

# Sistema electrónico para el control de riego automático

*Jose Ignacio Damigo Diaz*

*Universidad de Sevilla*

*Escuela superior de ingenieros*

*Departamento de Ingeniería Electrónica*



# Contenido

<b>Sistema de riego automático para huerto urbano controlado por arduino .....</b>	<b>i</b>
<b>Capítulo 1. Introducción .....</b>	<b>1</b>
<b>Capítulo 2. Análisis del problema.....</b>	<b>4</b>
<b>Capítulo 3. Diseño .....</b>	<b>6</b>
3.1. Conexión de los sensores.....	8
3.1.1. Sensor de luminosidad.....	8
3.1.2. Sensor de humedad del suelo.....	8
3.2. Salida de nuestro sistema.....	10
3.3. Diseño de los Menús .....	11
<b>Capítulo 4. Selección de materiales .....</b>	<b>13</b>
<b>Capítulo 5. Resultado .....</b>	<b>32</b>
<b>Capítulo 6. Conclusiones y trabajo futuro.....</b>	<b>36</b>
<b>Capítulo 7. Manual de usuario .....</b>	<b>40</b>
<b>Apéndice A. Código del programa.....</b>	<b>42</b>
A.1. Este texto está escrito en estilo Título 7. Es el equivalente a Título 2 para los apéndices.....	¡Error! Marcador no definido.
A.1.1. Este texto está escrito en estilo Título 8. Es el equivalente a Título 3 para los apéndices.....	¡Error! Marcador no definido.
<b>Apéndice B. Funcionamiento de los sensores de humedad (1).....</b>	<b>52</b>
<b>Referencias.....</b>	<b>¡Error! Marcador no definido.</b>







# Capítulo 1. Introducción

En la actualidad existen diversos sistemas de programación de riego, no obstante, la mayoría de ellos se basan en un temporizador que se ha de programar para regar X veces al día durante un tiempo predeterminado.

Sin embargo, estos sistemas de riego tienen el inconveniente de que no todos los días hace la misma humedad y temperatura, por lo que la cantidad y periodicidad adecuadas para un día con una temperatura moderada no serán adecuadas para un día más cálido. Igualmente, estos sistemas de riego no cambiarán su programación si ese día llueve, desperdiciando así una cantidad valiosa de agua y siendo bastante ineficiente en el uso de recursos, lo que hace que al final se tenga que estar ajustando el sistema de riego según el clima o la estación del año.

Por lo tanto, estos sistemas de riego autónomos no lo son tanto y requieren del usuario cierto nivel de atención, podríamos llamarlos sistemas de riego semiautónomos.

Ejemplos de este tipo de sistemas tenemos muchos, desde los más económicos como este de NatRain Capítulo 6.Apéndice B.[8]:



Ilustración 1.1: Programador de riego NatRain

Cuyo precio se sitúa en torno a los 20€, que tiene la desventaja de que no se puede poner un comienzo diferido, es decir, si queremos que el sistema riegue cada 12 horas, a las 9 de la mañana y a las 9 de la noche, tendremos que realizar dicha programación a las 9 de la mañana o a las 9 de la noche y seleccionar el programa de repetición cada 12 horas. Además, no podemos seleccionar qué días ha de regar, sólo la periodicidad y el tiempo de riego.

Tenemos en el mercado algún modelo uno más sofisticado, con una pantalla lcd para una programación más cómoda en diferido, como este modelo de GEOLIA (Capítulo 6.Apéndice B.[9]):



**Ilustración 1.2: Programador de riego GEOLIA**

Por un precio de unos 30€ y con él conseguimos algunas funciones extra, como el poder seleccionar qué días ha de regar y cuáles no, podemos introducir 4 programas distintos de riego, por ejemplo, para que por las mañanas el tiempo de riego sea mayor que por las tardes, además podemos programar el riego en diferido y elegir hora de inicio sea la hora del día que sea en la que lo programamos.

Sin embargo, este modelo sigue teniendo algunas limitaciones, por ejemplo, no se le puede conectar un sensor de humedad para evitar que riegue cuando no sea necesario, para esto tenemos que pasar a un modelo como este de Gardena (Capítulo 6.Apéndice B.[10]):



Ilustración 1.3: Programador de riego Gardena

Cuyo precio son unos 40€ y admite un sensor de humedad, el problema es que dicho sensor vale otros 40€ (ya estamos en unos 80€ el conjunto) y además sólo podemos conectarle un sensor, por lo que, si queremos poner varios, necesitamos comprar varios kits completos de programador de riego y sensor, por lo que el coste se multiplica. Además, este modelo tiene otras limitaciones, como es el hecho de no poder conectarle un sensor de luz para asegurarnos de que no riega en horas de sol intenso (con el riesgo de poder quemar las plantas) y la única solución es ir cambiando la hora de riego según la estación.

Existen sistemas de riego más sofisticados, pero su precio sube exponencialmente, por eso el objetivo del presente proyecto es el desarrollo de un sistema de riego desatendido y económico, basado en sensores que determinen el nivel de humedad de la tierra, permitiendo conectar varios de ellos para poder controlar la humedad de toda la zona a regar y no sólo una pequeña parte de ella, intentando que ésta permanezca en unos límites adecuados para el cultivo y que además consideren el nivel de luminosidad para evitar regar en las horas donde el sol puede quemar las plantas. De esta forma se evita el tener que estar reprogramando el sistema de riego en función de las estaciones y de la climatología.

## Capítulo 2. Análisis del problema

Necesitamos pues un sistema de riego que cumpla una serie de requisitos:

- Sistema lo más barato posible
- Cuyos componentes sean fáciles de adquirir y que dispongamos de múltiples proveedores para cada componente.
- Sea lo suficientemente flexible para adaptarse a terrenos variados, por lo que necesitamos que sea fácilmente ampliable.
- Sea lo suficientemente inteligente para detectar cuando debe y puede regar y cuando no, con independencia del clima y las estaciones, requiriendo del usuario la mínima interacción posible para ello.
- Sea capaz de adaptarse a flora variada con las mínimas variaciones posibles por parte del usuario.

Necesitaremos por tanto algún tipo de sistema electrónico de control, ésta no tiene que ser muy potente pero deberá tener suficientes conexiones de entrada y salida, donde podamos conectar todos los sensores necesarios, tendremos que buscar una solución para minimizar al máximo el número de conexiones utilizadas por estos, ya que si queremos escalar nuestro sistema, el número de conexiones de nuestros sensores puede ser uno de los factores más limitantes en este sentido.

Además, nuestro sistema debe ser capaz de detectar la cantidad de luz solar existente para evitar regar en las horas en las que la radiación solar supere cierto umbral.

Otro problema al que nos enfrentamos es que nuestro sistema debe adaptarse a una variada flora, por lo que en una primera aproximación podemos pensar en configurar múltiples perfiles para según qué planta se pretenda regar, el problema es que una vez más estaríamos limitados a estos perfiles y hay ciertas variedades de plantas que son muy delicadas con la cantidad de agua que necesitan y que no deben regarse bajo ningún concepto en las horas de más luz

solar o se quemarían. Es por esto que nos vamos a decidir por una configuración donde el usuario pueda decidir directamente por la cantidad de humedad del suelo y no sobre la variedad de planta, esto quizás requiere del usuario un mejor conocimiento sobre la flora que posee, pero a cambio le ofrece una flexibilidad total.

También necesitamos una interfaz de entrada para nuestra placa, para que el usuario pueda modificar ciertos parámetros de la configuración, y una interfaz de salida, donde el usuario pueda ver la actual configuración de la placa, ver qué valor está cambiando e información útil como cuánto tiempo hace que se regó por última vez y durante cuánto tiempo.

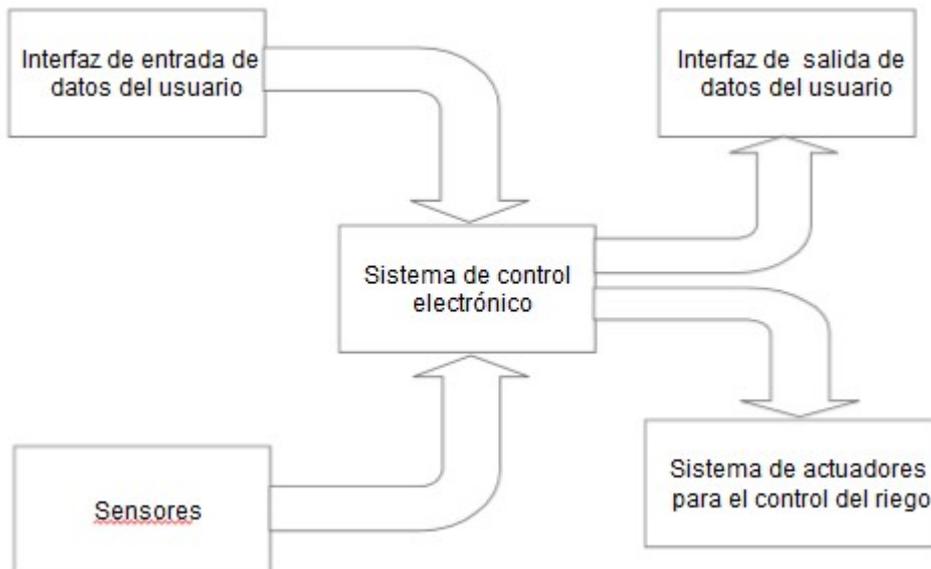


Ilustración 2.1: esquema de bloques de nuestro sistema

## Capítulo 3. Diseño

Por todo lo visto en el capítulo anterior, nos vamos a decantar por un sistema basado en Arduino acompañado de una serie de sensores (ampliados en un futuro según necesidad), ya que tanto las placas Arduino (o compatibles) como los accesorios para Arduino son baratos y fáciles de adquirir tanto localmente como a través de comercios online. Además, tenemos multitud de proveedores compatibles con ellos, librerías que nos facilitarán cualquier cosa que queramos modificar a nivel de software y una comunidad muy grande detrás de ellos.

El problema es que Arduino tiene un número limitado de entradas y salidas, pero debe poder adaptarse a un terreno de tamaño variable o incluso es posible que se quiera instalar en macetas, por lo que debemos buscar una forma de controlar la información proveniente de múltiples sensores desde un único puerto de nuestra placa para no estar limitados en este punto y que se pueda adaptar a todo tipo de configuraciones y terrenos.

Tenemos pues que diseñar un sistema de control de riego usando una placa Arduino, donde podamos conectar una cantidad variable de sensores a la cantidad mínima de puertos de nuestra placa Arduino. Como vamos a usar 2 tipos diferentes de sensores, la cantidad mínima para ello son 2 puertos, este pues será nuestro objetivo.

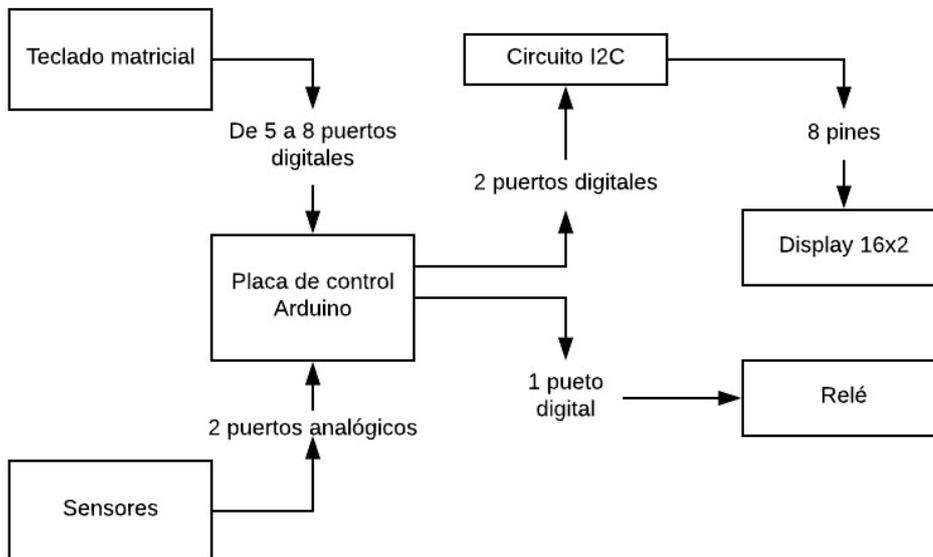
Además de los sensores, necesitamos otra serie de entradas y salidas para que el usuario pueda interactuar con nuestra placa, es decir, necesitamos una interfaz de salida y otra de entrada para nuestra placa.

