

PROYECTO DE EXTENSIÓN

Diseño Industrial-FAyD-UNaM

**2022 | PRODUCCIÓN
DIGITAL ABIERTA**

**REPARAR, CORREGIR,
PROBAR Y DOCUMENTAR**

©

Informe proyecto de extensión 2022

Proyecto de extensión

Es a partir de las diferentes experiencias desarrolladas en el laboratorio DAyT/producción digital abierta que se propone la realización de un taller de producción colectiva del paquete tecnológico para la creación de una impresora aditiva cerámica. Esta idea se ubica bajo la denominación Tecnologías 4.0. proponen nuevos significados aglutinando diversos y múltiples procesos de producción, algunos ya consolidados y otros en proceso de desarrollo a través de innovaciones disruptivas o sostenibles. Lo que propone la incorporación de una mayor flexibilidad e individualización de los procesos de fabricación, facilitando la individualización de los objetos/productos con la optimización de los procesos que agregan valor, así como una gran integración entre docentes/investigadores, becarios, graduados de las carreras de la FAyD. Nos interesa acercarnos a un concepto de diseño táctico.

La modalidad que se va a emplear es de tipo taller de producción, donde se van a desarrollar las diferentes actividades hasta alcanzar el objetivo, construir la impresora aditiva cerámica –IAC-, generar la documentación técnica para poder reproducir la misma y tomar conocimiento de los diferentes saberes -electrónica, mecánica, diseño asistido por ordenador, manufactura asistida por ordenador, programación arduino-.

El proyecto va a estar integrado por diferentes talleres específicos para resolver una instancia del paquete tecnológico de la IAC, que se llevarán adelante en el segundo cuatrimestre.

- 1.-Introducción de las tecnologías 4.0 y su aplicación a la manufactura. Reconocimiento de los componentes de los sistemas de producción aditiva y procesos de manufactura asistidos por ordenador –CAM-. (4 clases)
- 2.-Reconocimiento de los diferentes sistemas que integran una IAC. Pautas de montaje y armado de la mecánica y electrónica. (2 clases)
- 3.-Re-Diseño de componentes y piezas de la impresora. Modelado orientado a la producción aditiva de piezas (4 clases)
- 4.-Producción digital de elementos mecánicos destinados a la IAC mediante técnicas aditivas de material. (4 clases)
- 5.-Programación y puesta a punto de los diferentes sistemas. Arduino+Marlin. (2 clases)
- 6.- Pruebas y testeos. (4 clases)

Total: 20 clases/duración de los encuentros de 2 horas, 40 hrs..

Cronograma

Inicio:	
Corrección:	
Entrega:	

24 de abril

TALLER DE EXTENSIÓN 2021/2022

Producción Digital Abierta///

Hacer, documentar los procesos de puesta a punto de sistemas de producción aditivos.

Guía temática del taller.

Copyleft 2018 - 2021 el equipo de extensión.

Resumen.

Este documento contiene las instrucciones para participar activamente en el proyecto. Además, contiene enlaces a otras fuentes de información, así como información de cómo obtener los avances y últimas actualizaciones.

Tabla de contenidos

REPARAR, CORREGIR, PROBAR Y DOCUMENTAR

The banner features a central image of various tools and components on a green grid background. Text elements include: 'REPARAR, CORREGIR, PROBAR Y DOCUMENTAR UNA IMPRESORA 3D FDM' in a black box; 'Extensión' in a red box; a schedule table with columns for 'Inicio', '//22//', 'Cierra', and 'lunes'; a list of 'CAPACITADORES' (D.I. Javier Balcaza, D.I. Pedro Conalbi, D.I. Nazareno Viera, D.I. Zasha Kozlarski, D.I. Iván Riquelme); 'CURSO PRESENCIAL' in a black box; and logos for 'DR Lab' and 'FAYD UNAM' in the bottom right corner.

Inicio	//22//	Cierra	lunes
Abril	encuentros	octubre	15 a 18hs

4/04/2022

Extensión

CAPACITADORES:
D.I. Javier Balcaza
D.I. Pedro Conalbi
D.I. Nazareno Viera
D.I. Zasha Kozlarski
D.I. Iván Riquelme

CURSO PRESENCIAL

DR Lab FAYD UNAM

1. Bienvenido al curso

INTRODUCCIÓN

- 1.1. ¿Qué es la manufactura aditiva?
- 1.2. ¿Qué equipamiento se necesita?
- 1.3. ¿Cómo es el funcionamiento de una impresora FDM o aditiva?
- 1.4. ¿Cómo esta compuesta, qué sistemas integra?
- 1.5. ¿Qué es el lenguaje G-CODE?
- 1.5. Programación G-CODE, software libre.
- 1.6. ¿Qué es la plataforma Arduino?
- 1.7. Sobre copyrights y licencias de software

2. Reconocimiento de un dispositivo

2.1. Arquitecturas del sistema.

2.2. Compatibilidad con otros sistemas.

2.3. Plataforma Arduino, programación, modificaciones.

2.4. Mecánica: delta, cartesiana, core o scara. Tipo de transmisión, desgaste, mantenimiento.

2.5. Electrónica, fuente, placa, configuración de los fines de carrera tipo de sensor de temperatura, resistencia y controlador lcd.

2.5.1 Verificación de intensidad de corriente que suministran los driver a los motores NEMA.

2.5.1.1 Motores NEMA.

buscar datasheet del modelo de motor NEMA correspondiente, para conocer los valores nominales de intensidad.

**los motores paso a paso, también conocidos como steppers, se encuentran dentro de los motores de corriente continua, con la diferencia que se pueden emplear a velocidades de giro y potencias bajas gracias al controlador o driver. Además de poder controlar el movimiento del eje con una alta precisión, se puede controlar el ángulo de giro y la velocidad. Esta función depende de los pasos que posea el motor, generalmente los pasos angulares son: 1.8°, 5.625°, 7.5°, 11.25°, 18°, 45°, y 90°. Los NEMA 17, que son los que vamos a emplear vienen con un paso angular de 1,8, por lo que posee 200 pasos. Y poseen dos polos (dos bobinas) (<https://www.hwlibre.com/motor-paso-a-paso/>) También se pueden dividir los pasos, micro pasos / Microsteps, depende del tipo de driver que usemos y de la programación en arduino.*

Paso completo = Mayor torque, movimientos bruscos y mayor ruido.

Medio paso = Menor torque, movimientos suaves y menor ruido.

Micropasos = Menor torque aun, movimientos mucho mas suaves y mucho menos ruido.

2.5.1.2 Driver, tipos y manejo de corriente.

modelos:

•polulu A4988 (Rcs 0.050 Ω / 0.068 Ω) Corriente nominal 1.5 A – Corriente Max 2 A – Voltaje 8-35V – Microsteps 16 – Rs 0,05/0,1/0,2

•polulu DVR8825 (Rcs 0,1Ω) Corriente nominal 2 A – Corriente Max 2.5 A – Voltaje 8-45V – Microsteps 32 – Rs 0,1

el valor máximo de limitación de corriente se establece en la selección de Rs y el voltaje en el pin VREF. La función de transconductancia se aproxima por el valor máximo de limitación de corriente. IMax (A), que se establece por: $I_{Max} = V_{ref} / (8 \times R_{cs}) - A4988$ $I_{Max} = V_{ref} / (5 \times R_{cs}) - DVR8825$

Donde Rs es el valor de la resistencia de sensibilidad del driver y Vref es el voltaje de entrada en el pin REF. Hay que verificar, además en el datasheet con que porcentaje energiza el motor con un paso completo o como se configure (full-step, ...), en el A4988 lo realiza con un 70%.

(<http://www.javierarnedo.com/driver-a4988-nema-17-ajuste-voltaje-referencia/>)

(<https://www.diarioelectronicohoy.com/blog/descripcion-del-driver-a4988>)

(<https://www.hwlibre.com/nema-17/>)

(<http://carlini.es/manejar-un-motor-stepper-con-un-driver-a4988-y-arduino/>)

(<https://www.luisllamas.es/motores-paso-paso-arduino-driver-a4988-drv8825/>)

2.6. Slicer: proceso para generar el G-code por capas, diferentes software.

Laminadores	Cura. Slic3r. Simplify3D. PrusaSlicer.
Calibración.	Pronterface. Repetier. Printun.
CAD.	Meshmixer (reparación + edición). Fusion360. SolidWorks. Rhinoceros. 3DsMax. Cinema4D. Blender. Sketchup. Freecad. Openscad. Thinkercad

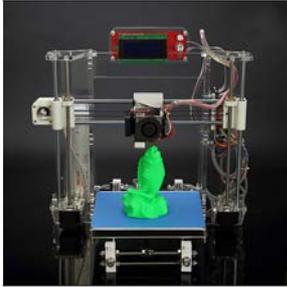
2.7. Forma comercial de los consumibles: filamento, pasta.

2.8. Post procesado de las piezas obtenidas.

2.9. Periféricos y otro hardware.



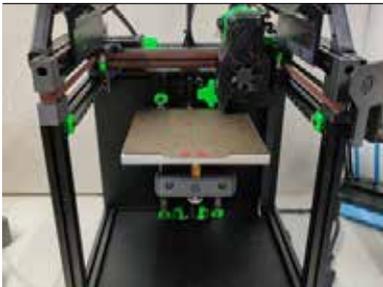
MENDEL
CARTESIANA



PRUSA
CARTESIANA



CARTESIANA



CORE XY



DELTA



SCARA

Arduino

Es una plataforma de hardware libre, por lo que se puede copiar, pegar, compartir, publicar libremente todo lo que se realice bajo esta plataforma. La plataforma se basa en un microcontrolador y en conjunto con el entorno de desarrollo es posible crear proyectos electrónicos de mando y control.

Un microcontrolador es un ordenador integrado en un chip, posee memoria, cpu y periféricos. Pueden realizar funciones simples o complejas, desde controlar un objeto a sistemas de objetos.

Al microcontrolador hay que sumarle el software, entorno de desarrollo, que es el software que facilita la escritura del código o programa y la carga del mismo a la placa. Dicho programa o código nos va a permitir dar las ordenes para controlar un objeto.

Programa:

En la primera sección del programa, `setup()`, se define la configuración del mismo:

- .codificar sentencias
- .iniciar los puertos de la placa
- .configurar las comunicaciones
- .definir parámetros que dependerán de cada programa que se desarrolle

En la segunda sección del programa, `loop()`, se codifica el programa que se ejecutará de forma cíclica

www.Arduino.Cc

(Reprap es la primera máquina de fabricación Autorreplicante de propósito general de la humanidad)

#pines digitales:

son las terminales que permiten relacionar la placa con otros objetos (datos de entrada y/o salida).

Una señal digital posee dos estados:

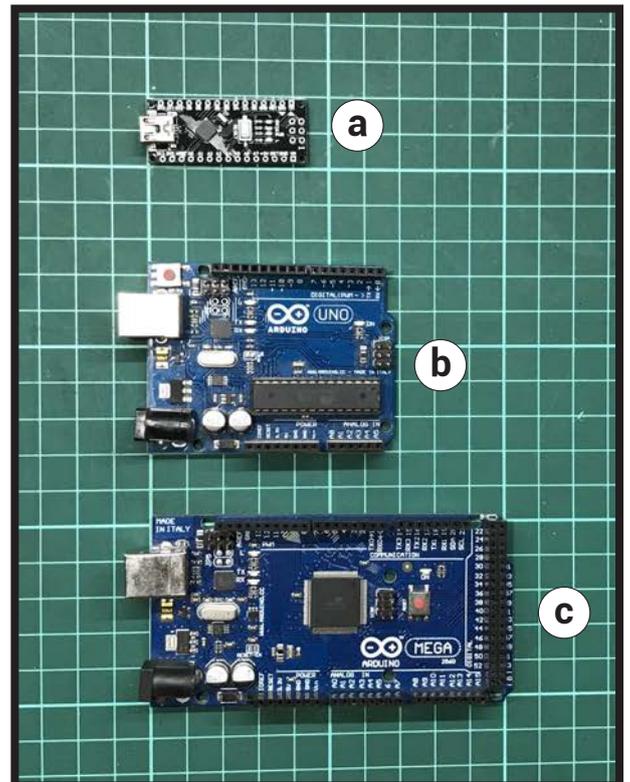
- 0 (low) indica una tensión de 0v en el pin
- 1 (high) indica una tensión de 5v en el pin

#pines analógicos:

el estado es variable, acepta un rango de diferentes valores de voltaje. El estado va de 0 a 5v.

