



Manual Técnico.

Programación de Sistemas Telemáticos

Msig. Adriana E. Collaguazo Jaramillo

Grupo: 8

Integrantes:

- Joseph Baquerizo
- Darwin Bermeo
- Claudia González
- Johan Loor
- Bolívar Núñez

MANUAL TÉCNICO

RESUMEN EJECUTIVO

En el siguiente proyecto se realizó el diseño de un sistema de control de aire regulados a través de una página web. Este sistema estará dirigido a los laboratorios de cómputo y de servidores de la empresa Adita S.A. el cual requiere que la temperatura sea la misma en todos los espacios de dicho laboratorio para así evitar que un lugar se encuentre más caliente o frío que otro, es decir que exista un equilibrio término dentro de los laboratorios. Para el diseño del sistema se hizo uso de módulos de sensores que permitieron recibir datos de la temperatura de un sector y luego estos mismo datos arrojados a una base datos de una página web, y para el control del aire se hizo uso de led infrarrojo para la manipulación de los sistemas de refrigeración de esa forma el usuario pueda personalizar el cambio de temperatura de manera remota de cada sector de la habitación y de esta forma lograr un control de temperatura de estos laboratorios.

DESCRIPCIÓN DEL PROBLEMA

Es común la disconformidad en una habitación por la temperatura a la que se encuentra, sea que esta es percibida como muy caliente o muy fría. Normalmente, esto es controlado a través de un aire acondicionado; sin embargo, esta no es regular en toda la habitación, por lo que personas en diferentes partes de la misma pueden tener opiniones diferentes de la temperatura en la habitación y de la temperatura que buscan, llevando a una discusión y reduciendo una interacción positiva entre sus ocupantes. Además, un ambiente demasiado frío puede provocar efectos negativos en la salud de los ocupantes, y un ambiente demasiado caliente puede provocar náuseas y desmayos, efectos también negativos al bienestar de la persona. Un ambiente de trabajo positivo es crucial en una empresa, optimizando los resultados obtenidos acerca de su productividad. En la empresa Adita S.A., específicamente en el área de mantenimiento y en los laboratorios de cómputo y de servidores, es necesario buscar un confort para sus empleados y que todos lleguen a un acuerdo.

OBJETIVOS ESPECÍFICOS

Los objetivos específicos que se plantearon para el siguiente proyecto fueron los siguientes:

- Crear un base de datos MySQL a partir del uso de la herramienta de software libre phpMyAdmin para tener un mejor control sobre los datos generados por los sensores implementados en el proyecto.
- Elaborar una página web con interfaz gráfica de usuario a partir de php, HTML y Java para el manejo de las funciones integradas del equipo regulador de temperatura a controlar.
- Establecer una conexión entre el microcontrolador Arduino con la base de datos MySQL mediante un módulo wi-Fi para la transferencia de datos almacenados.
- Obtener los códigos infrarrojos emitidos por controles enlazados al equipo regulador de temperatura mediante el módulo IR Receiver de Arduino para emitirlos como una señal propia del microcontrolador.

¿CÓMO FUNCIONA LA SOLUCIÓN?

Mediante la creación de una aplicación web implementar una interfaz gráfica que sea iterativa con el usuario que permita una fácil comprensión y el análisis de la temperatura de los laboratorios de cómputo y de sus servidores con el fin de mantener el respectivo control térmico evitando el sobrecalentamiento de los componentes y consumo incensario de energía. Para ello, se recolecta datos de temperatura y humedad en zonas estratégicas del lugar que serán almacenados en una base de datos y mostrados en el respectivo aplicativo web. El aplicativo web constara de una interfaz de para manipular la temperatura y otra para el análisis mediante gráficos de barras, lineal o circular. El sistema también debe tener un circuito Infrarrojo que permita la manipulación y control de forma remota del equipo refrigerativo.

¿QUÉ SE CONSTRUIRÁ PARA RESOLVER EL PROBLEMA?

Para la resolución del problema, se construirá una base de datos en MySQL manejada a partir de phpMyAdmin. Esta base servirá para almacenar los datos de temperatura y humedad adquiridos por los sensores implementados en el sistema por cada equipo utilizado para su detección.

También se construirá un circuito integrado con el microcontrolador Arduino que servirá para la recepción y emisión de señales infrarrojas pertenecientes a los equipos reguladores de temperatura con los que se desee trabajar.

Se elaborará una página web con una interfaz gráfica de usuario didáctica y entendible para el cliente. Esta página contará con ingreso de usuarios, gráficas de temperatura, sesión de control de temperatura e ingreso de nuevos equipos al sistema.

Además, se creará un circuito en donde participen los sensores capaces de detectar y enviar las señales de temperatura a la base de datos previamente descrita.

RECURSOS DE HARDWARE Y DE SOFTWARE

Hardware

Hardware	Cantidad
Módulo wifi esp8266	2
Arduino UNO	4
Modulo sensor de Temperatura	3
Módulo infrarrojo/receptor	1
Fuentes de poder 9V DC	5

Para el presente proyecto se pretende utilizar los siguientes dispositivos:

2 módulo esp8266, uno como receptor de información y el otro como emisor de ordenes hacia los dispositivos de enfriamiento.

4 módulos Arduino, tres de estos módulos serán los encargados de estar distribuidos en la habitación para el censado de temperatura. El cuarto módulo Arduino será el encargado de controlar la temperatura del ventilador.

3 módulos sensor de temperatura que serán acoplados a los 3 módulos de Arduino mencionado anteriormente.

Un módulo infrarrojo, conectado al módulo esp8266, para transmitir ordenes al cuarto módulo que será el que controle la temperatura del aire.

