



Universidad Nacional
de La Matanza

Sistemas Operativos Avanzados

Asistente de Cocina

“La Jarra de Ada”

Integrantes:

- Facundo Nicolás Vogel
- Sebastián Alejandro Robles Wagner
- Sebastián Gronski
- Santiago Tamashiro
- Federico Gomez Markowicz

Índice

Objetivo	3
Alcance del Sistema	3
Sistema embebido arduino	3
Aplicación Android	3
Descripción general del Proyecto	4
Construcción	6
Componentes	6
Descripción de Sensores	6
Celda de carga 5kg con amplificador Hx711	6
Sensor de humedad y temperatura DHT11	7
Sensor detector de obstáculos infrarrojo FC-51	8
Descripción de Actuadores	9
Motor de corriente continua	9
Buzzer	10
Luz Led	10
Otros	11
Puente H L298N	11
Implementación lógica de la Embebido	13
Circuito	15
Limitaciones del Prototipo	16
Aplicación Android	17
Descripción general	17
Actividades	17
Comunicación	18
Sensores	19
Instrucciones de uso	20
Limitaciones de la aplicación	27
Fuentes	28

Objetivo

El objetivo de este proyecto es facilitar la resolución de ciertas tareas a la hora de cocinar. Su función es extraer cantidades exactas de distintos ingredientes indicados a través de una receta dada, para que el usuario pueda desentenderse de dicha tarea y enfocarse solo en cocinar.

Alcance del Sistema

Sistema embebido arduino

El sistema embebido arduino es el que provee el funcionamiento inteligente para poder realizar la extracción del ingrediente satisfactoriamente que se encuentra dentro de la caja. Para cumplir su funcionamiento cuenta con los siguientes alcances:

- Debe detectar presencia de un recipiente para poder depositar el ingrediente.
- Extraer ingrediente mediante el uso de un tornillo sin fin.
- Comunicación bluetooth con el dispositivo, para saber la cantidad de ingrediente a extraer e informar la cantidad de ingrediente en el recipiente.
- Informe del estado de sus componentes por medio de luces led.
- Proveer información acerca del estado de los ingrediente, tales como humedad y temperatura.

Aplicación Android

La aplicación mobile para Android es el sistema que provee la comunicación con el teléfono mobile, esta comunicación es mediante vía bluetooth. Provee una interfaz sencilla de usar para el usuario. Cumple con los siguientes alcances:

- Permite introducir en una lista los distintos ingredientes con su cantidad. .
- Un menú para poder introducir las distintas recetas.
- Menú de bluetooth para la conexión con el sistema embebido.
- Control manual en la que ofrece poder operar manualmente (como activar el sin fin, detenerlo) mediante el uso de los sensores. Para cumplir estas funciones de control manual, el sistema hace uso de los sensores del framework de android: sensor de proximidad, giroscopio y luminosidad.

Descripción general del Proyecto

Asistente de cocina nos permite extraer cantidades exactas de ingrediente cuando se quiere cocinar en base a una receta, lo cual esta receta está introducida en la aplicación de android. Esto facilita al momento de cocinar, ya que permite al usuario desentenderse en cuanto y como extraer los ingrediente ya que se tienen cargados los ingredientes a utilizar al igual que sus cantidades.

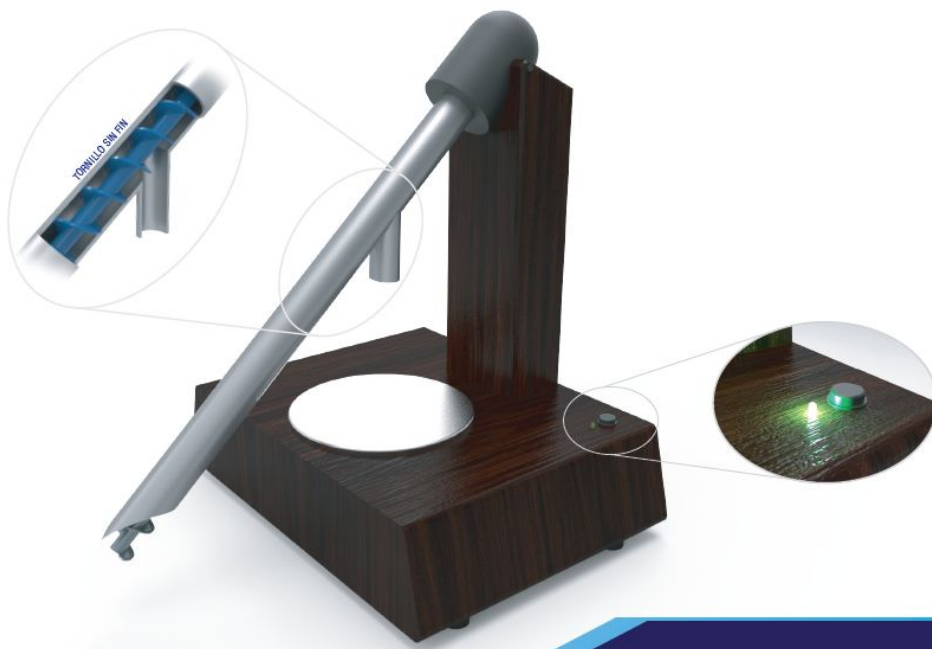
El asistente de cocina consiste en uno o más recipientes con ingredientes asentados sobre una base, la cual posee un brazo que se mueve entre los diferentes recipientes. Una vez posicionado, un sinfín extrae el ingrediente y lo deposita en un contenedor.

Para la extracción de ingrediente el sistema cuenta con un sensado del peso mediante una balanza y así poder extraer del recipiente la cantidad deseada. Lo cual para extraerlo se tiene un mecanismo de tornillo sin fin accionado por un motor y unos caños para dirigir el ingrediente hacia el recipiente destino.

El asistente puede ser controlado mediante una aplicación para celulares Android, la cuál puede cargar recetas para preestablecer las cantidades que deberá extraer el dispositivo. Además, puede sensar la humedad y temperatura para determinar si los alimentos están en buen estado para ser consumidos, lo cual lo informará desde la aplicación de Android.

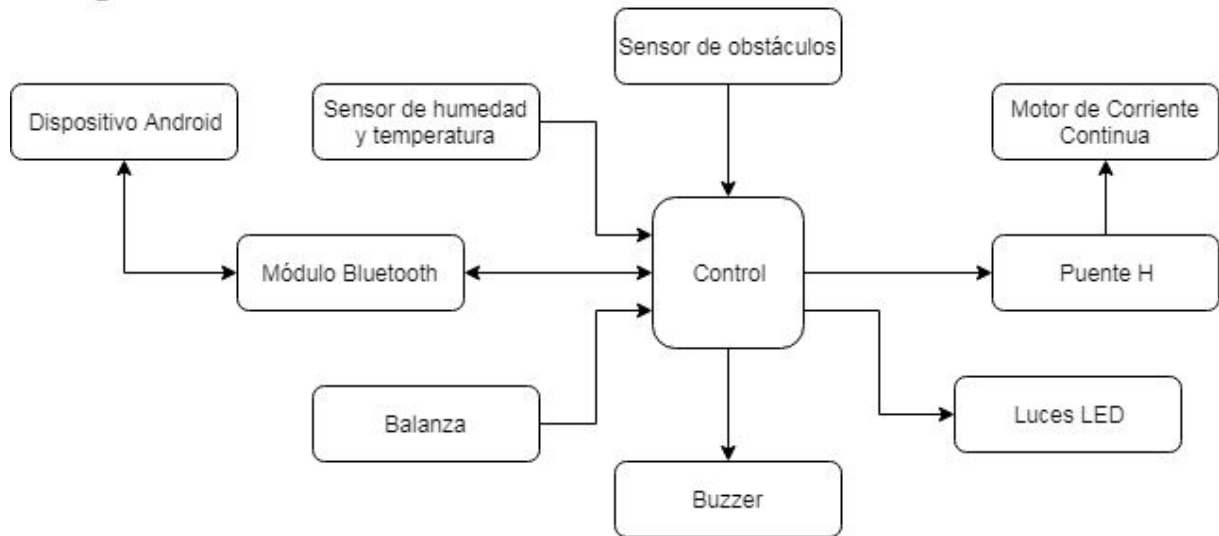
A continuación se visualiza el bosquejo de cómo se vería el producto:

