

MEMORIA DEL PROYECTO DE IOT

CONTROL DE UN SISTEMA ACUAPÓNICO MEDIANTE RASPBERRY

Jordi Sanpons & Carlos Pérez

(21/Junio/2019)



Contenido

Descripción y Objetivo de este Documento	2
PARTE I: LOS SISTEMAS ACUAPÓNICOS.....	3
Introducción	3
¿Qué es un Sistema Acuapónico?.....	3
¿Cómo funciona?	3
Elementos constitutivos de un Sistema Acuapónico.....	5
Acciones a Realizar por el Sistema IoT	6
1. Bombas de Agua del Estanque de Peces	6
2. Aireadores	7
3. Nivel de Agua del Estanque de Peces.....	7
4. Nivel/Evacuación del Agua en cada Bancal	8
5. Control del pH.....	8
6. Control de la Temperatura del Agua	8
7. Iluminación y Visión del Estanque	9
Ejemplos de Sistemas Acuapónicos.....	10
PARTE II: Proyecto Técnico	12
What's next?	22
Apéndice: Bibliografía y Enlaces de Interés.....	23

Descripción y Objetivo de este Documento

El presente documento tiene dos partes diferenciadas. Una primera Parte I donde se describe lo esencial de un Sistema Acuapónico y una Parte II donde se muestra el Proyecto Técnico basado en la placa Raspberry PI.

El objetivo de este documento es que el lector entienda el sistema acuapónico y los bloques esenciales del Proyecto Técnico.

No se ofrece una descripción exhaustiva de las líneas de código, funciones o librerías empleadas, conexiones eléctricas de los sensores, protocolos de comunicaciones, desarrollo de la página web y otros muchos detalles pues se excedería con mucho el propósito de este documento.

PARTE I: LOS SISTEMAS ACUAPÓNICOS

Introducción

Afin de comprender correctamente las acciones a realizar por el sistema IoT descrito en La Parte II, es necesario presentar una breve descripción previa que muestre qué es, cómo funciona y cuáles son las partes constitutivas esenciales de un sistema acuapónico tipo.

¿Qué es un Sistema Acuapónico?

Es un sistema destinado a la cría de plantas y cuyos nutrientes provienen de desechos producidos por peces. Por tanto, plantas y peces se crían conjuntamente en una relación de interdependencia que se definirá más adelante.

Las plantas producidas suelen ser comestibles y ornamentales pero también hay casos de árboles frutales.

La dimensión puede variar desde sistemas caseros de interior, sistemas de pocos m² en el patio de una casa para el consumo familiar de vegetales frescos, hasta sistemas comerciales de cientos o miles de m².

Se trata de sistemas escalables y cuya ampliación es directamente asumida mediante la apropiada expansión del sistema IoT descrito más adelante.

¿Cómo funciona?

El sistema reproduce un ciclo presente en la naturaleza y que se describe a continuación.

Los seres vivos que viven en el agua producen desechos provenientes de la digestión, de la descamación y de la propia respiración. A lo anterior, hay que añadir la descomposición de los propios animales tras su muerte, así como hojas, ramas y cualquier otro elemento externo que se añada al agua.

Estos desechos son incorporados al agua y deben ser eliminados para mantener a ésta limpia pues, de lo contrario, la contaminación que producirían acabaría matando a los seres que viven en ella.

En los ríos, el aporte de las fuentes y de la lluvia con agua limpia ayuda a su renovación pero en el caso de lagos y lagunas este aporte de agua nueva es insuficiente.

Un mecanismo existente en la naturaleza para la limpieza de todos estos elementos es el siguiente: los elementos de deshecho citados no son directamente absorbibles por las raíces de las plantas pero sí constituyen el alimento de una gran variedad de bacterias que de forma natural formarán colonias estables mientras se mantenga la presencia de alimento y las condiciones de temperatura, oxígeno y pH del agua adecuadas.

Una primera ola de estas bacterias comenzarán a consumir estos desechos quienes, a su vez, producirán nuevas formas de desechos que, de nuevo, serán alimento de otro grupo de bacterias quienes, una vez más, producirán nuevas sustancias de desechos.

Este ciclo supone que los desechos originales son así transformados por distintas oleadas de bacterias de forma que, finalmente, las sustancias producidas por éstas sí son ya nutrientes asimilables por las plantas situadas en

los márgenes del lago. Así, al ser absorbidas por las raíces de las plantas, son así eliminadas del agua con lo que se produce la purificación de ésta.

Este ciclo se perpetúa mientras se mantengan las condiciones de pH, temperatura y oxígeno adecuadas.



Las plantas descansan sobre una placa flotante (corcho, etc.) bajo la cual se desarrollan libremente las raíces. A medida que el bancal se llena de agua, las raíces toman los nutrientes disueltos en ella.

