Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

DISEÑO Y DESARROLLO DE UNA FUENTE DE JARDÍN INTELIGENTE PROGRAMABLE

Proyecto Final de Carrera

Grado en Ingeniería Informática

Autor: Carlos Quer Barberá

Director: Jose Luis Poza Luján

Septiembre 2014

Agradecimientos

Podría dedicar este apartado de manera individual a un gran número de personas, pero mucho me temo que me quedaría corto…

Quisiera comenzar agradeciendo en primer lugar a todos los compañeros del proyecto ArduEntorno, pues ha habido bastante contacto y ayuda entre nosotros. También al propio Jose Luis Poza, por tratar de llevarnos en todo momento para ir por el buen camino con sus consejos, ideas y, sobretodo, su predisposición para ayudarnos.

No podría tampoco olvidarme de *Imagina Group* por darme la oportunidad de comenzar mi carrera profesional, así como proporcionarme todas las facilidades posibles para permitirme desarrollar el proyecto.

Por otro lado, agradecer a toda mi familia y amigos por estar siempre ahí, siempre apoyándome y siempre aguantándome, con todo lo que ello conlleva. Sin este apoyo habría sido sencillamente imposible desarrollar el proyecto.

Resumen

Se quiere desarrollar una Fuente Inteligente controlada gracias a la placa Arduino. Dispondrá de una serie de componentes que podrán controlarse tanto manual como automáticamente, manipulando los diferentes parámetros según se adapten a las necesidades del usuario u optar por dejar esta operación a Arduino respectivamente.

Este automatismo será llevado a cabo gracias a la conexión del microcontrolador a un servidor meteorológico externo, mientras que la interacción del usuario con esta fuente se realizará mediante una Interfaz de Usuario desarrollada mediante Processing

Palabras clave: Arduino, Ethernet Shield, fuente inteligente programable, Ethernet, sensores y actuadores,

Abstract

It is wanted to develop an Intelligent Fountain controlled by an Ardruino board. It will have some components which will can be controlled manual or automatically, manipulating its different parameters in the way they adapt to the user's necessities, or let this option to Arduino.

This automatism will be done thanks to the connection between the microcontroller to an external meteorological server, whereas the interaction between the user and this fountain will be done with an UI developed with Processing.

Keywords: Arduino, Ethernet Shield, intelligent programmable fountain, Ethernet, sensors and actuators.

Contenidos

Figuras

Tablas

Códigos

1. Introducción

1.1 Motivación

La realización de este proyecto está motivada por diversos motivos de índole diversa que se exponen a continuación.

El Trabajo de Fin de Grado o TFG es la primera oportunidad de un alumno de realizar un proyecto de envergadura estrechamente ligado con su campo. Tras cursar la modalidad de Computación, se ha optado por realizar este proyecto, de una orientación relativamente alejada a dicha modalidad, con la intención de poder adquirir conocimientos de otras áreas. De esta manera se trata de ver cuáles son los campos de mayor agrado momentos antes de salir al mundo profesional

Por otro lado, Arduino ha demostrado ser una plataforma muy potente, sencilla de utilizar y de coste muy ajustado. Una herramienta tan poderosa no puede pasar desapercibida, de manera que se ha optado por la realización de un proyecto estrechamente ligado a mencionada plataforma.

1.2 Objetivos

El objetivo fundamental del proyecto aquí presentado es la realización de un elemento decorativo diferente a todo lo visto con anterioridad. Así, se ha tratado de unir dos campos que aparentemente no tienen demasiado en común, como son la Informática y la Decoración.

El resultado esperado es una fuente de jardín inteligente programable, que base su apariencia en la obtención de diferentes parámetros. Estos parámetros serán recibidos principalmente de dos maneras. Por un lado, existe la posibilidad de que sea el propio usuario el que los introduzca por él mismo, adaptándola a las necesidades de un momento concreto, mientras que por otro lado se puede optar por la recepción automática de los parámetros mediante la conexión a Internet a un servidor meteorológico.

Además, esta fuente de jardín inteligente estará conectada a un ordenador, de manera que se presentarán los resultados y, en definitiva, se realizará la comunicación entre la fuente y el usuario mediante una interfaz intuitiva y sencilla de utilizar.

1.3 Descripción del Documento

A continuación se explicará el planteamiento seguido para la realización del proyecto, indicando los pasos que se irán llevando a cabo para la presentación del mismo en el presente documento.

Tras la presente introducción se llevará a cabo un análisis del Estado de la Cuestión. De esta manera, se comparará el producto resultante con el resultado de otros proyectos similares, o que guardan determinados elementos en común. Por tanto, se llevará a cabo un estudio de los mencionados proyectos con el fin de poder ubicar el aquí expuesto, tratando de formar un marco sobre el que pueda moverse. Dicho análisis incluirá comparativas entre resultados objetivos, tanto cualitativos como cualitativos. Se hará, además, un estudio detallado de las diferentes tecnologías que se vean involucradas para el desarrollo y terminación del producto.

A continuación se presentará la especificación de requisitos. Cabe señalar que se ha optado por la realización de la misma siguiendo el Estándar IEEE830. De este modo, quedarán bien claras qué tareas podrá llevar a cabo el proyecto aquí desarrollado, incluyéndose todos los detalles de

interés. Para ello, se realizará el análisis detallado de los Requisitos Funcionales y No Funcionales del proyecto, quedando asimiladas qué acciones podrá realizar la fuente y cuáles no. Esta tarea será llevada a cabo mediante tablas y esquemas que faciliten la explicación.

Tras esto nos centraremos en el diseño de la fuente de jardín inteligente programable. De esta manera, se presentarán los diferentes componentes hardware necesarios para el desarrollo del proyecto, haciendo un análisis exhaustivo de las características más relevantes de los mismos. Una vez estudiados dichos componentes, se presentarán los algoritmos necesarios en forma de diagramas de secuencias que nos permitan entender las acciones que nuestra fuente deberá ser capaz de realizar.

En el apartado siguiente se hará especial hincapié a la implementación y la implantación del proyecto. Con este fin se presentarán, primero, los diferentes montajes de los componentes hardware presentados previamente. Una vez llevada a cabo dicha tarea, se procederá a la presentación de los diferentes códigos necesarios, haciendo especial hincapié sobre aquellos elementos de mayor interés o complejidad. Dispondremos así de la información necesaria tanto para desarrollar la Interfaz de Usuario como de los diferentes *sketchs* a cargar en el microcontrolador Arduino.

Finalmente se presentarán las conclusiones, en las que se expondrá un leve análisis del conjunto del proyecto. Se presentarán así los diferentes componentes abordados a lo largo del presente documento, estudiándose el trabajo realizado y, en definitiva, poniendo el punto y final al mismo.

2. Estado de la cuestión

2.1 Introducción

Previo a realizar el análisis del proyecto llevado a cabo se procede a realizar una leve presentación de las tecnologías similares que a día de hoy podemos encontrarnos. De esta manera se presentarán una serie de proyectos realizados, bien por independientes o por empresas, para tratar de llevar a cabo una comparativa con el aquí presentado.

Una vez llevada a cabo esta presentación de algunos de los proyectos que nos podemos encontrar, procederemos a realizar la comparativa entre sus características. Gracias a esta comparativa se tratará de encontrar el lugar sobre el que se depositará nuestro proyecto.

2.2 Sistemas Similares

A continuación van a ser presentadas una serie de tecnologías que pueden encontrarse en el mercado. De esta manera, compararemos dichos proyectos con el explicado en el presente documento con la finalidad de establecer un marco sobre el que situarnos. Estas alternativas serán de muy diversa índole, por lo que se tratará de llevar a cabo una comparativa con la que queden claras sus funcionalidades.

Debido a que el proyecto presentado a lo largo del presente documento está enfocado sobre diferentes áreas (jardinería, meteorología, decoración y fuentes), van a presentarse a continuación una serie de proyectos que comparten algunas de estas.

2.2.1 Irrduino

La premisa inicial del proyecto Irrduino es simple: permitir el riego de las plantas del hogar de forma remota. De esta manera, deberíamos poder gestionar todo este proceso mediante, por ejemplo, un Smartphone o un navegador de internet por poner algún ejemplo sencillo.

Gracias a esta automatización del proceso, este sistema de riego inteligente puede ser empleado para el ahorro del agua, para mejorar la calidad del riego y el tiempo gastado en dicho proceso.

Irrduino ha sido desarrollado por los ingenieros de Google J.J Barrons y Joe Fernandez. Se apoya sobre las tecnologías Android, Arduino, Python, Dart y Google App Engine, además de comunicarse vía Ethernet. El nombre proviene del juego de palabras Irrigation (riego) junto a Arduino, que es el núcleo del proyecto.

Esta comunicación vía remota permite activar los diferentes aspersores situados en las diferentes zonas a regar, permitiéndose que se activen cada cierto intervalo de tiempo y ajustándolo a nuestras necesidades.

El código ha sido liberado de modo que se puede descargar desde la página web de los desarrolladores [1]. Se dispone, por tanto, de toda la información necesaria para la creación del mismo como, por ejemplo, la integración de la plataforma Arduino con Android.

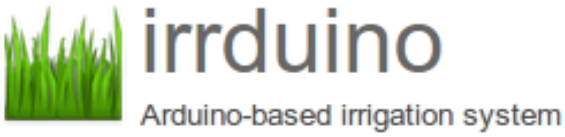

Figura 1: Logo del Proyecto Irrduino

Dicho código liberado está compuesto por diferentes proyectos software presentados a continuación.

Por un lado, nos encontramos IrrduinoController, núcleo del proyecto. Es el encargado de arrancar un pequeño servidor web y gestionar las diferentes peticiones sobre el mismo. Gestiona también los diferentes relés con los que activar y desactivar los aspersores. Disponemos además de IrrduinoRemote, una aplicación Android con la que gestionar las diferentes posibilidades que el proyecto nos ofrece. Finalmente nos encontramos con IrrduinoServer, un controlador alternativo escrito en Python.

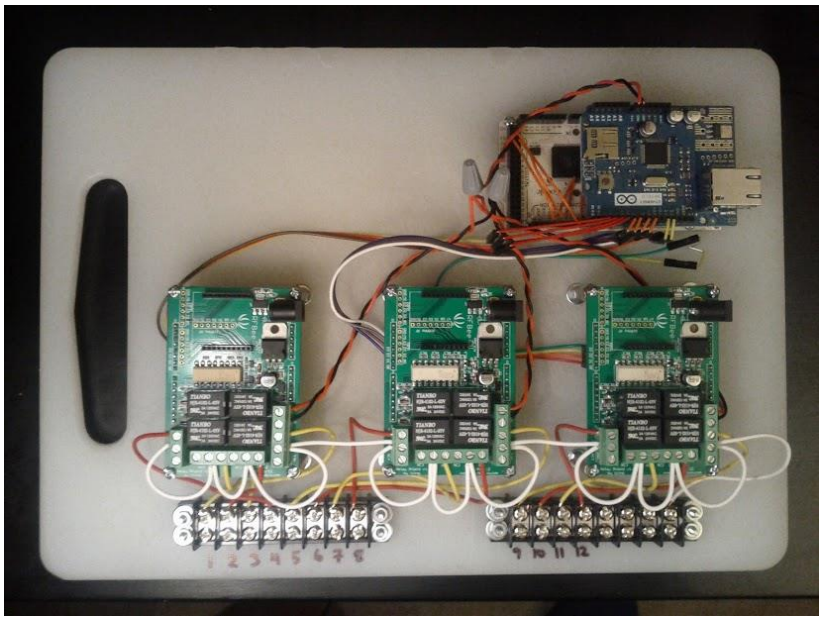

Figura 2: Montaje de Irrduino

Gracias a su base depositada sobre el microcontrolador Arduino este proyecto resulta una alternativa interesante y económica a los diferentes procesos de automatización del riego que podemos encontrarnos en el mercado.

2.2.2 Proyecto Meteo

La idea principal sobre la que se asienta este proyecto consiste en poder acceder remotamente a la información de una estación meteorológica conectada al Arduino. Así pues, podremos acceder a los diferentes datos ambientales, tales como la temperatura actual, máxima, mínima, la humedad, la presión atmosférica o la altitud sobre el nivel del mar desde cualquier sistema conectado a la red.

El núcleo del sistema está compuesto por la placa Arduino, sobre la que van conectados los diferentes sensores. Este Arduino irá leyendo los datos recibidos por estos sensores por intervalos regulables.

Figura 3: Captura de diferentes Interfaces

Para la visualización de los datos en los dispositivos el proyecto se apoya sobre la aplicación NetIO, que permite crear interfaces personalizadas e intercambiar datos entre ordenadores o microcontroladores. [3]

Los proveedores de Internet tienen una serie de IP públicas posteriormente repartidas entre sus clientes, adjudicándose a los diferentes routers adquieren direcciones temporales que cambian con el tiempo. Lógicamente, la placa Arduino deberá conocer en todo momento cuál es la IP de nuestro router. Para poder conocer, por tanto, la dirección IP del router cuando no nos encontremos conectados a la misma red que el mismo será necesario utilizar un DDNS, o Dynamic Domain Name Server. En este proyecto se ha utilizado el DDNS gratuito No-IP [6].

Para el montaje será necesaria una placa Arduino UNO, además de una Ethernet Shield. Tendremos también que emplear diferentes sensores de humedad, temperatura y presión atmosférica según las necesidades que se deseen cubrir.

Cabe señalar que en la web del desarrollador [2] podemos encontrar toda la información necesaria para fabricar nuestra propia estación meteorológica.

2.2.3 Lámpara RGB

Este proyecto se aleja levemente de los presentados anteriormente, pero parece una buena idea incorporarlo como proyecto a analizar debido a su función como elemento decorativo. Tenemos por tanto una lámpara controlada por Arduino, de manera que se iluminará con el color que nosotros consideremos conveniente. Destaca también su conexión vía Bluetooth, de manera que no son necesarios cables entre la lámpara y nuestro ordenador.

Esta lámpara está construida sobre una tira de 30 Leds RGB digital/indexable, encargada de darnos los diferentes colores que deseemos en un momento dado. Se tiene, además del microcontrolador Arduino, un módem Bluetooth encargado de realizar la conexión inalámbrica mencionada anteriormente.

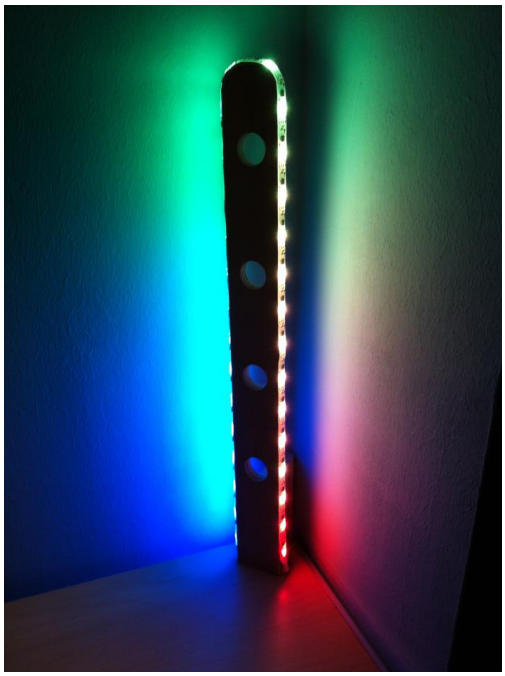

Figura 4: Lámpara RGB en funcionamiento

Por otro lado nos encontraremos con la interfaz desde la que podremos variar los colores mostrados en la lámpara. Esta interfaz presenta la paleta RGB, y simplemente seleccionando los colores deseados con el ratón podemos cambiar la iluminación de nuestra lámpara. La lámpara permite mantener guardado el patrón de colores, conservando dicha iluminación hasta que vuelva a ser actualizada.

Figura 5: Interfaz Lámpara RGB

Puede encontrarse mayor información sobre la Lámpara RGB en el blog personal del desarrollador [18].

2.2.4 Tipos de Fuentes

El siguiente apartado tiene un enfoque más abierto que los anteriormente presentados, pues se van a analizar los diferentes tipos de fuentes que podemos encontrarnos según su funcionamiento.

A día de hoy podemos encontrarnos con una gran cantidad de fuentes, adaptándose a las diferentes necesidades de los usuarios. De esta manera, podemos encontrarnos fuentes de diferentes tamaños, funcionamiento, decoración… No obstante, los diferentes chorros de las fuentes expulsarán el agua siguiendo una serie de mecanismos. De esta manera, según sea el funcionamiento de estos nos encontramos con los siguientes tipos de fuente:

2.2.3.1 Fuentes Automáticas

Las Fuentes Automáticas son los tipos de fuente más sencillos y básicos que podemos encontrarnos. El sistema únicamente se encargará de decidir cuándo se activarán y desactivarán los componentes que la componen.

Muchas de las fuentes que podemos ver por las calles emplean estos sistemas básicos para decidir cuándo bombear el agua, por ejemplo, según la hora. Debido a su sencillez, este mecanismo es también habitual para el uso de fuentes sencillas, como las de jardín.

Figura 6: Fuente de Agua Automática

2.2.3.2 Fuentes Programables

Las Fuentes Programables son gobernadas por un pequeño controlador programado con anterioridad. De esta manera, se prepara con anticipación un mecanismo y la fuente actúa en base al mismo.

De esta manera, podemos ajustar los diferentes parámetros de la fuente, obteniéndose como resultado un aspecto o funcionalidad adaptada a lo que efectivamente buscamos. Así, por ejemplo, pueden ajustarse la altura de los diferentes chorros de la fuente o indicar cuándo deben activarse para formar pequeñas formas.

Lógicamente modificando la programación del controlador podemos cambiar los diferentes parámetros de la fuente y obtener nuevas funcionalidades que se ajusten a lo que efectivamente se busca.

Figura 7: Fuente de Agua Programable

2.2.3.3 Fuentes Inteligentes

Finalmente podemos encontrarnos con las Fuentes Inteligentes. Al igual que las anteriores, disponen de un pequeño controlador que les permite recibir y gestionar los diferentes parámetros de la fuente.

No obstante, y a diferencia de las presentadas anteriormente, una Fuente Inteligente destaca porque realiza la obtención de estos parámetros en base a información recibida de manera automática. De esta manera, tras recibir la información adecuada se actuará en consecuencia, produciéndose una salida adaptada a las necesidades del momento.

Un ejemplo de fuente inteligente sería el **Reloj/Fuente Inteligente en Osaka Station**, en Japón. Dicha fuente dispone de un controlador que, gracias a la apertura y el cierre de diversas electroválvulas, permite la generación de imágenes decorativas. Tres veces por minuto, además, mostrará la hora local actual y la temperatura. Esto significa que la mayoría del tiempo la fuente no está mostrando la hora.

Algunos de los ejemplos de estas imágenes decorativas generadas por la fuente serían árboles, flores, hojas e incluso cascadas de agua.

