

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL



FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN

“DISEÑO DE RUTINAS UTILIZADAS EN EL MANEJO DE INTERFACES
USANDO UN SISTEMA ARDUINO PARA LA IMPRESIÓN EN 3
DIMENSIONES CON EJEMPLOS PRÁCTICOS EN PROTOTIPO DE
IMPRESORA 3D.”

TESINA DE SEMINARIO

Previa la obtención del Título de

INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES

Presentado por:

Jorge Luis Chong Freire

Geovanni Francisco Padilla Mora

GUAYAQUIL – ECUADOR

AÑO 2013

AGRADECIMIENTO

Quiero agradecer a Dios por bendecirme con la familia que tengo, por la confianza que el depositó en mí y por darme la fuerza para no rendirme.

A mi familia que siempre creyó en mí, sobre todo a mis padres que nunca dudaron de mí y siempre, siempre están conmigo dándome aliento sin importar lo que haga

A mis amigos por darme su apoyo y demostrarme que siempre podre confiar en ellos.

Al Ing. Carlos Valdivieso por haber compartido sus conocimientos con nosotros.

Geovanni Francisco Padilla Mora.

AGRADECIMIENTO

Primero que todo quisiera agradecer a Dios por haberme ayudado a mantener siempre fija mis metas profesionales y ayudarme con la perseverancia para poder vencer todos los obstáculos que se me han presentado.

Quisiera también agradecer a toda mi familia y seres queridos por siempre haber estado ahí cuando los he necesitado y por darme apoyo incondicional en toda decisión que haya tomado.

Al Ing. Carlos Valdivieso por habernos guiado en esta nuestra última etapa de vida universitaria y por habernos dado siempre su confianza.

Jorge Luis Chong Freire.

DEDICATORIA

A mi padre, por enseñarme a no darme por vencido ni aun vencido, por enseñarme que siempre, haga lo que haga, lo haga porque me gusta y que siempre trate de ser el mejor. A mí querida madre por brindarme su amor incondicional y su apoyo, por ser siempre mi soporte, a ellos dos por siempre confiar en mí.

Geovanni Francisco Padilla Mora.

A mi familia, amigos y todos los seres queridos que han estado presentes durante estos años y me han ayudado a crecer como persona y como profesional.

Jorge Luis Chong Freire

TRIBUNAL DE SUSTENTACIÓN

A handwritten signature in black ink, appearing to read 'Carlos Valdivieso A.', written over a horizontal line.

Ing. Carlos Valdivieso A.

PROFESOR DEL SEMINARIO DE GRADUACIÓN

A handwritten signature in black ink, appearing to read 'Hugo Villavicencio V.', written over a horizontal line.

Ing. Hugo Villavicencio V.

PROFESORA DELEGADA POR LA UNIDAD ACADÉMICA

INDICE GENERAL

DECLARACIÓN EXPRESA

INDICE GENERAL

INDICE DE FIGURAS

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

INTRODUCCIÓN...
Superior Politécnica del Litoral".

CAPÍTULO 1

DESCRIPCIÓN GENERAL DEL PROGRAMA

(Reglamento de Graduación de la ESPOL)

11 OBJETIVOS

12 PLAN DE ESTUDIOS

12.1 PLAN DE ESTUDIOS

12.2 PLAN DE ESTUDIOS

13 PLAN DE ESTUDIOS

14 PLAN DE ESTUDIOS

15 PLAN DE ESTUDIOS

16 PLAN DE ESTUDIOS

17 PLAN DE ESTUDIOS



Jorge Luis Chong Freire



Geovanni Francisco Padilla Mora

ÍNDICE GENERAL

ÍNDICE GENERAL.....	V
ÍNDICE DE FIGURAS	VIII
ÍNDICE DE TABLAS.....	XII
INTRODUCCIÓN	XIII
CAPÍTULO 1.....	1
DESCRIPCIÓN GENERAL DEL PROYECTO	1
1.1 Motivación	1
1.2 Historia	5
1.2.1 CNC.....	5
1.2.2 Código G	6
1.2.3 Arduino	6
1.4 Justificación de la Investigación	11
1.5 Objetivos.....	13
1.5.1 Objetivos Generales	13
1.5.2 Objetivos Específicos	14
CAPÍTULO 2.....	2
FUNDAMENTO TEÓRICO.....	2
2.1 Arduino	2
2.1.1 Descripción General.....	2

2.1.2 Partes del Arduino	16
2.1.3 Especificaciones.....	17
2.1.4 Tamaños.....	18
2.1.5 Ventajas.....	19
2.2 STL.....	21
2.2.1 Introducción	21
2.2.2 Especificaciones.....	22
2.3 Impresora 3D.....	23
2.4 Código G	24
2.4.1 Introducción	24
2.4.2 Rutinas	24
2.4.3 Órdenes básicas código G	25
CAPÍTULO 3.....	16
IMPLEMENTACIÓN DEL PROYECTO.....	16
3.1 Introducción	16
3.2 Diseño en STL.....	33
3.3 Arduino Software.....	35
3.4 Repetier Host.....	38
3.5 Slic3r	42
3.6 ProcessingGCodeViewer.....	44
3.7 Creación de las rutinas del código G	45
3.8 Creación del código G del Invader	48
3.9 Creación del código G del Engranaje	49
CAPÍTULO 4.....	53
RESULTADOS Y ANÁLISIS.....	53
4.1 Introducción	53
4.2 Análisis usando ProcessingGCodeViewer.....	53

4.2.1 Rutinas del Invader.....	53
4.2.2 Rutinas del Engranaje	59
4.2.3 Comparación de rutinas a partir del ProcessingGCodeViewer.....	63
4.3 Modelos Impresos	64
4.3.1 Introducción	64
4.3.2 Modelo Impreso Invader	65
4.3.3 Modelo Impreso Engranaje.....	69
CONCLUSIONES	74
RECOMENDACIONES	78
ANEXO	80
BIBLIOGRAFIA	83

ÍNDICE DE FIGURAS

Figura 1.1 Impresoras 3D.....	2
Figura 1.2 Impresión 3D hecha con una impresora FORM1.....	2
Figura 1.3 Impresión usando chocolate.....	3
Figura 1.4 Placa Arduino	6
Figura 1.5 Impresora 3D Mendel Prusa.....	7
Figura 1.6 Impresión de un busto en una impresora 3D moderna.....	9
Figura 1.7 Impresora 3D utilizada para este proyecto	12
Figura 2.1 Partes del Arduino Uno por colores.....	16
Figura 2.2 Ejemplo en infografía.....	22
Figura 3.1 Modelo en CAD con el programa BLENDER	34
Figura 3.2 Modelo en CAD con el programa LIGHTWAVE	35
Figura 3.3 Pantalla principal Arduino Software.....	36
Figura 3.4 Programa Blink ingresado en el software Arduino.....	37
Figura 3.5 Programa Sprinter cargado en el software Arduino.....	38

Figura 3.6 Arduino programado y conectado a la impresora y computador ..	38
Figura 3.7 Pantalla principal de Repetier Host	39
Figura 3.8 Opciones de modificación del STL	40
Figura 3.9 Opciones de conversión a código G	40
Figura 3.10 Opciones de conversión a código G	41
Figura 3.11 Controles de la impresora	41
Figura 3.12 Pantalla principal del Slic3r	42
Figura 3.13 Modelo de un carro cargado en el Slic3r	43
Figura 3.14 Opciones de impresión de Slic3r	44
Figura 3.15 ProcessingGCodeViewer analizando un STL de un carro.....	45
Figura 3.16 Configuración #1	47
Figura 3.17 Configuración #2	48
Figura 3.18 Modelo Invader.....	48
Figura 3.19 Invader cargado en el Slic3r.....	49
Figura 3.20 Modelo original del Engranaje	49
Figura 3.21 Modelo del Engranaje después de la modificación	50

Figura 4.1 Impresora 3D conectada al suministro de plástico	52
Figura 4.2 Invader Configuración #1	54
Figura 4.3 Capa 1 del Invader Configuración #1	55
Figura 4.4 Capa 3 del Invader Configuración #1	55
Figura 4.5 Capa 7 del Invader Configuración #1	56
Figura 4.6 Capa 11 de Invader Configuración #1	56
Figura 4.7 Invader Configuración #2	57
Figura 4.8 Capa 1 del Invader Configuración #2	57
Figura 4.9 Capa 7 del Invader Configuración #2	58
Figura 4.10 Capa 26 del Invader Configuración #2	58
Figura 4.11 Capa 1 del Engranaje.....	59
Figura 4.12 Capa de relleno del Engranaje Configuración #1	60
Figura 4.13 Capa 10 del Engranaje Configuración #1	60
Figura 4.14 Capa 1 del Engranaje Configuración #2.....	61
Figura 4.15 Capa 7 del Engranaje Configuración #2.....	62

Figura 4.16 Capa 16 del Engranaje Configuración #2	62
Figura 4.17 Comparación Invader Configuración#1 y #2.....	64
Figura 4.18 Imprimiendo el modelo Invader.....	65
Figura 4.19 Impresión completa del Invader.....	66
Figura 4.20 Modelo Impreso Invader con la Configuración #1.....	66
Figura 4.21 Modelo Impreso Invader con la Configuración #2.....	67
Figura 4.22 Parte posterior del Invader Configuración #1	68
Figura 4.23 Parte posterior del Invader Configuración #2	68
Figura 4.24 Modelo Impreso Engranaje con la Configuración #1	69
Figura 4.25 Modelo Impreso Engranaje con la Configuración #2	70
Figura 4.26 Parte posterior del Engranaje con la Configuración #1	71
Figura 4.27 Parte posterior del Engranaje con la Configuración #2.....	71
Figura 4.28 Vista lateral del Engranaje con la Configuración #1	72
Figura 4.29 Vista lateral del Engranaje con la Configuración #2	72
Figura 4.30 Carcasa protectora para un Raspberry Pi.....	76

ÍNDICE DE TABLAS

Tabla 2.1 Partes del Arduino	17
Tabla 2.2 Especificaciones físicas de los Microcontroladores Atmega168, Atmega328 y el Atmega1280	18
Tabla 3.1 Especificaciones de las Configuraciones.....	47
Tabla 4.1 Resultados del Invader	69
Tabla 4.2 Resultados del engranaje	73

INTRODUCCIÓN

El presente trabajo sobre impresoras 3D está enfocado en las rutinas utilizadas a la hora de imprimir, valiéndonos de la ayuda del Sistema Arduino.

Aunque en los primeros capítulos abarcamos tanto la historia como los marcos teóricos de los elementos que vamos a utilizar, tanto en hardware como en software, en los capítulos 3 y 4 centramos nuestros esfuerzos en explicar la importancia que tienen las rutinas de impresión y recalcar que la misma rutina no tiene por qué ser la mejor para figuras distintas.

De la rutina que se vaya a utilizar dependerá mucho el resultado final, tanto en tiempo como en inversión de material, además de la calidad del producto final, ya que como se verá en el desarrollo de este trabajo, no por usar más material el resultado final será mejor.

Para mostrar la importancia de las rutinas y su diferencia en varios trabajos se decidió trabajar con 2 figuras distintas tanto en tamaño como en forma, además se usaron 2 rutinas diferentes en las mismas figuras y con estas compararemos los resultados finales.

CAPÍTULO 1

DESCRIPCIÓN GENERAL DEL PROYECTO

1.1 Motivación

Nuestro proyecto sobre impresoras 3D, fue motivado en gran parte a la masificación de este producto. La tecnología 3D, por lo menos a nivel industrial, existe hace más de 30 años. En sus comienzos era utilizada en la construcción de maquetas y repuestos plásticos.

Ahora que los precios de estos equipos han bajado considerablemente y existe una comercialización masiva, nos interesó mucho este tema, sobre

todo el funcionamiento de cada una de sus partes. Ya que esta tecnología es nueva existe mucha información que surge cada día por parte de desarrolladores y comunidades online. Esta innovación tanto en hardware como en software puede llegar a ser confusa para el usuario. Es por esto que estudiaremos estas impresoras y crearemos un análisis completo y fácil de entender para cualquier persona.

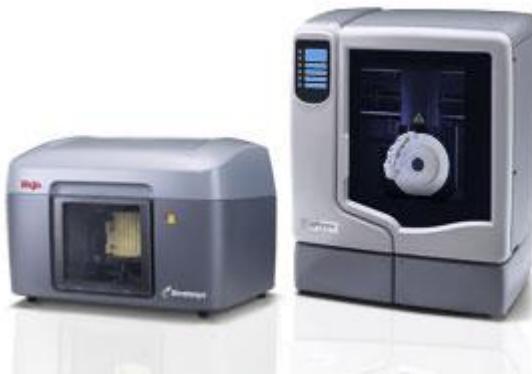


Figura 1.1 Impresoras 3D [1]

Al momento existen 3 formas de realizar impresiones 3D; estereolitográfica, Form1 y por tecnología FDM, por costos nosotros utilizaremos tecnología FDM.

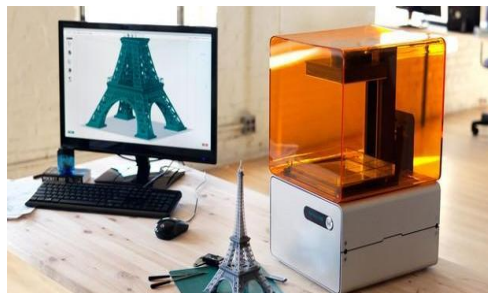


Figura 1.2 Impresión 3D hecha con una impresora FORM1 [2]

La tecnología FDM es algo menos precisa que las otras pero su ventaja se encuentra en su factor económico. Lo interesante de esta tecnología es que utiliza un hilo de plástico que puede ser ABS y PLA (utilizado en menor medida). Estos plásticos son usados comúnmente pero si se necesita utilizar la impresora con otro propósito, este plástico puede ser cambiado por cualquier otro material dejando así una gama casi infinita de posibilidades para el uso de la impresora 3D.

En la industria pastelera ya existen impresoras que hacen modelos muy complejos de chocolate. En el campo de la medicina la universidad Weill Cornell Medical College y la Universidad de Bioingeniería de Cornell lograron hacer una impresión 3D de una oreja humana utilizando geles inyectables compuestos de células vivas.



Figura 1.3 Impresión usando chocolate [3]

El éxito que han alcanzado las impresoras 3D en la actualidad se debe en gran medida a la filosofía "*hágalo usted mismo*", que permite al usuario crear objetos desde cero o fabricar sus propios repuestos para cualquier tipo de equipo que se encuentre dañado en casa o en una fábrica.

