

## 0.1 Hardware

### 0.1.1 Diagrama de bloques (Hardware)

A continuación se muestra el diagrama en bloques de hardware y los delimitadores de las distintas secciones, siendo estas:

- Potencia
- Cargador
- Sensado

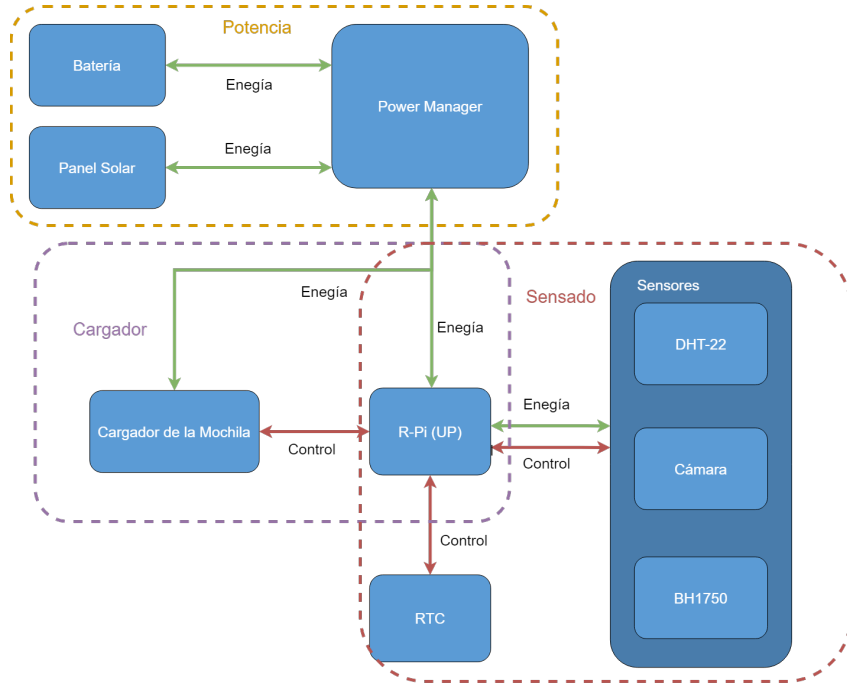


Figura 0.1.1.1: Diagrama en bloques del sistema de hardware.

La etapa de potencia es el conjunto de elementos necesarios para proveer de energía a toda la electrónica del proyecto, es decir a aquellos componentes y elementos que se encuentran en el nido.

El cargador es la etapa que se encarga, como su nombre indica realizar al carga inalámbrica de la UBM.

Finalmente la de sensado, es aquella que se encarga de la medición de las variables físicas y de su correcto almacenamiento, teniendo en cuenta que esto implica un conocimiento preciso de la hora.

### 0.1.2 Descripción detallada de cada bloque

#### 0.1.2.1 Potencia

Para la unidad de potencia se utilizarán paneles solares y una batería de gel de ciclo profundo conectados a la placa DFR0580, la cual se ocupa de cargar la batería. De esta se obtienen 4 salidas de tensión:

- Dos salidas de 5 V y 2.5 A [USB].
- Una salida de 5 V y 5 A.
- Una salida de 12 V y 8 A.

Con estas salidas se alimentarán todos los módulos, a excepción del oscilador de potencia, ya que esta necesita una etapa DC-DC para así la cual eleva la tensión. Para ello se emplea una fuente switching de topología Boost.

### 0.1.2.2 Cargador

El bloque del cargador de la UBM consiste en un receptor y un transmisor de potencia. La transmisión inalámbrica consta, por el lado del transmisor, de un oscilador HM-TRPW-RS232 de 915  $MHz$  el cual está comandado por la R-Pi y se comunica mediante UART; como así también de un amplificador de potencia de 3W máximos alimentado por una etapa DC-DC X16009 de 12 V a 15 V.

Por el lado del receptor, se encuentra el integrado P1110B, el cual almacena energía temporalmente en un capacitor para realizar posteriormente la carga de la UBM.