La interfaz de salida, para poder leer la información proporcionada por nuestra placa, será un display, con el objetivo de mantener los costes lo más bajos posibles, usaremos un display de 16 caracteres por 2 filas, éste usa 8 puertos digitales de salida de nuestra placa Arduino, es demasiado, por lo que necesitamos una solución para intentar ahorrar el máximo número de puertos

vamos a usar un circuito [I2C](#), con lo que disminuimos el número de puertos de 8 a sólo 2, además el coste de dichos circuitos es muy bajo.

La interfaz de entrada será un teclado, vamos a intentar mantenerlo lo más simple posible, ya que tenemos que tener en cuenta que los teclados baratos normalmente son matriciales, por lo que vamos a necesitar un puerto digital por cada fila de nuestro teclado y otro por cada columna. En un primer momento parece que lo más sencillo es el uso de cursores, tanto para movernos por los menús como para incrementar o disminuir los valores, de este modo con quizás 6 teclas (4 de los cursores, 1 para aceptar y 1 para cancelar/retroceder) tendríamos suficiente, con lo cual sólo necesitaríamos 5 puertos de nuestra placa. Sin embargo, si incrementamos el número de puertos usados a 8, tendríamos un teclado matricial con caracteres numéricos de 0 a 9 y una serie de teclas auxiliares que simplificarán la configuración y la navegación por los menús.

El actuador será un relé que active una electroválvula o una bomba de agua, para esto sólo vamos a necesitar 1 puerto digital de salida.



**Ilustración 3.1: Esquema de conexiones**

Por lo tanto, de momento, tenemos 2 entradas analógicas para los sensores, 2 puertos de salida para el display y de 5 a 8 puertos de entrada digital para el teclado, esto hacen un total de 2 pines analógicos y de 8 a 11 digitales, por lo tanto, parece que cualquier placa Arduino nos podría valer en este sentido, incluidas las Arduino mini, micro o nano.

## 3.1. Conexión de los sensores

Necesitamos dos tipos de sensores distintos, los sensores de luminosidad y los de humedad del suelo.

### 3.1.1. Sensor de luminosidad

Los sensores de luminosidad presentan un problema menor, puesto por muy extenso que sea nuestro territorio a regar, va a ser siempre lo suficientemente pequeño como para que anochezca o amanezca a la vez en toda la extensión del terreno.

Se podría dar el caso de que cierta parte del terreno esté a la sombra mientras que otra parte esté al sol, por lo que podríamos necesitar colocar más de un sensor lumínico, pero realmente este es un problema menor que podemos solucionar eligiendo una buena ubicación de nuestro sensor, o bien conectar un segundo sensor usando un segundo puerto, puesto que ya hemos visto que realmente casi cualquier placa Arduino tendrá suficientes puertos como para que usar uno extra no sea un problema.

Otra posibilidad sería dividir nuestro sistema, de forma que con una sola placa Arduino controlemos dos sectores independientes de riego, de todas formas esto podría quedar como una posible ampliación para un trabajo futuro.

El problema más genérico y al que vamos a plantear solución, es aquel en el que con un solo sensor es suficiente, puesto que ya podemos detectar cuando amanece, anochece o cuando son las horas de mayor intensidad lumínica, por lo tanto, estos no presentan problemas de conexión, sólo vamos a usar uno e irá conectado directamente a nuestra placa.

### 3.1.2. Sensor de humedad del suelo

Aquí es donde realmente podemos tener un problema, puesto que en el caso de tener un terreno grande o querer adaptar nuestro sistema a un huerto urbano con macetas o algo similar, sería fácil que necesitemos diez, veinte o treinta sensores para otros tantos recipientes o terrenos que podrán tener una evaporación de agua distinta y por lo tanto necesitarán distintas frecuencias de

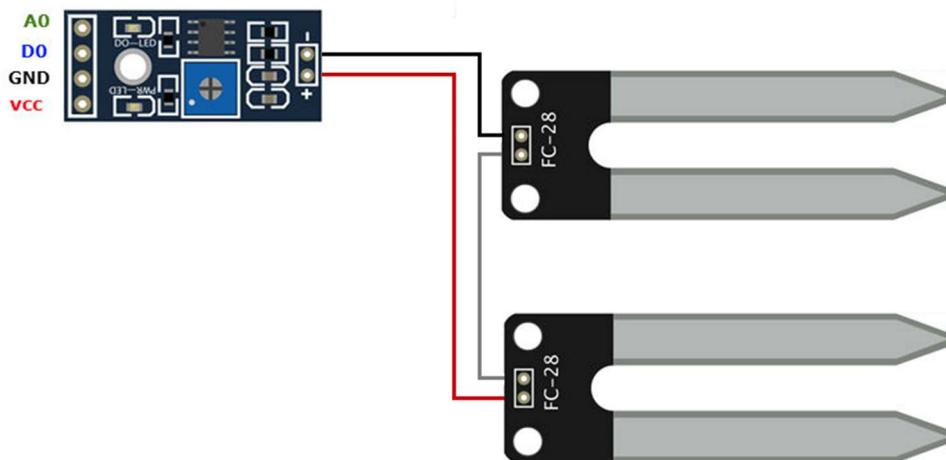
riego, ya sea porque algunos estén más expuestos al sol, a una fuente de calor o que sobre alguno exista alguna fuente de agua (una tubería o grifo que gotea, por ejemplo), por lo que necesitamos controlar la humedad individualmente.

En un principio podemos pensar en una circuitería que nos multiplexe la señal de cada sensor en el tiempo, de forma que en un solo puerto podamos añadir tantos sensores como queramos, pero la realidad es que estos sensores, realmente son dos metales conductores con una separación constante entre ellos, de forma que, al clavarse en el suelo, el agua que hay en éste, hace que la impedancia entre los dos conductores baje y bajará más cuanto más agua tenga el suelo.

Si queremos controlar cada zona individualmente, necesitaremos electroválvulas individuales o motores de riego individuales para poder regar cada zona individualmente, pero esto subiría mucho los costes de nuestro sistema y suponemos que en un terreno no tan grande, las diferencias de humedad de una zona a otra, aunque existan, no van a ser tan grandes, por lo que vamos a regar todo el sistema controlando cual es la zona más crítica, es decir, el terreno con menos humedad, de este modo, cuando alguno de los sensores marque menor humedad de un umbral fijado, vamos a poner en marcha el riego de todo el sistema.

Veamos primero como varia la conductividad del suelo en función de la humedad de éste, como podemos ver en el apéndice: Funcionamiento de los sensores de humedad, parece que la impedancia es lineal respecto a la humedad del suelo, sin embargo este modelo es para un sensor donde la distancia entre los electrodos es igual a la separación entre ellos, lo cual no se cumple en los sensores económicos para Arduino que tenemos en el mercado, además depende del material, sección y forma del electrodo usado, tendremos entonces que hacer algunas pruebas para determinar cómo varía, pero como veremos posteriormente en el capítulo de Selección de materiales, los sensores suelen usar una frecuencia de unos pocos kHz y suelen venir con un circuito integrado que proporciona directamente a nuestra placa un rango de valores, en nuestro caso (adelantándonos un poco) nuestro sensor tiene 1024 niveles y añadir sensores extra a esta placa apenas varía el valor de salida, mientras que modificar ligeramente el valor de humedad si lo hace, por lo tanto podemos

conectar en serie varios sensores a esta placa intermedia sin miedo a modificar el comportamiento de nuestro sistema. Quedaría como en la figura 3.1.



**Ilustración 3.2: Sensores conectados en serie**

Como vemos, tenemos dos salidas, D0 (salida digital) y A0 (salida analógica), la salida digital se activará únicamente cuando la humedad pase de cierto umbral, dicho umbral se regula con el potenciómetro. Parece una primera aproximación bastante sencilla, el problema es que para un usuario sin conocimientos técnicos parece bastante más complicado calibrar el potenciómetro que elegir un valor en un menú. Además, si se quiere sectorizar parte del riego teniendo un par de zonas independientes, habría que regular a mano todas las zonas y habría que hacerlo de forma experimental, añadiendo agua a la zona que rodea al sensor hasta llegar al umbral que se quiere para poder ajustar el potenciómetro.

Parece bastante más conveniente usar la salida analógica que nos proporciona 4096 niveles y por software hacer la elección del umbral.

### 3.2. Salida de nuestro sistema

La salida de nuestro sistema va a ser bastante simple, cuando estemos regando encenderemos un relé que activen las bombas de riego o bien podemos conectar unas electroválvulas si la presión del agua es suficiente para regar.

En nuestro sistema de pruebas vamos a usar un simple led que se encienda cuando esté regando o se apague cuando deje de regar, pero para instalar nuestro sistema en un entorno real nos bastaría con cambiar el led por la electroválvula o el relé.

### 3.3. Diseño de los Menús

El sistema de menú va a venir limitado por el display elegido, que, como veremos en el capítulo Selección de materiales, es de 2 líneas de 16 caracteres, por lo que no podremos presentar mucha información en cada pantalla, pero tenemos que dar la opción al usuario de configurar la sensibilidad de ambos tipos de sensores y mostrarle alguna información respecto al riego.

En el teclado tenemos cuatro caracteres alfabéticos (A, B, C y D), podríamos usarlos como cursores o bien usarlos como accesos directos a los apartados del menú, como sólo tenemos 2 parámetros que configurar, los usaremos como accesos directos a los distintos apartados del menú.

Así pues, podemos asignar la letra A para configurar la humedad y la letra B para configurar la luz, pero como también queremos mostrar en el display cierta información, como si el riego está activo o no, configuraremos la letra C para acceder al menú de configuración.

Además, es interesante para el usuario, saber cuánto tiempo ha estado regando el sistema cuando deja de regar, por lo que al parar de regar mostraremos durante un breve periodo de tiempo el tiempo que ha estado activo el riego.

Cuando empecemos a regar otra vez, mostraremos en el display cuanto tiempo hace desde la última vez que se regó.

#### 3.3.1. Menú para la configuración del sensor de humedad

Dejar a un usuario elegir entre los 1024 valores no parece lo más adecuado, así que vamos a reducirlo. 10 valores parecen suficientes para seleccionar la humedad de la tierra, le asignaremos a cada uno un rango de esos 1024 valores que nos proporciona el sensor, de forma que el valor 0 signifique que queremos la tierra más seca y 9 que la queremos lo más húmeda posible. Vamos a eliminar cierto rango de los extremos, ya que un valor 0 del sensor es difícilmente

alcanzable, incluso son los sensores sumergidos en agua dulce, los valores oscilan en torno a 320, por lo que tomaremos 280 como valor más bajo (significaría que el riego debe permanecer siempre activo, bajo cualquier condición de humedad) y a partir de ahí iremos subiendo a intervalos de 80 puntos en el valor del sensor hasta llegar a 1000 (valor para el cual la tierra debe estar casi totalmente seca). Es decir, valor 0 es siempre activo, valor 1 es tierra casi totalmente inundada y vamos subiendo en aridez hasta valor 9 que significa tierra casi totalmente seca.

### 3.3.2. Menú para la configuración del sensor de luminosidad

Los sensores de luminosidad tienen un rango de 1024 escalones, siendo 1024 a oscuras y 0 la máxima luminosidad. Sin embargo es muy difícil que superemos un valor de 600 y con una exposición directa a la luz solar, rondamos los valores 12 o 13 del sensor, por lo que estableceremos el mínimo en 6 y crearemos 10 niveles, igual que en el sensor de humedad, espaciando cada nivel en 66 escalones del sensor, de esta forma 9 significa regar siempre, tengamos la luz que tengamos (igual que con el sensor de humedad) y a partir de ahí vamos estableciendo un umbral de luz cada vez más bajo, hasta llegar al valor 0 que correspondería al valor de oscuridad absoluta (no regar nunca).

## Capítulo 4. Selección de materiales

### 4.1. Placa Arduino

Empezamos con el corazón de nuestro sistema, la placa Arduino. Vistas las necesidades de conectividad de nuestro sistema, sólo necesitamos 2 entradas analógicas (para los sensores) y 8 digitales (para el teclado) y 2 salidas digitales (una para el display y otra para el relé o electroválvula). Por lo tanto, en este sentido nos valdría incluso el Arduino el nano o el micro.

