

UNIVERSIDAD DE TARAPACÁ



UNIVERSIDAD DE TARAPACÁ
ARICA - CHILE

ESCUELA UNIVERSITARIA DE INGENIERÍA INDUSTRIAL, INFORMÁTICA Y DE SISTEMAS

EUIIS

Área de Ingeniería en Computación e Informática



“Informe final de proyecto” Sistema Hidropónico

Autor(es): Fabián Guarachi

Patricio

Tudela

Asignatura: Proyecto 2

Profesor(es): Diego Arcena

ARICA, 03/12/2018

Historial de cambios

Fecha	Versión	Descripción	Autor(es)
01/12/2018	1.0	Primera versión del documento	Patricio Tudela
03/11/2018	2.0	Informe 1 y 2 agregado	Patricio Tudela
12/12/2018	3.0	Agregada fotos del resultado final y otros.	Patricio Tudela

Tabla de contenidos

Historial de cambios	1
Tabla de contenidos	2
1. Introducción	3
2. Diseño de maqueta experimental	3
3. Rediseño de maqueta	4
3.1. Solucion nutritiva	4
5. Análisis y diseño	6
5.1. Análisis	6
5.1.1. Requerimientos funcionales.	6
5.1.2. Requerimientos no funcionales.	6
5.2. Diseño	7
5.2.1. Herramientas, hardware y software utilizados	7
5.2.2. Diagrama de la solución	8
5.2.3. Diagrama de casos de uso de sistema	9
5.2.3.1. Descripción de casos de uso	10
5.2.3.1.1. Recolectar Dato	10
5.2.3.1.2. Análisis de datos	11
5.2.3.1.3 Activar función reguladora	12
5.2.3.1.4 Mostrar datos	13
5.2.3.1.5 Iniciar sistema	14
5.2.3.1.6 Apagar sistema	15
5.2.4 Diagrama de roles	16
5.2.5 Diagrama de actividades (General)	17
5.2.6 Diagramas de secuencia	18
5.2.7. Diagrama de clases	24
5.2.8 Diseño de la Interfaz	25
5.6. Producto del análisis	27
6. Proceso de construcción de producto final	27
7. Carta Gantt	30
8. Conclusión	31
9. Referencias	32

1. Introducción

En este curso se inició con la idea de idear y crear un sistema hidropónico NFT autonómico, es decir, sin la interacción humana. Se entregó material para ser estudiado a parte de clases didácticas explicando el tema principal, la Hidroponía. Utilizando el pensamiento IoT se debía ocupar un dispositivo raspberry para el funcionamiento de este sistema, siendo esta parte el núcleo del proyecto.

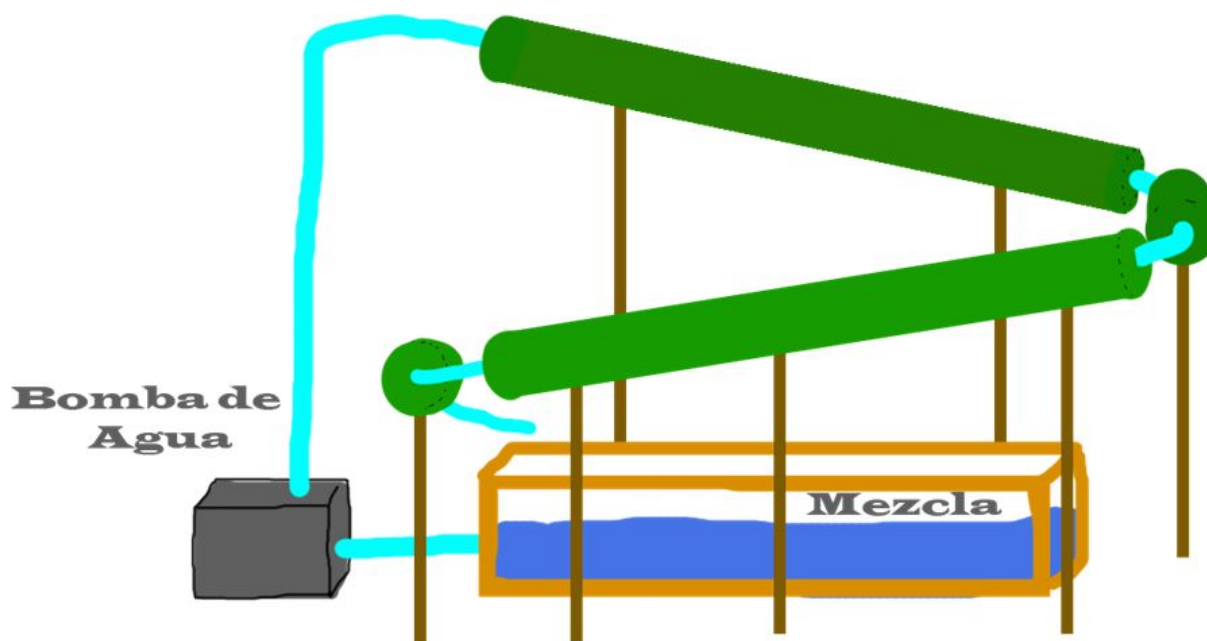
Este documento presenta todo lo realizado en el curso, conteniendo de análisis y diseño de maqueta, análisis y diseño de sistemas, maqueta, arquitecturas de sensores y cliente-servidor resultantes.

2. Diseño de maqueta experimental

Para la creación de la maqueta se tuvo que tener en cuenta lo estudiado y lo permitido en el proyecto. El uso de materiales reciclables y materiales óptimos para la plantación como también el ambiente del sistema.

Para el correcto funcionamiento de la caída del agua se ha optado por una inclinación de los tubos PVC aproximada de 10° , lo cual permite un flujo continuo de una película de agua. Se Optó por hacer dos agujeros por tubo que albergarían una planta cada uno, teniendo entre estos un espacio de 20 cm.

Después de lo descrito anteriormente además de una lluvia de ideas se ha ideado un diseño experimental el cual es el siguiente.



Este diseño está constituido por:

- Base de madera.
- Cinco pilares PVC de 32mm azules de 15, 20, 27 ,37 y 45 cm de altura.
- Tres pilares de PVC de 75mm grises de 60 cm de largo.
- Un contenedor de solución.
- Bomba de agua para mover la solución verticalmente.
- Tres “codos” PVC para intersección de los tubos.
- Dos agujeros por tubo para insertar las plantas.

Gracias a esta combinación de medidas logramos que los tubos PVC tengan una inclinación de aproximada de 10 grados el cual permite el movimiento esperado para el correcto riego de las plantas.

¿Porque este diseño?

Pensamos que era lo más adecuado para ahorrar espacio y por su fácil movilidad y limpieza ya que es posible retirar los tubos manteniendo la base y pilares. Fueron utilizados una mezcla entre materiales reciclados y otros que fueron comprados.

3. Rediseño de maqueta

Se ha revisado el diseño de la maqueta anterior y se a detectado inestabilidad en el soporte para el tubo a más altura. Debido al poco tiempo restante, falta de fondos y materiales, se ha decidido quitar este último tubo PVC utilizando sus materiales para el mejor soporte de los demás. Aun así se ha mantenido su funcionamiento e idea principal.

3.1. Solucion nutritiva

Con la ayuda de un experto en el tema obtuvimos información para lograr una solución ideal para ser utilizada, además de condiciones del ambiente y agua que deben ser tomadas en cuenta. Se resumen a continuación.