#pines de alimentación:

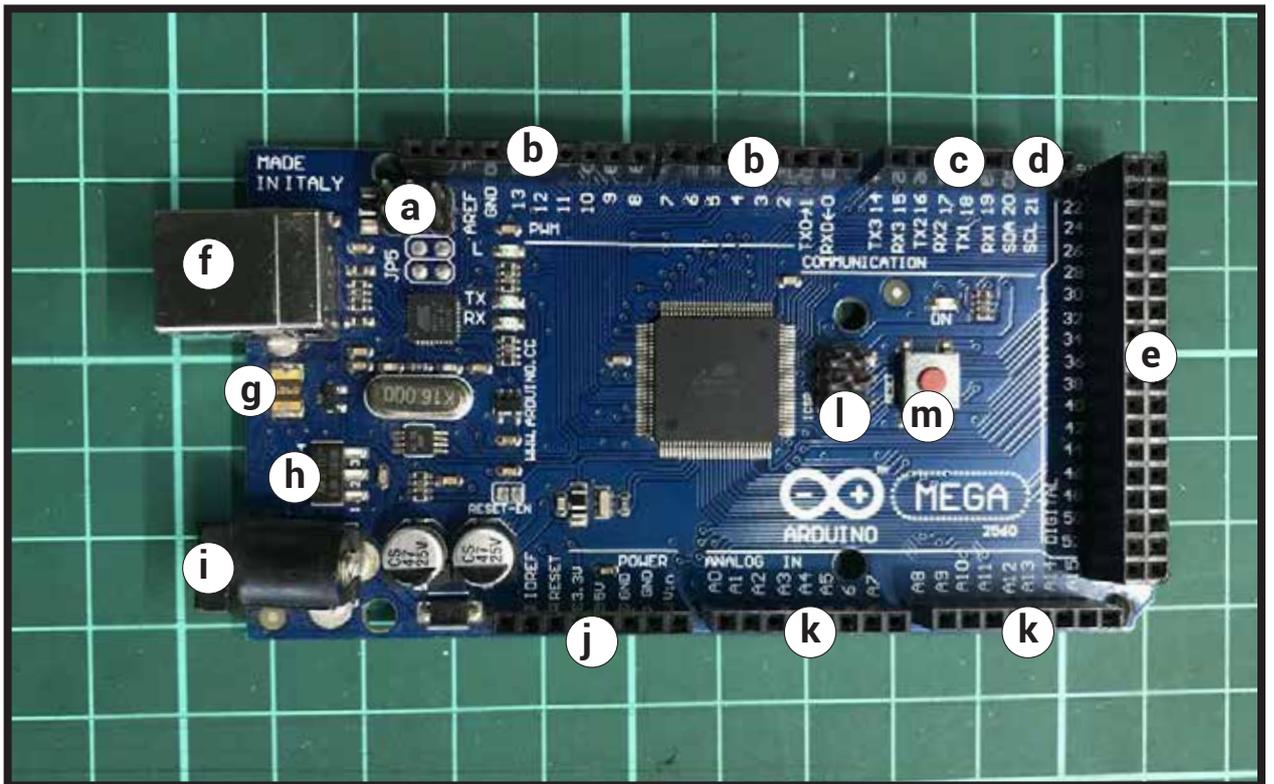
son las terminales que permiten a otros dispositivos obtener corriente para funcionar, pueden ser valores de 3.3v o 5v o 0v(gnd).



• a.Arduino NANO
#

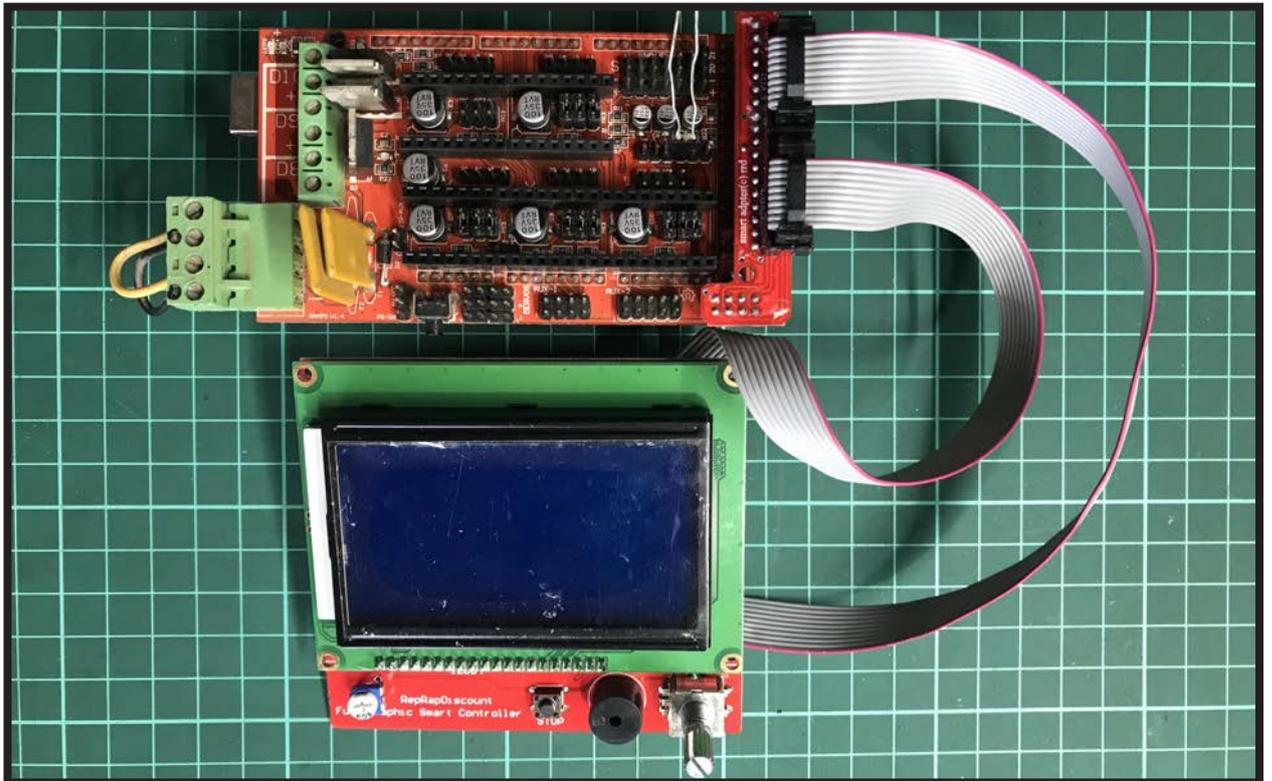
• b.Arduino UNO
ATM328
#microcontrolador atmega328p a 16mhz, puerto usb, alimentación 5v, sram de 2kb, 32kb de flash, 1kb de eeprom, 14 pines digitales y 6 con pwm y 6 salidas analógicas.

• c.Arduino MEGA
ATM2560
#microcontrolador atmega2560 a 16mhz, 5v, 4kb de eeprom, 8kb de sram y 256kb de flash, 54 pines digitales, 12 con pwm y 16 analógicas



ARDUINO MEGA /// ATM2560

- a. Interfaz usb-serie
- b. Salidas pwm (tipo modulación de ondas por pulso)
- c. comunicación serie
- d. I2c
- e. Entradas y salidas digitales de uso general
- f. Usb_tipo_b
- g. Fusible usb
- h. Regulador de voltaje
- i. Conector jack, 7/12v
- j. Alimentación
- k. Entradas analógicas
- l. Icsp
- m. Reset

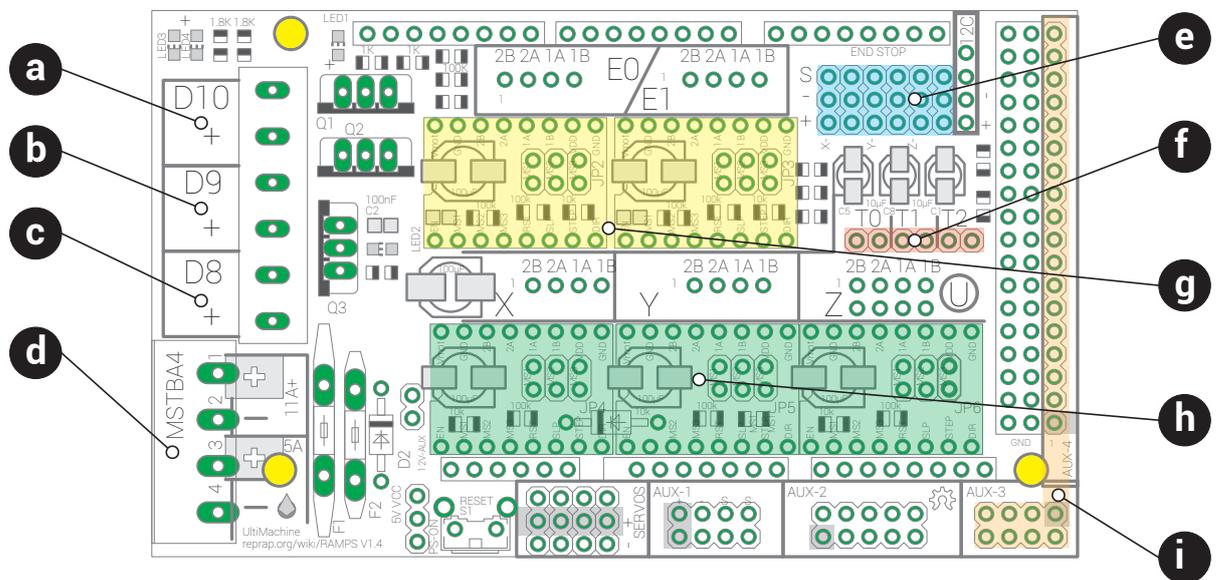


RAMPS_V1.4

#NOS PERMITE REALIZAR LAS CONEXIONES ENTRE LA PLACA ARDUINO Y LOS PERIFERICOS, ENTRADAS Y SALIDAS DE LOS DISTINTOS DISPOSITIVOS QUE INTEGRA LA IMPRESORA 3D.

RepRap Arduino Mega Pololu Shield, o RAMPS para abreviar. Está diseñado para adaptarse a toda la electrónica necesaria para un RepRap en un paquete pequeño a bajo costo. RAMPS interactúa un Arduino Mega con la poderosa plataforma Arduino MEGA y tiene mucho espacio para la expansión. El diseño modular incluye controladores paso a paso enchufables y electrónica de control del extrusor en un escudo Arduino MEGA para facilitar el servicio, el reemplazo de piezas, la capacidad de actualización y la expansión. Además, se pueden agregar varias placas de expansión Arduino al sistema siempre que la placa RAMPS principal se mantenga en la parte superior de la pila.

- a. Potencia resistencia extrusor
- b. Potencia fan de capa
- c. Potencia cama caliente
- d. Conexion de potencia desde la fuente de alimentacion
- e. Finales de carrera
- f. Conexiones termistor control de temperatura
- g. Conexiones motor steppers para los extrusores
- h. Conexiones motor steppers para los ejes x; y; z
- i. Conexiones pantalla/display aux-3+aux-4



• 1.FUENTE

(ES IMPORTANTE VERIFICAR SENTIDO DE LA CORRIENTE)

VOLT: 12V CC (CORRIENTE CONTINUA)
 AMPERE: SUMAR LOS CONSUMOS INDIVIDUALES.
 CAMA CALIENTE: 11A (130W)
 (PCB HEATBED MK2A)
 RESISTENCIA: 3.3A (40W)
 MOTOR STEPPER: 1.4A*5= 7A
 FAN: 150mA
 CONTROLADOR: 200mA

TOTAL: 21.65A

• 2.CAMA CALIENTE

(VERIFICAR QUE CORRESPONDA CON EL VALOR NOMINAL DE TENSION DE LA FUENTE.)

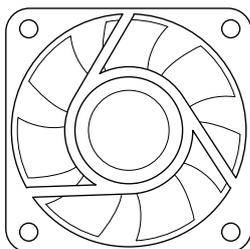


PRECAUCIÓN:
 ¡ANTES DE TOCAR, APAGAR LA SAMA TÉRMICA Y ESPERAR AL MENOS 10 MINUTOS!
 ¡TODAVÍA CALIENTE DESPUÉS DE APAGADO!
 ¡NO LO DEJES DESATENDIDO!
 ¡ALEJATE DE LOS NIÑOS!

ADVERTENCIA: ¡TODAVÍA CALIENTE DESPUÉS DE QUE EL LED SE APAGUE!

• 3.POTENCIA FAN DE CAPA

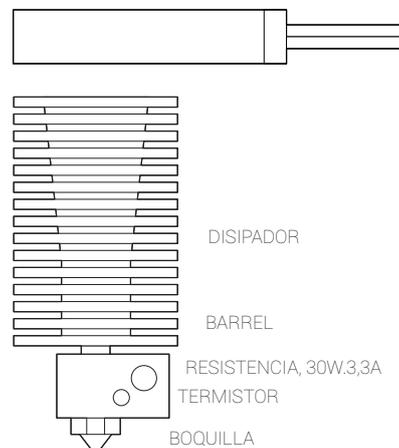
(VERIFICAR QUE CORRESPONDA CON EL VALOR NOMINAL DE TENSION DE LA FUENTE Y SENTIDO.)



• 4.POTENCIA EXTRUSOR

(VERIFICAR QUE CORRESPONDA CON EL VALOR NOMINAL DE TENSION DE LA FUENTE.)