Software

El presente proyecto se va a implementar mediante el uso de herramientas que poseen software libre como Arduino y Modulo wifi Esp8266. El código programado en Arduino se basa en poder adquirir los datos de los sensores de temperaturas que se adjuntan al microcontrolador para así enviarlos al módulo wifi, aprovechando la conexión que tienen estos componentes. La finalidad del código utilizado en el módulo wifi esp8266 es enviar los datos a la nube mediante el uso del internet. Se aprovecharán el uso de los servidores HTTP de Apache para poder realizar la transferencia de datos hacia una base de datos relacional como lo es MySQL. A partir de los datos cargados en la nube se mostrarán gráficas con forma de mapeo de temperatura en una página web mediante el uso de HTML, CSS o Javascript, donde el usuario tendrá la oportunidad de ejecutar acciones y tomar decisiones según los datos mostrados. La orden que tome el usuario será recibida por un módulo wiFi enlazado con el microcontrolador Arduino que estará conectado a un control remoto infrarrojo. El Arduino recibirá la orden que ejecute el usuario a través de la página web, mandará una señal al Arduino para utilizar el control infrarrojo que mandará señales a los dispositivos con los que se encuentre configurado para poder cambiar la temperatura o velocidad del aire.

IMPLEMENTACIÓN DEL PROYECTO

Diagrama Esquemático

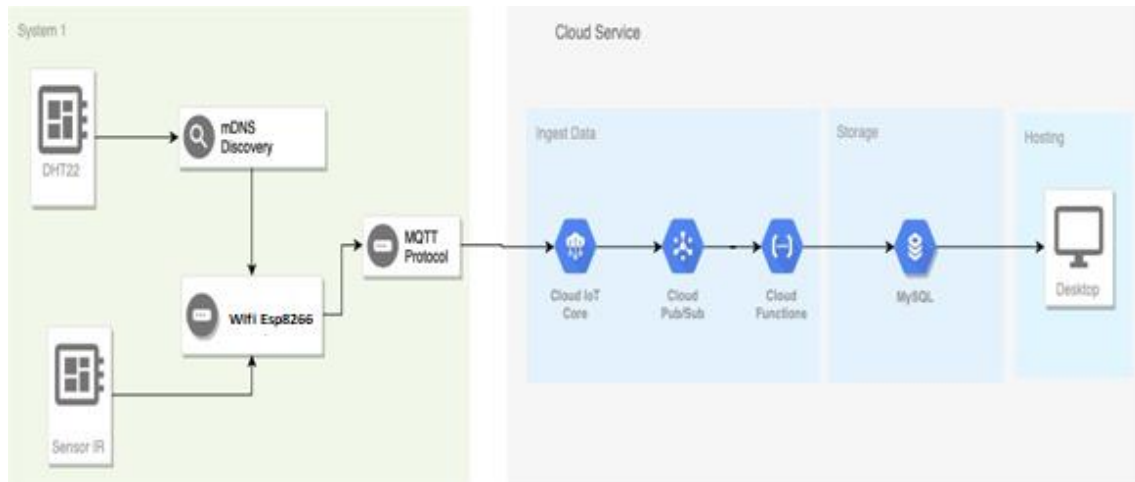


Diagrama de Despliegue

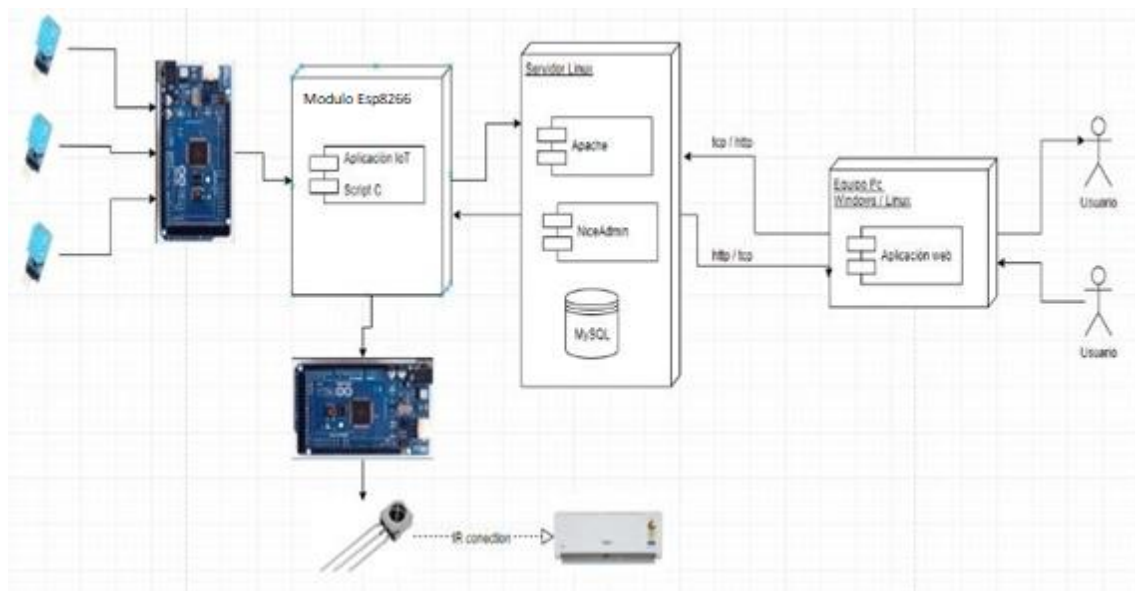
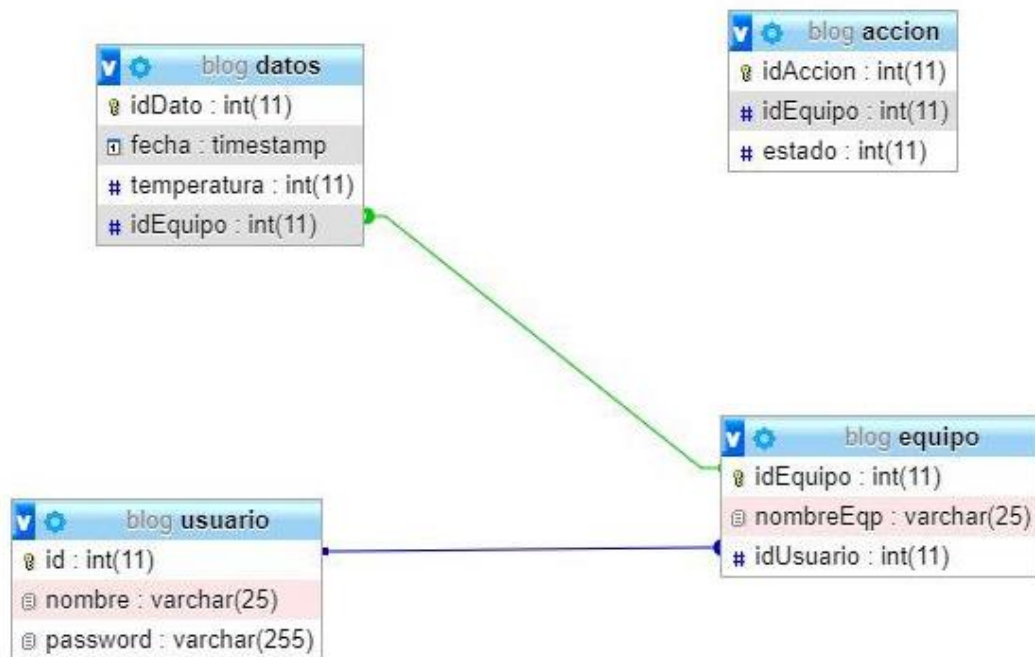
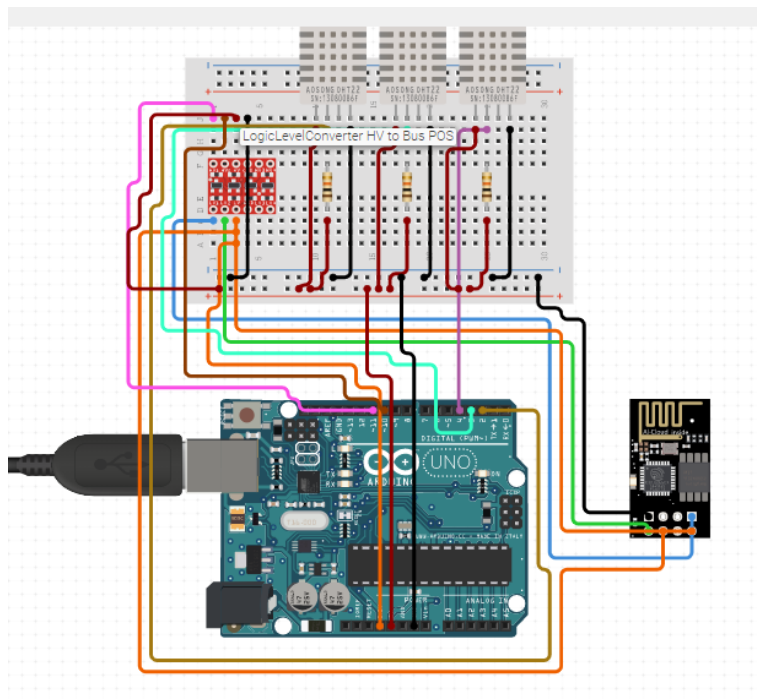


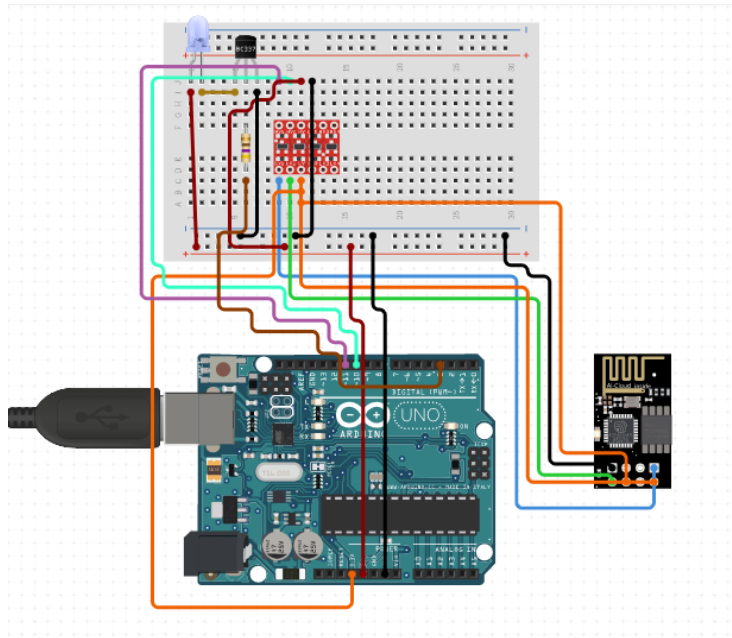
Diagrama Entidad-Relación.



Circuito para censar temperatura



Circuito para emitir señales infrarrojas



EXPLICACIÓN DE BASE DE DATOS MYSQL

Para la base de datos en MySQL se crearon 4 tablas con distintos campos, entre ellas se encuentra: datos, usuarios, equipos y acción.

La tabla de usuarios se encuentra conformada por 3 campos, los cuales son: idUsuario, nombre y password. El idUsuario sirve para poder otorgarle una clave única a cada usuario para poder distinguirlos durante el manejo de la base de datos en caso de que se desee extraer algún tipo de información de los mismos, es decir, es la primary key de dicha tabla de tipo int, ya que es incremental. Su nombre es el nombre de usuario que registraron en la pantalla principal de la página web y el password es la clave que deben ingresar junto a su nombre de usuario para poder validar que la persona que ingresa con dicho nombre es la correcta.

