

UNIVERSIDAD NACIONAL JOSE FAUSTINO SANCHEZ CARRION



DISPENSADOR AUTOMATICO DE COCTELES - BARBOT

DOCENTE:

ING. DIAZ RONCEROS ERNESTO

CURSO:

SISTEMAS EMBEBIDOS

INTEGRANTES:

OLORTEGUI PAJUELO YOBER

PORTOCARRERO REYES JIAN PHIE

HUACHO – 2025

Resumen

El presente proyecto tiene como objetivo el diseño y construcción de una máquina automatizada capaz de preparar cocteles de manera eficiente y controlada mediante el uso de un microcontrolador Arduino. La motivación principal surge de la necesidad de reducir el contacto humano en la preparación de bebidas, especialmente en contextos donde la higiene es prioritaria. Para ello, se desarrolló un prototipo denominado “BarBot”, el cual emplea bombas peristálticas para dispensar líquidos desde distintos recipientes según recetas programadas. El sistema fue implementado utilizando componentes electrónicos de bajo costo, y se programó una interfaz de usuario simple para la selección de bebidas. La metodología incluyó el diseño del circuito, la programación del algoritmo de mezclado, y la fabricación de una estructura de soporte. Se realizaron pruebas de funcionamiento que evidenciaron un tiempo de dispensado promedio de 15 segundos por bebida y una precisión aceptable en la dosificación. Los resultados validan la viabilidad del prototipo como una solución funcional para entornos domésticos o eventos sociales. Se concluye que la automatización en la preparación de bebidas puede mejorar la eficiencia operativa y minimizar riesgos sanitarios, aunque se identifican oportunidades de mejora en aspectos como la velocidad de servicio y la ampliación de recetas disponibles.

I.	Introducción	5
II.	Antecedentes	5
2.1.	DESARROLLO DE UN DISPENSADOR DE TRAGOS AUTOMATIZADO PARA ESTANDARIZAR EL SABOR EN LA PREPARACIÓN DE CÓCTELES (Perú, Trujillo, 2021)	5
2.2.	DISEÑO UN SISTEMA AUTOMATICO PARA LA PREPARACION DE COCTELES (Perú, Lima, 2025)	6
2.3.	ACTUALIZACIÓN DE LA INTERFAZ GRÁFICA Y PÁGINA WEB DE UN DISPENSADOR DE BEBIDAS ALCOHÓLICAS CON TECNOLOGÍA IOT (Colombia, Bogotá, 2024).....	7
2.4.	MÁQUINA COCTELERA BARTENDING 1.0 (Colombia, Bogotá, 2020)	8
III.	Justificación	8
IV.	Objetivos	9
4.1.	Objetivo General:.....	9
4.2.	Objetivos Específicos:	9
V.	Hipótesis	10
VI.	Marco teórico	12
6.1.	Dispensador automático de vino por copas (España, Cuscurrita de Rio Tirón, 2011) 12	
6.2.	The Inebriator (EE.UU, Chicago, 2012).....	12
6.3.	Beermatic (Japón, Hoshizaki, 2020).....	12
6.4.	Control de motores y programación	13
VII.	Metodología	13

7.1.	Investigación del mercado:	13
7.2.	Diseño del prototipo:.....	15
7.3.	Lista de materiales:	16
7.4.	Construcción del prototipo:.....	19
7.5.	Etapas del proceso:	23
7.6.	Tipo, nivel, enfoque y diseño de la investigación.....	24
VIII.	Diseño electrónico para la adquisición de señales.....	25
8.1.	Definición de código:.....	26
8.2.	Selección de cocteles	28
8.3.	Código.....	32
8.4.	Diseño final del prototipo	51
IX.	Resultados e impactos esperados	52
X.	Cronograma de actividades.....	53
XI.	Presupuesto	54
11.1.	Presupuesto de materiales y componentes.....	54
XII.	Referencias bibliográficas.....	55

I. Introducción

El presente proyecto surge como una respuesta a los problemas que enfrentan los barman en espacios de entretenimiento como bares y discotecas, o en una reunión en casa, donde la demanda elevada puede generar errores en la preparación de bebidas, desperdicio de ingredientes y pérdida de calidad en el servicio. El barman, siendo la cara visible del establecimiento, requiere de apoyo tecnológico para mantener la calidad y la eficiencia. En ese sentido, el diseño de un dispensador automático de cócteles se presenta como una solución viable.

Este proyecto se fundamenta en el uso de sistemas embebidos con esp32, motores paso a paso, uso de software para móviles y se busca es unas estructuras mecánicas automatizadas que imitan la labor del barman, garantizando la dosificación precisa de ingredientes.

II. Antecedentes

2.1. DESARROLLO DE UN DISPENSADOR DE TRAGOS AUTOMATIZADO PARA ESTANDARIZAR EL SABOR EN LA PREPARACIÓN DE CÓCTELES (Perú, Trujillo, 2021)

En 2021, un equipo de investigadores de la Universidad Privada del Norte desarrolló un prototipo de dispensador de tragos automatizado con el objetivo de estandarizar el sabor en la preparación de cócteles en discotecas y bares de Trujillo, Perú. El sistema fue diseñado para preparar automáticamente tres tipos de cócteles: “Cuba Libre” (ron con cola), “Aleluya” (vodka con refresco de naranja) y “Canario” (pisco con gaseosa Evervess), utilizando válvulas dosificadoras de 1.5 onzas que garantizan medidas precisas y repetibles.

El prototipo incorporó componentes electrónicos como Arduino Mega, servomotores y sensores, así como una interfaz gráfica mediante una pantalla táctil Nextion de 3.2 pulgadas. Además, se implementó un sistema mecánico que incluye un carrito motorizado para el desplazamiento del vaso a través de las diferentes estaciones de preparación, incluyendo la dispensación de hielo.

Para evaluar la efectividad del sistema, se realizaron pruebas cualitativas utilizando cuatro grupos focales, los cuales confirmaron que el sabor de los cócteles se mantenía consistente en cada preparación. Los resultados demostraron una estandarización del sabor del 100%, evidenciando la precisión en las medidas de los ingredientes y la eficiencia en la reducción de tiempos de preparación.

Este proyecto destaca la viabilidad de aplicar la automatización en el ámbito de la coctelería para mejorar la calidad y consistencia de las bebidas servidas, así como para optimizar los procesos en establecimientos con alta demanda.

2.2.DISEÑO UN SISTEMA AUTOMATICO PARA LA PREPARACION DE COCTELES (Perú, Lima, 2025)

En el contexto del sector de entretenimiento en el Perú, que cuenta con aproximadamente 75,500 establecimientos donde se sirven alimentos y bebidas, se identificó una problemática específica en una discoteca de Lima llamada Dizco.

Durante eventos de alta demanda, se observó que los bartenders experimentaban niveles elevados de estrés al preparar hasta 63 cócteles por hora, lo que podía derivar en problemas de salud para el personal, conflictos con los clientes, pérdidas económicas y una disminución en la calidad de las bebidas servidas.

Ante esta situación, se propuso el diseño de un sistema automático para la preparación de cócteles, con el objetivo principal de asistir al bartender y reducir los tiempos de preparación durante períodos de alta demanda. El proyecto se desarrolló utilizando la "Metodología de diseño para Sistemas Mecatrónicos", fruto de un convenio entre la PUCP y la Universidad Federal de Santa Catarina. El sistema diseñado logró reducir significativamente los tiempos de preparación de cócteles, alcanzando una disminución de hasta 8 veces en comparación con el método manual. Por ejemplo, el cóctel "Whiskey Fix" pasó de prepararse en 40 segundos a solo 5 segundos, y el "Chilcano" de 1 minuto a 7.5 segundos.

2.3.ACTUALIZACIÓN DE LA INTERFAZ GRÁFICA Y PÁGINA WEB DE UN DISPENSADOR DE BEBIDAS ALCOHÓLICAS CON TECNOLOGÍA IOT (Colombia, Bogotá, 2024)

Este proyecto se centró en la mejora de un sistema automatizado de dispensación de bebidas alcohólicas, mediante la implementación de una interfaz gráfica más intuitiva y el desarrollo de una página web interactiva, ambas basadas en tecnología IoT. El objetivo fue mejorar la experiencia del usuario, permitir el control remoto del dispositivo, y optimizar la funcionalidad del sistema a través de conectividad en tiempo real. Se utilizaron plataformas como Node-RED, servidores locales, y módulos de comunicación inalámbrica. Este trabajo evidencia la integración entre hardware y software en sistemas automatizados de uso cotidiano, sirviendo como referente para el diseño y control de dispositivos electrónicos en tiempo real.

2.4. MÁQUINA COCTELERA BARTENDING 1.0 (Colombia, Bogotá, 2020)

En esta tesis, se desarrolló un sistema automatizado para la preparación de cócteles, denominado "Bartending 1.0". El proyecto integró componentes electrónicos y mecánicos, como microcontroladores, bombas dosificadoras, sensores y una interfaz de usuario, permitiendo la selección y mezcla precisa de bebidas a partir de recetas preprogramadas.

El objetivo principal fue garantizar una dosificación exacta de ingredientes, reducir la intervención humana y ofrecer una experiencia uniforme en cada preparación. Además, se promovió el uso de tecnologías accesibles y escalables, demostrando el potencial de la ingeniería mecatrónica en la automatización de tareas específicas dentro del sector gastronómico.

Este trabajo representa un aporte importante como antecedente de referencia para proyectos que combinan control electrónico, automatización y diseño funcional, principios también aplicables en el desarrollo de sistemas como los inhibidores de frecuencia, especialmente en lo que respecta a la integración precisa de componentes electrónicos para cumplir funciones específicas.

III. Justificación

El desarrollo de nuevas tecnologías orientadas a resolver problemas específicos representa una gran oportunidad de mejora en diversos sectores. En este caso, se aborda la necesidad de los bartenders de elaborar cócteles de calidad de forma precisa y eficiente. Factores como el cansancio, la subjetividad o las condiciones del entorno pueden afectar la exactitud en la dosificación de ingredientes, lo que impacta directamente en la calidad del producto final.

Por ello, se propone el diseño e implementación de un sistema automatizado para la preparación de cócteles, que brinde apoyo a los profesionales del rubro, permitiéndoles agilizar su trabajo con mayor precisión, evitando desperdicios y asegurando la estandarización del sabor. Este sistema no solo beneficiará a los bartenders, sino también a los establecimientos de entretenimiento, mejorando su productividad, y a los consumidores finales, garantizando una experiencia más consistente.