TIPO CARTUCHO; 30W; 12V_CC



• 5.FINALES DE CARRERA

(ES IMPORTANTE IDENTIFICAR EL TIPO)

SWITCH/FINAL DE CARRERA MECÁNICO 3PIN

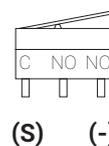
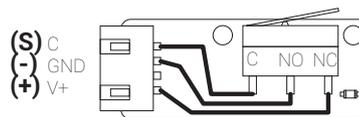
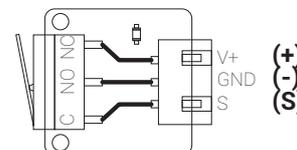
C:COMUN
 NO:NORMAL ABIERTO
 NC:NORMAL CERRADO

SWITCH/FINAL DE CARRERA MECÁNICO 4PIN

C:COMUN
 NO:NORMAL ABIERTO
 NC:NORMAL CERRADO

SWITCH/FINAL DE CARRERA MECANICO 2PIN

C:COMUN
 NO:NORMAL ABIERTO
 NC:NORMAL CERRADO

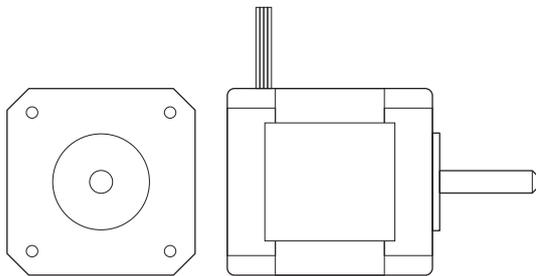


• **6.CONTROL DE TEMPERATURA**
(ES IMPORTANTE IDENTIFICAR EL VALOR DEL TERMISTOR; GENERALMENTE 100K)

T0>>>EXTRUSOR
T1>>>CAMA CALIENTE

• **7.8.CONEXIONES MOTOR STEPPERS**

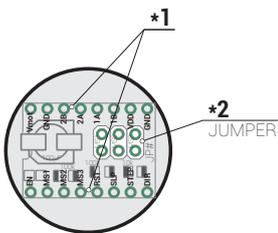
MOTOR NEMA 17
2 FASES/2 BOBINAS
4 CABLES, UN PAR POR BOBINA.



DRIVER
(VERIFICAR ORIENTACIÓN DEL DRIVER SEGÚN ENTRADAS)

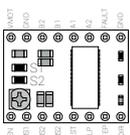
2 FASES/2 BOBINAS
4 CABLES, UN PAR POR BOBINA.

DRIVER SECTOR RAMPSV1.4



*1_POSICIÓN

DVR8825



*2_MICROPASOS

JP1.2.3.4.5.6

1	2	3	
no	no	no	full step
yes	no	no	half step
no	yes	no	1/4 step
yes	yes	no	1/8 step
no	no	yes	1/16 step
yes	yes	yes	1/32 step

JUMPERS

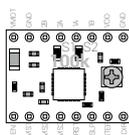
*3_CÁLCULO de AJUSTE V

$$V_{ref} = I_{TripMax} / 2$$

PRYEXT.DI.FAYD.UNaM

*1_POSICIÓN

A4988///POLOLU



*2_MICROPASOS

JP1.2.3.4.5.6

1	2	3	
no	no	no	full step
yes	no	no	half step
no	yes	no	1/4 step
yes	yes	no	1/8 step
no	no	yes	1/16 step
yes	yes	yes	1/32 step

JUMPERS

*3_CÁLCULO de AJUSTE V

$$I_{TripMax} = V_{ref} / (8 * R_s)$$

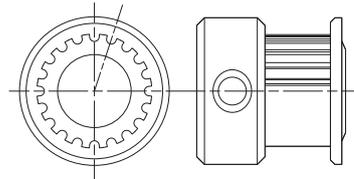
$$V_{ref} = I_{TripMax} * 8 * R_s$$

$$V_{ref} = 1.4A * 8 * 0.1 = 1,12V$$

Transmisión por medio de una polea gt2 de 20 dientes, con un paso de 2mm.; por lo que una vuelta representa (20dientes*2mm) 40mm de movimiento.

Para averiguar los step*mm, se debe dividir 200 pasos/40mm =

$$\text{Steps*mm} = 5$$

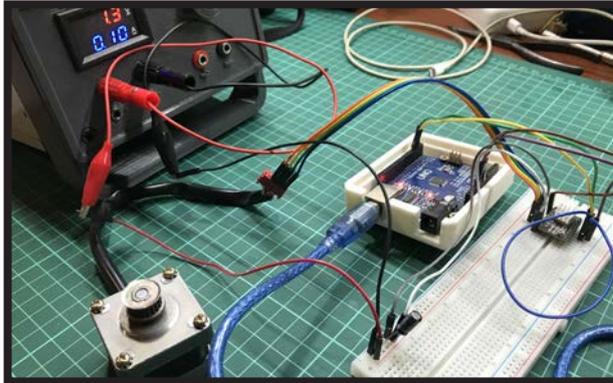


• **9..DISPLAY**

PANTALLA///AUXILIAR 3+ AUXILIAR 4
REPRAP_DISCOUNT_FULL_GRAPHIC_SMART_CONTROLLER



• BANCO DE PRUEBAS PARA MOTOR NEMA#17
ARDUINO UNO + DRIVER A4988



// Defina las conexiones de los pines y los pasos del motor por revolución

```
const int dirPin = 2;
const int stepPin = 3;
const int stepsPerRevolution = 200;
```

```
void setup()
```

```
{
    // Declarar pines como salidas
    pinMode(stepPin, OUTPUT);
    pinMode(dirPin, OUTPUT);
}
```

```
void loop()
```

```
{
    // Establecer la dirección del motor en el
    sentido de las agujas del reloj
    digitalWrite(dirPin, HIGH);
```

```
    // Girar el motor lentamente
    for(int x = 0; x < stepsPerRevolution; x++)
    {
        digitalWrite(stepPin, HIGH);
        delayMicroseconds(2000);
        digitalWrite(stepPin, LOW);
        delayMicroseconds(2000);
    }
```

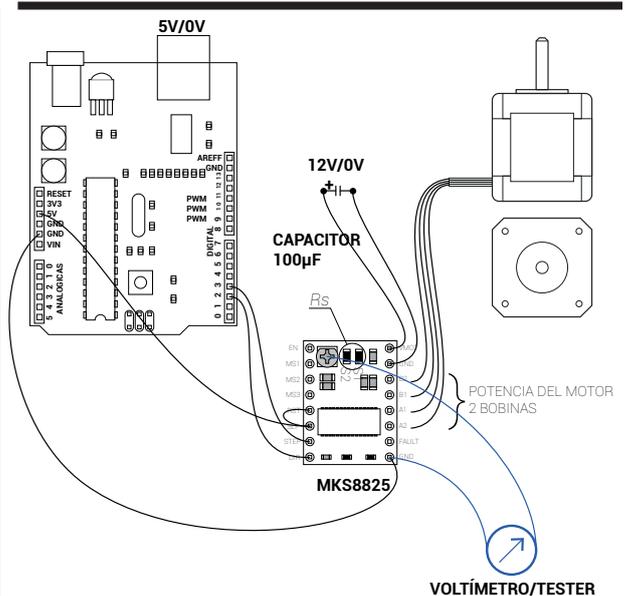
```
    delay(1000); // Wait a second
```

```
    // Establecer la dirección del motor en sentido
    contrario a las agujas del reloj
    digitalWrite(dirPin, LOW);
```

```
    // Girar el motor rápidamente
    for(int x = 0; x < stepsPerRevolution; x++)
    {
        digitalWrite(stepPin, HIGH);
        delayMicroseconds(1000);
        digitalWrite(stepPin, LOW);
        delayMicroseconds(1000);
    }
```

```
    delay(1000); // Wait a second
```

```
}
```



```

/* control_pap_a4988.ino
 * Control velocidad y sentido giro de motor Stepper con pololu
A4988
 */

```

```

int steps = 9; // pin step 9
int direccion = 3; // pin direccion 3
int potenciometro; // lectura del potenciometro
int boton = 7; // pin pulsador

```

```

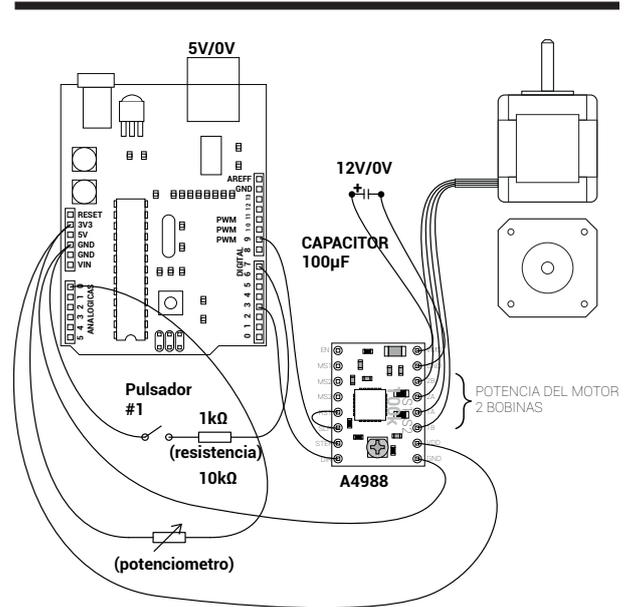
void setup() {
 // inicializamos pines como salidas.
 pinMode(steps, OUTPUT);
 pinMode(direccion, OUTPUT);
 // inicializamos pin como entrada
 pinMode(boton, INPUT);
}

```

```

void loop() {
 int sentido = digitalRead(boton); // leemos el boton de
 direccion
 digitalWrite(direccion, sentido); // cambiamos de dirección
 según pulsador
 potenciometro = analogRead(A0); // leemos el
 potenciometro
 potenciometro = map(potenciometro,0,1024,900,2500);//
 adaptamos el valor leído a un retardo
 digitalWrite(steps, HIGH); // Aquí generamos un flanco de
 subida LOW - HIGH
 delayMicroseconds(5); // Pequeño retardo para formar
 el pulso en STEP
 digitalWrite(steps, LOW); // y un flanco de bajada, el
 A4988 avanzara un paso del motor
 delayMicroseconds(potenciometro); // generamos un retardo
 con el valor leído del potenciómetro
}

```



<i>Modelo</i>	A4988	DRV8825
<i>Color</i>	Verde o Rojo	Morado
<i>Intensidad máxima</i>	2A	2.5A
<i>Tensión máxima</i>	35V	45V
<i>Microsteps</i>	16	32
<i>Rs típico</i>	0.05, 0.1 o 0.2	0.1

Fórmulas **$I_{max} = V_{ref} / (8 * R_s)$ $I_{max} = V_{ref} / (5 * R_s)$**
 $V_{ref} = I_{max} * 8 * R_s$ $V_{ref} = I_{max} * 5 * R_s$

<i>Modelo</i>	Rs	Fórmula reducida
<i>A4988</i>	50	$I_{max} = 0,625 * V_{ref}$
<i>A4988</i>	100	$I_{max} = 1,25 * V_{ref}$
<i>A4988</i>	200	$I_{max} = 2,2 * V_{ref}$
<i>DRV8825</i>	100	$I_{max} = 2 * V_{ref}$

<i>RESOLUCIÓN</i>		Pines MS1, MS2 y MS3		
<i>A4988</i>	<i>DRV8825</i>	MODE1	MODE2	MODE3
<i>Full step</i>	<i>Full step</i>	Low	Low	Low
<i>1/2 step</i>	<i>1/2 step</i>	High	Low	Low
<i>1/4 step</i>	<i>1/4 step</i>	Low	High	Low
<i>1/8 step</i>	<i>1/8 step</i>	High	High	Low
-	<i>1/16 step</i>	Low	Low	High
-	<i>1/32 step</i>	High	Low	High
-	<i>1/32 step</i>	Low	High	High
<i>1/16 step</i>	<i>1/32 step</i>	High	High	High

Dejar todos los pines desconectados dará lugar a usar modo Full Step

3. Rediseño

Modelado de una pieza con la finalidad de reproducir mediante un FDM.

4. Producción de piezas parte

5. Programación MARLIN

Marlin es un enorme programa en C++ compuesto por muchos archivos, pero aquí solo hablaremos de los dos archivos que contienen todas las opciones de configuración en tiempo de compilación de Marlin.

En programación de arduino las dos barras inclinadas significan que la línea que sigue no va a ser leída por el programador. Se puede usar para realizar un comentario o desactivar una acción.

5.0.1 Buscar el programa: Configuration.h contiene las principales opciones de configuración para el funcionamiento correcto de la maquina. Como ser: selección de hardware, idioma y controlador, y la configuración de las características y componentes más comunes.

5.1. Versión de configuración

es la versión de la configuración y no compilara si no esta definida.

```
#define CONFIGURATION_H_VERSION 020005
```

5.2. Versión del firmware

muestra el mensaje de inicio de marlin en el display (si lo posee), ayuda a identificar el tipo de configuración personalizada del marlin.

```
#define STRING_CONFIG_H_AUTHOR "(none, default config)"
```

```
#define SHOW_BOOTSCREEN
#define SHOW_CUSTOM_BOOTSCREEN
#define CUSTOM_STATUS_SCREEN_IMAGE
```

5.3. Información del hardware

define el indice integrado que empleara para la comunicación.

```
#define SERIAL_PORT 0
```

por default empleara el 0 que es el puerto usb, si hace falta se puede configurar otro puerto.