La tabla de datos se estructura con 4 campos: idDato, fecha, temperatura y equipo. El campo id es la primary key de los datos de tipo int e incremental, la cual sirve para diferenciarlos. La fecha es un dato tipo timestamp que sirve para poder saber el momento en el que se tomaron los datos que van a ser graficados en la página web, el cual incluye fecha y hora. La temperatura es un dato tipo int que se va a utilizar para poder graficar el comportamiento del ambiente censado por el sensor de temperatura adjuntado al Arduino.

Finalmente se tiene un foreign key que es el idEquipo, el cual nos sirve para relacionar las tablas creadas de la siguiente forma.

Un usuario puede registrar varios equipos, pero cada equipo tiene solo un usuario, por lo que podemos decir que si identificamos el equipo deseado, podemos saber quién es el usuario de dicho equipo. Además, los datos censados pertenecen a un solo equipo, pero un equipo puede censar varios datos, por lo que asociamos los datos tomados a su equipo correspondiente y evitar entrecruzamientos de datos con equipos distintos.

La tabla de equipos tiene 3 campo: idEquipo, nombreEqp y idUsuario. El idEquipo es una primary key de la tabla que permite identificar distintos y únicos equipos en el sistema. Se tiene nombreEqp que es el nombre del equipo que se está registrando al sistema y por último se tiene una foreign key que es idUsuario. La idUsuario nos permite hacer la relación entre la tabla de equipos y la tabla de usuarios para obtener la información de una manera más eficiente.

Finalmente tenemos la tabla de acción, la cual se conforma por 3 campos: idAccion, idEquipo y estado. El idAccion es una primary key de la tabla que nos permitirá distinguir cada acción realizada por el sistema. El idEquipo es un foreign key que nos permite relacionar las acciones ejecutadas de acuerdo a cada equipo implementado. El estado es un componente que nos indica las órdenes que se le ejecutan al equipo para que tome una acción, como encendido, apagado, entre otros, diferenciados numéricamente.

EXPLICACIÓN DE CÓDIGOS FUENTES

Se utilizó código php para realizar todas las validaciones, lecturas y escrituras en la base de datos creada de MySQL. Para poder colocar todos los elementos o componentes de la página web en su lugar correspondiente, se usó HTML junto con Bootstrap.

Además, utilizamos CSS para poder agregar o modificar componentes de tal forma que se incremente la estética de la misma página web. Con esto, se dieron distintos estilos y formatos para cada componente perteneciente a la interfaz gráfica.

Se realizó en la pantalla principal de la página un inicio de sesión en donde los usuarios pueden registrarse e inicializar su propia sesión. Este usuario tiene la posibilidad de

registrar su propio dispositivo o sensor con su respectivo código en el sistema, lo cual sería un registro a la base de datos.

Una vez que el usuario ha registrado su dispositivo, puede seleccionarlo para lograr observar una gráfica de los datos obtenidos por los respectivos sensores.

A continuación, se presenta el código utilizado para los sensores:

```
pst_temperatura_1.3
//conexion modulo ESP8266 al Arduino
//Vcc = 3.3V
//CH_PD = 3.3V
//GPIO = GND
//reset arduino = GND
//si se quiere enviar script, desconectar el GPIO de tierra una vez
//iniciado el modulo
//-----
/*****
    Include Libraries
    *****/
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <ESP8266WiFi.h>
#include <MySQL_Connection.h>
#include <MySQL_Cursor.h>
#include <ESP8266WiFiMulti.h>

/*****
    Define Instances and Constants
    *****/
#define DHTPIN 2
#define DHTTYPE DHT11
const char* WIFI_SSID = "SSID"; // Put here your Wi-Fi SSID
const char* WIFI_PASS = "PASSWORD"; // Put here your Wi-Fi password
DHT dht(DHTPIN, DHTTYPE);
/*
    Credenciales de MySQL
*/
IPAddress server_addr(127, 0, 0, 1); // IP of the MySQL *server* here
char user[] = "root"; // MySQL user login username
char password[] = "b4lt4z4r"; // MySQL user login password

WiFiClient client;
MySQL_Connection conn(&client);
```

pst_temperatura_1.3

```
ESP8266WiFiMulti WiFiMulti;
MySQL_Cursor* cur_mem;

/*
char INSERT_SQL[] = "INSERT INTO DB.table (fila) VALUES (valor)";
*/
char INSERT_SQL[] = "INSERT INTO blog.datos(idDato,temperatura,idEquipo) VALUES (%c,%f,%i)";
char query[128];
/*****
MODULO ESP8266 CONFIG
*****/
char* idDato = "";
//String fecha = "";
float temperature = 0.0;
int idEquipo = 0; // quemada en el equipo al "comprarlo"

/*****
Main Functions
*****/

void setup() {
  dht.begin();
  Serial.begin(115200);
  WiFi.mode(WIFI_STA); // se configura el modulo para trabajar como un ARduino
  WiFiMulti.addAP(WIFI_SSID, WIFI_PASS);
  Serial.println("/nConnected to network");
  while (WiFiMulti.run() != WL_CONNECTED) { // se comienza a realizar al conexion
    Serial.print("."); // a internet
    delay(500);
  }
  // si se logra establecer la conexion a internet, se imprime la ip
  if (WiFiMulti.run() == WL_CONNECTED) {
    Serial.println("Connected to network");
    IPAddress ip = WiFi.localIP();
    Serial.print("My IP address is: ");
  }
}
```

pst_temperatura_1.3

```
    Serial.print("My IP address is: ");
    Serial.println(ip);
}
else{
    Serial.print("ERROR AL CONECTAR A LA RED");
}

// intenta conectar al servido mysql
Serial.print("Connecting to SQL... ");
if (conn.connect(server_addr, 3306, user, password)) {
    Serial.println("OK.");
}
else {
    Serial.println("FAILED.");
}
// create MySQL cursor object
cur_mem = new MySQL_Cursor(&conn);
}

void loop() {
    temperature = dht.readTemperature();
    if (isnan(temperature)) { // se comprueba que el modulo DHT11 esta activo
        Serial.println(F("Error al leer el sensor DHT11")); //y funcionando
    }
    else { // si no hay error de lectura en el modulo DHT11 imprime valores en monitor serial
        Serial.print(F("Temperature: "));
        Serial.println(temperature);
        Serial.println("Values sent by the device");

        // envia los valores mediante el comando INSERT INTO de mysql a la base de datos especificada
        if (conn.connected()) {
            cur_mem->execute(INSERT_SQL);
            sprintf(query, INSERT_SQL, idDato, temperature, idEquipo); //*****
            cur_mem->execute(query);

