

Sistema de riego automatizado y distribuido con control web

AUTOR: FRANCISCO JAVIER SOLÍS FRANCO

TUTOR: ÁNGEL FRANCISCO JIMÉNEZ FERNÁNDEZ

COTUTORA: ELENA CEREZUELA ESCUDERO

GRADO EN INGENIERÍA INFORMÁTICA-INGENIERÍA DE COMPUTADORES

Contenido

1- Introducción y Objetivos.

2- Arquitectura.

3- Diseño Hardware.

4- Implementación Firmware.

5- Software.

6- Planificación y Costes.

7- Conclusiones.

8- Trabajo Futuro.

1- Introducción y Objetivos

Introducción:

- Agricultores con riegos automáticos pero con programación manual.
- Impacto medio ambiental por los consumos no controlados y los desplazamientos.
- Sequías por el cambio climático y pozos.
- Riego innecesario en caso de lluvia o humedad de la tierra suficiente.
- Entrada al IoT dotando de conectividad el sistema de riego.

1- Introducción y Objetivos

Objetivos primarios:

- Sistema empotrado y distribuido de riego automático.
- Obtención de datos (Temperatura, Humedad, consumo de agua, etc...).
- Estadísticas
- Uso de plataformas libres.

1- Introducción y Objetivos

Objetivos secundarios:

- Controlador de riego con Raspberry pi.
- Firmware para Arduino para la comunicación con Rpi.
- Base de datos.
- Control mediante Web.

1- Introducción y Objetivos

Tecnología usada:

- Arduino
- Raspberry Pi.
- Sensores y actuadores.
- Python.
- MySQL.
- CodeIgniter.
- Apache.
- Raspbian.
- Eclipse.
- MySQL WorkBench.