Además de las facilidades de la impresión 3D por sí sola, las posibilidades aumentan al acoplarle un Arduino. A las impresoras se le puede conectar esta placa y con esta controlar la impresión. Con el Sistema Arduino se pueden controlar los motores de las impresoras fácilmente, lo que nos ayudaría a controlar sus velocidades como sus posiciones.

El control de los motores ayuda a poder manejar la impresión a nuestro antojo, de poder moverla de derecha a izquierda o viceversa, La velocidad que creamos conveniente y la cantidad de material que tiene que usar la impresora en cada punto.

Este control se logra mediante las rutinas que utilizemos en las interfaces mediante el sistema Arduino, esto nos motiva a escoger el tema, el poder tener control total de la impresión mediante rutinas, que estas puedan ser cambiadas y poder obtener el mismo o tal vez un mejor resultado.

1.2 Historia

Para entender cómo funcionan las diferentes partes de nuestro proyecto es necesario entender como cada una de estas partes fueron creadas. En esta sección hablaremos primero de las máquinas CNC, después describiremos los comienzos del código G y por último haremos un análisis del hardware, el Arduino y la impresora 3D.

1.2.1 CNC

En 1952 la industria aerodinámica Norte-americana se encontró con la necesidad de realizar diferentes configuraciones para las hélices de helicópteros. Fue de esta necesidad de donde empieza a nacer el uso de equipos CNC. Computer Numerical Control se refiere al control y automatización de maquinarias mediante comandos programados en un medio de almacenamiento.[4]

Aunque desde sus inicios esta tecnología demostró que podía reducir costos no fue hasta 1972 que esta fue adoptada por la mayoría de las industrias. Este boom ocurre debido a la aparición de la minicomputadora en el mercado. Este le presenta a las compañías una manera más económica de programar las órdenes para la maquinaria.

1.2.2 Código G

En sus comienzos la adopción de CNC tenía muchos problemas, entre ellos la falta de un estándar. Cada fabricante tenía su propio lenguaje de programación. Esto fue resuelto en 1958 cuando nace el código G. El MIT (Massachusetts Institute of Technology) desarrollo su propio código para uso en el laboratorio de servomecanismos. Este fue adoptado por la alianza de la industria electrónica en 1960. Es en este punto en que diferentes compañías grandes empiezan a integrar esta tecnología y empieza su desarrollo.

1.2.3 Arduino

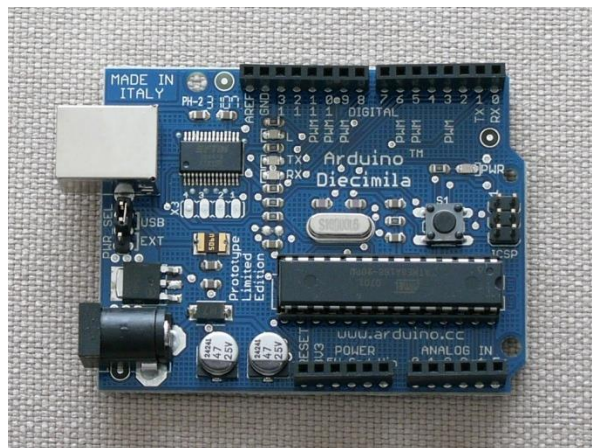


Figura 1.4 Placa Arduino [5]

En el 2005 en Italia se inició el proyecto Arduino of Iyrea, liderado por Massimo Banzi y David Cuartielles. El objetivo de este proyecto era crear un controlador para poder manejar pequeños dispositivos creados por estudiantes. Su gran gama de usos ha hecho de este proyecto Arduino una gran herramienta para el aprendizaje. Hasta ahora se han vendido más de 250 mil placas en todo el mundo y su tecnología ha sido incorporada en muchas universidades como la Standford y MIT. [6]

Con este Arduino se pueden manejar cosas muy simples como la iluminación de un LED hasta equipos más complejos como en nuestro caso una impresora 3D. Estas necesitan un controlador que convierta el código G en órdenes mecánicas y el Arduino puede cumplir esta necesidad fácilmente

1.2.4 Impresoras 3D

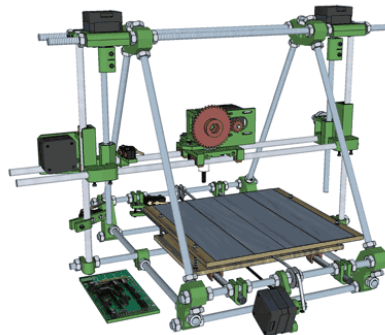


Figura 1.5 Impresora 3D Mendel Prusa [7]

Las impresoras 3D comenzaron a desarrollarse hace 30 años. Los primeros prototipos fueron creados por arquitectos para poder crear rápidamente pequeñas piezas para modelos y maquetas. Estas piezas eran de plástico o metal. En el 2006 el científico Adrian Bowyer de la universidad de Bath empezó un proyecto en el cual se creó la impresora RepRap (Replicating Rapid Prototyper).

Una de las cualidades más importantes de esta impresora es su código abierto. Esto hizo que esta tecnología empiece a ser más accesible para las personas. En el 2012 el francés Emmanuel Gilloz construye la FoldaRap, la primera impresora 3D de dimensiones más pequeñas la cual permite al usuario transportarla con facilidad.

Para crear una pieza se necesita un modelo en 3D esto se consigue con un archivo CAD en formato STL o VRML. Las impresoras 3D utilizan diferentes tipos de tecnología para crear modelos de diferentes materiales que pueden cubrir muchas necesidades. En los últimos años esta tecnología ha empezado a ser atractiva para una gran variedad de clientes.

1.3 Estado actual de la tecnología

En la actualidad esta tecnología está teniendo una gran acogida por parte de muchos desarrolladores. Mientras se mejoran los procesos y se aumenta la agilidad para pasar de un modelo en la computadora a un modelo impreso se están encontrando nuevos usos para estas impresoras

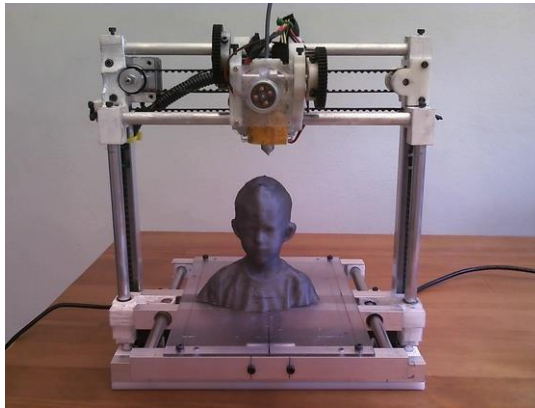


Figura 1.6 Impresión de un busto en una impresora 3D moderna [8]

En un comienzo las impresoras 3D eran usadas solo para maquetas y modelos para arquitectos pero ahora los usos son muy variados. Los usos actuales más comunes son la creación de esculturas, joyería, piezas plásticas o metálicas para maquinaria. En los últimos años también se están empezando a crear variaciones a la impresora, como por ejemplo se están cambiando elementos de la impresora y se están introduciendo partes

nuevas que sirven para utilizar la impresora de nuevas formas, como diseñar decoraciones de pasteles o hasta para elaborar prendas de vestir.

En este momento estos equipos están recibiendo atención de muchas industrias por lo cual están apareciendo diferentes marcas y modelos de impresoras. El problema que surge cuando una tecnología se encuentra en este estado es que existen muchas variaciones y no hay un enfoque general. Así como el código G esta tecnología tiene que llegar a un estándar para que su venta sea más atractiva para el consumo masivo.

La falta de estándar también tiene sus ventajas. Existe una gran variedad de programas para generar código G, cada uno con sus ventajas y desventajas. Para cada tipo de uso que se le va a dar a la impresora hay un software indicado.

Si se piensa darle un uso básico a la impresora y se va a usar como método de aprendizaje hay programas que nos ayudan a familiarizarnos con el proceso y si el uso de la impresora va a ser para crear piezas que necesitan un nivel de precisión muy alto también existe una interfaz que nos brinda las herramientas necesarias.

Un paso muy importante para llegar a un estándar en la impresión 3D es llegar a entender los diferentes usos que se le van a dar a estas impresoras y entender cuáles son las herramientas necesarias para los usuarios. Una vez que se lleguen a agrupar las herramientas más comunes también es importante crear una interfaz que sea amigable para el usuario.

Todos los programas que nos ayudan a imprimir en 3D primero tienen que crear el código G, las rutinas que conforman este código son muy importantes ya que estas influyen en el acabado de la figura. El enfoque principal de nuestra tesis es analizar las interfaces que nos ayudan a realizar impresiones en 3D y como las rutinas afectan los modelos impresos.

1.4 Justificación de la Investigación

Las impresoras 3D tienen muchas aplicaciones que no solo pueden ser aprovechadas por industrias grandes sino también por cualquier tipo de usuario. Para poder llegar a esto es necesario que se establezcan ciertos estándares y el software llegue a ser simple y a la vez tenga todas las herramientas necesarias para que todo usuario pueda realizar lo que tenga planeado.

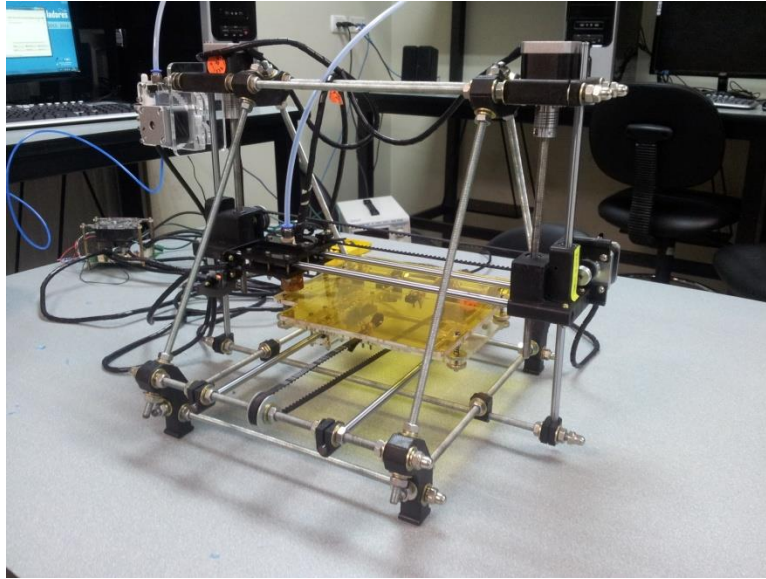


Figura 1.7 Impresora 3D utilizada para este proyecto

Es nuestra meta poder estudiar y manejar como las interfaces crean diferentes rutinas de código G. Con esta información podemos entender cuáles son las ventajas y desventajas de cada característica de una rutina. Analizando el estado de estas impresoras también podemos mirar al futuro hacer predicciones de cómo estas pueden llegar a tener un aun mayor uso en la vida de todo tipo de usuario.

1.5 Objetivos

1.5.1 Objetivos Generales

- Estudiar las rutinas creadas por las interfaces para la creación de las impresiones 3D.
- Familiarizarnos con el uso de programas para la manipulación y creación de figuras en tercera dimensión con extensión STL.
- Familiarizarnos con el uso y manipulación de programas que produzcan códigos G, conociendo sus ventajas y desventajas.
- Conocer los diferentes tipos de Arduino existentes en el mercado y familiarizarnos con el que vamos a utilizar.
- Conocer los diferentes tipos de impresión 3D, y entender sus ventajas y desventajas.

1.5.2 Objetivos Específicos

- Familiarizarnos con el uso del Sistema Arduino, incluyendo sus componentes y capacidades.
- Revisar y entender varias rutinas para el control de los motores de nuestra impresora 3D a través del Sistema Arduino.
- Realizar impresiones finales en tercera dimensión con su respectivo modelado en un programa que nos dé un archivo STL y sea transformado a código G.

CAPÍTULO 2

FUNDAMENTO TEÓRICO

2.1 Arduino

En esta sección hablaremos de cómo este dispositivo nos ayuda a controlar la impresora. Este equipo puede tener muchos usos por lo cual describiremos un poco sus componentes para entenderlo mejor.

2.1.1 Descripción General

El Arduino es una placa basada en hardware libre, creada para realizar proyectos de electrónica de una manera más sencilla, y conectar así el mundo físico con el digital.

Posee un microcontrolador Atmel AVR y algunos puertos de entrada y salida que pueden ser analógicos como digitales, ofreciendo una mayor flexibilidad a la hora de realizar proyectos de varios tipos.

2.1.2 Partes del Arduino

El Arduino posee un microcontrolador Atmel [9] AVR y los más usados son

- Atmega168
- Atmega328
- Atmega1280
- Atmega8

Aunque en el 2012 [10] ya salió un Arduino con un microcontrolador CortexM3.

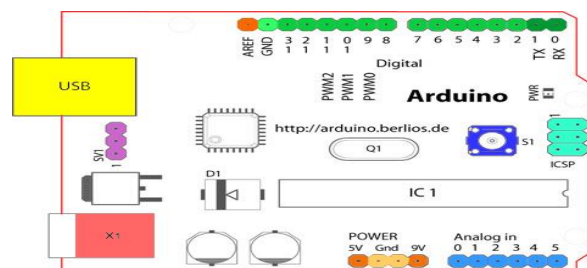


Figura 2.1 Partes del Arduino Uno por colores [11]

Estas son las partes del ARDUINO UNO [11], empezando desde la parte superior hacia la derecha se distribuyen de la siguiente manera:

PARTE FÍSICA	COLOR
Terminal de referencia analógica	naranja
Tierra digital	verde claro
Terminales digitales 2 --13	verde
Terminales digitales 0 --1 (e/s serie TX/RX)	verde oscuro
Reset	azul
ICSP	turquesa
Terminales analógicos 0 5	azul claro
Terminal de alimentación	naranja
Terminal de tierra	naranja claro
Entrada alimentación externa 9 – 12 vdc	rosa
Selector de alimentación externa o por usb	purpura
USB	amarillo

Tabla 2.1 Partes del Arduino

2.1.3 Especificaciones

Algunas de las especificaciones más importantes de los microcontroladores Atmega168, Atmega328 y el Atmega1280 que son usados por los Arduinos DIECIMILA, DUEMILANOVE Y MEGA [10].