```

Save Canceled.

```
        cur_mem->execute(query);
        delete cur_mem;
        Serial.println(INSERT_SQL);
    } else {
        Serial.print("Error al enviar a SQL");
    }
}

delay(5000);
}

```

Save Canceled.

A continuación, el código utilizado para la página web:

Creación de Usuarios

```
<?php
include_once 'app/config.inc.php';
include_once 'app/Conexion.inc.php';
include_once 'app/usuario.inc.php';
include_once 'app/RepositorioUsuario.inc.php';
include_once 'app/ValidadorRegistro.inc.php';
include_once 'app/Redireccion.inc.php';

/**
 * Se valida el usuario y contraseña con el que se desee registrar el usuario
 */

if (isset($_POST['enviar'])) {
    Conexion :: abrir_conexion();

    $validador = new ValidadorRegistro($_POST['nombre'],
        $_POST['clave1'], $_POST['clave2'], Conexion :: obtener_conexion());

    if ($validador -> registro_valido()) {

        $usuario = new usuario('', $validador-> obtener_nombre(),
            password_hash($validador -> obtener_clave(), PASSWORD_DEFAULT));

        $usuario_insertado = RepositorioUsuario :: insertar_usuario(Conexion ::
            obtener_conexion(), $usuario);

        if ($usuario_insertado) {
```

```
        Redireccion::redirigir(RUTA_LOGIN. '?nombre=' . $usuario ->
obtener_nombre());
```

```
    }
}
```

```
Conexion:: cerrar_conexion();
}
```

```
$titulo = 'Registro';
```

```
include_once 'plantillas/encabezado.inc.php';
?>
```

```
<div class="container">
    <div class="jumbotron">
        <h1 class="text-center">Formulario de registro</h1>
    </div>
</div>
```

```
<div class="container">
    <div class="row">
        <div class="col-md-6 text-center">
            <div class="panel panel-default">
                <div class="panel-heading">
                    <h3 class="panel-title">
                        Instrucciones
                    </h3>
                </div>
            </div>
        </div>
    </div>
</div>
```

```

<div class="panel-body">
  <br>
  <p class="text-justify">
    Para unirse al blog de AST (Analizador de Sensores Termicos),
    introduce un nombre de usuario y una contraseña, para que
empieces
    a utilizar todas las herramientas que presentamos para el analisis
    en tiempo real de datos de sensores termicos.
    Te recomendamos que uses una contraseña que contenga letras
    minúsculas, mayúsculas, números y símbolos.
  </p>
  <br>
  <a href="login.php">¿Ya tienes cuenta?</a>
  <br>
  <br>
  <br>
</div>
</div>
<div class="col-md-6 text-center">
  <div class="panel panel-default">
    <div class="panel-heading">
      <h3 class="panel-title">
        Introduce tus datos
      </h3>
    </div>
    <div class="panel-body">
      <form role="form" method="post">
        <?php
          if (isset($_POST['enviar'])) {
            include_once 'plantillas/registro_validado.inc.php';
          } else {
            include_once 'plantillas/registro_vacio.inc.php';
          }

        ?>
      </form>
    </div>
  </div>
</div>
</div>
</div>
<?php
include_once 'plantillas/piePag.inc.php';
?>

```

Login php:

```
<?php
include_once 'app/config.inc.php';
include_once 'app/Conexion.inc.php';
include_once 'app/RepositorioUsuario.inc.php';
include_once 'app/ValidadorLogin.inc.php';
include_once 'app/Redireccion.inc.php';
include_once 'app/ControlSesion.inc.php';

/**
 * Pantalla de login, en donde se verifica los datos como usuario y contraseña
 * con los cuales esta iniciando sesion. Posteriormente, se redirige a la pantalla
 * principal.
 */
if (ControlSesion :: sesion_iniciada()){
    Redireccion::redirigir(RUTA_PAG_PRINCIPAL);
}

if (isset($_POST['login'])) {
    Conexion::abrir_conexion();

    $validador = new ValidadorLogin($_POST['nombre'], $_POST['clave'],
    Conexion::obtener_conexion());

    if ($validador->obtener_error() == "" &&
        !is_null($validador->obtener_usuario())) {

        $id_user = $validador -> obtener_usuario()->obtener_id() ;
        $name_user = $validador -> obtener_usuario()->obtener_nombre();
        ControlSesion::iniciar_sesion($id_user,$name_user);
    }
}
```



```
        Redireccion::redirigir(RUTA_PAG_PRINCIPAL);
    }
    Conexion::cerrar_conexion();
}
```

```
$titulo = 'Login';
```

```
include_once 'plantillas/encabezado.inc.php';
?>
```

```
<div class="container">
    <br>
    <br>
    <br>
    <br>
    <br>
    <br>
    <div class="row">
        <div class="col-md-3">
        </div>
        <div class="col-md-6">
            <div class="panel panel-default">
                <div class="panel-heading text-center">
                    <h4>Iniciar sesión</h4>
                </div>
                <div class="panel-body text-center">
                    <form role="form" method="post|get"
action="pantallaGrafico.php">
                        <h2>Introduce tus datos</h2>
```

```

        <br>
        <label for="nombre" class="sr-only">Usuario</label>
        <input type="text" name="nombre" id="nombre" class="form-
control" placeholder="usuario"
        <?php

            if (isset($_POST['login']) && isset($_POST['nombre']) &&
!empty($_POST['nombre'])) {
                echo 'value="' . $_POST['nombre'] . "'";
            }
        ?>
        required autofocus>
        <br>
        <label for="clave" class="sr-only">Contraseña</label>
        <input type="password" name="clave" id="clave" class="form-
control" placeholder="Contraseña" required>
        <br>
        <?php
        if (isset($_POST['login'])) {
            $validador->mostrar_error();
        }
        ?>
        <button type="submit" name="login" class="btn btn-lg btn-
primary btn-block">
            Iniciar sesión
        </button>
        <br>
        <br>
        <a href="index.php">Crear cuenta</a>
    </form>
</div>
</div>
</div>
</div>
</div>