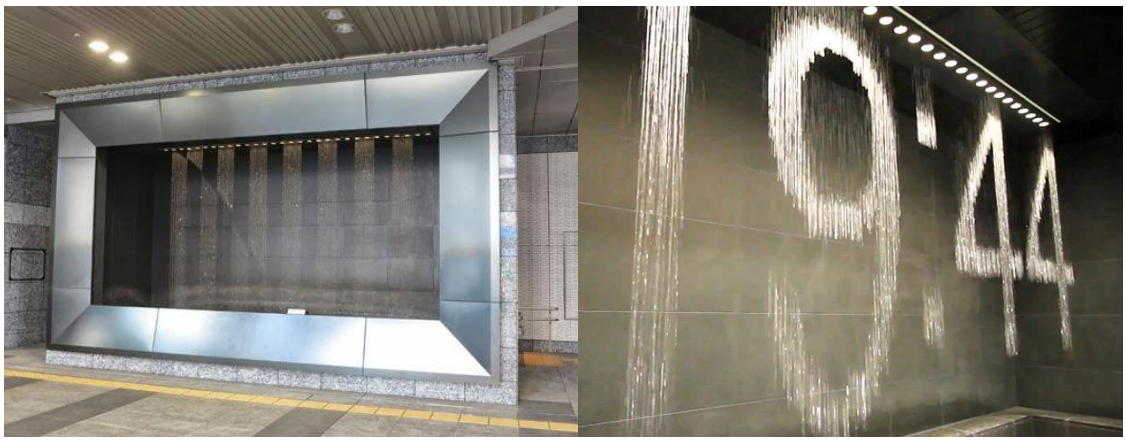

Figura 8: Reloj/Fuente Inteligente en Osaka Station

Otro ejemplo de fuentes de este tipo sería la Fuente Cibernética del Centro Comercial Parquesur, en España. Es una fuente situada en el centro de un lago artificial, y está compuesta por diferentes válvulas, surtidores y proyectores de iluminación para generar chorros de formas y colores variados. Dispone además de elementos musicales, pudiendo ajustar dichos chorros de agua y luces a la música a modo de espectáculo audiovisual.

Figura 9: Fuente Cibernética

2.3 Análisis

A continuación se dispone a realizar un análisis de cada una de las características de los proyectos presentados con anterioridad, pudiéndose ver qué funciones puede llevar a cabo cada uno de ellos. De este modo, se pretende establecer un marco sobre el que ubicaremos nuestro proyecto, comparándolo respecto a estos últimos.

Proyecto	*Lectura Temperatura*	*Riego*	*Decoración*	*Inalámbrico (Bluetooth)*	*Predicción Meteorológica*
Irrduino	NO	SÍ	NO	NO	NO
Proyecto Meteo	SÍ	NO	NO	NO	NO
Lámpara RGB	NO	NO	SÍ	SÍ	NO
Fuente Arduino	NO	NO	SÍ	NO	SÍ
Reloj Osaka Station	NO	NO	SÍ	NO	NO
Fuente Cibernética	NO	NO	SÍ	NO	NO

Tabla 1: Análisis

2.4 Tecnologías Involucradas

Debido a la variada naturaleza de los proyectos mencionados con anterioridad podemos encontrarnos una gran cantidad de tecnologías sin las cuales no sería posible.

Una de las más llamativas es el microcontrolador Arduino. Arduino es una plataforma Open Source apoyada tanto un hardware como en un software sencillo de utilizar. Su gran impacto se debe a dicha sencillez, pues no es necesario ser un experto para desarrollar muchas de nuestras ideas.

Arduino puede monitorizar los valores del medio ambiente gracias a una gran variedad de sensores que se conectan a sus entradas, además de poder controlar actuadores en base a mencionados valores. De esta forma, nos encontraremos con sensores y actuadores de diferente índole, como sensores de temperatura, presión y humedad o motores y luces. La placa puede obtenerse comprándola ya montada, si bien es cierto que disponemos de gran cantidad de información para construirla por nuestra cuenta [4]. La programación de esta placa se realiza mediante *Arduino programming language*, basado en Wiring, mientras que el desarrollo del entorno se realiza mediante *Processing*.

Por otro lado, la obtención de los datos meteorológicos se llevará a cabo mediante conexión a Internet. Dicha conexión se realizará gracias a la incorporación de una Ethernet Shield sobre el microcontrolador Arduino. Gracias a dicha Shield podremos conectar un cable Ethernet que una el controlador de la fuente a un router apropiado, realizándose la conexión y obteniéndose la información necesaria.

Será necesario también el empleo de diferentes sensores con los que monitorizar el entorno de la fuente. En base los parámetros obtenidos por estos sensores las acciones llevadas a cabo por la fuente serán notoriamente diferentes. Será imprescindible también el empleo de diferentes actuadores con los que dotar de la funcionalidad necesaria al microcontrolador Arduino. Todos estos componentes serán presentados en capítulos posteriores de la presente memoria.

2.5 Conclusiones

No es complicado ver que el abanico de posibilidades que se ha abierto respecto a proyectos desarrollados mediante Arduino o con funcionalidades aproximadas a nuestra fuente es muy amplio. En el presente documento se han presentado una serie de alternativas muy variadas y de diferentes campos, pudiendo obtenerse herramientas que se adapten a nuestras necesidades con apenas unas ligeras búsquedas en Internet. Lógicamente podemos encontrar una cantidad mucho mayor de proyectos de estas características, de modo que se han presentado los necesarios para ubicar nuestra alternativa.

De esta manera, nuestra fuente de jardín inteligente programable trata de englobar, a su modo, diferentes de estos campos recién presentados. Además no es habitual que se emplee la plataforma Arduino con estos fines, obteniéndose como resultado una fuente que trata de alejarse de todo lo visto con anterioridad. Debido a su desarrollo prácticamente íntegro sobre esta plataforma, el resultado es una fuente de costes económico y energético muy ajustados.

Se ha hecho, además, un leve análisis de las tecnologías protagonistas de nuestro proyecto. Estas tecnologías empleadas supondrán una mayor diferenciación respecto al resto de alternativas presentadas a lo largo de este mismo apartado. De este modo, con todos los datos aquí presentados se habrá ubicado nuestra fuente inteligente dentro de las diferentes alternativas que podemos encontrarnos a día de hoy.

3. Especificación de Requisitos

3.1 Introducción

Se va a proceder a continuación a desarrollar la especificación de requisitos (ERS) del proyecto realizado. Para ello nos apoyaremos sobre la última versión del estándar IEEE 830.

3.1.1 Propósito

La presente especificación de requisitos pretende mostrar qué es capaz, y qué no, de hacer el proyecto. Se incluirá por tanto toda la información necesaria sobre el comportamiento del mismo.

Cabe remarcar que esta ERS va dirigida tanto a potenciales compradores o usuarios como cualquiera que esté interesado en las funcionalidades que presenta.

3.1.2 Alcance

A partir de ahora nos referiremos al proyecto como Fuente Arduino.

La Fuente Arduino es una fuente de jardín inteligente, que actúa en base a los diferentes parámetros que reciba. Esta información será captada gracias a la conexión a un servidor de predicciones meteorológicas vía Internet. No obstante, el usuario puede optar por activar un modo manual en el que será él el encargado de introducir los parámetros, haciendo que actúe en base a sus necesidades. De esta manera, podrá modificar la altura del chorro y la iluminación en base a la información con la que esté trabajando.

Además, la Fuente Arduino dispone de un sistema de seguridad con el que evitar posibles desbordamientos debido a la excesiva cantidad de agua evacuando la misma. Este sistema puede emplearse también en caso de querer vaciarla para, por ejemplo, proceder a su posterior limpieza. Nos encontramos nuevamente, por tanto, con un sistema automático y otro manual.

El objetivo de este proyecto consiste en desarrollar una fuente que actúe de manera inteligente, y que no se limite únicamente a actuar como las fuentes programables. De esta manera dispondremos de una fuente que, más allá de un valor estético, tendrá un valor funcional al actuar como un sistema de predicciones meteorológicas.

3.1.3 Personal Involucrado

Nombre	Carlos Quer Barberá
Rol	Desarrollador ArduEntorno Fuente
Categoría profesional	Estudiante
Responsabilidades	Creación del sistema y su documentación
Información de contacto	carqueba@inf.upv.es

Tabla 2: Personal Involucrado - Carlos Quer

Nombre	Jose Luis Poza Luján
Rol	Supervisor
Categoría profesional	Profesor Contratado Doctor
Responsabilidades	Supervisión del proyecto
Información de contacto	jopolu@disca.upv.es

Tabla 3 - Personal Involucrado - Jose Luis Poza

Nombre	Miguel Juan Monter
Rol	Desarrollador
Categoría profesional	Estudiante
Responsabilidades	ArduEntorno Iluminación
Información de contacto	mijuamon@inf.upv.es

Tabla 4: Personal Involucrado - Miguel Juan

Nombre	Alberto Pedrera Ros
Rol	Desarrollador
Categoría profesional	Estudiante
Responsabilidades	ArduEntorno Energía
Información de contacto	alpedro@inf.upv.es

Tabla 5: Personal Involucrado - Alberto Pedrera

Nombre	Jesús Brun Conejos
Rol	Desarrollador
Categoría profesional	Estudiante
Responsabilidades	ArduEntorno Seguridad
Información de contacto	jesbruco@ei.upv.es

Tabla 6: Personal Involucrado - Jesús Brun

Nombre	Alberto Ramírez Losilla
Rol	Desarrollador
Categoría profesional	Estudiante
Responsabilidades	ArduEntorno Android
Información de contacto	alramlo@inf.upv.es

Tabla 7: Personal Involucrado - Alberto Ramírez

Nombre	Carlos Gil Rodríguez
Rol	Desarrollador
Categoría profesional	Estudiante
Responsabilidades	ArduEntorno Ambiente
Información de contacto	cargiro1@inf.upv.es

Tabla 8: Personal Involucrado - Carlos Gil

3.1.4 Definiciones, Acrónimos, Abreviaturas

Arduino: plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinares.

Shield: placas que pueden ser conectadas sobre la placa Arduino, permitiendo así extender las capacidades de ésta última. Suelen ser de bajo coste y sencillo montaje.

Ethernet Shield: shield que permite a Arduino la conexión a Internet.

Sketch: nombre que usa Arduino para definir un programa. Es la unidad de código que se sube y ejecuta en la placa.

Processing: lenguaje de programación que dispone de su propio entorno de desarrollo. Permite la creación de Interfaces de manera muy sencilla y visual. Dispone de una gran cantidad de librerías con las que extender sus funcionalidades. Libre para descargar y Open Source.

ControlP5: librería para Processing que permite añadir con gran facilidad diferentes controladores sobre las interfaces, permitiendo diferentes ventanas y agrupaciones de éstos.

Parseo: proceso de analizar una secuencia de símbolos a fin de determinar su estructura gramatical con respecto a una gramática formal.

EEPROM: *Electrically Erasable Programmable Read-Only Memory* (ROM programable y borrada eléctricamente). Arduino dispone de una memoria de este tipo, que será donde almacenaremos el contenido relacionado con la configuración.

3.1.5 Resumen

A continuación se realizará el análisis pormenorizado de los requisitos del sistema. Antes de comenzar, sin embargo, se llevará a cabo una breve explicación de la estructura que se seguirá para el mismo.

Se comenzará por tanto haciendo un análisis general del proyecto, tratando de encontrar así su ubicación dentro de otros realizados previamente, o actualmente en desarrollo. Dentro de este mismo bloque se presentará un ligero trazo de las funciones que el proyecto deberá presentar, sirviendo esto como una toma de contacto inicial con el mismo. Cabe señalar que a lo largo de este documento estas funciones serán ampliadas y explicadas en profundidad. Se presentarán, además, diferentes características de los usuarios, las restricciones y las dependencias del sistema, concluyendo con una breve explicación de la posible evolución futura del sistema.

Tras realizar esta ligera introducción se procederá a desarrollar tanto los requisitos específicos del sistema como los no funcionales. Cada uno de estos apartados serán presentados convenientemente, de manera que quede bien remarcado qué tareas, y cuáles no, puede llevar a cabo el sistema.

Se desarrollará pues dicho planteamiento.

3.2 Descripción General

3.2.1 Perspectiva del Producto

La Fuente Arduino es un producto independiente, completamente funcional por sí mismo. Sin embargo, cabe señalar que podría estar englobado dentro del proyecto "**ArduEntorno**". Este ArduEntorno engloba determinados proyectos encargados del desarrollo de un jardín con funcionalidades ampliadas.

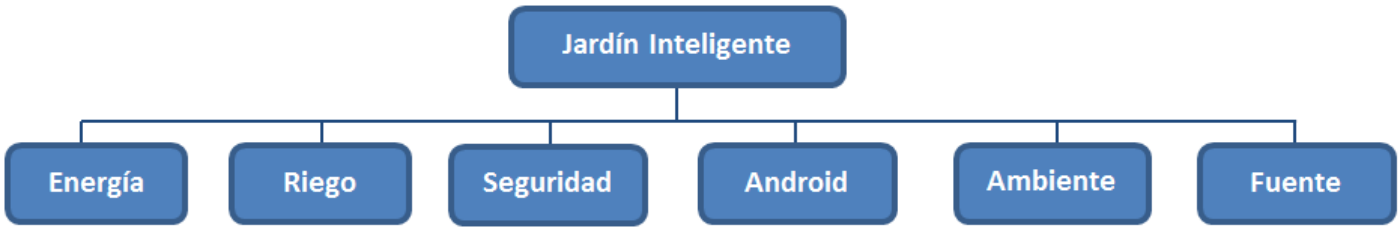

Figura 10: Perspectiva del Producto

De este modo, el proyecto Jardín Inteligente o **ArduEntorno** tiene como objetivo el desarrollo de elementos decorativos para un jardín que, más allá de únicamente la estética, tengan una determinada funcionalidad que les haga especialmente útiles.

Por otro lado, es fácil ver que la Fuente Arduino se engloba dentro del área de la domótica. Esté campo está compuesto por diferentes áreas que nos disponemos a presentar:

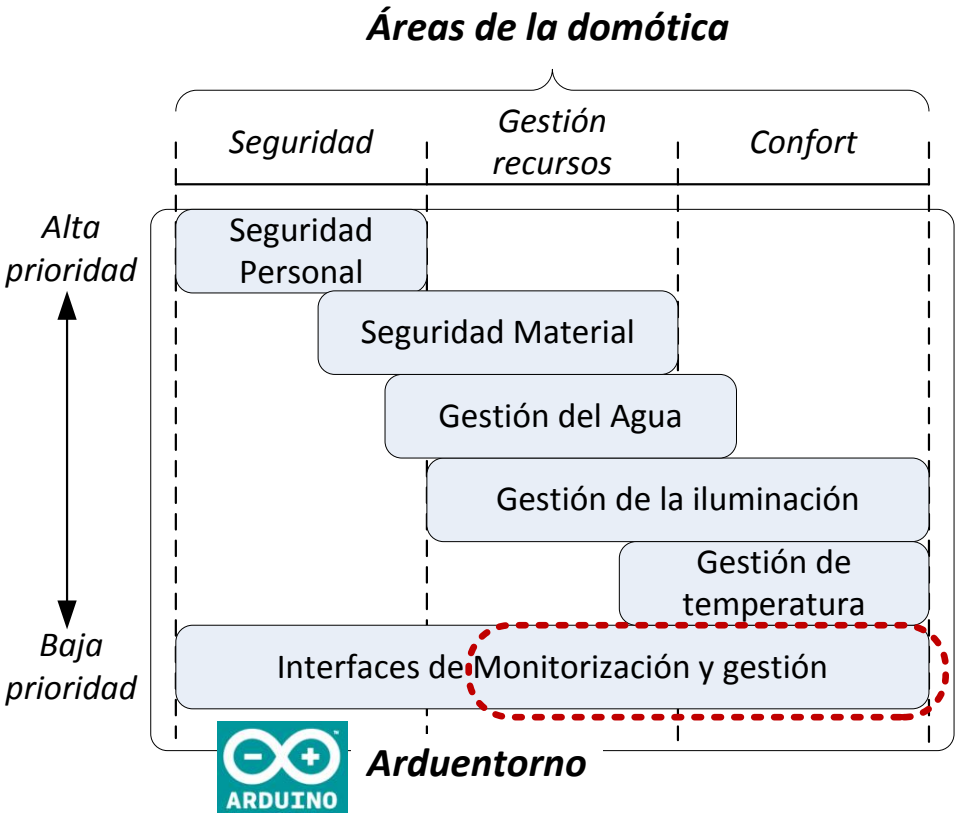

Figura 11: Áreas de la domótica

El proyecto aquí desarrollado engloba pues parte de la Gestión de Recursos, mientras que la parte de Confort es tomada por completo. Puede verse además que las áreas en las que el proyecto se ubica son de baja prioridad, debiéndose detener inmediatamente en caso de requerirse, mucho antes que otros sistemas de mayor prioridad.

3.2.2 Funcionalidad del Producto

La Fuente Arduino debe ser capaz de llevar a cabo las operaciones mencionadas a continuación.

Se dispondrá de una interfaz sencilla e intuitiva mediante la cual gobernaremos el funcionamiento de la Fuente. De este modo, dispondremos de una serie de controladores debidamente identificados que llevarán a cabo una función determinada.

Para comenzar se nos permitirá alternar entre la función inteligente y la función manual simplemente seleccionando el botón correspondiente. Al remarcar una de estas funciones se realizará la obtención de los datos de una manera u otra. Así, al seleccionar el primer modo podremos ajustar los parámetros de la altura de los chorros de la fuente y el mecanismo de iluminación al antojo del usuario, mientras que con el segundo modo se realizará el procesamiento de la información recibida por Internet y se realizará el cálculo de la altura del chorro y la iluminación de manera automática en base a éstos parámetros.

De este modo, la placa Arduino deberá ser capaz, primero, de interpretar el modo escogido. De este modo, recibirá de la Interfaz la opción escogida y actuará en base a esta elección. El

microcontrolador deberá ser capaz también de realizar la conexión a Internet a un servidor meteorológico, procesar toda la información y actuar en consecuencia. Finalmente, deberá poder interpretar los parámetros que el usuario introduzca en la Interfaz en caso de escoger la opción manual.

Sea cual sea la alternativa escogida dichos parámetros quedarán registrados en el propio microcontrolador que se encargará de generar una web HTML con ellos. De esta manera, desde cualquier dispositivo que se encuentre conectado a la misma red que la Fuente Arduino se podrá acceder a dicho website, permitiéndose ver los valores con los que la placa está trabajando en un momento dado.

Gracias a la memoria EEPROM que el microcontrolador Arduino pone a nuestra disposición se realizará el guardado del último estado conocido del sistema. De este modo, la placa tendrá la capacidad de recuperar este último estado en caso de resultar desconectada de la corriente. Se realizará, de este modo, la carga y descarga de estos datos de modo que se permita mantener estos parámetros.

La Fuente Arduino está dotada de un mecanismo de seguridad que impida que el agua desborde, captando la cantidad de agua y expulsando la que pueda suponer un problema. De este modo, gracias a una serie de sensores se montizorizará la cantidad de agua que hay en la fuente en cada momento, asegurando que permanece a un nivel aceptable. En caso de estar a un nivel muy bajo se detendrá inmediatamente la expulsión de agua por los chorros para evitar que la bomba sufra daños, mientras que si se capta una mayor cantidad de agua de la permitida se pondrá en marcha el sistema de evacuación. Se dispondrá además de un tercer sensor que detendrá inmediatamente la bomba que proporciona el chorro a la fuente si el nivel de agua no la cubre por completo, evitándose así daños sobre la misma.

En un momento dado puede ser interesante forzar el vaciado de la fuente para, por poner un ejemplo, proceder a su limpieza. De este modo, se ignorará la cantidad de agua que haya en un momento dado y se procederá a la evacuación del agua, deteniendo también el funcionamiento del chorro para evitar que la bomba trabaje en condiciones desfavorables.

