# Aplicación de línea de llenado automatizado

Contenido

[Aplicación de línea de llenado automatizado 1](#_Toc195049526)

[1.Requerimientos 3](#_Toc195049527)

[1.1. Generalidades 3](#_Toc195049528)

[1.2. Requerimientos del Sistema 3](#_Toc195049529)

[1.2.1 Banda Transportadora 3](#_Toc195049530)

[1.2.2 Sistema de Llenado 3](#_Toc195049531)

[1.2.3 Condiciones de Evaluación 3](#_Toc195049532)

[1.2.4 Consideraciones adicionales 3](#_Toc195049533)

[2. Diseño conceptual 5](#_Toc195049534)

[2.1 Arquitectura del sistema 5](#_Toc195049535)

[2.2 Diagrama de flujo del proceso 5](#_Toc195049536)

[2.3 Selección de componentes principales 6](#_Toc195049537)

[3. Diseño Mecánico y Electrónico del Prototipo. 8](#_Toc195049538)

[3.1 Modelo 3D del sistema mecánico 8](#_Toc195049539)

[3.2 Diseño del sistema electrónico 10](#_Toc195049540)

[3.2.1 Entrada de alimentación 11](#_Toc195049541)

[3.2.2 Indicadores lumínicos 11](#_Toc195049542)

[3.2.3 Botones de Inicio y Paro. 12](#_Toc195049543)

[3.2.4 Sensores de fin de carrera 13](#_Toc195049544)

[3.2.5 Sensores infrarrojos 14](#_Toc195049545)

[3.2.6 Puente H 14](#_Toc195049546)

[3.2.7 ESP 32 15](#_Toc195049547)

[3.2.8 Diseño de PCB 16](#_Toc195049548)

[4. Manufacturación del prototipo 19](#_Toc195049549)

[4.1. Fabricación del diseño mecánico 19](#_Toc195049550)

[4.1.1 Observaciones durante las pruebas preliminares 19](#_Toc195049551)

[4.1.2 Rediseño del sistema mecánico 19](#_Toc195049552)

[4.2 Fabricación del diseño electrónico 20](#_Toc195049553)

[4.2.1 Observaciones durante las pruebas preliminares del circuito 21](#_Toc195049554)

[4.2.2 Rediseño del sistema electrónico 21](#_Toc195049555)

[5. Programación del sistema de control embebido 23](#_Toc195049556)

[5.1 Definición de pines 23](#_Toc195049557)

[5.2 Definición de estados 24](#_Toc195049558)

[5.3 Variables globales 24](#_Toc195049559)

[5.4 Funciones de control 25](#_Toc195049560)

[5.4.1 interrupcionParoEmergencia() 25](#_Toc195049561)

[5.4.2 controlLED() 25](#_Toc195049562)

[5.5 Funciones de estado 25](#_Toc195049563)

[5.5.1 esatdo\_Inicio() 26](#_Toc195049564)

[5.5.2 esatdo\_ESPERA\_ACCIONAR\_BANDA() 26](#_Toc195049565)

[5.5.3 estado\_AVANZA\_LLENADO() 27](#_Toc195049566)

[5.5.4 estado\_LLENADO\_RECIPIENTE\_BAJO() y estado\_LLENADO\_RECIPIENTE\_ALTO() 27](#_Toc195049567)

[5.5.5 esatdo\_AVANZA\_FINAL() 29](#_Toc195049568)

[5.5.6 estado\_POCISION\_FINAL() 29](#_Toc195049569)

[5.5.7 estado\_PARO\_EMERGENCIA() 29](#_Toc195049570)

[5.6 setup() 31](#_Toc195049571)

[6. Conclusiones 33](#_Toc195049572)

[Anexos 34](#_Toc195049573)

[Código de Diagrama 1 34](#_Toc195049574)

[Código Diagrama Maquina de Estados 36](#_Toc195049575)

[Código Tabla representativa máquina de estados 38](#_Toc195049576)

[Modelos 3D a utilizar 41](#_Toc195049577)

[Referencias Bibliográficas 42](#_Toc195049578)

# 1.Requerimientos

## 1.1. Generalidades

Se debe desarrollar un sistema automatizado de llenado de recipientes que incluya una banda transportadora y un sistema de dispensación de líquido. El prototipo deberá operar de forma autónoma sin intervención manual durante el proceso de llenado.

## 1.2. Requerimientos del Sistema

### 1.2.1 Banda Transportadora

* Debe contar con tres sensores de posición:
  + Sensor de inicio: Detecta la presencia del recipiente en la posición inicial.
  + Sensor intermedio: Detiene la banda cuando el recipiente llega a la estación de llenado.
  + Sensor final: Detiene la banda una vez que el recipiente ha sido llenado y transportado al final del proceso.
* La banda transportadora debe operar bajo las siguientes condiciones:
  + Al presionar el botón de inicio:
    - Si no hay un recipiente en el sensor de inicio, la banda no debe encenderse.
    - Si sí hay un recipiente en el sensor de inicio, la banda debe encenderse después de 5 segundos.
  + La banda debe detenerse cuando el recipiente llegue al sensor intermedio.
  + Una vez completado el llenado, la banda debe reactivarse para transportar el recipiente hasta el sensor final.
  + La banda debe permanecer detenida en el sensor final hasta que el operador retire el recipiente.
  + No se podrá iniciar un nuevo ciclo hasta que el recipiente haya sido retirado.

### 1.2.2 Sistema de Llenado

* El sistema debe detectar la marca del recipiente y llenarlo hasta el nivel correspondiente.
* Se deben considerar dos posibles niveles de llenado según la marca del recipiente.
* Una vez alcanzado el nivel de llenado, el dispensador de líquido debe detenerse automáticamente.

### 1.2.3 Condiciones de Evaluación

* El sistema debe operar de manera completamente autónoma sin intervención manual.
* No está permitido sujetar el recipiente, manipular la banda transportadora ni el sistema de dispensación durante la operación.
* Se deberá presentar un reporte en video del funcionamiento del sistema una vez que haya sido validado y aprobado.

### 1.2.4 Consideraciones adicionales

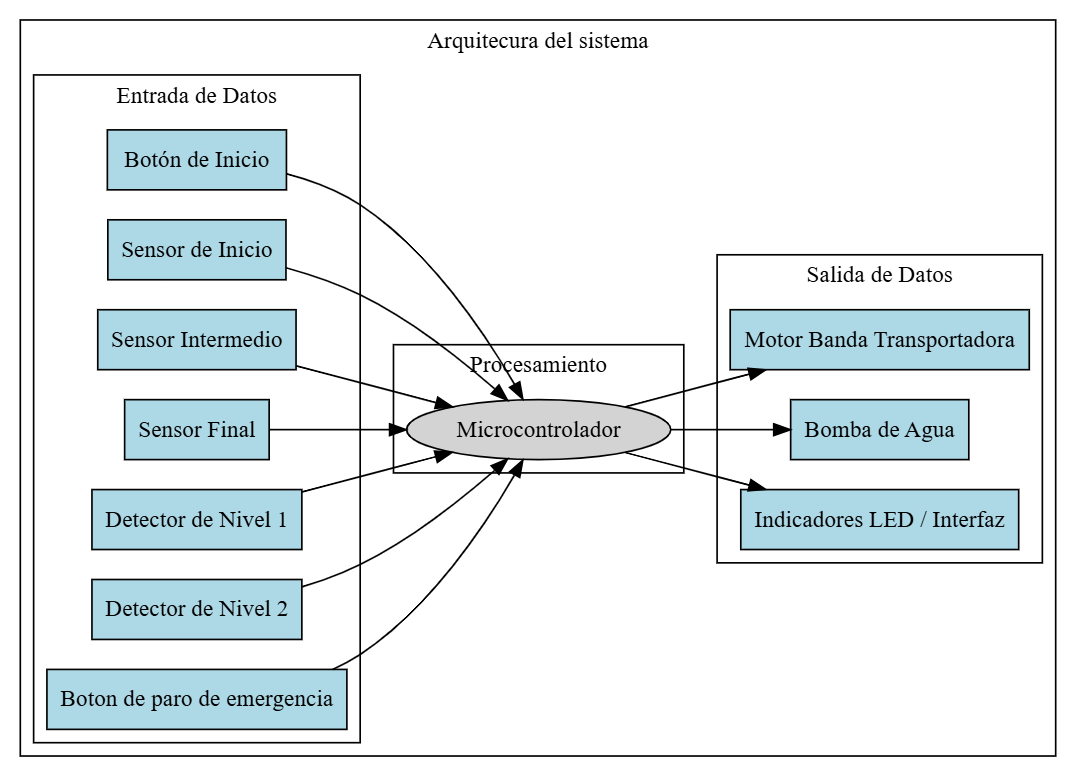
* Implementar un botón de paro de emergencia accesible para el operador.
* Agregar una señalización luminosa que indique cuando la banda esté en operación y cuando esté en reposo.

# 2. Diseño conceptual

En la fase de diseño conceptual, se establecen los principios fundamentales que rigen el funcionamiento del sistema, así como la selección de los componentes clave para su implementación. Se ha dividido el diseño en tres áreas principales: Arquitectura del sistema, Diagrama de flujo del proceso y Selección de componentes.