Adicionalmente, al automatizar el proceso, se contribuye significativamente a la reducción de costos operativos, al minimizar el desperdicio de insumos y optimizar el tiempo de preparación. Esto convierte al sistema en una herramienta atractiva no solo para grandes negocios, sino también para emprendedores y estudiantes del área electrónica interesados en soluciones tecnológicas con impacto práctico y económico.

IV. Objetivos

4.1.Objetivo General:

- Diseñar un sistema automatizado para la preparación de cócteles, que permita la gestión estructurada de ingredientes mediante un control eficiente.
- Implementar un dispensador automático que asegure el llenado preciso de líquidos, optimizando la dosificación y reduciendo el margen de error humano.

4.2.Objetivos Específicos:

- Analizar los requerimientos funcionales para la preparación automatizada de cócteles.
- Desarrollar un diseño estructural que permita la organización eficiente de los ingredientes.

- Seleccionar componentes electrónicos y mecánicos adecuados para la automatización del sistema.
- Diseñar una interfaz gráfica que facilite la selección personalizada de cócteles por parte del usuario.
- Integrar sensores y actuadores que permitan el control exacto del volumen dispensado.
- Programar un sistema de control que automatice el proceso de dosificación de líquidos.
- Realizar pruebas de precisión para validar el funcionamiento del sistema de llenado.
- Optimizar el sistema para reducir errores humanos y desperdicio de insumos durante la preparación.

V. Hipótesis

- Si se diseña un sistema automatizado con un control estructurado de ingredientes, entonces se logrará una preparación de cócteles más ordenada y eficiente, reduciendo la variabilidad en el sabor y la preparación.
- Si se implementa un dispensador automático con control de volumen, entonces se obtendrá una dosificación precisa de los líquidos, lo cual permitirá estandarizar el sabor y minimizar errores humanos en la preparación.
- La identificación precisa de los requerimientos funcionales permitirá establecer las bases técnicas necesarias para desarrollar un sistema automatizado eficaz en la preparación de cócteles.

- Un diseño estructural bien planteado facilitará la disposición lógica y accesible de los ingredientes, mejorando la eficiencia del sistema durante la preparación de bebidas.
- La correcta elección de los componentes electrónicos y mecánicos garantizará la fiabilidad, precisión y durabilidad del sistema automatizado de preparación de cócteles.
- Una interfaz gráfica intuitiva y funcional permitirá al usuario seleccionar fácilmente los cócteles, mejorando la experiencia de uso y reduciendo los errores operativos.
- La integración de sensores y actuadores calibrados permitirá un control preciso del volumen de líquido dispensado, asegurando uniformidad en las preparaciones.
- Un sistema de control debidamente programado automatizará eficientemente la dosificación de líquidos, reduciendo la intervención humana y mejorando la consistencia del producto final.
- Las pruebas de precisión permitirán validar el correcto funcionamiento del sistema de llenado, demostrando su capacidad para operar con exactitud en condiciones reales.
- La optimización del sistema automatizado disminuirá significativamente los errores humanos y el desperdicio de insumos, mejorando el rendimiento y la sostenibilidad operativa del dispensador.

VI. Marco teórico

6.1. Dispensador automático de vino por copas (España, Cuscurrita de Rio Tirón, 2011)

Están diseñados para conservar y servir vinos por copas con las últimas tecnologías a la temperatura ideal de servicio, disponiendo de su copa de vino para ser servida y degustada en cualquier momento, en perfectas condiciones, para lo cual se utiliza gas inerte, Nitrógeno que evita la oxidación del vino, manteniendo todas las propiedades incluso después de varios días de abierta la botella (hasta 21 días después). Evitando que el vino altere sus características cuando no es consumida la botella en su totalidad. De esta forma en un restaurante, vinoteca, tienda de vino, en las degustaciones y catas nos permite un incremento del volumen de vino vendido y un aumento del precio en cada copa vendida, rentabilizando al máximo cada botella. Además, nos permite disfrutar una excelente selección de vinos con el maridaje del plato elegido.

6.2. The Inebriator (EE.UU, Chicago, 2012)

Pretende servir de forma automática cubalibres. Todo el sistema se basa en un rail de movimiento lineal controlado mediante un Arduino Mega que mediante la ayuda de una pantalla permite seleccionar la bebida y todo el conjunto se encarga de desplazar el vaso a las distintas botellas repartidas a lo largo de un soporte. (Santamaría, P.,30 agosto 2012) Inebriator, Xataka Smart home.

6.3. Beermatic (Japón, Hoshizaki, 2020)

Este nuevo dispositivo tiene una capacidad de dispensado de 65 litros, lo que significa que puede servir hasta 130 jarras de medio litro de forma ininterrumpida y a la

temperatura perfecta, con tan solo pulsar un botón. Cuenta con dos grifos alimentados por dos circuitos independientes que provienen de dos barriles distintos, y es capaz de servir dos jarras de medio litro en 14 segundos, un 50% más rápido que el sistema convencional. El diseño compacto de Beermatic está pensado para adaptarse a todo tipo de entornos: bares, restaurantes, eventos, ferias cerveceras y caterings. Para más flexibilidad, este dispensador está equipado con un sistema de inclinación ajustable que es compatible con un amplio abanico de copas, vasos y jarras. (Profesional Horeca, noviembre 2020).

6.4.Control de motores y programación

El uso de estos sistemas permite el control preciso de motores y actuadores, lo que garantiza la dosificación exacta y la automatización del movimiento de vasos y dispensadores. La programación en Arduino IDE permite controlar todo el proceso desde la selección de la bebida hasta su entrega final.

VII. Metodología

Para el desarrollo del proyecto se seguirían las siguientes etapas metodológicas:

7.1.Investigación del mercado:

La tendencia hacia la automatización en el sector de alimentos y bebidas ha ido en aumento en los últimos años, particularmente en bares, restaurantes y eventos sociales donde se busca eficiencia, consistencia en el servicio y reducción de costos operativos. En este contexto, los dispensadores automáticos de bebidas han ganado terreno como una solución práctica, innovadora y atractiva para el cliente final.

En mercados internacionales, proyectos como The Inebriator o Bartending Machine han demostrado que existe un nicho interesado en máquinas automatizadas que

preparen cócteles de manera estandarizada. En el ámbito local, aún se encuentran pocos dispositivos que combinen automatización, conectividad Wifi y una interfaz web para selección de bebidas, lo cual representa una oportunidad de innovación tecnológica aplicable a hoteles, bares temáticos, eventos corporativos, ferias y más.

Además, el enfoque de este proyecto se alinea con las demandas del consumidor moderno, que valora:

- Experiencias interactivas, como seleccionar bebidas desde su celular o tablet.
- Velocidad en el servicio, especialmente en contextos de alta afluencia.
- Consistencia en sabor y presentación, gracias a la dosificación precisa de los ingredientes.

Por otro lado, el análisis de productos comerciales existentes muestra que muchos sistemas automatizados tienen precios elevados debido a su complejidad mecánica o su dependencia de software propietario. En conclusión, este prototipo está diseñado con componentes accesibles (ESP32, motores L298N, interfaz web desarrollada en HTML/JS) lo que lo convierte en una alternativa de bajo costo y adaptable para desarrolladores, emprendimientos o pequeñas empresas.

7.2. Aspectos legales considerados

Se analizaron leyes y normas relacionadas con la manipulación y expendio de bebidas alcohólicas, como la Ley N.º 28681 - Ley que regula la comercialización, consumo y publicidad de bebidas alcohólicas, Ley N.º 27972 - Ley Orgánica de Municipalidades.

Se identificaron requerimientos legales que podrían influir en el diseño y operación del prototipo.

7.3.Diseño del prototipo:

Se elaborará un diseño funcional en bloques que describe cada una de las etapas necesarias para la automatización del proceso de preparación de bebidas, comenzando desde la selección de cócteles hasta la ejecución física del servido. Este diseño busca integrar componentes electrónicos, mecánicos y de software para lograr un prototipo funcional, controlado de forma remota mediante Wifi.

1. Selección de Cócteles (Interfaz Web)

- El usuario accede al BarBot mediante una interfaz web desde su celular, tablet o PC.
- En esta interfaz se muestran las recetas disponibles, agrupadas por categoría (con y sin alcohol).
- Al seleccionar una receta, se envía la orden al microcontrolador (ESP32) a través de una petición HTTP.

2. Confirmación y Procesamiento

- El ESP32 valida que esté libre (estado IDLE) y que la receta exista en el archivo recipes.json almacenado en la tarjeta SD.
- Se extrae la información de la receta (ingredientes, cantidades y dispensadores involucrados).

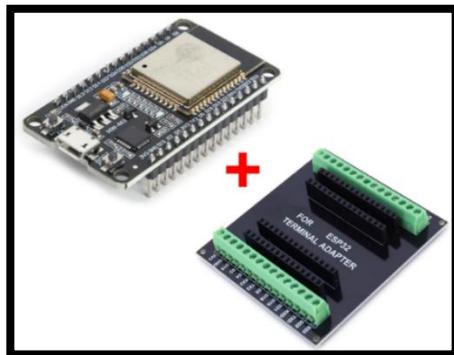
3. Ejecución del Proceso de Dosificación

- Movimiento Horizontal (base giratoria): Un motor paso a paso mueve el vaso hacia el dispensador correspondiente. Se calcula el número de pasos necesarios para cada posición.

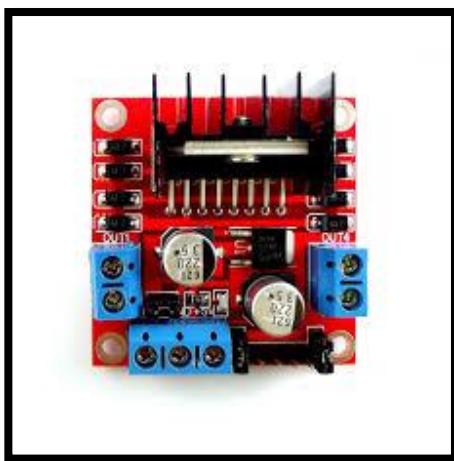
- Dosificación Vertical: Otro motor paso a paso baja el dispensador sobre el vaso para verter la cantidad especificada (en onzas), sube nuevamente y se desplaza al siguiente dispensador.
- Al finalizar la receta, el sistema espera unos segundos, regresa a la posición inicial y desactiva los motores.