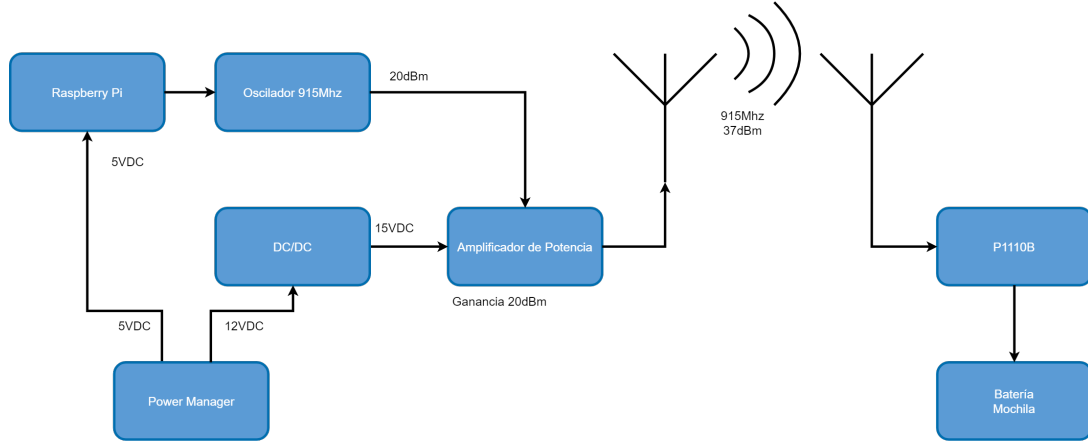


Figura 0.1.2.2.1: Diagrama en bloques cargador.

Para la transmisión de la señal de 915  $MHz$  se utilizará cable del tipo  $RG - 213$  el cual posee bajas pérdidas de  $23.054 \frac{dB}{100m}$ .

### 0.1.2.3 Sensado

En esta sección, además del conexionado de los sensores, se hace detalle en los pines de la R-Pi que serán utilizados.

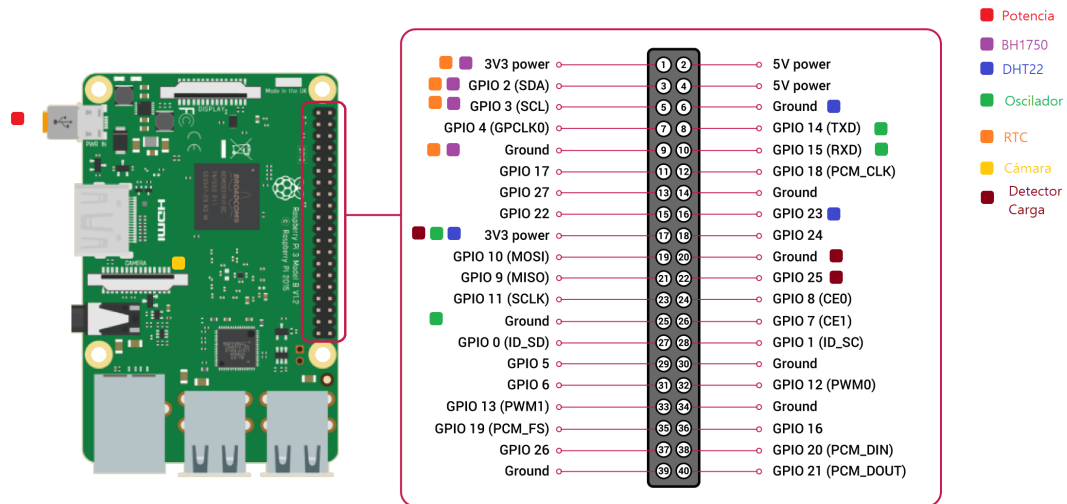


Figura 0.1.2.3.1: Conexionado Raspberry Pi.

El sensor de humedad y temperatura (DHT-22) se comunica de manera serial a través de un pin de GPIO. Para que esto sea posible, el sensor necesita alimentación de 3.3 V, tierra y el pin de GPIO por donde se realiza la comunicación, teniendo en cuenta que es necesario una resistencia de pull-up entre la línea de datos y 3.3 V.

El oscilador necesita de una señal de alimentación, tierra y su comunicación utilizará el protocolo UART. Por otro lado, la cámara simplemente se conecta al zócalo destinado para este propósito en la placa. Para el conexionado con el sensor de luminosidad y RTC se utilizará el protocolo de comunicación  $I^2C$  en modo multi-slave.

Finalmente, se cuenta con un pin de GPIO el cual indicará el momento de prendido del cargador.