## 2.1 Arquitectura del sistema

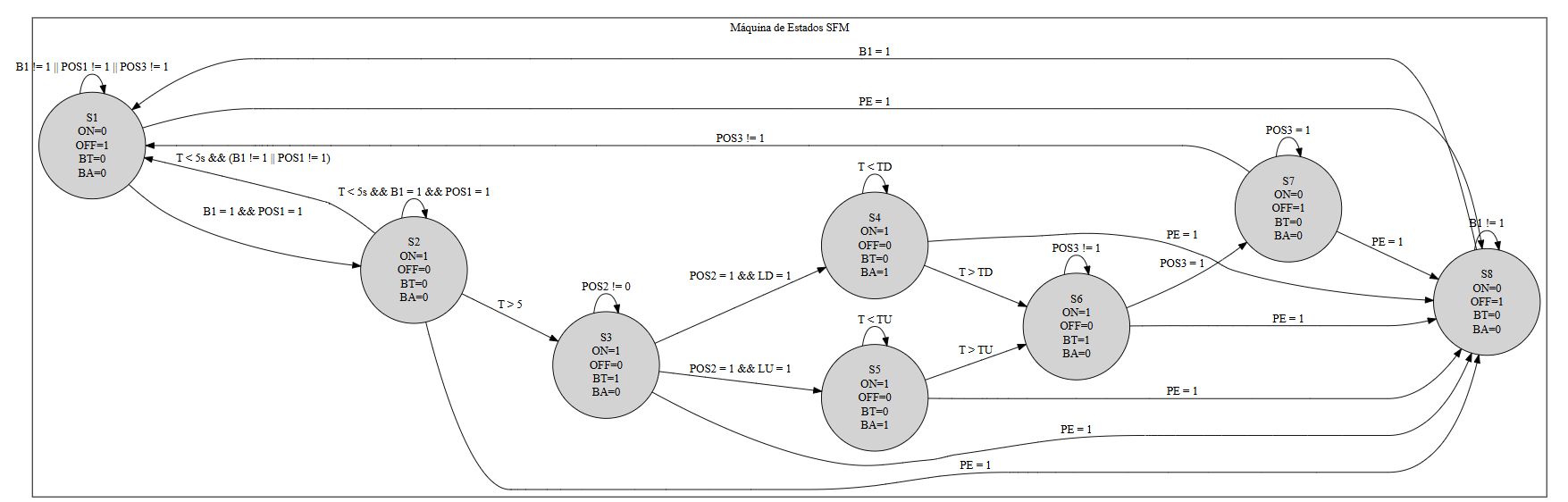
Representada mediante un diagrama de bloques funcional la interacción entre sensores, actuadores y el controlador principal.

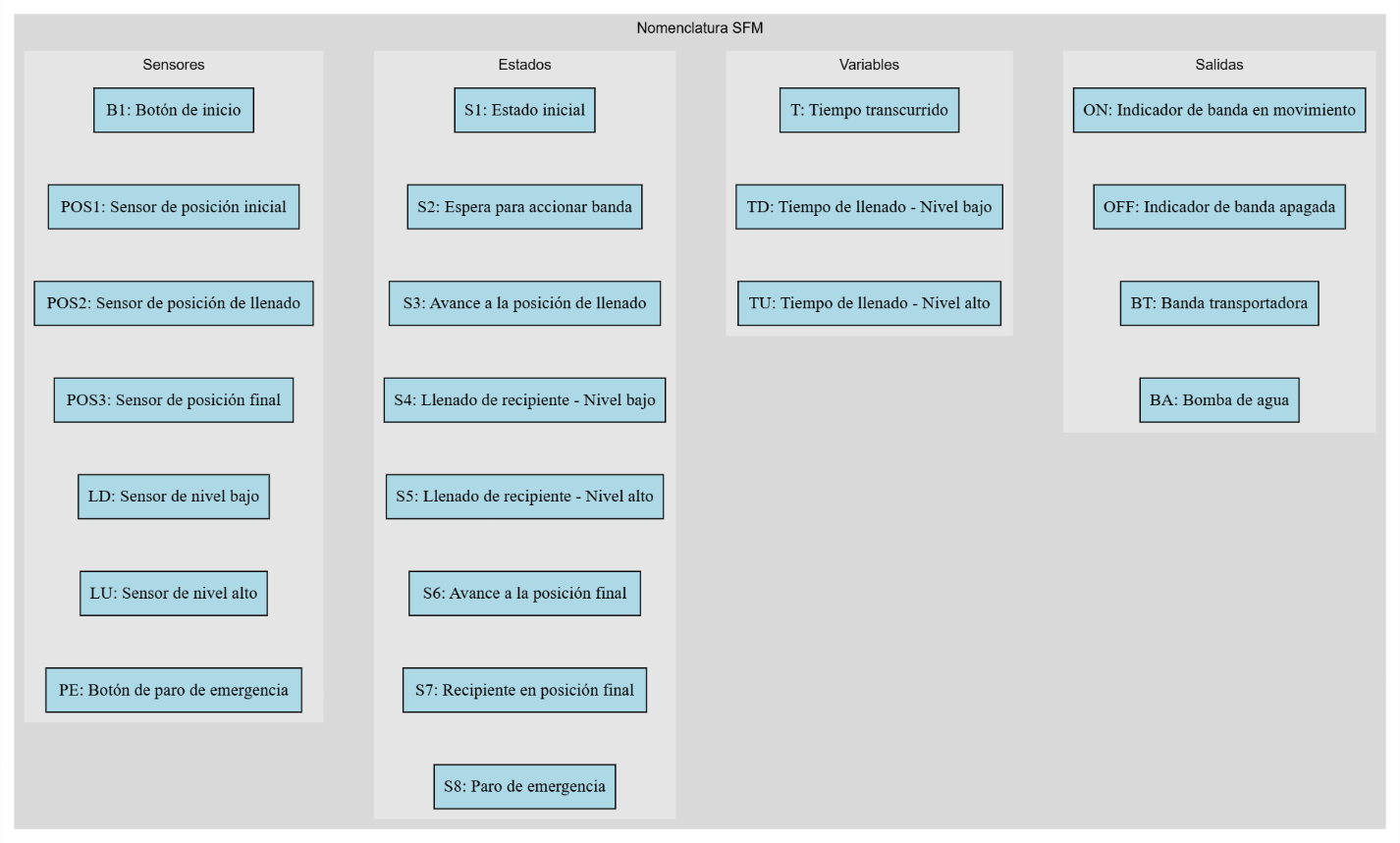


## 2.2 Diagrama de flujo del proceso

El proceso de llenado automatizado será llevado a cabo mediante una **máquina de estados finitos (FSM)**, donde cada estado representa una fase específica del ciclo de operación, y las transiciones entre estados se activan mediante condiciones predefinidas basadas en la entrada de los sensores y las intervenciones del operador.

En la máquina de estados, los **círculos** representan los diferentes **estados** del sistema, y dentro de cada uno se observan las salidas asociadas al estado actual, tales como los indicadores lumínicos, el estado de la banda transportadora y la bomba de agua. Las **flechas** entre los estados indican las **condiciones de transición** que determinan el paso de un estado a otro.





## 2.3 Selección de componentes principales

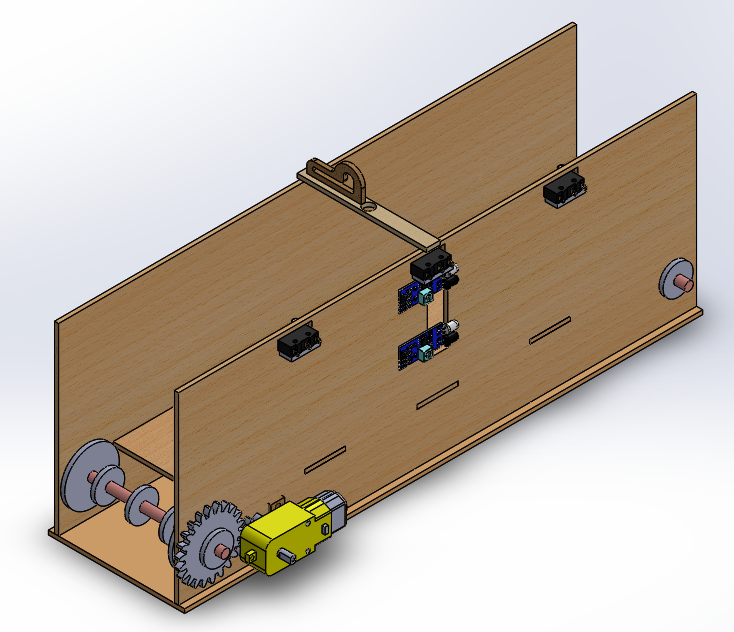
Para la implementación del prototipo, se han seleccionado componentes que garantizan un equilibrio entre funcionalidad, disponibilidad y costo, considerando las necesidades del sistema. A continuación, se describen los elementos principales:

* **Controlador:** ESP32, Se selecciona el microcontrolador ESP32 debido a su versatilidad, bajo costo y capacidades avanzadas de conectividad y procesamiento. Este dispositivo permite el control eficiente de entradas y salidas digitales, así como la ejecución de lógicas de control basadas en máquinas de estados, además de ofrecer conectividad Wi-Fi y Bluetooth, útil para futuras expansiones del sistema.
* **Sensores:**
  + **Sensores de final de carrera:** Se emplean para la detección física de la posición del recipiente en las distintas etapas del proceso (inicio, llenado y final). Estos sensores proporcionan señales digitales confiables y de respuesta rápida.
  + **Sensor infrarrojo:** Utilizado para identificar la marca que indica el nivel de llenado del recipiente. Esta solución permite diferenciar de forma no invasiva entre dos niveles posibles, lo que facilita la adaptación a distintos recipientes.
* **Actuadores:**
  + **Motorreductor:** Elegido por su compatibilidad con sistemas de bajo voltaje y por proporcionar un par de giro adecuado para el movimiento controlado de la banda transportadora.
  + **Bomba de agua sumergible:** Se opta por una bomba capaz de suministrar hasta 2 litros por minuto, suficiente para lograr un llenado ágil y eficiente sin comprometer la estabilidad del sistema.
  + **Puente H**: Este circuito es necesario para controlar los motores de forma bidireccional, permitiendo una gestión completa del movimiento en ambas direcciones, si fuese necesario.
  + **Led’s**: Se emplean para representar de manera visual los estados del sistema, facilitando el monitoreo del proceso por parte del operador.
* **Estructura:** Se utilizará **panel MDF** (tablero de fibra de densidad media) como material base para la construcción del prototipo. Este material es económico, fácil de mecanizar, uniforme y adecuado para el montaje de componentes electrónicos y mecánicos.
* **Recipientes:** Como parte del sistema, se seleccionan de forma arbitraria vasos de unicel de 8 oz (237 ml) como recipientes estándar. Esta elección se basa en su disponibilidad, uniformidad y facilidad de manipulación en el entorno de laboratorio.

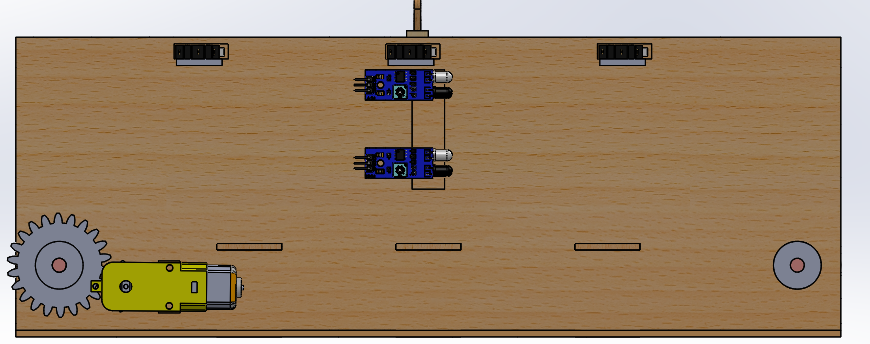
# 3. Diseño Mecánico y Electrónico del Prototipo.

## 3.1 Modelo 3D del sistema mecánico

Tomando como base los requerimientos establecidos en el apartado 1 y las especificaciones funcionales definidas en el diseño conceptual del apartado 2, así como las referencias [1], [2] y [3], se procedió a la elaboración de un modelo tridimensional del sistema mediante el software *SolidWorks*.



En el lateral derecho del modelo se incorporan tres sensores de fin de carrera, estratégicamente ubicados para detectar las posiciones inicial, intermedia y final del recipiente sobre la banda transportadora. Asimismo, se integran dos sensores infrarrojos para la identificación de la marca que indica el nivel de llenado, permitiendo distinguir entre dos alturas posibles de llenado.

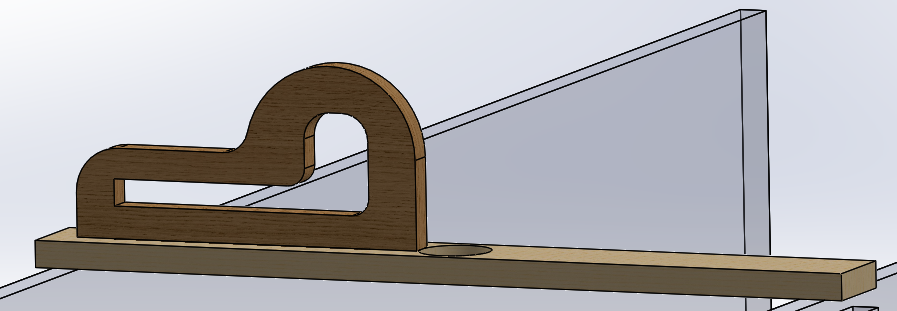


El accionamiento de la banda se logra mediante un motor acoplado a un conjunto de engranajes, los cuales transmiten el movimiento hacia el eje principal. Esta configuración permite una transmisión eficiente del torque y una velocidad controlada del desplazamiento del recipiente.

Para asegurar la estabilidad y funcionalidad del sistema, los ejes están provistos de un conjunto de ruedas que incrementan el diámetro efectivo de contacto con la banda. Además, estas ruedas limitan el desplazamiento axial de la banda, asegurando un guiado correcto durante su operación.



El proceso de llenado se realiza desde la parte superior del prototipo, mediante una manguera conectada a una bomba de agua sumergible. Esta manguera está fijada a un travesaño superior que la mantiene centrada sobre el recipiente en la zona de llenado.



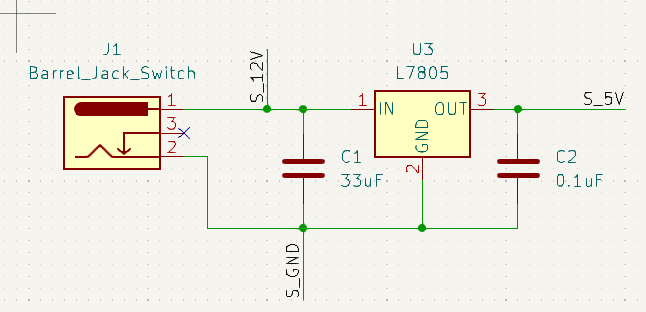
Por último, se incluye un soporte rígido debajo de la banda transportadora, el cual evita deformaciones por el peso del recipiente, especialmente cuando se encuentra lleno. Tanto las paredes laterales como el soporte estructural principal están unidos mediante uniones tipo *muesca* ajustadas a medida, garantizando la rigidez estructural del sistema. Estas uniones permiten también el desmontaje y montaje del sistema con facilidad, en caso de requerir mantenimiento o ajustes.



## 3.2 Diseño del sistema electrónico

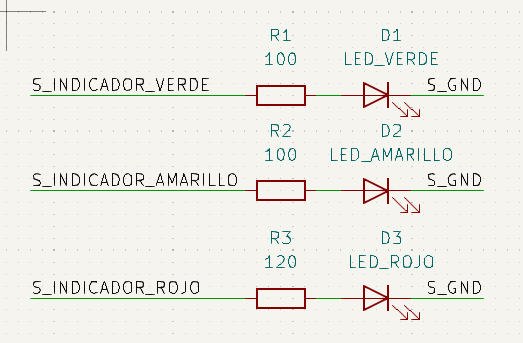
El diseño electrónico del sistema fue desarrollado utilizando el software *KiCad*, una herramienta de diseño de circuitos impresos de código abierto, interfaz amigable y su versatilidad. El circuito fue dividido en distintas etapas funcionales, con el fin de facilitar su análisis, implementación y posterior mantenimiento. A continuación, se describe cada una de las secciones que conforman el sistema electrónico.

### 3.2.1 Entrada de alimentación

La alimentación del sistema se realiza mediante una fuente externa de 12 V conectada a través de un conector tipo jack. A partir de esta entrada, se emplea un regulador de voltaje **L7805**, encargado de proporcionar 5 V estables para la mayoría de los módulos electrónicos del sistema. Dado que la **ESP32** cuenta con su propio regulador interno, esta se alimenta desde la misma línea de 5 V, permitiendo su operación a 3.3 V de manera segura y confiable

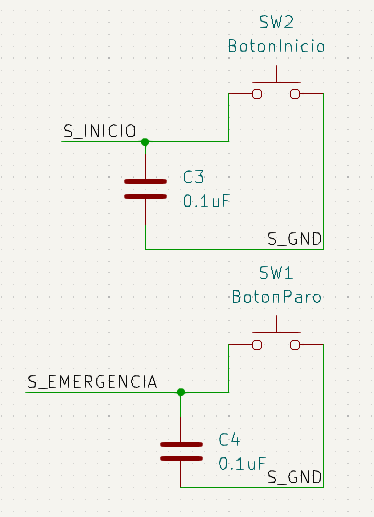
### 3.2.2 Indicadores lumínicos

Se incorporan tres indicadores LED de distintos colores: **verde**, **amarillo** y **rojo**, los cuales representan visualmente los diferentes estados del proceso. Para cada LED se realiza el cálculo correspondiente de la resistencia limitadora de corriente, tomando en cuenta la caída de voltaje específica del color seleccionado y el voltaje de alimentación. Esta etapa proporciona una retroalimentación clara al usuario sobre el estado actual del sistema.