5.3.1. BAUDERATE

Define la tasa de transferencia de datos, la velocidad de transferencia debe ser la más rápida posible, 115200 es una buena relación entre velocidad y estabilidad de la conexión.

```
#define BAUDRATE 115200
```

5.3.2. BLUETOOTH

Habilita la transferencia de datos mediante bluetooth para placas del tipo AT90USB.

```
#define BLUETOOTH
PRYEXT.DI.FAyD.UNaM
```

5.3.3. MOTHERBOARD

La placa base es uno de las configuraciones mas importantes, ya que el firmware necesita saber que tipo de placa para establecer las conexiones y configuración correcta para establecer las funciones correctas. En el programa boards.h se encuentran las placas mas recientes con sus denominaciones, buscar de acuerdo a la placa que se va a instalar.

```
#define MOTHERBOARD BOARD_RAMPS_14_EFB
```

algunos ejemplos:

```
Arduino
BOARD_RAMPS_14_EEF
RAMPS v1.4 (Extruder, Extruder, Fan)
pins_RAMPS_14.h
```

```
Arduino
BOARD_RAMPS_14_EFB
RAMPS v1.4 (Extruder, Fan, Bed)
pins_RAMPS_14.h
```

```
Arduino
BOARD_RAMPS_14_EFF
RAMPS v1.4 (Extruder, Fan, Fan)
pins_RAMPS_14.h
```

```
Arduino
BOARD_RAMPS_14_SF
RAMPS v1.4 (Spindle, Controller Fan)
pins_RAMPS_14.h
```

```
Arduino
BOARD_RAMPS_OLD
RAMPS v1.0, v1.1, v1.2
pins_RAMPS_OLD.h
```

5.3.4. Nombre de la maquina personalizado permite agregar un nombre a la maquina, el cual aparecerá en el display.

```
//#define CUSTOM_MACHINE_NAME "3D Printer"
```

5.3.5. UUID de la maquina

es la identificación de la maquina cuando a esta se la conecta en red.

```
//#define MACHINE_UUID
"00000000-0000-0000-0000-000000000000"
```

5.4. Información del extrusor.

5.4.1 Cantidad de extrusores
establece la cantidad de motores paso a paso que dispone la maquina, el marlin asume que todos los extrusores se mueven en grupo. Se puede setear hasta un máximo de 6 extrusores.
#define EXTRUDERS 1

5.4.2 diámetro del filamento
configura el diámetro del filamento con el fin de compensar la impresión. Debe coincidir con la configuración del slicer.

```
#define DEFAULT_NOMINAL_FILAMENT_DIA 1.75
```

Puede anular este valor con M404 W.

5.4.2 diámetro del filamento
configura el diámetro del filamento con el fin de compensar la impresión. Debe coincidir con la configuración del slicer.

```
#define DEFAULT_NOMINAL_FILAMENT_DIA 1.75
```

Puede anular este valor con M404 W.

5.4.3 boquilla única
habilita una boquilla única.

```
#define SINGLENOZZLE
```

5.5. Fuente de alimentación

permite que la placa controladora encienda y apague la fuente de alimentación de 12v con M80 y M81.
Requiere PS_ON_PIN.

```

//#define PSU_CONTROL
//#define PSU_NAME "Power Supply"

#if ENABLED(PSU_CONTROL)
  #define PSU_ACTIVE_HIGH false // Set 'false' for ATX
  (1), 'true' for X-Box (2)

  // #define PS_DEFAULT_OFF // Keep power off until
  enabled directly with M80

  // #define AUTO_POWER_CONTROL // Enable
  automatic control of the PS_ON pin
  #if ENABLED(AUTO_POWER_CONTROL)
    #define AUTO_POWER_FANS // Turn on PSU if
    fans need power
    #define AUTO_POWER_E_FANS
    #define AUTO_POWER_CONTROLLERFAN
    #define AUTO_POWER_CHAMBER_FAN
    // #define AUTO_POWER_E_TEMP 50 // (°C) Turn
    on PSU over this temperature
    // #define AUTO_POWER_CHAMBER_TEMP 30 // (°C)
    Turn on PSU over this temperature
    #define POWER_TIMEOUT 30
  #endif
  #endif

// #define PS_DEFAULT_OFF

```

5.6. Ajustes térmicos

5.6.1. Sensores térmicos

Establece

```
#define TEMP_SENSOR_0 1
#define TEMP_SENSOR_1 0
#define TEMP_SENSOR_2 0
#define TEMP_SENSOR_3 0
#define TEMP_SENSOR_4 0
#define TEMP_SENSOR_5 0
#define TEMP_SENSOR_BED 0
#define TEMP_SENSOR_CHAMBER 0
```

Es importante realizar una buena medición de la temperatura, emplee termistores de 100K. En Configuration.h se encuentra una lista con todas las posibilidades de termistores y terminares que se pueden emplear en marlin, es importante elegir una buena calidad ya que la medición de la temperatura, tanto del extrusor como de la cama es muy importante para obtener buenos resultados. Cada perfil está calibrado para un sensor de temperatura en particular, por lo que es importante ser lo más preciso posible.

```
// Dummy thermistor constant temperature readings,
for use with 998 and 999
```

```
#define DUMMY_THERMISTOR_998_VALUE 25
#define DUMMY_THERMISTOR_999_VALUE 100
```

si se habilita esta opción es posible usar dos sensores y tener una mayor precisión y en el caso que falle uno se cancela la impresión.

```
//#define TEMP_SENSOR_1_AS_REDUNDANT
#define MAX_REDUNDANT_TEMP_SENSOR_DIFF 10
```

5.6.2. estabilidad de temperatura

Se debe mantener una temperatura estable mientras la impresora esta por comenzar el proceso.

```
#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 residency timer x degC early.
```

```
#define TEMP_BED_RESIDENCY_TIME 10 // (seconds)
#define TEMP_BED_HYSTERESIS 3 // (degC) range
of +/- temperatures considered "close" to the target one
#define TEMP_BED_WINDOW 1 // (degC) Window
around target to start the residency timer x degC early.
```

```
#define TEMP_CHAMBER_HYSTERESIS 3 // (°C)
Temperature proximity considered "close enough" to the
target.
```

5.6.3. Rangos de temperatura

nos permite proteger a la impresora de sobrecalentarse o incendiarse por posibles fallos de lectura de temperatura por error o fallo de los sensores. Ajuste en los valores mas bajos que podría experimentar la maquina. Si algún sensor cae por debajo de la temperatura mínima marlin cancelara la impresión.

```
#define HEATER_0_MINTEMP 5
#define HEATER_1_MINTEMP 5
#define HEATER_2_MINTEMP 5
#define HEATER_3_MINTEMP 5
#define HEATER_4_MINTEMP 5
#define BED_MINTEMP 5
```

la temperatura máxima también nos permite proteger a la impresora, si se realiza una lectura por encima de estos valores marlin apagara inmediatamente la impresora.

```
#define HEATER_0_MAXTEMP 285
#define HEATER_1_MAXTEMP 275
#define HEATER_2_MAXTEMP 275
#define HEATER_3_MAXTEMP 275
#define HEATER_4_MAXTEMP 275
#define BED_MAXTEMP 130
```

5.6.4. PID

el marlin incorpora para la medición de la temperatura un PID proporcional integral derivado que ayuda a estabilizar el sistema de calefacción. Cuando se configura correctamente el sistema alcanza las temperaturas mas rápidamente, experimentando un menor desgaste de los elementos. Lo mas importante es que evitara el sobreimpulso excesivo.

5.6.5. Opciones de PID para el hotend

Si se deshabilita el PIDtemp el extrusor trabajara en modo bang-bang, es un modo binario donde el extrusor esta totalmente encendido o apagado por tiempos prolongados.

```
#define PIDTEMP
#define BANG_MAX 255 // limits current to nozzle
while in bang-bang mode; 255=full current
#define PID_MAX BANG_MAX // limits current to nozzle
while PID is active (see PID_FUNCTIONAL_RANGE
below); 255=full current
#define K1 0.95
```

```
#if ENABLED(PIDTEMP)
  //#define PID_EDIT_MENU
  //#define PID_AUTOTUNE_MENU
  //#define PID_PARAMS_PER_HOTEND
```

```
  //#define PID_DEBUG
  //#define PID_OPENLOOP 1
  //#define SLOW_PWM_HEATERS
  #define PID_FUNCTIONAL_RANGE 10
```

5.6.6. Valores PID

Los valores PID de muestra se incluyen como referencia, cada maquina lleva configuraciones especificas.

Tres instancias: controlador proporcional, integral y derivativo.

```
#if ENABLED(PID_PARAMS_PER_HOTEND)
  // Specify between 1 and HOTENDS values per array.
  // If fewer than EXTRUDER values are provided, the
  last element will be repeated.
  #define DEFAULT_Kp_LIST { 22.20, 22.20 }
  #define DEFAULT_Ki_LIST { 1.08, 1.08 }
  #define DEFAULT_Kd_LIST { 114.00, 114.00 }
#else
  #define DEFAULT_Kp 22.20
  #define DEFAULT_Ki 1.08
  #define DEFAULT_Kd 114.00
#endif
```

5.6.7. Opciones de PID para la cama

Habilite el PIDTEMPBED para usar el PID de la cama caliente. Ayuda a proteger y estirar la vida útil del dispositivo.

```
//#define PIDTEMPBED
```

```
//#define BED_LIMIT_SWITCHING
```

```
#define MAX_BED_POWER 255
```

No habilite si no está el hardware en condiciones.

5.6.8. Valores PID para la cama

```
#if ENABLED(PIDTEMPBED)
```

```
  // #define PID_BED_DEBUG // Sends debug data to the
  serial port.
```

```
  // 120V 250W silicone heater into 4mm borosilicate
  (MendelMax 1.5+)
```

```
  // from FOPDT model - kp=.39 Tp=405 Tdead=66, Tc
  set to 79.2, aggressive 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
```

5.7. Seguridad

5.7.1. Prevenir la extrusión en frio

La extrusión del material por debajo del limite de temperatura puede dañar al equipo, esta opción nos permite proteger al equipo de este fallo. El motor del extrusor no se moverá si esta por debajo del limite establecido.

```
#define PREVENT_COLD_EXTRUSION
#define EXTRUDE_MINTEMP 170
```

5.7.2. Evitar extrusión prolongada.

es posible que una extrusión prolongada dañe al equipo, ya sea por un error de programación o g-code. esta función va a evitar cualquier falla relacionada. En caso de una extrusora bowden el ajuste del largo tiene que ser igual o mayor al trayecto desde el bowden al extrusor.

```
#define PREVENT_LENGTHY_EXTRUDE
#define EXTRUDE_MAXLENGTH 200
```

5.7.3. protección térmica.

Es uno de los ajustes mas importantes para prevenir futuros fallos. En caso de cualquier lectura errónea de los sensores el marlin dejara de ejecutar el programa.

```
#define THERMAL_PROTECTION_HOTENDS // Enable
thermal protection for all extruders
#define THERMAL_PROTECTION_BED // Enable
thermal protection for the heated bed
#define THERMAL_PROTECTION_CHAMBER // Enable
thermal protection for the heated chamber
```

Los errores mas comunes son:

THERMAL RUNAWAY: si la temperatura media se aleja por mucho tiempo de la temperatura objetivo el marlin ejecutara el error. Puede ser un fallo del termistor o mala configuración del PID

HEATING FAILED: puede detectar el mal funcionamiento de un termistor o una mala configuración.

5.8. CINEMATICA

podemos encontrar en marlin 4 tipos de mecánica de movimientos para ajustar la impresora:

1:cartesiana, es la mas común, cada eje dispone de un motor paso a paso para generar el movimiento y son independientes.

2:delta, los movimientos están dados a partir de sincronizar tres movimientos verticales.

3:coreXY, es una arquitectura, también muy usada, en este caso hay dos ejes que comparten motores paso a paso para generar el movimiento, generalmente son el X e Y, suele ser mas veloz que la cartesiana.

4:scara, es de las menos usadas, es similar a los movimientos de un brazo antropométrico a partir de dos movimientos en uniones inclinadas.

5.8.1 COREXY

en el caso de la corees hay que definir cuales son los ejes que van a trabajar en simultaneo.

```
//#define COREXY
//#define COREXZ
//#define COREYZ
//#define COREYX
//#define COREZX
//#define COREZY
```

5.9. Finales de carrera

los finales carrera cumplen la función de comunicar la posición relativa de los elementos mecánicos, generalmente cuando el dispositivo arranca se dirige al punto 0 para establecer la referencia inicial. Este punto debe estar sincronizado con los datos de ajuste del programa slicer que se emplee. Es muy importante para el funcionamiento ya que nos permite establecer los limites dimensionales y proteger los motores paso a paso.

5.9.1. ENDSTOP PLUGS

generalmente los dispositivos van a emplear tres fines de carrera para determinar el "0", es posible que también se empleen finales de carrera para los máximos, en ese caso hay que habilitarlos.

```
#define USE_XMIN_PLUG
#define USE_YMIN_PLUG
#define USE_ZMIN_PLUG
//#define USE_XMAX_PLUG
//#define USE_YMAX_PLUG
//#define USE_ZMAX_PLUG
```

5.9.2. ENDSTOP PULLUPS

Por defecto viene ajustado con fines de carrera pullups, son fines de carrera que hay que alimentarlos con positivo y negativo, en el interior poseen una resistencia en serie con el contacto. En caso que sean pulules vamos a ver que la señal que emiten es HIGH (1) y en caso que sea pulldowns en LOW (0).