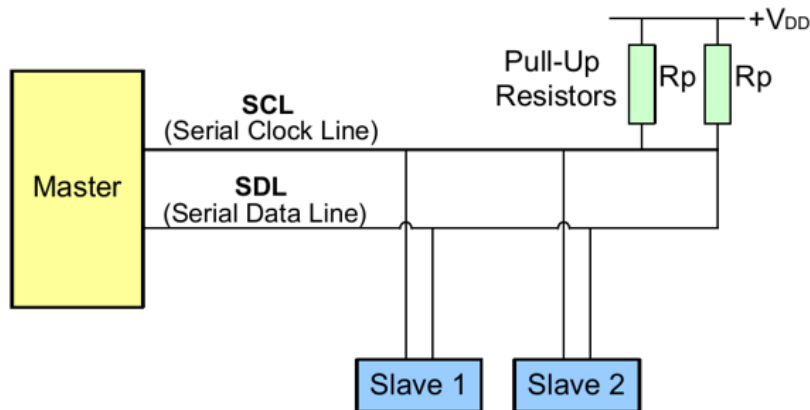


Figura 0.1.2.3.2: Conexionado  $I^2C$  Multi-slave.

### 0.1.3 Detalles de selección y cálculo de los elementos circuitales de cada bloque

#### 0.1.3.1 Cálculo potencias

Se asumen las siguientes variables para los cálculos de potencias, obtenidas de estadísticas propias de la zona donde se realiza el estudio **Referencia a de donde sacamos el dato me imagino en el final de la tesis.**

1. Duración noche = 8,5 horas.
2. Duración día = 15,5 horas.
3. Horas de sol efectivas = 8 horas.

Para el caso de la R-Pi, se tiene que su consumo mínimo normal es de  $P_{rpi_{min}} = 2\text{ W}$ . Sin embargo, para reducir este consumo, se desactivan los puertos de ethernet y HDMI ya que no se utilizará un entorno gráfico. Esto permite reducir aún más el consumo mínimo. Se estima **como lo estimamos?** que la R-Pi consumirá alrededor de  $P_{rpi_{est}} = 2.7\text{ W}$  en su funcionamiento normal. Luego, se tiene que

$$E_{rpi_{est}} = P_{rpi_{est}} \cdot 24\text{ hs} = 233.28\text{ kJ} \quad (1)$$

Los valores de potencia fueron obtenidos de **referencia a de donde sacamos el dato me imagino en el final de la tesis.**

Para el caso del cargador de la UBM, se quiere lograr transmitir  $2\text{ W}$  de potencia a través de la antena transmisora. Teniendo en cuenta la eficiencia de la antena transmisora del 70 % (TBC) y las pérdidas del cable RG-213 utilizado para transportar la señal, se tiene que la potencia necesaria en el amplificador de potencia es de  $2.85\text{ W}$ . Multiplicando por la duración de una noche se tiene que

$$E_{ant} = P_{ant} \cdot 8.5\text{ hs} = 130.89\text{ kJ} \quad (2)$$

Finalmente, el consumo energético por día es de

$$E_{sist} = E_{rpi_{est}} + E_{ant} = 364.17\text{ kJ} \quad (3)$$

Por lo que, por un lado, teniendo en cuenta que la batería, de  $V_{bat} = 12\text{ V}$ , debe tener una capacidad de almacenamiento equivalente a  $T_{reserva} = 4$  días sin recarga **estadística del lugar pls** y utilizando un coeficiente de seguridad de  $\gamma_{bat} = 1.5$ , se obtiene

$$Capacidad_{bat} = \frac{E_{sist} \cdot T_{reserva} \cdot 1000}{V_{bat} \cdot 3600} \cdot \gamma_{bat} = 50.58\text{ Ah} \quad (4)$$

Por otro lado, para los cálculos del panel solar, teniendo en cuenta que se quiere que en un día de sol promedio se logre abastecer al sistema de su consumo energético normal diario y recargar un  $\rho = 0.1$  de la capacidad total de la batería, y además teniendo en cuenta un coeficiente de seguridad de  $\gamma_{panel} = 1.75$ , se tiene que

$$Pot_{panel} = \left( 364.17 \text{ kJ} + \frac{Capacidad_{bat} \cdot V_{bat} \cdot 3600 \cdot \rho}{1000} \right) \cdot \gamma_{panel} \cdot \frac{1000}{60 \cdot 60 \cdot 8 \text{ hs}} = 35,41 \text{ W} \quad (5)$$

#### 0.1.3.2 Cálculo resistencias

Tanto para el Bus de  $I^2C$  como para el DHT22, se calcularon las resistencias de pull-up de la siguiente manera:

$$R_p = \frac{V_{dd}}{I_{R_p}} = \frac{3.3 \text{ V}}{1 \text{ mA}} = 3.3 \text{ k}\Omega \quad (6)$$

Acá van los cálculos de paneles, batería, de la boost si es que la hacemos o si usamos un integrado.

