



# **Desarrollo de Aplicaciones Distribuidas**

PROYECTO: POOLMATIK

Miembros:     Jose Adrian Diaz Navarro  
                     Jesus Lobato Salas

# Índice

1.Introducción.....	2
2.Componentes.....	3
3.Diseño .....	6
4. Análisis Competencia .....	8
5.Diagrama y descripción BD .....	9
6.API REST.....	10
7.Código.....	11

# 1.Introducción

Hoy día con el desarrollo tecnológico es posible automatizar cualquier tipo de tareas que se nos ocurra. En este proyecto nos basaremos en el concepto del IOT para poder monitorizar piscinas e independizar sus tareas lo máximo posible de la acción humana.

Para realizar dicha labor vamos a diseñar un prototipo que se encargue de:

- Medir los niveles de cloro y PH: esto se realizará mediante sensores de ph y cloro que recogerán datos en tiempo real.
- Medir cantidad de productos químicos: en el mantenimiento de una piscina se requieren como mínimo reductor de ph, incrementador de ph y cloro que han de ser líquidos para poder medir la cantidad restante mediante ultrasonidos.
- Controlar temperatura: según el clima externo el sistema regulará la temperatura garantizando un consumo mínimo del calentador del agua y por tanto un mayor ahorro energético.
- Control toldo: el sistema se encargará de recoger o desplegar el toldo según el horario asegurando que en las horas más calurosas los bañistas se despreocupen de quemarse la piel.
- Control cubre piscinas: de forma remota se podrá recoger o desplegar la tela que cubre la piscina
- Control de depuradora: de forma automática se activará la depuradora según el horario y la última vez que se realizó la depuración.
- Control de la iluminación: el sistema se encargará de recoger mediciones de luz del exterior para que cuando se haga de noche se enciendan los focos de la piscina y las luces del exterior.

El servidor se encargará de recopilar los niveles de ph, cloro y temperatura para actuar en consecuencia y conseguir los niveles óptimos en cada momento. El sistema podrá controlar los niveles de productos químicos que quedan en el tanque y mediante machine learning calcule cuándo se acabará dicho producto.

Además todos estos datos podrán consultarse remotamente desde un dispositivo con conexión a internet y accionar manualmente el encendido de la depuradora, recogida o desplegado del cubrepiscinas y toldo y selección de los niveles deseados de ph, cloro y temperatura.

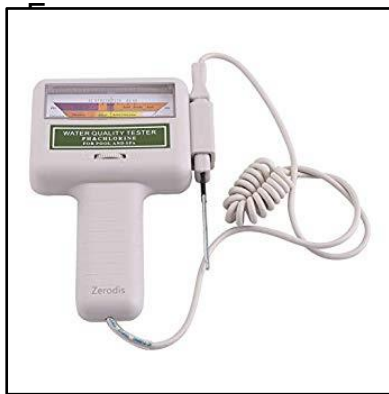
Es necesario que se tengan en cuenta valores como la concentración de cada producto químico utilizado para dosificar la cantidad necesaria.

## 2.Componentes

Los componentes que utilizaremos para la realización del proyecto dependerá de cada uno de los objetivos listados anteriormente. Además, usaremos la placa de desarrollo **Arduino** con su módulo **ESP8266(ESP-01)** que permitirá la comunicación con el cliente mediante WIFI. A continuación, podemos ver cada objetivo junto con los componentes que serán utilizados:

- Control PH y cloro:

Estos sensores estarán integrados en unas derivaciones “T” que se acoplan a los tubos que llevan el agua a la depuradora. Los sensores utilizados son el **PH4502C** de **DIY MORE** para medir el PH (Fig. 2) y un medidor de la marca **ZERODIS** (Fig 1) adaptado para poder leer los niveles de cloro.



9



2

- Control nivel de productos químicos:

Mediante un sensor de ultrasonidos **HC-SR04** (Fig. 3) colocado en la parte superior del tanque, se medirá la distancia entre el producto líquido y el sensor determinando así la cantidad restante en el interior del tanque.



Fig.3

- Control toldo y cubrepiscinas:

Usaremos dos motores eléctricos (Fig.4) para simular el control del cubrepiscinas y toldo.



Fig. 4

- Control de temperatura:

Mediremos tanto la temperatura ambiente como la de la propia piscina mediante dos sensores de temperatura uno encapsulado que se trata del sensor **DS18B20** (Fig. 5) para medir la temperatura del agua y el sensor **DHT11** (Fig. 6) que aparte de medir la temperatura exterior también mide la humedad, función que no usaremos en este proyecto.