Hay que tener en cuenta que si disponemos de finales de carrera tradicionales solo se deben emplear dos contactos.

```
#define ENDSTOPPULLUPS

#if DISABLED(ENDSTOPPULLUPS)
  // Disable ENDSTOPPULLUPS to set pullups individually
  // #define ENDSTOPPULLUP_XMAX
  // #define ENDSTOPPULLUP_YMAX
  // #define ENDSTOPPULLUP_ZMAX
  // #define ENDSTOPPULLUP_XMIN
  // #define ENDSTOPPULLUP_YMIN
  // #define ENDSTOPPULLUP_ZMIN
  // #define ENDSTOPPULLUP_ZMIN_PROBE
#endif
```

5.9.3. ENDSTOP invertir.

En caso que los finales de carrera den una señal errónea u opuesta a la que necesitamos se deben invertir, en este ajuste es posible realizar dicha inversión sin la necesidad de cambiar el fina de carrera.

```
// Mechanical endstop with COM to ground and NC to
Signal uses "false" here (most common setup).
#define X_MIN_ENDSTOP_INVERTING false // set to true
to invert the logic of the endstop.
#define Y_MIN_ENDSTOP_INVERTING false // set to true
to invert the logic of the endstop.
#define Z_MIN_ENDSTOP_INVERTING false // set to true
to invert the logic of the endstop.
#define X_MAX_ENDSTOP_INVERTING false // set to
true to invert the logic of the endstop.
#define Y_MAX_ENDSTOP_INVERTING false // set to
true to invert the logic of the endstop.
#define Z_MAX_ENDSTOP_INVERTING false // set to
true to invert the logic of the endstop.
#define Z_MIN_PROBE_ENDSTOP_INVERTING false //
set to true to invert the logic of the endstop.
```

5.9.4. STEPPER DRIVER

```
//#define X_DRIVER_TYPE A4988
//#define Y_DRIVER_TYPE A4988
//#define Z_DRIVER_TYPE A4988
//#define X2_DRIVER_TYPE A4988
//#define Y2_DRIVER_TYPE A4988
//#define Z2_DRIVER_TYPE A4988
//#define Z3_DRIVER_TYPE A4988
//#define E0_DRIVER_TYPE A4988
//#define E1_DRIVER_TYPE A4988
//#define E2_DRIVER_TYPE A4988
//#define E3_DRIVER_TYPE A4988
//#define E4_DRIVER_TYPE A4988
//#define E5_DRIVER_TYPE A4988
```

5.9.5. ENDSTOP interrupción.

Habilitar esta opción si los fines de carrera tienen la capacidad de interrupción. Generalmente son los ópticos.

```
//#define ENDSTOP_INTERRUPTS_FEATURE
```

5.9.6. ENDSTOP umbral de ruido

Habilitar si hay problemas de ruido en la lectura de los fines de carrera.

```
//#define ENDSTOP_NOISE_FEATURE
```

5.10. MOVIMIENTOS

5.10.1 Factores E diferentes

Este ajuste es cuando tenemos varios extrusores con pasos diferentes.

```
//#define DISTINCT_E_FACTORS
```

5.10.2. Pasos determinados por mm

Es uno de los ajustes más importantes del dispositivo, si esta correctamente configurada vamos a obtener piezas con precisión en las medidas.

Ajusta cuántos pasos equivalen a un milímetro en las diferentes trayectorias, vamos a ver que no son todas iguales.

Va a depender de varios factores, por ejemplo si la transmisión es por tornillo, poleas, si esta activado los micropasos y en el caso del extrusor como es la mecánica.

```
#define DEFAULT_AXIS_STEPS_PER_UNIT { 80, 80,
4000, 500 }
```

(ver:

<https://docs.google.com/spreadsheets/d/1bUPaCWFINw0933Y7nzCvadQ9cWwsZKzyNpsIBDahZzc/edit?usp=sharing>)

EJE Z	(200*4/1,25)	
pasos*mm = 640		
pasos motor/ vuelta	micropasos (1; 2; 4; 8; 16)	paso rosca
200	4	1.25

EJE X&Y	(200*4/2*20)		
pasos*mm= 20			
pasos motor/ vuelta	micropasos (1; 2; 4; 8; 16)	paso polea	número de dientes
200	4	2 GT2 paso: 2mm	20

EXTRUSOR	(200*4/π*10)	
pasos*mm= 25.464731		
pasos motor/ vuelta	micropasos (1; 2; 4; 8; 16)	diametro de la rueda dentada
200	4	10

5.10.3. Tasa máxima de alimentación predeterminada. En cualquier movimiento las velocidades estarán limitadas para proteger los mecanismos y componentes. la unidad relativa es en mm/s. Si las velocidades son muy altas el motor paso a paso puede llegar a perder pasos, por lo tanto precisión.

```
#define DEFAULT_MAX_FEEDRATE { 500, 500, 2.25, 45 }
```

5.10.4. Aceleración

5.10.4.1 Máxima aceleración predeterminada. es la aceleración o desaceleración con que va a ejecutar los cambios de movimientos. 3000 equivale a que va a llegar a la velocidad de 50mm/s en un segundo. La aceleración se expresa en mm/m2.

```
#define DEFAULT_MAX_ACCELERATION { 3000, 3000, 100, 10000 }
```

5.10.4.2. Aceleración predeterminada. son los valores predeterminados de aceleración.

```
#define DEFAULT_ACCELERATION 3000 // X, Y, Z
and E acceleration for printing moves
#define DEFAULT_RETRACT_ACCELERATION 3000 // E
acceleration for retracts
#define DEFAULT_TRAVEL_ACCELERATION 3000 // X,
Y, Z acceleration for travel (non printing) moves
```

No establecer valores demasiado altos. valores de aceleración grandes pueden provocar una vibración excesiva, pasos ruidosos o incluso saltos de pasos. Una aceleración baja produce un movimiento más suave, elimina la vibración y ayuda a reducir el desgaste de las piezas mecánicas.

5.10.4.3. JERK

es el cambio máximo de velocidad que se puede dar en un instante. No es igual a aceleración. Tanto la aceleración como el impulso afectan a la calidad de impresión, el jerk vendría a ser el impulso.

```
//#define CLASSIC_JERK
#if ENABLED(CLASSIC_JERK)
#define DEFAULT_XJERK 10.0
#define DEFAULT_YJERK 10.0
#define DEFAULT_ZJERK 0.3

//#define TRAVEL_EXTRA_XYJERK 0.0 // Additional
jerk allowance for all travel moves
//#define LIMITED_JERK_EDITING // Limit edit via
M205 or LCD to DEFAULT_aJERK * 2
#if ENABLED(LIMITED_JERK_EDITING)
#define MAX_JERK_EDIT_VALUES { 20, 20, 0.6, 10 } //
...or, set your own edit limits
#endif
#endif
```

```
#define DEFAULT_EJERK 5.0 // May be used by Linear
Advance
```

5.10.4.4. Desviación de union/intersección. ajusta el valor de la velocidad en las curvas, o cuando hay un cambio de trayectoria. la fórmula es la siguiente: $d = 0,4 * \text{jerk}^2 / \text{acel printer}$

```
#if DISABLED(CLASSIC_JERK)
#define JUNCTION_DEVIATION_MM 0.013 // (mm)
Distance from real junction edge
#endif
```

5.10.4.5. Aceleración de la curva S. el ajuste elimina la vibración durante la impresión. optimiza la curva de Bézier para acelerar el movimiento, produciendo cambios de dirección mucho más suaves

```
//#define S_CURVE_ACCELERATION
```

5.11 Opciones de sensor Z/ sensor de proximidad.

5.11.1. pines de sensor

El ajuste de esta opción se da cuando se tiene disponible el sensor de proximidad. Con esta opción marlin empleara el Z_MIN_PROBE_PIN. Con PROBE_MANUALLY se puede ajustar manualmente punto a punto la nivelación de la cama.

```
#define Z_MIN_PROBE_USES_Z_MIN_ENDSTOP_PIN
```

Esta opción se emplea en el caso de no emplear el pin de ZMIN.

```
//#define Z_MIN_PROBE_PIN 32
```

5.11.2. tipo de sensor

marlin es compatible con cualquier tipo de sensor que funciones como un interruptor. Solo hay que buscar el ajuste correcto.

5.11.3. sensor manual

Si no se tiene un sensor, igual se pueden usar de manera manual las opciones básicas de nivelación de cama AUTO_BED_LEVELING. Con MANUAL_PROBE_START_Z se establece la altura de Z con que ejecutara los m movimientos, si esta deshabilitada se moverá con altura Z=0. Si se habilita LCD_BED_LEVELING se agrega en el menú de la pantalla el proceso de nivelación.

```
//#define PROBE_MANUALLY
```

```
//#define MANUAL_PROBE_START_Z 0.2
```

5.11.4. Sensor fija

esta opción es cuando disponemos de una sonda fija, por ejemplo un sensor o la misma boquilla. Hay que especificar que tipo de sensor es.

```
//#define FIX_MOUNTED_PROBE
```

5.11.5. Sensor servoZ

Nos sirve para ajustar un sensor servoZ, por ejemplo un fin de carrera montado en un brazo giratorio. No es muy empleado.

```
//#define Z_PROBE_SERVO_NR 0 // Defaults to SERVO 0 connector.
```

```
//#define Z_SERVO_ANGLES { 70, 0 } // Z Servo Deploy and Stow angles
```

5.11.6. BLTouch

se configura un tipo especial de sensor ANTCLABS BLTouch. Emplea un conector de servo y se configura con ángulos específicos.

```
//#define BLTOUCH
```

5.11.6. TOUCHE MI PROBE

es un tipo de sensor fotoeléctrico

```
//#define TOUCH_MI_PROBE
PRYEXT.DI.FayD.UNaM
```

```
#if ENABLED(TOUCH_MI_PROBE)
  #define TOUCH_MI_RETRACT_Z 0.5 // Height
  at which the probe retracts
  //#define TOUCH_MI_DEPLOY_XPOS (X_MAX_BED +
  2) // For a magnet on the right side of the bed
  //#define TOUCH_MI_MANUAL_DEPLOY // For
  manual deploy (LCD menu)
#endif
```

5.11.7. Sensor selenoide
define un tipo de sensor.

```
//#define SOLENOID_PROBE
```

5.11.8. Sensor Z SLED

define un tipo de sensor.

```
//#define Z_PROBE_SLED
```

```
//#define SLED_DOCKING_OFFSET 5
```

5.11.9. Sensor piñón y cremallera

define un tipo de sensor.

```
//#define RACK_AND_PINION_PROBE
```

```
#if ENABLED(RACK_AND_PINION_PROBE)
  #define Z_PROBE_DEPLOY_X X_MIN_POS
  #define Z_PROBE_RETRACT_X X_MAX_POS
#endif
```

5.11.10. compensación de Sensor

especifica el ajuste de la distancia desde la boquilla al sensor, en los ejes X e Y es un valor entero, en el eje Z es un valor decimal.

5.11.11. distancia desde el borde

```
#define X_PROBE_OFFSET_FROM_EXTRUDER 10 // X
offset: -left +right [of the nozzle]
#define Y_PROBE_OFFSET_FROM_EXTRUDER 10 // Y
offset: -front +behind [the nozzle]
#define Z_PROBE_OFFSET_FROM_EXTRUDER 0 // Z
offset: -below +above [the nozzle]
```

```
#define PROBING_MARGIN 10
```

5.11.11. velocidad de testero

ajusta la velocidad del testero, esta debe ser lo mas rápido posible, salvo en el eje Z.

```
// X and Y axis travel speed (mm/m) between probes
#define XY_PROBE_SPEED 8000
```

```
// Feedrate (mm/m) for the first approach when
double-probing (MULTIPLE_PROBING == 2)
#define Z_PROBE_SPEED_FAST HOMING_FEEDRATE_Z
```

```
// Feedrate (mm/m) for the "accurate" probe of each
point
```

```
#define Z_PROBE_SPEED_SLOW
(Z_PROBE_SPEED_FAST / 2)
```

5.11.12. múltiples testeos

ajusta una serie de testeos para obtener una medición más precisa.

```
//#define MULTIPLE_PROBING 2
//#define EXTRA_PROBING 1
```

5.11.13. Espacio libre del sensor.

ajusta el espacio necesario para que los movimientos se realicen con seguridad de no colisionar con ningún objeto.

```
#define Z_CLEARANCE_DEPLOY_PROBE 10 // Z
Clearance for Deploy/Stow
#define Z_CLEARANCE_BETWEEN_PROBES 5 // Z
Clearance between probe points
#define Z_CLEARANCE_MULTI_PROBE 5 // Z
Clearance between multiple probes
// #define Z_AFTER_PROBING 5 // Z position after
probing is done
```

```
#define Z_PROBE_LOW_POINT -2 // Farthest
distance below the trigger-point to go before stopping
```

```
#define Z_PROBE_OFFSET_RANGE_MIN -20
#define Z_PROBE_OFFSET_RANGE_MAX 20
```

¡Asegúrese de tener suficiente espacio libre para que la sonda se mueva entre los puntos!

