



UNIVERSIDAD  
CATÓLICA  
DE CUENCA

**UNIVERSIDAD CATÓLICA DE CUENCA**

*Comunidad Educativa al Servicio del Pueblo*

**UNIDAD ACADÉMICA DE INFORMÁTICA,  
CIENCIAS DE LA COMPUTACIÓN E  
INNOVACIÓN TECNOLÓGICA.**

**CARRERA DE INGENIERÍA DE SISTEMAS**

**PROPUESTA DE DISEÑO DE UN PROTOTIPO PARA LA MEDICIÓN  
DE LA CALIDAD DE LA LECHE.**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL  
TÍTULO DE INGENIERO EN SISTEMAS.**

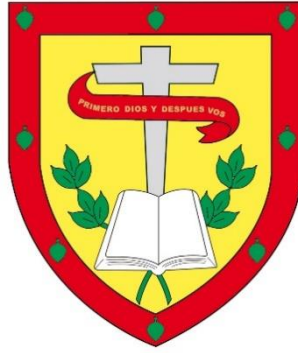
**AUTOR: CARLOS REINALDO MONTOYA PONCE.**

**DIRECTOR: ING. JOSÉ CARRILLO ZENTENO.**

**CAÑAR- ECUADOR**

**2022**

**DIOS, PATRIA, CULTURA Y DESARROLLO**



**UNIVERSIDAD CATÓLICA DE CUENCA**

*Comunidad Educativa al Servicio del Pueblo*

**UNIDAD ACADÉMICA DE INFORMÁTICA,  
CIENCIAS DE LA COMPUTACIÓN E INNOVACIÓN  
TECNOLOGICA.**

**CARRERA DE INGENIERÍA DE SISTEMAS**

**PROPUESTA DE DISEÑO DE UN PROTOTIPO PARA LA  
MEDICIÓN DE LA CALIDAD DE LA LECHE.**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL  
TÍTULO DE INGENIERO EN SISTEMAS.**

**AUTOR: CARLOS REINALDO MONTOYA PONCE.**

**DIRECTOR: ING. JOSÉ CARRILLO ZENTENO.**

**CAÑAR- ECUADOR**

**2022**

**DIOS, PATRIA, CULTURA Y DESARROLLO**



**Declaratoria de Autoría y Responsabilidad**

**Carlos Reinaldo Montoya Ponce** portador(a) de la cédula de ciudadanía N° **0302855093**. Declaro ser el autor de la obra: **“Propuesta de un Diseño de un prototipo para la medición de la calidad de la leche”**, sobre la cual me hago responsable sobre las opiniones, versiones e ideas expresadas. Declaro que la misma ha sido elaborada respetando los derechos de propiedad intelectual de terceros y eximo a la Universidad Católica de Cuenca sobre cualquier reclamación que pudiera existir al respecto. Declaro finalmente que mi obra ha sido realizada cumpliendo con todos los requisitos legales, éticos y bioéticos de investigación, que la misma no incumple con la normativa nacional e internacional en el área específica de investigación, sobre la que también me responsabilizo y eximo a la Universidad Católica de Cuenca de toda reclamación al respecto.

Cuenca, **24 de febrero de 2022**

F: .....

**Carlos Reinaldo Montoya Ponce**

**C.I. 0302855093**



## CERTIFICACIÓN

Certificó que el presente trabajo fue desarrollado por el Estudiante Carlos Reinaldo Montoya Ponce, bajo mi supervisión.

Ing. José Antonio Carrillo Zenteno

DIRECTOR DEL TRABAJO INVESTIGATIVO

UNIVERSIDAD CATÓLICA DE CUENCA.



## **DEDICATORIA**

El presente trabajo de titulación, lo dedico a mis padres quienes me enseñaron que el mejor conocimiento que se puede tener es el que se aprende por sí mismo. También a mis abuelitos por el apoyo incondicional, quienes me inculcaron grandes valores y me acompañaron en esta etapa, aportando a mi formación tanto profesional como ser humano.

## **AGRADECIMIENTO:**

Primeramente, agradezco a Dios por darme la capacidad y sabiduría para forjar mis sueños, a mis padres y abuelitos por el apoyo absoluto.

A esta prestigiosa casa de estudios y a sus catedráticos de la Unidad Académica de Tecnologías de la Información que día a día nos impartieron sus conocimientos, además de sus sanos consejos.

Al Ing. José Carrillo mi tutor que, bajo su tutela, con su apoyo y conocimientos bastos me instruyó incondicionalmente para la culminación de mi trabajo de titulación.

## DECLARACION

Yo, Carlos Reinaldo Montoya Ponce, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y que he consultado las referencias bibliográficas que se incluyen en este documento.

La Universidad Católica de Cuenca extensión Cañar puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y la Normativa actual de la institución.

---

Carlos Reinaldo Montoya Ponce

CI: 0302855093



## **RESPONSABILIDAD**

“La responsabilidad del contenido de esta tesis de grado, me corresponde exclusivamente; y el patrimonio intelectual de la misma a la Universidad Católica de Cuenca Extensión Cañar”.

---

Carlos Reinaldo Montoya Ponce

C.I: 030285509-3

## **APROBACIÓN DE TRIBUNAL DE GRADO**

El tribunal designado por el honorable consejo directivo de la Universidad Católica de Cuenca Extensión Cañar, Facultad de Ingeniería de Sistemas instalado para receptor la sustentación del trabajo final de investigación con el tema: “PROPUESTA DE DISEÑO DE UN PROTOTIPO PARA LA MEDICIÓN DE LA CALIDAD DE LA LECHE”, transcurrido el tiempo reglamentario procede a consignar la calificación de (\_\_\_\_\_/100).

Cañar, \_\_\_\_\_ de \_\_\_\_\_ del 2022

---

**PRESIDENTE**

---

**DIRECTOR**

---

**DELEGADO**

---

**SECRETARIO**



## INDICE DE CONTENIDO

DEDICATORIA .....	I
AGRADECIMIENTO: .....	II
CERTIFICACIÓN .....	III
DECLARACION .....	IV
RESPONSABILIDAD .....	V
APROBACIÓN DE TRIBUNAL DE GRADO .....	VI
INDICE DE CONTENIDO.....	7
Tabla de Ilustraciones.....	8
CAPITULO I .....	10
1 Manual de Usuario .....	10
1.1 Partes del Dispositivo .....	10
1.1.1 Descripción .....	11
1.1.2 Calibración .....	12
1.2 Uso del Dispositivo .....	13
1.3 Plataforma Web .....	14
1.3.1 Ingresar y Registrarse .....	15
1.3.2 Pantalla empresas .....	16
1.3.3 Pantalla clientes.....	17
1.3.4 Toma de datos .....	18
1.3.5 Datos históricos .....	19
Capitulo II .....	20
2 Manual del Programador .....	20
2.1 Programación .....	20
2.1.1 Desarrollo Web.....	20
2.1.2 Librerías y herramientas .....	21
2.1.3 Base de datos.....	21
2.1.4 Tablas .....	22
2.2 Controladores.....	25
2.2.1 Clase clientes .....	25
2.2.2 Clase general leche .....	27
2.2.3 Clase datos históricos. ....	28
2.2.4 Clase datos históricos 2 .....	30
2.2.5 Clase empresas .....	30
2.2.6 Clase Login .....	31



2.2.7	Clase Logout .....	32
2.2.8	Clase Principio .....	32
2.2.9	Clase Register .....	33
2.2.10	Clase Toma de Datos .....	33
2.2.11	Clase toma de datos2 .....	35
2.3	Models.....	35
2.3.1	Cientes model .....	35
2.3.2	Empresas model .....	35
2.3.3	General model cleche .....	36
2.3.4	Toma de datos model .....	37
2.4	Contents .....	37
2.4.1	Content clientes.....	37
2.4.2	Content datos históricos.....	37
2.4.3	Content empresas .....	38
2.4.4	Content toma de datos.....	38
2.5	Login.....	39
2.6	Register.....	39
2.7	Programación arduino .....	41
2.7.1	Implementación de la Placa Esp32 en el Ide Arduino .....	41
2.7.2	Librerías Utilizadas.....	43
2.7.3	Variables Arduino .....	44
3	ANEXOS .....	46
3.1.1	Anexo 1: Protocolo de Tesis.....	46
	Anexo 2: Código Fuente de la Plataforma .....	57
	Anexo 2: Código Fuente del Módulo Electrónico de Arduino .....	119
	Anexo 3: Código Fuente de la Base de Datos .....	127

## Tabla de Ilustraciones

ILUSTRACIÓN 1 PARTES DEL DISPOSITIVO. FUENTE: AUTOR .....	10
---	----



ILUSTRACIÓN 2 SENSOR PH. FUENTE: CERVANTES C, 2017, <a href="https://scidle.com/">HTTPS://SCIDLE.COM/</a> .....	12
ILUSTRACIÓN 3. SENSOR PH CON SU TAPA CONTENEDORA. FUENTE: CERVANTES C, 2017, <a href="https://scidle.com/">HTTPS://SCIDLE.COM/</a> .....	12
ILUSTRACIÓN 4. INSTRUMENTOS DE MEDICIÓN DEL DISPOSITIVO FUENTE: AUTOR.....	13
ILUSTRACIÓN 5 VISTA FRONTAL PROTOTIPO FUENTE: AUTOR.....	14
ILUSTRACIÓN 6. PANTALLA PRINCIPAL PLATAFORMA WEB FUENTE: AUTOR. ....	14
ILUSTRACIÓN 7. INICIAR SESIÓN Y REGISTRARSE PLATAFORMA WEB. FUENTE: AUTOR. ....	16
ILUSTRACIÓN 8. INICIAR SESIÓN Y REGISTRARSE PLATAFORMA WEB. FUENTE: AUTOR. ....	16
ILUSTRACIÓN 9. CLIENTES DE PLATAFORMA WEB. FUENTE: AUTOR. ....	17
ILUSTRACIÓN 10. CLIENTES PLATAFORMA WEB. FUENTE: AUTOR. ....	18
ILUSTRACIÓN 11. TOMAS DE DATOS PLATAFORMA WEB. FUENTE: AUTOR. ....	18
ILUSTRACIÓN 12. CONFIRMACIÓN DE ENTRADA DE DATOS PLATAFORMA WEB. FUENTE: AUTOR.....	19
ILUSTRACIÓN 13. CLIENTES PLATAFORMAS. WEB FUENTE: AUTOR. ....	19
ILUSTRACIÓN 14 PANEL DE CONTROL XAMPP. FUENTE: AUTOR. ....	20
ILUSTRACIÓN 15. MODELO RELACIONAL BASE DE DATO. FUENTE: AUTOR.....	21
ILUSTRACIÓN 16. TABLA USUARIO. FUENTE: AUTOR. ....	22
ILUSTRACIÓN 17. TABLA CLIENTE. FUENTE: AUTOR. ....	22
ILUSTRACIÓN 18. TABLAS TEMPORALES. FUENTE: AUTOR. ....	23
ILUSTRACIÓN 19. TABLA SENSORES PH. FUENTE: AUTOR. ....	23
ILUSTRACIÓN 20. TABLA SENSORES. FUENTE: AUTOR. ....	24
ILUSTRACIÓN 21. TABLA DE EMPRESAS. FUENTE: AUTOR. ....	24
ILUSTRACIÓN 22. TABLA DE CONTROL. FUENTE: AUTOR. ....	24
ILUSTRACIÓN 23. FUNCIÓN CONSTRUCTORA. FUENTE: AUTOR. ....	25
ILUSTRACIÓN 24. FUNCIÓN ÍNDEX. FUENTE: AUTOR. ....	26
ILUSTRACIÓN 25. RECIBE CLIENTES. FUENTE: AUTOR. ....	26
ILUSTRACIÓN 26. FUNCIÓN DATOS SENSORES. FUENTE: AUTOR. ....	27
ILUSTRACIÓN 28. CLASE DE DATOS HISTÓRICOS. FUENTE: AUTOR. ....	28
ILUSTRACIÓN 29. CLASE FILL CLIENTES. FUENTE: AUTOR.....	29
ILUSTRACIÓN 30. CLASE HISTÓRICOS 2. FUENTE: AUTOR. ....	30
ILUSTRACIÓN 31. CLASE EMPRESAS. FUENTE: AUTOR.....	30
ILUSTRACIÓN 32. FUNCIÓN DE RECIBIR EMPRESAS. FUENTE: AUTOR. ....	31
ILUSTRACIÓN 33. CLASE LOGIN. FUENTE: AUTOR. ....	31
ILUSTRACIÓN 34. CLASE LOGIN. FUENTE: AUTOR. ....	32
ILUSTRACIÓN 35. CLASE PRINCIPIO. FUENTE: AUTOR. ....	32
ILUSTRACIÓN 36. CLASE REGISTER. FUENTE: AUTOR. ....	33
ILUSTRACIÓN 37. CLASE TOMA DE DATOS. FUENTE: AUTOR.....	33
ILUSTRACIÓN 38. FUNCIÓN FILL CLIENTES. FUENTE: AUTOR.....	34
ILUSTRACIÓN 39. CLASE TOMA DE DATOS 2. FUENTE: AUTOR.....	35
ILUSTRACIÓN 40. PANTALLA ARDUINO IDE. FUENTE: AUTOR.....	41
ILUSTRACIÓN 41. MENÚ ARCHIVO ARDUINO. FUENTE: AUTOR.....	41
ILUSTRACIÓN 42. MENÚ PREFERENCIAS ARDUINO. FUENTE: AUTOR. ....	42
ILUSTRACIÓN 43. GESTOR DE TARJETAS DE ARDUINO. FUENTE: AUTOR. ....	42
ILUSTRACIÓN 44. MENÚ HERRAMIENTAS ARDUINO FUENTE: AUTOR.....	43
ILUSTRACIÓN 45. BÚSQUEDA DE LA PLACA ESP32 ARDUINO. FUENTE: AUTOR. ....	43
ILUSTRACIÓN 46. STRING DE DATOS PARA ENVÍO POR MÉTODO DE POST. FUENTE: AUTOR.....	45

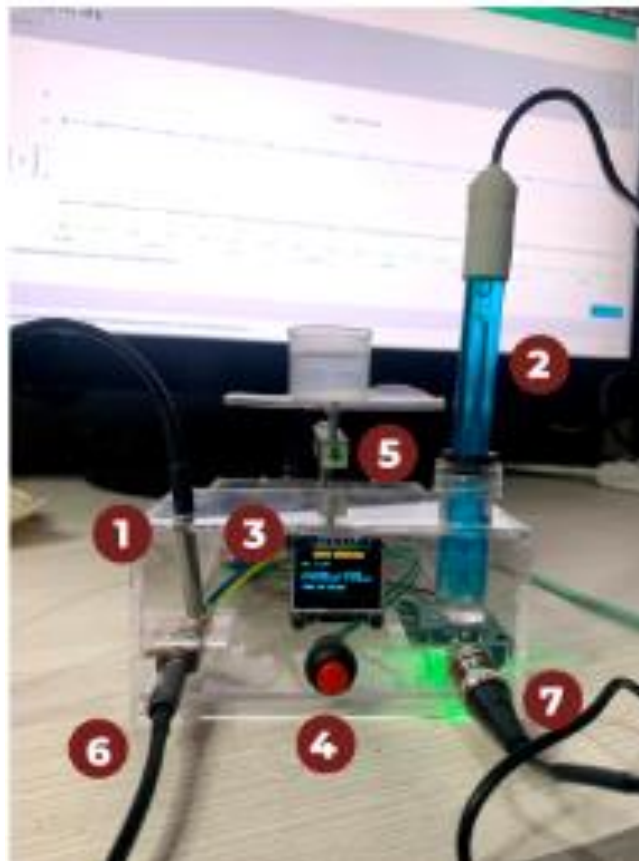
## CAPITULO I

### 1 Manual de Usuario

Este dispositivo se ha desarrollado con la finalidad de obtener un prototipo capaz de realizar una medición integral de la calidad de la leche, a través de la toma de datos para conocer su temperatura, densidad y nivel de pH.

Este prototipo está desarrollado en lenguaje C a través del IDE de arduino, además de php, mysql y javaScript, utilizando como framework a codeigniter.

#### 1.1 Partes del Dispositivo



*Ilustración 1 Partes del dispositivo. Fuente: Autor*

- 1.- Sensor de temperatura
- 2.- Sensor de Ph
- 3.- Pantalla Oled



- 4.- Botón Programación
- 5.- Balanza Digital
- 6.- Conector Sensor temperatura tipo Jack
- 7.- Conector Sensor Ph

### ***1.1.1 Descripción***

1. Sensor de temperatura: Sensor encargado de medir la temperatura en líquidos, con tolerancia de  $-55^{\circ}\text{C}$  y  $125^{\circ}\text{C}$ . Sensor auto calibrado.
2. Sensor de PH: Sensor encargado de medir el pH en líquidos, la escala de pH varía de 0 a 14. Sensor no auto calibrado.
3. Pantalla Oled: Pantalla para visualización de datos de recepción y envío.
4. Botón Programación: Botón para poner la placa en modo programación, (solo a ser manipulado por personal calificado).
5. Balanza Digital: Balanza digital, con copa de 12.5 ml para medición de la densidad en líquidos.
6. Conector Sensor temperatura tipo Jack.
7. Conector Sensor PH. Conector tipo sonda para sensor de ph.



*Ilustración 2 Sensor PH. Fuente: Cervantes C, 2017, <https://scidle.com/>*

### **1.1.2 Calibración**

El sensor está programado para tomar medidas desde 0 a 14. Sin embargo, podría surgir la necesidad de calibrar el sensor debido a factores ambientales. Por lo cual el sensor cuenta con 2 trimers de calibración, que facilitan la configuración de la precisión de ajuste que permite modificar la toma de datos de  $\pm 1.4$  unidades (Ilustración 2).

#### **NOTA**

Es importante, tener en cuenta que el sensor siempre que no esté en uso debe estar sumergido en algún líquido, contenido en la tapa (ilustración 3), para evitar el resecaimiento de la punta, lo cual puede desembocar en fallo en las lecturas del sensor.



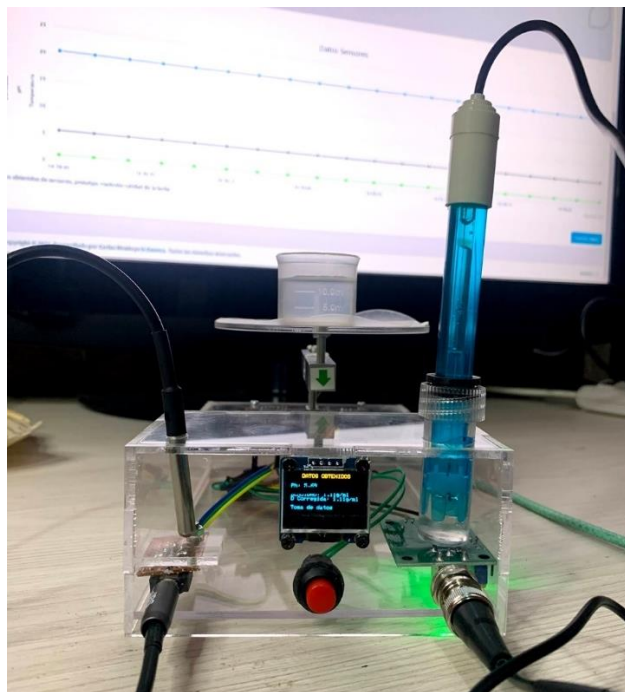
*Ilustración 3. Sensor pH con su tapa contenedora. Fuente: Cervantes C, 2017, <https://scidle.com/>*

## 1.2 Uso del Dispositivo



*Ilustración 4. Instrumentos de medición del dispositivo Fuente: Autor*

1. El dispositivo funciona con una fuente de alimentación de 5 voltios micro USB.
2. Conectar el sensor de temperatura a la entrada Jack 3.5m del dispositivo, como se muestra en la ilustración 1.
3. Conectar el sensor de ph a la entrada de sonda como se muestra en la ilustración 2.
4. Se incluye una copa de 12.5 ml, la cual se debe rellenar con el líquido a ser medido, como se muestra en la ilustración 3.



*Ilustración 5 Vista frontal prototipo Fuente: Autor*

1. Una vez conectados todos los sensores, el dispositivo mostrará a través de su pantalla el estado de operación, así como las mediciones realizadas.
2. El dispositivo, automáticamente comenzará a realizar el envío de datos a la plataforma web.

### 1.3 Plataforma Web



*Ilustración 6. Pantalla Principal plataforma Web Fuente: Autor.*

La plataforma Web cuenta con una interfaz de usuario, amigable que permite la recolección de datos, por empresas, y clientes. Todo esto gestionado por un usuario administrador. Que tiene acceso a los datos en la plataforma.

A continuación, se mostrarán las diferentes pantallas, y sus principales características.

### ***1.3.1 Ingresar y Registrarse***

1. Para ingresar a la plataforma en un servidor local, en la barra superior del navegador se debe escribir la palabra “localhost” lo cual se abrirá de forma automática la pantalla de iniciar sesión, en el cual existen dos campos para la autenticación del usuario ingresando las credenciales correspondientes en el caso de que ya tenerlas. caso contrario.se debe ingresar en el apartado registrarnos, lo cual abrirá la pantalla registrarse.
2. La pantalla de registro presenta un formulario, en el cual el usuario debe colocar los datos necesarios, y por último se presiona el botón de registro. La plataforma controla que no exista usuarios repetidos.
3. Luego de realizar el registro, consiguientemente en la pantalla de iniciar sesión de la plataforma, se procede a la autenticación y por ende a realizar la gestión de la misma.