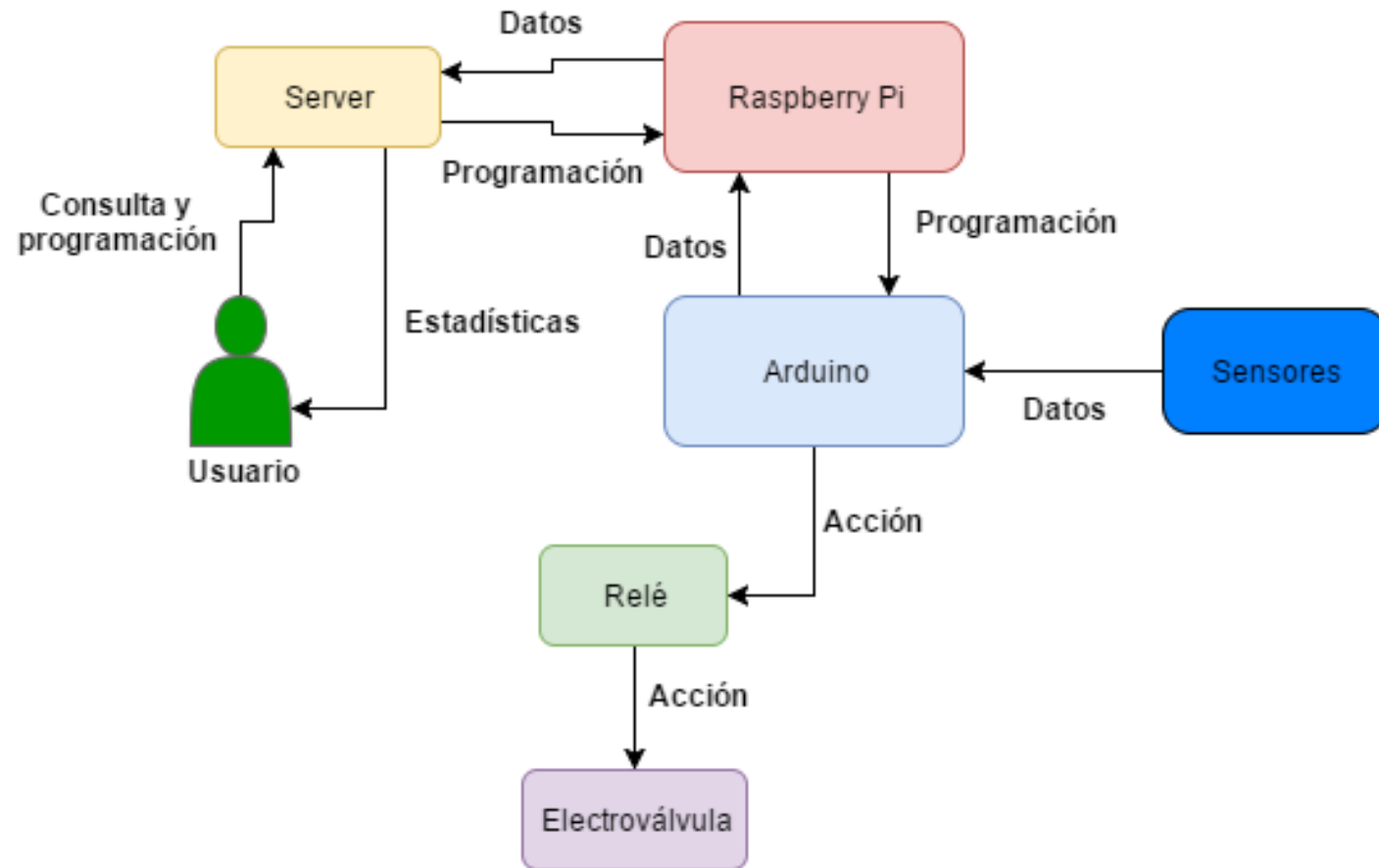
2- Arquitectura

Cliente-Servidor



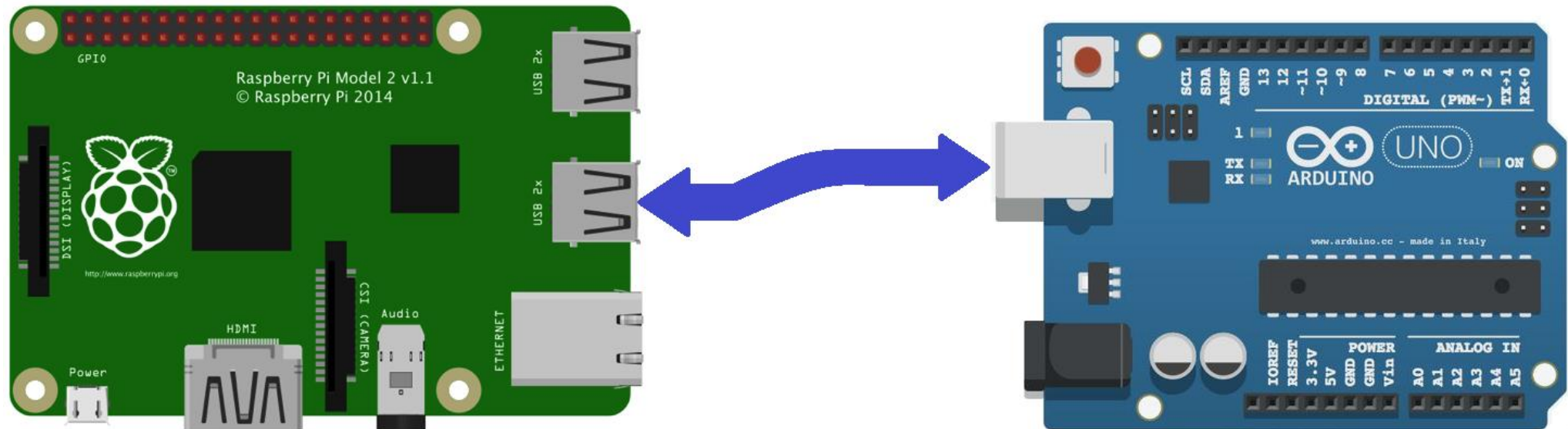
2- Arquitectura

Sistema completo



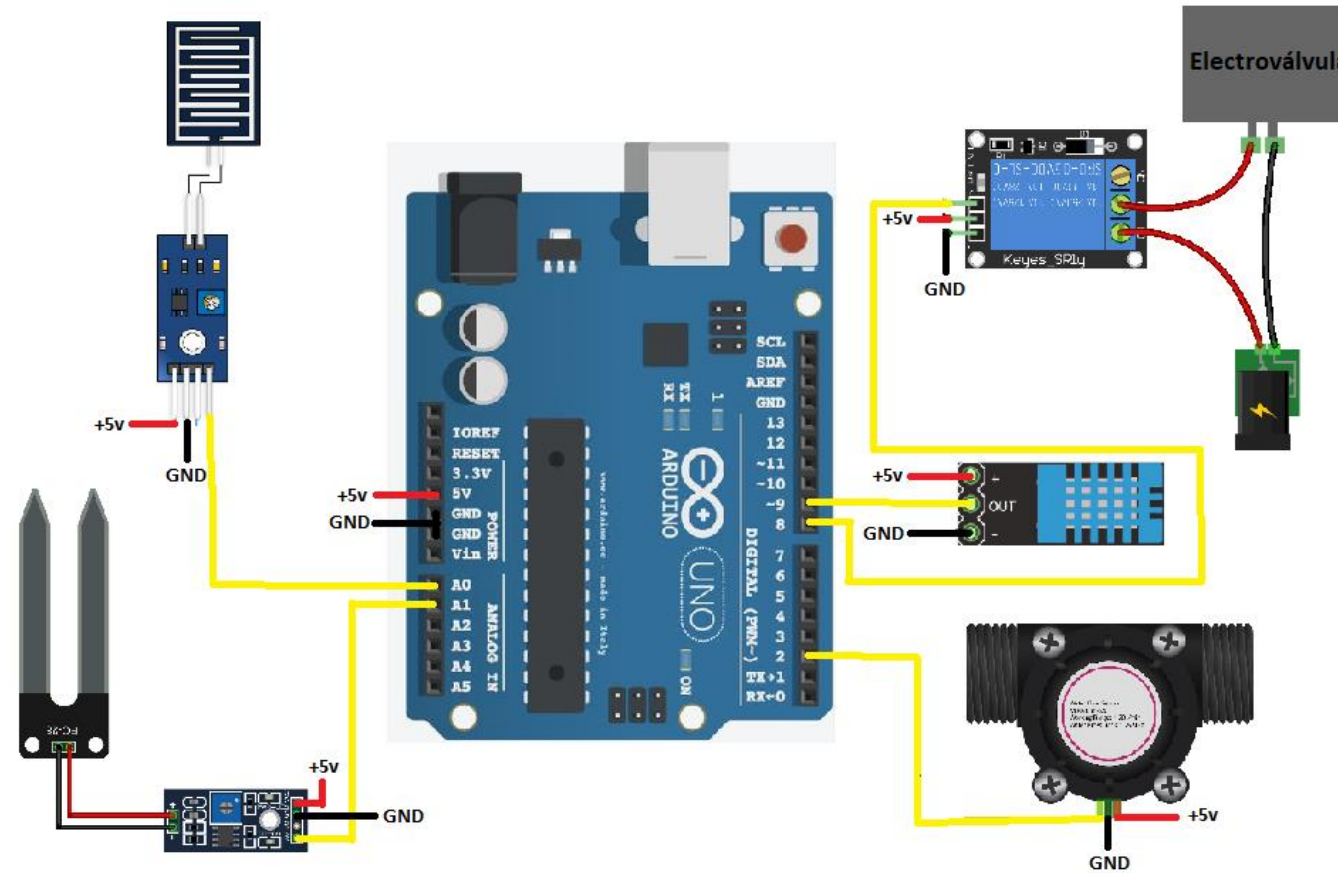
3- Diseño Hardware

Raspberry Pi



3- Diseño Hardware