Los otros aspectos que pueden ser limitantes son la capacidad de procesamiento y la memoria. En cuanto a capacidad, nuestro sistema va a ser realmente poco exigente, por lo que cualquier placa cumplirá los requerimientos sin problema.

En cuanto a la memoria, como veremos más adelante, nuestro programa va a ser muy simple y usará menos de 10kB de memoria flash, por lo que tampoco

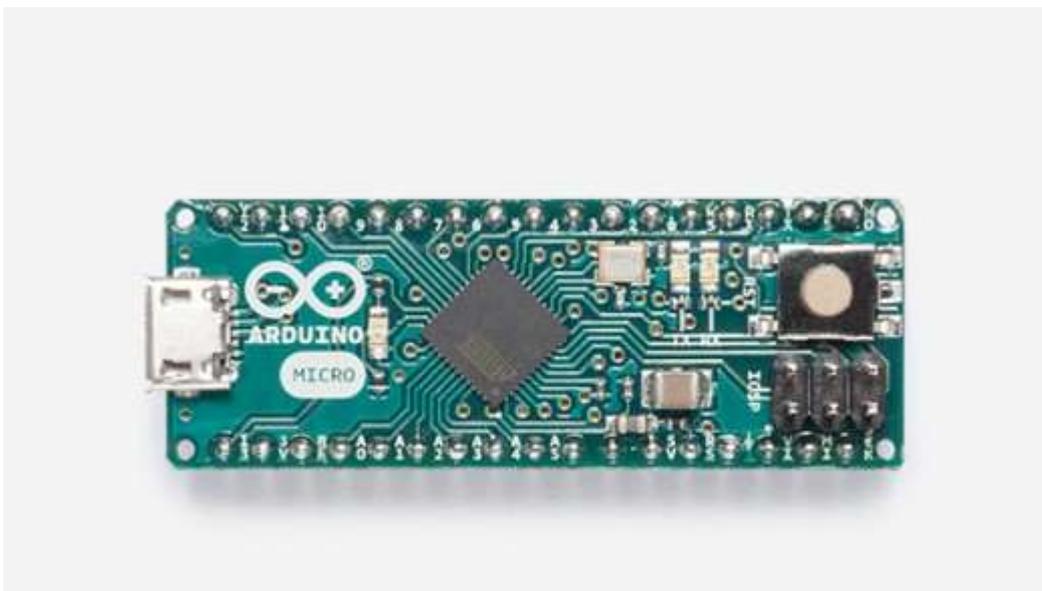
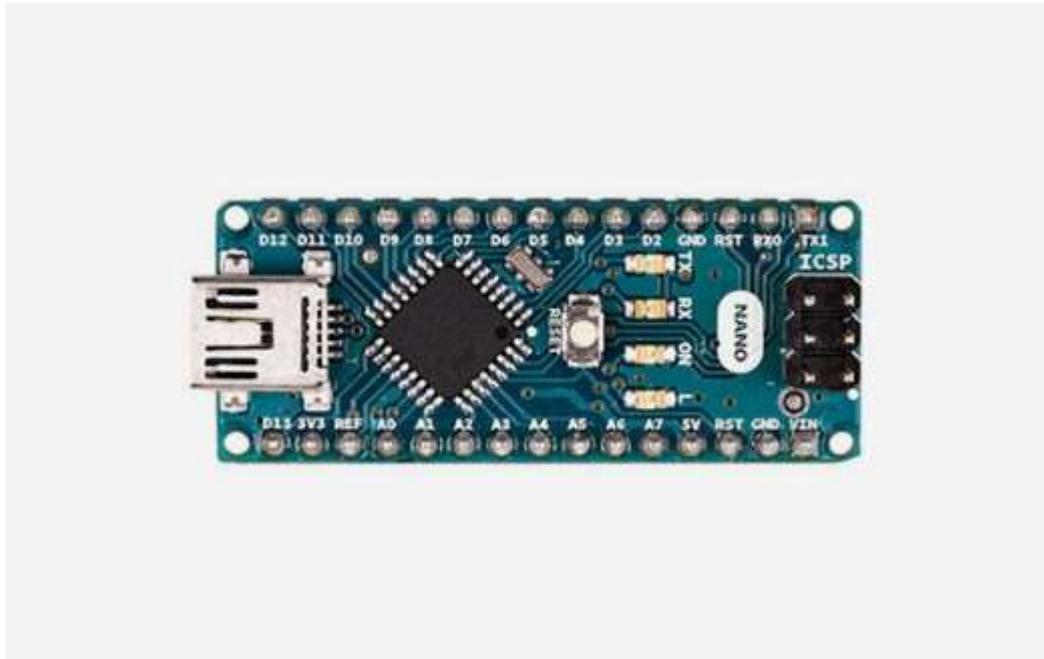


Ilustración 4.1: Arduino Micro

supone una limitación, por lo tanto, podemos elegir la placa más barata, en este caso sería el Arduino Micro, con un precio oficial de 18 dólares:

Aunque realmente, el más barato que podemos encontrar es el Arduino Nano:



**Ilustración 4.2: Arduino nano**

Que podemos encontrarlo en tiendas online, como ebay, por menos de 2,5€. Y es equivalente a nivel de prestaciones, al Arduino Uno.

El problema de estas dos placas es que están más pensadas para una fase de producción, ya que son muy baratas y compactas, ideales para usar en un entorno real, pero que para nuestro entorno de desarrollo no son tan ideales, puesto que no tienen el pineado para poder conectarle jumpers directamente.

Por esto y porque ya disponíamos de una placa compatible con Arduino Leonardo, que es una placa un poco más completa, pero muy parecida y cuyo precio es de unos 9\$, usaremos ésta en nuestro entorno de desarrollo, pero nuestra aplicación será fácilmente transportable a una Arduino Nano, que sería el modelo elegido en caso de querer llevar el proyecto a producción.



Ilustración 4.3: Arduino Leonardo

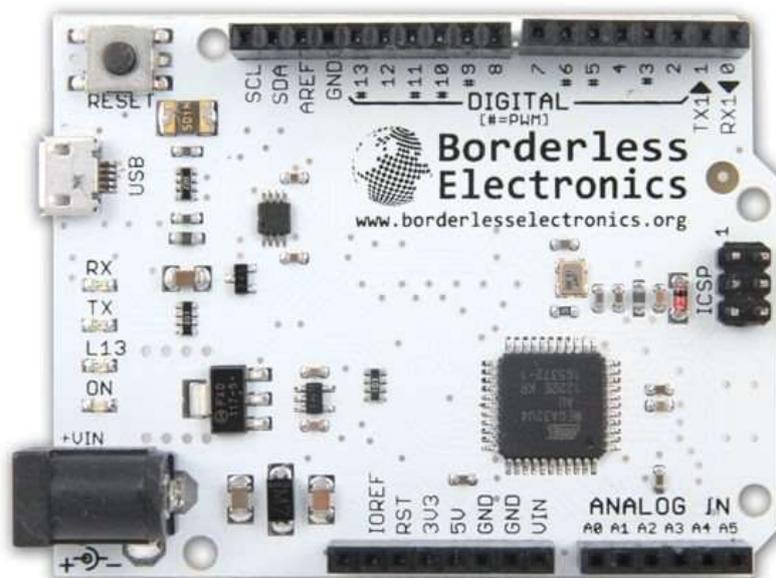


Ilustración 4.4: Borderless Electronics – Placa Arduino Leonardo compatible

## 4.2. Display

Cuando buscamos un display para Arduino, tenemos muchas opciones, incluidas pantallas táctiles, oled, etc., pero queremos algo barato, sin muchas pretensiones y con un módulo I2C para reducir el número de puertos, por lo que las principales alternativas son, en primer lugar, pantallas tft en torno a las 2 pulgadas de diagonal, cuyo precio ronda los 15€ en eBay (Capítulo 6.Apéndice B.[11]):



Ilustración 4.5: Pantalla tft 2", fuente: vendedor de eBay

Pantallas de 4 líneas y 20 caracteres, con módulo I2C incluido, cuyo precio ronda los 7 u 8€ (Capítulo 6.Apéndice B.[12])



Ilustración 4.6: Icd 20x4 con módulo I2C

Y la más barata de todas sería un display de 16 caracteres y 2 líneas, con módulo I2C incluido por menos de 3€ en ebay, a nosotros nos ha costado 2,45€ (Capítulo 6.Apéndice B.[13]).

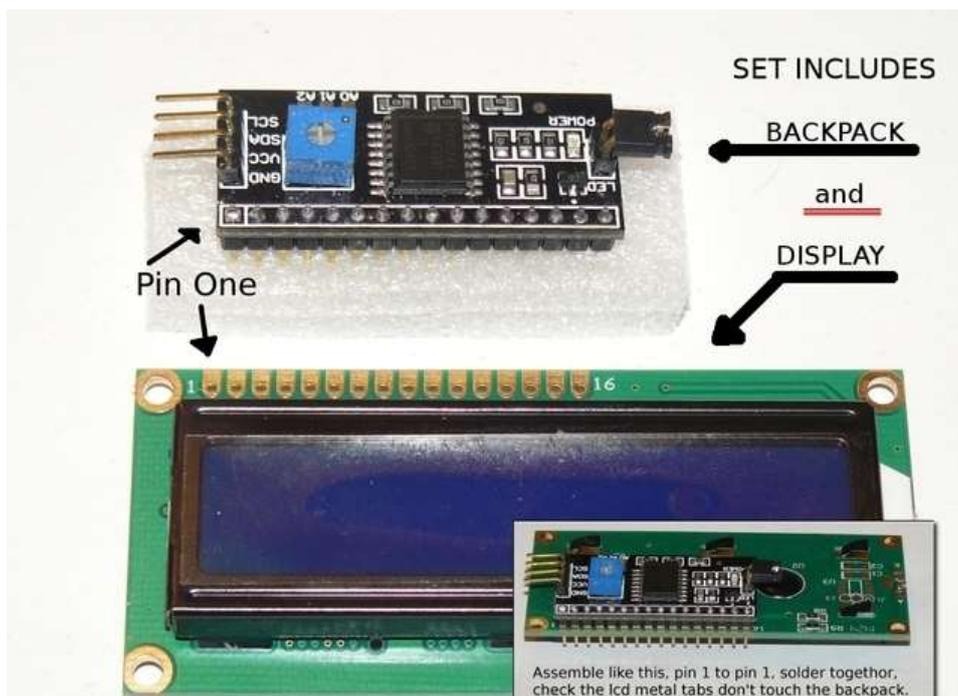
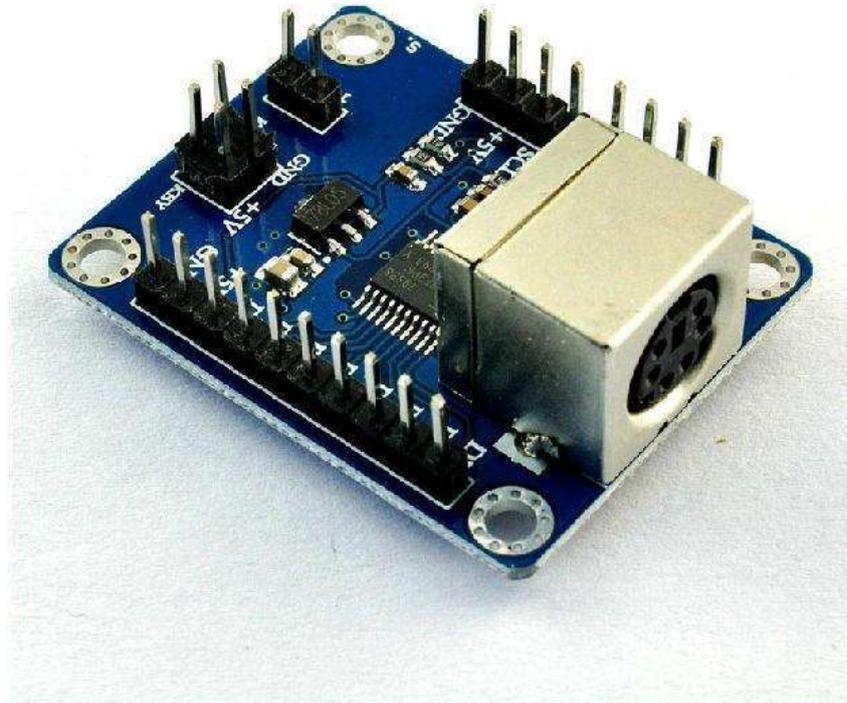


Ilustración 4.7: Icd 16x2 con módulo I2C





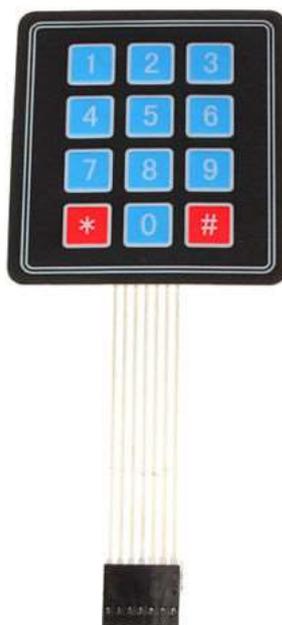
**Ilustración 4.10: Modulo adaptador a PS/2 con opción de conectar a puertos digitales o usar protocolo I2C**



**Ilustración 4.9: Modulo adaptador a PS/2 simplificado para usar sólo protocolo I2C**

Otras opciones más baratas y simples son los teclados matriciales, tenemos muchas opciones, pero las más populares son los teclados 4x4 y 4x3, ambos los podemos encontrar por 1€, por lo que, aunque el 4x4 use 8 puertos digitales y el

4x3 use sólo 7, nos vamos a decantar por el teclado 4x4, puesto que siempre podemos anular una de las columnas en caso de que no nos hiciese falta y ahorrarnos la conexión de ese puerto adicional, aunque como ya hemos visto antes, no tenemos problemas de escasez de puertos en nuestra placa y esa columna adicional nos supone bastante comodidad a la hora de usar los menús. A nosotros nos ha costado 1€ (Capítulo 6.Apéndice B.[16]).