Dispondremos también de un botón con el que interrumpir toda acción llevada a cabo por la Fuente, entrando en un estado estable. De este modo, el microcontrolador Arduino dejará de llevar a cabo acción alguna deteniendo la expulsión de agua, la iluminación y la evacuación de agua. Una vez se vuelva a habilitar el funcionamiento, la Fuente Arduino llevará a cabo su funcionamiento habitual.

3.2.3 Características de los Usuarios

En base las tareas que el usuario quiera que desempeñe la fuente y los conocimientos que tenga encontramos diferentes alternativas sobre el uso de la misma en base a éstos. A continuación, se procede a realizar un ligero análisis sobre los diferentes tipos de usuarios que podemos encontrarnos.

Tipo de usuario	Usuario Básico
Formación	Informática a nivel de Usuario
Habilidades	Manejo básico de los ordenadores.
Actividades	Poder alternar entre los diferentes modos que ofrece la fuente, pudiéndose modificar los parámetros de la misma.

Tabla 9: Usuario Básico

Tipo de usuario	Usuario Programador
Formación	Conocimientos en Processing y/o Arduino
Habilidades	Programación
Actividades	Modificar o ampliar los códigos facilitados para adaptarlos a sus necesidades. Dichos cambios deberán ser siempre documentados.

Tabla 10: Usuario Programador

3.2.4 Restricciones

Se hace especialmente necesario emplear herramientas Open Source, demostrándose la potencia que las mismas pueden ofrecernos. En caso de no poder cumplirse esta restricción debido a la total ausencia de las mismas se debería optar por utilizar la que mejor relación calidad-precio presente, tratándose de ajustar el precio total en la medida de lo posible.

Sería también conveniente emplear aquellos elementos que menor coste energético presenten, obteniéndose como resultado un producto de un coste muy ajustado.

Debido a que es necesario que el sistema disponga de acceso a Internet, gracias al cual obtendrá los valores en base a las predicciones tomadas por la estación meteorológica, se hace imprescindible disponer de un router con acceso a Internet al que conectar la fuente. En caso contrario, el sistema será completamente incapaz de tomar los parámetros en base a las predicciones meteorológicas.

3.2.5 Suposiciones y Dependencias

Debido a que se trabajará sobre la plataforma Arduino se hará necesario disponer de un Sistema Operativo que soporte su IDE. Encontramos instaladores en la página oficial para Windows, Mac OS X y Linux, además de todas las instrucciones necesarias para configurar el sistema correctamente para su arranque.

De manera análoga se hará necesario disponer de un Sistema Operativo que permita arrancar el Processing Development Environment (PDE), pudiendo ejecutarse así las interfaces desarrolladas sobre dicha plataforma.

3.2.6 Evolución Previsible del Sistema

El sistema, debido a su enfoque abierto, puede ampliarse en muchas y muy variadas direcciones. A continuación van a presentarse alguno de los diferentes enfoques que podrían emplearse para realizar una ampliación sobre el proyecto presentado en el presente documento.

La primera y más clara alternativa sería la de añadir nuevos parámetros a la fuente, permitiéndose alterar diferentes características de la misma en base a ellos. De este modo podría optarse, por ponerse algunos ejemplos, por añadir nuevas bombas de agua que actúen en base al viento registrado o superficies móviles que basen el movimiento sobre diferentes parámetros. Como puede verse, en este sentido el campo que se abre es muy amplio.

Lógicamente, todos estos cambios sobre los componentes hardware del sistema deberían verse reflejados en la interfaz, permitiendo al usuario la manipulación sobre los parámetros del mismo.

3.3 Requisitos Específicos

3.3.1 Requisitos comunes de los Interfaces

3.3.1.1 Interfaces de Usuario

Número de requisito	01		
Nombre de requisito	Requisito General de las Interfaces de Usuario		
Tipo	☒ Requisito	☐ Restricción	
Prioridad del requisito	☐ Alta/Esencial	☒ Media/Deseado	☐ Baja/ Opcional
Descripción	La Interfaz de Usuario será diseñada de tal manera que sea muy sencilla e intuitiva de utilizar por parte del Usuario, permitiendo además cumplir todos los requisitos expuestos en el presente apartado		

Tabla 11: Requisito General de las Interfaces de Usuario

Número de requisito	02		
Nombre de requisito	Desarrollo sobre la plataforma Processing		
Tipo	☐ Requisito	☒ Restricción	
Prioridad del requisito	☒ Alta/Esencial	☐ Media/Deseado	☐ Baja/ Opcional
Descripción	Todas las Interfaces de Usuario necesarias para el correcto funcionamiento del Proyecto serán desarrolladas sobre la plataforma Open Source Processing.		

Tabla 12: Desarrollo sobre la plataforma Processing

3.3.1.2 Interfaces Hardware

Número de requisito	03		
Nombre de requisito	Plataforma Arduino		
Tipo	☒ Requisito	☒ Restricción	
Prioridad del requisito	☒ Alta/Esencial	☐ Media/Deseado	☐ Baja/ Opcional
Descripción	La gestión y procesamiento de los diferentes parámetros necesarios para el funcionamiento de las diferentes alternativas que la fuente ofrece serán realizados mediante la plataforma Arduino.		

Tabla 13: Plataforma Arduino

3.3.1.3 Interfaces Sofware

Número de requisito	04		
Nombre de requisito	Consumo de Recursos Ajustado		
Tipo	☒ Requisito	☐ Restricción	
Prioridad del requisito	☐ Alta/Esencial	☒ Media/Deseado	☐ Baja/ Opcional
Descripción	El consumo de recursos por parte de la Aplicación debe ser lo más bajo posible, suponiendo la mínima carga sobre el sistema pero permitiéndose llevar a cabo los diferentes requisitos presentados.		

Tabla 14: Consumo de Recursos Ajustado

3.3.1.4 Interfaces Comunicación

Número de requisito	05		
Nombre de requisito	Conexión a Internet		
Tipo	☐ Requisito	☒ Restricción	
Prioridad del requisito	☒ Alta/Esencial	☐ Media/Deseado	☐ Baja/ Opcional
Descripción	La conexión al servidor meteorológico externo se llevará a cabo mediante la conexión a un router mediante Ethernet y redireccionando los datos hacia la ruta correspondiente.		

Tabla 15: Conexión a Internet

Número de requisito	06		
Nombre de requisito	Processing-Arduino		
Tipo	☐ Requisito	☒ Restricción	
Prioridad del requisito	☒ Alta/Esencial	☐ Media/Deseado	☐ Baja/ Opcional
Descripción	El intercambio de datos entre las Interfaces de Usuario desarrolladas bajo la plataforma Processing y el microcontrolador Arduino encargado de gestionar los diferentes parámetros de entrada y salida se llevará a cabo mediante Puerto Serie.		

Tabla 16: Processing-Arduino

3.3.2 Requisitos Funcionales

3.3.2.1 Visión global

A continuación se dispone a presentar el Caso de Uso general de nuestro sistema desarrollado, recogiéndose así todas las funciones que el mismo deberá ser capaz de llevar a cabo. Posteriormente se procesarán cada uno de los bloques generados, analizándose cada uno de los Requisitos Funcionales generados.

Dicho diagrama será como sigue a continuación:

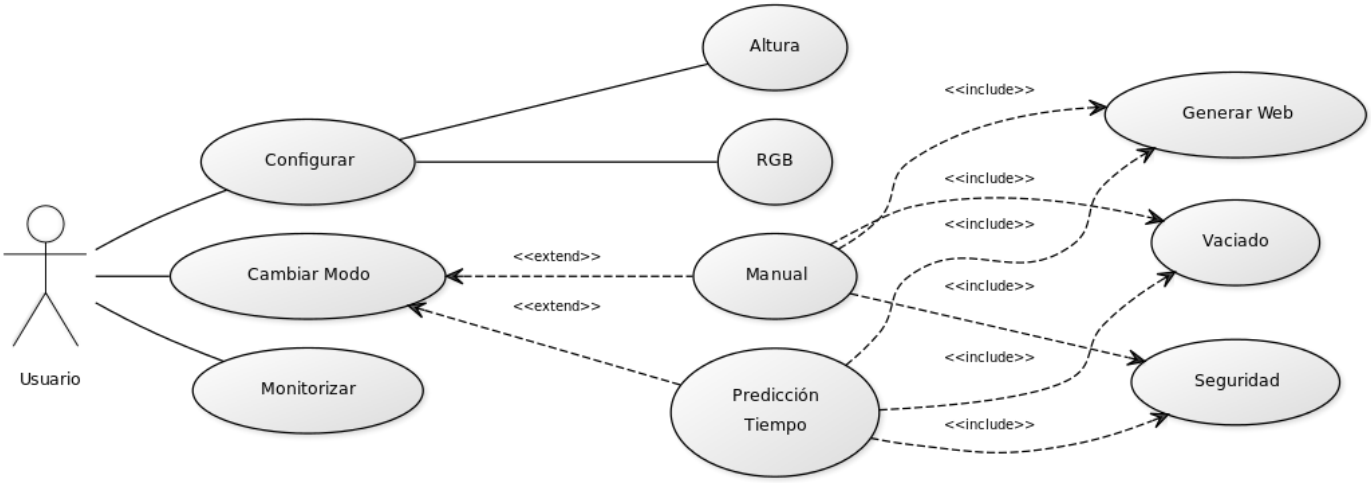

Figura 12: Diagrama de Casos de Uso del Sistema

Puede verse fácilmente que podemos encontrarnos tres bloques marcadamente diferenciados:

- Bloque Configurar: realizar la configuración de los parámetros de la fuente.

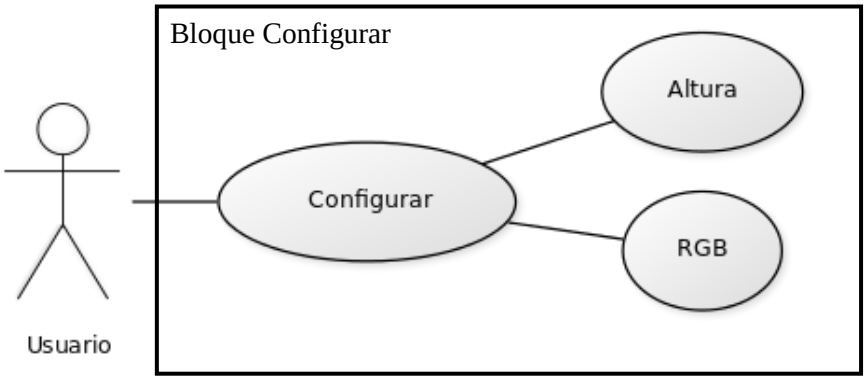

Figura 13: Bloque Configurar

- Bloque Cambiar Modo: alternar entre los modos disponibles.

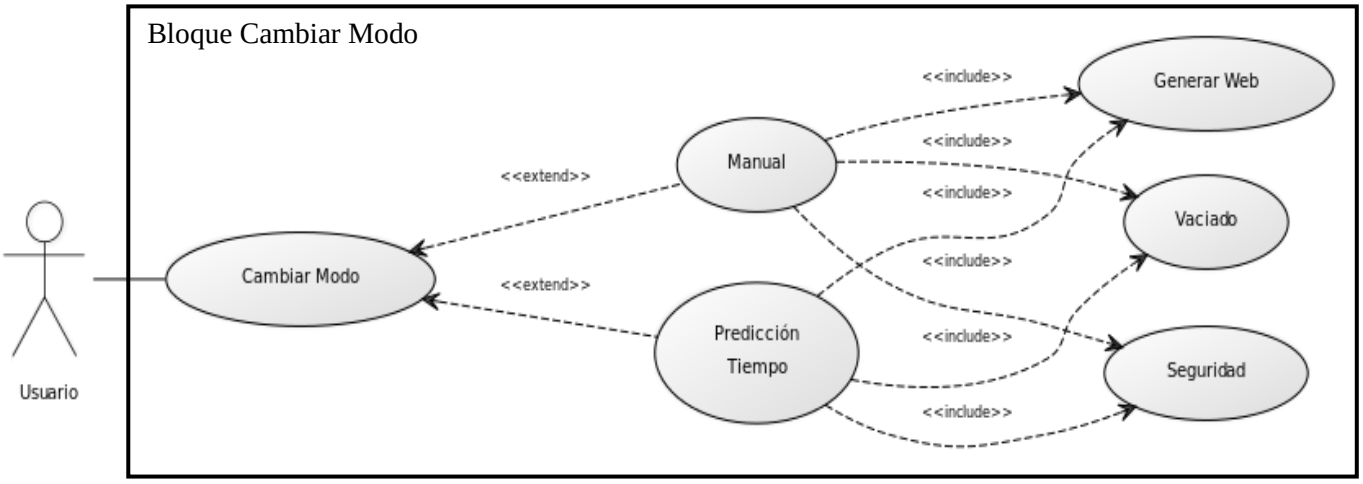

Figura 14: Bloque Cambiar Modo

- Bloque Monitorizar: visualizar los resultados.

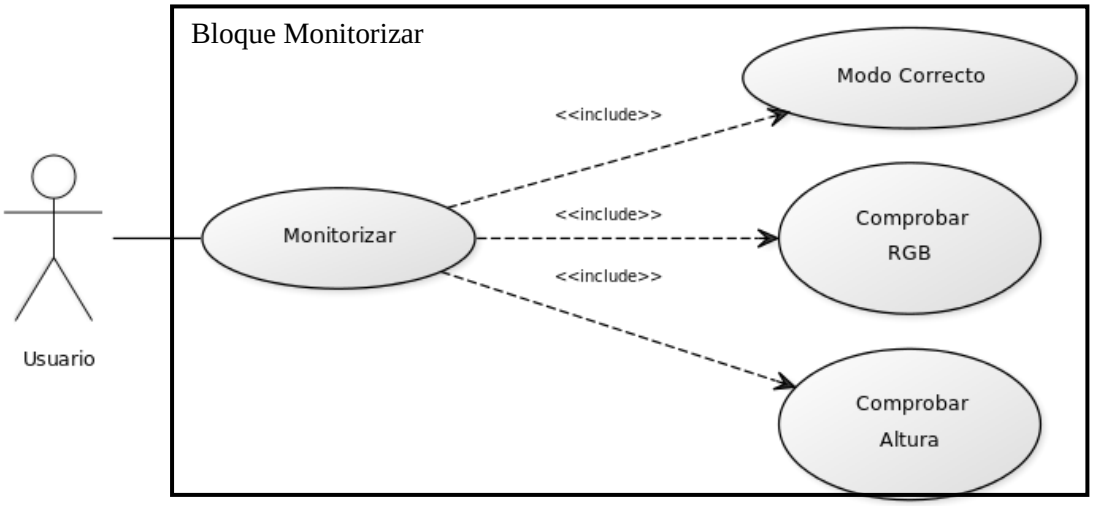

Figura 15: Bloque Monitorizar

Se dispone por tanto a realizar el análisis de cada uno de estos bloques y sus respectivos Requisitos Funcionales.

3.3.2.2 Bloque Configurar

Número de requisito	CU1.1		
Nombre de requisito	RGB		
Tipo	☒ Requisito	☐ Restricción	
Prioridad del requisito	☒ Alta/Esencial	☐ Media/Deseado	☐ Baja/ Opcional
Descripción	Se debe tener la posibilidad, tanto en modo Manual como Automático, de manipular el color del sistema de iluminación de la fuente en base a los parámetros de dichas alternativas.		
Inputs	Interfaz de usuario → Valores Analógicos de Rojo, Verde y Azul		
Outputs	RGB → Valores		
Proceso	Recepción de los valores Confirmar que se encuentran en el rango [0-255] Cambiar los valores de RGB Comprobar iluminación		
Restricciones	Solo se permite cambiar a usuarios validados		

Tabla 17: CU1.1 – RGB

Número de requisito	CU1.2		
Nombre de requisito	Altura		
Tipo	☒ Requisito	☐ Restricción	
Prioridad del requisito	☒ Alta/Esencial	☐ Media/Deseado	☐ Baja/ Opcional
Descripción	Se debe tener la posibilidad, tanto en modo Manual como Automático, de manipular la altura de la fuente en base a los parámetros de dichas alternativas.		
Inputs	Interfaz de usuario → Valor Analógico de Altura		
Outputs	RGB → Valores		
Proceso	Recepción de valor de altura Confirmar que se encuentran en el rango [0-255] Cambiar la altura de la fuente Comprobar altura		
Restricciones	Solo se permite cambiar a usuarios validados		

Tabla 18: CU1.2 - Altura

3.3.2.3 Bloque Selección Modo

Número de requisito	CU2.1		
Nombre de requisito	Selección Modo Manual		
Tipo	☒ Requisito	☐ Restricción	
Prioridad del requisito	☒ Alta/Esencial	☐ Media/Deseado	☐ Baja/ Opcional
Descripción	Este modo permite que el encargado de la introducción de los diferentes parámetros aceptados por la fuente la realice el propio usuario.		
Inputs	Interfaz de usuario → Valor del modo		
Outputs	Configuracion → Valor		
Proceso	Comprobar que el valor del modo es adecuado Comprobar el modo en el que se está funcionando Cambiar el valor del modo Comprobar que el modo se ha cambiado correctamente		
Restricciones	Solo se permite cambiar a usuarios validados		

Tabla 19: Selección Modo Manual

Número de requisito	CU2.2		
Nombre de requisito	Modo Ethernet (Predicción Tiempo)		
Tipo	☒ Requisito	☐ Restricción	
Prioridad del requisito	☒ Alta/Esencial	☐ Media/Deseado	☐ Baja/ Opcional
Descripción	La selección de este modo implica que los diferentes parámetros de la fuente serán actualizados en base a la información obtenida por el servidor meteorológico.		
Inputs	Interfaz de usuario → Valor del modo		
Outputs	Configuracion → Valor		
Proceso	Comprobar que el valor del modo es adecuado Comprobar el modo en el que se está funcionando Cambiar el valor del modo Comprobar que el modo se ha cambiado correctamente		
Restricciones	Solo se permite cambiar a usuarios validados		

Tabla 20: CU2.2 - Modo Ethernet (Predicción Tiempo)

Número de requisito	CU2.3		
Nombre de requisito	Generar Web		
Tipo	☒ Requisito	☐ Restricción	
Prioridad del requisito	☒ Alta/Esencial	☐ Media/Deseado	☐ Baja/ Opcional
Descripción	En base a los parámetros obtenidos en un momento dado por la placa Arduino se generará, gracias a la Ethernet Shield conectada a dicha plataforma, una página web a nivel local. A dicha web se tendrá acceso desde cualquier dispositivo que se encuentre conectado a la misma red.		
Inputs	Parámetros RGB Altura Temperatura Humedad		
Outputs	Web HTML Local		
Proceso	Comprobar que los valores recibidos son correctos Apertura Cliente Web Generación página HTML Comprobar corrección página HTML Detener Cliente Web		
Restricciones	La generación se llevará a cabo en lenguaje HTML		

Tabla 21: CU2.3 - Generar Web

Número de requisito	CU2.4		
Nombre de requisito	Vaciado		
Tipo	☒ Requisito	☐ Restricción	
Prioridad del requisito	☒ Alta/Esencial	☐ Media/Deseado	☐ Baja/ Opcional
Descripción	Al seleccionar esta alternativa la fuente detendrá todas sus acciones, y se procederá a llevar a cabo el vaciado del agua de la misma.		
Inputs	Interfaz de usuario → Valor del modo		
Outputs	Activación Bomba de Agua de Salida		
Proceso	Comprobar que el valor del modo es adecuado Comprobar el modo en el que se está funcionando Cambiar el valor del modo Comprobar que el modo se ha cambiado correctamente		
Restricciones	Solo se permite cambiar a usuarios validados		

Tabla 22: CU2.4 - Vaciado

Número de requisito	CU2.5		
Nombre de requisito	Seguridad		
Tipo	☒ Requisito	☐ Restricción	
Prioridad del requisito	☒ Alta/Esencial	☐ Media/Deseado	☐ Baja/ Opcional
Descripción	Gracias a los sensores que la placa Arduino tendrá conectados a sus correspondientes entradas se podrá monitorizar en cualquier momento el nivel del agua. Se obtendrán por tanto dos parámetros que indicarán si el agua ha superado dichos niveles, cada uno correspondiente a un sensor determinado.		
Inputs	Valor de Entrada Sensor 1 Valor de Entrada Sensor 2 Valor de Entrada Sensor 3		
Outputs	Activación Bomba correspondiente		
Proceso	Recepción Valores Comprobar corrección Concluir nivel de agua Actuar en consecuencia		
Restricciones	Solo se permite cambiar a usuarios validados		

Tabla 23: CU2.5 - Seguridad

Algunas de estas acciones tienen incluidas una serie de Requisitos Funcionales adicionales que se procede a detallar a continuación.