#### 0.1.4 Plan de pruebas de cada modulo

### 0.2 Software

#### 0.2.1 Esquema general del proyecto

Mi idea para esta parte era explicar que tecnologías íbamos a usar y eso

#### 0.2.2 Diagrama de estados y flujogramas

En este pasaje se muestran los diagramas de estados del sistema. Es así que se puede notar que el conjunto cuenta con 4 estados distintivos, siendo estos los llamados *Initial*, *Idle*, *Charging* y *Communicating*.

Vale la pena mencionar que en cualquiera de los estados, a excepción de *Initial*, el sistema esta realizando mediciones del ambiente mientras cuenta también con la posibilidad de comunicarse por Bluetooth con la electrónica de la mochila. Esto se cumple excluyendo el caso en que el nivel de carga de la UBM no sea suficiente.

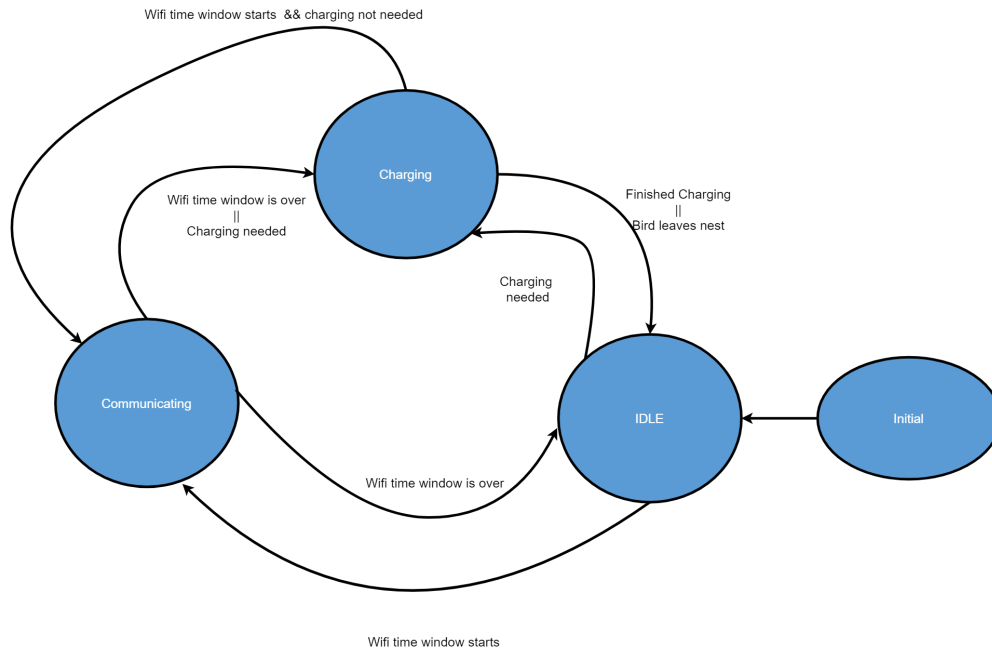


Figura 0.2.2.1: Diagrama de estados.

En el estado *Initial* es aquel donde se inicializan todos los drivers, estructuras de datos y configuraciones que se encuentren por defecto. No se vuelve a este estado una vez que este haya sido abandonado.

Initialize  
State

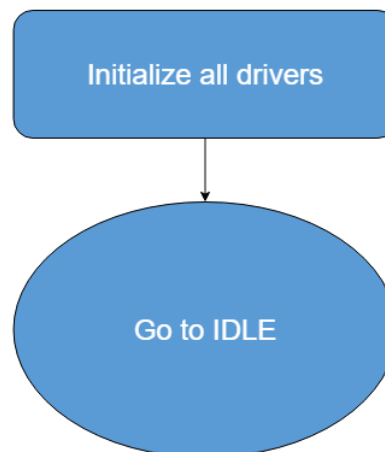


Figura 0.2.2.2: Diagrama de flujo: Estado Initial.

Luego se encuentra el estado *Idle*. En este se sensan las variables físicas con el periodo de muestreo acorde a cada especificado en **PONER LUGAR DONDE ESTE ESPECIFICADO ESTO**. Luego se fija si es momento de prender el hotspot wifi. En caso afirmativo se irá al estado *Communicating*. En caso contrario se analiza si hay que cargar la batería. Si se debe realizar dicha acción se irá a estado *Charging*, mientras que en el caso opuesto, si hay transmisión Bluetooth, actúa acorde y comienza el ciclo nuevamente.