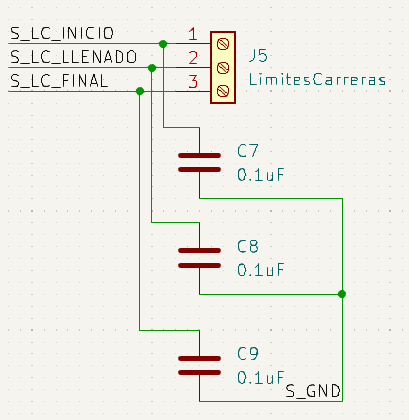
### 3.2.3 Botones de Inicio y Paro.

Los botones físicos de **inicio** y **paro de emergencia** se implementan en una configuración **Pull-Down**, con la adición de un **capacitor anti-rebote** (debounce) por hardware. Esta elección tiene como objetivo minimizar el procesamiento digital requerido en el microcontrolador, evitando el uso de rutinas de software para el tratamiento del rebote mecánico de los pulsadores.

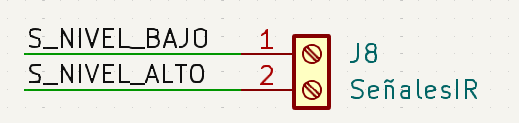


### 3.2.4 Sensores de fin de carrera

Los sensores de fin de carrera utilizados para detectar las posiciones clave del recipiente en la banda transportadora se integran mediante una lógica de conexión **Pull-Down con capacitor anti-rebote**, similar a la empleada para los botones. La señal de estos sensores se canaliza a través de **borneras**, lo que permite una conexión segura, ordenada y modular.

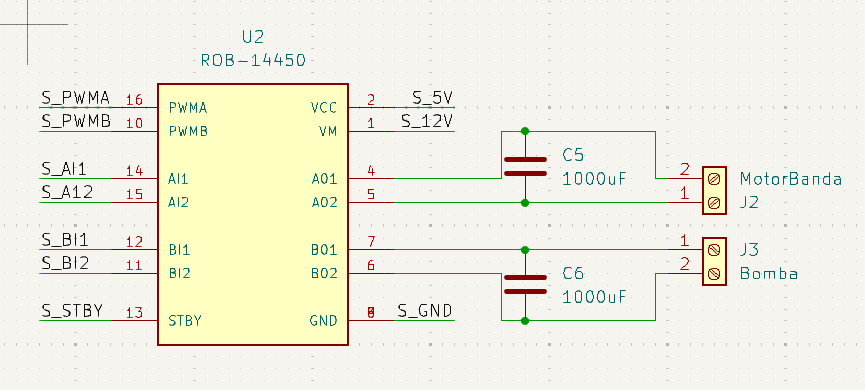


### 3.2.5 Sensores infrarrojos

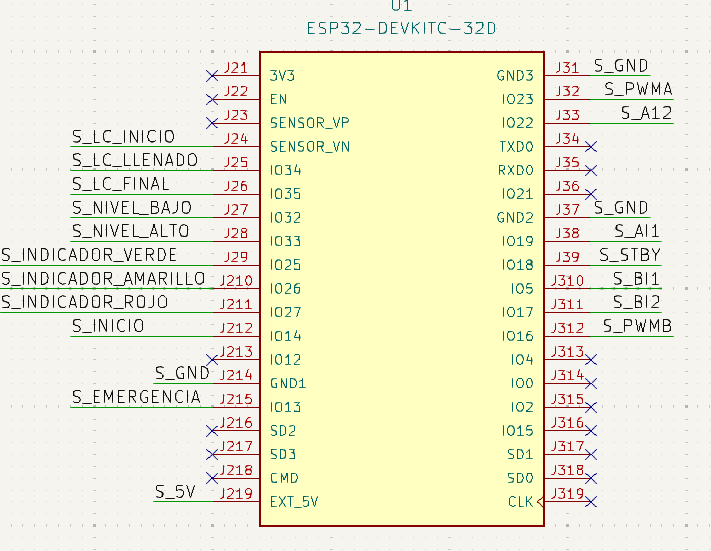
Los sensores infrarrojos encargados de detectar las marcas de nivel en el recipiente también se conectan mediante borneras. Sin embargo, debido a que su salida es de **alta impedancia** y presenta mayor estabilidad eléctrica, se prescinde del uso de capacitores anti-rebote en su configuración de entrada.

### 3.2.6 Puente H

El sistema de control de los actuadores se basa en el uso de un **módulo doble puente H TB6612FNG**, que permite el control de dos motores de forma independiente. Para cada motor se requieren dos señales digitales que determinan la dirección de rotación, y una señal **PWM** para el control de velocidad. Adicionalmente, el módulo dispone de una entrada de habilitación que permite activar o desactivar el modo de bajo consumo. Se colocan **capacitores de desacoplo** en la salida del puente H con el fin de minimizar el ruido electromagnético generado por los motores. La conexión de los motores al circuito se realiza a través de borneras, facilitando su montaje y reemplazo.



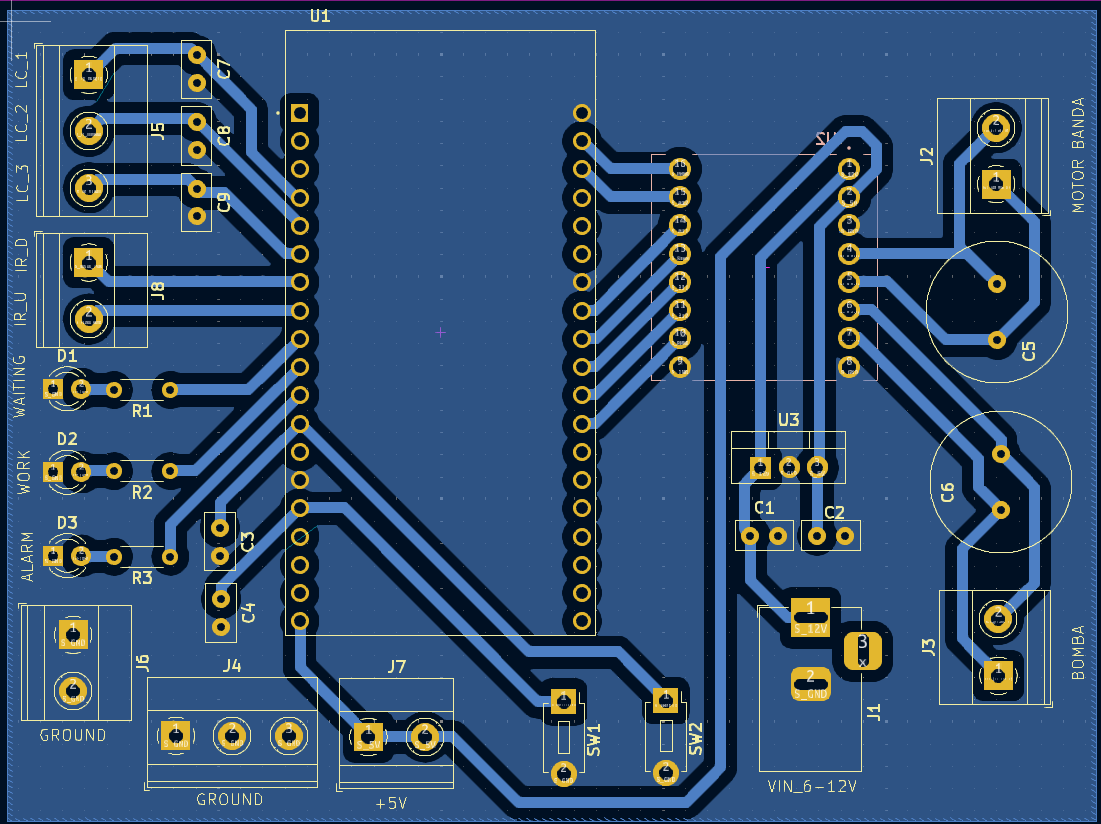
### 3.2.7 ESP 32

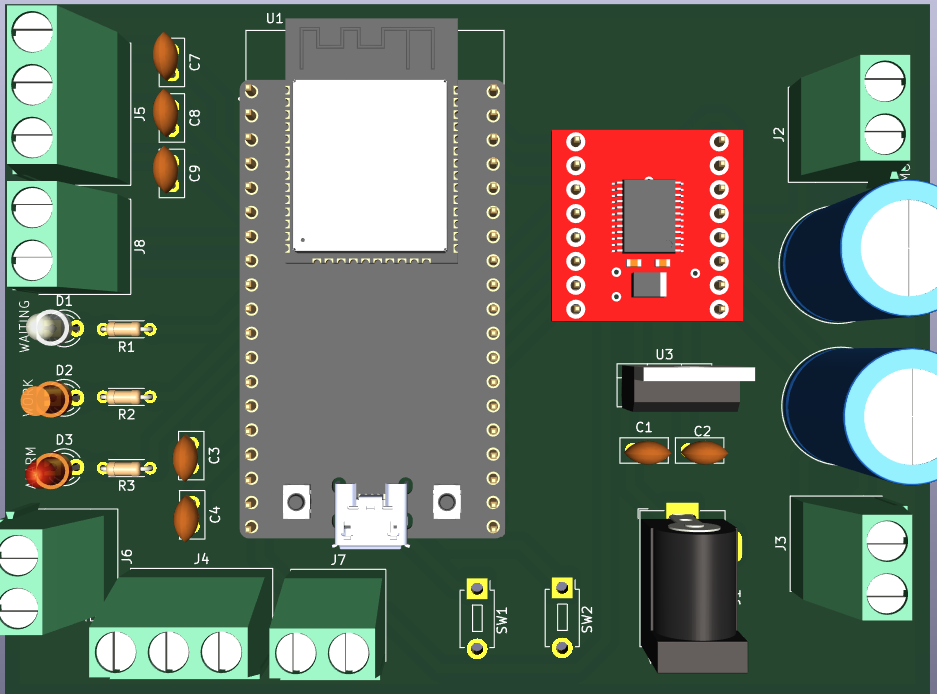
Finalmente, la integración de la **ESP32** se realiza mediante marcas de conexión directas en el esquemático, asignando los pines correspondientes a cada una de las señales provenientes de sensores, actuadores, indicadores y botones. La ESP32 actúa como el cerebro del sistema, ejecutando la lógica de control definida por la máquina de estados y coordinando el comportamiento de todos los componentes del sistema.