Arduino



3- Diseño Hardware

Sensores y Actuadores

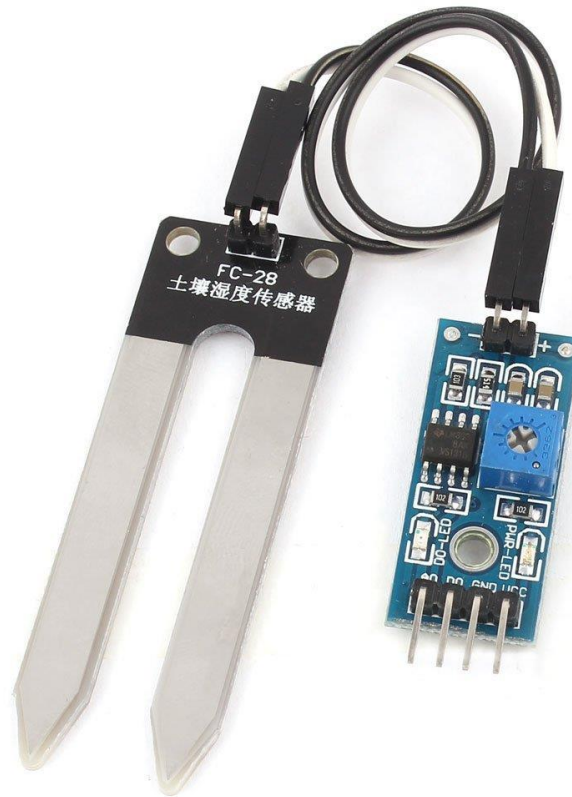
FC-37



3- Diseño Hardware

Sensores y Actuadores

FC-28



3- Diseño Hardware

Sensores y Actuadores

Relay



3- Diseño Hardware

Sensores y Actuadores