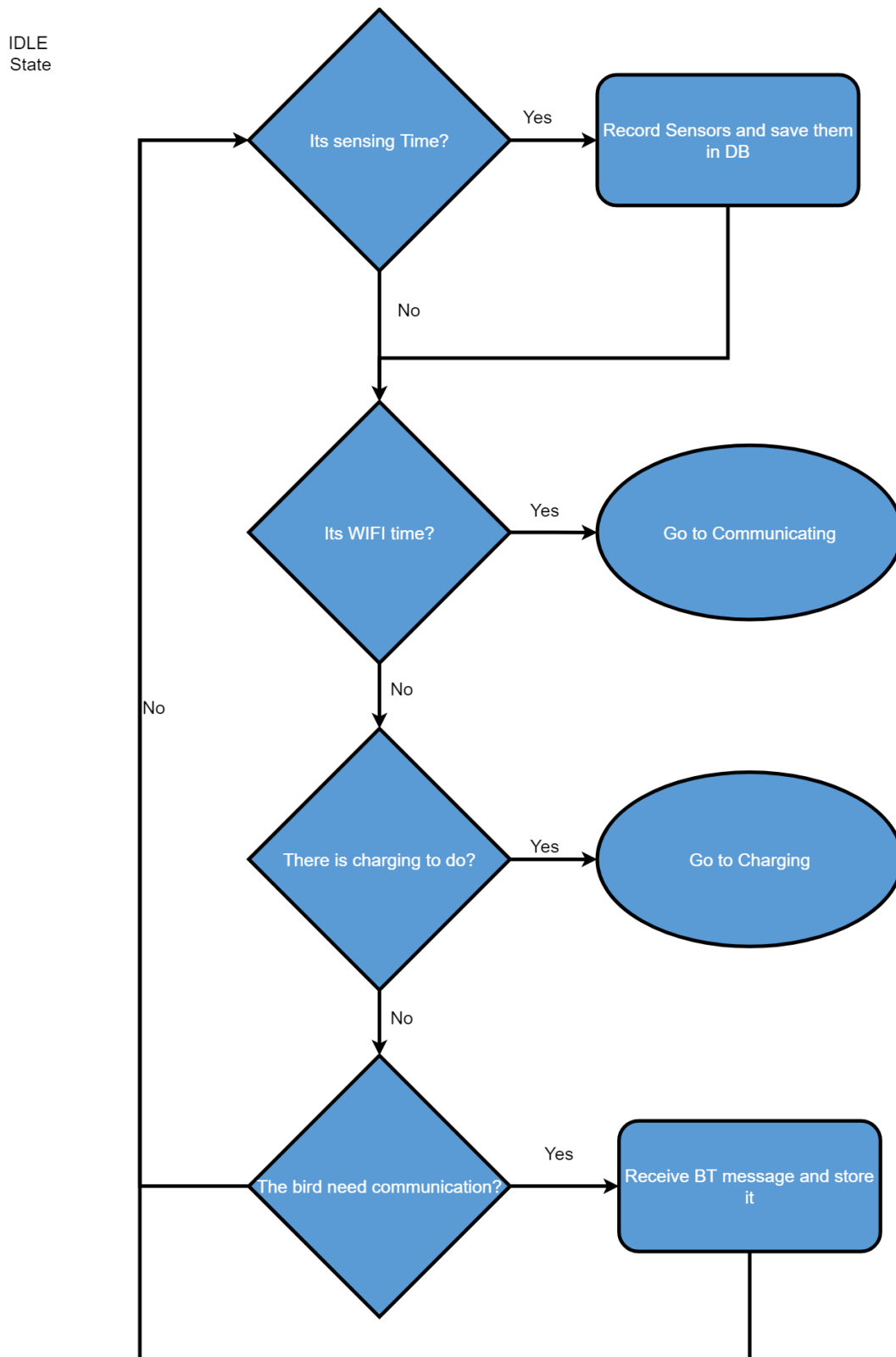


Figura 0.2.2.3: Diagrama de flujo: Estado *Iddle*.