Temperatura del ambiente: Alrededor de 25°C

Humedad del ambiente: 60%

Conductividad eléctrica del agua: 2.3 - 2.5 ds/m

PH de la solución: 4,8-5,2

Para lograr la mayoría de estos requerimientos se utilizará Ultrasol, una mezcla de macronutrientes que ayudan a cumplir estos requerimientos. Para ser exactos se utilizaran 27.75 gramos de este producto para un recipiente de capacidad de 5 litros.

4. Producto resultante

Este es la maqueta en una de sus formas finales. Cabe destacar, como se puede ver, que falta el recipiente que contendrá la solución nutritiva.



Maqueta Experimental.

5. Análisis y diseño

5.1. Análisis

5.1.1. Requerimientos funcionales.

- ❖ El sistema debe extraer datos del ambiente.
- ❖ El sistema debe detectar la profundidad del agua.
- ❖ El sistema debe detectar el ph de la solución.
- ❖ El sistema debe detectar la temperatura del ambiente.
- ❖ El sistema debe detectar la humedad del ambiente.
- ❖ El sistema debe detectar la conductividad eléctrica.
- ❖ El sistema debe validar y entregar datos recolectados.
- ❖ El sistema analiza datos recolectados y verifica si es necesario tomar acción.
- ❖ El sistema permite la activación de hardware para regular a este en caso necesario.
- ❖ El sistema podrá enviar datos recolectados y alertas a los clientes
- ❖ El sistema NO permite el ingreso manual de datos y/o mensajes.
- ❖ El sistema debe ser encendido y apagado por el cliente.

5.1.2. Requerimientos no funcionales.

Software:

- ❖ Para cada recolección de datos se necesitan de a lo mas 1 segundo para su correcto funcionamiento.
- ❖ La recolección del dato PH debe ser realizada una vez por dia (Se a reducido el tiempo de este para percatarse de este a baja escala, como lo es este proyecto). Deberá estar en un rango entre 4.8 y 5.2 de PH.
- ❖ La recolección del dato de temperatura de ambiente debe ser realizada cada 3 minutos. Esta debe estar a temperatura ambiente, alrededor de 25°C, mientras que la humedad debe estar al 60% de humedad con un gran margen permitido.
- ❖ La recolección del dato de conductividad eléctrica debe ser medido cada 10 minutos. Debe estar en un rango de 2.3 a 2.5 ds/m
- ❖ La recolección del dato de temperatura del agua será medido cada minuto. Debe estar alrededor de 21°C.
- ❖ El sistema operativo para el cliente debe ser Windows 10.
- ❖ El sistema operativo para el servidor debe ser Raspbian.
- ❖ El sistema utiliza el lenguaje de programación Python.
- ❖ El sistema deberá presentar una visualización adecuada para que cliente vea los datos recolectados de este.

Hardware:

- ❖ El sistema funciona solamente mientras esté encendido.
- ❖ El dispositivo Raspberry Pi requiere conexión a internet.

5.2. Diseño

5.2.1. Herramientas, hardware y software utilizados

Hardware:

- ❖ **Raspberry Pi 3 model B**: Ordenador de placa reducida, que sirve para pequeños y medianos proyectos dentro de la ingeniería.
- ❖ **Micro Usb**: cable que sirve de alimentación para el RB Pi.
- ❖ **Micro SD**: Tarjeta de almacenamiento que utiliza el RB Pi.
- ❖ **Protoboard WB-100**: Tablero que contendrá las conexiones eléctricas de sensores y otros componentes.
- ❖ **Sensor de flujo de agua YF-S201**: Sensor encargado de medir el flujo de agua dentro del sistema.
- ❖ **Sensor de distancia ultrasonico HC-SR05 HY-SRF05**: Sensor encargado de medir la capacidad del estanque de agua.
- ❖ **Sensor de conductividad eléctrica EC METER V1.0**: Permitirá medir la solución del agua.
- ❖ **Sensor de temperatura y humedad DHT22**: Encargado de medir la temperatura y humedad del sistema.
- ❖ **Sensor de ph para RaspBerry Pi+**: Sensor encargado de medir la calidad del agua del sistema.
- ❖ **Placa de 4 relays**: Placa para controlar altas cargas de corriente y voltaje.
- ❖ **Terminal adaptador de sensores V2.0**: Conector universal para sensores.
- ❖ **Mini Bomba de agua**: Será la encargada de bombear el agua en el sistema.
- ❖ **Valvula de solenoide MCID0726**: Será encargado de dar el paso del agua hacia la reserva.
- ❖ **Sensor de temperatura DC18820 tipo termocupla**: Sensor encargado de medir la temperatura del agua.

Software:

- ❖ **Raspbian**: Sistema operativo que utilizará el RB Pi
- ❖ **Python 3.6**: Lenguaje de programación que utilizará el RB Pi

5.2.2. Diagrama de la solución

Sensores/ actuadores

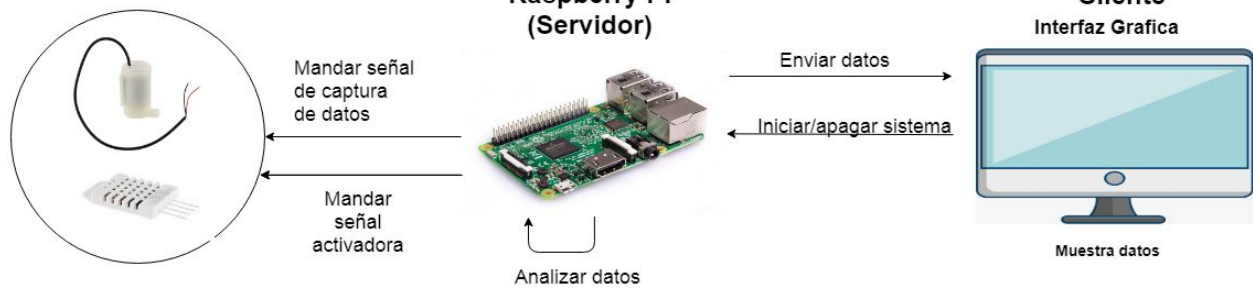


Figura 1. Diagrama de la solución.

Como vemos en el diagrama de la solución (figura 1), el Raspberry actuará como el servidor a la vez que controla todo lo que tiene que ver con la recolección de datos y regulación del sistema. Por otro lado una computadora personal funcionará como el cliente, quien deberá conectarse al servidor montado por el raspberry, esta computadora posee una interfaz gráfica encargada de mostrar los datos recolectados e información del sistema.