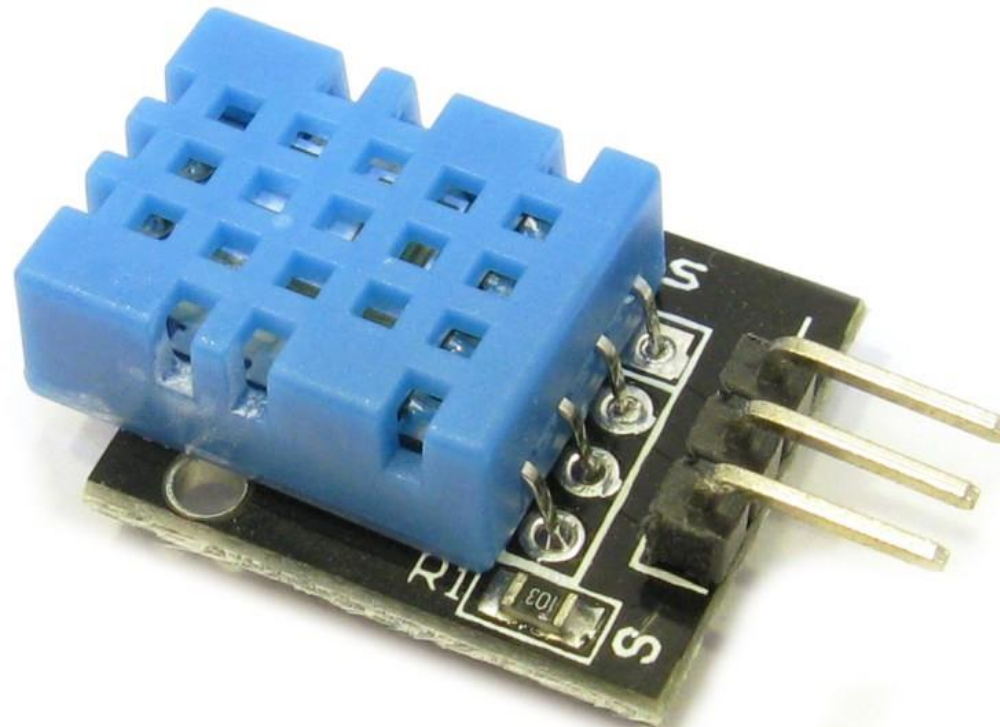
YF-S201



3- Diseño Hardware

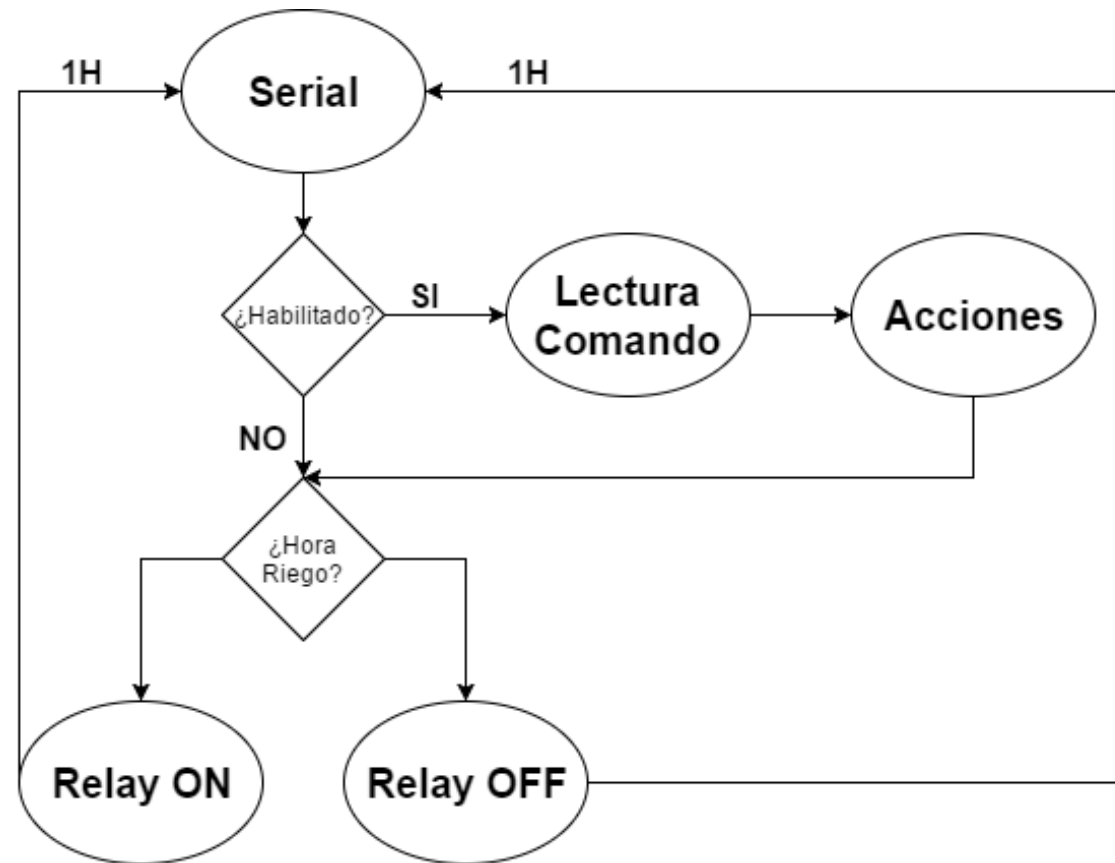
Sensores y Actuadores

DHT-11



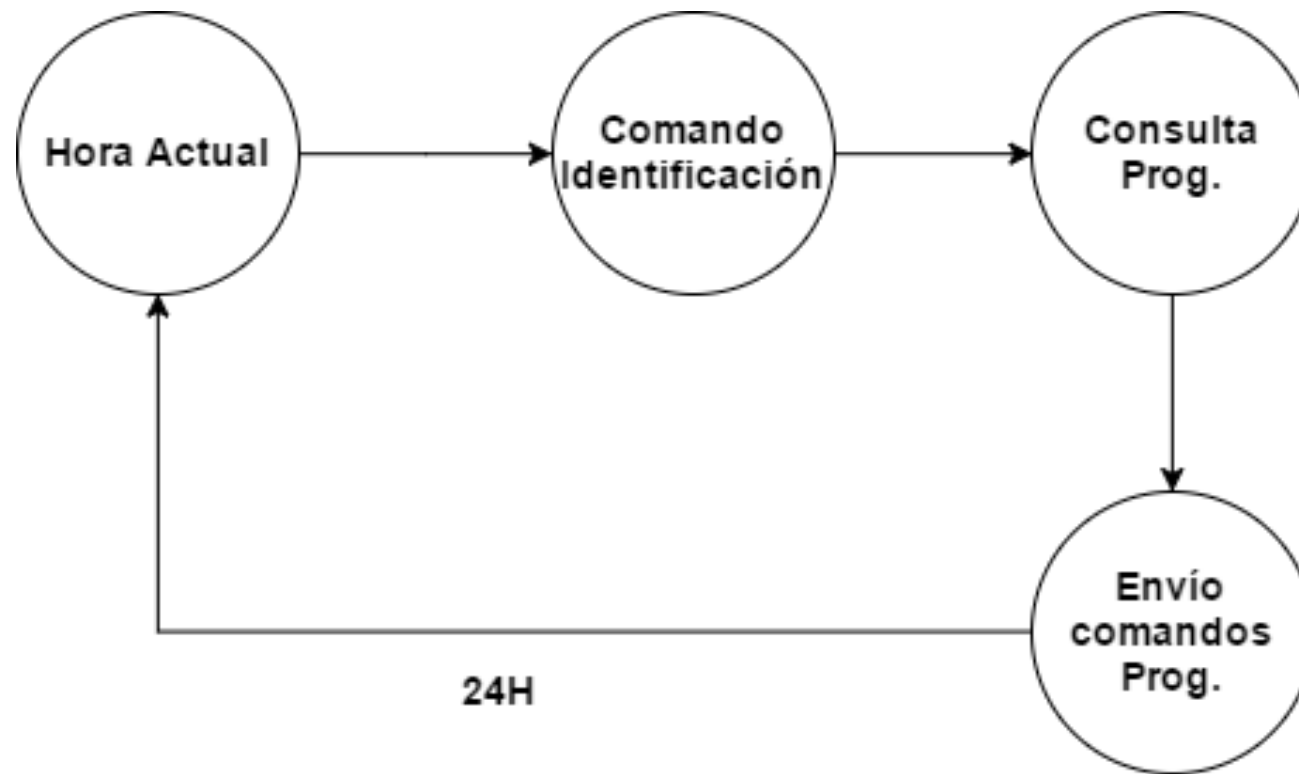
4- Implementación Firmware

Protocolo Arduino



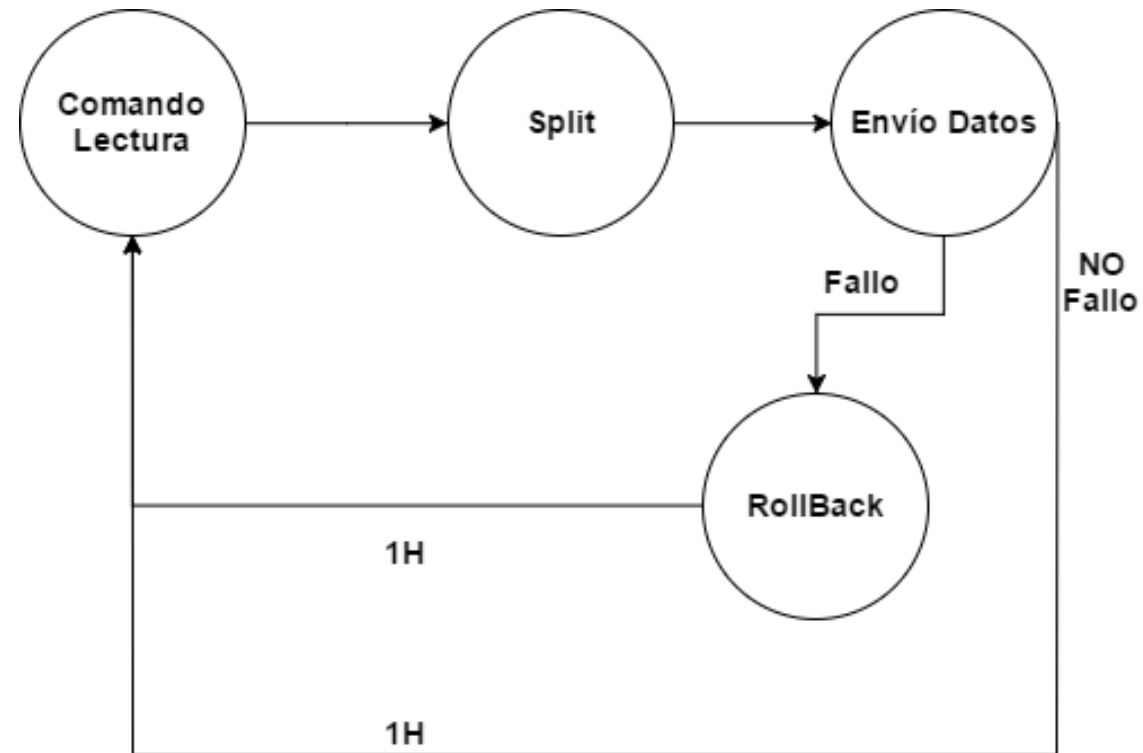