Cuando se vacía, y hasta que se vuelva a llenar, las raíces quedan al aire, tomando así la mayor parte del oxígeno necesario.



NOTA: Otros sistemas incorporan elementos verticales, perlita o guijarros en los bancales, lombrices para conseguir un extra de humus en el agua, etc, etc.

Elementos constitutivos de un Sistema Acuapónico

Conocido el proceso descrito en el apartado anterior, se entenderán los distintos elementos presentes en un sistema acuapónico.

Existen numerosas modalidades de sistemas acuapónicos. Presentaremos un sistema genérico pero perfectamente real sobre el que realizaremos el Proyecto IOT.

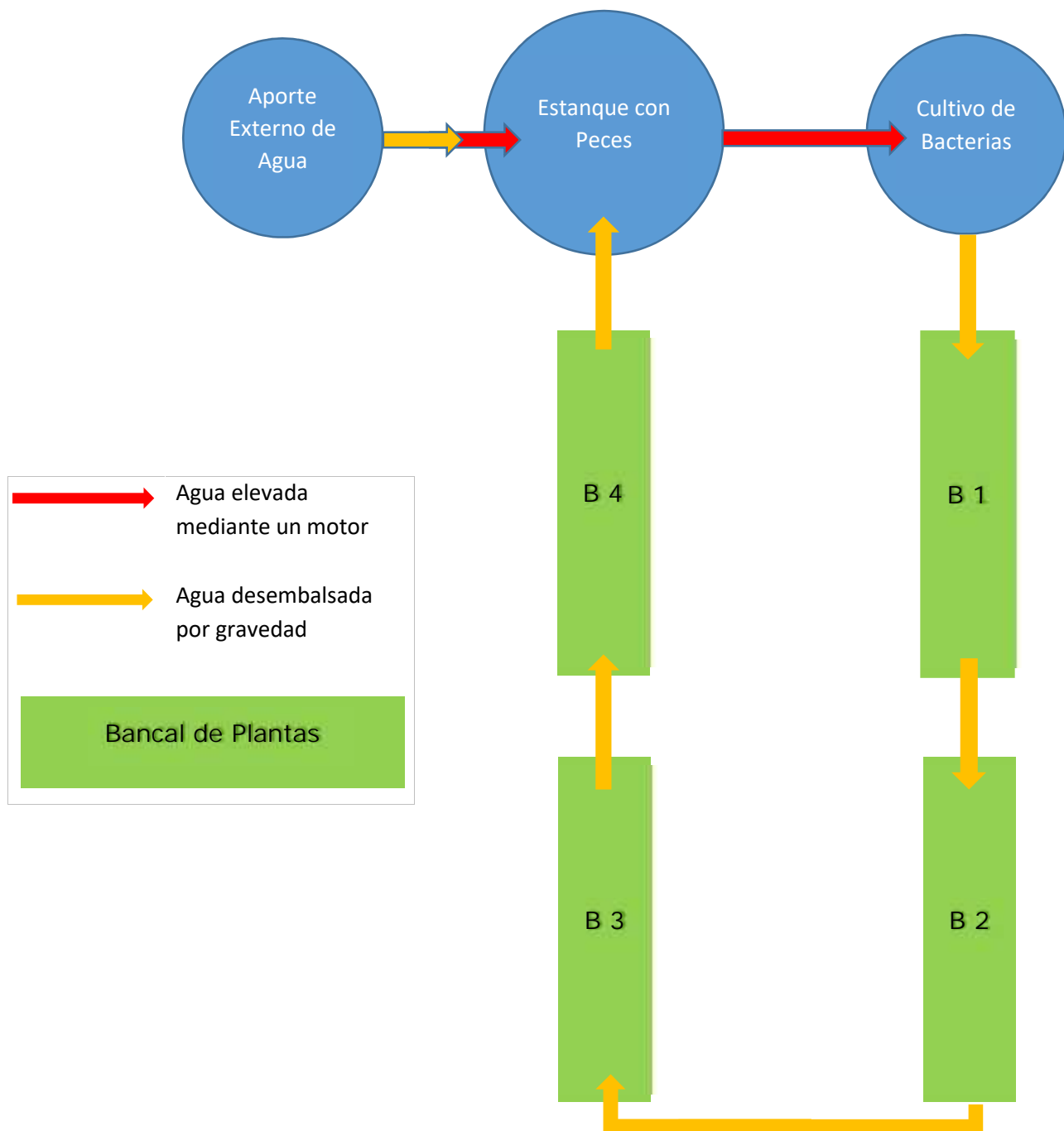


Fig 1: Vista de Planta

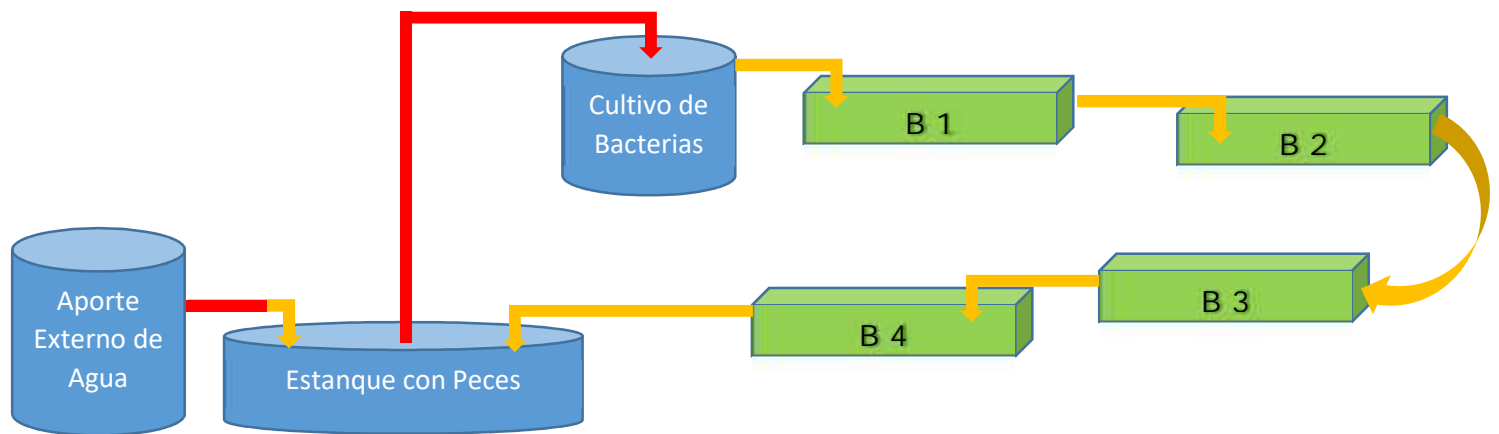


Fig 2: Vista lateral

El funcionamiento es el siguiente:

Unas bombas (sumergibles o externas) tomarán agua del fondo del Estanque de Peces y la conducirán al estanque de Cultivo de Bacterias. **Idealmente, éstas deberían ser los únicos motores presentes en el sistema.**

Del Estanque de Bacterias, el agua pasará por gravedad a un primer bancal de plantas (B1). Mientras éste se llena, las raíces irán absorbiendo los nutrientes disueltos. Cuando se complete el llenado, el agua se desembalsará, pasando a un siguiente bancal (B2).

Este proceso de recirculación del agua se completará cuando el último bancal (B4) desembalse en el Estanque de Peces con lo que se completará el ciclo.

Los estanques de peces y bacterias incorporarán unos aireadores que añadan oxígeno al agua.

La disminución de la cantidad de agua se deberá a la tomada por las raíces y a la que desaparezca por evaporación. Por tanto, se requiere de un aporte externo de agua que compense estas pérdidas.