### 3.2.8 Diseño de PCB

Se priorizó la **optimización del espacio** y la correcta distribución de las pistas, buscando una disposición compacta y funcional de los componentes.

La disposición final de la tarjeta logró integrarse en una superficie rectangular de **7 cm × 10 cm**, permitiendo un montaje eficiente dentro del chasis del sistema. Esta reducción en el área no compromete la accesibilidad de los conectores ni la ventilación de los componentes, garantizando tanto la **seguridad operativa** como la **facilidad de mantenimiento**.





# 4. Manufacturación del prototipo

Con base en el diseño conceptual y los modelos desarrollados previamente, se procedió a la construcción del prototipo físico, integrando tanto la estructura mecánica como el sistema electrónico en una plataforma funcional. Esta etapa permitió validar la factibilidad del diseño, así como comprobar su operatividad a través de ensayos experimentales. A continuación, se describe el proceso de manufactura, dividido en dos componentes principales: la estructura mecánica y el sistema electrónico.

## 4.1. Fabricación del diseño mecánico

La fabricación de la estructura se llevó a cabo empleando técnicas de corte láser y manufactura aditiva. Los siguientes elementos fueron construidos para formar el cuerpo principal del sistema:

1. **Estructura:** El soporte de la banda transportadora, las paredes laterales, la base, el travesaño superior y las ruedas fueron cortados en láser a partir de MDF de 3 mm de espesor.
2. **Eje principal:** Se utilizó una barra sólida de acero de ¼ de pulgada de diámetro.
3. **Sistema de transmisión:** Los engranes fueron impresos en 3D utilizando material PETG, seleccionado por su resistencia mecánica y estabilidad térmica.
4. **Banda transportadora:** Se empleó una cinta de caucho de 3 pulgadas de ancho por 1/8 de pulgada de espesor.
5. **Montaje de sensores y actuadores:** Se utilizaron elementos de fijación como tornillos y adhesivo industrial para asegurar los componentes electrónicos a la estructura.

### 4.1.1 Observaciones durante las pruebas preliminares

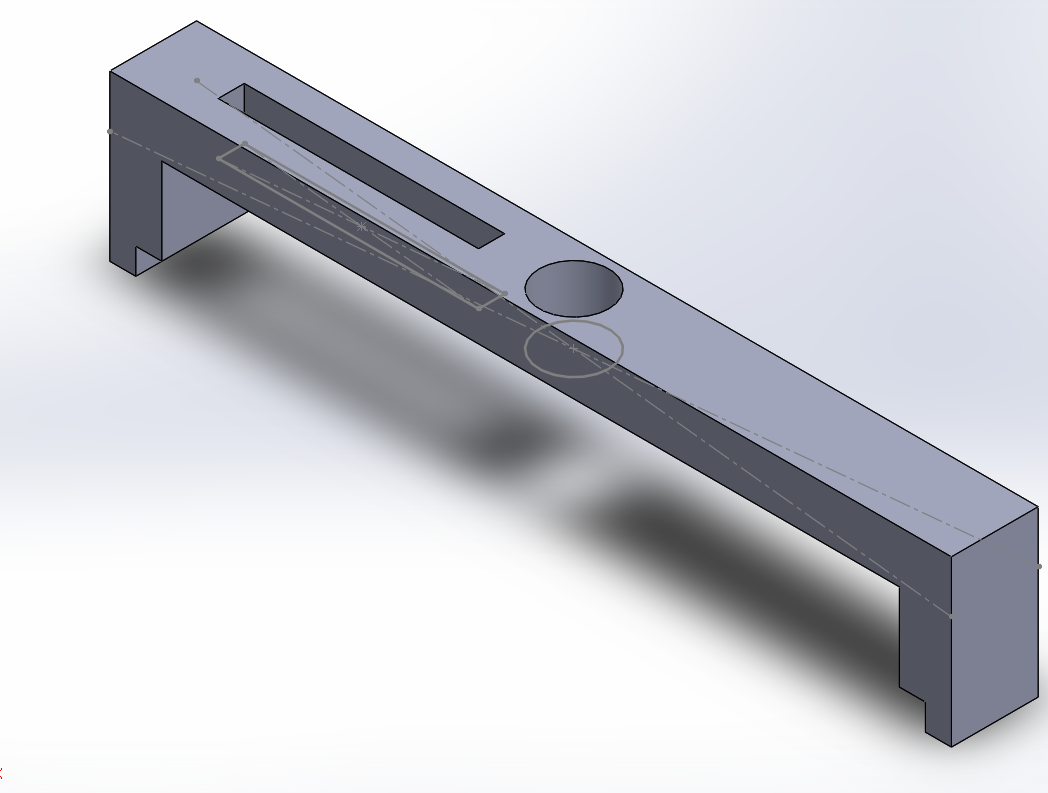
Durante las primeras pruebas de funcionamiento se identificaron las siguientes problemáticas:

1. Los sensores de fin de carrera requerían una fuerza de activación superior a la ejercida por un vaso de unicel en movimiento, provocando deformaciones o caída del recipiente.
2. El espesor de la banda elevaba el recipiente por encima del nivel previsto, lo que ocasionaba colisiones con el travesaño superior, impidiendo el llenado adecuado.
3. Los sensores infrarrojos no detectaban el punto exacto de la marca debido al ángulo de dispersión del haz de luz, generando activaciones anticipadas o retrasadas.

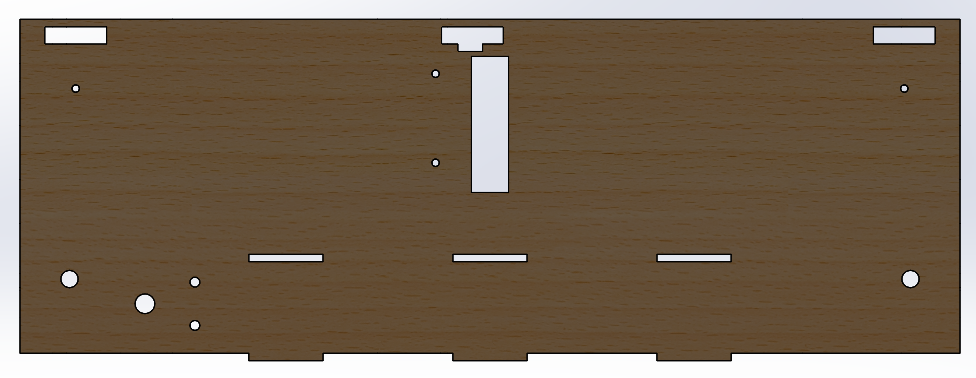
### 4.1.2 Rediseño del sistema mecánico

Con base en los hallazgos obtenidos, se realizaron las siguientes mejoras al diseño:

1. Se sustituyeron los sensores de fin de carrera por sensores infrarrojos, que no requieren contacto físico para su activación.
2. Se rediseñó e imprimió en 3D un nuevo travesaño, incrementando la altura disponible para el recipiente.



1. Se modificó el diseño de la pared lateral derecha para aumentar la separación entre los sensores infrarrojos, mejorando la precisión de detección.



## 4.2 Fabricación del diseño electrónico

El diseño electrónico fue implementado mediante el método artesanal de transferencia térmica ("método de plancha"). El procedimiento seguido fue el siguiente:

1. Impresión del circuito en papel térmico utilizando impresora láser, cuidando su correcta orientación.
2. Corte y limpieza de la placa fenólica con lana de acero para asegurar una superficie uniforme.
3. Inmersión de la placa en una solución de cloruro férrico diluido (2:1 con agua) para eliminar el cobre sobrante.
4. Perforación de la placa en los puntos correspondientes a las terminales de los componentes.
5. Limpieza del tóner residual con acetona para dejar expuestas las pistas de cobre.
6. Montaje y soldadura de los componentes electrónicos con estaño.

