|  |  |
| --- | --- |
|  | **2016** |
|  | Alejandro Rosales |

|  |
| --- |
| **Alimentador de Mascotas** |
|  |

# Índice

# 1-Introduccion

# 2-Funcionamiento

# 3-Elementos Utilizados

# 3.1-Hardware

# 3.2-Software-

# 4-Circuito

# 5-Codigo Arduino

# 6-Base de Datos

# 7-Manual de Usuario

# 1-Introducción

A mucha gente le pasa que se olvida de alimentar a sus mascotas o simplemente está ocupada con los quehaceres del hogar y no pueden hacerlo, este dispositivo facilita esa tarea, haciendo que pueda alimentar a su mascota con solo un click en su teléfono celular, desde cualquier parte con tan solo tener conexión a internet.

# 2-Funcionamiento

El alimentador funciona como un servidor que espera ser accionado por la aplicación y almacenara el historial de las veces que se alimentó a la mascota.

Desde la aplicación se puede chuequear si el alimentador está conectado, alimentar a la mascota y obtener el historial del alimentador, todo esto se realizara a través de la conexión a internet del teléfono, además de esto se puede programar una alarma para que se alimente a la mascota automáticamente a algún horario del día o en intervalos de 4, 8, 12 y 24 horas.

# 3-Elementos utilizados

# 3.1 Hardware:

-Arduino Mega

-Placa Ethernet Shield para Arduino

-Puente H (Circuito Integrado L293D)

-Alimentador echo con materiales reciclados

-Mecanismo de impresora que genera un movimiento con Motor 9v

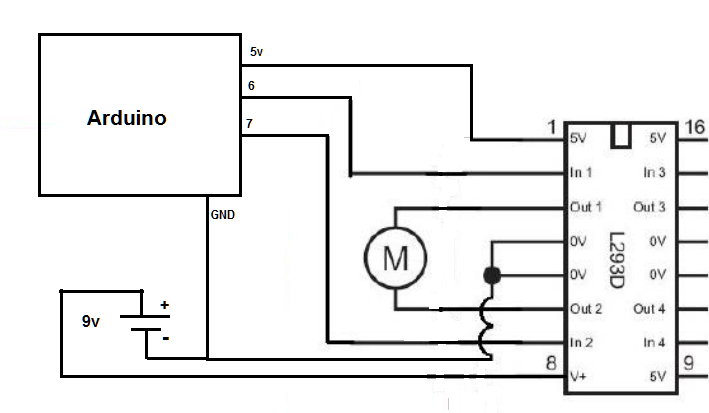
# 3.2 Software:

-Programación en Arduino: Incluye el manejo básico de las entradas y salidas, creación de un servidor web con recepción pedidos GET y envió de respuestas a los pedidos.

-Programación en Android: Incluye conexión a servidores web con envío de datos y recepción de respuesta, Bases de datos embebidas SQLlite, diseño gráfico de una aplicación, aplicación adaptable a distintos dispositivos, etc.

# 4-Circuito

El arduino controla el motor atravez del puente H L293D



# 5-Codigo Arduino

l#include <SPI.h>

#include <Ethernet.h>

byte mac[] = {

0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED

};

IPAddress ip(192, 168, 1, 177);

EthernetServer server(8080);

//pines que se utilizan para el motor

int Mizquierda=6;

int Mderecha=7;

//contenedor del pedido

String readString = String(30);

//contenedor del historial

String historial="";

void setup() {

Serial.begin(9600);

while (!Serial) {

; // espera a q se conecte el puerto serial

}

pinMode(Mizquierda,OUTPUT);

pinMode(Mderecha,OUTPUT);

digitalWrite(Mizquierda,LOW);

digitalWrite(Mderecha,LOW);

Ethernet.begin(mac, ip);

//ejecuta el servidor

server.begin();

}

void loop() {

EthernetClient client = server.available();

if (client) {

while (client.connected()) {

Serial.println("new client");//avisa nuevo cliente

if (client.available()) {

char c = client.read();

if(readString.length()<30) {

//lee lo q el cliente "escribio" letra por letra(URL)

readString.concat(c);

}

//si hay un pedido al servidor

if (c == '\n') {

//a es igual al valor de cantidad de caracteres

//del URL antes del "a="

int a = readString.indexOf("a=");

if(readString.substring(a,a+7)=="a=chico") {

//guarda los datos(hora/fecha) en el historial

actHistorial(readString.substring(a+7));

//devuelve una respuesta al cliente

client.println("HTTP/1.1 200 OK");

client.println();

client.println("1");

client.stop();

//acciona el alimentador

abrir();

delay(1000);

cerrar();

