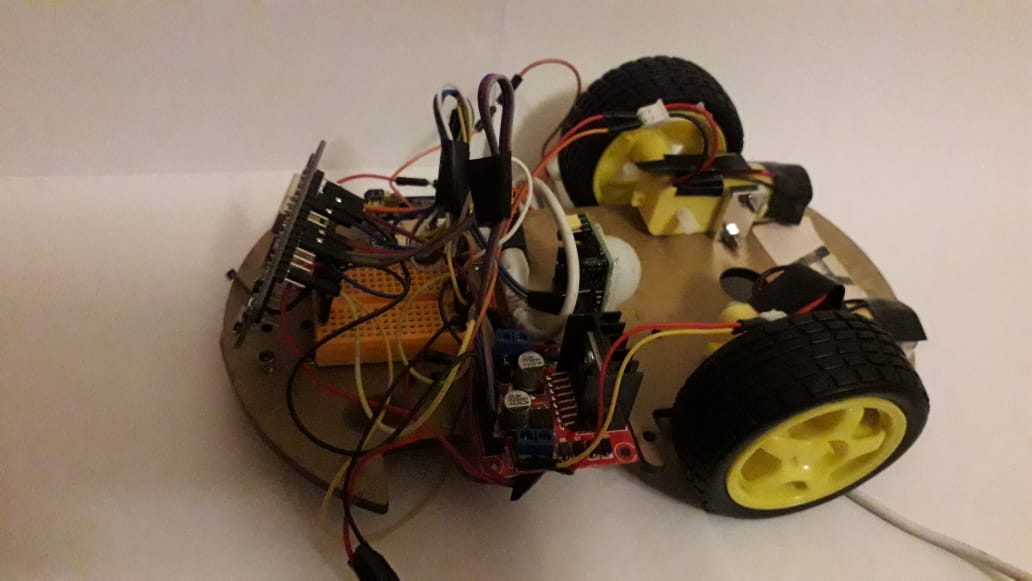
**MANUAL TÉCNICO DEL SISTEMA**

****

**DISPENSADOR DE PASTILLAS ROBÓTICO v1.0**

**AUTORES**

**Óscar Ávila**

**Andrés Guapi**

**Lenin Pardo**

**Steven Silva**

**Guayaquil, Ecuador**

**Enero, 2019**

Tabla de contenido

[INTRODUCCIÓN 3](#_Toc535879625)

[OBJETIVOS 3](#_Toc535879626)

[ACTORES DEL SISTEMA 3](#_Toc535879627)

[CASOS DE USO 4](#_Toc535879628)

[DISEÑO DE LA BASE DE DATOS 4](#_Toc535879629)

[DIAGRAMA ENTIDAD RELACIÓN 6](#_Toc535879630)

[CODIFICACIÓN DEL SISTEMA 6](#_Toc535879631)

[CÓDIGO DE CONTROL DE LA BASE MÓVIL 6](#_Toc535879632)

[DIAGRAMA DE LA BASE MÓVIL 8](#_Toc535879633)

[CÓDIGO DE LA ESTACIÓN 8](#_Toc535879634)

[DIAGRAMA DE LA ESTACIÓN 10](#_Toc535879635)

[CÓDIGO PYTHON DEL SERVIDOR 10](#_Toc535879636)

[CODIFICACIÓN DE LAS PANTALLAS 13](#_Toc535879637)

[CÓDIGO DE LA PANTALLA DE LOGUEO 13](#_Toc535879638)

[PANTALLA DE LOGUEO 14](#_Toc535879639)

[CÓDIGO DE FORMULARIO DE INGRESO DE ALARMA 14](#_Toc535879640)

[PANTALLA DE INGRESO DE ALARMA 16](#_Toc535879641)

[CÓDIGO DE LISTA DE ALARMAS 16](#_Toc535879642)

[PANTALLA DE LISTA DE ALARMAS 18](#_Toc535879643)

[CÓDIGO DE EDICION DE ALARMAS 18](#_Toc535879644)

[PANTALLA DE EDICIÓN DE ALARMAS 21](#_Toc535879645)

[AL USUARIO FINAL 22](#_Toc535879646)

# INTRODUCCIÓN

Muchas personas sufren de discapacidades paralizantes o paraplejias que les impiden movilizarse libremente, necesitando asistencia para la realización de actividades cotidianas como la toma de medicamentos. Este sistema de asistencia automatizado intenta resolver el problema mencionado mediante el uso de tres componentes básicos: una base móvil transportadora o carro transportador automatizado, encargado de llevar al paciente la medicina; una estación dispensadora que almacena la medicina y guarda la configuración del carro a la hora programada según las alarmas y una aplicación web que consiste en una interfaz de usuario para que se programen las alarmas de entrega de medicamentos.

La estación dispensadora es la que se encarga de controlar las horas a las que es necesario que la medicina sea proporcionada con la ayuda de un Arduino Mega y una Raspberry Pi 3. Esta última se encarga de modificar la base de datos y manejarlas, dándoles ordenes al Arduino para hacer el movimiento de motores necesarios. Mientras que un devkit ESP8266 es quien se encarga del transporte de las pastillas, con la ayuda de ciertos sensores, como son sensores PIR de movimiento y sensores IR.

Una vez que llega la hora del medicamento, el dispensador vierte las pastillas a la base móvil con un soporte para las pastillas. Donde el devkit ESP8266 se encarga de detectar el movimiento con la ayuda del sensor PIR para ponerse en marcha y hacer la función de un seguidor de línea, que, una vez entregadas las pastillas, regresa a su posición inicial.

# OBJETIVOS

Este manual contiene los detalles técnicos del sistema de medicación automatizada, para facilitar la modificación, actualización o mantenimiento del mismo con el fin de que analistas o programadores puedan leerlo e interpretarlo para los objetivos antes descritos.