5.2.3. Diagrama de casos de uso de sistema

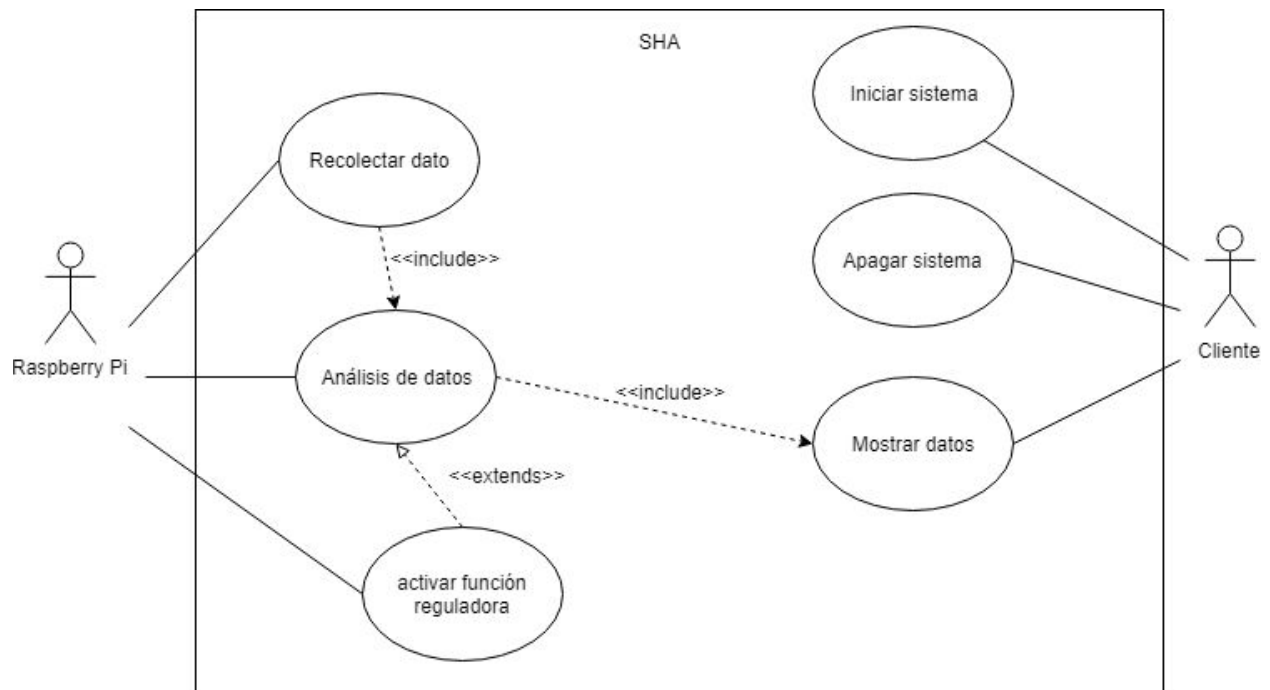


Figura 2. Diagrama de casos de uso

El caso de uso **Recolectar dato** engloba los siguientes casos de usos específicos para los sensores:

- ❖ Recolectar PH.
- ❖ Recolectar Temperatura de ambiente.
- ❖ Recolectar Humedad del ambiente.
- ❖ Recolectar Conductividad eléctrica.
- ❖ Recolectar flujo de agua.
- ❖ Recolectar profundidad

El caso de uso **Activar función reguladora** engloba los siguientes casos de usos específicos para los dispositivos actuadores:

- ❖ Activar bomba de agua.
- ❖ Abrir llave.

5.2.3.1. Descripción de casos de uso

A continuación se describirán los casos de usos vistos en el diagrama de casos de uso (figura 2).

5.2.3.1.1. Recolectar Dato

Como se observa en el diagrama de casos de uso, este caso se divide en varios otros específicos para cada sensor. Por lo tanto si se requiere recolectar el Ph se deberá especificar tal intención en el código.

Nombre	Recolectar dato	
Descripción:	Permite recolectar los datos de todo el sistema (físicos)	
Actor:	Dispositivo Raspberry Pi 3 Model B.	
Precondiciones:	El sensor debe estar disponible para su funcionamiento.	
Flujo Normal:		
RBPI 3 M B	Sistema de recolección de datos.	
1.-Mandar señal de recolección de dato a sensor.	2.-Adquirir <u>dato</u> . 3.-Enviar <u>dato</u> al servidor (Incluye C.U. Análisis de datos).	
Flujo alternativo		
RBPI 3 M B	Sistema de recolección de datos.	
	2.-Envía mensaje de error	
Pos Condiciones:	Los datos son recolectados exitosamente.	

Dato: El dato dependerá del sensor que esté recolectando tal dato y tendrá su formato único.

5.2.3.1.2. Análisis de datos

Nombre	Análisis de datos	
Descripción: Permite verificar que el dato cumpla con sus restricciones.		
Actor: Dispositivo RaspBerry Pi 3 Model B		
Precondiciones: El dato debe haber sido recolectado.		
Flujo Normal:		
RBPI 3 M B	Sistema de análisis de datos	
1.-Enviar <u>restricciones correspondientes</u> y dato.	2.-Analizar dato. 3.-Validar dato. 4.-Enviar <u>información</u> y dato a cliente.	
Flujo alternativo		
RBPI 3 M B	Sistema de análisis de datos	
	3.-Invalidar dato. 4.-Incluye al C.U. Activar función reguladora.	
Pos Condiciones: El dato es validado y enviado exitosamente.		

Restricciones correspondientes: Cada dato deberá ser analizado con sus restricciones correspondientes al sensor con el cual se extrajo. Ej: Rango de temperatura, rango de PH, etc. Estas restricciones son almacenadas por el mismo servidor.

Información: Puede ser un mensaje especificando el dato recogido, o una alerta del sistema.

5.2.3.1.3 Activar función reguladora

Como vimos en el diagrama de casos de uso este caso contiene casos específicos los cuales se encuentran generalizados, solo habrá que especificar qué función se quiere en el código para activar el dispositivo actuador.

Nombre	Activar función reguladora							
Descripción:	Permite al dispositivo actuar de acuerdo al análisis.							
Actor:	Dispositivo Raspberry Pi 3 Model B							
Precondiciones:	Se ha enviado una alerta.							
Flujo Normal:	<table border="1"> <tr> <td>RBPi 3 M B</td> <td>Sistema de activación de funciones reguladoras</td> </tr> <tr> <td>Este C.U será activado al encontrar un dato fuera del rango (Extiende a Análisis de datos)</td> <td></td> </tr> <tr> <td>1.-Enviar <u>parametros</u>.</td> <td>2.-Realizar función con parámetros dados.</td> </tr> </table>		RBPi 3 M B	Sistema de activación de funciones reguladoras	Este C.U será activado al encontrar un dato fuera del rango (Extiende a Análisis de datos)		1.-Enviar <u>parametros</u> .	2.-Realizar función con parámetros dados.
RBPi 3 M B	Sistema de activación de funciones reguladoras							
Este C.U será activado al encontrar un dato fuera del rango (Extiende a Análisis de datos)								
1.-Enviar <u>parametros</u> .	2.-Realizar función con parámetros dados.							
Pos Condiciones:	El problema a sido solucionado adecuadamente.							

Parámetros: Son datos necesarios para que los dispositivos actuadores puedan solucionar el problema eficazmente. Ej Temperatura fuera de rango. Ingresar agua al sistema mediante la bomba de agua: potencia: X, Tiempo Y segundos.

5.2.3.1.4 Mostrar datos

Nombre	Mostrar datos	
Descripción: Permite mostrar la información recolectada del sistema.		
Actor: Servidor (PC).		
Precondiciones: El dato es analizado (Es incluido en Análisis de datos)		
Flujo Normal:		
Servidor	Sistema de muestra de datos	
Este caso de uso es (Incluido en el C.U Análisis de datos) 1.-Solicita dato de recolección al sistema	2.-Muestra información y datos recolectados.	
Flujo alternativo:		
Servidor	Sistema de muestra de datos	
1.-Entrega alerta y datos recolectados.	2.- Muestra alerta y dato recolectado.	
Pos Condiciones: La información y datos son mostrados exitosamente al cliente.		

5.2.3.1.5 Iniciar sistema

Nombre	Iniciar sistema	
Descripción: Permite iniciar las funcionalidades de este sistema		
Actor: Servidor (PC).		
Precondiciones: El sistema debe estar apagado.		
Flujo Normal:		
Servidor	Sistema	
1.-Solicita inicio del sistema.	2.-Inicia funcionalidades.	
Pos Condiciones: Todos los dispositivos han iniciado actividades.		

5.2.3.1.6 Apagar sistema

Nombre	Apagar sistema	
Descripción: Permite desactivar las funcionalidades de este sistema.		
Actor: Servidor (PC).		
Precondiciones: El sistema debe estar encendido		
Flujo Normal:		
Servidor	Sistema	
1.-Envía señales de paro de actividades a dispositivos.	2.-Desactiva funcionalidades.	
Pos Condiciones: Todos los dispositivos han dejado de realizar actividades.		