ASISTENTE DE COCINA



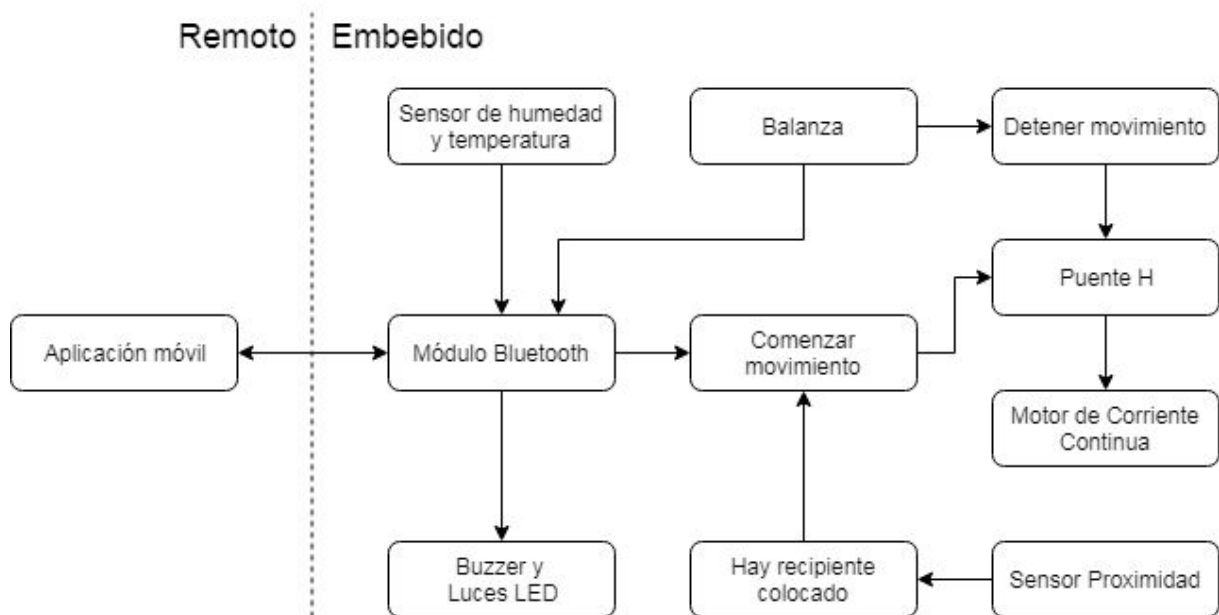
A grandes rasgos, el dispositivo debe funcionar siguiendo este diagrama:

Diagrama Funcional



Y reacciona según el siguiente diagrama:

Diagrama Lógico



Estos diagramas permiten tener una noción básica sobre qué debe hacer el sistema. Más adelante se detallarán las partes físicas que lo componen y cómo funcionan, y las partes del programa software que se encuentra compilado en el Arduino.

Construcción

Componentes

Los componentes utilizados para la elaboración del dispositivo son:

- Arduino Mega
- Celda de carga 5kg con amplificador HX711
- Sensor de humedad relativa y temperatura DHT11
- Sensor de obstáculos infrarrojo FC-51
- Buzzer
- Motor eléctrico de corriente continua
- Luz Led
- Puente H L298N
- Caja
- Caños PVC de 40mm
- Cables
- Resistencias
- Fuente de 9v

Descripción de Sensores

Celda de carga 5kg con amplificador Hx711



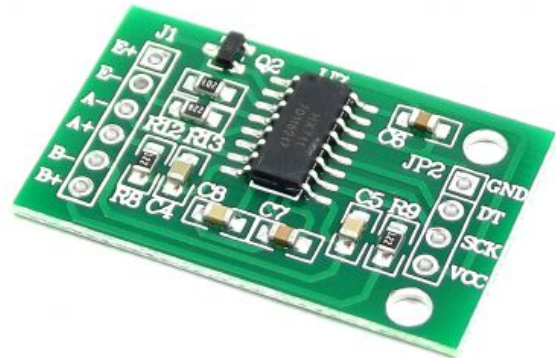
La celda de carga es un transductor que se encarga de convertir una fuerza (en nuestro caso, el peso de un objeto) en una señal eléctrica. Para esto, la celda posee una o más galgas extensiométricas configuradas en un puente Wheatstone, que es un circuito eléctrico en el cual, conociendo los valores de 3 resistencias, se puede calcular el valor de una cuarta resistencia desconocida.

Las galgas extensiométricas miden la deformación que sufre un cuerpo basándose en el efecto piezorresistivo, esto es, una propiedad característica de algunos materiales que cambian el valor de su resistencia eléctrica al ser sometidos a un esfuerzo.

Por lo tanto, una fuerza aplicada sobre una galga causará una variación en la resistencia eléctrica, lo que en el circuito se traducirá como una variación de diferencia de potencial que será proporcional al peso que está provocando la deformación en el material. Esta señal es amplificada e interpretada por el HX711.

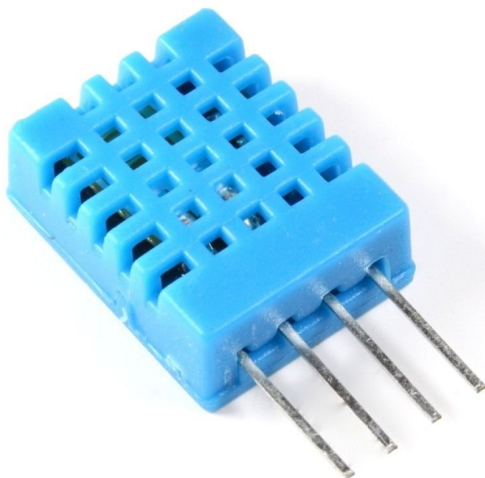
La celda de carga en nuestro caso es hueca en el medio ya que de lo contrario la deformación sería poco apreciable, y el cambio de diferencia de potencial resultaría imperceptible para el sensor.

Una vez instalado el sensor y la celda de carga, se debe comenzar el proceso de calibración, que es básicamente hallar el valor de la escala que se usará, ya que la escala es diferente para cada celda y cambia de acuerdo a la forma en la que se instaló, los materiales que se usaron para construirla etc.



Esto ocurre debido a que, al momento de la fabricación de la celda de carga, el puente de Wheatstone se encontraba en equilibrio. Pero distintos factores hacen que al momento de instalarla, el puente esté desequilibrado y éste debe ser corregido mediante software durante el proceso de calibración.

Sensor de humedad y temperatura DHT11



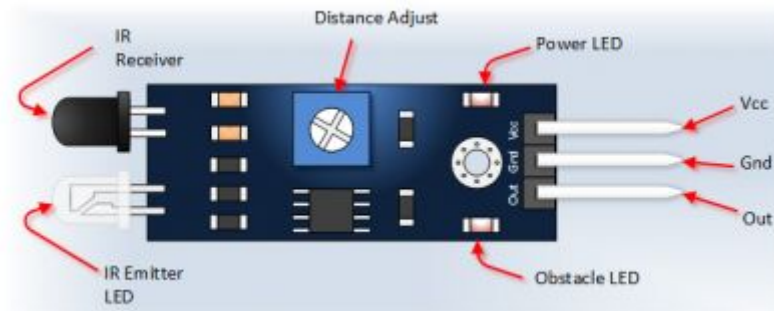
El sensor DHT11 tiene un termistor NTC montado sobre la misma unidad para medir la temperatura. El termistor es un semiconductor cuya resistividad varía según la temperatura del mismo. Al ser NTC, la resistencia disminuye a medida que aumenta la temperatura.

El sensor mide la humedad relativa del aire, esto es, $(\text{densidad de vapor} / \text{densidad del vapor en el punto de saturación}) \times 100\%$. La temperatura es importante debido a que la humedad relativa depende también de ella.

Para calcular la humedad, el DHT11 dispone de un sustrato que acumula humedad con dos electrodos en su superficie. Cuando el sustrato absorbe el vapor presente en el aire, se liberan iones que aumentan la conductividad entre los electrodos (por lo que la resistencia entre ellos disminuye) y este cambio de resistencia es proporcional a la humedad relativa. Por lo tanto, valores bajos de resistencia corresponden a una alta cantidad de humedad en el aire y valores altos de resistencia, a una baja cantidad de humedad en el aire.

Este cambio de resistencia deriva en un cambio en la diferencia de potencial, que es capturado por el chip del DHT11 y, a través de una biblioteca de Arduino, interpreta los valores de las señales eléctricas para calcular los valores de humedad y temperatura que llegan al Arduino.

