



UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior de Jaén

Trabajo Fin de Grado

DISEÑO DE UN SISTEMA DE ACABADO PARA PIEZAS IMPRESAS

Alumno: Abraham Mengíbar Caballero

Tutor: Rubén Dorado Vicente
Dpto: Ingeniería Mecánica

Septiembre, 2016



Universidad de Jaén
Escuela Politécnica Superior de Jaén
Departamento de Informática

Don Rubén Dorado Vicente, tutor del Proyecto Fin de Carrera titulado: Diseño de un sistema de acabado para piezas impresas, que presenta Abraham Mengíbar Caballero, autoriza su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, Septiembre de 2016

El alumno:

Los tutores:

Abraham Mengíbar Caballero

Rubén Dorado Vicente

Índice

1. INTRODUCCIÓN: INTERÉS Y OBJETIVO	5
1.1. Qué es el Prototipado rápido (RP).....	5
1.2. Qué es el Modelado por deposición fundida (FDM).....	8
1.3. Objetivo de este Trabajo Fin de Grado.....	13
1.4. Estructura de este Trabajo Fin de Grado.....	14
2. PROCESOS DE MEJORA DEL ACABADO SUPERFICIAL	15
2.1. Tratamientos químicos	16
2.1.1 Inmersión directa en el líquido abrasivo.....	16
2.1.2 Inmersión en atmósfera de vapor disolvente	18
2.2. Tratamientos mecánicos	20
2.2.1 Procesos mecánicos manuales	21
2.2.2 Procesos mecánicos utilizando equipos profesionales	21
2.3. Otros tipos de tratamientos superficiales.....	25
2.4. Pintura de vinilo líquido	26
3. DESARROLLO DEL SISTEMA	27
3.1. Diseño del sistema.....	27
3.1.1. Bocetos	27
3.1.2. Diseño en software CATIA	33
3.1.3. Archivos descargados de la página Thingiverse.....	41
3.2. Impresión 3D de las piezas de la máquina	43
3.2.1. Conversión en formato STL de los archivos CAD.....	43
3.2.2. Parámetros de impresión en el Software Repetier-Host.....	46
3.2.3. Envío a Impresora 3D de los ficheros STL	51
3.2.4. Extracción de las piezas impresas de la Impresora 3D	52
3.3. Corte de perfiles BOSCH y varillas lisas.....	52
3.4. Preparación de la electrónica	52
3.4.1. Modificación del código libre Marlin	53
3.4.2. Implementación en ARDUINO.....	56
3.4.3. Calibrado y puesta a punto.....	59
3.5. Ensamblaje del sistema	60
3.6. Problemas surgidos y soluciones en el proceso de desarrollo del sistema.....	64
4. REALIZACIÓN DE ENSAYOS	67

4.1.	Probetas.....	67
4.1.1.	Diseño de probetas	67
4.1.2.	Impresión de probetas.....	68
4.2.	Creación del programa G-code a introducir.....	70
4.2.1.	Transferencia del programa al sistema	71
4.3.	Manipulación de la pintura y adecuación del lugar	73
4.4.	Ensayos	73
4.5.	Medida de rugosidad.....	78
4.6.	Resultados obtenidos en los ensayos	79
4.7.	Procesado de los datos obtenidos.....	81
5.	DISCUSIÓN DE LOS RESULTADOS	83
6.	REALIZACIÓN DE ENSAYOS EN PIEZAS PROPORCIONADAS POR LA EMPRESA VALEO ILUMINACIÓN S.A.U.....	84
7.	CONCLUSIONES.....	87
1	ANEXO I: MODIFICACIONES DENTRO DEL CÓDIGO LIBRE MARLIN	89
2	Bibliografía	99

1. INTRODUCCIÓN: INTERÉS Y OBJETIVO

Durante los últimos años el mundo de la impresión 3D está creciendo de manera exponencial lo que nos lleva a plantearnos constantes mejoras en los procesos de **Modelado por Deposición Fundida** (*Fused Deposition Modeling*, FDM). La calidad de las piezas obtenidas mediante estos procesos depende de bastantes parámetros como pueden ser la velocidad de impresión, el color del hilo que se emplee, ya que según el color el material lleva distintos aditivos y el comportamiento llega a diferir, el material del hilo pudiendo ser desde plásticos de procedencia vegetal como el PLA hasta plásticos derivados del petróleo como el ABS o NYLON, el paso que tengan los motores que accionan la máquina de FDM o uno de los más importantes que es la altura de capa de impresión.

Este proceso de fabricación se utiliza para la creación de prototipos ya que permite crear de forma rápida, física y económica, modelos 3D generados en un ordenador pudiéndose evitar de este modo los altos costos que suponen los moldes y procesos de inyección de plástico. A este método de obtención del modelo físico de un prototipo virtual completo o solo ciertas partes se le denomina **Prototipado Rápido** (*Rapid Prototyping*, RP).

La lucha del método de FDM frente al de inyección radica en la calidad superficial que se obtiene puesto que mediante la inyección se pueden obtener superficies con unas tolerancias establecidas en el molde y superficies lisas mientras que en FDM las tolerancias superficiales varían en función de los parámetros anteriormente dichos obteniéndose unas superficies laterales rugosas. Para subsanar este problema, generalmente se recurre a procesos manuales. Este es el motivo por el cuál en este trabajo se va a desarrollar un sistema que permita, sin elevar los costes, mejorar el acabado superficial de forma semi-automática de las piezas impresas mediante FDM aplicando la novedosa pintura de vinilo líquido.

1.1. Qué es el Prototipado rápido (RP)

La impresión 3D también conocida como fabricación aditiva (del inglés *additive manufacturing*, AM) comenzó a desarrollarse en la década de 1980. En esta época se creó el sistema conocido como la estereolitografía, en el cual, las capas se añaden mediante el curado de fotopolímeros con láseres de luz ultravioleta. Lo que contribuyó

a la creación del diseño del formato STL (siglas provenientes del inglés “STereo Lithography”), ampliamente utilizado compatible con el software de impresión 3D.

“Los procesos de fabricación aditiva en la década de los 80 y principios de los 90 como por ejemplo la sinterización selectiva por láser o la fusión selectiva por láser, iban por su propio camino sin encontrarse ambos procesos. A pesar de que en esta industria se desarrollaron mejoras como la soldadura robot o el CNC (en inglés Computer Numerical Control), la idea de una nueva línea de producción capaz de transformar una materia prima en un prototipo físico creado capa a capa solo se asociaba a procesos de eliminación de material tales como el fresado CNC. Fue a mediados de los 90 cuando se desarrollaron nuevas técnicas de deposición de material que permitían crear nuevas geometrías las cuales requerían un alto grado de dificultad para los procesos de eliminación de material. (3dpinfo, 2016)”

Emergió así el Prototipado Rápido, ya que permite obtener prototipos o modelos creados en mediante diseño asistido por ordenador, CAD (del inglés computer-aided design), de una forma rápida y que suponía un menor coste comparado con la fabricación por eliminación de material. Este nuevo método de fabricación reduce el coste y el tiempo de fabricación de prototipos

En la Ilustración 1.1 se pueden ver ejemplos de prototipos creados a partir de Prototipado Rápido (all3dp.com, 2016)



Ilustración 1.1 Ejemplos de prototipos creados a partir de prototipado rápido (all3dp.com, 2016)

Como se puede apreciar en la ilustración 1.1, la utilidad del Prototipado Rápido es inmensa, desde diseño de elementos mecánicos como pueden ser los rodamientos, pasando por prototipos de vehículos tanto a escala como a tamaño real además de la fabricación de prototipos de prótesis corporales. Esto supone un empuje de todas las industrias que puedan integrar el Prototipado rápido dentro de su ámbito. Además de por la ya mencionada reducción de costes, por la minimización del tiempo de desarrollo de un producto desde su prototipo inicial hasta el producto terminado, es decir, disminuye el tiempo de diseño del ciclo de vida del producto, permitiendo de esta forma un desarrollo mucho más rápido y eficaz del producto final y teniendo en cuenta la capacidad de realizar ingeniería inversa de manera rápida y sencilla.

“Fue durante la época de los 90 cuando empezaron a surgir los múltiples tipos de prototipado rápido como son: Deposición de Plástico Fundido (FDM), Sinterización Láser Selectiva, Laminación de Deposición Selectiva (SDL) es decir utilizando láminas de papel, y los tipos más actuales como son: Inyección de Vinculación más conocido como la impresión 3D a color mediante polvo permitiendo la creación de prototipos con diferentes colores, Inyección Triple (Polyjet) que permite imprimir a la vez tres materiales diferentes, Procesamiento de Luz Digital (DLP) que utiliza un proyector para solidificar polímeros, muy similar al SLA, Fusión por Haz de Electrones (EBM) que funde el polvo metálico en cada capa.” (Sculpteo, 2016)

En este trabajo nos centraremos en el método de Deposición de Plástico Fundido ya que es de los más extendidos por su simplicidad, bajo coste y al alcance de muchas personas.

1.2. Qué es el Modelado por deposición fundida (FDM)

El fundador de la compañía líder de prototipado rápido, Scott Crump, a finales de la década de 1980 desarrolló, patentó esta tecnología, y fue comercializada en 1990. Este novedoso y revolucionario proceso de fabricación recibió el nombre de FDM (siglas de Fused Deposition Modeling) y que consiste en realizar una deposición de polímero fundido sobre una estructura a forma de soporte base, capa a capa. Dichas capas se solidifican de forma casi inmediata permitiendo la fabricación de objetos. Dependiendo de la geometría de la pieza en cuestión, es posible construir estructuras auxiliares a modo de soportes para lograr sujetar la geometría en las zonas más críticas antes de que se produzca su solidificación. En ciertos casos (impresión de alto nivel) si la máquina de impresión posee dos boquillas, es posible utilizar un material distinto para estas estructuras para permitir su eliminación mediante un proceso sencillo, que puede ser un proceso manual o mediante la aplicación de algún tipo de disolvente que deshaga el material de soporte sin perjudicar el objeto final. En cambio, si no es posible la utilización de un material distinto al del objeto (impresión de bajo nivel) se suele utilizar una densidad mucho menor para dichas estructuras, por lo cual dichos soportes serán mucho menos rígidos que el objeto final, permitiendo una eliminación sencilla sin dañar el objeto.

En la ilustración 1.2 se puede apreciar un esquema de impresora FDM con dos extrusores:

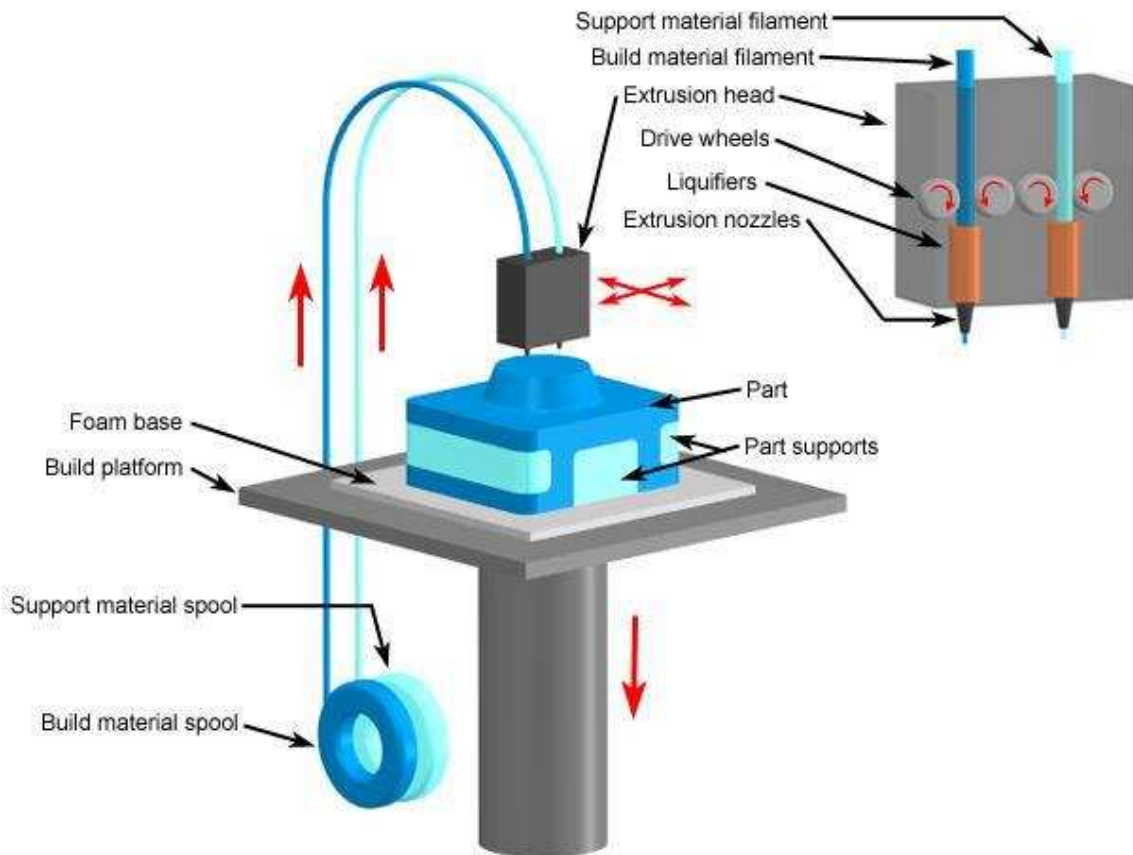


Ilustración 1.2 Esquema impresora FDM con doble extrusor (3dnatives, 2016)

El funcionamiento del sistema es sencillo, con el material, que se suministra bobinado en forma de hilo y puede ser de dos diámetros distintos, 1.95 mm y 3 mm, dependiendo del modelo de extrusor, se alimenta al extrusor mediante un tornillo estriado conocido como Hobbed-bolt que oprime el hilo contra un rodamiento obligando al hilo a entrar a través de un cambio de sección en el que una resistencia lo funde (Hot-End) y lo deposita sobre la cama soporte a través de la boquilla (Nozzle) la cual juega un papel muy importante en la calidad de impresión que depende del diámetro de la boquilla, comúnmente entre 0.1 y 0.4 mm . Por debajo de estas dimensiones existen los llamados nozzles HD (High Definition) con los cuales se llega a conseguir resultados finales muy precisos pero más lentos de fabricar. Para piezas grandes en las que prevalezca la rigidez frente a la calidad superficial se utilizan boquillas de diámetros más grandes. Para aumentar la velocidad de impresión se utilizan ventiladores que emiten un flujo directo de aire frío sobre la pieza que se está fabricando permitiendo un endurecimiento más rápido del material. Denotando que algunos materiales sufren defectos si se endurecen demasiado rápido, ya que las tensiones internas pueden llegar a deformar el objeto final.

Los materiales más conocidos y utilizados son el plástico de origen vegetal llamado PLA (Poliácido láctico) teniendo como ventajas frente a otros su fácil impresión, sus bajas temperaturas de fusión y una de las más importantes es que no desprende gases nocivos durante su fusión, el plástico derivado del petróleo ABS (Acrilonitrilo butadieno estireno) que es muy utilizado por su bajo coste, su alta resistencia a golpes y su buena flexibilidad. Como inconvenientes de este material se puede decir que existe una mayor dificultad de impresión ya que las tensiones internas que posee son muy grandes en el proceso de solidificación perjudicando a las piezas de gran tamaño además de que al ser derivado del petróleo, los gases que emite mientras se funde son nocivos para la salud. Existen más materiales como el Nylon muy resistentes a la tracción y elongación u otros muy novedosos como el conocido FLEX (termoplástico elastómero) que tiene la capacidad de deformarse o sufrir alargamientos y tras retirar dicho esfuerzo, es capaz de volver a su forma original, permitiendo la creación de piezas flexibles.

Además de los antedichos materiales puramente plásticos, se han creado materiales con un porcentaje de madera y otros que contienen cobre, plata o grafito que permiten obtener piezas conductoras de la electricidad utilizándose, por ejemplo, para la impresión 3D de circuitos impresos.

El campo de fabricación de FDM está en constante desarrollo, habiéndose podido adaptar a materiales como el acero o el vidrio y acabando con la barrera que se tenía con los materiales cerámicos ya que su proceso de fabricación necesita una presión y temperaturas que eran inalcanzables mediante la creación de resinas sensibles a la luz ultravioleta. Además de estos se han conseguido crear materiales orgánicos blandos como el chocolate o materiales biocompatibles que permiten fabricar prótesis tanto para personas como para animales.

En la ilustración 1.3 se muestra un ejemplo de una prótesis de tres costillas y una parte de esternón impresas en titanio mediante el proceso de FDM:

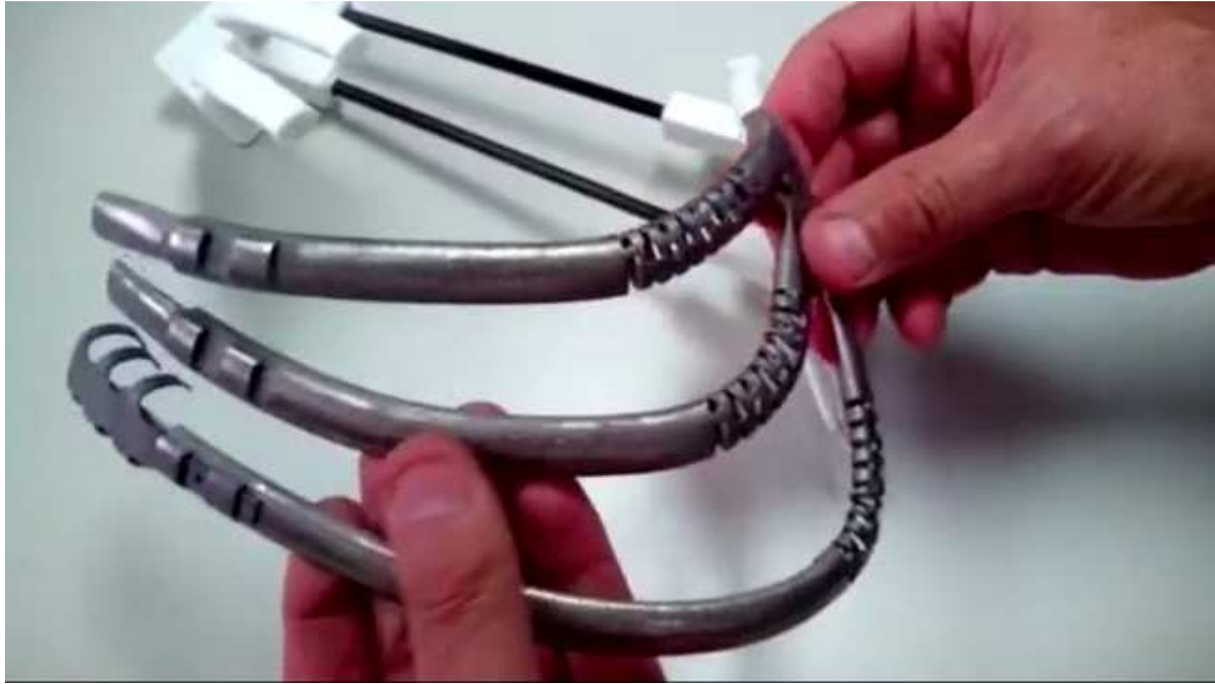


Ilustración 1.3 Prótesis impresa en titanio mediante FDM (Imprimalia3d, 2016)

El aumento del desarrollo de esta tecnología atrae a diversos campos ya que se pueden obtener piezas cuya geometría sea compleja de una forma muy rápida en comparación con procesos convencionales y de forma automatizada a excepción de las piezas que necesiten estructuras de soporte, que necesitarían la intervención humana para eliminar dichos soportes tras la finalización del proceso de FDM siendo este post-procesado normalmente de muy bajo coste y bajo tiempo de operación mientras que en otros procesos de fabricación como la inyección, además de ser un proceso mucho más caro debido al alto coste de la maquinaria necesaria, tiene un alto tiempo de post-procesado comparado con el FDM ya que habría que eliminar restos de colada, mazarotas y posibles rebabas.

El FDM ha llegado a poder fabricar con metales tales como el acero, y comparando este avance con el proceso más tradicional de fabricación por eliminación de material, es decir el mecanizado CNC, con el cual, el proceso de fabricación sería a la inversa, de un tocho o bloque de material, se procede a eliminar material hasta obtener las formas deseadas, se pueden ver que las ventajas son enormes, puesto que en una sola máquina de FDM se puede fabricar una pieza con un solo proceso imposible de confeccionar en una sola máquina de CNC, además del tiempo de fabricación en FDM es mínimo comparado con el tiempo requerido en CNC. También la industria del mecanizado CNC ha avanzado mucho en la última década permitiendo

hacer geometrías con un grado de dificultad alto (máquinas de 5 o más ejes). Pese a los avances en los procesos de arranque de material, todavía se requieren tediosos post-procesados obtener piezas complejas. La misma pieza compleja, puede imprimirse de una sola vez rellenando regiones planas mediante FDM. Además, en la ilustración 1.4 se puede ver que la curva de desarrollo de un producto mediante FDM es mucho menor y más plana que la de mecanizado CNC. Siendo desde el inicio el coste mucho menor para los prototipos iniciados mediante FDM.

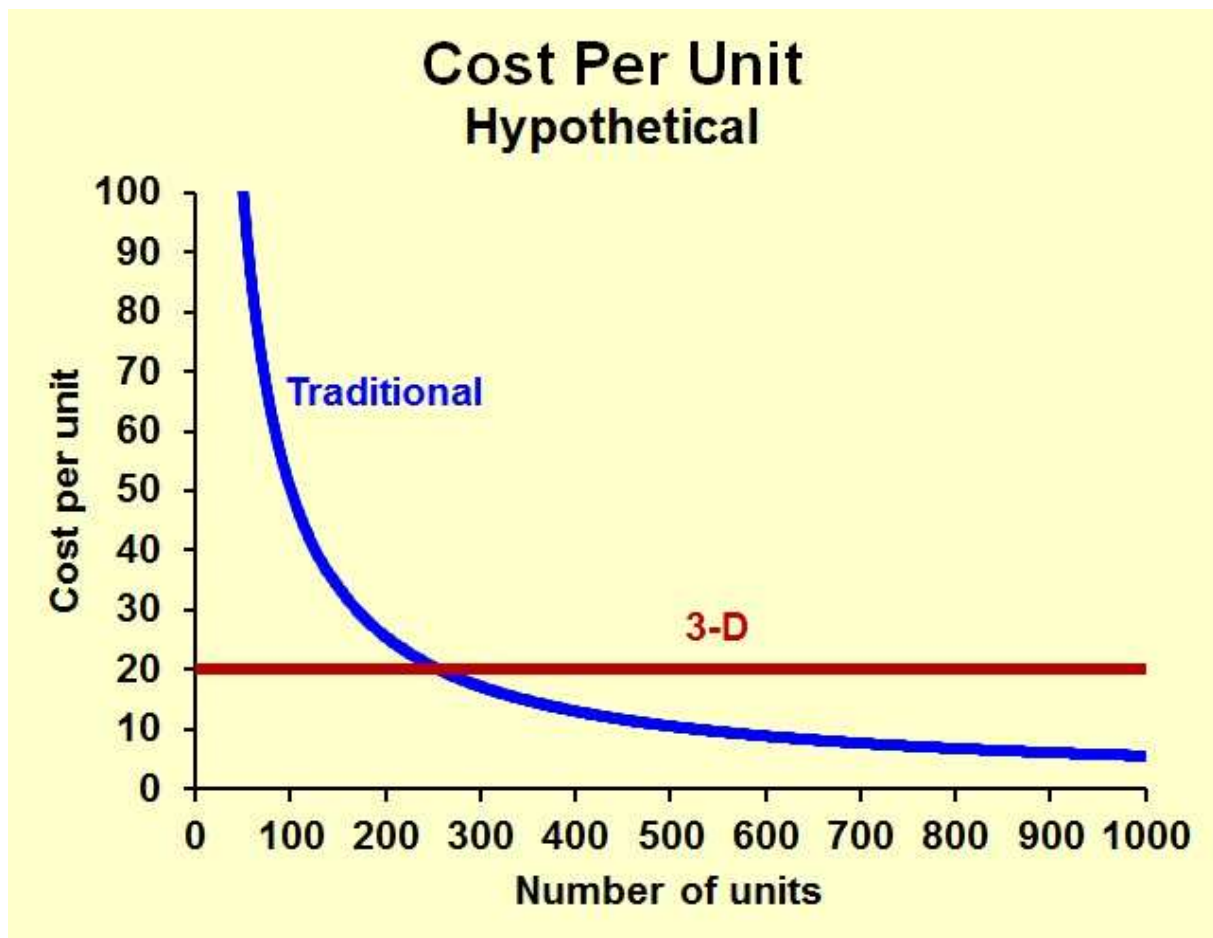


Ilustración 1.4 Curva de desarrollo de un producto mediante métodos tradicionales y FDM (Forbes, 2016)

El mayor inconveniente que reside en el FDM frente al mecanizado CNC es que a la hora de fabricar un prototipo, en el mecanizado CNC se puede utilizar el material del producto final, pudiendo realizar diversos ensayos sobre el prototipo que serían válidos para el producto final mientras que en FDM el campo de materiales es limitado lo que conlleva a que muchos de los prototipos realizados mediante FDM sólo sean modelos de forma o detalles no siendo válidos para ensayos de propiedades físicas.

En la ilustración 1.5 se puede apreciar la misma pieza fabricada mediante FDM y fabricada mediante mecanizado CNC



Ilustración 1.5 Prototipo fabricado mediante FDM (izquierda) y CNC (derecha) (UNI, 2016)

Como conclusiones del FDM se puede decir que es una tecnología barata al alcance de muchas personas, al contrario del mecanizado CNC que es muy caro. Además del enorme desarrollo que se ha producido de esta tecnología en tan solo 2 décadas y que no deja de avanzar a pasos agigantados. En los próximos años se establecerá como uno de los procesos de prototipado y fabricación en pequeñas series más utilizados. Además de la preparación de alimentos cocinados y órganos mediante el proceso de FDM. Cabe destacar también que la gran mayoría del software es libre, quiere decir que, cualquier persona puede utilizarlo y mejorarlo a su antojo además de poder fabricar mediante un proceso sencillo una máquina de FDM ya que los planos de bastantes máquinas de FDM también son de libre utilización.

1.3. Objetivo de este Trabajo Fin de Grado

Vistos los apartados anteriores, está claro que para obtener prototipos de forma rápida y a la vez con una buena calidad superficial se necesita, además de materiales de calidad, es decir, que tengan buena fluidez en el campo de fusión del mismo sin llegar a cristalizar, una selección de parámetros adecuados como el tamaño de la boquilla de la extrusora, velocidades de movimiento y extrusión y una altura de capa adecuada. Aun así, en la fabricación por FDM nunca se va a obtener un prototipo totalmente liso sin recurrir a post-procesados mecánicos o químicos. El objetivo de este Trabajo de Fin de Grado es el desarrollo de un sistema semi-automático que

permita una mejora controlada de la calidad superficial de piezas de revolución obtenidas a través de FDM mediante el pulverizado a presión de pintura de vinilo líquido obteniendo experimentalmente la curva de la mejora significativa o no de la calidad superficial mediante los parámetros más significativos que se puedan dar tras realizar los distintos experimentos. Todo el sistema se intentará realizar a partir de piezas con las cuales se construye una impresora 3D libre de FDM (piezas vitaminas llamadas en el ámbito FDM) además de piezas impresas mediante FDM en forma de agradecimiento a la comunidad (www.RepRap.org Ilustración 1.6) que hizo posible que el software y planos de las impresoras FDM sean Open Source.

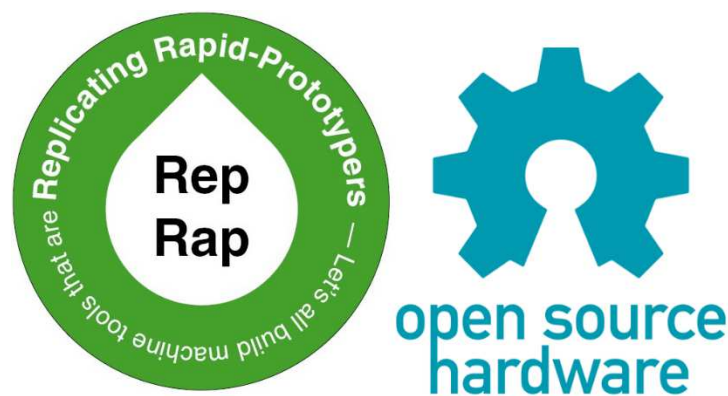


Ilustración 1.6 Logotipo de la comunidad RepRap (RepRap, 2016)

1.4. Estructura de este Trabajo Fin de Grado

Visto que tan importante es el acabado superficial de los prototipos obtenidos mediante FDM y de la necesidad de crear un sistema de tratado superficial semi-automático frente a los manuales existentes además de la prueba experimental de cómo afecta la pintura de vinilo líquido, se puede hacer una estructura clara de este Trabajo de Fin de Grado como sigue:

- Estudiar los distintos tipos de procesos de acabado superficial existentes e identificar su efectividad, coste y viabilidad.
- Planteamiento, diseño y desarrollo de la idea preconcebida de un sistema semi-automático de mejora de la calidad superficial de prototipos rápidos de revolución obtenidos mediante la tecnología de FDM.
- Impresión 3D de todas las piezas que sean posibles para la construcción del sistema de acabado diseñado anteriormente.