Fig. 5

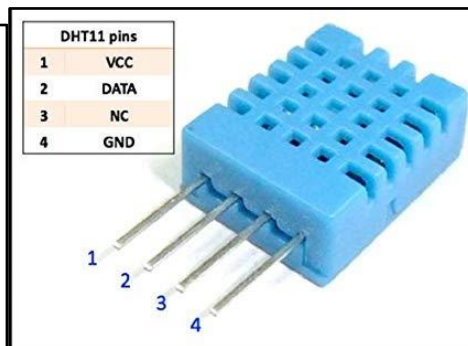


Fig. 6

- Control depuradora:

Debido a que no disponemos de una depuradora, usaremos un led (Fig. 7) para simular el comportamiento de la misma según los datos almacenados o el encendido manual de esta a través de un smartphone o pc.



Fig. 7

- Control iluminación:

Este objetivo se llevará a cabo mediante una fotoresistencia LDR (Fig. 8) el cual medirá la cantidad de luz para que active el sistema de iluminación según la necesidad.

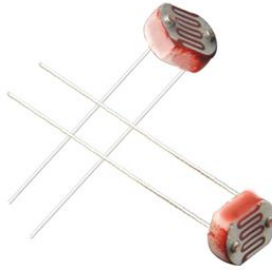
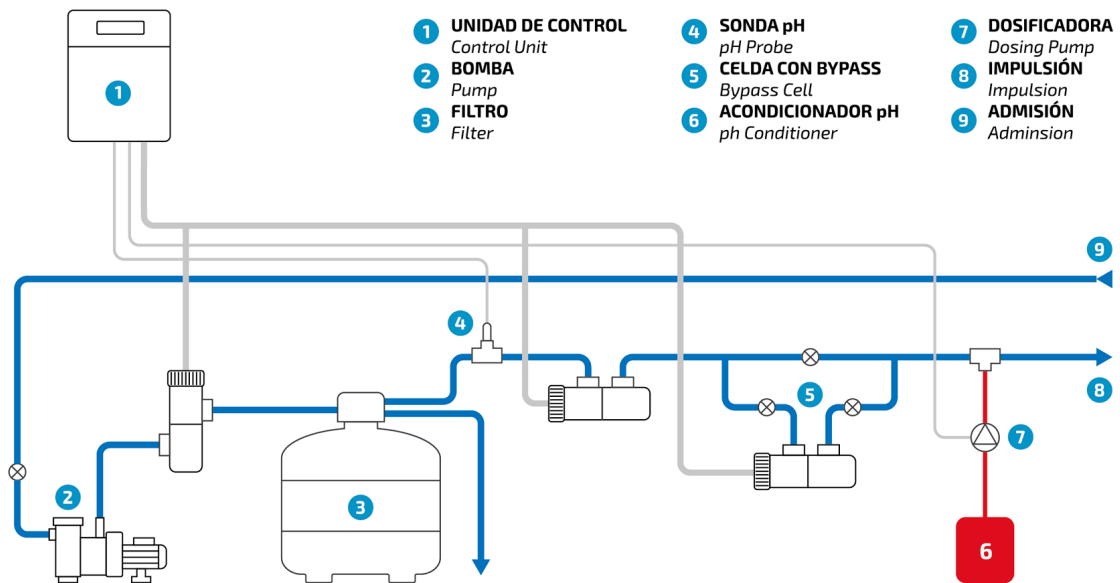