Sensor detector de obstáculos infrarrojo FC-51



El emisor es un LED que emite radiación en la frecuencia infrarroja (actuando como un simple LED) y el receptor es un fotodiodo o fototransmisor (capaz de detectar radiación infrarroja emitida por el transmisor). El sensor constantemente emite una señal infrarroja a través del emisor que es invisible al ojo humano.

Si hay un objeto cuya superficies es reflectante, la luz infrarroja colisionará con el objeto y rebotará en varias direcciones (incluida la del sensor) que será detectada por el receptor de infrarrojos, indicando la presencia de un obstáculo en frente del emisor, por lo que el sensor enviará una señal en HIGH a través del OUT y encenderá su luz LED de obstáculo.

Ahora bien, si el receptor no captura luz infrarroja, pueden ocurrir dos situaciones:

- No hay ningún obstáculo en la distancia operativa del sensor.
- Hay un obstáculo pero su superficie absorbe las radiaciones infrarrojas, por lo que no rebota y no es capturada por el receptor. En este caso, el sensor es incapaz de detectar la presencia del obstáculo.

Este módulo además posee un potenciómetro que permite ajustar el rango de detección (mínimo 2 cm, máximo 30 cm).

Descripción de Actuadores

Motor de corriente continua



Un motor de corriente continua es un dispositivo que se encarga de convertir energía eléctrica en movimiento rotatorio.

Un motor de corriente continua se compone principalmente de dos partes. El estátor (parte estática) da soporte mecánico al aparato y contiene los polos de la máquina, que pueden ser o bien devanados de hilo de cobre sobre un núcleo de hierro, o imanes permanentes.

El rotor (parte giratoria) es generalmente de forma cilíndrica, también devanado y con núcleo, alimentado con corriente directa a través de delgas (láminas de cobre aisladas unas de otras), que están en contacto alternante con escobillas fijas.

El funcionamiento se basa en la Ley de Lorentz: Todo conductor por el que pasa una corriente y está sumergido en un campo magnético experimentará una fuerza.

El rotor del motor forma una espira a través de la cual circulará corriente, y el estátor está conformado por 2 imanes a los costados de la espira que generarán el campo magnético. Los segmentos laterales de la espira (que se encuentran de forma perpendicular al campo magnético) experimentarán una Fuerza de Lorentz que serán de sentido opuesto. Esto provocará un que la espira comience a rotar.

Para mantener que la espira siga acelerando y mantener su velocidad angular, cada media vuelta se debe invertir el flujo de corriente mediante un conmutador, de forma que las Fuerzas de Lorentz que experimentan los segmentos cambien de sentido y así la espira continuará girando. Si no se invierte el sentido de la corriente, llegará un momento en el que la posición de la espira causa que la Fuerza de Lorentz se oponga al movimiento de rotación que había comenzado.

El sentido de giro de estos motores puede manipularse mediante un Puente H.

Buzzer

Un Buzzer (o zumbador) es un aparato que convierte energía eléctrica en sonido.

El funcionamiento radica en el efecto piezoeléctrico: Determinados materiales, al ser sometidos a un campo eléctrico se deforman. Es el mismo principio que rige el funcionamiento de la celda de carga, solo que este es el caso opuesto, aplicamos una diferencia de potencial al material para provocar que su volumen se expanda.



El buzzer contiene dentro una placa de cerámica y una placa metálica. Al recibir corriente, las placas se deforman y producen un sonido casi imperceptible. Para obtener un sonido continuo, el buzzer contiene además un oscilador para que la señal eléctrica que llega a las placas oscile justamente. Esto provoca que las placas se expandan y se contraigan constantemente, generando así el ruido característico de estos dispositivos.

Luz Led



LED significa Light Emitting Diode. Un diodo es un componente electrónico que permiten el paso de corriente en un solo sentido. Un diodo emisor de luz es un diodo que, cuando se conecta en el sentido que permite el paso de la corriente, emite luz.

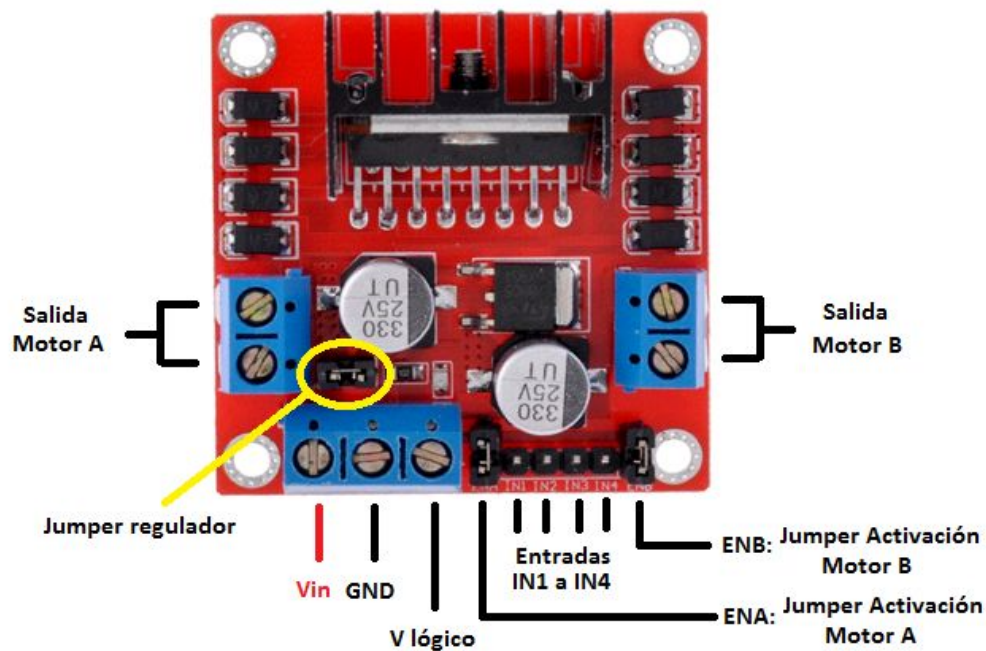
Los LED tienen dos patas, una más corta (Cátodo) que se conecta al negativo y una más larga (Ánodo) que se conecta al positivo. Si se conectan al revés, no pasará la corriente a través del diodo y no se emitirá luz alguna.

El fenómeno de emisión de luz se denomina Electroluminiscencia y se debe a que al aplicar una tensión determinada al material semiconductor del que está hecho el diodo, los electrones se recombinan liberando energía en forma de fotones. El color de la luz está determinado por el material semiconductor empleado en cuestión, por ejemplo para el color azul se puede usar una mezcla de Nitruro de Galio y Nitruro de Indio.

La diferencia de potencial a la que trabajan las luces LED dependen del color pero generalmente se encuentra entre 1.7V y 3.3V. La cantidad de luz que emite el LED depende de la tensión aplicada pero no es linealmente proporcional a ella, y una tensión muy elevada provocará que el LED se queme.

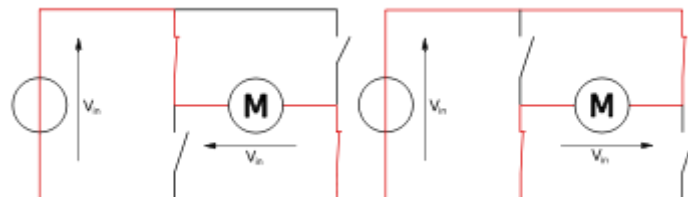
Otros

Puente H L298N



El módulo L298N se utiliza para controlar la velocidad y el sentido de giro de dos motores de corriente continua o un motor paso a paso, mediante dos puentes H dispuestos en el chip.

Un Puente H es un circuito formado por 4 transistores que permite invertir el sentido de la corriente que afecta al motor, cambiando así el sentido de giro del mismo.



El rango de tensiones en el que trabaja este módulo va desde 3V hasta 35V, y una intensidad de hasta 2A. A la hora de alimentarlo hay que tener en cuenta que la

electrónica del módulo consume unos 3V, así que los motores reciben 3V menos que la tensión con la que alimentemos el módulo.

Además el L298N incluye un regulador de tensión que nos permite obtener del módulo una tensión de 5V para alimentar el Arduino. Esto se logra colocando el **jumper regulador**, por lo que en **V lógico** habrá una salida de 5V que se puede conectarse a la placa Arduino. No obstante, este regulador sólo funciona si alimentamos el módulo con una tensión máxima de 12V a través de **V in**.