4- Implementación Firmware

Protocolo Raspberry (Configuración)



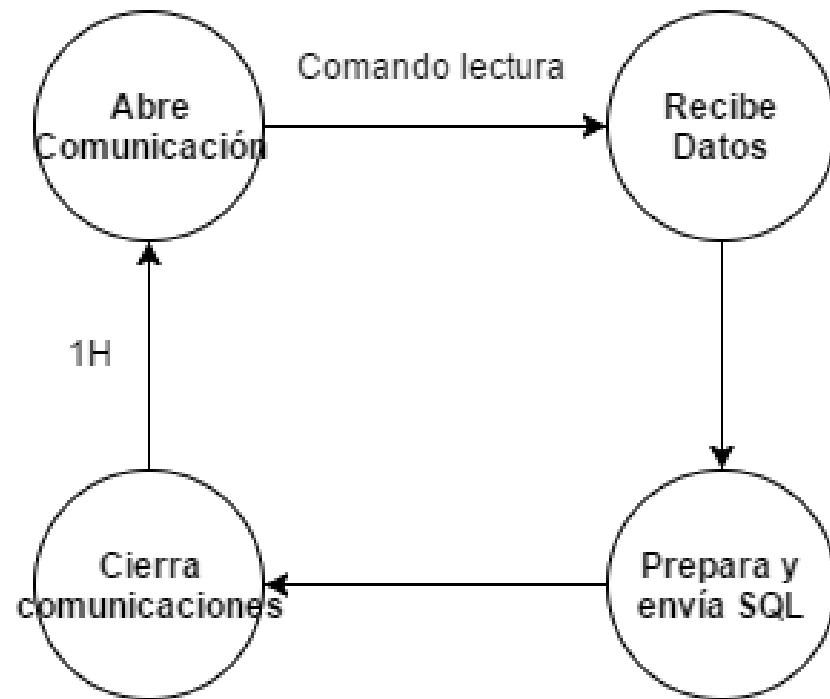
4- Implementación Firmware

Protocolo Raspberry (Inserción de datos)

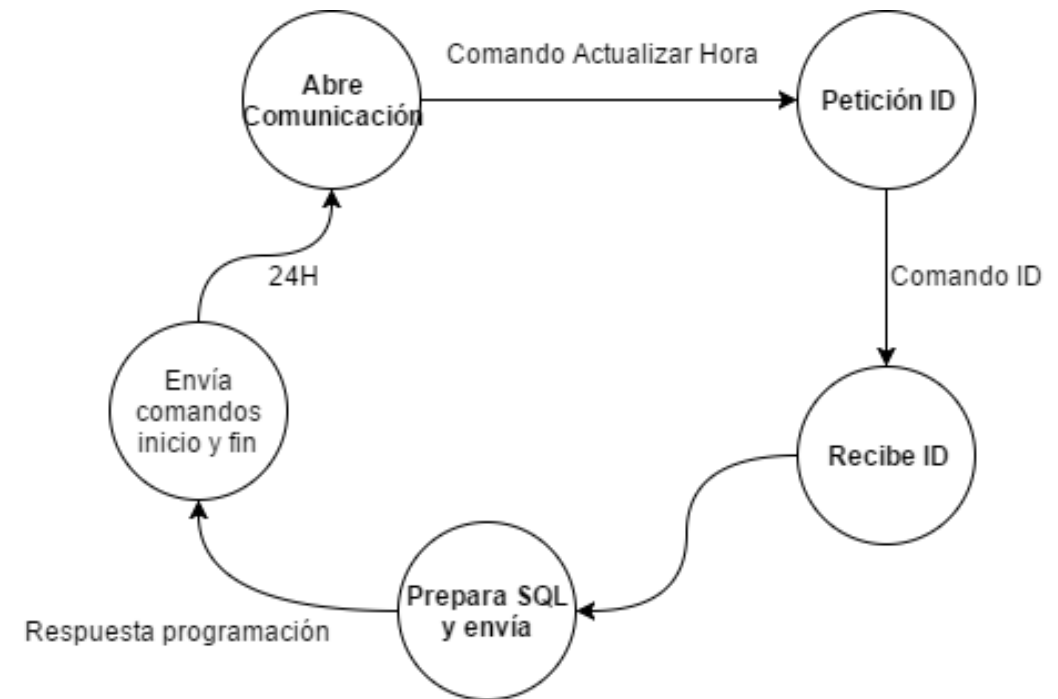


5- Software

Python:



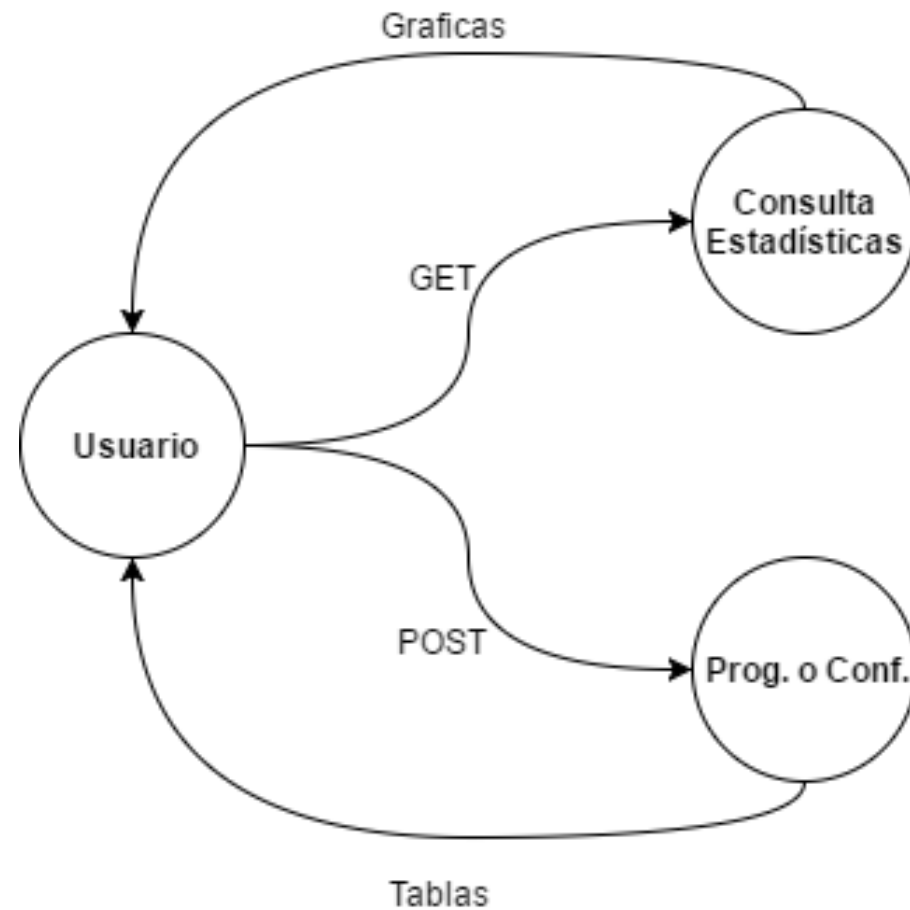
Cada Hora



Cada 24H

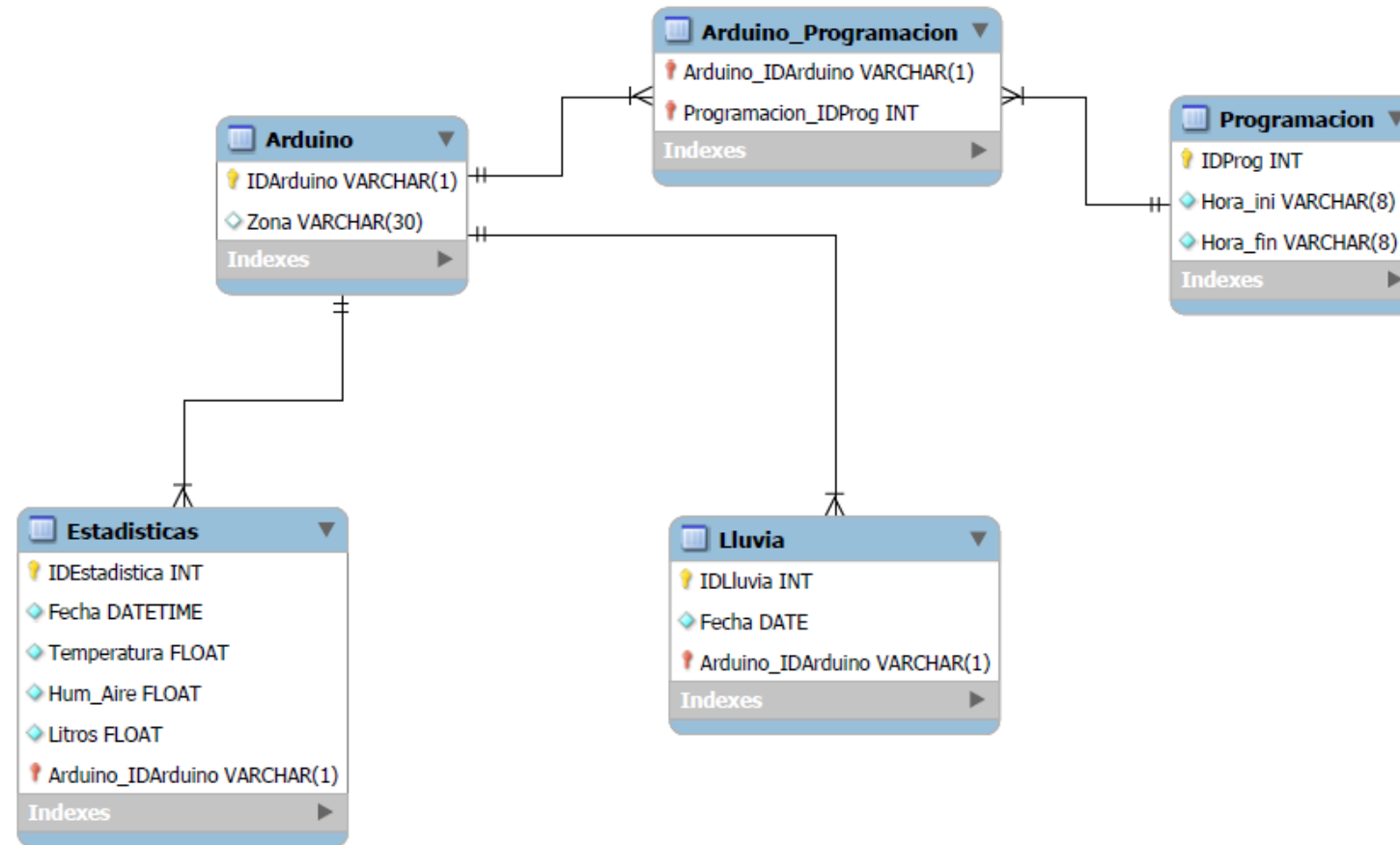
5- Software

API:



5- Software

Base de Datos:



5- Software

Interfaz de usuario:

[TFG](#) [Estadísticas](#) [Programación](#) [Configuración](#)

Programación de las horas de riego

1: Tomates->Inicio: 00:00->Fin: 00:00
