



UNIVERSIDAD
CATÓLICA
BOLIVIANA
COCHABAMBA

Departamento de Ciencias Exactas e Ingenierías

Proyecto Final IoT Dispensador de Agua Automatico

MALDONADO GIL GUILLERMO

MARTINEZ GUZMAN IGNACIO BORIS

Cochabamba - Bolivia

Diciembre de 2023

ÍNDICE

1. INTRODUCCIÓN.....	2
2.MARCO TEÓRICO.....	2
2.1.Sensores.-.....	2
2.2.Actuadores.-.....	2
2.3.Terminales Inteligentes.-.....	3
2.4. AWS IoT Core.-.....	3
2.5. Shadows.-.....	3
2.6. Reglas.-.....	3
2.7.MQTT.-.....	3
2.8.Alexa Skills.-.....	3
2.9 Función Lambda.-.....	3
2.10 Bomba de Agua.-.....	4
2.11 Módulo L298N (Puente H).-.....	4
2.12 Sensores Ultrasónicos.-.....	4
2.13 Celdas De Carga.-.....	5
2.14 Transmisor de celda de carga HX711.....	5
2.15 AWS IoT Analytics.....	5
3. MARCO PRÁCTICO.....	6
3.1.DISEÑO.....	6
3.1.1.Arquitectura.....	6
(3.1.1.1 Arquitectura del proyecto implementado,Fuente:Tinkercad Fuente Propia).....	6
3.1.2.Diagramas Dinámicos.....	7
(Diagrama 3.1.2.1 Funcionamiento del dispensador ; Elaboración Propia).....	7
(Diagrama 3.1.2.2 Funcionamiento del dispensador Con instrucciones de Alexa; Elaboración Propia).....	8
(Diagrama 3.1.2.3 Funcionamiento del dispensador en un diagrama de Estado; Elaboración Propia).....	9
(Diagrama 3.1.2.4 Funcionamiento de la balanza para el dispensado de agua de Estado; Elaboración Propia).....	10
(Diagrama 3.1.2.5 Funcionamiento del dispensador con la balanza implementada para el dispensado de agua de Estado; Elaboración Propia).....	11
3.2.DESARROLLO.....	12
3.3 CÓDIGO FUENTE.....	15
3.4 PRUEBAS.....	15
4. RESULTADOS Y CONCLUSIONES.....	21
5. BIBLIOGRAFÍA.....	23
6. ANEXOS.....	23ss

1. INTRODUCCIÓN

El concepto del Internet de las Cosas (IoT) empezó a transformar la forma en que interactuamos con el mundo que nos rodea. En términos sencillos, el IoT es la conexión inteligente de dispositivos y objetos a través de internet, permitiéndoles compartir datos y realizar acciones que mejoran la eficiencia, la comodidad y la toma de decisiones.

Una vez explicado el concepto básico de IoT, el núcleo de nuestro proyecto nace de un dispositivo capaz de entender y anticipar sus necesidades de hidratación, proporcionando agua de manera eficiente y sin complicaciones, donde dos sensores trabajan simultáneamente enviando constantemente datos de distancia a un tópico en AWS. Cuando estas distancias se alteran simultáneamente, desencadenan un conjunto de reglas en AWS, dando la señal para activar una bomba que dispensa agua con una cierta precisión. También se integrará la accesibilidad de Alexa Skills, permitiendo el control total del dispensador con comandos simples.

A lo largo de este documento, explicaremos en detalle la implementación de estas tecnologías, desde la conexión de sensores hasta la configuración de reglas en AWS y la integración con las capacidades de Skills de Alexa. Nuestro objetivo es no solo ofrecer un dispensador automático, sino también proporcionar una visión profunda de cómo la convergencia de estas tecnologías puede mejorar la eficiencia y la comodidad en la vida diaria.

Este proyecto representa un acercamiento al futuro, donde la inteligencia, la automatización y la accesibilidad convergen para transformar la forma en que interactuamos con el mundo que nos rodea.

2. MARCO TEÓRICO

2.1. Sensores.-

Los sensores son herramientas que detectan y responden a algún tipo de información del entorno físico. Básicamente funcionan de la siguiente manera; El sensor dependiendo del tipo de sensor que se está utilizando, detecta un cambio en el entorno y este sensor convierte este cambio en un lenguaje que una persona puede leer o entender ese cambio..

2.2. Actuadores.-

Los actuadores son un dispositivo capaz de transformar energía en una acción con la finalidad de automatizar un proceso. Existen varios tipos de actuadores ya sean hidráulicos, neumáticos o eléctricos.

2.3. Terminales Inteligentes.-

Una terminal inteligente es un dispositivo de entrada y salida de datos que tiene su propio procesador, memoria y firmware. Esto le permite realizar funciones de forma independiente de su host.

2.4. AWS IoT Core.-

AWS IoT Core es un servicio de nube que permite a los dispositivos de Internet de las cosas (IoT) conectarse a la nube y comunicarse entre sí. AWS IoT Core proporciona una plataforma segura y escalable para que los dispositivos IoT se registren, se autenticquen, se aprovisionan y se administren.

2.5. Shadows.-

Las sombras son una característica de AWS IoT Core que proporciona una copia local de los datos de un dispositivo IoT en la nube. Las sombras se utilizan para mantener los datos del dispositivo IoT sincronizados entre el dispositivo y la nube.

2.6. Reglas.-

Las reglas son una característica de AWS IoT Core que permite a los usuarios definir acciones que se tomarán cuando se cumplan determinadas condiciones. Las reglas se pueden utilizar para desencadenar eventos, enviar notificaciones o realizar otras acciones.

2.7. MQTT.-

MQTT es un protocolo de mensajería ligero que se utiliza para la comunicación entre dispositivos IoT. MQTT es un protocolo de publicación/suscripción que permite a los dispositivos enviar y recibir mensajes.

2.8. Alexa Skills.-

Las Alexa Skills son aplicaciones que se pueden usar con los dispositivos Alexa. Alexa Skills se pueden utilizar para controlar dispositivos, reproducir música, obtener información y realizar otras tareas.

2.9 Función Lambda.-

Las funciones Lambda son funciones de código sin servidor que se pueden ejecutar en respuesta a eventos. Las funciones Lambda se pueden utilizar para realizar una variedad de tareas, como procesar datos, desencadenar eventos o enviar notificaciones.

2.10 Bomba de Agua.-

Una bomba de agua es un dispositivo que se utiliza para mover agua. Las bombas de agua se utilizan en una variedad de aplicaciones, como el suministro de agua potable, el riego y la generación de energía.

2.11 Módulo L298N (Puente H).-

El módulo L298N posee dos canales de Puente H, se puede utilizar para controlar dos motores DC o un motor Paso a Paso, controlando el sentido de giro y velocidad.

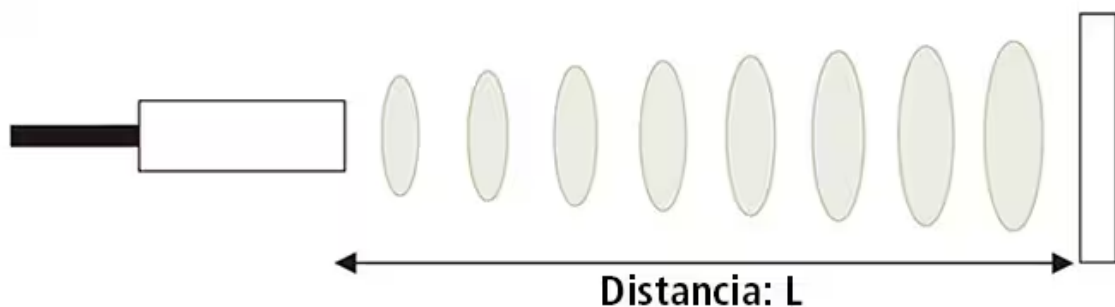
Básicamente está conformado por un driver L298N sus diodos de protección y un regulador de voltaje de 5V(78M05)

Posee un conector de 6 pines para ingresar las señales TTL para controlar los motores, una bornera de tres pines para la alimentación, y dos borneras de 2 pines para la salida a los motores.