- Preparación y modificación de todos los parámetros del software que controlará el sistema.
- Ensamblaje del sistema.
- Realización de diversos ensayos para la validación del sistema además de la comprobación de la pintura de vinilo líquido para la mejora de la calidad superficial.
- Para concluir, se realizará un procesado y una discusión de los resultados obtenidos, llegando a unas conclusiones finales.
- Prueba del sistema obtenido en las piezas proporcionadas por el departamento de Prototipos de la empresa Valeo Iluminación S.A.U.

2. PROCESOS DE MEJORA DEL ACABADO SUPERFICIAL

Como se ha comentado en anteriores apartados, a menudo, es necesario el uso de tratamientos de post-procesado para poder obtener la geometría deseada o incluso para poder modificar las propiedades superficiales del prototipo. Generalmente, la superficie se controla por 3 motivos distintos:

- Reducir la fricción: si el prototipo incorpora partes móviles, necesitará reducir la fricción en los puntos que contengan movimiento.
- Controlar el desgaste: debido a que, si el prototipo va a estar sometido a grandes esfuerzos o fatiga, habrá que tratarlo superficialmente para mejorar sus propiedades de resistencia o dureza.
- Mejorar aspecto: cuando es necesario que la superficie parezca totalmente lisa se requieren tratamientos de mejora superficial, ya sean químicos, mecánicos o de otro tipo.

(Poveda Martínez, Santiago)

En la ilustración 2.1, se puede ver la mejora superficial de la figura impresa mediante el método de FDM con distintos tratamientos superficiales, de izquierda a derecha, mejora de acabado mediante vapor de acetona, mediante chorro de partículas abrasivas y el último de mejora de las propiedades del material mediante recristalización.



Ilustración 2.1 Distintos tratamientos superficiales en una pieza fabricada mediante FDM (Impresoras3d, 2016)

Como se observa en la ilustración, dependiendo del acabado superficial, se pueden obtener superficies mates, satinadas o brillantes, además de poder eliminar las capas visibles convirtiendo la figura en totalmente lisa.

A continuación, se va a entrar en detalle en cada uno de los tipos de tratamientos superficiales que existen hasta llegar al que se ha optado en este trabajo, la pintura de vinilo líquido.

2.1. Tratamientos químicos

Estos tratamientos son los que mayor popularidad poseen, ya que, en su gran mayoría consisten en exponer la pieza a la acción de un líquido abrasivo con el material de la pieza, ya sea por inmersión directa en el líquido o mediante la creación de una atmosfera de vapor del mismo son conocidos como tratamientos Smoothing (de suavizado). Estos tratamientos son peligrosos ya que, por lo general, estos líquidos son inflamables además de muy tóxicos para la salud. Con ese tipo de tratamientos, es muy fácil cambiar las propiedades físicas del prototipo si no se controlan con alta precisión. Además, se tiene que tener en cuenta que, los tratamientos químicos son con los que mejor calidad superficial se obtiene en cuanto a aspecto se refiere, por lo que a menudo, suelen ir precedidos de tratamientos distintos como los mecánicos.

2.1.1 Inmersión directa en el líquido abrasivo

Según el material utilizado a la hora de crear el prototipo, habrá que usar distintos líquidos que sean abrasivos con el mismo. En los plásticos más extendidos en el FDM,

es decir el PLA y el ABS, los productos químicos más utilizados son respectivamente el cloroformo (CHCl_3) y la acetona industrial (propanona pura $\text{CH}_3(\text{CO})\text{CH}_3$) respectivamente, aunque existe una amplia variedad de productos que producen un efecto similar en estos materiales, estos son los más fáciles de encontrar además de tener un precio bajo. Estos productos, se evaporan fácilmente y son inflamables por lo que manipular y utilizar estos productos puede producir un riesgo serio habiendo que tomar las precauciones oportunas para su uso.

En el caso del ABS, al sumergir la pieza en el líquido, hay que conocer sus efectos ya que el ABS, en contacto con la acetona, primero se ablanda hasta comenzar a disolverse de forma progresiva. Con este método es muy sencillo eliminar las muy conocidas capas de las piezas impresas a costa de perder las tolerancias de la pieza ya que, al ser un proceso abrasivo de la superficie, esta resulta lisa y brillante a costa de disolverse las capas más externas. Para controlar este proceso, se debe extraer la pieza sumergida en un determinado momento de tiempo y para evitar que el material disuelto se vuelva a solidificar sobre la pieza y que la acetona que impregna la superficie siga actuando, se recurre a una inmersión en agua destilada para limpiar todos los restos que puedan quedar tanto de material disuelto como de acetona.

Para poder controlar mejor este proceso, se recurre a la mezcla de la acetona con agua ya que el resultado final que se busca es suavizar toda la superficie además del acabado brillante que se obtiene a pesar de perder los detalles más pequeños sin llegar a sobrepasar el tiempo de inmersión ya que deformaría totalmente el prototipo quedando inservible.

En contrapartida, se obtiene que no todos los fabricantes aditivan igualmente sus materiales, dando lugar a diversos comportamientos del ABS, por lo que no a todos les afectará en igual modo la inmersión en acetona, por lo que este método resulta totalmente empírico sin posibilidad alguna de crear un estándar para poder llevarlo a todos los fabricantes de este material del mismo modo.

Otras formas de aplicación de la acetona líquida es mediante lo que se conoce como el ABS juice, esto es, una mezcla de acetona con una cantidad exacta de ABS, que tiene múltiples usos, los más conocidos son su uso a modo de barniz para obtener el acabado brillante, como para la adhesión de piezas aplicándolo igual que si se

trataría de un pegamento o cola y por último su uso para impregnar las hotbed de las impresoras para evitar uno de los defectos más comunes conocido como Warping, que ocurre cuando la pieza se despegue de la cama caliente debido a que se ha endurecido muy rápido por la superficie externa y las tensiones internas del material de su interior hacen que se deforme la pieza.

Para el caso del PLA, los productos químicos se utilizarían del mismo modo, obteniendo resultados similares. Además, la acetona se usa para el PLA en los casos que se necesita que adquiera cierto comportamiento gomoso, sin perder su forma original.

En la ilustración 2.2, se puede ver este efecto al producirse la inmersión de ABS en acetona las aristas del prisma resultan redondeadas además de suavizar todos los bordes de la pieza impresa.

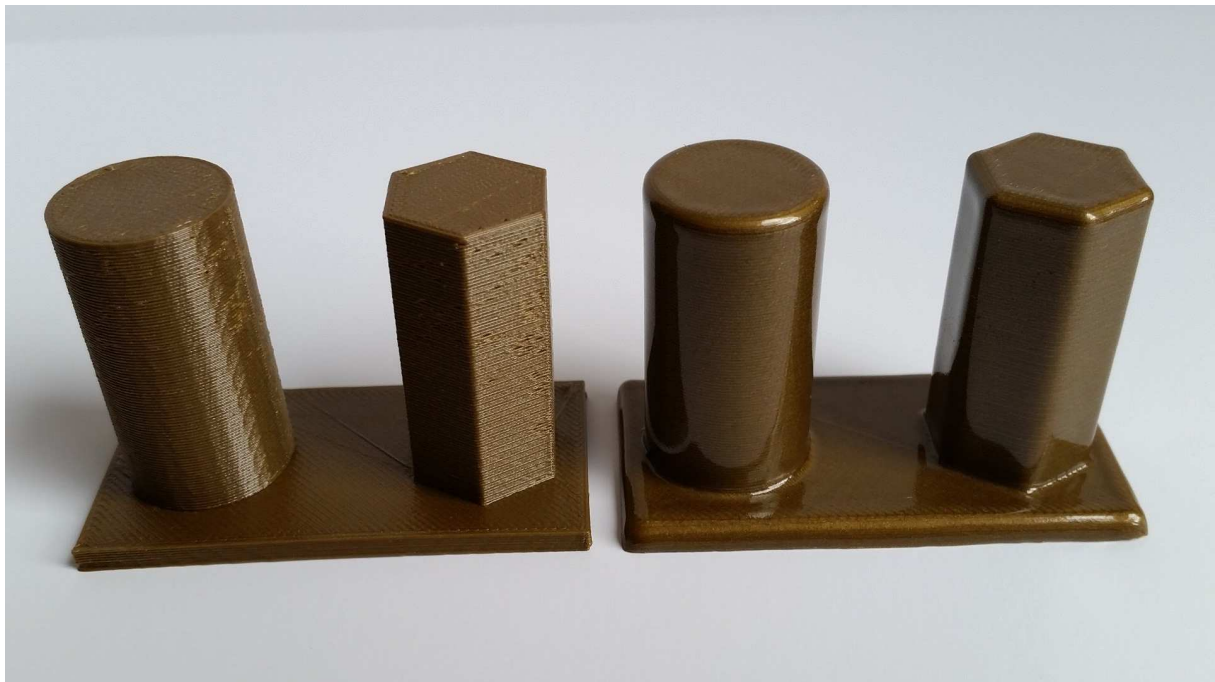


Ilustración 2.2 Pieza de ABS sin tratar (izquierda) y tratada con inmersión en acetona (derecha)
(3dwithus, 2016)

2.1.2 Inmersión en atmósfera de vapor disolvente

De forma análoga al apartado anterior, para estos procesos se utilizan los mismos materiales con sus respectivos productos abrasivos, sólo difiere el modo de aplicación y es que ahora se trata del método conocido como Smoothing o suavizado

con vapor, resultado de introducir la pieza en una atmosfera rica en vapor del producto disolvente, ya sea obteniendo ese vapor dejando que el disolvente se evapore o mediante el calentamiento progresivo del mismo, acelerando el proceso además de haciendo que se vuelva más inestable. Según si el vapor se consigue a temperatura ambiente o no, se tienen dos tipos de tratamiento, tratamientos en frio o en caliente. Aun utilizando los tratamientos en caliente, los efectos son mucho menos disolventes con el material que si se utilizan procesos de inmersión en el líquido. Esto lleva asociado a un aumento de los tiempos de tratamiento, al igual que es mucho más fácil controlar el proceso y obtener resultados aceptables con mayor simplicidad.

La principal desventaja de este tipo de tratamientos es que su aplicación tiene que ser dentro de recipientes o cámaras cerradas impidiendo que los gases nocivos y altamente inflamables escapen sin control. Como se ha dicho estos productos son muy nocivos para la salud y su efecto se multiplica al estar en estado gaseoso pudiendo formar atmosferas o nubes tóxicas extremadamente perjudiciales para la salud de las personas que se encuentren en el lugar.

En los últimos años, las compañías han empezado a crear cámaras que realizan este proceso de forma sencilla y controlada. El problema es el alto coste que esta tecnología tiene aún, haciéndolo casi inalcanzable para pequeñas empresas o uso doméstico.

La compañía taiwanesa Sky Tech presentó su máquina de Smoothing de bajo coste comparado con las máquinas de otras compañías, esta se muestra en la ilustración 2.3



Ilustración 2.3 Máquina de Smoothing de Sky Tech
(3dprintingindustry, 2016)

2.2. Tratamientos mecánicos

Este tipo de tratamientos, a menudo, se realizan antes de un tratamiento químico, pero no por ello siempre van precedidos de un segundo tratamiento. Con estos post-procesados de las piezas impresas mediante FDM, se trata de eliminar posibles rebabas, soportes de impresión o incluso suavizar la superficie de la pieza como se trataba de hacer con los anteriores. También se utilizan para ajustar las tolerancias cuando se requiere alta precisión como por ejemplo en ensamblajes que posean movimiento. Como desventaja de esta forma de post-procesado, se encuentra que siempre se va a tener que eliminar cierta cantidad de material, por lo que si se conoce de antemano que se necesita un post-procesado mecánico habrá que tenerlo en cuenta a la hora de crear el propio diseño en 3D con un ligero sobredimensionado de las zonas que se puedan ver afectadas.

La mayor ventaja frente a los tratamientos químicos es que no son perjudiciales para la salud de las personas por lo que las medidas de seguridad a la hora de manipular e interactuar en estos procesos resulta mucho más simple.

Dentro de estos procesos se engloban desde pulido con materiales abrasivos, lijados, mecanizados manuales o CNC e incluso utilizando equipos profesionales.

2.2.1 Procesos mecánicos manuales

Dentro de estos procesos más rudimentarios se encuentran procesos de corte de rebabas con cuchillas, suavizado con lijas o limas o incluso pulido manual con una multiherramienta rotativa como se puede ver en la ilustración 2.4 siempre trabajando a bajas velocidades para evitar calentar la superficie y que se deforme.

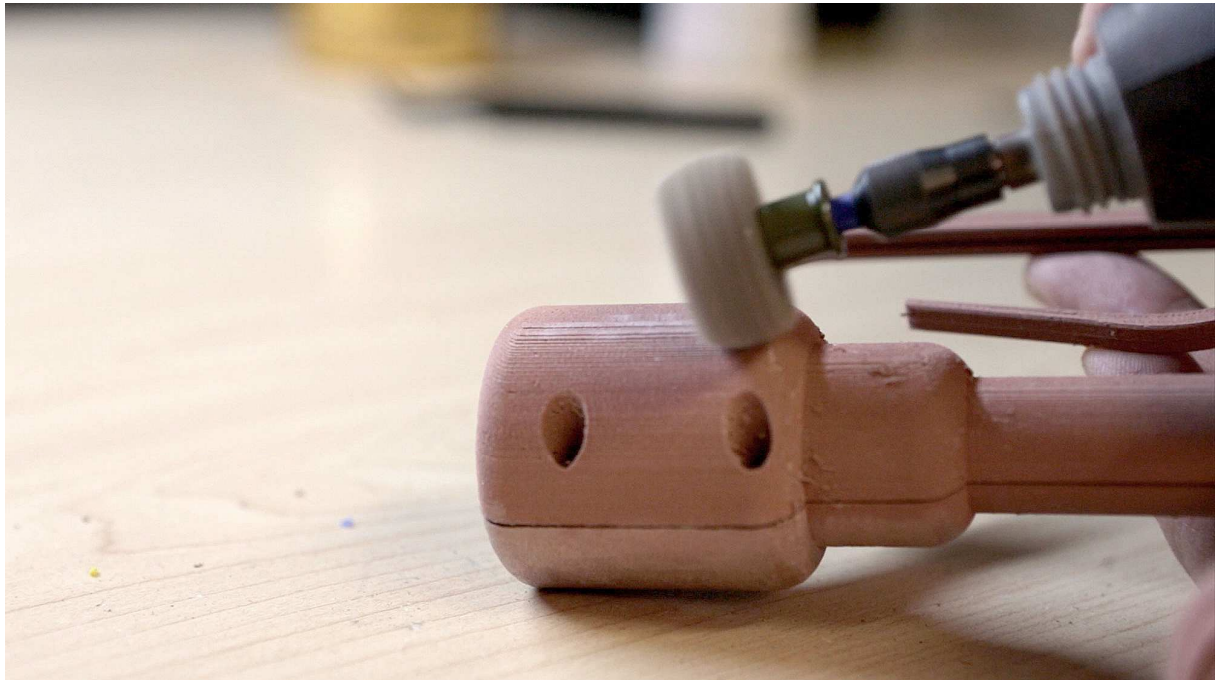


Ilustración 2.4 Acabado con multiherramienta rotativa (Adafruit, 2016)

La gran desventaja de los procesos manuales es que no dependen de parámetros regulados ya que dependen de la pericia que posea la persona que está realizando dicho proceso, pudiendo conseguir un resultado equiparable a una máquina si la experiencia es alta.

2.2.2 Procesos mecánicos utilizando equipos profesionales

Se pueden destacar el granallado, pulido de la superficie con abrasivos mediante el uso de vibraciones o incluso mecanizados CNC.

2.2.2.1 Granallado

Estos procesos se han transferido desde otros sectores adaptando únicamente el material que se dispara contra la pieza o incluso utilizando los mismos que se usan en otros sectores como la arena. Este proceso consiste en bombardear la pieza con pequeñas partículas de material mediante aire a presión, normalmente del mismo termoplástico del que esté fabricado la pieza o incluso de alguno de mayor dureza.

Cuando las partículas de material inciden sobre la pieza se produce una pequeña deformación plástica en la superficie eliminando el escalonado que producen las capas de las piezas impresas además de aportar una mayor robustez al prototipo al endurecerse la superficie. Cuanto más fino sea el material con el que se bombardea la superficie, mejor será el acabado superficial, esta diferencia se puede ver en la ilustración 2.5. También se utiliza para eliminar los soportes de impresión, hilos de material fundido que puedan haberse creado en el proceso de impresión o las posibles rebabas que hayan podido aparecer.

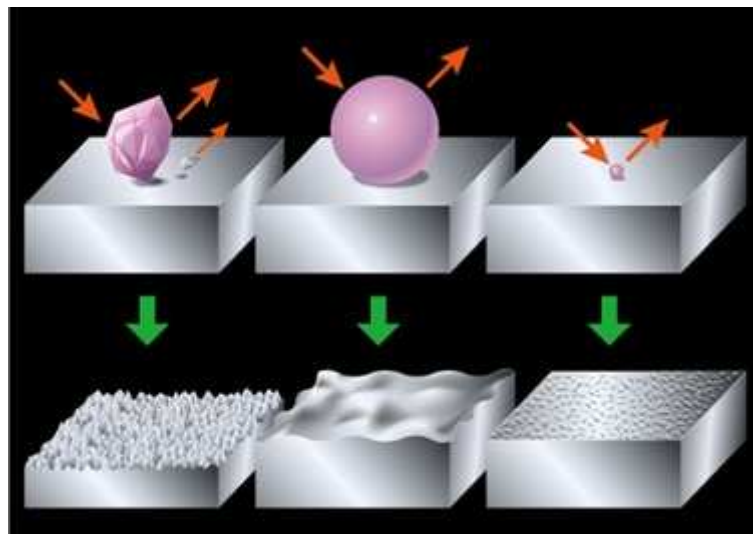


Ilustración 2.5 Diferencias de acabado según tipo y tamaño de material de granallado. (Frank day motors, 2016)

La ventaja de estos procesos es que además de ser rápidos, el acabado es brillante, similar al que se obtienen mediante procesos químicos, no contaminan el ambiente ni son perjudiciales para la salud, además de conservarse las tolerancias de impresión impuestas a las piezas.

Como se ha dicho, también es posible utilizar otros materiales como microperlas de vidrio o incluso arenas que posean distinta granularidad y abrasividad, aunque la desventaja de estos últimos, conocidos como arenados, es que la superficie final suele ser mate. En la ilustración 2.6, se muestra este proceso de arenado en un equipo de chorro de arena y que como se puede observar el granulado de la arena es tan fino que parece casi imperceptible al ojo humano.

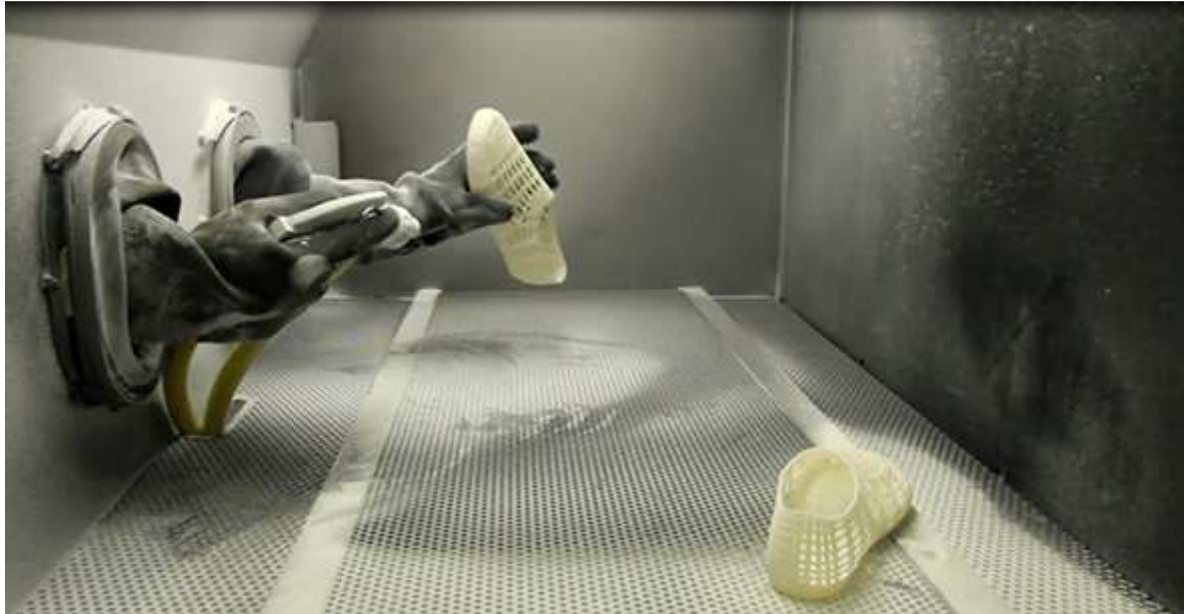


Ilustración 2.6 Equipo de chorro de arena (Machinedesign, 2016)

2.2.2.2 Pulido de superficie con abrasivos mediante el uso de vibraciones

Este tratamiento también es utilizado en otros sectores como por ejemplo en joyería. Consiste en introducir la pieza en un recipiente lleno de perlas de material abrasivo al cual se le transmite un movimiento de vibración que hace que el material de abrasión desgaste la pieza, consiguiendo el efecto parecido al del granallado. Dependiendo del tipo de material, se pueden obtener distintos resultados, desde abrillantado de la superficie a alisado de la superficie, los más utilizados son la piedra pómez, arenas de distinto granulado o pequeños trozos de termoplásticos además a veces, se suele añadir un poco de líquido para que absorba todo el polvo que se genere. Este sistema se puede ver en la ilustración 2.7 en la cual se tienen varias piezas impresas mediante FDM dentro del sistema de pulido por vibración.



Ilustración 2.7 Sistema de pulido por vibración (Gigaom, 2016)

La gran desventaja de este sistema frente al anterior es que toda la superficie se someterá al tratamiento mientras que, en el granallado, el tratamiento se puede focalizar en una sola superficie. Sin embargo, la gran ventaja es que no necesita de una cámara cerrada como en el granallado siendo más segura su aplicación además de limpio.

2.2.2.3 mecanizados CNC

Para finalizar los tratamientos mecánicos, hay que destacar los mecanizados CNC, como el fresado, torneado, taladrado entre otros. Para el caso de las piezas impresas en plástico, los mecanizados CNC se suelen hacer a bajas velocidades ya que pueden provocar que la pieza se caliente y comience a deformarse esta es la gran desventaja de estos tratamientos frente a otros. Para intentar frenar estos efectos, se utilizan sistemas sofisticados de refrigeración con taladrinas o aire a presión dependiendo del material que se vaya a mecanizar.

Generalmente se utilizan en los termoplásticos que tienen el punto de fusión más alto como el ABS o el NYLON ya que por ejemplo en el caso del PLA tiene un punto de fusión muy bajo y a bajas velocidades es posible que se deforme. La ventaja que tiene frente a otros es que se pueden por ejemplo rectificar orificios, si se requiere precisión por ejemplo en una deslizadera.

A continuación, en la ilustración 2.8, se tiene un fresado de unas piezas de NYLON a baja velocidad con un sistema de refrigeración de aire a presión para combatir el aumento de temperatura.

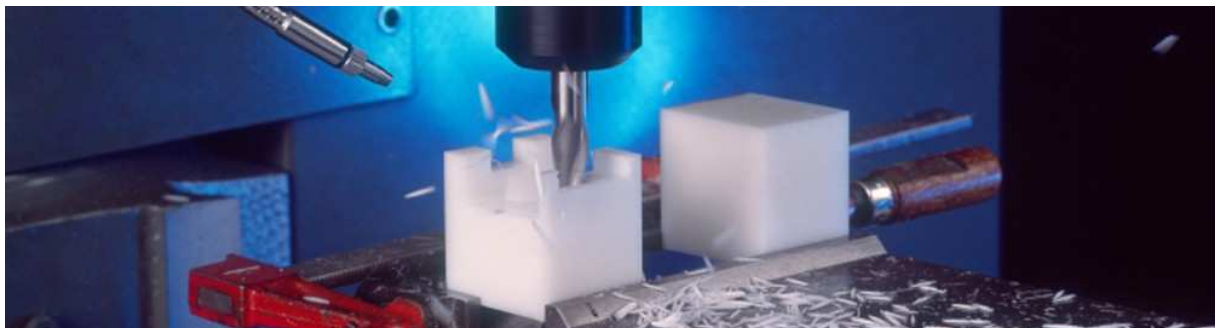


Ilustración 2.8 Fresado de pieza de NYLON fabricada mediante FDM (Plas-techfab, 2016)

2.3. Otros tipos de tratamientos superficiales

Dentro de la infinidad de tratamientos podemos encontrar otros menos utilizados como son:

- **Sistema de pintado de superficie con rotuladores** de colores mientras se imprime. Este sistema desarrollado por Kikai Labs va coloreando las capas conforme se depositan pudiendo crear imágenes dibujadas en las piezas impresas. Este método no mejora ni las propiedades ni suaviza la superficie, pero permite decorar piezas en distintos colores utilizando material de un sólo color.



Ilustración 2.9 Ejemplos del sistema desarrollado por Kikai Labs (Kikailabs, 2016)

- **Recubrimiento con resinas epóxicas.** Este tratamiento muy novedoso, permite impregnar la superficie de las piezas con una resina epóxica de doble compuesto. La más común es la XTC-3D, que suaviza la superficie de los escalonamientos de las capas y además permitiendo después de su aplicación lijados para obtener superficies totalmente lisas.



Ilustración 2.10 Recubrimiento de pieza impresa con resina epóxica (3dprinting, 2016)

- **Pintado de la superficie.** A menudo, este proceso se descarta considerándose que no se suele obtener un buen resultado, puesto que la pintura se escurre,

se forman burbujas en la superficie o incluso deforman la superficie. Sin embargo “Esto podría ser cierto si se utilizan ciertas pinturas que contengan productos químicos que se disuelven o alteran las propiedades del ABS o PLA. Sin embargo, esta afirmación no es cierta si se usan pinturas acrílicas en particular” (Bell, 2014)

2.4. Pintura de vinilo líquido

Este método es el elegido para el sistema que se intentará desarrollar en este trabajo, ya que no hay nada similar creado ni experimentado ya que la pintura de vinilo líquido es un producto novedoso que apareció en el mercado entre 2014 y 2015. La característica que lo hace tan peculiar es que se aplica como si de pintura se tratara, pero logrando que, tras su secado de varias capas, se puede despegar como si de vinilo adhesivo se tratara, manteniendo la superficie sin dañar su color o aspecto original. Es un producto que, a pesar de aplicarse como pintura, es capaz de agarrar bien en multitud de superficies y no escurrirse como ocurre en muchas pinturas, además de no dañar la superficie original ya que no contiene productos disolventes. Además, existe una amplia gama de colores con diferentes acabados como los mates, perlados o incluso brillantes. Dependiendo del número de capas de pintura de vinilo que se aplique, se obtendrán mejores resultados de suavizado en la superficie. Esta nueva pintura, es muy utilizada en el sector de la automoción ya que permite cambiar el color de los vehículos, permitiendo poder quitarlo en un futuro conservando la pintura original. En detalle se puede ver esta pintura de vinilo líquido en la ilustración 2.11:



Ilustración 2.11 Pintura de vinilo líquido (Autozona, 2016)

Consiguiendo un acabado bueno, este tratamiento superficial podría ser de gran utilidad dentro del mercado frente a los tratamientos superficiales ya conocidos como los químicos o mecánicos.

Se tratará de crear un sistema de acabado superficial de piezas de revolución semi-automático que rocíe este producto sobre las piezas impresas, sin necesidad de la actuación humana en el proceso, obteniendo una mejora de la calidad superficial de piezas fabricadas mediante FDM.