Bloque Predicción Tiempo:

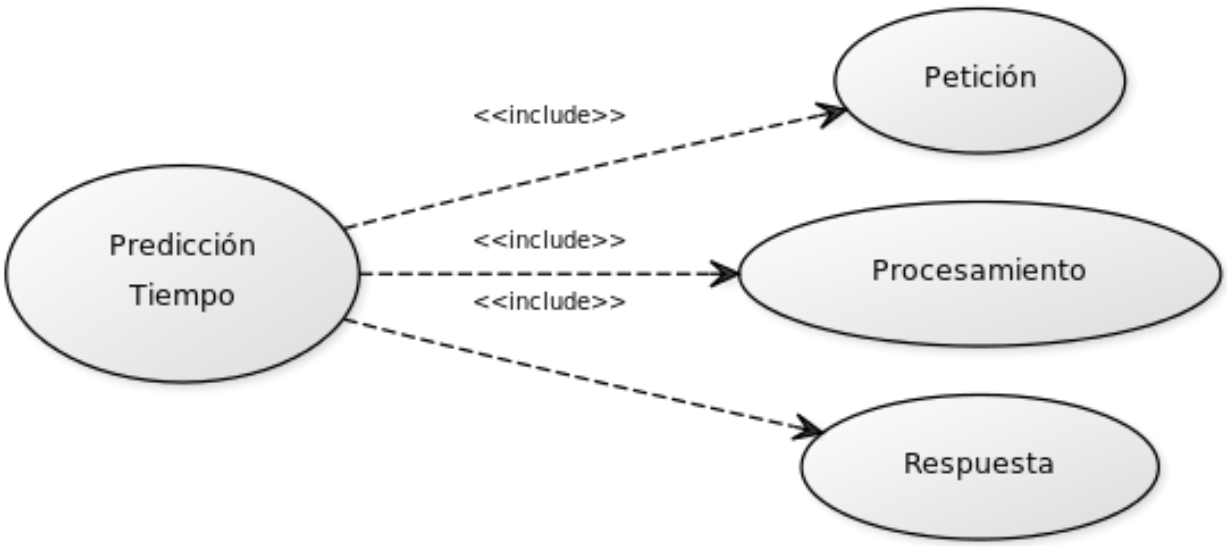

Figura 16: Predicción Tiempo

Siendo dichos Requisitos Funcionales como sigue en la siguiente página:

Número de requisito	CU3.1		
Nombre de requisito	Petición página web Servidor Meteorológico		
Tipo	☒ Requisito	☐ Restricción	
Prioridad del requisito	☒ Alta/Esencial	☐ Media/Deseado	☐ Baja/ Opcional
Descripción	El conjunto Arduino - Ethernet Shield debe tener la capacidad de enviar una petición apropiada al servidor meteorológico, de manera que este último pueda devolver una página web en formato HTML que Arduino sea capaz de interpretar.		
Inputs	Valor de Entrada Sensor 1 Valor de Entrada Sensor 2 Valor de Entrada Sensor 3		
Outputs	Petición HTML		
Proceso	Apertura Cliente Generación Petición HTML Comprobación de la corrección de la Petición Envío petición a servidor Externo		
Restricciones	Conexión a router con acceso a Internet Máximo de una petición cada 30 segundos (evitar sobrecarga)		

Tabla 24: CU3.1 - Petición página web Servidor Meteorológico

Número de requisito	CU3.2		
Nombre de requisito	Procesamiento		
Tipo	☒ Requisito	☐ Restricción	
Prioridad del requisito	☒ Alta/Esencial	☐ Media/Deseado	☐ Baja/ Opcional
Descripción	Con los datos obtenidos del Servidor Externo en formato HTML, el microcontrolador Arduino se hará cargo del procesamiento y parseo de dicho fichero, obteniéndose así los valores de Temperatura y Probabilidad de Lluvia		
Inputs	Respuesta HTML del Servidor Externo		
Outputs	Temperatura (en grados) Probabilidad de lluvia		
Proceso	Recepción web HTML Comprobación corrección de la web Parseo de la web Obtención parámetros Cierre del cliente Comprobar corrección de los parámetros		
Restricciones	Conexión a router con acceso a Internet		

Tabla 25: CU3.2 – Procesamiento

Número de requisito	CU3.3		
Nombre de requisito	Respuesta		
Tipo	☒ Requisito	☐ Restricción	
Prioridad del requisito	☒ Alta/Esencial	☐ Media/Deseado	☐ Baja/ Opcional
Descripción	Gracias a los parámetros obtenidos se actualizarán los valores de Iluminación y Altura de la fuente, indicando la previsión meteorológica.		
Inputs	Temperatura (en grados) Probabilidad de lluvia		
Outputs	Fuente actualizada en base a los valores recibidos.		
Proceso	Actualizar valores de la fuente Comprobar resultado		
Restricciones	Solo se permite cambiar a usuarios validados		

Tabla 26: CU3.3 - Respuesta

Seguridad:

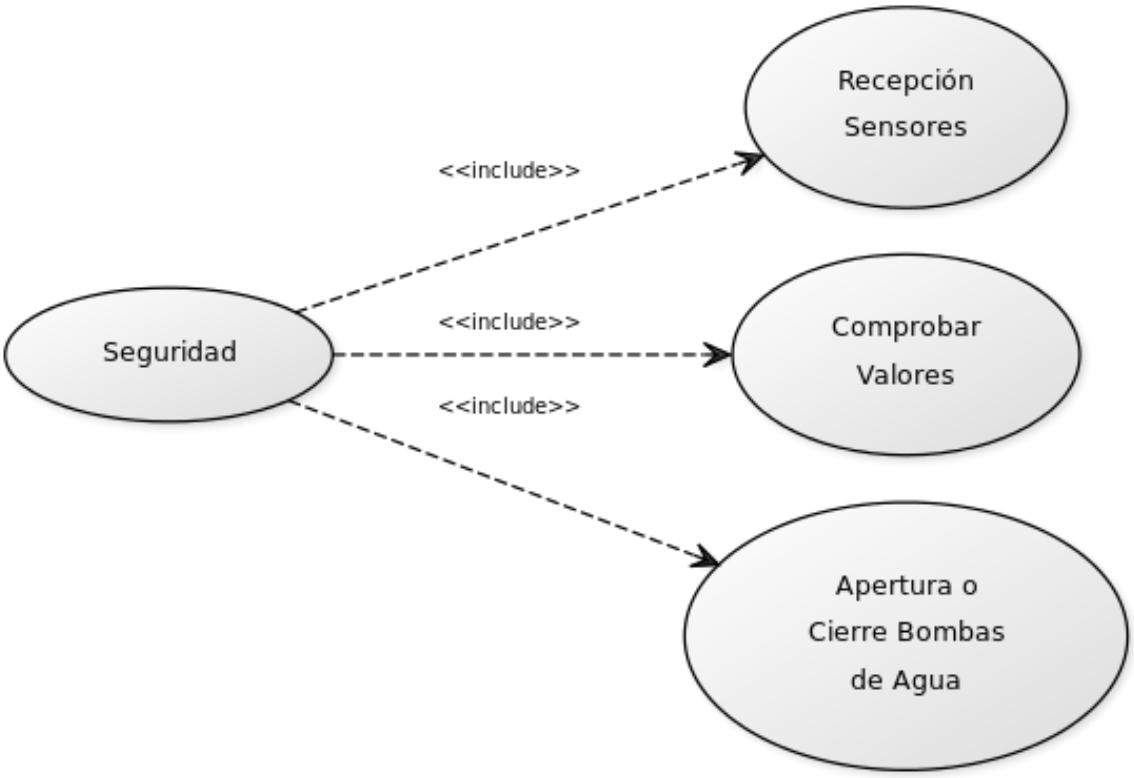

Figura 17: Seguridad

Siendo los requistos Funcionales asociados a dicho bloque tal y como siguen a continuación:

Número de requisito	CU4.1		
Nombre de requisito	Recepción Sensores		
Tipo	☒ Requisito	☐ Restricción	
Prioridad del requisito	☒ Alta/Esencial	☐ Media/Deseado	☐ Baja/ Opcional
Descripción	Se realiza la recepción de los diferentes sensores ubicados en la fuente, de manera que posteriormente se llevará a cabo el análisis de los mismos.		
Inputs	Modo Seguridad		
Outputs	Valor Sensor 1 Valor Sensor 2 Valor Sensor 3		
Proceso	Recepción Sensor 1 Recepción Sensor 2 Recepción Sensor 3 Comprobar corrección valores		
Restricciones	Sensores conectados a microcontrolador Arduino		

Tabla 27: CU4.1 - Recepción Sensores

Número de requisito	CU4.2		
Nombre de requisito	Comprobar Valores		
Tipo	☒ Requisito	☐ Restricción	
Prioridad del requisito	☒ Alta/Esencial	☐ Media/Deseado	☐ Baja/ Opcional
Descripción	En base a los valores obtenidos por los sensores de la fuente se concluye el nivel de agua que contiene la fuente, a lo que habrá que responder y actuar en consecuencia.		
Inputs	Valor Sensor 1 Valor Sensor 2 Valor Sensor 3		
Outputs	Señal Bomba de Agua entrada Señal Bomba de Agua salida		
Proceso	Comprobar valores de los sensores Concluir nivel de agua Envío de señales a las bombas		
Restricciones	Sensores conectados a microcontrolador Arduino		

Tabla 28: CU4.2 - Comprobar Valores

Número de requisito	CU4.3		
Nombre de requisito	Activación/Desactivación Bomba de Agua		
Tipo	☒ Requisito	☐ Restricción	
Prioridad del requisito	☒ Alta/Esencial	☐ Media/Deseado	☐ Baja/ Opcional
Descripción	Se procederá a la activación o desactivación de las bombas de agua de entrada o de salida en base a los valores obtenidos.		
Inputs	Señal Bomba de Agua entrada Señal Bomba de Agua salida		
Outputs	Bomba de Agua de Entrada activada/desactivada Bomba de Agua de Salida activada/desactivada		
Proceso	Activar/Desactivar Bomba de Agua de entrada Activar/Desactivar Bomba de Agua de salida Comprobar cambio correcto		
Restricciones	Sensores conectados a microcontrolador Arduino		

Tabla 29: CU4.3 - Activación/Desactivación Bomba de Agua

3.3.3 Requisitos No Funcionales

3.3.3.1 Rendimiento

Número de requisito	7		
Nombre de requisito	Robustez		
Tipo	☒ Requisito	☐ Restricción	
Prioridad del requisito	☒ Alta/Esencial	☐ Media/Deseado	☐ Baja/ Opcional
Descripción	Tanto la Interfaz de Usuario como el propio microcontrolador Arduino deberán ser capaces de soportar todas las acciones llevadas a cabo sobre el Proyecto, evitándose posibles errores o fallos de funcionamiento ocasionados por las mismas.		

Tabla 30: Robustez

Número de requisito	8		
Nombre de requisito	Tiempo Real		
Tipo	☒ Requisito	☐ Restricción	
Prioridad del requisito	☐ Alta/Esencial	☒ Media/Deseado	☐ Baja/ Opcional
Descripción	La placa Arduino debe ser capaz de procesar en Tiempo Real (o en su defecto la menor cantidad de tiempo posible) los diferentes parámetros, mostrándose por las salidas de la misma las acciones realizadas en base a los mismos.		

Tabla 31: Tiempo Real

3.3.3.2 Seguridad

Número de requisito	9		
Nombre de requisito	Seguridad Internet		
Tipo	☒ Requisito	☐ Restricción	
Prioridad del requisito	☐ Alta/Esencial	☒ Media/Deseado	☐ Baja/ Opcional
Descripción	Los mecanismos de seguridad en el envío y recepción de los datos del Servidor Meteorológico Externo corresponderán a los de los respectivos protoclos empleados.		

Tabla 32: Seguridad Internet

Número de requisito	10		
Nombre de requisito	Codificación		
Tipo	☒ Requisito	☒ Restricción	
Prioridad del requisito	☐ Alta/Esencial	☒ Media/Deseado	☐ Baja/ Opcional
Descripción	Los diferentes datos intercambiados entre las plataformas Processing y Arduino se llevarán a cabo siguiendo una codificación concreta, de modo que cualquier persona ajena al proyecto no tenga acceso a la manipulación de dichos parámetros.		

Tabla 33: Codificación

3.3.3.3 Fiabilidad

Número de requisito	11		
Nombre de requisito	Fiabilidad		
Tipo	☒ Requisito	☐ Restricción	
Prioridad del requisito	☐ Alta/Esencial	☒ Media/Deseado	☐ Baja/ Opcional
Descripción	La fiabilidad del Sistema debe ser lo más cercana posible al cien por cien, funcionando convenientemente durante los intervalos de tiempo necesarios.		

Tabla 34: Fiabilidad

3.3.3.4 Disponibilidad

Número de requisito	X		
Nombre de requisito	Disponibildad		
Tipo	☒ Requisito	☐ Restricción	
Prioridad del requisito	☐ Alta/Esencial	☒ Media/Deseado	☐ Baja/ Opcional
Descripción	La disponibilidad del sistema deberá ser lo más cercana posible al cien por cien, debiendo siempre mostrarse preparado ante cualquier acción realizada sobre el mismo.		

Tabla 35: Disponibilidad

3.3.3.5 Mantenibilidad

Número de requisito	X		
Nombre de requisito	Ampliaciones		
Tipo	☒ Requisito	☐ Restricción	
Prioridad del requisito	☐ Alta/Esencial	☒ Media/Deseado	☐ Baja/ Opcional
Descripción	El sistema deberá ser, en la medida de lo posible, susceptible de ser ampliado. Estas futuras ampliaciones deberán ajustarse a los criterios establecidos previamente, no poniendo en riesgo ninguno de los requisitos presentados con anterioridad.		

Tabla 36: Ampliaciones

3.3.3.6 Portabilidad

Número de requisito	X		
Nombre de requisito	Software Libre		
Tipo	☒ Requisito	☒ Restricción	
Prioridad del requisito	☐ Alta/Esencial	☒ Media/Deseado	☐ Baja/ Opcional
Descripción	Gracias a su pleno desarrollo sobre plataformas Open Source el sistema deberá poder ser importado al mayor número posible de plataformas. Esta portabilidad no deberá poner en riesgo ninguno de los requisitos anteriores.		

Tabla 37: Software Libre

3.4 Conclusiones

Tras el desarrollo del presente punto se han mostrado todas las posibilidades que la Fuente Arduino ofrece. De este modo, se ha proporcionado toda la información relativa a la funcionalidad de nuestro producto, debiendo quedar bien claro qué es capaz, y qué no, de realizar.

Además de los diferentes bloques y requisitos funcionales se han proporcionado datos de diferente índole sobre el proyecto, tales como acrónimos, siglas, perspectiva y, en definitiva, datos de interés.

En conclusión, y tras finalizar el presente apartado, debería haber quedado bien claro qué tareas es capaz de llevar a cabo nuestro proyecto, y cuáles no. Recordemos que dichas tareas han sido agrupadas en bloques, de manera que su localización pueda realizarse de manera sencilla. Estos componentes se han desarrollado, a su vez, en forma de diagramas y tablas.

4. Diseño del Sistema

4.1 Introducción

Se realizará el análisis detallado de todos y cada uno de los elementos de la arquitectura del proyecto. Así, los componentes hardware, software, módulos y datos que sean requeridos para el desarrollo de la Fuente Arduino serán presentados y explicados en profundidad.

Se comenzará por tanto presentando cada uno de estos elementos y su interconexión en el apartado Topología del Sistema. Además de un ligero esquema, en el que aparecerán reflejados dichos componentes, se llevará a cabo una explicación de la función que éstos llevarán a cabo.

Una vez realizada esta explicación se procederá a detallar cada uno de estos componentes en el apartado Especificación Hardware. Se presentarán así las características tanto de funcionamiento como de interés de los mismos.

Finalmente, se llevará a cabo una explicación detallada del comportamiento del Proyecto en el apartado Especificación Software. Cabe señalar que para esta tarea será empleado el Lenguaje Unificado de Modelado, más conocido por sus siglas en inglés UML.

4.2 Topología del Sistema

Se comenzará presentando, por tanto, la Topología del Sistema mediante un ligero esquema, que será posteriormente explicado. Dicho esquema será como sigue a continuación:

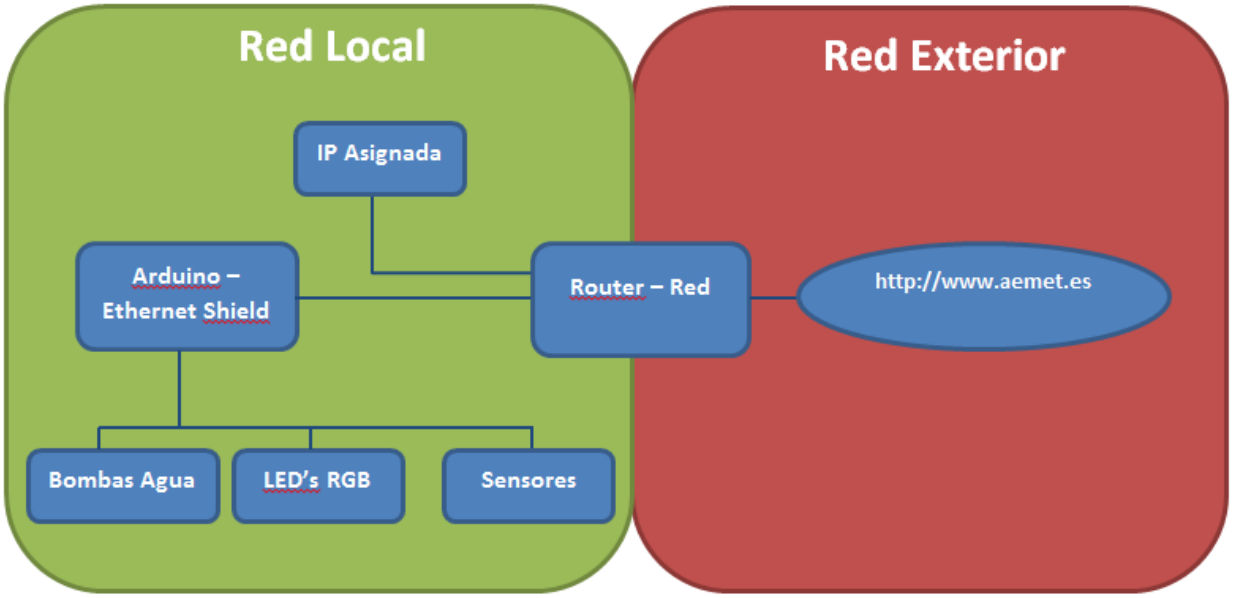

Figura 18: Topología del Sistema

Para comenzar, se dispone de una placa Arduino, sobre el que tendremos acoplada una Ethernet Shield que nos permitirá realizar la conexión a Internet. Este microcontrolador estará conectado a diferentes componentes de la propia fuente, a los que gobernará en base a diferentes parámetros. Para comenzar, manejará la altura del chorro de la fuente, además de controlar el

color de los diferentes LED's que tendrá conectados. Como, además, Arduino es el encargado de gestionar el nivel de agua dispondrá de unos sensores horizontales que le indicarán cuándo se supera el máximo o se alcanza el mínimo permitidos. En caso de alcanzar y/o superar estos valores permitidos, se activarán una serie de bombas de agua que añadirán o retirarán el agua necesaria.