**Ilustración 4.11: Teclado matricial 4x3**



**Ilustración 4.12: teclado matricial 4x4**

Podemos ver el funcionamiento de los pines de este teclado en la Ilustración 4.13: Funcionamiento teclado matricial:

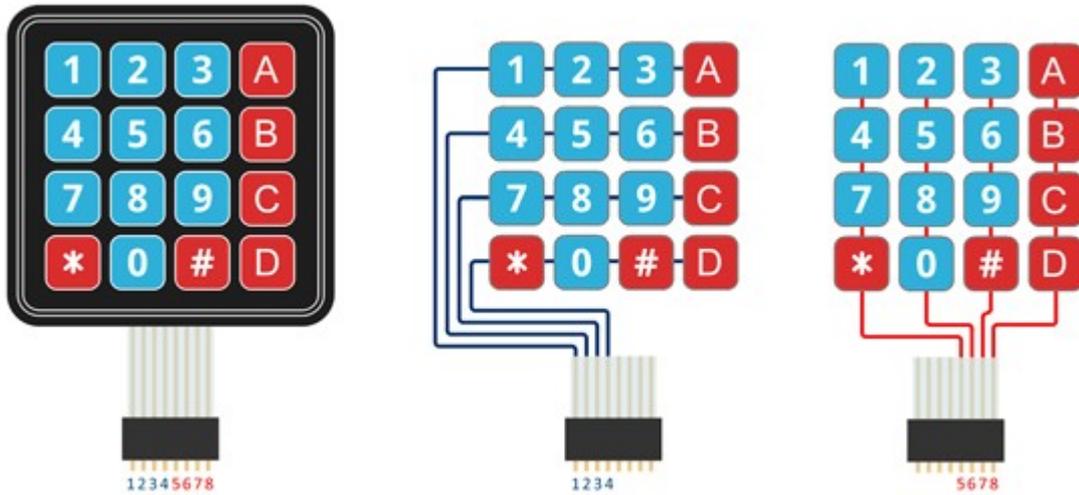


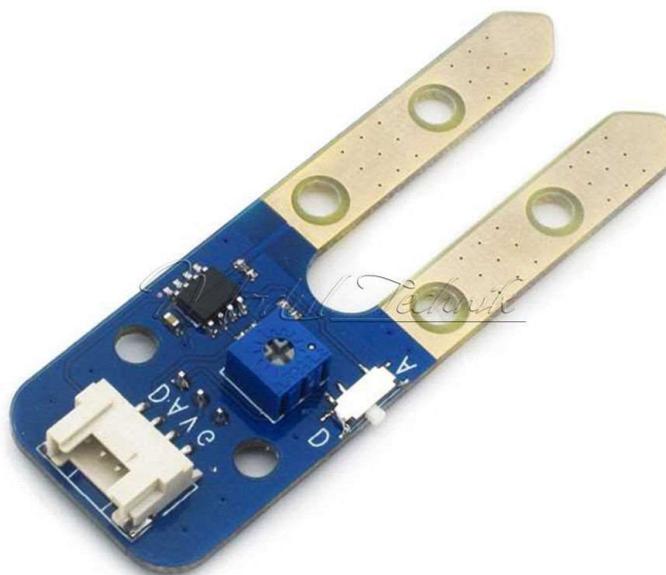
Ilustración 4.13: Funcionamiento teclado matricial

#### 4.4. Sensor de humedad

Este punto es bastante importante en cuanto a costes, ya que potencialmente podemos necesitar muchos de estos sensores en una implementación real en caso de tener que regar un terreno grande o con muchos recipientes distintos, es por eso que, aunque estos sensores son baratos, una diferencia de precio no tan grande podría ser determinante.

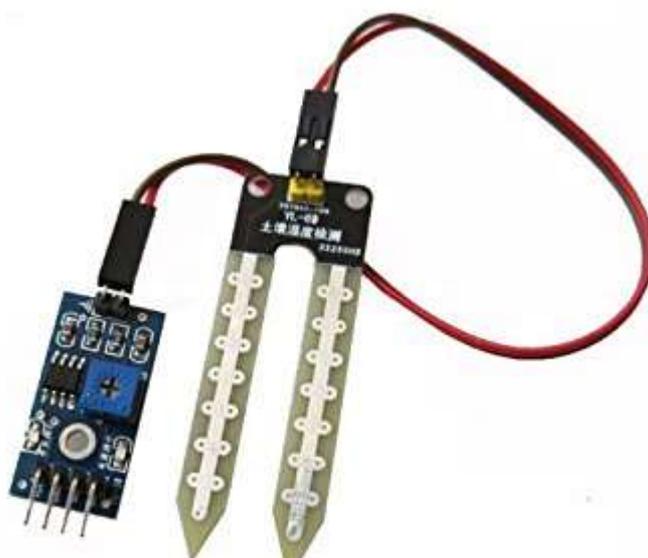
Tenemos bastante variedad en el mercado, aunque la estructura básica de todos es la misma, dos conductores separados una cierta cantidad, con una circuitería que mediante pulsos eléctricos miden la conductividad entre los dos conductores.

La diferencia principal entre los modelos que hay en el mercado, además del coste, es que algunos llevan dicha circuitería integrada en el propio sensor, como el modelo de la Ilustración 4.14: higrómetro con circuitería integrada en el propio sensor.



**Ilustración 4.14: higrómetro con circuitería integrada en el propio sensor**

O bien tenemos la circuitería en una placa independiente del propio sensor, como en la Ilustración 4.15: Higrómetro con circuitería en placa independiente.



**Ilustración 4.15: Higrómetro con circuitería en placa independiente**

El precio de ambas es parecido y ronda el euro.

Sin embargo, en la unidad que lleva la placa independiente es más fácil proteger la circuitería de la humedad, ya que ésta no tiene que estar a ras de suelo, por lo que la durabilidad será mejor.

Además, como ya vimos, nos interesa poder conectar varios sensores a una única placa, de forma que sólo ocupemos un puerto de nuestro Arduino, esto sólo lo podemos hacer si elegimos el sensor con la placa independiente. De esta forma también podemos abaratar costes, puesto que sólo necesitamos una placa de control para múltiples sensores, de forma que podemos reducir el coste por sensor sensiblemente, por ejemplo, si necesitamos 20 sensores, comprando sólo una placa y 20 sensores, reduciríamos a menos de la mitad el coste total de los sensores.

Estas son las dos principales razones por las que elegiremos la segunda opción, el sensor con la placa independiente. A nosotros nos ha costado 0,78€ cada sensor.

Este modelo de sensor es el FC-28 y tal como vemos en la Ilustración 4.16: Placa sensor humedad

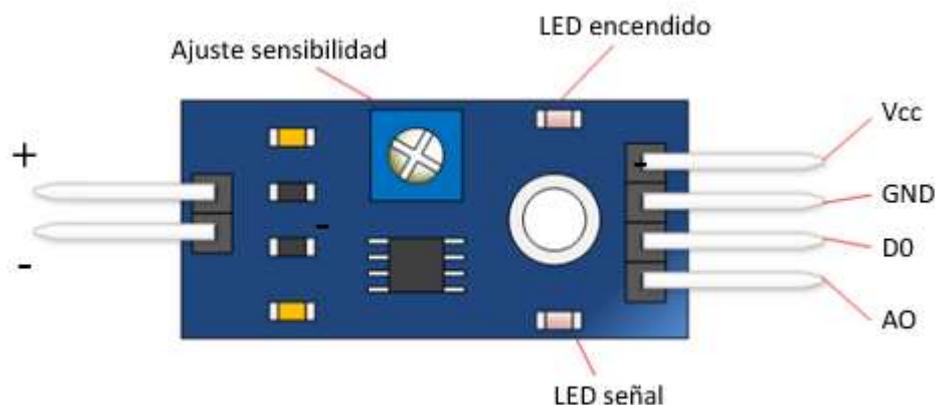


Ilustración 4.16: Placa sensor humedad

Tenemos un led que nos indica que la placa recibe alimentación y está funcionando, las patas marcadas con “+” y “-” son donde conectaremos los sensores en serie, la pata marcada con “Vcc” y “GND” serán las que conectemos a las patas de “5V” y “GND” de nuestro Arduino y luego tenemos dos salidas, una digital (D0) y otra analógica (A0).

Si usamos la digital sólo tendremos dos posibles valores, encendido o apagado, según la humedad del suelo sobrepase o no cierto límite. Dicho límite se regula con el potenciómetro y el led de señal nos indica si se está sobrepasando o no.

Como ya vimos, lo que nos interesa es la salida analógica, para poder configurar el umbral de humedad por software, veamos ahora que valores obtenemos con esta salida con algunas pruebas preliminares.

#### 4.4.1. Valores de la salida analógica

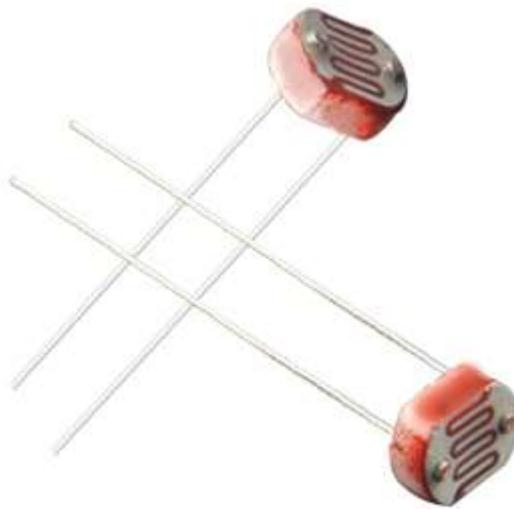
Con la salida analógica tendremos 1024 niveles distintos, según la humedad medida por el sensor, siendo un valor más bajo más humedad y un valor más alto menos humedad. Sumergir el sensor completamente en agua nos da un valor entre 280 y 290. Dejar el sensor al aire, nos da un valor de 1022 o 1023. Si conectamos 2 sensores en serie y sumergimos uno completamente y el otro lo dejamos al aire seguimos obteniendo el mismo valor. Si sumergimos ambos sensores hasta la mitad en agua, obtenemos un valor de 330-340, si mantenemos un sensor en esa posición y sumergimos el otro hasta cubrirlo completamente, el valor baja hasta 320, por lo que el cambio no es significativo. Si introducimos sólo un par de milímetros de cada sensor en agua, el valor obtenido es de alrededor de 550-560, si mantenemos un sensor en esa posición y el otro lo introducimos completamente en agua, el valor obtenido es de 520-530. Como vemos, en ningún caso es un cambio muy significativo y siempre predominará claramente el sensor con resistencia más alta, es decir, el que mide menos humedad, de forma que siempre se va a dar prioridad al terreno más necesitado de agua para regar.