2.12 Sensores Ultrasónicos.-

Los sensores ultrasónicos utilizan ondas sonoras para medir la distancia. Los sensores ultrasónicos se utilizan en una variedad de aplicaciones, como la detección de objetos, la medición de niveles y la medición de distancias.

Funciona de la siguiente manera:



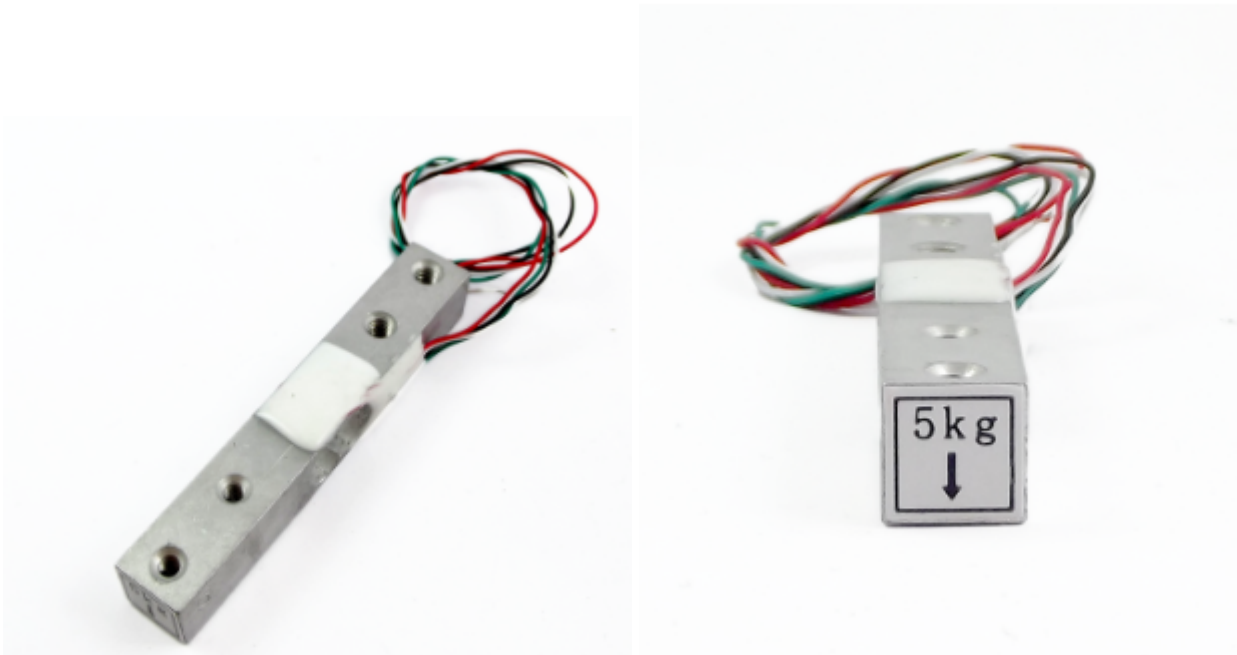
(1.1.Imagen de referencia de cómo funciona un Sensor Ultrasónico)

Este sensor cuenta con un cabezal, donde se genera una onda ultrasónica y se empieza a expandir hasta que choca con un objeto, empieza a retornar hasta este cabezal recibe la onda reflejada y lo que hace este sensor es medir la distancia al objeto contando el tiempo entre la emisión y la recepción(FIGURA 1.1).

2.13 Celdas De Carga.-

Una celda de carga es un transductor capaz de convertir una fuerza en una señal eléctrica, esto la hace a través uno o más galgas internas que posee, configuradas en un puente Wheatstone.

Existen diversos tipos de celdas de carga, en diversos modelos, el que utilizaremos para este tutorial es el que se muestra en la imagen. Usaremos una celda de carga de 5Kg que es el valor máximo que puede sensar, pero el tutorial también aplica a celdas de 20Kg y otros modelos.



2.14 Transmisor de celda de carga HX711

Este módulo es una interfaz entre las celdas de carga y el microcontrolador, permitiendo poder leer el peso de manera sencilla. Internamente se encarga de la lectura del puente wheatstone formado por la celda de carga, convirtiendo la lectura analógica a digital con su conversor A/D interno de 24 bits.

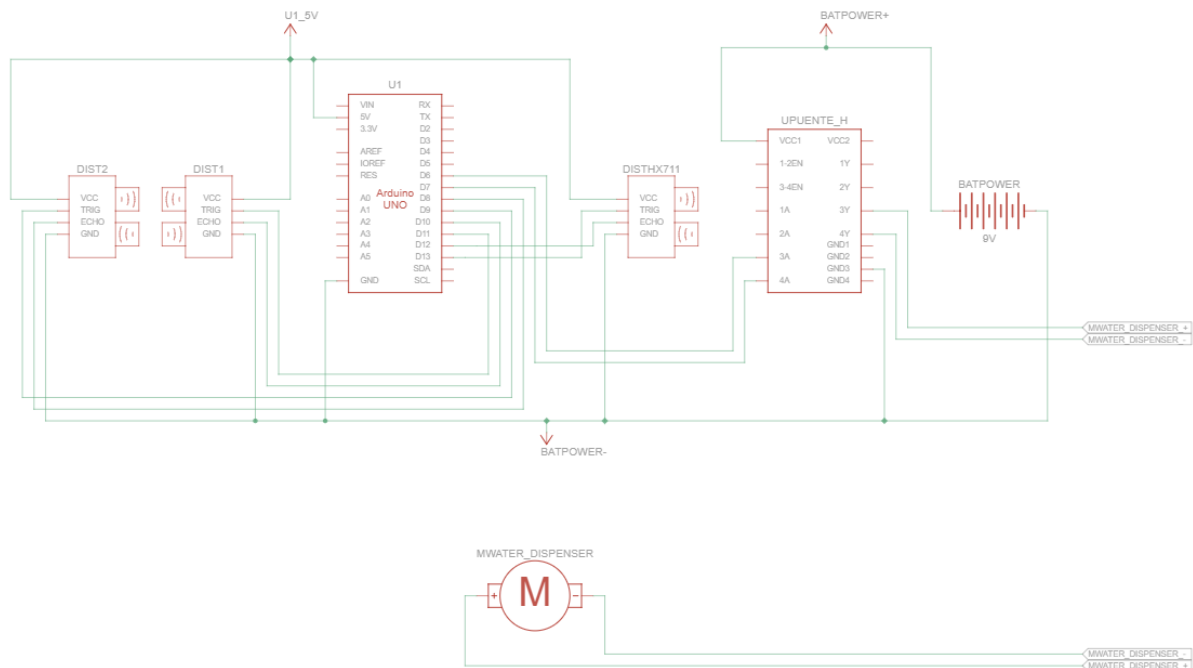
2.15 AWS IoT Analytics

Almacena los datos procesados del dispositivo en un almacén de datos de serie temporal optimizado para brindar tiempos de respuesta rápidos en consultas de IoT. Los datos sin procesar también se almacenan automáticamente para su procesamiento posterior o para volver a procesarlos en otro caso de uso.