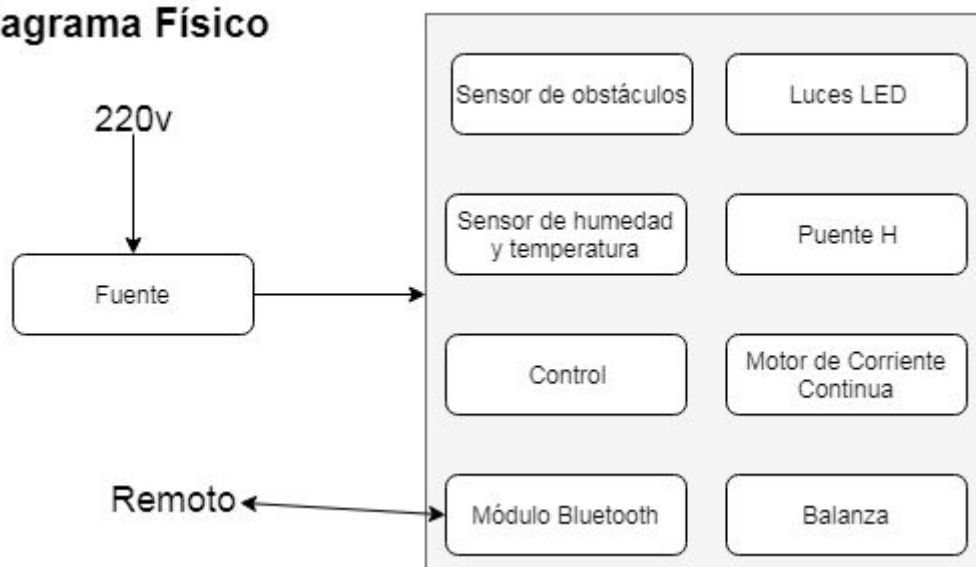
Es **extremadamente importante** notar que cuando el jumper regulador está colocado, **V lógico** se comporta como una salida de corriente y de lo contrario se comporta como una entrada de corriente; Si introducimos una corriente a través de **V lógico** cuando el **jumper regulador** está puesto, se quemará el módulo.

Los pines **IN1** e **IN2** se emplean para controlar el sentido de giro del **Motor A**, mientras que los pines **IN3** e **IN4** el del **Motor B**. Funcionan de forma que si **IN1** está en HIGH e **IN2** en LOW, el **Motor A** gira en un sentido, y si está **IN1** a LOW e **IN2** a HIGH lo hace en el otro. Lo mismo ocurre con los pines **IN3** e **IN4** para el **Motor B**.

Para controlar la velocidad de giro de los motores se deben quitar los **jumpers A y B** y usar los pines **ENA** y **ENB**. Estos deben conectarse a salidas PWM de Arduino de forma que éste enviará un valor entre 0 y 255 que controle la velocidad de giro. Si los jumpers colocados, los motores girarán a la siempre a la misma velocidad.

Las partes físicas del dispositivos se resumen en el siguiente diagrama:

Diagrama Físico



Implementación lógica de la Embebido

La el desarrollo de la lógica se implementó una serie de estados para llevar a cabo el funcionamiento, a estos estados se los tiene como constantes.

- Módulo Bluetooth

Para la implementación del Bluetooth se tiene la Bluetooth.h donde se definen la propia clase bluetooth con sus métodos a utilizar. Uno de los métodos a destacar son:

- isConected: Este sirve para indicar si se pudo establecer la coneccion con el teléfono mobile.
 - leerCantidad: Una vez que recibió el mensaje, este método sirve para identificar la cantidad de ingrediente que se especificó en el mensaje para extraer.
 - enviar: Este método sirve para enviar un mensaje a la aplicación android como por ejemplo la cantidad de ingrediente que queda en el recipiente.
- Leds: Estos tienen la función de informar eventos tales como dispositivo prendido que le pertenece al led rojo, conexión de Bluetooth azul, extrayendo amarillo, disponible verde y blanco no disponible. Para estos el uso de estos leds se utilizó el método digitalWrite y digitalRead.
 - Balanza

Para la implementación de la balanza se usó el método leerBalanza, que este invoca a get_value y así poder obtener la lectura de la balanza.

- Motor continua

En el motor de continua se tiene poder invertir el sentido de giro (puente h), para esto se implementaron los métodos "horario" y "antihorario". Luego "setearVelocidad" que permite incrementando el ancho de pulso de la señal pwm, y se detiene al motor con el método "parar".

- Humedad

Lee el valor de humedad del sensor por medio del metodo "leerHumedad".

El funcionamiento del software está modelado con los siguientes diagramas:

Diagrama Software

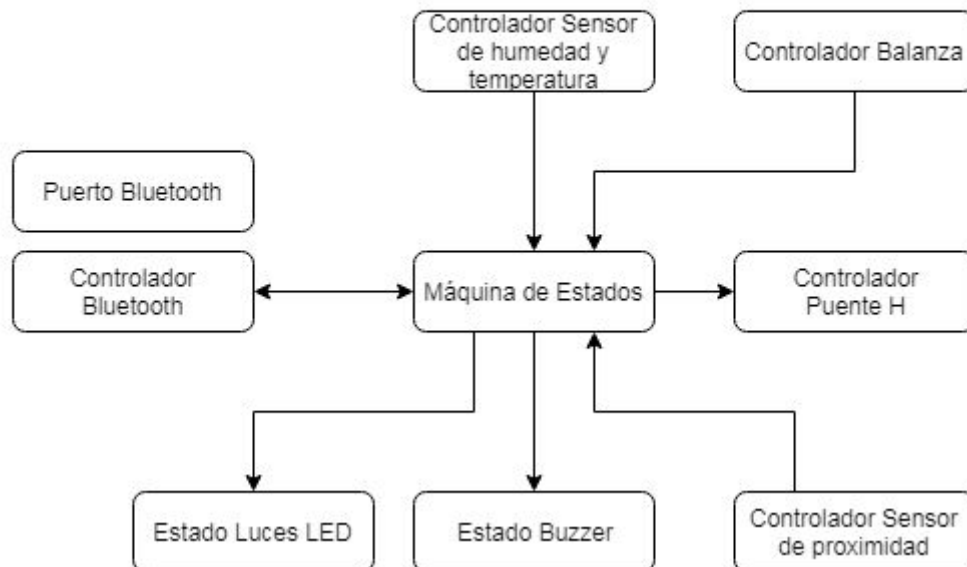
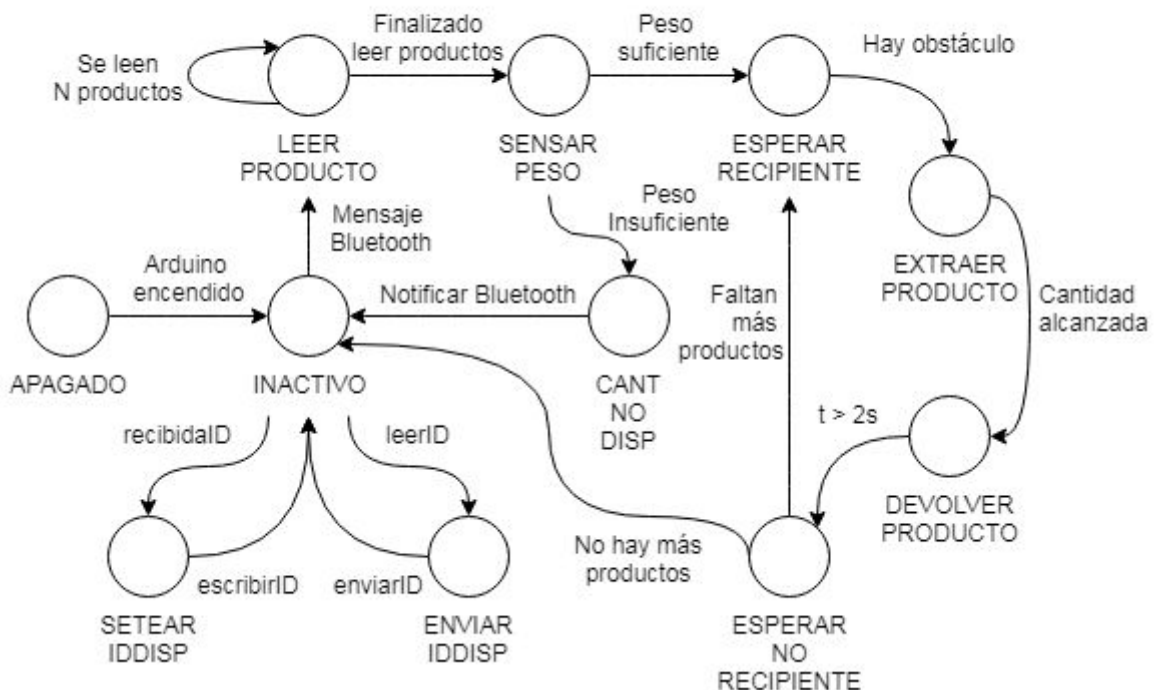