7.4.Lista de materiales:

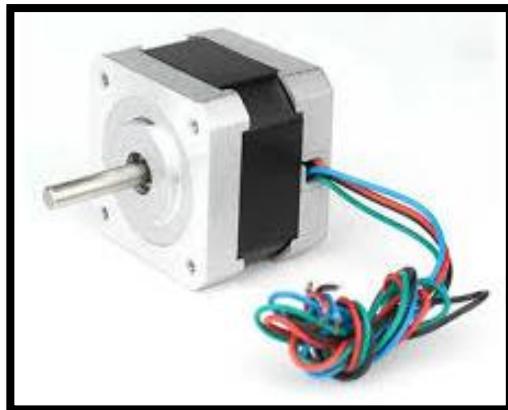
Se confeccionó una lista de todos los componentes electrónicos, estructurales y de fijación requeridos, tales como el ESP-32, drivers L298N, motores NEMA 17, dosificadores, perfiles de aluminio, cables, tornillos, entre otros.



ESP32 CON MODULO



DRIVER PUENTE H L298N



MOTOR PASO A PASO NEMA 17



SOPORTE PARA DISPENSADOR DE BEBIDAS



PERFILES DE ALUMINIO



TRANSFORMADOR DE 220 V AC A 12 V DC

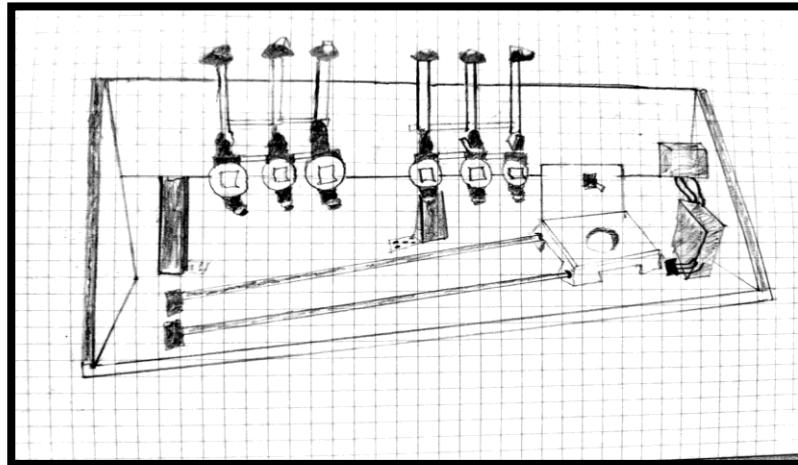


FINAL DE CARRERA

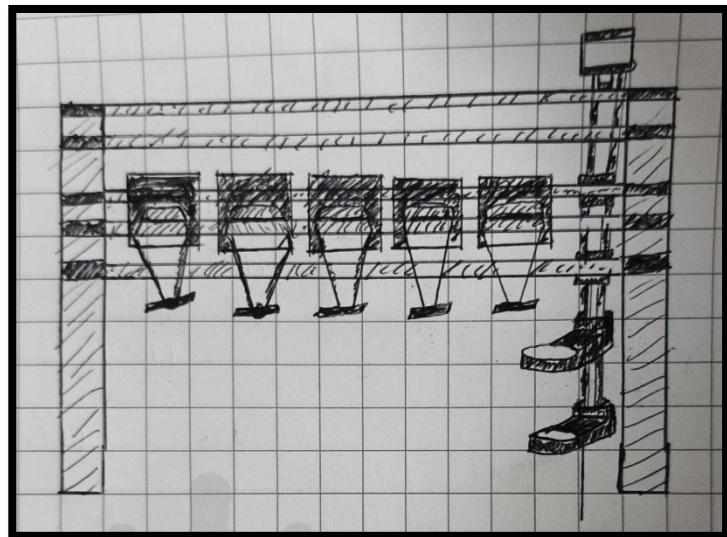
7.5. Construcción del prototipo:

Se ensamblará una estructura mecánica y se integraran los componentes electrónicos.

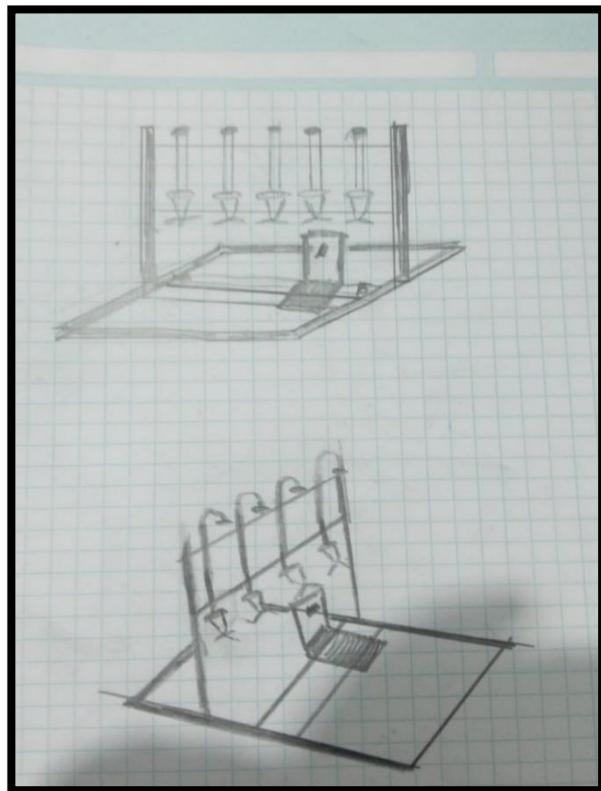
Se busca una arquitectura planeada para el dispensador de cocteles.



Bosquejo general de proyecto; Elaboración propia

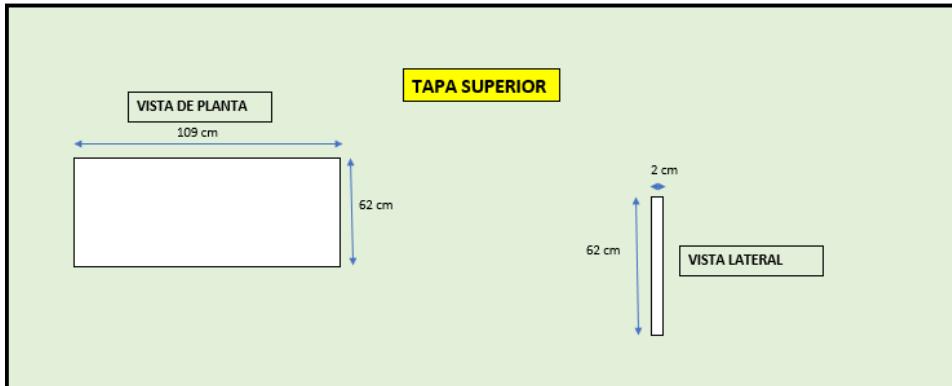


Bosquejo general de proyecto; Elaboración propia

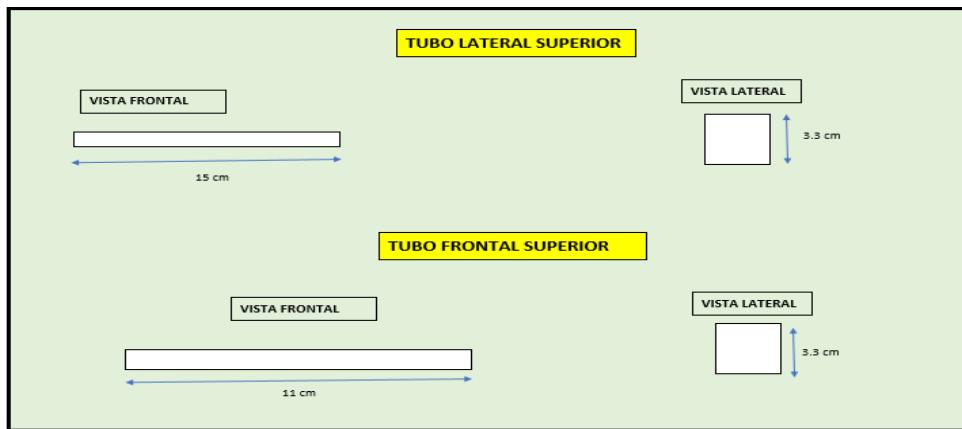


Bosquejo general de proyecto; Elaboración propia

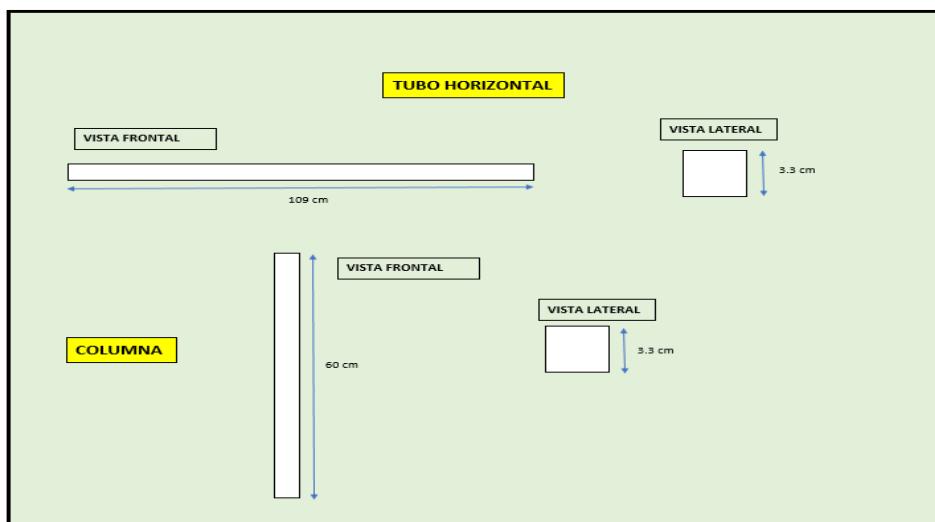
- Después de acordar conjuntamente el diseño de la estructura, realizamos un plano mecánico para la estructura