3. DESARROLLO DEL SISTEMA

3.1. Diseño del sistema

Una vez conocidos los múltiples métodos existentes, hay que centrarse en crear el sistema deseado que actúe con vinilo líquido. En primer lugar, se hará un “Brainstorming” sobre las distintas posibilidades y se crearán los distintos bocetos y se estudiará su viabilidad para su desarrollo posterior, partiendo de que las piezas serían de revolución.

3.1.1. Bocetos

3.1.1.1 Sistema adaptado a la impresora 3D

Para comenzar, la primera idea fue adaptar el sistema a la impresora 3D que había en el Departamento, RepRap BCN 3D que se muestra en la ilustración 3.1.1, el sistema trataría de un acople en forma de “C” al cual se le ajustarían unas boquillas de pintura en spray para que, tras la finalización de la impresión, el sistema rociaría la pieza impresa con vinilo líquido obteniéndose el resultado buscado.

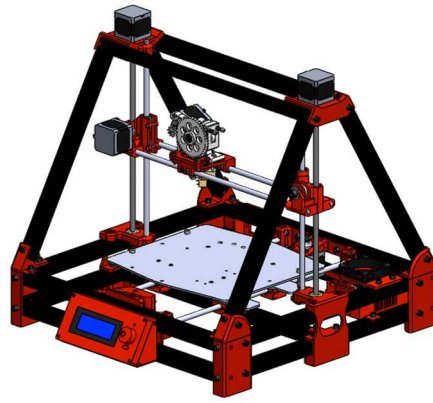


Ilustración 3.1.1 Impresora RepRap BCN 3D (Bcn3dtechnologies, 2016)

Tras preconcebir la idea inicial, se diseñó el boceto para estudiar más en detalle su factibilidad. Este se muestra en la ilustración 3.1.2.

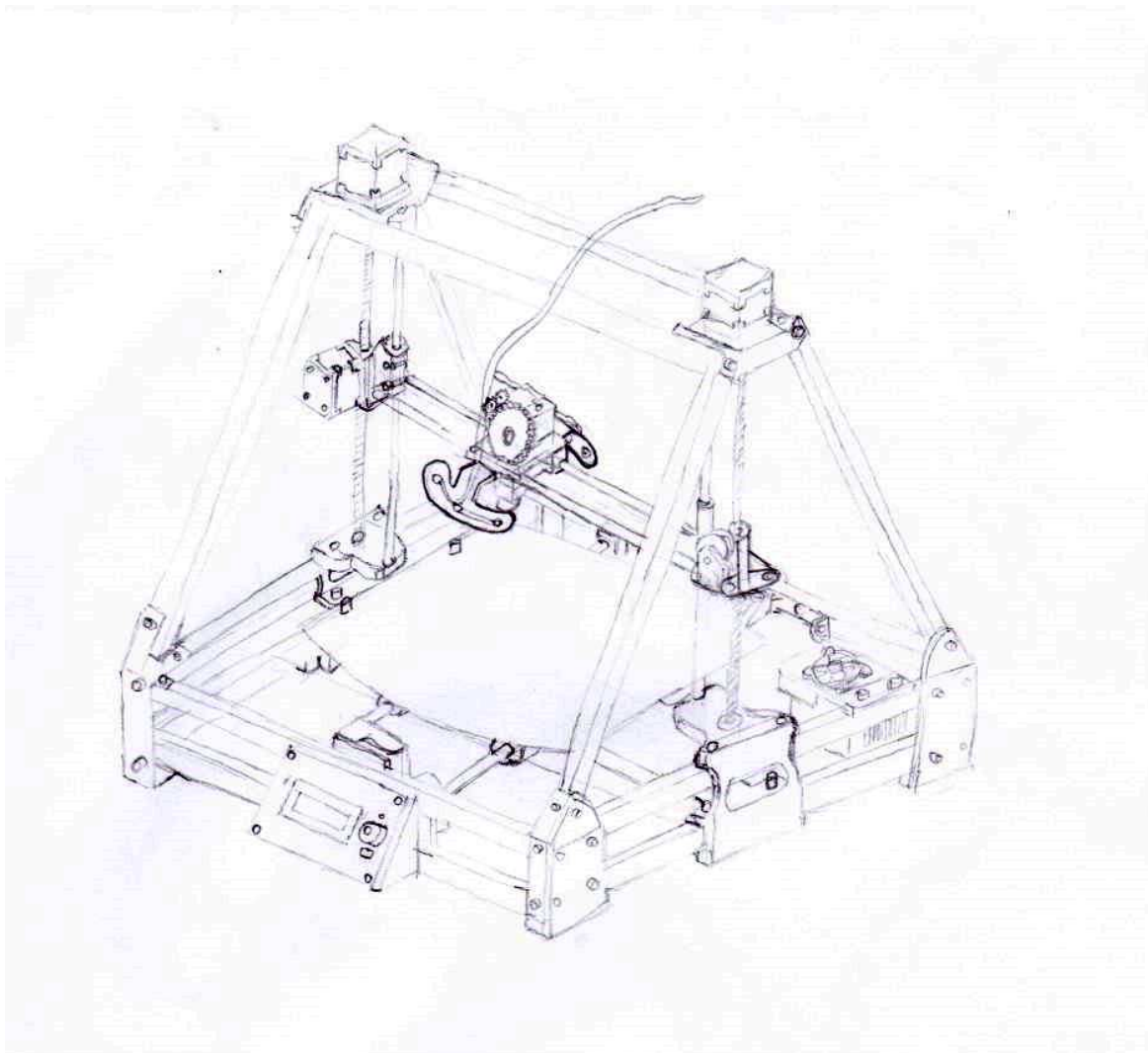


Ilustración 3.1.2 Primer boceto del sistema de pintura adaptado a la impresora 3D

Se puede ver como el sistema se sitúa en el carro del eje X junto al extrusor. Finalmente, este diseño se descartó porque sería difícil evitar manchar la impresora 3D. La cama necesita la máxima limpieza en cada impresión, y también su sistema mecánico por lo que se tomó la decisión de descartar esta idea y crear un sistema externo a la impresora, utilizando para ello componentes que se usan en las impresoras 3D como es su electrónica o sistemas de movimiento mediante motores paso a paso.

3.1.1.2 Sistemas externos a la impresora 3D.

- a) Sistema de doble boquilla y cama fija: Para este sistema, se necesitarían 2 motores paso a paso, que accionarían los husillos, desplazando verticalmente una boquilla a cada lado de la pieza, recubriéndola de vinilo. La cama sería estática. La gran desventaja de este sistema, es que la pieza se mantiene estática durante el proceso, pudiendo ser que los laterales tangentes a las boquillas de pintura se quedaran con menor cobertura. Este sistema se muestra en la ilustración 3.1.3. Se trataría de uno de los sistemas más sencillos, ya que solo habría movimiento en un eje, controlando la altura mediante el G-Code y el punto de partida mediante un final de carrera mecánico. La pintura se accionaría mediante un sistema de aire a presión externo.

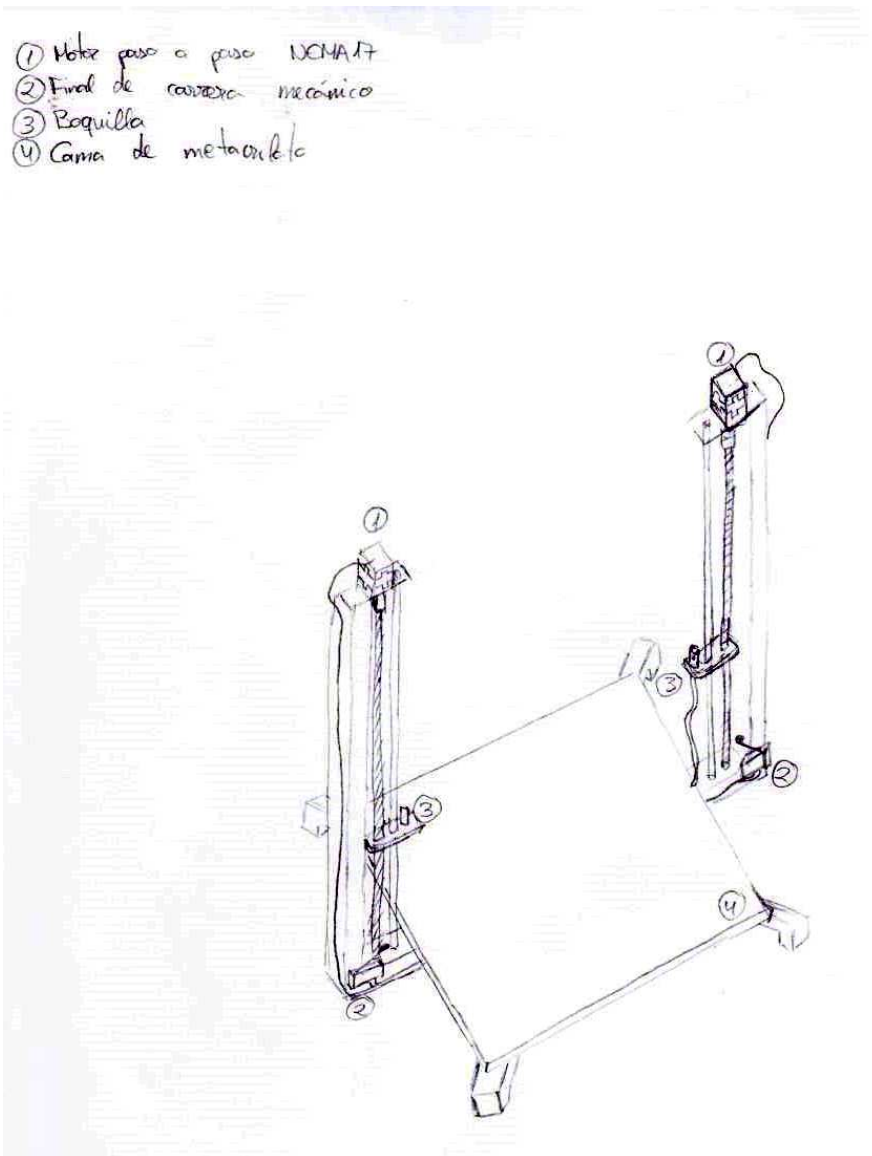


Ilustración 3.1.3 Sistema de doble boquilla y cama fija

- b) Sistema de boquilla móvil y cama giratoria: Este segundo sistema es totalmente lo contrario al anterior. En esta ocasión, se ha confeccionado un sistema con una cama circular y giratoria mediante un motor paso a paso el cual a su vez posee un pilar similar al que tenía la anterior con un motor paso a paso que mueve un pequeño carro que transporta la boquilla a lo largo del eje Z mediante una correa dentada con una polea en el extremo contrario al motor. En la ilustración 3.1.4 se puede ver cómo quedaría. Además, en este caso, se solucionaría el problema del anterior ya que, al girar la plataforma, la pieza sería pintada uniformemente por todo su perímetro externo. El coste sería similar al anterior ya que usa también 2 motores paso a paso además del sistema externo de aire

comprimido para accionar la pintura. Al igual que el anterior, el punto de inicio se controlaría mediante un final de carrera mecánico mientras que el movimiento de la plataforma y del carro de la boquilla se haría mediante el G-Code.

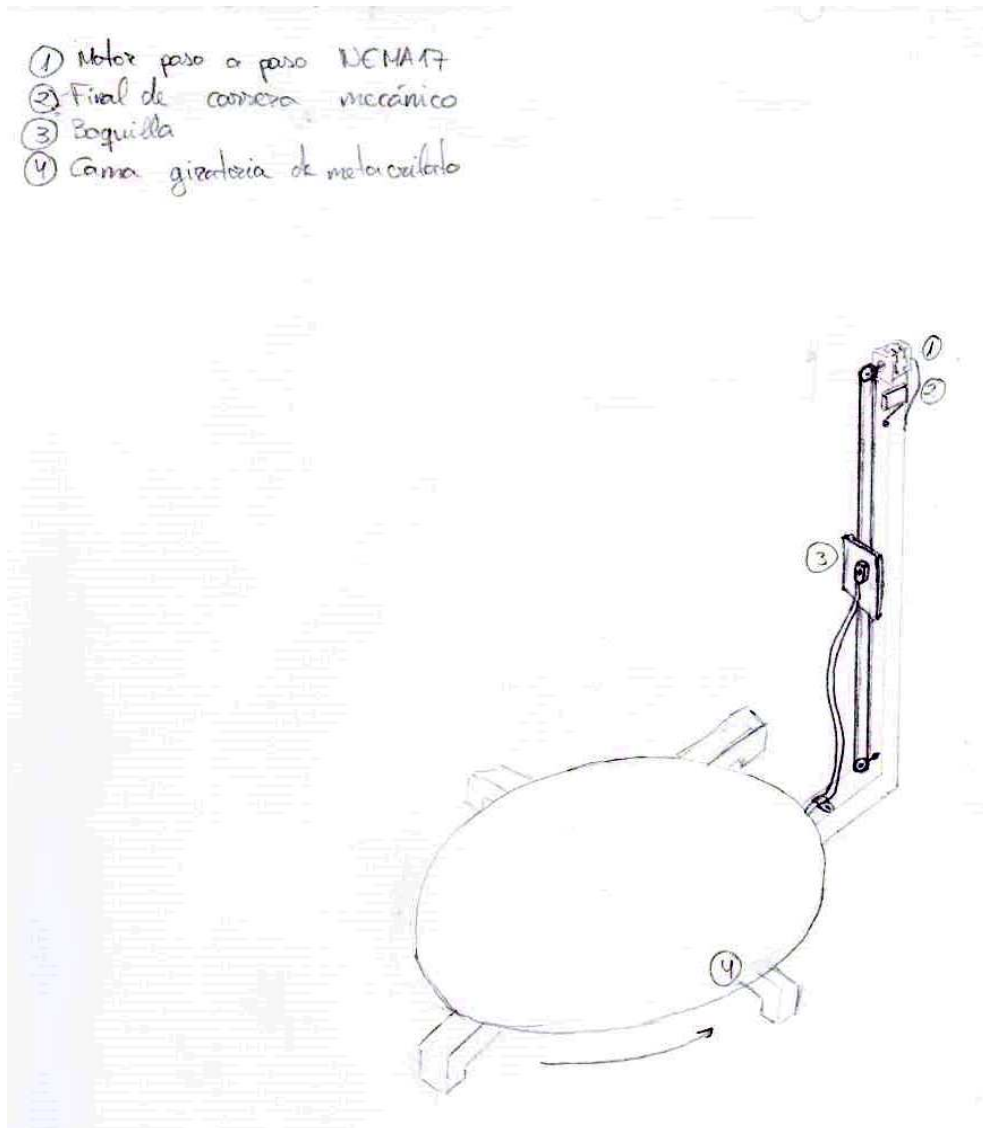


Ilustración 3.1.4 Sistema de boquilla móvil y cama giratoria

- Sistema de boquilla fija y cama giratoria: Finalmente, se consiguió crear un boceto de un sistema para el cual solo sería necesario un solo motor paso a paso. En este caso, se han utilizado ambos puntos fuertes de los sistemas anteriores, obteniendo un buen resultado. El motor paso a paso accionaría un husillo conectado a la plataforma que, además de conseguir el movimiento giratorio de la pieza, proporciona el movimiento de

elevación en el eje Z. En cuanto a la pintura, iría sujeta a una varilla metálica en un lateral del sistema, consiguiendo de esta manera pintar el perímetro de la pieza en su totalidad como se puede observar en la ilustración 3.1.5.

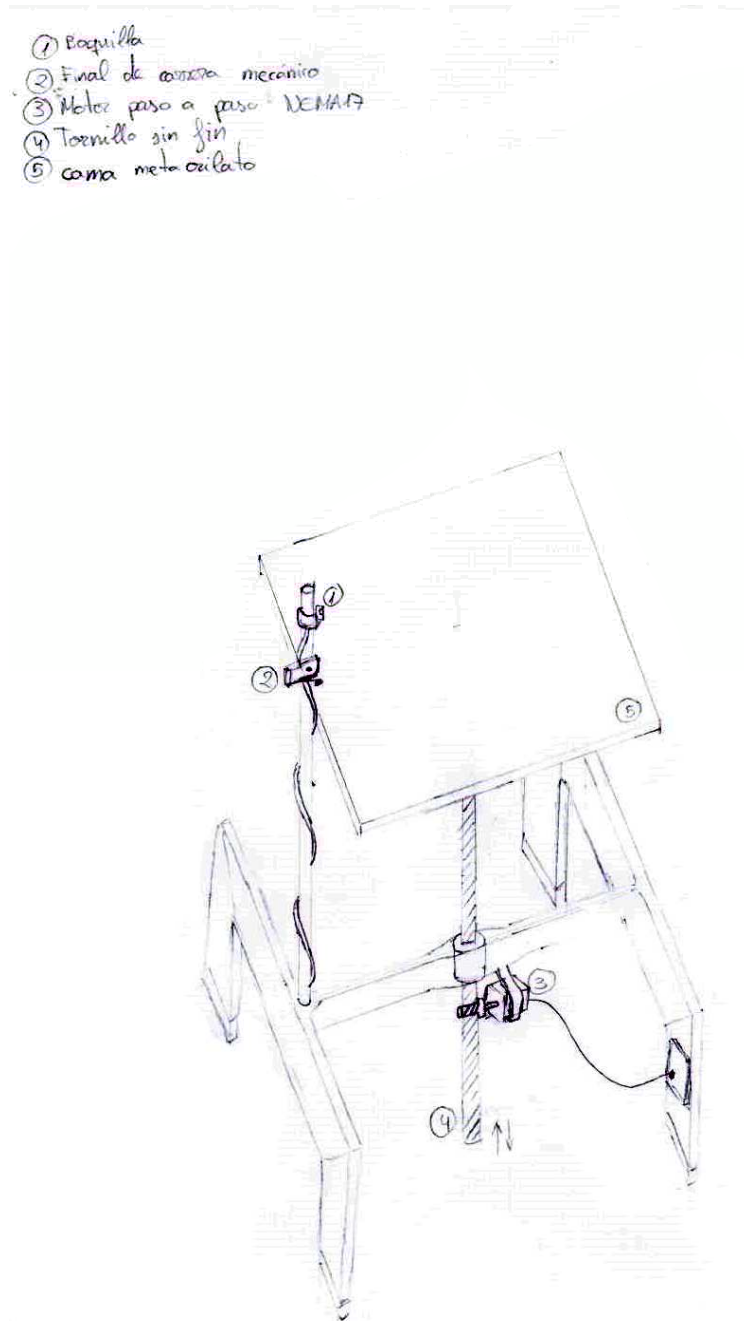


Ilustración 3.1.5 Sistema de boquilla fija y cama giratoria

Finalmente, tras decidir que este era el boceto a desarrollar y mejorar sobre esta idea, se descubrieron pequeños errores, fáciles de solucionar como por ejemplo; la

cama debía ser circular para que las esquinas de la misma no molestaran además de poder tener una superficie más grande, otro fallo sería la posición del motor, ya que el movimiento prediseñado en el boceto, sería un movimiento husillo-engranaje siendo este imposible tal y como se ha dibujado, ya que el motor lo tendría el engranaje y no el husillo como realmente funciona este sistema, por lo que posteriormente se diseña una mejora para subsanar este problema. Como último problema obtenido en todos los bocetos, es el sistema de pintura, ya que dependeríamos de un compresor externo con el que dar presión a la pintura a través de un pequeño depósito. Para ello, se opta por adaptar pintura ya envasada en botes de spray y diseñar una sujeción para el bote a la varilla, evitando de esta forma tener que rediseñar por completo el sistema del boceto ya que la boquilla seguiría estando en el mismo lugar.

3.1.2. Diseño en software CATIA

Hechos los bocetos y conociendo la idea de la máquina a desarrollar, se va a diseñar dicha máquina con el software de modelado 3D CATIA y los conocimientos adquiridos en la asignatura “Técnicas de ingeniería gráfica aplicadas a ingeniería mecánica” del Grado de Ingeniería Mecánica de la Universidad de Jaén, intentando a su vez, utilizar modelos ya creados en la web www.Thingiverse.com ya que esta web es una comunidad libre en la cual las personas crean sus diseños y los comparten de forma gratuita, ampliando los horizontes de la gran comunidad opensource de la impresión 3D.

Para tener una base sobre las dimensiones, se escogen las dimensiones que más se usan en las piezas de la impresora 3D ya que se intentará usar el máximo que se puedan. Además de lo anterior, para la estructura, se aprovechará una estructura de perfiles BOSCH ya creada que se tenía en el departamento ajustándola a las nuevas medidas con el fin de reutilizar en la mayor medida posible, elementos que ya haya en el departamento o en el taller mecánico.

Para el chasis o bancada de la máquina se van a utilizar perfiles BOSCH de 45x45 mm cuya longitud es de 370 mm, además de 4 patas que apoyarán el chasis en la estructura de 100 mm de longitud, además al travesaño central, se le hará el taladro por el cual pasará el husillo, obteniendo el resultado que se muestra en la ilustración 3.1.6.

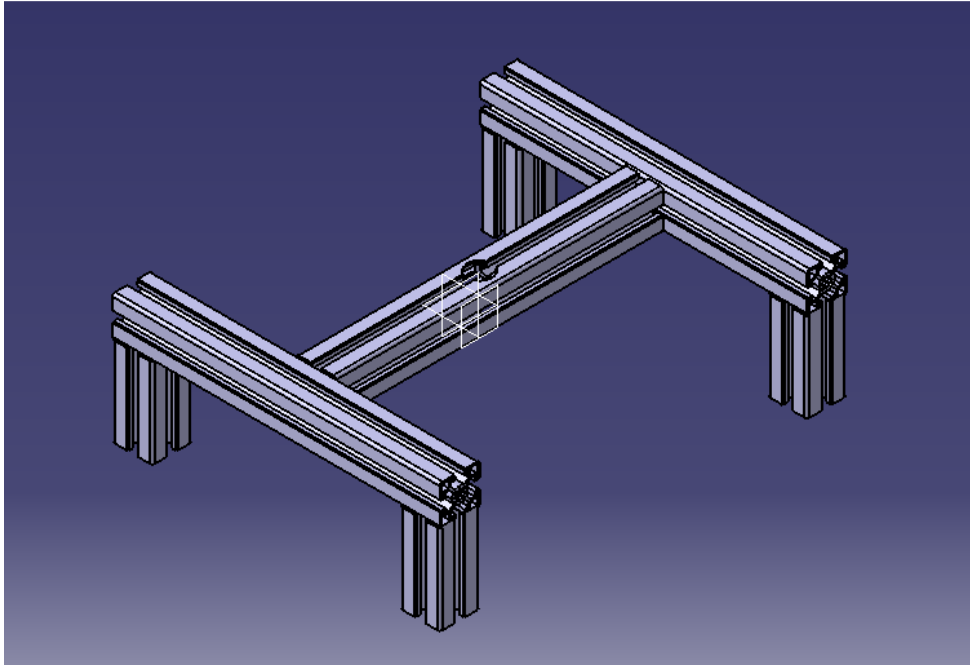


Ilustración 3.1.6 Chasis o bancada

A continuación, se diseñarán los distintos elementos que compondrán la máquina, como son:

- Cama circular de 220 mm de diámetro. Ilustración 3.1.6.a
- Soporte para el motor. Ilustración 3.1.6.b
- Varillas lisas. Ilustración 3.1.6.c
- Husillo de 8 mm de diámetro y paso 2 mm. Ilustración 3.1.6.d
- Acople del eje Z. Ilustración 3.1.6.e
- Sujeción de la varilla roscada a la cama. Ilustración 3.1.6.f
- Embellecedor varillas deslizaderas. Ilustración 3.1.6.g
- Accionador servo-spray. Ilustración 3.1.6.h.1/ Ilustración 3.1.6.h.2
- Soporte LCD. Ilustración 3.1.6.i
- Acople para las guías de los perfiles. Ilustración 3.1.6.j
- Soporte para el bote de spray. Ilustración 3.1.6.k.1/ Ilustración 3.1.6.k.2
- Soporte para servomotor. Ilustración 3.1.6.l
- Guía fijación para varillas. Ilustración 3.1.6.m

Las demás piezas utilizadas, es decir, la carcasa para el LCD y el soporte del final de carrera del eje Z han sido obtenidas de la página web Thingiverse y las piezas

comerciales, como tornillos, tuercas, motor NEMA, o rodamientos han sido descargadas de la página <https://github.com/>

A continuación, se muestra el diseño de cada una de las piezas diseñadas en el software CATIA además del correspondiente ensamblaje de la máquina.

A la plataforma circular, se le realizarán 4 orificios para poder sujetarla posteriormente al acople que la conectará con el husillo.

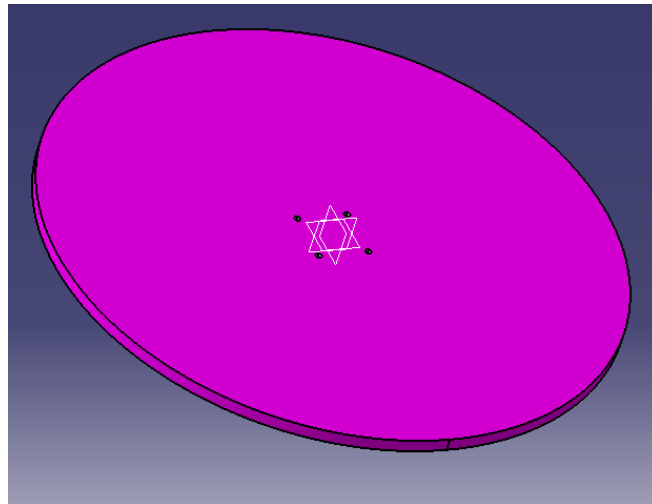


Ilustración 3.1.6.a Cama circular de 220 mm de diámetro

El soporte para el motor NEMA, además servirá para sujetar los dos rodamientos LM8UU con el fin de que el motor se desplace en este soporte a lo largo del eje Z mientras que el husillo gira en su tuerca y los rodamientos deslizan sobre las varillas lisas a modo de guías. Además, se obtendrá mediante FDM.

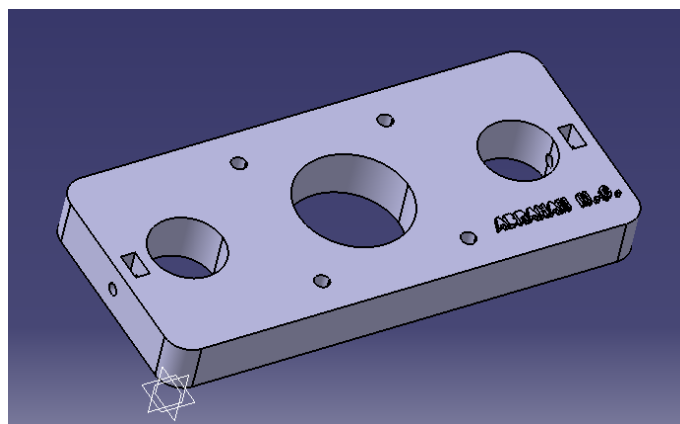


Ilustración 3.1.6.b Soporte para el motor

Las varillas lisas, se comprarán prefabricadas ya que sería casi imposible hacer mecanizar una barra de acero y dejarla en 8 mm de diámetro y con una longitud de mínimo 30 mm. En cuanto al husillo, este se comprará para evitar problemas con la rosca.

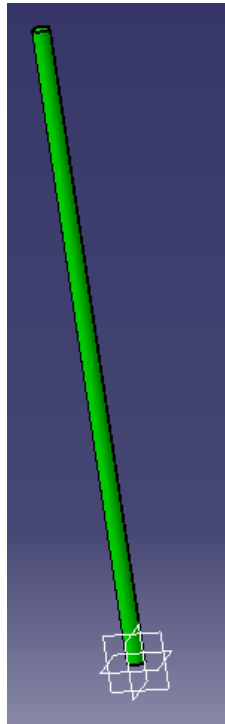


Ilustración 3.1.6.c Varillas lisas



Ilustración 3.1.6.d Husillo de 8 mm de diámetro y paso 2 mm

El acople del eje Z se utilizará el que se utiliza en impresoras FDM tipo Prusa evitando de esta forma tener que fabricar una pieza en aluminio tan pequeña. La sujeción de la varilla roscada a la cama o plataforma se unirá mediante tornillos de métrica 3, creando la rosca en la propia cama de metacrilato con un macho de roscar además se obtendrá mediante FDM.