**Objetivo general del sistema**

Este sistema está enfocado en mejorar la calidad de vida de las personas con paraplejia o discapacidad inmovilizante que requieran de medicación suministrada de manera puntual y periódica, permitiendo la automatización del proceso de entrega y la configuración de parámetros como tipo de medicamentos, dosis, frecuencia y fecha de inicio y fin.

**Objetivos específicos**

* Construir un sistema de entrega de medicamentos de manera automática y móvil hasta el lugar del paciente.
* Uso de base de datos para el manejo del horario de la medicación.
* Implementación de aplicación web para la configuración de las distintas entregas al paciente.

# ACTORES DEL SISTEMA

|  |  |
| --- | --- |
| ACTOR DEL NEGOCIO | DESCRIPCIÓN |
| Paciente | Es la persona objetivo del sistema, quien necesita medicación entregada de manera periódica de acuerdo a la receta entregada por un Médico. El paciente se encuentra en esta parapléjico por lo que es incapaz de moverse hacia la fuente de medicamentos (se encuentra en un lugar fijo donde se le suministra la medicina). |
| Auxiliar técnico | Encargado de ingresar la información de manejo de medicina para la base móvil, su presencia no es requerida en el entorno del paciente y puede realizar su trabajo de manera remota; recibe la información de parte de un médico. |
| Médico | El profesional que atiende al paciente y contacta al auxiliar técnico para que configure el sistema en base a su diagnóstico. No entra en contacto directo con el sistema. |

# CASOS DE USO

|  |  |
| --- | --- |
| Nombre del caso de uso | Configurar alarma |
| Actores | Auxiliar técnico, médico |
| Resumen | -El médico contacta al auxiliar técnico para que modifique las recetas del paciente de manera detallada.  -El auxiliar ingresa al sistema web se autentica y usa la opción editar alarma del sistema o crear una nueva dado el caso.  -Edita y guarda la información. |
| Precondiciones | -El sistema de gestión de bases de datos en la estación esta activo. |
| Poscondiciones | -Cambio de las tablas de la base de datos de la estación dispensadora. |
| Requisitos especiales | Ninguno. |

|  |  |
| --- | --- |
| Nombre del caso de uso | Entrega de medicamento |
| Actores | Paciente |
| Resumen | -El paciente recibe el medicamento de parte de la base móvil a determinadas horas.  -La base móvil regresa a su estación después de entregar los medicamentos. |
| Precondiciones | -La hora coincide con la almacenada en la alrma. |
| Poscondiciones | -Reducción de medicinas en la estación dispensadora. |
| Requisitos especiales | El camino seguido por la base móvil esta despejado sin obstáculos que franquear. |

# DISEÑO DE LA BASE DE DATOS

Para un funcionamiento eficiente se diseñaron las bases de datos que conforman el sistema con la aplicación MySQL debido a que esta aplicación nos permite un diseño amplio y concreto de las tablas y los campos que contiene la base de datos. La implementación de restricciones de seguridad y llaves foráneas entre otras cosas que incluye.

En primera instancia se muestran las tablas que conforman la base de datos.

|  |  |
| --- | --- |
| Medicamento | |
| #nombre\_med | Varchar(30) |
| \*dosis | Int(1) |
| O laboratorio | Varchar(30) |
| O tipo | Varchar(20) |

|  |  |
| --- | --- |
| Horario | |
| #id\_horario | Int(AI) |
| \*hora | Int(2) |
| \*minuto | Int(2) |
| \*periodicidad | Int(2) |
| F id\_alarma | int |

|  |  |
| --- | --- |
| Dia | |
| #id\_dia | Int(AI) |
| \*fecha\_inicio | Date |
| \*fecha\_fin | Date |
| F id\_alarma | Int |

|  |  |
| --- | --- |
| Alarma | |
| #id\_alarma | Int |
| \*Nombre | Varchar(30) |
| F nombre\_med | Varchar(30) |

Se describirá brevemente la función de cada una de estas tablas:

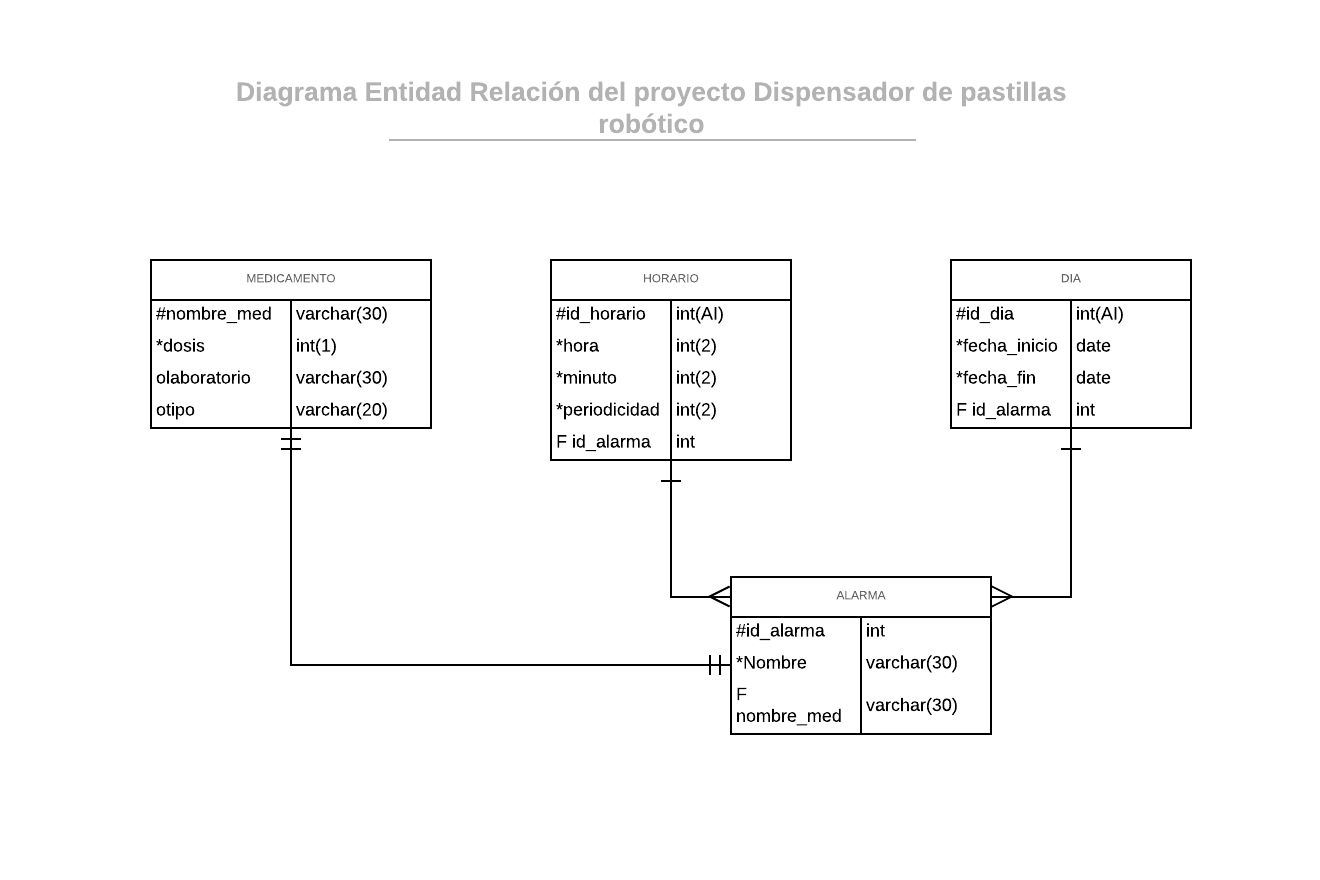
**Tabla Medicamento:** Su objetivo es listar el tipo y dosis del medicamento recetado al paciente.

**Tabla Horario:** Contiene la información de entrega de la medicina como tiempo entre dosis.

**Tabla Dia:** Registra el plazo o tiempo de duración de una alarma particular.

**Tabla Alarma:** Relaciona el medicamento y su información de entrega al paciente.

## DIAGRAMA ENTIDAD RELACIÓN



# CODIFICACIÓN DEL SISTEMA

La codificacion y depuracion se realizo en el IDE Pycharm mientras que las pruebas se ejecutaron en el prototipo.

## CÓDIGO DE CONTROL DE LA BASE MÓVIL

**import** machine

**import** time

#definicion de pines

#sensor PIR

sens**=**machine**.**Pin**(**13**,** machine**.**Pin**.**IN**)**

#sensores IR

ird**=**machine**.**Pin**(**5**,**machine**.**Pin**.**IN**,** machine**.**Pin**.**PULL\_UP**)**

iri**=**machine**.**Pin**(**4**,**machine**.**Pin**.**IN**,** machine**.**Pin**.**PULL\_UP**)**

