

## Resumen de Funciones

El propósito de este documento es recopilar las funciones del sistema e indicar si requiere de valores y la salida que tiene la función. Las funciones estarán agrupadas por Clases y Hardware.

Clases necesaria para cargar en hardware nodo receptor (Arduino Mega 2560)

### Clase Main

Nombre Función	Setup
Input	-
Output	-
Propósito	Configurar valores requeridos por hardware de arduino, como también iniciar la memoria sd e iniciar valores de la red Ethernet.

Nombre Función	Loop
Input	-
Output	-
Propósito	Mantener la ejecución del programa cargado en memoria de Arduino infinitamente.

Nombre Función	Confirmar
Input	Número que indica la cantidad de valores a recepcionar
Output	True o False
Propósito	Confirmar la cantidad de valores que viene en el mensaje proveniente del nodo sensor.

Nombre Función	control_sensor
Input	-
Output	-
Propósito	Poseer el control de un ciclo de trabajo sobre la adquisición de datos.

Nombre Función	<b>freeRam</b>
Input	-
Output	Valor que indica la memoria libre en bytes
Propósito	Obtener la memoria libre que existe en la memoria SRAM

## Clase Almacenar\_sd

Nombre Función	<b>Escribir</b>
Input	Data: valores a almacenar Nombre: Clave sensor
Output	True: Si escribe False: Si no escribe
Propósito	Función utilizada para escribir en la memoria sd presente en el nodo receptor.

Nombre Función	<b>Error_Sd</b>
Input	Valor clave indica tipo de error
Output	-
Propósito	Activa una secuencia de led para indicar error en nodo receptor

Nombre Función	<b>Init_Sd</b>
Input	-
Output	-
Propósito	Iniciar pin para utilizar memoria sd

## Clase Comando

Nombre Función	<b>modo_comando</b>
<b>Input</b>	-
<b>Output</b>	-
<b>Propósito</b>	Permitir entrar en la configuración de los módulos de comunicación (modo AT)

Nombre Función	<b>configurar_parametro</b>
<b>Input</b>	DL: Valor dirección DH: Dirección MAC equipo destino
<b>Output</b>	-
<b>Propósito</b>	Configurar la dirección DL y DH del módulo de comunicación para dirigir un mensaje.

Nombre Función	<b>salir_modos_comando</b>
<b>Input</b>	-
<b>Output</b>	-
<b>Propósito</b>	Salir de modo AT de manera correcta para que la configuración se guarde.

Nombre Función	<b>reset</b>
<b>Input</b>	-
<b>Output</b>	-
<b>Propósito</b>	Restaurar la configuración de los módulos de comunicación.
Nombre Función	<b>limpiar</b>
<b>Input</b>	-
<b>Output</b>	-
<b>Propósito</b>	Limpiar los buffers de recolección del módulo de comunicación.

<b>Nombre Función</b>	Leer_data
<b>Input</b>	-
<b>Output</b>	-
<b>Propósito</b>	Obtiene los bytes que están en el buffer del modulo de comunicación y almacena en una variable tipo string.

<b>Nombre Función</b>	get_data
<b>Input</b>	-
<b>Output</b>	String data
<b>Propósito</b>	Devuelve lo almacenado en la variable data

<b>Nombre Función</b>	escribir_data
<b>Input</b>	-
<b>Output</b>	-
<b>Propósito</b>	Escribe la información de respuesta del modulos a los comandos AT

<b>Nombre Función</b>	escribir_data
<b>Input</b>	-
<b>Output</b>	-
<b>Propósito</b>	Escribe la información de respuesta del modulos a los comandos AT

<b>Nombre Función</b>	explorar_red
<b>Input</b>	-
<b>Output</b>	-
<b>Propósito</b>	Descubrir los módulos de comunicación que están formando la red inalámbrica.

<b>Nombre Función</b>	Get_direccion
<b>Input</b>	Numero: Indica el número de la dirección
<b>Output</b>	String: La dirección mac de un módulos de comunicación
<b>Propósito</b>	Escribe la información de respuesta del módulos a los comandos AT

### Clase Comunicación

<b>Nombre Función</b>	enviar_mensaje
<b>Input</b>	Numero: UART a utilizar
<b>Output</b>	True: Envía mensaje False: No envía mensaje
<b>Propósito</b>	Envía un mensaje almacenado en memoria por un UART específico

<b>Nombre Función</b>	Recibir_mensaje
<b>Input</b>	Numero: UART a utilizar
<b>Output</b>	True: recibe mensaje False: No recibe mensaje
<b>Propósito</b>	Recibe un mensaje proveniente de algún UART

<b>Nombre Función</b>	delete_mensaje
<b>Input</b>	-
<b>Output</b>	-
<b>Propósito</b>	Borrar el mensaje almacenado si este ya cumplió su utilidad.

<b>Nombre Función</b>	Convertir
<b>Input</b>	String numero
<b>Output</b>	Int numero

<b>Propósito</b>	Convertir un string en un numero para mejor manipulación y ahorro de memoria
------------------	--

<b>Nombre Función</b>	get_mensaje
<b>Input</b>	-
<b>Output</b>	String: Entrega un mensaje almacenado
<b>Propósito</b>	Obtener un mensaje almacenado en memoria

<b>Nombre Función</b>	limpiar
<b>Input</b>	-
<b>Output</b>	-
<b>Propósito</b>	Limpiar el buffer de los puerto UART utilizados.

<b>Nombre Función</b>	set_mensaje
<b>Input</b>	String: mensaje a guardar
<b>Output</b>	-
<b>Propósito</b>	Reestablecer el mensaje para ser enviado.