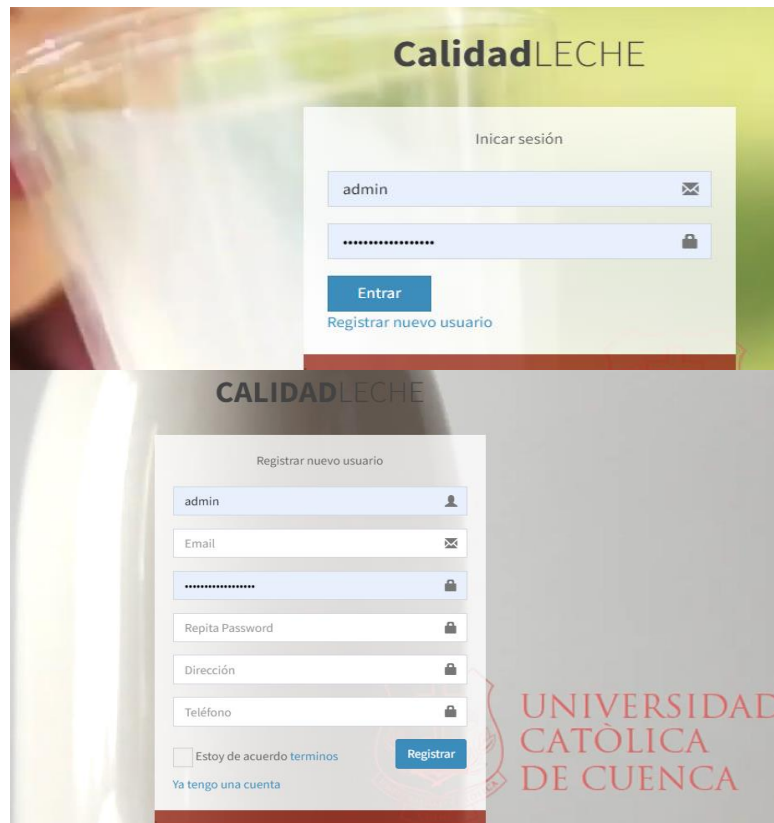


Ilustración 7. Iniciar Sesión y Registrarse plataforma web. Fuente: Autor.

### 1.3.2 Pantalla empresas

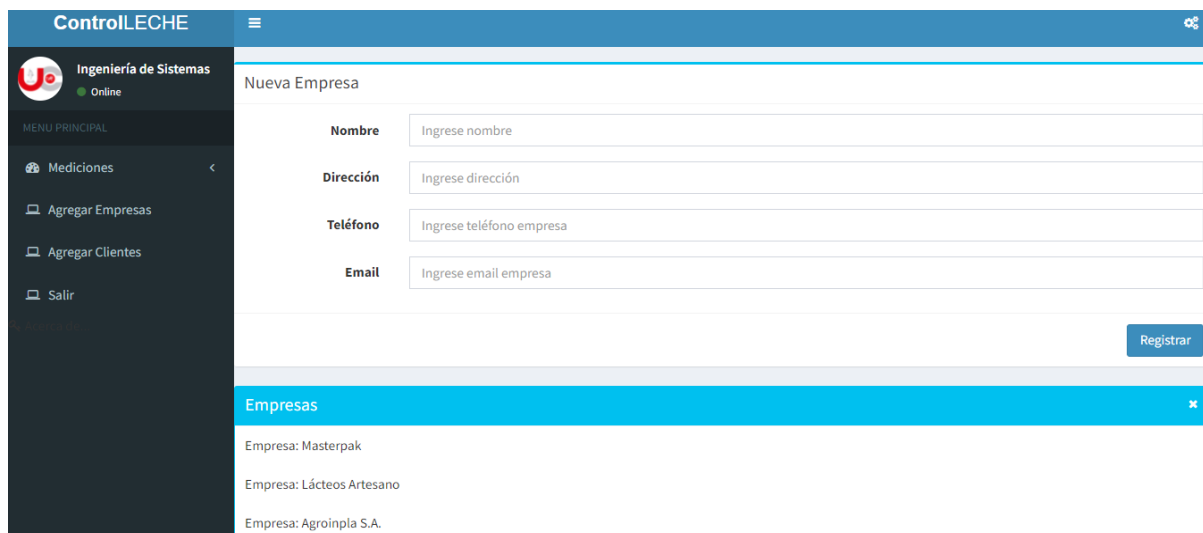


Ilustración 8. Pantalla de registro de una nueva empresa en la plataforma Web. Fuente: Autor.

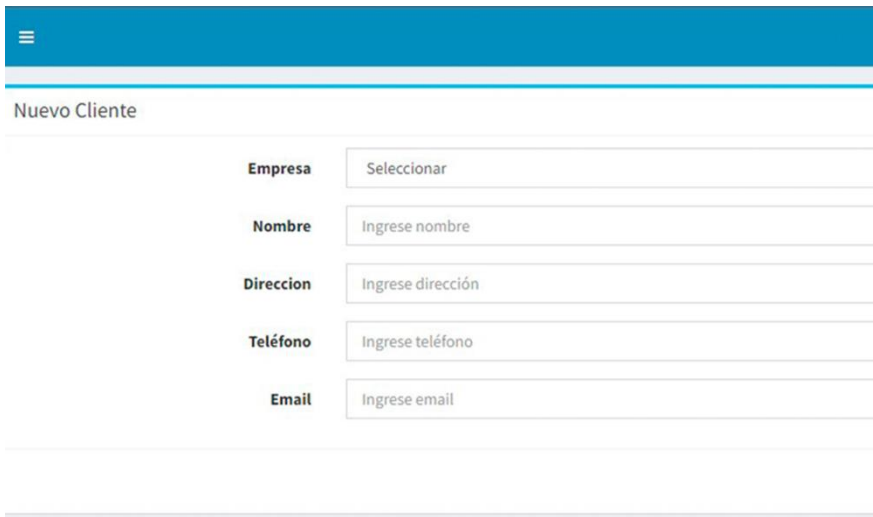
- a. Dado que se inicia sesión satisfactoriamente con el nombre de usuario y contraseña, se accede a la plataforma, la cual cuenta con un menú lateral en donde como primer paso, se encuentra la funcionalidad de agregar a las empresas, luego

se registra a los clientes con su respectiva empresa, que va a efectuar el control con la toma de datos.

- b. Por defecto la plataforma trae el campo “sin empresa” destinada para aquellos usuarios independientes que no pertenecen a una empresa formalizada.
- c. En la parte inferior la plataforma mostrará el listado de las empresas registradas.
- d. Los campos de formulario par registrar una nueva empresa cuentan con las validacion numérica y obligatoriedad en el llenado de todos los datos.

### ***1.3.3 Pantalla clientes***

Una vez ingresadas las empresas, se agrega los clientes pertenecientes a cada empresa, para la asignación de las empresas a sus respectivos representantes que este caso son los clientes, se debe seleccionar la empresa del listado desplegable (la plataforma no permite ingresar correos electrónicos repetidos, además todos los campos de formulario cliente son obligatorios).



*Ilustración 9. Clientes de plataforma Web. Fuente: Autor.*

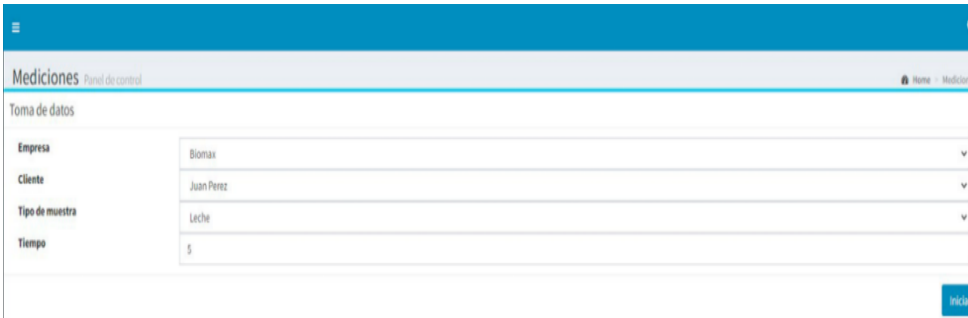
En la parte inferior, de la plataforma muestra el listado de los usuarios registrados.



Ilustración 10. Clientes Plataforma Web. Fuente: Autor.

### 1.3.4 Toma de datos

Ahora bien, con las empresas y clientes registrados previamente, se procede a realizar la toma de datos en tiempo real. Para ello, previamente encendido el dispositivo, únicamente se selecciona al cliente el cual va a realizar el control y asignar un tiempo de toma de datos en segundos.



Mediciones Panel de control

Toma de datos

Empresa	Biomax
Cliente	Juan Perez
Tipo de muestra	Leche
Tiempo	5

Iniciar

Ilustración 11. Tomas de datos plataforma Web. Fuente: Autor.

Una vez transcurrido este tiempo, se habilitará el botón para guardar los datos, a continuación para ratificar el proceso, se presiona en el botón guardar, en cual desplegará un pop-up confirmando que se han ingresado los datos.

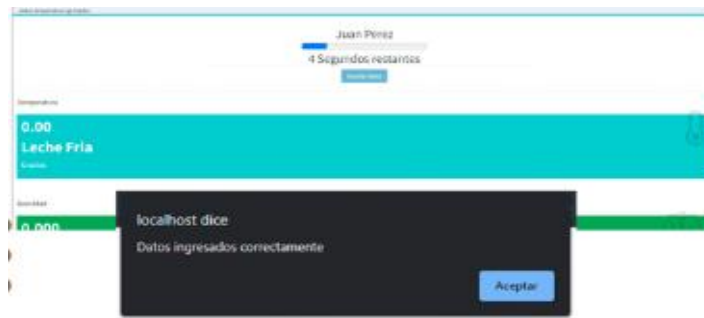


Ilustración 12. Confirmación de entrada de datos plataforma Web. Fuente: Autor.

### 1.3.5 Datos históricos

La plataforma web, contiene la funcionalidad de extraer los datos obtenidos en cada control, discriminando por Clientes y fechas. Para lo cual, en la pantalla Históricos, seleccionamos al cliente del cual se va a obtener los datos, así también la fecha en la cual fue realizado el control.



Ilustración 13. Pantalla Reporte de datos de plataforma Web representada en una grafica Fuente: Autor.

Una vez efectuada la consulta, en la plataforma se muestra una gráfica con el historial de las muestras con los indicadores de la parte lateral izquierda la escala de los sensores y en la parte inferior el tiempo, además se puede descargar el reporte en formato PDF y Excel.

## Capítulo II

### 2 Manual del Programador

La finalidad del presente manual del programador, es poner en conocimiento del lector los códigos fuente del proyecto desarrollado. Para ello se pretende explicar de manera concisa y rápida la programación utilizada en el desarrollo del software, esto con el fin de que el usuario pueda manejar a su disposición los parámetros y funciones que se encuentran expuestas en la programación. Todo lo anterior asumiendo que el presente manual está dirigido a personal capacitado y familiarizado con los diferentes software aquí empleados.

#### 2.1 Programación

El proyecto está dividido en 2 grandes partes, para el funcionamiento del mismo. Desarrollo web y Programación Arduino.

##### 2.1.1 Desarrollo Web

Encargado de realizar la conexión entre el hardware, con las vistas y la base de datos, en este caso el proyecto fue desarrollado con Xampp como servidor local, apoyado de Codeigniter 3.0 como framework, para el manejo de la arquitectura Modelo, Vista, Controlador.

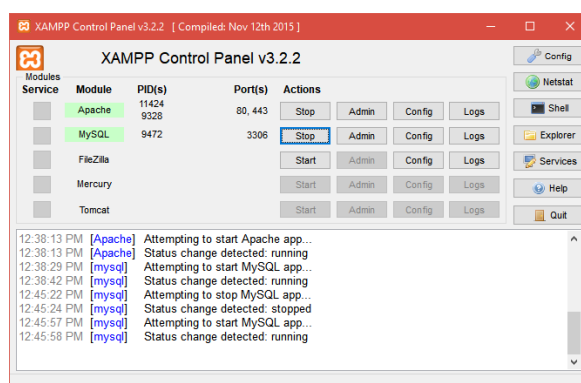


Ilustración 14 Panel de Control xampp. Fuente: Autor.

## 2.1.2 Librerías y herramientas

Para facilitar el desarrollo del proyecto fueron empleadas varias librerías, las cuales fueron:

- **Bootstrap:** Se menciona como una utilidad de desarrollo de aplicaciones web utilizado por el usuario.
- **Chart.js:** Es gratuita de código abierto necesario para el análisis de información de datos, que admite 8 tipos de gráficos:
- **Datatables.js:** Hace referencia a establecer tablas con paginado en búsquedas.
- **Highcharts:** Ayuda a la creación de gráficas.
- **Jquery:** permite agregar las actividades realizadas en un sitio web con bases del lenguaje.

## 2.1.3 Base de datos

En desarrollo de la base de datos fue realizado en Mysql, el cual viene incluido en el paquete xampp, para la cual se implementó un guía de base de datos relacional en cascada, con la siguiente estructura

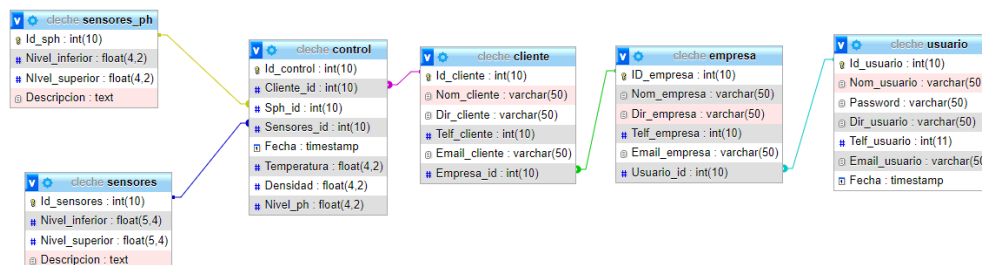


Ilustración 15. Modelo relacional base de dato. Fuente: Autor.

### 2.1.4 Tablas

**Usuario.** Contiene los datos de los usuarios administradores del sistema. Cada usuario, es independiente los unos de los otros, es decir, cada uno gestionará diferentes bases de datos de empresas y clientes.


#	Nombre	Tipo
<input type="checkbox"/> 1	<b>Id_usuario</b> 	int(10)
<input type="checkbox"/> 2	<b>Nom_usuario</b>	varchar(50)
<input type="checkbox"/> 3	<b>Password</b>	varchar(50)
<input type="checkbox"/> 4	<b>Dir_usuario</b>	varchar(50)
<input type="checkbox"/> 5	<b>Telf_usuario</b>	int(11)
<input type="checkbox"/> 6	<b>Email_usuario</b>	varchar(50)
<input type="checkbox"/> 7	<b>Fecha</b>	timestamp

Ilustración 16. Tabla usuario. Fuente: Autor.

**Cliente.** Contiene datos de los clientes registrados en el sistema, en relación con la tabla empresas, así también, su relación con los controles realizados para las tomas de muestras.



#	Nombre	Tipo
<input type="checkbox"/> 1	<b>Id_cliente</b> 	int(10)
<input type="checkbox"/> 2	<b>Nom_cliente</b>	varchar(50)
<input type="checkbox"/> 3	<b>Dir_cliente</b>	varchar(50)
<input type="checkbox"/> 4	<b>Telf_cliente</b>	int(10)
<input type="checkbox"/> 5	<b>Email_cliente</b>	varchar(50)
<input type="checkbox"/> 6	<b>Empresa_id</b> 	int(10)

Ilustración 17. Tabla Cliente. Fuente: Autor.

**Temporales.** Tabla encargada de recibir los datos provenientes del dispositivo electrónico, mantener dichos datos, hasta que le sean asignados un cliente y número de muestras. Es decir la tabla temporal captura los datos emitidos desde el módulo electrónico

arduino, cuando se presiona el botón iniciar para la toma de datos se guarda todos los datos relativos en la tabla temporales y luego que se confirma las muestras recogidas se selecciona en guardar, es en ese instante todos los datos acumulados se asigna al cliente y se guarda en la tabla control y se eliminan los datos de la tabla temporales.


#	Nombre	Tipo
1	temporales_id 	int(10)
2	temperatura	float(4,2)
3	densidad	float(4,3)
4	ph	float(3,1)
5	r_ph	varchar(50)
6	r_d	varchar(50)
7	d_corregida	float(4,3)

Ilustración 18. Tablas temporales. Fuente: Autor.

**Sensores ph.** Contiene las diferentes nomenclaturas y descripción que recibirá cada dato de acuerdo a su valor recibido del sensor de ph.


#	Nombre	Tipo
1	Id_sph 	int(10)
2	Nivel_inferior	float(4,2)
3	Nivel_superior	float(4,2)
4	Descripcion	text

Ilustración 19. Tabla sensores PH. Fuente: Autor.

**Sensores.** Contiene las diferentes nomenclaturas y descripción que recibirá cada dato de acuerdo a su valor recibido de los sensores de temperatura y densidad.

#	Nombre	Tipo
1	Id_sensores 	int(10)
2	Nivel_inferior	float(5,4)
3	Nivel_superior	float(5,4)
4	Descripcion	text

Ilustración 20. Tabla sensores. Fuente: Autor.

**Empresa.** Contiene los datos de las empresas registradas por el usuario, así también guarda relación con la tabla clientes, los cuales pertenecen a cada una de las empresas asignadas.

#	Nombre	Tipo
1	ID_empresa 	int(10)
2	Nom_empresa	varchar(50)
3	Dir_empresa	varchar(50)
4	Telf_empresa	int(10)
5	Email_empresa	varchar(50)
6	Usuario_id 	int(10)

Ilustración 21. Tabla de empresas. Fuente: Autor.

**Control.** Tabla encargada del almacenamiento de los datos obtenidos desde el hardware de medición y sensores.

#	Nombre	Tipo
<input type="checkbox"/>	1 Id_control 	int(10)
<input type="checkbox"/>	2 Cliente_id 	int(10)
<input type="checkbox"/>	3 Sph_id 	int(10)
<input type="checkbox"/>	4 Sensores_id 	int(10)
<input type="checkbox"/>	5 Fecha	timestamp
<input type="checkbox"/>	6 Temperatura	float(4,2)
<input type="checkbox"/>	7 Densidad	float(4,2)
<input type="checkbox"/>	8 Nivel_ph	float(4,2)

Ilustración 22. Tabla de Control. Fuente: Autor.

## 2.2 Controladores

Durante el desarrollo del proyecto se utilizó la estructura Modelo - Vista – Controlador, la cual proporciona una visión ordenada de cada una de las páginas que componen la plataforma.

A continuación, se detallarán los diferentes Scripts que estructuran el proyecto.

### 2.2.1 Clase clientes

Se encarga de la recepción de datos de los clientes. Se compone de 3 funciones, la función constructora, función índice (función por defecto) y función recibe clientes.

```
public function __construct()
{
    parent::__construct();
    $this->load->model('Clientes_model'); //carga el modelo

    $this->load->database(); //carga la
    $this->load->library('session'); //carga la librería

    if ($this->session->userdata('Id_usuario')=="){
        echo "Sin autorización";
        redirect(base_url('Login'), 'refresh');
    }
}
```

*Ilustración 23. Función Constructora. Fuente: Autor.*

**Función Constructora.** Está función es la encargada de cargar los modelos empleados en la página así también como cargar la conexión con la base de datos, y la librería de variables de sesión.

La función constructora está presente en cada uno de los Scripts de los controladores, cargando los parámetros antes mencionados, de acuerdo a los necesitados en cada script.

**Función Index.** Función encargada de cargar la información almacenada de clientes y empresas y mostrar las vistas de la página al usuario.

```

public function index()
{
    $usuario_id = $this->session->userdata('Id_usuario');
    $data['empresas'] = $this->Clientes_model->get_empresas($usuario_id);
    //recupero de variables de sesión el id del usuario para siempre mostrar o modificar
    $clientes = $this->Clientes_model->listar_clientes($usuario_id); //llamo a la func lis
    usuario en cuestión y NO LOS DE OTRO USUARIO. a la lista en forma de array la guardo e
    $data['clientes'] = $clientes; // a la lista $dispositivos la meto en un array llamado
    $papafrita['hamburguesa'] pero por lo general preparamos una variable con ese nombre,

    $this->load->view('head');
    $this->load->view('abre_body_wrapper');
    $this->load->view('header');
    $this->load->view('sidebar');
    $this->load->view('content_clientes',$data);
    $this->load->view('footer');

    $this->load->view('cierra_wrapper');
    $this->load->view('scripts');
}

```

Ilustración 24. Función Index. Fuente: Autor.