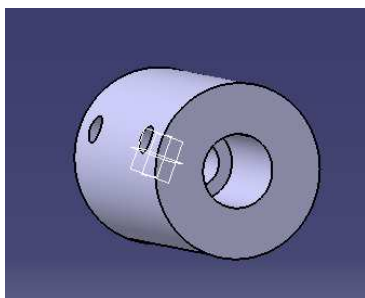


Ilustración 3.1.6.e Acople del eje Z.

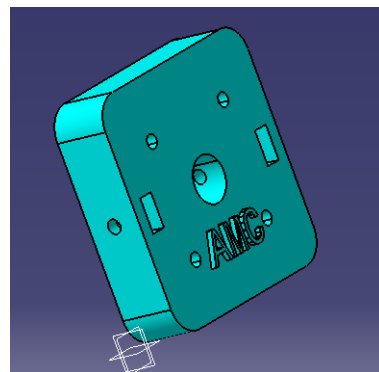


Ilustración 3.1.6.f Sujeción de la varilla roscada a la cama.

El embellecedor de las varillas lisas, tendrá como función evitar que las varillas lisas se desplacen haciendo que el deslizamiento no sea suave. Se sujetarán las varillas mediante una tuerca en el espacio creado para estas y un tornillo que oprima la varilla contra la pared del embellecedor, fijando su posición. Esta y el resto de piezas que restan, se crearán mediante impresión 3D.

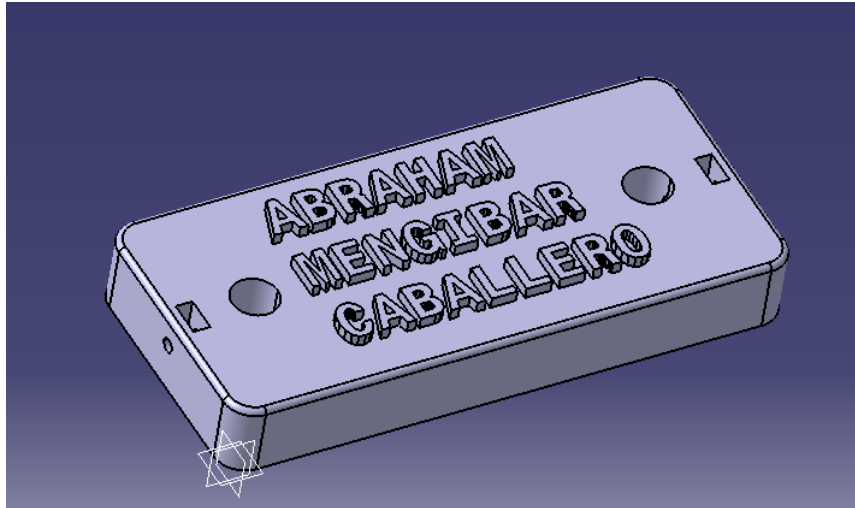


Ilustración 3.1.6.g Embellecedor varillas deslizaderas

Para el servo, el sistema creado ha sido un sistema de palanca de primer grado. Para esto, el extremo corto irá colocado sobre la palanca giratoria del servomotor mientras que el soporte se ha diseñado acorde a las medidas del bote de pintura de vinilo líquido, ya que este soporte encajará en la parte superior del bote de spray permitiendo que, al girar el servomotor, se genere la suficiente presión en la boquilla como para permitir la salida de la pintura a través de ésta. Ambas piezas irán unidas mediante una varilla de 3 mm.

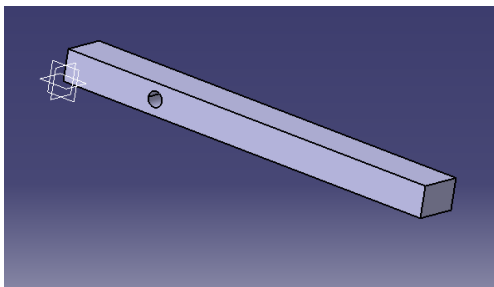


Ilustración 3.1.6.h.1 Accionador servo-spray.

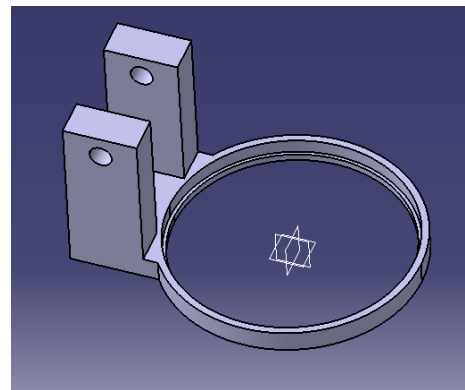


Ilustración 3.1.6.h.1 Accionador servo-spray

En cuanto al soporte para el LCD, se ha diseñado con una inclinación necesaria para permitir un visionado y ajuste cómodo del LCD y sus menús. El acople para las guías de los perfiles, se ha diseñado en función de las medidas del carril que tienen los perfiles BOSH de 45x45 mm ya que serán necesarios para sujetar los distintos elementos a los perfiles.

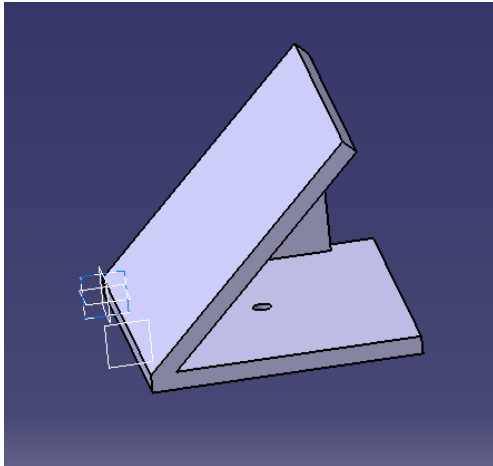


Ilustración 3.1.6.i Soporte LCD.

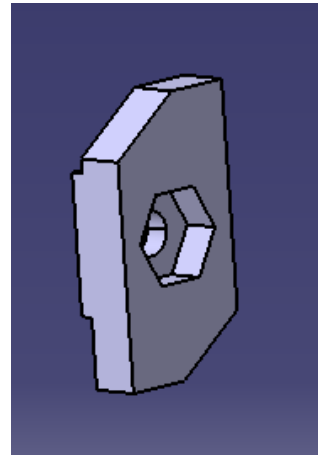


Ilustración 3.1.6.j Acople para las guías de los perfiles.

A continuación, se muestran las dos piezas del soporte para el bote de spray, que además sirve de fijación a la varilla lisa para regular su altura. En un principio, se ha diseñado en dos piezas para permitir variar el ángulo de inclinación de la boquilla. Además en la segunda ilustración, se puede ver como la sujeción es mediante un soporte uña, permitiendo ajustar la apertura de modo que se puedan usar botes de otros fabricantes que varíen las medidas de sus envases en un pequeño rango.

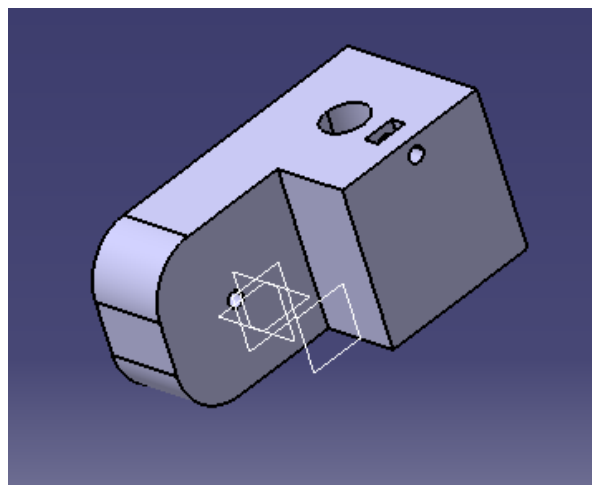


Ilustración 3.1.6.k.1 Soporte para el bote de spray

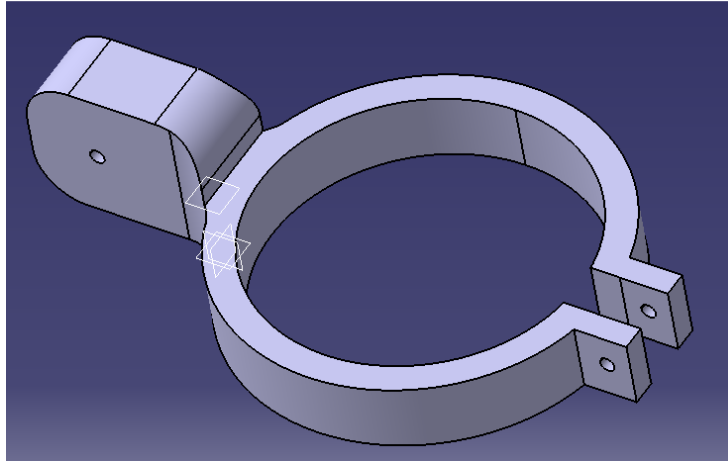


Ilustración 3.1.6.k.2 Soporte para el bote de spray

Debajo se tiene el soporte para el servomotor, este se ha desarrollado acorde a las medidas del servomotor a utilizar (MG996R del fabricante Tower-Pro) pudiendo anclarlo al soporte mediante los tornillos de fijación que el fabricante suministra con el propio servomotor.

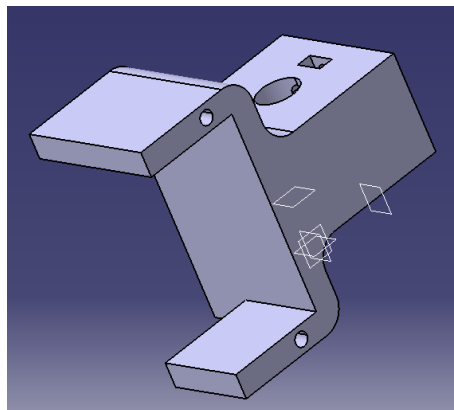


Ilustración 3.1.6.l Soporte para servomotor

Por último, la última pieza diseñada ha sido la guía para la fijación de las varillas, esta guía tiene la función de dar rigidez a las mismas, evitar que la base gire sobre su eje longitudinal además de anclarla mediante dos tornillos de métrica 3 a la bancada.

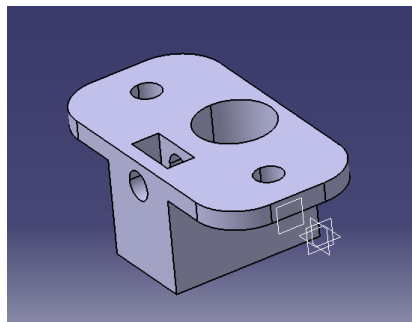


Ilustración 3.1.6.m Guía fijación para varillas.

Finalmente, utilizando además las piezas comerciales obtenidas de la web, como son la tuerca del husillo, los rodamientos, el motor NEMA y los tornillos y tuercas (Ilustración 3.1.7), se puede obtener el ensamblaje del sistema desarrollado. La bancada se montó sobre la mesa que se tenía en el departamento, adaptando la mesa a las medidas de la bancada.



Ilustración 3.1.7 Diseños 3D obtenidos de la página web <https://github.com/>

Como se puede ver en la Ilustración 3.1.8, el sistema desarrollado sería bastante compacto, pudiendo tenerlo en una habitación con espacio limitado, o junto a una impresora 3D permitiendo realizar el proceso de mejora de acabado superficial al finalizar la impresión, sin olvidar que se necesitará proteger los alrededores con un plástico para evitar la contaminación de pintura de los alrededores.

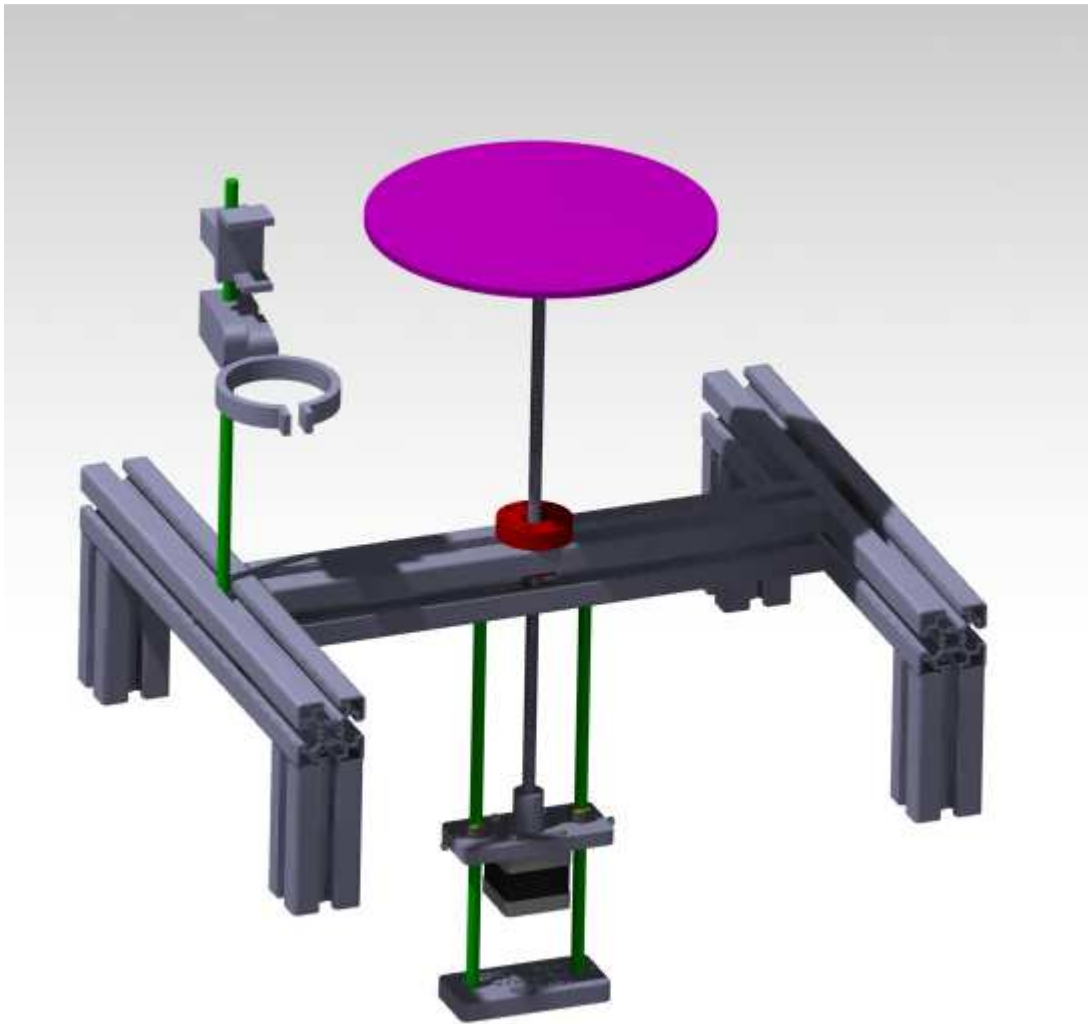


Ilustración 3.1.8 Ensamblaje del sistema.

3.1.3. Archivos descargados de la página Thingiverse

La página web www.Thingiverse.com se trata de una comunidad opensource en la cual sus usuarios comparten modelos creados por ellos mismos de forma gratuita facilitando a la comunidad de la impresión 3D obtener modelos 3D de piezas bastante comunes entre las utilizadas a la hora de la impresión 3D como accesorios para los componentes que constituyen las impresoras.

De este modo, se ha evitado diseñar desde cero, la carcasa para el LCD, la caja de la electrónica y el soporte para el final de carrera del eje Z. En esta página, por lo general, no se encuentran los archivos en el formato que se quiera, sino que, por lo general se encuentran en formato de impresión 3D conocido como STL, permitiendo insertar el mismo en el programa apropiado para crear el G-Code.

En la ilustración 3.1.9 se muestran las dos partes de la carcasa del LCD que se va a utilizar, además del soporte para sujetar el final de carrera del eje Z a una de las varillas lisas del mismo eje y en la ilustración 3.1.10 se tiene la caja de la electrónica

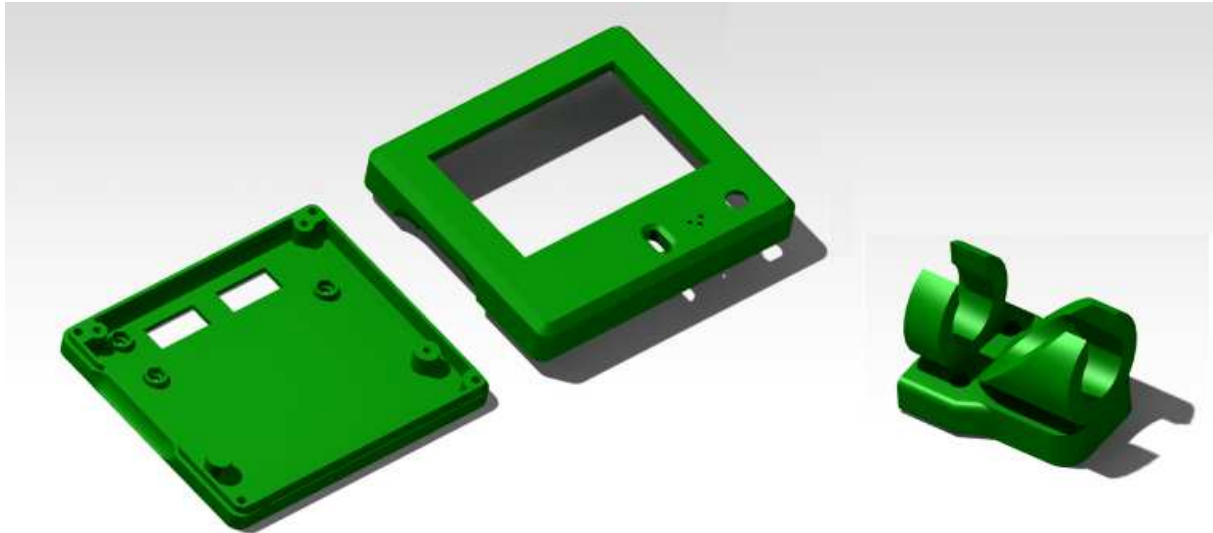


Ilustración 3.1.9 Piezas obtenidas de la página web www.Thingiverse.com a la izquierda la carcasa del LCD en dos piezas, a la derecha el soporte del final de carrera del eje Z

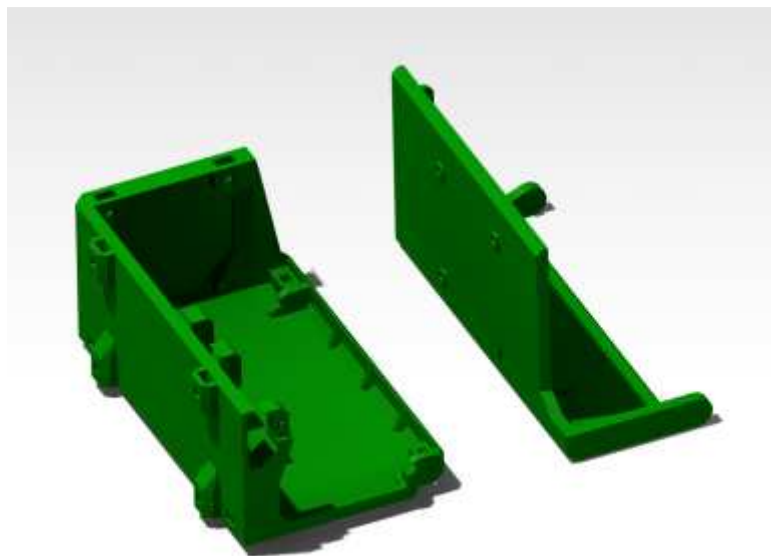


Ilustración 3.1.10 Piezas obtenidas de la página web www.Thingiverse.com

3.2. Impresión 3D de las piezas de la máquina

Ahora que ya está todo el sistema desarrollado, se tienen que preparar los archivos CAD para convertirlos al formato conveniente para poder adecuarlos al programa que crea el G-Code necesario para la impresión.

3.2.1. Conversión en formato STL de los archivos CAD

Como se ha dicho, es necesario convertir los archivos de CATIA con extensión “.CATpart” a STL, para ello primero se debe de configurar el software CATIA para obtener la máxima resolución en los archivos en formato STL. Para ello, dentro de la interfaz de CATIA, se seleccionará de la barra de menús la opción Tools y a continuación, se entrará en Options dentro del desplegable, tal y como se muestra en la ilustración 3.2.1.

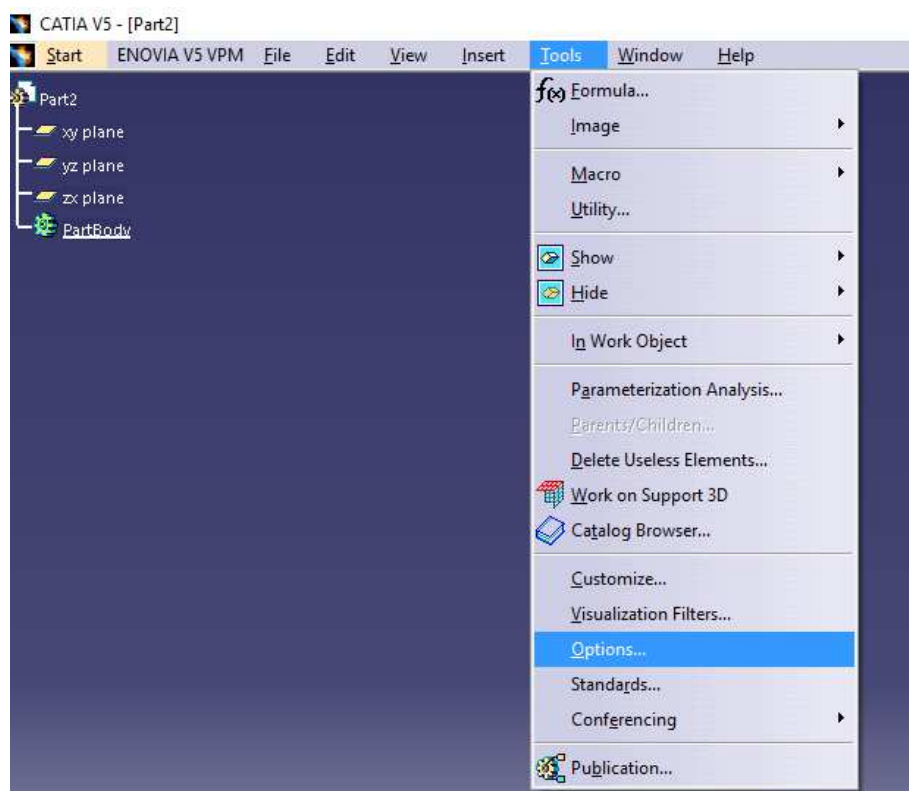


Ilustración 3.2.1 Menú Tools desplegado

A continuación, dentro de la nueva ventana que se ha abierto, se buscará en el menú de la izquierda la opción General, y desplegando esta, se debe hacer clic en Display. Abriendo la pestaña superior Performance, habrá que cambiar la opción 3D Accuracy, para esto se selecciona la Fixed y se introduce el valor mínimo que admite, siendo este 0,01, pudiendo ver la diferencia entre la opción preestablecida y la nueva

en la ilustración 3.2.2 donde se ve como en un principio el círculo era un polígono y tras el cambio, es mucho más preciso.

Si no realizamos este proceso previo a la exportación a formato STL, los resultados que obtendremos tras realizar la impresión serán de baja calidad ya que no se imprimirán curvas sino rectas, haciendo los círculos polígonos.

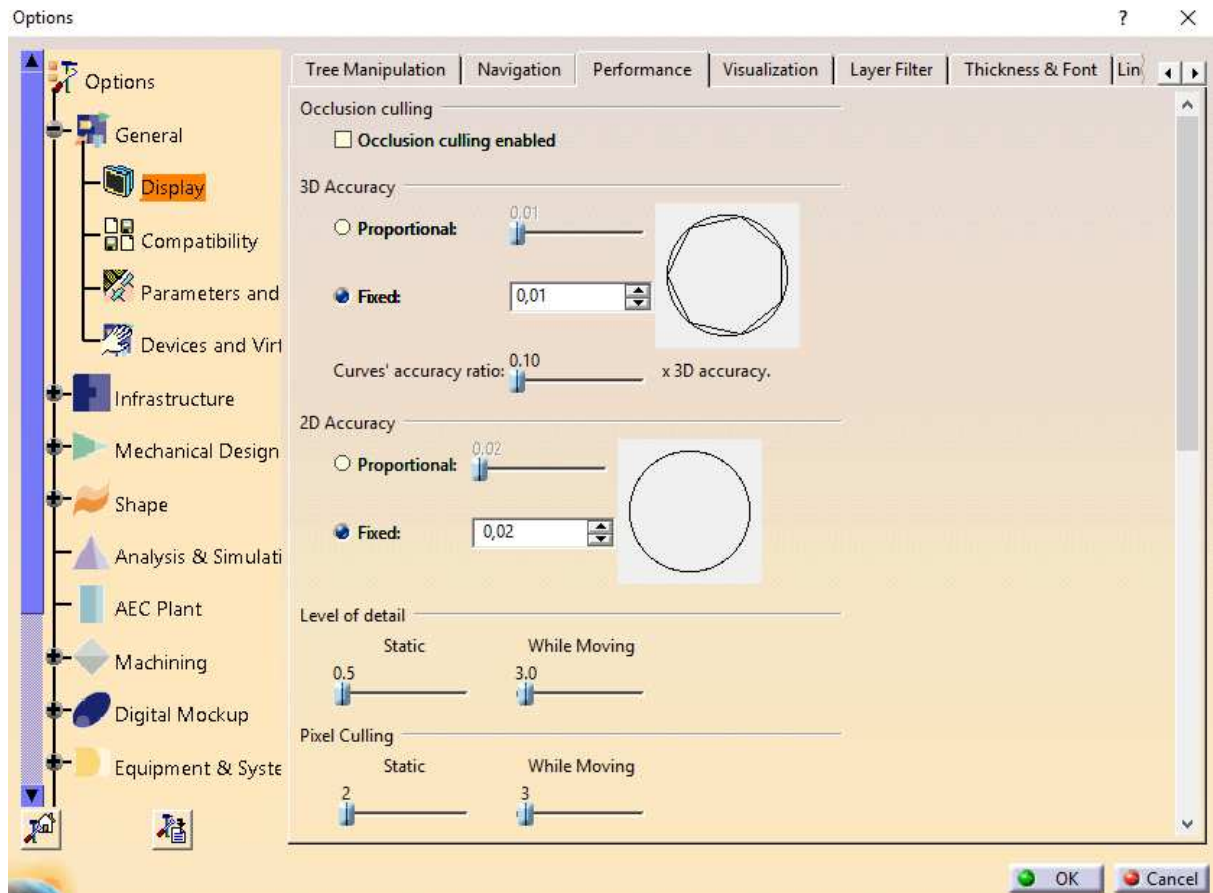


Ilustración 3.2.2 Pestaña Performance del menú Display

Una vez cambiado el valor aceptamos los cambios.

Para exportar el archivo a STL, los pasos a seguir son: clic en el File de la barra de menús y a continuación en Save As (Ilustración 3.2.3)

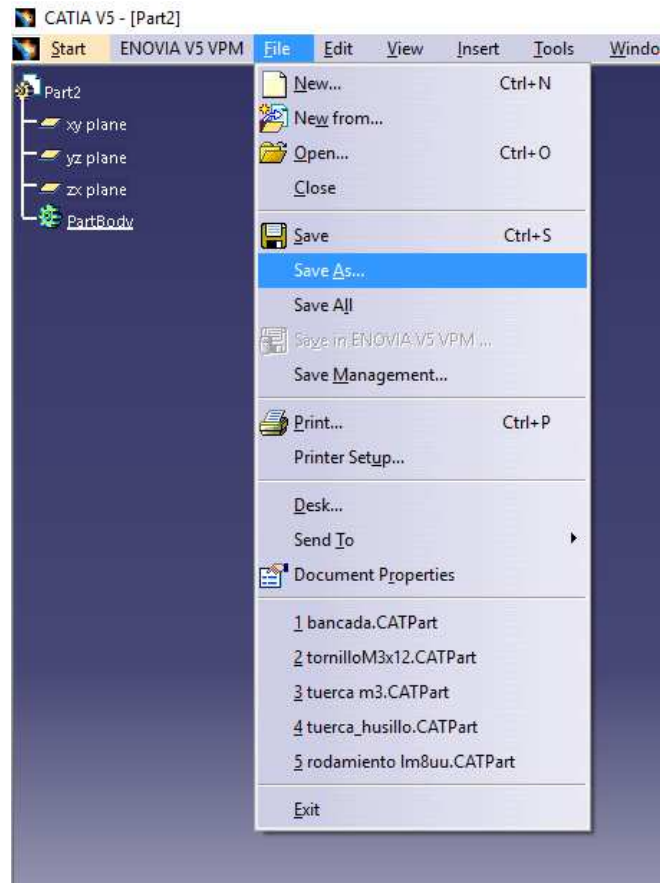


Ilustración 3.2.3 Menú File desplegado

Una vez en la ventana Save As, hay que abrir el desplegable Tipo: y seleccionar stl. Por último, se hace clic en guardar y listo, de esta forma tan sencilla, tendremos nuestros diseños en 3D en la máxima resolución que el programa puede aportar. Es muy importante tener en cuenta primero seleccionar la precisión de los círculos puesto que, sin esta precisión, se arruinaría todo el proceso venidero obligando a volver a ajustar la precisión, crear un nuevo G-Code y volver a fabricar la pieza en cuestión.