5.2.4 Diagrama de roles



Figura 3. Diagrama de roles.

Esta figura muestra los roles principales del sistema

5.2.5 Diagrama de actividades (General)

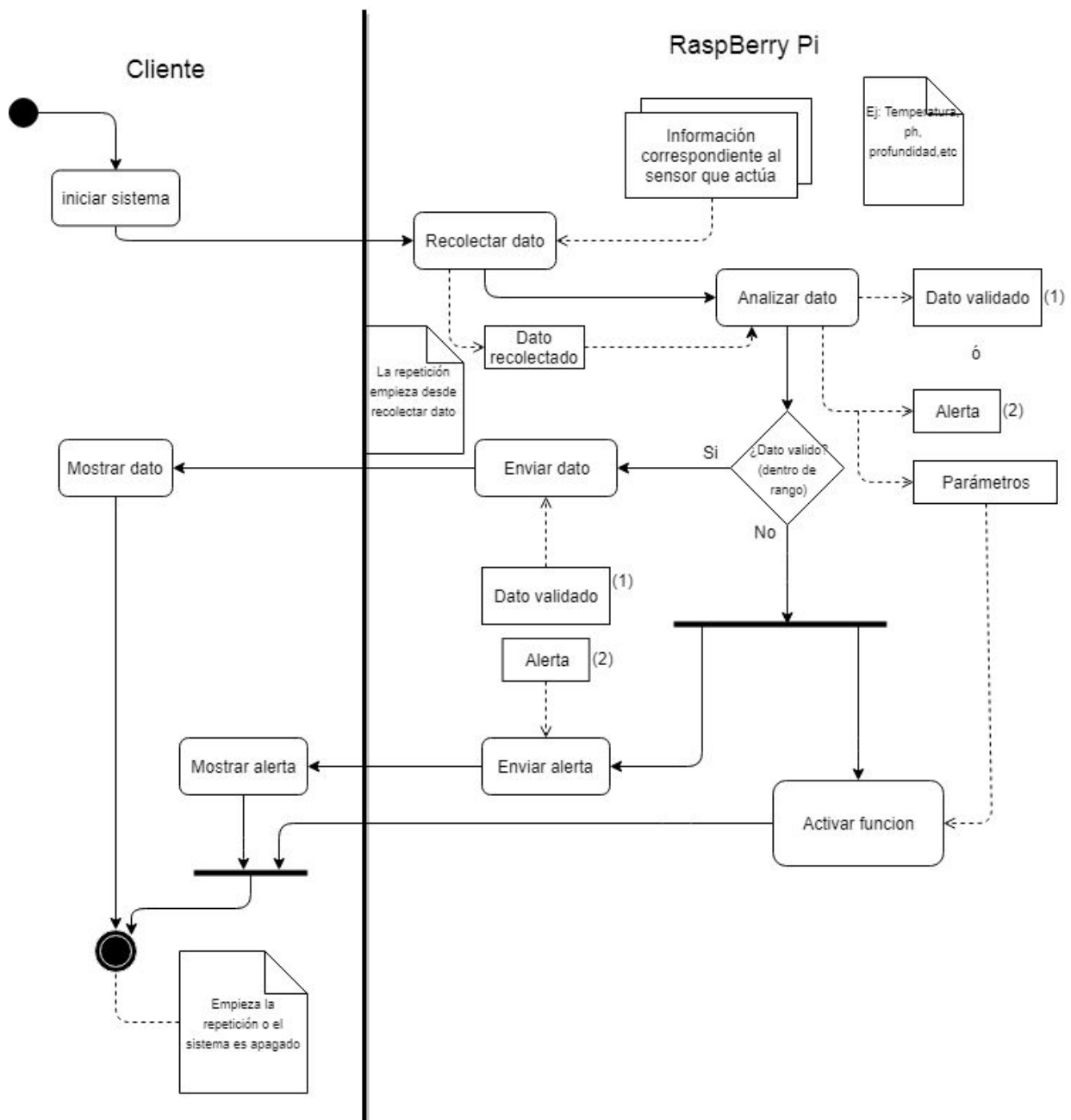


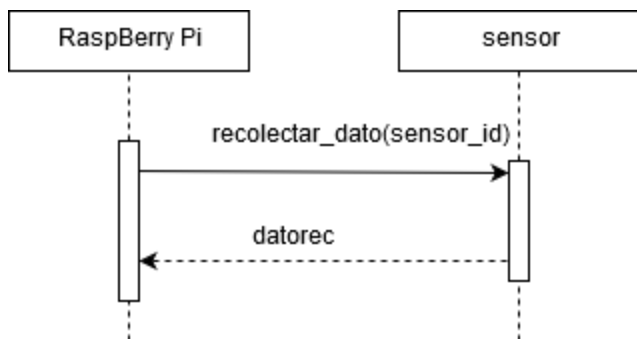
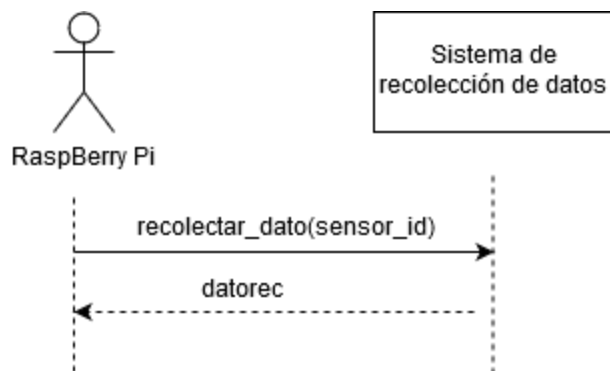
Figura 4. Diagrama de actividades general