<b>Nivel inmersión sensor 1 (%)</b>	<b>Nivel inmersión sensor 2 (%)</b>	<b>Valor en la placa</b>
0	0	1022-1023
5	5	550-560
5	100	530
50	50	330-340

50	100	320
100	100	280-290

## 4.5. Sensor de luminosidad

Aquí principalmente tenemos dos opciones, la primera y más barata es comprar el fotodiodo y fabricarnos nuestro propio fotodetector. En ebay podemos encontrar 20 sensores LDR por 1€



**Ilustración 4.17: Sensor LDR**

Para construirnos nuestro sensor, básicamente tenemos que conectarlo en serie con una resistencia y conectar el puerto analógico de nuestra placa entre el sensor y la resistencia, tal como vemos en el siguiente diagrama:

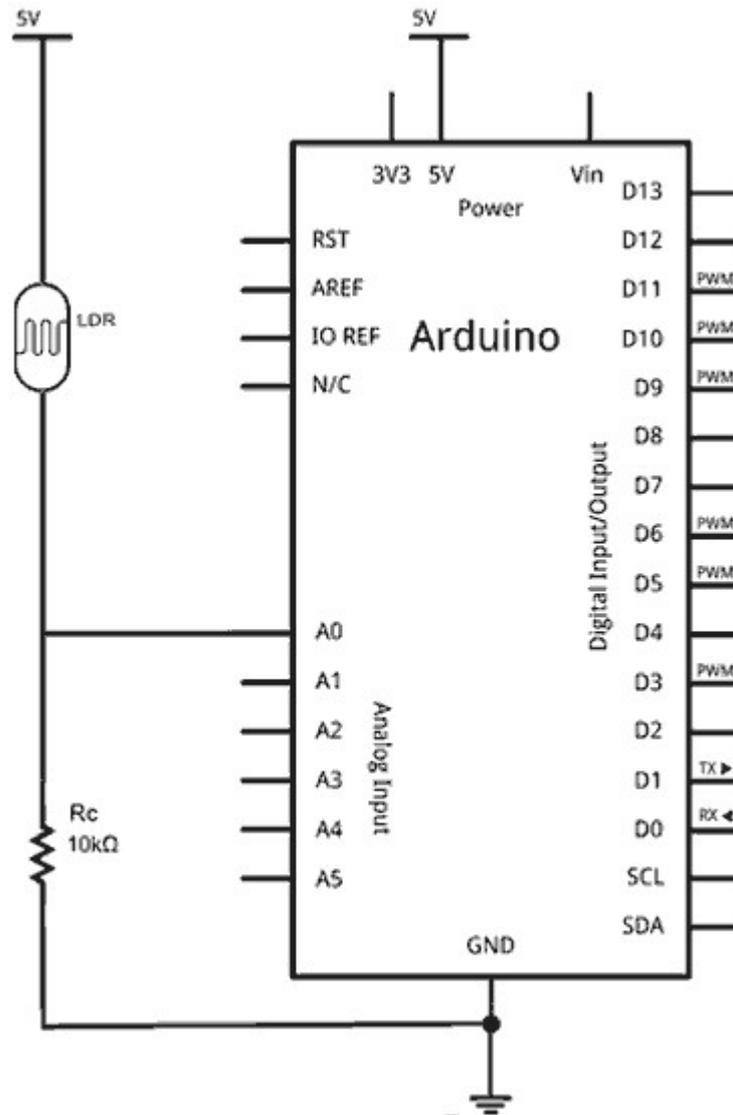
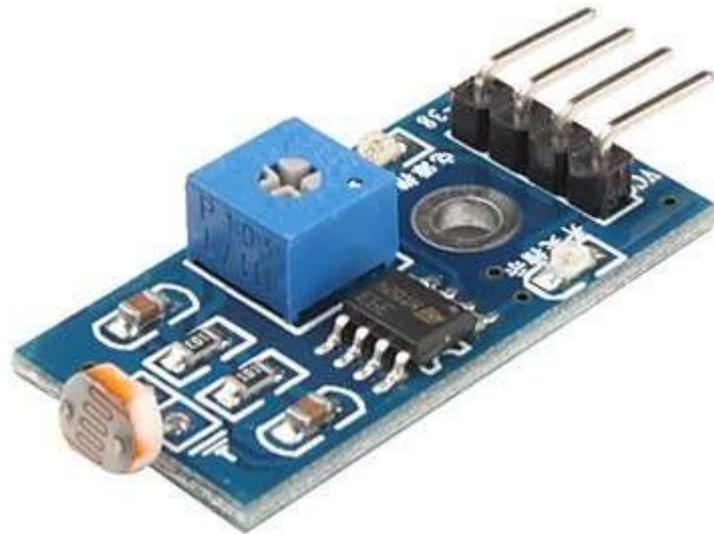


Ilustración 4.18: esquema montaje sensor LDR

Ésta es sin duda la opción más barata. La otra opción sería comprar el circuito ya montado, como el que vemos en la Ilustración 4.19: Sensor de luz fotoresistivo.



**Ilustración 4.19: Sensor de luz fotoresistivo**

Donde, al igual que en el sensor de humedad tenemos la posibilidad de usar una salida analógica o una digital. En principio, igual que con el sensor de humedad, nos interesa más la salida analógica, por lo que tanto esta como la primera opción podrían servirnos. Este circuito ronda el euro en tiendas online y aunque es ligeramente más caro que la opción anterior, apenas supone un sobrecoste siendo un poco más versátil y simplificando el montaje, por lo que nos vamos a decantar por esta opción. A nosotros esta placa nos ha costado 0,85€ en eBay. Veamos más detalladamente esta placa, aunque tiene las mismas conexiones que el sensor de humedad elegido.

Tenemos los pines marcados con “Vcc” y “GND” que irán conectados a 5v y a tierra respectivamente, un led para indicar que el sensor está funcionando, una salida digital (D0), que nos dará sólo dos valores posibles en función de si la luz supera o no cierto umbral y dicho umbral se podrá regular con el potenciómetro, un led de señal marcado como D0-LED que nos indicará si la luz detectada ha superado el umbral configurado o no y una salida analógica con un rango de valores.

### 4.5.1. Valores de la salida analógica

Con la salida analógica tendremos un rango de 1024 valores posibles, pero hay que tener en cuenta que, en una oscuridad absoluta, no vamos a llegar nunca a un valor de 850 y en condiciones reales, sólo con la luz de las estrellas, no vamos a llegar a un valor de 600.

En condiciones de exposición directa a la luz solar rondaremos un valor de 12 o 13, en un día soleado, a la sombra rondaremos valores de 300, en una habitación bien iluminada estaremos en valores que rondan los 100-120.

## 4.6. Dispositivo de salida

Aquí podríamos tener una variedad de dispositivos según el entorno en el que se vaya a utilizar, por ejemplo, si la fuente de agua es un estanque, depósito de agua o similar, necesitaremos un relé que active una bomba de agua. Si la fuente de agua tiene presión, por ejemplo, una conexión al suministro de agua de la ciudad, necesitaremos una electroválvula o incluso un servo que mueva una válvula manual. Lo más normal sería tener una electroválvula.

Normalmente, éstas suelen ser de 24v, por lo que nos tendríamos que buscar una fuente externa para la alimentación de dicha electroválvula (o un circuito boost para elevar nuestros 5v de alimentación de Arduino a esos 24v) y un relé DC que soporte dichos 24v. También podemos buscar una electroválvula AC de 220v como la de la Ilustración 4.20: Electrovalvula 220v, cuyo precio ronda los 9€.



Ilustración 4.20: Electrovalvula 220v

Para controlarla necesitaremos un relé AC cuyo precio ronda 1€. A nosotros nos ha costado 0,85€.

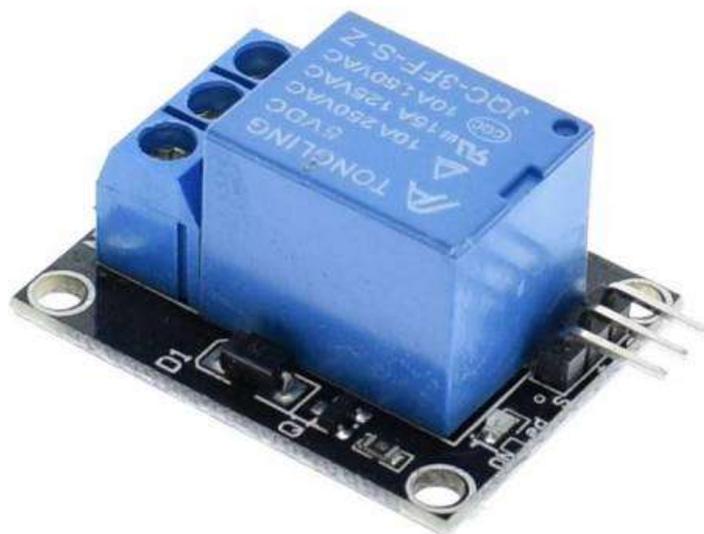


Ilustración 4.21: Relé AC para 125-250v controlado por 5v

Las conexiones con nuestra placa son muy simples, tienen 2 pines de alimentación (Vcc y GND) y un pin de entrada que conectaremos con la salida de nuestra placa.

De todas formas, nosotros, para nuestro entorno de desarrollo, simularemos la salida con un simple led que irá conectado al mismo pin de salida al que iría el relé, para montarlo en un entorno real sólo habría que sustituir el led por el relé.

## 4.7. Integración del sistema

Al final tenemos algo como lo que vemos en el siguiente esquema:

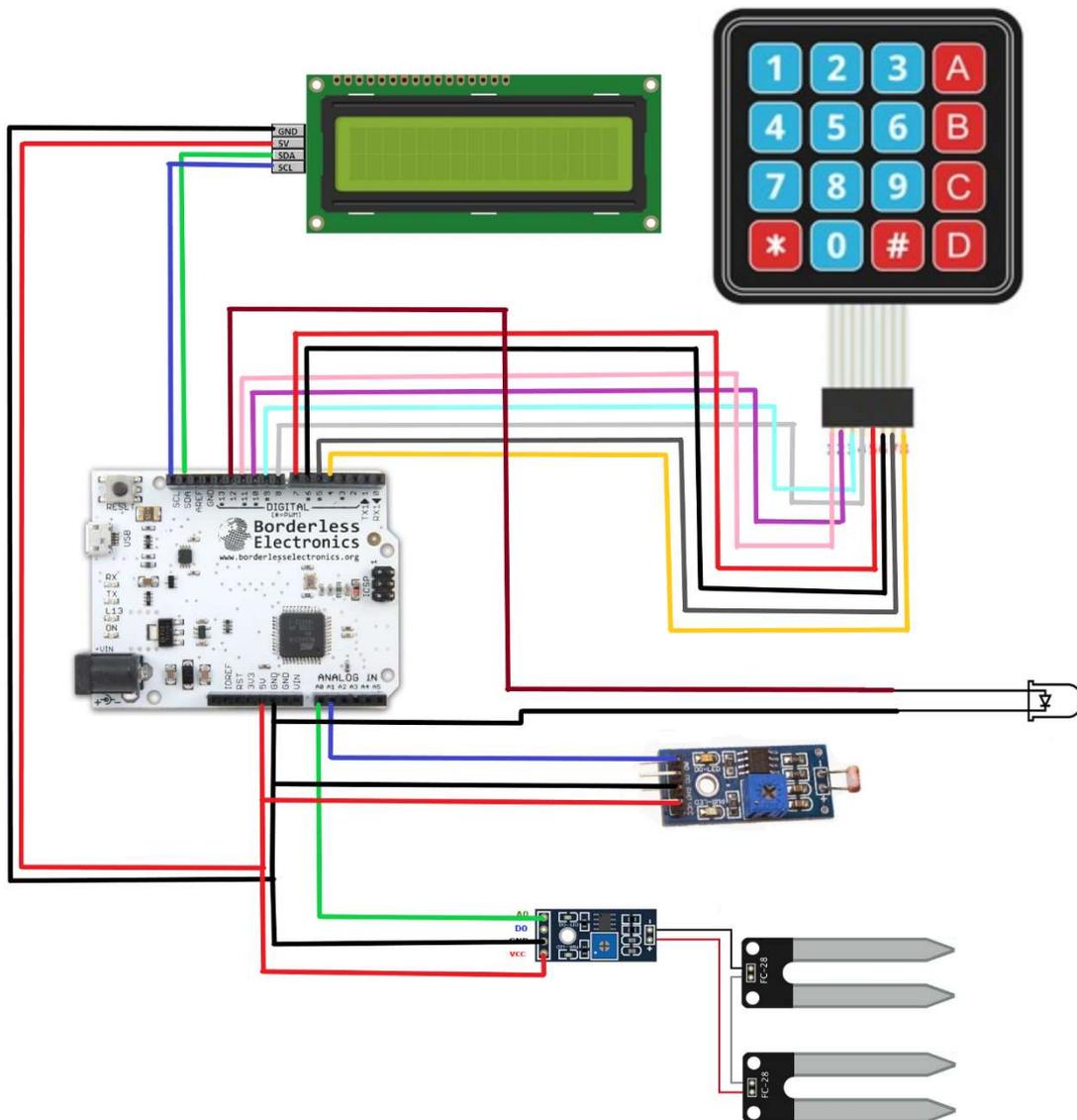


Ilustración 4.22: Esquema de conexiones

Como vemos, el display va conectado a los puertos “SCL” y “SDA” de nuestra placa, el teclado matriz 4x4 va conectado a los pines 4 a 10 de nuestra placa, el sensor de humedad al pin A0 y el sensor de luminosidad al pin A1. Hemos usado el pin digital 12 como salida de nuestro sistema, para el prototipo hemos usado un led, pero sólo tendríamos que conectar aquí un relé para usarlo en un sistema real.