Idealmente, los únicos motores presentes serán los de los aireadores y los que llevan el agua del estanque de Peces al de bacterias. La recirculación restante se producirá por gravedad.

Acciones a Realizar por el Sistema IoT

1. Bombas de Agua del Estanque de Peces

Como se dijo anteriormente, el cometido es elevar el agua desde el estanque de peces hasta el contenedor de bacterias e, idealmente,

Esta acción se realizará durante las 24 horas, 7 días a la semana, 365 días al año. Es un sistema crítico pues, en caso de fallo, todo el sistema se paraliza.

Es por ello que, afín de reducir la probabilidad de fallo, se incorporarán 2 bombas que conformarán así un sistema redundante.

El modo de funcionamiento es el siguiente: Sólo una de las bombas estará activa y trabajará durante 6 horas de forma continua. Tras finalizar este periodo, ésta se parará, siendo la otra bomba, hasta ahora en reposo, quien tome el relevo durante las siguientes 6 horas.

Esta alternancia se mantendrá de forma permanente y permitirá no sólo disminuir la probabilidad de fallo sino también aumentar la vida útil de las bombas al dividir por dos el tiempo de trabajo.

Cada bomba estará acompañada por 2 leds. Led verde que indicará que está funcionando correctamente en ese instante y led rojo que indica un mal funcionamiento de la misma.

En caso de fallo de la bomba activa, se producirá además una señal acústica de alarma para su reparación/sustitución, tomando la bomba en reposo el relevo de forma inmediata.

2. Aireadores

Para una mejora en la oxigenación del agua, se colocarán aireadores tanto en el estanque de peces como en el contenedor de bacterias.