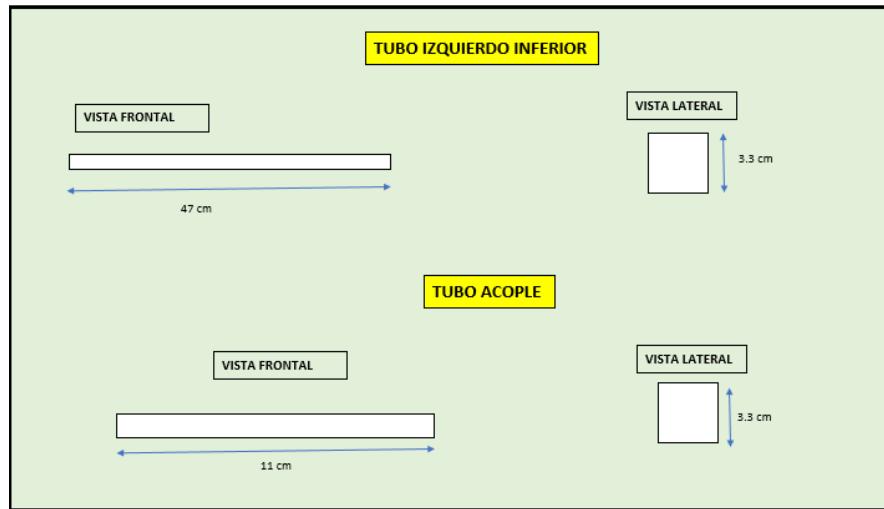
Plano mecánico; Elaboración propia



Plano mecánico; Elaboración propia

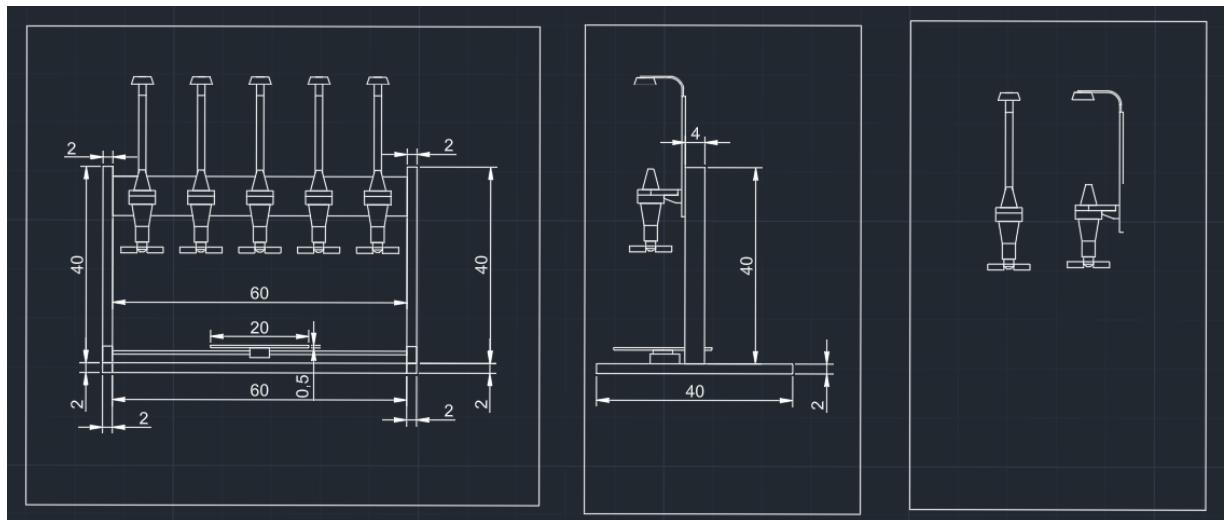


Plano mecánico; Elaboración propia



Plano mecánico; Elaboración propia

- Diseñamos en 3D con el programa AutoCAD de nuestro BARBOT, con acotaciones de medida y como quedaría implementado según lo elaborado en el plano y con los perfiles de aluminio y los dosificadores de bebidas.



Plano en AutoCAD de barbot

7.6.Etapas del proceso:

El prototipo del dispensador automatizado de cócteles (BarBot) opera en cuatro etapas funcionales, que estructuran el proceso completo de preparación de bebidas desde la interfaz hasta la dosificación. Estas etapas se describen a continuación:

1. Selección de coctel

El usuario accede a la interfaz web alojada en el servidor del ESP32. Desde allí puede visualizar una lista organizada de recetas disponibles, clasificadas por tipo (con alcohol / sin alcohol). Cada receta está definida en un archivo JSON cargado en la tarjeta SD del sistema.

2. Confirmación de orden

Una vez seleccionada la receta, el usuario confirma el pedido. El sistema verifica que el BarBot esté en estado IDLE (disponible) y luego cambia su estado a PREPARING. Se inicia el procesamiento de la receta seleccionada, con la lectura de sus ingredientes y cantidades desde el JSON.

3. Preparación automática

El sistema realiza una secuencia automatizada:

- Mueve la base giratoria al dispensador correspondiente.
- Dosifica las cantidades de cada ingrediente utilizando los 2 motores paso a paso.
- Entre cada ingrediente se realiza una espera breve para estabilidad del vaso.

- Al terminar todos los ingredientes, el sistema espera unos segundos y vuelve automáticamente al dispensador inicial para quedar listo para el siguiente pedido.

4. Finalización del proceso

Una vez completada la receta, el sistema cambia a estado IDLE, desactiva los motores para ahorrar energía y libera la interfaz para una nueva selección. El estado actual y los mensajes del sistema se actualizan automáticamente en la interfaz web en tiempo real.

7.7.Tipo, nivel, enfoque y diseño de la investigación

La presente investigación es de tipo aplicada, ya que tiene como finalidad desarrollar una solución tecnológica que responda a una necesidad práctica: automatizar la preparación de cócteles mediante un dispensador inteligente. En cuanto al nivel de investigación, este se clasifica como proyectivo, dado que parte del análisis de una problemática real para proponer, diseñar y construir un prototipo funcional.

El enfoque de la investigación es cuantitativo, ya que se emplean datos medibles como el volumen dispensado, los tiempos de ejecución y la precisión del sistema para evaluar su funcionamiento. Por último, el diseño de investigación es experimental, dado que se manipulan variables controladas y se realizan pruebas con el prototipo para observar su comportamiento y validar su eficacia.

VIII. Diseño electrónico para la adquisición de señales

El sistema BarBot está diseñado para automatizar la preparación de cócteles mediante el control electrónico de bombas peristálticas que dosifican cada ingrediente. Para lograr un funcionamiento autónomo, se requiere un diseño electrónico que permita adquirir señales que indiquen el estado del sistema y garanticen una operación segura.

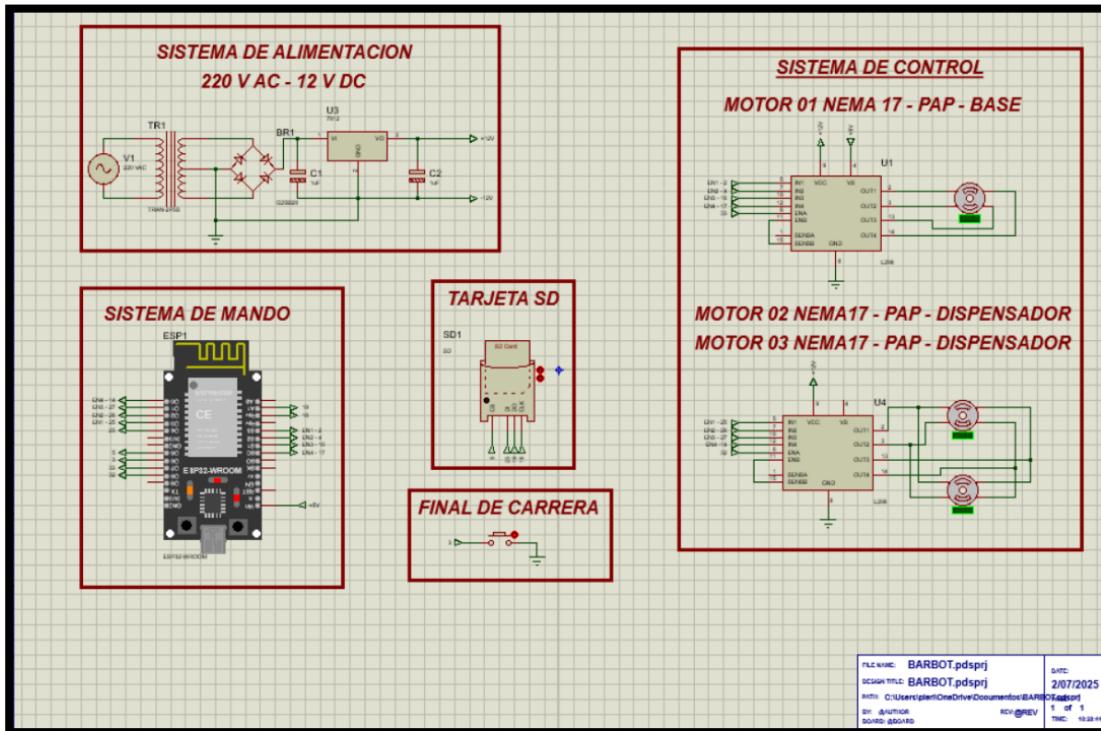


Figura 1: Esquema electrónico del sistema BarBot; Elaboración propia.

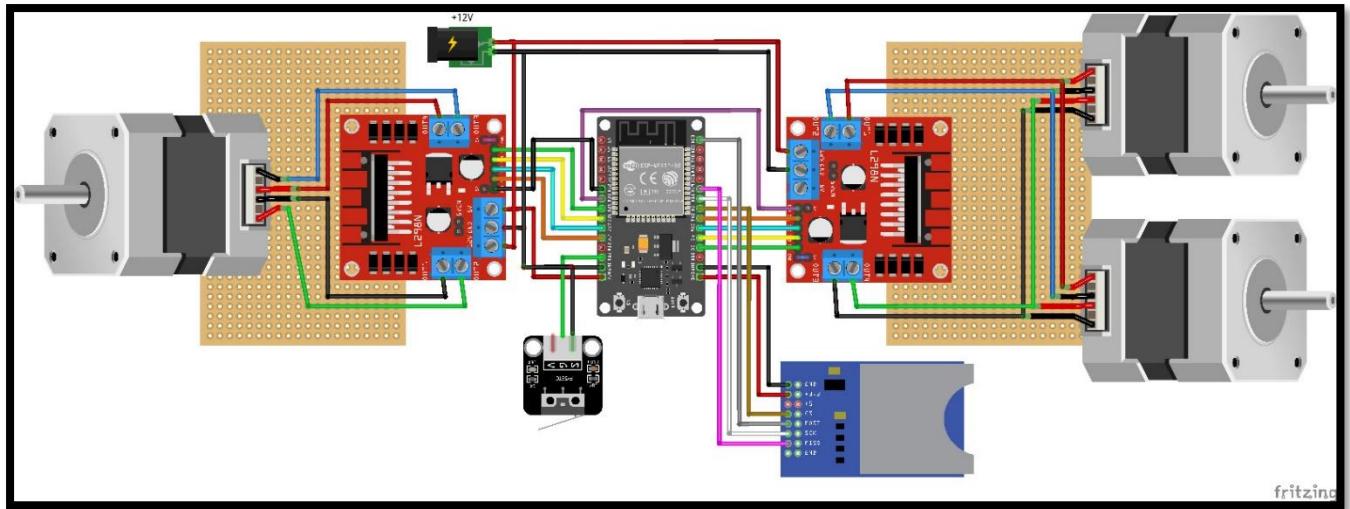


Figura 1: Esquema eléctrico del sistema BarBot; Elaboración propia.