Puede verse que el microcontrolador está conectado a un router, gracias al cual se podrá efectivamente acceder a redes externas a nuestra propia red local. Al conectar una Ethernet Shield al router se recibirá una dirección IP, a la que podremos acceder desde cualquier dispositivo conectado a la red local para visualizar el Website generado por el propio Arduino.

Todos los elementos presentados hasta ahora formarán parte de nuestra Red Local, siendo el router la pasarela o Gateway de cara al exterior. Finalmente, gracias al router se podrá acceder al website ubicado en el exterior: en nuestro caso de la Agencia Estatal de Meteorología. Esto será detallado en mayor profundidad en posteriores apartados.

4.3 Especificación Hardware

Tal y como se ha detallado con anterioridad se procede a desarrollar cada uno de los componentes hardware necesarios para poder llevar a cabo la Fuente Arduino. De este modo, para los diferentes componentes que sean necesarios se indicarán todos los detalles que resulten de interés a la hora de desarrollar el proyecto.

4.3.1 Arduino UNO revisión 3

La placa Arduino es uno de los núcleos principales sobre los que gira el proyecto, recayendo sobre ella gran parte del peso del mismo.Arduino es una plataforma electrónica Open-Source con la capacidad de monitorizar el entorno gracias a una gran variedad de sensores, llevando a cabo una serie de acciones gracias a los actuadores que tenga conectados. Un proyecto Arduino puede funcionar por sí mismo o colaborar con software que se arranque desde un ordenador.

Las placas de Arduino pueden comprarse ya preparadas, si bien existe la posibilidad de diseñar una a mano gracias a la licencia Open-Source de la documentación necesaria para ello. Toda esta información puede encontrarse en el propio Website de Arduino.

Existen diferentes modelos de Arduino adaptados a las necesidades del proyecto, ofreciéndose así diferentes alternativas. Cada una de estas opciones dispone de diferentes voltajes de entrada, así como entradas analógicas e inputs/outputs digitales, entre muchas otras características.

En nuestro caso, se ha optado por emplear una placa **Arduino UNO rev 3**. Presentaremos por tanto toda la información relevante sobre la misma.

Cabe señalar que la versión UNO de Arduino es sobre la que parten el resto de versiones. Puede recibir la potencia tanto vía USB como por un adaptador AC-DC El rango recomendable se encuentra entre 7 y 12 voltios.

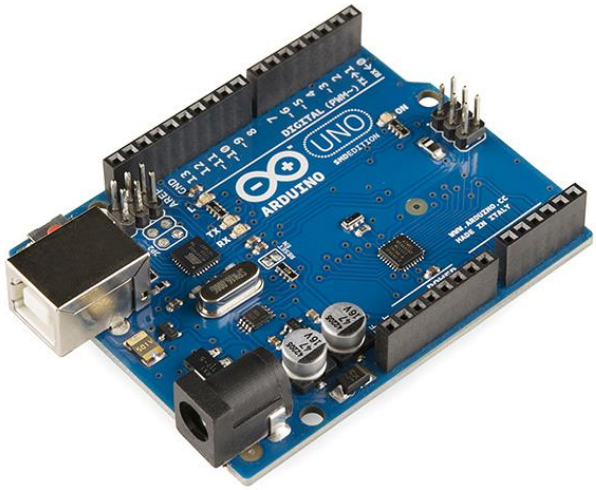

Figura 19: Arduino UNO Revisión 3

A continuación se muestran las especificaciones técnicas en forma de tabla, tal y como se indican en la página oficial de Arduino [7]:

Microcontrolador	ATmega328
Voltaje Operativo	5V
Voltaje de Entrada recomendado	7 – 12V
Voltaje de Entrada Límite	6 – 20V
Número de Pines Digitales I/O	14
Número de Pines Digitales PWM I/O	6
Número de Pines Analógicos	6
DC por Pin de I/O	40 mA
DC por Pin de 3.3V	50 mA
Memoria Flash	32 KB
Memoria Flash para el Gestor de Arranque	0.5 KB
SRAM	2 KB
EEPROM	1 KB
Velocidad de Reloj	16 MHz
Longitud	68.6 mm
Anchura	53.4 mm

Tabla 38: Especificaciones Arduino

4.3.2 Arduino Ethernet Shield R3

Gracias a la conexión de esta Shield sobre el microcontrolador Arduino presentado previamente se permite que este último tenga acceso a Internet. Es necesario conectar la Ethernet Shield a un router conectado a la red mediante un cable RJ45.

Al igual que todo lo relacionado con Arduino, toda la documentación necesaria de elementos tanto hardware como software se encuentra disponible con licencia Open-Source. Si bien se ha optado por el empleo de una Ethernet Shield ya desarrollada se podría diseñar una propia.

Figura 20: Arduino Ethernet Shield Revisión 3

Este modelo dispone de una ranura para una tarjeta micro-SD. Dispone, además, de una serie de LED's que indican información sobre la conexión, así como un botón que permite resetear tanto la Shield como la placa Arduino a la que se encuentra conectado.

PWR – Recibe energía	**RX** – Recepción de Datos
LINK – Presencia de Red	**TX** – Envío de Datos
FULLD – Conexión Full-Duplex	**COLL** – Detección de colisiones
100M – Red de 100 Mb/s	-

Tabla 39: LED's conexión Ethernet Shield

Cabe señalar que es necesario conectar la presente Shield a una placa Arduino compatible, de manera que éste pueda suministrarle los 5 Voltios necesarios para su correcto funcionamiento. A continuación se presentan algunas de las especificaciones técnicas de mayor interés, tal y como pueden encontrarse en la página principal de Arduino [9]

Controlador Ethernet	W5100. Buffer Interno 16KB
Acceso Tarjeta micro-SD	Pin 4
Velocidad Conexión	10/100 Mb
Número máximo conexiones	4
Conexión Arduino	Puerto SPI
Voltaje Operativo	5 Voltios

Tabla 40: Especificaciones Ethernet Shield

4.3.3 LED's RGB

Los LED's RGB se encargan de uno de los aspectos decorativos de la Fuente, como es la iluminación. Gracias al código cargado en el microcontrolador Arduino, estos LED's se iluminarán según las necesidades del usuario o según las predicciones meteorológicas recibidas de un servidor externo.

Estos LED's disponen de cuatro pines. Tres se corresponderán con cada uno de los colores de la escala RGB (rojo, verde y azul), y uno que es el cátodo, compartido por cada uno de los anteriores. Gracias a la creación de una diferencia de voltaje entre el cátodo y el voltaje de las salidas PWM a la que los pines RGB están conectados se consigue un color que oscila entre esos tres colores.

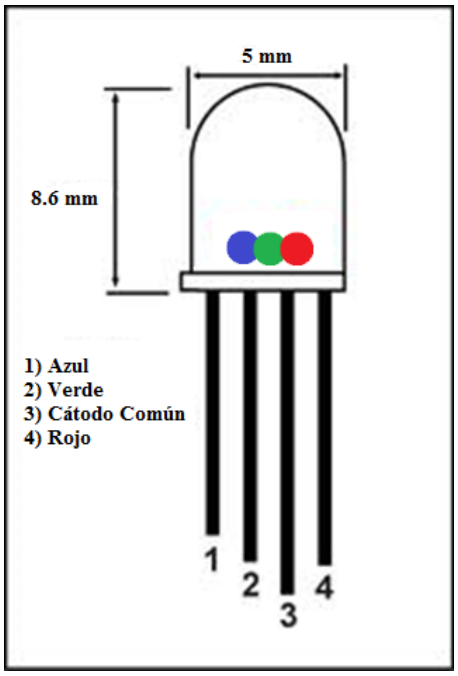

Figura 21: LED RGB Cátodo Común

Se podría haber optado por emplear LED's RGB de ánodo común, a lo que tendríamos que haber modificado ligeramente el funcionamiento del programa a la hora de calcular las diferentes diferencias de voltaje.

A continuación se presentan algunas de las características más relevantes de estos elementos, tal y como se detalla en la ficha técnica de los mismos [10]

Características
Color RGB
Cátodo Común
Medidas Estándar
Voltaje Máximo: 2.5(R), 4.1(G), 4 (B) Voltios
Corriente Máxima: 30(R), 25(G), 30 (B) mA
Tipo de Lente: Transparente
Ángulo de Emisión: 50º

Tabla 41: Especificaciones LED RGB

4.3.4 Bomba Agua- SolarPumpe Palermo

Otro de los aspectos que la placa Arduino se encargará de gestionar será la altura del chorro de la fuente. Dicho chorro será posible gracias a la aparición de la presente bomba de agua. Cabe señalar que la bomba dispone de una serie de sensores que impiden que ésta bombee si no está completamente sumergida, evitándose así daños en la misma.

Figura 22: Bomba de Agua modelo SolarPumpe Palermo

Según el voltaje que se le suministre a la bomba de agua, ésta tendrá una mayor o menor potencia. Esto provocará, lógicamente, que la altura del chorro de agua se corresponda a dicho voltaje, tal y como se muestra en el siguiente diagrama:

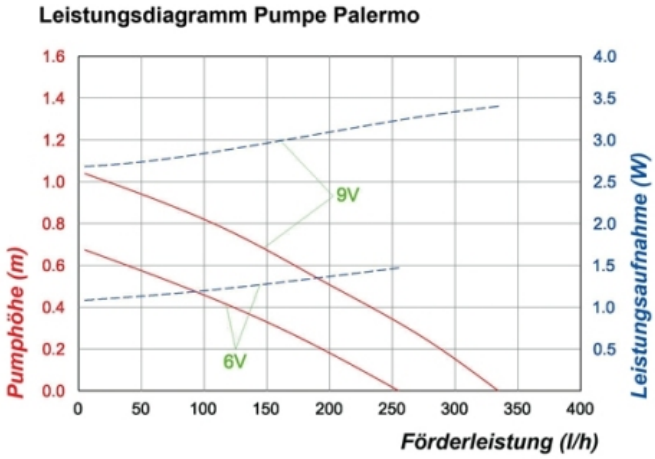

Figura 23: Diagrama Voltaje/Altura Bomba de Agua Palermo

A continuación se dispone a presentar sus especificaciones técnicas, tal y como vienen indicadas por el fabricante:

Rendimiento de bombeo	240 l/h (6V) – 330 l/h (9V)
Altura máxima de bombeo	0.7m (6V) – 0.9m (9V)
Corriente	230mA (6V) – 375ma(9V)
Área Tensión de Alimentación	6 – 9 VDC
Entrada Alimentación	1.3W (6V) – 3.0W (9V)
Protección	IP 68
Longitud del cable	5 metros
Peso	270 gramos
Diámetro salida	10/13 mm

Tabla 42: Especificaciones Solarpumpe Palermo

4.3.5 Sensor de Agua Nivel Horizontal PTFA3415

Los sensores de nivel horizontal serán los encargados de monitorizar el nivel de agua, indicando al microcontrolador Arduino qué acciones debe tomar. Dispone de una pequeña boya que indicará a la placa el nivel actual.

El resultado será que, según en la posición en la que se encuentre dicha boya, se producirá un cambio en la resistencia correspondiente a la distancia entre la zona superior del sensor y la superficie del líquido. Esta resistencia es inversamente proporcional a la distancia del líquido, de modo que a menor es el nivel del líquido mayor será dicha resistencia, y viceversa.

Figura 24: Sensor de Agua Nivel Horizontal

A continuación se presentan algunas de sus características más relevantes, aparecidas en la Ficha Técnica:

Tensión de Conmutación máxima	300 AC/DC
Corriente de Conmutación máxima	0.5 A
Protección	IP 68
Longitud del cable	1.5 metros
Temperatura Funcionamiento	-10º a 100º

Tabla 43: PTFA 3415

Mientras que sus medidas serán como sigue:

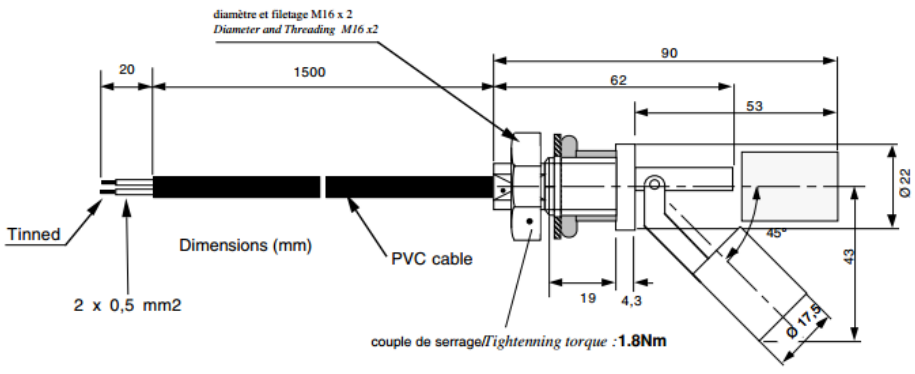

Figura 25: Medidas PTFA3415

4.3.6 Sensor de Agua TANE WS1

Este sensor de agua será empleado para informar al microcontrolador Arduino si el nivel de agua es inferior o no al de la Bomba del chorro de la fuente, evitando daños en la misma. La placa deberá actuar en consecuencia a los datos recibidos para no provocar desperfectos en el sistema.

A diferencia de los sensores de agua tradicionales, no es necesario que sea inmerso en el líquido: podemos unirlo a la superficie externa del contenedor, siempre que este no sea metálico. De este modo se ayuda, primero, a evitar daños en el sensor al estar en contacto con el agua y, segundo, evitamos que el agua se pueda contaminar a mayor velocidad.

Figura 26: Sensor de Agua TANE WS1

Siendo sus medidas como siguen:

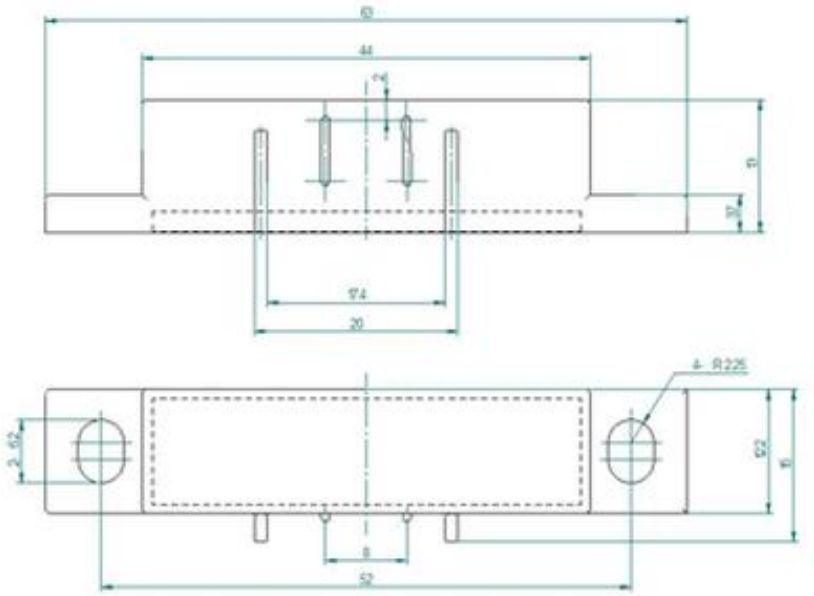

Figura 27: Medidas WS1

4.3.7 Circuito Integrado L293D

El L293D es de gran gran volar a la hora de controlar motores de corriente continua, y en general elementos de cargas de potencia media. Incluye un puente en H cuádruple que, aun pudiéndose crear mediante transistores, resulta de gran comodidad.

A diferencia del modelo L293, el L293D incorpora una serie de diodos para evitar dañar el integrado con las posibles corrientes parásitas generadas. Esto dota de una mayor sencillez su utilización, si bien en el modelo L293 se podría optar por aquellos que mejor se adapten a las cargas o necesidades.

Figura 28: L293D

Su patillaje se distribuye como sigue a continuación:

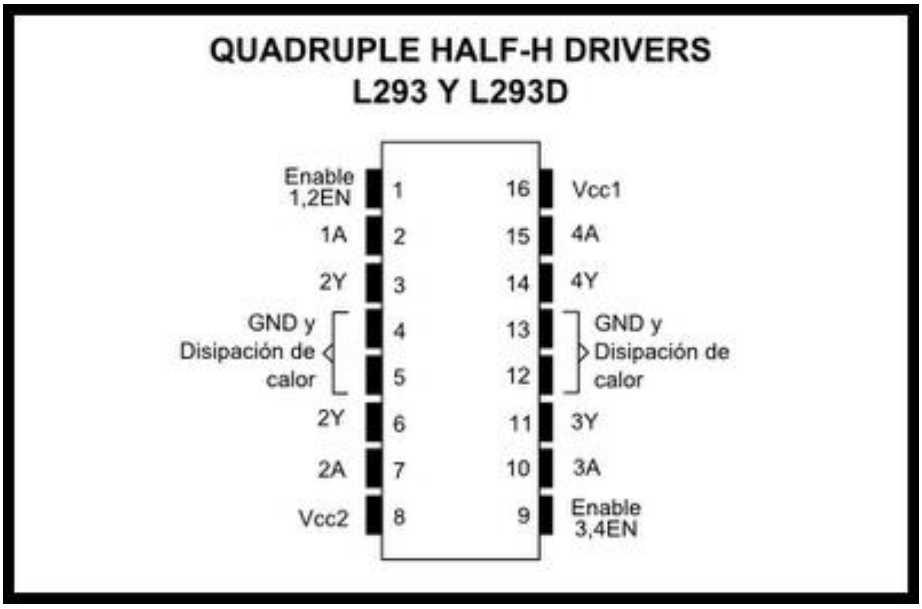

Figura 29: Puente en H Cuádruple (L293D)

Será necesaria la utilización de este circuito integrado para suministrar la potencia necesaria a aquellos elementos que requieran más de 5 voltios para su correcto funcionamiento, como es el caso de la Bomba de Agua. Gracias al Puente en H, se permite no únicamente proporcionar el voltaje necesario, si no también modificarlo para suministrar más o menos según lo optado por el microcontrolador.

Para finalizar, se dispone a presentar algunas de las características técnicas de mayor importancia del circuito integrado, tal y como aparecen reflejadas en su correspondiente ficha técnica:

Rango tensión alimentación Vcc1	36 V
Rango tensión alimentación Vcc2	36 V
Vcc1 Recomendada	4.5 – 7 V
Vcc2 Output Recomendada	Vcc1 – 36 V
Temperatura Funcionamiento	-10º a 100º
Rango de voltaje de Entrada	7 Voltios
Temperatura alcanzada	260º C

Tabla 44: L293D

4.4 Especificación Software

En el presente apartado se va a desarrollar la Especificación Software de la Fuente Arduino. De este modo, para cada uno de los bloques funcionales presentados con anterioridad se presentará su correspondiente Diagrama de Secuencias, seguido de una ligera explicación sobre el funcionamiento del mismo.