Fig. 8

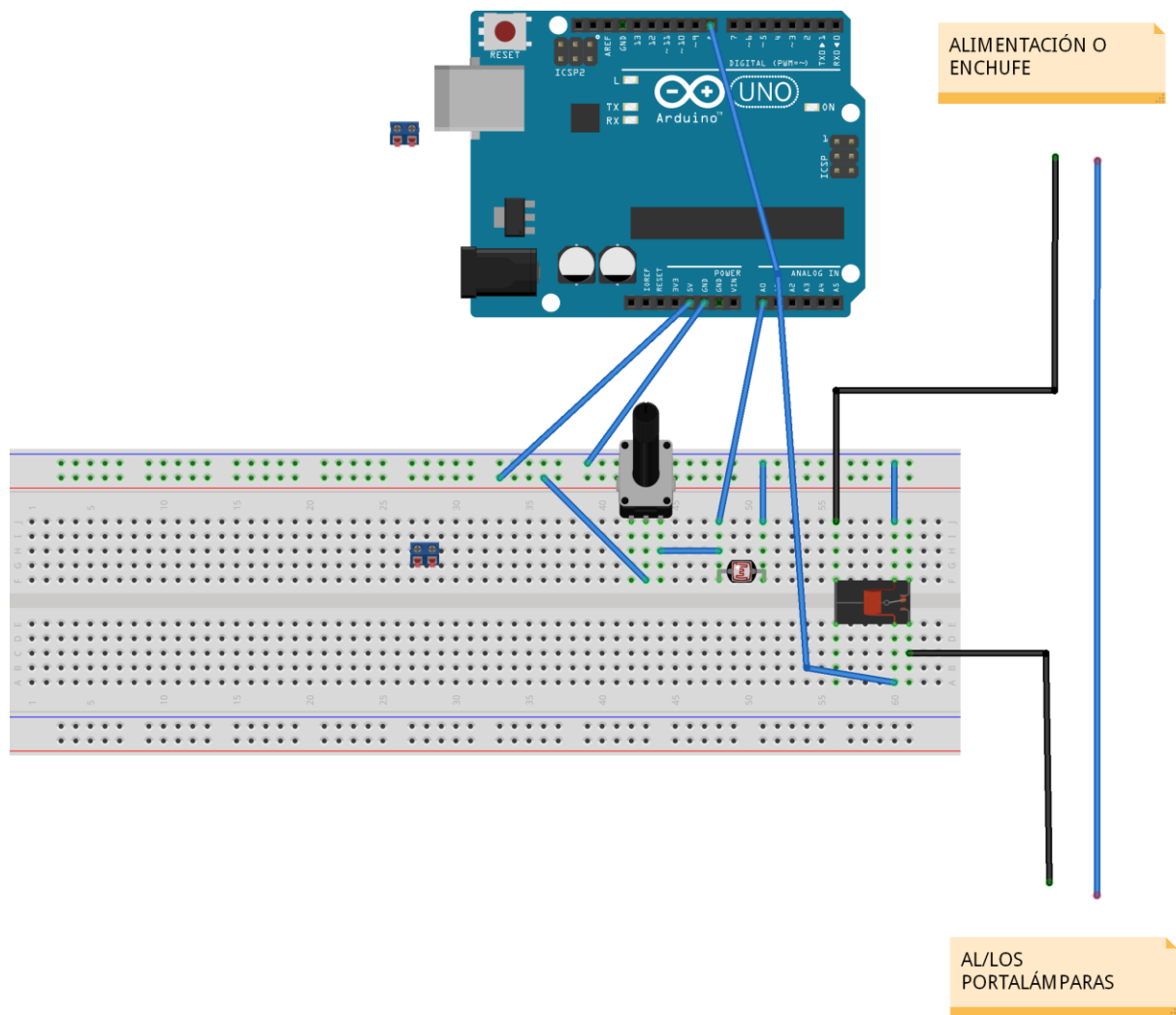
### 3.Diseño



A la salida de la depuradora se coloca la sonda de ph y cloro. Tras ellos siguen sus respectivos dosificadores.