5.11.14. pruebas de sensor.

permite verificar la precisión del sensor.

```
#define Z_MIN_PROBE_REPEATABILITY_TEST

// Before deploy/stow pause for user confirmation
// #define PAUSE_BEFORE_DEPLOY_STOW
#if ENABLED(PAUSE_BEFORE_DEPLOY_STOW)
  // #define PAUSE_PROBE_DEPLOY_WHEN_TRIGGERED
  // For Manual Deploy Allenkey Probe
#endif
```

5.11.15. Pruebas/sensor con boquilla en frio.

Ajuste para realizar la nivelación de cama en frio.

```
// #define PROBING_HEATERS_OFF // Turn heaters
off when probing
#if ENABLED(PROBING_HEATERS_OFF)
  // #define WAIT_FOR_BED_HEATER // Wait for bed to
heat back up between probes (to improve accuracy)
#endif
// #define PROBING_FANS_OFF // Turn fans off
when probing
// #define PROBING_STEPPERS_OFF // Turn steppers
off (unless needed to hold position) when probing
// #define DELAY_BEFORE_PROBING 200 // (ms) To
prevent vibrations from triggering piezo sensors
```

5.12. STEPPER DRIVERS

5.12.1. Habilitar Motor - ENABLE

es el ajuste de los pines para establecer el estado de los motores.

```
#define X_ENABLE_ON 0
#define Y_ENABLE_ON 0
#define Z_ENABLE_ON 0
#define E_ENABLE_ON 0 // For all extruders
```

0= LOW

1= HIGH

5.12.2. Habilitar Motor - DISABLE

se puede ajustar o desactivar el funcionamiento de los motores cuando no están en movimiento, para protegerlos de un sobrecalentamiento. Produce una baja estabilidad cuando se desactivan los motores. No es muy recomendable.

```
#define DISABLE_X false
#define DISABLE_Y false
#define DISABLE_Z false
```

```
// #define DISABLE_REDUCED_ACCURACY_WARNING
```

Habilitar esta opción para suprimir la advertencia, dado que es probable que se produzca una reducción de la precisión.

```
#define DISABLE_E false
// For all extruders
#define DISABLE_INACTIVE_EXTRUDER false //
Keep only the active extruder enabled
```

5.12.3. direccion del motor

Invierte el sentido de giro de los motores en caso que no se quiera o pueda acceder a las conexiones físicas.

```
#define INVERT_X_DIR false
#define INVERT_Y_DIR true
#define INVERT_Z_DIR false
```

```
#define INVERT_E0_DIR false
#define INVERT_E1_DIR false
#define INVERT_E2_DIR false
#define INVERT_E3_DIR false
#define INVERT_E4_DIR false
```

5.13. Límites y referencia 0

5.13.1. Altura Z de referencia

Ajusta la altura Z cuando se mueva en los ejes X e Y para que no se produzcan colisiones.

```

//#define NO_MOTION_BEFORE_HOMING // Inhibit
movement until all axes have been homed
//#define UNKNOWN_Z_NO_RAISE // Don't
raise Z (lower the bed) if Z is "unknown."

```

```

//For beds that fall when Z is powered
off.
//#define Z_HOMING_HEIGHT 4

```

5.13.2. Direccional de inicio.

La referencia en cada eje es -1 mínimo y 1 máximo.

```

#define X_HOME_DIR -1
#define Y_HOME_DIR -1
#define Z_HOME_DIR -1

```

5.13.3. Límites de movimiento. se ajusta el tamaño de la cama.

```

#define X_BED_SIZE 200
#define Y_BED_SIZE 200

```

esta opción establece los máximos y mínimos, donde se colocan los finales de carrera.

```

#define X_MIN_POS 0
#define Y_MIN_POS 0
#define Z_MIN_POS 0
#define X_MAX_POS X_BED_SIZE
#define Y_MAX_POS Y_BED_SIZE
#define Z_MAX_POS 170

```

5.13.3. Finales de carrera por software.

restringe los movimientos cuando se superan los límites físicos programados.

```

#define MIN_SOFTWARE_ENDSTOPS
#if ENABLED(MIN_SOFTWARE_ENDSTOPS)
  #define MIN_SOFTWARE_ENDSTOP_X
  #define MIN_SOFTWARE_ENDSTOP_Y
  #define MIN_SOFTWARE_ENDSTOP_Z
#endif

```

```

#define MAX_SOFTWARE_ENDSTOPS
#if ENABLED(MAX_SOFTWARE_ENDSTOPS)
  #define MAX_SOFTWARE_ENDSTOP_X
  #define MAX_SOFTWARE_ENDSTOP_Y
  #define MAX_SOFTWARE_ENDSTOP_Z
#endif

```

para habilitar y deshabilitar finales de carrera desde el display.

```

#if EITHER(MIN_SOFTWARE_ENDSTOPS,
MAX_SOFTWARE_ENDSTOPS)
  #define SOFT_ENDSTOPS_MENU_ITEM
#endif

```

5.14. Sensor de filamento.

Esta opción permite detener el proceso de impresión cuando no detecta presencia de filamento.

```

//#define FILAMENT_RUNOUT_SENSOR
#if ENABLED(FILAMENT_RUNOUT_SENSOR)
  #define NUM_RUNOUT_SENSORS 1 // Number of
  sensors, up to one per extruder. Define a FIL_RUNOU-
  T#_PIN for each.
  #define FIL_RUNOUT_INVERTING false // Set to true to
  invert the logic of the sensor.
  #define FIL_RUNOUT_PULLUP // Use internal
  pullup for filament runout pins.
  //#define FIL_RUNOUT_PULLDOWN // Use internal
  pulldown for filament runout pins.

  // Set one or more commands to execute on filament
  runout.
  // (After 'M412 H' Marlin will ask the host to handle the
  process.)
  #define FILAMENT_RUNOUT_SCRIPT "M600"

  // When using a runout switch (no encoder), after a
  runout is detected,
  // continue printing this length of filament before
  executing the runout script.
  // Useful for a sensor at the end of a feed tube.
  // If using an encoder disc, this is the length of filament
  that would print
  // without any movement from the sensor before it
  triggers a runout.
  // Requires 4 bytes SRAM per sensor, plus 4 bytes
  overhead.
  //#define FILAMENT_RUNOUT_DISTANCE_MM 25

  #ifndef FILAMENT_RUNOUT_DISTANCE_MM
  // Enable this option to use an encoder disc that
  toggles the runout pin as the filament moves.
  // Be sure to set FILAMENT_RUNOUT_DISTANCE_MM
  large enough to avoid false positives.
  // Start at the value of the sensor for one revolution
  and if you experience false positives,
  // increment the value by the same amount.
  // ie., 7mm is set, and you get false positives, set it to
  14 and try it again.
  //#define FILAMENT_MOTION_SENSOR
  #endif
#endif

```

5.15. Nivelación de cama.

la nivelación de cama nos ayuda a eliminar pequeñas imperfecciones, pero siempre es necesario tomar todas las medidas posibles antes de usar la nivelación automática.

Habilita la nivelación automática de cama. Solo es posible habilitar uno.

```

//#define AUTO_BED_LEVELING_3POINT
//#define AUTO_BED_LEVELING_LINEAR
//#define AUTO_BED_LEVELING_BILINEAR
//#define AUTO_BED_LEVELING_UBL
//#define MESH_BED_LEVELING

```

AUTO_BED_LEVELING_3POINT: palpa tres puntos en forma de triángulo, proporciona una matriz adecuada para compensar, pero siempre el plano permanecerá inclinado.

AUTO_BED_LEVELING_LINEAR: palpa la cama en una cuadrícula, produce una matriz mediante el método de mínimos cuadrados, pero siempre el plano permanecerá inclinado.

AUTO_BED_LEVELING_BILINEAR: palpa la cama en una cuadrícula, a diferencia de la anterior emplea una interpolación bilineal para generar la matriz.

AUTO_BED_LEVELING_UBL: combina las tres funciones de nivelación puntos, lineal y bilineal. Requiere controlador LCD.

MESH_BED_LEVELING: es una nivelación personalizada que se ajusta mediante una galga o papel, es más compleja.

5.16. Opciones de inicio. Homing.

5.16.1. Centro de cama
Define el cero del equipo.

```
//#define BED_CENTER_AT_0_0
```

5.16.2. Posición inicial manual.
Anulan los ajustes iniciales.

```
//#define MANUAL_X_HOME_POS 0
//#define MANUAL_Y_HOME_POS 0
//#define MANUAL_Z_HOME_POS 0 // Distance from
nozzle to printbed after homing
```

5.16.3. Homming seguro
Ajusta los trayectos en altura "z" por seguridad, en el recorrido inicial.

```
#define Z_SAFE_HOMING

#if ENABLED(Z_SAFE_HOMING)
  #define Z_SAFE_HOMING_X_POINT ((X_BED_SIZE) / 2)
  // X point for Z homing when homing all axes (G28).
  #define Z_SAFE_HOMING_Y_POINT ((Y_BED_SIZE) / 2)
  // Y point for Z homing when homing all axes (G28).
#endif
```

5.16.4. Velocidad de homing.
Ajustan las velocidades de inicio.

```
// Homing speeds (mm/m)
#define HOMING_FEEDRATE_XY (50*60)
#define HOMING_FEEDRATE_Z (4*60)
```

Valores muy altos pueden causar una mala resolución en el posicionamiento inicial.

Valida la referencia a partir de los fines de carrera.

```
#define VALIDATE_HOMING_ENDSTOPS
```

5.16.5. Compensación de inclinación de cama.
Corrige o ajusta posibles defasajes en los ejes xyz

```
//#define SKEW_CORRECTION

#if ENABLED(SKEW_CORRECTION)
  // Input all length measurements here:
  #define XY_DIAG_AC 282.8427124746
  #define XY_DIAG_BD 282.8427124746
  #define XY_SIDE_AD 200

  // Or, set the default skew factors directly here
  // to override the above measurements:
  #define XY_SKEW_FACTOR 0.0

  //#define SKEW_CORRECTION_FOR_Z
  #if ENABLED(SKEW_CORRECTION_FOR_Z)
    #define XZ_DIAG_AC 282.8427124746
    #define XZ_DIAG_BD 282.8427124746
    #define YZ_DIAG_AC 282.8427124746
    #define YZ_DIAG_BD 282.8427124746
```

```
#define YZ_SIDE_AD 200
#define XZ_SKEW_FACTOR 0.0
#define YZ_SKEW_FACTOR 0.0
#endif

// Enable this option for M852 to set skew at runtime
//#define SKEW_CORRECTION_GCODE
#endif
```

5.17. Características adicionales.

5.17.1 EEPROM.

```
///#define EEPROM_SETTINGS
```

EPROM es la memoria no volátil, en dicha memoria se pueden alojar datos que queramos recuperar entre reinicio y reinicio, siempre y cuando este activada la selección.

Los comandos relacionados con la EEPROM son:

M500: guarda todos los ajustes actuales en la EEPROM.
M501: Cargue todas las configuraciones guardadas por última vez en EEPROM.

M502: restablecer todas las configuraciones a sus valores predeterminados (según lo establecido por Configuration.h)

M503: Imprime la configuración actual (en RAM, no en EEPROM)

5.17.2 Opciones de EEPROM.

```
///#define DISABLE_M503 // Saves ~2700 bytes of  
PROGMEM. Disable for release!  
#define EEPROM_CHITCHAT // Give feedback on  
EEPROM commands. Disable to save PROGMEM.
```

En placas de 128K o mas pequeñas se desaconseja el empleo.

5.17.3 Temporizador de trabajos de impresión

```
#define PRINTJOB_TIMER_AUTOSTART.
```

Inicia y detiene automáticamente los trabajos de impresión.

5.17.4 Contador de impresiones.

```
///#define PRINTCOUNTER
```

Cuando esta activado realiza una estadística de algunas variables de impresiones.

Trabajos de impresión totales
Total de trabajos de impresión correctos
Total de trabajos de impresión fallidos
Impresión de tiempo total
Esta información se puede ver con el comando M78.

5.18.1 Idioma de la interfase.

```
#define LCD_LANGUAGE en
```

permite elegir el idioma para el controlador LCD.

5.18.2 Conjunto de caracteres HD44780.

```
#define DISPLAY_CHARSET_HD44780 JAPANESE
```

Solo funciona en determinados modelos de pantallas, permite la configuración de otros caracteres, como ser el cirílico, japonés u occidental.

5.18.3 LCD_info_pantalla_estilo.

```
#define LCD_INFO_SCREEN_STYLE 0
```

0 para clásico; 1 para el estilo de pantalla de información Prusa.

5.19.1 Tarjeta de memoria.

```
//#define SDSUPPORT
```

Habilita el funcionamiento de la tarjeta de memoria externa.

5.19.2 Velocidad SPI.

```
//#define SPI_SPEED SPI_HALF_SPEED  
//#define SPI_SPEED SPI_QUARTER_SPEED  
//#define SPI_SPEED SPI_EIGHTH_SPEED
```

Determina la velocidad de lectura de la tarjeta. Si tiene problemas de lectura intente con una velocidad mas lenta.

5.19.3 Habilitar CRC.

```
//#define SD_CHECK_AND_RETRY
```

Error de datos redundancia cíclica.

5.20 Elementos del menú de la pantalla LCD.

```
//#define NO_LCD_MENUS  
//#define SLIM_LCD_MENUS
```

Deshabilita el menú o elimina algunos componentes.