#motores

#izquierdo

motai**=**machine**.**Pin**(**0**,**machine**.**Pin**.**OUT**)**

motad**=**machine**.**Pin**(**2**,**machine**.**Pin**.**OUT**)**

motbi**=**machine**.**Pin**(**14**,**machine**.**Pin**.**OUT**)**

motbd**=**machine**.**Pin**(**12**,**machine**.**Pin**.**OUT**)**

**def** derecha**():**

motai**.**value**(**0**)**

motad**.**value**(**0**)**

motbi**.**value**(**0**)**

motbd**.**value**(**1**)**

**def** izquierda**():**

motai**.**value**(**0**)**

motad**.**value**(**1**)**

motbi**.**value**(**0**)**

motbd**.**value**(**0**)**

**def** parar**():**

motai**.**value**(**0**)**

motad**.**value**(**0**)**

motbi**.**value**(**0**)**

motbd**.**value**(**0**)**

**def** enmarcha**():**

**print(**"en camino"**)**

time**.**sleep**(**5**)**

**while** iri**.**value**()==**0 **or** ird**.**value**()==**0**:**

**if** iri**.**value**():**

derecha**()**

**if** ird**.**value**():**

izquierda**()**

parar**()**

**print(**"esperando que se recojan las pastillas"**)**

time**.**sleep**(**5**)**

**while** **not** sens**.**value**():**

**pass**

**print(**"girando"**)**

motai**.**value**(**1**)**

motad**.**value**(**0**)**

motbi**.**value**(**0**)**

motbd**.**value**(**1**)**

time**.**sleep**(**0.5**)**

**print(**"regresando"**)**

**while** iri**.**value**()==**0 **or** ird**.**value**()==**0**:**

**if** iri**.**value**():**

derecha**()**

**if** ird**.**value**():**

izquierda**()**

parar**()**

time**.**sleep**(**3**)**

**print(**"llego a la estacion"**)**

motai**.**value**(**1**)**

motad**.**value**(**0**)**

motbi**.**value**(**0**)**

motbd**.**value**(**1**)**

time**.**sleep**(**1**)**

parar**()**

**while** **True:**

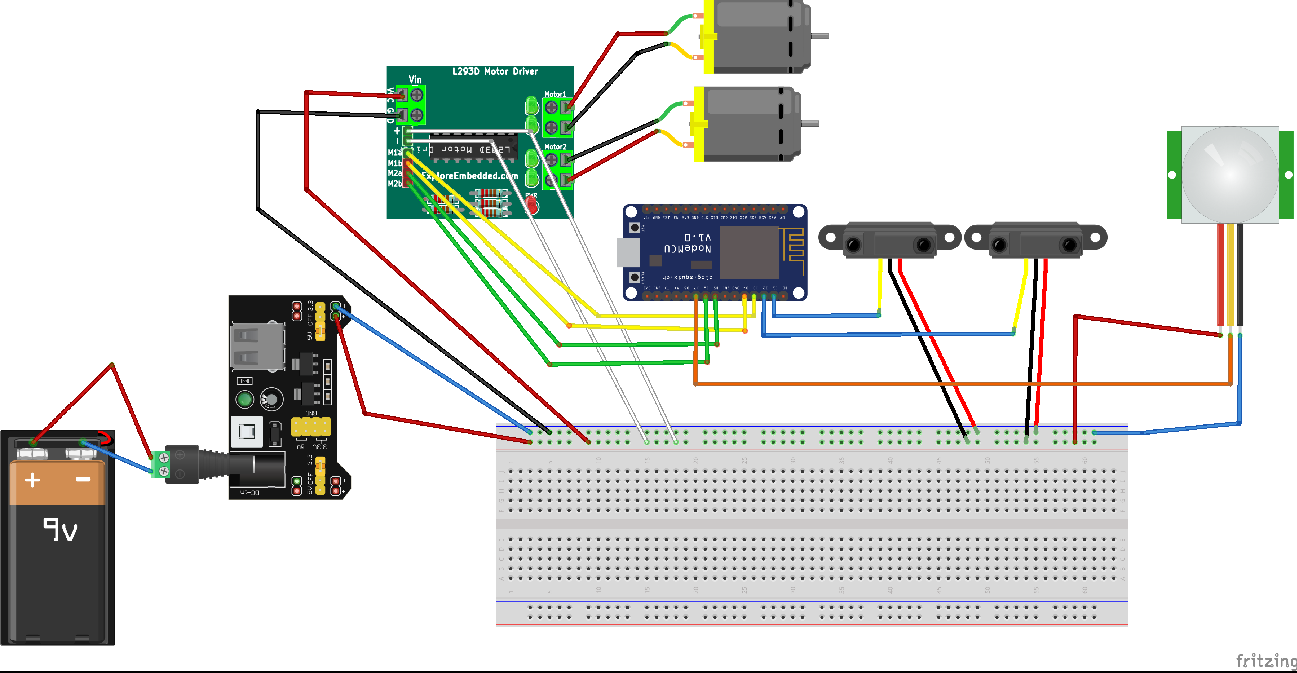
**print(**sens**.**value**())**

**if** sens**.**value**()==**1**:**

**print(**"ya"**)**

enmarcha**()**

## DIAGRAMA DE LA BASE MÓVIL

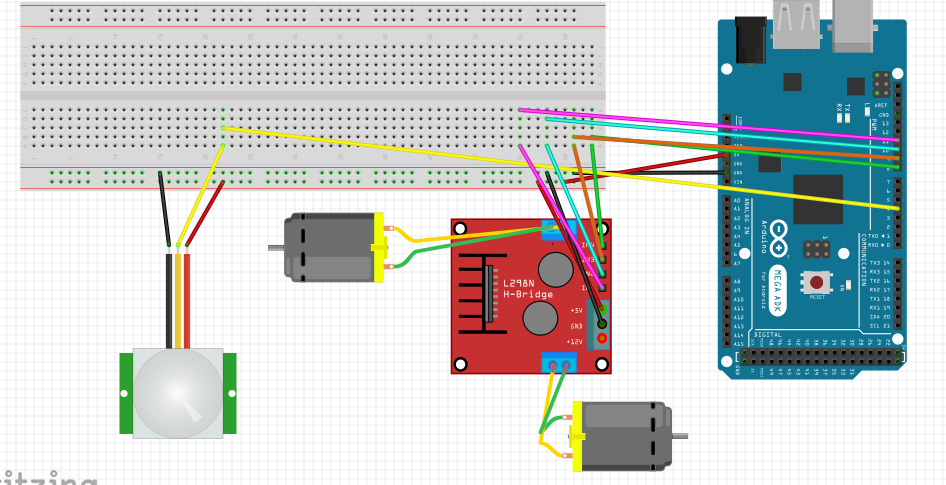


## CÓDIGO DE LA ESTACIÓN

1. int in1, in2, in3, in4, ENA, ENB, PIRPin;
2. String serIn;
3. in1 = 6;
4. in2 = 7;
5. in3 = 8;
6. in4 = 9;
7. ENA = 10;
8. ENB = 11;
9. PIRPin = 12;
11. void \_mForwardA()     //manera de mover hacia adelante el motor de las pastillas A
12. {
13. analogWrite(ENA,ABS);
14. //analogWrite(ENB,ABS);
15. digitalWrite(in1,HIGH);//digital output
16. digitalWrite(in2,LOW);
17. }
19. void \_mForwardB()     //manera de mover hacia adelante el motor de las pastillas B
20. {
21. //analogWrite(ENA,ABS);
22. analogWrite(ENB,ABS);
23. digitalWrite(in3,HIGH);//digital output
24. digitalWrite(in4,LOW);
25. }
27. void \_mBackA()      //manera de mover hacia atrás el motor de las pastillas A
28. {
29. analogWrite(ENA,ABS);
30. //analogWrite(ENB,ABS);
31. digitalWrite(in1,LOW);
32. digitalWrite(in2,HIGH);
33. }
35. void \_mBackB()      //manera de mover hacia atrás el motor de las pastillas B
36. {
37. //analogWrite(ENA,ABS);
38. analogWrite(ENB,ABS);
39. digitalWrite(in3,LOW);
40. digitalWrite(in4,HIGH);
41. }
43. void \_mStop()  //Método usado para detener los motores
44. {
45. digitalWrite(ENA,LOW);
46. digitalWrite(ENB,LOW);
47. }
49. void setup() {
50. // put your setup code here, to run once:
51. Serial.begin(9600);
52. while (Serial.available()>0){
53. serIn=Serial.read();
54. }
55. pinMode(in1,OUTPUT);
56. pinMode(in2,OUTPUT);
57. pinMode(in3, OUTPUT);
58. pinMode(in4, OUTPUT);
59. pinMode(ENA,OUTPUT);
60. pinMode(ENB,OUTPUT);
61. pinMode(PIRPin, INPUT);
62. \_mStop();
63. }