	Atmega168	Atmega328	Atmega1280
Voltaje operativo	5 V	5 V	5 V
Voltaje de entrada recomendado	7 - 12 V	7 - 12 V	7 - 12 V
Voltaje de entrada límite	6 - 20 V	6 - 20 V	6 - 20 V
Pines de entrada y salida digital	14 (6 PWM)	14 (6 PWM)	54 (14 PWM)
Pines de entrada analógica	6	6	16
Intensidad de corriente	40 mA	40 mA	40 mA
Memoria Flash	16KB (2KB reservados para el bootloader)	32KB (2KB reservados para el bootloader)	128KB (4KB reservados para el bootloader)
SRAM	1 KB	2 KB	8 KB
EEPROM	512 bytes	1 KB	4 KB
Frecuencia de reloj	16 MHz	16 MHz	16 MHz

Tabla 2.2 Especificaciones físicas de los Microcontroladores Atmega168, Atmega328 y el Atmega1280 [10]

2.1.4 Tamaños

De mayor a menor podemos colocarlos de la siguiente forma [12]:

- Arduino Mega
- Los Arduinos Diecimila, Duemilanova y Arduino Bluetooth
- Arduino Pro

- Arduino Nano
- Arduino Mini

2.1.5 Ventajas

- Arduino Mega

Es el más potente de todos, posee mayor número de pines de e/s. Es preferible su uso para trabajos grandes y de mayor complejidad. Utiliza el microcontrolador Atmega1280, que posee mayor memoria RAM.

- Arduino Bluetooth

Este Arduino viene con un módulo de transmisión de datos, que puede alcanzar hasta 100 metros. La facilidad que posee este Arduino es la de poder programarlo sin cables y poder hacer comunicación serie con varios dispositivos bluetooth.

- Arduino Pro

El Arduino Pro es un poco más robusto y posee un mejor acabado final. Además incorpora ciertas funcionalidades especiales tales como un conector especial para conectar una batería LiPo y realizar un montaje portátil.

- Arduino Nano

Este Arduino posee una gran ventaja para los estudiantes o personas que estén experimentando con proyectos de electrónica y es que puede ser colocado directamente en el protoboard.

- Arduino Duemilanova

Esta es una placa estándar.

- Arduino Diecimila

Es una versión anterior al Duemilanova con un microcontrolador diferente pero muy compatibles el uno con el otro tanto en programación como patillaje.

- Arduino Mini

Es una versión más pequeña de la placa Arduino normal. Su tamaño es de 30x18mm. Posee las mismas funcionalidades del Arduino Duemilanova. [12]

2.2 STL

2.2.1 Introducción

Es un formato estándar de archivo de prototipado rápido [13]. Archivos como estos utilizan una malla de triángulos pequeños para definir la forma de un objeto. Entre más pequeños sean los triángulos mejor se verá la superficie del objeto, claro que esto conlleva a gastar mayor tiempo y será un archivo más pesado.

Siempre los triángulos que conforman la malla deben encajar perfectamente sin haber huecos ni superposiciones. Estos sistemas con prototipado rápido aparecen en 1987 junto al proceso de estereolitografía de sistemas 3D. Por tanto, los archivos STL provienen de los formatos que leían las primeras máquinas de este tipo.

Esta extensión STL es un estándar para la mayor parte de sistemas de prototipado rápido, aunque en pocas ocasiones los programas de modelado 3D trabajan sobre este tipo de archivos directamente.

Como son raros los programas que trabajan sobre archivos STL, se debe hacer una exportación hacia este formato antes de realizar la impresión 3D,

incluso es muy común que este archivo sea una conversión intermedia de la mayoría de programas CAD.

Ejemplo en infografía:



Figura 2.2 Ejemplo en infografía [14]

Los archivos STL se crean a partir de 2 bases de datos [14]:

- Nube de puntos
- Modelado CAD

2.2.2 Especificaciones

El formato STL puede ser:

- Binario
- ASCII

Ambos contienen la misma información pero la diferencia radica en que uno puede ser modificado por un editor de texto y el otro necesita de un software para ser cambiado byte a byte [13].

Una especificación original indica que todas las coordenadas de los vértices debían ser positivas, aunque en la actualidad la mayor parte de los programas utilizados permiten colocar las facetas en localizaciones arbitrarias.

2.3 Impresora 3D

La impresora que vamos a usar es una Prusa. Esta impresora pertenece a la familia de las impresoras RepRap las cuales fueron creadas originalmente con el propósito de auto replicación de una impresora 3D.

La impresora Prusa está diseñada mayormente con partes plásticas que pueden ser impresas con la misma impresora y con una punta conocida como extrusor que se encarga de dispensar plástico PLA. Este plástico es un polímero biodegradable que se produce del ácido láctico. La punta del extrusor siempre tiene temperaturas altas para poder ir derritiendo el plástico antes de moldear el proyecto. Esta temperatura también puede ser controlada con el código G.

2.4 Código G

2.4.1 Introducción

También conocido como G-code, el código es un lenguaje de programación que es más usado en la tecnología CNC. Su propósito principal es convertir un modelo o comandos en órdenes exactas para una máquina. Le podemos decir a una máquina que hacer, como hacerlo y que tan rápido hacerlo.

En nuestro caso vamos a utilizar unas interfaces donde al ingresar un archivo de un modelo 3D, por ejemplo un archivo STL, nos devuelve el código G para la impresora. Este código G le va a dar órdenes a la impresora para que esta pueda moldear el proyecto.

2.4.2 Rutinas

Las órdenes creadas por el código G vendrán como vectores de posicionamiento. Dentro de esta información también vendrá la cantidad de material que tiene que ser usado en cada punto. Con estas dos órdenes básicas podemos poco a poco ir imprimiendo el proyecto.

Todas estas órdenes crean lo que llamamos rutinas. Estas se encargan de ir creando la impresión deseada. Son la lista de órdenes que deben ser realizadas para completar la impresión. Estas rutinas son creadas por los programas que usamos para convertir archivos de modelos a código G.

Si vamos a imprimir una figura muy simple como un cubo es muy fácil imaginarnos cuál va a ser la rutina que el programa va a designar para la impresora, pero mientras el modelo se hace más complejo es necesario el uso de una interface que nos ayude con el diseño.

Nuestro enfoque es estudiar y entender como estas interfaces diseñan estas rutinas y como estas poco a poco van creando la figura deseada. Para poder entenderlas es necesario familiarizarnos con el código G y sus órdenes básicas.

2.4.3 Órdenes básicas código G

Las siguientes son las órdenes básicas para generar código G [15]:

- G00 Interpolación Lineal Rápida

El código G00 es un código que le indica a la máquina que se mueva a un punto sin crear ningún tipo de trazo o utilizar material alguno. Este movimiento es mayormente usado para ubicar el extrusor en un punto neutro para comenzar el proceso de una impresión. El movimiento creado con el código G00 es un movimiento que no debe cruzarse con ninguna parte de la impresión para no causar daños

Ejemplo:

```
G00 X1 Y1
```

Esto le indica a la máquina que se ubique en la posición 1 en el eje X y 1 en al eje Y.

- G01 Interpolación Lineal a una velocidad programada

Este código es muy parecido al código G00, sus diferencias son que mientras el extrusor se mueve a el punto indicado tiene que hacerlo haciendo cortes o en nuestro caso ir utilizando material y además que este movimiento es a una velocidad que se le indica a la máquina.

Ejemplo:

```
G01 Z-0.125 F5
```

Esto le indica a la máquina que se ubique en la posición Z -0.125 y tiene que hacerlo a una velocidad de 5 pulgadas por minuto

- G02 y G03 Movimiento Circular

Estos códigos se encargan de hacer que el extrusor haga arcos y círculos. El G02 lo hace en sentido horario y el G03 en sentido anti horario. Hay dos maneras de dar estas órdenes.

La primera es utilizando un Radio

En esta orden se le da a la máquina un punto de inicio, un punto final y un valor R de radio.

Ejemplo:

```
G02 X2 Y2 I1 F20
```

Esta orden indica que el punto de inicio es 2 en el eje X y el punto final es 2 en el eje Y, el valor del radio es 1 y lo tiene que hacer a 20 pulgadas por minuto.

La segunda manera de utilizar estas órdenes es indicándole tres valores de coordenadas. Se le da a la máquina el valor inicial y final del círculo como en el ejemplo anterior pero también le damos la ubicación del centro de la circunferencia para que con esta información ya pueda dibujar el arco.

Ejemplo:

G03 X3 Y3 I0.5

Aquí le estamos diciendo a la impresora cual es el punto de inicio, el final y el centro de la circunferencia que es 0.5

- G04 Demora

Este código es para indicarle al extrusor que se debe quedar esperando en su ubicación actual una cierta cantidad de tiempo

Ejemplo:

G04 P2

El extrusor tiene que esperar en su posición 2 segundos

- G12 y G13 Pockets

Estas líneas de códigos son usadas para indicarle al extrusor que tiene que crear un círculo con una profundidad más grande que el tamaño de la punta del extrusor, es decir, se usa para la creación de huecos o “pockets”. La diferencia entra G12 y G13 es el movimiento horario o anti horario respectivamente.

- G15 y G16 Modo Polar

Estos dos códigos son usados para poder ingresar órdenes para el G00 y G01 en modo polar en vez de cartesiano. Con el código G16 se entra al modo polar y con el G15 se regresa al cartesiano.

Ejemplo:

G16

G01 X1 Y45

G15

G01 X2 Y3

Se cambia a polares y el extrusor se mueve al punto 2,7 en el eje x y 2,7 en el eje Y. Después se regresa al modo cartesiano y se mueve al punto 1,45.

- G17, G18 y G19 Planos

Estos códigos son usados para especificar en qué plano se está trabajando y en el cual se están ingresando las coordenadas. El G17 indica que es un plano X-Y, el G18 X-Z y el G19 Y

- G20 y G21 Unidades

Si se quiere trabajar en pulgadas es necesario utilizar el código G20 pero si se desea usar milímetros se tiene que introducir el código G21

- G28 Retorno a Referencia

Al comienzo del código podemos elegir un punto como referencia, varias veces durante en el código se necesitara regresar a este punto como un punto neutro. Al utilizar el código G28 podemos indicarle al extrusor que se mueva a ese punto. También tenemos que darle un punto intermedio, es decir la punta se moverá de su punto actual al punto medio y después al punto referencia.

Ejemplo:

(Punto de referencia: X0, Y0, Z0)

G28 X0 Y1 Z2

La punta se moverá de su ubicación actual al punto X0 Y1 Z2 y después a la referencia que en este caso es X0 Y0 Z0.

- G90 y G91 Modos de Desplazamiento

Estos dos códigos cambian la manera de cómo las coordenadas que ingresemos van a ser representadas por la máquina. Si trabajamos con el código G90 esto quiere decir que vamos a trabajar con un modo absoluto. Las coordenadas que ingresemos de movimiento van a ser tomadas desde el punto de origen. Si se trabaja con el código G91 se está trabajando en modo relativo. Las coordenadas de movimiento ingresadas van a ser tomadas desde el punto en que se encuentra.[7]

El código G nos ofrece una gran variedad de códigos, pero estos son los más comunes y los cuales van a ser herramientas muy importantes en el desarrollo de nuestro proyecto.

CAPÍTULO 3

IMPLEMENTACIÓN DEL PROYECTO

3.1 Introducción

Es nuestra meta poder analizar las rutinas con las cuales se hacen las impresiones 3D. Estas rutinas son creadas por diferentes interfaces y permiten al usuario crearlas a base de código G sin tener que entender cómo funciona este.

Primero que todo utilizaremos unas interfaces que nos dejan crear modelos en 3D. Una vez tengamos estos modelos necesitaremos la ayuda de otros

programas que fácilmente nos convierten estos modelos en código G que será interpretado como rutinas al momento de imprimir. En este capítulo vamos a describir las diferentes interfaces que vamos usar en nuestro proyecto y cuáles son las herramientas que más nos ayudaron.

3.2 Diseño en STL

Como se explicó en el capítulo anterior, son pocos los programas que trabajan con extensión STL, por tanto, lo que se hace es trabajar en otros programas de CAD que exportan a STL.

Lo que decidimos es utilizar 2 figuras distintas con las que vamos a trabajar y a las que pensamos hacerles modificaciones para poder observar los cambios en sus rutinas y verificar el resultado final.

Para obtener un mejor resultado y enfocarnos específicamente en las rutinas optamos por usar figuras ya hechas que se pueden encontrar en internet, esta decisión fue tomada para no incurrir en los errores del modelado y exista ambigüedad en las juntas de las figuras, además para obtener figuras con mayor relieve.

Pero en caso de querer realizar figuras propias existen algunos programas con los que se puede trabajar en CAD y exportar a STL como lo son:

- BLENDER
- LIGHTWAVE
- AUTOCAD 3D

Entre otros programas, algunos de estos programas tienen un alto precio en el mercado por su complejidad y otros como el BLENDER son gratuitos pero no por esto dejan de tener un gran potencial.