3. MARCO PRÁCTICO

3.1.DISEÑO

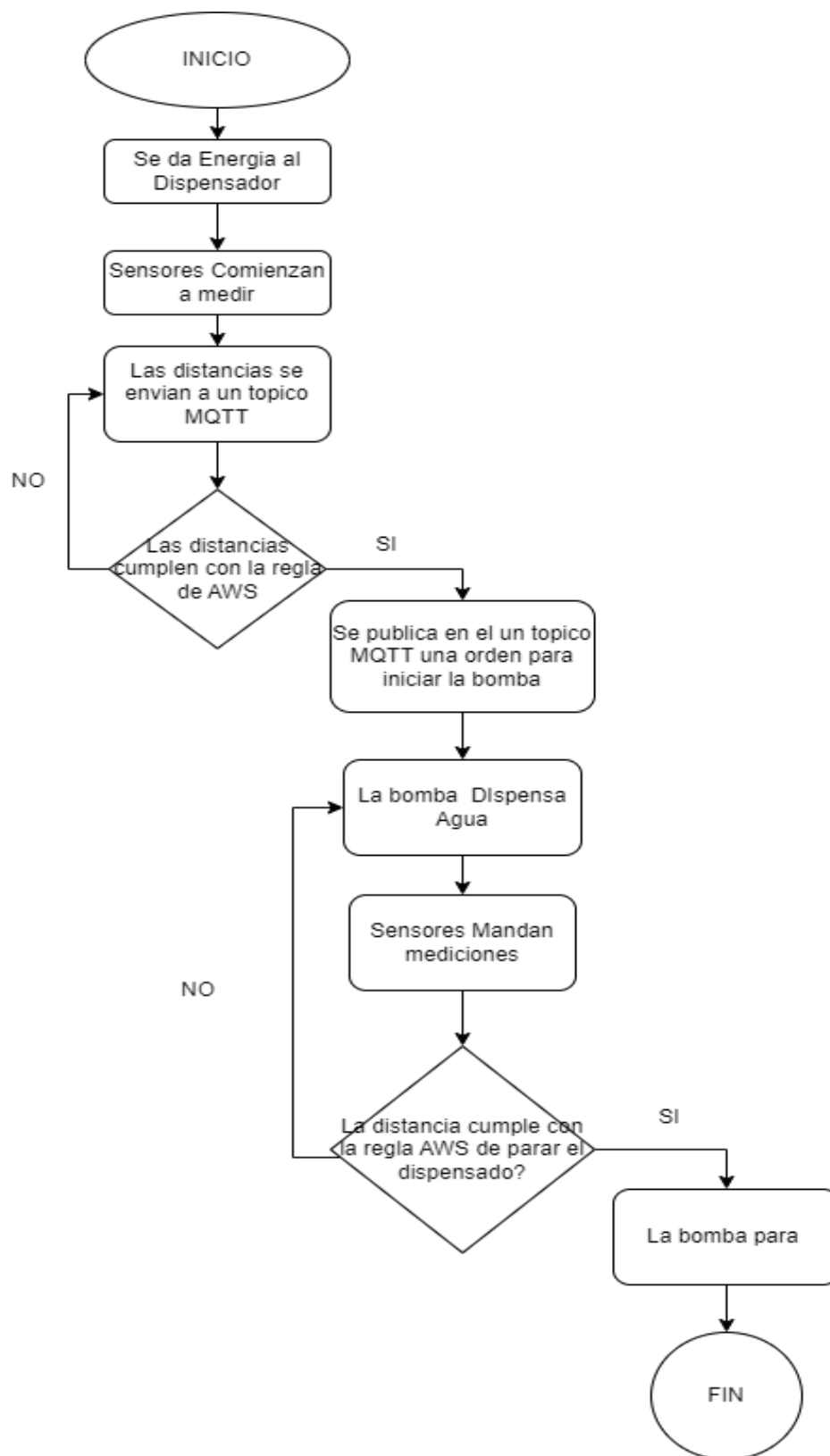
3.1.1.Arquitectura



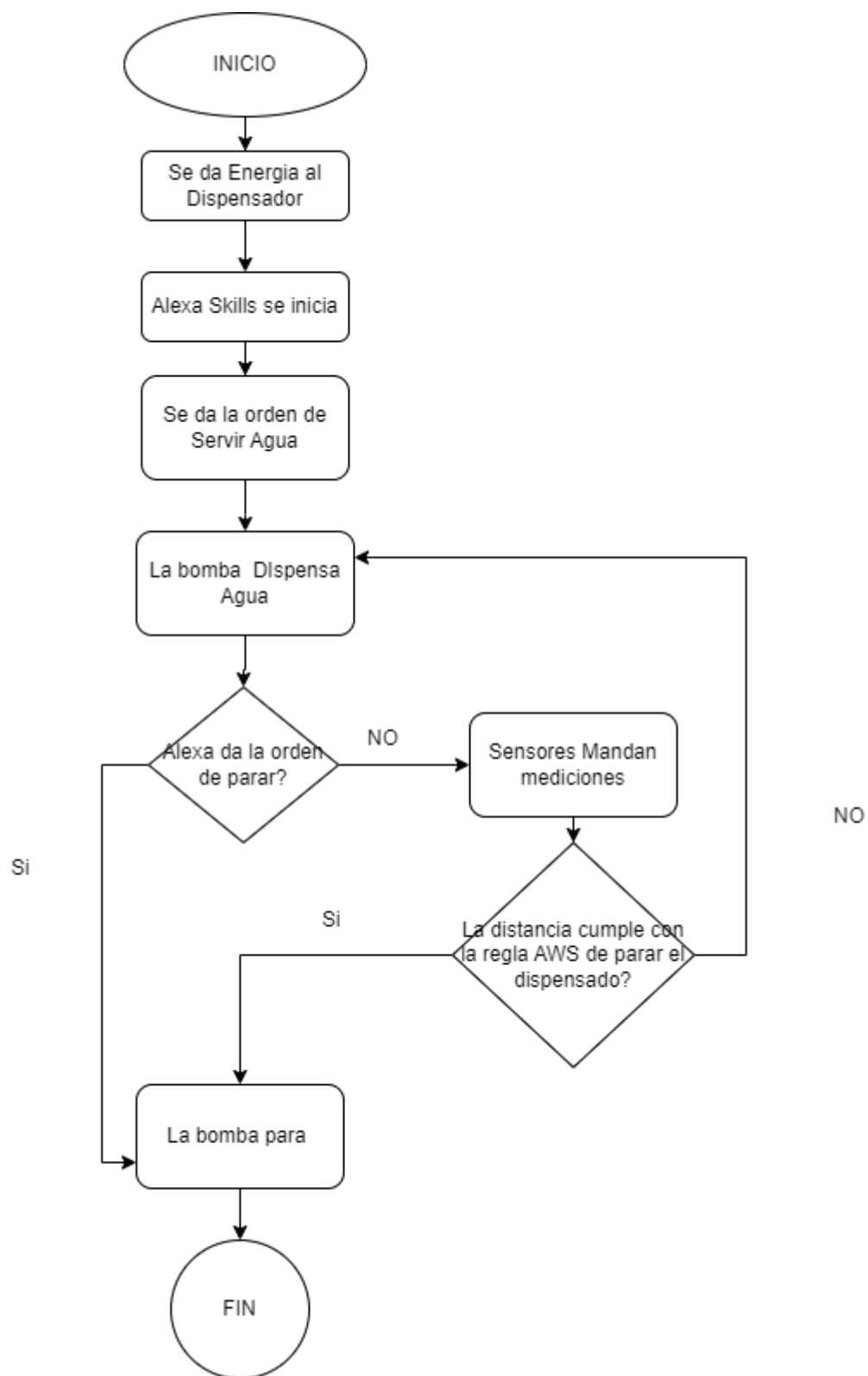
(3.1.1.1 Arquitectura del proyecto implementado,Fuente:Tinkercad Fuente Propia)

Como se puede observar en la Imagen^(1.2), Se aprecia la arquitectura del modelo o proyecto que se implementó en el simulador TINKERCAD donde dos sensores ultrasónicos se utilizan para medir las distancias del vaso y el líquido. Dependiendo de las distancias que miden enciende o apaga el dispensador. Nuestro proyecto cumple esta misma Arquitectura y lógica, con la única diferencia que no se utiliza un ARDUINO UNO sino un Tarjeta-ESP-32