Su funcionamiento puede ser constante sin más control que la detección de mal funcionamiento.

3. Nivel de Agua del Estanque de Peces

Como ya se ha comentado, consumo de agua depende de dos factores:

- La Evaporación
- Su Absorción por las plantas

Por tanto, es necesario añadir agua limpia al Estanque de Peces conforme ésta se consume.

Para ello se empleará un sensor de ultrasonidos que medirá la altura alcanzada por el agua y se establecerá un Valor Mínimo y otro Máximo cuyo valor medio determinará el Valor Óptimo del nivel de agua.

Tendremos dos casos posibles:

- La lectura del sensor llega al Valor Mínimo (Falta de Agua)

En este caso el sistema activará el relé que abra el paso de agua (Por gravedad o mediante una bomba). Este aporte de agua externa cesará cuando el sensor indique que se ha llegado al Valor Óptimo.

Si continúa bajando, se dará una señal de alarma para que un operario compruebe el origen del problema (Posible rotura del tanque, mal funcionamiento del sensor, etc)

- La lectura del sensor llega al Valor Máximo (Exceso de Agua)

En principio, sólo debería alcanzarse esta valor por un mal funcionamiento de la bomba de llenado.

Señal de alarma de mal funcionamiento para que un operario compruebe el problema.

4. Nivel/Evacuación del Agua en cada Bancal

El mismo sensor de ultrasonidos y con la misma metodología controlará el nivel del agua de cada bancal.

Cuando este nivel marque un nivel superior que indique que dicho bancal esté lleno, se producirán las siguientes acciones:

- Se parará/ralentizará la bomba de agua que esté activa en ese momento (Sólo en el caso del primer bancal).
- Se activará una electroválvula que se abra para evacuar el agua del bancal (En todos los bancales).

Esta agua pasará al siguiente bancal por gravedad y del último de ellos al Estanque de Peces.

El sensor también controlará el nivel cuando toda el agua haya sido evacuada.

Cuando se alcance este nivel mínimo, se producirán las siguientes acciones:

- Se cerrará la electroválvula.
- Se volverá a activar la bomba de agua del estanque de Peces (Sólo en el caso del primer bancal)

5. Control del pH

Es, junto al correcto funcionamiento de las bombas de agua, el otro factor crítico pues la deriva hacia valores indeseables puede ralentizar la actividad normal de peces y bacterias, incluso llegar a matarlos.

Cada tipo de pez tiene su margen óptimo de pH. Determinado éste, un medidor de pH irá haciendo mediciones del mismo de forma regular dando una señal de alarma cuando su valor se acerca a los extremos del margen óptimo y, por tanto, sea posible tomar medidas para su vuelta a la normalidad antes de alcanzar valores críticos.

Será suficiente con medir el pH del agua 4 veces por día.

6. Control de la Temperatura del Agua

Se controlará la temperatura mediante un termómetro en el agua y otro en el exterior para establecer así la dependencia entre ellos y determinar la posible necesidad de un sistema de calefacción.

7. Iluminación y Visión del Estanque

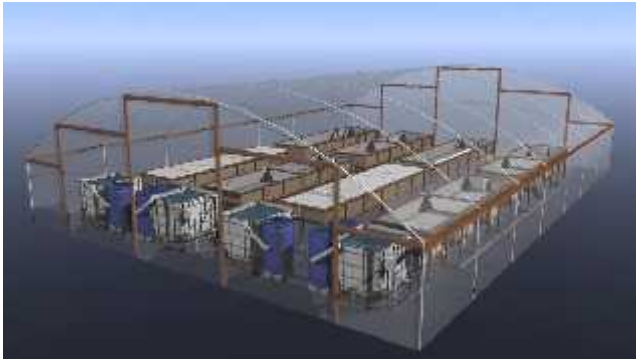
Colocación de una cámara(s) en una de las esquinas para tener la mejor visión posible del interior del estanque.

Interesante para controlar la actividad de los peces, su salud, el aprovechamiento de la comida, así como detectar posibles peces muertos que, además, podrían obturar la salida del agua.

También cuando se añaden alevines, ver su adaptación y ataques de adultos.

La actividad de la cámara sería a petición del operador que también activaría una serie de luces en el interior del estanque para observar cuando no haya luz diurna.