Este sistema electrónico permite que el BarBot funcione de forma eficiente y autónoma, con seguridad en el proceso de dispensado y facilidad de uso para el usuario final.

8.1. Definición de código:

- Tipo de Comunicación Utilizada

El sistema emplea comunicación Wifi mediante el módulo integrado del ESP32. Gracias a esto, el dispositivo crea un servidor web (10.79.227.94) al que se puede acceder desde cualquier navegador conectado a la misma red. A través de este servidor, el usuario puede:

- ✓ Seleccionar recetas.
- ✓ Enviar órdenes de preparación.

- Función Principal del Código

El programa convierte al ESP32 en el cerebro de un robot bartender automático (BarBot) que puede:

- ✓ Leer recetas almacenadas en una tarjeta SD.
- ✓ Mover horizontalmente la plataforma de la base hacia distintos dispensadores de líquidos del 1 al 5.
- ✓ Activa los motores verticales para verter la cantidad adecuada de cada ingrediente.

- Descripción de Funciones y Componentes

1. Conexión Wifi

El código se conecta a una red Wifi usando los datos definidos (ssid, password) y abre un servidor HTTP.

2. Interfaz Web

El servidor responde con una página HTML moderna, donde el usuario puede:

- ✓ Ver categorías de bebidas.
- ✓ Seleccionar recetas.

3. Motores Paso a Paso (L298N + Stepper)

- ✓ Motor de base: rota el vaso hacia cada dispensador según la receta.

- ✓ Motor dispensador: baja y sube para servir una cantidad precisa de onzas.

Ambos motores son controlados usando la biblioteca Stepper y los controladores L298N.

4. Tarjeta SD y Recetas

- ✓ El archivo recipes.json contiene todas las bebidas disponibles, con la cantidad de onzas y el número de cada dispensador.
- ✓ Este archivo se carga en la memoria del ESP32 al inicio del programa usando la librería ArduinoJson.

5. Endstop (Sensor de final de carrera)

Permite calibrar la posición inicial de la plataforma giratoria automáticamente cada vez que se enciende el sistema.

8.2. Selección de cocteles

Se realiza un estudio de mercado para la utilización de las bebidas y que cocteles poder preparar, después de discutir e investigar decidimos usar coca cola, ron, vodka, agua tónica y jugo de naranja.

Después de tener las bebidas seleccionadas elaboramos una lista de 17 cocteles en las cuales 15 llevarán alcohol y 2 serán sin alcohol.

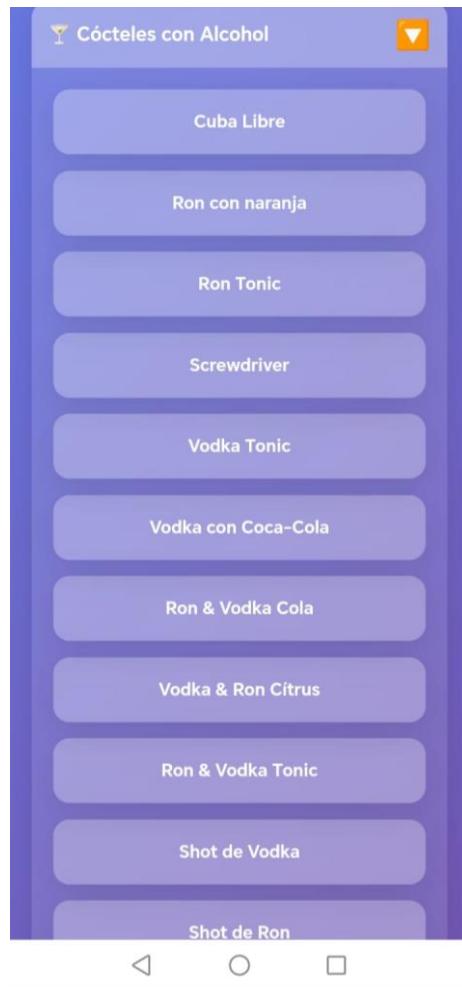
Numero de coctel	Nombre de coctel	Ingredientes (unidades de 1.5 oz)
1	Cuba libre	Ron:1, coca-cola:2
2	Ron con naranja	Ron: 1, jugo de naranja: 2
3	Ron tonic	Ron: 1, agua tónica: 2
4	Screwdriver	Vodka: 1, jugo de naranja: 2
5	Vodka tonic	Vodka: 1, agua tónica: 2
6	Vodka con Coca-Cola	Vodka: 1, Coca-Cola: 2
7	Ron & vodka cola	Ron: 1, vodka: 1, Coca-Cola: 2
8	Vodka & ron citrus	Ron: 1, vodka; 1, jugo de naranja: 2
9	Ron & vodka tonic	Ron: 1, vodka: 1, agua tónica: 2
10	Shot de vodka	Vodka: 1
11	Shot de ron	Ron: 1
12	Shot 50/50	Vodka: 1, ron: 1
13	Shot cítrico	Vodka: 1, jugo de naranja: 1
14	Naranja tónica	Jugo de naranja: 1, agua tónica: 1
15	Cítrico cola	Jugo de naranja: 1, Coca-Cola: 1
16	Vodka sunrise fake	Vodka: 1, Jugo de naranja: 1, Coca-Cola: 1
17	Ron fresh	Ron: 1, jugo de naranja: 1, agua tónica: 1

8.3.Definición de interfaz web

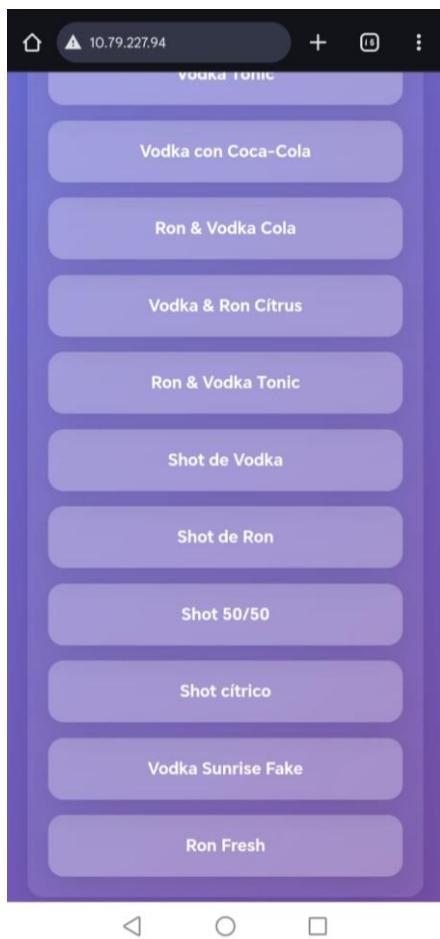
En el código se desarrolla un interfaz web para la selección del coctel, este servirá como interfaz de comunicación hacia el esp32



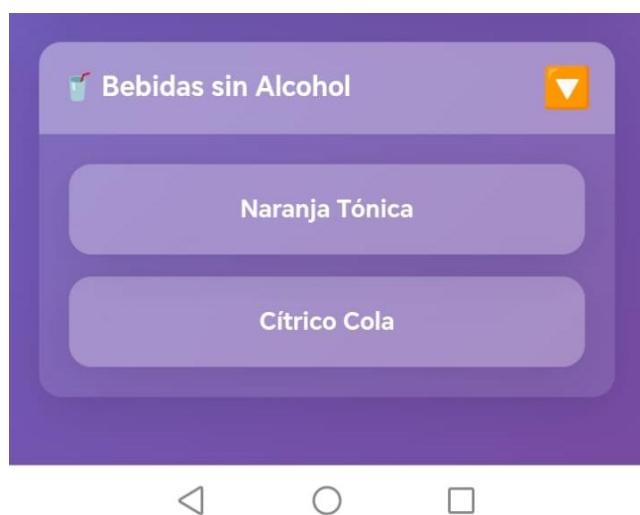
Menú de interfaz web del barbot para la selección de cocteles con o sin alcohol



Menú de interfaz web del barbot para la selección de cocteles con alcohol



Continuación del menú de interfaz web del barbot para la selección de cocteles con alcohol



Menú de interfaz web del barbot para la selección de cocteles sin alcohol

8.4. Código

8.4.1. Código ESP32

```
#include "FS.h"
#include "SD.h"
#include "SPI.h"
#include <ArduinoJson.h>
#include <Stepper.h>
#include <WiFi.h>
#include <WebServer.h>
// ===== CONFIG WIFI =====
const char* ssid = "m";
const char* password = "qwerty123";
// ===== CONFIG SD =====
#define SD_CS_PIN 5
// ===== ENDSTOP ===== // << NUEVO
#define ENDSTOP_PIN 13
// ===== PASOS & MOTORES =====
const int stepsPerRevolution = 200;
const int totalVueltas      = 45;
const int totalDispensadores = 5;
const long pasosPorDisp = (stepsPerRevolution * totalVueltas) /
(totalDispensadores - 1);
long currentSteps = 0;
// Base (L298N #1)
const int basePins[4]    = {25, 26, 27, 14};
const int baseEnablePin = 32;
Stepper baseStepper(stepsPerRevolution, basePins[0], basePins[1],
basePins[2], basePins[3]);
// Dispensador (L298N #2)
const int servePins[4]   = {2, 4, 16, 17};
const int serveEnablePin = 33;
Stepper serveStepper(stepsPerRevolution, servePins[0], servePins[1],
servePins[2], servePins[3]);
// ===== JSON =====
StaticJsonDocument<2048> doc;
// ===== WEB SERVER =====
WebServer server(80);
// ===== ESTADO DEL BARBOT =====
enum BarbotState {
    IDLE,
    PREPARING,
    MOVING,
    SERVING,
```

```

    WAITING,
    HOMING // << NUEVO
};

BarbotState currentState = IDLE;
String currentRecipe = "";
String statusMessage = "Listo para preparar bebidas";
int currentDispenser = 1;
// Función para desactivar completamente los motores
void disableAllMotors() {
    // Desactivar enables
    digitalWrite(baseEnablePin, LOW);
    digitalWrite(servoEnablePin, LOW);
    // Desactivar todas las bobinas del motor base
    digitalWrite(basePins[0], LOW);
    digitalWrite(basePins[1], LOW);
    digitalWrite(basePins[2], LOW);
    digitalWrite(basePins[3], LOW);
    // Desactivar todas las bobinas del motor dispensador
    digitalWrite(servoPins[0], LOW);
    digitalWrite(servoPins[1], LOW);
    digitalWrite(servoPins[2], LOW);
    digitalWrite(servoPins[3], LOW);
}
// Lee todo el archivo de SD y lo devuelve como String
String readFileToString(fs::FS &fs, const char * path) {
    File file = fs.open(path, FILE_READ);
    if (!file) {
        Serial.printf("Error abriendo %s\n", path);
        return "";
    }
    String s;
    while (file.available()) s += char(file.read());
    file.close();
    return s;
}
// ===== NUEVA FUNCIÓN DE HOMING ===== // << NUEVO
void homeBase() {
    currentState = HOMING;
    statusMessage = "Calibrando posicion...";
    Serial.println("⚡ Iniciando secuencia de Homing para la
base...");
    // Si el endstop ya está presionado, estamos en casa.
    if (digitalRead(ENDSTOP_PIN) == LOW) {
        Serial.println("⚠ Endstop ya presionado. Barbot en posicion
inicial.");
}