}else if (readString.substring(a,a+9)=="a=mediano") {

actHistorial(readString.substring(a+9));

client.println("HTTP/1.1 200 OK");

client.println();

client.println("2");

client.stop();

//acciona el alimentador

abrir();

delay(2000);

cerrar();

}else if (readString.substring(a,a+8)=="a=grande") {

actHistorial(readString.substring(a+8));

client.println("HTTP/1.1 200 OK");

client.println();

client.println("3");

client.stop();

//acciona el alimentador

abrir();

delay(3000);

cerrar();

}else if (readString.substring(a,a+6)=="a=hola") {

Serial.println("hola");

client.println("HTTP/1.1 200 OK");

client.println();

client.println("Conectado");

client.stop();

}else if (readString.substring(a,a+16)=="a=ActualizarHist") {

//envia el historial al cliente

client.println("HTTP/1.1 200 OK");

client.println();

client.println(historial);

client.stop();

}else if (readString.substring(a,a+12)=="a=BorrarHist") {

//limpia el string del historial

historial="";

//ennvia al cliente que se elimino

Serial.println("historial");

client.println("HTTP/1.1 200 OK");

client.println();

client.println("borrado");

client.stop();

}

client.stop();

readString="";

}

}

}

}

}

void abrir(){

digitalWrite(Mizquierda,HIGH);

delay(2000);

digitalWrite(Mizquierda,LOW);

}

void cerrar(){

digitalWrite(Mderecha,HIGH);

delay(2000);

digitalWrite(Mderecha,LOW);

}

void actHistorial(String dato){

//si el tamaño que del historial es grande lo limpia

if(historial.length()>500){

historial="";

}else if (historial.length()<2){

//si es el primer registro ingresado

historial=dato;

}else{

//si hay mas de 1 registro

historial=historial+"-"+dato;

}

}

# 6-Base de datos

Tablas de la base de datos ubicada dentro de la aplicación:

|  |  |  |
| --- | --- | --- |
| Historial | | |
| **Nombre** | **Tipo de Dato** | **Descripción** |
| HoraFecha | Texto | Contiene la fecha en la que se alimentó la mascota |
| Cantidad | Texto | Contiene el tipo de mascota que se alimentó(Chica, Mediana, Grande) |

|  |  |  |
| --- | --- | --- |
| Config | | |
| **Nombre** | **Tipo de Dato** | **Descripción** |
| IP | Texto | Contiene la IP del alimentador en la red |
| Puerto | Texto | Contiene el puerto por el cual se conecta al alimentador |
| CodSeg | Texto | Contiene el código de seguridad necesario para poder mandar ordenes al alimentador |

**7-Manual de usuario**

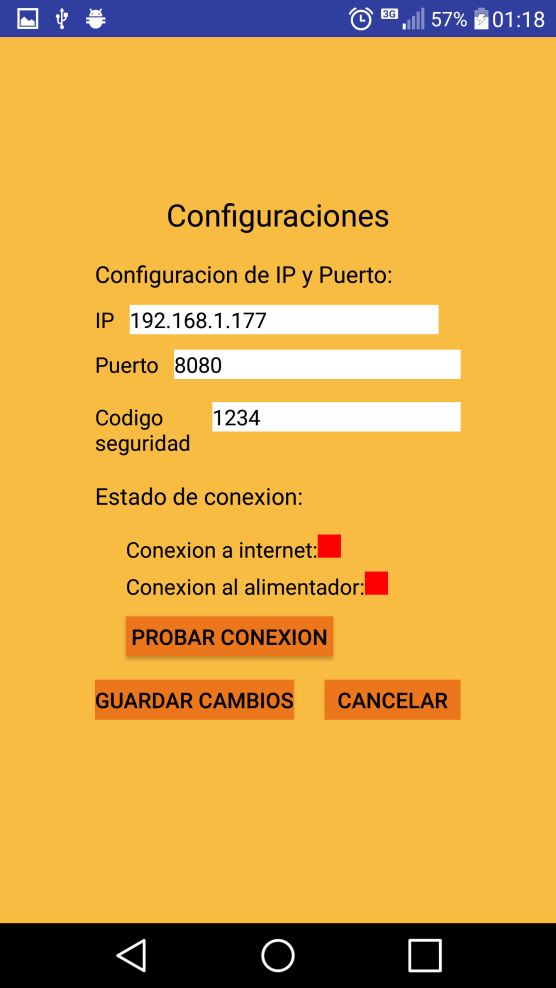
-Configurar la aplicación:

1-Se debe ir a la ventana de “Configuración”

2-Desde ahí chequea que los datos de IP,Puerto y Codigo de Seguridad son lo que están en la etiqueta de su alimentador. De ser incorrectos cambiarlos y presionar en el botón de “Guardar cambios”

3- Con el alimentador conectado pulse el botón “Probar conexión”, si la conexión se realizó correctamente se mostrara un mensaje de “alimentador conectado” en la pantalla.