Ejemplos de Sistemas Acuapónicos



Diseños de posibles sistemas



Bancales de Plantas



Distintos Sustratos



Sistemas caseros/domésticos



Sistemas Verticales

PARTE II: Proyecto Técnico

PROYECTO TECNICO

Sistema de acuaponía

1

DE LA TEORÍA ... A LA REALIDAD



2

DE LA TEORÍA ... A LA REALITAT



3

PLANTEJAMENT DEL PROJECTE (FASE 1)

- ▶ PROGRAMACIO I: Àrea de gestió de resultats en web
- ▶ PROGRAMACIO II: Gestió separada de cada mòdul
- ▶ ELECTRONICA I: Peculiaritats dels sensors i com integrar-los al sistema
- ▶ ELECTRONICA II: La Raspberry Pi 3 B+

4

PROGRAMACIO I

- ▶ Simulador de sensors, actuadors i altres
- ▶ Gestió de configuració importada des de arxius JSON
- ▶ Multiprocés. Fils pel servidor i per els mòduls

```

# =====
# FUNCIONS PER PER OPCIO DOTS DELS TANKS
def LLEGEIR_SENSORS_FACE():
    # Definir l'estructura de dades
    dades_face = {}
    # 1. Llistat de sensors
    s1 = DOTS_FACE_0_100()
    s2 = DOTS_FACE_0_100()
    s3 = DOTS_FACE_0_100()
    s4 = DOTS_FACE_0_100()
    dades_face = {s1, s2, s3, s4}
    return dades_face

def LLEGEIR_CONFIG_SENSORS_FACE():
def LLEGEIR_STATUS_ACTUADORS_FACE():
    
```

```

# Funció que crea una llista de fils independents
# Definim la llista de fils buida, i després anem afegint ...
threads = []

# Primer fil
# Definim amb def i lambda
pt = threading.Thread(target=pt1_1)
pt.start()
threads.append(pt)

# Des d'aquí ja sabem a que classe la volem per aconseguir el segon fil
time.sleep(10)

# Segon fil
# Funció idèntica del Tank 01
pt2 = threading.Thread(target=pt2_2)
pt2.start()
threads.append(pt2)
    
```

Salvem a: control.py - Programació

```

def LLEGEIR_CONFIG_SENSORS_FACE():
    """ Funció que llegeix les dades de la llista de configuració, les envia amb els tanks """
    # Per exemple, podem enviar [tank_01, "sensors", "sensors", "sensors", "sensors"]
    class_configuracio = [{"tank": "tank_01", "sensors": "sensors"}]

    # El resultat que la funció retorna és un diccionari de dades per a cada tank
    # per exemple:
    # {
    #     "tank": "tank_01",
    #     "sensors": {
    #         "s1": 100,
    #         "s2": 100,
    #         "s3": 100,
    #         "s4": 100
    #     }
    # }
    
```

5

PROGRAMACIO I - FLASK

- ▶ Funcionament basat en events (crides a URLs)
- ▶ Definició de rutes i funcions
- ▶ Definició de les dades emprades
- ▶ Motor de plantilles JINJA2
- ▶ Blocs i transferència de dades
- ▶ CSS i Javascript

```

# Definim la funció i la funció que retorna la llista de tanks
def index():
    # Definim la funció
    @app.route("/")
    def index():
        # Definim la funció
        @app.route("/sensors/resumen")
        def sensors_resumen():
            # Definim la funció
            @app.route("/actuadors/resumen")
            def actuadors_resumen():
                # Definim la funció
                dades = pagina_actuadors_resumen()
                return render_template("actuadors_resumen.html", dadesweb = dades)

            # Definim la funció
            @app.route("/actuadors/ctrl/tank_01")
            def actuadors_ctrl_tank_01():
                # Definim la funció
                dades = pagina_actuadors_ctrl_tank_01()
                return render_template("actuadors_ctrl.html", dadesweb = dades)
    
```

```

# Definim la funció
def index():
    # Definim la funció
    return render_template("index.html", dadesweb = dades)
    
```

6

Salvem a: control.py - Programació

PROGRAMACIO I - FLASK

```
{% extends "plantilla.html" %}

<!-- Preservar en una taula grafica els valors retornats dels 4 ACTUADORS del tank a -->
<!-- Recuperem del render, template('actuadors.html', dadesweb = dades) -->
<!-- dadesweb[0] on la 0 fa referència al número de tank processat, que es el mateix que el id_tank -->
<!-- dadesweb = [titol pagina, data marcatemporal, datapaisk] -->
<!-- dadesweb[0] = titol pagina -->
<!-- dadesweb[1] = [data, hora] Marca temporal de quan s'han fet les consultes als sensors -->
<!-- dadesweb[2] = [[tank, altura_status_AQUARIUM()], ...] -->
<!-- -->

{% block titol %}
{{ dadesweb[0] }}
{% endblock %}

{% block main %}

<p>Nombre de Tanks monitoritzats: {{ dadesweb[2][0] }} </p>
<p>
    Timestamp: chr+
    DATA: {{ dadesweb[1][0] }} chr+
    HORA: {{ dadesweb[1][1] }}
</p>
```

```
<!-- HEADER BLOCK -->
<header>
</header>

<!-- MAIN BLOCK -->
{% block main %}
{% endblock %}

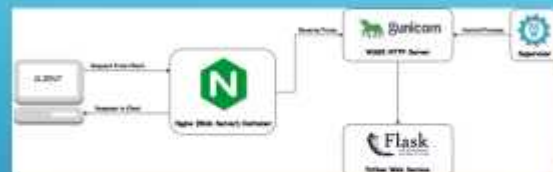
<!-- FOOTER BLOCK -->
<footer>
</footer>
```