4.3.1 Control Automático Nivel del Agua

El diagrama de secuencias asociado a dicha acción será como se muestra a continuación:

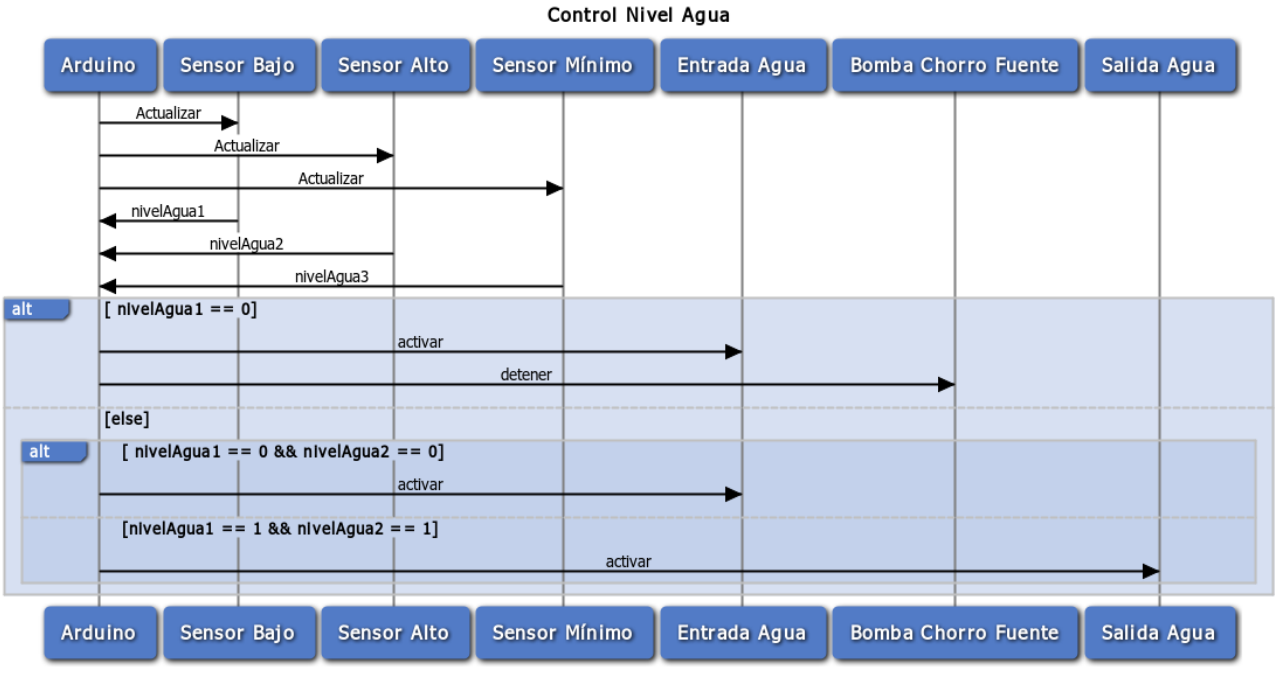

Figura 30: Diagrama Secuencias - Control Nivel Agua

El primero de los pasos será que Arduino reciba los datos de los sensores horizontales encargados de medir el nivel de agua que hay en la fuente en un instante dado. Según los valores digitales (0 o 1) devueltos por dichos sensores se podrá concluir el nivel del agua y actuar en consecuencia.

Los sensores están dispuestos como se indica seguidamente:

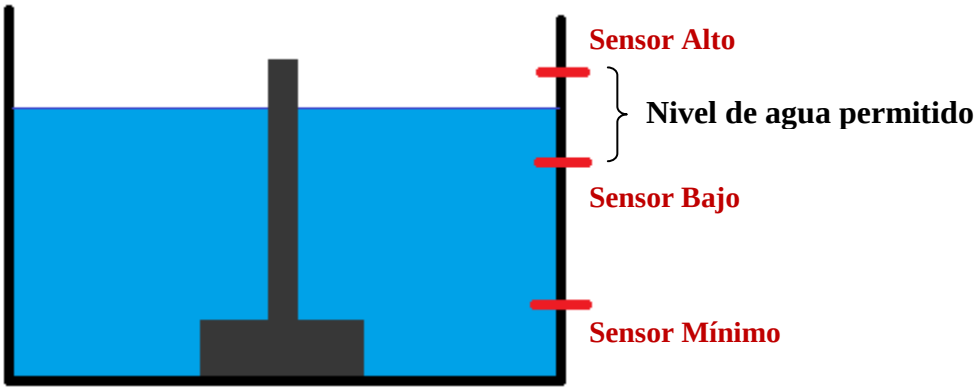

Figura 31: Sensores Control Nivel

Puede verse en el diagrama que lo primero que se comprobará será si el nivel del agua supera el Sensor Mínimo. En caso de no superarlo significa que la bomba puede resultar dañada al tratar de bombear una cantidad de agua que no la cubre por completo, por lo que deberá procederse a la detención de la misma y realizar el llenado correspondiente. En caso contrario, la lógica está dispuesta para que las electroválvulas conectadas al sistema añadan o retiren agua según el nivel de la misma.

4.4.2 Gestión Manual de los Parámetros

Se comenzará presentando el Diagrama de Secuencias asociado a la presente acción:

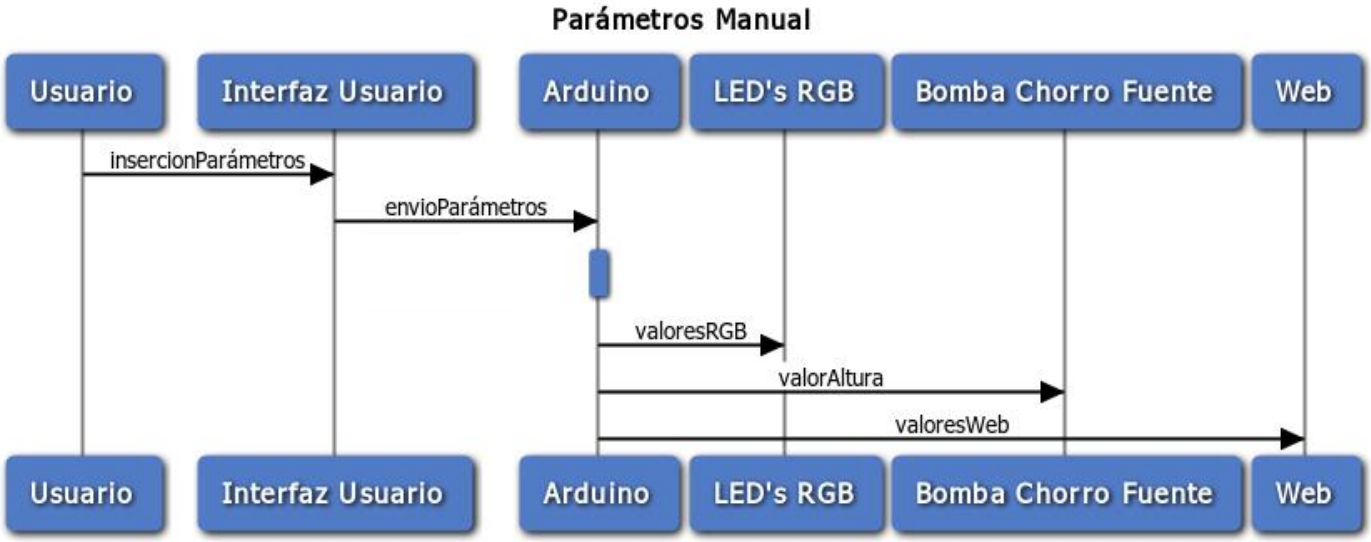

Figura 32: Diagrama Secuencias - Parámetros Manual

La gestión manual de los parámetros se llevará a cabo de manera muy sencilla. Tras la interacción del usuario con la interfaz (recordemos desarrollada en Processing) se elegirán los parámetros que este considere apropiados para un momento dado. La Interfaz de Usuario será la encargada de facilitar dichos parámetros a la placa Arduino mediante Puerto Serie.

Una vez Arduino ha recibido los parámetros introducidos por el usuario, se dispondrá a realizar el procesamiento adecuado de los mismos. Tras obtener, finalmente, los valores asociados a Rojo, Verde, Azul y Altura se activarán las salidas digitales PWM de la propia placa asociadas a

los LED's RGB y la Bomba de Chorro, respectivamente. Finalmente, Arduino tomará estos valores recién actualizados y modificará la página HTML local para que estos mismos aparezcan refrescados.

4.4.3 Detención

El diagrama de secuencias relativo a la Detención será como se presenta:

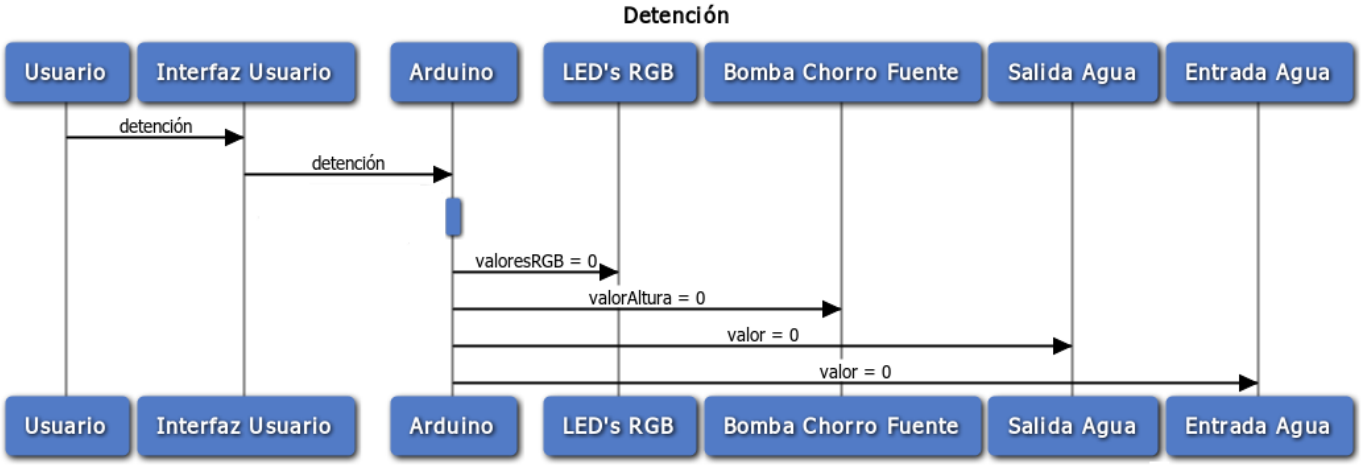

Figura 33: Diagrama Secuencias - Detención

Puede verse que esta acción tiene un funcionamiento muy similar a la variación de los parámetros en modo manual.

El usuario comenzará indicando a la Interfaz que desearía detener el funcionamiento del sistema. Processing se encargará de facilitar esta información al microcontrolador Arduino encargado, el cual realizará el procesamiento correspondiente. Tras concluir que, efectivamente, el usuario quiere detener la fuente, la placa pondrá todos los valores a 0, deteniendo por completo el funcionamiento de la fuente.

4.4.4 Vaciado Forzado

De igual modo que para los casos anteriores, se comenzará presentando el Diagrama de Secuencias de la presente acción:

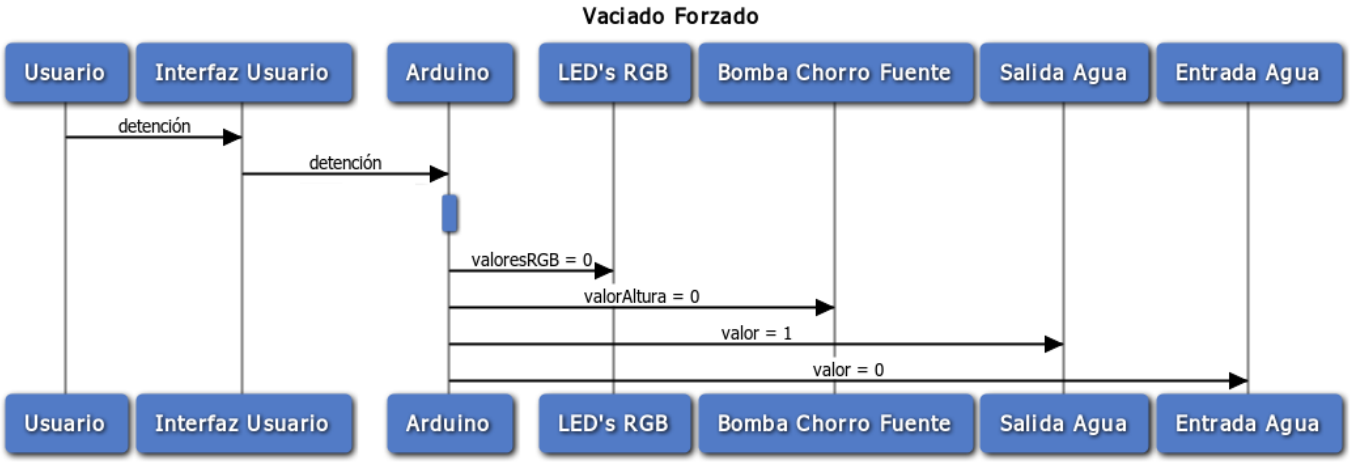

Figura 34: Diagrama Secuencias – Vaciado

En ocasiones será necesario vaciar la fuente, por ejemplo para tareas de mantenimiento. Esta acción será la encargada de llevar a cabo dicho proceso.

Puede verse que, nuevamente, la acción es muy similar a las presentadas anteriormente. Así, el usuario indicará a la interfaz que desea llevar a cabo el vaciado de la fuente. La interfaz facilitará dicha información a la placa Arduino, que procesará dichos datos y concluirá la acción a realizar.

Tras concluirse que, efectivamente, la acción a realizar corresponde a realizar el vaciado de la fuente, Arduino lanzará por sus salidas digitales PWM valores a 0 tanto para los LED RGB como para la altura del chorro de la fuente. Además, se forzará el cierre de la entrada de agua, mientras que se permitirá la salida de esta.

4.4.5 Gestión Parámetros mediante Previsión Meteorológica

Se dispone a continuación a presentar el Diagrama de Secuencias asociado a la presente acción:

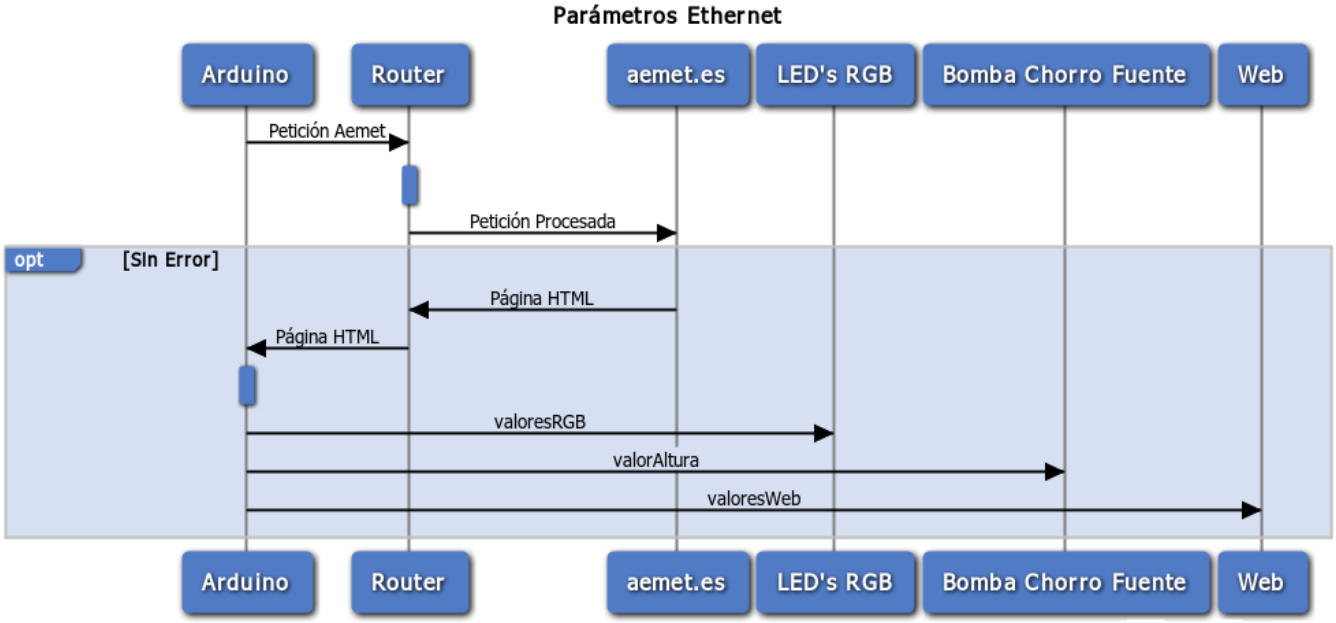

Figura 35: Diagramas Secuencias - Parámetros Ethernet

Al escoger el presente modo, el sistema se encargará de realizar periódicamente peticiones al servidor meteorológico externo, de manera que los datos de Temperatura y Probabilidad de Lluvia sean susceptibles a posibles cambios. Cabe señalar que el sistema realizará una petición cada 30 segundos, tratándose de evitar la sobrecarga tanto del sistema como del propio servidor meteorológico.

Así pues, la acción comenzará con el microcontrolador Arduino generando una web HTML en la que se incluirá una petición adecuada al servidor meteorológico al que va destinada. El router será el encargado de procesar y enviar dicha petición por la red hasta llegar a su destino, que no es otro que la web oficial de la Agencia Estatal de Meteorología [14].

Arduino recibirá como consecuencia de esta petición una web HTML, que incluirá el error en caso de haberlo o los resultados de la consulta. En caso de error el microcontrolador no llevará a cabo ninguna tarea más, no pudiéndose realizar la actualización de los valores.

En caso de poder realizarse la petición correctamente, Arduino recibirá y realizará el parseo correspondiente para recibir los parámetros que le resultan de interés: la Temperatura y la Probabilidad de Lluvia. Tras obtenerlos se operará con dichos resultados, obteniéndose unos valores de Rojo, Verde, Azul y Altura basados en estos. Finalmente, Arduino enviará los valores recién generados por sus salidas correspondientes a los LED's y a la Bomba de Agua correspondiente, así como generar la nueva Web local.

4.5 Conclusiones

Tras presentarse en el apartado anterior las tareas que nuestro sistema podía llevar a cabo, se ha procedido a desarrollar los diferentes algoritmos que emplearemos para que, efectivamente, la fuente pueda llevar a cabo dichas acciones.

Se ha comenzado desarrollando la topología del sistema, de manera que pudiera entenderse con mayor comodidad cómo se llevaba a cabo la agrupación de los diferentes componentes utilizados para el desarrollo de la fuente.

A continuación han sido presentados todos los componentes utilizados para el desarrollo del prototipo. Lógicamente, en base a estos componentes hardware utilizados se deberá adaptar toda la circuitería de la manera adecuada. Recordemos que se ha optado por la utilización de elementos de bajo coste económico y energético, haciendo el proyecto lo más ajustado posible.