```

```
    currentSteps = 0;
    currentDispenser = 1;
    currentState = IDLE;
    statusMessage = "Listo para preparar bebidas";
    return;
}
// Si no, mover hacia la izquierda hasta presionarlo
Serial.println("← Moviendo a la posicion inicial hasta presionar el endstop...");
// Activar motor base para el movimiento
digitalWrite(baseEnablePin, HIGH);
delay(10);
// Mover paso a paso hacia la izquierda (dirección negativa)
// hasta que el endstop se presione (lea LOW)
while (digitalRead(ENDSTOP_PIN) == HIGH) {
    baseStepper.step(-1); // Mueve un paso en dirección negativa
}
// Desactivar motor base
digitalWrite(baseEnablePin, LOW);
// Desactivar bobinas
digitalWrite(basePins[0], LOW);
digitalWrite(basePins[1], LOW);
digitalWrite(basePins[2], LOW);
digitalWrite(basePins[3], LOW);
Serial.println("☑ ¡Posicion inicial encontrada!");
// Calibrar la posición actual como el punto cero
currentSteps = 0;
currentDispenser = 1;
currentState = IDLE;
statusMessage = "Listo para preparar bebidas";
}
// ===== FUNCIONES WEB =====
void handleRoot() {
    String html = R"rawliteral(
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>BarBot Control</title>
    <style>
        * {
            margin: 0;
            padding: 0;

```

```
        box-sizing: border-box;
    }
body {
    font-family: 'Arial', sans-serif;
    background: linear-gradient(135deg, #667eea 0%, #764ba2
100%);
    min-height: 100vh;
    color: white;
    overflow-x: hidden;
}
.container {
    max-width: 500px;
    margin: 0 auto;
    padding: 20px;
}
.header {
    text-align: center;
    margin-bottom: 30px;
    padding: 20px;
    background: rgba(255, 255, 255, 0.1);
    border-radius: 15px;
    backdrop-filter: blur(10px);
    box-shadow: 0 8px 32px rgba(0, 0, 0, 0.1);
}
.header h1 {
    font-size: 2.5em;
    margin-bottom: 10px;
    text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.3);
}

.status-panel {
    background: rgba(255, 255, 255, 0.15);
    border-radius: 15px;
    padding: 20px;
    margin-bottom: 20px;
    text-align: center;
    backdrop-filter: blur(10px);
    box-shadow: 0 8px 32px rgba(0, 0, 0, 0.1);
}
.status-indicator {
    width: 20px;
    height: 20px;
    border-radius: 50%;
    display: inline-block;
    margin-right: 10px;
```

```
        animation: pulse 2s infinite;
    }
.status-idle { background-color: #4CAF50; }
.status-preparing { background-color: #FF9800; }
.status-moving { background-color: #2196F3; }
.status-serving { background-color: #9C27B0; }
.status-waiting { background-color: #FFC107; }
.status-homing { background-color: #00BCD4; }
@keyframes pulse {
    0% { box-shadow: 0 0 0 0 rgba(255, 255, 255, 0.7); }
    70% { box-shadow: 0 0 0 10px rgba(255, 255, 255, 0); }
    100% { box-shadow: 0 0 0 0 rgba(255, 255, 255, 0); }
}
.category-section {
    margin-bottom: 25px;
    background: rgba(255, 255, 255, 0.1);
    border-radius: 15px;
    overflow: hidden;
    backdrop-filter: blur(10px);
    box-shadow: 0 8px 32px rgba(0, 0, 0, 0.1);
}
.category-header {
    background: rgba(255, 255, 255, 0.2);
    padding: 20px;
    cursor: pointer;
    display: flex;
    justify-content: space-between;
    align-items: center;
    font-size: 1.2em;
    font-weight: bold;
    transition: all 0.3s ease;
}
.category-header:hover {
    background: rgba(255, 255, 255, 0.3);
}
.category-icon {
    font-size: 1.5em;
    transition: transform 0.3s ease;
}
.category-icon.collapsed {
    transform: rotate(-90deg);
}
.recipes-grid {
    display: grid;
```

```
        grid-template-columns: repeat(auto-fit, minmax(200px,
1fr));
        gap: 15px;
        padding: 20px;
        max-height: 0;
        overflow: hidden;
        transition: max-height 0.3s ease;
    }
.recipes-grid.expanded {
    max-height: 2000px;
}
.recipe-btn {
    background: rgba(255, 255, 255, 0.2);
    border: none;
    border-radius: 15px;
    padding: 20px;
    color: white;
    font-size: 1.1em;
    font-weight: bold;
    cursor: pointer;
    transition: all 0.3s ease;
    backdrop-filter: blur(10px);
    box-shadow: 0 8px 32px rgba(0, 0, 0, 0.1);
    min-height: 80px;
    display: flex;
    align-items: center;
    justify-content: center;
}
.recipe-btn:hover {
    background: rgba(255, 255, 255, 0.3);
    transform: translateY(-2px);
    box-shadow: 0 12px 40px rgba(0, 0, 0, 0.2);
}
.recipe-btn:active {
    transform: translateY(0);
}
.recipe-btn:disabled {
    opacity: 0.5;
    cursor: not-allowed;
    transform: none;
}
.current-position {
    text-align: center;
    margin-top: 20px;
    font-size: 1.2em;
```

```
        background: rgba(255, 255, 255, 0.1);
        padding: 15px;
        border-radius: 10px;
        backdrop-filter: blur(10px);
    }
    .loading {
        display: none;
        text-align: center;
        margin: 20px 0;
    }
    .spinner {
        border: 4px solid rgba(255, 255, 255, 0.3);
        border-top: 4px solid white;
        border-radius: 50%;
        width: 40px;
        height: 40px;
        animation: spin 1s linear infinite;
        margin: 0 auto 10px;
    }
    @keyframes spin {
        0% { transform: rotate(0deg); }
        100% { transform: rotate(360deg); }
    }
    @media (max-width: 480px) {
        .recipes-grid {
            grid-template-columns: 1fr;
        }
        .recipe-btn {
            min-height: 60px;
            font-size: 1em;
        }
        .category-header {
            font-size: 1.1em;
            padding: 15px;
        }
    }

```

</style>

```
</head>
<body>
    <div class="container">
        <div class="header">
            <h1>🤖 BarBot</h1>
            <p>Tu bartender automático</p>
        </div>
        <div class="status-panel">
```

```
        <div class="status-indicator status-idle"
id="statusIndicator"></div>
            <span id="statusText">Listo para preparar bebidas</span>
        </div>
        <div class="loading" id="loading">
            <div class="spinner"></div>
            <p>Preparando tu bebida...</p>
        </div>
        <div class="category-section">
            <div class="category-header">
                <span>칵테les con Alcohol</span>
                <span class="category-icon"
id="alcoholIcon">▼</span>
            </div>
            <div class="recipes-grid expanded" id="alcoholGrid">
            </div>
        </div>
        <div class="category-section">
            <div class="category-header">
                <span>칵테les sin Alcohol</span>
                <span class="category-icon"
id="nonalcoholIcon">▼</span>
            </div>
            <div class="recipes-grid expanded" id="nonalcoholGrid">
            </div>
        </div>
    </div>
    <script>
        let isPreparingDrink = false;

        // Función para actualizar el estado
        function updateStatus() {
            fetch('/status')
                .then(response => response.json())
                .then(data => {
                    document.getElementById('statusText').textContent = data.message;
                    const indicator =
document.getElementById('statusIndicator');
                    const loading =
document.getElementById('loading');
                    const recipesGrid =
document.getElementById('recipesGrid');