La función principal del estado *Charging* es la de cargar la UBM. Lo primero se hace en este estado es habilitar el cargador, luego, en caso de que haya un mensaje Bluetooth, se comienza la comunicación. Posteriormente se encarga del sensado, se verifica si la carga se terminó, para que cuando esto ocurra se desactive el cargador. Dependiendo si es momento de habilitar el hotspot o no se irá al estado *Communicating* o *Idle*.

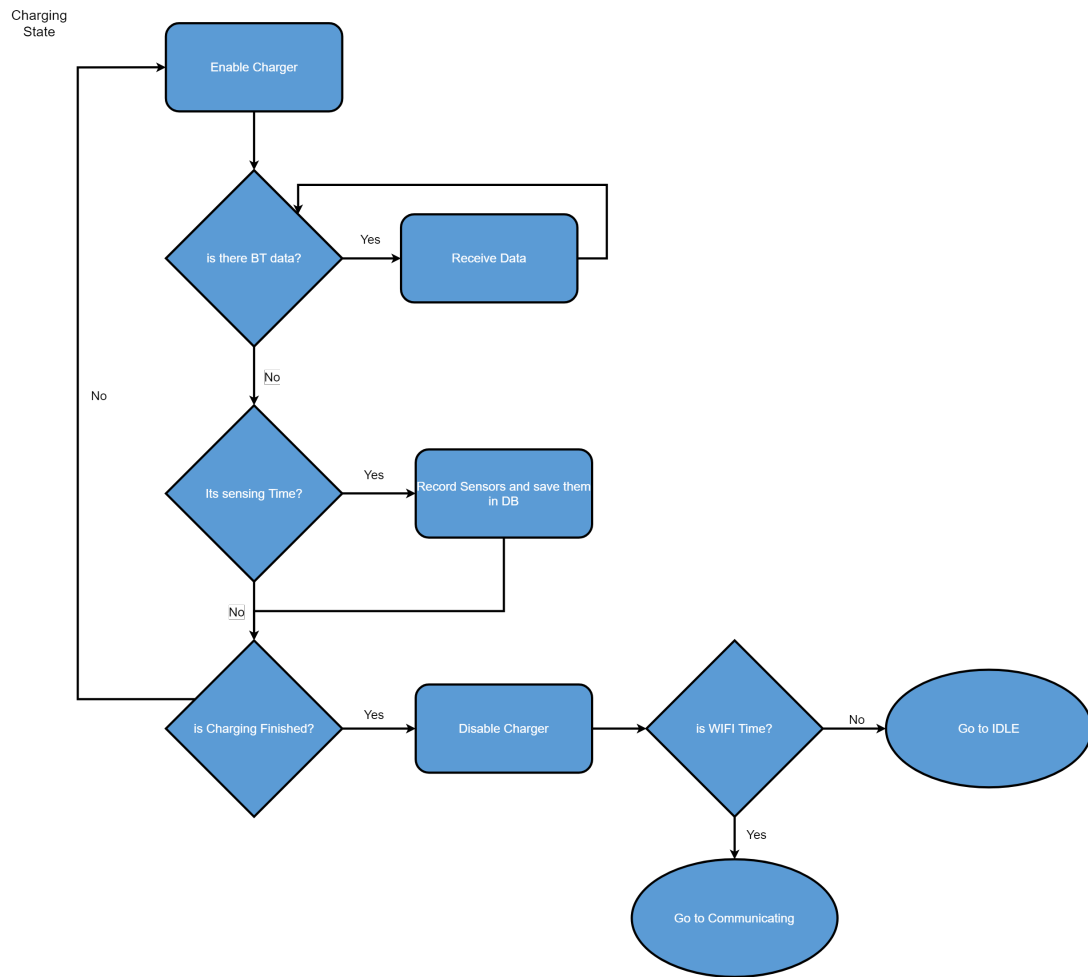


Figura 0.2.2.4: Diagrama de flujo: Estado *Charging*.

Finalmente en el estado *Communicating* es aquel en el que se habilita el hotspot y se levanta el servidor de Node-Red. Aquí se realiza la comunicación entre el nido y una computadora en al base del árbol. Esto se mantendrá hasta que haya pasado el tiempo especificado en **No se deberíamos decirlo en algún lado, probablemente hardware**. Además, se continua sensando las variables físicas.