Las variables empleadas en la función index son las siguientes:

- \$usuario\_id: Variable encargada de cargar el id del usuario de las variables de sesión
- \$data['empresas']: Variable encargada del almacenamiento de las empresas obtenidas con la consulta realizada al modelo “Clientes\_model/get\_empresas”
- \$clientes: Variable encargada del almacenamiento de las empresas obtenidas con la consulta realizada al modelo “Clientes\_model/listar\_clientes”

**Función recibe clientes.** Función encargada de recibir la información de los formularios de datos para registro de clientes.

```

public function recibe_clientes()
{
    $usuario_id = $this->session->userdata('Id_usuario'); //recupero de variables de sesión
    $Empresa_id=$_POST['empresas'];
    $Nom_cliente = strip_tags($_POST['nombre']); //limpio y recibo por post el alias del r
    $Dir_cliente = strip_tags($_POST['direccion']); //limpio y recibo por post la serie de
    $Telf_cliente = strip_tags($_POST['telefono']); //limpio y recibo por post el alias de
    $Email_cliente = strip_tags($_POST['mail']); //limpio y recibo por post la serie del r

    $return = $this->Clientes_model->nuevo_cliente($Empresa_id,$Nom_cliente,$Dir_cliente,$

    $clientes = $this->Clientes_model->listar_clientes($usuario_id); //llamo a la func lis
    usuario en cuestión y NO LOS DE OTRO USUARIO. a la lista en forma de array la guardo e

    $data['clientes'] = $clientes; // a la lista $clientes la meto en un array llamado $da
    $papafrita['hamburguesa'] pero por lo general preparamos una variable con ese nombre,

    header('Content-Type: application/json; charset=utf-8');
    echo json_encode($return);
}

```

Ilustración 25. Recibe Clientes. Fuente: Autor.

Descripción de las variables utilizadas en la función recibe clientes:

- Variables que reciben por método post la información del formulario Cliente:  
 \$usuario\_id, \$Empresa\_id, \$Nom\_cliente, \$Telf\_cliente, \$Dir\_cliente,  
 \$Email\_cliente
- \$return: variable encargada de recibir la información si existe o no un cliente con la información ingresada
- \$data['clientes']: Variable encargada del almacenamiento de las empresas obtenidas con la consulta realizada al modelo “Clientes\_model/listar\_clientes”

### 2.2.2 Clase general leche

Controlador encargado de recibir los datos desde el hardware (Arduino) a través de método post. Este controlador posee las siguientes funciones.

```

public function recibe_datos_sensores()
{
    $controlIngreso = $this->toma_de_datos_model->estadoingresodatos();
    $int_value = intval( $controlIngreso );
    echo gettype($int_value);
    echo $int_value;

    if ($int_value == 1){
        $temperatura=$_POST['temperatura'];
        $densidad=$_POST['densidad'];
        $d_corregida=$_POST['d_corregida'];
        $ph=$_POST['ph'];
        $r_d=$_POST['r_d'];
        $r_ph=$_POST['r_ph'];

        $this->toma_de_datos_model->nuevo_dato_sensores($temperatura,$
            densidad,$ph,$r_d,$r_ph,$d_corregida);
    }
    else
    {
        echo 'sistema no recibe datos';
    }
}

```

*Ilustración 26. Función Datos Sensores. Fuente: Autor.*

En la función recibe datos sensores, la variable control ingreso permite la recepción o no de datos.

Variables recibidas por método post, desde el hardware de arduino son:

- \$temperatura=\$\_POST['temperatura'];

- \$densidad=\$\_POST['densidad'];
- \$d\_corregida=\$\_POST['d\_corregida'];
- \$ph=\$\_POST['ph'];
- \$r\_d=\$\_POST['r\_d'];
- \$r\_ph=\$\_POST['r\_ph'];

### 2.2.3 Clase datos históricos.

**Función Index.** Clase enfocada a la consulta de datos a través de la consulta por clientes y fechas.

```

public function index()
{
    $usuario_id = $this->session->userdata('Id_usuario');
    $id_cliente='';
    $datei='';
    $datef='';
    $data['empresas'] = $this->toma_de_datos_model->get_empresas($usuario_id
    );
    $data['cliente'] = $this->toma_de_datos_model->get_clientes($id_cliente)
    ;
    $data['temperatura'] = $this->General_model_cleche->
    datos_historicos_temperatura($id_cliente,$datei,$datef);
    $data['densidad'] = $this->General_model_cleche->
    datos_historicos_densidad($id_cliente,$datei,$datef);
    $data['ph'] = $this->General_model_cleche->datos_historicos_ph($
    id_cliente,$datei,$datef);
    $this->load->view('head');
    $this->load->view('abre_body_wrapper');
    $this->load->view('header');
    $this->load->view('sidebar');
    $this->load->view('content_datos_historicos',$data);
    $this->load->view('footer');

    $this->load->view('cierra_wrapper');
    $this->load->view('scripts');
}

```

*Ilustración 27. Clase de datos históricos. Fuente: Autor.*

\$usuario\_id: Variable encargada de almacenar el id de usuario de la librería de variables de sesión.

Variables Inicializadas con valores nulos.

- \$id\_cliente:
- \$datei:
- \$datef:

Consulta de variables que cargan datos del usuario y clientes.

- \$data['empresas']:
- \$data['cliente']:

Variables de datos históricos, que realizan consultas al modelo denominado “General model cleche”

- \$data['temperatura']:
- \$data['densidad']:
- \$data['ph']:

**Función fill clientes.** Función encargada de llenar un item menú, a través de una consulta al modelo “toma de datos model/ get clientes”.

```
public function fill_clientes() {  
    $empresas = $this->input->post('empresasi');  
    echo "Hola";  
    //die();  
    if(!is_null($empresas)){  
        $clientes = $this->toma_de_datos_model->get_clientes($empresas);  
        echo "<option selected>Seleccionar</option>";  
        foreach($clientes as $fila){  
            echo "<option value='".$fila->Id_cliente.'">". $fila->  
                Nom_cliente . "</option>";  
        }  
    } else {  
        echo "<option selected>Seleccionar8</option>";  
    }  
}
```

*Ilustración 28. Clase fill Clientes. Fuente: Autor.*

Variables usadas:

**\$empresas:** Variable que recibe el id de empresa por método post desde la vista empresas.

## 2.2.4 Clase datos históricos 2

```

public function index()
{
    $usuario_id = $this->session->userdata('Id_usuario');
    // $data['empresas'] =
    $this->toma_de_datos_model->get_empresas($usuario_id);
    $id_cliente = $_POST['clientes1'];
    $datei = $_POST['datei'];
    $datef = $_POST['datef'];
    // echo ($id_cliente);

    $data['temperatura'] = $this->General_model_cleche->
    datos_historicos_temperatura($id_cliente, $datei, $datef);

    $data['densidad'] = $this->General_model_cleche->
    datos_historicos_densidad($id_cliente, $datei, $datef);
    $data['ph'] = $this->General_model_cleche->datos_historicos_ph($
    id_cliente, $datei, $datef);
    $data['fecha'] = $this->General_model_cleche->datos_historicos_fecha($
    id_cliente, $datei, $datef);

    header('Content-Type: application/json; charset=utf-8');
    echo json_encode($data);
}
  
```

Ilustración 29. Clase Históricos 2. Fuente: Autor.

Carga los datos históricos una vez que se haya llenado los campos correspondientes en la vista “Content datos históricos”

## 2.2.5 Clase empresas

Clase encargada de la recepción de datos de las empresas a ser registrada por el usuario

**Función Index.** Encargada de cargar las vistas con los datos del usuario y las empresas registradas

```

public function index()
{
    $usuario_id = $this->session->userdata('Id_usuario'); //recupero de
    variables de sesión el id del usuario para siempre mostrar o modificar
    valores que pertenezcan a dicho usuario

    $empresas = $this->Empresas_model->listar_empresas($usuario_id); //
    llamo a la func listar dispositivos del modelo Dispositivos_model,
    pasándole el userid ya que la función solo debe devolver los dispos del
    usuario en cuestión y NO LOS DE OTRO USUARIO. a la lista en forma de
    array la guardo en la variable $dispositivos
    $data['empresas'] = $empresas; // a la lista $dispositivos la meto en
  
```

Ilustración 30. Clase empresas. Fuente: Autor.

**Función recibe empresas.** Función encargada de la recepción de datos a través de método post, desde el formulario en la vista “Content empresas”.

```

public function recibe_empresas()
{
    $usuario_id = $this->session->userdata('Id_usuario'); //recupero de
    variables de sesión el id del usuario para siempre mostrar o modificar
    valores que pertenezcan a dicho usuario
    $nombre = strip_tags($_POST['nombre']); //limpio y recibo por post el
    alias del nuevo dispositivo
    $direccion = strip_tags($_POST['direccion']); //limpio y recibo por
    post el alias del nuevo dispositivo
    $telefono = strip_tags($_POST['telefono']); //limpio y recibo por post
    el alias del nuevo dispositivo
    $email = strip_tags($_POST['mail']); //limpio y recibo por post el
    alias del nuevo dispositivo

    $return = $this->Empresas_model->nueva_empresa($usuario_id,$nombre,$
    direccion,$telefono,$email); //le envío a la función "nueva
    $empresas = $this->Empresas_model->listar_empresas($usuario_id); //
    llamo a la func listar empresas del modelo empresas_model, pasándole el
    userID ya que la función solo debe devolver los dispos del usuario en
    cuestión y NO LOS DE OTRO USUARIO. a la lista en forma de array la
    guardo en la variable $dispositivos

    $data['empresas'] = $empresas;

    header('Content-Type: application/json; charset=utf-8');
    echo json_encode($return);
}
}

```

Ilustración 31. Función de recibir empresas. Fuente: Autor.

Descripción de las variables utilizadas en la función recibe empresas

**\$return:** Variable encargada de confirmar que no exista empresas con datos repetidos.

**\$data ['empresas']:** variable a ser enviada por json a la vista “Content empresas” con los datos, de las empresas registradas.

## 2.2.6 Clase Login

```

public function index()
{
    $data['msg'] = ''; //esto se lo paso a la vista
    $this->load->view('login',$data); //cargo la vista login
}

public function hacerlogin(){
    $email = strip_tags($this->input->post('email')); //limpio y recibo
    Email
    $password = strip_tags(sha1($this->input->post('password'))); // limpio
    y recibo pass y la encripto al vuelo
    //echo $email; echo $password; //para debuggear
    $return = $this->Login_model->login($email, $password); //le paso al
    modelo ambas para que me diga si puedo dejarlo entrar

    if ($return==1) {
        redirect(base_url('inicio'), 'refresh'); //si
        está todo ok redirecciono a tiemporeal
    }

    if ($return==0) { //si hay algo mal, recargo el login
        pasandole a la vista el mensaje correspondiente
        $data['msg'] = 'Credenciales inválidas'; //esto se
        lo paso a la vista
        $this->load->view('login', $data);
    }
}
}

```

Ilustración 32. Clase Login. Fuente: Autor.

**Función Index.** Función encargada de cargar la vista Login al usuario con la variable \$data.

**Función Hacer Login.** Función encargada de iniciar sesión cargando las variables ingresadas en la vista Login

Variables ingresadas a través del formulario login y recibidas por método post:

- \$email
- \$password
- \$return

### 2.2.7 Clase Logout

Clase encargada de eliminar variables de sesión y mostrar la vista Login al usuario

```
class logout extends CI_Controller {  
    public function __construct()  
    {  
        parent::__construct();  
        session_destroy();  
    }  
    public function index()  
    {  
        header("login.php");  
        redirect(base_url('login'), 'refresh');  
    }  
}
```

Ilustración 33. Clase Login. Fuente: Autor.

### 2.2.8 Clase Principio

Clase encargada de cargar las vistas de la pantalla principal del proyecto.

```
public function index()  
{  
    $this->load->view('head');  
    $this->load->view('abre_body_wrapper');  
    $this->load->view('header');  
    $this->load->view('sidebar');  
    $this->load->view('content_inicio');  
    $this->load->view('footer');  
    $this->load->view('cierra_wrapper');  
    $this->load->view('scripts');  
}  
}
```

Ilustración 34. Clase principio. Fuente: Autor.

## 2.2.9 Clase Register

```
public function index()
{
    $data['msg'] = "";
    $this->load->view('register',$data); //carga la vista login
}

public function hacerregister(){

    $usuario = $_POST['usuario']; //recibo y limpio usuario
    $email = $_POST['email']; //recibo y limpio email
    $password = $_POST['password']; // recibo limpio pass
    $rpassword = $_POST['passwordr']; //recibo y limpio rpass
    $direccion = $_POST['direccion']; // recibo limpio direccion
    $telf=$_POST['telefono'];// recibo limpio telf

    $return= $this->Register_model->register($usuario,$email, $password,$
    direccion,$telf);
    header('Content-Type: application/json; charset=utf-8');
    echo json_encode($return);
}
}
```

Ilustración 35. Clase Register. Fuente: Autor.

**Función hacer register.** Función encargada de recibir las variables a través de método post desde la vista Register, con las siguientes variables.

- \$usuario = \$\_POST['usuario']; //recibo y limpio usuario
- \$email = \$\_POST['email']; //recibo y limpio email
- \$password = \$\_POST['password']; // recibo limpio pass
- \$rpassword = \$\_POST['passwordr']; //recibo y limpio rpass
- \$direccion = \$\_POST['direccion']; // recibo limpio direccion
- \$telf=\$\_POST['telefono'];// recibo limpio telf
- \$return: Variable encargada de recibir la respuesta del método para verificar si existe el usuario con la misma información ingresada.

## 2.2.10 Clase Toma de Datos

**Función index.** Función encargada de cargar las vistas “toma de datos” al usuario, así como las variables de usuario y clientes.

```
public function index() {
    $usuario_id = $this->session->userdata('Id_usuario');
    $this->toma_de_datos_model->cierraingresodatos();
    $id_cliente='';
    $data['empresas'] = $this->toma_de_datos_model->get_empresas($usuario_id
    );
    $data['cliente'] = $this->toma_de_datos_model->get_clientes($id_cliente)
    ;
}
```

Ilustración 36. Clase toma de datos. Fuente: Autor.

VARIABLES USADAS:

- **\$usuario\_id:** Variable encargada de cargar el Id de usuario desde la librería sesión.
- **\$id\_cliente:** Variable inicializada en null que posteriormente será llenada.
- **\$data ['empresas']:** Variable encargada de pasar la información de empresas registradas a la vista “Content Toma de datos”.
- **\$data ['cliente']:** Variable encargada de pasar la información de clientes registrados a la vista “Content Toma de datos”.

```
public function fill_clientes() {  
    $empresas = $this->input->post('empresas1');  
  
    //die();  
    if(!is_null($empresas)){  
        $clientes = $this->toma_de_datos_model->get_clientes($empresas);  
        echo '<option selected>Seleccionar</option>';  
        foreach($clientes as $fila){  
            echo '<option value="'. $fila->Id_cliente .'". $fila->  
                Nom_cliente .'</option>';  
        }  
    } else {  
        echo '<option selected>Seleccionar</option>';  
    }  
}
```

Ilustración 37. Función fill Clientes. Fuente: Autor.

**Función fill clientes.** \$empresas: Variable encargada de recibir el id de empresa ingresada por el usuario desde el “content toma de datos”.

### 2.2.11 Clase toma de datos2

```
public function index() {
    // POST Data

    $usuario_id = $this->session->userdata('Id_usuario');
    $this->toma_de_datos_model->abreingresodatos();

    $id_empresa = $_POST['empresas'];
    $id_cliente = $_POST['clientes'];
    $tipo_liquido=$_POST['liquidosi'];
    $temporizador=$_POST['tiempo'];

    //$data['empresas'] =
    $this->toma_de_datos_model->get_empresas($usuario_id);
    $Nombre = $this->toma_de_datos_model->get_nombre_clientes($id_cliente);
    foreach ($Nombre as $cli){
        $nom=$cli->Id_cliente;
    }

    $_SESSION["id_clientesd"]=$id_cliente;
}
```

Ilustración 38. Clase toma de datos 2. Fuente: Autor.

Clase encargada de leer los datos ingresados por el usuario y empezar la lectura de los sensores desde el hardware del arduino.

## 2.3 Models

### 2.3.1 Clientes model

Dentro del cliente model contiene las siguientes funciones:

- **Nuevo cliente:** Se delega en obtener datos del controlador cliente y realizar la verificación que no exista redundancia de registros, en caso no existir registros repetidos ingresará los nuevos datos.
- **Listar clientes:** Se basa en consultar registros de la base de datos para recuperar los registros de los clientes almacenados y pasarlo a pantalla.
- **Get empresas:** Se faculta de recuperar los registros de las empresas almacenadas y pasarlas a un listbox.

### 2.3.2 Empresas model

Funciones que tiene del modelo de empresas

- **Nueva empresa:** Se encomienda de recoger información del controlador empresas y realizar la verificación que no exista redundancia de registros en la tabla empresa de la base de datos. De no existir registros repetidos ingresará los nuevos datos.
- **Listar empresas:** Se encomienda en realizar consultas para recuperar los registros de las empresas almacenadas y pasarlas a pantalla.

### **2.3.3 General model cleche**

Funciones del model cleche

Últimos valores. Función encargada de recoger los últimos valores almacenados en la tabla denominada temporales, la cual recibe los datos de los sensores conectados al arduino. Y posteriormente pasarlos al controlador “toma de datos” el cual los mostrará en tiempo real.

Funciones encargadas de extraer los datos de la tabla “control” consecuente a las credenciales de empresa, cliente y fecha, ingresada por el usuario.

- Datos históricos densidad
- Datos históricos ph
- Datos históricos temperatura
- Datos históricos fecha

**Función login.** Función encargada de recibir los datos ingresados desde la vista “login”, compararlos con los registros almacenados en la base de datos, y permitir el ingreso del usuario a la plataforma.

**Función register.** Función encargada de recibir los datos incrustados por el controlador “register”, compararlos con los registros almacenados en la base de datos y permitir el ingreso del nuevo usuario en el caso de que no exista redundancia en los registros.

### 2.3.4 *Toma de datos model*

Entre los estados para el ingreso de datos se tiene:

**Estado ingreso datos:** Función encargada de devolver 1 o 0 para el control de ingreso de datos

- **Nuevo dato sensores:** Función encargada de ingresar los datos a la tabla temporales, una vez que desde la vista “toma de datos” se haya ingresado las credenciales del cliente, del cual se realizará el control de calidad de la leche.
- **Abre ingreso datos:** Por medio de la variable de control interpreta que el estado es el indicado para el ingreso de datos, para aceptar que la plataforma habilite la recepción de datos.
- **Cierra ingreso datos:** Esta función descifra el estado de la variable de control para el ingreso de datos, para que la plataforma inhabilite la recepción de los mismos.

## 2.4 Contents

### 2.4.1 *Content clientes*

#### Librerías

- bower\_components/jquery/dist/jquery.min.js
- bower\_components/jquery-ui/jquery-ui.js

#### Métodos

- Post
- Ajax

### 2.4.2 *Content datos históricos*

#### Librerías

- bower\_components/jquery/dist/jquery.min.js



- [bower\\_components/jquery-ui/jquery-ui.js](#)
- <https://code.highcharts.com/highcharts.js>
- <https://code.highcharts.com/modules/series-label.js>
- <https://code.highcharts.com/modules/exporting.js>
- <https://code.highcharts.com/modules/export-data.js>
- <https://code.highcharts.com/modules/accessibility.js>

### **Métodos**

- Post
- Ajax

### **2.4.3 *Content empresas***

#### **Librerías**

- [plugins/iCheck/ichack.min.js](#)

### **Métodos**

- Post
- Ajax

### **2.4.4 *Content toma de datos***

#### **Librerías**

- [bower\\_components/jquery/dist/jquery.min.js](#)
- [bower\\_components/bootstrap-toggle/css/bootstrap-toggle.min.css](#)
- [bower\\_components/bootstrap-toggle/js/bootstrap-toggle.min.js](#)
- [bower\\_components/highcharts/Highcharts-9.2.2/code/highcharts.js](#)
- [bower\\_components/highcharts/Highcharts-9.2.2/modules/exporting.js](#)



- `bower_components/highcharts/Highcharts-9.2.2/modules/export-data.js`
- `bower_components/highcharts/Highcharts-9.2.2/modules/accessibility.js`

## Métodos

- Post
- Ajax

## 2.5 Login

### Librerías

`bower_components/bootstrap/dist/css/bootstrap.min.css`

`<!-- Font Awesome -->`

`bower_components/font-awesome/css/font-awesome.min.css`

`<!-- Ionicons -->`

`bower_components/Ionicons/css/ionicons.min.css`

`<!-- Theme style -->`

`dist/css/AdminLTE.min.css`

`plugins/iCheck/square/blue.css`

### Métodos

- Post

## 2.6 Register

### Librerías

- `<!-- Bootstrap 3.3.7 -->`

`bower_components/bootstrap/dist/css/bootstrap.min.css`



- <!-- Font Awesome -->

bower\_components/font-awesome/css/font-awesome.min.css

- <!-- Ionicons -->

bower\_components/Ionicons/css/ionicons.min.css

- <!-- Theme style -->

dist/css/AdminLTE.min.css

- <!-- iCheck -->

plugins/iCheck/square/blue.css

- <!-- Google Font -->

[https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,600,700,300it](https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,600,700,300italic,400italic,600italic)

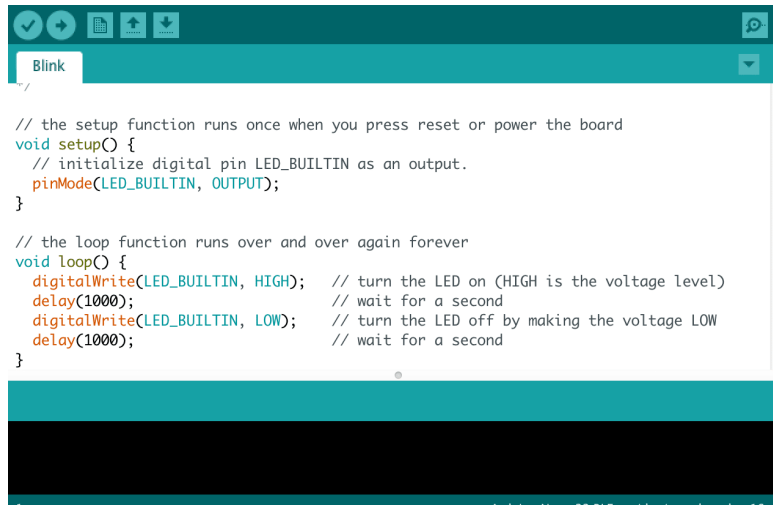
[alic,400italic,600italic](#)

#### **2.6.1.1 Métodos**

- Post

## 2.7 Programación arduino

Encargado del manejo y recolección de datos desde el hardware y sus diferentes módulos de desarrollo.



```
// Blink
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
```

Ilustración 39. Pantalla arduino IDE. Fuente: Autor.