66. void loop() {
67. // put your main code here, to run repeatedly:
68. String nombre\_pastilla;
69. String dosis;
70. int val = digitalRead(PIRPin);
71. if (Serial.available()>0){
72. serIn = Serial.read();
73. nombre\_pastilla = getValue(serIn, ';', 0);
74. dosis = getValue(serIn, ';', 1);
75. if (pastilla == "A"){
76. while (val == HIGH){
77. \_mForwardA();
78. }
79. \_mStop();
80. }
81. else if (pastilla == "B"){
82. while (val == HIGH){
83. \_mForwardB();
84. }
85. \_mStop();
86. }
87. }
88. }

## DIAGRAMA DE LA ESTACIÓN



## CÓDIGO PYTHON DEL SERVIDOR

1. import MySQLdb as mysql
2. import json
3. import threading
4. import logging
5. import time
6. import serial
7. import datetime
8. from time import gmtime, strftime
10. def consultameds():  #en esta función se hará una consulta a la base de datos para generar un archivo JSON
11. meds={}
12. db=mysql.connect("localhost","oavila","tlm123","proyecto\_sistemas\_telematicos2.0")  #se realiza un enlace con la base de datos
13. cursor=db.cursor()
15. try:
16. cursor.execute("select \* from medicamento")
17. resultsmed=cursor.fetchall()
18. #print(resultsmed)
19. for row in resultsmed:
20. #print(row[0])
21. cursor.execute("select horario.hora, horario.minuto, horario.periodicidad from medicamento, horario, alarma where alarma.nombre\_med like '"+row[0]+"' and alarma.id\_alarma like horario.id\_alarma")
22. resultshorario=cursor.fetchall()
23. #print(resultshorario)
25. cursor.execute("select dia.fecha\_inicio, dia.fecha\_final from dia, alarma where alarma.nombre\_med like '"+row[0]+"' and alarma.id\_alarma like dia.id\_alarma")
26. resultsdia=cursor.fetchall()
27. meds[row[0]]=[]
28. meds[row[0]].append({
29. 'dosis': row[1],
30. 'initdate':resultsdia[0][0].day,
31. 'initmonth':resultsdia[0][0].month,
32. 'inityear':resultsdia[0][0].year,
33. 'lastdate':resultsdia[0][1].day,
34. 'lastmonth':resultsdia[0][1].month,
35. 'lastyear':resultsdia[0][1].year,
36. 'hora':resultshorario[0][0],
37. 'minuto':resultshorario[0][1],
38. 'periodicidad':resultshorario[0][2],
39. })  #diccionario donde se guardará la información contenida en la base de datos
40. except:
41. db.rollback()
43. db.close()
45. with open('meds.txt', 'w') as outfile:   #se crea el archivo meds.txt donde se guardará el diccionario anteriormente creado
46. json.dump(meds, outfile, indent=4)  #se crea un JSON con un formato
48. def enviar(cadena):  #metodo para enviar string al arduino
49. ser = serial.Serial("/dev/ttyUSB0",9600)  #se verifica por qué puerto de la raspberry se enviarán los datos
50. ser.flushInput()
51. ser.write(cadena)  #el metodo write es para enviar por el serial el parámetro que se encuentre dentro de los paréntesis
53. def leerJSON():  #método que lee el JSON y devuelve la cadena de datos que se enviarán al arduino
54. cadena=""
55. dosis = 0
56. with open("meds.txt") as json\_file:  #se abre el JSON
57. data = json.load(json\_file)
58. for i in data:  #se utiliza para recorrer ambos diccionarios encontrados en el JSON
59. cadena=i+cadena
60. for p in data[i]:  #se utiliza para recorrer cada key del diccionario
61. dosis = p["dosis"]
62. cadena = "@"+cadena+";"+str(dosis)  #se crea la cadena que se enviará al arduino
63. return cadena
64. cadena=""
65. dosis=0
67. def verificarHora(nombre\_med):  #comparar la hora de la raspi con la hora y minuto del JSON
68. date = datetime.datetime.now().strftime("%H:%M")  #se obtiene la hora y minuto del servidor
69. #print("hora de la raspi\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")
70. #print(date)
71. name = ""
72. hour = ""
73. minute = ""
74. lista\_date = date.split(":")
75. #print(lista\_date[0])
76. #print(lista\_date[1])
77. with open("meds.txt") as json\_file:  #se abre el JSON
78. data = json.load(json\_file)
79. for i in data:
80. name = i
81. #print("nombre de la medicina")
82. #print(name)
83. for p in data[i]:
84. hour = str(p["hora"])
85. minute = str(p["minuto"])
86. #print("hora y minuto de la medicina")
87. #print(hour)
88. #print(minute)
89. if (nombre\_med == name and hour == lista\_date[0] and minute == lista\_date[1]):  #se compara si la hora y minuto del servidor son iguales a los encontrados en la base de datos
90. return True  #retorna un verdadero en caso de que la condición se cumpla
91. name = ""
92. hour = ""
93. minute = ""
95. def cambiarHora(nombre\_med):   #cambiar la ultima hora en que el paciente se tomo la pastilla en la base de datos
96. db=mysql.connect("localhost","oavila","tlm123","proyecto\_sistemas\_telematicos2.0")  #se realiza conexión con la base de datos
97. cursor=db.cursor()
98. hora=datetime.datetime.now().hour
99. minutos=datetime.datetime.now().minute
100. sqlquery="update horario, alarma set horario.hora = "+str(hora)+ " where horario.id\_alarma like alarma.id\_alarma and alarma.nombre\_med like '"+nombre\_med+"'"
101. #se reemplaza la hora anterior de la base de datos
102. cursor.execute(sqlquery)
103. time.sleep(1)
104. sqlquery="update horario, alarma set horario.minuto = "+str(minutos)+ " where horario.id\_alarma like alarma.id\_alarma and alarma.nombre\_med like '"+nombre\_med+"'"
105. #se reemplazan los minutos anteriores de la base de datos
106. cursor.execute(sqlquery)
107. db.close()