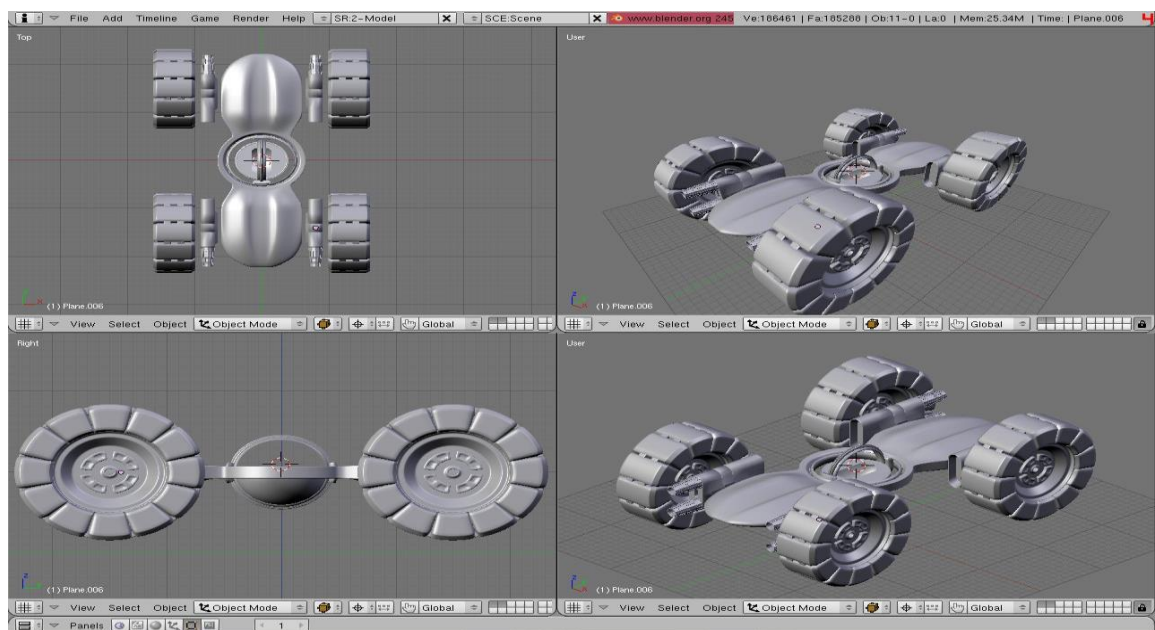


Figura 3.1 Modelo en CAD con el programa BLENDER [16]

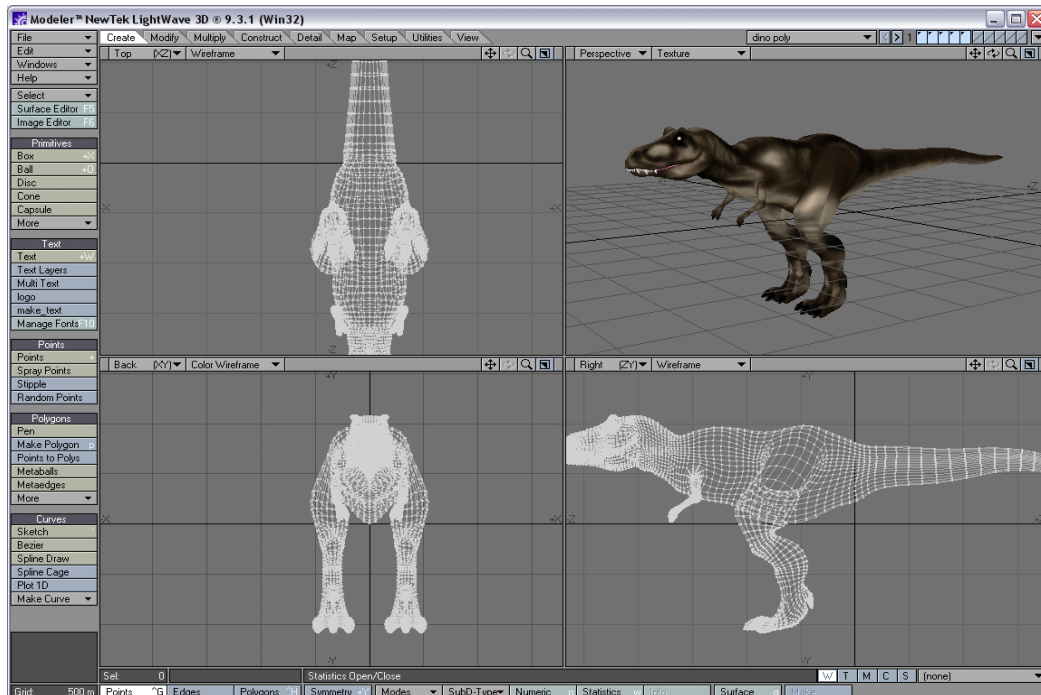


Figura 3.2 Modelo en CAD con el programa LIGHTWAVE [17]

3.3 Arduino Software

El primer programa que utilizaremos es el software del Arduino. Este puede ser descargado de la página oficial. Nos permite con un computador poder ingresar líneas de código C para que este cumpla con los requerimientos necesarios para realizar un proyecto. [18]

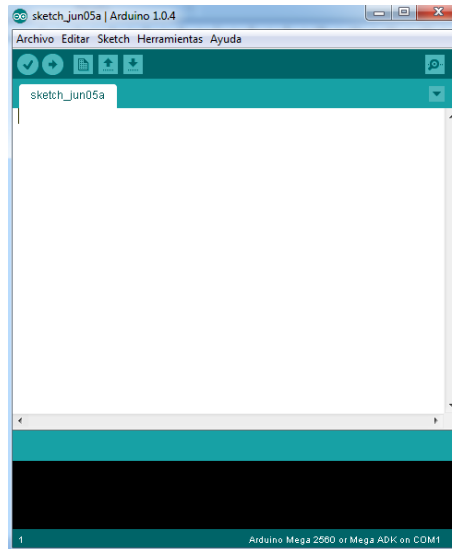
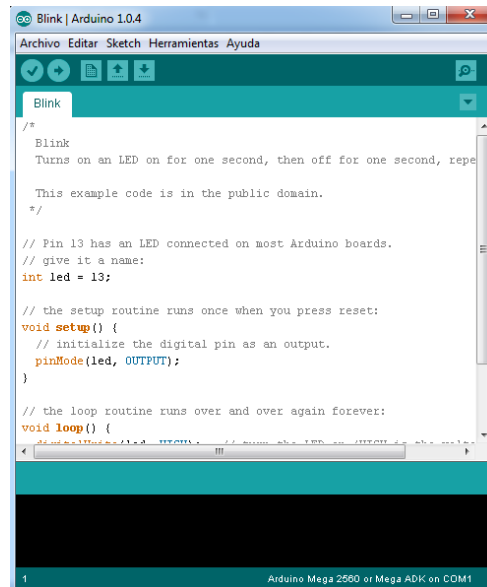


Figura 3.3 Pantalla principal Arduino Software

En la Figura 3.1 podemos ver la página de inicio del software. Esta interfaz es muy simple y fácil de entender para el usuario. Antes de cargar el software para que el Arduino maneje la impresora podemos ingresar a la placa otros programas básicos que ya vienen cargados en el software del Arduino.

El programa más básico es conocido como Blink y su única función es hacer que el led que se encuentra en la placa se encienda de manera intermitente. Estos programas de prueba son usados para poder confirmar que la placa se encuentra en buen estado y que la conexión del computador y el Arduino sea correcta.[19]

The image shows a screenshot of the Arduino IDE interface. The window title is "Blink | Arduino 1.0.4". The menu bar includes "Archivo", "Editar", "Sketch", "Herramientas", and "Ayuda". The toolbar contains icons for file operations and execution. The main text area displays the following code:

```
/*  
 * Blink  
 * Turns on an LED on for one second, then off for one second, repeatedly.  
 *  
 * This example code is in the public domain.  
 */  
  
// Pin 13 has an LED connected on most Arduino boards.  
// give it a name:  
int led = 13;  
  
// the setup routine runs once when you press reset:  
void setup() {  
  // initialize the digital pin as an output.  
  pinMode(led, OUTPUT);  
}  
  
// the loop routine runs over and over again forever:  
void loop() {  
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000);              // wait for a second  
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW  
  delay(1000);              // wait for a second  
}
```

The status bar at the bottom indicates "1" and "Arduino Mega 2560 or Mega ADK on COM1".

Figura 3.4 Programa Blink ingresado en el software Arduino

Una vez que ya hemos hecho pruebas con el Arduino podemos proceder a cargar el programa Sprinter. Este programa se encargara de preparar la placa para convertir las rutinas del código G en órdenes para los motores de la impresora.

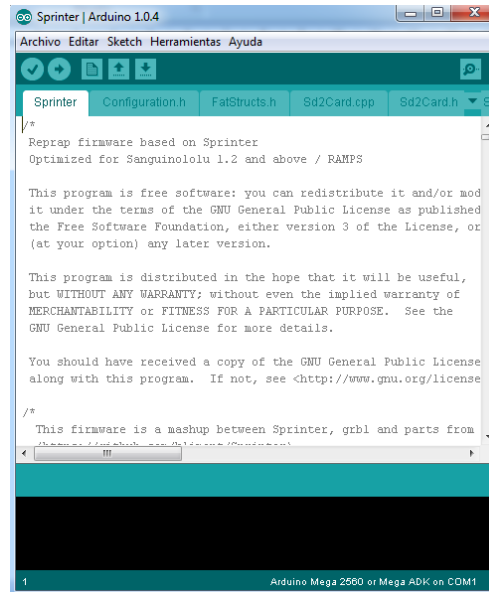


Figura 3.5 Programa Sprinter cargado en el software Arduino

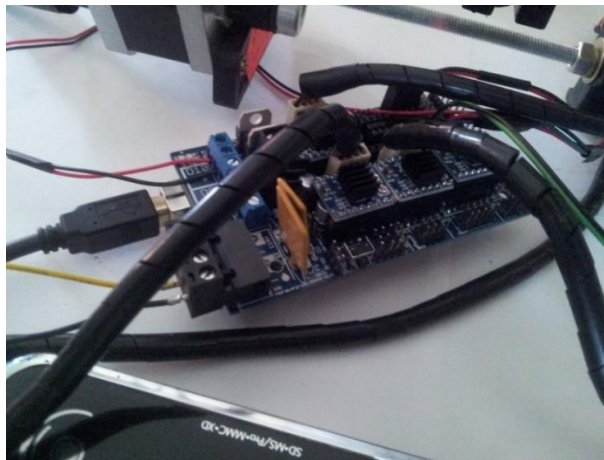


Figura 3.6 Arduino programado y conectado a la impresora y computador

3.4 Repetier Host

Repetier es un software gratuito diseñado para convertir modelos STL a código G.

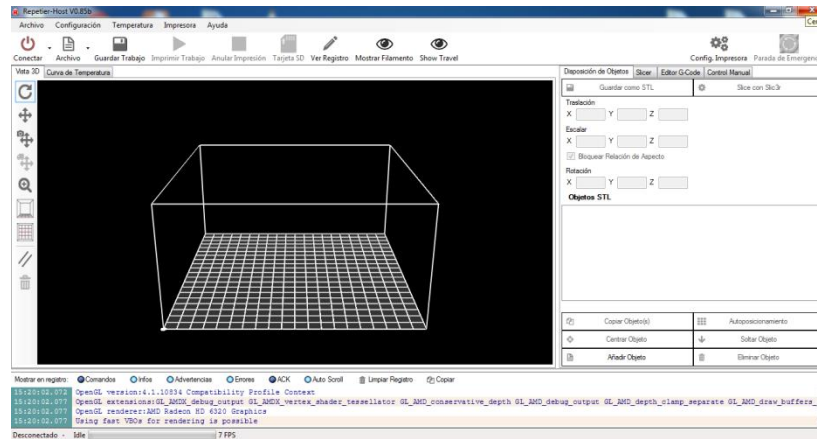


Figura 3.7 Pantalla principal de Repetier Host

Aparte de ayudarnos con la conversión de STL a código G, el Repetier también nos ayuda a manejar y modificar el STL. Esto puede ser de mucha ayuda si necesitamos hacer algún cambio rápido en el modelo antes de la impresión.

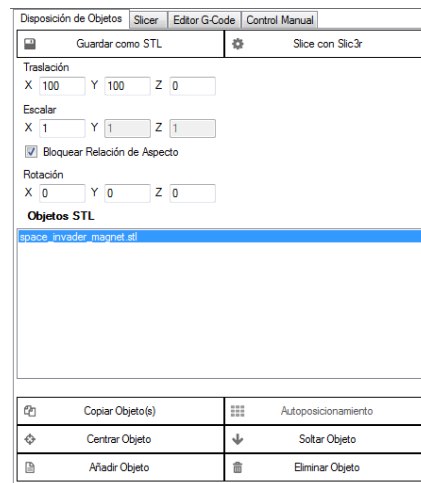


Figura 3.8 Opciones de modificación del STL

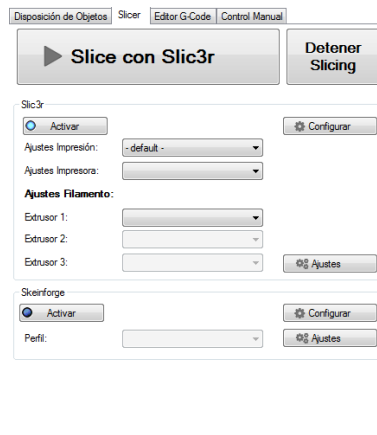


Figura 3.9 Opciones de conversión a código G

Para convertir un STL a código G, Repetier tiene integrado una versión de Slic3r. Dentro de Repetier podemos cambiar las configuraciones del Slic3r que viene integrado.



Figura 3.10 Opciones de conversión a código G

Repetier también puede ser configurado para trabajar directamente con la impresora. Una pestaña nos da los controles directos de la impresora.

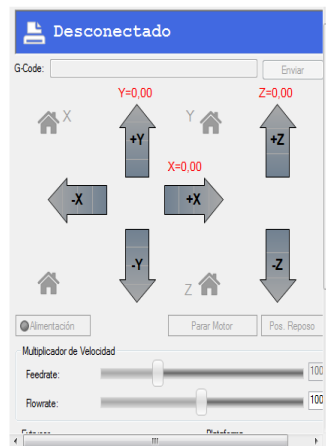


Figura 3.11 Controles de la impresora

3.5 Slic3r

Aunque a primera vista Slic3r parece un programa muy simple, este nos brinda muchas opciones muy útiles. Slic3r fue creado en el 2011 y desde ahí se ha mantenido en un continuo desarrollo. Su función principal es convertir un modelo 3D en código G.[20]

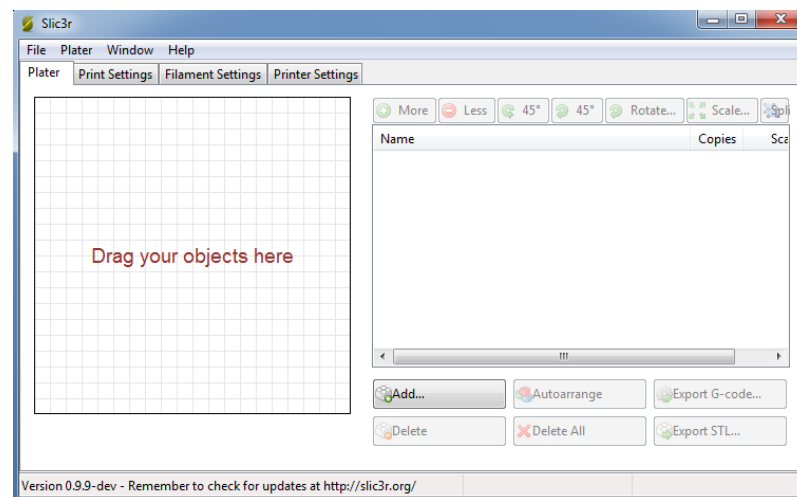


Figura 3.12 Pantalla principal del Slic3r

Como podemos apreciar en la imagen el Slic3r también tiene una interfaz muy simple y agradable para el usuario.