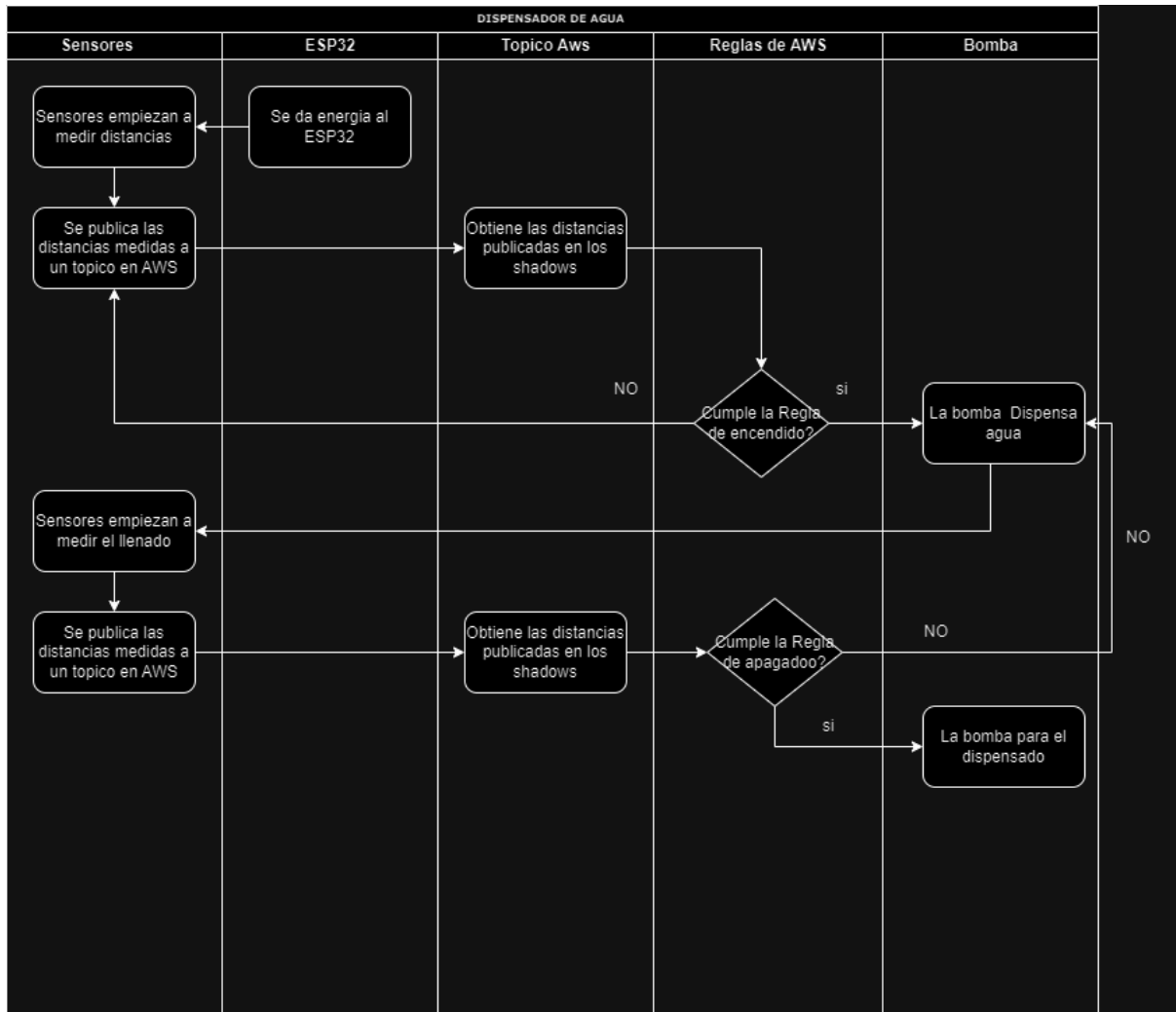
3.1.2. Diagramas Dinámicos



(Diagrama 3.1.2.1 Funcionamiento del dispensador ; Elaboración Propia)

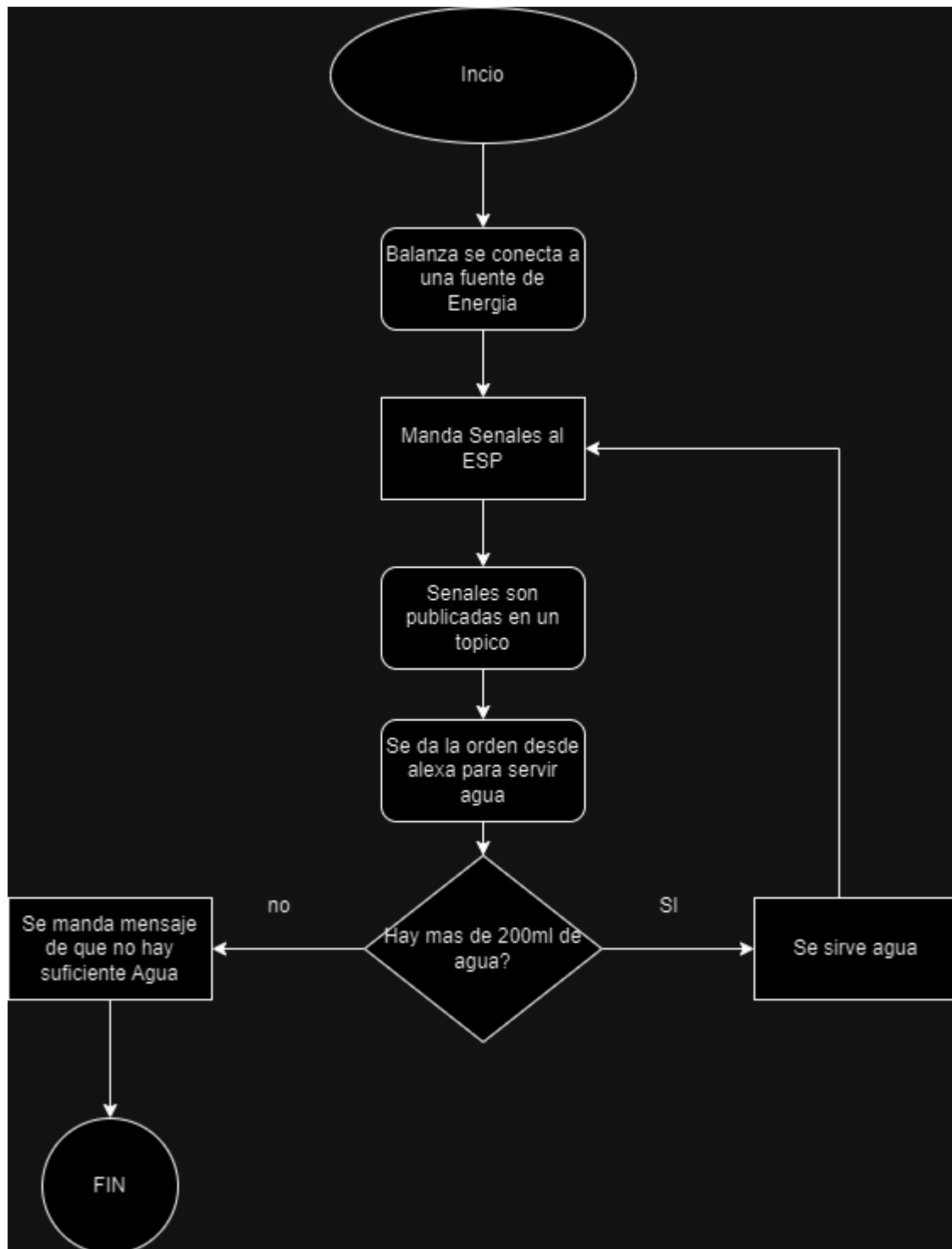


(Diagrama 3.1.2.2 Funcionamiento del dispensador Con instrucciones de Alexa; Elaboración Propia)