```

PantallaGrafico.php:

```
<!DOCTYPE html>
<?php
include_once 'app/Conexion.inc.php';
include_once 'app/Equipo.inc.php';
include_once 'app/Datos.inc.php';
include_once 'app/Accion.inc.php';
include_once 'app/usuario.inc.php';
include_once 'app/RepositorioDatos.inc.php';
include_once 'app/RepositorioEquipos.inc.php';
include_once 'app/RepositorioUsuario.inc.php';
include_once 'app/RepositorioAccion.inc.php';
include_once 'app/ValidarEquipo.inc.php';
include_once 'app/ControlSesion.inc.php';

//Se obtiene el id del usuario a partir del nombre con el que inicio sesion.
if (isset($_GET['nombre'])) {
    Conexion :: abrir_conexion();
    $usuario = RepositorioUsuario:: obtener_usuario_por_nombre(Conexion :: obtener_conexion(), $_GET['nombre']);
    Conexion::cerrar_conexion();
    $idUser = $usuario -> obtener_id();
}

// $idUser = '2';

//Se valida overifica el codigo y nombre del equipo que desea registrar en la plataforma.
if (isset($_POST['enviarEq'])) {

    Conexion :: abrir_conexion();
    $validador_equipo = new ValidarEquipo($_POST['idEquipo'], $_POST['nombreEquipo'], Conexion :: obtener_conexion());

    if ($validador_equipo->equipo_valido()) {
        $equipo_insertar = new Equipo($validador_equipo->obtener_idEquipo(), $validador_equipo->obtener_nombEquipo(), $idUser);

        $actuador_insertar = new Accion("", $validador_equipo->obtener_idEquipo(), "0");

        $equipo_insertado = RepositorioEquipos:: insertar_equipo(Conexion::obtener_conexion(), $equipo_insertar);

        $actuador_insertado = RepositorioAccion:: insertar_accion(Conexion::obtener_conexion(), $actuador_insertar);

        if ($equipo_insertado) {
            //print 'aquito validado correctamente';
        }
    }
    Conexion::cerrar_conexion();
}

/*Permite seleccionar el equipo con el cual desea interactuar para mostrar la grafica lineal con los valores de temperatura
 * y para el control del dispositivo, asi como de la tabla con los respectivos valores.
 */

if (isset($_POST['selectEquipo'])) {

    $data_equipo = explode("/", $_POST['allEquipos']);

    Conexion :: abrir_conexion();
    $actuador_info = RepositorioAccion::obtener_estado(Conexion :: obtener_conexion(), $data_equipo[0]);

    $datos_temp = RepositorioDatos::obtener_datos_por_idEqp(Conexion :: obtener_conexion(), $data_equipo[0]);
    Conexion::cerrar_conexion();
    if (count($datos_temp)) {
        $datos_graph = array();
        foreach ($datos_temp as $info) {
            $fecha = explode(" ", $info->obtenerFecha());
            $dato = [$fecha[1], $info->obtenerTemperatura()];
            $datos_graph[] = $dato;
        }
    }
}

/*
if (isset($_POST['aumentar'])) {
    print $actuador_info->obtener_estado();
    $valor = $actuador_info->obtener_estado();
    print $_POST['aumentar'];
    $error = "";
    if ($valor >= 0) {
        $valor += 1;
        Conexion :: abrir_conexion();
        $actualizar = RepositorioAccion:: actualizar_estado(Conexion::obtener_conexion(),
            $actuador_info->obtener_idAccion(), $valor);

        Conexion::cerrar_conexion();
    }
}
```

```

    } else {
        $error = "El dispositivo esta en su mas bajo estado";
    }
}

*/

include_once 'plantillas/encabezado.inc.php';
include_once 'plantillas/navbar.inc.php';
?>

<div class="jumbotron jumbotron-fluid">
    <div class="container">
        <h1 class="text-center">Graficas termicas</h1>
    </div>
</div>
<div class="container">
    <div class="row">
        <div class="col-md-4 text-center">
            <div class="panel panel-default">
                <div class="panel-heading">
                    <h3 class="panel-title"> Ingresar nuevo equipo</h3>
                </div>
                <div class="panel-body" >
                    <br>
                    <form role="form" method="post" action="<?php echo $_SERVER['PHP_SELF'] ?>">
                        <div class="form-group">
                            <label class="mr-sm-2">ID del equipo:</label>
                            <input type="text" class="form-group mb-2 mr-sm-2" id="idEqp" placeholder="ID equipo" name="idEquipo">
                            <?php
                                if (isset($_POST['enviarEqp'])) {
                                    $validador_equipo->mostrar_error_idEquipo();
                                }
                            ?>
                        </div>
                        <div class="form-group">
                            <label class="mr-sm-2">Nombre del equipo:</label>
                            <input type="text" class="form-group mb-2 mr-sm-2" id="nameEquipo" placeholder="Nombre" name="nombreEquipo">
                            <?php
                                if (isset($_POST['enviarEqp'])) {
                                    $validador_equipo->mostrar_error_nomEquipo();
                                }
                            ?>
                        </div>
                        <div class="form-group">
                            <label class="mr-sm-2">Nombre del equipo:</label>
                            <input type="text" class="form-group mb-2 mr-sm-2" id="nameEquipo" placeholder="Nombre" name="nombreEquipo">
                            <?php
                                if (isset($_POST['enviarEqp'])) {
                                    $validador_equipo->mostrar_error_nomEquipo();
                                }
                            ?>
                        </div>
                    <br>
                    <button type="submit" class="btn btn-default btn-primary mb-2" name="enviarEqp">Submit</button>
                </form>
            </div>
        </div>
    </div>
</div>

<div class="col-md-4 text-center"></div>

<div class="col-md-4 text-center">
    <div class="panel panel-default">
        <div class="panel-heading">
            <h3 class="panel-title"> Sesion de control de temperatura</h3>
        </div>
        <div class="panel-body" >
            <br>
            <form role="form" method="post" action="<?php echo $_SERVER['PHP_SELF'] ?>">
                <div class="form-group">
                    <select id="allEquipos" name="allEquipos" onchange="ShowSelected();">

                        <?php
                            include_once 'plantillas/listaEquipos.inc.php';
                        ?>
                    </select>
                    <br><br>
                    <button type="submit" class="btn btn-default btn-primary mb-2" name="selectEquipo">Submit</button>
                </div>
            </form>
        </div>
    </div>
</div>