2: Olivos->Inicio: 00:00->Fin: 00:00
3: Viña->Inicio: 00:00->Fin: 00:00
4: Huerto->Inicio: 00:00->Fin: 00:00

ID: 1 ▼

Inicio:

Fin:

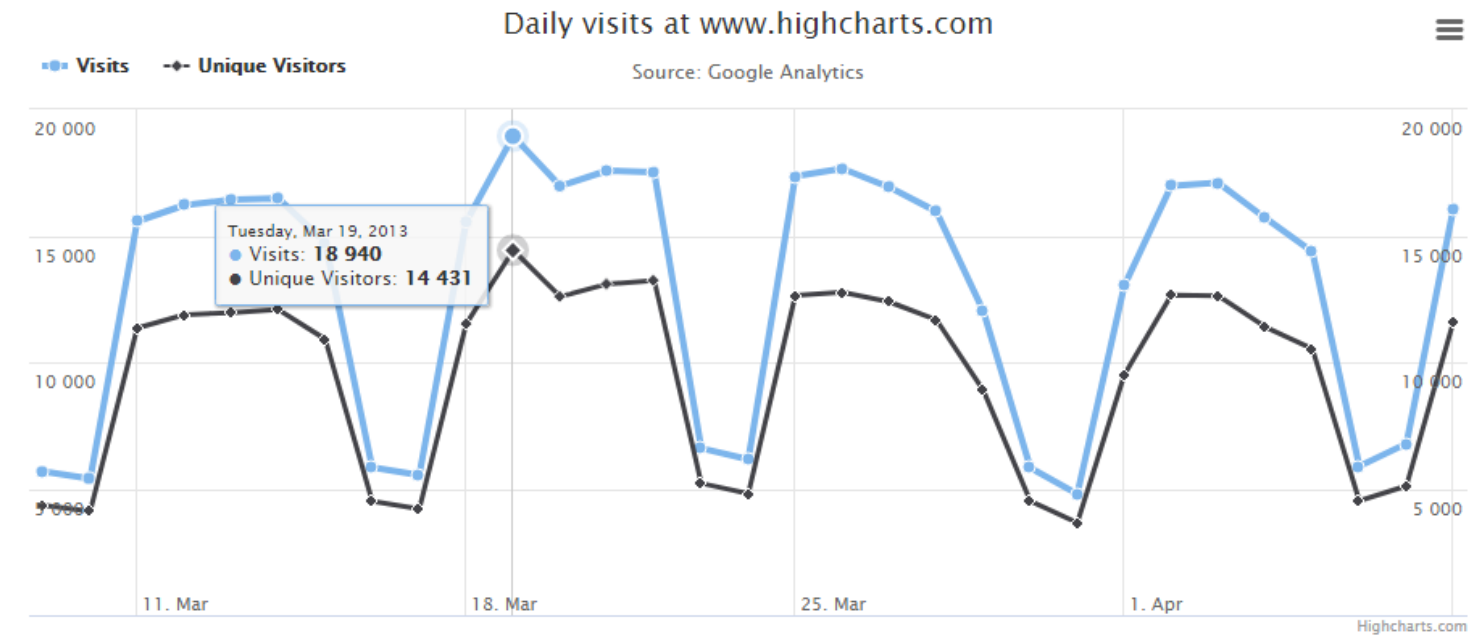
Guardar

© copyright 2017 - TFG realizado por Francisco Javier Solís Franco, todos los derechos reservados.

BootStrap, JS, PHP

5- Software

Interfaz de usuario:



HIGHCHARTS

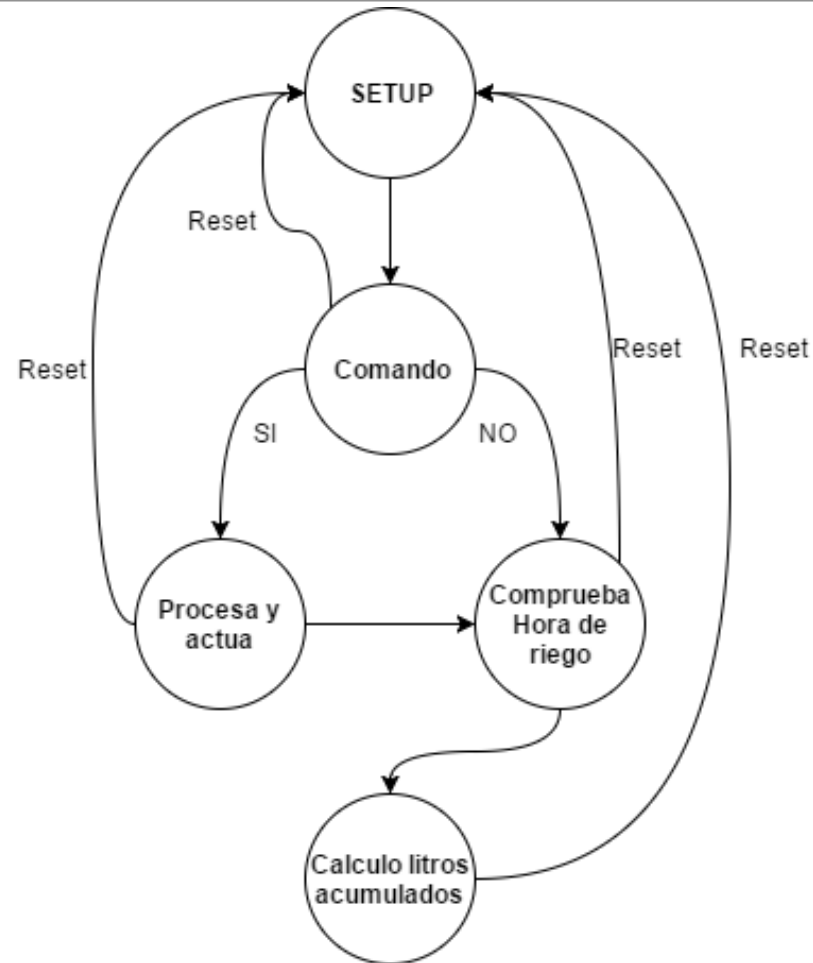
5- Software

Apache:

- Configuración Puerto
- PHP 5.0
- Web y API en htdocs (CodeIgniter)

5- Software

Arduino:



6- Planificación y Costes

Tareas:

- Definición del sistema.
- Tecnología a usar.
- UML.
- Programación BBDD.
- Testeo y errores BBDD.
- Programación y testeo individual del hardware.
- Programación conjunta.
- Testeo programación conjunta.
- Web.
- Comunicaciones.
- Integración con Hardware.
- Pruebas unitarias.
- Pruebas de campo.
- Corrección de errores.
- Documentación.
- Prototipo.

6- Planificación y Costes

Gantt: Planificador del proyecto

Resaltado del período 1

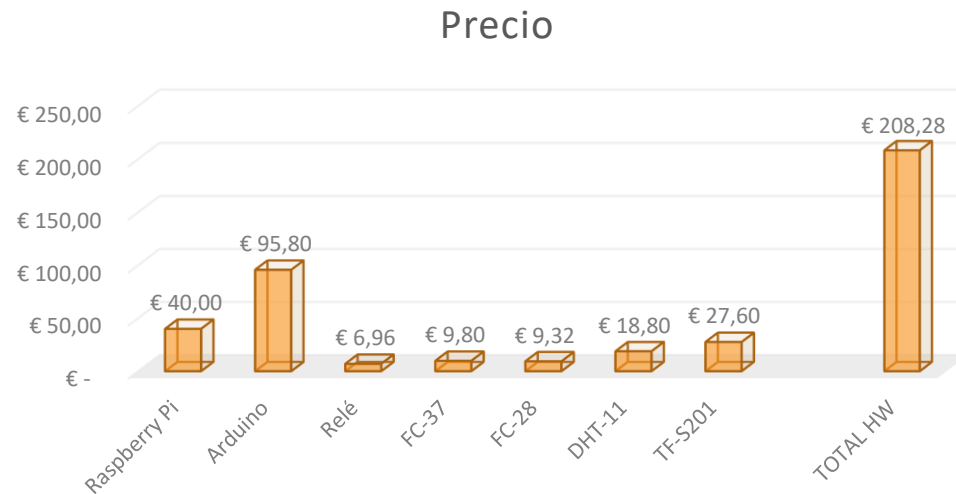
Plan



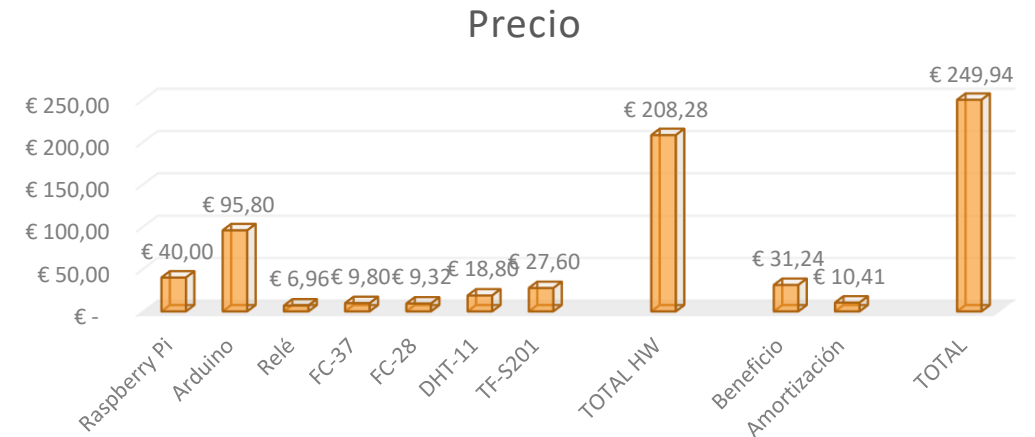
300H

6- Planificación y Costes

Costes:



Coste Prototipo completo



Coste con beneficio y amortización

7- Conclusiones

- Suplir las carencias de riegos baratos frente a fallos.
- Control remoto.
- Uso de tecnologías y plataformas de código abierto y libre.
- Control de los periodos de lluvia.
- Tecnología inalámbrica.
- Red de nodos.
- Retrasos en la planificación al usar distribuidores Asiáticos.

8- Trabajo Futuro

- Registro de lluvias.
- Red de nodos.
- Control por usuario.
- App (Smartphone) y BOT.
- Mejora de las estadísticas.(Interfaz)

DEMOSTRACIÓN y PREGUNTAS.