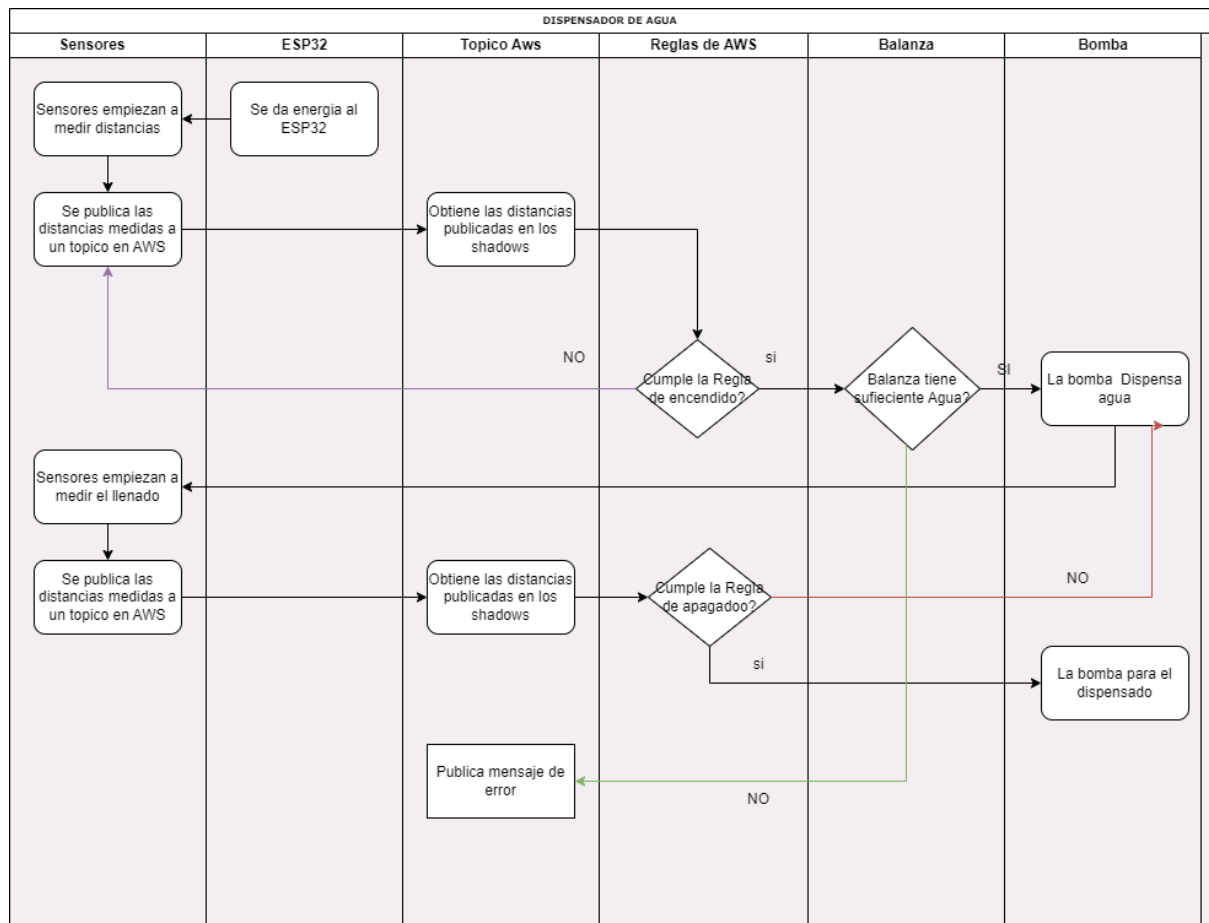
(Diagrama 3.1.2.3 Funcionamiento del dispensador en un diagrama de Estado; Elaboración Propia)

En los diagramas elaborados se puede observar el funcionamiento del dispensador de agua, donde se ve que una vez que el ESP32 recibe energía los sensores comienzan a medir a su vez publicarlos en el tópico de AWS donde reporta dos distancias, una vez publicadas las reglas comienzan a comprobar si cumple las reglas o no, cuando se acerca un vaso a los sensores, esta distancia disminuye por lo que hace que se cumpla la regla de encendido y esta regla publica una orden en otro tópico de AWS para hacer que se encienda la bomba, una vez encendida la bomba los sensores comienzan a controlar y a medir el dispensador de agua, de igual forma publicando las distancias en el tópico de AWS y cuando la regla de apagado se cumple al igual que cuando se enciende la bomba esta regla publica una señal en el tópico para poder apagar el dispensador de agua.



(Diagrama 3.1.2.4 Funcionamiento de la balanza para el dispensado de agua de Estado; Elaboración Propia)

En este Diagrama se observa detalladamente cómo implementamos el uso de la balanza para nuestro dispensador de Agua.



(Diagrama 3.1.2.5 Funcionamiento del dispensador con la balanza implementada para el dispensado de agua de Estado; Elaboración Propia)

En este Diagrama de Flujo se ve detalladamente como funciona nuestro dispensador con la balanza implementada y nos muestra cómo nos ayuda a saber la cantidad de agua que hay en ese momento, además de otras funciones(Alexa Skills) que se pueden observar en el diagrama 3.1.2.5.

3.2.DESARROLLO

Configuración del Hardware

- Conexión de Sensores:

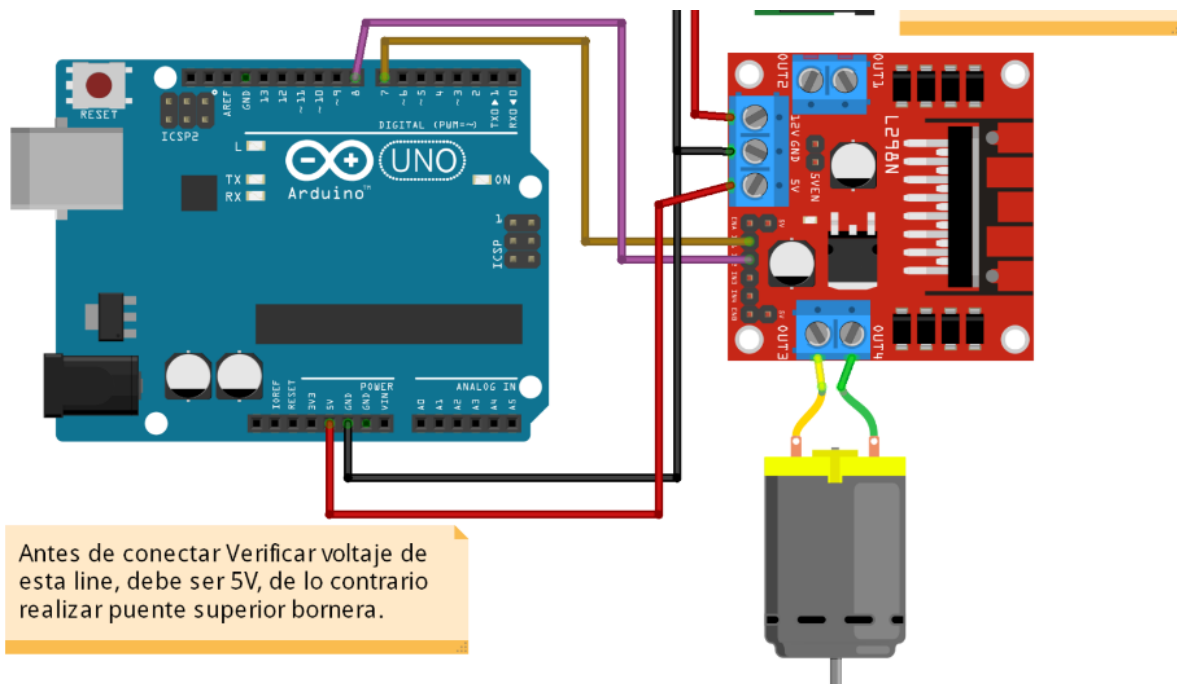
Conecta los sensores de distancia al ESP32.

Se verifica que los sensores estén recibiendo la alimentación adecuada y que el ESP32 pueda leer las lecturas de distancia correctamente.

- Configuración de la Bomba:

Para controlar una bomba de agua desde un ESP32, se necesita un puente H. Un puente H es un circuito electrónico que permite controlar la dirección del flujo de la corriente. En el caso de una bomba de agua, el puente H se utiliza para encender y apagar la bomba.

El puente H se compone de cuatro transistores, dos de los cuales se utilizan para encender la bomba y otros dos para apagarla. El ESP32 puede controlar los transistores del puente H mediante señales digitales.



Configuración de AWS IoT Core

- Creación de un Thing en AWS IoT Core:

Se accede a la consola de AWS IoT y se crea un Thing para representar al dispensador. Este genera certificados y claves para autenticar tu dispositivo en AWS IoT Core.