### 4.2.1 Observaciones durante las pruebas preliminares del circuito

Durante la integración del sistema electrónico, se presentaron los siguientes inconvenientes:

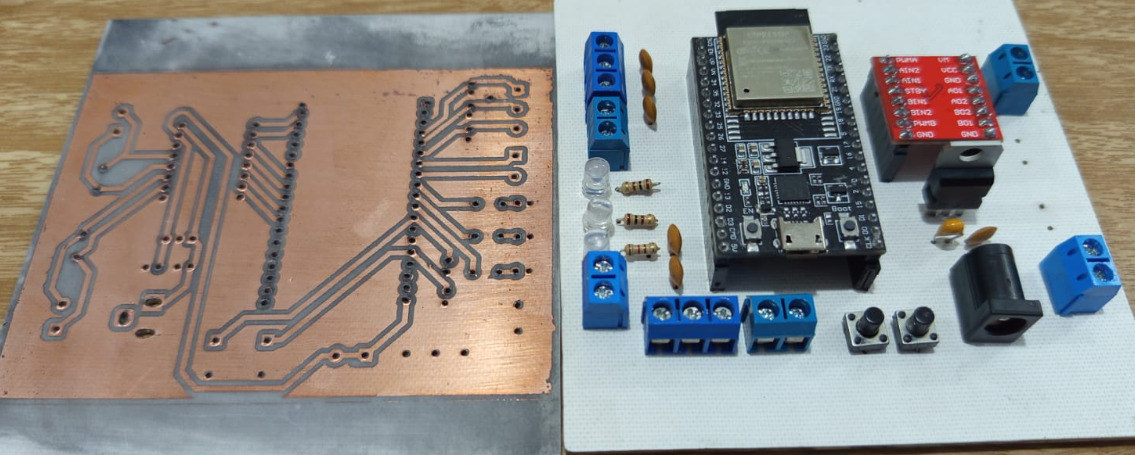
1. La inclusión de capacitores de desacoplo generaba reinicios inesperados en el módulo ESP32, posiblemente debido a corrientes transitorias o configuración inadecuada.
2. Las pistas expuestas en la placa facilitaban la ocurrencia de cortocircuitos por contacto accidental con elementos externos.

### 4.2.2 Rediseño del sistema electrónico

Con el objetivo de garantizar la funcionalidad y seguridad del sistema, se implementaron las siguientes acciones correctivas:

1. Se retiraron los capacitores de desacoplo, ya que su presencia no era indispensable para el correcto funcionamiento del prototipo.
2. Se aplicó una capa de adhesivo dieléctrico (cola loca) sobre las pistas expuestas para evitar cortocircuitos y proteger las conexiones.

Además, a pesar de que se sustituyeron los tres sensores de fin de carrera por dos sensores infrarrojos, esta modificación no implicó cambios en el diseño electrónico del prototipo. Las nuevas señales pudieron ser conectadas a las mismas terminales originalmente destinadas para los sensores mecánicos, sin afectar el funcionamiento lógico ni eléctrico del sistema. Por lo tanto, no fue necesario realizar ajustes adicionales en la placa ni en la distribución de pines.

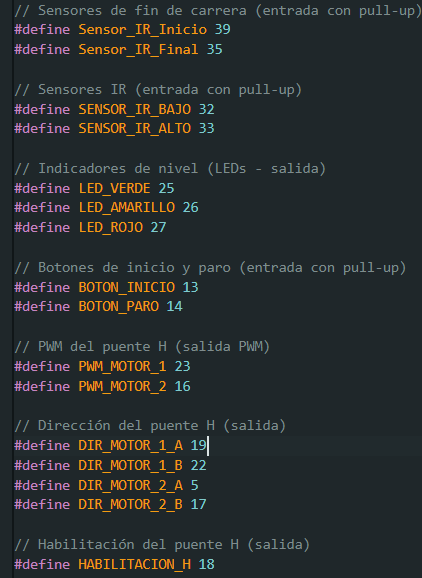


# 5. Programación del sistema de control embebido

A continuación, se describe detalladamente el código implementado en la ESP32, cuya finalidad es gestionar un sistema automático basado en una máquina de estados finitos. Esta controla la activación de motores, sensores infrarrojos, indicadores LED y botones de usuario para llevar a cabo un proceso de llenado automatizado.

## 5.1 Definición de pines

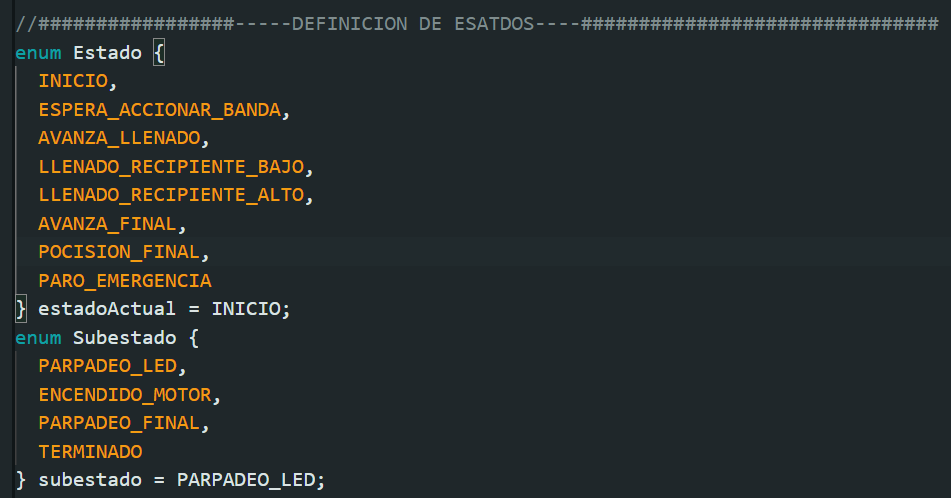
Se realiza la asignación de pines mediante directivas #define, con el objetivo de facilitar la comprensión y modificación del código. Los pines corresponden a sensores infrarrojos, botones, salidas PWM, direcciones del puente H y LEDs indicadores.



## 5.2 Definición de estados

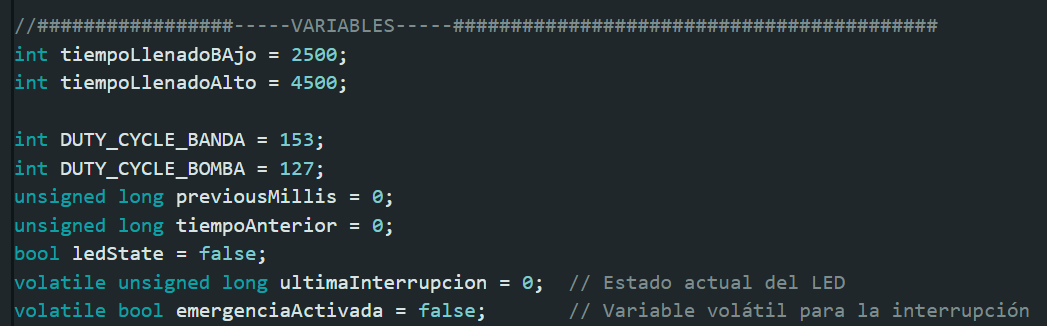
Se utilizan dos enumeraciones (enum) para representar:

* **Estados principales** del sistema (Estado): desde la inicialización hasta el paro de emergencia.
* **Subestados** (Subestado): utilizados en los procesos de llenado para indicar distintas fases como parpadeo inicial, llenado activo y transición al siguiente estado.



## 5.3 Variables globales

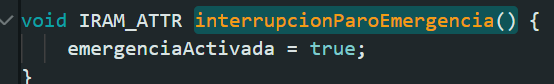
Se declaran variables para controlar los tiempos de llenado, ciclos de trabajo (duty cycles), control de LEDs, gestión de interrupciones y manejo de tiempos mediante funciones no bloqueantes (millis()).



## 5.4 Funciones de control

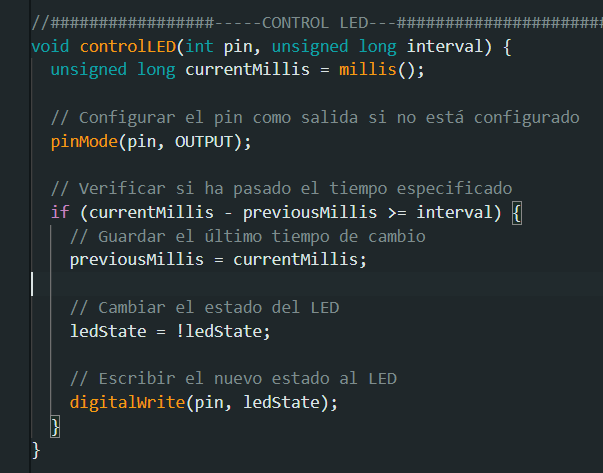
### 5.4.1 interrupcionParoEmergencia()

Función de interrupción (atributo IRAM\_ATTR) que se activa cuando se detecta una señal en el botón de paro. Cambia una bandera (emergenciaActivada) para transicionar al estado de emergencia.



### 5.4.2 controlLED()

Permite el parpadeo de un LED a un intervalo definido sin detener el programa, empleando la función millis() para gestión de tiempo no bloqueante.