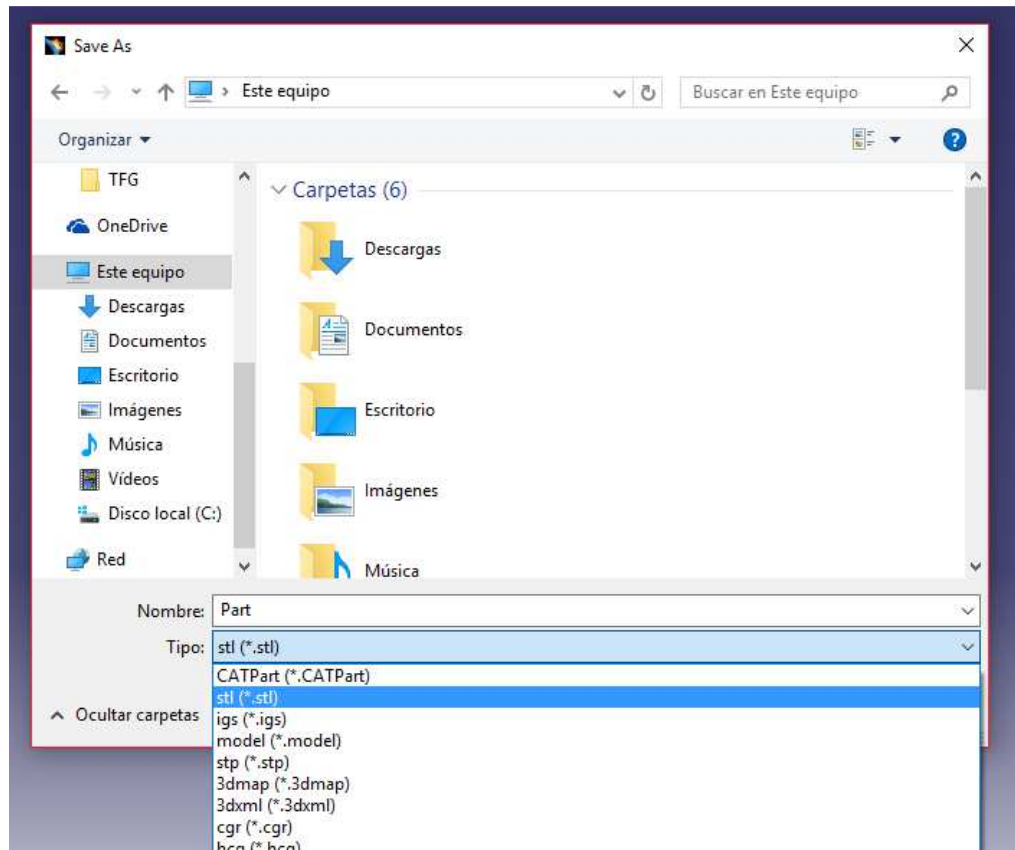


Ilustración 3.2.4 Ventana Save As y selección de la extensión del archivo

Obtenidos los archivos STL, se dejará a un lado todo el proceso en el programa de diseño y se pasará al programa creador del fichero G-Code.

3.2.2. Parámetros de impresión en el Software Repetier-Host

Esta vez se va a utilizar el software Repetier-Host, aunque existen diversos como Cura o Pronterface. Antes de generar los G-Code de cada pieza a fabricar, se deben configurar diversos parámetros que afectarán a la impresión como son las velocidades o temperaturas de impresión, así como la altura de capas o la densidad del relleno (evitando hacer piezas macizas para conseguir un ahorro de material).

Para configurar las impresiones, se ha de hacer clic en la pestaña Slicer que es el subprograma que secciona las piezas en múltiples capas para después, generar todos los movimientos que la impresora debe realizar para obtener la pieza. Una vez en esta pestaña, haciendo clic en Configuration se abre una nueva ventana en la cual se puede modificar toda la configuración de la impresión como se ve en la ilustración 2.3.5

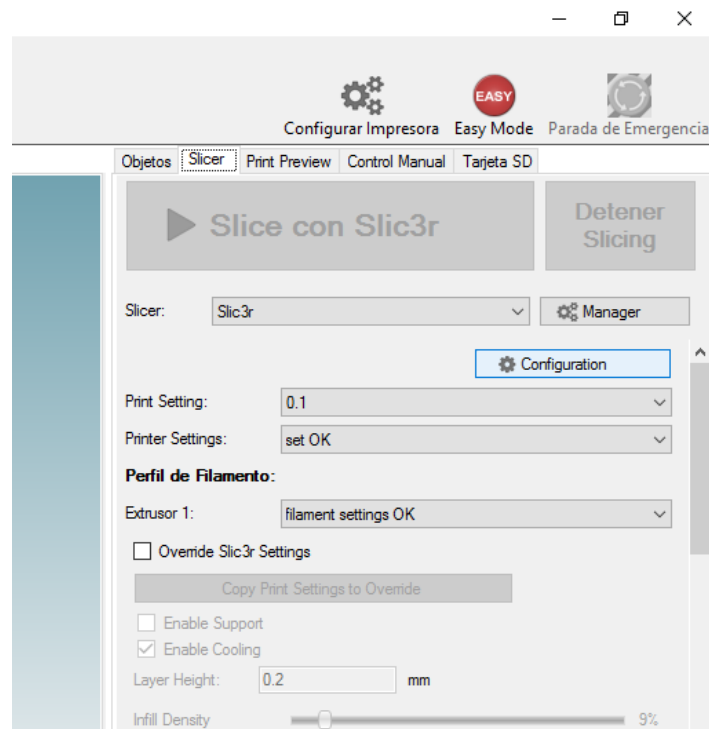


Ilustración 2.3.5 Pestaña Slicer dentro del programa Repetier-Host

En la primera pestaña (Print Setting), se selecciona Layers and Perimeters y se introduce 0.1 en Layer height, esto significa que tanto la primera capa como las restantes, tendrán una altura de 0.1 mm, siendo esta la máxima resolución que tiene la impresora que se va a utilizar para imprimir las piezas. En Vertical y en Horizontal Shells, introducimos 3, obligando a generar un grosor de 3 capas tanto en las tapas como en los laterales de la pieza, es decir 0.3 mm.

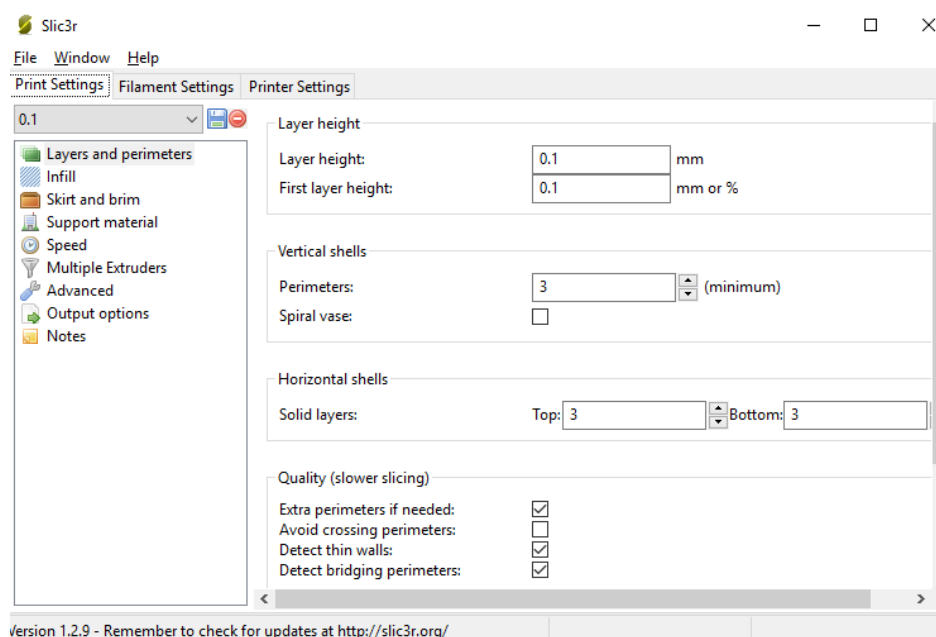


Ilustración 2.3.6 Ventana configuración Slic3r

A continuación, haciendo clic en la siguiente opción del menú vertical es decir Infill introducimos una densidad del 20% y un patrón de relleno tipo panal de abeja (Honeycomb) aportando este una gran resistencia a una baja densidad quedando como se ve en la ilustración 2.3.7.

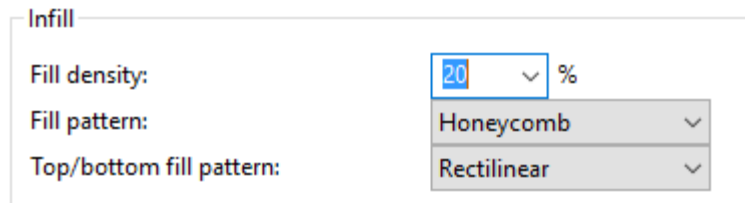


Ilustración 2.3.7 Parámetros de relleno

Seleccionando en el menú vertical Speed, se han de definir las opciones más importantes de la impresión que son las velocidades, la ilustración 2.3.8 muestra las velocidades utilizadas para cada parámetro, siendo velocidades bajas para los perímetros y aún más baja para los perímetros externos y más visuales y en cambio, siendo mucho más altas para los rellenos donde no se necesita tanta calidad.

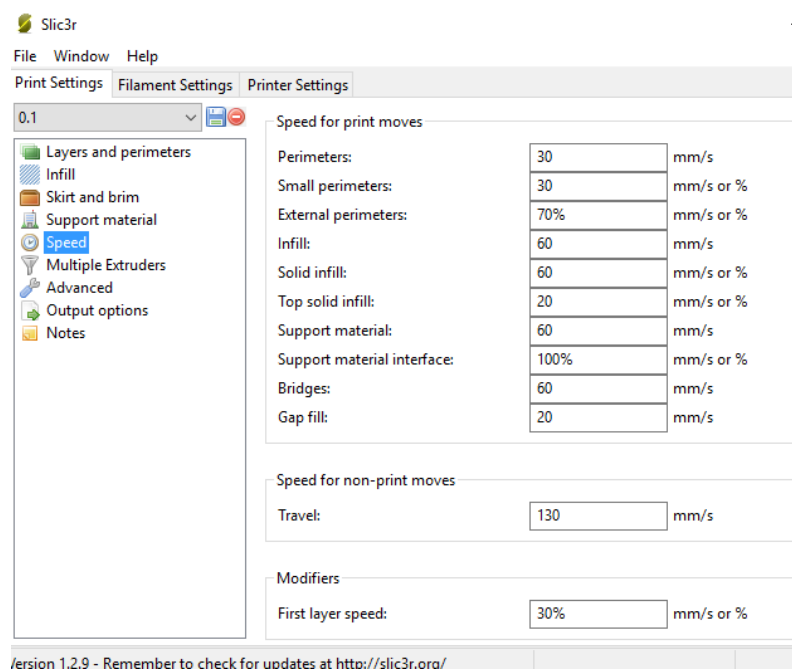
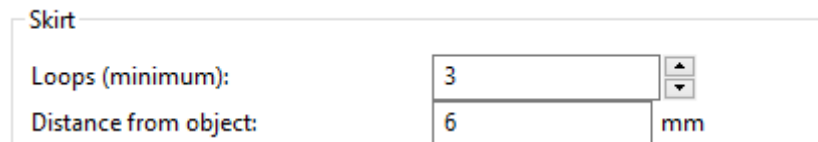


Ilustración 2.3.8 Parámetros de velocidad

Las opciones restantes, se dejarán las predefinidas por el programa, es decir, en la opción Skirt, deberá aparece una cantidad de 3 Loops, consiguiendo con esto que la extrusora imprima a una distancia predefinida de la pieza antes de comenzar limpiándose los restos que pudieran existir en la boquilla de la extrusora (Ilustración 2.3.9).



Skirt

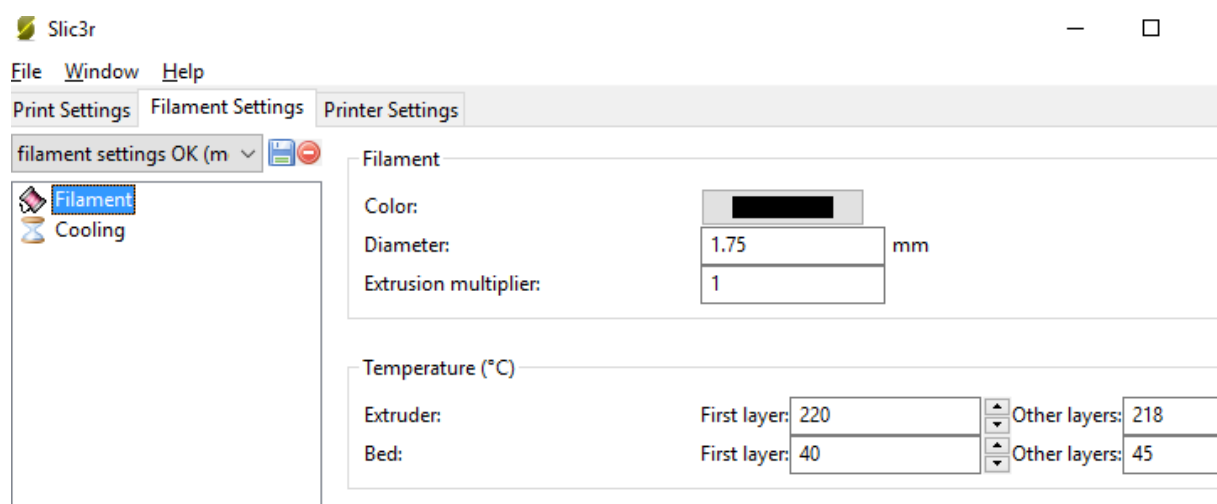
Loops (minimum): 3

Distance from object: 6 mm

Ilustración 2.3.9 Parámetros de Skirt

Ahora abriendo la pestaña Filament Settings, en las opciones del Filamento, se introduce diámetro 1.75 mm y multiplicador de extrusión 1 puesto que se va a utilizar un filamento de PLA de 1.75 mm de diámetro de calidad de la marca Smart Materials 3D teniendo una precisión de ± 0.03 mm.

En cuanto a las opciones de temperatura, se va a elegir para la primera capa una temperatura de 220 °C algo superior a las siguientes para que el material fluya con más facilidad, consiguiendo que se adhiera bien a la cama caliente evitando el defecto conocido como “warping”, para las siguientes capas la temperatura se bajara a 218 °C para conseguir las formas deseadas sin que se produzcan deformaciones. En cuanto a la cama caliente, para la primera capa, se asignará una temperatura de 40 °C y de 45 °C para las siguientes, consiguiendo con esto que la pieza no se enfríe con demasiada rapidez en su base. Quedando el menú como se puede ver más abajo en la ilustración 3.2.10.



Slic3r

File Window Help

Print Settings Filament Settings Printer Settings

filament settings OK (m)

Filament

Color: [Black]

Diameter: 1.75 mm

Extrusion multiplier: 1

Temperature (°C)

Extruder: First layer: 220 Other layers: 218

Bed: First layer: 40 Other layers: 45

Ilustración 2.3.10 Parámetros del filamento y temperaturas impresión

Una vez realizados estos pasos, se puede proceder a generar el archivo G-Code y guardarlo en la tarjeta SD que posteriormente introduciremos en la impresora para fabricar las distintas piezas. Para ello haciendo clic en el botón Slice con Slic3r, el

programa automáticamente generará el archivo G-Code con todos los parámetros predefinidos y listo para imprimir desde la impresora.

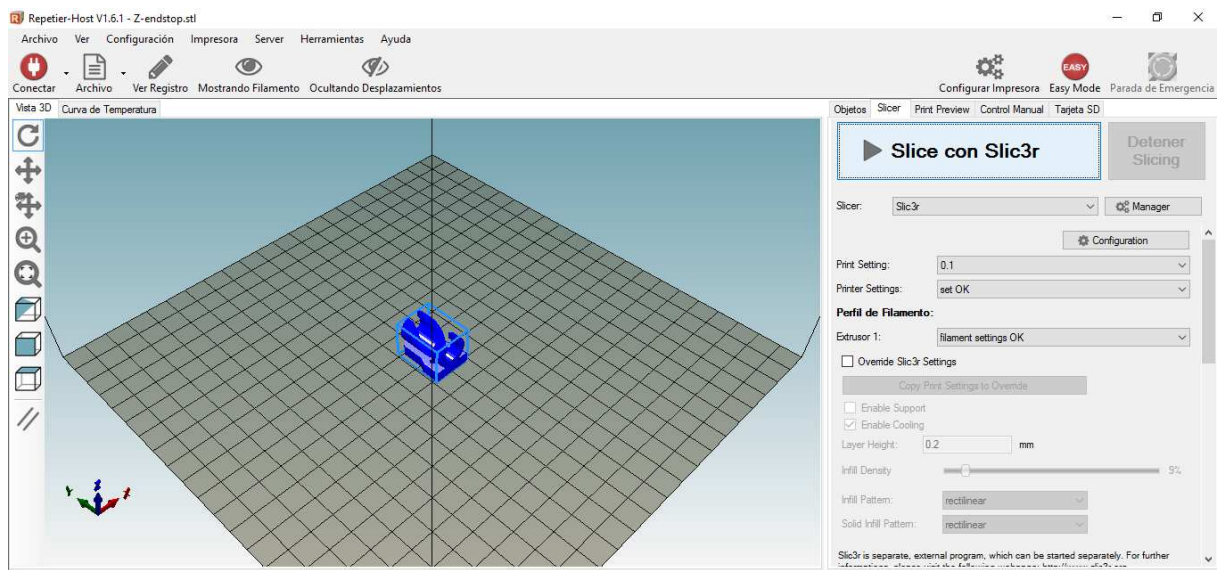


Ilustración 2.3.11 Interfaz del programa Repetier-Host antes de generar el G-Code

Además, al finalizar este proceso, nos permitirá la opción de guardarlo directamente en la SD de la impresora si está conectada al PC o guardar el archivo en un directorio dentro del ordenador. Mostrando una simulación de la impresión, y distintos datos como los metros de filamento necesarios o el tiempo requerido para su finalización, este menú se puede ver en la ilustración 2.3.12

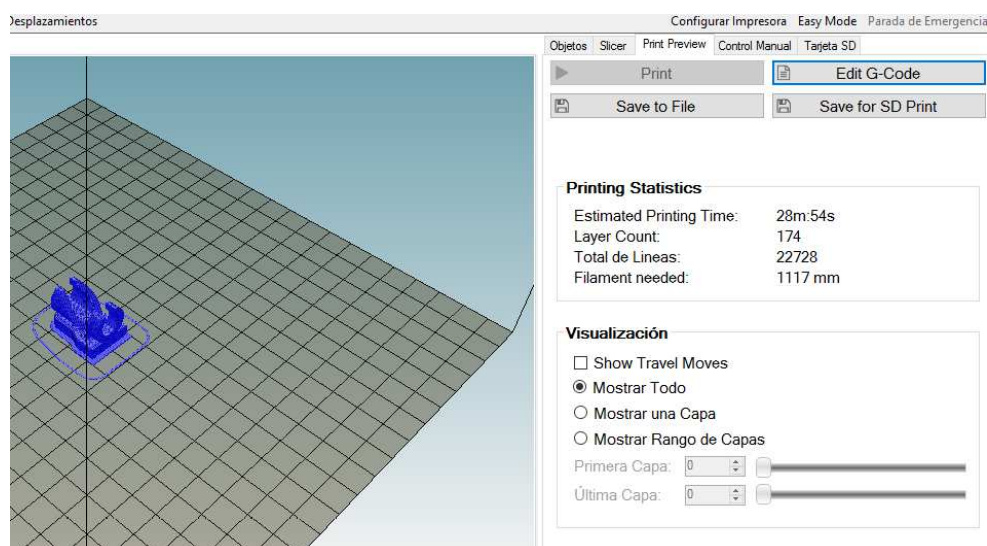


Ilustración 2.3.12 Interfaz de Repetier-Host tras generar el G-Code

Estos datos permitirán hacer una estimación del material necesario ya que una bobina de PLA de 1 kg, contiene aproximadamente unos 340 m de material.

3.2.3. Envío a Impresora 3D de los ficheros STL

Para poder transmitir estos archivos a la impresora, existen varios métodos, los más utilizados son con la impresora conectada al PC mediante USB, pudiendo controlar en todo momento todos los parámetros de impresión desde el ordenador, la desventaja es que el ordenador debe permanecer conectado a la impresora durante todo el proceso de impresión. El otro método más utilizado es mediante una tarjeta SD conectada al LCD de la impresora. El archivo G-Code generado anteriormente a partir del STL, se almacena en esta memoria flash pudiendo ser utilizado desde el LCD de la impresora, esta vez, los parámetros se controlan desde el propio LCD, siendo algo más complejo poder variar todos los parámetros ya que se encuentran en distintos menús.

3.2.3.1. Preparación de Impresora 3D

En primer lugar y antes de seleccionar el archivo a imprimir, se deben de comprobar ciertas cosas:

- Engrase de los husillos del eje Z y varillas lisas de todos los ejes.
- Limpieza de la cama caliente con acetona para eliminar cualquier resto de impresiones anteriores o de laca en la superficie.
- Una vez realizada una limpieza de la cama, se rocía la misma con laca, pulverizando una fina capa a unos 20-25 cm de la misma. Con esto evitaremos el común fallo del warping en el PLA, es decir que la pieza se desprende por alguna zona de la cama, deformándose por este lugar, este método no sirve para otros materiales por ejemplo el ABS.
- Calentamiento del nozzle 5 grados por encima de la temperatura máxima del plástico para conseguir que fluya y extruir un trozo del mismo para eliminar restos de impresiones anteriores.

3.2.3.2. Impresión de las piezas

Una vez realizados los puntos anteriores, se puede proceder a realizar la impresión seleccionando el archivo que se quiera de la tarjeta SD. Para las piezas de este sistema, se ha preferido optar por imprimirlas poco a poco en vez de llenar la

cama caliente de piezas consiguiendo con esto que las piezas tengan mejor calidad ya que el extrusor necesita hacer menos trayectos.

3.2.4. Extracción de las piezas impresas de la Impresora 3D

Cuando la impresión ha finalizado, el extrusor está configurado para que se aleje verticalmente de la pieza finalizada y vuelva a su posición de origen en el eje X para no molestar. Ahora es el momento de extraer la pieza de la cama caliente, este proceso es muy delicado ya que si no se tiene experiencia es muy fácil romper el objeto impreso. La forma más sencilla es con una cuchilla de cristales, proceder a introducirla suavemente a todo alrededor de la pieza, pudiendo ver la zona despegada por la parte inferior del cristal de la cama caliente. Una vez se compruebe que está despegada por todos los bordes, será muy sencillo desprenderla realizando un movimiento de rotación de la pieza sobre el cristal. Así se podrá obtener un acabado superficial en la primera capa brillante y muy liso. Si la capa de laca es demasiado gruesa, esta primera capa será algo rugosa.

3.3. Corte de perfiles BOSCH y varillas lisas

Teniendo ya todas las piezas impresas, es el momento de cortar los perfiles BOSCH de aluminio de 45x45 mm para poder fabricar la bancada del sistema desarrollado. Las piezas necesarias son 4 patas de 10 cm de longitud además de los 3 travesaños de 370 mm de longitud. Primero se miden estas medidas y se señalan en el perfil sin cortar y una vez señaladas correctamente se sitúa en la sierra de cinta horizontal, se alinea con la marca de corte y se cortan todas las piezas. El mismo proceso se utiliza para las varillas lisas, cuyos cortes se realizan a 400 mm de longitud, para las varillas guías del motor y para la varilla soporte del servo y el bote de spray de pintura.

3.4. Preparación de la electrónica

Por otro lado, queda poner a punto toda la electrónica para que el sistema funcione, como componentes se van a utilizar un Arduino MEGA 2560, con una placa Shell RAMPS 1.4 (siglas en inglés de RepRap Arduino Mega Pololu Shield), un pololu o driver A4988 por ser uno de los más utilizados a la vez que económicos en las

impresoras 3D, un regulador de voltaje L7805CV, un final de carrera mecánico de rodillo, un motor NEMA 17 y un servo motor Tower-Pro modelo MG996R. Cuando se tienen todos los componentes se han de preparar para su correcto funcionamiento, para ello, se debe comenzar por la adecuación del código open-source conocido como “Marlin”.

3.4.1. Modificación del código libre Marlin

Este software viene genérico, necesitando ser adecuado a cada uso, este software se ha descargado de <https://github.com/MarlinFirmware/Marlin> que es la página del proyecto Marlin. En este caso en concreto es necesario modificar el siguiente código:

- Se carga en el programa de Arduino el archivo de extensión “.ino” y se abre la pestaña configuration.h, primero hay que elegir la placa shield que se va a utilizar, en este caso, es el RAMPS 1.4 por lo que se debe poner el número 33 que es el que viene en la lista para este, quedando como se ve en la ilustración 3.4.1

```
#ifndef MOTHERBOARD
#define MOTHERBOARD 33
#endif
```

Ilustración 3.4.1 Selección placa Shield

- Ahora en la línea de la temperatura del nozzle, en la primera posición, se cambia a 0, quedando como en la ilustración 3.4.2 ya que en el sistema no va a haber ningún nozzle y este código está hecho para impresoras 3D que sí lo requieren. Si no se cambiara a cero, al encender el sistema daría un error puesto que no encontraría el sensor para medir la temperatura establecida.

```
#define HEATER_0_MINTEMP 0
#define HEATER_1_MINTEMP 5
#define HEATER_2_MINTEMP 5
#define BED_MINTEMP 5
```

Ilustración 3.4.2 Eliminada de temperatura mínima

- El siguiente paso es modificar la parte del código para aumentar el recorrido del eje Z que es el que interesa para el sistema

desarrollado, para ello debe quedar como aparece en la ilustración 3.4.3, es decir con un valor máximo de 250 mm, puesto que el husillo es de 300 mm y debemos de quitar las partes de los anclajes.

```
// Travel limits after homing
#define X_MAX_POS 190
#define X_MIN_POS 0
#define Y_MAX_POS 180
#define Y_MIN_POS 0
#define Z_MAX_POS 250
#define Z_MIN_POS 0
```

Ilustración 3.4.3 Establecimiento de la máxima longitud del eje Z

- Para definir el husillo, en este caso se montará un husillo de 8 mm de diámetro y de paso 2 mm, se debe de configurar la línea tal cuál aparece en la ilustración 3.4.4, la tercera cifra del corchete son los pasos por mm, para calcularlo se utiliza la siguiente fórmula:

$$\frac{\text{Pasos}}{\text{mm}} = \frac{\text{Pasos NEMA17} * \text{Resolucion Driver A4988}}{\text{Paso husillo}}$$

Y sustituyendo valores, se obtiene:

$$\frac{\text{Pasos}}{\text{mm}} = \frac{200 * 16}{2} = 1600 \frac{\text{Pasos}}{\text{mm}}$$

(Fórmula 3.4.1)

```
// default settings

#define DEFAULT_AXIS_STEPS_PER_UNIT {80,80,1600,777} // de
#define DEFAULT_MAX_FEEDRATE {330, 330, 3.3, 45}
#define DEFAULT_MAX_ACCELERATION {1500,1500,100,10000}
// (9000,9000,100,10000)
#define DEFAULT_ACCELERATION 1000 // X, Y, Z and
#define DEFAULT_RETRACT_ACCELERATION 2000 // X, Y, Z and E
```

Ilustración 3.4.4 Configuración de los pasos por mm del motor

- Más adelante, se encuentran las líneas de código relacionadas con el LCD, primero se activa la SD, para ello se quita la doble barra a la izquierda de la línea que contiene SDSUPPORT, asegurando que #define deje de ser de color gris como queda en la ilustración 3.4.5. Un poco más abajo se tienen las opciones para activar distintos tipos de LCD, en este caso, el LCD que se va a montar es un RepRapDiscount FULL GRAPHIC, por lo que se debe buscar la línea de este LCD y activarla del mismo modo que se ha activado el soporte para tarjeta SD, la única línea que se debe quedar activa es la que contiene #define quedando como en la ilustración 3.4.5.

```
//LCD and SD support
//#define ULTRA_LCD //general lcd support, also 16x2
//#define DOGLCD // Support for SPI LCD 128x64 (Controller:
#define SDSUPPORT // Enable SD Card Support in Hardware Con
//#define SDSLOW // Use slower SD transfer mode (not normal)
```

Ilustración 3.4.5 Activación de la tarjeta SD

```
//The RepRapDiscount FULL GRAPHIC Smart Controller (quadratic white PCB)
// http://reprap.org/wiki/RepRapDiscount\_Full\_Graphic\_Smart\_Controller
//
// ==> REMEMBER TO INSTALL U8glib to your ARDUINO library folder: http://code.c
#define REPRAP_DISCOUNT_FULL_GRAPHIC_SMART_CONTROLLER
```

Ilustración 3.4.5 Configuración del LCD

- Por último, queda activar el servo, esta opción viene desactivada de forma predefinida, ya que solo las impresoras 3D de tipo Delta lo usan, además de ciertas tipo Prusa para la nivelación de la cama caliente. Para esto, se activa la línea de la misma forma que se ha hecho anteriormente, es decir, eliminando la doble barra a la izquierda y a continuación se cambia el número de servos de cero a uno como se muestra en la ilustración 3.4.6, que será el servomotor que se va a utilizar dentro del sistema para activar el bote de spray.

```
// Number of servos
//
// If you select a configuration
// set it manually if you have mo
// leaving it undefined or defini
// If unsure, leave commented / d
//
#define NUM_SERVOS 1

#include "Configuration_adv.h"
#include "thermistortables.h"
```

Ilustración 3.4.6 Habilitación del servomotor

Con esta última modificación, el software queda configurado para implementarlo en Arduino, si todo ha ido bien y las configuraciones son las correctas, no debe de existir ningún tipo de problema a la hora de cargar este software dentro del Arduino MEGA 2560 ya que es totalmente compatible con Marlin.