- Configuración de Reglas:

Se crean reglas en AWS IoT Core basadas en las distancias enviadas por los sensores. Se asocia acciones a estas reglas para publicar mensajes en un tópico específico cuando se cumplan las condiciones.

Programación del ESP32

- Configuración del WiFi y Conexión a AWS IoT Core:

Se utiliza la biblioteca WiFi del ESP32 para conectarlo a tu red WiFi.

Se utilizan las bibliotecas de AWS IoT para autenticar el ESP32 en AWS IoT Core y suscribirse al tópico correspondiente.

- Manejo de Señales y Control de la Bomba:

Se programa el ESP32 para reaccionar a las señales recibidas del tópico de AWS IoT Core. Al recibir la señal de inicio, inicia la bomba; al recibir la señal de parada, se detiene la bomba.

Integración con Alexa y AWS Lambda

- Configuración de la Skill de Alexa:

Se crea una Skill de Alexa en el Alexa Developer Console para tu dispensador de agua. Configura las invocaciones y las interacciones de voz.

- Creación de Funciones Lambda:

Se crea una función Lambda en AWS Lambda para manejar las solicitudes de la Skill de Alexa.

Implementa la lógica para activar y desactivar el dispensador de agua según las solicitudes.

- Conexión de la Skill con AWS IoT Core:

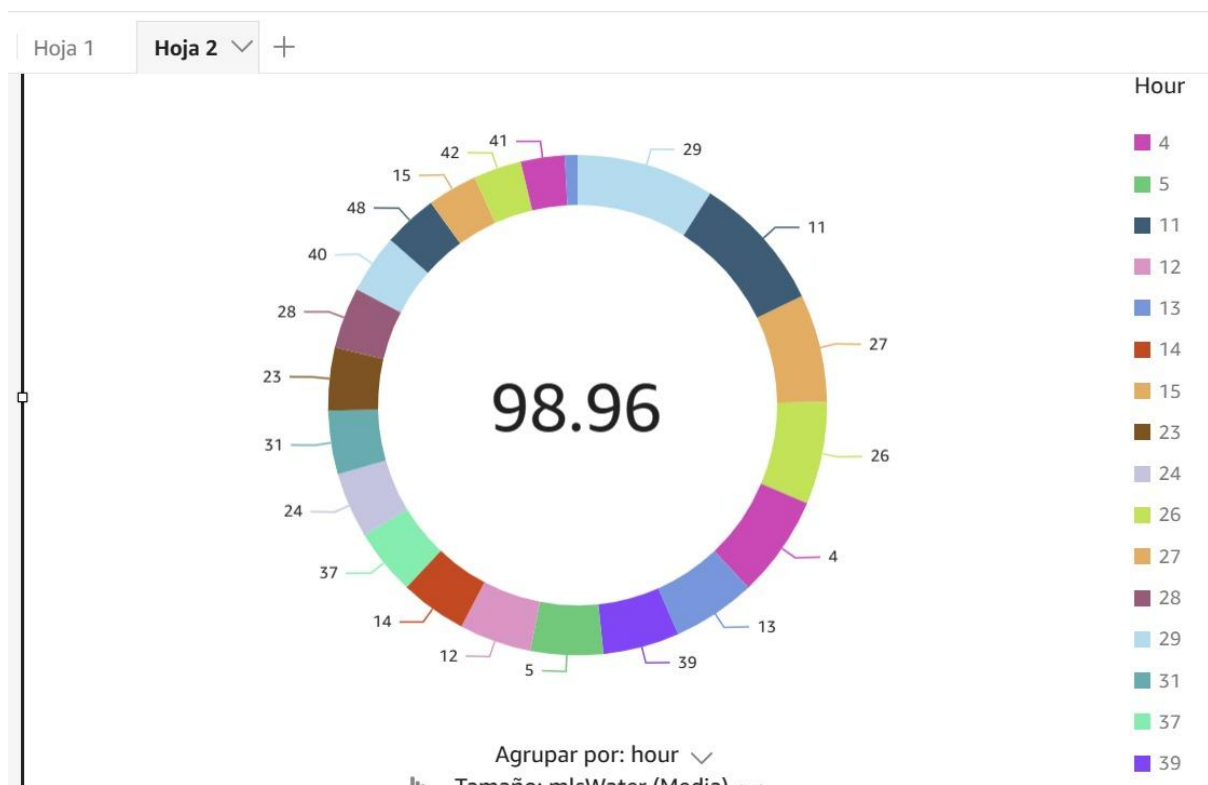
Se configura la Skill de Alexa para enviar comandos a tu dispositivo a través de AWS IoT Core, y con las funciones Lambda se envían señales al tópico de AWS IoT Core para activar o desactivar el dispensador.

Integración de Balanza para saber la cantidad de Agua

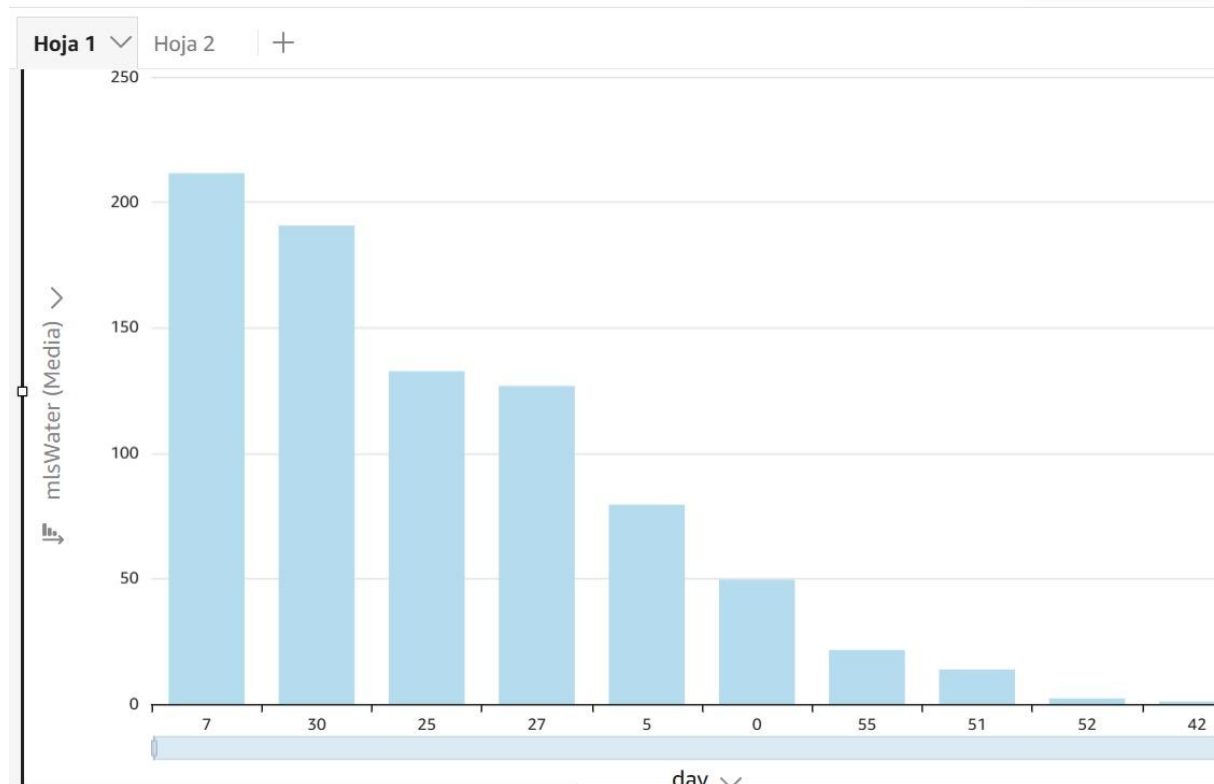
- Para integrar nuestra Balanza a nuestro sistema lo primero que se realizó es implementar y añadir el circuito a nuestro ESP32 este circuito está compuesto por una celda de carga que es el encargado de medir el cambio cuando se coloca un peso y esto manda una señal a un módulo que convierte esta señal en datos que nuestra ESP pueda entender.
- Después de eso se añadió al código ARDUINO instrucciones para que pueda leer este peso en miligramos y publicar en los shadows de AWS iot Core
- También se implementó un INTENT en Alexa Skills para poder controlar y saber cuánta agua queda disponible en nuestro Dispenser.