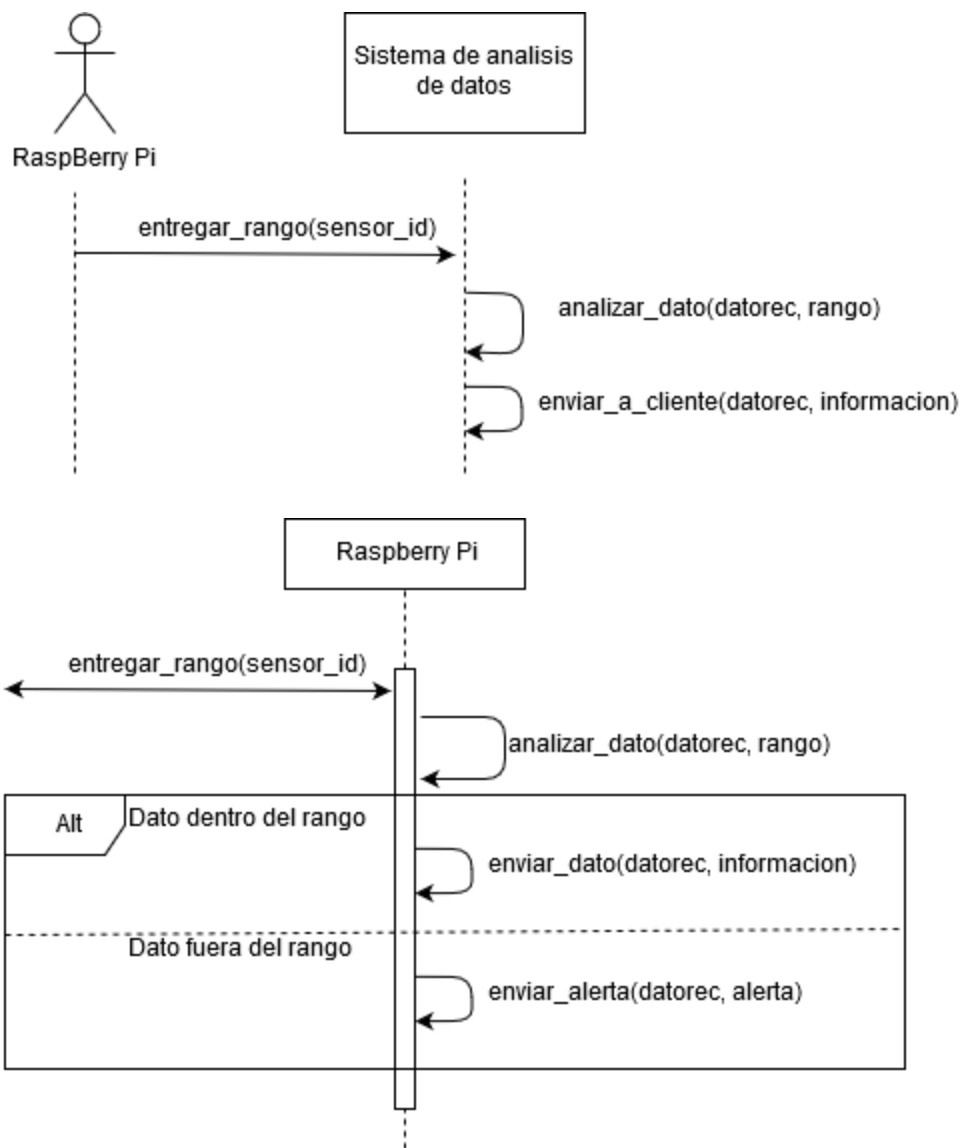
Esta figura muestra los pasos en forma de actividades que el sistema realiza

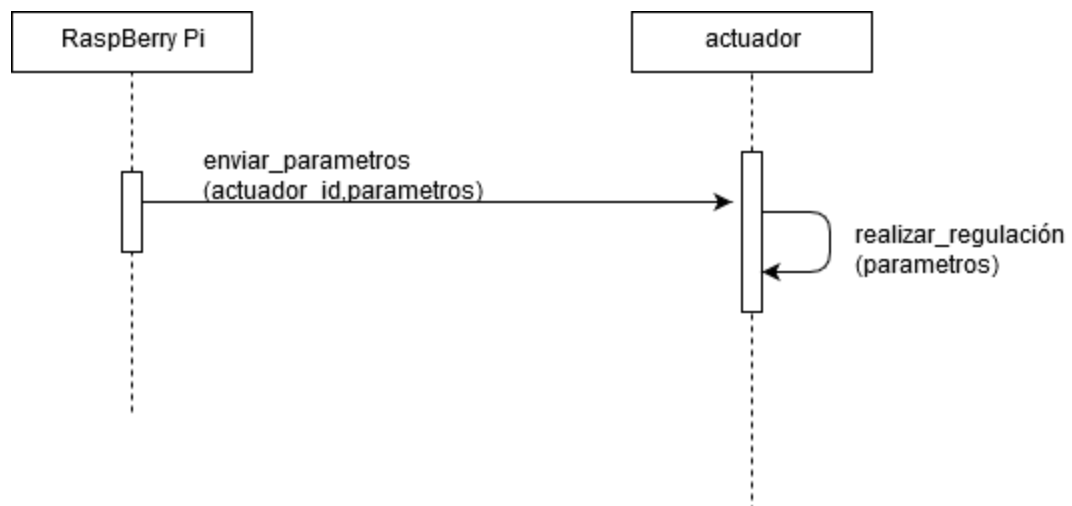
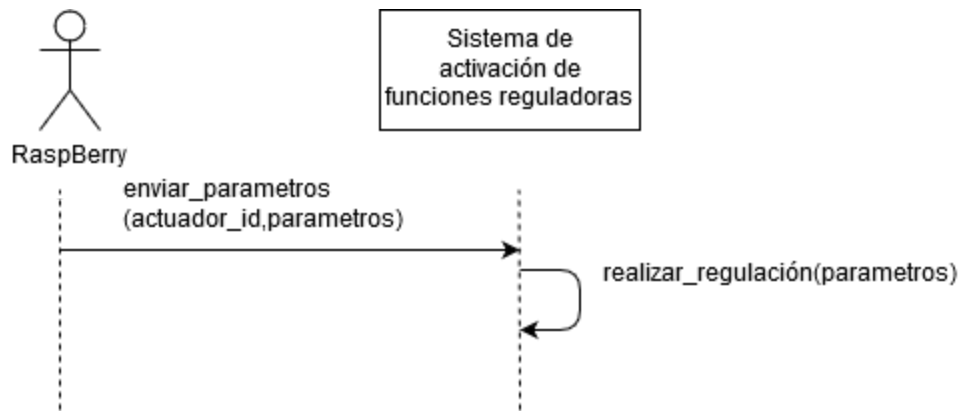
5.2.6 Diagramas de secuencia

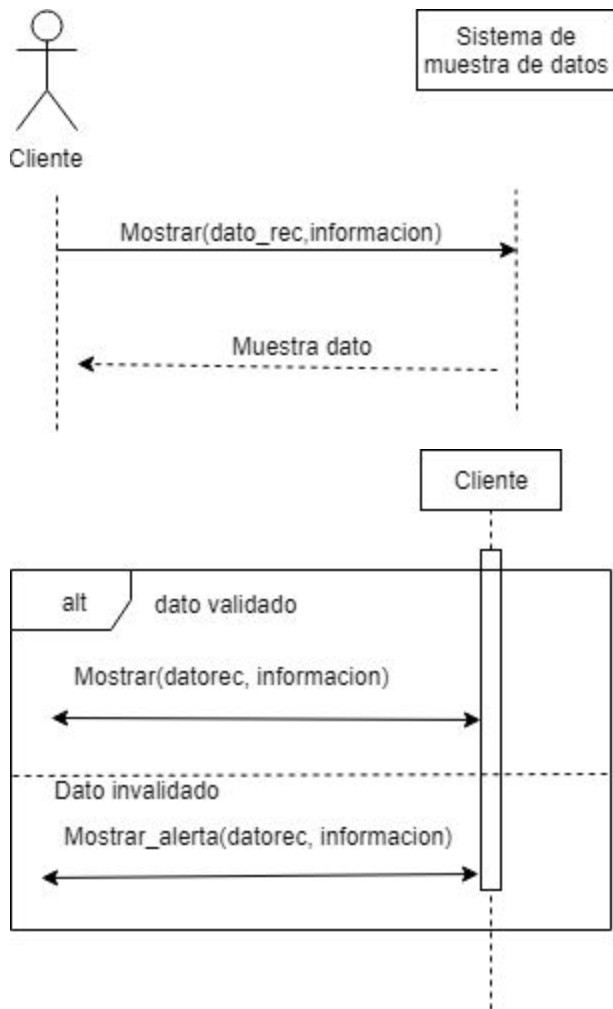
A continuación se presentan los diagramas de secuencias de nivel 0 y nivel 1 respecto a cada caso de uso.

