**UNIVERSIDAD DE TARAPACÁ**



**ESCUELA UNIVERSITARIA DE INGENIERÍA INDUSTRIAL, INFORMÁTICA Y DE SISTEMAS**



Área de Ingeniería en Computación e Informática

**“Análisis y diseño del sistema”**

**Sistema Hidropónico**

**Autor(es): Fabián Guarachi**

**Patricio Tudela**

**Asignatura: Proyecto 2**

**Profesor(es): Diego Aracena**

ARICA, 13/11/2018

# **Historial de cambios**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor(es)** |
| 4/10/2018 | 1.0 | Primera versión del documento | Patricio Tudela |
| 30/10/2018 | 2.0 | Se han agregado casos de uso y descripciones. | Patricio Tudela |

# Tabla de contenidos

[**Historial de cambios**](#_d19ssuooyf07) **1**

[**Tabla de contenidos**](#_z4i5ygrwgchd) **2**

[**1.Introduccion**](#_3yc2q6uahpb1) **3**

[**2.Análisis**](#_m3lr1xkr9fc7) **3**

[2.1.Requerimientos funcionales.](#_ym5rps8r289x) 3

[2.2.Requerimientos no funcionales.](#_9fl7iwetjef4) 3

[**3.Diseño**](#_vzw52u7al2ph) **4**

[3.1.Herramientas, hardware y software utilizados](#_ofp3dgonwhvt) 4

[3.2.Diagrama de la solución](#_nivcw4p0917g) 5

[3.3.Diagrama de casos de uso de sistema](#_2o1ugs1udxob) 6

[3.3.1.Descripción de casos de uso](#_ygiy6pjq3u3l) 7

[3.3.1.1 Recolectar Dato](#_8n4lm2p2qc16) 7

[3.3.1.2 Analisis de datos](#_sesdjxeeah0o) 8

[3.3.1.3 Activar función reguladora](#_i4gww5d371c3) 9

[3.3.1.4 Mostrar datos](#_nqtiqhu7d8oh) 10

[3.3.1.5 Iniciar sistema](#_mjbvcajmxsyn) 11

[3.3.1.6 Apagar sistema](#_p8xcbx3z02uc) 12

[3.3.1.7 Pedir Dato](#_b07wisvv6jdj) 13

[3.4 Diagrama de roles](#_ky5om684xupc) 14

[3.5 Diagrama de actividades ( General )](#_9rz46pdyys15) 15

[3.6 Diagramas de secuencia](#_pxwk5v7ee3in) 16

[3.7.Diagrama de clases](#_na34yylnlr8d) 23

[3.8 Diseño de la Interfaz](#_gm247wm499x7) 24

[**4.Trabajo a Futuro**](#_vg76zj5r4yfp) **27**

[**5.Conclusión**](#_7mhic6jkgl6h) **27**

[**6.Referencias**](#_elqadpk3cuzp) **28**

# **1.Introduccion**

En este documento se presentará el análisis y diseño del sistema, cómo operarán las partes implicadas en ello. Se mostrará el diseño de la arquitectura cliente-servidor junto con el diseño de los sensores y dispositivos actuadores acoplados al sistema.

# **2.Análisis**

## **2.1.Requerimientos funcionales.**

* El sistema debe extraer datos del ambiente e hidroponia.
* El sistema supervisará 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.

## **2.2.Requerimientos no funcionales.**

Software:

* El tiempo de envío de mensajes no deberá sobrepasar los 3 segundos.
* Cada sensor tendrá un tiempo de un segundo para recolectar y enviar datos al sistema luego el turno será entregado a otro sensor.
* El sistema operativo para el servidor debe ser Raspbian.
* El sistema operativo para los clientes debe ser window 10.
* El sistema utiliza el lenguaje de programación Python.
* No debe haber intervención humana.
* El sistema deberá presentar una visualización adecuada para que cliente vea los datos recolectados de este.
* Los datos recolectados que estén fuera de rango serán catalogados como alertas.

Hardware:

* El sistema funciona mientras esté encendido.
* No debe haber intervención humana.
* El dispositivo Raspberry Pi requiere conexion a internet.