3.4.2. Implementación en ARDUINO

Cuando el software Marlin es el correcto, el siguiente paso es cargarlo en el hardware Arduino MEGA, los pasos a seguir son sencillos, pero hay que seguirlo estrictamente ya que, si no se hacen en orden, no se podrá interactuar con la placa Arduino desde el ordenador.

- Instalar los drivers para que el ordenador reconozca el Arduino MEGA, simplemente se pueden descargar de la página oficial de Arduino <https://www.arduino.cc/en/Main/Software> para distintos sistemas operativos.
- Una vez instalados los drivers necesarios, se debe seleccionar la placa que se va a utilizar, dentro del menú superior Herramientas, se despliega la opción Placa y se selecciona Arduino/Genuino MEGA or MEGA 2560, tal y como se muestra en la ilustración 2.4.7.

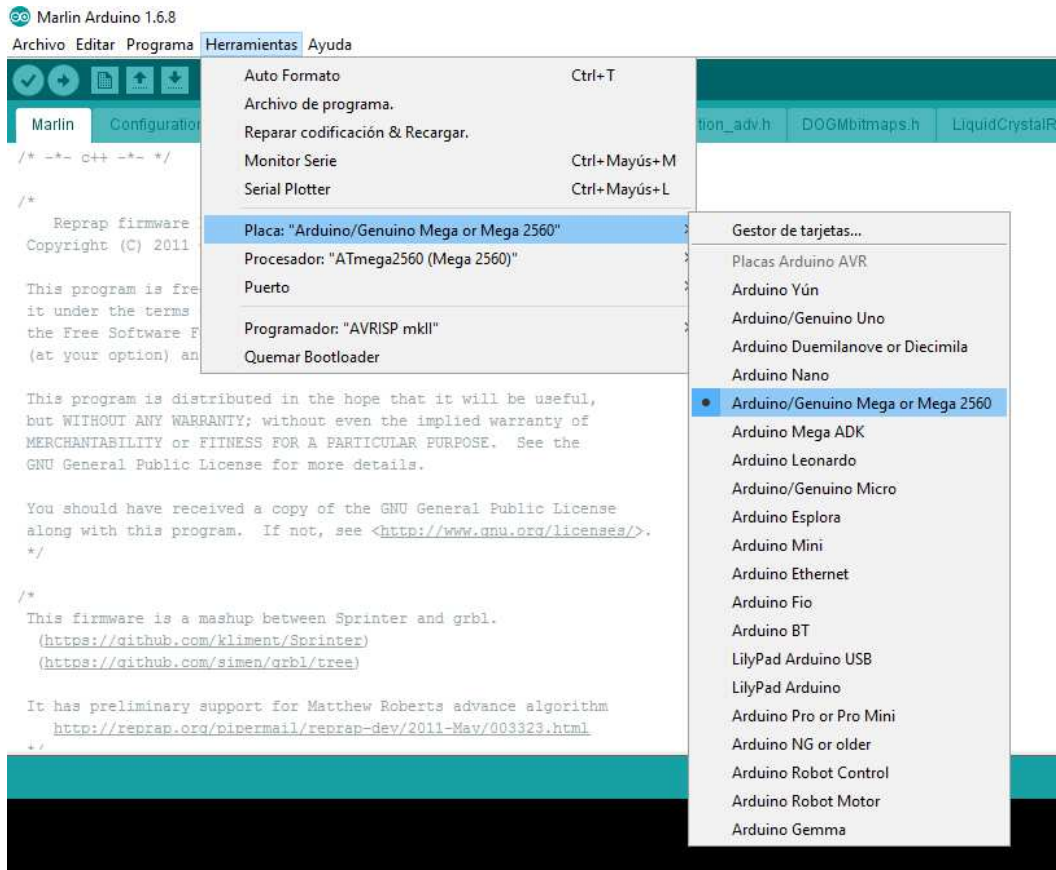


Ilustración 3.4.7 Menú Herramientas

- A continuación, se abre el software Marlin modificado anteriormente, para ello se selecciona del menú superior la opción Archivo y Abrir, buscando el directorio donde está guardado el Marlin modificado y abriendo el archivo Marlin.h, se puede ver en la ilustración 3.4.8 el proceso seguido.

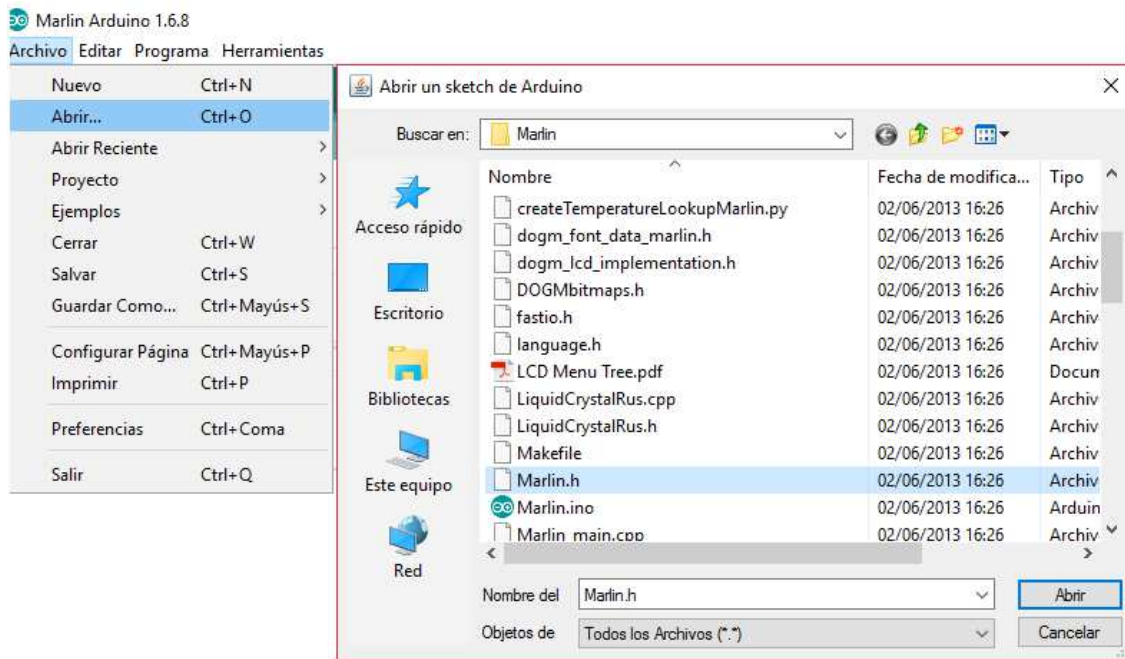


Ilustración 3.4.8 Menú Archivo con ventana abrir desplegada

- Una vez hecho este proceso, y teniendo conectada la placa Arduino al puerto USB del ordenador, sólo queda pulsar en el botón Subir que se indica en amarillo en la ilustración 3.4.9.



Ilustración 3.4.9 Botón Subir marcado en amarillo

- Una vez aparezca la palabra “Subido” como se ve en la imagen 3.4.10, en la parte inferior de la interfaz, se puede desconectar la placa Arduino del USB, quedando cargado el software Marlin en el Arduino.

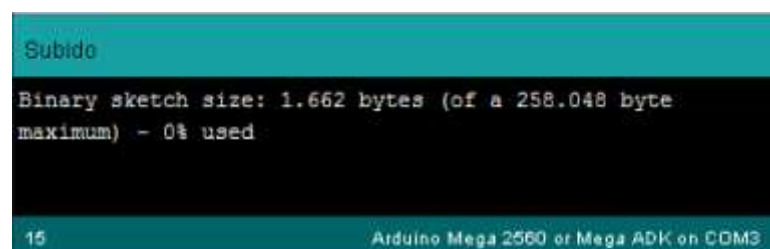


Ilustración 3.4.10 Aviso de proceso completado correctamente

El siguiente paso es calibrar en vacío todos los componentes electrónicos, con el fin de evitar posibles catástrofes una vez esté todo el sistema ensamblado.

3.4.3. Calibrado y puesta a punto

En este siguiente paso, se conectará todo fuera de la estructura, para comprobar que todo funciona como debería, es decir que el final de carrera detiene el motor paso a paso, que el motor gira a una velocidad adecuada y que el servo gira el ángulo requerido en el momento necesario. Para conseguir hacer funcionar el servomotor, es necesario un regulador de voltaje L7805CV ya que, si la alimentación de este se conectara a la placa Arduino, se corre el riesgo de quemarla por lo que se necesita cambiar de 12 V a 5 V que es justamente lo que hace este pequeño componente electrónico el cual se puede ver en la ilustración 3.4.11:

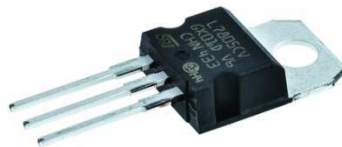


Ilustración 3.4.11 Regulador de voltaje L7805CV (Nextiafenix, 2016)

Para calibrar driver o pololu, es necesario que circule por éste una intensidad de 0.2 A para que el motor tenga la fuerza suficiente sin llegar a sobrecalentarse. Tras conectar todo en su sitio, se pasa el negativo de la fuente de alimentación por el amperímetro y se le manda una acción larga al motor, mientras tanto, viendo la intensidad que circula, se ajustará con un destornillador cerámico el pequeño tornillo del potenciómetro que trae el driver, hasta conseguir que marque 0.2 A, quedando todo como se muestra en la ilustración 3.4.12.

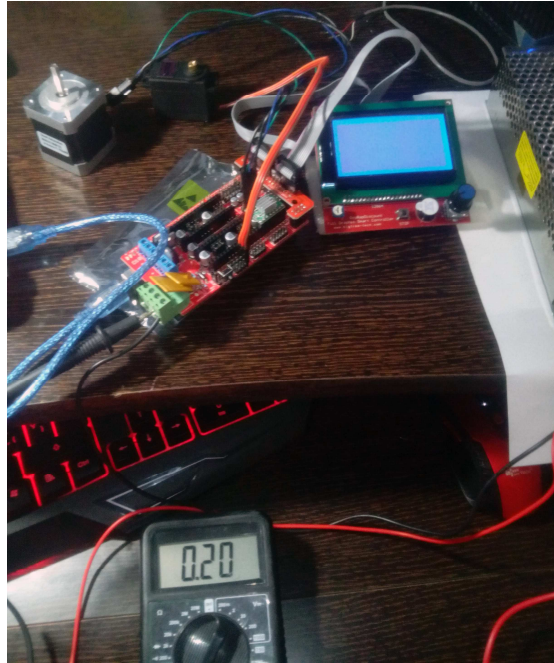


Ilustración 3.4.12 Calibrado del driver o pololu

Tras tener bien calibrado el motor, y comprobar que efectivamente el final de carrera para el motor cuando se activa y que el servo gira el ángulo que se le esté mandando, se considera calibrada la electrónica para ser montada en el ensamblaje.

3.5. Ensamblaje del sistema

A continuación, se procede al ensamblaje de todo el sistema diseñado. Este proceso es de los más importantes ya que, si algo se monta de forma errónea, se podrían romper piezas ya fabricadas. Además, en este punto será donde comenzarán a surgir algunos problemas que no hayan surgido durante el proceso de diseño y desarrollo de la máquina.

- En primer lugar, se va a formar la bancada además de la estructura que hubo que modificar, nivelando las patas para que el sistema se apoye perfectamente sobre el suelo, intentando eliminar todo tipo de vibración que pudiera ocasionar un mal nivelado, quedando todo unido como en la ilustración 3.5.1, la bancada no irá fija a la estructura para poder transportar el sistema de manera sencilla en un futuro.

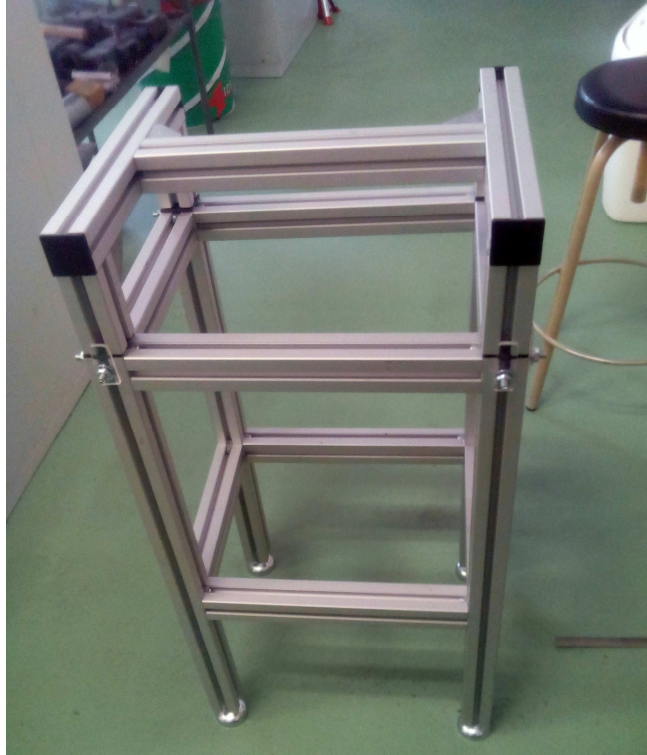


Ilustración 3.5.1 Bancada y estructura de perfiles BOSCH

- A continuación, en la ilustración 3.5.2 se muestra el resultado de montar las piezas soporte en sus correspondientes sitios, es decir, al husillo se le monta en un extremo el soporte para la cama (a la derecha) y al motor nema se le monta el soporte correspondiente, fijando además los rodamientos en sus huecos (a la izquierda).

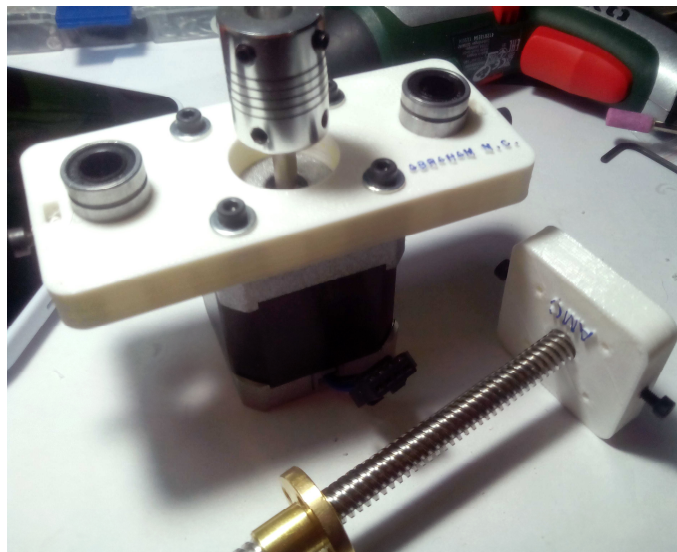


Ilustración 3.5.2 Soportes motor (derecha) y husillo (izquierda)

- Ahora es el turno de colocar las varillas lisas en sus respectivos orificios y además de fijar la tuerca del husillo a la bancada, permitiendo montar el husillo y el motor sobre las varillas lisas y, por último, fijando el husillo al acople del motor mediante los dos tornillos sin cabeza que dispone el acople, el sistema queda tal y como se ve en la ilustración 3.5.3.



Ilustración 3.5.3 Montaje de la parte móvil del ensamblaje

- Ya solo queda fijar la cama al soporte, fijar las varillas a la bancada con sus respectivas guías y montar los soportes del servomotor y del bote de spray en la guía superior quedando el sistema del servomotor-bote de spray que se muestra en la ilustración 3.5.4.



Ilustración 3.5.4 Sistema de palanca servomotor – bote de spray

- Para terminar, se fija el embellecedor de las varillas lisas en su extremo inferior como queda en la ilustración 3.5.5 y estarán todas las piezas no electrónicas montadas.

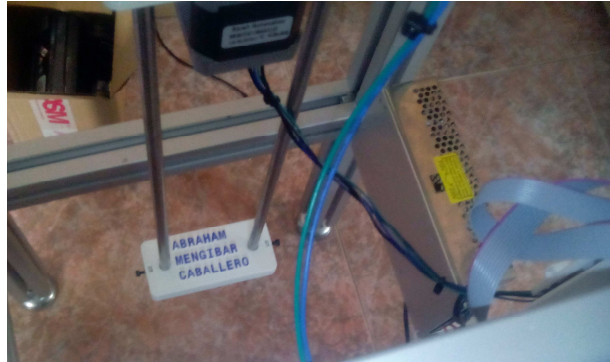


Ilustración 3.5.5 Embellecedor de varillas lisas

- Ahora, se monta la electrónica en el sistema. Para ello, primero se sitúa la caja del Arduino en su lugar y se fija con dos acoples para los perfiles BOSCH. Después se puede montar el Arduino en su lugar y se extraen los cables de la caja pasándolos por las ranuras de los perfiles de aluminio para que no molesten, se adecua la fuente de alimentación en su lugar y se conecta el servomotor. Ahora se coloca el final de carrera en su fijación y se ajusta a la altura necesaria. Por último, se coloca el soporte del LCD en un sitio accesible y se fija el LCD. Quedando el ensamblaje completamente montado y listo para ser probado. El ensamblaje terminado se puede ver en la ilustración 3.5.6.

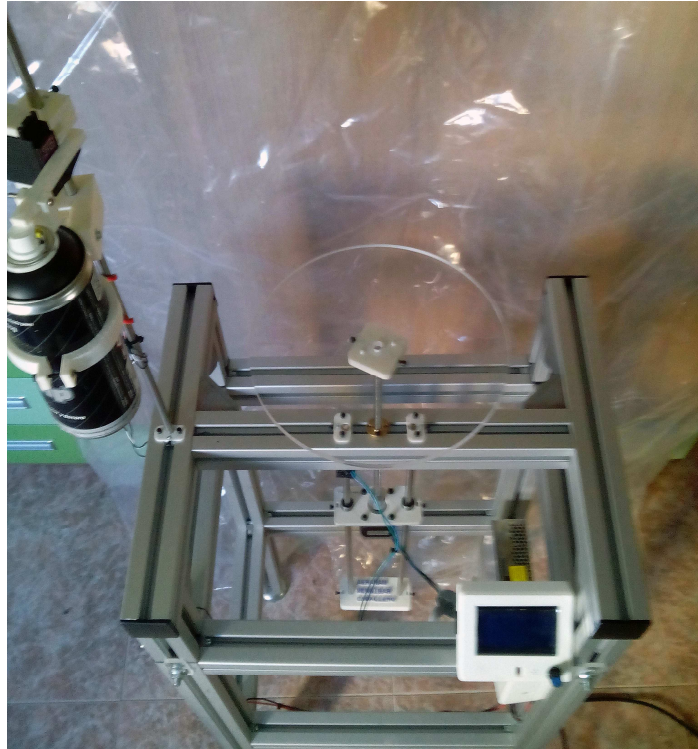


Ilustración 3.5.6 Ensamblaje terminado

3.6. Problemas surgidos y soluciones en el proceso de desarrollo del sistema

En esta sección, se van a exponer los diferentes problemas surgidos y las soluciones adoptadas durante todo el proceso de desarrollo.

- Uno de los primeros problemas que aparecieron, fue como poder alimentar el servomotor entre su rango sin quemar la placa Arduino. Si se conectaba directamente a los puertos que trae la placa Shield RAMPS, existía una posibilidad muy alta de que se quemara, ya que el servomotor que se iba a utilizar era de 9 kg, es decir, bastante grande. Se debe tener en cuenta que, a pesar de que el Arduino se alimente a través de la placa RAMPS conectada a la fuente de alimentación de 12 V, esta alimentación, se reparte entre el Arduino, el motor Nema 17, el driver o pololu y el LCD. Si además se conectara el servomotor, faltaría corriente para alimentar el sistema. Puesto que la fuente de alimentación dispone de tres salidas, se optó por alimentar el servomotor de forma independiente a la placa con otra salida de la fuente, teniendo ahora el problema de que la fuente saca un voltaje de 12 V y el servomotor funciona en un rango de voltaje

operativo de: 4.8 V a 7.2 V. Para evitar tener que utilizar un potenciómetro o algún pequeño circuito con resistencias, se prefirió utilizar un simple componente electrónico que proporciona el resultado esperado. Este componente es un regulador de voltaje L7805CV que transforma el voltaje de entrada de 12 V en uno inferior de salida de 5 V estando este voltaje entre el rango del servomotor. Con este sencillo componente, se pudo subsanar el problema de la alimentación externa del servomotor sin necesidad de un sistema complicado.

- Otro pequeño problema que se observó al comenzar a utilizar la máquina, es que, en su primer diseño no se les había puesto un embellecedor a las varillas lisas en su extremo inferior, esto daba como resultado que las varillas lisas se movían debido al movimiento del motor y su soporte a lo largo de estas ya que son las varillas lisas junto con los rodamientos, los encargados de hacer que el motor se desplace a lo largo de su longitud sin girar sobre sí mismo. Para eliminar este pequeño problema, se diseñó el embellecedor de las varillas deslizaderas que, colocado en su extremo inferior, fijaban la posición de estas, evitando que se produjera un movimiento de las mismas cuando el motor se deslizaba por ellas.
- Como último problema y más grave, fue idear el sistema de accionamiento de la boquilla de spray con el servomotor. En un primer lugar, el sistema creado fue un sistema de martillo. Aquí se trataba de un acople conectado a la salida del servomotor con una maza en el extremo opuesto para que el servomotor al girar, accionara la boquilla del bote de spray. El problema fue que, al accionar el servo, este no tenía la suficiente fuerza para apretar la boquilla, a pesar de ser un servo bastante grande, la presión ejercida era insuficiente. En la ilustración 3.6.1 se ve el diseño del acople de maza y el sistema montado en la ilustración 3.6.2.

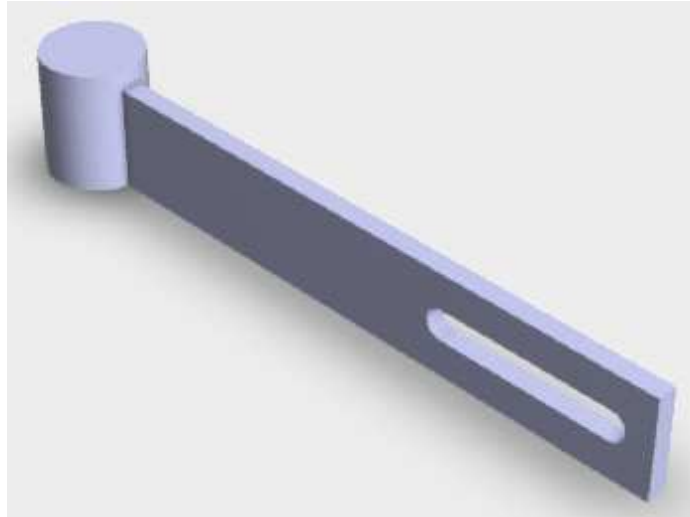


Ilustración 3.6.1 Sistema de martillo



Ilustración 3.6.2 Sistema de martillo montado

Puesto que este sistema no funcionaba correctamente, se recurrió al sistema finalmente utilizado de palanca de primer grado, siendo este último totalmente efectivo y funcional para el propósito que se buscaba, siendo la que aparece en la ilustración 3.6.3.



Ilustración 3.6.3 Sistema de palanca de primer grado

4. REALIZACIÓN DE ENSAYOS

Tras tener el sistema con un correcto funcionamiento y listo para realizar los ensayos, primero se deben de crear las probetas en las cuales se realizarán los ensayos.

4.1. Probetas

Se van a desarrollar unas probetas de acuerdo a las premisas con las que se ideó el sistema, es decir, las dimensiones máximas de las piezas de revolución deben contener una altura máxima de hasta 25 cm, además de ser piezas fabricadas mediante FDM. Estas probetas se imprimirán en PLA ya que es uno de los materiales más utilizados en la fabricación FDM.

4.1.1. Diseño de probetas

Igual que para el resto de piezas diseñadas para este proyecto, se utilizará el software CATIA. Las probetas serán cilindros con un diámetro igual a 60 mm y una altura de 100 mm cuyo diseño se muestra en la ilustración 4.1.1, aunque se podrían haber diseñado con mayor altitud, no se ha hecho puesto que los resultados serían los mismos independientemente de la altura a la que se le aplique el acabado.

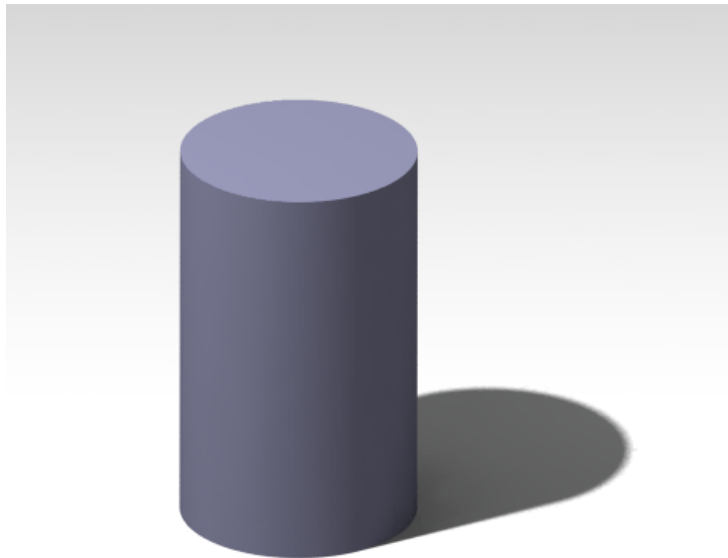


Ilustración 4.1.1 Probeta de 60 diámetro y 100 mm de longitud

4.1.2. Impresión de probetas

Para imprimir las probetas, se va a hacer de una en una para obtener la mejor calidad de impresión posible. Además, se imprimirán cuatro exactamente iguales puesto que se realizarán cuatro ensayos con distinta cantidad de capas de pintura de vinilo líquido, las cantidades serán 4, 6, 8 y 10 respectivamente.

En cuanto a los parámetros de impresión, se utilizarán los mismos que para las piezas impresas utilizadas en el ensamblaje del sistema es decir altura de capa 0.1 mm y mismas velocidades y temperaturas de impresión, exceptuando la densidad. Puesto que el interés de las probetas es puramente externo, estas se imprimirán huecas, es decir con la densidad o infill al 0%.