### 2.7.1 Implementación de la Placa Esp32 en el Ide Arduino

Paso 1. Adicionar las URLs para placas ESP32

Se programa en arduino desde las URLs mediante las placas ESP32.

Lo primero es ejecutar Arduino IDE y se accede a preferencias.

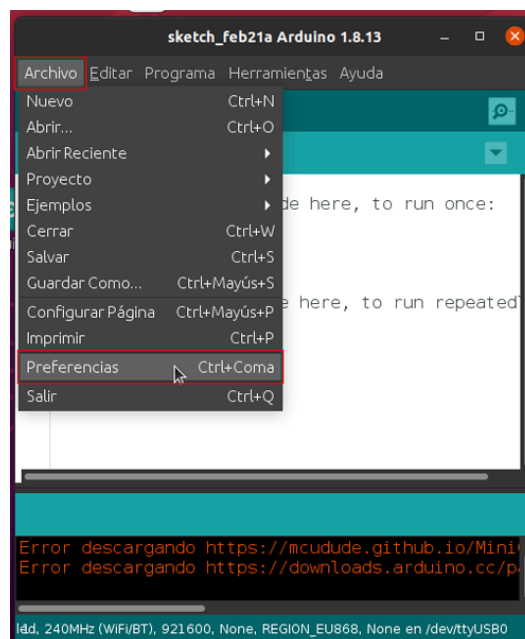


Ilustración 40. Menú archivo arduino. Fuente: Autor.

Se debe dirigir al “Gestor de tarjetas adicionales”.

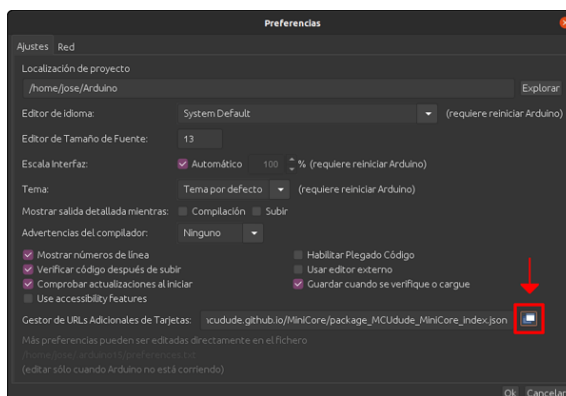


Ilustración 41. Menú preferencias arduino. Fuente: Autor.

En la nueva interfaz se debe agregar las URLs:

[https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json):

Con esta dirección el gestor de placas tendrá acceso a un conjunto elevado de placas y módulos ESP32 de varios fabricantes.

[https://resource.heltec.cn/download/package\\_heltec\\_esp32\\_index.json](https://resource.heltec.cn/download/package_heltec_esp32_index.json):

Con esta otra el gestor de placas tendrá acceso a las placas de desarrollo ESP32 comercializadas por Heltec.

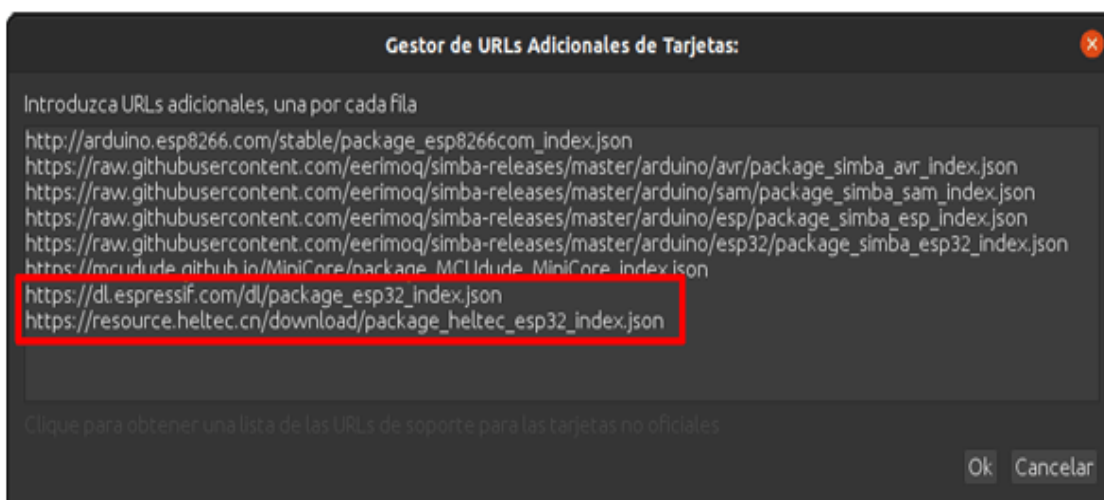


Ilustración 42. Gestor de tarjetas de arduino. Fuente: Autor.

Una vez agregadas las URLs es importante dar en OK para así poder visualizar las preferencias, en el paso 2 se debe instalar y configurar la placa en Gestor de Tarjetas

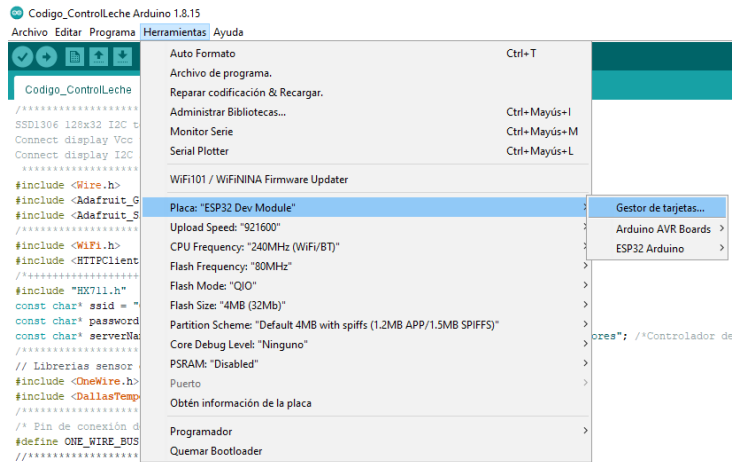


Ilustración 43. Menú herramientas arduino Fuente: Autor.

Se abre en el gestor de placas al actualizar la base de datos y para escribir se debe seleccionar “esp32” en la búsqueda disponible.



Ilustración 44. Búsqueda de la placa esp32 arduino. Fuente: Autor.

Para poder instalar el paquete se debe utilizar la primera opción mencionada evitando pérdidas de tiempo hasta que se descargue los archivos seleccionados en ESP32.

### 2.7.2 Librerías Utilizadas

**#include <Wire.h>:** Encargado de habilitar la conexión entre sensores y placa controladora.

**#include <Adafruit\_GFX.h>:** Librería para el manejo del sensor de pH

**#include <Adafruit\_SSD1306.h>:** Librería para el manejo de la pantalla Oled

**#include <ESP8266WiFi.h>:** Librería para habilitar el uso del wifi de la placa esp328266

**#include <ESP8266HTTPClient.h>**: Librería para habilitar el modo de cliente http en la placa controladora

**#include "HX711.h"**: Librería para manejo de la placa codificadora para la medición de densidad.

**#include <OneWire.h>**: Librería para el manejo del sensor de temperatura

**#include <DallasTemperature.h>**: Librería para el manejo del sensor de temperatura

### 2.7.3 *Variables Arduino*

**ssid** = Variable en la cual se almacena el nombre de la red a la cual va estar conectado el módulo electrónico arduino y el dispositivo que gestione la plataforma.

**password** = Clave de la red wifi

**serverName** = Nombre y/o dirección del servidor al cual serán enviados los datos por método Post, en este caso es

"http://192.168.0.103/Controlador\_general\_cleche/recibe\_datos\_sensores".

**calibration\_value**: Variable de calibración para el sensor de pH

**phval**: Variable inicial de lectura de pH, medida entre 0 y 4095

**avgval**: Variable que almacena un promedio de 10 lecturas de pH para posteriormente sacar el promedio de dichas lecturas.

**Volt**: Variable que almacena el voltaje entregado por el sensor de pH.

**ph\_act**: Valor que almacena el valor de pH utilizando la variable phval y avgval

**temperatura**: Lectura directa de temperatura en líquidos.

**gramos**: Lectura directa del peso en gramos obtenida de la balanza.

**Densidad**: Densidad de un líquido basado en 12.5 ml y su peso en gramos

**d\_corregida:** Densidad corregida en base a la temperatura.

**Datos:** Cadena tipo String a ser enviada por método Post al servidor, con la siguiente información.

```
String datos = "temperatura=";  
datos += 15;  
datos += "&densidad=";  
datos += 1;  
datos += "&d_corregida=";  
datos += 1.5;  
datos += "&ph=";  
datos += 7;  
datos += "&r_ph=";  
datos += r_ph;  
datos += "&r_d=";  
datos += r_d;
```

*Ilustración 45. String de datos para envío por método de Post. Fuente: Autor.*

### 3 ANEXOS

#### 3.1.1 Anexo 1: Protocolo de Tesis

##### A. TITULO

Propuesta de un diseño de un prototipo para la medición de la calidad de la leche

##### B. DOMINIO, LINEA Y ÁMBITOS DE INVESTIGACIÓN

<b>Energía eléctrica y tecnologías de la información para la innovación y el desarrollo sostenible</b>	<b>Ciencia de los ordenadores, Analítica de datos y Algoritmos computacionales</b>	a. Analítica de Datos	
		b. Ingeniería de Software	X
		c. Algoritmos computacionales	
		d. Inteligencia de negocios	
		e. Gobierno de Ti	
		f. Auditoria y seguridad informática	
		g. Simulación	

##### C. PLANTEAMIENTO DEL PROBLEMA

De acuerdo con el Reglamento 605/2010 de la Comisión Europea Se han formulado requisitos higiénicos para la producción de leche cruda, preparación y venta de subproductos; además, debido a la mayor toxicidad y resistencia de ciertos microorganismos, se han establecido condiciones higiénicas y de salud animal y requisitos de certificación veterinaria. El propósito es integrar crudos Leche y productoslácteos para consumo humano Introducidos en la Unión Europea. [1]

En Ecuador ya contamos con leyes y reglamentos para el control de la calidad de la leche, la Resolución No. 401 en la "Especificación de Calidad de la Leche Agrocalidad"y la norma INEN 9: 2012 para leche cruda, que establecen parámetros para diferentes ingredientes y calidad. Cuento la variedad de bacterias utilizando leche de vaca como materia prima. [2]

Actualmente en nuestro medio en la parte de la producción lechera, es regulado por medio de Agrocalidad, institución que centra su objetivo para el control y calidad del producto lácteo, las comunidades aledañas al cantón Cañar, se caracterizan por ser zonas potencialmente agrícolas y ganaderas, no contienen una técnica de medición adecuada Para controlar la calidad de la leche. Por ende, existe la necesidad de implementar un Sistema automatizado que brinde mayores posibilidades de garantizar un mejor flujo de



datos.

Por otra parte, ayuda a la gestión de control y calidad del producto lácteo, por otra parte, también el problema reside en que existen sistemas electrónicos de precios muy elevados, de los cuales las grandes ganaderías pueden acceder con una lechería automatizada a largo plazo en cambio los productores pequeños se les hace imposible, adquirir sistemas sofisticados.

#### **E. OBJETIVOS ESPECÍFICOS**

1. Realizar un estudio de la problemática y técnicas del control de la calidad de la leche
2. Diseñar un módulo electrónico que obtenga la medición de los parámetros de la calidad de la leche
3. Presentar el prototipo para la medición de la calidad de la leche

#### **F. JUSTIFICACION**

La producción lechera es una actividad que ofrece desarrollo y una fuente de ingresos para la sociedad, para este tiempo la ciencia y la tecnología se muestra como una tendencia que cada vez se va globalizando, permitiendo al productor analizar los datos y extraer un análisis de estos, que son necesarios para mantener un control y monitoreo de su producción. [3]

El análisis de los datos se focaliza en obtener información necesaria en cuanto al comportamiento de los patrones que influyen en la medición de la calidad de la leche, los resultados obtenidos serán determinantes para la toma de decisiones, actualmente para realizar la medición de la leche se realiza de forma manual no a gran escala, donde interviene de la mano de obra y material.

En cuanto a todo este proceso, es necesario tener toda la información a la mano utilizando herramientas tecnológicas, con el objeto de optimizar los recursos por consiguiente obtener resultados incontestables.

Los pequeños productores comprenden el 33,6% del total de productores nacionales que equivale a una producción de 178.010 litros diarios, estos productores no tienen equipos tecnológicos para la producción y tratamiento de leche cruda. Así que hoy se ve la producción de productos a partir de leche cruda

Dificultado debido a los tiempos, la cantidad de leche procesada y la adición de



procesamiento de forma artesanal sin estricto control higiénico, por estas razones se desarrolla el prototipo para medir la calidad de la leche mediante Arduino, que sirve como referencia que permitirá tener un proceso seguro, eficiente y controlado en todas las etapas.

La realización de este proyecto es importante porque contribuye a la realización de estudios macros significativos para el uso máximo de leche cruda en pequeños productores, beneficiándolos del aumento del costo por litro al obtener sus propios productos y asegurar el uso de los recursos. [4]

#### G. ALCANCE

- Este proyecto se enfocará directamente a la medición de la calidad de la producción de la leche bovina.
- El Módulo eléctrico desarrollado, incorpora sensores para medir el pH, densidad, temperatura.
- Base de datos alojada en un servidor web gratuito (apache)
- Se logrará que el prototipo cuente con un módulo de visualización para el usuario que permita ver los resultados
- El sistema del prototipo aplicará el método CRUD.

#### H. CONCEPTOS REACIONADOS

**Arduino Uno:** El Arduino Uno es una placa de microcontrolador conectado a tierra en el ATmega328 (hoja de datos). Comprende de 14 pines de entrada / salida digital (de los cuales 6 pueden utilizarse como salidas PWM), 6 entradas analógicas, un 16 MHz.

Resonador cerámico, una facilitación para la conectividad USB, una toma de corriente, un encabezado ICSP y un botón de reinicio. Aquí la energía eléctrica se transforma en sonido para enviar el pulso. El sonido que se recibe de vuelta se convierte en electricidad. Así, el lapso entre la señal de sonido enviada y recibida se utiliza para estimar la distancia al objeto. El espaciado entre sensores es perseguido por sus ángulos de haz. Los sensores deben estar espaciados para que no interfieran entre sí. Esta interferencia a veces se denomina "Diafonía". El objetivo debe estar montado perpendicular al eje del sensor. [5]

**IOT:** IoT visualiza un mundo totalmente conectado, donde las cosas están capaces de comunicar datos medidos e interactuar con los demás. Esto hace posible una representación digital del mundo real, a través del cual muchas aplicaciones inteligentes en una variedad de industrias. Pueden ser desarrolladas. Estos incluyen: Smart homes, Wearables, Smart ciudades, salud, automoción, medio ambiente, agua inteligente, etc. Las soluciones de IoT se están implementando en muchas áreas, optimizando las industrias de la producción. Características muy específicas, grandes volúmenes de



datos y necesidades grandes y potencia durante largos períodos. Se ha aplicado a muchas otras áreas, y la información inmutabilidad está garantizada en aplicaciones que va más allá de las criptomonedas. [6]

**Inteligencia Artificial:** Es la habilidad de los ordenadores para hacer actividades que normalmente requieren inteligencia humana. Pero, para brindar una definición más detallada, podríamos decir que la IA es la capacidad de las máquinas para usar algoritmos, aprender de los datos y utilizar lo aprendido en la toma de decisiones tal y como lo haría un ser humano. Sin embargo, a diferencia de las personas, los dispositivos basados en IA no necesitan descansar y pueden analizar grandes volúmenes de información a la vez. Asimismo, la proporción de errores es significativamente menor en las máquinas que realizan las mismas tareas que sus contrapartes humanas. La idea de que los ordenadores o los programas informáticos puedan tanto aprender como tomar decisiones es particularmente importante y algo sobre lo que deberíamos ser conscientes, ya que sus procesos están creciendo exponencialmente con el tiempo. Debido a estas dos capacidades, los sistemas de inteligencia artificial pueden realizar ahora muchas de las tareas que antes estaban reservadas sólo a los humanos. [7]

**Determinación PH:** El pH normal de la leche fresca es de 6.5-6,7. Valores superiores generalmente se observan en leches mastíticas, mientras que valores inferiores indican presencia de calostro o descomposición bacteriana

La determinación del pH de la leche puede hacerse por un método colorimétrico utilizando indicadores, pero resulta inexacto por la opacidad de la leche que interfiere en la lectura del color y además porque solo da valores aproximados. EL método más adecuado es el electrométrico empleado un electrodo de vidrio en combinación con un electrodo de referencia. El potencial se mide directamente en términos de pH en la escala de un potenciómetro calibrado con una solución buffer de pH conocido. [8]

**Características Sensoriales de la Leche:** Las Características organolépticas de la leche constituyen un atributo de calidad fundamental en cualquier producto alimenticio, la presencia de sabores, olores, colores o texturas atípicas en la leche limitan seriamente su adecuación al uso. La leche es un líquido blanco opaca, este aspecto característico manifiesta cuando toda la caseína se encuentra en forma micelar cuando disminuye la proporción de caseína la leche toma un aspecto grisáceo como el caso del calostro de los primeros días.

**Sensores:** Los Sensores son componentes utilizados para leer e interpretar las variables de entorno, el sensor se refiere a un elemento que producto una señal relacionada con la cantidad que se está midiendo, por ejemplo, en el caso de un elemento para medir la temperatura mediante resistencia. Con frecuencia se utiliza el término transductor en vez de sensor. Un transductor se define como el elemento que al someterlo a un cambio físico experimenta un cambio relacionado. Un sensor o transductor es analógico si ofrece un salida que son análoga y de esta manera cambia de forma



continua y por lo general tiene una salida cuyo tamaño es proporcional al tamaño de la variable que se está midiendo, el término digital se emplea cuando los sistemas ofrecen salidas que son digitales por naturaleza, por ejemplo, una secuencia de encendido/apagado principalmente, que arrojan un número cuyo valor se relaciona con el tamaño de la variable que se está midiendo. [9]

**Arduino ethernet Shield:** El Arduino ethernet shield nos da la capacidad de conectar un Arduino a una red ethernet. Es la parte física que implementa la pila de protocolos TCP/IP. Está basada en el chip ethernet Wiznet W5100. El Wiznet W5100 provee de una pila de red IP capaz de soportar TCP y UDP. Soporta hasta cuatro conexiones de sockets simultáneas. Usa la librería Ethernet para leer y escribir los flujos de datos que pasan por el puerto ethernet. Me permitirá escribir sketches que se conecten a internet usando esta shield. [10]

**Sensor de PH:** El medidor de pH es un instrumento utilizado para medir la acidez o la alcalinidad de una solución, también llamado de pH. El pH es la unidad de medida que describe el grado de acidez o alcalinidad y es medido en una escala que va de 0 a 14.