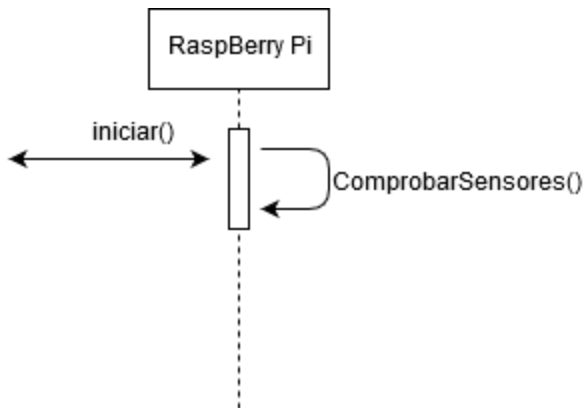
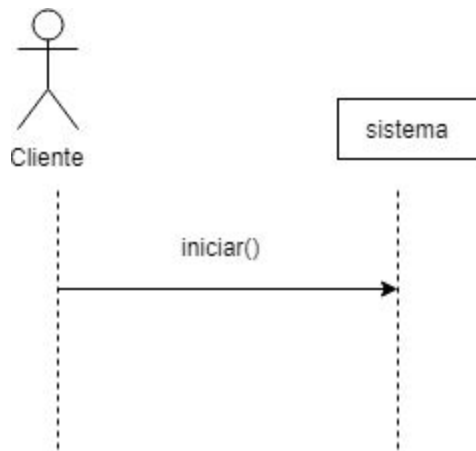
Recolectar dato:

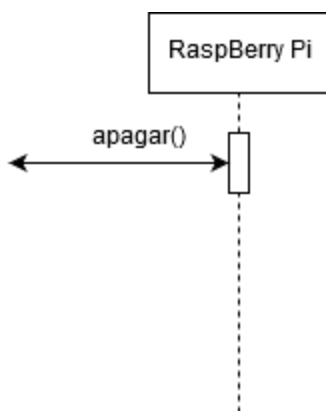
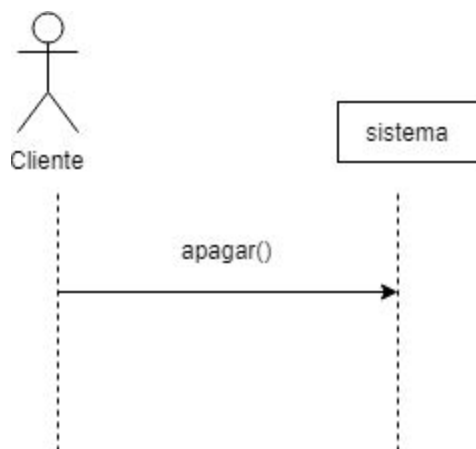


Análisis de datos:

Activar función reguladora:

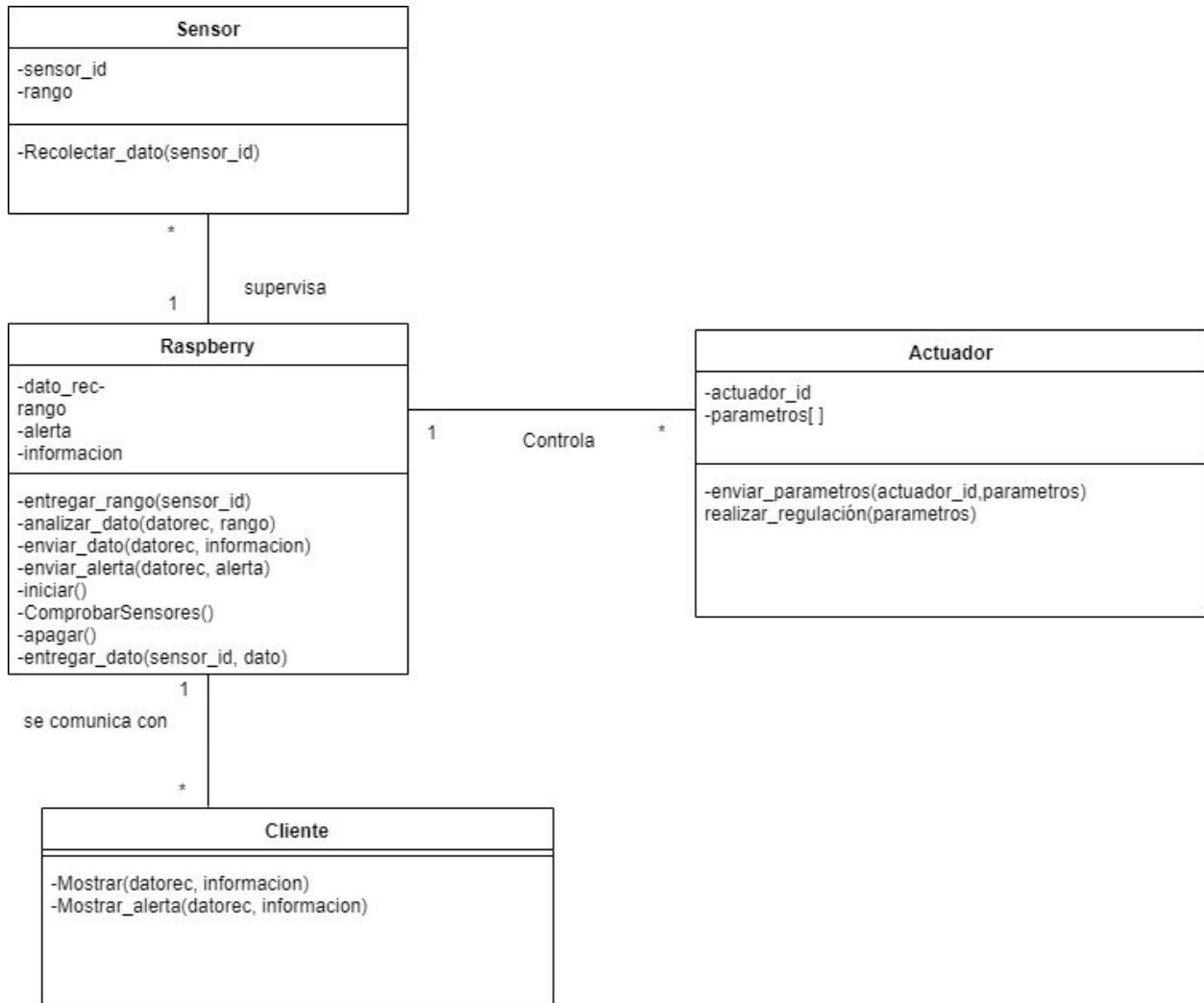
Mostrar dato:

Iniciar sistema:

Apagar sistema:

5.2.7. Diagrama de clases

A continuación se presenta el diagrama de clases a nivel de diseño.



5.2.8 Diseño de la Interfaz

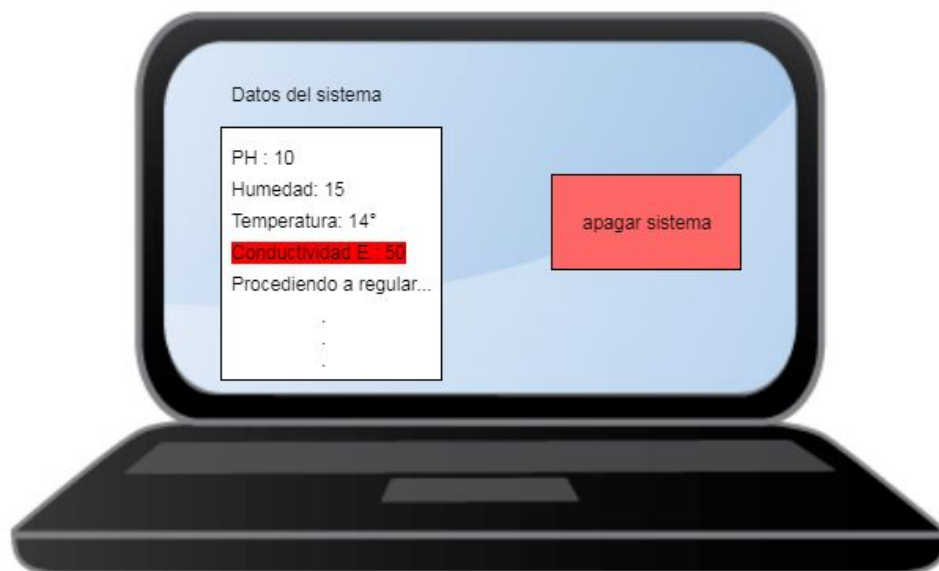
A continuación se presentan los diseños de interfaz que el usuario podrá utilizar para interactuar con el servidor (Ver datos, pedir datos).