110. while (True):  #while para que el programa se ejecute siempre dentro del servidor
111. consultameds()
112. listaMed = ["A","B"]
113. for i in listaMed:
114. if (verificarHora(i)==True):
115. print("entro-------------")
116. doc = leerJSON()
117. print(doc)
118. enviar(doc)
119. cambiarHora(i)

# CODIFICACIÓN DE LAS PANTALLAS

El diseño se llevó a cabo en NetBeans al igual que la codificación de las mismas, la cual se muestra en la siguiente sección:

## CÓDIGO DE LA PANTALLA DE LOGUEO

<!DOCTYPE HTML PUBLIC “-//W3C//DTD HTML 4.01 Transitional//EN">

<!--Pagina de logueo de usuario-->

<html>

<head>

<meta http-equiv=**"Content-Type"** content=**"text/html"**; charset=UTF-8″>

<link rel=**"stylesheet"** href=**"estilo.css"** media=**"all"**>

<title></title>

</head>

<body>

<form id=**"formacceso"** action=**"validar.php"** method=**"post"**>

<table align=**"center"** >

<caption>**Acceso al sistema**</caption>

<tr>

<td colspan=2>

<span>**Usuario:**</span><br/>

<input class= **"cajas"** type=**"text"** name=**"usuario"**>

</td>

</tr>

<tr>

<td colspan=2>

<span>**Clave:**</span><br/>

<input class=**"cajas"** type=**"password"** name=**"clave"**>

</td>

</tr>

<tr>

<td>

<br/>

<input type=**"submit"** class=**"boton"** value=**"ingresar"**>

</td>

<td>

<br/>

<input type=**"reset"** class=**"boton"** value=**"cancelar"**>

</td>

</tr>

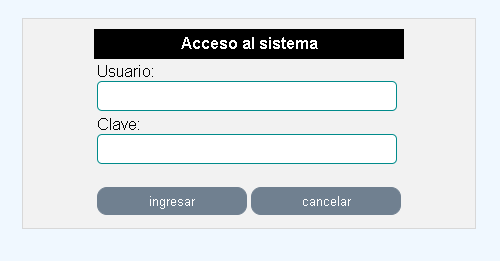
</table>

</form>

</body>

</html>

## PANTALLA DE LOGUEO



## CÓDIGO DE FORMULARIO DE INGRESO DE ALARMA

<html>

<!--Formulario de entrada de datos para nueva alarma-->

<head>

<link rel=**"stylesheet"** href=**"estilo.css"** media=**"all"**>

<div class=**"mensaje"**>**Bievenido**<h3><?php **echo** $nombreusuario ?></h3><a href=**"cerrar.php"**> **\ cerrar session**</a></div>

</head>

<body>

<br>

<form action=**"guardar.php"** method=**"post"**>

<table align=**"center"** style=**"border:1px solid black; padding: 15px;"**>

<caption>**Nueva Alarma**</caption>

<tr>

<td colspan=2>

<span>**Nombre:**</span><br/>

<input class= **"cajas"** type=**"text"** name=**"nombre"** required>

</td>

</tr>

<tr>

<td colspan=2>

<span>**Medicamento:**</span><br/>

<input class= **"cajas"** type=**"text"** name=**"medicamento"** required>

</td>

</tr>

<tr>

<td colspan=2>

<span>**Dosis:**</span><br/>

<input class=**"cajas"** type=**"number"** name=**"dosis"** required>

</td>

</tr>

<tr>

<td colspan=2>

<span>**Laboratorio:**</span><br/>

<input class=**"cajas"** type=**"text"** name=**"laboratorio"**>

</td>

</tr>

<tr>

<td colspan=2>

<span>**Tipo:**</span><br/>

<input class=**"cajas"** type=**"text"** name=**"tipo"**>

</td>

</tr>

<tr>

<td colspan=2>

<span>**Hora:**</span><br/>

<input class=**"cajas"** type=**"number"** name=**"hora"**>

</td>

</tr>

<tr>

<td colspan=2>

<span>**Minutos:**</span><br/>

<input class=**"cajas"** type=**"number"** name=**"minuto"**>

</td>

</tr>

<tr>

<td colspan=2>

<span>**Frecuencia:**</span><br/>

<input class=**"cajas"** type=**"number"** name=**"periodicidad"** required>

</td>

</tr>

<tr>

<td colspan=2>

<span>**Inicio:**</span><br/>

<input class=**"cajas"** type=**"date"** name=**"inicio"** required>

</td>

</tr>

<tr>

<td colspan=2>

<span>**Fin:**</span><br/>

<input class=**"cajas"** type=**"date"** name=**"fin"** required>

</td>

</tr>

<tr>

<td>

<br/>

<input type=**"submit"** class=**"boton"** value=**"Guardar"**>

</td>

<td>

<br/>

<input type=**"button"** class=**"boton"** onclick=**"location.href='alarmas.php'"** value=**"Ver alarmas"**>