Las informaciones cuantitativas dadas por el valor del pH expresan el grado de acidez de un ácido o de una base en términos de la actividad de los iones de hidrógeno. El valor del pH de determinada sustancia está directamente relacionado a la proporción de las concentraciones de los iones de hidrógeno  $[H^+]$  e hidroxilo  $[OH^-]$ . Si la concentración de  $H^+$  es mayor que la de  $OH^-$ , el material es ácido; el valor del pH es menor que 7. Si la concentración de  $OH^-$  es mayor que la de  $H^+$ , el material es básico, con un pH con valor mayor que 7. Si las cantidades de  $H^+$  y de  $OH^-$  son las mismas, el material es neutral y su pH es 7. Ácidos y bases tienen, respectivamente, iones de hidrógeno y de hidroxilo libres. [11]

**Sensor de Temperatura SONDA DS18B20:** Sensor DS18B20

El DS18B20 es un sensor digital de temperatura que utiliza el protocolo 1-Wire para comunicarse, este protocolo necesita solo un pin de datos para comunicarse y permite conectar más de un sensor en el mismo bus, el encapsulado de fábrica es tipo TO-92 similar al empleado en transistores pequeños. Con este sensor podemos medir temperatura desde los  $-55^{\circ}C$  hasta los  $125^{\circ}C$  y con una resolución programable desde 9 bits hasta 12 bits. Cada sensor tiene una dirección única de 64bits establecida de fábrica, esta dirección sirve para identificar al dispositivo con el que se está comunicando, puesto que en un bus 1-wire pueden existir más de un dispositivo. [12]

## I. TRABAJOS RELACIONADOS

Existen distintos autores que han desarrollado investigaciones relacionadas sobre el tema, cuyos resultados han generado una guía de las mejores prácticas a tomarse a consideración. A continuación, se describe algunas de ellas:

Un estudio similar realizado en la Universidad Politécnica de Valencia en la Facultad de Ingeniería del Diseño, ensayo titulado “Medida automática de la producción de leche de ganado caprino.” que tiene como objetivo: diseñar un sistema de medida electrónica accesible para pequeños ganaderos. Para de esta forma poder dotar tanto a grandes ganaderos como pequeños de los sistemas necesarios para el correcto funcionamiento de su ocupación. En este estudio realizan una comparativa relevante entre los diferentes tipos de Arduino que se puede utilizar y optar con el sistema que más nos favorezca. [13]

En lo que se refiere a Latinoamérica se encuentra un estudio realizado en la Universidad Nacional de Trujillo del Perú estudio denominado “Diseño y montaje de un pasteurizador solar y evaluación en el tratamiento de leche de cabra”, que tiene como objeto diseñar e instalar un pasteurizador solar en una zona rural, evaluar el efecto tiempo temperatura en la pasteurización de leche de cabra. Lo destacable del proyecto es que el control de temperatura se realizó con tres sensores LM35 acoplados a una tarjeta Arduino Uno y a un ordenador en la cual se instalaron drivers de la tarjeta, el software LabVIEW y una interfaz de LabVIEW para Arduino, en donde se realizó la programación gráfica. [14]

A nivel nacional de Determina la Investigación similar realizada en la Universidad/ Escuela Superior Politécnica de Chimborazo de la República del Ecuador en la facultad de Ingeniería Electrónica, elaborado por Freddy Patricio Ajila Zaquinaula, Christian Sánchez-Acosta, Juan Carlos Pacheco-Reinoso, Arisdorgan Diéguez Almaguer, proyecto que titula “Automatización de una sala de ordeño para el control de compuertas, dosificación de alimento y medición de leche, caso de estudio: ESPOCH-Ecuador”, que se traza como objetivo: el desarrollo de un sistema automatizado para una sala de ordeño en el Centro Experimental Tunshi de la Escuela Superior Politécnica de Chimborazo-Ecuador definen una propuesta que está integrada por tres subsistemas que cumplen funciones específicas; estos son: sistema de control de compuertas, sistema de dosificación de alimento y sistema de medición de leche; subsistemas que utilizan un Controlador Lógico Programable (CLP) que se encarga de la comunicación con los dispositivos como sensores, electroválvulas y actuadores neumáticos. [15]



## J. METODOLOGÍA

La metodología que se utiliza para el desarrollo del presente proyecto de investigación se utilizará el método Mixto.

La aplicación de este método,

Para comparar las diferentes técnicas de medición de la calidad de la leche, tomando como referencia las distintas escalas.

Además, para el desarrollo del prototipo se recurre a la utilización de la herramienta Frigzi para la representación gráfica del diseño de este. Para el repositorio de datos se utilizará la base de datos MySQL, el módulo electrónico será elaborado en el IDE Arduino.

Que se cargará al microcontrolador de la Tarjeta de Arduino Uno que interactuarán con los sensores entre sí conectados a la placa, recibiendo datos captados por los sensores y enviando información de respuesta inmediata en un display para visualización del usuario.

La Arquitectura destinada para el proyecto es la MVC (Modelo Vista Controlador), el funcionamiento característico sería:

1. El usuario interactúa de alguna forma con la vista, es decir, con la interfaz de usuario, por ejemplo, pulsando cualquier botón de la aplicación. Esa solicitud llega al controlador, que se encargará de gestionarla de ahora en adelante hasta que esté resuelta.
2. El controlador gestiona el evento que se acaba de disparar, normalmente a través de un gestor de eventos, y comienza la comunicación tanto con los modelos como con las vistas. Se consulta la lógica de la aplicación y la lógica de negocio, de modo que se solicitará a los modelos los datos que sean necesarios, mientras que haremos la petición a la vista de cómo deben ser mostrados.
3. El controlador se comunica con el modelo, actualizándolo siempre, y realizando las modificaciones que el usuario, a través de su petición, haya solicitado
4. El controlador, con los datos obtenidos de los modelos, se encargará de enviarlos a las vistas, haciendo de nexo de comunicación. Sería corriente tanto una cosa como la otra, todo depende de nuestra implementación.
5. Las vistas muestran la salida al usuario.
6. La interfaz de usuario se pone a la espera de nuevas solicitudes por parte del usuario, reiniciando el ciclo. [16]

En Correlación con este tipo de estudio que tiene el propósito de evaluar la calidad de la leche y la relación que existen entre las diferentes variables.

**Metodología Programación XP:** La Programación extrema es una metodología ágil cuyas características más relevantes parten del Manifiesto Ágil. Se las conoce como ágiles por la rapidez en que suelen adaptarse al contexto de trabajo. XP se enmarca en las metodologías ágiles por su enfoque en el trabajo en equipo y la forma en como permite la rápida integración de las actividades del grupo. Esta metodología permite una revisión continua del código elaborado por el grupo de trabajo, permite la ejecución de pruebas unitarias y elaboración de prototipos mostrados al cliente en intervalos de tiempo más cortos. La metodología XP se divide en seis fases:

Fase I: Planificación Ser efectúan las historias de usuario, en donde se recopilan los requerimientos del sistema.

Fase II: Diseño En esta fase, se priorizan las historias de usuario, se estima el esfuerzo, se estructura el cronograma de actividades y se coordinan las fechas de la primera entrega. Fase III:

Codificación Se establece iteraciones cortas, no mayores a tres semanas, en donde se define la arquitectura del sistema.

Fase IV: Prueba Esta fase trata sobre las pruebas previas que se realizan al sistema antes de que llegue completamente al cliente.

[17]

Para acaparar lo expuesto se debe seguir los siguientes pasos:

- Revisar Artículos relacionados al tema de fuentes Primarias
- Análisis de Requerimientos Herramientas y Materiales
- Selección y Diseño de la Base de Datos
- Presentación y Pruebas del Sistema
- Codificación del Módulo Electrónico



K. CRONOGRAMA DE ACTIVIDADES								
N°	ACTIVIDAD	MES						MEDIOS DE VERIFICACIÓN
		I	II	III	IV	V	VI	
1	<b>Realizar un estudio de la problemática y técnicas del control de la calidad de la leche.</b>							
1.1	Revisar Artículos relacionados al tema de fuentes Primarias	2						Antecedentes
1.2	Efectuar una Comparación entre las diferentes técnicas	2						Desarrollo del Informe
2	<b>Diseñar un módulo electrónico que obtenga la medición de los parámetros de la calidad de la leche</b>							
2.1	Análisis de Requerimientos Herramientas y Materiales		1					Hardware
2.2	Diagramación y Conexión de los Componentes		1					Desarrollo del Informe
2.3	Selección y Diseño de la Base de Datos		2					Desarrollo del Informe
2.4	Codificación del Módulo Electrónico			4	4	4		Desarrollo del Informe
3	<b>Presentar un prototipo para la medición de la calidad de la leche</b>							
3.1	Prueba del Sistema						1	Pruebas del Prototipo
3.2	Presentación del Sistema						1	Pruebas del Prototipo
3.3	Manual del Sistema						1	Manual

#### L. DECLARACION FINAL

Los abajo firmantes declaramos bajo juramento que el proyecto descrito en este documento no ha sido presentado a otra institución nacional o internacional para su financiamiento, no causa perjuicio al ambiente, es de nuestra autoría y no transgrede norma ética alguna.

#### M. PARTICIPANTES

DIRECTOR:	ING. JOSÉ CARRILLO ZENTENO
ESTUDIANTE 1	EST. CARLOS REINALDO MONTOYA PONCE



**N. FIRMAS DE RESPONSABILIDAD**

**Lugar:** CAÑAR

**Fecha:** 12/02/2021

**Firmas:**

Nombre: Ing. José Carrillo  
CC:  
Ing. José Carrillo  
**Director del Proyecto**

Nombre: Carlos Montoya  
C.C.:  
**Estudiante / Egresado**

**O. APROBACIÓN**

**Firmas:**

Nombre: Ing. José Carrillo  
CC:  
**Primer Par Revisor**

Nombre:  
C.C.:  
**Segundo Par Revisor**



## P. REFERENCIAS

- Molina Cruz, J. (2013). *dspace.uclv.edu.cu*. Obtenido de <https://dspace.uclv.edu.cu/bitstream/handle/123456789/7841/Tesis%20MGT%20JavierMolinaCruz.pdf?sequence=1&isAllowed=y>
- Arrieta Rodríguez, C. A. (2018). Análisis DOFA del sector logístico portuario de la ciudad de Cartagena. *Revista Científica Anfibios ISSN: 2665-1513*, 28-43.
- Audisio, N. J. (2006). *Gestión por beneficios*. Córdoba: Editorial Brujas.
- Burgwal, G., & Juan, C. C. (1999). *Planificación estratégica y operativa aplicada a gobiernos locales: manual de facilitación : incluye materiales para los participantes*. Editorial Abya Yala.
- Cervantes, O. A. (2015). *Administración Estratégica - Análisis PEST*.
- Crespo, E. N. (2019). *Competitividad en el sector TIC*. Obtenido de [https://www.andaluciaesdigital.es/c/document\\_library/get\\_file?uuid=0f330afa-a9fb-412e-8295-fdda12fe2c56&groupId=20195](https://www.andaluciaesdigital.es/c/document_library/get_file?uuid=0f330afa-a9fb-412e-8295-fdda12fe2c56&groupId=20195)
- Cristina Fernández, J. B. (2020). *Las plataformas digitales, la productividad y el empleo en Colombia*. Bogotá: FEDESARROLLO.
- Daniel Martínez, A. M. (2012). *La elaboración del plan estratégico y su implantación a través del cuadro de mando integral*. Madrid: Díaz de Santos.
- Dayana Forero, E. S. (2020). Implementation model of Digital Transformation projects in Boutique Hotels. *18th LACCEI International Multi-Conference for Engineering, Education, and Technology*. Bolívar: LACEI.
- Figueroa Morán, G. L., Paladines Morán, J. P., Paladines Morán, J. N., Caicedo Plúa, C. R., & Romero Castro, M. I. (2017). *MODELO DE PLAN ESTRATÉGICO DE SISTEMAS PARA LA GESTIÓN Y ORGANIZACIÓN A TRAVÉS DE UNA PLATAFORMA INFORMÁTICA*. 3Ciencias.
- Flores Ruiz, C. A. (2014). *repositorio.unc.edu.pe*. Obtenido de <https://repositorio.unc.edu.pe/bitstream/handle/UNC/704/T%20657%20F634%202014.pdf?sequence=1&isAllowed=y>
- Fred R., D. (2003). *Conceptos de administración estratégica*. Mexico: Pearson Educación.
- Gonzales Agüero, P. E. (12 de Diciembre de 2019). *Core*. Obtenido de Core: <https://core.ac.uk/download/pdf/322973618.pdf>
- Gutiérrez, L. S. (Noviembre de 2020). El Uso de las Plataformas Digitales, la Nueva Dimensión del Marketing Digital. *Universidad Cooperativa de Colombia*. Obtenido de [repository.ucc.edu.co: https://repository.ucc.edu.co/bitstream/20.500.12494/33404/1/2020\\_uso\\_plataforma\\_digitales.pdf](https://repository.ucc.edu.co/bitstream/20.500.12494/33404/1/2020_uso_plataforma_digitales.pdf)
- Ian Lawrence Webster, A. C. (2014). *Planeación y Gestión Estratégica de las TI*. Bogotá: RENATA.
- Iris López-de-Solís, C. M.-L. (2011). Nuevas estrategias de negocio y valorización de los archivos audiovisuales en Internet. *El profesional de la información*, 8. Obtenido de [http://eprints.rclis.org/34157/1/nuevas\\_estrategias.pdf](http://eprints.rclis.org/34157/1/nuevas_estrategias.pdf)
- ISOTools. (s.f.). *www.isotools.org*. Obtenido de [www.isotools.org: https://www.isotools.org/soluciones/estrategia/balanced-scorecard/](https://www.isotools.org/soluciones/estrategia/balanced-scorecard/)
- KIRBY, L. G. (Septiembre de 2020). *UIDE*. Obtenido de <https://repositorio.uide.edu.ec/bitstream/37000/4410/1/T-UIDE-0100.pdf>
- Luna González, A. C. (2016). *Plan estratégico de negocios*. Grupo Editorial Patria: Mexico.
- Manes, J. M. (2014). *Gestión estratégica para instituciones educativas*. Buenos Aires: Ediciones Granica S.A.
- Ordaz Zubia, V. Y. (2000). *Análisis y crítica de la metodología para la realización de planes regionales en el estado de Guanajuato*. Guanajuato: Juan Carlos Martínez Coll.
- Ortiz, N. G. (2020). Estrategia del Océano Azul para el sector portuario (sincromodalidad y digitalización). *Entre Ciencia e Ingeniería*, 50-57.
- Palacios Acero, L. C. (s.f.). *DIRECCIÓN ESTRATÉGICA*. ECOE Ediciones.



Pedros, D. M., & Gutiérrez, A. M. (2012). *La elaboracion del plan estratégico a través del Cuadro de Mando Integral*. Madrid: Ediciones Díaz de Santos.

Pereyra, L. E. (2020). *Administración II*. Mexico: Klik.

Pérez Mesa, J. C. (2016). *Operaciones y gestión de empresas turísticas*. Universidad Almería.

Robert S. Kaplan, D. P. (1996). *The Balanced Scorecard: Translating Strategy into Action*. Boston: Harvard Business Review Press.

Rodriguez Trujillo, R. R. (s.f.). *PlanificaciÓn EstratÉgica*. Clube de Autores (managed).

Rojas, J. L. (2009). Procedimiento para la elaboración de un análisis FODA como una herramienta de planeación estratégica en las empresas. *Ciencia Administrativa IIESCA*, 54-61.

Ruiz Leiva, H. F., & Briones Amador, K. K. (04 de 12 de 2017). *repositorio.unan.edu.ni*. Obtenido de <https://repositorio.unan.edu.ni/8973/1/18835.pdf>

Zamarreño Aramendia, G. (2019). *Marketing Estratégico*. España: Editorial Elearning, S.L.

## ***Anexo 2: Codigo Fuente de la Plataforma***

### **Capa Modelo**

```
<?php
defined('BASEPATH') or exit('No direct script access allowed');
class Clientes_model extends CI_Model
{
    public function
nuevo_cliente($Empresa_id,$Nom_cliente,$Dir_cliente,$Telf_cliente,$Email_cliente)
    {
        //compruebo si no hay alguien con ese mail.
        $this->db->select('Email_cliente');
        $this->db->from('cliente');
        $this->db->where('Email_cliente', $Email_cliente);
        $query = $this->db->get();

        if($query->num_rows() > 0) {
            return "existe";
        } else {
            $data = array(
                'Nom_cliente' => $Nom_cliente,
                'Dir_cliente' => $Dir_cliente,
                'Telf_cliente' => $Telf_cliente,
```



```
'Email_cliente' => $Email_cliente,
'Empresa_id' => $Empresa_id
);
if ($this->db->insert('cliente', $data)) { //inserto el nuevo usuario
    return 1;
}else{
    return "fallo";
}
}
}
public function listar_clientes($usuario_id)
{
    $this->db->select('Nom_cliente,Dir_cliente,Telf_cliente,Email_cliente'); // selecciono
columnas de tabla dispositivos
    $this->db->from('cliente'); // de la tabla dispositivos
    $result = $this->db->get()->result_array(); // obtengo el resultado en un array!
    //echo "<pre>";
    //print_r($result);
    //echo "<pre>";
    //die(); //descomentar estas dos líneas para debugear...
    return $result; //devuelvo el array con la lista de dispositivos al controlador
}
public function get_empresas($usuario_id) {
    $this->db->where('Usuario_id', $usuario_id);
    $this->db->order_by('Nom_empresa', 'asc');
    $empresas = $this->db->get('empresa');

    if($empresas->num_rows() > 0){
        return $empresas->result();
    }
}
}
```



```
<?php
defined('BASEPATH') or exit('No direct script access allowed');
class Empresas_model extends CI_Model
{
    public function nueva_empresa($usuario_id,$nombre,$direccion,$telefono,$email)
    {
        //compruebo si no hay alguien con ese mail.
        $this->db->select('Email_empresa');
        $this->db->from('empresa');
        $this->db->where('Email_empresa', $email);
        $query = $this->db->get();
        if($query->num_rows() > 0) {
            return "existe";
        } else {
            $this->db->select('Nom_empresa');
            $this->db->from('empresa');
            $this->db->where('Nom_empresa', $nombre);
            $query1 = $this->db->get();
            if($query1->num_rows() > 0){
                return "existe";
            }
            else {
                $data = array(
                    'Nom_empresa' => $nombre,
                    'Dir_empresa' => $direccion,
                    'Telf_empresa' => $telefono,
                    'Email_empresa'=> $email,
                    'Usuario_id' => $usuario_id
                );
                if ($this->db->insert('empresa', $data)) { //inserto la nueva empresa
                    return 1;
                }else{
```



```
        return "fallo";
    }
}
}
}
public function listar_empresas($usuario_id)
{
    $this->db->select('Nom_empresa,Dir_empresa,Telf_empresa,Email_empresa'); //
seleciono columnas de tabla dispositivos
    $this->db->from('empresa'); // de la tabla dispositivos
    $this->db->where('Usuario_id', $usuario_id);
    $result = $this->db->get()->result_array(); // obtengo el resultado en un array!
    //echo "<pre>";
    //print_r($result);
    //echo "<pre>";
    //die(); //descomentar estas dos líneas para debugear...
    return $result; //devuelvo el array con la lista de dispositivos al controlador
}
}
<?php
defined('BASEPATH') or exit('No direct script access allowed');

class General_model_cleche extends CI_Model
{
    public function ultimos_valores()
    {
        $this->db->select('temperatura,densidad,ph,r_ph,r_d,d_corregida'); // selecciono t1 y t2
        $this->db->from('temporales'); // de la tabla datos
        // $this->db->where('temporales_id',$temporales_id); //donde el dispositivo id = 1 (por
ahora)
        $this->db->order_by('temporales_id','desc'); //ordeno id descendente
        $this->db->limit(1); //solo un resultado por ende el ultimo
        $result = $this->db->get()->result_array();
    }
}
```



```
return $result;
}
public function get_clientes($empresas) {
    $this->db->where('Empresa_id', $empresas);
    $this->db->order_by('Nom_cliente', 'asc');
    $clientes = $this->db->get('cliente');
    if($clientes->num_rows() > 0){
        return $clientes->result();
    }
}
public function ingreso_datos($cliente_id)
{
    //$this->db->where('temporales_id',$temporales_id); //donde el dispositivo id = 1 (por
ahora)
    $this->db->order_by('temporales_id','desc'); //ordeno id descendente
    $result = $this->db->get('temporales');
    if($result->num_rows() > 0){
        $a = $result->result();
        foreach ($a as $re){
            $d=$re->d_corregida;
            $ph=$re->ph;
            $t=$re->temperatura;
            if ($d >= 1.028 && $d <= 1.033){
                $id_d="1";
            }
            elseif ($d < 1.028){
                $id_d="2";
            }
            elseif ($d > 1.033){
                $id_d="3";
            }
            else
```



```
{
    $id_d="4";
}
if ($ph >= 1.10 && $ph <= 1.80){
    $id_ph="1";
} elseif ($ph >= 1.90 && $ph <= 6.30) {
    $id_ph="2";
}
elseif ($ph >= 6.40 && $ph <= 6.80){
    $id_ph="3";
}
elseif ($ph >= 6.90 && $ph <= 14){
    $id_ph="4";
}
else {
    $id_ph="131";
}

//echo $id_d,$id_ph;

$data = array(
    'Cliente_id' => $cliente_id,
    'Sph_id' => $id_ph,
    'Sensores_id' => $id_d,
    'Temperatura' => $t,
    'Densidad' => $d,
    'Nivel_ph' => $ph
);
$this->db->insert('control', $data);
$this->db->empty_table('temporales');
//die();
}}
```



```
    echo '<script>alert("Datos ingresados correctamente")</script>';
}
public function sensor_historicos($id_cliente)
{
    $this->db->where('Cliente_id', $id_cliente);
    $this->db->order_by('id_control', 'asc');
    $dato_sensor = $this->db->get('control');
    return $dato_sensor->result();
}
public function datos_historicos_temperatura($id_cliente,$datei,$datef)
{
    $this->db->select('a.Temperatura'); //
    $this->db->from('control a'); // de la tabla dispositivos
    $this->db->join('sensores b','a.Sensores_id = b.Id_sensores');
    $this->db->where('a.Cliente_id',$id_cliente);
    $this->db->where('a.Fecha BETWEEN "'. date('Y-m-d H:i:s', strtotime($datei)). "' and '".
date('Y-m-d H:i:s', strtotime($datef))."'");
    //$this->db->limit(100);//solo un resultado por ende el ultimo
    $results = $this->db->get()->result_array(); // obtengo el resultado en un array!
    $result = "";
    foreach ($results as $resultado) {
        $result = $result.$resultado['Temperatura'] . ",";
    }
    return $result;
}
public function datos_historicos_densidad($id_cliente,$datei,$datef)
{
    $this->db->select('a.Densidad'); // selecciono columnas de tabla dispositivos
    $this->db->from('control a'); // de la tabla dispositivos
    $this->db->join('sensores b','a.Sensores_id = b.Id_sensores');
    $this->db->where('a.Cliente_id',$id_cliente);
    $this->db->where('a.Fecha BETWEEN "'. date('Y-m-d H:i:s', strtotime($datei)). "' and '".
date('Y-m-d H:i:s', strtotime($datef))."'");
```



```
$results = $this->db->get()->result_array(); // obtengo el resultado en un array!  
$result = "";  
foreach ($results as $resultado) {  
    $result = $result.$resultado['Densidad'] . ",";  
}  
return $result;  
}  
public function datos_historicos_ph($id_cliente,$datei,$datef)  
{  
    $this->db->select('a.Nivel_pH'); // selecciono columnas de tabla dispositivos  
    $this->db->from('control a'); // de la tabla dispositivos  
    $this->db->join('sensores b','a.Sensores_id = b.Id_sensores');  
    $this->db->where('a.Cliente_id',$id_cliente);  
    $this->db->where('a.Fecha BETWEEN "' . date('Y-m-d H:i:s', strtotime($datei)). '" and "' .  
date('Y-m-d H:i:s', strtotime($datef)). '"');  
    $results = $this->db->get()->result_array(); // obtengo el resultado en un array!  
    $result = "";  
    foreach ($results as $resultado) {  
        $result = $result.$resultado['Nivel_pH'] . ",";  
    }  
    return $result;  
}  
public function datos_historicos_fecha($id_cliente,$datei,$datef)  
{  
    $this->db->select('a.Fecha'); //  
    $this->db->from('control a'); // de la tabla dispositivos  
    $this->db->join('sensores b','a.Sensores_id = b.Id_sensores');  
    $this->db->where('a.Cliente_id',$id_cliente);  
    $this->db->where('a.Fecha BETWEEN "' . date('Y-m-d H:i:s', strtotime($datei)). '" and "' .  
date('Y-m-d H:i:s', strtotime($datef)). '"');  
    //$this->db->limit(100);//solo un resultado por ende el ultimo  
    $results = $this->db->get()->result_array(); // obtengo el resultado en un array!  
    $result = "";
```



```
foreach ($results as $resultado) {
    $result = $result.$resultado['Fecha'] . ",";
}
return $result;
}
}
<?php
defined('BASEPATH') or exit('No direct script access allowed');
class Login_model extends CI_Model
{
    public function login($email, $password)
    {
        $this->db->select('*'); // selecciono todo
        $this->db->from('usuario'); // de la tabla usuarios
        $this->db->where('Email_usuario', $email); //donde el email sea igual al email que le
paso
        $this->db->where('Password', $password); //donde la clave sea igual a la clave que yo le
paso

        $query = $this->db->get(); //obtengo los resultados
        if ($query->num_rows() == 1) { //si hay mas de un resultado es que EXITO!
            $result = $query->row();
            //todas las variables las pongo en sesion para cuando las necesite.
            $this->session->set_userdata(array(
                'Id_usuario'          => $result->Id_usuario,
                'Email_usuario'       => $result->Email_usuario,
                'Fecha'               => $result->Fecha,
                'Nom_usuario'         => $result->Nom_usuario
            ));
            return 1; //entonces le devuelvo un 1 al controlador
        } else {
            return 0; //entonces le devuelvo un 0 al controlador indicando que no coinciden
credenciales