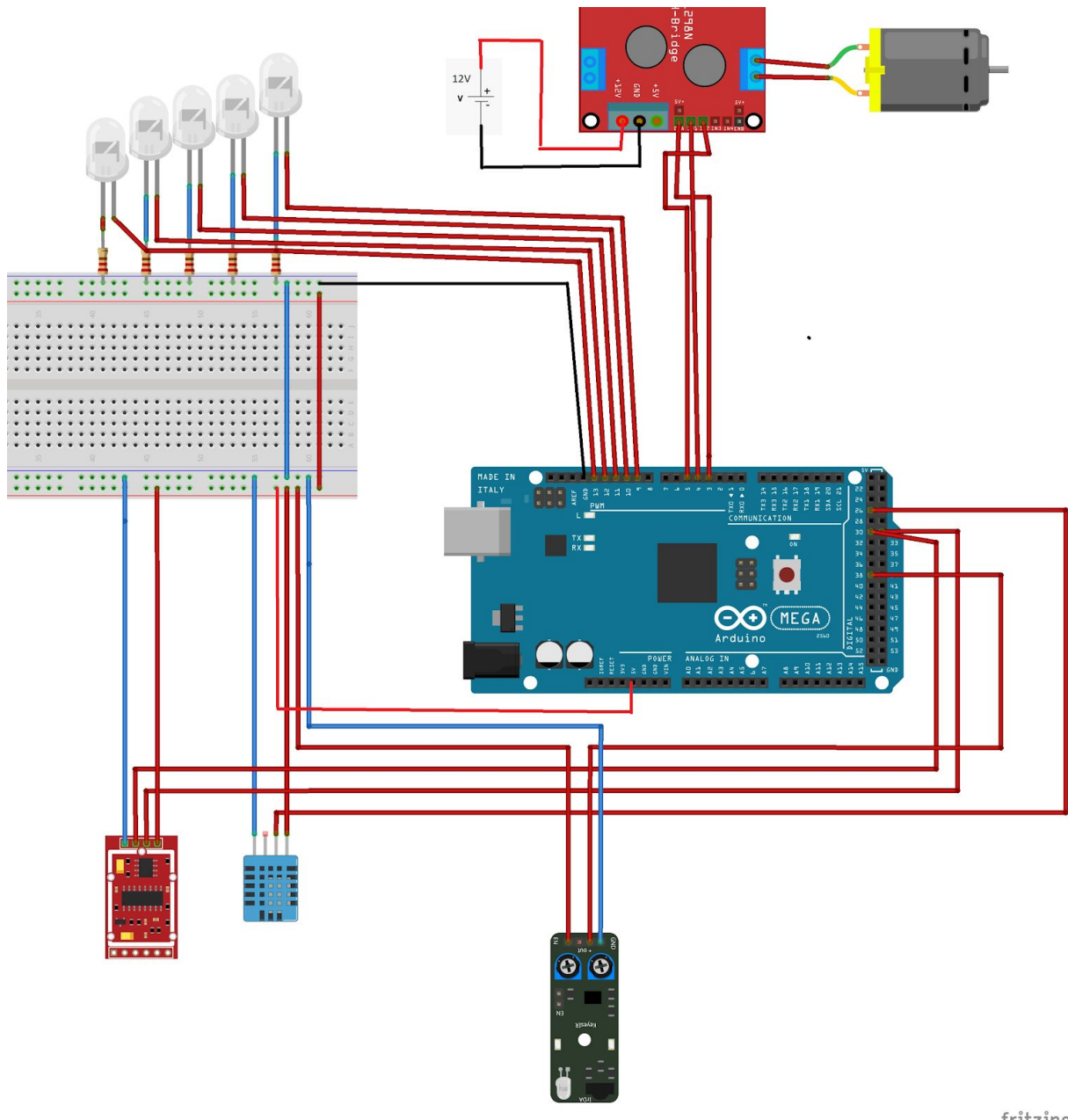
Diagrama de Estados (Sketch)



Los estados que no fueron implementados en el prototipo (por ejemplo, para bajar/subir el brazo) no están modelados en el diagrama.

Circuito

A continuación se muestra el circuito implementado con el embebido y como está conectado con sus componentes.



Como es puede ver se utilizo una fuente aparte de 12v para poder alimentar al puente H. El bluetooth no estaba disponible en el programa es por eso que no está en este diagrama de circuito. El mismo tiene pines positivo y negativo, y dos pines de señal que están conectados al arduino. El pin Rx del módulo del bluetooth está conectado al pin Tx de la placa del Arduino y el pin Tx del módulo de bluetooth está conectado al pin Rx de la placa Arduino.

Limitaciones del Prototipo

El diseño del prototipo actual tiene varias limitaciones con respecto a la idea que se tiene del producto final. En cierta parte por falta de tiempo y presupuesto pero también debido a cuestiones técnicas de las que no se tenía suficiente conocimiento para saber, antes de llevar a cabo el proyecto, si el emprendimiento es viable o no.

Muchas dificultades surgieron durante el desarrollo, especialmente en la construcción del dispositivo en sí. Estos inconvenientes fueron solucionados con métodos no ideales en relación con el concepto que teníamos cuando comenzamos con el proyecto.

A continuación se listan algunas de las limitaciones que se tienen actualmente y que deberían mejorarse para el producto final:

- La extracción de ingredientes no puede lograrse desde varios recipientes automáticamente, sino que se deben intercambiar manualmente de forma que el sinfín pueda extraerlos. Para el producto final se debería implementar un método para direccionar el sinfín hacia cada recipiente de forma automática para realizar la extracción de ingredientes.
- El sensor de humedad y temperatura está ubicado en un lugar donde no es realmente de mucha utilidad. Lo ideal, en este caso, sería que esté ubicado en el mismo recipiente donde se encuentra el ingrediente. Esto implicaría que los ingredientes deban ser colocados en recipientes especiales, cada uno con un sistema embebido propio sincronizados a través del sistema central en la caja, sin embargo ejecutar esta idea implicaría muchos costos y tornarían totalmente inviable al proyecto.
- El tamaño y medida del sinfín y tubo no son los óptimos, idealmente serían lo más corto posible. Además, el diámetro del sinfín impone un límite sobre los ingredientes que pueden ser extraídos por el dispositivo.

Solucionar todos estos defectos detallados suponen también que se deba hacer un rediseño del producto y volver a pensar ciertas cualidades con respecto a lo que se espera lograr para el producto final, ya que las mejoras propuestas posiblemente no sean factibles con el diseño actual.

Aplicación Android

Descripción general

La aplicación es necesaria para cargar ingredientes, recetas y controlar el dispositivo en general.

Funciona para cualquier dispositivo Android con sistema operativo versión **Oreo 8.1** o superior.