Pantalla inicial



Se puede apreciar que en esta pantalla hay un botón que inicia el sistema

Pantalla principal



En esta pantalla el usuario podrá ver los datos del sistema en tiempo real. También podrá apagar el sistema lo cual lo llevará a la pantalla anterior y un botón "Pedir dato" el cual abrirá una pantalla con dicho nombre.

5.6. Producto del análisis

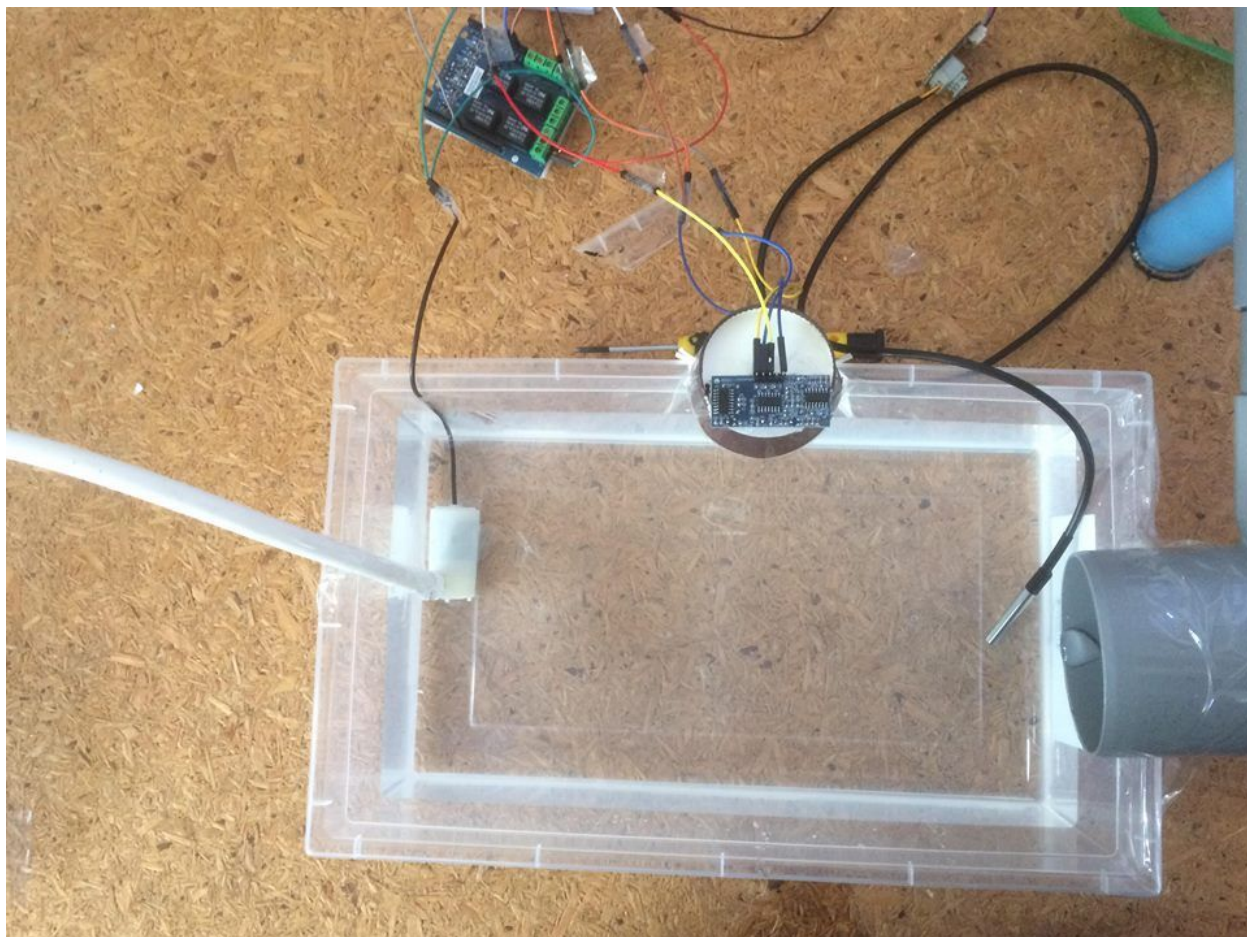
Después de este análisis y requerimientos vistos, se ha decidido por utilizar sockets para la comunicación entre el servidor(Raspberry Pi) y el cliente(PC/interfaz). Ya que utilizamos Python podemos usar la librería Threading para crear diferentes hilos, lo que permite que se pueden mandar varios datos distintos a la vez sin que estos “colisionen” entre sí.

6. Proceso de construcción de producto final

Se inició comprando los materiales especificados en el diseño y se procedió a un armado de la maqueta durante un par de semanas.



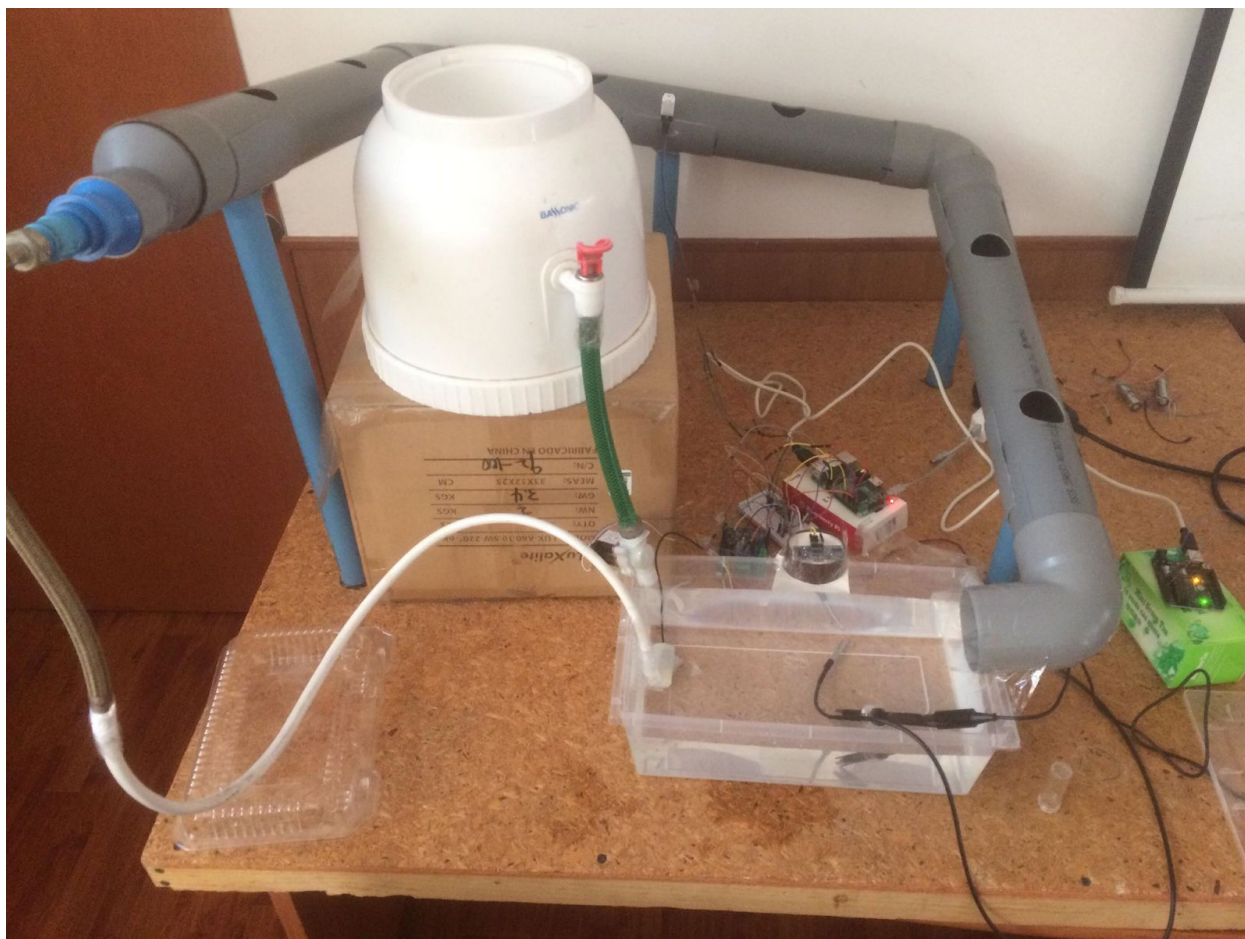
Luego se procedió a agregar un pote el cual contendría la solución.