```



```
    }  
  }  
}  
<?php  
defined('BASEPATH') OR exit('No direct script access allowed');  
class Register_model extends CI_Model {  
    public function register($usuario,$email, $password,$direccion,$self)  
    {  
        //compruebo si no hay alguien con ese mail.  
        $this->db->select('Email_usuario');  
        $this->db->from('usuario');  
        $this->db->where('Email_usuario', $email);  
        $query = $this->db->get();  
        if($query->num_rows() > 0) {  
            return "existe";  
        } else {  
            $password = sha1($password); //encripto clave antes de grabarla  
            $hoy = date ( "Y-m-d H:i:s" );  
            $data = array(  
                'Nom_usuario'=> $usuario,  
                'Email_usuario' => $email,  
                'Password' => $password,  
                'Telf_usuario' => $self,  
                'Dir_usuario' => $direccion  
            );  
            if ($this->db->insert('usuario', $data)){ //inserto el nuevo usuario  
                return 1;  
            }else{  
                return "fallo";  
            }  
        }  
    }  
}
```

```

public function cambio_clave($password_anterior, $password, $email){
    $password = $password = sha1($password);
    if( $this->db->query("update users set password = ".$password." where mail
= ".$email."")){
        return 1;
    }else{
        return 0;
    }
}
}
}
<?php
defined('BASEPATH') or exit('No direct script access allowed');
class toma_de_datos_model extends CI_Model
{
    public function get_empresas($usuario_id) {
        $this->db->where('Usuario_id', $usuario_id);
        $this->db->order_by('Nom_empresa', 'asc');
        $empresas = $this->db->get('empresa');
        if($empresas->num_rows() > 0){
            return $empresas->result();
        }
    }
}
public function estadoingresodatos()
{
    $this->db->select('control_ingreso'); // selecciono columnas de controlEntrada
    $this->db->from('control_entrada'); // de la tabla entrada
    $control_ingreso = $this->db->get()->result_array();
    $controlIngreso = "";
    foreach ($control_ingreso as $resultado) {
        $controlIngreso = $controlIngreso.$resultado['control_ingreso'] . ",";
    }
    return $controlIngreso;
}

```



```
}  
  
public function  
nuevo_dato_sensores($temperatura,$densidad,$ph,$r_d,$r_ph,$d_corregida)  
{  
    $data = array(  
        'temperatura' => $temperatura,  
        'densidad' => $densidad,  
        'ph' => $ph,  
        'r_d' => $r_d,  
        'r_ph' => $r_ph,  
        'd_corregida' => $d_corregida  
    );  
    //$this->db->where('temporales_id', '1');  
    $this->db->insert('temporales', $data);  
    //$this->db->update('temporales', $data);  
    echo 'datos temporales ingresados';  
}  
  
public function get_clientes($empresas) {  
    $this->db->where('Empresa_id', $empresas);  
    $this->db->order_by('Nom_cliente', 'asc');  
    $clientes = $this->db->get('cliente');  
    if($clientes->num_rows() > 0){  
        return $clientes->result();  
    }  
}  
  
public function cierraingresodatos() {  
    $data = array(  
        'control_ingreso' => 0  
    );  
    //$this->db->where('temporales_id', '1');  
    //$this->db->insert('temporales', $data);
```



```
        $this->db->update('control_entrada', $data);
    }
    public function abreingresodatos() {
        $data = array(
            'temperatura' => 0,
            'densidad' => 0,
            'ph' => 0,
            'r_d' => 0,
            'r_ph' => 0,
            'd_corregida' => 0
        );
        //$this->db->where('temporales_id', '1');
        $this->db->insert('temporales', $data);
        //$this->db->update('temporales', $data);
        echo 'datos temporales ingresados';
        $data = array(
            'control_ingreso' => 1
        );
        //$this->db->where('temporales_id', '1');
        //$this->db->insert('temporales', $data);
        $this->db->update('control_entrada', $data);
    }
    public function get_nombre_clientes($id_cliente) {
        $this->db->where('Id_cliente', $id_cliente);
        //$this->db->select('Nom_cliente');
        //$this->db->order_by('Nom_cliente', 'asc');
        $clientes = $this->db->get('cliente');
        if($clientes->num_rows() > 0){
            return $clientes->result();
        }
    }
}
```

## Capa Vista

```
<div class="image-fondo cabecera">
```

```
  <h1></h1>
```

```
</div>
```

```
<div class="content">
```

```
  <h1>SISTEMA DE CONTROL DE CALIDAD DE LA LECHE</h1>
```

```
  <h3>Carlos Montoya - Universidad Católica de Cuenca</h3>
```

```
  <!-- Use a button to pause/play the video with JavaScript -->
```

```
</div>
```

```
<video autoplay muted loop id="myVideo">
```

```
  <source src="/dist/img/FONDO2.mp4" type="video/mp4">
```

```
</video>
```

```
<style>
```

```
#myVideo {
```

```
  height: 850px;
```

```
  line-height: 200px;
```

```
  object-fit: cover;
```

```
  right: 0;
```

```
  bottom: 0;
```

```
  z-index: -1;
```

```
  min-width: 100%;
```

```
  min-height: 100%;
```

```
}
```

```
/* Add some content at the bottom of the video/page */
```

```
.content {  
    position: fixed;  
    bottom: 0;  
    background: rgba(0, 0, 0, 0.3);  
    color: #f1f1f1;  
    width: 100%;  
    padding: 20px;  
    text-align: center;  
}  
  
/* Style the button used to pause/play the video */  
#myBtn {  
    width: 200px;  
    font-size: 18px;  
    padding: 10px;  
    border: none;  
    background: #000;  
    color: #fff;  
    cursor: pointer;  
}  
  
#myBtn:hover {  
    background: #ddd;  
    color: black;  
}  
</style>  
  
<!DOCTYPE html>  
<html>  
<head>
```



```
<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<title>AdminLTE 2 | Log in</title>

<!-- Tell the browser to be responsive to screen width -->

<meta content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no"
name="viewport">

<!-- Bootstrap 3.3.7 -->

<link rel="stylesheet" href="<?php echo
base_url('bower_components/bootstrap/dist/css/bootstrap.min.css')?>">

<!-- Font Awesome -->

<link rel="stylesheet" href="<?php echo base_url('bower_components/font-
awesome/css/font-awesome.min.css')?>">

<!-- Ionicons -->

<link rel="stylesheet" href="<?php echo
base_url('bower_components/Ionicons/css/ionicons.min.css')?>">

<!-- Theme style -->

<link rel="stylesheet" href="<?php echo base_url('dist/css/AdminLTE.min.css')?>">

<!-- iCheck -->

<link rel="stylesheet" href="<?php echo base_url('plugins/iCheck/square/blue.css')?>">

<!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and media queries -->
<!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
<!--[if lt IE 9]>
<script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>
<script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
<![endif]-->

<!-- Google Font -->

<link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,600,700,300italic,4
00italic,600italic">

</head>

<body class="hold-transition login-page">
```

```

<div class="login-box">
  <div class="login-logo">
    <a href=" ../index2.html"><b>Admin</b>LTE</a>
  </div>
  <!-- /.login-logo -->
  <div class="login-box-body">
    <p class="login-box-msg">Sign in to start your session</p>

<?php echo base_url('el resto de la ruta'); ?>

  <form action="/login/hacerlogin" method="post">
    <div class="form-group has-feedback">
      <input type="email" class="form-control" placeholder="Email" name="email">
      <span class="glyphicon glyphicon-envelope form-control-feedback"></span>
    </div>
    <div class="form-group has-feedback">
      <input type="password" class="form-control" placeholder="Password"
name="password">
      <span class="glyphicon glyphicon-lock form-control-feedback"></span>
    </div>
    <div class="row">
      <div class="col-xs-8">
        <div class="checkbox icheck">
          <label>
            <input type="checkbox"> Remember Me
          </label>
        </div>
      </div>
    </div>
    <!-- /.col -->
    <div class="col-xs-4">
      <button type="submit" class="btn btn-primary btn-block btn-flat">Sign In</button>

```

</div>

<!-- /.col -->

</div>

</form>

<div class="" style="color:red">

<?php echo \$msg ?> <!-- /estoy imprimiendo el mensaje que me pasa la vista cuando el user pone mal la credencial -->

</div>

<div class="social-auth-links text-center">

<p>- OR -</p>

<a href="#" class="btn btn-block btn-social btn-facebook btn-flat"><i class="fa fa-facebook"></i> Sign in using

Facebook</a>

<a href="#" class="btn btn-block btn-social btn-google btn-flat"><i class="fa fa-google-plus"></i> Sign in using

Google+</a>

</div>

<!-- /.social-auth-links -->

<a href="#">I forgot my password</a><br>

<a href="/register" class="text-center">Register a new membership</a>

</div>

<!-- /.login-box-body -->

</div>

<!-- /.login-box -->

<!-- jQuery 3 -->

```
<script src="<?php echo
base_url('bower_components/jquery/dist/jquery.min.js')?>"></script>
<!-- Bootstrap 3.3.7 -->
<script src="<?php echo
base_url('bower_components/bootstrap/dist/js/bootstrap.min.js')?>"></script>
<!-- iCheck -->
<script src="<?php echo base_url('plugins/iCheck/ichack.min.js')?>"></script>
<script>
$(function () {
    $('input').iCheck({
        checkboxClass: 'icheckbox_square-blue',
        radioClass: 'iradio_square-blue',
        increaseArea: '20%' /* optional */
    });
});
</script>
</body>
</html>

<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title>AdminLTE 2 | Registration Page</title>
    <!-- Tell the browser to be responsive to screen width -->
    <meta content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no"
name="viewport">
    <!-- Bootstrap 3.3.7 -->
    <link rel="stylesheet" href="<?php echo
base_url('bower_components/bootstrap/dist/css/bootstrap.min.css')?>">
    <!-- Font Awesome -->
```



```
<link rel="stylesheet" href="<?php echo base_url('bower_components/font-
awesome/css/font-awesome.min.css')?>">

<!-- Ionicons -->

<link rel="stylesheet" href="<?php echo
base_url('bower_components/Ionicons/css/ionicons.min.css')?>">

<!-- Theme style -->

<link rel="stylesheet" href="<?php echo base_url('dist/css/AdminLTE.min.css')?>">

<!-- iCheck -->

<link rel="stylesheet" href="<?php echo base_url('plugins/iCheck/square/blue.css')?>">

<!-- Google Font -->

<link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,600,700,300italic,4
00italic,600italic">

</head>

<body class="hold-transition register-page">

<div class="register-box">

<div class="register-logo">

<a href=" ../index2.html"><b>Admin</b>LTE</a>

</div>

<div class="register-box-body">

<p class="login-box-msg">Register a new membership</p>

<form action="/register/hacerregister" method="post">

<div class="form-group has-feedback">

<input type="email" class="form-control" placeholder="Email" name="email">

<span class="glyphicon glyphicon-envelope form-control-feedback"></span>

</div>

<div class="form-group has-feedback">
```

```

<input type="password" class="form-control" placeholder="Password"
name="password">
    <span class="glyphicon glyphicon-lock form-control-feedback"></span>
</div>
<div class="row">
    <div class="col-xs-8">
        <div class="checkbox icheck">
            <label>
                <input type="checkbox"> I agree to the <a href="#">terms</a>
            </label>
        </div>
    </div>
</div>
<!-- /.col -->
<div class="col-xs-4">
    <button type="submit" class="btn btn-primary btn-block btn-flat">Register</button>
</div>
<!-- /.col -->
</div>
</form>

<div class="" style="color:red">
    <?php echo $msg ?>
</div>

<div class="social-auth-links text-center">
    <p>- OR -</p>
    <a href="#" class="btn btn-block btn-social btn-facebook btn-flat"><i class="fa fa-
facebook"></i> Sign up using
    Facebook</a>
    <a href="#" class="btn btn-block btn-social btn-google btn-flat"><i class="fa fa-google-
plus"></i> Sign up using
    Google+</a>

```

</div>

<a href="/login" class="text-center">I already have a membership</a>

</div>

<!-- /.form-box -->

</div>

<!-- /.register-box -->

<!-- jQuery 3 -->

<script src="<?php echo  
base\_url('bower\_components/jquery/dist/jquery.min.js')?>"></script>

<!-- Bootstrap 3.3.7 -->

<script src="<?php echo  
base\_url('bower\_components/bootstrap/dist/js/bootstrap.min.js')?>"></script>

<!-- iCheck -->

<script src="<?php echo base\_url('plugins/iCheck/ichack.min.js')?>"></script>

<script>

\$(function () {

\$('#input').iCheck({

checkboxClass: 'checkbox\_square-blue',

radioClass: 'radio\_square-blue',

increaseArea: '20%' /\* optional \*/

});

});

</script>

</body>

</html>

<!-- Content Wrapper. Contains page content -->

<div class="content-wrapper">

<!-- Content Header (Page header) -->

```
<section class="content">
```

```
<!-- MIREN QUE POR CADA FILA CON CONTENIDO PORNGO UN "ROW" ESTE ES EL ROW QUE CONTIENE EL FORM -->
```

```
<div class="row">
```

```
<div class="col col">
```

```
<div class="box box-info">
```

```
<div class="box-header with-border">
```

```
<h3 class="box-title">Nuevo Cliente</h3>
```

```
</div>
```

```
<!-- /.box-header -->
```

```
<!-- form start -->
```

```
<form class="form-horizontal" method="POST" action="<?php echo base_url('Clientes/recibe_clientes')?>">
```

```
<div class="box-body">
```

```
<div class="form-group">
```

```
<label for="inputPassword3" class="col-sm-2 control-label">Empresa</label>
```

```
<div class="col-sm-10">
```

```
<select id="empresas" class="form-control" name="Empresa_id" onchange="d1(this)">
```

```
<option selected>Seleccionar</option>
```

```
<?php
```

```
foreach ($empresas as $i) {
```

```
    echo '<option value="'. $i->ID_empresa .'>'. $i->Nom_empresa .'</option>';
```

```
    }
```

```
?>
```

```
</select>
```

```
</div>
```

```
</div>
```

```
<div class="form-group">
```



```
<label for="inputEmail3" class="col-sm-2 control-label">Nombre</label>
<div class="col-sm-10">
  <input id="nombre" type="text" class="form-control" id="C1"
placeholder="Ingrese nombre" name="Nom_cliente">
</div>
</div>

<div class="form-group">
  <label for="inputPassword3" class="col-sm-2 control-label">Direccion</label>
  <div class="col-sm-10">
    <input id="direccion" type="text" class="form-control" id="C2"
placeholder="Ingrese dirección" name="Dir_cliente">
  </div>
</div>

<div class="form-group">
  <label for="inputPassword3" class="col-sm-2 control-label">Teléfono</label>
  <div class="col-sm-10">
    <input id="telefono" type="number" class="form-control" id="C3"
placeholder="Ingrese teléfono" name="Telf_cliente">
  </div>
</div>

<div class="form-group">
  <label for="inputPassword3" class="col-sm-2 control-label">Email</label>
  <div class="col-sm-10">
    <input id="mail" type="email" class="form-control" id="C4"
placeholder="Ingrese email" name="Email_cliente">
  </div>
</div>
</div>
```



```
<!-- /.box-body -->
<div class="box-footer">
  <input id="submit" type="button" class="btn btn-primary pull-right"
value="Registrar" />
</div>
<!-- /.box-footer -->
</form>
</div>
</div>
</div>

<!-- MIREN QUE POR CADA FILA CON CONTENIDO PORNGO UN "ROW" ESTE ES
EL ROW QUE CONTIENE LA LISTA DE DISPOSITIVOS -->
<div class="row">
  <div class="col">
    <div class="box box-info box-solid">
      <div class="box-header with-border">
        <h3 class="box-title">Clientes</h3>
        <div class="box-tools pull-right">
          <button type="button" class="btn btn-box-tool" data-widget="remove"><i
class="fa fa-times"></i></button>
        </div>
      <!-- /.box-tools -->
    </div>
    <!-- /.box-header -->
    <div class="box-body">
      <?php
      foreach ($clientes as $cliente) {
        echo " Nombre: ".$cliente['Nom_cliente']."<br>." Dirección:
".$cliente['Dir_cliente']."<br>."<br>";
      }
      ?>
    </div>
  </div>
</div>
</div>
```



```
</div>
<!-- /.box-body -->
</div>
<!-- /.box -->
</div>
</div>
</section>
</div>

<script src="<?php echo
base_url('bower_components/jquery/dist/jquery.min.js')?>"></script>
<script src="<?php echo base_url('bower_components/jquery-ui/jquery-ui.js')?>"></script>

<script>
    $(document).ready(function() {
        $("#submit").click(function() {

            var empresas = $("#empresas").val();
            var nombre = $("#nombre").val();
            var direccion = $("#direccion").val();
            var telefono = $("#telefono").val();
            var mail = $("#mail").val();

            //alert(clientesi);

            if(empresas == "" || nombre == "" || direccion == "" || telefono == "" || mail == "") {
                alert("Por favor rellene todos los campos.");
                return false;
            }

            $.ajax({
                type: "POST",
```

```
url: "Clientes/recibe_clientes",
data: {
  empresas: empresas,
  nombre: nombre,
  direccion: direccion,
  telefono: telefono,
  mail: mail
},
cache: false,
success: function(data) {
  if (data=='1')
  {
    alert('Cliente creado con exito');
    window.location.assign("Clientes");

  } else if (data=='existe') {
    alert('Cliente ya existe, verifique los datos , por favor');
  }
},
error: function(xhr, status, error) {
  console.error(xhr);
}
});
});
});
</script>
<!-- Content Wrapper. Contains page content -->
<div class="content-wrapper">
  <!-- Content Header (Page header) -->
  <section class="content-header">
```



```
<h1>
```

```
  Históricos
```

```
<small>Panel de Control</small>
```

```
</h1>
```

```
<ol class="breadcrumb">
```

```
  <li><a href="pinicio"><i class="fa fa-dashboard"></i> Home</a></li>
```

```
  <li class="active">Monitoreo</li>
```

```
</ol>
```

```
</section>
```

```
<!-- Main content -->
```

```
<section class="content">
```

```
<!-- Small boxes (Stat box) -->
```

```
  <div class="box box-info">
```

```
    <div class="box-header with-border">
```

```
      <h3 class="box-title">Historial de Datos</h3>
```

```
    </div>
```

```
  <!-- /.box-header -->
```

```
  <!-- form start -->
```

```
  <form class="form-horizontal" method="POST" action="<?php echo  
base_url('prueba2')?>">
```

```
    <div class="box-body">
```

```
      <div class="form-group">
```

```
        <label for="inputPassword3" class="col-sm-2 control-label">Empresa</label>
```

```
        <div class="col-sm-10">
```

```
          <select id="empresasi" class="form-control" name="empresas"  
onchange="d1(this)">
```

```
            <option selected>Seleccionar</option>
```

```
            <?php
```

```
        foreach ($empresas as $i) {
            echo '<option value="'. $i->ID_empresa .'>'. $i-
Nom_empresa .'</option>';

        }
    ?>
</select>
</div>
</div>

<div class="form-group">
    <label for="inputPassword3" class="col-sm-2 control-label">Cliente</label>
    <div class="col-sm-10">
        <select id="clientes" class="form-control" name="clientes"
onchange="d1(this)">
            <option selected>Seleccionar</option>
        </select>
    </div>
</div>

<div class="form-group">
    <label for="inputPassword3" class="col-sm-2 control-label">Desde</label>
    <div class="col-sm-10">
        <input id="iddatei" name="datei" type="datetime-local" step="1">
    </div>
</div>

<div class="form-group">
    <label for="inputPassword3" class="col-sm-2 control-label">Hasta</label>
    <div class="col-sm-10">
```



```
<input id="iddatef" name="datef" type="datetime-local" step="1">
</div>
</div>
</div>
<!-- /.box-body -->
<div class="box-footer">
  <input id="submit" type="button" class="btn btn-primary pull-right"
value="Iniciar" />
</div>
<figure class="highcharts-figure">
  <div id="container"></div>
  <div id="container2"></div>
</figure>
<div id="graph"></div>
<!-- /.box-footer -->
</form>
</div>
</section>
<section class="content" id=idParent>
</section>
<!-- /.content -->
</div>

<script src="<?php echo
base_url('bower_components/jquery/dist/jquery.min.js')?>"></script>

<script src="<?php echo base_url('bower_components/jquery-ui/jquery-ui.js')?>"></script>

<script type="text/javascript">
  $(document).ready(function() {
    $("#empresasi").change(function() {
```

```

    $("#empresasi option:selected").each(function() {
        empresas = $('#empresasi').val();

        $.post("<?php echo base_url(); ?>Datos_historicos/fill_clientes", {
            empresasi : empresas
        }, function(data) {
            $("#clientes").html(data);
        });
    });
});
});
});
</script>
<!-- /.content -->
<script src="<?php echo base_url('bower_components/Highcharts-7.1.0/code/highcharts.js')?>"></script>
<script src="<?php echo base_url('bower_components/Highcharts-7.1.0/code/modules/series-label.js')?>"></script>
<script src="<?php echo base_url('bower_components/Highcharts-7.1.0/code/modules/exporting.js')?>"></script>
<script src="<?php echo base_url('bower_components/Highcharts-7.1.0/code/modules/export-data.js')?>"></script>
<script src="https://code.highcharts.com/highcharts.js"></script>
<script src="https://code.highcharts.com/modules/series-label.js"></script>
<script src="https://code.highcharts.com/modules/exporting.js"></script>
<script src="https://code.highcharts.com/modules/export-data.js"></script>
<script src="https://code.highcharts.com/modules/accessibility.js"></script>

<script>
    $(document).ready(function() {
        $("#submit").click(function() {