AWS IOT ANALYTICS

Para concluir con nuestro dispensador con el uso de las herramientas de AWS utilizamos AWS IOT ANALYTICS para poder llevar reportes de nuestro proyecto como ser la cantidad de agua que se tiene en el día, cada cuanto tiempo se recarga agua a nuestro dispensador y por último la cantidad de agua que se dispensó.



En este gráfico se puede observar la cantidad total de agua disponible por hora. y la media total de la cantidad de agua que se tiene cada hora, este gráfico nos ayuda a saber que tanta cantidad de agua se consume por hora



En este gráfico se puede ver la cantidad de agua que hay disponible en el dispensador, y que a medida que el tiempo pasa la cantidad de agua disminuye, también gracias a este reporte podemos ver que día se consume más cantidad de agua.

3.3 CÓDIGO FUENTE

[ibmg25/Final_IoT \(github.com\)](https://github.com/ibmg25/Final_IoT)

3.4 PRUEBAS

Una vez implementado el circuito y desarrollado el código y conexiones correspondientes, se procedió a la realización de diversas pruebas. Se adjuntan algunas imágenes de las mismas al final del documento.

Pruebas de los sensores ultrasónicos:

Nro Prueba	Medida Sensor Ultrasónico 1 (cm)	Medida Sensor Ultrasónico 2 (cm)	Medida Real (cm)
1	3	3	3
2	7	7	7
3	5	5	5
4	9	8	8
5	7	7	7
6	5	5	5
7	7	7	7
8	18	19	18
9	12	12	12
10	18	18	18
11	15	16	15
12	21	24	23
13	14	14	14
14	35	37	38
15	47	45	47
16	53	50	51
17	40	42	42
18	32	32	32
19	29	30	31
20	22	22	22
21	53	54	55
22	1	1	1

23	29	28	29
24	11	11	11
25	42	45	44

Pruebas con la bomba de agua alimentada por el pin 5V del ESP32 para llenar un vaso de 200 ml:

Nro Prueba	Tiempo llenado (s)
1	19.23
2	18.89
3	19.15
4	19.49
5	18.94
6	18.46
7	19.03
8	19.81
9	19.37
10	19.20

Pruebas con un dispensador comercial para llenar un vaso de 200 ml:

Nro Prueba	Tiempo llenado (s)
1	8.45
2	8.53
3	8.51
4	8.46
5	8.51
6	8.49
7	8.48
8	8.52
9	8.50
10	8.49

Pruebas bomba de agua alimentada con batería externa perteneciente al dispensador comercial para llenar un vaso de 200 ml:

Nro Prueba	Tiempo llenado (s)
1	11.93
2	11.97
3	12.06
4	11.94
5	12.04
6	11.95
7	12.03
8	12.06
9	11.94
10	11.98

Pruebas bomba de agua alimentada con una fuente externa de 6V para llenar un vaso de 200 ml:

Nro Prueba	Tiempo llenado (s)
1	12.43
2	12.47
3	12.36
4	12.44
5	12.51
6	12.45
7	12.53
8	12.46
9	12.48
10	12.44

Pruebas Sensor de Peso:

Nro Prueba	Peso Real (gr)	Medida Obtenida (gr)
1	20	0
2	150	8
3	40	0
4	230	29
5	10	3
6	500	509
7	750	761
8	600	593
9	980	972
10	800	812
11	1500	1507
12	2000	2056
13	1800	1788
14	2000	2041
15	1500	1525

Pruebas Alexa:

Nro Prueba	Reconoce On	Tiempo resp. On (s)	Reconoce Off	Tiempo resp. Off (s)
1	Si	0.49	Si	0.46
2	No	-	-	-
3	Si	2.09	Si	0.83
4	Si	0.52	Si	0.61
5	Si	0.44	Si	0.56
6	Si	0.61	No	-
7	Si	0.50	Si	0.62
8	Si	1.46	Si	0.74

9	Si	0.45	Si	1.25
10	Si	0.71	Si	0.80
11	Si	0.43	Si	0.97
12	No	-	-	-
13	Si	0.50	Si	0.52
14	Si	0.48	Si	0.79
15	Si	0.74	Si	1.91
16	Si	0.62	Si	0.68
17	Si	1.83	Si	0.54
18	No	-	-	-
19	Si	0.91	Si	2.03
20	Si	0.49	Si	0.62

Pruebas Alexa Peso:

Nro	Cantidad Aprox Agua	Respuesta Alexa
1	0 ml	No hay suficiente agua
2	250 ml	Hay 291 ml
3	1000 ml	Hay 1053 ml
4	2000 ml	Dispensador lleno
5	0 ml	No hay suficiente agua
6	250 ml	No hay suficiente agua
7	1000 ml	Hay 1071 ml
8	2000 ml	Dispensador lleno
9	0 ml	No hay suficiente agua
10	250 ml	No hay suficiente agua
11	1000 ml	Hay 1012 ml
12	2000 ml	Dispensador lleno
13	0 ml	No hay suficiente agua
14	250 ml	Hay 272 ml

15	1000 ml	Hay 1059 ml
16	2000 ml	Dispensador lleno
17	0 ml	No hay suficiente agua
18	250 ml	No hay suficiente agua
19	1000 ml	Hay 1025 ml
20	2000 ml	Dispensador lleno
21	0 ml	No hay suficiente agua
22	250 ml	No hay suficiente agua
23	1000 ml	Hay 1029 ml
24	2000 ml	Dispensador lleno

4. RESULTADOS Y CONCLUSIONES

A partir de las pruebas realizadas, se obtuvieron los siguientes resultados y posteriormente aportaron a las conclusiones:

Error en la medición

Respecto al error de medición, ya se había calculado en proyectos anteriores, y las nuevas pruebas corresponden a un error bastante similar.

El error del sensor ultrasónico es de aproximadamente 5%.

El error es bastante significativo en medidas pequeñas, sin embargo, las medidas tomadas fueron correctas aproximadamente el 50% de las veces.

El restante 50% de las pruebas, la medición, por las medidas con las que se trabajaron, llegó a tener un error de entre 1 y 3 cm.

Tiempo de llenado del vaso de 200 ml

El tiempo de llenado del vaso es relativamente elevado. Se tarda casi 12 segundos en llenar el vaso desechable cuando se alimenta la bomba de agua con una fuente externa de 6V. Se realizaron comparaciones con un dispensador de agua eléctrico comercial y se obtuvieron resultados similares en el tiempo de llenado.

Conexión con Alexa

Las pruebas con Alexa fueron donde mayor problema se tuvo. En un par de ocasiones, Alexa no reconocía el comando correctamente, al no interpretarlo completamente igual a como se especifica en el intent, no realizaba la acción correspondiente.

También pasó un par de veces que a pesar de que Alexa reconocía el comando y decía que estaba ejecutando la acción correspondiente, el circuito no la realizaba, lo que se atribuye a problemas de conexión.

Sin embargo, en aproximadamente el 80% de las pruebas, Alexa reconocía correctamente el comando y la acción se ejecutaba correctamente con un tiempo de reacción casi despreciable (menor a 1 segundo).

Sensor de Peso

Para pesos mayores a 500 gr, el sensor de peso obtenía resultados coherentes. A pesar de no ser un sensor muy preciso, el error no supera el 10% de la medición, sin embargo, al no requerir mucha precisión, el proyecto es funcional.

Respecto a medidas menores, al estar usando un sensor de peso que soporta hasta 10 kg, se tuvo problemas con la precisión al momento de obtener estas medidas, sacando resultados lejanos a la realidad.

Conexión con del sensor de peso en Alexa

Para saber la cantidad aproximada de agua que había en el dispensador se le preguntaba a Alexa por el último peso registrado por el dispositivo. Alexa respondió bien en la mayoría de las pruebas, solo fallaba eventualmente cuando el agua estaba cerca a los 200 ml. Esto debido a las razones explicadas en el punto anterior.