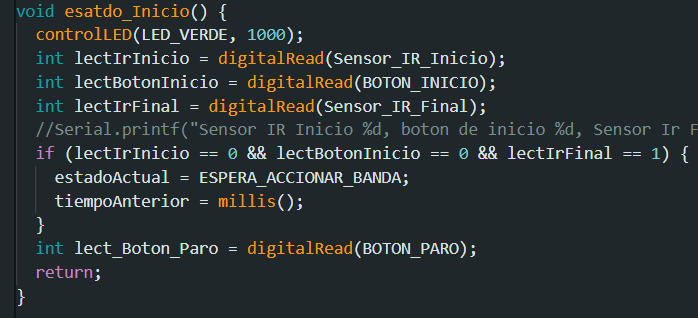


## 5.5 Funciones de estado

Cada función define el comportamiento asociado a un estado de la máquina:

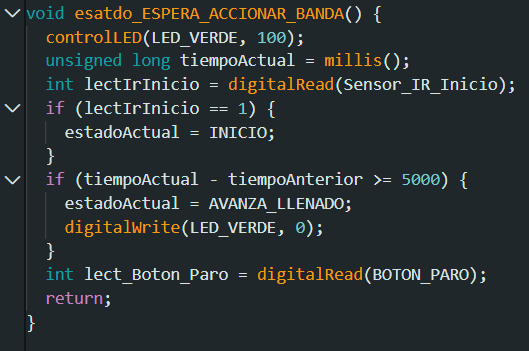
### 5.5.1 esatdo\_Inicio()

Verifica las condiciones iniciales para arrancar el proceso: el sensor de inicio debe estar activo, el botón presionado y el sensor final inactivo. Si se cumplen, se avanza al siguiente estado.



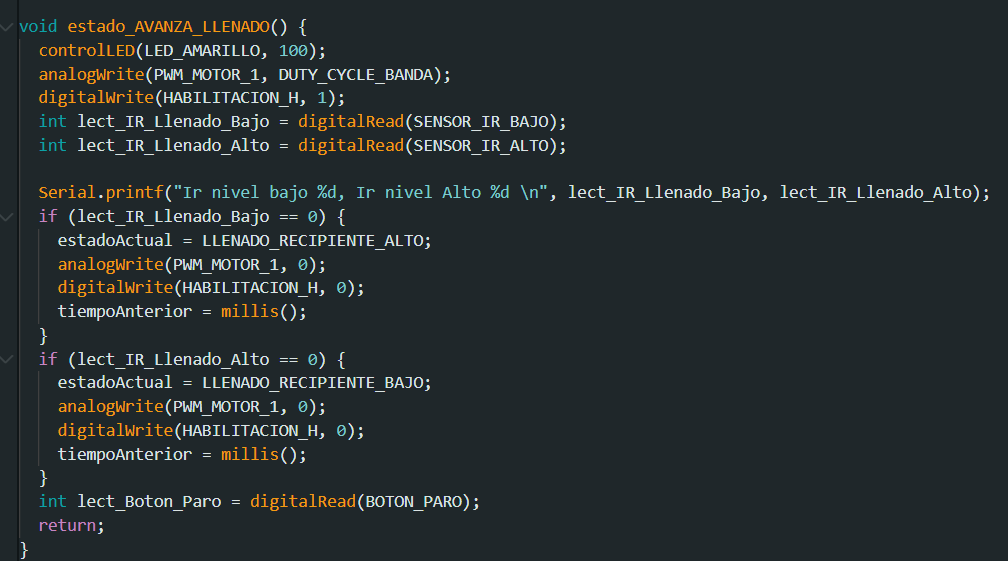
### 5.5.2 esatdo\_ESPERA\_ACCIONAR\_BANDA()

Realiza un retardo de seguridad antes de accionar la banda transportadora, monitoreando si el objeto aún se encuentra en la posición inicial. Transiciona a AVANZA\_LLENADO después de 5 segundos.



### 5.5.3 estado\_AVANZA\_LLENADO()

Activa el motor de la banda para llevar el recipiente a la estación de llenado. Según el sensor IR que detecte el recipiente, se decide entre llenar un recipiente bajo o alto.



### 5.5.4 estado\_LLENADO\_RECIPIENTE\_BAJO() y estado\_LLENADO\_RECIPIENTE\_ALTO()

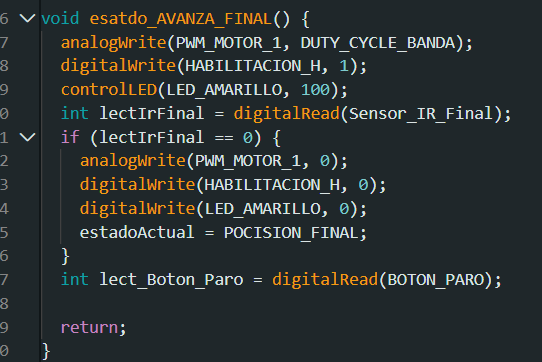
Ambas funciones son equivalentes en estructura, diferenciándose únicamente por el tiempo de llenado (2500 ms y 4500 ms, respectivamente).  
Cada una realiza:

* Parpadeo inicial del LED amarillo (estado PARPADEO\_LED)
* Activación del motor de la bomba (estado ENCENDIDO\_MOTOR)
* Parpadeo final como indicador visual (estado PARPADEO\_FINAL)
* Transición al siguiente estado (AVANZA\_FINAL)



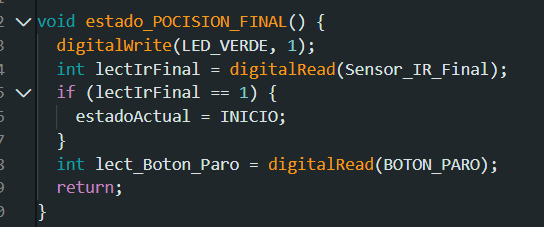
### 5.5.5 esatdo\_AVANZA\_FINAL()

Reactiva la banda para trasladar el recipiente al punto final. Cuando el sensor IR final detecta su presencia, se detiene el motor.



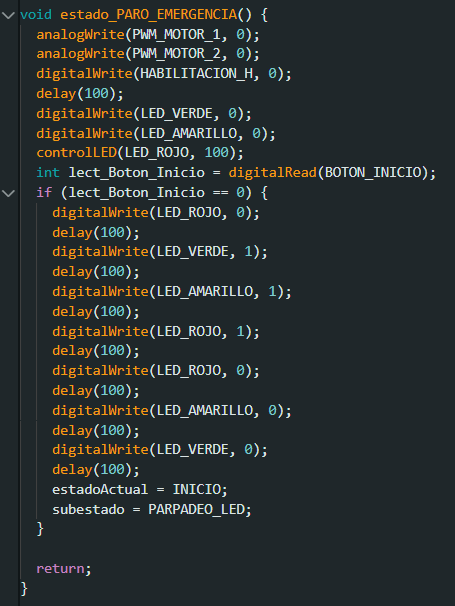
### 5.5.6 estado\_POCISION\_FINAL()

Estado de espera donde el sistema permanece hasta que el recipiente abandone la posición final. Una vez liberado el sensor, se reinicia el ciclo.

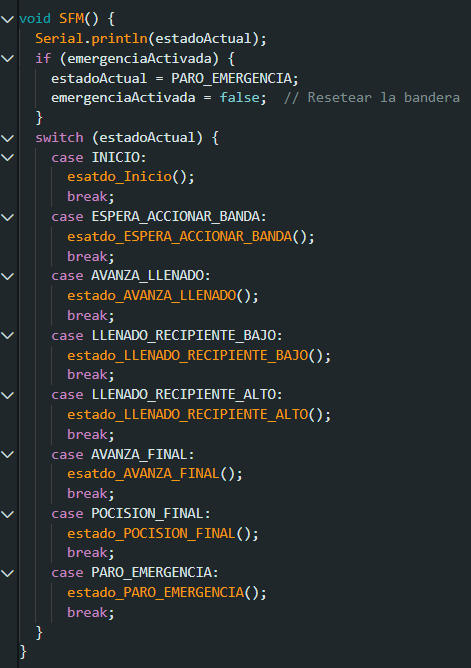


### 5.5.7 estado\_PARO\_EMERGENCIA()

Ante una condición de paro, se detienen todos los actuadores, se activa una señal visual (LED rojo) y se espera a que el botón de inicio sea presionado nuevamente para retornar al estado inicial.

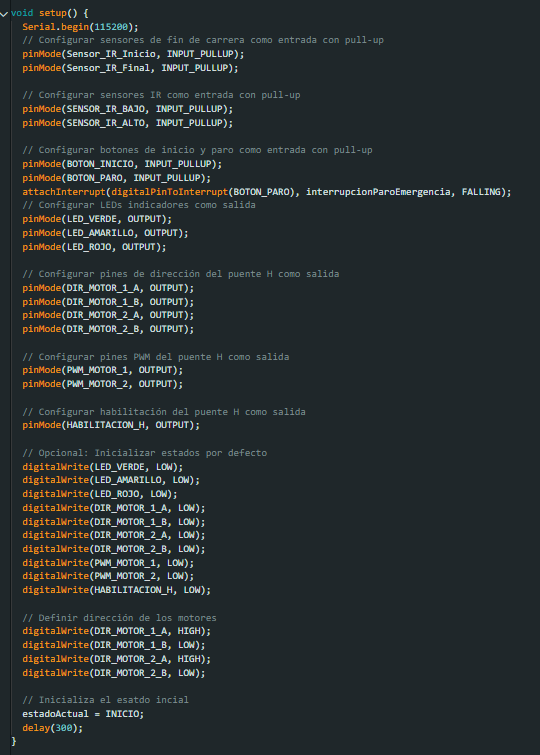


**5.5.8 SFM()**  
Esta función actúa como el núcleo de la máquina de estados (State Finite Machine). Evalúa la variable estadoActual y ejecuta la función correspondiente. Si se detecta una interrupción por paro, se prioriza la transición al estado de emergencia.