Para poder usarla correctamente, el usuario debe proveer a la aplicación con los siguientes permisos:

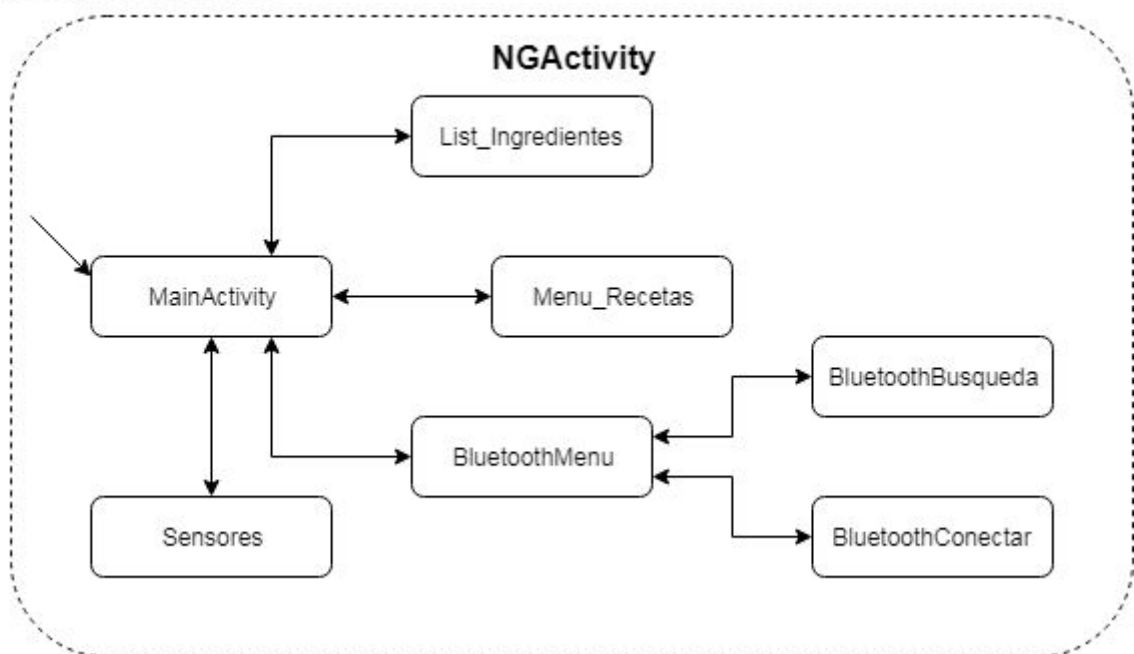
- Acceso al módulo Bluetooth del sistema operativo.
- Lectura y Escritura al medio de almacenamiento externo.
- Acceso a la ubicación y estado del teléfono celular.

Algo a tener en cuenta es que la aplicación prácticamente será inútil si el celular no dispone de un módulo Bluetooth funcional.

Actividades

La aplicación en sí consiste en una serie de **Activities** que hacen de interfaz gráfica para la interacción con el usuario. El usuario puede navegar entre las distintas Activities según el siguiente diagrama:

Diagrama de Activities



Al iniciar la aplicación, la primera ventana en mostrarse es **MainActivity** que actúa como menú principal. A partir de allí, el usuario puede dirigirse a las demás actividades mediante botones desplegados en la pantalla. La actividad **BluetoothMenu** cuenta además con otras 2 actividades, una utilizada para que el usuario pueda buscar y emparejar el dispositivo Bluetooth correspondiente al Appsisistente de Cocina y la otra para efectivamente establecer la conexión. El usuario siempre puede volver hacia la actividad anterior con el botón < del celular.

Todas las actividades heredan de una clase padre denominada **NGActivity**, el propósito de esto es:

- Proveer a las actividades con métodos y atributos comunes a todas (por ejemplo, el método **MostrarToast**)
- Vincular las actividades al **ServicioBluetooth** y suscribirlas a los **broadcasts** emitidos por éste.

Comunicación

Todas las activities pueden recibir y enviar mensajes al sistema embebido a través del servicio Bluetooth, incluso aquellas que no se comunican directamente con el embebido (por ejemplo, **MainActivity**). Esto fue diseñado intencionalmente ya que los servicios en Android se ejecutan en background y no pueden interactuar con la interfaz gráfica, por lo tanto si el servicio necesita mostrar una notificación al usuario, debe hacerlo a través de la activity que esté activa en el momento.

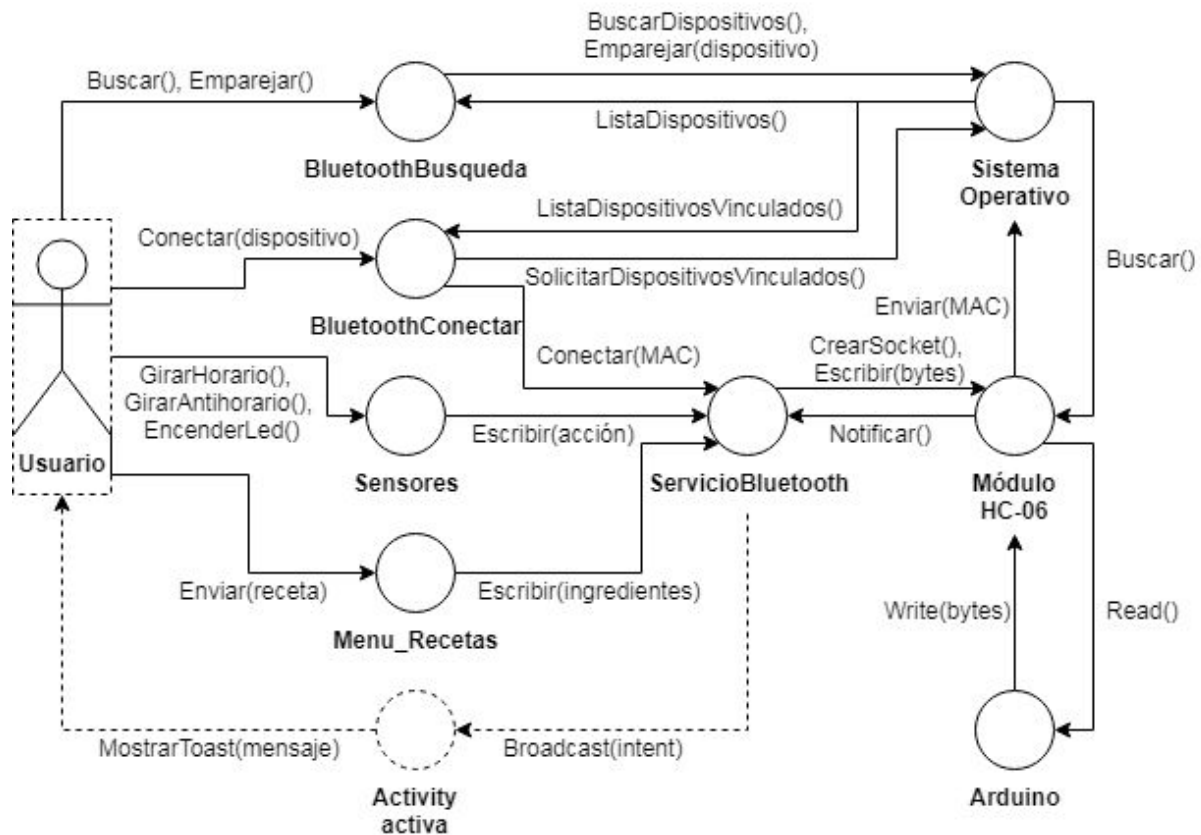
Para poder comunicarse con el módulo HC-06 del embebido, éste debe estar emparejado con el celular; si esto no es así, el usuario puede hacerlo tanto desde la aplicación como desde el sistema operativo Android.

Una vez que se han emparejado, se debe crear un socket Bluetooth entre la aplicación y el HC-06 para permitir el envío y recepción de texto plano. Una vez creado, se confirma el establecimiento de la conexión al usuario y ya puede comenzar a utilizar el resto de las funciones de la aplicación.

El servicio Bluetooth de la aplicación posee 2 hilos secundarios: Uno para crear el socket y establecer la conexión y otro para enviar y recibir mensajes del Arduino. Esto es necesario debido a que ambos procesos son intensivos y bloquean el hilo sobre el que se ejecutan, si se llevaran a cabo en el hilo principal, podrían llegar a congelar a la aplicación.