</td>

</tr>

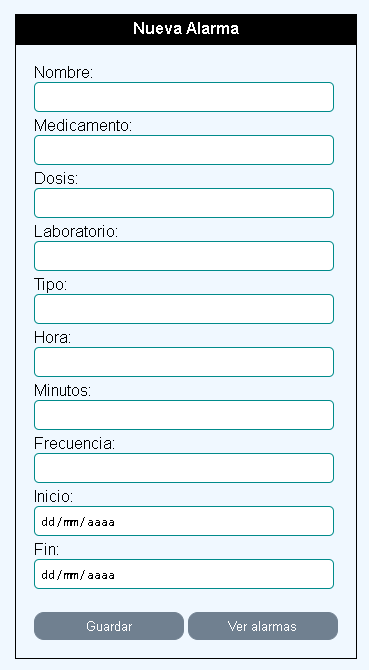
</table>

</form>

</body>

</html>

## PANTALLA DE INGRESO DE ALARMA



## CÓDIGO DE LISTA DE ALARMAS

<?php

//inicia o resume la sesion de usuario

**session\_start**();

//incluye el archivo con las credenciales para conectar a la base de datos

**include\_once** "conexion.php";

//consulta todas las alarmas disponibles registradas

$alarmas=**mysqli\_query**($con,"select al.id\_alarma as id,al.nombre as nombre,al.nombre\_med as medicina,med.dosis as dosis, dia.fecha\_inicio as inicio, dia.fecha\_final as fin, horario.periodicidad as frecuencia from alarma as al inner join medicamento as med on med.nombre\_med=al.nombre\_med inner join dia on dia.id\_alarma=al.id\_alarma inner join horario on horario.id\_alarma=al.id\_alarma;");

//cierra la conexion

**mysqli\_close**($con);

?>

<!--Panel de presentacion de las alarmas registradas-->

<html>

<head>

<link rel=**"stylesheet"** href=**"estilo2.css"** media=**"all"**>

<div class=**"mensaje"**><?php

//muestra mensaje de bienvenida de ususario y cierre de sesion

**echo** "Bievenido<h3>".$\_SESSION["usuario"].'</h3><a href="cerrar.php">cerrar session</a> <a href="panel.php">\volver al menu principal</a>'; ?></div>

</head>

<body>

<br>

<table align=**"center"** style=**"border:1px solid black; padding: 15px;"**>

<tr><td colspan=3 bgcolor=**"#000000"**><font color=**"white"**><center>**Alarmas guardadas**</center></font></td></tr>

<?php

//para cada alarma devuelta en la consulta se crea una entrada en la tabla de presentacion

**while**($row=**mysqli\_fetch\_array**($alarmas)) { ?>

<tr>

<td colspan=3>

<h3><?php

//muestra datos de alarma

**echo** '<center>'.$row["nombre"].'</center>';?></h3>

<?php **echo** $row["medicina"]." / ".$row["dosis"];?><br>

</td>

</tr>

<tr class=**"cajas"**>

<td colspan=2>

<?php

//muestra datos de alarma

**echo** "<b>Inicio:</b> ".$row["inicio"];?><br>

<?php **echo** "<b>Fin:</b> ".$row["fin"];?><br>

<?php **echo** "<b>Frecuencia:</b> ".$row["frecuencia"];?>

</td>

<td>

<?php

//crea botones de edicion y borrado para cada alarma

**echo** '<input type="button" class="boton2" onclick="location.href=\'accion.php?accion=editar&id='.$row["id"].'\'" value="Editar"><br>' ?>

<?php **echo** '<input type="button" class="boton2" onclick="location.href=\'accion.php?accion=borrar&id='.$row["id"].'\'" value="Borrar">' ?>

</td>

</tr><br>

<?php

}

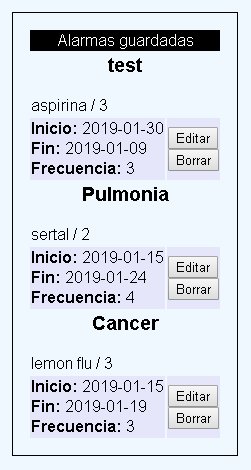
?>

</table>

</body>

</html>

## PANTALLA DE LISTA DE ALARMAS



## CÓDIGO DE EDICION DE ALARMAS

<?php

//inicia o resume la sesion de usuario

**session\_start**();

//incluye el archivo con las credenciales para conectar a la base de datos

**include\_once** "conexion.php";

//si el parametro accion pasado por url es borrar

**if**($\_GET["accion"]=='borrar'){

//borra los datos de asociados a la alarma con id pasado por url

**mysqli\_query**($con,"delete from horario where id\_alarma=".$\_GET["id"].";");

**mysqli\_query**($con,"delete from dia where id\_alarma=".$\_GET["id"].";");

**mysqli\_query**($con,"delete from medicamento where nombre\_med=(select nombre\_med from alarma where id\_alarma=".$\_GET["id"].");");

**mysqli\_query**($con,"delete from alarma where id\_alarma=".$\_GET["id"].";");

**mysqli\_close**($con);

//muestra alerta de alarma eliminada

**echo** "<script>alert('Alerta eliminada')</script>";

//redirecciona a la pagina de alarmas

**echo** "<script>location.href='alarmas.php';</script>";

}**else**{

//si la opcion elegida no es borrar muestra un formulario con toda la informacion de alarma para editar