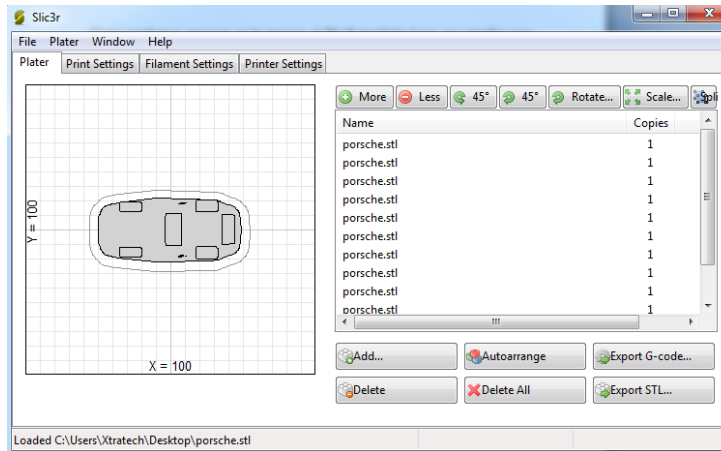


Figura 3.13 Modelo de un carro cargado en el Slic3r

Una desventaja de este programa es que no es compatible con todos los tipos de impresoras 3D. Por otro lado su ventaja es que el código G que este genera tiende a ser simple y fácil de entender.

Para la creación del código G Slic3r nos da varias opciones como el espesor de cada línea de impresión, la cantidad de material, cantidad de capas de base, etc.

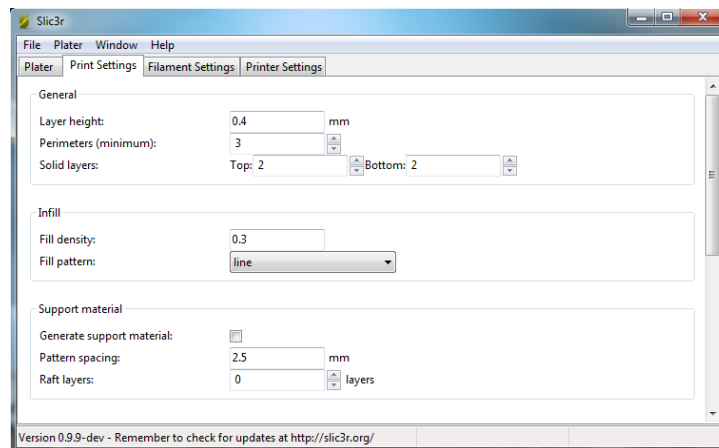


Figura 3.14 Opciones de impresión de Slic3r

3.6 ProcessingGCodeViewer

Existen muchos programas para analizar código G, nosotros vamos a trabajar con el ProcessingGCodeViewer. Este es un pequeño proyecto creado por una pequeña comunidad en internet. Su simpleza es su mejor característica, nos deja analizar las rutinas de impresión creadas por los programas fácilmente.

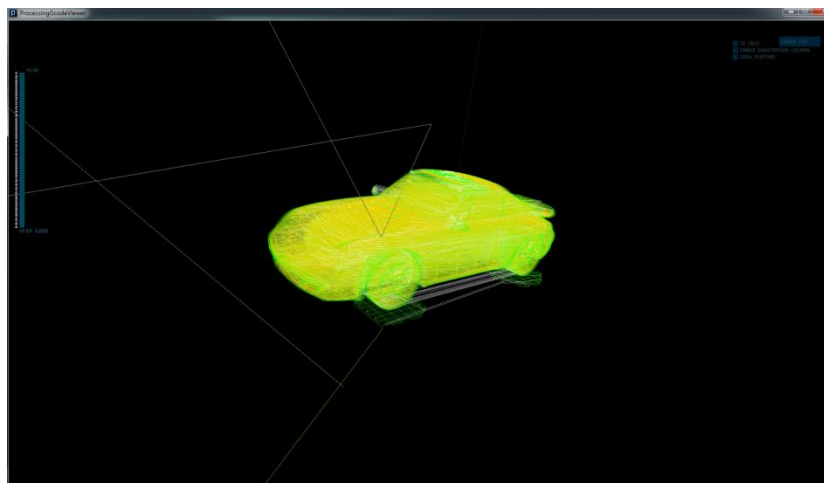


Figura 3.15 ProcessingGCodeViewer analizando un STL de un carro

3.7 Creación de las rutinas del código G

Para poder analizar las rutinas, vamos a trabajar con dos modelos STL. El primero es conocido como un Invader y el segundo es un Engranaje. Para poder analizar las rutinas para cada modelo vamos a utilizar dos configuraciones diferentes para la creación del código G. Las modificaciones que tendrá cada configuración serán:

- Espesor de impresión de capas externas
- Cantidad de capas en su parte inferior
- Cantidad de capas en su parte superior
- Tipo de relleno
- Espesor de impresión para el tipo de relleno

Espesor de impresión de capas externas.- se refiere al ancho que tendrán las líneas de impresión en las capas superiores como inferiores.

Cantidad de capas en su parte inferior.- hace referencia a la cantidad de capas que se imprimirán en la parte inferior del diseño.

Cantidad de capas en su parte superior.- hace referencia a la cantidad de capas que se imprimirán en la parte superior del diseño.

Tipo de relleno.- Cuando hablamos del tipo de relleno nos referimos a las capas intermedias que no van a poder ser vistas después de la impresión. Estas capas no son rellenas de material ya que se gastarías mucha materia prima, es por esto que se tiene que elegir qué tipo de relleno deseamos.

Espesor de impresión para el tipo de relleno.- se refiere al ancho que tendrán las líneas de impresión en las capas interiores.

Las configuraciones a usar serán conocidas en adelante como Configuración #1 y Configuración #2.

MODIFICACIONES	CONFIGURACIÓN #1	CONFIGURACIÓN #2
ESPESOR DE IMPRESIÓN DE CAPAS EXTERNAS	0,5 mm	0,3 mm
NÚMERO DE CAPAS INFERIORES	3	1
NÚMERO DE CAPAS SUPERIORES	3	1
TIPO DE RELLENO	HoneyComb	Rectilinear
ESPESOR DE IMPRESIÓN PARA EL TIPO DE RELLENO	0,5 mm	0,3 mm

Tabla 3.1 Especificaciones de las Configuraciones

La tabla 3.1 se pueden apreciar las diferencias entre cada una de las configuraciones, lo que esperamos lograr es ver una gran diferencia entre la calidad y los tiempos de impresión.

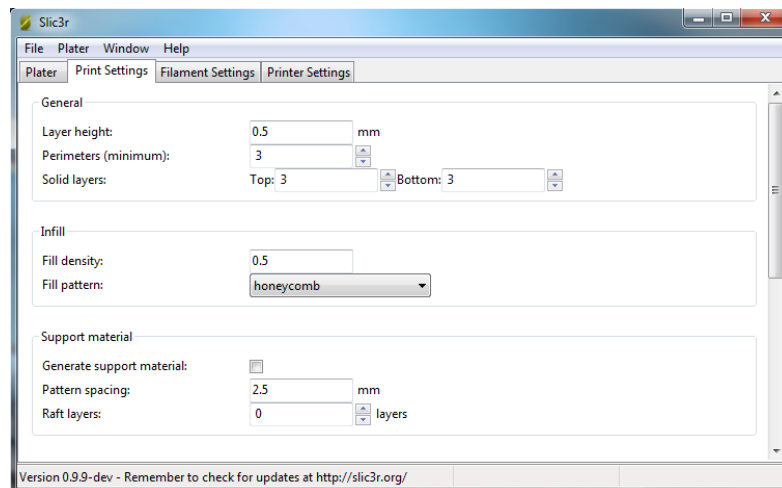


Figura 3.16 Configuración #1

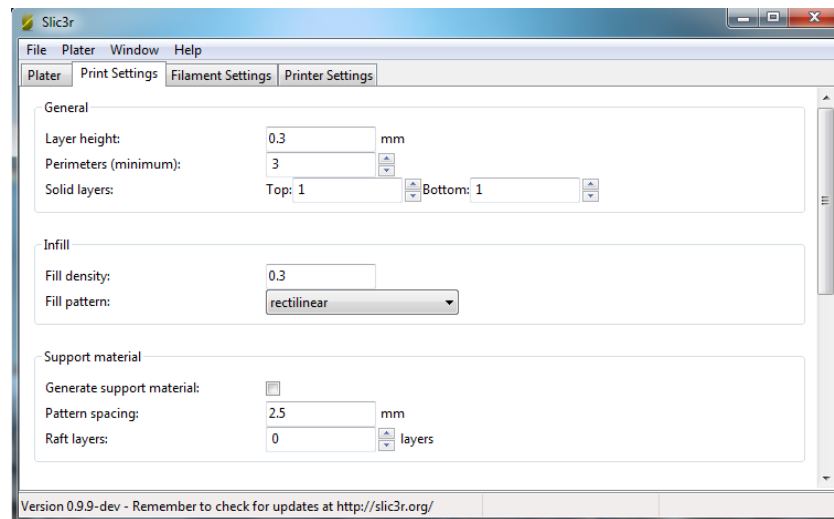


Figura 3.17 Configuración #2

3.8 Creación del código G del Invader

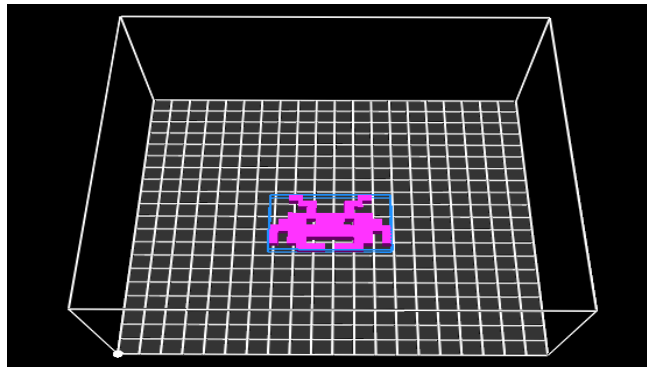


Figura 3.18 Modelo Invader

El primer paso va a ser cargar el modelo en el Slic3r. Para poder generar el código G tanto para la Configuración #1 como para la Configuración #2

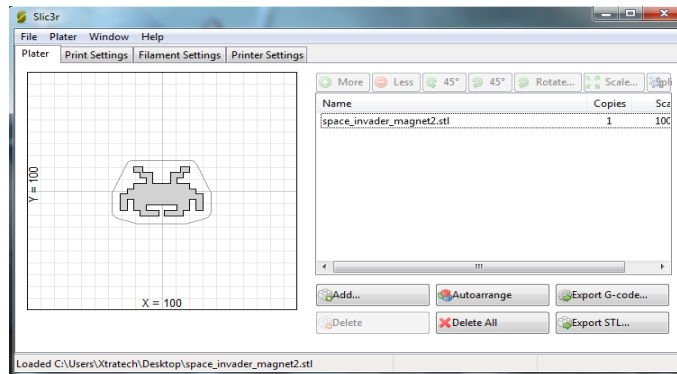


Figura 3.19 Invader cargado en el Slic3r

3.9 Creación del código G del Engranaje

Ahora que ya tenemos los códigos del Invader podemos empezar a realizar el mismo proceso con el modelo de una Engranaje.

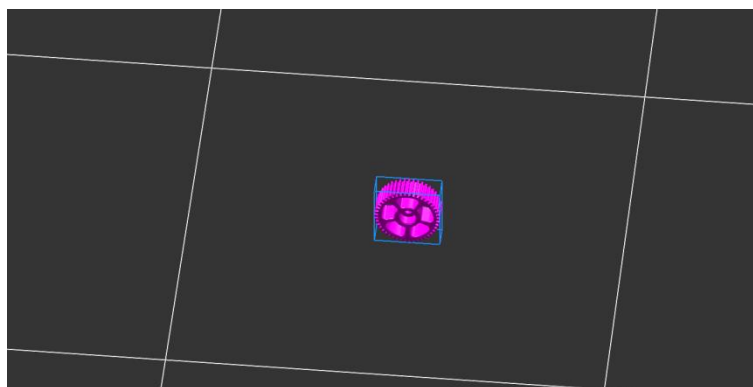


Figura 3.20 Modelo original de la Engranaje

Como se puede apreciar en la figura 3.20 el modelo del Engranaje nos presenta dos dificultades. La primera es que este modelo es muy pequeño por lo cual no podríamos imprimirlo fácilmente y otro problema es la orientación del engranaje. El modelo se encuentra de manera vertical, el problema es causado es debido a que el plástico está a altas temperaturas cuando se imprime. Esto causara desfiguraciones en el modelo final.

Esto ocurriría porque el momento que la impresora haga las partes huecas de la figura estas no tendrían donde asentarse. Debido a esto vamos a utilizar el programa Repetier para poder modificar el modelo en escala y orientación para poder obtener un modelo óptimo para impresión 3D.

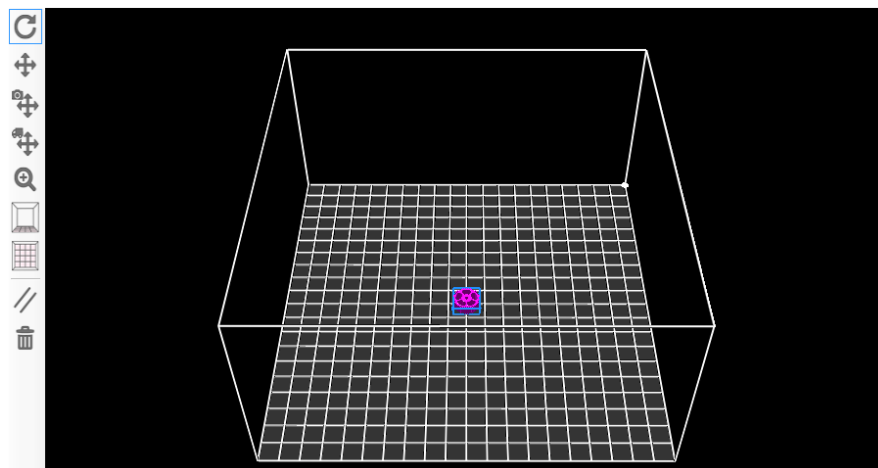


Figura 3.21 Modelo del Engranaje después de la modificación

Como podemos observar en la figura 3.21 el modelo del Engranaje ya se encuentra de manera horizontal y es 10 veces más grande. Se pudo haber agrandado mucho más el modelo pero hemos decidido dejarlo relativamente pequeño para observar las limitaciones de la impresora al momento de imprimir objetos de menor tamaño. Después de este paso vamos a proceder a crear dos códigos G con las Configuraciones #1 y #2 de manera exacta a como se lo hizo con el Invader.