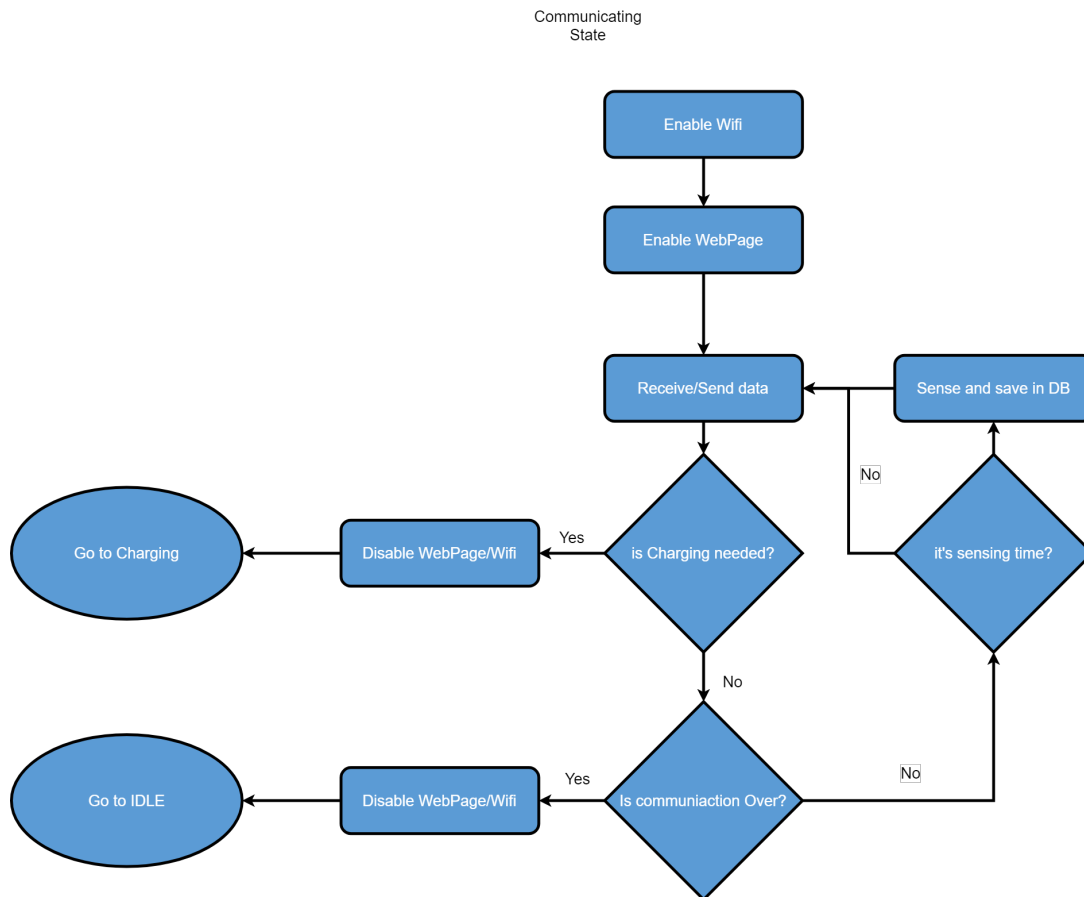


Figura 0.2.2.5: Diagrama de flujo: Estado *Communicating*.

### 0.2.3 Análisis de complejidad

### 0.2.4 Descripción de subrutinas

### 0.2.5 Listado de elementos del código

### 0.2.6 Plan de prueba de módulos y de depuración de Software

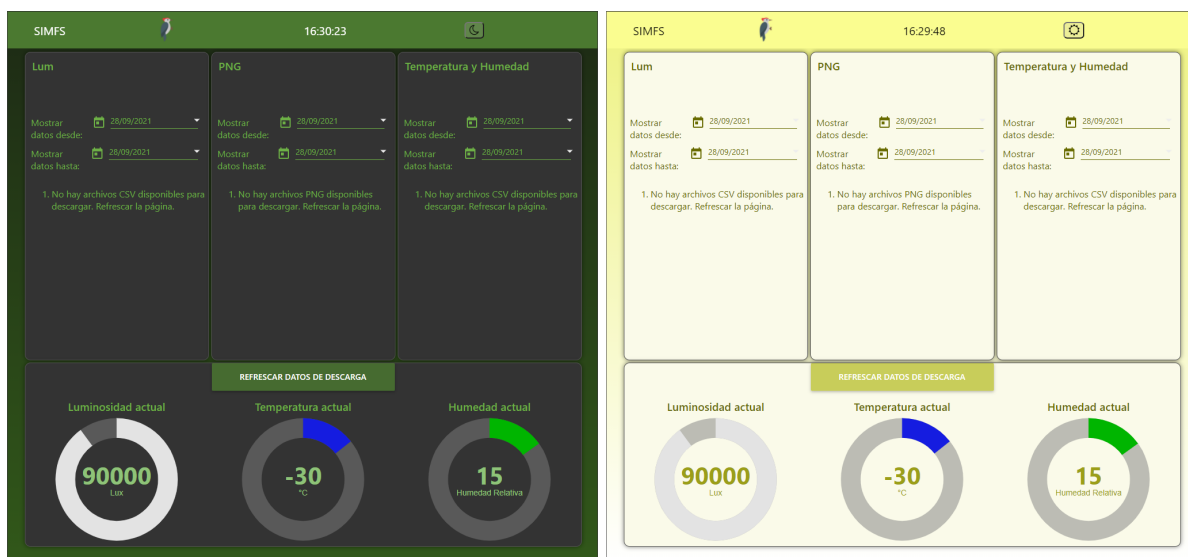
### 0.2.7 Interfaz gráfica de usuario)