Finalmente, y tras presentarse los diferentes diagramas de secuencias asociados a la especificación software de nuestro sistema,

5. Implementación

5.1 Introducción

A continuación se procede al desarrollo de todos los aspectos relacionados para llevar a cabo la implementación e implantación de la Fuente Arduino. De este modo, se llevará a análisis los aspectos de mayor importancia tanto a nivel Hardware como a nivel Software de mencionado Proyecto.

De este modo, se comenzará presentando todos los aspectos relacionados con el Hardware de nuestro proyecto. Se tratará de clarificar los elementos requeridos a la hora de desarrollar un prototipo funcional de la Fuente Arduino aquí propuesta. Para llevar esto a cabo se presentarán una serie de capturas tanto de los diferentes circuitos como del prototipo implementado.

Una vez finalizado este punto se procederá a analizar los aspectos más relevantes del Software del proyecto, tanto del código a cargar en el microcontrolador Arduino como el implementado en Processing para el desarrollo de la Interfaz de Usuario. Se presentarán además las diferentes capturas de pantalla que podremos encontrarnos en el prototipo software desarrollado.

5.2 Implementación Hardware

Se comenzará por tanto llevando a cabo el análisis de la conexión de los diferentes componentes hardware necesarios para la implementación del sistema. Este proceso será llevado a cabo mediante módulos separados, de manera que las diferentes partes del proyecto aquí desarrollado puedan desarrollarse con mayor claridad. Lógicamente, de cara al producto final, estos componentes deberían unirse en uno único y con total funcionalidad.

5.2.1 Conexión a Internet

Para permitir a la placa Arduino realizar la conexión a Internet con la que realizar el acceso al servidor meteorológico únicamente será necesario acoplarle una Ethernet Shield compatible a la mismo.

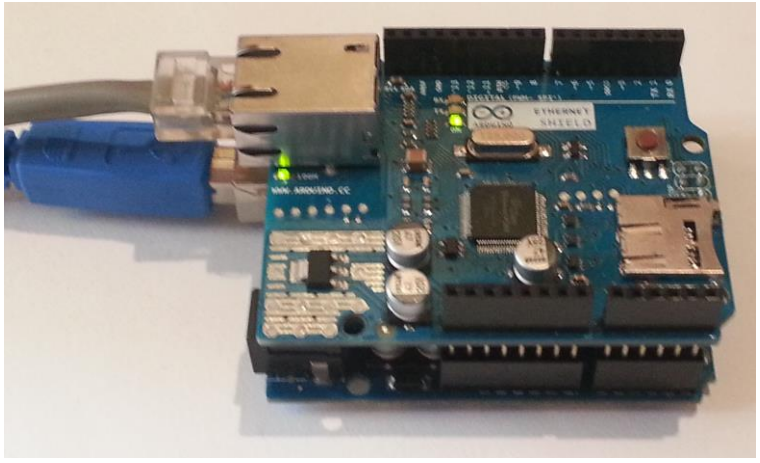

Figura 36: Arduino + Ethernet Shield

Deberemos asegurarnos que el cable Ethernet está conectado a un router con acceso a Internet, así como comprobar que todas las patillas están bien situadas. El montaje se realizará sobre dicha Shield.

5.2.2 Gestión del Agua

Gracias a los diferentes sensores que el microcontrolador tendrá incorporados se tendrá la capacidad de regular el nivel del agua. Tal y como se ha mencionado con anterioridad dispondremos de tres sensores, cada uno encargado de llevar a cabo una tarea.

Para realizar esta conexión deberemos tener bien presente que será necesario incluir una serie de resistencias de Pull-up (a +5V) o Pull-down (a GND) para evitar que las entradas digitales de Arduino registren valores anómalos.

El esquema para esta implementación será como sigue:

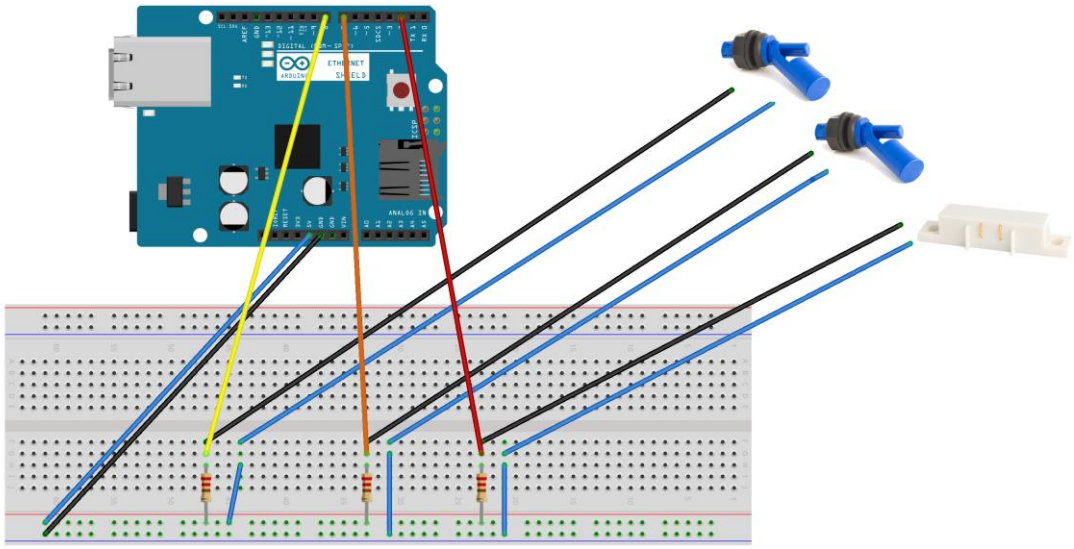

Figura 37: Esquema Nivel Agua

Dicho esquema, tras ser implementado, resultará como sigue a continuación:

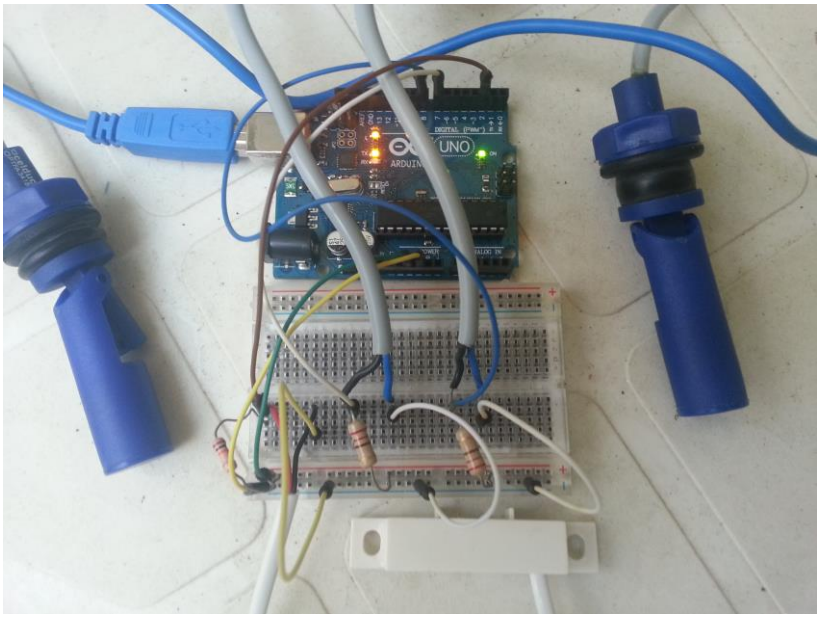

Figura 38: Implementación Nivel Agua

Para poder llevar a cabo el vaciado/llenado de la fuente será necesario disponer de electroválvulas controladas por el propio microcontrolador. Gracias a que necesitamos sencillamente que estén activadas o cerradas, únicamente tendremos que conectar dichos componentes a un relé, al que incluiremos la salida digital de Arduino, GND y el voltaje necesario para su correcto funcionamiento.

5.2.3 Iluminación

Independientemente de la alternativa escogida a la hora de actualizar los diferentes parámetros de la fuente, ésta debe actuar en consecuencia a los valores de los mismos. Uno de estos conjuntos de parámetros serán aquellos que indican al sistema de iluminación RGB la cantidad de Rojo, Verde y Azul será requerida en un momento dado.

De este modo, se tendrán que conectar las salidas digitales de Arduino a los correspondientes canales RGB del sistema de iluminación empleado. Nótese que las salidas digitales de la propia placa deben ser salidas PWM para que funcione correctamente.

De este modo, el circuito será como sigue:

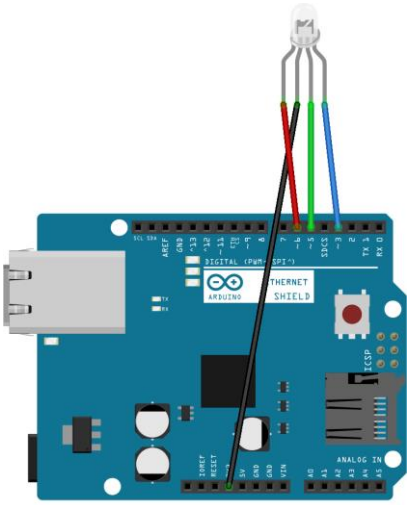

Figura 39: Esquema Iluminación

Cuyo resultado de la implementación será como sigue:

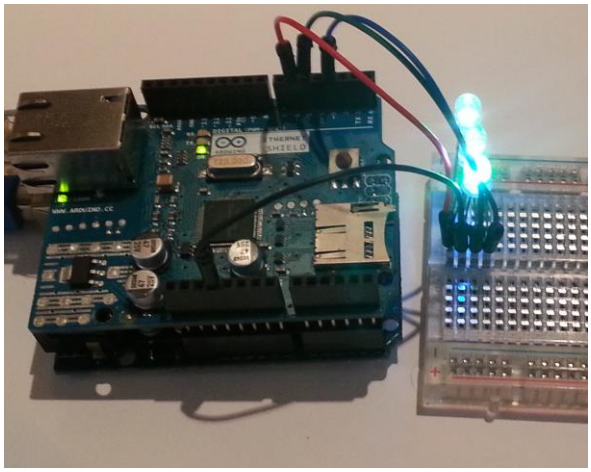

Figura 40: Implementación Iluminación

5.2.4 Altura

Debido a que, en líneas generales, las bombas son elementos que requieren mayor potencia que los 5 voltios que Arduino puede suministrar será necesaria una alimentación externa a la que conectarla. Dicha conexión será posible gracias al circuito integrado L293D y sus diferentes puentes en H que dispone.

De este modo, y teniendo en cuenta la distribución de las patillas de dicho circuito integrado, el esquema para la implementación será como sigue:

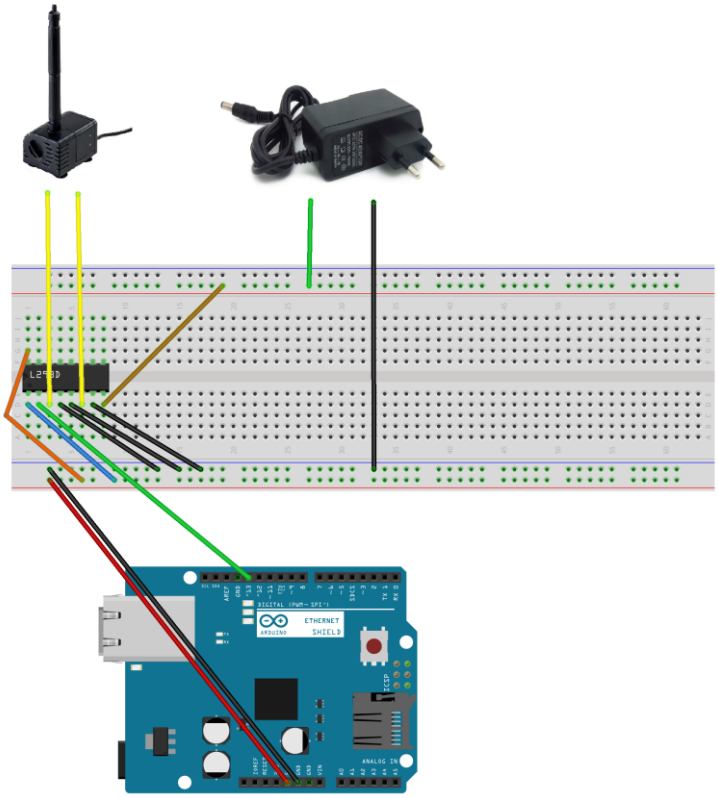

Figura 41: Esquema Altura

El resultado de esta implementación se muestra a continuación:

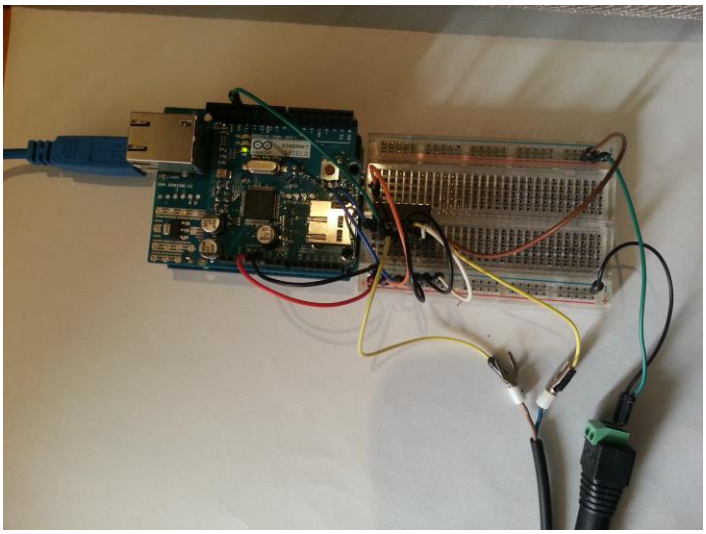

Figura 42: Implementación Altura

Tras desarrollar dicho circuito se tendrá la posibilidad de suministrar el voltaje necesario a la fuente y gestionarlo mediante el microcontrolador Arduino. Puede verse que tanto las entradas de habilitación como las de control que no necesitamos han sido conectadas a 5V o GND, respectivamente.

El resultado por tanto será como sigue:

Figura 43: Fuente en Funcionamiento

5.3 Implementación Software

Tras desarrollarse el análisis en profundidad de los aspectos más relevantes del Hardware de la Fuente Arduino se dispone a llevar a cabo el desarrollo pormenorizado del Software del mismo, tal y como se ha explicado.

Para ello, se irán presentando cada una de las características de mayor importancia o complejidad del proyecto, realizando un análisis detallado sobre las mismas.

5.3.1 Interfaz de Usuario

La Interfaz de Usuario será la encargada de permitir la interacción entre el Usuario y la placa Arduino encargada de controlar la fuente. Es fundamental por tanto el desarrollo de una Interfaz de Usuario sencilla e intuitiva de utilizar, permitiéndose así comunicar al microcontrolador de dónde realizará la obtención de sus parámetros.

Debido a la gran sencillez que este presenta a la hora de llevar a cabo la comunicación con Arduino mediante Puerto Serie, se ha optado por la utilización de Processing para desarrollar la Interfaz de Usuario. Se recuerda que se puede realizar la descarga del mismo desde la página oficial [11].

Para este proyecto se ha optado por el empleo de la librería ControlP5. Dicha librería incluye numerosos controladores a ser presentados por la Interfaz de Usuario de manera muy sencilla, facilitando gran parte del proceso de desarrollo de la misma. Puede descargarse desde la página oficial [12], donde se incluye toda la documentación necesaria.

La interfaz resultante será como se muestra:

Figura 44: Interfaz Fuente Arduino (Processing)

En este caso, tanto el Slider horizontal de Altura, las perillas del color como el botón Actualizar han sido implementadas gracias a la librería ControlP5 mencionada con anterioridad. Vamos a ver cómo se ha realizado su implementación:

```
cp5 = new ControlP5(this);
cp5.addSlider("Altura", 0, 100, 0, 125, 315, 250, 30);
cp5.addKnob("Rojo")
     .setRange(0,255)
     .setValue(125)
     .setPosition(150,360)
     .setRadius(25);
 cp5.addKnob("Verde")
     .setRange(0,255)
     .setValue(125)
     .setPosition(220,360)
     .setRadius(25);
 cp5.addKnob("Azul")
     .setRange(0,255)
     .setValue(125)
     .setPosition(290,360)
     .setRadius(25);
cp5.addButton("Actualizar")
    .setPosition(390,360)
    .setSize(70,50);
```

Código 1: Interfaz ControlP5

Para trabajar con la librería ControlP5 es necesaria la creación de un objeto de tipo ControlP5, en nuestro caso de nombre **cp5**. Sobre este mismo objeto se irán añadiendo los diferentes elementos de la interfaz. Cada uno de estos elementos se identifica con un nombre, y se le añaden los diferentes parámetros para adaptarlos a las necesidades de la propia Interfaz. A cada uno de estos identificadores se le puede asignar un método relacionado con su funcionalidad, de manera que asignaremos a las variables necesarias el contenido de dichos elementos.

5.3.2 Envío de Datos

En caso de seleccionar el Modo Manual, el usuario (y por tanto la interfaz desarrollada por Processing) será el encargado de establecer los parámetros como considere oportuno. Será esta interfaz la que realizará el envío de los datos recién introducidos hacia Arduino. Lógicamente, ambos elementos deberán compartir un protocolo tanto para enviar como para recibir, de manera que ambas plataformas puedan entenderse.

Para este caso se ha optado por realizar estos envíos mediante un array de 5 posiciones, a las cuales tienen asignado un valor. Esto puede verse con mayor facilidad gracias al diagrama presentado a continuación:

Figura 45: Array de Datos

Cada una de estas posiciones permite la adición de un número entero que se encuentre entre 0 y 255. El campo de Modo, concretamente, se codifica como sigue:

0	Detención
1	Vaciado Forzado
2	Modo Manual
3	Modo Ethernet (Previsión Meteorológica)

Tabla 45: Codificación Datos

En el caso del Modo Manual, los campos Altura, Rojo, Verde y Azul se rellenarán convenientemente entre los rangos establecidos. Sin embargo, si elegimos cualquiera de los otros modos disponibles estos campos estarán rellenos con un 0, indicándole a la placa Arduino que estos campos no son necesarios. Es decir, Arduino realizará la lectura de dicho array desde la primera posición, y actuará en consecuencia al modo que encuentre. Cabe señalar que esta asignación será llevada a cabo por Processing.

Este envío de datos, por tanto, será como se muestra a continuación:

```
byte out[] = new byte[5];
out[0] = byte(2);
out[1] = byte(altura);
out[2] = byte(rojo);
out[3] = byte(verde);
out[4] = byte(azul);
port.write(out);
```

Código 2: Envío de Datos

5.3.3 Recepción de Datos

Tal y como se ha mencionado con anterioridad, mientras que Processing se encargará de llevar a cabo el envío de los datos siguiendo la codificación desarrollada, será el microcontrolador Arduino quien deberá recibir dichos datos y actuar en consecuencia.

De este modo, al comienzo de cada bucle de control se realizará la lectura por Puerto Serie en busca de nuevos datos, que signifiquen que Processing ha actualizado los valores. Tras realizarse (o no, en caso de no haber nuevos datos) dicha actualización, se analizará el primero de los valores y se actuará en consecuencia. Justo después, se realizará la creación de la página web local.