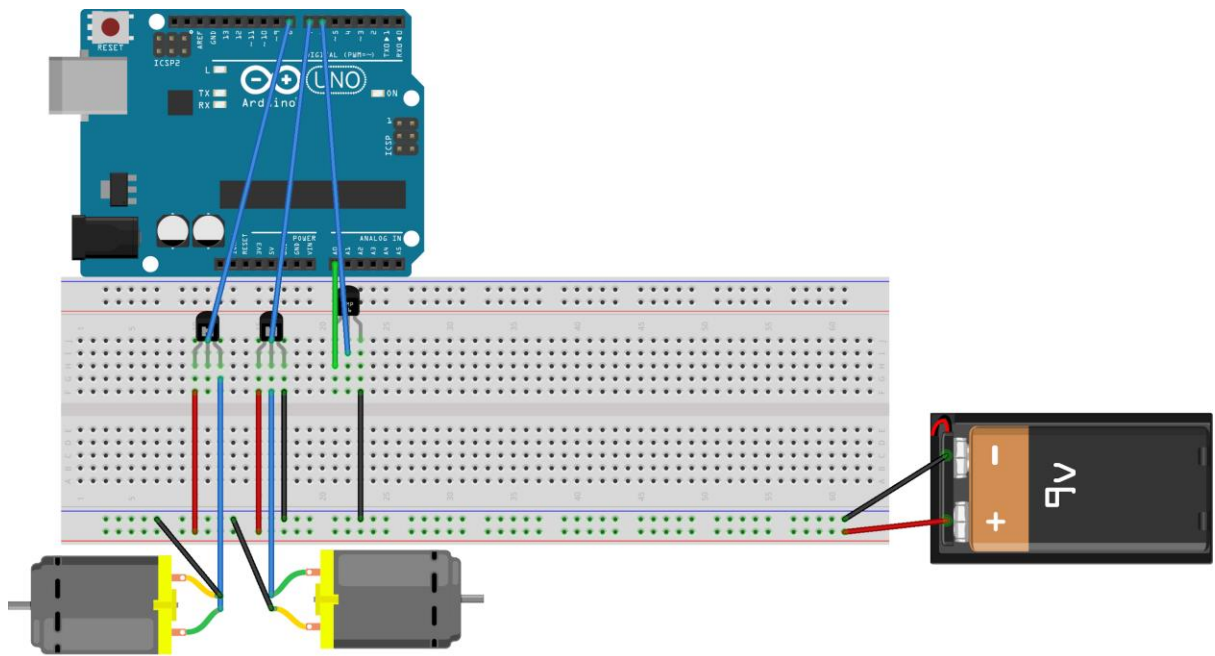
Sensor de ultrasonidos para medir nivel de productos químicos



fritzing

Aquí tenemos el esquema eléctrico del sistema de iluminación automático que gracias al potenciómetro, puede ser calibrado según los niveles de luz de distintos entornos.





fritzing

En este esquema se representan las conexiones para el funcionamiento del toldo y cubrepiscinas además de la medición de luz exterior.

## 4. Análisis Competencia

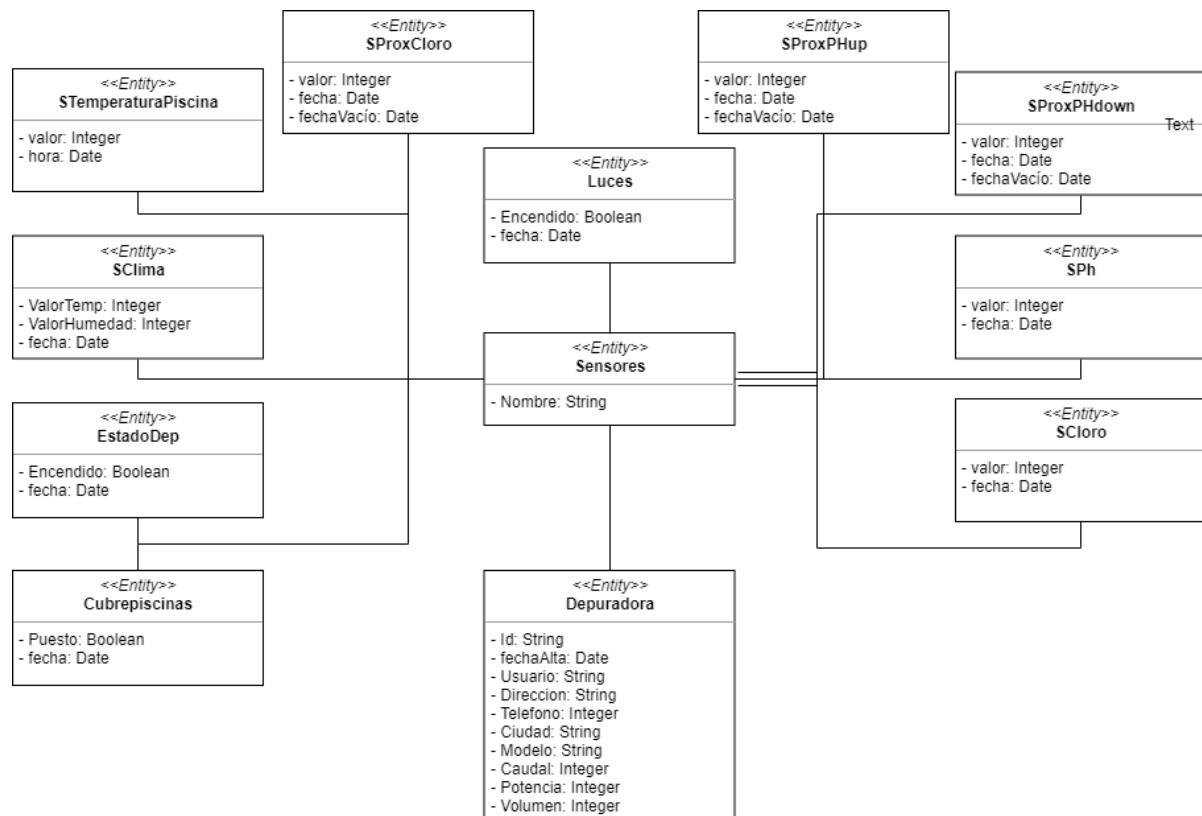
En el mercado existen productos que automatizan el mantenimiento de una piscina con precios muy elevados(entre 700 y 2000 euros). Con este proyecto se pretende disminuir la inversión y proporcionar más funciones útiles para el consumidor.