5.21 Controlador LCD.

Marlin incluye soporte para los controladores mas utilizados:

REPRAP_DISCOUNT_SMART_CONTROLLER Un controlador LCD basado en caracteres de 20 x 4 con rueda de clic.

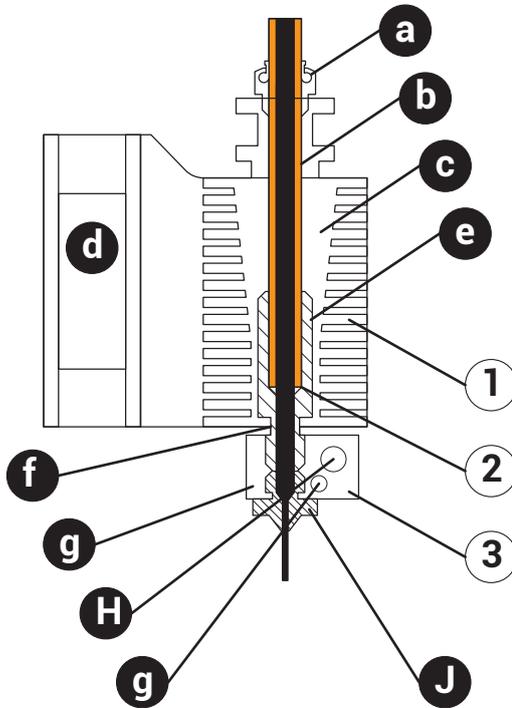
REPRAP_DISCOUNT_FULL_GRAPHIC_SMART_CONTROLLER Un controlador LCD monocromático de 128 x 64 píxeles con rueda de clic. Capaz de mostrar gráficos de mapa de bits simples y hasta 5 líneas de texto.

La mayoría de los otros controladores LCD son variantes de estos. Habilite solo una de las siguientes opciones para su controlador específico.

info extraída de : (<https://marlinfw.org/docs/configuration/configuration.html>)

6. Pruebas y testeos

6.1. ¿Como prevenir un fallo?.



- a.Acople rapido ltubo PTFE
- b.Tubo PTFE
- c.Disipador
- d.Ventilador
- e.Barrel
- f.Puente termico
- g.Cubo
- h.Resistencia
- i.Termistor
- j.Boquilla

- 1.Zona fria
- 2.En este sector tenemos que tener el cuidado que el tubo de PTFE haga contacto en el fondo del barrel y este bien ajustado. Ya que es donde se producen la expansión del material y se atasca.
- 3.Zona caliente

PROBLEMA DE ATASCAMIENTO

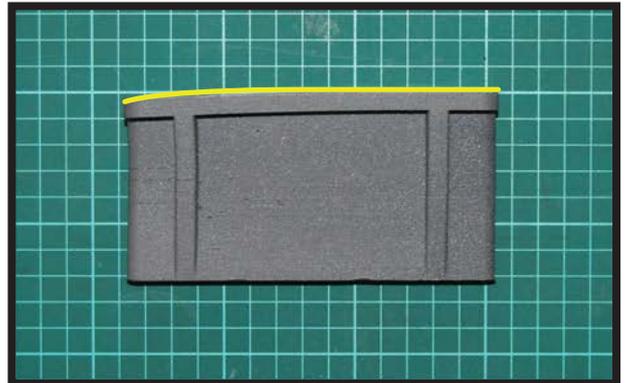
6.2. Tipos comunes de fallos y su solución.

- Maraña



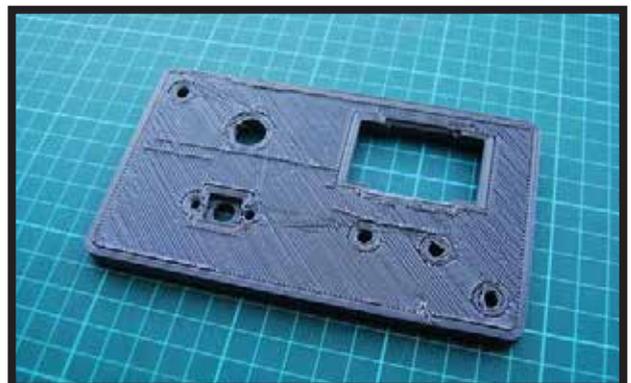
*Mala adherencia a la cama.

- despegue de la cama;warping

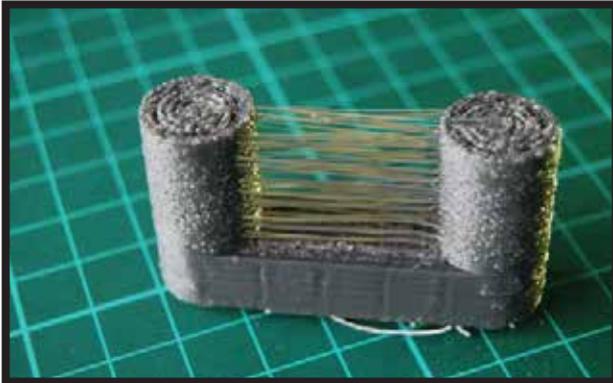


*Mala adherencia a la cama
 *Gran volumen de material (contracción)
 *Enfriamiento durante la impresión
 *Mayormente en ABS

- Primera capa defectuosa



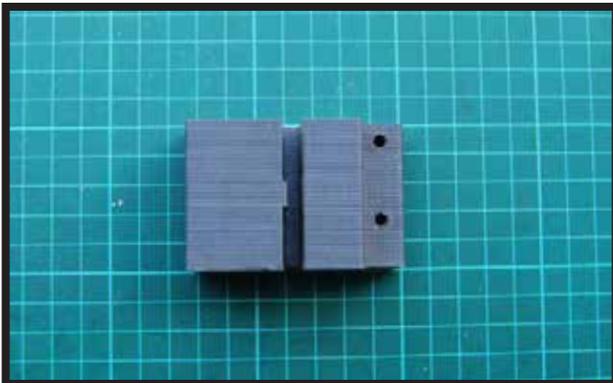
- hilos



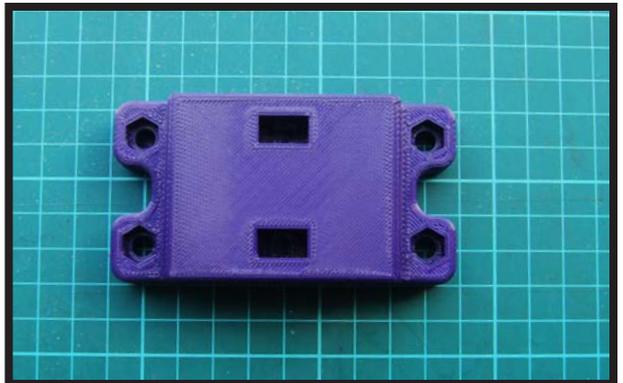
- marcas o puntos en la superficie; costura



- imperfección en la superficie



- ultima capa defectuosa; superposición (overlap)



- imperfección en la superficie



- * *Altura de capa*

- soportes

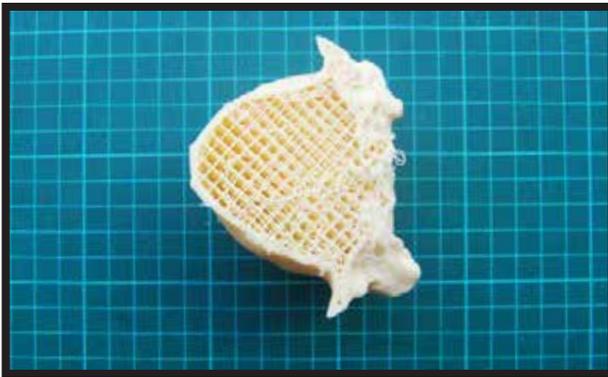


- primer capa incompleta
- primer capa incompleta
- ultima capa incompleta <<<overlap>>>
- ultima capa incompleta <<<overlap>>>
- textura en la pieza
- textura en la pieza
- marcas o puntos en la superficie
- disminución o aumento de las medidas de la pieza
- obstrucción del extrusor

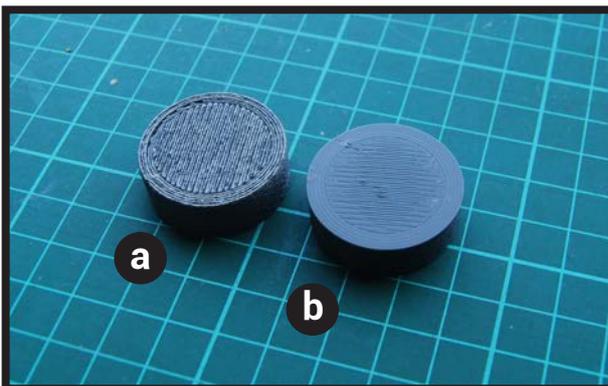
- despegue entre capa; delaminación; cracking



- pieza incompleta



- Primera capa defectuosa



*a. Con raft

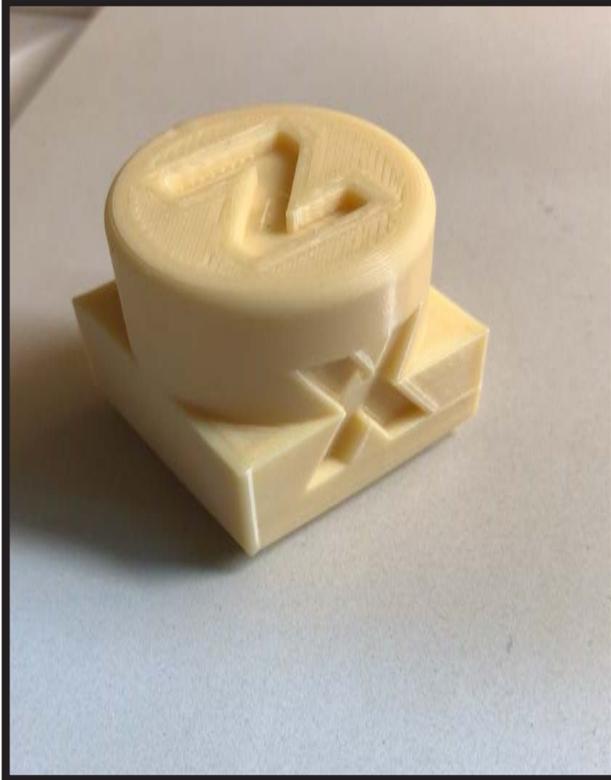
Altura balsa a primer capa

*b. Con falda

Altura cama a primer capa

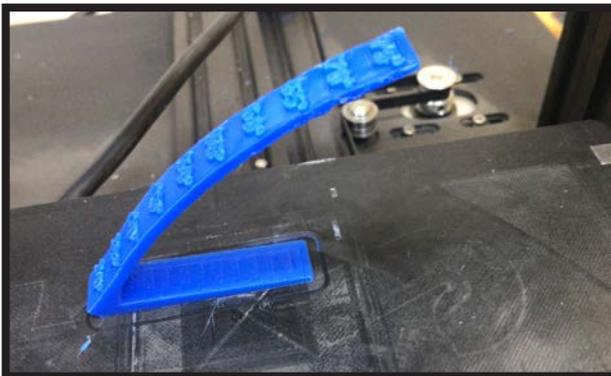
6.3. Puesta a punto de un dispositivo de manufactura digital.

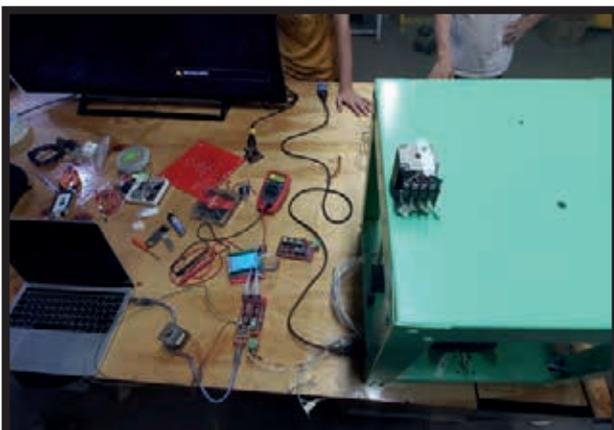
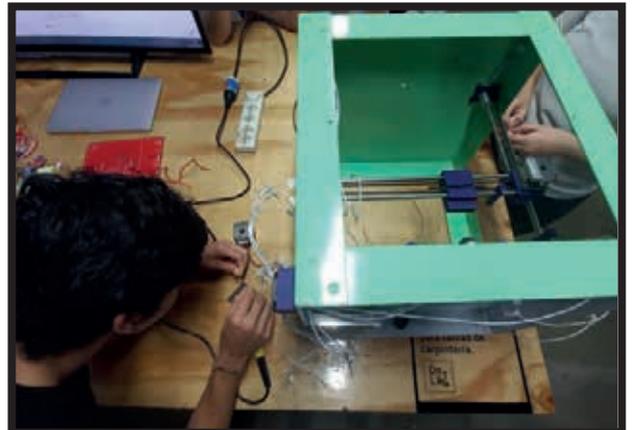
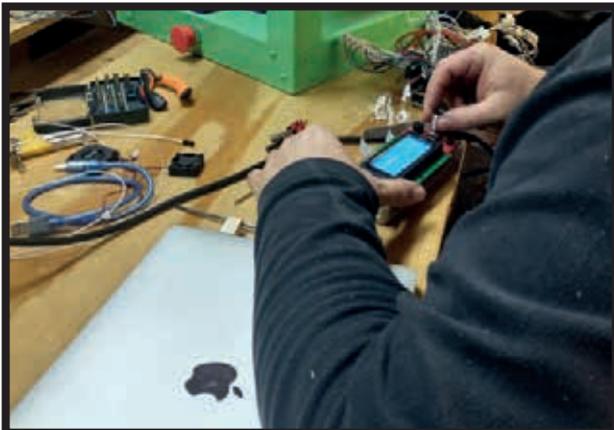
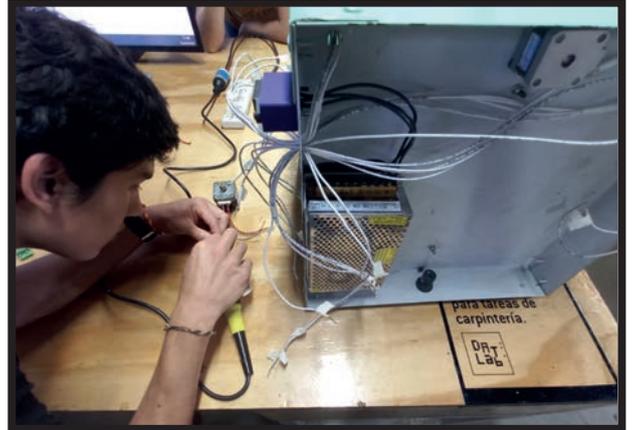
- Cubo de calibración.