```



```
var empresasi = $("#empresasi").val();
var clientesi = $("#clientes").val();
var datei = $("#iddatei").val();
var datef = $("#iddatef").val();
//alert(clientesi)
if(empresasi=='Seleccionar' || clientesi=='Seleccionar' || empresasi==" || clientesi==" || datei==" || datef==" ) {
    alert("Por favor rellene todos los campos.");
    return false;
}
$.ajax({
    type: "POST",
    url: "Datos_historicos2",
    data: {
        empresasi: empresasi,
        clientesi: clientesi,
        datei: datei,
        datef: datef
    },
    cache: false,
    success: function(data) {
        var tempO = data.temperatura;
        tempS = JSON.stringify(tempO);
        //alert(tempS);
        const tempA = tempS,
        tempC = tempA.match(/\d+(?:\.\d+)?/g).map(Number)
        var denO = data.densidad;
        denS = JSON.stringify(denO);
        //alert(denS);
        const denA = denS,
        denC = denA.match(/\d+(?:\.\d+)?/g).map(Number)
```



```
var phO = data.ph;
phS = JSON.stringify(phO);
//alert(phS);
const phA = phS,
phC = phA.match(/\d+(?:\.\d+)?/g).map(Number)
var dateO = data.fecha;
dateS = JSON.stringify(dateO);
var dateV = dateS;
var dateC = dateV.split(',');
//alert(components);
//alert(typeof(tempS));
Highcharts.chart('container', {
title: {
  text: 'Historicos medicion Calidad de la leche'
},
subtitle: {
  text: 'Carlos Montoya'
},
yAxis: {
  title: {
    text: 'Niveles de ph,densidad,temperatura'
  }
},
xAxis: [{
  categories: dateC,
  labels: {
    rotation: 45,
    align: 'left'
  },
  title: {
```



```
        text: 'Fecha'
      }
    },
    {
      tickInterval: 1,
      type: 'categories',
      title: {
        text: 'Numero de muestra'
      },
      opposite: true
    }
  ],
  legend: {
    layout: 'vertical',
    align: 'right',
    verticalAlign: 'middle'
  },
  plotOptions: {
    series: {
      label: {
        connectorAllowed: false
      },
      pointStart: 0
    }
  },
  series: [{
    name: 'Temperatura',
    data: tempC
  }, {
    name: 'Densidad',
    data: denC
  }
]
```



```
}, {  
  name: 'Niveles de PH',  
  data: phC  
}],  
exporting: {  
  menuItemDefinitions: {  
    "downloadXLS": {  
      text: 'Descargar EXCEL'  
    },  
    "viewFullscreen": {  
      text: 'Pantalla Completa'  
    },  
    "printChart": {  
      text: 'Imprimir'  
    },  
    "downloadPDF": {  
      text: 'Descargar PDF'  
    }  
  },  
  buttons: {  
    contextButton: {  
      menuItems: ["viewFullscreen", "separator", "printChart",  
"downloadPDF", "downloadXLS"]  
    }  
  }  
},  
  
responsive: {  
  rules: [{  
    condition: {
```



```
        maxWidth: 500
    },
    chartOptions: {
        legend: {
            layout: 'horizontal',
            align: 'center',
            verticalAlign: 'bottom'
        }
    }
}
}
}

});

document.getElementById('idParent').innerHTML = components[2];
},
error: function(xhr, status, error) {
    console.error(xhr);
}
});
});
});
</script>
<script type="text/javascript">
</script>

<!-- Content Wrapper. Contains page content -->
<script src="<?php echo
base_url('bower_components/jquery/dist/jquery.min.js')?>"></script>
<!-- Bootstrap 3.3.7 -->
<script src="<?php echo
base_url('bower_components/bootstrap/dist/js/bootstrap.min.js')?>"></script>
```

```
<!-- iCheck -->
```

```
<script src="<?php echo base_url('plugins/iCheck/ichack.min.js')?>"></script>
```

```
<div class="content-wrapper">
```

```
<!-- Content Header (Page header) -->
```

```
<section class="content">
```

```
<!-- MIREN QUE POR CADA FILA CON CONTENIDO PORNGO UN "ROW" ESTE ES EL ROW QUE CONTIENE EL FORM -->
```

```
<div class="row">
```

```
<div class="col col">
```

```
<div class="box box-info">
```

```
<div class="box-header with-border">
```

```
<h3 class="box-title">Nueva Empresa</h3>
```

```
</div>
```

```
<!-- /.box-header -->
```

```
<!-- form start -->
```

```
<form class="form-horizontal" method="POST" action="<?php echo base_url('Empresas/recibe_empresas')?>">
```

```
<div class="box-body">
```

```
<div class="form-group">
```

```
<label for="inputEmail3" class="col-sm-2 control-label">Nombre</label>
```

```
<div class="col-sm-10">
```

```
<input id="nombre" type="text" class="form-control" id="E1" placeholder="Ingrese nombre" name="nombre">
```

```
</div>
```

```
</div>
```

```
<div class="form-group">
```

```
<label for="inputEmail3" class="col-sm-2 control-label">Dirección</label>
```

```
<div class="col-sm-10">
```



```
<input id="direccion" type="text" class="form-control" id="E2"
placeholder="Ingrese dirección" name="direccion">
</div>
</div>

<div class="form-group">
<label for="inputEmail3" class="col-sm-2 control-label">Teléfono</label>
<div class="col-sm-10">
<input id="telefono" type="number" class="form-control" id="E3"
placeholder="Ingrese teléfono empresa" name="telefono">
</div>
</div>

<div class="form-group">
<label for="inputEmail3" class="col-sm-2 control-label">Email</label>
<div class="col-sm-10">
<input id="mail" type="email" class="form-control" id="E4"
placeholder="Ingrese email empresa" name="mail">
</div>
</div>

</div>
<!-- /.box-body -->
<div class="box-footer">

<input id="submit" type="button" class="btn btn-primary pull-right"
value="Registrar" />
</div>
<!-- /.box-footer -->
</form>
</div>
</div>
```



```
</div>
<div class="row">
<div class="col">
  <div class="box box-info box-solid">
    <div class="box-header with-border">
      <h3 class="box-title">Empresas</h3>
      <div class="box-tools pull-right">
        <button type="button" class="btn btn-box-tool" data-widget="remove"><i
class="fa fa-times"></i></button>
      </div>
      <!-- /.box-tools -->
    </div>
    <!-- /.box-header -->
    <div class="box-body">

      <?php

        foreach ($empresas as $empresa) {
          echo " Empresa: ".$empresa['Nom_empresa']."<br>".<br>";
        }
      ?>
    </div>
    <!-- /.box-body -->
  </div>
  <!-- /.box -->
</div>
</div>

</section>
</div>
<script>
```



```
$(document).ready(function() {  
    $("#submit").click(function() {  
        var nombre = $("#nombre").val();  
        var direccion = $("#direccion").val();  
        var telefono = $("#telefono").val();  
        var mail = $("#mail").val();  
        //alert(clientesi);  
        if(nombre == ""||direccion == ""||telefono == ""||mail == "") {  
            alert("Por favor rellene todos los campos.");  
            return false;  
        }  
        $.ajax({  
            type: "POST",  
            url: "Empresas/recibe_empresas",  
            data: {  
                nombre: nombre,  
                direccion: direccion,  
                telefono: telefono,  
                mail: mail  
            },  
            cache: false,  
            success: function(data) {  
                if (data=='1')  
                {  
                    alert('Empresa creada con exito');  
                    window.location.assign("Empresas");  
                } else if (data=='existe') {  
                    alert('Empresa ya existe, verifique los datos , por favor');  
                }  
            }  
        });  
    });  
});
```

```

    },
    error: function(xhr, status, error) {
        console.error(xhr);
    }
});
});
});
</script>
<!-- Content Wrapper. Contains page content -->
<script src="<?php echo
base_url('bower_components/jquery/dist/jquery.min.js')?>"></script>
<link rel="stylesheet" href="<?php echo base_url() ?>bower_components/bootstrap-
toggle/css/bootstrap-toggle.min.css" />
<script src="<?php echo base_url('bower_components/bootstrap-toggle/js/bootstrap-
toggle.min.js')?>"></script>
<script src="<?php echo base_url('bower_components/highcharts/Highcharts-
9.2.2/code/highcharts.js')?>"></script>
<script src="<?php echo base_url('bower_components/highcharts/Highcharts-
9.2.2/modules/exporting.js')?>"></script>
<script src="<?php echo base_url('bower_components/highcharts/Highcharts-
9.2.2/modules/export-data.js')?>"></script>
<script src="<?php echo base_url('bower_components/highcharts/Highcharts-
9.2.2/modules/accessibility.js')?>"></script>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.0/jquery.min.js"></script>
<div class="content-wrapper">
<!-- Content Header (Page header) -->
<section class="content-header">
<h1>
    Mediciones
    <small>Panel de control</small>
</h1>
<div class="row">
    <div class="col col">

```



```
<div class="box box-info">
  <div class="box-header with-border">
    <h3 class="box-title">Toma de datos</h3>
  </div>
  <!-- /.box-header -->
  <!-- form start -->
  <form class="form-group has-feedback" method="POST" action="<?php echo
base_url('Prueba2')?>">
    <div class="box-body">

      <div class="form-group">
        <label for="inputPassword3" class="col-sm-2 control-label">Empresa</label>
        <div class="col-sm-10">
          <select id="empresasi" class="form-control" name="empresas"
onchange="dl(this)">

            <?php

              foreach ($empresas as $i) {
                echo '<option value="'. $i->ID_empresa .'>'. $i-
>Nom_empresa .'</option>';
              }
            ?>
            <option selected>Seleccionar</option>
          </select>
        </div>
      </div>

      <div class="form-group">
        <label for="inputPassword3" class="col-sm-2 control-label">Cliente</label>
        <div class="col-sm-10">
```

```
<select id="clientes" class="form-control" name="clientes"
onchange="d1(this)">
```

```
<option selected>Seleccionar</option>
```

```
</select>
```

```
</div>
```

```
</div>
```

```
<div class="form-group">
```

```
<label for="inputPassword3" class="col-sm-2 control-label">Tipo de
muestra</label>
```

```
<div class="col-sm-10">
```

```
<select id="liquidosi" class="form-control" name="liquidos"
onchange="d1(this)">
```

```
<option selected>Seleccionar</option>
```

```
<option value="vino">Vino</option>
```

```
<option value="limon">Limon</option>
```

```
<option value="agua">Agua</option>
```

```
<option value="leche">Leche</option>
```

```
</select>
```

```
</div>
```

```
</div>
```

```
<div class="form-group">
```

```
<label for="inputEmail3" class="col-sm-2 control-label">Tiempo</label>
```

```
<div class="col-sm-10">
```

```
<input id="tiempo" type="number" class="form-control" id="C1"
placeholder="Ingrese tiempo de toma de datos en segundos" name="Temporizador">
```

```
</div>
```

```
</div>
```

```
<div class="progress">
```

```
<div class="progress-bar w-75" role="progressbar" aria-valuenow="75" aria-
valuemin="0" aria-valuemax="100"></div>
```

</div>

</div>

<div class="box-footer">

<!-- <button type="submit" class="btn btn-primary pull-right">Iniciar</button-->

<input id="submit" type="button" class="btn btn-primary pull-right" value="Iniciar" />

</div>

</form>

</div>

</div>

</div>

<ol class="breadcrumb">

<li><a href="pinicio"><i class="fa fa-dashboard"></i> Home</a></li>

<li class="active">Mediciones</li>

</ol>

</section>

<section class="content" id=idParent>

</section>

<figure class="highcharts-figure">

<div id="container2"></div>

<p class="highcharts-description">

</p>

</figure>

```
<script src="<?php echo  
base_url('bower_components/jquery/dist/jquery.min.js')?>"></script>
```

```
<script src="<?php echo base_url('bower_components/jquery-ui/jquery-ui.js')?>"></script>
```

```
<script src="<?php echo base_url('bower_components/jquery-ui/jquery-ui.js')?>"></script>
```

```
<script type="text/javascript">
```

```
$(document).ready(function() {
```

```
$("#empresasi").change(function() {
```

```
$("#empresasi option:selected").each(function() {
```

```
empresas = $('#empresasi').val();
```

```
$.post("<?php echo base_url(); ?>Toma_de_datos/fill_clientes", {
```

```
empresasi : empresas
```

```
}, function(data) {
```

```
$("#clientes").html(data);
```

```
});
```

```
});
```

```
});
```

```
});
```

```
</script>
```

```
<!-- /.content -->
```

```
<script>
```

```
$(document).ready(function() {
```

```
$("#submit").click(function() {
```

```
var empresasi = $('#empresasi').val();
```

```
var clientes = $('#clientes').val();
```

```
var liquidosi = $("#liquidosi").val();  
  
var tiempo = $("#tiempo").val();  
  
if(empresasi=='Seleccionar' || clientesi=='Seleccionar' || liquidosi=='Seleccionar'  
|| empresasi=='' || clientesi=='' || liquidosi=='' || tiempo=='') {  
  
    alert("Por favor rellene todos los campos.");  
  
    return false;  
  
}  
  
$.ajax({  
    type: "POST",  
    url: "toma_de_datos2",  
    data: {  
        empresasi: empresasi,  
        clientesi: clientesi,  
        liquidosi: liquidosi,  
        tiempo: tiempo  
    },  
    <?php $time = "<script>document.write(tiempo)</script>"?>  
    cache: false,  
    success: function(data) {  
        document.getElementById('idParent').innerHTML = data;  
        var timeleft = tiempo  
        var downloadTimer = setInterval(function(){  
            if(timeleft <= 0){  
                clearInterval(downloadTimer);  
                document.getElementById("countdown").innerHTML = "Finalizado";  
                document.getElementById("guardar").disabled = false;  
            }  
        }  
    }  
}
```

```
else{
    document.getElementById("countdown").innerHTML = timeleft + " Segundos
restantes";
    document.getElementById("guardar").disabled = true;
}
document.getElementById("progressBar").value = tiempo - timeleft;

timeleft -= 1;
}, 1000);

},
error: function(xhr, status, error) {
console.error(xhr);
}
});
//aqui va codigo
Highcharts.chart('container2', {
    chart: {
        type: 'spline',
        animation: Highcharts.svg, // don't animate in old IE
        marginRight: 10,
        events: {
            load: function () {

                // set up the updating of the chart each second
                var series = this.series[0];
                var series2 = this.series[1];
                var series3 = this.series[2];
                setInterval(function () {
                    var x = (new Date()).getTime(), // current time
                        y = t;
```



```
        z = ph;
        w = d;
        series.addPoint([x, y], true, true);
        series2.addPoint([x, z], true, true);
        series3.addPoint([x, w], true, true);
    }, 2000);
}
}
},

time: {
    useUTC: false
},

title: {
    text: 'Datos Sensores'
},

accessibility: {
    announceNewData: {
        enabled: true,
        minAnnounceInterval: 15000,
        announcementFormatter: function (allSeries, newSeries, newPoint) {
            if (newPoint) {
                return 'New point added. Value: ' + newPoint.y;
            }
            return false;
        }
    }
},
```



```
xAxis: {  
  type: 'datetime',  
  tickPixelInterval: 150  
},
```

```
yAxis: [{  
  title: {  
    text: 'Temperatura'  
  },  
  plotLines: [{  
    value: 0,  
    width: 1,  
    color: '#808080'  
  }]  
},
```

```
{  
  title: {  
    text: 'pH'  
  },  
  plotLines: [{  
    value: 0,  
    width: 1,  
    color: '#808080'  
  }]  
},
```

```
{  
  title: {  
    text: 'Densidad'  
  },  
},
```



```
plotLines: [{
  value: 0,
  width: 1,
  color: '#808080'
}]
}],
tooltip: {
  headerFormat: '<b>{series.name}</b><br/>',
  pointFormat: '{point.x:%Y-%m-%d %H:%M:%S}<br/>{point.y:.2f}'
},

legend: {
  enabled: false
},

exporting: {
  enabled: false
},

series: [{
  name: 'Temperatura',
  data: (function () {
    // generate an array of random data
    var data = [],
        time = (new Date()).getTime(),
        i;
    for (i = -19; i <= 0; i += 1) {
      data.push({
        x: time + i * 1000,
        y: 0
      })
    }
  })()
}]
}
```



```
        });  
    }  
    return data;  
}()  
,  
{  
    name: 'Nivel de pH',  
    data: (function () {  
        // generate an array of random data  
        var data = [],  
            time = (new Date()).getTime(),  
            i;  
        for (i = -19; i <= 0; i += 1) {  
            data.push({  
                x: time + i * 1000,  
                y: 0  
            });  
        }  
        return data;  
    })()  
,  
{  
    name: 'Densidad',  
    data: (function () {  
        // generate an array of random data  
        var data = [],  
            time = (new Date()).getTime(),  
            i;  
        for (i = -19; i <= 0; i += 1) {
```



```
        data.push({
            x: time + i * 1000,
            y: 0
        });
    }
    return data;
}())

    ]
});

});

var t;
var ph;
var d;

window.onload = function() {

var ultimos_valores_th = function () {
$.ajax({
    url: "<?php echo base_url('Controlador_general_cleche/ultimos_valores')?>",
    data: {

    },
    cache: false,
    type: "GET",
    success: function(response) {
        console.log(response);
        var objeto_th = JSON.parse(response);
        console.log(objeto_th.temperatura);
```



```
t = parseFloat(objeto_th.temperatura);
    if (objeto_th.temperatura>15){
        $("#pantalla_dato1_th" ).html(objeto_th.temperatura);
        $("#pantalla_dato2_th" ).html("Leche Caliente");
    }
    else {
        $("#pantalla_dato1_th" ).html(objeto_th.temperatura);
        $("#pantalla_dato2_th" ).html("Leche Fria");
    }
},
error: function(xhr) {
}
});

setTimeout(ultimos_valores_th, 1000);
};
setTimeout(ultimos_valores_th, 100);

var ultimos_valores_ph = function () {
$.ajax({
    url: "<?php echo base_url('Controlador_general_cleche/ultimos_valores')?>",
    data: {

    },
    cache: false,
    type: "GET",
    success: function(response) {
        console.log(response);
        var objeto_ph = JSON.parse(response);
```



```
console.log(objeto_ph.ph);

console.log(objeto_ph.r_ph);

ph = parseFloat(objeto_ph.ph);

    $("#pantalla_dato1_ph" ).html(objeto_ph.ph);
    $("#pantalla_dato2_ph" ).html(objeto_ph.r_ph);
},
error: function(xhr) {

}

});

setTimeout(ultimos_valores_ph, 1000);

};

setTimeout(ultimos_valores_ph, 100);

var ultimos_valores_d = function () {
$.ajax({
    url: "<?php echo base_url('Controlador_general_cleche/ultimos_valores')?>",
    data: {
    },
    cache: false,
    type: "GET",
    success: function(response) {
        console.log(response);
        var objeto_d = JSON.parse(response);
        console.log(objeto_d.densidad);
        d = parseFloat(objeto_d.densidad);

        $("#pantalla_dato1_d" ).html(objeto_d.densidad);
        $("#pantalla_dato2_d" ).html(objeto_d.r_d);
```



```
    },  
    error: function(xhr) {  
    }  
});  
setTimeout(ultimos_valores_d, 1000);  
};  
setTimeout(ultimos_valores_d, 100);  
}  
</script>  
</div>
```