SEBASTIAN@GMAIL - PRESENTACIO

7

PROGRAMACIO I - GESTIO WEB

- ▶ Sistema NGINX + MariaDB + Gunicorn
- ▶ Sistema de proves en Flask
 - ▶ Aplicació en Flask
 - ▶ Plantilles HTML | JINJA2
 - ▶ Intercanvi de dades entre aplicacions



SEBASTIAN@GMAIL - PRESENTACIO

8

PROGRAMACIO I - GESTIO WEB

INICI

Sensors

- TOT31
- Tank 01
- Tank 02
- Tank 03
- Tank 04

Actuadors

- TOT31
- Tank 01
- Tank 02
- Tank 03
- Tank 04

Crònograma

- Tank 01

Config

- Sensors
- Override Actuadors
- Restart Raspberry

Ajuda

Tank ID	Sensor_1	Sensor_2	Sensor_3	Sensor_4
	Temperatura	Humitat	Nivell	Llum
	[20 , 30]	[50 , 60]	[80 , 90]	[0 , 100]
[1]	22	77	75	12
	[30 , 40]	[20 , 25]	[75 , 95]	[0 , 0]
[2]	89	30	89	22
	[40 , 50]	[80 , 90]	[60 , 90]	[25 , 75]
[3]	32	8	68	33
	[50 , 60]	[0 , 99]	[50 , 90]	[100 , 100]
[4]	39	97	38	67

DADES CORRESPONDENTS AL TANK Nº 01
(DATA: 18/06/2019 HORA: 23:15:30)

SENSOR	Minim	Maxim	ACTUAL	FLAGS
Temperatura	20	30	35	
Humitat	50	60	75	
Nivell	80	90	84	
Llum	0	100	27	

INSTANTANEA EN TEMPS ACTUAL:



© 2019 ACUAPONIC - PrewittCo

9

PROGRAMACIO I - GESTIO WEB

INICI

Sensors

- TOT31
- Tank 01
- Tank 02
- Tank 03
- Tank 04

Actuadors

- TOT31
- Tank 01
- Tank 02
- Tank 03
- Tank 04

Crònograma

- Tank 01

Config

- Sensors
- Override Actuadors
- Restart Raspberry

Ajuda

Nombre de Tanks monitoritzats: 4

TimeStamp:
DATA: 18/06/2019
HORA: 23:15:55

Tank ID	Actuador_1	Actuador_2	Actuador_3	Actuador_4
1	False	True	False	False
2	False	False	False	False
3	False	False	True	True
4	False	True	False	False

ACTUADOR STATUS FLAGS

Actuador 1 True

Actuador 2 True

Actuador 3 False

Actuador 4 True

© 2019 ACUAPONIC - PrewittCo

10

PROGRAMACIO II – MODUL (TANK 01)

```
Experiment 1 program .....
# Define status actuators as: (True, False, False, True)
.....
SENSOR1: [DHT10 - Humitat] ----- Range (min/max): (0 - 55)
Valor recollit: 47,2000076293849
SENSOR2: [DHT17 - Temperatura] ----- Range (min/max): (21 - 30)
Valor recollit: 27,00000041851073
SENSOR3: [DS18B20 - Temperatura] ----- Range (min/max): (0 - 100)
Valor recollit: 20,0
SENSOR4: [HC-SR04 - Distància] ----- Range (min/max): (205 - 200)
Valor recollit: 85,33861200418197
SENSOR5: [LDR/Pulsador - Boolean] ----- Range (min/max): (0 - 1)
Valor recollit: 0
.....
Actuador_1 / Actuador_2 / Actuador_3 / Actuador_4 (STAT AL LLEGIR SENSORS)
0 0 0 1
.....
Actuador_1 / Actuador_2 / Actuador_3 / Actuador_4 (STAT AL PROCESAR SENSORS)
True False False True
```