A continuación se muestra un diagrama que visualiza el flujo de los mensajes entre el usuario, la aplicación, Android y el embebido:

Diagrama de Comunicaciones



Sensores

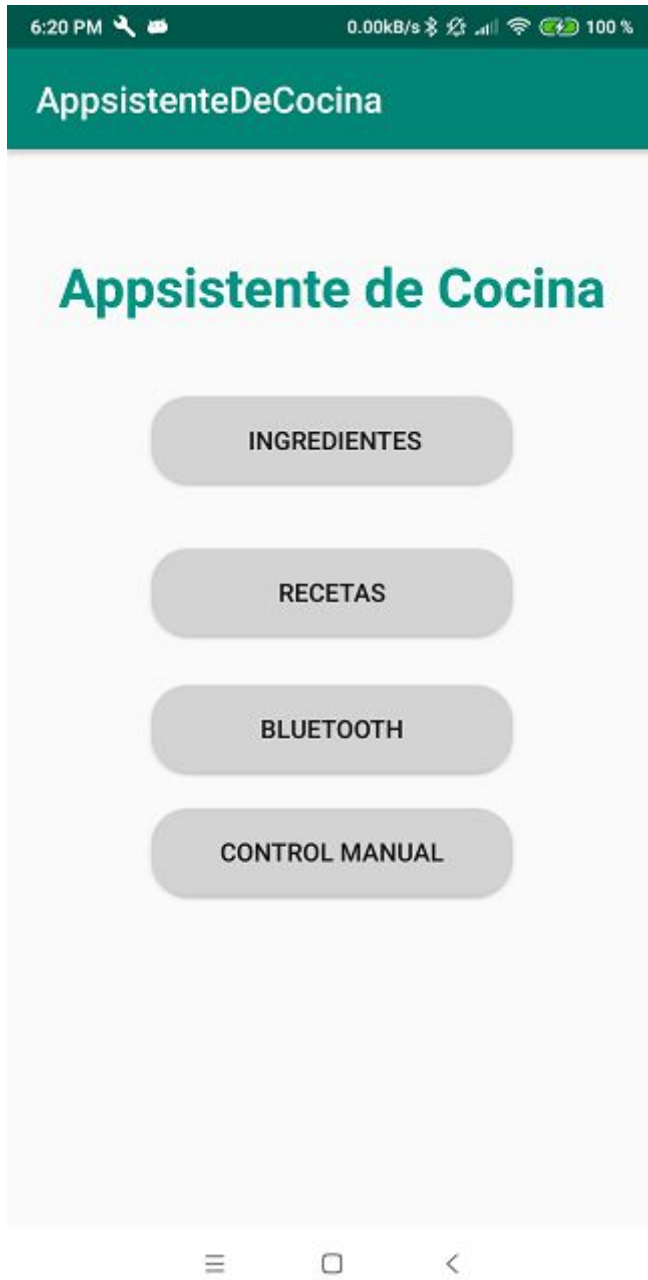
En la activity Sensores se disponen de diferentes controles para manipular al embebido, que se activan mediante la información recibida por los sensores del celular. Los sensores utilizados por la aplicación son los siguientes:

TYPE_GRAVITY: Empleando el acelerómetro, se determina el vector fuerza de gravedad para determinar la orientación del dispositivo. Si el dispositivo está derecho, provoca que el sinfín gire hacia en sentido horario, de lo contrario girará en sentido antihorario.

TYPE_PROXIMITY: Mediante la información del sensor de proximidad ubicado en la parte frontal del celular, se puede pausar o activar el movimiento del sinfín colocando la mano sobre la pantalla.

TYPE_LIGHT: Cuando el sensor de luz detecta poca luz en el ambiente, las luces LED del dispositivo se encenderán en un patrón lineal.

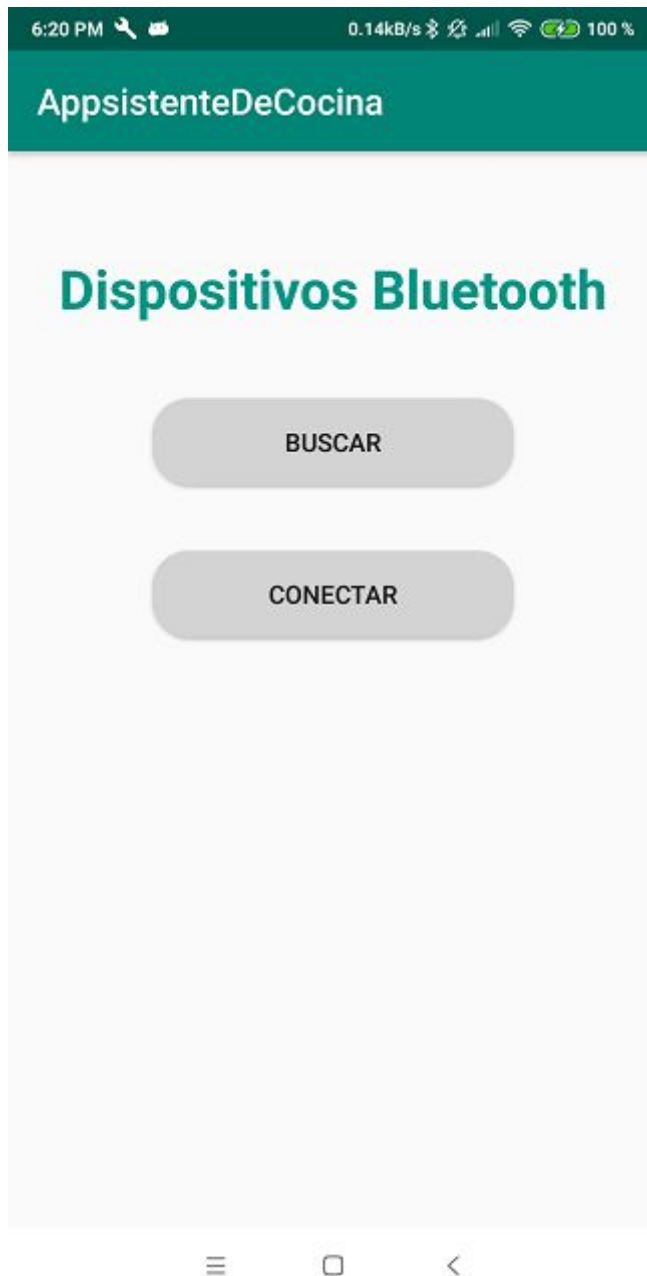
Instrucciones de uso



1 - Una vez ejecutada la aplicación, se nos presentará el menú principal.

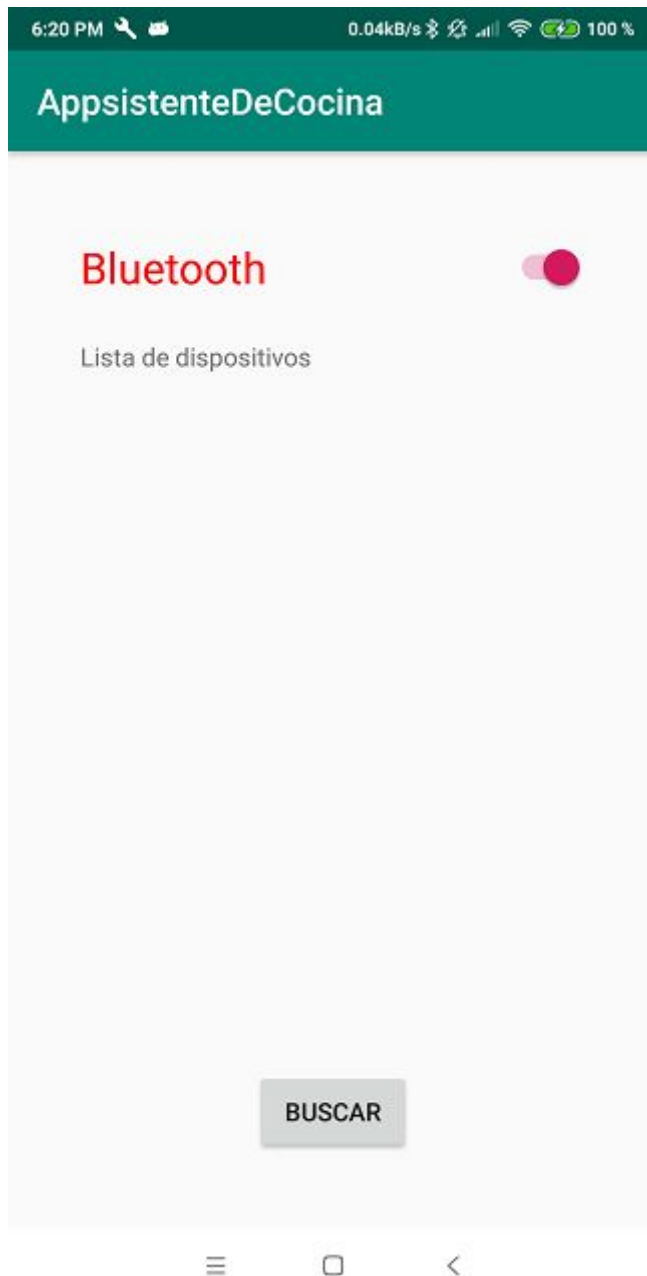
Antes de poder usar alguna de las funciones de la aplicación, primero debemos establecer conexión con el dispositivo.

Presionamos el botón "Bluetooth".