```

```
// Actualizar indicador de estado
indicator.className = 'status-indicator status-' + data.state.toLowerCase();
// Mostrar/ocultar loading
if (data.state === 'PREPARING' || data.state === 'MOVING' || data.state === 'SERVING' || data.state === 'WAITING' || data.state === 'HOMING') {
    loading.style.display = 'block';
    recipesGrid.style.opacity = '0.5';
    isPreparingDrink = true;
    disableAllButtons();
} else {
    loading.style.display = 'none';
    recipesGrid.style.opacity = '1';
    isPreparingDrink = false;
    enableAllButtons();
}
})
.catch(error => console.error('Error:', error));
}
// Función para preparar una receta
function prepareRecipe(recipeName) {
    if (isPreparingDrink) return;
    fetch('/prepare', {
        method: 'POST',
        headers: {
            'Content-Type': 'application/x-www-form-urlencoded',
        },
        body: 'recipe=' + encodeURIComponent(recipeName)
    })
    .then(response => response.text())
    .then(data => {
        console.log('Respuesta:', data);
        updateStatus();
    })
    .catch(error => {
        console.error('Error:', error);
        alert('Error al preparar la bebida');
    });
}
// Función para deshabilitar todos los botones
function disableAllButtons() {
    const buttons = document.querySelectorAll('.recipe-btn');
```

```
        buttons.forEach(btn => btn.disabled = true);
    }
    // Función para habilitar todos los botones
    function enableAllButtons() {
        const buttons = document.querySelectorAll('.recipe-
btn');
        buttons.forEach(btn => btn.disabled = false);
    }
    // Función para toggle de categorías
    function toggleCategory(category) {
        const grid = document.getElementById(category + 'Grid');
        const icon = document.getElementById(category + 'Icon');

        if (grid.classList.contains('expanded')) {
            grid.classList.remove('expanded');
            icon.classList.add('collapsed');
            icon.textContent = '▶';
        } else {
            grid.classList.add('expanded');
            icon.classList.remove('collapsed');
            icon.textContent = '▼';
        }
    }
    // Función para formatear nombres de recetas
    function formatRecipeName(name) {
        const recipeNames = {
            'CubaLibre': 'Cuba Libre',
            'RonConNaranja': 'Ron con naranja',
            'RonTonic': 'Ron Tonic',
            'Screwdriver': 'Screwdriver',
            'VodkaTonic': 'Vodka Tonic',
            'VodkaCola': 'Vodka con Coca-Cola',
            'RonVodkaCola': 'Ron & Vodka Cola',
            'RonVodkaNaranja': 'Vodka & Ron Cítrus',
            'RonVodkaTonic': 'Ron & Vodka Tonic',
            'ShotVodka': 'Shot de Vodka',
            'ShotRon': 'Shot de Ron',
            'Shot5050': 'Shot 50/50',
            'ShotCitrico': 'Shot cítrico',
            'NaranjaTonic': 'Naranja Tónica',
            'CitricoCola': 'Cítrico Cola',
            'VodkaSunriseFake': 'Vodka Sunrise Fake',
            'RonFresh': 'Ron Fresh'
        };
        return recipeNames[name] || name;
    }
}
```

```
        }
        // Función para categorizar recetas
        function categorizeRecipes(recipes) {
            const alcoholic = {};
            const nonAlcoholic = {};
            // Recetas con alcohol (contienen dispensador 2 (vodka)
            o 4 (ron))
                const alcoholicRecipes = [
                    'CubaLibre', 'RonConNaranja', 'RonTonic',
                    'Screwdriver', 'VodkaTonic',
                    'VodkaCola', 'RonVodkaCola', 'RonVodkaNaranja',
                    'RonVodkaTonic',
                    'ShotVodka', 'ShotRon', 'Shot5050', 'ShotCitrino',
                    'VodkaSunriseFake', 'RonFresh'
                ];
                // Recetas sin alcohol
                const nonAlcoholicRecipes = ['NaranjaTonic',
                    'CitrinoCola'];
                for (const [name, ingredients] of
Object.entries(recipes)) {
                    if (alcoholicRecipes.includes(name)) {
                        alcoholic[name] = ingredients;
                    } else if (nonAlcoholicRecipes.includes(name)) {
                        nonAlcoholic[name] = ingredients;
                    }
                }
                return { alcoholic, nonAlcoholic };
            }
            // Cargar recetas al inicio
            function loadRecipes() {
                fetch('/recipes')
                    .then(response => response.json())
                    .then(recipes => {
                        const { alcoholic, nonAlcoholic } =
categorizeRecipes(recipes);
                        // Cargar recetas con alcohol
                        const alcoholGrid =
document.getElementById('alcoholGrid');
                        alcoholGrid.innerHTML = '';
                        for (const [name, ingredients] of
Object.entries(alcoholic)) {
                            const button =
document.createElement('button');
                            button.className = 'recipe-btn';
                            button.textContent = formatRecipeName(name);
                        }
                    })
            }
        
```

```

        button.onclick = () => prepareRecipe(name);
        alcoholGrid.appendChild(button);
    }
    // Cargar bebidas sin alcohol
    const nonAlcoholGrid =
document.getElementById('nonalcoholGrid');
    nonAlcoholGrid.innerHTML =
 '';
        for (const [name, ingredients] of
Object.entries(nonAlcoholic)) {
            const button =
document.createElement('button');
            button.className = 'recipe-btn';
            button.textContent = formatRecipeName(name);
            button.onclick = () => prepareRecipe(name);
            nonAlcoholGrid.appendChild(button);
        }
    })
    .catch(error => console.error('Error cargando
recetas:', error));
}
// Actualizar estado cada 2 segundos
setInterval(updateStatus, 2000);
// Cargar todo al inicio
loadRecipes();
updateStatus();
</script>
</body>
</html>
)rawliteral";
server.send(200, "text/html", html);
}
void handleStatus() {
String stateStr;
switch (currentState) {
    case IDLE: stateStr = "IDLE"; break;
    case PREPARING: stateStr = "PREPARING"; break;
    case MOVING: stateStr = "MOVING"; break;
    case SERVING: stateStr = "SERVING"; break;
    case WAITING: stateStr = "WAITING"; break;
    case HOMING: stateStr = "HOMING"; break; // << NUEVO
}
String json = "{";
json += "\"state\":\"" + stateStr + "\",";
json += "\"message\":\"" + statusMessage + "\",";

```

```

        json += "\"dispenser\":\"" + String(currentDispenser) + ",";
        json += "\"recipe\":\"" + currentRecipe + "\"";
        json += "}";
        server.send(200, "application/json", json);
    }
    void handleRecipes() {
        String json = readFileToString(SD, "/recipes.json");
        server.send(200, "application/json", json);
    }
    void handlePrepare() {
        if (currentState != IDLE) {
            server.send(400, "text/plain", "Barbot ocupado");
            return;
        }
        String recipeName = server.arg("recipe");
        if (recipeName.isEmpty()) {
            server.send(400, "text/plain", "Nombre de receta requerido");
            return;
        }
        // Buscar receta en el JSON
        JsonObject root = doc.as<JsonObject>();
        if (!root.containsKey(recipeName)) {
            server.send(404, "text/plain", "Receta no encontrada");
            return;
        }
        server.send(200, "text/plain", "Preparando " + recipeName);
        // Preparar la receta (esto se ejecutará después de responder)
        prepareRecipeWeb(recipeName);
    }
    void prepareRecipeWeb(String recipeName) {
        currentState = PREPARING;
        currentRecipe = recipeName;
        statusMessage = "Preparando " + recipeName + "...";
        Serial.printf("→ [WEB] Preparando %s ...\n", recipeName.c_str());

        JsonObject root = doc.as<JsonObject>();
        JSONArray ings = root[recipeName].as<JSONArray>();
        for (JsonObject ing : ings) {
            int disp = ing["disp"].as<int>();
            int oz   = ing["oz"].as<int>();
            Serial.printf("  • Dispensador %d: %d oz\n", disp, oz);
            statusMessage = "Moviendo a dispensador " + String(disp);
            gotoDispenser(disp);
            delay(500);
            statusMessage = "Sirviendo " + String(oz) + " oz";
        }
    }
}

```

```

    serve0z(oz);
    delay(500);
}
// Esperar 7 segundos
currentState = WAITING;
statusMessage = "Esperando 7 segundos...";
Serial.println("☒ Esperando 7 segundos...");
delay(7000);
// Regresar al dispensador 1
statusMessage = "Regresando al dispensador 1";
Serial.println("☒ Regresando al dispensador 1...");
gotoDispenser(1);
// Desactivar motores y resetear estado
disableAllMotors();
currentState = IDLE;
currentRecipe = "";
statusMessage = "Listo para preparar bebidas";
Serial.println("☒ ¡Listo! Barbot en posición inicial.\n");
}

void setup() {
  Serial.begin(115200);
  delay(1000);
  Serial.println("\n==== Barbot Inicializando ====");
  // Configurar todos los pines como salida primero
  pinMode(baseEnablePin, OUTPUT);
  pinMode(serveEnablePin, OUTPUT);
  for (int i = 0; i < 4; i++) {
    pinMode(basePins[i], OUTPUT);
    pinMode(servePins[i], OUTPUT);
  }
  // Configurar pin del endstop // << NUEVO
  pinMode(ENDSTOP_PIN, INPUT_PULLUP);
  // Desactivar completamente todos los motores al inicio
  disableAllMotors();
  // Montar SD
  if (!SD.begin(SD_CS_PIN)) {
    Serial.println("SD mount failed!");
    statusMessage = "Error: No se pudo montar SD";
    while (1);
  }
  // Leer y parsear JSON
  String json = readFileToString(SD, "/recipes.json");
  auto err = deserializeJson(doc, json);
  if (err) {
    Serial.print("JSON parse failed: ");

```

```
    Serial.println(err.c_str());
    statusMessage = "Error: JSON inválido";
    while (1);
}
Serial.println("Recetas cargadas:");
JsonObject root = doc.as<JsonObject>();
for (auto kv : root) {
    Serial.print(" • ");
    Serial.println(kv.key().c_str());
}
// Configurar velocidades de steppers
baseStepper.setSpeed(190);
serveStepper.setSpeed(35);
// ===== INICIAR SECUENCIA DE HOMING ===== // << MODIFICADO
homeBase();
// Conectar a WiFi
Serial.println("\nConectando a WiFi...");
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.print(".");
}
Serial.println();
Serial.println("WiFi conectado!");
Serial.print("IP address: ");
Serial.println(WiFi.localIP());
// Configurar servidor web
server.on("/", handleRoot);
server.on("/status", handleStatus);
server.on("/recipes", handleRecipes);
server.on("/prepare", HTTP_POST, handlePrepare);
server.begin();
Serial.println("Servidor web iniciado");

Serial.println("\n== BARBOT LISTO ==");
Serial.println("Controles disponibles:");
Serial.println("• Web: http://" + WiFi.localIP().toString());
Serial.println("• Serial: Teclea el nombre de la receta y presiona
ENTER");
Serial.println();
statusMessage = "Listo para preparar bebidas";
}
void loop() {
// Manejar servidor web
server.handleClient();
```

```

// Manejar comandos Serial (solo si no está ocupado)
if (Serial.available() && currentState == IDLE) {
    String line = Serial.readStringUntil('\n');
    line.trim();
    if (line.isEmpty()) return;
    // Buscar receta en el JSON
    JsonObject root = doc.as<JsonObject>();
    if (!root.containsKey(line)) {
        Serial.println("X Receta no encontrada.");
        return;
    }
    prepareRecipeSerial(line);
}
}

void prepareRecipeSerial(String recipeName) {
    currentState = PREPARING;
    currentRecipe = recipeName;
    statusMessage = "Preparando " + recipeName + "...";
    Serial.printf("→ [SERIAL] Preparando %s ...\n", recipeName.c_str());
    JsonObject root = doc.as<JsonObject>();
    JsonArray ings = root[recipeName].as<JsonArray>();
    for (JsonObject ing : ings) {
        int disp = ing["disp"].as<int>();
        int oz   = ing["oz"].as<int>();
        Serial.printf("  • Dispensador %d: %d oz\n", disp, oz);
        statusMessage = "Moviendo a dispensador " + String(disp);
        gotoDispenser(disp);
        delay(500);
        statusMessage = "Sirviendo " + String(oz) + " oz";
        serveOz(oz);
        delay(500);
    }
    // Esperar 7 segundos
    currentState = WAITING;
    statusMessage = "Esperando 7 segundos...";
    Serial.println("☒ Esperando 7 segundos...");
    delay(7000);
    // Regresar al dispensador 1
    statusMessage = "Regresando al dispensador 1";
    Serial.println("☒ Regresando al dispensador 1...");
    gotoDispenser(1);
    // Desactivar motores y resetear estado
    disableAllMotors();
    currentState = IDLE;
    currentRecipe = "";
}