El código resultante será como sigue:

```
// Comprobar que se ha recibido un array de 5 elementos
if (Serial.available() > 4) {
  // Leemos los elementos del array:
  for (int i=0; i<=4; i++) {
    incomingByte[i] = Serial.read();
  }
}
/*
En base a la primera posición del Array (código) recibido
por Processing lanzamos los diferentes métodos.
*/
if(incomingByte[0] == codigoEthernet){
  //Se permite la entrada cada 30 segundos (sobrecarga)
       tiempoActual = millis();
       Serial.println(tiempoActual - tiempoAnterior);
       if(tiempoActual - tiempoAnterior >= retardo) {
         tiempoAnterior = tiempoActual;
         conexion();
       }
}
if(incomingByte[0] == codigoManual){
       manual();
}
 if(incomingByte[0] == codigoStop){
       detener();
}
 if(incomingByte[0] == codigoVaciado){
       vaciar();
}
webCreation();
```

Código 3: Recepción de Datos

Puede verse que, tal y como ha sido mencionado con anterioridad y para evitar la sobrecarga tanto del sistema como del servidor externo, se añade un umbral que impide que se realicen conexiones durante un intervalo de tiempo (retardo). En nuestro caso se ha optado por establecerlo en 30 segundos.

5.3.4 Configuración Inicial. Persistencia

Aprovechando la memoria EEPROM que Arduino pone a nuestra disposición podremos realizar la carga y el almacenamiento de los datos obtenidos en un momento dado, de modo que estos persistirán aun desconectando el microcontrolador. De este modo, tendremos la persistencia necesaria.

La capacidad de esta EEPROM depende del modelo de la placa Arduino. En nuestro caso, dispondremos de 1024 bytes para realizar los almacenamientos. Esta cantidad será suficiente para las necesidades de nuestro proyecto.

De este modo, al realizar la carga al arrancar la placa se incluirán también los datos que había anteriormente. Gracias a que sabemos cómo se realiza el envío y recepción de datos, estos almacenamientos y posterior carga se realizarán de manera muy sencilla.

Lo primero que tendremos que hacer será la importación de la librería EEPROM, incluida por defecto en el IDE de Arduino.

```
#include <EEPROM.h>
```

Código 4: Include EEPROM

Para realizar la carga inicial deberemos utilizar la función **read** de la propia librería. Esta recibe un único parámetro: un entero que indicará la posición en la cual se realizará la carga. De este modo, esta carga se realizará como sigue a continuación:

```
incomingByte[0] = EEPROM.read(0);
incomingByte[1] = EEPROM.read(1);
incomingByte[2] = EEPROM.read(2);
incomingByte[3] = EEPROM.read(3);
incomingByte[4] = EEPROM.read(4);
```

Código 5: EEPROM Carga Inicial

Es fácil ver que realizaremos esta carga sobre el mismo array de 5 posiciones sobre el que almacenamos los diferentes parámetros de la recepción de Processing.

Finalmente, deberemos asegurarnos que, cada que vez que (independientemente del motivo) alguno de estos parámetros se modifique realizaremos el almacenamiento en la posición correspondiente. Esta será llevado a cabo mediante la función **write** de la librería EEPROM, que recibe la posición de memoria en la que se almacenará y el dato correspondiente. Así, se llevará a cabo del siguiente modo:

```
EEPROM.write(0,incomingByte[0]);
EEPROM.write(1,incomingByte[1]);
EEPROM.write(2,incomingByte[2]);
EEPROM.write(3,incomingByte[3]);
EEPROM.write(4,incomingByte[4]);
```

Código 6: EEPROM Escritura

5.3.5 Comprobación Nivel del Agua (Seguridad)

El sistema dispone de una serie de sensores encargados de monitorizar los niveles de agua, pudiendo actuar en consecuencia para tratar de asegurar que éstos se mantienen dentro de un intervalo aceptable.

La entrada y salida de agua se lleva a cabo mediante electroválvulas conectadas a un relé, decidiéndose cuándo deben activarse y cuándo no en base a dichos niveles.

La monitorización de los valores del nivel del agua se lleva a cabo como puede verse:

```
delay(75);
int valorAlto = digitalRead(horizontalAlto);
int valorBajo = digitalRead(horizontalBajo);
int valorSeguridad = digitalRead(seguridad);
if(valorSeguridad == 1){
  detener();
  llenar();
}
else{
  if(valorAlto == 1 && valorBajo == 0){
    //Bucle principal...
  }
  else if(valorAlto == 1 && valorBajo == 1){
    llenar();
  }
  else if(valorAlto == 0 && valorBajo == 0){
    vaciar();
  }
}
```

Código 7: Control Nivel Agua

Para llenar y vaciar el recipiente únicamente se tendrá que añadir un valor HIGH o LOW al relay encargado de gestionar dichas entradas o salidas, respectivamente.

5.3.6 Generación Página Web Local

Justo al finalizar todas las acciones del bucle principal, y tras actualizar por tanto los valores de los parámetros de la fuente, nos dispondremos a generar una página web gracias a que el microcontrolador Arduino dispone de una Ethernet Shield acoplada. Dicha web podrá ser accedida desde cualquier dispositivo conectado a su misma red.

Dicha página web tendrá el siguiente aspecto:

Figura 46: Captura Web Local Escritorio

A la que, como acaba de mencionarse, puede accederse desde cualquier dispositivo que esté conectado a su misma red, como el caso del siguiente Teléfono Móvil:

Figura 47: Captura Web Móvil

Para acceder al website recién generado únicamente tendremos que abrir el navegador Web utilizado habitualmente y teclear la dirección IP que Arduino tenga asignada dentro de la red.

La web será generada en lenguaje HTML. Además, con la intención de dotarle de un aspecto más llamativo sin complicar excesivamente su código se ha optado por emplear Jquery Mobile. Se ha escogido esta alternativa debido a que permite generar websites de calidad que pueden ser visualizados desde cualquier dispositivo móvil, que es la idea principal.

Lo primero que se tendrá que hacer, por tanto, será inicializar el cliente y, una vez realizada dicha conexión, proceder a la generación de dicha web. Tras finalizar, deberemos asegurarnos cerrar la conexión, permitiéndose actualizar los valores en la siguiente pasada del bucle de control.

Tras comprobar, por tanto, la disponibilidad y la conexión del cliente generaremos la página en formato HTML y el framework Jquery Mobile. Un fragmento de dicho código será como sigue:

```
webClient.println("HTTP/1.1 200 OK");
webClient.println("Content-Type: text/html");
webClient.println();
//Página web en formato HTML
webClient.println("<html>");
webClient.println("<head>");
webClient.println("<title>Fuente Arduino</title>");
webClient.print("<link rel=\"stylesheet\" href=\"http://code.jquery.com");
webClient.println("/mobile/1.2.0/jquery.mobile-1.2.0.min.css\" />");
webClient.print("<script src=\"http://code.jquery.com");
webClient.println("/jquery-1.8.2.min.js\"></script>");
webClient.print("<script src=\"http://code.jquery.com/mobile");
webClient.println("/1.2.0/jquery.mobile-1.2.0.min.js\"></script>");
webClient.println("</head>");
webClient.println("<body>");
webClient.println("<div data-role=\"header\" data-theme =\"a\"> ");
webClient.println("<h1>Resultados Fuente Arduino</h1>");
```

Código 8: Generación HTML

5.3.7 Servidor Meteorológico Externo

Se ha repetido en diferentes ocasiones que la Ethernet Shield conectada al microcontrolador Arduino ha de encargarse de generar una petición a un servidor externo, recibir la web de respuesta y facilitársela a la placa. Será ésta quien deberá llevar a cabo el parseo de los datos que resultan de interés: en nuestro caso, se deberá obtener la temperatura y la probabilidad de lluvia predichas por mencionado servidor.

Figura 48: Petición-Respuesta-Parsing

Se comenzará analizando por tanto el primero de estos aspectos: la petición. Antes de comenzar será necesario realizar la conexión., en este caso a un servidor externo. Esta conexión será realizada por el puerto 80.

Tras llevarse a cabo dicha conexión y comprobar la disponibilidad, se realizará la conexión de manera similar a cuando generábamos nuestra página a nivel local. El fragmento de código responsable será, por tanto, como sigue:

```
if (client.connect(servidor, 80)) {
   client.print("GET /es/eltiempo/prediccion/municipios");
   client.println("/valencia-id46250#detallada HTTP/1.1");
   client.println("Host: www.aemet.es");
   client.println("Connection: close");
   client.println();
}
```

Código 9: Petición Servidor Externo

Puede verse que dicha petición está realizada siguiendo el estilo estipulado por el protocolo HTTP, de manera que el host tenga la capacidad de entenderla. De este modo, se realiza un GET con la página recurso que queremos obtener del servidor externo **www.aemet.es**. En dicho recurso es donde encontramos la previsión meteorológica de una localidad concreta: en nuestro caso se ha optado por Valencia.

Tras realizarse la petición, recibiremos en la misma variable ***client*** empleada para la misma la respuesta obtenida. En caso de acierto recibiremos una web que deberemos parsear, mientras que si obtenemos error recibiremos código de error HTTP.

Analizando la web objetivo se podrá llevar a cabo el parseo apropiado. Recordemos que se está buscando obtener los valores de Temperatura y Probabilidad de Lluvia.

El primero de los parámetros se obtendrá como sigue:

```
String cadena = "";
salida = false;
if(client.connected()){
    client.find("<th title=\"Temperaturas");
    client.find("<td class=\"borde_rb no_wrap\">");
    while (client.connected() && salida == false) {
        if (client.available() && salida == false) {
          char c = client.read();
          if(c != '&'){
            cadena.concat(c);
          }
          else {
            salida = true;
          }
        }
      } //del while connected
  } //del if(client.connected())
  temperatura = cadena.toInt();
```

Código 10: Parseo Temperatura

Mientras que la Probabilidad de Precipitación se obtendrá como se muestra a continuación:

```
cadena = "";
salida = false;
if(client.connected()){
    client.find("Probabilidad de precipitación");
    client.find("<td class=\"borde_rb\" >");
    while (client.connected() && salida == false) {
        if (client.available() && salida == false) {
          char c = client.read();
          if(c != '&'){
            cadena.concat(c);
          }
          else {
            salida = true;
          }
        }
      } //del while connected
  } //del if(client.connected())
prec = cadena.toInt();
```

Código 11: Parseo Prob. Precipitación

5.4 Conclusiones

Tras presentarse en el apartado Diseño del Sistema los diferentes elementos hardware necesarios para el proyecto, se ha procedido en el presente punto todos los pasos necesarios para realizar su agrupación de la manera apropiada, obteniéndose como resultado la Fuente Arduino.

Presentado mediante bloques, se ha comenzado presentando los diferentes componentes hardware relativos a los diferentes aspectos relacionados con las acciones de la fuente. De este modo, se han desarrollado cada una de las conexiones de estos componentes de manera que puedan comunicarse apropiadamente con el microcontrolador Arduino y llevar a cabo las acciones que tengan asociadas.

Aclarados estos conceptos se ha realizado el análisis del software necesario para el correcto funcionamiento de la Fuente Arduino. Cada uno de estos fragmentos sometidos a estudio disponía de una serie de peculiaridades que se ha tratado de explicar a fondo.

Todo el contenido desarrollado a lo largo del presente punto ha tratado de aclarar todas las posibles dudas o dificultades que pueden surgir a la hora de realizar la implementación e implantación de la Fuente Arduino. Gracias al contenido presentado se tendrá la capacidad, por tanto, de desarrollar una fuente inteligente con las características desarrolladas a lo largo de la presente memoria.

6. Conclusiones

6.1 Trabajo realizado

La Fuente Arduino, uno de los componentes del proyecto ArduEntorno, es el fruto del trabajo realizado durante los últimos meses. El resultado es, como se ha mencionado repetidas veces a lo largo del presente documento, una fuente de jardín inteligente programable con una serie de funcionalidades controladas gracias al microcontrolador Arduino.

Una de las tareas fundamentales pues ha sido desarrollar el programa cargado sobre dicha placa. Este código permite al microcontrolador actuar en base a los parámetros que reciba, activando los actuadores necesarios según el caso. Especialmente relevante es la conexión al servidor meteorológico externo llevada a cabo gracias a la Ethernet Shield, así como el parseo de la página HTML recibida de esta.

Otro detalle de gran importancia radica en el desarrollo de la Interfaz de Usuario mediante Processing. Esta Interfaz es la encargada de recibir e interpretar las acciones del Usuario e informar al microcontrolador los deseos del mismo. El código se encargará, primero, de presentar la Interfaz al Usuario y, segundo, de realizar dicha comunicación a Arduino vía Puerto Serie.

Ha sido necesario también implementar el Hardware necesario para el correcto funcionamiento de la Fuente Arduino. De este modo se han detallado todos los circuitos necesarios a la hora de desarrollar este sistema, de manera que dicho hardware pueda funcionar como ciertamente corresponde. Así pues, además de las propias conexiones realizadas sobre la placa Arduino, se han planteado los diferentes aspectos necesarios para suministrar a los diferentes componentes del sistema una cantidad mayor a los 5 Voltios que Arduino es capaz de suministrar.

Una de las mayores dificultades que tiene el desarrollo de la Fuente Arduino ha sido, sin duda, el hecho de estar trabajando con agua y electricidad. Toda precaución era poca al manipular los diferentes componentes sabiendo que cualquier leve contacto con el agua podría suponer un grave problema. Lógicamente, esto es un problema que ya se sabía mucho antes de tan siquiera comenzar con el desarrollo del proyecto.

Además, el aspecto Hardware del proyecto ha provocado que se haya requerido una circuitería relativamente compleja, de modo que se ha tenido que andar con mucha precaución a la hora de trabajar con el prototipo funcional.

6.2 Aportaciones

A continuación se dispone a analizar las diferentes aportaciones realizadas durante el desarrollo del proyecto. Se plantearán por tanto todos los aspectos de mayor innovación de la Fuente Arduino, analizando su impacto de cara a desarrollar proyectos de mayor o menor similitud al mismo.

Se ha comenzado el documento realizando un análisis detallado de diferentes sistemas similares al aquí presentado. De este modo, este estudio puede resultar de gran interés de cara a ubicar proyectos que guarden rasgos en común con estos. Recordemos que dichos sistemas son lo

suficientemente diferenciados entre sí que puede resultar de gran interés para proyectos de diversa índole.

Por otro lado, se ha llevado a cabo una Especificación de Requisitos de un sistema que emplea conjuntamente Internet, Servicios Web y Hardware de Control. Las alternativas aquí presentadas no son habituales en los diferentes sistemas que podemos encontrarnos. Esta especificación será de mucha utilidad de cara a desarrollar proyectos similares al desarrollado durante el presente documento, pudiendo aprovecharse elementos aquí presentados.

Se ha diseñado, además, un sistema integral que emplea conjuntamente Internet (recuperar la información meteorológica de un servidor externo), Servicios Web (intercomunicación) y hardware de control (microcontrolador Arduino). Cabe señalar que puede llevarse a cabo la conexión a Internet gracias a la Ethernet Shield conectada sobre la propia placa.

Finalmente, y como elemento más resaltado, se ha aportado una fuente de jardín inteligente programable completamente funcional. Esta fuente está controlada gracias a la placa Arduino, sobre la que se conectarán los diferentes sensores y actuadores. Tal y como se ha comentado en diferentes ocasiones a lo largo del documento actual, el aspecto de la fuente dependerá de los diferentes parámetros obtenidos en un momento dado.

6.3 Ampliaciones

El proyecto presenta la posibilidad, tal y como se ha mencionado en diferentes ocasiones, de ser ampliado. Es sencillo ver que esta ampliación puede realizarse enfocada hacia diferentes perspectivas. A continuación se presentarán algunas alternativas que podrían resultar de interés de cara al desarrollo de una fuente con un mayor número de funcionalidades que la aquí desarrollada.

Para comenzar, podría resultar de gran interés añadir diferentes elementos hardware que permitieran el control de la fuente sin necesidad de emplear la interfaz desarrollada mediante Processing. De este modo podría optarse por sustituir o acompañar dicha UI con componentes tales como pantallas o actuadores, que nos permitirían interactuar con los diferentes elementos de la fuente.

Otra alternativa sería añadir nuevos componentes hardware que afectaran el comportamiento o el aspecto de la fuente en base a sus propios parámetros. Añadiendo una mayor cantidad de bombas de agua o elementos decorativos, por poner algunos ejemplos sencillos, podría conseguirse una fuente todavía más vistosa que la desarrollada a lo largo del presente documento.

Obviamente, es fácil pensar que al añadir nuevos componentes hardware al sistema sería interesante ampliar las funcionalidades software con las que poder suministrar nuevos parámetros a estos. Bien con la incorporación de nuevos sensores, haciendo el parseo adecuado sobre el website del servidor meteorológico externo o cualquier alternativa que se elija se puede incrementar el número de funciones desarrolladas por la Fuente Arduino. Por poner algún ejemplo, pueden reproducirse diferentes tipos de música o sonidos en base a los parámetros que dispone la fuente en un momento dado. En caso de ampliar estas funciones, lo más habitual sería también modificar la Interfaz de Usuario para adaptarla a estos casos.

Puede verse, con un sencillo análisis, que la cantidad de ampliaciones a realizar sobre la Fuente Arduino pueden ser muy diversas. El abanico de posibilidades es francamente amplio, tocando

diferentes temas, y podría ser interesante incorporar alguna de estas alternativas según las necesidades del usuario.

Debido al enfoque abiertamente Open Source del proyecto, debería considerarse utilizar herramientas de este tipo a la hora de desarrollar las diferentes ampliaciones. Cualquier alternativa no Open Source debería no ser, en un principio, tomada en cuenta salvo que fuera estrictamente necesario, pues esto iría en contra de la naturaleza del proyecto.

7. Referencias

[1] Irrduino. An Arduino-based irrigation control system. *https://code.google.com/p/irrduino/*

[2] JosemaTM. Estación Meteorológica inalámbrica con Arduino. Proyecto Meteo. *http://www.josematm.com/estacion-meteorologica-inalambrica-con-arduino/*

[3] NetIO. Controller Aplication. *http://netio.davideickhoff.de/*

[4] Arduino. http://www.arduino.cc/

[5] Blog Personal de Joe Fernández. *http://joeshacks.blogspot.com.es/*

[6] No-IP Página Principal. *http://www.noip.com/*

[7] Diagramas de Secuencias. *https://www.websequencediagrams.com/*

[8] Especificación Arduino. *http://arduino.cc/en/Main/arduinoBoardUno*

[9] Espec. Ethernet Shield. *http://arduino.cc/en/Main/ArduinoEthernetShield*

[10] Ficha Técnica LED RGB. *http://arduino.cc/documents/datasheets/LEDRGB-L-154A4SURK.pdf7*

[11] Lenguaje de Programación Processing. *http://processing.org/*

[12] Librería ControlP5 para el desarrollo de Interfaces de Usuario. *http://www.sojamo.de/libraries/controlP5/*

[13] yuml.me Desarrollo Casos de Uso. *http://yuml.me/*

[14] Agencia Estatal de Meteorología (AEMET). *http://www.aemet.es/es/portada*

[15]Fuente Cibernética Parquesur. *http://www.parquesur.com/W/do/centre/fuente-cibernetica*

[16] L293D Texas Instruments Datasheet. *http://pdf1.alldatasheet.com/datasheet-pdf/view/27189/TI/L293D.html*

[17] Desarrollo de esquemas con Fritzing. *http://fritzing.org/home/*

[18] Lámpara RGB desarrollada por Miguel Ángel de Frutos. *http://madebyfrutos.wordpress.com/2013/01/07/my-rgb-lamp/*