### Clase Red

<b>Nombre Función</b>	Set_mac
<b>Input</b>	Byte mac
<b>Output</b>	-
<b>Propósito</b>	Almacenar mac a utilizar

<b>Nombre Función</b>	set_ip_local
<b>Input</b>	Byte ip
<b>Output</b>	-
<b>Propósito</b>	Almacenar ip local a utilizar

<b>Nombre Función</b>	set_ip_server
<b>Input</b>	Byte ip
<b>Output</b>	-
<b>Propósito</b>	Almacenar ip servidor a utilizar.

<b>Nombre Función</b>	set_mask
<b>Input</b>	Byte mask
<b>Output</b>	-
<b>Propósito</b>	Almacenar mascara a utilizar

<b>Nombre Función</b>	set_port_server
<b>Input</b>	Int: número de puerto
<b>Output</b>	-
<b>Propósito</b>	Almacenar el puerto del servidor al cual conectarse

<b>Nombre Función</b>	iniciar_red
<b>Input</b>	-
<b>Output</b>	True: Inicio de red correcto False: Inicio de red fallido
<b>Propósito</b>	Iniciar la red para conectarse al servidor.

<b>Nombre Función</b>	leer_mensaje
<b>Input</b>	-
<b>Output</b>	True: Lectura correcta False: Lectura incorrecta
<b>Propósito</b>	Lee mensaje proveniente del servidor y los almacena

<b>Nombre Función</b>	enviar_mensaje
<b>Input</b>	-
<b>Output</b>	-
<b>Propósito</b>	Enviar un mensaje almacenado en memoria hacia el servidor

<b>Nombre Función</b>	borrar_mensaje
<b>Input</b>	-
<b>Output</b>	-
<b>Propósito</b>	Borrar el mensaje almacenado si este ya cumplió su utilidad.

<b>Nombre Función</b>	get_mensaje
<b>Input</b>	-
<b>Output</b>	String: Entrega un mensaje almacenado
<b>Propósito</b>	Obtener un mensaje almacenado en memoria

<b>Nombre Función</b>	set_mensaje
<b>Input</b>	String: mensaje a guardar
<b>Output</b>	-
<b>Propósito</b>	Reestablecer el mensaje para ser enviado.

<b>Nombre Función</b>	server_disponible
<b>Input</b>	-
<b>Output</b>	True: Disponible False: No disponible
<b>Propósito</b>	Revisa si el servidor al cual conectarse se encuentra disponible.



<b>Nombre Función</b>	Procedimiento
<b>Input</b>	String:mensaje a enviar
<b>Output</b>	True: Realiza ejecución False: No realiza ejecución.
<b>Propósito</b>	Función que realiza los pasos necesarios para enviar correctamente al servidor los datos.

<b>Nombre Función</b>	enviar_por_red
<b>Input</b>	String:mensaje a enviar
<b>Output</b>	-
<b>Propósito</b>	Enviar por la red hacia el servidor la información necesaria.  Formato del mensaje: Fecha;id_tipo_sensor;Mac;id_ubicacion;data

<b>Nombre Función</b>	error_red
<b>Input</b>	Int: Clave para indicar error
<b>Output</b>	-
<b>Propósito</b>	Indica error por una secuencia de led

Para nodo sensor (Arduino Uno) tenemos las siguientes funciones:

### Clase Principal

<b>Nombre Función</b>	<b>Setup</b>
<b>Input</b>	-
<b>Output</b>	-
<b>Propósito</b>	Configurar valores requeridos por hardware de arduino, como también iniciar la memoria sd e iniciar valores de la red Ethernet.

Nombre Función	Loop
Input	-
Output	-
Propósito	Mantener la ejecución del programa cargado en memoria de Arduino infinitamente.

Nombre Función	R
Input	-
Output	-
Propósito	Conectar una método de una clase con una interrupción, con el fin de registrar un valor para el sensor de flujo.

### Clase Almacén

Nombre Función	Almacenar
Input	Double: Valor registrado
Output	-
Propósito	Almacena en una lista valores registrados de algún sensor.

Nombre Función	Liberar
Input	-
Output	-
Propósito	Libera la lista almacenada, con esta acción se libera memoria SRAM de placa arduino uno

Nombre Función	Enviar_medicion
Input	-
Output	String: valores almacenados

<b>Propósito</b>	Recoge los valores de la lista y crea un string para devolver.
------------------	--

<b>Nombre Función</b>	<b>Set_sensor</b>
<b>Input</b>	String: nombre sensor
<b>Output</b>	-
<b>Propósito</b>	Guarda el identificador del sensor utilizado

<b>Nombre Función</b>	<b>Get_sensor</b>
<b>Input</b>	-
<b>Output</b>	String; nombre sensor
<b>Propósito</b>	Devuelve el nombre del sensor

## Clase Comunicación

Esta clase es la misma que la descrita anteriormente

## Clase SensorHumedad

<b>Nombre Función</b>	<b>transformarMedida</b>
<b>Input</b>	-
<b>Output</b>	Doublé
<b>Propósito</b>	Dado el valor analógico recogido por el sensor este es transformado a una escala de porcentaje

<b>Nombre Función</b>	<b>mostrarMedida</b>
<b>Input</b>	-
<b>Output</b>	Double: Medida
<b>Propósito</b>	Envia la medida recogida

Para las clases de SensorTemperatura, SensorNivel, Sensorflujo aplican las mismas funciones descrita en la sección de la Clase Sensorhumedad.