El coste de los componentes es:

- NodeMCU 8€
- sensor luz y transistores aprox. 4€
- sensor ph y cloro 12€ y 30€ respectivamente.
- sensor ultrasonidos 4€
- total 58 €

El resto de componentes(motores, leds..) los usamos a modo de representación a elementos externos que el consumidor debe disponer por lo que no valoramos dichos gastos en nuestro producto.

## 5.Diagrama y descripción BD



La entidad Depuradora almacena los datos de cada usuario y la depuradora. En torno a éste se añadirán los sensores necesarios que almacenarán los datos que envíe un ESP específico de cada depuradora.

La tabla sensores almacena todos los sensores existentes con una **foreign key** a una depuradora, el nombre se refiere al tipo de sensor que va referido a su tabla correspondiente. De esta manera es posible consultar los sensores que contiene una piscina mediante la clave foránea de la depuradora.

El resto de tablas almacenan los datos enviados por el ESP8266 con sus respectivos valores leídos mediante los sensores, el campo fecha lo genera el servidor automáticamente al introducir nuevos datos en las tablas.

## 6.API REST

### Consultas GET:

[/sensores/sensclima/:idSen](#) método: handleClima

Esta consulta nos devuelve el registro de valores devueltos por el ESP sobre el clima en el que se encuentra la piscina. **:idSen** es el id del sensor a consultar

[/sensores/:nombre/:idSen](#) método: handleSpecificSens

Nos devuelve el registro de valores del tipo de sensor que indicamos en **:nombre** y en **:idSen** nos referimos el id del sensor para indicar el de nuestra piscina.

[/:idDep/sensores/all](#) método: handleAllSensors

Devuelve los sensores pertenecientes a una piscina introduciendo el id de la depuradora en **:idDep**.

[/:idDep/sensores/:sensor](#) método: handleSensors

Devuelve los sensores pertenecientes a una piscina ( **:idDep** ) del tipo sensor indicado en **:sensor**

[/:idDep/luces](#) método: handleLuces

Devuelve el registro de encendido/apagado de las luces de la piscina ( **:idDep** )

[/:idDep/estados](#) método: handleDepEncendido

Devuelve el registro de encendido/apagado de la depuradora ( **:idDep** )

### Consultas POST:

[/sensores/agregar](#) método: handleAgrega2

En el cuerpo se envía un Json con las claves **"idDep"** de tipo int y **"nombre"** de tipo String. Con ésta petición se agregan nuevos sensores a una depuradora ( **"idDep"** ) de tipo ( **"nombre"** )

[/sensores/:nombre/agregar](#) método: handleAgrega1

Agrega nuevas entradas a las tablas de los sensores para registrar valores nuevos con su fecha de lectura. **:nombre** indica el tipo de sensor, es decir, la tabla. El cuerpo contiene las claves **"idSen"** que es el id del sensor en la tabla sensores y **"valor"**, de tipo entero, que es el valor leído.

[/luces/ONOF](#) método: handleLucesONOF

Registra el encendido o apagado de las luces de la piscina. El cuerpo contiene las claves **"idDepuradora"** y **"encendido"** que es 0 si se ha apagado o 1 si se ha encendido.

[/dep/ONOF](#) método: handleDepONOF

Registra el encendido o apagado de la depuradora de la piscina. El cuerpo contiene las claves **"idDepuradora"** y **"encendido"** que es 0 si se ha apagado o 1 si se ha encendido.

## 7.Código

El código se encuentra en el siguiente repositorio público, dentro de la carpeta Vertx/servidor:

<https://github.com/dnadri1398/Piscimatic>

La carpeta clases contiene las clases necesarias para crear objetos de sensores usados al introducir entradas en las tablas.