Podemos ver una foto del prototipo en la Ilustración 4.23: Foto del prototipo:

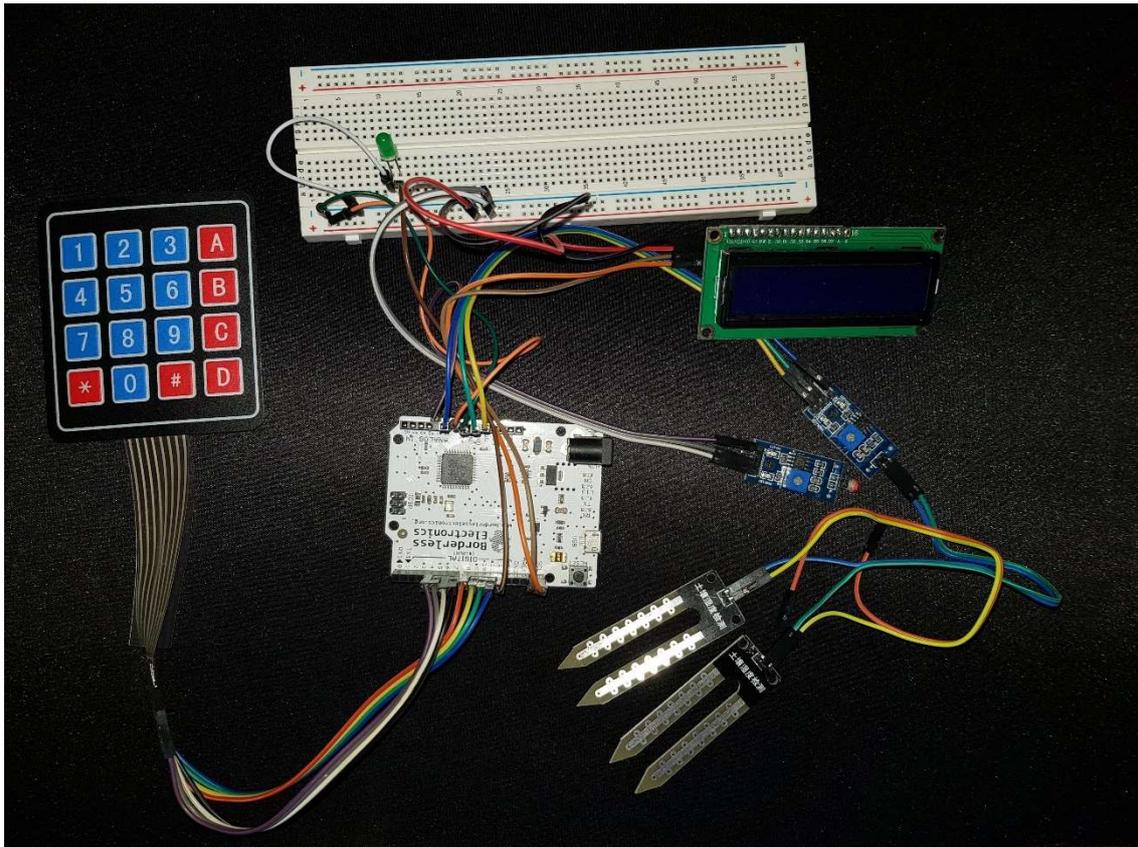


Ilustración 4.23: Foto del prototipo

## Capítulo 5. Pruebas de funcionamiento

Una vez el sistema está montado, lo ponemos en funcionamiento con los sensores al aire y vemos que nada más encenderse, el riego se activa, pues se dan las condiciones lumínicas y de humedad puestas por defecto (100 para el límite de luminosidad y 500 para el de humedad), en cuanto ponemos el sensor de luminosidad al sol o lo iluminamos con una linterna vemos el tiempo que ha estado regando y el riego se para, cuando apagamos la linterna, el riego se vuelve a activar y vemos cuanto tiempo hace que dejó de regar por última vez.

Lo mismo sucede con las condiciones de humedad, en cuando sumergimos los sensores en agua lo suficiente, el riego se para y en cuanto los volvemos a sacar el riego se activa.

Si nuestro sistema llega al límite de humedad y para el riego, en cuanto sacamos un poco uno de los sensores de humedad del agua, el riego se activa nuevamente, por lo que cumplimos la premisa hecha inicialmente, que el límite de riego lo marcará el sensor que marque una humedad menor, de forma que prioricemos regar una maceta o terreno seco a pesar de que ello signifique sobre hidratar otra maceta o terreno.

Probamos ahora a cambiar los límites sin cambiar las condiciones de los sensores cuando el sistema está estable. Empecemos con el sensor lumínico. Con el sistema regando, probamos a bajar el límite de luz aceptada, el riego se para inmediatamente. Si ahora volvemos a subir el límite, el riego comienza nuevamente.

Lo mismo sucede al repetir dicha prueba con el límite de humedad.

El tiempo de reacción, aunque no es un factor crítico, es bastante bueno, mucho menor de un segundo, por lo que esto no va a suponer ningún problema en nuestro sistema.

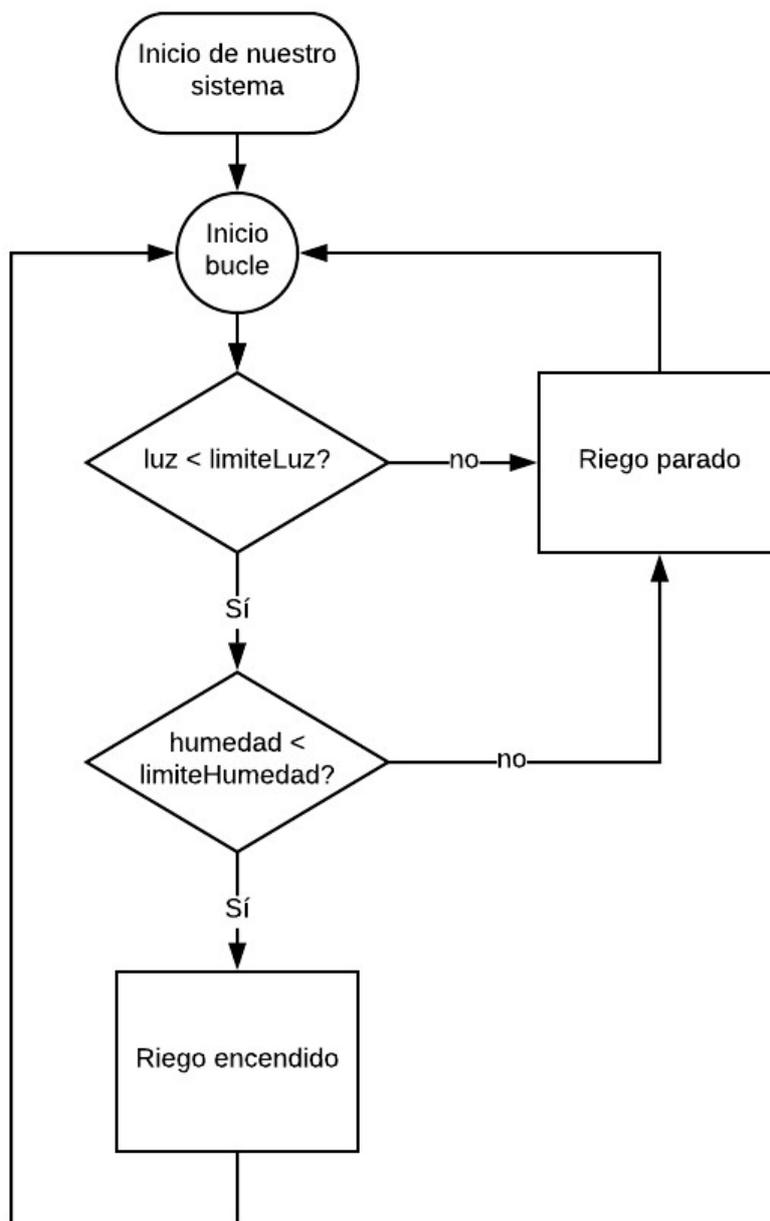
Podríamos pensar que nuestro sistema necesitaría una histéresis para evitar que esté constantemente regando, parando y volviendo a regar, sin embargo, si el sensor no lo colocamos justo al lado del caudal de riego, esto no supone mayor problema.

Hemos hecho algunas pruebas y como el agua tarda un tiempo en filtrarse por el terreno, esto provee cierto factor de histéresis a nuestro sistema, por lo que no es necesario que nos preocupemos de eso. Por ejemplo, hemos probado a colocar el sensor a unos 20 centímetros del caudal de riego en una maceta, con un caudal no demasiado grande y hemos comprobado que cuando se llega al límite de humedad marcado, el riego se para y la humedad registrada por el sensor sigue aumentando durante unos 15 a 40 segundos antes de estabilizarse, por lo que pasa bastante tiempo antes de volver a pasar el límite de humedad establecido.

## 5.1. Comportamiento de nuestro sistema

Para entender mejor cómo funciona nuestro sistema y cómo es la toma de decisiones, veámoslo en un diagrama de flujo.

La lógica básica de nuestro sistema, sin que intervengan los menús o ninguna interacción con el usuario sería la siguiente:



**Ilustración 5.1: Diagrama de flujo de toma de decisiones**

Si introducimos el cambio de valores de nuestro sistema por parte del usuario y los mensajes que se muestran por pantalla cuando se activa o para el riego, tendríamos algo como esto:

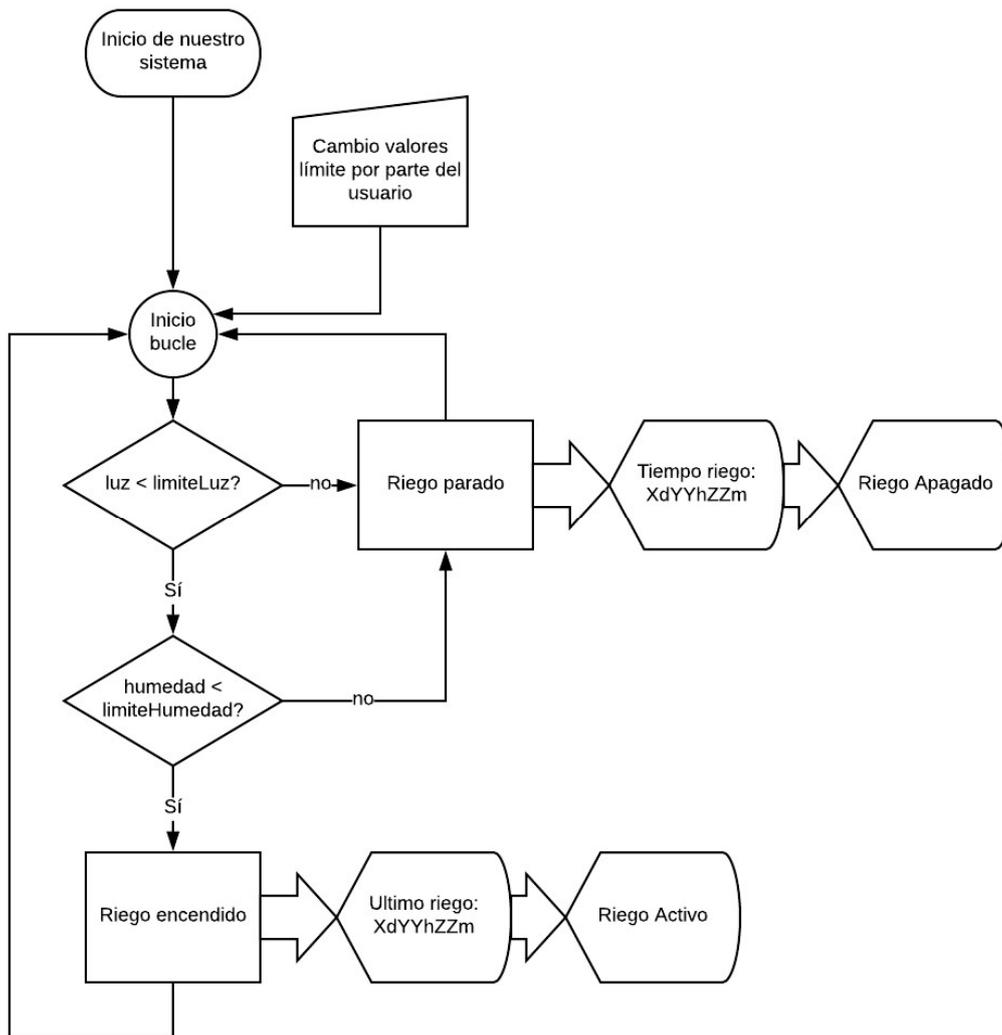


Ilustración 5.2: Diagrama de bloques con interacción por parte del usuario y mensajes en pantalla