```

```

        <form role="form" method="post" name="control" action="<?php echo $_SERVER['PHP_SELF'] ?>">
            <div class="form-group">
                <input type="text" method="post" id="cantidad" name="cantidad"
                <?php
                if (isset($_POST['selectEquipo'])) {
                    echo 'value="' . $actuador_info->obtener_estado() . '"';
                }
                ?>
                <input type="text" value="" class="form-control" />
            </div>
            <div class="form-group">
                <div class="row">
                    <div class="col-md-6 text-center">
                        <input type="button" id="aumentar" name="aumentar" value="aumentar" onclick="control.cantidad.value++;
                        validaForm(this.form)"><i class="fas fa-angle-double-up"></i>
                    </div>
                    <div class="col-md-6 text-center">
                        <input type="button" id="disminuir" name="disminuir" value="disminuir" onclick="control.cantidad.value--;
                        validaForm(this.form)"><i class="fas fa-angle-double-down"></i>
                    </div>
                </div>
            </div>
            <br>
            <br>
        </form>
    </div>
    <div>
        <br>
        <br>
        <br>
    </div>
</div>
<div class="container">
    <div>

```

```

        <div id="curve_chart"></div>
    </div>
    <br>
    <br>
    <br>
    <br>
    <table class="table">
        <thead class="thead-dark">
            <tr>
                <th>id</th>
                <th>Fecha</th>
                <th>Temperatura</th>
            </tr>
        </thead>
        <tbody>
            <?php
            include_once 'plantillas/tablaDatos.inc.php';
            ?>
        </tbody>
    </table>
    <div class="panel-heading">
        <span class="glyphicon glyphicon-time"></span>
    </div>
    <div class="panel-body">
        <div id="curve_chart"></div>
    </div>
</div>

<?php
include_once 'plantillas/piePag.inc.php';
?>

```

listaEquipos.inc.php:

```
<?php

/*
 * Se obtiene todos los equipos registrados con el usuario y
 * son mostrados en un select view.
 */

Conexion :: abrir_conexion();

$idEquipos = RepositorioEquipos ::
obtener_equipos_por_idUser(Conexion::obtener_conexion(),$idUser);
foreach ($idEquipos as $fila){

    $nombreEqp = $fila -> obtener_nombreEqp();
    $idEqp = $fila->obtener_idEquipo();

    $texto = $idEqp."/". $nombreEqp;
    echo "<option value= $texto >$nombreEqp</option>";

}

Conexion::cerrar_conexion();
```

Encabezado:

```
<html>
<head>
<title>Termical Analyzer</title>
<meta charset="UTF-8">
<meta http-equiv="X-UA-compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">

<?php
if (!isset($titulo) || empty($titulo)) {
    $titulo = 'AST';
}

echo "<title>$titulo</title>";
?>

<link href="css/bootstrap.min.css" rel="stylesheet">
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.cs
s" integrity="sha384-
Vkoo8x4CGsO3+Hhvx8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifj
h" crossorigin="anonymous">
<link rel="stylesheet" href="css/styles.css">
</head>
<body>
```

8:48 a. m.

piePag.inc.php:

En este se muestran todos los scripts utilizados e incluso los que permiten realizar la gráfica lineal utilizando GoogleChart.

```
<script type="text/javascript">
function validaForm(form) {
    if (form.cantidad.value >= 1 && form.cantidad.value <= 5) {
        if (form.cantidad.value !== "<?php echo $actuador_info->obtener_estado(); ?>" ) {
            <?php /*
                Conexion :: abrir_conexion();
                $actualizar = RepositorioAccion:: actualizar_estado(Conexion::obtener_conexion(),
                    $actuador_info->obtener_idAccion(), '<script> document.write(form.cantidad.value) </script>');

                Conexion::cerrar_conexion();*/
            ?>
            alert("<?php echo $actuador_info -> obtener_estado(); ?>");
        }
    }
    if (form.cantidad.value < 1) {
        form.cantidad.value = 1;
        alert("El dispositivo se encuentra en su estado mas bajo");
    }
    if (form.cantidad.value > 5) {
        form.cantidad.value = 5;
        alert("El dispositivo se encuentra en su estado mas alto");
    }
}
</script>
</body>
</html>
```

tablaDatos.inc.php:

Permite obtener todos los datos de acuerdo al equipo seleccionado en el list view.

```
<?php
/**
 * Permite obtener todos los datos de acuerdo al equipo seleccionado en el list
view
 */
if (isset($_POST['selectEquipo'])) {
    foreach ($datos_temp as $fila) {
        $idDato = $fila -> obtenerIdDato();
        $fecha = $fila -> obtenerFecha();
        $temperatura = $fila -> obtenerTemperatura();

        echo '<tr class = "'.table-light.'">';
        echo "<td>$idDato</td>";
        echo "<td>$fecha</td>";
        echo "<td>$temperatura</td>";
        echo "</tr>";
    }
}
```

registro_validado.inc.php:

Fragmento que permite visualizar algún error en cuanto a la validación de los campos ingresados para el registro de un usuario.

```
<div class="form-group">
  <label>Nombre de usuario</label>
  <input type="text" class="form-control mb-2 mr-sm-2" name="nombre"
placeholder="usuario" <?php $validador -> mostrar_nombre() ?>>
  <?php
    $validador -> mostrar_error_nombre();
  ?>
</div>
<div class="form-group">
  <label>Contraseña</label>
  <input type="password" class="form-control mb-2 mr-sm-2"
name="clave1">
  <?php
    $validador -> mostrar_error_clave1();
  ?>
</div>
<div class="form-group">
  <label>Repita la contraseña</label>
  <input type="password" class="form-control mb-2 mr-sm-2"
name="clave2">
  <?php
    $validador -> mostrar_error_clave2();
  ?>
</div>
<br>
<button type="reset" class="btn btn-default btn-primary">Limpiar
formulario</button>
<br>
<br>
```

Styles.css:

Archivo con todos los estilos utilizados, como colores espaciados, bordes, etc.


```

body {
    background-image: url('../img/fondo2.jpg');
}

.jumbotron {
    background-color: #9E4633;
    color: white;
    border-bottom-color: black;
    border-bottom-width: 5;
    border-style: solid;
}

.panel-heading {
    background-color: #9E4633 !important;
    color: white !important;
    border-bottom-color: black;
    border-bottom-width: 5;
    border-style: solid;
}

.form-group {
    margin-left: 30px;
    margin-right: 30px;
}

.panel-body {
    color: black !important;
    background-color: white !important;
    padding-left: 30px;
    padding-right: 30px;
    padding-bottom: 20px;
}

.navbar {
    background-color: #2b2b2b;
}

.navbar-brand:link, .navbar-brand:visited {
    color: deepskyblue !important;
}

.navbar-brand:hover, .navbar-brand:active {
    color: white !important;
}

#navbar li a:link, #navbar li a:visited {
    color: white !important;
}

#navbar li a:hover, #navbar li a:active {
    color: deepskyblue !important;
}

table, th, td {
    border: 1px solid black;
}

.dropdown-menu {
    background-color: #2b2b2b;
}

.open > a {
    background-color: #2b2b2b !important;
}

.btn-primary {
    background-color: deepskyblue !important;
    border: none;
}

.btn-primary:hover {
    background-color: #2b2b2b !important;
}

.contenido-articulo, .perfil {
    background-color: white;
    padding: 2em;
}

```

Accion.inc.php:

```
<?php

/**
 * Se crea un objeto tipo Accion con los parametros de la tabla accion
 * id_accion, id_equipo, estado
 */
class Accion{
    private $id_accion;
    private $id_Equipo;
    private $estado;

    public function __construct($id_accion, $id_Equipo, $estado ) {
        $this -> id_accion = $id_accion;
        $this -> id_Equipo = $id_Equipo;
        $this-> estado = $estado;
    }

    public function obtener_idEquipo() {
        return $this-> id_Equipo;
    }
    public function obtener_idAccion() {
        return $this-> id_accion;
    }

    public function obtener_estado() {
        return $this-> estado;
    }

    public function cambiar_estado($estado) {
        return $this-> estado = $estado;
    }
}
```

ANÁLISIS DE PRESUPUESTO

Debido a que los sistemas intangibles utilizados son de software libre, es decir sin costo, solo se tomaría en cuenta el hardware implementado a lo largo del proyecto.

Nombre	Precio c/u (\$)	Cantidad	Precio Tota(\$)
Módulo wifi esp8266	12	2	24
Arduino UNO	17	4	68
Sensor DHT22	2	3	6
Módulo IR receptor/emisor	8	1	8
Batería 9v	3	5	15
		Total	113

Podemos observar que el costo total del proyecto es elevado en cuanto a los elementos que se deberían adquirir para hacer otro igual, sin embargo, nuestro grupo ya contaba con los microcontroladores Arduino necesarios y las baterías de 9v. Por ello, el costo de este proyecto fue en realidad de 65\$.

Este valor puede ser reducido si se busca una manera más eficiente de elaborar el mismo proyecto, quizá con otros componentes que sean más baratos o un sistema de conexiones que no requiera tantos elementos.

CONCLUSIONES

- Se diseñó una página web de tal manera que el usuario pueda no solo visualizar los datos de temperatura de diferentes sectores de una habitación mediante graficas sino también poder interactuar con ella para realizar una personalización automática de los aires.
- Se logró recoger datos de temperatura de cada sector de la habitación mediante el uso de módulos de sensores, además de poder enviarlos, a partir de una conexión wifi al Arduino, a una base de datos MySQL no solo con el valor de dicha temperatura sino también de acuerdo la fecha en que fueron tomados esos datos.
- Se logró diseñar un control que emitiera señales infrarrojas hacia los sistemas de refrigeración, para el cambio de temperatura hacia diferentes sectores de una

habitación, de acuerdo a la orden que se enviara desde la página web desarrollada e interactuada por el usuario.

RECOMENDACIONES

- Para simplificar el prototipo final o hacerlo mas pequeño, se puede utilizar un Arduino nano, sin embargo, estos solo tienen un puerto serial.
- Al momento de colocar el sensor de temperatura, se recomienda que este al menos a 5 metros del dispositivo refrigerante, con el fin de que mida la temperatura del cuarto o zona de medición y no cense la temperatura del dispositivo refrigerante en sí.