4.1.2.1. Extracción de probetas de la Impresora 3D

Puesto que se trata de piezas altamente delicadas y altas, la extracción de la cama caliente se hará mediante otro sistema. Esta vez, se dejará enfriar la pieza una hora sobre el cristal, pero fuera de la cama caliente. Una vez transcurrido este tiempo, se verterá agua fría por la parte trasera del cristal de la cama caliente consiguiendo con esto que la pieza se despegue casi sin ayuda del cristal. Con este método, se consigue que no se raje la pieza por ninguna capa además de no dañar la primera capa que es la que está en contacto con el cristal de la cama caliente. Las 4 probetas impresas se pueden ver en la ilustración 4.1.2.



Ilustración 4.1.2 Probetas impresas mediante FDM

4.1.2.2. Encintado de probetas

Una vez impresas las probetas se procederá a encintar con cinta de enmascarar una parte de las mismas, para una vez realizados los ensayos, poder medir sobre ambas superficies la rugosidad con el rugosímetro.

4.1.2.3. Posicionamiento de las probetas en el sistema

Una vez encintadas las probetas, se utilizará una pequeña tira de cinta adhesiva de doble cara como la que aparece en la ilustración 4.1.3 en su base para fijarla a la cama del sistema desarrollado, con el fin de que la pieza no se mueva durante el proceso de pintado. El centro de la cama es fácil reconocerlo ya que tiene un pequeño agujero en el centro que fue necesario hacerle al crear su geometría por el técnico de laboratorio.

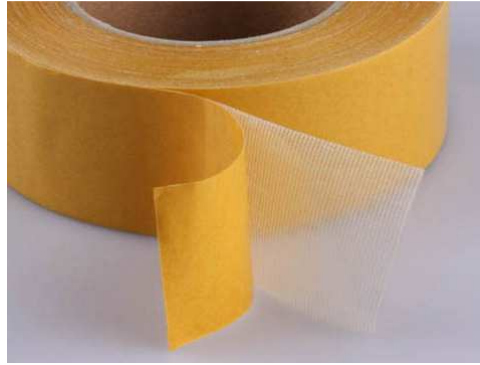


Ilustración 4.1.3 Cinta adhesiva de doble cara (Google, 2016)

4.2. Creación del programa G-code a introducir

Para crear el programa en lenguaje G-code, es necesario tener ciertos conocimientos previos acerca de este lenguaje de programación. Primero, se va a explicar qué movimientos se quiere para luego implementarlos en el código. El código se puede generar con cualquier editor de texto.

En primer lugar, cuando el usuario ponga en marcha el programa, la máquina debe subir la plataforma hasta su punto más alto, para ello el husillo girará sobre su tuerca empujado por el motor hasta que la plataforma del motor accione el final de carrera mecánico. Una vez esté en este punto, la máquina se parará y, al mismo tiempo, se accionará el servomotor que presionará la boquilla, obteniendo la pulverización y además el motor girará en sentido contrario al anterior, bajando la plataforma hasta la altura que se haya programado. Cuando la plataforma haya bajado al punto deseado, el servomotor dejará de accionar el bote de spray, dando por finalizado el proceso de una capa.

A continuación, se muestra el código utilizado, explicado por líneas.

```
M280 P0 S80  
G28 Z  
G4 S1  
M280 P0 S10  
G1 F5000  
G1 Z100  
G4 P100  
M280 P0 S80  
G4 P100  
M84
```

M280 P0 S80: posiciona el servo de la posición P a los grados S respecto de su origen. En este caso, en su posición de reposo, este servomotor es de 180°, pero como se conoce que no se obtiene tanta precisión si se trabaja en los límites, se prefiere un rango entre 10° y 170°.

G28 Z: se ordena al eje Z que se sitúe en su punto de origen.

G4: es un comando de espera, en la S se fijan los segundos de espera, si no se usa este comando, es muy fácil que el código no funcione bien.

M280 P0 S10: se trata del mismo comando utilizado antes, hace que el servo gire a su posición de 10°.

G1 F5000: se fija el avance por minuto para el husillo

G1 Z100: ahora se ordena que el husillo se mueva hasta la posición 100 mm que es la altura de las probetas.

G4 P100: mismo comando de espera, pero ahora en milisegundos con el parámetro P, es decir esperaría 100 milisegundos.

M280 P0 S80: vuelve el servomotor a su posición de reposo.

G4 P100: vuelve a esperar 100 milisegundos.

M84: libera de cualquier retención al motor.

Con esto, se tendría el código G-code listo para introducirlo en la tarjeta SD del LCD del sistema.

4.2.1. Transferencia del programa al sistema

Una vez completado el código, hay que guardarlo con la extensión “.gcode” para que sea reconocido por el LCD y la máquina pueda interpretarlo. Para ello:

- En este caso, se ha utilizado la aplicación Bloc de notas. Para exportarlo a G-code, simplemente se hace clic en Archivo situado en la barra

superior de menú, a continuación, en Guardar como...tal y como se muestra en la ilustración 4.2.1.

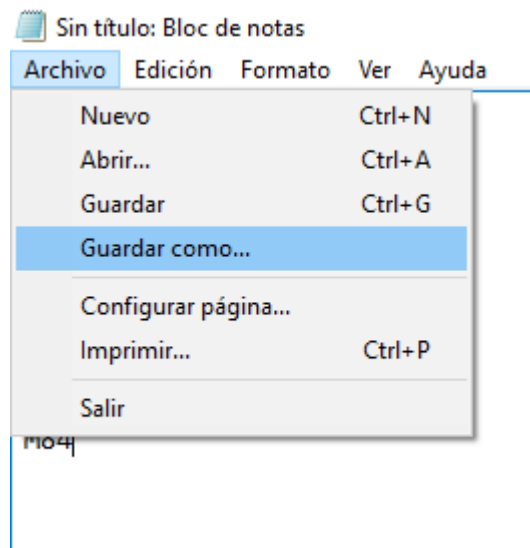


Ilustración 4.2.1 Menú Archivo

Tras desplegarse la ventana de guardado, se le pone nombre y a continuación “.gcode”, eliminando el “.txt” que viene por defecto. Quedando como se puede ver en la ilustración 4.2.2, se podría guardar directamente en la tarjeta SD para el sistema.

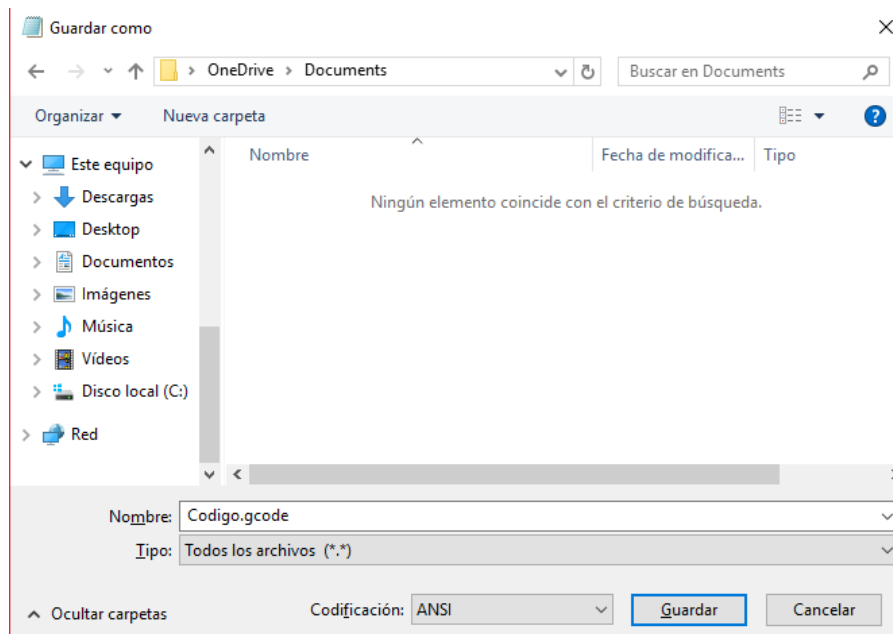


Ilustración 4.2.2 Ventana Guardar Como desplegada

4.3. Manipulación de la pintura y adecuación del lugar

Este punto es muy importante para la seguridad de las personas que van a utilizar el sistema, ya que, al tratarse de pintura en aerosol, se necesita un sitio con buena ventilación o en su defecto el uso de mascarillas durante el proceso de acabado. En este caso se hará uso de una habitación cerrada y pequeña por lo que será primordial utilizar mascarillas durante el uso del sistema, además de ventilar bien la sala tras el uso de la máquina. De esta forma se evita el polvo en suspensión o posibles corrientes de aire que se pueden generar en una habitación abierta.

Para preparar el entorno de la máquina, solo es necesario cubrir con un plástico la zona situada enfrente de la boquilla de pintura, ya que se pensó en poner un panel de acetato que envolviera la máquina, pero para reiterados usos, es más económico y limpio un plástico.

Una vez tapado todo, ya está el sistema y su entorno preparado para el uso para el cuál se ha creado.

4.4. Ensayos

Se van a realizar cuatro ensayos. En el primero se utilizarán 4 capas de producto, para ir incrementando de dos en dos hasta un máximo de 10 capas.

Para realizar el ensayo:

- Primero se posiciona la probeta con cinta adhesiva de doble cara en su base en el centro de la plataforma.
- A continuación, se verifica que los soportes del servo y el bote de pintura están en sus marcas. El bote de pintura primero se agita como recomienda el fabricante y a después se posiciona a 11 cm desde el borde inferior del soporte medidos hasta su base, quedando como en la ilustración 4.4.1 y finalmente fijando su posición con el perno del soporte.



Ilustración 4.4.1 Posición correcta bote de spray

- Ahora se conecta el sistema a la corriente eléctrica y si el servomotor está en su posición de reposo, se coloca la palanca sujeta al bote de pintura en su sitio, además se gira la boquilla para hacerla coincidir con la marca en el soporte de la palanca, obteniendo su posición final como se puede ver en la ilustración 4.4.2. Si no fuera así, se utiliza el comando OFF PINTURA que se ha introducido en la SD del LCD para posicionar el servomotor en su posición de reposo, si existiera el caso de no estar bien posicionado (corte de luz o algún otro problema).



Ilustración 4.4.2 Posición final sistema de pintado

- Ahora está todo listo para iniciar el G-code desde el LCD y dejar que la máquina realice su ciclo.
- Entre distintas capas, se va a dejar secar el producto mínimo 20 minutos para evitar que las sucesivas capas se escurran unas sobre otras. En este caso, el tiempo de espera se ha realizado mediante temporizador digital, por evitar un corte de luz durante el proceso automático, que simplemente se haría añadiendo al código G-code una línea de espera de 20 minutos es decir 1200 segundos y volver a repetir el código ya generado tantas veces como capas se quieran, el ejemplo para 4 capas quedaría (los comentarios en G-code se ponen tras un punto y coma):

M280 P0 S80

G28 Z

G4 S1

M280 P0 S10

G1 F5000

G1 Z100

G4 P100

M280 P0 S80

G4 P100

M84

G4 S1200; Aquí se esperan 20 minutos y se vuelve a repetir el ciclo.

M280 P0 S80

G28 Z

G4 S1

M280 P0 S10

G1 F5000

G1 Z100

G4 P100

M280 P0 S80

G4 P100

M84

G4 S1200; de nuevo, se esperan 20 minutos y se repite el ciclo

M280 P0 S80

G28 Z

G4 S1

M280 P0 S10

G1 F5000

G1 Z100

G4 P100

M280 P0 S80

G4 P100

M84

G4 S1200; por última y cuarta vez, se espera el tiempo de secado y se repite el ciclo

M280 P0 S80

G28 Z

G4 S1

M280 P0 S10

G1 F5000

G1 Z100

G4 P100

M280 P0 S80

G4 P100

M84

- Cuando se ha completado el proceso, se deja secar mínimo otros 20 minutos y con cuidado se quita la cinta de enmascarar para evitar que la pintura se seque y se convierta en vinilo. Por último, se pone en un lugar aislado de corrientes de aire y se deja curar 12 horas.
- Como se ha dicho anteriormente, este proceso se realizará en las 4 probetas, para 4, 6, 8 y 10 capas de pintura de vinilo líquido con el fin de comprobar como mejora el acabado superficial de las piezas impresas mediante FDM. Las 4 probetas se pueden ver en la ilustración 4.4.3



Ilustración 4.4.3 Probetas tras los distintos ensayos

- De izquierda a derecha en la anterior ilustración, se tienen 4, 6, 8 y 10 capas, se puede apreciar como a medida que las capas aumentan,

comienza a desaparecer el efecto de las líneas de las capas de la impresión en las piezas.

- Cuando se han terminado los ensayos, se deja que los restos de pintura adheridos a la plataforma se sequen, tras 12 horas se podrán despegar como si se tratara de una lámina de vinilo, volviendo a su estado original sin apenas dificultad.

4.5. Medida de rugosidad

Para medir la rugosidad, se hará uso del rugosímetro Mitutoyo SJ-210 del departamento siguiendo los parámetros establecidos en la norma ISO1997 además de una velocidad de avance de 0.5 mm/s además de un λ_c de 2.5 obtenido del manual del rugosímetro, específico para el rango de mediciones que se va a obtener. El rugosímetro se configura para que muestre datos de Ra (rugosidad media) y Rz (desviación de la rugosidad media) ambos medidos en micras. Primero se debe de crear una pequeña estructura para poner a nivel el rugosímetro con las probetas, para esto se utilizan unas cuñas metálicas para conseguir la altura deseada, además de un soporte metálico en “V” para apoyar las probetas. Para el rugosímetro se usa un soporte metálico plano, teniendo cuidado de no apretar el palpador del rugosímetro contra la pieza, ya que éste es muy delicado. El montaje queda como se muestra en la ilustración 4.5.1

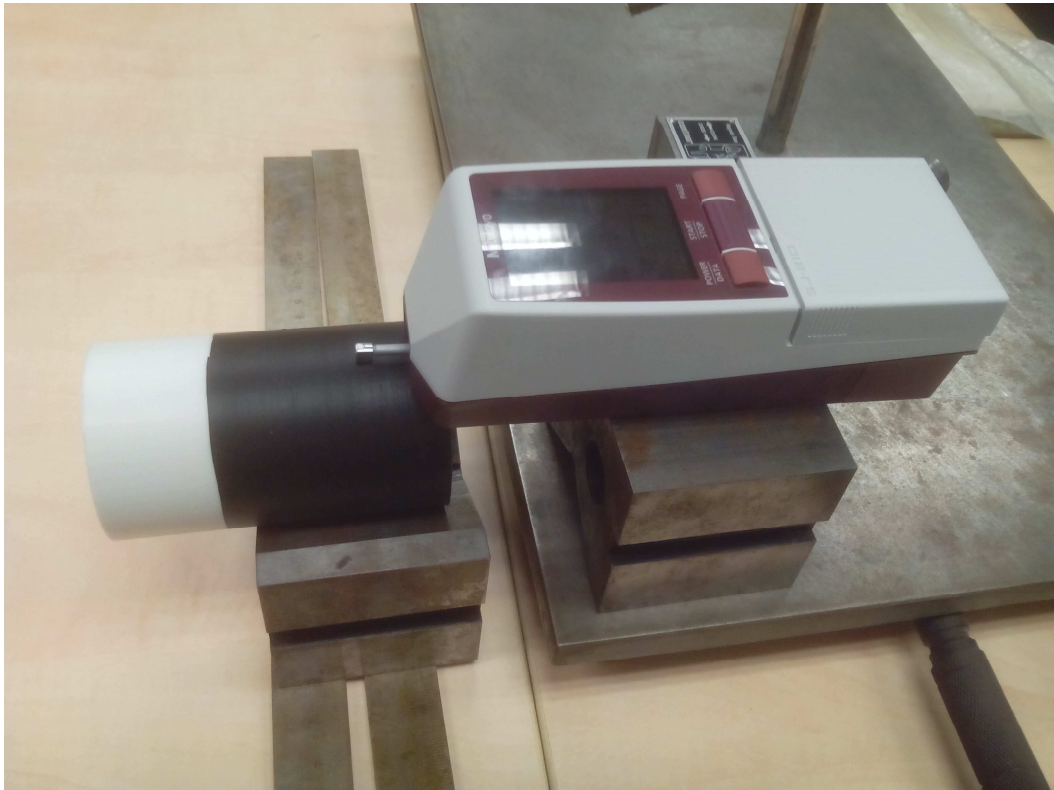


Ilustración 4.5.1 Montaje para medida con rugosímetro

Una vez que la estructura para medir está lista, se puede proceder a obtener los datos de cada uno de los ensayos.

4.6. Resultados obtenidos en los ensayos

Primero se miden las rugosidades de las 4 probetas por el lado sin tratar y se obtiene la media. Después se mide la rugosidad de cada probeta en la misma línea que se ha medido la rugosidad por el lado sin tratar, pero esta vez por la parte tratada con la pintura de vinilo líquido. Las medidas se realizan 3 veces para probar que la medida es correcta y que el rugosímetro no está falseando ningún dato. En las ilustraciones 4.6.1.a – f se pueden ver las medidas obtenidas por orden, es decir sin tratar seguido de 4, 6, 8 y 10 capas.



Ilustración 4.6.1-a. Pieza sin tratar



Ilustración 4.6.1-b. Tratamiento de 4 capas



Ilustración 4.6.1-c. Tratamiento de 6 capas



Ilustración 4.6.1-d Tratamiento de 8 capas



Ilustración 4.6.1-e. Tratamiento de 10 capas

A simple vista se puede observar que, de tener una rugosidad media de más de 15 micras, en el último ensayo con 10 capas, se ha obtenido menos de 2 micras lo que hace presuponer que el proceso de acabado sigue algún tipo de tendencia cuando el número de capas va en aumento. En el siguiente apartado se tratarán estos datos en una hoja de cálculo para comprobar esta tendencia.

4.7. Procesado de los datos obtenidos

En esta sección, se va a crear una tabla con los datos obtenidos de rugosidades medias y desviaciones de rugosidad para los ensayos hechos anteriormente. A continuación, se muestra dicha tabla:

Nº capas	Ra (micras)	Rz (micras)
0	15,612	87,436
4	8,131	42,990
6	5,299	27,139
8	3,594	18,532
10	1,812	12,409

Tabla 4.7.1

Para poder crear una gráfica que muestre las desviaciones de la rugosidad media a la vez que la rugosidad media, primero se ha de confeccionar una tabla con

los valores máximos y mínimos de desviaciones para cada número de capas. Para hacerlo, se utilizan las siguientes fórmulas 4.7.1-a y 4.7.1-b:

$$Desviación superior = \frac{Desviación total}{2} + Ra$$

$$Desviación inferior = -\frac{Desviación total}{2} + Ra$$

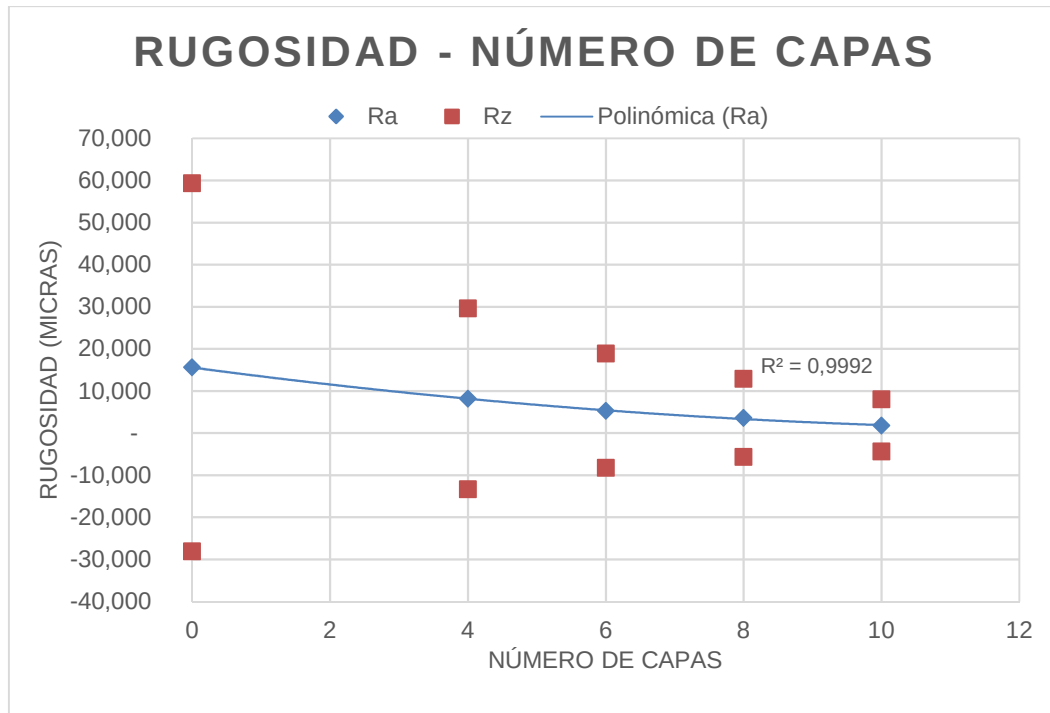
(Fórmula 4.7.1-a y 4.7.1-b)

De esta forma se obtiene la siguiente tabla 4.7.2:

Nº capas	Rz (micras)
0	59,33
	-28,106
4	29,626
	-13,364
6	18,8685
	-8,2705
8	12,86
	-5,672
10	8,0165
	-4,3925

Tabla 4.7.2

Puesto que los datos ya han sido procesados, ya se puede crear la gráfica con los datos necesarios para su posterior discusión, ésta se muestra a continuación en la gráfica 4.7.1:



Gráfica 4.7.1

Quedando el procesado de datos realizado, a continuación, se procede a su discusión.

5. DISCUSIÓN DE LOS RESULTADOS

Efectivamente como se intuía, la rugosidad media sigue una línea de tendencia polinómica de grado 2. En el gráfico se muestra el coeficiente de determinación R^2 el cuál se ajusta a 1 perfectamente. Además, se observa como la desviación va disminuyendo a medida que las capas aumentan esto hace intuir que a mayor número de capas esta desviación se conseguirá que sea casi nula, obteniendo una superficie totalmente lisa, como por ejemplo ocurre en los tratamientos con acetona.

Al observar lo que disminuye la rugosidad en función del número de capas, se puede tener en cuenta a la hora de diseñar el objeto dejando las tolerancias adecuadas para su posterior tratamiento superficial. Además, se observa que a mayor número de capas de pintura de vinilo líquido se genera una capa protectora de la pieza cada vez más resistente.

Queda demostrado que con el sistema que se ha desarrollado y ensayado se pueden obtener una mejora considerable de la rugosidad desde el número mínimo de

capas siendo este proceso muy económico. Como es lógico, a mayor número de capas, el proceso se encarece económicamente, pero dependiendo del uso que se necesite, puede buscarse el mínimo número de capas para lograr una determinada rugosidad. Además, este sistema de acabado superficial, también tiene la utilidad de dejar la superficie con un tacto bastante suave similar a la goma. Aplicable a impresiones con otros materiales que puedan tener un acabado más poroso o áspero. Ésta pintura de vinilo líquido tiene la cualidad de no necesitar imprimación para no escurrirse sobre múltiples superficies, este es uno de los motivos por el cuál este sistema funciona bien con esta clase de pintura.

6. REALIZACIÓN DE ENSAYOS EN PIEZAS PROPORCIONADAS POR LA EMPRESA VALEO ILUMINACIÓN S.A.U.

Debido al interés generado por parte del equipo de Prototipos 1 de la empresa Valero Iluminación S.A.U., se nos pidió realizar dos ensayos en piezas impresas por esta empresa que no eran válidas para su uso debido a la calidad superficial. Una de las piezas está fabricada mediante SLS con poliamida PA, con un acabado bastante áspero y la otra mediante FDM fabricada en nylon 12, teniendo bastante marcadas las capas en ciertas zonas.

Al igual que para los ensayos del trabajo, las piezas han sido enmascaradas por una parte para que pudieran verse las diferencias entre la pieza obtenida y la pieza con el proceso de mejora superficial como se muestra la pieza de SLS en la imagen 6.1.

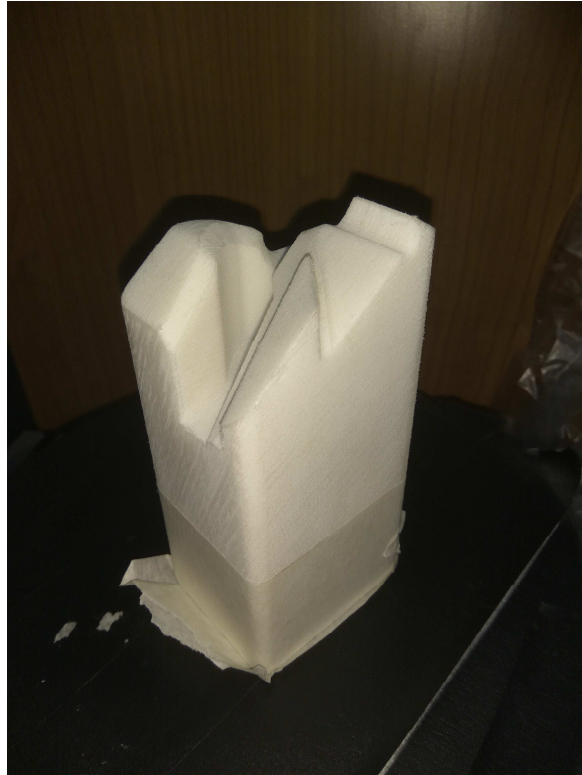


Ilustración 6.1 Pieza de SLS sin tratar

Esta vez se van a realizar ambos ensayos con una cantidad de capas de 6, con el fin de mejorar la calidad superficial, limitando eliminar las tolerancias con las que se imprimieron dichas piezas. Los resultados obtenidos son buenos, habiendo mejorado en gran medida para la pieza de SLS, eliminando las asperezas como se muestra en la ilustración 6.2 y mejorando bastante en ciertas zonas en la pieza impresa por FDM como se puede ver en la ilustración 6.3.

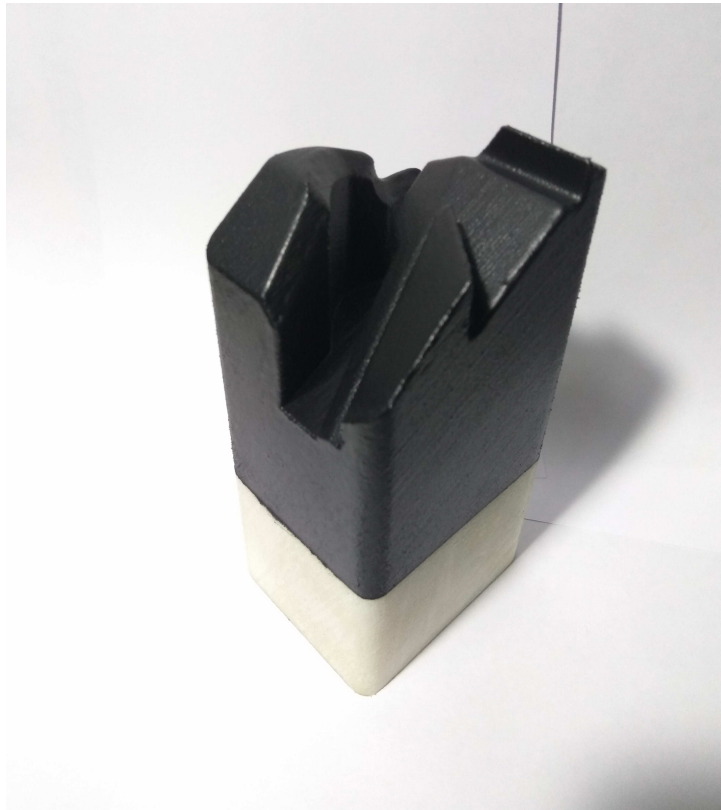


Ilustración 6.2 Pieza de SLS tras acabado superficial



Ilustración 6.3 Pieza de FDM tras acabado superficial

Además, a la pieza de FDM se le ha realizado un corte en la pintura para que se pueda ver el resultado tras retirar la capa de pintura de vinilo líquido, ya que no daña la superficie, este corte se puede observar en la ilustración 6.4, además de cómo queda el trozo de pintura de vinilo.