## Capítulo 6. Conclusiones y trabajo futuro

Recordemos que nuestro objetivo era tener un sistema que cumpliera los siguientes requisitos:

1. Sistema lo más barato posible
2. Cuyos componentes sean fáciles de adquirir y que dispongamos de múltiples proveedores para cada componente.
3. Sea lo suficientemente flexible para adaptarse a terrenos variados, por lo que necesitamos que sea fácilmente ampliable.
4. Sea lo suficientemente inteligente para detectar cuando debe y puede regar y cuando no, con independencia del clima y las estaciones, requiriendo del usuario la mínima interacción posible para ello.
5. Sea capaz de adaptarse a flora variada con las mínimas variaciones posibles por parte del usuario.

Como hemos visto, el usuario puede modificar la humedad del suelo y la cantidad de luz con la que se riega de forma rápida, pulsando un par de botones, por lo que el último punto lo hemos cumplido.

Respecto a que el sistema sea capaz de saber cuando regar y cuando no, ya hemos visto que nuestro sistema no usa temporizadores, sino que tiene un sistema de toma de decisiones en función de la lectura de sus sensores, por lo que si el clima es más frío o más húmedo, nuestro sistema seguirá manteniendo la tierra con la misma humedad espaciando más los riegos, vemos, por tanto, que este punto también lo hemos cumplido.

Además, hemos elegido unos componentes muy genéricos que podemos comprar tanto en muchas tiendas de electrónica locales, como en multitud de comercios online, no tenemos más que buscar cualquiera de nuestros componentes en eBay, por ejemplo, y veremos que tenemos multitud de coincidencias de distintos vendedores, por lo que el punto 2 también lo cumplimos.

Veamos ahora el aspecto económico mencionado en el punto 1 y la flexibilidad de nuestro sistema mencionado en el punto 3.

Tenemos pues la siguiente tabla de costes para nuestro sistema completo, suponiendo que compramos la electroválvula ya que consideramos que será lo más común y obviando el coste del cableado:

<b>Elemento</b>	<b>Coste por unidad (€)</b>	<b>Número de unidades</b>	<b>Coste total (€)</b>
Arduino Leonardo Compatible	9	1	9
Display 16x2	2,45	1	2,45
Teclado matricial 4x4	1	1	1
Sensor de humedad	0,78	2	1,56
Sensor de luminosidad	0,85	1	0,85
Relé	0,85	1	0,85
Electroválvula 220v	9	1	9
<b>Coste total del sistema</b>			<b>24,71</b>

**Tabla 1: Tabla de costes**

Todos estos precios son para nuestro prototipo, donde hemos usado un Arduino Leonardo, pero para un sistema real hay que recordar que deberíamos usar un Arduino nano, por lo que el coste de nuestro sistema bajaría en 6,5€ y se quedaría en 18,21€.

El coste del cableado variará enormemente en función del terreno a regar y del número de sensores a utilizar, pero la mayor parte del cableado que necesitamos será para conectar los sensores, para ello nos vale un cable fino, por ejemplo, de calibre AWG30, cuyo coste es de unos 0,1€ el metro o incluso menos.

Sumando el coste de la fuente de alimentación (por ejemplo, un cargador usb), el cable usb-A a microusb y los cables para alimentar la electroválvula, es difícil que llegemos a los 25€ de coste, a no ser que tengamos un terreno muy amplio y tengamos que usar muchos sensores con mucha distancia entre ellos.

Vamos a intentar poner nuestra placa de control cerca de la fuente de agua y de la toma de corriente (siempre que el entorno nos lo permita), pues estos cables son los de mayor coste.

Como vemos, hemos conseguido nuestro objetivo de conseguir un sistema barato, flexible y con más prestaciones de los que hay en el mercado, puesto que supera a un sistema comercial cuyo coste es de 80€ (Ilustración 1.3: Programador de riego Gardena) y que sólo dispone de un sensor de humedad y no de sensor de luminosidad. Además, añadir sensores extra apenas nos supone unos céntimos, por lo que podremos adaptarlo a diferentes entornos sin problema.

Nuestro sistema sería fácilmente adaptable para controlar distintas zonas independientes, no tendríamos más que conectar algunos sensores de humedad a otros puertos y asociar esas entradas con otras salidas a las que conectaremos otros relés. Sin tener que cambiar de placa Arduino, tenemos 4 entradas analógicas disponibles y 6 digitales, por lo que podríamos crear un sistema sectorizado en 5 zonas, de forma que sólo se riegue aquella que necesita ser regada, aunque el sistema se encarecerá algo, unos 11€ por zona si sólo colocamos un sensor de humedad en cada zona (9€ de la electroválvula, casi 1€ del relé y casi 1€ del sensor de humedad).

Si aun así quisiéramos tener más zonas, podemos optar por placas con más entradas y salidas, como la Arduino mega, con la que pasamos de 12 entradas analógicas a 16 (nuestra principal limitación) y de 20 I/O digitales a 54 (aunque en realidad esto no nos limita). El precio de esta placa es de entre 10€ y 11€.

O bien usar varias placas Arduino nano, que se podrían montar en forma de sistema aislado o bien comunicadas entre ellas para compartir ciertos elementos como el display, teclado e incluso el sensor de luminosidad. De esta forma, una de las placas sería la maestra, a la que irían conectados todos estos sistemas, y las demás serían las esclavas, cuya principal función sería la de proveer más puertos.

Creemos por tanto que hemos cumplido todos los objetivos marcados al inicio, poniendo al alcance de cualquier usuario un sistema de riego completamente desatendido, fácilmente configurable y adaptable a multitud de entornos, además dicho sistema no requerirá para su instalación ninguna habilidad técnica, puesto que el usuario que lo adquiera sólo debe clavar los sensores de humedad en el suelo, conectar la electroválvula entre la toma de agua y el sistema de riego que tuviera con anterioridad (goteo, manguera, etc.), encender nuestro dispositivo y configurarlo, por lo que su instalación no toma más que unos minutos y a partir

de ahí el sistema de riego es totalmente desatendido, sin tener que ser modificado en invierno, verano o en los días de lluvia.

Dicho sistema será útil para múltiples perfiles de usuarios, desde aquellos que quieren despreocuparse del riego, como aquellos que por diversas circunstancias no pueden atenderlo, como por ejemplo agricultores que tengan que desplazarse normalmente sólo para regar sus cultivos y que hasta ahora no podían instalar sofisticados sistemas de riego por su coste, o usuarios con jardines, macetas, huertos urbanos, etc. que, en ciertos periodos, como vacaciones, no pueden atender el riego de sus plantas.

## Capítulo 7. Manual de usuario

Veamos ahora como funciona nuestro prototipo.

Nada más encenderlo se nos muestra en el display la pantalla principal, donde vemos si el riego está activo o no y nos avisa de que si pulsamos la tecla C podemos entrar al menú. En caso de que el riego se active nada más encenderlo, veremos también que hace pocos segundos que se regó por última vez, esto es debido a que el contador de tiempo se pone a 0 al encender el dispositivo.

Si pulsamos la tecla C se nos muestra por pantalla el menú principal, donde vemos que las teclas para entrar al menú para configurar la humedad es la tecla A y para entrar al menú de luminosidad es la tecla B. Sin embargo, estos son accesos directos globales, de forma que podemos entrar a cualquiera de los dos menús tanto desde aquí como desde la pantalla principal pulsando A o B.

Si desde el menú principal o desde cualquiera de los otros menús pulsamos la tecla D, volveremos a la pantalla principal sin modificar ningún valor.

Al pulsar la tecla A entramos en el menú de configuración del sensor de humedad, de forma que se nos presenta la información de que podemos seleccionar 10 niveles de configuración, siendo el 0 el nivel de menor humedad y 9 el nivel donde seleccionamos que queremos la máxima humedad posible.

Una vez seleccionado el nivel de humedad volvemos a la pantalla principal, pero si el nuevo valor cambia el estado del riego, antes de volver veremos la pantalla de transición de regando a no regando o de no regando a regando.

Al pulsar la tecla B entramos en el menú de configuración de la luminosidad, de forma que, como en el menú de humedad, se nos informa de que tenemos 10 niveles de luminosidad disponibles, desde el 0 otra vez el más restrictivo, es decir, con 0 admitimos la mínima cantidad de luz posible para que el riego se inicie y con 9 admitimos la máxima cantidad de luz posible para que el riego se inicie.

Una vez seleccionado el nivel de luminosidad volvemos a la pantalla principal, pero si el nuevo valor cambia el estado del riego, antes de volver veremos la pantalla de transición de regando a no regando o de no regando a regando.

En la pantalla de transición de un estado: “Riego Activo” a un estado: “Riego Apagado” vemos la información del tiempo que ha durado el riego.

En la pantalla de transición de un estado: “Riego Apagado” a un estado: “Riego Activo” vemos la información del tiempo que hace que se regó por última vez.