Luego, como se ve en la imagen anterior, se agregaron las conexiones de raspberry y los sensores.

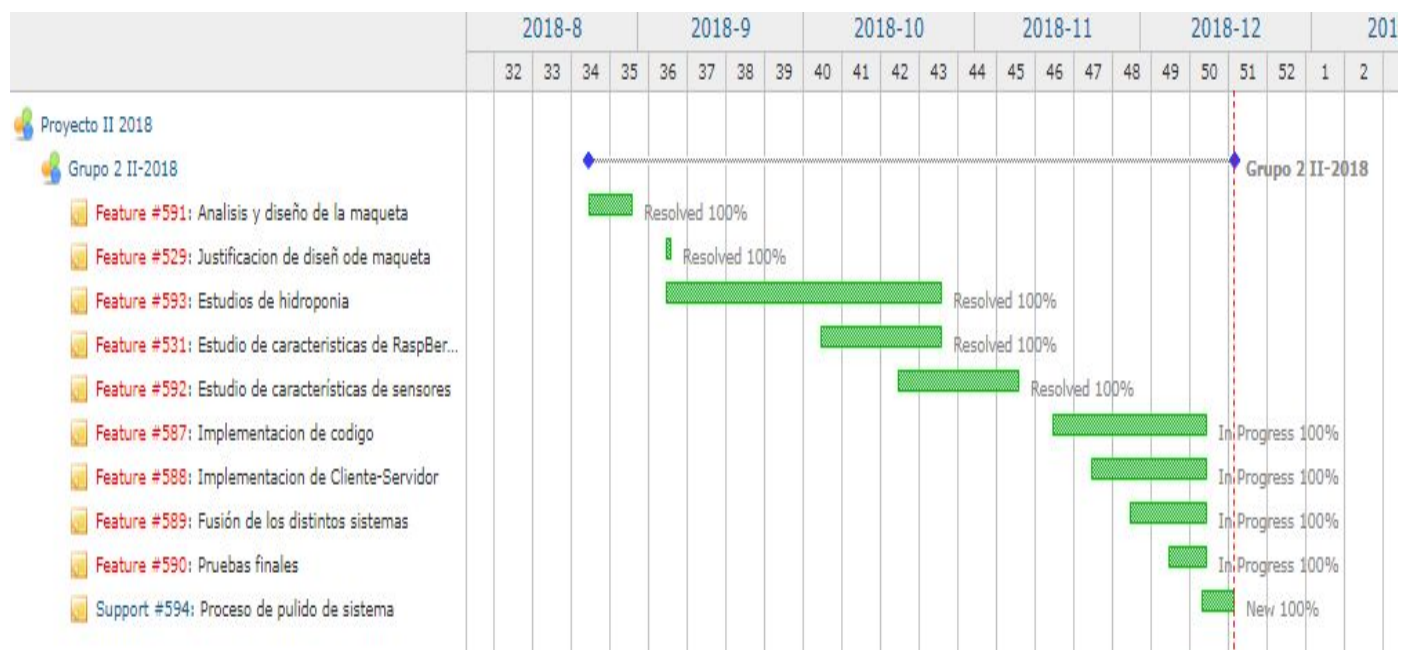


Luego fueron agregados los actuadores y sensores restantes



Como podemos ver también fue agregado el dispositivo arduino One para que los sensores de PH y EC funcionen correctamente.

7. Carta Gantt



8. Conclusión

Luego de un arduo trabajo, se ha logrado completar el objetivo final del proyecto. Siendo una de las partes más difíciles la implementación de los sensores. Se revisó los antiguos informes enviados y se revisó nuevamente el estado de la maqueta a la cual se le tuvo que hacer unas modificaciones.

En la parte del proyecto, de análisis y diseños de sistemas y arquitecturas necesarias para la realización de proyectos, se comprende de mejor forma el contenido necesario para este tipo de documentos. Se aprende nuevas forma de comunicación en este caso de Sockets. Es revisado el diseño de esta arquitectura modificando ciertos aspectos de esta.

Para la recolección de datos fue necesario ocupar hilos, así se independizó a los sensores de los demás, pudiendo trabajar con varios a la vez. Como los sensores son precisamente para sentir, no hubo problemas en su implementación, ya que la mayoría tenía una documentación disponible en internet, con lo cual se pudo idear un código para adaptarlo al sistema. También hubo problemas con los sensores de PH y EC los cuales no eran compatibles con el Raspberry Pi, por lo que se tuvo que usar el Arduino Uno para su correcto funcionamiento.

Para las comunicaciones fueron utilizados sockets, el dispositivo Raspberry Pi haciendo uso de estos monta un servidor local al cual se le fija una ip y un puerto a la cual el cliente, en este caso la interfaz, debe conectarse para ver los datos del sistema. Los sockets también tenían la suficiente documentación en internet, por lo cual no fue complicado idear un código para que se adapte al sistema.

Algunas dificultades encontradas fueron el hecho de la mediana documentación en la red sobre implementación de sensores de PH Y EC que complicaron la generación de diseños, sobre todo al estar en la mayoría para Arduino. Se debió utilizar un Arduino Uno para que funcionaran correctamente. Lo más complicado ocurrió cuando se quiso idear códigos para la regulación del sistema, dando errores complicados de resolver. También se necesitó crear algoritmos que actúan paralelamente de acuerdo a la regulación del sistema.

Una buena forma de ahorrar espacio y tierras de una forma productiva es hacer hidroponía. Utilizando menos materiales químicos y más ingenio pueden llegar a crearse sistemas tan eficaces y eficientes, más aún hoy en día en donde la tecnología abarca más y más áreas provocando como vemos en este proyecto la automatización de sistemas reduciendo la carga humana y aumentando los productos agrícolas como nunca antes.

9.Referencias

- ❖ <https://www.raspberrypi.org>.
- ❖ <https://www.ecured.cu/Socket>.
- ❖ <https://www.python.org/>.
- ❖ Intranet “Tecnología de objetos 2017”.
- ❖ Intranet taller de tecnicas de programacion.