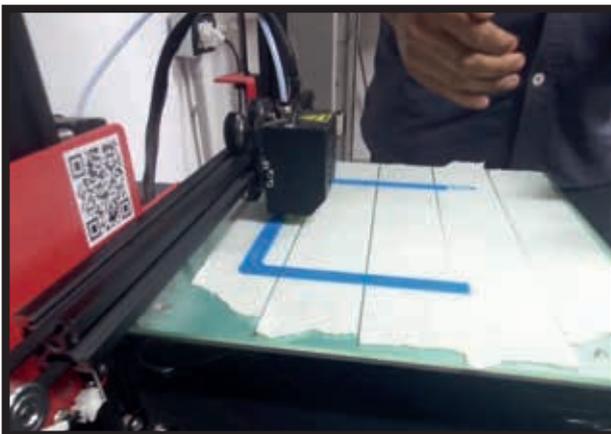
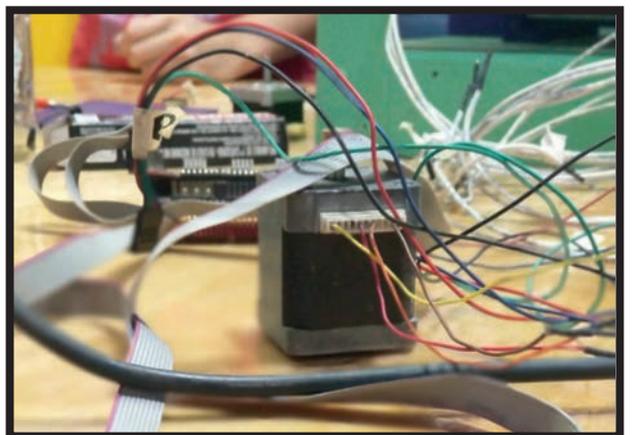
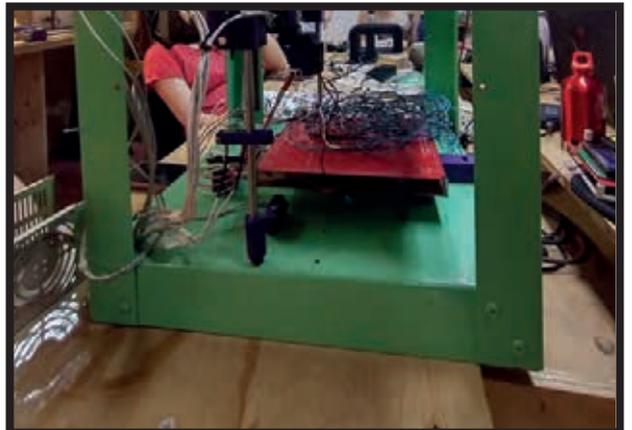


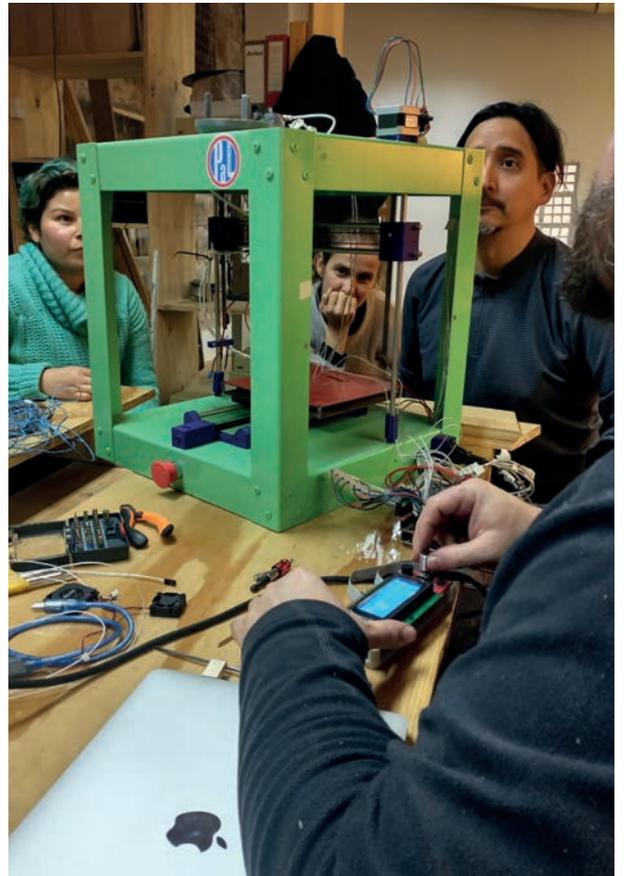
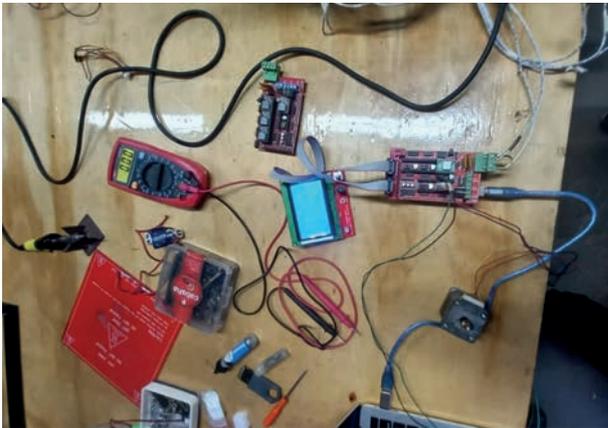
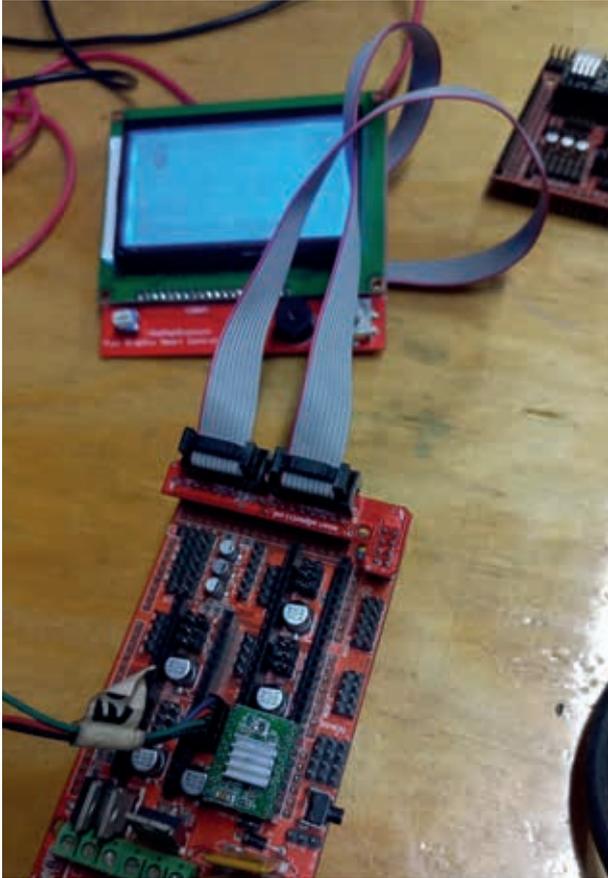
**6.4. Equipamiento básico.
6.5. Árbol de errores.**

- Calibración de ángulo de voladizo..



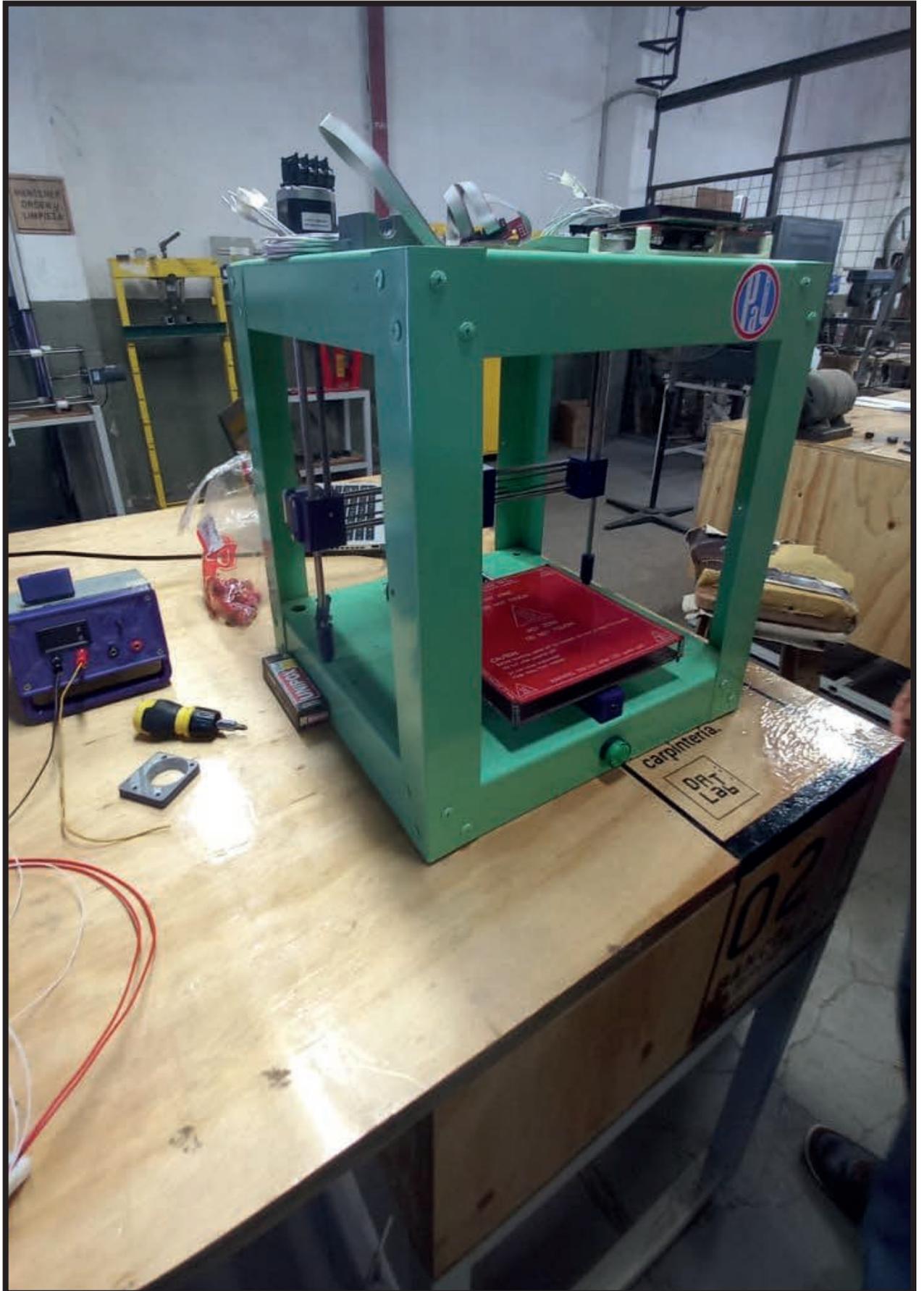












Resultados:

Se logro REPARAR, CORREGIR, PROBAR Y DOCUMENTAR la experiencia sobre un dispositivo de fabricación digital: impresora 3D FDM.

Se diseñaron, probaron y fabricaron varias piezas; la fabricación se realizo en impresión 3d en ABS y PLA, con el fin de completar las piezas de la impresora reparada.

Se fabrico un prototipo de impresora 3D, con el fin de reparar, probar, corregir y documentar las experiencias.

Participaron más de 30 alumnos y profesores en el procesos.