Externo:

* Se necesita un ambiente adecuado (temperatura, espacio, etc)
* Sin intervención humana.

# **3.Diseño**

## **3.1.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:**Permitira 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**:Sera la encargada de bombear el agua en el sistema.
* **Valvula de solenoide MCID0726:**Sera 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 utilizara el RB PI

## **3.2.Diagrama de la solución**

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, encargada de mostrar los datos recolectados e información del sistema como también pedir datos específicos al sistema sobre datos recientes.

Debido a esta configuración se ha optado hacer las comunicaciones entre el RaspBerry Pi y el cliente (PC) por medio de sockets.

## **3.3.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.

### **3.3.1.Descripción de casos de uso**

A continuación se describirán los casos de usos vistos en el diagrama de casos de uso

(figura 2).

#### **3.3.1.1 Recolectar Dato**

Como vimos 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 dato.  3.-Enviar dato 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.

#### **3.3.1.2 Analisis 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 restricciones correspondientes y dato. | 2.-Analizar dato.  3.-Validar dato.  4.-Enviar información 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:q**  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.

#### 

#### **3.3.1.3 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**:   |  |  | | --- | --- | | 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 parametros. | 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.

#### **3.3.1.4 Mostrar datos**

|  |  |
| --- | --- |
| **Nombre** | Mostrar datos |
| **Descripción**:  Permite mostrar la información recolectada del sistema. | |
| **Actor**:  Cliente (PC). | |
| **Precondiciones**:  El dato es analizado (Es incluido en Análisis de datos) | |
| **Flujo Normal**:   |  |  | | --- | --- | | Cliente | 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**:   |  |  | | --- | --- | | Cliente | 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. | |

#### 

#### **3.3.1.5 Iniciar sistema**

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

#### **3.3.1.6 Apagar sistema**

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

#### 

#### **3.3.1.7 Pedir Dato**

|  |  |
| --- | --- |
| **Nombre** | Pedir dato |
| **Descripción**:  Permite al cliente pedir un dato en específico | |
| **Actor**:  Cliente ( PC ). | |
| **Precondiciones**:  Los sensores deben estar en correcto funcionamiento. | |
| **Flujo Normal**:   |  |  | | --- | --- | | Cliente | Sistema pedir dato | | 1.-Entrega tipo y dato requerido | 2.-Valida dato  3.-(Incluye C.U. Recolectar dato)  4.-Muestra dato |   **Flujo alternativo**:   |  |  | | --- | --- | | Cliente | Sistema de Pedido de datos | |  | 2.-Dato invalido  3.-Requiere ingreso de dato. | | |
| **Pos Condiciones:**  El dato es mostrado exitosamente. | |

## 

## **3.4 Diagrama de roles**

## 

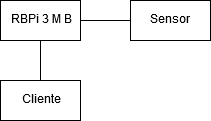


Figura 3. Diagrama de roles.

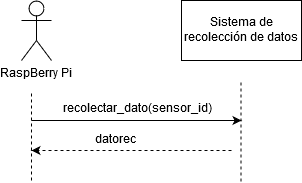
## **3.5 Diagrama de actividades ( General )**