CAPÍTULO 4

RESULTADOS Y ANÁLISIS

4.1 Introducción

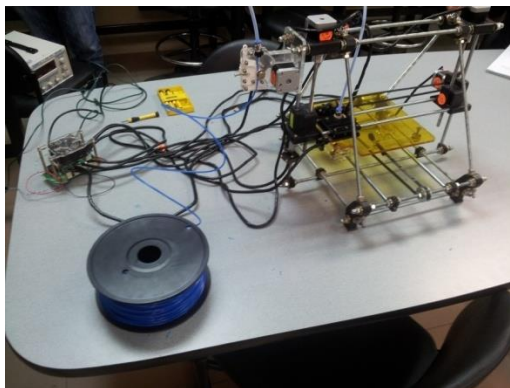


Figura 4.1 Impresora 3D conectada al suministro de plástico

Ahora que ya tenemos los Códigos G de los dos modelos podemos empezar a analizar como estas variaciones en las rutinas nos dan modelos diferentes aun cuando estos vinieron del mismo modelo STL. El análisis será de dos partes, primero analizaremos estos códigos con el programa ProcessingGCodeViewer y después analizaremos las figuras impresas

4.2 Análisis usando ProcessingGCodeViewer

El programa ProcessingGCodeViewer nos deja analizar las figuras que se imprimirían con nuestro código G. Una herramienta muy importante de este software es poder analizar capa por capa y poder mover la figura en un espacio 3D con facilidad.

Gracias a esta herramienta podemos ver que al realizar los cambios de Configuración #1 a Configuración #2 la cantidad de capas disminuyen o aumentan en nuestro Invader como en el Engranaje.

4.2.1 Rutinas del Invader

Primero analizaremos el modelo creado con la configuración #1

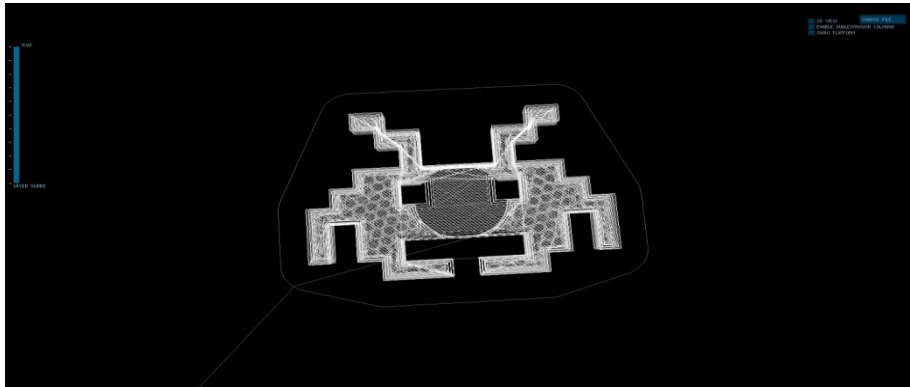


Figura 4.2 Invader Configuración #1

Lo primero que podemos notar es que el modelo fue hecho en 11 capas. Y si vemos en el centro de la imagen podemos ver como da la impresión de unos pequeños hexágonos, estas son las capas de relleno hechas con la configuración Honeycomb.

En la Figura 4.4 podemos analizar la primera capa a ser impresa. Esta capa como podemos ver tiene bastantes líneas de impresión, esto es debido a que esta es una de las 3 capas de base. Estas capas son impresas utilizando bastante material para que sean más sólidas. También podemos apreciar cómo se deja un espacio circular en la parte posterior de la figura, esto es debido a que el modelo del Invader fue diseñado para tener una cavidad en la parte de atrás para introducir un imán y que el modelo impreso sirva de adorno para cualquier superficie metálica.

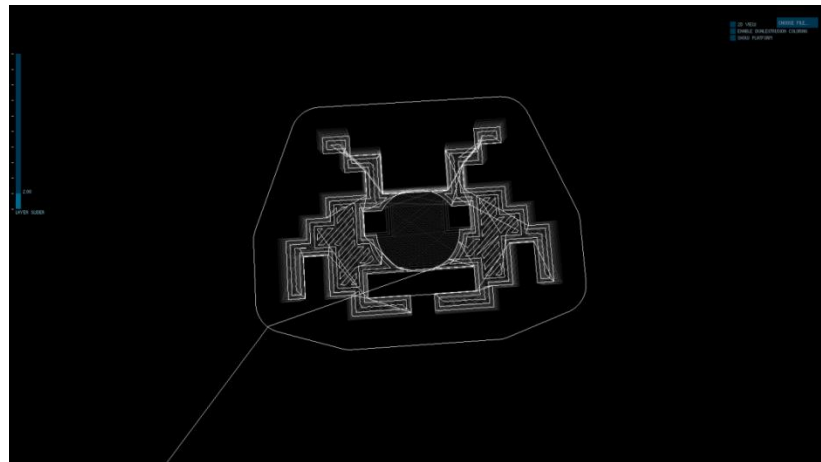


Figura 4.3 Capa 1 del Invader Configuración #1

En la capa 3 podemos ver como se siguen imprimiendo varias líneas. Esta es la última capa de base.



Figura 4.4 Capa 3 del Invader Configuración #1

En las capas intermedias podemos apreciar mejor el diseño de relleno Honeycomb.

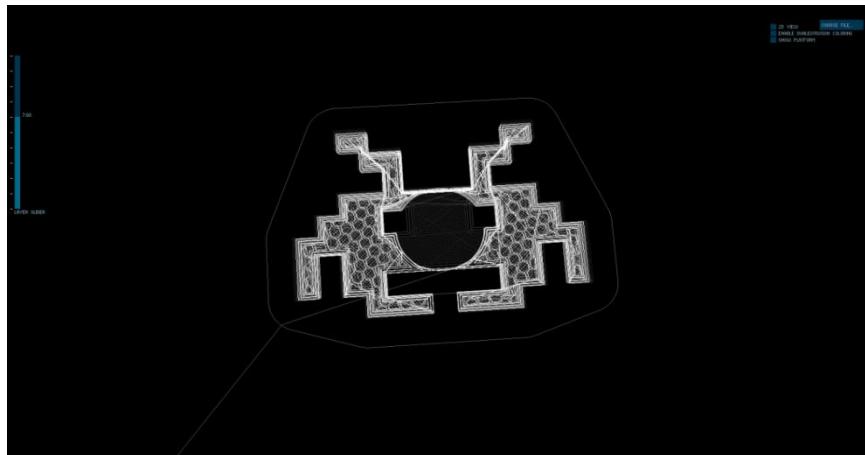


Figura 4.5 Capa 7 del Invader Configuración #1

En la capa 11 podemos ver como otra vez se empiezan a crear varias líneas de impresión para sellar la figura.

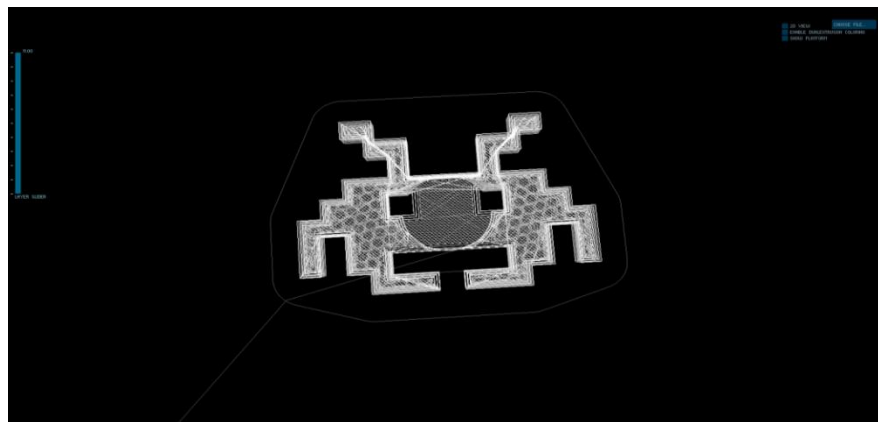


Figura 4.6 Capa 11 de Invader Configuración #1

Ahora podemos analizar las rutinas del Invader con la Configuración #2.

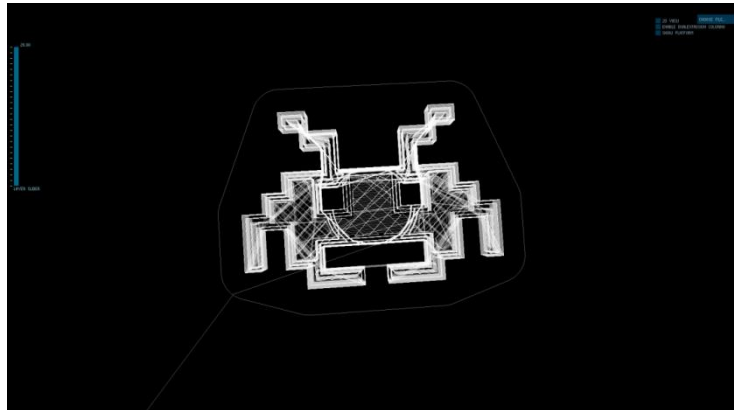


Figura 4.7 Invader Configuración #2

A primera vista podemos darnos cuenta como el diseño ahora va a ser impreso en 26 capas. Al ver el modelo este parece ser un poco más hueco que el de la Configuración #1. Se ven menos líneas en su interior.

Al analizar su primera capa vemos que esta se parece mucho a la primera capa de la configuración #1 ya que son capas base pero en la configuración #2 solo existe una capa así.

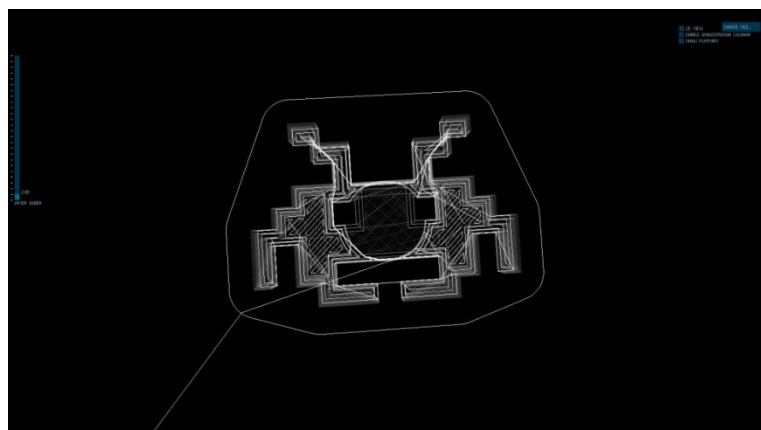


Figura 4.8 Capa 1 del Invader Configuración #2

Desde la capa 2 en adelante ya se empiezan a imprimir las capas de relleno que en este caso son tipo rectilíneas.

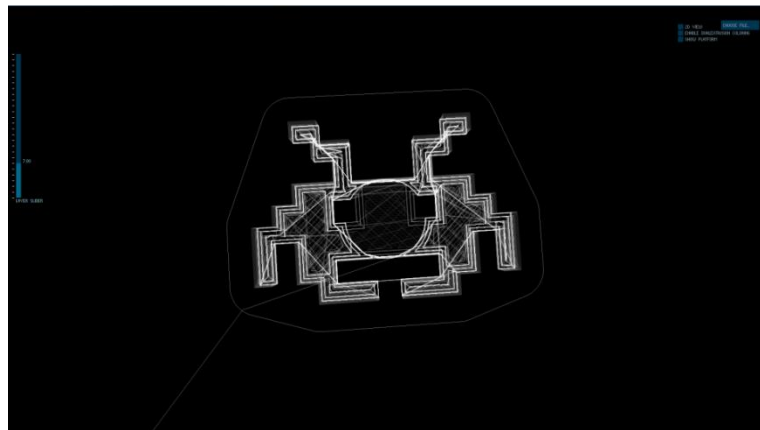


Figura 4.9 Capa 7 del Invader Configuración #2

En la última capa vemos como se vuelve a generar una capa igual a la primera para sellar la figura.

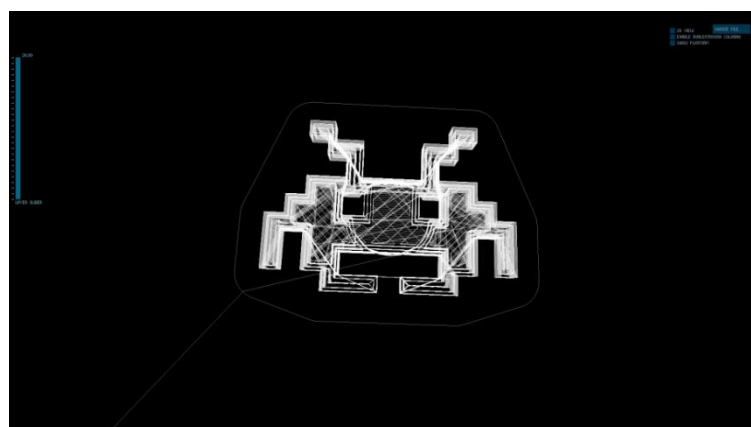


Figura 4.10 Capa 26 del Invader Configuración #2

4.2.2 Rutinas del Engranaje

Después de haber terminado el análisis del Invader podemos proceder a comenzar a realizar el mismo análisis del Engranaje. Empezaremos con la Configuración #1. Lo primero que notamos es como la rutina está compuesta de 10 capas

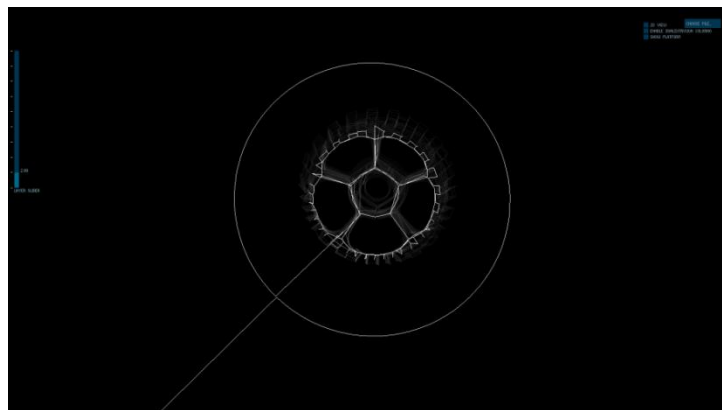


Figura 4.11 Capa 1 del Engranaje

En la primera capa podemos ver como se hace una capa muy simple, esto es debido a la forma de la figura. El Invader tiene una base plana y el Engranaje tiene una base curva. Vemos también en esta capa como se empiezan a crear pequeñas líneas de impresión alrededor del círculo principal para poder empezar a crear los dientes del engranaje.