TASKES

- 0.- Recuperar i preparar dades de configuració de sensors i actuadors
- 1.- Llegir sensors T1
- 2.- Llegir status actuadors T1
- 3.- **Processar dades**
- 4.- Modificar si es cau status actuadors T1
- 5.- Empaquetar les dades
- 6.- Imprimir les dades a consola
- 7.- Enviar les dades al servidor web

```
{
  "Tank_ID", "Sensor_0", "Sensor_1", "Sensor_2", "Sensor_3", "Sensor_4",
  [1,
    ["humitat", "DHT22", [0, 10], [0, 100]],
    ["temperatura", "DHT22", [25, 35], [-40, 180]],
    ["temperatura", "DS18B20", [10, 30], [-50, 200]],
    ["nivell", "SRHC04", [90, 99], [2, 400]]
  ],
}
```

13

ELECTRONICA I

► Sensors:

- Temperatura i Humitat. Serie DHT
- Temperatura Serie DS18B20 (protocolo 1 wire)
- Nivell / Distància ULTRASO SRHC04
- LDR

► Actuadors:

- Relés, bombes



14

ELECTRONICA II

► Raspberry

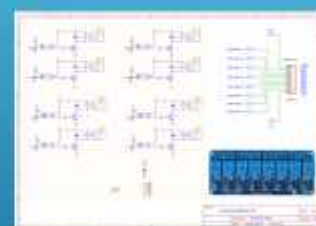
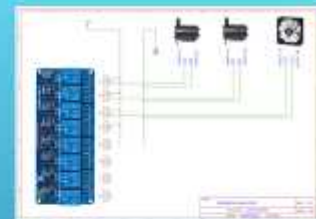
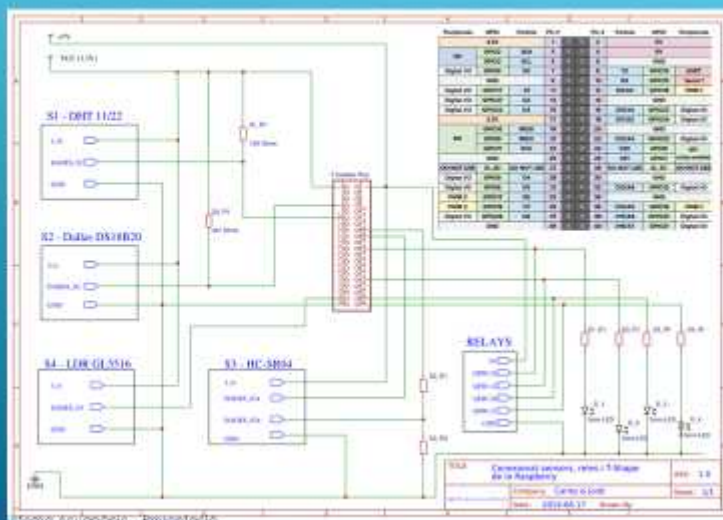


Peripherals	GPIO	Port/IO	Pin #	Pin #	Port/IO	GPIO	Port/IO
3.3V			1	2	5V		
GPIO	GPIO4	SDA	3	4	5V		
GPIO	GPIO5	SCL	5	6	GPIO		
Digital I/O	GPIO4	DE	7	8	TX	GPIO14	UART
GND			9	10	RX	GPIO15	Serial 1
Digital I/O	GPIO17	D1	11	12	D0-D3	GPIO18	PWM 1
Digital I/O	GPIO22	D2	13	16			
Digital I/O	GPIO23	D3	15	18	D0-D3	GPIO24	Digital I/O
3.3V			19	20	D0-D3	GPIO25	Digital I/O
Digital I/O	GPIO10	MDIO	18	23			
GPIO	GPIO6	MISO	21	22	D0-D3	GPIO26	Digital I/O
GPIO	GPIO11	MOSI	23	24	CPU	GPIO27	SP
GND			25	26	CE1	GPIO27	Chip enable
DO NOT USE	GPIO2	DO NOT USE	27	28	DO NOT USE	GPIO28	DO NOT USE
Digital I/O	GPIO5	D4	29	30			
Digital I/O	GPIO6	D5	31	32	D0-D3	GPIO29	Digital I/O
PWM 2	GPIO12	D6	33	34			
PWM 3	GPIO13	D7	35	36	D0-D3	GPIO30	PWM 1
Digital I/O	GPIO20	D8	37	38	D0-D3	GPIO31	Digital I/O
GND			39	40	D0-D3	GPIO32	Digital I/O