Si ocurre un error chequear que el alimentador esté conectado correctamente y repita el paso 2 y 3



-Alimentar a la mascota:

1-Seleccione el botón “Alimentar” en el menú principal

2-Seleccione el tipo de mascota

3-Precione el botón “Aceptar”



Si ocurre un “error” chequee su conexión a internet y que la configuración sea la correcta y repita los pasos anteriores.

-Establecer una Alarma:

1-Debe ir a la ventana de “Alarma”

2-Activar la alarma si no está activada

3-Configurar una hora a la que se activara la alarma

4-Seleccionar el tipo de mascota que tiene

5-Seleccionar un intervalo, esto indicara cada cuantas horas se alimentara la mascota, por ejemplo si selecciona “cada 4hs” y establece la alarma a las 13hs, la alarma se activara a las 13hs, luego a las 17hs, luego a las 21hs y así seguidamente.

6-Precione el botón de “Aceptar”



Se le notificara que la alarma fue establecida mediante un mensaje en la pantalla

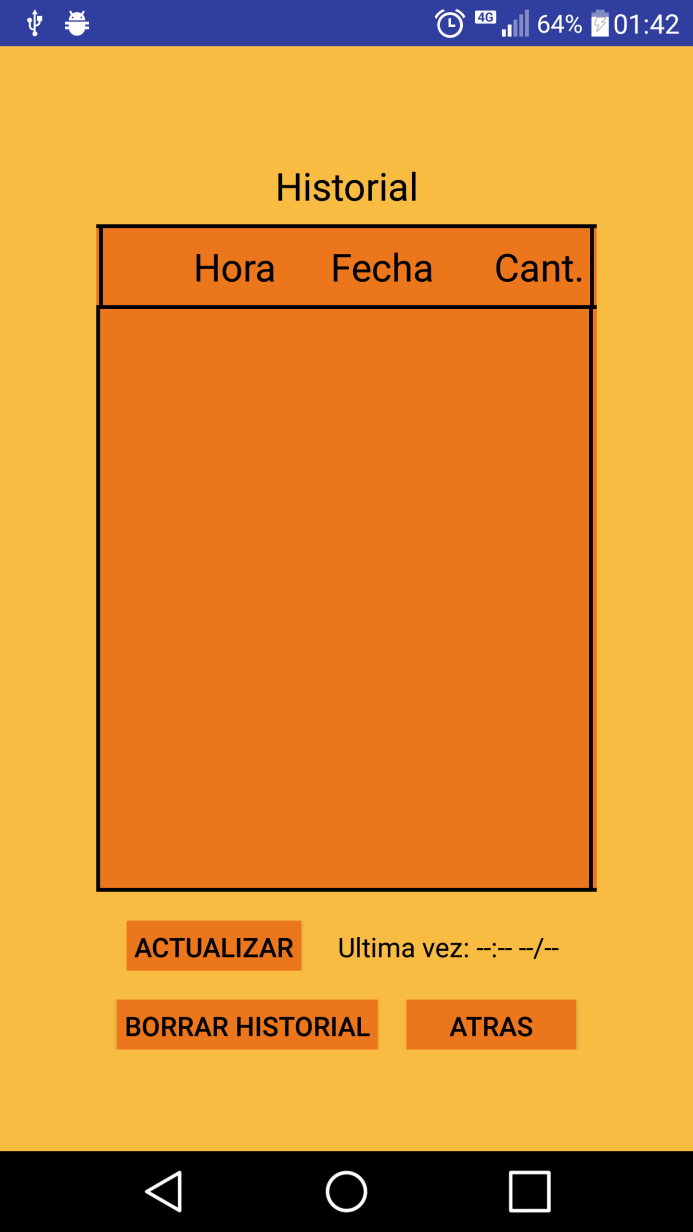
-Ver, borrar y actualizar el historial

A tener en cuenta: El historial no puede ser borrado ni actualizado sin conexión a internet, si no hay conexión se cargara los datos de la última actualización.

1-Para ver el historial diríjase a la ventana “Historial”

2-Una vez que entre se actualizara el historial automáticamente

3-Para borrar el historial presione en el botón “Borrar historial”



Si ocurre un “error” chequee su conexión a internet y que la configuración sea la correcta e inténtelo nuevamente.