En las siguientes capas podemos ver como se empiezan a imprimir las capas de relleno. Debido a la forma de la figura es más difícil poder apreciar el tipo de relleno Honeycomb.

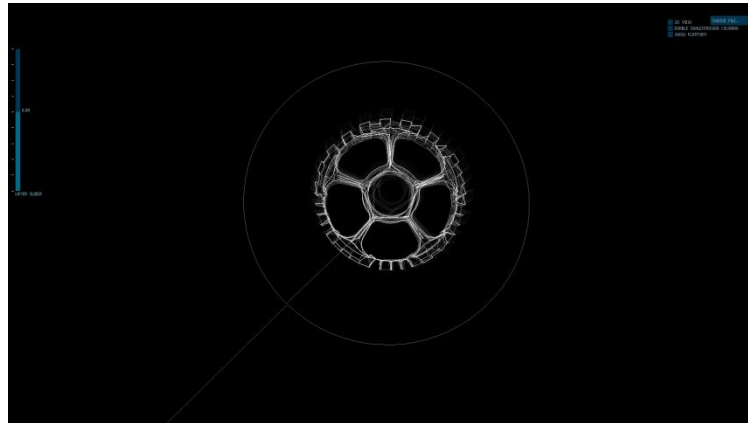


Figura 4.12 Capa de relleno del Engranaje Configuración #1

En la última capa podemos apreciar cómo se sella la figura.

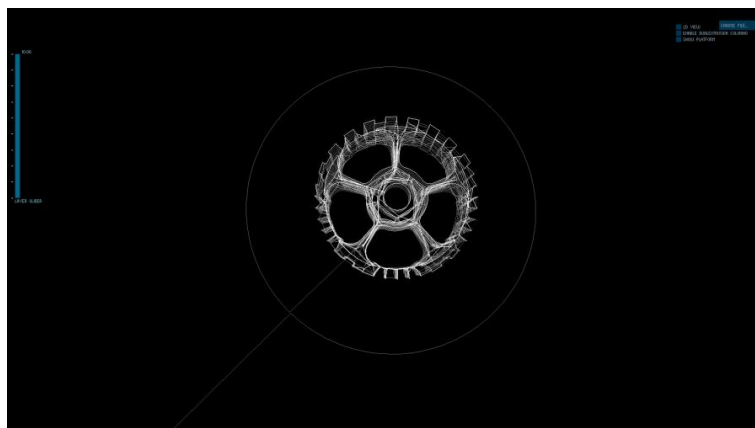


Figura 4.13 Capa 10 del Engranaje Configuración #1

Ahora procederemos a analizar las rutinas creadas por la Configuración #2. A primera vista podemos apreciar como en el caso del Engranaje tanto como del Invader la cantidad de capas subió de 10 a 16. En la primera capa podemos ver algo muy parecido a la Configuración #1 la única diferencia es que estas líneas tendrán un espesor menor.

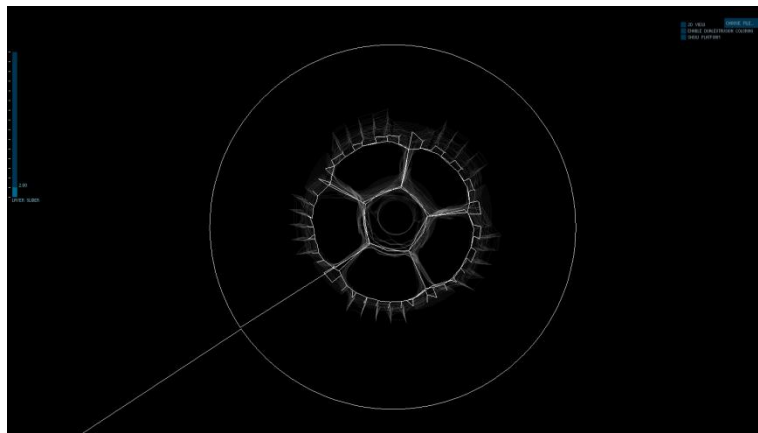


Figura 4.14 Capa 1 del Engranaje Configuración #2

En las capas de relleno también podemos ver como son muy parecidas a las de la Configuración #1. En este caso tampoco se puede apreciar mucho el tipo de relleno que es Rectilíneo.

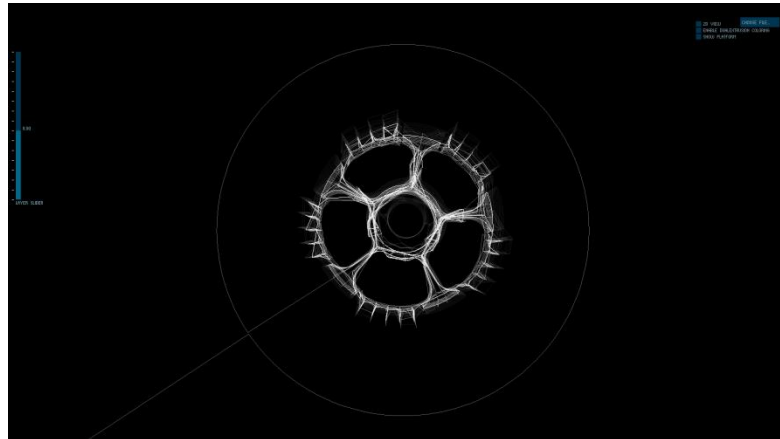


Figura 4.15 Capa 7 del Engranaje Configuración #2

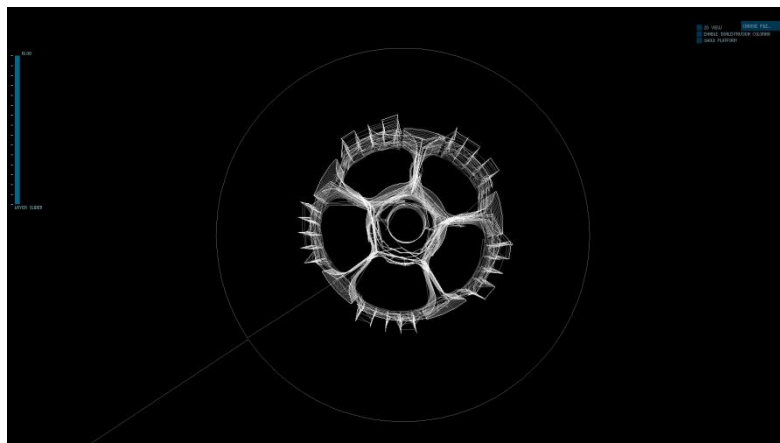


Figura 4.16 Capa 16 del Engranaje Configuración #2

4.2.3 Comparación de rutinas a partir del ProcessingGCodeViewer

Como pudimos apreciar encontramos muchas diferencias al analizar las rutinas. Fue muy fácil notar que cuando usamos la Configuración #2 se creaban más capas. Esto es debido a que en esta Configuración se usó un espesor más pequeño para cada línea de impresión.

En las rutinas del Invader pudimos analizar mejor el tipo de relleno. Esto se debe a que el Invader tiene una gran parte interna donde solo se tiene que imprimir relleno en cambio el Engranaje como es una figura más pequeña y tiene muchos detalles como los dientes donde no hay como apreciar el relleno.

Cuando usamos el ProcessingGCodeViewer para ver las figuras con todas las capas ya podemos darnos cuentas como el Invader de Configuración #1 se ve más lleno de líneas de impresión que el de la Configuración #2 esto va a hacer que la figura este mas llena de material.

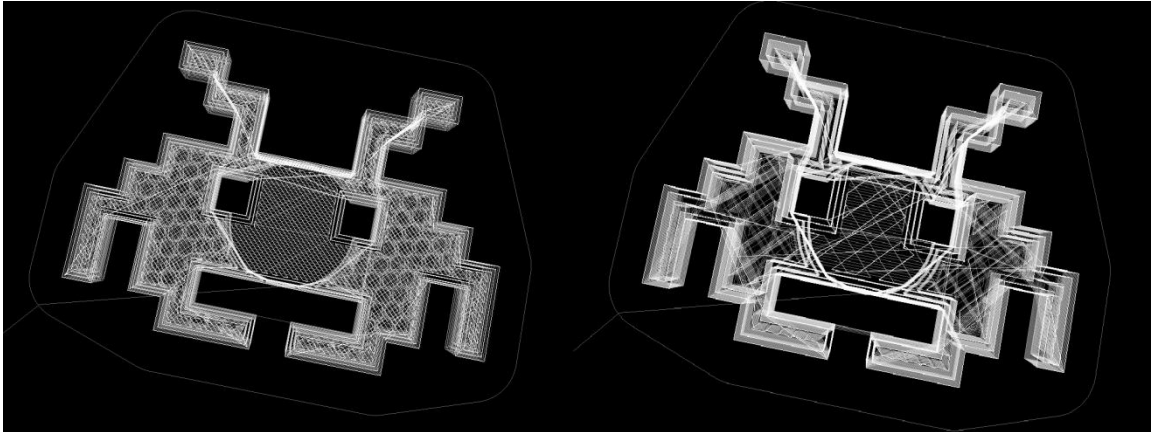


Figura 4.17 Comparación Invader Configuración#1 y #2

En las capas inferiores y superiores podemos ver patrones de impresión muy parecidos lo único que cambia es su espesor. Para poder seguir analizando las diferencias de estas dos Configuraciones debemos enfocarnos también en los modelos impresos.

4.3 Modelos Impresos

4.3.1 Introducción

Una vez que ya hemos comparado las rutinas por medio de simulaciones también debemos analizar y comparar los modelos impresos. En estos podremos ver como las rutinas hacen diferencias en el modelo final y

podremos llegar a conclusiones acerca de las ventajas y desventajas de cada configuración de rutina.

4.3.2 Modelo Impreso Invader

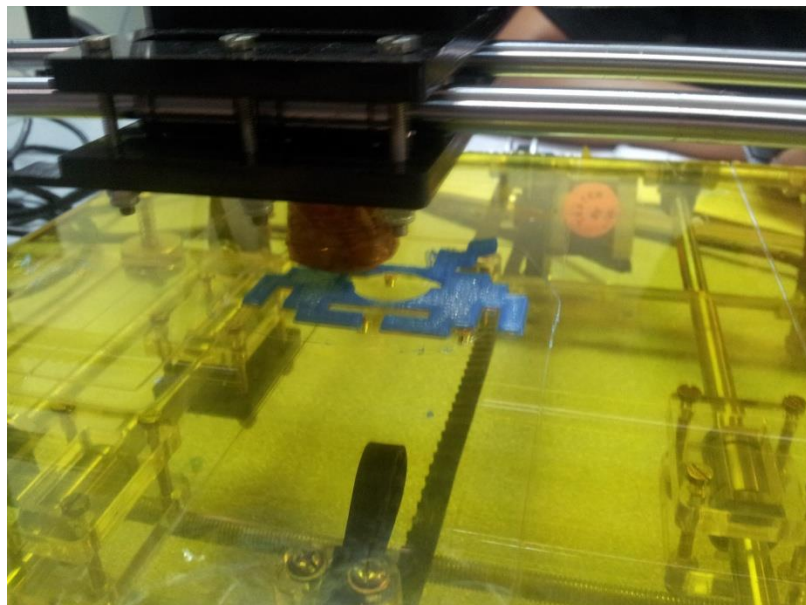


Figura 4.18 Imprimiendo el modelo Invader

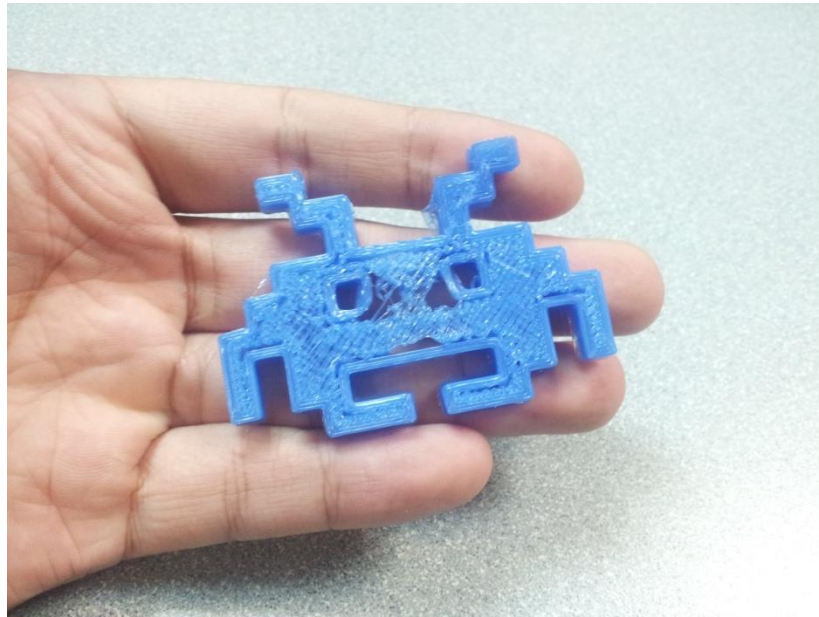


Figura 4.19 Impresión completa del Invader



Figura 4.20 Modelo Impreso Invader con la Configuración #1



Figura 4.21 Modelo Impreso Invader con la Configuración #2

Los modelos impresos del Invader con las Configuraciones #1 y #2 son muy parecidos a lo que pudimos simular en el ProcessingGCodeViewer. En la Configuración #1 podemos apreciar como se ve que su relleno tiene bastante material. En las esquinas podemos ver como no se llenó completamente de material debido a pequeñas imperfecciones al momento de imprimir. En el modelo impreso con la Configuración #2 se ve claramente como este tiene muchas líneas menos de impresión en su relleno.