Figura 4. Diagrama de actividades general

## **3.6 Diagramas de secuencia**

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

**Recolectar dato**:



## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

**Análisis de datos:**

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

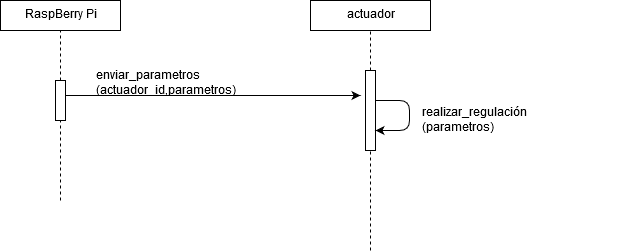
## 

**Activar función reguladora:**

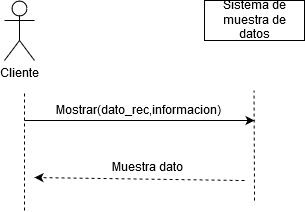
## 

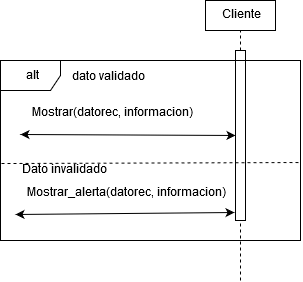
## 

## 



**Mostrar dato:**





## 

**Iniciar sistema:**

## 

## 

## 

## 

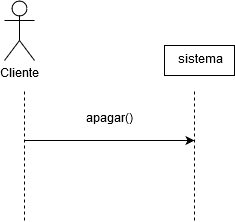
## 

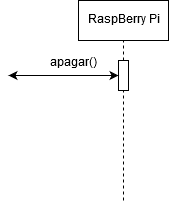
## 

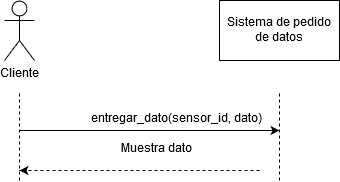
## 

## 

**Apagar sistema:**





**Pedir dato:**

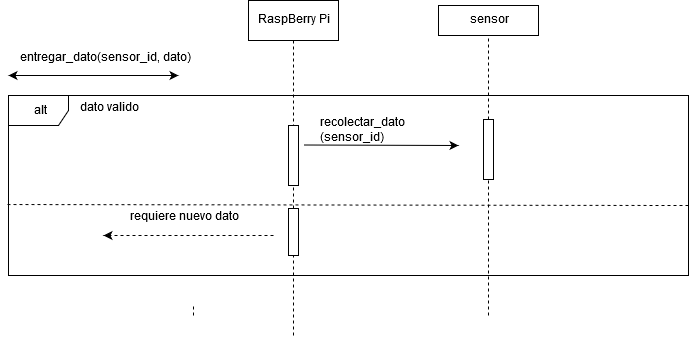
## 

## 

## 

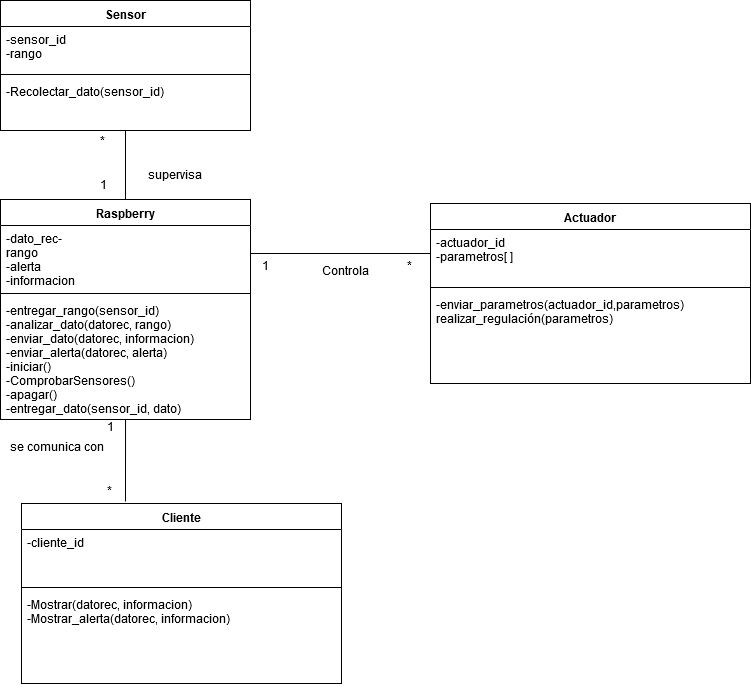
## 

## 



## **3.7.Diagrama de clases**

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



## 

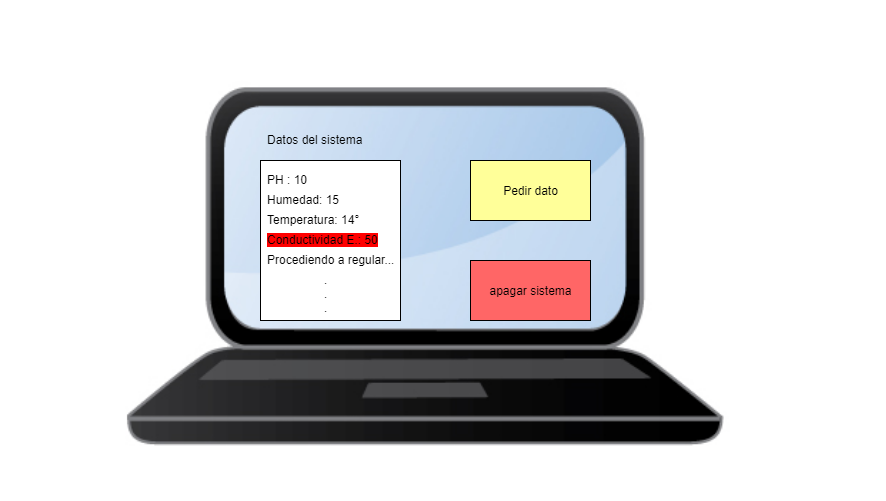
## **3.8 Diseño de la Interfaz**

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

Pantalla inicial

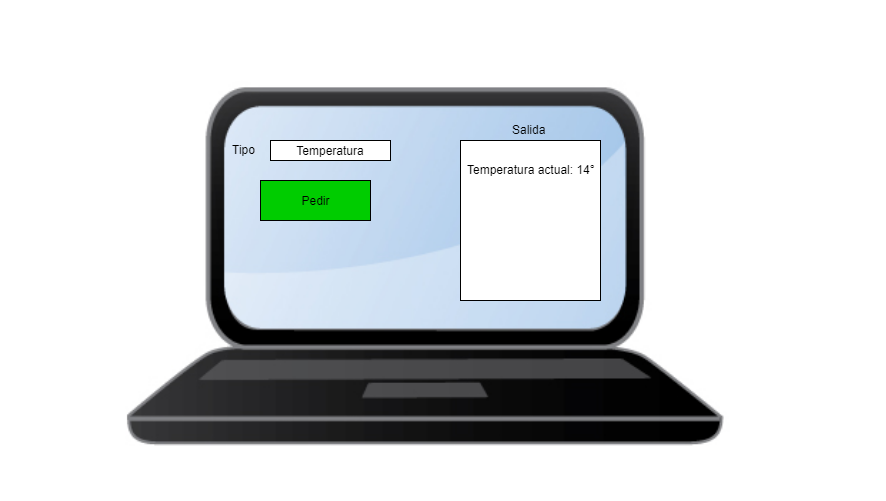
Se puede apreciar que en esta pantalla hay un botón que inicia el sistema

Pantalla principal



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

Pantalla “Pedir dato”



En esta pantalla el usuario podrá ver la información que él elija, especificando el tipo de información que quiere utilizando las palabras clave de estos (Temperatura, Ph, EC, etc)

# **4.Trabajo a Futuro**

Actividades a realizar a futuro:

* Implementación de código del RaspBerry Pi
* Implementación de interfaz gráfica.
* Implementación de la arquitectura cliente-servidor.
* Implementar base de datos
* Acoplar distintas partes.
* Pruebas con maqueta.
* Pruebas finales.

# **5.Conclusión**

En esta parte del proyecto nuevamente se lleva a cabo el 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.

Algunas dificultades encontradas fueron el hecho de la mediana documentación en la red sobre implementación de sensores que complicaron la generación de diseños, sobre todo al estar en la mayoría para Arduino. Se debió idear formas de traducir el código para su utilidad en el dispositivo RaspBerry Pi para realizar un correcto diseño para este. También hubo dudas de que sistema ocupar para el almacenamiento de los datos, optando por dejar este aspecto para el siguiente paso.

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

# **6.Referencias**

* https://www.raspberrypi.org
* https://www.ecured.cu/Socket
* https://www.python.org/
* Intranet “Tecnologia de objetos 2017”