$res=**mysqli\_fetch\_array**(**mysqli\_query**($con,"select al.nombre as nombre,al.nombre\_med as medicina,med.dosis as dosis,med.laboratorio as laboratorio,med.tipo as tipo,horario.minuto as minutos, horario.hora as hora, dia.fecha\_inicio as inicio, dia.fecha\_final as fin, horario.periodicidad as frecuencia from alarma as al inner join medicamento as med on med.nombre\_med=al.nombre\_med inner join dia on dia.id\_alarma=al.id\_alarma inner join horario on horario.id\_alarma=al.id\_alarma where al.id\_alarma=".$\_GET["id"].";"));

?>

<!--Formulario con informacion de alarma a editar-->

<html>

<head>

<link rel=**"stylesheet"** href=**"estilo.css"** media=**"all"**>

<div class=**"mensaje"**>**Bievenido**<h3><?php

//muestra mensaje de bienvenida y cierre de sesion

**echo** $\_SESSION["usuario"] ?></h3><a href=**"cerrar.php"**>**cerrar session**</a></div>

</head>

<body>

<br>

<form action=**"editar.php"** method=**"post"**>

<table align=**"center"** style=**"border:1px solid black; padding: 15px;"**>

<caption>**Editar alarma**</caption>

<tr>

<td colspan=2>

<?php

//envia id de alarma como parametro oculto

**echo** '<input type="hidden" name="id" value="'.$\_GET["id"].'" />';?>

<span>**Nombre:**</span><br/>

<?php

//muestra el nombre de alarma

**echo** '<input class= "cajas" type="text" name="nombre" value="'.$res["nombre"].'" required>';?>

</td>

</tr>

<tr>

<td colspan=2>

<span>**Medicamento:**</span><br/>

<?php

//muestra el tipo de medicamento

**echo** '<input class= "cajas" type="text" name="medicamento" value="'.$res["medicina"].'"required>';?>

</td>

</tr>

<tr>

<td colspan=2>

<span>**Dosis:**</span><br/>

<?php

//muestra la dosis de medicamento

**echo** '<input class="cajas" type="number" name="dosis" value="'.$res["dosis"].'" required>';?>

</td>

</tr>

<tr>

<td colspan=2>

<span>**Laboratorio:**</span><br/>

<?php

//muestra el nombre de laboratorio del medicamento

**echo** '<input class="cajas" type="text" value="'.$res["laboratorio"].'" name="laboratorio">';?>

</td>

</tr>

<tr>

<td colspan=2>

<span>**Tipo:**</span><br/>

<?php

//muestra el tipo de medicamento

**echo** '<input class="cajas" type="text" value="'.$res["tipo"].'" name="tipo">';?>

</td>

</tr>

<tr>

<td colspan=2>

<span>**Hora:**</span><br/>

<?php

//muestra la hora de paso para el medicamento

**echo** '<input class="cajas" type="number" value="'.$res["hora"].'" name="hora">';?>

</td>

</tr>

<tr>

<td colspan=2>

<span>**Minutos:**</span><br/>

<?php

//muestra los minutos de paso para los medicamentos

**echo** '<input class="cajas" type="number" value="'.$res["minutos"].'" name="minuto">';?>

</td>

</tr>

<tr>

<td colspan=2>

<span>**Frecuencia:**</span><br/>

<?php

//muestra la frecuencia de entrega de medicamento

**echo** '<input class="cajas" type="number" name="periodicidad" value="'.$res["frecuencia"].'" required>';?>

</td>

</tr>

<tr>

<td colspan=2>

<span>**Inicio:**</span><br/>

<?php

//muestra la fecha de inciio de entrega de medicamento

**echo** '<input class="cajas" type="date" name="inicio" value="'.$res["inicio"].'" required>';?>

</td>

</tr>

<tr>

<td colspan=2>

<span>**Fin:**</span><br/>

<?php

//muestra la fecha de fin de entrega de medicamento

**echo** '<input class="cajas" type="date" name="fin" value="'.$res["fin"].'" required>';?>

</td>

</tr>

<tr>

<td>

<br/>

<input type=**"submit"** class=**"boton"** value=**"Guardar"**>

</td>

<td>

<br/>

<input type=**"button"** class=**"boton"** onclick=**"location.href='alarmas.php'"** value=**"Cancelar"**>

</td>

</tr>

</table>

</form>

</body>

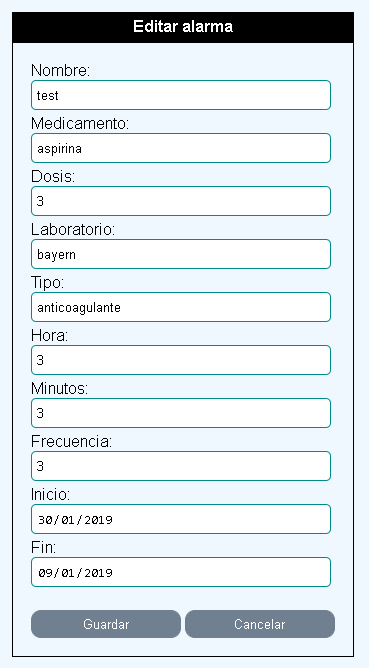
</html>

<?php

}

?>

## PANTALLA DE EDICIÓN DE ALARMAS



# AL USUARIO FINAL

En este manual técnico se describen los componentes básicos con el objetivo que se puedan leer, interpretar y analizar las partes con las que se conforma el sistema desarrollado a fin de que se le desee realizar modificaciones futuras o bien actualizaciones para mejorar su eficiencia y de ser posible sea base para algunos sistemas futuros a desarrollarse que sean afines a este.

En este manual se podrá encontrar información referente a:

Diseño de bases de datos

Vista de pantallas

Codificación de las pantallas

Actores y casos de uso

Dispensador de pastillas robótico por Oscar Ávila, Andrés Guapi, Lenin Pardo, Steven Silva se distribuye bajo una Licencia Creative Commons-Atribución-no-Comercial-Compartir Igual 4.0 Internacional.