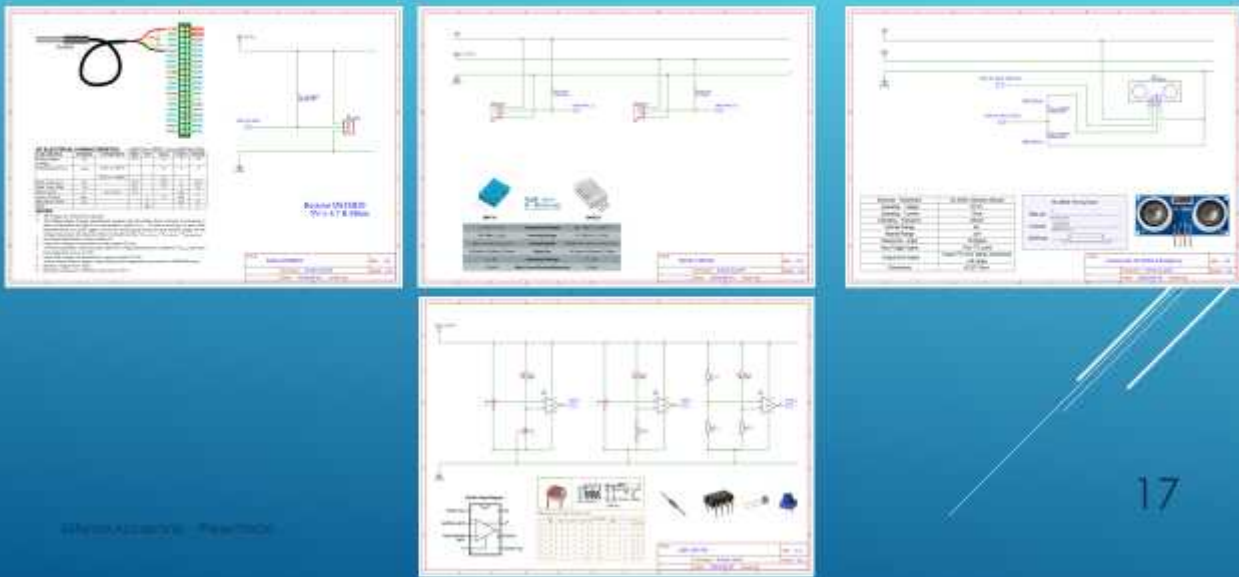
15

ELECTRONICA - ESQUEMES



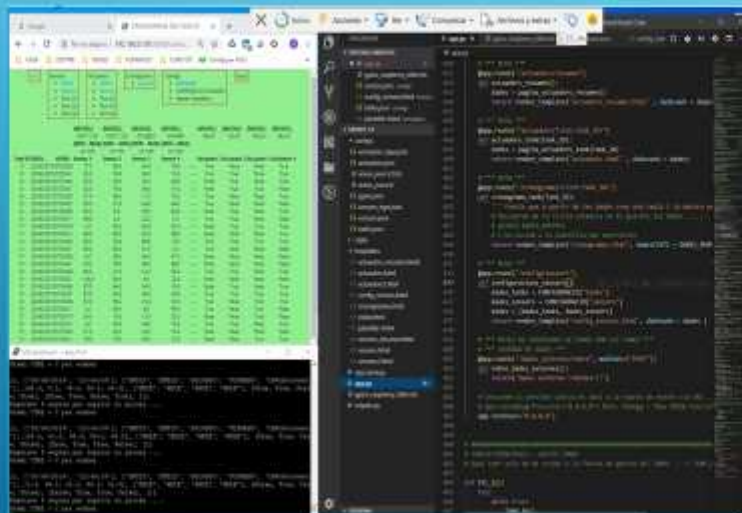
16

ELECTRONICA – ESQUEMES



17

TOT JUNT



18

FUTURES AMPLIACIONS

- ▶ Mòduls independents (ESP32)
- ▶ Comunicacions:
 - ▶ LORAWAN / MQTT
- ▶ Segurització entorn:
 - ▶ Docker
 - ▶ Servidor NGINX + GUNICORN
- ▶ Aplicació mòbil
- ▶ Ampliació funcions pagina web
- ▶ Gestió i interconnexió mòduls
- ▶ Control des de Alexa ?

What's next?

Debido a la falta de tiempo, la persona que ha realizado el Proyecto Técnico no ha tenido tiempo para concluir el proyecto en su totalidad.

Se pretende desarrollar el mismo en un futuro cercano, mejorando algunas de las partes aquí expuestas y añadiendo otras como alarmas y control distantes, sumando así aspectos más propios de IoT.

Esta es nuestra intención sincera a día de hoy pero como se puede leer en la pantalla 2 (página 12) de este documento....

La ambición humana es infinita

La pereza también lo es

Juan Perogrullo (El Olvidado)

Apéndice: Bibliografía y Enlaces de Interés

- Universidad de Hawaii: <https://maui.hawaii.edu/aquaponics/>
- Universidad de las Islas Vírgenes: <https://www.uvi.edu/research/agricultural-experiment-station/aquaculture-home/aquaponic-systems/default.aspx>
- Universidad de Chicago: <https://www.luc.edu/sustainability/initiatives/urban-agriculture/aquaponics/>
- Libro: "How to Successfully Grow Aquaponics Plants", Celine Walker