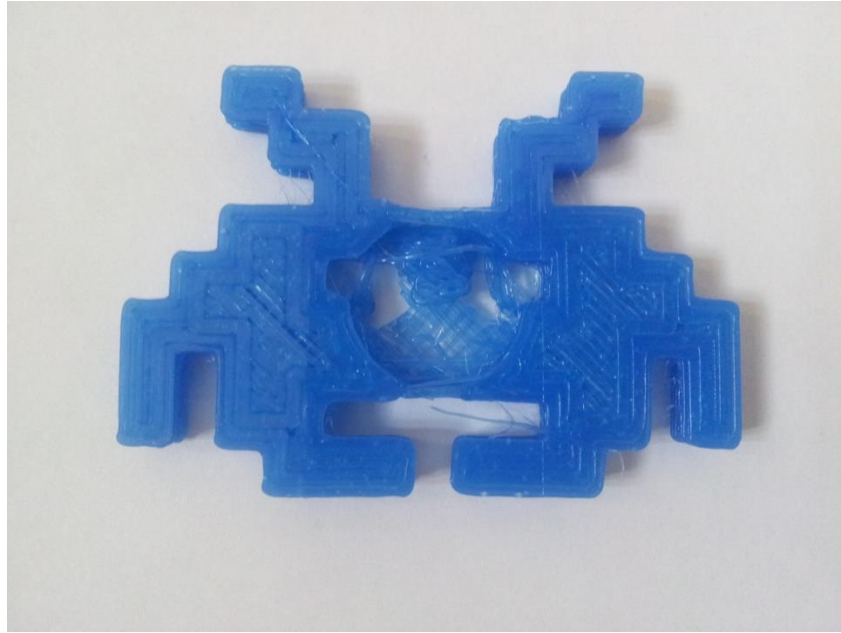


Figura 4.22 Parte posterior del Invader Configuración #1

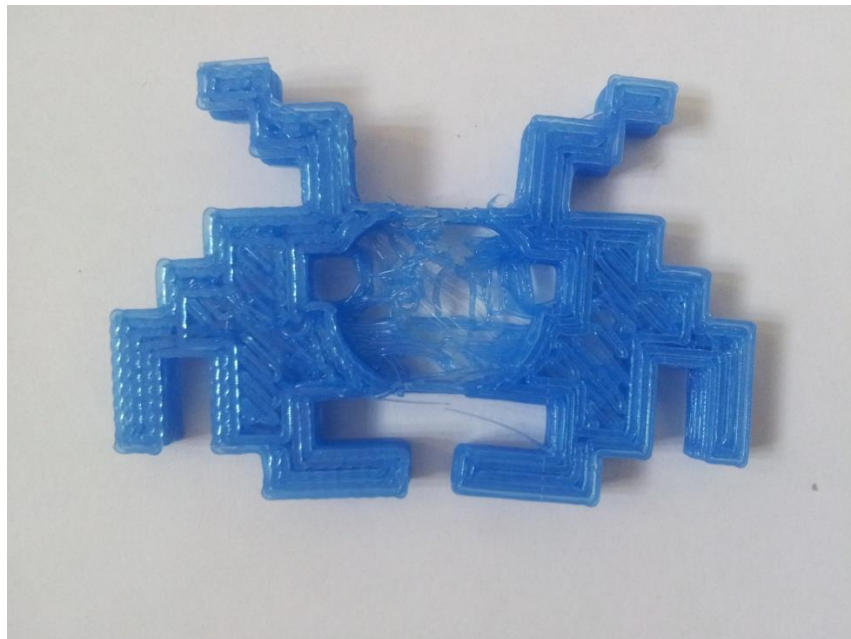


Figura 4.23 Parte posterior del Invader Configuración #2

En la parte de posterior de las figuras podemos apreciar como las dos tiene la misma capa inferior con la misma rutina impresa solo que el Invader hecho con la configuración #1 al tener más capas inferiores crea una base más sólida para el modelo.

El resultado después de imprimir con las dos configuraciones es la siguiente:

	CONFIGURACION #1	CONFIGURACION #2
TIEMPO	21 minutos	17 minutos
CANTIDAD DE MATERIAL EN CM	53 cm	43 cm

Tabla 4.1 Resultados del Invader

Estas diferencias son por el espesor de las líneas de impresión y la cantidad de líneas que hubo que imprimir.

4.3.3 Modelo Impreso Engranaje



Figura 4.24 Modelo Impreso Engranaje con la Configuración #1



Figura 4.25 Modelo Impreso Engranaje con la Configuración #2

Al analizar los modelos impresos de los Engranajes nos es tan fácil poder notar como los rellenos han hecho cambios en la figura. Lo que sí es claro a primera vista es como la figura hecha con la Configuración #2 está mejor definida y se asemeja más al modelo original.



Figura 4.26 Parte posterior del Engranaje con la Configuración #1



Figura 4.27 Parte posterior del Engranaje con la Configuración #2

En la parte posterior de las figuras impresas podemos ver las mismas observaciones que pudimos realizar de la vista superior.



Figura 4.28 Vista lateral del Engranaje con la Configuración #1

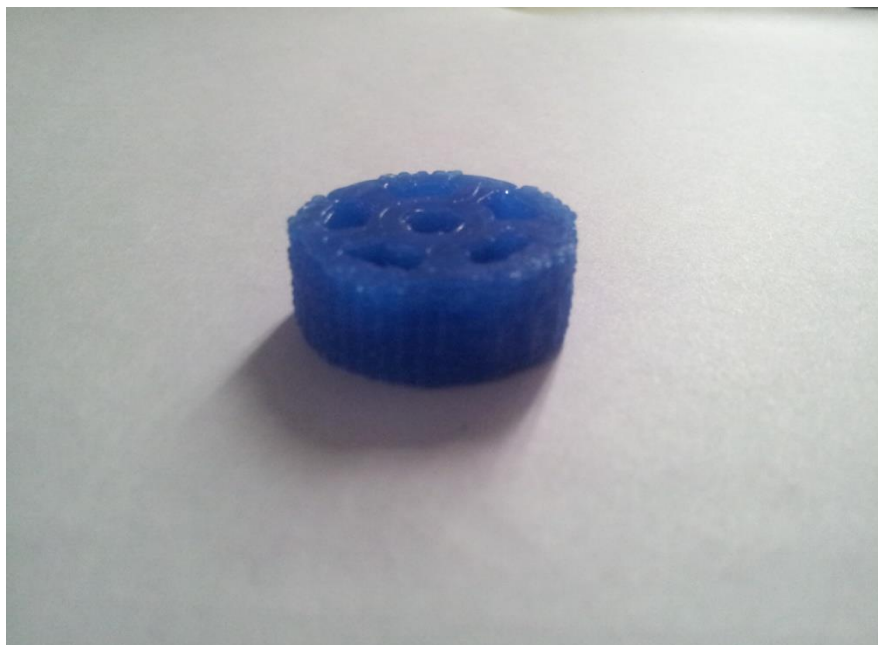


Figura 4.29 Vista lateral del Engranaje con la Configuración #2

Al ver la lista lateral de los modelos podemos ver como el modelo con la Configuración #2 vuelve a ser el que tiene mejor definición.

	CONFIGURACION #1	CONFIGURACION #2
TIEMPO	14 minutos	12 minutos
CANTIDAD DE MATERIAL EN CM	17 cm	15 cm

Tabla 4.2 Resultados del engranaje

Estas diferencias al igual que con el Invader son debido a las líneas de impresión y su espesor.

CONCLUSIONES

1. Al cambiar el espesor de las líneas de impresión la cantidad de capas aumenta, esto se debe a que aunque se cambie el espesor las dimensiones del modelo tienen que ser las mismas. Esto solo presenta una diferencia en el detalle de las capas laterales del modelo impreso.
2. El tipo de relleno que se usa va a definir qué tan sólida va a ser la figura. Esto puede ser un factor muy importante al momento de imprimir un modelo ya que ciertos modelos pueden tener un uso en el cual necesiten ser más durables. También va a definir los tiempos de impresión.

3. En el caso del Invader pudimos ver claramente como la Configuración #1 fue la más adecuada para imprimir este modelo. Esta decisión fue hecha tomando en cuenta la calidad y la durabilidad del producto final. Para el Engranaje los resultados fueron opuestos, la mejor impresión fue realizada con la rutina de la Configuración #2. Las líneas de impresión más finas hicieron que se pudiera crear un detalle más exacto en el modelo.
4. No hubo una configuración que haya tenido resultados favorables para los dos modelos, cada uno presentó mejores resultados con una configuración diferente. Esto nos lleva a concluir que para poder llegar a una impresión exacta se van a tener que ajustar las configuraciones y rutinas de impresión para cada modelo.
5. La impresión 3D es el futuro, tiene una gran gama de usos que van desde la educación pasando por la medicina, arquitectura hasta llegar a tener usos no tan positivos. Ya se ha impreso armas de fuego con impresoras 3D. Como toda tecnología la impresión 3D también tiene usos provechosos como dañinos para la sociedad.



Figura 4.30 Carcasa protectora para un Raspberry Pi

6. Las rutinas son la base de la impresión 3D, en este proyecto pudimos apreciar como realizando cambios en las rutinas llegamos a definiciones diferentes en los modelos. Para poder elegir rutinas más provechosas para nuestros proyectos es importante fomentar por parte de los desarrolladores herramientas simples que nos ayuden a la calibración de rutinas para minimizar errores.

7. Existen diferentes interfaces que nos ayudan con la impresión 3D. Cada una de estas tiene su ventaja y desventajas. Para poder saber cuál es la mejor interfaz para un usuario es necesario saber cuál será el uso de la impresora, si será solo para recreación o para una industria.

8. Al trabajar con el Arduino pudimos apreciar como esta placa tiene una gran variedad de usos. Su programación es fácil, esto es debido a la simpleza del software con el cual es programado. Los programas para el Arduino son relativamente fáciles de crear ya que están escritos en el lenguaje C.

RECOMENDACIONES

1. Recomendamos conocer los diferentes tipos de Configuraciones a la hora de imprimir, esto es muy importante ya que de esto dependerá nuestro tiempo de impresión y acabado final, además nos ayudara ya que como se explico la misma configuración no dará el mismo acabado a todas las figuras.
2. También es muy importante conocer el tipo de trabajo que se va a realizar para de ahí poder elegir la impresora a comprar ya que en el mercado existen una gran variedad de impresoras 3D, con diferentes precios y también diferentes especificaciones, y que van a dar un resultado final distinto aun cuando se imprima el mismo diseño.

3. Hay que tener en cuenta varios factores a la hora de imprimir como lo son la nivelación de la base donde se va a imprimir, la temperatura del extrusor y la alimentación eléctrica de los motores.

4. También se debe tener cuidado con el flujo del plástico, ya que este no debe ser interrumpido porque fallaría el resultado final, así que se recomienda tener liberado un poco de cable pero no tanto como para que este se enrede.

ANEXO

Extracto del código G:

; generated by Slic3r 0.9.9-dev on 2013-06-05 at 18:45:11

; layer_height = 0.5
; perimeters = 3
; top_solid_layers = 3
; bottom_solid_layers = 3
; fill_density = 0.4
; perimeter_speed = 30
; infill_speed = 60
; travel_speed = 130
; nozzle_diameter = 0.5
; filament_diameter = 3
; extrusion_multiplier = 1
; perimeters extrusion width = 0.53mm
; infill extrusion width = 0.53mm
; solid infill extrusion width = 0.53mm
; top infill extrusion width = 0.53mm
; first layer extrusion width = 1.00mm

G21 ; set units to millimeters

M107

M104 S200 ; set temperature

G28 ; home all axes

G1 Z5 F5000 ; lift nozzle

M109 S200 ; wait for temperature to be reached

G90 ; use absolute coordinates

G92 E0

M82 ; use absolute distances for extrusion

G1 F1800.000 E-1.00000

G92 E0

G1 Z0.500 F7800.000

G1 X64.931 Y78.120

G1 F1800.000 E1.00000

G1 X65.611 Y77.630 F540.000 E1.05293

G1 X66.091 Y77.350 E1.08802

G1 X66.851 Y77.010 E1.14059

G1 X67.111 Y76.920 E1.15796

G1 X81.281 Y72.200 E2.10107

G1 X82.201 Y71.960 E2.16111

G1 X82.511 Y71.910 E2.18094

G1 X83.141 Y71.850 E2.22090

G1 X116.851 Y71.850 E4.34953

G1 X117.481 Y71.910 E4.38950

G1 X118.101 Y72.030 E4.42937

G1 X118.711 Y72.200 E4.46936

G1 X133.141 Y77.010 E5.42984

G1 X133.901 Y77.350 E5.48241

G1 X134.381 Y77.630 E5.51750

BIBLIOGRAFIA

[1] Mariano Ojeda; <http://tecnologiadelosplasticos.blogspot.com/> ; Marzo del 2013

[2] Alberto Macias; <http://www.baquia.com/posts/2012-10-02-todo-lo-que-necesitas-saber-de-las-impresoras-3d> ; Marzo del 2013

[3] Carlos Castrillón; <http://www.avancetecnologicos.org/choc-creator-impresora-3d-en-chocolate.html> ; Marzo del 2013

[4] Bob Warfield; CNC Machine Overview and Computer Numerical Control History; <http://www.cnccookbook.com/CCCNCMachine.htm> ; Abril del 2013

[5] Casa de las Culturas;
<http://casadelasculturas.blogspot.com/search?q=arduino> ; Marzo del 2013

[6] Rómulo Benítez; Historia de Arduino y su nacimiento;
<http://botscience.wordpress.com/2012/06/05/historia-de-arduino-y-su-nacimiento/>;Marzo ; Marzo del 2013

[7] Mablack Robótica, <http://mablackrobotica.blogspot.com/>; Abril del 2013

[8] Carsa Android and More
<http://carsandroidandmore.blogspot.com/2013/04/impresoras-en-3d.html> ; Abril del 2013

- [9] Pablo Santamaría; ¿Qué es un Arduino? ;
<http://www.xatakahome.com/domotica/que-es-un-arduino-en-xataka-smart-home-te-lo-explicamos> ; Marzo del 2013
- [10] Wikipedia, <http://es.wikipedia.org/wiki/Arduino> ; Abril del 2013
- [11] Arduino, <http://arduino.cc/es/Reference/Board?from=Guide.Board> ; Marzo del 2013
- [12] Pablo Toimil; Tipos, características y diferencias de las placas de Arduino; Mackenzie;
http://es.makezine.com/2009/08/13/tipos_caractersticas_y_diferencias_de_pl/ ;
Marzo del 2013
- [13] Caddy News; El Formato de Fichero STL;
http://www.caddypain.com/noticias/May06_4.htm; Marzo del 2013
- [14] Taller de soluciones,
<http://tallerdesoluciones.blogs.inti.gob.ar/2009/10/22/%C2%BFque-es-un-archivo-STL/>; Marzo del 2013
- [15] Bob Grosse, Summary of G-codes;
http://www.tormach.com/g_code_table.html; Mayo del 2013

- [16] Blender artists <http://blenderartists.org/forum/showthread.php?119645-Futuristic-F1-for-2008-Blender-F1-Challenge>; Marzo del 2013
- [17] Mr John Lim, <http://5gamer.wordpress.com/2008/11/04/> ; Marzo del 2013
- [18] Michael McRoberts, Beginning Arduino, Marzo del 2013
- [19] Massimo Banzi, Getting Started with Arduino, Marzo del 2013
- [20] Brian Evans, Practical 3D Printers The Science and Art of 3D Printing, Abril del 2013