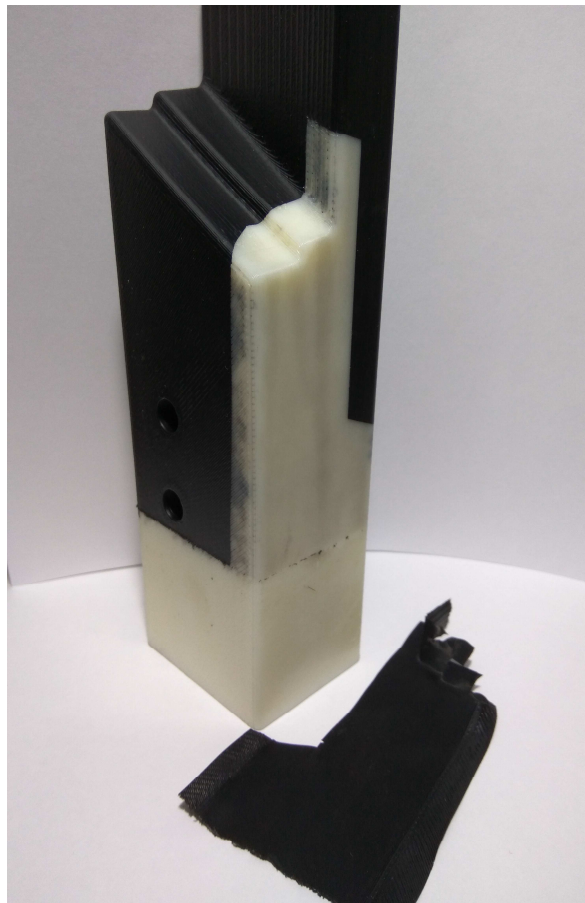


Ilustración 6.4 Pieza de FDM con un corte en la pintura de vinilo líquido

7. CONCLUSIONES

Como conclusiones sobre este Trabajo de Fin de Grado, se puede decir que ha sido una experiencia muy grata. Tras preconcebir una idea se ha podido ir de forma progresiva haciéndola realidad, en gran parte, los conocimientos adquiridos durante el Grado en Ingeniería mecánica han hecho posibles el desarrollo desde cero del sistema de acabado superficial de las piezas impresas que, como se ha podido demostrar es bastante útil además de ser barato.

En un mundo que no para de cambiar y que está en constante avance, cualquier idea preconcebida debe ser desarrollada y ver si es útil o no, puesto que en el ejemplo de este TFG ha despertado el interés de una gran empresa que utiliza la impresión 3D. Este interés apremia a la hora de realizar el trabajo, ya que sirve de aliciente además de tener unos objetivos o metas el saber que todo este trabajo puede ser útil para más personas.

Gracias a los recursos de los que disponemos a través de la Universidad de Jaén, la Escuela Politécnica Superior de Jaén y el profesorado, se ha podido llevar a cabo desde el diseño con software comercial como CATIA, el acceso a los materiales para la construcción del ensamblaje o la electrónica, hasta las medidas con los instrumentos de medida de precisión como el rugosímetro utilizado.

En un futuro, este sistema se podría mejorar por ejemplo automatizándolo totalmente, podría incluso servir en cadenas de montaje y acabado de fabricación 3D disminuyendo los tiempos de secado con pasos por hornos como las carrocerías de los vehículos.

1 ANEXO I: MODIFICACIONES DENTRO DEL CÓDIGO LIBRE MARLIN

```
#ifndef CONFIGURATION_H
#define CONFIGURATION_H

// This configuration file contains the basic settings.
// Advanced settings can be found in Configuration_adv.h
// BASIC SETTINGS: select your board type, temperature sensor type, axis
// scaling, and endstop configuration

//User specified version info of this build to display in [Pronterface,
//etc] terminal window during startup.
//Implementation of an idea by Prof Braino to inform user that any changes
//made
//to this build by the user have been successfully uploaded into firmware.
#define STRING_VERSION_CONFIG_H __DATE__ " " __TIME__ // build date and
time
#define STRING_CONFIG_H_AUTHOR "Darth Raul" //Who made the changes.

// SERIAL_PORT selects which serial port should be used for communication
// with the host.
// This allows the connection of wireless adapters (for instance) to non-
// default port pins.
// Serial port 0 is still used by the Arduino bootloader regardless of this
// setting.
#define SERIAL_PORT 0

// This determines the communication speed of the printer
#define BAUDRATE 115200
// #define BAUDRATE 115200

///// The following define selects which electronics board you have. Please
// choose the one that matches your setup
// 10 = Gen7 custom (Alfons3 Version)
// "https://github.com/Alfons3/Generation_7_Electronics"
// 11 = Gen7 v1.1, v1.2 = 11
// 12 = Gen7 v1.3
// 13 = Gen7 v1.4
// 3 = MEGA/RAMPS up to 1.2 = 3
// 33 = RAMPS 1.3 / 1.4 (Power outputs: Extruder, Bed, Fan)
// 34 = RAMPS 1.3 / 1.4 (Power outputs: Extruder0, Extruder1, Bed)
// 4 = Duemilanove w/ ATMega328P pin assignment
// 5 = Gen6
// 51 = Gen6 deluxe
// 6 = Sanguinololu < 1.2
// 62 = Sanguinololu 1.2 and above
// 63 = Melzi
// 64 = STB V1.1
// 7 = Ultimaker
// 71 = Ultimaker (Older electronics. Pre 1.5.4. This is rare)
// 8 = Teensylu
// 80 = Rumba
// 81 = Printrboard (AT90USB1286)
// 82 = Brainwave (AT90USB646)
// 9 = Gen3+
// 70 = Megatronics
// 701= Megatronics v2.0
// 702= Minitronics v1.0
// 90 = Alpha OMCA board
// 91 = Final OMCA board
```

```

// 301 = Rambo

#ifndef MOTHERBOARD
#define MOTHERBOARD 33
#endif

// This defines the number of extruders
#define EXTRUDERS 1

//// The following define selects which power supply you have. Please
choose the one that matches your setup
// 1 = ATX
// 2 = X-Box 360 203Watts (the blue wire connected to PS_ON and the red
wire to VCC)

#define POWER_SUPPLY 1

//=====
//=====Thermal
Settings =====
//=====
//
//--NORMAL IS 4.7kohm PULLUP!-- 1kohm pullup can be used on hotend sensor,
using correct resistor and table
//
//// Temperature sensor settings:
// -2 is thermocouple with MAX6675 (only for sensor 0)
// -1 is thermocouple with AD595
// 0 is not used
// 1 is 100k thermistor - best choice for EPCOS 100k (4.7k pullup)
// 2 is 200k thermistor - ATC Semitec 204GT-2 (4.7k pullup)
// 3 is mendel-parts thermistor (4.7k pullup)
// 4 is 10k thermistor !! do not use it for a hotend. It gives bad
resolution at high temp. !!
// 5 is 100K thermistor - ATC Semitec 104GT-2 (Used in ParCan) (4.7k
pullup)
// 6 is 100k EPCOS - Not as accurate as table 1 (created using a fluke
thermocouple) (4.7k pullup)
// 7 is 100k Honeywell thermistor 135-104LAG-J01 (4.7k pullup)
// 8 is 100k 0603 SMD Vishay NTCS0603E3104FXT (4.7k pullup)
// 9 is 100k GE Sensing AL03006-58.2K-97-G1 (4.7k pullup)
// 10 is 100k RS thermistor 198-961 (4.7k pullup)
//
// 1k ohm pullup tables - This is not normal, you would have to have
changed out your 4.7k for 1k
// (but gives greater accuracy and more stable
PID)
// 51 is 100k thermistor - EPCOS (1k pullup)
// 52 is 200k thermistor - ATC Semitec 204GT-2 (1k pullup)
// 55 is 100k thermistor - ATC Semitec 104GT-2 (Used in ParCan) (1k pullup)

#define TEMP_SENSOR_0 1
#define TEMP_SENSOR_1 0
#define TEMP_SENSOR_2 0
#define TEMP_SENSOR_BED 1

// Actual temperature must be close to target for this long before M109
returns success
#define TEMP_RESIDENCY_TIME 10 // (seconds)

```

```

#define TEMP_HYSTERESIS 3          // (degC) range of +/- temperatures
considered "close" to the target one
#define TEMP_WINDOW 1             // (degC) Window around target to start the
recidency timer x degC early.

// The minimal temperature defines the temperature below which the heater
will not be enabled It is used
// to check that the wiring to the thermistor is not broken.
// Otherwise this would lead to the heater being powered on all the time.
#define HEATER_0_MINTEMP 0
#define HEATER_1_MINTEMP 5
#define HEATER_2_MINTEMP 5
#define BED_MINTEMP 5

// When temperature exceeds max temp, your heater will be switched off.
// This feature exists to protect your hotend from overheating
accidentally, but *NOT* from thermistor short/failure!
// You should use MINTEMP for thermistor short/failure protection.
#define HEATER_0_MAXTEMP 260
#define HEATER_1_MAXTEMP 275
#define HEATER_2_MAXTEMP 275
#define BED_MAXTEMP 130

// If your bed has low resistance e.g. .6 ohm and throws the fuse you can
duty cycle it to reduce the
// average current. The value should be an integer and the heat bed will be
turned on for 1 interval of
// HEATER_BED_DUTY_CYCLE_DIVIDER intervals.
// #define HEATER_BED_DUTY_CYCLE_DIVIDER 4

// PID settings:
// Comment the following line to disable PID and enable bang-bang.
#define PIDTEMP
#define BANG_MAX 256 // limits current to nozzle while in bang-bang mode;
256=full current
#define PID_MAX 256 // limits current to nozzle while PID is active (see
PID_FUNCTIONAL_RANGE below); 256=full current
#ifndef PIDTEMP
  // #define PID_DEBUG // Sends debug data to the serial port.
  // #define PID_OPENLOOP 1 // Puts PID in open loop. M104/M140 sets the
output power from 0 to PID_MAX
  #define PID_FUNCTIONAL_RANGE 10 // If the temperature difference between
the target temperature and the actual temperature
                                // is more then PID_FUNCTIONAL_RANGE then
the PID will be shut off and the heater will be set to min/max. (10)
  #define PID_INTEGRAL_DRIVE_MAX 255 //limit for the integral term
  #define K1 0.95 //smoothing factor withing the PID
  #define PID_dT ((16.0 * 8.0)/(F_CPU / 64.0 / 256.0)) //sampling period of
the temperature routine

// If you are using a preconfigured hotend then you can use one of the
value sets by uncommenting it
// Ultimaker
  #define DEFAULT_Kp 19.82
  #define DEFAULT_Ki 1.31
  #define DEFAULT_Kd 75.04

// Makergear
//   #define DEFAULT_Kp 7.0
//   #define DEFAULT_Ki 0.1
//   #define DEFAULT_Kd 12

```



```

// Mendel Parts V9 on 12V
//   #define  DEFAULT_Kp 63.0
//   #define  DEFAULT_Ki 2.25
//   #define  DEFAULT_Kd 440
#endif // PIDTEMP

// Bed Temperature Control
// Select PID or bang-bang with PIDTEMPBED.  If bang-bang,
BED_LIMIT_SWITCHING will enable hysteresis
//
// uncomment this to enable PID on the bed.  It uses the same frequency
PWM as the extruder.
// If your PID_dT above is the default, and correct for your
hardware/configuration, that means 7.689Hz,
// which is fine for driving a square wave into a resistive load and does
not significantly impact you FET heating.
// This also works fine on a Fetek SSR-10DA Solid State Relay into a 250W
heater.
// If your configuration is significantly different than this and you don't
understand the issues involved, you probabaly
// shouldn't use bed PID until someone else verifies your hardware works.
// If this is enabled, find your own PID constants below.
#define PIDTEMPBED
//
// #define BED_LIMIT_SWITCHING

// This sets the max power delived to the bed, and replaces the
HEATER_BED_DUTY_CYCLE_DIVIDER option.
// all forms of bed control obey this (PID, bang-bang, bang-bang with
hysteresis)
// setting this to anything other than 256 enables a form of PWM to the bed
just like HEATER_BED_DUTY_CYCLE_DIVIDER did,
// so you shouldn't use it unless you are OK with PWM on your bed.  (see
the comment on enabling PIDTEMPBED)
#define MAX_BED_POWER 256 // limits duty cycle to bed; 256=full current

#ifdef PIDTEMPBED
//120v 250W silicone heater into 4mm borosilicate (MendelMax 1.5+)
//from FOPDT model - kp=.39 Tp=405 Tdead=66, Tc set to 79.2, argressive
factor of .15 (vs .1, 1, 10)
    #define  DEFAULT_bedKp 10.00
    #define  DEFAULT_bedKi .023
    #define  DEFAULT_bedKd 305.4

//120v 250W silicone heater into 4mm borosilicate (MendelMax 1.5+)
//from pidautotune
//   #define  DEFAULT_bedKp 97.1
//   #define  DEFAULT_bedKi 1.41
//   #define  DEFAULT_bedKd 1675.16

// FIND YOUR OWN: "M303 E-1 C8 S90" to run autotune on the bed at 90
degreesC for 8 cycles.
#endif // PIDTEMPBED

//this prevents dangerous Extruder moves, i.e. if the temperature is under
the limit
//can be software-disabled for whatever purposes by
#define PREVENT_DANGEROUS_EXTRUDE

```

```

//if PREVENT_DANGEROUS_EXTRUDE is on, you can still disable (uncomment)
very long bits of extrusion separately.
#define PREVENT_LENGTHY_EXTRUDE

#define EXTRUDE_MINTEMP 170
#define EXTRUDE_MAXLENGTH (X_MAX_LENGTH+Y_MAX_LENGTH) //prevent extrusion
of very large distances.

//=====
==
//=====Mechanical
Settings=====
//=====
==

// Uncomment the following line to enable CoreXY kinematics
// #define COREXY

// coarse Endstop Settings
#define ENDSTOPPULLUPS // Comment this out (using // at the start of the
line) to disable the endstop pullup resistors

#ifndef ENDSTOPPULLUPS
    // fine Enstop settings: Individual Pullups. will be ignored if
    ENDSTOPPULLUPS is defined
    #define ENDSTOPPULLUP_XMAX
    #define ENDSTOPPULLUP_YMAX
    #define ENDSTOPPULLUP_ZMAX
    #define ENDSTOPPULLUP_XMIN
    #define ENDSTOPPULLUP_YMIN
    // #define ENDSTOPPULLUP_ZMIN
#else
    #define ENDSTOPPULLUP_XMAX
    #define ENDSTOPPULLUP_YMAX
    #define ENDSTOPPULLUP_ZMAX
    #define ENDSTOPPULLUP_XMIN
    #define ENDSTOPPULLUP_YMIN
    #define ENDSTOPPULLUP_ZMIN
#endif

// The pullups are needed if you directly connect a mechanical endswitch
between the signal and ground pins.
const bool X_ENDSTOPS_INVERTING = true; // set to true to invert the logic
of the endstops.
const bool Y_ENDSTOPS_INVERTING = true; // set to true to invert the logic
of the endstops.
const bool Z_ENDSTOPS_INVERTING = true; // set to true to invert the logic
of the endstops.
// #define DISABLE_MAX_ENDSTOPS

// For Inverting Stepper Enable Pins (Active Low) use 0, Non Inverting
(Active High) use 1
#define X_ENABLE_ON 0
#define Y_ENABLE_ON 0
#define Z_ENABLE_ON 0
#define E_ENABLE_ON 0 // For all extruders

// Disables axis when it's not being used.
#define DISABLE_X false

```

```
#define DISABLE_Y false
#define DISABLE_Z true
#define DISABLE_E false // For all extruders

#define INVERT_X_DIR false // for Mendel set to false, for Orca set to
true
#define INVERT_Y_DIR false // for Mendel set to true, for Orca set to
false
#define INVERT_Z_DIR true // for Mendel set to false, for Orca set to
true
#define INVERT_E0_DIR false // for direct drive extruder v9 set to true,
for geared extruder set to false
#define INVERT_E1_DIR false // for direct drive extruder v9 set to true,
for geared extruder set to false
#define INVERT_E2_DIR false // for direct drive extruder v9 set to true,
for geared extruder set to false

// ENDSTOP SETTINGS:
// Sets direction of endstops when homing; 1=MAX, -1=MIN
#define X_HOME_DIR -1
#define Y_HOME_DIR -1
#define Z_HOME_DIR -1

#define min_software_endstops true //If true, axis won't move to
coordinates less than HOME_POS.
#define max_software_endstops true //If true, axis won't move to
coordinates greater than the defined lengths below.
// Travel limits after homing
#define X_MAX_POS 190
#define X_MIN_POS 0
#define Y_MAX_POS 180
#define Y_MIN_POS 0
#define Z_MAX_POS 250
#define Z_MIN_POS 0

#define X_MAX_LENGTH (X_MAX_POS - X_MIN_POS)
#define Y_MAX_LENGTH (Y_MAX_POS - Y_MIN_POS)
#define Z_MAX_LENGTH (Z_MAX_POS - Z_MIN_POS)

// The position of the homing switches
//#define MANUAL_HOME_POSITIONS // If defined, MANUAL_*_HOME_POS below
will be used
//#define BED_CENTER_AT_0_0 // If defined, the center of the bed is at
(X=0, Y=0)

//Manual homing switch locations:
#define MANUAL_X_HOME_POS 0
#define MANUAL_Y_HOME_POS 0
#define MANUAL_Z_HOME_POS 0

//// MOVEMENT SETTINGS
#define NUM_AXIS 4 // The axis order in all axis related arrays is X, Y, Z,
E
#define HOMING_FEEDRATE {2000, 2000, 100, 0} // set the homing speeds
(mm/min) [50*60,50*60, 4*60, 0]

// default settings

#define DEFAULT_AXIS_STEPS_PER_UNIT {80,80,1600,777} // default steps
per unit for ultimaker
#define DEFAULT_MAX_FEEDRATE {330, 330, 3.3, 45} // (mm/sec)
```

```

#define DEFAULT_MAX_ACCELERATION      {1500,1500,100,10000}    // X, Y, Z,
E maximum start speed for accelerated moves. E default values are good for
skeinforge 40+, for older versions raise them a lot.
                                     //(9000,9000,100,10000)
#define DEFAULT_ACCELERATION          1000    // X, Y, Z and E max
acceleration in mm/s^2 for printing moves [3000]
#define DEFAULT_RETRACT_ACCELERATION  2000    // X, Y, Z and E max
acceleration in mm/s^2 for r retracts [3000]

// Offset of the extruders (uncomment if using more than one and relying on
firmware to position when changing).
// The offset has to be X=0, Y=0 for the extruder 0 hotend (default
extruder).
// For the other hotends it is their distance from the extruder 0 hotend.
// #define EXTRUDER_OFFSET_X {0.0, 20.00} // (in mm) for each extruder,
offset of the hotend on the X axis
// #define EXTRUDER_OFFSET_Y {0.0, 5.00}  // (in mm) for each extruder,
offset of the hotend on the Y axis

// The speed change that does not require acceleration (i.e. the software
might assume it can be done instantaneously)
#define DEFAULT_XYJERK                15    // (mm/sec) [20.0]
#define DEFAULT_ZJERK                 0.4    // (mm/sec)
#define DEFAULT_EJERK                 5     // (mm/sec) [5.0]

//=====
//=====Additional
Features=====
//=====
//=====

// EEPROM
// the microcontroller can store settings in the EEPROM, e.g. max
velocity...
// M500 - stores paramters in EEPROM
// M501 - reads parameters from EEPROM (if you need reset them after you
changed them temporarily).
// M502 - reverts to the default "factory settings". You still need to
store them in EEPROM afterwards if you want to.
//define this to enable eeprom support
//#define EEPROM_SETTINGS
//to disable EEPROM Serial responses and decrease program space by ~1700
byte: comment this out:
// please keep turned on if you can.
//#define EEPROM_CHITCHAT

// Preheat Constants
#define PLA_PREHEAT_HOTEND_TEMP 200
#define PLA_PREHEAT_HPB_TEMP 50
#define PLA_PREHEAT_FAN_SPEED 255    // Insert Value between 0 and 255

#define ABS_PREHEAT_HOTEND_TEMP 240
#define ABS_PREHEAT_HPB_TEMP 100
#define ABS_PREHEAT_FAN_SPEED 200    // Insert Value between 0 and 255
(tenia 255)

//LCD and SD support
//#define ULTRA_LCD //general lcd support, also 16x2
//define DOGLCD // Support for SPI LCD 128x64 (Controller ST7565R graphic
Display Family)

```

```

#define SDSUPPORT // Enable SD Card Support in Hardware Console
//#define SDSLOW // Use slower SD transfer mode (not normally needed -
uncomment if you're getting volume init error)

//#define ULTIMAKERCONTROLLER //as available from the ultimaker online
store.
//#define ULTIPANEL //the ultipanel as on thingiverse

// The RepRapDiscount Smart Controller (white PCB)
// http://reprap.org/wiki/RepRapDiscount\_Smart\_Controller
//#define REPRAP_DISCOUNT_SMART_CONTROLLER

// The GADGETS3D G3D LCD/SD Controller (blue PCB)
// http://reprap.org/wiki/RAMPS\_1.3/1.4\_GADGETS3D\_Shield\_with\_Panel
//#define G3D_PANEL //(cambiado, le he quitado las // del principio)

//The RepRapDiscount FULL GRAPHIC Smart Controller (quadratic white PCB)
// http://reprap.org/wiki/RepRapDiscount\_Full\_Graphic\_Smart\_Controller
//
// ==> REMEMBER TO INSTALL U8glib to your ARDUINO library folder:
http://code.google.com/p/u8glib/wiki/u8glib
#define REPRAP_DISCOUNT_FULL_GRAPHIC_SMART_CONTROLLER

// The RepRapWorld REPRAPWORLD_KEYPAD v1.1
// http://reprapworld.com/?products\_details&products\_id=202&cPath=1591\_1626
//#define REPRAPWORLD_KEYPAD
//#define REPRAPWORLD_KEYPAD_MOVE_STEP 10.0 // how much should be moved
when a key is pressed, eg 10.0 means 10mm per click

//automatic expansion
#if defined (REPRAP_DISCOUNT_FULL_GRAPHIC_SMART_CONTROLLER)
    #define DOGLCD
    #define U8GLIB_ST7920
    #define REPRAP_DISCOUNT_SMART_CONTROLLER
#endif

#if defined(ULTIMAKERCONTROLLER) ||
defined(REPRAP_DISCOUNT_SMART_CONTROLLER) || defined(G3D_PANEL)
    #define ULTIPANEL
    #define NEWPANEL
#endif

#if defined(REPRAPWORLD_KEYPAD)
    #define NEWPANEL
    #define ULTIPANEL
#endif

//I2C PANELS

//#define LCD_I2C_SAINSMART_YWROBOT
#ifdef LCD_I2C_SAINSMART_YWROBOT
    // This uses the LiquidCrystal_I2C library (
https://bitbucket.org/fmalpartida/new-liquidcrystal/wiki/Home )
    // Make sure it is placed in the Arduino libraries directory.
    #define LCD_I2C_TYPE_PCF8575
    #define LCD_I2C_ADDRESS 0x27 // I2C Address of the port expander
    #define NEWPANEL
    #define ULTIPANEL
#endif

// PANELOLU2 LCD with status LEDs, separate encoder and click inputs

```

```

// #define LCD_I2C_PANEL0LU2
#ifdef LCD_I2C_PANEL0LU2
    // This uses the LiquidTWI2 library v1.2.3 or later (
    https://github.com/lincomatic/LiquidTWI2 )
    // Make sure the LiquidTWI2 directory is placed in the Arduino or
    Sketchbook libraries subdirectory.
    // (v1.2.3 no longer requires you to define PANEL0LU in the LiquidTWI2.h
    library header file)
    // Note: The PANEL0LU2 encoder click input can either be directly
    connected to a pin
    //      (if BTN_ENC defined to != -1) or read through I2C (when BTN_ENC
    == -1).
    #define LCD_I2C_TYPE_MCP23017
    #define LCD_I2C_ADDRESS 0x20 // I2C Address of the port expander
    #define LCD_USE_I2C_BUZZER //comment out to disable buzzer on LCD
    #define NEWPANEL
    #define ULTIPANEL
#endif

// Panucatt VIKI LCD with status LEDs, integrated click & L/R/U/P buttons,
// separate encoder inputs
// #define LCD_I2C_VIKI
#ifdef LCD_I2C_VIKI
    // This uses the LiquidTWI2 library v1.2.3 or later (
    https://github.com/lincomatic/LiquidTWI2 )
    // Make sure the LiquidTWI2 directory is placed in the Arduino or
    Sketchbook libraries subdirectory.
    // Note: The pause/stop/resume LCD button pin should be connected to the
    Arduino
    //      BTN_ENC pin (or set BTN_ENC to -1 if not used)
    #define LCD_I2C_TYPE_MCP23017
    #define LCD_I2C_ADDRESS 0x20 // I2C Address of the port expander
    #define LCD_USE_I2C_BUZZER //comment out to disable buzzer on LCD
    (requires LiquidTWI2 v1.2.3 or later)
    #define NEWPANEL
    #define ULTIPANEL
#endif

#ifdef ULTIPANEL
    // #define NEWPANEL //enable this if you have a click-encoder panel
    #define SDSUPPORT
    #define ULTRA_LCD
    #ifdef DOGLCD // Change number of lines to match the DOG graphic display
        #define LCD_WIDTH 20
        #define LCD_HEIGHT 5
    #else
        #define LCD_WIDTH 20
        #define LCD_HEIGHT 4
    #endif
#else //no panel but just lcd
    #ifdef ULTRA_LCD
        #ifdef DOGLCD // Change number of lines to match the 128x64 graphics
        display
            #define LCD_WIDTH 20
            #define LCD_HEIGHT 5
        #else
            #define LCD_WIDTH 16
            #define LCD_HEIGHT 2
        #endif
    #endif
#endif

```

```

// Increase the FAN pwm frequency. Removes the PWM noise but increases
heating in the FET/Arduino
//#define FAST_PWM_FAN

// M240 Triggers a camera by emulating a Canon RC-1 Remote
// Data from: http://www.doc-diy.net/photo/rc-1\_hacked/
// #define PHOTOGRAPH_PIN 23

// SF send wrong arc g-codes when using Arc Point as fillet procedure
//#define SF_ARC_FIX

// Support for the BariCUDA Paste Extruder.
//#define BARICUDA

/*****\
*
* R/C SERVO support
*
* Sponsored by TrinityLabs, Reworked by codexmas
*
*****/

// Number of servos
//
// If you select a configuration below, this will receive a default value
// and does not need to be set manually
// set it manually if you have more servos than extruders and wish to
manually control some
// leaving it undefined or defining as 0 will disable the servo subsystem
// If unsure, leave commented / disabled
//
#define NUM_SERVOS 3

#include "Configuration_adv.h"
#include "thermistortables.h"

#endif //__CONFIGURATION_H

```

2 Bibliografía

(2016). Obtenido de UNI: <http://www.uni.edu/>

3dnatives. (2016). Obtenido de www.3dnatives.com

3dpinfo. (2016). Obtenido de <http://www.3dpinfo.com/>

3dprinting. (2016). Obtenido de <http://3dprinting-blog.com/>

3dprintingindustry. (2016). Obtenido de <http://.com/>

3dwithus. (2016). Obtenido de <http://3dwithus.com/>

Adafruit. (2016). Obtenido de <https://learn.adafruit.com/>

all3dp.com. (2016). Obtenido de <https://all3dp.com/best-sites-free-stl-files-3d-printing/>

Autozona. (2016). Obtenido de <http://autozona.com/>

Bcn3dtechnologies. (2016). Obtenido de <https://www.bcn3dtechnologies.com>

Bell, C. (2014). *Maintaining and Troubleshooting Your 3D Printer*. Berkely, CA, USA: Apress.

Forbes. (2016). Obtenido de www.forbes.com

Frank day motors. (2016). Obtenido de <http://www.frank-day-motors.com/>

Gigaom. (2016). Obtenido de <https://gigaom.com>

Google. (2016). Obtenido de www.google.es

<https://www.sculpteo.com/es/glosario/prototipado-rapido-3d/>. (s.f.).

Impresoras3d. (2016). Obtenido de www.impresoras3d.com

Imprimalia3d. (2016). Obtenido de <http://www.imprimalia3d.com>

Kikailabs. (2016). Obtenido de <http://kikailabs.com.ar/>

Machinedesign. (2016). Obtenido de <http://machinedesign.com/>

Nextiafenix. (2016). Obtenido de <http://www.nextiafenix.com/>

Plas-techfab. (2016). Obtenido de <http://plas-techfab.com/>

Poveda Martínez, Santiago. (s.f.). *UPM*. Obtenido de http://ocw.upm.es/expresion-grafica-en-la-ingenieria/ingenieria-grafica-metodologias-de-diseno-para-proyectos/Teoria/LECTURA_COMPLEMENTARIA/MATERIALES/acabados.pdf

RepRap. (2016). Obtenido de www.RepRap.org

Sculpteo. (2016). Obtenido de (<https://www.sculpteo.com/es/glosario/prototipado-rapido-3d/>, s.f.)