### Capa Controlador

```
<?php  
defined('BASEPATH') OR exit('No direct script access allowed');  
class Controlador_general_cleche extends CI_Controller {  
    public function __construct()  
{  
        parent::__construct();  
        $this->load->model('General_model_cleche');    //carga el modelo  
        $this->load->model('toma_de_datos_model');    //carga el modelo  
        $this->load->database();    //carga la base de datos  
        $this->load->library('session'); //carga la libreria de manejo de sesiones  
(variables)  
        $this->load->helper('date');  
    }  
    public function recibe_datos_sensores()  
    {  
        $controlIngreso = $this->toma_de_datos_model->estadoingresodatos();  
        $int_value = intval( $controlIngreso );  
        echo gettype($int_value);  
        echo $int_value;  
        if ($int_value == 1){
```

```

        $temperatura=$_POST['temperatura'];
        $densidad=$_POST['densidad'];
        $d_corregida=$_POST['d_corregida'];
        $ph=$_POST['ph'];
        $r_d=$_POST['r_d'];
        $r_ph=$_POST['r_ph'];
        $this->toma_de_datos_model-
>nuevo_dato_sensores($temperatura,$densidad,$ph,$r_d,$r_ph,$d_corregida);
    }
    else
    {
        echo 'sistema no recibe datos';
    }
}

public function ultimos_valores()
{
    //$user=$this->session->userdata('Id_usuario');
    //$temporales_id = '1';
    $return = $this->General_model_cleche->ultimos_valores();
    echo json_encode($return[0]); //saco como json el array
}
}

<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Datos_historicos extends CI_Controller {
    public function __construct()
    {
        parent::__construct();
        $this->load->model('General_model_cleche'); //carga el modelo
    }
}

```



```
$this->load->model('toma_de_datos_model');

$this->load->database();           //carga la base de datos

$this->load->library('session');   //carga la libreria de manejo de
sesiones (variables)

if ($this->session->userdata('Id_usuario')=="){

    echo "Sin autorización";

    redirect(base_url('Login'), 'refresh');

}

}

public function index()
{
    $usuario_id = $this->session->userdata('Id_usuario');
    $id_cliente="";
    $datei="";
    $datef="";

    $data['empresas'] = $this->toma_de_datos_model->get_empresas($usuario_id);
    $data['cliente'] = $this->toma_de_datos_model->get_clientes($id_cliente);
    $data['temperatura'] = $this->General_model_cleche-
>datos_historicos_temperatura($id_cliente,$datei,$datef);

    $data['densidad'] = $this->General_model_cleche-
>datos_historicos_densidad($id_cliente,$datei,$datef);

    $data['ph'] = $this->General_model_cleche-
>datos_historicos_ph($id_cliente,$datei,$datef);

    $this->load->view('head');

    $this->load->view('abre_body_wrapper');

    $this->load->view('header');

    $this->load->view('sidebar');

    $this->load->view('content_datos_historicos',$data);

    $this->load->view('footer');

    $this->load->view('cierra_wrapper');

    $this->load->view('scripts');
```

```

    }

    public function fill_clientes() {
        $empresas = $this->input->post('empresasi');
        echo "Hola";
        //die();
        if(!is_null($empresas)){
            $clientes = $this->toma_de_datos_model->get_clientes($empresas);
            echo '<option selected>Seleccionar</option>';
            foreach($clientes as $fila){
                echo '<option value="'. $fila->Id_cliente .' ">'. $fila->Nom_cliente .'</option>';
            }
        } else {
            echo '<option selected>Seleccionar8</option>';
        }
    }
}
}
<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Login extends CI_Controller {
    public function __construct()
    {
        parent::__construct();
        $this->load->model('Login_model');//carga el modelo
        $this->load->database(); //carga la
base de datos
        $this->load->library('session'); //carga la libreria de manejo de sesiones (variables)
        $this->load->library('encrypt'); //carga la libreria de encriptador
    }

    public function index()
    {
        $data['msg'] = ""; //esto se lo paso a la vista
    }
}

```

```

        $this->load->view('login',$data); //carga la vista login
    }

    public function hacerlogin(){
        $email = strip_tags($this->input->post('email')); //limpio y recibo Email
        $password = strip_tags(sha1($this->input->post('password'))); // limpio y recibo pass y la
        encripto al vuelo

        //echo $email; echo $password; //para debugear

        $return = $this->Login_model->login($email, $password); //le paso al modelo ambas para
        que me diga si puedo dejarlo entrar

        if ($return==1) {
            redirect(base_url('inicio'), 'refresh'); //si está todo ok redirecciono a tiemporeal
        }

        if ($return==0) { //si hay algo mal, recargo el login pasandole a la vista el mensaje
        correspondiente

            $data['msg'] = 'Credenciales inválidas'; //esto se lo paso a la vista

            $this->load->view('login', $data);

        }

    }
}

<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class inicio extends CI_Controller {
    public function __construct()
    {
        parent::__construct();

        $this->load->database(); //carga la base de datos (me conecto)

        $this->load->library('session'); //carga la libreria de manejo de sesiones
        (variables)

        if ($this->session->userdata('Id_usuario')=="){
            echo "Sin autorización";
        }
    }
}

```

```

        redirect(base_url('Login'), 'refresh');
    }
}

public function index()
{
    $this->load->view('head');
    $this->load->view('abre_body_wrapper');
    $this->load->view('header');
    $this->load->view('sidebar');
    $this->load->view('content_pinicio');
    $this->load->view('footer');
    $this->load->view('cierra_wrapper');
    $this->load->view('scripts');
}
}

<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Register extends CI_Controller {

    public function __construct()
    {
        parent::__construct();
        $this->load->model('Register_model');    //carga el modelo
        $this->load->database();                //carga la base de datos
        $this->load->library('session');        //carga la libreria de manejo de sesiones
(variables)
        $this->load->library('encrypt');        //carga la libreria de encriptador
    }

    public function index()
    {

```



```
$data['msg'] = "";
    $this->load->view('register',$data); //carga la vista login
}
public function hacerregister(){
    $usuario = $_POST['usuario']; //recibo y limpio usuario
    $email = $_POST['email']; //recibo y limpio email
    $password = $_POST['password']; // recibo limpio pass
    $rpassword = $_POST['passwordr']; //recibo y limpio rpass
    $direccion = $_POST['direccion']; // recibo limpio direccion
    $telf=$_POST['telefono'];// recibo limpio telf
    $return= $this->Register_model->register($usuario,$email, $password,$direccion,$telf);
    header('Content-Type: application/json; charset=utf-8');
    echo json_encode($return);
}
}
```

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class toma_de_datos extends CI_Controller {
public function __construct()
{
    parent::__construct();
    $this->load->model('toma_de_datos_model');
    $this->load->database(); //carga la base de datos (me conecto)
    $this->load->library('session'); //carga la libreria de manejo de sesiones
(variables)
    if ($this->session->userdata('Id_usuario')=="){
        echo "Sin autorización";
        redirect(base_url('Login'), 'refresh');
    }
}
```

```

}

public function index()    {
    $usuario_id = $this->session->userdata('Id_usuario');
    $this->toma_de_datos_model->cierraingresodatos();
    $id_cliente="";
    $data['empresas'] = $this->toma_de_datos_model->get_empresas($usuario_id);
    $data['cliente'] = $this->toma_de_datos_model->get_clientes($id_cliente);
    $this->load->view('scripts');
    $this->load->view('head');
    $this->load->view('abre_body_wrapper');
    $this->load->view('header');
    $this->load->view('sidebar');
    $this->load->view('content_toma_de_datos',$data);
    $this->load->view('footer');
    $this->load->view('cierra_wrapper');
}

public function fill_clientes() {

$empresas = $this->input->post('empresasi');
//die();
if(!is_null($empresas)){
    $clientes = $this->toma_de_datos_model->get_clientes($empresas);
    echo '<option selected>Seleccionar</option>';
    foreach($clientes as $fila){
        echo '<option value="'. $fila->Id_cliente .' ">'. $fila->Nom_cliente .'</option>';
    }
} else {
    echo '<option selected>Seleccionar</option>';
}
}
}

```

}

## *Anexo 2: Código Fuente del Módulo Electrónico de Arduino*

SSD1306 128x32 I2C test

Connect display Vcc to 5V and GND to ground

Connect display I2C SCL to MCU SCL and SDA to MCU SDA

```
#include <Wire.h>
```

```
#include <Adafruit_GFX.h>
```

```
#include <Adafruit_SSD1306.h>
```

```
/******
```

```
#include <WiFi.h>
```

```
#include <HTTPClient.h>
```

```
/*+++++  
+++*/
```

```
#include "HX711.h"
```

```
const char* ssid = "Galaxy A3283ED"; /*Red del localhost*/
```

```
const char* password = "ejmz5316"; /*contraseña red*/
```

```
const char* serverName =
```

```
"http://192.168.144.179/Controlador_general_cleche/recibe_datos_sensores"; /*Controlador  
de destino para toma de datos*/
```

```
/******
```

```
// Librerías sensor de temperatura */
```

```
#include <OneWire.h>
```

```
#include <DallasTemperature.h>
```

```
/******
```

```
/* Pin de conexión del sensor de temperatura */
```

```
#define ONE_WIRE_BUS 26
```

```
/******
```

HX711 balanza;

/\*\*\*\*\*\*  
/

// Setup a oneWire instance to communicate with any OneWire devices

// (not just Maxim/Dallas temperature ICs)

OneWire oneWire(ONE\_WIRE\_BUS);

/\*\*\*\*\*\*  
/

// Pass our oneWire reference to Dallas Temperature.

DallasTemperature sensors(&oneWire);

/\*\*\*\*\*\*  
/

#define SCREEN\_WIDTH 128 // OLED display width, in pixels

#define SCREEN\_HEIGHT 64 // OLED display height, in pixels

// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)

Adafruit\_SSD1306 display(SCREEN\_WIDTH, SCREEN\_HEIGHT, &Wire);

// Initialization

float calibration\_value = 16.20;

int phval = 0;

unsigned long int avgval;

int buffer\_arr[10],temp;

void setup() {

  Serial.begin(9600);

  WiFi.begin(ssid, password);

  Serial.println("Connecting");

  while(WiFi.status() != WL\_CONNECTED) {

    delay(500);

    Serial.print(".");

  }

```
Serial.println("");  
Serial.print("Connected to WiFi network with IP Address: ");  
Serial.println(WiFi.localIP());  
Serial.println("Timer set to 5 seconds (timerDelay variable), it will take 5 seconds before  
publishing the first reading.");
```

```
balanza.begin(19, 18);  
Serial.print("Lectura del valor del ADC: ");  
Serial.println(balanza.read());  
Serial.println("No ponga ningun objeto sobre la balanza");  
Serial.println("Destarando...");  
Serial.println("...");  
balanza.set_scale(407386.364); // Establecemos la escala  
balanza.tare(20); //El peso actual es considerado Tara.  
Serial.println("Listo para pesar");
```

```
Serial.println("Dallas Temperature IC Control Library Demo");  
// Start up the library  
sensors.begin();  
// SSD1306_SWITCHCAPVCC = generate display voltage from 3.3V internally  
if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) { // I2C Address 0x3C for 128x32  
  Serial.println(F("SSD1306 allocation failed"));  
  for(;;); // Don't proceed, loop forever  
}  
// Clear the buffer, set small font size and print 4 lines of text to display  
display.clearDisplay();  
display.setTextSize(1); // Set for small font  
display.setTextColor(WHITE);  
display.setCursor(0,0); // Set cursor to upper left  
display.println("U. CATOLICA DE CUENCA");  
display.println("");
```

```
display.println("PROTOTIPO MEDICION"); // Println moves the cursor down
display.println("CALIDAD DE LA LECHE");
display.println("");
display.setCursor(20,40);
display.println("CARLOS MONTOYA");

display.display();
delay (5000);
}
void loop() {

    for(int i=0;i<10;i++)
    {
        buffer_arr[i]=analogRead(A0);
        delay(30);
    }
    for(int i=0;i<9;i++)
    {
        for(int j=i+1;j<10;j++)
        {
            if(buffer_arr[i]>buffer_arr[j])
            {
                temp=buffer_arr[i];
                buffer_arr[i]=buffer_arr[j];
                buffer_arr[j]=temp;
            }
        }
    }
    avgval=0;
    for(int i=2;i<8;i++)
```



```
avgval+=buffer_arr[i];

float volt=(float)avgval*3.3/4095/6; // 4095 es la toma de datos de la placa esp

float ph_act = -5.70 * volt + calibration_value;

/*****/

Serial.print(" Requesting temperatures...");

sensors.requestTemperatures(); // Send the command to get temperature readings

Serial.println("DONE");

Serial.print("Temperature is: ");

float temperatura= sensors.getTempCByIndex(0);

/*****/

Serial.print(sensors.getTempCByIndex(0)); // Why "byIndex"?

Serial.print("Peso: ");

Serial.print(balanza.get_units(20),3);

Serial.println(" kg");

float gramos = -(balanza.get_units(20))*1000;

float densidad= ((gramos-2)/(12));/*12.5ml es la medida que se tomará de cada muestra -
2 es el peso despreciable del recipiente*/

float ph = ph_act;

float d_corregida;

String r_d;

if (temperatura> 15){

    d_corregida = densidad + 0.0002*(temperatura-15);

}

else if (temperatura< 15){

    d_corregida = densidad - 0.0002*(15-temperatura);

}

if (d_corregida >= 1.028 && d_corregida <= 1.033){

    r_d="Leche Normal";

}

}
```



```
else if (d_corregida < 1.028){
    r_d="Leche Ácida";
}
else if (d_corregida > 1.033){
    r_d="Leche Descremada";
}
else
{
    r_d="Sin datos";
}

display.setCursor(0,0);
//display.drawPixel(x,y, color)
display.clearDisplay();
display.setTextSize(1);
display.setCursor(20,0);
display.println("DATOS OBTENIDOS");
display.println("");
display.print("Ph: ");display.println(ph_act);
display.print("TEMP:");display.print(temperatura);display.println(" C");
display.print("DENSIDAD: ");display.print(densidad);display.println(" g/ml");
display.print("D Corregida: ");display.print(d_corregida);display.println(" g/ml");
display.println("");
display.println("Toma de datos");
display.display();
String r_ph;

if (ph>= 1.10 && ph <= 1.80){
    r_ph="Leche Calostrá";
} else if (ph>= 1.90 && ph <= 6.30) {
    r_ph="Leche Ácida";
```



```
}  
else if (ph>= 6.40 && ph <= 6.80){  
    r_ph="Leche Normal";  
}  
else if (ph>= 6.90 && ph <= 14){  
    r_ph="Leche Mastitica";  
}  
else {  
    r_ph="Sin datos";  
}  
String datos = "temperatura=";  
datos += temperatura;  
datos += "&densidad=";  
datos += densidad;  
datos += "&d_corregida=";  
datos += d_corregida;  
datos += "&ph=";  
datos += ph;  
datos += "&r_ph=";  
datos += r_ph;  
datos += "&r_d=";  
datos += r_d;  
//Check WiFi connection status  
if(WiFi.status()== WL_CONNECTED){  
    WiFiClient client;  
    HTTPClient http;  
    // Your Domain name with URL path or IP address with path  
    http.begin(client, serverName);  
  
    // Specify content-type header
```



```
http.addHeader("Content-Type", "application/x-www-form-urlencoded");  
  
// Data to send with HTTP POST  
String httpRequestData = datos;  
  
// Send HTTP POST request  
int httpResponseCode = http.POST(httpRequestData);  
  
Serial.print("HTTP Response code: ");  
  
Serial.println(httpResponseCode);  
  
String payload = http.getString();  
  
Serial.println(payload);  
  
    // Free resources  
    http.end();  
  
}  
  
else {  
    Serial.println("WiFi Disconnected");  
  
}  
  
delay (1000);  
  
}
```

### *Anexo 3: Código Fuente de la Base de Datos*

```
-- phpMyAdmin SQL Dump
-- version 5.1.1
-- https://www.phpmyadmin.net/
--
-- Servidor: 127.0.0.1
-- Tiempo de generación: 03-02-2022 a las 21:47:54
-- Versión del servidor: 10.4.21-MariaDB
-- Versión de PHP: 8.0.10

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";
-- Base de datos: `cleche`
-- Estructura de tabla para la tabla `cliente`

CREATE TABLE `cliente` (
  `Id_cliente` int(10) NOT NULL,
  `Nom_cliente` varchar(50) NOT NULL,
  `Dir_cliente` varchar(50) NOT NULL,
  `Telf_cliente` int(10) NOT NULL,
  `Email_cliente` varchar(50) NOT NULL,
  `Empresa_id` int(10) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- Estructura de tabla para la tabla `control`
CREATE TABLE `control` (
```

```
`Id_control` int(10) NOT NULL,  
`Cliente_id` int(10) NOT NULL,  
`Sph_id` int(10) NOT NULL,  
`Sensores_id` int(10) NOT NULL,  
`Fecha` timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE  
current_timestamp(),  
`Temperatura` float(4,2) NOT NULL,  
`Densidad` float(4,2) NOT NULL,  
`Nivel_ph` float(4,2) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

-- Estructura de tabla para la tabla `control\_entrada`

```
CREATE TABLE `control_entrada` (  
  `controlIngresoId` int(10) NOT NULL,  
  `control_ingreso` int(10) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

-- Estructura de tabla para la tabla `empresa`

```
CREATE TABLE `empresa` (  
  `ID_empresa` int(10) NOT NULL,  
  `Nom_empresa` varchar(50) NOT NULL,  
  `Dir_empresa` varchar(50) NOT NULL,  
  `Telf_empresa` int(10) NOT NULL,  
  `Email_empresa` varchar(50) NOT NULL,  
  `Usuario_id` int(10) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Estructura de tabla para la tabla `sensores`

```
CREATE TABLE `sensores` (  
  `Id_sensores` int(10) NOT NULL,
```

`Nivel\_inferior` float(5,4) NOT NULL,

`Nivel\_superior` float(5,4) NOT NULL,

`Descripcion` text NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

Estructura de tabla para la tabla `sensores\_ph`

CREATE TABLE `sensores\_ph` (

`Id\_sph` int(10) NOT NULL,

`Nivel\_inferior` float(4,2) NOT NULL,

`Nivel\_superior` float(4,2) NOT NULL,

`Descripcion` text NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- Estructura de tabla para la tabla `temporales`

CREATE TABLE `temporales` (

`temporales\_id` int(10) NOT NULL,

`temperatura` float(4,2) NOT NULL,

`densidad` float(4,3) NOT NULL,

`ph` float(3,1) NOT NULL,

`r\_ph` varchar(50) NOT NULL,

`r\_d` varchar(50) NOT NULL,

`d\_corregida` float(4,3) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

Estructura de tabla para la tabla `usuario`

CREATE TABLE `usuario` (

`Id\_usuario` int(10) NOT NULL,

`Nom\_usuario` varchar(50) NOT NULL,

`Password` varchar(50) NOT NULL,

`Dir\_usuario` varchar(50) NOT NULL,



```
`Telf_usuario` int(11) NOT NULL,  
`Email_usuario` varchar(50) NOT NULL,  
`Fecha` timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE  
current_timestamp()  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```



**Carlos Reinaldo Montoya Ponce** portador(a) de la cédula de ciudadanía N° **0302855093**. En calidad de autor/a y titular de los derechos patrimoniales del trabajo de titulación **“Propuesta de un Diseño de un prototipo para la medición de la calidad de la leche”** de conformidad a lo establecido en el artículo 114 Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación, reconozco a favor de la Universidad Católica de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos y no comerciales. Autorizo además a la Universidad Católica de Cuenca, para que realice la publicación de éste trabajo de titulación en el Repositorio Institucional de conformidad a lo dispuesto en el artículo 144 de la Ley Orgánica de Educación Superior.

Cuenca, **24 de febrero de 2022**

F:  .....

**Carlos Reinaldo Montoya Ponce**

**C.I. 0302855093**