2 - En este menu disponemos de 2 opciones, "Buscar" y "Conectar". Primero debemos buscar la lista de dispositivos Bluetooth cercanos.

Presionamos el botón "Buscar".



3 - Se nos presentará una pantalla con una lista de dispositivos, inicialmente vacía.

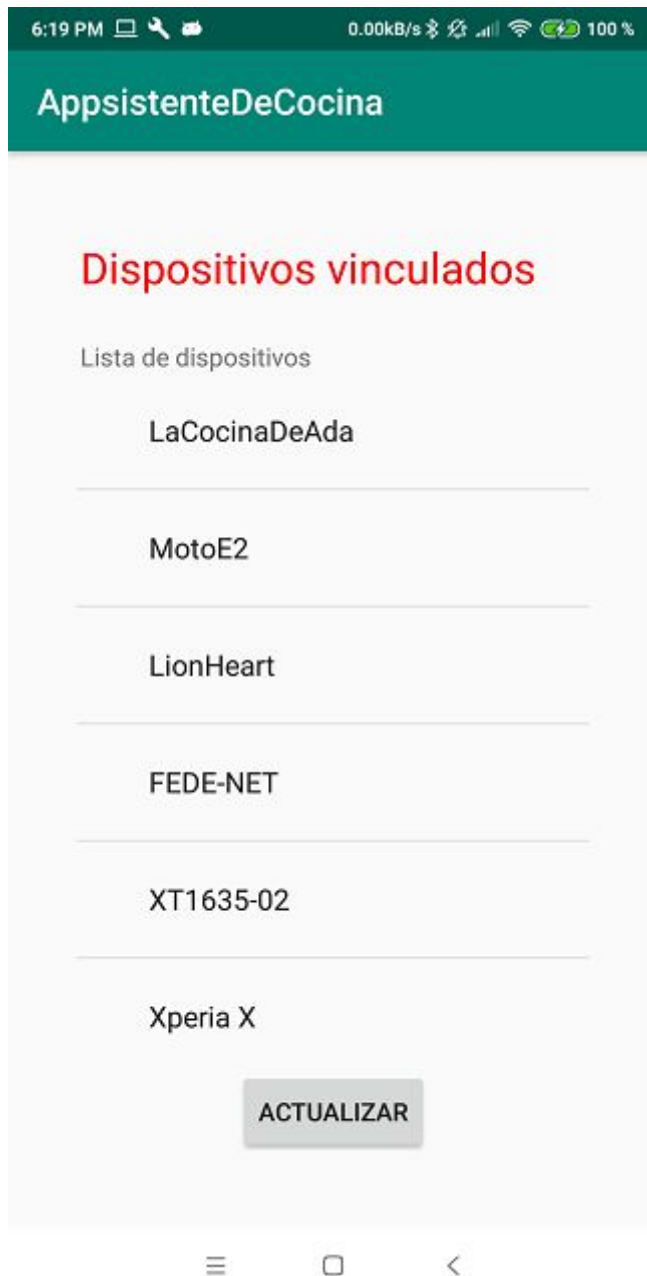
Si no tenemos Bluetooth activado en el celular, primero debemos activarlo: Presionamos el toggle “Bluetooth”.

La aplicación pedirá que le otorgue ciertos permisos en una ventana emergente.

Una vez otorgados los permisos, presionamos el botón “Buscar”. Luego de unos segundos, la lista se llenará con los dispositivos Bluetooth encontrados. Debe buscar el que corresponde al Asistente de Cocina, denominado “LaCocinaDeAda”.

Una vez ubicado, presione en el botón “Emparejar” del Asistente de Cocina. Ahora que el celular está vinculado con el dispositivo, volvemos atrás a la pantalla anterior.

Notar que este procedimiento puede realizarse también a través de la pestaña “Bluetooth” de su celular.



4 - Presionamos el botón “Conectar”, esto nos llevará a la pantalla que se ve en la izquierda.

Aquí aparecerán todos los dispositivos que están emparejados con el celular. Presionamos sobre “LaCocinaDeAda” para establecer la conexión.

Una vez finalizado, volvemos atrás hasta el menú principal y ya podemos comenzar a utilizar la aplicación.



5 - Presionamos en el menú principal, el botón “Ingredientes” que nos llevará a la pantalla que se ve en la izquierda.

En esta pantalla debemos agregar los ingredientes que deseamos usar con el asistente. Presionamos en el botón “+” para abrir una ventana emergente donde podremos ingresar nombre, peso y id del ingrediente.

Repetimos el paso por cada ingrediente y volvemos atrás al menú principal.



6 - Presionamos en el menú principal, el botón “Recetas” que nos llevará a la pantalla que se ve en la izquierda.

Aquí se disponen de recetas precargadas por la aplicación que podemos enviar al asistente para que las prepare.

Presionamos sobre la que deseamos preparar, y si todo marcha bien, el asistente comenzará pronto con la preparación.



OPTATIVO - En el menú principal hay un botón adicional “Sensores” que nos lleva a la pantalla que se ve en esta página.

Aquí se puede interactuar manualmente con ciertas partes del asistente. Si activamos el control manual, podemos:

- Activar el sinfín.
- Detener el sinfín acercando la mano a la pantalla del celular.
- Alternar el sentido de giro del sinfín según la orientación del celular.
- Encender las luces LED del asistente cuando estamos en la oscuridad.

Limitaciones de la aplicación

Actualmente la aplicación no posee muchos déficits con respecto a lo que se espera lograr en el producto final, pero sí hay ciertos puntos a tener en cuenta:

- La interfaz gráfica no parece responder bien en diferentes tamaños de pantalla. Esto se puede visualizar especialmente en aquellos celulares que tienen un tamaño de pantalla inferior a 6 pulgadas.
- La desconexión del socket Bluetooth no intencional (por ejemplo, porque se aleja demasiado el celular del sistema embebido, o se corta el suministro de energía del sistema embebido) no está contemplada y el comportamiento de la aplicación en caso de que ocurra alguna de estas situaciones es indefinido.
- La idea es que la aplicación guarde las recetas e ingredientes en la nube, de forma que un servidor las pueda procesar y enviar información a la aplicación acerca de sugerencias de recetas, notificaciones de ofertas, lugares donde comprar ingredientes etc. Esto no está implementado en el prototipo actual y las recetas se guardan en la memoria secundaria del celular.

El primer ítem puede ser solucionado con suficiente testing sobre diferentes dispositivos para determinar cómo se deben diseñar las vistas de la aplicación. El segundo se puede resolver con suficiente tiempo y no debería haber problemas para manejarlo en el producto final.

Para el tercer ítem hubo una implementación para manejar HTTP Requests hacia un servidor remoto, donde se alojaba una base de datos implementada con Google Spreadsheets. Si bien se logró que haya una comunicación entre aplicación y servidor, el asunto es que esta implementación suponía otros problemas debido a la forma en la que se manejaron las cuestiones de seguridad que impone Google para con sus APIs; Esto provocaba que la aplicación funcionara correctamente en 1 sólo dispositivo (aquel que tenía las credenciales necesarias para llevar a cabo la comunicación cliente - servidor) y hubiera resultado molesto para la presentación de la aplicación. La solución a este problema supone 2 alternativas:

- Desplegar un servidor alojado en alguna parte, programado manualmente por nosotros mismos y con una base de datos implementada también por nosotros mismos para comunicarse con la aplicación. Esto supondría dedicarle más tiempo y recursos al proyecto de los que disponemos.
- Configurar la aplicación y API de Google de acuerdo con el estándar OAuth 2.0 de forma que se pueda acceder desde cualquier dispositivo. Esto requeriría profundizar sobre la forma en la que la aplicación interactúa con la API.

Fuentes

<https://www.hbm.com/en/7325/the-working-principle-of-a-compression-load-cell/>

https://naylampmechatronics.com/blog/25_tutorial-trasmisor-de-celda-de-carga-hx711-ba.html

<https://www.instructables.com/id/Measuring-Humidity-Using-Sensor-DHT11/>

https://wiki.eprolabs.com/index.php?title=IR_Obstacle_Sensor

<https://www.prometec.net/l298n/>

https://es.wikipedia.org/wiki/Motor_de_corriente_continua

<https://www.ingmecafenix.com/electronica/el-buzzer/>

<https://www.areatecnologia.com/electronica/como-es-un-led.html>

<https://oauth.net/2/>

<https://developer.android.com/docs>