```

```

statusMessage = "Listo para preparar bebidas";
Serial.println("☑ ¡Listo! Barbot en posición inicial.\n");
}

// Mueve la base al dispensador d (1-totalDispensadores)
void gotoDispenser(int d) {
    if (d < 1 || d > totalDispensadores) {
        Serial.printf("☒ Dispensador %d fuera de rango\n", d);
        return;
    }
    currentState = MOVING;
    long target = (long)(d - 1) * pasosPorDisp;
    long delta = target - currentSteps;
    if (delta == 0) {
        Serial.printf("    → Ya estás en el dispensador %d\n", d);
        currentDispenser = d;
        return;
    }
    // Activar motor base
    digitalWrite(baseEnablePin, HIGH);
    delay(10);
    // Mover de una sola vez
    baseStepper.step(delta);
    // Desactivar motor base inmediatamente
    digitalWrite(baseEnablePin, LOW);
    // Desactivar bobinas del motor base
    digitalWrite(basePins[0], LOW);
    digitalWrite(basePins[1], LOW);
    digitalWrite(basePins[2], LOW);
    digitalWrite(basePins[3], LOW);
    currentSteps = target;
    currentDispenser = d;
    Serial.printf("    → Base en dispensador %d (pasos: %ld)\n", d,
currentSteps);
}
// Sirve oz onzas con el motor del dispensador
void serveOz(int oz) {
    if (oz <= 0) {
        Serial.println("☒ Cantidad de onzas inválida");
        return;
    }
    currentState = SERVING;
    digitalWrite(serveEnablePin, HIGH);
    delay(10);
    for (int i = 0; i < oz; i++) {
        Serial.printf("        Sirviendo onza %d/%d...\n", i+1, oz);
    }
}

```

```

    // Subir
    serveStepper.step(-stepsPerRevolution * 3);
    delay(3000);
    // Bajar
    serveStepper.step(stepsPerRevolution * 3);
    delay(2000);
}
// Desactivar motor dispensador
digitalWrite(serveEnablePin, LOW);
// Desactivar bobinas del motor dispensador
digitalWrite(servePins[0], LOW);
digitalWrite(servePins[1], LOW);
digitalWrite(servePins[2], LOW);
digitalWrite(servePins[3], LOW);
Serial.println("    → Servicio completado");
}

```

8.4.2. Código en microSD

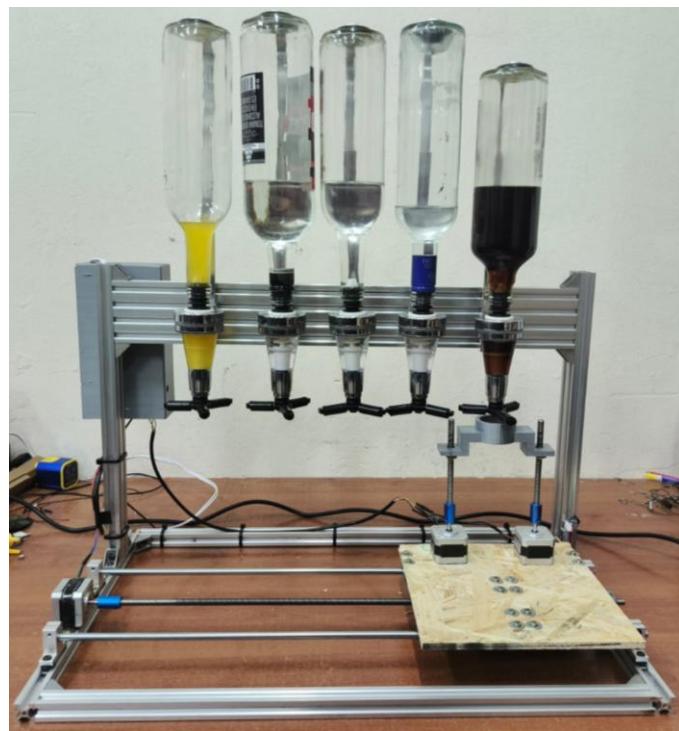
```
{
  "CubaLibre": [
    { "disp": 4, "oz": 1 },
    { "disp": 1, "oz": 2 }
  ],
  "RonConNaranja": [
    { "disp": 4, "oz": 1 },
    { "disp": 5, "oz": 2 }
  ],
  "RonTonic": [
    { "disp": 4, "oz": 1 },
    { "disp": 3, "oz": 2 }
  ],
  "Screwdriver": [
    { "disp": 2, "oz": 1 },
    { "disp": 5, "oz": 2 }
  ],
  "VodkaTonic": [
    { "disp": 2, "oz": 1 },
    { "disp": 3, "oz": 2 }
  ],
  "VodkaCola": [
    { "disp": 2, "oz": 1 },
    { "disp": 1, "oz": 2 }
  ],
}
```

```
"RonVodkaCola": [
    { "disp": 4, "oz": 1 },
    { "disp": 2, "oz": 1 },
    { "disp": 1, "oz": 2 }
],
"RonVodkaNaranja": [
    { "disp": 4, "oz": 1 },
    { "disp": 2, "oz": 1 },
    { "disp": 5, "oz": 2 }
],
"RonVodkaTonic": [
    { "disp": 4, "oz": 1 },
    { "disp": 2, "oz": 1 },
    { "disp": 3, "oz": 2 }
],
"ShotVodka": [
    { "disp": 2, "oz": 1 }
],
"ShotRon": [
    { "disp": 4, "oz": 1 }
],
"Shot5050": [
    { "disp": 2, "oz": 1 },
    { "disp": 4, "oz": 1 }
],
"ShotCitrlico": [
    { "disp": 2, "oz": 1 },
    { "disp": 5, "oz": 1 }
],
"NaranjaTonic": [
    { "disp": 5, "oz": 1 },
    { "disp": 3, "oz": 1 }
],
"CitrlicoCola": [
    { "disp": 5, "oz": 1 },
    { "disp": 1, "oz": 1 }
],
"VodkaSunriseFake": [
    { "disp": 2, "oz": 1 },
    { "disp": 5, "oz": 1 },
    { "disp": 1, "oz": 1 }
],
"RonFresh": [
    { "disp": 4, "oz": 1 },
    { "disp": 5, "oz": 1 },
```

```
        { "disp": 3, "oz": 1 }  
    ]  
}
```

8.5. Diseño final del prototipo

Después de lo acorado en los planos y armado de proyecto en 3D de AutoCAD se llegó implementar como lo esperamos con medidas y funciones correctas, se adiciona en la parte posterior del barbot el sistema de alimentación con los componentes usados, como se adjunta en imágenes.



Dispensador implementado



Sistema de control de barbot

IX. Resultados e impactos esperados

Obtenemos un resultado muy favorable de acuerdo a lo planificado, la comunicación web hacia el ESP32 es satisfactoria para mandar la orden y cumpla la función de preparar el cóctel, este proyecto es un prototipo que contará con mejoras al futuro.

Se espera que el dispensador automatizado reduzca el tiempo de preparación de bebidas, minimice el error humano, y mejore la eficiencia del servicio, que este tipo de sistema pueda ser implementado en eventos o ferias temporales, permitiendo un servicio de bebidas eficiente sin necesidad de personal capacitado en coctelería, lo cual abriría nuevas oportunidades de negocio en contextos itinerantes o de autoservicio.

X. Cronograma de actividades

XI. Presupuesto

11.1. Presupuesto de materiales y componentes

PRESUPUESTO DE MATERIALES Y COMPONENTES			
Cantidad	Nombre del producto	Precio unitario	Precio total
2	DRIVER I298n	S/14.00	S/28.00
1	Esp32	S/45.00	S/45.00
3	MOTOR PAP NEMA 17	S/40.00	S/120.00
1	FUENTE DE VOLTAJE 12V 3A	S/65.00	S/65.00
1	KIT DOSIFICADOR DE LIQUIDOS x 6 und	S/235.00	S/235.00
1	TORNILLOS Allen M5X25mm (Pack 10 und)	S/4.50	S/3.50
1	TORNILLOS Allen M5X12mm (Pack 10 und)	S/3.50	S/2.50
1	TUERCAS Allen M5x12mm (Pack 10 und)	S/2.50	S/1.50
1	TUERCAS Allen M5x25mm (Pack 10 und)	S/2.50	S/1.50
4	CABLE P/MOTOR PAP (1M x color)	S/12.00	S/48.00
2	BARILLA DE ACERO 8mm (1M)	S/18.00	S/36.00
1	MADERA TIPO MDF 25 cm x 25 cm	S/25.00	S/25.00
3	ACOPLE RIGIDO 5X8mm	S/7.00	S/21.00
1	HUSILLO DE 60 CM	S/49.40	S/49.40
2	HUSILLO DE 10 CM	S/7.60	S/15.20
4	BRAKET VERTICAL SK12	S/6.00	S/24.00
1	BRAKET VERTICAL HUSILLO	S/3.00	S/3.00
	PERFILES DE ALUMINIO	S/441.00	S/441.00
1	JUEGO DE LLAVES ALLEN	S/10.00	S/10.00
		TOTAL	S/1174.6

XII. Referencias bibliográficas

Profesional Horeca. (2020). Beermatic: dispensador de cerveza automático. Profesional Horeca. <https://www.profesionalhoreca.com>

Rodríguez, J., Alfonso, F., & Buitrago, R. (2019). Diseño e implementación de un sistema automatizado para preparación de cócteles. Universidad Piloto de Colombia.

<http://repository.unipiloto.edu.co/bitstream/handle/20.500.12277/5462/PG-17-2-17%20OK.pdf>

Gutiérrez, J., Sulmi, C., Rosario, A., Alfonso, S., & Marco, O. (2019). DISPENSADOR ECOLÓGICO DE BEBIDAS SABORIZADAS A BASE DE FRUTAS DESHIDRATADAS.

Universidad San Ignacio de Loyola -

<https://repositorio.usil.edu.pe/entities/publication/6402b3f9-d516-4c9b-a4c6-8e89adfbc634>

Profesional Horeca. (2020). Beermatic: dispensador de cerveza automático -

<https://www.profesionalhoreca.com>

“DESARROLLO DE UN DISPENSADOR DE TRAGOS AUTOMATIZADO PARA ESTANDARIZAR EL SABOR EN LA PREPARACIÓN DE CÓCTELES” -

https://laccei.org/LACCEI2021-VirtualEdition/full_papers/FP107.pdf