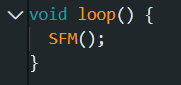
## 5.6 setup()

Inicializa todos los pines de entrada y salida. Configura las direcciones de los motores, habilita interrupciones y establece el estado inicial del sistema. También se realiza la definición de dirección de los motores para establecer sentido de giro.



**5.7 loop()**

Ejecuta continuamente la función SFM(), evaluando el estado del sistema y ejecutando las acciones correspondientes.



# 6. Conclusiones

El desarrollo e implementación del sistema de control embebido mediante una ESP32 permitió automatizar un proceso de llenado utilizando sensores, motores y actuadores controlados a través de una máquina de estados finitos. A través de este enfoque, se logró una secuencia lógica y estructurada que garantiza el correcto funcionamiento del sistema ante distintas condiciones, incluyendo la gestión de paros de emergencia.

La estructura del código empleando subestados y control no bloqueante mediante millis() demostró ser eficaz para mantener la respuesta del sistema sin depender de retardos que interrumpan la ejecución del flujo principal.

Este proyecto permitió reforzar habilidades en programación de microcontroladores, uso de sensores infrarrojos, control de motores con puente H, diseño de sistemas secuenciales y gestión de interrupciones. Además, representó una oportunidad para aplicar conceptos fundamentales de automatización y control en un entorno práctico.

Como mejora futura, se sugiere:

* Implementar una interfaz web o pantalla LCD para visualizar en tiempo real el estado del sistema.
* Agregar un sistema de registro de eventos o historial.
* Incorporar sensores adicionales para mejorar la precisión del llenado o la seguridad del proceso.

# Anexos

## Código de Diagrama 1

digraph ArquitecturaDelSistema {

    rankdir=LR;

    node [shape=box, style=filled, fillcolor=lightblue];

    subgraph cluster\_General{

    label="Arquitecura del sistema"

    subgraph cluster\_entrada {

        label="Entrada de Datos";

        boton\_inicio [label="Botón de Inicio"];

        sensor\_inicio [label="Sensor de Inicio"];

        sensor\_intermedio [label="Sensor Intermedio"];

        sensor\_final [label="Sensor Final"];

        detector\_nivel\_1 [label="Detector de Nivel 1"];

        detector\_nivel\_2 [label="Detector de Nivel 2"];

        boton\_paro\_Emergencia[label="Boton de paro de emergencia"]

    }

    subgraph cluster\_procesamiento {

        label="Procesamiento";

        microcontrolador [label="Microcontrolador", shape=ellipse, fillcolor=lightgray];

    }

    subgraph cluster\_salida {

        label="Salida de Datos";

        motor\_banda [label="Motor Banda Transportadora"];

        valvula\_dispensador [label="Bomba de Agua"];

        indicadores [label="Indicadores LED / Interfaz"];

    }

    boton\_inicio -> microcontrolador;

    sensor\_inicio -> microcontrolador;

    sensor\_intermedio -> microcontrolador;

    sensor\_final -> microcontrolador;

    detector\_nivel\_1 -> microcontrolador;

    detector\_nivel\_2 -> microcontrolador;

    boton\_paro\_Emergencia -> microcontrolador;

    microcontrolador -> motor\_banda;

    microcontrolador -> valvula\_dispensador;

    microcontrolador -> indicadores;

}

}

## Código Diagrama Maquina de Estados

subgraph cluster\_MaquinaEstados {

        label="Máquina de Estados SFM";

        graph [color=black,];

        node [shape=circle, fillcolor=lightgray];

        S1 [label="S1 \n ON=0\nOFF=1\nBT=0\nBA=0"];

        S2 [label="S2 \n ON=1\nOFF=0\nBT=0\nBA=0"];

        S3 [label="S3 \n ON=1\nOFF=0\nBT=1\nBA=0"];

        S4 [label="S4 \n ON=1\nOFF=0\nBT=0\nBA=1"];

        S5 [label="S5 \n ON=1\nOFF=0\nBT=0\nBA=1"];

        S6 [label="S6 \n ON=1\nOFF=0\nBT=1\nBA=0"];

        S7 [label="S7 \n ON=0\nOFF=1\nBT=0\nBA=0"];

        S8 [label="S8 \n ON=0\nOFF=1\nBT=0\nBA=0"];

        S1 -> S1 [label="B1 != 1 || POS1 != 1 || POS3 != 1"];

        S1 -> S2 [label="B1 = 1 && POS1 = 1"];

        S2 -> S1 [label="T < 5s && (B1 != 1 || POS1 != 1)"];

        S2 -> S2 [label="T < 5s && B1 = 1 && POS1 = 1"];

        S2 -> S3 [label="T > 5"];

        S3 -> S3 [label="POS2 != 0"];

        S3 -> S4 [label="POS2 = 1 && LD = 1"];

        S4 -> S4 [label="T < TD"];

        S4 -> S6 [label="T > TD"];

        S3 -> S5 [label="POS2 = 1 && LU = 1"];

        S5 -> S5 [label="T < TU"];

        S5 -> S6 [label="T > TU"];

        S6 -> S6 [label="POS3 != 1"];

        S6 -> S7 [label="POS3 = 1"];

        S7 -> S7 [label="POS3 = 1"];

        S7 -> S1 [label="POS3 != 1"];

        S1 -> S8 [label="PE = 1"];

        S2 -> S8 [label="PE = 1"];

        S3 -> S8 [label="PE = 1"];

        S4 -> S8 [label="PE = 1"];

        S5 -> S8 [label="PE = 1"];

        S6 -> S8 [label="PE = 1"];

        S7 -> S8 [label="PE = 1"];

        S8 -> S8 [label="B1 != 1"];

        S8 -> S1 [label="B1 = 1"];

    }

## Código Tabla representativa máquina de estados

digraph Nomenclatura {

    rankdir=LR;  // Organiza de izquierda a derecha

    nodesep=0.6; // Espaciado horizontal entre nodos

    ranksep=0.8; // Espaciado vertical entre niveles

    node [shape=box, style=filled, fillcolor=lightblue];

    // Significados

    subgraph cluster\_Significados {

        label="Significados";

        graph [style=filled, color=gray85, fontname="Arial", fontsize=12];

        // Sensores

        subgraph cluster\_Sensores {

            label="Sensores";

            graph [style=filled, color=gray90];

            boton\_inicio [label="B1: Botón de inicio"];

            posicion\_inicio [label="POS1: Sensor de posición inicial"];

            posicion\_llenado [label="POS2: Sensor de posición de llenado"];

            posicion\_final [label="POS3: Sensor de posición final"];

            sensor\_nivel\_bajo [label="LD: Sensor de nivel bajo"];

            sensor\_nivel\_alto [label="LU: Sensor de nivel alto"];

            paro\_emergencia [label="PE: Botón de paro de emergencia"];

        }

        // Estados

        subgraph cluster\_Estados {

            label="Estados";

            graph [style=filled, color=gray90];

            Estado1 [label="S1: Estado inicial"];

            Estado2 [label="S2: Espera para accionar banda"];

            Estado3 [label="S3: Avance a la posición de llenado"];

            Estado4 [label="S4: Llenado de recipiente - Nivel bajo"];

            Estado5 [label="S5: Llenado de recipiente - Nivel alto"];

            Estado6 [label="S6: Avance a la posición final"];

            Estado7 [label="S7: Recipiente en posición final"];

            Estado8 [label="S8: Paro de emergencia"];

        }

        // Variables

        subgraph cluster\_Variables {

            label="Variables";

            graph [style=filled, color=gray90];

            tiempo [label="T: Tiempo transcurrido"];

            tiempo\_bajo [label="TD: Tiempo de llenado - Nivel bajo"];

            tiempo\_alto [label="TU: Tiempo de llenado - Nivel alto"];

        }

        // Salidas

        subgraph cluster\_Salidas {

            label="Salidas";

            graph [style=filled, color=gray90];

            indicador\_on [label="ON: Indicador de banda en movimiento"];

            indicador\_off [label="OFF: Indicador de banda apagada"];

            activacion\_transportadora [label="BT: Banda transportadora"];

            activacion\_bomba\_agua [label="BA: Bomba de agua"];

        }

    }

    boton\_inicio->Estado1[style=invis];

    Estado1->tiempo[style=invis];

    tiempo->indicador\_on[style=invis];

}

## Modelos 3D a utilizar

Sensor Infrarrojo:

<https://grabcad.com/library/fc-51-1>

Puente H:

<https://grabcad.com/library/puente-h-l298-1>

Limit Switch:

<https://grabcad.com/library/kw12-3-no-nc-bump-switch-limit-switch-1>

Motoreductor:

<https://grabcad.com/library/motorreductor-1-48-1>

Vaso Referencia:

<https://sketchfab.com/3d-models/styrofoam-cups-bd449b0eadea43c0aaaef933b7070c60>

# Referencias Bibliográficas

[1]<https://www.youtube.com/watch?v=HmmEi8s1oJ4>

[2]<https://www.youtube.com/watch?v=eww3crwOPv8>

[33]<https://www.youtube.com/watch?v=WRdwi3lZ08o>