Conclusiones

- Los sensores ultrasónicos para medir las distancias y determinar cuándo se debe activar el dispensador funcionan correctamente a pesar de no ser tan precisos; ya que simplemente deben detectar la presencia del vaso y la altura del líquido, no se tuvo problemas en este aspecto.
- El sensor de peso es preciso cuando se coloca una botella de 2 litros llena, mientras la botella va perdiendo líquido, se miden pesos coherentes cercanos a la realidad, sin embargo, a partir de valores menores a 300 ml, el sensor pierde mucha precisión y obtiene resultados lejanos a los valores reales.
- Alexa responde correctamente a la consulta de la cantidad de agua en el dispensador. Para simplificar un poco las respuestas Alexa responde lleno si el peso es de 2 litros o más; no hay suficiente agua si hay menos de 200 ml; y la cantidad aproximada de agua que hay si es que el valor se encuentra entre estos dos límites.
- La bomba de agua alimentada por el ESP32 es lenta al momento de llenar el vaso de agua y no compite contra otros dispensadores existentes en el mercado al tardar más del doble de tiempo en llenar un vaso de 200 ml.
- Al alimentar la bomba con una fuente externa de 6V, se obtuvieron resultados similares al tiempo de llenado del dispensador comercial.
- Este dispensador tiene la ventaja de contar con sensores y al mismo tiempo estar conectado con Alexa, lo que en cierto modo genera más retraso en el llenado de un vaso, pero está automatizado y no requiere encenderse manualmente.
- La conexión con Alexa trae algunos problemas y no siempre funciona correctamente. En un porcentaje de casos Alexa no reconoce bien el comando o a pesar de reconocerlo no se ejecuta en el circuito. No obstante, las pruebas muestran que en un 80% de los intentos, Alexa ejecutó las acciones correspondientes sin un tiempo de retraso significativo.
- Finalmente, se concluye que el proyecto cumple satisfactoriamente con los requisitos del proyecto, al conectarse correctamente con AWS IoT Core para la implementación de reglas y con Alexa Skills para la ejecución de comandos desde su interfaz. Además, el dispensador de agua funciona y llena el vaso ya sea cuando los sensores lo detectan, o cuando Alexa se lo ordena.

Recomendaciones

- Por cuestiones de comodidad y simplicidad se alimenta el dispensador con el pin 5V del ESP32, sin embargo, para mayor eficiencia en el llenado, se recomienda alimentar la bomba con una batería externa de mayor voltaje.
- En caso de querer competir con dispensadores comerciales en lo que respecta al llenado, se recomienda adquirir una bomba de mayor potencia.
- Se recomienda que la celda que se debe utilizar para el sensor de peso sea de una magnitud menor, no superando el límite de 5 kg.

5. BIBLIOGRAFÍA

<https://docs.aws.amazon.com/lambda/latest/dg/welcome.html>

<https://docs.aws.amazon.com/iot/latest/developerguide/what-is-aws-iot.html>

<https://docs.espressif.com/projects/esp-idf/en/latest/esp32/>

<https://docs.aws.amazon.com/lambda/latest/dg/getting-started.html>

<https://docs.aws.amazon.com/iot/latest/developerguide/what-is-aws-iot.html>

<https://developer.amazon.com/en-US/docs/alexa/sdk/alexa-skills-kit-sdk.html>

https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf

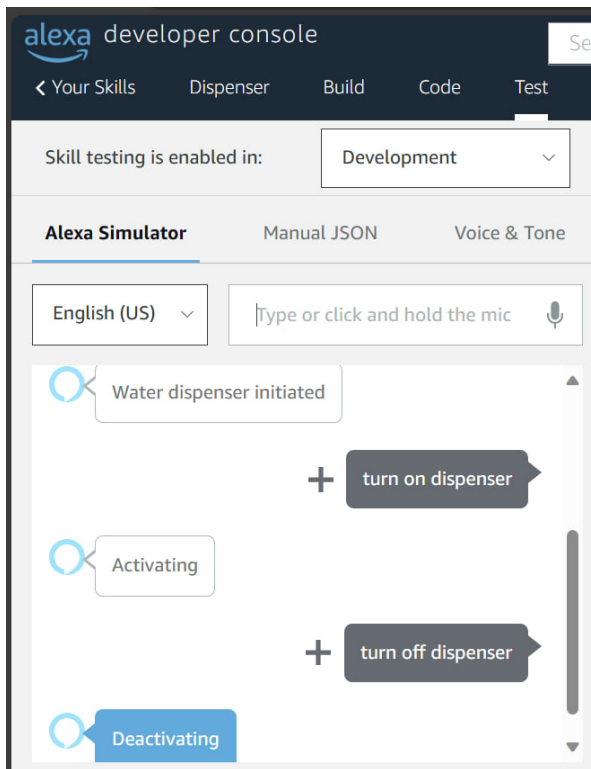
<https://www.electroschematics.com/wp-content/uploads/2013/07/HCSR04-datasheet-version-1.pdf>

<https://naylorlampmechatronics.com/sensores-proximidad/10-sensor-ultrasonido-hc-sr04.html>

6. ANEXOS

6.1 Proyecto para las pruebas





```
{
  "state": {
    "desired": {
      "dispenser": "off"
    }
  }
}
```

► Propiedades

▼ \$aws/things/dispenser/shadow/update
November 27, 2023, 19:44:08 (UTC-0400)

```
{
  "state": {
    "desired": {
      "dispenser": "on"
    }
  }
}
```

🔍 🔔 ⓘ ⚙️ Ohio ▾

▼ \$aws/things/dispenser/shadow/update/accepted
November 27, 2023, 19:44:08 (UTC-0400)

```
{
  "state": {
    "desired": {
      "dispenser": "on"
    }
  },
  "metadata": {
    "desired": {
      "dispenser": {
        "timestamp": 1701128648
      }
    }
  },
  "version": 295,
  "timestamp": 1701128648
}
```

🔍 🔔 ⓘ ⚙️ Ohio ▾

```
{
  "state": {
    "reported": {
      "distanceGlassSensor": 55,
      "distanceLiquidSensor": 805
    }
  }
}
```

► Propiedades

▼ publish/led November 27, 2023, 19:49:14 (UTC-0400)

```
{
  "state": {
    "reported": {
      "distanceGlassSensor": 4,
      "distanceLiquidSensor": 5
    }
  }
}
```

Estado
🟢 Activo

tema de ingesta básico
📄 \$aws/rules/encenderDispensador

Instrucción SQL

Instrucción SQL	Versión de SQL
SELECT 'on' AS action FROM 'publish/led' WHERE state.reported.distanceGlassSensor<10 AND state.reported.distanceLiquidSensor<10	2016-03-23

Acciones

Acción de error

Etiquetas

Acciones (1)

Ver detalles

Las acciones se producen cuando se activa un evento. Se ejecutan en orden ascendente, hasta que todas las acciones se completan o se produce un error. Para agregar o eliminar acciones, deberá editar la regla.

Servicio	Acción
○ Republish to AWS IoT topic	Volver a publicar mensajes en un tema de AWS IoT

CUSTOM

> Invocations

> Interaction Model

> Intents (8)

Annotation Sets

Intent History

JSON Editor

> Assets

Slot Types (0)

Multimodal Responses

Tasks

Interfaces

Save

Evaluate model

Update live skill

Last successful build: 11/27/2023 @ 6:01 PM

View history

Build skill

AMAZON.NavigateHomeIntent	-	-	Required	Edit
AMAZON.FallbackIntent	-	-	Built-In	Edit Delete
dispenserOnIntent	2	-	Custom	Edit Delete
dispenserOffIntent	2	-	Custom	Edit Delete

< 1 – 8 of 8 Intents >

Errors and Warnings

English (US) Feedback

© 2010 – 2023, Amazon.com, Inc. or its affiliates. All Rights Reserved. Terms Docs Forums Blog Alexa Developer Home