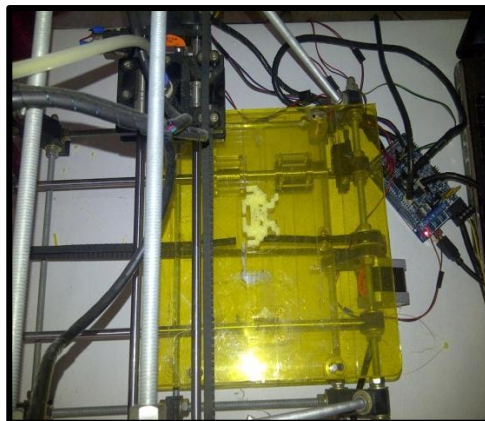
**Figura 4.6d: Impresión de Vértice, Objeto Completo**

A continuación se mostraran ejemplos del potencial que tienen estas nuevas impresoras, con un el respectivo modelado 3D de cada objeto se pudo obtener lo siguiente, el primero es un case de un Raspberry pi muy util para aislar sus circuitos de polvo y protegerlo de caidas. **Ver Figura 4.6e.**



**Figura 4.6e: Impresión Case Raspberry pi**

El segundo es un muñeco del clasico juego de video invader.



**Figura 4.6f: Impresión Invader**

## **CONCLUSIONES**

1. Con el diseño versátil de la Impresora 3D Prusa Mendel se puede abarcar una variada gama de posibilidades en cuanto a las diferentes figuras que podemos crear, esto ayudaría de forma radical en la arquitectura, ya que el modelado de una figura creada previamente en un software se lo puede obtener en cuestión de minutos, evitándose así la fabricación de maquetas a mano y optimizando el proceso de los diseños utilizados en la arquitectura.
2. La manera adecuada para controlar la parada de los motores de paso en la impresora Prusa Mendel, utilizado en el diseño de nuestro controlador es en base a finales de carrera mecánicos, realizando

previamente ajustes en el cableado de los motores de los tres ejes, estos finales de carrera remplazan a los ópticos utilizados con el controlador incluido por defecto en su fabricación.

3. El principal uso de el diseño del controlador de la impresora 3D para este proyecto esta dirigido a recrear el proyecto RepRap, este se lleva acabo con la autoreplicación de las partes de una impresora 3D, para asi poder reproducir una impresora funcional, esto incluyen netamente las partes plasticas solo de su forma estructural.
4. Para controlar la corriente de paso a los motores, fue imprescindible el uso de cuatro Pololu`s, estos son drivers que se los configura manualmente y funcionan como una resistencia variable que nos permite restringir o dar paso la corriente necesaria para que los motores de paso tengan un correcto funcionamiento. El valor óptimo de corriente obtenido luego de varias mediciones fue 200 mA y nos asegura su correcta operación y desempeño.
5. El uso de una nueva tecnologia poco conocida en el Ecuador, el impacto que tendra luego de unos años en el paises de todo el mundo, la facilidad de adquirir una y poder replicarla, nos muestra la amplia gama de aplicaciones en que se puede emplear el uso de la

impresora3D, hacen que el proyecto RepRap sea llamativo para la gente emprendedora que quiera generar un nuevo mercado en nuestro medio.



## RECOMENDACIONES

1. Para la correcta programación de la tarjeta Arduino es recomendable cargar el firmware sprinter , ya que con este se logran obtener mayor desempeño en el diseño del controlador, se recomienda también revisar las variables en el código de programación específicamente las variables de las paradas de los motores ya que un error en estas líneas de código conllevan a que los motores giren en un solo sentido.
2. Al momento del ensamblaje de la estructura de la impresora 3D, se deben calcular de una manera precisa la posición de sus partes ya que sin esto se puede debilitar la estructura con el constante

movimiento de sus motores, además se recomienda la correcta nivelación de el area de impresión, esto se lo debe realizar con respecto a la estructura donde se encuentre instalada la impresora.

3. Si se desea reducir el tiempo de duración de una impresión se debe manipular el número de capas con la que esta consta, es decir mediante software uno puede modificar esta característica, la calidad de la imagen se reducirá de acuerdo al número de capas que sean eliminadas.

## BIBLIOGRAFÍA

[1] AdrianBowyer, Proyecto RepRap – Especificaciones

<http://www.RepRap.org/wiki/about>

Fecha: 15 de Marzo del 2013 .

[2] AdrianBowyer , Conexiones tarjeta Ramps\_1.4 – especificación de conexiones Marzo 2013

[www.reprarp.org/wiki/Ramps\\_1.4](http://www.reprarp.org/wiki/Ramps_1.4)

Fecha: 15 de Marzo del 2013.

[3] McRoberts, Michael. Beginning Arduino, editorial Apress, New York, 2010.

[www.springeronline.com](http://www.springeronline.com)

Fecha: 15 de Marzo del 2013.

[4] Evants, Brian. Practical 3D printers the Science and art of 3d printing, editorial Apress, New York, 2012.

[www.springeronline.com](http://www.springeronline.com)

Fecha: 9 de Abril del 2013.

[5] Floyd Kelly, James. Hood-Daniel, Patric. Printing in plastic, Build your own 3D printer, editorial Apress, New York, 2011.

[www.springeronline.com](http://www.springeronline.com)

Fecha: 11 de Abril del 2013.

[6] Timis, Harold. Practical Arduino engineering, editorial Apress, New York, 2011.

[www.springeronline.com](http://www.springeronline.com)

Fecha: 11 de Abril del 2013.

[7] DomainSponsor, Driver Pololu. Especificaciones técnicas.

[www.polulo.com/catalog/product/1182](http://www.polulo.com/catalog/product/1182)

Fecha: 5 de Mayo del 2013.

[8] NextDayReprap, Next Day RepRap. Conceptos Generales.

[www.nextdayreprap.co.uk/1-0-bill-of-materials-prusa-mendel-build-manual](http://www.nextdayreprap.co.uk/1-0-bill-of-materials-prusa-mendel-build-manual)

Fecha: 5 de Mayo del 2013.

[9] SOLIDOODLE, Software de Diseño 3D.

[www.solidoodle.com/how-to-2/how-to-install-software/](http://www.solidoodle.com/how-to-2/how-to-install-software/)

[10] Zentoolworks, Arduino Stepper Motor. Especificaciones para controlar motores de paso en base Arduino

[wiki.zentoolworks.com/index.php/Arduino\\_Stepper\\_Motor\\_Control](http://wiki.zentoolworks.com/index.php/Arduino_Stepper_Motor_Control)

Fecha: 11 de Mayo del 2013.

[11] Impresora 3D, Introducción, concepto Impresora 3D.

[www.impresora3D.com](http://www.impresora3D.com)

Fecha: 15 de Marzo del 2013.