## Apéndice A. Código del programa

Aquí podemos ver el código fuente del programa:

```
#include <elapsedMillis.h>

#include <Keypad.h>

#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x3F, 16, 2); // Inicia el LCD en la
dirección 0x3F, con 16 caracteres y 2 líneas

const byte ROWS = 4; //four rows
const byte COLS = 4; //four columns
char keys[ROWS][COLS] = {
    {'1','2','3','A'},
    {'4','5','6','B'},
    {'7','8','9','C'},
    {'*','0','#','D'}
};

byte rowPins[ROWS] = {11, 10, 9, 8}; //connect to the row
pinouts of the keypad

byte colPins[COLS] = {7, 6, 5, 4}; //connect to the column
pinouts of the keypad
```

```
//pin sensor humedad
const int humedadPin = A0;
//pin del rele
int relePin = 12;
//pin sensor luz
int luzPin = A1;
//variable para contar tiempo encendido
elapsedMillis timeElapsed;
int daysElapsed;
int hourElapsed;
int segElapsed;
int minElapsed;
boolean previousStatusOn = false;
int limiteHumedad = 500;
int limiteLuz = 100;
char key = '9';
int humedad;
int luz;

Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins,
ROWS, COLS );

//configuración inicial de nuestro sistema
void setup(){
  lcd.init();
  lcd.backlight();
  Serial.begin(9600);
```

```
pinMode(relePin, OUTPUT);
}

void loop(){
    humedad = analogRead(humedadPin);
    luz = analogRead(luzPin);
    key = keypad.getKey();
    int key_1 = key - 48;

    if (key == 'A'){
        menuHumedad();
    }
    if (key == 'B'){
        menuLuz();
    }
    if (key == 'C'){
        printMenu();
    }
    if (key == 'D' && previousStatusOn == true){
        printEncendido();
    }
    if (key == 'D' && previousStatusOn == false){
        printApagado();
    }

    printEnLoop();

    //mostramos por el puerto serie el tiempo transcurrido
```

```
Serial.print("timeElapsed: ");  
  
Serial.print(timeElapsed);  
  
if(humedad > limiteHumedad && luz > limiteLuz &&  
previousStatusOn == false)  
{  
    empezarRegar();  
}  
  
else if((humedad < limiteHumedad || luz < limiteLuz) &&  
previousStatusOn == true){  
    pararRegar();  
}  
}
```

```
void empezarRegar() {  
    digitalWrite(relePin, HIGH);  
    calcTimeElapsed();  
    printTimeElapsedLastRiego();  
    printEncendido();  
    timeElapsed = 0;  
    previousStatusOn = true;  
}
```

```
void pararRegar() {  
    digitalWrite(relePin, LOW);  
    calcTimeElapsed();  
    printTimeElapsedRegando();  
    printApagado();  
}
```

```
previousStatusOn = false;

timeElapsed = 0;

}

//función para imprimir el menú de para acceder a la
configuración de los sensores de humedad o luz

void printMenu() {

    lcd.setCursor(0, 0);

    lcd.print("Elige humedad: A");

    lcd.setCursor(0, 1);

    lcd.print("Elige luz: B    ");

    lcd.print(key);

}

//función para imprimir el menú de configuracion del sensor
de humedad

void menuHumedad() {

    lcd.clear();

    lcd.setCursor(0, 0);

    lcd.print("Elige humedad0-9");

    lcd.setCursor(0, 1);

    lcd.print("Seleccionado: ");

    key = keypad.waitForKey();

    limiteHumedad = 1000 - (80 * (key - 48));

    lcd.print(key);

    delay(1000);

    if(previousStatusOn) {
```

```
    printEncendido();
}
else{
    printApagado();
}
}

//función para imprimir el menú de configuracion del sensor
de luz
void menuLuz() {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Elige luz:0-9");
    lcd.setCursor(0, 1);
    lcd.print("Seleccionado: ");
    key = keypad.waitForKey();
    limiteLuz = 600 - (66 * (key - 48));
    lcd.print(key);
    delay(1000);
    if(previousStatusOn) {
        printEncendido();
    }
    else{
        printApagado();
    }
}
```

```
//función que muestra el menu con la información de que el
riego está encendido
void printEncendido(){
    lcd.setCursor(0, 0);
    lcd.print("Riego Activo      ");
    lcd.setCursor(0, 1);
    lcd.print("Menu: C          ");
    //Ahora imprimimos por el puerto serie que empezamos a
regar
    Serial.print("Encendido");
    Serial.println(" ");
    Serial.print("humedad: ");
    Serial.print(humedad);
    Serial.print("limiteHumedad: ");
    Serial.print(limiteHumedad);
    Serial.println(" ");
}

//función que muestra el menu con la información de que el
riego está apagado
void printApagado(){
    lcd.setCursor(0, 0);
    lcd.print("Riego Apagado      ");
    lcd.setCursor(0, 1);
    lcd.print("Menu: C          ");
}
}
```

```
//esta funcion muestra por el puerto serie toda la info
void printEnLoop(){
    Serial.print("humedad: ");
    Serial.print(humedad);
    Serial.print(" limiteHumedad: ");
    Serial.print(limiteHumedad);
    Serial.print(" limiteLuz: ");
    Serial.print(limiteLuz);
    Serial.print(" luz: ");
    Serial.print(luz);
    Serial.println(" ");
}

//función para calcular el tiempo transcurrido desde que se
reinició el contador en formato dias, horas, minutos y
segundos
void calcTimeElapsed(){
    long timeRemaining = timeElapsed;
    daysElapsed = timeRemaining/86400000;
    timeRemaining = timeRemaining - daysElapsed*86400000;
    hourElapsed = timeRemaining/3600000;
    timeRemaining = timeRemaining - hourElapsed*3600000;
    minElapsed = timeRemaining/60000;
    timeRemaining = timeRemaining - minElapsed*60000;
    segElapsed = timeRemaining/1000;
}
```

```
//función que muestra por pantalla la información de cuanto tiempo se ha estado regando
```

```
void printTimeElapsedRegando() {  
    lcd.clear();  
    lcd.setCursor(0, 0);  
    lcd.print("Tiempo riego:   ");  
    lcd.setCursor(0, 1);  
    lcd.print(daysElapsed);  
    lcd.print("d");  
    lcd.print(hourElapsed);  
    lcd.print("h");  
    lcd.print(minElapsed);  
    lcd.print("\'");  
    lcd.print(segElapsed);  
    lcd.print("\'");  
    delay(3000);  
}
```

```
//función que muestra por pantalla la información de cuanto tiempo hace que se regó por última vez
```

```
void printTimeElapsedLastRiego() {  
    lcd.clear();  
    lcd.setCursor(0, 0);  
    lcd.print("Ultimo riego:   ");  
    lcd.setCursor(0, 1);  
    lcd.print(daysElapsed);  
    lcd.print("d");  
}
```

```
lcd.print(hourElapsed);  
lcd.print("h");  
lcd.print(minElapsed);  
lcd.print("\'");  
lcd.print(segElapsed + 3);  
lcd.print("\");  
delay(3000);  
}
```

## Apéndice B. Funcionamiento de los sensores de humedad (Sensores de Humedad, s.f.)

El principio de funcionamiento de estos sensores es el siguiente: Al introducir una pequeña corriente A.C. a través de un suelo o sustrato mediante una diferencia de potencial entre dos electrodos se presenta una impedancia opuesta al paso de la corriente. Esta impedancia es inversamente proporcional a la humedad del medio y la corriente eléctrica que pasa,  $I_t$ , es directamente proporcional al contenido volumétrico de humedad del sustrato o suelo.

La conducción de corriente entre dos electrodos,  $I_t$ , es igual a la Conductancia  $K_t$  multiplicada por el voltaje aplicado  $V_t$ . ..... ( $I_t = K_t \times V_t$ )

La conductancia a su vez,  $K_t$ , es igual al producto de varios factores a saber:

- 1.- Una constante  $K_s$  que depende del sustrato.
- 2.- Del contenido Volumétrico de Humedad.
- 3.- De la Conductividad Eléctrica del agua contenida en los poros del sustrato.
- 4.- De la geometría de los electrodos, función en este caso representada por la Constante de celda  $K_c$ . También, aunque en menor medida, del diámetro de

los electrodos. Este factor también está comprendido dentro de la Constante de celda.

5.- Del íntimo contacto entre los electrodos y el Sustrato;

6. De la Temperatura del sustrato, aunque esta es una variable de pequeña variación en suelos agrícolas tropicales.

$K_t = K_s \times f(\% \text{ volumétrico de Humedad}) \times f(\text{C.E. Intersticial}) \cdot f(\text{Contacto}) \cdot f(\text{temp})$

La Conductividad Electrica Granel CEG es igual

$CEG = K_t \times K_c$

Cuando la Longitud de los electrodos es igual a la distancia entre los mismos,  $L/D = 1$ , entonces  $K_c$  tiende a ser constante. De aquí se desprende que todos los electrodos que tengan  $L/D = 1$  darán aproximadamente la misma lectura independientemente de su tamaño. Esto se ha ensayado para  $L/D = 1$  y Diámetro  $< 1/15 D$ .

La Calibración de los sensores se realiza mediante inmersión de los mismos en una solución de conductividad eléctrica conocida.

## Referencias

- [1] <https://www.arduino.cc/en/Main/Products>
- [2] <https://store.arduino.cc/arduino-uno-rev3>
- [3] <https://store.arduino.cc/arduino-leonardo-with-headers>
- [4] <https://store.arduino.cc/arduino-micro>
- [5] <https://store.arduino.cc/arduino-nano>
- [6] <https://store.arduino.cc/arduino-mega-2560-rev3>
- [7] <https://www.indiegogo.com/projects/9-arduino-compatible-starter-kit-anyone-can-learn-electronics#/>
- [8] <https://www.cocopot.es/programadores-de-riego/374-programador-de-riego-natrain-sin-presi%C3%B3n-8424385250812.html>
- [9] <http://www.leroymerlin.es/fp/16789892/programador-de-grifo-geolia-1-via>
- [10] <https://www.gardena.com/es/productos/riego/programadores-de-riego/easycontrol/967235201/>
- [11] <https://www.ebay.com/itm/2-0-320x240-Serial-UART-I2C-SPI-TFT-LCD-Display-for-Arduino-ESP8266-PI-PIC-AVR-/253463244342>
- [12] <https://thingnovation.com/modules-modulos/displays-lcd-tft/modulo-lcd2004-i2c-2-arduino-display-lcd-uno-r3-mega-20x4-2004-azul-o-verde-478.html>
- [13] <https://www.ebay.es/itm/IIC-I2C-TWI-SP-I-Serial-Interface1602-16X2-Character-LCD-Module-Display-Yellow/252513792517?hash=item3acafeb205:g:sUQAAOSwu4BVk1Gj>

- 
- [14] <https://www.ebay.es/itm/PS-2-Adapter-PS-2-PS2-Keyboard-keypad-Tastefeld-Modulo-For-Arduino-PS2-New/112569761747?hash=item1a35ae3fd3:g:j~AAAOSwIINZwW1I>
- [15] <https://www.ebay.es/itm/4x3-Matrix-Array-12-Key-Membrane-Switch-Keypad-Keyboard-f-Arduino-AVR/162721368238?hash=item25e2f308ae:g:J04AAOSwa81Z7AtH>
- [16] <https://www.ebay.es/itm/16-Key-4x4-Matrix-Array-Membrane-Keypad-Keyboard-AVR-12V-Arduino-Module/322968093773?hash=item4b3266044d:g:Vc4AAOSw6MNaRwIH>
- [17] <http://www.circuitstoday.com/interfacing-hex-keypad-to-arduino>
- [18] <https://www.ebay.es/itm/1-x-Medidor-De-Humedad-Del-Suelo-Deteccion-Sensor-Modulo-Robot-Car-Para-Arduino/162091877029?hash=item25bd6dc2a5:g:On0AAOSwLnBXUX~U>
- [19] [https://www.ebay.es/itm/10pcs-Soil-Humedad-Higrometro-Sensor-Modulo-Humedad-Deteccion-para-Arduino-AVR/392030300834?hash=item5b46d3aaa2:m:mIotBQdQp4DofXlejIMhZa\\_w](https://www.ebay.es/itm/10pcs-Soil-Humedad-Higrometro-Sensor-Modulo-Humedad-Deteccion-para-Arduino-AVR/392030300834?hash=item5b46d3aaa2:m:mIotBQdQp4DofXlejIMhZa_w)
- [20] <https://www.luisllamas.es/arduino-humedad-suelo-fc-28/>
- [21] <https://www.ebay.es/itm/20PCS-Photoresistor-LDR-CDS-5mm-Light-Dependent-Resistor-Sensor-GL5516-Arduino-I/263551651275?hash=item3d5ce70dcb:g:25QAAOSwyXFaoETU>
- [22] <https://www.luisllamas.es/medir-nivel-luz-con-arduino-y-fotoresistencia-ldr/>
- [23] <https://www.ebay.es/itm/Modulo-de-Rele-Sensor-de-Luz-Fotosensible-LM393-Rele-para-Arduino-SA/332645540164?hash=item4d73382144:g:EwQAAOSweKda8kkk>
- [24] <https://www.ebay.es/itm/1x-Modulo-Rele-5V-1-Canal-Arduino-ONE-CHANNEL-RELAY-MODULE-BOARD-SHIELD/331951429221?hash=item4d49d8d665:g:JRIAAOSwknJX1nBk>
- [25] <https://www.banggood.com/es/220V-2-Way-Normally-Closed-Brass-Electric-Solenoid-Valve-for-Air-Water-Valve-p->

[1210247.html?gmcCountry=ES&currency=EUR&utm\\_source=googleshopping&utm\\_medium=cpc\\_ods&utm\\_content=heath&utm\\_campaign=plamech-es-pc&cur\\_warehouse=CN](https://www.nxp.com/docs/en/application-note/AN10216.pdf)

[26] <https://www.nxp.com/docs/en/application-note/AN10216.pdf>

[27] [http://www.drcalderonlabs.com/Aparatos/Sensores\\_de\\_Humedad/Sensores\\_de\\_Humedad.htm](http://www.drcalderonlabs.com/Aparatos/Sensores_de_Humedad/Sensores_de_Humedad.htm)

[28] <https://playground.arduino.cc/Code/ElapsedMillis>

[29] <https://playground.arduino.cc/Code/Keypad>

[30] <https://github.com/fdebrabander/Arduino-LiquidCrystal-I2C-library>