Para el desarrollo de la GUI, se valió del uso de **Node-RED**, software que permite el desarrollo de servicios online. Node-Red, mediante un sistema de nodos y sumado a la programación en JavaScript, permite generar una página web, la cual cumple la función de interfaz con el usuario.

Dado que el servidor estará corriendo en la R-Pi, al conectarse a dicha red, se podrá acceder a la página mencionada, donde se brindaran links de descarga de los datos propios del nido. Además, el mismo sistema provee la posibilidad de agregar un paso de autenticación al ingresar al servidor mediante el uso de un usuario y contraseña.

Es así que una primera versión de la página es como la que se muestra en las Figuras (0.2.7.1a) y (0.2.7.1b), donde se proveen links de descarga para los datos almacenados, algunas de las últimas mediciones realizadas y la posibilidad de seleccionar entre que fechas se desean visualizar las mediciones.





(a) Interfaz con usuario versión oscura.

(b) Interfaz con usuario versión clara.

Figura 0.2.7.1: Página del servidor a la cual accede el usuario.

Finalmente, cabe destacar que el sistema de edición de la GUI puede ser accedido a través de un navegador de la misma manera que la interfaz del usuario. Para evitar que el usuario (o alguna persona no deseada) acceda a esta sección, se emplea un link privado, el cual es confidencial y no será brindado al usuario. También se vale del mismo sistema de autenticación que emplea la interfaz gráfica, creando un usuario único de administrador.

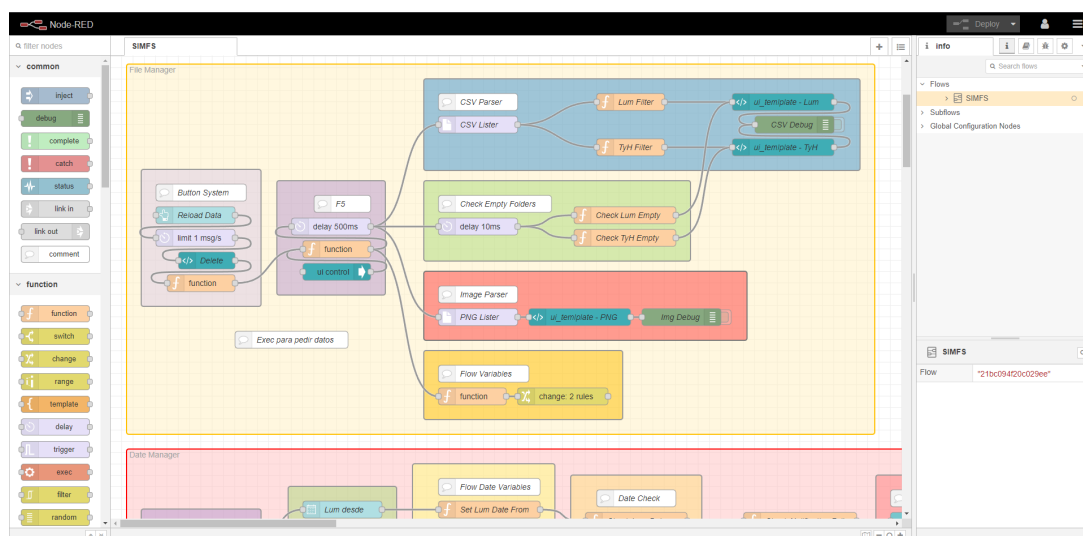


Figura 0.2.7.2: Flujo de nodos del servidor.